Network Setup Guide on Linux with Python

Your Name

November 14, 2024

1 Network Setup on Linux with Python

This document describes how to set up a network topology on a Linux environment, using Python for automation. We will configure web servers, DNS, and DHCP services, along with some Python scripts for testing and automation.

1.1 Step 1: Install Necessary Packages

To begin, update the system and install Apache, BIND (DNS), and the ISC DHCP server.

```
# Update system packages
sudo apt update && sudo apt upgrade -y

# Install Apache (or Nginx), bind9 (DNS), and isc-dhcp-server (DHCP)
sudo apt install apache2 bind9 isc-dhcp-server python3 -y
```

1.2 Step 2: Configure Web Servers

1.2.1 Create Directories for Each Site

Create directories for each website that will be hosted.

```
sudo mkdir -p /var/www/sitel.com /var/www/site2.com /var/www/site3.com
```

1.2.2 Create HTML Pages

Add basic HTML files to each directory.

```
echo "<html><body><h1>Welcome to sitel.com</h1></body></html>" | sudo tee /var/www/sitel.com/index.html
echo "<html><body><h1>Welcome to site2.com</h1></body></html>" | sudo tee /var/www/site2.com/index.html
echo "<html><body><h1>Welcome to site3.com</h1></body></html>" | sudo tee /var/www/site3.com/index.html
```

1.2.3 Configure Apache

Create separate Apache configuration files for each site in `/etc/apache2/sites-available/`.

```
# sitel.com configuration
sudo bash -c 'cat > /etc/apache2/sites-available/sitel.com.conf' << EOF
<VirtualHost *:80>
    ServerName www.sitel.com
    DocumentRoot /var/www/sitel.com
</VirtualHost>
EOF

# site2.com configuration
sudo bash -c 'cat > /etc/apache2/sites-available/site2.com.conf' << EOF
<VirtualHost *:80>
```

```
    ServerName www.site2.com
    DocumentRoot /var/www/site2.com
</VirtualHost>
EOF

# site3.com configuration
sudo bash -c 'cat > /etc/apache2/sites-available/site3.com.conf' << EOF
<VirtualHost *:80>
    ServerName www.site3.com
    DocumentRoot /var/www/site3.com
</VirtualHost>
EOF

# Enable each site and restart Apache
sudo a2ensite sitel.com.conf site2.com.conf site3.com.conf
sudo systemctl restart apache2
```

1.3 Step 3: Configure DNS Server (BIND9)

Edit BIND configuration to add forward zones for each domain.

```
sudo nano /etc/bind/named.conf.local
```

Add the following configuration:

1.4 Step 4: Configure DHCP Server (ISC-DHCP-SERVER)

Open `/etc/dhcp/dhcpd.conf` and define the subnet and IP range.

```
sudo nano /etc/dhcp/dhcpd.conf
```

Add the following configuration for your subnet:

```
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.100 192.168.1.200;
    option routers 192.168.1.1;
    option domain-name-servers 192.168.1.1;
    option domain-name "localdomain";
}
```

1.5 Python Scripts for Automation

To automate DNS resolution testing, you can use a Python script.

```
import socket

def check_dns_resolution(hostname):
    try:
        ip = socket.gethostbyname(hostname)
        print(f"{hostname} resolved to {ip}")
    except socket.gaierror:
        print(f"DNS resolution failed for {hostname}")

# Test DNS resolution
check_dns_resolution("www.sitel.com")
check_dns_resolution("www.site2.com")
check_dns_resolution("www.site3.com")
```

1.6 Testing Configuration

To verify DNS resolution, use the following command:

```
nslookup www.sitel.com
```

And to test each website, use `curl`.

```
curl http://www.sitel.com
curl http://www.site2.com
curl http://www.site3.com
```