Guide: Setting Up a Samba Active Directory Domain Controller on Ubuntu Server

This document provides a comprehensive, step-by-step guide for setting up an Active Directory Domain Controller (AD DC) on an Ubuntu Server using Samba. The guide covers the entire process, from the initial creation of a virtual machine in VirtualBox to configuring the server, provisioning the domain, and successfully joining a Windows 10 client to the newly created environment.

The scope of this project involves creating a self-contained domain for testing and development purposes. All components, including the server and client, will operate on an internal virtual network. The key network parameters for this configuration are summarized below.

Parameter	Value
Domain	fct.kel.ac.lk
Domain Controller IP	172.16.0.10
Subnet Mask	/24
Default Gateway	172.16.0.1

1.0 Phase 1: Ubuntu Server Virtual Machine Setup

The foundation of any server role is a properly configured virtual machine. This section details the creation and network adapter configuration of the Ubuntu Server VM within VirtualBox. Executing these steps correctly is a critical prerequisite for establishing a stable server environment.

The virtual machine creation process in VirtualBox is as follows:

1. Create a new virtual machine named Ubuntu DC.
2. Select the Ubuntu Server ISO image as the installation media.
3. Choose to **Skip unattended installation** to perform a manual setup.
4. Allocate the following hardware resources:
    - **Base Memory (RAM):** 4 GB
    - **Processors:** 4 cores
    - **Disk Size:** 25 GB

Once the base VM is created, configure its network adapters before the first boot:

- **Adapter 1:** Set to NAT. This provides the VM with internet access for downloading packages.
- **Adapter 2:** Set to Internal Network. This adapter will be used for all communication between the domain controller and its clients.

With the VM hardware and networking defined, the next step is to proceed with the Ubuntu Server installation.

2.0 Phase 2: Ubuntu Server Installation and Initial Network Configuration

The operating system installation phase is where the server's network identity is first established. Correctly configuring the network interfaces during this stage is essential for the server's future role as a domain controller, as it defines the static IP address that all clients will use for authentication and DNS services.

The key steps during the Ubuntu Server installation are outlined below:

1. Boot the virtual machine from the Ubuntu Server ISO.
2. Select the desired language (English) and proceed through the initial setup screens.
3. When prompted for network configuration, you will see two interfaces. The first (NAT) will have a DHCP-assigned address, while the second (Internal Network) will be unconfigured.
4. Configure a static IP address for the internal network adapter (ENP0S8). This is a critical step.
    - Select the ENP0S8 interface and edit its IPv4 settings. Change the method to Manual.
    - Enter the following network details:
        - **Subnet:** 172.16.0.0/24
        - **Address:** 172.16.0.10
        - **Gateway:** 172.16.0.1
        - **Name server:** 172.16.0.10
5. Continue with the remaining installation choices:
    - **Proxy address:** Leave blank.
    - **Mirror address:** Use the default provided.
    - **Storage:** Configure the installer to use the entire disk.
    - **Profile setup:** Create a local user account, setting the user's name, server name, and password (e.g., 1234).
    - **SSH Setup:** Check the box to **Install OpenSSH server**.
    - **Server Snaps:** Do not install any additional services at this stage.

After the installation completes, the system will reboot. It may report a failure to eject the ISO image, but this message can be safely ignored as VirtualBox typically handles the virtual media automatically.

(Note: For users who prefer to manage the server via SSH, the NAT adapter in VirtualBox can be temporarily switched to Bridged Adapter. This will assign an IP from your physical network, allowing you to connect with an SSH client. The narrator in the source tutorial works directly in the console.)

Once the installation is complete and the server has rebooted, the next crucial phase is the post-installation system configuration.

3.0 Phase 3: Post-Installation System Configuration

After the base OS is installed, the server must be prepared for the Samba AD DC role. This involves synchronizing time, updating software, and configuring core network identity files (/etc/hostname and /etc/hosts). These steps ensure the server can correctly identify itself on the network before any domain services are installed.

1. **Correct the System Time Zone**
2. First, gain superuser privileges to execute administrative commands without repeatedly typing sudo.
3. Next, check the current time and date settings.
4. Set the correct time zone. Time synchronization is critical for Kerberos authentication. In this setup, Asia/Colombo is used.
5. **Install Network Tools and Update System Packages**
6. Install the net-tools package, which provides useful network diagnostic utilities.
7. Update and upgrade all system packages to their latest versions. *Note: To ensure the NAT adapter is used for internet access, the internal network adapter (ENP0S8) may need to be temporarily disabled in the VirtualBox UI during this step. This step is highly recommended for production but was skipped in the source video to save time.*
8. **Configure Hostname and Hosts File**
9. These files are essential for mapping the server's hostname and domain to its static IP address.
    - **Step 1: Change the Hostname** Edit the /etc/hostname file and change its content to DC to reflect its role as a Domain Controller.
    - **Step 2: Update the Hosts File** Edit the /etc/hosts file to map the server's static IP address to its fully qualified domain name (FQDN) and its short hostname.
    - Ensure the following line is present and correctly configured:
10. **Reboot the Server**
11. A reboot is necessary to apply the hostname changes.

With the base system configuration complete, the server is now ready for the installation of Samba and its related components.

4.0 Phase 4: Installing and Provisioning the Samba AD DC

This is the core phase where the Ubuntu server is transformed into a functional Active Directory Domain Controller. This involves installing Samba and its dependencies, configuring Kerberos, and running the domain provisioning process to create the Active Directory domain itself.

1. **Install Required Packages**
2. The following packages are required to enable AD DC functionality.

Package	Purpose
samba	Provides the core services for domain controller roles and file sharing.
krb5-config	Manages Kerberos authentication for secure logins and user ticketing.
winbind	Allows the Linux system to communicate with and understand a Windows domain.
smbclient	A command-line tool for accessing network shares, useful for testing.

Install all required packages with a single command:

```
sudo apt install samba krb5-config winbind smbclient
```

1. **Configure Kerberos**
2. During the installation, the krb5-config package will prompt for configuration details. Enter the following values, ensuring the realm is in uppercase as is standard for Kerberos.
    1. **Default Kerberos version 5 realm:** FCT.KEL.AC.LK (Confirm the auto-filled value).
    2. **Kerberos servers for your realm:** DC.FCT.KEL.AC.LK
    3. **Administrative server for your Kerberos realm:** DC.FCT.KEL.AC.LK
3. **Provision the Samba Domain**
    1. **Step 1: Back up the Original Samba Configuration** As a precautionary measure, rename the default Samba configuration file before creating a new one.
    2. **Step 2: Run the Domain Provision Tool** Execute the Samba tool to begin the interactive domain provisioning process.
    3. **Step 3: Provide Provisioning Details** Enter the following values when prompted by the tool:
        - **Realm:** fct.kel.ac.lk
        - **Domain:** FCT
        - **Server Role:** dc

- **DNS backend:** SAMBA_INTERNAL
- **DNS forwarder IP address:** Enter 8.8.8.8 and press Enter. You will be prompted for an additional forwarder; enter 4.2.2.2.
- **Administrator Password:** Provide a strong password. A simple password like 1234 will be rejected as it does not meet minimum complexity requirements.

After successfully provisioning the domain, the next step is to configure system services and DNS to use the new Samba AD DC.

5.0 Phase 5: Final Service and DNS Configuration

With the domain provisioned, the server is not yet self-aware. This crucial phase re-configures the server to use its *own* services for authentication and DNS. We will replace the default Kerberos configuration with the one generated by Samba, disable conflicting system services, and force the server to use itself as its DNS resolver—a critical step for a functioning domain controller.

1. **Copy the New Kerberos Configuration**
2. Copy the Kerberos configuration file generated during the Samba provisioning process to the system-wide location. This ensures all system components use the correct Kerberos settings for the new domain.
3. **Manage System Services**
    - **Step 1: Disable Conflicting Services** Disable services that would interfere with Samba's ability to operate as a full AD DC.
    - **Step 2: Unmask the Samba AD DC Service** The primary Samba AD DC service is masked (blocked) by default in Ubuntu. Unmask it to allow it to be enabled and started.
    - **Step 3: Enable and Start the Samba AD DC Service** Enable the service to ensure it starts automatically on boot.
4. **Configure the Local DNS Resolver**
    - **Step 1: Remove the Existing resolv.conf File** The existing file is often managed by other services and must be replaced with a static configuration.
    - **Step 2: Create a New resolv.conf File** Use a text editor to create a new configuration file.
    - **Step 3: Add the Correct Configuration** This configuration directs the server to use its own local Samba DNS for all name resolution. This is critical for the DC to find its own records. Add the following content to the file:

With the server's core services and DNS correctly configured, it's time to verify the setup and create a user for the domain.

6.0 Phase 6: Verification and Domain User Creation

Before connecting a client, it is essential to verify that the domain is functioning correctly and that internal name resolution works as expected. This section covers these critical verification steps and the creation of a standard domain user account for client login testing.

1. **Check the Domain Functional Level**
2. Verify that the domain has been provisioned correctly by checking its functional level. The expected output should show a level of 2008_R2, which is perfectly acceptable for modern environments.
3. **Create a New Domain User**
4. Create a new user account within the Active Directory domain. This user will be used later to log in from the Windows 10 client. You will be prompted to enter and confirm a strong password for the new user.
5. **Test Domain Name Resolution**
    - The initial ping to DC.fct.kel.ac.lk may fail. As shown in the source, this is because the internal network adapter (ENP0S8) was likely disabled earlier to allow for package updates. **Re-enable the adapter in the VirtualBox UI** to restore communication on the internal network.
    - Next, use ping to verify that the server can resolve and reach the domain by its FQDN, the DC's FQDN, and its short hostname.
    - Finally, use nslookup to confirm that the DNS server correctly reports the IP address for the domain's FQDN. Note that immediately after network changes, this command may initially return the IP of the NAT adapter. A reboot is often necessary to refresh the network state and ensure the query correctly resolves to the internal IP address (172.16.0.10) of the domain controller.

With the domain verified and a user created, the Ubuntu server is fully configured. The final phase is to set up the Windows client and join it to the domain.

7.0 Phase 7: Windows 10 Client Configuration and Domain Join

The ultimate test of the Samba AD DC setup is successfully joining a Windows client to the domain. This section outlines the required VM settings for the Windows client, its network configuration, and the final domain join procedure.

1. **Windows 10 VM Setup in VirtualBox**
2. Create and configure a Windows 10 virtual machine with the following key settings:
    - **Base Memory (RAM):** 4 GB
    - **Processors:** 5 cores
    - **Network:** The network adapter **must be set to Internal Network** to ensure it is on the same virtual network as the Ubuntu DC server.
3. **Windows 10 Installation Highlights**

4. During the Windows installation, make the following choices:
    - **Installation Type:** Select **Windows 10 Pro**. A "Home" edition cannot be joined to a domain.
    - **Setup:** Choose to **Continue with limited setup** to create an offline local account initially.
    - **Local User:** Create a local user (e.g., ashur) and set a password.
5. **Configure the Windows Network Adapter**
    - **Step 1:** After booting into Windows, open the Run dialog (Win + R) and type ncpa.cpl to open the Network Connections panel.
    - **Step 2:** Right-click the network adapter, select 'Properties', and uncheck the box for 'Internet Protocol Version 6 (TCP/IPv6)' to disable it.
    - **Step 3:** Select 'Internet Protocol Version 4 (TCP/IPv4)', click 'Properties', and configure the static IP settings as follows:
        - **IP address:** 172.16.0.120
        - **Subnet mask:** 255.255.255.0
        - **Default gateway:** 172.16.0.1
        - **Preferred DNS server:** 172.16.0.10. This setting is critical. The client **must** point to the Samba DC's IP for all DNS queries to resolve domain resources.
6. **Verify Connectivity and Join the Domain**
    - **Step 1: Verify Connectivity** Open a Command Prompt and ping the domain controller's IP address (172.16.0.10) to confirm a stable network connection.
    - **Step 2: Handle Potential nslookup Failure** As demonstrated in the source video, running nslookup fct.kel.ac.lk on the Windows client may fail and return "unknown" even when the ping test succeeds. This is a common scenario. Do not be deterred by this; proceed with the domain join attempt.
    - **Step 3: Access System Properties** Right-click 'This PC', select 'Properties', and click 'Advanced system settings'.
    - **Step 4: Change Computer's Domain** Navigate to the 'Computer Name' tab, click 'Change...', select the 'Domain' radio button, and enter the domain name: fct.kel.ac.lk.
    - **Step 5: Authenticate** When prompted for credentials, enter the username Administrator and the strong password that was set during the Samba domain provisioning phase.
    - **Step 6: Restart** Upon successful authentication, you will receive the "Welcome to the fct.kel.ac.lk domain" message. Close all dialogues and restart the machine to apply domain membership changes.

After the restart, the last action is to log in using the domain user account created earlier to confirm successful integration.

8.0 Phase 8: Logging In with a Domain User

This is the final validation step. A successful login confirms that the Samba domain controller can authenticate users, allowing them to access client machines on the network and receive domain policies.

To log in with the domain user account:

1. On the Windows login screen, select the **'Other user'** option.
2. Verify that the login prompt shows **"Sign in to: FCT"** below the password field, indicating it is ready to authenticate against the domain.
3. Enter the username created on the Samba server: FCTuser01.
4. Enter the corresponding strong password for that user.
5. Press Enter and confirm that you successfully log in. Windows will prepare a new user profile for this domain account.

By successfully logging in with a domain user, the setup is complete. You have now deployed a functional Samba Active Directory Domain Controller on an Ubuntu Server and fully integrated a Windows 10 client, demonstrating a successful cross-platform identity and authentication system.