🖥️Windows Server and Ubuntu Client Domain Setup (Project Report)

📌Project Overview

In this project, I set up a Windows Server 2022 VM as an Active Directory Domain Controller and connected an Ubuntu 24.04 Desktop client to it through a host-only internal network. Both systems also had access to the internet using NAT. This setup allowed me to simulate a real enterprise domain environment in VirtualBox for learning and demonstration purposes. 💡

⚙️Windows Server VM Configuration

🧰VirtualBox Adapter Setup

I configured two network adapters for the Windows Server VM:

- **Adapter 1 (NAT)** 🩲: For internet access. This was left on DHCP.
- **Adapter 2 (Host-Only Adapter)** 🔗: For internal domain communication using `vboxnet0`.

🛠️Static IP Assignment (Host-Only Adapter)

After launching the server, I opened CMD and ran `ipconfig` to identify the Host-Only adapter. Initially, it showed an APIPA address like `192.168.x.x`. I changed the IP settings manually to:

- **IP Address**: `192.168.100.1`
- **Subnet Mask**: `255.255.255.0`
- **Gateway**: *(left blank)*
- **DNS**: 192.168.100 (since the server hosts its own DNS for AD)

The NAT adapter was set to DHCP to allow internet access. 🕐

🏺 Ubuntu Client VM Configuration

🧰VirtualBox Adapter Setup

I set up two adapters for the Ubuntu client:

- **Adapter 1 (NAT)** 🩲: For internet (DHCP)
- **Adapter 2 (Host-Only)** 🔗: Internal LAN using `vboxnet0`

🔎Identifying Network Interfaces

I ran `ip a` and identified the Host-Only adapter (usually `enp0s8`) which initially had an address like `192.168.x.x`. I configured it manually using the GUI with these settings:

- **IP Address**: `192.168.100.3`
- **Netmask**: `/24`
- **Gateway**: `192.168.100.1`
- **DNS**: `192.168.100.1`
- **Search Domain**: `physicstutors.org`

To prevent the host-only NIC from being used for internet routing, I ran:

```
sudo nmcli connection modify "enp0s8" ipv4.never-default yes
sudo nmcli connection up "enp0s8"
```

🔌Connectivity Test

On Ubuntu 🧪

- I successfully pinged the Windows Server using:

  ```
  ping 192.168.100.1
  ```

- I also confirmed DNS resolution with:

  ```
  nslookup physicstutors.org
  ```

On Windows Server 🧪

- I pinged the Ubuntu client:

  ```
  ping 192.168.100.3
  ```

- I also verified internet access with:

  ```
  ping google.com
  ```

💼Domain Join (Ubuntu)

📦Installing Required Packages

I installed the following:

```
sudo apt update
sudo apt install realmd sssd krb5-user samba-common samba-common-bin adcli
```

During the Kerberos configuration, I entered the realm as:

```
PHYSICSTUTORS.ORG
```

📝krb5.conf Fix (If Needed)

If I skipped or misconfigured Kerberos, I corrected it using:

```
sudo nano /etc/krb5.conf
```

Set:

```
[libdefaults]
default_realm = PHYSICSTUTORS.ORG
```

🔐Joining the Domain

I discovered and joined the domain using:

```
sudo realm discover physicstutors.org
sudo realm join --user=Administrator physicstutors.org
```

I entered the domain Administrator password when prompted, and the join was successful. 🔗

📊Summary Table

System	IP Address	Gateway	DNS	Role
🖥️Windows Server	192.168.100.1	*(Blank)*	192.168.100.1	AD Domain Controller

System	IP Address	Gateway	DNS	Role
🏺 Ubuntu Client	192.168.100.3	192.168.100.1	192.168.100.1	AD Client (Ubuntu)

Conclusion

This setup enabled me to practice core Windows Server administration skills, including Active Directory, DNS configuration, static IP assignment, and domain joining, from a Linux system. 📚

📏Screenshots from this setup have been added to my GitHub repository as part of my DevOps and System Admin learning portfolio.

Prepared by: Yinka Ajibola ✍️