

Installation - Docker - Kubernetes

1) Installation de Docker.....	1
1.1) Prérequis.....	2
1.2) Installation.....	2
1.3) Autre installation de Docker (Installation d'une version spécifique)	2
1.4) Autre installation de Docker (Installation à partir d'un paquet).....	2
1.5) La mise à niveau (upgrade) Docker CE.....	3
1.6) Vérifications	3
1.7) Désinstallation	3
2) Installation de kubectl	3
2.1) Installation classique	4
2.2) Installation avec snap sur Ubuntu	4
3) Installation de minikube	4
3.1) Les prérequis	4
3.1.1) Pour VirtualBox.....	4
3.1.2) Pour les drivers KVM (paquet docker-machine-driver-kvm-ubuntu16.04).....	5
3.1.3) Ajouter les dépôts suivants	5
3.2) Installation de minikube.....	5
3.3) Exportation des variables d'environnement	5
3.4) Lancement de minikube	5
3.5) Accès à l'interface graphique	7
Les principales commandes (CLI) de Kubernetes :.....	7
Webographie.....	9

1) Installation de Docker

Ici, l'installation proposée concernant Docker est pour Ubuntu. Il suffit d'adapter l'installation pour les autres distributions en se référant au site officiel dont les liens sont les suivants :

<https://docs.docker.com/install/linux/docker-ce/ubuntu/>

<https://docs.docker.com/install/linux/docker-ce/debian/>

<https://docs.docker.com/install/linux/docker-ce/fedora/>

<https://docs.docker.com/install/linux/docker-ce/centos/>

1.1) Prérequis

```
sudo apt-get remove docker docker-engine docker.io
```

```
sudo apt-get update
```

```
sudo apt-get install apt-transport-https ca-certificates curl software-properties-common
```

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

Vérifier que vous avez la clé avec l'empreinte 9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88 en indiquant dans la recherche les 8 derniers caractères.

```
sudo apt-key fingerprint 0EBFCD88
```

Rajouter le dépôt Docker :

Si vous voulez installer builds des répertoires edge ou test, ajouter le répertoire edge ou test en ajoutant le mot edge ou test (ou les deux) après le mot stable.

```
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
```

```
sudo apt-get update
```

1.2) Installation

```
sudo apt-get install docker-ce
```

1.3) Autre installation de Docker (Installation d'une version spécifique)

```
apt-cache madison docker-ce
```

```
docker-ce | 18.03.0~ce-0~ubuntu | https://download.docker.com/linux/ubuntu xenial/stable amd64 Packages
```

Installer une version spécifique par son nom de paquetage complètement qualifié (fully qualified package name), dont le nom du paquetage (docker-ce) “=” nom de la version

Par exemple, docker-ce=18.03.0~ce-0~ubuntu.

```
sudo apt-get install docker-ce=<VERSION>
```

Vérifier que Docker CE est installé correctement en lançant l'image hello-world.

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

1.4) Autre installation de Docker (Installation à partir d'un paquet)

Si vous n'utilisez pas le répertoire de Docker CE, vous pouvez télécharger le fichier .deb de votre distribution et l'installer manuellement. Vous avez besoin de télécharger un nouveau fichier chaque fois que vous voulez upgrader Docker CE.

Aller à l'adresse <https://download.docker.com/linux/ubuntu/dists/>

Choisir la version Ubuntu, aller au [pool/stable/](#) et choisir amd64, armhf, ppc64el, ou s390x.
Télécharger le fichier .deb pour la version de Docker que vous voulez installer.
Pour installer un paquetage edge, changer le mot stable dans l'url pour edge (edge donne les nouvelles features chaque mois).

Installer Docker CE, changer le chemin ci-dessous par le vôtre, celui où vous avez téléchargé le paquetage Docker.

```
sudo dpkg -i /path/to/package.deb
```

Vérifier que Docker CE est installé correctement en exécutant une image hello-world.

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

1.5) La mise à niveau (upgrade) Docker CE

Pour mettre à niveau Docker

```
add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu  
$(lsb_release -cs) stable"
```

Premièrement exécuter **sudo apt-get update**, ensuite suivre les instructions d'installation, choisir la nouvelle version que vous voulez installer.

1.6) Vérifications

```
docker --version
```

```
sudo docker info
```

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

```
sudo docker image ls
```

```
sudo docker container ls
```

all, all in quiet mode

```
sudo docker container ls --all
```

```
sudo docker container ls -aq
```

1.7) Désinstallation

Désinstaller le paquet Docker CE

```
sudo apt-get purge docker-ce
```

Images, containers, volumes, fichiers de configuration personnels sur votre ordinateur ne sont pas automatiquement supprimés.

Pour supprimer toutes les images, containers et volumes:

```
sudo rm -rf /var/lib/docker
```

2) Installation de kubectl

(Se référer aux sites officiels en cas de problèmes)

<https://kubernetes.io/docs/tasks/tools/install-kubectl/#before-you-begin>

2.1) Installation classique

```
sudo apt-get update && sudo apt-get install -y apt-transport-https
```

```
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -
```

```
sudo touch /etc/apt/sources.list.d/kubernetes.list
```

```
echo "deb http://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee -a  
/etc/apt/sources.list.d/kubernetes.list
```

```
sudo apt-get update
```

```
sudo apt-get install -y kubectl
```

2.2) Installation avec snap sur Ubuntu

Si vous êtes sur Ubuntu ou une des autres distributions Linux qui supportent snap package manager, vous pouvez installer avec: `sudo snap install kubectl --classic`

`kubectl version` (pour vérifier que la version installée est suffisamment à jour).

3) Installation de minikube

(Se référer aux sites officiels en cas de problèmes)

<https://kubernetes.io/docs/tasks/tools/install-minikube/>

<https://kubernetes.io/docs/setup/minikube/>

<https://github.com/kubernetes/minikube/blob/master/docs/drivers.md#kvm2-driver>

<https://github.com/kubernetes/minikube/blob/master/docs/drivers.md#kvm-driver>

3.1) Les prérequis

La virtualisation VT-x/AMD-v doit être activée dans le BIOS.

`egrep -c '(vmx|svm)' /proc/cpuinfo` (si différent de zéro, le système supporte la virtualisation)

On utilise soit virtualbox ou kvm.

Installer Docker et VirtualBox ou prendre kvm (ou kvm2) en installant le driver.

3.1.1) Pour VirtualBox

Installation de VirtualBox et `Oracle_VM_VirtualBox_Extension_Pack`.

```
sudo apt install virtualbox
```

```
wget http://download.virtualbox.org/virtualbox/5.0.20/Oracle_VM_VirtualBox_Extension_Pack-5.0.20-106931.vbox-extpack
```

```
sudo VBoxManage extpack install ./Oracle_VM_VirtualBox_Extension_Pack-5.0.20-106931.vbox-extpack
```

Désinstallation d'`Oracle_VM_VirtualBox_Extension_Pack`

```
sudo VBoxManage extpack uninstall "Oracle VM VirtualBox Extension Pack"
```

3.1.2) Pour les drivers KVM (paquet docker-machine-driver-kvm-ubuntu16.04)

Installer les paquets suivants :

```
sudo apt install qemu-kvm libvirt-clients libvirt-daemon-system libvirt-bin
```

Puis,

```
sudo usermod -a -G libvirt $(whoami)
newgrp libvirt
```

Pour les anciennes versions d'ubuntu:

```
sudo usermod -a -G libvirtd $(whoami)
newgrp libvirtd
```

```
curl -LO https://storage.googleapis.com/minikube/releases/latest/docker-machine-driver-kvm2
&& chmod +x docker-machine-driver-kvm2 && sudo mv docker-machine-driver-kvm2
/usr/local/bin/
```

3.1.3) Ajouter les dépôts suivants

```
curl -Lo minikube https://storage.googleapis.com/minikube/releases/latest/minikube-linux-
amd64 && chmod +x minikube && sudo mv minikube /usr/local/bin/
```

```
curl -Lo kubectl https://storage.googleapis.com/kubernetes-release/release/$(curl -s
https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl
&& chmod +x kubectl && sudo mv kubectl /usr/local/bin/
```

3.2) Installation de minikube

Récupérer la dernière version du paquet minikube (ici minikube_0.28-2.deb).

```
sudo dpkg -i minikube_0.28-2.deb
```

3.3) Exportation des variables d'environnement

(Se mettre en administrateur)

```
export MINIKUBE_WANTUPDATENOTIFICATION=false
export MINIKUBE_WANTREPORTERRORPROMPT=false
export MINIKUBE_HOME=$HOME
export CHANGE_MINIKUBE_NONE_USER=true
```

```
mkdir -p $HOME/.kube
touch $HOME/.kube/config
export KUBECONFIG=$HOME/.kube/config
```

3.4) Lancement de minikube

Sans utiliser de driver.

```
sudo minikube start --vm-driver=none (pas sans --vm-driver=none)
```

Avec le driver

```
sudo minikube start --vm-driver kvm2
```

```
sudo minikube start --vm-driver kvm2
```

```
Starting local Kubernetes v1.10.0 cluster...
```

```
Starting VM...
```

```
Downloading Minikube ISO
```

```
160.27 MB / 160.27 MB [=====] 100.00%
```

```
0s
```

```
Getting VM IP address...
```

```
Moving files into cluster...
```

```
Downloading kubeadm v1.10.0
```

```
Downloading kubelet v1.10.0
```

```
Finished Downloading kubelet v1.10.0
```

```
Finished Downloading kubeadm v1.10.0
```

```
Setting up certs...
```

```
Connecting to cluster...
```

```
Setting up kubeconfig...
```

```
Starting cluster components...
```

```
Kubectl is now configured to use the cluster.
```

```
Loading cached images from config file.
```

Afficher l'interface.

```
sudo minikube dashboard
```

```
sudo minikube dashboard
```

```
Opening kubernetes dashboard in default browser...
```

```
Running Firefox as root in a regular user's session is not supported. ($HOME is /home/compute1 which is owned by compute1.)
```

```
Running Firefox as root in a regular user's session is not supported. ($HOME is /home/compute1 which is owned by compute1.)
```

```
Running Firefox as root in a regular user's session is not supported. ($HOME is /home/compute1 which is owned by compute1.)
```

```
/usr/bin/xdg-open: 851: /usr/bin/xdg-open: iceweasel: not found
```

```
/usr/bin/xdg-open: 851: /usr/bin/xdg-open: seamonkey: not found
```

```
/usr/bin/xdg-open: 851: /usr/bin/xdg-open: mozilla: not found
```

```
/usr/bin/xdg-open: 851: /usr/bin/xdg-open: epiphany: not found
```

```
/usr/bin/xdg-open: 851: /usr/bin/xdg-open: konqueror: not found
```

```
/usr/bin/xdg-open: 851: /usr/bin/xdg-open: chromium: not found
```

```
/usr/bin/xdg-open: 851: /usr/bin/xdg-open: chromium-browser: not found
```

```
/usr/bin/xdg-open: 851: /usr/bin/xdg-open: google-chrome: not found
```

```
/usr/bin/xdg-open: 851: /usr/bin/xdg-open: www-browser: not found
```

```
/usr/bin/xdg-open: 851: /usr/bin/xdg-open: links2: not found
```

```
/usr/bin/xdg-open: 851: /usr/bin/xdg-open: elinks: not found
```

```
/usr/bin/xdg-open: 851: /usr/bin/xdg-open: links: not found
/usr/bin/xdg-open: 851: /usr/bin/xdg-open: lynx: not found
/usr/bin/xdg-open: 851: /usr/bin/xdg-open: w3m: not found
xdg-open: no method available for opening 'http://192.168.39.175:30000'
sudo minikube ip
```

sudo minikube service list

```
|-----|-----|-----|
| NAMESPACE | NAME           | URL           |
|-----|-----|-----|
| default   | kubernetes     | No node port  |
| kube-system | kube-dns       | No node port  |
| kube-system | kubernetes-dashboard | http://192.168.39.175:30000 |
|-----|-----|-----|
```

sudo minikube status

```
minikube: Running
cluster: Running
kubectl: Correctly Configured: pointing to minikube-vm at 192.168.39.175
```

Lui donner une version particulière :

```
sudo minikube start --kubernetes-version v1.7.3
```

3.5) Accès à l'interface graphique

sudo minikube dashboard

```
http://IP:30000
```

```
sudo kubectl config current-context
```

```
sudo kubectl cluster-info
```

```
sudo minikube stop
```

```
sudo minikube get-k8s-versions
```

A la suite du redémarrage de la session ou du système :

```
sudo minikube delete
```

```
sudo rm -rf .minikube/
```

```
sudo rm -rf /etc/kubernetes/
```

```
sudo minikube start --vm-driver kvm2
```

```
sudo minikube dashboard
```

Les principales commandes (CLI) de Kubernetes :

```
`annotate` | `kubectl annotate (-f FILENAME \ TYPE NAME \ TYPE/NAME) KEY_1=VAL_1 ...
KEY_N=VAL_N [--overwrite] [--all] [--resource-version=version] [flags]`
```

Ajouter ou mettre à jour les annotations d'une ou plusieurs ressources.

api-versions | `kubectl api-versions [flags]`

Lister les versions d'API disponibles.

apply | `kubectl apply -f FILENAME [flags]`

Appliquer une modification de configuration à une ressource à partir d'un fichier ou à une entrée stdin

attach | `kubectl attach POD -c CONTAINER [-i] [-t] [flags]`

Attacher à un conteneur en cours d'exécution soit pour afficher le flux de sortie ou interagir avec le conteneur (stdin).

autoscale | `kubectl autoscale (-f FILENAME \ TYPE NAME \ TYPE/NAME) [--min=MINPODS] --max=MAXPODS [--cpu-percent=CPU] [flags]`

Mettre à l'échelle automatiquement l'ensemble de pods gérés par un contrôleur de réplication.

cluster-info | `kubectl cluster-info [flags]`

Afficher les informations de point de terminaison sur le master et les services du cluster.

config | `kubectl config SUBCOMMAND [flags]`

Modifie les fichiers kubeconfig. Voir les sous-commandes individuelles pour plus de détails.

create | `kubectl create -f FILENAME [flags]`

Créer une ou plusieurs ressources à partir d'un fichier ou d'un stdin.

delete | `kubectl delete (-f FILENAME \ TYPE [NAME \ /NAME \ -l label \ --all]) [flags]`

Supprimer des ressources d'un fichier stdin ou spécifier des sélecteurs d'étiquette, des noms, des sélecteurs de ressources ou des ressources.

describe | `kubectl describe (-f FILENAME \ TYPE [NAME_PREFIX \ /NAME \ -l label]) [flags]`

Afficher l'état détaillé d'une ou plusieurs ressources.

edit | `kubectl edit (-f FILENAME \ TYPE NAME \ TYPE/NAME) [flags]`

Modifier et mettre à jour la définition d'une ou plusieurs ressources sur le serveur en utilisant l'éditeur par défaut.

exec | `kubectl exec POD [-c CONTAINER] [-i] [-t] [flags] [-- COMMAND [args...]]`

Exécuter une commande sur un conteneur dans un pod.

explain | `kubectl explain [--include-extended-apis=true] [--recursive=false] [flags]`

Obtenir la documentation de diverses ressources. Par exemple, des pods, des nœuds, des services, etc.

expose | `kubectl expose (-f FILENAME \ TYPE NAME \ TYPE/NAME) [--port=port] [--protocol=TCP\UDP] [--target-port=number-or-name] [--name=name] [----external-ip=external-ip-of-service] [--type=type] [flags]`

Exposer un replication controller, un service ou un pod en tant que nouveau service Kubernetes.

get | `kubectl get (-f FILENAME \ TYPE [NAME \ /NAME \ -l label]) [--watch] [--sort-by=FIELD] [--o \ --output]=OUTPUT_FORMAT [flags]`

Énumérer une ou plusieurs ressources.

label | `kubectl label (-f FILENAME \ TYPE NAME \ TYPE/NAME) KEY_1=VAL_1 ...`

`KEY_N=VAL_N [--overwrite] [--all] [--resource-version=version] [flags]`
Ajouter ou mettre à jour les étiquettes d'une ou de plusieurs ressources.

`logs` | `kubectl logs POD [-c CONTAINER] [--follow] [flags]`
Imprimer les logs d'un conteneur dans un pod.

`patch` | `kubectl patch (-f FILENAME \ TYPE NAME \ TYPE/NAME) --patch PATCH [flags]`
Mettre à jour un ou plusieurs champs d'une ressource à l'aide du processus patch de fusion stratégique.

`port-forward` | `kubectl port-forward POD [LOCAL_PORT:]REMOTE_PORT [...[LOCAL_PORT_N:]REMOTE_PORT_N] [flags]`
Transférer un ou plusieurs ports locaux vers un pod.

`proxy` | `kubectl proxy [--port=PORT] [--www=static-dir] [--www-prefix=prefix] [--api-prefix=prefix] [flags]`
Exécuter un proxy sur l'API-server de Kubernetes.

`replace` | `kubectl replace -f FILENAME`
Remplacer une ressource d'un fichier ou de l'entrée stdin.

`rolling-update` | `kubectl rolling-update OLD_CONTROLLER_NAME ([NEW_CONTROLLER_NAME] --image=NEW_CONTAINER_IMAGE \ -f NEW_CONTROLLER_SPEC) [flags]`
Effectuer une mise à jour continue en remplaçant progressivement le replication controller spécifié et ses pods.

`run` | `kubectl run NAME --image=image [--env="key=value"] [--port=port] [--replicas=replicas] [--dry-run=bool] [--overrides=inline-json] [flags]`
Exécuter une image spécifiée sur le cluster.

`scale` | `kubectl scale (-f FILENAME \ TYPE NAME \ TYPE/NAME) --replicas=COUNT [--resource-version=version] [--current-replicas=count] [flags]`
Mettre à jour la taille du replication controller spécifié.

`stop` | `kubectl stop`
Obsolète : à la place, voir `kubectl delete`.

`version` | `kubectl version [--client] [flags]`
Afficher la version de Kubernetes exécutée sur le client et le serveur.

Webographie

<https://docs.docker.com/install/linux/docker-ce/ubuntu/>

[https://fr.wikipedia.org/wiki/Docker_\(logiciel\)](https://fr.wikipedia.org/wiki/Docker_(logiciel))

<https://www.lebigdata.fr/docker-definition>

<https://coreos.com/rkt/>

<http://cri-o.io/>

<https://katacontainers.io/>

<https://kubernetes.io/docs/home/?path=users&persona=app-developer&level=foundational>

<https://kubernetes.io/docs/tutorials/>

<https://kubernetes.io/docs/tutorials/hello-minikube/>

<https://www.tutorialspoint.com/kubernetes/index.htm>

<https://www.youtube.com/watch?v=QDvxQWKY17s>