

SETUP VAGRANT

I. Prérequis

II. Installations

III. Troubleshoot

IV. Commandes vagrant

V. Troubleshoot Vagrant

I. Prérequis

- Une machine avec ~ 16Go de RAM
- Virtualbox \geq 6.1 installé
- Vagrant récent installé

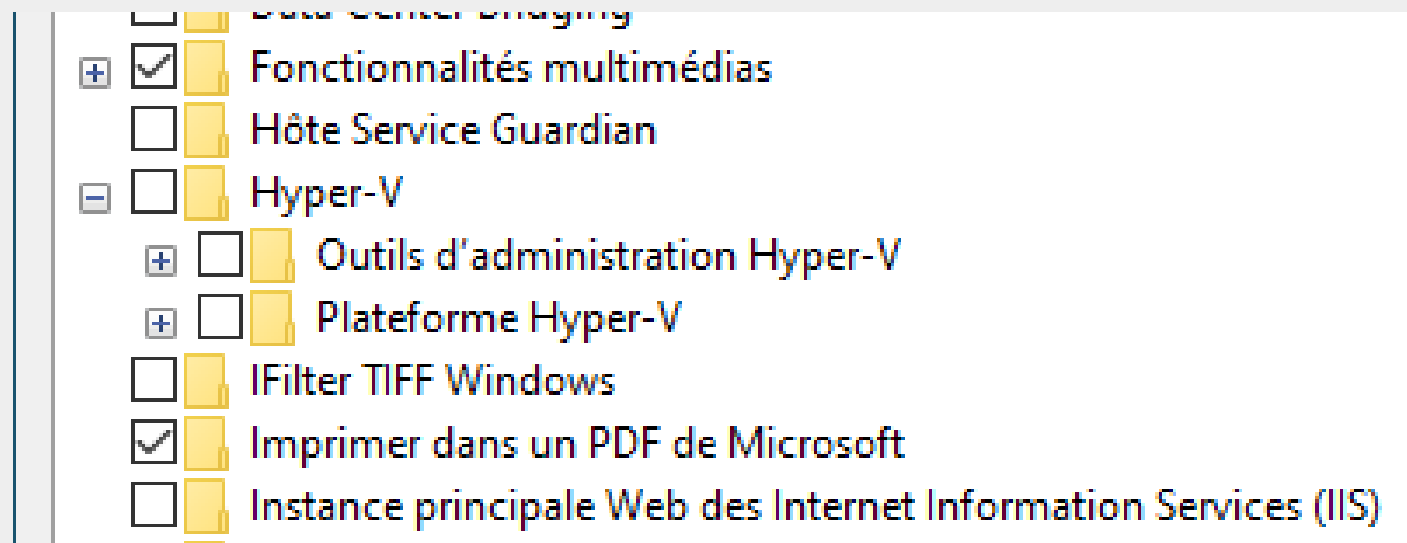
II. Installations

- précautions (**cf infra**):
 - désactiver les autres hyperviseurs KVM/Qemu, hyper-v, ...
 - vérifier si la virtualisation (VT-X ou AMD-V dans le BIOS)
- Virtualbox: **ici**
- Vagrant: **ici**

III. Troubleshoot VirtualBox

désactiver hyper-v sous Windows

- moteur de recherche de windows > « activer / désactiver les fonctionnalités Windows »



Virtualisation hardware activée

- Windows: « gestionnaire de tâches » (Ctrl + Suppr. + Alt)
 - onglet « performances »

Vitesse de base :	3,60 GHz
Sockets :	1
Cœurs :	8
Processeurs logiques :	8
Virtualisation :	Activé
Cache de niveau 1 :	512 Ko
Cache de niveau 2 :	2,0 Mo
Cache de niveau 3 :	12,0 Mo

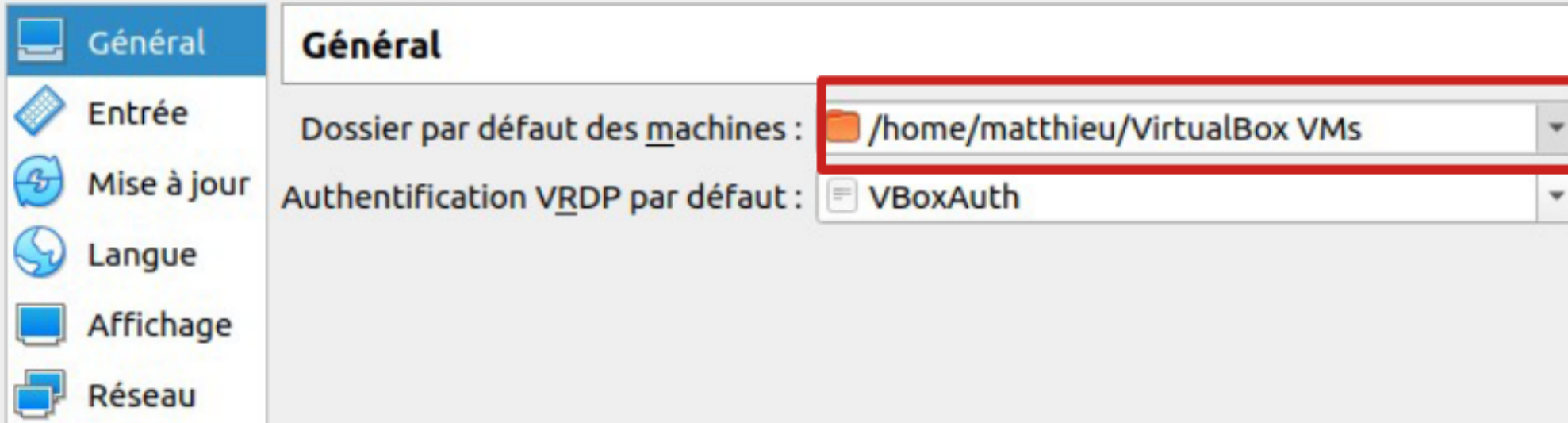
- Linux 64 bit: observer les flags **vmx** ou **svm** dans

```
cat /proc/cpuinfo
```

Dossier des VMS

- Windows / Linux / Mac OS: par défaut `~\VirtualBox VMs`





“ *on peut changer ce dossier si le disque courant est rempli !!* ”

IV. Lancer la VM

Principe

- Vagrant permet de lancer des **Machines Vrtuelles** sur des **Hyperviseurs** à partir d'**images** et via un **script**
- Ces vms sont nommées *box vagrant*, le script *Vagrantfile*
- Une fois vagrant installé, on affiche la liste des boxes présentes dans un terminal `vagrant box list`
- télécharger une box depuis **<https://portal.cloud.hashicorp.com/vagrant/discover>**
`vagrant box add <box_name> [--box-version=<version>]`

Fichier Vagrantfile

- script Ruby pilotant création et provisionnement de la VM

```
Vagrant.configure(2) do |config|  
  
  # int= "nom de l'interface réseau connectée au routeur (ip a || ipconfig /all)"  
  # ip = "adresse ip disponible sur le sous réseau local (ping pour tester)"  
  # cidr = "24 (si masque réseau en 255.255.255.0)"  
  ...  
  # machine.vm.network "public_network", bridge: "#{int}",  
  #   ip: "#{ip}",  
  #   netmask: "#{cidr}"
```

- dans un terminal, dans le dossier contenant ce fichier: **vagrant up**
- se connecter à la VM en ssh : **vagrant ssh**

DNS local

- si le Vagrantfile définit **une interface de type « bridge publique »**
=> pour se connecter en *HTTP / TLS* depuis la machine hôte
- éditer le fichier **hosts** de la machine hôte avec les *droits admin !!!*
- associer l'adresse IP de cette interface au nom d'hôte de la VM

```
## LINUX / MAC OS => /etc/hosts
## WINDOWS          => c:\Windows\System32\drivers\etc\hosts
## -----IP----- ---Domain---
xxx.yyy.zzz.ttt <hostname>
```

V. Troubleshoot Vagrant

dossier des boxes

- `~/ .vagrant.d/boxes/`
- vérifier le nom et la taille de l'image

changer de version

1. `vagrant box remove <box_name> [--box-remove=<version>]`
2. `vagrant box add <box_name> [--box-version=<version>]`

« Rip it up & Start again »

en cas de problème de lancement

1. si vm lancée mais non connectée, *arrêter* `vagrant halt`
2. *détruire complètement* la vm `vagrant destroy [-f]`
3. supprimer le dossier courant `.vagrant/`
4. re-vérifier la configuration du **Vagrantfile**
5. `vagrant up`

changement de configuration (RAM / CPU ...)

- `vagrant reload` == *halt & up*

manipuler plusieurs VMs

- si le **Vagrantfile** définit plusieurs VMs
- `vagrant up | halt | destroy` : pour *toutes les VMs*
- `vagrant up | halt | destroy | ssh <hostname>` : *pour une VM*

configuration SSH

- `vagrant ssh`: exécute un client ssh avec une *configuration embarquée* dans l'image vagrant
- configuration avec: `vagrant ssh-config`
- utilise l'interface **NAT** créée automatiquement avec la box
- analogue à ce qu'on peut faire dans `~/.ssh/config`