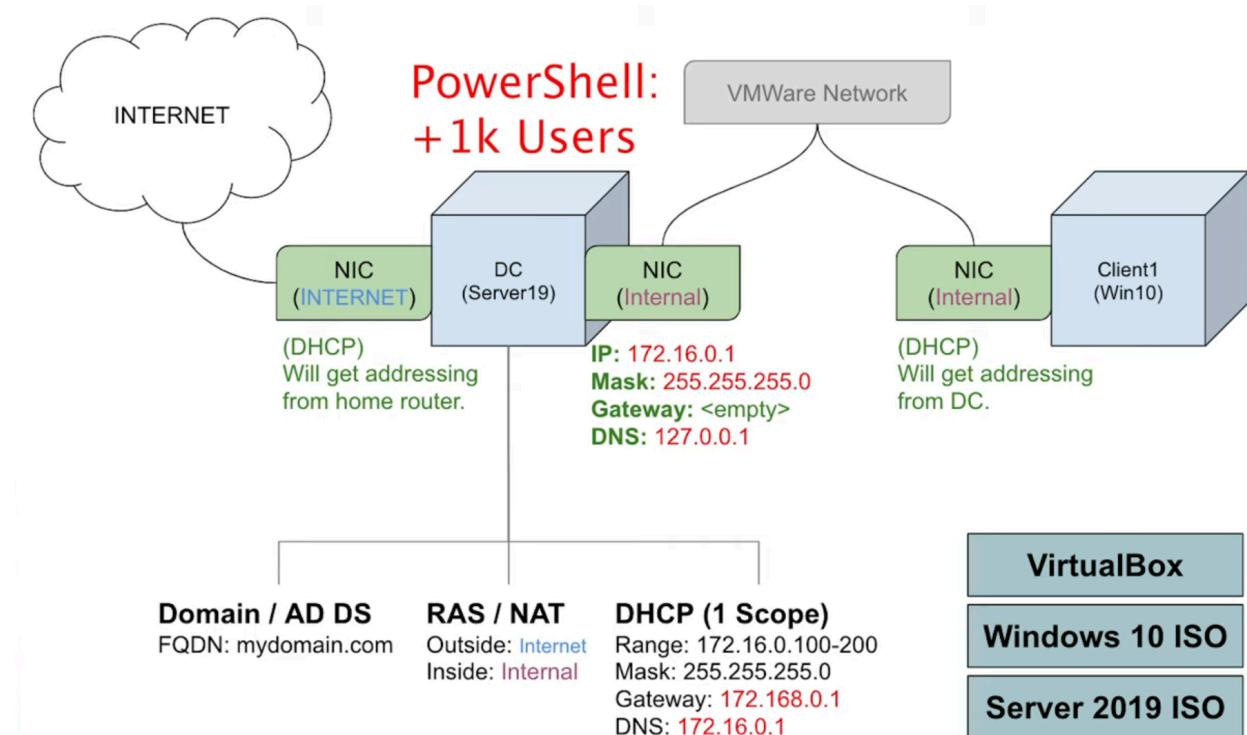


Active Directory Setup using a Virtual Environment

The Setup



VirtualBox

- Setting up a Virtual Environment using VirtualBox
- Download VirtualBox and the Extension Pack (The extension pack allows for a smoother experience)
- VirtualBox Download: <https://www.virtualbox.org/wiki/Downloads>

Windows 10 ISO Download

- This will be the Operating System used for our clients.
- Windows 10 ISO Download:
<https://www.microsoft.com/en-us/software-download/windows10>

Server 2019 ISO Download

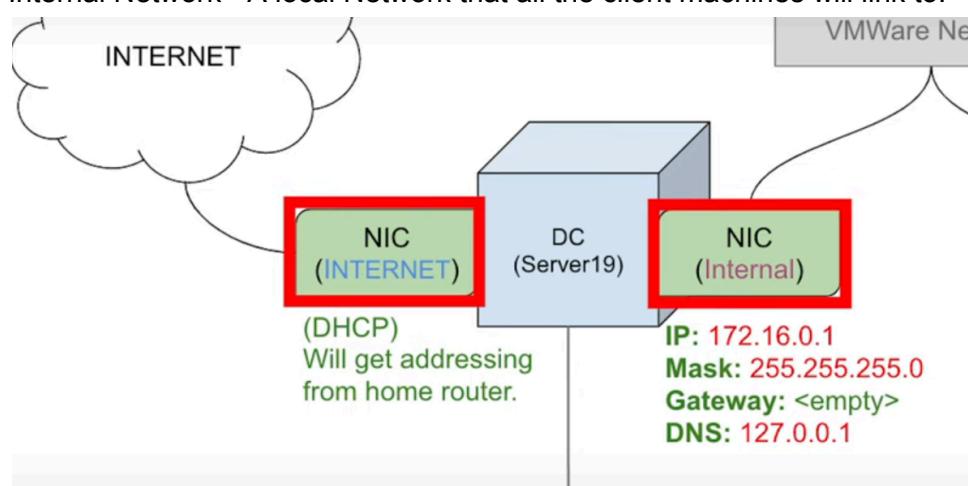
- This will be used for our Active Directory and will be our main DomainController. Basically the where all users are managed and the gatekeeper of the network

- Server 2019 ISO Download:
<https://www.microsoft.com/en-us/evalcenter/download-windows-server-2019>

Setting up Windows 2019 Server

The Initial Setup

- Setup with Windows Server (Desktop Experience) as it gives you a GUI instead of just a CMD line.
- Allocate sensible Memory, CPU & Storage
- Two networks will be made:
 - A NAT (DHCP). Will be getting addresses from our home router.
 - Internal Network - A local Network that all the client machines will link to.

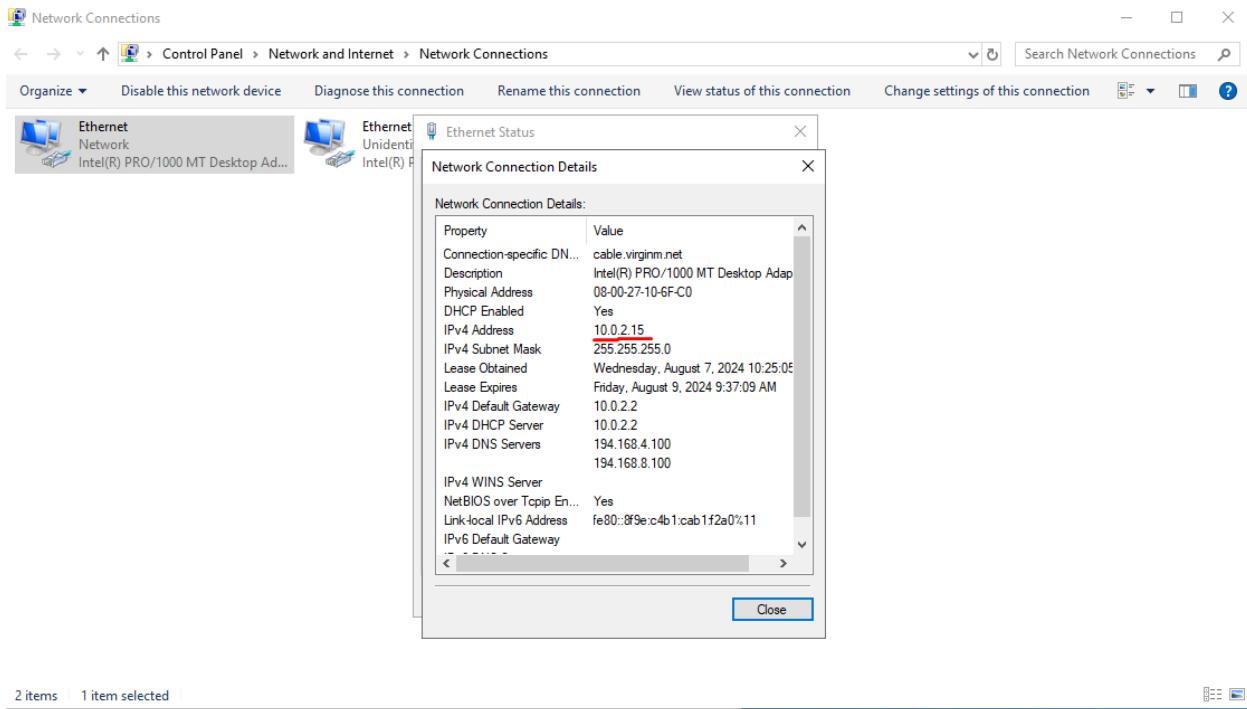


- Start Server 2019
- After Server 2019 is setup, create the VirtualBox Guest Edition (to run the Virtual Machine easier) by going to Devices>Insert Guest Additions CD Image... on your Virtual Machine. Wait for it to load then go on My Computer on your Server 2019>CD Drive (D:) VirtualBox Guest Additions>VBoxWindowsAdditions-amd64>Yes to everything and install.

Updating Network Connections

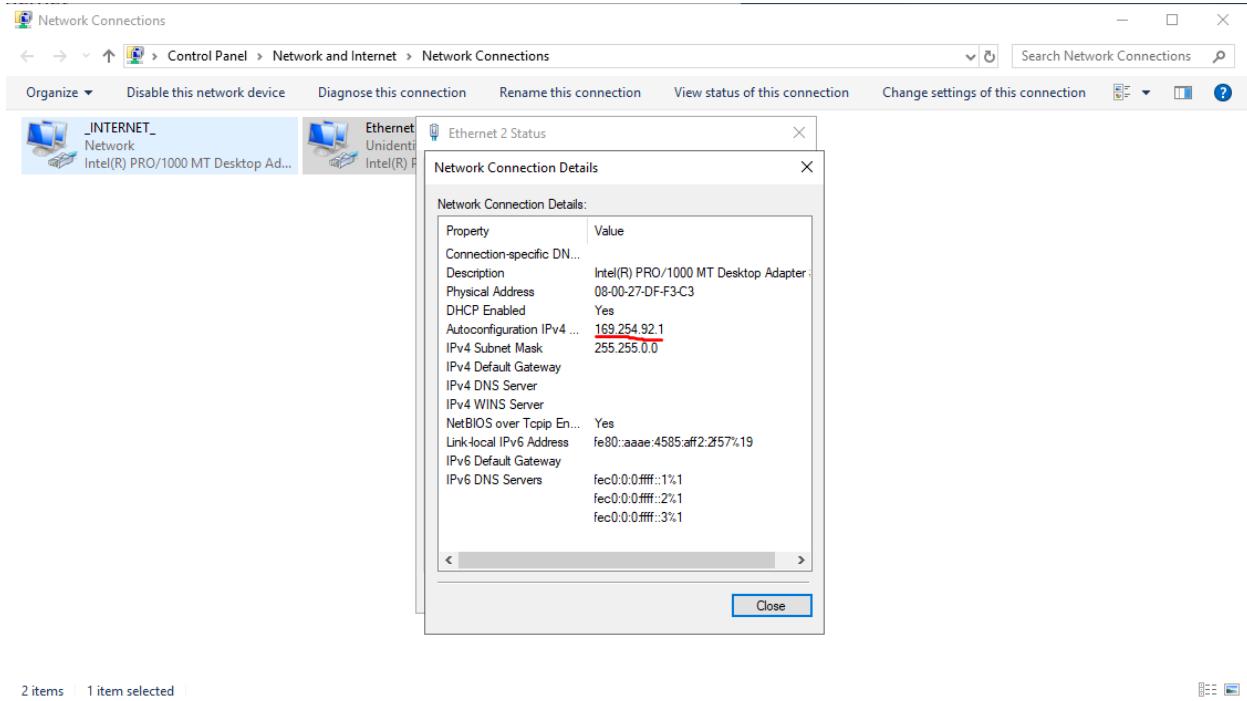
Internet Network - Home Router Connection

- Check to see which Network is which.

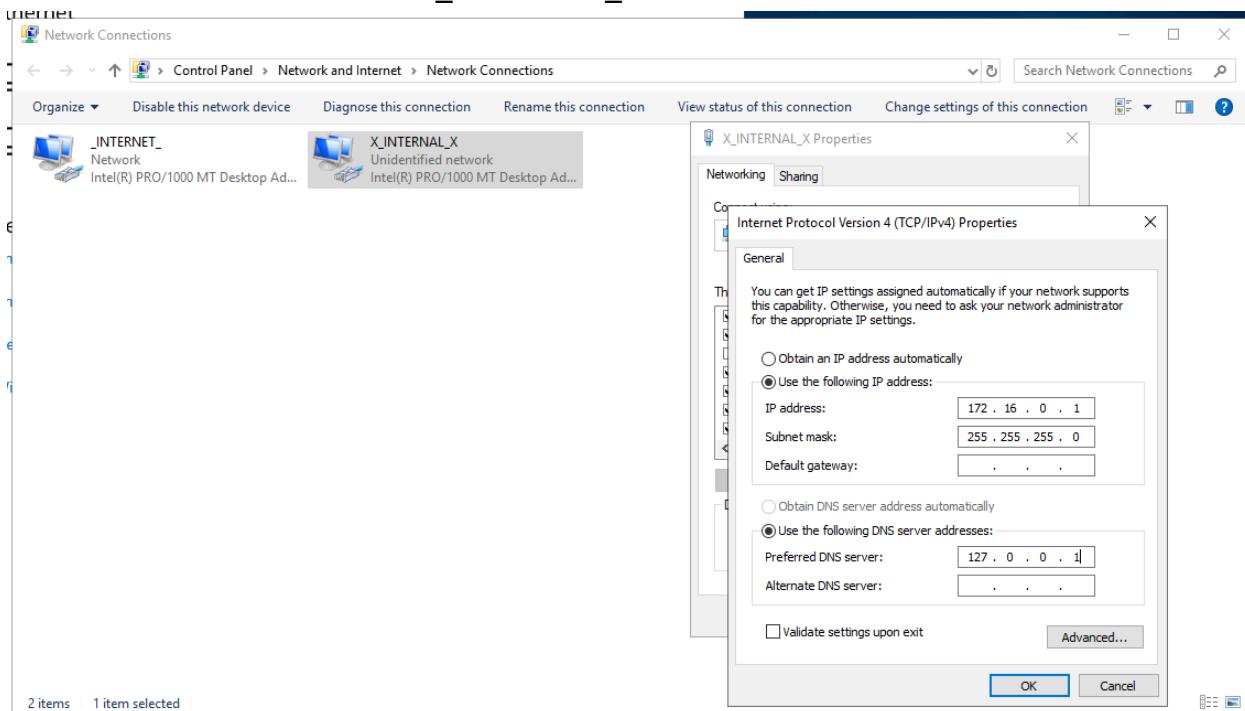


- This lets us know that this is the address from our home router.
- This will then be renamed _INTERNET_ for clarity sake

Internal Network



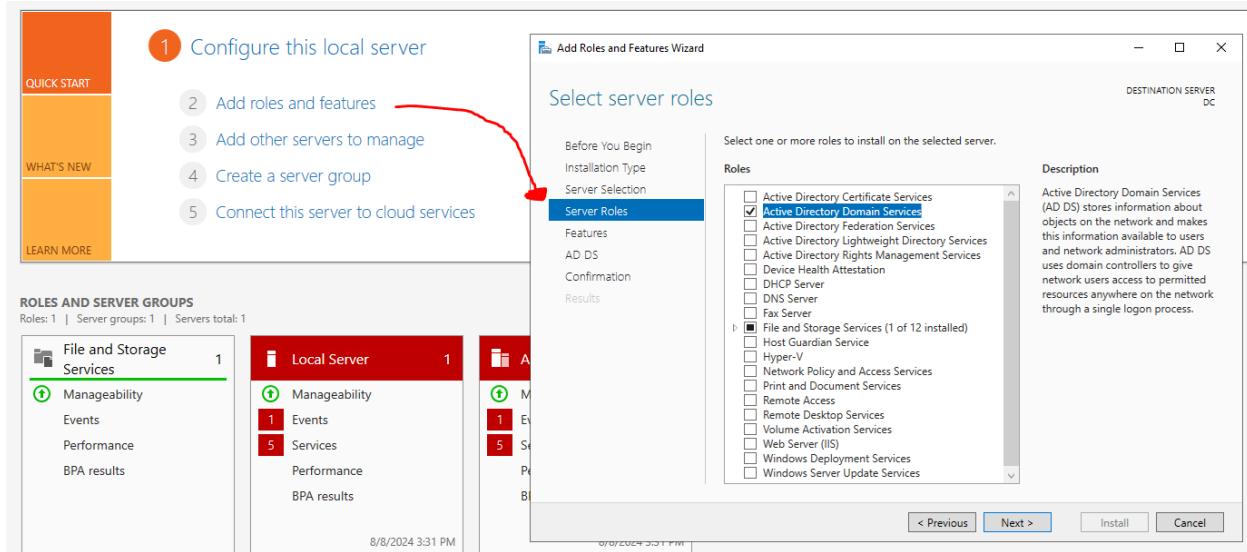
- This confirms the second network is our internal network - the network that all clients will connect to.
- The reason for the IP address given is because it was looking for DHCP server to get an IP Address from, but it couldn't find one so it used an automatically assigned IP address
- This will be renamed to X_INTERNAL_X



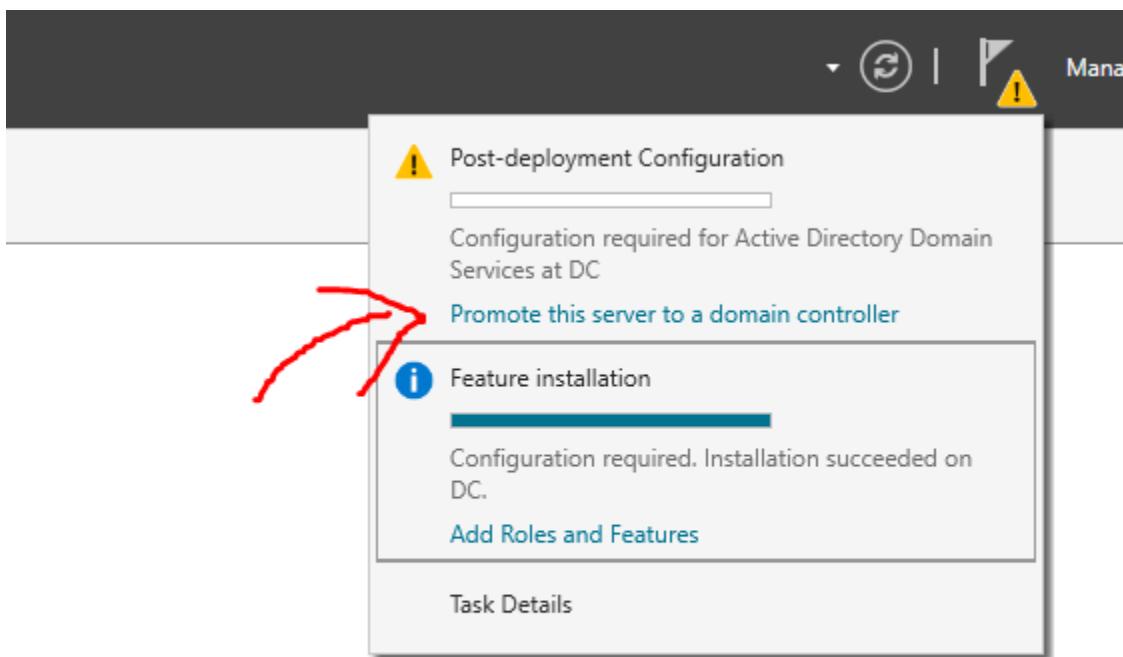
- An assigned IP address will be given instead of an automatic one. It will follow what is shown on the diagram shown under Setup.

Creation of Active Directory Domain Services (AD DS)

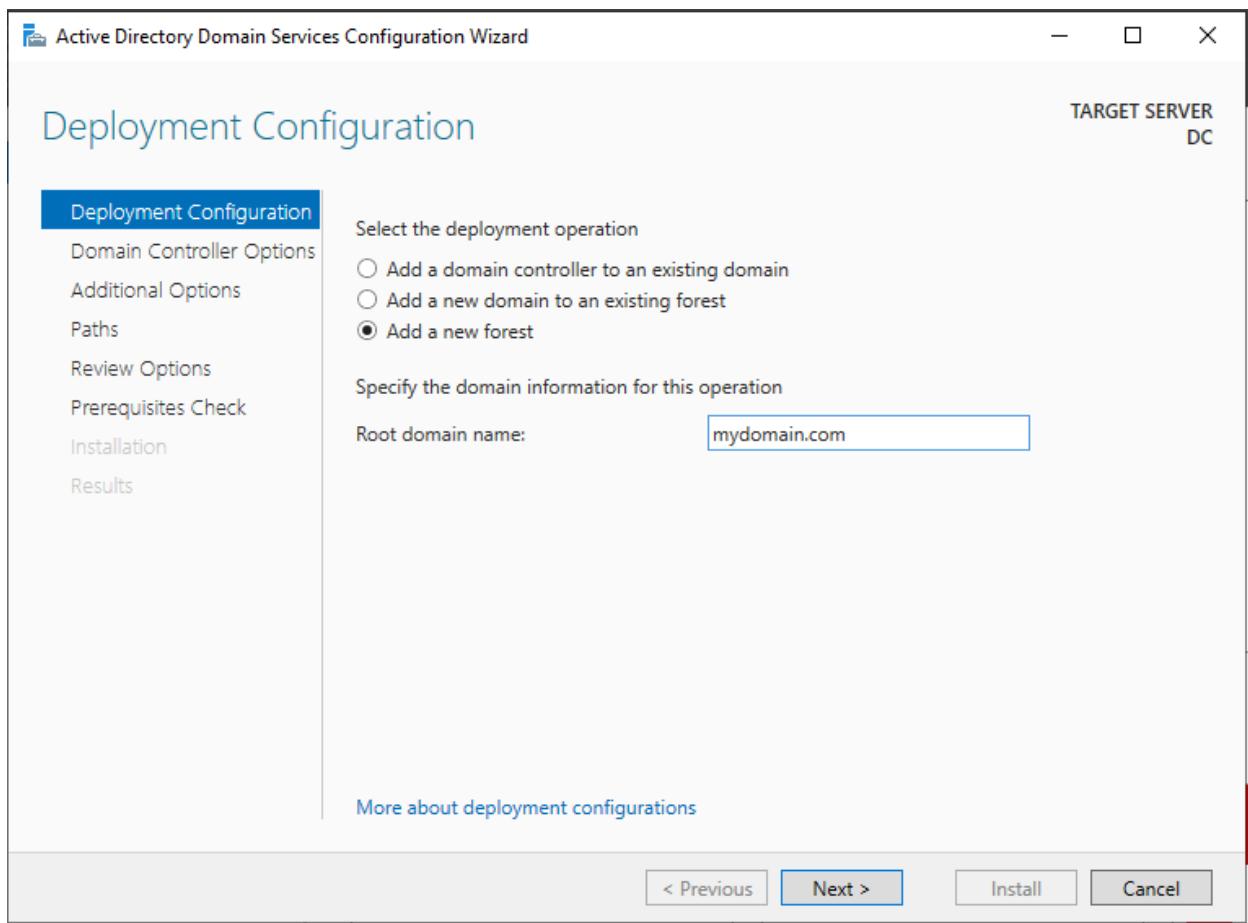
Setting up an Active Directory Domain Services



Here we'll be installing the Active Directory Domain Services



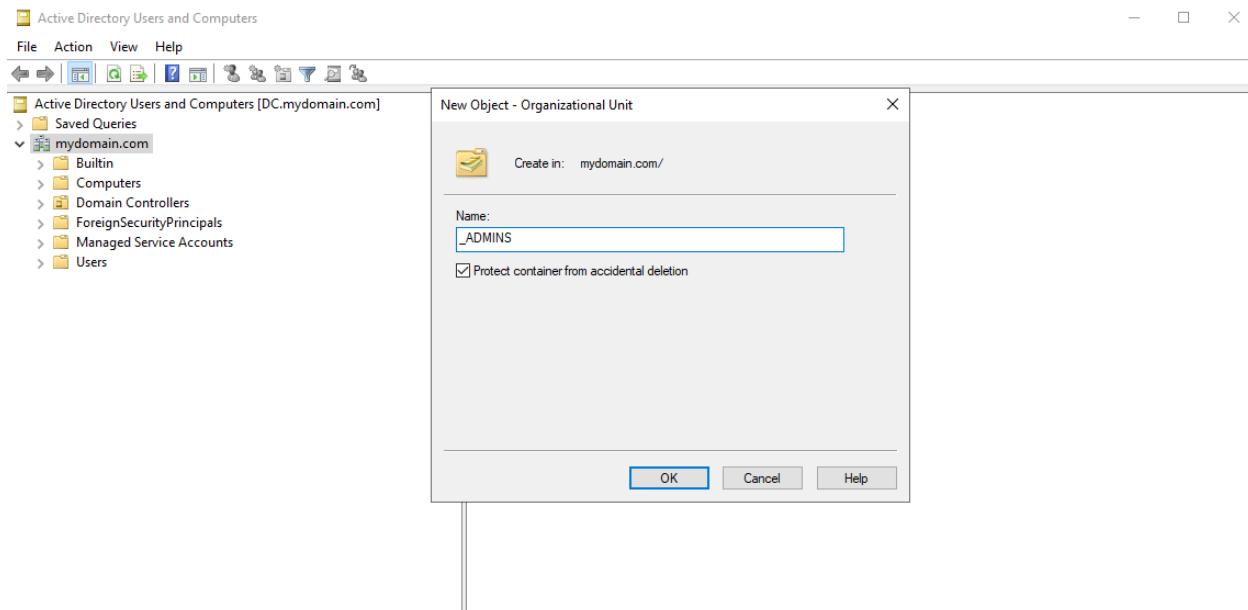
After installation, promote the server to a domain controller



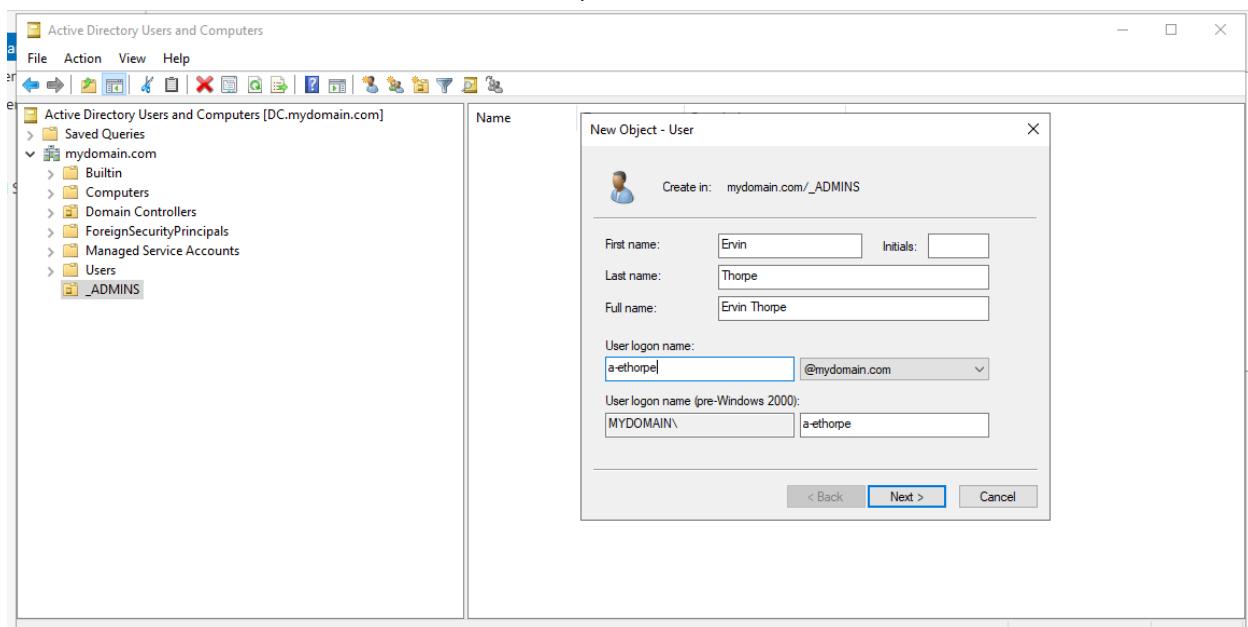
Deployment Configuration

- Add a new forest
- Root domain name: mydomain.com (can change this to what you want)
- Create a password
- Keep going NEXT until installation then restart

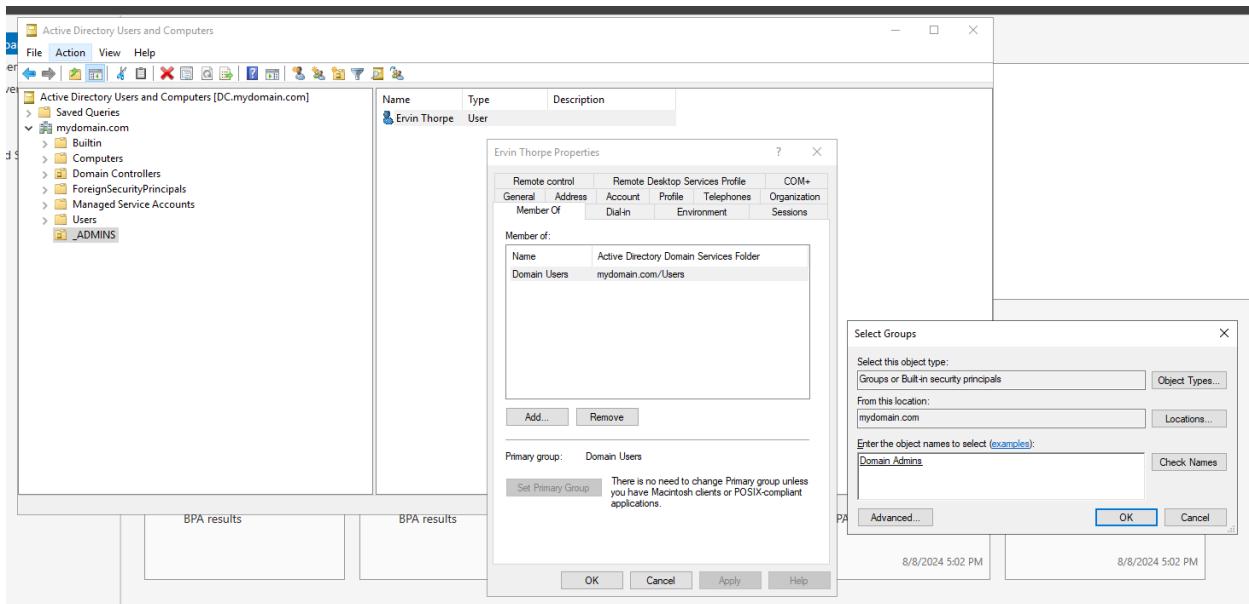
Using Active Directory Users and Computers



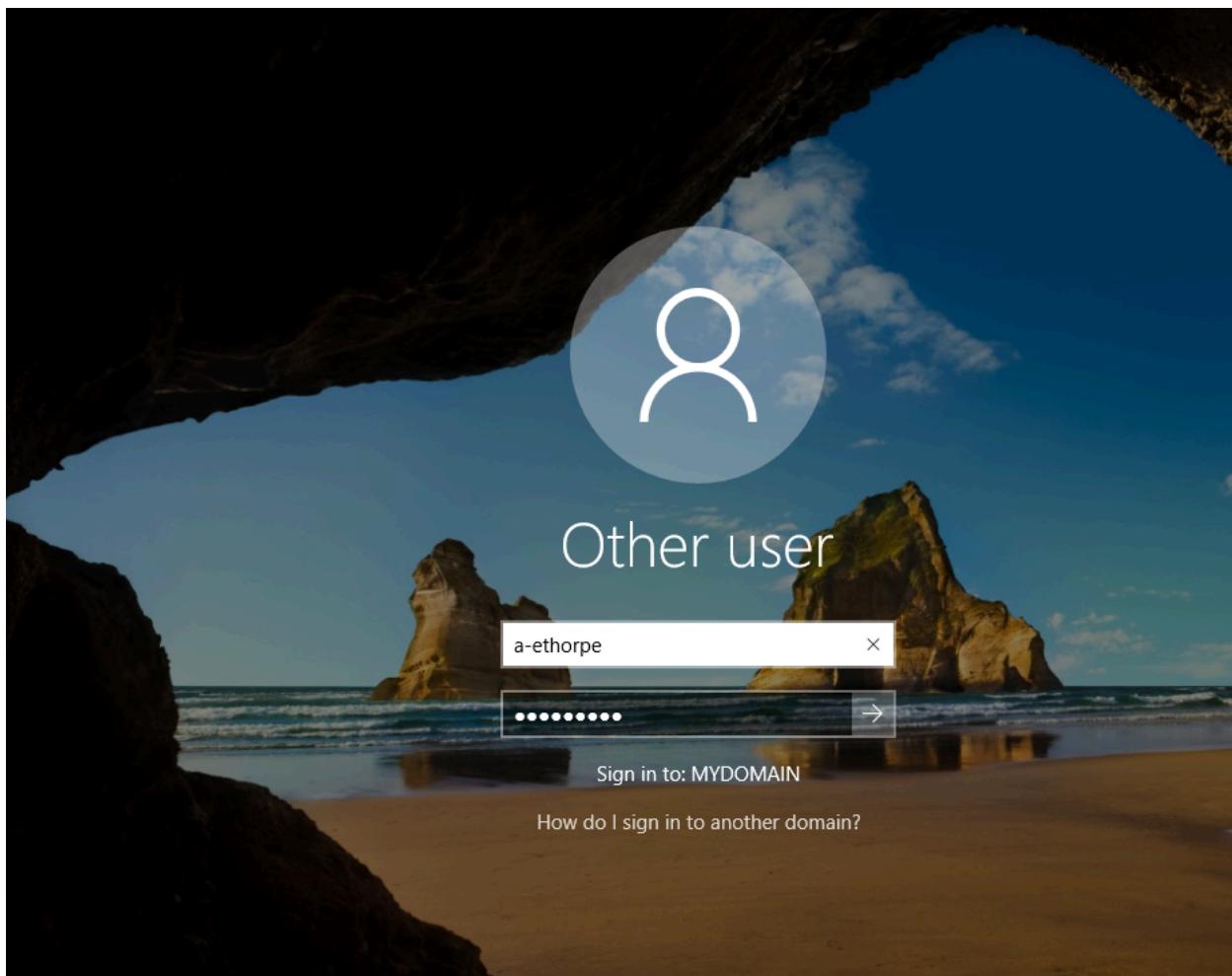
- Accessing the Active Directory - Users and Computers
- Going to mydomain.com and creating a new Organisation Unit called _ADMINS (this is where we will add our admin account)



- Setting up a new user under _ADMINS.
 - This will be myself, and we'll use the logon name, a-ethorpe
 - Setup a Password as well



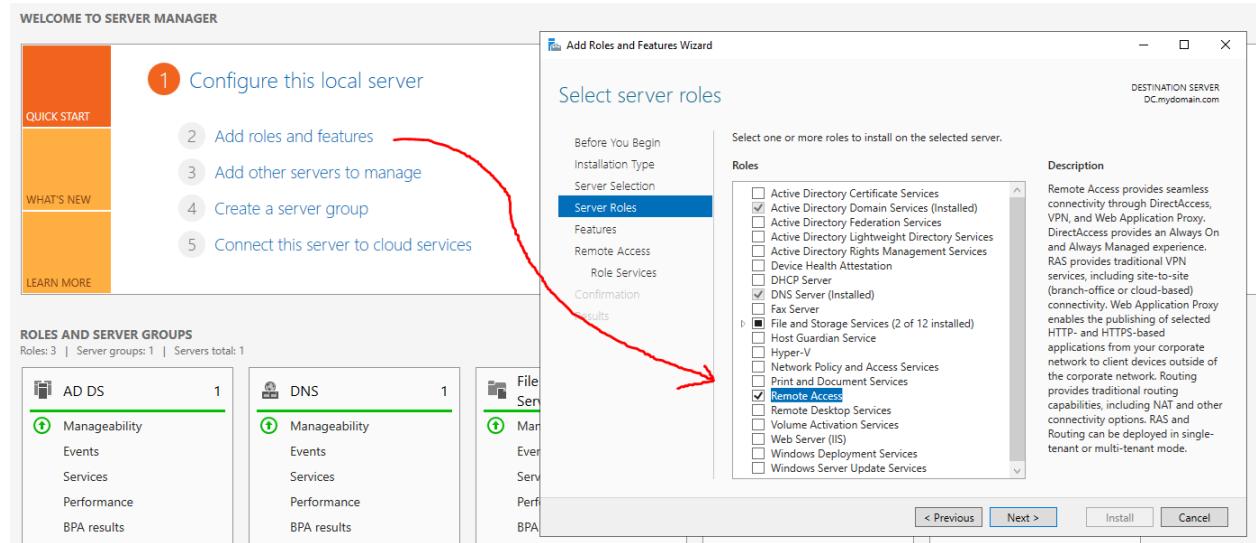
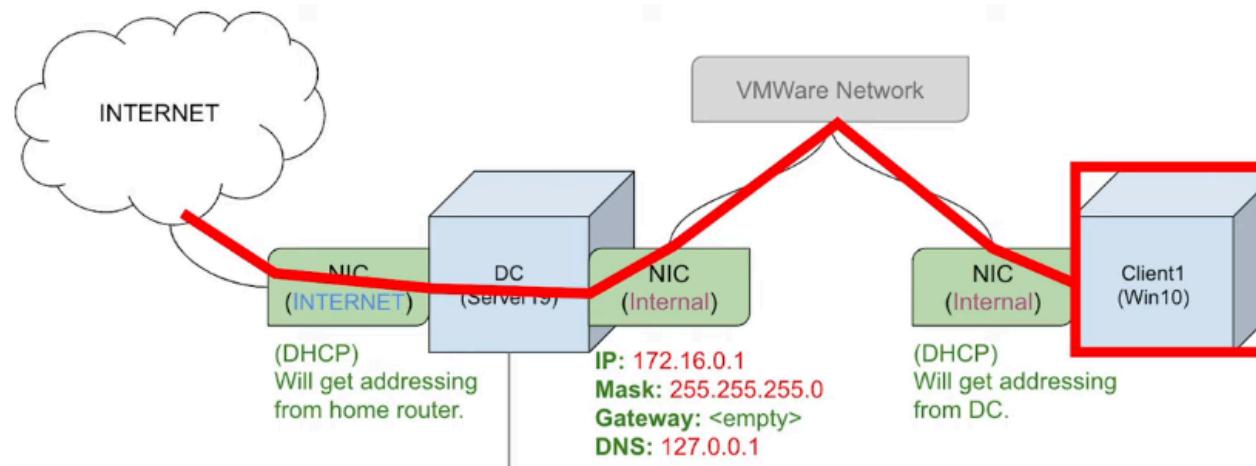
- We'll then make it a domain admin account



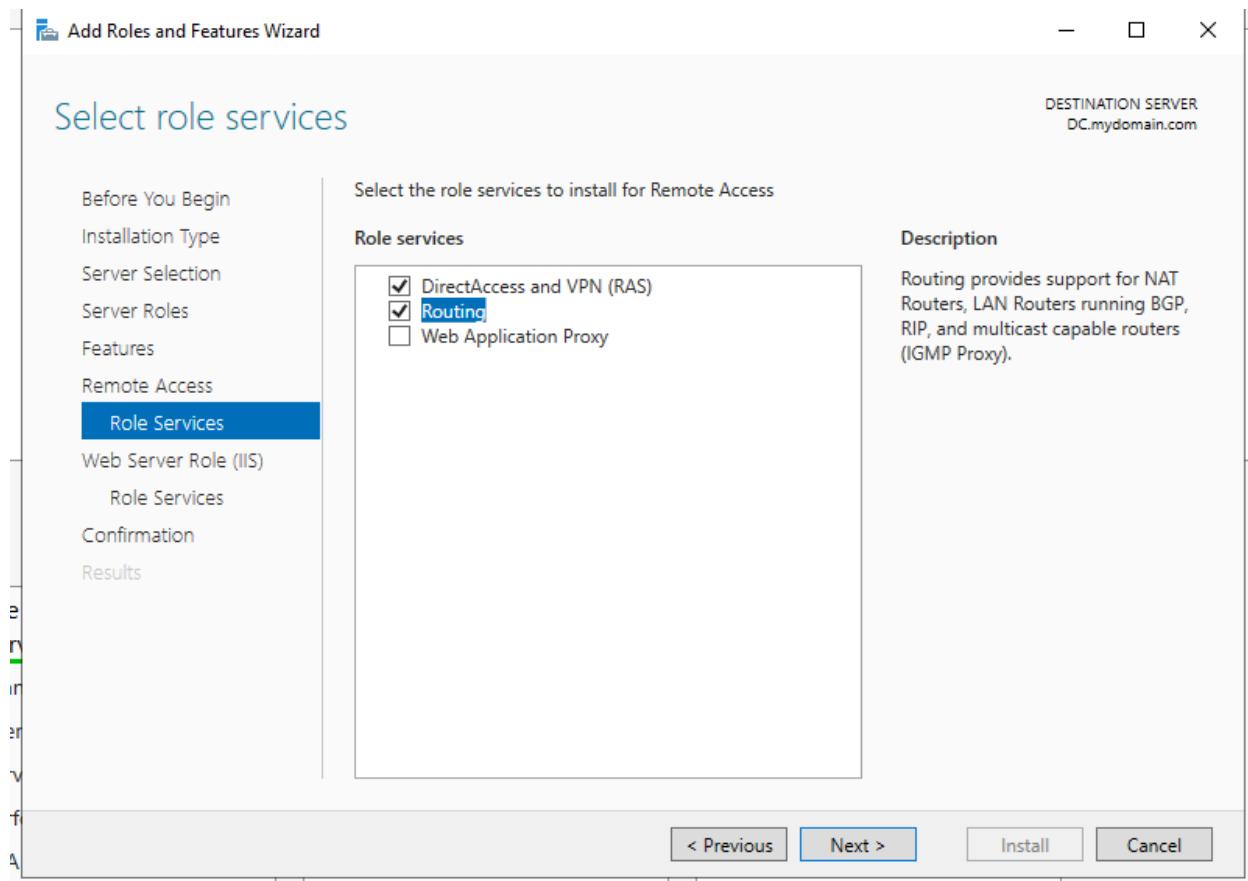
- We can now sign into MYDOMAIN with our admin account

Setting up a RAS / NAT

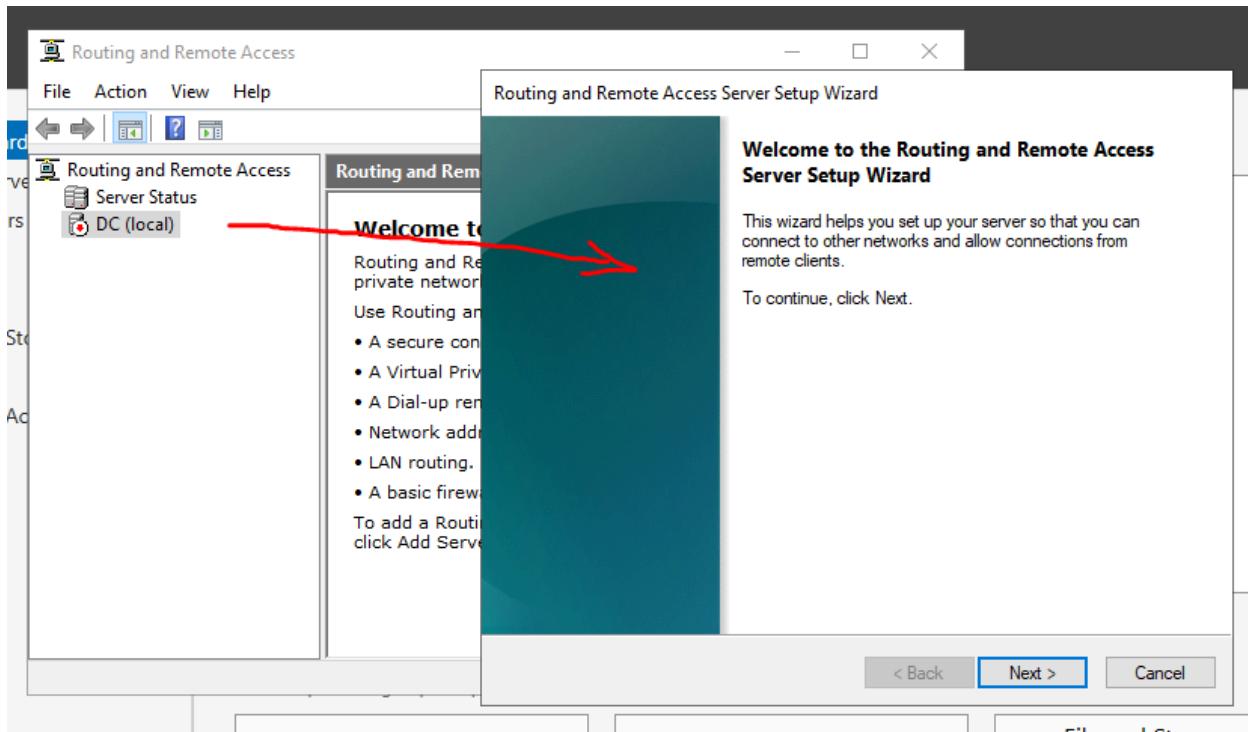
Next we'll be setting up a NAT to ensure our clients can access the internet, even through a virtual environment by using the internal network that will get internet access from our DC.



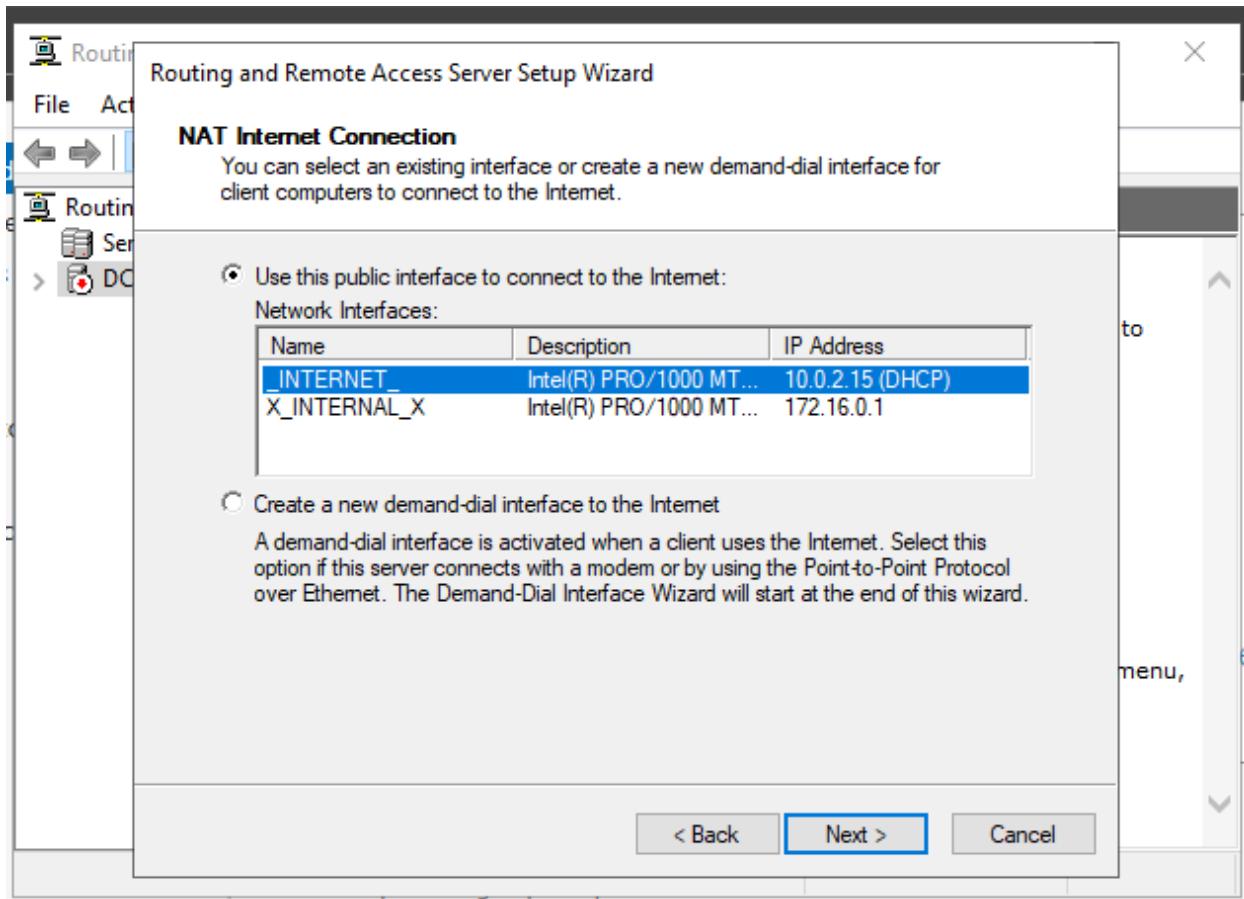
- Add roles and features, then select Remote Access



- Select routing - DirectAccess and VPN is automatically added
- Install



- Configure and Enable Routing and Remote Access
- Install NAT - Allow internal clients to connect to the Internet using one public IP Address

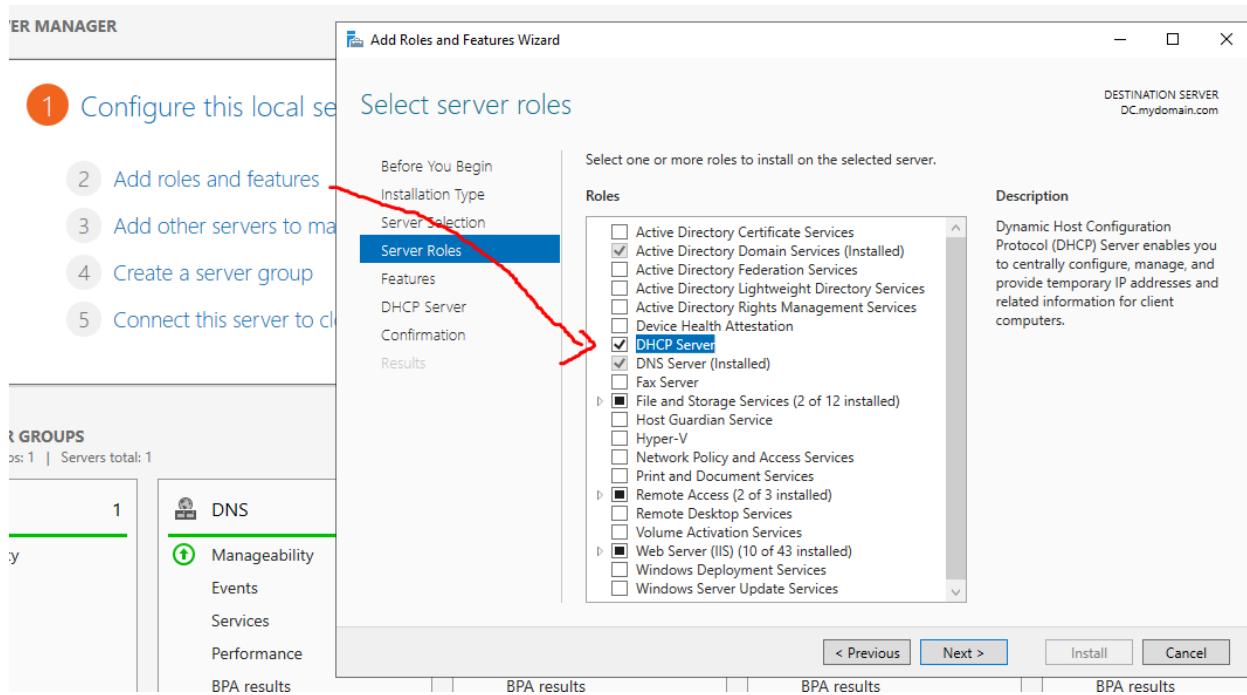


- Select _INTERNET_
- Next > Finish

Now the clients should be able to access the internet as long as we setup a DHCP - which is the next step

DHCP Server Setup

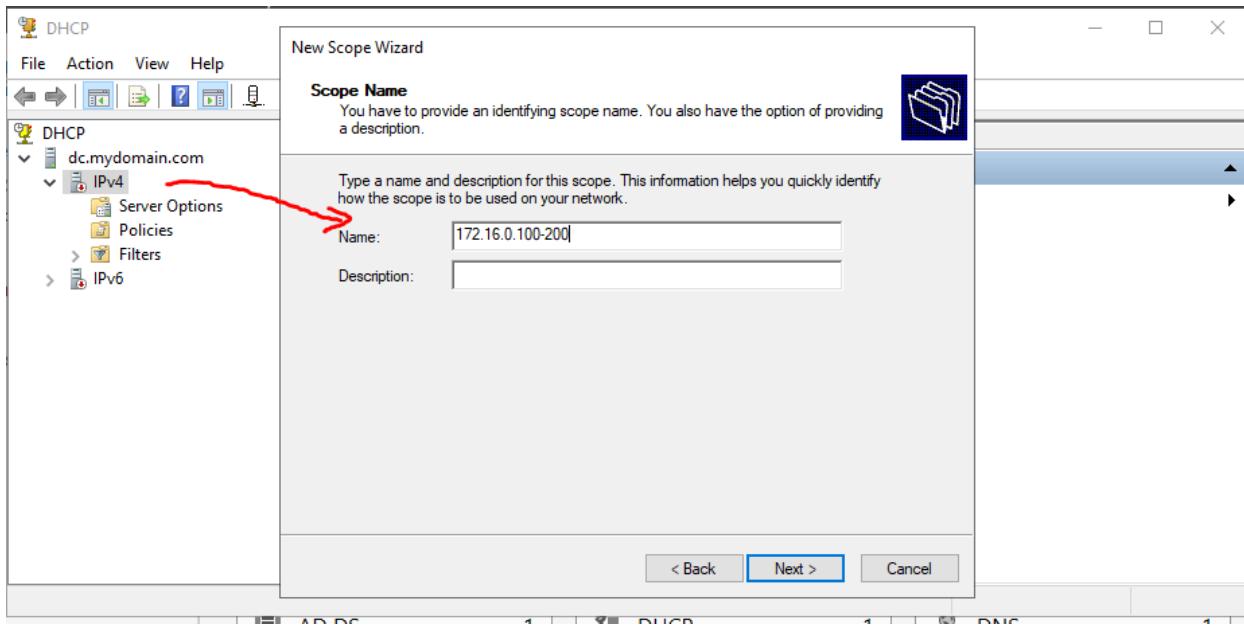
Installing DHCP



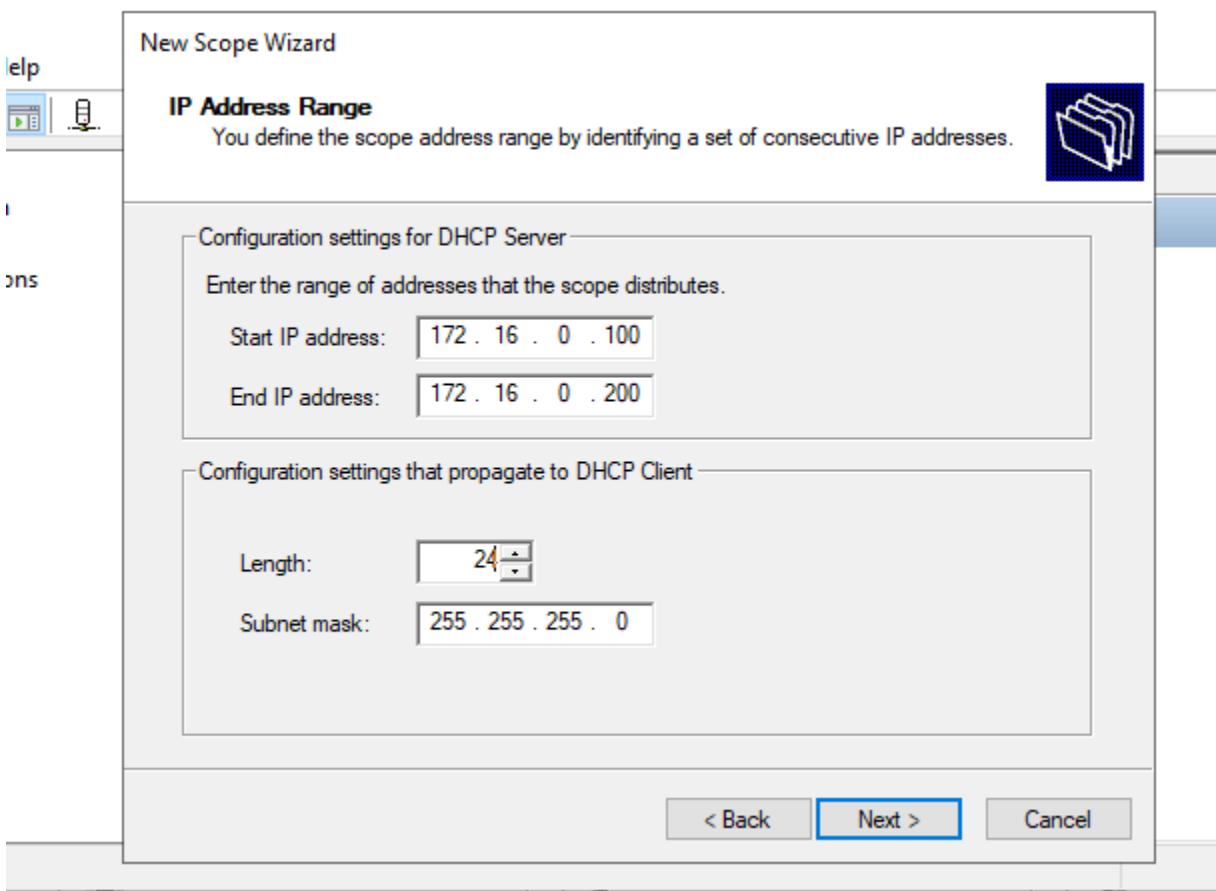
- Setup DHCP server this time

Setting up DHCP

The purpose of the DHCP is to ensure that the clients on the network can automatically get their IP addresses

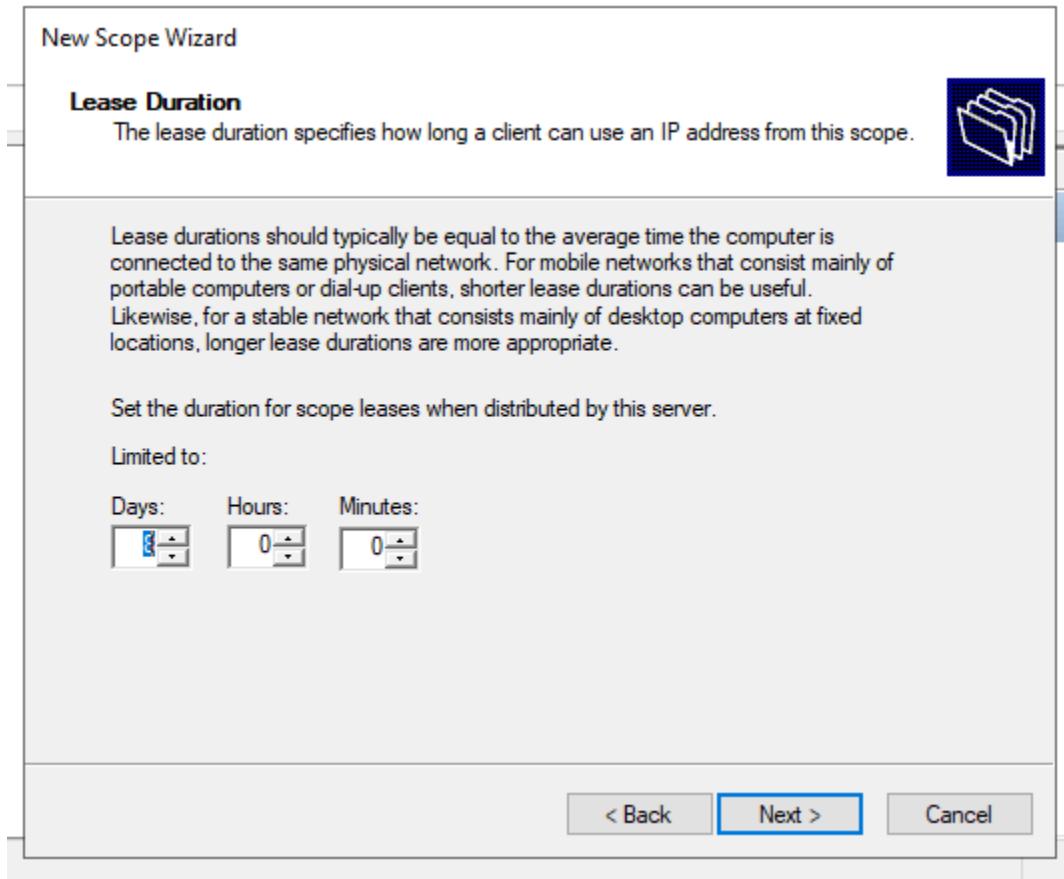


- New Scope
 - I name it what the scope will be, which is 172.16.0.100-200

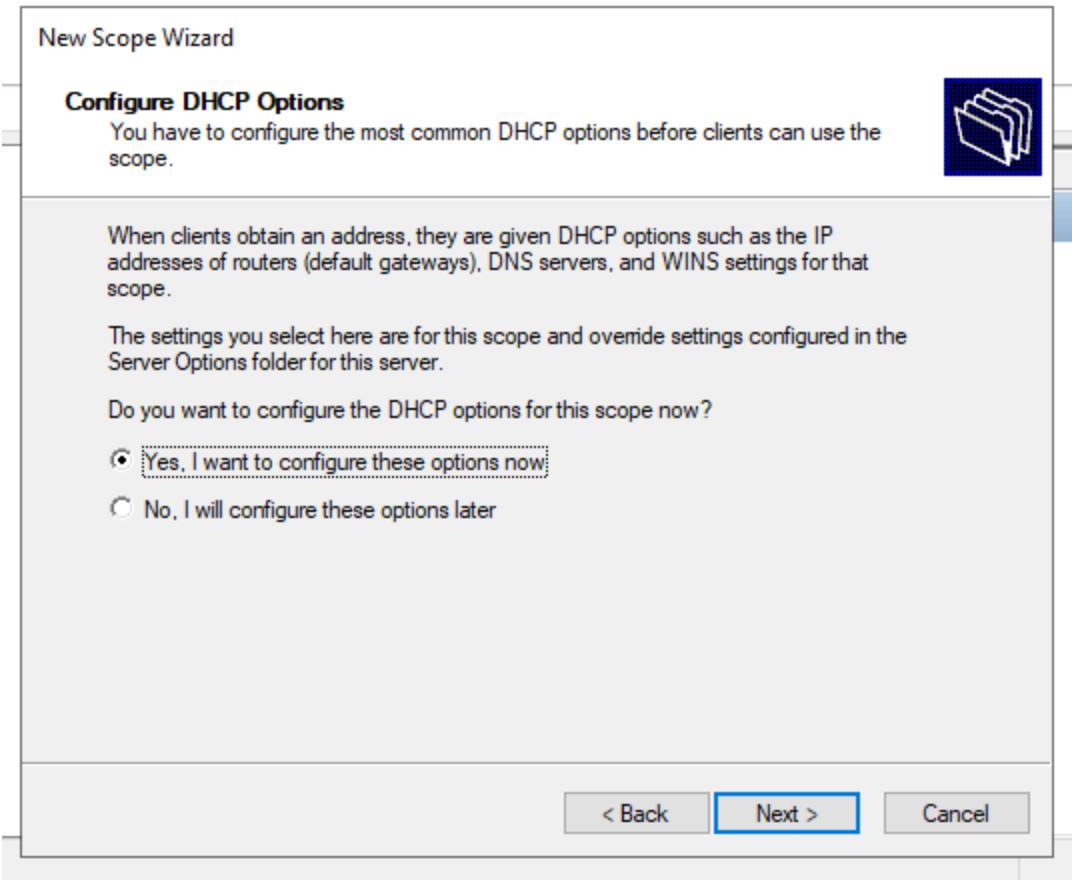


- Start IP Address = 172.16.0.100
- End IP Address = 172.16.0.200

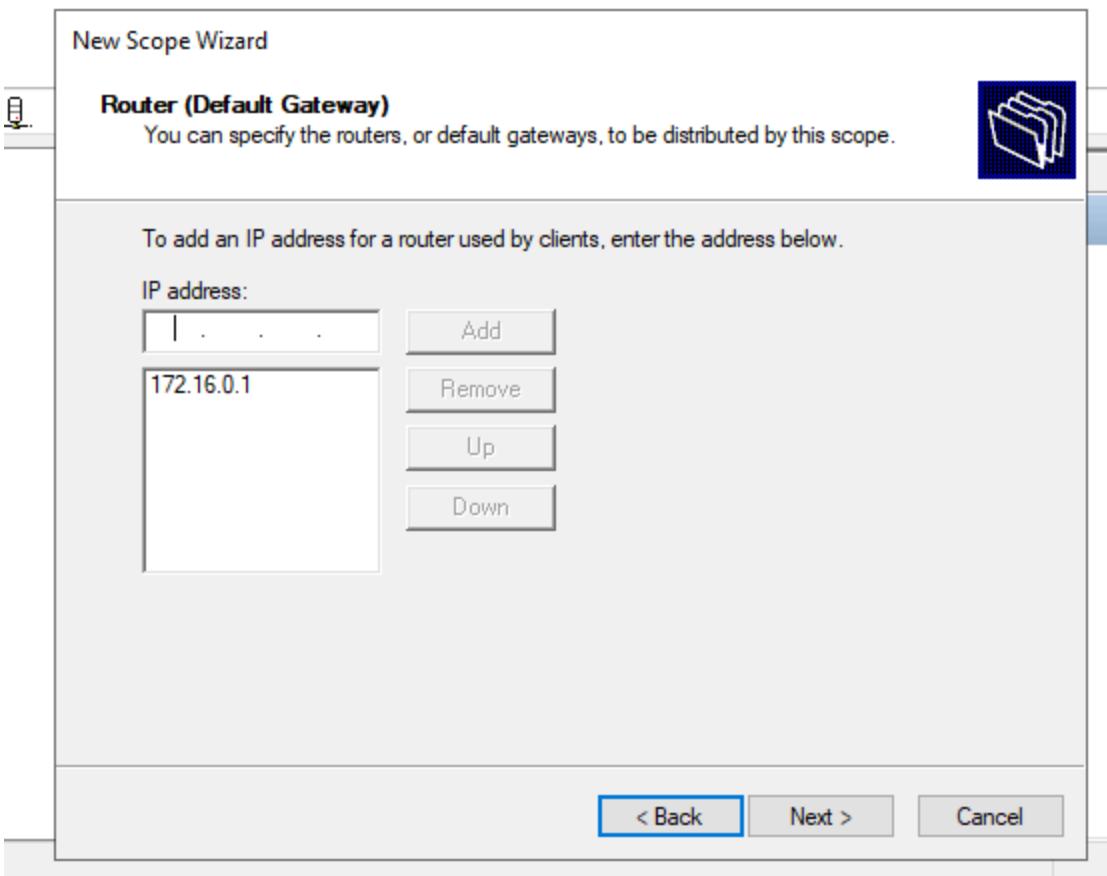
- Length: 24
- Subnet mask: 255, 255, 255, 0
- We won't add Exclusions and Delays for now



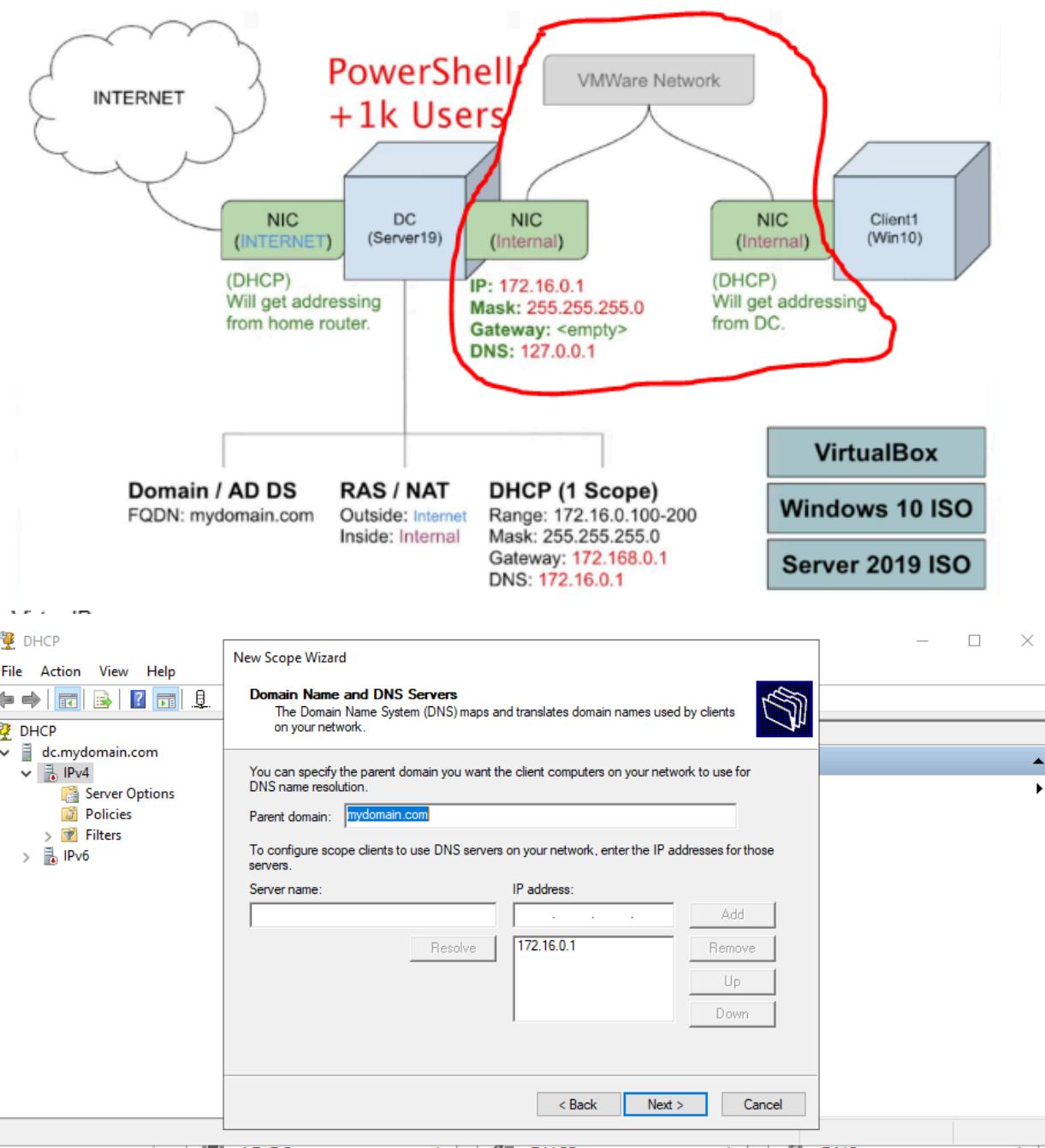
- No one can lease this IP until it expires basically
- Next



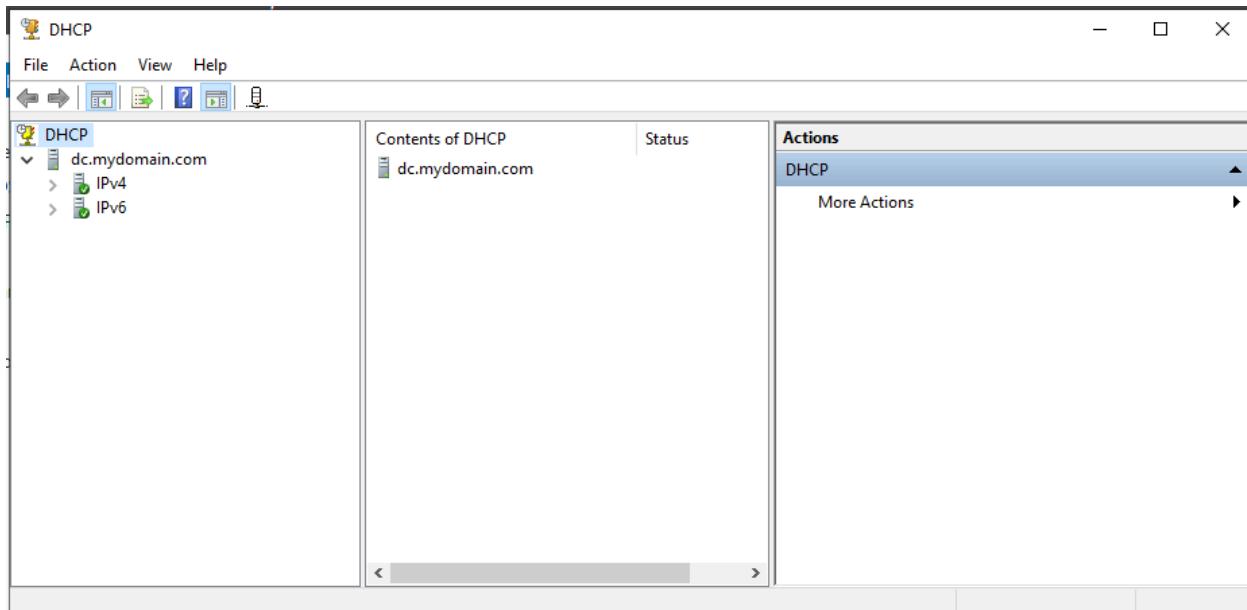
- Yes we do want to configure these options. We want our clients to access the internet



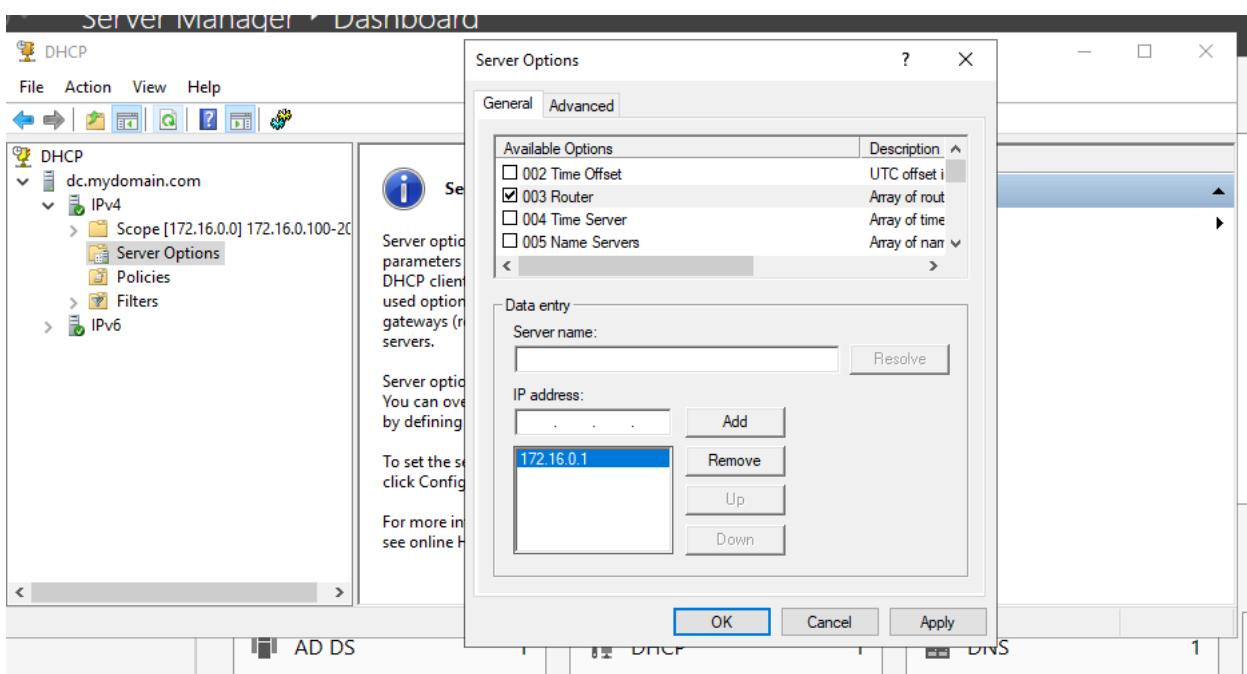
- You then add an IP address for a router used by clients. This will use the internal IP address



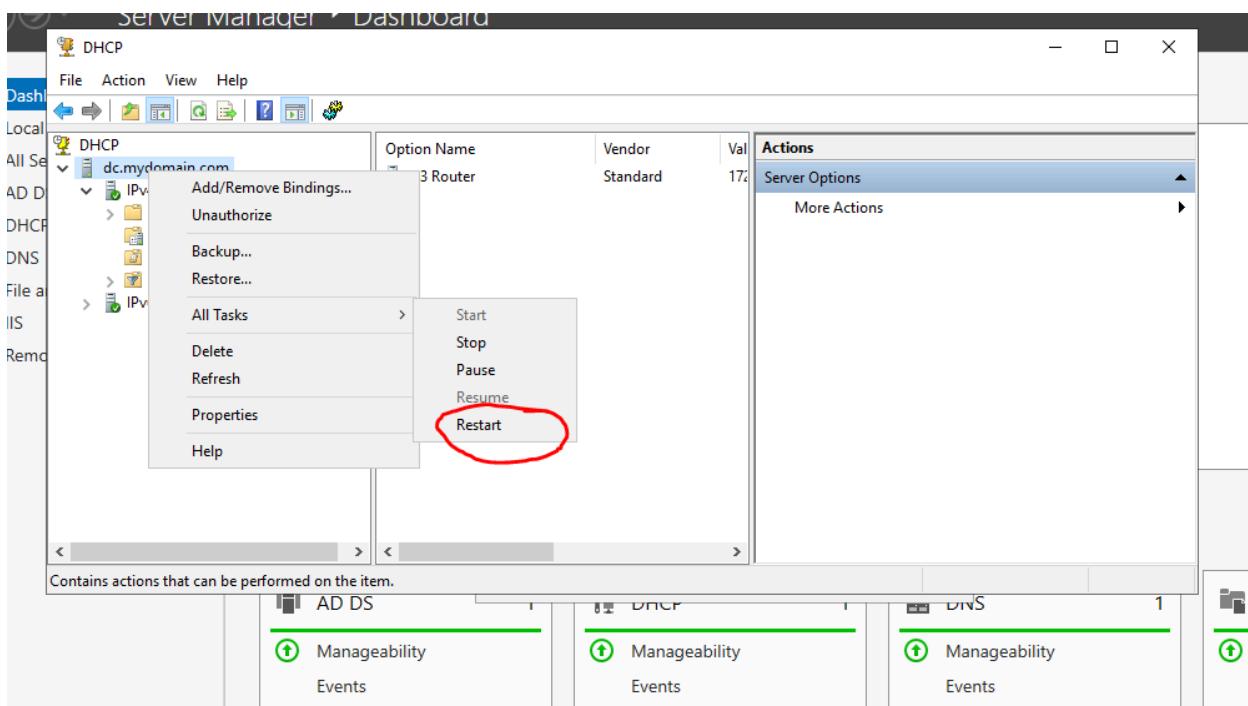
- Setup the Domain Name to be mydomain.com
- IP address to be: 172.16.0.1
- Keep going next and confirm then finish.



- Right click IPv4 and refresh it
- Right click dc.mydomain.com, select authorise
- Right click it again, select refresh
- Now everything should be up and running



- To ensure that the router is fully setup Right click Server Options > Configure Options
- Make sure Router is selected
 - Add the DomainController IP address > Apply > OK

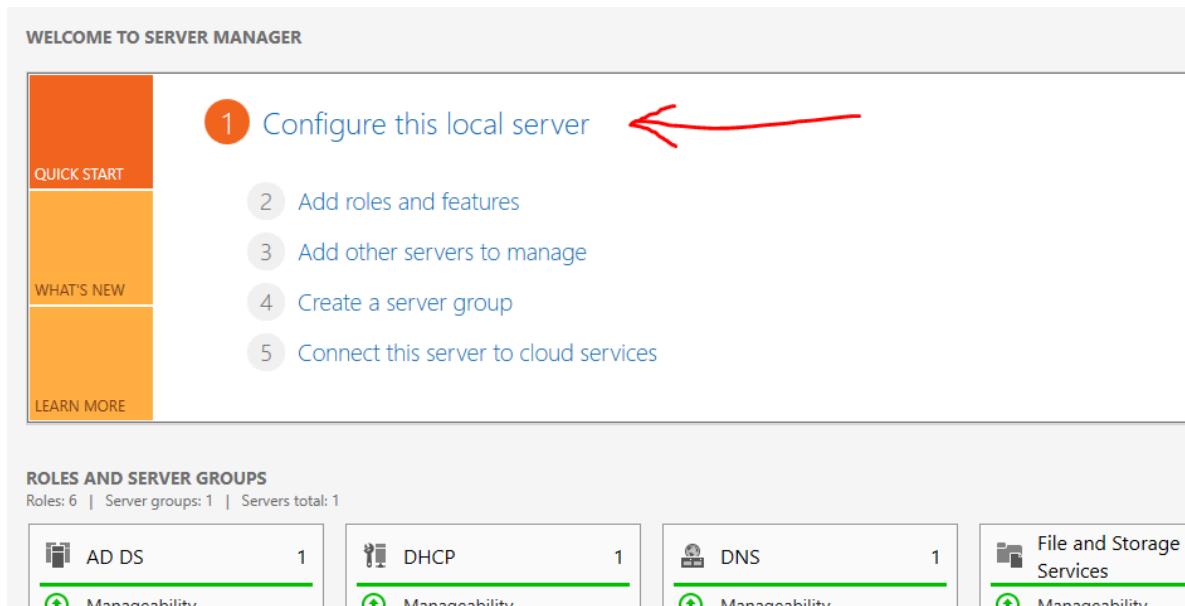


- This will apply all the changes made.

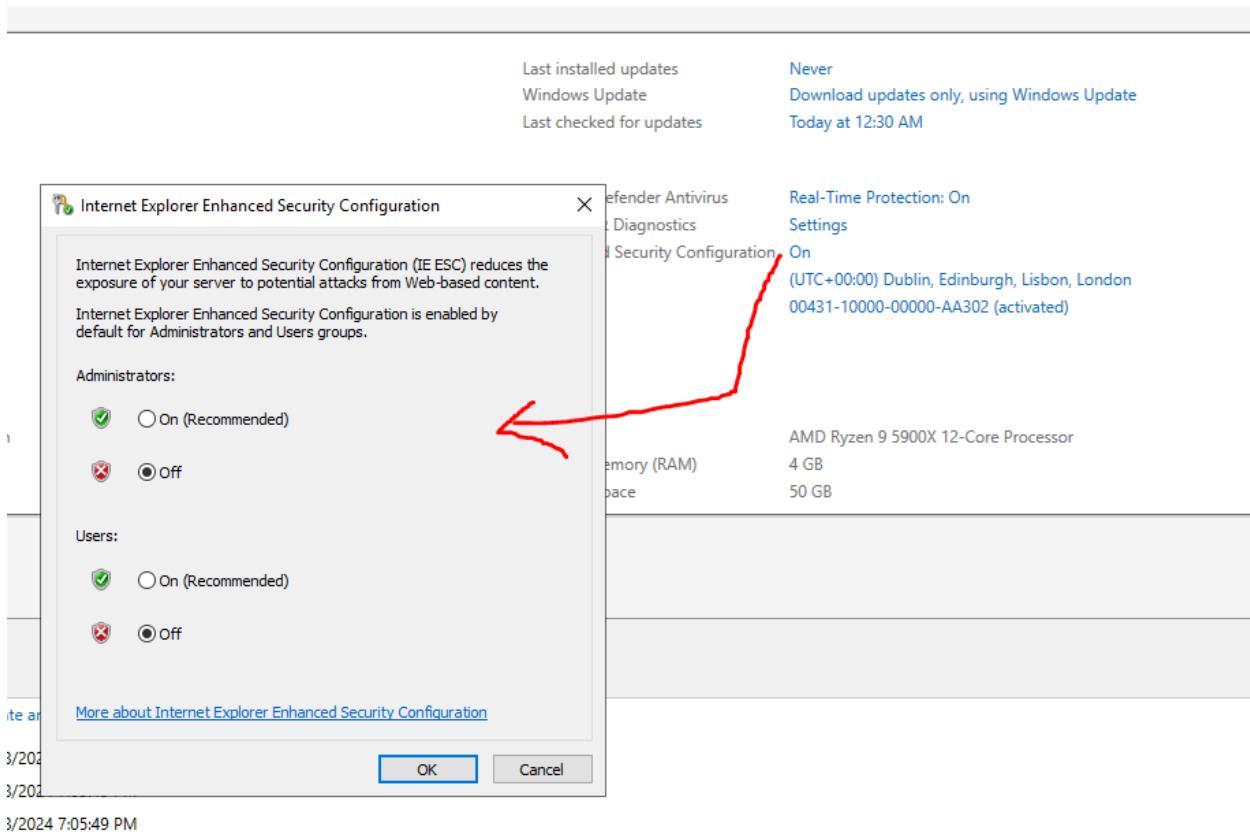
Setting up Users

Allowing for Script to be used

For this, WE WILL GO THROUGH STEPS TO ONLY DO IN A HOME PROJECT. IT'S AN UNSECURE METHOD BUT IT ALLOWS FOR DATA TO BE IMPLEMENTED WITH EASE.



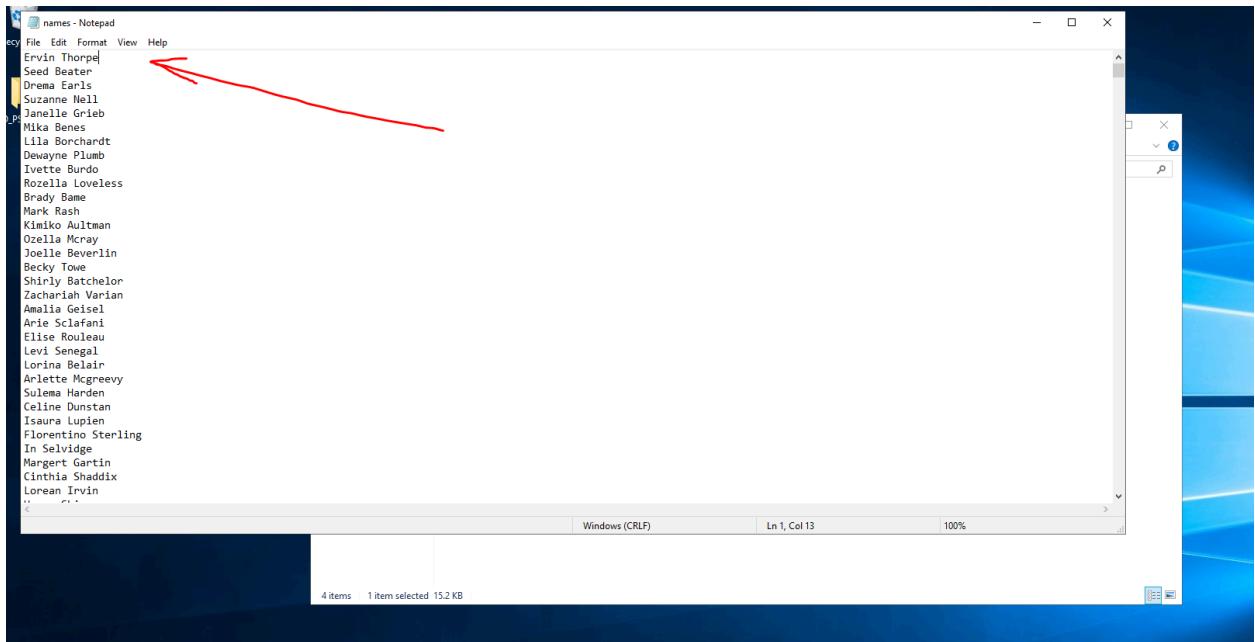
- Configure this local server



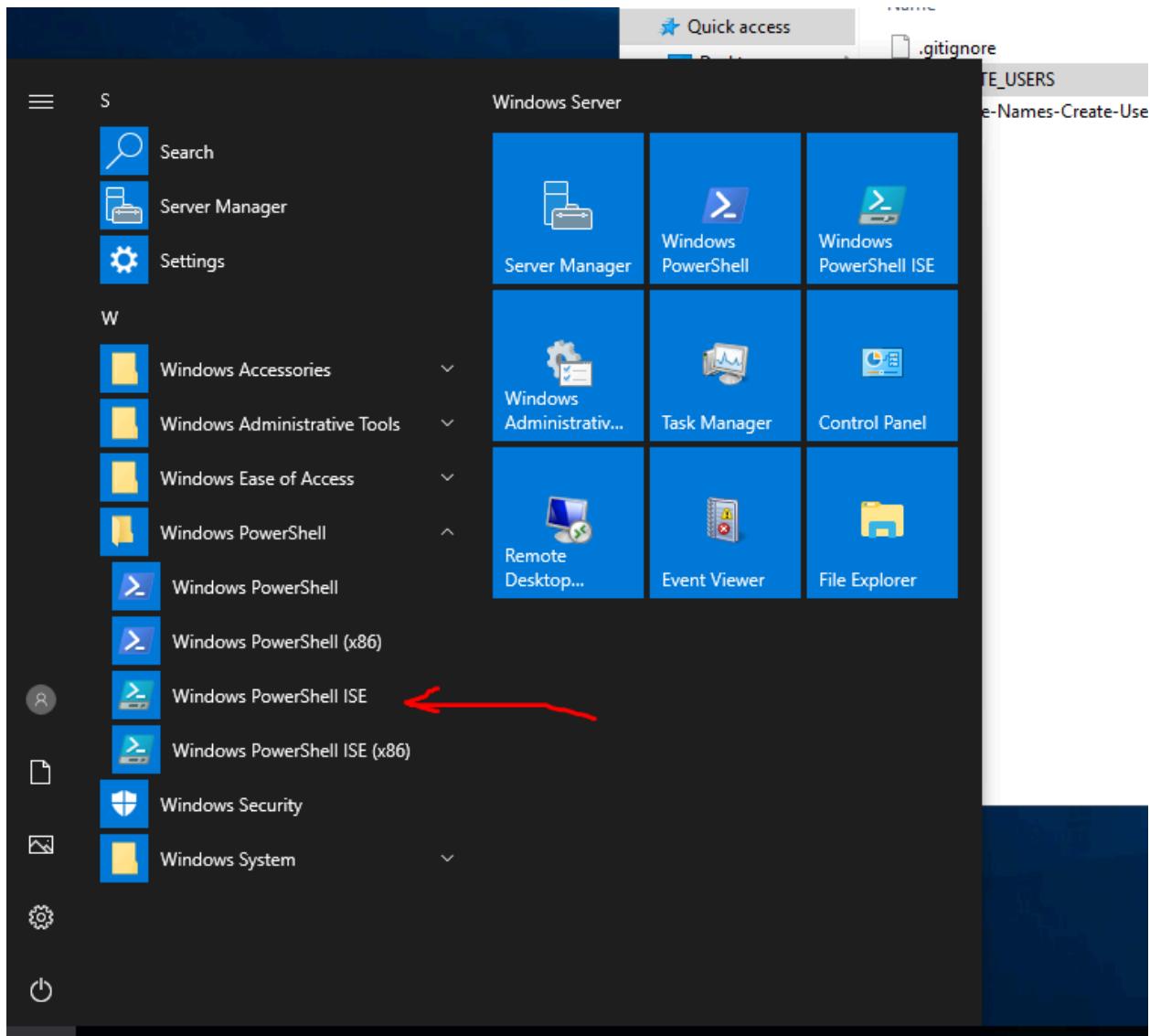
- Select Enhanced Security Configuration
- Turn admin and users both off.

Download the script and users from this link:

https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqbTIMYTZCS3F3U08xN2VXem9mRFNMak85U25BZ3xBQ3Jtc0trUWQ3TJV3Y0dBSKVQUDhjMDNKVEZIWGFLX196RnotV280VmIHNHVkX1diLTgwMndXcWJVREI3d19oOUx0QldmRHdFcXJZWIBYNFdqamRqQ05tNEZ1dzdUZjh6NnZ6SHZGNjkxdHzodGJ1MW9sdUpxWQ&q=https%3A%2F%2Fgithub.com%2Fjoshmadakor1%2FAD_PS%2Farchive%2Frefs%2Fheads%2Fmaster.zip&v=MHsl8hJmgql



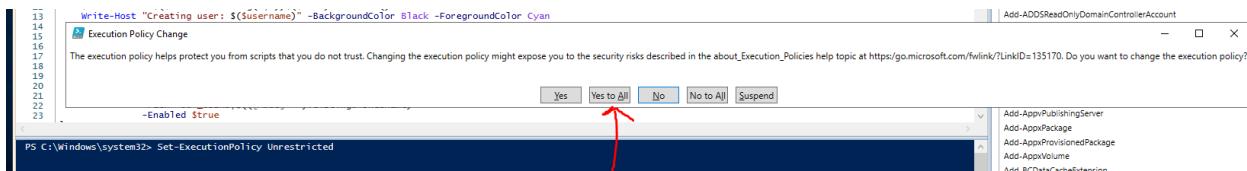
- Export the folder onto desktop for ease
- Add your name to the top of the list



- Go to Windows PowerShell ISE from the start menu
- Right click > Open with Administrator

The screenshot shows the Windows PowerShell ISE interface. On the left is the script editor with the file 1_CREATE_USERS.ps1 open. The script contains code to read names from a file, convert a password to a secure string, and create users in Active Directory. On the right is a 'Commands' pane listing various Active Directory cmdlets. A context menu is open over the script code, with the 'Run' option highlighted.

- Open the application, then open 1_CREATE_USERS PowerShell script
- We'll then have to unrestrict the policies to use this script THIS IS ONLY FOR HOMELAB PURPOSES, NOT INTENDED FOR REAL PRODUCTION



- Type Set-ExecutionPolicy Unrestricted
- Select Yes to All

Script Short Explanation

The screenshot shows the Windows PowerShell ISE interface with the script 1_CREATE_USERS.ps1 open. The top section of the script, which defines variables like \$PASSWORD_FOR_USERS and \$USER_FIRST_LAST_LIST, and sets the password to 'Password1', is highlighted with a red box.

- Top section are the variables, these variables will be used within the script
- We'll set the password for the user to be a simple "Password1"

- The Get-Content will be getting all the information from names.txt and putting it into the user_first_last_list as an array

```

1 # ----- Edit these Variables for your own Use Case ----- #
2 $PASSWORD_FOR_USERS = "Password1"
3 $USER_FIRST_LAST_LIST = Get-Content .\names.txt
4 #
5
6 $password = ConvertTo-SecureString $PASSWORD_FOR_USERS -AsPlainText -Force
7 New-ADOrganizationalUnit -Name _USERS -ProtectedFromAccidentalDeletion $false
8
9 foreach ($n in $USER_FIRST_LAST_LIST) {
10     $first = $n.Split(" ")[0].ToLower()
11     $last = $n.Split(" ")[1].ToLower()
12     $username = $($first.Substring(0,1))$($last)".ToLower()
13     Write-Host "Creating user: $($username)" -BackgroundColor Black -ForegroundColor Cyan
14
15     New-AdUser -AccountPassword $password
16         -GivenName $first

```

● It's taking our plaintext user password and turns it into an object that powershell can use as a secure password
 ● New-ADOrganizationUnit creates a new Organisational Unit called _USERS and enables it to be accidentally deleted - just to quicken the process for the lab if it's needed to be deleted.

```

5
6 $password = ConvertTo-SecureString $PASSWORD_FOR_USERS -AsPlainText -Force
7 New-ADOrganizationalUnit -Name _USERS -ProtectedFromAccidentalDeletion $false
8
9 foreach ($n in $USER_FIRST_LAST_LIST) {
10     $first = $n.Split(" ")[0].ToLower()
11     $last = $n.Split(" ")[1].ToLower()
12     $username = $($first.Substring(0,1))$($last)".ToLower()
13     Write-Host "Creating user: $($username)" -BackgroundColor Black -ForegroundColor Cyan
14
15     New-AdUser -AccountPassword $password
16         -GivenName $first

```

● This is the loop that creates the usernames for the different users.
 ● It loops through each name
 ● Splits the name into an array [0,1] , 0 being the first name and 1 being the last name
 ● 0 = first name, is then assigned to the \$first variable
 ● 1 = last name, is then assigned to the \$last variable
 ● \$username is then assigned by taking the first letter of the \$first variable and connecting it with the \$last variable
 ● All variables use .ToLower() to lowercase all the letters
 ● Write-Host simply just writes in the host “Creating User: USERNAME” for clarity. Background colour and foreground colour is changed for visual clarity.

```

$last = $n.Split(" ")[1].ToLower()
$username = $($first.Substring(0,1))$($last)".ToLower()
Write-Host "Creating user: $($username)" -BackgroundColor Black -ForegroundColor Cyan

```

```

New-AdUser -AccountPassword $password
-GivenName $first
-Surname $last
-DisplayName $username
-Name $username
-EmployeeID $username
-PasswordNeverExpires $true
-Path "ou=_USERS,$([ADSI]'").distinguishedName"
-Enabled $true

```

- This then takes all the variables we've set and adds them to the different variables that are given on the Active Directory. This way we're *automatically* creating users without having to do it manually

Administrator: Windows PowerShell ISE

File Edit View Tools Debug Add-ons Help

1_CREATE_USERS.ps1 X

```

1 # ----- Edit these Variables for your own Use Case ----- #
2 $PASSWORD_FOR_USERS = "Password1"
3 $USER_FIRST_LAST_LIST = Get-Content .\names.txt
4 #
5
6 $password = ConvertTo-SecureString $PASSWORD_FOR_USERS -AsPlainText -Force
7 New-ADOrganizationalUnit -Name _USERS -ProtectedFromAccidentalDeletion $false
8
9 foreach ($n in $USER_FIRST_LAST_LIST) {
10     $first = $n.Split(" ")[0].ToLower()
11     $last = $n.Split(" ")[1].ToLower()
12     $username = $($first.Substring(0,1))$($last).ToLower()
13     Write-Host "Creating user: $($username)" -BackgroundColor Black -ForegroundColor Cyan
14
15     New-ADUser -AccountPassword $password `
16                 -GivenName $first `
17                 -Surname $last `
18                 -DisplayName $username `
19                 -Name $username `
20                 -EmployeeID $username `
21                 -PasswordNeverExpires $true `
22                 -Path "ou=_USERS,$([ADSI]'').distinguishedName" `
23                 -Enabled $true
24 }

```

PS C:\Windows\System32> cd C:\Users\a-ethorpe\Desktop\AD_PS-master

PS C:\Users\a-ethorpe\Desktop\AD_PS-master> ls

Directory: C:\Users\a-ethorpe\Desktop\AD_PS-master

Mode	LastWriteTime	Length	Name
-a---	8/8/2024 7:17 PM	1811	.gitignore
-a---	8/8/2024 7:17 PM	1025	1_CREATE_USERS.ps1
-a---	8/8/2024 7:17 PM	1532	Generate-Names-Create-Users.ps1
-a---	8/8/2024 7:19 PM	15582	names.txt

PS C:\Users\a-ethorpe\Desktop\AD_PS-master>

- In order to run the script, we have to change the directory to the correct location
- This will be cd C:\Users\ADMINNAME\Desktop\AD_PS-master
 - cd is Change Directory
- You can then use ls to list all the files within this directory
- From here you can run the script at the top where the arrow is pointing to the green play button

Administrator: Windows PowerShell ISE

```

File Edit View Tools Debug Add-ons Help
1_CREATE_USERS.ps1 X
1 # ----- Edit these Variables for your own Use Case ----- #
2 $PASSWORD_FOR_USERS = "password1"
3 $USER_FIRST_LAST_LIST = Get-Content .\names.txt
4 #
5
6 $password = ConvertTo-SecureString $PASSWORD_FOR_USERS -AsPlainText -Force
7 New-ADOrganizationalUnit -Name _USERS -ProtectedFromAccidentalDeletion $false
8
9 foreach ($n in $USER_FIRST_LAST_LIST) {
10     $first = $n.Split(" ")[0].ToLower()
11     $last = $n.Split(" ")[1].ToLower()
12     $username = "$($first.Substring(0,1))$($last)".ToLower()
13     Write-Host "Creating user: $($username)" -BackgroundColor Black -ForegroundColor Cyan
14
15     New-AdUser -AccountPassword $password ` 
16         -GivenName $first ` 
17         -Surname $last ` 
18         -DisplayName $username ` 
19         -Name $username ` 
20         -EmployeeID $username ` 
21         -PasswordNeverExpires $true ` 
22         -Path "ou=_USERS,$([ADSISearcher]::new().distinguishedName)" ` 
23         -Enabled $true
24 }

```

Creating user: lmagill
Creating user: pdubose
Creating user: afiles
Creating user: mclaypool
Creating user: tdeibler
Creating user: mmeisinger
Creating user: tjoines
Creating user: ademello
Creating user: nrott
Creating user: ischutte
Creating user: osimmental
Creating user: sselvage
Creating user: mtokar
Creating user: slesher
Creating user: leggleton
Creating user: mtomaszewski
Creating user: lreveles
Creating user: rclendenin
Creating user: gborg
Creating user: bmershon
Creating user: ckosinski
Creating user: niasso
Creating user: lkwock
Creating user: njaggers
Creating user: mderippio
Creating user: thaug

- The script is now running

Active Directory Users and Computers

File Action View Help

Find Users, Contacts, and Groups

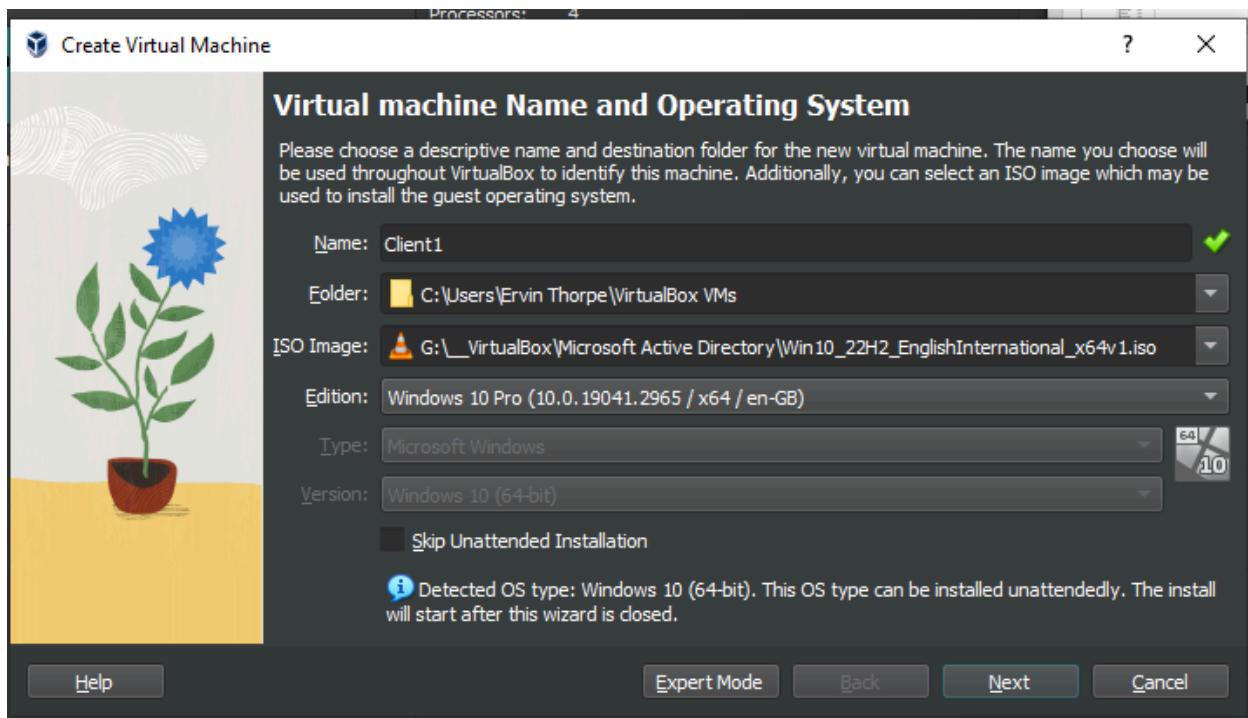
Find: Users, Contacts, and Groups In: _USERS Browse...
Name: ethorpe Find Now Stop Clear All

Search results:

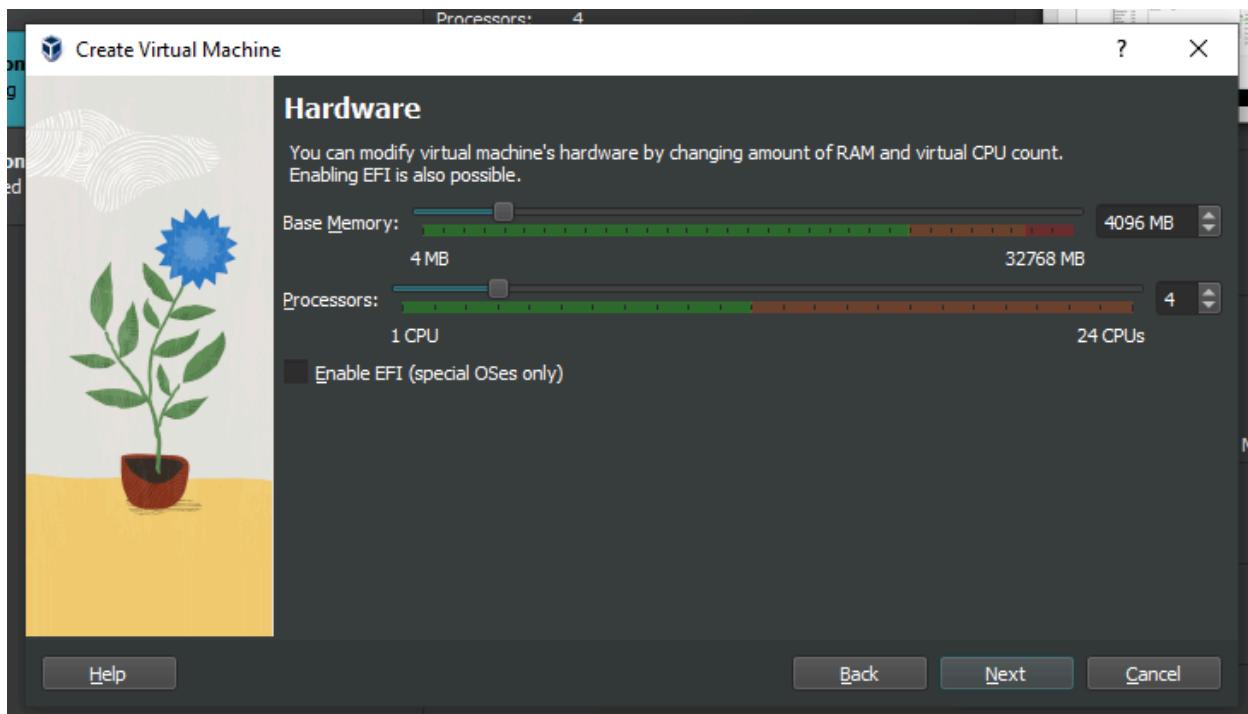
Name	Type	Description
ethorpe	User	

- Once it's done, all the users should be populated and you can search for a particular user by searching the name. In this case, I found myself.

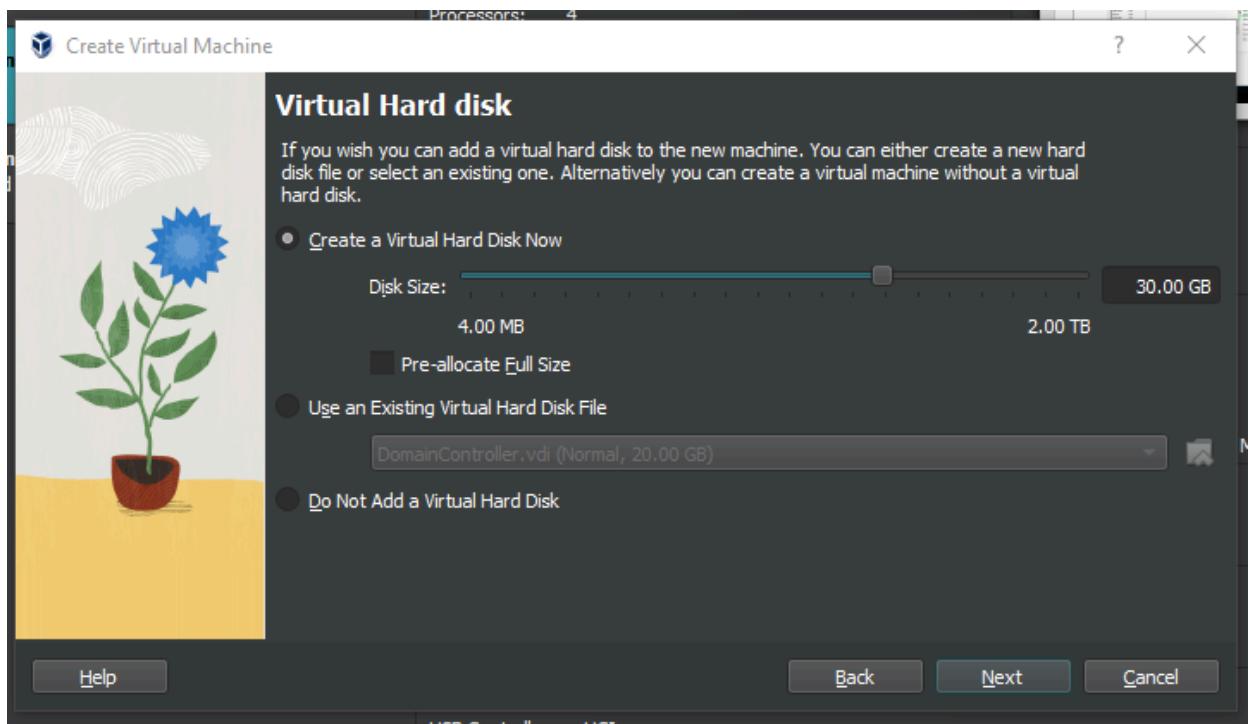
Client Windows 10 setup



- We want to add a new virtual machine.
- Name it Client1
- Select the ISO Image - Windows 10 Pro (you have to select Pro or the machine won't be able to connect to the domain controller)
- Select Next, then Next again

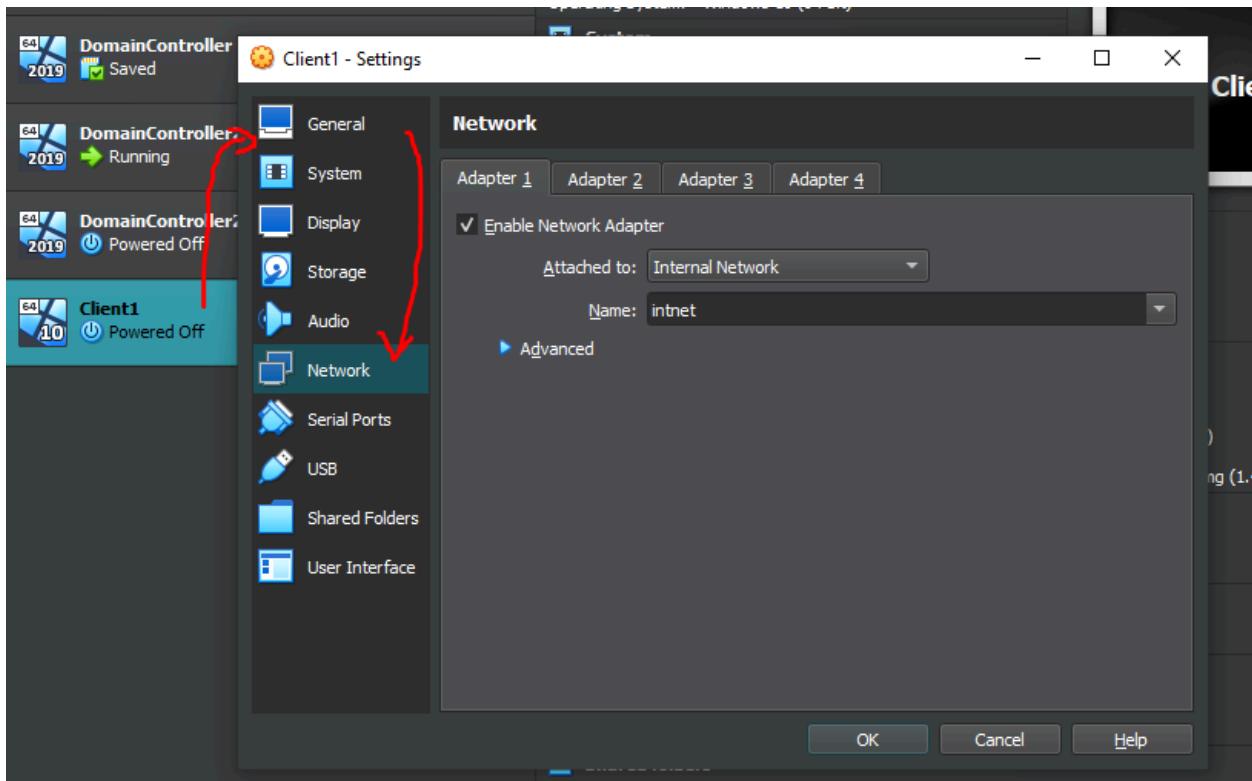


- Select a suitable set of hardware for your system
- Next

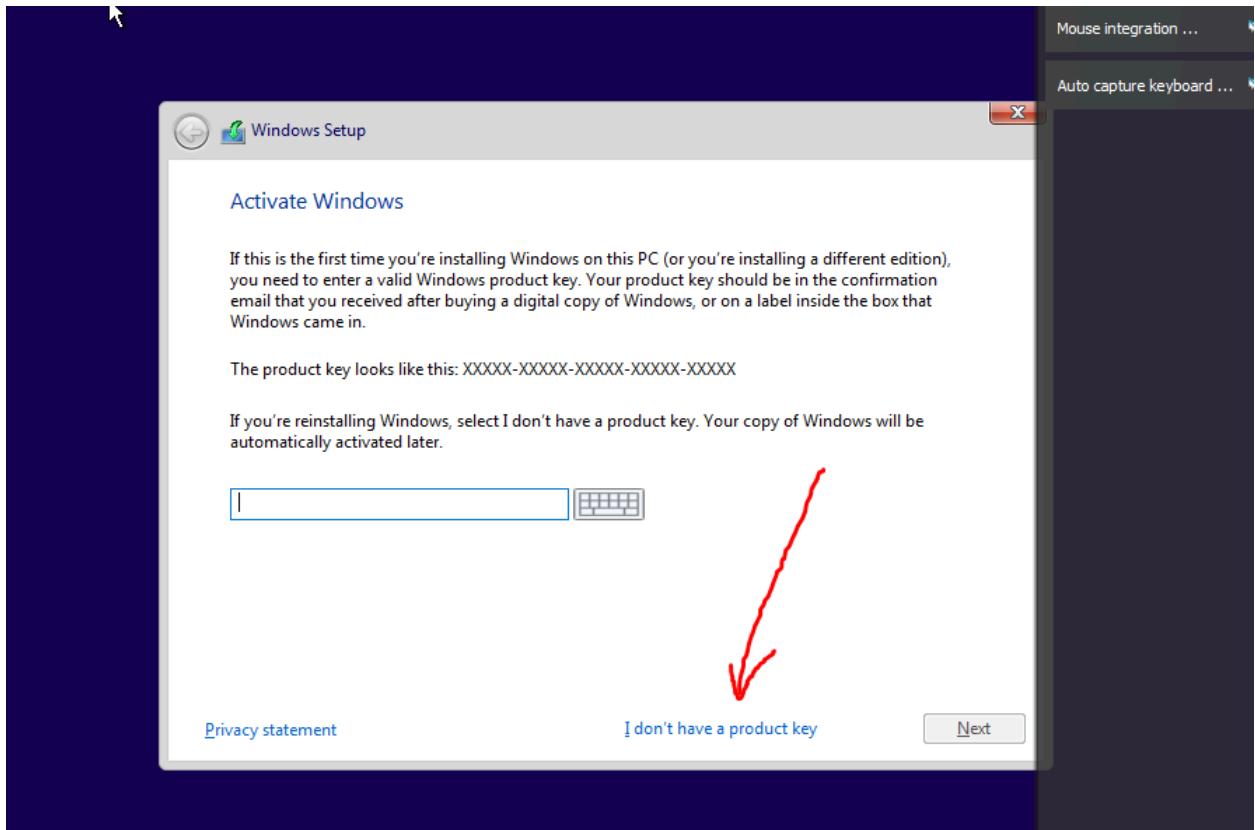


- Select a suitable space for your hard disk
- Next
- Finish

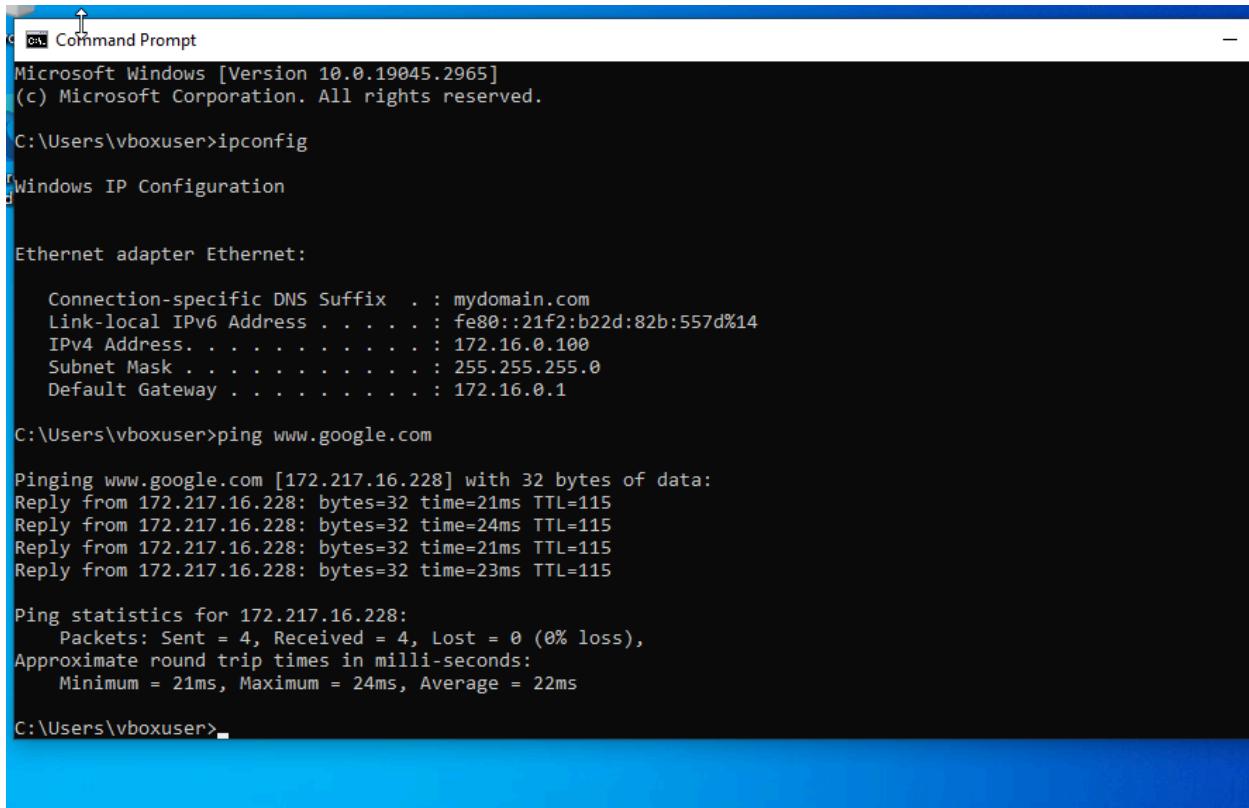
- Sometimes the system won't work properly. If this happens, just delete the Virtual Machine and make another Client1 again (You may have to delete the folder on your machine as well)
 - A weird fix that works for me is highlighting the product key and deleting all the characters - even though it shouldn't exist



- Then go to settings, go to network and make sure the adapter is set to Internal Network
- This way it will access the internet through the DomainController



- Select I don't have a product key
- Continue through all options
- Windows 10 will then be setup



```
C:\> Command Prompt
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.

C:\Users\vboxuser>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

  Connection-specific DNS Suffix . : mydomain.com
  Link-local IPv6 Address . . . . . : fe80::21f2:b22d:82b:557d%14
  IPv4 Address. . . . . : 172.16.0.100
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . : 172.16.0.1

C:\Users\vboxuser>ping www.google.com

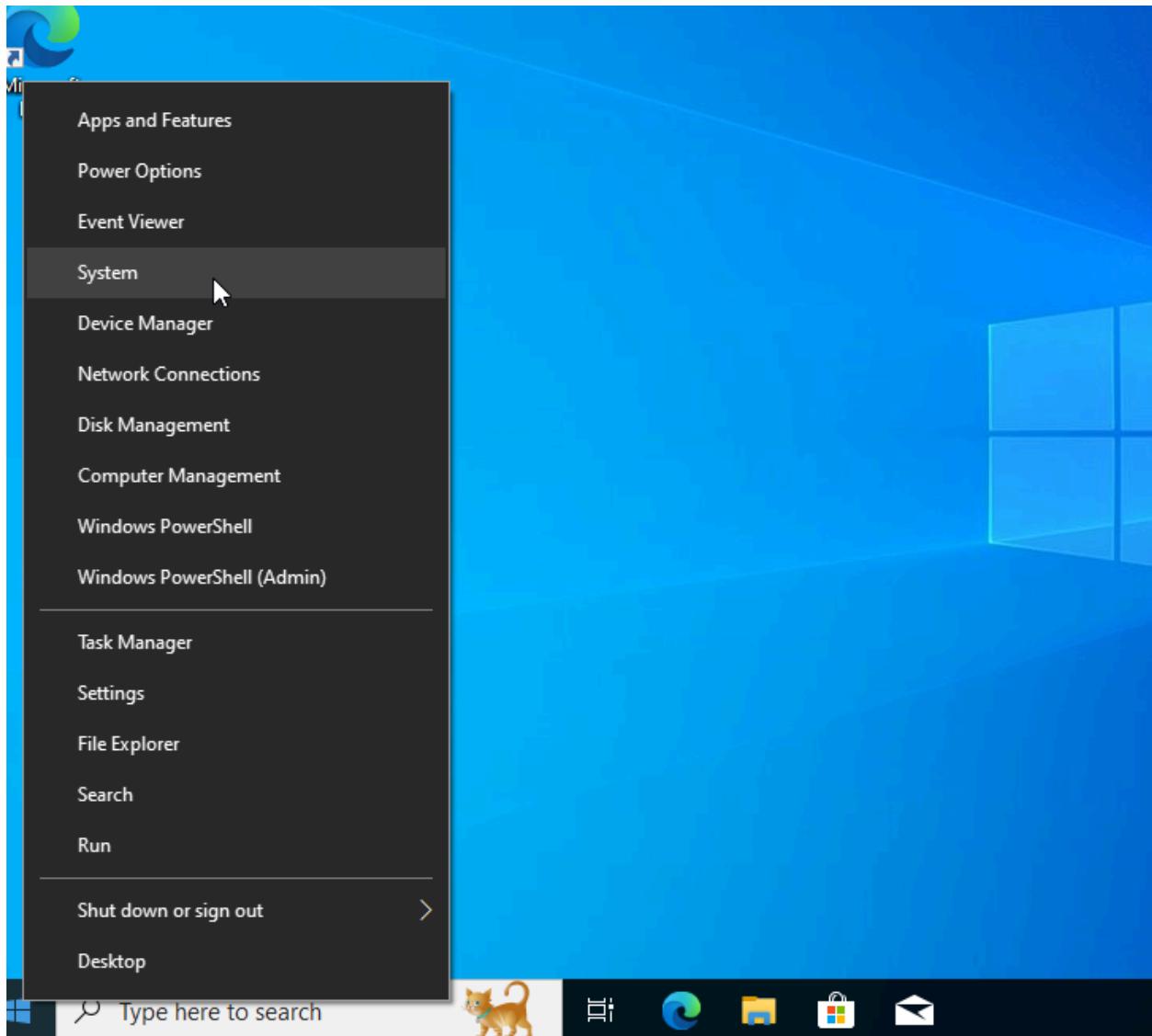
Pinging www.google.com [172.217.16.228] with 32 bytes of data:
Reply from 172.217.16.228: bytes=32 time=21ms TTL=115
Reply from 172.217.16.228: bytes=32 time=24ms TTL=115
Reply from 172.217.16.228: bytes=32 time=21ms TTL=115
Reply from 172.217.16.228: bytes=32 time=23ms TTL=115

Ping statistics for 172.217.16.228:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
  Approximate round trip times in milli-seconds:
    Minimum = 21ms, Maximum = 24ms, Average = 22ms

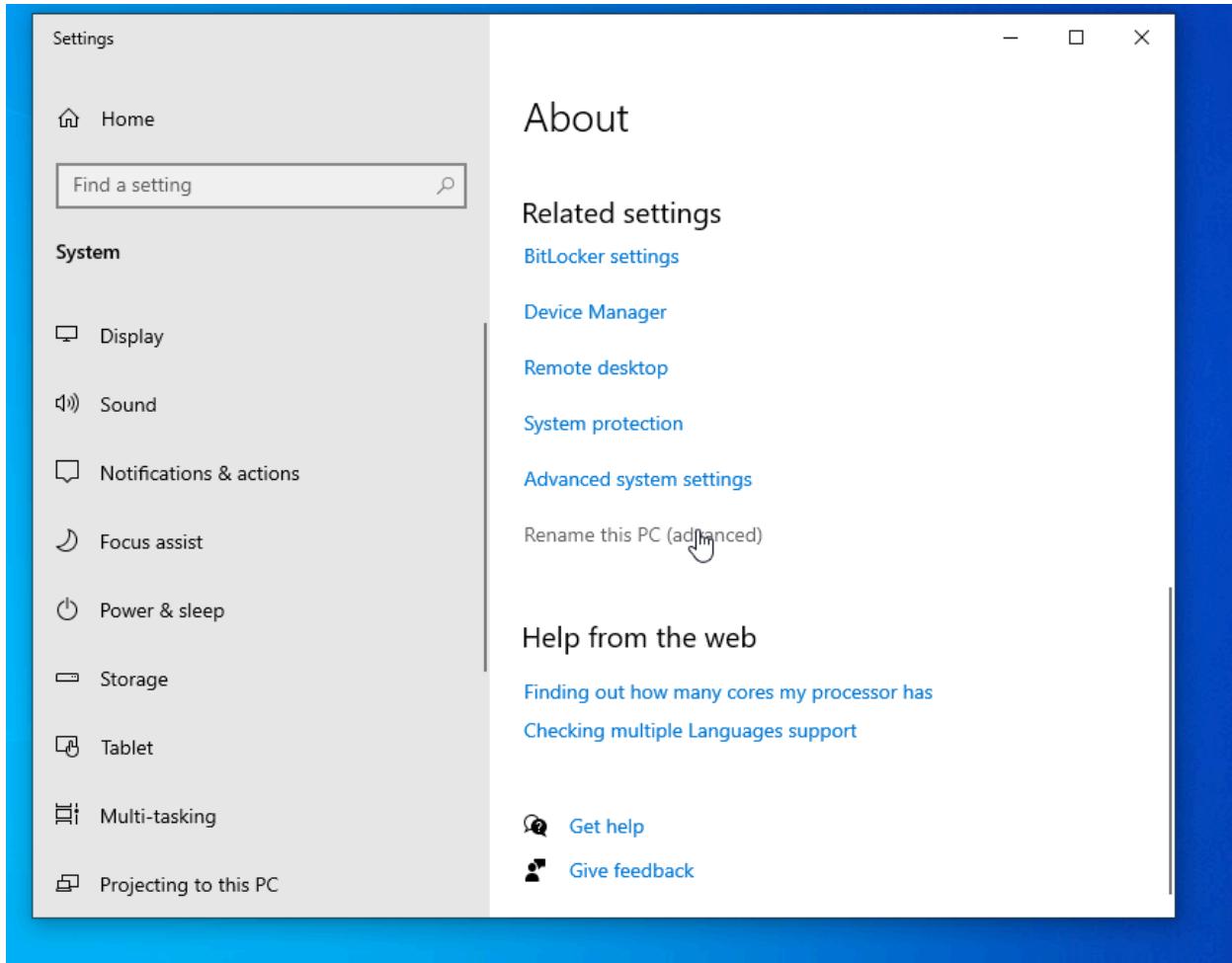
C:\Users\vboxuser>
```

- To check the Client is connected to the correct domain IP address go to command prompt>type in ipconfig
- To check if it has access to the internet, type ping www.google.com

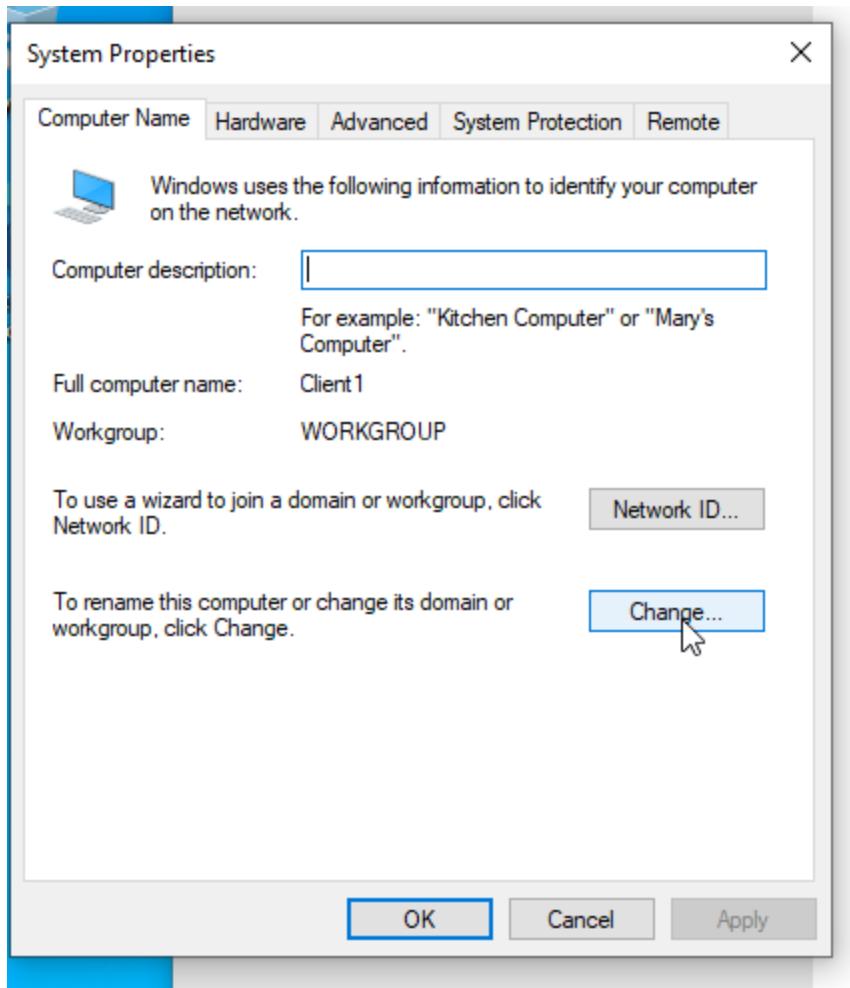
Next to ensure the computer joins the domain and has an IP address leased onto it, follow the steps:



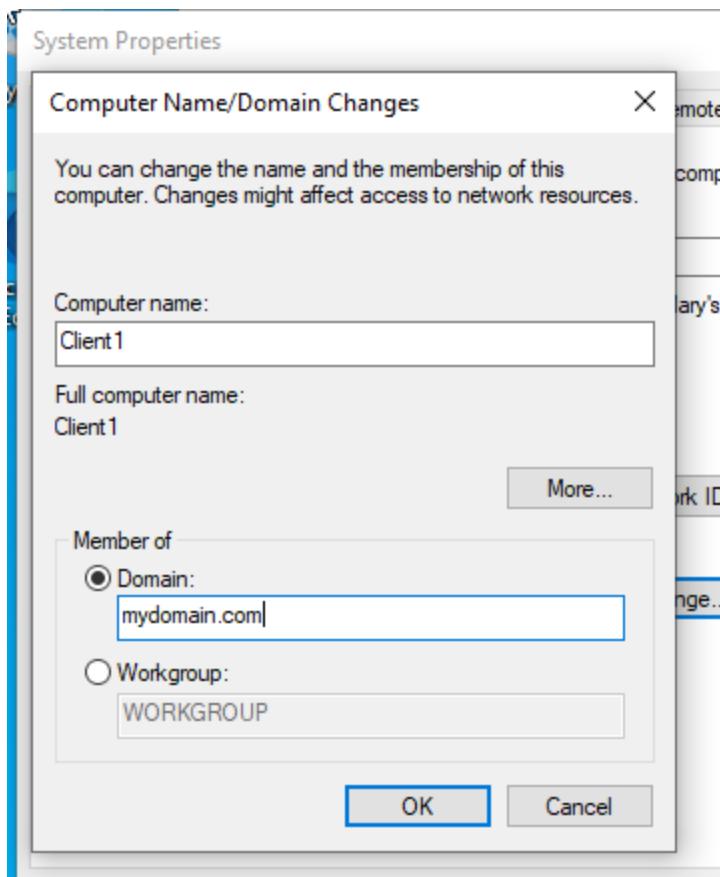
- Right click start and select System



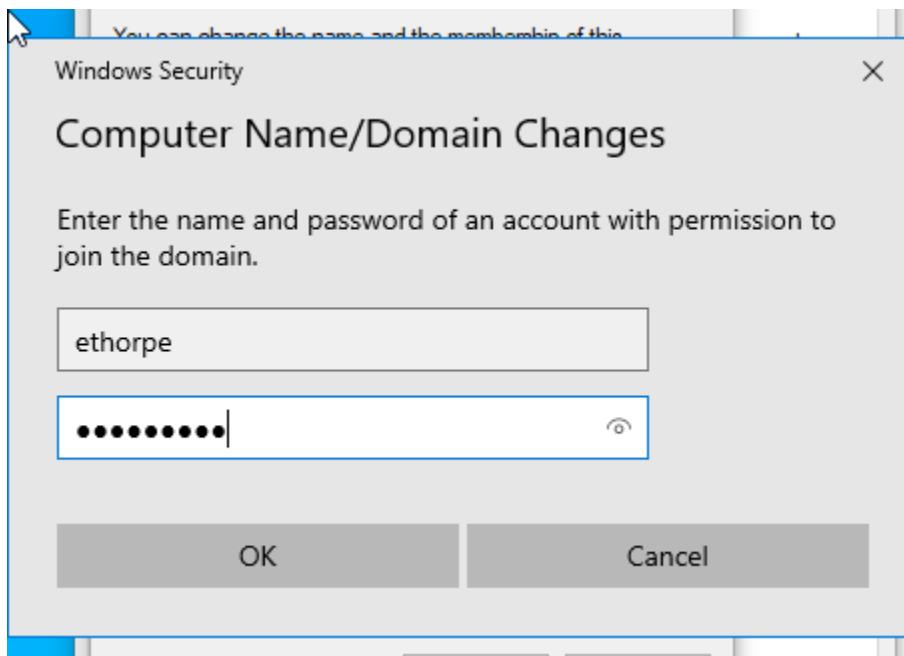
- Select “Rename this PC (advanced)”



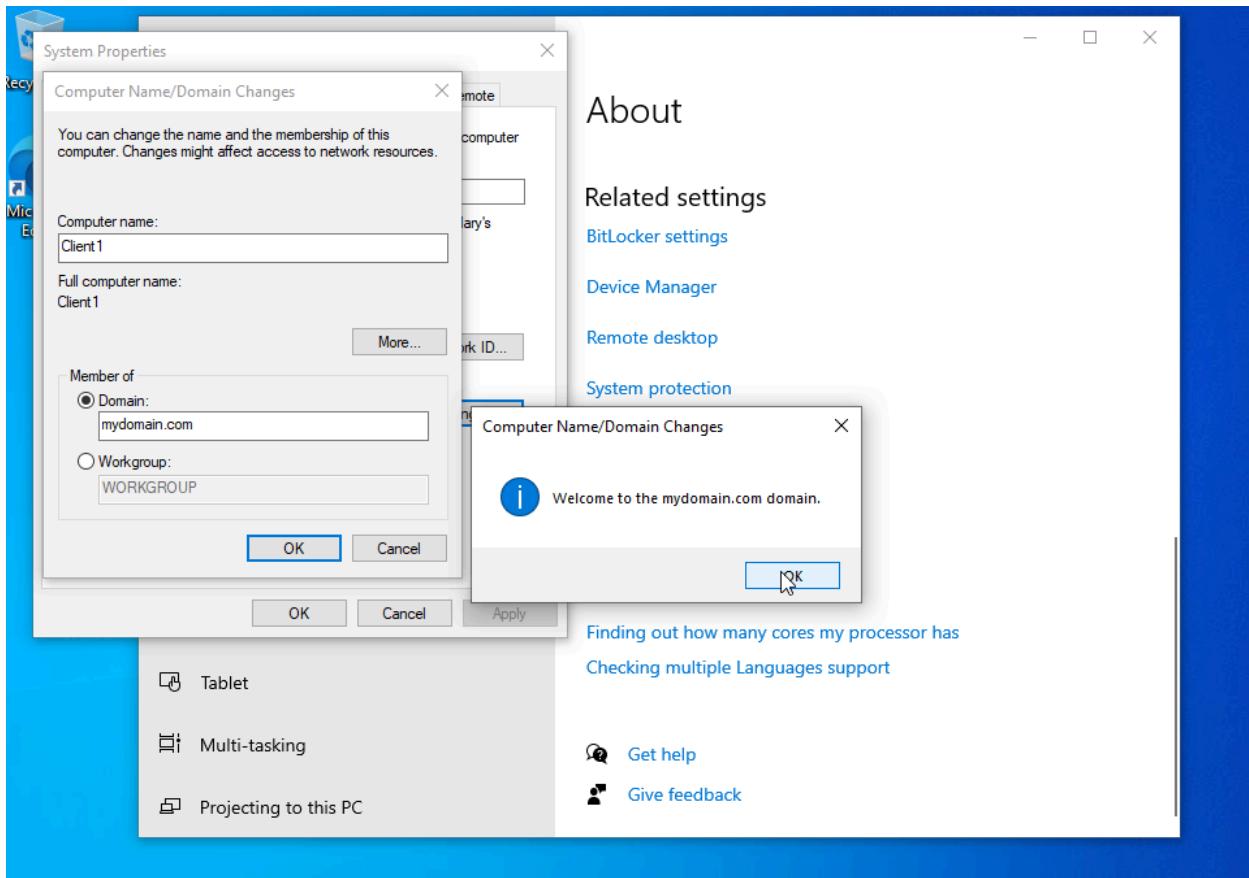
- Select Change



- Select Domain, then press OK



- Enter a valid user and password from our active directory



- We're in!

Final Checks

The screenshot shows the Microsoft DHCP Management console. The left pane displays a tree structure for 'dc.mydomain.com' under 'IPv4'. The right pane shows a table with columns: Client IP Address, Name, and Lease Expiration. One row is selected, showing '172.16.0.100' in the 'Client IP Address' column. A red arrow points from the text 'We're in!' in the previous slide to this IP address. The 'Actions' pane on the right has 'Address Leases' selected.

Client IP Address	Name	Lease Expiration
172.16.0.100	Client1.mydomain....	8/16/2024 8:43:40 PM

- If we go back to our server, go to tools>DHCP, we can see that our Client1 now has a lease to it

The screenshot shows the Windows Active Directory Users and Computers management console. On the left, there's a navigation pane with a tree view of the domain structure under 'mydomain.com'. The 'Computers' node is selected. On the right, a table lists a single computer entry:

Name	Type	Description
CLIENT1	Computer	

A red arrow points from the text above the screenshot to the 'Name' column of the table, specifically highlighting the entry for 'CLIENT1'.

- We can also see the CLIENT1 machine is now a computer!

Now if we go back to our Client1 Machine, we can check from there to see if the user is connected to the domain.

The screenshot shows a Windows Command Prompt window. The title bar says 'Command Prompt'. The window displays the following text:

```

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

C:\Users\ethorpe>whoami
mydomain\ethorpe
C:\Users\ethorpe>

```

- Type in, whoami
- If it comes up with mydomain\ethorpe, that means that you're using the account ethorpe from the domain, mydomain.
- This means the ENTIRE infrastructure works!! YAY!

You now have a full infrastructure that works. This is similar to how a school system or small business system works. Of course with more security measures.