<https://youtu.be/wS9BRmhKm50?si=5MelPjfxLrQJNOjW> 🡺 AD creation in azure vm  
<https://blog.netwrix.com/fsmo-roles#Multi-master_model_vs_Single-master_model>   
  
<https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview>   
  
  
<https://learn.microsoft.com/en-us/troubleshoot/windows-server/active-directory/fsmo-roles>   
  
<https://www.websentra.com/active-directory-guide/>   
  
 Forest: example.com

|

------------------------------------------------

| | |

Domain: us.example.com Domain: uk.example.com Domain: apac.example.com

| | |

--------------- --------------- ---------------

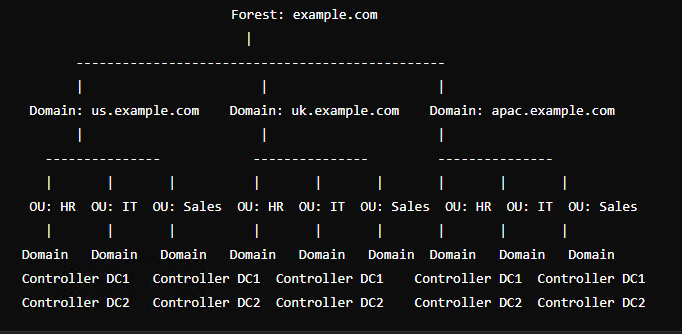
| | | | | | | | |

OU: HR OU: IT OU: Sales OU: HR OU: IT OU: Sales OU: HR OU: IT OU: Sales

| | | | | | | | |

Domain Domain Domain Domain Domain Domain Domain Domain Domain

Controller DC1 Controller DC1 Controller DC1 Controller DC1 Controller DC1

Controller DC2 Controller DC2 Controller DC2 Controller DC2 Controller DC2  
  
  
  
Sure! Let's walk through a **realistic example** of an **Active Directory (AD) Hierarchical Structure** in an organization, and I'll describe how the components fit together with some simple diagrams to make the structure clearer.

**Example Scenario: "Contoso Corporation"**

**Scenario Overview:**

* **Company Name**: Contoso Corporation
* **Domains**:
  + **corp.contoso.com** (Headquarters)
  + **east.contoso.com** (East Coast Offices)
* **Organizational Units (OUs)** for each domain:
  + Sales, HR, IT departments at each location
* **Sites**: Two sites – one for the headquarters and another for the East Coast office.

**1. Forest**

The **forest** is the highest level of the AD hierarchy. In this case, the entire Contoso Corporation exists within a single forest. The forest ensures that all domains within it share the same AD schema and can trust each other.

* **Forest Name**: contoso.com
* The **global catalog** is used to search across domains in the forest.

**2. Domains**

Each domain represents a logical boundary of administrative control and DNS namespace.

* **Domain 1**: corp.contoso.com (Headquarters, USA)
* **Domain 2**: east.contoso.com (East Coast Offices)

Each domain has its own **Domain Controllers** (DCs) responsible for authentication, authorization, and replication within that domain. However, they are part of the same forest and can share information when needed.

**3. Organizational Units (OUs)**

OUs provide a way to organize objects (users, computers, groups) within a domain and allow for delegation of administrative control. Let's break it down by domain.

* **Domain**: corp.contoso.com
  + **OU: Sales**
    - John Doe (User object)
    - Jane Smith (User object)
  + **OU: HR**
    - Alice Johnson (User object)
    - Bob Lee (User object)
  + **OU: IT**
    - Mark Green (User object)
    - Sarah White (User object)
* **Domain**: east.contoso.com
  + **OU: Sales**
    - Peter Hall (User object)
    - Linda Clark (User object)
  + **OU: HR**
    - Emily Carter (User object)
    - David King (User object)
  + **OU: IT**
    - Thomas Brown (User object)
    - Michelle White (User object)

**4. Objects**

Objects are the individual entities within an AD domain. Examples of objects include **users**, **computers**, **groups**, and **printers**. Each object has a unique identifier called a **Distinguished Name (DN)**.

* **User Object**: John Doe (in Sales OU of corp.contoso.com)
  + DN: CN=John Doe,OU=Sales,DC=corp,DC=contoso,DC=com
* **User Object**: Peter Hall (in Sales OU of east.contoso.com)
  + DN: CN=Peter Hall,OU=Sales,DC=east,DC=contoso,DC=com

**Visual Representation of AD Hierarchical Structure**

Now, let’s visualize this structure. We'll use a diagram to illustrate how the AD hierarchy is organized for Contoso Corporation.

**1. Forest Level (contoso.com)**

At the **forest level**, all domains (corp and east) exist within the same forest.

lua

Copy code

Forest: contoso.com

|

|-- Domain: corp.contoso.com

| |-- OU: Sales

| | |-- User: John Doe

| | |-- User: Jane Smith

| |-- OU: HR

| | |-- User: Alice Johnson

| | |-- User: Bob Lee

| |-- OU: IT

| |-- User: Mark Green

| |-- User: Sarah White

|

|-- Domain: east.contoso.com

|-- OU: Sales

| |-- User: Peter Hall

| |-- User: Linda Clark

|-- OU: HR

| |-- User: Emily Carter

| |-- User: David King

|-- OU: IT

|-- User: Thomas Brown

|-- User: Michelle White

**2. Domain Controllers**

Each domain has **Domain Controllers (DCs)** that authenticate and authorize users and computers within the domain. They also replicate changes to other DCs within the domain.

* **Domain: corp.contoso.com**:
  + DC1: DC1.corp.contoso.com
  + DC2: DC2.corp.contoso.com
* **Domain: east.contoso.com**:
  + DC1: DC1.east.contoso.com
  + DC2: DC2.east.contoso.com

lua

Copy code

corp.contoso.com

|

|-- DC1: DC1.corp.contoso.com

|-- DC2: DC2.corp.contoso.com

east.contoso.com

|

|-- DC1: DC1.east.contoso.com

|-- DC2: DC2.east.contoso.com

**3. Organizational Units and Objects**

Within each domain, we organize users and other objects into **Organizational Units (OUs)**. Let’s break down the OUs and the user objects within them.

lua

Copy code

Domain: corp.contoso.com

|

|-- OU: Sales

| |-- User: John Doe

| |-- User: Jane Smith

|

|-- OU: HR

| |-- User: Alice Johnson

| |-- User: Bob Lee

|

|-- OU: IT

|-- User: Mark Green

|-- User: Sarah White

Domain: east.contoso.com

|

|-- OU: Sales

| |-- User: Peter Hall

| |-- User: Linda Clark

|

|-- OU: HR

| |-- User: Emily Carter

| |-- User: David King

|

|-- OU: IT

|-- User: Thomas Brown

|-- User: Michelle White

**4. Group Policy and Delegation of Control**

Group Policies (GPOs) can be applied at the **Domain** or **OU** level to enforce settings for users and computers.

* **For example**: You could apply a password policy at the **domain level** for corp.contoso.com, and a different policy at the **OU level** for the IT department within that domain.
* **Delegation**: An administrator might delegate control of the **Sales OU** to the Sales team, allowing them to manage only the user accounts within their department, but not others.

**Real-World Application:**

Let’s imagine an employee named **John Doe**, who works in the **Sales department** at **Contoso’s headquarters** (in the domain corp.contoso.com):

1. **Authentication**: When John logs in, his credentials are verified by a domain controller in corp.contoso.com. The domain controller checks if his user object (John Doe) exists and if his password is correct.
2. **Group Policy**: If the Sales department has a specific GPO applied to it, it will enforce certain settings, such as desktop wallpaper or login restrictions.
3. **Delegation**: The HR manager can reset John’s password because they’ve been delegated control of the **Sales OU**, but they cannot change anything in the **IT OU**.

In **Active Directory**, a **Tree** is a collection of one or more domains that share a contiguous **DNS namespace**. Domains within the same tree are linked together in a hierarchical, parent-child relationship. The **tree** allows organizations to structure their domains in a way that reflects both their business structure and geographic or organizational requirements.

**Key Points:**

* **DNS Namespace**: All domains in a tree share a contiguous DNS namespace, meaning that child domains inherit the parent domain's namespace. For example, if the parent domain is techcorp.com, a child domain might be us.techcorp.com.
* **Parent-Child Relationship**: Domains in a tree are linked through trust relationships, where the parent domain trusts the child domain by default.
* **Hierarchical Structure**: The tree creates a hierarchy of domains, with one domain acting as the parent and others as child domains.

A **Tree** helps an organization structure its domains efficiently across different regions, business units, or functional divisions while maintaining a consistent naming convention.

**Example of a Tree in Active Directory**

Let’s look at a **real-world example** for a company called **GlobalTech**.

**Scenario: GlobalTech’s Domain Structure**

GlobalTech has multiple regions, such as the **US**, **Europe**, and **Asia**. The company wants to maintain a clear, logical structure for its domains, ensuring that each region has its own domain, but all domains share a common root DNS namespace.

Here’s how GlobalTech might organize its **Active Directory Tree**:

1. **Root Domain (Parent)**:
   * GlobalTech’s **root domain** is **globaltech.com**.
   * The **DNS namespace** starts with globaltech.com, which is the base for all other domains in the tree.
2. **Child Domains**:
   * GlobalTech has different regional offices, and each region gets its own **child domain**:
     + **us.globaltech.com** for the **United States** office.
     + **eu.globaltech.com** for the **European** office.
     + **asia.globaltech.com** for the **Asian** office.

Each of these domains is a **child domain** under the root domain (globaltech.com), creating a tree structure with a **contiguous namespace**.

**Active Directory Tree Structure:**

scss

Copy code

GlobalTech Tree (globaltech.com)

|

|-- Domain (globaltech.com) <-- Root Domain (Parent)

| |

| |-- Domain (us.globaltech.com) <-- Child Domain (US Office)

| |-- Domain (eu.globaltech.com) <-- Child Domain (European Office)

| |-- Domain (asia.globaltech.com) <-- Child Domain (Asian Office)

|

|-- Trusts: Parent-child trusts automatically established.

**Explanation of the Structure:**

1. **Root Domain (globaltech.com)**:
   * This is the top-level domain of the tree. All objects (users, groups, computers) in GlobalTech are initially placed in this domain, or in one of the child domains.
   * The root domain defines the **root of the DNS namespace** for the organization.
2. **Child Domains (us.globaltech.com, eu.globaltech.com, asia.globaltech.com)**:
   * These child domains are part of the same **DNS namespace** (globaltech.com), but each region has its own domain for better management and administration.
   * The child domains inherit the namespace from the root domain but can be managed independently.
   * For example, **us.globaltech.com** can have its own set of domain controllers, users, groups, and policies, but it's still part of the **globaltech.com** namespace.
3. **Trust Relationship**:
   * **Implicit Trusts**: All child domains automatically trust the parent domain (globaltech.com), and each child domain trusts the other child domains. For example, **us.globaltech.com** trusts **eu.globaltech.com**, **asia.globaltech.com**, and vice versa, without needing explicit trust configurations.
   * This allows users in **us.globaltech.com** to access resources in **eu.globaltech.com** or **asia.globaltech.com**, assuming proper permissions are granted.
4. **Group Policy Application**:
   * Group policies can be applied at different levels in the tree. For example:
     + **Root Domain (globaltech.com)**: A policy to set password complexity rules for the entire organization.
     + **Child Domain (us.globaltech.com)**: A policy to apply specific security settings for the U.S. office.
     + **Specific OUs**: Additional policies can be applied to Organizational Units (OUs) within any of the domains to further refine security or configuration settings.

**Key Features of an Active Directory Tree:**

* **Contiguous DNS Namespace**: All domains in the tree share the same top-level DNS name (globaltech.com in this case), creating a unified naming convention across regions.
* **Hierarchical Domain Structure**: The parent-child relationship means that administrative authority is delegated from the root domain to child domains. This allows for centralized management (at the root) or decentralized management (at the child domain level).
* **Trusts**: Trusts between domains in the tree are automatically created, allowing users in any domain to authenticate and access resources across the entire tree.
* **Scalability**: The tree structure is scalable, as more child domains can be added as the organization grows (for example, adding new regions or business units).

**Visual Representation of the Tree:**

scss

Copy code

GlobalTech Tree (globaltech.com)

|

|-- globaltech.com (Root Domain)

| |

| |-- us.globaltech.com (Child Domain - U.S.)

| |-- eu.globaltech.com (Child Domain - Europe)

| |-- asia.globaltech.com (Child Domain - Asia)

**Example of an Expanded Tree:**

As GlobalTech continues to expand, they could add more child domains to reflect further subdivisions of their business. For instance, the European region might have different business units that require separate domains, such as:

scss

Copy code

GlobalTech Tree (globaltech.com)

|

|-- globaltech.com (Root Domain)

| |

| |-- us.globaltech.com (Child Domain - U.S.)

| |-- eu.globaltech.com (Child Domain - Europe)

| | |

| | |-- uk.eu.globaltech.com (Child Domain - U.K.)

| | |-- de.eu.globaltech.com (Child Domain - Germany)

| |-- asia.globaltech.com (Child Domain - Asia)

In this example:

* The **European child domain** eu.globaltech.com is further subdivided into **UK** and **Germany** domains (uk.eu.globaltech.com, de.eu.globaltech.com).
* This allows for further specialization and management of resources at a more granular level.

🡺After creation of windows server u have made the private IP address Static by network settings 🡺 ip configs 🡺 make it static…  
  
🡺 Then login to VM 🡺 go to control panel🡺 network settings 🡺 network adapter settings 🡺 ethernet 🡺 properties ==Ipv4 configs🡺 make it preferred one for ip address and DNS settings.  
  
A computer screen shot of a computer

Description automatically generated  
  
  
A computer screen shot of a black screen

Description automatically generated  
  
  
A screenshot of a computer screen

Description automatically generated  
  
🡺 This is AD-Integrated DNS.. Then install ADDS & DNS SERVER ROLES ON VM and promote it as Domain controller.  
  
  
  
  
  
  
  
  
🡺 we got to face an issue   
  
A screenshot of a computer

Description automatically generated  
  
The output from nslookup exportss.com still shows that your **Domain Controller** (with IP 10.0.0.10) is resolving the domain exportss.com correctly, but it also indicates that the **DNS server** responding to the query is shadowworld.internal.cloudapp.net (instead of dc1.exportss.com or whatever your Domain Controller's FQDN is).

This suggests that your **hostname** has not been fully aligned with the expected Domain Controller name in your domain, or there might still be a mismatch in the system’s DNS settings.

**Possible Causes**

1. **Hostname Change Wasn't Applied**: It’s possible that the VM was not properly renamed, or that the name change hasn't propagated properly within Active Directory.
2. **DNS Server Records Not Updated**: The DNS records for dc1.exportss.com might not have been created or updated automatically in the DNS system.
3. **Cached DNS Entries**: There could be DNS caching happening on your system, preventing it from properly resolving to the new hostname.

**Steps to Address the Issue**

**Step 1: Verify Hostname**

Let's make sure the **hostname** of the server matches what you expect.

1. Log in to your Domain Controller.
2. Open **PowerShell** or **Command Prompt**.
3. Run the following command to check the current hostname of the VM:

powershell

Copy code

hostname

* + If it returns something like shadowworld or a different name from what you expect (e.g., dc1), you need to rename it properly.

**Step 2: Rename the Domain Controller (If Needed)**

If the hostname is incorrect or still set to the default (shadowworld), you need to rename the Domain Controller to align with the **domain's name** (e.g., dc1.exportss.com).

1. Run this PowerShell command to rename the server:

powershell

Copy code

Rename-Computer -NewName dc1 -Restart

This will rename the server to dc1, and it will restart the machine.

* + If you choose a different name (e.g., dc1), replace dc1 with your desired name. Just make sure it reflects the **FQDN** you'd like for your Domain Controller (dc1.exportss.com).

1. After the restart, you can run hostname again to ensure that the new name (dc1) is applied.

**Step 3: Update DNS Records**

After renaming the server, the DNS records should be automatically updated in Active Directory-integrated DNS. However, it's a good idea to double-check that everything is configured correctly.

1. **Open DNS Manager** on the Domain Controller:
   * Go to **Server Manager** > **Tools** > **DNS**.
2. **Verify the A (Host) Record**:
   * In the **Forward Lookup Zones** section, find your **domain** (exportss.com).
   * Check if there's an **A record** for dc1 (or whatever the name of the Domain Controller is).
     + The record should resolve dc1.exportss.com to the IP address 10.0.0.10.
3. **Verify SRV Records**:
   * Make sure the **SRV records** (e.g., \_ldap.\_tcp.dc.\_msdcs.exportss.com) are present for the Domain Controller (dc1.exportss.com).

**Step 4: Flush DNS Cache**

Sometimes, your local machine or other clients might cache DNS information. To make sure the changes propagate correctly, you should clear the DNS cache:

1. On your Domain Controller or any machine having DNS resolution issues, run the following command in **Command Prompt** or **PowerShell**:

powershell

Copy code

ipconfig /flushdns

1. **Restart the DNS Client Service** (optional): If flushing the DNS cache doesn't resolve the issue, you can restart the DNS client service to ensure fresh DNS resolution:

powershell

Copy code

Restart-Service -Name dnscache

**Step 5: Verify DNS Resolution Again**

After performing the steps above, run the nslookup command again to verify that everything is working correctly:

bash

Copy code

nslookup exportss.com

* This time, the **DNS server** should be reported as dc1.exportss.com or the appropriate FQDN of your Domain Controller, and the domain should resolve to 10.0.0.10.

The output should look like this:

makefile

Copy code

Server: dc1.exportss.com

Address: 10.0.0.10

Name: exportss.com

Address: 10.0.0.10  
  
🡺After restarting the shado.exp.com(right click on it 🡺 all tasks 🡺 restart from DNS) it worked out.  
  
A screenshot of a computer

Description automatically generated  
  
A screenshot of a computer

Description automatically generated  
  
🡺 Now I want replicate this domain controller with same domain.  
  
🡺 After creating another Azure vm I had made network changes in this.  
  
A screenshot of a computer

Description automatically generated  
  
A computer screen shot of a black screen

Description automatically generated  
  
🡺 we got to see other issue  
A screenshot of a computer

Description automatically generated  
  
  
  
  
  
  
  
  
🡺for this we Disable NLA Authentication from azure portal 🡺 & we are able to login to server..  
  
  
A screenshot of a computer

Description automatically generated  
  
🡺 we can do it before from server itself   
  
A screenshot of a computer

Description automatically generated