

## **Active Directory Home Lab**

### **Windows Server 2019 & Windows 10 (VirtualBox)**

#### **Overview**

This document describes the design, deployment, and validation of a small Active Directory environment built in a virtualized lab. The environment includes a Windows Server 2019 Domain Controller providing Active Directory Domain Services (AD DS), DNS, DHCP, and routing/NAT, along with a Windows 10 client joined to the domain.

#### **Technologies**

- Windows Server 2019
- Windows 10
- Active Directory Domain Services (AD DS)
- DNS and DHCP
- RRAS (Routing and Network Address Translation)
- VirtualBox

#### **Lab Architecture Overview**

- 1 Domain Controller (Windows Server 2019)
  - AD DS
  - DNS (authoritative for domain)
  - DHCP (internal scope)
  - RRAS/NAT (internal → external routing)
- 1 Client (Windows 10 Pro)
  - Domain-joined
  - DHCP-assigned IP
  - Authenticates using domain users

**Author:** Mohamed Eljerary

**Purpose:** Hands-on enterprise Windows infrastructure exposure, practice, and documentation

## **Summary**

This lab demonstrates the deployment of a basic Active Directory environment representative of a small enterprise network. A Windows Server 2019 system was configured as a Domain Controller to provide centralized authentication, name resolution, IP address management, and outbound network access for internal clients.

Key objectives of the lab included configuring DNS as a dependency for Active Directory, implementing DHCP for automated IP assignment, and enabling routing and network address translation (NAT) to allow internal clients to reach external networks. A Windows 10 client was joined to the domain and validated through domain user authentication and connectivity testing.

The environment was verified through successful domain joins, DNS resolution checks, DHCP lease assignment, user authentication testing, and client internet access via the Domain Controller.

## **Acknowledgment**

This lab was informed by publicly available instructional material on Active Directory, including content by Josh Madakor's IT guides. The configuration, validation, and documentation reflect my own implementation and troubleshooting decisions.

## Design Decisions and Tradeoffs

- **Two Network Interfaces on the Domain Controller**  
The Domain Controller was configured with separate internal and external network interfaces to isolate internal domain traffic from external network access while allowing controlled routing between networks.
- **RRAS/NAT on the Domain Controller**  
RRAS with NAT was implemented to enable internal clients to access external networks without exposing internal addressing directly, reflecting common small-environment and lab routing designs.
- **Centralized DHCP on the Domain Controller**  
DHCP was hosted on the Domain Controller to provide centralized IP address management and seamless integration with Active Directory and DNS.
- **Bulk User Provisioning via PowerShell**  
Automated user creation was used to simulate a realistic enterprise environment and demonstrate administrative automation rather than manual, one-off account creation.

## Installing Our Hypervisor Software

Which virtualizer are we going to use? The guide I'm using uses VirtualBox so for now we'll use that and then maybe in the future when I try replicating, I try moving to VMware since I've used that for classes previously, and then maybe proxmox, but for the purposes of this lab I'm won't focus too much on the practice of virtualization beyond what is required for this lab. I mainly want to expose myself to Active Directory.

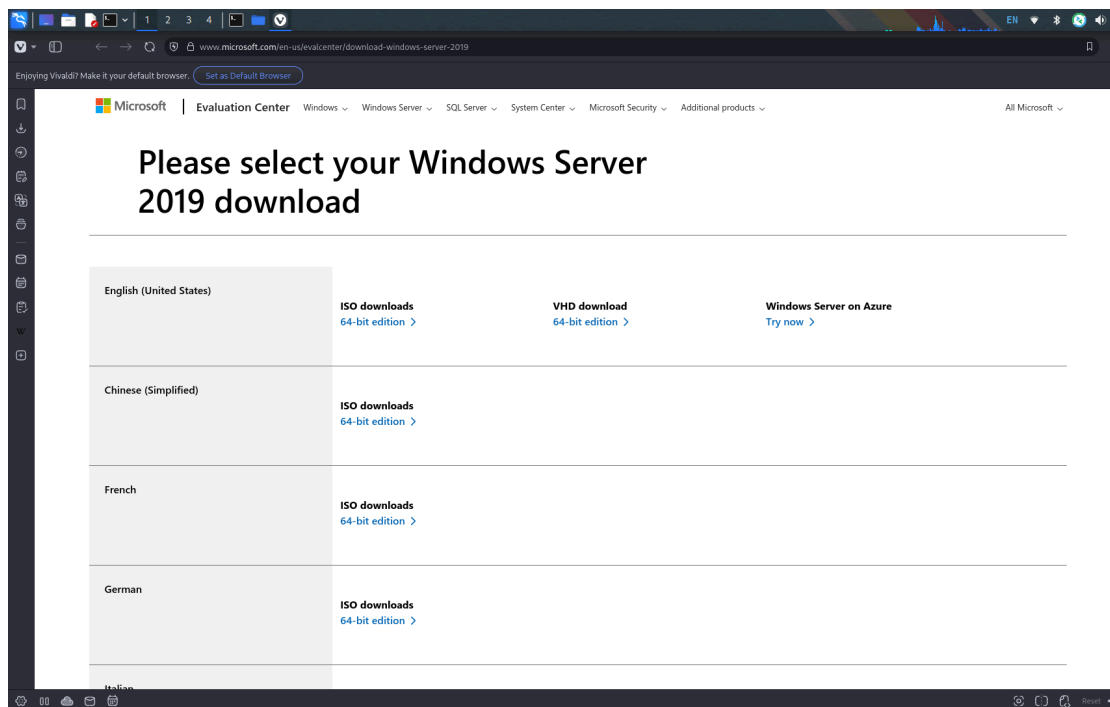
I'm on a Kali Linux machine so to start I have to check the RAM installed on my machine, the CPU cores/threads, and the Disk space available in order to ensure proper setup later for the Windows Server 2019 and Windows 10 VMs.

After running `free -h` in my Kali Linux shell terminal, we can see that RAM is at 23 GB total, plenty. With `lscpu | grep -E '^CPU(s)|^Thread|^Core'` CPU is at 8 threads (4 cores × 2 threads), so ideal for two or more VMs. Disk space, after running `df -h /` is 467 GB total, 174 GB free, so enough room for multiple OS images + snapshots.

To install the VM software from the video (virtualbox), I ran `sudo apt install virtualbox virtualbox-ext-pack -y` to download it onto my computer (after updating of course). After restarting my computer I opened the shell terminal again and created the first new directory called ActiveDirectoryLab\_Project.

## Install Windows Server 2019 & Windows 10 ISO's

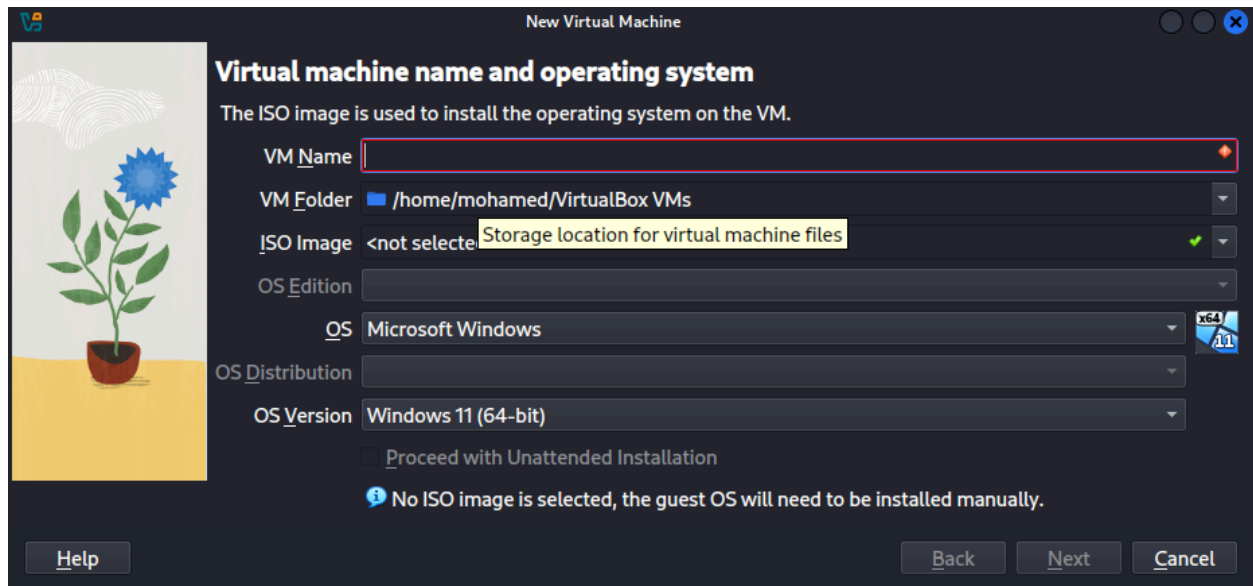
Install the ISO from <https://www.microsoft.com/en-us/evalcenter/download-windows-server-2019>



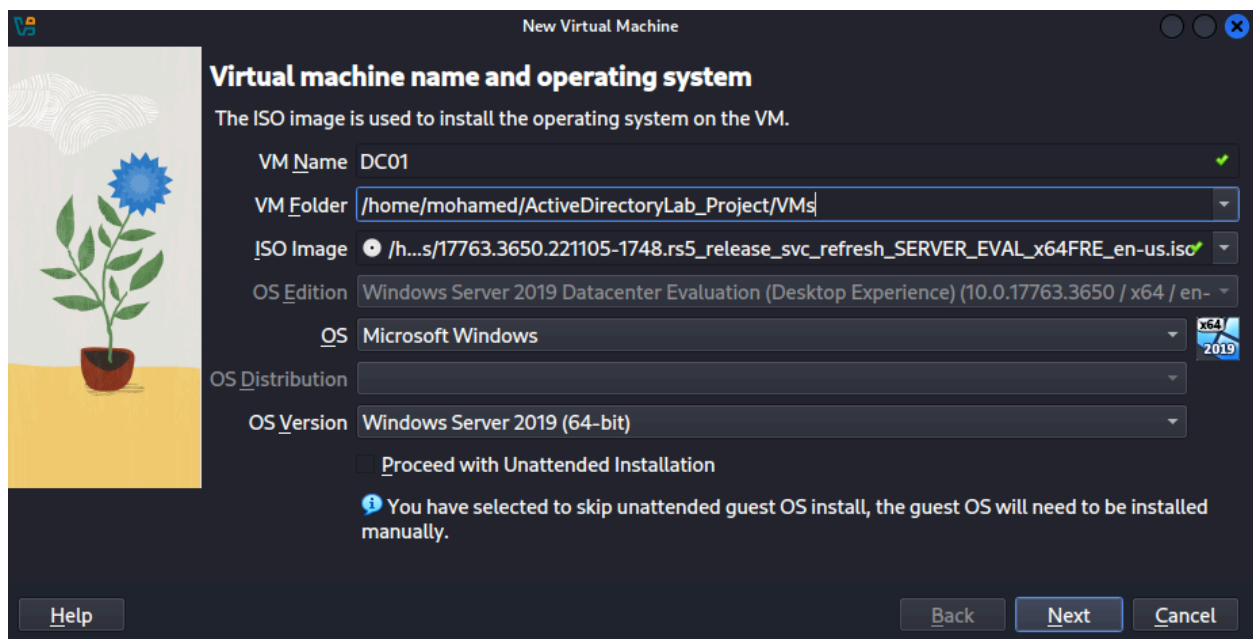
I want the option for the US English ISO Downloads, so I make a new folder called ISO in the new active directory folder, and download it to `\ActiveDirectoryLab_Project\ISOs\` location. After this finished, I opened virtualbox (the download takes a minute for the windows server so what I did before proceeding was also install the windows 10 ISO from <https://www.microsoft.com/en-ca/software-download/windows10iso> to just not have to go back later when the portion concerning windows 10 starts).

## New Windows Server 2019 VM In VirtualBox

Upon opening virtualbox I clicked new, then in the new window that pops up you see:

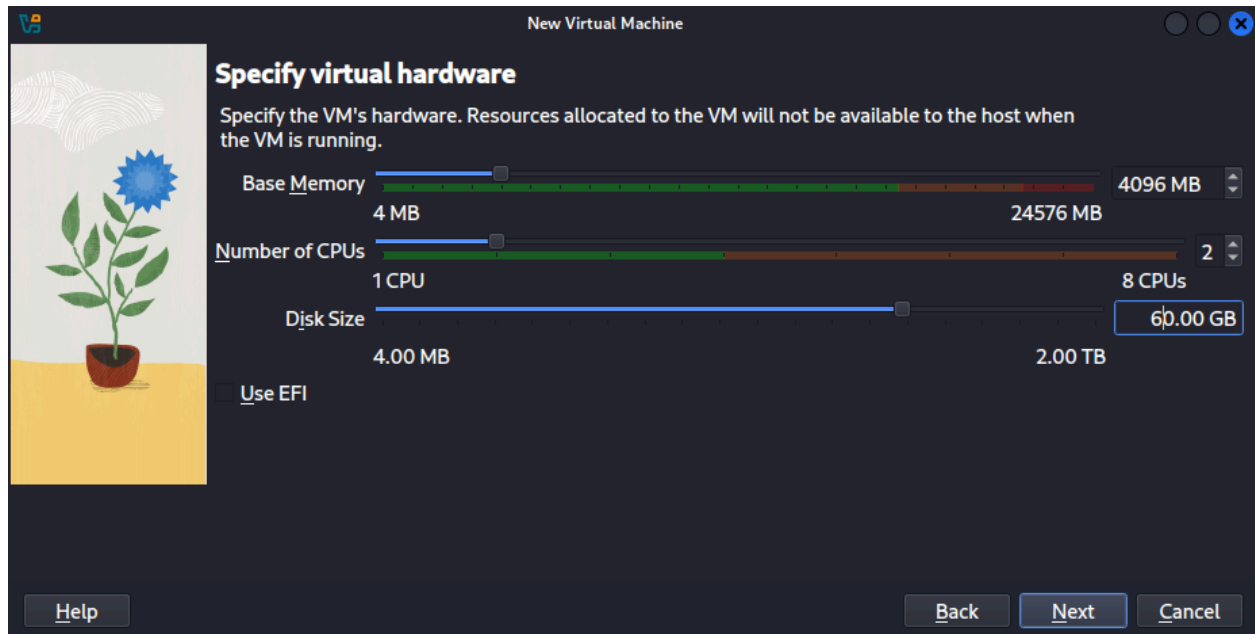


I filled it with this information and clicked next:



VM name is DC01 (easy, domain controller 1), I put its location in the active directory folder within a new VMs folder in parent folder (make it). I attached the downloaded ISO, selected the OS edition as the “datacenter evaluation desktop” option. For OS of course Microsoft Windows with the matching version (server 2019 64-bit). Unchecked proceed with unattended installation.

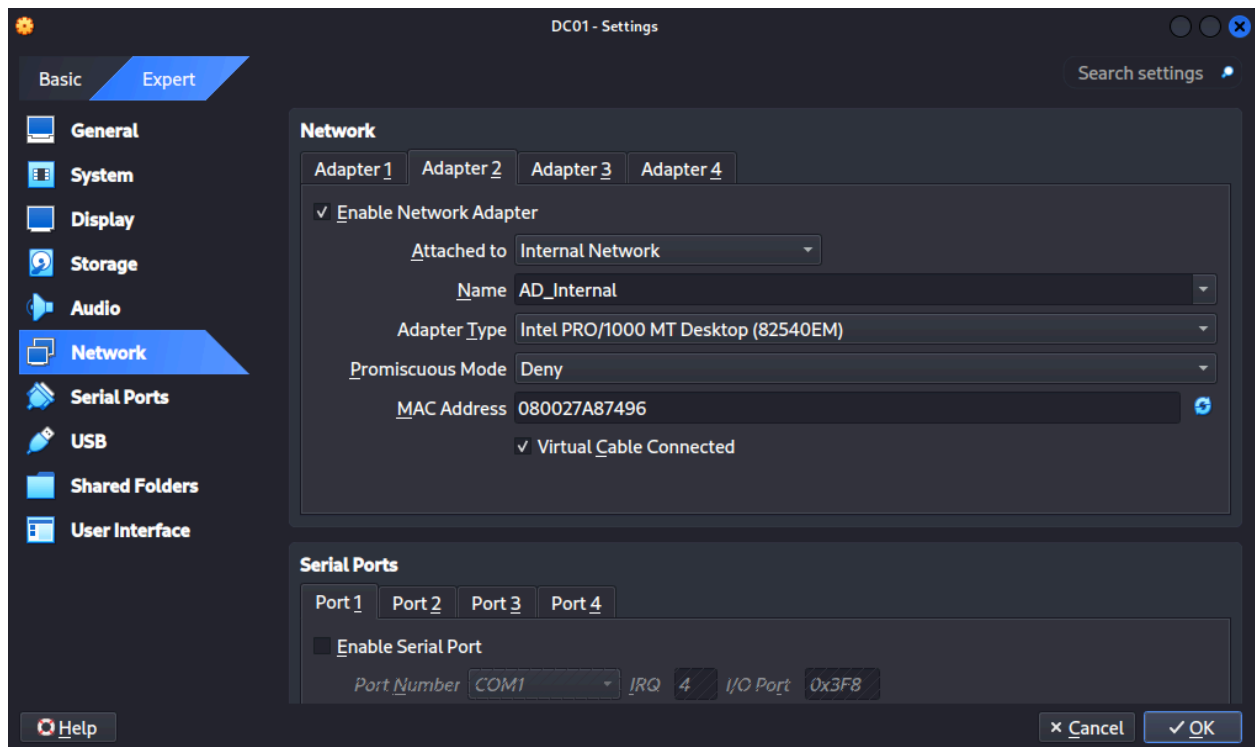
After proceeding I had to choose the hardware options.



For this I chose, as shown, 4096 MB for memory (if we have too little we’ll have a slow and unstable AD service), 2 for CPU (too many slows host and another VM I have so low is good), and 60 for Disk Size (making sure there’s space for the database, DHCP, DNS, and logs).

### Domain Controller VM Network Configuration In VirtualBox

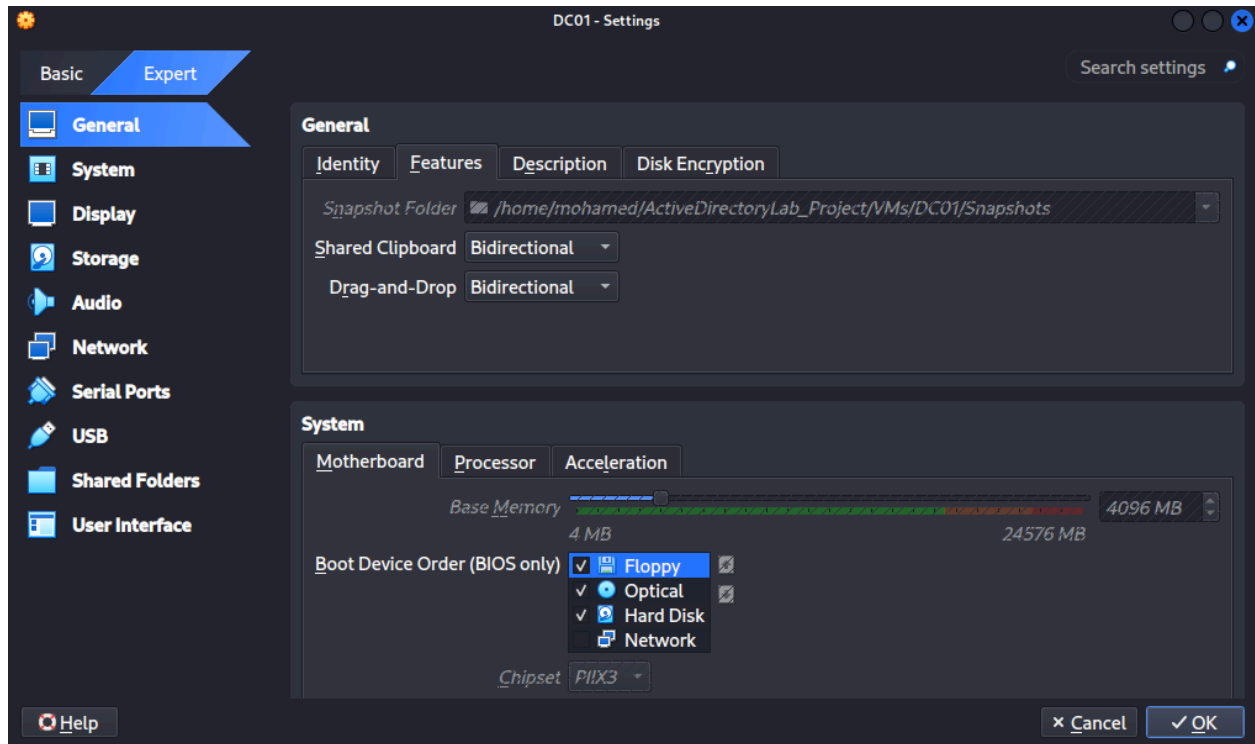
After pressing next I am asked to confirm to finish, then finish. Next we open the settings on our DC01 vm and go to Networks and change from basic to expert view. For Adapter 1 I keep the default checked on enable and attached to NAT. This allows the server to have an internet connection. For adapter 2, the private network where the domain, DHCP, DNS, and client machines will communicate. Here I had the following options as shown:



I checked enable network adapter, then attached it to Internal Network for the purpose of serving as the local active directory LAN.

### VM Experience Improvement (Bidirectional Clipboard & Drag-and-Drop)

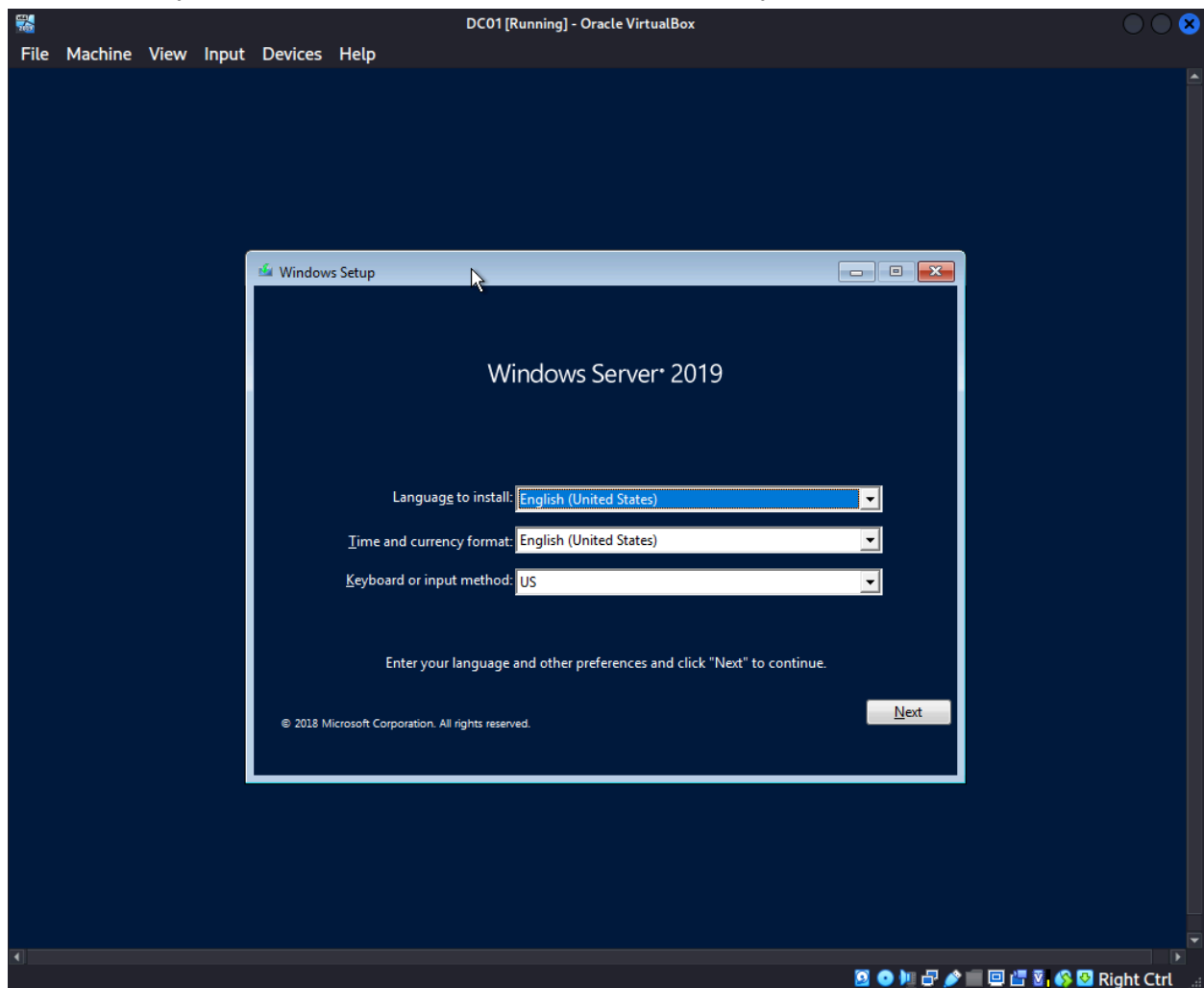
Click settings again on the DC01 vm and this time (still in expert view), in General, go to features and switch shared clipboard to bidirectional so I can use the clipboard between the VM and my personal machine and drag & drop to bidirectional so I can do it as well between both the VM and my machine.



Click ok and then the vm is ready to start in virtualbox.

## Starting Domain Controller

Start the newly created DC01 VM in VirtualBox. This is what you see:

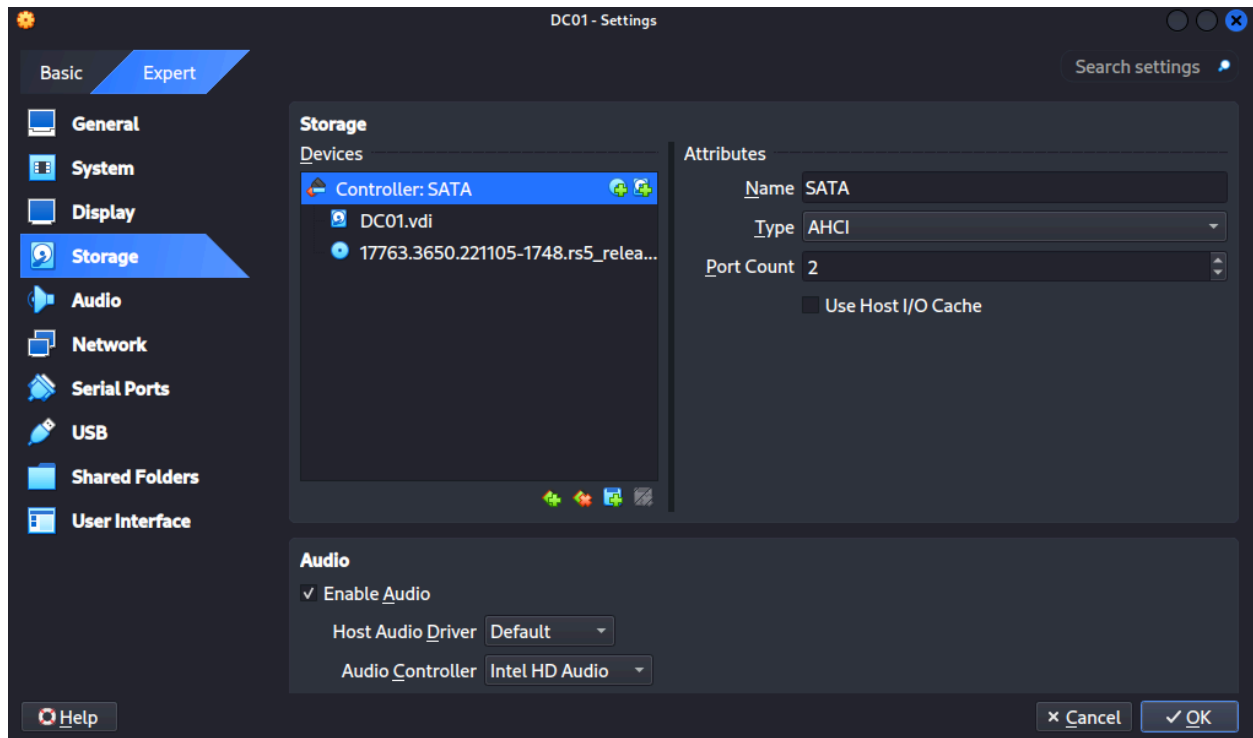


After clicking next and install, choose the Desktop Experience option for the operating system selection. I initially put the Datacenter option when I was setting up the VM, the video chooses Standard so I'm choosing this now but ideally during the VM set up for the ISO and new VM set up in virtualbox for consistency. For the installation type after we can put custom since this machine on this VM is being built from scratch right now, no prior version exists. The installation will take a second, leave it alone while it does its thing, it will restart a couple times so wait until the Windows Customize settings page. For this project we'll just use the password "[Password1](#)" (without the quotes, also this is a terrible password if it wasn't for a lab project on my own network). When you press finish and you see the ctrl-alt-delete page, since the VM makes doing this from your actual computer keys not work, we'll go to the toolbar on the VM in input, then click "insert ctrl-alt-delete". After signing in everything will load up to a normal Windows machine, but we need to make the VM more interactable with our own view so we are going to go to the VM toolbar again and click "Insert Guest Additions CD Image...". This will ask you to Download something so just proceed. Go to your file explorer, then find the VBox CD image. My

personal environment and network have a complicated network configuration so doing the “Insert Guest Additions CD Image...” didn’t properly install the image.

## Adding VBox Guest Additions (For Improved VM Handling) to DC Manually

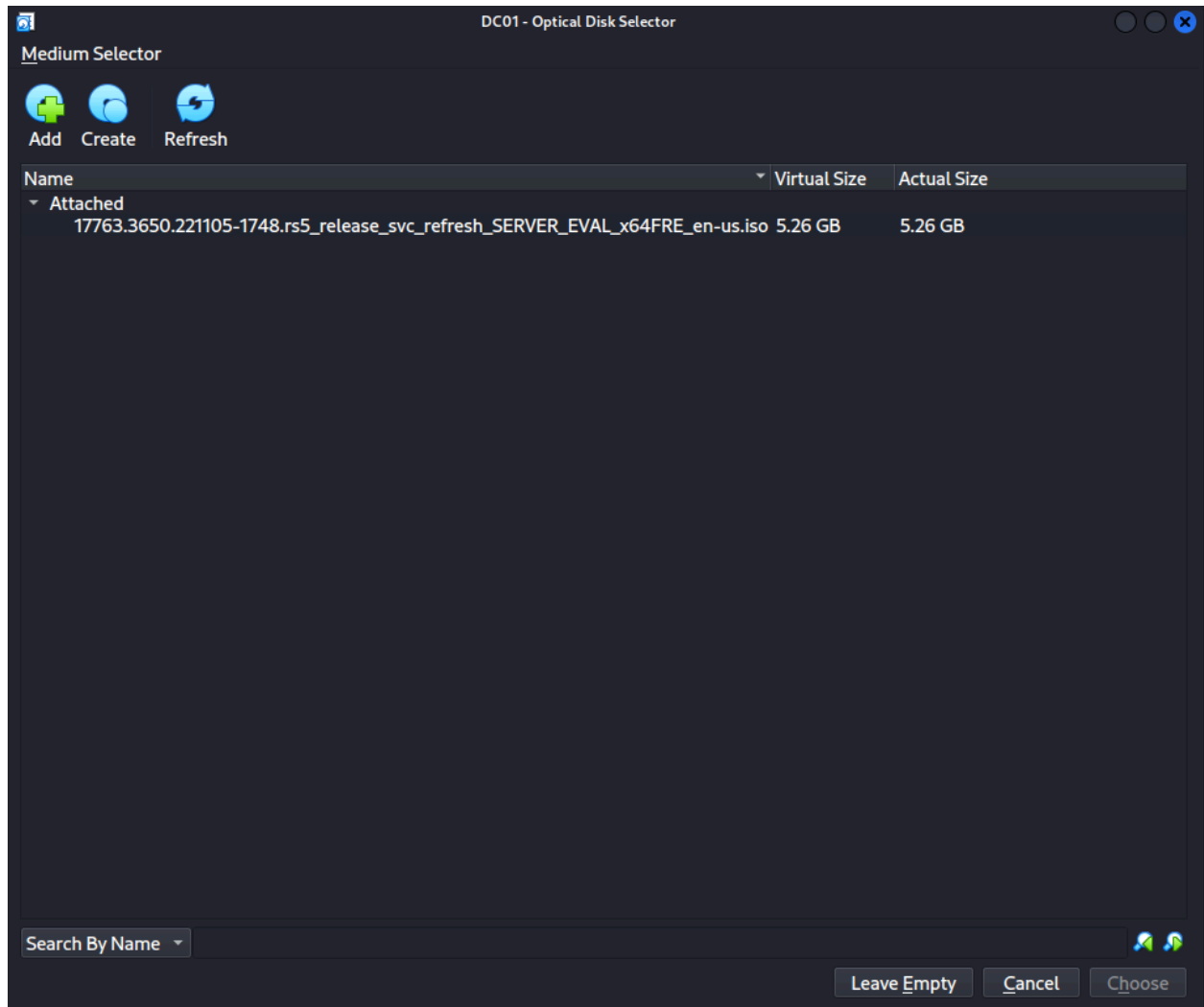
Since I suspected why, I decided to install the image and move it on my machine. So let’s proceed with this little detour in the setup just to show how to install the guest editions to get around the complicated-network issue on my end. First, shutdown (power off) the VM, then go to Settings in VirtualBox for the VM just like in the picture:



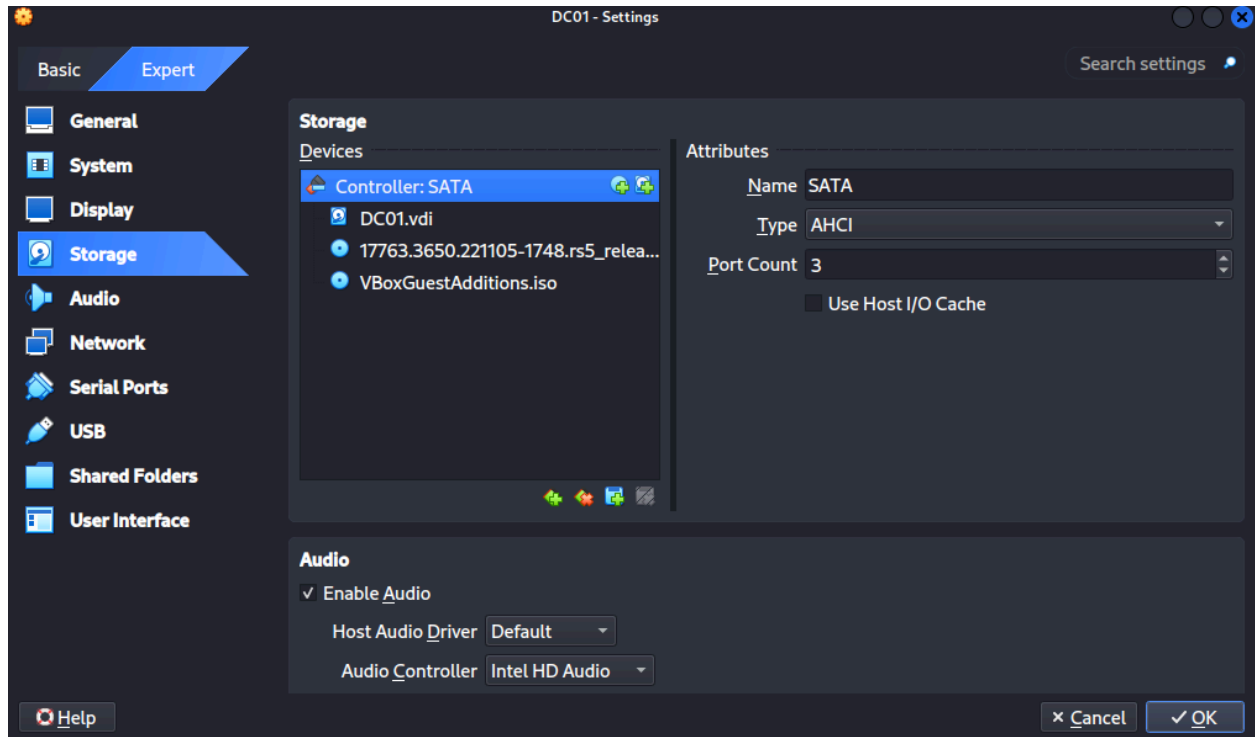
We have to also go to a terminal window on our computer (not the VM which should be powered off now) to install the VBox ISO image with this command:

```
wget https://download.virtualbox.org/virtualbox/7.0.20/VBoxGuestAdditions_7.0.20.iso -O  
~/VBoxGuestAdditions.iso
```

After this finishes, go back to the VM settings, select the blue disc with the green plus sign next to “Controller: SATA”. That will bring you to the next picture:

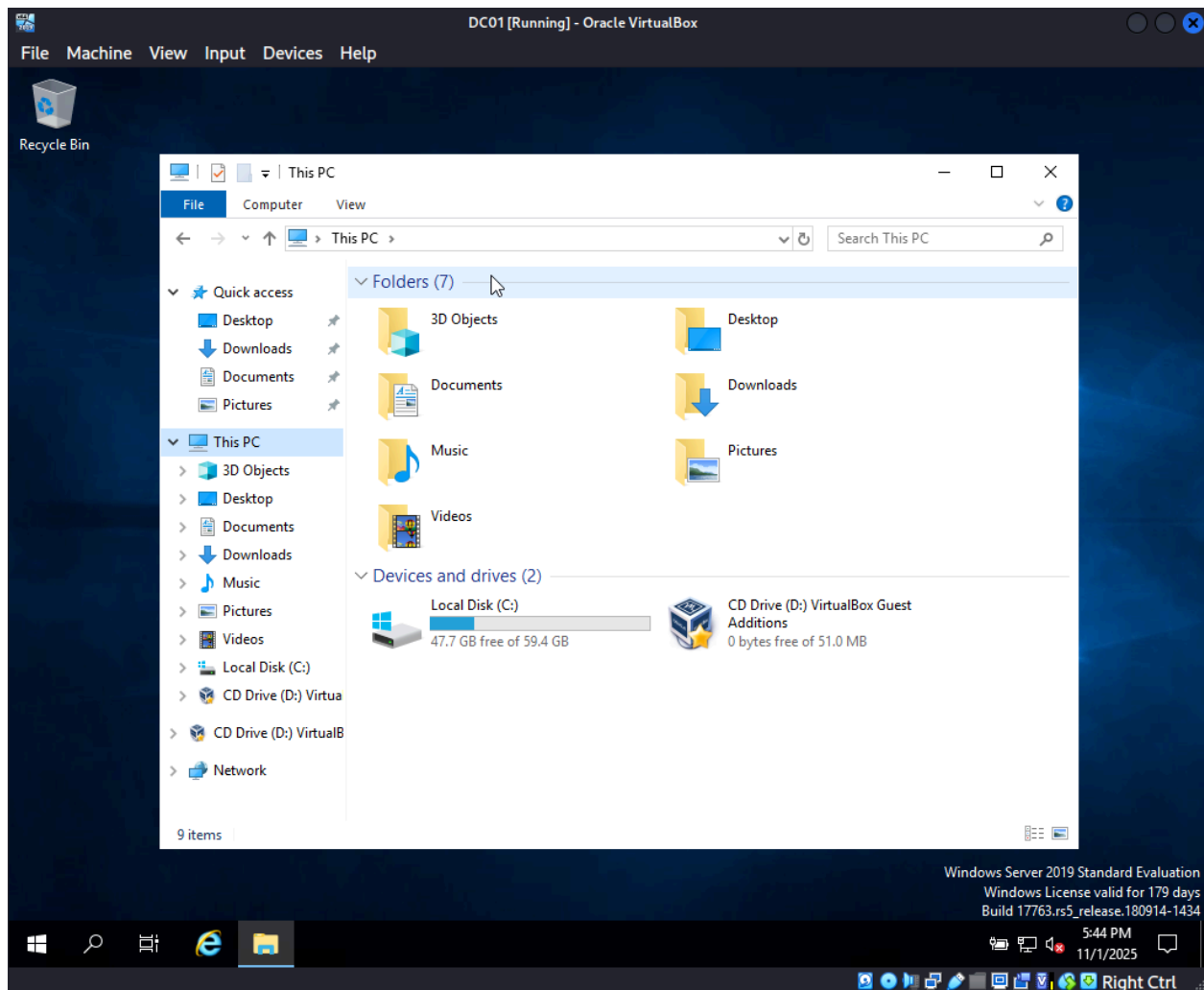


Here we have to click add again, then we find the VBox image that we installed with the terminal command on this machine.



After choosing our ISO, we see here how the guest ISO is listed under devices. We can also select the other CD image that starts with 17... and remove it just to keep everything clean.

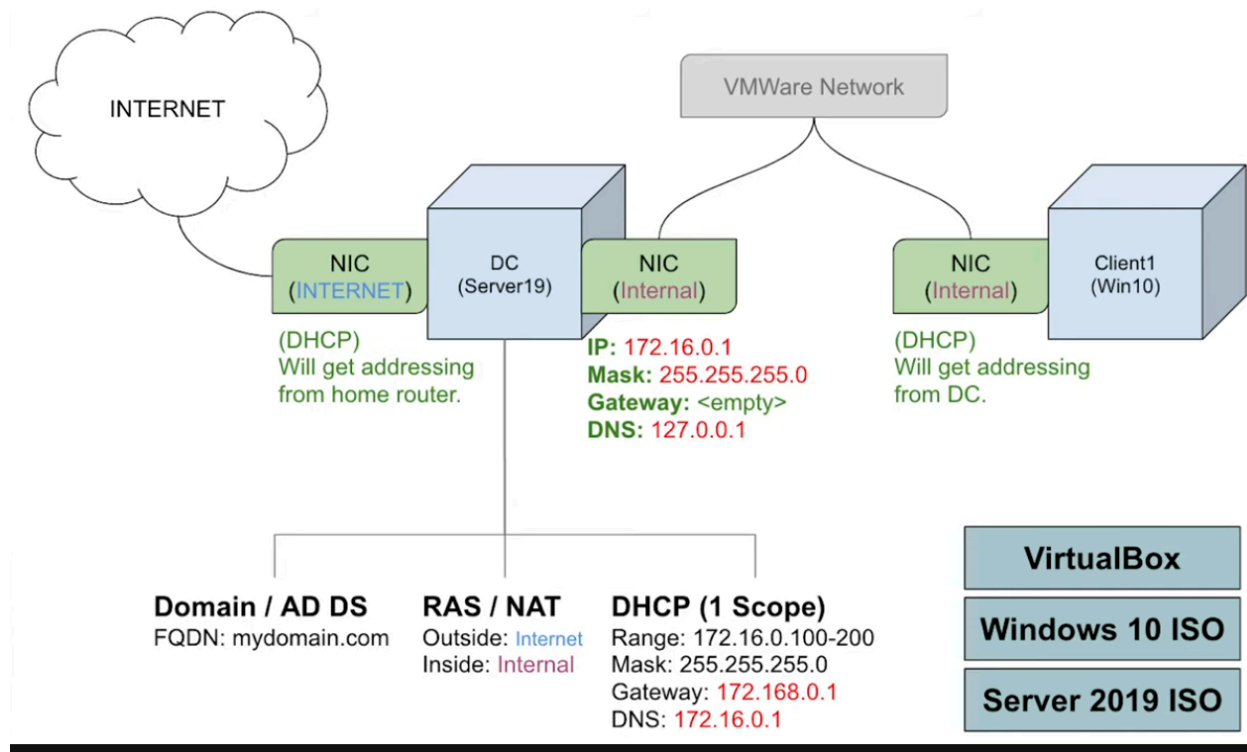
Once we start the VM again, we put in our password and check the file explorer again just to see if we have the required CD drive with the name VirtualBox Guest Additions in “This PC”:



Since we see it, we go double click it, then double click the amd64 file there. Proceed through all “next” and install options until you’re asked about reboot, and at this point you’ll choose the manually reboot later option, then select finish. Shut down the VM through the Windows icon, then start the VM. Log in again, the quality of the VM should have improved now.

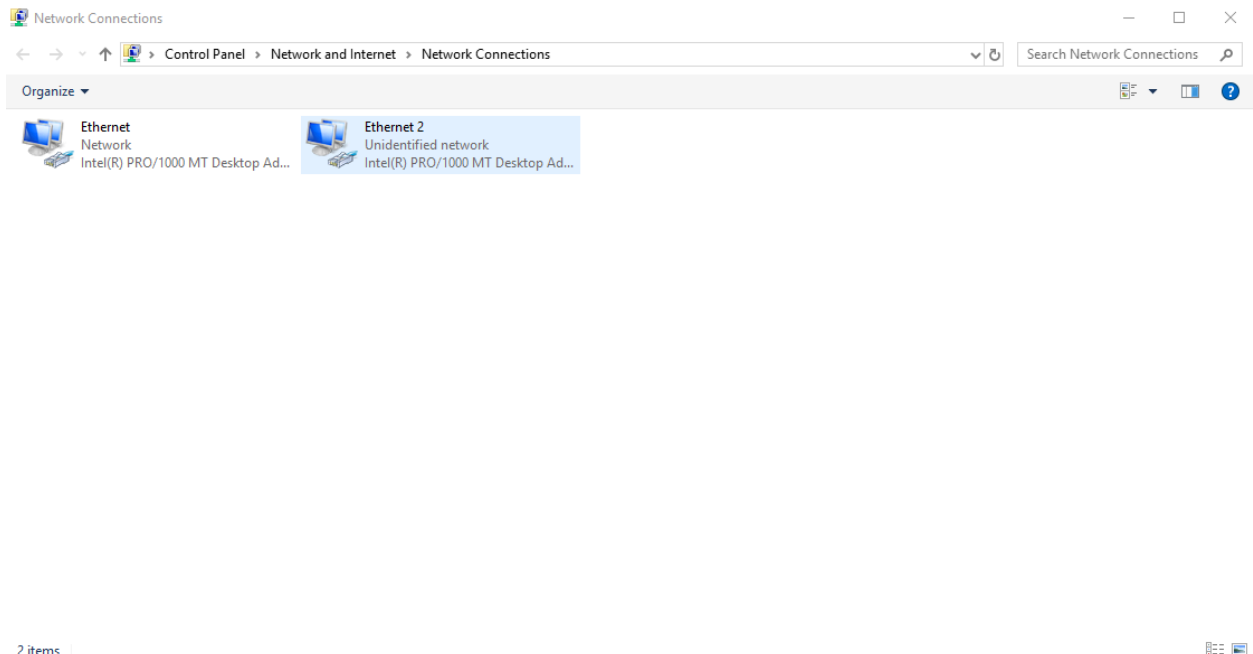
## Setup IP Addressing

Let's set up IP addressing. As you can see in this figure (made by the guide):

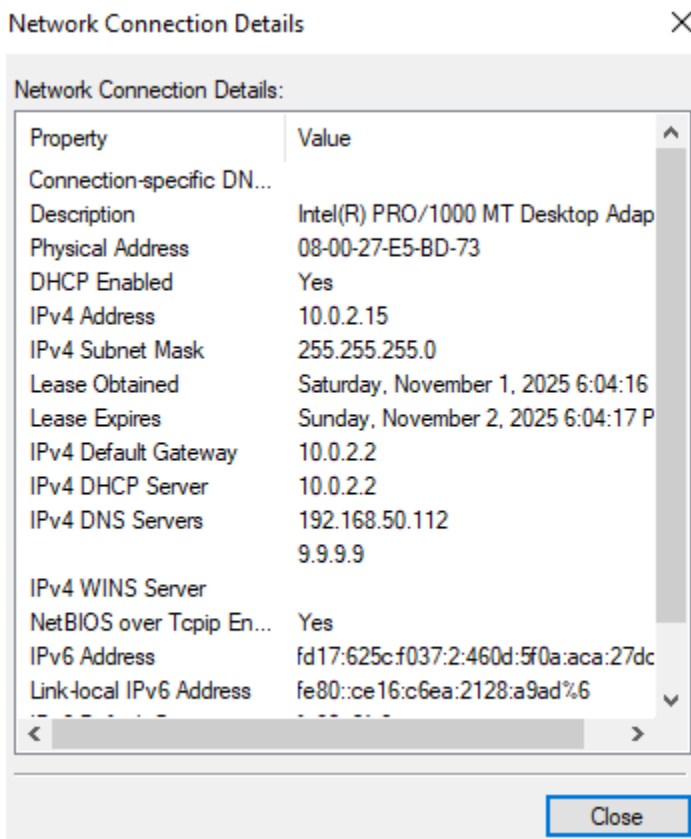


Here, we see two NICs. The left NIC (network interface card). This corresponds to the adapters on the VM, one for the external connection that connects the VM to the internet on my computer, and an internal one. The external one is set up automatically but we need to do the internal set up.

First click on the network tool on the bottom right bar of the windows VM screen. Then click Network, then select “Change Adapter”. Here you will see this screen:

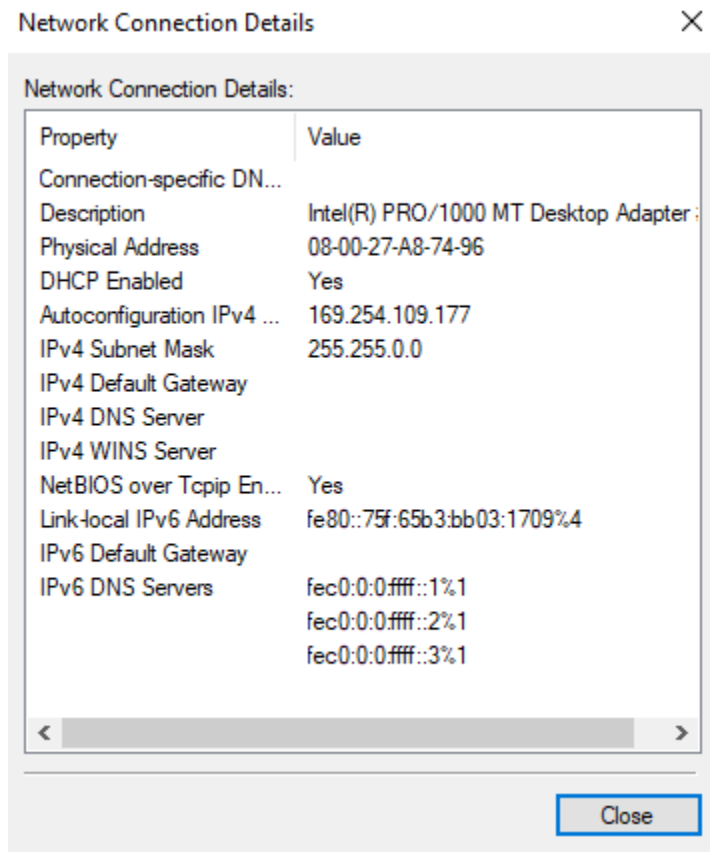


When we right click and select status on the first ethernet option, we see this:



This we can assume is the actual internet connection if we see our details here (the address space is VirtualBoxNAT 10.0.2.0/24). Because of this, we'll rename the first ethernet option and

rename it to `_INTERNET_` . By process of elimination we know the second is the internal connection but here are the details:



We can tell this is the internal connection because of the 169. IP address listed there, which we can assume means that this adapter was attempting to connect to a DHCP server to get an IP address, which it was unable to, so it received an automatic IP address. Rename it to something obvious like `X_Internal_X`. Then right click on the newly named connection and click properties. Double click on IPv4, then click "Use The Following IP Addresses". Here we will see this:

Internet Protocol Version 4 (TCP/IPv4) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

☐ Obtain an IP address automatically

☒ Use the following IP address:

IP address:

Subnet mask:

Default gateway:

☐ Obtain DNS server address automatically

☒ Use the following DNS server addresses:

Preferred DNS server:

Alternate DNS server:

☐ Validate settings upon exit

Advanced...

OK Cancel

Here we will enter the values from the diagram.

Internet Protocol Version 4 (TCP/IPv4) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

☐ Obtain an IP address automatically

☒ Use the following IP address:

IP address: 172 . 16 . 0 . 1

Subnet mask: 255 . 255 . 255 . 0

Default gateway: . . .

☐ Obtain DNS server address automatically

☒ Use the following DNS server addresses:

Preferred DNS server: 172 . 16 . 0 . 1

Alternate DNS server: . . .

☐ Validate settings upon exit

Advanced...

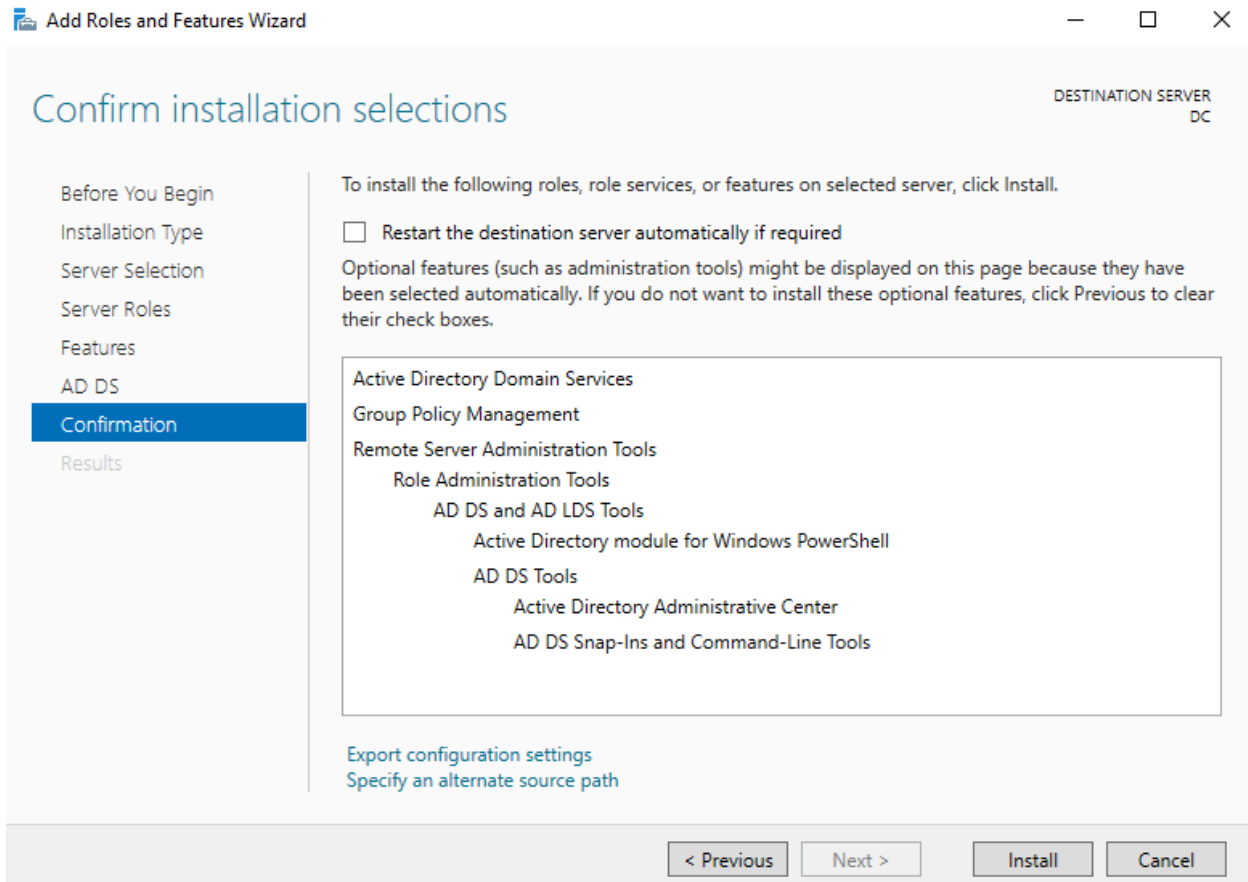
OK Cancel

We could also put the loopback version for the preferred DNS server but here I won't. Now we have the NICs both configured from the diagram.

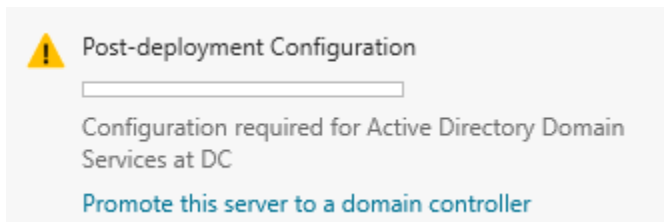
Next we will rename the PC. Right click the start menu then System, then Rename This PC. Here we will just name it "DC" for Domain Controller. Click next then restart.

### Installing Active Directory Domain Services & Creating a Domain

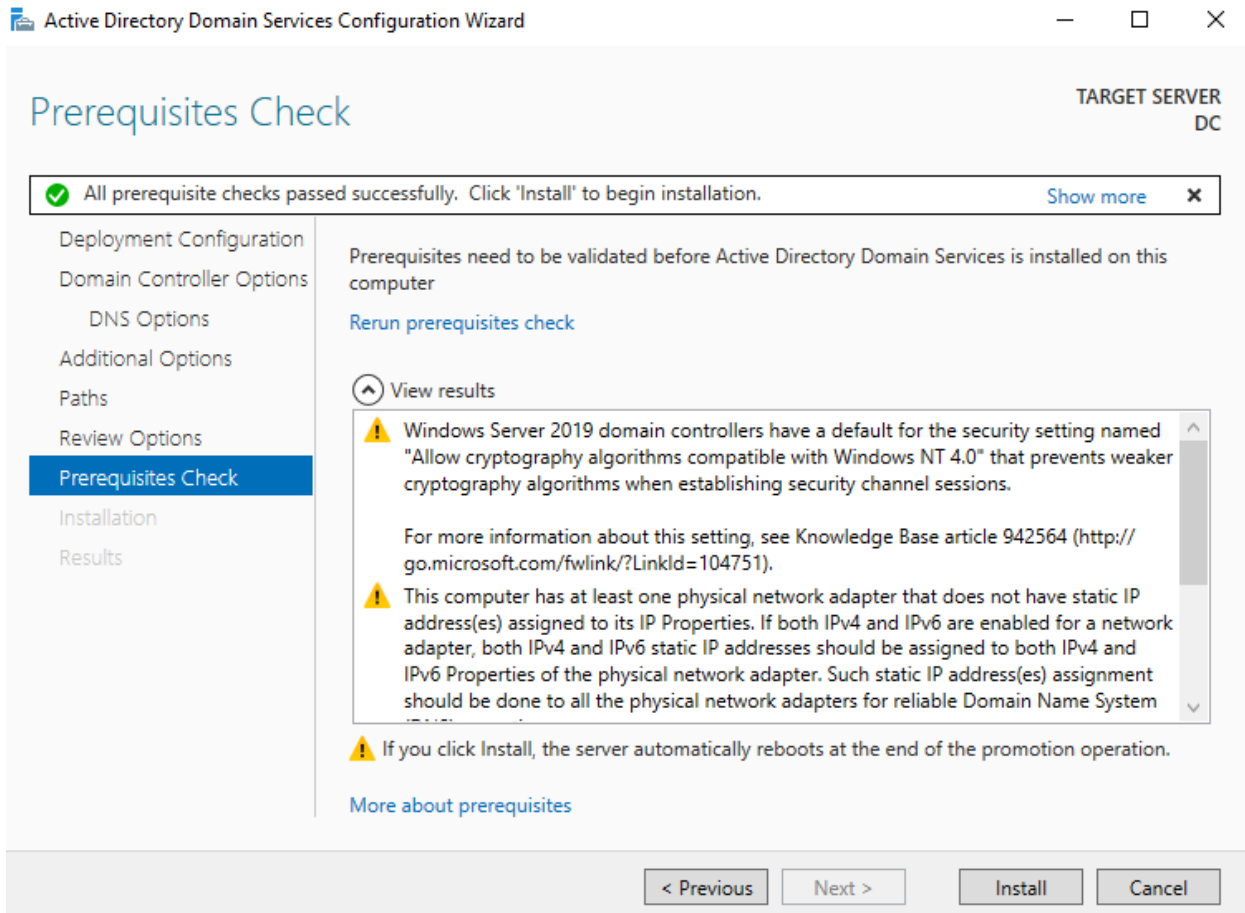
Now we can install Active Directory Domain Services (AD DS), and create a domain. We have to have Server Manager open here and then we can go to Add roles and features. Click next until "Select Destination Server", make sure our "DC" is selected, then double click Active Directory Domain Services in the long list, click next, then next again until you see install. You should be here now:



Click install and let it run as it will take time. The role is now installed. Go to the Server Manager notifications and click the blue link you see here:



After that the wizard will pop up again, and you should select Add a new forest, then in root domain name we'll just call it [mydomain.com](#). Then click next and add the same password again in Domain Controller Options. Then proceed next all through until you see this screen:



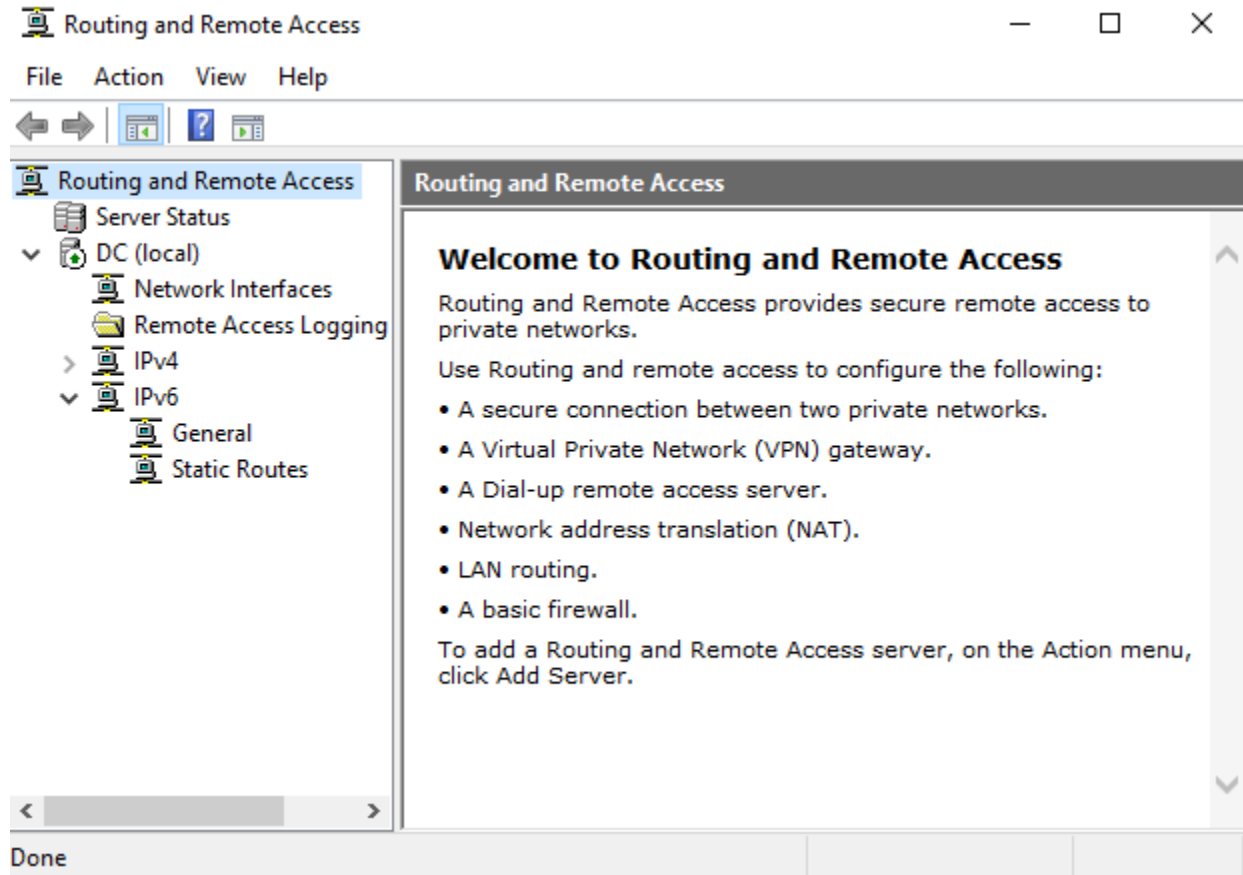
At which point you will install and it will restart. This will take a second.

To create our own dedicated domain admin account instead of the built in administrator account, we can click Start->Windows Administrative Tools->Active Directory Users & Computers. When the console opens right click [mydomain.com](http://mydomain.com)->New->Organizational Unit. Name it \_ADMIN, uncheck the protection box about deletion, then in the newly created \_ADMIN add a new user by right clicking, New->User. Fill in first and last name and then for logon name, just to follow the video I put a-meljerary. Add the password we've been using (Password1) and click never expires, then finish. Right click the new user and click properties, then click the 'member of' tab, then click add and type "domain admins". Click check names, it finds it, then ok->apply->ok.

## Configuring RRAS For Routing and Network Address Translation (NAT)

Sign out at the start menu, and now login but instead of logging in as administrator as usual, this time click other and type the username (the logon name mentioned earlier). We now have to add the RAS/NAT (remote access server/network address translation) to route traffic from the internal AD network to an external network and perform Network Address Translation (NAT). On server manager, click "add roles and features" in the dashboard. Then proceed next on the first screen and next on role or feature based installation, next on our server, select the remote access role and click next, ignore the features and click next. When you see role services click

routing, click add feature when it pops the console up, then proceed all the way until you see the install to click. Close the console when it finishes the install, then click tools in the dashboard then “routing & remote access”. Right click DC and click the top “configure” option. Click next on the wizard then select NAT, then select the \_INTERNET\_ interface option that we had named earlier. Now NAT should be up and running as you can see here:



## Installing and Configuring DHCP (Scope and Authorization)

Now let's set up our DHCP server on our domain controller. On the server manager dashboard click add roles. Proceed “next” until you see roles. Select DHCP server and then “add features” and then next all the way to install. When it finishes, in server manager click tools, then DHCP. The purpose of DHCP is to automatically assign IP addresses to computers on this network. We need to set up the scope which we can refer to the diagram for.

Here we have to right click IPv4, select “new scope”, then for the name, let's just put the range which is 172.16.0.100-200, with then the range at start would be 100 and the end would be 200. For length put 24 which is 255.255.255.0. After clicking next we don't need to worry about excluding addresses or putting a duration on the IP (just put 8). After next, click yes on configure DHCP options, click next and at router put in the internal NIC IP which will be the DNS, click add then next. Click next on DNS server (notice already selected), then next on WINS server.

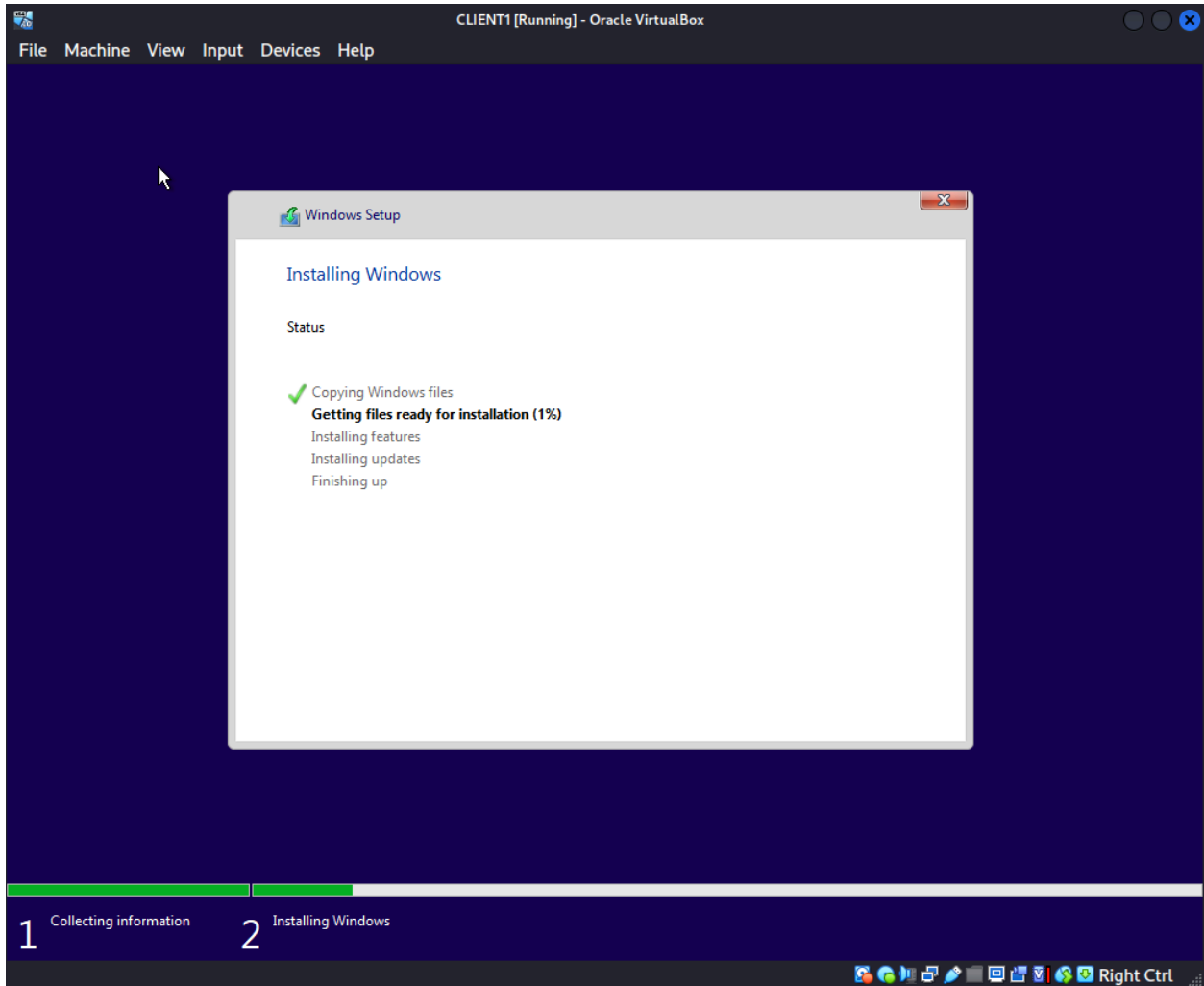
Right click [dc.mydomain.com](http://dc.mydomain.com) in the DHCP console, then click authorize. Right click it again and click refresh. Now our DNS should be working.

## **Bulk User Creation Using PowerShell**

Let's create a bunch of users with a powershell script. First, we need to make a configuration change that allows the domain controller to browse the internet (typically don't want to do this in the field per usual policy but this is just a lab). On the dashboard click configure this domain controller, then on the "IE enhanced security configuration", switch admin and user to off. Open internet explorer, go to [https://github.com/joshmadakor1/AD\\_PS/archive/master.zip](https://github.com/joshmadakor1/AD_PS/archive/master.zip) and when it downloads click save as and put it in Desktop. Extract the file. Open the 'names' text file you see and add your name at the top as the first listed name. Then on the windows start menu right click Windows Powershell ISE, and click run as administrator. In the powershell window that opens, click open, then select the 1\_CREATE\_USERS.ps1 from the folder in desktop. First we have to enter the command **Set-ExecutionPolicy Unrestricted** and then yes to all on the following popup. Then change directory to where the script is located (here it's cd C:\users\meljerary\desktop\AD\_PS-master) then click the playbutton at the top of the powershell window to run the script. Once it finishes, we should have some users created.

## **New Windows 10 VM (Client) In VirtualBox**

Now all we have left to do is create the Windows 10 VM in virtual box. Click 'new' on the virtualbox manager, then put the name as CLIENT1 and the os as Windows 10 64 bit. Click finish then open the settings on it. Change shared clipboard and drag-and-drop to bidirectional (in features under general, also put 4096 for base memory, 2 cpu's). Then in network for adapter 1, attach it to the internal network, then click ok to close the settings console. Double click the CLIENT1 vm, and when you get the disk message, locate the Win10 ISO and select it to mount and boot. At the windows set up screen when it boots click next then install now. Click I dont have a product key at the windows activation, then select Windows 10 Pro for the operating system we want to select. Click next, then accept the terms that come up and click next, then choose custom installation, then proceed for the install.



When it restarts, put in US, then US keyboard, skip second keyboard, I don't have internet, proceed with limited setup. For your username put user, press next on password since we don't need one, switch to no on all the privacy settings and accept. Start the domain controller vm if you closed it and wait for the client to set up. Open up cmd and check the ip configuration with ipconfig. You should see:

```
C:\Users\user>ipconfig

Windows IP Configuration

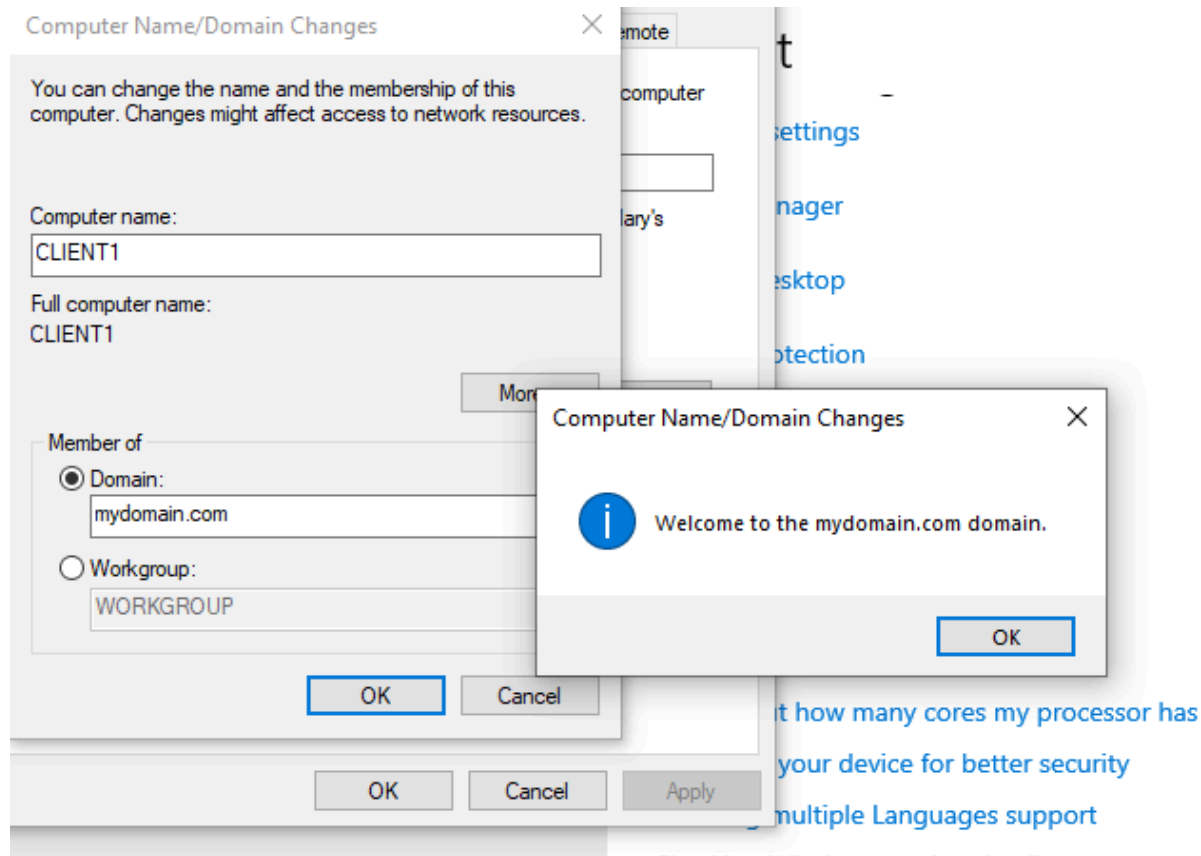
Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : mydomain.com
    Link-local IPv6 Address . . . . . : fe80::c030:92ba:e7c5:81e2%6
    IPv4 Address. . . . . : 172.16.0.100
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 172.16.0.1
```

Then test a ping to [www.google.com](http://www.google.com) and [mydomain.com](http://mydomain.com).

## Renaming and Joining a Client System To The Domain

Now let's change the hostname. If you type hostname in the command line it will be DESKTOP and some number, so right click the start menu and click system, then scroll down and click rename this pc (advanced). Click "change" in the little window, and put the name as CLIENT1, and below this for "member of", put domain then type [mydomain.com](http://mydomain.com), then ok at the bottom, then Computer Name/Domain Changes console that pops up, put the username and password of our prior admin user accounts that we made on the domain controller. You should now see this:



Click ok, then ok, then close the computer name/domain changes window, then restart. In the domain controller vm now, we can verify CLIENT1 in the AD users and computers in the computers container, and in the DHCP address leases in our scope in IPv4.

## Validating Domain User Authentication On The Client

Let's test the users we added with the script before on the client now. Login as other user in the restarted client vm, then put in the user we put in at the top of the names text file from before with is your name, and the user version is first initial last name together, and the repeated password we've been using. We should be able to successfully log in. If we open a cmd window, we should be able to identify ourselves with whoami, and we should see:

```
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.

C:\Users\meljerary>whoami
mydomain\meljerary

C:\Users\meljerary>_
```

This shows we are a member of mydomain on this computer, and we are meljerary.

## Verification and Outcomes

The following checks were performed to confirm correct operation of the environment:

- Active Directory Domain Services installed and domain created successfully
- Domain Controller configured as authoritative DNS server for the domain
- DNS forwarders configured and external name resolution verified
- DHCP role installed, scope created, and server authorized in Active Directory
- Windows 10 client received an IP lease from the DHCP server
- Client system successfully joined to the domain
- Domain user authentication verified on the client system
- Client internet connectivity confirmed through RRAS/NAT

These validations confirm that the environment functions as intended and reflects a typical small Active Directory deployment.

## Future Extensions

- Implement Group Policy Objects (GPOs) for password, desktop, and security policy enforcement
- Design additional Organizational Units (OUs) for role-based administration
- Add a secondary Domain Controller to demonstrate redundancy and replication
- Configure basic security auditing and event log monitoring
- Introduce file services with NTFS permissions and access control
- Snapshot and recovery testing to simulate failure scenarios