Lab 1: Apache Web Server Installation & Maintenance

Moloy Banerjee - 2020831021
CSE-446 Web Engineering Lab

November 8, 2025

Abstract

This report details the steps taken to install and maintain the Apache Web Server, configure virtual hosts for multiple domains, and deploy simple dynamic websites using HTML and JavaScript for a user registration form and a product order form.

1 Task 1: Setting up Apache Web Server

This task involves installing the Apache web server, configuring the firewall, and basic domain setup.

Commands for Task 1

```
# Create Task1 directory
mkdir -p Lab1/Task1
cd Lab1/Task1

# Step 1: Install Apache
sudo apt update
sudo apt install apache2

# Step 2: Configure Firewall
sudo ufw app list
sudo ufw allow 'Apache'
sudo ufw status

# Step 3: Check Web Server
sudo systemctl status apache2

# Add webserverlab.com to hosts
echo "127.0.0.1 webserverlab.com" | sudo tee -a /etc/hosts

# Test the web server
curl http://localhost
curl http://webserverlab.com
```

1.1 Checkpoint 1

The Apache default page should be displayed when accessing `http://webserverlab.com` or `http://localhost` after completing the installation and hosts file modification.

2 Task 2: Setting up Virtual Hosts

This task sets up two distinct virtual hosts, `example.com` and `anothervhost.com`.

2.1 Part 1: Setting up example.com

Commands for example.com Setup

```
# Create Task2 directory
mkdir -p Lab1/Task2
cd Lab1/Task2

# Create directory structure for example.com
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 index.html for example.com
cat > /var/www/example.com/html/index.html << 'EOF'
<html>
<head>
<title>Welcome to Example.com</title>
</head>
<body>
<h1>Success! The example.com server block is working!</h1>
</body>
</html>
EOF

# Create virtual host configuration
sudo tee /etc/apache2/sites-available/example.com.conf << 'EOF'
<VirtualHost *:80>
ServerAdmin admin@example.com
ServerName example.com
ServerAlias www.example.com
DocumentRoot /var/www/example.com/html
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
EOF

# Enable the site and disable default
sudo a2ensite example.com.conf
sudo a2dissite 000-default.conf

# Test configuration and restart
sudo apache2ctl configtest
sudo systemctl restart apache2

# Add example.com to hosts
echo "127.0.0.1 example.com" | sudo tee -a /etc/hosts
```

2.1.1 Checkpoint 2

Accessing http://example.com should display the custom content: `Success! The example.com server block is working!`.

2.2 Testing Different Domains (Checkpoint 3)

Commands for Testing Domain Routing

```
curl http://example.com
curl http://webserverlab.com
curl http://127.0.0.1
```

2.2.1 Checkpoint 3 Explanation

- **curl** http://example.com: This should display the custom content for example.com because Apache matches the ServerName or ServerAlias in the example.com.conf virtual host.
- **curl** http://webserverlab.com: This should also display the custom content for example.com. After disabling 000-default.conf, the first enabled virtual host (example.com.conf) typically becomes the **default virtual host** for any requests that don't match a specific ServerName, which includes webserverlab.com.
- **curl** http://127.0.0.1: Similar to webserverlab.com, this IP-based request doesn't match a ServerName, so it falls back to displaying the content of the **default virtual host**, which is example.com.

2.3 Part 2: Setting up anothervhost.com

Commands for anothervhost.com Setup

```
#Create directory structure for 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

#Create index.html for anothervhost.com
cat > /var/www/anothervhost.com/html/index.html << 'EOF'
<html>
<head>
<title>Welcome to AnotherVHost.com</title>
</head>
<body>
<h1>Success! The anothervhost.com virtual host is working!</h1>
<p>This is a different virtual host with unique content.</p>
</body>
</html>
EOF

# Create virtual host configuration
sudo tee /etc/apache2/sites-available/anothervhost.com.conf << 'EOF'
<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>
EOF

# Enable the site
sudo a2ensite anothervhost.com.conf
sudo apache2ctl configtest
sudo systemctl restart apache2

#Add to hosts
echo "127.0.0.1 anothervhost.com" | sudo tee -a /etc/hosts
```

2.3.1 Checkpoint 4

Accessing http://anothervhost.com should now display the custom content for this new virtual host: `Success! The anothervhost.com virtual host is working! This is a different virtual host with unique content.`.

3 Task 3: Hosting Dynamic Websites

This task deploys dynamic (client-side JavaScript) content to the two virtual hosts.

3.1 Dynamic Website 1: User Registration Form (`example.com`)

Bash Command to Deploy Website 1

```
#Create Task3 directory
mkdir -p Lab1/Task3
cd Lab1/Task3

#Create dynamic content for example.com
cat > /var/www/example.com/html/index.html << 'EOF'
<!DOCTYPE html>
<html>
<head>
<title>Example.com User Registration</title>
<style>
body { font-family: Arial, sans-serif; margin: 40px; }
.container { max-width: 500px; margin: 0 auto; }
input[type="text"], input[type="email"], input[type="password"] {
    width: 100%; padding: 10px; margin: 5px 0; border: 1px solid #ddd;
}
input[type="submit"] {
    background: #4CAF50; color: white; padding: 10px 20px;
    border: none; cursor: pointer; margin-top: 10px;
}
.message { margin-top: 20px; padding: 10px; border-radius: 5px; }
</style>
</head>
<body>
<div class="container">
<h1>Example.com User Registration</h1>
<form onsubmit="return validateForm()">
<label>Full Name: </label>
<input type="text" id="name" required>
<label>Email:</label>
<input type="email" id="email" required>
<label>Password: </label>
<input type="password" id="password" required>
<label>Confirm Password: </label>
<input type="password" id="confirmPassword" required>
<label>Phone Number: </label>
<input type="text" id="phone" required>
<input type="submit" value="Register">
</form>
<div id="message" class="message"></div>
</div>
<script>
const messageDiv = document.getElementById('message');

function showMessage(text, color) {
    messageDiv.innerHTML = text;
    messageDiv.style.color = color;
    messageDiv.style.border = '2px solid ${color}';
    messageDiv.style.backgroundColor = color === 'green' ? '#e8f5e8' : '#ffe8e8';
}
```

```
function validateForm() {
    event.preventDefault();
    let name = document.getElementById(’name’).value;
    let email = document.getElementById(’email’).value;
    let password = document.getElementById(’password’).value;
    let confirmPassword = document.getElementById(’confirmPassword’).value;
    let phone = document.getElementById(’phone’).value;

    if (password !== confirmPassword) {
        showMessage(’Error: Passwords do not match!’, ’red’);
        return false;
    }

    if (phone.length < 10) {
        showMessage(’Error: Phone number must be at least 10 digits!’, ’red’);
        return false;
    }

    showMessage(’Success! Welcome <b>${name}</b>. Registration complete for ${email}. Phone: ${phone
    return false;
}
</script>
</body>
</html>
EOF
```

3.2 Dynamic Website 2: Product Order Form (anothervhost.com)

Bash Command to Deploy Website 2

```
#Create dynamic content for anothervhost.com
cat > /var/www/anothervhost.com/html/index.html << ’EOF’
<!DOCTYPE html>
<html>
<head>
<title>AnotherVHost - Product Order</title>
<style>
body { font-family: Arial, sans-serif; margin: 40px; }
.container { max-width: 600px; margin: 0 auto; }
.form-group { margin: 15px 0; }
label { display: block; margin-bottom: 5px; }
input, select, textarea { width: 100%; padding: 8px; border: 1px solid #ddd; }
button { background: #2196F3; color: white; padding: 10px 20px; border: none; cursor: pointer; }
.order-summary { margin-top: 20px; padding: 15px; border: 1px solid #ddd; }
</style>
</head>
<body>
<div class="container">
<h1>AnotherVHost - Product Order Form</h1>
<form onsubmit="return processOrder()">
<div class="form-group">
<label>Product Selection: </label>
<select id="product" onchange="updatePrice()">
<option value="">Select a product</option>
<option value="laptop" data-price="1200">Laptop - \$1200</option>
<option value="phone" data-price="800">Smartphone - \$800</option>
<option value="tablet" data-price="500">Tablet - \$500</option>
</select>
</div>
```

```
<div class="form-group">
<label>Quantity: </label>
<input type="number" id="quantity" min="1" value="1" onchange="updatePrice()">
</div>
<div class="form-group">
<label>Shipping Address: </label>
<textarea id="address" rows="3" required></textarea>
</div>
<div class="form-group">
<label>Payment Method: </label>
<select id="payment">
<option value="credit">Credit Card</option>
<option value="debit">Debit Card</option>
<option value="paypal">PayPal</option>
</select>
</div>
<button type="submit">Place Order</button>
</form>
<div id="orderSummary" class="order-summary" style="display: none;">
<h3>Order Summary</h3>
<div id="summaryContent"></div>
</div>
</div>
<script>
function updatePrice() {
    const productSelect = document.getElementById(’product’);
    const quantityInput = document.getElementById(’quantity’);
    const selectedOption = productSelect.options[productSelect.selectedIndex];

    if (selectedOption.value) {
        const price = selectedOption.getAttribute(’data-price’);
        const quantity = parseInt(quantityInput.value);
        const total = price * quantity;

        document.getElementById(’orderSummary’).style.display = ’block’;
        document.getElementById(’summaryContent’).innerHTML = ‘
        <p><strong>Product:</strong> ${selectedOption.text}</p>
        <p><strong>Unit Price:</strong> \$${price}</p>
        <p><strong>Quantity:</strong> ${quantity}</p>
        <p><strong>Total Amount:</strong> \$${total}</p>
        ‘;
    } else {
        document.getElementById(’orderSummary’).style.display = ’none’;
    }
}

function processOrder() {
    event.preventDefault();
    const product = document.getElementById(’product’).value;
    const quantity = document.getElementById(’quantity’).value;
    const address = document.getElementById(’address’).value;
    const payment = document.getElementById(’payment’).value;

    if (!product) {
        alert(’Please select a product!’);
        return false;
    }
    if (!address.trim()) {
        alert('Please enter shipping address!');
        return false;
    }

    const selectedOption = document.getElementById('product').options[document.getElementById('produ
    const price = selectedOption.getAttribute('data-price');
    const total = price * quantity;

    alert(`Order placed successfully!\n\nProduct: ${selectedOption.text}\nQuantity: ${quantity}\nTot
    document.querySelector('form').reset();
    document.getElementById('orderSummary').style.display = 'none';
    return false;
}
</script>
</body>
</html>
EOF

# Restart Apache to apply changes
sudo systemctl restart apache2
```

3.3 Checkpoint 5

The following dynamic websites should be working:

- http://example.com - User Registration Form
- http://anothervhost.com - Product Order Form

4 Summary

All tasks were completed with the following folder structure:

- Lab1/Task1/ - Apache installation and basic setup
- Lab1/Task2/ - Virtual hosts configuration
- Lab1/Task3/ - Dynamic websites deployment