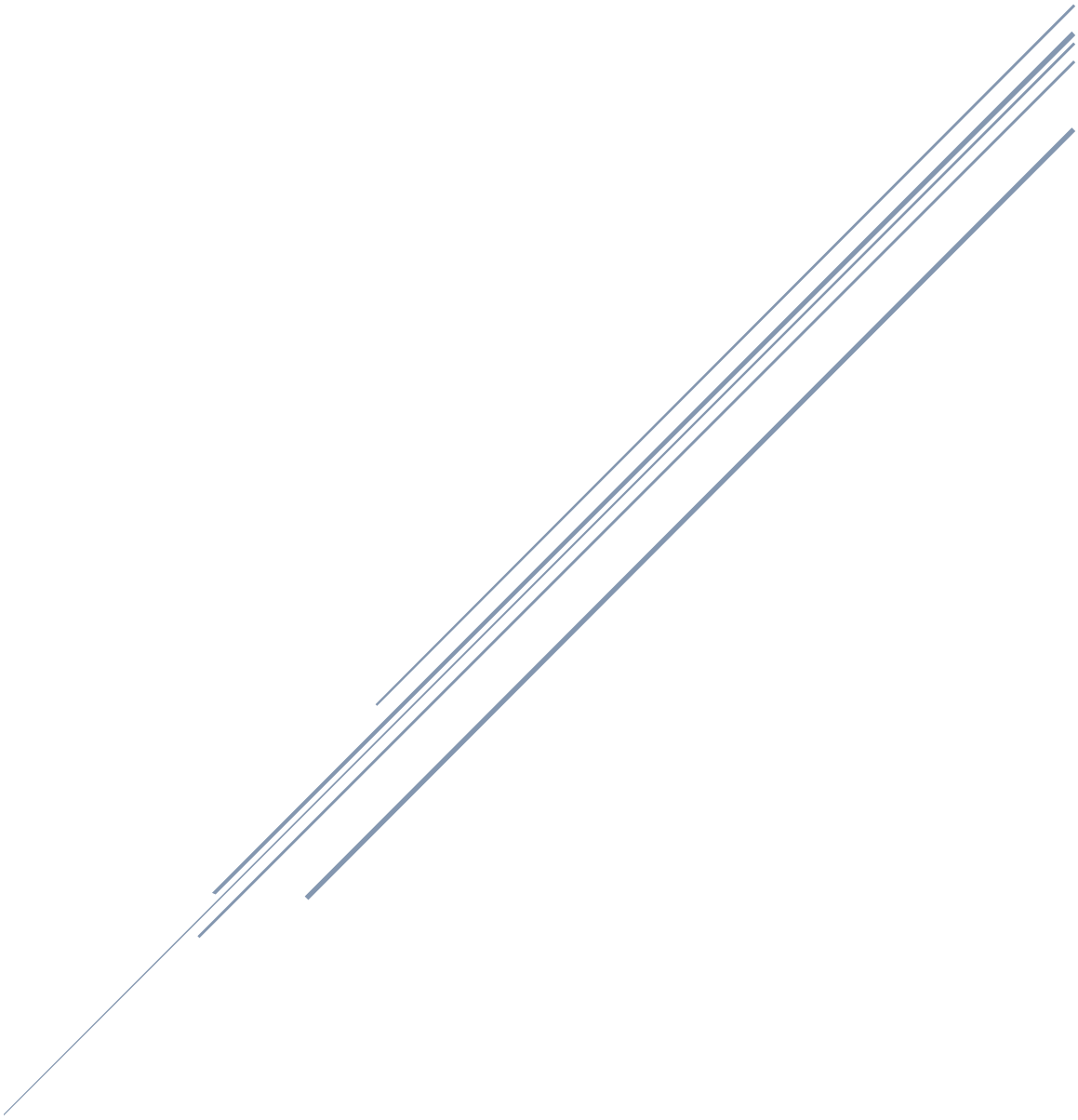
PARTICLES IN NETWORK AND SYSTEM



1. Three computers (laptops or desktops)
2. A network switch (can be a physical switch or a software-based switch in a Network Topology:

- All three computers will be connected to the switch, creating a basic local area network (LAN).

Step 1: Physical Connection

1. Connect all three computers to the network switch using Ethernet cables.
2. Ensure that the Ethernet cables are securely plugged into the Ethernet ports on both the computers and the switch.

Step 2: Network Configuration

1. On each computer, go to the network settings and assign a static IP address to each one. For example:
    - Computer 1: IP Address: 192.168.1.10, Subnet Mask: 255.255.255.0, Default Gateway: (leave blank for now)
    - Computer 2: IP Address: 192.168.1.20, Subnet Mask: 255.255.255.0, Default Gateway: (leave blank for now)
    - Computer 3: IP Address: 192.168.1.30, Subnet Mask: 255.255.255.0, Default Gateway: (leave blank for now)
2. Save the network settings on each computer.

Step 3: Test Network Communication

1. Open the Command Prompt (Windows) or Terminal (Mac/Linux) on Computer 1.
2. Ping Computer 2 using its IP address: `ping 192.168.1.20`.
3. If successful, you should see a reply from Computer 2. This confirms that the two computers can communicate.

Step 4: Set Default Gateway

1. Choose one computer to act as the default gateway. For this tutorial, let's use Computer 1 as the default gateway.
2. Go to the network settings on Computer 1 and set its default gateway to the IP address of the switch (e.g., 192.168.1.1).
3. Save the network settings on Computer 1.

Step 5: Test Internet Connectivity

1. On Computer 1, open a web browser and try to access a website (e.g., www.example.com).
2. If successful, this confirms that Computer 1 can access the internet through the switch acting as the default gateway.

Sharing Files

3. Now that the network is set up, you can share files between computers using the network.
4. You can create a shared folder on one computer and give access to other computers on the network to read and write files in that folder. This allows for easy file sharing and collaboration within the network.

In this tutorial, we will create a simple network using two Linux Ubuntu servers. We will set up a basic client-server architecture where one server acts as a web server and the other as a client to access the web server's content.

Equipment Needed:

1. Two Linux Ubuntu servers (can be virtual machines or physical servers)
2. Internet connection for software installation and updates

Network Topology:

- Server 1 (Web Server): This server will host a basic website.
- Server 2 (Client): This server will access the website hosted on Server 1.

Step 1: Install Ubuntu Servers

1. Install Linux Ubuntu Server on both Server 1 and Server 2.
2. During the installation, set up the network interfaces with appropriate IP addresses for each server.
    - Server 1: IP Address: 192.168.1.10 (or any desired IP), Subnet Mask: 255.255.255.0, Gateway: IP of your router (for internet access)
    - Server 2: IP Address: 192.168.1.20 (or any desired IP), Subnet Mask: 255.255.255.0, Gateway: IP of your router (for internet access)

Step 2: Update the Servers

1. After installation, update the servers to ensure they have the latest software and security patches. Open the Terminal and run the following commands.

   sudo apt update.
2. **sudo apt upgrade**

   On Server 1, install the Apache web server using the following command

   sudo apt install apache2
3.

Go to the web server's default web directory

cd /var/www/html

sudo nano index.html

3. Inside the file, add some content (e.g., "Welcome to My Website!").
4. Save and close the file (Ctrl + X, Y, Enter).

Step 5: Test the Web Server

1. Open a web browser on Server 1 or any other device connected to the same network.
2. Enter the IP address of Server 1 (e.g., 192.168.1.10) in the browser's address bar.
3. If successful, you should see the content of the basic website you created.

Step 6: Access the Web Server from Server 2

1. On Server 2, open a web browser.
2. Enter the IP address of Server 1 (e.g., 192.168.1.10) in the browser's address bar.
3. If successful, Server 2 should now access the website hosted on Server 1

Congratulations! You have successfully set up a small network with two Linux Ubuntu servers, where one server acts as a web server and the other as a client to access the web server's content.

4. You learned how to install Apache web server and access a basic website from another server within the network.
5. Example Scenario: Expanding the Network
6. You can further expand the network by adding more servers, configuring additional services like a database server or email server, and implementing security measures such as firewalls and access controls.
7. (Note: This tutorial provides a basic setup for beginners. In real-world scenarios, larger and more complex networks would involve additional servers, services, and security configurations.)
8.

Equipment Needed:

1. Two Windows Server 2019 machines (can be virtual machines or physical servers)
2. Internet connection for software installation and updates

Network Topology:

- Server 1 (Domain Controller): This server will act as a domain controller for the network.
- Server 2 (Client): This server will join the domain managed by the domain controller.

Step 1: Install Windows Server 2019

1. Install Windows Server 2019 on both Server 1 and Server 2.
2. During the installation, set up the network interfaces with appropriate IP addresses for each server.
    - Server 1: IP Address: 192.168.1.10 (or any desired IP), Subnet Mask: 255.255.255.0, Gateway: IP of your router (for internet access)
    - Server 2: IP Address: 192.168.1.20 (or any desired IP), Subnet Mask: 255.255.255.0, Gateway: IP of your router (for internet access)

Step 2: Update the Servers

1. After installation, update the servers to ensure they have the latest software and security patches. Open Command Prompt as Administrator and run the following commands

    sconfig

    Use option 6 to download and install updates.

Step 3: Configure Server 1 as Domain Controller

1. On Server 1, open Server Manager.
2. Click "Add roles and features" and go through the wizard to install the Active Directory Domain Services (AD DS) role.
3. After the installation completes, click on "Promote this server to a domain controller."
4. Follow the wizard to set up a new forest and domain with a domain name of your choice (e.g., mydomain.local).
5. Set a Directory Services Restore Mode (DSRM) password.
6. Complete the wizard to promote Server 1 as a domain controller.

Step 4: Join Server 2 to the Domain

1. On Server 2, open Server Manager.
2. Click "Add roles and features" and go through the wizard to install the Active Directory Domain Services (AD DS) role.
3. After the installation completes, open the "System Properties" window and click on "Change" to join a domain.

4. Enter the domain name set up on Server 1 (e.g., mydomain.local) and provide domain administrator credentials.
5. Restart Server 2 when prompted to apply the changes.

Step 5: Test Domain Membership

1. Log in to Server 2 using domain administrator credentials.
2. Open File Explorer and check if you can access shared resources on Server 1 by entering "\server1" in the address bar.
3. If successful, Server 2 has joined the domain managed by Serv.

System Security Practices

Introduction: System security is essential for protecting your Linux servers from potential threats and vulnerabilities. In this practical example, we will demonstrate key security practices and configurations to enhance the security of a Linux server.

Scenario: You have set up a Linux server (e.g., Ubuntu Server) to host a web application. The server is accessible over the internet, and you want to ensure its security to prevent unauthorized access and potential attacks.

Step 1: Update and Patch Management

- Regularly update the server's packages and apply security patches to keep it up-to-date.
- Example: Open a terminal and run the following commands to update the package list and apply updates:

  sudo apt update
- sudo apt upgrade

  Configure Firewall

Set up a firewall to control incoming and outgoing network traffic. Example: Use the Uncomplicated Firewall (UFW) to enable only necessary ports (e.g., HTTP port 80 and HTTPS port 443) and deny others.

sudo ufw enable

sudo ufw allow 80/tcp

sudo ufw allow 443/tcp

ep 3: Secure SSH Access (additional)

- Change the default SSH port and disable root login to enhance SSH security.
- Example: Edit the SSH configuration file to change the port and disable root login

sudo nano /etc/ssh/sshd_config

Change "Port" to a custom port number (e.g., 2222)

Set "PermitRootLogin" to "no"

sudo systemctl restart sshd

implement Fail2Ban

- Install and configure Fail2Ban to block suspicious IP addresses that attempt multiple failed login attempts.
- Example: Install Fail2Ban and create a custom jail configuration for SSH

  sudo apt install fail2ban
- sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
- sudo nano /etc/fail2ban/jail.local
- # Add the following under [sshd]
- enabled = true

Step 6: Limit User Access

- Grant specific permissions to users based on their roles to restrict unauthorized access.
- Example: Create a new user and give it the necessary permissions (e.g., for web application management):

  sudo adduser newuser

sudo usermod -aG sudo newuser

Step 7: Monitor Logs

- Regularly monitor system logs for suspicious activities or errors.
- Example: Use `journalctl` to view system logs: **journalctl -xe**