Lab 6 – Managing Computer Objects and Organizational Units

Task 1: Managing Computer Objects

Step 1: Create Workstations OU using ADAC

System: DC107 (Primary Domain Controller for vlabs07.com)

Explanation:

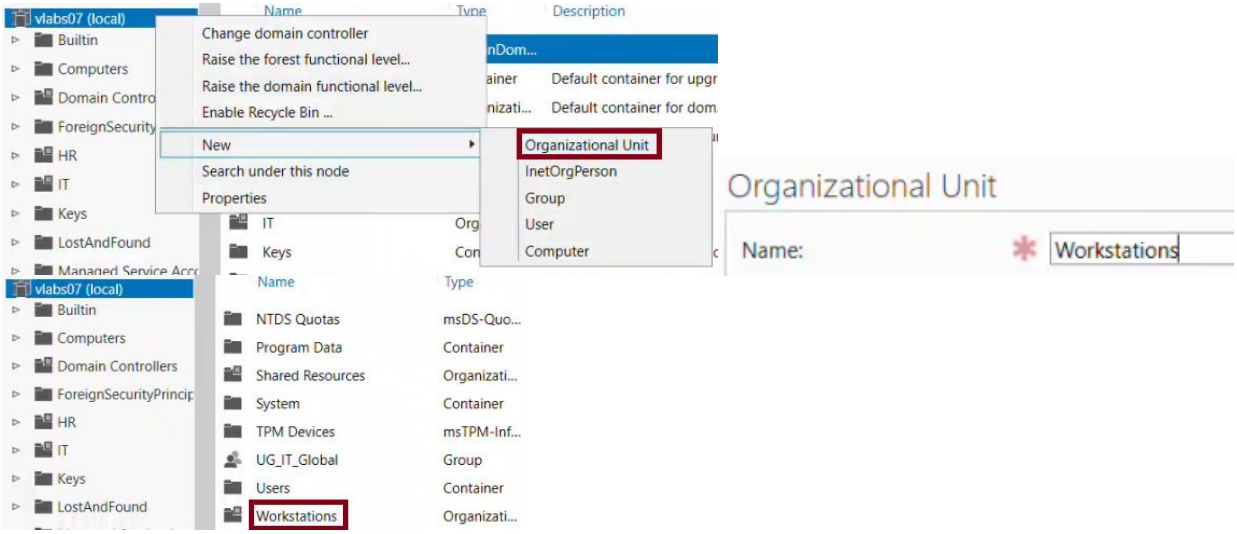
I used Active Directory Administrative Center (ADAC) on DC107 to create a new Organizational Unit called Workstations at the root of the vlabs07.com domain.

This OU will be used to store and manage computer accounts for client machines.

Steps:

- Logged in to DC107
- Opened Active Directory Administrative Center
- Navigated to the vlabs07.com domain
- Right-clicked the domain name > New > Organizational Unit
- Named the OU: Workstations
- Clicked OK to create it

Screenshot:



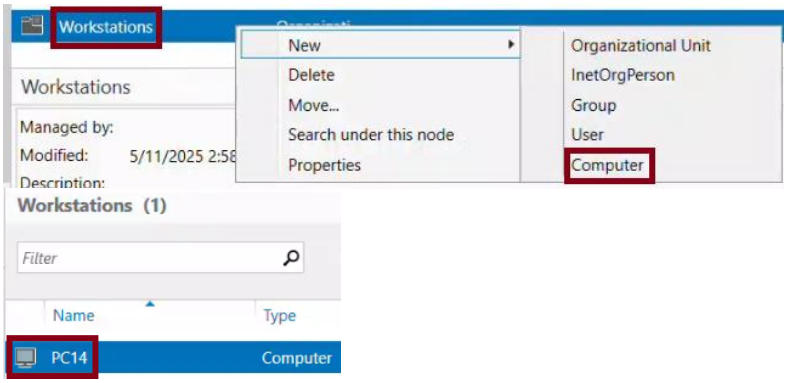
▨ Step 2: Create computer object PC14 using ADAC
System: DC107

Explanation:

I created a new computer object named PC14 inside the Workstations OU using ADAC.

Steps:
- In ADAC, navigated to the Workstations OU
- Right-clicked > New > Computer
- Entered the name: PC14
- Clicked OK to create the computer object

Screenshot:



▨ Step 3: Verify the creation of PC14 using PowerShell
System: DC107

Command:

```
Get-ADComputer -Identity "PC14"
```

Explanation:

I used this command to verify that PC14 was successfully created in Active Directory.

- Get-ADComputer retrieves computer account information
- -Identity "PC14" specifies the object name to look up

Screenshot:

```
PS C:\Users\Administrator> Get-ADComputer -Identity "PC14"


DistinguishedName : CN=PC14,OU=Workstations,DC=vlabs07,DC=com
DNSHostName       :
Enabled           : True
Name              : PC14
ObjectClass       : computer
ObjectGUID        : 3538bf09-f7dc-4a8d-8377-f3fa2bcb837d
SamAccountName    : PC14$
SID               : S-1-5-21-2428485534-1961598418-1246186656-1120
UserPrincipalName :
```

▨ Step 4: Rename PC14 to PC14-Updated using PowerShell
System: DC107

Command:

```
Rename-ADObject -Identity "CN=PC14,OU=Workstations,DC=vlabs07,DC=com" -
NewName "PC14-Updated"
```

Verification:

```
Get-ADComputer -Filter 'Name -like "PC14*"' | Select-Object Name,
DistinguishedName
```

Explanation:

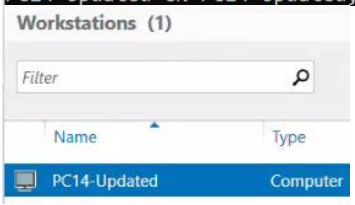
I renamed the computer object from PC14 to PC14-Updated using this command. Then I verified that the rename was successful by searching for all objects whose names start with "PC14".

- Rename-ADObject changes the Common Name (CN) of the object
- The Identity must match the full Distinguished Name of the original object
- The second command uses a name filter and displays the name and path of the renamed computer

Screenshot:

```
PS C:\Users\Administrator> Rename-ADObject -Identity "CN=PC14,OU=Workstations,DC=vlabs07,DC=com" -NewN
ame "PC14-Updated"
PS C:\Users\Administrator> Get-ADComputer -Filter 'Name -like "PC14*"' | Select-Object Name, Distingui
shedName

Name         DistinguishedName
----         -----------------
PC14-Updated CN=PC14-Updated,OU=Workstations,DC=vlabs07,DC=com
```



▨ Step 5: Remove PC14-Updated using PowerShell
System: DC107

Command:

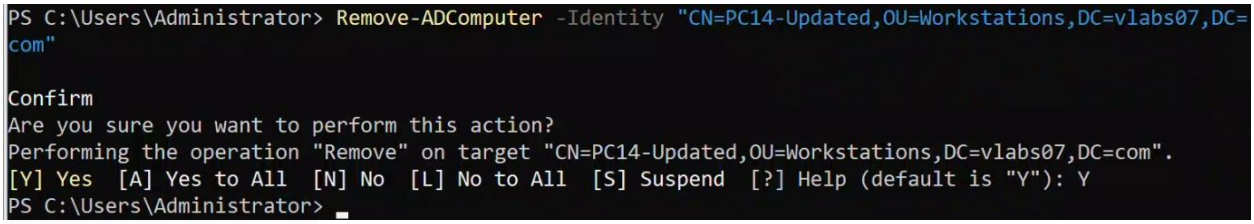
```
Remove-ADComputer -Identity "CN=PC14-
Updated,OU=Workstations,DC=vlabs07,DC=com" -Confirm:$false
```

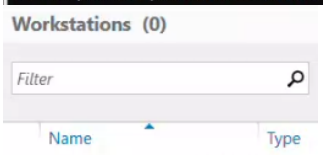
Explanation:

I deleted the computer object named PC14-Updated using its full Distinguished Name, since the simple name or SamAccountName did not match after the rename.

- Remove-ADComputer deletes a computer object from Active Directory
- The Identity must include the full path if the objects name was changed
- -Confirm:$false skips the confirmation prompt and deletes the object directly

Screenshot:





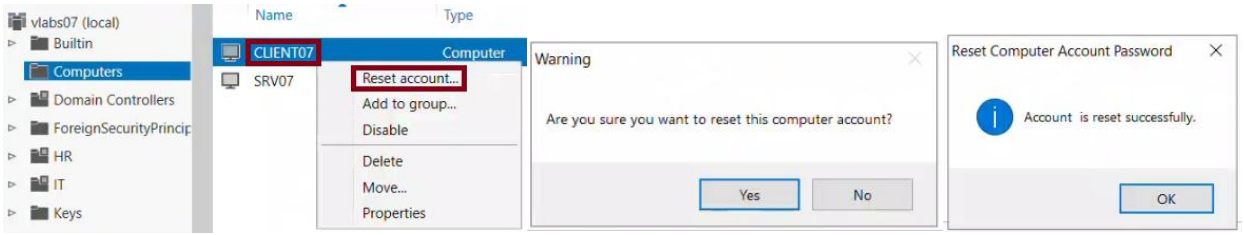
▨ Step 6: Reset the secure channel for Client07 using ADAC
System: DC107

Explanation:

I used ADAC to reset the secure channel for Client07 in case it had issues communicating with the domain.

Steps:
- In ADAC, located the computer object named Client07
- Right-clicked it > selected Reset Account
- Clicked OK to confirm

Screenshot:



▨ Step 7: Test the secure channel from Client07 using PowerShell
System: Client07

Command:

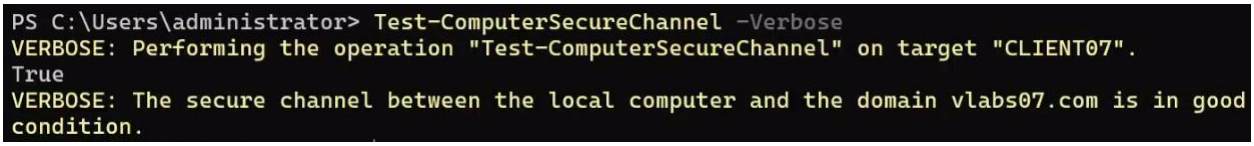
```
Test-ComputerSecureChannel -Verbose
```

Explanation:

I ran this command from the Windows 11 client (Client07) to check the secure channel with the domain.

- Test-ComputerSecureChannel verifies the trust relationship between the local computer and the domain
- -Verbose provides detailed output
- The result "True" and the message confirm the secure channel is working correctly

Screenshot:



Task 2: Redirecting the Computers Container

▨ Step 1: Verify the default Computers container location using PowerShell
System: DC107

Command:

```
(Get-ADDomain).ComputersContainer
```

Explanation:

I ran this command to see the current default container for newly joined computer objects.

- Get-ADDomain retrieves domain-level configuration
- .ComputersContainer shows where new computer accounts are placed by default
- By default, this points to: CN=Computers,DC=vlabs07,DC=com

Screenshot:

▨ Step 2: Redirect the default Computers container to the Workstations OU using PowerShell
System: DC107

Command:

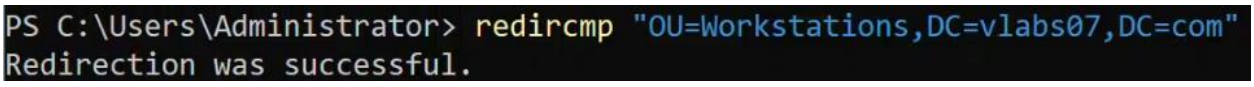
```
redircmp "OU=Workstations,DC=vlabs07,DC=com"
```

Explanation:

I used the redircmp command to change the default location for new computer objects.

- redircmp permanently redirects the default container for computer accounts
- This will move new domain-joined computers into the Workstations OU automatically

Screenshot:



▨ Step 3: Verify that the redirection was applied using PowerShell
System: DC107

Command:

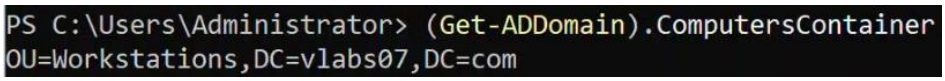
```
(Get-ADDomain).ComputersContainer
```

Explanation:

I ran the same command again to verify that the redirection was successful.

- The output should now show: OU=Workstations,DC=vlabs07,DC=com
- This confirms that new computers will now be created in the Workstations OU by default

Screenshot:



Task 3: Moving Computer Objects

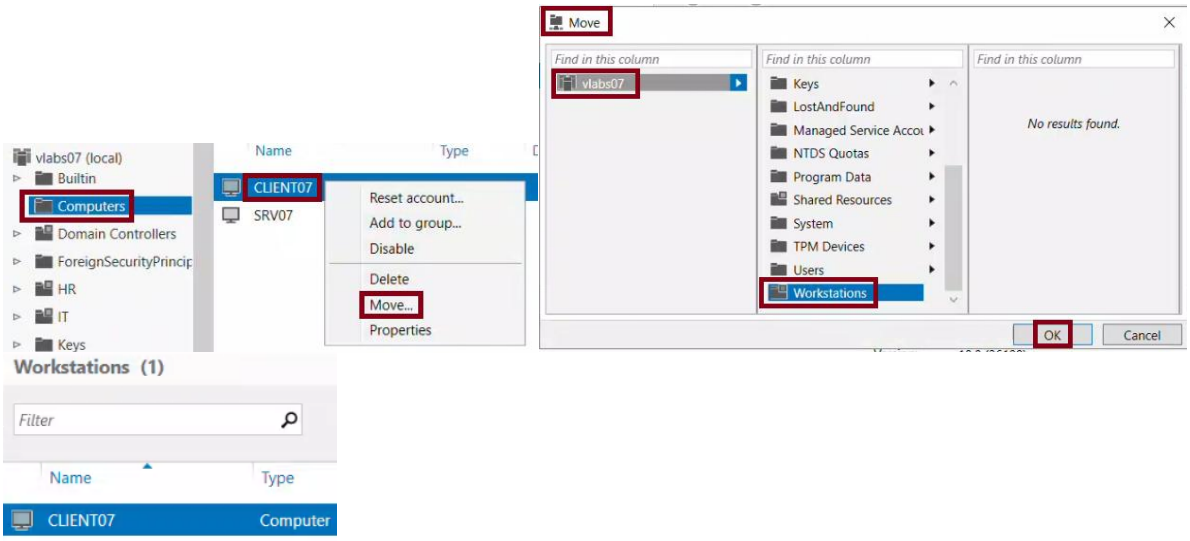
▨ Step 1: Move Client07 from the Computers container to the Workstations OU **using** ADAC
System: DC107

Explanation:

I used Active Directory Administrative Center **(**ADAC**)** to move the Client07 computer object into the Workstations OU.

Steps:
- Opened ADAC on DC107
- Navigated to the Computers container
- Right-clicked Client07 and selected "Move"
- Chose the Workstations OU
- Clicked OK to complete the move

Screenshot:



▨ Step 2: Create an OU named Servers using PowerShell
System: DC107

Command:

```
New-ADOrganizationalUnit -Name "Servers" -Path "DC=vlabs07,DC=com" -
ProtectedFromAccidentalDeletion $true
```

Verification:

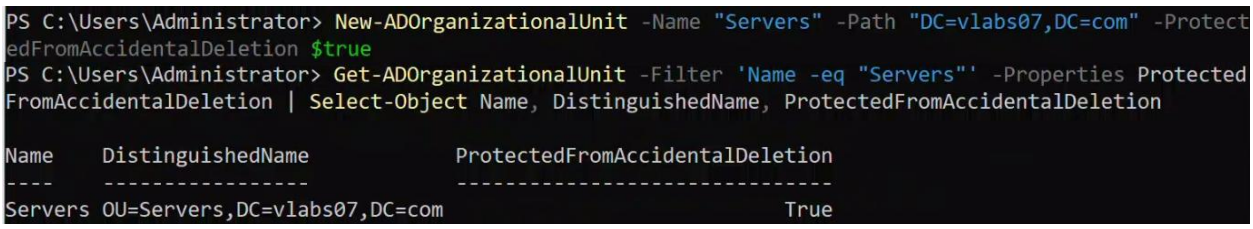
```
Get-ADOrganizationalUnit -Filter 'Name -eq "Servers"' -Properties
ProtectedFromAccidentalDeletion | `
Select-Object Name, DistinguishedName, ProtectedFromAccidentalDeletion
```

Explanation:

I created a new Organizational Unit named Servers at the root of the vlabs07.com domain.

- New-ADOrganizationalUnit creates a new OU in Active Directory
- -Path defines where the OU should be created (in this case, at the domain root)
- -ProtectedFromAccidentalDeletion is a switch that controls deletion protection
  → Adding $true explicitly enables this protection
  → If I omit it or use $false, the OU can be deleted without a warning
- I used the second command to verify that the OU was created and that the protection flag is set to True

Screenshot:



▨ Step 3: Move SRV07 to the Servers OU using PowerShell
System: DC107

Command:

```
Get-ADComputer -Identity "SRV07" | Move-ADObject -TargetPath
"OU=Servers,DC=vlabs07,DC=com"
```

Verification:

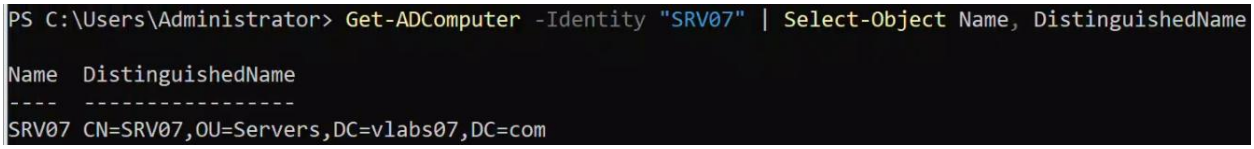
```
Get-ADComputer -Identity "SRV07" | Select-Object Name, DistinguishedName
```

Explanation:

I used this command to move the SRV07 computer object into the Servers OU.

- Get-ADComputer retrieves the computer object from Active Directory
- Move-ADObject moves it to the specified target OU
- I confirmed the move was successful by checking the DistinguishedName, which now shows it is located in the Servers OU

Screenshot:



Task 4: Changing the Default Quota for Creating Computer Objects

---

▨ Step 1: Change the Default Quota for creating Computer Objects to 0 using PowerShell
System: DC107

Command:

```
Set-ADDomain -Identity "vlabs07.com" -Replace @{ "ms-DS-MachineAccountQuota" = 0 }
```

Explanation:

By default, any authenticated user can join up to 10 computers to the domain.

- Set-ADDomain modifies domain-level properties
- -Replace allows me to update the value of specific attributes
- ms-DS-MachineAccountQuota controls how many computer accounts a non-admin user can create
- Setting it to 0 means that only domain admins (or users explicitly delegated) can join computers to the domain

Screenshot:

```
PS C:\Users\Administrator> Set-ADDomain -Identity "vlabs07.com" -Replace @{ "ms-DS-MachineAccountQuota
" = 0 }
```

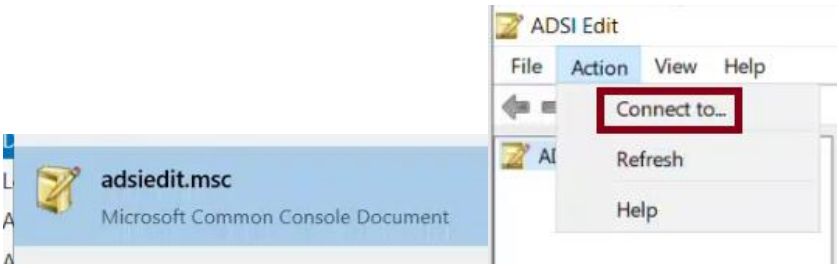
▨ Step 2: Verify the change using ADSI Edit
System: DC107

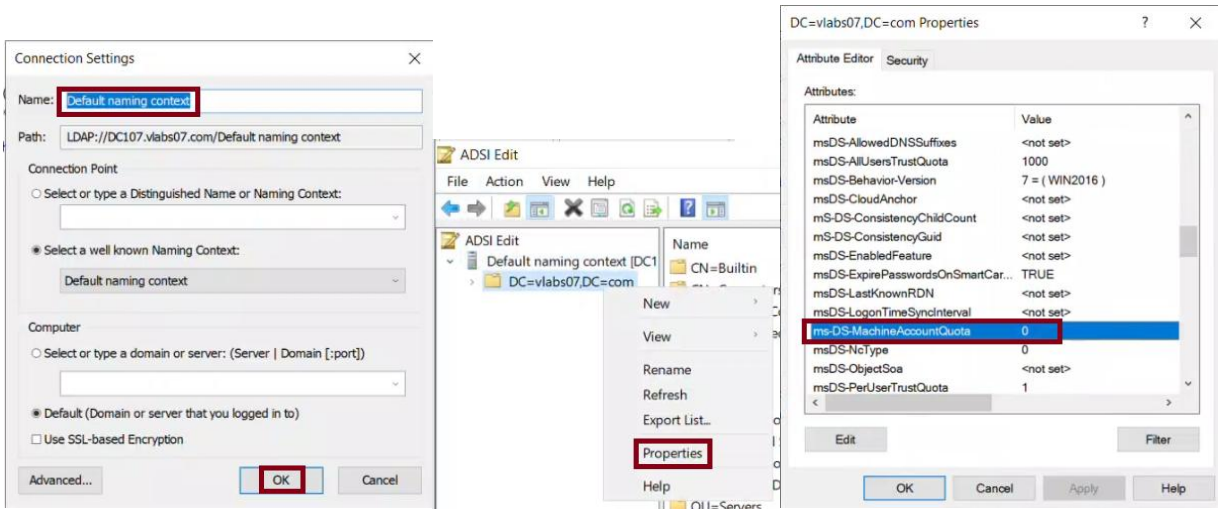
Explanation:

I used ADSI Edit to visually confirm that the ms-DS-MachineAccountQuota attribute was set to 0.

Steps:
- Open ADSI Edit (type `adsiedit.msc` in the Run dialog)
- Connect to the "Default naming context"
- Expand: DC=vlabs07,DC=com
- Right-click the domain > Properties
- Scroll to `ms-DS-MachineAccountQuota` and verify its value is set to 0

Screenshot:




Connection Settings
Name: Default naming context
Path: LDAP://DC107.vlabs07.com/Default naming context
Connection Point
Select or type a Distinguished Name or Naming Context:
Select a well known Naming Context:
Default naming context
Computer
Select or type a domain or server: (Server | Domain [:port])
Default (Domain or server that you logged in to)
Use SSL-based Encryption
Advanced...
OK
Cancel
ADSI Edit
File Action View Help
ADSI Edit
Default naming context [DC1
DC=vlabs07,DC=com
Name
CN=Builtin
New
View
Rename
Refresh
Export List...
Properties
Help
DC=vlabs07,DC=com Properties
Attribute Editor
Security
Attributes:
Attribute
Value
msDS-AllowedDNSSuffixes
<not set>
msDS-AllUsersTrustQuota
1000
msDS-Behavior-Version
7 = (WIN2016)
msDS-CloudAnchor
<not set>
msDS-ConsistencyChildCount
<not set>
msDS-ConsistencyGuid
<not set>
msDS-EnabledFeature
<not set>
msDS-ExpirePasswordsOnSmartCar...
TRUE
msDS-LastKnownRDN
<not set>
msDS-LogonTimeSyncInterval
<not set>
ms-DS-MachineAccountQuota
0
msDS-NcType
0
msDS-ObjectSoa
<not set>
msDS-PerUserTrustQuota
1
Edit
Filter
OK
Cancel
Apply
Help

Task 5: Managing Organizational Units (OUs)

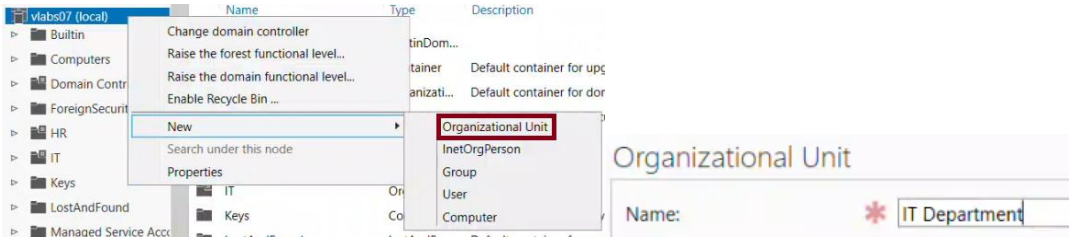
▨ Step 1: Create IT Department OU under vlabs07.com using ADAC
System: DC107

Explanation:

I used Active Directory Administrative Center (ADAC) to create a new Organizational Unit called IT Department at the root of the vlabs07.com domain.

Steps:
- Logged in to DC107
- Opened Active Directory Administrative Center
- Navigated to the vlabs07.com domain
- Right-clicked the domain name > New > Organizational Unit
- Named the OU: IT Department
- Clicked OK to create it

Screenshot:



▨ Step 2: Verify that IT Department OU has been created using PowerShell
System: DC107

Command:

```
Get-ADOrganizationalUnit -Filter 'Name -eq "IT Department"' | Select-Object Name, DistinguishedName
```

Explanation:

I used this command to confirm that the IT Department OU was created successfully.

- Get-ADOrganizationalUnit retrieves OU information from Active Directory
- -Filter 'Name -eq "IT Department"' searches for the exact OU by name
- Select-Object shows the name and location in the directory

Screenshot:

```
PS C:\Users\Administrator> Get-ADOrganizationalUnit -Filter 'Name -eq "IT Department"' | Select-Object
 Name, DistinguishedName

Name          DistinguishedName
----          -----------------
IT Department OU=IT Department,DC=vlabs07,DC=com
```

▨ Step 3: Add a description to the IT Department OU using PowerShell
System: DC107

Command:

```
Set-ADOrganizationalUnit -Identity "OU=IT Department,DC=vlabs07,DC=com" -
Description "Handles IT operations and security"
```

Verification:

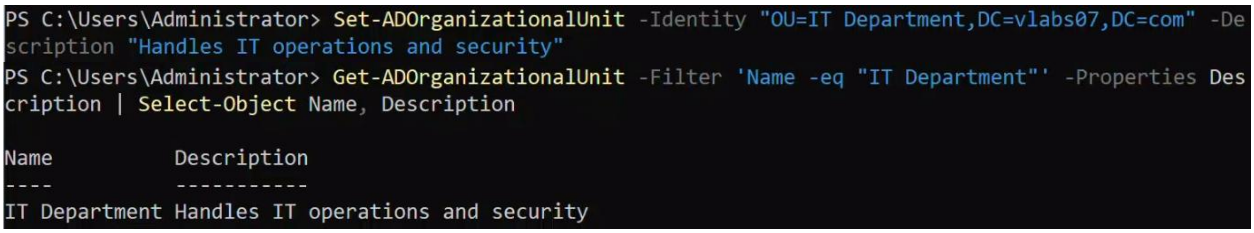
```
Get-ADOrganizationalUnit -Filter 'Name -eq "IT Department"' -Properties
Description | Select-Object Name, Description
```

Explanation:

I added a description to the IT Department OU to indicate its purpose.

- Set-ADOrganizationalUnit modifies properties of an existing OU
- -Identity uses the full Distinguished Name of the OU
- -Description sets a note that appears in Active Directory properties
- The verification command confirms that the description is visible and correct in AD

Screenshot:



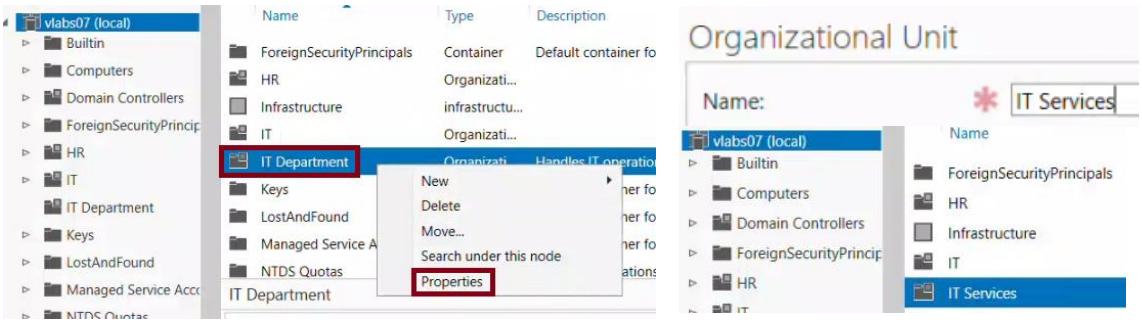
▨ Step 4: Rename IT Department to IT Services using ADAC
System: DC107

Explanation:

I used Active Directory Administrative Center (ADAC) to rename the IT Department OU to IT Services.

Steps:
- Opened ADAC on DC107
- Navigated to the vlabs07.com domain
- Located the IT Department OU in the right-hand pane
- Double-clicked the IT Department OU to open its properties
- In the "Name" field, changed the name to: IT Services
- Clicked OK to apply the change

Screenshot:



Step 5: Create Finance OU using ADAC and verify the creation using PowerShell

System: DC107

Explanation:

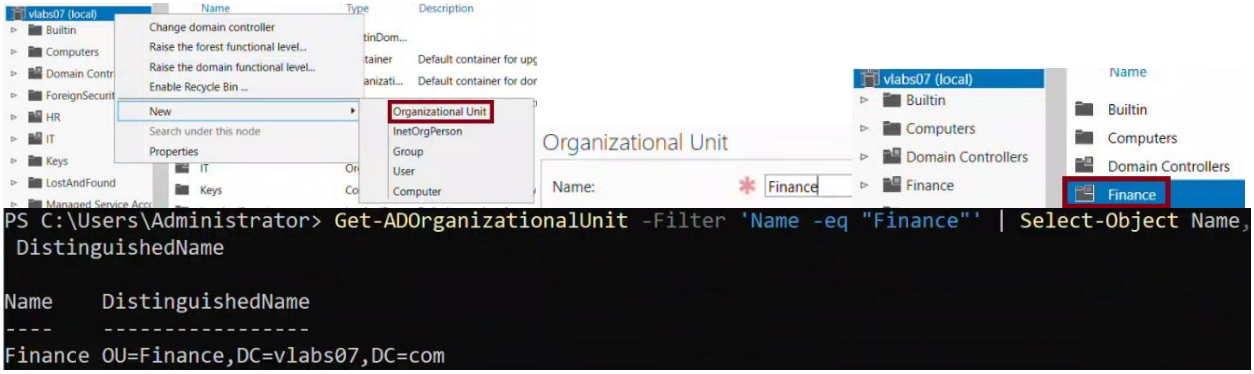
I created the Finance OU in ADAC and then verified its existence using PowerShell.

ADAC Steps:
- Navigated to vlabs07.com
- Right-clicked the domain > New > Organizational Unit
- Named the OU: Finance
- Clicked OK

PowerShell Verification:

```
Get-ADOrganizationalUnit -Filter 'Name -eq "Finance"' | Select-Object Name, DistinguishedName
```

Screenshot:



▨ Step 6: Delete the Finance OU using PowerShell and verify the deletion
System: DC107

Command:

```
Set-ADOrganizationalUnit -Identity "OU=Finance,DC=vlabs07,DC=com" -
ProtectedFromAccidentalDeletion $false
Remove-ADOrganizationalUnit -Identity "OU=Finance,DC=vlabs07,DC=com" -
Confirm:$false
```

Verification:

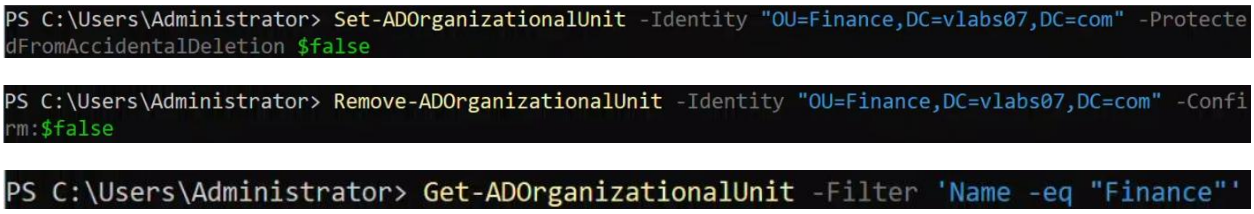
```
Get-ADOrganizationalUnit -Filter 'Name -eq "Finance"'
```

Explanation:

Before deleting the Finance OU, I disabled accidental deletion protection. Then I used Remove-ADOrganizationalUnit to delete it, followed by a verification command.

- Set-ADOrganizationalUnit unprotects the OU
- Remove-ADOrganizationalUnit deletes the OU
- If the verification command returns nothing, it means the OU was successfully deleted

Screenshot:



Task 6: Delegating Control of an OU

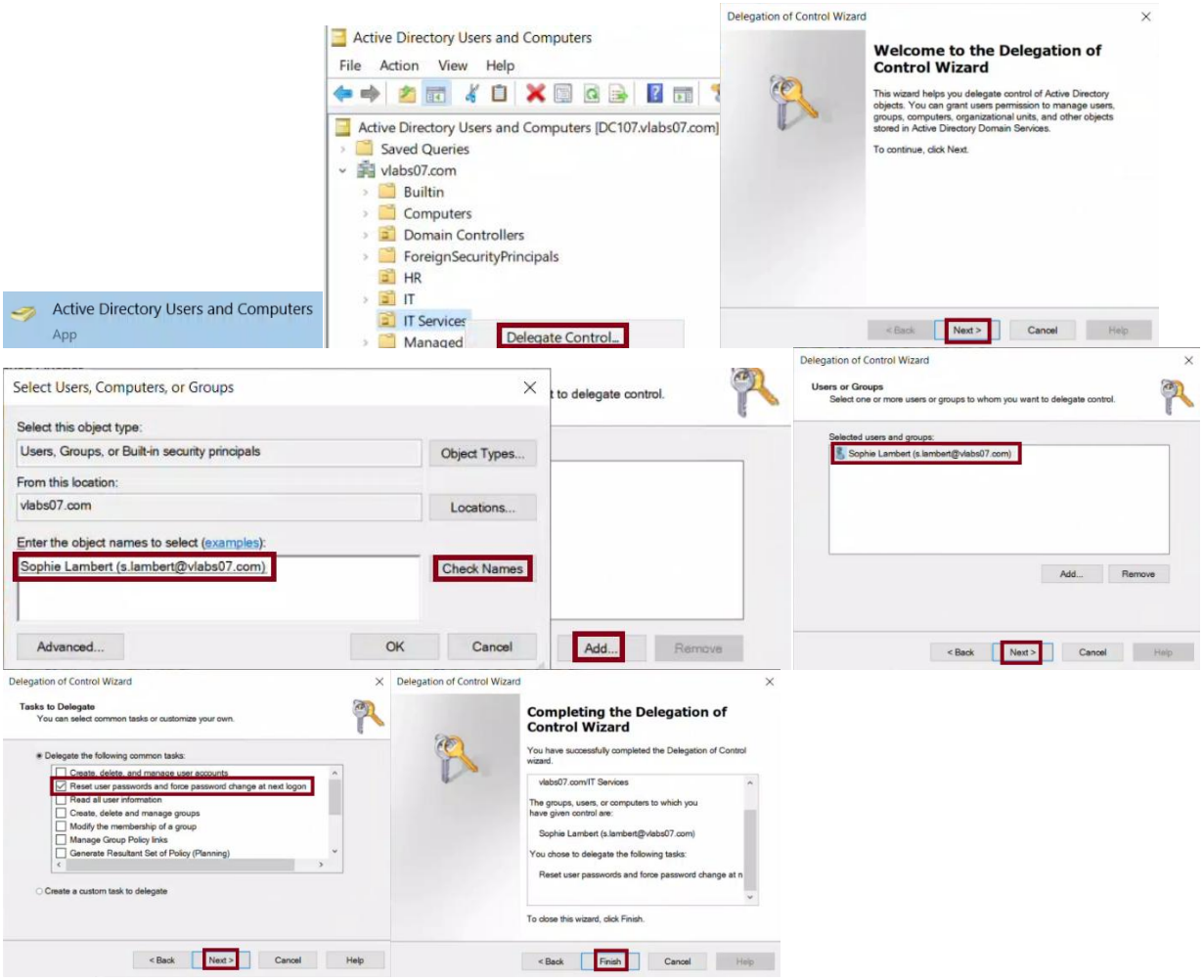
▨ Step 1: Delegate Reset Password permissions to Sophie Lambert on the IT Services OU **using** ADUC
System: DC107

Explanation:

I used Active Directory Users and Computers **(**ADUC**)** to delegate "Reset Password" permissions to Sophie Lambert on the IT Services OU.

Steps:
- Opened ADUC on DC107 **(**type `dsa.msc` **in** the Run dialog**)**
- Navigated to the IT Services OU
- Right-clicked the OU and selected "Delegate Control..."
- Clicked Next **in** the wizard
- Clicked "Add..." and selected Sophie Lambert
- Clicked Next
- Chose "Reset user passwords and force password change at next logon"
- Clicked Next **>** Finish to complete the delegation

Screenshot:



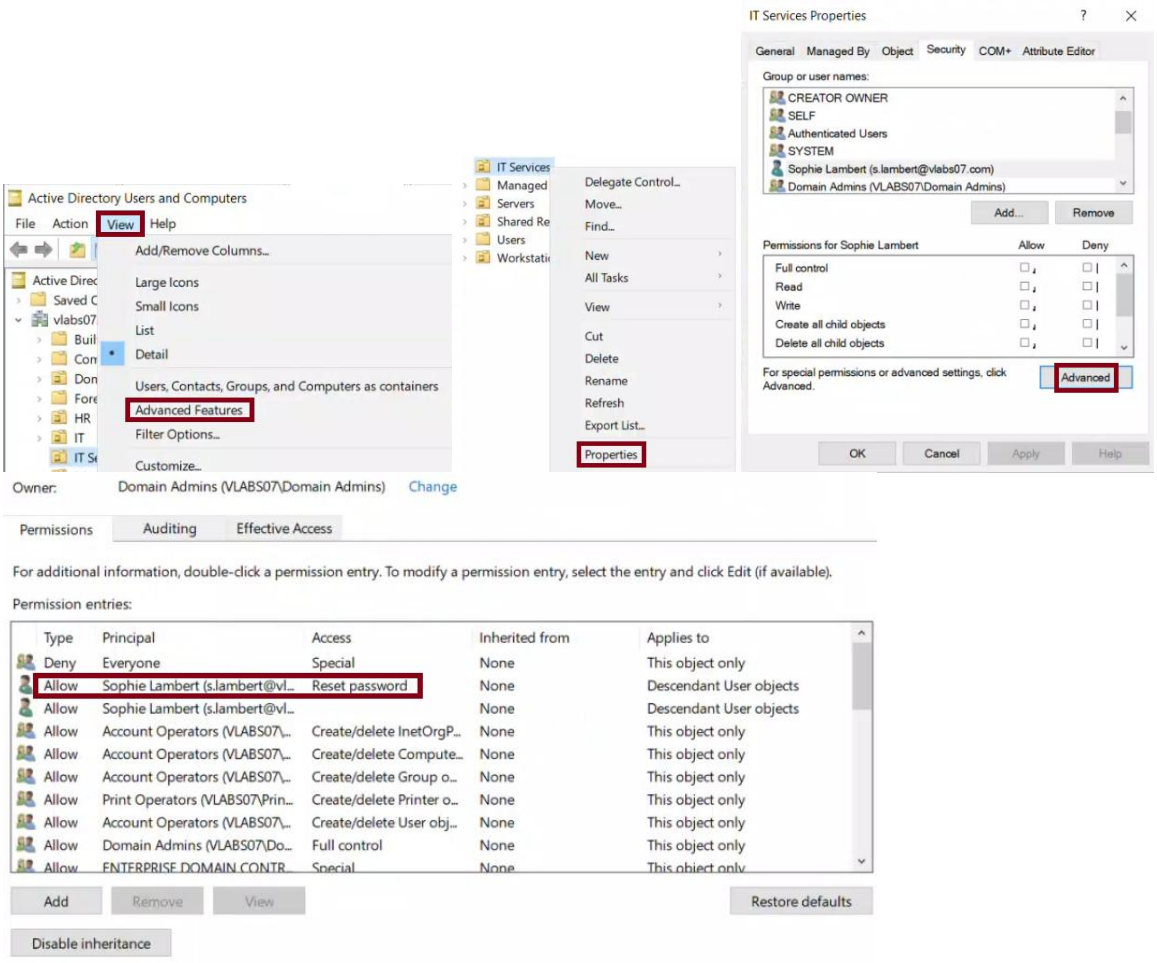
▨ Step 2: Check which users or groups have been delegated control over the IT Services OU using ADUC
System: DC107

Explanation:

I used Active Directory Users and Computers (ADUC) to check which users have delegated permissions on the IT Services OU. To view the Security tab, I had to enable Advanced Features first.

Steps:
- Opened ADUC on DC107 (type `dsa.msc` in the Run dialog)
- Clicked the "View" menu and selected "Advanced Features"
- Navigated to the IT Services OU
- Right-clicked IT Services > Properties
- Clicked on the "Security" tab
- Verified that Sophie Lambert was listed with delegated permissions
- Clicked "Advanced" to view more details about the specific rights

Screenshot:



▨ Step 3: List the delegation permissions on the IT Services OU **using** PowerShell
System: DC107

Command:

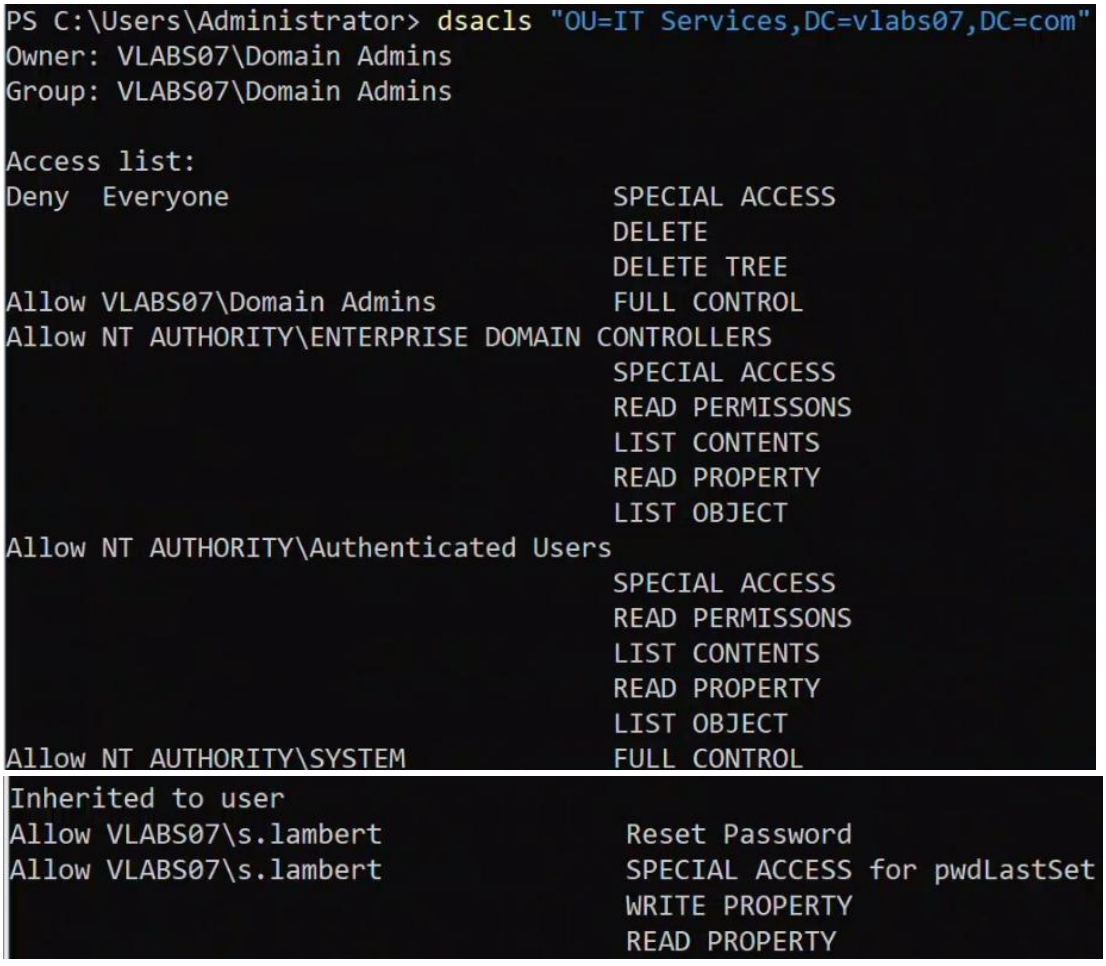
```
dsacls "OU=IT Services,DC=vlabs07,DC=com"
```

Explanation:

I used the `dsacls` command to view the delegated permissions applied to the IT Services OU.

- dsacls displays the access control list **(**ACL**)** of an AD object
- I ran it on the full Distinguished Name of the IT Services OU
- The output lists all permissions, including delegated rights **for** users like Sophie Lambert

Screenshot:



Task 7: Managing Permissions on OUs

▨ Step 1: Deny deletion of objects inside IT Services OU for Sophie Lambert using PowerShell
System: DC107

Command:

```
dsacls "OU=IT Services,DC=vlabs07,DC=com" /D "vlabs07\s.lambert:SD"
```

Validation:

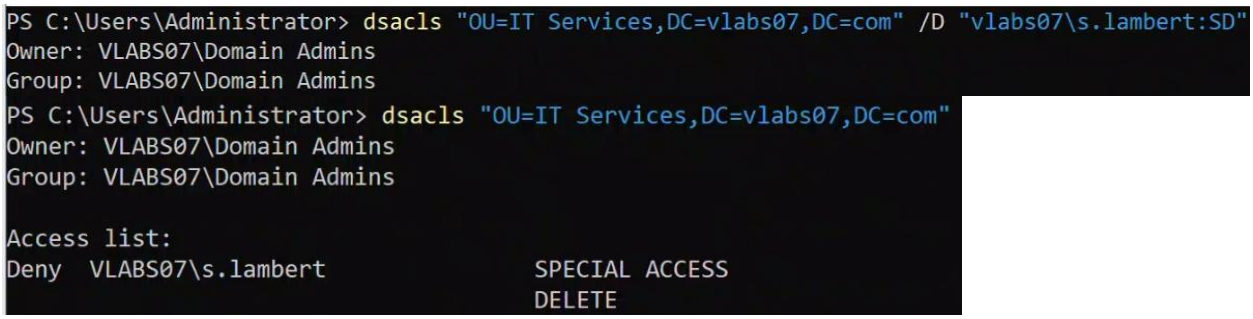
```
dsacls "OU=IT Services,DC=vlabs07,DC=com"
```

Explanation:

I used the dsacls command to deny Sophie Lambert the ability to delete the IT Services OU.

- /D sets a Deny permission
- SD stands for Standard Delete
- This prevents her from deleting the OU object itself
- The validation confirms that a deny entry was added correctly

Screenshot:



▨ Step 2: Grant Generic Read (GR) permission on the IT Services OU to Sophie Lambert using PowerShell
System: DC107

Command:
---
dsacls "OU=IT Services,DC=vlabs07,DC=com" /G "vlabs07\s.lambert:GR"

Validation:
---
dsacls "OU=IT Services,DC=vlabs07,DC=com"

Explanation:

I used dsacls to grant Sophie Lambert Generic Read (GR) permission on the IT Services OU.

- `/G` stands for "Grant" — it assigns an **Allow** permission.
- GR allows her to view the OU and its contents without modifying anything
- This matches the permissions format taught in the slides
- The validation output confirms that the appropriate permissions were applied correctly

Screenshot:

```
PS C:\Users\Administrator> dsacls "OU=IT Services,DC=vlabs07,DC=com" /G "vlabs07\s.lambert:GR"
Owner: VLABS07\Domain Admins
Group: VLABS07\Domain Admins
PS C:\Users\Administrator> dsacls "OU=IT Services,DC=vlabs07,DC=com" /G "vlabs07\s.lambert:GR"
Owner: VLABS07\Domain Admins
Group: VLABS07\Domain Admins

Access list:
Deny  VLABS07\s.lambert              SPECIAL ACCESS
                                     DELETE
Deny  Everyone                       SPECIAL ACCESS
                                     DELETE
                                     DELETE TREE
Allow VLABS07\s.lambert              SPECIAL ACCESS
                                     READ PERMISSONS
                                     LIST CONTENTS
                                     READ PROPERTY
                                     LIST OBJECT
```

▨ Step 3: Grant write permission to modify the telephoneNumber attribute for users in the HR OU using PowerShell
System: DC107

Command:

```
dsacls "OU=HR,DC=vlabs07,DC=com" /G "vlabs07\GG_HR_Admins:WP;telephoneNumber"
```

Validation:

```
dsacls "OU=HR,DC=vlabs07,DC=com"
```

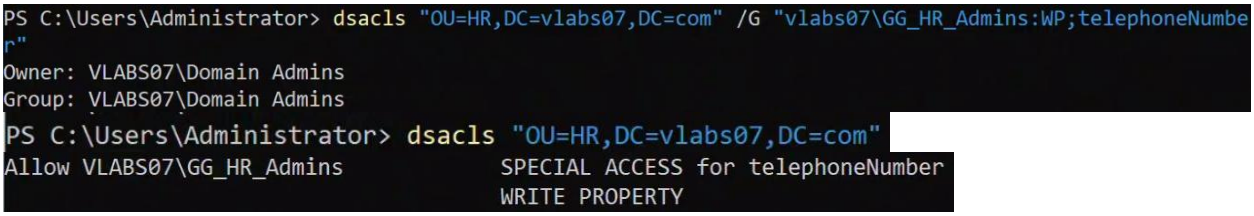
Expected output should include:
```
Allow VLABS07\GG_HR_Admins
    WRITE PROPERTY
    Property 'telephoneNumber'
```

Explanation:

I granted the GG_HR_Admins security group permission to modify the telephoneNumber attribute for user accounts in the HR OU.

- `/G` means "Grant" — it gives Allow permissions.
- `vlabs07\GG_HR_Admins` is the group that receives the permission.
- `WP` stands for **Write Property**.
- `telephoneNumber` is the attribute being delegated.
- This command matches the format shown in the course slide and applies the permission without needing object type or inheritance flags.

Screenshot:



▨ Step 4: Remove all permissions for Lucas Bernard on the HR OU using PowerShell
System: DC107

Command:

```
dsacls "OU=HR,DC=vlabs07,DC=com" /R "vlabs07\l.bernard"
```

Explanation:

I removed all delegated permissions for Lucas Bernard on the HR OU.

- /R revokes all explicit permissions for the specified user

Screenshot:

```
PS C:\Users\Administrator> dsacls "OU=HR,DC=vlabs07,DC=com" /R "vlabs07\l.bernard"
Owner: VLABS07\Domain Admins
Group: VLABS07\Domain Admins
```

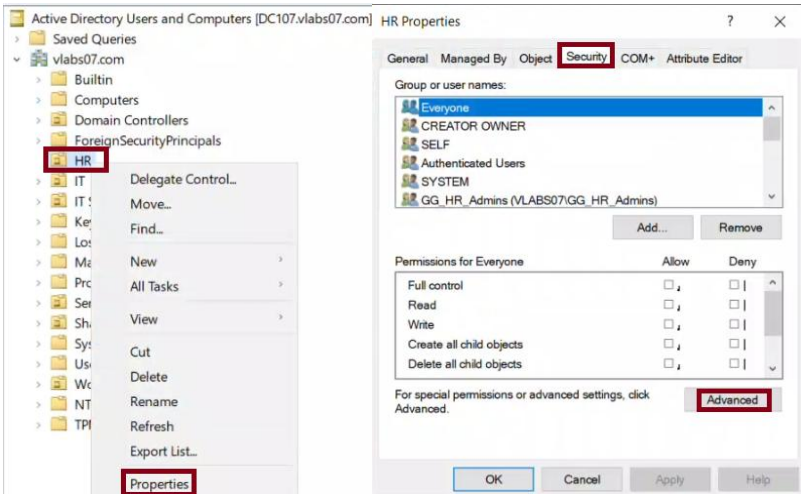
▨ Step 5: Check delegated users and permissions on the HR OU using ADUC
System: DC107

Explanation:

I opened ADUC to check which users have permissions on the HR OU.

Steps:
- Opened ADUC (dsa.msc)
- Clicked View > Enabled "Advanced Features"
- Navigated to the HR OU
- Right-clicked > Properties > Security tab
- Clicked "Advanced" to view detailed permissions
- Verified that Lucas Bernard was no longer listed

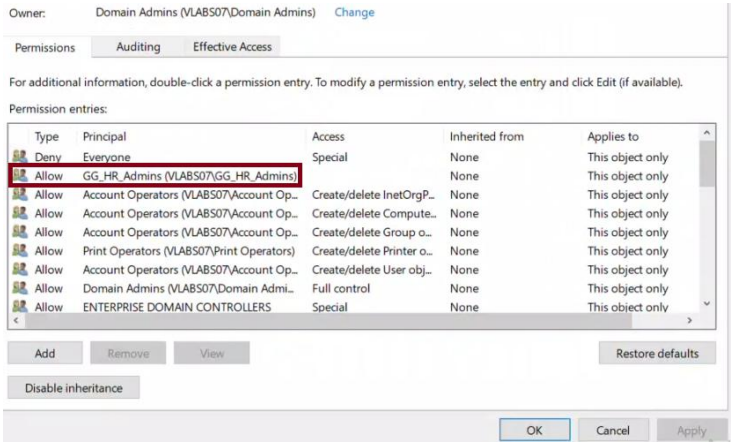
Screenshot:



Note:
Although GG_HR_Admins appears in the Security > Advanced permission list, the Access column may be blank.

This is normal when permissions are applied at the attribute level (like telephoneNumber), because the GUI doesn't display those details in this summary view.

To confirm that the permission was applied, I used the dsacls command (Step 3), which showed:
Allow VLABS07\GG_HR_Admins
    WRITE PROPERTY
    telephoneNumber



▨ Step 6: Reset permissions on the IT Services OU to default using PowerShell
System: DC107

Command:

```
dsacls "OU=IT Services,DC=vlabs07,DC=com" /resetDefaultDacl
```

Validation:

```
dsacls "OU=IT Services,DC=vlabs07,DC=com"
```

Expected output:
Only inherited permissions should remain — custom entries like Sophie Lambert should no longer be listed

Explanation:

I used dsacls with the /resetDefaultDacl option to reset the permissions on the IT Services OU to their default inherited state.

- This removes any custom Allow or Deny rules
- It ensures the OU inherits permissions from the parent container

Screenshot:

```
PS C:\Users\Administrator> dsacls "OU=IT Services,DC=vlabs07,DC=com" /resetDefaultDacl
Owner: VLABS07\Domain Admins
Group: VLABS07\Domain Admins
PS C:\Users\Administrator> dsacls "OU=IT Services,DC=vlabs07,DC=com"
```

▨ Step 7: Verify that permissions on the IT Services OU were reset **using** ADUC
System: DC107

Explanation:

I confirmed that all custom permissions on the IT Services OU were removed.

Steps:
- Opened ADUC and enabled Advanced Features
- Navigated to the IT Services OU
- Right-clicked > Properties > Security > Advanced
- Verified that only inherited permissions remained

Screenshot: