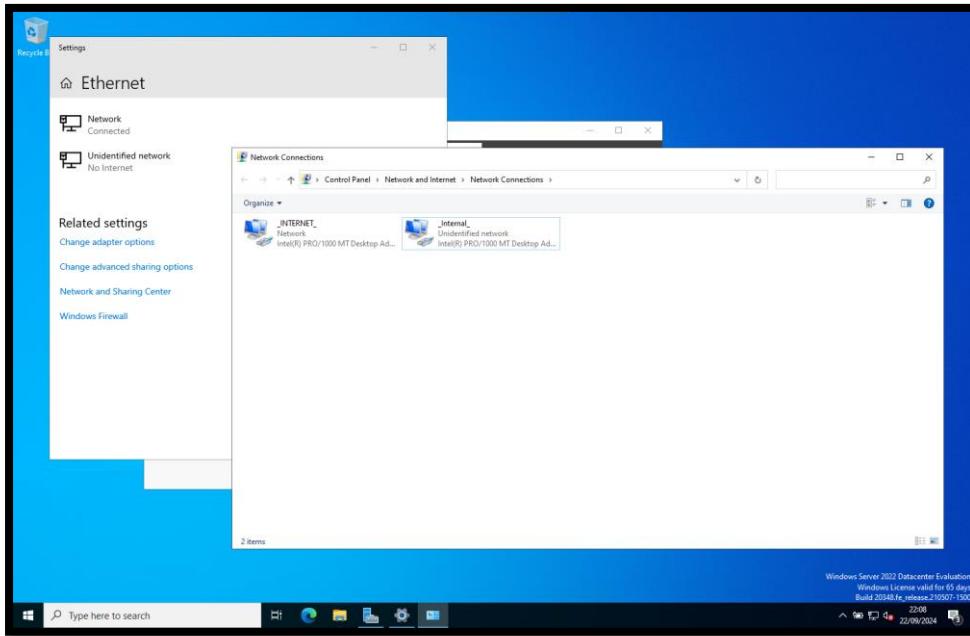
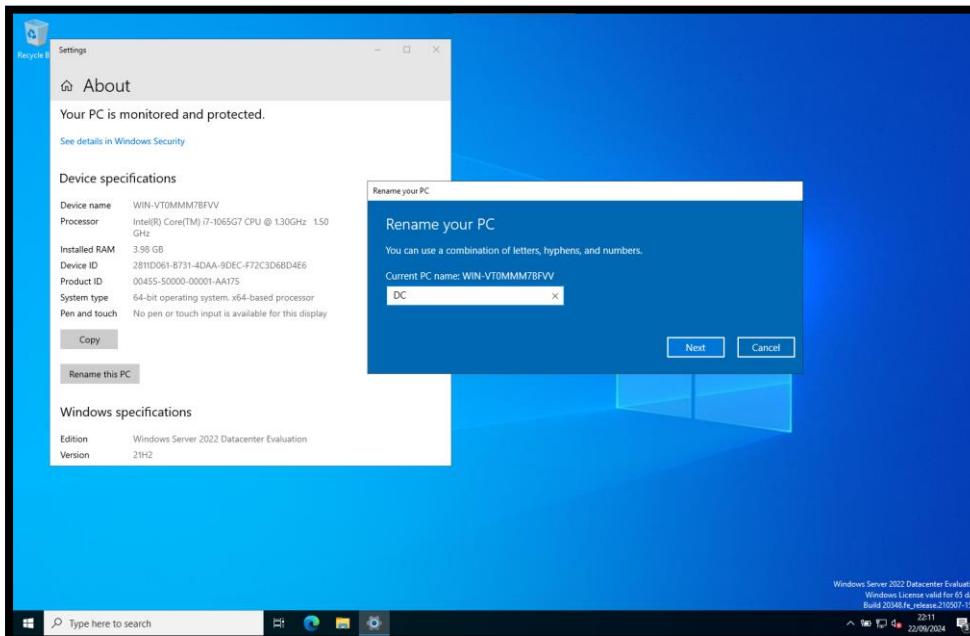


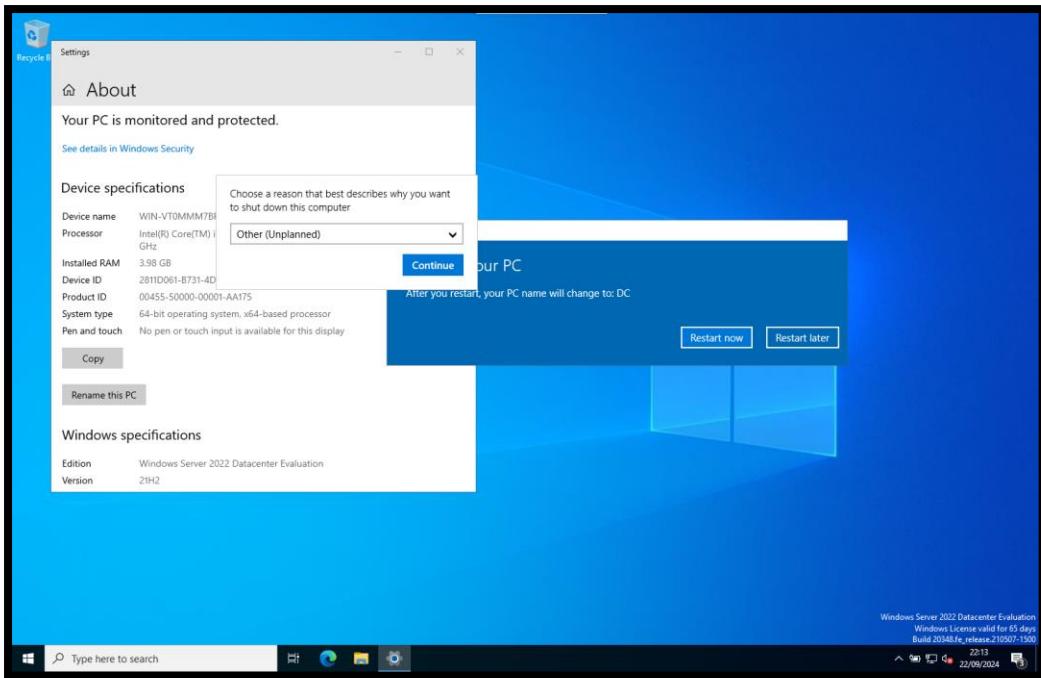
## Windows Server 2022 Lab



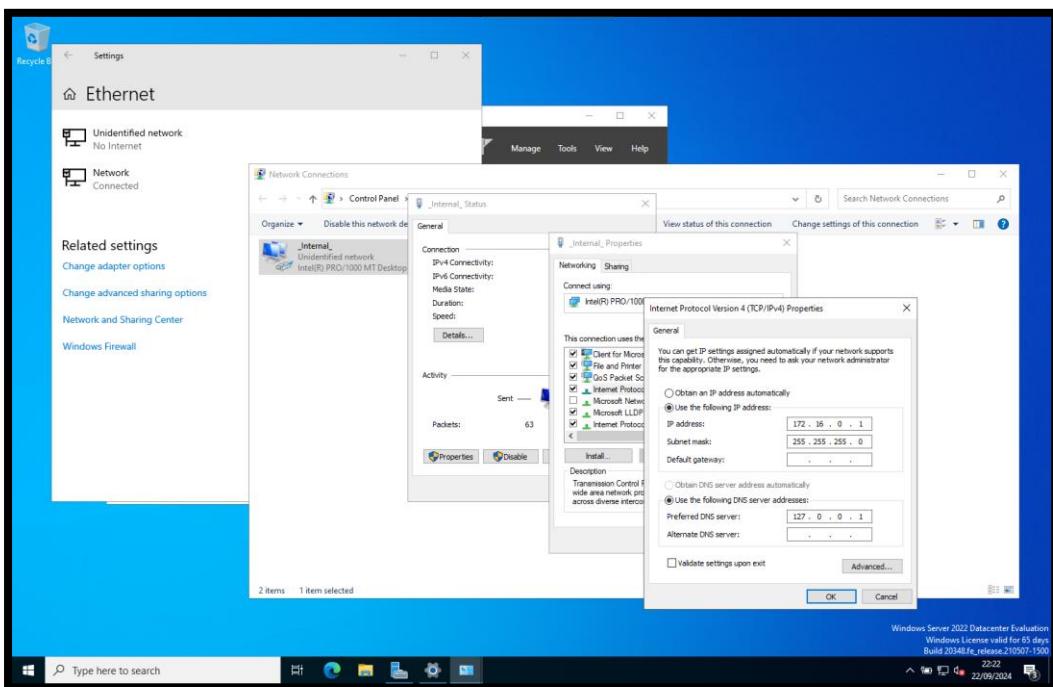
Setting up the internet and internal networks. Click on network icon near date.



Right click on start button to get the system and navigate to renaming the PC option.



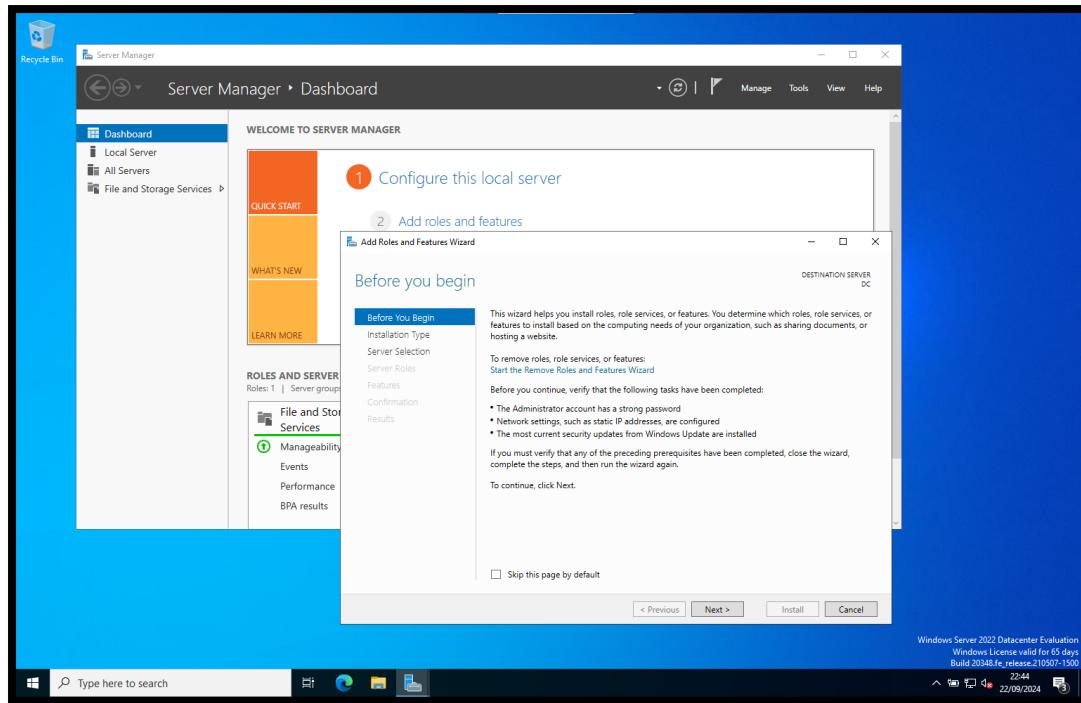
Restart PC.



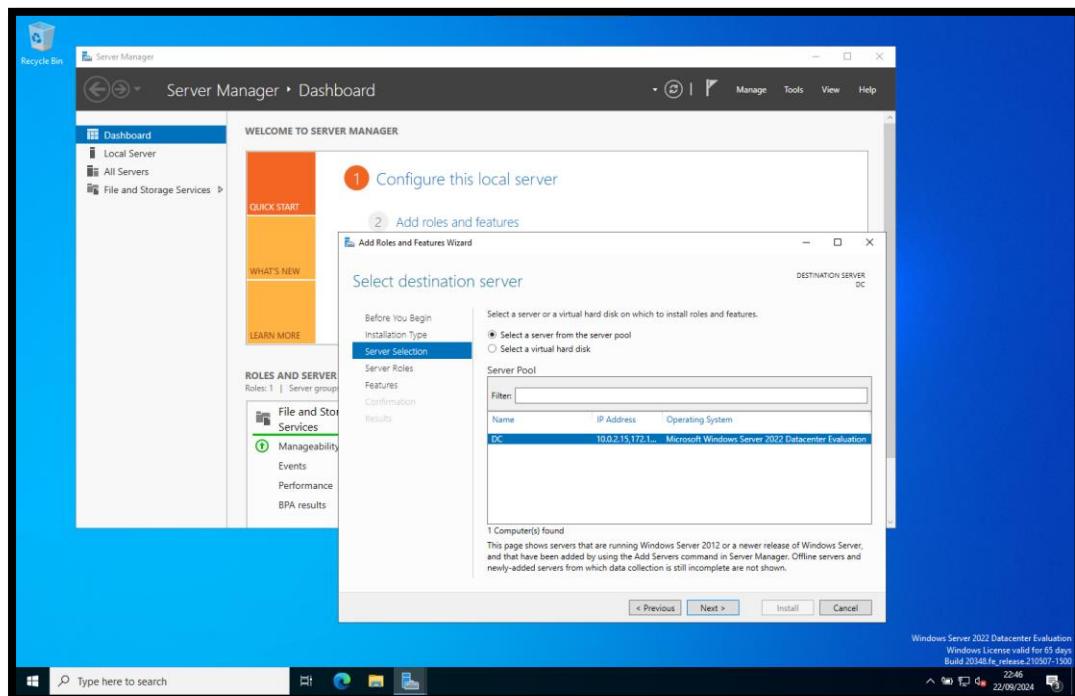
Setting up the IP address. Start by clicking on network icon near the date. Go to the internal network, select properties change the IPv4. Select use the following IP address. Assign 172.16.0.1 IP address, subnet mask 255.255.255.0. Not going to use the default gateway because the domain controller itself is going to serve as the default gateway.

For the DNS server, when I install active directory, it automatically installs DNS. So this server is going to use itself as the DNS server. To do this I can enter a loopback address of 172.0.0.1, it's like a generic address that refers to myself. So, whenever a computer pings like 127.0.0.1, they're pinging themselves automatically. Click ok, ok and close.

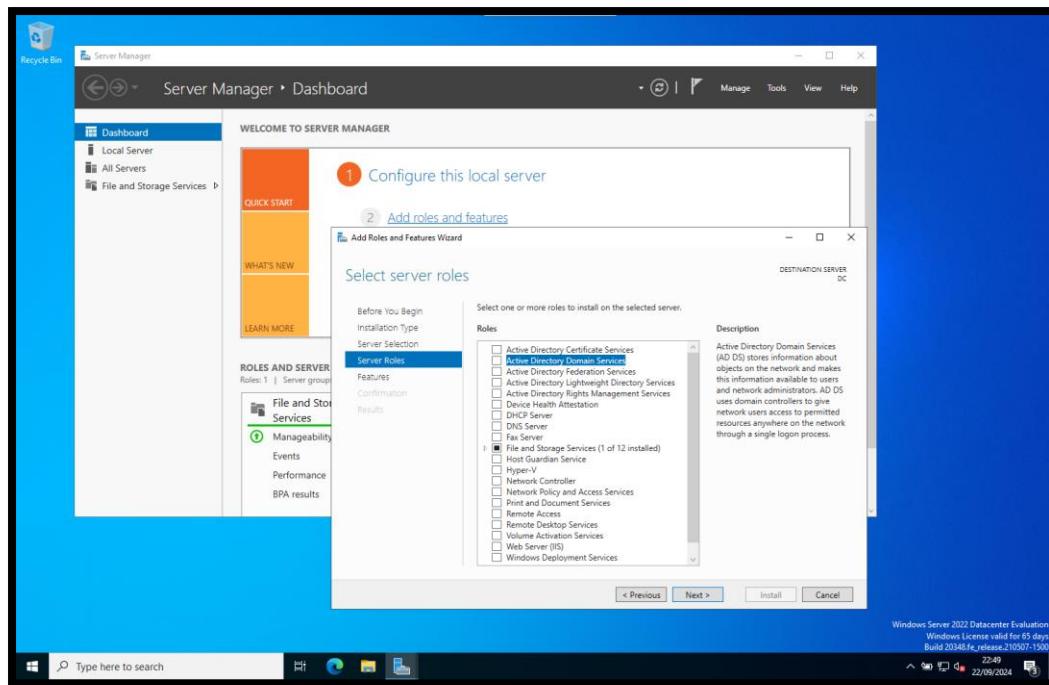
## Installing active directory and creating a domain



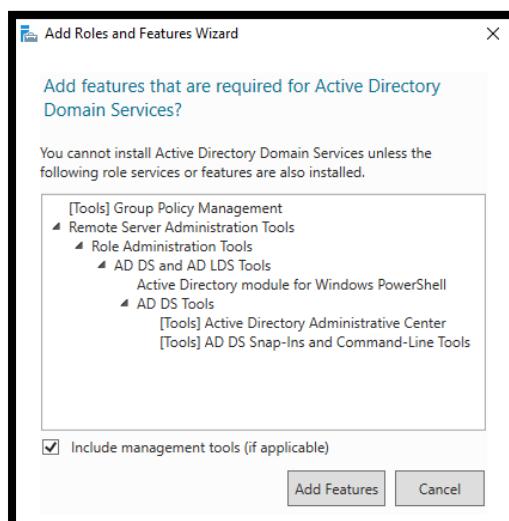
Add roles and features. Click next, next.



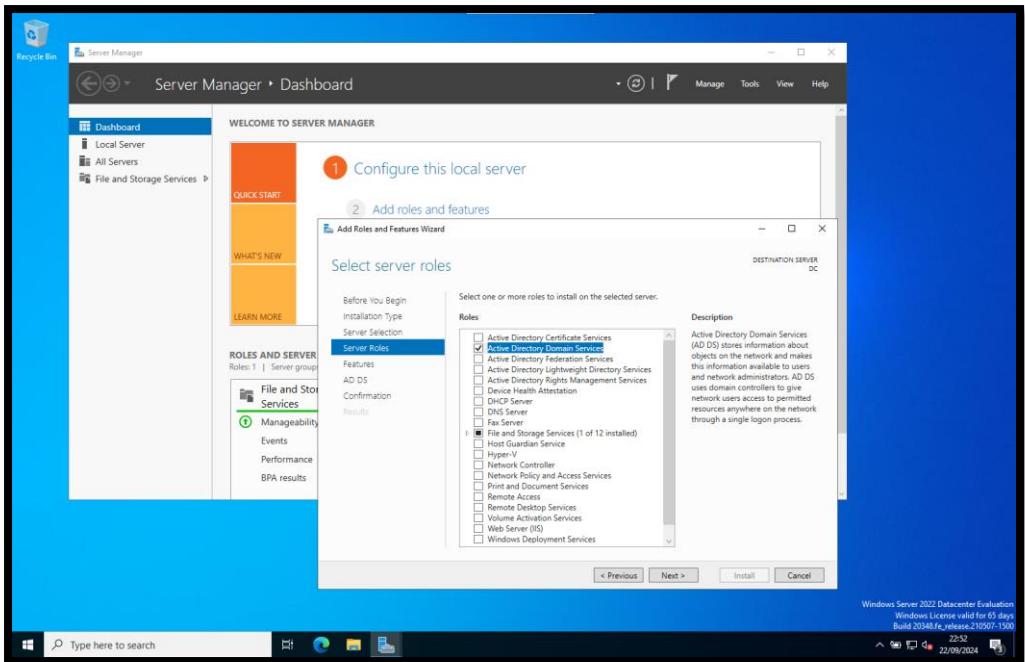
This is where you select a server to install active directory domain services, next.



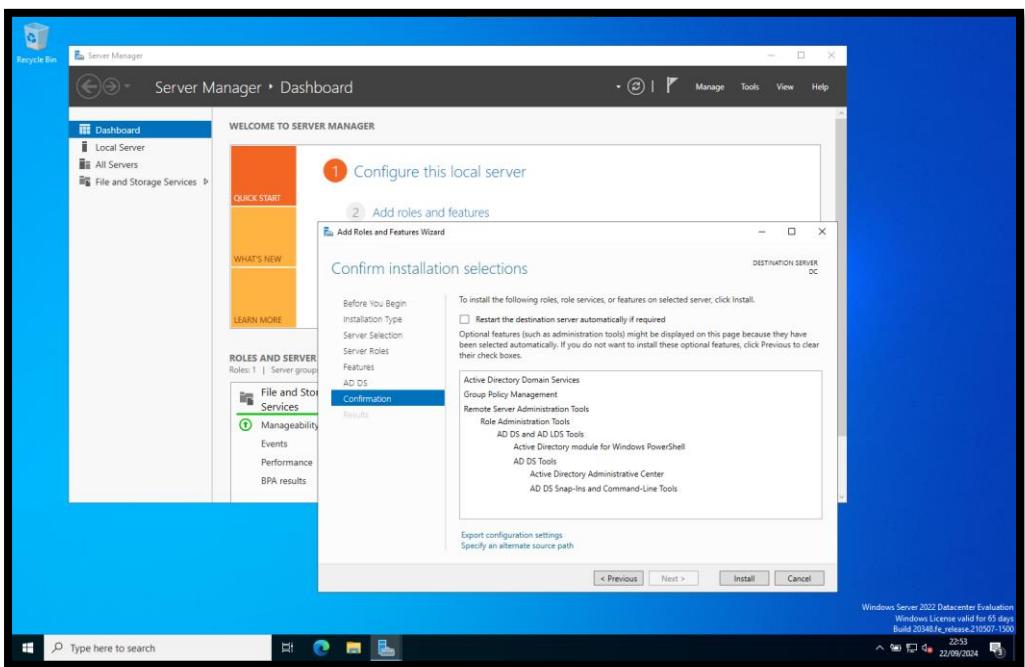
Select active directory domain services.



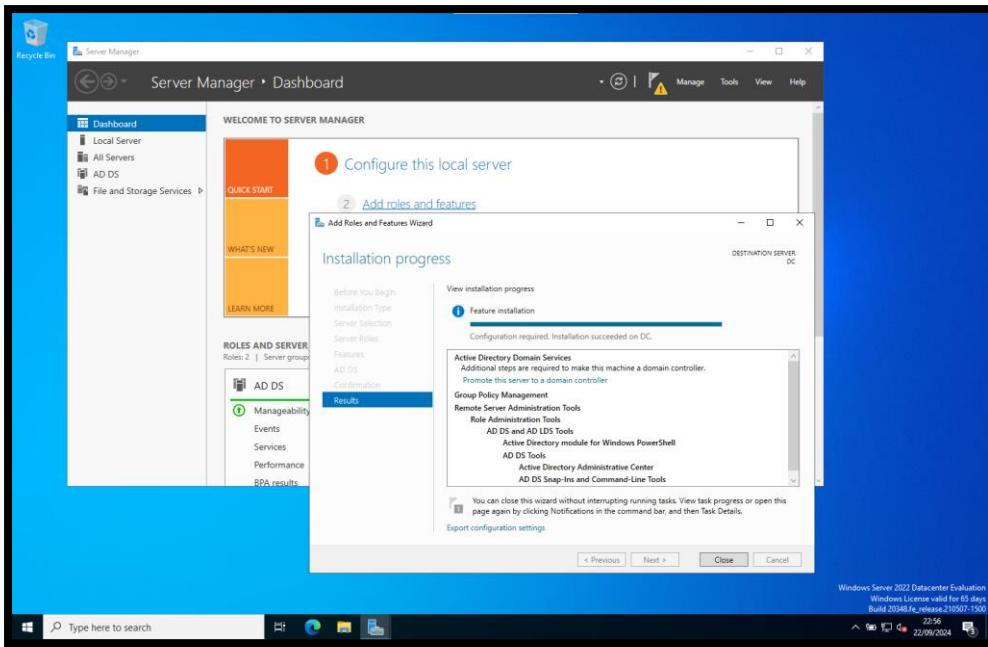
Pop up add features.



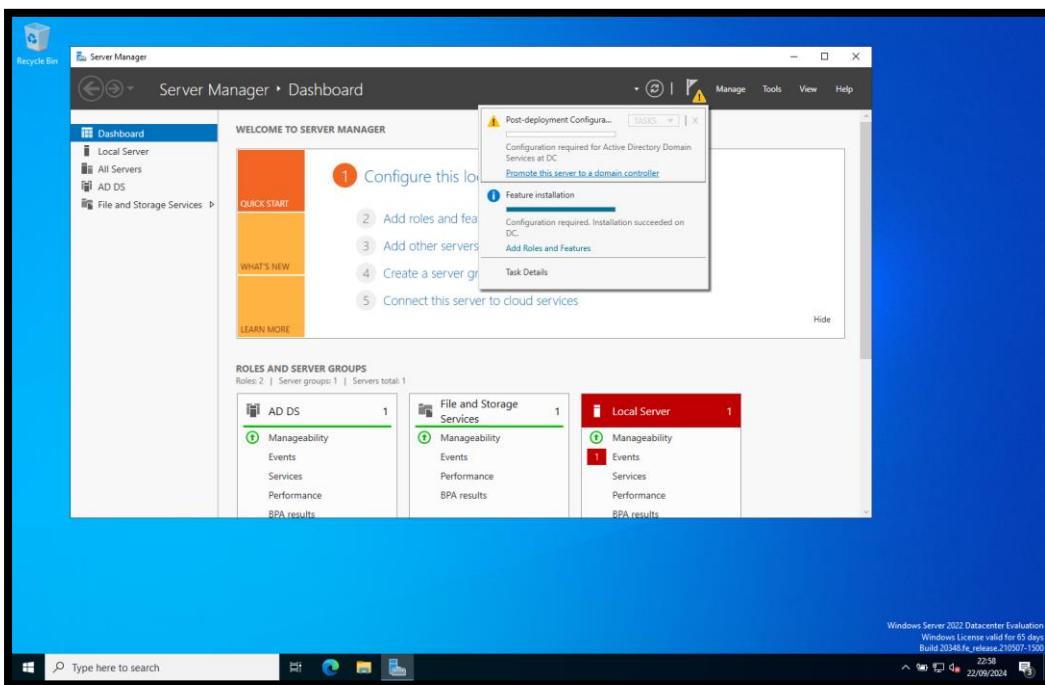
Next, next.



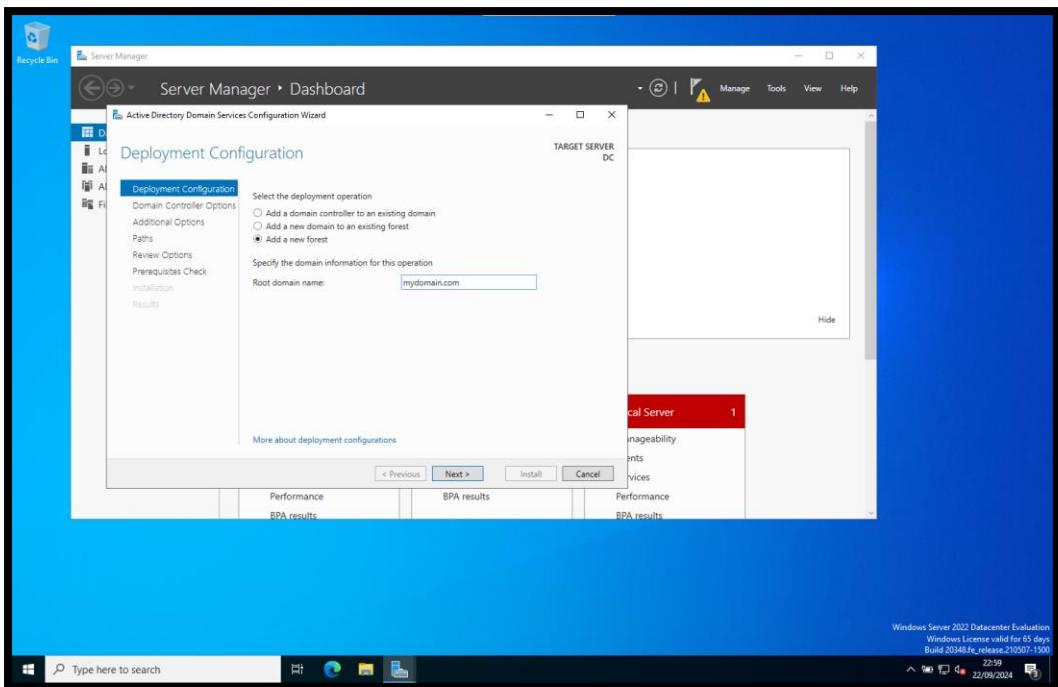
Install.



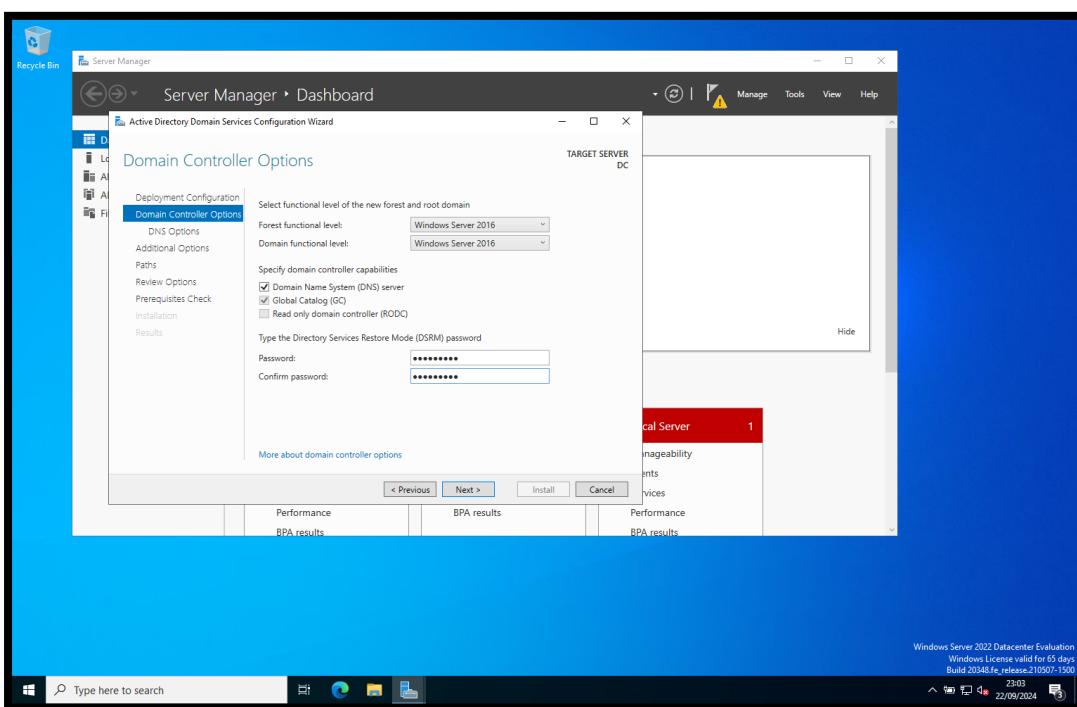
All installed close.



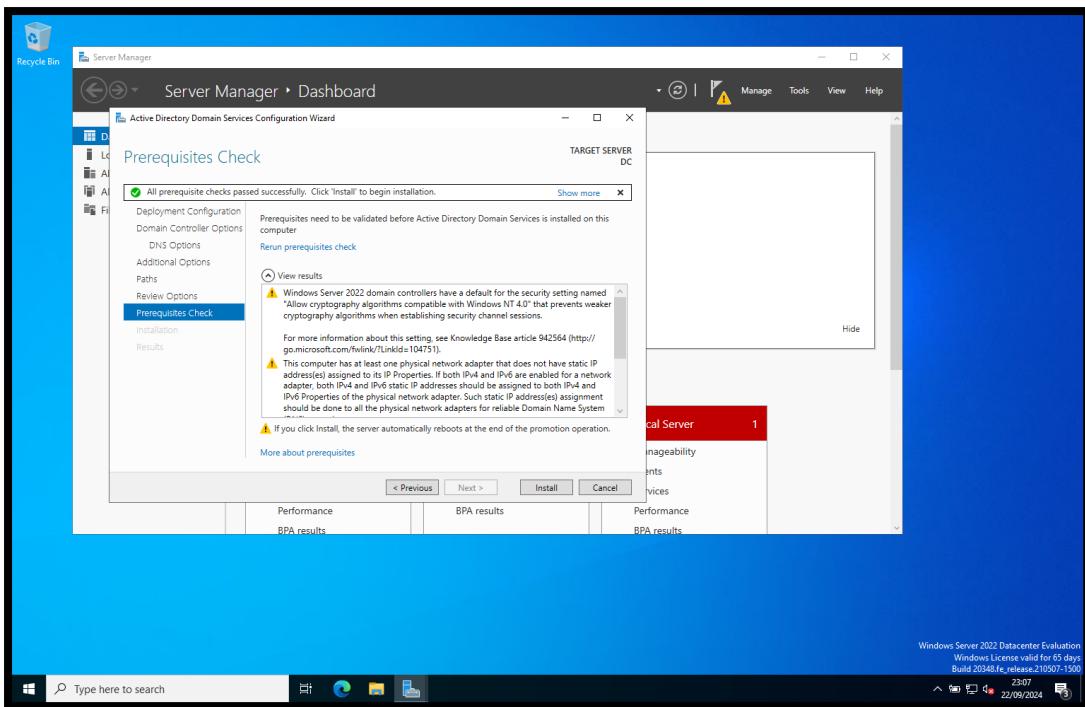
The software for the domain service is installed but the domain itself is not installed.



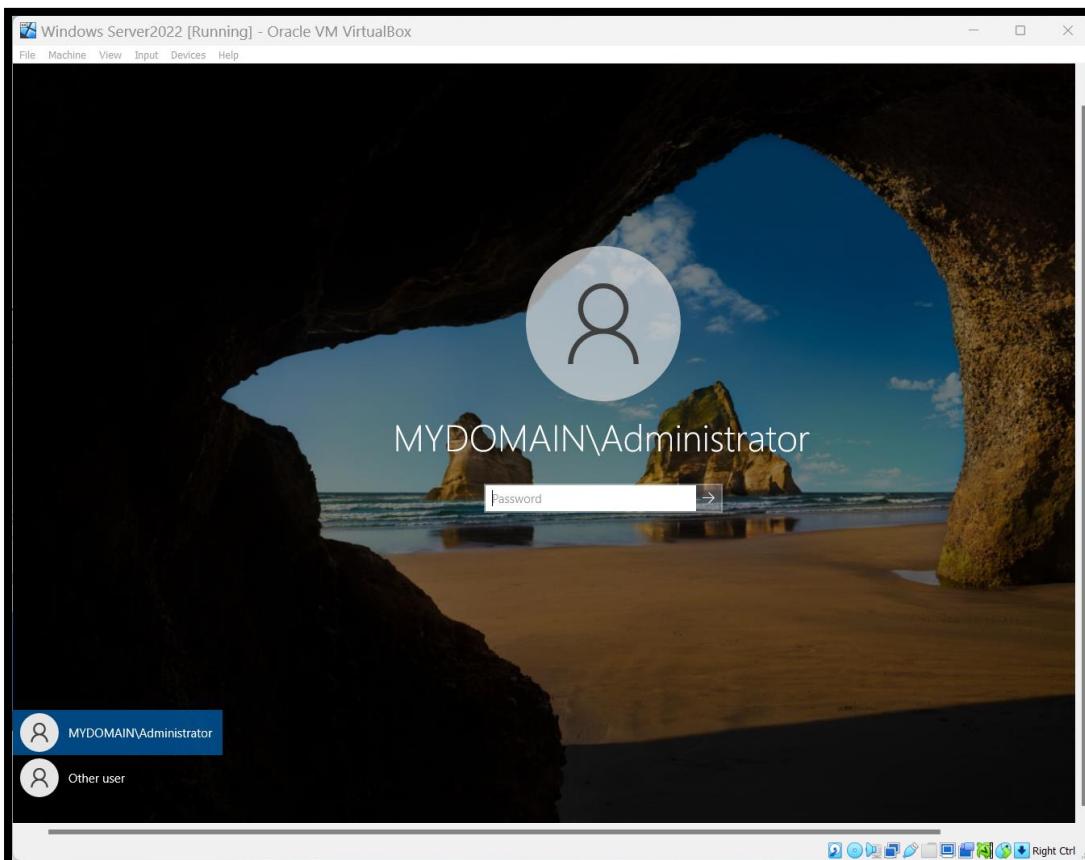
Select add a new forest and name the domain.



Next, next, next, next.

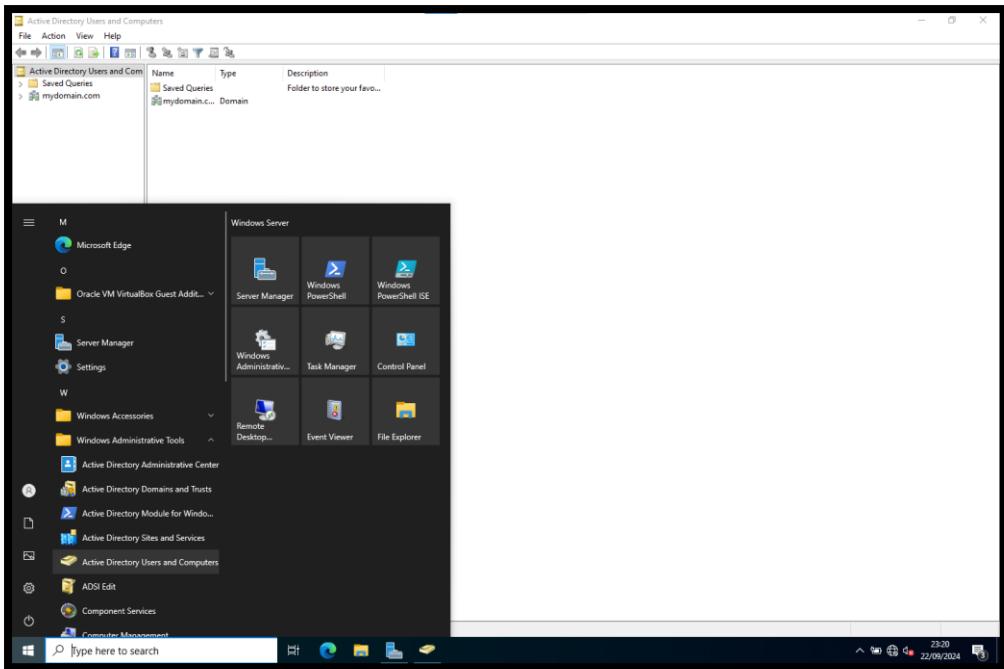


Install and after close for automatic restart.

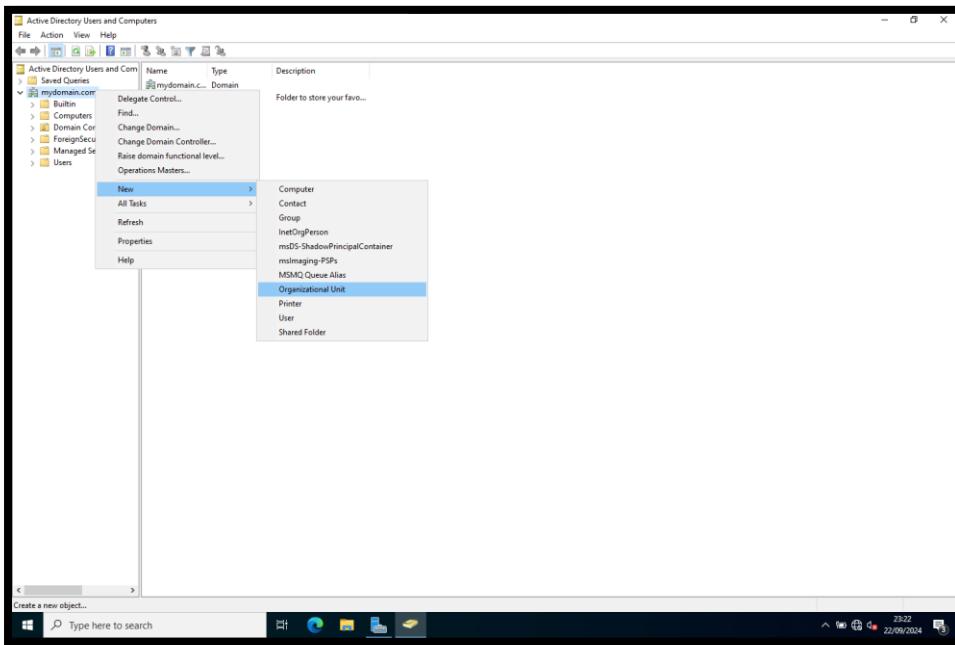


The domain/Administrator has now been created.

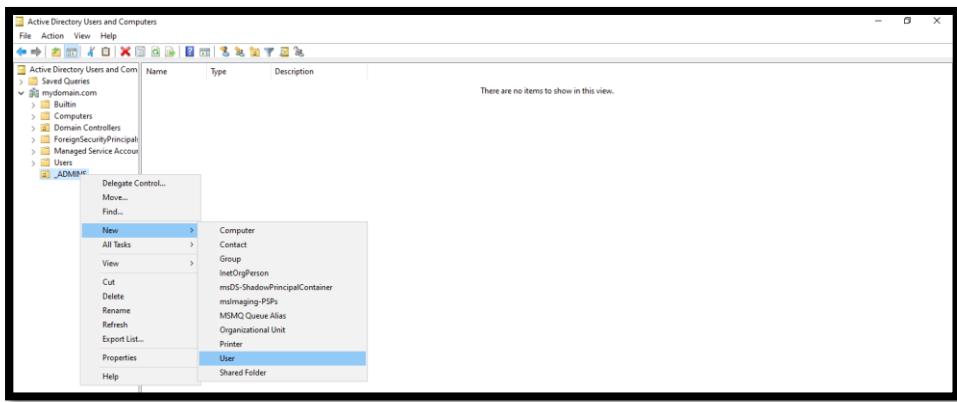
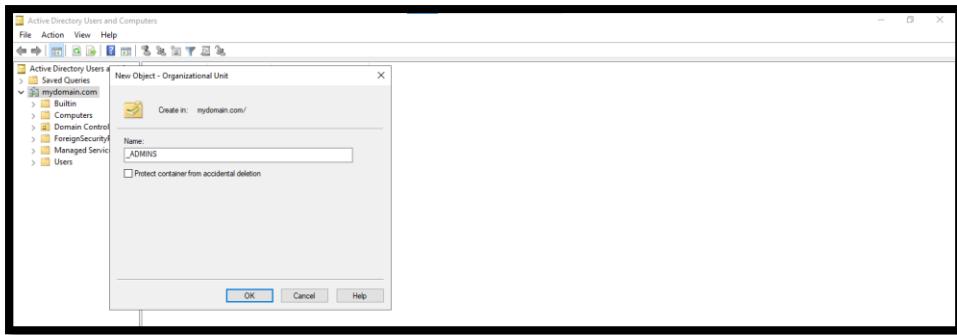
Now to create my own dedicated domain admin account instead of using the built in administrator.



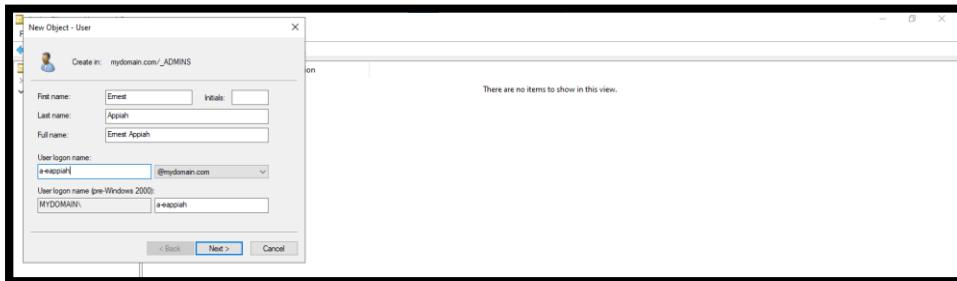
Go to start Windows Administrative Tools, select active directory users and computers.



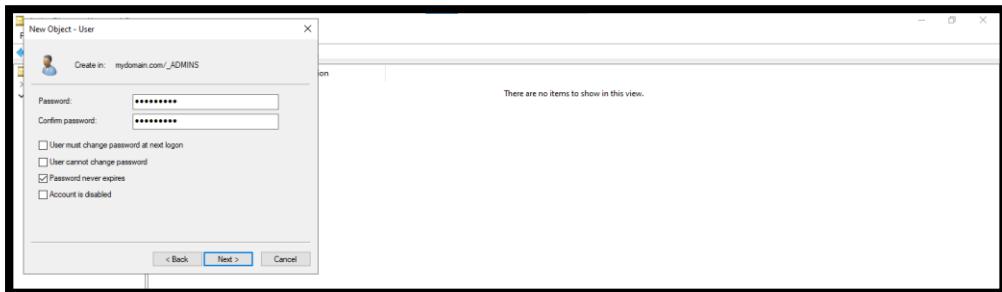
Let's create an OU to put the admin account in. OU: Organizational Unit this is like a folder in active directory. Right click on mydomain.com go to new, then Organizational Unit.



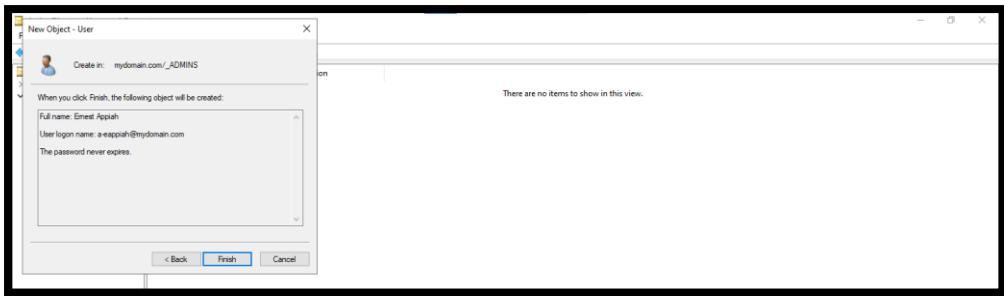
Inside of here create a new user.



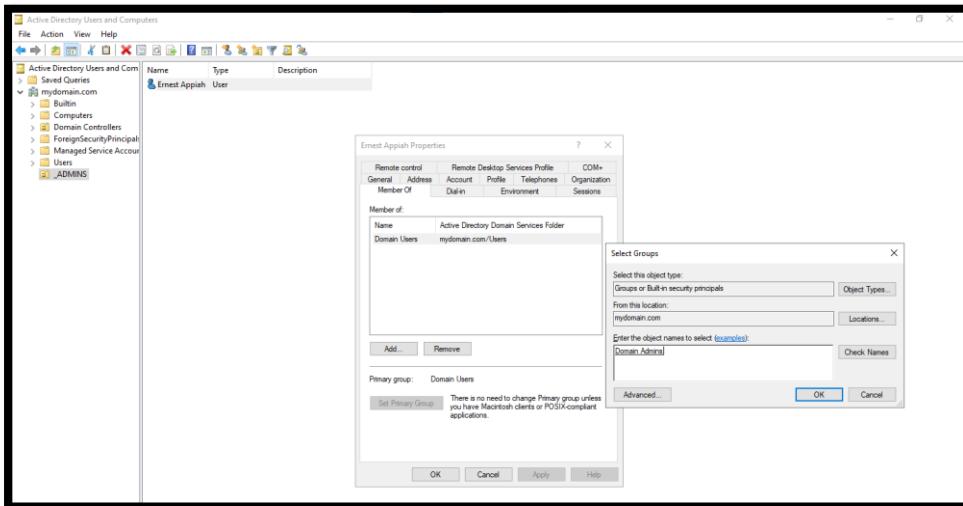
a-my name a means admin account.



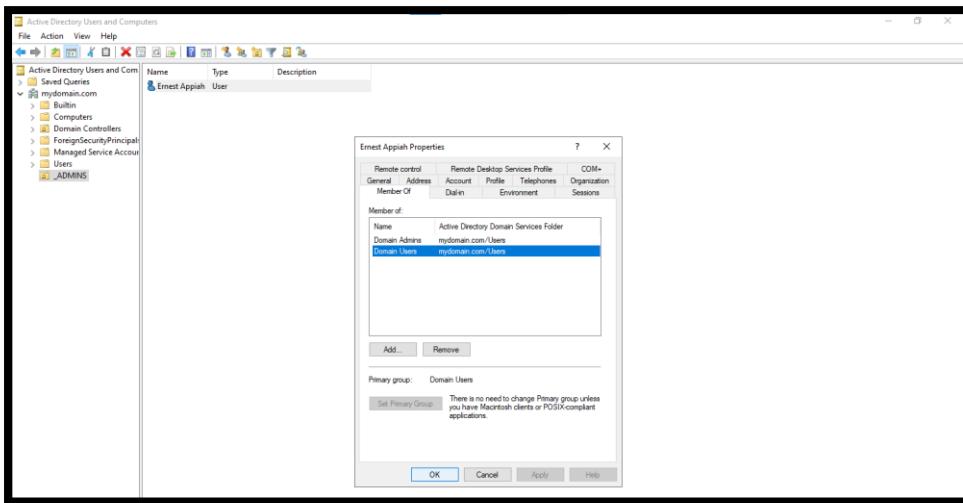
I selected password never expires just for this occasion, select next.



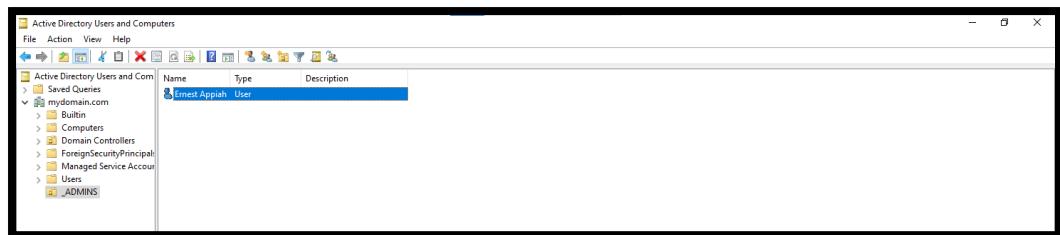
Finish set up.



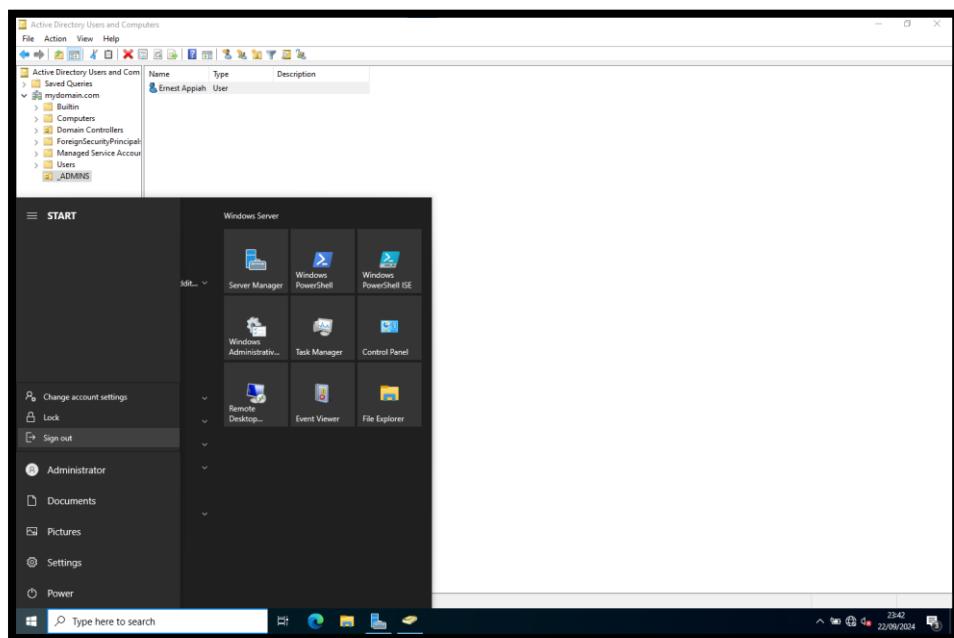
Now to turn this into an admin account select member of, add, type in domain admins tap check names and ok.



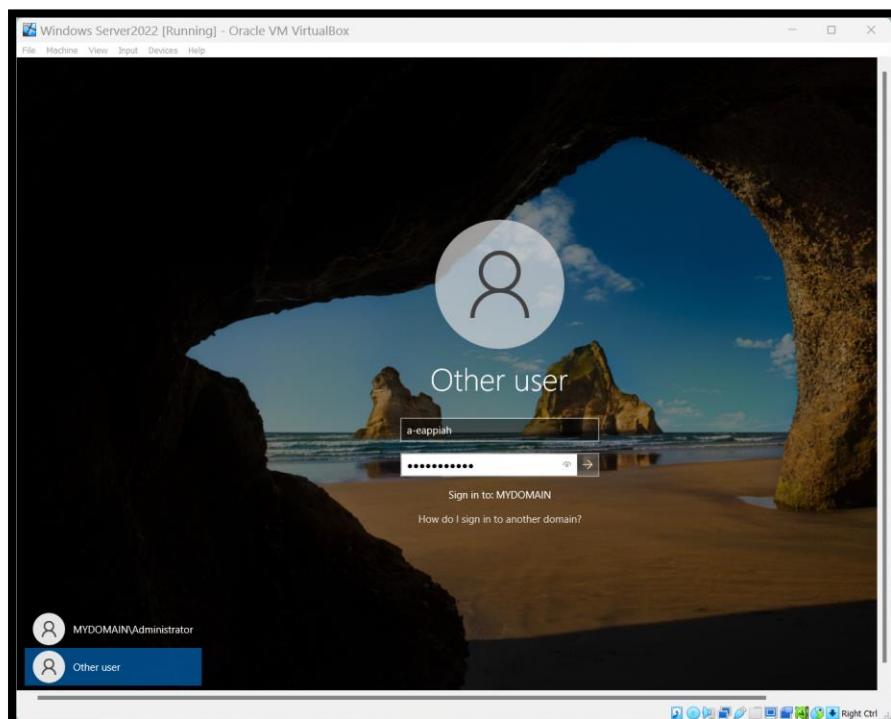
Apply ok.



Now I have my own admin domain account.

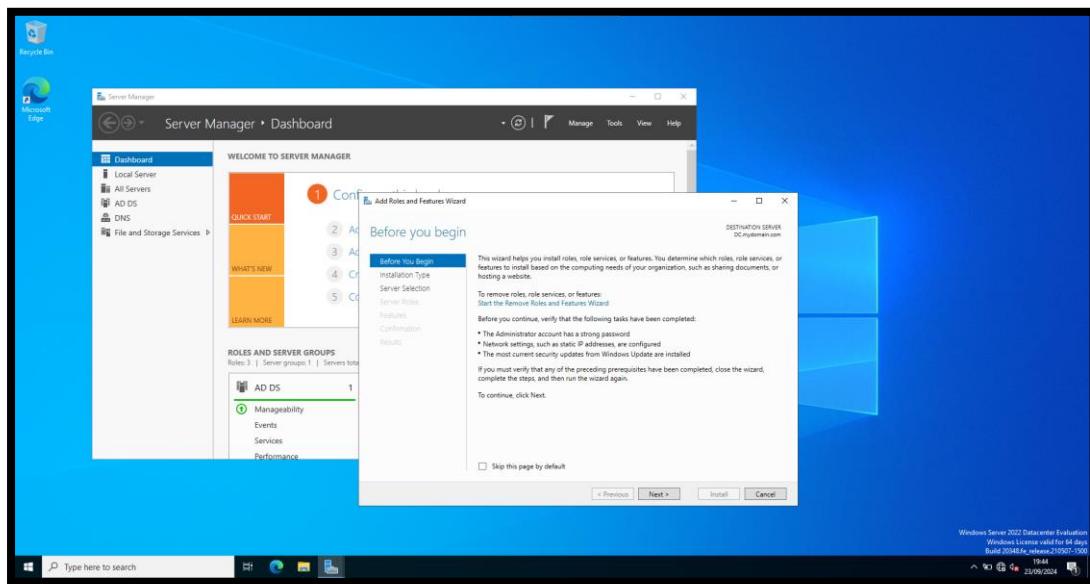


Sign out.

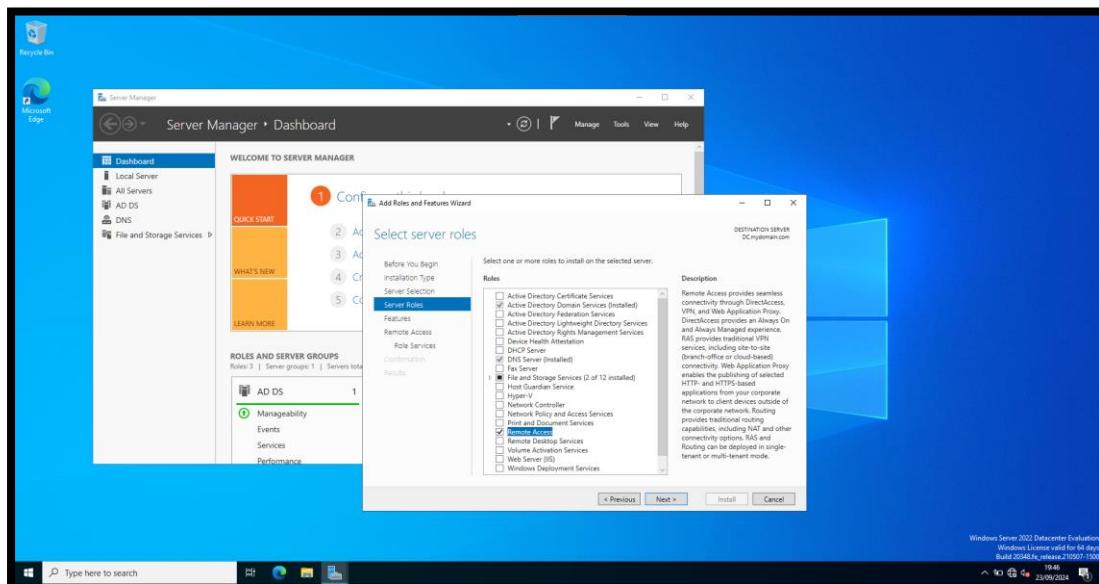


I signed into my admin account using other user option.

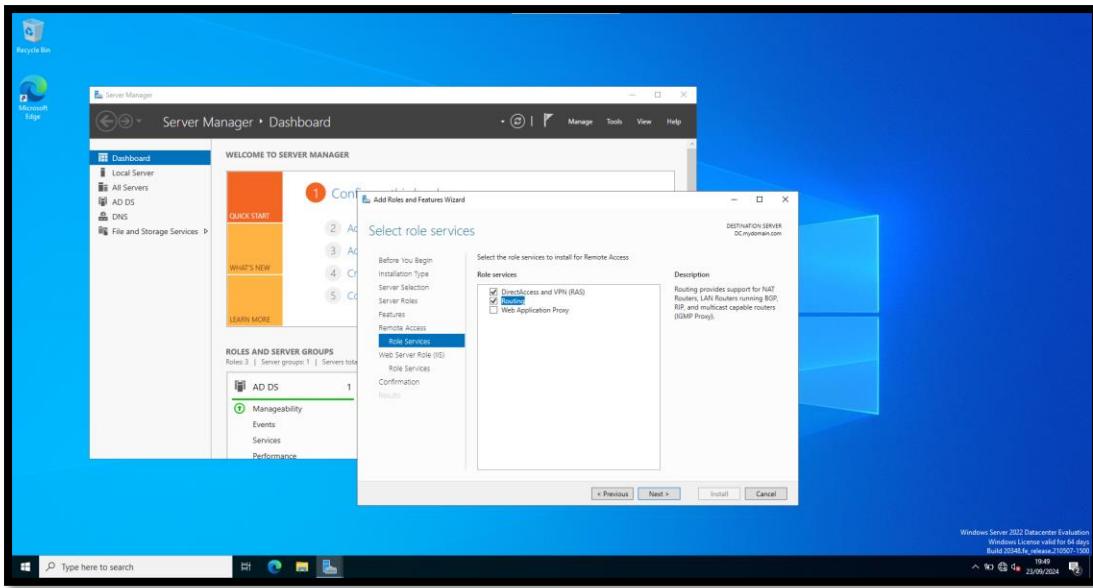
Next to install remote access server and network access translation.



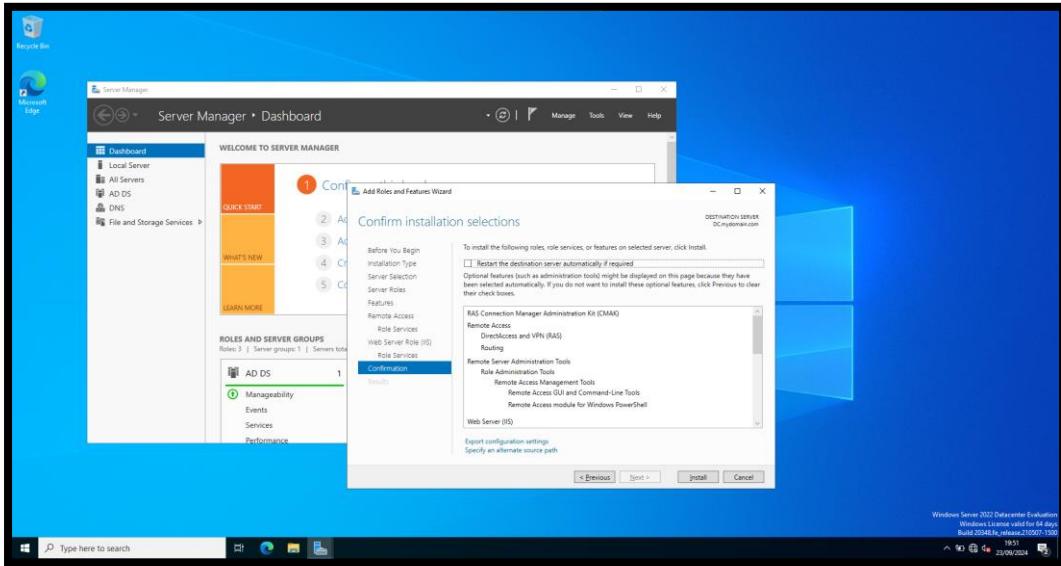
Add roles and features next, next



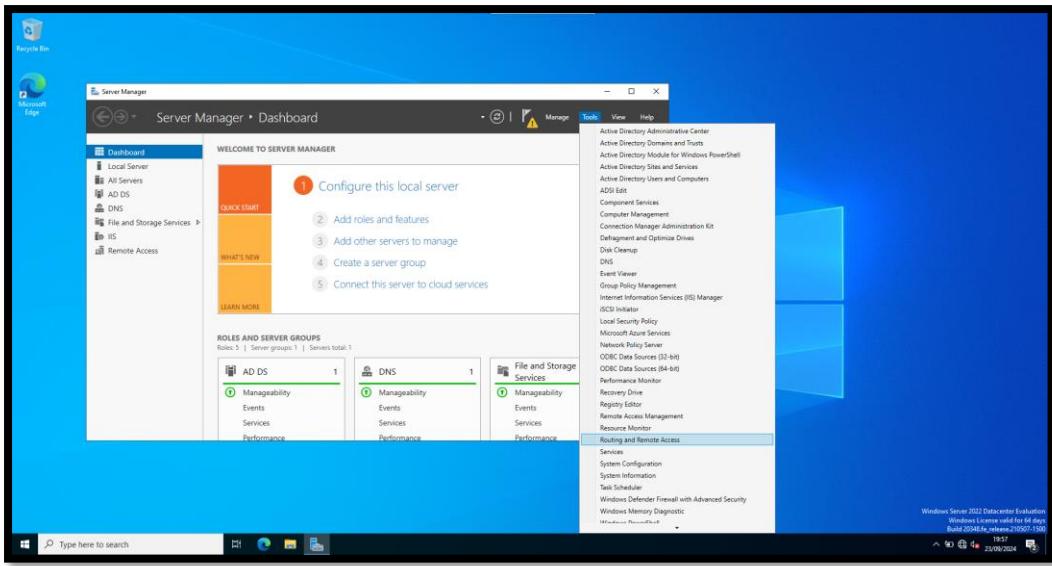
Remote access



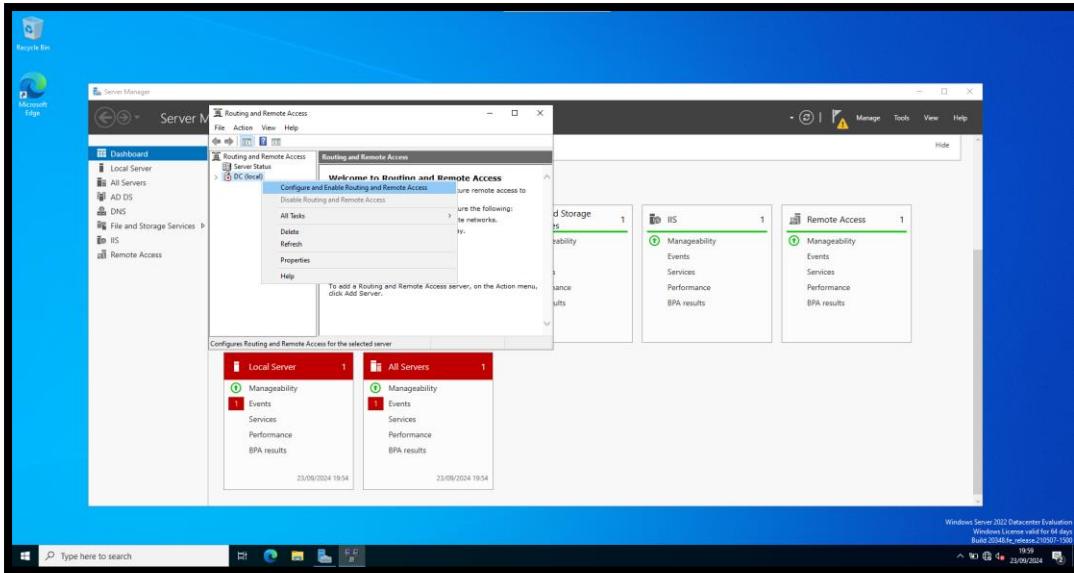
## Routing



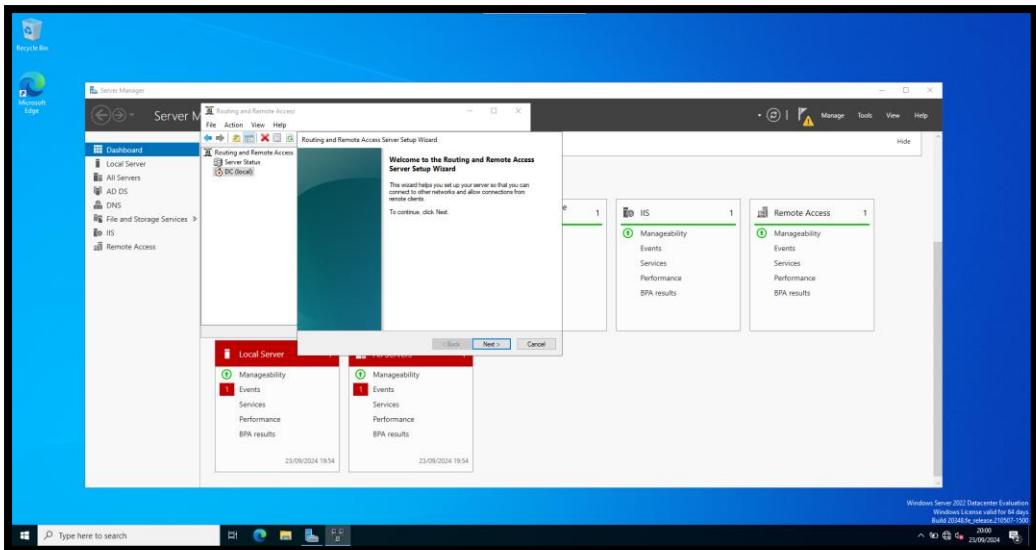
## Install



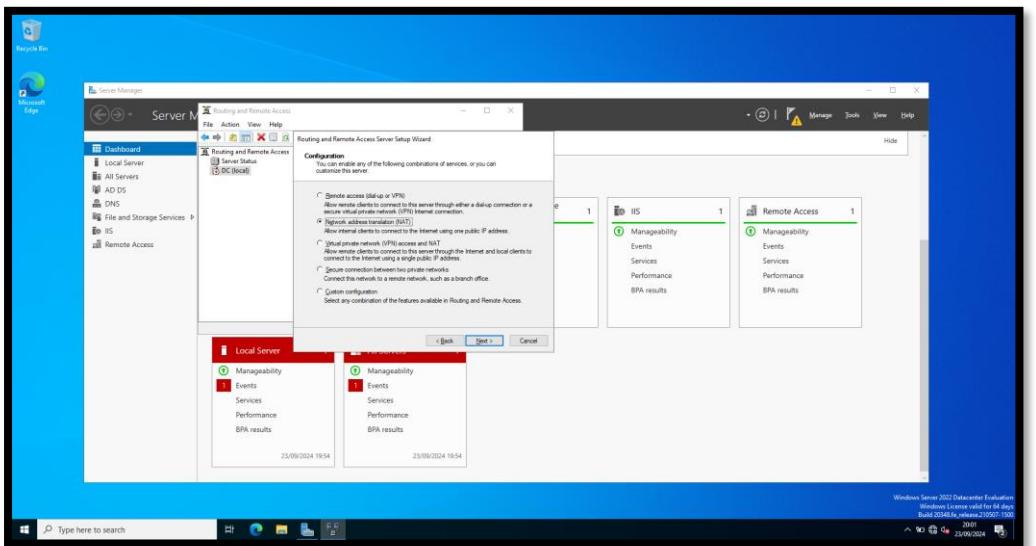
Go to tools routing and remote access.



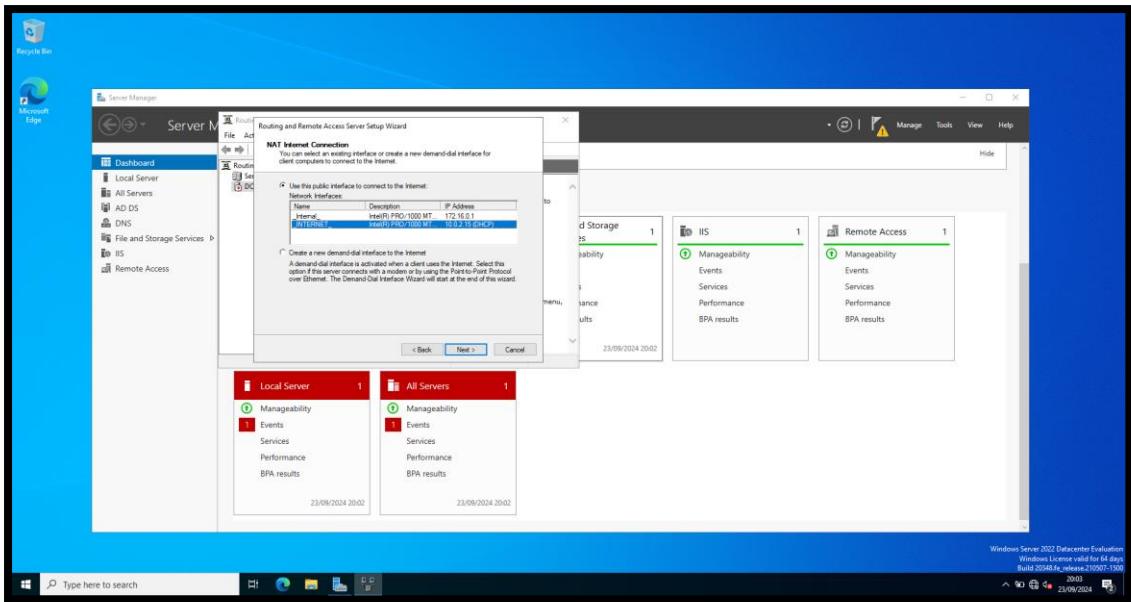
DC local configure and enable routing remote access.



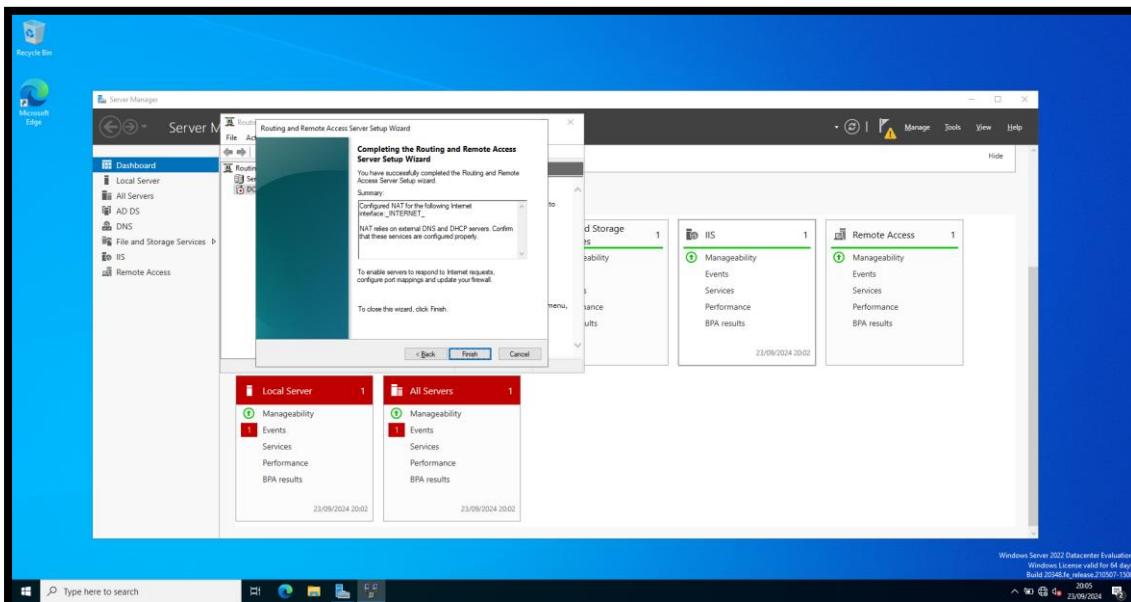
Next



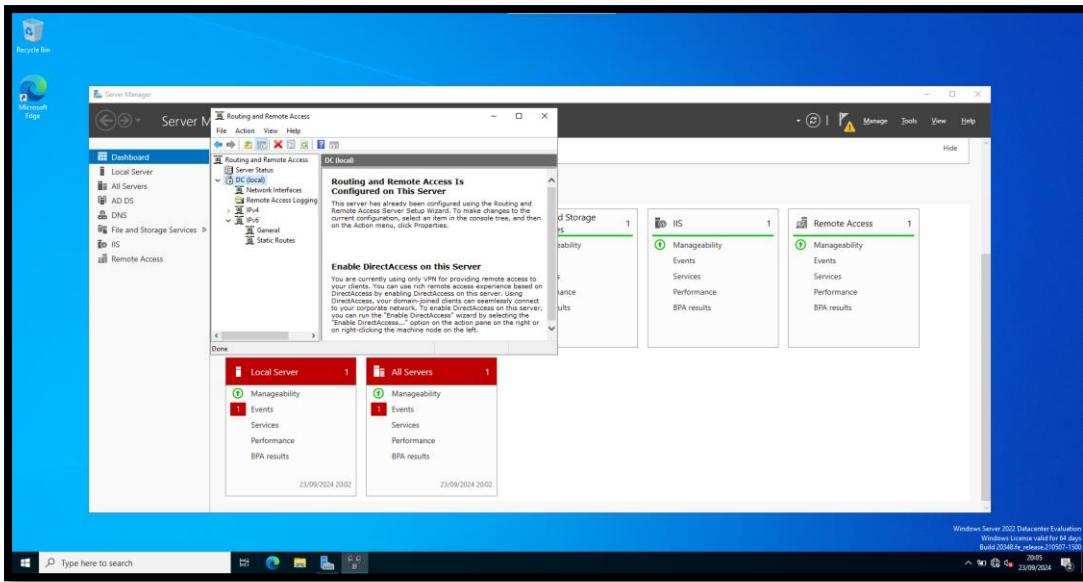
Enable NAT, next.



Select the one I called internet to connect to the internet.



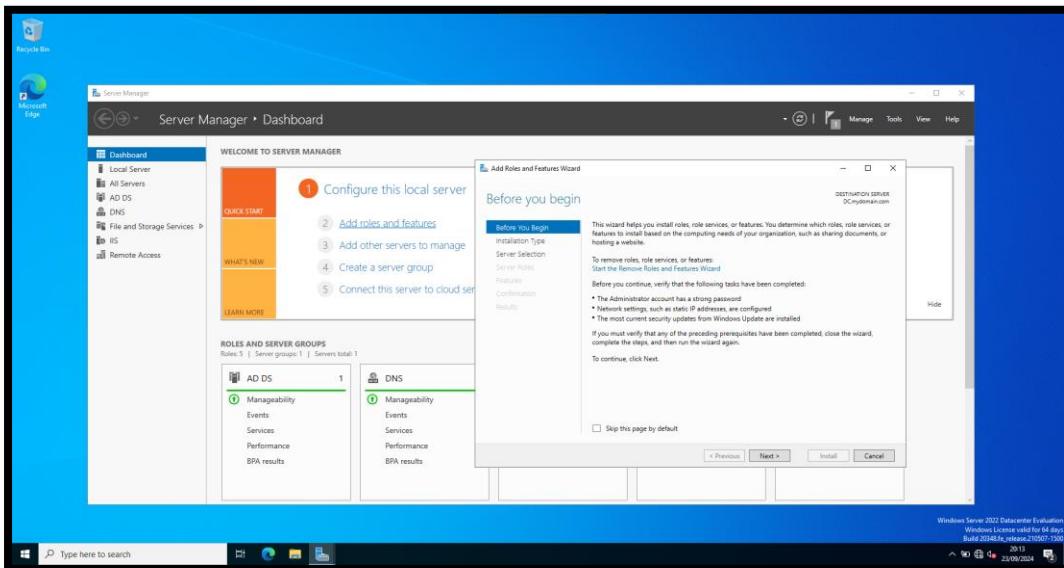
Finish



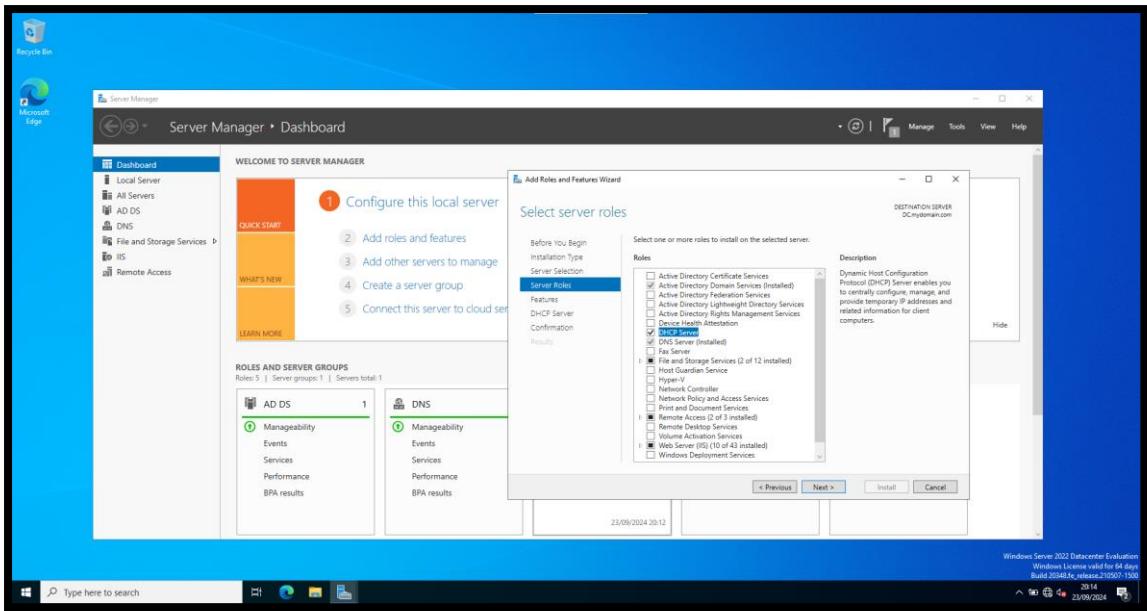
The DC local is gone from red to green, so this is configured.

Set up DHCP server on our domain controller

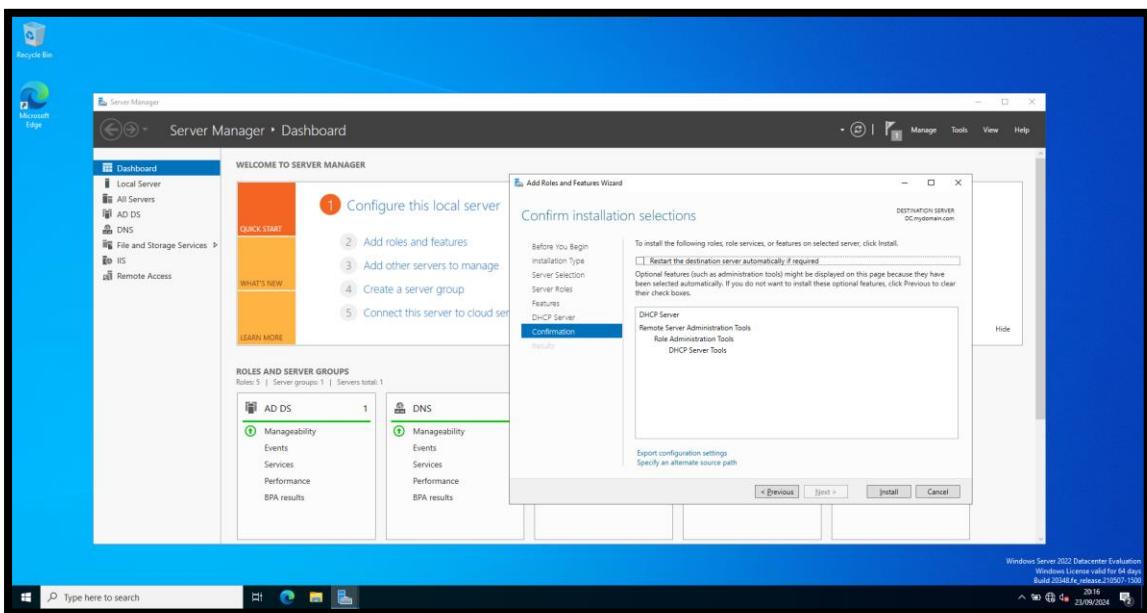
This will allow the windows 10 clients get access to the internet.



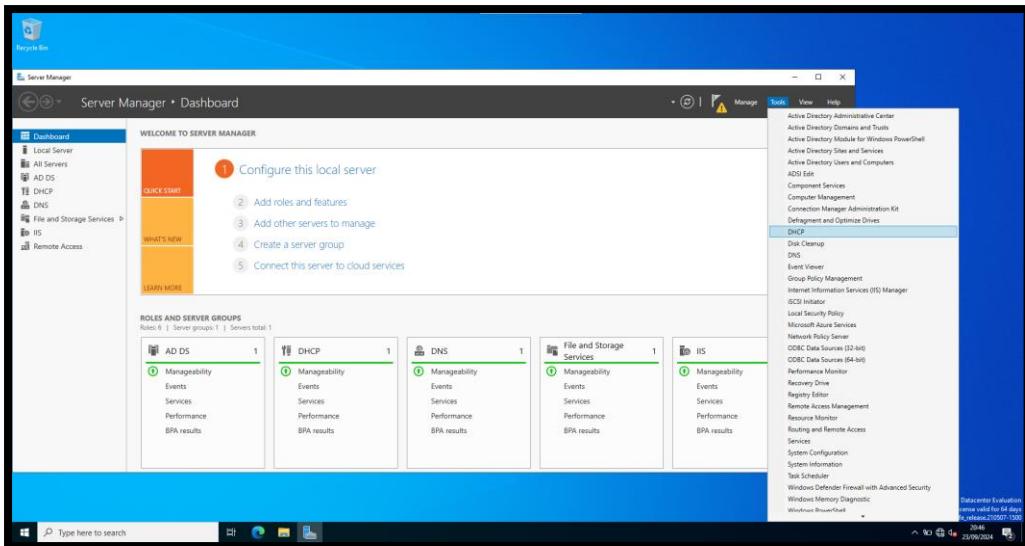
Go to add roles and features, next, next.



Select DHCP server, next, next.

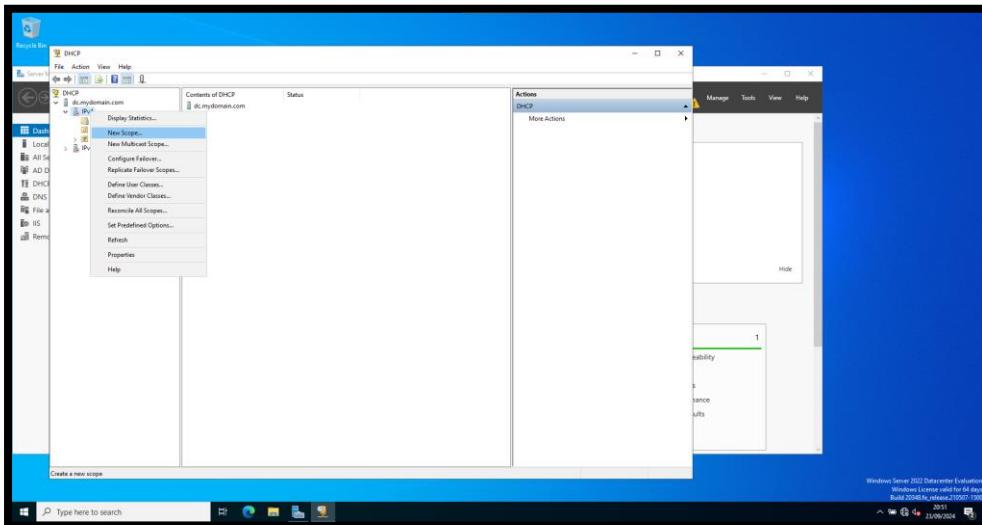


Install

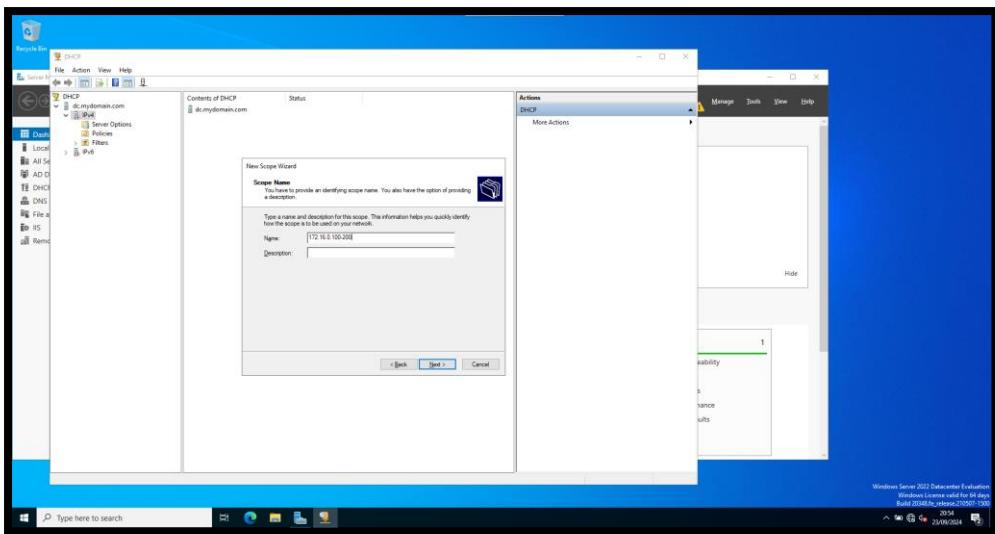


Tools DHCP the whole purpose of DHCP is to allow computers on the network like client computers on the network to automatically get their IP addresses.

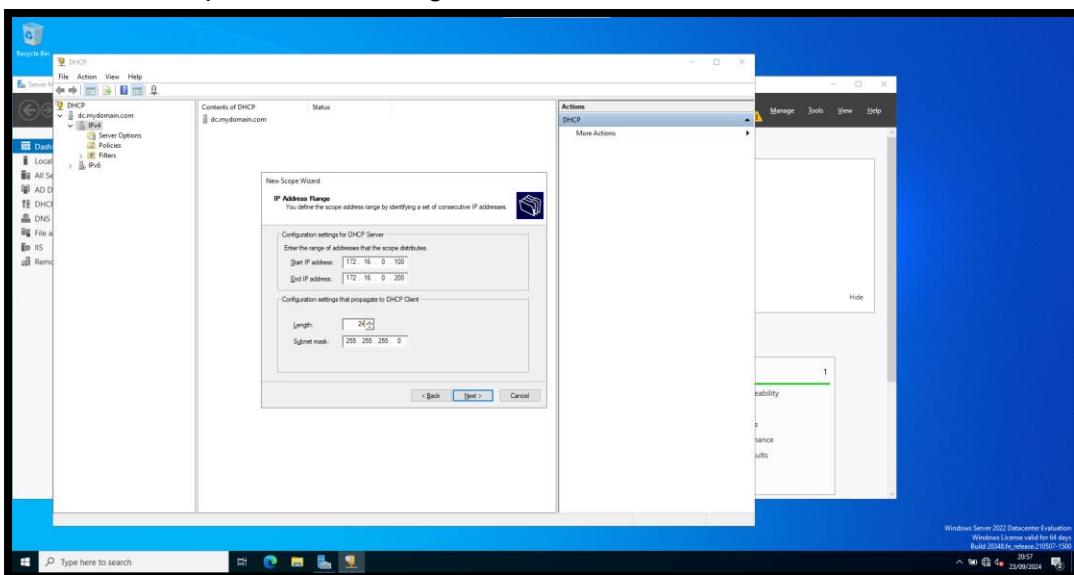
We going to create scope that will give the IP addresses in this range this 172.16.0.100 with this subnet mask.



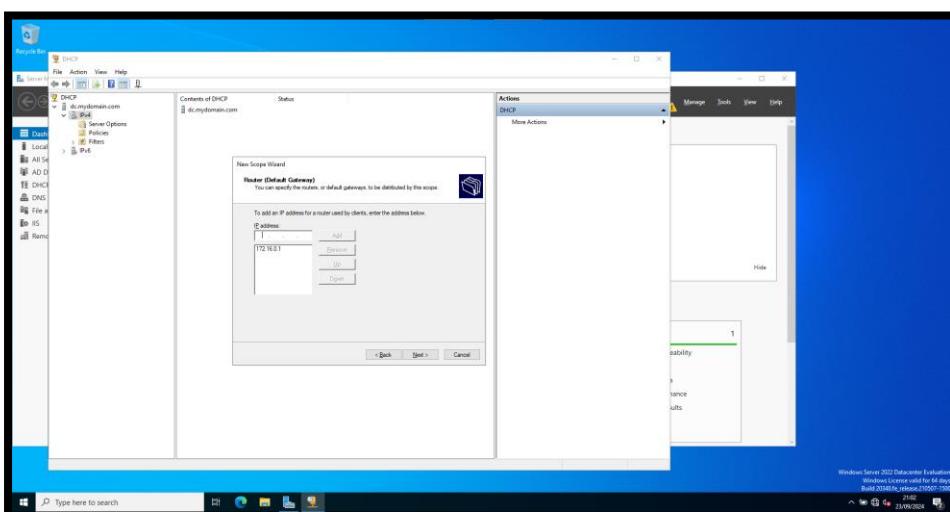
IPV4 right click new scope.



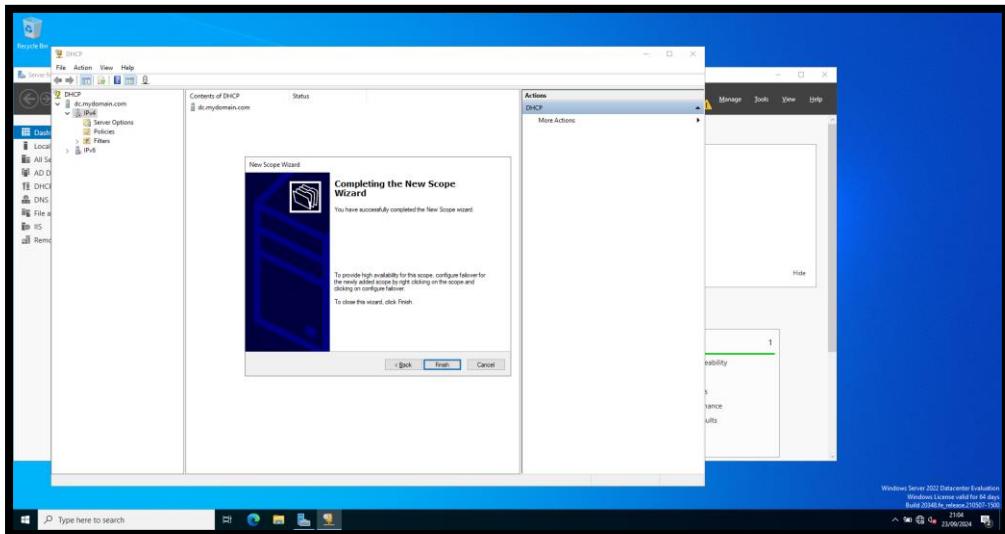
For the new scope use the IP range 172.16.0.100-200, next.



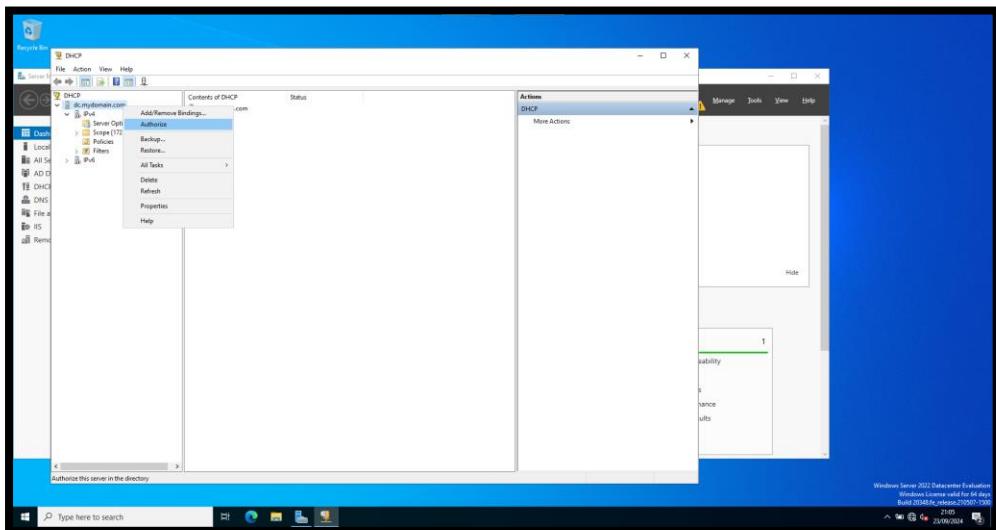
The 172.16.0.100-200 one for 100 and the another to 200 with a length of 24 subnet mask, next, next.



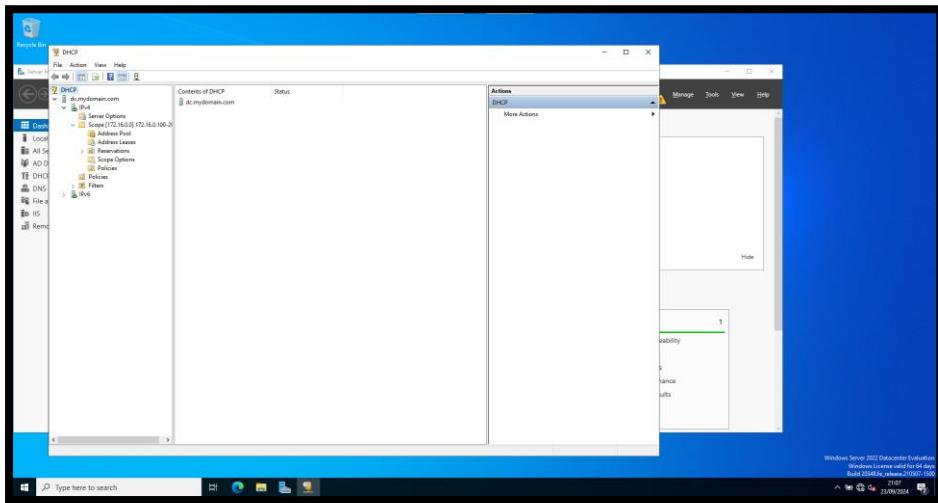
Type in IP address 172.16.0.1 click add, next, next, next.



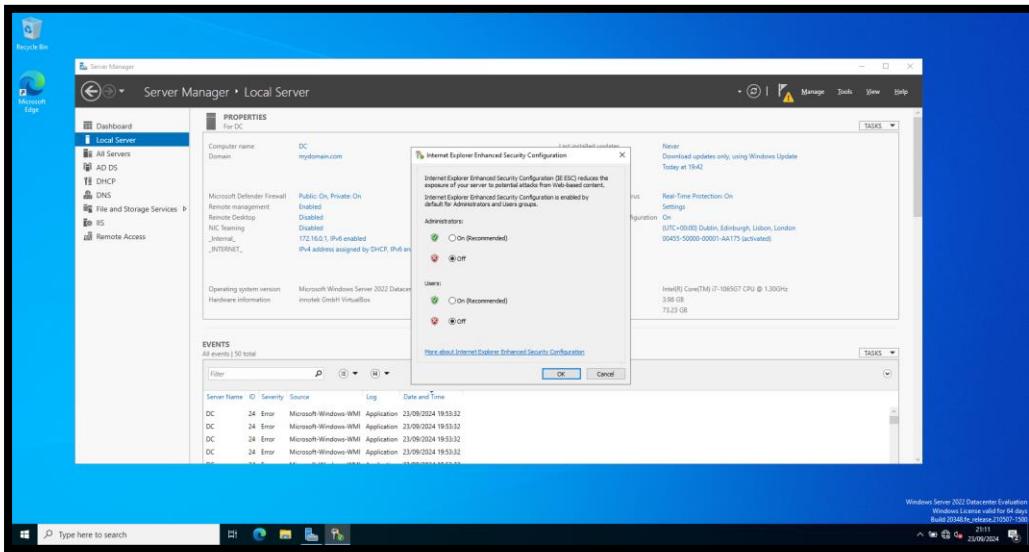
Finish



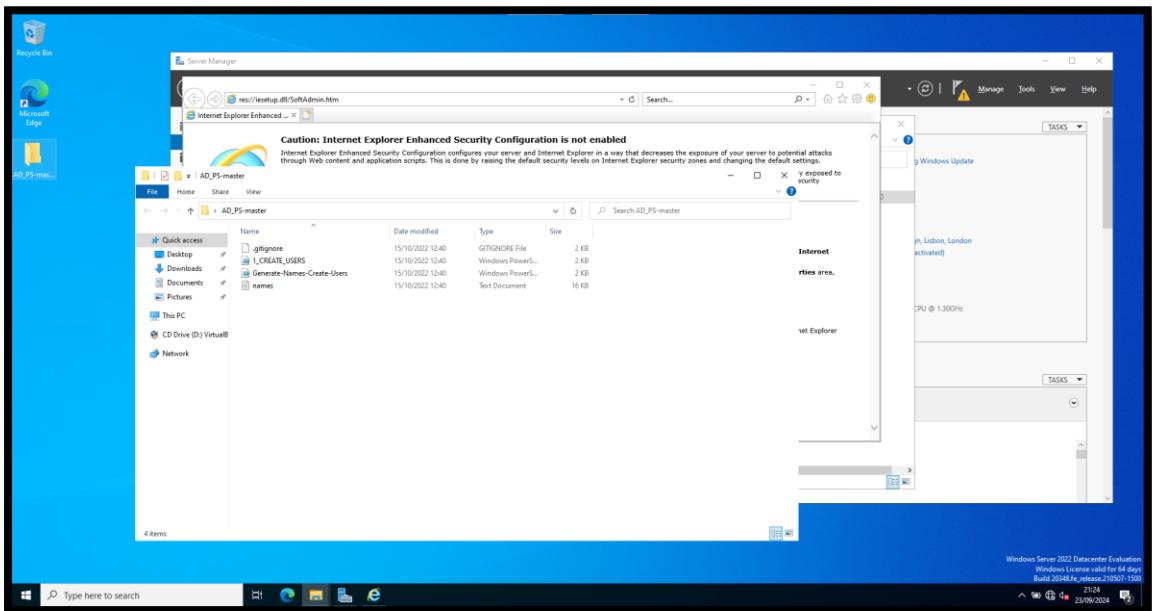
Right click on domain, select authorize then refresh.



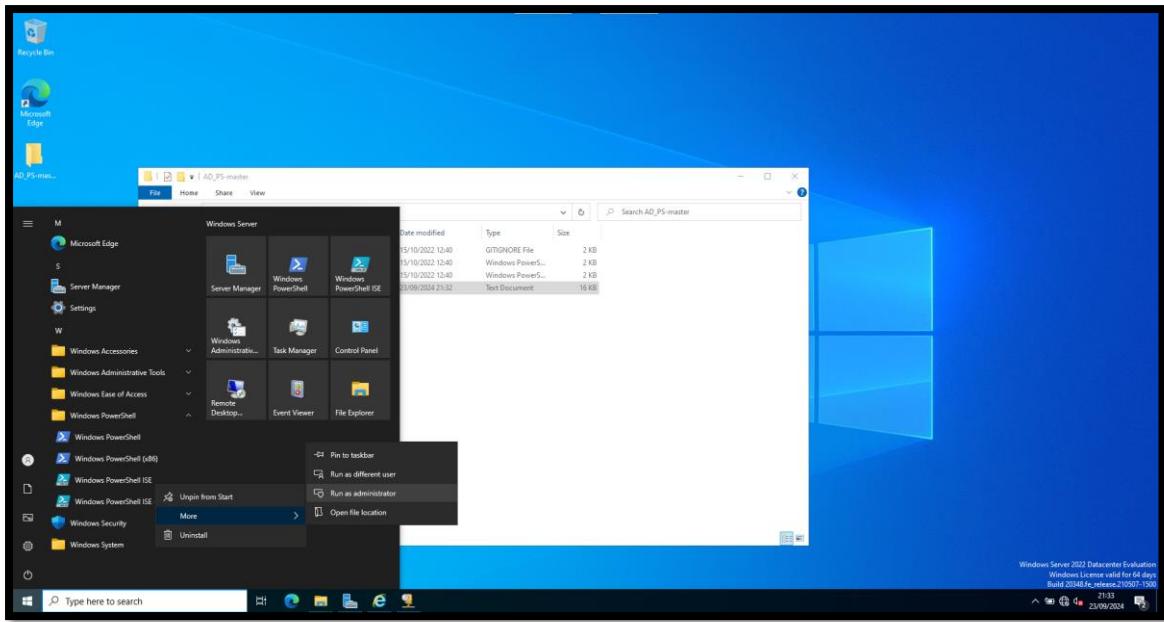
The scope has been configured.



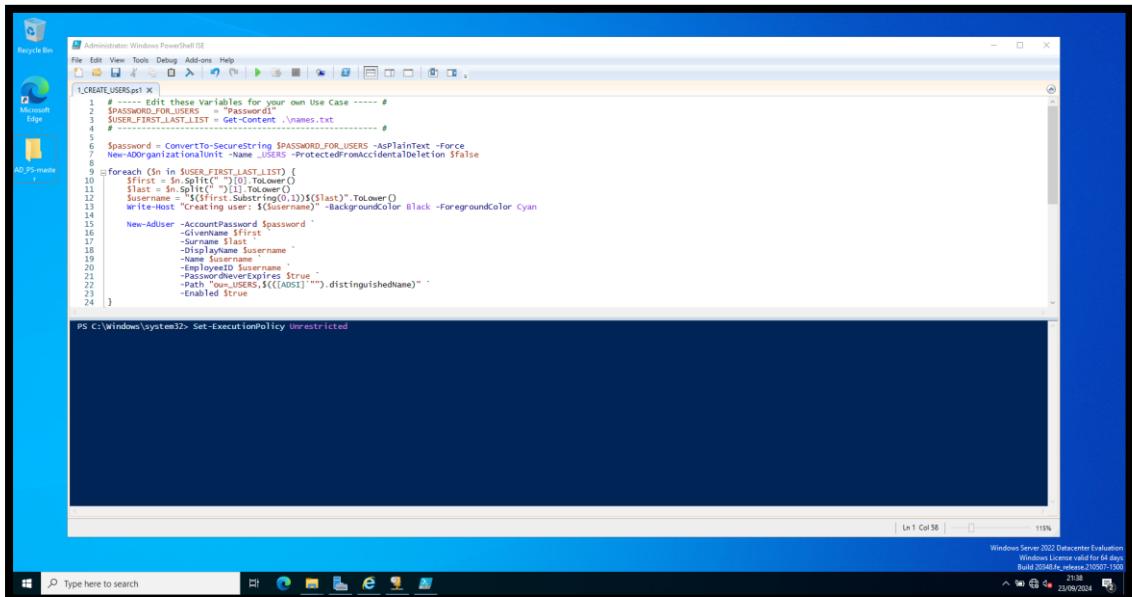
Go to configure local server turn off internet explorer enhanced security configuration. You won't normally do this in a production environment but this is just for this lab, it's just to prevent spam warnings asking are you sure you want to load this page.



PowerShell script for multiple users.



I going to run a PowerShell script to create multiple users using PowerShell.



Before I run anything I have to enable the execution of all scripts on this server.

```

1.CREATE_USERS.ps1
# Edit these Variables For your own Use Case ----- #
2.$PASSWORD_FOR_USERS = "Password123"
3.$USER_FIRST_LAST_LIST = Get-Content .\names.txt
4.#
5.$password = ConvertTo-SecureString $PASSWORD_FOR_USERS -AsPlainText -Force
6.New-ADUser -Name ($User) -AccountPassword $password -Enabled $true -ProtectedFromAccidentalDeletion $false
7.New-ADUser -Name ($User) -AccountPassword $password -Enabled $true -ProtectedFromAccidentalDeletion $false
8.#
9.ForEach ($n in $USER_FIRST_LAST_LIST) {
10.    $fFirst = $n.Split(",")[0].ToLower()
11.    $fLast = $n.Split(",")[1].ToLower()
12.    $User = $fFirst + "-" + $fLast
13.    Write-Host "Creating user: $($username)" -BackgroundColor Black -ForegroundColor Cyan
14.    New-ADUser -AccountPassword $password `-
15.        -GivenName $fFirst `-
16.        -Surname $fLast `-
17.        -DisplayName $User `-
18.        -EmployeeID $username `-
19.        -Path "OU=Users,DC=(ADSTEST),distinguishedName" `-
20.        -Enabled $true
21.}
22.#
23.#
24.}

PS C:\Windows\system32> Set-ExecutionPolicy Unrestricted

```

Execution policy Change  
The execution policy helps protect you from scripts that you do not trust. Changing the execution policy might expose you to the security risks described in the about\_Execution\_Policies help topic at https://go.microsoft.com/fwlink/?LinkId=135170. Do you want to change the execution policy?

Yes Yes to All No No to All Suspend

Select yes to all.

```

1.CREATE_USERS.ps1
# Edit these Variables For your own Use Case ----- #
2.$PASSWORD_FOR_USERS = "Password123"
3.$USER_FIRST_LAST_LIST = Get-Content .\names.txt
4.#
5.$password = ConvertTo-SecureString $PASSWORD_FOR_USERS -AsPlainText -Force
6.New-ADUser -Name ($User) -AccountPassword $password -Enabled $true -ProtectedFromAccidentalDeletion $false
7.New-ADUser -Name ($User) -AccountPassword $password -Enabled $true -ProtectedFromAccidentalDeletion $false
8.#
9.ForEach ($n in $USER_FIRST_LAST_LIST) {
10.    $fFirst = $n.Split(",")[0].ToLower()
11.    $fLast = $n.Split(",")[1].ToLower()
12.    $User = $fFirst + "-" + $fLast
13.    Write-Host "Creating user: $($username)" -BackgroundColor Black -ForegroundColor Cyan
14.    New-ADUser -AccountPassword $password `-
15.        -GivenName $fFirst `-
16.        -Surname $fLast `-
17.        -DisplayName $User `-
18.        -EmployeeID $username `-
19.        -Path "OU=Users,DC=(ADSTEST),distinguishedName" `-
20.        -Enabled $true
21.}
22.#
23.#
24.}

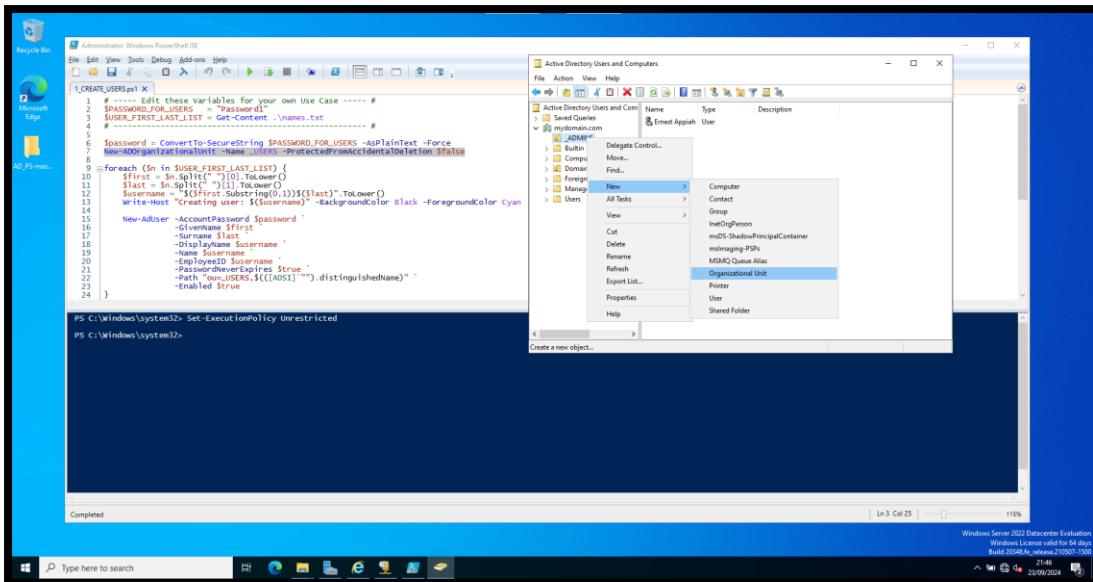
PS C:\Windows\system32> Set-ExecutionPolicy Unrestricted
PS C:\Windows\system32>

```

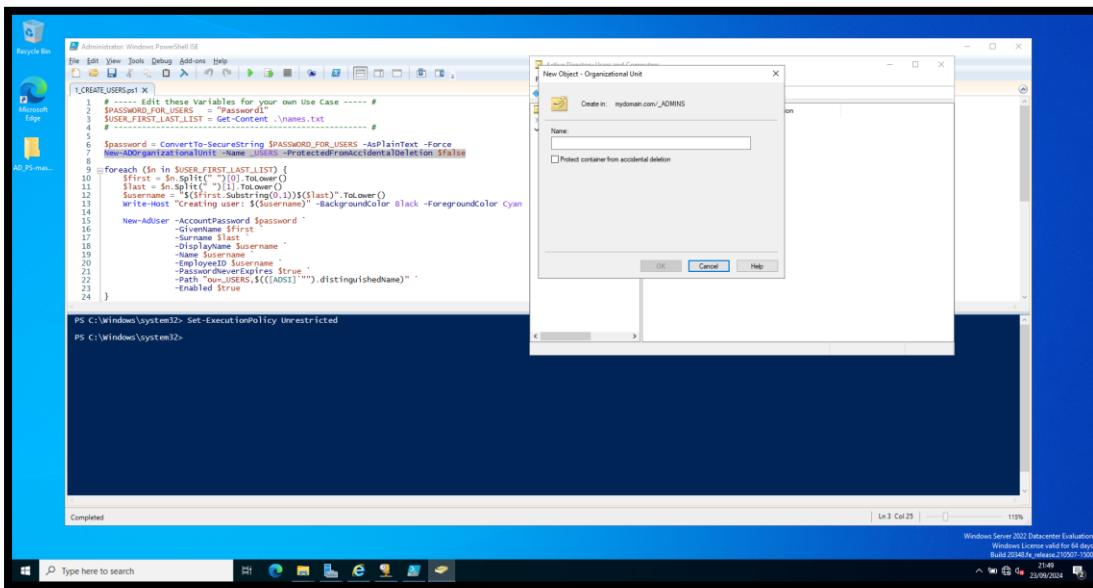
names - Notepad

First Name	Last Name
Jeanne	Leib
Mika	Benes
Lila	Borchardt
Douglas	Burke
Ivette	Burdo
Rozella	Loveless
Brad	Gane
John	Rees
Kimiko	Aultman
Ozella	Aray
Lockie	Barlin
Becky	Tone
Shirly	Batchelor
Zachariah	Barman
Amelia	Getzel
Arie	Sclafani
Eliza	Collins
Lee	Seidel
Lorina	Belair
Arlette	McGreavy
Julian	Wade
Celine	Dunstan
Isaura	Luplen
Elspeth	Sterling
In	Selridge
Margaret	Gartin
Cynthia	Shaddix
Constance	Wade
Myman	Skipper
Iceland	Paz
Malissa	Basuki
Karl	Chico
Domingo	Blondo
Ursula	Rinaldi
Christa	Thorne

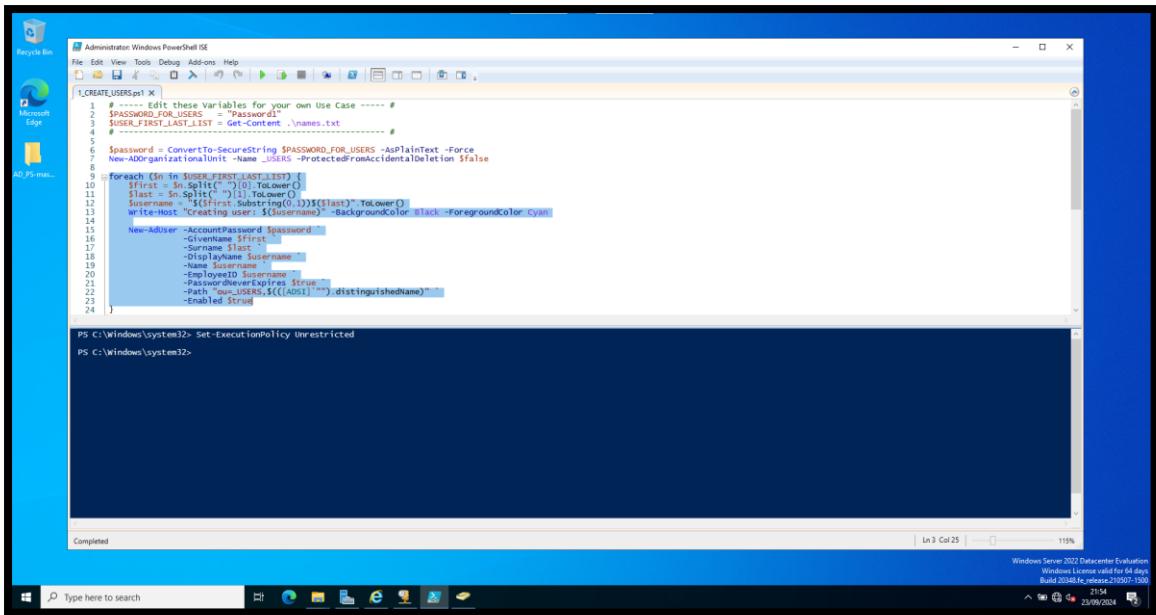
The get-content is linked to the names.txt file which holds all the names of users.



The new-ADOrganizationalUnit -Name line creates these users.

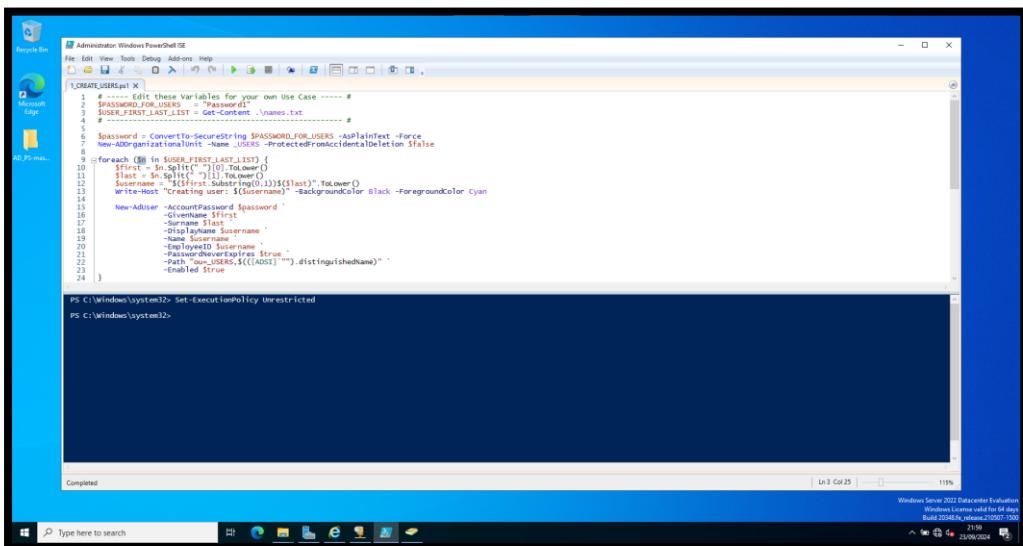


The ProtectedFromAccidentalDeletion \$false part just unticks the box in the new object-Organizational Unit.



```
1.CREATE_USERS.ps1 X
Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
C:\Windows\system32> .\CREATE_USERS.ps1
1 #----- Edit these Variables for your own Use Case ----- #
2 $PASSWORD_FOR_USERS = "Password123"
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     $nFirst = $n.Split(" ")[0].ToLower()
11     $nLast = $n.Split(" ")[1].ToLower()
12     $username = "$($nFirst.Substring(0,1))$($nLast)".ToLower()
13     Write-Host "Creating user: $($username)" -BackgroundColor Black -ForegroundColor Cyan
14     New-AdUser -AccountPassword $password
15         -GivenName $nFirst
16         -Surname $nLast
17         -DisplayUsername
18         -Name $username
19         -EmployeeID $username
20         -EmployeeNumber $username
21         -FaxNeverExpires $true
22         -Path "ou=USERS,$((ADSI)::(''))".distinguishedname"
23         -Enabled $true
24 }
PS C:\Windows\system32> Set-ExecutionPolicy Unrestricted
PS C:\Windows\system32>
Completed
```

The for each block of code, loops through the first to last name of each user and creates the individual accounts. It loops for each and every user to create their accounts.



```
1.CREATE_USERS.ps1 X
Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
C:\Windows\system32> .\CREATE_USERS.ps1
1 #----- Edit these Variables for your own Use Case ----- #
2 $PASSWORD_FOR_USERS = "Password123"
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     $nFirst = $n.Split(" ")[0].ToLower()
11     $nLast = $n.Split(" ")[1].ToLower()
12     $username = "$($nFirst.Substring(0,1))$($nLast)".ToLower()
13     Write-Host "Creating user: $($username)" -BackgroundColor Black -ForegroundColor Cyan
14     New-AdUser -AccountPassword $password
15         -GivenName $nFirst
16         -Surname $nLast
17         -DisplayUsername
18         -Name $username
19         -EmployeeID $username
20         -EmployeeNumber $username
21         -FaxNeverExpires $true
22         -Path "ou=USERS,$((ADSI)::('')).distinguishedname"
23         -Enabled $true
24 }
PS C:\Windows\system32> Set-ExecutionPolicy Unrestricted
PS C:\Windows\system32>
```

This \$n near the for each is representation of that current user that's being examined.

The screenshot shows a Windows Server 2022 PowerShell window with the title 'Administrator: Windows PowerShell [ISE]'. The code in the window is as follows:

```

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 $password = ConvertTo-SecureString $PASSWORD_FOR_USERS -AsPlainText -Force
6 New-ADOrganizationalUnit -Name _USERS -ProtectedFromAccidentalDeletion $false
7
8 foreach ($n in $USER_FIRST_LAST_LIST) {
9     $First = $n.Split(",")[0].ToLower()
10    $Last = $n.Split(",")[-1].ToLower()
11    $Username = "$($First.Substring(0,1))$($Last)".ToLower()
12    $write-host "Creating user: $Username" -BackgroundColor Black -ForegroundColor Cyan
13
14    New-AddUser -AccountPassword $password `
15        -GivenName $First `
16        -Surname $Last `
17        -DisplayName $Username `
18        -Name $Username `
19        -EmployeeID $username `
20        -PasswordNeverExpires $true `
21        -Path "ou=_USERS,$((ADSI)::'').distinguishedName" `
22        -Enabled $true
23 }

```

Below the code, the command `Set-ExecutionPolicy Unrestricted` is run. To the right of the PowerShell window, a Notepad window titled 'names - Notepad' displays a list of names:

- Ernest Apiah
- Sally Bass
- Drama Earls
- Suzanne Hell
- Janelle Gridb
- Halle Joss
- Lila Borchardt
- Dwayne Plumb
- Ivonne Vedo
- Roxella Loveless
- Brady Bane
- Mark Koss
- Klaus Aultman
- Ozella Mcray
- Jocelle Beverlin
- Bobby Goss
- Shirley Batchelor
- Zachariah Varian
- Andria Geisert
- Levi Fiedel
- Elise Rouleau
- Levi Sunggal
- Ismael Belair
- Alelette Moreeuv

The PowerShell window shows the command completed at the bottom.

The \$first line will split and take the first name and leave space. The \$last part will split and take the last name and leave space. The \$username line will take the first letter of the first name and concatenate it to the last name and put it in lowercase. The Write-Host will alert us that a user has been created and show this in the colour of cyan.

The screenshot shows a Windows Server 2022 PowerShell window with the title 'Administrator: Windows PowerShell [ISE]'. The code is identical to the previous one:

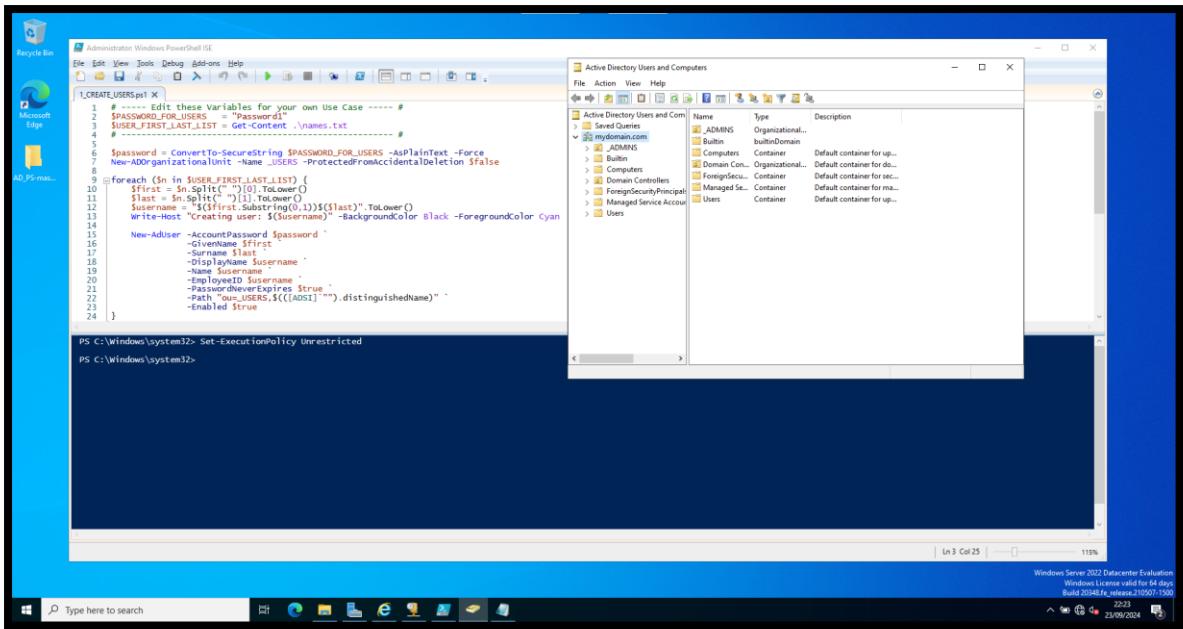
```

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 $password = ConvertTo-SecureString $PASSWORD_FOR_USERS -AsPlainText -Force
6 New-ADOrganizationalUnit -Name _USERS -ProtectedFromAccidentalDeletion $false
7
8 foreach ($n in $USER_FIRST_LAST_LIST) {
9     $First = $n.Split(",")[0].ToLower()
10    $Last = $n.Split(",")[-1].ToLower()
11    $Username = "$($First.Substring(0,1))$($Last)".ToLower()
12    $write-host "Creating user: $Username" -BackgroundColor Black -ForegroundColor Cyan
13
14    New-AddUser -AccountPassword $password `
15        -GivenName $First `
16        -Surname $Last `
17        -DisplayName $Username `
18        -Name $Username `
19        -EmployeeID $username `
20        -PasswordNeverExpires $true `
21        -Path "ou=_USERS,$((ADSI)::'').distinguishedName" `
22        -Enabled $true
23 }

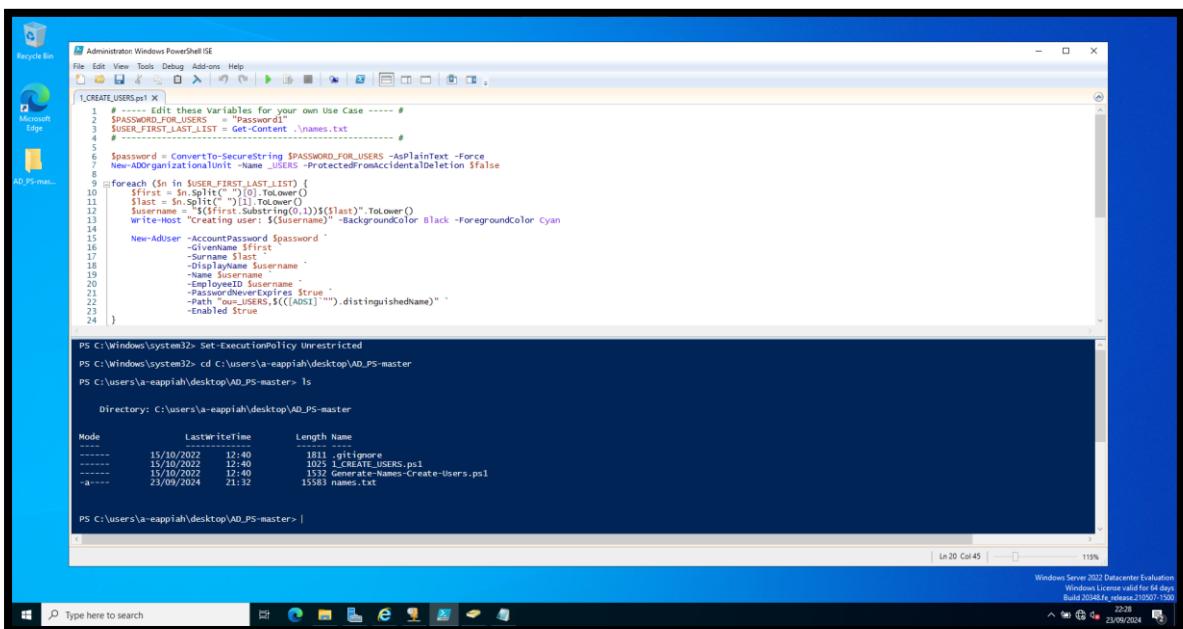
```

Below the code, the command `Set-ExecutionPolicy Unrestricted` is run. To the right of the PowerShell window, the 'Active Directory Users and Computers' snap-in is open, showing the 'New Object - User' dialog box. The 'Create in' dropdown is set to 'mydomain.com/\_ADMNS'. The 'First name' field is set to 'Initials' and the 'Last name' field is empty. The 'User logon name' dropdown is set to '@mydomain.com' and the 'User logon name (pre-Windows 2000)' dropdown is set to 'MYDOMAIN'. The 'Next >' button is visible at the bottom of the dialog.

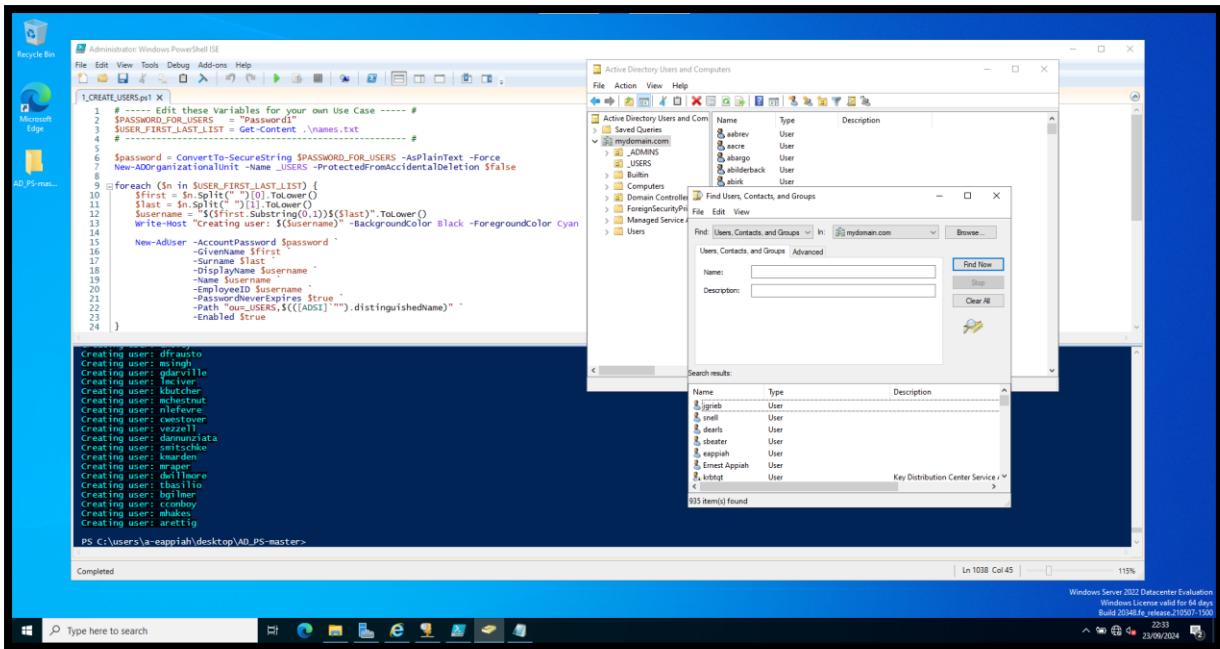
The New-Adduser block of code creates a new user in active directory. The \$password is assigning Password1. PasswordNeverExpires \$true that equivalent to checking that box when I create it with the GUI. The ou is going to get put in \_USERS it's created yet but that's what the: new-ADOrganizationalUnit -Name line does. Enabled \$true means the user account is going to be enabled/usable.



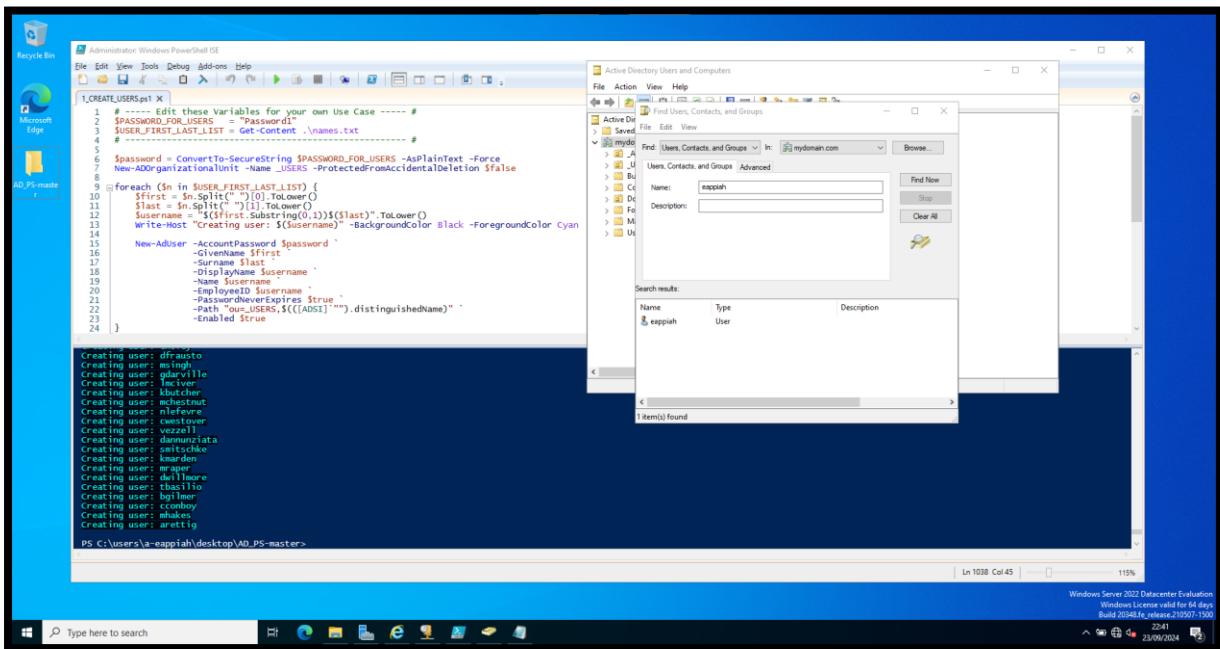
When I refresh mydomain.com I don't see the ou in here the user's ou.



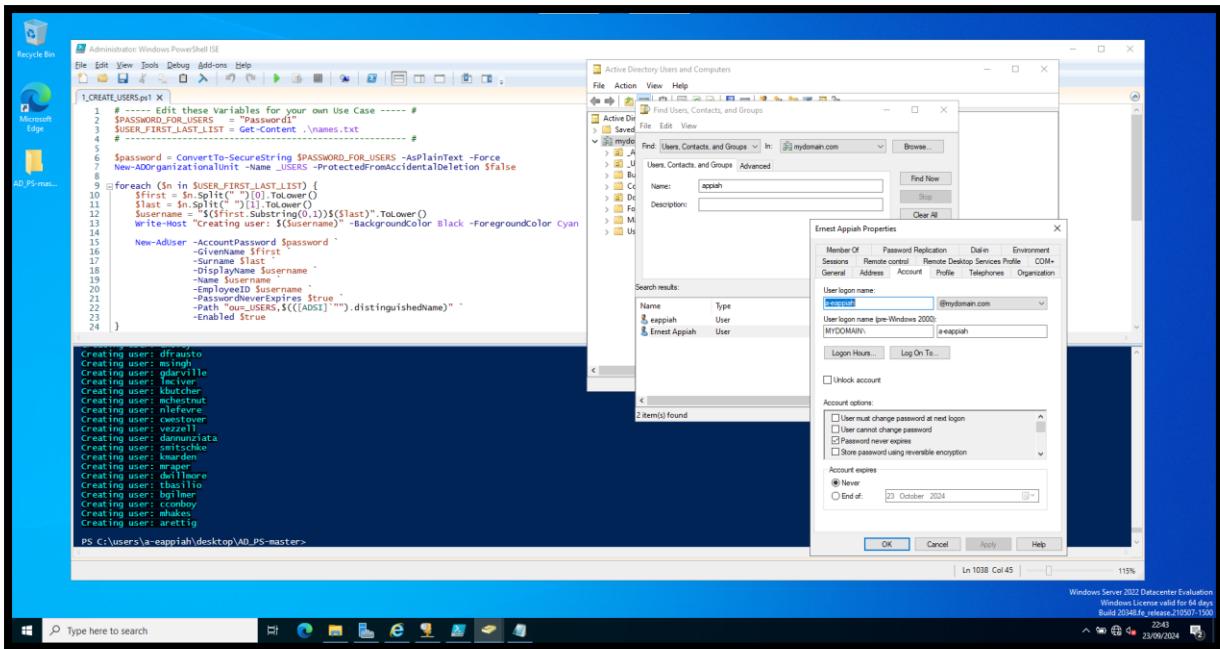
Change directory to the user account on windows server which is a-eappiah to access the folder holding the names text file. The ls shows the names.txt file is in there.



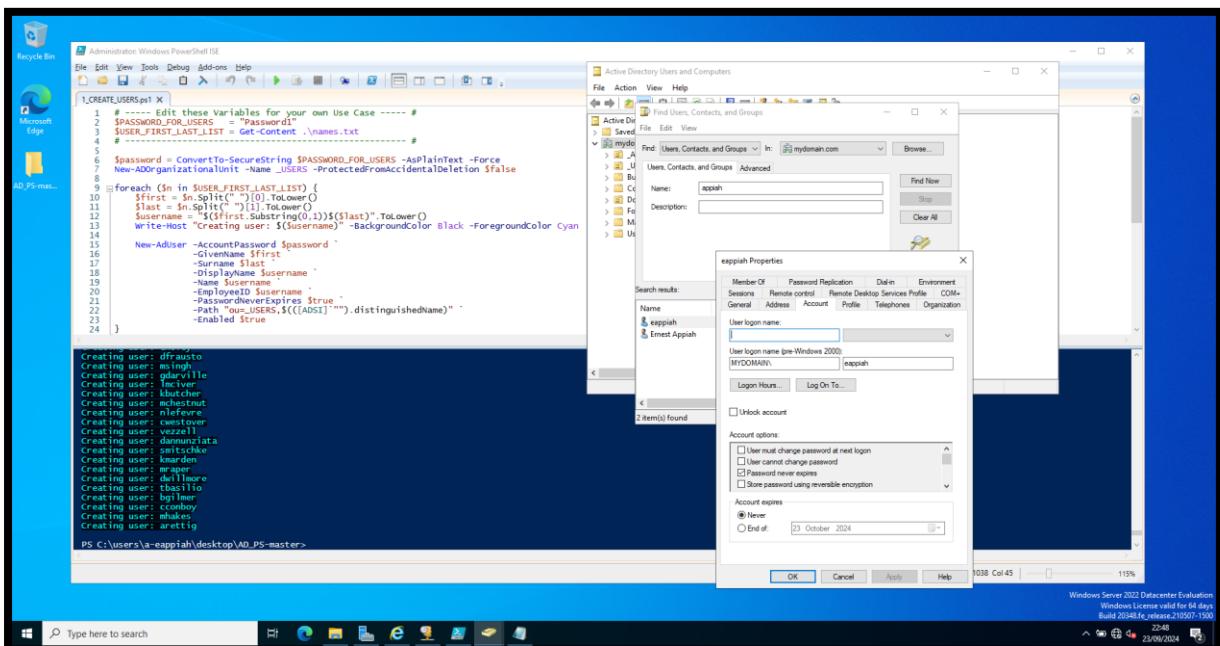
While that is creating users go to mydomain.com refresh you can now see \_USERS select and that users are being created. Also, if you right click domain and select find make sure the option next to find is on Users, Contacts and Groups, click on find now and you can see a list of the users like 935 users but there will be around 1000 when done. Now all the users created. In the code I can change the background colour and text colour when the names are being generated in the running of the script if I choose to.



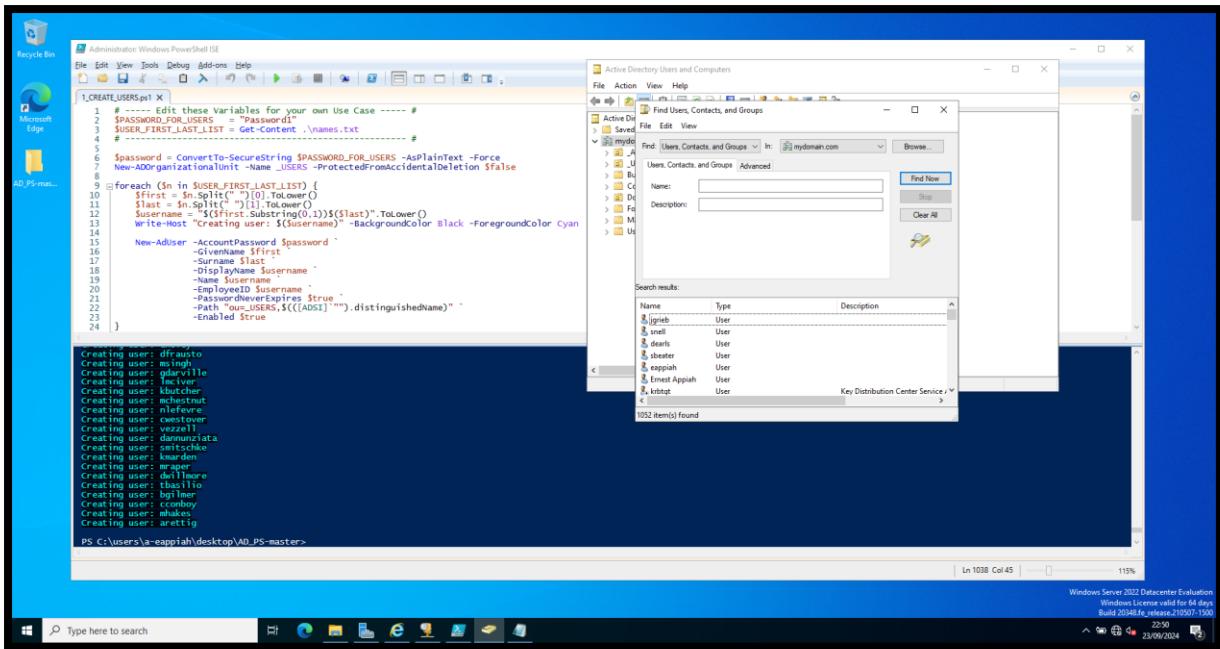
In the domain find and I will search for my name and it comes up.



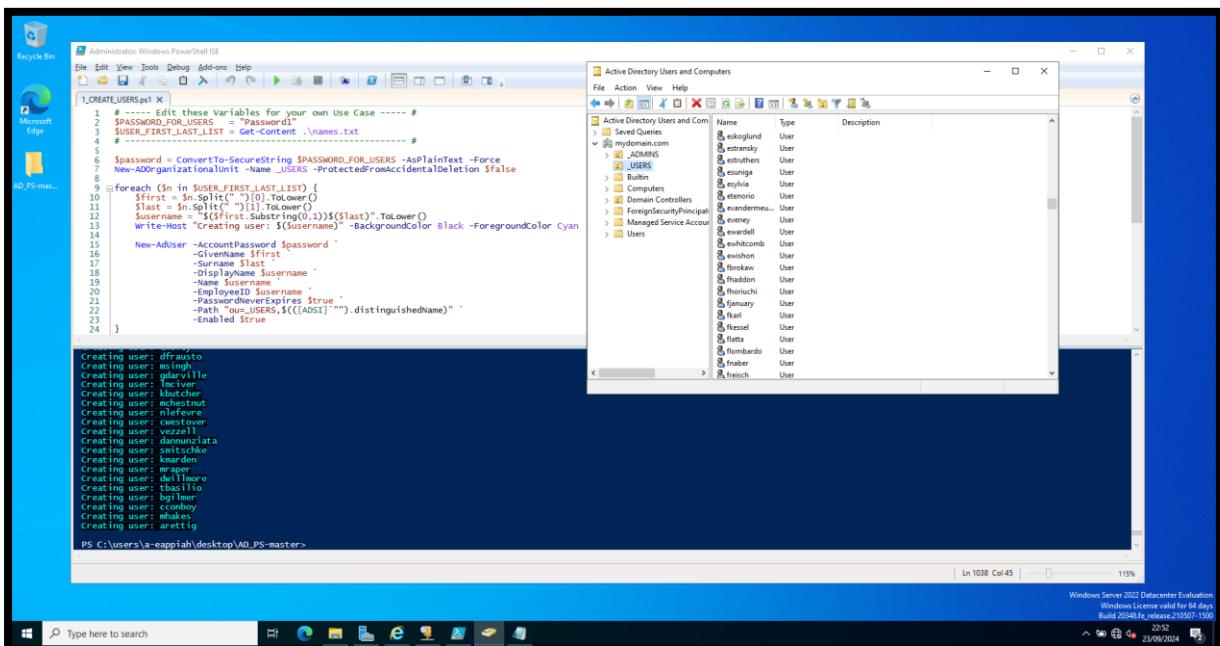
I put in just my last name and it found both of my accounts this one being my admin account.



This one my normal account.

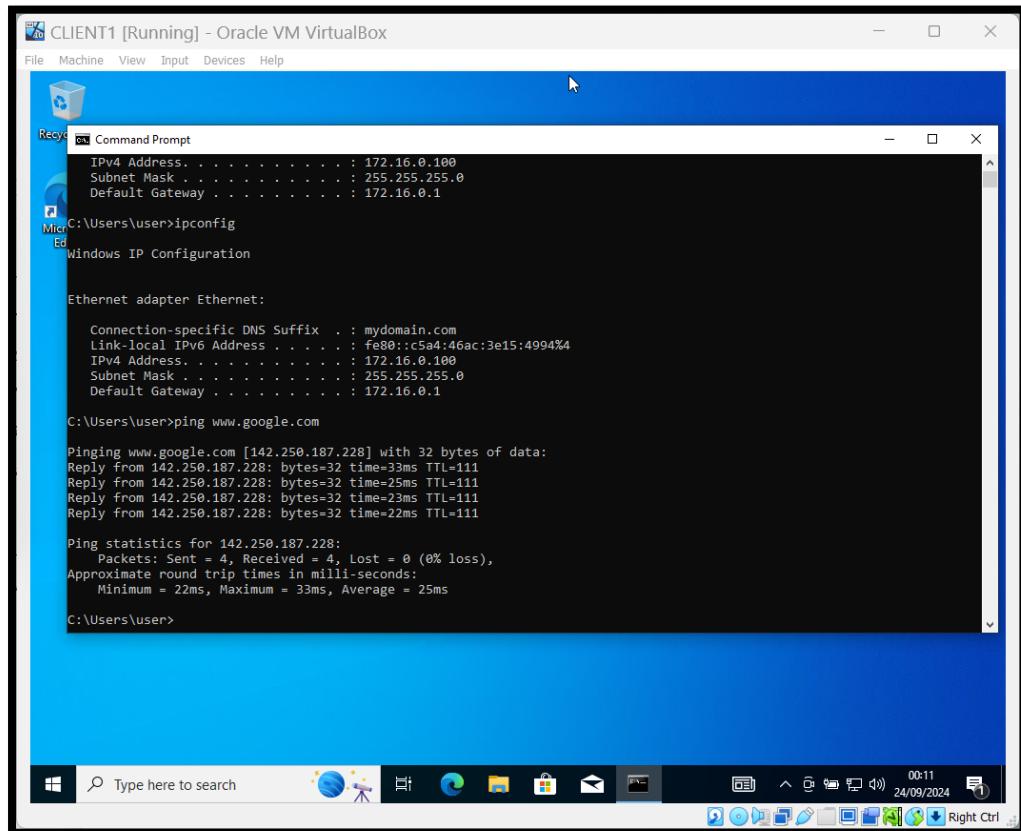


I searched for blank and it found over a 1000 accounts.

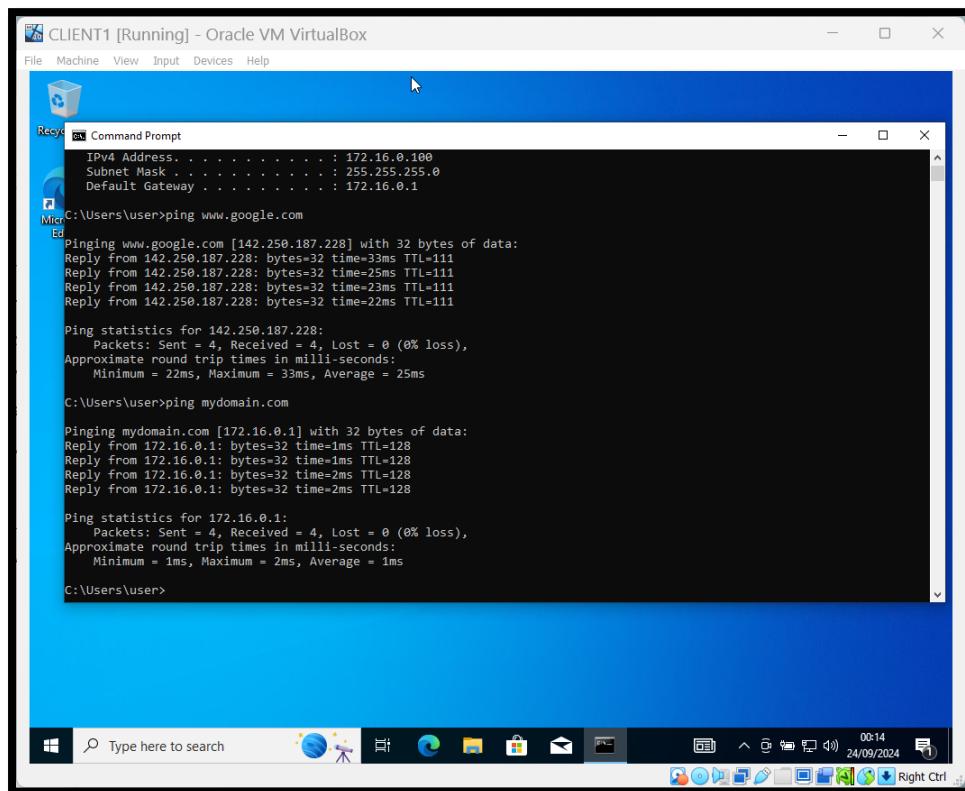


I can scroll through all the user accounts.

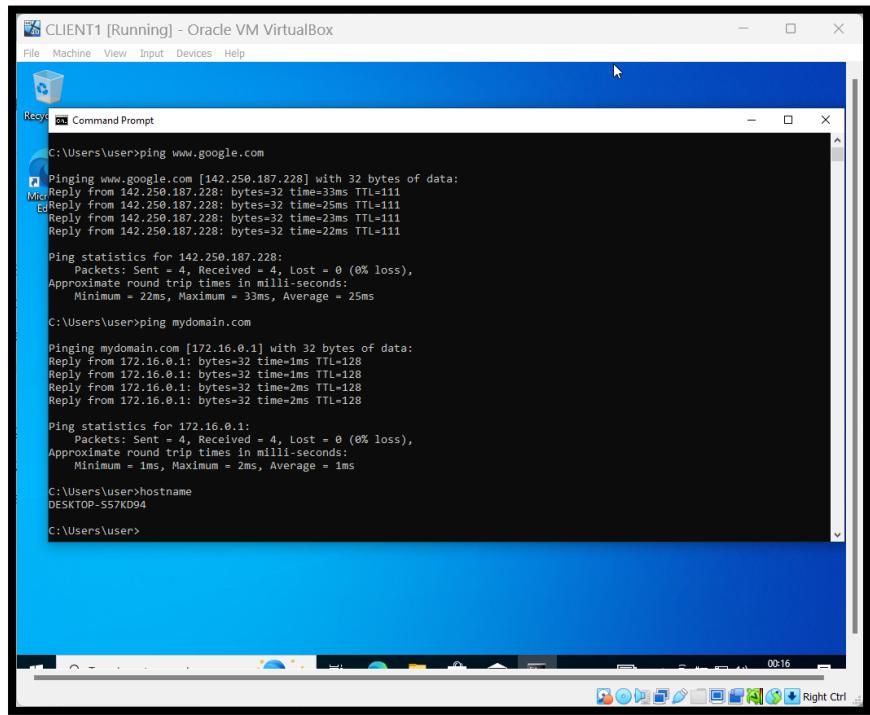
## Windows 10 machine



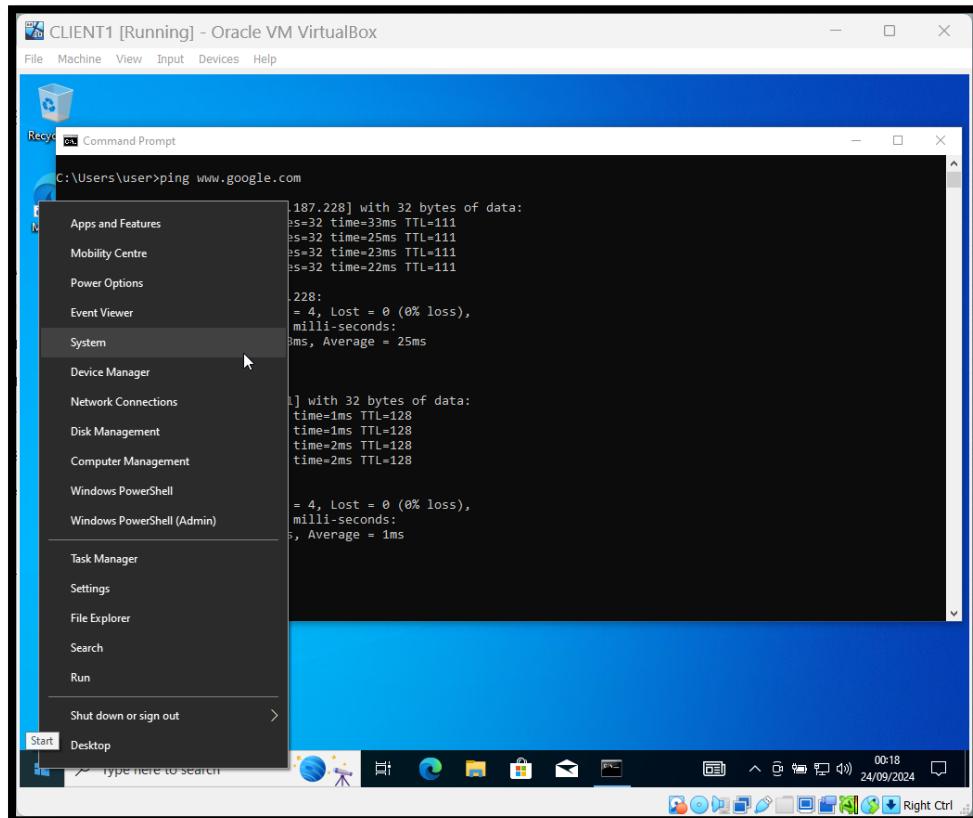
The internet connection is working on this Windows 10 machine client 1. I can ping to google.com.



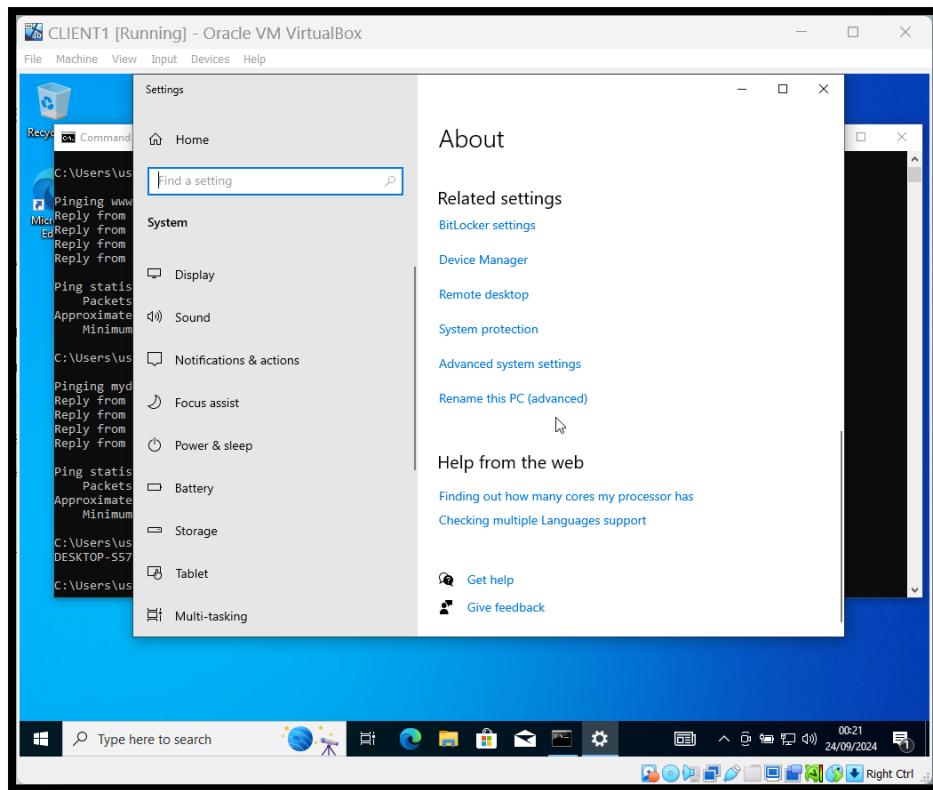
I can ping mydomain.com so my domain controller is responding to this client machine.



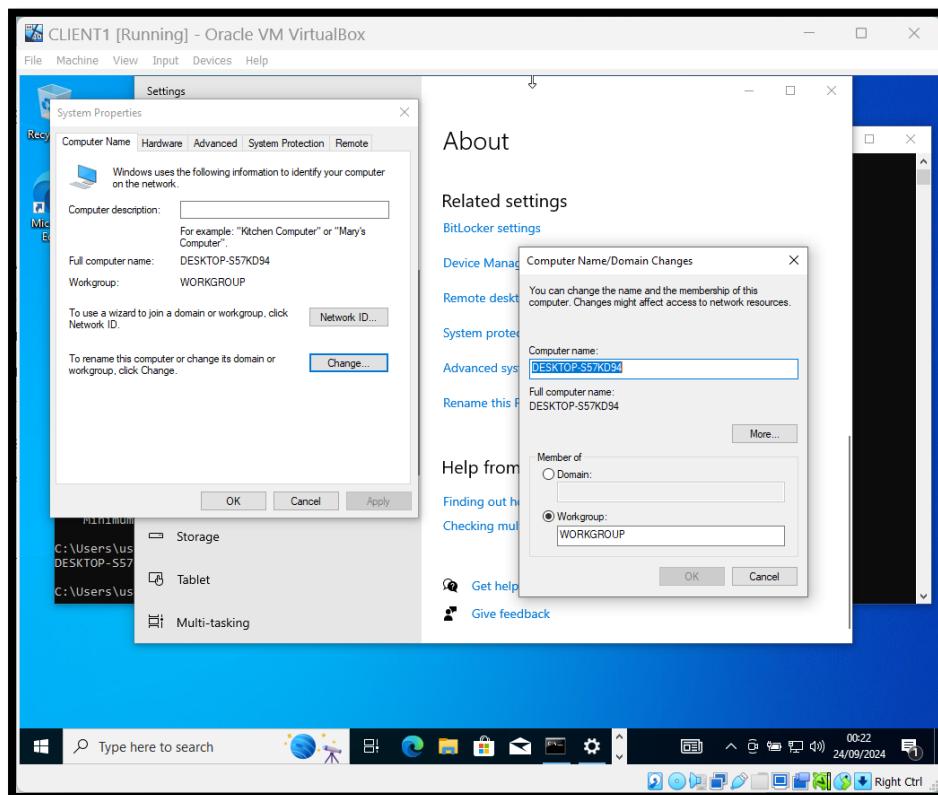
Lets change the hostname.



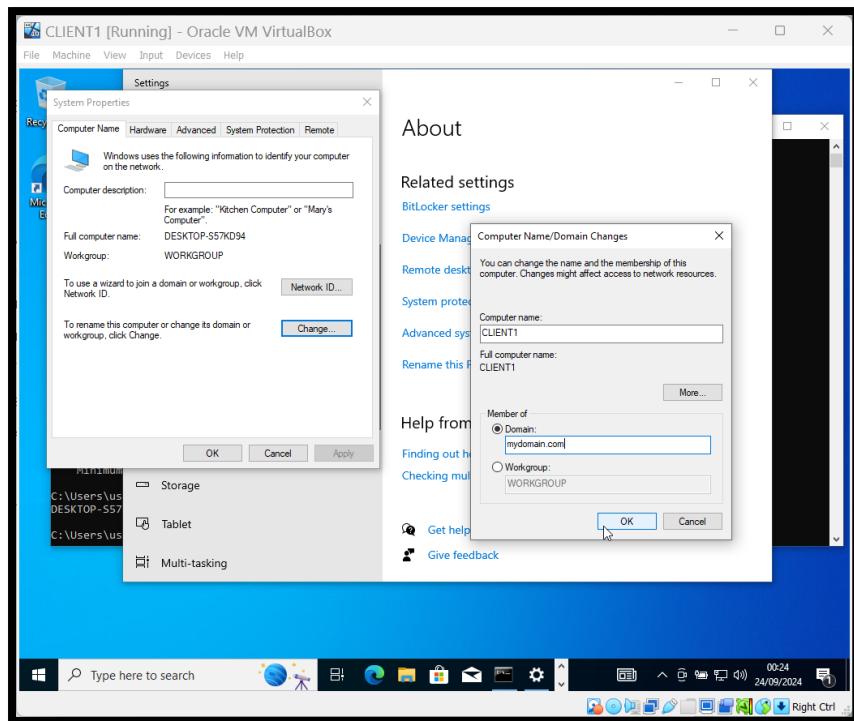
Right click go to system.



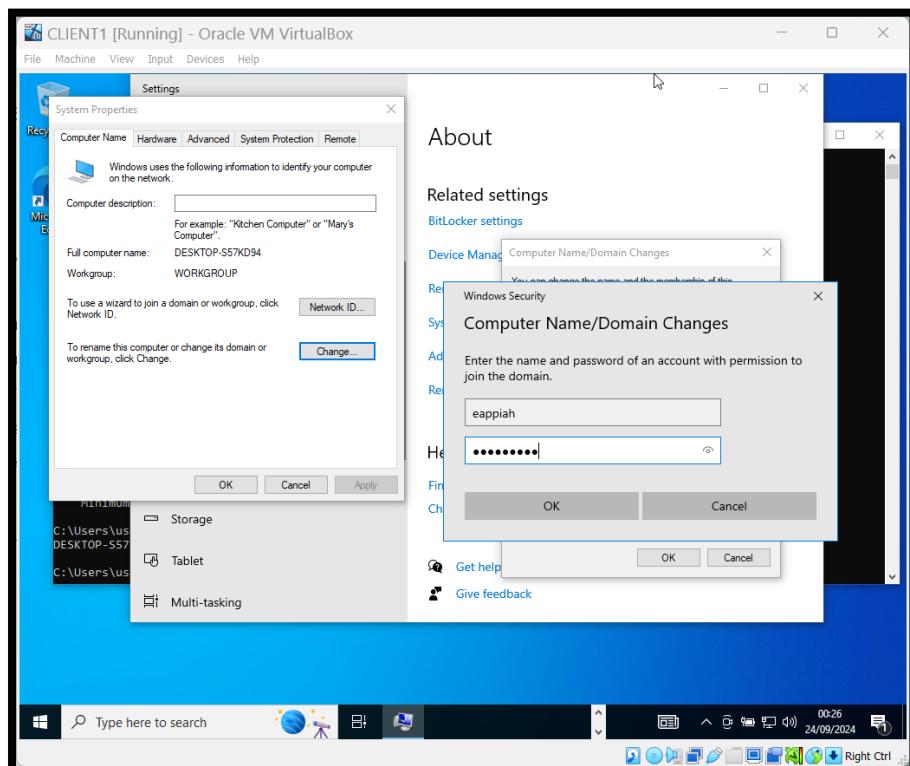
Go to this rename this PC (Advance). I can join the domain at the same time doing it this way.



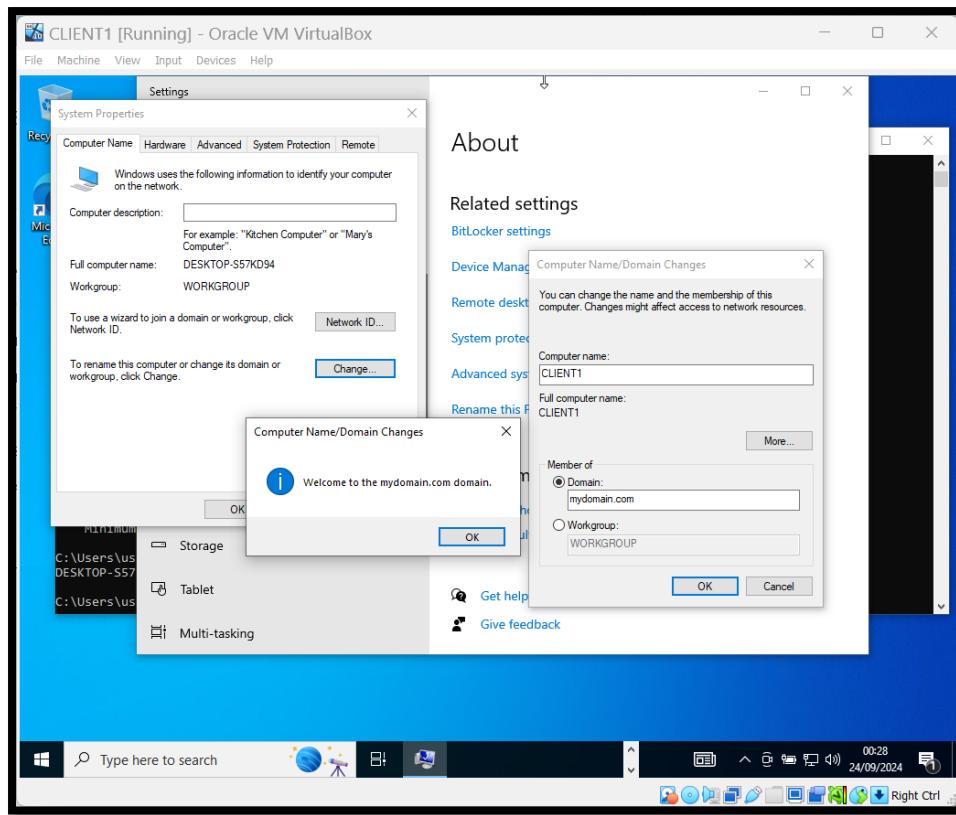
Select change.



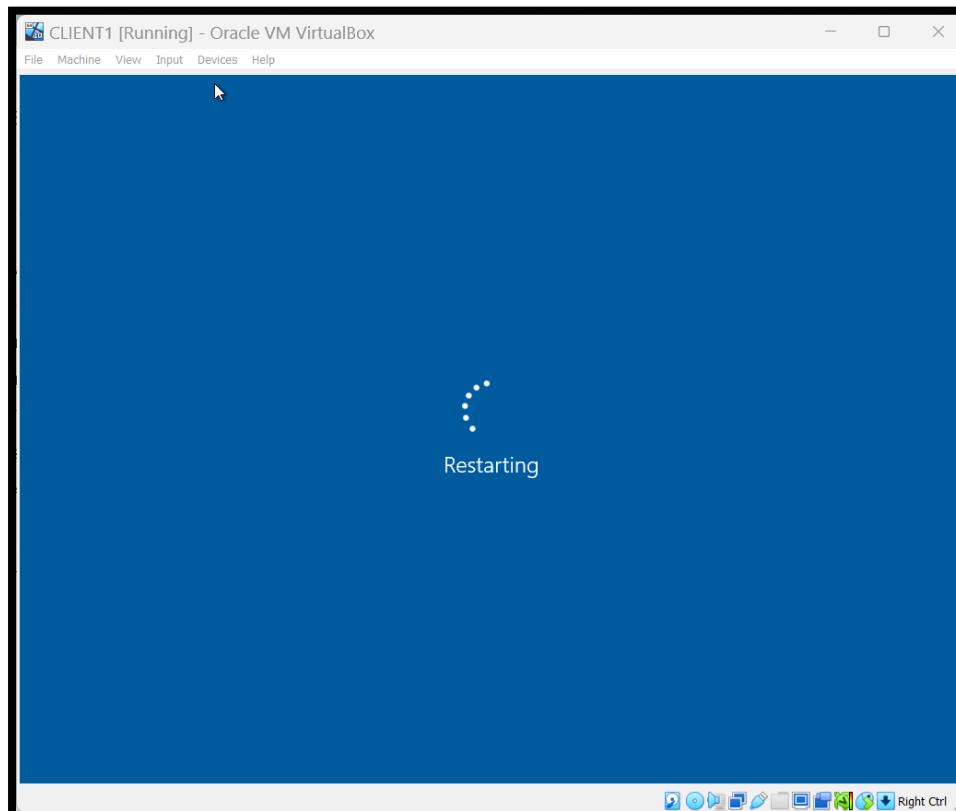
Use the name of the CLIENT1 of the Windows 10 machine and select mydomain.com the name of the domain.



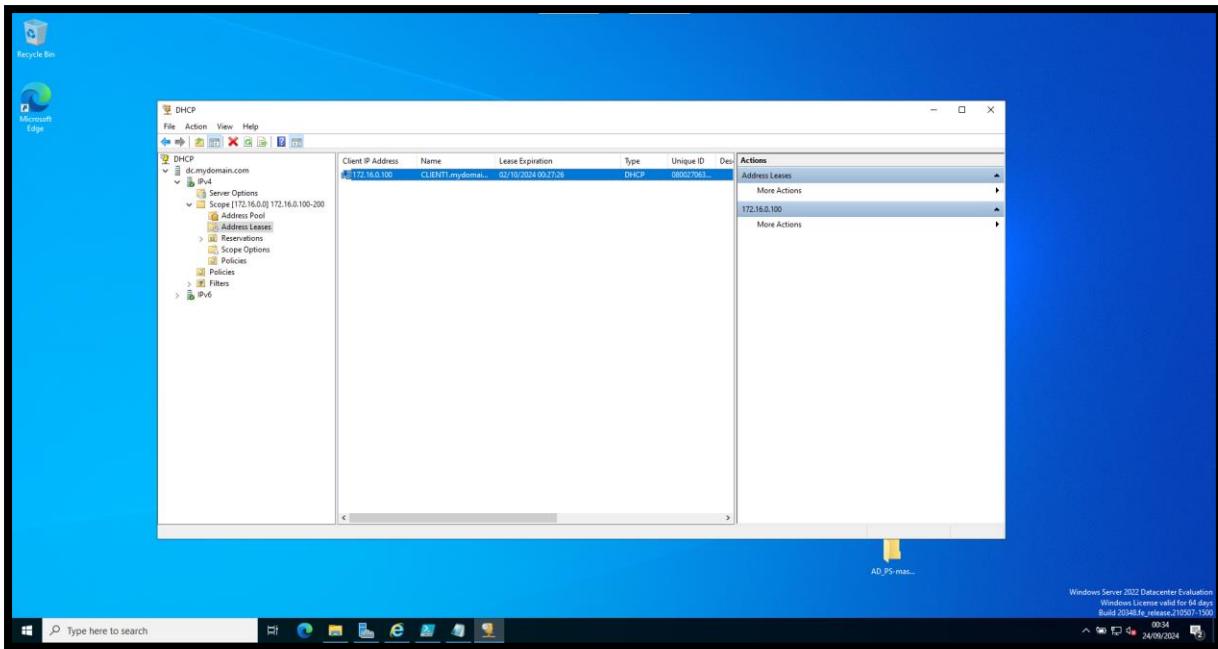
I will type in my normal user account details.



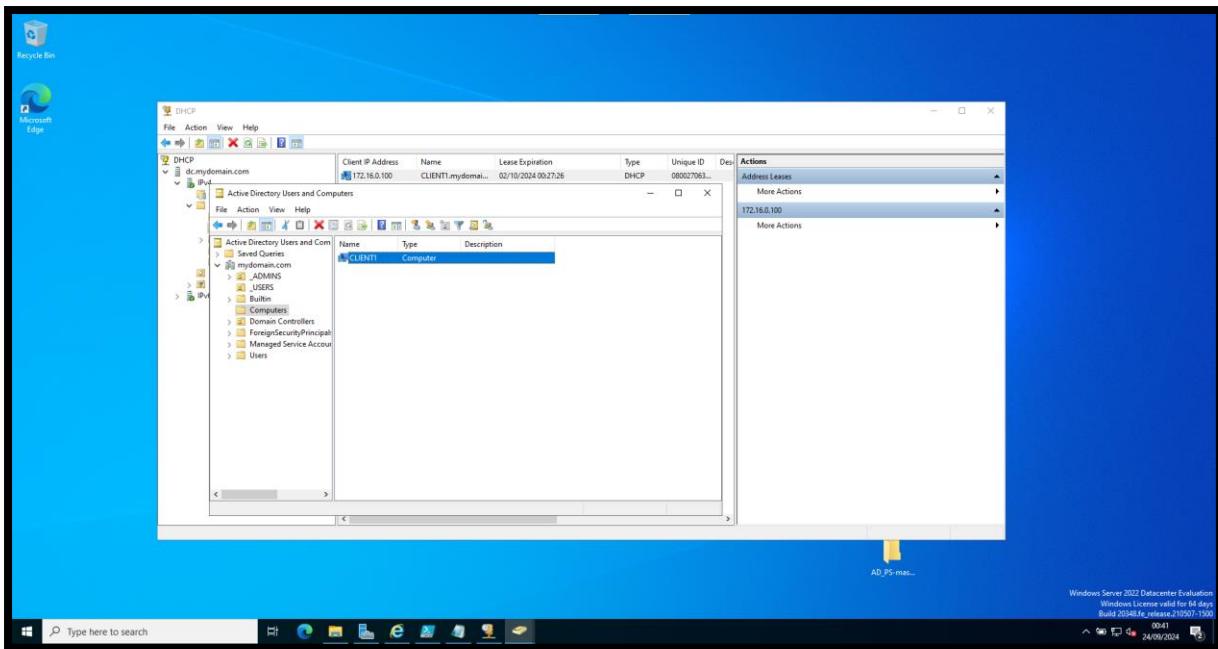
The normal user account worked and has connected to mydomain.com domain. Ok and restart machine.



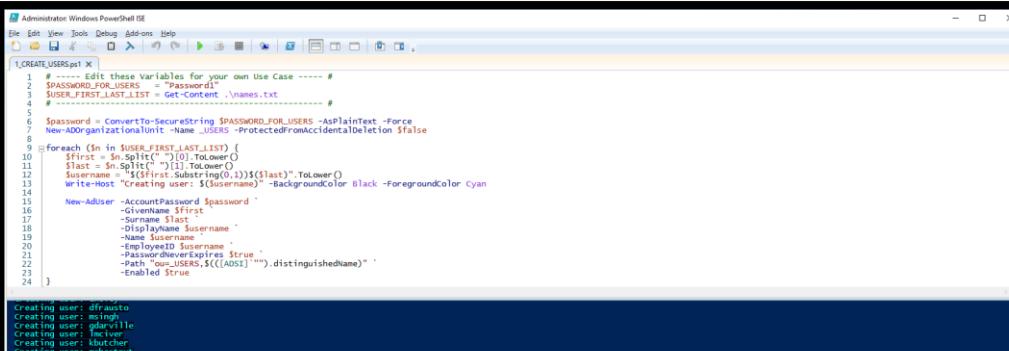
Machine restarting.



In Windows server 2022 domain controller. Go to DHCP, scope, Address Leases, we have one lease from my client computer. So when I created my client computer and joined it to the network, it reached out to the DCHP server automatically and requested an address and then the DCHP server gave it an address. Now we have this lease in here.



Now we can see after I joined this client computer to the domain this client computer automatically comes in here. This knows that this computer is a member of the domain.



The screenshot shows a Windows PowerShell ISE window with the title bar "CREATE\_USERS.ps1". The code is a PowerShell script for creating users in Active Directory. It includes variable declarations, a loop to read user names from a file, and a command to create each user with specific attributes like a password and a distinguished name.

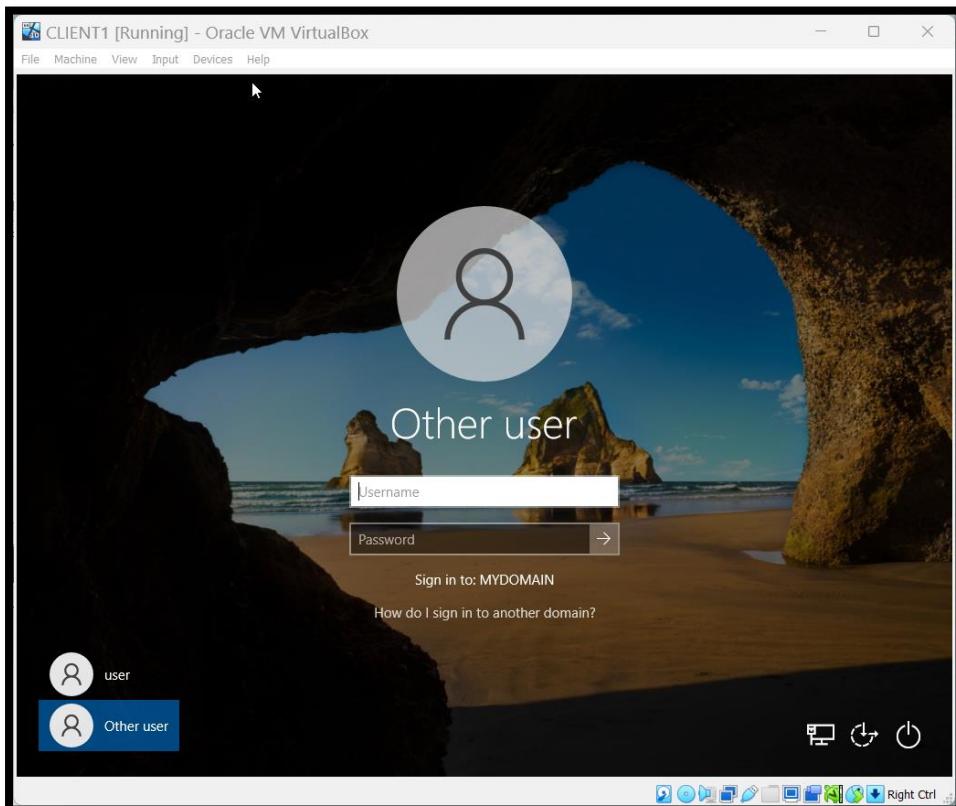
```
1 # ----- Edit these Variables for your own Use Case ----- #
2 $PASSWORD_FOR_USERS = "Password1"
3 $USER_FIRST_LAST_LIST = Get-Content .\names.txt
4 # ----- End of Variables ----- #
5
6 $password = ConvertTo-SecureString $PASSWORD_FOR_USERS -AsPlainText -Force
7 New-ADOrganizationalUnit -Name "USERS" -ProtectedFromAccidentalDeletion $false
8
9 foreach ($n in $USER_FIRST_LAST_LIST) {
10     $first = $n.Split(" ")[0].ToLower()
11     $last = $n.Split(" ")[1].ToLower()
12     $username = "$($first).$($last)"
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             -EmployeeNumber $username `-
21             -PasswordNeverExpires $true `-
22             -Path "OU=USERS,$((Get-ADObject $n).distinguishedName)" `-
23             -Enabled $true
24 }

```

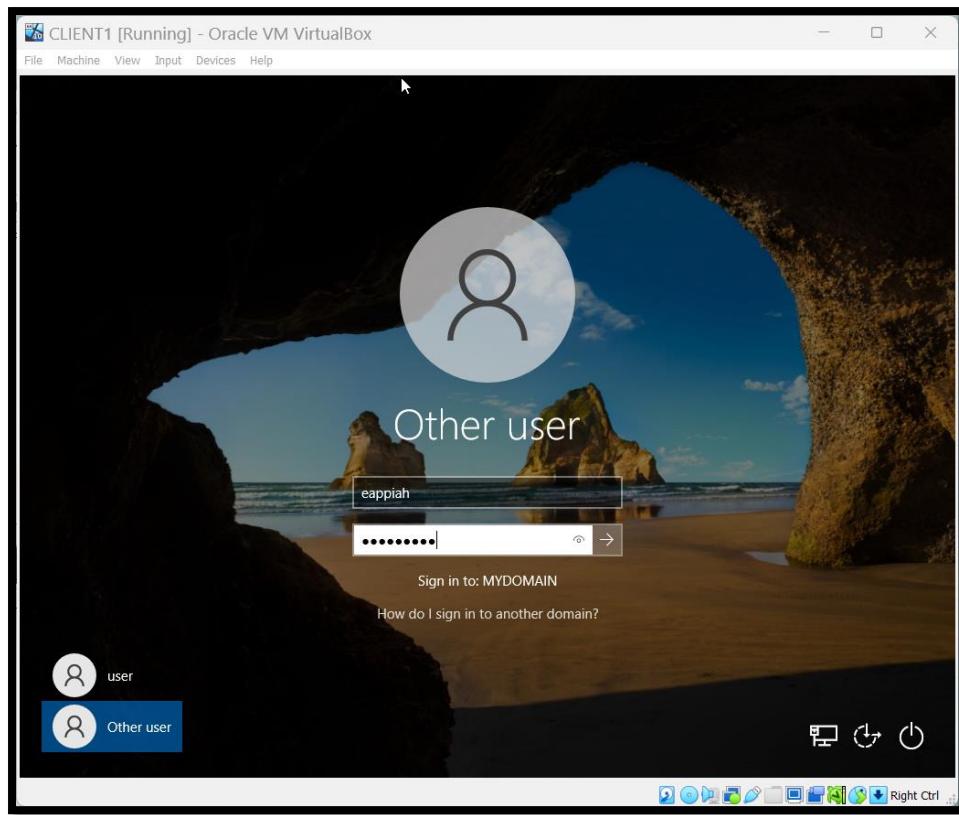
Output of the script:

```
Creating user: afrikando
Creating user: amitng
Creating user: galiville
Creating user: jayshree
Creating user: kuttcher
Creating user: achestnut
Creating user: amitng
Creating user: cwestover
Creating user: vezzell
Creating user: amitng
Creating user: smitschke
Creating user: kaurden
Creating user: amitng
Creating user: dwiltheore
Creating user: amitng
Creating user: boller
Creating user: conboy
Creating user: amitng
Creating user: arttig
```

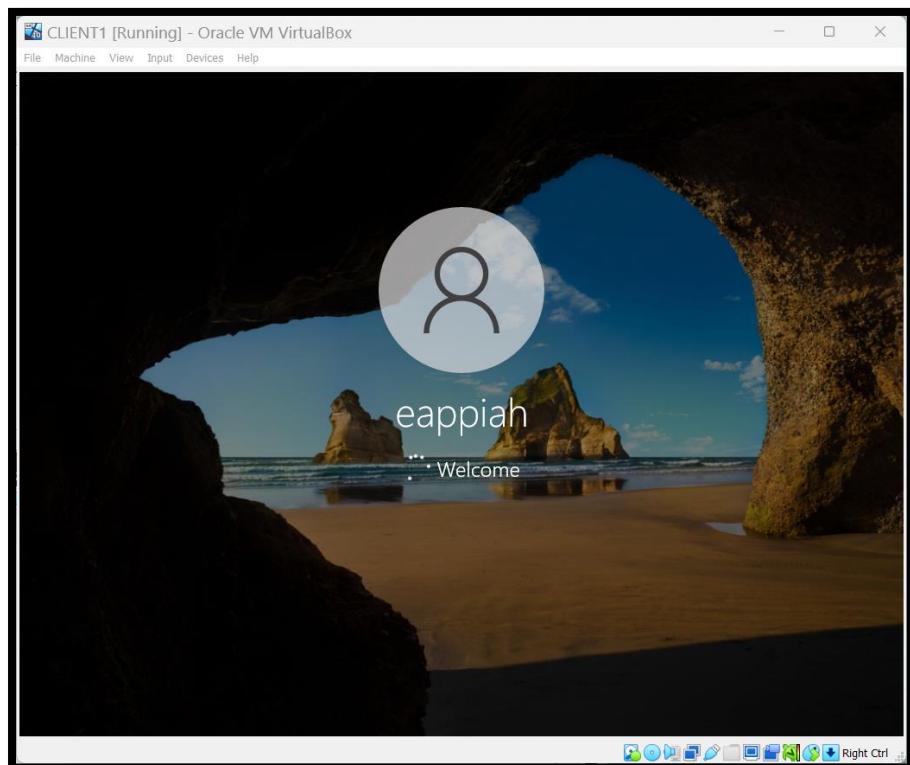
I can use any of these accounts to log into that client1 computer because the computer is joined to the domain.



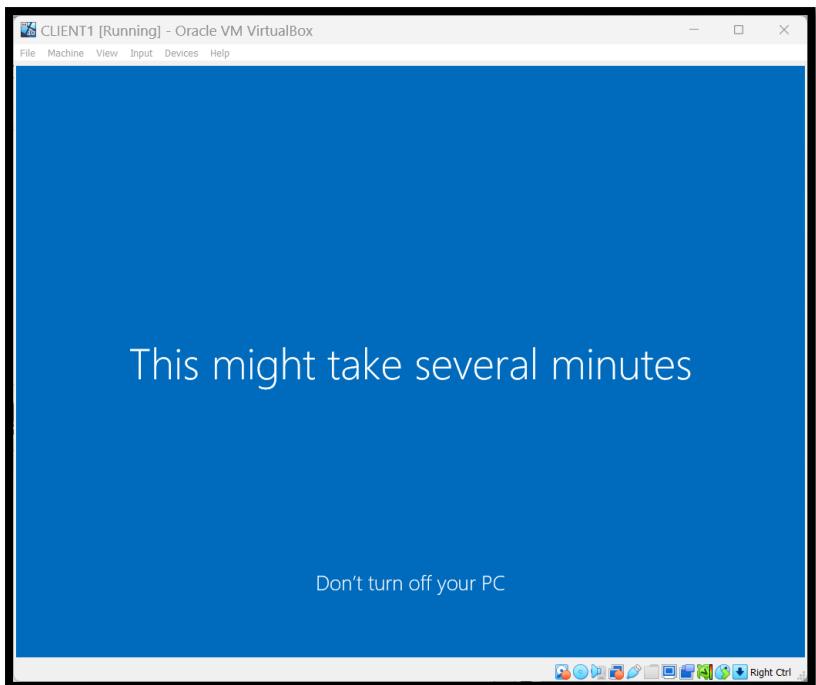
Sign in using other user to sign into mydomain.com.



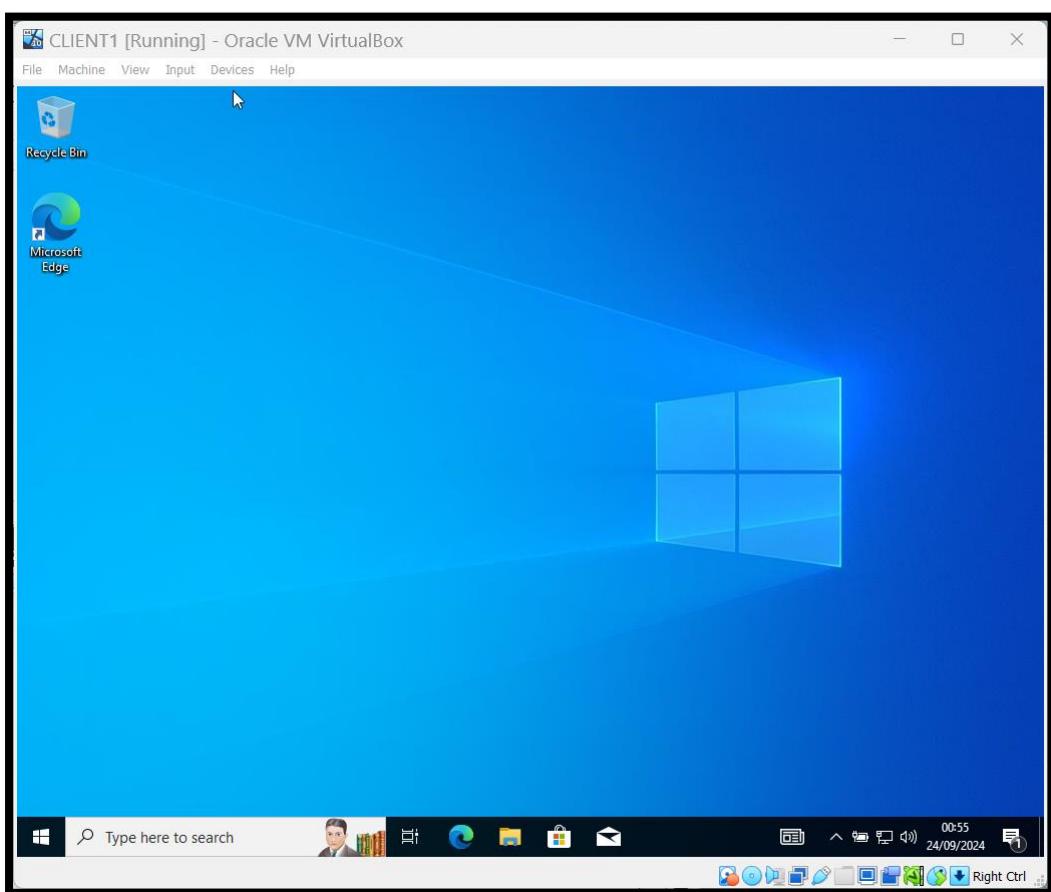
I will use my user account.



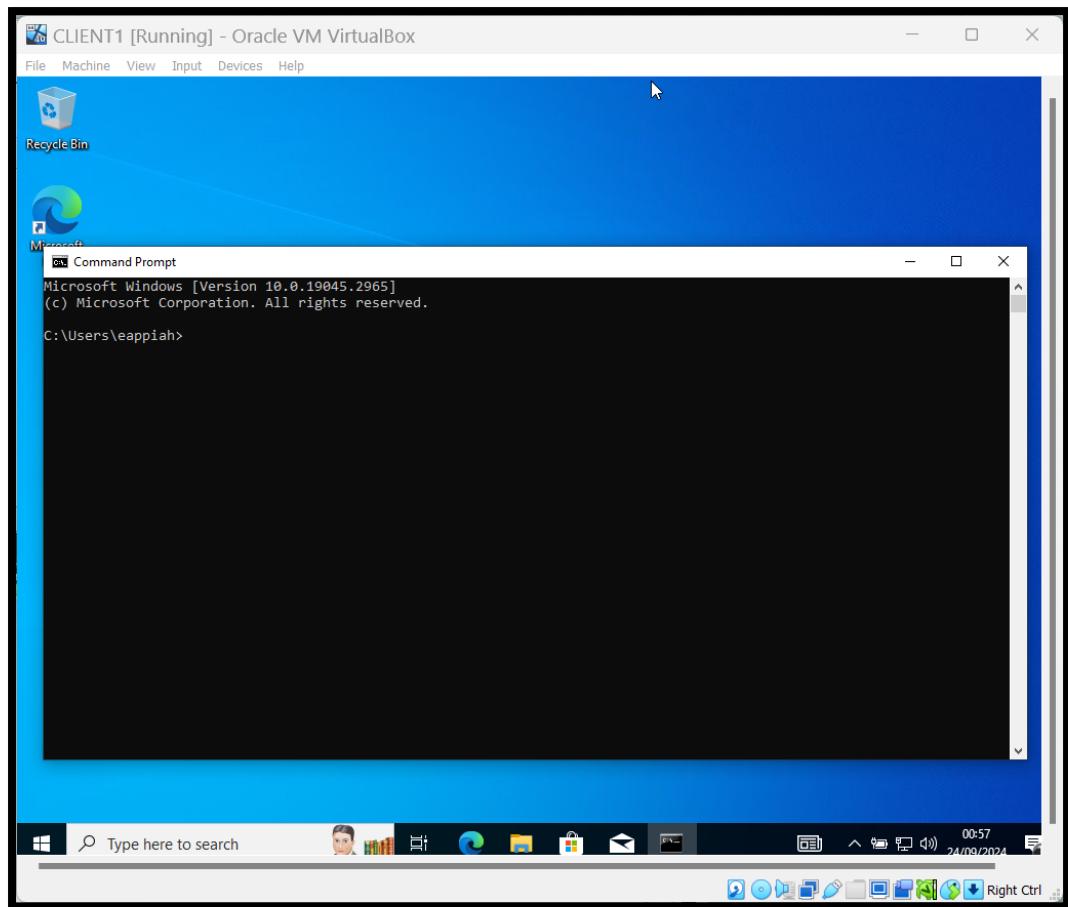
Signing in.



It's working it taking a few minutes because it's the first time log in to this user on this machine so it's setting it up.



I logged in.



In the command line you can see the user is my name Ernest Appiah.