

DÉCOUVERTE ET INSTALLATION DE DOCKER

Introduction

Cette partie du cours va être très rapide mais aussi très importante. Vous allez connaître les différentes versions et éditions existantes de Docker. Une fois qu'on aura fini les présentations, on passera ensuite directement à l'étape d'installation et de vérification d'install.

Vue d'ensemble des éditions et des versions de Docker

Docker est disponible en deux éditions:

- **Docker Community Edition** (CE)
- **docker Enterprise Edition** (EE)

Docker Community Edition (CE) est idéale pour les développeurs individuels et les petites équipes cherchant à se familiariser avec Docker et à expérimenter des applications basées sur des conteneurs. De plus cette **version est gratuite**. Ça sera la version que nous utiliserons.

Docker Enterprise Edition (EE) est conçue pour les équipes de développement d'entreprise et les équipes système qui créent, expédient et exécutent des applications critiques pour la production à grande échelle (Elle n'est pas gratuite).

La version communautaire Docker dispose de trois types de canaux de mise à jour, **stable**, **test** et **Nightly** :

- **Stable** : cette version vous donne les dernières releases pour une disponibilité générale
- **Test** : cette version vous fournit des pré-versions prêtes à être testées avant la disponibilité générale
- **Nightly** : cette version vous présente les dernières versions de build en cours pour la prochaine release, elle fournit donc un accès plus rapide aux nouvelles fonctionnalités et correctifs pour les tests.

Pour ce cours, nous utiliserons la version communautaire de Docker

Installation Linux

Dans cette partie je vais **installer le moteur Docker** sur mon OS **Fedora version 29** avec la version 18.09.6 du moteur Docker. Si vous avez une autre distribution, je vous conseille alors de vous référer directement à la documentation qui vous expliquera pas à pas quelles sont les commandes à lancer pour votre installation.

On va commencer par ajouter le repository officiel de Docker sur notre machine. Cette étape va nous permettre de profiter des dernières mises à jour stable directement depuis le repository Docker.

voici la commande à lancer :

```
sudo dnf config-manager \  
  --add-repo \  
  https://download.docker.com/linux/fedora/docker-ce.repo
```

Dans mon cas je n'utiliserai que la version stable de docker mais si vous souhaitez activer le repository et les mises à jour des versions Nightly et Test depuis le repository de Docker, alors il faut lancer les commandes suivantes :

Pour la version **Nightly** :

```
sudo dnf config-manager --set-enabled docker-ce-nightly
```

Pour la version **Test** :

```
sudo dnf config-manager --set-enabled docker-ce-test
```

Information

Si vous souhaitez désactiver la version Nightly ou Test, il suffit de remplacer `--set-enabled` par `--set-disabled`.

Une fois le repository Docker ajouté, l'étape suivante est l'installation de Docker :

```
sudo dnf install docker-ce docker-ce-cli containerd.io
```

Une fois votre installation finie, l'étape suivante est de **vérifier que Docker CE est correctement installé** en vérifiant d'abord la version du moteur :

```
sudo docker --version
```

Résultat:

```
Docker version 18.09.6, build 481bc77
```

Maintenant il faut **activer le service docker** en tapant la commande suivante :

```
sudo systemctl start docker
```

Pour activer le service docker automatiquement après un boot, il suffit de lancer la commande suivante :

```
sudo systemctl enable docker
```

Pour finir nos tests de vérifications d'install, on va lancer la commande suivante :

```
sudo docker run hello-world
```

Cette commande va télécharger une image (vous comprendrez ce terme plus tard) de test et l'exécute dans un conteneur. Lorsque le conteneur va s'exécuter, il vous affichera un message d'information et se fermera automatiquement

Résultat:

```
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
1b930d010525: Pull complete
Digest: sha256:0e11c388b664df8a27a901dce21eb89f11d8292f7fca1b3e3c4321bf7897bffe
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/
```

Bonus Linux

Par défaut on ne peut lancer les commandes docker qu'avec l'utilisateur root et les autres utilisateurs ne peuvent y accéder qu'en utilisant le `sudo`.

En ce qui me concerne et pour la suite de ce cours, je vais **autoriser mon compte utilisateur Unix à lancer les commandes docker sans passer par le `sudo`**. Donc à partir de maintenant je ne vais plus utiliser le `sudo` dans mes exemples, sauf si obligation.

Cette manipulation est assez simple à faire, il suffit de créer un **groupe Unix appelé `docker`** et de lui ajouter notre utilisateur.

On va commencer par créer notre groupe Unix avec la commande suivante :

```
sudo groupadd docker
```

Puis on rajoute notre utilisateur au groupe docker

```
sudo usermod -aG docker $USER
```

Ensuite, déconnectez-vous et reconnectez-vous (si ça ne fonctionne pas alors il faut redémarrer complètement votre machine) pour que votre appartenance au groupe soit réévaluée.

Enfin, vérifiez que vous pouvez exécuter la commande docker suivante sans sudo :

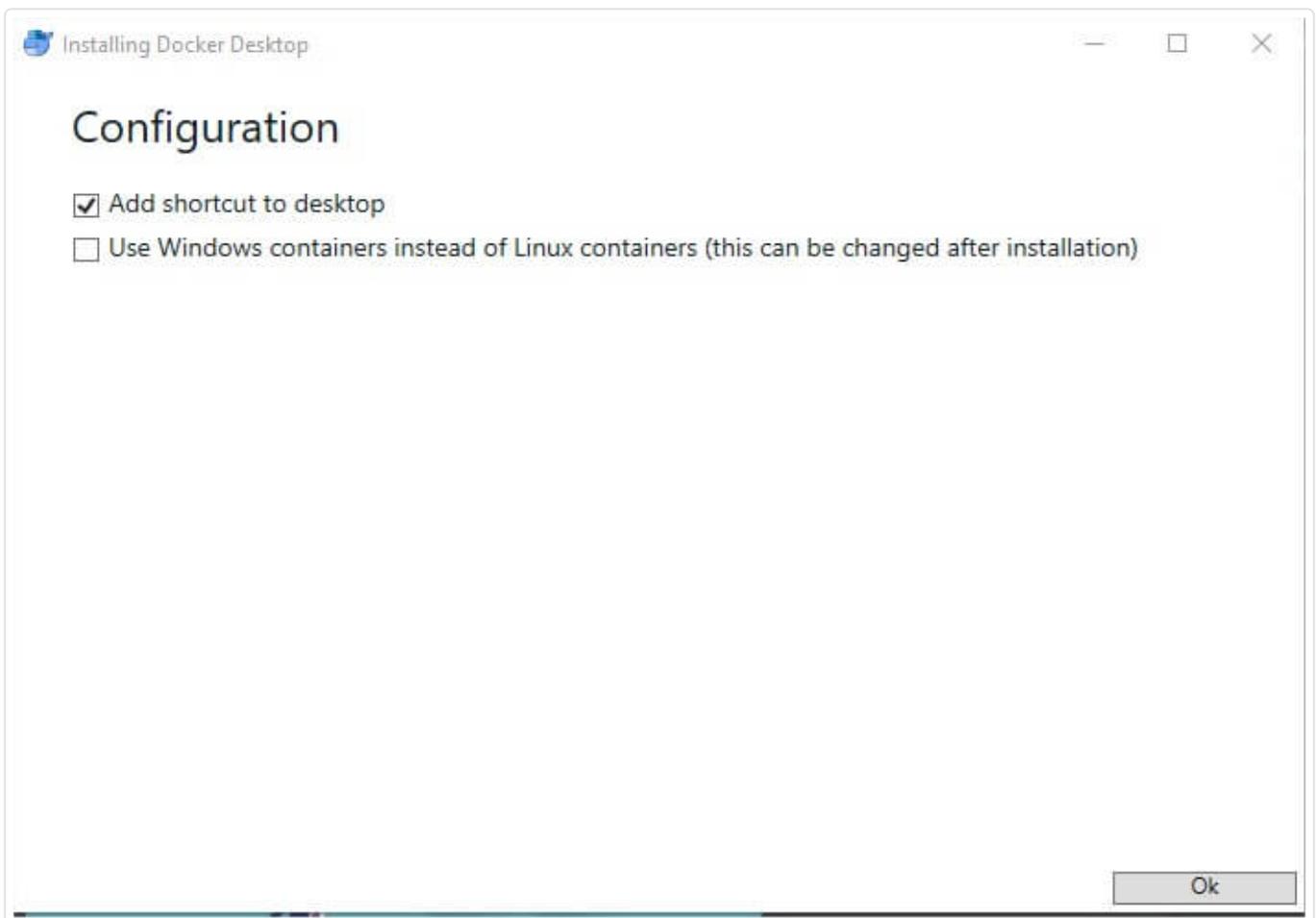
```
docker ps
```

Installation Windows

Voici la **configuration requise pour une installation windows native** :

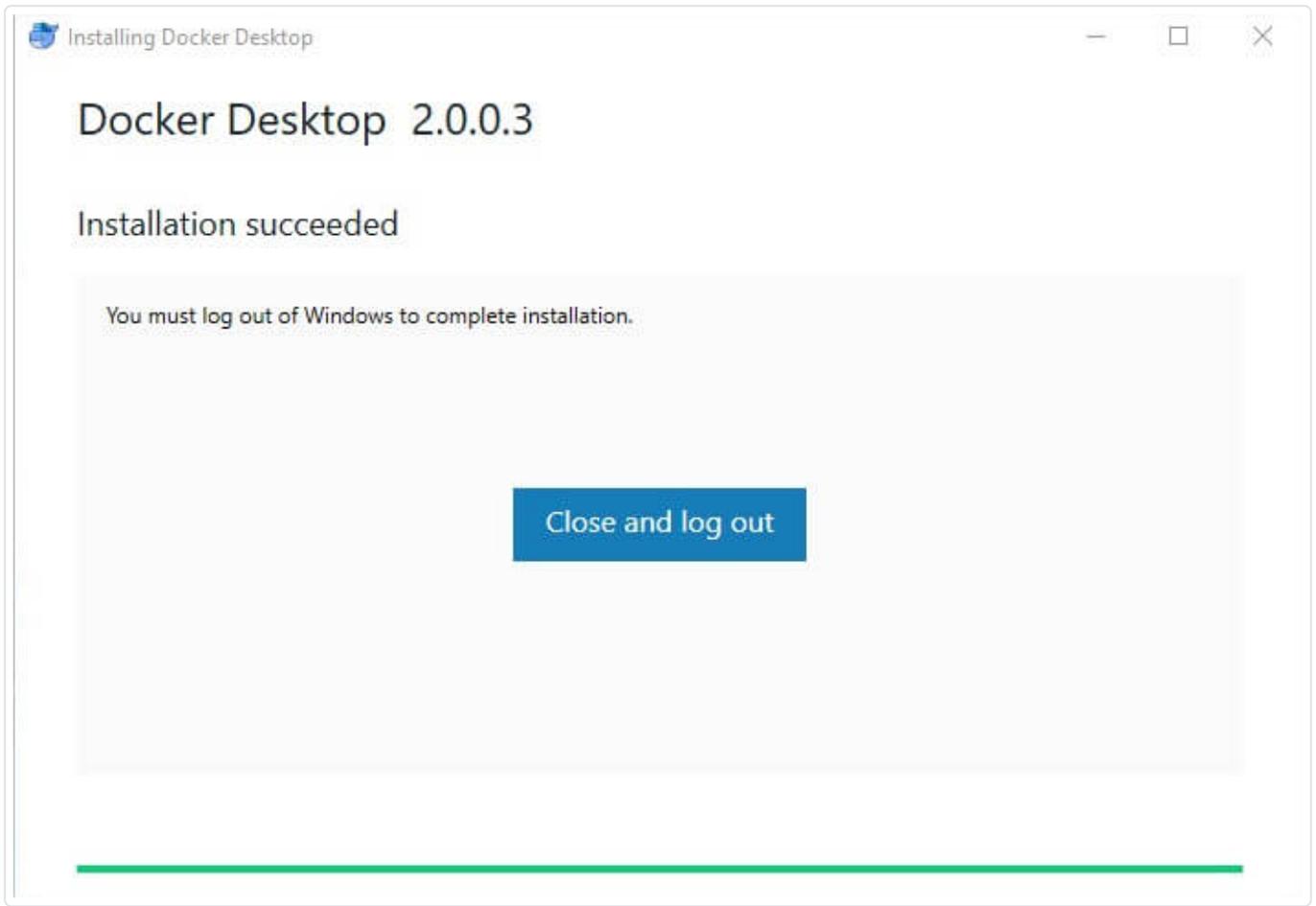
- Windows 10 64 bits: Pro, Entreprise ou Education (version 15063 ou ultérieure).
- La virtualisation est activée dans le BIOS (normalement elle est activée par défaut).
- Au moins 4 Go de RAM

Une fois les conditions satisfaites, rendez-vous et inscrivez sur le Docker Hub en cliquant [ici](#). Une fois inscrit et connecté, cliquez sur le bouton "Get Docker" pour télécharger l'exécutable Docker CE et suivez les instructions ci-dessous :



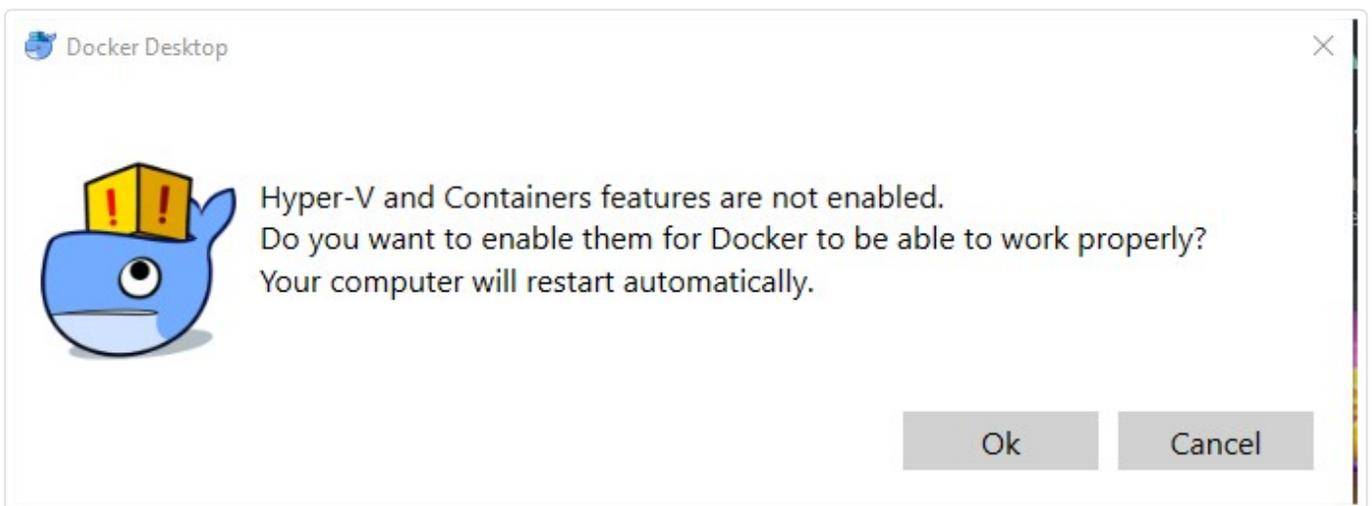
Cliquez sur le bouton "OK".

Une fois votre installation terminée, vous recevrez le message suivant :



Cliquez sur "Close and log out". Vous allez ensuite être automatiquement déconnecté de votre session Windows.

Si vous n'avez pas activé Hyper-V, alors Docker s'en chargera en l'activant automatiquement pour vous :



Cliquez sur "OK" pour activer Hyper-V. Par la suite votre machine va automatiquement se redémarrer à la fin de l'activation d'Hyper-V.

Après votre redémarrage, Vous verrez la fenêtre suivante indiquant que le moteur Docker est bien installé :



Welcome



- Docker Desktop is now up and running!

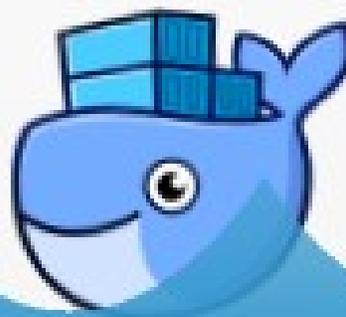
Open your favorite terminal and start typing [Docker commands](#).



Microsoft PowerShell

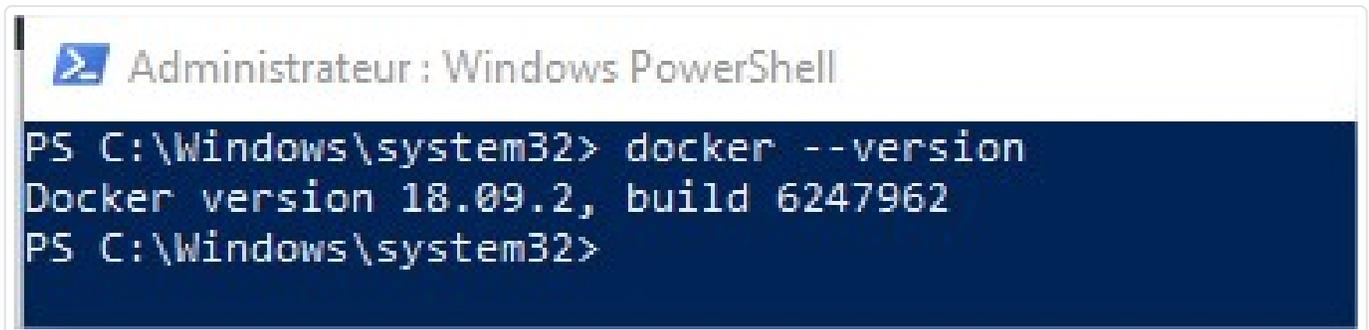


```
> docker ps
```



Login with your Docker ID:

Dernière étape, lancez votre powershell en tant qu'administrateur et exécutez la commande suivante afin de vérifier que votre Docker CE c'est correctement installé :

A screenshot of a Windows PowerShell terminal window titled "Administrateur : Windows PowerShell". The terminal shows the command "docker --version" being executed, resulting in the output "Docker version 18.09.2, build 6247962". The prompt "PS C:\Windows\system32>" is visible before and after the command.

```
Administrateur : Windows PowerShell
PS C:\Windows\system32> docker --version
Docker version 18.09.2, build 6247962
PS C:\Windows\system32>
```

Si **votre système ne remplit pas les conditions requises vous pouvez alors installer Docker Toolbox**, qui utilise Oracle Virtual Box au lieu de Hyper-V en cliquant [ici](#).

Voilà c'est bon, vous pouvez passer au chapitre suivant :).