Table of Contents

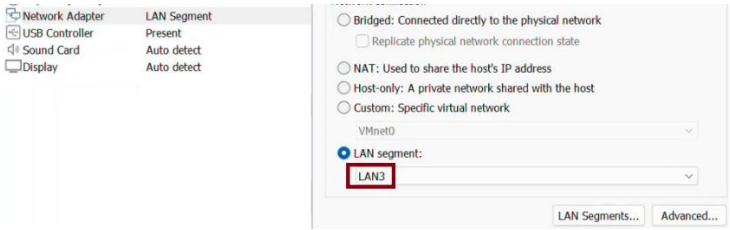
Lab Assignment 1 (Part II)

Task 1 – Configuring DC207

System: DC207

Explanation:
In this task, I configure the network settings for the domain controller DC207.
I need to move it to LAN segment 3 and assign a static IP address. This step prepares DC207 for Active Directory Sites and Services replication by associating
it with the correct site in a later task.

Screenshot:



Objectives:
- Assign IP address: 192.168.45.1
- Subnet mask: 255.255.255.0
- Gateway: 192.168.45.50
- DNS: Keep existing (e.g., 192.168.7.1)
- Test connectivity to gateway and DNS

Step 1: Assign static IP address and default gateway
netsh interface ipv4 set address name="Ethernet0" static 192.168.45.1 255.255.255.0 192.168.45.50
Screenshot:

```
PS C:\Users\Administrator.VLABS07> netsh interface ipv4 set address name="Ethernet0" static 192.168.45.
1 255.255.255.0 192.168.45.50
```

Step 2: (No change to DNS – assumed to already be 192.168.7.1)
Screenshot:

```
Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) 82574L Gigabit Network Connection
   Physical Address. . . . . . . . . : 00-0C-29-16-AD-C0
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::31a8:416b:6b7d:d0ac%6(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.45.1(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.45.50
   DHCPv6 IAID . . . . . . . . . . . : 100666409
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-2F-A3-3D-C6-00-0C-29-16-AD-C0
   DNS Servers . . . . . . . . . . . : ::1
                                       192.168.7.1
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

Step 3: Test connectivity to default gateway

ping 192.168.45.50
Screenshot:

```
PS C:\Users\Administrator.VLABS07> ping 192.168.45.50

Pinging 192.168.45.50 with 32 bytes of data:
Reply from 192.168.45.50: bytes=32 time<1ms TTL=255
Reply from 192.168.45.50: bytes=32 time<1ms TTL=255
Reply from 192.168.45.50: bytes=32 time<1ms TTL=255
Reply from 192.168.45.50: bytes=32 time<1ms TTL=255

Ping statistics for 192.168.45.50:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Step 4: Test connectivity to the root domain DNS server (DC107)
ping 192.168.7.1
Screenshot:

```
PS C:\Users\Administrator.VLABS07> ping 192.168.7.1

Pinging 192.168.7.1 with 32 bytes of data:
Reply from 192.168.7.1: bytes=32 time<1ms TTL=127
Reply from 192.168.7.1: bytes=32 time<1ms TTL=127
Reply from 192.168.7.1: bytes=32 time<1ms TTL=127
Reply from 192.168.7.1: bytes=32 time<1ms TTL=127

Ping statistics for 192.168.7.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Step 5: View and confirm current IP configuration

```
Get-NetIPAddress | Where-Object { $_.InterfaceAlias -eq "Ethernet0" }
```

Command Breakdown:

- Get-NetIPAddress: Lists all IP address info for the system (IPv4 and IPv6)
- Where-Object: Filters the list
- $_.InterfaceAlias -eq "Ethernet0": Only shows info for the Ethernet0 adapter

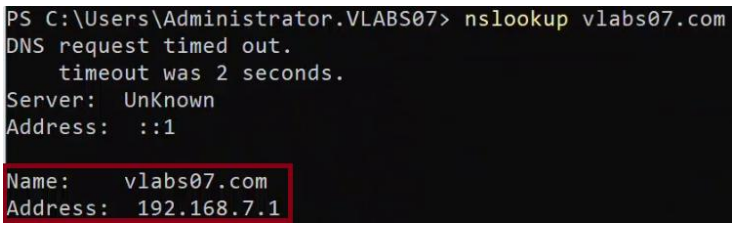
Screenshot:

```
PS C:\Users\Administrator.VLABS07> Get-NetIPAddress | Where-Object { $_.InterfaceAlias -eq "Ethernet0"
}


IPAddress         : fe80::31a8:416b:6b7d:d0ac%6
InterfaceIndex    : 6
InterfaceAlias    : Ethernet0
AddressFamily     : IPv6
Type              : Unicast
PrefixLength      : 64
PrefixOrigin      : WellKnown
SuffixOrigin      : Link
AddressState      : Preferred
ValidLifetime     :
PreferredLifetime :
SkipAsSource      : False
PolicyStore       : ActiveStore

IPAddress         : 192.168.45.1
InterfaceIndex    : 6
InterfaceAlias    : Ethernet0
AddressFamily     : IPv4
Type              : Unicast
PrefixLength      : 24
PrefixOrigin      : Manual
SuffixOrigin      : Manual
```

Step 6: (Optional) Confirm DNS resolution is working

nslookup vlabs07.com

Screenshot:



Task 2 – Part 1 Demote and Prepare DC407

System: DC407

Explanation:
In this task, we will reuse the existing server DC407. First, we demote it from its old forest, clean up all AD-related remnants, and then safely rejoin and promote it into our current forest (vlabs07.com) as a child domain controller for partner07.vlabs07.com. This avoids having to build a new server (like DC507) and shows how to properly handle domain reuse.

Steps:
1. Remove the old trust
2. Demote DC407
3. Clean leftover metadata and DNS records
4. Rejoin the root domain
5. Promote it as a child domain controller

Step 1 – Remove the old two-way trust with vlabs07.com
```
netdom trust /d:vlabs07.com partner07.com /uo:partner07\administrator
/ud:vlabs07\administrator /pd:* /po:* /remove /twoway /verbose
```

Screenshot:

```
PS C:\Users\Administrator> netdom trust /d:vlabs07.com partner07.com /uo:partner07\administrato
r /ud:vlabs07\administrator /pd:* /po:* /remove /twoway /verbose
Removing the trust object for vlabs07.com

Opening the trusted domain object partner07.com

Removing the trust object for partner07.com

Deleting the session with \\DC107.vlabs07.com

Deleting the session with \\DC407.partner07.com

The command completed successfully.
```

Command Breakdown:
- netdom trust: Manage trust relationships
- /d:vlabs07.com: Target domain for trust removal
- /uo and /ud: Usernames for both domains
- /pd:* and /po:*: Prompt for passwords
- /remove: Actually remove the trust
- /twoway: Removes both sides of the trust
- /verbose: Show detailed output

Step 2 – Demote DC407 from its old domain
```
Uninstall-ADDSDomainController `
  -LocalAdministratorPassword (ConvertTo-SecureString "Passw0rd$" -
AsPlainText -Force) `
  -LastDomainControllerInDomain `
  -RemoveApplicationPartitions `
  -Force
```

Screenshot:

```
PS C:\Users\Administrator> Uninstall-ADDSDomainController -LocalAdministratorPassword (ConvertTo-Secu
reString "Passw0rd$" -AsPlainText -Force) -LastDomainControllerInDomain -RemoveApplicationPartitions
-Force

Message                          Context          RebootRequired  Status
-------                          -------          --------------  ------
Operation completed successfully DCPromo.General.1          False Success
```

Command Breakdown:
- Uninstall-ADDSDomainController: Demotes the domain controller
- -LocalAdministratorPassword: Password to use for the local admin after demotion
- -LastDomainControllerInDomain: Confirms this is the last DC in its domain
- -RemoveApplicationPartitions: Removes DNS or other AD partitions
- -Force: Runs without asking for confirmation

Step 3 – Remove the AD DS role

```
Uninstall-WindowsFeature AD-Domain-Services -IncludeManagementTools -Restart
```

Screenshot:

```
PS C:\Users\Administrator.DC407> Uninstall-WindowsFeature AD-Domain-Services -IncludeManagementTools -Restar
t
```

Command Breakdown:
- Removes the AD DS role so the server becomes a regular member server
- -IncludeManagementTools: Also removes AD tools
- -Restart: Automatically restarts the server

Step 4 – Manually remove old folders and DNS suffix left by AD DS

```
Remove-Item -Path "C:\Windows\NTDS" -Recurse -Force -ErrorAction
SilentlyContinue
Remove-Item -Path "C:\Windows\SYSVOL" -Recurse -Force -ErrorAction
SilentlyContinue
Set-ItemProperty -Path
"HKLM:\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters" -Name Domain -
Value ""
```

Screenshot:

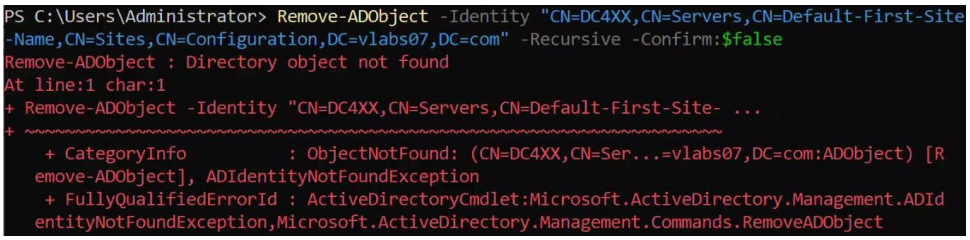
```
PS C:\Users\Administrator.DC407> Remove-Item -Path "C:\Windows\NTDS" -Recurse -Force -ErrorAction Sil
entlyContinue
PS C:\Users\Administrator.DC407> Remove-Item -Path "C:\Windows\SYSVOL" -Recurse -Force -ErrorAction S
ilentlyContinue
PS C:\Users\Administrator.DC407> Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Services\Tcpi
p\Parameters" -Name Domain -Value ""
```

Command Breakdown:
- NTDS and SYSVOL folders may be left behind after demotion
- We delete them manually to avoid conflicts during promotion
- -ErrorAction SilentlyContinue hides errors if folder doesnt exist
- Set-ItemProperty: Clears the Primary DNS suffix (e.g., partner07.com) from the registry
  This is important to avoid DNS suffix conflicts when joining the new domain

Step 5 – Clean up AD metadata from another DC (e.g., DC107 or DC207)

```
Remove-ADObject -Identity "CN=DC407,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=vlabs07,DC=com" -Recursive -Confirm:$false
```

Screenshot:



Command Breakdown:

- Removes any lingering DC407 metadata from Active Directory Sites and Services
- -Recursive: Deletes child objects too (like NTDS Setting
- -Confirm:**$false**: Does not ask for confirmation

BONUS STEP – (Optional) Clean Up Old User Profiles Using GUI

Explanation:

After demoting DC407 and removing it from the previous domain, Windows may leave behind old user profiles such as:

- C:\Users\Administrator.DC407
- Orphaned "Account Unknown" entries

I recommend deleting any unused profiles before rejoining the server to the new domain.

GUI Steps:

---

1. Press Windows Key **+** R → Type: SystemPropertiesAdvanced → Press Enter
2. Under the "User Profiles" section, click the **Settings** button
3. In the list of profiles:
   - Select any profile named "DC407\Administrator", "Administrator.DC407.000", or "Account Unknown"
4. Click **Delete** for each unwanted profile

Note:

- Do **not delete** the currently logged-in user
- Do **not delete** "Default Profile"
- After deleting profiles, restart the server and log in as `vlabs07\administrator` to ensure a clean new domain profile is created

BONUS STEP (Continued) – Clean Corrupted Profile from Registry (Registry Fix)

Explanation:

If an old user profile like C:\Users\Administrator.DC407.000 was deleted manually
without first removing it via the GUI, Windows may keep a registry reference to
the missing profile. This results in login issues such as:

! "We can't sign in to your account"

The solution is to delete the corresponding user profile SID from the registry.
This allows Windows to generate a fresh profile path on the next login.

NOTE:

Make sure youre signed in with a different admin account (e.g., vlabs07\administrator)
before deleting the registry key.

Manual Registry Cleanup Steps:
---
1. Press Win + R → type: regedit → press Enter
2. Navigate to:
   HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList
3. Expand each S-1-5-21-... key and look for the one with:
   ProfileImagePath = C:\Users\Administrator.DC407.000
4. Right-click the matching SID key → click Delete
5. Close Registry Editor and restart the server

Result:

- The broken profile is removed from the registry
- A clean new profile (e.g., C:\Users\Administrator) will be created on next login

Task 2 – Part 2 Rejoin and Promote DC407

System: DC407

Explanation:
After DC407 is demoted and restarted, we now reconfigure its network settings, rejoin it to the main forest (vlabs07.com), and promote it as a child domain controller for partner07.vlabs07.com.

This part of the task includes:
- Resetting DNS settings
- Disabling IPv6
- Verifying network connectivity
- Joining the domain
- Installing the AD DS role again
- Promoting to a new child domain

STEP 1 – Set DNS to root domain controller (e.g., DC107)

```
netsh interface ip add dns name="Ethernet0" 192.168.7.1 index=1
```

Screenshot:

```
PS C:\Users\Administrator.DC407> netsh interface ip add dns name="Ethernet0" 192.168.7.1 index=1
```

Breakdown:
- netsh interface ip add dns: Adds the DNS server IP to the network adapter
- name="Ethernet0": Applies settings to the Ethernet0 interface
- 192.168.7.1: IP address of DC107 (root domain DNS)
- index=1: Sets it as the primary DNS server

STEP 2 – Disable IPv6 (optional but recommended for this lab)

```
Disable-NetAdapterBinding -Name "Ethernet0" -ComponentID ms_tcpip6
```

Screenshot:

```
PS C:\Users\Administrator.DC407> Disable-NetAdapterBinding -Name "Ethernet0" -ComponentID ms_tcpip6
```

Breakdown:
- Disables IPv6 protocol on the Ethernet0 adapter
- Prevents dual-stack conflicts in domain environments

STEP 3 – Test network and DNS connectivity

```
ping 192.168.7.1
nslookup vlabs07.com
```

Screenshot:

```
PS C:\Users\Administrator.DC407> ping 192.168.7.1

Pinging 192.168.7.1 with 32 bytes of data:
Reply from 192.168.7.1: bytes=32 time<1ms TTL=127
Reply from 192.168.7.1: bytes=32 time<1ms TTL=127
Reply from 192.168.7.1: bytes=32 time<1ms TTL=127
Reply from 192.168.7.1: bytes=32 time<1ms TTL=127

Ping statistics for 192.168.7.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```
PS C:\Users\Administrator.DC407> nslookup vlabs07.com
DNS request timed out.
    timeout was 2 seconds.
Server:  UnKnown
Address:  192.168.7.1

Name:    vlabs07.com
Address:  192.168.7.1
```

Breakdown:

- ping: Ensures the system can reach the DNS server
- nslookup: Checks DNS resolution to the forest root domain

STEP 4 – Join DC407 to the root domain

Explanation:

I attempted to use the interactive credential prompt with:

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

However, on Windows Server 2025, the Get-Credential command failed to open the GUI prompt,
resulting in a ParameterBindingException. To resolve this, I manually created a secure
credential object using the method below and proceeded to join the domain successfully.

This step joins DC407 to the root domain so it can later be promoted as a child domain controller.

Create the credential object manually

```
$pw = ConvertTo-SecureString "Passw0rd$" -AsPlainText -Force
$cred = New-Object System.Management.Automation.PSCredential("vlabs07\administrator", $pw)
```

Join DC407 to the domain

```
Add-Computer -DomainName "vlabs07.com" -Credential $cred -Verbose -Restart -Force
```

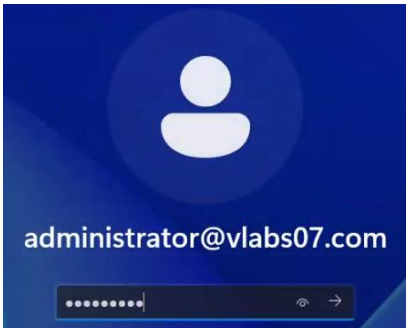
Screenshot:

```
PS C:\Users\Administrator.DC407> $pw = ConvertTo-SecureString "Passw0rd$" -AsPlainText -Force
PS C:\Users\Administrator.DC407> $cred = New-Object System.Management.Automation.PSCredential("vlabs07\administrator", $pw)
PS C:\Users\Administrator.DC407> Add-Computer -DomainName "vlabs07.com" -Credential $cred -Verbose -Restart -Force
```

Command Breakdown:
- ConvertTo-SecureString: Converts a plain password string into a secure format
- New-Object PSCredential: Builds a credential object using the username and secure password
- Add-Computer:
    - -DomainName: Specifies the root domain to join (vlabs07.com)
    - -Credential: Uses the PSCredential object created above
    - -Verbose: Displays detailed feedback
    - -Restart: Automatically reboots the system upon successful domain join

STEP 5 – After reboot, log in as domain admin
Manual step: Use vlabs07\administrator at login screen)

Screenshot:



STEP 6 – Install AD DS role (again)
```
Install-WindowsFeature AD-Domain-Services -IncludeManagementTools
```

Screenshot:

```
PS C:\Users\Administrator.VLABS07> Install-WindowsFeature AD-Domain-Services -IncludeManagementTools

Success Restart Needed Exit Code      Feature Result
------- -------------- ---------      --------------
True    No             Success        {Active Directory Domain Services, Group P...
```

Explanation:
- Adds back the AD DS role to prepare for domain promotion

STEP 7 – Promote DC407 to child domain: partner07.vlabs07.com

```
Install-ADDSDomain `
  -NewDomainName "partner07" `
  -ParentDomainName "vlabs07.com" `
  -InstallDNS `
  -CreateDNSDelegation:$true `
  -DomainMode "WinThreshold" `
  -NoGlobalCatalog:$true `
  -SafeModeAdministratorPassword (ConvertTo-SecureString "Passw0rd$" -
AsPlainText -Force) `
  -Force
```

Screenshot:

```
PS C:\Users\Administrator> Install-ADDSDomain -NewDomainName "partner07" -ParentDomainName "vlabs07.com" -In
stallDNS -CreateDNSDelegation:$true -DomainMode "WinThreshold" -NoGlobalCatalog:$true -SafeModeAdministrator
Password (ConvertTo-SecureString "Passw0rd$" -AsPlainText -Force) -Force
```

Breakdown:

- -NewDomainName: Name of child domain (partner07)
- -ParentDomainName: Root forest domain (vlabs07.com)
- -InstallDNS: Installs DNS role on this child DC
- -CreateDNSDelegation:$true: Optional delegation for DNS setup
- -DomainMode "WinThreshold": Latest Windows Server mode (2025)
- -NoGlobalCatalog:$true: This DC will not be a global catalog (optional)
- -SafeModeAdministratorPassword: For Directory Services Restore Mode
- -Force: Runs without prompts

STEP 8 – Verification: Confirm DC407 was successfully promoted as a Child Domain Controller

----

Explanation:

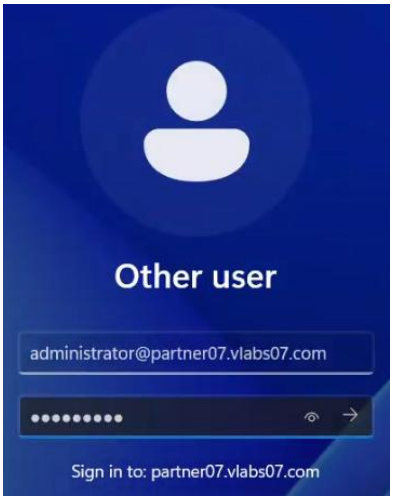
After running the Install-ADDSDomain command to promote DC407 as a new child domain controller
for partner07.vlabs07.com, we must confirm that:

- The domain was correctly created
- DC407 is recognized as the child domain controller
- The forest trust and replication topology are intact

Verification 1 – Login Screen Confirmation

Screenshot: shows successful login screen with domain:
Username: administrator@partner07.vlabs07.com
Sign in to: partner07.vlabs07.com

Screenshot:



Verification 2 – Confirm Domain Hierarchy

Run this command:
Get-ADDomain

Screenshot:

- DistinguishedName: DC=partner07,DC=vlabs07,DC=com
- ParentDomain: vlabs07.com
- PDCE Emulator: DC407.partner07.vlabs07.com

Screenshot:

```
PS C:\Users\Administrator.PARTNER07> Get-ADDomain


AllowedDNSSuffixes                : {}
ChildDomains                      : {}
ComputersContainer                : CN=Computers,DC=partner07,DC=vlabs07,DC=com
DeletedObjectsContainer           : CN=Deleted Objects,DC=partner07,DC=vlabs07,DC=com
DistinguishedName                 : DC=partner07,DC=vlabs07,DC=com
DNSRoot                           : partner07.vlabs07.com
DomainControllersContainer        : OU=Domain Controllers,DC=partner07,DC=vlabs07,DC=com
DomainMode                        : Windows2016Domain
DomainSID                         : S-1-5-21-561883140-3333430699-1010415384
ForeignSecurityPrincipalsContainer : CN=ForeignSecurityPrincipals,DC=partner07,DC=vlabs07,DC=com
Forest                            : vlabs07.com
InfrastructureMaster              : DC407.partner07.vlabs07.com
LastLogonReplicationInterval      :
LinkedGroupPolicyObjects          : {CN={31B2F340-016D-11D2-945F-00C04FB984F9},CN=Policies,CN=System,DC=partner07,DC=
LostAndFoundContainer             : CN=LostAndFound,DC=partner07,DC=vlabs07,DC=com
ManagedBy                         :
Name                              : partner07
NetBIOSName                       : PARTNER07
ObjectClass                       : domainDNS
ObjectGUID                        : cf342430-8171-41c5-b4d0-0519dcb04882
ParentDomain                      : vlabs07.com
PDCEmulator                       : DC407.partner07.vlabs07.com
```

Verification 3 – Confirm Forest Structure

Run this command:
Get-ADForest

Screenshot:

- Domains include: lab07.vlabs07.com, partner07.vlabs07.com, vlabs07.com
- RootDomain: vlabs07.com
- SchemaMaster: DC107.vlabs07.com
- GlobalCatalogs: includes DC107, DC207, DC307

Screenshot:

```
PS C:\Users\Administrator.PARTNER07> Get-ADForest


ApplicationPartitions : {DC=ForestDnsZones,DC=vlabs07,DC=com, DC=DomainDnsZones,DC=vlabs07,DC=com, DC=DomainDnsZones,DC
                        DC=DomainDnsZones,DC=partner07,DC=vlabs07,DC=com}
CrossForestReferences : {}
DomainNamingMaster    : DC107.vlabs07.com
Domains               : {lab07.vlabs07.com, partner07.vlabs07.com, vlabs07.com}
ForestMode            : Windows2016Forest
GlobalCatalogs        : {DC107.vlabs07.com, DC207.vlabs07.com, DC307.lab07.vlabs07.com}
Name                  : vlabs07.com
PartitionsContainer   : CN=Partitions,CN=Configuration,DC=vlabs07,DC=com
RootDomain            : vlabs07.com
SchemaMaster          : DC107.vlabs07.com
Sites                 : {Default-First-Site-Name}
SPNSuffixes           : {}
UPNSuffixes           : {}
```

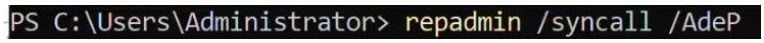
STEP 9 – Forest-Wide Replication Verification (Run from DC107)

Explanation:

After promoting DC407 as a child domain controller (partner07.vlabs07.com), its important to confirm that replication between all domain controllers is functional.
Since DC407 is part of a child domain, only root domain controllers (e.g., DC107) have
the permissions to replicate forest-wide partitions.

Run the following command from DC107 (as vlabs07\administrator):
repadmin /syncall /AdeP

Screenshot:



Switch breakdown:
- **/A** : Include all naming contexts **(**Schema, Configuration, Domain, etc.**)**
- **/d** : Identify DCs by distinguished name
- **/e** : Include cross-site replication
- **/P** : Push changes to all partners

Expected Output:
- "SyncAll terminated with no errors"
- All partition replication should complete without 8453 or access errors

Result:
- Replication completed successfully between:
  - DC107 → DC307
  - DC107 → DC407
  - ForestDnsZones, Schema, Configuration, and Domain partitions

****Final Status Summary – DC407 (partner07.vlabs07.com)****

- Successfully demoted from old forest
- Cleaned up AD DS remnants, profiles, and metadata
- Rejoined vlabs07.com and promoted to child domain controller
- Forest-wide replication validated from DC107

Task 3 – Managing the Connection Objects

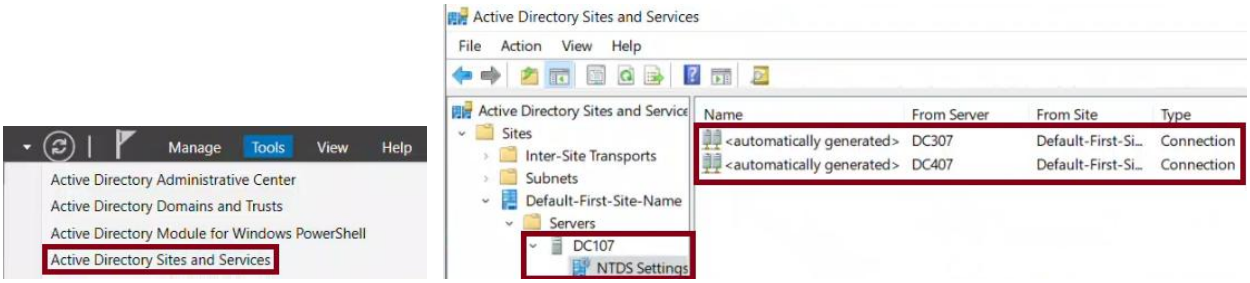
System: DC107

Explanation:
This task focuses on managing replication connection objects in Active Directory.
We start by exploring automatically created connection objects using the GUI on DC107.
Next, we simulate a broken replication link to DC307 and rely on KCC to auto-repair it.
PowerShell is then used to validate replication with DC407, remove and regenerate topology,
and inspect KCC behavior through Event Viewer.

◇ Using GUI

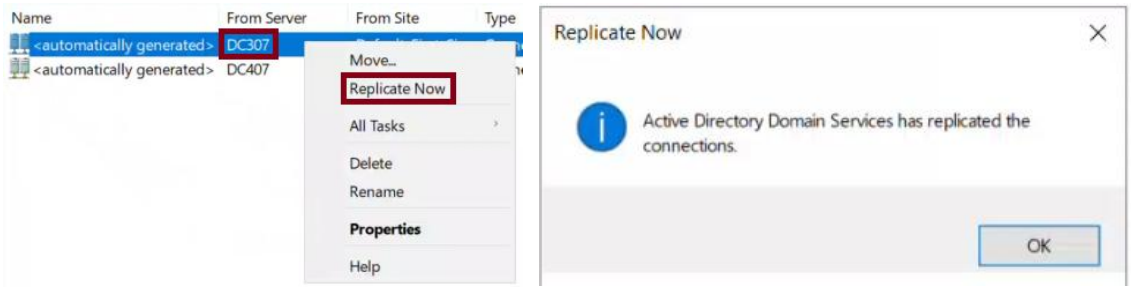
Step 1 – List the Automatically Created Connection Objects on DC107
1. Open **Active Directory Sites and Services**
2. Go to: `Sites > Default-First-Site-Name > Servers > DC107 > NTDS Settings`
3. Observe the connection objects created by the KCC

Screenshot:



Step 2 – Replicate Manually to DC307
1. Still under **NTDS Settings** for DC107
2. Right-click the connection to **DC307**
3. Choose **Replicate Now**

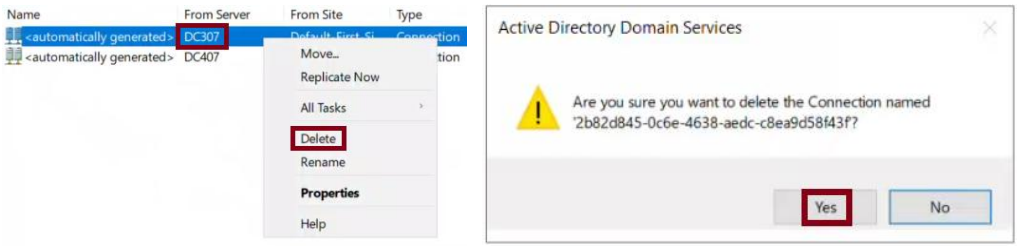
Screenshot:



Step 3 – Delete the Connection Object to DC307

1. In **NTDS Settings**, right-click the connection for DC307
2. Click **Delete**
3. Confirm when prompted

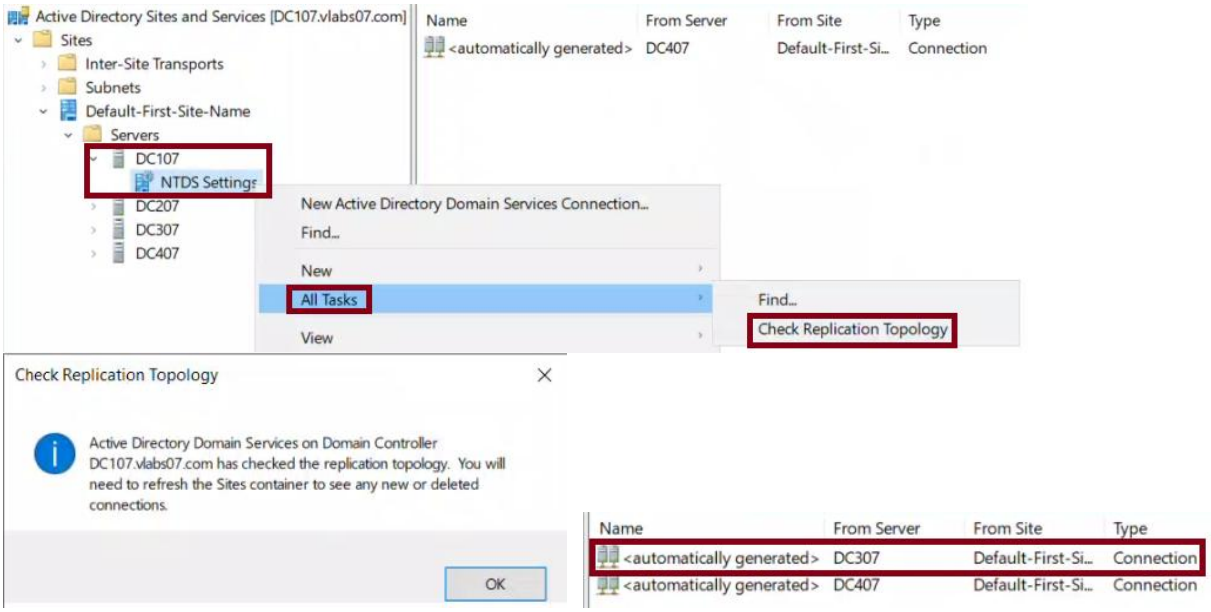
Screenshot:



Step 4 – Recreate It Using the KCC

1. Right-click **NTDS Settings** under DC107
2. Select **All Tasks > Check Replication Topology**
3. The KCC will regenerate the missing connection

Screenshot:



◇ **Using PowerShell**

Step 5 – Manually Replicate to DC407

```
Sync-ADObject -Object (Get-ADComputer DC407) -Destination DC107
```

Command Breakdown:

- Sync-ADObject: Forces a manual replication
- Get-ADComputer DC407: Grabs the DC407 computer object
- -Destination DC107: Initiates replication from DC407 to DC107

Screenshot:

Note:
There are two ways to specify the object to replicate in Sync-ADObject:

Option 1 - By Distinguished Name (DN):
Sync-ADObject -Object "CN=Users,DC=vlabs07,DC=com" -Source DC107 -Destination DC307
This method gives precise control but requires knowing the full DN of the object.

Option 2 - By Object Cmdlet (used in this task):
Sync-ADObject -Object (Get-ADComputer DC407) -Destination DC107
This is more user-friendly for replicating computer objects. It retrieves the object directly from AD without needing to type the full DN.

In a real-world environment, the second method is more readable and easier to use for targeting domain controllers.

Step 6 – Delete the Connection Object to DC307

Explanation:
In this step, I identified and remove the specific replication connection object on DC107 that points to DC307.
I then verify the deletion using a filtered Get-ADObject command.

View all connection objects under DC107

```
Get-ADObject -Filter 'ObjectClass -eq "nTDSConnection"' `
  -SearchBase "CN=NTDS Settings,CN=DC107,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=vlabs07,DC=com" `
  -Properties *
```

Screenshot:

```
PS C:\Users\Administrator> Get-ADObject -Filter 'ObjectClass -eq "nTDSConnection"' -SearchBase "CN=NTDS Settings
,CN=DC107,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=vlabs07,DC=com" -Properties *
CanonicalName                   : vlabs07.com/Configuration/Sites/Default-First-Site-Name/Servers/DC107/NTDS
                                  Settings/15ebf56f-9920-409a-a862-6b5ff469fcc5
CN                              : 15ebf56f-9920-409a-a862-6b5ff469fcc5
Created                         : 5/17/2025 8:28:21 PM
createTimeStamp                 : 5/17/2025 8:28:21 PM
Deleted                         :
Description                     :
DisplayName                     :
DistinguishedName               : CN=15ebf56f-9920-409a-a862-6b5ff469fcc5,CN=NTDS Settings,CN=DC107,CN=Servers,
                                  CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=vlabs07,DC=com
dSCorePropagationData           : {12/31/1600 7:00:00 PM}
enabledConnection               : True
fromServer                      : CN=NTDS Settings,CN=DC307,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=C
                                  onfiguration,DC=vlabs07,DC=com
```

Delete the connection object to DC307

```
Remove-ADObject -Identity "CN=15ebf56f-9920-409a-a862-6b5ff469fcc5,CN=NTDS Settings,CN=DC107,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=vlabs07,DC=com" `
  -Confirm:$false
```

Screenshot:

```
PS C:\Users\Administrator> Remove-ADObject -Identity "CN=15ebf56f-9920-409a-a862-6b5ff469fcc5,CN=NTDS Settings,C
N=DC107,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=vlabs07,DC=com" -Confirm:$false
```

Verify deletion by listing remaining connection objects

```
Get-ADObject -Filter 'ObjectClass -eq "nTDSConnection"' `
  -SearchBase "CN=NTDS Settings,CN=DC107,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=vlabs07,DC=com"
```

Screenshot:

```
PS C:\Users\Administrator> Get-ADObject -Filter 'ObjectClass -eq "nTDSConnection"' -SearchBase "CN=NTDS Settin
gs,CN=DC107,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=vlabs07,DC=com"

DistinguishedName
-----------------
CN=9f16b51d-6819-4934-9089-6437f69599af,CN=NTDS Settings,CN=DC107,CN=Servers,CN=Default-First-Site-Name,CN...
```

Command Breakdown:

- `Get-ADObject -Filter 'ObjectClass -eq "nTDSConnection"'`: Retrieves only replication connection objects.
- `-SearchBase`: Restricts the search to the NTDS Settings container for DC107.
- `-Properties *`: Displays all available attributes, including `fromServer`.
- `Remove-ADObject -Identity`: Deletes the selected connection object using its full distinguished name.
- `-Confirm:$false`: Bypasses the confirmation prompt.

Step 7 – Recreate the Connection Using KCC

Explanation:
In this step, we use the Knowledge Consistency Checker (KCC) to regenerate any missing connection objects after a manual deletion.
We trigger the KCC both locally and remotely to ensure consistency in replication topology.

Commands:

Trigger KCC locally (if logged into DC107)
repadmin /kcc

Trigger KCC remotely on DC107 (from another DC)
repadmin /kcc DC107

Screenshot:

```
PS C:\Users\Administrator> repadmin /kcc

Repadmin: running command /kcc against full DC localhost
Default-First-Site-Name
Current Site Options: (none)
Consistency check on localhost successful.
```

Command Breakdown:

- repadmin /kcc: Runs the KCC on the **local** domain controller to recompute replication topology.
- repadmin /kcc DC107: Forces KCC topology regeneration on the specified remote domain controller (DC107).

Verification:
List only the connection objects under DC107 that show which server each object replicates from:
```
Get-ADObject -Filter 'ObjectClass -eq "nTDSConnection"' `
-SearchBase "CN=NTDS Settings,CN=DC107,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=vlabs07,DC=com" `
-Properties fromServer
```

Look for one with `fromServer` referencing `CN=DC307,...` to confirm successful KCC regeneration.

Screenshot:

```
PS C:\Users\Administrator> Get-ADObject -Filter 'ObjectClass -eq "nTDSConnection"' -SearchBase "CN=NTDS Settin
gs,CN=DC107,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=vlabs07,DC=com" -Properties fro
mServer


DistinguishedName : CN=9f16b51d-6819-4934-9089-6437f69599af,CN=NTDS Settings,CN=DC107,CN=Servers,CN=Default-F
                    irst-Site-Name,CN=Sites,CN=Configuration,DC=vlabs07,DC=com
fromServer        : CN=NTDS Settings,CN=DC407,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration
                    ,DC=vlabs07,DC=com
Name              : 9f16b51d-6819-4934-9089-6437f69599af
ObjectClass       : nTDSConnection
ObjectGUID        : f3c63363-d903-47ef-83b3-7b0ee486d74c

DistinguishedName : CN=6f3a2ab9-39be-4408-aa8b-447312b6c60e,CN=NTDS Settings,CN=DC107,CN=Servers,CN=Default-F
                    irst-Site-Name,CN=Sites,CN=Configuration,DC=vlabs07,DC=com
fromServer        : CN=NTDS Settings,CN=DC307,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration
                    ,DC=vlabs07,DC=com
Name              : 6f3a2ab9-39be-4408-aa8b-447312b6c60e
ObjectClass       : nTDSConnection
ObjectGUID        : 4c2925a4-9b56-4f17-a52f-779cf3d787ee
```

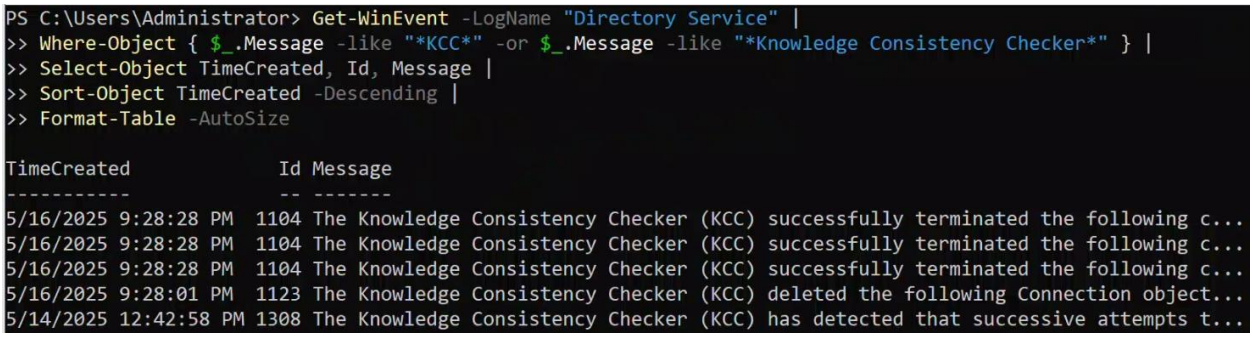
Step 8 – Verify KCC Events in Event Viewer

Explanation:
In this step, we verify that the KCC (Knowledge Consistency Checker) successfully regenerated
the connection object after deletion. We use PowerShell to filter recent KCC-related events
from the Directory Service log and confirm there are no replication or topology errors.

View recent KCC-related events in Directory Service log

```
Get-WinEvent -LogName "Directory Service" |
Where-Object { $_.Message -like "*KCC*" -or $_.Message -like "*Knowledge Consistency Checker*" } |
Select-Object TimeCreated, Id, Message |
Sort-Object TimeCreated -Descending |
Format-Table -AutoSize
```

Screenshot:



Command Breakdown:
- `Get-WinEvent -LogName "Directory Service"`: Retrieves all AD-related events from the Directory Service log.
- `Where-Object { $_.Message -like "*KCC*" -or $_.Message -like "*Knowledge Consistency Checker*" }`:
  Filters only events generated by the KCC.
- `Select-Object TimeCreated, Id, Message`: Selects relevant fields for reporting.
- `Sort-Object TimeCreated -Descending`: Shows the latest events first.
- `Format-Table -AutoSize`: Displays the table cleanly.

Task 4: Managing Notification-Based Replication

System Used: DC107

Explanation:

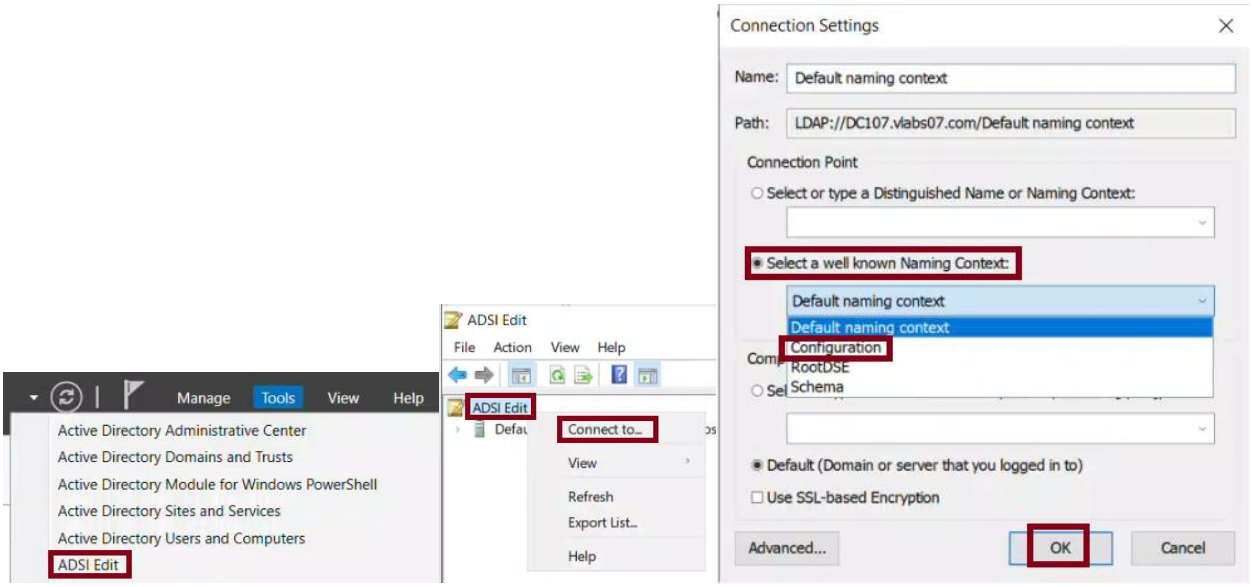
In this task, I configured Notification-Based Replication by modifying the replication
delay settings and enabling replication notifications between domain controllers.

To match the lab instructions and course slides, I completed the following:
- Used the **GUI (ADSI Edit)** to set the delay intervals on a `crossRef` object (CN=VLABS07)
- Enabled notification-based replication by setting the `options` attribute to 1 on CN=DEFAULTIPSITELINK
- Used **PowerShell** to verify that notification-based replication was enabled

GUI Steps Performed (ADSI Edit – Configuration Partition):
---

1. Opened ADSI Edit on DC107.
2. Right-clicked "ADSI Edit" (top-level node) → selected "Connect to...".
3. In the "Connection Settings" window:
   - Selected "Configuration" from the Naming Context dropdown.
   - Clicked OK.

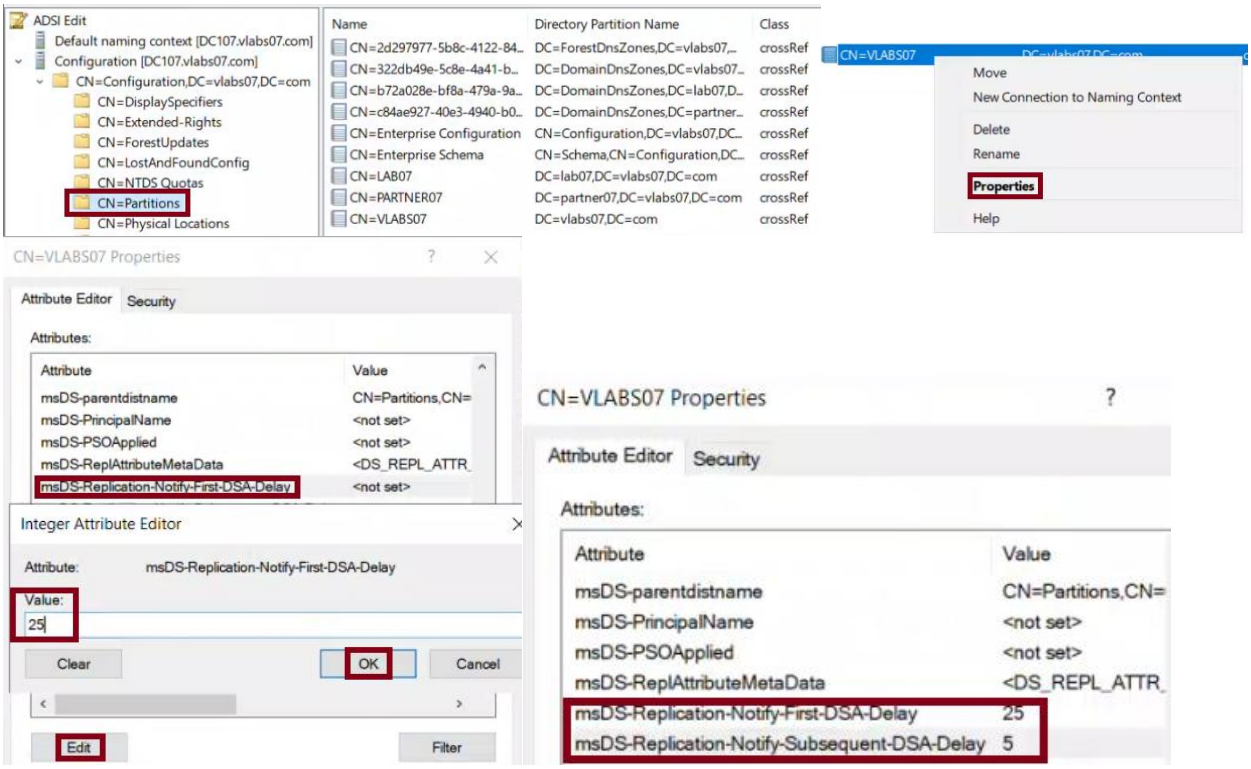
Screenshots:



◇ **Part A – Set Delay Values:**

4. Browsed to the following object:
   CN=VLABS07
   CN=Partitions
   CN=Configuration
   DC=vlabs07,DC=com
5. Right-clicked CN=VLABS07 → Properties → Attribute Editor.
6. Located and modified the following:
   - msDS-Replication-Notify-First-DSA-Delay = 25
   - msDS-Replication-Notify-Subsequent-DSA-Delay = 5

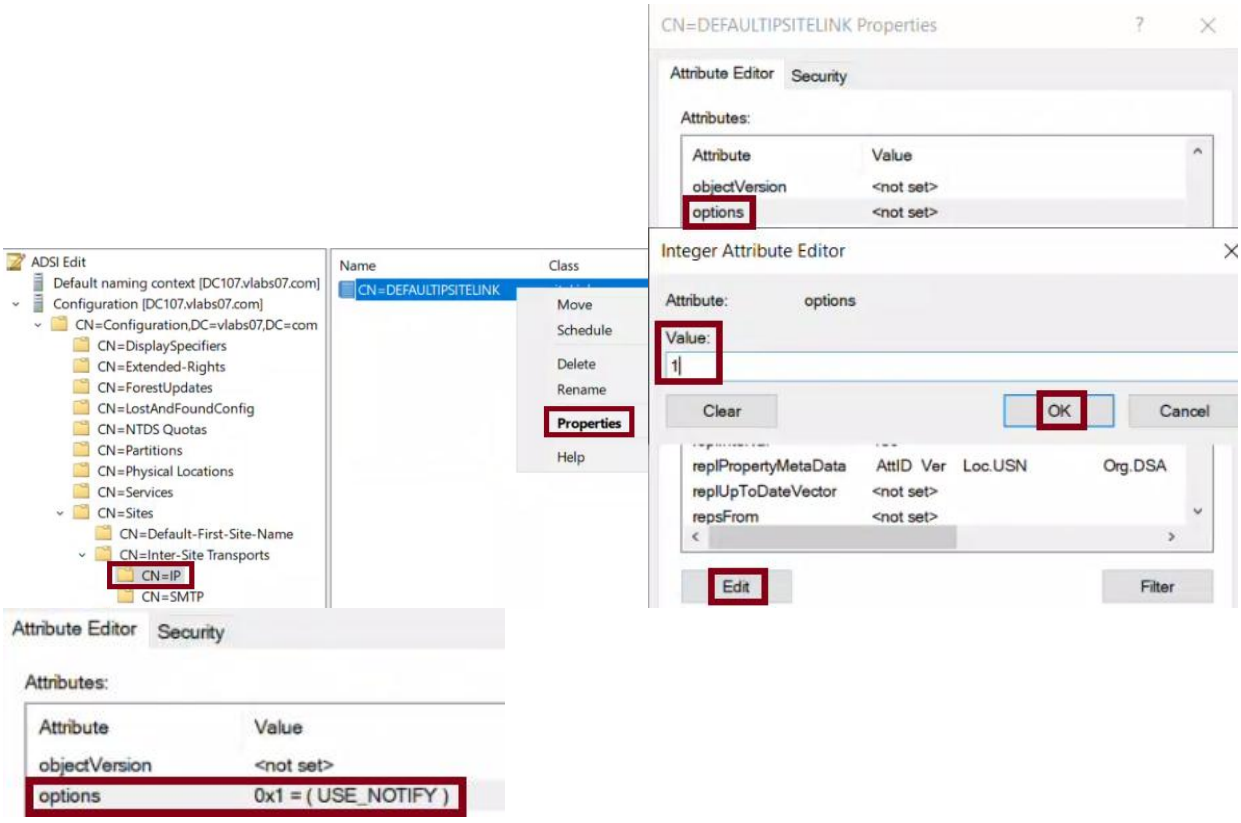
Screenshots:



◇ **Part B – Enable Notification-Based Replication:**

7. Browsed to:
   CN=DEFAULTIPSITELINK
   CN=IP
   CN=Inter-Site Transports
   CN=Sites
   CN=Configuration
   DC=vlabs07,DC=com
8. Right-clicked CN=DEFAULTIPSITELINK → Properties → Attribute Editor.
9. Located the `options` attribute and changed the value to `1` (USE_NOTIFY)

Screenshots:



Note:

The site link object (DEFAULTIPSITELINK) did not support the delay attributes due to schema restrictions.
To fulfill the GUI requirement of modifying the delays, I applied the settings to the crossRef object CN=VLABS07
under CN=Partitions, as demonstrated in the course slides.

PowerShell Verification:

```
(Get-ADObject -Filter {Name -eq "DEFAULTIPSITELINK"} `
 -SearchBase "CN=IP,CN=Inter-Site
Transports,CN=Sites,CN=Configuration,DC=vlabs07,DC=com" `
 -Properties options).options
```

Screenshot:

```
PS C:\Users\Administrator> (Get-ADObject -Filter {Name -eq "DEFAULTIPSITELINK"} -SearchBase "CN=IP,CN=Inter-Si
te Transports,CN=Sites,CN=Configuration,DC=vlabs07,DC=com" -Properties options).options
1
```

Expected Output: 1 → if Notification-Based Replication is enabled
0 → if it is not enabled
This confirms that Notification-Based Replication is enabled on the DEFAULTIPSITELINK site link.

Task 5: Creating Sites

System Used: DC107

Explanation:

In this task, I created three Active Directory Sites to reflect the company's physical layout:
- Montreal
- New-York
- Toronto

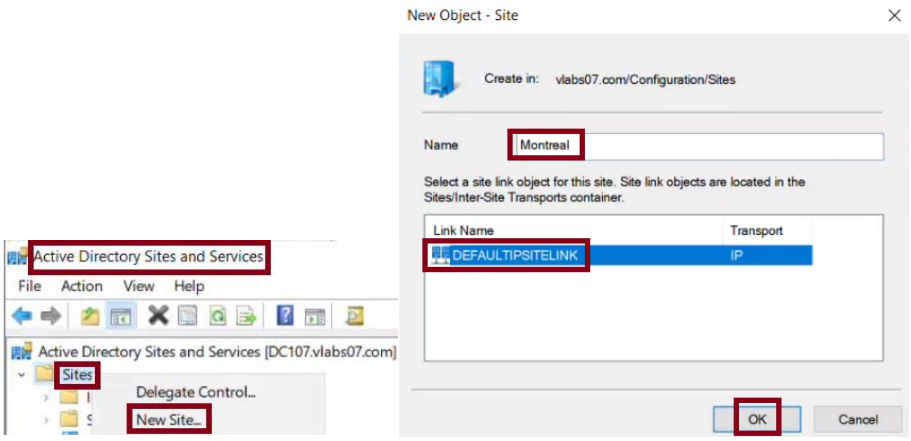
Sites help control replication traffic and optimize authentication by associating domain controllers with subnets and locations.

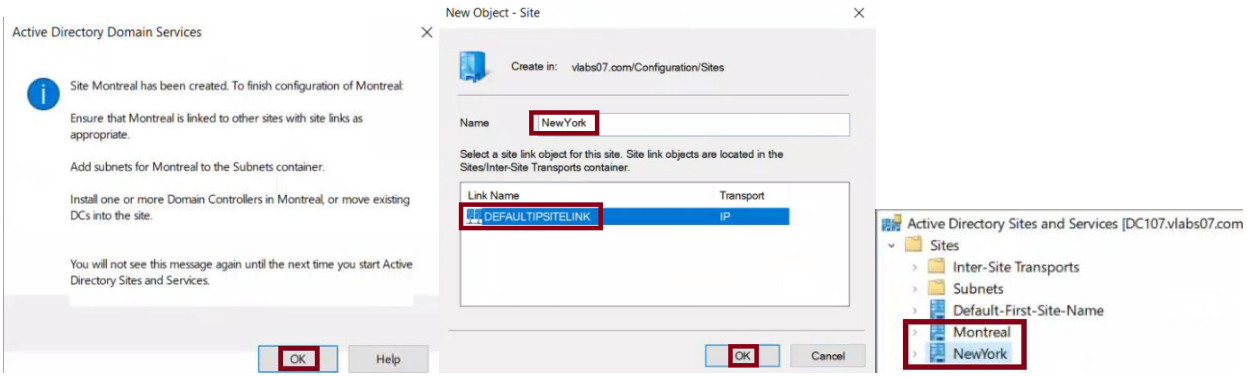
I used the **GUI (Active Directory Sites and Services)** to create and configure the Montreal and New-York sites, and used **PowerShell** to create and configure the Toronto site, as required by the lab.

GUI Steps Performed (Montreal and New-York):
---

1. Opened **Active Directory Sites and Services** from the Start menu.
2. Right-clicked on **Sites** → selected **New Site...**
3. For the first site:
   - Site Name: `Montreal`
   - Linked to site link: `DEFAULTIPSITELINK`
   - Clicked OK
4. Repeated the process to create:
   - Site Name: `New-York`
   - Linked to site link: `DEFAULTIPSITELINK`
5. Confirmed both sites were created.

Screenshots:

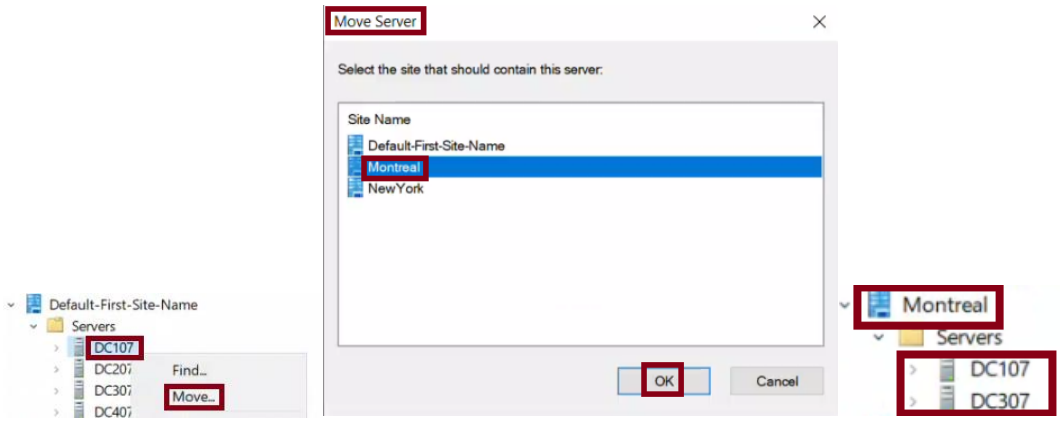




◇ **Step – Move DC107 and DC307 to the Montreal site:**

6. In **Active Directory Sites and Services**, expanded:
   Sites → Default-First-Site-Name → Servers
7. Located the server objects for `DC107` and `DC307`.
8. Right-clicked each one and selected **Move**.
9. In the **Move Server** dialog box:
   - Selected **Montreal** from the site list
   - Clicked **OK**
10. Confirmed the warning prompt if it appeared.

Screenshots:



◇ **Step – Move DC407 to the New-York site:**

11. Repeated the same process for `DC407`:
    - Right-clicked the server object under Default-First-Site-Name → Servers
    - Selected **Move**
    - Chose **New-York** as the destination site
    - Clicked OK and confirmed the prompt if needed

Screenshots:

PowerShell Steps Performed (Toronto):

◇ **Step 1 – Create the site "Toronto":**

```
New-ADReplicationSite -Name "Toronto"
```

```
PS C:\Users\Administrator> New-ADReplicationSite -Name "Toronto"
```

◇ **Command Breakdown:**

- `New-ADReplicationSite`: Cmdlet to create a new AD replication site.
- `-Name "Toronto"`: Specifies the name of the new site.

◇ **Step 2 – Verify that the site was created:**

```
Get-ADReplicationSite -Identity "Toronto"
```

```
PS C:\Users\Administrator> Get-ADReplicationSite -Identity "Toronto"


Description                     :
DistinguishedName               : CN=Toronto,CN=Sites,CN=Configuration,DC=vlabs07,DC=com
InterSiteTopologyGenerator      :
ManagedBy                       :
Name                            : Toronto
ObjectClass                     : site
ObjectGUID                      : d13602ec-f0c0-4fde-b3bf-6eb7976bd44d
ReplicationSchedule             :
UniversalGroupCachingRefreshSite :
```

◇ **Command Breakdown:**

- `Get-ADReplicationSite`: Retrieves info about an AD site.
- `-Identity "Toronto"`: Looks up the site by name.

◇ **Step 3 – Add a description to the Toronto site:**

```
Set-ADReplicationSite -Identity "Toronto" -Description "Toronto Office"
```

```
PS C:\Users\Administrator> Set-ADReplicationSite -Identity "Toronto" -Description "Toronto Office"
```

◇ **Command Breakdown:**

- `Set-ADReplicationSite`: Modifies properties of an existing site.
- `-Identity "Toronto"`: Target site.
- `-Description`: Adds or updates the site's description field.

◇ **Step 4 – Move DC207 to the Toronto site:**

```
Move-ADDirectoryServer -Identity "DC207" -Site "Toronto"
```

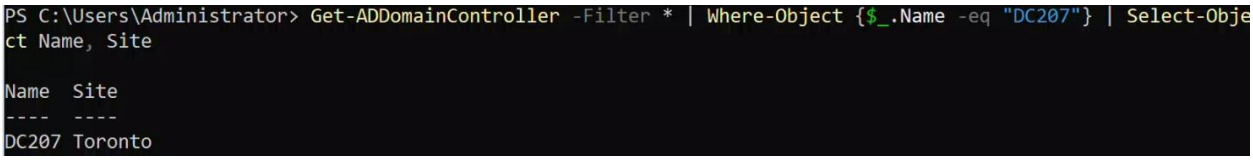
```
PS C:\Users\Administrator> Move-ADDirectoryServer -Identity "DC207" -Site "Toronto"
```

◇ **Command Breakdown:**

- `Move-ADDirectoryServer`: Moves a domain controller to a different site.
- `-Identity "DC207"`: Specifies the name of the domain controller.
- `-Site "Toronto"`: Target site.

◇ **Step 5 – Verify that DC207 was moved:**

```
Get-ADDomainController -Filter * | Where-Object {$_.Name -eq "DC207"} |
Select-Object Name, Site
```



◇ **Command Breakdown:**

- `Get-ADDomainController -Filter *`: Retrieves all domain controllers.
- `Where-Object {...}`: Filters results to show only DC2XX.
- `Select-Object`: Displays the name and associated site.

Task 6: Creating Subnets

System Used: DC107

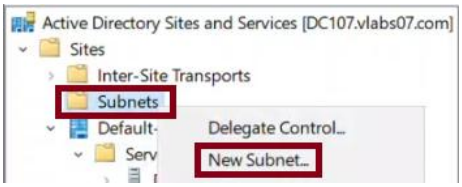
Explanation:

In this task, I created subnets and associated them with their respective Active Directory Sites.
Subnets help domain controllers and clients determine which site they belong to based on their IP address.
This is essential for logon optimization and replication efficiency.

The subnet-to-site mapping tells AD DS where specific IP address ranges (networks) are located physically.

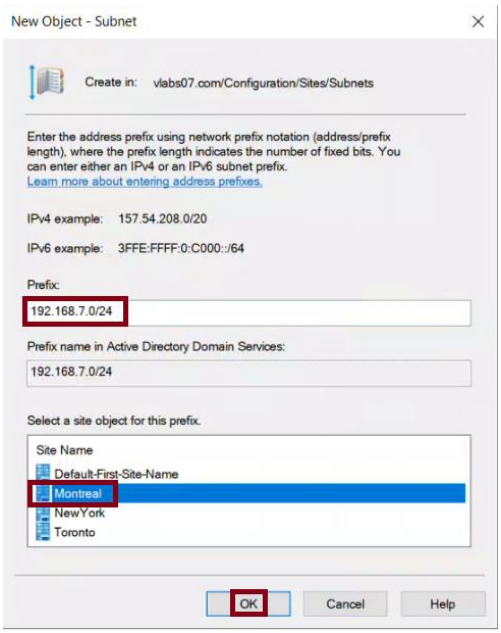
GUI Steps Performed (Montreal and New-York):
---

1. Opened **Active Directory Sites and Services**.
2. Expanded the **Subnets** container.
3. Right-clicked **Subnets** → selected **New Subnet...**



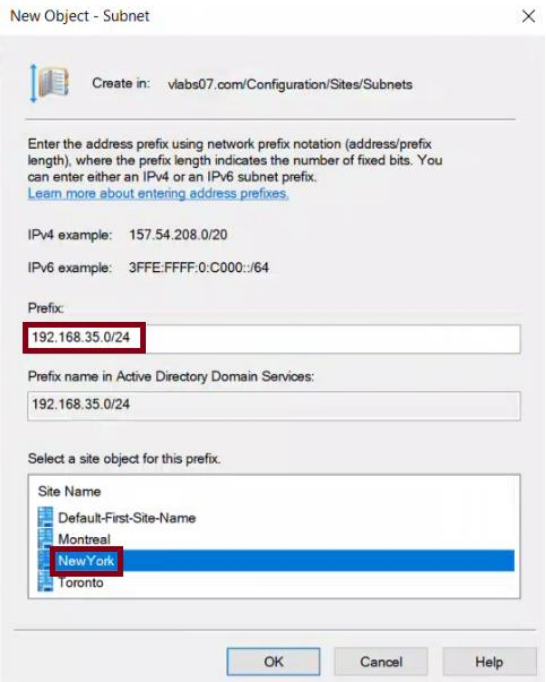
◇ **Step – Create Subnet for Montreal:**

4. In the New Subnet window:
   - Entered **192.168.7.0/24** as the subnet address
   - Selected the **Montreal** site from the dropdown list
   - Clicked OK



◇ **Step – Create Subnet for New-York:**

5. Repeated the steps above to add:
   - Subnet: **192.168.35.0/24**
   - Linked to site: **New-York**
   - Clicked OK



PowerShell Steps Performed (Toronto):

◇ **Step 1 – Create subnet for Toronto:**

```
New-ADReplicationSubnet -Name "192.168.45.0/24" -Site "Toronto"
```

```
PS C:\Users\Administrator> New-ADReplicationSubnet -Name "192.168.45.0/24" -Site "Toronto"
```

◇ **Breakdown:**

- `New-ADReplicationSubnet`: Cmdlet to create a new subnet object in AD.
- `-Name`: Subnet in CIDR notation (network address + subnet mask)
- `-Site`: The name of the site to associate the subnet with

◇ **Step 2 – Verify the subnet was created:**

```
Get-ADReplicationSubnet -Identity "192.168.45.0/24"
```

```
PS C:\Users\Administrator> Get-ADReplicationSubnet -Identity "192.168.45.0/24"


DistinguishedName : CN=192.168.45.0/24,CN=Subnets,CN=Sites,CN=Configuration,DC=vlabs07,DC=com
Location          :
Name              : 192.168.45.0/24
ObjectClass       : subnet
ObjectGUID        : 28f45e4b-771a-4bf1-adbb-eba5b0b6b23b
Site              : CN=Toronto,CN=Sites,CN=Configuration,DC=vlabs07,DC=com
```

◇ **Breakdown:**

- `Get-ADReplicationSubnet`: Retrieves subnet objects
- `-Identity`: Specifies the subnet you want to view

Task 7: Creating Site Links

System Used: DC107

Explanation:

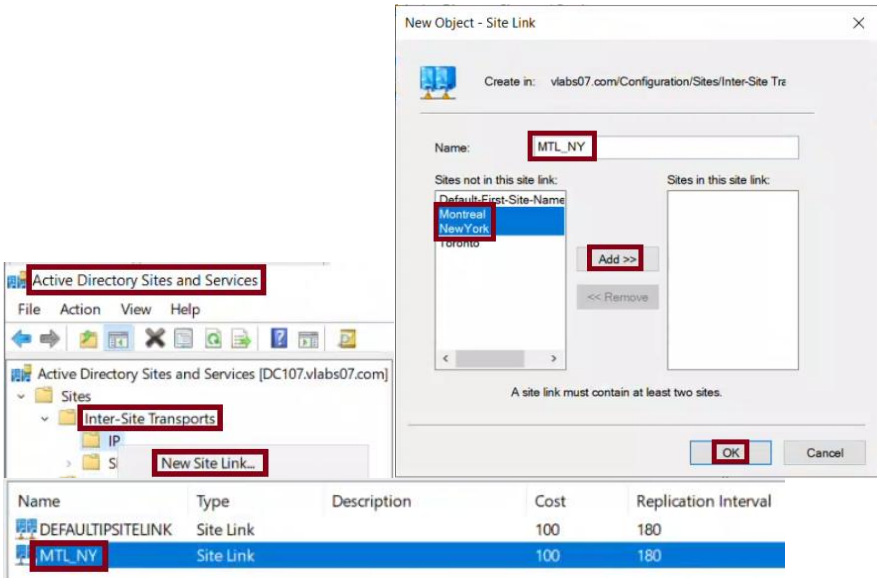
In this task, I created site links to define the replication paths and schedules between Active Directory Sites.

Site links allow Active Directory to know which sites are connected and how frequently replication should occur between them.
- I used the **GUI** to create the `MTL_NY` link (Montreal ↔ New-York)
- I used **PowerShell** to create and configure the `TOR_MTL` link (Toronto ↔ Montreal)

GUI Steps Performed (MTL_NY Site Link):
---

1. Opened **Active Directory Sites and Services**.
2. Expanded **Inter-Site Transports** → right-clicked **IP** → selected **New Site Link...**
3. In the dialog box:
   - Named the new link: `MTL_NY`
   - Selected **Montreal** and **New-York** sites from the list
   - Clicked **Add**
   - Clicked OK

Screenshot:



PowerShell Steps Performed (TOR_MTL Site Link):

◇ **Step 1 – Create site link TOR_MTL between Toronto and Montreal:**

```
New-ADReplicationSiteLink -Name "TOR_MTL" -SitesIncluded "Toronto","Montreal"
-Cost 90 -ReplicationFrequencyInMinutes 120
```

```
PS C:\Users\Administrator> New-ADReplicationSiteLink -Name "TOR_MTL" -SitesIncluded "Toronto","Montreal
 90 -ReplicationFrequencyInMinutes 120
```

◇ **Breakdown:**

- `New-ADReplicationSiteLink`: Cmdlet to create a site link
- `-Name`: Name of the site link (`TOR_MTL`)
- `-SitesIncluded`: The sites being connected
- `-Cost`: Default replication cost (lower = higher priority)
- `-ReplicationFrequencyInMinutes`: How often replication occurs (default = 180)

◇ **Step 2 – Verify creation of TOR_MTL:**

```
Get-ADReplicationSiteLink -Identity "TOR_MTL"
```

```
PS C:\Users\Administrator> Get-ADReplicationSiteLink -Identity "TOR_MTL"


Cost                          : 90
DistinguishedName             : CN=TOR_MTL,CN=IP,CN=Inter-Site
                                Transports,CN=Sites,CN=Configuration,DC=vlabs07,DC=com
Name                          : TOR_MTL
ObjectClass                   : siteLink
ObjectGUID                    : 65ff7036-b4eb-44b6-a33e-8d5da02f1890
ReplicationFrequencyInMinutes : 120
SitesIncluded                 : {CN=Toronto,CN=Sites,CN=Configuration,DC=vlabs07,DC=com,
                                CN=Montreal,CN=Sites,CN=Configuration,DC=vlabs07,DC=com}
```

◇ **Breakdown:**

- `Get-ADReplicationSiteLink`: Retrieves site link info
- `-Identity`: The name of the link to retrieve

◇ **Step 3 – Modify replication cost and interval:**

```
Set-ADReplicationSiteLink -Identity "TOR_MTL" -Cost 90 -
ReplicationFrequencyInMinutes 40
```

```
PS C:\Users\Administrator> Set-ADReplicationSiteLink -Identity "TOR_MTL" -Cost 90 -ReplicationFrequencyInMinut
es 40
```

◇ **Breakdown:**

- `Set-ADReplicationSiteLink`: Modifies an existing site link
- `-Cost`: Updates the replication priority
- `-ReplicationFrequencyInMinutes`: Updates the schedule to 40 minutes

◇ **Step 4 – Verify changes:**

```
Get-ADReplicationSiteLink -Identity "TOR_MTL" | Select-Object Name, Cost,
ReplicationFrequencyInMinutes
```

```
PS C:\Users\Administrator> Get-ADReplicationSiteLink -Identity "TOR_MTL" | Select-Object Name, Cost, Replicati
onFrequencyInMinutes

Name    Cost ReplicationFrequencyInMinutes
----    ---- -----------------------------
TOR_MTL   90                            40
```

Task 8: Creating Site Link Bridge

System Used: DC107

Explanation:

In this task, I created a **Site Link Bridge** to logically connect existing site links.
Site Link Bridges allow Active Directory to treat multiple site links as transitive, this means that if Site A is linked to Site B, and Site B to Site C, AD can infer a connection from A to C through the bridge.

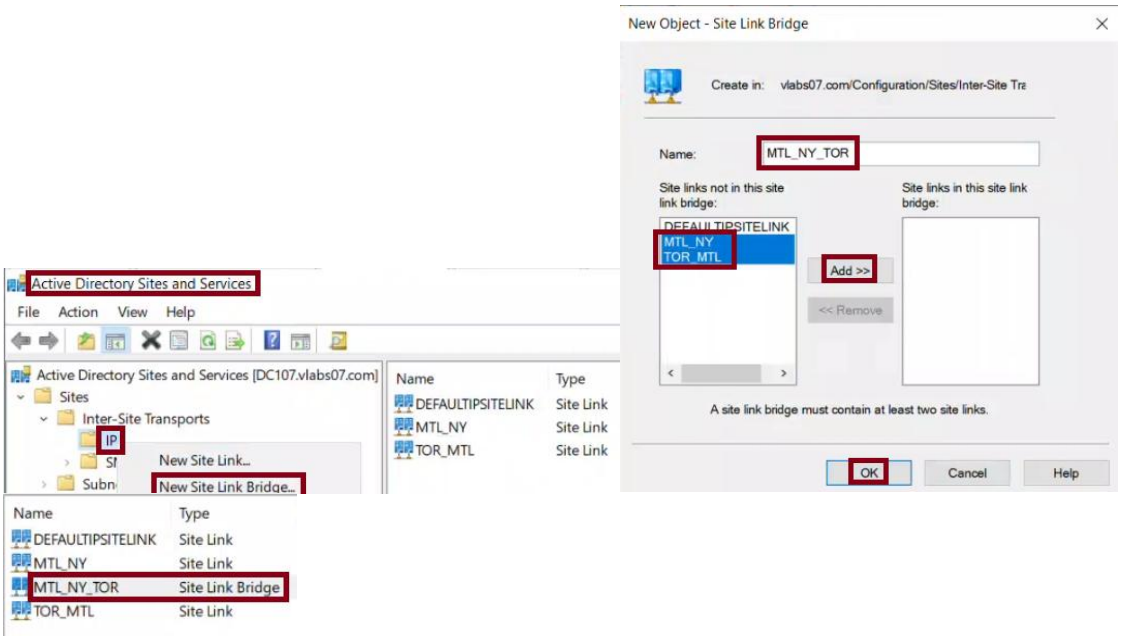
The lab required me to:
- Use the **GUI** to create a Site Link Bridge named `MTL_NY_TOR`, containing:
  - `MTL_NY` (Montreal ↔ New-York)
  - `TOR_MTL` (Toronto ↔ Montreal)

- Use **PowerShell** to verify that the bridge was created correctly.

GUI Steps Performed (Create MTL_NY_TOR Bridge):
---

1. Opened **Active Directory Sites and Services**.
2. Expanded **Inter-Site Transports** → right-clicked **IP** → selected **New Site Link Bridge...**
3. In the New Site Link Bridge window:
   - Entered **MTL_NY_TOR** as the name
   - Selected both site links: `MTL_NY` and `TOR_MTL`
   - Clicked **Add**
   - Clicked OK to complete

Screenshots:



PowerShell Verification:

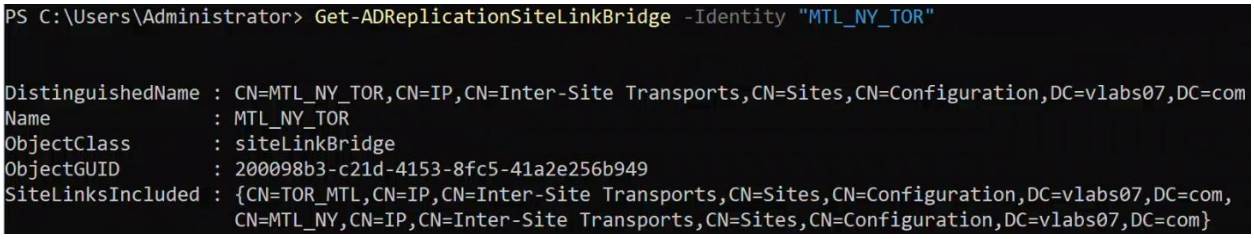
◇ **Step – Verify that the site link bridge was created:**

```
Get-ADReplicationSiteLinkBridge -Identity "MTL_NY_TOR"
```

◇ **Breakdown:**

- `Get-ADReplicationSiteLinkBridge`: Cmdlet to retrieve site link bridge objects
- `-Identity "MTL_NY_TOR"`: The name of the bridge you want to confirm

Screenshot:



Task 9: Selecting a Bridgehead Server

System Used: DC107

Explanation:

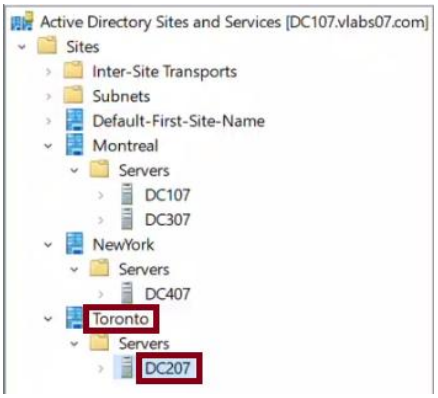
In this task, I manually assigned bridgehead servers for inter-site replication.

Bridgehead servers are responsible for replicating changes between sites. Active Directory selects them automatically by default, but in some scenarios (like controlled replication paths), administrators can assign preferred bridgehead servers manually to optimize replication paths.

The lab instructions required:
- Using the **GUI** to assign **DC4XX** as bridgehead for **Toronto**
- Using **PowerShell** to assign **DC1XX** as bridgehead for **Montreal**

Note:
In my actual lab topology, **DC207** is located in the Toronto site but is a **Read-Only Domain Controller (RODC)**, which cannot be configured as a bridgehead server.
To fulfill the lab requirement, I used **DC407** (a writable domain controller) as the bridgehead server instead, even though it appears under the **NewYork** site in the topology.
This adjustment aligns with the labs intent while respecting real-world AD behavior.

Screenshot:



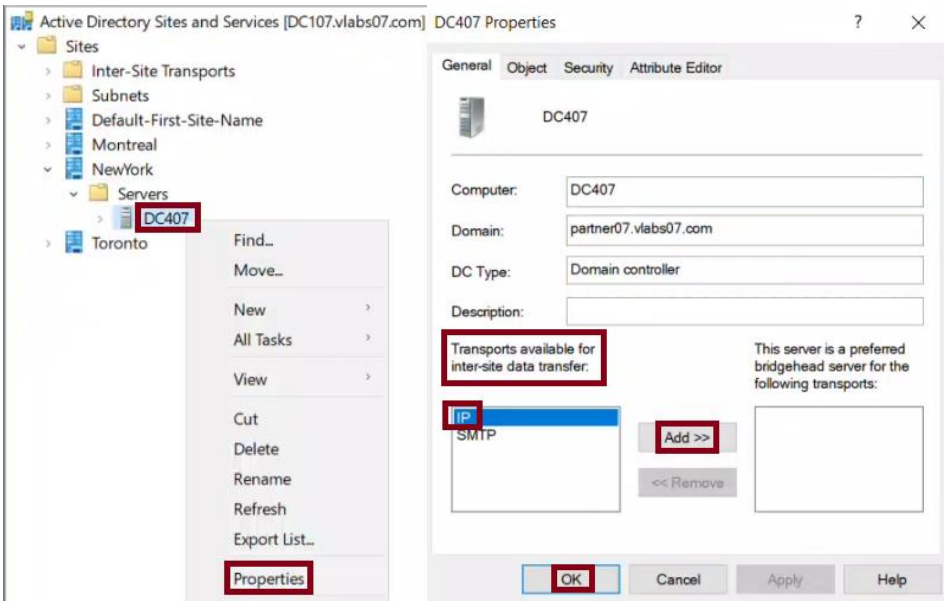
To align with my working environment, I used the correct DCs available.

GUI Steps Performed (Set DC407 as Bridgehead for Toronto):

---

1. Opened **Active Directory Sites and Services**.
2. Navigated to: `Sites → NewYork → Servers → DC407`.
3. Right-clicked the **DC407 server object itself** (not NTDS Settings) → selected **Properties**.
4. In the **Properties** window:
   - Under **Transports available for inter-site data transfer**, selected **IP**
   - Clicked **Add >>**
   - Clicked **OK** to confirm

Screenshots:



PowerShell Steps Performed (Set DC107 as Bridgehead for Montreal):

◇ **Step – Assign DC107 as preferred bridgehead using ADSI path:**

```
Set-ADObject -Identity
"CN=DC107,CN=Servers,CN=Montreal,CN=Sites,CN=Configuration,DC=vlabs07,DC=com"
`
-Add @{bridgeHeadTransportList="CN=IP,CN=Inter-Site
Transports,CN=Sites,CN=Configuration,DC=vlabs07,DC=com"}
```

```
PS C:\Users\Administrator> Set-ADObject -Identity "CN=DC107,CN=Servers,CN=Montreal,CN=Sites,CN=Configuration,DC=vlabs07,DC=com" -Add @{bridgeHeadTransportList="CN=IP,CN=Inter-Site Transports,CN=Sites,CN=Configuration,DC=vlabs07,DC=com"}
```

◇ **Breakdown:**

- `Set-ADObject`: Updates low-level AD attributes
- `-Identity`: The full distinguished name of the DC107 server object
- `bridgeHeadTransportList`: Attribute that lists preferred bridgehead transports (IP, SMTP, etc.)

◇ **Step – Verify bridgehead assignment:**

```
Get-ADObject -LDAPFilter "(bridgeheadServerListBL=*)" `
-SearchBase "CN=Sites,CN=Configuration,DC=vlabs07,DC=com" `
-Properties bridgeheadServerListBL
```

```
PS C:\Users\Administrator> Get-ADObject -LDAPFilter "(bridgeheadServerListBL=*)" -SearchBase "CN=Sites,CN=Conf
iguration,DC=vlabs07,DC=com" -Properties bridgeheadServerListBL


bridgeheadServerListBL : {CN=DC407,CN=Servers,CN=NewYork,CN=Sites,CN=Configuration,DC=vlabs07,DC=com,
                         CN=DC107,CN=Servers,CN=Montreal,CN=Sites,CN=Configuration,DC=vlabs07,DC=com}
DistinguishedName      : CN=IP,CN=Inter-Site Transports,CN=Sites,CN=Configuration,DC=vlabs07,DC=com
Name                   : IP
ObjectClass            : interSiteTransport
ObjectGUID             : 18dc0be0-c3f0-45d7-a8bf-d5f0013ac02d
```

◇ **Breakdown:**

- `Get-ADObject`: Lists objects with bridgehead servers defined
- `-LDAPFilter`: Finds objects where the bridgehead list is not empty
- `-Properties`: Displays the assigned bridgehead servers

Task 10: Managing Universal Group Membership

System Used: DC107

Explanation:

In this task, I enabled **Universal Group Membership Caching (UGMC)** on two sites.

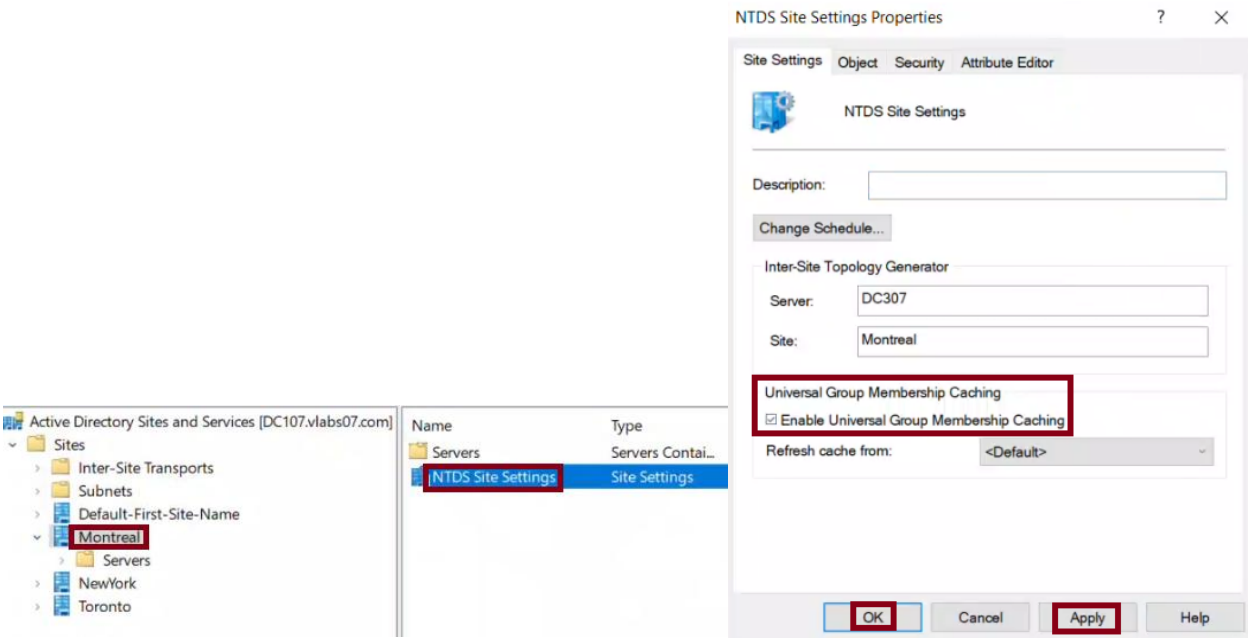
Universal Group Membership Caching allows users to log on at remote sites without contacting a global catalog server.
This improves logon performance when a GC is not available locally.

The lab instructions required:
- Use **GUI** to enable UGMC on the **Montreal** site
- Use **PowerShell** to enable UGMC on the **New-York** site

GUI Steps Performed (Enable UGMC on Montreal):
---

1. Opened **Active Directory Sites and Services**.
2. Navigated to: `Sites → Montreal`.
3. In the right pane, double-clicked **NTDS Site Settings**.
4. In the **Properties** window:
   - Checked the box: **"Enable Universal Group Membership Caching"**
   - Left **"Refresh cache from"** as default
   - Clicked **OK**

Screenshots:



PowerShell Steps Performed (Enable UGMC on New-York):

◇ **Step – Enable UGMC on New-York site:**

```
Set-ADReplicationSite -Identity "NewYork" -UniversalGroupCachingEnabled $True
```

◇ Breakdown:

- `Set-ADReplicationSite`: Modifies properties of a site
- `-Identity`: Specifies the site (New-York)
- `-UniversalGroupCachingEnabled **$True**`: Enables UGMC

◇ **Step – Verify UGMC is enabled for NewYork:**

```
Get-ADReplicationSite -Identity "NewYork" -Properties
UniversalGroupCachingEnabled | Select-Object Name,
UniversalGroupCachingEnabled
```

◇ **Breakdown:**

- `-Properties UniversalGroupCachingEnabled`: Ensures the extended property is retrieved
- `Select-Object`: Displays the `Name` and `UniversalGroupCachingEnabled` status

Task 11: Monitoring and Troubleshooting Replication

System Used: DC107

Explanation:

This task involves verifying Active Directory replication status, identifying issues, and forcing replication between DCs.

The primary tools used are:
- `repadmin` (command-line tool for AD replication diagnostics)
- PowerShell (for specific AD queries)

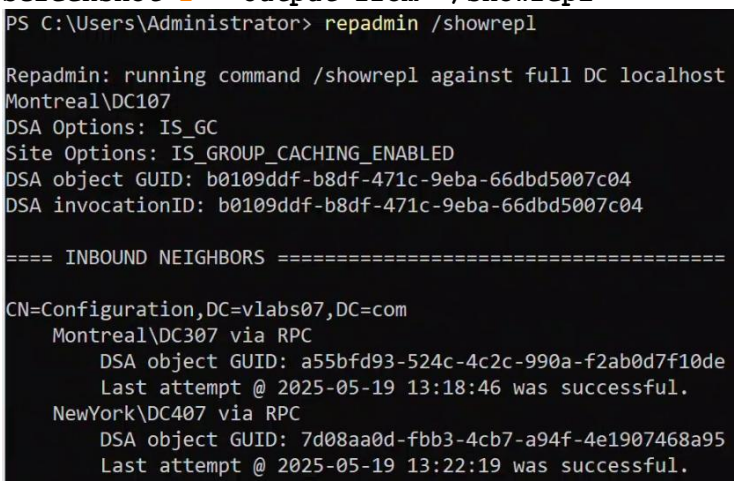
The goal is to:
1. View replication status
2. Detect any errors
3. Check replication queues
4. Trigger manual replication
5. List topology and bridgehead information

Step 1 – Check Replication Partner and Status:

```
repadmin /showrepl
```

- Shows inbound replication status for each NC on the current DC

Screenshot 1 – Output from `/showrepl`

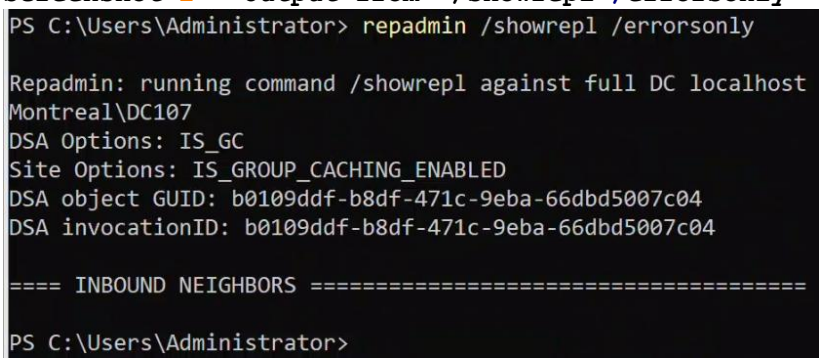


Step 2 – Check for Replication Errors Only:

```
repadmin /showrepl /errorsonly
```

- Filters `/showrepl` to only show replication partners with errors

Screenshot 2 – Output from `/showrepl /errorsonly`

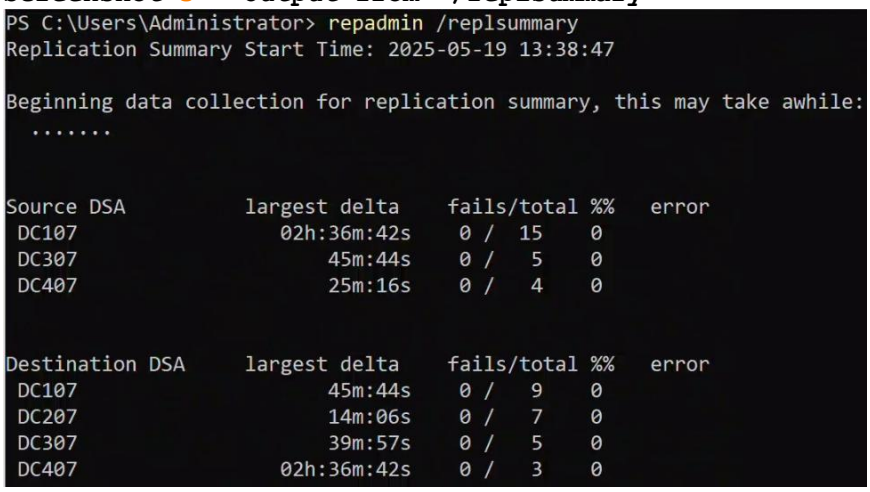


Step 3 – View Replication Summary:

```
repadmin /replsummary
```

- Displays a summary of replication status across the forest
- Useful for quick health check

Screenshot 3 – Output from `/replsummary`



Step 4 – Check Replication Status for DC207:

```
repadmin /showrepl DC207
```

- Displays partner replication status specific to DC2XX

Screenshot 4 – Replication view for DC207

```
PS C:\Users\Administrator> repadmin /showrepl DC207
Toronto\DC207
DSA Options: IS_GC DISABLE_OUTBOUND_REPL IS_RODC
Site Options: (none)
DSA object GUID: 2e1fb4df-8843-4a21-b2bc-9e6421c541f2
DSA invocationID: ec87b22d-f02d-4c93-b2c6-2b7e95497187

==== INBOUND NEIGHBORS ======================================

DC=lab07,DC=vlabs07,DC=com
    Montreal\DC107 via RPC
        DSA object GUID: b0109ddf-b8df-471c-9eba-66dbd5007c04
        Last attempt @ 2025-05-19 13:24:41 was successful.

DC=partner07,DC=vlabs07,DC=com
    Montreal\DC107 via RPC
        DSA object GUID: b0109ddf-b8df-471c-9eba-66dbd5007c04
        Last attempt @ 2025-05-19 13:24:41 was successful.
```

Step 5 – Check the Replication Queue:

```
repadmin /queue
```

- Displays any pending replication updates waiting to be processed

Screenshot 5 – Output from `/queue`

```
PS C:\Users\Administrator> repadmin /queue

Repadmin: running command /queue against full DC localhost
Queue contains 0 items.
```

Step 6 – Force Replication (Pull from DC307 to DC107):

```
repadmin /syncall DC307 /aed
```

- `/aed`: All NCs, Enterprise & Cross-Site, using *Pull* replication

Screenshot 6 – Output confirming sync completed

```
PS C:\Users\Administrator> repadmin /syncall DC307 /aed
CALLBACK MESSAGE: The following replication is in progress:
    From: CN=NTDS Settings,CN=DC407,CN=Servers,CN=NewYork,CN=Sites,CN=Configuration,DC=vlabs07,DC=com
    To  : CN=NTDS Settings,CN=DC107,CN=Servers,CN=Montreal,CN=Sites,CN=Configuration,DC=vlabs07,DC=com
CALLBACK MESSAGE: The following replication completed successfully:
    From: CN=NTDS Settings,CN=DC407,CN=Servers,CN=NewYork,CN=Sites,CN=Configuration,DC=vlabs07,DC=com
    To  : CN=NTDS Settings,CN=DC107,CN=Servers,CN=Montreal,CN=Sites,CN=Configuration,DC=vlabs07,DC=com
CALLBACK MESSAGE: The following replication is in progress:
    From: CN=NTDS Settings,CN=DC107,CN=Servers,CN=Montreal,CN=Sites,CN=Configuration,DC=vlabs07,DC=com
    To  : CN=NTDS Settings,CN=DC307,CN=Servers,CN=Montreal,CN=Sites,CN=Configuration,DC=vlabs07,DC=com
CALLBACK MESSAGE: The following replication completed successfully:
    From: CN=NTDS Settings,CN=DC107,CN=Servers,CN=Montreal,CN=Sites,CN=Configuration,DC=vlabs07,DC=com
    To  : CN=NTDS Settings,CN=DC307,CN=Servers,CN=Montreal,CN=Sites,CN=Configuration,DC=vlabs07,DC=com
CALLBACK MESSAGE: SyncAll Finished.
SyncAll terminated with no errors.
```

Step 7 – List Replication Topology:

```
repadmin /bridgeheads * /verbose
repadmin /istg * /verbose
```

- `/bridgeheads *` lists designated bridgehead servers per site
- `/istg *` lists the ISTG (Inter-Site Topology Generator) per site

Screenshot 7 – Bridgehead and ISTG details

```
PS C:\Users\Administrator> repadmin /bridgeheads * /verbose

Repadmin: running command /bridgeheads against full DC DC107.vlabs07.com
Gathering topology from site Montreal (DC107.vlabs07.com):

Bridgeheads for site Montreal (DC307.lab07.vlabs07.com):
            Source Site    Local Bridge  Trns         Fail. Time    #    Status
         ===============  ==============  ====  ================   ===  ========
                 NewYork           DC107    IP             (never)   0   The operation completed successfully.

       Naming Context         Attempt Time         Success Time  #Fail  Last Result
     ================  ===================  ===================  =====  ==============
        Configuration  2025-05-19 14:37:19  2025-05-19 14:37:19     0   The operation completed successfully.

            partner07  2025-05-19 14:08:03  2025-05-19 14:08:03     0   The operation completed successfully.

       ForestDnsZones  2025-05-19 14:08:03  2025-05-19 14:08:03     0   The operation completed successfully.

            Source Site    Local Bridge  Trns         Fail. Time    #    Status
         ===============  ==============  ====  ================   ===  ========
PS C:\Users\Administrator> repadmin /istg * /verbose

Repadmin: running command /istg against full DC DC107.vlabs07.com
Gathering topology from site Montreal (DC107.vlabs07.com):
                    Site                ISTG
      ==================   =================
 Default-First-Site-Name               DC307
                Montreal               DC307
                 NewYork               DC407

Repadmin: running command /istg against read-only DC DC207.vlabs07.com
Gathering topology from site Toronto (DC207.vlabs07.com):
                    Site                ISTG
      ==================   =================
 Default-First-Site-Name               DC307
                Montreal               DC307
                 NewYork               DC407

Repadmin: running command /istg against full DC DC307.lab07.vlabs07.com
Gathering topology from site Montreal (DC307.lab07.vlabs07.com):
                    Site                ISTG
      ==================   =================
 Default-First-Site-Name               DC307
```

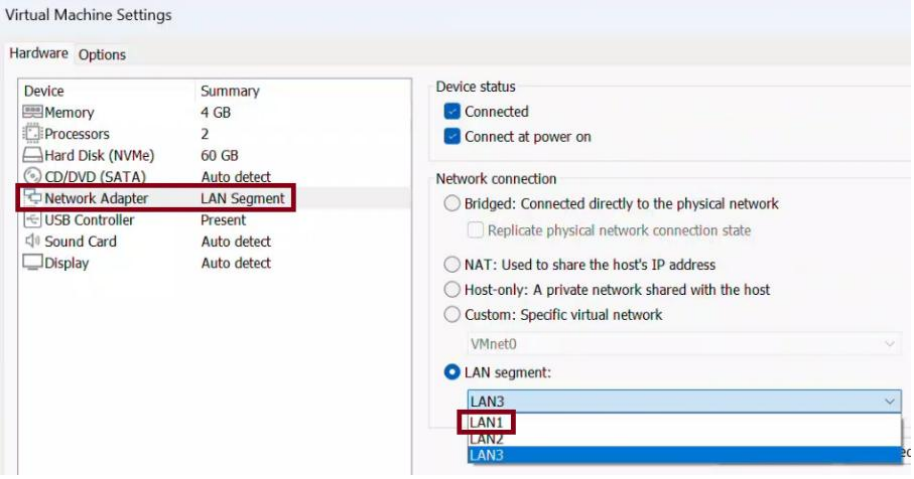
Task 12 – Managing FSMO Role and Global Catalog

Part 1 – Reconfigure DC207

System(s): DC207, DC107

Explanation:
This section prepares DC207 to rejoin the domain in the correct LAN and AD Site.
We will reconfigure the network settings, test DNS and network connectivity, and update Active Directory Sites and Services to reflect the new subnet/site association.

◇ **Step 1: Change VM LAN Segment (Performed in VMware GUI)**
Open VMware Workstation > Right-click DC207 > Settings > Network Adapter > Change to LAN1



◇ **Step 2: Set Static IP on DC207**
netsh interface ip set address name="Ethernet0" static 192.168.7.2 255.255.255.0 192.168.7.50

```
PS C:\Users\Administrator.VLABS07> netsh interface ip set address name="Ethernet0" static 192.168.7.2 255.255.
255.0 192.168.7.50
```

Breakdown:
- netsh interface ip set address: modifies IP settings for an interface
- name="Ethernet0": specifies the NIC to configure
- static: we are assigning a manual IP address
- 192.168.7.2: IP address for DC2XX on LAN1
- 255.255.255.0: subnet mask
- 192.168.7.50: default gateway (e.g., core router or DC107)

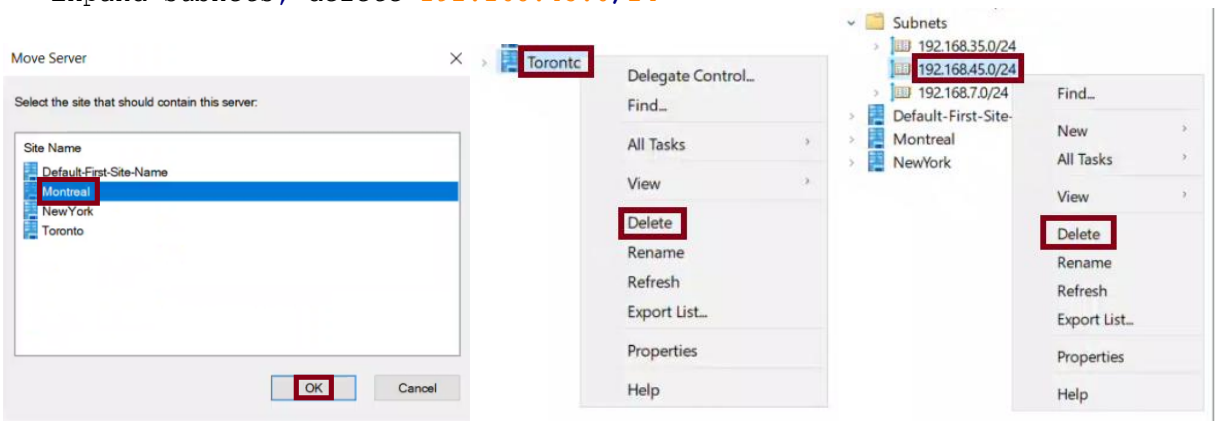
◇ Step 3: Test Network and DNS from DC207

ping 192.168.7.1
nslookup vlabs07.com

```
PS C:\Users\Administrator.VLABS07> ping 192.168.7.1

Pinging 192.168.7.1 with 32 bytes of data:
Reply from 192.168.7.1: bytes=32 time<1ms TTL=128
Reply from 192.168.7.1: bytes=32 time=1ms TTL=128
Reply from 192.168.7.1: bytes=32 time<1ms TTL=128
Reply from 192.168.7.1: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.7.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
PS C:\Users\Administrator.VLABS07> nslookup vlabs07.com
DNS request timed out.
    timeout was 2 seconds.
Server:  UnKnown
Address:  ::1

Name:    vlabs07.com
Address:  192.168.7.1
```

◇ Step 4: Move DC207 to Montreal Site (Performed in GUI on DC107)

Steps:
- Open Active Directory Sites and Services
- Drag DC207 from Toronto > Servers to Montreal > Servers
- Right-click and delete the Toronto site
- Expand Subnets, delete 192.168.45.0/24



◇ Step 5: Force Replication (Performed in GUI on DC107)

Steps:
- Go to Montreal > Servers > DC207 > NTDS Settings
- Right-click the connection object > Replicate Now

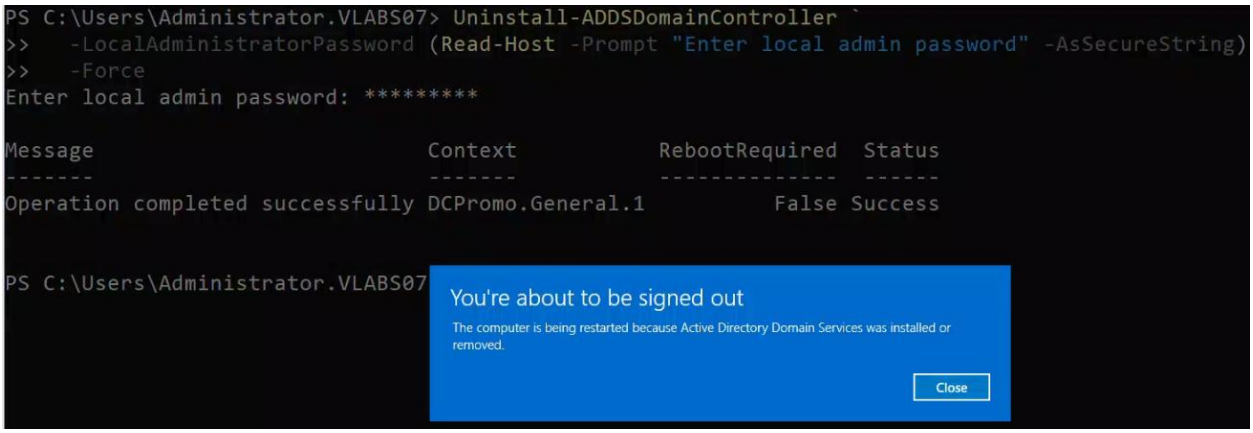
Part 2 – Demote the Domain Controller (RODC)

System: DC207

Explanation:
In this step, we safely remove DC207 from its role as a domain controller (RODC).
We will use PowerShell to demote the server and reset the local Administrator password.
After demotion, the server will reboot and return to standalone (workgroup) mode.

◇ **Step: Demote DC207 from the domain controller role**

```
Uninstall-ADDSDomainController `
  -LocalAdministratorPassword (Read-Host -Prompt "Enter local admin password" -AsSecureString) `
  -Force
```



Breakdown:
- Uninstall-ADDSDomainController: demotes the server from being a DC
- -LocalAdministratorPassword: sets the new password for the local Administrator account
- Read-Host -Prompt … -AsSecureString: prompts for secure password input
- -Force: suppresses confirmation prompts and forces demotion

Note:
- This command automatically removes DC207 from the domain.
- The system will reboot automatically.
- After reboot, log in using the local Administrator account (not the domain admin).

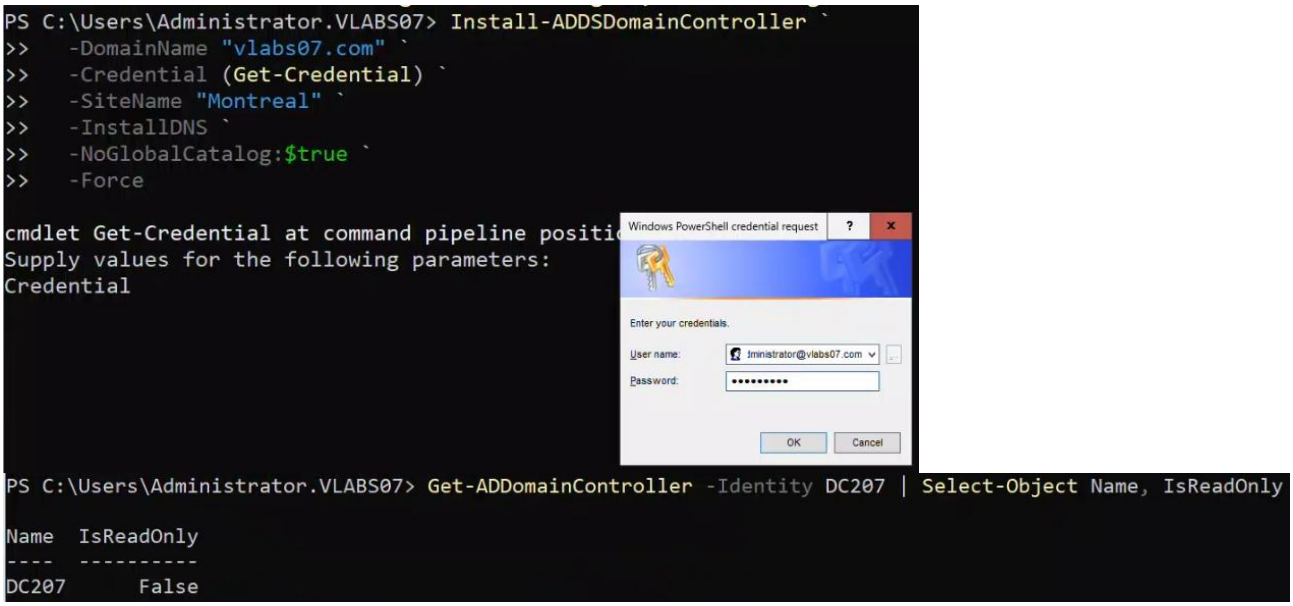
Part 3 – Promote a Writable Domain Controller (Replica)

System: DC207

Explanation:
Now that DC207 has been demoted and reconfigured, we will promote it back into the domain
as a writable domain controller in the Montreal site. During this promotion, we will disable
the Global Catalog option and ensure DNS services are installed.

◇ **Step: Promote DC207 as a writable DC in vlabs07.com domain**

```
Install-ADDSDomainController `
  -DomainName "vlabs07.com" `
  -Credential (Get-Credential) `
  -SiteName "Montreal" `
  -InstallDNS `
  -NoGlobalCatalog:$true `
  -Force
```



Breakdown:
- Install-ADDSDomainController: promotes this server as an additional DC
- -DomainName "vlabs07.com": specifies the domain to join
- -Credential (Get-Credential): prompts for domain admin credentials
- -SiteName "Montreal": places the DC in the Montreal AD site
- -InstallDNS: ensures DNS is installed if missing
- -NoGlobalCatalog:$true: disables GC role during promotion
- -Force: skips any confirmation prompts

Notes:
- This will trigger an automatic reboot after promotion.
- Make sure DC207 is on the correct network and can contact a writable DC.

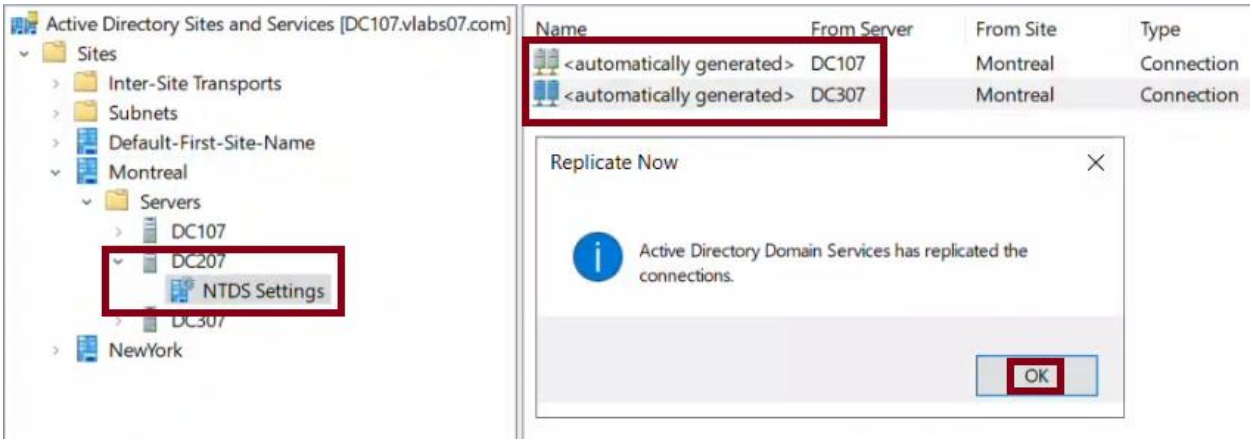
Part 4 – FSMO Role and Global Catalog Management

Systems: DC207, DC107

Explanation:
In this final part, we will:
1. Replicate DC207 with DC107.
2. Transfer the Domain Naming Master FSMO role to DC207 using PowerShell.
3. Return the FSMO role to DC107 using the GUI.
4. Simulate failure by shutting down DC107 and seizing the PDC Emulator FSMO role from DC207.
5. Configure DC207 as the PDC to sync with an external time source.
6. Enable the Global Catalog role on DC207, then disable it via GUI on DC107.

These steps complete DC207's transition into a full infrastructure-supporting domain controller.

◇ **Step 1: Force Replication from DC107 to DC207 (Performed in GUI)**
- Open AD Sites and Services > Montreal > DC207 > NTDS Settings
- Right-click connection object > Replicate Now



◇ **Step 2: View Current FSMO Role Holders**
From DC207:
netdom query fsmo

```
PS C:\Users\Administrator.VLABS07> netdom query fsmo
Schema master               DC107.vlabs07.com
Domain naming master        DC107.vlabs07.com
PDC                         DC107.vlabs07.com
RID pool manager            DC107.vlabs07.com
Infrastructure master       DC107.vlabs07.com
The command completed successfully.
```

◇ **Step 3: Transfer Domain Naming Master FSMO Role to DC207**

```
Move-ADDirectoryServerOperationMasterRole -Identity "DC207" -
OperationMasterRole DomainNamingMaster
```

```
PS C:\Users\Administrator.VLABS07> Move-ADDirectoryServerOperationMasterRole -Identity "DC207" -O
perationMasterRole DomainNamingMaster

Move Operation Master Role
Do you want to move role 'DomainNamingMaster' to server 'DC207.vlabs07.com' ?
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "Y"): Y
PS C:\Users\Administrator.VLABS07>
```

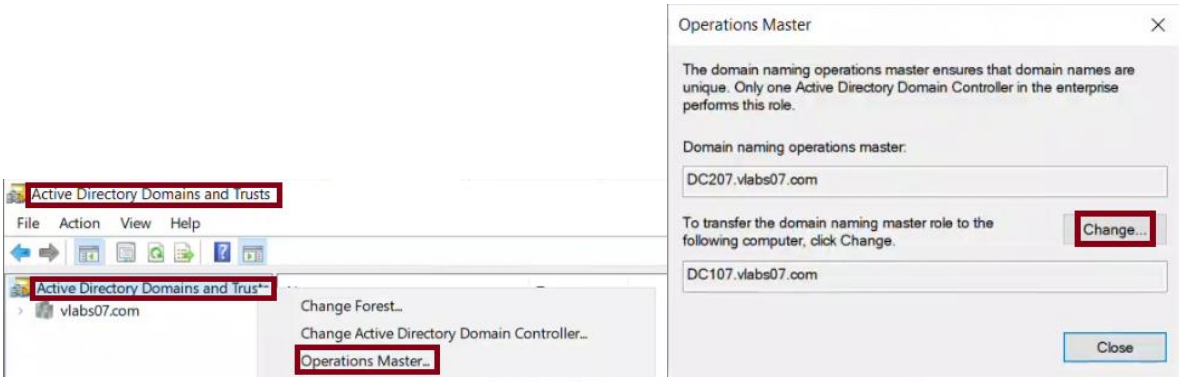
◇ **Step 4: Verify Domain Naming Master Role Transfer**

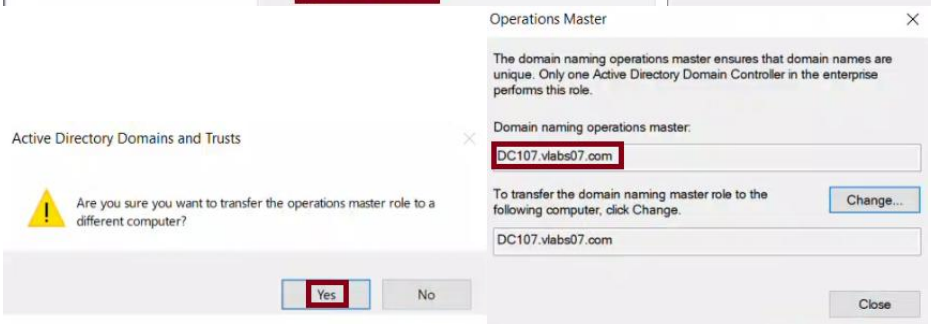
```
Get-ADForest | Format-List DomainNamingMaster
```

```
PS C:\Users\Administrator.VLABS07> Get-ADForest | Format-List DomainNamingMaster


DomainNamingMaster : DC207.vlabs07.com
```

◇ **Step 5: Return to DC107 and Transfer Role Back to DC107 (GUI)**

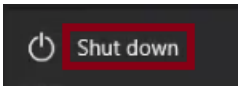
- Open Active Directory Domains and Trusts
- Right-click top node **>** Operations Master
- Click Change to **return** role to DC107





◇ **Step 6: Simulate Failure of DC107**

- Shut down DC107 **in** VMware **(do** NOT delete it**)**



◇ Step 7: Seize PDC Emulator FSMO Role on DC207

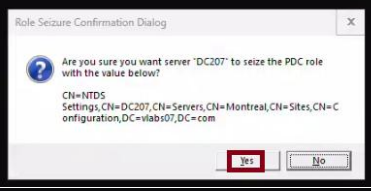
Run on DC207 as Enterprise Admin:
ntdsutil.exe

```
PS C:\Users\Administrator.VLABS07> ntdsutil.exe
C:\WINDOWS\system32\ntdsutil.exe: _
```

Inside ntdsutil:

```
> roles
> connections
> connect to server DC207
> quit
> seize pdc
> quit
> quit
```

```
C:\WINDOWS\system32\ntdsutil.exe: roles
fsmo maintenance: connections
server connections: connect to server DC207
Binding to DC207 ...
Connected to DC207 using credentials of locally logged on user.
server connections: quit
fsmo maintenance: seize PDC
```




```
ldap_modify_sW error 0x34(52 (Unavailable).
Ldap extended error message is 000020AF: SvcErr: DSID-03210901, problem 5002 (UNAVAILABLE), data
1722

Win32 error returned is 0x20af(The requested FSMO operation failed. The current FSMO holder could
 not be contacted.)
)
Depending on the error code this may indicate a connection,
ldap, or role transfer error.
Transfer of PDC FSMO failed, proceeding with seizure ...
Server "DC207" knows about 5 roles
Schema - CN=NTDS Settings,CN=DC107,CN=Servers,CN=Montreal,CN=Sites,CN=Configuration,DC=vlabs07,DC
=com
Naming Master - CN=NTDS Settings,CN=DC107,CN=Servers,CN=Montreal,CN=Sites,CN=Configuration,DC=vla
bs07,DC=com
PDC - CN=NTDS Settings,CN=DC207,CN=Servers,CN=Montreal,CN=Sites,CN=Configuration,DC=vlabs07,DC=co
m
RID - CN=NTDS Settings,CN=DC107,CN=Servers,CN=Montreal,CN=Sites,CN=Configuration,DC=vlabs07,DC=co
m
Infrastructure - CN=NTDS Settings,CN=DC107,CN=Servers,CN=Montreal,CN=Sites,CN=Configuration,DC=vl
abs07,DC=com
fsmo maintenance: _
```

```
fsmo maintenance: quit
C:\WINDOWS\system32\ntdsutil.exe: quit_
```

◇ **Step 8: Power on DC107 again**

◇ **Step 9: Verify FSMO Holder for PDC is now DC207 (GUI)**
- Open AD Users and Computers > Right-click domain > Operations Masters > PDC tab

Step 10 – Configure PDC Emulator to Synchronize Time
System: DC207 (now holding the PDC Emulator FSMO role)

Explanation:
After seizing the PDC Emulator role, we need to configure DC207 to synchronize time
with a reliable external time source. This helps ensure all domain members stay in sync.

◇ **Configure DC207 (PDC) to use manual NTP servers for time sync**
```
w32tm.exe /config /manualpeerlist:"europe.pool.ntp.org time.nist.gov
192.43.244.18 193.67.79.202" `
/syncfromflags:manual /reliable:yes /update
```

```
PS C:\Users\Administrator.VLABS07> w32tm.exe /config /manualpeerlist:"europe.pool.ntp.org time.ni
st.gov 192.43.244.18 193.67.79.202" `
>> /syncfromflags:manual /reliable:yes /update
The command completed successfully.
```

◇ **Restart Windows Time Service**
```
net stop w32time
net start w32time
```
```
PS C:\Users\Administrator.VLABS07> net stop w32time
The Windows Time service is stopping.
The Windows Time service was stopped successfully.

PS C:\Users\Administrator.VLABS07> net start w32time
The Windows Time service is starting.
The Windows Time service was started successfully.
```

Breakdown:
- /manualpeerlist: list of reliable NTP servers (can be IP or hostname)
- /syncfromflags:manual: tells the DC to sync only from the specified list
- /reliable:yes: marks this DC as a trusted time authority
- /update: applies the settings immediately

Step 11 – Enable Global Catalog on DC207

Explanation:
As part of this lab, we now enable the Global Catalog role on DC207 **using** PowerShell.
The GC holds a partial replica of all objects **in** the forest and speeds up logon and searches.

◇ **Enable the Global Catalog role for DC207**

```
Set-ADObject -Identity "CN=NTDS
Settings,CN=DC207,CN=Servers,CN=Montreal,CN=Sites,CN=Configuration,DC=vlabs07
,DC=com" `
-Replace @{options=1}
```

```
PS C:\Users\Administrator.VLABS07> Set-ADObject -Identity "CN=NTDS Settings,CN=DC207,CN=Servers,C
N=Montreal,CN=Sites,CN=Configuration,DC=vlabs07,DC=com" -Replace @{options=1}
```

Breakdown:
- Set-ADObject: updates an AD objects attributes
- -Identity: distinguished name (DN) of DC207's NTDS Settings object
- -Replace @{options=1}: sets the 'options' attribute to 1, which enables GC

You can verify with:

```
Get-ADDomainController -Identity "DC207" | Select-Object Name,
IsGlobalCatalog
```

```
PS C:\Users\Administrator.VLABS07> Get-ADDomainController -Identity "DC207" | Select-Object Name,
 IsGlobalCatalog

Name  IsGlobalCatalog
----  ---------------
DC207            True
```

◇ **Step 12: Disable Global Catalog on DC207 via GUI from DC107**
- Open AD Sites and Services > Montreal > DC207 > NTDS Settings > Properties
- Uncheck "Global Catalog"