Active Directory Domain Services (AD DS) Explained

Active Directory Domain Services (AD DS) is the core server role in **Microsoft's Active Directory (AD)** that centrally manages and stores information about **network resources**, such as users, computers, and devices, within a domain. It is a fundamental component of the Windows Server operating system environment.

…….

Key Functions and Benefits

AD DS provides several crucial functions and benefits for an organization's IT infrastructure:

- **Centralized Identity and Access Management (IAM):** AD DS serves as a centralized point of administration, allowing IT teams to manage user accounts, passwords, and security policies from a single location.
- **Authentication and Authorization:** It is responsible for **authenticating** users (verifying who they are, typically via user ID and password) and **authorizing** their access (determining what resources they are allowed to use) on the network.
- **Single Sign-On (SSO):** Users can log in once to their machine and seamlessly access various network resources (files, printers, applications) for which they are authorized without needing to enter credentials repeatedly.
- **Policy-Based Administration:** Administrators can use **Group Policy** to apply configuration settings and security restrictions consistently to groups of users or computers across the entire domain.
- **Hierarchical Structure:** It organizes resources into a logical, hierarchical structure of **Domains**, **Trees**, and **Forests**, making management and scalability easier.

  ………
…

Core Components and Concepts

AD DS relies on several established protocols and components:

- **Domain Controller (DC):** A server that runs the AD DS role and stores a copy of the AD directory database. DCs are responsible for responding to authentication requests and replicating directory data to other DCs.
- **Domain:** A logical group of objects (users, computers) that share the same AD database and security policies.
- **Forest:** The highest-level logical container, which is a collection of one or more domain trees that share a common schema (set of rules for objects) and a single global catalog.
- **Schema:** The blueprint or set of rules that defines the classes of objects (like 'user' or 'computer') and the attributes they can have in the AD forest.

- **Global Catalog:** A Domain Controller that stores a partial, searchable copy of all objects in the entire forest, allowing users to find objects regardless of which domain in the forest they reside.
- **Protocols**: AD DS relies on protocols like LDAP (Lightweight Directory Access Protocol) for directory querying, Kerberos for secure authentication, and DNS (Domain Name System) for naming and location services.

Defining the Active Directory Domain

A Domain is a **logical group** of network objects (users, computers, groups, printers, etc.) that share three critical things:

1. **A Single Central Database:** All objects within the Domain are stored in a single, common Active Directory database (the directory).
2. **A Common Security Authority:** The Domain is the **security boundary**. Users are authenticated and authorized *by* the Domain, and security policies (Group Policies) are applied *to* the Domain.
3. **A Single Administrative Authority**: The Domain is managed by a specific group of administrators (Domain Admins). They have full control over all objects and settings within that Domain.

How Domains are Managed

- **Domain Controllers (DCs):** The servers that run Active Directory Domain Services (AD DS) and hold the complete, writable copy of the Domain database are called **Domain Controllers**. Every Domain must have at least one DC, but usually has two or more for redundancy.
- **Replication:** All DCs in a single Domain participate in **multi-master replication**, meaning any change made on one DC (like a password update or adding a user) is copied to all other DCs in that same Domain

Domain's Place in the AD Hierarchy

The Active Directory hierarchy organizes resources from the smallest administrative unit (Objects) to the largest security boundary (Forest). The Domain serves as the central management unit within this structure.

1. Below the Domain (Sub-Containers)

These containers exist *inside* a single Domain and are used for granular management:

- **Objects:** These are the smallest logical entities in AD. They represent real-world resources like **users**, **computers**, and **groups**.
- **Organizational Units (OUs):** OUs are containers *within* a Domain used to organize objects into logical groupings (e.g., OUs for "HR," "Sales," or "Servers

Key Purposes of OUs

OUs are essential because they serve three crucial administrative purposes that the Domain itself cannot efficiently handle alone:

1. Delegation of Administration 🧑💻

OUs allow administrators to grant specific administrative rights (e.g., reset passwords, create users, modify group memberships) over a defined set of objects **without** giving them control over the entire Domain.

- **Example:** You can delegate control of the "Sales OU" to the Sales team lead, allowing them to manage only the user accounts within their department, while restricting them from touching any accounts in the "IT OU" or "Finance OU."

2. Applying Group Policy 🔐

OUs are the **recommended level** to link **Group Policy Objects (GPOs)**. Policies applied to an OU only affect the objects inside that OU, giving administrators precise control over settings.

- **Example:** You can link a GPO to the "Student Computers OU" to restrict software installation and enforce a specific desktop background, while applying a different, less restrictive GPO to the "Teacher Computers OU."

3**. Organization and Visibility** 📂

OUs provide a logical hierarchy for all objects, making the directory easy to navigate, search, and manage for both administrators and applications.

Type of object

1. User Accounts

- **Definition:** Represents a human user (person) who needs access to the network resources.

- . **Function**: Used for authentication (verifying the user's identity via username and password) and establishing the user's authorization rights.
- **Key Security Role**: The User SID is included in the user's security token upon successful logon, granting access to resources based on the permissions assigned to the user (though permissions are usually assigned to groups)

Computer Accounts

- **Definition:** Represents a physical or virtual machine (workstation, laptop, or server) that has been **joined** to the Active Directory Domain.
- **Function:** Computers must also be authenticated by the Domain (using a machine account password) before they can receive Group Policy settings or access network resources.
- **Key Security Role:** The Computer SID allows you to grant permissions or apply policies specifically to a machine (e.g., "Only this server can access the Financial database"). This is crucial for security.

Groups

- **Definition:** A collection of User and Computer accounts organized together. Groups are used as an administrative shortcut.
- **Function:** Simplifies management by allowing administrators to assign permissions or apply Group Policies once to the group, rather than individually to hundreds of users.
- **Key Security Role:** The Group's SID is also added to the user's security token upon logon. The user then inherits all the permissions granted to that group. **This is the recommended way to assign permissions in AD.**

Domain Hierarchy (Logical Structure)

These terms describe the structural relationship of a Domain within the entire AD Forest:

Term	**Classification**	**Description**
Root Domain	**Domain Hierarchy**	The **first** Domain created in the Forest. It defines the name of the entire Forest (e.g., company.com) and is the foundation for all other Trusts.

Child Domain	**Domain Hierarchy**	A Domain created beneath the Root or another Parent Domain. It shares a contiguous namespace (e.g., sales.company.com is a child of company.com).
Parent Domain	**Domain Hierarchy**	A Domain that has a Child Domain directly beneath it in the naming structure. (e.g., company.com is the parent of sales.company.com).

Domain Controller Roles (Server Function)

These terms describe the specific function of a Domain Controller server:

Term	**Classification**	**Description**
Read-Only Domain Controller (RODC)	**DC Role**	A special type of Domain Controller that holds a **non-writable** copy of the Active Directory database. It is often deployed in remote branch offices where physical security is a concern.
Primary	**Historical/Obsolete Term**	Short for **Primary Domain Controller (PDC)**. This term is **obsolete** in modern Active Directory (Windows 2000 and later), which uses **multi-master replication** where all standard DCs are equal.

The tree

The **Tree** is the second-highest level in the Active Directory (AD) logical hierarchy, sitting above the Domain but below the Forest.

🌳 The Active Directory Tree

A Tree is defined by two key characteristics:

1. Shared Contiguous DNS Namespace

All Domains within a single Tree must share a **contiguous DNS namespace**. This means that the names of all Domains in the Tree are derived from the same root Domain name.

- **Example:**
  - **Root Domain (Parent):** company.com
  - **Child Domain 1:** sales.company.com
  - **Child Domain 2:** hr.company.com

  All three of these Domains belong to the same Tree because they share the .company.com namespace.

The forest

The **Forest** is the highest logical structure in Active Directory Domain Services (AD DS). It represents the absolute top boundary for security, directory data, and administration.

If the Domain is the single building and the Tree is the campus, the **Forest is the entire corporation**.

🌳 Defining the Active Directory Forest

A Forest is defined as a collection of one or more **Trees** (or Domains) that share three essential components

Tree vs. Forest

- A **single-Domain Forest** is technically both a **Domain** and a **Tree**.
- When you add a second Domain with a completely different namespace (e.g., you create newcompany.net in the same structure as company.com), you are creating a **second Tree** within the same **Forest**.

Hierarchy Level	Contains	Key Boundary/Shared Element
Forest (Highest)	Multiple Trees/Domains	**Single Schema** and **Global Catalog** and **one DNS**
Tree (Mid-Level)	Multiple Domains	Contiguous DNS Namespace (Trust Boundary).

Domain (Core)	OUs and Objects	Single AD Database (Administration/Replication

Requirements and prerequisites to successfully install your first Domain Controller

I. Server Prerequisites

You must have a physical or virtual server running an appropriate version of Windows Server ready for promotion.

1. **Operating System (OS):** The server must be running a version of **Windows Server** (e.g., Server 2019, Server 2022). It cannot be a client OS like Windows 10 or 11.
2. **Edition:** The server must be installed as the **Standard** or **Datacenter** edition. (The Essentials edition has domain limitations.)
3. **Local Account:** You must be logged in with a **Local Administrator** account on the server to begin the installation.
4. **No Existing Roles:** The server should be a **clean installation** with no conflicting roles already installed, especially no existing Active Directory Domain Services (AD DS) components.

II. Network Requirements

The Domain relies heavily on stable, correctly configured networking, especially DNS.

1. **Static IP Address:** The server that will become the Domain Controller (DC) **must** have a **static IP address** configured. DCs cannot use DHCP.
2. **DNS Configuration:**
    - The primary component of Active Directory is **DNS (Domain Name System)**. The server must be able to resolve DNS names.
    - During the installation, you must ensure that **DNS Server Role** is installed (or selected for installation) on the DC.
3. **Domain Name:** You need to choose the **Fully Qualified Domain Name (FQDN)** for your Domain (e.g., contoso.local or corp.example.com). This name must be unique and should not conflict with any external names.