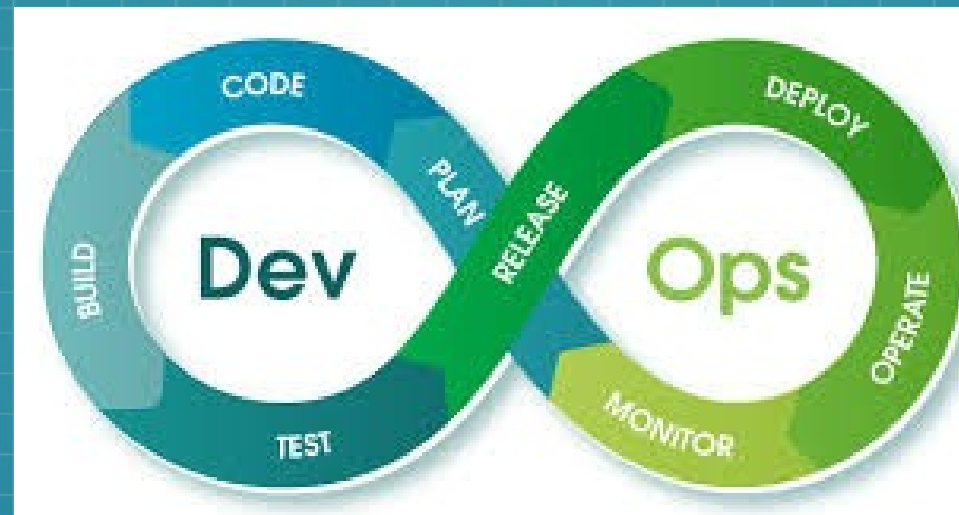
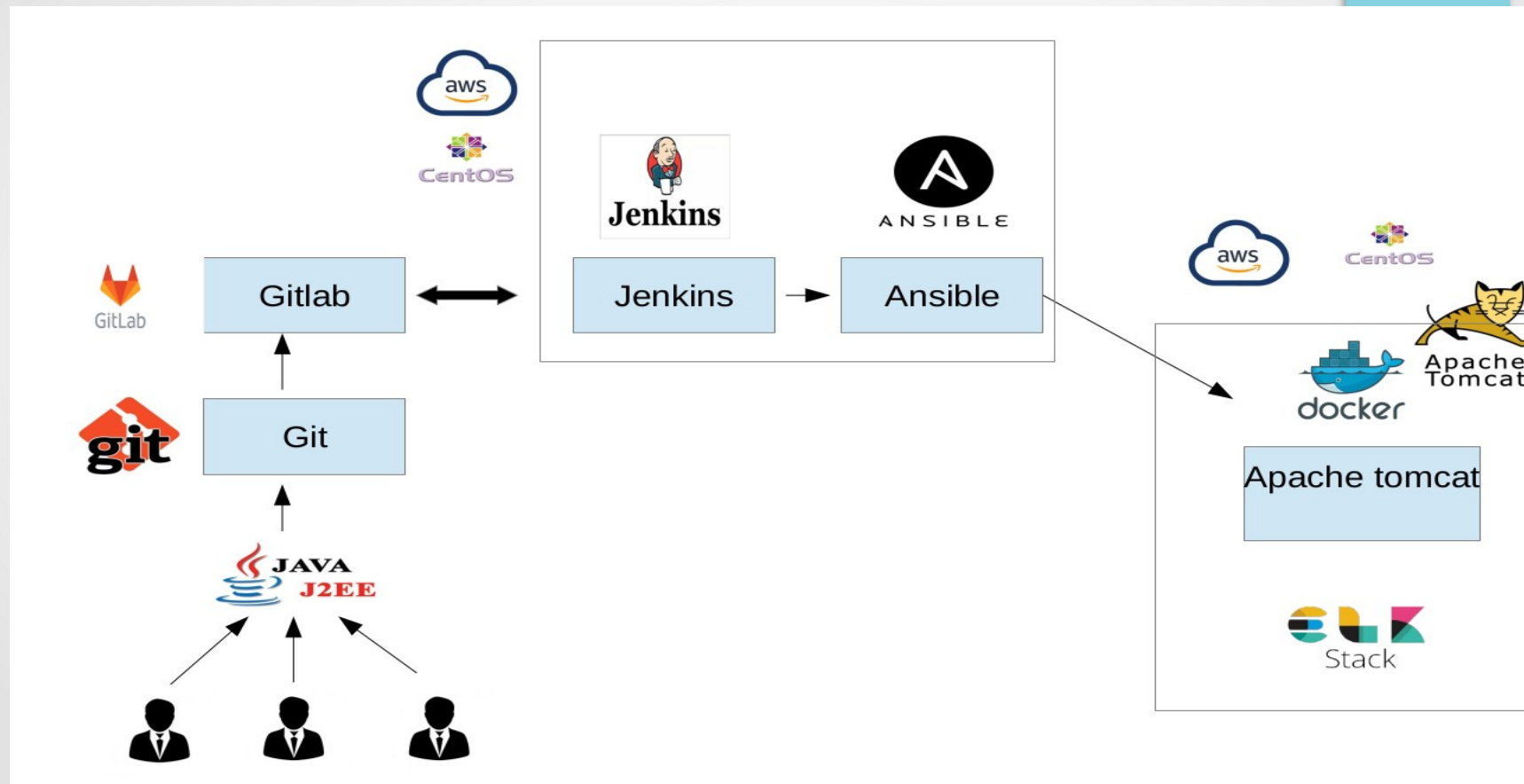


Formation Devops par la pratique



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Objectif : Réaliser une pipeline Git-Gitlab-Jenkins-Ansible-Docker



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Objectif : Réaliser une pipeline Git-Gitlab-Jenkins-Ansible-Docker

Pour atteindre l'objectif, on va décomposer le pipeline en plusieurs projets pour bien comprendre comment ça passe.

Projets :

- 1)- Déploiement Manuelle
- 2)- Git-Gitlab-Jenkins
- 3)- Git-Gitlab-Jenkins-Tomcat
- 4)- Git-Gitlab-Jenkins-Ansible-Tomcat
- 5)- Git-Gitlab-Jenkins-Ansible-Docker
- 6)- ELK Stack
- 7)- kubernetes
- 8)- Prometheus-Grafana



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Parcours de Formation DEVOPS

- Révision commande Linux
- Git, Gitlab
- AWS (EC2)
- Jenkins
- Ansible
- Docker
- ELK stack
- Kubernetes
- Prometheus-Grafana



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Vagrant



Vagrant est un outil de ligne de commande pour créer et gérer des environnements de machines virtuelles. Par défaut, Vagrant peut provisionner des machines au-dessus de VirtualBox, Hyper-V et Docker. D'autres fournisseurs tels que Libvirt (KVM), VMware et AWS peuvent être installés via le système de plugins Vagrant.

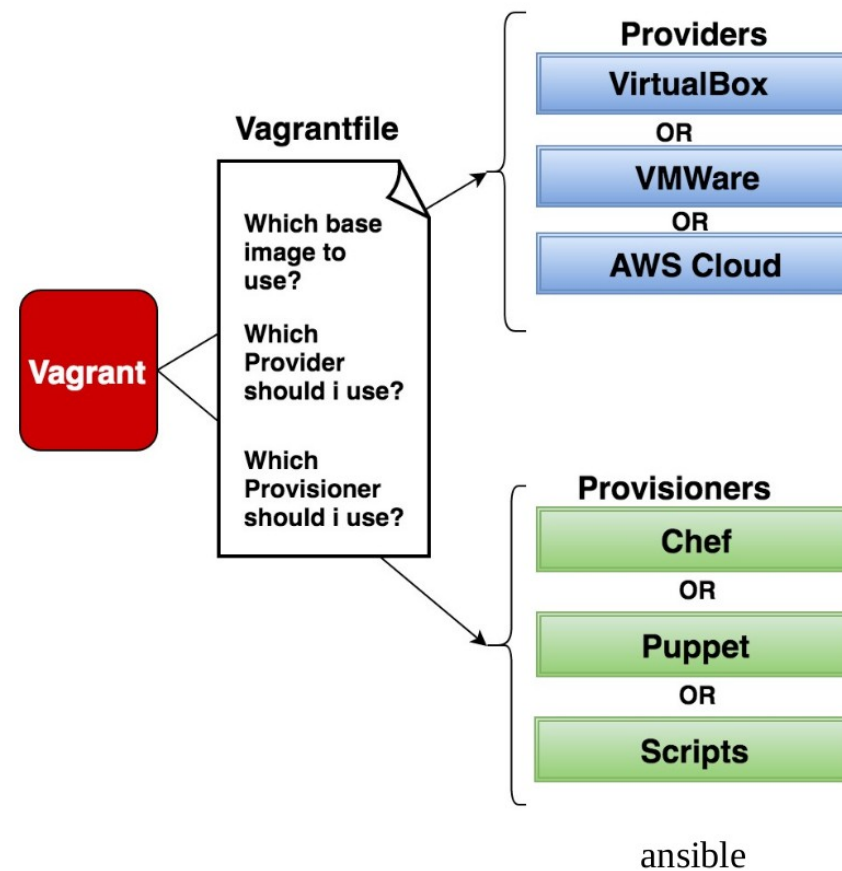
Vagrant est généralement utilisé par les développeurs pour mettre en place un environnement de développement qui correspond à l'environnement de production.

Les boîtes Vagrant sont configurées à l'aide de la configuration spéciale Vagrantfile écrite en Ruby, mais peuvent inclure des scripts de configuration supplémentaires écrits en bash, Chef ou Puppet. Les développeurs d'une équipe peuvent télécharger les fichiers de configuration Vagrant à partir d'une source commune et recréer le même environnement de développement localement.



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Vagrant



Vagrant

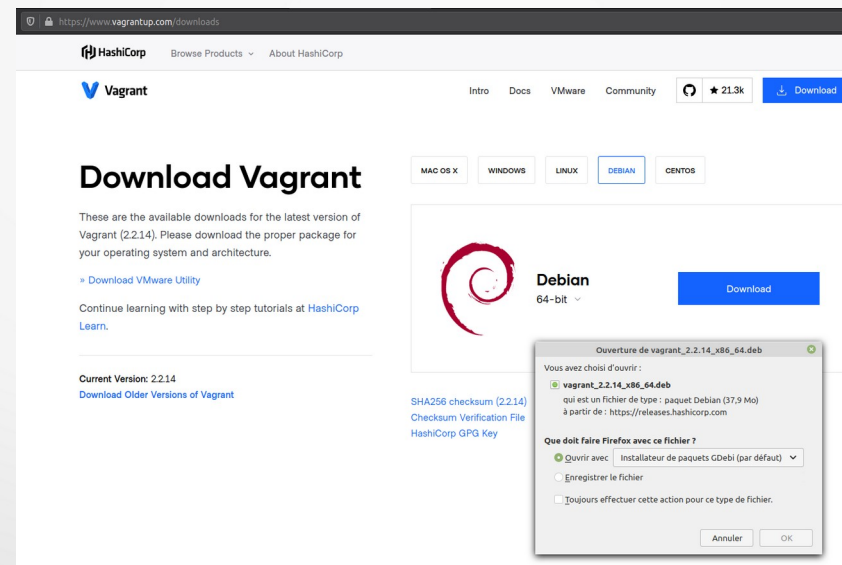
Installez Vagrant sur Ubuntu/Linux-mint

Dans ce tutoriel, nous allons vous montrer comment installer Vagrant sur une machine Ubuntu. Nous utiliserons le fournisseur VirtualBox, qui est le fournisseur par défaut de Vagrant.

Pour installer Vagrant sur votre système Ubuntu, on peut utiliser Le paquet Vagrant, qui est disponible dans les référentiels d'Ubuntu a l'aide du commande :

\$ sudo apt install vagrant.

Ou on téléchargerons et installerons la dernière version de Vagrant à partir du site officiel de Vagrant. <https://www.vagrantup.com/downloads>



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Vagrant

Téléchargez le package Vagrant à l'aide de la commande curl suivante :

```
$ curl -O https://releases.hashicorp.com/vagrant/2.2.14/vagrant\_2.2.14\_x86\_64.deb
```

Une fois le .deb fichier téléchargé, installez-le en tapant:

```
$ sudo apt install ./vagrant_2.2.14_x86_64.deb
```

Pour vérifier que l'installation a réussi, exécutez la commande suivante qui imprime la version de Vagrant:

```
$ vagrant --version
```

```
bilel@bilel-G3-3590:~$ vagrant --version  
Vagrant 2.2.6  
bilel@bilel-G3-3590:~$
```

Maintenant que Vagrant est installé sur votre système Ubuntu, créons un environnement de développement.

La première étape consiste à créer un répertoire qui sera le répertoire racine du projet et contiendra le fichier Vagrantfile. Vagrantfile est un fichier Ruby qui décrit comment configurer et provisionner la machine virtuelle.

Créez le répertoire du projet et accédez-y avec:

```
bilel@bilel-G3-3590:~$ mkdir vagrant_projet1  
bilel@bilel-G3-3590:~$ cd vagrant_projet1/  
bilel@bilel-G3-3590:~/vagrant_projet1$ ls  
bilel@bilel-G3-3590:~/vagrant_projet1$
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuelle de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Vagrant

Ensuite, initialisez un nouveau Vagrantfile à l'aide de la commande **vagrant init** et spécifiez la box que vous souhaitez utiliser.

```
bilel@bilel-G3-3590:~$ mkdir vagrant_projet1
bilel@bilel-G3-3590:~$ cd vagrant_projet1/
bilel@bilel-G3-3590:~/vagrant_projet1$ ls
bilel@bilel-G3-3590:~/vagrant_projet1$ vagrant init
A `Vagrantfile` has been placed in this directory. You are now
ready to `vagrant up` your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
`vagrantup.com` for more information on using Vagrant.
bilel@bilel-G3-3590:~/vagrant_projet1$ ls
Vagrantfile
bilel@bilel-G3-3590:~/vagrant_projet1$
```

Vous pouvez ouvrir le Vagrantfile, lire les commentaires et faire des ajustements en fonction de vos besoins.

```
bilel@bilel-G3-3590:~/vagrant_projet1$ cat Vagrantfile
# -*- mode: ruby -*-
# vi: set ft=ruby :

# All Vagrant configuration is done below. The "2" in Vagrant.configure
# configures the configuration version (we support older styles for
# backwards compatibility). Please don't change it unless you know what
# you're doing.
Vagrant.configure("2") do |config|
  # The most common configuration options are documented and commented below.
  # For a complete reference, please see the online documentation at
  # https://docs.vagrantup.com.

  # Every Vagrant development environment requires a box. You can search for
  # boxes at https://vagrantcloud.com/search.
  config.vm.box = "base"

  # Disable automatic box update checking. If you disable this, then
  # boxes will only be checked for updates when the user runs
  # `vagrant box outdated`. This is not recommended.
  # config.vm.box_check_update = false
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuelle de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Vagrant

initialisez un nouveau Vagrantfile sans commentaire

```
bilel@bilel-63-3590:~/vagrant_projet1$ ls
bilel@bilel-63-3590:~/vagrant_projet1$ vagrant init --m
A `Vagrantfile` has been placed in this directory. You are now
ready to `vagrant up` your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
`vagrantup.com` for more information on using Vagrant.
bilel@bilel-63-3590:~/vagrant_projet1$ ls
Vagrantfile
bilel@bilel-63-3590:~/vagrant_projet1$ cat Vagrantfile
Vagrant.configure("2") do |config|
  config.vm.box = "base"
end
bilel@bilel-63-3590:~/vagrant_projet1$
```

Les box sont le format de package pour les environnements Vagrant et sont spécifiques au fournisseur. Vous pouvez trouver une liste des box Vagrant disponibles au public sur la page du catalogue des box Vagrant . <https://app.vagrantup.com/boxes/search>



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Va



Search

Pricing

Vagrant

Help

Create an Account

Sign In






Discover Vagrant Boxes

Search for boxes by operating system, included software, architecture and more



Provider any virtualbox vmware libvirt more

Sort by Downloads Recently Created Recently Updated

	ubuntu/trusty64 20190514.0.0 Official Ubuntu Server 14.04 LTS (Trusty Tahr) builds (End of standard support)	virtualbox	Downloads 30,558,436	Released about 1 year ago
	laravel/homestead 10.1.1 Official Laravel local development box.	hyperv parallels virtualbox vmware and 1 more providers	Downloads 14,231,014	Released 3 months ago
	hashicorp/precise64 1.1.0 A standard Ubuntu 12.04 LTS 64-bit box.	hyperv virtualbox vmware_fusion	Downloads 6,780,779	Released almost 7 years ago
	centos/7 2004.01 CentOS Linux 7 x86_64 Vagrant Box	hyperv libvirt virtualbox vmware and 3 more providers	Downloads 5,348,408	Released 8 months ago
	ubuntu/xenial64 20210114.0.0 Official Ubuntu 16.04 LTS (Xenial Xerus) builds	virtualbox	Downloads 3,445,185	Released 5 days ago



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Vagrant

Dans cet exemple, nous utiliserons la box centos/7. Exécutez la commande suivante pour initialiser un nouveau Vagrantfile: **\$ vagrant init centos/7**

```
bilel@bilel-G3-3590:~/vagrant_projet1$ vagrant init --m centos/7
A `Vagrantfile` has been placed in this directory. You are now
ready to `vagrant up` your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
`vagrantup.com` for more information on using Vagrant.
bilel@bilel-G3-3590:~/vagrant_projet1$ ls
Vagrantfile
bilel@bilel-G3-3590:~/vagrant_projet1$ cat Vagrantfile
Vagrant.configure("2") do |config|
  config.vm.box = "centos/7"
end
```

Exécutez la commande **vagrant up** pour créer et configurer la machine virtuelle comme spécifié dans le Vagrantfile:



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Vagrant

```
bilel@bilel-63-3590:~/vagrant_projet1$ vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Box 'centos/7' could not be found. Attempting to find and install...
    default: Box Provider: virtualbox
    default: Box Version: >= 0
==> default: Loading metadata for box 'centos/7'
    default: URL: https://vagrantcloud.com/centos/7
==> default: Adding box 'centos/7' (v2004.01) for provider: virtualbox
    default: Downloading: https://vagrantcloud.com/centos/boxes/7/versions/2004.01/providers/virtualbox.box
    default: Download redirected to host: cloud.centos.org
    default: Calculating and comparing box checksum...
==> default: Successfully added box 'centos/7' (v2004.01) for 'virtualbox'!
==> default: Importing base box 'centos/7'...
==> default: Matching MAC address for NAT networking...
==> default: Checking if box 'centos/7' version '2004.01' is up to date...
==> default: Setting the name of the VM: vagrant_projet1_default_1611055639519_27971
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
    default: Adapter 1: nat
==> default: Forwarding ports...
    default: 22 (guest) => 2222 (host) (adapter 1)
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
    default: SSH address: 127.0.0.1:2222
    default: SSH username: vagrant
    default: SSH auth method: private key
    default:
    default: Vagrant insecure key detected. Vagrant will automatically replace
    default: this with a newly generated keypair for better security.
    default:
    default: Inserting generated public key within guest...
    default: Removing insecure key from the guest if it's present...
    default: Key inserted! Disconnecting and reconnecting using new SSH key...
==> default: Machine booted and ready!
==> default: Checking for guest additions in VM...
    default: No guest additions were detected on the base box for this VM! Guest
    default: additions are required for forwarded ports, shared folders, host only
    default: networking, and more. If SSH fails on this machine, please install
    default: the guest additions and repackage the box to continue.
    default:
    default: This is not an error message; everything may continue to work properly,
    default: in which case you may ignore this message.
==> default: Rsyncing folder: /home/bilel/vagrant_projet1/ => /vagrant
bilel@bilel-63-3590:~/vagrant_projet1$
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuelle de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Vagrant



Pour accéder ssh a la machine virtuelle, exécutez: **\$ vagrant ssh**

```
bilel@bilel-63-3590:~/vagrant_projet1$ vagrant status
Current machine states:

default                                running (virtualbox)

The VM is running. To stop this VM, you can run `vagrant halt` to
shut it down forcefully, or you can run `vagrant suspend` to simply
suspend the virtual machine. In either case, to restart it again,
simply run `vagrant up`.
bilel@bilel-63-3590:~/vagrant_projet1$ vagrant ssh
Last login: Tue Jan 19 11:45:27 2021 from 10.0.2.2
[vagrant@localhost ~]$ exit
logout
Connection to 127.0.0.1 closed.
bilel@bilel-63-3590:~/vagrant_projet1$
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Vagrant

Principaux commandes Vagrant

```
bilel@bilel-63-3590:~/vagrant_projet1$ vagrant
Usage: vagrant [options] <command> [<args>]

    -v, --version          Print the version and exit.
    -h, --help             Print this help.

Common commands:
    box                    manages boxes: installation, removal, etc.
    cloud                  manages everything related to Vagrant Cloud
    destroy                stops and deletes all traces of the vagrant machine
    global-status          outputs status Vagrant environments for this user
    halt                  stops the vagrant machine
    help                  shows the help for a subcommand
    init                  initializes a new Vagrant environment by creating a Vagrantfile
    login
    package               packages a running vagrant environment into a box
    plugin                manages plugins: install, uninstall, update, etc.
    port                  displays information about guest port mappings
    powershell            connects to machine via powershell remoting
    provision             provisions the vagrant machine
    push                  deploys code in this environment to a configured destination
    rdp                   connects to machine via RDP
    reload                restarts vagrant machine, loads new Vagrantfile configuration
    resume                resume a suspended vagrant machine
    snapshot              manages snapshots: saving, restoring, etc.
    ssh                   connects to machine via SSH
    ssh-config            outputs OpenSSH valid configuration to connect to the machine
    status                outputs status of the vagrant machine
    suspend              suspends the machine
    up                   starts and provisions the vagrant environment
    upload                upload to machine via communicator
    validate              validates the Vagrantfile
    version               prints current and latest Vagrant version
    winrm                 executes commands on a machine via WinRM
    winrm-config          outputs WinRM configuration to connect to the machine

For help on any individual command run `vagrant COMMAND -h`

Additional subcommands are available, but are either more advanced
or not commonly used. To see all subcommands, run the command
`vagrant list-commands`.

bilel@bilel-63-3590:~/vagrant_projet1$
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuelle de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Vagrant

```
bilel@bilel-G3-3590:~/vagrant_projet1$ vagrant ssh-config
Host default
  HostName 127.0.0.1
  User vagrant
  Port 2222
  UserKnownHostsFile /dev/null
  StrictHostKeyChecking no
  PasswordAuthentication no
  IdentityFile /home/bilel/vagrant_projet1/.vagrant/machines/default/virtualbox/private_key
  IdentitiesOnly yes
  LogLevel FATAL
bilel@bilel-G3-3590:~/vagrant_projet1$
```

Quand on tape vagrant ssh c'est comme : `ssh vagrant@localhost -p 2222 -i /home/bilel/vagrant_projet1/.vagrant/machines/default/virtualbox/private_key`
Password = vagrant

```
bilel@bilel-G3-3590:~/vagrant_projet1$ ssh vagrant@localhost -p 2222 -i .vagrant/machines/default/virtualbox/private_key
Last login: Tue Jan 19 11:56:03 2021 from 10.0.2.2
[vagrant@localhost ~]$
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Vagrant

Vous pouvez arrêter la machine virtuelle avec la commande suivante: **\$ vagrant halt**

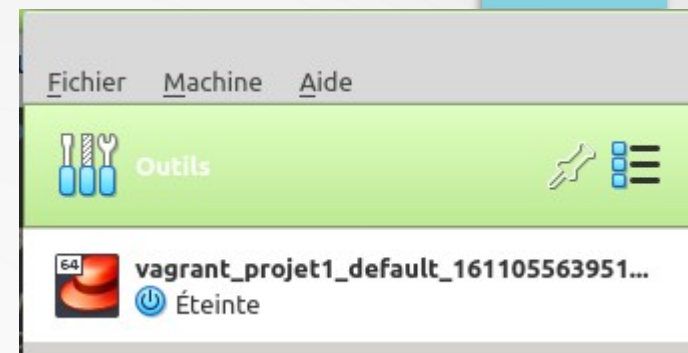
```
bilel@bilel-G3-3590:~/vagrant_projet1$ vagrant status
Current machine states:

default                        running (virtualbox)

The VM is running. To stop this VM, you can run `vagrant halt` to
shut it down forcefully, or you can run `vagrant suspend` to simply
suspend the virtual machine. In either case, to restart it again,
simply run `vagrant up`.
bilel@bilel-G3-3590:~/vagrant_projet1$ vagrant halt
==> default: Attempting graceful shutdown of VM...
bilel@bilel-G3-3590:~/vagrant_projet1$ vagrant status
Current machine states:

default                        poweroff (virtualbox)

The VM is powered off. To restart the VM, simply run `vagrant up`
bilel@bilel-G3-3590:~/vagrant_projet1$
```

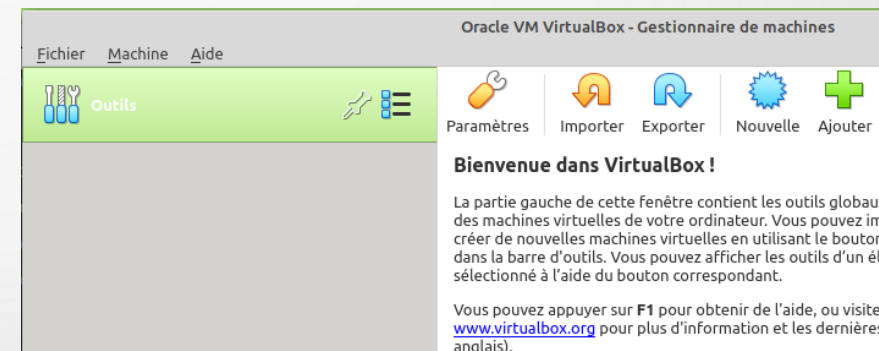


La commande suivante arrête la machine si elle est en cours d'exécution et détruit toutes les ressources créées lors de la création de la machine: **\$ vagrant destroy**

```
bilel@bilel-G3-3590:~/vagrant_projet1$ vagrant destroy
default: Are you sure you want to destroy the 'default' VM? [y/N] y
==> default: Destroying VM and associated drives...
bilel@bilel-G3-3590:~/vagrant_projet1$ vagrant status
Current machine states:

default                        not created (virtualbox)

The environment has not yet been created. Run `vagrant up` to
create the environment. If a machine is not created, only the
default provider will be shown. So if a provider is not listed,
then the machine is not created for that environment.
bilel@bilel-G3-3590:~/vagrant_projet1$
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Vagrant

Vagrant définit plusieurs machines

```
bilel@bilel-G3-3590:~/vagrant_projet1$ ls
Vagrantfile
bilel@bilel-G3-3590:~/vagrant_projet1$ cat Vagrantfile
Vagrant.configure("2") do |config|

  config.vm.define "web" do |web|
    web.vm.box = "generic/alpine38"
  end

  config.vm.define "db" do |db|
    db.vm.box = "generic/alpine38"
  end

end
bilel@bilel-G3-3590:~/vagrant_projet1$ vagrant up
Bringing machine 'web' up with 'virtualbox' provider...
Bringing machine 'db' up with 'virtualbox' provider...
==> web: Box 'generic/alpine38' could not be found. Attempting to find and install...
web: Box Provider: virtualbox
web: Box Version: >= 0
==> web: Loading metadata for box 'generic/alpine38'
web: URL: https://vagrantcloud.com/generic/alpine38
==> web: Adding box 'generic/alpine38' (v3.1.22) for provider: virtualbox
web: Downloading: https://vagrantcloud.com/generic/boxes/alpine38/versions/3.1.22/providers/virtualbox.box
web: Download redirected to host: vagrantcloud-files-production.s3.amazonaws.com
web: Calculating and comparing box checksum...
==> web: Successfully added box 'generic/alpine38' (v3.1.22) for 'virtualbox'!
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuelle de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Vagrant

```
bilel@bilel-G3-3590:~/vagrant_projet1$ vagrant status
Current machine states:
```

```
web                running (virtualbox)
db                 running (virtualbox)
```

This environment represents multiple VMs. The VMs are all listed above with their current state. For more information about a specific VM, run `vagrant status NAME`.

```
bilel@bilel-G3-3590:~/vagrant_projet1$
```

On peut gérer chaque vm avec son nom

```
bilel@bilel-G3-3590:~/vagrant_projet1$ vagrant status web
Current machine states:
```

```
web                running (virtualbox)
```

The VM is running. To stop this VM, you can run `vagrant halt` to shut it down forcefully, or you can run `vagrant suspend` to simply suspend the virtual machine. In either case, to restart it again, simply run `vagrant up`.

```
bilel@bilel-G3-3590:~/vagrant_projet1$ vagrant status db
Current machine states:
```

```
db                 running (virtualbox)
```

The VM is running. To stop this VM, you can run `vagrant halt` to shut it down forcefully, or you can run `vagrant suspend` to simply suspend the virtual machine. In either case, to restart it again, simply run `vagrant up`.

```
bilel@bilel-G3-3590:~/vagrant_projet1$
```

Fichier Machine Aide



 **vagrant_projet1_web_1611058261572_4...**
➔ En Fonction

 **vagrant_projet1_db_16110582866...**
➔ En Fonction



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Vagrant

```
bilel@bilel-G3-3590:~/vagrant_projet1$ vagrant halt web
==> web: Attempting graceful shutdown of VM...
bilel@bilel-G3-3590:~/vagrant_projet1$ vagrant status
Current machine states:

web                poweroff (virtualbox)
db                 running (virtualbox)

This environment represents multiple VMs. The VMs are all listed
above with their current state. For more information about a specific
VM, run `vagrant status NAME`.
bilel@bilel-G3-3590:~/vagrant_projet1$ vagrant ssh db
alpine38:~$ uname -a
Linux alpine38.localdomain 4.14.167-0-virt #1-Alpine SMP Thu Jan 23 10:58:18 UTC 2020 x86_64 Linux
alpine38:~$ exit
logout
Connection to 127.0.0.1 closed.
bilel@bilel-G3-3590:~/vagrant_projet1$
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Vagrant

Vous pouvez sauvegarder et partager votre box avec la commande suivante :
\$ vagrant package --output package.box

```
bilel@bilel-G3-3590:~/vagrant_projet1$ vagrant init --m
A `Vagrantfile` has been placed in this directory. You are now
ready to `vagrant up` your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
`vagrantup.com` for more information on using Vagrant.
bilel@bilel-G3-3590:~/vagrant_projet1$ ls
Vagrantfile
bilel@bilel-G3-3590:~/vagrant_projet1$ nano Vagrantfile
bilel@bilel-G3-3590:~/vagrant_projet1$ cat Vagrantfile
Vagrant.configure("2") do |config|
  config.vm.box = "generic/alpine38"
end
bilel@bilel-G3-3590:~/vagrant_projet1$ vagrant up
```

```
bilel@bilel-G3-3590:~/vagrant_projet1$ vagrant status
Current machine states:

default                                running (virtualbox)

The VM is running. To stop this VM, you can run `vagrant halt` to
shut it down forcefully, or you can run `vagrant suspend` to simply
suspend the virtual machine. In either case, to restart it again,
simply run `vagrant up`.
bilel@bilel-G3-3590:~/vagrant_projet1$ vagrant ssh
alpine38:~$ pwd
/home/vagrant
alpine38:~$ touch bilel.txt
alpine38:~$ ls
bilel.txt
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Vagrant

```
bilel@bilel-G3-3590:~/vagrant_projet1$ vagrant package --output mybox.box
==> default: Attempting graceful shutdown of VM...
==> default: Clearing any previously set forwarded ports...
==> default: Exporting VM...
==> default: Compressing package to: /home/bilel/vagrant_projet1/mybox.box
bilel@bilel-G3-3590:~/vagrant_projet1$ ls
mybox.box  Vagrantfile
bilel@bilel-G3-3590:~/vagrant_projet1$ vagrant destroy
    default: Are you sure you want to destroy the 'default' VM? [y/N] y
==> default: Destroying VM and associated drives...
bilel@bilel-G3-3590:~/vagrant_projet1$ ls
mybox.box  Vagrantfile
bilel@bilel-G3-3590:~/vagrant_projet1$ rm Vagrantfile
bilel@bilel-G3-3590:~/vagrant_projet1$ ls
mybox.box
bilel@bilel-G3-3590:~/vagrant_projet1$
```

Comment utilisé (importer) un vagrant box ??
vagrant box add package.box --name



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Vagrant

```
bilel@bilel-G3-3590:~/vagrant_projet1$ ls
mybox.box
bilel@bilel-G3-3590:~/vagrant_projet1$ vagrant box add mybox.box --name web_box
==> box: Box file was not detected as metadata. Adding it directly...
==> box: Adding box 'web_box' (v0) for provider:
    box: Unpacking necessary files from: file:///home/bilel/vagrant_projet1/mybox.box
==> box: Successfully added box 'web_box' (v0) for 'virtualbox'!
bilel@bilel-G3-3590:~/vagrant_projet1$ vagrant init --m web_box
A `Vagrantfile` has been placed in this directory. You are now
ready to `vagrant up` your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
`vagrantup.com` for more information on using Vagrant.
bilel@bilel-G3-3590:~/vagrant_projet1$ ls
mybox.box  Vagrantfile
bilel@bilel-G3-3590:~/vagrant_projet1$ cat Vagrantfile
Vagrant.configure("2") do |config|
  config.vm.box = "web_box"
end
bilel@bilel-G3-3590:~/vagrant_projet1$ vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'web_box'...
==> default: Matching MAC address for NAT networking...
==> default: Setting the name of the VM: vagrant_projet1_default_1611068332889_77235
```

```
==> default: Mounting shared folders...
    default: /vagrant => /home/bilel/vagrant_projet1
bilel@bilel-G3-3590:~/vagrant_projet1$ vagrant ssh
alpine38:~$ ls
bilel.txt
alpine38:~$
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Vagrant

comment utiliser Vagrant pour créer un conteneur Docker

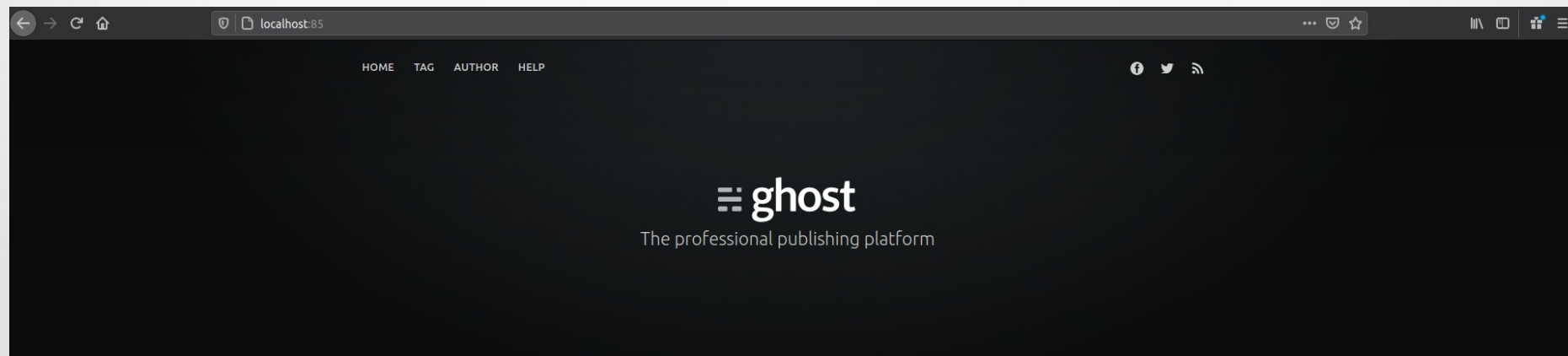
```
bilel@bilel-G3-3590:~/vagrant_projet1$ ls
bilel@bilel-G3-3590:~/vagrant_projet1$ vagrant init --m
A 'Vagrantfile' has been placed in this directory. You are now
ready to 'vagrant up' your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
'vagrantup.com' for more information on using Vagrant.
bilel@bilel-G3-3590:~/vagrant_projet1$ ls
Vagrantfile
bilel@bilel-G3-3590:~/vagrant_projet1$ nano Vagrantfile
bilel@bilel-G3-3590:~/vagrant_projet1$ cat Vagrantfile
Vagrant.configure("2") do |config|
  config.vm.provider "docker" do |container1|
    container1.image = "ghost"
    container1.ports = ["85:2368"]
  end
end
bilel@bilel-G3-3590:~/vagrant_projet1$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
bilel@bilel-G3-3590:~/vagrant_projet1$ vagrant up
Bringing machine 'default' up with 'docker' provider...
/usr/share/rubygems-integration/all/gems/vagrant-2.2.6/lib/vagrant/action/warden.rb:113: warning: Using the last argument as keyw
ord parameters is deprecated; maybe ** should be added to the call
/usr/share/rubygems-integration/all/gems/vagrant-2.2.6/lib/vagrant/action/builtin/message.rb:6: warning: The called method `initi
alize' is defined here
==> default: Creating and configuring docker networks...
==> default: Creating the container...
default:   Name: vagrant_projet1_default_1611069053
default:   Image: ghost
default:   Volume: /home/bilel/vagrant_projet1:/vagrant
default:   Port: 85:2368
default:
default: Container created: 356a6349af544f4c
==> default: Enabling network interfaces...
==> default: Starting container...
==> default: Provisioners will not be run since container doesn't support SSH.
bilel@bilel-G3-3590:~/vagrant_projet1$
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Vagrant

```
bilel@bilel-G3-3590:~/vagrant_projet1$ vagrant up
Bringing machine 'default' up with 'docker' provider...
/usr/share/rubygems-integration/all/gems/vagrant-2.2.6/lib/vagrant/action/warden.rb:113: warning: Using the last argument as keyword parameters is deprecated; maybe ** sh
ould be added to the call
/usr/share/rubygems-integration/all/gems/vagrant-2.2.6/lib/vagrant/action/builtin/message.rb:6: warning: The called method `initialize' is defined here
==> default: Creating and configuring docker networks...
==> default: Creating the container...
default:   Name: vagrant_projet1_default_1611069053
default:   Image: ghost
default:   Volume: /home/bilel/vagrant_projet1:/vagrant
default:   Port: 85:2368
default:
default: Container created: 356a6349af544f4c
==> default: Enabling network interfaces...
==> default: Starting container...
==> default: Provisioners will not be run since container doesn't support SSH.
bilel@bilel-G3-3590:~/vagrant_projet1$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
356a6349af54   ghost         "docker-entrypoint.s..." 2 minutes ago  Up 2 minutes  0.0.0.0:85->2368/tcp     vagrant_projet1_default_1611069053
bilel@bilel-G3-3590:~/vagrant_projet1$
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuelle de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Vagrant

Vous avez appris comment installer et utiliser Vagrant sur votre machine, Nous vous avons également montré comment créer un environnement de développement de base.

Pour trouver plus d'informations sur Vagrant, visitez la page de documentation officielle de Vagrant .

<https://www.vagrantup.com/docs>

<https://stackshare.io/stackups/docker-machine-vs-vagrant>



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Docker-Machine

Docker Machine est un outil de provisioning et de gestion des hôtes Docker (hôtes virtuels exécutant le moteur Docker). Vous pouvez utiliser Docker Machine pour créer des hôtes Docker sur votre ordinateur personnel ou sur le datacenter de votre entreprise à l'aide d'un logiciel de virtualisation tel que VirtualBox ou VMWare, vous pouvez aussi déployer vos machines virtuelles chez des fournisseurs de cloud, tels que Azure, AWS, Google Compute Engine, etc ..

À l'aide de la commande docker-machine, vous pouvez démarrer, inspecter, arrêter et redémarrer un hôte géré ou mettre à niveau le client et le moteur Docker et configurer un client Docker pour qu'il puisse communiquer avec votre hôte. En bref il crée automatiquement des hôtes Docker, y installe le moteur Docker, puis configure les clients docker. (Il crée des serveurs, installe Docker sur eux, puis configure le client Docker pour leur parler).

Dans cet tutoriel vous allez apprendre comment déployer et gérer vos hôtes docker avec Docker Machine, Nous allons utiliser le driver de plateforme de gestion de virtualisation locale Virtualbox. le principe reste cependant le même avec les autres drivers



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Docker-Machine

Voici la commande qui permet d'installer Docker Machine sous Linux.

```
base=https://github.com/docker/machine/releases/download/v0.16.0 &&  
curl -L $base/docker-machine-$(uname -s)-$(uname -m) >/tmp/docker-machine &&  
sudo install /tmp/docker-machine /usr/local/bin/docker-machine
```

Pour pouvoir activer l'auto-completion des commandes Docker Machine, il suffit de créer un script qu'on va nommer **docker-machine-prompt.bash** dans le dossier **/etc/bash_completion.d** ou dans le dossier **/usr/local/etc/bash_completion.d** et de coller dedans le contenu ci-dessous :

```
sudo nano /etc/bash_completion.d/docker-machine-prompt.bash
```

```
base=https://raw.githubusercontent.com/docker/machine/v0.16.0  
for i in docker-machine-prompt.bash docker-machine-wrapper.bash docker-machine.bash  
do  
    sudo wget "$base/contrib/completion/bash/${i}" -P /etc/bash_completion.d  
done
```

Enfin, il faut lancer la commande source pour charger votre script d'auto-completion :

```
source /etc/bash_completion.d/docker-machine-prompt.bash
```

Docker machine utilise le concept des drivers (en fr : pilotes). Les drivers vous permettent depuis votre Docker machine. de créer un ensemble complet de ressources sur vos machines virtuelles sur des services tiers tels qu'Azure, Amazon, VirtualBox, etc. Vous retrouverez la liste des différents drivers <https://docs.docker.com/machine/drivers/>



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Docker-Machine

Avant de vous décrire l'utilisation de certains drivers. Voici d'abord la commande qui permet de créer une machine virtuelle depuis votre Docker Machine :

```
docker-machine create --drive <DRIVER NAME> <MACHINE NAME>
```

Une fois docker-machine et virtualbox sont installés vous pouvez dès lors créer votre hôte Docker en lançant la commande create en utilisant le driver virtualbox avec les options par défaut :

```
docker-machine create --driver virtualbox vbox-test
```

Ensuite, vérifiez la liste des machines Docker disponible en exécutant la commande suivante :

```
docker-machine ls
```



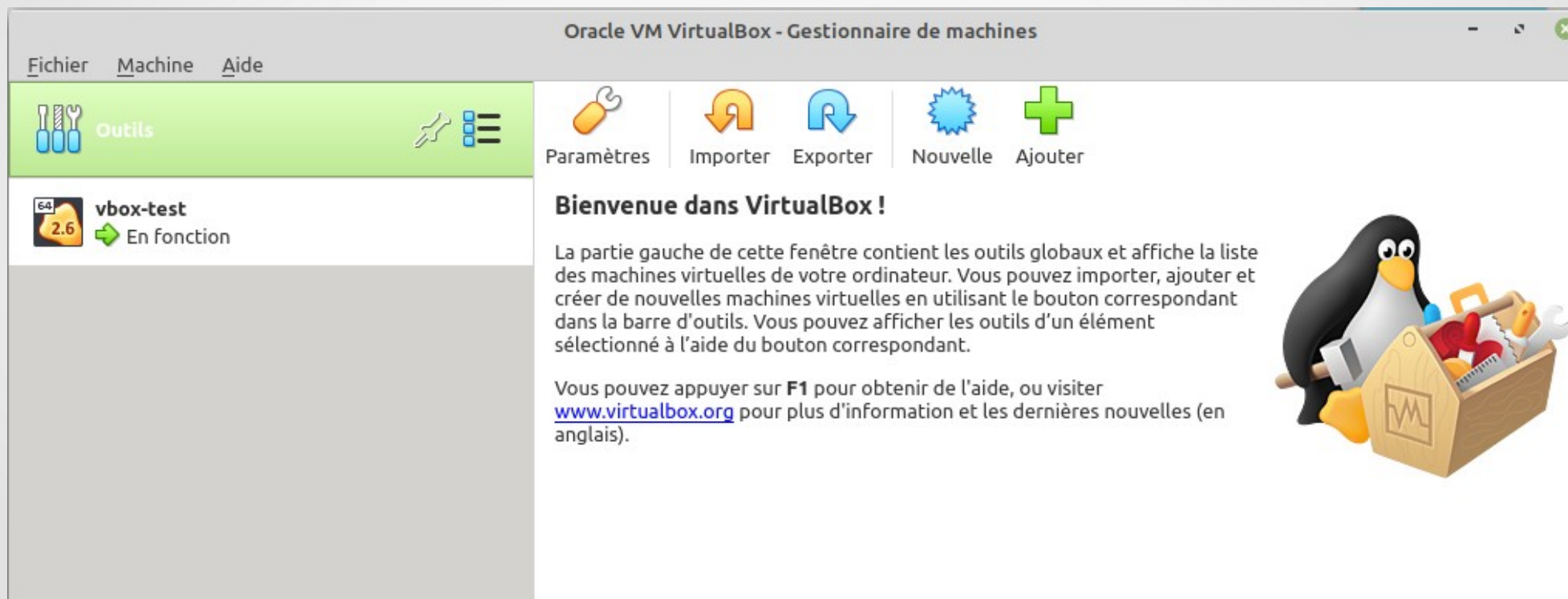
Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

```
bilel@bilel-G3-3590:~$ docker-machine create --driver virtualbox vbox-test
Creating CA: /home/bilel/.docker/machine/certs/ca.pem
Creating client certificate: /home/bilel/.docker/machine/certs/cert.pem
Running pre-create checks...
(vbox-test) Image cache directory does not exist, creating it at /home/bilel/.docker/machine/cache...
(vbox-test) No default Boot2Docker ISO found locally, downloading the latest release...
(vbox-test) Latest release for github.com/boot2docker/boot2docker is v19.03.12
(vbox-test) Downloading /home/bilel/.docker/machine/cache/boot2docker.iso from https://github.com/boot2docker/boot2docker/releases/download/v19.03.12/boot2docker.iso...
(vbox-test) 0%....10%....20%....30%....40%....50%....60%....70%....80%....90%....100%
Creating machine...
(vbox-test) Copying /home/bilel/.docker/machine/cache/boot2docker.iso to /home/bilel/.docker/machine/machines/vbox-test/boot2docker.iso...
(vbox-test) Creating VirtualBox VM...
(vbox-test) Creating SSH key...
(vbox-test) Starting the VM...
(vbox-test) Check network to re-create if needed...
(vbox-test) Found a new host-only adapter: "vboxnet0"
(vbox-test) Waiting for an IP...
Waiting for machine to be running, this may take a few minutes...
Detecting operating system of created instance...
Waiting for SSH to be available...
Detecting the provisioner...
Provisioning with boot2docker...
Copying certs to the local machine directory...
Copying certs to the remote machine...
Setting Docker configuration on the remote daemon...
Checking connection to Docker...
Docker is up and running!
To see how to connect your Docker Client to the Docker Engine running on this virtual machine, run: docker-machine env vbox-test
bilel@bilel-G3-3590:~$ docker-machine ls
NAME      ACTIVE  DRIVER      STATE     URL                         SWARM   DOCKER      ERRORS
vbox-test -        virtualbox   Running   tcp://192.168.99.100:2376   -       v19.03.12   -
bilel@bilel-G3-3590:~$
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuelle de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Docker-Machine



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Docker-Machine

```
bilel@bilel-G3-3590:~$ docker-machine ls
NAME      ACTIVE DRIVER      STATE URL                                SWARM DOCKER  ERRORS
vbox-test -      virtualbox  Running tcp://192.168.99.100:2376 v19.03.12
```

D'après le résultat, notre hôte vbox-test est bien présent avec l'état Running et possède le moteur docker en version v19.03.12

Si vous retournez à la fin du résultat de la commande `docker-machine create`, vous remarquerez le message suivant (traduit en français) : "Pour voir comment connecter Docker à cette machine, exécutez: **docker-machine env vbox-test**". Cette manipulation, va nous permettre de récupérer les variables d'environnements de la nouvelle VM à exporter

Docker is up and running!

To see how to connect your Docker Client to the Docker Engine running on this virtual machine, run: `docker-machine env vbox-test`

```
bilel@bilel-G3-3590:~$ docker-machine ls
```

```
docker-machine env vbox-test
```

Résultat :

```
bilel@bilel-G3-3590:~$ docker-machine env vbox-test
export DOCKER_TLS_VERIFY="1"
export DOCKER_HOST="tcp://192.168.99.100:2376"
export DOCKER_CERT_PATH="/home/bilel/.docker/machine/machines/vbox-test"
export DOCKER_MACHINE_NAME="vbox-test"
# Run this command to configure your shell:
# eval $(docker-machine env vbox-test)
bilel@bilel-G3-3590:~$
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Docker-Machine

Le résultat nous indique clairement que si on souhaite utiliser le moteur Docker de la machine virtuelle sur notre shell courant il faut alors utiliser la commande suivante :

```
eval $(docker-machine env vbox-test)
```

En exécutant cette commande sur votre shell courant, alors n'importe quelle commande Docker que vous exécuterez, sera dorénavant directement prise en compte par votre hôte Docker vbox-test et non plus par votre hôte maître.

Note : si vous voulez exécuter les commandes docker sur votre hôte, utilisez la commande : eval \$(docker-machine env -u)

Par ailleurs si vous souhaitez vérifier sur quelle hôte Docker se lanceront vos prochaines commandes docker alors soit vous vérifiez si une étoile existe dans la colonne ACTIVE de la commande **docker-machine ls**. Soit plus simple encore, vous lancez la commande suivante :

```
docker-machine active
```

Résultat :

```
bilel@bilel-G3-3590:~$ docker-machine ls
NAME          ACTIVE  DRIVER        STATE     URL                                     SWARM   DOCKER   ERRORS
vbox-test     -       virtualbox     Running   tcp://192.168.99.100:2376              v19.03.12
bilel@bilel-G3-3590:~$ eval $(docker-machine env vbox-test)
bilel@bilel-G3-3590:~$ docker-machine ls
NAME          ACTIVE  DRIVER        STATE     URL                                     SWARM   DOCKER   ERRORS
vbox-test     *       virtualbox     Running   tcp://192.168.99.100:2376              v19.03.12
bilel@bilel-G3-3590:~$ docker-machine active
vbox-test
bilel@bilel-G3-3590:~$
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Docker-Machine

Le résultat nous indique distinctement, que nos futures commandes docker sur le shell courant s'exécuteront directement sur la machine Docker vbox-test.

Afin de vous prouver que c'est effectivement le cas, je vais télécharger et exécuter l'image httpd sur le shell courant :

```
bilel@bilel-G3-3590:~$ docker-machine ls
NAME      ACTIVE  DRIVER      STATE      URL                      SWARM   DOCKER   ERRORS
vbox-test *      virtualbox   Running    tcp://192.168.99.100:2376           v19.03.12
bilel@bilel-G3-3590:~$ docker run -tid --name vbox_test_httpd -p 8080:80 httpd
Unable to find image 'httpd:latest' locally
latest: Pulling from library/httpd
a076a628af6f: Pull complete
e444656f7792: Pull complete
0ec35e191b09: Pull complete
4aad5d8db1a6: Pull complete
eb1da3ea630f: Pull complete
Digest: sha256:2fab99fb3b1c7ddfa99d7dc55de8dad0a62dbe3e7c605d78ecbdf2c6c49fd636
Status: Downloaded newer image for httpd:latest
3154196b67c26e3cfffdf6bd38301766760989f6cb685a821c2ed95bf224af1c6
bilel@bilel-G3-3590:~$
```

A présent, ouvrez un nouveau terminal et vérifiez les conteneurs disponibles, vous verrez ainsi que vous ne retrouverez pas le conteneur vbox-test-httpd créé précédemment :

```
bilel@bilel-G3-3590:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS
bilel@bilel-G3-3590:~$
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuelle de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Docker-Machine

Cependant si je retourne sur mon ancien shell avec la machine docker vbox-test activée, dans ce cas j'obtiens bien un résultat avec le conteneur vbox-test-httpd :

```
bilel@bilel-G3-3590:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                NAMES
3154196b67c2   httpd     "httpd-foreground"      4 minutes ago  Up 4 minutes  0.0.0.0:8080->80/tcp  vbox_test_
httpd
bilel@bilel-G3-3590:~$
```

Vous n'êtes pas encore convaincu ? Alors connectez vous carrément à la machine Docker à l'aide de la commande suivante :

```
docker-machine ssh vbox-test
```

Revérifiez une nouvelle fois la liste de vos conteneurs et vous verrez que le conteneur vbox-test-httpd est bien dedans :

```
bilel@bilel-G3-3590:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                NAMES
3154196b67c2   httpd     "httpd-foreground"      4 minutes ago  Up 4 minutes  0.0.0.0:8080->80/tcp  vbox_test_
httpd
bilel@bilel-G3-3590:~$ docker-machine ssh vbox-test
( '>')
/) TC (\   Core is distributed with ABSOLUTELY NO WARRANTY.
(/-_-_-\)   www.tinycorelinux.net

docker@vbox-test:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                NAMES
3154196b67c2   httpd     "httpd-foreground"      7 minutes ago  Up 7 minutes  0.0.0.0:8080->80/tcp  vbox_test_
httpd
docker@vbox-test:~$
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuelle de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Docker-Machine

Si vous pensez avoir fini d'utiliser une machine Docker, vous pouvez l'arrêter avec la commande **docker-machine stop** et la redémarrer plus tard avec la commande **docker-machine start**, exemple :

```
bilel@bilel-G3-3590:~$ docker-machine ls
NAME          ACTIVE  DRIVER      STATE      URL                      SWARM   DOCKER   ERRORS
vbox-test     *       virtualbox   Running    tcp://192.168.99.100:2376           v19.03.12
bilel@bilel-G3-3590:~$ docker-machine stop vbox-test
Stopping "vbox-test"...
Machine "vbox-test" was stopped.
bilel@bilel-G3-3590:~$ docker-machine ls
NAME          ACTIVE  DRIVER      STATE      URL      SWARM   DOCKER   ERRORS
vbox-test     -       virtualbox   Stopped                    Unknown
bilel@bilel-G3-3590:~$ docker-machine start vbox-test
Starting "vbox-test"...
(vbox-test) Check network to re-create if needed...
(vbox-test) Waiting for an IP...
Machine "vbox-test" was started.
Waiting for SSH to be available...
Detecting the provisioner...
Started machines may have new IP addresses. You may need to re-run the `docker-machine env` command.
bilel@bilel-G3-3590:~$ docker-machine ls
NAME          ACTIVE  DRIVER      STATE      URL                      SWARM   DOCKER   ERRORS
vbox-test     *       virtualbox   Running    tcp://192.168.99.100:2376           v19.03.12
bilel@bilel-G3-3590:~$
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuelle de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Docker-Machine

Enfin, vous pouvez surcharger les ressources allouées automatiquement par défaut à hôte Docker en utilisant les options venant avec le driver virtualbox. Dans cet exemple je vais créer une machine Docker avec 30 Go d'espace disque (20 Go par défaut) et avec 2 Go de ram (1Go par défaut)

```
docker-machine create -d virtualbox \
--virtualbox-disk-size "30000" \
--virtualbox-memory "2000" \
vbox-test1
```

```
bilel@bilel-G3-3590:~$ docker-machine create -d virtualbox --virtualbox-disk-size "30000" --virtualbox-memory "2000" vbox-test1
Running pre-create checks...
Creating machine...
(vbox-test1) Copying /home/bilel/.docker/machine/cache/boot2docker.iso to /home/bilel/.docker/machine/machines/vbox-test1/boot2docker.
iso...
(vbox-test1) Creating VirtualBox VM...
(vbox-test1) Creating SSH key...
(vbox-test1) Starting the VM...
(vbox-test1) Check network to re-create if needed...
(vbox-test1) Waiting for an IP...
Waiting for machine to be running, this may take a few minutes...
Detecting operating system of created instance...
Waiting for SSH to be available...
Detecting the provisioner...
Provisioning with boot2docker...
Copying certs to the local machine directory...
Copying certs to the remote machine...
Setting Docker configuration on the remote daemon...
Checking connection to Docker...
Docker is up and running!
To see how to connect your Docker Client to the Docker Engine running on this virtual machine, run: docker-machine env vbox-test1
bilel@bilel-G3-3590:~$ docker-machine ls
NAME      ACTIVE  DRIVER      STATE     URL                  SWARM   DOCKER   ERRORS
vbox-test *      virtualbox  Running   tcp://192.168.99.100:2376  -       v19.03.12
vbox-test1 -        virtualbox  Running   tcp://192.168.99.101:2376  -       v19.03.12
bilel@bilel-G3-3590:~$
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuelle de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Docker-Machine

Vous retrouverez plus d'informations sur les options liées au driver virtualbox : <https://docs.docker.com/machine/drivers/virtualbox/>

Comme je n'ai plus besoin de mes machines, je peux alors les supprimer. Pour ce faire, je vais utiliser la commande **docker-machine rm <MACHINE NAME>**. Cette commande aura pour effet de supprimer définitivement la machine Docker de votre plateforme de gestion de virtualisation locale mais aussi de la supprimer de votre fournisseur de cloud, si jamais vous en utilisez un.

```
bilel@bilel-G3-3590:~$ docker-machine ls
NAME          ACTIVE  DRIVER      STATE     URL                  SWARM   DOCKER      ERRORS
vbox-test     *       virtualbox   Running   tcp://192.168.99.100:2376           v19.03.12
vbox-test1    -       virtualbox   Running   tcp://192.168.99.101:2376           v19.03.12
bilel@bilel-G3-3590:~$ docker-machine rm vbox-test1
About to remove vbox-test1
WARNING: This action will delete both local reference and remote instance.
Are you sure? (y/n): y
Successfully removed vbox-test1
bilel@bilel-G3-3590:~$ docker-machine ls
NAME          ACTIVE  DRIVER      STATE     URL                  SWARM   DOCKER      ERRORS
vbox-test     *       virtualbox   Running   tcp://192.168.99.100:2376           v19.03.12
bilel@bilel-G3-3590:~$
```

Nous avons utilisé Docker Machine pour créer des hôtes Docker localement, cela nous montre à quel point il est facile de déployer des machines Docker n'importe où et de centraliser la gestion de ces VMs depuis une seule machine maître. Comme à mon habitude, je partage avec vous un aide-mémoire résumant les différentes commandes de Docker Machine.



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuelle de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Docker-Machine

```
## Créer une machine Docker
docker-machine create -d <DRIVER NAME> <MACHINE NAME>
  -d ou --driver : choisir un driver
## Rendre une machine Docker active
eval $(docker-machine env <MACHINE NAME>)
# Lister les machines Docker
docker-machine ls
# Vérifier quelle est la machine Docker active dans le shell courant
docker-machine active
## Supprimer un ou plusieurs machine(s) Docker
docker-machine rm <MACHINE NAME>
  -f ou --force : forcer la suppression
## Se connecter en ssh sur une machine Docker
docker-machine ssh <MACHINE NAME>
## Stopper une machine Docker
docker-machine stop <MACHINE NAME>
## Démarrer une machine Docker
docker-machine start <MACHINE NAME>
## Redémarrer une machine Docker
docker-machine restart <MACHINE NAME>
## Récupérer des informations sur une machine Docker
docker-machine inspect <MACHINE NAME>
## Récupérer les variables d'environnements d'une machine Docker
docker-machine env <MACHINE NAME>
## Mettre à niveau une machine Docker vers la dernière version de Docker
docker-machine upgrade <MACHINE NAME>
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuelle de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

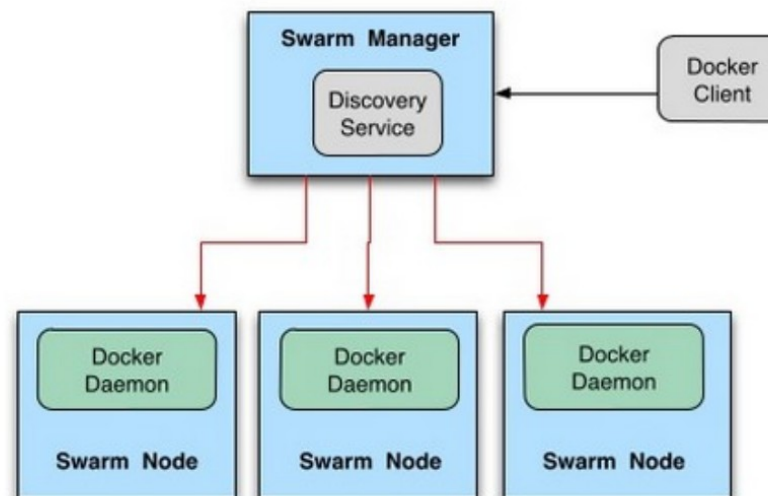
Docker Swarm

Un Swarm est un groupe de machines exécutant le moteur Docker et faisant partie du même cluster. Docker swarm vous permet de lancer des commandes Docker auxquelles vous êtes habitué sur un cluster depuis une machine maître nommée **manager/leader Swarm**. Quand des machines rejoignent un Swarm, elles sont appelés **nœuds**.

Les managers Swarm sont les seules machines du Swarm qui peuvent exécuter des commandes Docker ou autoriser d'autres machines à se joindre au Swarm en tant que workers. Les workers ne sont là que pour fournir de la capacité et n'ont pas le pouvoir d'ordonner à une autre machine ce qu'elle peut ou ne peut pas faire.

<https://docs.docker.com/engine/swarm/>

Architecture Diagram



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Docker Swarm

Dans le vocabulaire Swarm nous ne parlons plus vraiment de conteneurs mais plutôt de services.

Un service n'est rien d'autre qu'une description de l'état souhaité pour vos conteneurs. Une fois le service lancé, une tâche est alors attribuée à chaque nœud afin d'effectuer le travail demandé par le service.

les différents concepts de Docker Swarm

Il est important au préalable de bien comprendre la définition de chaque concept afin d'assimiler facilement le fonctionnement global de Docker Swarm. Je vais donc vous résumer toutes ces notions à travers un seul exemple :

A- Une entreprise vous demande de déployer une application web tournant sous le serveur web nginx. Avant toute chose, vous allez commencer par définir les caractéristiques et les états de votre conteneur, comme par exemple :

- Trois conteneurs minimums par application afin de supporter des grandes charges de travail
- Une utilisation maximale de 100 Mo de mémoire pour chaque conteneur
- les conteneurs se baseront sur l'image NGINX
- Le port 8080 sera mappé sur le port 80
- Redémarrer automatiquement les conteneurs s'ils se ferment suite à une erreur

Pour le moment vous avez défini l'état et les comportements de vos conteneurs dans votre service, par la suite quand vous exécuterez votre service, chaque nœud se verra attribuer alors une ou plusieurs tâches jusqu'à satisfaire les besoins définis par votre service.



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuelle de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Docker Swarm

B- Votre chef de projet vous demande alors de lui expliquer le process de déploiement de l'application web dans un Cluster Swarm.

Comme vous êtes un très bon ingénieur , vous lui expliquez les étapes suivantes :

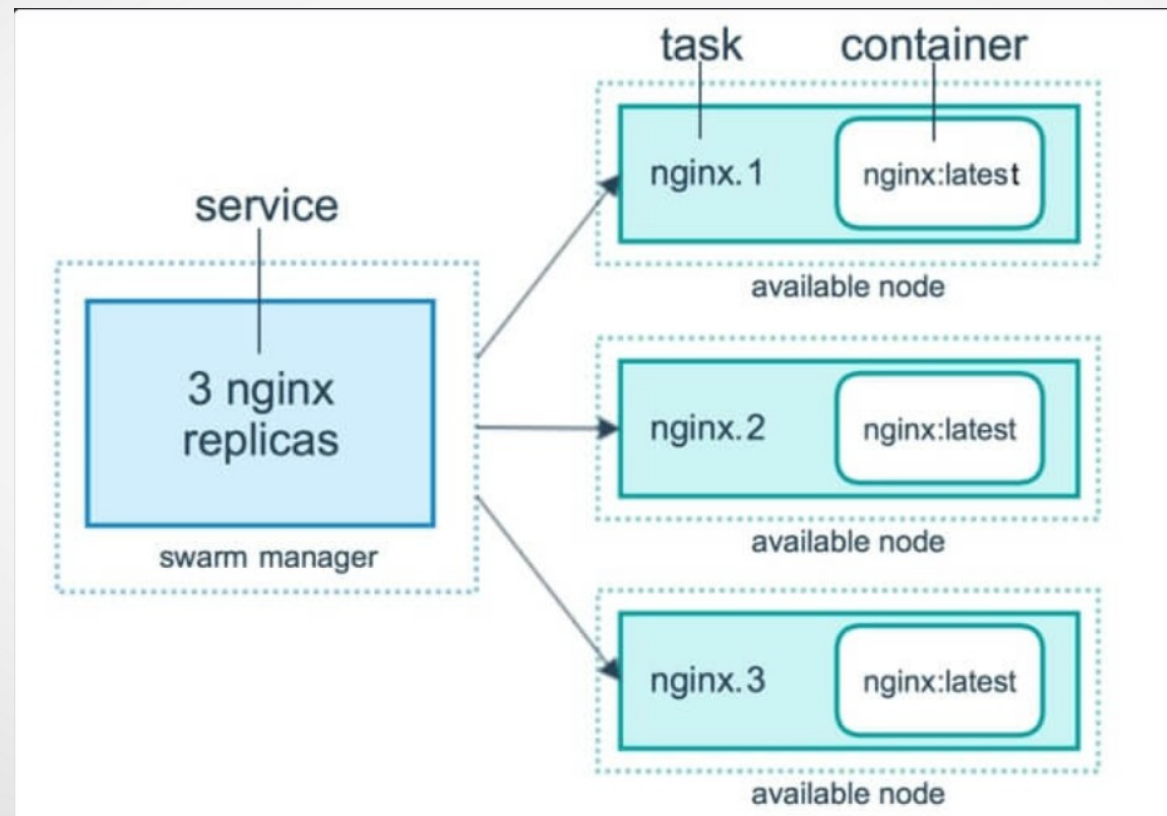
- Un service sera créé dans lequel nous spécifierons qu'elle sera image utilisée et nous estimerons quelles seront les charges de travail suffisantes pour les conteneurs qui seront en cours d'exécution.
- La demande sera ensuite envoyée au manager Swarm (leader) qui planifie l'exécution du service sur des nœuds particuliers.
- Chaque nœud se voit assigné une ou plusieurs tâche(s) en fonction de l'échelle qui a été définie dans le service.
- Chaque tâche aura un cycle de vie, avec des états comme NEW , PENDING et COMPLETE.
- Un équilibreur de charge sera mis en place automatiquement par votre manager Swarm afin de supporter les grandes charges de travaux.



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Docker Swarm

Voici un schéma qui reprend les étapes citées précédemment :



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

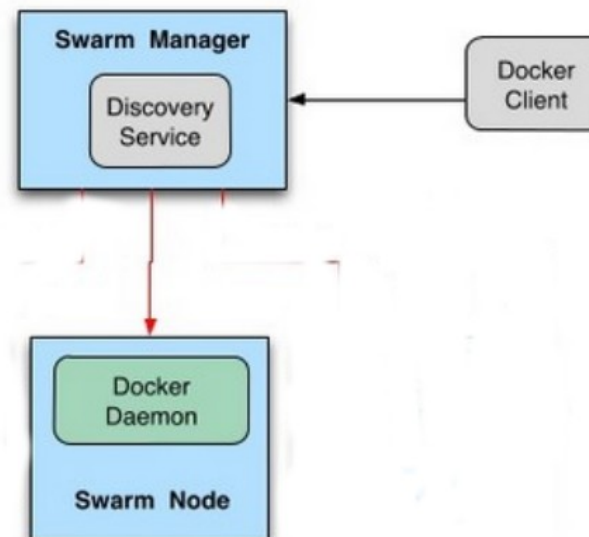
Docker Swarm

I :- Créer d'un cluster Swarm

1- Création des nœuds

Dans le chapitre précédent , nous avons vu comment déployer des machines Docker avec l'outil Docker Machine. Nous allons réutiliser cet outil de façon à créer deux VMs avec le driver virtualbox.

Architecture Diagram



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Docker Swarm

```
docker-machine create --driver virtualbox mymanager  
docker-machine create --driver virtualbox worker
```

```
bilel@bilel-G3-3590:~$ docker-machine ls  
NAME      ACTIVE  DRIVER      STATE      URL                     SWARM   DOCKER   ERRORS  
mymanager -       virtualbox  Running    tcp://192.168.99.102:2376  v19.03.12  
worker    -       virtualbox  Running    tcp://192.168.99.103:2376  v19.03.12  
bilel@bilel-G3-3590:~$
```

Vous avez maintenant créé deux machines virtuelles, nommées mymanager et worker qu'on utilisera plus tard en tant que nœuds dans notre Swarm. Nous allons exploiter la première machine virtuelle mymanager en tant que manager Swarm et l'autre machine sera un nœud de travail.

2- Activer le mode swarm

L'avantage de Docker Swarm est qu'il est déjà intégré dans Docker par défaut, donc pas besoin d'installation supplémentaire.

Dans cette partie, nous allons activer le mode Swarm sur la machine Docker **mymanager** de la même manière elle deviendra la leader de notre Swarm. Pour ce faire, nous utiliserons la commande **docker-machine ssh**

```
bilel@bilel-G3-3590:~$ docker-machine ssh mymanager  
( '~>' )  
/) TC (\   Core is distributed with ABSOLUTELY NO WARRANTY.  
(/_--_-\)   www.tinycorelinux.net  
docker@mymanager:~$
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Docker Swarm

Voici la commande qui permet d'activer le mode Swarm sur une machine d'un Swarm :

```
docker swarm init --advertise-addr 192.168.99.102
```

```
docker@mymanager:~$ docker swarm init --advertise-addr 192.168.99.102
Swarm initialized: current node (z7jx84nuc6lw29ivrqirtew4) is now a manager.
```

To add a worker to this swarm, run the following command:

```
docker swarm join --token SWMTKN-1-1s7rhw45yt8garfmurhtqyk2fdl14og68av109roe34699yrs0-3dnk3tbselle5j9qen5bx2mrg 192.168.99.102:2377
```

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.

```
docker@mymanager:~$
```

Votre cluster Swarm est prêt maintenant à accueillir de nouvelles machines. De plus, le résultat nous indique clairement comment rajouter une machine à notre Swarm, et c'est exactement ce que nous allons faire dans la prochaine étape.

3- Joindre une machine au cluster Swarm

Pour ajouter un travailleur à un swarm, exécutez la commande suivante : `$ docker swarm join`

```
bilel@bilel-63-3590:~$ docker-machine ssh worker
( '>' )
/) TC (\   Core is distributed with ABSOLUTELY NO WARRANTY.
(/' _--_\)   www.tinycorelinux.net

docker@worker:~$ docker swarm join --token SWMTKN-1-1s7rhw45yt8garfmurhtqyk2fdl14og68av109roe34699yrs0-3dnk3tbselle5j9qen5bx2mrg 192.168.99.102:2377
This node joined a swarm as a worker.
docker@worker:~$
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Docker Swarm

Voici la commande qui permet d'afficher les différents nœuds de votre Swarm : `$ docker node ls`

```
docker@mymanager:~$
docker@mymanager:~$ docker node ls
```

ID	HOSTNAME	STATUS	AVAILABILITY	MANAGER STATUS	ENGINE VERSION
z7jx84nuc61vw29ivrqrtew4 *	mymanager	Ready	Active	Leader	19.03.12
qgflfe2bpqpuegz7gvtmdlipm	worker	Ready	Active		19.03.12

```
docker@mymanager:~$
```

D'ailleurs le résultat nous indique formellement dans la colonne MANAGER STATUS que c'est le nœud mymanager le Big Boss

Rappel : Rappelez-vous que seuls les managers du Swarm comme mymanager exécutent les commandes Docker, les workers ne sont là juste pour fournir de la capacité.

Jusqu'à présent, nous utilisons des commandes Docker en ssh sur le nœud manager. Ce n'est pas vraiment pratique comme solution, en vue de nous faciliter la vie, désormais nous chargerons les variables d'environnements de notre leader Swarm sur notre shell courant, dès lors toutes les prochaines commandes docker lancées depuis le shell courant s'exécuteront directement sur la machine mymanager :

La commande suivante, permet de configurer votre shell pour lui permettre de communiquer avec le nœud mymanager :

```
eval $(docker-machine env mymanager)
```

```
bilel@bilel-G3-3590:~$ docker-machine ls
```

NAME	ACTIVE	DRIVER	STATE	URL	SWARM	DOCKER	ERRORS
mymanager	-	virtualbox	Running	tcp://192.168.99.102:2376		v19.03.12	
worker	-	virtualbox	Running	tcp://192.168.99.103:2376		v19.03.12	

```
bilel@bilel-G3-3590:~$ eval $(docker-machine env mymanager)
bilel@bilel-G3-3590:~$ docker-machine ls
```

NAME	ACTIVE	DRIVER	STATE	URL	SWARM	DOCKER	ERRORS
mymanager	*	virtualbox	Running	tcp://192.168.99.102:2376		v19.03.12	
worker	-	virtualbox	Running	tcp://192.168.99.103:2376		v19.03.12	

```
bilel@bilel-G3-3590:~$ docker node ls
```

ID	HOSTNAME	STATUS	AVAILABILITY	MANAGER STATUS	ENGINE VERSION
z7jx84nuc61vw29ivrqrtew4 *	mymanager	Ready	Active	Leader	19.03.12
qgflfe2bpqpuegz7gvtmdlipm	worker	Ready	Active		19.03.12

```
bilel@bilel-G3-3590:~$
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Docker Swarm

II :- Déployer une application à service individuel dans un Swarm avec Docker Service

1- Créer un service

La commande `docker service` est utilisée lors de la gestion d'un service individuel dans un cluster Swarm.

Voici une idée de ce qu'un service peut définir comme comportement et état d'un conteneur :

- Le nom de l'image et le tag que les conteneurs du nœud doivent exécuter.
- Combien de conteneurs participent au service.
- Les ports à exposer à l'extérieur du cluster Swarm.
- Comment doit agir le conteneur suite à une erreur.

etc ...

Pour l'exemple ci-dessous, on souhaite spécifier les comportements suivants :

- Un conteneur qui se base sur ma propre image SWARM
- Trois conteneurs doivent être exécutés au minimum.
- Redémarrez le conteneur s'il se ferme suite à une erreur.
- Limiter l'utilisation de la mémoire à 100 Mo
- Mapper le port 5000 sur le port 5000



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuelle de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Docker Swarm

Ci-dessous la commande qui crée un service avec ses options respectant les caractéristiques définies plus haut :

```
docker service create --name myapp \  
--replicas 3 \  
-p 5000:5000 \  
--restart-condition=on-failure \  
--limit-memory 100M \  
issaouib/swarm:first
```

Résultat :

```
bilel@bilel-G3-3590:~$ docker service create --name myapp \  
> --replicas 3 \  
> -p 5000:5000 \  
> --restart-condition=on-failure \  
> --limit-memory 100M \  
> issaouib/swarm:first  
972siuvo4qjq04i4f08ugr8ri  
overall progress: 3 out of 3 tasks  
1/3: running [=====>]  
2/3: running [=====>]  
3/3: running [=====>]  
verify: Service converged  
bilel@bilel-G3-3590:~$
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Docker Swarm

Vous pouvez connaître l'état d'avancement du service dans votre Swarm en lançant la commande suivante :

```
$ docker service ls
```

```
bilel@bilel-G3-3590:~$ docker service ls
```

ID	NAME	MODE	REPLICAS	IMAGE	PORTS
972siuvo4qjq	myapp	replicated	3/3	issaouib/swarm:first	*:5000->5000/tcp

```
bilel@bilel-G3-3590:~$
```

Vous pouvez également lister les différentes tâches de votre service afin de vérifier par exemple sur quel nœud s'est exécutée votre tâche.

```
$ docker service ps
```

```
bilel@bilel-G3-3590:~$ docker service ls
```

ID	NAME	MODE	REPLICAS	IMAGE	PORTS
972siuvo4qjq	myapp	replicated	3/3	issaouib/swarm:first	*:5000->5000/tcp

```
bilel@bilel-G3-3590:~$ docker service ps myapp
```

ID	NAME	IMAGE	NODE	DESIRED STATE	CURRENT STATE	ERROR	PORTS
vtqgq29st0fi	myapp.1	issaouib/swarm:first	worker	Running	Running 3 minutes ago		
9ytrlcwi4zre	myapp.2	issaouib/swarm:first	mymanager	Running	Running 3 minutes ago		
aqvwkqh7x15n	myapp.3	issaouib/swarm:first	worker	Running	Running 3 minutes ago		

```
bilel@bilel-G3-3590:~$
```

Selon le résultat nous avons deux répliques dans le nœud **worker** et une seule réplique dans le nœud **mymanager**



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Docker Swarm

2- Accéder à votre cluster

Actuellement, nous avons trois de conteneurs tournant dans notre Swarm, tous cyclant de manière aléatoire, grâce à un équilibreur de charge créée automatiquement par Swarm. Oui oui, vous n'avez pas besoin de configurer un Load Balancer, Swarm s'occupe de tous.

Vous ne me croyez pas ? J'ai faits exprès sur mon image SWARM d'afficher le hostname du conteneur, par défaut le hostname du conteneur équivaut à son ID. Nous allons récupérer les IDs de tous nos conteneurs grâce aux commandes suivantes : `$ docker ps`

Je commence par afficher l'ID des conteneurs de mon nœud **mymanager**

```
bilel@bilel-G3-3590:~$ docker-machine ls
```

NAME	ACTIVE	DRIVER	STATE	URL	SWARM	DOCKER	ERRORS
mymanager	*	virtualbox	Running	tcp://192.168.99.102:2376		v19.03.12	
worker	-	virtualbox	Running	tcp://192.168.99.103:2376		v19.03.12	

```
bilel@bilel-G3-3590:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
3d213b298d83	issaouib/swarm:first	"python ./app.py"	12 minutes ago	Up 12 minutes		myapp.2.9ytrlcwi4zre7auwew3u6xv47

```
bilel@bilel-G3-3590:~$
```

Je récupère ensuite l'ID des conteneurs de mon nœud **worker** :

```
bilel@bilel-G3-3590:~$ eval $(docker-machine env worker)
```

```
bilel@bilel-G3-3590:~$ docker-machine ls
```

NAME	ACTIVE	DRIVER	STATE	URL	SWARM	DOCKER	ERRORS
mymanager	-	virtualbox	Running	tcp://192.168.99.102:2376		v19.03.12	
worker	*	virtualbox	Running	tcp://192.168.99.103:2376		v19.03.12	

```
bilel@bilel-G3-3590:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
9367aae797e3	issaouib/swarm:first	"python ./app.py"	14 minutes ago	Up 14 minutes		myapp.1.vtqqq29st0fi4srlxyys0pn15
c7a4c57c05fe	issaouib/swarm:first	"python ./app.py"	14 minutes ago	Up 14 minutes		myapp.3.aqvwkqh7x15n6nzk0wbs595rz

```
bilel@bilel-G3-3590:~$
```



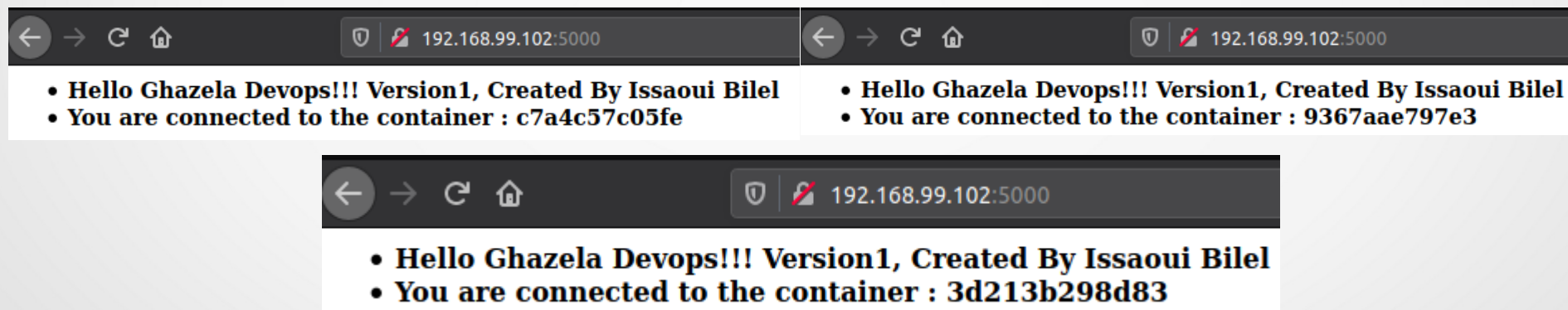
Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Docker Swarm

Pour résumer nous avons :

- **3d213b298d83** : ID du conteneur du nœud **mymanager**.
- **9367aae797e3** : ID du premier conteneur du nœud **worker**.
- **c7a4c57c05fe** : ID du deuxième conteneur du nœud **worker**.

Nous allons accéder à notre application à partir de l'adresse IP du nœud mymanager (vous pouvez utiliser également l'IP du nœud worker) afin de vérifier le hostname récupéré par la page, voici mes résultats sous forme de Screenshots :



On peut clairement constater que tous les conteneurs du cluster sont appelés de manière aléatoire grâce à l'équilibreur de charge mis en place par Swarm.



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Docker Swarm

3- Mise à l'échelle de votre service

Votre application revendique encore plus de puissance ? Pas de soucis, voici la commande qui permet de scaler automatiquement vos conteneurs sans les redémarrer : `$ docker service scale myapp=5`

```
bilel@bilel-G3-3590:~$ docker service ls
ID                NAME      MODE                REPLICAS            IMAGE              PORTS
972siuvo4qjq      myapp     replicated          3/3                 issaouib/swarm:first *:5000->5000/tcp
bilel@bilel-G3-3590:~$ docker service scale myapp=5
myapp scaled to 5
overall progress: 5 out of 5 tasks
1/5: running [=====]
2/5: running [=====]
3/5: running [=====]
4/5: running [=====]
5/5: running [=====]
verify: Service converged
bilel@bilel-G3-3590:~$
```

Vérifions maintenant le nombre de tâches de nos nœuds afin de s'assurer du nombre de répliques

```
bilel@bilel-G3-3590:~$ docker service ls
ID                NAME      MODE                REPLICAS            IMAGE              PORTS
64paw4iox52v      myapp     replicated          5/5                 issaouib/swarm:first *:5000->5000/tcp
bilel@bilel-G3-3590:~$ docker service ps myapp
ID                NAME      IMAGE              NODE                DESIRED STATE      CURRENT STATE
l3mbwjeilzmi      myapp.1   issaouib/swarm:first worker             Running             Running 2 minutes ago
lwxcdbjdgey7      myapp.2   issaouib/swarm:first mymanager           Running             Running 2 minutes ago
tppjyc4aplak      myapp.3   issaouib/swarm:first worker             Running             Running 2 minutes ago
yz2ros4i02c6      myapp.4   issaouib/swarm:first mymanager           Running             Running 16 seconds ago
lpwj98nsa85b      myapp.5   issaouib/swarm:first mymanager           Running             Running 16 seconds ago
bilel@bilel-G3-3590:~$
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuelle de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

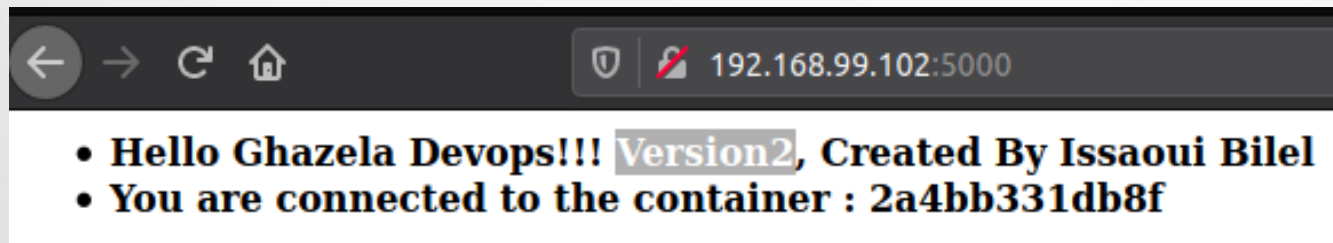
Docker Swarm

4- Mettre à jour votre service

Il est possible de mettre à jour votre application depuis votre nouvelle image sans interruption de service.

Dans cet exemple, je vais simuler une nouvelle version de notre application grâce à ma nouvelle image que j'ai poussée dans le Docker Hub :

```
bilel@bilel-63-3590:~$ docker service update --image issaouib/swarm:second myapp
myapp
overall progress: 5 out of 5 tasks
1/5: running [=====>]
2/5: running [=====>]
3/5: running [=====>]
4/5: running [=====>]
5/5: running [=====>]
verify: Service converged
bilel@bilel-63-3590:~$
```



Sans aucune interruption de service, nous avons pu mettre à jour 5 conteneurs déployés sur différents nœuds et ceci depuis une seule commande ! Je pense que vous commencez maintenant à comprendre la puissance de Docker Swarm



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Docker Swarm

5- Supprimer votre service

Vous n'avez plus besoin de votre service ? Aucun problème, vous pouvez utiliser la commande suivante pour supprimer votre service : `$ docker service rm myapp`

```
bilel@bilel-G3-3590:~$ docker service rm myapp
myapp
bilel@bilel-G3-3590:~$ docker service ls
ID                NAME                MODE                REPLICAS                IMAGE                PORTS
bilel@bilel-G3-3590:~$
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Docker Swarm

III :- Déployer une application multi-services dans un Swarm avec Docker stack

Docker Stack peut être utilisée pour gérer une application multi-services dans votre cluster Swarm. Pour faire simple, vous pouvez considérer que la commande `docker service` est identique à la commande `docker run` et que la commande `docker stack` est comparable à la commande `docker-compose`.

Nous allons utiliser les super pouvoirs de notre nœud **mymanager** pour déployer nos applications Docker multi-services dans notre cluster Swarm.

Comme pour la création de notre service précédemment, nous allons définir les différentes caractéristiques de nos conteneurs :

- Deux API sous forme de deux services différents.
- Trois conteneurs par service doivent être exécutés au minimum.
- Redémarrez les services s'il se ferme suite à une erreur.
- Limiter l'utilisation de la mémoire à 50 MO.
- La première API écoute sur le port 5000 mais la deuxième API utilise le port 5001 comme port cible et le port 5000 comme port source.



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Docker Swarm

Pas besoin d'apprendre de nouvelles choses pour créer nos services, car nous allons réutiliser les connaissances vues dans les chapitres précédents puisqu'il suffit juste de créer un fichier docker-compose.yml :

```
services:
  api1:
    image: issaouib/swarm:api1
    deploy:
      replicas: 3
      resources:
        limits:
          memory: 50M
      restart_policy:
        condition: on-failure
    ports:
      - "5000:5000"

  api2:
    image: issaouib/swarm:api2
    deploy:
      replicas: 3
      resources:
        limits:
          memory: 50M
      restart_policy:
        condition: on-failure
    ports:
      - "5001:5000"
```

La commande suivante permet de déployer votre application multi-services dans votre Swarm :

```
$ docker stack deploy -c docker-compose.yml api-app
```

```
bilel@bilel-G3-3590:~$ docker stack deploy -c docker-compose.yml api-app
Creating network api-app_default
Creating service api-app_api1
Creating service api-app_api2
bilel@bilel-G3-3590:~$
```

Les commandes de Docker Stack restent très identiques aux commandes docker services , par exemple pour lister les différentes tâches dans votre Swarm, vous utiliserez la commande suivante :

```
$ docker stack ps api-app
```

```
bilel@bilel-G3-3590:~$ docker stack ls
NAME      SERVICES      ORCHESTRATOR
api-app    2              Swarm
bilel@bilel-G3-3590:~$ docker stack ps api-app
```

ID	NAME	IMAGE	NODE	DESIRED STATE	CURRENT STATE
tlo6dd3u2d97	api-app_api2.1	issaouib/swarm:api2	mymanager	Running	Running 2 minutes ago
ufkdisrbjzy8	api-app_api1.1	issaouib/swarm:api1	mymanager	Running	Running 2 minutes ago
7m5ly3ud3f26	api-app_api2.2	issaouib/swarm:api2	worker	Running	Running 2 minutes ago
g9vzt2iwh4mh	api-app_api1.2	issaouib/swarm:api1	worker	Running	Running 2 minutes ago
08dpvhq8h9l0	api-app_api2.3	issaouib/swarm:api2	worker	Running	Running 2 minutes ago
po7zjvgmocld	api-app_api1.3	issaouib/swarm:api1	mymanager	Running	Running 2 minutes ago

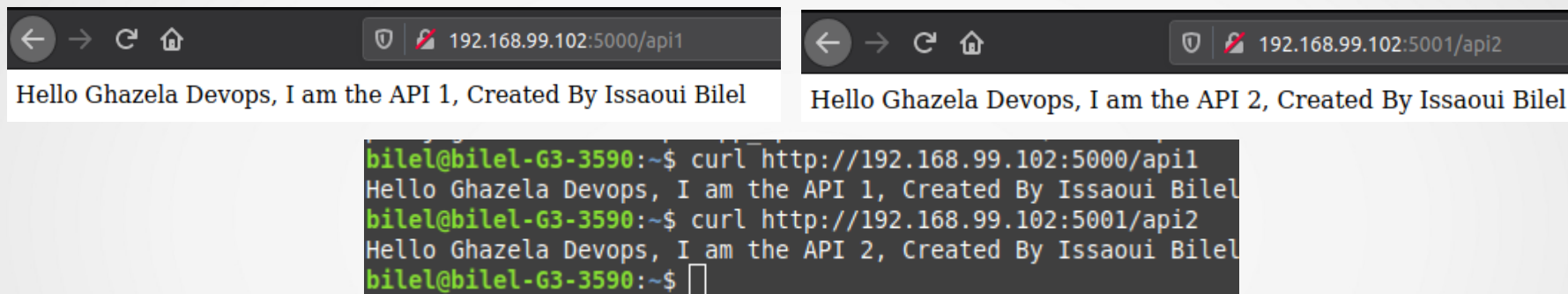
```
bilel@bilel-G3-3590:~$
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuelle de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Docker Swarm

D'après le résultat, on peut s'apercevoir que nous avons bien deux services distincts avec trois répliques pour chaque service. Même chose, pour accéder à nos services on peut utiliser soit l'adresse IP du nœud mymanager ou l'IP du nœud worker :



The image shows two browser screenshots and a terminal screenshot. The first browser screenshot shows a GET request to `192.168.99.102:5000/api1` with the response: "Hello Ghazela Devops, I am the API 1, Created By Issaoui Bilel". The second browser screenshot shows a GET request to `192.168.99.102:5001/api2` with the response: "Hello Ghazela Devops, I am the API 2, Created By Issaoui Bilel". The terminal screenshot shows the following commands and output:

```
bilel@bilel-g3-3590:~$ curl http://192.168.99.102:5000/api1
Hello Ghazela Devops, I am the API 1, Created By Issaoui Bilel
bilel@bilel-g3-3590:~$ curl http://192.168.99.102:5001/api2
Hello Ghazela Devops, I am the API 2, Created By Issaoui Bilel
bilel@bilel-g3-3590:~$
```

À partir de là, vous pouvez faire tout ce que vous avez appris dans les anciens chapitres et mettre à l'échelle votre application en modifiant simplement le fichier **docker-compose.yml** et d'exécuter simplement à nouveau la commande **docker stack deploy** pour prendre en compte vos nouvelles modifications.

Autre chose, si jamais vous rajoutez une nouvelle machine dans votre Swarm, il suffit juste de relancer la commande **docker stack deploy** pour que votre application tire parti des nouvelles ressources de votre nouveau nœud.

Si vous n'avez plus besoin de votre pile, vous pouvez alors la supprimer grâce à la commande suivante : `$ docker stack rm api-app`



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Docker Swarm

Conclusion

Je pense que comme moi vous êtes vraiment impressionné par la simplicité d'utilisation du Docker Swarm. L'un des plus gros avantages de Docker Swarm, c'est qu'il vous permet de vous concentrer sur le développement votre application et de ne pas vous soucier de l'endroit où cette dernière va s'exécuter. De plus les services sont constamment monitorés par Swarm. En plus du monitoring, Docker Swarm se charge de la réparation automatique, vous aidant ainsi à garder vos services de votre cluster en bon état de fonctionnement en comparant en permanence l'état désiré avec l'état actuel.

Docker Swarm offre d'autres fonctionnalités et je vous suggère de creuser davantage dans la documentation officielle. Pour information ce n'est pas le seul orchestrateur de conteneurs qui existe dans le marché, vous pouvez par exemple aussi vous intéresser à Kubernetes.

Comme à mon habitude, voici un récapitulatif des commandes Docker Swarm vu ensemble :

```
# Gestion du cluster Swarm
docker swarm
# Gestion des conteneurs uni-service
docker service
# Gestion des conteneurs multi-services
docker stack
# Gestion des nœuds
docker node
# Activer le mode Swarm
docker swarm init
# Joindre une machine au cluster Swarm
docker swarm join --token <token> <myvm ip>:<port>
# Lister les différents nœuds de votre Swarm
docker node ls
# Inspecter un nœud
docker node inspect <NODE NAME>
# Retirer un nœud de votre Swarm (ne supprime pas la VM)
docker node rm <NODE NAME>
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuelle de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Docker Swarm

```
# Créer un service
docker service create <IMAGE NAME>
  --name : nom du service
  --replicas <number> : nombre de tâches
  --publish published=<cible>,target=<source> : mapper un port
  --restart-condition=<condition> : condition de redémarrage en cas d'erreur
  --limit-memory <number> : limiter l'utilisation de la mémoire
  --limit-cpu <number> : limiter l'utilisation du CPU
# Visualiser l'état d'avancement de vos services Swarm
docker service ls
# lister les différentes tâches de votre service
docker service ps <SERVICE NAME>
# Mise en échelle des répliques de votre service
docker service scale <SERVICE NAME>=<NUMBER>
# Mise à jour de des conteneurs de votre service
docker service update --image <IMAGE NAME>:<TAG> <SERVICE NAME>
# Supprimer un service
docker service rm flaskc
# Déployer une nouvelle pile ou met à jour une pile existante
docker stack deploy -c <Docker Compose File> <STACK NAME>
# Lister tous les services de votre pile
docker stack services <STACK NAME>
# Répertoire les tâches de la pile
docker stack ps <STACK NAME>
# Supprimer tous les services de votre pile
docker stack rm <STACK NAME>
# Lister le nombre de services de votre pile
docker stack ls
```



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuelle de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Docker Swarm



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Docker Swarm



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Docker Swarm



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Docker Swarm



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Docker Swarm



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155

Docker Swarm



Mr. BILEL Issaoui, Formateur DevOps chez Ghazela Technology Academy, www.ghazelatc.com. Tél. +21654260000. Le support est à usage personnel, il est propriété intellectuel de l'académie, il n'est pas à usage commercial, contact@ghazelatc.com, +21671866142, +21654828018, +21627862155