Cahier TPs Docker

Rev 1.3 oct/2020

Table des matières

1 - Installation

Lab 1-1 : Installation sur CentOS

1/ Mettre à jour Centos

```
[root@centos1 ~]# yum update -y
Modules complémentaires chargés : fastestmirror
Loading mirror speeds from cached hostfile
 * base: fr.mirror.babylon.network
 * extras: centos.mirrors.ovh.net
 * updates: centos.mirrors.ovh.net
No packages marked for update
```

2/ Installer les utilitaires yum

```
[root@centos1 ~]# yum install -y yum-utils
Modules complémentaires chargés : fastestmirror
Loading mirror speeds from cached hostfile
 * base: fr.mirror.babylon.network
 * extras: fr.mirror.babylon.network
 * updates: centos.mirrors.ovh.net
Le paquet yum-utils-1.1.31-40.el7.noarch est déjà installé dans sa
dernière version
Rien à faire
```

3/ Ajouter le dépôt docker

```
[root@centos1 ~]# yum-config-manager –add-repo \
      https://download.docker.com/linux/centos/docker-ce.repo
Modules complémentaires chargés : fastestmirror
adding repo from: https://download.docker.com/linux/centos/docker-
ce.repo
grabbing file https://download.docker.com/linux/centos/docker-ce.repo to
/etc/yum.repos.d/docker-ce.repo
repo saved to /etc/yum.repos.d/docker-ce.repo
```

4/ Installer docker CE

```
[root@centos1 ~]# yum install docker-ce docker-ce-cli containerd.io -y
Modules complémentaires chargés : fastestmirror
Loading mirror speeds from cached hostfile
 * base: fr.mirror.babylon.network
 * extras: fr.mirror.babylon.network
 * updates: centos.mirrors.ovh.net
Résolution des dépendances
--> Lancement de la transaction de test
---> Le paquet docker-ce.x86_64 0:17.03.1.ce-1.el7.centos sera installé

.../...

Installé :
  docker-ce.x86_64 0:17.03.1.ce-1.el7.centos

Dépendances installées :
  audit-libs-python.x86_64 0:2.6.5-3.el7_3.1            checkpolicy.x86_64 0:2.5-4.el7
  docker-ce-selinux.noarch 0:17.03.1.ce-1.el7.centos    libcgroup.x86_64 0:0.41-11.el7
  libseccomp.x86_64 0:2.3.1-2.el7                       libselinux-python.x86_64 0:2.5-6.el7
  libsemanage-python.x86_64 0:2.5-5.1.el7_3             libtool-ltdl.x86_64 0:2.4.2-22.el7_3
  policycoreutils-python.x86_64 0:2.5-11.el7_3          python-IPy.noarch 0:0.75-6.el7
  setools-libs.x86_64 0:3.3.8-1.1.el7

Terminé !
```

5/ Démarrer et vérifier l'installation de docker

```
[root@centos1 ~]# systemctl start docker

[root@centos1 ~]# docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
78445dd45222: Pull complete
Digest:
sha256:c5515758d4c5e1e838e9cd307f6c6a0d620b5e07e6f927b07d05f6d12a1ac8d7
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working
correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker
Hub.
 3. The Docker daemon created a new container from that image which runs
the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which
sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
```

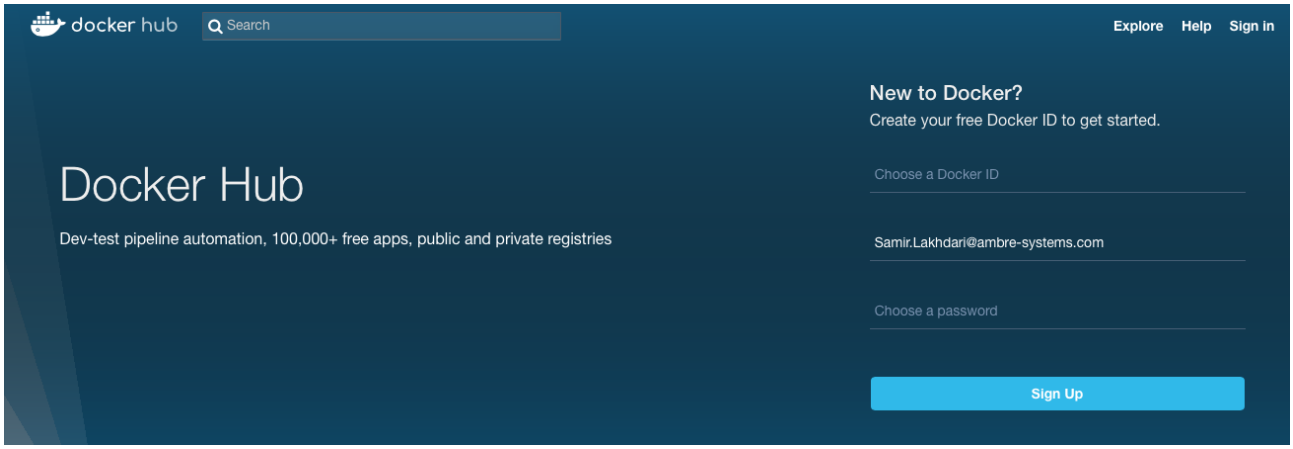
```
https://cloud.docker.com/
```

6/ afficher la version de docker

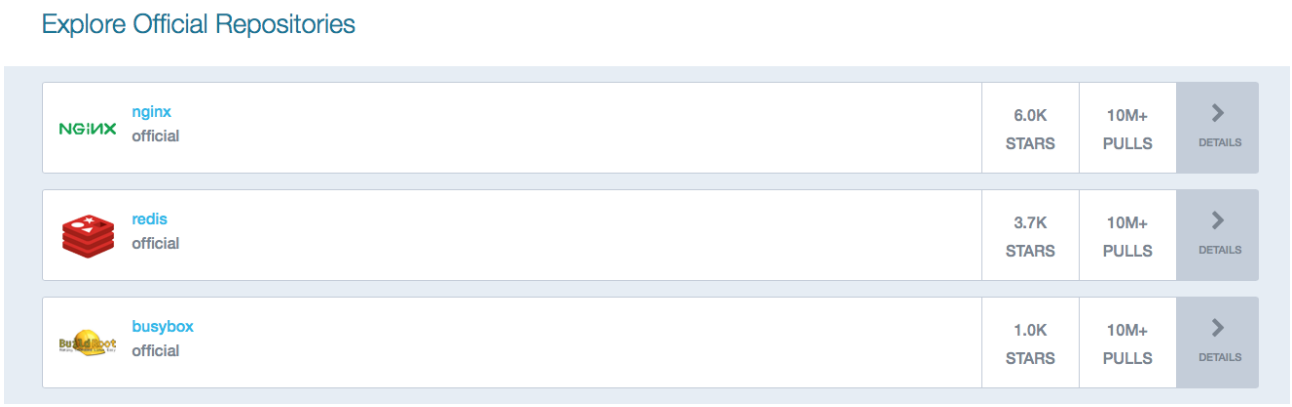
```
[root@centos1 ~]# docker version
Client:
 Version:      17.03.1-ce
 API version:  1.27
 Go version:   go1.7.5
 Git commit:   c6d412e
 Built:        Mon Mar 27 17:05:44 2017
 OS/Arch:      linux/amd64

Server:
 Version:      17.03.1-ce
 API version:  1.27 (minimum version 1.12)
 Go version:   go1.7.5
 Git commit:   c6d412e
 Built:        Mon Mar 27 17:05:44 2017
 OS/Arch:      linux/amd64
 Experimental: false
```

Lab 1-2 : Docker HUB

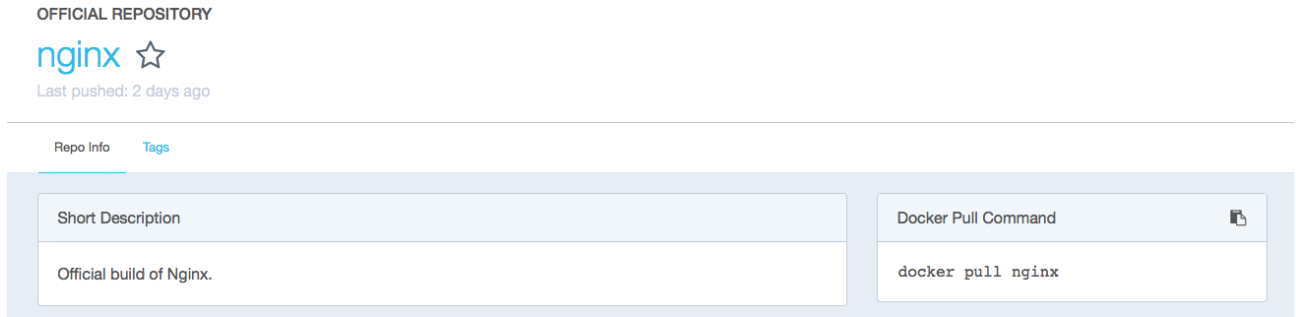
1/ Créer un compte sur le site https://hub.docker.com



2/ Cliquer sur `Explore` pour parcourir les repository officiels



3/ Cliquer sur `DETAILS` pour plus d'informations



Lab 1-3 : Images

1/ Télécharger une image à partir du HUB Docker

```
[root@centos1 ~]# docker pull ubuntu:14.04
14.04: Pulling from library/ubuntu
8f229c550c2e: Pull complete
8e1fb71e8df6: Pull complete
f75a34586856: Pull complete
8744e322b832: Pull complete
d5165bfce78f: Pull complete
Digest:
sha256:edf05697d8ea17028a69726b4b450ad48da8b29884cd640fec950c904bfb50ce
Status: Downloaded newer image for ubuntu:14.04


[root@centos1 ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED
SIZE
ubuntu              14.04               302fa07d8117        4 weeks ago
188 MB
hello-world         latest              48b5124b2768        4 months ago
1.84 kB
```

2 - Containers

Lab 2-1 : Exécuter un container

1/ Exécuter un container à partir de l'image ubuntu

```
[root@centos1 ~]# docker run ubuntu:14.04 echo "hello world"
hello world
```

2/ Exécuter cette commande et observer le PID de la commande ps

```
[root@centos1 ~]# docker run ubuntu:14.04 ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 09:25 ?        00:00:00 ps -ef
```

3/ Lister les containers en exécution

```
[root@centos1 ~]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS               NAMES
```

3/ Lister tous les containers

```
[root@centos1 ~]# docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED
STATUS                      PORTS               NAMES
329a4a4265d4        ubuntu:14.04        "ps -ef"                 2 minutes
ago       Exited (0) 2 minutes ago
ecstatic_mirzakhani
4cce702d15ad        ubuntu:14.04        "echo 'hello world'"     4 minutes
ago       Exited (0) 4 minutes ago                       naughty_bardeen
12ff38e588fe        hello-world         "/hello"                 7 minutes
ago       Exited (0) 7 minutes ago
upbeat_brahmagupta
```

Lab 2-2 : Accès au terminal d'un container

1/ Créez un conteneur à l'aide de l'image ubuntu 14.04 et connectez-vous au terminal

```
[root@centos1 ~]# docker run -it ubuntu:14.04 bash
root@2789725e7f65:/#
```

2/ Créez un fichier dans le conteneur puis sortez du conteneur

```
root@2789725e7f65:/# touch fic1

root@2789725e7f65:/# ls
bin  boot  dev  etc  fic1  home  lib  lib64  media  mnt  opt  proc  root
run  sbin  srv  sys  tmp  usr  var

root@2789725e7f65:/# exit
exit
[root@centos1 ~]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS               NAMES
```

3/ Exécutez une deuxième fis la commande :

```
[root@centos1 ~]# docker run -it ubuntu:14.04 bash

root@eafed0ffc27e:/# ls
bin  boot  dev  etc  home  lib  lib64  media  mnt  opt     proc  root
run  sbin  srv  sys  tmp  usr  var
root@eafed0ffc27e:/#
```

Que s'est-il passé ?

Lab 2-3 : Exécution en mode détaché

1/ Exécutez la commande suivante :

```
[root@centos1 ~]# docker run -d centos ping 127.0.0.1 -c 60
Unable to find image 'centos:latest' locally
latest: Pulling from library/centos
Digest:
sha256:bba1de7c9d900a898e3cadbae040dfe8a633c06bc104a0df76ae24483e03c077
Status: Downloaded newer image for centos:latest
2daefc5740000d7d026733baf3d74614caf8415eb1aa604451cfea6da7a6bf4f

[root@centos1 ~]# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
STATUS              PORTS               NAMES
2daefc574000        centos              "ping 127.0.0.1 -c 60"   5
seconds ago       Up 4 seconds
wizardly_colden
```

2/ Attendez quelques secondes puis lister les contenuers :

```
[root@centos1 ~]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS               NAMES
```

Lab 2-4 : S'attacher à un conteneur et se détacher d'un conteneur

1/ Exécutez la commande suivante :

```
[root@centos1 ~]# docker run -d centos ping 127.0.0.1 -c 60
4b4b1dab96d2021429a80ece3a18cb222feaaf7f8895157884f601adc6c3b91b

[root@centos1 ~]# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
STATUS              PORTS                   NAMES
4b4b1dab96d2        centos              "ping 127.0.0.1 -c 60"   13
seconds ago      Up 13 seconds
quirky_sammet

[root@centos1 ~]# docker attach 4b4b1dab96d2
64 bytes from 127.0.0.1: icmp_seq=26 ttl=64 time=0.054 ms
64 bytes from 127.0.0.1: icmp_seq=27 ttl=64 time=0.057 ms
64 bytes from 127.0.0.1: icmp_seq=28 ttl=64 time=0.057 ms
64 bytes from 127.0.0.1: icmp_seq=29 ttl=64 time=0.053 ms
64 bytes from 127.0.0.1: icmp_seq=30 ttl=64 time=0.054 ms
```

Appuyez sur CTRL-P-Q pour se détacher du conteneur, que se passe-t-il ?

2/ Exécutez la commande suivante :

```
[root@centos1 ~]# docker run -d -it centos ping 127.0.0.1 -c 60
1e1114dae615b92e1e79ba91fcfacb034b609036d7b65452b693a906b1ad3305

[root@centos1 ~]# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
STATUS              PORTS                   NAMES
1e1114dae615        centos              "ping 127.0.0.1 -c 60"   4
seconds ago       Up 4 seconds
goofy_snyder

[root@centos1 ~]# docker attach 1e1114dae615
64 bytes from 127.0.0.1: icmp_seq=18 ttl=64 time=0.056 ms
64 bytes from 127.0.0.1: icmp_seq=19 ttl=64 time=0.052 ms
64 bytes from 127.0.0.1: icmp_seq=20 ttl=64 time=0.055 ms
64 bytes from 127.0.0.1: icmp_seq=21 ttl=64 time=0.051 ms
64 bytes from 127.0.0.1: icmp_seq=22 ttl=64 time=0.052 ms
64 bytes from 127.0.0.1: icmp_seq=23 ttl=64 time=0.050 ms
64 bytes from 127.0.0.1: icmp_seq=24 ttl=64 time=0.049 ms
```

Appuyez sur CTRL-P-Q pour se détacher du conteneur

```
[root@centos1 ~]# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
```

```
STATUS              PORTS                    NAMES
1e1114dae615        centos              "ping 127.0.0.1 -c 60"   28
seconds ago      Up 27 seconds
goofy_snyder
```

Lab 2-5 : La commande exec

1/ Exécutez un conteneur en mode arrière plan :

```
[root@centos1 ~]# docker run -d -it centos bash
9d5a92f0a6e47006d9b8cfec89dcb12cb1dea5f657055674202993b916a42c1e

[root@centos1 ~]# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
STATUS              PORTS               NAMES
9d5a92f0a6e4        centos              "bash"                   5
seconds ago       Up 4 seconds
stupefied_babbage

[root@centos1 ~]# docker attach 9d5a92f0a6e4
[root@9d5a92f0a6e4 /]# ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 10:13 ?        00:00:00 bash
root        13     1  0 10:13 ?        00:00:00 ps -ef
[root@9d5a92f0a6e4 /]#
```

Appuyez sur CTRL-P-Q pour se détacher du conteneur

```
[root@centos1 ~]#
[root@centos1 ~]# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
STATUS              PORTS               NAMES
9d5a92f0a6e4        centos              "bash"                   29
seconds ago      Up 29 seconds
stupefied_babbage

[root@centos1 ~]# docker exec -it 9d5a92f0a6e4 bash
[root@9d5a92f0a6e4 /]# ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 10:13 ?        00:00:00 bash
root        18     0  0 10:14 ?        00:00:00 bash
root        30    18  0 10:14 ?        00:00:00 ps -ef
[root@9d5a92f0a6e4 /]#
[root@9d5a92f0a6e4 /]# exit
```

Remarquez les PPID

Lab 2-6 : La commande `logs`

1/ Exécutez un conteneur en mode arrière plan :

```
[root@centos1 ~]# docker run -d ubuntu:14.04 ping 127.0.0.1 -c 100
4ca6589443712b47acececa403d9656f132dfbd54302e46cf261841f194cc307
[root@centos1 ~]# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
STATUS              PORTS               NAMES
4ca658944371        ubuntu:14.04        "ping 127.0.0.1 -c..."   7
seconds ago       Up 7 seconds
friendly_feynman
[root@centos1 ~]# docker logs 4ca658944371
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.037 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.054 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.046 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.052 ms
64 bytes from 127.0.0.1: icmp_seq=5 ttl=64 time=0.054 ms
64 bytes from 127.0.0.1: icmp_seq=6 ttl=64 time=0.054 ms
64 bytes from 127.0.0.1: icmp_seq=7 ttl=64 time=0.043 ms
64 bytes from 127.0.0.1: icmp_seq=8 ttl=64 time=0.045 ms

[root@centos1 ~]# docker logs -f 4ca658944371
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.037 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.054 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.046 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.052 ms
64 bytes from 127.0.0.1: icmp_seq=5 ttl=64 time=0.054 ms
64 bytes from 127.0.0.1: icmp_seq=6 ttl=64 time=0.054 ms
64 bytes from 127.0.0.1: icmp_seq=7 ttl=64 time=0.043 ms
64 bytes from 127.0.0.1: icmp_seq=8 ttl=64 time=0.045 ms
64 bytes from 127.0.0.1: icmp_seq=9 ttl=64 time=0.053 ms
^C
[root@centos1 ~]# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
STATUS              PORTS               NAMES
4ca658944371        ubuntu:14.04        "ping 127.0.0.1 -c..."   51
seconds ago      Up 50 seconds
friendly_feynman
[root@centos1 ~]# docker logs --tail 10 -f 4ca658944371
64 bytes from 127.0.0.1: icmp_seq=58 ttl=64 time=0.056 ms
64 bytes from 127.0.0.1: icmp_seq=59 ttl=64 time=0.056 ms
64 bytes from 127.0.0.1: icmp_seq=60 ttl=64 time=0.058 ms
64 bytes from 127.0.0.1: icmp_seq=61 ttl=64 time=0.059 ms
```

Lab 2-7 : Les commandes `stop - start`

1/ Exécutez un conteneur en mode arrière plan :

```
[root@centos1 ~]# docker run -d -it alpine ping 127.0.0.1
Unable to find image 'alpine:latest' locally
latest: Pulling from library/alpine
df20fa9351a1: Pull complete
Digest:
sha256:185518070891758909c9f839cf4ca393ee977ac378609f700f60a771a2dfe321
Status: Downloaded newer image for alpine:latest
c818ba5a13d96b29c1b1458279f9156f9e237c31bac234b07f9b60ed6cee0ce2

[root@centos1 ~]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS               NAMES
c818ba5a13d9        alpine              "ping 127.0.0.1"    5 seconds
ago       Up 4 seconds                            keen_feistel
```

2/ Arrêtez le conteneur :

```
[root@centos1 ~]# docker stop c818ba5a13d9
c818ba5a13d9

[root@centos1 ~]# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS                          PORTS               NAMES
c818ba5a13d9        alpine              "ping 127.0.0.1"    32 seconds
ago      Exited (137) 5 seconds ago                       keen_feistel
```

3/ redémarrez le conteneur :

```
[root@centos1 ~]# docker start c818ba5a13d9
c818ba5a13d9

[root@centos1 ~]# docker ps -a
CCONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS               NAMES
c818ba5a13d9        alpine              "ping 127.0.0.1"    41 seconds
ago      Up 1 second                             keen_feistel
```

Lab 2-8 : La commande `inspect`

1/ La commande `inspect` permet d'afficher les propriétés :

```
[root@centos1 ~]# docker inspect dfbeb772aac7
[
    {
        "Id":
"dfbeb772aac7661129d9bea2ac800363e191b19cd8e3423b43e97fe214a00e14",
        "Created": "2017-05-15T16:57:53.538978324Z",
        "Path": "nginx",
        "Args": [
            "-g",
            "daemon off;"
        ],
        "State": {
            "Status": "running",
            "Running": true,
            "Paused": false,
            "Restarting": false,
            "OOMKilled": false,
            "Dead": false,
            "Pid": 5472,
            "ExitCode": 0,
            "Error": "",
            "StartedAt": "2017-05-15T17:04:44.374755739Z",
            "FinishedAt": "2017-05-15T17:01:17.479906146Z"
        },
        "Config": {
            "Hostname": "dfbeb772aac7",
            "Domainname": "",
            "User": "",
            "AttachStdin": false,
            "AttachStdout": false,
            "AttachStderr": false,
            "ExposedPorts": {
                "80/tcp": {}
            },
            "Tty": false,
            "OpenStdin": false,
            "StdinOnce": false,
            "Env": [

"PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
                "NGINX_VERSION=1.13.0-1~stretch",
                "NJS_VERSION=1.13.0.0.1.10-1~stretch"
            ],
            "Cmd": [
                "nginx",
                "-g",
                "daemon off;"
```

```
            ],
        "NetworkSettings": {
            "Bridge": "",
            "Ports": {
                "80/tcp": [
                    {
                        "HostIp": "0.0.0.0",
                        "HostPort": "32770"
                    }
                ]
            },
            "Gateway": "172.17.0.1",
            "IPAddress": "172.17.0.2",
            "IPPrefixLen": 16,
            "IPv6Gateway": "",
…/...
```

2/ La commande `inspect` et l'option `-format` :

```
[root@centos1 ~]# docker inspect
--format='{{.NetworkSettings.IPAddress}}' dfbeb772aac7
172.17.0.2

[root@centos1 ~]# docker inspect --format='{{.NetworkSettings}}'
dfbeb772aac7
{{ eef5ce9a601560e24a0b33633ad6848c41b55cb358fce824d1441c888c258bc1
false  0 map[80/tcp:[{0.0.0.0 32770}]]
/var/run/docker/netns/eef5ce9a6015 [] []}
{ee91cd1c31ecdc3be99b1df469a9b6a9668ced9e8ff7b0df4dd78618ec783637
172.17.0.1  0 172.17.0.2 16  02:42:ac:11:00:02}
map[bridge:0xc4200c4900]}

[root@centos1 ~]# docker inspect --format='{{json .NetworkSettings}}'
dfbeb772aac7
{"Bridge":"","SandboxID":"eef5ce9a601560e24a0b33633ad6848c41b55cb358fce8
24d1441c888c258bc1","HairpinMode":false,"LinkLocalIPv6Address":"","LinkL
ocalIPv6PrefixLen":0,"Ports":{"80/tcp":
[{"HostIp":"0.0.0.0","HostPort":"32770"}]},"SandboxKey":"/var/run/docker
/netns/eef5ce9a6015","SecondaryIPAddresses":null,"SecondaryIPv6Addresses
":null,"EndpointID":"ee91cd1c31ecdc3be99b1df469a9b6a9668ced9e8ff7b0df4dd
78618ec783637","Gateway":"172.17.0.1","GlobalIPv6Address":"","GlobalIPv6
PrefixLen":0,"IPAddress":"172.17.0.2","IPPrefixLen":16,"IPv6Gateway":"",
"MacAddress":"02:42:ac:11:00:02","Networks":{"bridge":
{"IPAMConfig":null,"Links":null,"Aliases":null,"NetworkID":"ca1e694382c6
c950adae359bc00befa30b91d891449de5c425f4169549cecb13","EndpointID":"ee91
cd1c31ecdc3be99b1df469a9b6a9668ced9e8ff7b0df4dd78618ec783637","Gateway":
"172.17.0.1","IPAddress":"172.17.0.2","IPPrefixLen":16,"IPv6Gateway":"",
"GlobalIPv6Address":"","GlobalIPv6PrefixLen":0,"MacAddress":"02:42:ac:11
:00:02"}}}
```

Lab 2-9 : Supprimer un conteneur

1/ Listez les conteneurs arrêtés :

```
[root@centos1 ~]# docker ps --filter='status=exited'
CONTAINER ID        IMAGE               COMMAND                  CREATED
STATUS                      PORTS               NAMES
dfbeb772aac7        nginx               "nginx -g 'daemon ..."   33
minutes ago      Exited (0) 42 seconds ago
quizzical_swirles
```

2/ Supprimez le conteneur arrêté :

```
[root@centos1 ~]# docker rm dfbeb772aac7
dfbeb772aac7
```

3/ Supprimez tous les conteneurs arrêtés :

```
[root@centos1 ~]# docker rm $(docker ps -aq)
6217fa8ecf9b
64a388190e2a
3931d3631805
```

3 - Gestion des images

Lab 3-1 : Modifications dans un container

1/ Exécutez un container à partir de l'image `centos` et installez `wget` :

```
[root@centos1 ~]# docker run -it centos:7 bash
[root@95bb28144c48 /]# yum install wget -y
Loaded plugins: fastestmirror, ovl
Loading mirror speeds from cached hostfile
 * base: fr.mirror.babylon.network
 * extras: fr.mirror.babylon.network
 * updates: centos.mirror.fr.planethoster.net
Resolving Dependencies
--> Running transaction check
---> Package wget.x86_64 0:1.14-13.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
================================================================================
 Package                       Arch                                  Version
Repository                     Size
================================================================================
================================================================================
Installing:
 wget                          x86_64                                1.14-
13.el7                            base                                546 k

Transaction Summary
================================================================================
================================================================================
Install  1 Package

Total download size: 546 k
Installed size: 2.0 M
Downloading packages:
warning: /var/cache/yum/x86_64/7/base/packages/wget-1.14-
13.el7.x86_64.rpm: Header V3 RSA/SHA256 Signature, key ID f4a80eb5:
NOKEY
Public key for wget-1.14-13.el7.x86_64.rpm is not installed
wget-1.14-13.el7.x86_64.rpm
| 546 kB  00:00:00
Retrieving key from file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
Importing GPG key 0xF4A80EB5:
 Userid     : "CentOS-7 Key (CentOS 7 Official Signing Key)
<security@centos.org>"
 Fingerprint: 6341 ab27 53d7 8a78 a7c2 7bb1 24c6 a8a7 f4a8 0eb5
 Package    : centos-release-7-3.1611.el7.centos.x86_64 (@CentOS)
 From       : /etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
Running transaction check
Running transaction test
Transaction test succeeded
```

```
Running transaction
  Installing : wget-1.14-13.el7.x86_64
1/1
install-info: No such file or directory for /usr/share/info/wget.info.gz
  Verifying  : wget-1.14-13.el7.x86_64
1/1

Installed:
  wget.x86_64 0:1.14-13.el7

Complete!
```

2/ Comparez le conteneur modifié et l'image de base :

```
[root@centos1 ~]# docker diff 95bb28144c48
C /etc
A /etc/wgetrc
C /root
A /root/.bash_history
A /usr/bin/wget
C /var
C /var/lib
C /var/lib/rpm
C /var/lib/yum
C /var/lib/yum/history
C /var/lib/yum/history/2017-05-10
A /var/lib/yum/history/2017-05-10/3
A /var/lib/yum/history/2017-05-10/3/config-repos
A /var/lib/yum/history/2017-05-10/3/saved_tx
A /var/lib/yum/history/2017-05-10/3/config-main
A /var/lib/yum/history/2017-05-10/4
A /var/lib/yum/history/2017-05-10/4/config-main
A /var/lib/yum/history/2017-05-10/4/config-repos
A /var/lib/yum/history/2017-05-10/4/saved_tx
C /var/lib/yum/history/history-2017-05-10.sqlite
C /var/lib/yum/history/history-2017-05-10.sqlite-journal
C /var/lib/yum/rpmdb-indexes
A /var/lib/yum/rpmdb-indexes/conflicts
A /var/lib/yum/rpmdb-indexes/file-requires
A /var/lib/yum/rpmdb-indexes/obsoletes
A /var/lib/yum/rpmdb-indexes/pkgtups-checksums
A /var/lib/yum/rpmdb-indexes/version
C /var/lib/yum/yumdb
A /var/lib/yum/yumdb/w/062ccdaa45d226931f12106e692458f16fb179e6-wget-
1.14-13.el7-x86_64/ts_install_langs
A /var/lib/yum/yumdb/w/062ccdaa45d226931f12106e692458f16fb179e6-wget-
1.14-13.el7-x86_64/var_infra
A /var/lib/yum/yumdb/w/062ccdaa45d226931f12106e692458f16fb179e6-wget-
1.14-13.el7-x86_64/releasever
C /var/log
C /var/log/yum.log
```

Lab 3-2 : Créer une nouvelle image

1/ Créez une nouvelle image à partir du conteneur modifié :

```
[root@centos1 ~]# docker commit 95bb28144c48 masociete/moncentos:1.0
sha256:1df7429bcb148fd50b9177a40c7296cbbc1ca31c288b9469d8a218b8e41b009e

[root@centos1 ~]# docker images
REPOSITORY            TAG                 IMAGE ID            CREATED
SIZE
masociete/moncentos   1.0                 1df7429bcb14        5 seconds
ago       281 MB
centos                7                   8140d0c64310        3 days ago
193 MB
centos                latest              8140d0c64310        3 days ago
193 MB
nginx                 latest              3448f27c273f        5 days ago
109 MB
ubuntu                14.04               302fa07d8117        4 weeks
ago         188 MB
hello-world           latest              48b5124b2768        4 months
ago        1.84 kB
```

2/ Créez un conteneur à partir de la nouvelle image :

```
[root@centos1 ~]# docker run -it masociete/moncentos:1.0 bash
[root@bf8d4d6074de /]# wget
wget: missing URL
Usage: wget [OPTION]... [URL]...

Try `wget --help' for more options.
```

Lab 3-3 : Dockerfile

1/ Créez un fichier Dockerfile dans un nouveau répertoire :

```
[root@centos1 ~]# mkdir moncentos
[root@centos1 ~]# cd moncentos
[root@centos1 moncentos]# vi Dockerfile
FROM centos:7
RUN yum update
RUN yum install wget -y

[root@centos1 moncentos]# docker build -t moncentos .
Sending build context to Docker daemon 2.048 kB
Step 1/3 : FROM centos:7
 ---> 8140d0c64310
Step 2/3 : RUN yum update
 ---> Running in 6a6a584f5b8a
Loaded plugins: fastestmirror, ovl
Determining fastest mirrors
 * base: mirror.neify.es
 * extras: mirrors.ircam.fr
 * updates: mirrors.ircam.fr
No packages marked for update
 ---> 6031f9d5dbbb
Removing intermediate container 6a6a584f5b8a
Step 3/3 : RUN yum install wget -y
 ---> Running in 89c2e2557c76
Loaded plugins: fastestmirror, ovl
Loading mirror speeds from cached hostfile
 * base: mirror.neify.es
 * extras: mirrors.ircam.fr
 * updates: mirrors.ircam.fr
Resolving Dependencies
--> Running transaction check
---> Package wget.x86_64 0:1.14-13.el7 will be installed
--> Finished Dependency Resolution

.../...

Running transaction
  Installing : wget-1.14-13.el7.x86_64
1/1
install-info: No such file or directory for /usr/share/info/wget.info.gz
  Verifying  : wget-1.14-13.el7.x86_64
1/1

Installed:
  wget.x86_64 0:1.14-13.el7

Complete!
 ---> a07cb8ea7398
Removing intermediate container 89c2e2557c76
```

```
Successfully built a07cb8ea7398

[root@centos1 moncentos]# docker images
REPOSITORY            TAG                 IMAGE ID            CREATED
SIZE
moncentos             latest              a07cb8ea7398        5 seconds
ago       282 MB
masociete/moncentos   1.0                 1df7429bcb14        8 minutes
ago       281 MB
centos                7                   8140d0c64310        3 days ago
193 MB
centos                latest              8140d0c64310        3 days ago
193 MB
nginx                 latest              3448f27c273f        5 days ago
109 MB
ubuntu                14.04               302fa07d8117        4 weeks
ago         188 MB
hello-world           latest              48b5124b2768        4 months
ago        1.84 kB
```

2/ Modifiez le fichier Dockerfile et ajoutez l'installation d'un nouveau paquet, remarquez l'utilisation du cache :

```
[root@centos1 moncentos]# cat Dockerfile
FROM centos:7
RUN yum update
RUN yum install wget -y
RUN yum install zip -y

[root@centos1 moncentos]# docker build -t moncentos .
Sending build context to Docker daemon 2.048 kB
Step 1/4 : FROM centos:7
 ---> 8140d0c64310
Step 2/4 : RUN yum update
 ---> Using cache
 ---> 6031f9d5dbbb
Step 3/4 : RUN yum install wget -y
 ---> Using cache
 ---> a07cb8ea7398
Step 4/4 : RUN yum install zip -y
 ---> Running in e21877fea220
Loaded plugins: fastestmirror, ovl
Loading mirror speeds from cached hostfile
 * base: mirror.neify.es
 * extras: mirrors.ircam.fr
 * updates: mirrors.ircam.fr
Resolving Dependencies
--> Running transaction check
---> Package zip.x86_64 0:3.0-11.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
========
 Package        Arch              Version                 Repository
Size
================================================================================
========
Installing:
 zip            x86_64            3.0-11.el7              base
260 k

Transaction Summary
================================================================================
========
Install  1 Package

Total download size: 260 k
Installed size: 796 k
Downloading packages:
```

```
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : zip-3.0-11.el7.x86_64
1/1
  Verifying  : zip-3.0-11.el7.x86_64
1/1

Installed:
  zip.x86_64 0:3.0-11.el7

Complete!
 ---> e6adaba320e0
Removing intermediate container e21877fea220
Successfully built e6adaba320e0
```

3/ Visualisez l'historique de la construction des image :

```
[root@centos1 moncentos]# docker history moncentos
IMAGE               CREATED             CREATED BY
SIZE                COMMENT
e6adaba320e0        7 minutes ago       /bin/sh -c yum install zip -y
20.1 MB
a07cb8ea7398        17 minutes ago      /bin/sh -c yum install wget -y
20 MB
6031f9d5dbbb        17 minutes ago      /bin/sh -c yum update
69.3 MB
8140d0c64310        3 days ago          /bin/sh -c #(nop)  CMD
["/bin/bash"]            0 B
<missing>           3 days ago          /bin/sh -c #(nop)  LABEL
name=CentOS Base ...   0 B
<missing>           3 days ago          /bin/sh -c #(nop) ADD
file:f3be3f14a2136b0...   193 MB
```

4/ Modifiez le ficher Dockerfile comme ceci, et que remarquez-vous :

```
[root@centos1 moncentos]# cat Dockerfile
FROM centos:7
RUN yum update
RUN yum install wget zip -y

[root@centos1 moncentos]# docker build -t moncentos .
Sending build context to Docker daemon 2.048 kB
Step 1/3 : FROM centos:7
 ---> 8140d0c64310
Step 2/3 : RUN yum update
 ---> Using cache
 ---> 6031f9d5dbbb
Step 3/3 : RUN yum install wget zip -y
 ---> Running in b3afbfe2f833
Loaded plugins: fastestmirror, ovl
Loading mirror speeds from cached hostfile
 * base: mirror.neify.es
 * extras: mirrors.ircam.fr
 * updates: mirrors.ircam.fr
Resolving Dependencies
--> Running transaction check
---> Package wget.x86_64 0:1.14-13.el7 will be installed
---> Package zip.x86_64 0:3.0-11.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package         Arch              Version                 Repository
Size
```

```
================================================================================
========
Installing:
 wget            x86_64            1.14-13.el7            base
546 k
 zip             x86_64            3.0-11.el7             base
260 k

Transaction Summary
================================================================================
========
Install  2 Packages
  Installing : zip-3.0-11.el7.x86_64
1/2
  Installing : wget-1.14-13.el7.x86_64
2/2
Installed:
  wget.x86_64 0:1.14-13.el7               zip.x86_64 0:3.0-11.el7

Complete!
 ---> b7c5fb141e81
Removing intermediate container b3afbfe2f833
Successfully built b7c5fb141e81
```

4 - Réseau docker

Lab 4-1 : Utilisation du réseau par défaut

1/ Lancez un conteneur en mode daemon :

```
[root@centos1 ~]# docker run --name mycentos1 -d -it centos:7 bash
b7ca2432f21ff213bb7a92ed826ff65e5d067d6377b4cb4da09b81697ddb1e52

[root@centos1 ~]# docker network inspect bridge
[
    {
        "Name": "bridge",
        "Id":
"0a0eaece6487cf24674b19640c6a5e4dff46c7fcc114172e6a3c7ee6a24981f3",
        "Created": "2017-05-18T12:59:03.343980732-04:00",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
            "Config": [
                {
                    "Subnet": "172.17.0.0/16",
                    "Gateway": "172.17.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Containers": {

"b7ca2432f21ff213bb7a92ed826ff65e5d067d6377b4cb4da09b81697ddb1e52": {
                "Name": "mycentos1",
                "EndpointID":
"11f810809753675a5a7e07e653c3b7eaf2bf76389c1f075cbe06f3dad3289565",
                "MacAddress": "02:42:ac:11:00:02",
                "IPv4Address": "172.17.0.2/16",
                "IPv6Address": ""
            }
        },
        "Options": {
            "com.docker.network.bridge.default_bridge": "true",
            "com.docker.network.bridge.enable_icc": "true",
            "com.docker.network.bridge.enable_ip_masquerade": "true",
            "com.docker.network.bridge.host_binding_ipv4": "0.0.0.0",
            "com.docker.network.bridge.name": "docker0",
            "com.docker.network.driver.mtu": "1500"
        },
        "Labels": {}
    }
]
```

2/ Lancez un autre conteneur et testez le réseau :

```
[root@centos1 ~]# docker run --name mycentos2  -it centos:7 bash
[root@fe73b8e62f72 /]# ping 172.17.0.2
PING 172.17.0.2 (172.17.0.2) 56(84) bytes of data.
64 bytes from 172.17.0.2: icmp_seq=1 ttl=64 time=0.084 ms
64 bytes from 172.17.0.2: icmp_seq=2 ttl=64 time=0.087 ms
^C
--- 172.17.0.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.084/0.085/0.087/0.009 ms
```

```
[root@fe73b8e62f72 /]# ping mycentos1
ping: mycentos1: Name or service not known
```

Lab 4-2 : Création d'un bridge

1/ Créez un bridge :

```
[root@centos1 ~]# docker network  create --driver bridge mybridge
c173166780b2a706bd0d80d1c36816e0f636b12f489640644a3c80e7b39c7e46

[root@centos1 ~]# docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
0a0eaece6487        bridge              bridge              local
36d4cba75cb7        host                host                local
c173166780b2        mybridge            bridge              local

[root@centos1 ~]# docker run --name mycentos3 --net=mybridge -d -it
centos:7 bash
eb9b92481282e772379a66c74f5b7723b35583fe180a386b2d7f821ef50a0417

[root@centos1 ~]# docker run --name mycentos4 --net=mybridge -it
centos:7 bash
[root@d24e13f3f36f /]# ping mycentos3
PING mycentos3 (172.21.0.2) 56(84) bytes of data.
64 bytes from mycentos3.mybridge (172.21.0.2): icmp_seq=1 ttl=64
time=0.060 ms
64 bytes from mycentos3.mybridge (172.21.0.2): icmp_seq=2 ttl=64
time=0.088 ms
64 bytes from mycentos3.mybridge (172.21.0.2): icmp_seq=3 ttl=64
time=0.089 ms
64 bytes from mycentos3.mybridge (172.21.0.2): icmp_seq=4 ttl=64
time=0.087 ms
64 bytes from mycentos3.mybridge (172.21.0.2): icmp_seq=5 ttl=64
time=0.085 ms
^C
--- mycentos3 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3999ms
rtt min/avg/max/mdev = 0.060/0.081/0.089/0.015 ms
```

Lab 4-3 : Connexion à plusieurs réseaux

1/ Créez un bridge :

```
[root@centos1 ~]# docker exec -it mycentos3 bash
[root@ eb9b92481282 /]# ping mycentos1
ping: mycentos1: Name or service not known


[root@centos1 ~]# docker network connect mybridge mycentos1

[root@centos1 ~]# docker exec -it mycentos3 bash
[root@eb9b92481282 /]# ping mycentos1
PING mycentos1 (172.21.0.3) 56(84) bytes of data.
64 bytes from mycentos1.mybridge (172.21.0.3): icmp_seq=1 ttl=64
time=0.115 ms
64 bytes from mycentos1.mybridge (172.21.0.3): icmp_seq=2 ttl=64
time=0.084 ms
64 bytes from mycentos1.mybridge (172.21.0.3): icmp_seq=3 ttl=64
time=0.085 ms
64 bytes from mycentos1.mybridge (172.21.0.3): icmp_seq=4 ttl=64
time=0.086 ms
^C
--- mycentos1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3000ms
rtt min/avg/max/mdev = 0.084/0.092/0.115/0.016 ms
```

Lab 4-4 : Mapping de port

1/ Lancez un conteneur avec mapping manuel :

```
[root@centos1 ~]# docker run -d -p 80:80 -p 90:8080 nginx
cb8a814718547d67f61a13ad00afc52c279514bc109e173c2078006023b04f58

[root@centos1 ~]# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
STATUS              PORTS                                    NAMES
cb8a81471854        nginx               "nginx -g 'daemon ..."   3
seconds ago       Up 2 seconds        0.0.0.0:80->80/tcp, 0.0.0.0:90-
>8080/tcp   jovial_visvesvaraya

[root@centos1 ~]# docker port cb8a81471854
80/tcp -> 0.0.0.0:80
8080/tcp -> 0.0.0.0:90
```

2/ Lancez un conteneur avec mapping automatique :

```
[root@centos1 ~]# docker run -d -P nginx
aa8379b2e9034102ccdda87d76be405d0c484572aeede94816abfe534da17c2d

[root@centos1 ~]# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
STATUS              PORTS                    NAMES
aa8379b2e903        nginx               "nginx -g 'daemon ..."   4
seconds ago       Up 3 seconds        0.0.0.0:32771->80/tcp
hardcore_jepsen

[root@centos1 ~]# docker port aa8379b2e903
80/tcp -> 0.0.0.0:32771
```

3/ Mapping de port dans Dockerfile :

```
[root@centos1 monimage]# cat Dockerfile
FROM centos:7
RUN yum install -y wget
EXPOSE 80 8080
RUN mkdir /data -p
RUN echo "Mes donnees" > /data/test
VOLUME /data

[root@centos1 monimage]# docker build -t monimage .
Sending build context to Docker daemon 2.048 kB
Step 1/6 : FROM centos:7
 ---> 8140d0c64310
Step 2/6 : RUN yum install -y wget
 ---> Running in 081f90b1d52c
Loaded plugins: fastestmirror, ovl
Determining fastest mirrors
 * base: centos.mirrors.ovh.net
…/...
 ---> Running in 3d4a557ff671
```

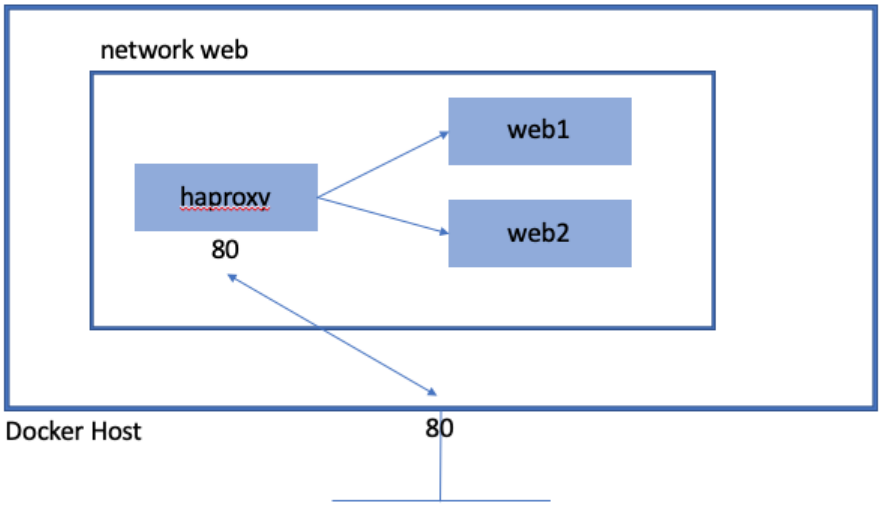
```
 ---> f11f404334b3
Removing intermediate container 3d4a557ff671
Successfully built f11f404334b3
```

```
[root@centos1 monimage]# docker run -d -it -P monimage
78d4ef6bc5395d1f302164fe08ce3478ec976ee398ebc973bd33f2a661cf396c

[root@centos1 monimage]# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
STATUS              PORTS                                            NAMES
78d4ef6bc539        monimage            "/bin/bash"              8 seconds ago       Up 7
seconds        0.0.0.0:32773->80/tcp, 0.0.0.0:32772->8080/tcp   quizzical_kilby
aa8379b2e903        nginx               "nginx -g 'daemon ..."   5 minutes ago       Up 5
minutes        0.0.0.0:32771->80/tcp                            hardcore_jepsen

[root@centos1 monimage]# docker port 78d4ef6bc539
8080/tcp -> 0.0.0.0:32772
80/tcp -> 0.0.0.0:32773
```

Lab 4-5 : HAPROXY

Nous allons créer une application avec 3 conteneurs dans un réseau privé :

- web1 : un serveur nginx non mappé qui doit afficher Server ONE
- web2 : un serveur nginx non mappé qui doit afficher Server TWO
- myhaproxy : un serveur haproxy mappé, qui recevra en FrontEnd les requêtes HTTP des clients sur le port 80 du Docker Host et qui redirigera en BackEnd vers les 2 serveurs web1 et web2



1/ Créez un réseau de type bridge nommé “web“ :

```
[centos@host11 ~]$ docker network create web
17b1c64f833b6a816b2ec29b1abc649c90a99e79e61c7fd74560e3d38c89be09

[centos@host11 ~]$ docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
81ace09760c9        bridge              bridge              local
3bdb47615925        host                host                local
fa98a03a1254        none                null                local
17b1c64f833b        web                 bridge              local
```

2/ Créez 2 images “nginx1“ et “nginx2“ avec des fichiers index.html spécifiques :

```
[centos@host11 ~]$ cat web1/Dockerfile
FROM nginx:1.19-alpine
COPY index.html /usr/share/nginx/html

[[centos@host11 ~]$ cat web1/index.html
Server ONE

[centos@host11 ~]$ docker build -t nginx1 web1
Sending build context to Docker daemon  3.072kB
Step 1/2 : FROM nginx:1.19-alpine
1.19-alpine: Pulling from library/nginx
df20fa9351a1: Already exists
091a6e3499e9: Pull complete
b4bea01b9731: Pull complete
62c992d61d2c: Pull complete
b675ffa804eb: Pull complete
Digest:
sha256:5fcbe9a6b09b6525651d1e5d5a2df373eec1a13c75f0eaa724a369f43ce589f4
Status: Downloaded newer image for nginx:1.19-alpine
 ---> bd53a8aa5ac9
Step 2/2 : COPY index.html /usr/share/nginx/html
 ---> d30d9f9fef2d
Successfully built d30d9f9fef2d
Successfully tagged nginx1:latest

[centos@host11 ~]$ cat web2/Dockerfile
FROM nginx:1.19-alpine
COPY index.html /usr/share/nginx/html

[centos@host11 ~]$ cat web2/index.html
Server TWO

[centos@host11 ~]$ docker build -t nginx2 web2
Sending build context to Docker daemon  3.072kB
Step 1/2 : FROM nginx:1.19-alpine
 ---> bd53a8aa5ac9
Step 2/2 : COPY index.html /usr/share/nginx/html
 ---> 0d4ce5042a9a
Successfully built 0d4ce5042a9a
Successfully tagged nginx2:latest


[centos@host11 ~]$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED
SIZE
nginx2              latest              0d4ce5042a9a        About a
minute ago   22.3MB
nginx1              latest              d30d9f9fef2d        4 minutes
ago        22.3MB
nginx               1.19-alpine         bd53a8aa5ac9        3 days ago
22.3MB
```

3/ Créez l' image myhaproxy avec de fichier de configuration fourni :

```
[centos@host11 ~]$ cat myhaproxy/Dockerfile
FROM haproxy:1.7
COPY haproxy.cfg /usr/local/etc/haproxy/haproxy.cfg

[centos@host11 ~]$ cat myhaproxy/haproxy.cfg
frontend http_frontend
        bind *:80
        default_backend http_backend
backend http_backend
        mode http
        balance roundrobin
        server server1 web1:80 check
        server server2 web2:80 check

[centos@host11 ~]$ docker build -t myhaproxy myhaproxy
Sending build context to Docker daemon  3.072kB
Step 1/2 : FROM haproxy:1.7
1.7: Pulling from library/haproxy
d121f8d1c412: Pull complete
e80f03437728: Pull complete
1e7f8e283348: Pull complete
Digest:
sha256:3b0f07fbeddeecea551eaaf8bd729083a4243979ff029ff52dc2717fe8562c27
Status: Downloaded newer image for haproxy:1.7
 ---> 01dc8306566c
Step 2/2 : COPY haproxy.cfg /usr/local/etc/haproxy/haproxy.cfg
 ---> c81ed6af4981
Successfully built c81ed6af4981
Successfully tagged myhapoxy:latest

[centos@host11 ~]$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED
SIZE
myhaproxy            latest              c81ed6af4981        5 seconds
ago       82.8MB
nginx2              latest              0d4ce5042a9a        5 minutes
ago       22.3MB
nginx1              latest              d30d9f9fef2d        8 minutes
ago       22.3MB
nginx               1.19-alpine         bd53a8aa5ac9        3 days ago
22.3MB
haproxy             1.7                 01dc8306566c        4 weeks ago
82.8MB
```

4/ Lancez les 3 conteneurs :

```
[centos@host11 ~]$ docker run -d --network=web --name=web1 nginx1
19db91952e50e1d76d4fb53cc908178e0da4ebfb9ad0acd486aa89bd63259cdb

[centos@host11 ~]$ docker run -d --network=web --name=web2 nginx2
```

```
b28b75df34f852dfe13f60e331670241837bf06ad21197800915e2877b13bfae

[centos@host11 ~]$ docker run -d --network=web --name=myhaproxy -p 80:80
myhaproxy

[centos@host11 ~]$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
STATUS              PORTS               NAMES
d1c09d4ba21c        myhaproxy           "/docker-entrypoint.…"   5
seconds ago       Up 5 seconds        0.0.0.0:80->80/tcp   myhaproxy
b28b75df34f8        nginx2              "/docker-entrypoint.…"   36
seconds ago      Up 36 seconds       80/tcp               web2
19db91952e50        nginx1              "/docker-entrypoint.…"   43
seconds ago      Up 43 seconds       80/tcp               web1

[centos@host11 ~]$ curl 0
Server ONE
[centos@host11 ~]$ curl 0
Server TWO
[centos@host11 ~]$ curl 0
Server ONE
```

5 - Gestion des volumes

Lab 5-1 : Création des volumes

1/ Créez un volume `vol1`:

```
[root@centos1 ~]# docker volume  create --name vol1
vol1
```

2/ Listez les volumes:

```
[root@centos1 ~]# docker volume ls
DRIVER              VOLUME NAME
local               vol1
```

3/ Montez le volume `vol1` dans un conteneur centos:

```
[root@centos1 ~]# docker run -it -v vol1:/datas centos bash
[root@4c6dad9a32de /]# df -h
Filesystem               Size  Used Avail Use% Mounted on
overlay                   14G  2.7G   12G  19% /
tmpfs                    920M     0  920M   0% /dev
tmpfs                    920M     0  920M   0% /sys/fs/cgroup
/dev/mapper/centos-root   14G  2.7G   12G  19% /datas
shm                       64M     0   64M   0% /dev/shm
tmpfs                    920M     0  920M   0% /sys/firmware
```

4/ Créez un ficher dans conteneur centos:

```
[root@4c6dad9a32de /]# cat >> /datas/fic1
11111
22222
```

5/ Faites un `exit` pour sortir du conteneur centos:

```
[root@4c6dad9a32de /]# exit
```

6/ Créez une image à partir du conteneur :

```
[root@centos1 ~]# docker commit 4c6dad9a32de mycentos:2.0
sha256:f5599a7dcaf4a2d8ce49d5d783a8f2c9e372cb793f3429a377897a110df4279b
```

7/ Lancez un conteneur à partir de la nouvelle image, que remarquez-vous :

```
[root@centos1 ~]# docker run -it mycentos:2.0 bash
[root@7030d2429bef /]# df
```

```
Filesystem              1K-blocks    Used Available Use% Mounted on
overlay                  14530560 2739588  11790972  19% /
tmpfs                      942064       0    942064   0% /dev
tmpfs                      942064       0    942064   0% /sys/fs/cgroup
/dev/mapper/centos-root  14530560 2739588  11790972  19% /etc/hosts
shm                         65536       0     65536   0% /dev/shm
tmpfs                      942064       0    942064   0% /sys/firmware
```

Lab 5-2 : Utilisation des volumes

1/ Listez les volumes :

```
[root@centos1 ~]# docker volume ls
DRIVER              VOLUME NAME
local               vol1
```

2/ Affichez les caractéristiques d'un volume :

```
[root@centos1 ~]# docker volume inspect vol1
[
    {
        "Driver": "local",
        "Labels": {},
        "Mountpoint": "/var/lib/docker/volumes/vol1/_data",
        "Name": "vol1",
        "Options": {},
        "Scope": "local"
    }
]
```

3/ Vérifiez la persistance des données dans le volume :

```
[root@centos1 ~]# cd /var/lib/docker/volumes/vol1/_data
[root@centos1 _data]# ls
fic1
[root@centos1 _data]# cat fic1
11111
22222
```

4/ Lancez un nouveau conteneur et accéder au même volume et créez un autre fichier :

```
[root@centos1 _data]# docker run -it -v vol1:/datas centos bash
[root@f089099d3945 /]# cat /datas/fic1
11111
22222

[root@f089099d3945 /]# cat >> /datas/fic2
abc
def
```

5/ Passez le conteneur en arrière plan par CTRL+P+Q puis connectez vous par un exec :

```
[root@centos1 _data]# docker exec -it  f089099d3945 bash
```

```
[root@f089099d3945 /]# ls /datas
fic1  fic2
```

Lab 5-3 : Suppression des volumes

1/ Listez les volumes :

```
[root@centos1 ~]# docker volume ls
DRIVER              VOLUME NAME
local               vol1
```

2/ Supprimez le volume `vol1` :

```
[root@centos1 ~]# docker volume rm vol1
Error response from daemon: unable to remove volume: remove vol1: volume is in use -
f089099d39450501ed623261447ac22476c4402a8a65bb094924cfb9216ace62

[root@centos1 ~]# docker stop f089099d3945
f089099d3945

[root@centos1 ~]# docker rm f089099d3945
f089099d3945

[root@centos1 ~]# docker volume rm vol1
vol1
```

Lab 5-4 : Montez un volume host

1/ Montez un volume host :

```
[root@centos1 ~]# mkdir html_datas
[root@centos1 ~]# cd html_datas/
[root@centos1 html_datas]# cat >> index.html
Hello On My Web Server

[root@centos1 html_datas]# docker run -it -v /root/html_datas:/data/www
centos
[root@4981354ced38 /]# cd /data/www/
[root@4981354ced38 www]# cat index.html
Hello On My Web Server
[root@4981354ced38 www]# exit
exit
```

2/ Volumes pour les logs :

```
[root@centos1 ~]# docker volume create --name nginx_logs
nginx_logs

[root@centos1 ~]#  docker run -d -P --name nginx_server -v
/root/html_datas:/usr/share/nginx/html -v nginx_logs:/var/log/nginx
nginx
d00640df5f65335c4f511a2135feb030b48a60af530fa661c1931182d6bffe59

[root@centos1 ~]# docker ps
CONTAINER ID        IMAGE                  COMMAND
CREATED             STATUS              PORTS
NAMES
d00640df5f65        nginx                  "nginx -g 'daemon ..."   6
seconds ago       Up 6 seconds        0.0.0.0:32770->80/tcp
nginx_server

[root@centos1 ~]# curl http://0.0.0.0:32770
Hello On My Web Server

[root@centos1 ~]# docker exec -it nginx_server bash
root@d00640df5f65:/# cd /var/log/nginx/
root@d00640df5f65:/var/log/nginx# ls
access.log  error.log
root@d00640df5f65:/var/log/nginx# tail -f access.log
```

3/ Inspectez les logs :

```
[root@centos1 ~]# docker volume inspect nginx_logs
[
    {
        "Driver": "local",
        "Labels": {},
```

```
        "Mountpoint": "/var/lib/docker/volumes/nginx_logs/_data",
        "Name": "nginx_logs",
        "Options": {},
        "Scope": "local"
    }
]
```

```
[root@centos1 ~]# cd /var/lib/docker/volumes/nginx_logs/_data
[root@centos1 _data]# ls
access.log  error.log
```

Lab 5-5 : Les volumes dans un Dockerfile

1/ Montez un volume host :

```
[root@centos1 ~]# mkdir html_datas
[root@centos1 ~]# cd html_datas/
[root@centos1 html_datas]# cat >> index.html
Hello On My Web Server

[root@centos1 html_datas]# docker run -it -v /root/html_datas:/data/www
centos
[root@4981354ced38 /]# cd /data/www/
[root@4981354ced38 www]# cat index.html
Hello On My Web Server
[root@4981354ced38 www]# exit
exit
```

Lab 5-6 : HAPROXY avec les volumes

Reprendre l'exercice HPROXY, mais lancez les conteneurs avec les images standards, et pour modifier les fichiers index.html et haproxy.cfg, on passe par le montage de volumes des répertoires et fichiers créés précédemment :

1/ Lancez les serveurs web1 et web2 :

```
[centos@host11 ~]$ docker run -d --network=web --name=web1 -v
/home/centos/web1:/usr/share/nginx/html nginx:1.19-alpine
2763a5fb05f6a5338015b303990118e06c730124adb2047ad6a25692edf6d7e7

[centos@host11 ~]$ docker run -d --network=web --name=web2 -v
/home/centos/web2:/usr/share/nginx/html nginx:1.19-alpine
85702c3f11939aeb90e39ea851c7ade1353a5d5089c61c5bfe73d449b824fc3b
```

2/ Lancez le haproxy :

```
[centos@host11 ~]$ docker run -d --network=web --name=myhaproxy -p 80:80
-v /home/centos/myhapoxy/haproxy.cfg:/usr/local/etc/haproxy/haproxy.cfg
haproxy:1.7
33ab2b7ade33aa3a79e2de1a64d09718f5c1a496d8a9a9daaebf5072bdc165b5
```

6 - Registre privé

Lab 6-1 : Configurer le Docker Daemon

1/ Modifiez la configuration du Docker daemon pour autoriser un registre privé non secure :

```
[centos@host11 ~]$ ip a
2: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP
group default qlen 1000
    link/ether 00:50:56:8c:b3:20 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.11/24 brd 192.168.1.255 scope global noprefixroute
ens192
       valid_lft forever preferred_lft forever

[centos@host11 ~]$ sudo vi /etc/docker/daemon.json
{
     "insecure-registries": ["192.168.1.11:5000"]
}

[centos@host11 ~]$ sudo systemctl restart docker
[centos@host11 ~]$ docker info
.../...
 Registry: https://index.docker.io/v1/
 Labels:
 Experimental: false
 Insecure Registries:
  192.168.1.11:5000
  127.0.0.0/8
 Live Restore Enabled: false
```

2/ Créez un volume pour le stockage des images dans le registry :

```
[centos@host11 ~]$ docker volume create registry
registry

[centos@host11 ~]$ docker volume ls
DRIVER              VOLUME NAME
local               registry
```

3/ Lancez le conteneur registry :

```
[centos@host11 ~]$ docker run -d -p 5000:5000 --restart=always --name
registry -v registry:/var/lib/registry registry:2.7
Unable to find image 'registry:2.7' locally
2.7: Pulling from library/registry
cbdbe7a5bc2a: Pull complete
47112e65547d: Pull complete
46bcb632e506: Pull complete
c1cc712bcecd: Pull complete
3db6272dcbfa: Pull complete
Digest:
```

```
sha256:8be26f81ffea54106bae012c6f349df70f4d5e7e2ec01b143c46e2c03b9e551d
Status: Downloaded newer image for registry:2.7
b842ba9788a113bc261f444d7b2479aca5994317a32c612ba4aa2ec1278d9529

[centos@host11 ~]$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
STATUS              PORTS                    NAMES
b842ba9788a1        registry:2.7        "/entrypoint.sh /etc…"   5
seconds ago       Up 4 seconds        0.0.0.0:5000->5000/tcp   registry
```

4/ Créez une image ou un tag associé à l'URL de votre registry :

```
[centos@host11 ~]$ docker tag myhaproxy 192.168.1.11:5000/myhaproxy

[centos@host11 ~]$ docker images
REPOSITORY                    TAG                 IMAGE ID
CREATED             SIZE
192.168.1.11:5000/myhaproxy   latest              c81ed6af4981        36
minutes ago      82.8MB
myhaproxy                     latest              c81ed6af4981        36
minutes ago      82.8MB
registry                      2.7                 2d4f4b5309b1        3
months ago        26.2MB
```

5/ Poussez l'image sur votre registry :

```
[centos@host11 ~]$ docker push 192.168.1.11:5000/myhaproxy
The push refers to repository [192.168.1.11:5000/myhaproxy]
a72dfd64436d: Pushed
bb8cfb63b6af: Pushed
aa2495af74c7: Pushed
07cab4339852: Pushed
latest: digest:
sha256:251c14ce839e7332fc55dd633de449b0ff22cf778631554f66bbaffdc94b239e
size: 1154
```

5/ Vérifiez le contenu du registry :

```
[centos@host11 ~]$ curl 192.168.1.11:5000/v2/_catalog
{"repositories":["myhaproxy"]}

[centos@host11 ~]$ curl 192.168.1.11:5000/v2/myhaproxy/tags/list
{"name":"myhaproxy","tags":["latest"]}
```

7 - Docker Machine

Lab 7-1 : Installation de Docker-Machine

1/ Installez docker-machine sur votre poste de travail, à partir de l'URL suivante :

https://docs.docker.com/machine/install-machine/

```
[centos@host11 ~]$
base=https://github.com/docker/machine/releases/download/v0.16.0 &&
>   curl -L $base/docker-machine-$(uname -s)-$(uname -m) >/tmp/docker-
machine &&
>   sudo mv /tmp/docker-machine /usr/local/bin/docker-machine &&
>   chmod +x /usr/local/bin/docker-machine
  % Total    % Received % Xferd  Average Speed   Time    Time     Time
Current
                                 Dload  Upload   Total   Spent    Left
Speed
100   651  100   651    0     0   1086      0 --:--:-- --:--:-- --:--:--
1085
100 26.8M  100 26.8M    0     0  3674k      0  0:00:07  0:00:07 --:--:--
6229k
```

2/ Vérifiez l'installation et la version :

```
[centos@host11 ~]$ docker-machine version
docker-machine version 0.16.0, build 702c267f
```

```
[centos@host11 ~]$ docker-machine ls
NAME   ACTIVE   DRIVER   STATE   URL   SWARM   DOCKER   ERRORS
```

Lab 7-2 : Création d'un Docker Host de type Virtualbox

1/ Lancez la création d'un Docker Host :

```
[centos@host11 ~]$ docker-machine create --driver virtualbox node1
Creating CA: /home/centos/.docker/machine/certs/ca.pem
Creating client certificate: /home/centos/.docker/machine/certs/cert.pem
Running pre-create checks...
(node1) Image cache directory does not exist, creating it at
/home/centos/.docker/machine/cache...
(node1) No default Boot2Docker ISO found locally, downloading the latest
release...
(node1) Latest release for github.com/boot2docker/boot2docker is
v19.03.12
(node1) Downloading /home/centos/.docker/machine/cache/boot2docker.iso
from
https://github.com/boot2docker/boot2docker/releases/download/v19.03.12/b
oot2docker.iso...
(node1)
0%....10%....20%....30%....40%....50%....60%....70%....80%....90%....100
%
Creating machine...
(node1) Copying /home/centos/.docker/machine/cache/boot2docker.iso to
/home/centos/.docker/machine/machines/node1/boot2docker.iso...
(node1) Creating VirtualBox VM...
(node1) Creating SSH key...
(node1) Starting the VM...
(node1) Check network to re-create if needed...
(node1) Found a new host-only adapter: "vboxnet0"
(node1) 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 node1
```

2/ Listez les hosts :

```
[centos@host11 ~]$ docker-machine ls
NAME    ACTIVE   DRIVER       STATE     URL
SWARM   DOCKER      ERRORS
node1   -        virtualbox   Running   tcp://192.168.99.100:2376
```

```
v19.03.12
```

Lab 7-3 : Configuration de l'environnement

1/ Exécution de commandes à partir de ssh :

```
[centos@host11 ~]$ docker-machine ssh node1
   ( '>')
  /) TC (\   Core is distributed with ABSOLUTELY NO WARRANTY.
 (/-_--_-\)           www.tinycorelinux.net

docker@node1:~$ exit
logout

[centos@host11 ~]$ docker-machine ssh node1 docker version
Client: Docker Engine - Community
 Version:           19.03.12
 API version:       1.40
 Go version:        go1.13.10
 Git commit:        48a66213fe
 Built:             Mon Jun 22 15:42:53 2020
 OS/Arch:           linux/amd64
 Experimental:      false

Server: Docker Engine - Community
 Engine:
  Version:          19.03.12
  API version:      1.40 (minimum version 1.12)
  Go version:       go1.13.10
  Git commit:       48a66213fe
  Built:            Mon Jun 22 15:49:35 2020
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          v1.2.13
  GitCommit:        7ad184331fa3e55e52b890ea95e65ba581ae3429
 runc:
  Version:          1.0.0-rc10
  GitCommit:        dc9208a3303feef5b3839f4323d9beb36df0a9dd
 docker-init:
  Version:          0.18.0
  GitCommit:        fec3683
[centos@host11 ~]$
```

2/ Lancez une serveur web sur le node1 :

```
[centos@host11 ~]$ docker-machine ssh node1 docker run -d -p 80:80 httpd
Unable to find image 'httpd:latest' locally
latest: Pulling from library/httpd
d121f8d1c412: Already exists
9cd35c2006cf: Pulling fs layer
b6b9dec6e0f8: Pulling fs layer
fc3f9b55fcc2: Pulling fs layer
802357647f64: Pulling fs layer
```

```
3f9b55fcc2: Download complete
fc3f9b55fcc2: Pull complete
802357647f64: Pull complete
Digest:
sha256:5ce7c20e45b407607f30b8f8ba435671c2ff80440d12645527be670eb8ce1961
Status: Downloaded newer image for httpd:latest
be7fb1ab69e0d46ff72b3c017c74f374e0c0cb183587675c2915b6cdb5be1e45
```

```
[centos@host11 ~]$ docker-machine ssh node1 docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS               NAMES
be7fb1ab69e0        httpd               "httpd-foreground"   5 seconds
ago       Up 4 seconds        0.0.0.0:80->80/tcp   inspiring_thompson

[centos@host11 ~]$ docker-machine ls
NAME    ACTIVE   DRIVER       STATE     URL
SWARM   DOCKER      ERRORS
node1   -        virtualbox   Running   tcp://192.168.99.100:2376
v19.03.12

[centos@host11 ~]$ curl 192.168.99.100
<html><body><h1>It works!</h1></body></html>
```

3/ Configurez les variables d'environnement :

```
[centos@host11 ~]$ docker-machine env node1
export DOCKER_TLS_VERIFY="1"
export DOCKER_HOST="tcp://192.168.99.100:2376"
export DOCKER_CERT_PATH="/home/centos/.docker/machine/machines/node1"
export DOCKER_MACHINE_NAME="node1"
# Run this command to configure your shell:
# eval $(docker-machine env node1)

[centos@host11 ~]$ docker-machine ls
NAME    ACTIVE   DRIVER       STATE     URL
SWARM   DOCKER      ERRORS
node1   -        virtualbox   Running   tcp://192.168.99.100:2376
v19.03.12

[centos@host11 ~]$ eval $(docker-machine env node1)
[centos@host11 ~]$ docker-machine ls
NAME    ACTIVE   DRIVER       STATE     URL
SWARM   DOCKER      ERRORS
node1   *        virtualbox   Running   tcp://192.168.99.100:2376
v19.03.12

Notes : remarquez le caractère étoile dans la colonne ACTIVE

[centos@host11 ~]$ docker run -d -p 81:80 httpd
0957e8bb5d8190eeefdf1858c94f09e4dce221eeed03ce0c4bc0a655859c5f1e

[centos@host11 ~]$ curl 192.168.99.100:81
<html><body><h1>It works!</h1></body></html>
```

8 - SWARM

Lab 8-1 : Création du cluster SWARM

1/ Initialisez le premier nœud :

```
[centos@host11 ~]$ docker-machine ls
NAME    ACTIVE   DRIVER       STATE     URL
SWARM   DOCKER      ERRORS
node1   *        virtualbox   Running   tcp://192.168.99.100:2376
v19.03.12
node2   -        virtualbox   Running   tcp://192.168.99.101:2376
v19.03.12
node3   -        virtualbox   Running   tcp://192.168.99.102:2376
v19.03.12

[centos@host11 ~]$ eval $(docker-machine env node1)
[centos@host11 ~]$ docker swarm init --advertise-addr 192.168.99.100
Swarm initialized: current node (gj55ngip7s9gqbmqvmjo7sra4) is now a
manager.

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

    docker swarm join --token SWMTKN-1-
1nez4gns8b1ew6lbkog2hkq0ux2bvfgx5td2f4jfudm0q04h4q-
alu4tlxog94q2ay2uunsp04zn 192.168.99.100:2377

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

[centos@host11 ~]$ docker node ls
ID                            HOSTNAME            STATUS
AVAILABILITY        MANAGER STATUS      ENGINE VERSION
gj55ngip7s9gqbmqvmjo7sra4 *   node1               Ready
Active              Leader              19.03.12
```

2/ Initialisez les autres nœuds en tant que Workers :

```
[centos@host11 ~]$ eval $(docker-machine env node2)
[centos@host11 ~]$ docker swarm join --token SWMTKN-1-
1nez4gns8b1ew6lbkog2hkq0ux2bvfgx5td2f4jfudm0q04h4q-
alu4tlxog94q2ay2uunsp04zn 192.168.99.100:2377
This node joined a swarm as a worker.

[centos@host11 ~]$ eval $(docker-machine env node3)
[centos@host11 ~]$ docker swarm join --token SWMTKN-1-
1nez4gns8b1ew6lbkog2hkq0ux2bvfgx5td2f4jfudm0q04h4q-
alu4tlxog94q2ay2uunsp04zn 192.168.99.100:2377
This node joined a swarm as a worker.

[centos@host11 ~]$ eval $(docker-machine env node1)
[centos@host11 ~]$ docker node ls
ID                          HOSTNAME            STATUS              AVAILABILITY        MANAGER
STATUS      ENGINE VERSION
```

```
gj55ngip7s9gqbmqvmjo7sra4 *   node1               Ready               Active              Leader
19.03.12
hffnp9jzz1k8d2mn7pjj5q9p2     node2               Ready               Active
19.03.12
q13l8kwhs7curabef6hk2p2dx     node3               Ready               Active
19.03.12
```

3/ Promouvoir les autres nœuds en tant que Managers:

```
[centos@host11 ~]$ docker node promote node2
Node node2 promoted to a manager in the swarm.

[centos@host11 ~]$ docker node promote node3
Node node3 promoted to a manager in the swarm.

[centos@host11 ~]$ docker node ls
ID                            HOSTNAME            STATUS              AVAILABILITY        MANAGER
STATUS      ENGINE VERSION
gj55ngip7s9gqbmqvmjo7sra4 *   node1               Ready               Active              Leader
19.03.12
hffnp9jzz1k8d2mn7pjj5q9p2     node2               Ready               Active              Reachable
19.03.12
q13l8kwhs7curabef6hk2p2dx     node3               Ready               Active              Reachable
19.03.12
```

Lab 8-2 : Création de services SWARM

1/ Créez un service web de type replicated avec 2 replicas, basé sur un service httpd :

```
[centos@host11 ~]$ docker service create --mode=replicated --replicas=2
--name=web -p 80:80 httpd
6ige3via7m8dge0z72iv7j8gn
overall progress: 1 out of 2 tasks
overall progress: 2 out of 2 tasks
1/2: running   [==================================================>]
2/2: running   [==================================================>]
verify: Waiting 3 seconds to verify that tasks are stable...
verify: Service converged

[centos@host11 ~]$ docker service ls
ID                  NAME                MODE                REPLICAS
IMAGE               PORTS
6ige3via7m8d        web                 replicated          2/2
httpd:latest        *:80->80/tcp

[centos@host11 ~]$ docker service ps web
ID                  NAME                IMAGE               NODE
DESIRED STATE       CURRENT STATE            ERROR               PORTS
873sqg41abiz        web.1               httpd:latest        node1
Running             Running 24 seconds ago
d7vnxngk1u27        web.2               httpd:latest        node2
Running             Running 16 seconds ago
```

2/ Faites évoluer le service web à 4 replicas :

```
[centos@host11 ~]$ docker service scale web=4
web scaled to 4
overall progress: 4 out of 4 tasks
1/4: running   [==================================================>]
2/4: running   [==================================================>]
3/4: running   [==================================================>]
4/4: running   [==================================================>]
verify: Service converged

[centos@host11 ~]$ docker service ps web
ID                  NAME                IMAGE               NODE
DESIRED STATE       CURRENT STATE               ERROR
PORTS
873sqg41abiz        web.1               httpd:latest        node1
Running             Running about a minute ago
d7vnxngk1u27        web.2               httpd:latest        node2
Running             Running about a minute ago
gd944b61tu2v        web.3               httpd:latest        node3
Running             Running 12 seconds ago
i9ctsxi8a1d6        web.4               httpd:latest        node3
Running             Running 12 seconds ago
```

3/ Créez un service debug de type global, basé sur alpine :

```
[centos@host11 ~]$ docker service create --mode=global --name=debug
alpine sleep 3600
ozlddzt73dznww75tr71b00yz
overall progress: 3 out of 3 tasks
gj55ngip7s9g: running
[==================================================>]
q13l8kwhs7cu: running
[==================================================>]
hffnp9jzz1k8: running
[==================================================>]
verify: Service converged

[centos@host11 ~]$ docker service ls
ID                  NAME                MODE                REPLICAS
IMAGE               PORTS
ozlddzt73dzn        debug               global              3/3
alpine:latest
6ige3via7m8d        web                 replicated          4/4
httpd:latest        *:80->80/tcp

[centos@host11 ~]$ docker service ps debug
ID                  NAME                                    IMAGE
NODE                DESIRED STATE       CURRENT STATE            ERROR
PORTS
3ipcpr4xj49v        debug.q13l8kwhs7curabef6hk2p2dx   alpine:latest
node3               Running             Running 20 seconds ago
y7xvkw5xrjv0        debug.hffnp9jzz1k8d2mn7pjj5q9p2   alpine:latest
node2               Running             Running 20 seconds ago
2wsvcv6khxa0        debug.gj55ngip7s9gqbmqvmjo7sra4   alpine:latest
node1               Running             Running 20 seconds ago
```

Lab 8-3 : Cas de pannes

1/ Pour simuler une panne, arrêtez le node1 :

```
[centos@host11 ~]$ docker-machine ls
NAME    ACTIVE   DRIVER       STATE     URL
SWARM   DOCKER      ERRORS
node1   *        virtualbox   Running   tcp://192.168.99.100:2376
v19.03.12
node2   -        virtualbox   Running   tcp://192.168.99.101:2376
v19.03.12
node3   -        virtualbox   Running   tcp://192.168.99.102:2376
v19.03.12

[centos@host11 ~]$ docker node ls
ID                            HOSTNAME            STATUS
AVAILABILITY        MANAGER STATUS      ENGINE VERSION
gj55ngip7s9gqbmqvmjo7sra4 *   node1               Ready
Active              Leader              19.03.12
hffnp9jzz1k8d2mn7pjj5q9p2     node2               Ready
Active              Reachable           19.03.12
q13l8kwhs7curabef6hk2p2dx     node3               Ready
Active              Reachable           19.03.12

[centos@host11 ~]$ docker-machine stop node1
Stopping "node1"...
Machine "node1" was stopped.

[centos@host11 ~]$ docker-machine ls
NAME    ACTIVE   DRIVER       STATE     URL
SWARM   DOCKER      ERRORS
node1   -        virtualbox   Stopped
Unknown
node2   -        virtualbox   Running   tcp://192.168.99.101:2376
v19.03.12
node3   -        virtualbox   Running   tcp://192.168.99.102:2376
v19.03.12

[centos@host11 ~]$ eval $(docker-machine env node2)
[centos@host11 ~]$ docker node ls
ID                            HOSTNAME            STATUS
AVAILABILITY        MANAGER STATUS      ENGINE VERSION
gj55ngip7s9gqbmqvmjo7sra4     node1               Unknown
Active              Unreachable         19.03.12
hffnp9jzz1k8d2mn7pjj5q9p2 *   node2               Ready
Active              Reachable           19.03.12
q13l8kwhs7curabef6hk2p2dx     node3               Ready
Active              Leader              19.03.12
```

2/ Vérifiez les services :

```
[centos@host11 ~]$ docker service ls
ID                  NAME                MODE                REPLICAS
```

```
IMAGE               PORTS
ozlddzt73dzn        debug               global              2/2
alpine:latest
6ige3via7m8d        web                 replicated          4/4
httpd:latest        *:80->80/tcp

[centos@host11 ~]$ docker service ps web
ID                  NAME                IMAGE               NODE
DESIRED STATE       CURRENT STATE               ERROR
PORTS
noj3uz7jvbcv        web.1               httpd:latest        node2
Running             Running about a minute ago
873sqg41abiz         \_ web.1           httpd:latest        node1
Shutdown            Running 11 minutes ago
d7vnxngk1u27        web.2               httpd:latest        node2
Running             Running 2 minutes ago
gd944b61tu2v        web.3               httpd:latest        node3
Running             Running about a minute ago
i9ctsxi8a1d6        web.4               httpd:latest        node3
Running             Running about a minute ago

[centos@host11 ~]$ docker service ps debug
ID                  NAME                                    IMAGE
NODE                DESIRED STATE       CURRENT STATE           ERROR
PORTS
3ipcpr4xj49v        debug.q13l8kwhs7curabef6hk2p2dx   alpine:latest
node3               Running             Running 5 minutes ago
y7xvkw5xrjv0        debug.hffnp9jzz1k8d2mn7pjj5q9p2   alpine:latest
node2               Running             Running 5 minutes ago
2wsvcv6khxa0        debug.gj55ngip7s9gqbmqvmjo7sra4   alpine:latest
node1               Shutdown            Running 5 minutes ago
```

3/ redémarrez le node1 est regardez ce qui se passe :

```
[centos@host11 ~]$ docker-machine start node1
Starting "node1"...
(node1) Check network to re-create if needed...
(node1) Waiting for an IP...
Machine "node1" 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.

[centos@host11 ~]$ docker service ls
ID                  NAME                MODE                REPLICAS
IMAGE               PORTS
ozlddzt73dzn        debug               global              3/3
alpine:latest
6ige3via7m8d        web                 replicated          4/4
httpd:latest        *:80->80/tcp
```

```
[centos@host11 ~]$ docker service ps web
ID                  NAME                IMAGE               NODE
DESIRED STATE       CURRENT STATE            ERROR               PORTS
noj3uz7jvbcv        web.1               httpd:latest        node2
Running             Running 4 minutes ago
873sqg41abiz         \_ web.1           httpd:latest        node1
Shutdown            Shutdown 13 seconds ago
d7vnxngk1u27        web.2               httpd:latest        node2
Running             Running 4 minutes ago
gd944b61tu2v        web.3               httpd:latest        node3
Running             Running 4 minutes ago
i9ctsxi8a1d6        web.4               httpd:latest        node3
Running             Running 4 minutes ago

[centos@host11 ~]$ docker service ps debug
ID                  NAME                                IMAGE
NODE                DESIRED STATE       CURRENT STATE             ERROR
PORTS
xl83ewu5mq8o        debug.gj55ngip7s9gqbmqvmjo7sra4   alpine:latest
node1               Running             Running 22 seconds ago
3ipcpr4xj49v        debug.q13l8kwhs7curabef6hk2p2dx   alpine:latest
node3               Running             Running 8 minutes ago
y7xvkw5xrjv0        debug.hffnp9jzz1k8d2mn7pjj5q9p2   alpine:latest
node2               Running             Running 8 minutes ago
2wsvcv6khxa0        debug.gj55ngip7s9gqbmqvmjo7sra4   alpine:latest
node1               Shutdown            Shutdown 23 seconds ago
```

9 - Docker Compose

Lab 9-1 : Installer docker-compose

1/ Installez docker-compose sur votre poste de travail, à partir de l'URL suivante :

`https://docs.docker.com/compose/install/`

```
[centos@host11 ~]$ sudo curl -L
"https://github.com/docker/compose/releases/download/1.27.4/docker-
compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
  % Total    % Received % Xferd  Average Speed   Time    Time     Time
Current
                                 Dload  Upload   Total   Spent    Left
Speed
100   651  100   651    0     0   1671      0 --:--:-- --:--:-- --:--:--
1669
100 11.6M  100 11.6M    0     0  3178k      0  0:00:03  0:00:03 --:--:--
4635k

[centos@host11 ~]$ sudo chmod +x /usr/local/bin/docker-compose

[centos@host11 ~]$ docker-compose version
docker-compose version 1.27.4, build 40524192
docker-py version: 4.3.1
CPython version: 3.7.7
OpenSSL version: OpenSSL 1.1.0l  10 Sep 2019
```

2/ Reprenez le TP haproxy et reformulez le avec docker-compose :

```
[centos@host11 ~]$ cat docker-compose.yml
version: '3'
services:
  myhaproxy:
    image: haproxy:1.7
    volumes:
      -
"/home/centos/myhaproxy/haproxy.cfg:/usr/local/etc/haproxy/haproxy.cfg"
    ports:
      - "80:80"
    depends_on:
      - web1
      - web2
    networks:
      - web
  web1:
    image: nginx:1.19-alpine
    container_name: web1
    volumes:
      - "/home/centos/web1:/usr/share/nginx/html"
    networks:
      - web
  web2:
```

```
    image: nginx:1.19-alpine
    container_name: web2
    volumes:
      - "/home/centos/web2:/usr/share/nginx/html"
    networks:
      - web
networks:
  web:
```

3/ Lancez les services docker-compose :

```
[centos@host11 ~]$ eval $(docker-machine env -u)

[centos@host11 ~]$ docker-compose up -d
Creating network "centos_web" with the default driver
Creating web2 ... done
Creating web1 ... done
Creating centos_myhaproxy_1 ... done


[centos@host11 ~]$ docker-compose ps
       Name                     Command               State
Ports
------------------------------------------------------------------------------
--------
centos_myhaproxy_1   /docker-entrypoint.sh hapr ...   Up
0.0.0.0:80->80/tcp
web1                 /docker-entrypoint.sh ngin ...   Up      80/tcp
web2                 /docker-entrypoint.sh ngin ...   Up      80/tcp

[centos@host11 ~]$ curl 0
Server ONE
[centos@host11 ~]$ curl 0
Server TWO
```

Lab 9-2 : Utiliser docker-compose en mode SWARM

1/ Créez une pile compose/swarm pour lancer un service global debug basé sur alpine, et un service replicated avec 4 replicas basé sur httpd :

```
[centos@host11 ~]$ cat docker-compose.yml
version: "3"
services:
  alpine:
    image: alpine
    command: ["sleep","3600"]
    deploy:
      mode: global
  web:
    image: httpd
    deploy:
      replicas: 4
      restart_policy:
        condition: on-failure
      resources:
        limits:
          cpus: "0.1"
          memory: 50M
    ports:
      - "80:80"

[centos@host11 ~]$ eval $(docker-machine env node1)
[centos@host11 ~]$ docker-machine ls
NAME    ACTIVE   DRIVER       STATE     URL
SWARM   DOCKER      ERRORS
node1   *        virtualbox   Running   tcp://192.168.99.100:2376
v19.03.12
node2   -        virtualbox   Running   tcp://192.168.99.101:2376
v19.03.12
node3   -        virtualbox   Running   tcp://192.168.99.102:2376
v19.03.12

[centos@host11 ~]$ docker node ls
ID                            HOSTNAME            STATUS
AVAILABILITY        MANAGER STATUS      ENGINE VERSION
gj55ngip7s9gqbmqvmjo7sra4 *   node1               Ready
Active              Reachable           19.03.12
hffnp9jzz1k8d2mn7pjj5q9p2     node2               Ready
Active              Reachable           19.03.12
q13l8kwhs7curabef6hk2p2dx     node3               Ready
Active              Leader              19.03.12
```

2/ Lancez la pile :

```
[centos@host11 ~]$ docker stack deploy -c docker-compose.yml web_stack
Creating network web_stack_default
Creating service web_stack_alpine
```

```
Creating service web_stack_web
```

2/ Visualisez le résultat :

```
[centos@host11 ~]$ docker stack ls
NAME                SERVICES            ORCHESTRATOR
web_stack           2                   Swarm

[centos@host11 ~]$ docker service ls
ID                  NAME                MODE                REPLICAS
IMAGE               PORTS
el5djsjm6h7w        web_stack_alpine    global              3/3
alpine:latest
02d6axyvuexb        web_stack_web       replicated          4/4
httpd:latest        *:80->80/tcp

[centos@host11 ~]$ docker stack ps web_stack
ID                  NAME                                    IMAGE
NODE                DESIRED STATE       CURRENT STATE           ERROR
PORTS
mc9j37qystur        web_stack_alpine.q13l8kwhs7curabef6hk2p2dx
alpine:latest       node3               Running             Running 14
seconds ago
ro0jw1h2rtg5        web_stack_alpine.hffnp9jzz1k8d2mn7pjj5q9p2
alpine:latest       node2               Running             Running 14
seconds ago
1x25imz3ik1m        web_stack_alpine.gj55ngip7s9gqbmqvmjo7sra4
alpine:latest       node1               Running             Running 14
seconds ago
ixt7f65f900z        web_stack_web.1
httpd:latest        node2               Running             Running 12
seconds ago
i2hha5exlzzs        web_stack_web.2
httpd:latest        node3               Running             Running 13
seconds ago
uuc4n8djodt6        web_stack_web.3
httpd:latest        node1               Running             Running 12
seconds ago
3z3c3xkdx5it        web_stack_web.4
httpd:latest        node2               Running             Running 12
seconds ago
```

Lab 9-3 : Ajouter les services Visualizer et Cadvisor

1/ Visualizer est un outil graphique qui permet de visualizer les services sur un cluster SWARM. Vous trouverez ce service à l'adresse suivante :

```
https://hub.docker.com/r/dockersamples/visualizer
```

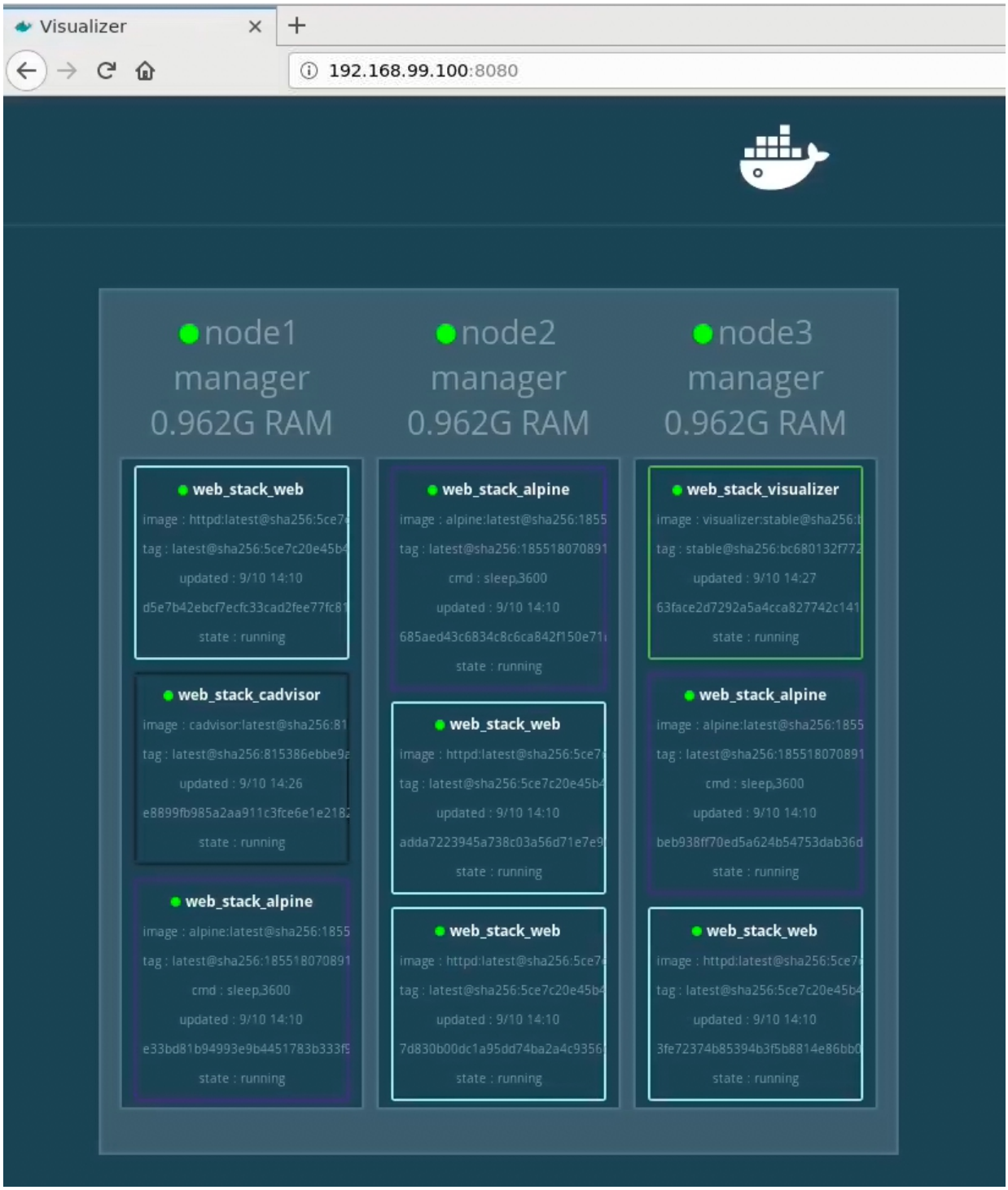
2/ Cadvisor est un outil Google qui remonte non seulement les consommations de ressources du système linux global, mais aussi des conteneurs.Vous trouverez ce service à l'adresse suivante :

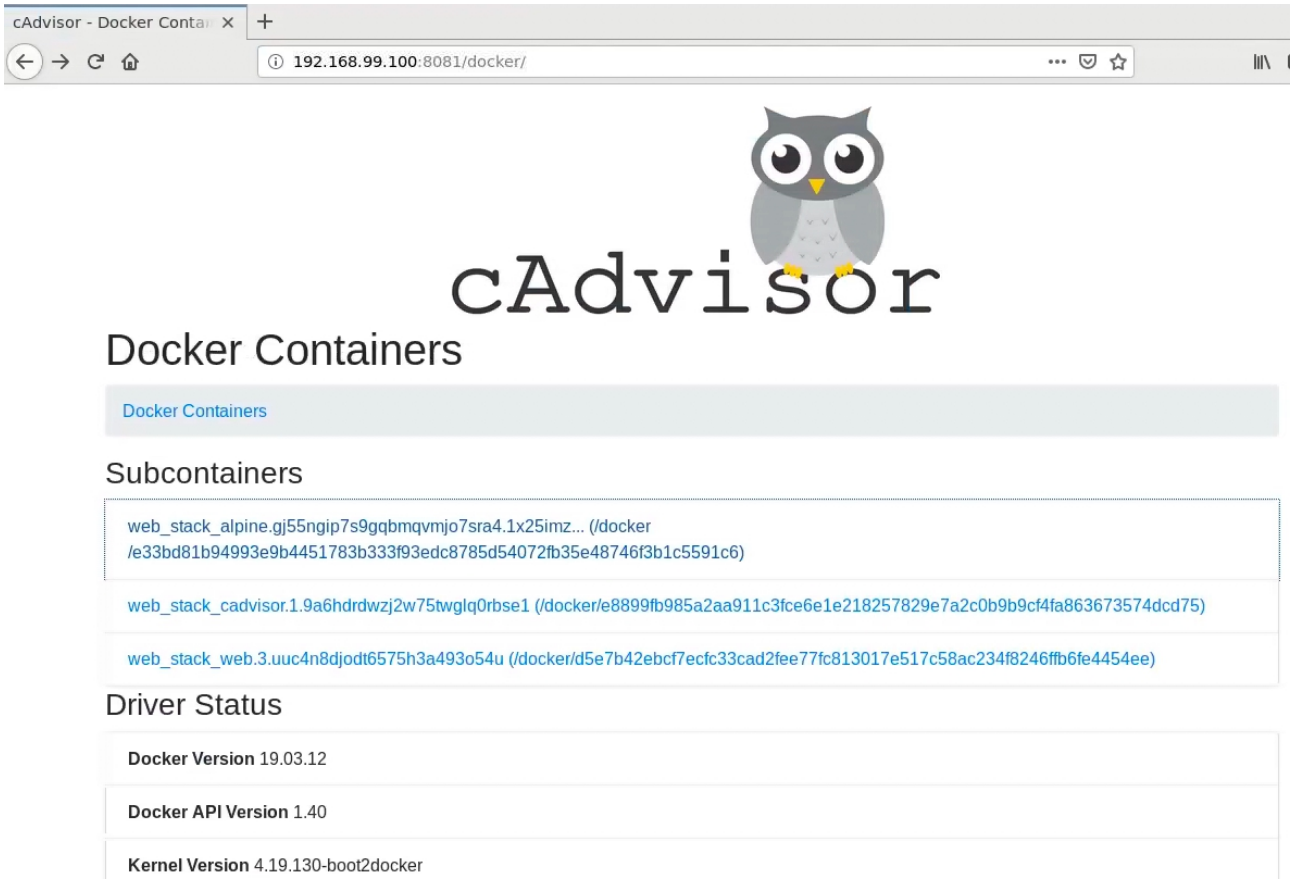
```
https://github.com/google/cadvisor
```

```
[centos@host11 ~]$ cat docker-compose.yml
version: "3"
services:
  alpine:
    image: alpine
    command: ["sleep","3600"]
    deploy:
      mode: global
  web:
    image: httpd
    deploy:
      replicas: 4
      restart_policy:
        condition: on-failure
      resources:
        limits:
          cpus: "0.1"
          memory: 50M
    ports:
      - "80:80"
  visualizer:
    image: dockersamples/visualizer:stable
    ports:
      - "8080:8080"
    volumes:
      - "/var/run/docker.sock:/var/run/docker.sock"
    deploy:
      placement:
        constraints: [node.role == manager]
  cadvisor:
    image: google/cadvisor
    ports:
      - "8081:8080"
    volumes:
      - /:/rootfs:ro
      - /var/run:/var/run:rw
      - /sys:/sys:ro
      - /var/lib/docker/:/var/lib/docker:ro
```

3/ Mettez à jour la pile de services :

```
[centos@host11 ~]$ docker stack deploy -c docker-compose.yml web_stack
Creating service web_stack_cadvisor
Updating service web_stack_alpine (id: el5djsjm6h7w1qrgqktvs4o6x)
Updating service web_stack_web (id: 02d6axyvuexbe4r288ayc7g31)
Creating service web_stack_visualizer

[centos@host11 ~]$ docker stack ls
NAME                SERVICES            ORCHESTRATOR
web_stack           4                   Swarm

[centos@host11 ~]$ docker stack ps web_stack
ID                  NAME                                    IMAGE
NODE                DESIRED STATE       CURRENT STATE            ERROR
PORTS
mc9j37qystur        web_stack_alpine.q13l8kwhs7curabef6hk2p2dx
alpine:latest                       node3               Running
Running 16 minutes ago
ro0jw1h2rtg5        web_stack_alpine.hffnp9jzz1k8d2mn7pjj5q9p2
alpine:latest                       node2               Running
Running 16 minutes ago
1x25imz3ik1m        web_stack_alpine.gj55ngip7s9gqbmqvmjo7sra4
alpine:latest                       node1               Running
Running 16 minutes ago
wgquqd9ui8xf        web_stack_visualizer.1
dockersamples/visualizer:stable     node3               Running
Running 16 seconds ago
9a6hdrdwzj2w        web_stack_cadvisor.1
google/cadvisor:latest              node1               Running
Running 30 seconds ago
ixt7f65f900z        web_stack_web.1
httpd:latest                        node2               Running
Running 16 minutes ago
i2hha5exlzzs        web_stack_web.2
httpd:latest                        node3               Running
Running 16 minutes ago
uuc4n8djodt6        web_stack_web.3
httpd:latest                        node1               Running
Running 16 minutes ago
3z3c3xkdx5it        web_stack_web.4
httpd:latest                        node2               Running
Running 16 minutes ago
```


Visualizer
192.168.99.100:8080
node1
manager
0.962G RAM
node2
manager
0.962G RAM
node3
manager
0.962G RAM
web_stack_web
image : httpd:latest@sha256:5ce7
tag : latest@sha256:5ce7c20e45b4
updated : 9/10 14:10
d5e7b42ebcf7ecfc33cad2fee77fc81
state : running
web_stack_cadvisor
image : cadvisor:latest@sha256:81
tag : latest@sha256:815386ebbe9a
updated : 9/10 14:26
e8899fb985a2aa911c3fce6e1e2182
state : running
web_stack_alpine
image : alpine:latest@sha256:1855
tag : latest@sha256:185518070891
cmd : sleep,3600
updated : 9/10 14:10
e33bd81b94993e9b4451783b333f9
state : running
web_stack_alpine
image : alpine:latest@sha256:1855
tag : latest@sha256:185518070891
cmd : sleep,3600
updated : 9/10 14:10
685aed43c6834c8c6ca842f150e71
state : running
web_stack_web
image : httpd:latest@sha256:5ce7
tag : latest@sha256:5ce7c20e45b4
updated : 9/10 14:10
adda7223945a738c03a56d71e7e9
state : running
web_stack_web
image : httpd:latest@sha256:5ce7
tag : latest@sha256:5ce7c20e45b4
updated : 9/10 14:10
7d830b00dc1a95dd74ba2a4c9356
state : running
web_stack_visualizer
image : visualizer:stable@sha256:b
tag : stable@sha256:bc680132f772
updated : 9/10 14:27
63face2d7292a5a4cca827742c141
state : running
web_stack_alpine
image : alpine:latest@sha256:1855
tag : latest@sha256:185518070891
cmd : sleep,3600
updated : 9/10 14:10
beb938ff70ed5a624b54753dab36d
state : running
web_stack_web
image : httpd:latest@sha256:5ce7
tag : latest@sha256:5ce7c20e45b4
updated : 9/10 14:10
3fe72374b85394b3f5b8814e86bb0
state : running


cAdvisor - Docker Conta
192.168.99.100:8081/docker/
cAdvisor
Docker Containers
Docker Containers
Subcontainers
web_stack_alpine.gj55ngip7s9gqbmqvmjo7sra4.1x25imz... (/docker
/e33bd81b94993e9b4451783b333f93edc8785d54072fb35e48746f3b1c5591c6)
web_stack_cadvisor.1.9a6hdrdwzj2w75twglq0rbse1 (/docker/e8899fb985a2aa911c3fce6e1e218257829e7a2c0b9b9cf4fa863673574dcd75)
web_stack_web.3.uuc4n8djodt6575h3a493o54u (/docker/d5e7b42ebcf7ecfc33cad2fee77fc813017e517c58ac234f8246ffb6fe4454ee)
Driver Status
Docker Version 19.03.12
Docker API Version 1.40
Kernel Version 4.19.130-boot2docker