Windows Server

User

- A user in Windows Server refers to an account that is created to allow a person or system process to log in and access system resources.
- Users can have different permissions based on their roles.
- Types of users:
  - **Local User**: A user account that exists only on a single computer or server.
  - **Domain User**: A user account that is managed by Active Directory (AD) and can log into any system within the domain.
  - **Built-in Users**: Accounts like Administrator and Guest, which are default accounts in Windows Server.

Users Profile

- A User Profile is a collection of settings and data associated with a specific user account.
- It contains personalized desktop settings, documents, application configurations, and other user-specific data.
- Types of User Profiles in Windows Server
  - Local User Profile
  - Roaming User Profile (RUP)
  - Mandatory User Profile
  - Temporary User Profile
  - Super Mandatory Profile

Users Profile - Local User Profile

- Created when a user logs into a computer for the first time.
- Stored in **C:\Users\Username**.
- Changes are saved locally and do not roam with the user.

Users Profile - Roaming User Profile (RUP)

- Stored on a **network location**.
- Users get the **same desktop experience** on any computer they log in to.
- Configured via **Group Policy**.

Users Profile - Mandatory User Profile

- A **preconfigured profile** that users cannot modify.
- Any changes made during a **session are discarded** after logout.
- Used for **kiosks** and **shared computers**.

Users Profile - Temporary User Profile

- Created when **Windows cannot load the normal user profile**.
- All changes are **lost after logout**.
- Indicates profile corruption issues.

Users Profile - Super Mandatory Profile

- A **stricter version** of the Mandatory Profile.
- The user cannot log in if the profile is unavailable.
- Profile folder name ends with .man (like: **\\Server\Profiles\UserProfile.man**).

Standalone Server

- A Standalone Server is a traditional, independent server that operates without being integrated into a larger server infrastructure.
- It typically consists of a single physical unit with its own power supply, storage, processing, and network connections.
- Characteristics of a Standalone Server:
  - **Independence**: Functions as a single entity, not dependent on other servers.
  - **Form Factor**: Usually a tower, rack-mounted, or desktop-style.
  - **Hardware**: Contains its own CPU, RAM, storage, and network interface.
  - **Usage**: Commonly used for small businesses, local applications, or dedicated services (e.g., file servers, print servers).
  - **Scalability**: Limited, requires adding more standalone servers for expansion.
  - **Management**: Managed individually through local or remote administration tools (e.g., RDP, PowerShell, Windows Admin Center).

Blade Server

- A Blade Server is a compact, high-density server that fits into a chassis along with multiple other blades.
- These servers share **power, cooling, and network resources** provided by the chassis.
- Characteristics of a Blade Server:
  - **Compact Design**: Multiple thin server blades fit into a single blade enclosure (chassis).
  - **Shared Resources**: Power supplies, cooling fans, and network connections are shared, reducing redundancy and energy consumption.
  - **High Density**: More computing power in less physical space compared to standalone servers.
  - **Scalability**: Easily scalable by adding or removing blades from the chassis.
  - **Management**: Managed centrally using integrated tools like HPE iLO, Dell iDRAC, or Cisco UCS Manager.
  - **Usage**: Ideal for data centers, virtualization, cloud computing, and enterprise applications.

Standalone Server vs. Blade Server

Standalone Server vs. Blade Server

Feature	Standalone Server	Blade Server
Independence	Fully independent	Depends on chassis
Form Factor	Tower or rack-mounted	Blade chassis with multiple blades
Resource Sharing	No sharing	Shared power, cooling, networking
Scalability	Limited, requires additional servers	Easily scalable within chassis
Power Consumption	Higher due to individual components	Lower due to shared power & cooling
Ideal for	Small businesses, specific tasks	Large enterprises, virtualization, cloud computing

NTFS Permissions in Windows Server

- NTFS (New Technology File System) **permissions control access** to files and folders on NTFS-formatted drives in Windows Server.
- These permissions ensure security by **allowing or restricting user** actions such as reading, writing, or modifying data.
- Types of NTFS Permissions
  - NTFS Permissions for **Folder**
  - NTFS Permissions for **Files**

NTFS Permissions for Files

Permission	Description
Full Control	Users can do anything with the file, including changing permissions.
Modify	Users can read, write, and delete the file but not change permissions.
Read & Execute	Users can open and run the file but cannot modify it.
Read	Users can open and view the file.
Write	Users can modify the file but not delete it.

NTFS Permissions for Folders

Permission	Description
Full Control	Users can read, write, modify, delete, and take ownership of files and subfolders.
Modify	Users can read, write, modify, and delete files but cannot change permissions.
Read & Execute	Users can view and run files but cannot modify them.
List Folder Contents	Users can see file and folder names but cannot open files.
Read	Users can view files but cannot modify them.
Write	Users can create and modify files but cannot delete them.

Working of NTFS Permissions

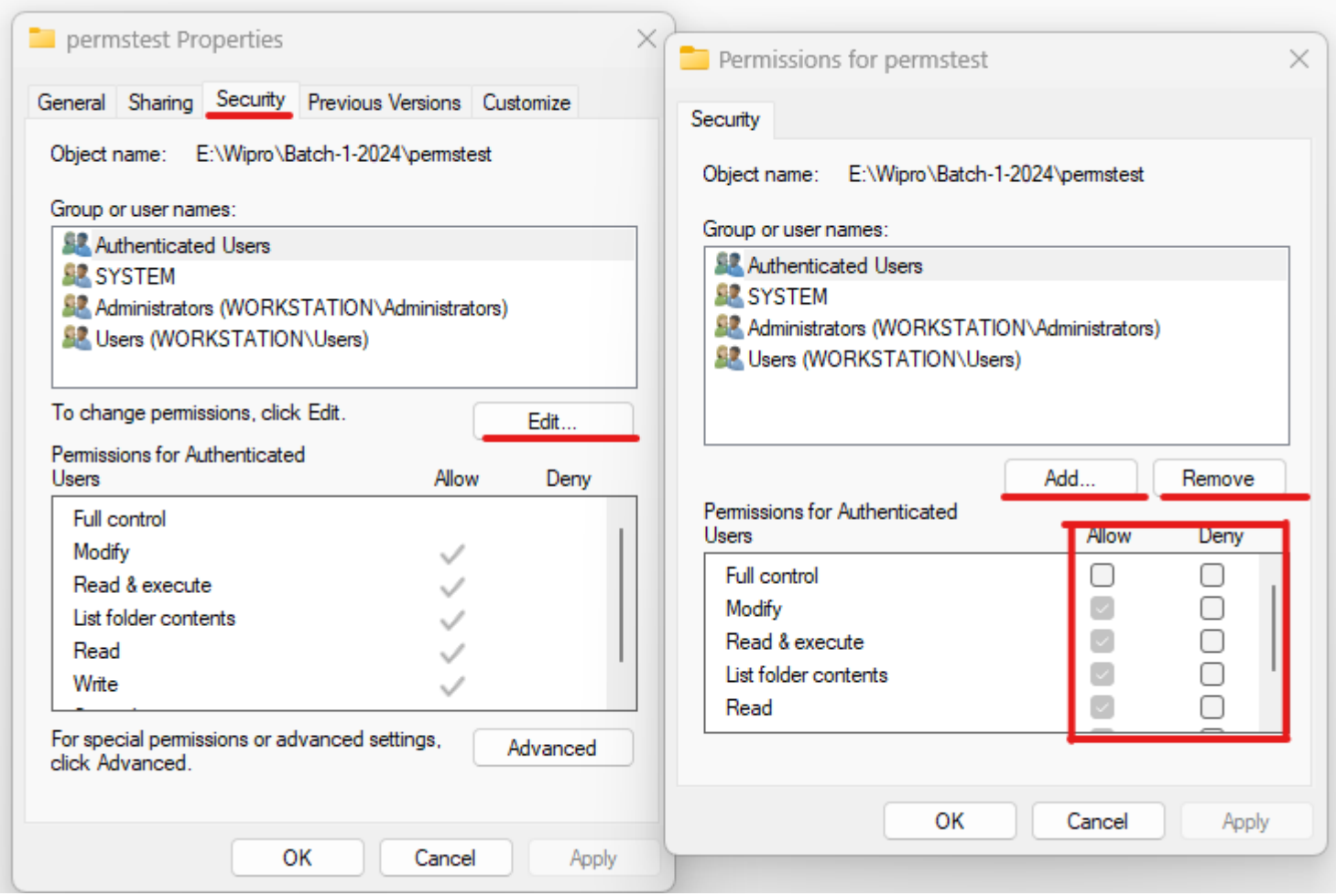
- **Inheritance**: Permissions are inherited from the parent folder unless explicitly changed.
- **Explicit vs. Inherited Permissions**:
  - Explicit Permissions: Set manually on a file/folder.
  - Inherited Permissions: Passed down from the parent folder.
- **Permission Precedence**:
  - Deny overrides Allow: If a user is denied a permission at any level, they cannot perform the action.
  - Most Restrictive Applies: If a user has multiple permissions, the most restrictive permission takes effect.

NTFS vs Share Permissions

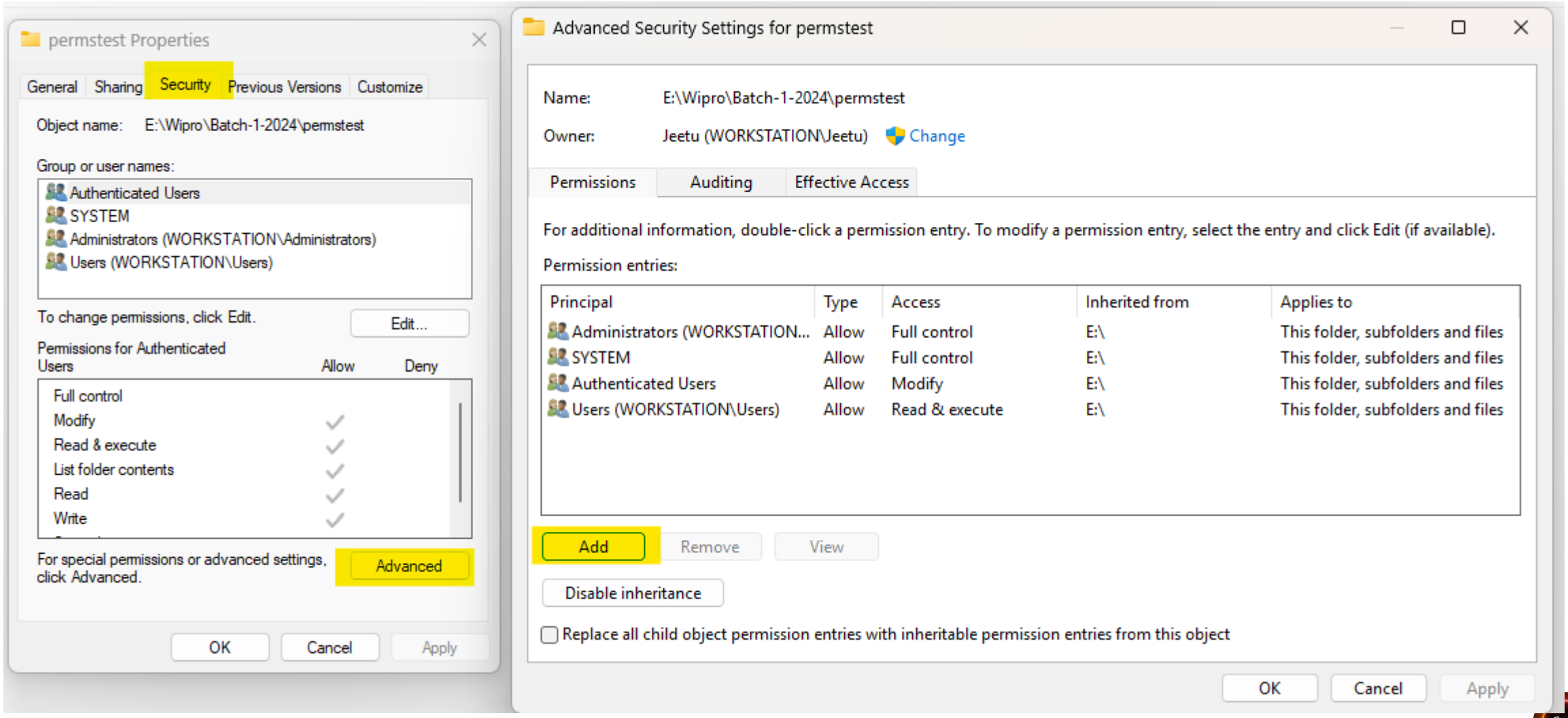
- NTFS permissions control access to files locally and over the network, whereas Share Permissions apply only to network shares.

Feature	NTFS Permissions	Share Permissions
Applies To	Local and network users	Only network users
Security Level	More granular	Less granular
Inheritance	Supports inheritance	No inheritance

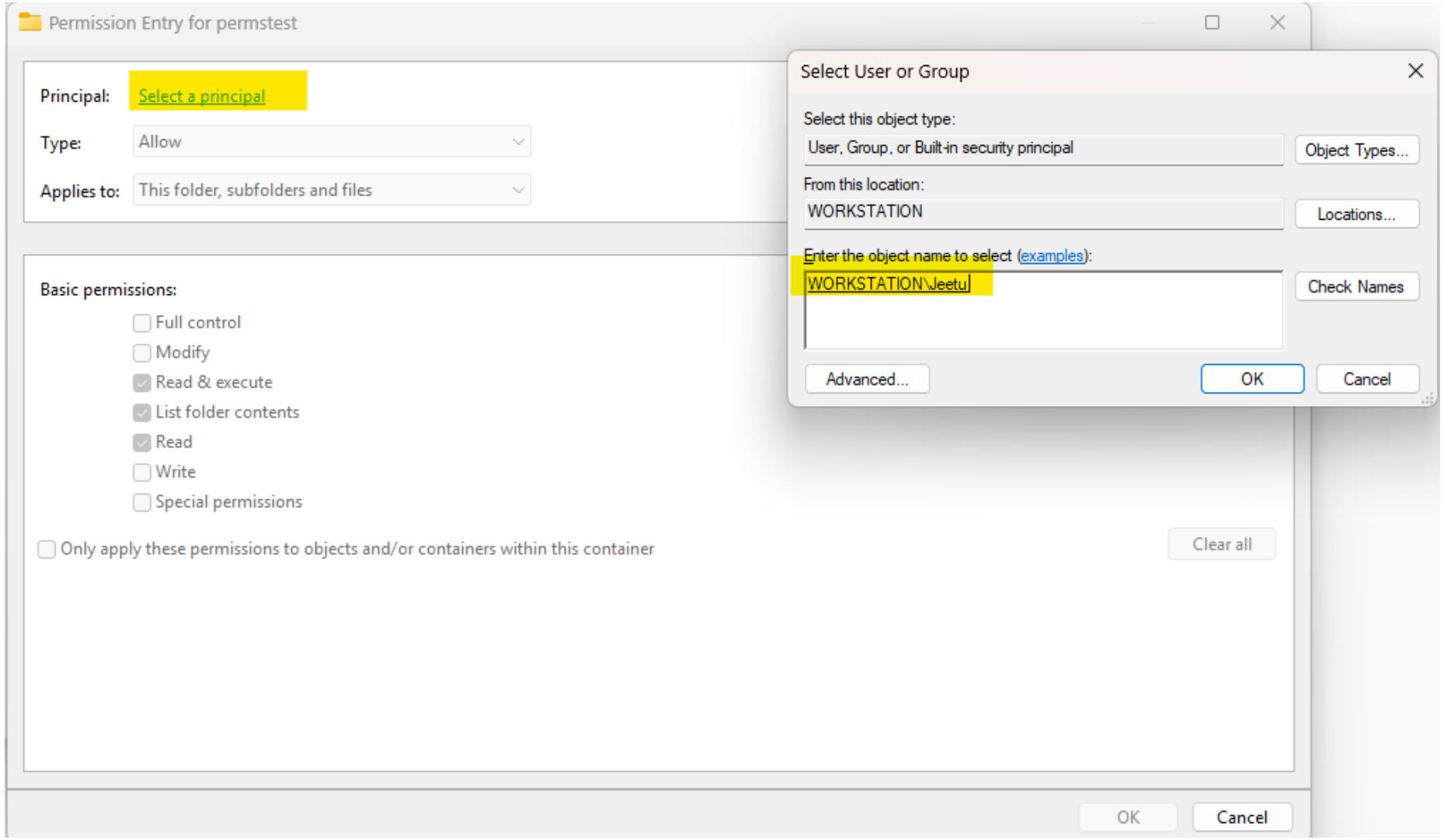
NTFS Permissions



Advance NTFS Permissions



Advance NTFS Permissions



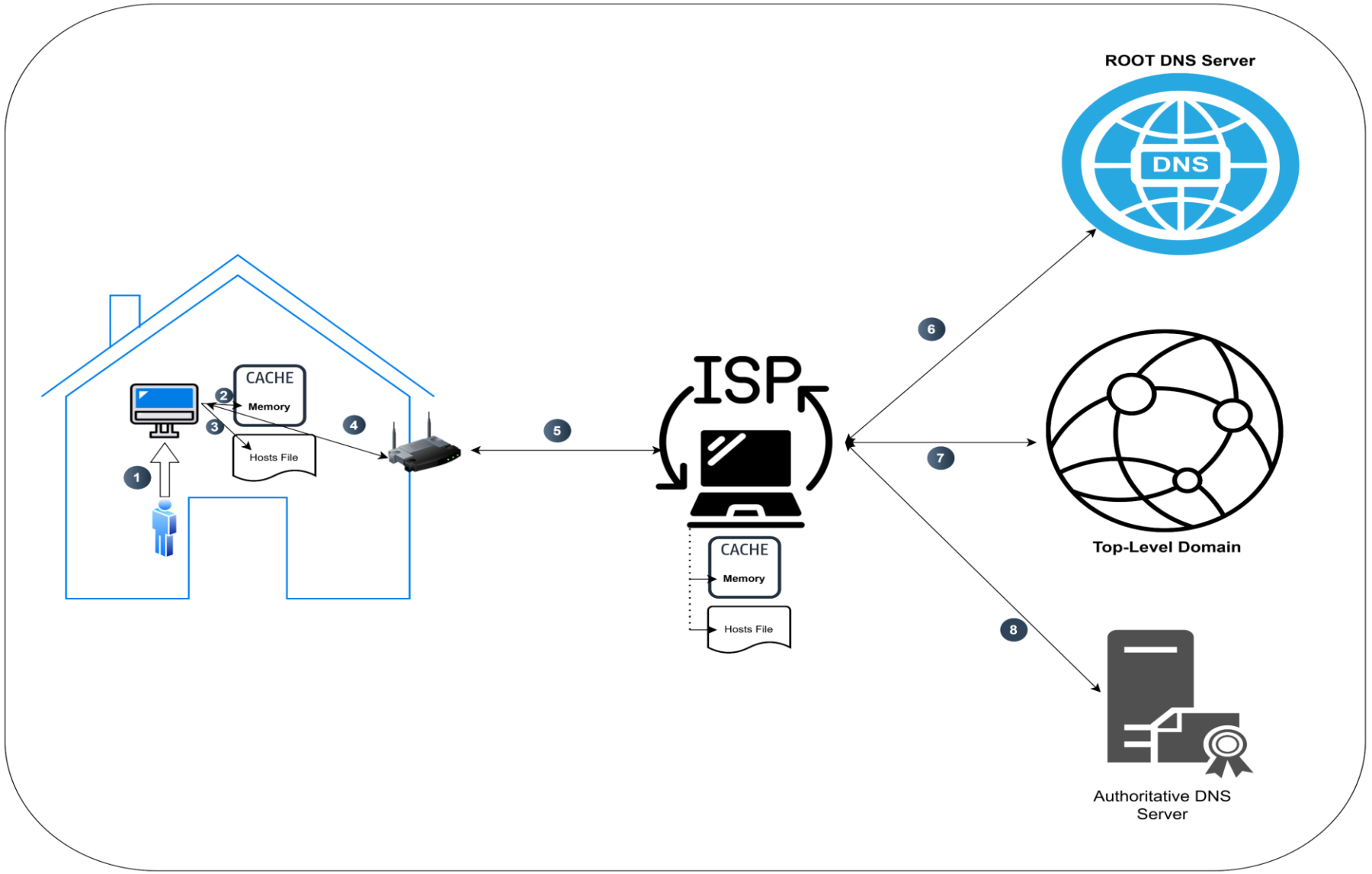
Domain Name System (DNS)

- DNS is a hierarchical system that translates human-readable domain names (e.g., www.jeetusingh.in) into IP address (e.g., 192.168.10.10).
- Types of DNS Servers
  - Recursive Resolver (DNS Recursor)
  - Root DNS Server
  - TLD (Top-Level Domain) DNS Server
  - Authoritative DNS Server

Types of DNS Servers

1. **Recursive Resolver (DNS Recursor)**
   - Acts as an intermediary between client requests and authoritative DNS servers.
   - Stores previously resolved domain names in cache to speed up future requests.
2. **Root DNS Server**
   - First point of contact in the DNS hierarchy.
   - There are 13 sets of root servers worldwide.
3. **TLD (Top-Level Domain) DNS Server**
   - Manages domains based on TLDs (e.g., .com, .org, .net).
   - Example: Verisign manages .com and .net.
4. **Authoritative DNS Server**
   - Stores and provides the final IP address of a domain.
   - Maintained by domain owners or hosting providers.

How DNS Works?



How DNS Works?

1. User wants to access website
2. System checks in its cache memory, if it finds the IP here, it returns IP else
3. it checks it in the HOSTS file.
4. If name resolution did not happen within same system, then it searches the same at ISP (Internet Service Provider).
5. ISP also searches the IP in its cache memory & then in HOSTS file.
6. If IP isn't found, then it looks for the ROOT DNS Server. In case if IP isn't found, then it looks for Top-Level Domain (.COM, .NET, .IN, .GOV).
7. Top-Level Domain (TLD) mostly provides the response. In case, this also fails, then it looks for the Authoritative DNS Server (where actually site is stored), then it responds back.
8. Authoritative DNS Server is actually the server that resolves name to IP and then it responds back.

Types of DNS Records

Record Type	Description	Example
A (Address Record)	Maps a domain name to an IPv4 address.	`example.com → 192.168.1.1`
AAAA (IPv6 Address Record)	Maps a domain name to an IPv6 address.	`example.com → 2001:db8::1`
CNAME (Canonical Name Record)	Maps a domain alias to a real domain name.	`www.example.com → example.com`
MX (Mail Exchange Record)	Specifies mail servers for email delivery.	`mail.example.com → 192.168.2.2`
TXT (Text Record)	Stores arbitrary text, often used for security (SPF, DKIM).	SPF, DKIM, DMARC records
NS (Name Server Record)	Defines authoritative name servers for a domain.	`example.com → ns1.example.com`
PTR (Pointer Record)	Reverse DNS lookup (IP to domain).	`192.168.1.1 → example.com`
SRV (Service Record)	Specifies servers for services like SIP or LDAP.	`_sip._tcp.example.com`
SOA (Start of Authority)	Contains domain admin info and zone settings.	Includes serial number, refresh time, etc.

DNS Query Types

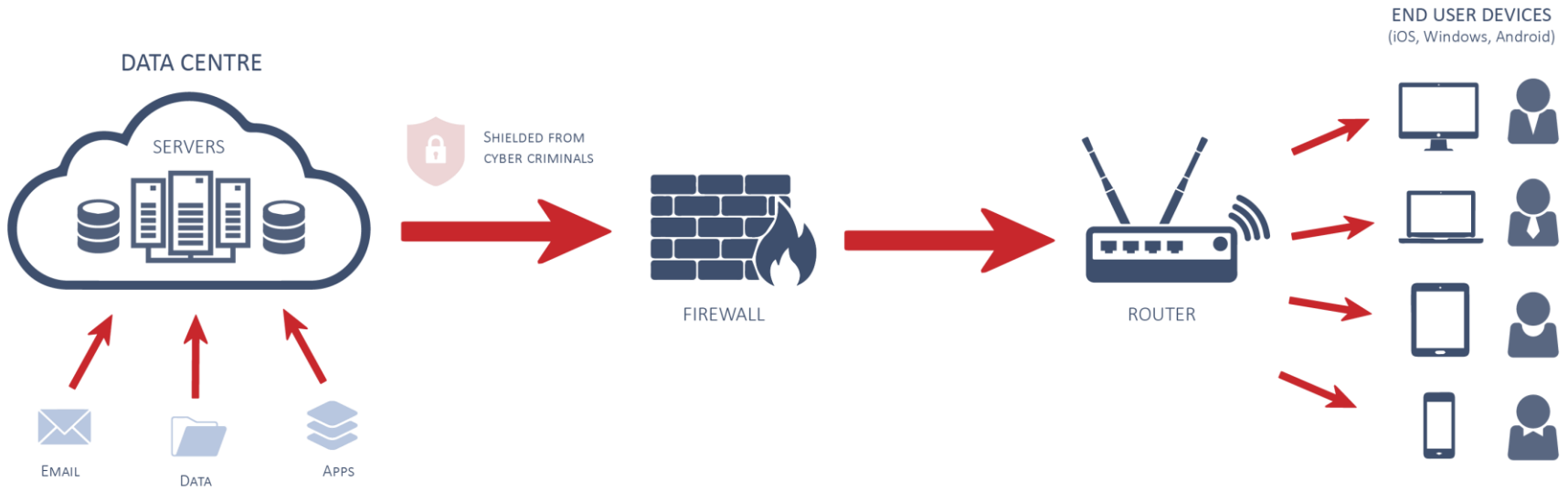
Query Type	Description
Recursive Query	The DNS resolver fetches a complete answer or returns an error.
Iterative Query	The DNS resolver gets partial answers and continues querying other servers.
Reverse Query	Converts an IP address into a domain name (uses PTR records).

Public vs. Private DNS

- **Public DNS**
  - Used for **internet-facing** domains.
  - Google DNS (8.8.8.8), Cloudflare (1.1.1.1)
- **Private DNS**
  - Used for **internal networks** (Active Directory).
  - dc.training.in

Remote Desktop Services (RDS)

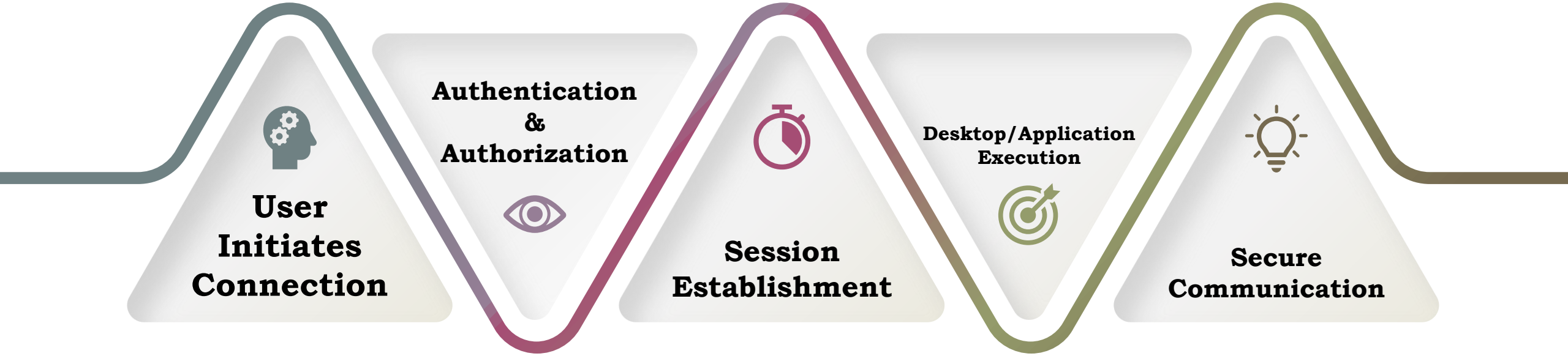
- RDS is a Windows Server role that enables users to remotely access desktops and applications hosted on a Windows Server.
- It allows multiple users to connect simultaneously to a centralized server, reducing hardware costs and improving manageability.



Key Features of RDS

- **Remote Desktop Session Host (RDSH)** – Hosts Windows-based applications and desktops for multiple users.
- **Remote Desktop Gateway (RD Gateway**) – Allows secure remote access over the internet via HTTPS (port 443).
- **Remote Desktop Web Access (RD Web Access)** – Provides a web-based interface to access remote applications and desktops.
- **Remote Desktop Connection Broker (RD Connection Broker)** – Manages session load balancing and reconnections.
- **Remote Desktop Licensing (RD Licensing)** – Manages RDS Client Access Licenses (RDS CALs).
- **RemoteApp** – Publishes applications so they run as if they were installed locally.

How RDS Works?



How RDS Works?

1. **User Initiates Connection** → A user connects via Remote Desktop Client (mstsc.exe) or RD Web Access.
2. **Authentication & Authorization** → The RD Gateway or RD Session Host verifies the user credentials.
3. **Session Establishment** → RD Connection Broker assigns the user to an available session host.
4. **Desktop/Application Execution** → The user accesses a remote desktop or published application.
5. **Secure Communication** → Traffic is encrypted using SSL/TLS.

RDS Components

Component	Function
Remote Desktop Session Host (RDSH)	Hosts remote desktops and applications.
Remote Desktop Gateway (RDG)	Provides secure access over the internet using HTTPS (443).
Remote Desktop Web Access (RDWA)	Web portal for accessing RDS resources.
Remote Desktop Connection Broker (RDCB)	Manages user sessions and load balancing.
Remote Desktop Licensing (RDL)	Manages RDS CALs (Per User / Per Device).

Types of RDS Deployments

- **Session-Based Virtualization**
  - Multiple users share a single Windows Server instance.
  - Applications run on the server but appear local to the user.
- **Virtual Desktop Infrastructure (VDI)**
  - Each user gets a dedicated virtual machine (VM).
  - Provides better user isolation and performance.

RDS Licensing & RDS CALs

- RDS requires **Client Access Licenses (CALs)** for users or devices.
- Two types of RDS CALs:
  1. Per User CAL – Assigned to a user (best for personal use).
  2. Per Device CAL – Assigned to a device (best for shared devices).
- CALs must be installed on an RDS Licensing Server.

Installing & Configuring RDS on Windows Server

- **Step 1: Install the RDS Role**
  - Open Server Manager → Click Manage → Add Roles and Features.
  - Select Remote Desktop Services Installation.
  - Choose Standard Deployment or Quick Start.
  - Select the required RDS components (RDSH, RD Gateway, etc.).
  - Complete the installation and restart the server.
- **Step 2: Configure Remote Desktop Licensing**
  - Open Remote Desktop Licensing Manager (licmgr.exe).
  - Activate the server.
  - Install RDS CALs.
- **Step 3: Publish Applications (RemoteApp)**
  - Open Server Manager → Remote Desktop Services.
  - Click RemoteApp Programs → Add desired applications.
  - Distribute the application to users.


That's all Folks!