Windows Server 2019

Sunday, July 20, 2025 11:10 AM

Using Server Manager After Installing Windows Server 2019 – Summary

✅ Initial Post-Installation Steps

- **Set Administrator Password**: Required before any further actions. Must be complex (uppercase, lowercase, numbers, symbols) and at least 8 characters.
- **Login**: After password is set, Windows finalizes setup and logs you into the desktop environment.

💡 Server Manager Overview

- **Server Manager Launches Automatically**: In the desktop experience version, it helps with immediate setup tasks.
- **First Task Prompt**: "Configure this local server" — can be accessed from the dashboard or via the "Local Server" tab.

🧠 Key Configurations

- **Assign Static IP Address**:
- Access via Legacy network settings
- Go to IPv4 settings and manually set IP, subnet mask, gateway, and DNS
- Use server's own IP as DNS (if it's to become a DNS/domain controller)
- **Rename the Computer**:
- Go to system properties and change the name (e.g., dc01)
- Requires **reboot** to apply changes

Initial Configuration Using PowerShell – Summary

✅ First-Time Setup (No GUI)

- **Set Administrator Password**:
  - Navigate with arrow keys and use Tab to switch fields
  - Must follow complexity rules (uppercase, lowercase, number, symbol)
- **Login**: Logs into Command Prompt by default
  - Launch PowerShell:

```
> PowerShell
```

⚙️ Network Configuration in PowerShell

🔍 *Step 1: Identify Network Adapter*

```
> Get-NetAdapter
```

- Note the **InterfaceIndex** (e.g., 6)

🌐 *Step 2: Assign Static IP Address*

```
> New-NetIPAddress -InterfaceIndex 6 -IPAddress 192.168.0.231
-PrefixLength 24 -DefaultGateway 192.168.0.1
```

🧭 *Step 3: Set DNS Server*

```
> Set-DnsClientServerAddress -InterfaceIndex 6 -
ServerAddresses 192.168.0.230
```

🖥️ Rename the Computer

1. Check current hostname:
   `> hostname`
2. Rename computer (replace with your values):
   `> Rename-Computer -ComputerName OLDNAME -NewName file01`
3. Reboot to apply changes:
   `> shutdown -r`

🔁 Outcome

After reboot:

- Server has a secure **admin password**
- Static **IP address + DNS**
- Assigned **hostname** (file01)
  Ready for domain roles or further server setup.

Configuring Multiple IPs per NIC in Windows Server – Summary

🧠 Why Multiple IPs per NIC?

- **Single NIC, multiple IPs** are useful for hosting **multiple websites** on the same server.

- Web servers can assign **specific IPs to each site**, even if they share hardware.
- This approach **avoids the need for multiple NICs** while allowing **IP-level differentiation**.

⚙️ Desktop Method (GUI)

1. Go to **Server Manager** → Right-click NIC → **Properties**.
2. Select **IPv4**, click **Advanced**.
3. Add **new IP & subnet mask** (no extra gateway needed).
4. Apply changes, and Server Manager will reflect multiple IPv4 addresses.

💻 PowerShell Method

1. Open **PowerShell (Admin)**.
2. Run Get-NetAdapter to find **interface index** (e.g., 5).
3. Add IP:
   ```
   > New-NetIPAddress -InterfaceIndex 5 -IPAddress 192.168.0.242 -PrefixLength 24
   ```
4. Confirm using:
   ```
   > ipconfig
   ```

✅ Result

- One NIC now hosts **multiple IP addresses** (e.g., 192.168.0.240, .241, .242).
- Only **one gateway** is required.
- Enables **multi-site hosting** without extra hardware.

Understanding NIC Teaming in Windows Server – Summary

🧩 What is NIC Teaming?

- NIC Teaming allows **multiple network interfaces (NICs)** to work together as **one logical NIC**.
- Shares a **single IP address**, but increases **throughput** and offers **redundancy**.
- Useful for **file servers**, **Hyper-V**, and **high-availability scenarios**.

💡 Benefits of NIC Teaming

- **Increased Bandwidth**: More concurrent data streams via multiple NICs.
- **Fault Tolerance**: If one NIC fails, traffic continues through the remaining NICs without interruption.
- **Simplified Management**: Only **one IP address** to configure and manage.

🛠️ Teaming Modes

1. **Address Hashing**:
   - Ideal for **many small data streams** (e.g., file servers).
   - Incoming traffic uses one NIC; **outgoing is load balanced** across NICs.
2. **Hyper-V Mode**:
   - Best for **virtual environments** with **high-volume VM traffic**.
   - Uses multiple NICs under one IP for the host, while VMs appear to use one interface.
3. **Dynamic Load Balancing**:
   - Windows intelligently **switches between modes**.
   - Default and best for **mixed traffic** situations.

📌 Real-World Use Case

- Like "**shotgunning**" in dial-up days: combining multiple connections to serve more users or sessions at once — but smarter and fault-tolerant.

Creating Different Volume Types – Summary

🖥️ Disk Management Overview

- Access Disk Management via right-click on Start > *Disk Management*.
- Shows physical drives (e.g., Disk 0, optical drives, thumb drives).
- Existing OS partition (C:) is often set to max space; unused drives appear unallocated.

✂️ Shrinking a Volume

- To resize a partition (e.g., reduce C: from 127GB to 100GB), use **Shrink Volume**.
- Specify the amount of space to remove (not the final size directly).
- The freed space becomes unallocated for creating new volumes.

🔁 Adding a Mirror Volume

- **Mirroring** creates fault-tolerant copies on another disk.
- Requires converting **Basic Disks** to **Dynamic Disks** first.
- After conversion, right-click volume > *Add Mirror* > select another dynamic disk.

🔄 Converting Basic to Dynamic Disks

- Right-click on a disk > *Convert to Dynamic Disk*.
- Enables advanced features: mirrored, striped (RAID 0), spanned, or RAID 5 volumes.
- Warning: may impact non-Windows dual boot setups.

⚙️ Volume Types After Dynamic Conversion

- **Simple Volume**: Basic volume on one disk.
- **Spanned Volume**: Combines space from multiple disks.
- **Striped Volume (RAID 0)**: Performance-focused, requires ≥2 disks.
- **Mirrored Volume (RAID 1)**: Fault-tolerant, requires 2 disks.
- **RAID 5**: Requires ≥3 disks, combines fault tolerance and performance.

Supported File System – Summary

📄 FAT/FAT32

- **Oldest file system** still available in Windows Server.
- **Widely compatible** across platforms — readable by most OSes.
- **Major limitations**: max file and volume size, no file-level security.
- Only **share permissions** protect files.
- **Not recommended** for modern, secure environments.

🔐 NTFS (New Technology File System)

- **Preferred choice** for most Windows Server volumes.
- Supports **large files & volumes** with better performance.
- Offers **granular file-level security** (e.g., add-only, no delete).
- Supports **quotas, compression, encryption**, etc.
- Ideal if **user-level control and auditing** are required.

🛡️ ReFS (Resilient File System)

- Introduced in **Windows Server 2016**.
- Designed for **resilience & fault tolerance**.
- Lacks some NTFS features but evolving rapidly.
- **Includes data deduplication** (removes redundancy, improves efficiency).
- Great for **databases like Exchange Server** — reduces OS interference.
- Recommended when **granular user management is not needed**.

✅ Choosing the Right File System

- Use **NTFS** if you need:
  - File-level security
  - User storage quotas
- Detailed auditing and management
- Use **ReFS** if:
  - You want better performance and redundancy
  - You're hosting **database applications**
  - You don't need user-level file access control

Improvements to Storage Spaces Direct (S2D) - Summary

🔧 What is S2D?

- **Storage Spaces Direct (S2D)** is a **software-defined storage** solution using **clustering** to enhance performance and **resilience**.
- Introduced in **Windows Server 2016 Datacenter**, it supports **HDDs, SSDs**, and even **non-volatile memory (NVDIMM)**.

🧱 How It Works

- Combines multiple servers into **one storage pool** with automatic **replication** and **data distribution**.
- Uses **Cluster Shared Volumes (CSV)** with **ReFS or NTFS** to manage file systems efficiently.
- When new servers are added, S2D **auto-integrates** their drives and **rebalances** storage.

📈 Scalability Enhancements in Server 2019

- Max **servers**: 16
- Max **drives**: 416
- Max **storage pool**: **4 PB** (up from 1 PB)
- Max **volume size**: **64 TB** (doubled from 32 TB)
- Max **per-server capacity**: 400 TB

🛡️ Key Requirements & Notes

- Only available on the **Datacenter Edition** of Server 2016/2019
- Requires **Active Directory domain membership**
- Designed for **future scalability**, anticipating storage demands

Storage Migration Service (SMS) - Summary

🚀 Purpose of SMS

- **Introduced in Windows Server 2019**
- Allows seamless **migration of file servers**, including:
  - Files
  - Configurations

    - Shares and permissions
- Users don't notice changes — **no need to remap drives or paths**.

🧠 Why SMS Matters

- Traditional file servers hold:
    - Carefully planned shares
    - Security policies via AD groups
    - Drive quotas, auditing, etc.
- Migrating all this manually is **tedious** — SMS **automates** it.

🧩 SMS Components

- **Source Server**:
    - Must be **Windows Server 2003 or later**
- **Target Server**:
    - Fresh **Windows Server 2019**
    - Can be **physical**, **virtual**, or **Azure-based**
- **Orchestrator Server** (optional):
    - Manages the migration process
    - Can be same as target or a separate server

🛠️ Requirements

- All servers must be **joined to Active Directory**
- Install **Storage Migration Service** feature on:
    - Target server
    - Orchestrator server (if used)
- **No installation required** on source server

📋 Migration Steps

1. **Inventory**
    - Scans source server, gathers config, AD references
2. **Transfer**
    - Copies all data + shares + settings to target
3. **Cutover**
    - Target server **assumes identity** of old server
    - No changes needed for users (shortcuts, drive maps stay intact)

Installing Roles and Features in Windows Server – Summary

🧩 Planning First

- Have a **clear plan** before installing roles.
- Example: A server might serve as **Domain Controller**, **DNS**, and **Certificate Authority**.
- Certificate Services need **Active Directory + DNS** first.

🖥️ Methods to Install Roles

- Via **Server Manager (GUI)** – ideal for visual guidance.
- Via **PowerShell** – for automation/scripts.
- Via **Windows Admin Center** – web-based UI.

🧭 Steps in Server Manager Wizard

1. **Start Wizard**:
    - Click **Add Roles and Features** from Dashboard or **Manage menu**.
2. **Installation Type**:
    - Choose **"Role-based or feature-based installation"**.
3. **Select Server**:
    - Pick your target machine (e.g., **DC01**).
4. **Select Roles**:
- ✅ **DNS Server**
    - Includes management tools.
- ✅ **Active Directory Domain Services**
    - Adds **PowerShell modules** and **Admin Tools**.

🔧 Additional Steps & Confirmation

- Additional dependencies/tools may be prompted.
- Read descriptions (especially for **Active Directory**).
- Final screen shows **confirmation list**.
- Click **Install** – may take time based on system.

Configuring a Role – Summary

⚙️ Post-Role Installation Setup

- After installing **Active Directory Domain Services (AD DS)** and **DNS Server**, a blue link appears:
  👉 **"Promote this server to a domain controller"**
- If closed accidentally, access the link later via **Server Manager > Notifications (flag icon)**.

🏗️ Domain Controller Configuration Steps

- **Choose setup type**:
  🆕 Create a **new domain** or
  🔄 Join an **existing** one
- Example used: creating a domain for *landonhotel.com*.

🔧 Forest & Domain Functional Levels

- Select a **Forest Functional Level** — affects compatibility with other servers.
- Don't blindly accept defaults:
  ✅ Consider OS versions,
  ✅ Existing applications,
  ✅ Compatibility needs.

🔐 Restore Mode Password & NetBIOS Name

- Provide a **Directory Services Restore Mode (DSRM)** password.
- NetBIOS name suggested automatically (e.g., LANDONHOTEL) — usually fine to accept.

🗂️ Storage Paths

- Specifies locations for:
  📁 AD **Database**,
  🧾 **Log files**,
  📂 **SYSVOL** folder
- Paths can be changed if needed (e.g., for mirrored drives).

✅ Final Review & Installation

- Review all settings before proceeding.
- Some warnings may appear (e.g., DNS delegation) — often safe to ignore in test environments.
- Click **Install** — process may take time and involve reboots.
- Once complete, the server is now a **Domain Controller**.

Managing Features on Demand – Summary

💾 Role Installation Behavior

- Earlier, you needed installation media for each feature/role.
- Since **Windows Server 2008**, all role/feature files are pre-copied to disk.
- Stored in: *C:\Windows\WinSxS* (Side-by-side folder).

📁 About WinSxS Folder

- Contains files for all installed & **available but not yet installed** features.
- Can consume several **GBs of space**.
- ⚠️ **Never delete manually** — doing so will break installed roles.

⚙️ Freeing Up Disk Space (Safely)

- Use PowerShell to **remove** files for *unused* roles/features.
- Steps:
  1. Launch PowerShell.
  2. Run:
     ```
     > Get-WindowsFeature | Where-Object { $_.Installed -eq $false }
     ```
     Lists all *not installed* features.
5. Then run:
   ```
   > Get-WindowsFeature | Where-Object { $_.Installed -eq $false } |
   Uninstall-WindowsFeature -Remove
   ```
   This deletes unneeded files from **WinSxS** safely.

🔐 Security Note

- Removing unneeded features also **improves security**.
- Prevents accidental or unauthorized installation of roles later.

Creating migration tools – Summary

🧭 Migration vs Upgrade

- **Upgrade** keeps the same hardware/server.
- **Migration** means moving roles/configs to a **new server**, keeping roles/features intact.

🖥️ Source & Target Servers

- **Source Server**: Existing one (e.g., Windows Server 2012 R2) with roles already running.
- **Target Server**: New one (e.g., Windows Server 2019), fresh install with required roles but **not configured** yet.

🛠️ Step 1: Add Migration Feature

Run on **Target Server**:

```
> Add-WindowsFeature Migration
```

- This creates a folder: C:\Windows\System32\ServerMigrationTools.

⚙️ Step 2: Generate Migration Tools

Use the following command inside that folder:

```
> SMIGDeploy.exe /package /architecture amd64 /os WS12R2 /path
C:\SMIG
```

- */package* → Create tool package
- */architecture* → amd64 for 64-bit systems
- */os* → WS12R2 for Windows Server 2012 R2
- */path* → Must pre-create (e.g., C:\SMIG)

📤 Step 3: Copy Tools to Source Server

- Once tools are created, **copy** the SMIG folder to the **Source Server**.
- You can now begin **exporting role configuration** from source.

Exporting and importing with SMIG – Summary

📤 Exporting from Source Server

- Create SMIG tools for the **source server**, copy them to it.
- Run *SMIGDeploy.exe* to launch a PowerShell session with necessary cmdlets.
- Use *Get-SmigServerFeature* to list **migratable roles**, each with a **FeatureID**.
- Create a folder to store exports.
- Run
  `> Export-SmigServerSetting -FeatureID <id> -Path <export path>.`
- Export may require **credentials** due to sensitive config data.

📥 Importing to Target Server

- Ensure the **desired role is already installed** (e.g., DHCP).
- Stop the role's service:
  `> Stop-Service <ServiceName> (e.g., Stop-Service DHCPServer).`
- Launch SMIG tools via **Server Manager > Tools > Windows Server Migration Tools**.
- Use Import -
  `> SmigServerSetting -FeatureID <id> -Path <import path> -Force to apply.`
- Without *-Force*, import might silently fail (*Success: False*).
- Finally, **restart the service** after import.

🧠 Key Points

- Migration only supports roles eligible for **in-place upgrade**.
- You can **upgrade on existing hardware** or **migrate to a new machine** using this method.