

## Training Day - 8 Report:

### Exploring SSH, Web Hosting, and Apache Configuration

In this guide, we explore essential tools and techniques for managing web servers and secure remote connections. The focus is on SSH for secure communication, web hosting essentials, and configuring Apache to serve websites.

#### 1. SSH (Secure Shell)

- **What is SSH?** SSH is a protocol used to securely connect to remote machines over a network. It allows users to access servers, transfer files, and run commands remotely with encryption, ensuring secure data transmission.
- **Basic SSH Commands:**
  - `ssh user@hostname`: Connect to a remote machine.
  - `scp file.txt user@hostname:/path/to/destination`: Copy files over SSH.
  - `ssh-keygen`: Generate SSH keys for password-less login.
- **Use Cases:** SSH is essential for server management, code deployment, and secure file transfers.

#### 2. Web Hosting Basics

- **What is Web Hosting?** Web hosting involves storing website files on a server so that they can be accessed over the internet. Popular types of hosting include shared, VPS (Virtual Private Server), and dedicated hosting.
- **Web Hosting Components:**
  - **Domain Name:** A unique address for the website.
  - **Web Server:** Software like Apache or Nginx that serves website files.
  - **Database:** Manages website data (e.g., MySQL).
- **How Web Hosting Works:** Users access a website by typing the domain name into a browser. The request is routed to the server, which serves the appropriate files.

#### 3. Apache Web Server Configuration

- **What is Apache?** Apache is one of the most widely used open-source web servers. It serves static and dynamic content over the internet.

- **Installing Apache:** On Ubuntu:

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

### Configuring Apache:

- **Default Document Root:** By default, Apache stores website files in /var/www/html. You can change this by editing the 000-default.conf file in the /etc/apache2/sites-available/ directory.
- **Virtual Hosts:** Apache allows hosting multiple websites on a single server using Virtual Hosts. Example configuration:

```
<VirtualHost *:80>  
    ServerAdmin webmaster@localhost  
    DocumentRoot /var/www/example  
    ServerName example.com  
</VirtualHost>
```

- **Enabling Modules:** Apache has various modules (e.g., mod\_rewrite for URL rewriting). Enable them using a2enmod <module\_name>.
- **Restarting Apache:** After making configuration changes, restart Apache to apply them:

```
sudo systemctl restart apache2
```

### Conclusion:

This guide covers the basics of secure communication using SSH, essential concepts in web hosting, and configuring Apache as a web server. Mastery of these tools is fundamental for managing servers, deploying websites, and ensuring secure remote access for administration tasks.