Version : **2024.01**

Dernière mise-à-jour : 2024/12/03 14:37

LDF904 - Utilisation d'Ansible avec Docker et Windows

Contenu du Module

LAB #1 - Ansible et Docker

1.1 - Présentation de Docker

Docker est une application de virtualisation légère qui utilise des **images** et des **conteneurs**.

Une **image** est un paquet exécutable contenant tout ce qu'il est nécessaire afin d'exécuter un logiciel donné, incluant :

- le code
- un runtime
- des bibliothèques,
- des variables d'environnement
- des fichiers de configuration

Un **conteneur** est une instance de l'image en cours d'exécution en mémoire. Elle est isolée de l'environnement de l'hôte par défaut mais peut accéder à des fichiers et de ports de l'hôte selon la configuration.

Les conteneurs exécutent des applications nativement en utilisant le noyau de la machine hôte. De ce fait les performances d'un conteneur sont supérieures à celles d'une machine virtuelle qui doit passer par un hyperviseur pour accéder aux ressources de la machine hôte :

Docker existe en deux versions **Docker-CE** (Docker Community Edition) et **Docker-EE** (Docker Enterprise Edition). Pour consulter les différences entre les deux versions, consultez le lien **https://docs.docker.com/engine/installation/**.

1.2 - Installer docker

Docker n'est pas dans le dépôts de Debian. Afin de l'installer il convient d'ajouter le dépôt de docker. Premièrement, il est nécessaire d'installer les paquets permettant à Debian d'utiliser un dépôt en https :

```
root@debian11:~# apt-get update
...
root@debian11:~# apt-get install apt-transport-https ca-certificates curl gnupg2 software-properties-common
Reading package lists... Done
```

```
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20210119).
gnupg2 is already the newest version (2.2.27-2+deb11u2).
The following packages were automatically installed and are no longer required:
  libopengl0 linux-headers-5.10.0-15-amd64 linux-headers-5.10.0-15-common
Use 'apt autoremove' to remove them.
The following additional packages will be installed:
  python3-distro-info python3-software-properties unattended-upgrades
Suggested packages:
  bsd-mailx default-mta | mail-transport-agent needrestart powermgmt-base
The following NEW packages will be installed:
  apt-transport-https curl python3-distro-info python3-software-properties
  software-properties-common unattended-upgrades
0 upgraded, 6 newly installed, 0 to remove and 0 not upgraded.
Need to get 661 kB of archives.
After this operation, 1,567 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

Téléchargez la clef GPG officielle de docker :

```
root@debian11:~# curl -fsSL https://download.docker.com/linux/debian/gpg | apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
```

Vérifiez que l'ID de la clef est **9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88** :

```
root@debian11:~# apt-key fingerprint 0EBFCD88
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
pub   rsa4096 2017-02-22 [SCEA]
      9DC8 5822 9FC7 DD38 854A  E2D8 8D81 803C 0EBF CD88
uid           [ unknown] Docker Release (CE deb) <docker@docker.com>
sub   rsa4096 2017-02-22 [S]
```

Ajoutez le dépôt **stable** de docker :

```
root@debian11:~# add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/debian $(lsb_release -cs) stable"
```

> **Important** - Notez que la commande **lsb_release -cs** retourne le nom de la distribution Debian, à savoir dans ce cas **stretch**.

Installez maintenant le paquet **docker-ce** :

```
root@debian11docker:~# apt-get update
...
root@debian11:~# apt-get install docker-ce
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libopengl0 linux-headers-5.10.0-15-amd64 linux-headers-5.10.0-15-common
Use 'apt autoremove' to remove them.
The following additional packages will be installed:
  containerd.io docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras
  docker-compose-plugin git git-man liberror-perl libslirp0 pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite git-daemon-run | git-daemon-sysvinit
  git-doc git-el git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli
  docker-ce-rootless-extras docker-compose-plugin git git-man liberror-perl
  libslirp0 pigz slirp4netns
0 upgraded, 12 newly installed, 0 to remove and 0 not upgraded.
Need to get 121 MB of archives.
```

```
After this operation, 452 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

Dernièrement, vérifiez la version de Docker client et serveur :

```
root@debian11:~# docker version
Client: Docker Engine - Community
 Version:           24.0.5
 API version:       1.43
 Go version:        go1.20.6
 Git commit:        ced0996
 Built:             Fri Jul 21 20:35:45 2023
 OS/Arch:           linux/amd64
 Context:           default

Server: Docker Engine - Community
 Engine:
  Version:          24.0.6
  API version:      1.43 (minimum version 1.12)
  Go version:       go1.20.7
  Git commit:       1a79695
  Built:            Mon Sep  4 12:32:16 2023
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.6.22
  GitCommit:        8165feabfdfe38c65b599c4993d227328c231fca
 runc:
  Version:          1.1.8
  GitCommit:        v1.1.8-0-g82f18fe
 docker-init:
  Version:          0.19.0
  GitCommit:        de40ad0
```

Important - Notez que le paquet docker-ce a besoin des paquets **containerd.io** et **docker-ce-cli**. Notez aussi que la procédure ci-dessus installe la version la plus récente de Docker.

Dans le cas où vous souhaitez installer une version différente, il convient d'abord de constater les versions disponibles :

```
root@debian11:~# apt-cache madison docker-ce
 docker-ce | 5:24.0.6-1~debian.11~bullseye | https://download.docker.com/linux/debian bullseye/stable amd64
Packages
 docker-ce | 5:24.0.5-1~debian.11~bullseye | https://download.docker.com/linux/debian bullseye/stable amd64
Packages
 docker-ce | 5:24.0.4-1~debian.11~bullseye | https://download.docker.com/linux/debian bullseye/stable amd64
Packages
 docker-ce | 5:24.0.3-1~debian.11~bullseye | https://download.docker.com/linux/debian bullseye/stable amd64
Packages
 docker-ce | 5:24.0.2-1~debian.11~bullseye | https://download.docker.com/linux/debian bullseye/stable amd64
Packages
 docker-ce | 5:24.0.1-1~debian.11~bullseye | https://download.docker.com/linux/debian bullseye/stable amd64
Packages
 docker-ce | 5:24.0.0-1~debian.11~bullseye | https://download.docker.com/linux/debian bullseye/stable amd64
Packages
 docker-ce | 5:23.0.6-1~debian.11~bullseye | https://download.docker.com/linux/debian bullseye/stable amd64
Packages
 docker-ce | 5:23.0.5-1~debian.11~bullseye | https://download.docker.com/linux/debian bullseye/stable amd64
Packages
 docker-ce | 5:23.0.4-1~debian.11~bullseye | https://download.docker.com/linux/debian bullseye/stable amd64
Packages
 docker-ce | 5:23.0.3-1~debian.11~bullseye | https://download.docker.com/linux/debian bullseye/stable amd64
Packages
 docker-ce | 5:23.0.2-1~debian.11~bullseye | https://download.docker.com/linux/debian bullseye/stable amd64
Packages
```

```
 docker-ce | 5:23.0.1-1~debian.11~bullseye | https://download.docker.com/linux/debian bullseye/stable amd64
Packages
 docker-ce | 5:23.0.0-1~debian.11~bullseye | https://download.docker.com/linux/debian bullseye/stable amd64
Packages
 docker-ce | 5:20.10.24~3-0~debian-bullseye | https://download.docker.com/linux/debian bullseye/stable amd64
Packages
 docker-ce | 5:20.10.23~3-0~debian-bullseye | https://download.docker.com/linux/debian bullseye/stable amd64
Packages
 docker-ce | 5:20.10.22~3-0~debian-bullseye | https://download.docker.com/linux/debian bullseye/stable amd64
Packages
 docker-ce | 5:20.10.21~3-0~debian-bullseye | https://download.docker.com/linux/debian bullseye/stable amd64
Packages
 docker-ce | 5:20.10.20~3-0~debian-bullseye | https://download.docker.com/linux/debian bullseye/stable amd64
Packages
 docker-ce | 5:20.10.19~3-0~debian-bullseye | https://download.docker.com/linux/debian bullseye/stable amd64
Packages
 docker-ce | 5:20.10.18~3-0~debian-bullseye | https://download.docker.com/linux/debian bullseye/stable amd64
Packages
 docker-ce | 5:20.10.17~3-0~debian-bullseye | https://download.docker.com/linux/debian bullseye/stable amd64
Packages
 docker-ce | 5:20.10.16~3-0~debian-bullseye | https://download.docker.com/linux/debian bullseye/stable amd64
Packages
 docker-ce | 5:20.10.15~3-0~debian-bullseye | https://download.docker.com/linux/debian bullseye/stable amd64
Packages
 docker-ce | 5:20.10.14~3-0~debian-bullseye | https://download.docker.com/linux/debian bullseye/stable amd64
Packages
 docker-ce | 5:20.10.13~3-0~debian-bullseye | https://download.docker.com/linux/debian bullseye/stable amd64
Packages
 docker-ce | 5:20.10.12~3-0~debian-bullseye | https://download.docker.com/linux/debian bullseye/stable amd64
Packages
 docker-ce | 5:20.10.11~3-0~debian-bullseye | https://download.docker.com/linux/debian bullseye/stable amd64
Packages
 docker-ce | 5:20.10.10~3-0~debian-bullseye | https://download.docker.com/linux/debian bullseye/stable amd64
Packages
```

```
 docker-ce | 5:20.10.9~3-0~debian-bullseye | https://download.docker.com/linux/debian bullseye/stable amd64
Packages
 docker-ce | 5:20.10.8~3-0~debian-bullseye | https://download.docker.com/linux/debian bullseye/stable amd64
Packages
 docker-ce | 5:20.10.7~3-0~debian-bullseye | https://download.docker.com/linux/debian bullseye/stable amd64
Packages
 docker-ce | 5:20.10.6~3-0~debian-bullseye | https://download.docker.com/linux/debian bullseye/stable amd64
Packages
```

Dans le cas où vous souhaiteriez installer la version **24.0.1** de Docker, la commande d'installation deviendrait :

```
# apt-get install docker-ce=5:24.0.1-1~debian.11~bullseye docker-ce-cli=5:24.0.1-1~debian.11~bullseye
containerd.io
```

Si vous préférez utiliser le script d'installation de Docker, il convient d'abord de le télécharger :

> **Important** - Notez que ces scripts ne doivent pas être utilisés dans un environnement de production.

```
root@debian11:~# curl -fsSL https://get.docker.com -o get-docker.sh

root@debian11:~# ls
get-docker.sh
```

Ensuite, il convient d'exécuter le script :

```
root@debian11:~# chmod +x get-docker.sh

root@debian11:~# ./get-docker.sh
# Executing docker install script, commit: c2de0811708b6d9015ed1a2c80f02c9b70c8ce7b
Warning: the "docker" command appears to already exist on this system.
```

```
If you already have Docker installed, this script can cause trouble, which is
why we're displaying this warning and provide the opportunity to cancel the
installation.

If you installed the current Docker package using this script and are using it
again to update Docker, you can safely ignore this message.

You may press Ctrl+C now to abort this script.
+ sleep 20
^C
```

Important - Notez l'utilisation de **^C** pour ne PAS continuer l'exécution du script.

Démarrez un conteneur de l'image hello-world :

```
root@debian11:~# docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
719385e32844: Pull complete
Digest: sha256:dcba6daec718f547568c562956fa47e1b03673dd010fe6ee58ca806767031d1c
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
```

```
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

Important - Notez que si l'image servant à générer le conteneur n'est pas présente sur le système hôte, celle-ci est téléchargée automatiquement depuis un dépôt (par défaut le dépôt **docker.io**) en utilisant la commande **docker pull**.

Lancez maintenant un conteneur postgresql :

```
root@debian11:~# docker run -d --name postgresql centos sleep infinity
7d948c33b2f42e1823ae22353815ea674695fa1dc9e6bd70acc49a80651845c5
```

Vérifiez l'état du conteneur :

```
root@debian11:~# docker ps
CONTAINER ID   IMAGE     COMMAND            CREATED         STATUS         PORTS     NAMES
7d948c33b2f4   centos    "sleep infinity"   2 minutes ago   Up 2 minutes             postgresql
```

Ajoutez **trainee** au groupe **docker** afin qu'il puisse gérer les conteneurs puis redevenir l'utilisateur trainee :

```
root@debian11:/home/trainee/.ansible/roles# usermod -aG docker trainee
root@debian11:/home/trainee/.ansible/roles# groups trainee
trainee : trainee cdrom floppy audio dip video plugdev netdev lpadmin scanner vboxusers docker
```

Redevenez l'utilisateur **trainee** :

```
root@debian11:/home/trainee/.ansible/roles# exit
logout
trainee@debian11:~/.ansible/roles$ groups
trainee cdrom floppy audio dip video plugdev netdev lpadmin scanner vboxusers
trainee@debian11:~/.ansible/roles$ newgrp docker
trainee@debian11:~/.ansible/roles$ groups
docker cdrom floppy audio dip video plugdev netdev lpadmin scanner vboxusers trainee
trainee@debian11:~/.ansible/roles$
```

1.3 - La Connexion à Docker

Créez maintenant le Rôle **docker** :

```
trainee@debian11:~/.ansible/roles$ mkdir docker
```

Modifiez ensuite le fichier playbook.yaml :

```
trainee@debian11:~/.ansible/roles$ vi playbook.yaml
trainee@debian11:~/.ansible/roles$ cat playbook.yaml
---
- hosts: all
  tasks:
    - name: message
      debug: msg="Conteneur - {{ inventory_hostname }} sous {{ansible_distribution}}"
```

Modifiez le fichier **inventory** :

```
trainee@debian11:~/.ansible/roles$ cp inventory inventory.old
trainee@debian11:~/.ansible/roles$ vi inventory
trainee@debian11:~/.ansible/roles$ cat inventory
postgresql ansible_connection=docker
```

Supprimez le fichier **/home/trainee/.ansible/roles/group_vars/all.yaml** :

```
trainee@debian11:~/.ansible/roles$ rm -rf /home/trainee/.ansible/roles/group_vars/all.yaml
```

Exécutez la commande **ansible-playbook** :

```
trainee@debian11:~/.ansible/roles$ ansible-playbook -i inventory playbook.yaml

PLAY [all]
*********************************************************************************************************
*************************************************************************

TASK [Gathering Facts]
*********************************************************************************************************
*********************************************************************
ok: [postgresql]

TASK [message]
*********************************************************************************************************
*************************************************************************
ok: [postgresql] => {
    "msg": "Conteneur - postgresql sous CentOS"
}

PLAY RECAP
*********************************************************************************************************
*************************************************************************
postgresql                 : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
```

```
ignored=0
```

LAB #2 - Préparer Windows 10

Ansible est capable de travailler avec les versions de Windows™ suivantes :

- Windows™ 7,
- Windows™ 8.1,
- Windowst™ 10,
- Windowst™ 11,
- Windows™ Server 2008,
- Windows™ Server 2008 R2,
- Wondows™ Server 2012,
- Windows™ Server 2012 R2,
- Windows™ Server 2016,
- Windows™ Server 2019,
- Windows™ Server 2022.

Ansible nécessite sur chaque hôte :

- PowerShell 3.0 ou supérieur
- Au moins .NET 4.0.

2.1 - Mettre à Jour PowerShell et .NET

Afin de mettre à jour les versions de PowerShell et .NET, Ansible fournit un script appelé **Upgrade-PowerShell.ps1**. Exécutez PowerShell en tant qu'Administrateur :

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.


```

```
Testez le nouveau système multiplateforme PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> $url =
"https://raw.githubusercontent.com/jborean93/ansible-windows/master/scripts/Upgrade-PowerShell.ps1"
PS C:\Windows\system32> $file = "$env:temp\Upgrade-PowerShell.ps1"
PS C:\Windows\system32> $username = "trainee"
PS C:\Windows\system32> $password = "a39dae707d"
PS C:\Windows\system32> (New-Object -TypeName System.Net.WebClient).DownloadFile($url, $file)
PS C:\Windows\system32> Set-ExecutionPolicy -ExecutionPolicy Unrestricted -Force
PS C:\Windows\system32> &$file -Version 5.1 -Username $username -Password $password -Verbose
COMMENTAIRES : 2019-10-24T09:36:23 - INFO - starting script
COMMENTAIRES : 2019-10-24T09:36:23 - INFO - current and target PS version are the same, no action is required
COMMENTAIRES : 2019-10-24T09:36:23 - INFO - clearing auto logon registry properties
```

Au cas où, vous pouvez enlever l'auto-login de l'administrateur et remettre la politique d'exécution à **Restricted** avec les commandes suivantes :

```
PS C:\Windows\system32> Set-ExecutionPolicy -ExecutionPolicy Restricted -Force
PS C:\Windows\system32> $reg_winlogon_path = "HKLM:\Software\Microsoft\Windows NT\CurrentVersion\Winlogon"
PS C:\Windows\system32> Set-ItemProperty -Path $reg_winlogon_path -Name AutoAdminLogon -Value 0
PS C:\Windows\system32> Remove-ItemProperty -Path $reg_winlogon_path -Name DefaultUserName -ErrorAction
SilentlyContinue
PS C:\Windows\system32> Remove-ItemProperty -Path $reg_winlogon_path -Name DefaultPassword -ErrorAction
SilentlyContinue
PS C:\Windows\system32>
```

2.2 - Configurer WinRM

Pour mettre en place un **listener** WinRM en HTTP et un **listener** WinRM en HTTPS, créer un certificat auto-signé et activer l'authentification de base sur le service WinRM, utilisez le script **ConfigureRemotingForAnsible.ps1** :

```
PS C:\Windows\system32> $url =
"https://raw.githubusercontent.com/AlbanAndrieu/ansible-windows/master/files/ConfigureRemotingForAnsible.ps1"
PS C:\Windows\system32> $file = "$env:temp\ConfigureRemotingForAnsible.ps1"
```

```
PS C:\Windows\system32> (New-Object -TypeName System.Net.WebClient).DownloadFile($url, $file)
PS C:\Windows\system32> powershell.exe -ExecutionPolicy ByPass -File $file
Self-signed SSL certificate generated; thumbprint: 17502EEEAC259F4C76D0F199A4B803E94E980CAD



wxf                 : http://schemas.xmlsoap.org/ws/2004/09/transfer
a                   : http://schemas.xmlsoap.org/ws/2004/08/addressing
w                   : http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd
lang                : fr-FR
Address             : http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
ReferenceParameters : ReferenceParameters

Ok.



PS C:\Windows\system32>
```

2.3 - Consulter les Informations sur WinRM

Pour voir la liste des **listeners** ainsi créés, utilisez la commande suivante :

```
PS C:\Windows\system32> winrm enumerate winrm/config/Listener
Listener
    Address = *
    Transport = HTTP
    Port = 5985
    Hostname
    Enabled = true
    URLPrefix = wsman
    CertificateThumbprint
    ListeningOn = 10.0.2.58, 10.29.0.34, 127.0.0.1, ::1
```

```
Listener
    Address = *
    Transport = HTTPS
    Port = 5986
    Hostname = DESKTOP-1OV5NMP
    Enabled = true
    URLPrefix = wsman
    CertificateThumbprint = 17502EEEAC259F4C76D0F199A4B803E94E980CAD
    ListeningOn = 10.0.2.58, 10.29.0.34, 127.0.0.1, ::1

PS C:\Windows\system32>
```

Pour obtenir les informations concernant le certificat, exécutez les commandes suivantes en remplacant le contenu de $thumbprint avec la valeur que vous avez obtenue :

```
PS C:\Windows\system32> $thumbprint = "17502EEEAC259F4C76D0F199A4B803E94E980CAD"
PS C:\Windows\system32> Get-ChildItem -Path cert:\LocalMachine\My -Recurse | Where-Object { $_.Thumbprint -eq
$thumbprint } | Select-Object *


PSPath                   :
Microsoft.PowerShell.Security\Certificate::LocalMachine\My\17502EEEAC259F4C76D0F199A4B803E94
                           E980CAD
PSParentPath             : Microsoft.PowerShell.Security\Certificate::LocalMachine\My
PSChildName              : 17502EEEAC259F4C76D0F199A4B803E94E980CAD
PSDrive                  : Cert
PSProvider               : Microsoft.PowerShell.Security\Certificate
PSIsContainer            : False
EnhancedKeyUsageList     : {Authentification du serveur (1.3.6.1.5.5.7.3.1)}
DnsNameList              : {DESKTOP-1OV5NMP, DESKTOP-1OV5NMP}
SendAsTrustedIssuer      : False
EnrollmentPolicyEndPoint : Microsoft.CertificateServices.Commands.EnrollmentEndPointProperty
EnrollmentServerEndPoint : Microsoft.CertificateServices.Commands.EnrollmentEndPointProperty
PolicyId                 :
```

```
Archived                : False
Extensions              : {System.Security.Cryptography.Oid, System.Security.Cryptography.Oid,
                          System.Security.Cryptography.Oid, System.Security.Cryptography.Oid}
FriendlyName            :
IssuerName              : System.Security.Cryptography.X509Certificates.X500DistinguishedName
NotAfter                : 22/10/2022 11:45:21
NotBefore               : 23/10/2019 11:45:21
HasPrivateKey           : True
PrivateKey              : System.Security.Cryptography.RSACryptoServiceProvider
PublicKey               : System.Security.Cryptography.X509Certificates.PublicKey
RawData                 : {48, 130, 5, 49...}
SerialNumber            : 73F2EADFBABB86B7438B30B157A1C3FB
SubjectName             : System.Security.Cryptography.X509Certificates.X500DistinguishedName
SignatureAlgorithm      : System.Security.Cryptography.Oid
Thumbprint              : 17502EEEAC259F4C76D0F199A4B803E94E980CAD
Version                 : 3
Handle                  : 2829331536176
Issuer                  : CN=DESKTOP-1OV5NMP
Subject                 : CN=DESKTOP-1OV5NMP



PS C:\Windows\system32>
```

Dernièrement pour obtenir des informations concernant le service, exécutez les deux commandes suivantes :

```
PS C:\Windows\system32> winrm get winrm/config/Service
Service
    RootSDDL = O:NSG:BAD:P(A;;GA;;;BA)(A;;GR;;;IU)S:P(AU;FA;GA;;;WD)(AU;SA;GXGW;;;WD)
    MaxConcurrentOperations = 4294967295
    MaxConcurrentOperationsPerUser = 1500
    EnumerationTimeoutms = 240000
    MaxConnections = 300
    MaxPacketRetrievalTimeSeconds = 120
```

```
    AllowUnencrypted = false
    Auth
        Basic = true
        Kerberos = true
        Negotiate = true
        Certificate = false
        CredSSP = false
        CbtHardeningLevel = Relaxed
    DefaultPorts
        HTTP = 5985
        HTTPS = 5986
    IPv4Filter = *
    IPv6Filter = *
    EnableCompatibilityHttpListener = false
    EnableCompatibilityHttpsListener = false
    CertificateThumbprint
    AllowRemoteAccess = true

PS C:\Windows\system32> winrm get winrm/config/Winrs
Winrs
    AllowRemoteShellAccess = true
    IdleTimeout = 7200000
    MaxConcurrentUsers = 2147483647
    MaxShellRunTime = 2147483647
    MaxProcessesPerShell = 2147483647
    MaxMemoryPerShellMB = 2147483647
    MaxShellsPerUser = 2147483647

PS C:\Windows\system32>
```

2.4 - Créer un Utilisateur Local pour Ansible

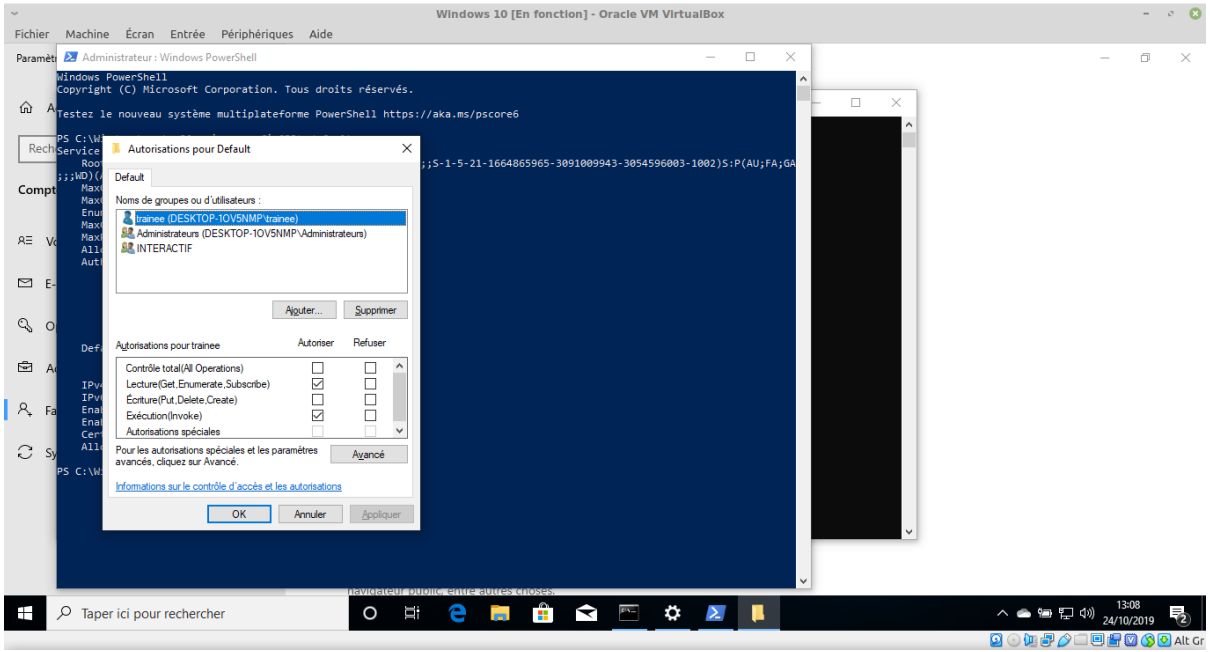
Retournez à l'accueil de Guacamole et connectez-vous à la machine virtuelle **Windows_10.0.2.58_VNC**.

Lancez PowerShell en tant que l'administrateur et exécutez ensuite la commande suivante :

```
PS C:\Windows\system32> winrm configSDDL default
Service
    RootSDDL =
O:NSG:BAD:P(A;;GA;;;BA)(A;;GR;;;IU)(A;;GXGR;;;S-1-5-21-1664865965-3091009943-3054596003-1002)S:P(AU;FA;GA;;;WD)(A
U;SA;GXGW;;;WD)
    MaxConcurrentOperations = 4294967295
    MaxConcurrentOperationsPerUser = 1500
    EnumerationTimeoutms = 240000
    MaxConnections = 300
    MaxPacketRetrievalTimeSeconds = 120
    AllowUnencrypted = false
    Auth
        Basic = true
        Kerberos = true
        Negotiate = true
        Certificate = false
        CredSSP = false
        CbtHardeningLevel = Relaxed
    DefaultPorts
        HTTP = 5985
        HTTPS = 5986
    IPv4Filter = *
    IPv6Filter = *
    EnableCompatibilityHttpListener = false
    EnableCompatibilityHttpsListener = false
    CertificateThumbprint
    AllowRemoteAccess = true

PS C:\Windows\system32>
```

Dans la fenêtre **Autorisations pour Default**, ajoutez l'utilisateur pour Ansible (trainee) et accordez lui les permissions **Exécution** en plus des permissions **Lecture** :



LAB #3 - Préparer le Contrôleur Ansible

3.1 - Installer pywinrm

Sous Debian 11, le paquet **python3-winrm** est déjà installé.

Modifiez le fichier **/etc/hosts** :

```
trainee@debian11:~/.ansible/roles$ su -
Password: fenestros

root@debian11:~# vi /etc/hosts

root@debian11:~# cat /etc/hosts
127.0.0.1       localhost
```

```
10.0.2.46       debian11.ittraining.loc debian11
10.0.2.54 web01.i2tch.loc web01
10.0.2.55 web02.i2tch.loc web02
10.0.2.56 web03.i2tch.loc web03
10.0.2.57 web04.i2tch.loc web04
10.0.2.52 targeta.i2tch.loc targeta
10.0.2.53 targetb.i2tch.loc targetb
10.0.2.58 windows10

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

3.2 - Tester la Configuration

Pour tester la configuration, commencez par modifier le fichier inventory en tant que trainee :

```
root@debian11:~# exit
logout

trainee@debian11:~$ vi inventory

trainee@debian11:~$ cat inventory
[windows]
windows10

[windows:vars]
ansible_user=trainee
ansible_password=a39dae707d
ansible_connection=winrm
ansible_winrm_server_cert_validation=ignore
```

Procédez au test :

```
trainee@debian11:~$ ansible windows -i inventory -m win_ping
windows10 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

LAB #4 - Travailler avec Ansible et Windows

4.1 - Obtenir les Informations sur Windows 10

De la même façon qu'avec Linux, il est possible d'utiliser le module **setup** pour obtenir des Facts :

```
trainee@debian11:~$ ansible windows -i inventory -m setup
windows10 | SUCCESS => {
    "ansible_facts": {
        "ansible_architecture": "64 bits",
        "ansible_architecture2": "x86_64",
        "ansible_bios_date": "04/01/2014",
        "ansible_bios_version": "rel-1.14.0-0-g155821a1990b-prebuilt.qemu.org",
        "ansible_date_time": {
            "date": "2022-03-10",
            "day": "10",
            "epoch": "1646895432,89327",
            "epoch_int": 1646895433,
            "epoch_local": "1646899032,89327",
            "hour": "07",
            "iso8601": "2022-03-10T06:57:12Z",
            "iso8601_basic": "20220310T075712893274",
            "iso8601_basic_short": "20220310T075712",
```

```
            "iso8601_micro": "2022-03-10T06:57:12.893274Z",
            "minute": "57",
            "month": "03",
            "second": "12",
            "time": "07:57:12",
            "tz": "Romance Standard Time",
            "tz_offset": "+01:00",
            "weekday": "Thursday",
            "weekday_number": "4",
            "weeknumber": "9",
            "year": "2022"
        },
        "ansible_distribution": "Microsoft Windows 10 Professionnel",
        "ansible_distribution_major_version": "10",
        "ansible_distribution_version": "10.0.19042.0",
        "ansible_domain": "",
        "ansible_env": {
            "ALLUSERSPROFILE": "C:\\ProgramData",
            "APPDATA": "C:\\Users\\trainee\\AppData\\Roaming",
            "COMPUTERNAME": "DESKTOP-P0SAHCP",
            "ComSpec": "C:\\Windows\\system32\\cmd.exe",
            "CommonProgramFiles": "C:\\Program Files\\Common Files",
            "CommonProgramFiles(x86)": "C:\\Program Files (x86)\\Common Files",
            "CommonProgramW6432": "C:\\Program Files\\Common Files",
            "DriverData": "C:\\Windows\\System32\\Drivers\\DriverData",
            "HOMEDRIVE": "C:",
            "HOMEPATH": "\\Users\\trainee",
            "LOCALAPPDATA": "C:\\Users\\trainee\\AppData\\Local",
            "LOGONSERVER": "\\\\DESKTOP-P0SAHCP",
            "NUMBER_OF_PROCESSORS": "8",
            "OS": "Windows_NT",
            "OneDrive": "C:\\Users\\trainee\\OneDrive",
            "PATHEXT": ".COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC;.CPL",
            "PROCESSOR_ARCHITECTURE": "AMD64",
```

```
            "PROCESSOR_IDENTIFIER": "Intel64 Family 15 Model 6 Stepping 1, GenuineIntel",
            "PROCESSOR_LEVEL": "15",
            "PROCESSOR_REVISION": "0601",
            "PROMPT": "$P$G",
            "PSExecutionPolicyPreference": "Unrestricted",
            "PSModulePath": "C:\\Users\\trainee\\Documents\\WindowsPowerShell\\Modules;C:\\Program
Files\\WindowsPowerShell\\Modules;C:\\Windows\\system32\\WindowsPowerShell\\v1.0\\Modules",
            "PUBLIC": "C:\\Users\\Public",
            "Path":
"C:\\Windows\\system32;C:\\Windows;C:\\Windows\\System32\\Wbem;C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\;C
:\\Windows\\System32\\OpenSSH\\;C:\\Users\\trainee\\AppData\\Local\\Microsoft\\WindowsApps",
            "ProgramData": "C:\\ProgramData",
            "ProgramFiles": "C:\\Program Files",
            "ProgramFiles(x86)": "C:\\Program Files (x86)",
            "ProgramW6432": "C:\\Program Files",
            "SystemDrive": "C:",
            "SystemRoot": "C:\\Windows",
            "TEMP": "C:\\Users\\trainee\\AppData\\Local\\Temp",
            "TMP": "C:\\Users\\trainee\\AppData\\Local\\Temp",
            "USERDOMAIN": "DESKTOP-P0SAHCP",
            "USERDOMAIN_ROAMINGPROFILE": "DESKTOP-P0SAHCP",
            "USERNAME": "trainee",
            "USERPROFILE": "C:\\Users\\trainee",
            "windir": "C:\\Windows"
        },
        "ansible_fqdn": "DESKTOP-P0SAHCP",
        "ansible_hostname": "DESKTOP-P0SAHCP",
        "ansible_interfaces": [
            {
                "connection_name": "Ethernet 2",
                "default_gateway": "10.0.2.1",
                "dns_domain": null,
                "interface_index": 13,
                "interface_name": "Intel(R) PRO/1000 MT Network Connection",
```

```
            "macaddress": "92:E1:64:91:2B:57"
        }
    ],
    "ansible_ip_addresses": [
        "fe80::1c9:fbad:7f52:9542%13",
        "10.0.2.58"
    ],
    "ansible_kernel": "10.0.19042.0",
    "ansible_lastboot": "2022-03-10 07:13:15Z",
    "ansible_machine_id": "S-1-5-21-309167381-963478272-52300377",
    "ansible_memfree_mb": 14006,
    "ansible_memtotal_mb": 16384,
    "ansible_netbios_name": "DESKTOP-P0SAHCP",
    "ansible_nodename": "DESKTOP-P0SAHCP",
    "ansible_os_family": "Windows",
    "ansible_os_installation_type": "Client",
    "ansible_os_name": "Microsoft Windows 10 Professionnel",
    "ansible_os_product_type": "workstation",
    "ansible_owner_contact": "",
    "ansible_owner_name": "trainee",
    "ansible_pagefilefree_mb": 2432,
    "ansible_pagefiletotal_mb": 2432,
    "ansible_powershell_version": 5,
    "ansible_processor": [
        "0",
        "GenuineIntel",
        "Common KVM processor",
        "1",
        "GenuineIntel",
        "Common KVM processor",
        "2",
        "GenuineIntel",
        "Common KVM processor",
        "3",
```

```
            "GenuineIntel",
            "Common KVM processor",
            "4",
            "GenuineIntel",
            "Common KVM processor",
            "5",
            "GenuineIntel",
            "Common KVM processor",
            "6",
            "GenuineIntel",
            "Common KVM processor",
            "7",
            "GenuineIntel",
            "Common KVM processor"
        ],
        "ansible_processor_cores": 8,
        "ansible_processor_count": 1,
        "ansible_processor_threads_per_core": 1,
        "ansible_processor_vcpus": 8,
        "ansible_product_name": "Standard PC (i440FX + PIIX, 1996)",
        "ansible_product_serial": null,
        "ansible_reboot_pending": true,
        "ansible_swaptotal_mb": 0,
        "ansible_system": "Win32NT",
        "ansible_system_description": "",
        "ansible_system_vendor": "QEMU",
        "ansible_uptime_seconds": 2647,
        "ansible_user_dir": "C:\\Users\\trainee",
        "ansible_user_gecos": "",
        "ansible_user_id": "trainee",
        "ansible_user_sid": "S-1-5-21-309167381-963478272-52300377-1001",
        "ansible_virtualization_role": "guest",
        "ansible_virtualization_type": "kvm",
        "ansible_win_rm_certificate_expires": "2025-03-08 07:34:49",
```

```
        "ansible_windows_domain": "WORKGROUP",
        "ansible_windows_domain_member": false,
        "ansible_windows_domain_role": "Stand-alone workstation",
        "gather_subset": [
            "all"
        ],
        "module_setup": true
    },
    "changed": false
}
```

4.2 - Exécutez une Commande

Créez le PlayBook **command.yml** :

```
trainee@debian11:~$ vi command.yml
trainee@debian11:~$ cat command.yml
---
- name: ipconfig
  hosts: windows
  tasks:
    - name: run ipconfig
      win_command: ipconfig
      register: ipconfig
    - debug: var=ipconfig
```

Exécutez le PlayBook :

```
trainee@debian11:~$ ansible-playbook command.yml -i inventory

PLAY [ipconfig]
**********************************************************************************************************
**********************************************************************
```

```
TASK [Gathering Facts]
*********************************************************************************************************************
*********************************************************************
ok: [windows10]

TASK [run ipconfig]
*********************************************************************************************************************
*********************************************************************
changed: [windows10]

TASK [debug]
*********************************************************************************************************************
*********************************************************************
ok: [windows10] => {
    "ipconfig": {
        "changed": true,
        "cmd": "ipconfig",
        "delta": "0:00:00.111976",
        "end": "2022-03-10 07:14:51.164294",
        "failed": false,
        "rc": 0,
        "start": "2022-03-10 07:14:51.052317",
        "stderr": "",
        "stderr_lines": [],
        "stdout": "\r\nConfiguration IP de Windows\r\n\r\n\r\nCarte Ethernet Ethernet 2 :\r\n\r\n   Suffixe DNS
propre  la connexion. . . : \r\n   Adresse IPv6 de liaison locale. . . . .: fe80::1c9:fbad:7f52:9542%13\r\n
Adresse IPv4. . . . . .: 10.0.2.58\r\n   Masque de sous-r seau. . . .. . . . . :
255.255.255.0\r\n   Passerelle par d faut. . . .. . . . . : 10.0.2.1\r\n",
        "stdout_lines": [
            "",
            "Configuration IP de Windows",
            "",
            "",
            "Carte Ethernet Ethernet 2 :",
```

```
            "",
            "   Suffixe DNS propre à la connexion. . . : ",
            "   Adresse IPv6 de liaison locale. . . . .: fe80::1c9:fbad:7f52:9542%13",
            "   Adresse IPv4. . . . . . . . . . . . . .: 10.0.2.58",
            "   Masque de sous-réseau. . . .. . . . . . : 255.255.255.0",
            "   Passerelle par défaut. . . .. . . . . . : 10.0.2.1"
        ]
    }
}

PLAY RECAP
*********************************************************************************************************
*********************************************************************
windows10                  : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```

4.3 - Exécuter un script PowerShell

Créez un script PowerShell pour démarrer le service Windows Update :

```
trainee@debian11:~$ vi script1.ps1
trainee@debian11:~$ cat script1.ps1
Start-Service -Name wuauserv
```

Créez ensuite un PlayBook appelé **winupdate.yml** :

```
trainee@debian11:~$ vi winupdate.yml
trainee@debian11:~$ cat winupdate.yml
- name: "PowerShell script"
  hosts: "windows"
  gather_facts: "false"
  tasks:
    - name: "Windows Update"
```

```
      script: "script1.ps1"
```

Exécutez le PlayBook :

```
trainee@debian11:~$ ansible-playbook winupdate.yml -i inventory

PLAY [PowerShell script]
******************************************************************************************************************
*************************************************************************

TASK [Windows Update]
******************************************************************************************************************
****************************************************************************
changed: [windows10]

PLAY RECAP
******************************************************************************************************************
***************************************************************************************
windows10                  : ok=1    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```

4.4 - Installer un Logiciel avec Chocolatey

Chocolatey est un gestionnaire de paquets pour Windows™.

Créez le PlayBook **firefox.yml** :

```
trainee@debian11:~$ vi firefox.yml
trainee@debian11:~$ cat firefox.yml
---
- name: Install Firefox using Chocolatey
  hosts: all
  tasks:
```

```
    - name: Install Firefox
      win_chocolatey:
        name: firefox
        state: present
```

Exécutez le PlayBook :

```
trainee@debian11:~$ ansible-playbook firefox.yml -i inventory

PLAY [Install Firefox using Chocolatey]
**********************************************************************************************************
**********************************************************

TASK [Gathering Facts]
**********************************************************************************************************
*********************************************************************
ok: [windows10]

TASK [Install Firefox]
**********************************************************************************************************
*********************************************************************
[WARNING]: Chocolatey was missing from this system, so it was installed during this task run.
changed: [windows10]

PLAY RECAP
**********************************************************************************************************
*********************************************************************
windows10                  : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```

4.5 - Créer un Utilisateur Local

Créez le PlayBook **user.yml** :

```
trainee@debian11:~$ vi users.yml
trainee@debian11:~$ cat users.yml
---
- name: Create a user
  hosts: windows
  tasks:
    - name: Create Jean
      win_user:
        name: jean
        password: P@$$w0rd
        state: present
        groups:
          - utilisateurs
```

Exécutez le PlayBook :

```
trainee@debian11:~$ ansible-playbook users.yml -i inventory

PLAY [Create a user]
**********************************************************************************************************
**************************************************************************

TASK [Gathering Facts]
**********************************************************************************************************
**************************************************************************
ok: [windows10]

TASK [Create Jean]
**********************************************************************************************************
**************************************************************************
changed: [windows10]

PLAY RECAP
**********************************************************************************************************
```

```
**********************************************************************************
windows10                  : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```

Vérifiez que le compte a été créé :