

IVC-Vagrant-3 : Partage et réutilisation d'environnements virtuels

Lom M. HILLAH

Table des matières

1	Introduction	3
1.1	Objectifs	3
1.2	Conventions	3
1.3	Pré-requis	4
2	Rechercher un box	4
3	Installer un nouveau box	4
4	Mettre à jour un box	5
5	Vérifier l'existence de mises à jour	5
6	Supprimer un box	5
7	Distribuer un box dans le dépôt Vagrant Cloud	6
8	Partager l'accès à sa VM via Vagrant Cloud	6
8.1	Accès via HTTPS	7
8.2	Accès via SSH	7

9 Créer son propre box	8
9.1 Avec Vagrant	8
9.2 Avec Packer	9
10 Feedback	9
11 Licence	9

1 Introduction

Atelier pratique pour la création et le partage de box.

1.1 Objectifs

L'atelier Vagrant-2 explique comment automatiser la gestion de configuration d'une machine vagrant avec un outil comme Ansible. Au-delà, on pourrait également vouloir créer sa propre distribution de base et la partager, ou encore de partager l'accès à la VM courante pour effectuer des démonstrations (p. ex. donner l'accès à une application web hébergée dans la VM), ou pour solliciter une assistance à distance (via une session SSH).

Dans cet atelier, nous apprenons à :

- chercher des box sur Hashicorp Cloud et les réutiliser,
- publier son box du Hashicorp Cloud,
- partager son environnement virtuel :
 - via HTTP à travers Hashicorp Cloud
 - via SSH à travers Hashicorp Cloud
- créer son propre box.

1.2 Conventions

Nous utiliserons la ligne de commande dans un terminal. Le prompt sera symbolisé par :

```
$>
```

Les configurations sont indiquées dans une police monospace comme dans l'exemple suivant :

```
config.vm.synced_folder "/Users/aUser/deployed-assets/", \
"/var/www/assets", disabled: true
```

Dans l'exemple ci-dessus, le caractère \ indique que la configuration est écrite sur une seule ligne. Le passage à la ligne est employé pour améliorer la lisibilité et éviter la troncature de la ligne par la marge.

1.3 Pré-requis

- Vous devez avoir suivi l'atelier Vagrant-1
- [De préférence, il faudrait également avoir suivi l'atelier Vagrant-2].
- Vous devez créer un compte sur Vagrant Cloud (le dépôt hébergeant les box) pour être en mesure de réaliser certaines manipulations.
 - <https://app.vagrantup.com/>

2 Rechercher un box

Pour rechercher des box publiés dans le dépôt, il faut se connecter à la page <https://app.vagrantup.com/boxes/search>. Par exemple, vous pouvez chercher tous les box basés sur ubuntu que nous utilisons dans cet atelier, Un filtre permet de chercher les box selon le provider.

Lorsqu'on clique sur un box, on découvre les providers supportés (ex. VirtualBox, VMware). Cela permet de savoir si le box qui nous intéresse est géré par le provider installé sur la machine hôte.

3 Installer un nouveau box

Nous avons déjà vu comment installer un box dans l'atelier Vagrant-1. Cette fois-ci nous utilisons un box publié dans le dépôt Vagrant Cloud :

```
$> vagrant box add hashicorp/precise64
```

Il est cependant généralement plus simple d'indiquer le nom du box (publié dans Vagrant Cloud) qui vous intéresse directement dans le `Vagrantfile` du nouveau projet que vous souhaitez mettre en place.

4 Mettre à jour un box

La mise à jour d'un box est utile, car le fournisseur peut avoir mis à jour des logiciels spécifiquement déployés dans ce box, par exemple pour des raisons de sécurité.

- Pour mettre à jour le box du projet Vagrant courant :

```
$> vagrant box update
```

- Pour mettre à jour un box spécifique :

```
$> vagrant box update --box nom-du/box
```

Vous pouvez également indiquer le nom du provider (e.g. `virtualbox`) avec l'option `--provider`

5 Vérifier l'existence de mises à jour

Pour vérifier l'existence de mises à jour, il suffit d'utiliser la commande :

```
$> vagrant box outdated --global
```

Sans l'option `--global`, seule l'existence de mises à jour pour le box courant (dans le projet Vagrant courant) sera vérifiée.

6 Supprimer un box

Vous pouvez supprimer un box de votre catalogue local des boxes avec la commande :

```
vagrant box remove nom-du/box
```

Cette commande supporte les options suivantes :

- `--force` : supprime sans confirmation
- `--provider nom-du-provider` : le nom du provider du box à supprimer (p. ex. virtualbox)
- `--box-version version-du-box` : la version du box à supprimer.

7 Distribuer un box dans le dépôt Vagrant Cloud

Si vous ne disposez pas d'un compte payant, tout box que vous y enregistrerez sera public.

Pour créer un box dans le dépôt, il faut se connecter à <https://app.vagrantup.com/boxes/new>

Suivez les instructions pour décrire le box ; indiquez la version et les providers supportés.

Dans le cas d'un compte gratuit, il faut indiquer l'URL à partir duquel le box sera téléchargé. Il n'est pas possible dans ce cas de le déposer directement dans le dépôt Vagrant Cloud.

8 Partager l'accès à sa VM via Vagrant Cloud

Il faut disposer d'un compte sur Vagrant Cloud pour partager l'accès à son environnement virtuel.

Vous devez d'abord vous y connecter. Pour vérifier si vous êtes connecté :

```
$> vagrant login --check
```

Si vous n'êtes pas encore connecté à Vagrant Cloud, vous pouvez le faire avec :

```
$> vagrant login
```

Il faudra fournir votre login et mot de passe créés lors de la création de compte, ou alors un token d'authentification que vous aurez généré pour ce faire (allez dans les paramètres de sécurité de votre compte).

Pour se déconnecter :

```
$> vagrant login --logout
```

8.1 Accès via HTTPS

Il faut que votre VM dispose de sa propre adresse IP dans votre réseau local (configuration Bridge), ou qu'un port de la machine hôte soit redirigé vers le port adéquat sur la VM.

Avant de commencer le partage, vous devez installer le plugin **vagrant-share** : <https://www.vagrantup.com/docs/share>

Pour activer l'accès, utilisez la commande :

```
$> vagrant share
```

Vous pouvez optionnellement indiquer les ports http ou https exposés sur la VM, avec l'option `-http` ou `https`.

Vagrant générera alors une URL unique pour partager la connexion à votre VM. Communiquez cette URL à un collègue qui, à l'aide d'un browser pourra se connecter à votre environnement, de n'importe où dans le monde.

Pour arrêter le partage, il suffit de faire `Ctrl-C`.

8.2 Accès via SSH

Par défaut l'accès à la VM est partagée via HTTP(S) avec la commande `vagrant share`. Il est cependant également possible de partager l'accès à votre environnement virtuel via SSH :

```
$> vagrant share --ssh
```

On peut :

- désactiver l'utilisation de mot de passe avec `--ssh-no-password` au lieu de `--ssh`;
- requérir une seule connexion ssh avec `--ssh-once`. Les clés sont détruites après la connexion, rendant toute connexion ultérieure impossible;
- désactiver le partage via HTTP avec `--disable-http`.

Une fois le nom de connexion (généré par Vagrant) et le mot de passe communiqués à tierce partie, celle-ci peut se connecter à votre VM avec :

```
$> vagrant connect --ssh nom-de-connexion-généré-pour-votre-vm
```

Il faut cependant que la personne qui se connecte à votre VM en SSH possède un compte sur Vagrant Cloud.

Rappel des commandes essentielles pour effectuer un partage : <https://www.vagrantup.com/docs/share>

9 Créer son propre box

Pour créer votre propre box, plusieurs méthodes existent :

- utiliser directement l'interface graphique de VirtualBox pour en créer,
- combiner vagrant et VirtualBox,
- utiliser une technologie comme QEMU/KVM,
- utiliser [Packer](#)

9.1 Avec Vagrant

Dans le cas du repackaging d'un box existant, dont vous avez configuré à votre goût l'environnement virtuel (c.-à-d vous y avez installé des logiciels spécifiques, ex. PHP, Apache, etc.), vous pouvez utiliser la commande `vagrant package`, comme dans l'exemple suivant :

```
$> vagrant package --output ivc2021.box
```

Vagrant placera le nouveau box empaqueté à l'emplacement indiqué par l'option `--output`.

Vous pouvez ajouter le nouveau box à votre catalogue local avec la commande :

```
$> vagrant box add ivc2021 ivc2021.box
```

Puis, vous pouvez détruire l'environnement courant et supprimer son `Vagrantfile`¹ avant d'initialiser votre nouveau box :

¹Faites quand même une sauvegarde du `Vagrantfile`


```
$> vagrant destroy  
$> rm -f Vagrantfile  
$> vagrant init ivc2021
```

Pour en savoir plus dans l'utilisation de vagrant pour la création de box, référez-vous à la documentation suivante : <https://www.vagrantup.com/docs/boxes/base>

9.2 Avec Packer

Packer est un outil servant à créer des images pour des plateformes comme AWS, VMware, VirtualBox, Docker et bien d'autres², à partir d'un seul fichier de configuration (en JSON). Packer supporte également l'intégration de gestionnaires de configuration pour la configuration avancée des images construites (ex. installation de logiciels avec Ansible).

Documentation de Packer : <https://learn.hashicorp.com/packer>

10 Feedback

Lom M. Hillah (lhillah@parisnanterre.fr)

11 Licence

Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

²<https://www.packer.io/docs/builders>