

Implementation of Multiple Web Servers In Single Network

A project submitted in partial fulfillment of the requirements
for the degree of Bachelor of Science in Computer Science &
Engineering (CSE)

Name of Students	Roll No.
Md. Mamanur Rashid	(47)
Md. NasirUddin	(44)
MostafizurRahman	(10)
Md. LokmanHossan	(09)
Md. Anwarul Islam	(16)

**Department of Computer Science & Engineering
Dhaka International University**

APPROVAL

The project titled “Implementation of Multiple Web Servers in Single Network” submitted by Md. Mamunur Rashid(roll-47), Md. Nasir Uddin (roll-44), Mostafizur Rahman(roll-10), Lokman Hossan (roll-09), Md. Anwarul Islam (roll-16) CSE Evening 37th batch to the Department of Computer Science & Engineering at Dhaka International University, Bangladesh has been accepted as satisfactory for the partial fulfillment of the requirements for the Degree of Science and Engineering and approved as to its style and content .

Supervised By

.....

Md. Tahzib-Ul-Islam

Assistant Professor

Dhaka International University

Department of CSE

DECLARATION

With pleased declare that work presented in this project is successful and the outcome is carefully investigated supervisor Md. Tahzib-Ul-Islam, Assistant Professor of Department of Computer Science & Engineering. Dhaka International University, Dhaka, Bangladesh. project work performed by Md. Mamunur Rashid, Md. Nasir Uddin, Mostafizur Rahman, Md. Anwarul Islam, Lokman Hossan, the student of Bachelor of Science in Computer Science & Engineering (B.Sc.) here by solemnly declare that the work presented in this project has not been previously submitted to any other university/college/ or generation for any academic qualification/certificate/degree.

.....
Md. Mamunur Rashid(47) Reg:275799	Md. Nasir Uddin (44) Reg:275795
.....
Md. Anwarul Islam (16) Reg:275767	Lokman Hossan (09) Reg:275760
.....	
Mostafizur Rahman(10) Reg:275761	
Signature of Candidates	

ABSTRACT

As the web server based system is being used more and more, having separate web servers for each task to distribute the web server's load are gaining much more popularity over having one main web server to process all the tasks. When the user tries to access each web server that contains a number of web documents that are linked to each other via hyper-links within the domain, each web server asks the user to follow the verification process even though the user is identical, and this project the user from using the system efficiently. The role based access control method, which is the most suitable access control concept available now for the distributed web server based system within the domain, is used in this paper. Additionally the method for controlling the level of web document contents available to the user based on the user's access permission rights is introduced to reduce the granularity of the document content access.

ACKNOWLEDGEMENTS

First and foremost, we are grateful to Allah, the almighty and merciful, without whose patronage and blessing this project would not have been successfully completed. He gave us zeal, confidence, power of determination and courage and vanquished all the stumbling hardness that we faced on the way.

It is an auspicious occasion for us as students of Department of CSE, one of the prestigious academic centers of the Dhaka International University, to express our deep feelings of gratitude to the department and especially to our supervisor, Head of the Department, teachers and also to the departmental staff.

We would like to thank our supervisor, **Md.Tahzib-Ul-Islam**, Asst. Professor, Department of CSE, Dhaka International University for his effective suggestion, guidance, inspiration, co-operation and valuable advice and special supervision in the preparation of this report.

Specially, we would like to thank, **Associate. Prof. Md. Abdul Baset** Chairman, Department of CSE, Dhaka International University for providing us a good educational environment and proper guideline towards of the project paper.

Specially, we would like to thank, **Prof. Dr. A.T.M Mahbubur Rahman** Dean, Faculty of Science & Engineering, Dhaka International University for providing us a good educational environment and proper guideline towards of the project paper.

Also we would like to give special thanks to, **Prof. Dr. Hafiz Md. Hasan Babu**, Adviser, Department of CSE, Dhaka International University for his very valuable directives and special attention.

Our parents are very much keen and hopeful in the best performance of the dissertation we are going to submit. We wish we could fulfill their aspiration. We also pay regards to our friends in the department who, through their interest and work, are our constant source of inspiration.

INDEX

Contents	Page No.
Approval	I
Declaration	II
Abstract	III
Acknowledgement	IV
Contents	V- VI
List of Figure	VII-XII
Chapter 1: Introduction	01-03
1.1 Overview.....	02
1.2 Objective	02
1.3 Motivation.....	02
1.4 Our study.....	02
1.5 Book outline.....	02
Summary.....	03
Chapter 2: Background Study	04-07
2.1 Linux operating system	05
2.2 Component of Linux operating system	05
2.3 Package installion.....	05
2.4 HTTP web server.....	05
2.5 Virtual web server.....	06
2.6 HTTPS web server.....	06
2.7 Process of restricted.....	07
Summary.....	07
Chapter 3: Server configuration	08-66
3.1 Package installation	09

3.2.1 FTP server.....	09
3.2.2 YUM configuration.....	15
3.3 HTTP web server.....	24
3.4 Virtual web server.....	37
3.5 Step of restriction process.....	45
3.6 HTTPS web server.....	53
3.7 Network diagram.....	65
Summary.....	66

Chapter 4: Server implementation 67-77

4.1 FTP server.....	68
4.2 HTTP server.....	69
4.3 Virtual server.....	71
4.4 Process of restricted.....	73
4.5 HTTPS server.....	75
Summary.....	77

Chapter 5: Conclusion 78-86

5.1 Outcome.....	79
5.2 Limitation.....	79
5.3 Future scope.....	79
5.4 Conclusion	79
Bibliography.....	80
Troubleshooting.....	81

List of figures

Page no.

Figure 3.1.....	09
Figure 3.2.....	10
Figure 3.3	10
Figure 3.4	11
Figure 3.5.....	11
Figure 3.6.....	12
Figure 3.7.....	12
Figure 3.7.1.....	13
Figure 3.7.2.....	13
Figure 3.7.3.....	14
Figure 3.7.4.....	14
Figure 3.8.....	15
Figure 3.9.1.....	15
Figure 3.9.2.....	16
Figure 3.9.3.....	16
Figure 3.9.4.....	17
Figure 3.9.5.....	17
Figure 3.9.6.....	18
Figure 3.9.7.....	18
Figure 3.9.8.....	19
Figure 3.9.9.....	19
Figure 3.9.10.....	20
Figure 3.9.11.....	20
Figure 3.9.12.....	21

Figure 3.9.13.....	21
Figure 3.9.14.....	22
Figure 3.9.15.....	22
Figure 3.9.16.....	23
Figure 3.9.17.....	23
Figure 3.10.1.....	24
Figure 3.10.2.....	24
Figure 3.10.3.....	25
Figure 3.10.4.....	25
Figure 3.10.5.....	26
Figure 3.10.6.....	26
Figure 3.10.7.....	27
Figure 3.10.8.....	27
Figure 3.10.9.....	28
Figure 3.10.10.....	28
Figure 3.10.11.....	29
Figure 3.10.12.....	29
Figure 3.10.13.....	30
Figure 3.10.14.....	30
Figure 3.10.15.....	31
Figure 3.10.16.....	31
Figure 3.10.17.....	32
Figure 3.10.18.....	32
Figure 3.10.19.....	33
Figure 3.10.20.....	33

Figure 3.10.21.....	34
Figure 3.10.22.....	34
Figure 3.10.23.....	35
Figure 3.10.24.....	35
Figure 3.10.25.....	36
Figure 3.10.26.....	36
Figure 3.10.27.....	37
Figure 3.11.1.....	37
Figure 3.11.2.....	38
Figure 3.11.3.....	38
Figure 3.11.4.....	39
Figure 3.11.5.....	39
Figure 3.11.6.....	40
Figure 3.11.7.....	40
Figure 3.11.8.....	41
Figure 3.11.9.....	41
Figure 3.11.10.....	42
Figure 3.11.11.....	42
Figure 3.11.12.....	43
Figure 3.11.13.....	43
Figure 3.11.14.....	44
Figure 3.11.15.....	44
Figure 3.11.16.....	45
Figure 3.12.1.....	45
Figure 3.12.2.....	46

Figure 3.12.3.....	46
Figure 3.12.4.....	47
Figure 3.12.5.....	47
Figure 3.12.6.....	48
Figure 3.12.7.....	48
Figure 3.12.8.....	49
Figure 3.12.9.....	49
Figure 3.12.10.....	50
Figure 3.12.11.....	50
Figure 3.12.12.....	51
Figure 3.12.13.....	51
Figure 3.12.14.....	52
Figure 3.12.15.....	52
Figure 3.12.16.....	53
Figure 3.12.17.....	53
Figure 3.12.18.....	54
Figure 3.12.19.....	54
Figure 3.12.20.....	55
Figure 3.12.21.....	55
Figure 3.12.22.....	56
Figure 3.12.23.....	56
Figure 3.12.24.....	57
Figure 3.12.25.....	57
Figure 3.12.26.....	58
Figure 3.12.27.....	58

Figure 3.12.28.....	59
Figure 3.12.29.....	59
Figure 3.12.30.....	60
Figure 3.12.31.....	60
Figure 3.12.32.....	61
Figure 3.12.33.....	61
Figure 3.12.34.....	62
Figure 3.12.35.....	62
Figure 3.12.36.....	63
Figure 3.12.37.....	63
Figure 3.12.38.....	64
Figure 3.12.39.....	64
Figure 3.12.40.....	65
Figure 3.13.1.....	65
Figure 3.13.2.....	66
Figure 4.1.....	68
Figure 4.2.....	68
Figure 4.3.....	69
Figure 4.4.....	69
Figure 4.5.....	70
Figure 4.6.....	70
Figure 4.7.....	71
Figure 4.8.....	71
Figure 4.9.....	72
Figure 4.10.....	72

Figure 4.11.....	73
Figure 4.12.....	73
Figure 4.13.....	74
Figure 4.14.....	74
Figure 4.15.....	75
Figure 4.16.....	75
Figure 4.17.....	76
Figure 4.18.....	76
Figure 4.19.....	77
Figure 5.1.....	81
Figure 5.2.....	81
Figure 5.3.....	82
Figure 5.4.....	82
Figure 5.5.....	83
Figure 5.6.....	83
Figure 5.7.....	84
Figure 5.8.....	84
Figure 5.9.....	85
Figure 5.10.....	86

Chapter 1

Introduction

1.1 Overview

The primary function of a web server is to store, process and deliver web pages to clients. The communication between client and server takes place using the Hypertext Transfer Protocol (HTTP). Pages delivered are most frequently HTML documents, which may include images, style sheets and scripts in addition to text content.

1.2 Objectives

- Identify the key configuration files, log files, and content directories used by Apache httpd.
- Configure Apache httpd to provide IP-based and name-based virtual hosts.
- Configure Apache httpd to provide TLS-encrypted virtual hosts.
- Configure Apache httpd to serve dynamic database-driven web content.
- Configure the FTP server to store the data.
- Configure the YUM client server to serve all packages are stored.
- SSL engine on then all SSL file are activated.
- Hosts IP are set for the domain name.

Sections

- Configuring Apache HTTPD.
- Configuring and Troubleshooting Virtual Hosts.
- Configuring HTTPS.
- Integrating Dynamic Web Content.

1.3 Motivation

- Http web server all data information are static.
- Virtual web server is share multiple website by its domain name.
- Https can secure the web server main used of https is authentication of website and protection of the exchanged data.
- A dynamic web server contain information that will be changed depending on the viewers by few seconds of the day.

1.4 Our study

For doing this project we need to study about others different types of web server .The identification beyond these recent occurring web server tasks .study about the different typical web server system of different types of client.

1.5 Book outline

In this chapter describe the aim and objectives of this project. Rest of the book is organized as follows.

In chapter 2, it has been described the necessary studies required for the project completion.

In chapter 3, it has been create the overall server of the project.

In chapter 4, it has been described how to take input and showing output.

Summary

In this chapter the objectives and total overview of the project .The main objectives and how the full project is design and how it will be helpful for people and corporate office all are describe in this chapter. The rest of the description will explain in next chapter according to the outline.

Chapter 2

Background Study

2.1 Linux operating system

Linux is a multitasking, multiuser operating system that is known for its stability. Although modified by numerous people, its robustness stems from its Unix-like architecture that keeps applications isolated from the core operating system. Linux is a Unix-like, open source and community-developed operating system for computers, servers, mainframes, mobile devices and embedded devices. It is supported on almost every major computer platform including x86, ARM and SPARC, making it one of the most widely supported operating systems. The Linux open source operating system, or Linux OS, is a freely distributable, cross-platform operating system based on Unix that can be installed on PCs, laptops, netbooks, mobile and tablet devices, video game consoles, servers, supercomputers and more [1].

2.2 Components of Linux System

Linux Operating System has primarily three components

Kernel: Kernel is the core part of Linux. It is responsible for all major activities of this operating system. It consists of various modules and it interacts directly with the underlying hardware. Kernel provides the required abstraction to hide low level hardware details to system or application programs.

System Library: System libraries are special functions or programs using which application programs or system utilities accesses kernel's features. These libraries implements most of the functionalities of the operating system and do not require kernel module's code access rights[2].

System Utility: System Utility programs are responsible to do specialized, individual level tasks.

2.3 Package installation

FTP server: The File Transfer Protocol (FTP) is the standard network protocol used for the transfer of computer files between a client and server on a computer network. FTP is built on a client-server model architecture and uses separate control and data connections between the client and the server. FTP users may authenticate themselves with a clear-text sign-in protocol, normally in the form of a username and password, but can connect anonymously if the server is configured to allow it. For secure transmission that protects the username and password, and encrypts the content, FTP is often secured with SSL/TLS (FTPS). SSH File Transfer Protocol (SFTP) is sometimes also used instead; it is technologically different [3].

YUM installation: Yellow dog update, modified (YUM) is a program that manages installation, updates and removal for red hat package manager (RPM) systems. YUM allows the user to update groups of machines without having to update each RPM separately [4].

2.4 HTTP web server

A web server is a computer system that processes requests via HTTP, the basic network protocol used to distribute information on the World Wide Web. The term can refer to the entire system, or specifically to the software that accepts and supervises the HTTP requests. HTTP means Hypertext Transfer Protocol. Http web server is a static web server. Because it contains all data information are static.

The primary function of a web server is to store, process and deliver web pages to clients. The communication between client and server takes place using the Hypertext Transfer Protocol (HTTP). Pages delivered are most frequently HTML documents, which may include images, style sheets and scripts in addition to text content. A user agent, commonly a web browser or web crawler, initiates communication by making a request for a specific resource using HTTP and the server responds with the content of that resource or an error message if unable to do so. There source is typically a real file on the server's secondary storage, but this is not necessarily the case and depends on how the web server is implemented. While the primary function is to serve content, a full implementation of HTTP also includes ways of receiving content from clients. This feature is used for submitting web forms, including uploading of files [5].

2.5 Virtual web server

A server, usually a web server that shares computer resources with other virtual servers. In this context, the virtual part simply means that it is not a dedicated server-- that is, the entire computer is not dedicated to running the server software. Virtual web servers are a very popular way of providing low-cost web hosting services. Instead of requiring a separate computer for each server, dozens of virtual servers can co-reside on the same computer. In most cases, performance is not affected and each web site behaves as if it is being served by a dedicated server. However, if too many virtual servers reside on the same computer, or if one virtual server starts hogging resources, web pages will be delivered more slowly [6].

2.6 HTTPS web server

HTTP Secure (HTTPS) is an adaptation of the Hypertext Transfer Protocol (HTTP) for secure communication over a computer network, and is widely used on the internet. In HTTPS, the communication protocol is encrypted by Transport Layer Security (TLS), or formerly, its predecessor, Secure Sockets Layer (SSL).

A large portion of webpages found on the internet use HTTP, most often because the information being requested is less likely to need to be secure. For example, accessing a news article is unlikely to involve the exchange of sensitive information. You can tell if a page is using HTTPS by looking at the URL displayed in the web browser. A webpage using HTTPS will start with https://.

Sending or accessing sensitive information on a website is much more likely to be conducted through HTTPS because of its increased level of encryption. Accessing a brokerage account or online banking platform, for example, involves the exchange of information that a user would likely want to remain secure.

Request and response data is encrypted using a security layer such as Secure Sockets Layer (SSL) or Transport Layer Security (TLS). In order to speed up encryption, websites place SSL or TLS certificates in libraries where the encoding and decoding of website information takes place.

When navigating a webpage the browser automatically looks at the URL to determine whether the page is using HTTP or HTTPS. If the page is using HTTPS the browser exchanges some SSL parameters with the web server, and then opens up a secure connection. The web browser automatically seeks out the certificate, which doesn't require any action on the user's part in most situations [7].

2.7 Process of restricted

You can authorize access to the content on your web site in two ways. If your site uses an authentication method that identifies its users, you can set up authorization rules that allow access to some users and deny access to others. If you want to authorize access based on domain names or an IP address space, you can use IP and domain restriction rules to allow access to some domains and deny access to others. You can mix and match these rules to configure an authorization scheme that provides the best level of security for your content.

Summary

A web server is a computer that stores web server software and a web sites component files. Support physical data interchange with other devices connected to the web.it process the request via HTTP .the basic network protocol used to distribute information the www (World Wide Web).

Chapter 3

Server Configuration

3.1 Package installation

YUM installation: Yellow dog update, modified (YUM) is a program that manages installation, updates and removal for red hat package manager (RPM) systems. YUM allows the user to update groups of machines without having to update each RPM separately [8].

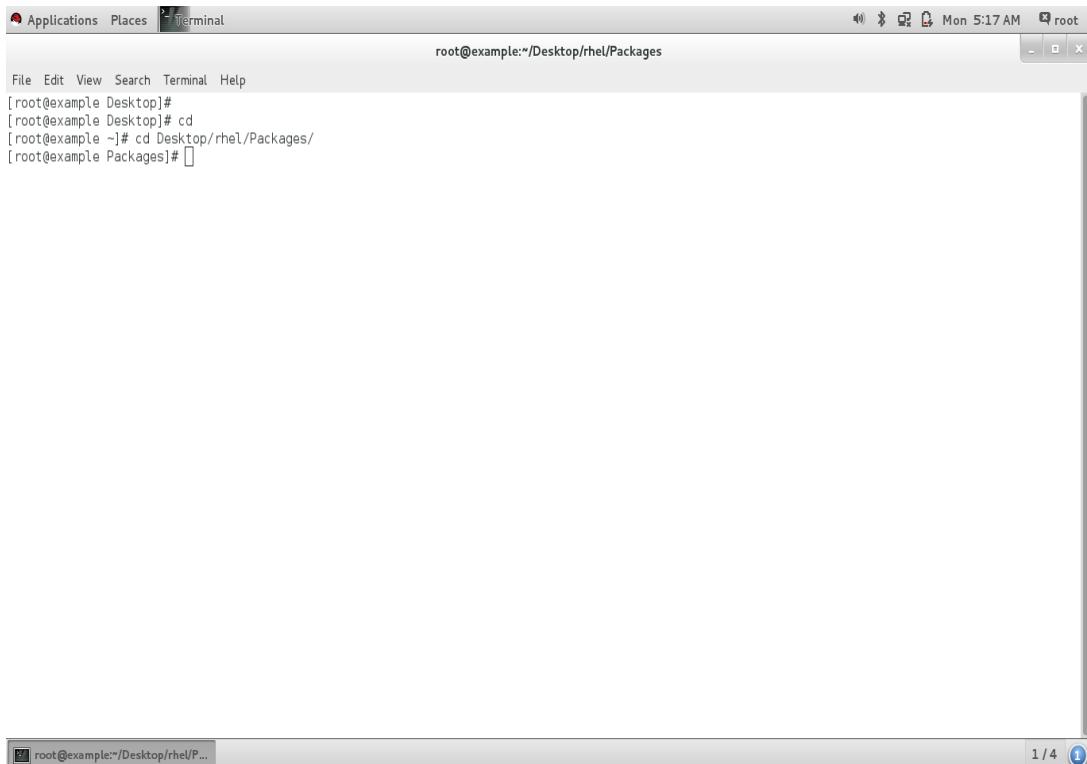
3.2 FTP server

Step 3.2.1.0

- Open the ISO file then copy ISO file in a folder.

Step 3.2.1.1

- Now press (cd Desktop/rhel/packages) to enter the folder.



The screenshot shows a terminal window titled 'Terminal' with a grey header bar containing 'Applications', 'Places', and 'Terminal'. The window title bar also shows the path 'root@example:~/Desktop/rhel/Packages'. The main area of the terminal displays the following command history:

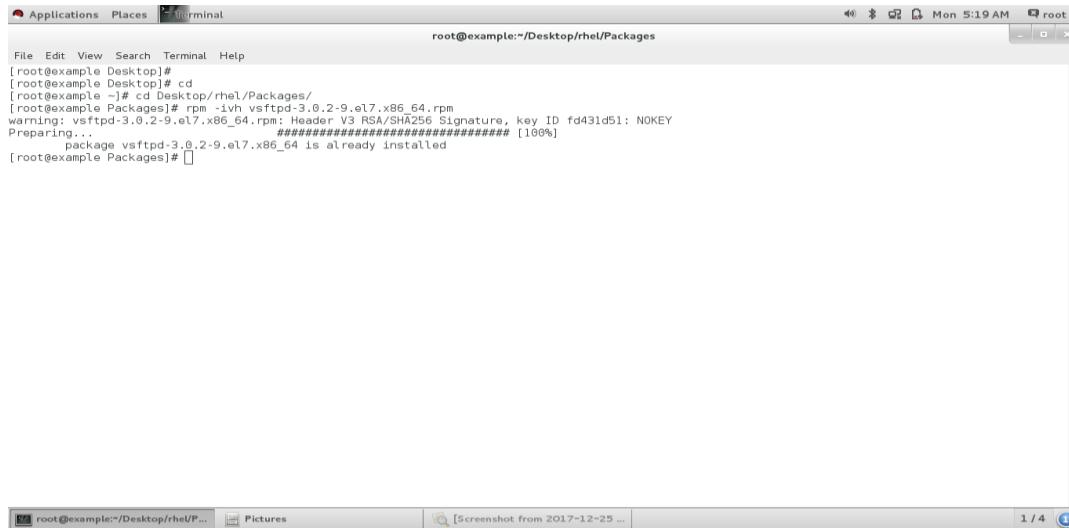
```
File Edit View Search Terminal Help
[root@example Desktop]#
[root@example Desktop]# cd
[root@example ~]# cd Desktop/rhel/Packages/
[root@example Packages]#
```

The terminal window has a dark grey background and light grey text. At the bottom right, there is a status bar with the text '1 / 4' and a small blue circular icon.

Figure 3.1: Copy ISO file

Step 3.2.1.2

- Press (rpm –ivvhftpd) for installed packages.

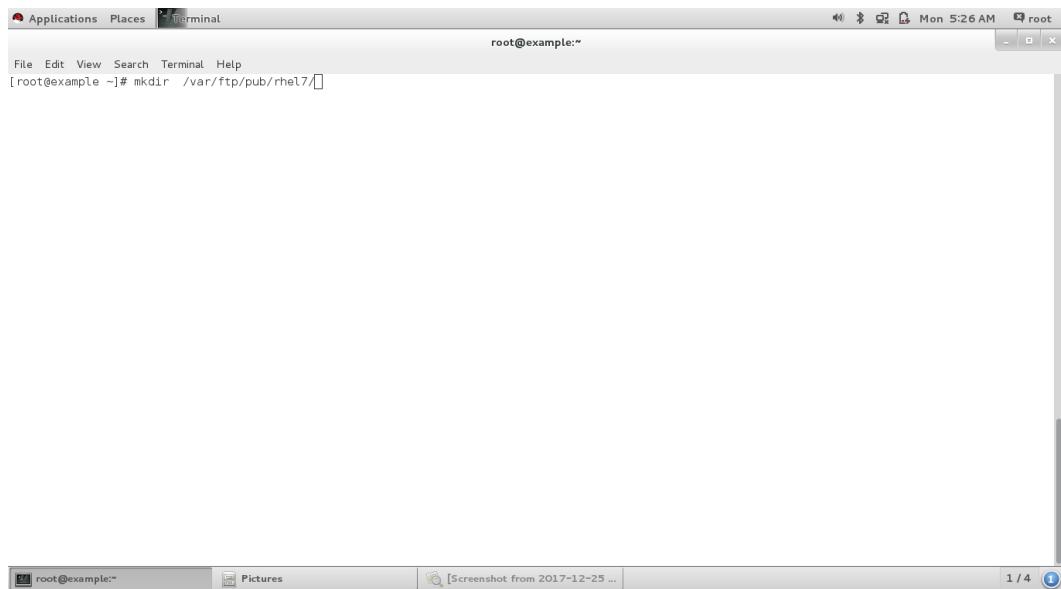


A screenshot of a terminal window titled "Terminal". The window shows a command-line session as root. The user runs the command "rpm -ivh vsftpd-3.0.2-9.el7.x86_64.rpm". The output indicates that the package is already installed, with a warning message about a missing GPG signature and a note that the key ID is NOKEY. The terminal window has a standard Linux desktop interface at the bottom, including icons for Applications, Places, and a search bar.

Figure 3.2: Install packages

Step 3.2.1.3

- This is used for create a folder or directory.
- This is the main path of FTP.

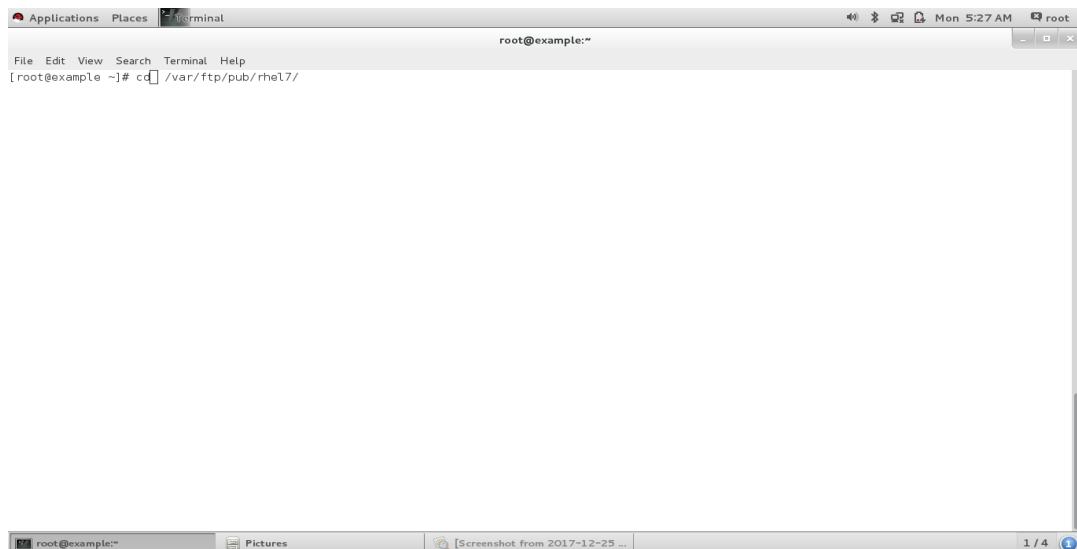


A screenshot of a terminal window titled "Terminal". The window shows a command-line session as root. The user runs the command "mkdir /var/ftp/pub/rhel7". The terminal window has a standard Linux desktop interface at the bottom, including icons for Applications, Places, and a search bar.

Figure 3.3: Folder create

Step 3.2.1.4

- To checked the folder creation.

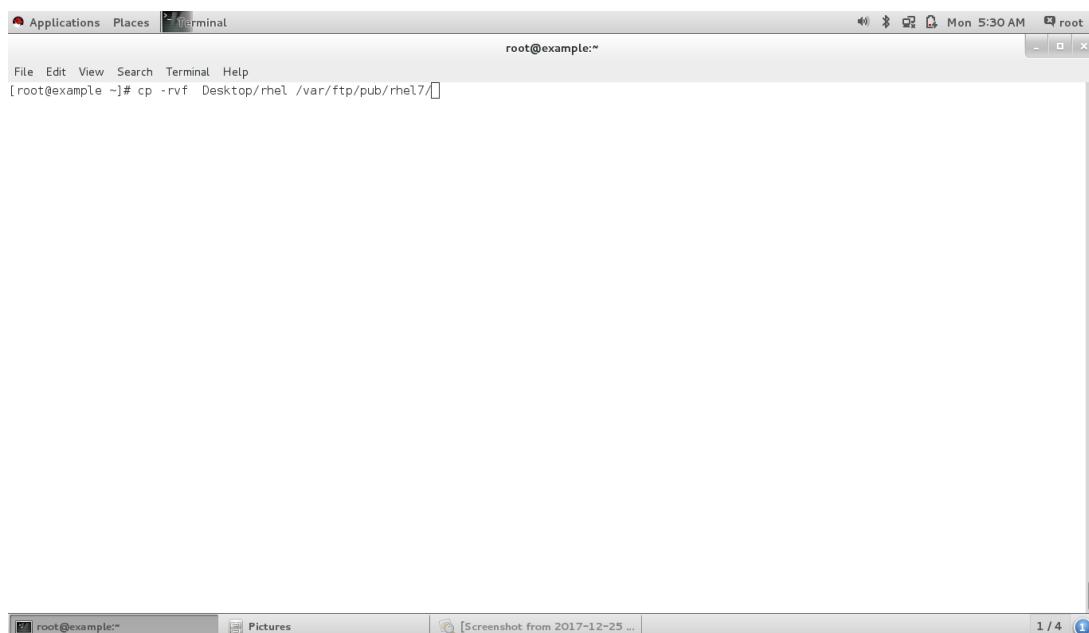


A screenshot of a terminal window titled "Terminal". The window shows a command-line interface with the prompt "root@example:~#". The user has typed the command "cd /var/ftp/pub/rhel7/" and is pressing the Enter key. The window title bar includes the application menu "Applications Places", the window title "Terminal", and system status like "Mon 5:27 AM" and "root". The bottom of the window shows a dock with icons for "root@example:~" (active), "Pictures", and "[Screenshot from 2017-12-25 ...]". A status bar at the bottom right indicates "1 / 4" and has a refresh icon.

Figure 3.4: Folder create

Step 3.2.1.5

- Press (cp – rvf Desktop/rhel /var/ftp/pub/)
- This command used for copy ISO file and transferred the main path of FTP.



A screenshot of a terminal window titled "Terminal". The window shows a command-line interface with the prompt "root@example:~#". The user has typed the command "cp -rvf Desktop/rhel /var/ftp/pub/rhel7/" and is pressing the Enter key. The window title bar includes the application menu "Applications Places", the window title "Terminal", and system status like "Mon 5:30 AM" and "root". The bottom of the window shows a dock with icons for "root@example:~" (active), "Pictures", and "[Screenshot from 2017-12-25 ...]". A status bar at the bottom right indicates "1 / 4" and has a refresh icon.

Figure 3.5: Copy ISO file

Step 3.2.1.6

- This command is used to check the packages are copy correctly?

The screenshot shows a terminal window titled "root@example:/var/ftp/pub/rhel7". The command "ll" is run to list files in the directory. The output shows various files and directories including "addons", "EFL", "EULA", "GPL", "images", "isolinux", "LiveOS", "media.repo", "Packages", "release-notes", "reldatad", "rpms", "TRANS.TBL", and "TRANS.TBL". The timestamp for the last file listed is May 7, 2014.

```
[root@example Desktop]# cd /var/ftp/pub/rhel7/
[root@example ~]# ll
total 356
drwxr-xr-x. 4 root root    52 May  7 2014 addons
drwxr-xr-x. 3 root root   17 May  7 2014 EFL
-rw-r--r--. 1 root root  8266 Apr  4 2014 EULA
-rw-r--r--. 1 root root 18892 Mar  6 2012 GPL
drwxr-xr-x. 3 root root   54 May  7 2014 images
drwxr-xr-x. 2 root root  4096 May  7 2014 isolinux
drwxr-xr-x. 2 root root   41 May  7 2014 LiveOS
-rw-r--r--. 1 root root  108 May  7 2014 media.repo
drwxr-xr-x. 2 root root 225288 May  7 2014 Packages
drwxr-xr-x. 24 root root  4096 May  7 2014 release-notes
drwxr-xr-x. 2 root root  4096 May  7 2014 reldatad
drwxr-xr-x. 8 root root 4096 Dec 25 2014 rpms
-rw-r--r--. 1 root root 3375 Apr  1 2014 RPM-GPG-KEY-redhat-beta
-rw-r--r--. 1 root root 3211 Apr  1 2014 RPM-GPG-KEY-redhat-release
-rw-r--r--. 1 root root 1568 May  7 2014 TRANS.TBL
[root@example rhel7]#
```

Figure 3.6: Checked copy

Step 3.2.1.7

- These commands are used to checked and enable the FTP service.

The screenshot shows a terminal window titled "root@example:~". The user runs "systemctl start vsftpd", "systemctl enable vsftpd", and "systemctl status vsftpd". The output shows the service is active and running since Mon 2017-12-25 05:01:22 EST; 33min ago. The main PID is 1702 and the CGroup is /system.slice/vsftpd.service. The log messages indicate the service is starting and has started successfully.

```
[root@example ~]# systemctl start vsftpd
[root@example ~]# systemctl enable vsftpd
[root@example ~]# systemctl status vsftpd
vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled)
   Active: active (running) since Mon 2017-12-25 05:01:22 EST; 33min ago
     Main PID: 1702 (vsftpd)
    CGroup: /system.slice/vsftpd.service
           └─1702 /usr/sbin/vsftpd/vsftpd.conf

Dec 25 05:01:22 example.com systemd[1]: Starting Vsftpd ftp daemon...
Dec 25 05:01:22 example.com systemd[1]: Started Vsftpd ftp daemon.
Dec 25 05:33:48 example.com systemd[1]: Started Vsftpd ftp daemon.
[root@example ~]#
```

Figure 3.7: Enable the FTP service

Step 3.2.1.8

- Entrance the pc firewall
- Open the FTP port.
- And set the FTP service port as permanently open.

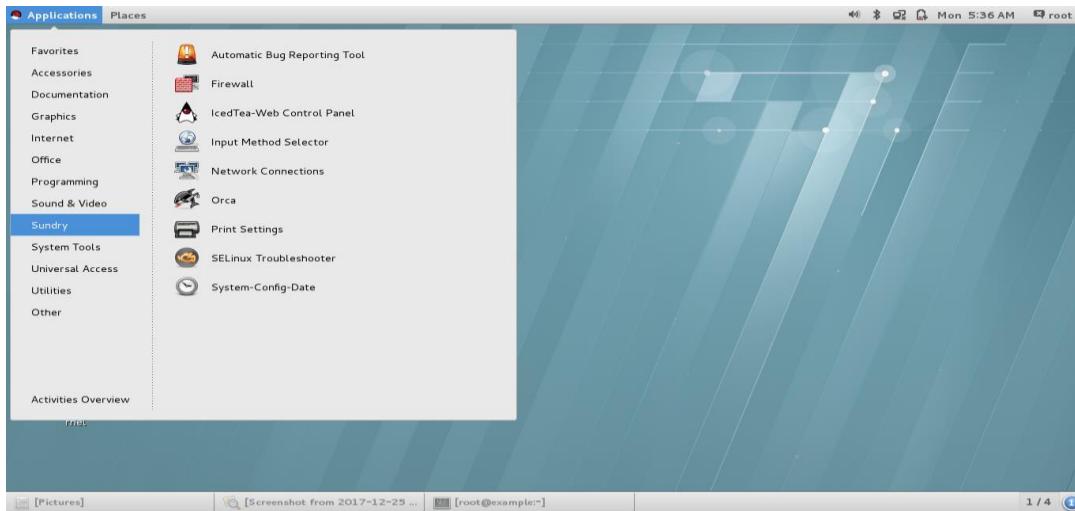


Figure 3.7.1: PC Firewall, FTP portend FTP service port

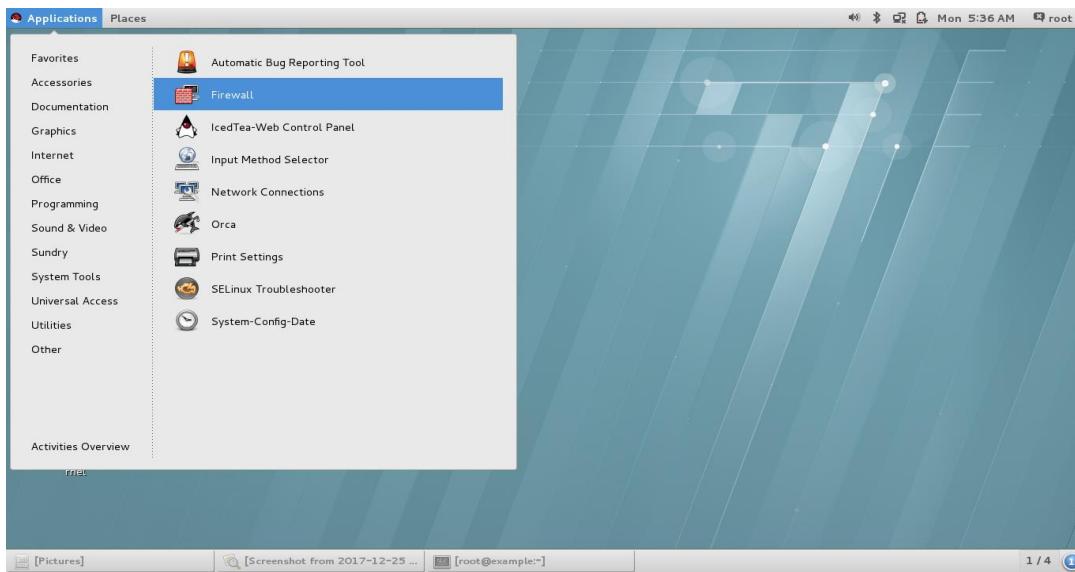


Figure 3.7.2: PC Firewall, FTP portend FTP service port

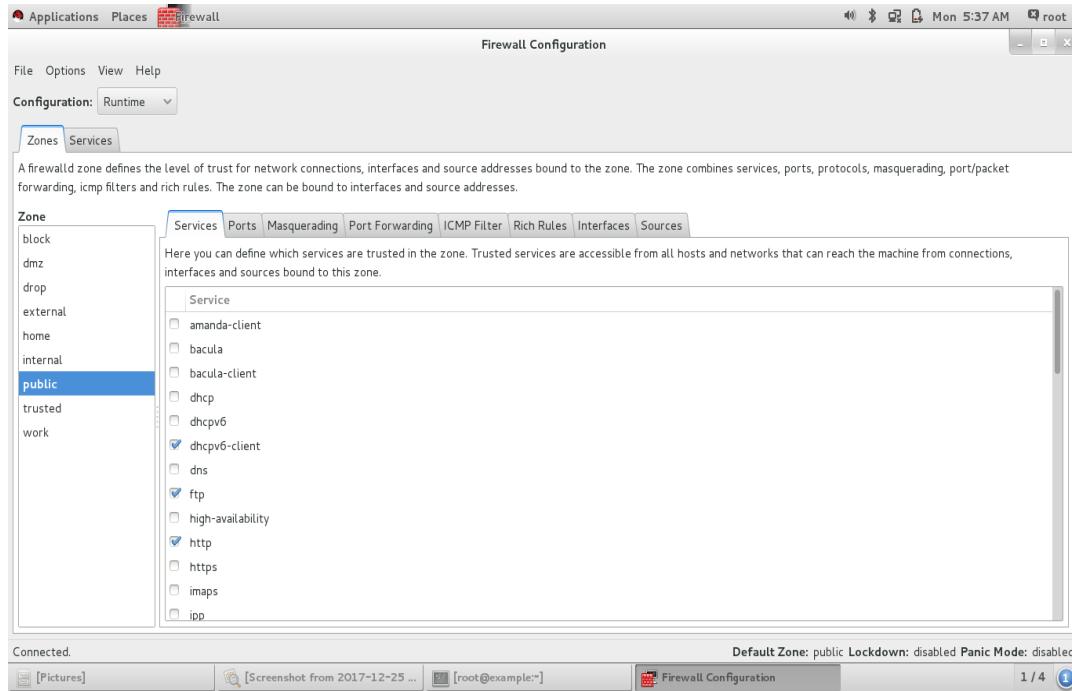


Figure 3.7.3: PC Firewall, FTP portend FTP service port

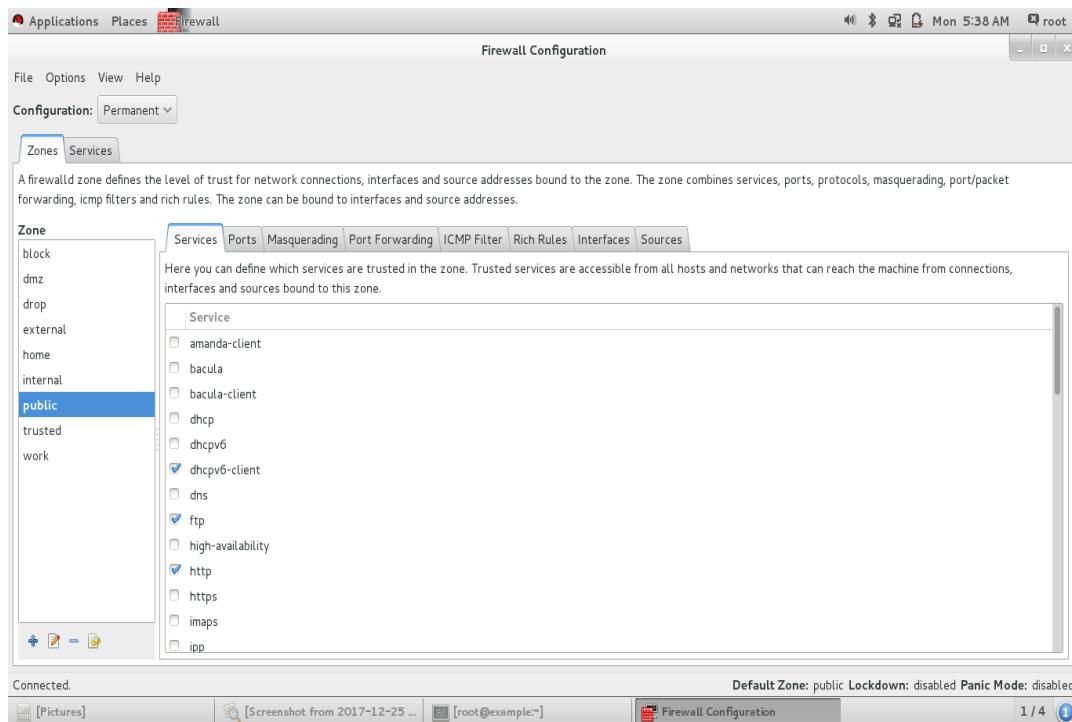


Figure 3.7.4: PC Firewall, FTP portend FTP service port

Step 3.2.1.9

- At first checked the PC IP.
- If do not have any IP, then set an IP address.
- Open the browser
- Press (ftp://192.168.0.2) to open the FTP server.

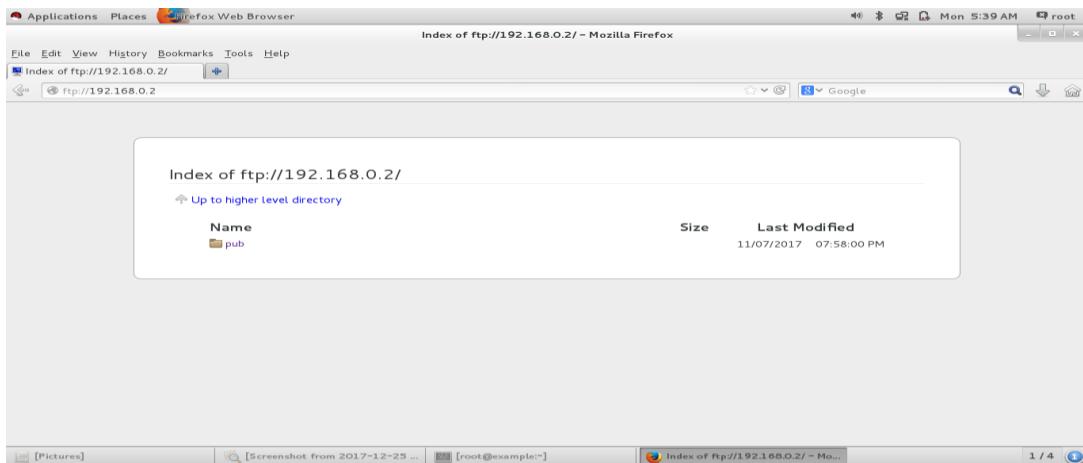


Figure 3.8: Checked the PC IP and set IP address

3.2.2 YUM client configuration

Step 3.2.2.0

- At first go to Terminal.
- Then change directory from desktop to root.

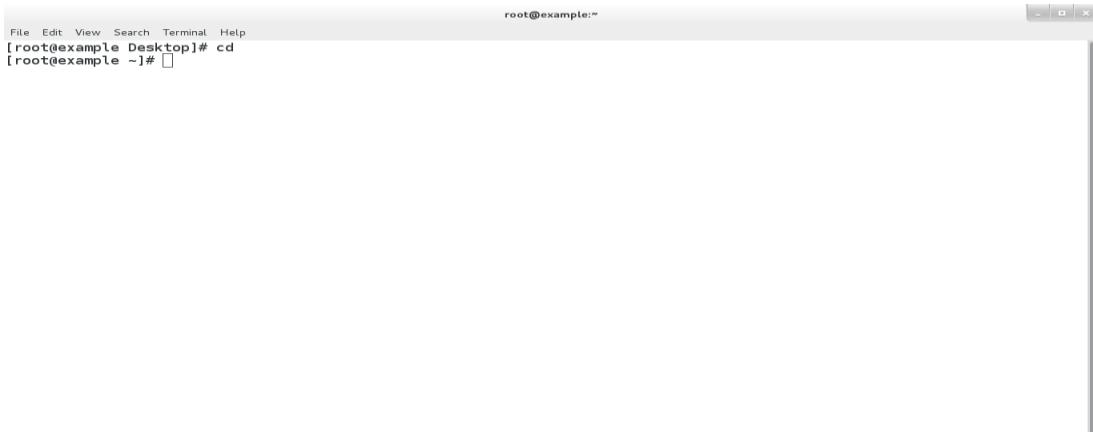


Figure 3.9.1: Terminal and change directory

Step 3.2.2.1

- This is the main path of YUM.



A screenshot of a terminal window titled "root@example:/etc/yum.repos.d". The window has a standard Linux-style title bar with icons for minimize, maximize, and close. The terminal content shows the user navigating to the directory /etc/yum.repos.d:

```
File Edit View Search Terminal Help
[root@example Desktop]# cd
[root@example ~]# cd /etc/yum.repos.d/
[root@example yum.repos.d]#
```

Figure 3.9.2: Main path of YUM

Step 3.2.2.2

- This is the long list, which checked the previous file.



A screenshot of a terminal window titled "root@example:/etc/yum.repos.d". The window has a standard Linux-style title bar with icons for minimize, maximize, and close. The terminal content shows the user running the "ll" command to list files in the directory:

```
File Edit View Search Terminal Help
[root@example yum.repos.d]# ll
total 8
-rw-r--r--. 1 root root 118 Dec 11 01:59 packagekit-media.repo
-rw-r--r--. 1 root root 77 Nov 7 15:37 server.repo
[root@example yum.repos.d]#
```

Figure 3.9.3: Checked the previous file

Step 3.2.2.3

- This command is used for delete all previous file.



A screenshot of a terminal window titled "root@example:/etc/yum.repos.d". The window has a standard title bar with icons for minimize, maximize, and close. The main area shows the command "rm -rf server.repo" being typed by the root user. The terminal is a light gray color with black text.

```
root@example:/etc/yum.repos.d
[root@example yum.repos.d]# rm -rf server.repo
[root@example yum.repos.d]#
```

Figure 3.9.4: Delete all previous file

Step 3.2.2.4

- To create a new file, but when saved then used (.repo) file format after file name.



A screenshot of a terminal window titled "root@example:/etc/yum.repos.d". The window has a standard title bar with icons for minimize, maximize, and close. The main area shows the command "vim server.repo" being typed by the root user. The terminal is a light gray color with black text.

```
root@example:/etc/yum.repos.d
[root@example yum.repos.d]# vim server.repo
```

Figure 3.9.5: Create a new file

Step 3.2.2.5

- This is the new file blank page.



Figure 3.9.6: New file blank page

Step 3.2.2.6

- Now press (i) for insert mode for written.



Figure 3.9.7: Insert mode for written

Step 3.2.2.7

This is configuration of file system

Base= constant value.

Name = any kind of name

Base URL= This is URL, where kept the packages. This is the most important path.

Enableed = to enable to path.

Gpgcheck = to checked the errors.

- Then press esc.
 - Press (:wq!) for save and exit.

```
File Edit View Search Terminal Help root@example:/etc/yum.repos.d [base] name=project baseurl=ftp://192.168.0.2/pub/rhel7 enabled=1 gpgcheck=0 ~ :wq !
```

Figure 3.9.8: Configuration of file system

Step 3.2.2.8

- Press (`yum clean all`) to clean all previous file on baseurl.

```
File Edit View Search Terminal Help root@example:/etc/yum.repos.d
[root@example yum.repos.d]# vim server.repo
[root@example yum.repos.d]# yum clean all
Loaded plugins: langpacks, product-id, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use subscription-manager to register.
Cleaning up everything
[root@example yum.repos.d]# 
```

Figure 3.9.9: Clean previous file

Step 3.2.2.9

- Press (yum repo list) to show all packages.

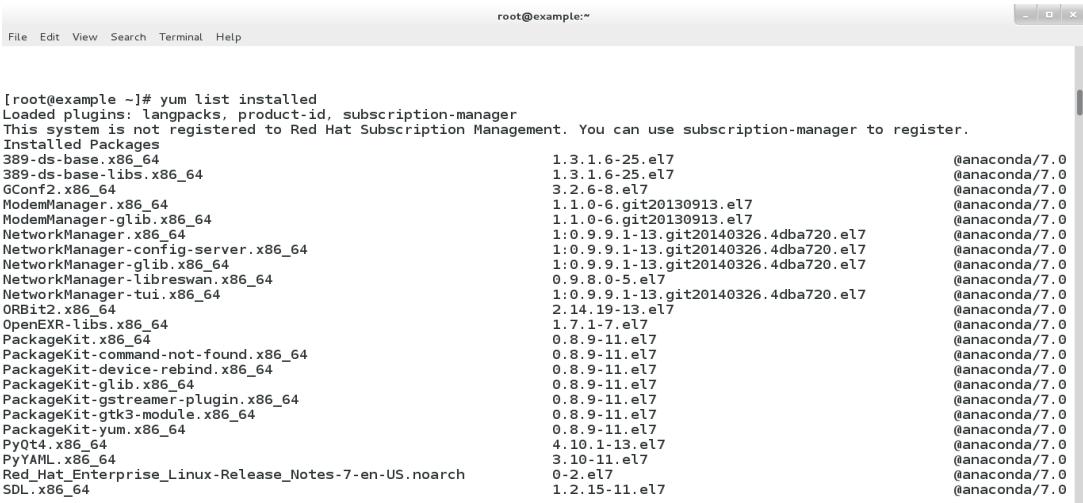


```
root@example:/etc/yum.repos.d
File Edit View Search Terminal Help
[root@example yum.repos.d]# yum repolist
Loaded plugins: langpacks, product-id, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use subscription-manager to register.
repo id                                repo name          status
base                                    project           4,305
repolist: 4,305
[root@example yum.repos.d]#
```

Figure 3.9.10: Showing packages

Step 3.2.2.10

- Press (yum list installed) to show installed packages.



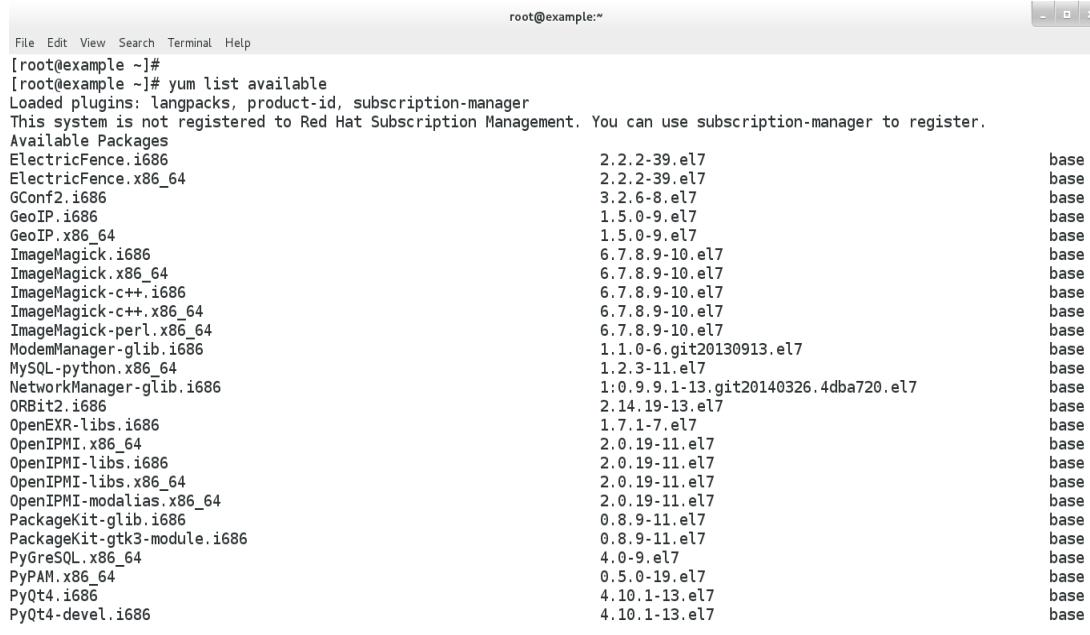
```
root@example:~
File Edit View Search Terminal Help

[root@example ~]# yum list installed
Loaded plugins: langpacks, product-id, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use subscription-manager to register.
Installed Packages
389-ds-base.x86_64                  1.3.1.6-25.el7          @anaconda/7.0
389-ds-base-libs.x86_64               1.3.1.6-25.el7          @anaconda/7.0
GConf2.x86_64                         3.2.6-8.el7             @anaconda/7.0
ModemManager.x86_64                   1.1.0-6.git20130913.el7 @anaconda/7.0
ModemManager-glib.x86_64              1.1.0-6.git20130913.el7 @anaconda/7.0
NetworkManager.x86_64                 1:0.9.9.1-13.git20140326.4dba720.el7 @anaconda/7.0
NetworkManager-config-server.x86_64   1:0.9.9.1-13.git20140326.4dba720.el7 @anaconda/7.0
NetworkManager-glib.x86_64             1:0.9.9.1-13.git20140326.4dba720.el7 @anaconda/7.0
NetworkManager-libreswan.x86_64       0.9.8.0-5.el7            @anaconda/7.0
NetworkManager-tui.x86_64              1:0.9.9.1-13.git20140326.4dba720.el7 @anaconda/7.0
ORBit2.x86_64                         2.14.19-13.el7          @anaconda/7.0
OpenEXR-libs.x86_64                   1.7.1-7.el7              @anaconda/7.0
PackageKit.x86_64                     0.8.9-11.el7             @anaconda/7.0
PackageKit-command-not-found.x86_64   0.8.9-11.el7              @anaconda/7.0
PackageKit-device-rebind.x86_64        0.8.9-11.el7              @anaconda/7.0
PackageKit-glib.x86_64                 0.8.9-11.el7              @anaconda/7.0
PackageKit-gstreamer-plugin.x86_64    0.8.9-11.el7              @anaconda/7.0
PackageKit-gtk3-module.x86_64          0.8.9-11.el7              @anaconda/7.0
PackageKit-yum.x86_64                  0.8.9-11.el7              @anaconda/7.0
PyQt4.x86_64                          4.10.1-13.el7            @anaconda/7.0
PyYAML.x86_64                         3.10-11.el7              @anaconda/7.0
Red_Hat_Enterprise_Linux-Release_Notes-7-en-US.noarch 0-2.el7                @anaconda/7.0
SDL.x86_64                           1.2.15-11.el7            @anaconda/7.0
```

Figure 3.9.11: Install packages

Step 3.2.2.11

- Press (yum list available) to show needed packages, which was not installed yet.

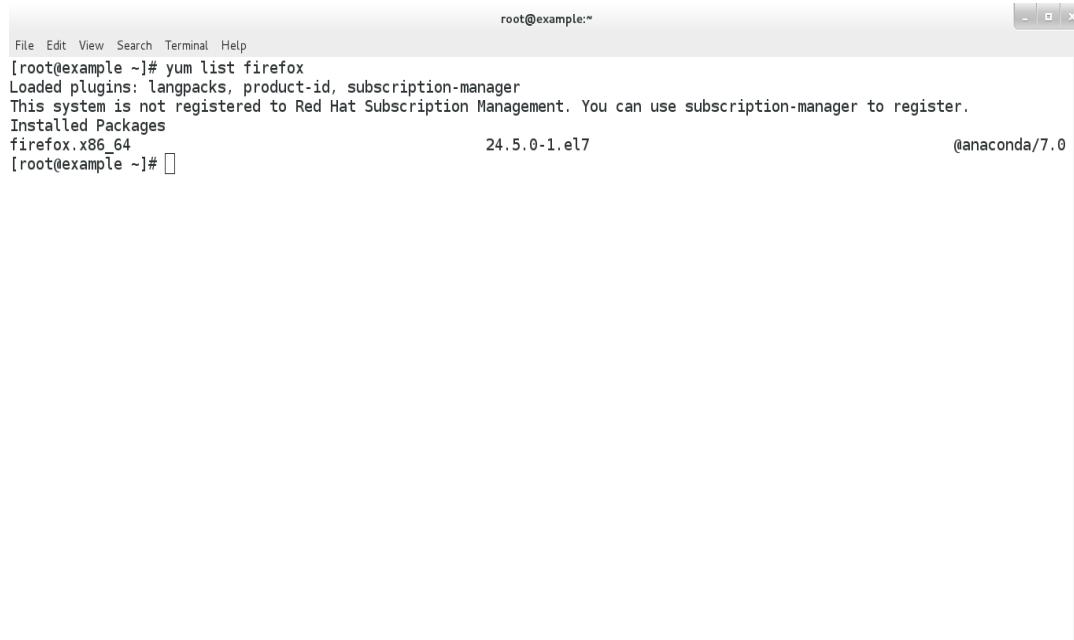


```
root@example ~]# 
[root@example ~]# yum list available
Loaded plugins: langpacks, product-id, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use subscription-manager to register.
Available Packages
ElectricFence.i686                                2.2.2-39.el7          base
ElectricFence.x86_64                               2.2.2-39.el7          base
GConf2.i686                                         3.2.6-8.el7          base
GeoIP.i686                                         1.5.0-9.el7          base
GeoIP.x86_64                                       1.5.0-9.el7          base
ImageMagick.i686                                  6.7.8.9-10.el7       base
ImageMagick.x86_64                               6.7.8.9-10.el7       base
ImageMagick-c++.i686                            6.7.8.9-10.el7       base
ImageMagick-c++.x86_64                           6.7.8.9-10.el7       base
ImageMagick-perl.x86_64                          6.7.8.9-10.el7       base
ModemManager-glib.i686                           1.1.0-6.git20130913.el7
MySQL-python.x86_64                             1.2.3-11.el7          base
NetworkManager-glib.i686                         1:0.9.9.1-13.git20140326.4dba720.el7
ORBit2.i686                                       2.14.19-13.el7       base
OpenEXR-libs.i686                                1.7.1-7.el7          base
OpenIPMI.x86_64                                 2.0.19-11.el7         base
OpenIPMI-libs.i686                              2.0.19-11.el7         base
OpenIPMI-libs.x86_64                           2.0.19-11.el7         base
OpenIPMI-modalias.x86_64                        2.0.19-11.el7         base
PackageKit-glib.i686                           0.8.9-11.el7          base
PackageKit-gtk3-module.i686                     0.8.9-11.el7          base
PyGreSQL.x86_64                                4.0-9.el7             base
PyPAM.x86_64                                    0.5.0-19.el7          base
 PyQt4.i686                                      4.10.1-13.el7         base
 PyQt4-devel.i686                               4.10.1-13.el7         base
```

Figure 3.9.12: Showing packages

Step 3.2.2.12

- Press (yum list Firefox) to show firefox installed yet.

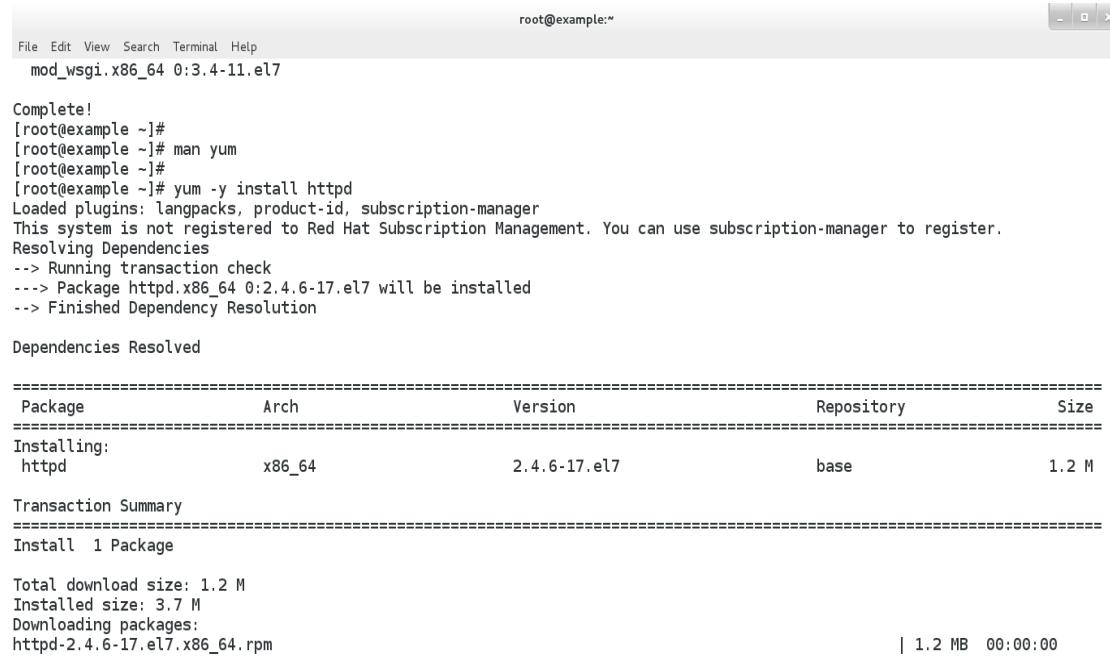


```
root@example ~]# 
[root@example ~]# yum list firefox
Loaded plugins: langpacks, product-id, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use subscription-manager to register.
Installed Packages
firefox.x86_64                                     24.5.0-1.el7          @anaconda/7.0
[root@example ~]# 
```

Figure 3.9.13: Firefox Install

Step 3.2.2.13

- Press (yum -y install httpd) this show how to install httpd packages.

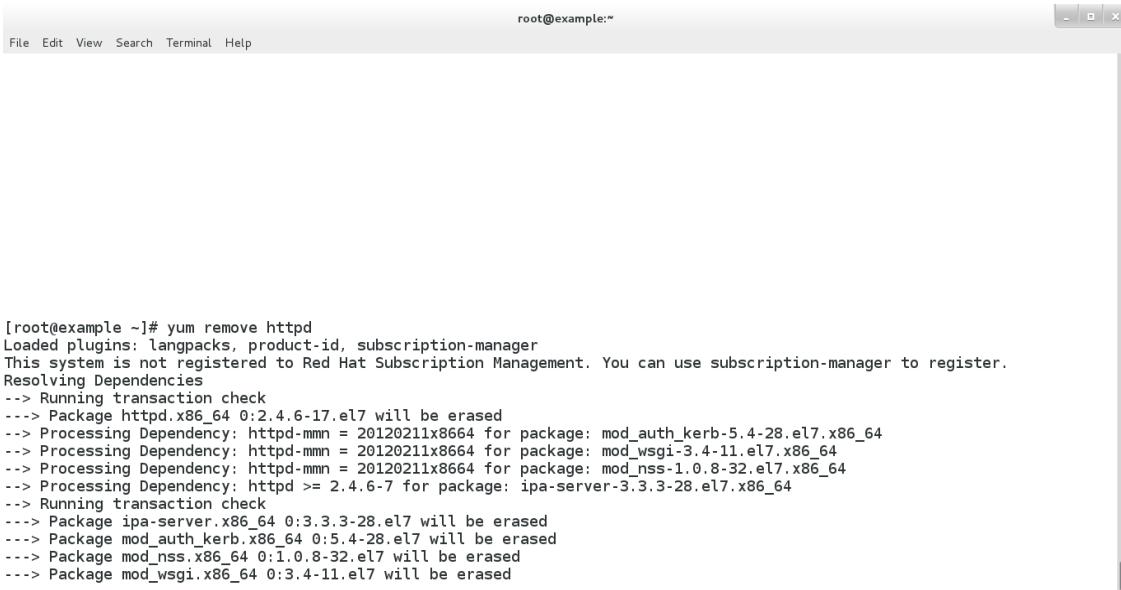


```
File Edit View Search Terminal Help root@example:~ mod_wsgi.x86_64 0:3.4-11.el7 Complete! [root@example ~]# [root@example ~]# man yum [root@example ~]# [root@example ~]# yum -y install httpd Loaded plugins: langpacks, product-id, subscription-manager This system is not registered to Red Hat Subscription Management. You can use subscription-manager to register. Resolving Dependencies --> Running transaction check --> Package httpd.x86_64 0:2.4.6-17.el7 will be installed --> Finished Dependency Resolution Dependencies Resolved ===== Package           Arch         Version          Repository        Size ====== Installing: httpd           x86_64       2.4.6-17.el7      base            1.2 M Transaction Summary ====== Install 1 Package Total download size: 1.2 M Installed size: 3.7 M Downloading packages: httpd-2.4.6-17.el7.x86_64.rpm | 1.2 MB 00:00:00
```

Figure 3.9.14: Install HTTPD packages

Step 3.2.2.14

- Press (yum remove httpd) to remove package.



```
File Edit View Search Terminal Help root@example:~ [root@example ~]# yum remove httpd Loaded plugins: langpacks, product-id, subscription-manager This system is not registered to Red Hat Subscription Management. You can use subscription-manager to register. Resolving Dependencies --> Running transaction check --> Package httpd.x86_64 0:2.4.6-17.el7 will be erased --> Processing Dependency: httpd-mmm = 20120211x8664 for package: mod_auth_kerb-5.4-28.el7.x86_64 --> Processing Dependency: httpd-mmm = 20120211x8664 for package: mod_wsgi-3.4-11.el7.x86_64 --> Processing Dependency: httpd-mmm = 20120211x8664 for package: mod_nss-1.0.8-32.el7.x86_64 --> Processing Dependency: httpd >= 2.4.6-7 for package: ipa-server-3.3.3-28.el7.x86_64 --> Running transaction check --> Package ipa-server.x86_64 0:3.3.3-28.el7 will be erased --> Package mod_auth_kerb.x86_64 0:5.4-28.el7 will be erased --> Package mod_nss.x86_64 0:1.0.8-32.el7 will be erased --> Package mod_wsgi.x86_64 0:3.4-11.el7 will be erased
```

Figure 3.9.15: Remove packages

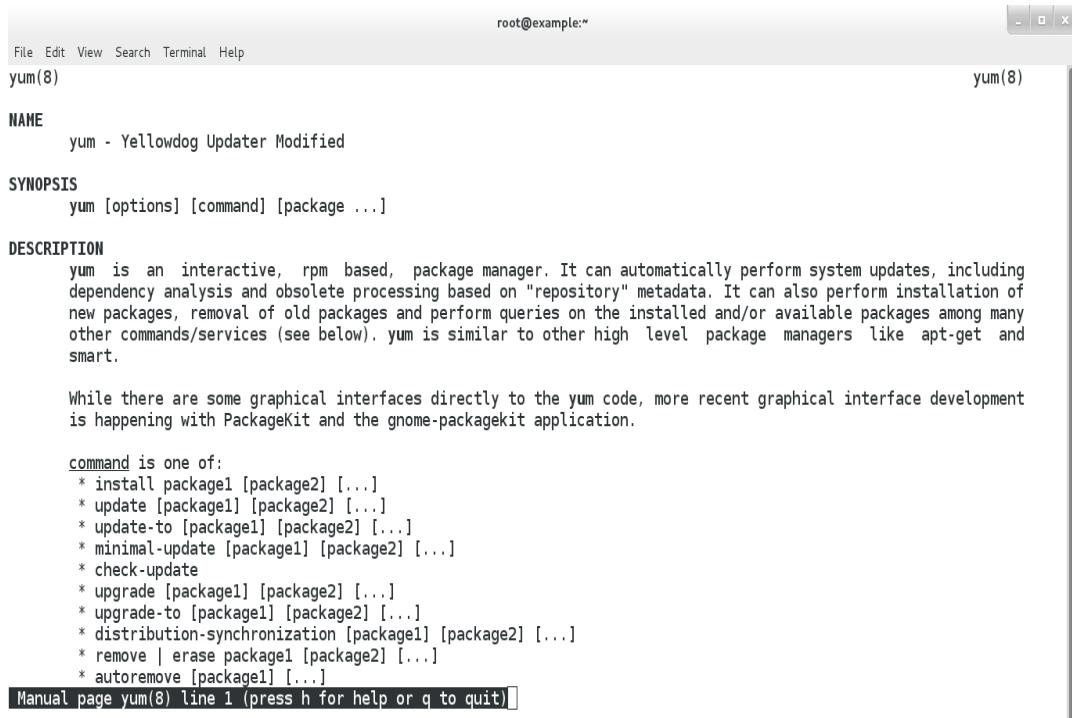
Step 3.2.2.15

- Press (man yum) to show full list of yum.



A screenshot of a terminal window titled "root@example:~". The window has a standard Linux-style title bar with icons for minimize, maximize, and close. The main area of the terminal shows the command "[root@example ~]# man yum" followed by a large amount of text representing the man page for the yum package manager. The text is too long to fit in the image, so it is represented by a large white space area.

Figure 3.9.16: List of yum



A screenshot of a terminal window titled "root@example:~". The window has a standard Linux-style title bar with icons for minimize, maximize, and close. The main area of the terminal shows the command "[root@example ~] man(8) yum(8)". Below this, the man page for yum(8) is displayed. The page includes sections for NAME, SYNOPSIS, and DESCRIPTION. The DESCRIPTION section provides a detailed overview of yum's functionality, mentioning its role as an interactive rpm-based package manager for dependency analysis, updates, and package management. The SYNOPSIS section shows the command syntax: "yum [options] [command] [package ...]". The NAME section identifies the command as "yum - Yellowdog Updater Modified". The bottom of the page shows a prompt: "Manual page yum(8) line 1 (press h for help or q to quit)".

Figure 3.9.17: List of yum

3.3 HTTP server

Step 3.3.1

- Press (cd) for change directory.

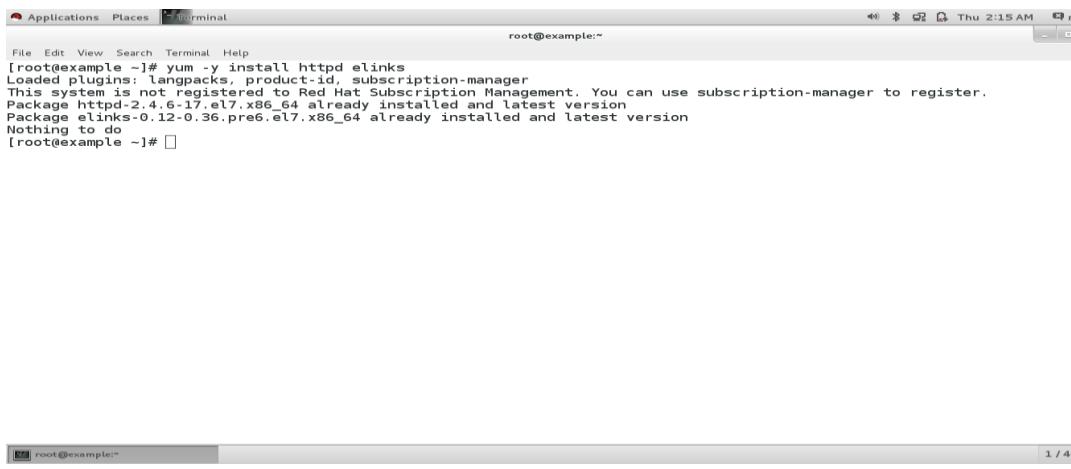


A screenshot of a terminal window titled "Terminal". The window shows a command-line interface with the prompt "root@example:~". The user has typed the command "cd" and is awaiting the system's response. The window has a standard Linux desktop interface with a title bar, menu bar, and scroll bars.

Figure 3.10.1: Change directory

Step: 3.3.2

- Press (yum -y install httpd elinks) for packages installation.
- Httpd for web server packages.
- Elinks for shown output.



A screenshot of a terminal window titled "Terminal". The window shows a command-line interface with the prompt "root@example:~". The user has typed the command "yum -y install httpd elinks" and is awaiting the system's response. The terminal displays the output of the yum command, which includes package details and a confirmation message. The window has a standard Linux desktop interface with a title bar, menu bar, and scroll bars.

Figure 3.10.2: Package installation

Step 3.3.3

- Press (cd /var/www/html/)
- This is the main path of web server; all data and code are stored here.

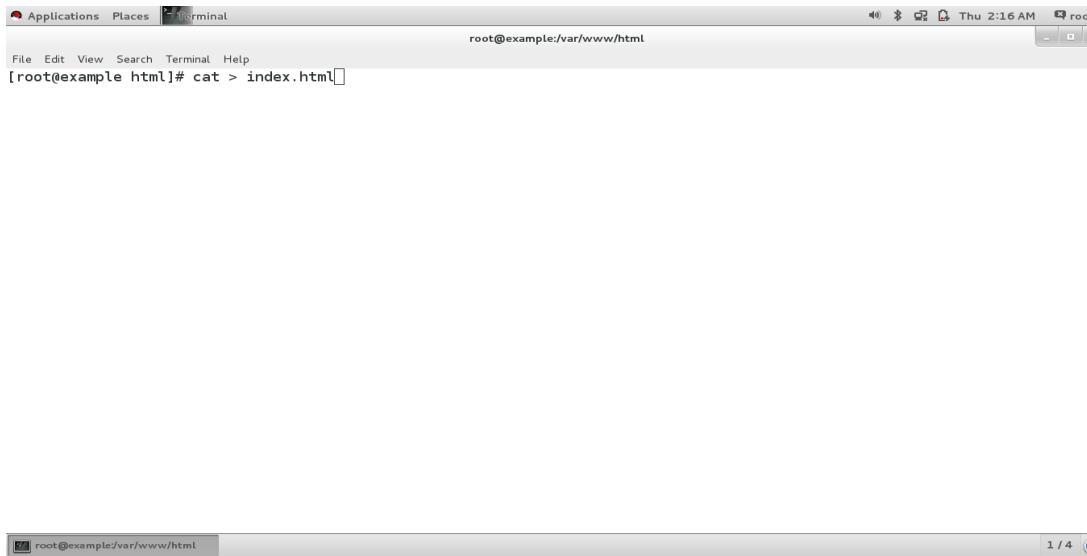


A screenshot of a terminal window titled "Terminal". The window has a title bar with "Applications", "Places", and "Terminal". The status bar at the top right shows "root@example:~" and the date/time "Thu 2:15 AM". The terminal window itself has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The command line shows the root user navigating to the directory "/var/www/html" with the command "[root@example ~]# cd /var/www/html/". The prompt then changes to "[root@example html]#".

Figure 3.10.3: Data and code stored

Step 3.3.4

- Press(cat /index.html)
- All html code are stored here.

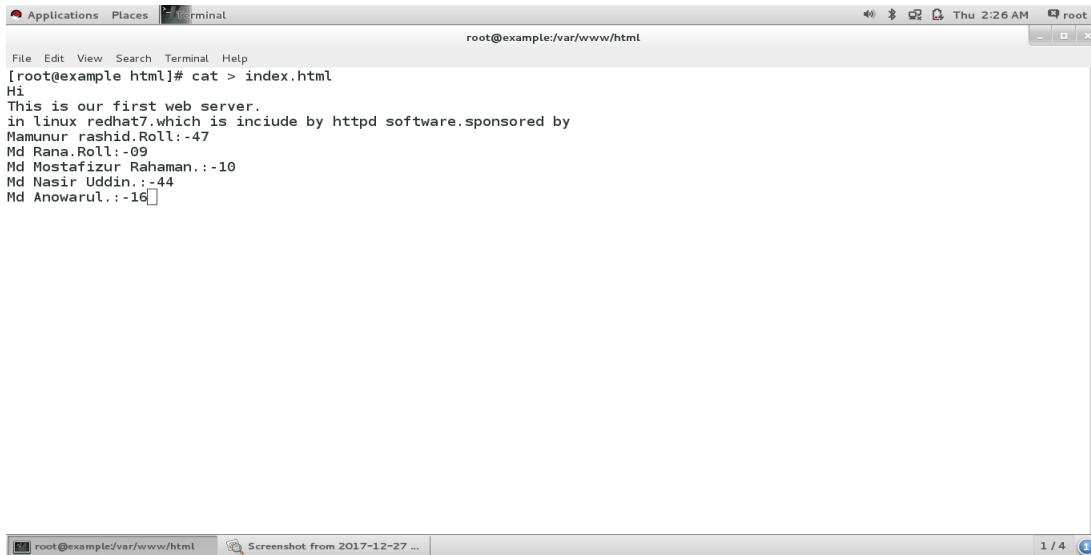


A screenshot of a terminal window titled "Terminal". The window has a title bar with "Applications", "Places", and "Terminal". The status bar at the top right shows "root@example:~" and the date/time "Thu 2:16 AM". The terminal window itself has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The command line shows the root user running the command "[root@example html]# cat > index.html". The prompt then changes to "[root@example html]#".

Figure 3.10.4: HTML code stored

Step 3.3.5

- Here write something that will show in output page.
- Now press (ctrl +d) for saved page.



A screenshot of a terminal window titled "Terminal". The window shows the command "root@example:/var/www/html" at the top. The main area contains the following text:

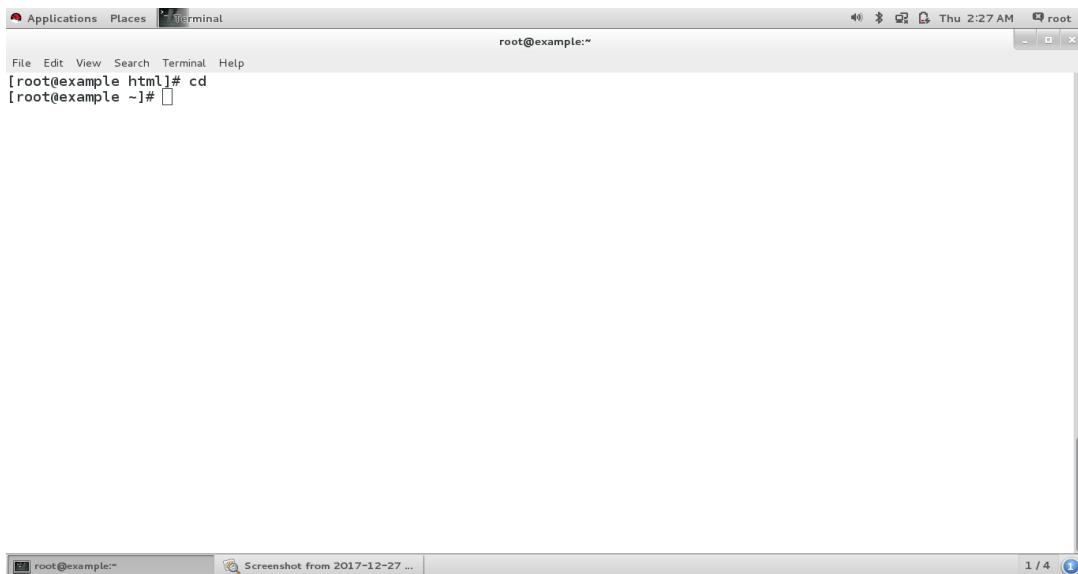
```
File Edit View Search Terminal Help
[root@example html]# cat > index.html
Hi
This is our first web server.
in linux redhat7.which is incide by httpd software.sponsored by
Mamunur rashid.Roll:-47
Md Rana.Roll:-09
Md Mostafizur Rahaman.:-10
Md Nasir Uddin.:-44
Md Anowarul.:-16
```

The terminal window has a standard Linux desktop interface with icons for Applications, Places, and a search bar. The status bar at the bottom shows "Screenshot from 2017-12-27 ...".

Figure 3.10.5: Output and saved page

Step 3.3.6

- Press (cd) for exit html mode.



A screenshot of a terminal window titled "Terminal". The window shows the command "root@example:/var/www/html" at the top. The main area contains the following text:

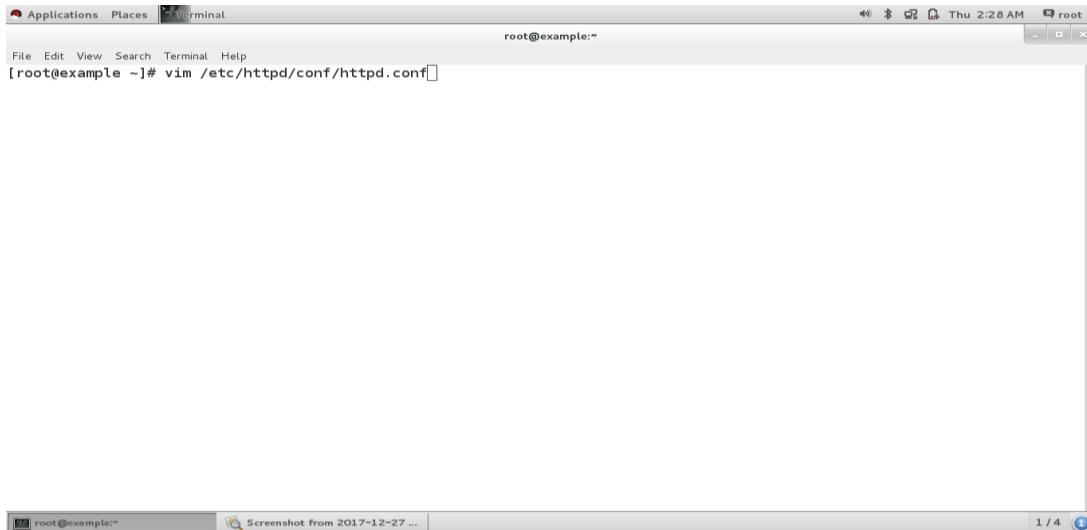
```
File Edit View Search Terminal Help
[root@example html]# cd
[root@example ~]#
```

The terminal window has a standard Linux desktop interface with icons for Applications, Places, and a search bar. The status bar at the bottom shows "Screenshot from 2017-12-27 ...".

Figure 3.10.6: Exit HTML mode

Step 3.3.7

- Press (vim /etc/httpd/conf/httpd.conf) to set domain name, directory index, document root etc.

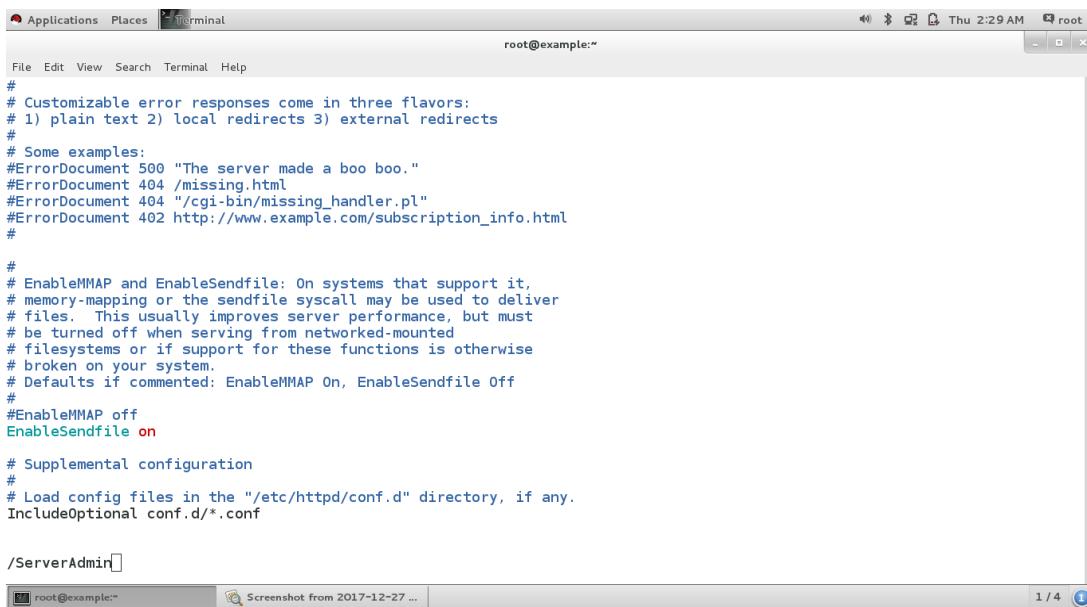


A screenshot of a terminal window titled "root@example:~". The window shows the command `[root@example ~]# vim /etc/httpd/conf/httpd.conf` being typed. The terminal is running as root, indicated by the "root" icon in the title bar and the "root" prefix in the prompt. The window has a standard Linux desktop interface with a menu bar, toolbar, and status bar at the bottom.

Figure 3.10.7: Domain name, directory index, document root

Step 3.3.8

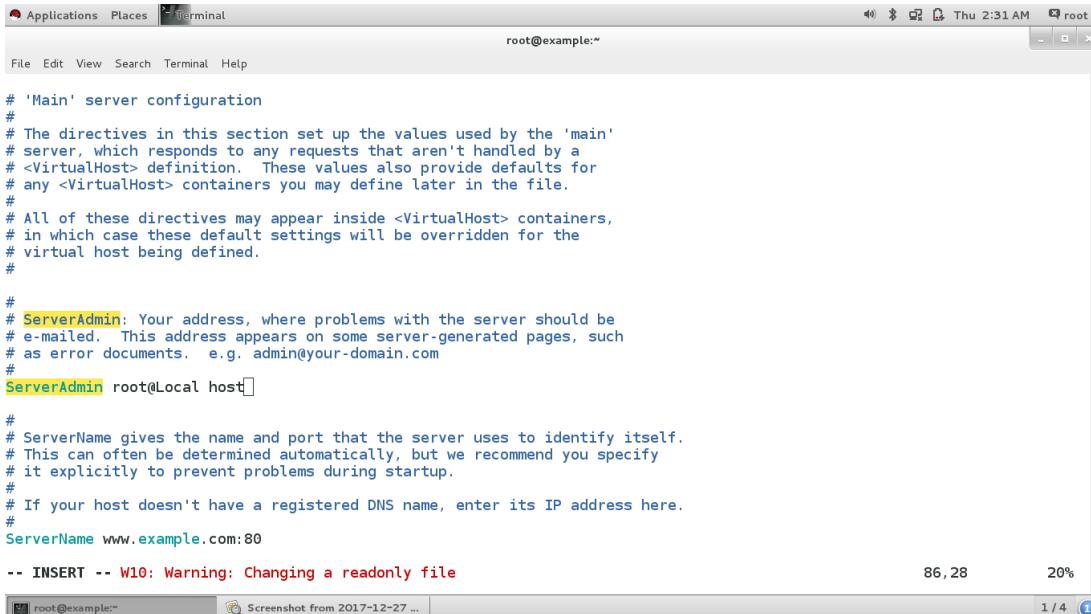
- Press (/serverAdmin) for search.



A screenshot of a terminal window titled "root@example:~". The window shows the command `/ServerAdmin` being typed. The terminal is running as root, indicated by the "root" icon in the title bar and the "root" prefix in the prompt. The window has a standard Linux desktop interface with a menu bar, toolbar, and status bar at the bottom.

Figure 3.10.8: Search

Step 3.3.9



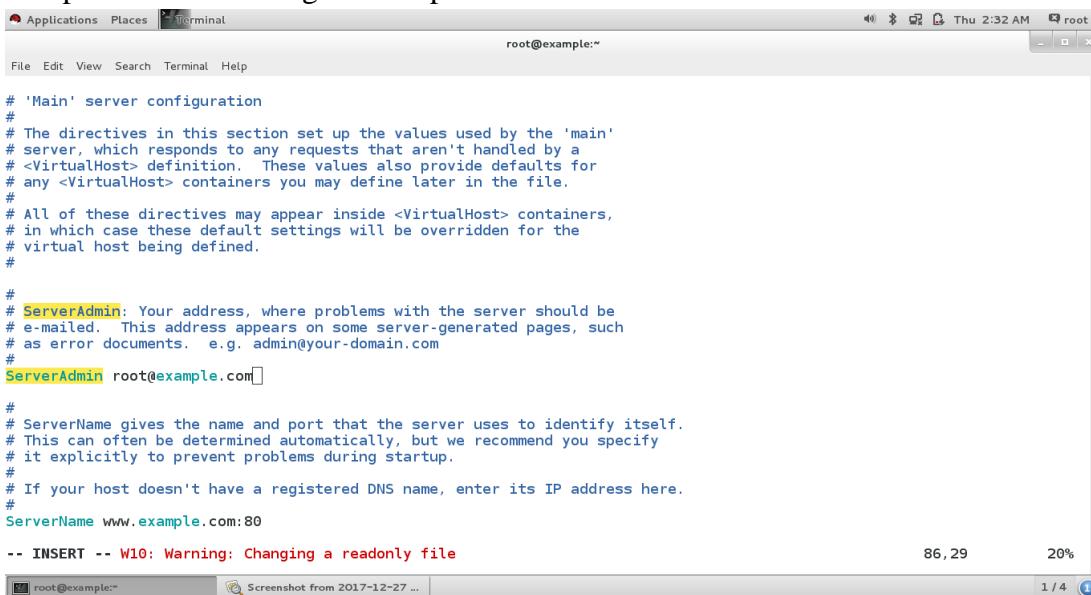
A screenshot of a terminal window titled "Terminal". The window shows a command-line interface with the user "root@example:~". The text in the terminal is a configuration file for Apache, specifically the main server configuration. A search operation is in progress, with the word "ServerAdmin" highlighted in yellow. The terminal also displays a warning message: "-- INSERT -- W10: Warning: Changing a readonly file". The status bar at the bottom right shows "86, 28" and "20%".

```
# 'Main' server configuration
#
# The directives in this section set up the values used by the 'main'
# server, which responds to any requests that aren't handled by a
# <VirtualHost> definition. These values also provide defaults for
# any <VirtualHost> containers you may define later in the file.
#
# All of these directives may appear inside <VirtualHost> containers,
# in which case these default settings will be overridden for the
# virtual host being defined.
#
#
# ServerAdmin: Your address, where problems with the server should be
# e-mailed. This address appears on some server-generated pages, such
# as error documents. e.g. admin@your-domain.com
#
ServerAdmin root@Local host
#
# ServerName gives the name and port that the server uses to identify itself.
# This can often be determined automatically, but we recommend you specify
# it explicitly to prevent problems during startup.
#
# If your host doesn't have a registered DNS name, enter its IP address here.
#
ServerName www.example.com:80
-- INSERT -- W10: Warning: Changing a readonly file
```

Figure 3.10.9: Search

Step 3.3.10

- Now go to insert mode, then replace from root@localhost to root@pc host name.
- Here pc host name acting as example.



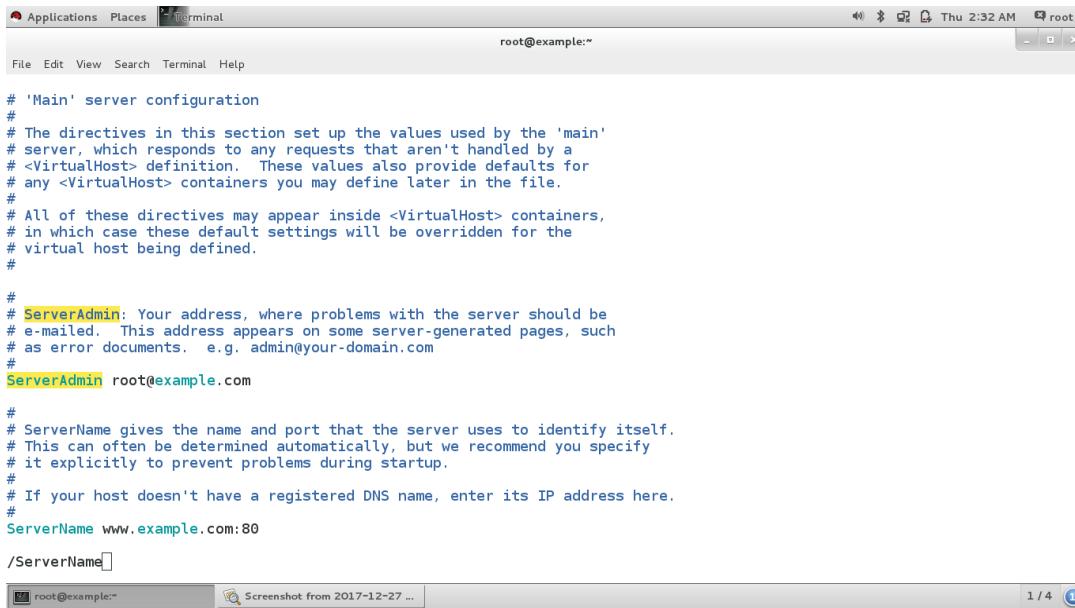
A screenshot of a terminal window titled "Terminal". The window shows a command-line interface with the user "root@example:~". The text in the terminal is the same Apache configuration file as in Figure 3.10.9. In this version, the word "ServerAdmin" has been replaced with "root@example.com", as indicated by the cursor position. The terminal also displays a warning message: "-- INSERT -- W10: Warning: Changing a readonly file". The status bar at the bottom right shows "86, 29" and "20%".

```
# 'Main' server configuration
#
# The directives in this section set up the values used by the 'main'
# server, which responds to any requests that aren't handled by a
# <VirtualHost> definition. These values also provide defaults for
# any <VirtualHost> containers you may define later in the file.
#
# All of these directives may appear inside <VirtualHost> containers,
# in which case these default settings will be overridden for the
# virtual host being defined.
#
#
# ServerAdmin: Your address, where problems with the server should be
# e-mailed. This address appears on some server-generated pages, such
# as error documents. e.g. admin@your-domain.com
#
ServerAdmin root@example.com
#
# ServerName gives the name and port that the server uses to identify itself.
# This can often be determined automatically, but we recommend you specify
# it explicitly to prevent problems during startup.
#
# If your host doesn't have a registered DNS name, enter its IP address here.
#
ServerName www.example.com:80
-- INSERT -- W10: Warning: Changing a readonly file
```

Figure 3.10.10: Insert mode, then replace

Step 3.3.11

- Press (/Server Name) for the server name.

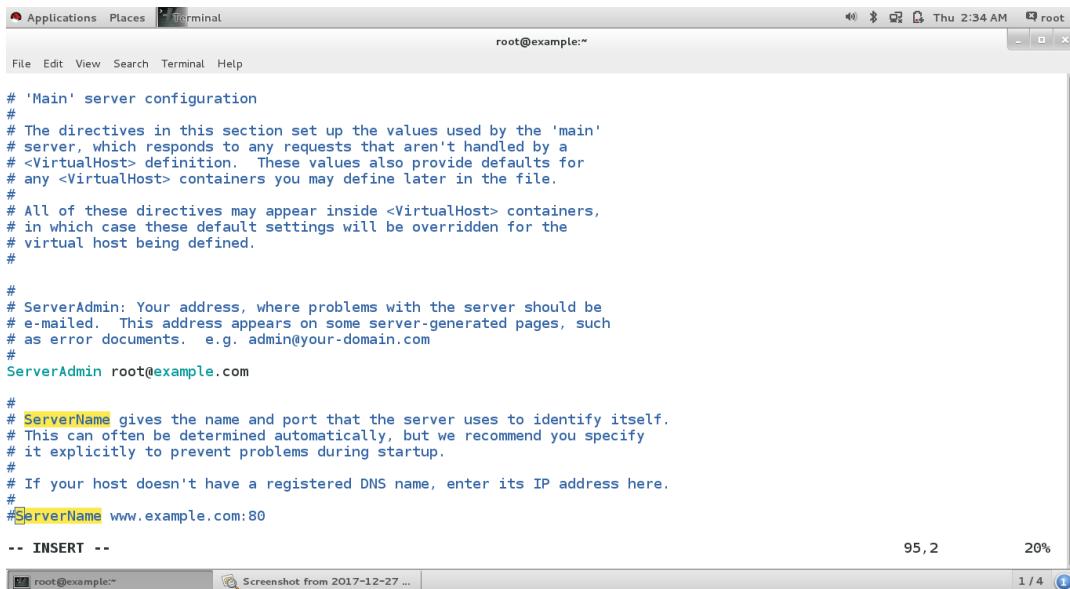


A screenshot of a terminal window titled "Terminal". The window shows the Apache configuration file (httpd.conf) with the "ServerName" directive highlighted in yellow. The terminal prompt is "root@example:~". The status bar at the bottom right shows "Screenshot from 2017-12-27 ...". The window title bar includes "Applications", "Places", "Terminal", and the date/time "Thu 2:32 AM". The status bar also shows "root" and "1 / 4".

```
# 'Main' server configuration
#
# The directives in this section set up the values used by the 'main'
# server, which responds to any requests that aren't handled by a
# <VirtualHost> definition. These values also provide defaults for
# any <VirtualHost> containers you may define later in the file.
#
# All of these directives may appear inside <VirtualHost> containers,
# in which case these default settings will be overridden for the
# virtual host being defined.
#
#
# ServerAdmin: Your address, where problems with the server should be
# e-mailed. This address appears on some server-generated pages, such
# as error documents. e.g. admin@your-domain.com
#
ServerAdmin root@example.com
#
# ServerName gives the name and port that the server uses to identify itself.
# This can often be determined automatically, but we recommend you specify
# it explicitly to prevent problems during startup.
#
# If your host doesn't have a registered DNS name, enter its IP address here.
#
ServerName www.example.com:80
/ServerName
```

Figure 3.10.11: Server name

Step 3.3.12



A screenshot of a terminal window titled "Terminal". The window shows the Apache configuration file (httpd.conf) with the "ServerName" directive highlighted in yellow. The terminal prompt is "root@example:~". The status bar at the bottom right shows "Screenshot from 2017-12-27 ...". The window title bar includes "Applications", "Places", "Terminal", and the date/time "Thu 2:34 AM". The status bar also shows "root" and "1 / 4".

```
# 'Main' server configuration
#
# The directives in this section set up the values used by the 'main'
# server, which responds to any requests that aren't handled by a
# <VirtualHost> definition. These values also provide defaults for
# any <VirtualHost> containers you may define later in the file.
#
# All of these directives may appear inside <VirtualHost> containers,
# in which case these default settings will be overridden for the
# virtual host being defined.
#
#
# ServerAdmin: Your address, where problems with the server should be
# e-mailed. This address appears on some server-generated pages, such
# as error documents. e.g. admin@your-domain.com
#
ServerAdmin root@example.com
#
# ServerName gives the name and port that the server uses to identify itself.
# This can often be determined automatically, but we recommend you specify
# it explicitly to prevent problems during startup.
#
# If your host doesn't have a registered DNS name, enter its IP address here.
#
#ServerName www.example.com:80
-- INSERT --
```

Figure 3.10.12: Server name

- Now set the server name as server1.example.com.

```

Applications Places Terminal
root@example:~
File Edit View Search Terminal Help
# 'Main' server configuration
#
# The directives in this section set up the values used by the 'main'
# server, which responds to any requests that aren't handled by a
# <VirtualHost> definition. These values also provide defaults for
# any <VirtualHost> containers you may define later in the file.
#
# All of these directives may appear inside <VirtualHost> containers,
# in which case these default settings will be overridden for the
# virtual host being defined.
#
# ServerAdmin: Your address, where problems with the server should be
# e-mailed. This address appears on some server-generated pages, such
# as error documents. e.g. admin@your-domain.com
#
ServerAdmin root@example.com
#
# ServerName gives the name and port that the server uses to identify itself.
# This can often be determined automatically, but we recommend you specify
# it explicitly to prevent problems during startup.
#
# If your host doesn't have a registered DNS name, enter its IP address here.
#
ServerName server1.example.com:80

```

95, 18 20%

root@example:~ Screenshot from 2017-12-27 ... 1 / 4

Figure 3.10.13: Set the server name

Step 3.3.13

- Now press Esc from insert mode.
- Then press (Shift + g) go to bellow of page.

```

Applications Places Terminal
root@example:~
File Edit View Search Terminal Help
#
# EnableMMAP and EnableSendfile: On systems that support it,
# memory-mapping or the sendfile syscall may be used to deliver
# files. This usually improves server performance, but must
# be turned off when serving from networked-mounted
# filesystems or if support for these functions is otherwise
# broken on your system.
# Defaults if commented: EnableMMAP On, EnableSendfile Off
#
#EnableMMAP off
EnableSendfile on

# Supplemental configuration
#
# Load config files in the "/etc/nginx/conf.d" directory, if any.
IncludeOptional conf.d/*.conf

-- INSERT --

```

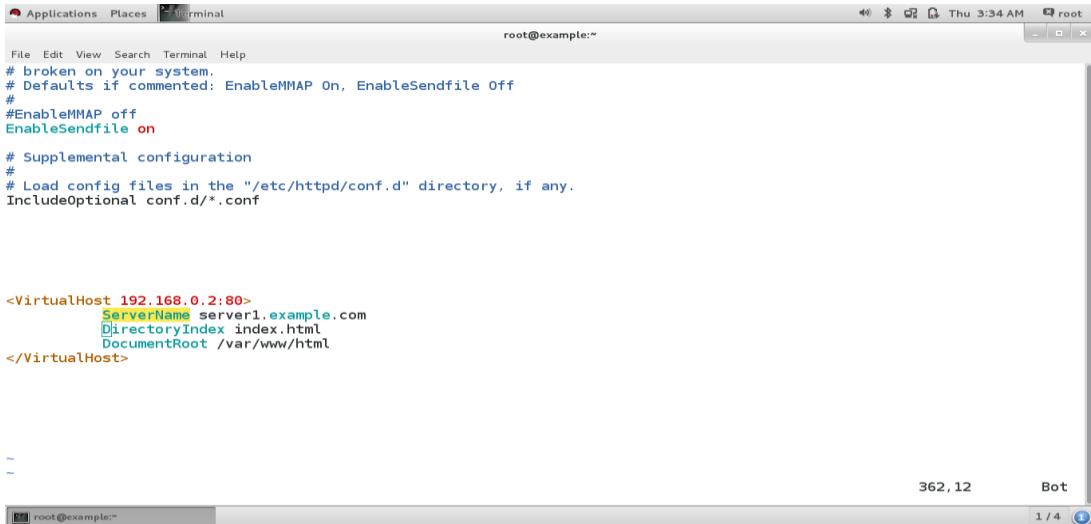
366, 1 Bot

root@example:~ Screenshot from 2017-12-27 ... 1 / 4

Figure 3.10.14: Insert mode

Step 3.3.14

- Now set server name, directory index and document root.



The screenshot shows a terminal window titled "Terminal" running as root on a Linux system. The window title bar includes "root@example:~". The terminal displays the Apache configuration file (httpd.conf) in a text editor. A specific section for a virtual host is highlighted in orange:

```
File Edit View Search Terminal Help
# broken on your system.
# Defaults if commented: EnableMMAP On, EnableSendfile Off
#
#EnableMMAP off
EnableSendfile on

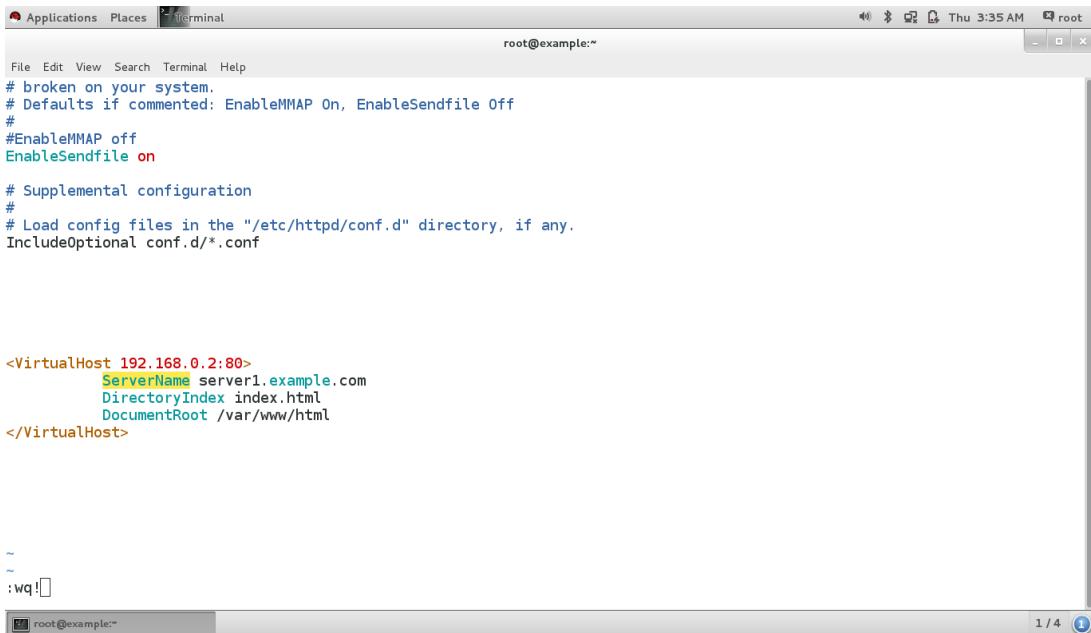
# Supplemental configuration
#
# Load config files in the "/etc/httpd/conf.d" directory, if any.
IncludeOptional conf.d/*.conf

<VirtualHost 192.168.0.2:80>
    ServerName server1.example.com
    DirectoryIndex index.html
    DocumentRoot /var/www/html
</VirtualHost>

~ ~
362,12 Bot
1 / 4
```

Figure 3.10.15: Server name, directory index and document root

- Now press Esc from insert mode.
- And press (:wq!) for save and exit.



The screenshot shows a terminal window titled "Terminal" running as root on a Linux system. The window title bar includes "root@example:~". The terminal displays the Apache configuration file (httpd.conf) in a text editor. A specific section for a virtual host is highlighted in orange:

```
File Edit View Search Terminal Help
# broken on your system.
# Defaults if commented: EnableMMAP On, EnableSendfile Off
#
#EnableMMAP off
EnableSendfile on

# Supplemental configuration
#
# Load config files in the "/etc/httpd/conf.d" directory, if any.
IncludeOptional conf.d/*.conf

<VirtualHost 192.168.0.2:80>
    ServerName server1.example.com
    DirectoryIndex index.html
    DocumentRoot /var/www/html
</VirtualHost>

~ ~
:wq![]
```

Figure 3.10.16: Insert mode, save and exit

Step 3.3.14

- Press (httpd -t) to checked syntax errors.



A screenshot of a terminal window titled "root@example:~". The window shows the command `httpd -t` being run, followed by the output "Syntax OK". The terminal has a standard Gnome-style interface with a menu bar and a status bar at the bottom indicating the date and time.

Figure 3.10.17: Syntax error

Step 3.3.15

- Press (vim /etc/hosts) to set IP and server name in hosts.



A screenshot of a terminal window titled "root@example:~". The window shows the command `vim /etc/hosts` being run. The input field of the terminal shows the command line. The terminal has a standard Gnome-style interface with a menu bar and a status bar at the bottom indicating the date and time.

Figure 3.10.18: IP and server name in hosts

A screenshot of a terminal window titled "Terminal". The window shows the contents of the "/etc/hosts" file. The file contains the following entries:

```
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
```

The terminal window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The status bar at the bottom shows "root@example:" and the date and time "Thu 2:42 AM". A scroll bar is visible on the right side of the terminal window.

Figure 3.10.19: IP and server name in hosts

A screenshot of a terminal window titled "Terminal". The window shows the contents of the "/etc/hosts" file. The file contains the following entries:

```
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.0.2 server1.example.com
```

The terminal window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The status bar at the bottom shows "root@example:" and the date and time "Thu 3:37 AM". A scroll bar is visible on the right side of the terminal window.

Figure 3.10.20: IP and server name in hosts

A screenshot of a terminal window titled "Terminal". The window shows the command-line interface of a Linux system. The user is root, as indicated by the "root@example:" prompt at the top. The terminal displays the contents of the "/etc/hosts" file. The file contains several entries: "127.0.0.1 localhost localdomain localhost4 localhost4.localdomain4", "::1 localhost localdomain localhost6 localhost6.localdomain6", and "192.168.0.2 server1.example.com". Below these entries, there are approximately 20 blank lines followed by a command ":wq!". The bottom status bar shows the path "root@example:" and the file name "Pictures".

Figure 3.10.21: IP and server name in hosts

Step 3.3.16

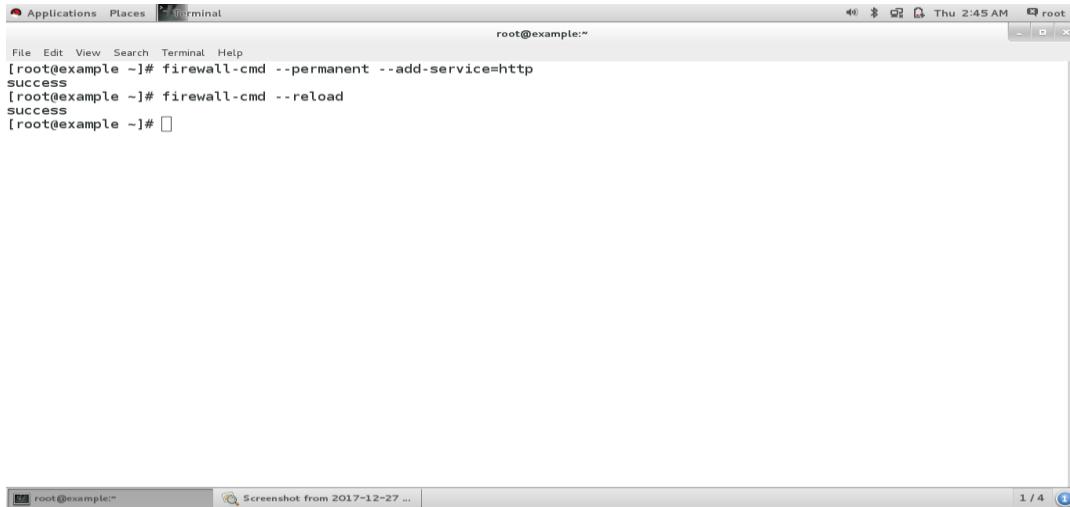
- Now press web server enable and restart.

A screenshot of a terminal window titled "Terminal". The user is root, as indicated by the "root@example:" prompt. The terminal shows the execution of two commands: "systemctl enable httpd.service" and "systemctl restart httpd.service". Both commands are completed successfully, as indicated by the "[root@example ~]#" prompt. The bottom status bar shows the path "root@example:" and the file name "Screenshot from 2017-12-27 ...".

Figure 3.10.22: Web server enable and restart

Step 3.3.17

- Now add firewall and reload.



A screenshot of a terminal window titled "root@example:~". The window shows the following command history:

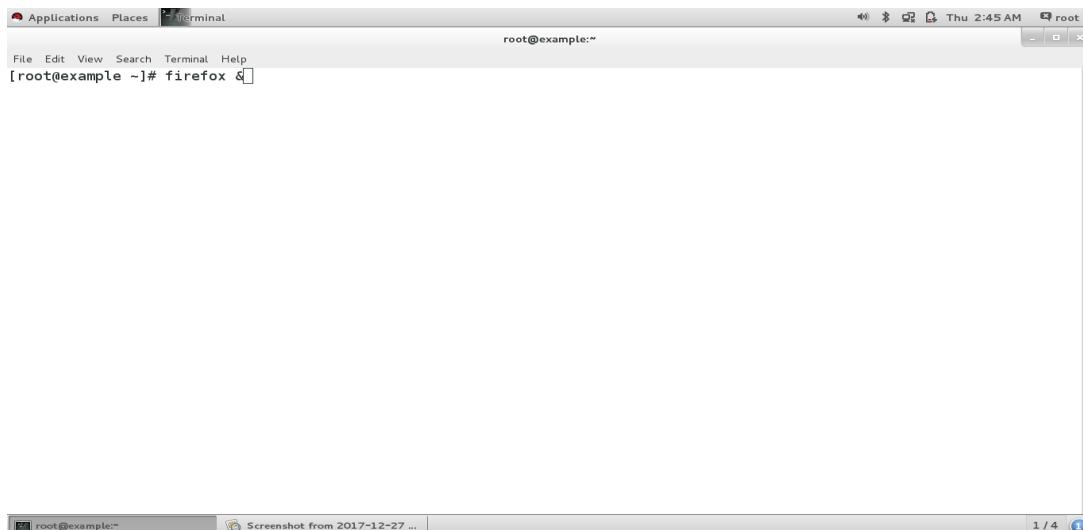
```
root@example:~# firewall-cmd --permanent --add-service=http
success
root@example:~# firewall-cmd --reload
success
root@example:~#
```

The terminal window has a standard Linux desktop interface with a title bar, menu bar, and status bar at the bottom. The status bar shows "root" and the date/time "Thu 2:45 AM".

Figure 3.10.23: Add firewall and reload

Step 3.3.18

- Press (Firefox&) to hit browser with command.



A screenshot of a terminal window titled "root@example:~". The window shows the following command:

```
root@example:~# firefox &
```

The terminal window has a standard Linux desktop interface with a title bar, menu bar, and status bar at the bottom. The status bar shows "root" and the date/time "Thu 2:45 AM".

Figure 3.10.24: Hit browser with command

Step 3.3.19

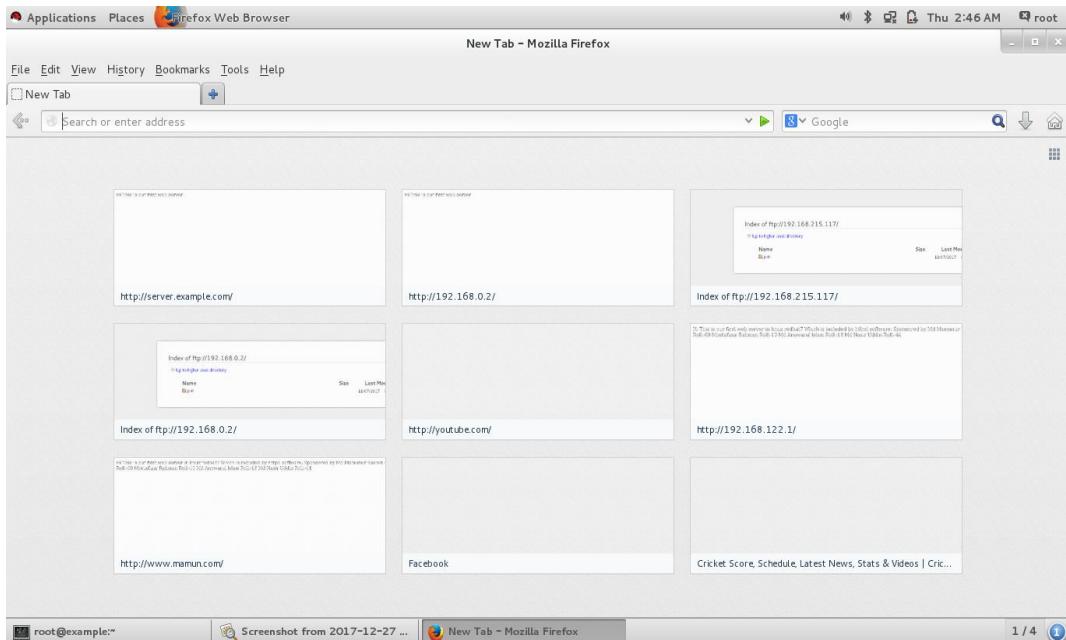


Figure 3.10.25: Open browser

➤ Now browser hit with IP address.

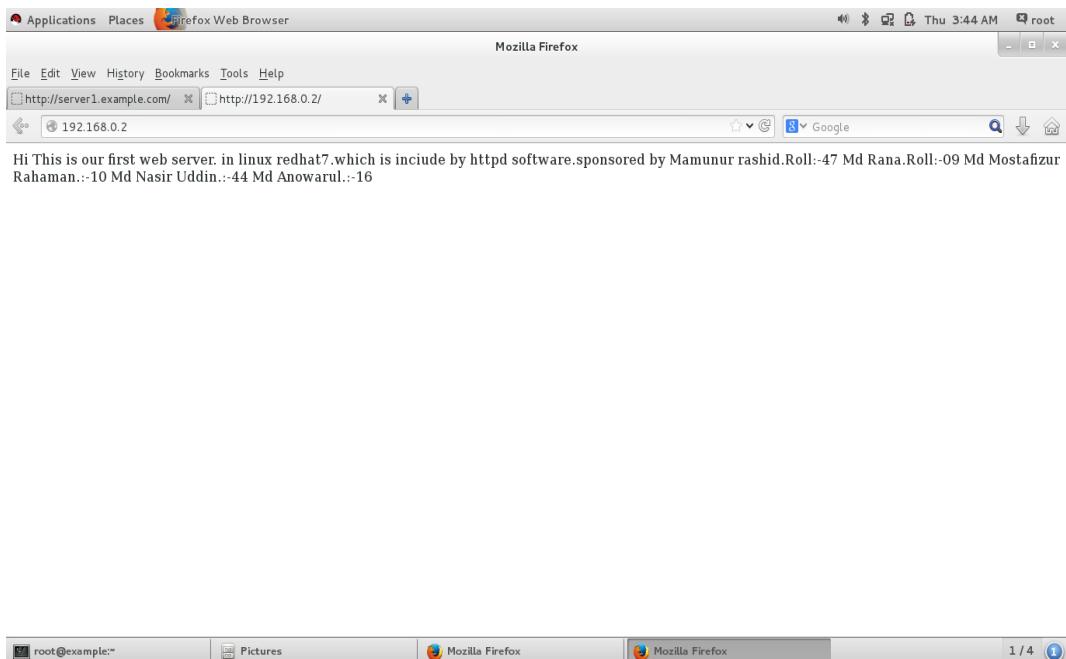


Figure 3.10.26: Browser hit with IP

- Now browser hit with server name.

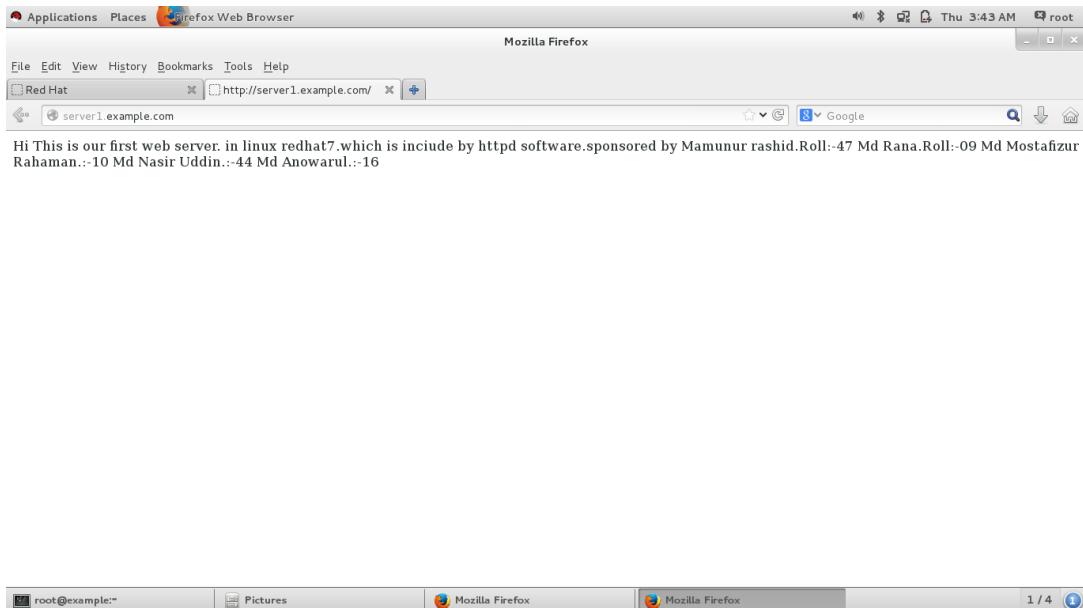


Figure 3.10.27: Browser hit

3.4 Virtual web server

Step 3.4.0

- Press (mkdir /var/www/hello) to create a sub folder.

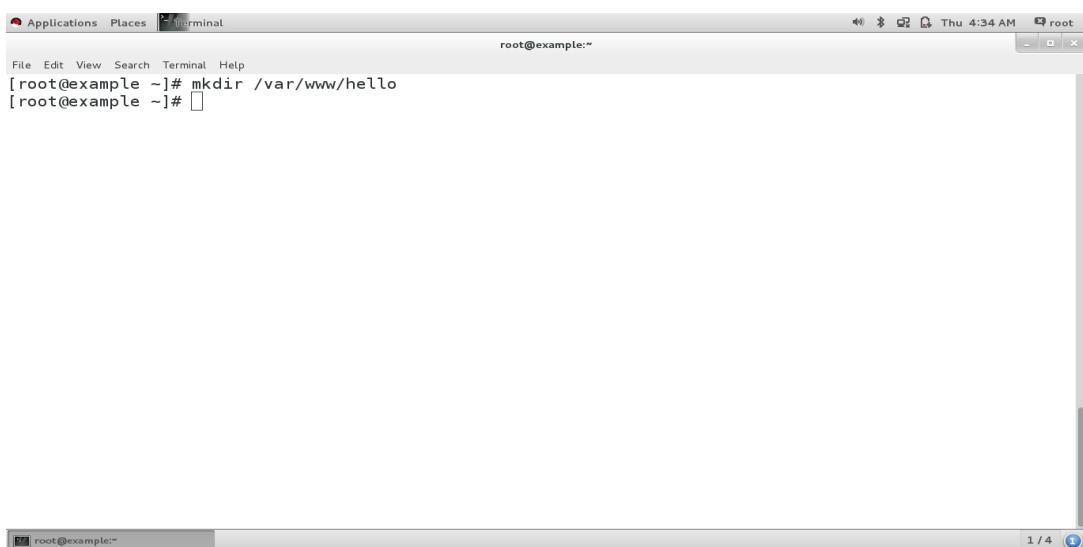
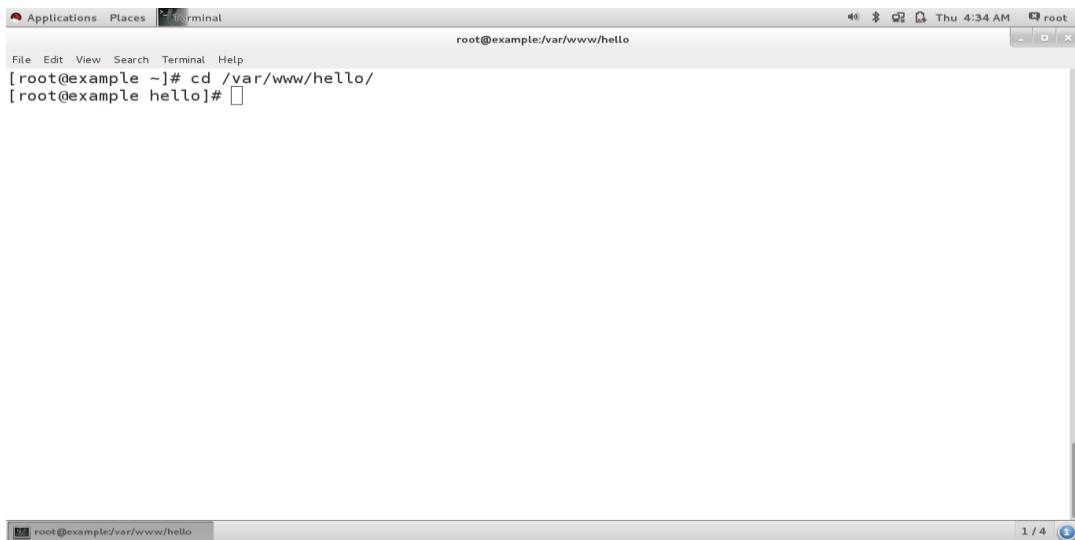


Figure 3.11.1: Create sub folder

Step 3.4.1

- Press (cd /var/www/hello) to enter the sub folder.

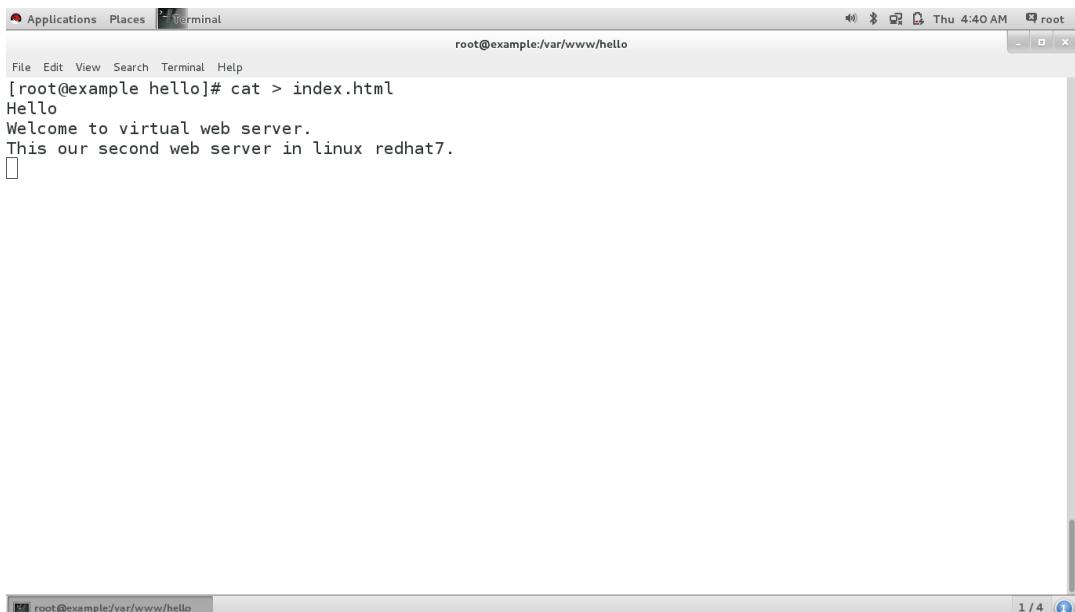


A screenshot of a terminal window titled "Terminal". The window shows a command-line interface with the following text:
root@example ~]# cd /var/www/hello
[root@example hello]#
The window has a standard Linux desktop interface with a title bar, menu bar, and status bar at the top. The status bar indicates "Thu 4:34 AM" and "root". The terminal area is scrollable, with a scrollbar on the right side.

Figure 3.11.2: Sub folder

Step 3.4.2

- Press (cat > index.html) to create a file.

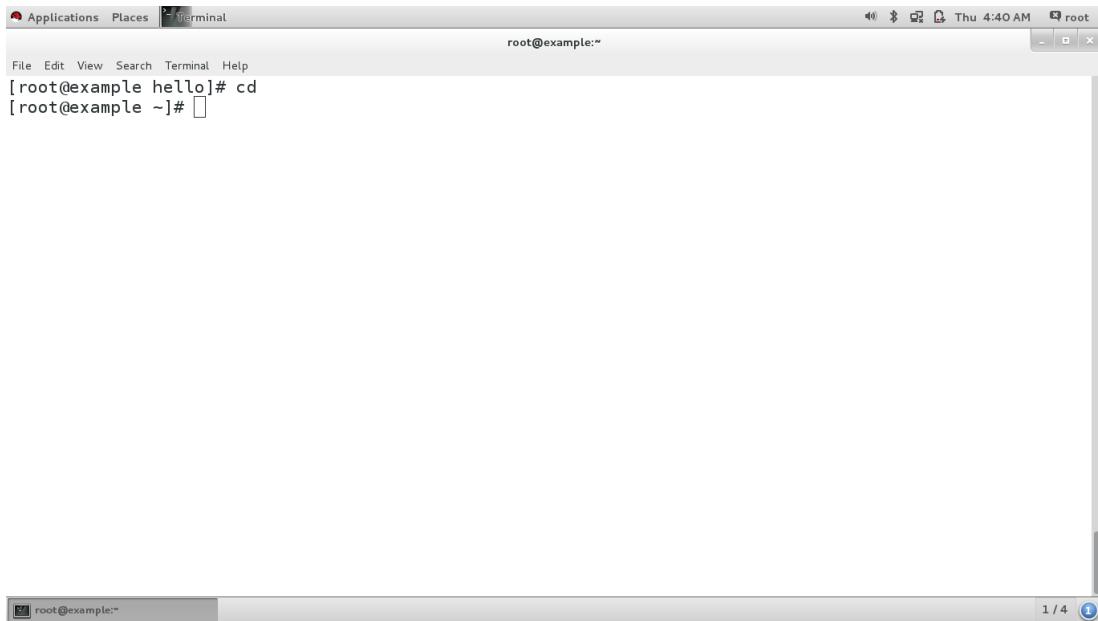


A screenshot of a terminal window titled "Terminal". The window shows a command-line interface with the following text:
root@example hello]# cat > index.html
Hello
Welcome to virtual web server.
This our second web server in linux redhat7.
The window has a standard Linux desktop interface with a title bar, menu bar, and status bar at the top. The status bar indicates "Thu 4:40 AM" and "root". The terminal area is scrollable, with a scrollbar on the right side.

Figure 3.11.3: Create a file

Step 3.4.3

- Press (cd) to exit from sub folder.



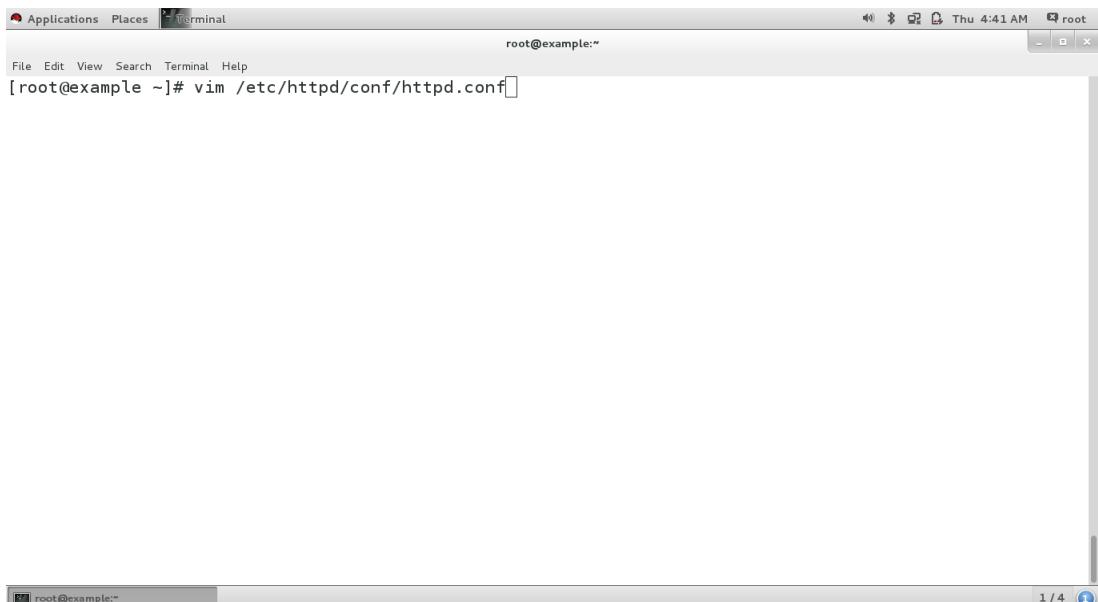
A screenshot of a terminal window titled "Terminal". The window has a title bar with "Applications", "Places", and "Terminal". The status bar at the top right shows "root@example:~" and the date/time "Thu 4:40 AM". The main area of the terminal shows the command history:

```
root@example:~# cd  
[root@example ~]#
```

The terminal window is part of a larger desktop environment with a toolbar at the bottom.

Figure 3.11.4: Exit of sub folder

Step 3.4.43



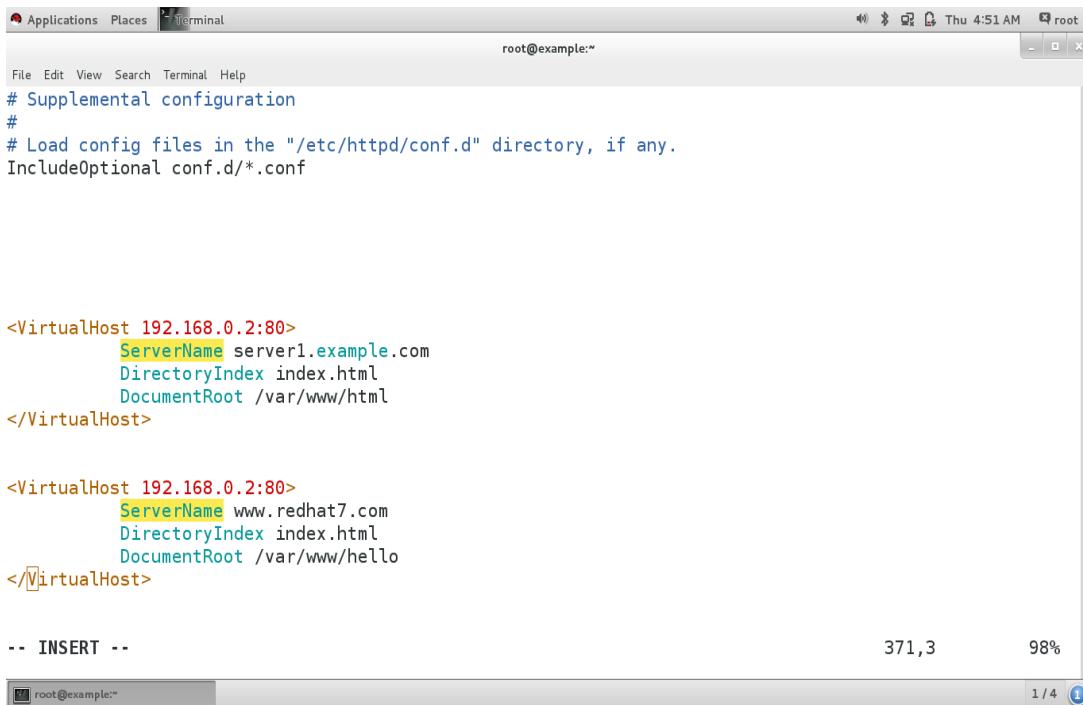
A screenshot of a terminal window titled "Terminal". The window has a title bar with "Applications", "Places", and "Terminal". The status bar at the top right shows "root@example:~" and the date/time "Thu 4:41 AM". The main area of the terminal shows the command:

```
[root@example ~]# vim /etc/httpd/conf/httpd.conf
```

The terminal window is part of a larger desktop environment with a toolbar at the bottom.

Figure 3.11.5: Exit of sub folder

- To set server name, directory index and document root.



```

root@example:~#
# Supplemental configuration
#
# Load config files in the "/etc/httpd/conf.d" directory, if any.
IncludeOptional conf.d/*.conf

<VirtualHost 192.168.0.2:80>
    ServerName server1.example.com
    DirectoryIndex index.html
    DocumentRoot /var/www/html
</VirtualHost>

<VirtualHost 192.168.0.2:80>
    ServerName www.redhat7.com
    DirectoryIndex index.html
    DocumentRoot /var/www/hello
</VirtualHost>

-- INSERT --
371,3      98%
root@example:~#
1 / 4

```

Figure 3.11.6: Server name, directory index and document root

- Now press (:wq!) to save and exit.



```

root@example:~#
# Supplemental configuration
#
# Load config files in the "/etc/httpd/conf.d" directory, if any.
IncludeOptional conf.d/*.conf

<VirtualHost 192.168.0.2:80>
    ServerName server1.example.com
    DirectoryIndex index.html
    DocumentRoot /var/www/html
</VirtualHost>

<VirtualHost 192.168.0.2:80>
    ServerName www.redhat7.com
    DirectoryIndex index.html
    DocumentRoot /var/www/hello
</VirtualHost>

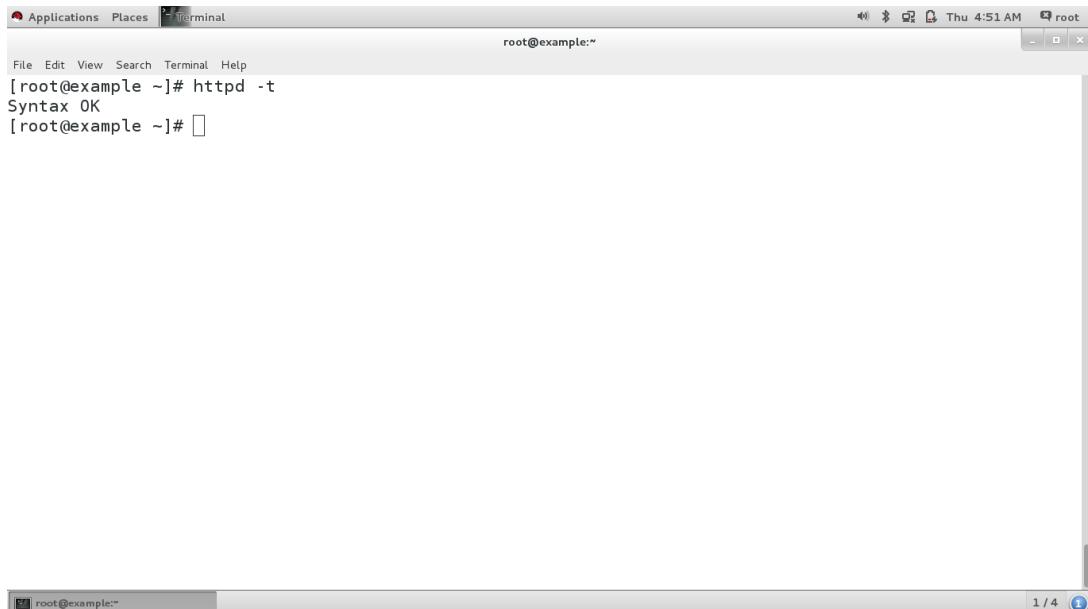
:wq!
root@example:~#
1 / 4

```

Figure 3.11.7: Save and exit

Step 3.4.5

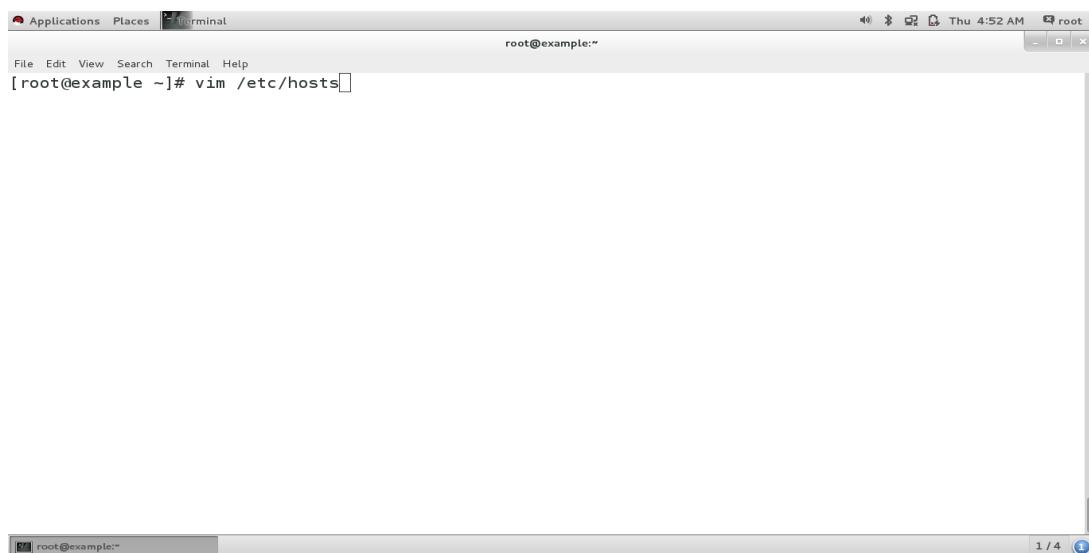
- Press (httpd -t) to checked syntax error.



A screenshot of a terminal window titled "terminal". The window shows the command [root@example ~]# httpd -t being run by root. The output indicates "Syntax OK". The terminal has a standard Gnome-like interface with a title bar, menu bar, and status bar.

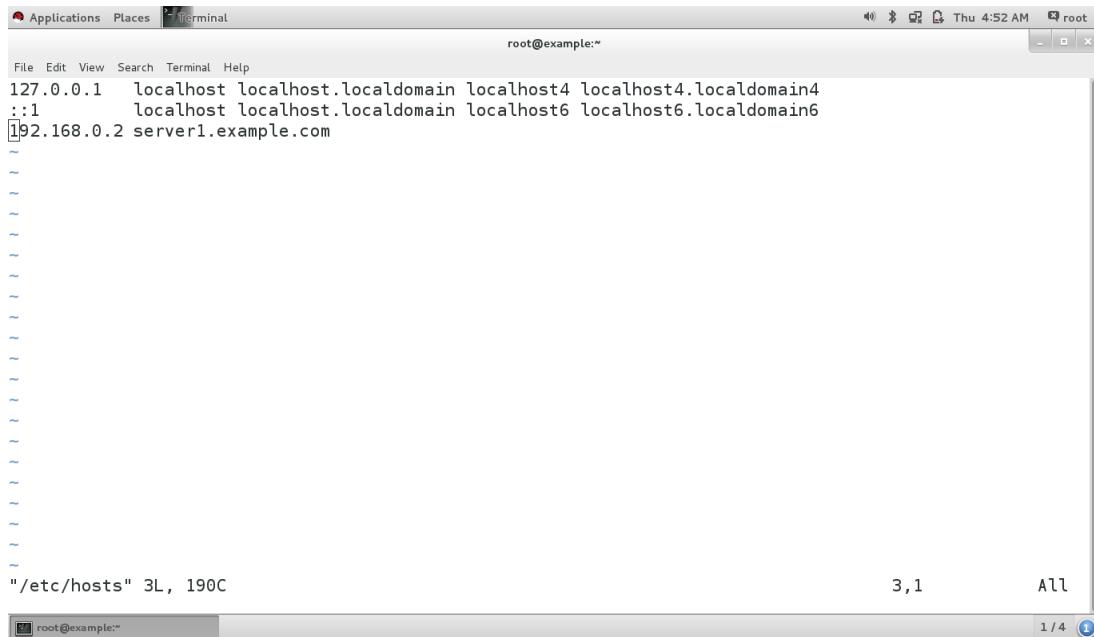
Figure 3.11.8: Syntax error

Step 3.4.6



A screenshot of a terminal window titled "terminal". The window shows the command [root@example ~]# vim /etc/hosts being run by root. The terminal is currently empty, awaiting input. The interface is identical to Figure 3.11.8.

Figure 3.11.9: Syntax error



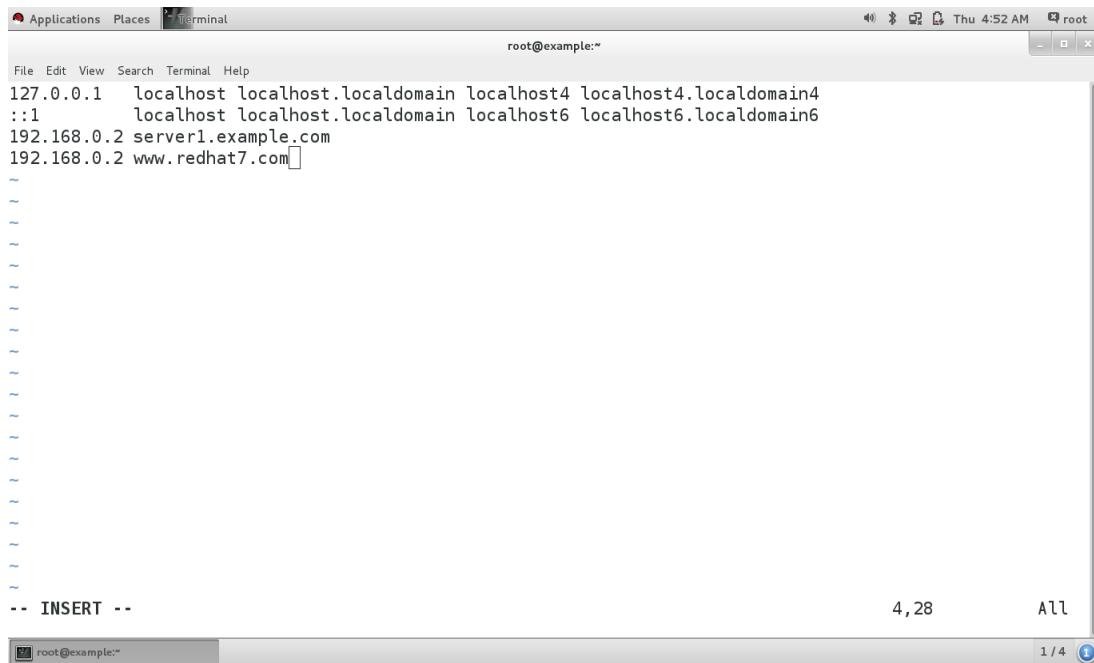
A screenshot of a terminal window titled "Terminal". The window shows the command line interface of a Linux system. The user is root, as indicated by the "root@example:~" prompt at the top right. The terminal window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The status bar at the bottom right shows the date and time as "Thu 4:52 AM" and the user as "root". The main area of the terminal displays the contents of the "/etc/hosts" file. The file contains the following entries:

```
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.0.2 server1.example.com
```

The entry "192.168.0.2 server1.example.com" is highlighted with a blue selection bar. Below the file content, the status bar shows "3,1" and "All". At the bottom of the terminal window, there is a toolbar with icons for file operations like "New", "Open", "Save", etc., and a status bar showing "1 / 4" and a small icon.

Figure 3.11.10: Syntax error

- Now set IP address and virtual server name.



A screenshot of a terminal window titled "Terminal". The window shows the command line interface of a Linux system. The user is root, as indicated by the "root@example:~" prompt at the top right. The terminal window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The status bar at the bottom right shows the date and time as "Thu 4:52 AM" and the user as "root". The main area of the terminal displays the contents of the "/etc/hosts" file. The file contains the following entries:

```
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.0.2 server1.example.com
192.168.0.2 www.redhat7.com
```

The entry "192.168.0.2 www.redhat7.com" is highlighted with a blue selection bar. Below the file content, the status bar shows "4,28" and "All". At the bottom of the terminal window, there is a toolbar with icons for file operations like "New", "Open", "Save", etc., and a status bar showing "1 / 4" and a small icon.

Figure 3.11.1: IP address and virtual server

- Press(:wq!) to save and exit.

Figure 3.11.12: Save and exit

Step 3.4.7

- Now set virtual server enable and restart.



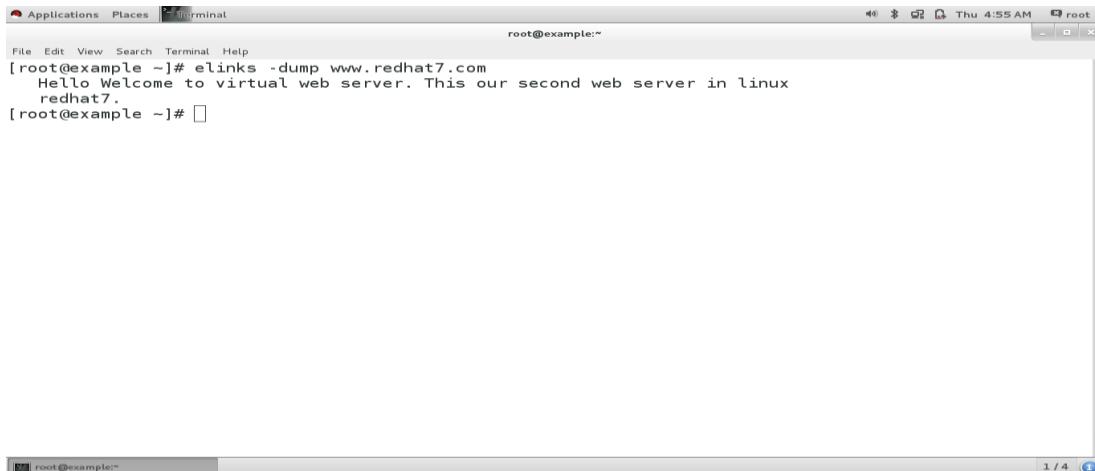
The screenshot shows a terminal window in an Ubuntu desktop environment. The title bar says "Terminal". The menu bar includes "Applications", "Places", "Terminal", "File", "Edit", "View", "Search", "Terminal", and "Help". The status bar at the bottom shows "root@example:~". The terminal content is as follows:

```
[root@example ~]# systemctl enable httpd.service
[root@example ~]# systemctl restart httpd.service
[root@example ~]#
```

Figure 3.11.13: Enable and restart

Step 3.4.8

- Press (elinks -dumpwww.redhat7.com) to checked output with command.



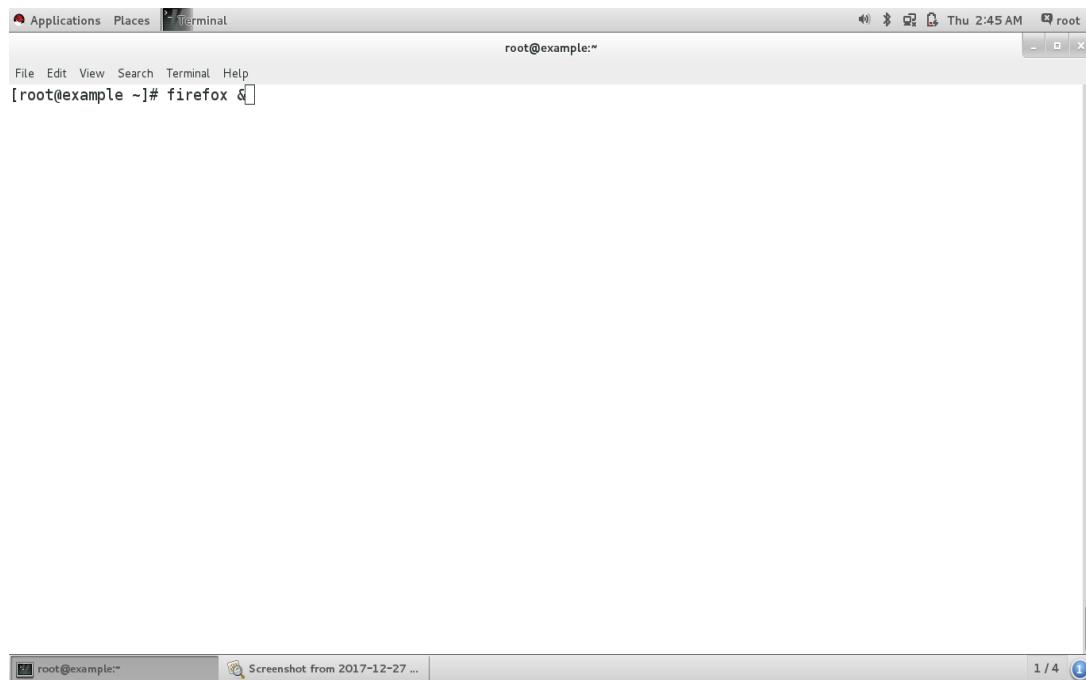
The screenshot shows a terminal window titled "Terminal" with the title bar "root@example ~". The window contains the following text:

```
File Edit View Search Terminal Help
[root@example ~]# elinks -dump www.redhat7.com
Hello Welcome to virtual web server. This our second web server in linux
redhat7.
[root@example ~]#
```

The terminal window has a scroll bar on the right side. The status bar at the bottom shows "root@example:" and "1 / 4".

Figure 3.11.14: Checked output with command

Step 3.4.9



The screenshot shows a terminal window titled "Terminal" with the title bar "root@example ~". The window contains the following text:

```
File Edit View Search Terminal Help
[root@example ~]# firefox &
```

The terminal window has a scroll bar on the right side. The status bar at the bottom shows "root@example:" and "1 / 4".

Figure 3.11.15: Checked output with command

- Now checked output.

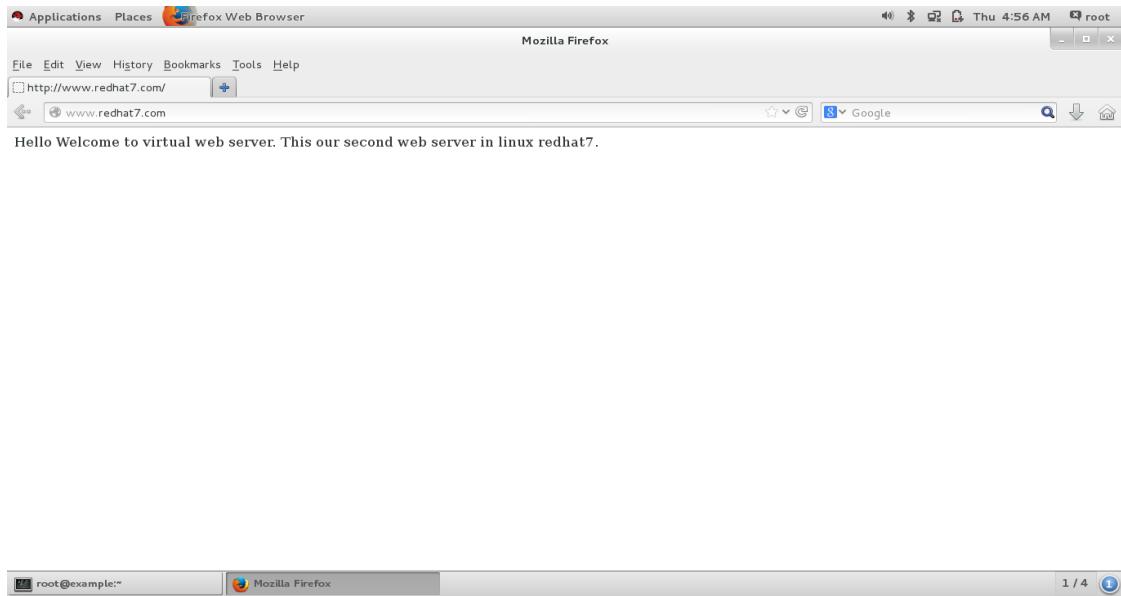


Figure 3.11.16: Checked output

3.5 Step of restriction process

Step 3.5.0

- Press(mkdir /var/www/linux) to create a sub folder

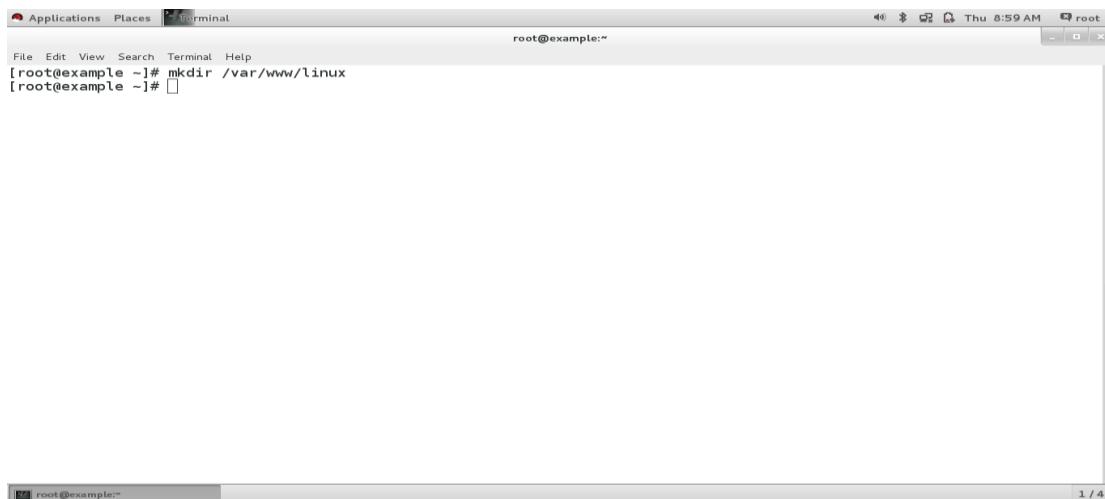
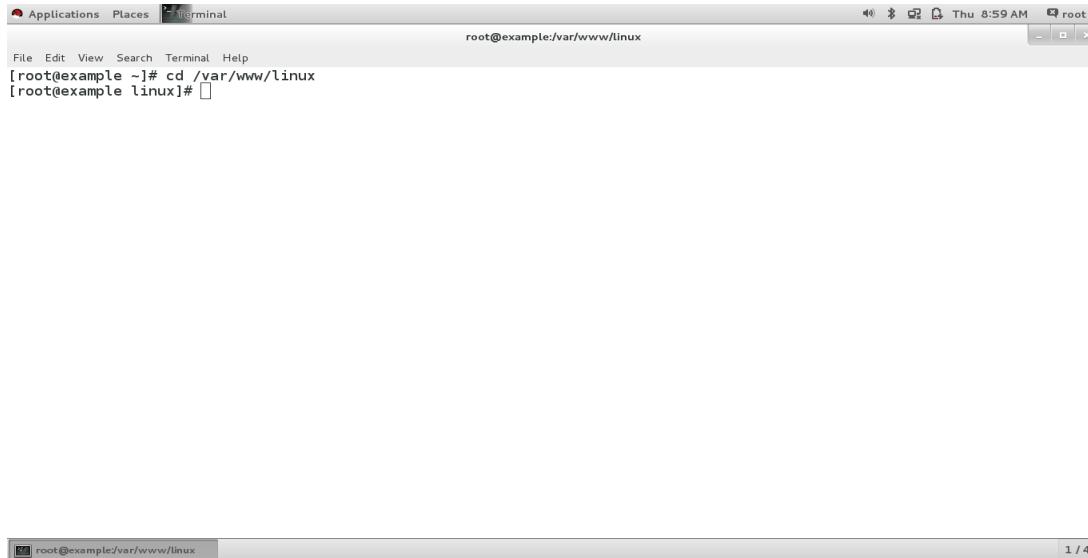


Figure 3.12.1: Create sub folder

Step 3.5.1

- Press (cd /var/www/linux) to enter the sub folder.

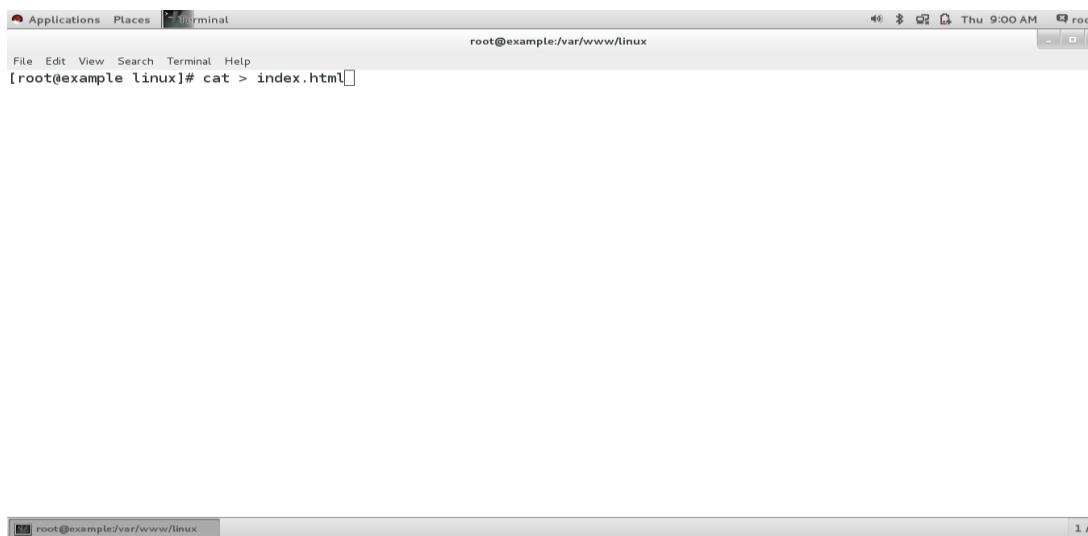


A screenshot of a terminal window titled "terminal". The window is running as root on a Linux system. The title bar shows "root@example:~]" and the status bar shows "root@example:/var/www/linux". The terminal window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The main area of the terminal shows the command [root@example ~]# cd /var/www/linux being typed. The status bar at the bottom of the terminal window also displays "root@example:/var/www/linux".

Figure 3.12.2: Sub folder

Step 3.5.2

- Press (cat > index.html) to create a new index file for stored new information.

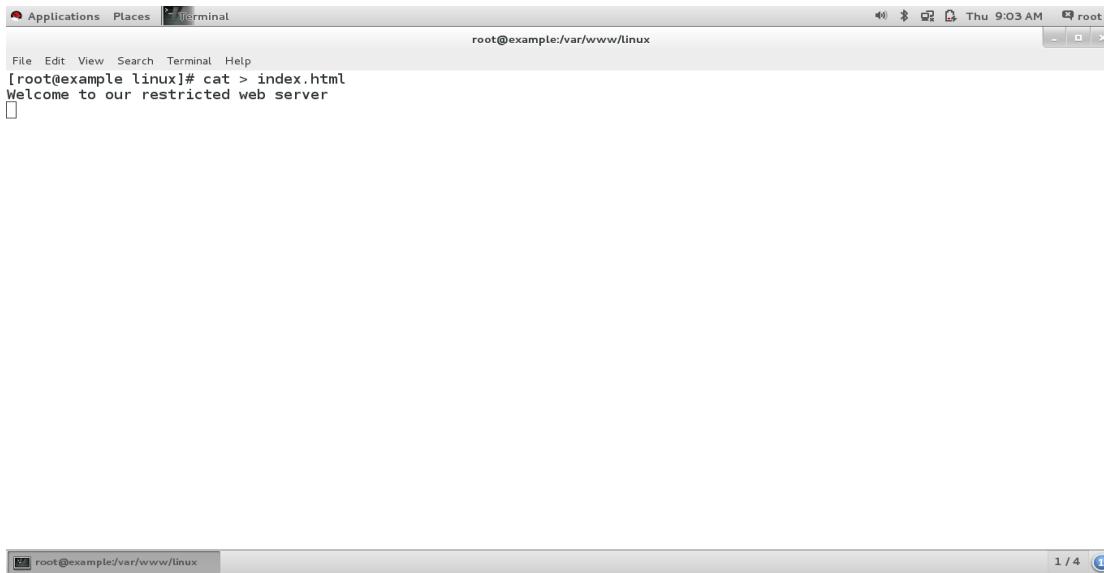


A screenshot of a terminal window titled "terminal". The window is running as root on a Linux system. The title bar shows "root@example:~]" and the status bar shows "root@example:/var/www/linux". The terminal window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The main area of the terminal shows the command [root@example linux]# cat > index.html being typed. The status bar at the bottom of the terminal window also displays "root@example:/var/www/linux".

Figure 3.12.3: New index file

Step 3.5.3

- Now show the webpage information. Which user are allowed.



A screenshot of a Linux desktop environment showing a terminal window. The window title is "Terminal". The terminal prompt is "root@example:~\$". The user has run the command "cat > index.html" and entered the content "Welcome to our restricted web server". The terminal window is part of a larger desktop interface with a menu bar at the top and a taskbar at the bottom.

Figure 3.12.4: Webpage information

Step 3.5.4

- To exit from sub folder press (cd).



A screenshot of a Linux desktop environment showing a terminal window. The window title is "Terminal". The terminal prompt is "root@example:~\$". The user has run the command "cd" to exit the current directory. The terminal window is part of a larger desktop interface with a menu bar at the top and a taskbar at the bottom.

Figure 3.12.5: Exit of sub folders

Step 3.5.5

- Now press (vim /etc/httpd/conf/httpd.conf) this is main path of restricted configuration. Here set the all configuration of allow and deny protocol.



A screenshot of a terminal window titled "Terminal". The window shows the command "root@example:~ # vim /etc/httpd/conf/httpd.conf" being typed. The terminal is running as root, as indicated by the "root" icon in the title bar and the "root" prompt at the bottom. The window has a standard Linux desktop interface with a menu bar and a status bar at the top.

Figure 3.12.6: Restricted configuration

Step 3.5.6

- Here set the server name ,directory index, Document root, Allow IP and Deny IP.
- Press (:wq!) to Save and exit.



A screenshot of a terminal window titled "Terminal". The window displays the configuration of multiple virtual hosts in the httpd.conf file. The configuration includes setting server names like "server1.example.com" and "www.redhat7.com", directory indexes like "index.html", and document roots like "/var/www/html" and "/var/www/hello". It also includes a section for a virtual host on port 192.168.0.2:80, which contains a directory block with "allow" and "deny" rules. The command ":wq!" is shown at the bottom, indicating the user is saving and exiting the editor.

Figure 3.12.7: Set se.name,dir.index,do.root,deny IP

Step 3.5.7

- Press (httpd -t) to checked syntax errors.

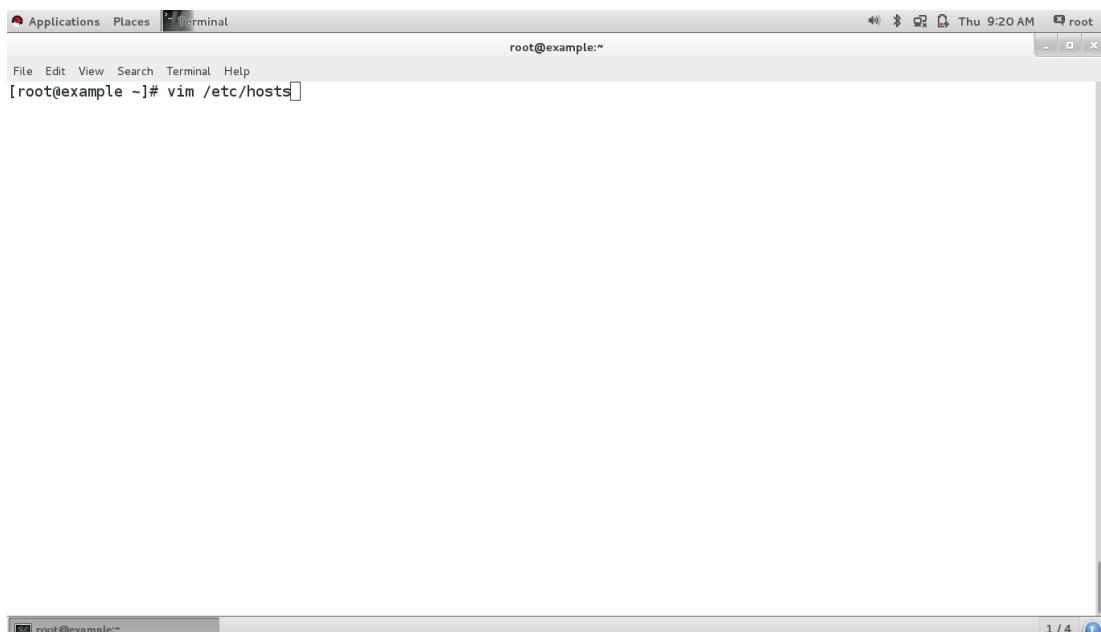


A screenshot of a terminal window titled "Terminal". The window shows the command [root@example ~]# httpd -t being run, followed by the output "Syntax: OK". The terminal window has a standard Linux desktop interface with a title bar, menu bar, and status bar indicating the date and time.

Figure 3.12.8: Syntax error

Step 3.5.8

- Press (vim /etc/hosts) to set the restricted server name and IP address.

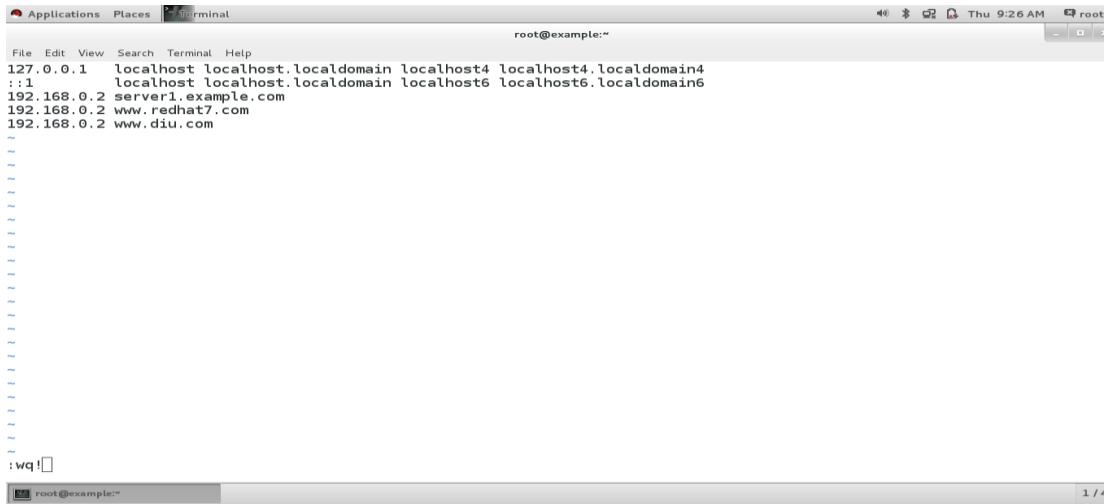


A screenshot of a terminal window titled "Terminal". The window shows the command [root@example ~]# vim /etc/hosts being run. The terminal window has a standard Linux desktop interface with a title bar, menu bar, and status bar indicating the date and time.

Figure 3.12.9: Restricteed server name and IP

Step 3.5.9

- Now show the server name and IP address , which is previously setted.
- Press (:wq!) to save and exit.



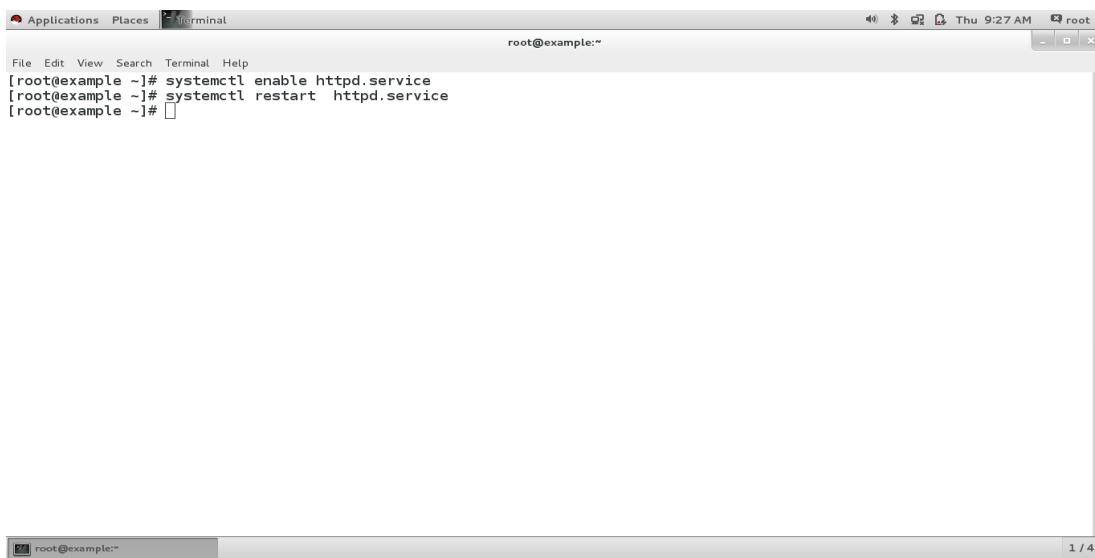
```
root@example:~# cat /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.0.2 server1.example.com
192.168.0.2 www.redhat7.com
192.168.0.2 www.diu.com

root@example:~# :wq!
```

Figure 3.12.10: Showing server name and IP

Step 3.5.10

- To enable and restart the restricted web server.



```
root@example:~# systemctl enable httpd.service
[root@example ~]# systemctl restart httpd.service
[root@example ~]#
```

Figure 3.12.11: Enable and restart

Step 3.5.11

- Now open the browser and hit server domain name, if allowed then access the web server.

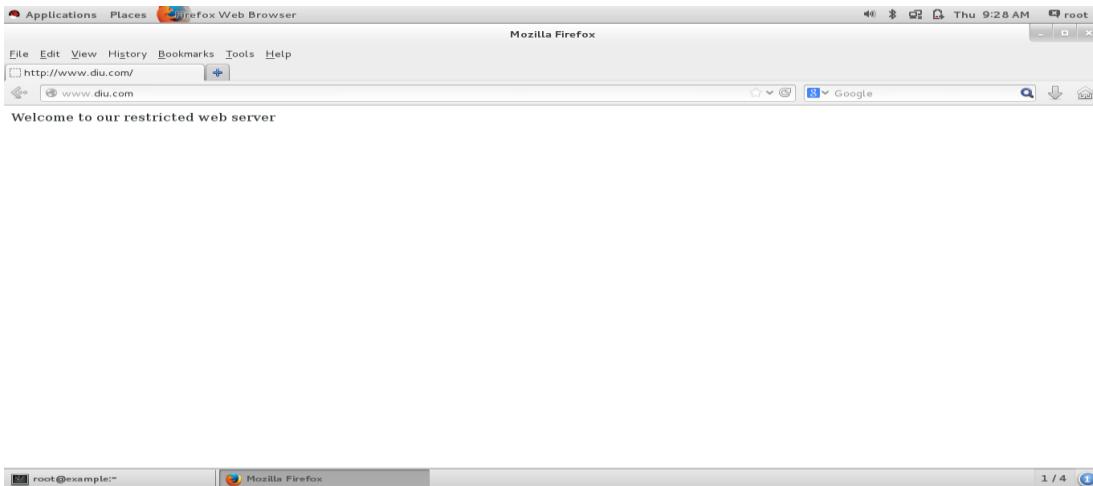


Figure 3.12.12: Opening Browser

Step 3.5.12

- Now hit the browser from deny IP.

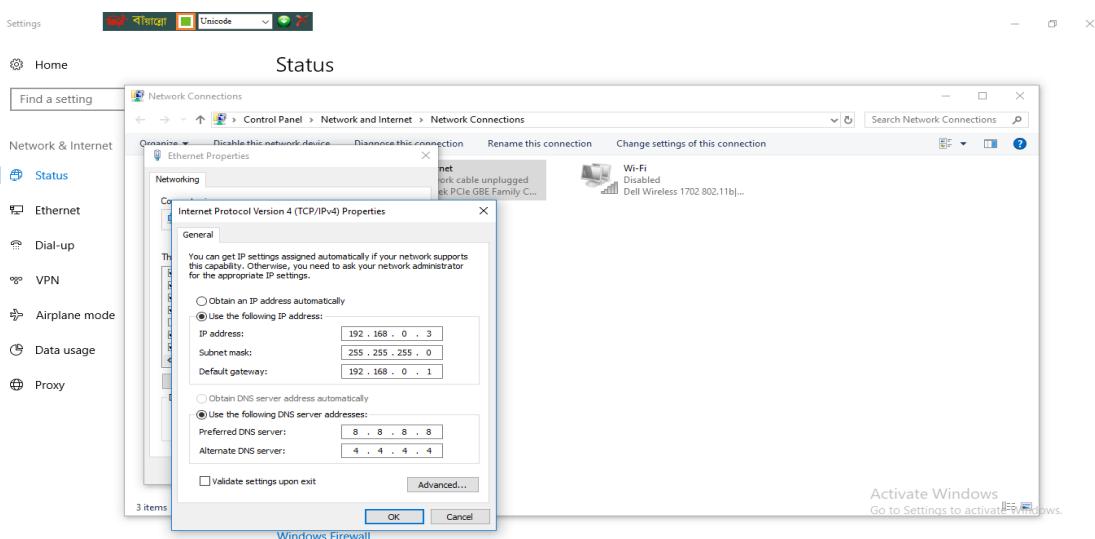


Figure 3.12.13: Show of IP

- Now show webpage as this.

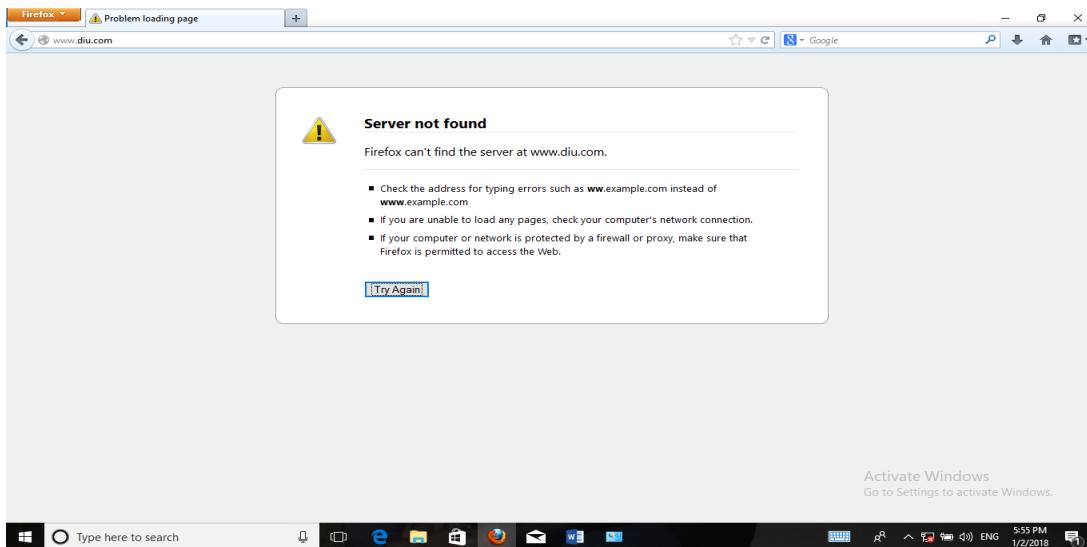


Figure 3.12.14: Show web page

Step 3.5.13

- This is allow system from another IP.

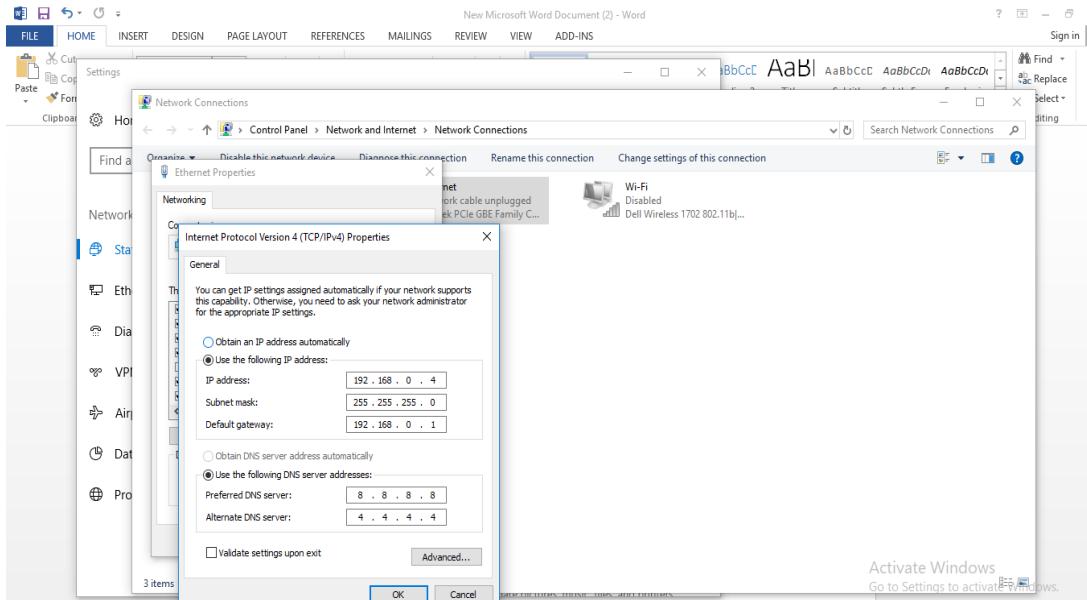


Figure 3.12.15: IP address

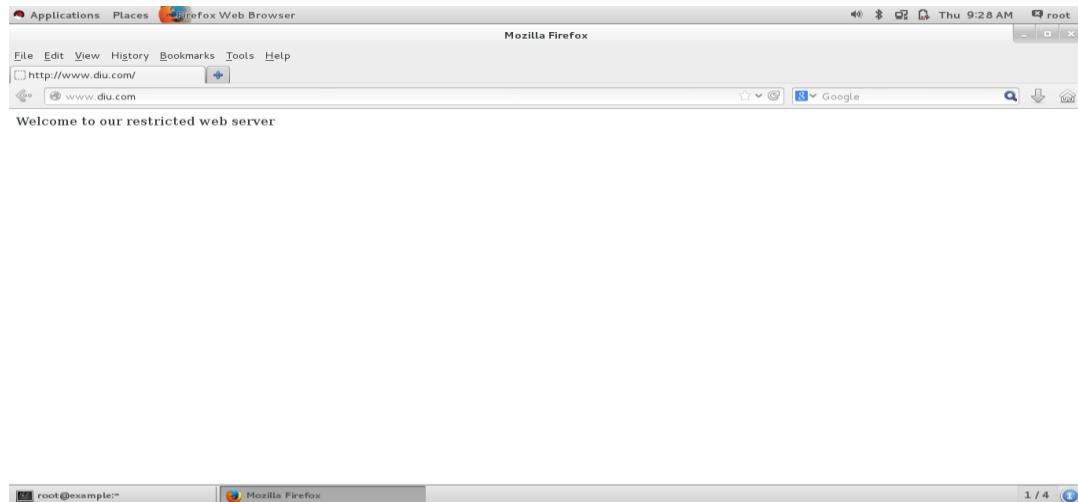


Figure 3.12.16: IP address

3.6 HTTPS Server

Step 3.6.0

- Press (cd desktop/server-cert ls) to check server cert folder.
- And save them in this path.

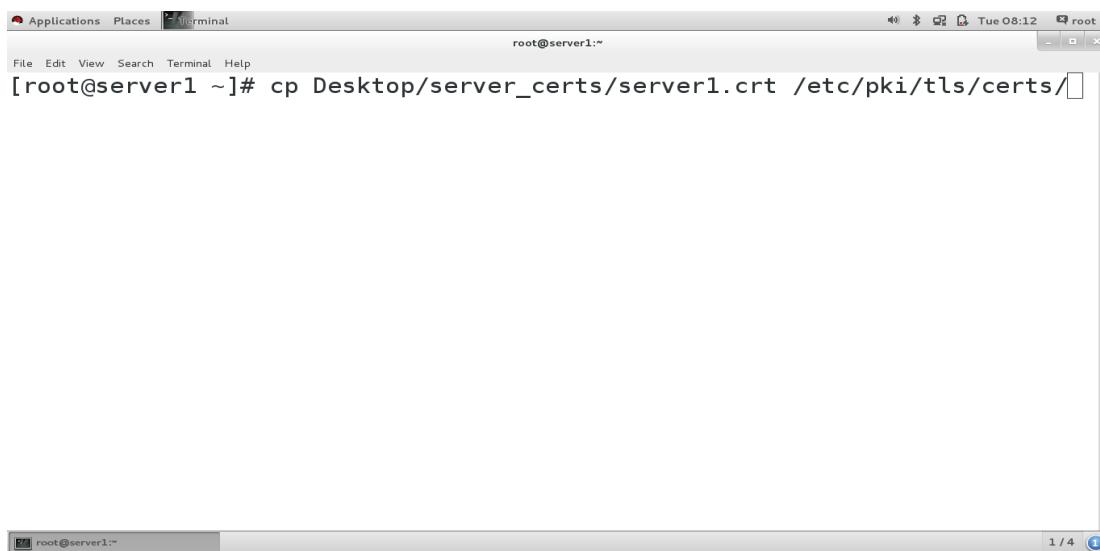
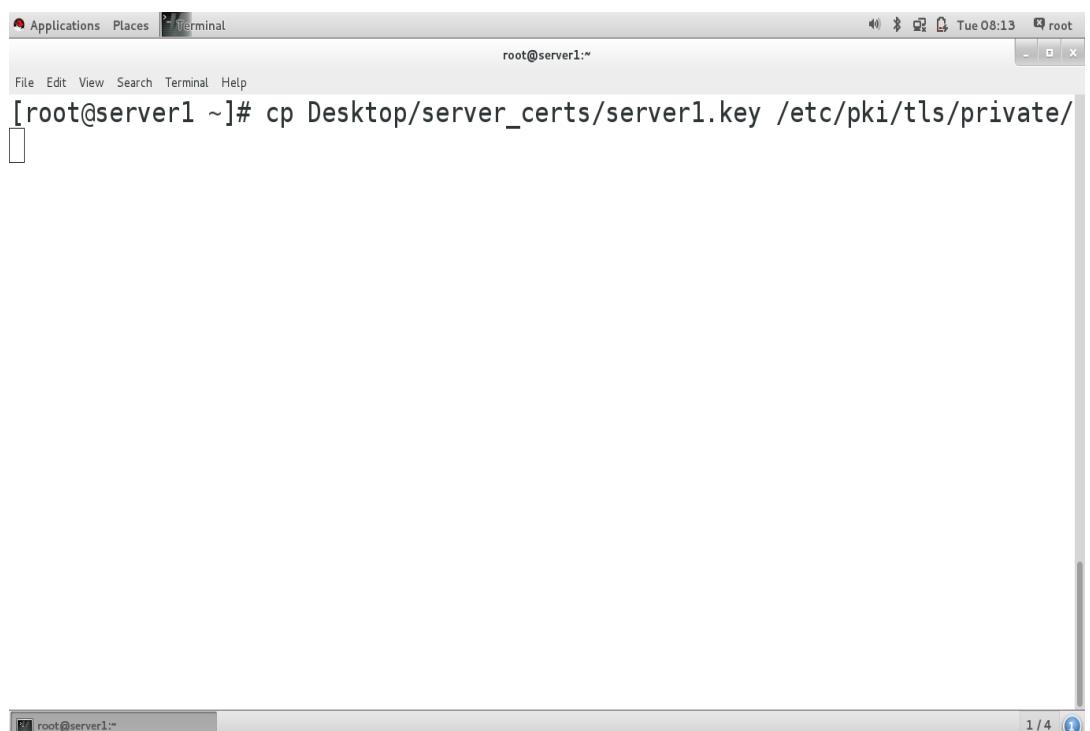


Figure 3.12.17: Copy of certs file



```
Applications Places Terminal
root@server1:~#
File Edit View Search Terminal Help
[root@server1 ~]# cp Desktop/server_certs/example-ca.crt /etc/pki/tls/certs
/
```

Figure 3.12.18: Copy of certs file

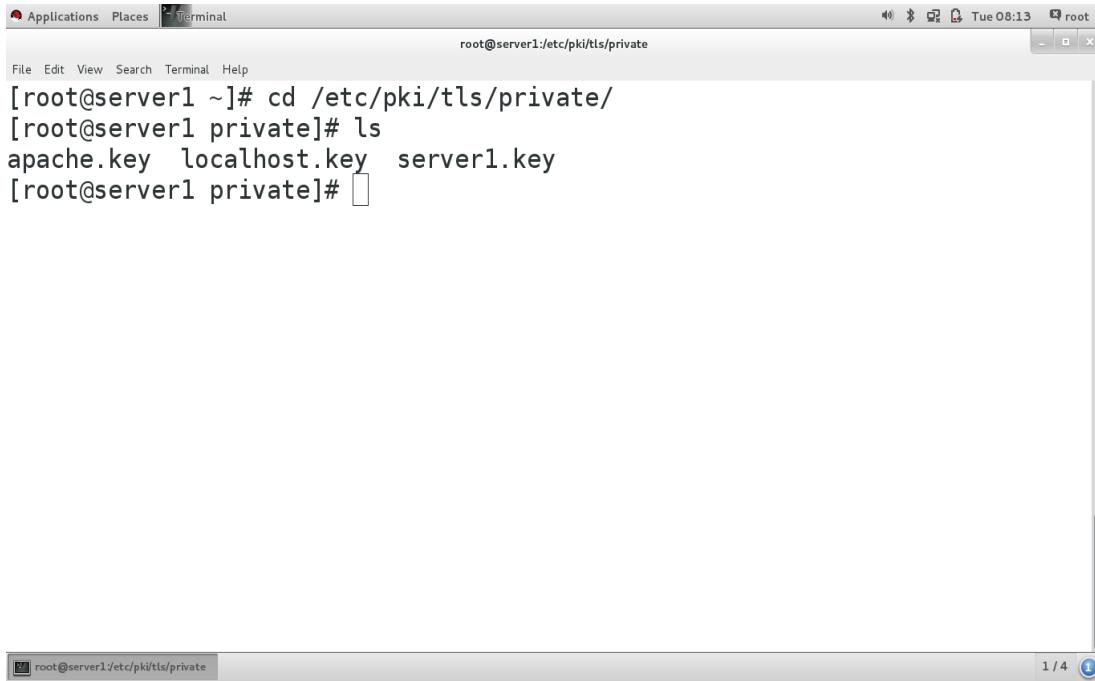


```
Applications Places Terminal
root@server1:~#
File Edit View Search Terminal Help
[root@server1 ~]# cp Desktop/server_certs/server1.key /etc/pki/tls/private/
/
```

Figure 3.12.19: Copy of certs key file

Step 3.6.1

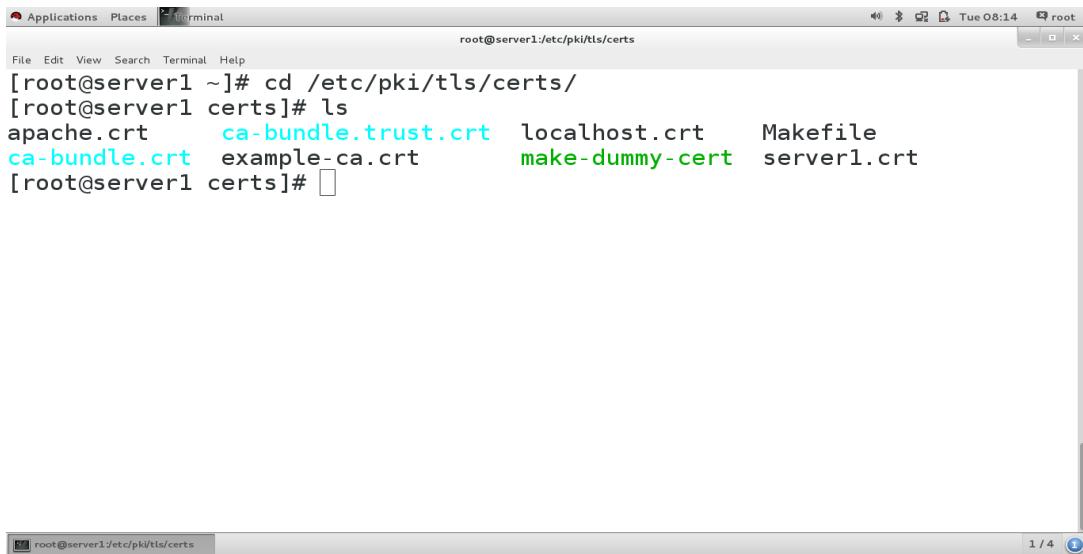
- To checked that file was copied properly ?



A screenshot of a terminal window titled "Terminal". The window shows the command line interface of a Linux system. The prompt indicates the user is root at server1, in the directory /etc/pki/tls/private. The user has run the command "ls" to list files in this directory. The output shows three files: apache.key, localhost.key, and server1.key.

```
root@server1:/etc/pki/tls/private
[root@server1 ~]# cd /etc/pki/tls/private/
[root@server1 private]# ls
apache.key  localhost.key  server1.key
[root@server1 private]#
```

Figure 3.12.20: Checked file



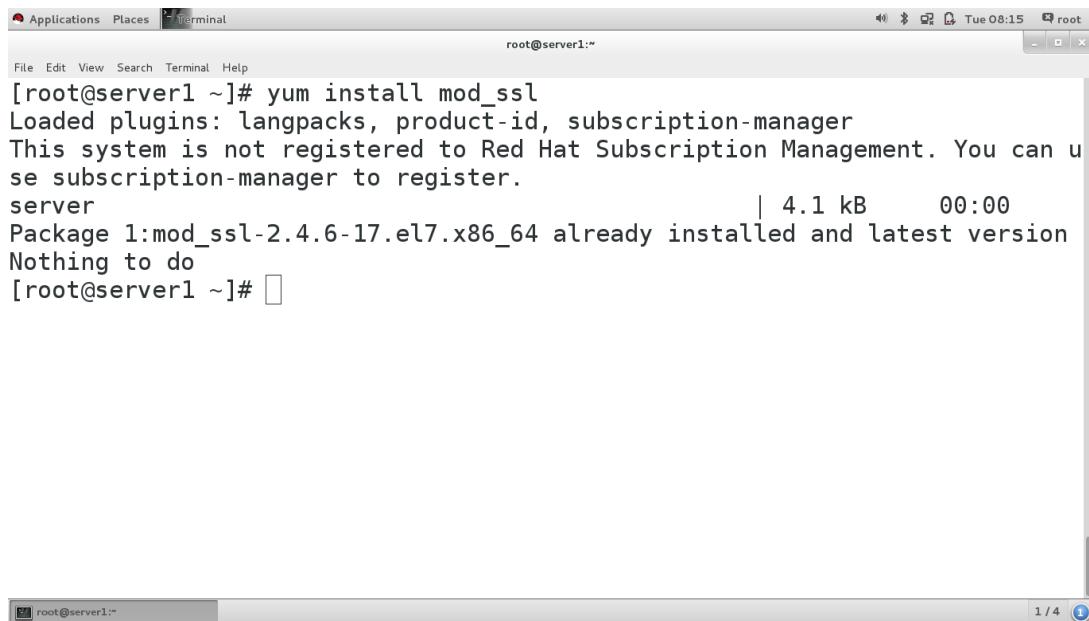
A screenshot of a terminal window titled "Terminal". The window shows the command line interface of a Linux system. The prompt indicates the user is root at server1, in the directory /etc/pki/tls/certs. The user has run the command "ls" to list files in this directory. The output shows several files: apache.crt, ca-bundle.trust.crt, localhost.crt, Makefile, ca-bundle.crt, example-ca.crt, make-dummy-cert, and server1.crt.

```
root@server1:/etc/pki/tls/certs
[root@server1 ~]# cd /etc/pki/tls/certs/
[root@server1 certs]# ls
apache.crt      ca-bundle.trust.crt  localhost.crt    Makefile
ca-bundle.crt   example-ca.crt     make-dummy-cert  server1.crt
[root@server1 certs]#
```

Figure 3.12.21: Checked file

Step 3.6.2

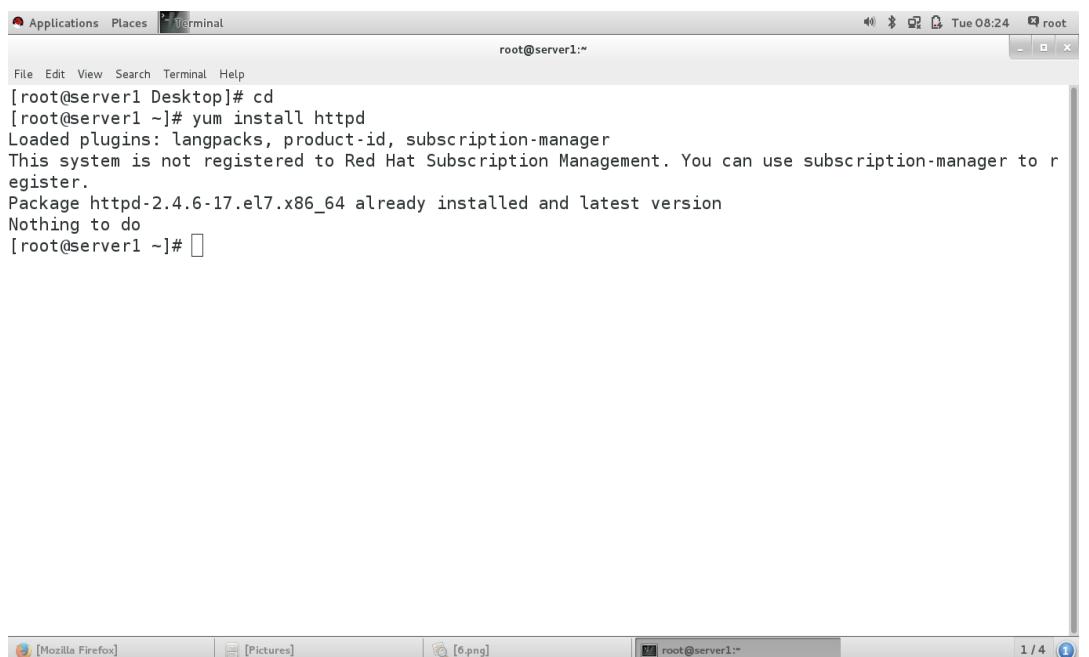
- Now install the packages(httpd& mod-ssl).



A screenshot of a terminal window titled "Terminal". The window shows a root shell session on a server named "server1". The user runs the command "yum install mod_ssl". The output indicates that the package is already installed and up-to-date. The terminal window has a standard Linux desktop interface with icons for Applications, Places, and Terminal at the top. The status bar at the bottom shows "root@server1:~" and the date/time "Tue 08:15". A scroll bar is visible on the right side of the terminal window.

```
[root@server1 ~]# yum install mod_ssl
Loaded plugins: langpacks, product-id, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use subscription-manager to register.
server | 4.1 kB     00:00
Package 1:mod_ssl-2.4.6-17.el7.x86_64 already installed and latest version
Nothing to do
[root@server1 ~]#
```

Figure 3.12.22: Installing packages



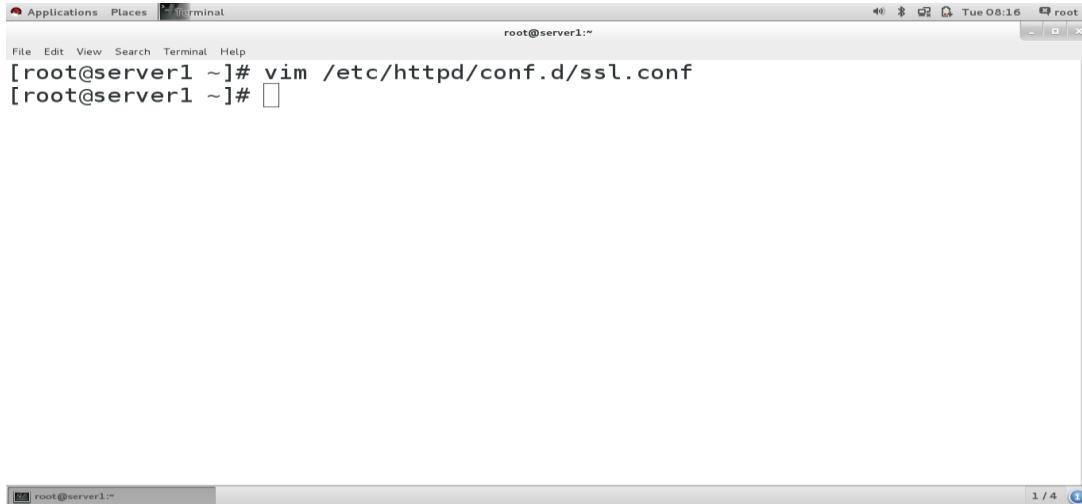
A screenshot of a terminal window titled "Terminal". The window shows a root shell session on a server named "server1". The user runs the command "cd Desktop" followed by "yum install httpd". The output shows that the package is already installed and up-to-date. The terminal window has a standard Linux desktop interface with icons for Applications, Places, and Terminal at the top. The status bar at the bottom shows "root@server1:~" and the date/time "Tue 08:24". A scroll bar is visible on the right side of the terminal window.

```
[root@server1 Desktop]# cd
[root@server1 ~]# yum install httpd
Loaded plugins: langpacks, product-id, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use subscription-manager to register.
Package httpd-2.4.6-17.el7.x86_64 already installed and latest version
Nothing to do
[root@server1 ~]#
```

Figure 3.12.23: Installing packages

Step 3.6.3

- Press (vim /etc/httpd/conf.d/ssl.conf)to access ssl configuration.

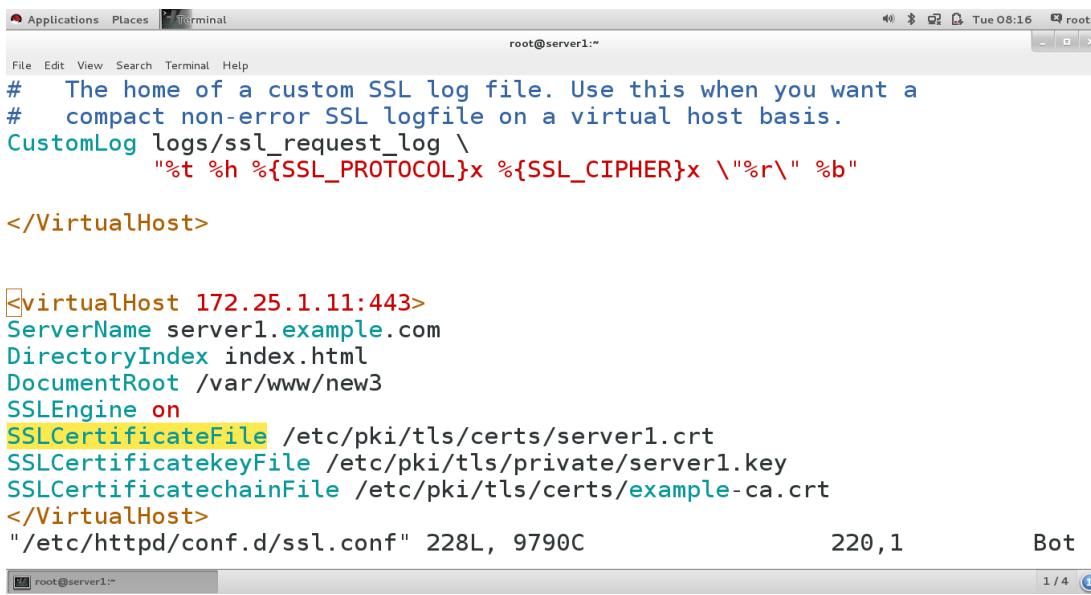


```
root@server1 ~]# vim /etc/httpd/conf.d/ssl.conf
[root@server1 ~]#
```

Figure 3.12.24: SSL configuration

Step 3.6.4

- Now configure under previous configuration file.



```
#   The home of a custom SSL log file. Use this when you want a
#   compact non-error SSL logfile on a virtual host basis.
CustomLog logs/ssl_request_log \
    "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"

</VirtualHost>

<virtualHost 172.25.1.11:443>
ServerName server1.example.com
DirectoryIndex index.html
DocumentRoot /var/www/new3
SSLEngine on
SSLCertificateFile /etc/pki/tls/certs/server1.crt
SSLCertificatekeyFile /etc/pki/tls/private/server1.key
SSLCertificatechainFile /etc/pki/tls/certs/example-ca.crt
</VirtualHost>
"/etc/httpd/conf.d/ssl.conf" 228L, 9790C          220,1      Bot
```

Figure 3.12.25: Configuration file

Step 3.6.5

- Now save them.

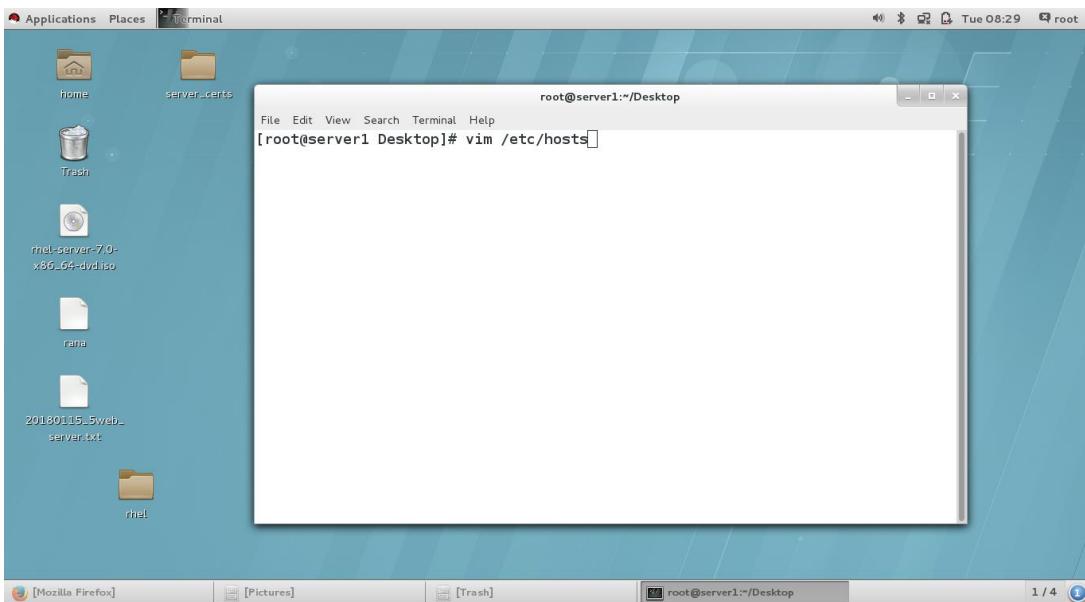


Figure 3.12.26: Save files

Step 3.6.6

- Press(vim /etc/hosts) to access host file.
- And set domain name.

A screenshot of a terminal window showing the contents of the /etc/hosts file. The window title is 'root@server1:~'. The file contains the following entries:

```
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
#192.168.0.2 www.rana.com
172.25.1.11 server1.example.com
```

The text is displayed in a monospaced font, with the first two lines being standard loopback entries and the last two being custom host definitions. The terminal window also shows the version '5,0-1' and 'All' at the bottom right.

Figure 3.12.27: Host file

Step 3.6.7

- Now enable and restart the server.



The image shows a screenshot of a Linux desktop environment, specifically Ubuntu, with a terminal window open. The terminal window has a title bar "Terminal" and a status bar "root@server1:~". The main area of the terminal shows the following command-line session:

```
[root@server1 ~]# systemctl enable httpd.service
[root@server1 ~]# systemctl restart httpd.service
```

Figure 3.12.28: Enable and restart server

A screenshot of a Linux desktop environment. At the top, there is a horizontal menu bar with icons for Applications, Places, Terminal, and a power button. The title bar of the active window says "Terminal". The terminal window shows the command "root@server1:~" followed by a blank line. Below the terminal window, the desktop background is visible, showing a light blue gradient. A vertical dock on the left side contains icons for Pictures, Home, and Applications. A horizontal dock at the bottom contains icons for Pictures, Home, Applications, and a terminal window labeled "root@server1:~".

Figure 3.12.29: webpage information

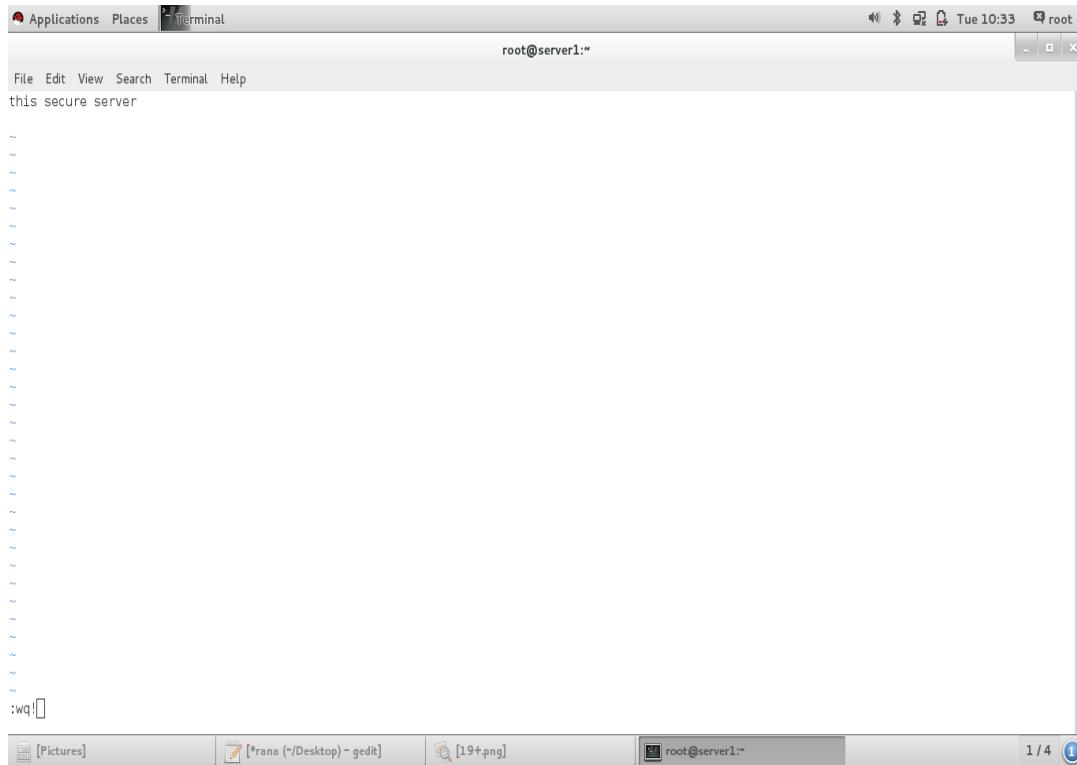


Figure 3.12.30: webpage information

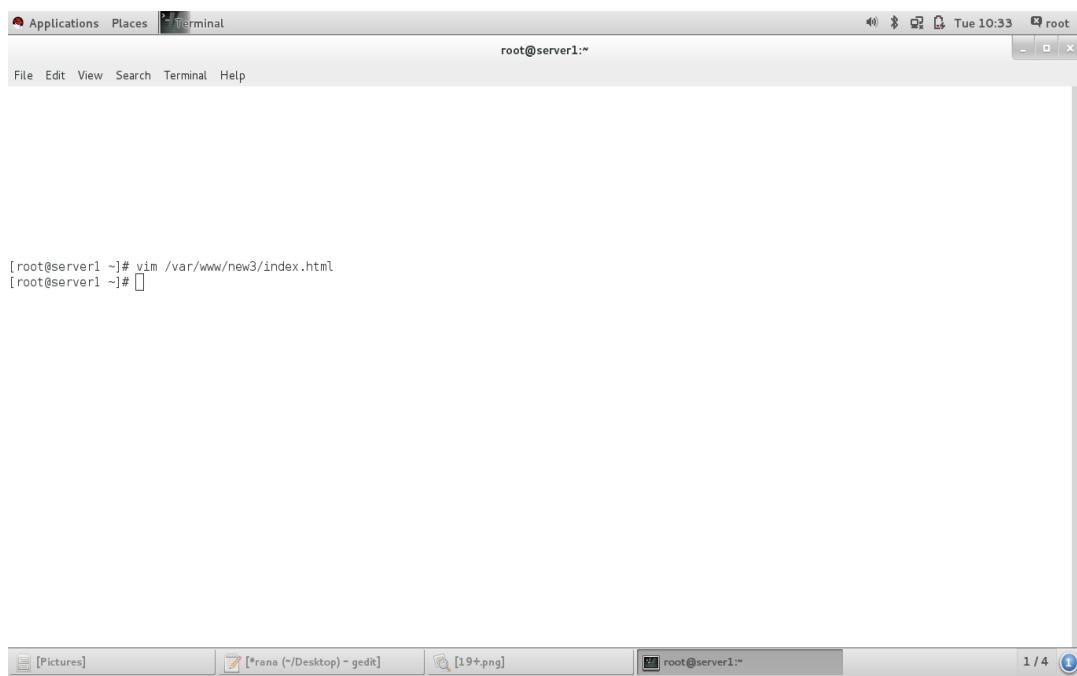


Figure 3.12.31: this is input file

Step 3.6.8

- Set pc IP address.

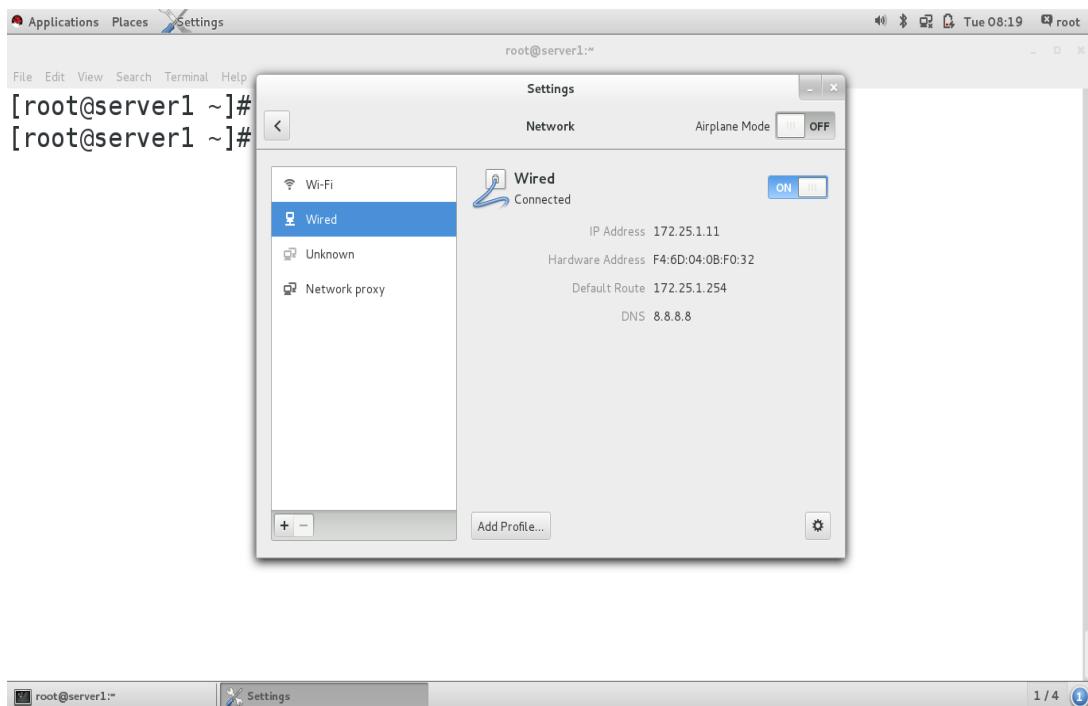


Figure 3.12.32: IP address

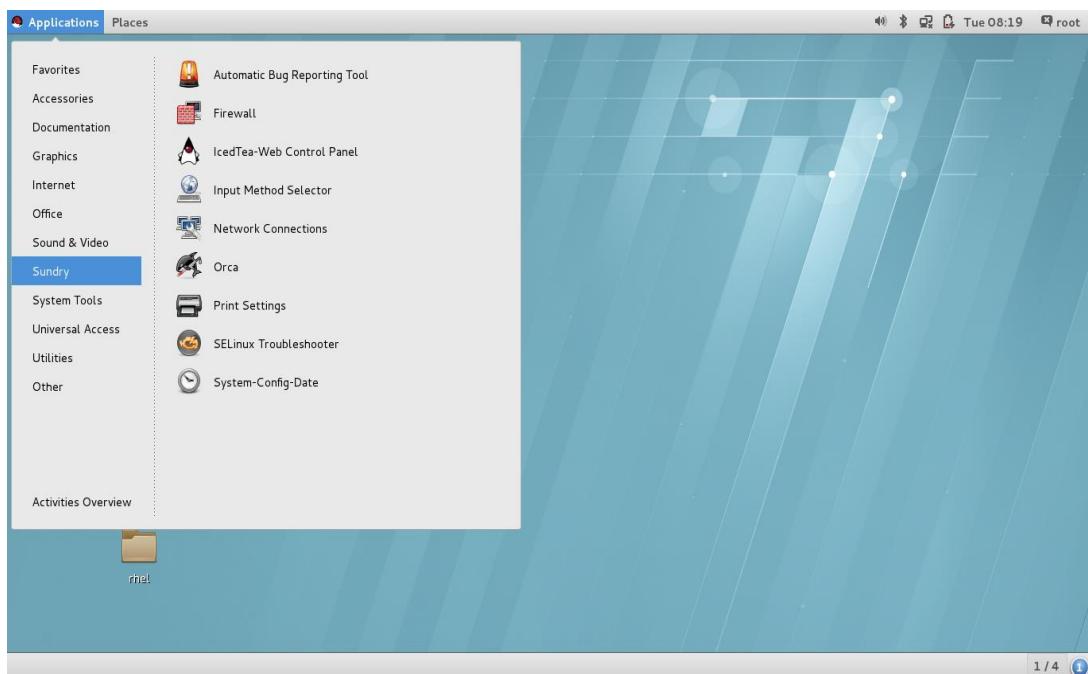


Figure 3.12.33: IP address

- Https port open as permanently.

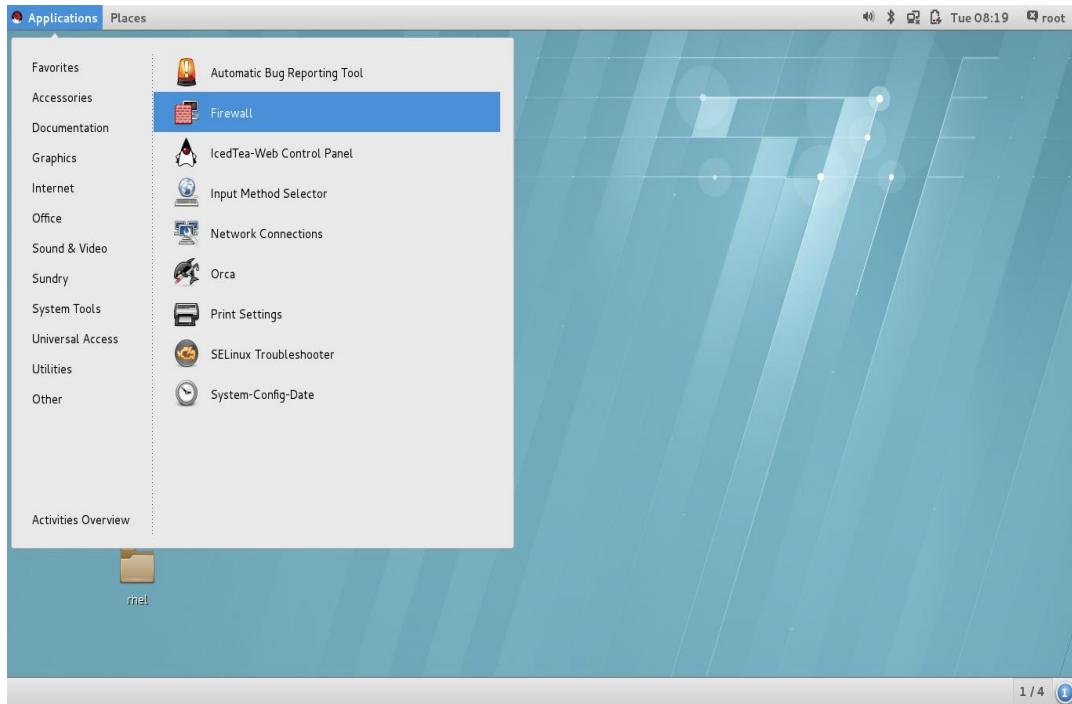


Figure 3.12.34: Opening HTTPS port

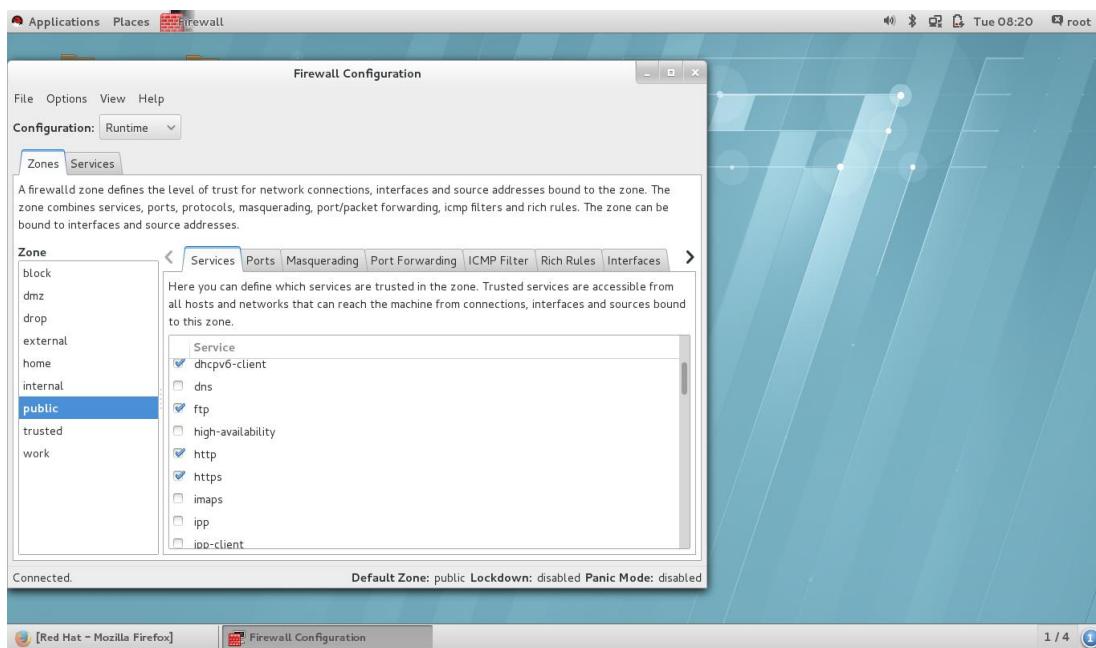


Figure 3.12.35: Opening HTTPS port

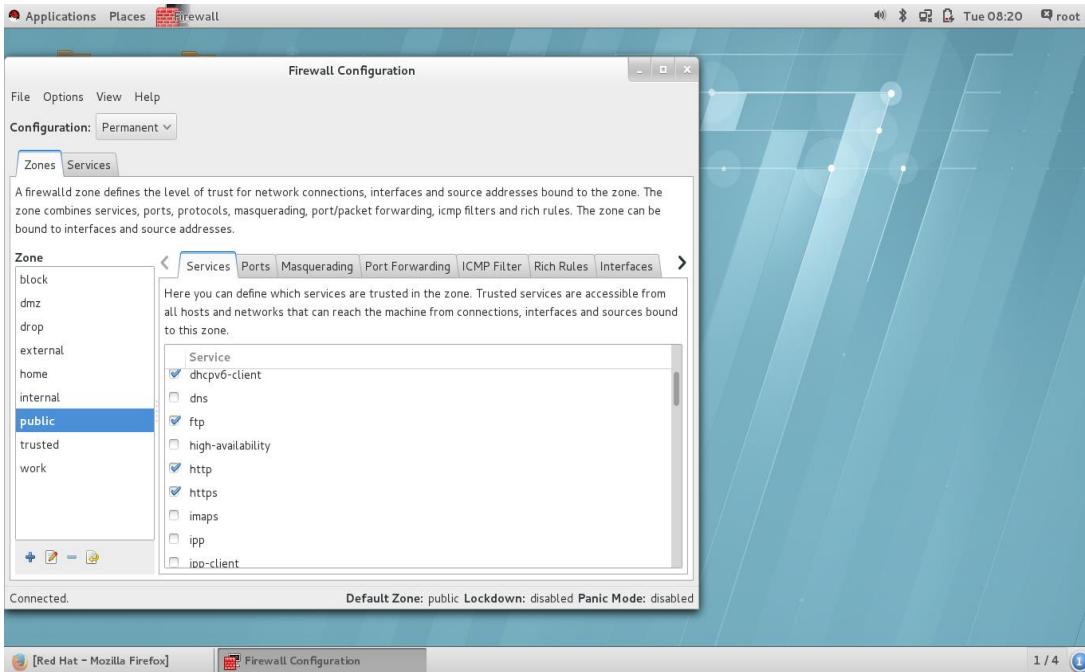


Figure 3.12.36: Opening HTTPS port

Step 3.6.9

- Open the browser.

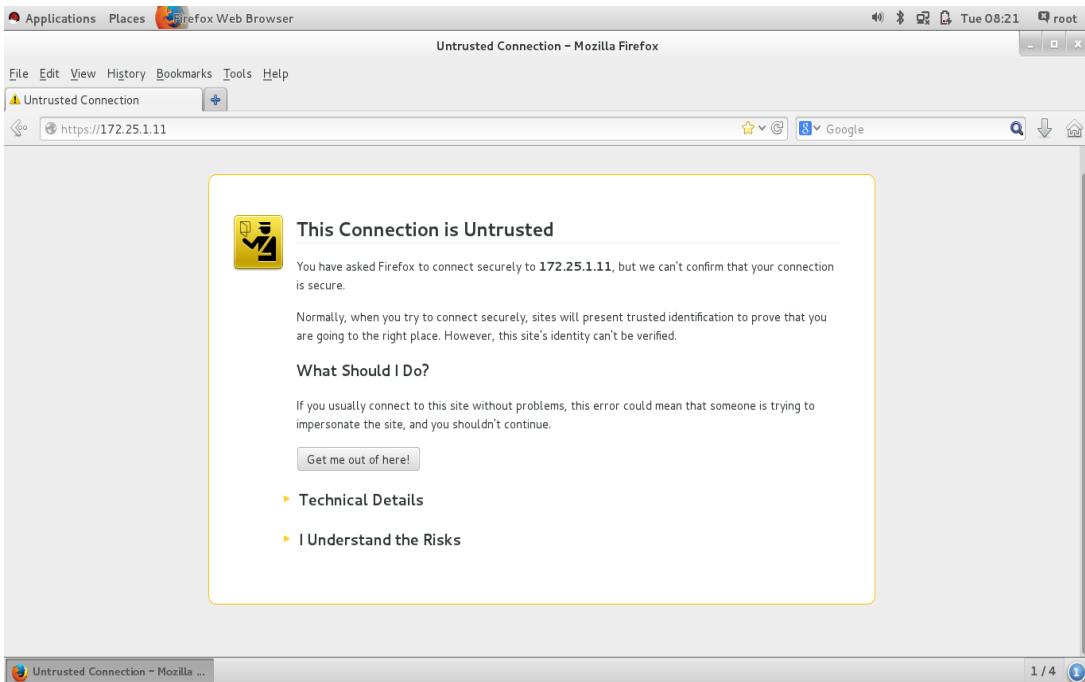


Figure 3.12.37: Opening browser

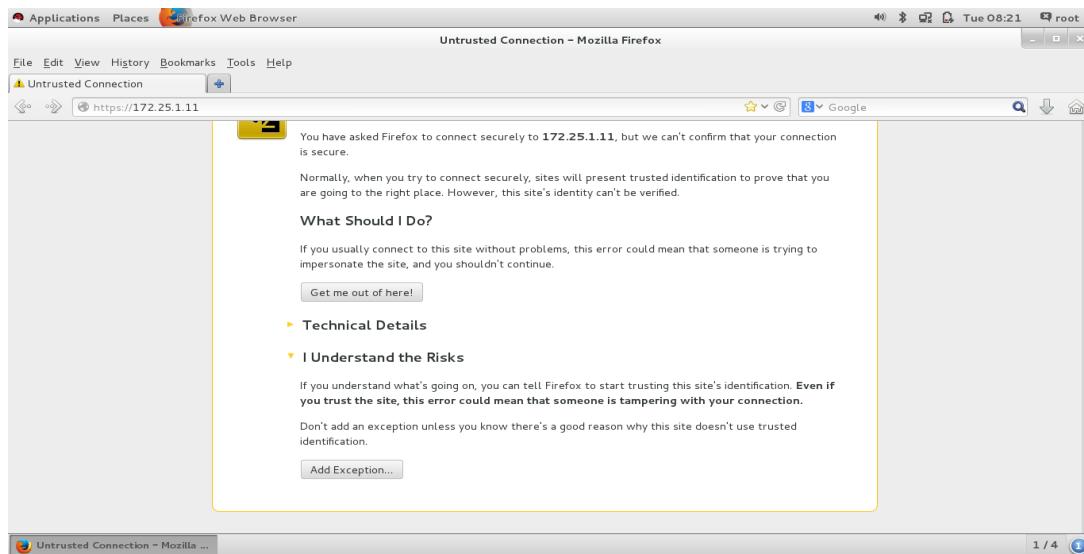


Figure 3.12.38: Opening browser

➤ Add security exception

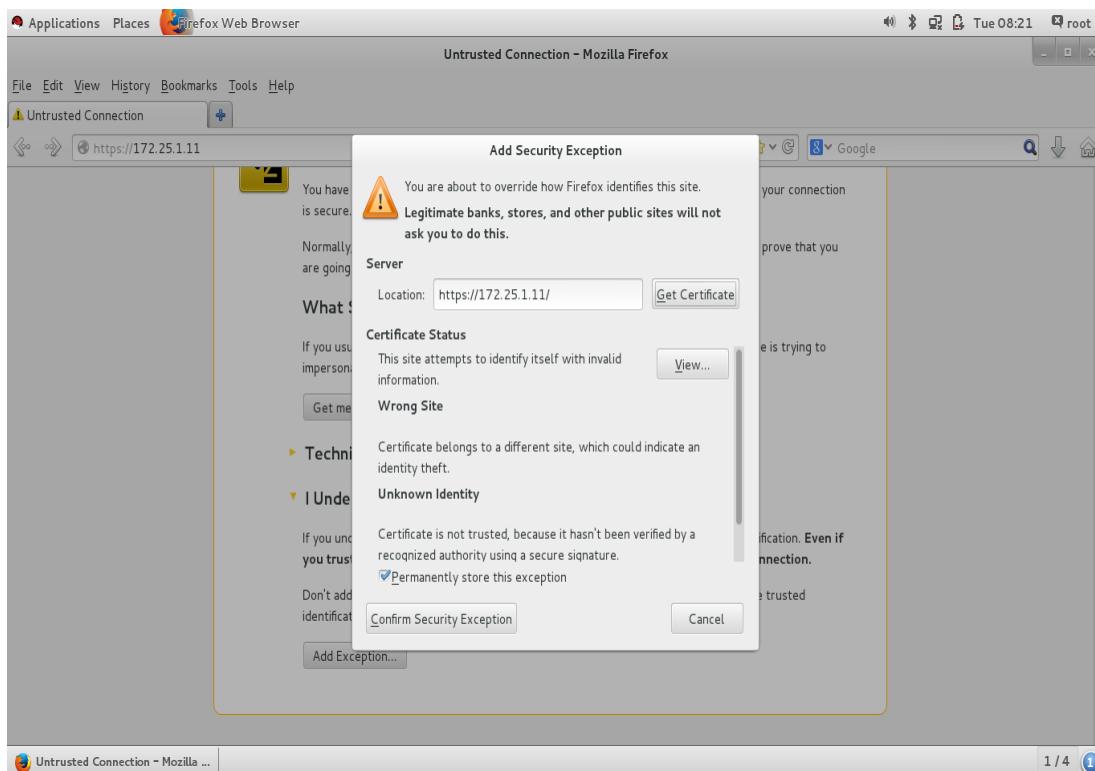


Figure 3.12.39: Security

- Get certificate.



Figure 3.12.40: Certificate

- Confirm security exception then get output.

3.7.0 Network design

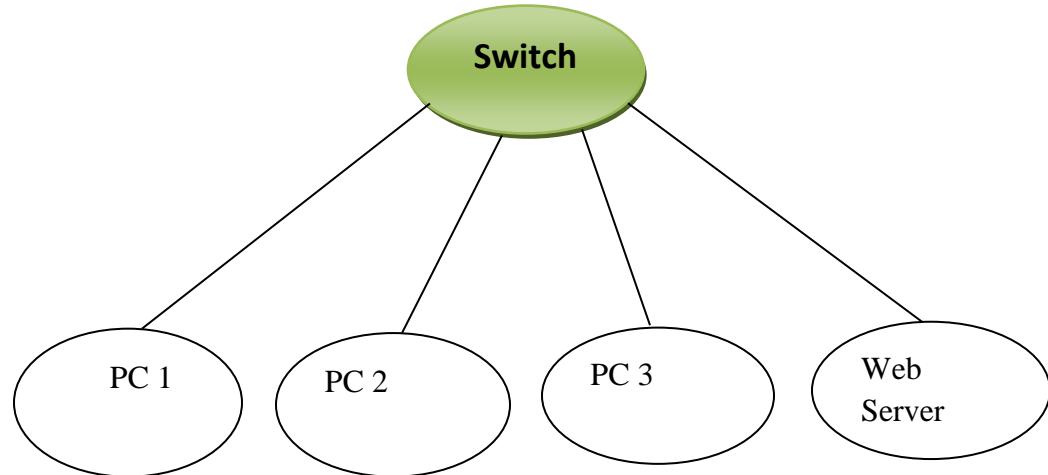


Figure 3.13.1: Network diagram

- This is Local Network , can access only connected IP.

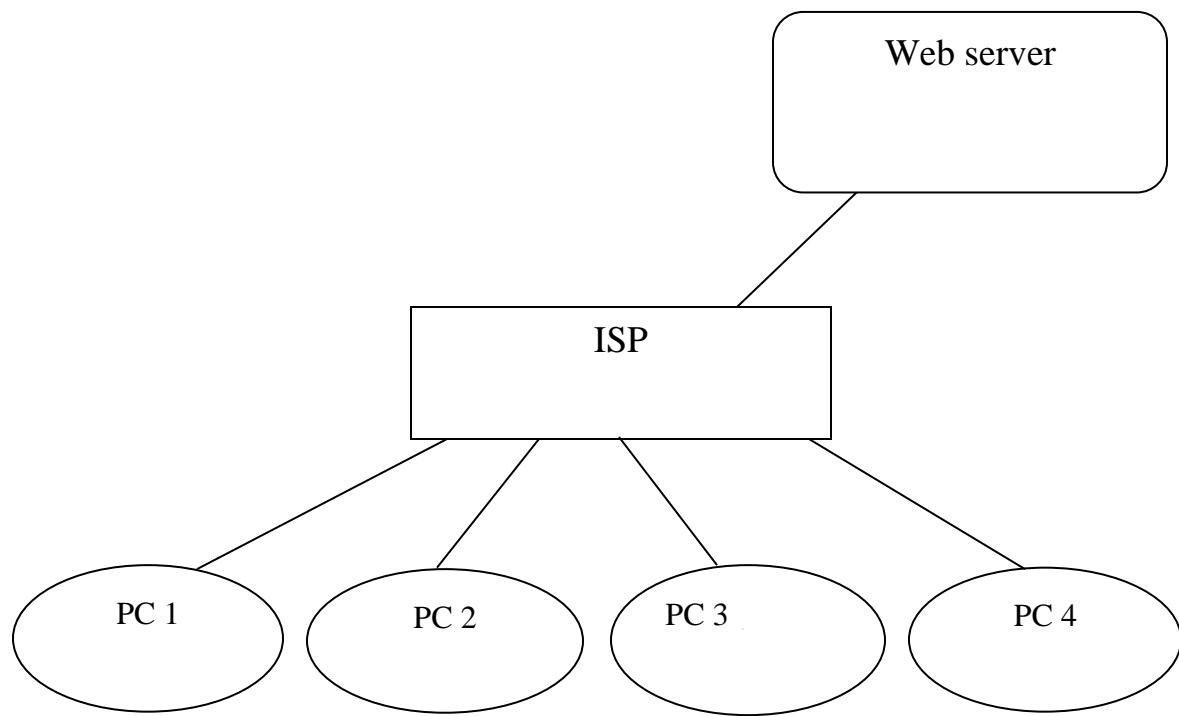


Figure 3.13.2: Network diagram

- This is www (world wide web) based network .It can access from anywhere via internet.

Summary

FTP, HTTP, Virtual, HTTPS server are configured step by step and process of restricted and package installation process.

Chapter 4

Server implementation

4.1 FTP server

Input: This is the main input path of ftp (cd /var/ftp/pub/).

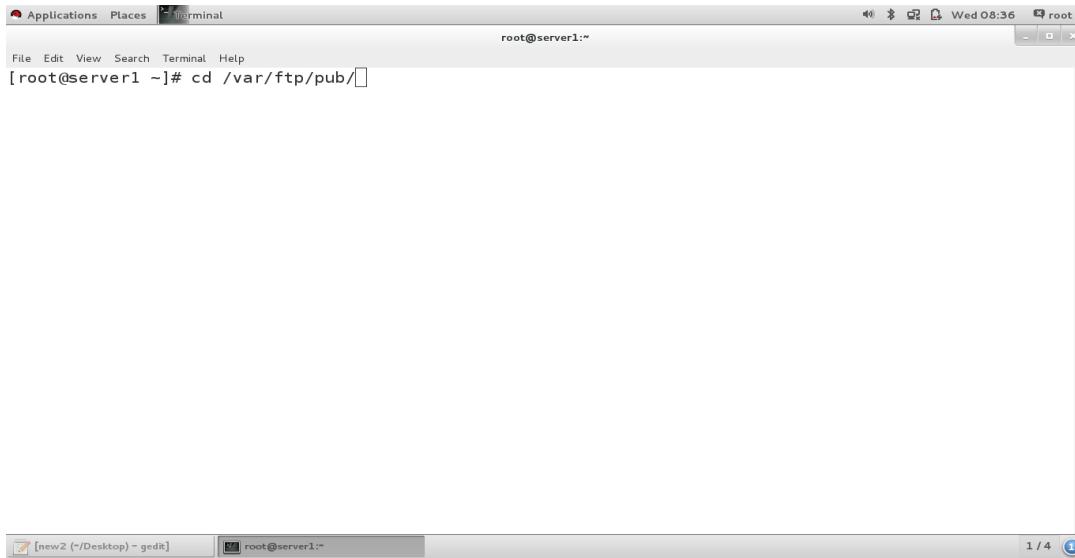


Figure 4.1: Input path of FTP

Output

1: Open the browser

Press ftp ip (ftp://172.25.1.11)

Show the output

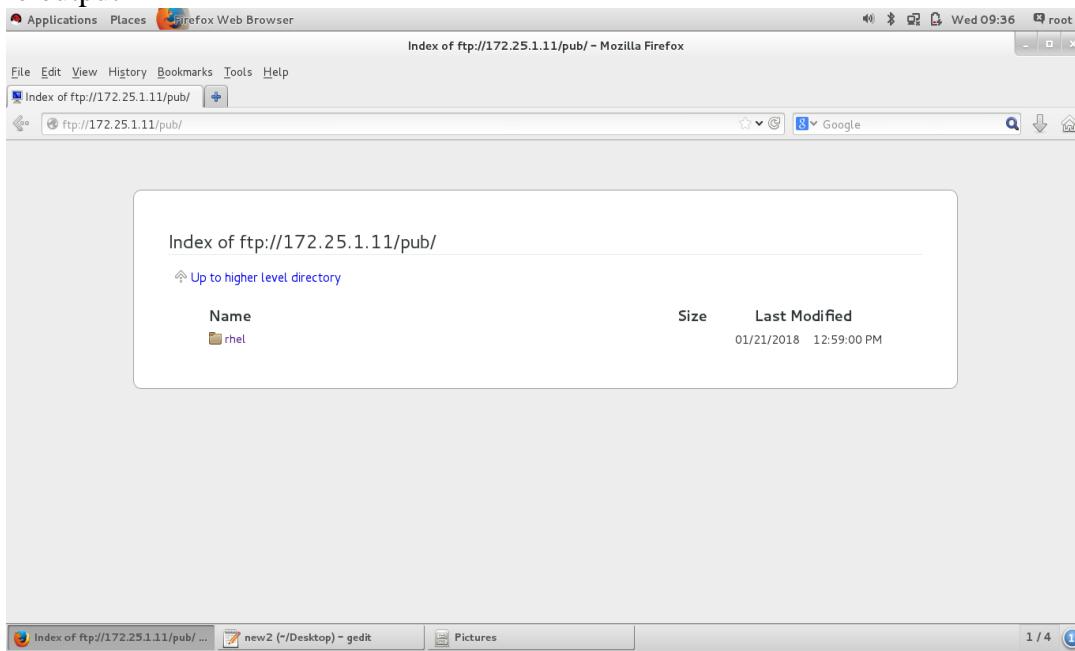
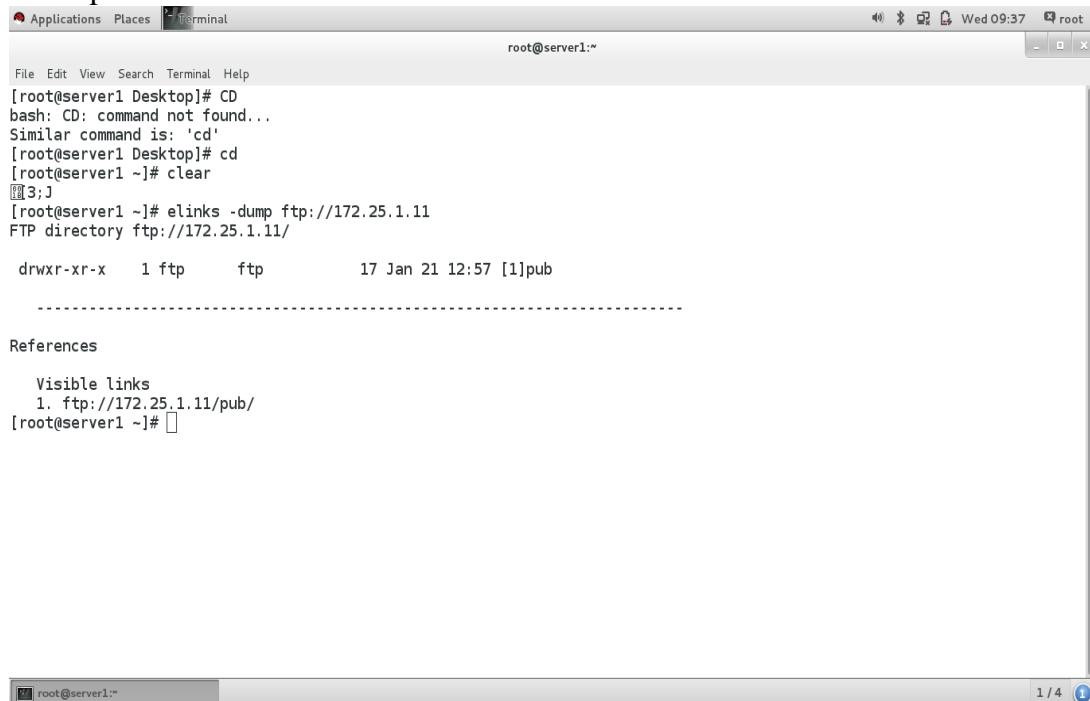


Figure 4.2: Opening browser

2: Open the terminal

Press (elinks -dump ftp://172.25.1.11)

Show the output

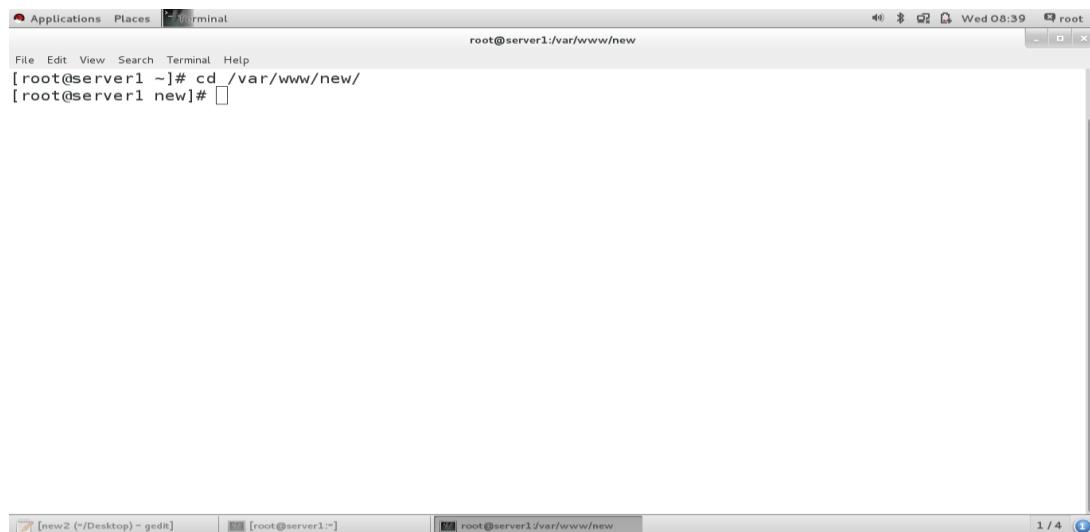


```
root@server1:~# Applications Places Terminal root@server1:~#
File Edit View Search Terminal Help
[root@server1 Desktop]# CD
bash: CD: command not found...
Similar command is: 'cd'
[root@server1 Desktop]# cd
[root@server1 ~]# clear
[3;J
[root@server1 ~]# elinks -dump ftp://172.25.1.11
FTP directory ftp://172.25.1.11/
drwxr-xr-x 1 ftp      ftp          17 Jan 21 12:57 [1]pub
-----
References
Visible links
1. ftp://172.25.1.11/pub/
[root@server1 ~]#
```

Figure 4.3: Opening Terminal

4.2 HTTP server

Input: This is the main input path of http server (cd/var/www/new/).



```
root@server1:~# Applications Places Terminal root@server1:~#
File Edit View Search Terminal Help
[root@server1 ~]# cd /var/www/new/
[root@server1 new]#
```

Figure 4.4: Input path of HTTP server

- This is main input file (vim index.html)

```
Applications Places terminal
root@server1:/var/www/new
File Edit View Search Terminal Help
[root@server1 ~]# cd /var/www/new/
[root@server1 new]# vim index.html
```

The screenshot shows a terminal window titled "terminal" running as root on a server. The command "cd /var/www/new/" has been entered and executed. The next command, "vim index.html", is being typed. The window title bar also shows "root@server1:/var/www/new". Below the terminal window, there are several other windows visible in the taskbar, including "new2 (~/Desktop) - gedit" and "root@server1:~".

Figure 4.5: Input file

Output

1: Open the browser

Press host name/ip (<http://www.mamun.com>)

Show the output

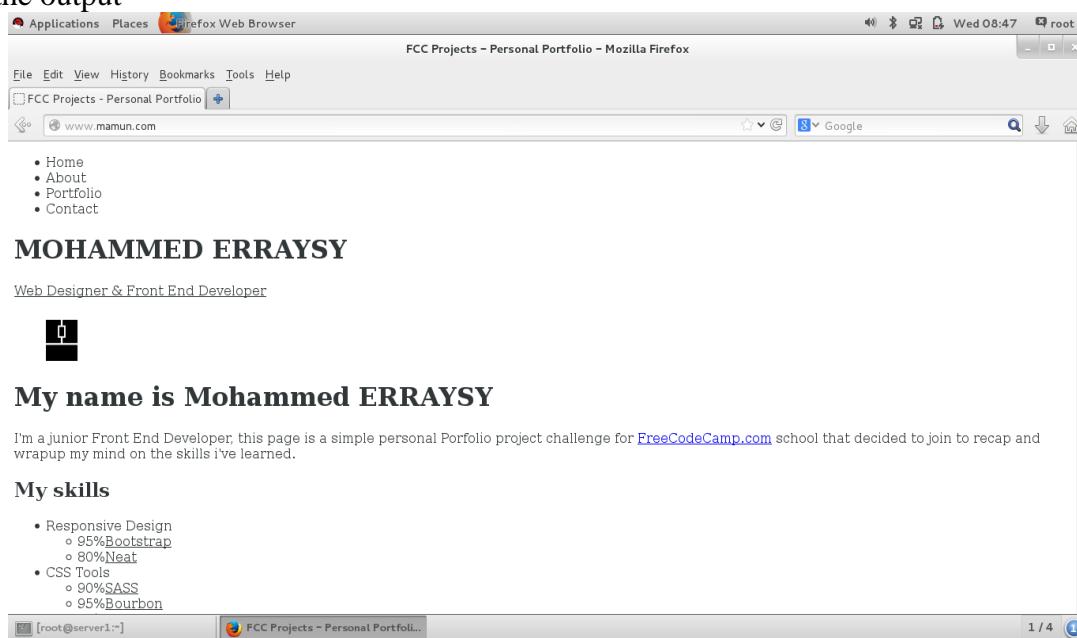
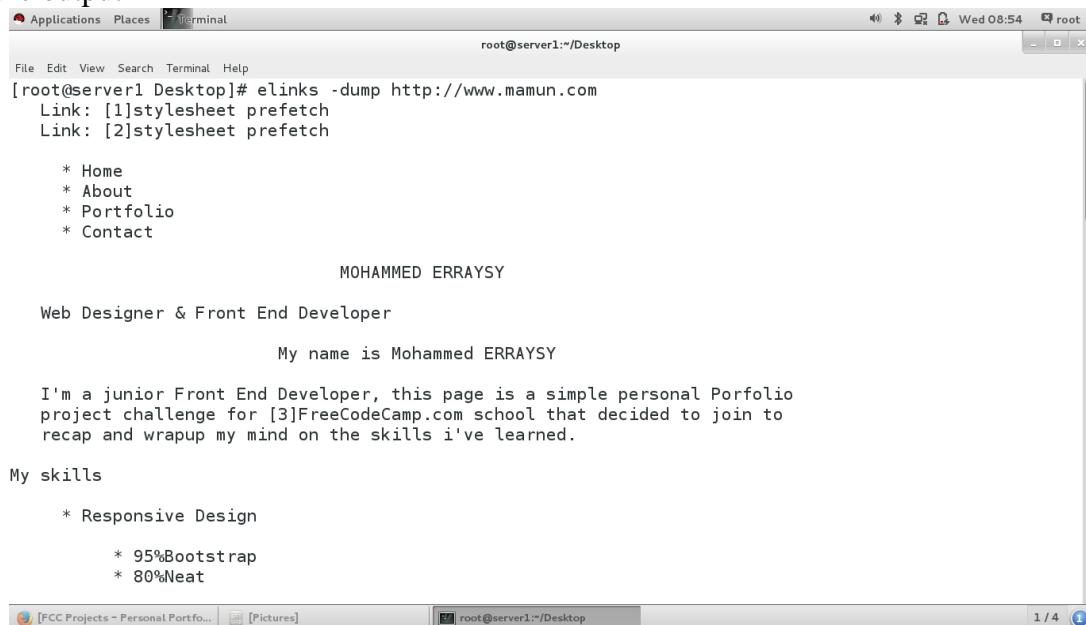


Figure 4.6: Opening browser

2: Open the terminal

Press (elinks -dump http://www.mamun.com)

Show the output



```
[root@server1 Desktop]# elinks -dump http://www.mamun.com
Link: [1]stylesheet prefetch
Link: [2]stylesheet prefetch
* Home
* About
* Portfolio
* Contact

MOHAMMED ERRASY

Web Designer & Front End Developer

My name is Mohammed ERRASY

I'm a junior Front End Developer, this page is a simple personal Porfolio
project challenge for [3]FreeCodeCamp.com school that decided to join to
recap and wrapup my mind on the skills i've learned.

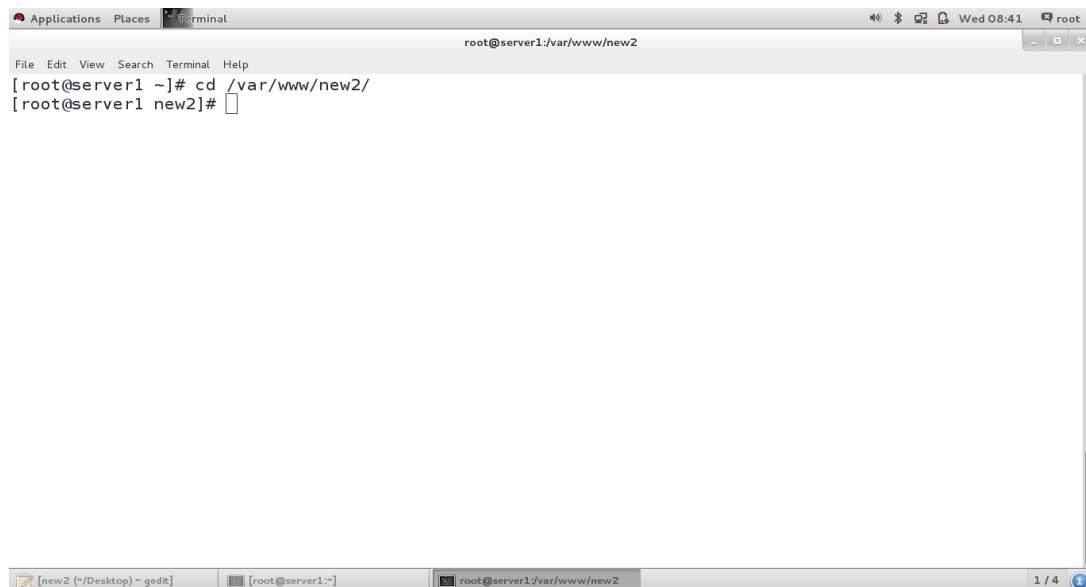
My skills

* Responsive Design
* 95%Bootstrap
* 80%Neat
```

Figure 4.7: Opening Terminal

4.3 Virtual server

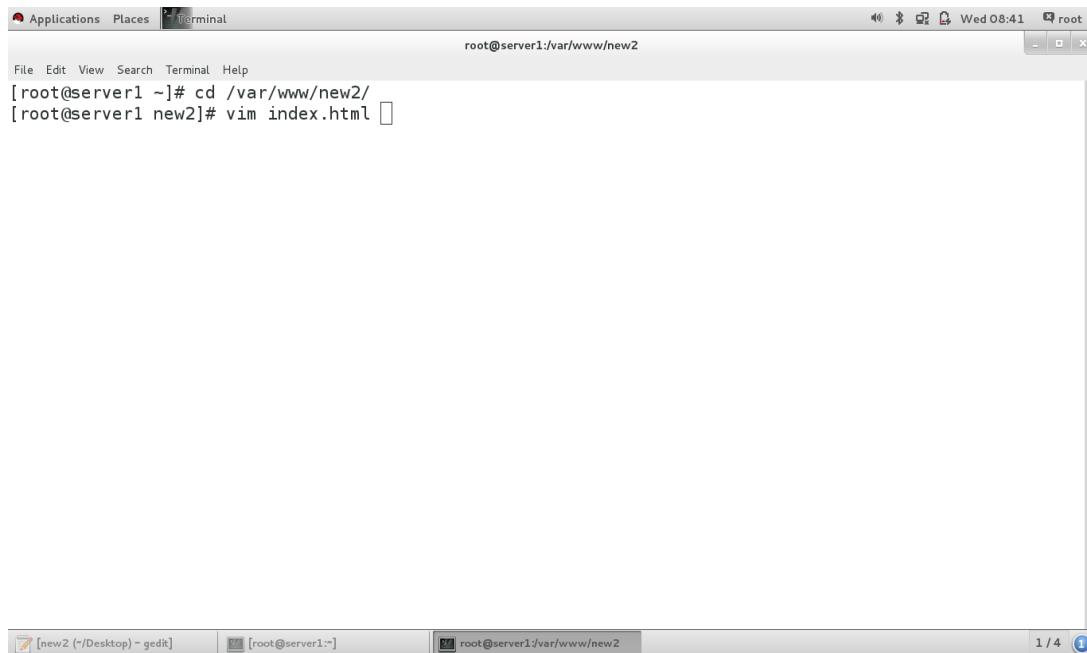
Input: This is the main input path of http server (cd /var/www/new2/)



```
[root@server1 ~]# cd /var/www/new2/
[root@server1 new2]#
```

Figure 4.8: Input path of HTTP server

- This is main input file (vim index.html)



```
Applications Places Terminal
root@server1:/var/www/new2
File Edit View Search Terminal Help
[root@server1 ~]# cd /var/www/new2/
[root@server1 new2]# vim index.html
```

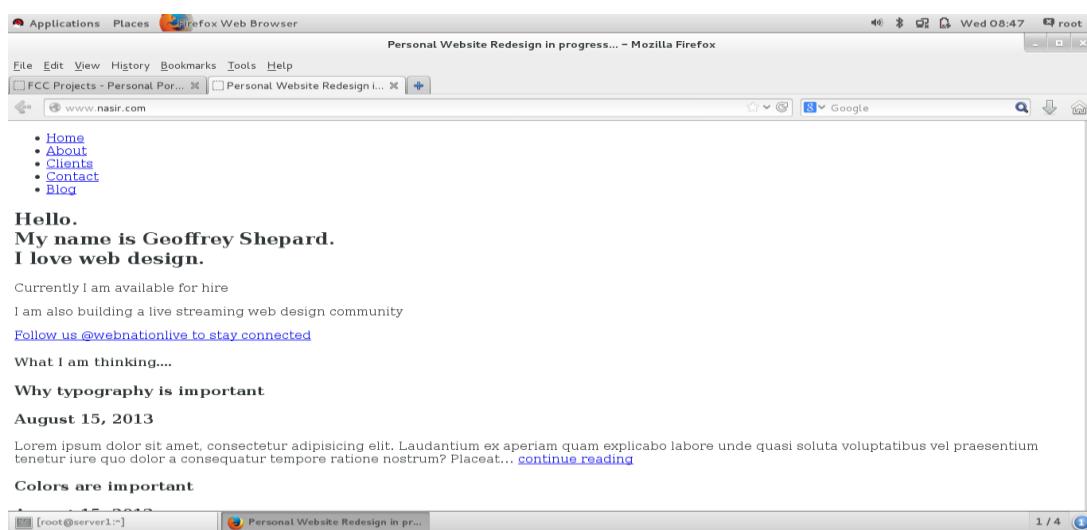
Figure 4.9: Main input file

Output

1: Open the browser.

Press host name/ip (<http://www.nasir.com>)

Show the output



Personal Website Redesign in progress... – Mozilla Firefox

File Edit View History Bookmarks Tools Help

FCC Projects - Personal Port... Personal Website Redesign i... +

www.nasir.com Google

- Home
- About
- Clients
- Contact
- Blog

Hello.
My name is Geoffrey Shepard.
I love web design.

Currently I am available for hire
I am also building a live streaming web design community
[Follow us @webnationlive to stay connected](#)

What I am thinking....

Why typography is important

August 15, 2013

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Laudantium ex aperiam quam explicabo labore unde quasi soluta voluptatibus vel praesentium tenetur iure quo dolor a consequatur tempore ratione nostrum? Placeat... [continue reading](#)

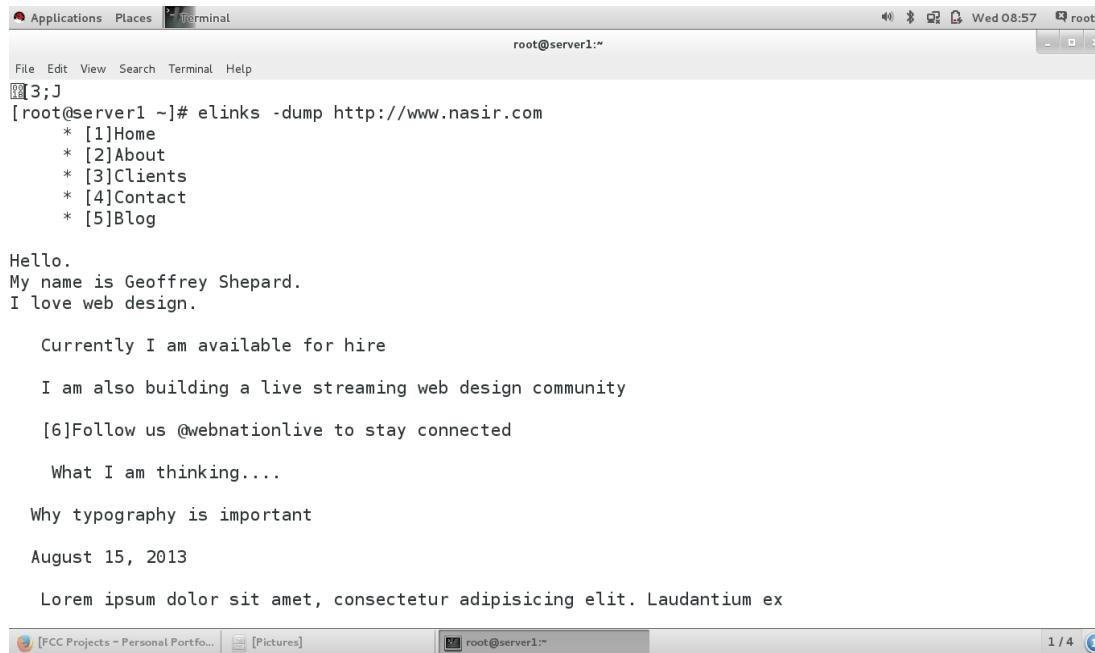
Colors are important

Figure 4.10: Opening browser

2: Open the terminal

Press (elinks -dump http://www.nasir.com)

Show the output



```
root@server1:~# elinks -dump http://www.nasir.com
[3;J
[1]Home
* [2]About
* [3]Clients
* [4]Contact
* [5]Blog

Hello.
My name is Geoffrey Shepard.
I love web design.

Currently I am available for hire

I am also building a live streaming web design community

[6]Follow us @webnationlive to stay connected

What I am thinking.....

Why typography is important

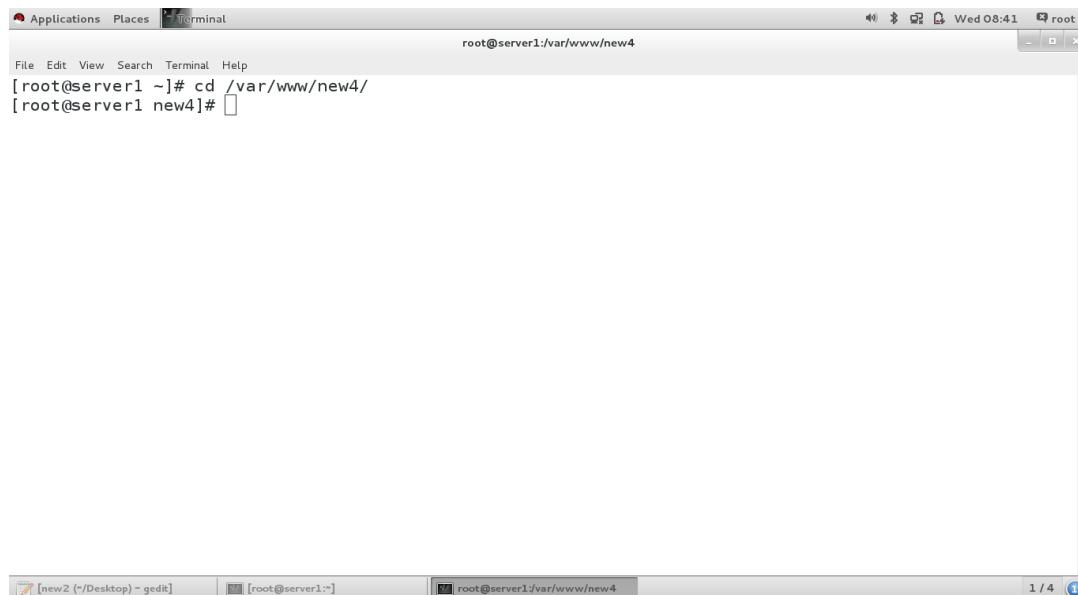
August 15, 2013

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Laudantium ex
```

Figure 4.11: Opening Terminal

4.4 Process of restricted

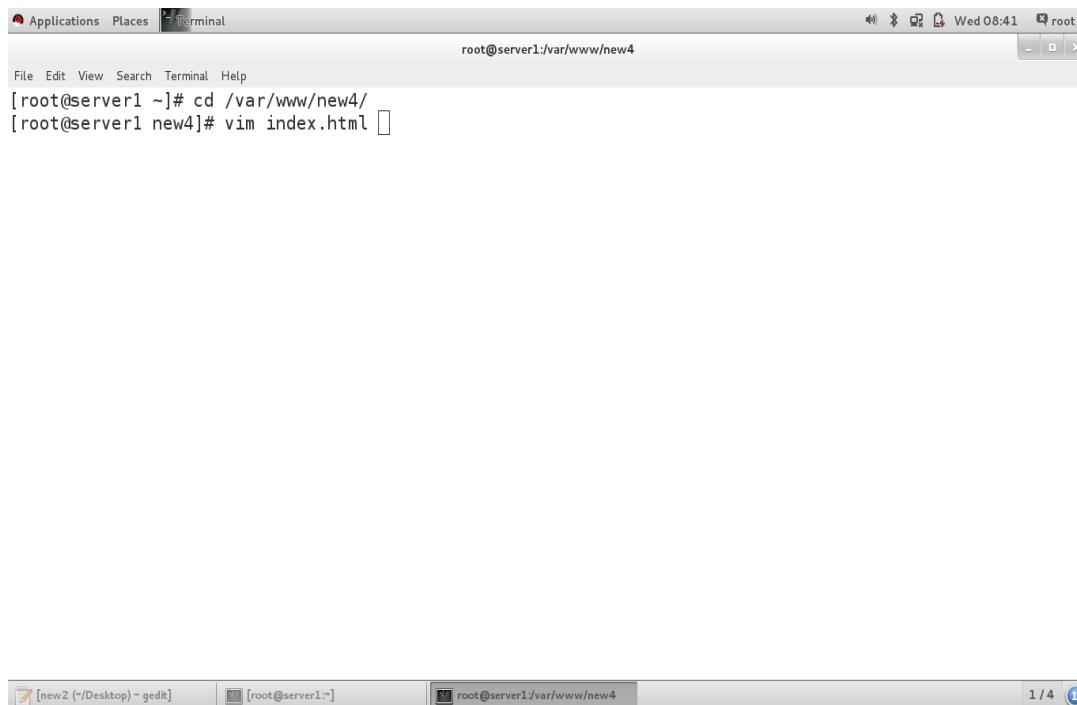
Input: This is the main input path of http server (cd/var/www/new4/)



```
root@server1:~# cd /var/www/new4/
[root@server1 new4]#
```

Figure 4.12: Input path of HTTP server

- This is the main input file (vim index.html)



```
Applications Places Terminal
root@server1:/var/www/new4
File Edit View Search Terminal Help
[root@server1 ~]# cd /var/www/new4/
[root@server1 new4]# vim index.html
```

The screenshot shows a terminal window titled "Terminal". The title bar includes the application menu ("Applications", "Places"), the window title ("Terminal"), and system status icons (volume, battery, network). The status bar at the bottom right shows the date and time ("Wed 08:41") and user ("root"). The main pane displays a command-line session where the user navigates to the directory "/var/www/new4/" and runs the command "vim index.html". The terminal window is part of a larger desktop environment, as evidenced by the taskbar at the bottom.

Figure 4.13: Input file

Output

1: Open the browser

Press host name/ip (wttt://www.anwar.com)

Show the output

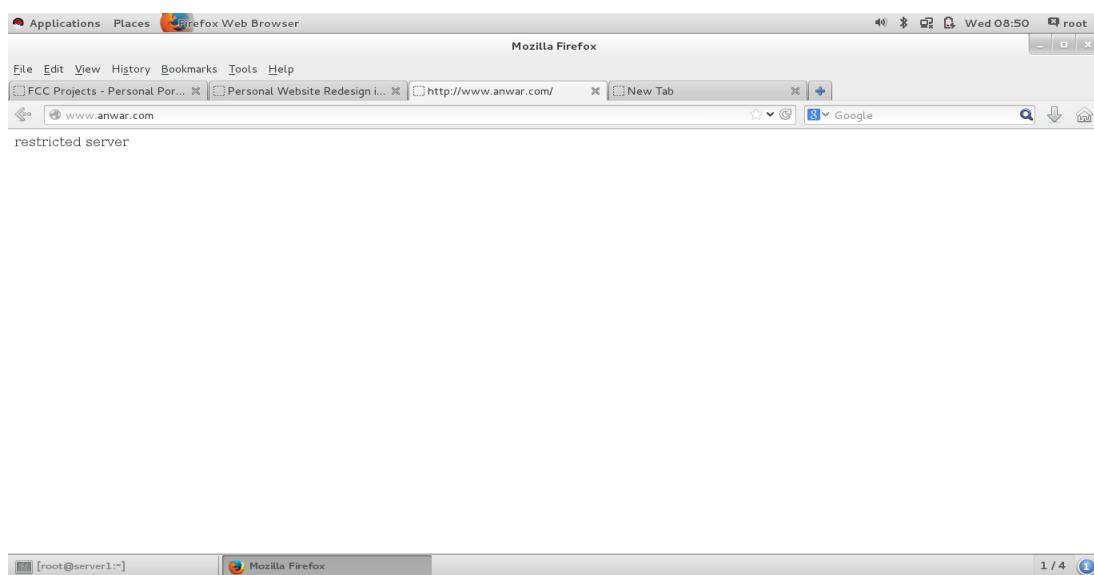


Figure 4.14: Opening browser

2: Open the terminal

Press (elinks -dump http://www.anwar.com)

Show the output

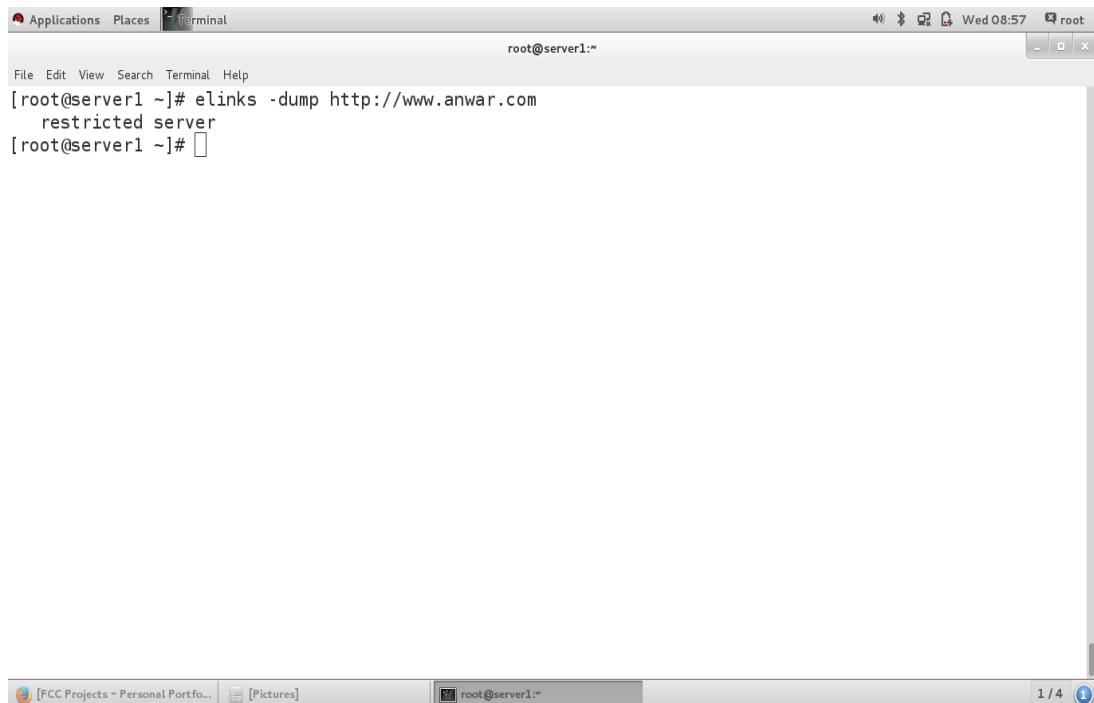


Figure 4.15: Opening Terminal

4.5 HTTPS server

Input: This is the main input path of http server (cd/var/www/new3/)

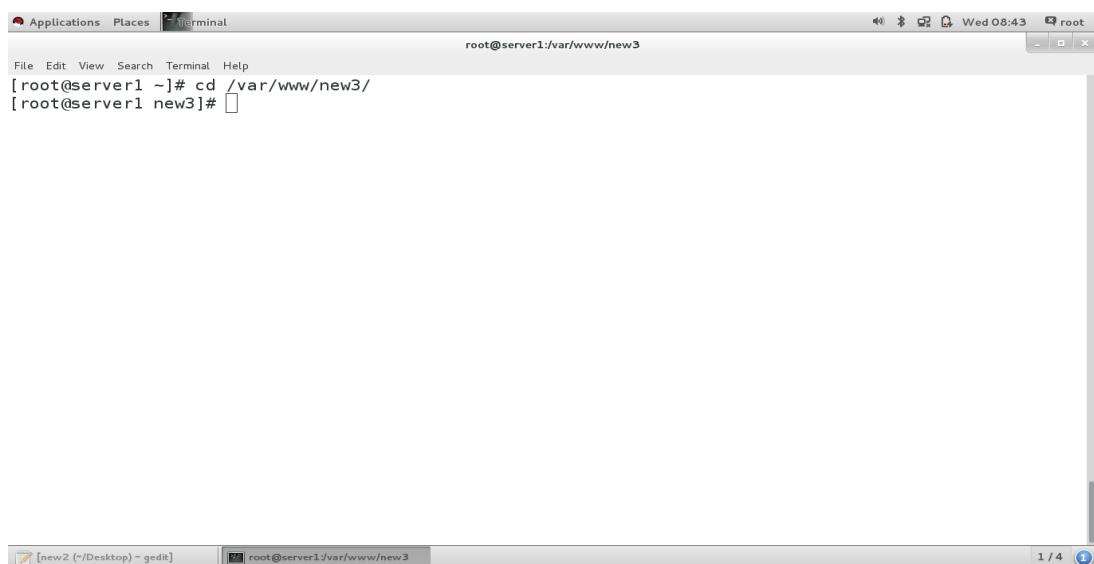
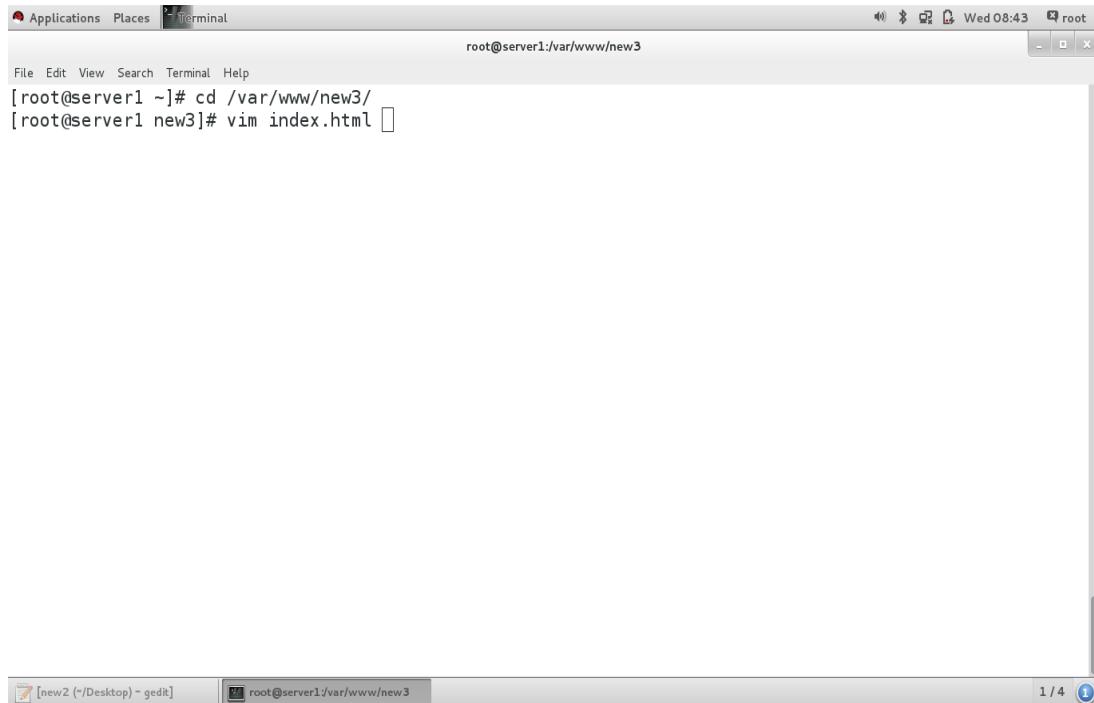


Figure 4.16: Input path of HTTP server

- This is the main input file (vim index.html)



```
Applications Places Terminal
root@server1:/var/www/new3
File Edit View Search Terminal Help
[root@server1 ~]# cd /var/www/new3/
[root@server1 new3]# vim index.html
```

The screenshot shows a terminal window titled "Terminal". The title bar includes the application menu ("Applications", "Places"), the window title ("Terminal"), and system status ("Wed 08:43"). The terminal window displays a command-line session where the user is navigating to the directory "/var/www/new3/" and opening the file "index.html" with the Vim editor. The status bar at the bottom shows the current file path and the command entered.

Figure 4.17: Main input file

Output

- 1: Open the browser
Press host name/ip (<https://server1.example.com>)
Show the output

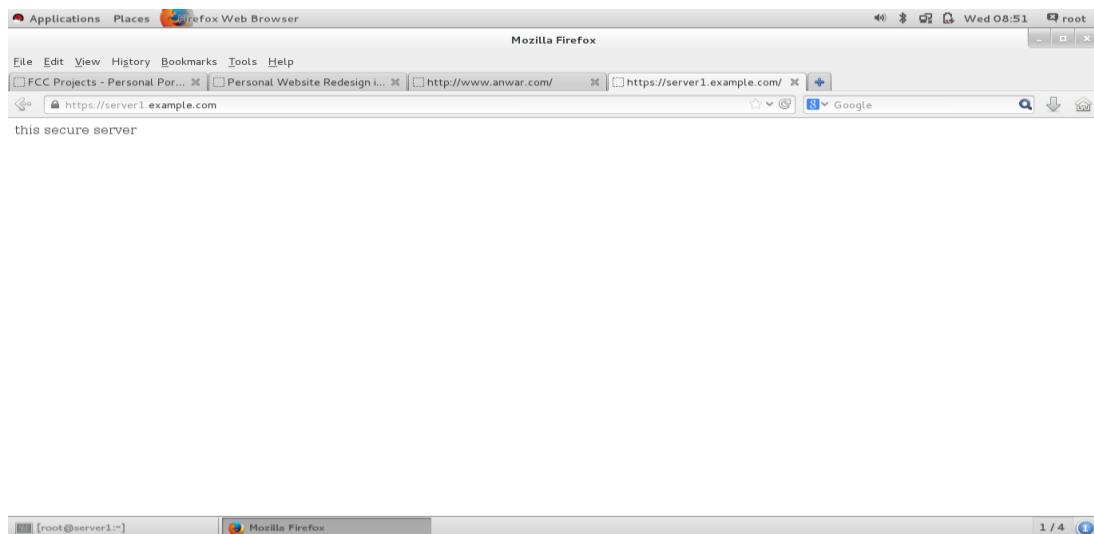


Figure 4.18: Opening browser

2: Open the terminal

Press (elinks -dump https://server1.example.com)

Show the output

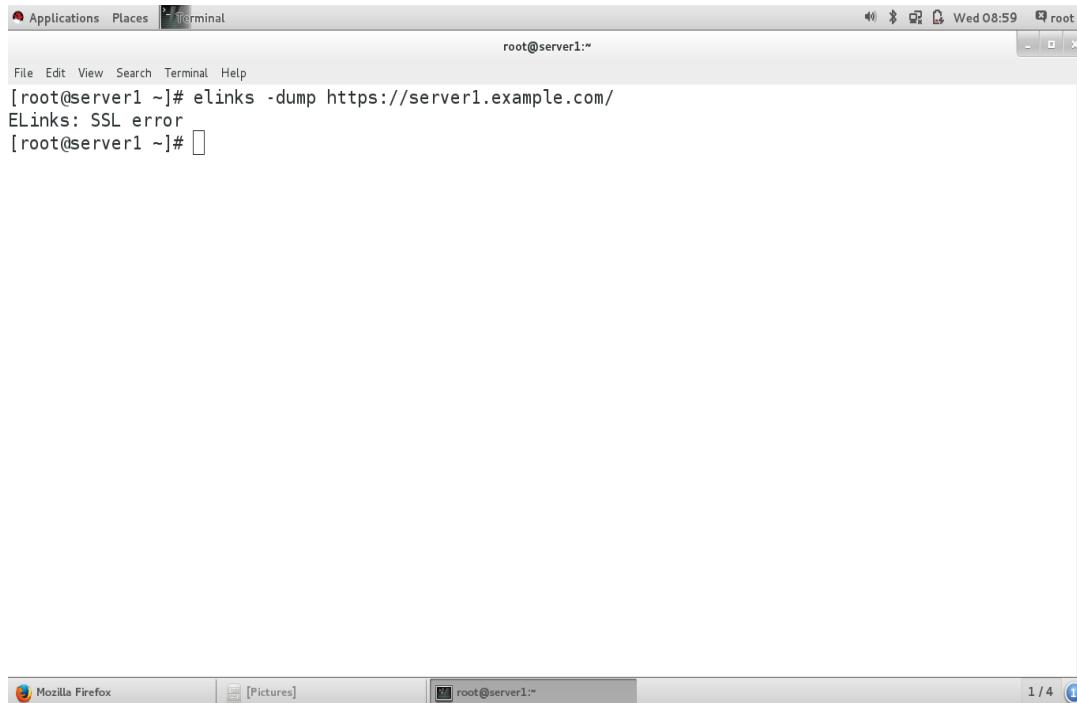


Figure 4.19: Opening Terminal

Summary

HTTP, Virtual, HTTPS are given the input for his main path and showing the output in this server.

Chapter 5

Conclusion

5.1 Outcome

As mentioned above, although our system had been completed but it is not perfect, we had planned to make some enhancement in the future. We think that our system still has potential to grow. Besides, we will include more functions and introduce more widgets to the system. We also plan to enhance the interface so that it looks more attractive and interactive. The application has been tested with live data and has provided a successful result. Hence the web server has proved to work efficiently.

- The main purposes of our work are:
- To integrate and digitalize the whole system
- To provide authorized access from all over the world
- To provide security of the system
- To control automatically the hole system of all types of organization.
- To prevent the loss of income due to poor accountability of present status an organization.
- To provide a full range of reports that will satisfy informational requirements.

5.2 Limitations

- This web server has some limitations which are mention are below.
- Students have some knowledge about the computer.
- Need a server that is expensive.
- Only admin can add any new user internally not in database directly.
- Need potential coding knowledge and maintenance personals.
- Require individual environment.
- We are unable to provide video conferencing in our project because of shortage of time.

5.3 Future Scope

- If we want to know any information about our web server anywhere it will be possible in the future.
- To ensure reliability and information availability.
- We can get any data within short time which is stored in database.
- We think that our system still has potential to grow. Besides, we will include more functions and introduce more widgets to the system.

5.4 Conclusion

This is globalizations era. Everything becoming automatically day by day. So in this moment our multiple web servers is meaningful with these globalizations. This multiple web server makes our work easy. Our multiple web servers can save our time. Its maintain admin part that is very important in our present time.

BIBLIOGRAPHY

1. <http://www.c-sharpcorner.com/article/introduction-to-linux->
2. https://en.wikipedia.org/wiki/File_Transfer_Protocol
3. red hat system administrator 1 student workbook
4. https://en.wikipedia.org/wiki/Web_server
5. https://www.webopedia.com/TERM/V/virtual_server.html
6. <https://en.wikipedia.org/wiki/HTTPS>
7. https://www.youtube.com/watch?v=sg_qvtp9WMc

Troubleshooting

Step1

- Press (cd /var/ftp/pub) then copy another file in this path, if missing any file then do not work YUM server.

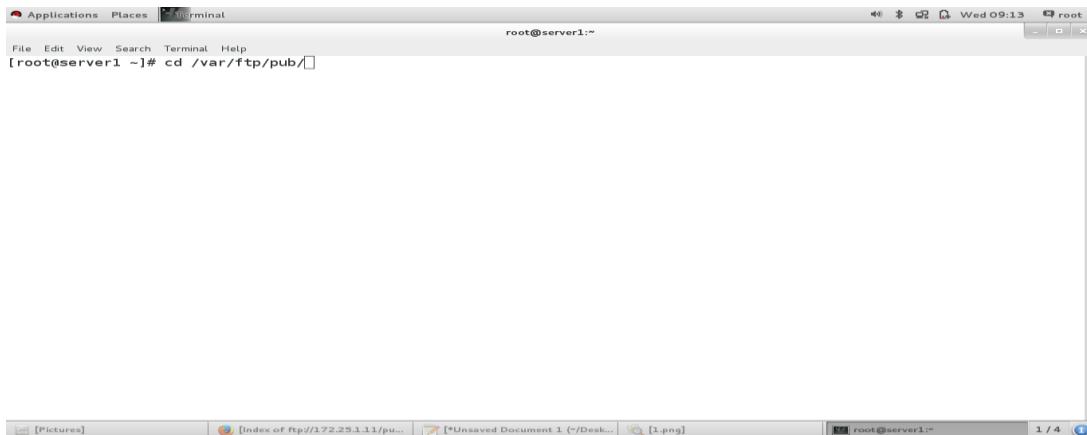


Figure 5.1: Copy another file

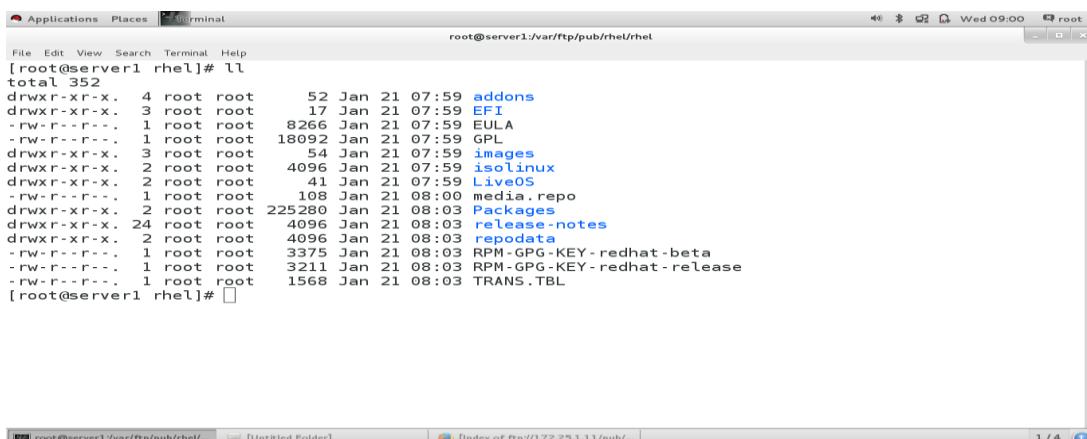


Figure 5.2: Show the file

Step 2

- Press (cd /etc/yum.repos.d/)to enter to enter vim server.repo.
- Now create a file in this path as server.repo name.
- And create appropriate baseurl. If missing any file they do not work properly.

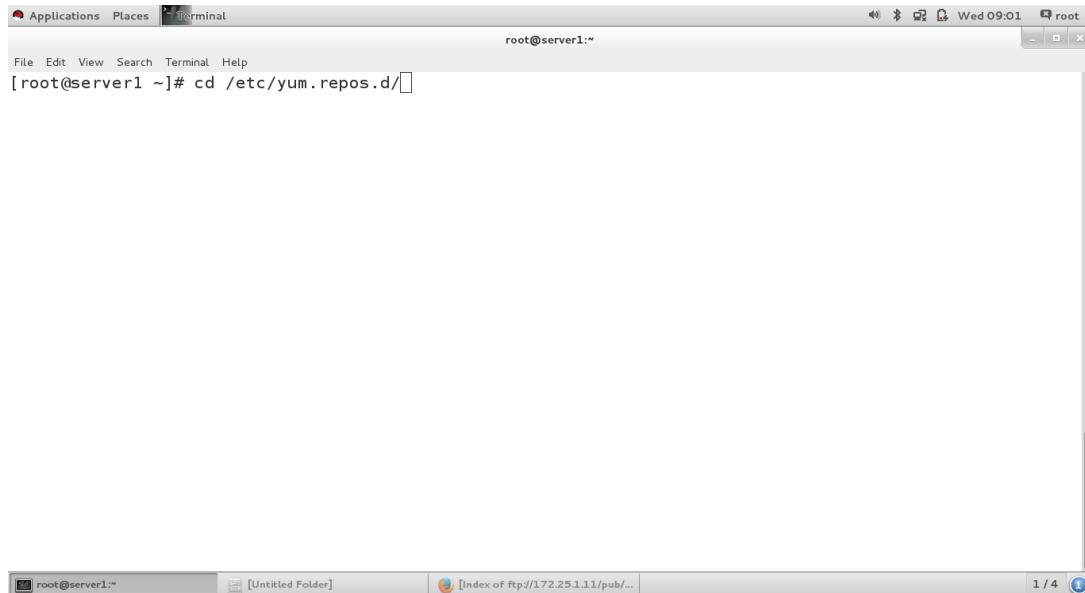


Figure 5.3: Create server.repo

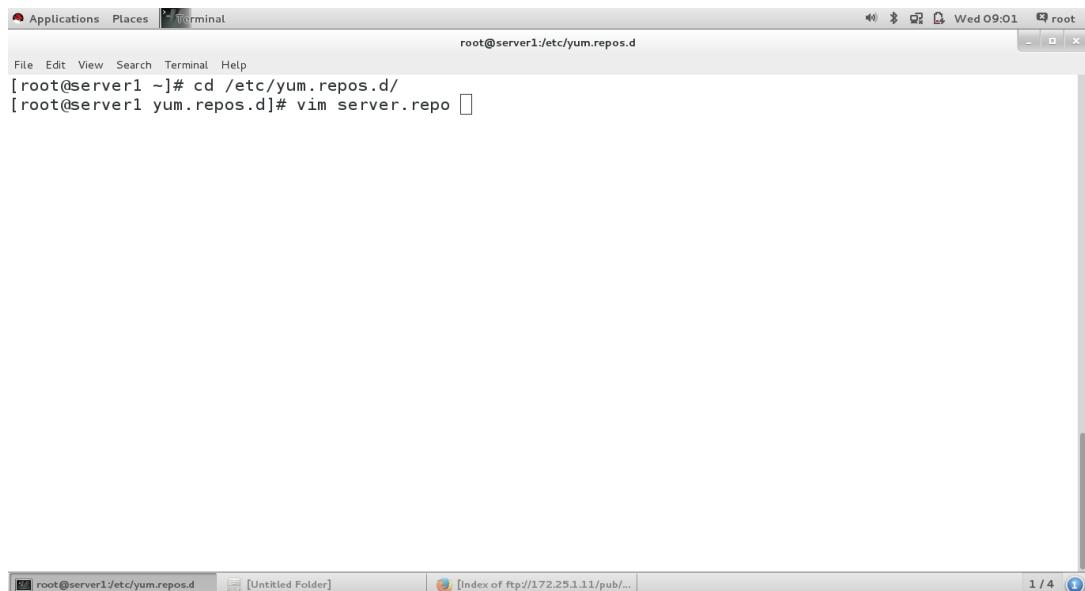


Figure 5.4: Create server.repo

The screenshot shows a terminal window titled 'Terminal' with the command 'root@server1:/etc/yum.repos.d'. The file 'server.repo' is being edited. The content of the file is:

```
[server]
name=my repo
baseurl=ftp://172.25.1.11/pub/rhel/rhel
gpgcheck=0
enable=1
```

The status bar at the bottom right indicates '3,1' lines and 'All' changes. The bottom of the terminal shows the command 'server.repo' with statistics: 6L, 83C.

Figure 5.5: Create server.repo

Step 3

- Now create a file as name of index.html.
Note: -File name index.html is compulsory,

The screenshot shows a terminal window titled 'Terminal' with the command 'root@server1:~'. The user runs the command 'vim /var/www/new3/index.html'. The status bar at the bottom right indicates '1 / 4' pages.

Figure 5.6: Index.html

Step 4

- Now open this path (**vim/etc/httpd/conf.d/ssl.conf**) then install all packages properly. If missing any package then server do not run.

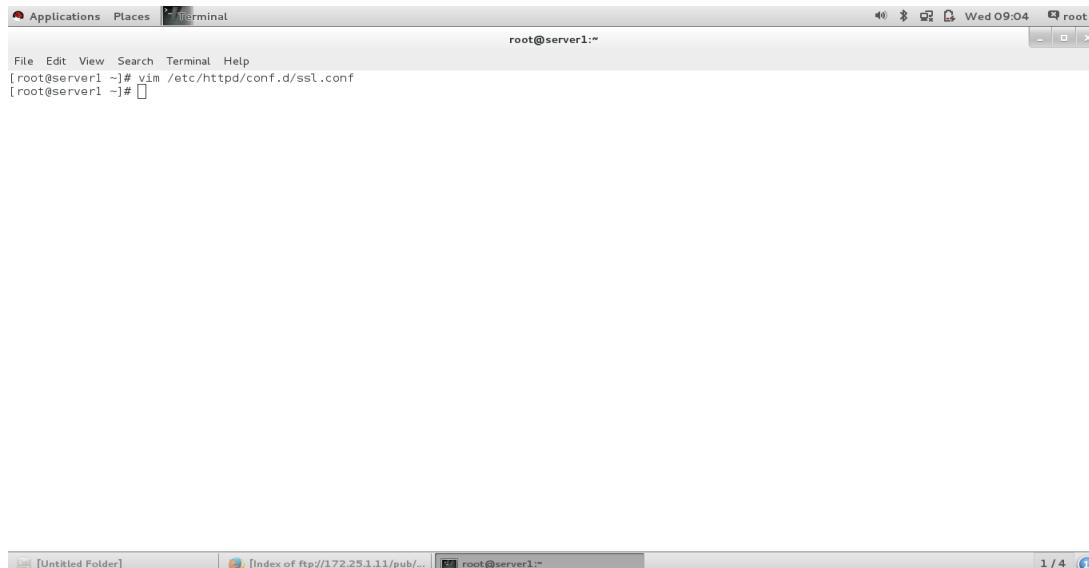


Figure 5.7: Opening path

- Configure the SSL engine on.

```
# This forces an accurate shutdown when the connection is closed, i.e. a
# SSL close notify alert is send and mod_ssl waits for the close notify
# alert of the client. This is 100% SSL/TLS standard compliant, but in
# practice often causes hanging connections with brain-dead browsers. Use
# this only for browsers where you know that their SSL implementation
# works correctly.
# Notice: Most problems of broken clients are also related to the HTTP
# keep-alive facility, so you usually additionally want to disable
# keep-alive for those clients, too. Use variable "nokeepalive" for this.
# Similarly, one has to force some clients to use HTTP/1.0 to workaround
# their broken HTTP/1.1 implementation. Use variables "downgrade-1.0" and
# "force-response-1.0" for this.
BrowserMatch "MSIE [2-5]" \
    nokeepalive ssl-unclean-shutdown \
    downgrade-1.0 force-response-1.0

# Per-Server Logging:
# The home of a custom SSL log file. Use this when you want a
# compact non-error SSL logfile on a virtual host basis.
CustomLog logs/ssl_request_log \
    "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"

</VirtualHost>

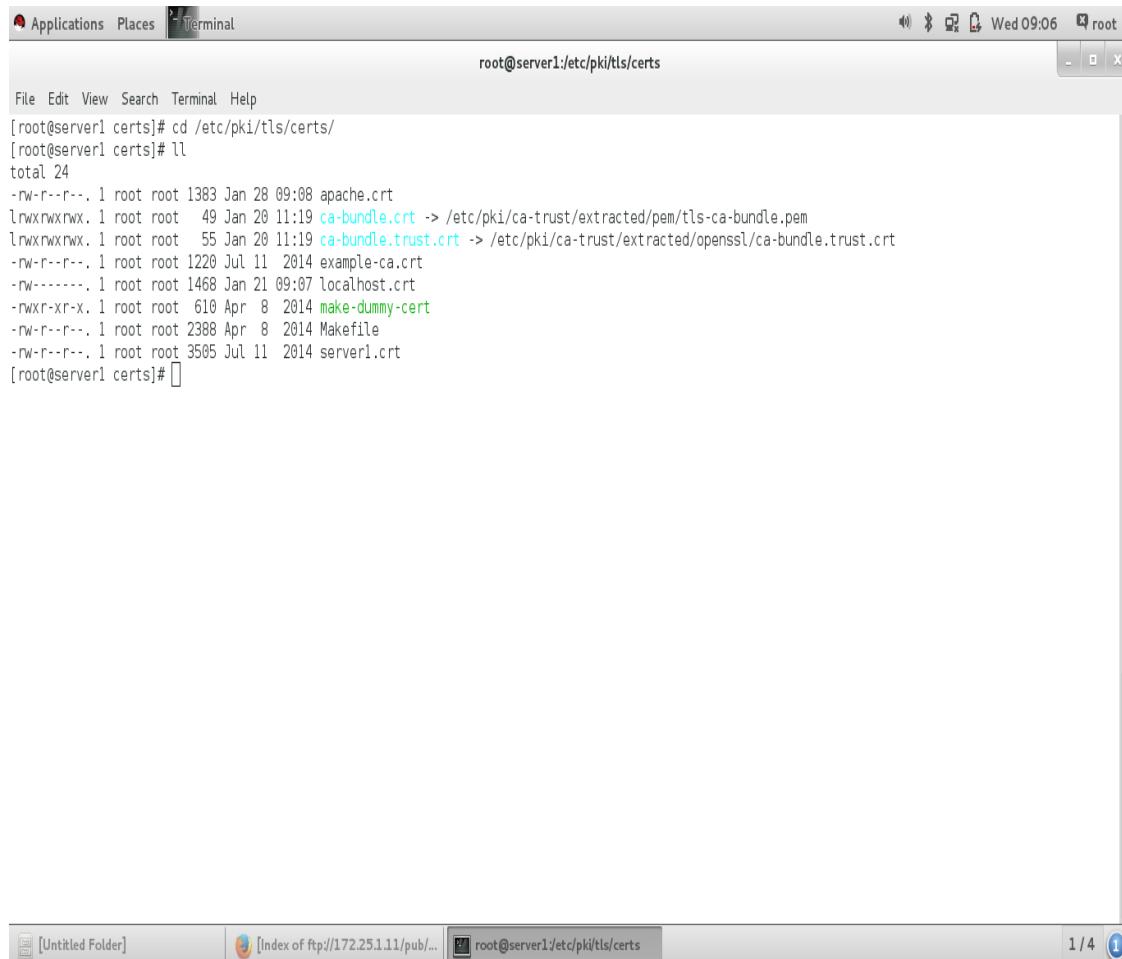
<virtualHost 172.25.1.11:443>
ServerName server1.example.com
DirectoryIndex index.html
DocumentRoot /var/www/new3
SSLEngine on
SSLCertificateFile /etc/pki/tls/certs/server1.crt
SSLCertificateKeyFile /etc/pki/tls/private/server1.key
SSLCertificateChainFile /etc/pki/tls/certs/example-ca.crt
</VirtualHost>
"/etc/httpd/conf.d/ssl.conf" 228L, 9790C
```

The terminal window shows the configuration of the SSL engine. It includes directives for SSL shutdown behavior, logging, and a specific virtual host configuration for port 443. The configuration file is '/etc/httpd/conf.d/ssl.conf'. The terminal window also shows the file size (228L) and modification time (9790C).

Figure 5.8: SSL engine on

Step 5

➤ Press (`cd /etc/pki/tls/certs/`) then all file copied here.

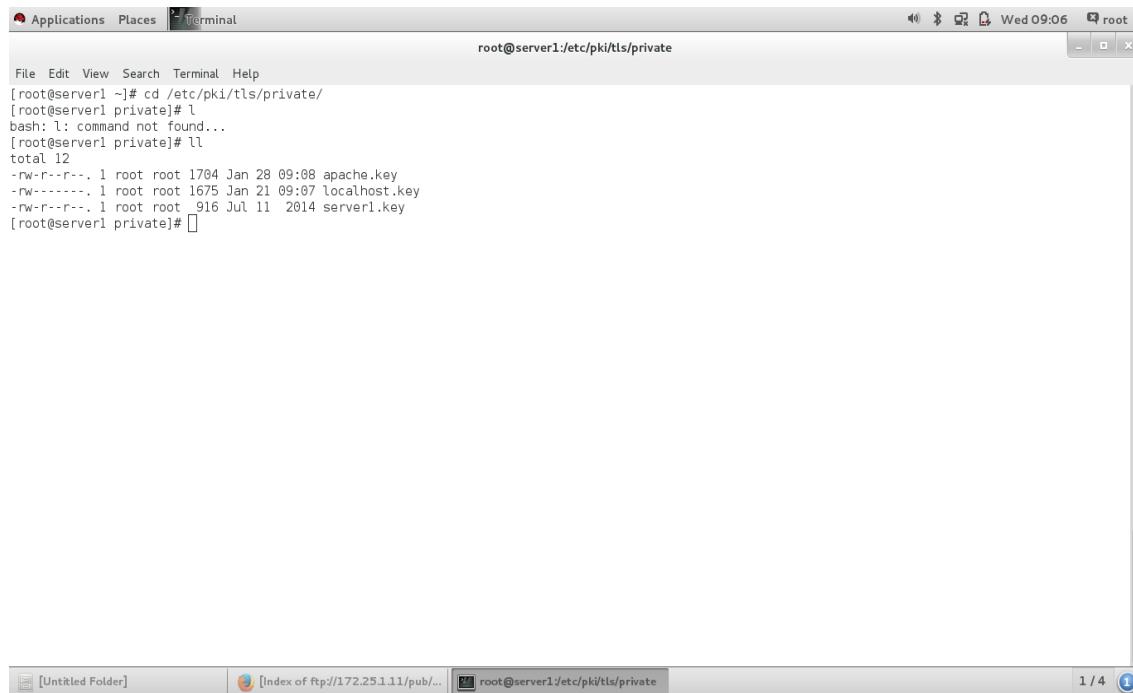


The screenshot shows a terminal window titled "Terminal" running as root on a server. The command `cd /etc/pki/tls/certs/` was entered, followed by `ll` to list the files. The output shows several certificate files:

```
File Edit View Search Terminal Help
[root@server1 certs]# cd /etc/pki/tls/certs/
[root@server1 certs]# ll
total 24
-rw-r--r--. 1 root root 1383 Jan 28 09:08 apache.crt
lrwxrwxrwx. 1 root root   49 Jan 20 11:19 ca-bundle.crt -> /etc/pki/ca-trust/extracted/pem/tls-ca-bundle.pem
lrwxrwxrwx. 1 root root   55 Jan 20 11:19 ca-bundle.trust.crt -> /etc/pki/ca-trust/extracted/openssl/ca-bundle.trust.crt
-rw-r--r--. 1 root root 1220 Jul 11 2014 example.ca.crt
-rw-----. 1 root root 1468 Jan 21 09:07 localhost.crt
-rwxr-xr-x. 1 root root  610 Apr  8 2014 make-dummy-cert
-rw-r--r--. 1 root root 2388 Apr  8 2014 Makefile
-rw-r--r--. 1 root root 3505 Jul 11 2014 server1.crt
[root@server1 certs]#
```

Figure 5.9: File copied

- Press (**cd/etc/pki/tls/private/**) then all file copied here.



The screenshot shows a terminal window with the following content:

```
File Edit View Search Terminal Help
[root@server1 ~]# cd /etc/pki/tls/private/
[root@server1 private]# l
bash: l: command not found...
[root@server1 private]# ll
total 12
-rw-r--r--, 1 root root 1704 Jan 28 09:08 apache.key
-rw-----, 1 root root 1675 Jan 21 09:07 localhost.key
-rw-r--r--, 1 root root  916 Jul 11 2014 server1.key
[root@server1 private]# 
```

Figure 5.10: File copied

End