**TP sur Vagrant sur Ubuntu**

**Maimouna Ndiaye**  
**DevOps - Devoir**  
**Professeur : Ngor Seck**  
**Février 2025**

### **Introduction**

Dans le cadre de ce travail pratique, l'objectif principal est de créer des machines virtuelles (VM) sous Linux en utilisant **Vagrant**. Vagrant est un outil de gestion de machines virtuelles, permettant de configurer des environnements de développement de manière simple et automatisée.

Ce TP a pour but de familiariser les étudiants avec la création et la gestion de VM sur des systèmes de virtualisation comme VirtualBox, tout en appliquant des concepts d’infrastructure as code. À travers ce TP, nous allons découvrir comment utiliser Vagrant pour automatiser la configuration d'une VM Linux, en définissant les ressources et les paramètres nécessaires pour le bon fonctionnement de cette machine virtuelle.

Au fil de ce travail, nous verrons également l'importance de la gestion de l'environnement de développement pour garantir des configurations reproductibles et cohérentes, facilitant ainsi les déploiements et les tests dans des environnements virtuels.

### **1. Configuration du Vagrantfile**

Le fichier **Vagrantfile** permet de définir les paramètres de la machine virtuelle, notamment :

* **La box utilisée** : ubuntu/trusty64, qui est une image préconfigurée d'Ubuntu.
* **Mise à jour des box désactivée** (vm.box\_check\_update = false) pour éviter les mises à jour automatiques.
* **Configuration réseau** :
  + forwarded\_port redirige le port 8080 de l'hôte vers le port 8082 de la VM pour l'accès aux applications web.
  + private\_network avec l'IP statique 192.168.33.10 pour un accès direct à la VM depuis l'hôte.
  + public\_network permettant un accès à la VM depuis le réseau local.
* **Dossier synchronisé** (synced\_folder) pour partager des fichiers entre l'hôte et la VM.
* **Fournisseur VirtualBox** avec des paramètres spécifiques :
  + vb.gui = false pour un démarrage en mode headless (sans interface graphique).
  + vb.memory = "1024" pour allouer 1024 Mo de RAM à la VM.

### **Code du Vagrantfile**

ruby

CopierModifier

Vagrant.configure("2") do |config|

config.vm.define "server" do |webserver|

webserver.vm.box = "ubuntu/trusty64"

webserver.vm.box\_check\_update = false

webserver.vm.network "forwarded\_port", guest: 8080, host: 8082

webserver.vm.network "private\_network", type: "static", ip: "192.168.33.10"

webserver.vm.network "public\_network"

webserver.vm.synced\_folder "./tomcatwebapps", "/opt/tomcat/webapps"

webserver.vm.provider "virtualbox" do |vb|

vb.gui = false

vb.name = "webserver-tomcat"

vb.memory = "1024"

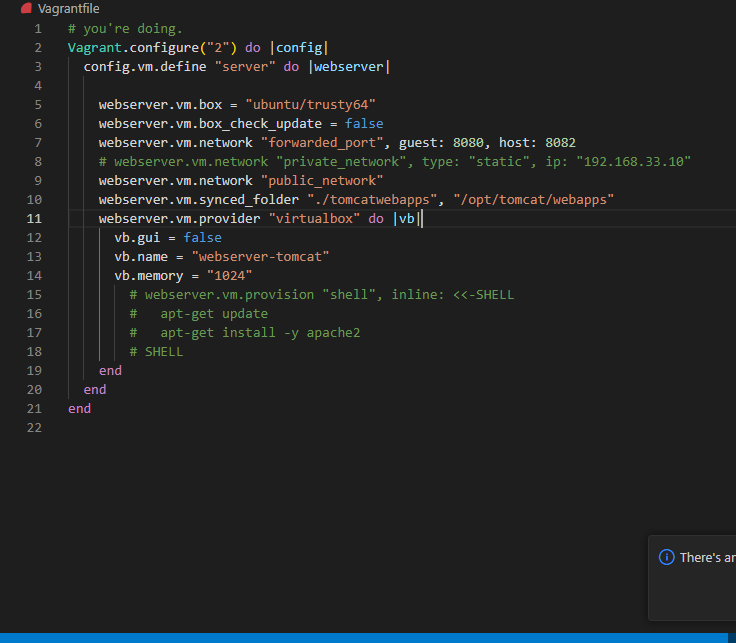
end

End

End

### **Capture d'écran : Configuration du Vagrantfile**

*Figure 1 : Le fichier Vagrantfile configuré pour une VM Linux avec VirtualBox*



### **2. Lancement de la VM avec Vagrant**

Après avoir configuré le Vagrantfile, la commande suivante permet de lancer la machine virtuelle :

Bash

CopierModifier

Vagrant up

* Cette commande permet de :
  + Télécharger le box si elle n'est pas déjà disponible localement.
  + Créer la VM avec les paramètres spécifiés.
  + Configurer les réseaux et les ports.
  + Démarrer la VM en arrière-plan.

### **Problème rencontré**

Lors de l'exécution de vagrant up, une erreur s'est produite :

* La VM webserver-tomcat a échoué à démarrer avec un **exit code 1**.
* Le message d'erreur indique un problème avec VBoxManage.exe, l'outil de gestion de VirtualBox.
* La cause probable de cette erreur pourrait être liée à :
  + Un problème de compatibilité avec VirtualBox.
  + Une configuration réseau incorrecte.
  + Un manque de ressources système (mémoire vive ou CPU).

### **Capture d'écran : Erreur de démarrage**

*Figure 2 : Erreur lors du démarrage de la VM avec Vagrant*



### **3. Vérification dans VirtualBox**

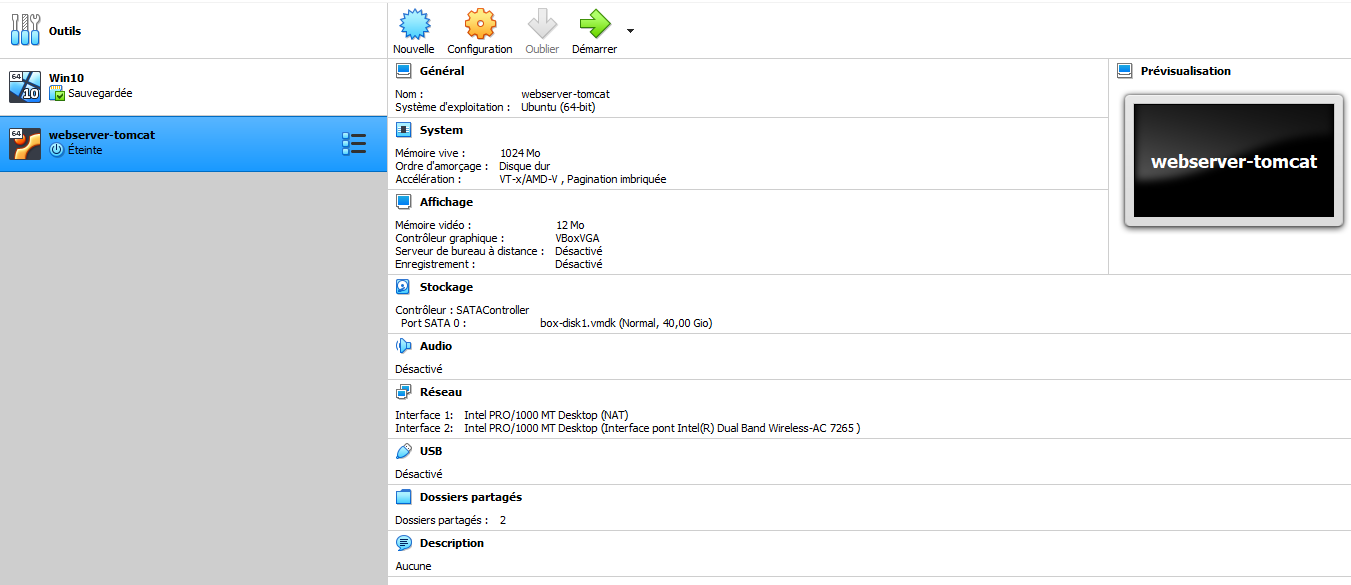
Après l'erreur de démarrage, une vérification dans **VirtualBox** montre que la VM webserver-tomcat a bien été créée avec les configurations suivantes :

* **Nom** : webserver-tomcat
* **Système d'exploitation** : Ubuntu (64-bit)
* **Mémoire vive** : 1024 Mo
* **Adaptateurs réseau** :
  + Interface 1 : NAT pour l'accès à Internet.
  + Interface 2 : Pont réseau (Bridge) pour un accès local.
* **Dossiers partagés** : Deux dossiers partagés entre l'hôte et la VM.

Cependant, la VM apparaît comme **Éteinte**, confirmant ainsi l'erreur de démarrage observée précédemment.

### **Capture d'écran : Vérification dans VirtualBox**

*Figure 3 : Configuration de la VM dans VirtualBox après l'exécution de Vagrant*



### **4. Analyse et solutions possibles**

Pour résoudre l'erreur de démarrage, plusieurs approches peuvent être envisagées :

* **Vérification des logs** : Les fichiers de logs dans le dossier Logs\VBoxHardening.log peuvent fournir plus de détails sur l'erreur.
* **Compatibilité de VirtualBox** : S'assurer que VirtualBox est à jour et compatible avec la version de Vagrant utilisée.
* **Configuration réseau** : Tester d'autres paramètres réseau pour éviter les conflits d'adresses IP.
* **Droits administrateur** : Lancer VirtualBox et Vagrant en tant qu'administrateur pour éviter les problèmes de permissions.

Cependant dans notre cas de figure, le soucis était principalement lié au fait que nous avons Docker Desktop installé sur la machine ce qui bloqué le bon fonctionnement de virtualbox.

Ci-dessous quelques commandes de base de vagrant:

* **vagrant init** : crée un fichier Vagrantfile dans le répertoire courant. Ce fichier contient la configuration de la Vagrant Box à utiliser. Tu peux utiliser cette commande pour créer un nouveau projet Vagrant.
* **vagrant up** : démarre la Vagrant Box. Si la box n’est pas encore installée, Vagrant la téléchargera et la configura avant de la démarrer.
* **vagrant ssh** : se connecte à la Vagrant Box en utilisant SSH. Tu peux utiliser cette commande pour accéder à la console de la box et exécuter des commandes sur le système d’exploitation de la box.
* **vagrant halt** : arrête la Vagrant Box. Tu peux utiliser cette commande pour mettre la box en pause ou pour l’arrêter complètement.
* **vagrant destroy** : supprime la Vagrant Box. Cette commande arrête la box et la supprime de ta machine. Tu peux utiliser cette commande pour nettoyer ton environnement de développement.

### **Conclusion**

Ce TP a permis de se familiariser avec **Vagrant** pour la création et la gestion de machines virtuelles sous **Linux**. Bien que la configuration ait été correctement définie dans le fichier Vagrantfile, l'erreur de démarrage montre l'importance de bien comprendre les interactions entre Vagrant et VirtualBox, en particulier en ce qui concerne la gestion réseau et les permissions système.

Des solutions potentielles ont été proposées pour résoudre ce problème, illustrant ainsi l'approche méthodique nécessaire à la résolution de problèmes dans des environnements virtualisés.