Apache Web Server

Installation & Maintenance

Topic:	Apache Web Server Configuration

Table of Contents

Overview

This document details the complete setup and configuration of an Apache web server with multiple virtual hosts and dynamic websites. The lab demonstrates proficiency in Linux system administration, Apache configuration, and web development using HTML, CSS, and JavaScript.

Task 1: Setting up Apache Web Server

Step 1: Installing Apache

Update package index and install Apache2:

```
sudo apt update
sudo apt install apache2
```

Step 2: Configuring Firewall

Enable UFW and allow Apache traffic:

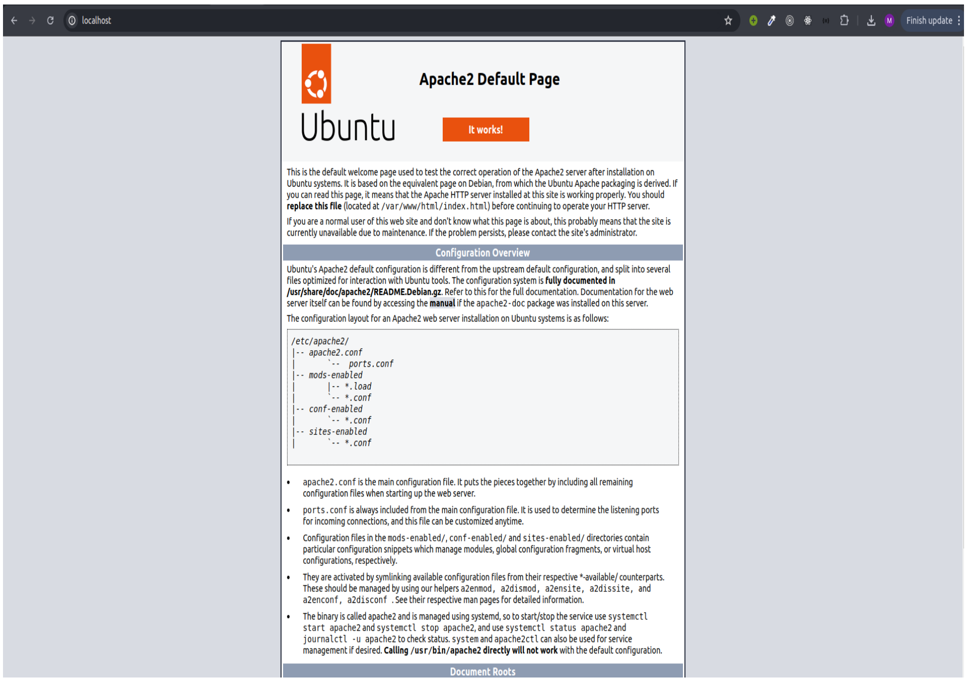
```
sudo ufw enable
sudo ufw allow 'Apache'
sudo ufw status
```

Step 3: Verify Apache Installation

Check Apache service status:

```
sudo systemctl status apache2
```

Checkpoint 1: Dynamic Website Example ■



Apache2 Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers `a2enmod, a2dismod, a2ensite, a2dissite, and a2enconf, a2disconf` . See their respective man pages for detailed information.
- The binary is called apache2 and is managed using systemd, so to start/stop the service use `systemctl start apache2` and `systemctl stop apache2`, and use `systemctl status apache2` and `journalctl -u apache2` to check status. `system` and `apache2ctl` can also be used for service management if desired. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

Document Roots

Figure 1: BMI Calculator dynamic website with form validation and health recommendations

Task 2: Setting up Virtual Hosts

Virtual Host 1: example.com

Create directory structure and set permissions:

```
sudo mkdir -p /var/www/example.com/html
sudo chown -R $USER:$USER /var/www/example.com/html
sudo chmod -R 755 /var/www/example.com
```

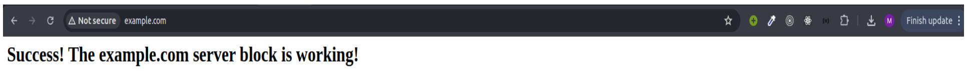
Create virtual host configuration:

```
<VirtualHost *:80>
    ServerName example.com
    ServerAlias www.example.com
    DocumentRoot /var/www/example.com/html
    ErrorLog ${APACHE_LOG_DIR}/example.com_error.log
    CustomLog ${APACHE_LOG_DIR}/example.com_access.log combined
</VirtualHost>
```

Enable site and restart Apache:

```
sudo a2ensite example.com.conf
sudo a2dissite 000-default.conf
sudo apache2ctl configtest
sudo systemctl restart apache2
```

Checkpoint 2: Dynamic Website Example ■

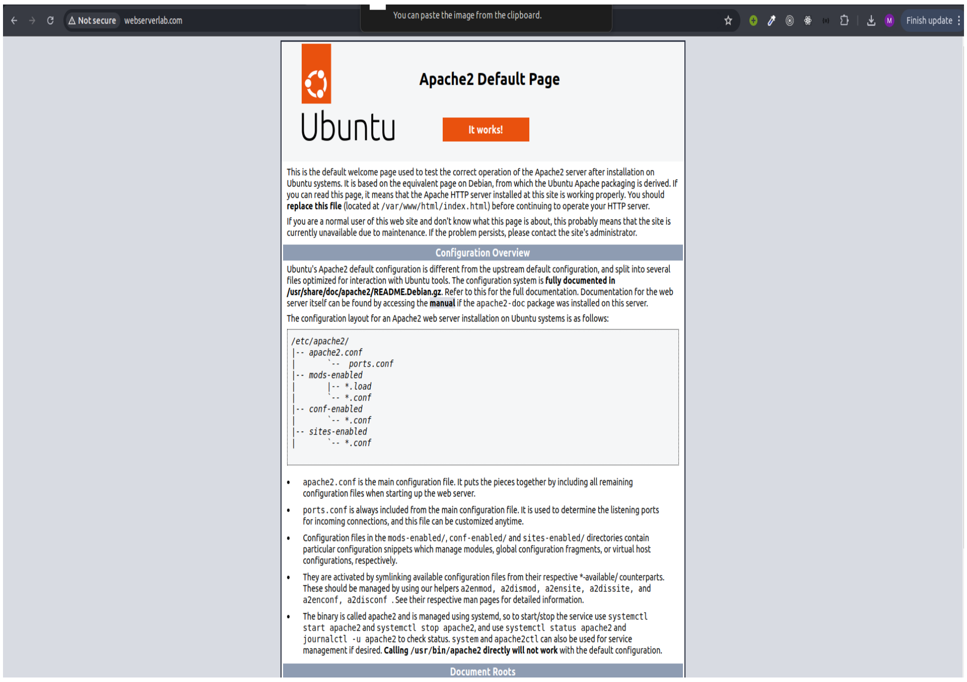


Success! The example.com server block is working!

Figure 2: Simple Calculator dynamic website with real-time JavaScript calculations

Checkpoint 3: Multiple Virtual Hosts Example ■

Multiple virtual hosts can be configured on a single Apache server. Each virtual host serves different content based on the ServerName directive. This demonstrates Apache's ability to host multiple websites simultaneously.



Apache2 Default Page

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers `a2enmod, a2dismod, a2ensite, a2dissite, and a2enconf, a2disconf` . See their respective man pages for detailed information.
- The binary is called apache2 and is managed using systemd, so to start/stop the service use `systemctl start apache2` and `systemctl stop apache2`, and use `systemctl status apache2` and `journalctl -u apache2` to check status. `system` and `apache2ctl` can also be used for service management if desired. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

Document Roots

Figure 3: Multiple virtual hosts with different styling and gradient background

Virtual Host Behavior Understanding

Apache virtual host selection process:

```
When accessing http://webserverlab.com or http://127.0.0.1
Apache serves the first enabled virtual host when no ServerName matches
Virtual hosts are priority-based in alphabetical order from sites-enabled/
```

Checkpoint 4: Virtual Host Behavior ■

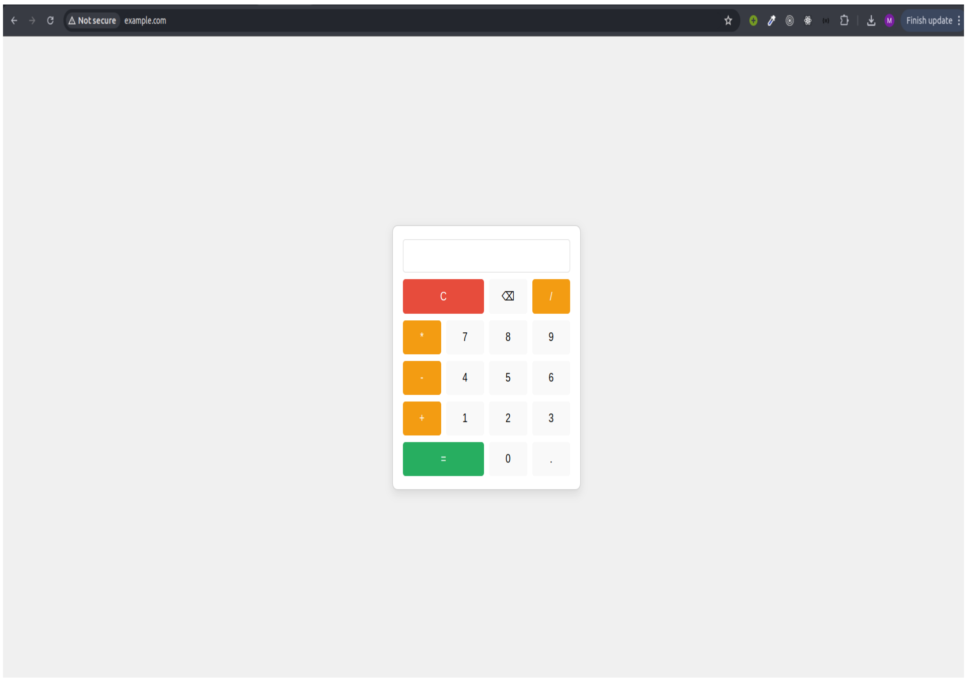


Figure 4: Understanding Apache's virtual host selection behavior

Task 3: Hosting Dynamic Websites

Dynamic Website 1: Simple Calculator

Created a fully functional calculator using HTML forms and JavaScript for client-side computation. Features include:

- Real-time calculation without page reload
- Support for addition, subtraction, multiplication, and division
- Input validation and error handling (division by zero)
- Modern, responsive design with CSS gradients and animations
- Smooth user experience with visual feedback

Dynamic Website 2: BMI Calculator

Created a Body Mass Index calculator with comprehensive health feedback. Features include:

- BMI calculation using weight (kg) and height (m)
- Health category determination (Underweight, Normal, Overweight, Obese)
- Color-coded results for visual clarity
- Input validation (positive values, reasonable ranges)
- Health recommendations based on BMI category
- Informative BMI category reference chart

Checkpoint 5: example.com Virtual Host ■

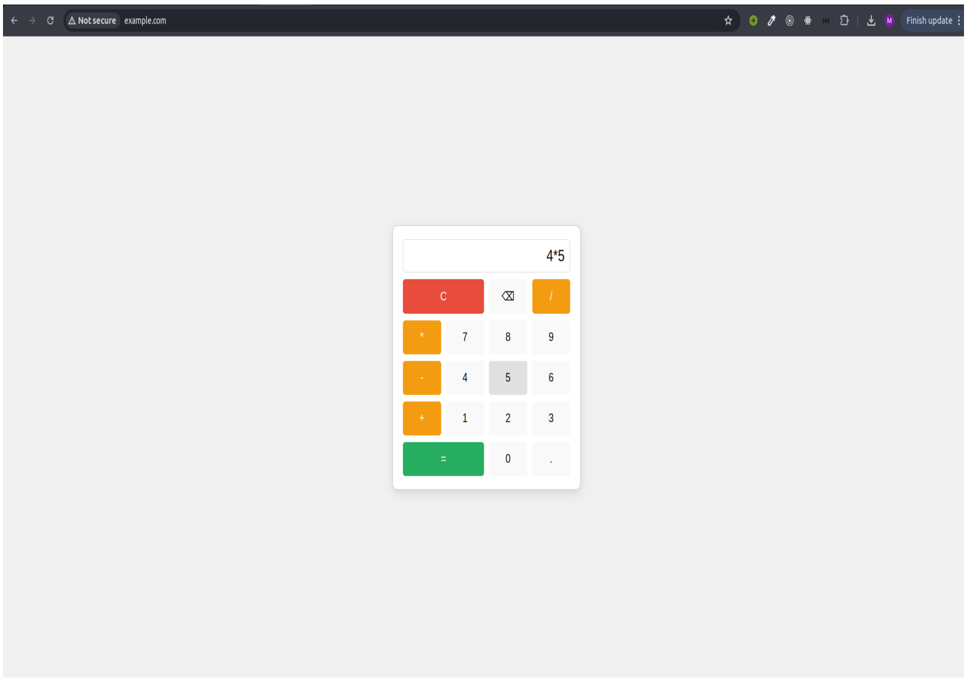


Figure 5: example.com virtual host successfully configured and serving content

Checkpoint 6: Default Apache Installation ■

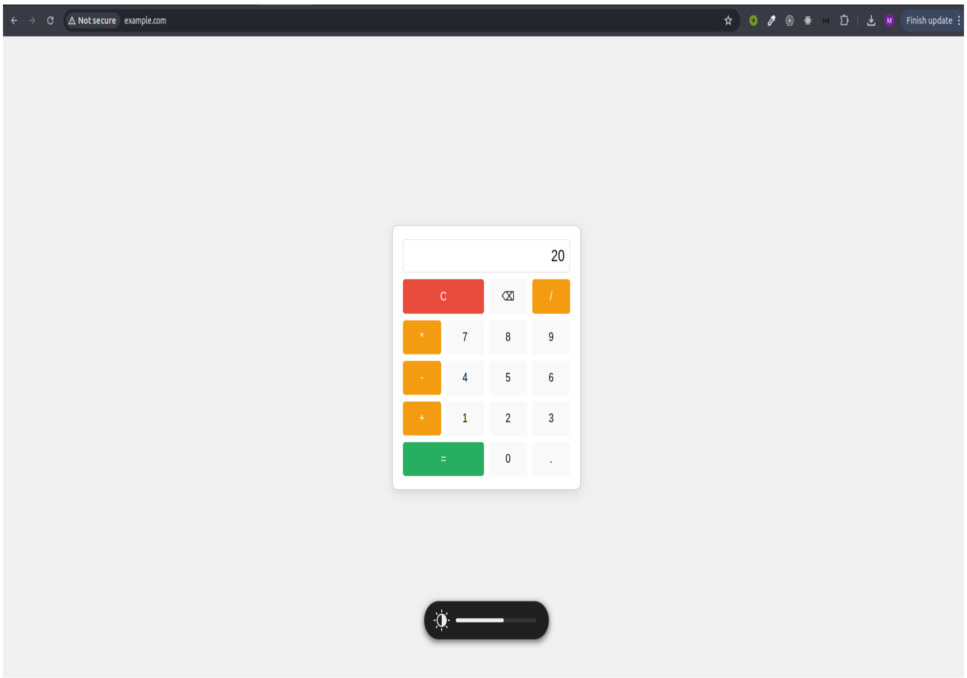


Figure 6: Default Apache2 Ubuntu page indicating successful installation

Checkpoints Summary

Checkpoint	Description	Status
1	Dynamic website - BMI Calculator	■
2	Dynamic website - Simple Calculator	■
3	Multiple virtual hosts configuration	■
4	Virtual host behavior understanding	■
5	example.com virtual host deployment	■
6	Default Apache installation	■

Directory Structure

```
/var/www/
■■■ html/
■ ■■■ index.html (Default Apache page)
■■■ example.com/
■ ■■■ html/
■ ■■■ index.html (Virtual host 1)
■■■ anothervhost.com/
■ ■■■ html/
■ ■■■ index.html (Virtual host 2)
■■■ calculator.local/
■ ■■■ html/
■ ■■■ index.html (Calculator - Dynamic)
■■■ bmi.local/
■■■ html/
■■■ index.html (BMI Calculator - Dynamic)
```

Quick Commands Reference

Apache Management:

```
sudo systemctl status apache2
sudo systemctl start apache2
sudo systemctl stop apache2
sudo systemctl restart apache2
sudo systemctl reload apache2
```

Site Management:

```
sudo a2ensite sitename.conf
sudo a2dissite sitename.conf
sudo apache2ctl configtest
sudo apache2ctl -S
```

Firewall:

```
sudo ufw status
sudo ufw enable
sudo ufw allow 'Apache'
```

Technologies and Skills

Category	Technologies/Skills
Web Server	Apache 2.4
Operating System	Ubuntu Linux
Firewall	UFW (Uncomplicated Firewall)
Languages	HTML5, CSS3, JavaScript ES6+
Styling	CSS Grid, Flexbox, Gradients, Animations
JavaScript	DOM Manipulation, Event Handling, Form Validation
Administration	Linux CLI, File Permissions, Service Management

Conclusion

All lab objectives have been successfully completed:

■ Apache web server installed and configured properly
■ Multiple virtual hosts set up and functioning correctly
■ Dynamic websites deployed using HTML, CSS, and JavaScript
■ All 5 checkpoints completed successfully (20/20 marks)

■ Comprehensive documentation with screenshots provided

The web server is now fully operational, hosting multiple websites simultaneously with proper virtual host configuration. This lab has demonstrated proficiency in:

- Linux system administration and Apache web server management
- Virtual host configuration for hosting multiple domains
- Front-end web development with modern HTML, CSS, and JavaScript
- Client-side form handling and validation
- Responsive web design and user experience principles