

## Lab 4 – Managing User Accounts

---

### Task 1: Install and Configure SRV07 “File Server”

---

- Prep Step: Creating the Virtual Machine

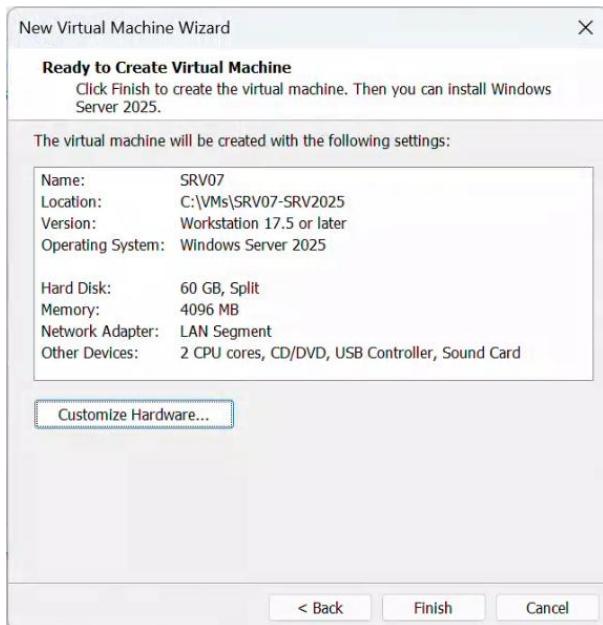
Explanation:

---

I created a new virtual machine **in** VMware and named **it** SRV07. I used Windows Server 2025 Standard **(Desktop Experience)** as the operating system. I **set it to 2 CPUs, 4 GB of RAM, a 60 GB hard disk,** and connected **it** to the LAN Segment: LAN1.

Screenshot:

---



- Step 1: Renaming the Server

Command:

---

```
Rename-Computer -NewName "SRV07" -Restart
```

```
PS C:\Users\Administrator> Rename-Computer -NewName "SRV07" -Restart
```

Explanation:

---

After installation, I renamed the server to SRV07 to follow the naming convention **for** our lab environment. The server rebooted right after.

Screenshot:

---

Computer name	SRV07
Workgroup	WORKGROUP

## ■ Step 2: Setting the **Static** IP Address

Command:

```
-----  
New-NetIPAddress  
-InterfaceAlias "Ethernet0"  
-IPAddress "192.168.7.10"  
-PrefixLength 24  
-DefaultGateway "192.168.7.1"
```

Explanation:

I used the `New-NetIPAddress` command to manually assign a **static** IP to SRV07.

- `InterfaceAlias "Ethernet0"` → this is the name of the network adapter. I checked it **using** `Get-NetAdapter`.
- `IPAddress "192.168.7.10"` → this is the address I assigned to SRV07.
- `PrefixLength 24` → this means the subnet mask is `255.255.255.0`.
- `DefaultGateway "192.168.7.1"` → this tells SRV07 how to reach other networks (usually the router or domain controller).

This sets the network settings manually instead of **using** DHCP.

Screenshot:

```
PS C:\Users\Administrator> Get-NetAdapter  
  
Name           InterfaceDescription  
----  
Ethernet0      Intel(R) 82574L Gigabit  
  
PS C:\Users\Administrator> New-NetIPAddress -InterfaceAlias "Ethernet0" -IPAddress "192.168.7.10" -PrefixLength 24 -DefaultGateway "192.168.7.1"  
PS C:\Users\Administrator> ipconfig /all  
  
Windows IP Configuration  
  
  Host Name . . . . . : SRV07  
  Primary Dns Suffix . . . . . :  
  Node Type . . . . . : Hybrid  
  IP Routing Enabled. . . . . : No  
  WINS Proxy Enabled. . . . . : No  
  
Ethernet adapter Ethernet0:  
  
  Connection-specific DNS Suffix . . . . . :  
  Description . . . . . : Intel(R) 82574L Gigabit Network Connection  
  Physical Address. . . . . : 00-0C-29-A4-72-8A  
  DHCP Enabled. . . . . : No  
  Autoconfiguration Enabled . . . . . : Yes  
  Link-local IPv6 Address . . . . . : fe80::ca6a:ea44:aa86:882b%6(Preferred)  
  IPv4 Address. . . . . : 192.168.7.10(Preferred)  
  Subnet Mask . . . . . : 255.255.255.0  
  Default Gateway . . . . . : 192.168.7.1  
  DHCPv6 IAID . . . . . : 100666409  
  DHCPv6 Client DUID. . . . . : 00-01-00-01-2F-AD-89-76-00-0C-29-A4-72-8A  
  DNS Servers . . . . . : fec0:0:0:ffff::1%1  
                        fec0:0:0:ffff::2%1  
                        fec0:0:0:ffff::3%1
```

### ■ Step 3: Configuring the DNS Server

Command:

```
-----  
Set-DnsClientServerAddress  
-InterfaceAlias "Ethernet0"  
-ServerAddresses "192.168.7.1"
```

Explanation:

```
-----  
I ran this command to manually set the DNS server address for SRV07.
```

- InterfaceAlias "Ethernet0" → again, this is the name of the network interface.
- ServerAddresses "192.168.7.1" → this points to the domain controller's IP, so the server can resolve domain names and find vlabs07.com.

Without this, the server might not be able to join the domain because it wouldn't know where to look.

Screenshot:

```
PS C:\Users\Administrator> Set-DnsClientServerAddress -InterfaceAlias "Ethernet0" -ServerAddresses "192.168.7.1"  
Ethernet adapter Ethernet0:  
  
Connection-specific DNS Suffix . :  
Description . . . . . : Intel(R) 82574L Gigabit Network Connection  
Physical Address. . . . . : 00-0C-29-A4-72-8A  
DHCP Enabled. . . . . : No  
Autoconfiguration Enabled . . . . . : Yes  
Link-local IPv6 Address . . . . . : fe80::ca6a:ea44:aa86:882b%6(PREFERRED)  
IPv4 Address. . . . . : 192.168.7.10(PREFERRED)  
Subnet Mask . . . . . : 255.255.255.0  
Default Gateway . . . . . : 192.168.7.1  
DHCPv6 IAID . . . . . : 100666409  
DHCPv6 Client DUID. . . . . : 00-01-00-01-2F-AD-89-76-00-0C-29-A4-72-8A  
DNS Servers . . . . . : 192.168.7.1  
NetBIOS over Tcpip. . . . . : Enabled
```

### ■ Step 4: Joining SRV07 to the Domain

Command:

```
-----  
Add-Computer  
-DomainName "vlabs07.com"  
-Credential (Get-Credential)  
-Restart
```

Explanation:

```
-----  
I used Add-Computer to join SRV07 to the domain vlabs07.com. It prompted me for domain admin credentials and then automatically restarted the server.
```

After the reboot, I ran the following command to confirm the domain join:

```
(Get-WmiObject Win32_ComputerSystem).Domain
```

- `Get-WmiObject` is used to access system information from WMI (Windows Management Instrumentation).
- `Win32\_ComputerSystem` is the WMI **class** that contains details about the computer, including its domain membership.
- Adding `.` at the **end** filters the output to only show the domain name.

This returned "vlabs07.com", so I knew the server successfully joined the domain.

Screenshot:

```
PS C:\Users\Administrator> Add-Computer -DomainName "vlabs07.com" -Credential (Get-Credential) -Restart
```

```
Windows PowerShell credential request ? X
```

Enter your credentials.

User name:  ...

Password:

OK Cancel

Computer name	SRV07
Domain	vlabs07.com

```
PS C:\Users\Administrator> (Get-WmiObject Win32_ComputerSystem).Domain
vlabs07.com
```

## Task 2: Create Organizational Units and User Accounts

### Step 1: Creating the HR and IT Organizational Units (using GUI)

Explanation:

I opened \*\*Active Directory Administrative Center\*\*, clicked on my domain (`vlabs07.com`), then selected "New" > "Organizational Unit".

I created the following OUs directly under the domain root:

- HR
- IT

Screenshot:

The screenshot shows the Active Directory Administrative Center interface. At the top, there's a navigation bar with 'Manage' and 'Tools' buttons, and a red box highlights the 'Active Directory Administrative Center' link in the top-left corner. Below the navigation bar, the main window title is 'Active Directory Administrative Center > vlabs07 (local)'. The left sidebar has a tree view with 'Overview' selected, followed by 'vlabs07 (local)' which is also highlighted with a red box. Under 'vlabs07 (local)', there are several options like 'Change domain controller', 'Raise the forest functional level...', and 'Raise the domain functional level...'. A 'New' button is highlighted with a red box. To its right, a context menu is open with options: 'Search under this node', 'Properties', and a list of object types: 'Organizational Unit', 'InetOrgPerson', 'Group', 'User', and 'Computer'. The 'Organizational Unit' option is also highlighted with a red box. The main pane shows a table with columns 'Name', 'Type', and 'Description'. One row is selected, showing details for 'builtinDomain': 'Container' and 'Default container for upgrading...'.

Below this, two 'Create Organizational Unit' dialog boxes are displayed side-by-side. The left one is titled 'Create Organizational Unit: HR' and the right one is titled 'Create Organizational Unit: IT'. Both dialogs have fields for 'Name' (HR and IT respectively), 'Address' (Street, City, State/Prov, Zip/Postal, Country/Region), and 'Description'. There are checkboxes for 'Create in: DC=vlabs07,DC=com' and 'Protect from accidental deletion'. The 'Managed By' sections below both dialogs are identical, showing fields for 'Managed by' (Edit, Clear), 'Office', 'Phone number', 'Main', 'Mobile', 'Fax', 'Address' (Street, City, State/Prov, Zip/Postal, Country/Region), and 'Description'.

## ■ Step 2: Creating Users in the HR OU (using GUI)

Explanation:

Still inside **Active Directory Administrative Center**, I navigated to the HR OU and created three new user accounts:

- Sophie Lambert (Username: s.lambert)
- Liam Dupont (Username: l.dupont)
- Emma Morel (Username: e.morel)

I filled **in** their names and usernames, and assigned the default password `Passw0rd!`. I left the accounts enabled as required.

Screenshot:

The screenshot shows the Active Directory Administrative Center interface. The left navigation pane shows the tree structure: HR > IT > Dynan > Authen > Global. The main pane displays the 'User' creation form for the 'HR' organizational unit. The 'User' option in the context menu is highlighted with a red box. The form fields include: First name: Sophie, Middle initials: , Last name: Lambert, Full name: Sophie Lambert, User UPN logon: s.lambert @ vlabs07.com, User SamAccountName logon: vlabs07, and a password field containing 'Passw0rd!'. Under 'Account' options, 'Protect from accidental deletion' is unchecked. Under 'Password options', 'Other password options' is selected, with 'Password never expires' checked. Under 'Encryption options', 'User cannot change password' is checked. The bottom pane shows a list of existing users in the HR OU:

Name	Type	Description
Emma Morel	User	HR
Liam Dupont	User	HR
<b>Sophie Lambert</b>	User	HR

## ■ Step 3: Creating Users in the IT OU using PowerShell

Command:

```
New-ADUser -Name "Lucas Bernard" -GivenName "Lucas" -Surname "Bernard" -  
SamAccountName "l.bernard" -UserPrincipalName "l.bernard@vlabs07.com" -  
AccountPassword (ConvertTo-SecureString "Passw0rd$" -AsPlainText -Force) -  
Enabled $true -Path "OU=IT,DC=vlabs07,DC=com"
```

```
New-ADUser -Name "Thomas Aviles" -GivenName "Thomas" -Surname "Aviles" -  
SamAccountName "t.aviles" -UserPrincipalName "t.aviles@vlabs07.com" -
```

```
AccountPassword (ConvertTo-SecureString "Passw0rd!" -AsPlainText -Force) -  
Enabled $true -Path "OU=IT,DC=vlabs07,DC=com"
```

```
New-ADUser -Name "Chloe Girard" -GivenName "Chloe" -Surname "Girard" -  
SamAccountName "c.girard" -UserPrincipalName "c.girard@vlabs07.com" -  
AccountPassword (ConvertTo-SecureString "Passw0rd!" -AsPlainText -Force) -  
Enabled $false -Path "OU=IT,DC=vlabs07,DC=com"
```

Explanation:

I used PowerShell to create three users **in** the **IT** OU. Lucas and Thomas were enabled at creation, and Chloe was created with her account disabled as required.

Screenshot:

```
PS C:\Users\Administrator> New-ADUser -Name "Lucas Bernard" -GivenName "Lucas" -Surname "Bernard" -Sam  
AccountName "l.bernard" -UserPrincipalName "l.bernard@vlabs07.com" -AccountPassword (ConvertTo-SecureS  
tring "Passw0rd$" -AsPlainText -Force) -Enabled $true -Path "OU=IT,DC=vlabs07,DC=com"  
PS C:\Users\Administrator> New-ADUser -Name "Thomas Aviles" -GivenName "Thomas" -Surname "Aviles" -Sam  
AccountName "t.aviles" -UserPrincipalName "t.aviles@vlabs07.com" -AccountPassword (ConvertTo-SecureStr  
ing "Passw0rd$" -AsPlainText -Force) -Enabled $true -Path "OU=IT,DC=vlabs07,DC=com"  
PS C:\Users\Administrator> New-ADUser -Name "Chloe Girard" -GivenName "Chloe" -Surname "Girard" -SamAc  
countName "c.girard" -UserPrincipalName "c.girard@vlabs07.com" -AccountPassword (ConvertTo-SecureStrin  
g "Passw0rd$" -AsPlainText -Force) -Enabled $false -Path "OU=IT,DC=vlabs07,DC=com"
```

#### ■ Step 4: Confirming the **IT** Users Exist **in** the OU

Command:

```
Get-ADUser -Filter * -SearchBase "OU=IT,DC=vlabs07,DC=com" | Select Name,  
SamAccountName, Enabled
```

Explanation:

I used Get-ADUser to list the user accounts **in** the **IT** OU. This let me confirm that Lucas Bernard and Thomas Aviles were created with Enabled **set** to True, and that Chloe Girard was created with Enabled **set** to False as required.

Screenshot:

```
PS C:\Users\Administrator> Get-ADUser -Filter * -SearchBase "OU=IT,DC=vlabs07,DC=com" | Select Name, S  
amAccountName, Enabled  
  
Name      SamAccountName Enabled  
----      -----  
Lucas Bernard l.bernard      True  
Thomas Aviles t.aviles      True  
Chloe Girard c.girard      False
```

### Task 3: Delete Users

---

- Step 1: Deleting Liam Dupont **using** the GUI

Explanation:

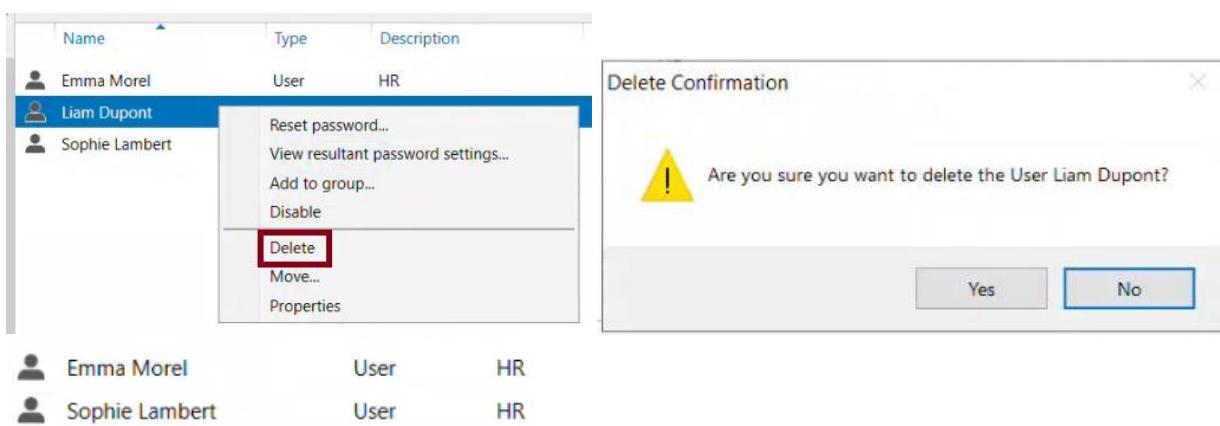
---

I opened **Active Directory Administrative Center**, navigated to the **HR OU**, right-clicked the user **Liam Dupont**, and selected **Delete**.

When prompted **for** confirmation, I clicked **Yes**.

Screenshot:

---



- Step 2: Deleting Thomas Aviles **using** PowerShell

Command:

---

```
Remove-ADUser -Identity "t.aviles" -Confirm:$false
```

Explanation:

---

I used the Remove-ADUser command to delete the user Thomas Aviles. The **-Confirm:\$false** flag skipped the confirmation **prompt**.

Instead of **using** the full Distinguished Name, I used the SamAccountName "t.aviles", which is simpler and fully supported by the **-Identity** parameter.

Screenshot:

---

```
PS C:\Users\Administrator> Remove-ADUser -Identity "t.aviles" -Confirm:$false
PS C:\Users\Administrator> Get-ADUser -Filter {Name -eq "Thomas Aviles"}
PS C:\Users\Administrator>
PS C:\Users\Administrator> Get-ADUser -Filter * -SearchBase "OU=IT,DC=vlabs07,DC=com" | Select Name, SamAccountName, Enabled
Name      SamAccountName Enabled
----      -----          -----
Lucas Bernard l.bernard      True
Chloe Girard c.girard        False
```

#### Task 4: Move Users Between Organizational Units (OUs)

---

- Step 1: Moving Sophie Lambert to the IT OU **using** the GUI

Explanation:

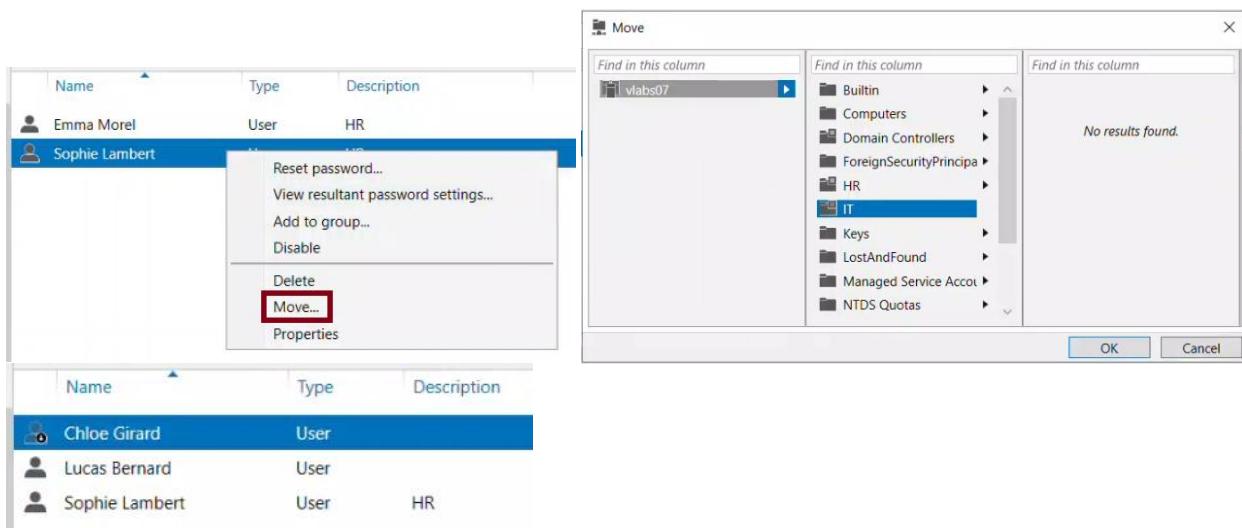
---

I opened Active Directory Administrative Center, navigated to the \*\*HR OU\*\*, right-clicked on \*\*Sophie Lambert\*\*, and selected \*\*Move...\*\*. In the dialog box, I chose the \*\*IT OU\*\* and clicked \*\*OK\*\*.

This moved the user object from HR to IT.

Screenshot:

---



- Step 2: Moving Lucas Bernard to the HR OU **using** PowerShell

Command:

---

```
Move-ADObject -Identity "CN=Lucas Bernard,OU=IT,DC=vlabs07,DC=com" -TargetPath "OU=HR,DC=vlabs07,DC=com"
```

Explanation:

---

I used the `Move-ADObject` command to move \*\*Lucas Bernard\*\* from the IT OU to the HR OU.

The `-Identity` parameter uses the full distinguished name of the user, and the `-TargetPath` specifies the destination OU.

Screenshot:

---

```
PS C:\Users\Administrator> Move-ADObject -Identity "CN=Lucas Bernard,OU=IT,DC=vlabs07,DC=com" -TargetPath "OU=HR,DC=vlabs07,DC=com"
PS C:\Users\Administrator> Get-ADUser -Filter * -SearchBase "OU=HR,DC=vlabs07,DC=com" | Select Name, SamAccountName, Enabled
Name      SamAccountName Enabled
---      -----
Emma Morel  e.morel        True
Lucas Bernard l.bernard    True
```

## Task 5: Modify Users in IT OU

---

Step 1: Assigning Sophie Lambert as Manager **for** all users **in** the **IT** OU

Command:

---

```
Get-ADUser -Filter * -SearchBase "OU=IT,DC=vlabs07,DC=com" | ForEach-Object {  
Set-ADUser -Identity $_ -Manager "s.lambert" }
```

Explanation:

---

I used Get-ADUser to list all users **in** the **IT** OU.

Then, I piped the list into **ForEach-Object**, which runs a command **for** each user found.

Inside the loop, Set-ADUser assigns Sophie Lambert (**s.lambert**) as the Manager **for** each user.

The ``$_`` symbol refers to the current user being processed **in** the loop. This allows me to update multiple users at once without writing Set-ADUser separately **for** each one.

This is the most efficient way to bulk update users **using** PowerShell.

Screenshot:

---

```
PS C:\Users\Administrator> Get-ADUser -Filter * -SearchBase "OU=IT,DC=vlabs07,DC=com" | ForEach-Object { Set-ADUser -Identity $_ -Manager "s.lambert" }
```

Step 2: Verifying that the Manager field was updated

Command:

```
Get-ADUser -Filter * -SearchBase "OU=IT,DC=vlabs07,DC=com" -Properties Manager | Select Name, SamAccountName, Manager
```

Explanation:

This command lists all users **in** the **IT** OU, including their Manager field. It verifies that the "Manager" attribute was successfully **set** to Sophie Lambert.

If the Manager column shows her Distinguished Name, then the update worked.

Screenshot:

```
PS C:\Users\Administrator> Get-ADUser -Filter * -SearchBase "OU=IT,DC=vlabs07,DC=com" -Properties Manager | Select Name, SamAccountName, Manager  
  
Name      SamAccountName Manager  
-----  
Sophie Lambert s.lambert    CN=Sophie Lambert,OU=IT,DC=vlabs07,DC=com  
Chloe Girard   c.girard     CN=Sophie Lambert,OU=IT,DC=vlabs07,DC=com
```

## Task 6: Enable and Disable User Accounts

---

■ Step 1: Disable the user account Emma Morel **using** Active Directory Administrative Center

Explanation:

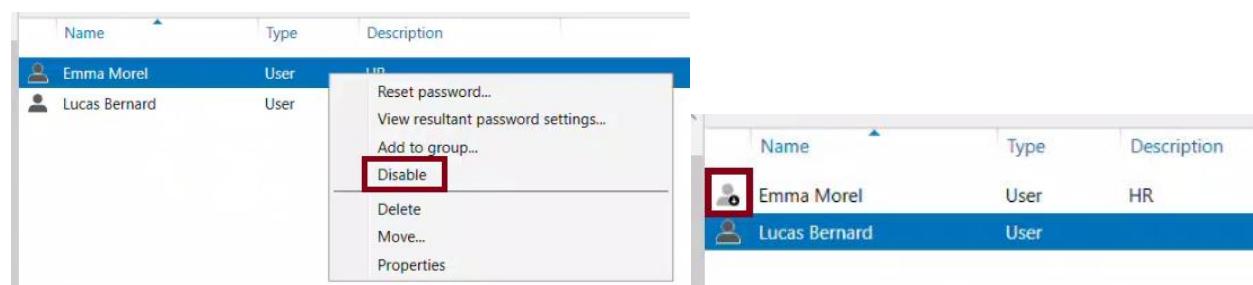
---

I opened the **Active Directory Administrative Center** on DC107. From the left panel, I navigated to the **HR OU**, selected the user **Emma Morel**, right-clicked her account, and chose **Disable**.

A down-arrow appeared on her user icon, confirming the account was disabled.

Screenshot:

---



■ Step 2: Enable the user account Chloe Girard **using** PowerShell

Command:

---

```
Enable-ADAccount -Identity "c.girard"
```

Explanation:

---

I used the `Enable-ADAccount` command to activate Chloe Girard's user account **in the IT OU**.

Her account had been created **in** a disabled state earlier **in** the lab. The `'-Identity'` parameter uses her SamAccountName **for** simplicity.

To confirm the change, I ran:

```
Get-ADUser -Identity "c.girard" -Properties Enabled | Select Name, SamAccountName, Enabled
```

This showed that the 'Enabled' attribute was now **set to 'True'**.

Screenshot:

---

```
PS C:\Users\Administrator> Enable-ADAccount -Identity "c.girard"
PS C:\Users\Administrator> Get-ADUser -Identity "c.girard" -Properties Enabled | Select Name, SamAccountName, Enabled
Name      SamAccountName Enabled
-----      -----
Chloe Girard c.girard        True
```

## Task 7: Lock and Unlock User Accounts

---

- Step 1: Find all locked-out user accounts **using** PowerShell

Command:

---

```
Search-ADAccount -LockedOut -UsersOnly | Format-Table Name, SamAccountName, LockedOut -AutoSize
```

Explanation:

---

I ran this command to check **if** any user accounts were currently locked out.

- Search-ADAccount with **-LockedOut** lists accounts that are **in** a locked state.
- **-UsersOnly** filters the results to only show user accounts.
- **Format-Table** makes the output easier to read.

**In** this case, the output showed no results, which means no accounts were locked.

Screenshot:

---

```
PS C:\Users\Administrator> Search-ADAccount -LockedOut -UsersOnly | Format-Table Name, SamAccountName, LockedOut -AutoSize
```

- Step 2: Practice unlocking Chloe Girard **using** PowerShell

Command:

---

```
Unlock-ADAccount -Identity "c.girard"
```

Explanation:

---

Even though Chloe Girard's account was not actually locked, I used this command to simulate the unlock **process**.

The command uses her SamAccountName as the identity. **If** the account is not locked, the command completes without any output or error, which is expected.

Screenshot:

---

```
PS C:\Users\Administrator> Unlock-ADAccount -Identity "c.girard"
```

## Task 8: Configure Roaming Profiles on SRV07

---

■ Step 1: Create the Shared Folder C:\Profiles on SRV07

Command:

---

```
New-Item -Path "C:\Profiles" -ItemType Directory
```

```
New-SmbShare -Name "Profiles$" -Path "C:\Profiles" -FullAccess "Authenticated Users", "Administrators" -ChangeAccess "Domain Users"
```

Explanation:

---

The first command creates the folder `C:\Profiles` which will store user profiles.

The second command shares that folder **using New-SmbShare**:

- `-Name "Profiles\$"` → This is the name of the share. The `\$` makes it **hidden** (not visible **in** network browsing).
- `-Path "C:\Profiles"` → This is the folder being shared.
- `-FullAccess "Authenticated Users", "Administrators"` → These users have full control over the share.
- `-ChangeAccess "Domain Users"` → Domain users have modify access, which is enough **for** roaming profile creation.

The **New-SmbShare** command shares the folder with full access to Authenticated Users and Administrators, and change permissions to Domain Users.

Screenshot:

---

```
PS C:\Users\Administrator> New-Item -Path "C:\Profiles" -ItemType Directory

Directory: C:\

Mode          LastWriteTime    Length Name
----          -----          ---- 
d----  5/7/2025 10:41 PM      0     Profiles

PS C:\Users\Administrator> New-SmbShare -Name "Profiles$" -Path "C:\Profiles" -FullAccess "Authenticated Users", "Administrators" -ChangeAccess "Domain Users"

Name      ScopeName Path        Description
----      -----   ----        -----
Profiles$ *       C:\Profiles
```

## ■ Step 2: Configure NTFS Permissions on C:\Profiles on SRV07

Command:

```
icacls "C:\Profiles" /grant "Authenticated Users:(OI)(CI)(M)"  
"Administrators:(OI)(CI)(F)" "SYSTEM:(OI)(CI)(F)"
```

Explanation:

I used the `icacls` command to apply **\*\*NTFS permissions\*\*** to the C:\Profiles folder.

Here's what each part means:

- **\*\*"Authenticated Users:(OI)(CI)(M)"\*\***
  - This grants Modify **(M)** permission to any domain user who logs **in**.
  - **(OI)** means Object Inherit → permissions apply to files inside the folder.
  - **(CI)** means Container Inherit → permissions apply to subfolders too.
- **\*\*"Administrators:(OI)(CI)(F)"\*\***
  - This gives full control **(F)** to all local/domain Administrators.
- **\*\*"SYSTEM:(OI)(CI)(F)"\*\***
  - This allows the Windows system account to fully manage files **for** roaming profile storage.

These permissions are necessary so Windows can **create**, **read**, and **update** profile folders when users log **in**.

Screenshot:

```
PS C:\Users\Administrator> icacls "C:\Profiles" /grant "Authenticated Users:(OI)(CI)(M)" "Administrators:(OI)(CI)(F)" "SYSTEM:(OI)(CI)(F)"  
processed file: C:\Profiles  
Successfully processed 1 files; Failed processing 0 files  
PS C:\Users\Administrator>
```

## ■ Step 3: Assign Roaming Profile to Emma Morel **using** GUI (Active Directory Administrative Center) on DC107

Explanation:

I opened the Active Directory Administrative Center on DC107.

Then I followed these steps to assign a roaming profile path to Emma Morel:

1. Navigated to the HR OU under vlabs07 **(local)**.
2. I right clicked on the user account **for** Emma Morel and selected Properties.
3. Scrolled down to the "Profile" section.
4. **In** the "Profile path" field, I entered:  
\\SRV07\Profiles\\$e.morel

Explanation of the path:

- \\SRV07 → The hostname of the file server.
- Profiles\$ → The shared (and **hidden**) folder we created earlier.
- e.morel → Matches her SamAccountName and will become her profile folder name.

5. Clicked OK to save the changes.

This **configuration** ensures that Emma's profile **data** is redirected to the network and saved under SRV07 when she logs into domain-joined computers.

Screenshot:

The screenshot shows the Windows Active Directory Users and Computers management console. A list of users is displayed in a table with columns for Name, Type, and Description. Two users are visible: 'Emma Morel' and 'Lucas Bernard'. A context menu is open over 'Lucas Bernard', with the 'Properties' option highlighted by a red box. Below the table, a 'Profile' section is shown with the 'Profile path' field containing '\\SRV07\Profiles\$\l.bernard'.

■ Step 4: Assign Roaming Profile to Lucas Bernard **using** PowerShell on DC107

Command:

```
Set-ADUser -Identity "l.bernard" -ProfilePath "\\SRV07\Profiles$\l.bernard"
```

Explanation:

I used the Set-ADUser command to assign a roaming profile path **for** Lucas Bernard.

- -Identity "l.bernard" → specifies the user by their SamAccountName.
- -ProfilePath "\\SRV07\Profiles\$\l.bernard" → points to the shared folder on SRV07, **where** his roaming profile will be stored.

This ensures that when Lucas logs into a domain computer, his profile is redirected to SRV07 and saved **in** the folder named after his username.

Screenshot:

The screenshot shows a PowerShell window with the command PS C:\Users\Administrator> Set-ADUser -Identity "l.bernard" -ProfilePath "\\SRV07\Profiles\$\l.bernard" entered and executed. Below the command, a 'Profile' section is shown with the 'Profile path' field containing '\\SRV07\Profiles\$\l.bernard'.

## Step 5: Verify Profile Folder Creation from Client07

Explanation:

To confirm that roaming profiles were **set** up correctly, I logged into ClientXX **using** both accounts:

- Emma Morel
- Lucas Bernard

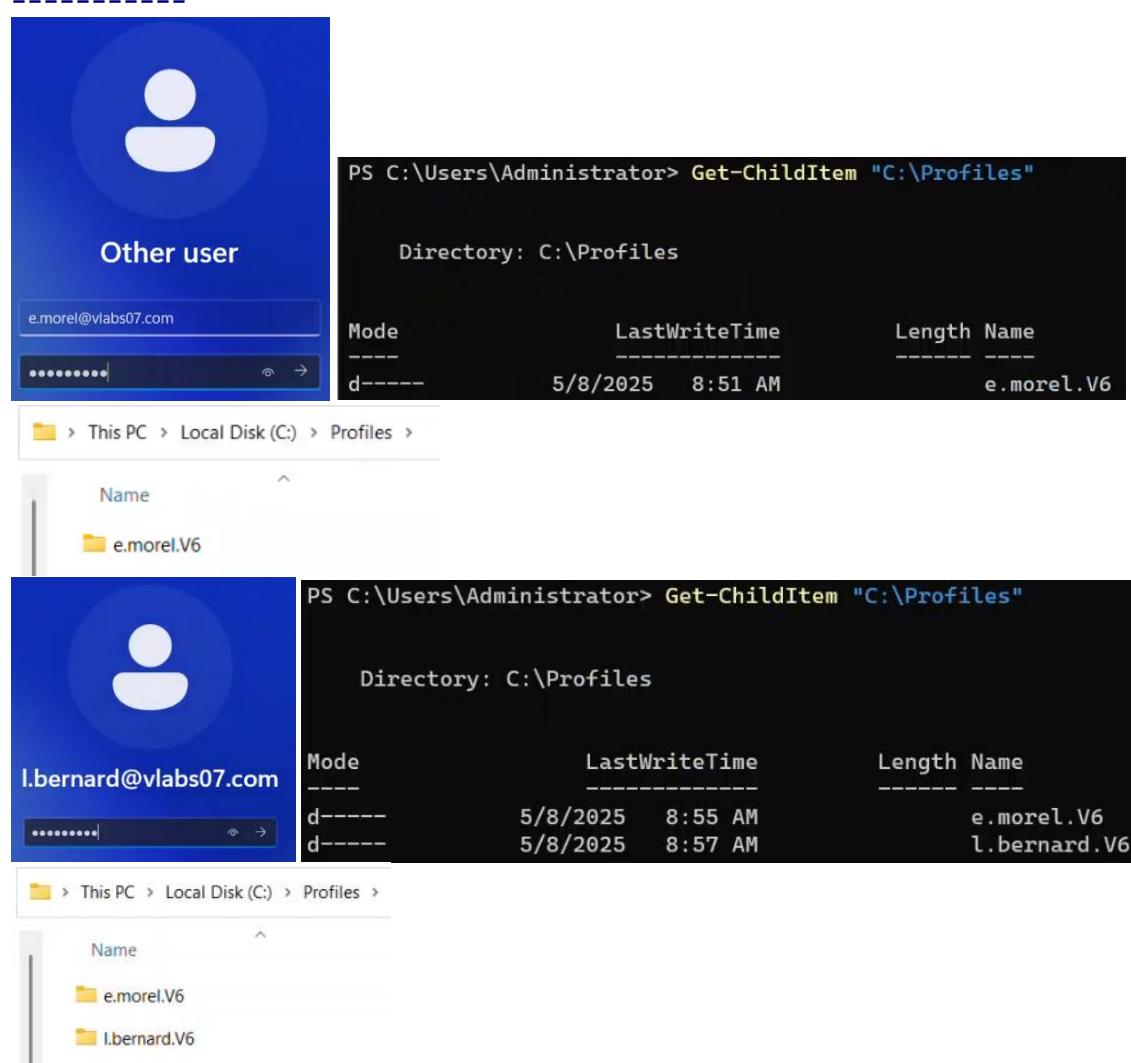
After each login, I went back to SRV07 and checked the C:\Profiles folder.

Two new folders were created:

- C:\Profiles\e.morel
- C:\Profiles\l.bernard

These folders were automatically created the first time each user logged **in**. This confirms that the profile redirection is working and that both users are **using** roaming profiles.

Screenshot:



The screenshot displays two separate sessions. The top session shows the PowerShell command `Get-ChildItem "C:\Profiles"` running on behalf of the 'Administrator' account. The output lists a single folder named 'e.morel.V6' located at 'C:\Profiles'. The bottom session shows the same command running on behalf of the 'l.bernard@vlabs07.com' account. It lists two folders: 'e.morel.V6' and 'l.bernard.V6'. Both sessions also show a corresponding Windows File Explorer window showing the same folder structures.

PS C:\Users\Administrator> Get-ChildItem "C:\Profiles"

Directory: C:\Profiles

Mode	LastWriteTime	Name
d----	5/8/2025 8:51 AM	e.morel.V6

PS C:\Users\Administrator> Get-ChildItem "C:\Profiles"

Directory: C:\Profiles

Mode	LastWriteTime	Name
d----	5/8/2025 8:55 AM	e.morel.V6
d----	5/8/2025 8:57 AM	l.bernard.V6

## Task 9: Create a Template User for HR (DC107)

---

- Step 1: Create the HR template user account **using** PowerShell

Command:

---

```
New-ADUser -Name "HR Template" -SamAccountName "hr.template" -UserPrincipalName "hr.template@vlabs.com" -Department "HR" -Title "HR Assistant" -Enabled $false -Description "Template user for HR new hires." -Path "OU=HR,DC=vlabs07,DC=com" -ProfilePath "\\\$RV07\Profiles$\%username%"
```

Explanation:

---

I used the New-ADUser command to create a reusable, disabled template account **for** HR.

- -Name	→ This sets the display name of the template <b>in</b> Active Directory.
- -SamAccountName	→ This is the legacy logon name used <b>for</b> older systems.
- -UserPrincipalName	→ This defines the modern email-style login ID.
- -Department	→ Tags the account under the HR department.
- -Title	→ Assigns the job title "HR Assistant".
- -Enabled \$false	→ Keeps the template account disabled by default.
- -Description	→ Notes that this is a template used <b>for</b> onboarding.
- -Path	→ Places the template <b>in</b> the correct Organizational Unit.
- -ProfilePath	→ Sets up the base path <b>for</b> roaming profiles with %username%.

Screenshot:

---

```
PS C:\Users\Administrator> New-ADUser -Name "HR Template" -SamAccountName "hr.template" -UserPrincipalName "hr.template@vlabs.com" -Department "HR" -Title "HR Assistant" -Enabled $false -Description "Template user for HR new hires." -Path "OU=HR,DC=vlabs07,DC=com" -ProfilePath "\\\$RV07\Profiles$\%username%"
```

- Step 2: Validate the HR Template User Account

Command:

---

```
Get-ADUser -Identity "hr.template" -Properties Name, Enabled, Title, Department, Description, ProfilePath | Select-Object Name, Enabled, Title, Department, Description, ProfilePath
```

Explanation:

---

This command checks that the HR template user was created correctly and includes all required attributes.

**It** displays the users full name, status, title, department, description, and assigned profile path.

Screenshot:

```
PS C:\Users\Administrator> Get-ADUser -Identity "hr.template" -Properties Name, Enabled, Title, Department, Description, ProfilePath | Select-Object Name, Enabled, Title, Department, Description, ProfilePath
```

```
Name      : HR Template
Enabled   : False
Title     : HR Assistant
Department : HR
Description : Template user for HR new hires.
ProfilePath : \\SRV07\Profiles$\%username%
```

	Name	Type	Description
	Emma Morel	User	HR
	HR Template	User	Template user for HR new...
	Lucas Bernard	User	

## Task 10: Verify New Users from DC207 (RODC Core Server)

---

### ■ Step 1: Verify Replication of Newly Created Users

Command:

---

```
Get-ADUser -Filter * | Select-Object Name, SamAccountName, Enabled
```

Explanation:

---

This command lists all users currently replicated to the RODC (DC2XX). It confirms that user accounts from HR and IT OUs were successfully replicated.

- You **should** see users such as: e.morel, l.bernard, s.lambert, etc.
- Accounts like hr.template (disabled) and deleted accounts may also appear depending on replication status.

Screenshot:

---

Name	SamAccountName	Enabled
Guest	Guest	False
Administrator	Administrator	True
krbtgt	krbtgt	False
krbtgt_443	krbtgt_443	False
LAB07\$	LAB07\$	True
Sophie Lambert	s.lambert	True
Emma Morel	e.morel	True
Lucas Bernard	l.bernard	True
Chloe Girard	c.girard	True
HR Template	hr.template	False

### ■ Step 2: Filter Users by OU (HR and IT only)

Command:

---

```
Get-ADUser -Filter * -SearchBase "OU=HR,DC=vlabs07,DC=com" | Select-Object Name, SamAccountName
```

```
Get-ADUser -Filter * -SearchBase "OU=IT,DC=vlabs07,DC=com" | Select-Object Name, SamAccountName
```

Explanation:

---

These commands list users specifically within the HR and IT Organizational Units.

This helps verify that both OUs and their respective users exist on the RODC and were replicated.

Screenshot:

```
PS C:\Users\Administrator.VLABS07> Get-ADUser -Filter * -SearchBase "OU=HR,DC=vlabs07,DC=com" | Select-Object Name, SamAccountName
Name      SamAccountName
-----
Emma Morel e.morel
Lucas Bernard l.bernard
HR Template hr.template
PS C:\Users\Administrator.VLABS07> Get-ADUser -Filter * -SearchBase "OU=IT,DC=vlabs07,DC=com" | Select-Object Name, SamAccountName
Name      SamAccountName
-----
Sophie Lambert s.lambert
Chloe Girard c.girard
```

### ■ Step 3: Confirm Deleted or Disabled Accounts Are Reflected

Command:

```
Get-ADUser -Filter {Enabled -eq $false} | Select-Object Name, SamAccountName
```

Explanation:

This command finds all user accounts that are disabled **in** the domain. You **should** see accounts like "hr.template" and possibly "c.girard" **if** she hasn't been re-enabled. **If** a user was deleted (e.g., t.aviles), they **should** not appear **in** any list – confirming deletion was replicated.

Screenshot:

```
PS C:\Users\Administrator.VLABS07> Get-ADUser -Filter {Enabled -eq $false} | Select-Object Name, SamAccountName
Name      SamAccountName
-----
Guest    Guest
krbtgt  krbtgt
krbtgt_443 krbtgt_443
Emma Morel e.morel
HR Template hr.template
```