Lab Report: Apache Web Server, Virtual Hosts & Dynamic Website Deployment

Task-1: Apache Web Server Setup

1. Installing Apache

To install Apache on Ubuntu, the following command was executed:

sudo apt install apache2

After installation, the Apache service starts automatically and begins serving pages from `/var/www/html`.

---

2. Firewall Configuration (UFW)

What is a Firewall?

A firewall acts as a protective barrier for your system. It:

- Blocks all incoming connections unless explicitly allowed
- Protects a server from unauthorized access
- Controls which ports and applications are publicly accessible

Effect of the Command:
sudo ufw allow 'Apache'

This allows:

- **HTTP traffic (Port 80)**
- Lets users view your web pages in their browser

In My Case:

Since UFW was **inactive** on the system:

- No ports were blocked
- Apache was accessible without any UFW configuration

Still, for lab consistency, the commands used were:

sudo ufw allow 'Apache'
sudo ufw enable

---

3. Mapping Local Domains

The `/etc/hosts` file was updated:

127.0.0.1 localhost
127.0.0.1 webserverlab.com
127.0.0.1 example.com
127.0.0.1 anothervhost.com

Why This Step is Important

- Creates **fake domain names** for local use
- Allows hosting multiple sites on one machine
- Helps in testing without purchasing real domains

How It Works

Typing `http://example.com` in the browser now loads the **local Apache server**, not the internet.

---

4. Verifying Apache Installation

Tested by visiting:

- `http://localhost`
- `http://webserverlab.com`
- `http://127.0.0.1`

The default Apache page appeared successfully.

Task-2: Virtual Host Setup

Apache Virtual Hosts allow hosting multiple websites on the same server.

Step-1: Creating a Single Virtual Host (example.com)

A. Directory Setup

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

B. Creating Website Content

File: `/var/www/example.com/html/index.html`

```
<html>
<head>
    <title>Welcome to Example.com</title>
</head>
<body>
    <h1>Success! example.com virtual host is working.</h1>
</body>
</html>
```

C. Virtual Host Configuration

File: `/etc/apache2/sites-available/example.com.conf`

```
<VirtualHost *:80>
    ServerAdmin admin@example.com
    ServerName example.com
    ServerAlias www.example.com
    DocumentRoot /var/www/example.com/html
    ErrorLog ${APACHE_LOG_DIR}/example_error.log
    CustomLog ${APACHE_LOG_DIR}/example_access.log combined
</VirtualHost>
```

D. Activating the Virtual Host

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

E. Testing

Visiting `http://example.com` displayed the custom page.

Step-2: Default Virtual Host Behavior Testing

Only `example.com.conf` was enabled, so every unknown domain request loaded **example.com**.

Example:

- `http://webserverlab.com` → Loaded example.com
- `http://127.0.0.1` → Loaded example.com

This happens because **Apache uses the first enabled virtual host as default**.

To fix this issue, **each domain must have a separate virtual host configuration**, which is done in Step-3.

Step-3: Multiple Virtual Hosts Configuration

Objective

To configure and run multiple websites simultaneously using Apache Virtual Hosts.

---

1. Directory Structure for Second Website (anothervhost.com)

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

2. HTML Content Creation

File: `/var/www/anothervhost.com/html/index.html`

```
<!DOCTYPE html>
<html>
<head>
    <title>Welcome to AnotherVHost.com</title>
    <style>
        body { background-color: #e6f7ff; font-family: Arial; margin: 40px; }
        .box { background: #fff; padding: 30px; border-radius: 12px;
               box-shadow: 0 0 10px rgba(0,0,0,0.1); }
        h1 { color: #ff6600; }
    </style>
</head>
<body>
    <div class="box">
        <h1>🎉 AnotherVHost.com is Running!</h1>
        <p>This is the second virtual host site.</p>
    </div>
</body>
</html>
```

---

3. Virtual Host Config File

File: `/etc/apache2/sites-available/anothervhost.com.conf`

```
<VirtualHost *:80>
    ServerAdmin admin@anothervhost.com
    ServerName anothervhost.com
```

```
    ServerAlias www.anothervhost.com
    DocumentRoot /var/www/anothervhost.com/html
    ErrorLog ${APACHE_LOG_DIR}/anothervhost_error.log
    CustomLog ${APACHE_LOG_DIR}/anothervhost_access.log combined
</VirtualHost>
```

---

4. Activation

sudo a2ensite anothervhost.com.conf
sudo systemctl restart apache2

---

5. Hosts File Update

127.0.0.1 example.com
127.0.0.1 anothervhost.com

Testing & Verification

Test 1: Individual Access

- `http://example.com` → Showed example website
- `http://anothervhost.com` → Displayed second website

Test 2: Multiple Sites Running Together

Both worked simultaneously with different:

- Designs
- Content
- Domains

Test 3: Config Validation

sudo apache2ctl configtest # Syntax OK
ls /etc/apache2/sites-enabled/ # Both configs listed

Task-3: Dynamic Website Deployment (HTML + JavaScript)

Two interactive websites were deployed.

Website 1 – User Registration System (example.com)

Features

- Name, Email, Age, Country, Phone fields
- JS validation
- Dynamic result display
- Auto-reset
- Simple styling

Core JavaScript Code

```
document.getElementById("registrationForm").addEventListener("submit", function(e){
    e.preventDefault();

    let name = document.getElementById("fullName").value;

    alert("Registration Successful for: " + name);
});
```

Website 2 – Advanced Calculator (anothervhost.com)

Features

- Addition

- Subtraction
- Multiplication
- Division
- Modulus
- Power
- Square root
- Error handling
- Responsive layout

Key JS Function

```
function performAllCalculations(){
   let a = parseFloat(document.getElementById("num1").value);
   let b = parseFloat(document.getElementById("num2").value);

   document.getElementById("result").innerHTML = `
      Addition: ${a+b}<br>
      Subtraction: ${a-b}<br>
      Multiplication: ${a*b}<br>
      Division: ${b!==0 ? a/b : "Error"}<br>
      Modulus: ${a%b}<br>
      Power: ${a**b}<br>
      Square Root: ${Math.sqrt(a)}
   `;
}
```

Key Concepts Demonstrated

✔ Client-Side Processing

All logic handled by JavaScript without server-side code.

✔ DOM Manipulation

Used to update results dynamically.

✔ Input Validation

Ensured proper data before processing.

✔ Virtual Host Management

Successfully configured multiple sites on one Apache server.

Conclusion

In this lab session:

- Apache was installed and tested
- Local domains were configured
- Multiple virtual hosts were created
- Two dynamic interactive websites were deployed
- JavaScript was used for client-side form handling and calculations