Hands-on Lab Description

ThoTh Lab

CS-SYS-00003 – Basic Web Service (Apache) Setup on Linux

CONTENTS

1 Task 1 Install Apache2 4
2 Task 2 Setup Apache2 4
3 Related Information and Resource 6

Category:

```
CS-SYS: Computer System
```

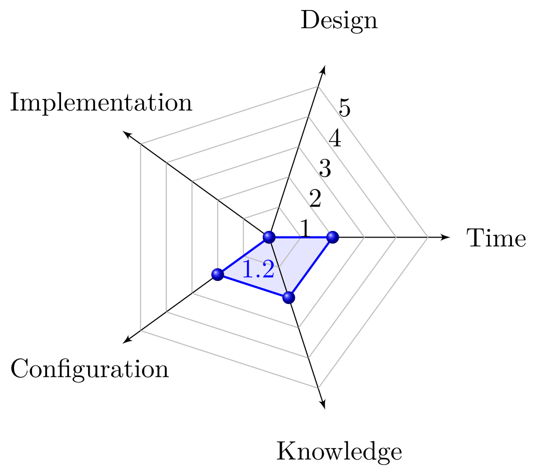
Objectives:

```
Learn Web service basis
Learn how to set up Apache2 Web server with virtual hosts
```

Estimated Lab Duration:

```
Expert: 20 minutes
Novice: 100 minutes
```

Difficulty Diagram:



Difficulty Table.

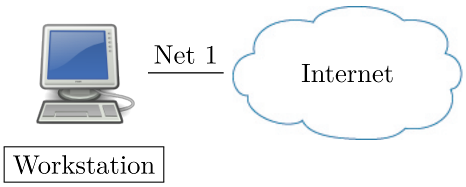
Measurements	Values (0-5)
Time	2
Design	0
Implementation	0
Configuration	2
Knowledge	2
Score (Average)	1.2

Required OS:

```
Linux: Ubuntu 18.04 LTS
```

Lab Running Environment:

```
ThoTh Lab: https://thothlab.org
```



```
Workstation: Linux (Ubuntu 18.04 LTS)
Network Setup:
Internet is connected through Net 1: 192.168.0.0/24
```

Lab Preparations:

```
  Initial setup: basic Ubuntu 18.04 LTS is required for this lab
  Basic Linux knowledge and operations. Reference Lab: CS-SYS-00001.
```

A Web server is a software responsible for accepting HTTP requests from clients, which are known as Web browsers, and serving them HTTP responses along with optional data contents, which usually are Web pages such as HTML documents and linked objects (images, video, etc.).

Apache is the most commonly used Web server on Linux systems. Web service clients typically request and view Web pages using Web browser applications such as Chrome, Firefox, Opera, Chromium, or Internet Explorer, etc. Users enter a Uniform Resource Locator (URL). For example, to view the home page of the Arizona State University's Web site, a user will enter only the URL in the address bar in his/her Web browser:

```
www.asu.edu
```

The most common protocol used to transfer Web pages is the Hyper Text Transfer Protocol (HTTP). Protocols such as Hyper Text Transfer Protocol over Secure Sockets Layer (HTTPS), and File Transfer Protocol (FTP), a protocol for uploading and downloading files, are also supported. Apache Web Servers are often used in combination with the MySQL database engine, the HyperText Preprocessor (PHP) scripting language, and other popular scripting languages such as Python and Perl. This configuration is usually termed as LAMP (Linux, Apache, MySQL and Perl/Python/PHP) and forms a powerful and robust platform for the development and deployment of Web-based applications.

1 Task 1 Install Apache2

The Apache2 web server is available in Ubuntu Linux. To determine if Apach2 is installed on Ubuntu, you can issue the following command:

```
$    apache2 -v
```

If not, you can install Apache2. At a terminal prompt, you can enter the following command:

```
$    apt install apache2
```

2 Task 2 Setup Apache2

Apache2 is configured by placing directives in plain text configuration files. Configuration files of Apache2 are installed in the folder */etc/apache2/* by default. The following directives are separated between the files and directories:

```
apache2.conf: the main Apache2 configuration file. Contains settings that are global
    to Apache2.
httpd.conf (may not exist): historically the main Apache2 configuration file, named
    after the httpd daemon. Now the file does not exist. In older versions of Ubuntu
    the file might be present, but empty, as all configuration options have been moved
     to the following referenced directories.
conf-available: this directory contains available configuration files. All files that
     were previously in /etc/apache2/conf.d should be moved to /etc/apache2/conf-
    available.
conf-enabled: holds symlinks to the files in /etc/apache2/conf-available. When a
    configuration file is symlinked, it will be enabled the next time apache2 is
    restarted.
envvars: file where Apache2 environment variables are set.
```

```
mods-available: this directory contains configuration files to both load modules and configure them. However, not all modules will have specific configuration files.
mods-enabled: holds symlinks to the files in /etc/apache2/mods-available. When a module configuration file is symlinked it will be enabled the next time apache2 is restarted.
ports.conf: houses the directives that determine which TCP ports Apache2 is listening on.
sites-available: this directory has configuration files for Apache2 Virtual Hosts. Virtual Hosts allow Apache2 to be configured for multiple sites that have separate configurations.
sites-enabled: like mods-enabled, sites-enabled contains symlinks to the /etc/apache2/sites-available directory. Similarly when a configuration file in sites-available is symlinked, the site configured by it will be active once Apache2 is restarted.
magic: instructions for determining MIME type based on the first few bytes of a file.
```

Apache2 ships with a virtual-host-friendly default configuration. That is, it is configured with a single default virtual host (using the VirtualHost directive), which can be modified or used as-is if you have a single site, or used as a template for additional virtual hosts if you have multiple sites. If left alone, the default virtual host will serve as your default site, or the site users will see if the URL they enter does not match the *ServerName* directive of any of your custom sites. To modify the default virtual host, edit the file */etc/apache2/sites-available/000-default.conf*. The directives set for a virtual host only apply to that particular virtual host. If you wish to configure a new virtual host or site, copy that file into the same directory with a name you choose. For example:

```
$    cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/test-and-demo.conf
```

To set up a simple website such as "Hello World" on your website. You will not need to make changes of may default setting. Most the of the following presented configurations are informational, and you can edit these new files to configure the new site using some of the directives described below:

1. The *ServerAdmin* directive specifies the email address to be advertised for the server's administrator. The default value is *webmaster@localhost*. This should be changed to an email address that is delivered to you (if you are the server's administrator); e.g., *dijiang@asu.edu*. If your website has a problem, Apache2 will display an error message containing this email address to report the problem to. Find this directive in your site's configuration file in */etc/apache2/sites-available*.

2. The *Listen* directive specifies the port, and optionally the IP address, Apache2 should listen on. If the IP address is not specified, Apache2 will listen on all IP addresses assigned to the machine it runs on. The default value for the Listen directive is 80. Change this to 127.0.0.1:80 to cause Apache2 to listen only on your loopback interface so that it will not be available to the Internet, to (for example) 81 to change the port that it listens on, or leave it as is for normal operation. This directive can be found and changed in its own file */etc/apache2/ports.conf*.

3. The *ServerName* directive is optional and specifies what Fully Qualified Domain Name (FQDN) your site should answer to. The default virtual host has no ServerName directive specified, so it will respond to all requests that do not match a ServerName directive in another virtual host. If you have just acquired the domain name `test-and-demo.com` and wish to host it on your Ubuntu server, the value of the ServerName directive in your virtual host configuration file should be `test-and-demo.com`. Add this directive to the new virtual host file you created earlier (/etc/apache2/sites-available/test-and-demo.conf).

4. The ServerName directive is optional and specifies what Fully Qualified Domain Name (FQDN) your site should answer to. The default virtual host has no ServerName directive specified, so it will respond to all requests that do not match a ServerName directive in another virtual host. If you have just acquired the domain name ubunturocks.com and wish to host it on your Ubuntu server, the value of the ServerName

directive in your virtual host configuration file should be ubunturocks.com. Add this directive to the new virtual host file you created earlier (such as */etc/apache2/sites-available/test-and-demo.conf*).

5. You may also want your site to respond to *www.test-and-demo.com*, since many users will assume the *www* prefix is appropriate. Use the *ServerAlias* directive for this. You may also use wildcards in the ServerAlias directive. For example, the following configuration will cause your site to respond to any domain request ending in *.test-and-demo.com*.

```
      ServerAlias *.test-and-demo.com
```

6. The *DocumentRoot* directive specifies where Apache2 should look for the files that make up the site. The default value is */var/www/html*, as specified in */etc/apache2/sites-avail:able/000-default.conf*. If desired, change this value in your site's virtual host file, such as */var/www/html/test*, and remember to create that directory if necessary!

7. Then, enable the new VirtualHost using the *a2ensite* utility and restart Apache2:

```
$       sudo a2ensite test-and-demo
$       sudo systemctl restart apache2.service % or issue: sudo systemctl reload
          apache2; or issue: sudo apache2 restart
```

8. You can also use the *a2dissite* utility to disable sites. This is can be useful when troubleshooting configuration problems with multiple VirtualHosts:

```
$       sudo a2dissite 000-default % disable the default site
$       sudo systemctl restart apache2.service % or issue sudo apache2 stop
```

To test your newly established web site, you can type *localhost* from your server's browser address bar.

3 Related Information and Resource

```
Install and configure Apache:
    https://ubuntu.com/tutorials/install-and-configure-apache#1-overview
```