BAHIR DAR UNIVERSITY

BAHIR DAR INSTITUTE OF TECHNOLOGY
FACULTY OF COMPUTING
DEPARTMENT OF INFORMATION SYSTEMS

OPERATING SYSTEM ASSIGNMENT:window servwr 12 Virtual Installation & Hardware Evaluation

Name= kidus worku
ID= BDU1601883

Submit to = lec.alemitu.M

Submission date 06/05/2018 e.c

Table of Contents

---

1. Introduction: Why This Matters

1.1 Welcome to the World of Servers

Imagine walking into a large office building. The lights are on, the internet works, printers are humming, and everyone can access their files. Who makes all this happen? **Servers.** They're the unsung heroes—powerful computers running 24/7 that manage everything from email to security to file sharing.

But here's the problem: you can't just walk up to a company's main server and start experimenting. One wrong click could take down the entire network. That's where **virtualization** comes in—it lets you create a perfect digital copy of a server on your own laptop, where you can learn, break things, and fix them without consequences.

1.2 Why Virtualization? The Game Changer

Think of virtualization like building a model airplane instead of flying a real one. You learn all the same principles—aerodynamics, controls, mechanics—but if you crash, you just pick up the pieces and start over. Virtualization software (like Oracle VirtualBox) creates a **"virtual computer"** inside your actual computer. This virtual machine (VM) behaves exactly like a physical server, but it's completely contained in a safe digital bubble.

1.3 Why Windows Server 2012 R2?

You might wonder: "Why install an older server operating system? Isn't everything moving to the cloud?" Excellent questions. Here's why:

- **Foundational Learning:** Many current server technologies evolved from systems like Windows Server 2012 R2. Learning this gives you historical context.
- **Still in Use:** Believe it or not, many companies still run older server systems because upgrading is expensive and risky.

- **Perfect for Practice:** It's stable, well-documented, and teaches you core concepts without overwhelming you with the latest features.

---

2. Objectives: What You'll Actually Learn

2.1 Beyond Theory: Hands-On Magic

In classrooms, you learn about **partitions, file systems, and boot processes** as abstract concepts. Here's what happens when you actually do it:

- **Partitioning:** You'll see that partitioning a hard drive is like dividing a blank notebook into sections—one for the operating system, one for programs, one for data. You make these decisions yourself.
- **Boot Process:** You'll watch as the virtual machine goes through the exact same startup sequence as a physical server—BIOS, bootloader, kernel loading.
- **File Systems:** You'll choose NTFS and understand *why* it's the right choice (security features, reliability).

2.2 Skills You Can Put on Your Resume

After completing this project, you can honestly say you have experience with:

- **Virtual Machine Management:** Creating, configuring, and maintaining VMs
- **Server OS Installation:** Not just desktop Windows, but server-grade installation
- **Basic Server Roles:** Like setting up Active Directory (the system that manages user accounts in companies)
- **Troubleshooting:** Solving real problems that come up during installation

These aren't just academic exercises—they're exactly the skills junior system administrators use daily.

---

3. Requirements: Gathering Your Digital Tools

3.1 Hardware: What Your Computer Needs

Let's demystify the technical requirements:

- **Processor:** Any relatively modern CPU (from the last 5-7 years) works. The key is it needs to support **virtualization technology** (Intel VT-x or AMD-V). This is like having a special "virtualization assistant" built into your processor.
- **RAM:** Think of RAM as your desk space. Your host OS (like Windows 10) needs some, and your virtual server needs some too. 8GB total gives you enough "desk space" for both.
- **Storage:** 50GB sounds like a lot, but remember—you're creating an entire computer within your computer. Modern games often take up more space!
- **Internet:** For downloading the software and ISO file. The Windows Server 2012 R2 evaluation copy is **free for 180 days**—perfect for learning.

3.2 Software: The Programs That Make It Possible

- **Oracle VM VirtualBox:** Free, reliable, and available for Windows, Mac, and Linux. It's like the stage where your virtual server will perform.
- **Windows Server 2012 R2 ISO:** Think of this as the "installer DVD" but in digital form. You download it from Microsoft's website.
- **Your Host OS:** Whatever you're running now (Windows 10/11, macOS, Linux) works fine.

3.3 Mindset: Patience and Curiosity

The most important requirement isn't technical—it's your attitude. You will encounter confusing screens. Things won't work the first time. This is **completely normal**. Every system administrator has been through this. The difference between frustration and learning is simple: when something doesn't work, get curious instead of annoyed. Ask "Why?" and start investigating.

---

4. Step-by-Step Installation: Walking Through Each Click

4.1 Setting the Stage: VirtualBox Basics

When you first open VirtualBox, it might look intimidating with its menus and settings. Let's break it down:

- **Main Interface:** Shows all your virtual machines (empty at first)
- **"New" Button:** This starts the creation process
- **Settings:** Where you tweak things after creation

Think of VirtualBox as a **virtual computer factory**. You're about to place an order for a custom-built computer.

4.2 Creating Your First Virtual Machine

Here's what happens when you click "New":

1. **Name Your VM:** "Windows Server 2012 R2" – descriptive names help later
2. **Choose OS Type:** This tells VirtualBox what kind of computer to emulate
3. **Memory Allocation:** 4096 MB = 4GB. This is your VM's "thinking space"
4. **Hard Disk Creation:** 50GB dynamically allocated means it starts small and grows as needed
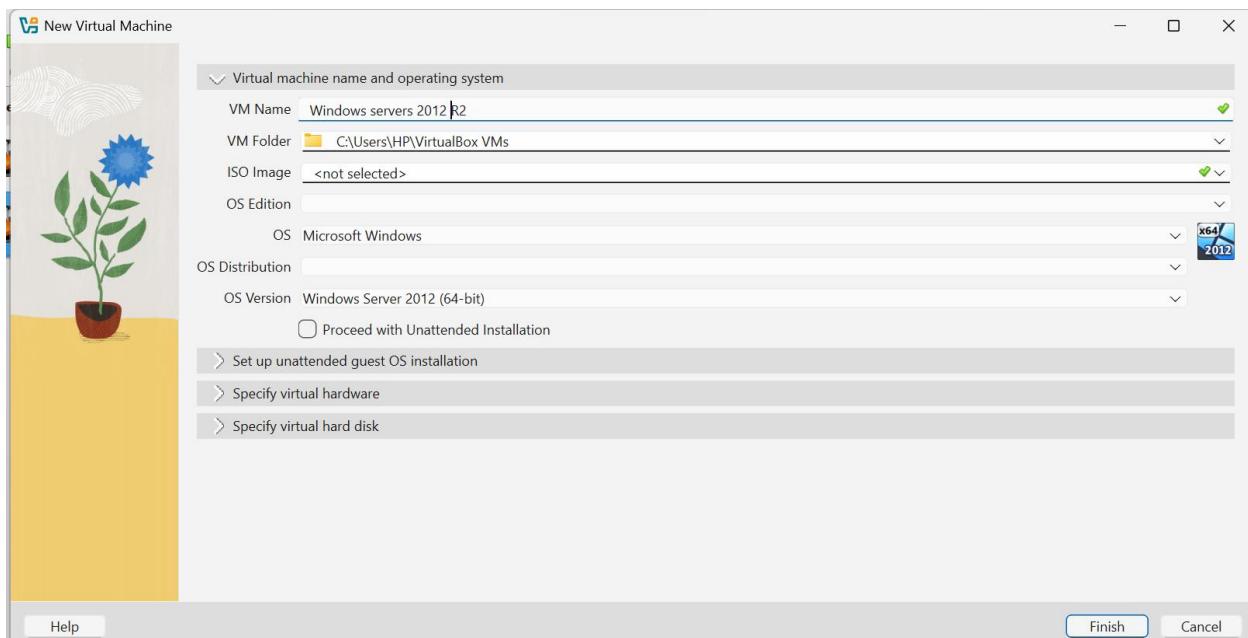

Figure 1: Naming your virtual server – like naming a pet, it makes it feel real

The "Unattended Installation" Checkbox:
VirtualBox offers to automate installation. We deliberately **didn't check this** because we want to learn each step manually. It's like choosing to bake a cake from scratch instead of using a mix.

4.3 The Installation Process – Every Screen Explained

Screen 1: Language and Keyboard Selection
This seems obvious, but it matters for multinational companies. Servers in Japan need different keyboard layouts than servers in France.

Screen 2: License Terms
You've seen these before—pages of legal text. The important part: Microsoft grants you permission to use the software under specific conditions. The evaluation version is for testing and learning only.
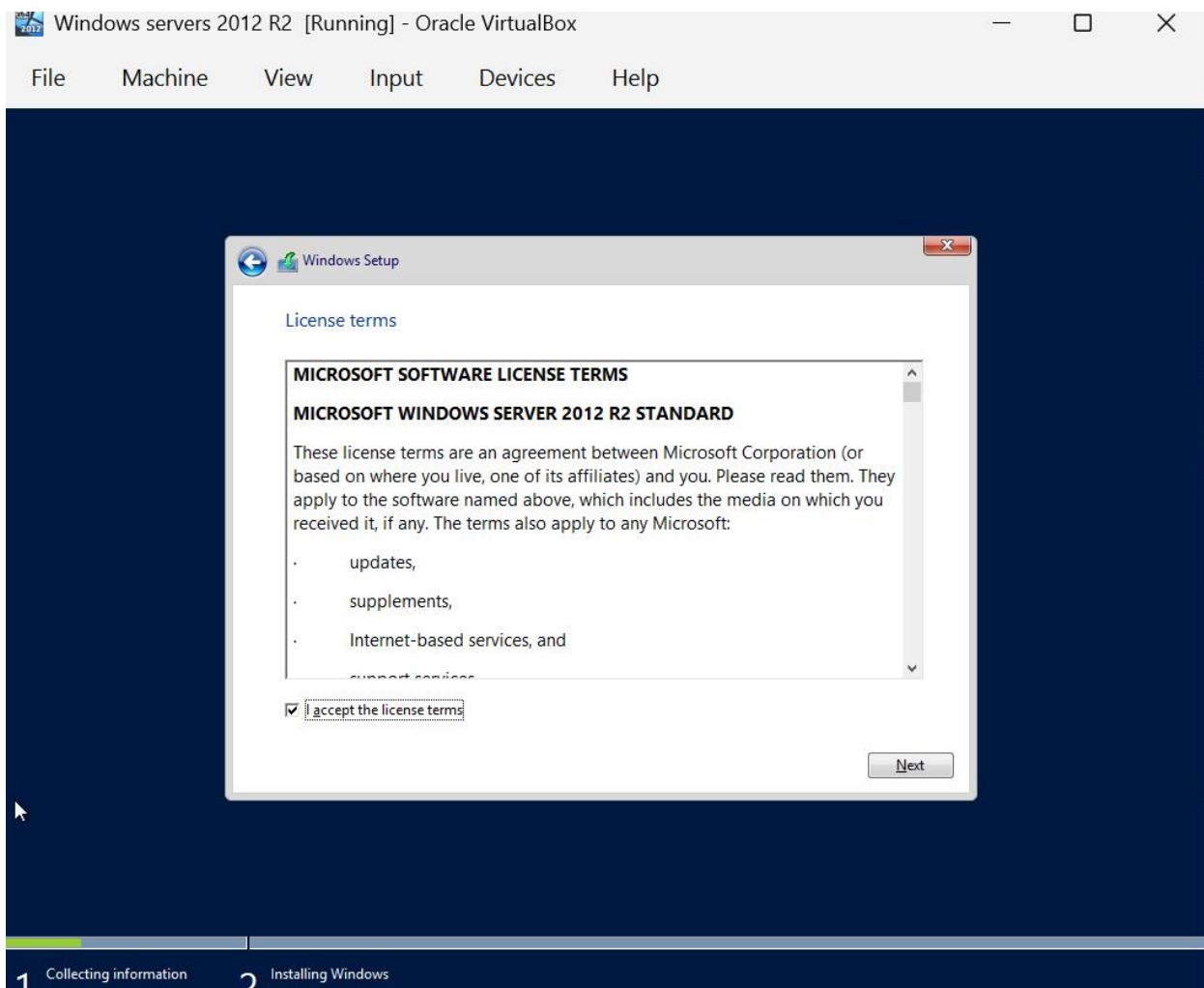
Figure 2: The digital "contract" – read it, or at least pretend to

Screen 3: Installation Type
"Custom" installation lets you control everything. "Upgrade" would keep old settings— but we're starting fresh on empty virtual hardware.

Screen 4: Disk Partitioning

Here's where the magic happens. You see one 50GB "disk" (actually a file on your real hard drive). When you click "Next," Windows:

1. Creates a small system partition (100MB for boot files)
2. Creates the main C: drive partition
3. Formats everything with NTFS

Screen 5: The Installation Progress Bar

This is the waiting game. Windows is:

- Copying thousands of files
- Installing core components
- Setting up the registry (Windows' configuration database)
- Installing updates
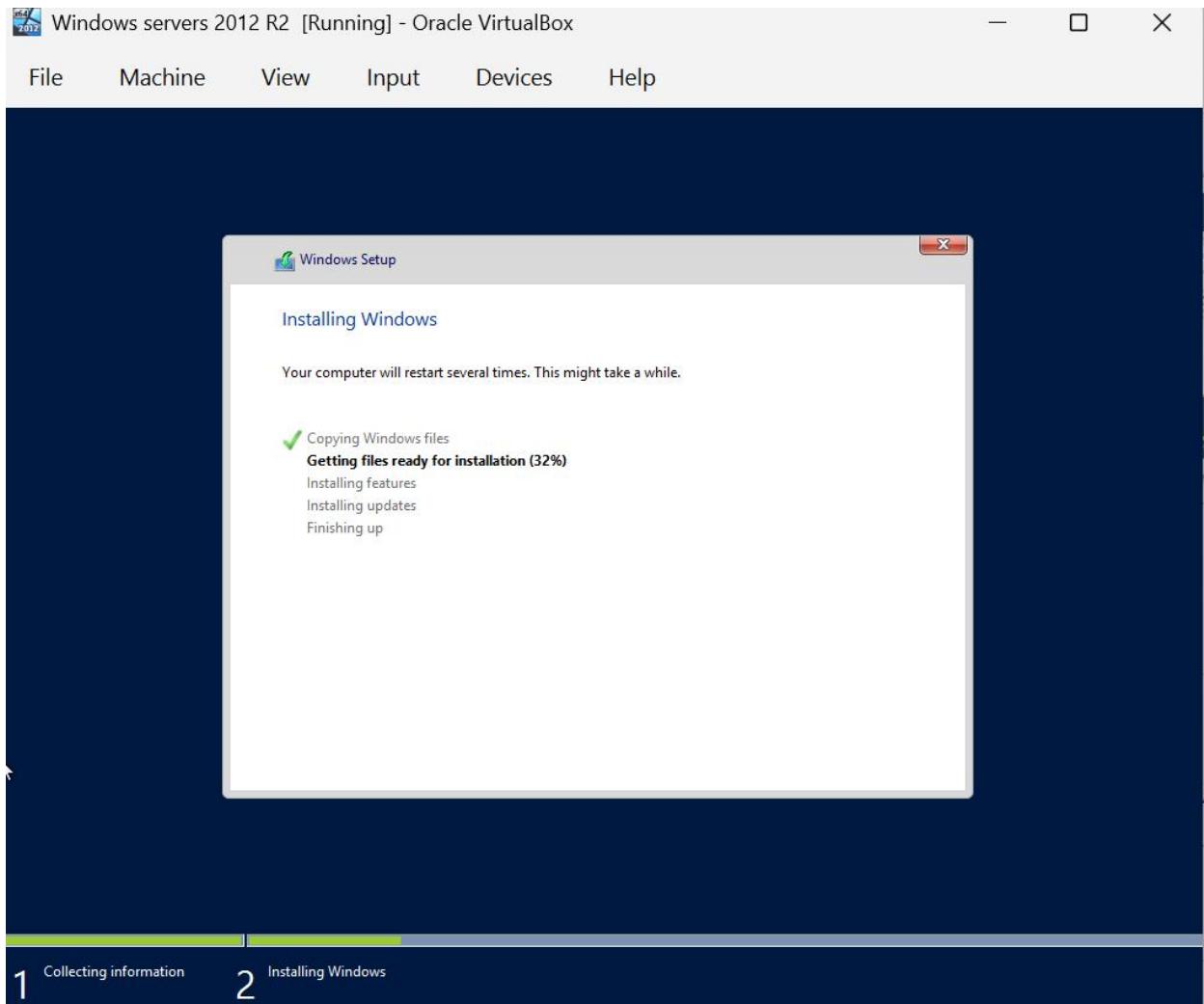
Figure 3: The progress bar – your new best friend and occasional source of anxiety

**The Restarts:**
The VM will restart several times. This is normal! Each restart moves to a new phase of setup.

4.4 First Boot: That "It Works!" Moment

After the final restart, you see the login screen. But wait—there's only one user: **Administrator**. On a server, there's no casual "user account." Every login is significant.
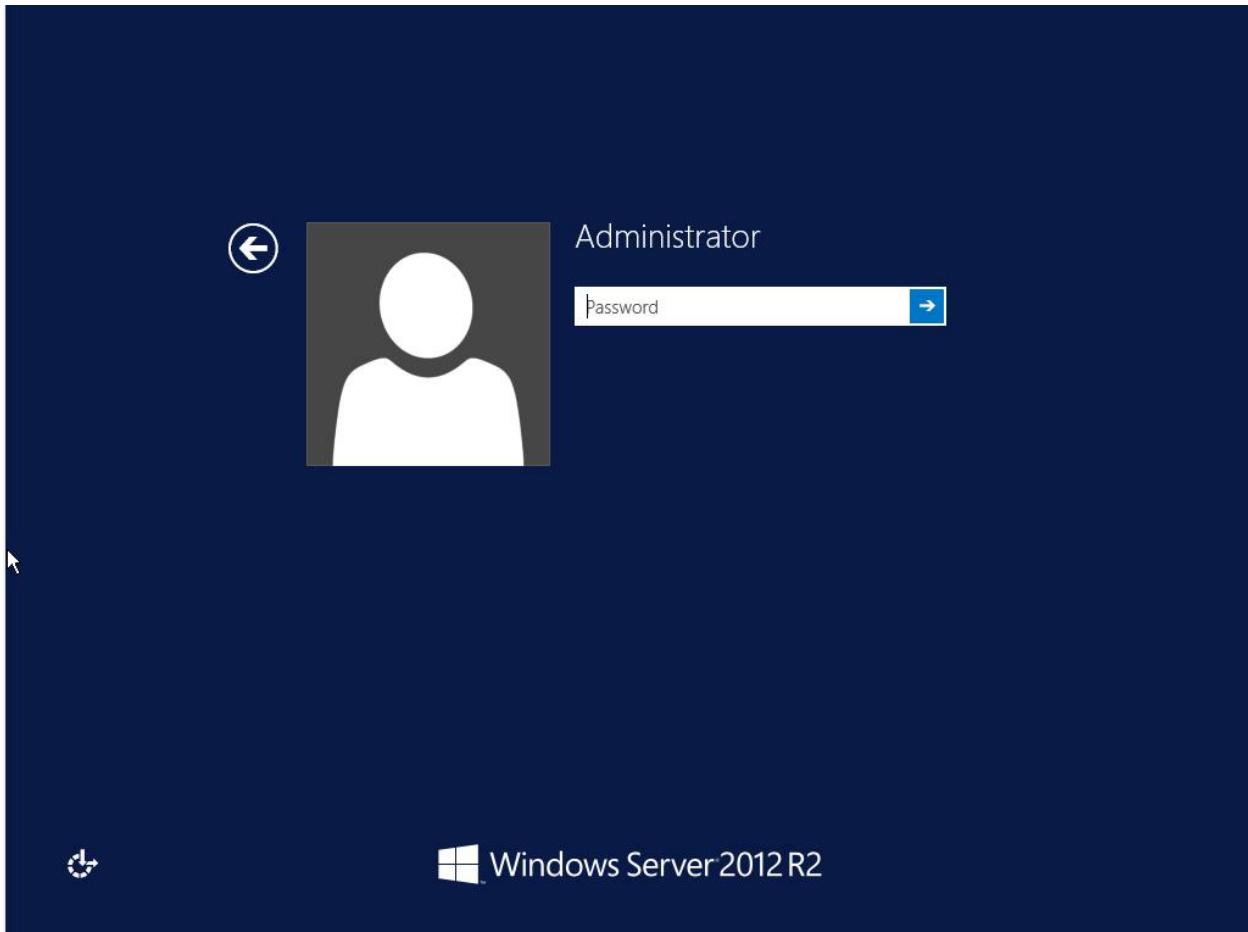
Figure 4: The keys to the kingdom – Administrator access

Setting the Administrator Password:
This is more serious than your social media password. Server passwords should be:

- Long (12+ characters)
- Complex (mixing letters, numbers, symbols)
- Memorable (so you don't write it down)

A good technique: Use a sentence like **"MyFirstServer2012R2!"**

4.5 Initial Configuration: Making It Useful

Server Manager – Your Control Panel
Upon login, Server Manager opens automatically. This is your mission control for everything the server does.
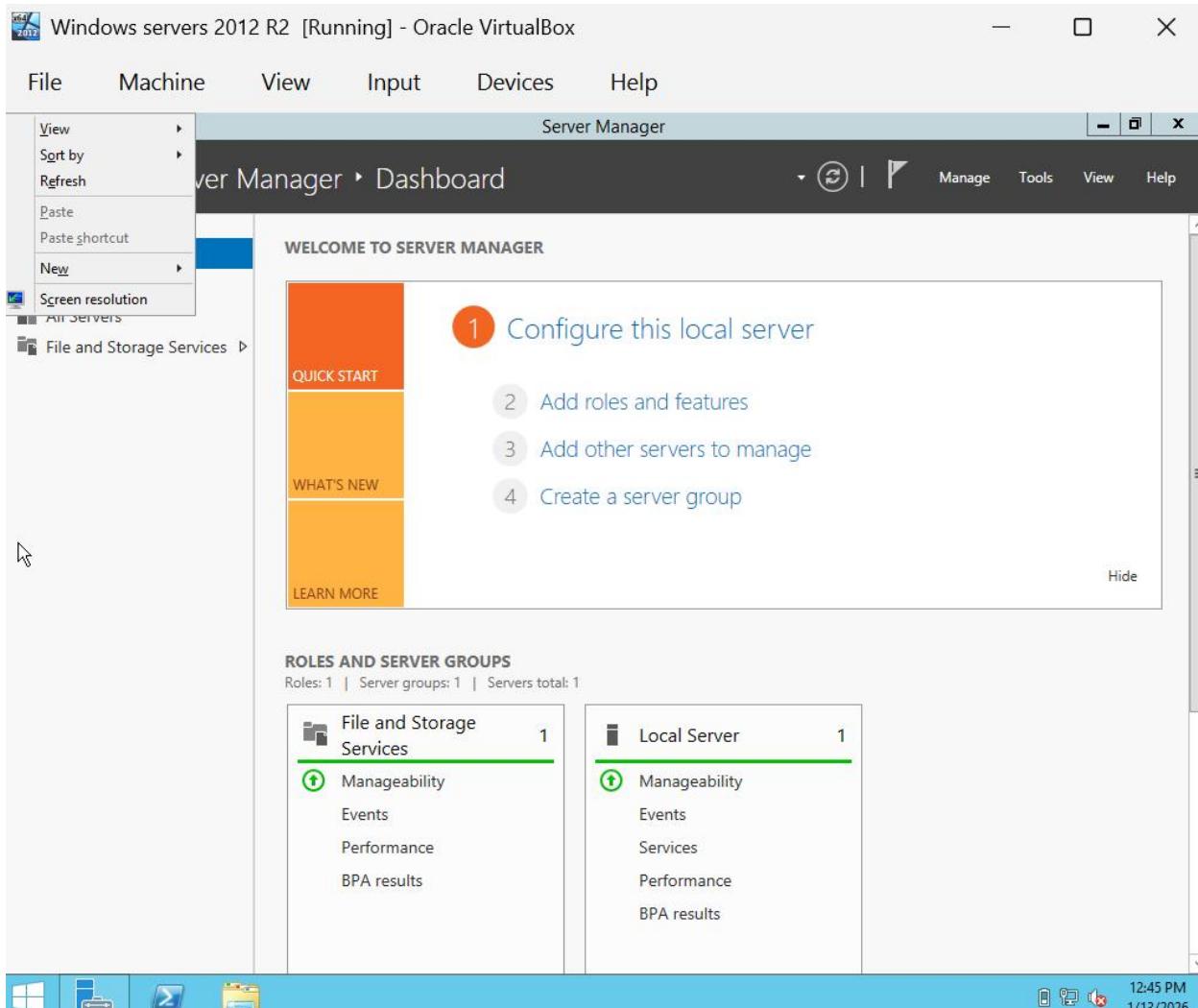
Figure 5: Server Manager – where you'll spend most of your time

Let's explore what you're seeing:

- **Local Server:** Information about *this* server
- **Roles and Features:** Services the server provides (file sharing, web server, etc.)
- **Tools:** For managing users, disks, and performance

Adding Your First Role: Active Directory

Most Windows servers in companies serve as **domain controllers**—they manage user accounts and security policies. Here's how we add this capability:

1. Click "Add roles and features"
2. Select "Active Directory Domain Services"
3. Follow the wizard, accepting defaults for now
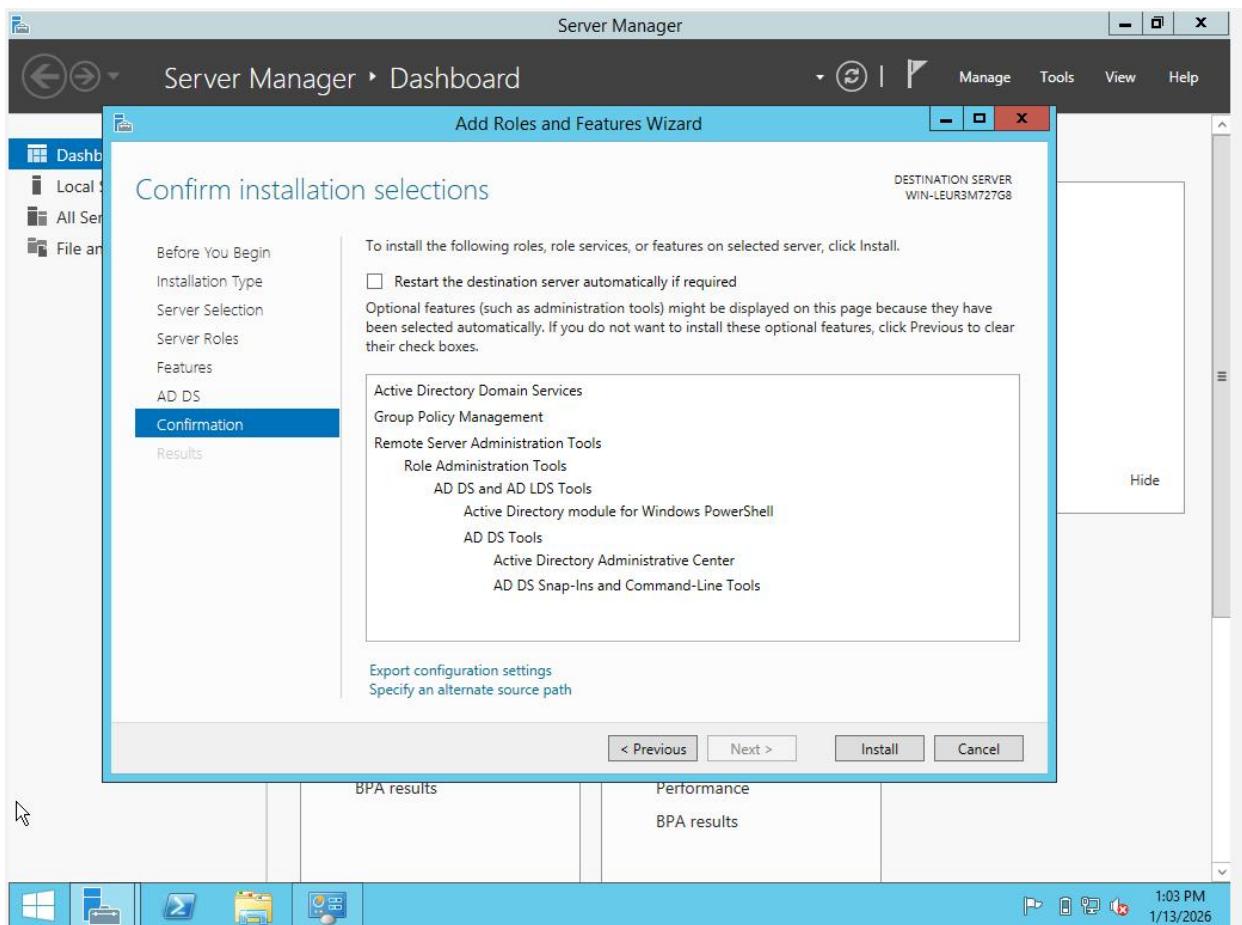
Figure 6: Adding Active Directory – like giving your server a promotion

What Happens During Installation:
Windows installs necessary files and prepares the server for its new role. This takes a few minutes. The progress screen shows exactly what's being installed.
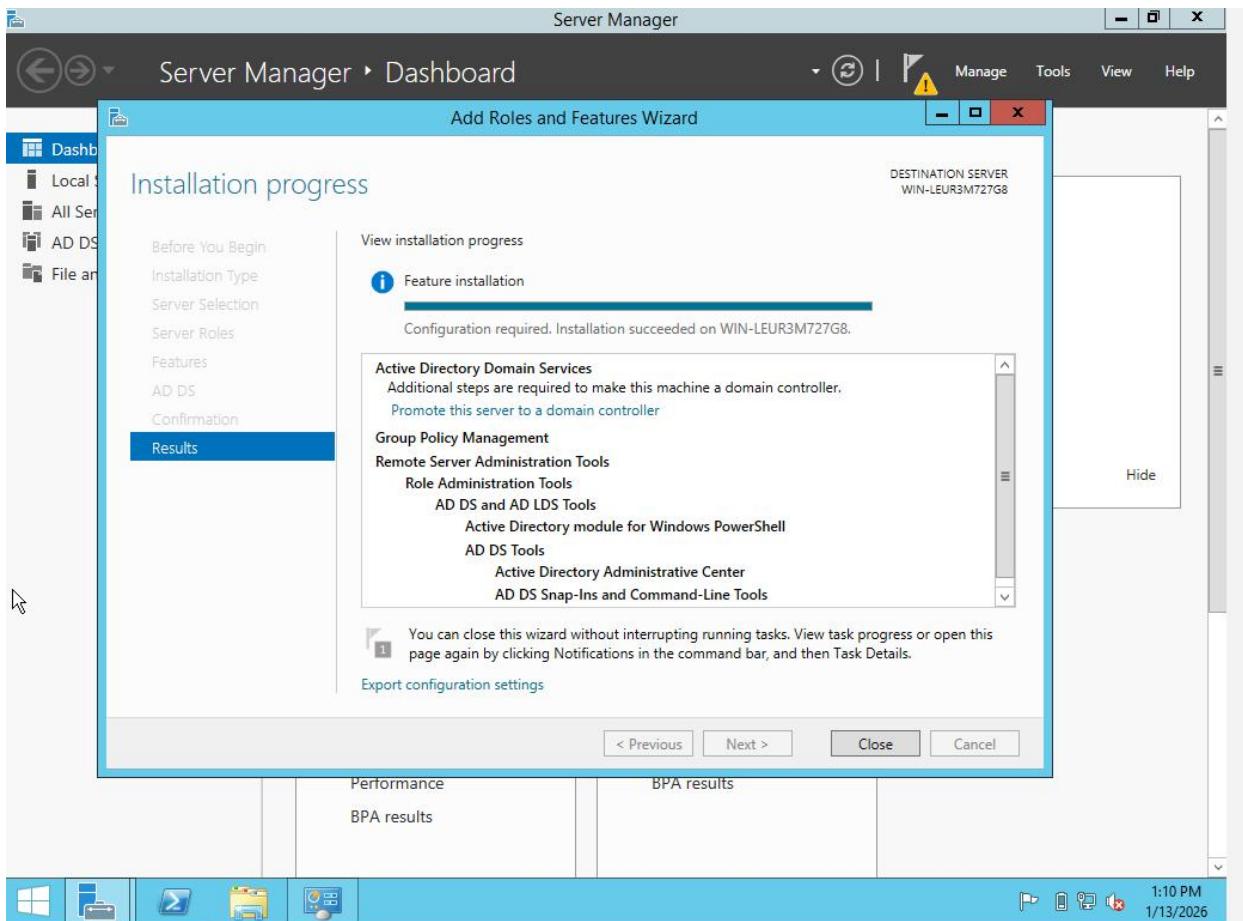
Figure 7: Watching the installation – each component has a purpose

Promoting to Domain Controller:
After installation, you see a notification: "Additional steps required." This means you need to run the **Domain Services Configuration Wizard** to actually turn on Active Directory. It's like installing a car engine (the role) but still needing to connect it to the transmission (the configuration).

4.6 Verifying Everything Works

File Explorer Check:
Open File Explorer (the folder icon) to see your server's "desktop." You'll notice it looks similar to regular Windows but without consumer apps like games or media players.
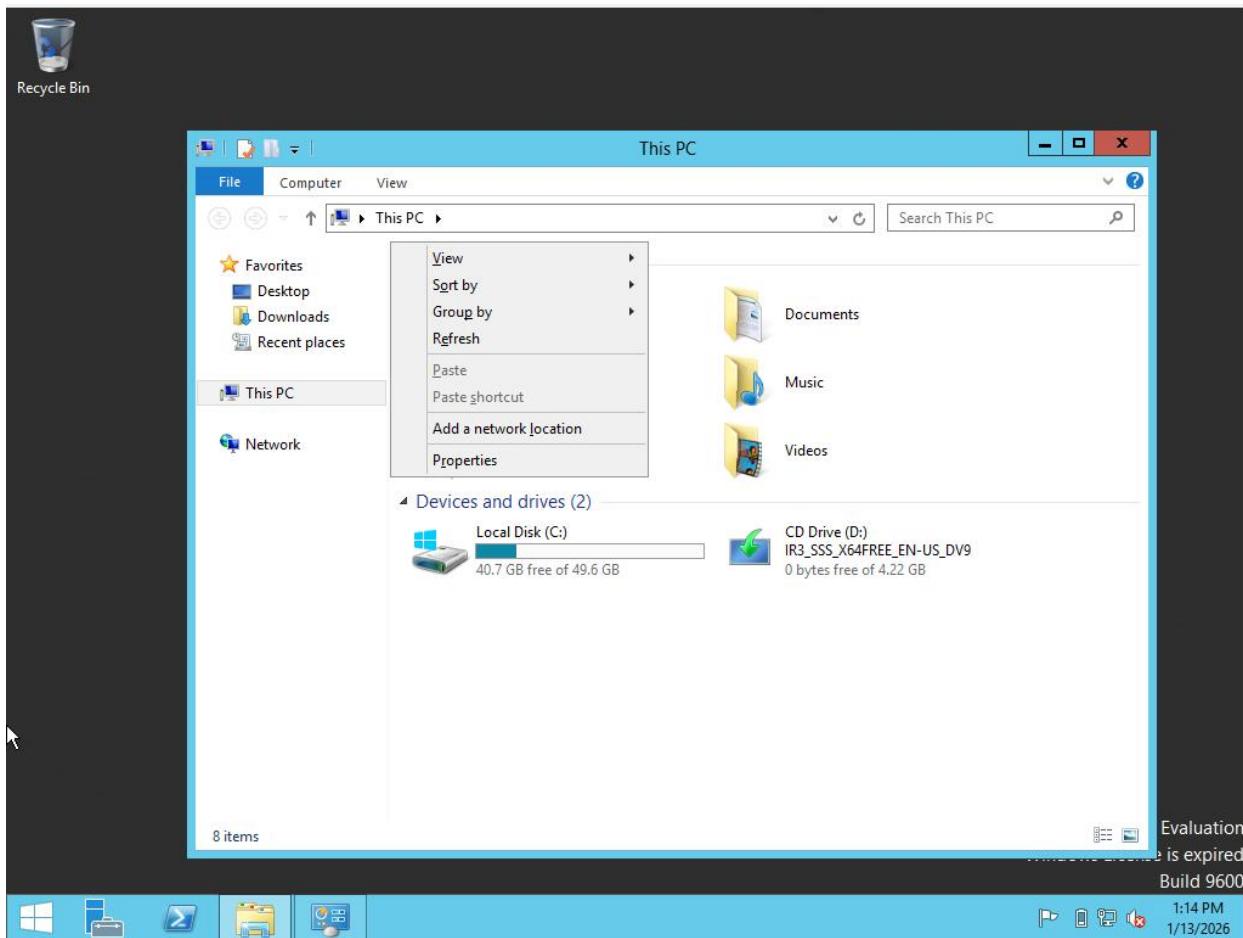
Figure 8: The server's "This PC" view – clean and professional

Notice:

- **C: drive** – Where the OS lives
- **D: drive** – The virtual DVD with your installation ISO
- No user folders filled with personal files

Network Check:
Open Command Prompt and type `ipconfig`. You should see an IP address assigned to your virtual network adapter. If it starts with 192.168 or 10.x, you're connected!

5. Issues Faced: When Things Didn't Go Smoothly

5.1 The "Unattended Installation" Confusion

What happened: During VM creation, VirtualBox showed a checkbox saying "Proceed with Unattended Installation."

Why it was confusing: The option sounds convenient—automatic installation! But what exactly would it automate? Would it skip important steps? Would it use default settings I might want to change?

The learning moment: In IT, "convenient" options often hide complexity. By choosing manual installation, I:

- Saw every decision point
- Understood what each setting meant
- Could document the exact process

This is crucial for troubleshooting later. If something goes wrong with an automated install, you have no idea which step failed.

5.2 Guest Additions: The Missing Puzzle Piece

What happened: After installation, the virtual server's screen was small (800x600 resolution), the mouse didn't move smoothly between host and VM, and copy-paste didn't work.

Why it happened: VirtualBox creates a basic virtual graphics card and mouse that work but aren't optimized. Guest Additions are special drivers that make the VM feel more "integrated" with your host system.

The irony: You can't install Guest Additions immediately because Windows Server 2012 R2 doesn't include the necessary foundation (.NET Framework 3.5) by default. It's a chicken-and-egg problem!

5.3 Networking: Why Can't It See the Internet?

What happened: The virtual server showed "Limited connectivity" or no internet access.

Why it happened: VirtualBox defaults to **NAT (Network Address Translation)** networking. This gives the VM internet access through the host but isolates it from other devices on your network.

The real-world parallel: This is exactly how home routers work! Your phone and laptop share one public IP address. NAT provides security through isolation, but for a server that needs to be reached by other computers, it's problematic.

---

6. Solutions: How We Fixed It

6.1 Embracing Manual Installation

Instead of feeling frustrated by the unattended installation option, I recognized it as a **learning opportunity checklist**:

1. **Partitioning:** I could see exactly how Windows divides the disk
2. **File system choice:** I consciously selected NTFS
3. **User account setup:** I created a proper Administrator password
4. **Initial configuration:** I went through each first-run screen

The result? Deep understanding instead of surface-level familiarity.

6.2 Installing Guest Additions – The Right Way

Here's the step-by-step solution:

1. **Enable .NET Framework 3.5:**

- Open Server Manager
- Click "Add roles and features"
- Navigate to Features
- Check ".NET Framework 3.5 Features"
- Complete installation (may require the installation ISO)
2. **Insert Guest Additions ISO:**

- In VirtualBox menu: Devices > Insert Guest Additions CD Image
- This mounts a virtual CD in the VM
3. **Run the Installer:**

- Open File Explorer
- Navigate to the virtual CD drive
- Run `VBoxWindowsAdditions.exe`

- Accept all defaults during installation
4. **Restart the VM:**

- After installation completes, restart
- Enjoy full screen resolution, seamless mouse, and shared clipboard

Why this matters: These are the exact steps you'd use to install drivers on a physical server. The principles are identical.

6.3 Bridging the Network Gap

To make the server accessible:

1. **Shut down the VM** (fully powered off)
2. **Open VM Settings > Network**
3. **Change "Attached to" from NAT to "Bridged Adapter"**
4. **Start the VM**
5. **Renew IP address:**

```
ipconfig /release
ipconfig /renew
```

Bridged networking explained: This connects the VM directly to your physical network, as if it were another physical computer plugged into your router. The VM gets its own IP address from your router's DHCP server.

Trade-off: More accessibility means less isolation. In production, you'd add a firewall. For learning, bridged networking is perfect.

7. Filesystem Support: Choosing the Right Digital Filing Cabinet

7.1 What is a Filesystem Anyway?

Imagine a library with thousands of books. How do you organize them? Alphabetically? By genre? By color? The **filesystem** is the librarian's system for organizing, finding, and protecting books (files).

Different filesystems have different strengths:

- **Finding books quickly** (performance)
- **Preventing books from being damaged** (reliability)
- **Controlling who can read which books** (security)
- **Recovering books after a disaster** (journaling)

7.2 NTFS vs. ReFS vs. FAT: A Simple Analogy

Filesystem	Analogy	Best For
NTFS	A modern library with security cameras, climate control, and a computerized catalog	Windows system drives, shared folde anywhere security matters
ReFS	A brand-new library with self-healing shelves that detect and fix damage automatically	Large data volumes, virtualization storage, where data integrity is critica
FAT32	A community book swap box—simple, works everywhere, but no security or organization	USB drives, memory cards, devices th need to work with everything
exFAT	A bigger book swap box that handles larger books	External hard drives, flash storage wit large files

7.3 Why NTFS Wins for Windows Server

Windows Server 2012 R2 defaults to NTFS for the system drive because:

1. **Security:** NTFS supports **Access Control Lists (ACLs)**. You can say "User A can read this file, User B can edit it, User C can't see it at all."
2. **Reliability: Journaling** means NTFS keeps a log of changes. If the power fails during a file write, it can recover cleanly.
3. **Large Volume Support:** NTFS can handle partitions up to **256 TB** (theoretical limit).
4. **Features:** Encryption (BitLocker), compression, disk quotas, shadow copies (file versioning).

Real-world example: A company's HR department stores salary information on a server. With NTFS:

- Only HR managers can access the folder
- Files are automatically encrypted
- Every access is logged for auditing
- Previous versions are kept if someone accidentally deletes something

---

8. Advantages & Disadvantages: The Honest Truth

8.1 Why Virtualization Feels Like Cheating (In a Good Way)

Advantage 1: Safety First
Physical server installation is like performing surgery. One mistake can be catastrophic. Virtualization is like practicing on a medical simulator—all the learning, none of the risk.

Advantage 2: The Power of Snapshots
Before making a major change (like installing a new role), take a **snapshot**. This saves the exact state of the VM. If something breaks, revert to the snapshot. It's a time machine for your server!

Advantage 3: Resource Efficiency
One physical server can host **multiple virtual servers**. A company might run their email server, file server, and web server on the same physical machine as separate VMs. This saves money, power, and space.

Advantage 4: Portability
A virtual machine is just a collection of files. You can:

- Copy it to another computer
- Move it to a different virtualization platform
- Back it up to external storage
- Email it to a colleague (if compressed)

8.2 The Trade-Offs: What You Give Up

Disadvantage 1: Performance Overhead
Every layer between software and hardware adds overhead. Virtualization is like translating between languages—necessary but slower than native speech. For most

applications, the 5-15% performance loss is acceptable. For high-performance computing (scientific simulations, real-time trading), physical might still be better.

Disadvantage 2: Complexity
Instead of managing physical servers, you're managing:

- Physical hosts
- Hypervisor software
- Virtual networks
- Storage for VM files
- Snapshots and backups

More flexibility means more complexity.

Disadvantage 3: Licensing Costs
Some software licenses are per-physical-CPU. Virtualization can make licensing more complex and sometimes more expensive.

Disadvantage 4: The "It's Just a VM" Mentality
Because VMs are easy to create, people create too many ("VM sprawl"). Without discipline, you end up with dozens of forgotten VMs consuming resources.

---

9. Conclusion: Looking Back at What You Built

9.1 What You Actually Accomplished

When you started, you had:

- A laptop
- Some software downloads
- Theoretical knowledge

Now you have:

- A fully functional Windows Server 2012 R2 installation
- Understanding of virtualization concepts
- Hands-on experience with server configuration
- Troubleshooting skills

- Documentation of your process

  More importantly, you've crossed the threshold from **consumer** to **creator**. You don't just use technology; you build and control it.

9.2 From Classroom to Real World

Everything you learned translates directly to IT jobs:

- **Help Desk Technician:** "The user can't log in." Your Active Directory experience helps you understand what might be wrong.
- **Junior System Administrator:** "We need to set up a test server for the new application." You know how to create a VM and install an OS.
- **Network Engineer:** "The server isn't reachable from other subnets." Your networking troubleshooting helps diagnose the issue.

  You're not just learning for an exam; you're building a foundation for a career.

---

10. Future Outlook: Where to Go From Here

10.1 Next Steps in Your Learning Journey

Immediate Next Steps:

1. **Create a second VM** with a different OS (maybe Ubuntu Server)
2. **Network them together** (server-to-server communication)
3. **Set up file sharing** between the two
4. **Break something on purpose** and practice restoring from snapshot

Medium-Term Goals:

1. **Learn PowerShell** – the automation language for Windows Server
2. **Explore Active Directory deeper** – create users, groups, organizational units
3. **Set up a web server** (IIS on Windows, Apache on Linux)
4. **Learn basic security** – firewall configuration, user permissions

10.2 Industry Trends You Should Know

Containers (Docker, Kubernetes):
Think of containers as "lightweight VMs." They share the host OS kernel, making them faster to start and more efficient. Many applications are moving from VMs to containers.

Infrastructure as Code:
Instead of clicking through installation wizards, you write code that describes your desired state. Tools like Terraform, Ansible, and PowerShell DSC can recreate your entire server environment from a script.

Cloud Platforms (Azure, AWS, Google Cloud):
The cloud is essentially someone else's virtualized data center. Everything you learned about VMs applies directly to cloud instances.

Hybrid Environments:
Most companies use a mix of physical servers, on-premise VMs, and cloud services. Understanding all three makes you valuable.

10.3 Personal Recommendations

1. **Build a Home Lab:**

- Old computer + free virtualization software = unlimited learning
- Try different OS combinations
- Simulate a small company network

2. **Join Communities:**

- Reddit: r/homelab, r/sysadmin
- Discord servers for IT professionals
- Local tech meetups

3. **Document Everything:**

- Keep a digital notebook of what you learn
- Take screenshots of configurations
- Write down error messages and solutions

4. **Specialize Gradually:**

- Start broad (general server administration)
- Then focus (security, networking, cloud, etc.)
- Follow what interests you

Final Reflection

This project started as an assignment but became something more—a doorway into a profession. That moment when your virtual server booted successfully wasn't just a technical achievement; it was a confidence builder. You proved to yourself that you can understand complex systems, solve real problems, and build functional technology.

The screenshots throughout this document aren't just illustrations; they're milestones. Each one represents a decision, a learning moment, a small victory. From the initial VM creation screen to the fully operational Server Manager, you've traveled further than you might realize.

Remember: every expert was once a beginner who didn't give up. You've taken the first substantial step. Where you go from here is up to you, but you now have the foundation to build upon.