This lab is designed to help you practice managing computer objects and organizational units (OUs) in Active Directory using Active Directory Administrative Center (ADAC) and PowerShell

Lab 6 - Managing Computer Objects and Organizational Units

420-636-AB-Network Installation and Administration II

Teacher: Antoine Tohme
Student: Monica Perez Mata
Student id : 2498056

Contents

Lab 6 - Managing Computer Objects and Organizational Units

1 Lab Overview

This lab is designed to help you practice managing computer objects and organizational units (OUs) in Active Directory using Active Directory Administrative Center (ADAC) and PowerShell. You will perform tasks related to creating, deleting, modifying, and delegating control over OUs, as well as managing computer objects.

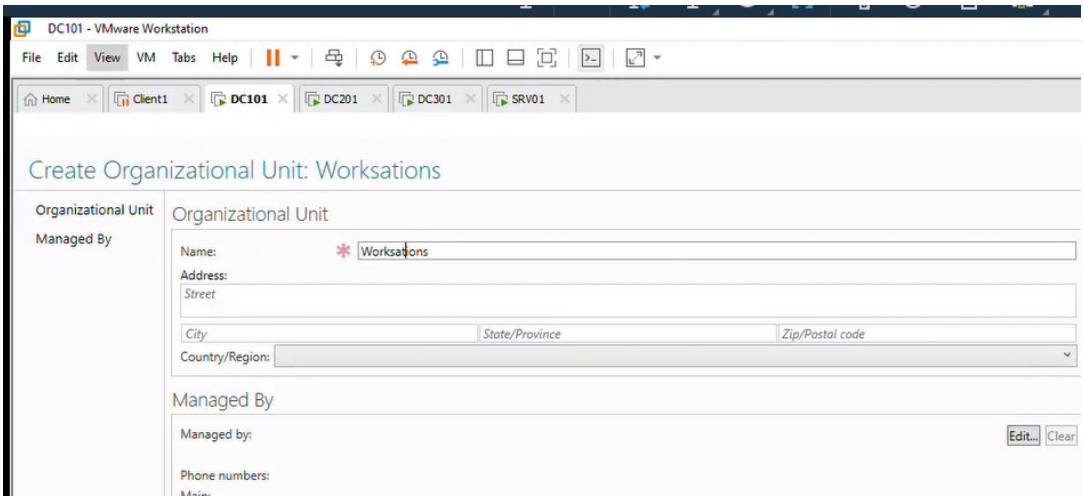
2 Lab Requirements

- Domain Name: **vlabs1.com**
- Servers: **DC101 (Windows Server 2022, Primary DC for vlabs1.com)**
- Client Machine: **Windows 11 Client1(domain-joined)**

2.1 Tasks

2.1.1 Task 1: Managing Computer Objects

1. Create **Workstations** OU using **ADAC.**
2. Log in to **DC101** (Primary DC for `vlabs1.com`).
3. Open **Active Directory Administrative Center (ADAC)**.
4. Navigate to **Tree View** → **vlabs1 (Local)** → Right-click domain → **New** → **Organizational Unit**.
5. Name: `Workstations` → Click **OK**.

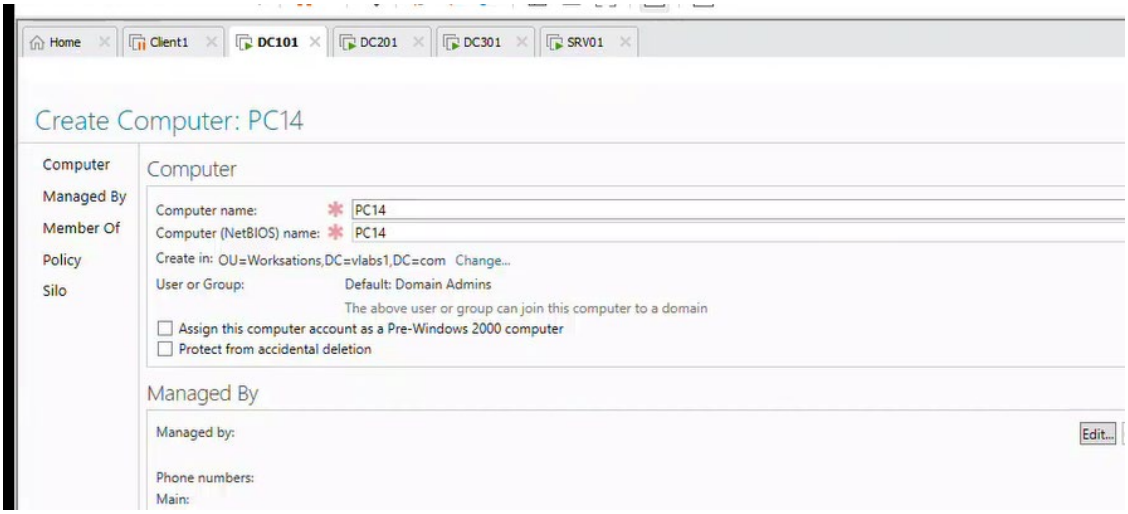


6. Create a computer object **PC14** in the **Workstations** OU using **ADAC**.

a) In ADAC, navigate to **Workstations** OU.

b) Right-click → **New** → **Computer**.

c) Set **Computer Name**: PC14 → Ensure **Workstations** OU is selected → Click **OK**.



7. Verify the creation of this computer object using **PowerShell.**

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

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


DistinguishedName : CN=PC14,OU=Worksations,DC=vlabs1,DC=com
DNSHostName       :
Enabled           : True
Name              : PC14
ObjectClass       : computer
ObjectGUID        : 7e5ba516-4c61-4598-afaf-ead800cb16ea
SamAccountName    : PC14$
SID               : S-1-5-21-1268601764-4050707287-4025116504-1122
UserPrincipalName :


PS C:\Users\Administrator>
```

8. Rename **PC14** to **PC14-Updated** using **PowerSehll**

Rename-ADObject -Identity $(Get-ADComputer -Identity PC14).DistinguishedName -NewName "PC14-Updated"

```
PS C:\Users\Administrator> Rename-ADObject -Identity $(Get-ADComputer -Identity PC14).DistinguishedName -NewName "PC14-Updated"
```

Get-ADComputer -Filter "Name -eq 'PC14-Updated'"

```
PS C:\Users\Administrator> Get-ADComputer -Filter "Name -eq 'PC14-Updated'"


DistinguishedName : CN=PC14-Updated,OU=Worksations,DC=vlabs1,DC=com
DNSHostName       :
Enabled           : True
Name              : PC14-Updated
ObjectClass       : computer
ObjectGUID        : 7e5ba516-4c61-4598-afaf-ead800cb16ea
SamAccountName    : PC14$
SID               : S-1-5-21-1268601764-4050707287-4025116504-1122
UserPrincipalName :
```

Note - Rename-ADObject changes the display name (Name/CN), not sAMAccountName = PC14$

9. Remove **PC14** using **PowerShell.**

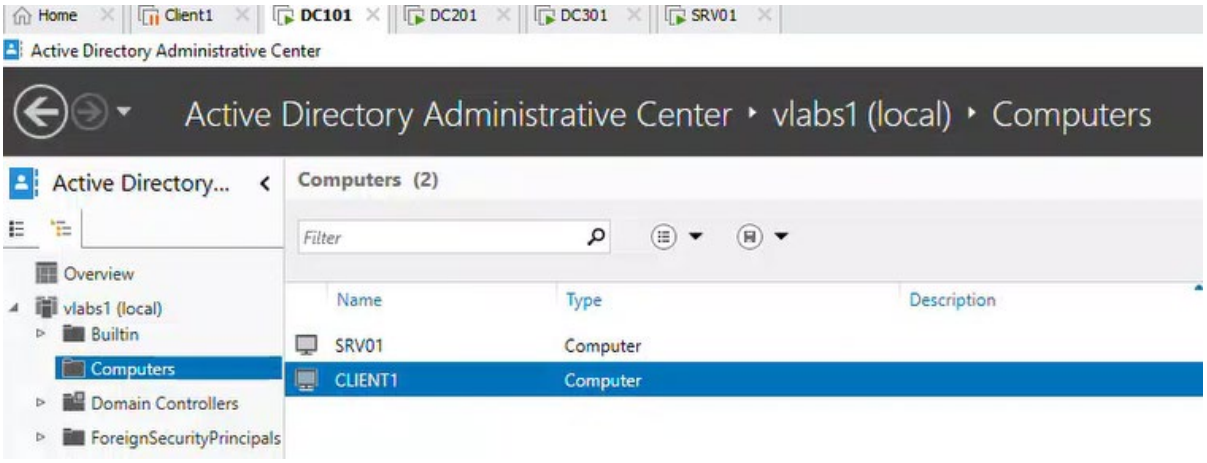
Remove-ADComputer -Identity "PC14" -Confirm:$false

```
PS C:\Users\Administrator> Remove-ADComputer -Identity "PC14" -Confirm:$false
PS C:\Users\Administrator>
```

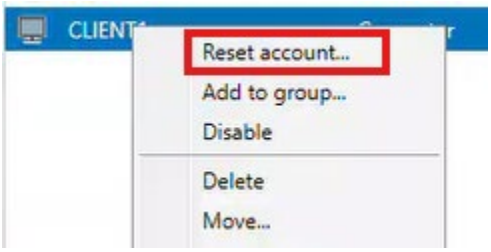
10. Reset the **secure channel** for **Client1** using **ADAC.**

a) In ADAC, navigate to the **Computers** container (or the OU where `Client1` resides).

b)



c) Right-click **Client1** → **Reset Account** → Confirm.



11. Test the **secure channel** for **Client1** using **PowerShell** ***(Do this command from the Windows 11 Client).***

a) Log in to **Client1** (Windows 11) as an administrator.

b) Open PowerShell and run:

```
Test-ComputerSecureChannel
```

```
PS C:\Users\administrator> Test-ComputerSecureChannel
True
PS C:\Users\administrator>
```

```
Test-ComputerSecureChannel -Server DC101.vlabs1.com -Verbose
```

```
PS C:\Users\administrator> Test-ComputerSecureChannel -Server DC101.vlabs1.com -Verbose
VERBOSE: Performing the operation "Test-ComputerSecureChannel" on target "CLIENT1".
True
VERBOSE: The secure channel between the local computer and the domain vlabs1.com is in good condition.
PS C:\Users\administrator>
PS C:\Users\administrator>
```

2.1.2 Task 2: Redirecting the Computers Container

1. Verify the default **Computers container** location using **PowerShell.**

```
Get-ADDomain | Select-Object ComputersContainer
```

```
PS C:\Users\Administrator> # Check the current default Computers container location
PS C:\Users\Administrator> Get-ADDomain | Select-Object ComputersContainer

ComputersContainer
------------------
CN=Computers,DC=vlabs1,DC=com


PS C:\Users\Administrator>
```

2. Redirect the default location of the **Computers container** to the **Workstations OU** using **PowerShell.**

```
redircmp "OU=Worksations,DC=vlabs1,DC=com"
```

```
PS C:\Users\Administrator> redircmp "OU=Worksations,DC=vlabs1,DC=com"
Redirection was successful.
PS C:\Users\Administrator>
```

3. Verify that the redirection was applied using **PowerShell.**

```
Get-ADDomain | Select-Object ComputersContainer
```

```
PS C:\Users\Administrator> Get-ADDomain | Select-Object ComputersContainer

ComputersContainer
------------------
OU=Worksations,DC=vlabs1,DC=com


PS C:\Users\Administrator>
```

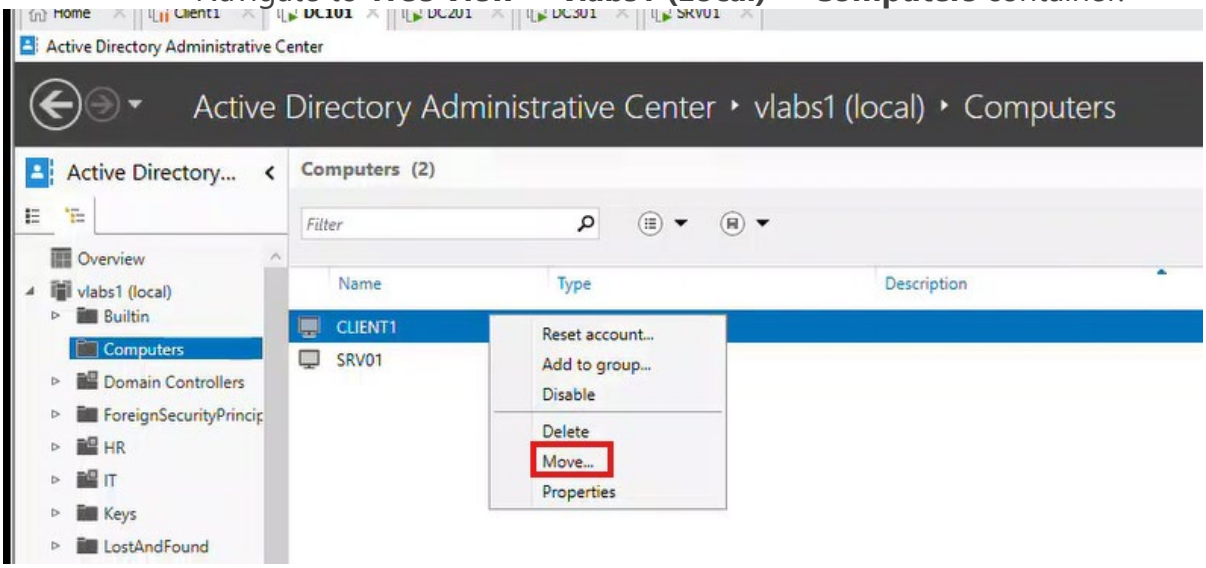
2.1.3 Task 3: Moving Computer Objects

1. Move **Client1** from **Computers container** to **Workstations OU** using **ADAC.**

a) **Open ADAC on DC101** (Domain Controller):

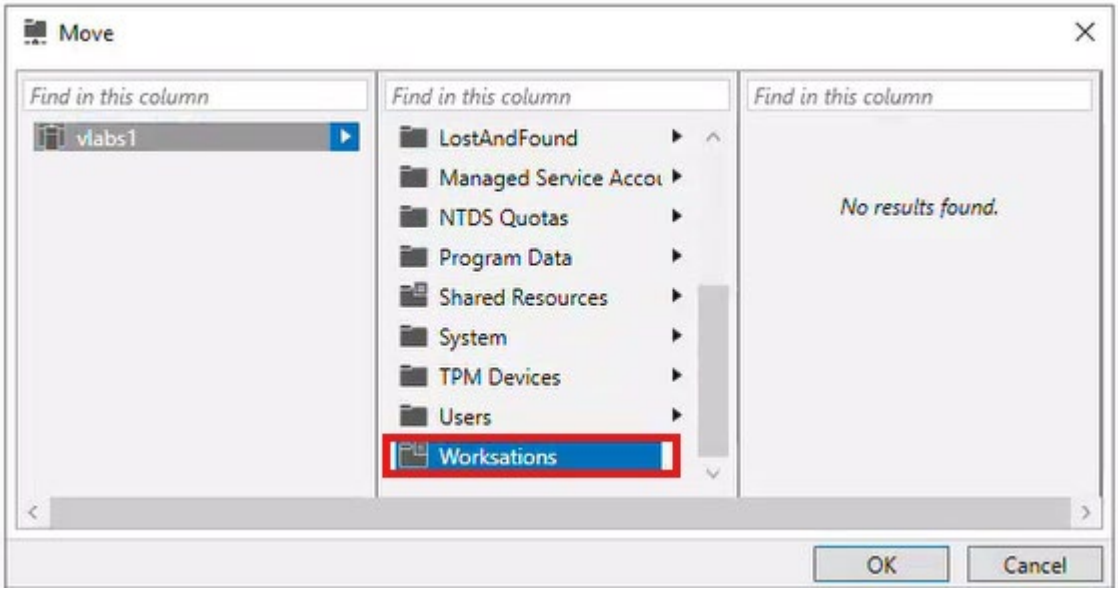
- Launch **Active Directory Administrative Center**.

- Navigate to **Tree View** → **vlabs1 (Local)** → **Computers** container.



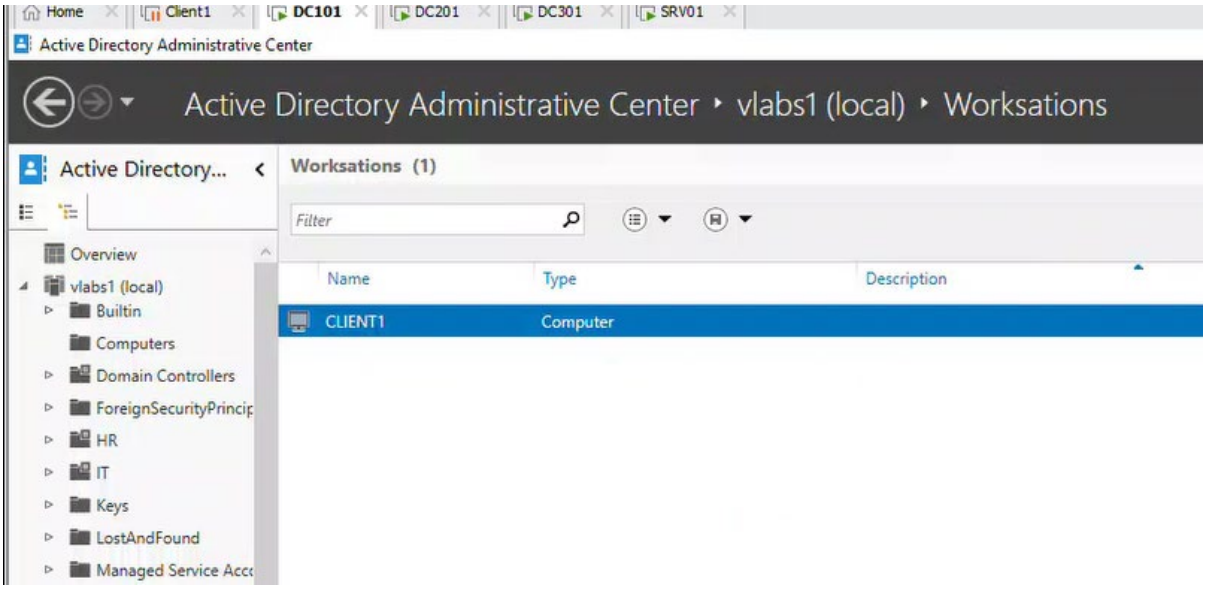
b) **Move Client1**:

- Right-click **Client1** → **Move** → Select the **Workstations OU** → Click **OK**.



c) **Verify**:

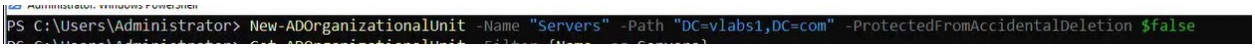
- Navigate to **Workstations OU** → Confirm `Client1` is listed.



2. Create an OU named **Servers** using **PowerShell.**

Run PowerShell as Administrator:

New-ADOrganizationalUnit -Name "Servers" -Path "DC=vlabs1,DC=com" -ProtectedFromAccidentalDeletion $false



Verify:

Get-ADOrganizationalUnit -Filter {Name -eq “Servers”}

```
PS C:\Users\Administrator> Get-ADOrganizationalUnit -Filter {Name -eq "Servers"}


City                    :
Country                 :
DistinguishedName       : OU=Servers,DC=vlabs1,DC=com
LinkedGroupPolicyObjects : {}
ManagedBy               :
Name                    : Servers
ObjectClass             : organizationalUnit
ObjectGUID              : 5733a995-2887-4dce-ae2f-9dd219a95dd3
PostalCode              :
State                   :
StreetAddress           :



PS C:\Users\Administrator>
```

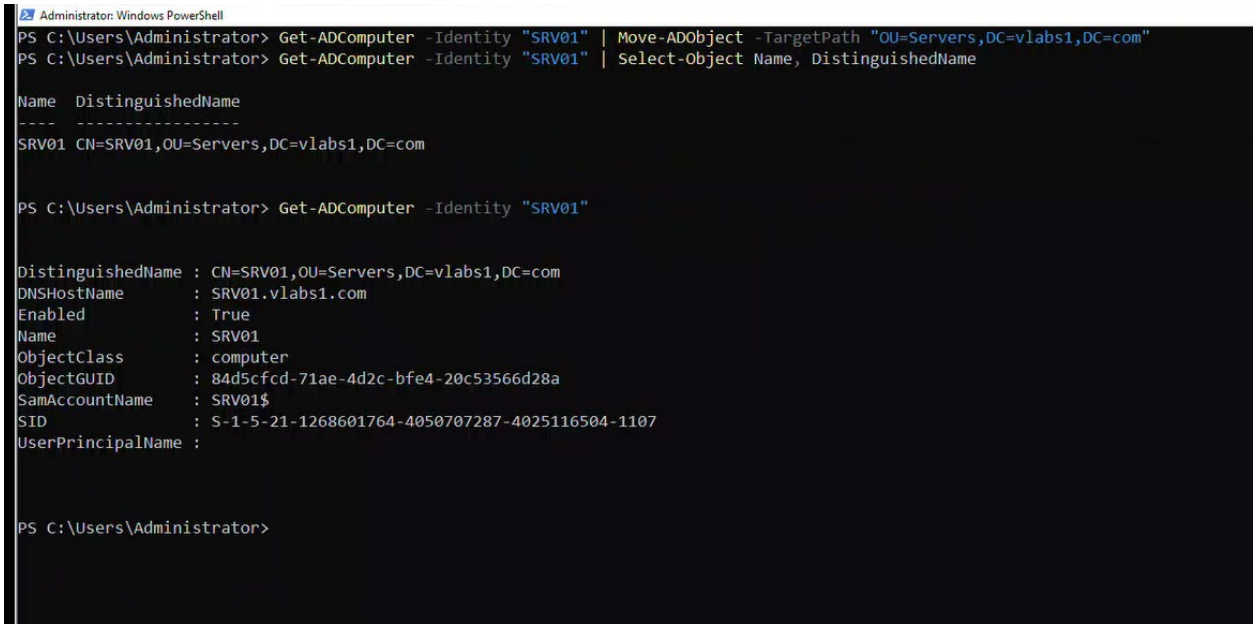
3. Move **SRV01** to the **Servers** OU using **PowerShell.**

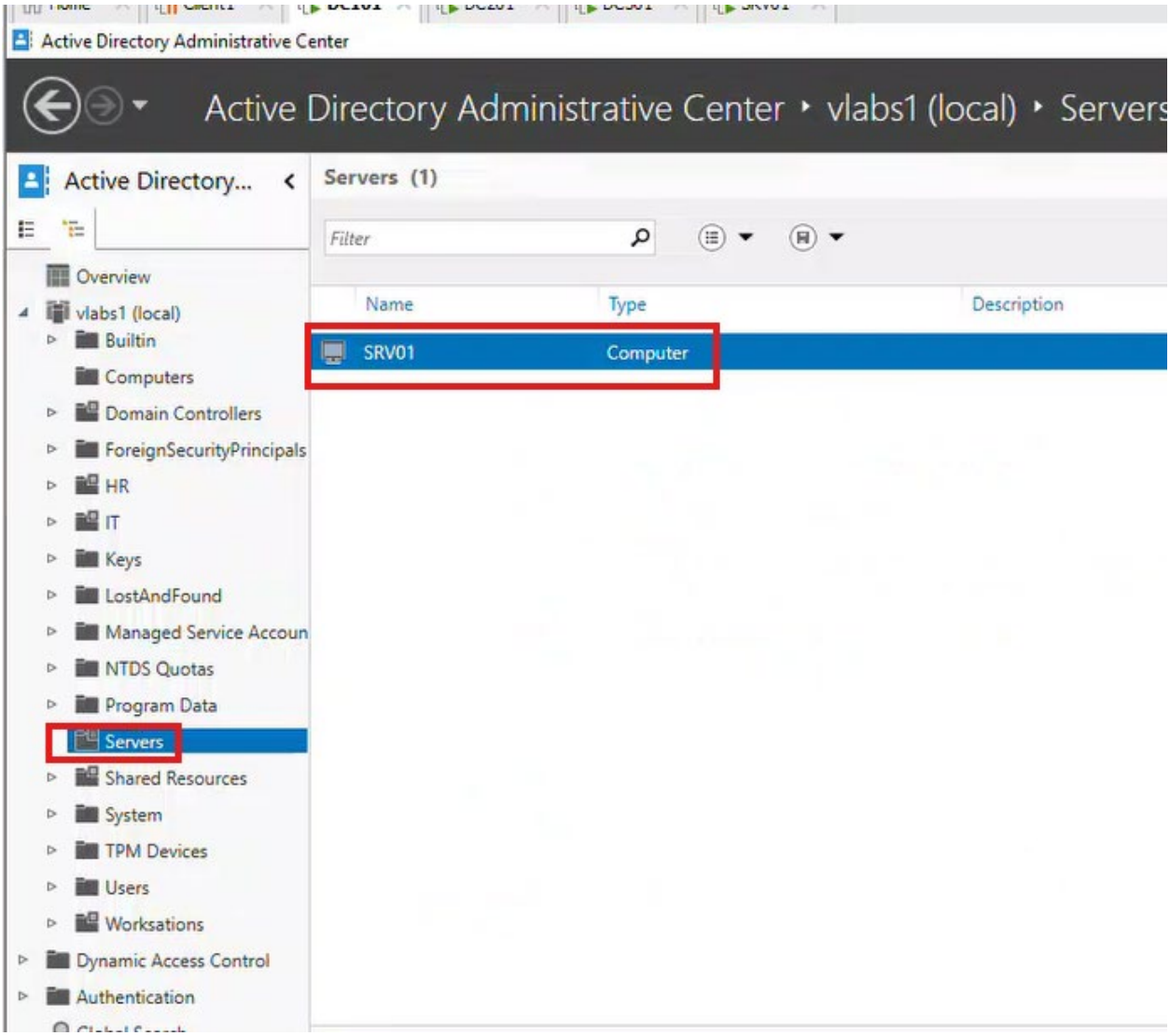
   Move SRV01:

   Get-ADComputer -Identity "SRV01" | Move-ADObject -TargetPath "OU=Servers,DC=vlabs1,DC=com"

   Verify:

   Get-ADComputer -Identity "SRV01" | Select-Object Name, DistinguishedName





2.1.4 Task 4: Changing the Default Quota for Creating Computer Objects

1. Change the **Default Quota for creating Computer Objects** to **0** using **PowerShell.**

 # Set the quota to 0

Set-ADDomain -Identity "vlabs1.com" -Replace @{"ms-DS-MachineAccountQuota"=0}

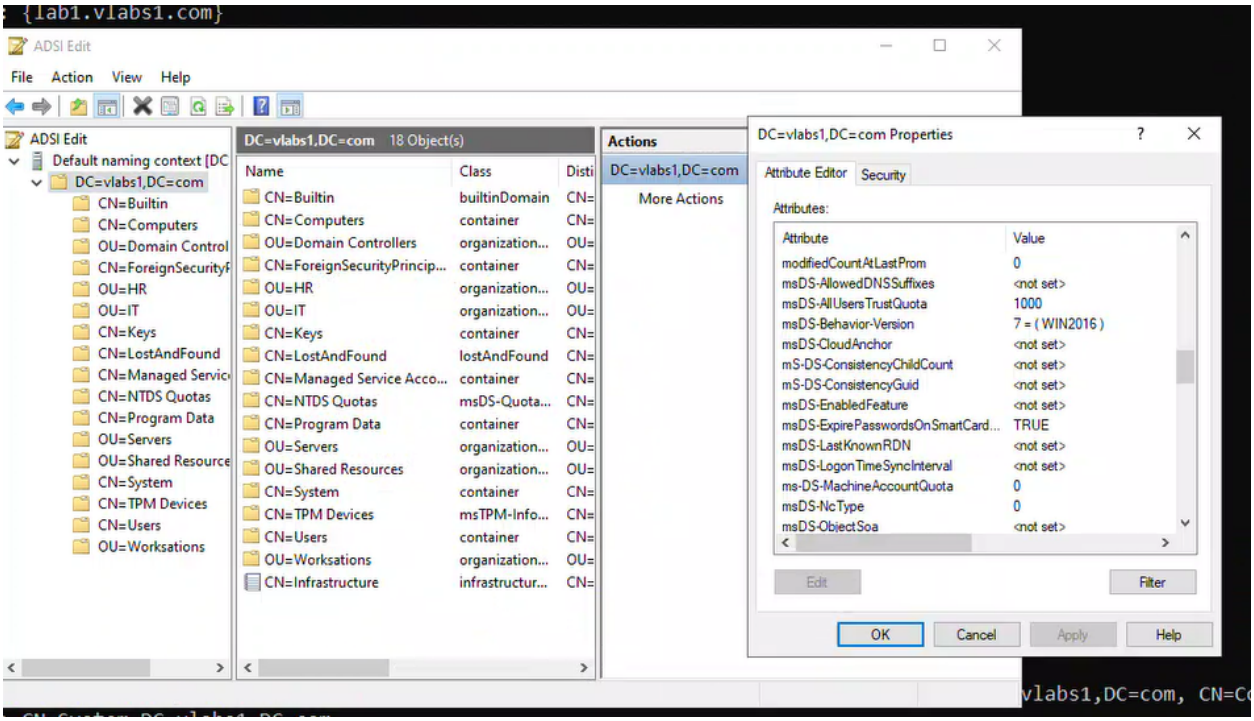
```
PS C:\Users\Administrator> Set-ADDomain -Identity "vlabs1.com" -Replace @{"ms-DS-MachineAccountQuota"=0}
```

Verify

Get-ADObject -Identity "DC=vlabs1,DC=com" -Properties ms-DS-MachineAccountQuota | Select-Object ms-DS-MachineAccountQuota

When this attribute is changed to 0, only user accounts with privileges to add computer objects to the domain are explicitly allowed to join devices to the domain. By default, only members of the Domain Admins and Account Operators groups have these privileges, but these privileges can be specifically delegated.

2. Verify the change using **ADSI Edit.**



2.1.5 Task 5: Managing Organizational Units (OUs)

1. Create **IT Department** OU under **vlabs.com** using **ADAC:**

a) Open Active Directory Administrative Center (ADAC) on DC101.
b) Navigate to Tree View → vlabs1 (Local) → Right-click domain → New → Organizational Unit.
c) Name: IT Department → Click OK.

2. Verify that **IT Department** OU has been created using **PowerShell.**

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

Get-ADOrganizationalUnit -Filter "Name -eq 'IT Department'"

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

Name          DistinguishedName
----          -----------------
IT Department OU=IT Department,DC=vlabs1,DC=com


PS C:\Users\Administrator> Get-ADOrganizationalUnit -Filter "Name -eq 'IT Department'"


City                    :
Country                 :
DistinguishedName       : OU=IT Department,DC=vlabs1,DC=com
LinkedGroupPolicyObjects : {}
ManagedBy               :
Name                    : IT Department
ObjectClass             : organizationalUnit
ObjectGUID              : 5da7743d-394e-4070-ae74-29aeb9854aaa
PostalCode              :
State                   :
StreetAddress           :
```

3. Add a description: **"Handles IT operations and security"** to the **IT Department** OU using **PowerShell.**

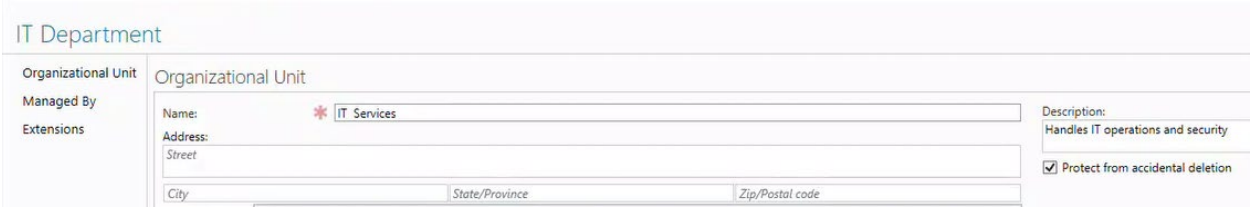
Set-ADOrganizationalUnit -Identity "OU=IT Department,DC=vlabs1,DC=com" -Description "Handles IT operations and security"

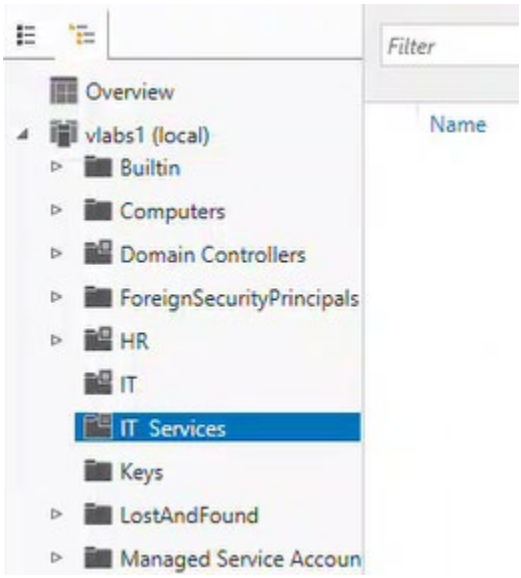
```
PS C:\Users\Administrator> Set-ADOrganizationalUnit -Identity "OU=IT Department,DC=vlabs1,DC=com" -Description "Handles IT operations and security"
PS C:\Users\Administrator>
```

```
PS C:\Users\Administrator> Get-ADOrganizationalUnit -Identity "OU=IT Department,DC=vlabs1,DC=com" -Properties Description | Select-Object Name, Description

Name          Description
----          -----------
IT Department Handles IT operations and security
```

4. Rename **IT Department** to **IT Services** using **ADAC.**

   a) In ADAC, navigate to IT Department OU → Right-click → Properties.
   b) Enter new name: IT Services → Press Enter.





5. Create **Finance** OU using **ADAC** and verify the creation using **PowerShell.**

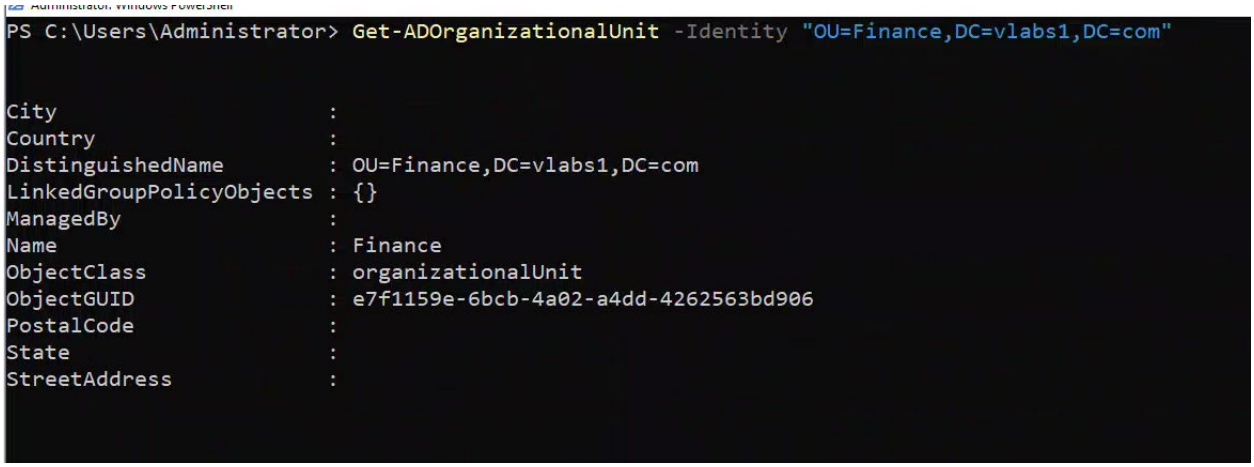
Get-ADOrganizationalUnit -Identity "OU=Finance,DC=vlabs1,DC=com"

```
PS C:\Users\Administrator> Get-ADOrganizationalUnit -Identity "OU=Finance,DC=vlabs1,DC=com"


City                        :
Country                     :
DistinguishedName           : OU=Finance,DC=vlabs1,DC=com
LinkedGroupPolicyObjects    : {}
ManagedBy                   :
Name                        : Finance
ObjectClass                 : organizationalUnit
ObjectGUID                  : e7f1159e-6bcb-4a02-a4dd-4262563bd906
PostalCode                  :
State                       :
StreetAddress               :



PS C:\Users\Administrator>
```

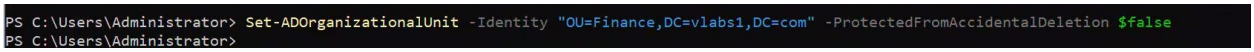
6. Delete the **Finance** OU using **PowerShell** and verify the deletion in

PowerShell.

Get-ADOrganizationalUnit -Identity "OU=Finance,DC=vlabs1,DC=com"

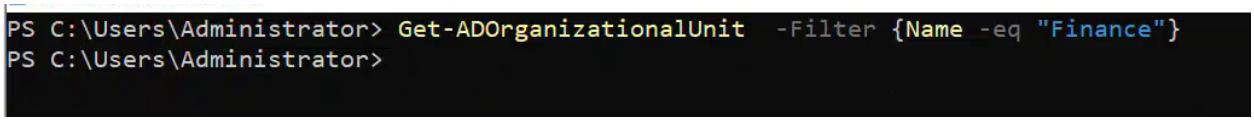


Set-ADOrganizationalUnit-Identity "OU=Finance,DC=vlabs1,DC=com" -ProtectedFromAccidentalDeletion$false



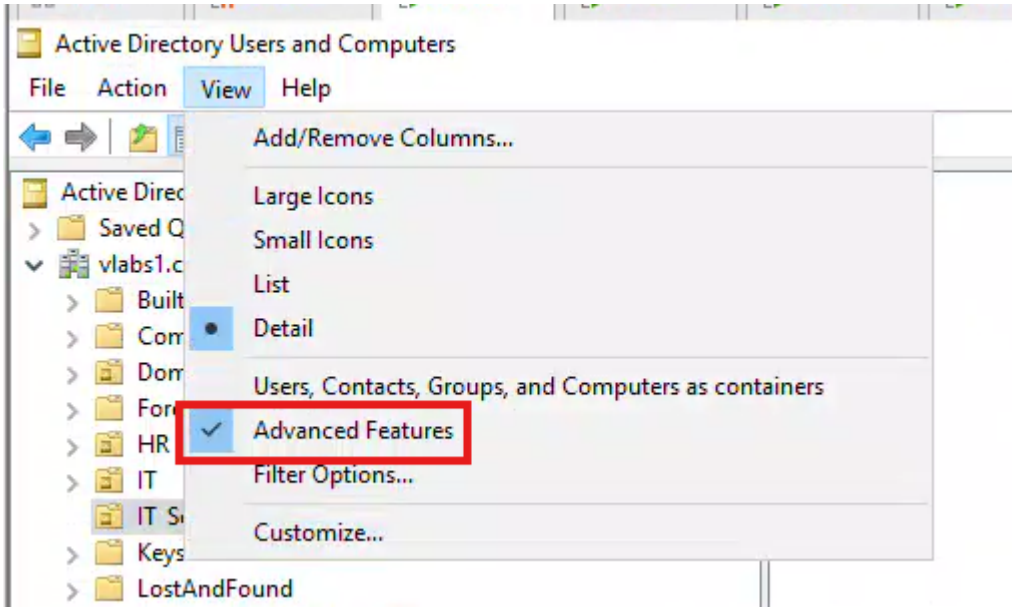
Remove-ADOrganizationalUnit -Identity "OU=Finance,DC=vlabs1,DC=com" -Confirm:$false

Get-ADOrganizationalUnit -Filter {Name -eq "Finance"}



2.1.6 Task 6: Delegating Control of an OU

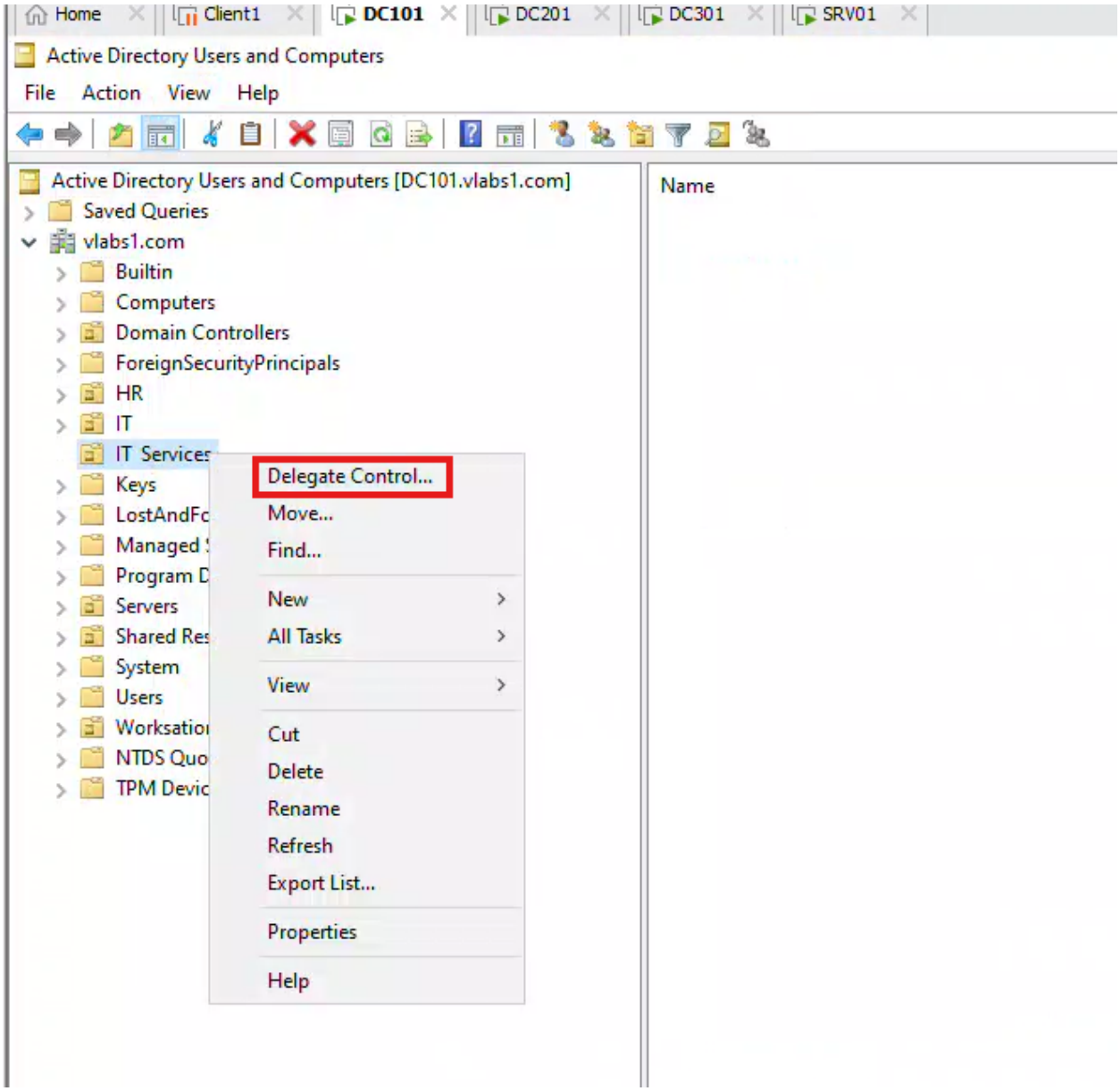
1. Use the **Delegation of Control Wizard** in **AD Users and Computers** to delegate **Reset Password permissions** to **Sophie Lambert** on the **IT Services** OU.
    a) **Open Active Directory Users and Computers (ADUC):**
        - On **DC101**.
    b) **Enable Advanced Features (if hidden):**
        - Go to **View** → **Advanced Features**.

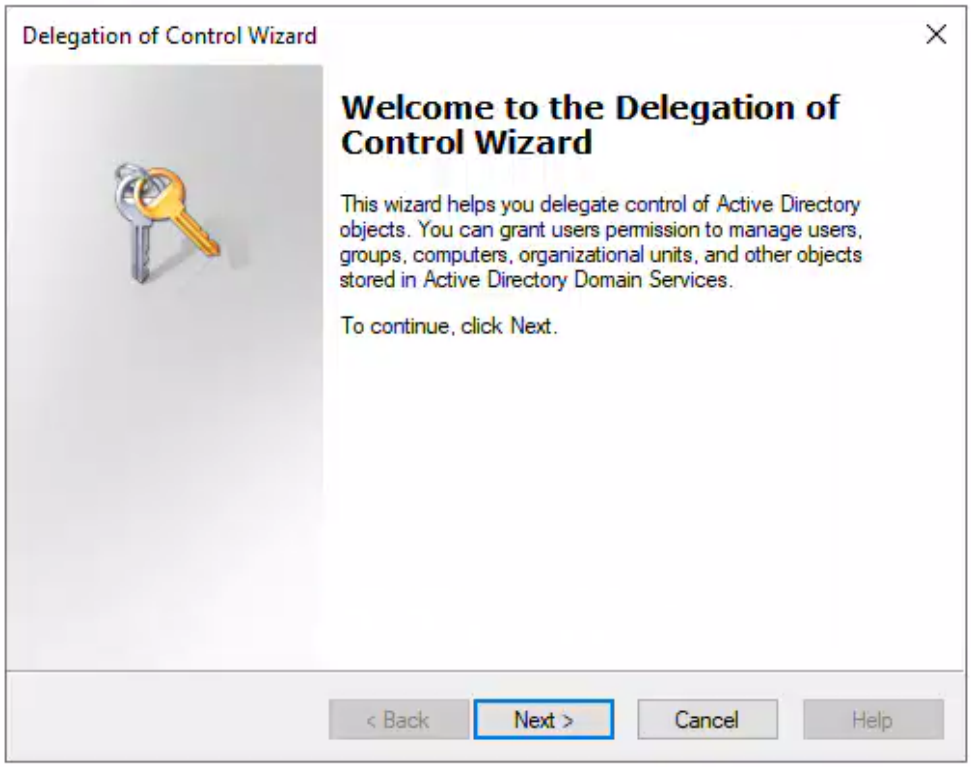


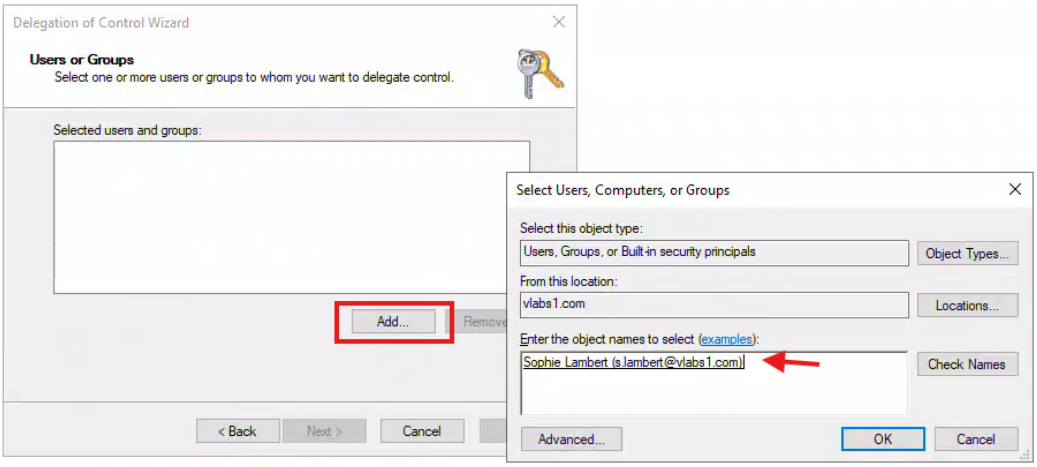
c) **Delegate Control:**

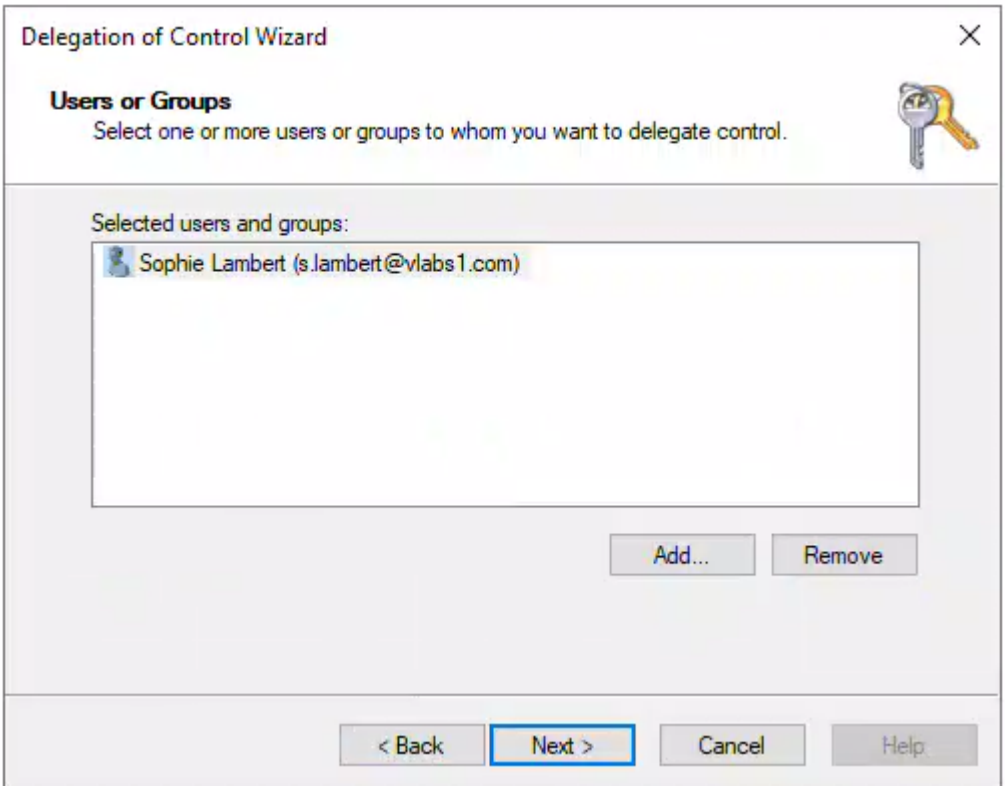
- Right-click the **IT Services OU** → **Delegate Control** → **Next**.
- Click **Add** → Enter `Sophie Lambert` → **Check Names** → **OK** → **Next**.
- Select **Delegate the following common task**
    - Reset Passwords
    - Create delete and manage user accounts
    - Modify group membership

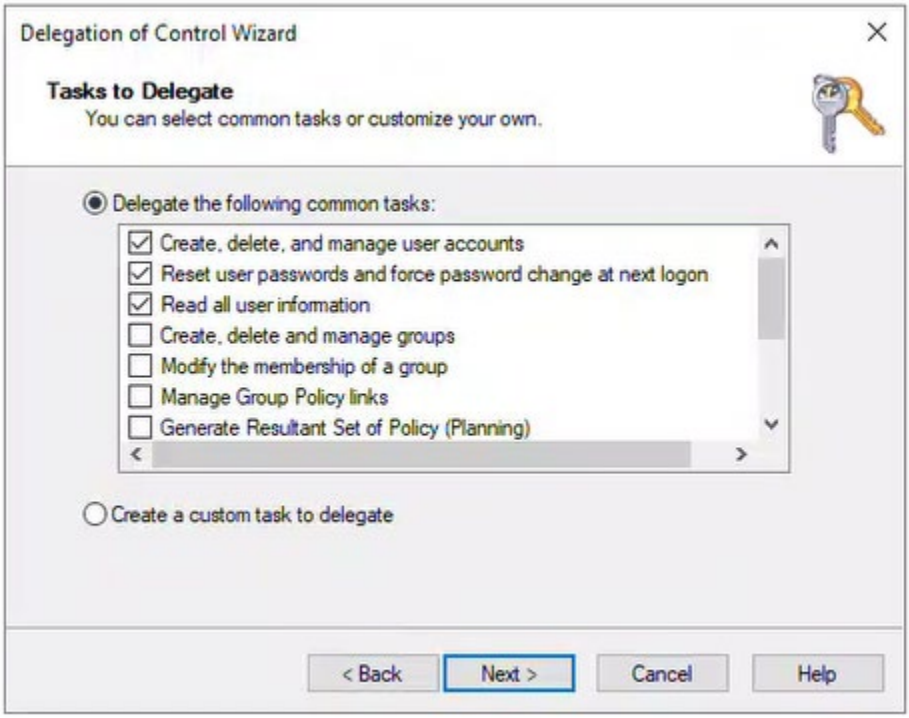
Next → **Finish**.


Home
Client1
DC101
DC201
DC301
SRV01
Active Directory Users and Computers
File
Action
View
Help
Active Directory Users and Computers [DC101.vlabs1.com]
Saved Queries
vlabs1.com
Builtin
Computers
Domain Controllers
ForeignSecurityPrincipals
HR
IT
IT Services
Keys
Managed
Program D
Servers
Shared Res
System
Users
NTDS Quo
TPM Devic
Delegate Control...
Move...
Find...
New
All Tasks
View
Cut
Delete
Rename
Refresh
Export List...
Properties
Help
Name

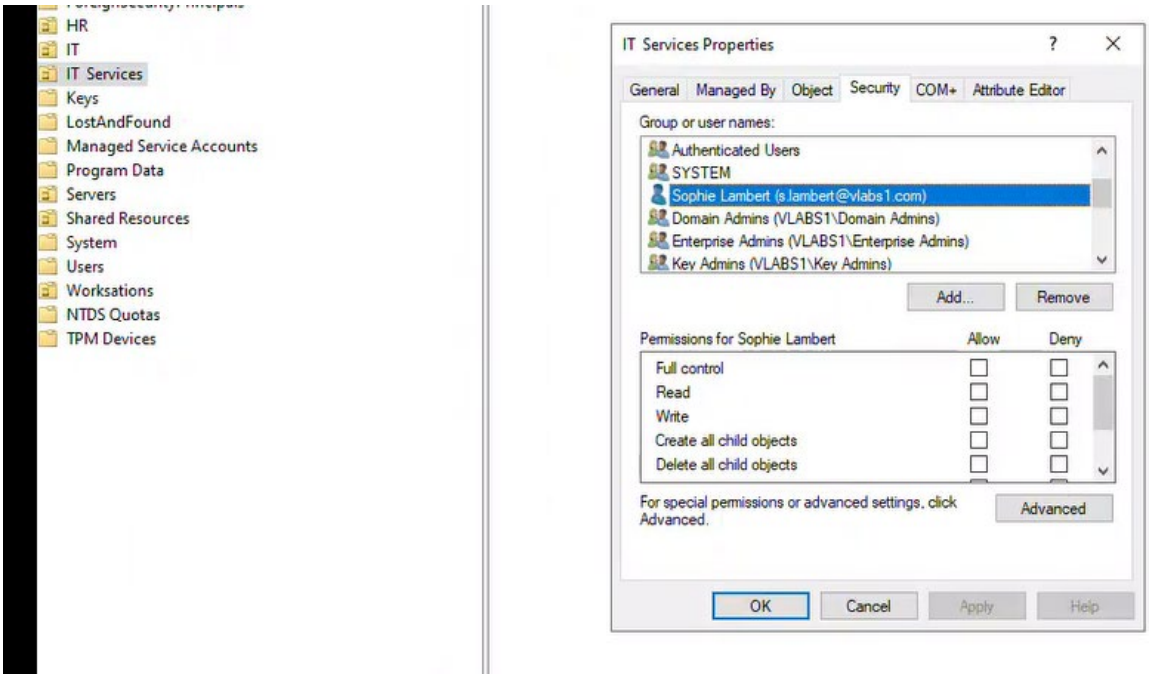

Delegation of Control Wizard
Welcome to the Delegation of Control Wizard
This wizard helps you delegate control of Active Directory objects. You can grant users permission to manage users, groups, computers, organizational units, and other objects stored in Active Directory Domain Services.
To continue, click Next.
< Back
Next >
Cancel
Help

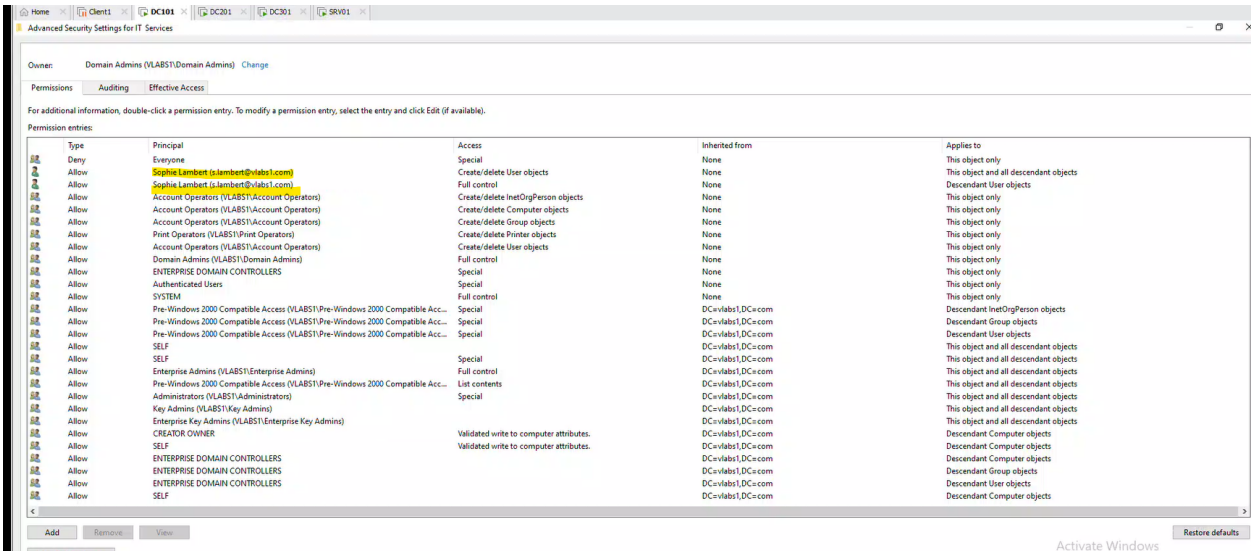

Delegation of Control Wizard
Users or Groups
Select one or more users or groups to whom you want to delegate control.
Selected users and groups:
Add...
Remove
< Back
Next >
Cancel
Select Users, Computers, or Groups
Select this object type:
Users, Groups, or Built-in security principals
Object Types...
From this location:
vlabs1.com
Locations...
Enter the object names to select (examples):
Sophie Lambert (s.lambert@vlabs1.com)
Check Names
Advanced...
OK
Cancel


Delegation of Control Wizard
Users or Groups
Select one or more users or groups to whom you want to delegate control.
Selected users and groups:
Sophie Lambert (s.lambert@vlabs1.com)
Add...
Remove
< Back
Next >
Cancel
Help

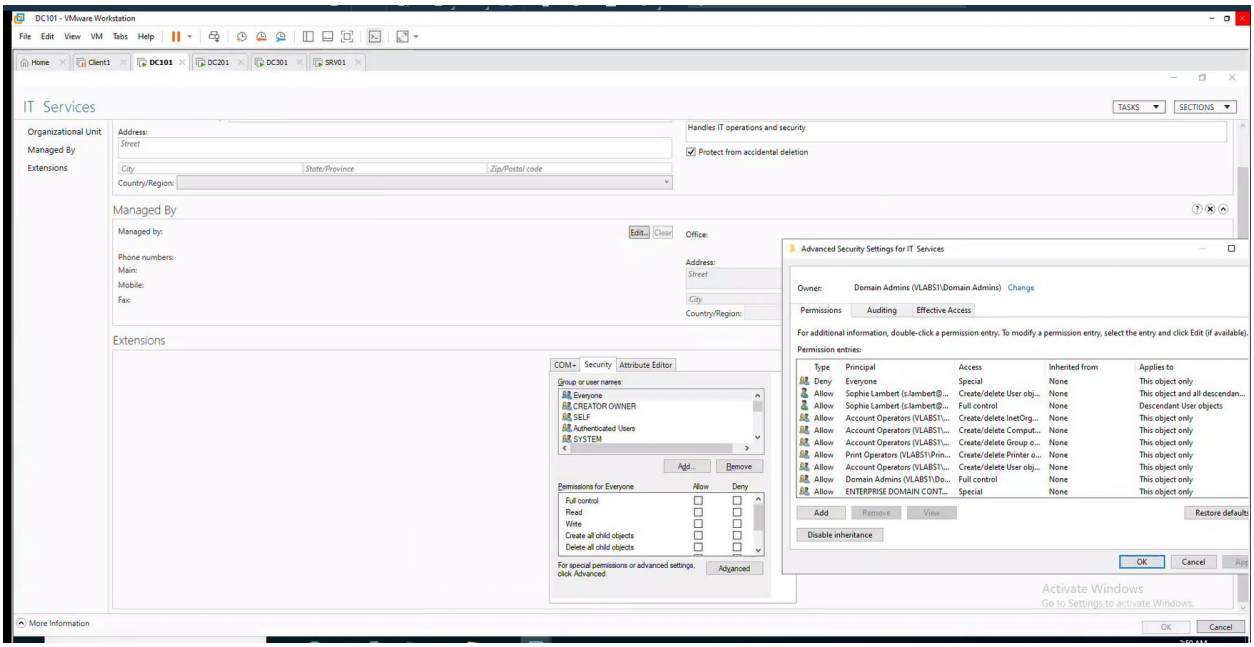

Delegation of Control Wizard
Tasks to Delegate
You can select common tasks or customize your own.
Delegate the following common tasks:
Create, delete, and manage user accounts
Reset user passwords and force password change at next logon
Read all user information
Create, delete and manage groups
Modify the membership of a group
Manage Group Policy links
Generate Resultant Set of Policy (Planning)
Create a custom task to delegate
< Back
Next >
Cancel
Help

2. Check which users or groups have been delegated control over the **IT Services** OU using **AD Users and Computers**
    a) Right-click the **IT Services OU** → **Properties** → **Security** tab.
    b) Click **Advanced** → Find **Sophie Lambert** in the list.
    c) Check her permissions:
        a. **Type:** Allow
        b. **Applies to:** Descendant User objects
        c. **Permissions:** Reset password


HR
IT
IT Services
Keys
LostAndFound
Managed Service Accounts
Program Data
Servers
Shared Resources
System
Users
Worksations
NTDS Quotas
TPM Devices
IT Services Properties
General
Managed By
Object
Security
COM+
Attribute Editor
Group or user names:
Authenticated Users
SYSTEM
Sophie Lambert (s.lambert@vlabs1.com)
Domain Admins (VLABS1\Domain Admins)
Enterprise Admins (VLABS1\Enterprise Admins)
Key Admins (VLABS1\Key Admins)
Add...
Remove
Permissions for Sophie Lambert
Allow
Deny
Full control
Read
Write
Create all child objects
Delete all child objects
For special permissions or advanced settings, click Advanced.
Advanced
OK
Cancel
Apply
Help


Advanced Security Settings for IT Services
Owner: Domain Admins (VLABS1\Domain Admins) Change
Permissions
Auditing
Effective Access
For additional information, double-click a permission entry. To modify a permission entry, select the entry and click Edit (if available).
Permission entries:
Type
Principal
Access
Inherited from
Applies to
Add
Remove
View
Restore defaults
Activate Windows

From ADAC



3. List the delegation permissions on the **IT Services** OU using **PowerShell.**

```
Get-Acl "AD:OU=IT Services,DC=vlabs25,DC=com" | Select-Object -ExpandProperty Access | Where-Object { $_.IdentityReference -like "*Lambert*" }
```

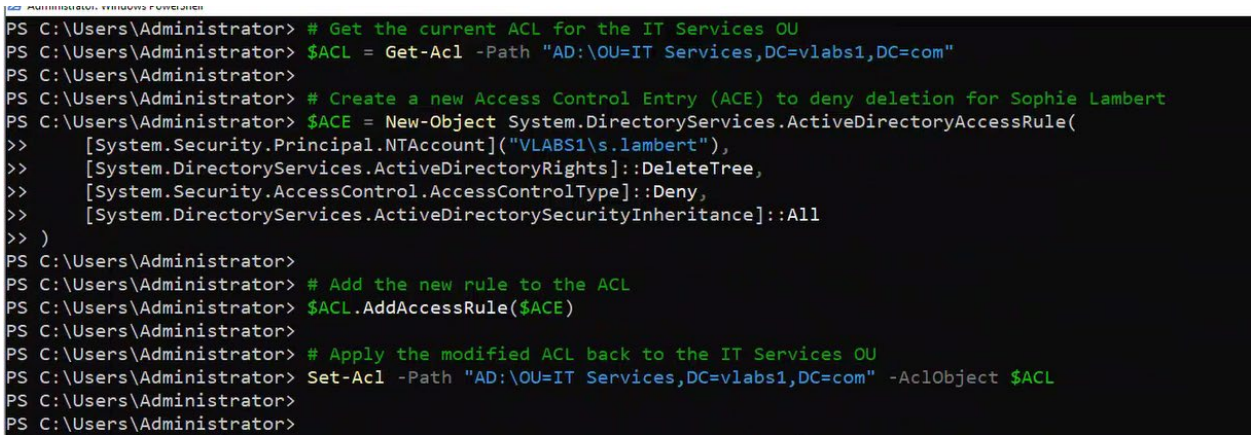
```
PS C:\Users\Administrator> Get-Acl "AD:OU=IT Services,DC=vlabs1,DC=com" | Select-Object -ExpandProperty Access | Where-Object { $_.IdentityReference -like "*L
ambert*" }


ActiveDirectoryRights : GenericAll
InheritanceType       : Descendents
ObjectType            : 00000000-0000-0000-0000-000000000000
InheritedObjectType   : bf967aba-0de6-11d0-a285-00aa003049e2
ObjectFlags           : InheritedObjectAceTypePresent
AccessControlType     : Allow
IdentityReference     : VLABS1\s.lambert
IsInherited           : False
InheritanceFlags      : ContainerInherit
PropagationFlags      : InheritOnly

ActiveDirectoryRights : CreateChild, DeleteChild
InheritanceType       : All
ObjectType            : bf967aba-0de6-11d0-a285-00aa003049e2
InheritedObjectType   : 00000000-0000-0000-0000-000000000000
ObjectFlags           : ObjectAceTypePresent
AccessControlType     : Allow
IdentityReference     : VLABS1\s.lambert
IsInherited           : False
InheritanceFlags      : ContainerInherit
PropagationFlags      : None
```

2.1.7 Task 7: Managing Permissions on OUs

1. Use **PowerShell** to **deny deletion of objects** inside **IT Services** OU for **Sophie Lambert.**

```
# Get the current ACL for the IT Services OU
$ACL = Get-Acl -Path "AD:\OU=IT Services,DC=vlabs1,DC=com"
```

```
# Create a new Access Control Entry (ACE) to deny deletion for Sophie Lambert
$ACE = New-Object System.DirectoryServices.ActiveDirectoryAccessRule(
    [System.Security.Principal.NTAccount]("VLABS1\s.lambert"),
    [System.DirectoryServices.ActiveDirectoryRights]::DeleteTree,
    [System.Security.AccessControl.AccessControlType]::Deny,
    [System.DirectoryServices.ActiveDirectorySecurityInheritance]::All
)

# Add the new rule to the ACL
$ACL.AddAccessRule($ACE)

# Apply the modified ACL back to the IT Services OU
Set-Acl -Path "AD:\OU=IT Services,DC=vlabs1,DC=com" -AclObject $ACL
```



Verify

```
Get-Acl -Path "AD:\OU=IT Services,DC=vlabs1,DC=com" | Select-Object -ExpandProperty Access| Where-Object { $_.IdentityReference -like "*Lambert"}
```

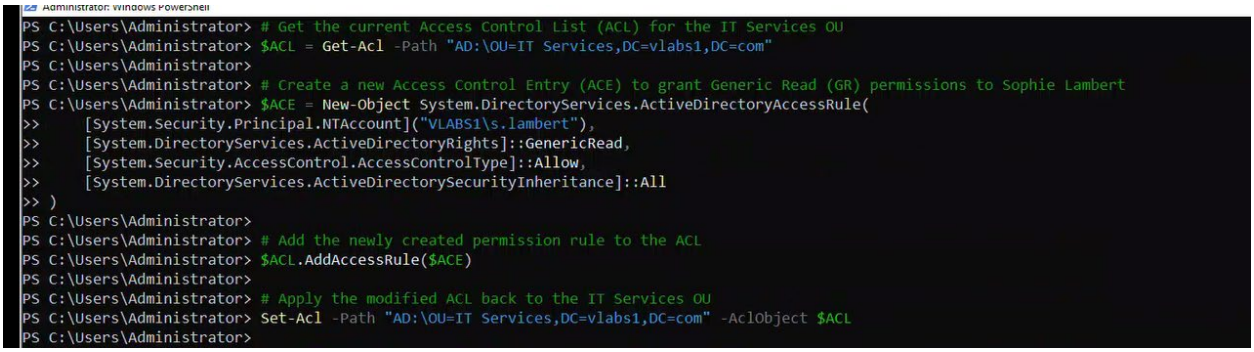
```
PS C:\Users\Administrator> Get-Acl "AD:OU=IT Services,DC=vlabs1,DC=com" | Select-Object -ExpandProperty Access | Where-Object { $_.IdentityReference -like "*L
ambert*" }


ActiveDirectoryRights : DeleteTree
InheritanceType       : All
ObjectType            : 00000000-0000-0000-0000-000000000000
InheritedObjectType   : 00000000-0000-0000-0000-000000000000
ObjectFlags           : None
AccessControlType     : Deny
IdentityReference     : VLABS1\s.lambert
IsInherited           : False
InheritanceFlags      : ContainerInherit
PropagationFlags      : None

ActiveDirectoryRights : GenericAll
InheritanceType       : Descendents
ObjectType            : 00000000-0000-0000-0000-000000000000
InheritedObjectType   : bf967aba-0de6-11d0-a285-00aa003049e2
ObjectFlags           : InheritedObjectAceTypePresent
AccessControlType     : Allow
IdentityReference     : VLABS1\s.lambert
IsInherited           : False
InheritanceFlags      : ContainerInherit
PropagationFlags      : InheritOnly

ActiveDirectoryRights : CreateChild, DeleteChild
InheritanceType       : All
ObjectType            : bf967aba-0de6-11d0-a285-00aa003049e2
InheritedObjectType   : 00000000-0000-0000-0000-000000000000
ObjectFlags           : ObjectAceTypePresent
AccessControlType     : Allow
```

2. Use **PowerShell** to **grant Generic Read (GR)** permissions on **IT Services** OU to **Sophie Lambert.**

```
# Get the current Access Control List (ACL) for the IT Services OU

$ACL = Get-Acl -Path "AD:\OU=IT Services,DC=vlabs1,DC=com"


# Create a new Access Control Entry (ACE) to grant Generic Read (GR) permissions to Sophie Lambert

$ACE = New-Object System.DirectoryServices.ActiveDirectoryAccessRule(

   [System.Security.Principal.NTAccount]("VLABS1\s.lambert"),

   [System.DirectoryServices.ActiveDirectoryRights]::GenericRead,

   [System.Security.AccessControl.AccessControlType]::Allow,

   [System.DirectoryServices.ActiveDirectorySecurityInheritance]::All

)


# Add the newly created permission rule to the ACL

$ACL.AddAccessRule($ACE)
```

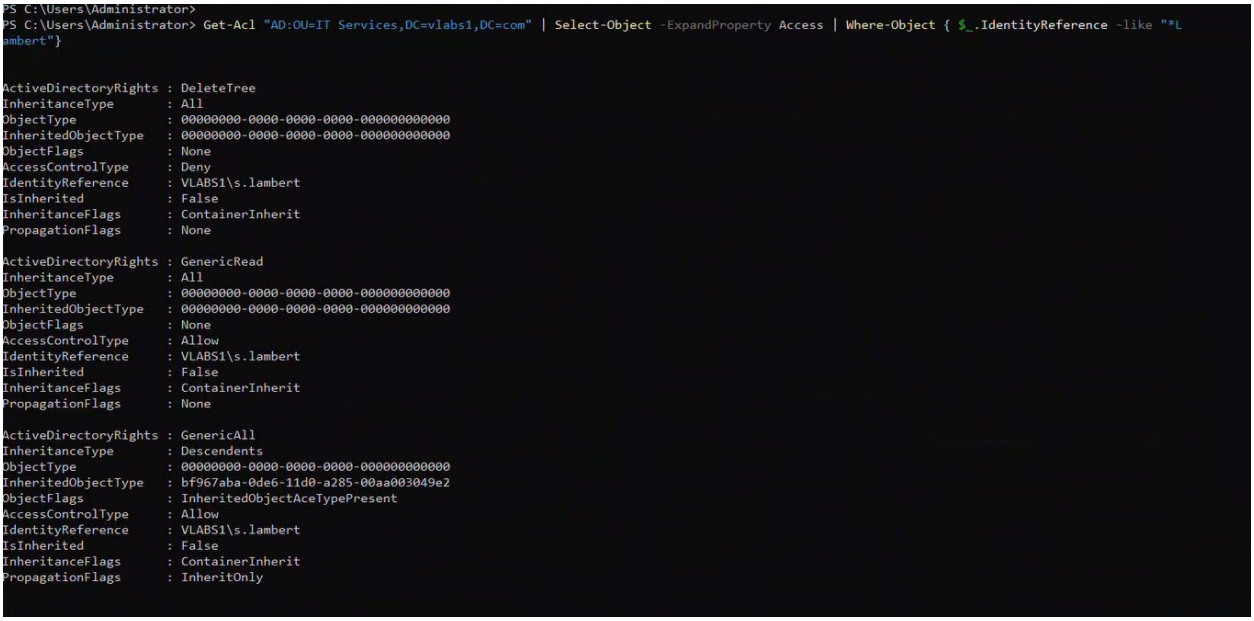
 # Apply the modified ACL back to the IT Services OU

Set-Acl -Path "AD:\OU=IT Services,DC=vlabs1,DC=com" -AclObject $ACL



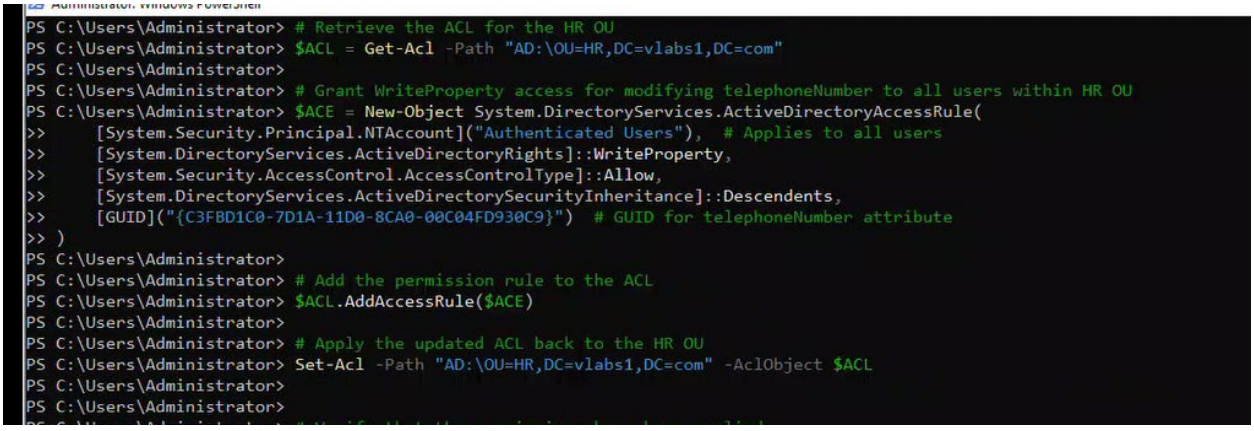
 # Verify that the permissions were successfully applied

Get-Acl -Path "AD:\OU=IT Services,DC=vlabs1,DC=com" | Select-Object -ExpandProperty Access| Where-Object { $_.IdentityReference -like "*Lambert"}



3. Use **PowerShell** to **grant write permissions for modifying the telephoneNumber** attribute for **all the users** in the **HR** OU.

```
# Retrieve the ACL for the HR OU
$ACL = Get-Acl -Path "AD:\OU=HR,DC=vlabs1,DC=com"

# Grant WriteProperty access for modifying telephoneNumber to all users within HR OU
$ACE = New-Object System.DirectoryServices.ActiveDirectoryAccessRule(
    [System.Security.Principal.NTAccount]("Authenticated Users"),  # Applies to all users
    [System.DirectoryServices.ActiveDirectoryRights]::WriteProperty,
    [System.Security.AccessControl.AccessControlType]::Allow,
    [System.DirectoryServices.ActiveDirectorySecurityInheritance]::Descendents,
    [GUID]("{C3FBD1C0-7D1A-11D0-8CA0-00C04FD930C9}")  # GUID for telephoneNumber attribute
)

# Add the permission rule to the ACL
$ACL.AddAccessRule($ACE)

# Apply the updated ACL back to the HR OU
Set-Acl -Path "AD:\OU=HR,DC=vlabs1,DC=com" -AclObject $ACL
```



 # Verify that the permissions have been applied

Get-Acl -Path "AD:\OU=HR,DC=vlabs1,DC=com" | Select-Object -ExpandProperty Access

```
PS C:\Users\Administrator> Get-Acl -Path "AD:\OU=HR,DC=vlabs1,DC=com" | Select-Object -ExpandProperty Access


ActiveDirectoryRights : DeleteTree, Delete
InheritanceType       : None
ObjectType            : 00000000-0000-0000-0000-000000000000
InheritedObjectType   : 00000000-0000-0000-0000-000000000000
ObjectFlags           : None
AccessControlType     : Deny
IdentityReference     : Everyone
IsInherited           : False
InheritanceFlags      : None
PropagationFlags      : None

ActiveDirectoryRights : GenericRead
InheritanceType       : None
ObjectType            : 00000000-0000-0000-0000-000000000000
InheritedObjectType   : 00000000-0000-0000-0000-000000000000
ObjectFlags           : None
AccessControlType     : Allow
IdentityReference     : NT AUTHORITY\ENTERPRISE DOMAIN CONTROLLERS
IsInherited           : False
InheritanceFlags      : None
PropagationFlags      : None

ActiveDirectoryRights : GenericRead
InheritanceType       : None
ObjectType            : 00000000-0000-0000-0000-000000000000
InheritedObjectType   : 00000000-0000-0000-0000-000000000000
ObjectFlags           : None
AccessControlType     : Allow
IdentityReference     : NT AUTHORITY\Authenticated Users
IsInherited           : False
InheritanceFlags      : None
PropagationFlags      : None

ActiveDirectoryRights : GenericAll
InheritanceType       : None
ObjectType            : 00000000-0000-0000-0000-000000000000
InheritedObjectType   : 00000000-0000-0000-0000-000000000000
ObjectFlags           : None
AccessControlType     : Allow
IdentityReference     : NT AUTHORITY\SYSTEM
IsInherited           : False
InheritanceFlags      : None
PropagationFlags      : None
```

(Get-Acl -Path "AD:\OU=HR,DC=vlabs1,DC=com").Access | Where-Object { $_.IdentityReference -like "*Authenticated Users*" }

```
PS C:\Users\Administrator> (Get-Acl -Path "AD:\OU=HR,DC=vlabs1,DC=com").Access | Where-Object { $_.IdentityReference -like "*Authenticated Users*" }


ActiveDirectoryRights : GenericRead
InheritanceType       : None
ObjectType            : 00000000-0000-0000-0000-000000000000
InheritedObjectType   : 00000000-0000-0000-0000-000000000000
ObjectFlags           : None
AccessControlType     : Allow
IdentityReference     : NT AUTHORITY\Authenticated Users
IsInherited           : False
InheritanceFlags      : None
PropagationFlags      : None

ActiveDirectoryRights : WriteProperty
InheritanceType       : Descendents
ObjectType            : 00000000-0000-0000-0000-000000000000
InheritedObjectType   : c3fbd1c0-7d1a-11d0-8ca0-00c04fd930c9
ObjectFlags           : InheritedObjectAceTypePresent
AccessControlType     : Allow
IdentityReference     : NT AUTHORITY\Authenticated Users
IsInherited           : False
InheritanceFlags      : ContainerInherit
PropagationFlags      : InheritOnly


PS C:\Users\Administrator>
```

4. Use **PowerShell** to **remove all permissions** for **Lucas Bernard** on the **HR** OU.

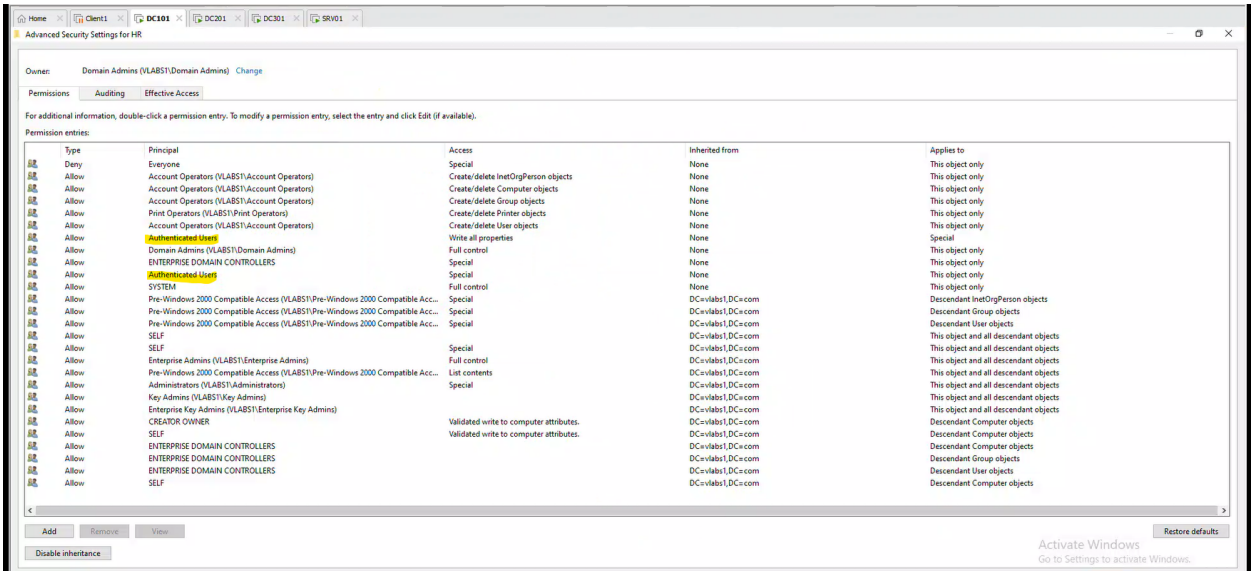
```
# Retrieve the current ACL for the HR OU
$ACL = Get-Acl -Path "AD:\OU=HR,DC=vlabs1,DC=com"

# Find and remove all access rules for Lucas Bernard
$ACL.Access | Where-Object { $_.IdentityReference -eq "VLABS1\l.bernard" } |
ForEach-Object { $ACL.RemoveAccessRule($_) }

# Apply the updated ACL back to the HR OU
Set-Acl -Path "AD:\OU=HR,DC=vlabs1,DC=com" -AclObject $ACL

# Verify that permissions have been removed
Get-Acl -Path "AD:\OU=HR,DC=vlabs1,DC=com" | Select-Object -ExpandProperty
Access
```

```
Administrator: Windows PowerShell
PS C:\Users\Administrator> # Retrieve the current ACL for the HR OU
PS C:\Users\Administrator> $ACL = Get-Acl -Path "AD:\OU=HR,DC=vlabs1,DC=com"
PS C:\Users\Administrator>
PS C:\Users\Administrator> # Find and remove all access rules for Lucas Bernard
PS C:\Users\Administrator> $ACL.Access | Where-Object { $_.IdentityReference -eq "VLABS1\l.bernard" } | ForEach-Object { $ACL.RemoveAccessRule($_) }
PS C:\Users\Administrator>
PS C:\Users\Administrator> # Apply the updated ACL back to the HR OU
PS C:\Users\Administrator> Set-Acl -Path "AD:\OU=HR,DC=vlabs1,DC=com" -AclObject $ACL
PS C:\Users\Administrator>
```

```
PS C:\Users\Administrator> Get-Acl -Path "AD:\OU=HR,DC=vlabs1,DC=com" | Select-Object -ExpandProperty Access


ActiveDirectoryRights : DeleteTree, Delete
InheritanceType       : None
ObjectType            : 00000000-0000-0000-0000-000000000000
InheritedObjectType   : 00000000-0000-0000-0000-000000000000
ObjectFlags           : None
AccessControlType     : Deny
IdentityReference     : Everyone
IsInherited           : False
InheritanceFlags      : None
PropagationFlags      : None

ActiveDirectoryRights : GenericRead
InheritanceType       : None
ObjectType            : 00000000-0000-0000-0000-000000000000
InheritedObjectType   : 00000000-0000-0000-0000-000000000000
ObjectFlags           : None
AccessControlType     : Allow
IdentityReference     : NT AUTHORITY\ENTERPRISE DOMAIN CONTROLLERS
IsInherited           : False
InheritanceFlags      : None
PropagationFlags      : None

ActiveDirectoryRights : GenericRead
InheritanceType       : None
ObjectType            : 00000000-0000-0000-0000-000000000000
InheritedObjectType   : 00000000-0000-0000-0000-000000000000
ObjectFlags           : None
AccessControlType     : Allow
IdentityReference     : NT AUTHORITY\Authenticated Users
IsInherited           : False
InheritanceFlags      : None
```

5. Check which **users** or **groups** have been delegated permission over the **HR** OU using **AD Users and Computers.**

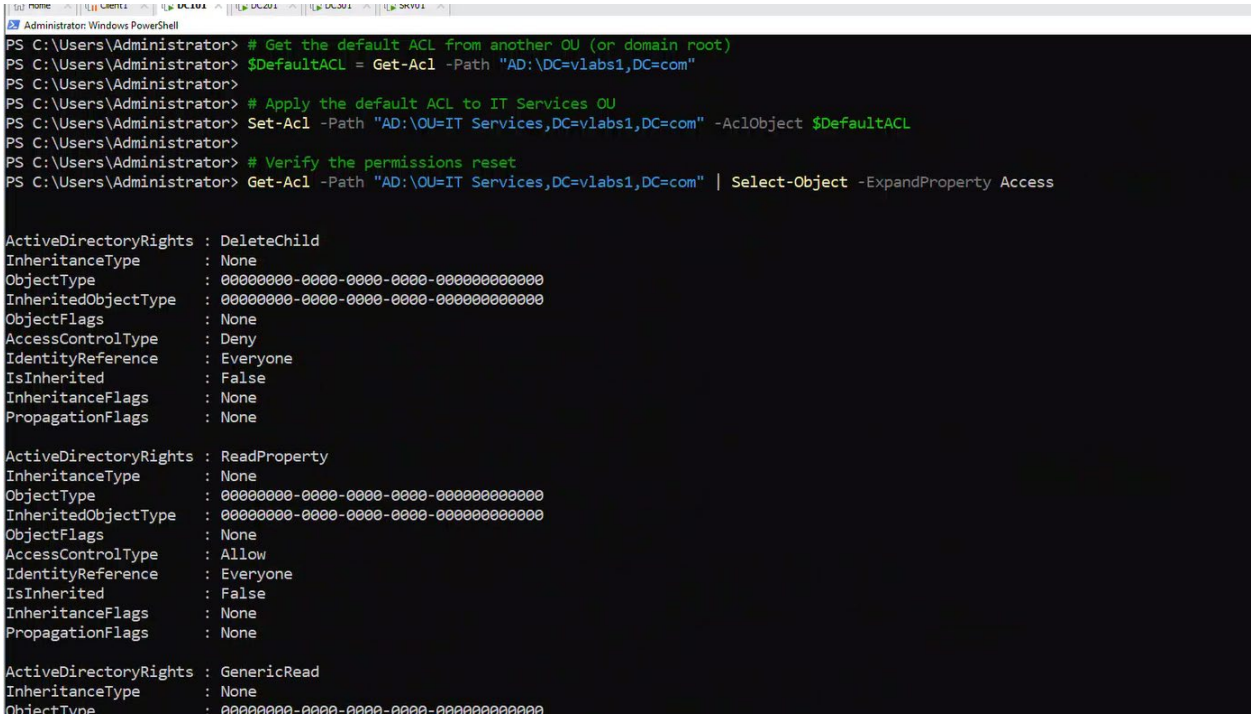
- **Open Active Directory Users and Computers.**
- **Enable Advanced Features (View menu).**
- **Right-click HR OU → Properties → Security tab → Advanced to view permissions.**



6. Reset permissions of the **IT Services** OU to **default** using **PowerShell.**

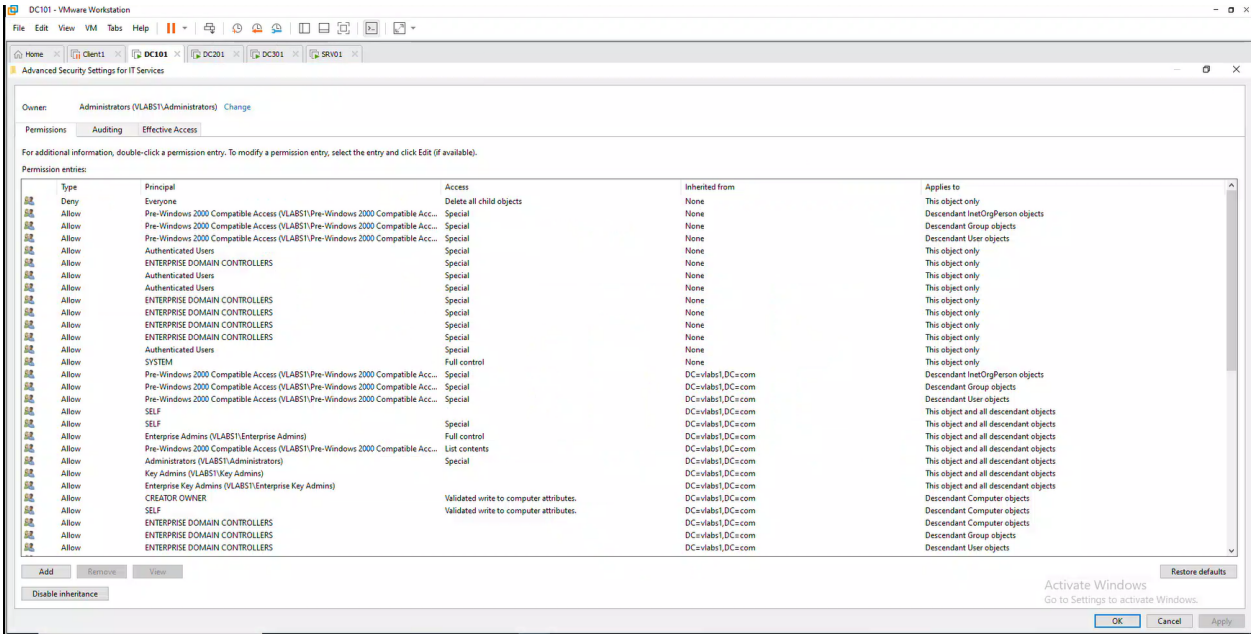
```
# Get the default ACL from another OU (or domain root)

$DefaultACL = Get-Acl -Path "AD:\DC=vlabs1,DC=com"
```

```
# Apply the default ACL to IT Services OU

Set-Acl -Path "AD:\OU=IT Services,DC=vlabs1,DC=com" -AclObject $DefaultACL


# Verify the permissions reset

Get-Acl -Path "AD:\OU=IT Services,DC=vlabs1,DC=com" | Select-Object -ExpandProperty Access
```



7. Check that the permissions of the **IT Services** OU has been reset to default using **AD Users and Computers.**

- **Open Active Directory Users and Computers.**
- **Enable Advanced Features (View menu).**
- **Right-click IT Services OU → Properties → Security tab → Advanced to view permissions.**


DC101 - VMware Workstation
File Edit View VM Tabs Help
Home Client1 DC101 DC201 DC301 SRV01
Advanced Security Settings for IT Services
Owner: Administrators (VLABS1\Administrators) Change
Permissions Auditing Effective Access
For additional information, double-click a permission entry. To modify a permission entry, select the entry and click Edit (if available).
Permission entries:
Type Principal Access Inherited from Applies to
Deny Everyone Delete all child objects None This object only
Allow Pre-Windows 2000 Compatible Access (VLABS1\Pre-Windows 2000 Compatible Acc... Special None Descendant InetOrgPerson objects
Allow Pre-Windows 2000 Compatible Access (VLABS1\Pre-Windows 2000 Compatible Acc... Special None Descendant Group objects
Allow Pre-Windows 2000 Compatible Access (VLABS1\Pre-Windows 2000 Compatible Acc... Special None Descendant User objects
Allow Authenticated Users Special None This object only
Allow ENTERPRISE DOMAIN CONTROLLERS Special None This object only
Allow Authenticated Users Special None This object only
Allow Authenticated Users Special None This object only
Allow ENTERPRISE DOMAIN CONTROLLERS Special None This object only
Allow ENTERPRISE DOMAIN CONTROLLERS Special None This object only
Allow ENTERPRISE DOMAIN CONTROLLERS Special None This object only
Allow ENTERPRISE DOMAIN CONTROLLERS Special None This object only
Allow Authenticated Users Special None This object only
Allow SYSTEM Full control None This object only
Allow Pre-Windows 2000 Compatible Access (VLABS1\Pre-Windows 2000 Compatible Acc... Special DC=vlabs1,DC=com Descendant InetOrgPerson objects
Allow Pre-Windows 2000 Compatible Access (VLABS1\Pre-Windows 2000 Compatible Acc... Special DC=vlabs1,DC=com Descendant Group objects
Allow Pre-Windows 2000 Compatible Access (VLABS1\Pre-Windows 2000 Compatible Acc... Special DC=vlabs1,DC=com Descendant User objects
Allow SELF DC=vlabs1,DC=com This object and all descendant objects
Allow SELF Special DC=vlabs1,DC=com This object and all descendant objects
Allow Enterprise Admins (VLABS1\Enterprise Admins) Full control DC=vlabs1,DC=com This object and all descendant objects
Allow Pre-Windows 2000 Compatible Access (VLABS1\Pre-Windows 2000 Compatible Acc... List contents DC=vlabs1,DC=com This object and all descendant objects
Allow Administrators (VLABS1\Administrators) Special DC=vlabs1,DC=com This object and all descendant objects
Allow Key Admins (VLABS1\Key Admins) DC=vlabs1,DC=com This object and all descendant objects
Allow Enterprise Key Admins (VLABS1\Enterprise Key Admins) DC=vlabs1,DC=com This object and all descendant objects
Allow CREATOR OWNER Validated write to computer attributes. DC=vlabs1,DC=com Descendant Computer objects
Allow SELF Validated write to computer attributes. DC=vlabs1,DC=com Descendant Computer objects
Allow ENTERPRISE DOMAIN CONTROLLERS DC=vlabs1,DC=com Descendant Computer objects
Allow ENTERPRISE DOMAIN CONTROLLERS DC=vlabs1,DC=com Descendant Group objects
Allow ENTERPRISE DOMAIN CONTROLLERS DC=vlabs1,DC=com Descendant User objects
Add Remove View
Restore defaults
Disable inheritance
Activate Windows
Go to Settings to activate Windows.
OK Cancel Apply