

TSSR - Compte rendu TP7: Scripts Powershell Sous Debian et Windows Serveur

Jean-Paul MELISSE

30 Janvier 2025

1 Introduction

L'objectif de ce TP est de nous familiariser avec les scripts Powershell sous Debian. Pour cela, on va créer une série de 10 petits scripts sur une machine virtuelle Debian. Ce petit document va nous décrire les étapes à suivre.

2 Créer un dossier de travail

On choisit un emplacement de travail où on va garder tous les fichiers installés et la configuration de la machine Debian. On décide de sauvegarder dans le chemin : "C:\Users\Jean-Paul\Desktop\tp7\", comme montre la Figure 1.

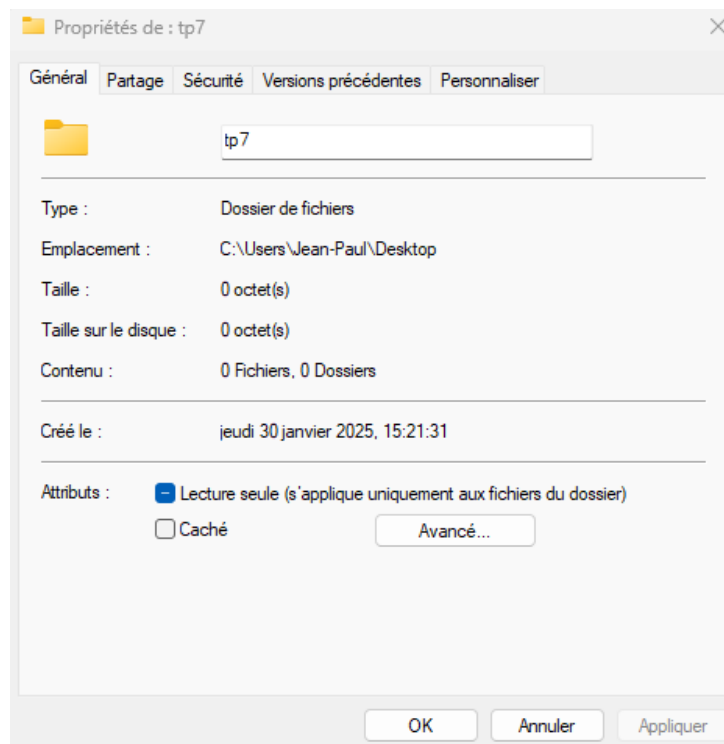


Figure 1

3 Télécharger Debian

La suite est de télécharger le système d'exploitation Debian. On y accède via le lien <https://www.debian.org/>. On clique sur "Téléchargement". On le sauvegarde aussi dans le dossier tp4 sous le format .iso (Comme montre la Figure 2).

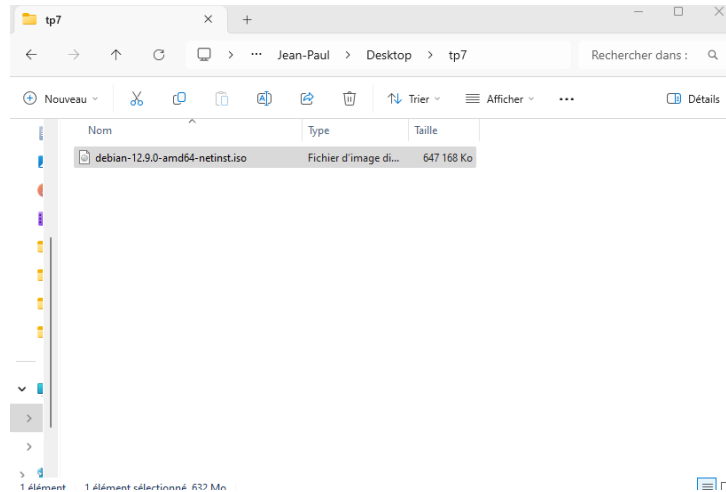


Figure 2

4 Télécharger Windows Serveur

Dans ce TP, on aura besoin aussi d'un système d'exploitation Windows. On va donc le télécharger via le lien <https://www.microsoft.com/en-us/evalcenter/download-windows-server-2022>. On choisit le carré "English (United States)" et le carré "ISO Download", et on clique sur "64-bit edition". On le sauvegarde dans le dossier tp7 sous le format .iso (Voir la Figure 3).

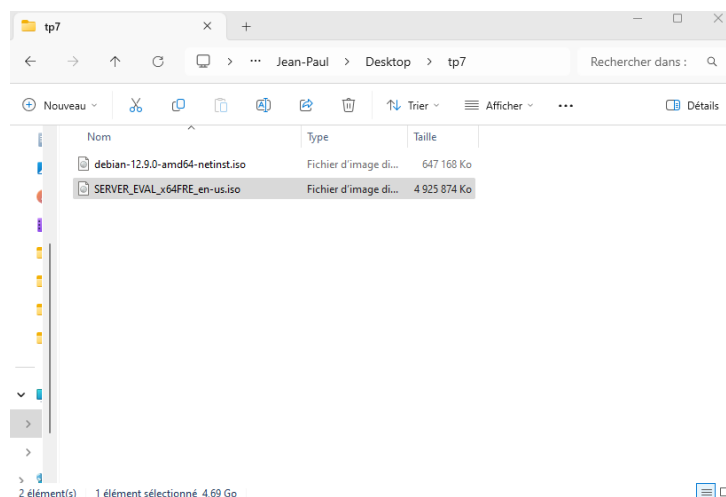


Figure 3

5 Créer la machine virtuelle tp7-debian

On ouvre VirtualBox. On clique sur "Nouveau" pour créer une nouvelle machine virtuelle. On met "tp7-debian" comme nom de la machine virtuelle. On choisit son emplacement de sauvegarde et on

importe le CD contenant le système d'exploitation de Debian. En détectant le CD, le type "Linux", le subtype "Debian" et la version "Debian (64 bits)" sont automatiquement mis. On coche la case "Skip Unattended Installation" et on clique sur "Suivant". On lui donne une mémoire vive (RAM) de 2048 Mo et un processeur, et on clique sur "Suivant". On lui donne un disque dur de 20 Go et on clique sur "Suivant". On regarde bien le récapitulatif et on clique sur "Finish" (Voir la Figure 4).

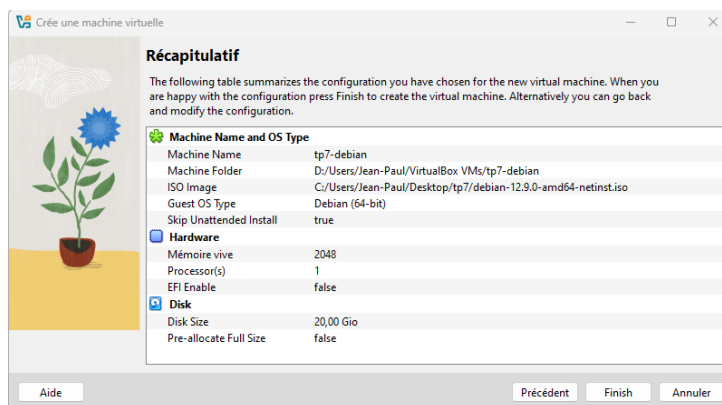


Figure 4

Maintenant que la nouvelle machine est créée, il faut la configurer. On clique sur "Configurations". On clique sur "Réseau" et on change le mode d'accès réseau à "Accès par pont", puis on clique sur "OK". Le mode "Accès par pont" va permettre à la machine virtuelle de récupérer une adresse IP directement depuis un serveur DHCP (par exemple : notre box). Cela permet de connecter la machine virtuelle à l'Internet. Maintenant qu'on a configuré la machine virtuelle, on clique sur "Démarrer" pour l'allumer.

6 Installer Debian sur tp7-debian

Après avoir démarré la machine virtuelle, on choisit l'option "Graphical Install". On choisit la langue "English" et on clique sur "Continue". Le fait qu'on ait choisi l'anglais comme langue, Debian va nous proposer des pays anglophones par défaut pour la géolocalisation. On choisit donc "other" et on clique sur "Continue". Il nous propose donc dans un premier temps, des zones continentales. On choisit "Europe" pour la localisation car on se trouve en Europe et on clique sur "Continue". Enfin, on choisit "France" pour la localisation et on clique sur "Continue". Debian comprend qu'il y a une incohérence entre la langue choisie et le pays. Il nous propose différentes versions de la langue anglaise en fonction des pays. On choisit "United States - en_US.UTF-8" et on clique sur "Continue". Pour la configuration du clavier, on choisit "French" et on clique sur "Continue". Après avoir reçu par le service DHCP une adresse IP, on renomme le nom de l'hôte (Hostname) en "tp7-debian" et on clique sur "Continue" (Figure 5).

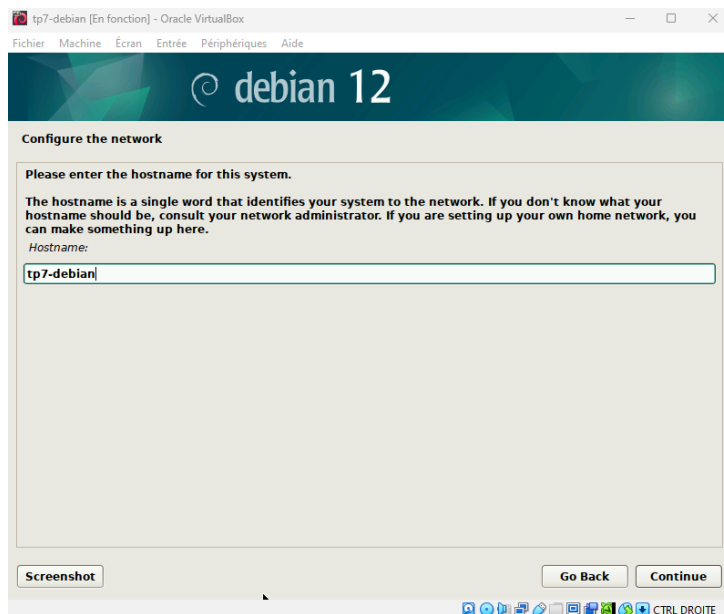


Figure 5

On lui donne un nom de domaine (Domain Name), qui sera "afpa" ici et on clique sur "Continue". On donne un mot de passe à l'administrateur root (dans ce cas ici, le mot de passe est "root" et on le met deux fois) et on clique sur "Continue". On crée un nouvel utilisateur qu'on nomme "user" et on clique sur "Continue" deux fois. Comme pour l'administrateur root, on donne un mot de passe à l'utilisateur user (dans ce cas ici, le mot de passe est "user" et on le met deux fois) et on clique sur "Continue".

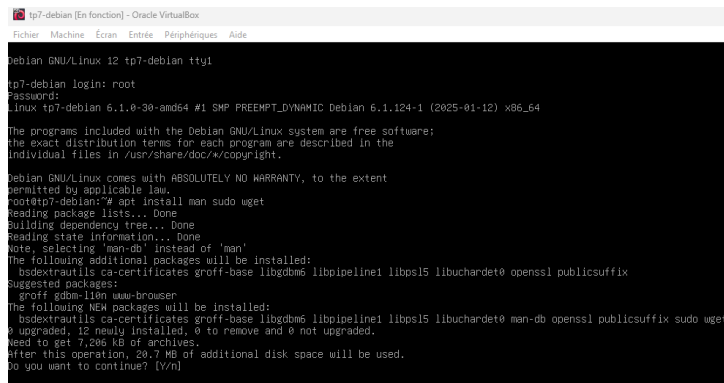
On choisit l'option "Guide - use entire disk" pour la partition du disque et on clique sur "Continue" deux fois. Pour la partition, on choisit "All files in one partition" et on clique sur "Continue" deux fois. On coche la case "Yes" pour appliquer les changements sur la partition du disque et on clique sur "Continue". Comme on a mis un seul disque dur pour cette machine virtuelle, on n'a pas besoin de scanner d'autres disques. On coche la case "No" et on clique sur "Continue".

La prochaine étape concerne le gestionnaire de paquets "Advanced Packaging Tool" (APT). Ce gestionnaire contient tous les programmes et applications utilisés pour les machines Debian et Ubuntu. Il gère aussi les mises à jour de ces applications. On choisit le pays "France" et on clique sur "Continue". On choisit le miroir pour le gestionnaire de paquets : "deb.debian.org" et on clique sur "Continue" deux fois. Ce qui signifie que quand on va utiliser le gestionnaire APT, il va se connecter à ce miroir pour vérifier ou installer les paquets. On coche la case "No" car on ne veut pas participer aux études statistiques et on clique sur "Continue".

On décoche toutes les cases des logiciels et on clique sur "Continue". On coche la case "Yes" pour installer le boot GRUB et on clique sur "Continue". On choisit l'option "/dev/sda" pour installer GRUB dans cet emplacement et on clique sur "Continue". On clique sur "Continue" pour finaliser l'installation de Debian et redémarrer la machine. Note : Après le redémarrage de Debian, on peut faire un clone de notre machine virtuelle.

7 Installer les paquets nécessaires sur tp7-debian

Le but ici est d'installer les paquets nécessaires pour le bon fonctionnement de notre debian. Après le redémarrage de notre machine virtuelle, on se connecte avec les identifiants de l'utilisateur root. On tape la commande "apt install man sudo wget" pour installer le paquet manuel, la commande sudo et le paquet wget pour récupérer des fichiers à partir de serveurs HTTP, HTTPS ou FTP. (Voir la Figure 6).



```
tp7-debian [En fonction] - Oracle VM VirtualBox
Fichier  Machine  Écran  Entrée  Périphériques  Aide

debian GNU/Linux 12 tp7-debian tty1
tp7-debian login: root
password:
linux tp7-debian 6.1.0-30-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.124-1 (2025-01-12) x86_64

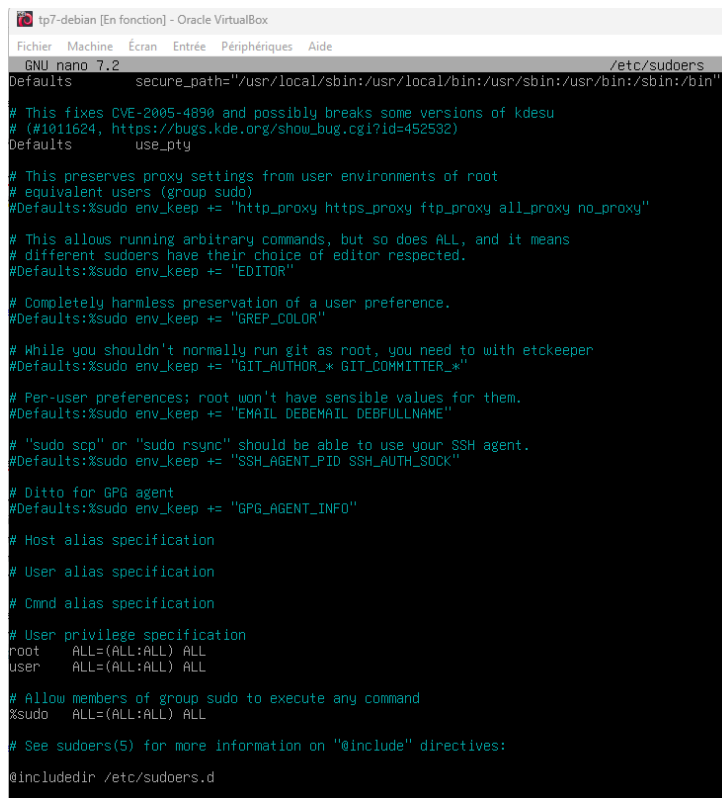
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@tp7-debian:~# apt install man sudo wget
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'man-db' instead of 'man'
The following additional packages will be installed:
  bsdextrautils ca-certificates groff-base libgdcm libpipeline1 libpsl5 libuchardet0 openssl publicsuffix
Suggested packages:
  groff gdm-l10n www-browser
The following NEW packages will be installed:
  bsdextrautils ca-certificates groff-base libgdcm libpipeline1 libpsl5 libuchardet0 man-db openssl publicsuffix sudo wget
0 upgraded, 12 newly installed, 0 to remove and 0 not upgraded.
Need to get 7,266 kB of archives.
After this operation, 28.7 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Figure 6

8 Configurer sudo sur tp7-debian

La suite maintenant est de configurer la commande sudo afin qu'on puisse l'utiliser avec l'utilisateur user. Après avoir installé sudo, on va modifier le fichier sudoers pour ajouter l'utilisateur user dans les droits de permissions. On fait une copie du fichier sudoers en tapant la commande "cp /etc/sudoers /etc/sudoers.ori" pour éviter d'écraser le fichier original. Pour modifier ce fichier, on tape la commande : "nano /etc/sudoers". Dans ce fichier, on ajoute après la ligne du root la commande : "user ALL=(ALL:ALL) ALL". On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée" pour sauvegarder avec le même nom. On peut visualiser cette étape sur la Figure 7.

A screenshot of a terminal window titled "tp7-debian [En fonction] - Oracle VirtualBox". The terminal shows the nano 7.2 editor editing the file /etc/sudoers. The content of the file is as follows:

```
Defaults    secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"

# This fixes CVE-2005-4890 and possibly breaks some versions of kdesu
# (#1011624, https://bugs.kde.org/show_bug.cgi?id=452532)
Defaults    use_pty

# This preserves proxy settings from user environments of root
# equivalent users (group sudo)
#Defaults:sudo env_keep += "http_proxy https_proxy ftp_proxy all_proxy no_proxy"

# This allows running arbitrary commands, but so does ALL, and it means
# different sudoers have their choice of editor respected.
#Defaults:sudo env_keep += "EDITOR"

# Completely harmless preservation of a user preference.
#Defaults:sudo env_keep += "GREP_COLOR"

# While you shouldn't normally run git as root, you need to with etckeeper
#Defaults:sudo env_keep += "GIT_AUTHOR_* GIT_COMMITTER_*"

# Per-user preferences; root won't have sensible values for them.
#Defaults:sudo env_keep += "EMAIL DEBEMAIL DEBFULLNAME"

# "sudo scp" or "sudo rsync" should be able to use your SSH agent.
#Defaults:sudo env_keep += "SSH_AGENT_PID SSH_AUTH_SOCK"

# Ditto for GPG agent
#Defaults:sudo env_keep += "GPG_AGENT_INFO"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
user    ALL=(ALL:ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "@include" directives:

@include /etc/sudoers.d
```

Figure 7

On a fini de configurer le fichier sudo. On peut se déconnecter du compte en utilisant la commande "logout".

9 Installer powershell sur tp7-debian

On se connecte avec les identifiants de l'utilisateur user. On tape la commande : "sudo apt-get update" pour mettre à jour tous les paquets disponibles du gestionnaire apt. Puis on tape la commande : "wget https://github.com/PowerShell/PowerShell/releases/download/v7.4.6/powershell_7.4.6-1.deb_amd64.deb" pour récupérer powershell depuis l'archive github. Puis, on tape la commande : "sudo dpkg -i powershell_7.4.6-1.deb_amd64.deb" pour extraire le paquet powershell et l'ajouter dans l'archive apt-get. Enfin, on tape la commande : "sudo apt-get install -f" pour installer powershell sur la machine. On peut voir si powershell a bien été installé en tapant la commande : "pwsh" et c'est bien le cas (voir la Figure 8).

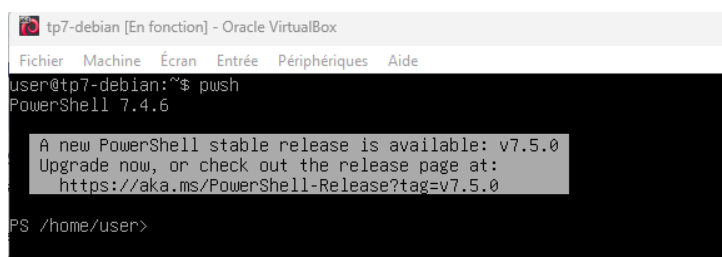
A screenshot of a terminal window titled "tp7-debian [En fonction] - Oracle VirtualBox". The terminal shows the user@tp7-debian:~\$ pwsh command being executed. The output is: PowerShell 7.4.6. Below this, a message box appears with the text: "A new PowerShell stable release is available: v7.5.0. Upgrade now, or check out the release page at: https://aka.ms/PowerShell-Release?tag=v7.5.0". At the bottom, the prompt PS /home/user> is visible.

Figure 8

10 Les sorties en PowerShell

Par la suite du TP, on se concentre sur les scripts powershell. On crée un nouveau dossier "tp7-jean-paul-melisse" qui va contenir tous les scripts Bash. Puis on se rend dans ce dossier en tapant la commande "cd tp7-jean-paul-melisse/"

Les entrées-sorties désignent les échanges de données entre le processeur (CPU) et les périphériques. Les entrées sont les données envoyées par les périphériques d'entrée (clavier, scanner, . . .) et lues par le CPU. Les sorties sont les données reçues par les périphériques de sortie (écran, imprimante, . . .) et écrites par le CPU. Dans cette section, nous nous intéressons aux sorties et à écrire à l'écran. Pour écrire à l'écran, nous utilisons la commande Write-Host. Pour en savoir plus sur l'utilisation de cette commande, tapez Get-Help Write-Host. On va créer un script 01-write.ps1 qui écrit hello world à l'écran. Pour cela, on tape la commande "sudo nano 01-write.ps1". Dans ce fichier, on écrit la ligne : 'Write-Host "hello world"', comme montré sur la Figure 9.

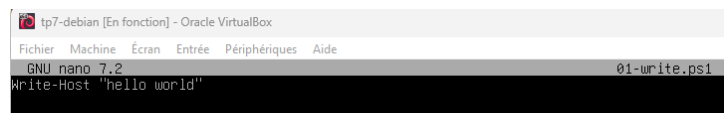


Figure 9

On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée" pour sauvegarder. On peut vérifier l'exécution du script en tapant la commande : "pwsh 01-write.sh" (Voir la Figure 10).

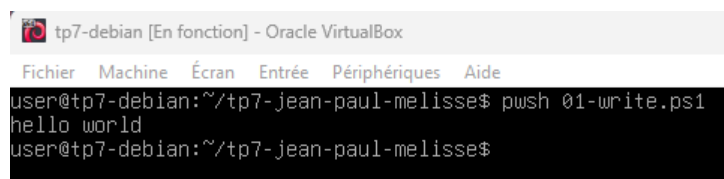


Figure 10

11 Les variables en PowerShell

Les variables permettent d'affecter une valeur à un nom. Le nom d'une variable commence toujours par \$ (dollar). Pour créer une variable, il suffit de lui affecter une valeur, exemple : \$name = "value". Pour utiliser une variable créée, il suffit d'indiquer son nom, exemple : \$name. On va créer un script 02-variable.ps1 qui crée la variable \$first_name en lui affectant la valeur samson et qui l'écrit à l'écran. Pour cela, on tape la commande "sudo nano 02-variable.ps1". Dans ce fichier, on écrit les lignes suivantes :

- 1 \$first_name = "samson"
- 2 Write-Host hello \$first_name

On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée" pour sauvegarder. On peut voir l'étape précédente sur la Figure 11. On vérifie l'exécution du script en tapant la commande : "pwsh 02-variable.ps1" (voir la Figure 12).

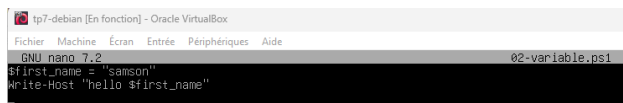


Figure 11

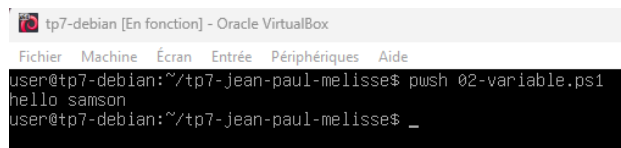


Figure 12

12 Les entrées en PowerShell

Dans cette section, nous nous intéressons aux entrées et à lire au clavier. Pour lire au clavier, nous utilisons la commande "Read-Host". On va créer un script "03-read.ps1" qui va lire au clavier une chaîne de caractères représentant un prénom et qui l'écrit à l'écran. On crée aussi un alias "write" pour "Write-Host", et un alias "read" pour "Read-Host" et qui utilise ces alias pour lire au clavier une chaîne de caractères représentant un prénom et l'écrire à l'écran. On tape la commande "sudo nano 03-read.ps1". Dans ce fichier, on écrit les lignes suivantes :

- 1 Set-Alias write Write-Host
- 2 Set-Alias read Read-Host
- 3 \$first_name = read "first name? "
- 4 write "Your name is : \$first_name"

On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée" pour sauvegarder. On peut voir l'étape précédente sur la Figure 13. On vérifie l'exécution du script en tapant la commande : "pwsh 03-read.ps1" (voir la Figure 14).

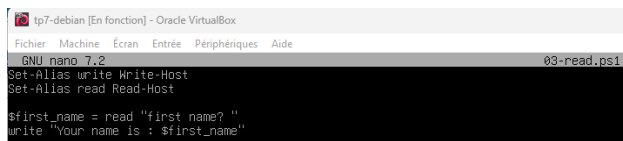


Figure 13

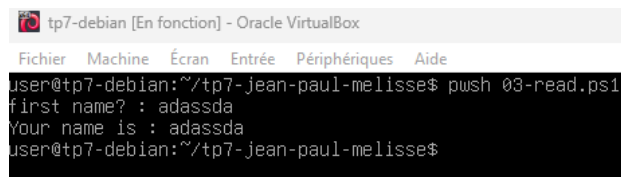


Figure 14

13 Les conditions en PowerShell

Bien que les instructions d'un script soient habituellement exécutées de façon séquentielle, il est possible d'en exécuter à l'intérieur d'une structure conditionnelle.

- Syntaxe :

- 1 if (CONDITION-1)
- 2 {
- 3 STATEMENT-1
- 4 }
- 5 elseif (CONDITION-2)


```

6 {
7     STATEMENT-2
8 }
9 else
10 {
11     STATEMENT-3
12 }

```

On va créer un script 04-grades.ps1 qui lit au clavier un entier représentant une note sur 20, écrit que c'est un échec si la note est entre 0 et 9 et écrit que c'est un succès si la note est entre 10 et 20. Si la note est en dehors de ces deux plages, le script doit écrire un message d'erreur et retourner 1 au lieu de 0. Si la note ne peut pas être convertie vers un entier, le script doit écrire un message d'erreur et retourner 1 au lieu de 0. On tape la commande "sudo nano 04-grades.ps1". Dans ce fichier, on écrit les lignes suivantes :

```

1 $grade = Read-Host "grade? "
2 if (-not ($grade -match '[+-]?\d+$' ))
3 {
4     Write-Host "error: unable to convert the grade to an integer"
5     exit 1
6 }
7 $grade = [int]$grade
8 if ($grade -ge 0 -and $grade -le 9)
9 {
10     Write-Host "faillure"
11     exit 0
12 }
13 elseif ( $grade -ge 10 -and $grade -le 20)
14 {
15     Write-Host "success"
16     exit 0
17 }
18 else

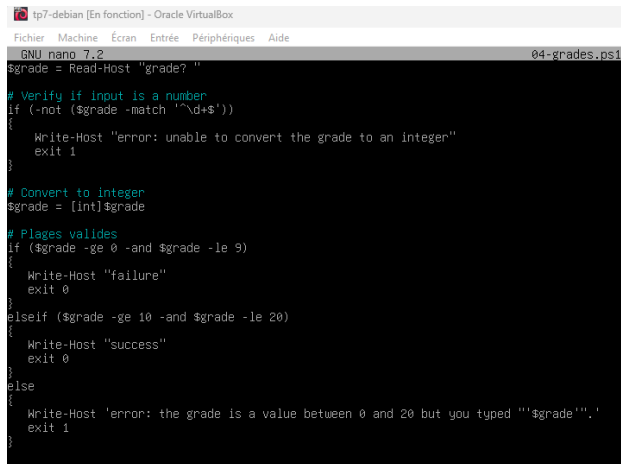
```

```

18 {
19     Write-Host 'error : the grade is a value between 0 and 20 but you typed "'$GRADE "'"
20     exit 1
21 }

```

On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée" pour sauvegarder. On peut voir l'étape précédente sur la Figure 15. On vérifie l'exécution du script en tapant la commande : "pwsh 04-grades.ps1" (voir la Figure 16).



```

04-grades.ps1
$grade = Read-Host "grade? "

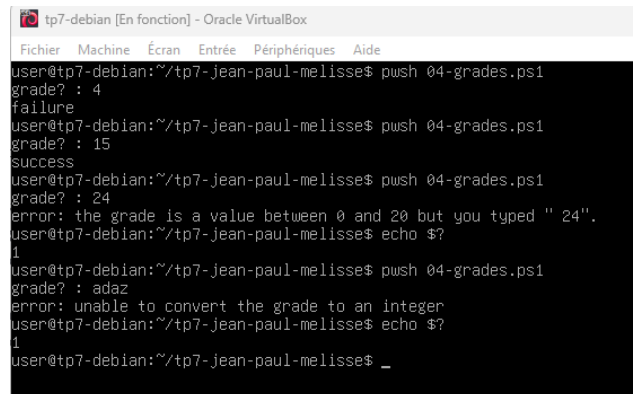
# Verify if input is a number
if (-not ($grade -match "^\d+$"))
{
    Write-Host "error: unable to convert the grade to an integer"
    exit 1
}

# Convert to integer
$grade = [int]$grade

# Plages valides
if ($grade -ge 0 -and $grade -le 9)
{
    Write-Host "failure"
    exit 0
}
elseif ($grade -ge 10 -and $grade -le 20)
{
    Write-Host "success"
    exit 0
}
else
{
    Write-Host 'error: the grade is a value between 0 and 20 but you typed "'$grade'"'
    exit 1
}

```

Figure 15



```

user@tp7-debian:~/tp7-jean-paul-melisse$ pwsh 04-grades.ps1
grade? : 4
failure
user@tp7-debian:~/tp7-jean-paul-melisse$ pwsh 04-grades.ps1
grade? : 15
success
user@tp7-debian:~/tp7-jean-paul-melisse$ pwsh 04-grades.ps1
grade? : 24
error: the grade is a value between 0 and 20 but you typed " 24".
user@tp7-debian:~/tp7-jean-paul-melisse$ echo $?
1
user@tp7-debian:~/tp7-jean-paul-melisse$ pwsh 04-grades.ps1
grade? : adaz
error: unable to convert the grade to an integer
user@tp7-debian:~/tp7-jean-paul-melisse$ echo $?
1
user@tp7-debian:~/tp7-jean-paul-melisse$ _

```

Figure 16

14 Les expression arithmétiques en PowerShell

Les expressions arithmétiques permettent d'effectuer des calculs (additions, soustractions, . . .). On crée un script 05-calculator.ps1 qui lit au clavier une chaîne de caractères représentant un opérateur, lit un entier représentant une première opérande, lit un entier représentant une seconde opérande et applique l'opérateur sur les opérandes. Si l'opérateur n'est pas supporté, le script doit écrire un message d'erreur et retourner 1 au lieu de 0. Si une opérande ne peut pas être convertie vers un entier, le script doit écrire un message d'erreur et retourner 1 au lieu de 0. Pour cela, on tape la commande "sudo nano 05-calculator.ps1". Dans ce fichier, on écrit les lignes suivantes :

- 1 Write-Host "addition"
- 2 Write-Host "substraction"
- 3 Write-Host "multiplication"
- 4 Write-Host "division"
- 5 Write-Host "remainder"
- 6 Write-Host "exponentiation"
- 7

```

8 $operator = Read-Host "operator? "
9
10 try
11 {
12     $first_operand = Read-Host "first operand? "
13     if (-not ($first_operand -match '[+-]?\d+$' ))
14     {
15         throw "unable to convert 1st operand to an integer"
16     }
17     $second_operand = Read-Host "second operand? "
18     if (-not ($second_operand -match '[+-]?\d+$' ))
19     {
20         throw "unable to convert 2nd operand to an integer"
21     }
22     $first_operand = [int]$first_operand
23     $second_operand = [int]$second_operand
24 }
25 catch
26 {
27     Write-Host "error: $_"
28     exit 1
29 }
30
31 try
32 {
33     switch ($operator)
34     {
35         "addition"
36         {

```

```

37         $result = $first_operand + $second_operand
38         Write-Host "result: $result"
39         exit 0
40     }
41     "subtraction"
42     {
43         $result = $first_operand - $second_operand
44         Write-Host "result: $result"
45         exit 0
46     }
47     "multiplication"
48     {
49         $result = $first_operand * $second_operand
50         Write-Host "result: $result"
51         exit 0
52     }
53     "division"
54     {
55         if ($second_operand -eq 0)
56         {
57             throw "Division by zero"
58         }
59         $result = $first_operand / $second_operand
60         Write-Host "result: $result"
61         exit 0
62     }
63     "remainder"
64     {
65         if ($second_operand -eq 0)

```

```

66         {
67             throw "Division by zero"
68         }
69         $result = $first_operand % $second_operand
70         Write-Host "result: $result"
71         exit 0
72     }
73     "exponentiation"
74     {
75         $result = [Math]::Pow($first_operand, $second_operand)
76         Write-Host "result: $result"
77         exit 0
78     }
79     default
80     {
81         throw "the operator is addition, substraction, multiplication, division, remain-
            der, or 'nexponentiation but you typed '$operator'."
82     }
83 }
84 }
85 catch
86 {
87     Write-Host "error: $_"
88     exit 1
89 }

```

On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée" pour sauvegarder. On peut voir une partie de l'étape précédente sur la Figure 17. On vérifie l'exécution du script en tapant la commande : "pwsh 05-calculator.ps1" (voir la Figure 18).

```

GNU nano 7.2                                05-calculator.ps1
Write-Host "addition"
Write-Host "subtraction"
Write-Host "multiplication"
Write-Host "division"
Write-Host "remainder"
Write-Host "exponentiation"

$operator = Read-Host "operator? "

# Verify if operands can be convert into integers
try
{
    $first_operand = Read-Host "first operand? "
    if (-not ($first_operand -match "\d+$"))
    {
        throw "unable to convert the 1st operand to an integer"
    }

    $second_operand = Read-Host "second operand? "
    if (-not ($second_operand -match "\d+$"))
    {
        throw "unable to convert the 2nd operand to an integer"
    }

    # Convert operands to integers
    $first_operand = [int]$first_operand
    $second_operand = [int]$second_operand
}
catch
{
    Write-Host "error: $_"
    exit 1
}

# Doing operation in terms of the operator
try
{
    switch ($operator)
    {
        "addition"
        {
            $result = $first_operand + $second_operand
            Write-Host "result: $result"
            exit 0
        }
    }
}

```

Figure 17

```

tp7-debian [En fonction] - Oracle VirtualBox
Fichier  Machine  Écran  Entrée  Périphériques  Aide

user@tp7-debian:~/tp7-jean-paul-melisse$ push 05-calculator.ps1
addition
subtraction
multiplication
division
remainder
exponentiation
operator? : addition
first operand? : 5
second operand? : 4
result: 9
user@tp7-debian:~/tp7-jean-paul-melisse$ push 05-calculator.ps1
addition
subtraction
multiplication
division
remainder
exponentiation
operator? : division
first operand? : 4
second operand? : 0
error: Division by zero
user@tp7-debian:~/tp7-jean-paul-melisse$ echo $?
1
user@tp7-debian:~/tp7-jean-paul-melisse$ push 05-calculator.ps1
subtraction
multiplication
division
remainder
exponentiation
operator? : addition
first operand? : e
error: unable to convert the 1st operand to an integer
user@tp7-debian:~/tp7-jean-paul-melisse$ echo $?
1
user@tp7-debian:~/tp7-jean-paul-melisse$ push 05-calculator.ps1
addition
subtraction
multiplication
division
remainder
exponentiation
operator? : rzevez
first operand? : 4
second operand? : 5
error: the operator is addition, subtraction, multiplication, division, remainder, or
exponentiation but you typed "rzevez".
user@tp7-debian:~/tp7-jean-paul-melisse$

```

Figure 18

15 Les itérations en PowerShell

Bien que les instructions d'un script soient habituellement exécutées de façon séquentielle, il est possible d'en exécuter à l'intérieur d'une structure itérative (boucle).

- Syntaxe :

```

1 while ( CONDITION-1 )
2 {
3     STATEMENT-1
4 }

```

On va donc créer un script 06-countdown.ps1 qui lit au clavier un entier représentant le début d'un compte à rebours, écrit cette valeur à l'écran, lui retire un, attend une seconde et recommence jusqu'à atteindre zéro. Si le début du compte à rebours ne peut pas être converti vers un entier, le script doit écrire un message d'erreur et retourner 1 au lieu de 0. On tape la commande "sudo nano 06-countdown.ps1". Dans ce fichier, on écrit les lignes suivantes :

```

1 $VALUE = Read-Host "value? "
2 try
3 {
4     if (-not ($VALUE -match '[+-]?\d+$' ))
5     {

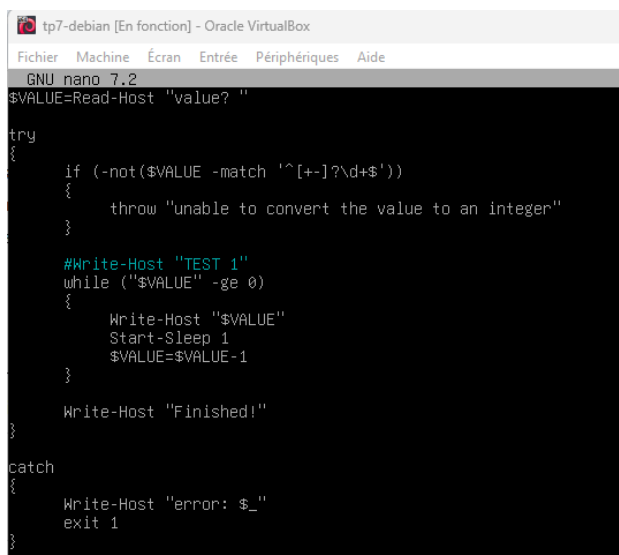
```

```

6          throw "unable to convert the value to an integer"
7      }
8
9      while ( "$VALUE" -ge 0 )
10     {
11         Write-Host "$VALUE"
12         Start-Sleep 1
13         $VALUE=$VALUE-1
14     }
15     Write-Host "Finished!"
16 }
17 catch
18 {
19     Write-Host "error: $_"
20     exit 1
21 }

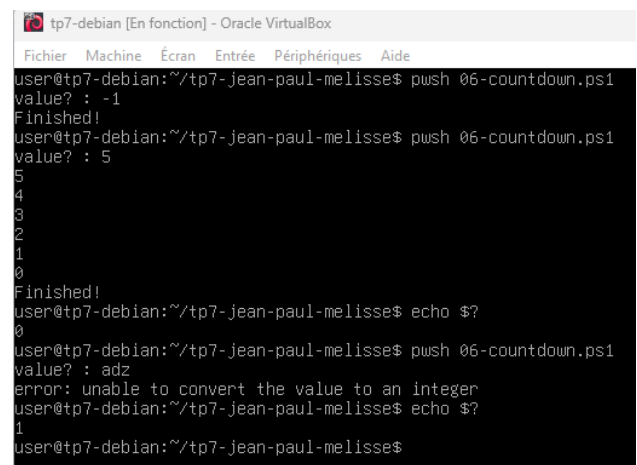
```

On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée" pour sauvegarder. On peut voir l'étape précédente sur la Figure 19. On vérifie l'exécution du script en tapant la commande : "pwsh 06-countdown.ps1" (voir la Figure 20).



The screenshot shows a terminal window titled 'tp7-debian [En fonction] - Oracle VirtualBox'. Inside, the GNU nano 7.2 editor is open, displaying a PowerShell script. The script starts with '\$VALUE=Read-Host "value? "', followed by a try block containing an if statement to check if the value is a valid integer. If not, it throws an error. Then, it enters a while loop that writes the value, sleeps for 1 second, and decrements it until it reaches 0. A catch block handles any errors by writing an error message and exiting with code 1.

Figure 19



The screenshot shows the same terminal window as Figure 19, but now the script has been executed. The user has entered 'pwsh 06-countdown.ps1'. The output shows the script running, displaying values from 5 down to 0, and then 'Finished!'. The user then enters 'echo \$?' and gets '0'. Next, the user enters 'pwsh 06-countdown.ps1' again, but this time enters 'adz' when prompted for a value. The output shows an error: 'error: unable to convert the value to an integer', followed by '1' and 'Finished!'. Finally, the user enters 'echo \$?' and gets '1'.

Figure 20

16 Les fonctions en PowerShell

Les fonctions ont un nom, des paramètres et une valeur de retour. Elles permettent de décomposer le code en plusieurs calculs indépendants. Une fonction peut être définie une fois et appelée autant de fois que nécessaire.

- Syntaxe pour définir une fonction :

```
1 function NAME
2 {
3     STATEMENT-1
4     STATEMENT-2
5 }
6 function NAME (PARAMETER-NAME-1, PARAMETER-NAME-2)
7 {
8     STATEMENT-1
9     STATEMENT-2
10 }
11 function NAME
12 {
13     param(PARAMETER-NAME-1, PARAMETER-NAME-2)
14     STATEMENT-1
15     STATEMENT-2
16 }
```

- Syntaxe pour appeler une fonction :

```
1 NAME PARAMETER-VALUE-1
```

Les paramètres peuvent être obtenus grâce à \$1 pour le premier paramètre et ainsi de suite pour les autres paramètres. La valeur de retour peut être obtenue grâce à \$? (dollar et point d'interrogation). On va créer un script "07-even.ps1" qui définit la fonction "even" à un paramètre "NUMBER" et qui retourne 0 si "NUMBER" est pair, sinon 2. Ce script lit au clavier un entier, appelle la fonction "even" en lui passant en paramètre l'entier précédemment lu au clavier et écrit à l'écran le résultat obtenu. Si le nombre ne peut pas être converti vers un entier, le script doit écrire un message d'erreur et retourner 1 au lieu de 0. Pour cela, on tape la commande "sudo nano 07-even.ps1". Dans ce fichier, on écrit les lignes suivantes :

```
1 $NUMBER = Read-Host "number? "
2 function even
3 {
4     param(
```



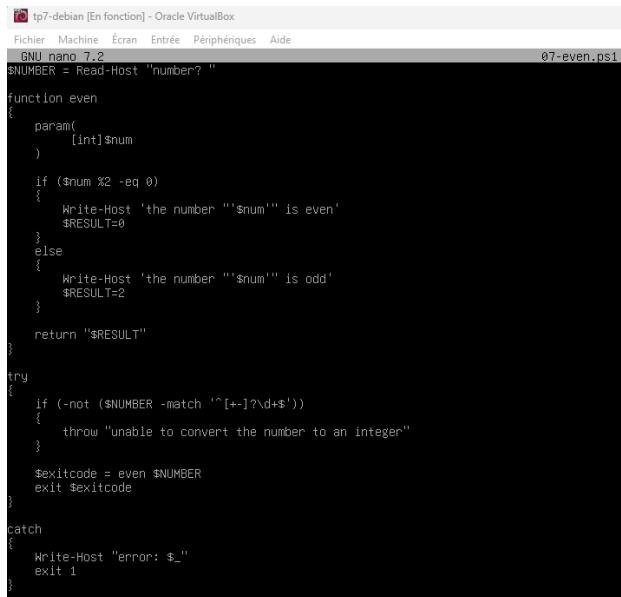
```

5         [int]$num
6     )
7     if ( $num %2 -eq 0 )
8     {
9         Write-Host 'the number "$num" is even'
10        $RESULT=0
11    }
12    else
13    {
14        Write-Host 'the number "$num" is odd'
15        $RESULT=2
16    }
17    return "$RESULT"
18 }
19
20 try
21 {
22     if (-not ($NUMBER -match '[+-]?\d+$' ))
23     {
24         throw "unable to convert the value to an integer"
25     }
26
27     $exitcode = even $NUMBER
28     exit $exitcode
29 }
30 catch
31 {
32     Write-Host "error: $_"
33     exit 1

```

34 }

On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée" pour sauvegarder. On peut voir l'étape précédente sur la Figure 21. On vérifie l'exécution du script en tapant la commande : "pwsh 07-even.ps1" (voir la Figure 22).



```
tp7-debian [En fonction] - Oracle VirtualBox
Fichier Machine Écran Entrée Périphériques Aide
GNU nano 7.2 07-even.ps1
$NUMBER = Read-Host "number? "

function even
{
    param(
        [int]$num
    )

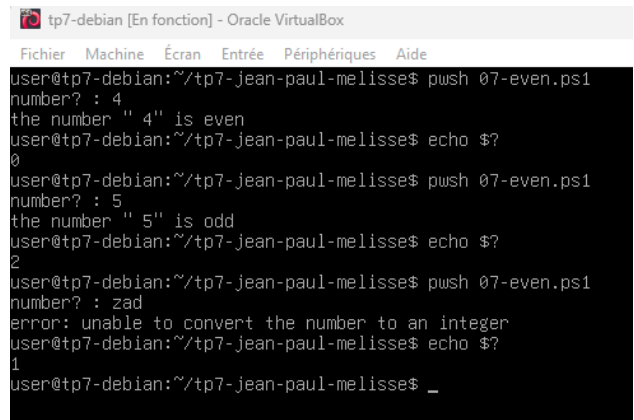
    if ($num %2 -eq 0)
    {
        Write-Host 'the number "$num" is even'
        $RESULT=0
    }
    else
    {
        Write-Host 'the number "$num" is odd'
        $RESULT=2
    }
}

return "$RESULT"

try
{
    if (-not ($NUMBER -match '^[+]?[0-9]+$'))
    {
        throw "unable to convert the number to an integer"
    }

    $exitcode = even $NUMBER
    exit $exitcode
}
catch
{
    Write-Host "error: $_"
    exit 1
}
```

Figure 21



```
tp7-debian [En fonction] - Oracle VirtualBox
Fichier Machine Écran Entrée Périphériques Aide
user@tp7-debian:~/tp7-jean-paul-melisse$ pwsh 07-even.ps1
number? : 4
the number " 4" is even
user@tp7-debian:~/tp7-jean-paul-melisse$ echo $?
0
user@tp7-debian:~/tp7-jean-paul-melisse$ pwsh 07-even.ps1
number? : 5
the number " 5" is odd
user@tp7-debian:~/tp7-jean-paul-melisse$ echo $?
2
user@tp7-debian:~/tp7-jean-paul-melisse$ pwsh 07-even.ps1
number? : zad
error: unable to convert the number to an integer
user@tp7-debian:~/tp7-jean-paul-melisse$ echo $?
1
user@tp7-debian:~/tp7-jean-paul-melisse$ _
```

Figure 22

17 Les paramètres en PowerShell

Des paramètres peuvent être passés aux scripts en ligne de commande de la même manière que des paramètres peuvent être passés aux fonctions. On crée un script 08-parameters.ps1 qui écrit à l'écran le nom du script (paramètre zéro) et les paramètres passés. On tape la commande "sudo nano 08-parameters.ps1". Dans ce fichier, on écrit les lignes suivantes :

```
1 Write-Host "program: $($MyInvocation.MyCommand.Name)"
2 while ("${args.Count} -gt 0 )
3 {
4     Write-Host "parameter:" $args[0]
5     $args = $args[1..$args.Count]
6 }
```

On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée" pour sauvegarder. On peut voir l'étape précédente sur la Figure 23. On vérifie l'exécution du script en tapant la commande : "pwsh 08-parameters.ps1" (voir la Figure 24).

```

GNU nano 7.2 08-parameters.ps1
Write-Host "program: $($MyInvocation.MyCommand.Name)"
while ($args.Count -gt 0)
{
    Write-Host "parameter:" $args[0]
    $args = $args[1..$args.Count]
}

```

Figure 23

```

user@tp7-debian:~/tp7-jean-paul-melisse$ push 08-parameters.ps1
program: 08-parameters.ps1
user@tp7-debian:~/tp7-jean-paul-melisse$ push 08-parameters.ps1 ad
program: 08-parameters.ps1
parameter: ad
user@tp7-debian:~/tp7-jean-paul-melisse$ push 08-parameters.ps1 ad zdza zdazd
program: 08-parameters.ps1
parameter: ad
parameter: zdza
parameter: zdazd
user@tp7-debian:~/tp7-jean-paul-melisse$

```

Figure 24

18 Obtenir des informations système en PowerShell

Dans cette partie du TP, on va créer un script "09-sysinfo.ps1" qui écrit à l'écran des informations sur le système. Pour cela, on tape la commande "sudo nano 09-sysinfo.ps1". Dans ce fichier, on écrit les lignes suivantes :

- 1 Write-Host "System information:"
- 2 \$CPU=\$(grep "model name" /proc/cpuinfo | sed 's/^.*: //')
- 3 Write-Host "* CPU: \$CPU"
- 4 Write-Host " * Cores: \$(grep -c 'core id' /proc/cpuinfo | sort -u | wc -l)"
- 5 Write-Host " * Threads: \$(grep -c '^processor' /proc/cpuinfo)"
- 6 \$RAM=\$(grep "MemTotal" /proc/meminfo | sed -E 's/^[^0-9]//g')
- 7 \$RAM_GB=\$((RAM/1024/1024))
- 8 Write-Host "* RAM:" \$RAM_GB.ToString("F2")" GB"
- 9 \$ROM=\$(grep 'sda\$' /proc/partitions | sed 's/^[]*[0-9]*[]*[0-9]*[]*\([0-9]*\) .*/\1/')
- 10 \$ROM_GB=\$((ROM/1024/1024))
- 11 Write-Host "* ROM:" \$ROM_GB" GB"

On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée" pour sauvegarder. On peut voir l'étape précédente sur la Figure 25. On vérifie l'exécution du script en tapant la commande : "pwsh 09-sysinfo.ps1" (voir la Figure 26).

```

GNU nano 7.2 09-sysinfo.ps1
Write-Host "System information"
$CPU = $(grep "model name" /proc/cpuinfo | sed 's/^.*: //')
Write-Host "* CPU: $CPU"
$Cores = $(grep -c 'core id' /proc/cpuinfo | sort -u | wc -l)
Write-Host " * Cores: $Cores"
$Threads = $(grep -c '^processor' /proc/cpuinfo)
Write-Host " * Threads: $Threads"
$RAM = $(grep "MemTotal" /proc/meminfo | sed -E 's/^[^0-9]//g')
$RAM_GB = $RAM/1024/1024
Write-Host "* RAM:" $RAM_GB.ToString("F2")" GB"
$ROM = $(grep 'sda$' /proc/partitions | sed 's/^[ ]*[0-9]*[ ]*[0-9]*[ ]*\([0-9]*\) .*/\1/')
$ROM_GB = $ROM/1024/1024
Write-Host "* ROM:" $ROM_GB" GB"

```

Figure 25

```

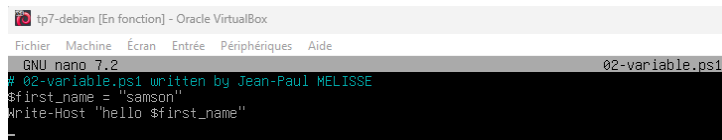
user@tp7-debian:~/tp7-jean-paul-melisse$ pwsh 09-sysinfo.ps1
System information
* CPU: Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz
 * Cores: 1
 * Threads: 1
 * RAM: 1.92 GB
 * ROM: 20 GB
user@tp7-debian:~/tp7-jean-paul-melisse$ _

```

Figure 26

19 Les commentaires en PowerShell

Pour commenter un code, il suffit de précéder le texte par # (croisillon), exemple : # this is my comment. On va ajouter un commentaire à la première ligne de tous les scripts qu'on a créés afin d'y indiquer "# <fichier> written by jean-paul melisse" (avec <fichier> à remplacer par le nom du fichier. Seuls les caractères suivants sont autorisés pour ce commentaire : 0-9 (numériques), a-z (minuscules), - (tiret) et .(point). On peut voir un exemple sur la Figure 27.

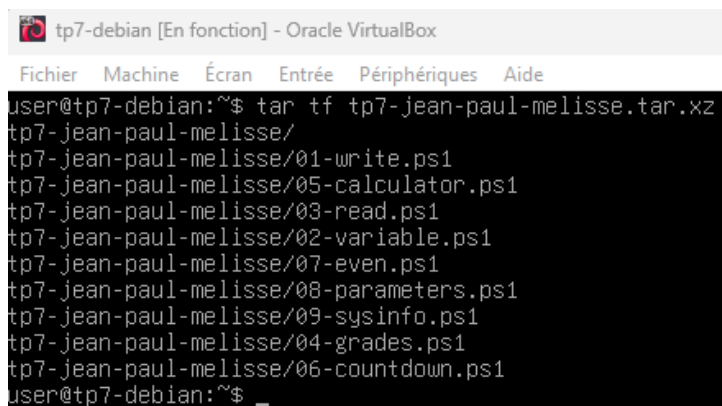
A screenshot of a PowerShell script editor window titled "tp7-debian [En fonction] - Oracle VirtualBox". The window has a menu bar with "Fichier", "Machine", "Écran", "Entrée", "Périphériques", and "Aide". The editor shows a file named "02-variable.ps1". The content of the file is:

```
# 02-variable.ps1 written by Jean-Paul MELISSE
$first_name = "samson"
Write-Host "hello $first_name"
```

Figure 27

20 Création de l'archive et son exportation

Pour cette partie de ce TP, on va archiver le dossier "tp7-jean-paul-melisse" et l'exporter sur la machine physique. Pour commencer, on revient sur le dossier parent (/home/user/) avec la commande : "cd ..". Puis, on crée l'archive "tp7-jean-paul-melisse.tar.xz" en tapant la commande "sudo tar -caf tp7-jean-paul-melisse.tar.xz tp7-jean-paul-melisse". [Attention : il est possible que les utilitaire xz ne soient pas installés. Il faut les installer avec la commande : "sudo apt install xz-utils".] On peut vérifier le contenu de l'archive en tapant la commande : "tar tf tp7-jean-paul-melisse.tar.xz | sort" (Voir la Figure 28).

A screenshot of a terminal window titled "tp7-debian [En fonction] - Oracle VirtualBox". The terminal shows the command "user@tp7-debian:~\$ tar tf tp7-jean-paul-melisse.tar.xz" and its output:

```
tp7-jean-paul-melisse/
tp7-jean-paul-melisse/01-write.ps1
tp7-jean-paul-melisse/05-calculator.ps1
tp7-jean-paul-melisse/03-read.ps1
tp7-jean-paul-melisse/02-variable.ps1
tp7-jean-paul-melisse/07-even.ps1
tp7-jean-paul-melisse/08-parameters.ps1
tp7-jean-paul-melisse/09-sysinfo.ps1
tp7-jean-paul-melisse/04-grades.ps1
tp7-jean-paul-melisse/06-countdown.ps1
user@tp7-debian:~$ _
```

Figure 28

La suite est de créer un dossier partagé. On retourne sur la machine physique et on crée le dossier "share" dans le chemin : "C:\Users\Jean-Paul\Desktop\tp7\". Puis, on va sur VirtualBox et on clique sur "tp7-debian" et "Configuration". Puis on clique sur "Shared Folders" et sur "Ajouter un nouveau dossier partagé". On met dans le chemin du dossier : "C:\Users\Jean-Paul\Desktop\tp7\share". On met le nom du dossier : "share" et on clique sur "Ok". On peut voir sur la Figure 29 que le dossier "share" a bien été partagé avec la machine virtuelle "tp7-debian".

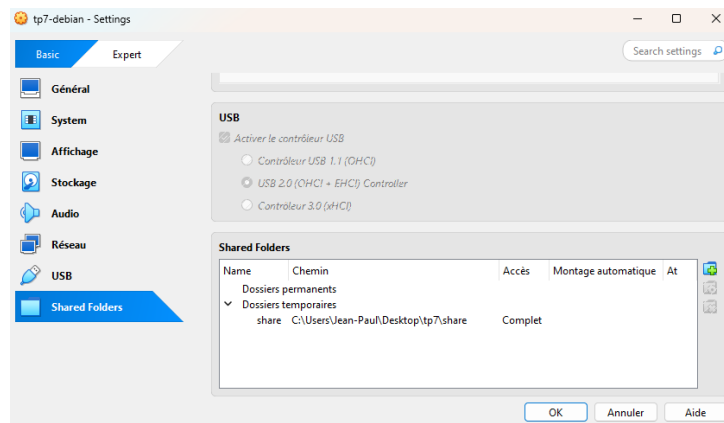


Figure 29

On retourne sur la machine "tp7-debian". On tape la commande "mkdir ~/mnt" pour créer un nouveau dossier dans le dossier personnel de l'utilisateur "user". On tape la commande : "sudo mount -t vboxsf share ~/mnt" pour lier le dossier "mnt" qui se trouve dans la machine "tp6-debian" au dossier partagé "share" venant de la machine physique. Puis, on copie l'archive "tp7-jean-paul-melisse.tar.xz" dans le dossier "mnt" en tapant la commande "sudo cp tp7-jean-paul-melisse.tar.xz mnt/". On retourne sur la machine physique et on regarde le dossier "share". On voit, comme dans la Figure 30, que l'archive est bien sur la machine physique. On peut l'exporter pour une utilisation ultérieure.

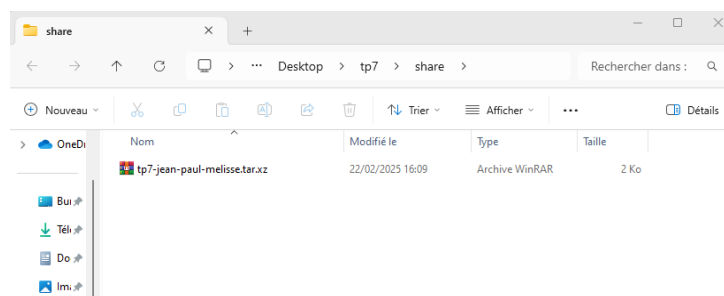


Figure 30

21 Créer la machine virtuelle tp7-windows-server

Maintenant qu'on a vu comment utiliser Powershell sur Debian, on va de même sur Windows serveur. On retourne sur VirtualBox. On clique sur "Nouveau" pour créer une nouvelle machine virtuelle. On met "tp7-windows-server" comme nom de la machine virtuelle. On choisit son emplacement de sauvegarde et on importe le CD contenant le système d'exploitation de Windows Serveur. En détectant le CD, le type "Microsoft Windows" et la version "Windows 2022 (64 bits)" sont automatiquement mis. On coche la case "Skip Unattended Installation" et on clique sur "Suivant". On lui donne une mémoire vive (RAM) de 2048 Mo et un processeur, et on clique sur "Suivant". On lui donne un disque dur de 50 Go et on clique sur "Suivant". On regarde bien le récapitulatif et on clique sur "Finish" (Voir la Figure 31).

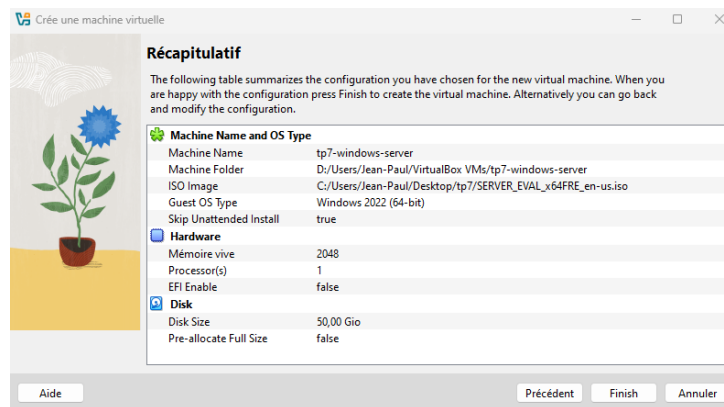


Figure 31

Maintenant que la nouvelle machine est créée, il faut la configurer. On clique sur "Configurations". On clique sur "Réseau" et on change le mode d'accès réseau à "Réseau interne", puis on clique sur "OK". Maintenant qu'on a configuré la machine virtuelle, on clique sur "Démarrer" pour l'allumer.

22 Installer Windows Server sur tp7-windows-server

Après avoir démarré la machine virtuelle, on clique sur n'importe quelle touche pour passer sur le bootable du CD. On choisit pour le paramètre "Language to install: English (United States)" et on choisit pour l'option "Time and currency format: French (France)". Pour le clavier (Keyboard or input method), on choisit "French (Legacy, AZERTY)" et on clique sur "Next". On clique sur le bouton "Install now". On choisit l'option "Windows Server 2022 Standard Evaluation (Desktop Experience)" et on clique sur Next. On coche la case "I agree everything will be deleted", et on clique sur "Next". On choisit l'option "custom: Install Microsoft server OS only". On choisit "Drive 0 Unallocated Space" pour l'emplacement de l'installation de Windows et on clique sur "Next" pour lancer l'installation.

Après s'est redémarré, on met un mot de passe pour l'utilisateur "Administrator" qui sera "Azerty123!". On met le mot de passe 2 fois, puis on clique sur "Finish" (cf: Figure 32).

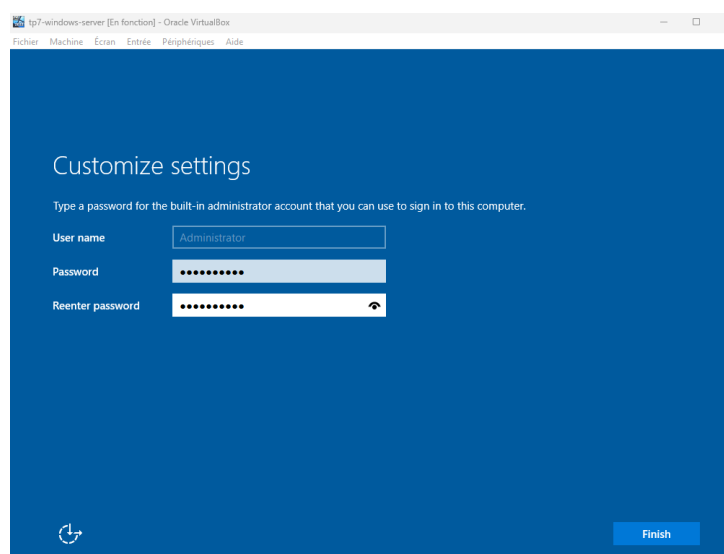


Figure 32

On appuie sur "Ctrl+Alt+Suppr" et on met le mot de passe pour se connecter en tant qu'Administrator sur le tp7-windows-server.

23 Ajouter la fonctionnalité ADDS

Maintenant qu'on a installé Windows Serveur, on va lui ajouter la fonctionnalité ADDS pour qu'il devienne le contrôleur de domaine. Sur la machine "tp7-windows-server", on clique sur "Manage", puis on clique sur "Add Roles and Features". On coche la case "Skip this page by default" et on clique sur "Next". On choisit l'option "Role-based" pour ajouter une fonctionnalités au serveur et on clique sur "Next". On coche la case "Select a server from the server pool" et on sélectionne notre serveur avant de cliquer sur "Next". On coche la case "Active Directory Domain Services" et on clique sur "Add Features" pour ajouter la fonctionnalités ADDS. Puis on clique sur "Next" autant de fois que nécessaire. On clique sur "Install" pour installer cette fonctionnalité (cf: Figure 33).

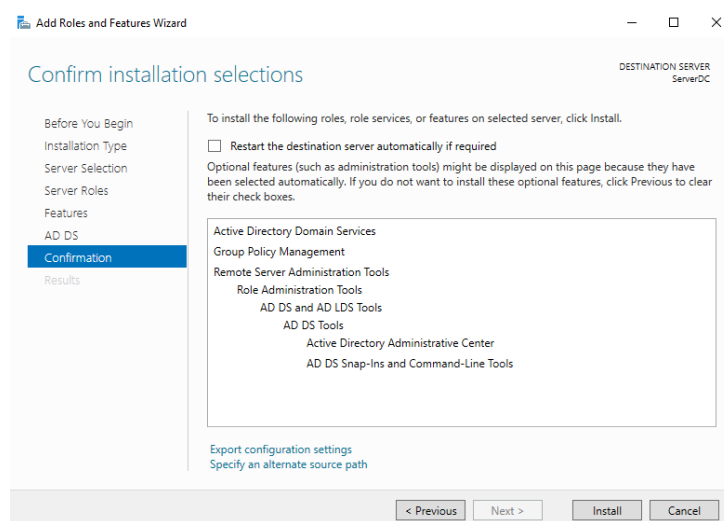


Figure 33

Après l'installation, on clique sur "Close" pour fermer la fenêtre.

24 Ajouter le domaine

Après avoir ajouté le rôle ADDS à notre serveur, on va le faire passer en contrôleur de domaine, créant ainsi un nouveau domaine. On clique sur le drapeau "Notifications" du "Server Manager" et on clique sur "Promote this server to a domain controller". On coche la case "Add a new forest" pour ajouter un nouveau racine qu'on nomme "afpa.local" (Voir la Figure 34). Puis on clique sur "Next".

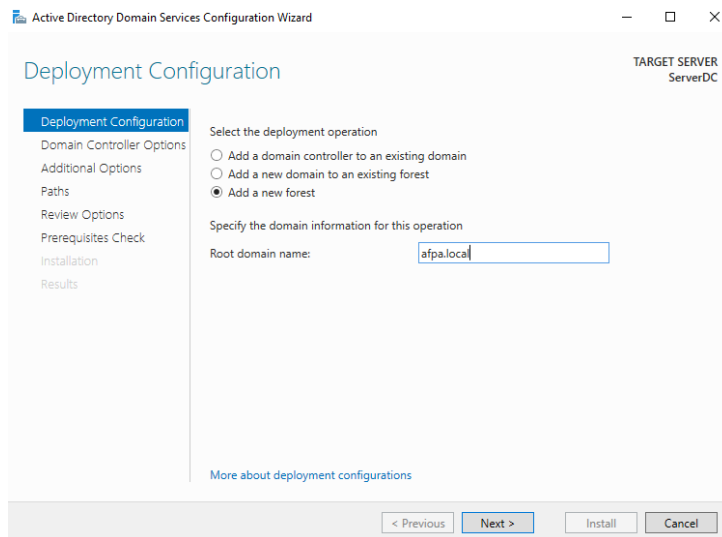


Figure 34

On met 2x un mot de passe pour le controleur de domaine (ici mdp : Azerty123!), puis on clique sur "Next" autant de fois que nécessaire. On clique sur "Install" pour installer les configurations pour que le serveur devient un controleur de domaine. Après l'installation, le serveur va se redémarrer.

25 Ajouter un utilisateur via le Server Manager

Après le redémarrage du serveur, on se connecte sur le nouveau avec les identifiant de l'utilisateur "AFPA\Administrator". On va ajouter un nouvel utilisateur à partir du Server Manager. Sur la fenêtre du Server Manger, on clique sur "Tools", puis sur "Active Directory Users and Computers". On clique sur notre domaine "afpa.local". On fait un clic-droit sur "Users > New > User" pour créer un nouvel utilisateur. On crée l'utilisateur "Abc" et on clique sur "Next" (cf: Figure 35).

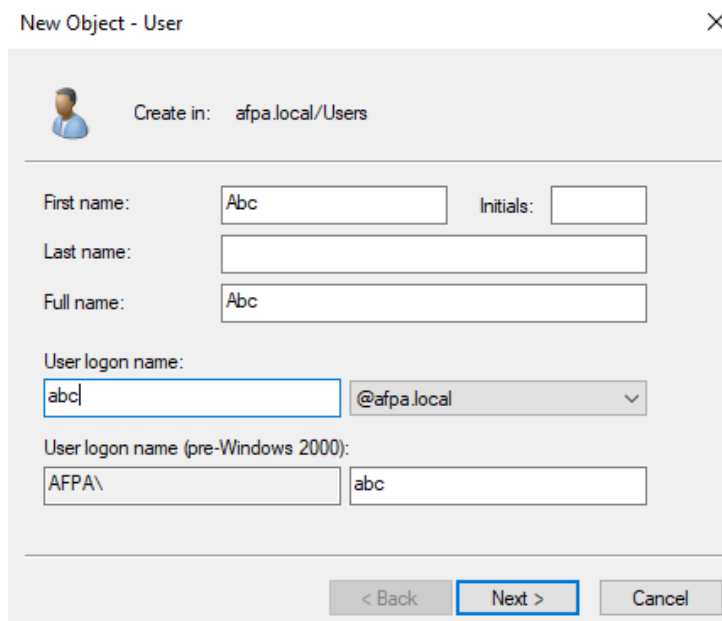


Figure 35

On lui donne un mot de passe temporaire (ici mdp : Azerty123) et on clique sur "Next". Puis, on clique

sur "Finish" pour finaliser la création du nouvel utilisateur.

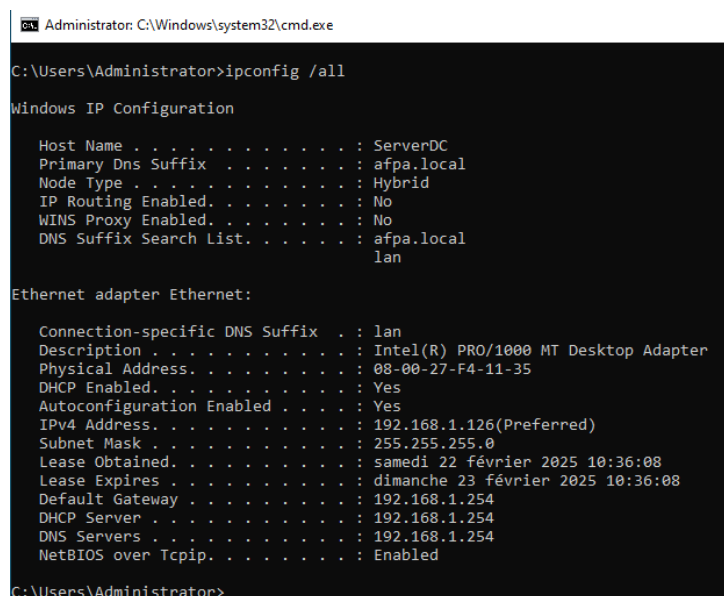
26 Configurer le réseau de tp7-windows-server

La prochaine étape est d'installer Powershell. Pour cela, il faut que notre server soit connecté à l'internet. On retourne sur VirtualBox. On clique sur "tp7-windows-server" et sur "Configurations". On clique sur "Réseau" et on change le mode d'accès réseau à "Accès par pont", puis on clique sur "OK". Le mode "Accès par pont" va permettre à la machine virtuelle de récupérer une adresse IP directement depuis un serveur DHCP (par exemple : notre box). Cela permet de connecter la machine virtuelle à l'Internet.

On retourne sur notre machine "tp7-windows-server" et on appuie sur Win + R. Ce qui lance le programme "RUN". Dans le programme RUN, on tape "ncpa.cpl" et on clique sur "OK". Ce qui va ouvrir une nouvelle fenêtre avec toutes les cartes réseau. On fait un clic-droit sur la carte réseau qu'on utilise (ici "Ethernet") et on clique sur "Properties". On double-clique sur "Internet Protocol Version 4 (TCP/IPv4)". On coche la case "Obtain an IP address automatically" et la case "Obtain DNS server address automatically". Puis on clique sur "OK" deux fois.

On ouvre une invite de commande : On appuie sur "Win + R", ce qui lance le programme "RUN". Dans le programme RUN, on tape "cmd" et on clique sur "OK". Sur l'invite de commande, on tape la commande "ipconfig /release", puis "ipconfig /renew" pour redémarrer la configuration réseau.

On tape la commande "ipconfig /all" pour vérifier si la configuration réseau est bonne et c'est le cas (Voir la Figure 36).



```
Administrator: C:\Windows\system32\cmd.exe

C:\Users\Administrator>ipconfig /all

Windows IP Configuration

Host Name . . . . . : ServerDC
Primary Dns Suffix . . . . . : afpa.local
Node Type . . . . . : Hybrid
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No
DNS Suffix Search List. . . . . : afpa.local
                                  lan

Ethernet adapter Ethernet:

Connection-specific DNS Suffix . : lan
Description . . . . . : Intel(R) PRO/1000 MT Desktop Adapter
Physical Address. . . . . : 08-00-27-F4-11-35
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
IPv4 Address. . . . . : 192.168.1.126(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Lease Obtained. . . . . : samedi 22 février 2025 10:36:08
Lease Expires . . . . . : dimanche 23 février 2025 10:36:08
Default Gateway . . . . . : 192.168.1.254
DHCP Server . . . . . : 192.168.1.254
DNS Servers . . . . . : 192.168.1.254
NetBIOS over Tcpip. . . . . : Enabled

C:\Users\Administrator>
```

Figure 36

27 Télécharger PowerShell sous Windows Server

Sur notre serveur, on ouvre le navigateur Edge. On tape sur la barre de recherche le lien <https://github.com/PowerShell/PowerShell> et on tape sur la touche "Entrée". On clique sur "Releases" et on choisit la version "7.4.6 de PowerShell". Sur l'onglet 7.4.6, on clique sur "Assets" et on clique sur "PowerShell-7.4.6-win-x64.msi" pour télécharger ce package. On le sauvegarde dans un nouveau dossier "tp7" (cf: Figure 37).

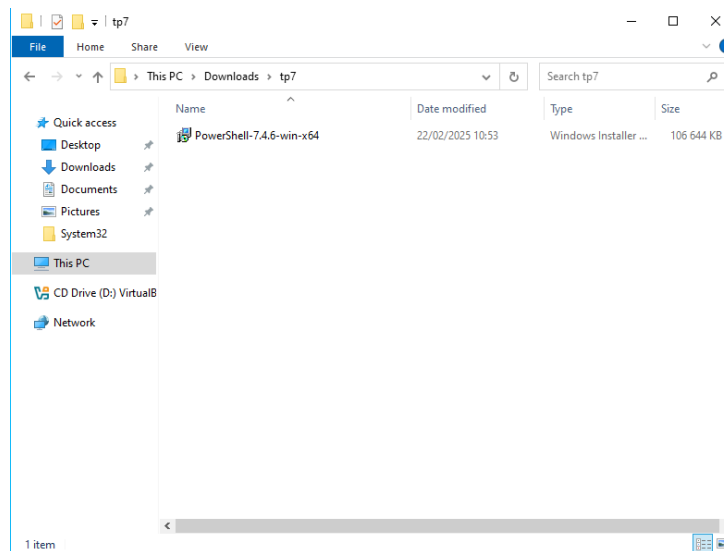


Figure 37

28 Installer PowerShell sur tp7-windows-server

Après avoir télécharger le package powershell, on double-clique sur le package "PowerShell-7.4.6-win-x64.msi" pour l'exécuter. On clique sur "Next" autant de fois que nécessaire, puis on clique sur "Install" pour installer powershell sur Windows Serveur (Voir la Figure 38).

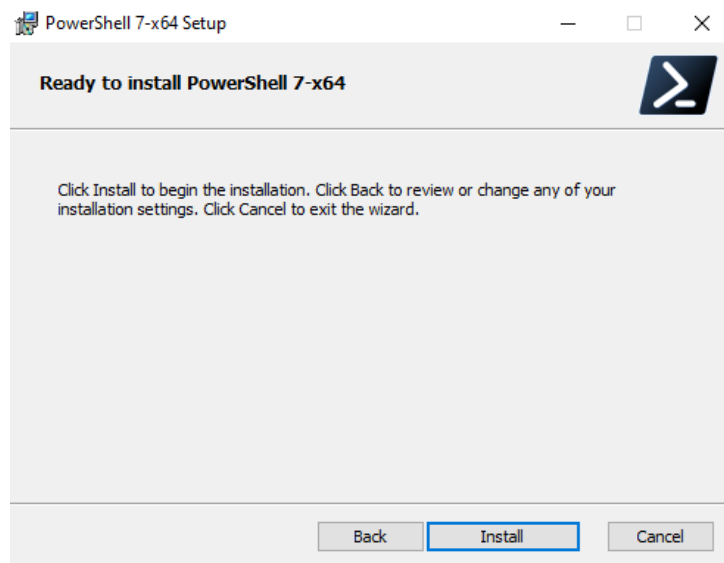


Figure 38

Après l'installation de powershell, on clique sur "Finish" pour finaliser la configuration de powershell et fermer cette fenêtre.

29 Ajouter un utilisateur à l'AD via PowerShell

On va cette fois-ci créer un nouvel utilisateur en passant par powershell. On appuie sur Win + R. Ce qui lance le programme "RUN". Dans le programme RUN, on tape "pwsh" et on clique sur "OK". Sur

powershell, on tape la commande : 'New-ADUser -SamAccountName "Def"' pour ajouter l'utilisateur "Def" à l'AD (Voir la Figure 39).

```
Administrator: C:\Program Files\PowerShell\7\pwsh.exe
PowerShell 7.4.6
PS C:\Users\Administrator> New-ADUser -SamAccountName "Def"

cmdlet New-ADUser at command pipeline position 1
Supply values for the following parameters:
Name: Def
PS C:\Users\Administrator> _
```

Figure 39

30 Tester l'utilisateur ajouté à l'AD

Après la création de l'utilisateur "Def" avec powershell, on va le vérifier dans le Server Manager. On ouvre le Server Manager et on clique sur "Tools", puis sur "Active Directory Users and Computers". On clique sur notre domaine "afpa.local", puis sur "Users". On voit, comme sur la Figure 40, que l'utilisateur "Def" a bien été créé.

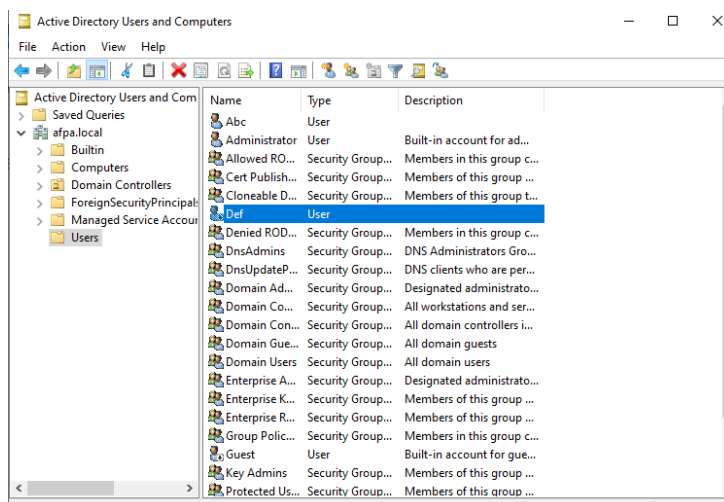


Figure 40