Lab 3: Installing an AD domain

Server System Management - Windows Server Labs

<STUDENT NAME>

Academic Year: 2021-2022

Table of Contents

[Introduction 3](#_Toc96291360)

[Learning Goals 4](#_Toc96291361)

[Knowledge (what you need to know) 4](#_Toc96291362)

[Abilities (what you need to be able to do) 4](#_Toc96291363)

[Requirements 4](#_Toc96291364)

[Recap Cmdlets & Preparing the infra 5](#_Toc96291365)

[Installing the domain 7](#_Toc96291366)

[Windows GUI Server 7](#_Toc96291367)

[Windows CORE Server 8](#_Toc96291368)

[Windows 11 Client 9](#_Toc96291369)

[Windows Admin Center 10](#_Toc96291370)

[Pester tests 11](#_Toc96291371)

[Recap Script 12](#_Toc96291372)

[Extensions - Optional Assignments 12](#_Toc96291373)

## Introduction

Up until now the virtual machines exist in a **Windows Workgroup** model. This lab fundamentally changes this model to a **Windows (Active Directory) Domain**. The GUI-server will act as a **domain controller** (from now on we will call this a **DC**). The core server and the Windows 11 client will become part of this domain. The core server is therefore at the end of the lab a **member server**.

Once the domain has been setup we will also take a look at the different possibilities to manage servers remotely with the sole exception of the **remote desktop protocol (RDP)** as this will get its own lab later on.

## Learning Goals

# Knowledge (what you need to know)

* What do you need to install to create an AD domain?
* What tools are typically used for (remote) server administration? What are RSAT-tools? What is Windows Admin Center?
* What is the difference between a local account and a domain account?

# Abilities (what you need to be able to do)

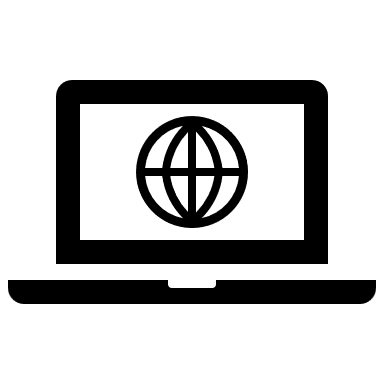
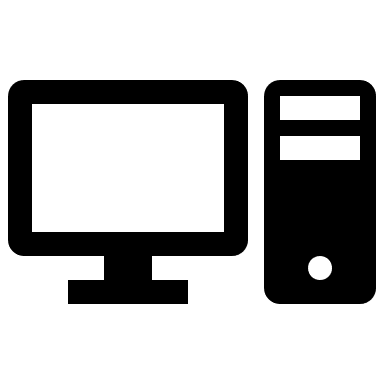
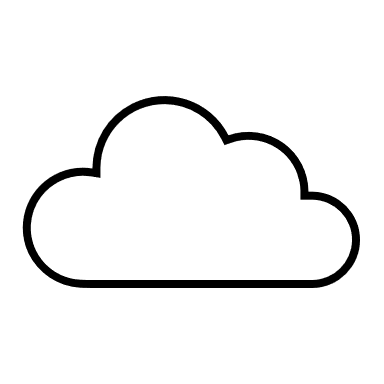
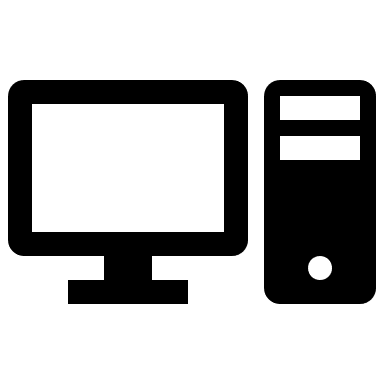
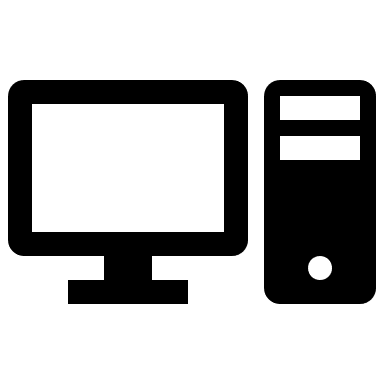
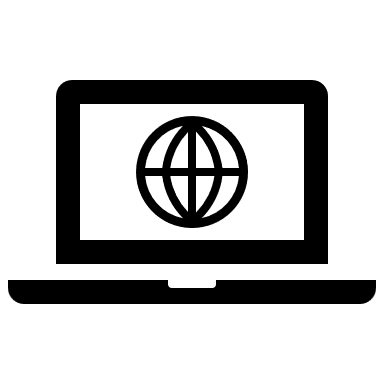
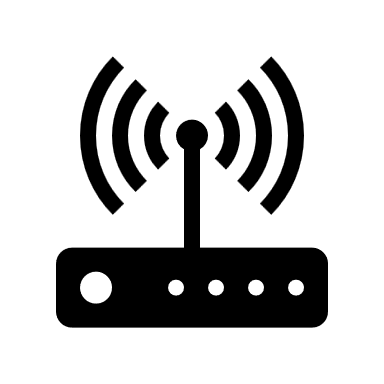
* Create an AD domain
* Install a domain controller
* Add servers & clients to a domain
* Perform remote administration with
  + RSAT tools
  + Windows Admin Center
  + invoke-command
  + enter-pssession

## Requirements

* Three virtual machines, installed and configured as described in Lab 1
  + Windows Server with GUI
  + Windows Server Core
  + Windows Client

## Recap Cmdlets & Preparing the infra

After finishing this first part, your Windows network should look like Figure 1. Take note of the architecture.



Internet

VMware NAT

+ gateway

(+ DNS)

**192.168.x.2/24**

Host laptop

**192.168.x.1/24**

Win11 Client

**192.168.x.y/24**

GUI Server

**192.168.x.100/24**

Core Server

**192.168.x.101/24**

VMware DHCP server

**192.168.x.254/24**

VMnet8 virtual switch

**VMware virtual**

**infrastructure**

**Virtual Machines**

Figure 1 VMware (virtual) network environment for the Windows labs

1. Log in to all your virtual machines and open a PowerShell window with elevated privileges (run as Administrator). On your GUI-server try to find a **legacy** command (and write it down below) that shows the name of the machine. To get the same information in PowerShell a possible solution is

Hostname.exe

[System.Net.Dns]::GetHostName()

1. Search a verb-noun cmdlet that shows all environment variables (Dutch: “omgevingsvariabelen”). Tip: The cmdlet itself has nothing to do with environment variables.

get-item Env:

1. Make use of *Get-Alias -definition <cmdlet>*  with the cmdlet of the previous question to find out its different aliases.

gi -> Get-Item

1. Use the environment variables as an alternative way to retrieve the name of the computer. Try this once with the cmdlet used in question 2 and once with the cmdlet: Get-Content. Do you notice the difference?

gi Env:COMPUTERNAME

get-content Env:COMPUTERNAME

1. To make it easier to identify the different machines over the network, we want you to change the computer name of your virtual machines. Take a look and search how you can do this in the GUI of Windows 11 but use a PowerShell cmdlet instead for all 3 machines (add an optional parameter to the cmdlet to automatically reboot your machines). Afterwards verify the names with PowerShell or in the GUI.

Graphical user interface, application

Description automatically generated

* 1. Name of the GUI-server: **GUI-<firstname>** where you replace <firstname> with your own name, for example: **GUI-THOMAS.** Make sure the total length of the computer name remains below 15 characters!

Rename-Computer -NewName "GUI-BEN" -Restart

* 1. Name of the CORE-server: **CORE-<firstname>**, for example: **CORE-THOMAS.** Make sure the total length of the computer name remains below 15 characters!

Rename-Computer -NewName "CORE-BEN" -Restart

* 1. Name of the CLIENT: **WIN11-<firstname>**, for example: **WIN11-THOMAS.** Make sure the total length of the computer name remains below 15 characters!

Rename-Computer -NewName "WIN11-BEN" -Restart

1. Use a **legacy** Windows command to find the IPv4 address of the GUI-server. Write down the command and the IPv4 address.

Get-NetIPAddress -AddressFamily IPv4 (don’t know if legacy)

1. Search for a PowerShell cmdlet to receive all IP information about the NIC with name “ethernet0”. Tip: the name here is called “interfacealias”.

Get-NetIPConfiguration -interfacealias ethernet0

1. Are these IP addresses assigned statically or dynamically?

dynamically

1. Try to use piping to combine cmdlets to limit the output to only show IPv4 addresses

Get-NetIPConfiguration | Get-NetIPAddress -addressfamily ipv4

1. Usually **servers** on a private network (=”on prem”) have a fixed (static) IP address that does not change over time. In our lab environment we configured our virtual machines to be on the NAT-network of VMware Workstation. Virtual machines connected to this virtual switch typically receive an IP address from VMware of the type 192.168.x.y where x is different for everyone but the same for all your virtual machines. If you want to change this x manually you can go in VMware to “edit > Virtual Network Editor > change settings > NAT > subnet IP”. You are free to change this (for example to 32 for NAT) if you don’t change this note the x on your machine and keep it in mind. Write down your “x” and your Subnet IP and Subnet mask.

0 & 255.255.255.0

1. Use **a PS cmdlet** to change the IPv4 configuration of your GUI-server to a static IP adress. Verify in the GUI afterwards.
   1. IPv4 address – change the last digit to 100 à new IPv4 address = 192.168.x.100 (see previous question for x)

New-NetIPAddress -InterfaceIndex 3 -IPAddress 192.168.0.100 -PrefixLength 24 -DefaultGateway 192.168.0.2

* 1. Subnet Mask – (see previous question)

Ipconfig

* 1. Default Gateway – This is a virtual device from VMware Workstation: 192.168.x.2 (see previous question for x).

ipconfig

1. The previous cmdlet didn’t allow us to change the DNS configuration. Use the *Set-DNSClientServerAddress* cmdlet and use 192.168.x.2 as the IP address for the DNS-server. Verify in the GUI with **ncpa.cpl**.

Set-DnsClientServerAddress -InterfaceIndex 3 -ServerAddresses 192.168.0.2

Graphical user interface, text, application

Description automatically generated

1. Use the **sconfig** tool to change the IP configuration on your **CORE-server**. Use the same settings as your GUI-server but add 1 to the IP address: **192.168.x.101**. Subnet Mask, default gateway and DNS settings are the same. Verify with a ping to the 192.168.x.2, followed by a ping to google.com (or 172.21.1.111 if you are on campus at Howest, they block pings to the internet).

Text

Description automatically generated

1. Try pinging from your CORE-server to your GUI-server. This should fail even though they are part of the same network. The *Windows Defender Firewall* by default blocks ICMP-messages (used by the ping command). Use the **GUI** to change 2 rules in the Windows Defender Firewall (on the GUI server). You should be able to ping with IPv4 and IPv6 packets (no matter the source computer). Verify by pinging from your CORE-server and your host to your GUI-server with *ping -4 192.168.x.100* and *ping -6 192.168.x.100*.

Graphical user interface, text

Description automatically generated

1. Perform the same changes on your **CORE-server** with PS-cmdlets. You will need to Enable 2 rules, one for IPv4 and one for IPv6. Write the cmdlets down below and verify by pinging.

netsh advfirewall firewall add rule name="ICMP Allow incoming V4 echo request" protocol=icmpv4:8,any dir=in action=allow

netsh advfirewall firewall add rule name="ICMP Allow incoming V6 echo request" protocol=icmpv6:8,any dir=in action=allow

1. Finally, use Windows Admin Center to modify the necessary firewall rules of the **Win11 client**, so that it also allows pinging (to your WIN11 client).

Calendar

Description automatically generated with low confidence

1. **Custom prompt**: From now we expect a custom PowerShell prompt on every screenshot. It should show the time & date, the domain, the username, the computername and the path. Make use of the following function to establish this custom prompt. Edit the $PROFILE with your favorite editor and reload your PowerShell window. On Leho (FAQ Lab 3) you can find a bigger example of a custom $PROFILE configuration.

function prompt {"`n$(get-date) | PS [$Env:userdomain\\$Env:username@$Env:computername] `n$($PWD.ProviderPath) > "}

1. On your CORE server perform the cmdlet: *Get-WindowsFeature* & *Get-WindowsCapability -Online*. What are the differences between the two?

Get-windowsfeature gives all the available and installed windows features whereas get-windowscapability shows available languages

1. How many Features & Capabilities are available on the 3 different machines. Tip: Measure-Object.

Core: feature=209, capability=314

GUI: feature=266, capability=316

Client: X

1. Try to create a one-liner (with piping) to show all installed Features.

Get-WindowsFeature | Where InstallState -Eq installed

## Installing the domain

### Windows GUI Server

1. Log in to your GUI server with the Administrator account. Use the *Server Manager* and install the role “*Active Directory Domain Services*”. Tip: use the menu and choose “*Manage > Add Roles and Features > Role-based installation*”
2. When the installation is finished, Windows will notify you that there are additional actions required. Click on the yellow (triangle) notification under the flag icon and choose “*Promote this server to a domain controller*”. Choose the third option (as it is the first domain controller we create) and enter as **root domain name**: **corp-<firstname>.serverlabs.be** (where you change <firstname> to your first name and delete the <>). All other default options are ok and we suggest to use “Friday13th!” once again as a password.
3. Your server will be restarted after installation. You need to sign in once again. Note that you are **not allowed to login with a local user on a DC**. Windows suggests to login with an **domain user account.** What is the name of the suggested domain user account? Verify this by using a known (but still very relevant) **legacy** command that starts with the letter ‘w’. A possible PowerShell variant is

[Security.Principal.WindowsIdentity]::GetCurrent().Name

whoami

1. Because you promoted your server to a DC additional tools have been installed on the system. Verify this with the *server manager* and write down at least 2 tools below.

Graphical user interface

Description automatically generated with low confidence

1. Open the **ADU&C = Active Directory Users and Computers** tool and verify whether your own server can be found in the “*Domain Controllers*” container. You should find the letters **GC.** What do they mean (tip: remember the lectures from Operating System Concepts)? What is the primary goal of the GC?

What is GC in DC type? A Global Catalog server is a domain controller that stores copies of all Active Directory objects in the forest. It stores a complete copy of all objects in the directory of your domain and a partial copy of all objects of all other forest domains.

1. Use **ADU&C** once more to find out what groups the built-in Administrator account is a member of. Write these groups down below.

Graphical user interface, table

Description automatically generated with medium confidence

1. Verify if your DC is able to browse the internet. This **may or may not** **fail**. Why? Because the DNS-server settings on the DC after installation point to the server itself (and no longer to the default gateway of VMware). Verify this by issuing the DNS settings of the server either using **ncpa.cpl, sconfig, cmd** or **PowerShell.** Write down below what IP-address(es) the server used as DNS-server.

127.0.0.1

1. In other words the DC uses a loopback address (and its own DNS server) to perform address (DNS) translations. To fix this issue (or verify everything is fine) we will configure our own DNS server to **forward** DNS questions to the default gateway of VMware: 192.168.x.2. Make use of the DNS management console, look/change “*Forwarders”* and verify you can browse the internet again **(Caution: it is possible you don’t have to do anything but verify!)**
2. In AD there can only be one Primary Domain Controller (PDC). One of the main functions of this PDC is the synchronization of time with other Domain Controllers and throughout the domain. The time of all computers are usually synced with the DC. Use the following command to check the time configuration on your DC:
   1. *w32tm /query /configuration*
   2. Take a look at the output and search in the **[TimeProviders]** section the type.
   3. There are usually 2 possible time provider types:
      1. **NTP**: Network Time Protocol – this protocol is used to synchronize with a remote time server on the internet.
      2. **NT5DS**: this provider uses the hardware clock of the VMware tools.
   4. If the DC uses **NT5DS** we suggest changing it to **NTP**. You can use the following commands:

***net stop w32time***

***w32tm /config /syncfromflags:manual /manualpeerlist:”0.pool.ntp.org, 1.pool.ntp.org, 2.pool.ntp.org”***

***w32tm /config /reliable:yes***

***net start w32time***

### Windows CORE Server

1. Next, we want to add our CORE server to the freshly installed domain. To make this possible we need to fix a DNS problem as our CORE server is not yet able to resolve our new domain name. Currently, the CORE server uses the VMware virtual DNS server, which does not know the domain. Fix this by changing the DNS server address on your CORE server to a suitable IP address. Verify by pinging “corp-<firstname>.serverlabs.be” from the CORE server, which should succeed.

Change preferred dns to 192.168.0.100

1. Use a PS cmdlet to add your server core machine to the AD domain (and reboot the core server).

Add-Computer -DomainName corp-ben.serverlabs.be -Restart

1. When the server core is rebooted, log in with the Administrator-account. Did you just log in with the local Administrator account or the domain Administrator account?

Local

1. Use a powershell cmdlet to install the windows feature “*RSAT-AD-PowerShell*”

Install-WindowsFeature -Name "RSAT-AD-PowerShell" -IncludeAllSubFeature

1. Use a **legacy** Windows-command to log off (there are multiple options).

logoff

1. Once a computer is part of a domain, local user accounts are typically rarely used to log in, and users/administrators prefer to use their domain account. Log in with the domain administrator account.
   1. Tip 1: Press 2x on **esc** on the log on screen to change users
   2. Tip 2: You can login to a domain with:
      1. **domainname\username**
      2. **username@domainname**
2. Once logged in verify with the sconfig tool that your CORE server is part of the domain. Afterwards verify on your GUI server with the ADU&C console that the core server is present in a specific container. What is the name of this container and finally put a screenshot of that container and the contents below.

Graphical user interface, text, application

Description automatically generated

1. If necessary, configure the custom powershell prompt for the domain administrator (see question 18)

### Windows 11 Client

1. Start your Windows 11 client machine if it wasn’t already running and log in with your **local** administrator/user account. Try to add this computer to the domain, using a PowerShell command. Before the machine can be successfully added to the domain however, you will notice that you have to fix something first (remember what we did for the CORE Server). Fix this issue and add the Windows 11 Client to the domain. Afterwards log in with your **domain** administrator account.

Set-DnsClientServerAddress -InterfaceIndex 3 -ServerAddresses 192.168.0.100

1. If necessary, configure the custom powershell prompt for the domain administrator (see question 18)
2. We will use this windows 11 client as our primary work machine. To connect with other machines (the DC for example) it is interesting to install specific software. This software stack is called the **RSAT-TOOLS** = Remote Server Administration Tools. You can install them with PowerShell but also through the GUI using the “*add an optional feature*” feature. Install the *DNS Server Tools* through the **GUI**.
3. Use a PS cmdlet to install the *server manager, group policy management tools, DHCP tools, Active Directory Domain Services and Lightweight Directory Services tools*.

Tip: Get-WindowsCapability -Name RSAT\* -Online

add-WindowsCapability -name "Rsat.ServerManager.Tools~~0.0.1.0" -online

add-WindowsCapability -name "Rsat.GroupPolicy.Management.Tools0.0.1.0" -online

add-WindowsCapability -name " Rsat.DHCP.Tools0.0.1.0" -online

add-WindowsCapability -name " Rsat.ActiveDirectory.DS-LDS.Tools~~0.0.1.0" -online

1. Reboot the windows 11 client to finish the installation of certain tools. Log in with your domain administrator account. Test out the new installed Server Manager from your windows 10 and add the GUI and CORE servers.
2. Another cool way to manage remote servers (and more options will follow later or in other courses like SSH, VNC, RPD, etc.) is to make use of PowerShell sessions. Through said sessions it is possible to make a connection from your Windows 11 to remote servers. You can start such a session from the Server Manager and right clicking on a server or by using the **Enter-PSSession <computername>.** Try this out and connect to the DC (option 1) and the CORE (option 2). Once connected try to retrieve the hostname of both machines. Paste a screenshot that shows the 2 hostnames and the prompt below. Use *exit* to stop the session.

Text

Description automatically generated

1. **Enter-PSSession** is a handy tool (try it!) to make a remote session. You could compare it a bit with SSH. There is however a PS cmdlet that allows us to *“invoke”* a command to a remote machine. Search this cmdlet and try it out. “Invoke” a command to the GUI machine that asks all IPv4 and IPv6-addresses of the machine. Only show the name and the IP addresses of all interfaces.

invoke-command -ComputerName GUI-BEN -ScriptBlock { Get-NetIPAddress | select PSComputername, ipaddress }

* 1. Advanced question: for the question above, you will typically combine two PS cmdlets in a “pipe” construct. This pipe (and the second cmdlet) can be included inside the Invoke-Command environment (the second cmdlet is then also executed on the GUI server) OR outside this environment (the second cmdlet is executed on the Win11 machine). Try both options and compare them. What (type of) data is returned to your Win11 machine by the “Invoke-Command” part of your command in each case?

One is a list and the other one a table

1. To be able to fully manage your servers remotely an additional firewall rule needs to be configured. Use the command of the previous question (so run this from your windows 11) to enable a firewall-rule. Tip: The firewall rule you need to configure remotely:
   1. enable-netfirewallrule -displaygroup "Windows Defender Firewall Remote Management"
   2. Do this **first** for the **GUI server** (not yet on your CORE).
   3. Now open the **mmc.exe** tool on the Win11 machine (cfr lectures from last year). Use the menu to “add/remove snap-ins”, go to “Windows Defender Firewall “ and choose another computer. Add your GUI-server. If everything went well, you should now get the initial overview of the firewall rules (don’t click on inbound or outbound rules yet).
   4. Try to add your CORE to the mmc tool as well and verify that you are unable to get an overview because the firewall rule used in “a” has not been configured yet for the CORE server.
   5. Finally enable the firewall rule for your CORE as well and verify that the connection now works correctly.
   6. Now try to go to the inbound rules on one of the servers. Depending on your setup, this may cause an error/exception, caused by a long-term bug in mmc.exe (Yes, “big software has bugs”). On Windows 10 machines, remote administration of firewall rules will almost always fail, Windows 11 machines may require restarting mmc.exe from time to time. In case you run into problems, use mmc.exe on the GUI server (or use Windows Admin Center for remote administration).
2. Use the mmc console created in the previous question to configure (activate) the following inbound rule groups on both your GUI server and CORE server. Afterwards you can close the console.
   1. File and Printer Sharing
   2. Remote Service Management
   3. Remote Event Log Management
   4. Remote Scheduled Tasks Management
   5. Remote Volume Management
3. To test some of the previous settings, open up the server manager on your win11 machine. Connect with your CORE server and open the GUI tool for computer management. Create a local user with the name “corelocaluser” and once again password: “Friday13th!”. Make sure the password does not need to be changed and that it won’t expire. This should work.

## Windows Admin Center

Because Windows Admin Center (WAC) is relatively new and less frequently used in the field, we don’t use it intensively yet in these labs. However, it should be noted that most (if not all) the (remote) administration operations that you have performed in this lab can also be done in WAC from the Win11 machine. In this section, we’ll explore the WAC interface for basic remote server management.

1. Make sure you are logged in as the domain administrator on your Win11 machine, and start WAC. If requested, select to use the default WAC profile (cf. lab 1). You are now in the “All Connections” view, where you can already see the link to the Win11 machine’s configuration page. Note that this machine is now shown as part of your domain (check the full name)! Now use the “+ Add” button to also add links to the GUI and CORE servers. Hint: Search the Active Directory domain for all servers. If you are successful, links to the two servers should also appear in the “All connections” page.
2. Now connect to the CORE server. This should open the “Server manager” page for this server. Browse the different sub-pages (called “plugins”) that are available, and find out if it would be possible to do the following operations:
   1. Add/remove **local** users to the server 🡺 possible
   2. Add/remove **domain** users 🡺 not possible
   3. Add/remove roles and features to the server 🡺 possible
   4. Change the firewall rules of the server 🡺 possible
3. Go back to the “All connections” page, and connect to the GUI server. Because this is a domain controller, some options and plugins will act differently, compared to the CORE server. Find out if it would be possible to do the following operations:
   1. Add/remove **local** users to the server 🡺 this tool cannot be used by domain controller
   2. Add/remove **domain** users 🡺 this tool cannot be used by domain controller
4. Even though this does not show up at first, it is in fact possible to manage domain users in WAC when you are connected to the domain controller. However, the plugin that is needed for this, is considered experimental (an incomplete beta) by Microsoft, and it has to be manually added in WAC. To do this, go to the WAC “Settings” (gear icon in the top right), and open the “extensions” page. Here you can install the “Active Directory” plugin. Do this, and go back to the server management page of the GUI server. Now the additional “Active Directory” plugin should be visible. Can you find the domain users?

Yes

## Pester tests

Let’s prepare to run the pester test file for this lab, and check if the most important settings are configured correctly. Last week, we used a “legacy” (but officially deprecated) way to configure pester, but now we’ll use the recommended configuration with PowerShell objects. We’ll also use the $PROFILE environment to automatically re-use these configurations for the next labs.

1. Add the following lines to the $PROFILE file: (look at <https://pester.dev/docs/usage/configuration> and <https://pester.dev/docs/commands/New-PesterConfiguration> to understand what we are doing here) (replace <firstname> by your own name)

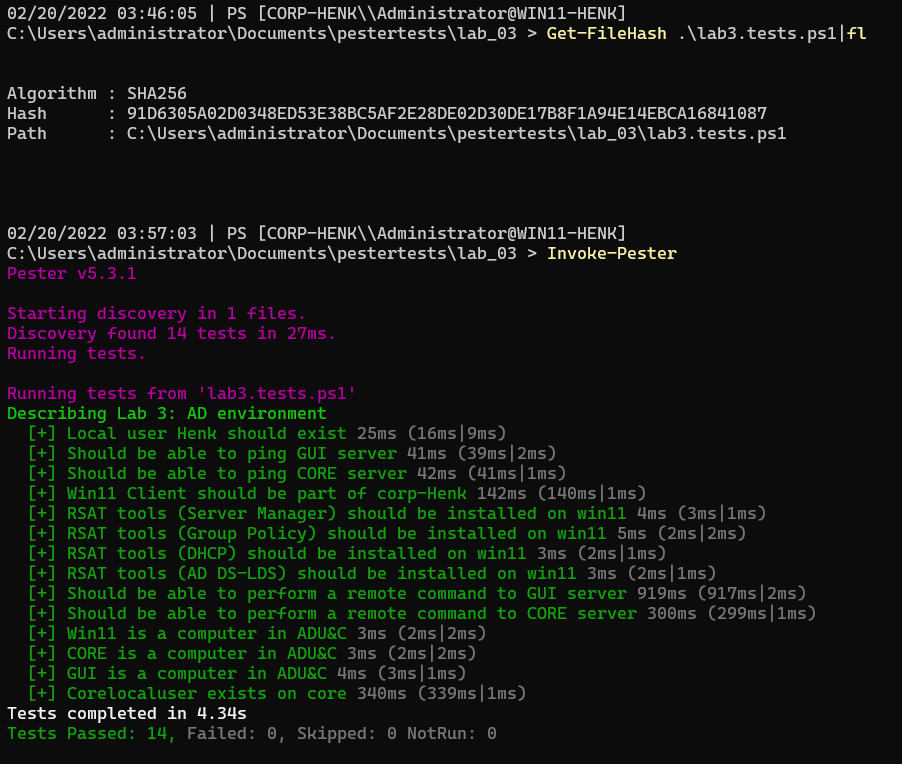
$PesterPreference = New-PesterConfiguration;

$PesterPreference.output.verbosity = 'Detailed';

$username="<firstname>" ; $domain="corp-<firstname>";

Now restart the powershell window to activate the new $PROFILE configuration.

1. Download the pester script for this lab from Leho and run it as indicated in the screenshot below (make sure that all 3 VMs are running and run Invoke-Pester in the folder on the Win11 machine where the pester script is located). Create your own screenshot and upload it to leho. Make sure that the following things are visible on the screenshot:
   1. The **personalized prompt**
   2. The commands **Get-FileHash** and **Invoke-Pester**
   3. The **complete output** for both commands



## Recap Script

Create a lab3-summary.txt file (containing one or multiple scripts) that focusses on **documentation** first (= it is fine if it is a series of commands), that:

* Renames the virtual machines to the proper names asked in the labs.
* Changes the IP/DNS settings accordingly
* Adds the machines to the domain
* Installs the RSAT/AD tools & cmdlets
* Modifies the firewall rules where necessary

Copy paste your file below and upload it on leho.

param([Parameter(Mandatory)][string] $newName)

Rename-Computer -NewName "$($newName)"

New-NetIPAddress -InterfaceIndex 3 -IPAddress 192.168.0.100 -PrefixLength 24 -DefaultGateway 192.168.0.2

Set-DnsClientServerAddress -InterfaceIndex 3 -ServerAddresses 192.168.0.2

Add-Computer -DomainName corp-ben.serverlabs.be

add-WindowsCapability -name "Rsat.ServerManager.Tools~~0.0.1.0" -online

add-WindowsCapability -name "Rsat.GroupPolicy.Management.Tools0.0.1.0" -online

add-WindowsCapability -name " Rsat.DHCP.Tools0.0.1.0" -online

add-WindowsCapability -name " Rsat.ActiveDirectory.DS-LDS.Tools~~0.0.1.0" -online

netsh advfirewall firewall add rule name="ICMP Allow incoming V4 echo request" protocol=icmpv4:8,any dir=in action=allow

netsh advfirewall firewall add rule name="ICMP Allow incoming V6 echo request" protocol=icmpv6:8,any dir=in action=allow

## Extensions - Optional Assignments

* Populate your $PROFILE environment to add functionality for:
  + A Linux “du” variant in PowerShell
  + (Advanced) Transform the output of the “quser” legacy command to a PowerShell object that can be used in PowerShell
* Install the network scanner nmap and scan your windows environment. What ports are open? Can you find the exact versions and types of the operating system?