

# Installation de Docker sous Windows 10 Pro<sup>1</sup>

*Pour tous les logiciels à installer ici, vous devez le faire directement sur votre ordinateur et non dans une machine virtuelle. Vous devez avoir Windows Professionnel (7 ou 10). Vous devez désactiver votre antivirus.*

Trois approches s'offrent à vous. Vous en choisissez une seulement. La première oblige un redémarrage de l'ordinateur pour passer de Docker à la virtualisation ou vice versa. Les 2 autres approches sont plus instables et peuvent ne pas fonctionner sur votre ordinateur.

## Approche 1

Si vous avez déjà installé Docker, désinstallez-le complètement.

D'abord, il faut activer l'**Hyper-V dans Windows 10** en ouvrant une console Powershell et taper la commande suivante :

```
Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V -All
```

**Vous devez ensuite redémarrer Windows 10.**

Suivant les directives du [site suivant](#), voici les étapes pour créer une nouvelle entrée dans le menu « boot » de Windows 10.

Ouvrez une console CMD (pas Powershell cette fois-ci) en mode administrateur et tapez les commandes suivantes :

```
bcdedit /copy {current} /d "Disable Hyper-V"
```

Le message suivant apparaîtra avec votre numéro GUID :

The entry was successfully copied to {<votre GUID apparaîtra ici>}.

Exécutez la commande suivante en remplaçant « entrez votre GUID » par votre GUID.

```
bcdedit /set {<entrez votre GUID>} hypervisorlaunchtype off
```

---

<sup>1</sup> Version 3, février 2020.

Le message suivant s'affichera :

The operation completed successfully.

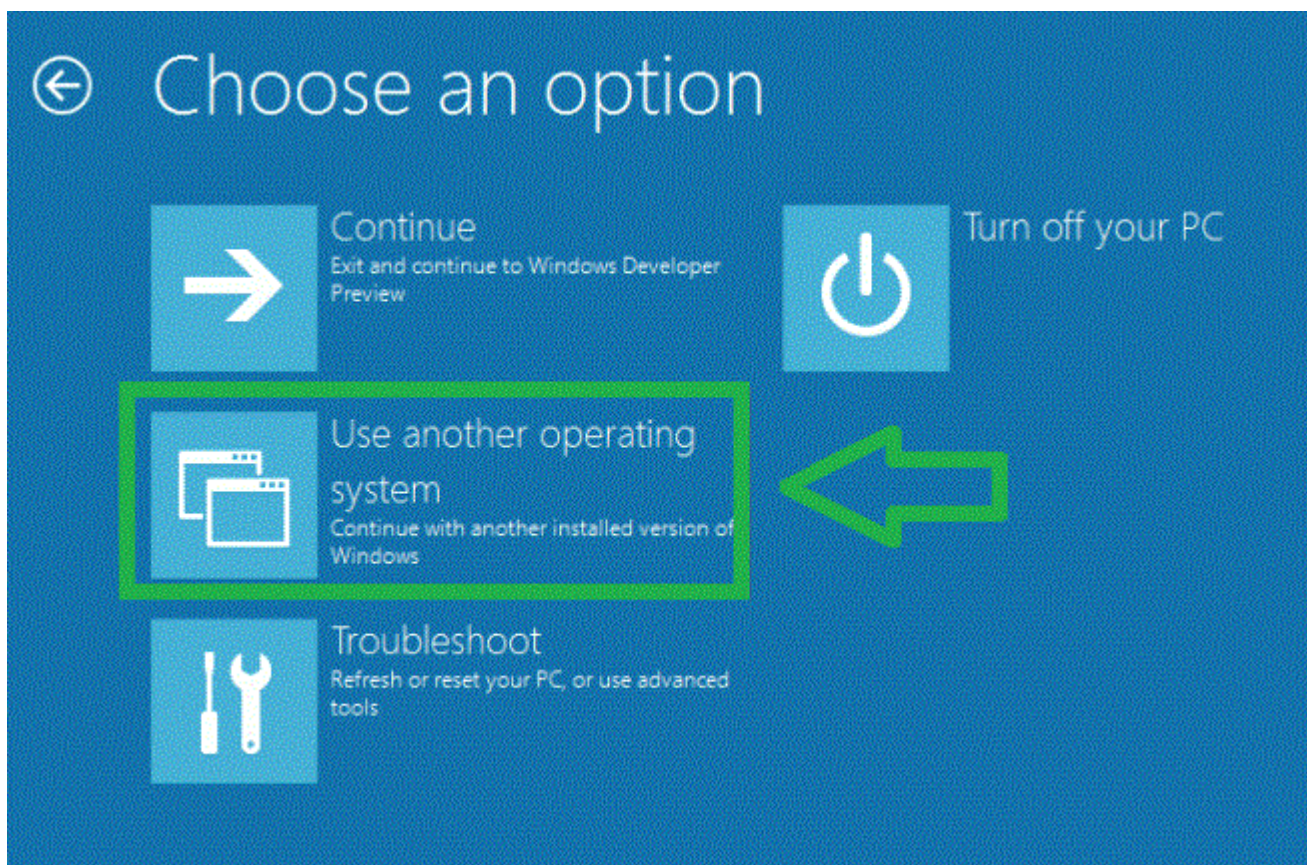
Voici un exemple d'exécution :

```
Microsoft Windows [Version 10.0.17763.437]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>bcdedit /copy {current} /d "Disable Hyper-V"
The entry was successfully copied to {1258b0da-24e1-11e9-a8ad-f9bd8867be5c}.

C:\WINDOWS\system32>bcdedit /set {1258b0da-24e1-11e9-a8ad-f9bd8867be5c} hypervisorlaunchtype off
The operation completed successfully.
```

Redémarrez votre ordinateur **en enfonçant la touche SHIFT tout au long du démarrage**. Vous devriez obtenir un choix comme suit :



Si vous choisissez « Use another OS » et ensuite, le choix « Disable Hyper-V », vous vous retrouverez dans votre environnement Windows **sans Hyper-V** et vous pourrez démarrer vos machines

virtuelles avec Virtualbox ou VMWare. Si vous choisissez plutôt « Continue », vous vous retrouverez sous Windows **avec Hyper-V** et vous pourrez alors installer Docker et Docker-Compose.

Pour installer Docker, suivez les étapes de ce site <https://docs.docker.com/docker-for-windows/install/>

Pour installer Docker-Compose, suivez les étapes de ce site <https://docs.docker.com/compose/install/>  
Choisissez l'onglet « **Windows Server** » et suivez les étapes d'installation dans Powershell.

Quand vous êtes dans Windows 10 sans Hyper-V, Docker vous indiquera qu'il ne peut pas fonctionner. Si vous êtes dans Windows 10 avec Hyper-V, c'est Virtualbox ou VMWare qui dira qu'il ne fonctionne pas.

## Approche 2

- Installez Virtualbox 6.0 (version antérieure qu'on [trouve ici](#)).
- Assurez-vous de fermer votre antivirus et parefeu.
- Téléchargez et installez **Docker Toolbox** disponible [ici](#).
- Si la commande n'est pas reconnue sous Powershell, il faut modifier la variable PATH sous Windows.
- Voici les commandes à exécuter dans Powershell pour finaliser l'installation. Les commandes vont créer une machine virtuelle dans Virtualbox qui contiendra Docker:

```
docker-machine create --driver virtualbox box
docker-machine env --shell powershell box
docker-machine env --shell powershell box | Invoke-Expression
```

- On peut vérifier si la machine virtuelle est bien installée par la commande suivante :

```
docker-machine ls
```

- On peut vérifier si la machine virtuelle est active:

```
docker-machine status box
```

- Si la machine est arrêtée, on tape:

```
docker-machine start box
```

- Après votre travail avec Docker, quand on a terminé de faire fonctionner Docker, on tape:

```
docker-machine stop box
```

- Pour connaître l'adresse IP du serveur Docker, on tape:

```
docker-machine ip box
```

L'adresse IP servira à accéder à un serveur installé dans un conteneur Docker comme par exemple, <http://192.168.99.102>

- Vérifiez sur Docker fonctionne bien en tapant par exemple :

```
docker run hello-world
```

Cela ne devrait pas générer d'erreur. Si c'est le cas, il faut recommencer.

- Si votre machine Docker ne fonctionne plus bien, on tape ce qui suit. Ensuite, il faut reprendre les étapes ci-haut pour recréer la machine Docker sous Virtualbox.

```
docker-machine kill box
docker-machine rm box
```

Si la reprise des étapes précédentes ne fonctionnent pas encore, il vous reste à essayer l'approche 2 ci-bas.

## Approche 3

- Installez Virtualbox 6.0 (version antérieure qu'on [trouve ici](#)). Ne pas installer la version 6.1.
- Installez Vagrant [disponible ici](#).
- Ouvrez une console Powershell et vérifiez que la commande **vagrant** est disponible.
- Créez le dossier **c:\bionic** et rendez-vous dans ce dossier via une console Powershell.
- Tapez les commandes suivantes :

```
vagrant init demers/bionic64-dockercompose
vagrant up
```

- La première commande crée un fichier **Vagrantfile** automatiquement dans le dossier courant.
- Cette deuxième commande démarre une machine virtuelle dans Virtualbox.

- Si vous recevez plusieurs fois le message d'erreur suivant :

`default: Warning: Authentication failure. Retrying...`

vous devrez arrêter la processus en tapant les touches CTRL-C (ou attendre pendant quelques minutes).

- Ne fermez pas la console Powershell. Ouvrez l'application Virtualbox. Vous verrez qu'une machine virtuelle est en exécution (en fonction). Cliquez sur cette machine.
- Tapez le bouton « **Afficher** ». Vous verrez un « Login ». Vous entrez « **vagrant** ».
- Le mot de passe est aussi « **vagrant** ».

- Par la suite, tapez la commande suivante (à l'intérieur de la machine virtuelle) :

```
cp -f authorized_keys .ssh
```

- Aucun message de réponse ne devrait apparaître (à part l'invite Linux).
- Retournez dans la console **Powershell** et tapez :

```
vagrant halt
```

- La machine virtuelle va s'arrêter. Ensuite, tapez la commande suivante :

```
vagrant reload
```

- Il ne devrait plus y avoir d'erreurs. Sinon, communiquez avec votre professeur.
- Vous pouvez maintenant accéder à la machine virtuelle par la commande :

```
vagrant ssh
```

- Vous serez alors connecter sur votre machine virtuelle dans une console Bash.
- Vous pouvez alors utiliser la commande **docker** et **docker-compose**.
- Le dossier **/vagrant** dans la machine virtuelle Linux vous permet d'accéder aux fichiers qui se trouvent sur votre ordinateur dans **C:\bionic\** C'est un dossier de partage.
- Vous quittez la machine virtuelle en tapant **exit**
- Vous arrêtez la machine virtuelle en tapant :

```
vagrant halt
```