Table of contents

Labs

Assignments

Lab 1 - Environment Setup

Lab Preparations

Purpose of Lab 1

In this lab you will stand up the base environment you'll use for the course. You'll obtain official installation media and keys, prepare VMware Workstation, and deploy two servers:

- srv1: Windows Server 2025 Datacenter, GUI (Desktop Experience)
- srv2: Windows Server 2025 Datacenter, CLI (Core)

You'll complete essential post-install tasks (time zone, naming, activation, updates), then place both servers on a two-NIC layout so they can talk to each other on a private subnet while still reaching the internet for updates.

By the end, the machines are clean, consistent, and ready for later labs (security hardening, DNS, DHCP, and eventually AD).

Objectives

By the end of this lab, you will be able to:

- Acquire **Windows Server 2025 Datacenter** installation media and your individual product key from Azure Education and store them securely.
- Access **VMware Workstation** on a Seneca lab PC (locally or via MyApps) or install it on a personal PC.

- Provision two VMs with the required specs and networks:
  - srv1-`senecausername` (GUI) with NAT + VMnet10 NICs.
  - srv2-`senecausername` (Core) with NAT + VMnet10 NICs.
- Complete essential post-install tasks on each server.
- Configure NIC2 on each server for the internal lab subnet 10.0.`UID`.0/24:
  - srv1 → 10.0.`UID`.1/24
  - srv2 → 10.0.`UID`.2/24
- Verify basic Internet connectivity prerequisites for later labs within each VM.

Minimium Requirements

Before beginning, you must have:

1. Attended the Week 1 lecture.
2. Read through the Week 1 slides, and have them handy as a reference for concepts.
3. Working access to portal.azure.com with your Seneca credentials.
4. Access to VMware Workstation (Seneca lab PC or personal PC meeting course specs).
5. Your external SSD with at least 250 GB of dedicated space for this course.
6. Your assigned **UID** (from Blackboard Grades) handy for addressing.
7. Your physically printed **OSM620 Lab Logbook** for notetaking and saving commands.
8. Optional, but recommended: Caffeine delivery system.

Investigation 1: Downloading Installation Media

In this investigation, you will be downloading your Windows OS installation media by logging into your Seneca-based Azure account. You will also generate your personal serial keys.

1. Navigate to the Microsoft Azure site: portal.azure.com
2. Use your Seneca e-mail address and password to login.
3. Once on the main Azure page, look for the Search bar at the top of the page.
4. In the Search bar, type: **Education**, then hit Enter.
5. In the **Education | Overview** page, look to the left. You will see menu items already displayed on screen. (*Overview*, *Learning resources*, etc.)
6. Inside **Learning resources** in the left menu, click on **Software**.
7. In the main **Software** page, there is a Search bar just below the word Software (it says *Search* inside it.)
8. In that search field, type and enter: **Windows Server 2025 Datacenter**
9. In the item that appears below (there should only be one), click the link for **Windows Server 2025 Datacenter**.
10. On the right, an information box appears describing the software. Using your mouse to hover over this information box, scroll down to the bottom.
11. You should now see two items: **View Key** and **Download**
12. Click on **Download** first to begin downloading the Server 2025 ISO. You will need this for your operating system installation. (Don't forget where you've saved it!)
13. While the ISO file is downloading, click on **View Key**.
14. Copy this key into a text file that you save locally on your personal

computer or personal USB key. You will need this for the Server installation and for any reinstalls later in the semester. **Do not lose this key and do NOT share it with anyone!**

15. Repeat steps 8-14 for the **Windows 11 Education** ISO and serial key. You will need that for later.

Investigation 2: Using VMware Workstation

The use of *VMware Workstation* is required for this course. You can either use a Seneca Lab computer (Option 1a / Option 1b) or a personal computer (Option 2).

If you use an external SSD, you can use both!

> **WARNING:** Seneca Lab computers erase any locally saved data when restarted from their internal drives. Save all work to an external drive or online storage.

Option 1a: Seneca Lab Computers w/VMware Workstation Installed Locally

How do you know if your particular Seneca Lab classroom has VMware Workstation installed?

When you log in, the VMware Workstation icon will be visible on the desktop. If it is not, jump to *Option 1b*.

1. Log in to the Seneca workstation with your Seneca username (don't include @myseneca.ca) and password.
2. Once the desktop has fully loaded, double-click on the **VMware**

Workstation icon on the desktop. (If not present, jump to *Option 1b* instructions.)

3. When the **VMware Workstation** application appears on screen, continue to the next Investigation.

Option 1b: Seneca Lab Computers *without* VMware Workstation Installed

Do not use this option if your lab computer already has *VMware Workstation* installed. Go back to *Option 1a*.

1. Log in to the Seneca workstation with your Seneca username (don't include @myseneca.ca) and password.
2. Once the desktop has fully loaded, double-click on the **Seneca MyApps** icon on the desktop.
3. In the browser that launches, log in with your Seneca credentials.
4. Do not ignore dialog boxes! Read them carefully instead of just clicking *Cancel* or randomly.
5. In the search bar in the middle of the screen, type *VMware* and hit **Enter.**
6. One result should show up, **VMware WS Pro**. Click this result.
7. In the new *VMware WS Pro* page, click on the version number (Example: 17.6.1). This will launch the application over the network.
8. Click on allow for any dialog boxes that appear. Remember, **do not ignore dialog boxes!**
9. Be patient. It can take a bit for the background files to transfer over the network.
10. When complete, the **VMware Workstation** application should appear on screen. (You will also see the **VMware Workstation** icon on the desktop. You can launch it from here if you accidentally close the program.)
11. Proceed to the next investigation.

Option 2: Personal Computer

Part 1: Computer Hardware Requirements

To run this course and its tech on a personal computer, it must meet the following minimum system requirements:

1. CPU: **6-core Intel/AMD**
2. RAM: **32 GB**
3. Storage: **250 GB SSD free**, *reserved for this course* (if external, USB3.0 or higher)
4. Internet connection: **High speed, stable**

> **CPU architecture is vital.** You *cannot* use ARM or SnapDragon based computers for this course. This includes all Apple Silicon computers (M1/M2/M3/M4, etc).

Part 2: Registering Your Broadcom Account

Broadcom (formerly VMware) now requires account registration before you can download the VMware Workstation software. Follow the instructions below to do so. You only have to do this once. (If you already have an account using your Seneca e-mail, skip to Part 3.)

1. Visit the Broadcom registration site: profile.broadcom.com/web/registration
2. Provide your Seneca e-mail address and continue.
3. You will be sent a verification code. Check your Seneca e-mail and enter the code on the page.
4. On the *Complete your Registration* page, enter your information and click **Create Account.**
5. On the *Registered Successfully!* page, you can skip the rest and simply

click **I'll do it later**.

6. At the top right of the screen, click on the **Login** button. (No, you aren't logged in automatically. Yes, it's annoying.)
7. Enter your Seneca e-mail address as your username and the password you chose on Part 4 as your credentials.
8. When complete, you should see your name in the top right of the screen instead of *Login.* If not, attempt to log in again or ask for help.

Part 3: Downloading VMware Workstation

1. Go to this link: Download VMware Workstation
2. Click on **VMware Workstation Pro 17 for Windows**
3. Click on the highest version available. (17.6.3 as of last update)
4. On the next screen, click on the blue link inside *I agree to the Terms and Conditions* to open a new tab with their terms and conditions. Do not close this window.
5. Back in the previous window/tab that's still open, now click the checkbox for *I agree to the Terms and Conditions*.
6. Click on the little cloud icon next to the name of the download file.
7. Click the **Yes** button when the dialog box pops up.
8. In the *Trade Compliance and Download Conditions* screen, fill in the following school information:
    i. Address1: **1750 Finch Ave E**
    ii. City: **North York**
    iii. State/Province: **Ontario**
    iv. Country: **Canada**
    v. Zip/Postal Code: **M2J 2X5**
9. ***Finally*** click on the cloud icon again to begin the actual download.
10. If you ever need to download the software again, you won't have to go

through any of the registration mess. Follow Part 3, Steps 1-3, then Part 9.

Part 4: Install VMware Workstation

1. Find the downloaded .exe file and run the installer.
2. Follow all prompts and install.

Part 5: Launch VMware Workstation

1. Open **VMware Workstation** from your desktop or the Start Menu.
2. When asked, paste in your serial key.
3. Proceed to the next Investigation.

Investigation 3: VM1 Installation - Windows Server 2025 Datacenter (*srv1*)

- Hypervisor: **VMware Workstation**
- Name: **srv1-cjohnson30**
- RAM: **16 GB**
- CPU: **6 cores**
- Storage: **250 GB**
- Networking: **2 NICs**

Part 1: Setup Instructions

1. In the main window, you should see a large + symbol icon titled **Create a New Virtual Machine**. Click it.

2. In the new dialog box, keep *Typical* selected and click the **Next** button.
3. On the next screen, *Guest Operating System Installation*, do the following:
    i. Select *Installer disc image file (ISO):*
    ii. Now click **Browse**.
    iii. Navigate to where you saved your ***Windows Server 2025 Datacenter*** downloaded ISO and select it.
    iv. Once selected, the previous screen should now say:

    > **Windows Server 2025 detected.**
    >
    > This operating system will use Easy Install.

    v. If it doesn't, you haven't selected the right file, or your download was corrupted. **Ask for help.**
    vi. Click **Next**.
4. On the "Easy Install Information" screen, do the following:
    i. Paste in your serial key.
    ii. Version of Windows to install: Select *Windows Server 2025 Datacenter*
    iii. Personalize Windows:
        a. Full Name: **Administrator**
        b. Password (both fields): Select a strong password **you will remember**. You will use this same password for all VMs in this course.
        c. ***Do not select* "*Log on automatically*"**.
        d. Click **Next**.

5. On the "Name the Virtual Machine" screen, do the following:

    i. Virtual machine name: **srv1-*senecausername***

    > **Explanation:** For example, if my Seneca e-mail address is `cjohnson30@myseneca.ca`, then my Seneca username is *cjohnson30*. This would give me a VM name of *srv1-cjohnson30*.

    ii. Location: If using an external SSD (like with our lab computers), click **Browse** and navigate to your external SSD.

        a. Create the following directory structure in your SSD: *OSM620* > *Virtual Machines/srv1-cjohnson30*

        b. Select this new *srv1-cjohnson30* folder.

        c. Make sure you now see this change in the Location field.

        > **Example:** *Z:/OSM620/Virtual Machines/srv1-cjohnson30*

    iii. Click **Next**.

6. On the "Specify Disk Capacity" screen, do the following:

    i. Maximum disk size (GB): **250**
    ii. Select *Split virtual disk into multiple files*.
    iii. Click Next.

7. On the "Ready to Create Virtual Machine" screen, do the following:

    i. Click on **Customize hardware...**

8. On the new "Hardware" screen, do the following:

    i. Select *Memory*, and change the value to: **16384**

ii. Select *Processors*, and change:
    a. Number of processors: **1**
    b. Number of cores per processor: **6**
    c. Virtualize Intel VT-x/EPT or AMD-V/RVI: **Checked**
    d. Virtualize CPU performance counters: ***Unchecked***
    e. Virtualize IOMMU (IO memory management unit): **Checked**
iii. Select *Network Adapter* and confirm:
    a. *Connected at power on:* **Checked**
    b. *NAT*: **Checked**
iv. Click on the ***Add...*** button on the bottom left of the *Hardware* window.
    a. Select *Network Adapter* and click **Finish**.
    b. Back in the *Hardware* window, click on *Network Adapter 2*.
    c. Under *Network connection*, click the **Custom: Specific virtual network** radio button.
    d. Just below that, click the drop-down (it likely says *VMnet0* by default). Find and select **VMnet10**.
    e. Click **Close**.

9. Back in the "Ready to Create Virtual Machine" screen, click **Finish**.

10. The virtual machine should launch.

11. If you get a dialog box about *Side channel mitigations*, check the box for *Do not show this hint again* and click **OK**.

12. Your new Virtual Machine should now finish creating and then turn on and begin the OS installation.

13. Windows installation is automated at this point and won't require any input from you. It may restart several times.

> **Time Note:** Installation may take some time.
>
> Feel free to get some caffeine or make a sandwich.

14. Eventually, you will be presented with the desktop and the VMware Tools installer having completed and asking if you'd like to restart. Choose **Yes**.

15. Once you've restarted, your installation is complete.

Investigation 4: Post-Installation Tasks (*srv1*)

After installing a new operating system, there are always a number of **post-installation tasks** to complete. **These aren't optional!**

Part 1: Applying Time Zone Settings

This one is fairly straight-forward. Having the proper time zone set (EST) is essential for proper time keeping and ensuring encrypted webpages connect properly.

1. In the *Server Manager* application, click on **Local Server** in the left-hand menubar.
2. In the main *Properties* area, on the right-hand column, look for the *Time Zone* line. It should say **(UTC-05:00) Eastern Time (US & Canada)**.
3. If the *Time Zone* line item doesn't say the above, click on the displayed time zone and change it to UTC-05:00 as seen above.

Part 2: Server Name Change

The default name applied to your new server will be semi-randomized. For

proper identification (and to not wonder which server you're on when you have several), we're going to change this.

1. In the *Server Manager* application, click on **Local Server** in the left-hand menubar.
2. In the main *Properties* area, on the left-hand column, look for the *Computer name* line.
3. Click the current computer name.
4. In the *System Properties* dialog box that pops up, find the **Change** button and click it. (Ignore the *Computer Description* field. It's tempting, but wrong!)
5. In the new *Computer Name/Domain Changes* dialog box that pops up, find the *Computer name* field. Replace it with **srv1-SenecaUsername**.
6. When you click **OK**, the system will warn you about restarting. Choose to restart the system when asked.
7. Once you've restarted and logged back in, go back to the *Server Manager* from Part 1 and double-check your new computer name is correct. **Do not skip this step!**
8. If it is, you're done!

Part 3: Windows Activation

Activating Windows unlocks certain settings and features. Since you've used your valid serial key (right?), you can activate with Microsoft easily.

1. In the *Server Manager* application, click on **Local Server** in the left-hand menubar.
2. In the main *Properties* area, on the right-hand column, look for the *Product ID* line.
3. Click on the **Not activated** link.
4. Follow the instructions in the popup dialog box. If unable to activate easily,

ask your professor for help.

Part 4: Installing OS Updates

A critical part of a security-conscious mindset is running regular updates. **This is NOT something you do only once at the start of installation.** You should be running these regularly to keep up to date with security fixes and zero-day exploits.

1. If you're already in the *Settings* application, look through the left menubar for **Windows Update** and click it. (Otherwise, click on the *Start* menu and search for **Updates**.)
2. In the *Windows Update* main screen, scroll down to **Advanced Options** and click it.
3. The very first option is *Receive updates to other Microsoft products*. Toggle this from **Off** to **On**.
4. At the top of the screen, where it says *Windows Update > Advanced Options*, click **Windows Update** to go back to the previous screen.
5. You will likely already see updates ready. Click on **Download & install all**.
6. As you might expect, this can take a while. Timing will depend on your Internet connection, how fast your computer is, how fast your SSD is, and how many updates there are. Please be patient. Your computer may restart.
7. Once updates have begun, take a break while it does its thing. Grab a drink, make a sandwich, text a friend.
8. After updates are complete, go back into *Windows Update* and click **Check for updates** again. There may be (and often times are) more.
9. If there are more updates, complete Steps 5-8 again until there are no more updates available.
10. In *Windows Update*, scroll back down to **Advanced options** again and

click it.

11. Inside *Windows Update > Advanced options* scroll down to **Optional updates** and click it.
12. Select all available updates that appear (you may have to expand some lists).
13. Click **Download & install**.

Part 5: Configuring Network Interface Card 2 (NIC2)

We have two network interfaces on this virtual machine. NIC1 is set to DHCP and is our Internet connection. We don't touch that one. It's configuration is automatic.

However, NIC2 requires manual configuration. Why do we need NIC2? It's what we'll be using to communicate with *srv2* and our other VMs.

1. In the *Server Manager* application, click on **Local Server** in the left-hand menubar.
2. In the main *Properties* area, on the left-hand column, look for the *Ethernet1* line.
3. Click on the **IPv4 address assigned by DHCP** link.
4. In the *Network Connections* window that pops up, right-click on the *Ethernet1* icon.
5. Click *Rename* from the drop-down, and rename the adapter to: **Internal Network**
6. If that doesn't work, select the *Ethernet1* icon and hit **F2** on your keyboard to rename as above.
7. Right-click on the renamed *Internal Network* icon, and select *Properties* from the drop-down menu.

8. In the new *Internal Network Properties* dialog box, find the *Internet Protocol Version 6 (TCP/IPv6)* entry and uncheck it.
9. Now, look for the *Internet Protocol Version 4 (TCP/IPv4)* entry, select it (do not uncheck), and look for the **Properties** button just below and click it.
10. In the new *Internet Protocol Version 4 (TCP/IPv4) Properties* dialog box, select the *Use the following IP address:* radio button.
11. In the fields directly below, enter the following:

    i. IP address: **10.0.*UID*.1**

    > *UID* is a placeholder. In Blackboard in Grades, there is a UID column. Each student has been assigned a unique ID number for this course. You will use that number in the address above.
    >
    > For example, if my UID on Blackboard is *40*, then my address in the step above is **10.0.40.1**. (Yours will be different!)

    ii. Subnet mask: **255.255.255.0**

    iii. Leave all other fields blank.

    iv. Click **OK**.

    v. Back in the *Internal Network Properties* dialog box, click **Close**.

    vi. Switch back to the *Server Manager > Local Server* application, and hit the *Refresh* icon. (It's the icon that looks like a music shuffle symbol to the left of the flag icon at the top right.)

    vii. You should now see: **Internal Network 10.0.*UID*.1**

    viii. Go back to the *Network Connections* window, and find the other adapter icon. (Likely called *Ethernet0*)

ix. Rename this to: **External Network**

x. Go back to *Server Manager* > *Local Server*, refresh, and confirm you can see the new *External Network* name.

xi. If you do, you're done!

Part 6: Internet Connectivity Check w/Edge

We'll double-check we can access the Internet using the built-in Microsoft Edge application.

1. Open *Microsoft Edge*.
2. Go through the first-run questions.
3. When able, use the browser to navigate to **eff.org**.
4. If the website loads, move on to the next step. If not, **ask your professor for help**.

Part 7: Download and Install Firefox

There are a ton of feature and privacy reasons *not* to use Microsoft Edge. Instead, we'll download and install **Mozilla Firefox** and use that going forward.

1. Open *Microsoft Edge*.
2. Navigate to: Download Mozilla Firefox
3. Wait for the installer to finish, then open it.
4. Follow the installer instructions.
5. Once complete, open *Firefox* and navigate to **eff.org** to check everything is okay.
6. Close *Microsoft Edge* forever.

Investigation 5: VM2 Installation - Windows Server 2025 Core (*srv2*)

- Hypervisor: **VMware Workstation**
- Name: **srv2-cjohnson30**
- RAM: **4 GB**
- CPU: **2 cores**
- Storage: **250 GB**
- Networking: **2 NICs**

Part 1: Setup Instructions

1. In the main window, you should see a large + symbol icon titled **Create a New Virtual Machine**. Click it.

2. In the new dialog box, keep *Typical* selected and click the **Next** button.

3. On the next screen, *Guest Operating System Installation*, do the following:

    i. Select *Installer disc image file (ISO):*

    ii. Now click **Browse**.

        a. Navigate to where you saved your ***Windows Server 2025 Datacenter*** downloaded ISO and select it.

        b. Once selected, the previous screen should now say:

> **Windows Server 2025 detected.**
>
> This operating system will use Easy Install.

    c. If it doesn't, you haven't selected the right file, or your download was corrupted. **Ask for help.**

iii. Click **Next**.

4. On the "Easy Install Information" screen, do the following:

   i. Paste in your serial key.
   ii. Version of Windows to install: Select *Windows Server 2025 Datacenter **(Core)***
   iii. Personalize Windows:
      a. Full Name: Administrator
      b. Password (both fields): Select a strong password **you will remember**. You will use this same password for all VMs in this course.
   iv. ***Do not select* "*Log on automatically*"**.
   v. Click **Next**.

5. On the "Name the Virtual Machine" screen, do the following:

   i. Virtual machine name: **srv2-*senecausername***

      > **Explanation:** For example, if my Seneca e-mail address is `cjohnson30@myseneca.ca`, then my Seneca username is *cjohnson30*. This would give me a VM name of *srv2-cjohnson30*.

   ii. Location: If using an external SSD (like with our lab computers), click **Browse** and navigate to your external SSD.

      a. Create the following directory structure in your SSD: *OSM620 > Virtual Machines > srv2-cjohnson30*
      b. Select this new *srv2-cjohnson30* folder.

    c. Make sure you now see this change in the Location field. (Example: *Z:/OSM620/Virtual Machines/srv2-cjohnson30*)

iii. Click **Next**.

6. On the "Specify Disk Capacity" screen, do the following:

   i. Maximum disk size (GB): **250**
   ii. Select *Split virtual disk into multiple files*.
   iii. Click Next.

7. On the "Ready to Create Virtual Machine" screen, do the following:

   i. Click on **Customize hardware...**

8. On the new "Hardware" screen, do the following:

   i. Select *Memory*, and change the value to: **4096**
   ii. Select *Processors*, and change:
      a. Number of processors: **1**
      b. Number of cores per processor: **2**
      c. Virtualize Intel VT-x/EPT or AMD-V/RVI: **Checked**
      d. Virtualize CPU performance counters: ***Unchecked***
      e. Virtualize IOMMU (IO memory management unit): **Checked**
   iii. Select *Network Adapter* and confirm:
      a. *Connected at power on:* **Checked**
      b. *NAT*: **Checked**
   iv. Click on the ***Add...*** button on the bottom left of the *Hardware* window.
      a. Select *Network Adapter* and click **Finish**.
      b. Back in the *Hardware* window, click on *Network Adapter 2*.
      c. Under *Network connection*, click the **Custom: Specific virtual network** radio button.

    d. Just below that, click the drop-down (it likely says *VMnet0* by default). Find and select **VMnet10**.
    e. Click **Close**.

9. Back in the "Ready to Create Virtual Machine" screen, click **Finish**.

10. The virtual machine should launch.

11. If you get a dialog box about *Side channel mitigations*, check the box for *Do not show this hint again* and click **OK**.

12. Your new Virtual Machine should now finish creating and then turn on and begin the OS installation.

13. Windows installation is automated at this point and won't require any input from you. It may restart several times.

    > **Time Note:** Installation may take some time.
    >
    > Feel free to get some caffeine or make a sandwich.

14. Eventually, you will be presented with the Command Prompt window open to the *SConfig* text-based application, and the VMware Tools installer having completed and asking if you'd like to restart. Choose **Yes**.

15. Once you've restarted, your installation is complete.

Investigation 6: Post-Installation Tasks (*srv2*)

After installing a new operating system, there are always a number of **post-installation tasks** to complete. **These aren't optional!**

Part 1: Applying Time Zone Settings

This one is fairly straight-forward. Having the proper time zone set (EST) is essential for proper time keeping and ensuring encrypted webpages connect properly.

1. In the *SConfig* application, select Option 9 (*Date and Time*). Use your keyboard.
2. The *Date and Time* dialog box pops up.
3. Look for the *Time Zone* line. It should say **(UTC-05:00) Eastern Time (US & Canada)**.
4. If the *Time Zone* line item doesn't say the above, click on the *Change time zone...* button and change it to UTC-05:00 as seen above.
5. Click **OK** to close out of *Date and Time*.
6. Back in *SConfig*, choose Option 9 again to confirm your changes have stuck. If yes, continue to Part 2.

Part 2: Server Name Change

The default name applied to your new server will be semi-randomized. For proper identification (and to not wonder which server you're on when you have several), we're going to change this.

1. In the *SConfig* application, selection Option 2 (*Computer name*). Use your keyboard.
2. In the new *Computer name* screen, enter your new computer name in the waiting text field: **srv2-SenecaUsername**
3. Press **Enter** on your keyboard to confirm the change.
4. The system now asks you about restarting. Enter **Y** to choose yes and hit the **Enter** key to confirm.

5. Once you've restarted and logged back in, go back to the *Computer name* screen from Part 2 and double-check your new computer name is correct. **Do not skip this step!**
6. If it is, you're done!

Part 3: Windows Activation

Activating Windows unlocks certain settings and features. Since you've used your valid serial key (right?), you can activate with Microsoft easily.

1. In the *SConfig* application, select Option 11 (*Windows activation*). Use your keyboard.
2. In the new *Windows activation* screen, enter **2** (*Activate Windows*) and hit **Enter** to begin the activation process with Microsoft.
3. Follow the instructions on screen. If unable to activate easily, **ask your professor for help**.

Part 4: Installing OS Updates

A critical part of a security-conscious mindset is running regular updates. **This is NOT something you do only once at the start of installation.** You should be running these regularly to keep up to date with security fixes and zero-day exploits.

1. In the *SConfig* application, select Option 5 (*Update settings*). Use your keyboard.
2. In the new *Update setting* screen, enter **5** (*Opt-in to Microsoft Update*) and hit **Enter** to confirm.
3. The next screen will ask you confirm again. Enter **y** and hit **Enter** to continue. (Then, hit **Enter** again.)
4. In the *SConfig* application, select Option 6 (*Install updates*). Use your

keyboard.

5. In the new *Install updates* screen, enter **1** (*All quality updates*) and hit **Enter** to confirm.
6. After a short check, you are asked which updates to install. Select **a** to install all updates and hit **Enter** to confirm.
7. As you might expect, this can take a while. Timing will depend on your Internet connection, how fast your computer is, how fast your SSD is, and how many updates there are. Please be patient.
8. Once updates have begun, take a break while it does its thing. Grab a drink, make a sandwich, text a friend.
9. If asked to restart, choose **yes**.
10. After updates are complete, go through Steps 4-6 again. Do so until the system tells you there are no new updates. (It may take a few cycles to get them all.)
11. When complete, shut down *srv2*. Use the on-screen menu options in *SConfig* to do so.

Part 5: Configuring Network Interface Card 2 (NIC2)

We have two network interfaces on this virtual machine. NIC1 is set to DHCP and is our Internet connection. We don't touch that one. It's configuration is automatic.

However, NIC2 requires manual configuration. Why do we need NIC2? It's what we'll be using to communicate with *srv1* and our other VMs.

1. In the *SConfig* application, select Option 8 (*Network Settings*).

2. In the *Network settings* screen, look for the unconfigured network adapter. It's likely the second one, and will likely have an address starting with **169.**

If you aren't sure, ask your professor for help.

3. Select that adapter using the menu option and your keyboard, and hit **Enter** to confirm.

4. In the *Network adapter settings* screen, select Option 4 (*Rename network adapter*).

5. Enter the new network adapter name in the field: **Internal Network**

6. Hit **Enter** twice to confirm.

7. When the *Network adapter settings* screen refreshes, you should see the new name. If not, repeat Steps 4-6 again, or ask for help from your professor.

8. Now, select Option 1 (*Set network adapter address*).

9. Select **S** for *Static IP address*.

10. Enter your new IP address for this machine. It will take the following form: **10.0.*UID*.2**

    > *UID* is a placeholder. In Blackboard in Grades, there is a UID column. Each student has been assigned a unique ID number for this course. You will use that number in the address above.
    >
    > For example, if my UID on Blackboard is *40*, then my address in the step above is **10.0.40.2**. (Yours will be different!)

11. In the *Enter subnet mask* field, stick with the default by keeping the field blank and hitting **Enter**.

12. In the *Enter default gateway* field, use: **10.0.*UID*.1**

13. Hit **Enter** once the configuration has completed.

14. Back in the *Network adapter settings* screen, confirm your changes from above. You should see them all displayed. If not, repeat the missing steps or as your professor for help.

15. If everything looks good, hit **Enter** to go back to the main menu screen.

16. Go back into *Network settings* and change the other network adapter's name to: **External Network**

17. Make no other changes to this adapter.

18. You're done!

Lab 2 – Hyper-V, NAT, and Windows Clients

Investigation 1: Hyper-V Role Installation and Switch Setup *(srv1)*

In this investigation, you will install the Hyper-V role on srv1 and create a Hyper-V virtual switch that uses your existing Internal Network adapter.

Your Hyper-V switch will be named Internal Network (Hyper-V).

Part 1: Install the Hyper-V Role

1. Open the **Server Manager** application and go to **Add Roles and Features**.
2. **Installation Type:** Keep Role-based or feature-based installation selected.
3. **Server Selection:** Ensure your local server (*srv1*) is selected.
4. **Server Roles:** Check ***Hyper-V***.
    i. When prompted to "*Add features that are required for Hyper-V*", click **Add Features**.
5. **Features:** Leave defaults (the *Hyper-V Management Tools* are already included).
6. **Hyper-V page**:
    i. When asked about Virtual Switches, select **Internal Network** only.
    ii. Migration for live migrations: leave unchecked.
    iii. Default Stores: leave defaults.

7. Confirmation: Check "*Restart the destination server automatically if required*".
8. Click **Install**.
9. Wait for installation to complete. If prompted, allow the reboot. Log back in when finished.

> **Note:** It’s normal for *Server Manager* to take a minute or two after reboot to show the Hyper-V tile.

Part 2: Rename the Hyper-V Virtual Switch

As part of our Hyper-V role installation, we selected the *Internal Network* (Part 1, Step 6.1).

This has created a virtual switch on the server. For easy management, we'll rename it.

Our new virtual switch name will be: `HQ Network`

1. In Server Manager, click Tools (top-right) and select Hyper-V Manager.
2. In the left pane, click your server name (e.g., srv1-...).
3. In the right Actions pane, click Virtual Switch Manager….
4. Under the Virtual Switches column on the left, look for the switch that's already been created. It will likely have a name similar to: `Intel(R) 82574L Gigabit Network Connection - Virtual Switch`
5. In the Virtual Switch Properties window:
    i. Name: Change to `HQ Network`

ii. Ensure the **External network** radio button is selected.

> **Reminder:** This is the adapter you renamed earlier to Internal Network and set to 10.0.`UID`.1.

iii. Ensure "*Allow management operating system to share this network adapter*" is checked.

iv. Click OK. If you see a warning about temporary network disruption, click Yes.

> **Naming note:** You are creating an External Hyper-V switch, but you're binding it to the Windows NIC you named Internal Network.
>
> That's correct.
>
> You will end up with a third network adapter called: `vEthernet (HQ Network)`

6. Go back to *Server Manager > Local Server* and refresh. You should now see the following two networks:

   - **External Network**: Assigned by DHCP
   - **vEthernet (HQ Network)**: IPv4 addess assigned by DHCP, IPv6 enabled

Part 3: Configure Network Setting for Hyper-V Switch

Now that we've created the Hyper-V network adapter, we have to give it an IP address. This will let our Hyper-V VMs talk to the server, and eventually, bridge our Internet connection.

1. In *Server Manager > Local Server*, click on the entry next to **vEthernet (HQ Network)**. This opens the *Network Connections* window.
2. Verify you now see a third network adapter: `vEthernet (HQ Network)`
3. Right-click it → Properties and do the following:
    i. Internet Protocol Version 6 (TCP/IPv6): **Unchecked**
    ii. Internet Protocol Version 4 (TCP/IPv4): Select and click **Properties**
        a. IPv4 Address: **10.0.`UID`.1**
        b. Subnet Mask: **255.255.255.0**
        c. Leave all other fields blank and click OK.
4. Back in *Server Manager > Local Server*, refresh. You should now see the following two networks:
    - **External Network**: Assigned by DHCP
    - **vEthernet (HQ Network)**: 10.0.`UID`.1

Investigation 2: RRAS and NAT

Our server now has access to two networks:

- **External Network**: Connection to the Internet
- **vEthernet (HQ Network)**: Internal network only between *srv1* and the Hyper-V VMs we'll create in Investigation 3. This network *does not* have access to the Internet.

If we want to give our Hyper-V machines Internet access, we need to route network traffic between the two networks using *srv1* as a bridge. This keeps our Hyper-V machines on a secure network, and Internet access is handled and monitored by our Windows Server.

This will involve setting up routing and network address translation (NAT).

Part 1: Installing the Remote Access Role

1. Open the **Server Manager** application and go to **Add Roles and Features**.
2. **Installation Type:** Keep Role-based or feature-based installation selected.
3. **Server Selection:** Ensure your local server (*srv1*) is selected.
4. **Server Roles:** Check ***Remote Access***.
5. **Features:** Leave defaults (the *Hyper-V Management Tools* are already included).
6. **Remote Access page:** Click Next.
    i. The **Add Roles and Features Wizard** dialog box appears. Leave defaults and select *Add Features*.
7. **Remote Access > Role Services page:** Check the following and click *Next* when ready:
    - Direct Access and VPN (RAS)
    - Routing
8. Keep defaults and select *Next* on the following pages:
    i. Web Server Role (IIS)
    ii. Web Server Role (IIS) > Role Services
9. Confirmation: Click **Install**.
10. In the **Results** page: When installation has completed, click on the "*Open the Getting Started Wizard*" link.
11. The wizard opens. Make no changes and close it. (This just removes the system nagging.)
12. Back in **Results**, click Close.

Part 2: Enabling NAT with RRAS

You will now install the tools for routing and NAT bridging discussed at the top of this investigation using the **Routing and Remote Access Server** (RRAS) we've just installed.

1. Open the **Server Manager** application and go to **Tools > Routing and Remote Acceess**.
2. This opens the *Routing and Remote Access* application.
3. In the right-hand column, find SRV1, right-click it, and select **Configure and Enable Routing and Remote Access**.
4. The setup wizard appears. Click *Next*.
5. **Configuration:** Select the *Custom configuration* radio button.
6. **Custom Configuration:** Select (checkbox) only the following:
    - NAT
    - LAN routing
7. **Summary window**: Click *Finish*.
8. The **Routing and Remote Access** service status dialog box appears. Click *Start service*.
9. It may take a few moments for the service to fully start. Be patient.
10. Once the *Routing and Remote Access* application reappears, move to the next part.

Part 3: Configuring NAT with RRAS

It's now time to configure the routing and NAT bridging between the two networks.

1. Open the **Routing and Remote Access** application.

2. In the right-hand column, expand *SRV1*, then expand *IPv4*.
3. Right-click on *NAT* and select *New Interface*.
4. Select the **External Network** entry and click *OK*.
5. In the *Network Address Translation Properties* window, select the following and click OK when ready:
    - Public interface connected to the Internet
    - Enable NAT on this interface
6. Right-click on *NAT* and select *New Interface* again.
7. Select the **vEthernet (HQ Network)** entry and click OK.
8. In the *Network Address Translation Propertie*s window, select the following and click OK when ready:
    - Private interface connected to private network
9. Back in the *NAT* window, you should see two created interfaces:
    - External Network
    - vEthernet (HQ Network)
10. Keep the application open on *NAT* and minimize. We'll come back to it later.
11. That's it! Move on to the next Investigation.

Investigation 3: Installing Windows Client 1 with Hyper-V *(client1)*

Part 4: Attach VMs (when you create them)

When you begin creating Hyper-V VMs on srv1, select Internal Network (Hyper-V) for the VM’s Network Adapter.

Do not select the “Internal” switch type here. Your switch is an External Hyper-V switch bound to the Internal Network NIC, and its exact name is Internal

Network (Hyper-V).

Hyper-V is now installed on srv1, and your VMs can now use the Internal Network (Hyper-V) for the lab subnet.

Investigation 4: Post-Installation Tasks *(client1)*

Investigation 5: Cloning client1 to Create client2

Lab 3 - Security and Remote Connectivity

Lab Preparation

Purpose / Objectives of Lab 3

In this lab, you will conduct several Windows system administration tasks to secure your servers against would-be attackers and gain preliminary experience with the command line interface.

If you encounter technical issues, please contact your professor via e-mail or in your section's Microsoft Teams group.

Minimum Requirements

Before beginning, you must have:

1. Successfully completed Lab 2.
2. Attended the Week 3 lecture
3. Read through the Week 3 slides, and have them handy as a reference for concepts.
4. Your external SSD (or personal computer) with your VMs from Labs 1 and 2.
5. Your VM login credentials.
6. Optional, but recommended: Caffeine delivery system.

Key Concepts

Security: From the Beginning

In the not-too-distant past, companies would focus on getting their product and systems working and relegating security as their last step, often as an afterthought. When security is only considered at the end of a project, it's very difficult to remember all the ways in which your product interacts and things can get missed.

This created several high-profile breaches in the 90s and early 2000s, and our approach to security had to be reconsidered.

As a result, we now consider security **from the beginning**. As you create applications, add users to databases, create links between services, you *must* keep security in mind at every step of development. Securing as you go is the best method, but even something as simple as simply documenting unsecured parts of your code as you go can be enough (assuming you go back and fix them!)

Generally, we apply the concept of Principle of Least Privilege to security. Essentially, this boils down to locking everything down as much as possible and only allowing what and who you need through. Open access makes you a target. You'll be applying this principle to the firewall later in this lab.

We also take a look at defaults. Most systems and software come with pre-configured defaults to make out-of-the-box setup easy. This can take the form of a default username and password, default ports, etc. In a well secured system, these are often changed to avoid hack attempts. If you know the default, there's a high chance that hackers know it as well. You'll be changing some defaults in this lab.

This is not an exhaustive list of applied security, but it does give use a bit of working knowledge. You'll need it for this lab as well as in our later work.

Firewalls

In short, **a firewall is a utility that sits on your computer between your network connection and the rest of your system.**

Any application, service, or other data that is sent or received by your computer goes through your firewall first. The dominant network protocol is TCP/IP, which means we're dealing with *packets*.

A firewall looks at these packets.

To be clear, the firewall doesn't look *inside* packets, but just at the outside data like IP address and/or port destination, etc. The actual transmitted data is still secure and unread.

Generally with firewalls, we apply the *Principle of Least Privilege* by dropping all new connections by default, and allowing a few exceptions. This is known as **whitelisting**.

Editing Text Files

As you will sometimes be working in the Windows PowerShell command line environment, it is useful to learn a least one common method of editing text files.

Although programmers and developers usually use graphical IDE's to code and compile programs (Visual Studio, Sublime, Eclipse, etc), they can create source code using a text editor and compile their code directly on the server to generate executable programs (without having to transfer them for compilation or execution).

Developers very often use a text editor to modify configuration files. In this course, you will become familiar with the process of installing, configuring, and running network services. Text editors are an important tool to help this setup, but are also used to "tweak" or make periodic changes in service configurations.

The most readily-available command line text editor built into Windows is **Notepad**.

However, Notepad is not available in Server Core. To edit a text file in that environment, we have three main options:

1. Use PowerShell's object-oriented programming to send an edit directly into a text file. This is cumbersome and is not interactive (you don't see the text file on screen), but it is the only built-in option.
2. Transferring the file to a different computer that does have a text editor (like your Windows 11 client), modifying the file there, and transferring it back to your Server Core machine.
3. Using **Visual Studio Code** to connect remotely the the environment. This is by far the coolest and most convenient option.

We'll be doing all three in this lab to show you how, but going forward, VS Code will be our go-to for interacting with Server Core in most instances. (You will be tested on all three options, so don't skip them!)

Investigation 1: Windows Defender (Firewall)

In this investigation, we're going to take a look at the Windows Defender firewall to see how firewall rules can be applied to secure our servers and let the few things we want to allow through.

It should be noted that the Windows Defender firewall *should not be your only*

defense. Production environments will often have managed networks that restrict access on a larger scale, some of which you will get into with your networking courses.

These work together, though we will only be focusing (mostly) on the Windows firewall in this course.

Part 1: Windows Server GUI (srv1) - Configuring the IIS Role

On this server, we will configure our first server role, IIS (a web server). This will be used only for testing connectivity between your other VMs and working with the firewalls.

We will spend more time with Server Roles in a later lab. For now, simply follow the instructions.

1. In the *Server Manager* application, in the menu bar on the left side of the window, click on **Local Server**.
2. Now, at the top right of the window, click **Manage** > **Add Roles and Features**.
3. The *Add Roles and Features Wizard* dialogue box pops up.
4. On the first page, *Before you begin*, check the box next to **Skip this page by default**, then click **Next**. This allows us to skip this page any other time we want to add a role or feature in later labs.
5. On the *Installation Type* page, select **Role-based or feature-based installation** and click **Next**.
6. On the *Server Selection* page, select **Select a server from the server pool**, then **srv1-username**, and finally **Next.**
7. Note: In later labs, we'll be able to target other servers for remotely installing features. Here, we're selecting the local server.
8. On the *Server Roles* page, this is where we'll select our IIS web server. Scroll

through the listing until you find **Web Server (IIS)**, check the box next to it, and click **Next**.

9. A secondary dialogue box, *Add features that are required for Web Server (IIS)?*, pops up. Make sure the **Include management tools (if applicable)** box is checked, then click **Add Features**.
10. Back in the *Server Roles* page, click **Next** again.
11. On the *Features* page, leave all settings as their defaults and click **Next**.
12. On the *Web Server Role (IIS)* page, click **Next**.
13. On the *Role Services* page, keep all defaults and click **Next**.
14. Finally, on the *Confirmation* page, click **Install**.
15. This may take a few minutes. Please be patient.
16. When the wizard has the status message "*Installation succeeded on srv1.*", setup is complete and you can click **Close**.
17. Verify your local web server works by opening *Firefox* and going to the following address: **127.0.0.1**
18. If you see the **Internet Information Services** splash page, you've successfully completed the installation and can move on to the next part!

(Insert instructions for loading this page on the Windows 11 client to test connectivity.) (Insert instructions for loading this page on Windows Server Core if possible, even just unrendered HTML code.)

Part 2: Testing VM Network Connectivity

As part of your Lab 1 environment setup, you tested your connection to the Internet from each of your 3 virtual machines.

It's now time to test if we can use each VM to connect to the others.

1. On your Windows Server GUI (srv1) VM, open Command Prompt.
2. Grab this VM's IP address with the following command and write it down:

`ipconfig`

3. On your Windows Client (client 1) VM, open Command Prompt.
4. Enter the following command to test our ability to talk to srv1: `ping "srv1-ipaddress"` where *srv1-ipaddress* is the IP address from Step 2.
5. Example: `ping 192.168.1.14`
6. It doesn't work, does it? That's normal at this stage and we'll fix it below. For now, we'll try connecting to our web server on srv1. That should show us the two VMs can talk to each other.
7. Open *Firefox* on your **Windows 11 Client (client1)** and use the IP address from Step 2 (your srv1 IP address) as the URL.
8. Example: `http://192.168.1.14`
9. Can you see the IIS splash page that you saw in Part 1, Step 17? If so, you have connectivity!
10. Continue to Part 3.

Part 3: Windows Server GUI (srv1) - Applying Firewall Rules

We will now apply our security-conscious policy by configuring our firewall on this server.

At the moment, the server's firewall is configured using defaults. As mentioned above, defaults are a security risk as they are known to everyone and can be used against you. If an attacker *doesn't* know your configuration, it's harder for them to know what's open and what's not and how to attack.

As you saw from Part 2, ping between client1 and srv1 didn't work. This is because, by default, the ability to ping a server is turned off. A ping (a type of ICMP packet) is typically used to see if you can get a response from a server. Having a server respond let's an attacker know there's a machine there that they can then try to break into.

We want to turn on ping so we can test connections between our machines, but we have to be careful. Turning that on through the firewall too broadly opens us up to that vulnerability.

We are going to turn it on *only* for our local network. Our VMs will be allowed to ping each other, but anything outside of our subnet (192.168.1.0/24) can't. Best of both worlds.

To do this, we're going to work with the **Windows Defender Firewall** on srv1.

1. In the *Server Manager* application, go to *Local Server*.
2. In the *Properties* section of this page, look for the **Microsoft Defender Firewall** line item.
3. Next to the line, it should say **Public: On**. Click on this.
4. The *Windows Security > Firewall & Network protection* application opens.
5. On this screen, you will see three networks: *Domain network*, *Private network*, and *Public network*.
6. We'll spend more time with these in later weeks, but for now, all three should say: **Firewall is on**.
7. Below this, click on the link that says: **Advanced settings**
8. This opens the *Windows Defender Firewall with Advanced Security* application.
9. On the first page, you'll notice the same three profiles: *Domain Profile*, *Private Profile*, and *Public Profile*.
10. All three have the same overall rules:
11. **Windows Defender Firewall is on**: This just confirms the firewall is active for this network profile.
12. **Inbound connections that do not match a rule are blocked**: This means that all incoming connections and requests (like trying to ping this machine) are blocked *by default*. If you want to allow a certain type of connection or service into your server, you have to make a specific rule for

it. This is whitelisting and our *Principle of Least Priviledge* in action.

13. **Outbound connections that do not match a rule are allowed**: This means all network data leaving the server is automatically allowed. The logic here is that if the server is the one deciding the send out information, it's likely fine. (Note: The only time we ever whitelist outbound connections is for specialized security settings like government compliance.)
14. At the moment, we're dealing with the *Public Profile* context. Let's allow ICMP ping!
15. On the left-hand menu bar, click **Inbound Rules**.
16. This loads a ton of already-defined rules. Thankfully, the one we need has already been defined. We just need to turn it on.
17. At the top of this area, click **Name** to order the list by name (this is not the default).
18. Now, scroll down until you can find the following and double-click it: **Core Networking Diagnostics - ICMP Echo Request (ICMPv4-In)**, **Private, Public**.
19. **Note:** There are two of these rules! Look for the one that has **Private, Public** under the Profile header. The wrong one says Domain.
20. This opens this rule's configuration window, starting on the *General* tab.
21. In the *General* tab, look for the following and check the box next to it: **Enabled**
22. Click **Apply**, then **OK**.
23. Back in the main *Inbound Rules* window, you should see that same rule line now says **Yes** under the *Enabled* header.
24. You've enabled ICMP ping! Let's go test it.
25. Switch back to your client1 machine, and run the same ping you ran earlier pointing to srv1. (Refer to *Part 2, Step 4*.)
26. Does it work?
27. It if does, congratulations! You've just enabled ping for connectivity checking to *srv1* and gone through your first foray into the Windows Firewall.

> ℹ **A note for later labs:** By default, this rule is set to only allow incoming pings (ICMP requests) from computers on your **VM network**—that is, other virtual machines on the same VMware NAT or host-only network as your server—not from the wider Internet or any physical computers in the classroom.
>
> You can see this by checking the **Scope** tab in the rule's properties, where "Remote IP address" is set to "Local subnet."
>
> We'll spend more time on how **Scope** and **Private/Public profiles** affect your firewall rules in our *Secondary Network* assignment.

> 📝 **Lab Question:** Why did we *not* have to do this for our IIS web server setup?
>
> In the **Inbound Rules** list, scroll through to see if you can find the rule that's allowing web server pages to be requested from *srv1*.
>
> Write down the name of the rule in your Lab Logbook when you find it (it's not called IIS) and explain why you think you didn't have to enable this rule yourself. Think back to when you installed the IIS Server Role.

Part 4: Windows Server Core (srv2) - Applying Firewall Rules

Let's apply the same incoming ping firewall rule to our Server Core machine so we can check it's network connectivity as well.

1. Login to your Server Core (srv2) machine.
2. The *sfconfig* text-based application automatically launches.
3. Select option 8 to find this machine's IP address. Write it down.
4. Back in your Windows 11 Client (client1), try to ping this address. Does it work?

5. Just as in srv1, it doesn't.
6. Go back to srv2.
7. If you're still in the *Network settings* page, leave the field blank and hit **Enter** to go back to the main screen.
8. Select option 15 to exit to PowerShell.
9. As there's no GUI, we need to use PowerShell for firewall management.
10. Let's take a look at the existing rules, just like in Part 2. Run the following PowerShell command:

```
Get-NetFirewallRule | Where-Object DisplayName -Like '*ICMPv4-In*'
```

11. Several results appear in the search. Look for the one with the name: **Core Networking Diagnostics - ICMP Echo Request (ICMPv4-In)**
12. This firewall rule object, starting from *Name* all the way down to *PackageFamilyName* lists all the configured properties for this rule.
13. Read through these properties. Recognize any from our GUI version?
14. This rule matches both our incoming ping requests and keeps to *Private, Public* profiles. As with our GUI version, the scope is defaulted to local subnet only.
15. Let's turn on this incoming firewall rule by running the following command:

```
Enable-NetFirewallRule -Name CoreNet-Diag-ICMP4-EchoRequest-In
```

16. This selects the right rule and enables it.
17. One of the things you **must** get into the habit of doing with CLI commands is ***double-checking your work***.
18. Let's do that now by asking the system if it was actually enabled:

```
Get-NetFirewallRule -Name CoreNet-Diag-ICMP4-EchoRequest-In
```

19. Look for the *Enabled:* field. See how it's changed to **True**? It worked!
20. Optional: We can even check the scope for local subnet only as we did in the GUI version if we want with the following command:

```
Get-NetFirewallRule -Name CoreNet-Diag-ICMP4-EchoRequest-In | Get-
NetFirewallAddressFilter
```

Output:

```
LocalAddress  : Any
RemoteAddress : LocalSubnet4
```

21. Last, because there are two incoming ICMP rules (*Domain* profile and *Private, Public* profile), let's check that only the *Private, Public* rule is enabled:

Command:

```
Get-NetFirewallRule -DisplayName "Core Networking Diagnostics -
ICMP Echo Request (ICMPv4-In)" | Select-Object DisplayName,
Profile, Enabled
```

22. This shows a brief status of both rules. The *Domain* version should show **False** under the *Enabled* header, while the *Private, Public* version should show **True**.

Output:

```
DisplayName
Profile Enabled
-----------
------- -------
```

23. Now, let's test our ping. Switch over to your Windows 11 Client (client1).
24. Open a Command Prompt window, and run the following command: `ping srv2-ipaddress` (Refer to Part 3, Step 3).
25. Does it work?
26. It if does, congratulations! You've just enabled ping for connectivity checking to *srv2* and gone through your first foray into the PowerShell!

Part 5: Windows Client (client1) - Applying Firewall Rules

Finally, let's enable ping on our Windows Client machine.

1. Login to *client1*.
2. Click the **Start** button and type the following search: **firewall**
3. Of the options that appear, select: **Windows Defender Firewall with Advanced Security**
4. Just as in *Part 2* with *srv1*, navigate to **Inbound Rules**.
5. Click the *Name* header to sort by name.
6. Find the following rule:
7. Name: **Core Networking Diagnostics - ICMP Echo Request (ICMPv4-In)**
8. Profile: **Private, Public**
9. Double-click it to open the rule's configuration settings.
10. In the *General* tab, find and check the box next to **Enabled**.
11. Click **Apply**, then **OK**.
12. Open a *Command Prompt* window and run the following to get your client's IP address: `ipconfig`
13. Switch to your *Server Core (srv2)* machine.
14. In PowerShell, run the following command: `ping client1-ipaddress` (Where client1-ipaddress is the address from Step 10.)

15. Example: `ping 192.168.1.15`
16. Does it work?
17. It if does, congratulations! You've just enabled ping for connectivity checking to *client1* and have full connectivity checking for your entire environment! This will become ***very*** handy in Labs 3-4.

Investigation 2: Remote Management - Windows Server GUI (srv1)

While you have direct access to all three VMs from VMware, these days most machines are remote. This means you typically do not have direct, physical access to the servers.

Even if you do, having remote access to allow you to manage your servers from the comfort of your office (or home!) instead of walking to each physical machine is far better.

In this investigation, we'll set up remote access to our Windows Server GUI (srv1) so we can connect to it using your Windows 11 Client VM.

Part 1: Enabling Remote Desktop Connections

In this part, we'll turn on Remote Desktop (RDP) on the server. This will allow you to remotely connect to *srv1* from your Windows 11 Client.

1. In the *Server Manager* application, navigate to *Local Server*.
2. In the main *Properties* window, find the line entry for: **Remote Desktop**
3. Click the link next to it: **Disabled**
4. In the new *System Properties* window, you should automatically be on the

Remote tab.

5. Look down to the section on **Remote Desktop**.
6. Toggle the option for: **Allow remote connections to this computer**
7. Ensure the checkbox next to this is on: **Allow connections only from computers running Remote Desktop with Network Level Authentication (recommended)**
8. Click **Apply**, then **OK**.

Part 2: Adding your RDP Firewall Rules

In this part, we'll add a firewall rule to allow the connection over the local network.

1. On *srv1*, open **Windows Defender Firewall with Advanced Security**.
2. In the *Inbound Rules* section, look for the following rules:
3. Remote Desktop - Shadow (TCP-In)
4. Remote Desktop - User Mode (TCP-In)
5. Remote Desktop - User Mode (UDP-In)
6. If they're all enabled (they should be), you're good to go!
7. Why check? Always check your firewall rules when enabling a new service, even if it *usually* enables the associated firewall rule automatically. **Never assume!**
8. If these rules are **not** enabled, enable them now.

Part 3: Connecting to Windows Server (srv1) from Windows 11

In this part, we'll verify our work by connecting to the server using our Windows 11 client VM.

1. Make sure you've got your *srv1* IP address from earlier. (You wrote it down,

right?)

2. On *client1*, click the **Start** menu.
3. Search for and open: **Remote Desktop Connection**
4. In the newly opened application, in the *Computer* field, enter the IP address for *srv1*.
5. Click **Connect**.
6. Enter your *srv1* username and password, then click **OK**.
7. A security dialogue box pops up. **Check the box next to "*Don't ask me again for connections to this computer*" before the next step.**
8. Click **Yes**.
9. If you can see your *srv1* desktop, congratulations! You now have remote access.
10. To quit the remote session, find the floating HUD at the top of the screen with the IP address and click on the **X** icon.
11. Note: **This will not shutdown your *srv1* VM. It only ends your remote session.**
12. Note 2: A remote session will automatically lock your VM's direct session. You will need to unlock it again when you go back to it through VMware Workstation directly.

Investigation 3: Remote Management – Windows Server Core (srv2) with Remote Desktop

In this investigation, we'll set up remote access to our Windows Server Core (srv2) so we can connect to it using your Windows 11 Client VM.

Part 1: Enabling Remote Desktop Connections

In this part, we'll turn on Remote Desktop (RDP) on the Core server (*srv2*).

1. Login to *srv2*.
2. If you're at the PowerShell command prompt, run the following command to get back into the text-based *Server Manager* application: `sconfig`
3. Select option 7.
4. In the *Remote desktop* page, select option **E** to enable Remote Desktop.
5. It will ask you to select a security level. Select **Option 1**.
6. Press **Enter** to complete and go back to the main *Server Manager* page.
7. **Double-check your work.** Select option 7 again.
8. Is the *Remote desktop status* changed and correct?
9. If not, go through Steps 4-9 again.
10. If yes, press **Enter** without entering any options to go back to the main screen without making any changes.

Part 2: Adding a RDP Firewall Rules

In this part, we'll check that the RDP firewall rules have been enabled.

1. Select **Option 15** to exit to PowerShell.
2. Run the following PowerShell command to check the status of your RDP firewall rules:

Command:

```
Get-NetFirewallRule | Where-Object { $_.DisplayName -like '*Remote
Desktop*' -and $_.Direction -eq 'Inbound' } |
    Select-Object Name, DisplayName, Profile, Enabled
```

Output:

```
Name                          DisplayName
Profile Enabled
----                          -----------
------- -------
RemoteDesktop-In-TCP-WS       Remote Desktop - (TCP-WS-
In)            Any   False
RemoteDesktop-In-TCP-WSS      Remote Desktop - (TCP-WSS-
In)           Any   False
RemoteDesktop-Shadow-In-TCP   Remote Desktop - Shadow (TCP-
In)        Any    True
RemoteDesktop-UserMode-In-TCP Remote Desktop - User Mode (TCP-
In)     Any    True
RemoteDesktop-UserMode-In-UDP Remote Desktop - User Mode (UDP-
In)     Any    True
```

3. Check that your output matches the above. This is the same as on *srv1*, just at the command line. Notice with are enabled and which are not?
4. If you need to, grab the IP address for *srv2* here by running this command in PowerShell: `ipconfig`

Part 3: Connecting to Windows Server (srv2) from Windows 11 via RDP

In this part, we'll verify our work by connecting to *srv2* using our Windows 11 client VM. This is essentially the same as our instructions for *srv1*, but with a different IP address.

1. Make sure you've got your *srv2* IP address from earlier. (You wrote it down, *right*?)
2. On *client1*, click the **Start** menu.
3. Search for and open: **Remote Desktop Connection**

4. In the newly opened application, in the *Computer* field, enter the IP address for *srv1*.
5. Click **Connect**.
6. Enter your *srv2* username and password, then click **OK**.
7. A security dialogue box pops up. **Check the box next to "*Don't ask me again for connections to this computer*" before the next step.**
8. Click **Yes**.
9. If you can see your *srv2* Command Prompt window, congratulations! You now have remote access.
10. To quit the remote session, find the floating HUD at the top of the screen with the IP address and click on the **X** icon.
11. Note: **This will not shutdown your *srv2* VM. It only ends your remote session.**
12. Note 2: A remote session will automatically lock your VM's direct session. You will need to unlock it again when you go back to it through VMware Workstation directly.

Part 4: File system navigation

(A primer on file system navigation in Windows. Have them create a few directories and empty files using CLI only.)

Part 5: Editing a file

(This is where we do the single example of using PowerShell to modify a text file.)

Investigation 4: Remote Management – Windows Server

Core (srv2) with SSH

Part 1: Enabling SSH Connections

In this part, we'll turn on incoming SSH connections so we can connect to our server using Visual Studio Code from the Windows 11 client.

1. Login to your Server Core *(srv2)*.
2. Exit the *Server Manager* program by selecting **Option 15**.
3. In PowerShell, run the following command to install the OpenSSH Server

```
Add-WindowsCapability -Online -Name OpenSSH.Server~~~~0.0.1.0
```

4. After it installs, we have to start the SSH service. Run the following:

```
Start-Service sshd
```

5. We now have to set the SSH service to start automatically every time the system boots. Run the following:

```
Set-Service -Name sshd -StartupType 'Automatic'
```

```
# Install OpenSSH Server
Add-WindowsCapability -Online -Name OpenSSH.Server~~~~0.0.1.0

# Start and enable the service
Start-Service sshd
Set-Service -Name sshd -StartupType 'Automatic'
```

Part 2: Adding an SSH Firewall Rule

1. We'll now add an SSH rule. One already exists, but it's only attached to the *Private* profile and would not work for us.
2. Instead, we'll create a custom rule. Run the following command:

```
New-NetFirewallRule -Name sshd -DisplayName 'OpenSSH Server (sshd)'
-Enabled True -Direction Inbound -Protocol TCP -Action Allow
-LocalPort 22 -Profile Private,Public
```

3. This rule sets the following:
4. Name: sshd
5. Display Name: OpenSSH Server (sshd)
6. Enabled: True
7. Profile: Private, Public
8. Direction: Inbound
9. Action: Allow
10. Protocol: TCP
11. Port: 22
12. Verify your work! First, let's look at the firewall object by running the following command:
13. Now, let's verify the protocol and port with the following command:

Command:

```
Get-NetFirewallRule -Name 'sshd'
```

Output:

```
Name                          : sshd
DisplayName                   : OpenSSH Server (sshd)
Description                   :
DisplayGroup                  :
Group                         :
Enabled                       : True
Profile                       : Private, Public
Platform                      : {}
Direction                     : Inbound
Action                        : Allow
EdgeTraversalPolicy           : Block
LooseSourceMapping            : False
LocalOnlyMapping              : False
Owner                         :
PrimaryStatus                 : OK
Status                        : The rule was parsed successfully
from the store. (65536)
EnforcementStatus             : NotApplicable
PolicyStoreSource             : PersistentStore
PolicyStoreSourceType         : Local
RemoteDynamicKeywordAddresses : {}
PolicyAppId                   :
PackageFamilyName             :
```

Command:

```
Get-NetFirewallRule -Name 'sshd' | Get-NetFirewallPortFilter |
Select-Object Name, Protocol, LocalPort
```

Output:

```
Name Protocol LocalPort
---- -------- ---------
     TCP      22
```

6. If your own output looks like the above, congratulations! If not, try your configuration commands again or ask for help.

Part 3: Connecting to Windows Server Core (srv 2) from Windows 11 via SSH

1. On your Windows 11 Client (*client1*), open Command Prompt.
2. Enter the following command: `ssh Administrator@srv1ipaddress` (Where *srv2ipaddress* is your actual *srv2* IP address.)
3. Example: `ssh Administrator@192.168.1.15`
4. The first time you connect, a prompt will ask you if you are sure. Type **yes** and hit **Enter**.
5. Type your password when asked and hit **Enter**.
6. Note: You will *not* see asterisks or other characters as you type. **This is normal.** It is taking your keyboard input. It may take you a few tries to get used to it.
7. Once logged in, you are now on *srv2*'s Command Prompt environment.
8. To get back into PowerShell (which you should), run: `powershell`
9. It is **not** recommended to run the *sconfig* program over an SSH connection. If you need to use the *sconfig / Server Manager* text-based program, connect through your RDP connection in Part 1.

> **ℹ A note about SSH and *Server Manager*:**
> Your SSH connection will allow you to run the *sconfig* Server Manager program and load it. However, many of it's functions fail over an SSH connection because it relies on background interactive processes that can't run on an SSH session. They must be run from the RDP connection, despite it *looking* the same.
>
> For an example of this, run `sconfig` from your SSH session and try to run Windows Updates. If it finds any and you try to install them, it will fail. This is

> an SSH+sconfig problem; run the same thing through RDP and it will work fine.

Investigation 5: Remote Management - Windows Server Core (srv2) with Visual Studio Code + SSH

Part 1 : Connecting to srv2 with Visual Studio Code

(insert instructions on how to use the UI to connect, including saving the connection for future use)

Part 2: A Quick Tour of VS Code + Remote SSH

(A brief tour of the three panes: File navigation, text editor, and PowerShell terminal)

Part 3: VS Code for Fun and Profit

(Brief exercises having them do some file management with the navigation pane, opening and modifying a file, and then running a PowerShell command)

(Insert caveat about most **sconfig** commands not working through SSH or VS Code, they must be done through RDP because of how interaction works.)

Lab 4 - Implementing DNS

Investigation 1: Configuring DNS on Windows Server (*srv1*)

Part 1: Setting a Static IP Address on *srv1*

As *srv1* will be our primary network services provider, we have to give it a **static IP address**. This is so it doesn't periodically change; if this were to occur, any other machines relying on this one would suddenly be unable to connect and lose access to all services provided, including the very DNS we're about to set up!

Note: You will be changing network information. It is best to do *Part 1* directly on the machine, not through a remote connection.

To set this server with an static IP address:

1. Open Command Prompt and run the following: `ipconfig /all`
2. Write down the current address information *srv1* has:
3. IPv4 address: 10.88.111.14
4. Subnet Mask: 255.255.255.0
5. Default Gateway: 10.88.111.254
6. DNS Servers: 66.207.192.6, 206.223.173.7
7. In *Server Manager > Local Server*, in the *Properties* area of the page, find the line entry: **Ethernet0**
8. Click the link next to it: **IPv4 address assigned by DHCP, IPv6 enabled**

9. This will open the *Network Connections* window.
10. Right-click on *Ethernet0* and select **Properties** from the drop-down menu.
11. In the new *Ethernet0 Properties* window, click on the line item (**not the checkbox**) for **Internet Protocol Version 4 (TCP/IPv4)**.
12. Further down on the same window, find and click on the **Properties** button.
13. In the new pop-up window *Internet Protocol Version 4 (TCP/IPv4) Properties*, in the *General* tab, select the **Use the following IP address:** option.
14. Enter the following information in the fields below from your previously-written *srv1* address information:
15. IP address
16. Subnet mask
17. Default gateway
18. Select the **Use the following DNS server addresses:** option.
19. Enter the following information in the fields below from your previously-written *srv1* address information:
20. Preferred DNS server: **127.0.0.1** (< This is important!)
21. Alternate DNS server: Choose one from your written records, it doesn't matter which.
22. Click **OK**.
23. Back in the *Ethernet0 Properties* window, click **Close**.
24. Open *Firefox* and go to **eff.org** to confirm you haven't lost Internet connectivity. If you can't access the page, go back through these steps or ask for help. Do not continue forward until this works!
25. Close the *Network Connections* window.
26. Verify your change in *Server Manager > Local Server*. The earlier line entry should now simply say your IP address instead of "assigned by DHCP".

Part 2: Installing the DNS Server Role on *srv1*

1. **Login to srv1** with your Administrator account.
2. Open the **Server Manager** application (icon is on the taskbar or Start menu).
3. In the left menu, click on **Local Server** if you're not already there.
4. At the top right, click **Manage > Add Roles and Features**.
5. In the **Add Roles and Features Wizard** window:
    i. On *Installation Type*, select **Role-based or feature-based installation**, then **Next**.
    ii. On *Server Selection*, confirm **Select a server from the server pool** and choose **srv1**, then **Next**.
    iii. On *Server Roles*, scroll down and check the box for **DNS Server**.
    iv. A pop-up will ask to add required features, click **Add Features**.
    v. Back on the *Server Roles* page, click **Next**.
    vi. On the *Features* page, leave defaults, click **Next**.
    vii. On the *DNS Server* page, read and click **Next**.
    viii. On the *Confirmation* page, click **Install**.
    ix. Wait for installation to finish (may take a few minutes).
    x. When you see *Installation succeeded on srv1*, click **Close**.

Part 3: Configuring a New DNS Zone

1. Back in **Server Manager**, in the top right, click **Tools > DNS** to open the DNS Manager.
2. In the left pane, expand **srv1** and then **Forward Lookup Zones**.
3. Right-click on **Forward Lookup Zones** and select **New Zone…**
4. In the **New Zone Wizard**:

i. On *Zone Type*, select **Primary zone**, click **Next**.
ii. On *Zone Name*, enter ***yourSenecaUsername*.com**, replacing *yourSenecaUsername* with your actual username.
iii. For example, mine might be: **cjohnson30.com**
iv. Click **Next**.
v. On *Zone File*, keep **Create a new file** selected, click **Next**.
vi. On *Dynamic Update*, keep **Do not allow dynamic updates** (default), then click **Next**.
vii. Click **Finish** to create the zone.

Part 4: Adding Host (A) Records

1. In **DNS Manager**, expand your new **yourSenecaUsername.com** zone.
2. Right-click in the right pane and select **New Host (A or AAAA)…**
3. For each of your lab VMs, create a record:
4. **srv1**
   - Name: srv1
   - IP address: (Use the IPv4 address of srv1 from ipconfig)
   - Click **Add Host**
5. **srv2**
   - Name: srv2
   - IP address: (Use the IPv4 address of srv2 from ipconfig)
   - Click **Add Host**
6. **client1**
   - Name: client1
   - IP address: (Use the IPv4 address of client1 from ipconfig)
   - Click **Add Host**
7. When finished, you should see all three names listed in the zone.

Part 5: Configure DNS Forwarders on *srv1*

We want *srv1* to handle all DNS queries for our other machines, including locations on the Internet. To do this, we have to tell *this* server where to look it if doesn't have the answer locally. If it can't answer locally, it will forward the request to an Internet-based DNS server.

1. In **DNS Manager** on srv1, right-click on your server’s name in the left pane.
2. Select **Properties**.
3. Go to the **Forwarders** tab.
4. Click **Edit…**
5. In the *IP address* field, enter the IP address from your earlier list, then hit **Enter** for it to verified. If it can reach it, you should see a domain name populate next to the IP address.
6. Add a second DNS for fallback, using Google's DNS: **8.8.8.8**
7. Make sure Google's DNS is the **second** entry in the list.
8. Click **OK** to add.
9. Click **OK** again to close Properties.

Part 6: Testing DNS Resolution and Entries Locally

Our first task is to confirm this works locally by telling the server to talk to itself first for DNS resolution. This lets us test our DNS service and records while removing the network and firewall pieces.

1. Open **Command Prompt**.
2. To test DNS resolution functionality, run the following commands one at a time:

```
nslookup srv1.cjohnson30.com
nslookup srv2.cjohnson30.com
nslookup client1.cjohnson30.com
nslookup eff.org
```

7. If each returns the proper IP address value, you're done! If not, revisit your earlier configuration steps and check to see what might be missing.
8. Don't be afraid to ask for help!

Investigation 2: Configuring Machines For DNS Use

Part 1: Windows Server Core (srv2)

1. Log in as Administrator.
2. At the prompt, run:

```
Get-NetIPAddress
```

3. Look for the section that contains *srv2*'s normal IP address. Make a note of the **InterfaceIndex** number just below it. (For example, 4. Yours may differ.)
4. Set the DNS server to srv1’s IP (replace X.X.X.X with srv1’s actual address, and 4 with your NIC’s ifIndex):

```
Set-DnsClientServerAddress -InterfaceIndex 4 -ServerAddresses
192.168.1.14
```

5. Confirm the change:

```
Get-DnsClientServerAddress
```

6. To test DNS resolution functionality, run the following commands one at a time:

```
nslookup srv1.cjohnson30.com
nslookup srv2.cjohnson30.com
nslookup client1.cjohnson30.com
nslookup eff.org
```

7. If each returns the proper IP address value, you're done! If not, revisit your earlier configuration steps and check to see what might be missing.
8. Don't be afraid to ask for help!
9. Last, ping each other machine to prove you no longer need to remember IP addresses:

```
ping srv1.cjohnson30.com
ping client1.cjohnson30.com
```

13. Cool, right?

Part 3: Windows 11 Client (client1)

1. Login as Administrator.
2. Right-click the **Network** icon in the system tray, click **Network and Internet settings**.
3. Click **Ethernet**.
4. Scroll down to *DNS server assignment* and click the **Edit** button next to it.
5. In the *Edit DNS settings* pop-up window, click the drop-down menu to change it from *Automatic (DHCP)* to **Manual**.

6. Find the new *IPv4* option and toggle it **On**.
7. In the *Preferred DNS* field, enter the *srv1* IP address.
8. Leave all others on their defaults and click **Save**.
9. Open Command Prompt and type:

```
ipconfig /flushdns
```

9. To test DNS resolution functionality, run the following commands one at a time:

```
nslookup srv1.cjohnson30.com
nslookup srv2.cjohnson30.com
nslookup client1.cjohnson30.com
nslookup eff.org
```

10. If each returns the proper IP address value, you're done! If not, revisit your earlier configuration steps and check to see what might be missing.
11. Don't be afraid to ask for help!
12. Last, ping each server to prove you no longer need to remember IP addresses:

```
ping srv1.cjohnson30.com
ping srv2.cjohnson30.com
```

13. Cool, right?

Investigation 3: Configuring Reverse Lookup (PTR) Records

Part 1: Create a Reverse Lookup Zone

1. In **DNS Manager** on *srv1*, right-click **Reverse Lookup Zones** in the left pane.
2. Click **New Zone…**
3. In the **New Zone Wizard**:
    i. On *Zone Type*, select **Primary zone**. Click **Next**.
    ii. On *Reverse Lookup Zone Name*, choose **IPv4 Reverse Lookup Zone**. Click **Next**.
    iii. On *Network ID*, enter the first three octets of your VMs' subnet (for example, if your VMs' IPs are 192.168.131.x, enter 192.168.131). Click **Next**.
    iv. On *Zone File*, keep defaults. Click **Next**.
    v. On *Dynamic Update*, keep defaults (Do not allow dynamic updates). Click **Next**.
    vi. Click **Finish** to create the zone.

Part 2: Add PTR Records for Each VM

1. We're going to automatically add PTR records using our existing A records for ease of use.
2. Inside the *Forward Lookup Zones*, in the **cjohnson30.com** zone, right-click each A record you created (srv1, srv2, client1) and select **Properties**.
3. Check the box next to **"Update associated pointer (PTR) record"** and click **OK**.

4. If the box was already checked, uncheck it, click **Apply**, check it again, then click **Apply** again. This forces the PTR generation.
5. This should automatically create PTR records in your new reverse zone.
6. Double-check your work. Click on the folder inside *Reverse Lookup Zone* to see its contents. If empty, right-click on the folder and click *Refresh*.
7. They should now appear. If they do not, retrace your steps of ask for help.

Part 3: Test Reverse DNS Resolution

1. On *srv2*, open PowerShell.
2. Run **nslookup** on each VM's IP address:

```
nslookup 192.168.1.14
nslookup 192.168.1.15
nslookup 192.168.1.16
```

3. You should see the PTR record and corresponding hostname for each IP address.

Output

```
Server:  srv1.cjohnson30.com
Address:  10.88.111.14

Name:    client1.cjohnson30.com
Address:  10.88.111.15
```

4. Notice that the server entry no longer says "UnKnown" as it did before. Now it can identify itself!

Congratulations, you've now set up basic DNS services from *srv1* to each VM in your environment!

Lab 5 - Implementing DHCP

Investigation 1: Disabling VMware's NAT DHCP

When you first created your three VMs, you selected the built-in VMware NAT network. Along with providing Internet connectivity, it also provides a DHCP server to automatically assign each virtual machine its network information.

We are going to be rolling our own DHCP server on *srv1*. This means we have to **disable** the DHCP server in VMware Workstation. Otherwise, we'll have two DHCP servers by the end of this lab, causing all kinds of conflicts.

Your *srv2* and *client1* VMs will temporarily lose IP information until we complete this lab, but since we set *srv1* to static network configuration in Lab 3, it won't be affected!

Let's disable the VMware DHCP server.

Part 1: Disable VMware DHCP

1. Open **Virtual Network Editor** in VMware Workstation (needs to be run as Administrator).
2. Find your NAT network (usually VMnet8).
3. Uncheck or clear the DHCP option for this network.
4. Save/apply your changes.
5. You may need to restart your VMs or even the VMware NAT service for the

change to take effect.

Investigation 2: Installing the DHCP Server Role on *srv1*

Part 1: Add the DHCP Server Role

1. **Login to srv1** with your Administrator account.
2. Open **Server Manager**.
3. In the left menu, click on **Local Server** if you're not already there.
4. At the top right, click **Manage > Add Roles and Features**.
5. In the **Add Roles and Features Wizard**:
    i. On *Installation Type*, choose **Role-based or feature-based installation**. Click **Next**.
    ii. On *Server Selection*, confirm **Select a server from the server pool** and choose **srv1**. Click **Next**.
    iii. On *Server Roles*, scroll down and check **DHCP Server**.
    iv. If prompted, click **Add Features**.
    v. Click **Next** through *Features* and *DHCP Server* pages.
    vi. On *Confirmation*, click **Install**.
    vii. Wait for installation to finish (may take a few minutes).
    viii. When you see *Installation succeeded on srv1*, click **Close**.

Part 2: Post-Install Configuration

1. In **Server Manager**, at the top, a yellow exclamation (!) triangle will appear. Click it, then click **Complete DHCP configuration**.
2. In the **DHCP Post-Install Configuration Wizard**:

i. Click **Commit**.
ii. Click **Close**.

Investigation 3: Configuring DHCP on srv1

Part 1: Create a New IPv4 Scope

1. In **Server Manager**, at the top right, click **Tools > DHCP**.
2. In **DHCP**, expand your server's name (srv1) under **IPv4**.
3. Right-click **IPv4** and choose **New Scope…**
4. In the **New Scope Wizard**:
   i. Click **Next**.
   ii. For *Scope Name*, enter: **OSM620 Lab Scope**
   iii. Click **Next**.
   iv. For *IP Address Range*:
   v. **Start IP address:** 192.168.245.100
   vi. **End IP address:** 192.168.245.199
   vii. **Subnet mask:** 255.255.255.0
   viii. Click **Next**.
   ix. For *Add Exclusions*, leave blank. Click **Next**.
   x. For *Lease Duration*, leave default (8 days). Click **Next**.
   xi. For *Configure DHCP Options*, select **Yes, I want to configure these options now**. Click **Next**.

Part 2: Configure DHCP Options

1. **Router (Default Gateway):**

2. Enter your gateway IP from Lab 3, click **Add**, then **Next**.
3. You can also check it now by opening Command Prompt and running: `ipconfig`
4. **Domain Name and DNS Servers:**
5. For *Parent domain*, enter your DNS domain (e.g., cjohnson30.com).
6. For *DNS Servers*, your earlier DNS IP addresses should already be present.
7. Click **Next**.
8. **WINS Servers:**
9. Leave blank. Click **Next**.
10. **Activate Scope:**
11. Select **Yes, I want to activate this scope now**. Click **Next**.
12. Click **Finish**.

Part 3: Confirm DHCP Scope and Settings

1. In the DHCP console, expand **IPv4 > OSM620 Lab Scope > Address Pool**.
2. Make sure the address range is correct (should be .100 to .199).
3. Under **Address Leases**, you won't see anything yet—leases will appear as clients connect.

Part 4: Setting DHCP Reservations

Any servers on our network should have unchanging IP addresses so they can always be found. We did this manually with *srv1*, but we can use **Reservations** to give our other machines specific IP addresses.

When one of our VMs first boots up, it will go through the four-way DHCP handshake with the DHCP server to get networking information. By default, the DHCP server on *srv1* will assign a randomly chosen, currently-unused IP

address from our address pool (.100 to .199).

An IP reservation will cause the DHCP server to, instead, *always* give a specific machine the exact IP address we tell it to, and never give it out to others.

Normally, you'd only do this for other servers on your network and let basic clients receive a random address, but we'll reserve addresses for both *srv2* and *client1* for proof-of-concept.

1. On *srv2*, run `ipconfig /all` and write down the value on the *Physical Address* line. We'll need this shortly.
2. Do the same thing on *client1*.
3. Back on *srv1*, right-click on the *Reservations* folder and select **New Reservation**.
4. For *srv2*:
5. Reservation name: **srv2-senecaUsername** (Ex: srv2-cjohnson30)
6. IP address: **192.168.245.102**
7. MAC address: **00-0C-29-7F-45-47**
8. Description: **Windows Server 2025 Datacenter Core**
9. Supported types: **Both**
10. Click **Add**.
11. Verify your new reservation is there by looking inside the *Reservations* folder.
12. For client1:
13. Reservation name: **client1-senecaUsername** (Ex: client1-cjohnson30)
14. IP address: **192.168.245.103**
15. MAC address: **00-0C-29-88-7E-73**
16. Description: **Windows 11 Education Client**
17. Supported types: **Both**
18. Click **Add**.

19. Verify your new reservation is there by looking inside the *Reservations* folder.

Investigation 4: Switching Clients to Use the New DHCP Server for DNS

Your *srv2* (Server Core) and *client1* (Windows 11) VMs are already set to receive their IP address via DHCP, but you previously set their DNS settings to point directly at *srv1* during Lab 3. Now that *srv1* is a fully working DHCP server, you'll need to reset both VMs' DNS settings to automatic so they get both their IP and DNS server from *srv1*.

Part 1: Set srv2 (Server Core) DNS to DHCP

1. Log in as Administrator.
2. At the prompt, run:

```
Get-NetIPInterface
```

3. Note the **InterfaceIndex** for your network adapter. (Likely 4, but check!)
4. Set the DNS server to automatic (DHCP):

```
Set-DnsClientServerAddress -InterfaceIndex 4 -ResetServerAddresses
```

5. To apply changes, renew your DHCP lease:

```
ipconfig /release
```

6. Verify that the new DNS server is set by running:

```
Get-DnsClientServerAddress
```

7. You should see the *srv1* IP address listed as the DNS server.

Part 2: Set *client1* (Windows 11) DNS to DHCP

1. Login as Administrator.
2. Right-click the **Network** icon in the system tray, click **Network and Internet settings**.
3. Click **Ethernet**.
4. Scroll down to *DNS server assignment* and click the **Edit** button next to it.
5. In the *Edit DNS settings* pop-up window, click the drop-down menu to change it from *Manual* to **Automatic (DHCP)**.
6. Leave all others on their defaults and click **Save**.
7. Open Command Prompt and run:

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

8. Verify your IP settings, including DNS:

```
ipconfig /all
```

9. You should see your new IP address along with the same DNS address as before, but all supplied through DHCP.

Investigation 5: Testing DHCP Functionality

Part 1: Check Leases in DHCP Manager

1. On *srv1*, open **DHCP Manager**.
2. Expand **IPv4 > OSM620 Lab Scope > Address Leases**.
3. Confirm you see entries for *srv2* and *client1* with their assigned IPs.

Part 2: Test DHCP & DNS Functionality

1. On both *srv2* and *client1*, run:

```
ipconfig
```

2. Confirm both the **IPv4 address** and **DNS server** match your DHCP scope and *srv1*'s IP address.
3. Try resolving and pinging your servers:

```
nslookup srv1.cjohnson30.com
nslookup eff.org
ping srv1.cjohnson30.com
```

4. If everything works, you're done! (If it doesn't, double-check your steps or ask for help.)

Assignment 1

Assignment 2