PROJET DOCKER

```
Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/focal64"

  # Manager Node
  config.vm.define "manager" do |manager|
    manager.vm.hostname = "manager"
    manager.vm.network "private_network", type: "dhcp"
    manager.vm.provider "virtualbox" do |vb|
      vb.memory = "1024"
      vb.cpus = 2
    end
  end

  # Worker Nodes
  (1..2).each do |i|
    config.vm.define "worker#{i}" do |worker|
      worker.vm.hostname = "worker#{i}"
      worker.vm.network "private_network", type: "dhcp"
      worker.vm.provider "virtualbox" do |vb|
        vb.memory = "1024"
        vb.cpus = 2
      end
    end
  end
end
```

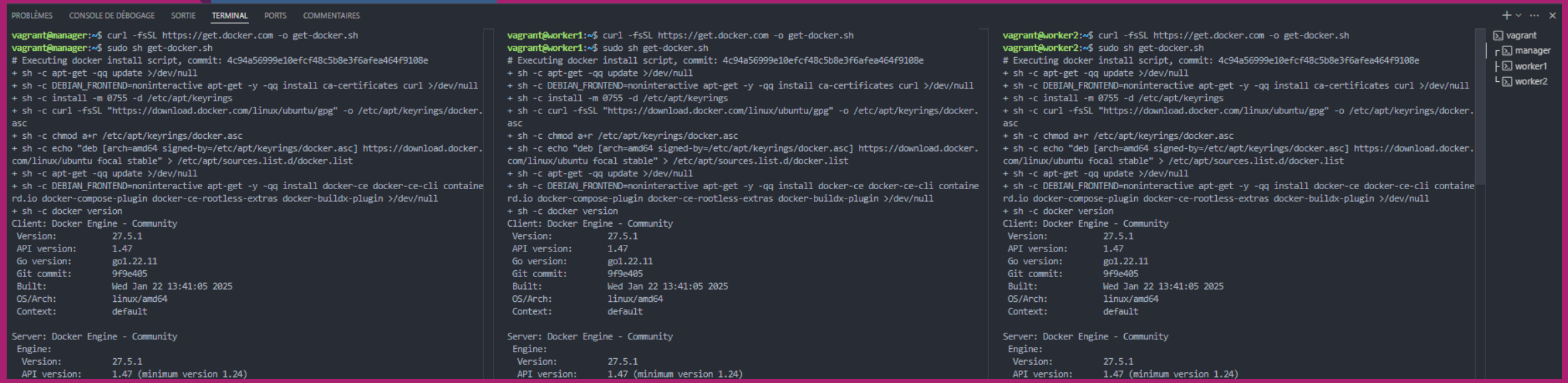
CONFIGURATION DU VAGRANTFILE

- Création des VMs
- Lancer la création et le démarrage des VMs avec « vagrant up ».

INSTALLER DOCKER SUR TOUTES LES VM

```
curl -fsSL https://get.docker.com -o get-docker.sh
sudo sh get-docker.sh
```

```
docker –version        // Vérifier que docker est bien installé
```



INITIALISER LE CLUSTER SWARM SUR LE MANAGER

```
vagrant@manager:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 02:45:be:c8:08:41 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 metric 100 brd 10.0.2.255 scope global dynamic enp0s3
       valid_lft 84846sec preferred_lft 84846sec
    inet6 fd00::45:beff:fec8:841/64 scope global dynamic mngtmpaddr noprefixroute
       valid_lft 86342sec preferred_lft 14342sec
    inet6 fe80::45:beff:fec8:841/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:b9:2b:d2 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.3/24 metric 100 brd 192.168.56.255 scope global dynamic enp0s8
       valid_lft 546sec preferred_lft 546sec
    inet6 fe80::a00:27ff:feb9:2bd2/64 scope link
       valid_lft forever preferred_lft forever
4: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:d6:b1:34:13 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
```

- Récupération de l'ip du manager
- Initialiser Docker Swarm avec la bonne IP :

```
docker swarm init --advertise-addr 192.168.56.3
```

```
vagrant@manager:~$ docker swarm init --advertise-addr 192.168.56.3
Swarm initialized: current node (fb8rabpp8ndg97cgdug4ji0nv) is now a manager.

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

    docker swarm join --token SWMTKN-1-4omfh22wo1rn2x0yj0pxxfke1dv3a4ymzoovrmjmrt7ixickfu-bwn7g2bvji0rp7uzfvzq29pql 192.168.56.3:2377

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

- Récupération du token afin de pouvoir ajouter les workers :

```
docker swarm join --token SWMTKN-1-4omfh22wo1rn2x0yj0pxxfke1dv3a4ymzoovrmjmrt7ixickfu-bwn7g2bvji0rp7uzfvzq29pql 192.168.56.3:2377
```

AJOUTER LES NŒUDS WORKER AU CLUSTER SWARM

```
vagrant@worker1:~$ docker swarm join --token SWMTKN-1-4omfh22wo1rn2x0yj0pxxfke1dv3a4ymzoovrmjmrt7ixickfu-bwn
7g2bvji0rp7uzfvzq29pql 192.168.56.3:2377
This node joined a swarm as a worker.
vagrant@worker1:~$
```

```
vagrant@worker2:~$ docker swarm join --token SWMTKN-1-4omfh22wo1rn2x0yj0pxxfke1dv3a4ymzoovrmjmrt7ixickfu-bwn
7g2bvji0rp7uzfvzq29pql 192.168.56.3:2377
This node joined a swarm as a worker.
vagrant@worker2:~$
```

VÉRIFIER QUE LE CLUSTER EST BIEN FORMÉ

```
docker node ls
```

```
vagrant@manager:~$ docker node ls
ID                            HOSTNAME   STATUS    AVAILABILITY   MANAGER STATUS   ENGINE VERSION
fb8rabpp8ndg97cgdug4ji0nv *   manager    Ready     Active         Leader           27.5.1
olr1nava8jkokzzebckgy9hv6     worker1    Ready     Active                          27.5.1
su7pl2wqrtlluilhihdjlhncz     worker2    Ready     Active                          27.5.1
vagrant@manager:~$
```

```
docker info | grep Swarm
```

```
vagrant@manager:~$ docker info | grep Swarm
WARNING: No swap limit support
 Swarm: active
```

```
explo@Yoda-PC MINGW64 /d/Projets/swarm-cluster/voting-app-docker (main)
$ docker image ls
REPOSITORY   TAG       IMAGE ID       CREATED              SIZE
result-app   latest    bd08ac796cc1   About a minute ago   202MB
worker-app   latest    dc61f643a821   2 minutes ago        321MB
vote-app     latest    a7dfdc4c52f8   4 minutes ago        110MB
```

CONSTRUIRE MANUELLEMENT LES IMAGES DOCKER

Sur la machine principale, se placer dans le dossier contenant l'application

```
cd voting-app-docker/
```

Puis exécuter les commandes suivantes :

```
docker build -t vote-app ./vote
docker build -t worker-app ./worker
docker build -t result-app ./result
```

```
explo@Yoda-PC MINGW64 /d/Projets/swarm-cluster/voting-app-docker (main)
$ docker build -t vote-app ./vote
[+] Building 10.7s (11/11) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 270B
 => [internal] load metadata for docker.io/library/python:3.11-alpine
 => [auth] library/python:pull token for registry-1.docker.io
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [1/5] FROM docker.io/library/python:3.11-alpine@sha256:9af3561825050da182afc74b106388af570b99c500a69c8216263aa245a2001b
 => => resolve docker.io/library/python:3.11-alpine@sha256:9af3561825050da182afc74b106388af570b99c500a69c8216263aa245a2001b
 => => sha256:5c3947958a836ea9bf6f6c0c4ac022493784474649e0500f6d6dad4a3f9be7be 249B / 249B
 => => sha256:69d04f35a207537b61d6fe9bbd170c391d10ed493d1bf40d4850c170fe4894e9 16.17MB / 16.17MB
 => => sha256:dfb81f221332c555fd5a9cd799ece5510b3a131a383a831d5b610f9fe5a040bb 458.84kB / 458.84kB
 => => sha256:1f3e46996e2966e4faa5846e56e76e3748b7315e2ded61476c24403d592134f0 3.64MB / 3.64MB
 => => extracting sha256:1f3e46996e2966e4faa5846e56e76e3748b7315e2ded61476c24403d592134f0
 => => extracting sha256:dfb81f221332c555fd5a9cd799ece5510b3a131a383a831d5b610f9fe5a040bb
 => => extracting sha256:69d04f35a207537b61d6fe9bbd170c391d10ed493d1bf40d4850c170fe4894e9
 => => extracting sha256:5c3947958a836ea9bf6f6c0c4ac022493784474649e0500f6d6dad4a3f9be7be
 => [internal] load build context
 => => transferring context: 6.24kB
 => [2/5] WORKDIR /app
 => [3/5] COPY requirements.txt ./
 => [4/5] RUN pip install --no-cache-dir -r requirements.txt
 => [5/5] COPY . .
 => exporting to image
 => => exporting layers
 => => exporting manifest sha256:d17d7d427bea76b5a1e18fbe90aa4acfe05142765fc8ea3510dd086362162218
 => => exporting config sha256:386ffa67044497 1fa20244c10a218b36ebb81e8d178bcec4e0017b77074368c
 => => exporting attestation manifest sha256:2ba03823369c94fb7abe649e386054fd632f98ad066364234140e316d5e570b1
 => => exporting manifest list sha256:a7dfdc4c52f82dd092437bb068aa2ece0bc7f8da9ff2e0d49381ffa76132942b
 => => naming to docker.io/library/vote-app:latest
 => => unpacking to docker.io/library/vote-app:latest
```

```
explo@Yoda-PC MINGW64 /d/Projets/swarm-cluster/voting-app-docker (main)
$ docker build -t worker-app ./worker
[+] Building 75.2s (15/15) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 843B
 => [internal] load metadata for mcr.microsoft.com/dotnet/sdk:7.0
 => [internal] load metadata for mcr.microsoft.com/dotnet/aspnet:7.0
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [runtime 1/3] FROM mcr.microsoft.com/dotnet/aspnet:7.0@sha256:c7d9ee6cd01afe9aa80642e577c7cec9f5d87f88e5d70bd36fd61072079bc55b
 => => resolve mcr.microsoft.com/dotnet/aspnet:7.0@sha256:c7d9ee6cd01afe9aa80642e577c7cec9f5d87f88e5d70bd36fd61072079bc55b
 => => sha256:f194078e85f8008c084163778aaf266434f7182e0d4f783646f41b388c88a13a 10.12MB / 10.12MB
 => => sha256:f1b39e168c1c776458e172f157167607b9fd3cc550af8e6ff0a7dd363c1e64ea 153B / 153B
 => => sha256:534ba947de6ac79fd6168f4a93847954b23bab2782700bbfff7f31e61a03e8d4 32.46MB / 32.46MB
 => => sha256:82bb7a80de578404d92b5ae5e67f3de90eab30027694d2609be35ad25b09e3bc 14.97MB / 14.97MB
 => => sha256:728328ac3bde9b85225b1f0d60f5c149f5635a191f5d8eaeeb00e095d36ef9fd 31.43MB / 31.43MB
 => => extracting sha256:728328ac3bde9b85225b1f0d60f5c149f5635a191f5d8eaeeb00e095d36ef9fd
 => => extracting sha256:82bb7a80de578404d92b5ae5e67f3de90eab30027694d2609be35ad25b09e3bc
 => => extracting sha256:534ba947de6ac79fd6168f4a93847954b23bab2782700bbfff7f31e61a03e8d4
 => => extracting sha256:f1b39e168c1c776458e172f157167607b9fd3cc550af8e6ff0a7dd363c1e64ea
 => => extracting sha256:f194078e85f8008c084163778aaf266434f7182e0d4f783646f41b388c88a13a
 => [build 1/6] FROM mcr.microsoft.com/dotnet/sdk:7.0@sha256:d32bd65cf5843f413e81f5d917057c82da99737cb1637e905a1a4bc2e7ec6c8d
 => => resolve mcr.microsoft.com/dotnet/sdk:7.0@sha256:d32bd65cf5843f413e81f5d917057c82da99737cb1637e905a1a4bc2e7ec6c8d
 => => sha256:c549dbd140e5a09bcc7e4b1da89ed4b96a0340a730303fbe2e903d46cf06456e 13.99MB / 13.99MB
 => => sha256:5e263829ce76bc29ff631384bd5e93356d28625461ca507fadc46ec4ac549fb4 180.57MB / 180.57MB
 => => sha256:22689bb63f95af71c8c0079742ff0afa570fa6cfc30ecfa787397689082 9a24d 25.37MB / 25.37MB
 => => extracting sha256:22689bb63f95af71c8c0079742ff0afa570fa6cfc30ecfa787397689082 9a24d
 => => extracting sha256:5e263829ce76bc29ff631384bd5e93356d28625461ca507fadc46ec4ac549fb4
 => => extracting sha256:c549dbd140e5a09bcc7e4b1da89ed4b96a0340a730303fbe2e903d46cf06456e
 => [internal] load build context
 => => transferring context: 9.03kB
 => [runtime 2/3] WORKDIR /app
 => [build 2/6] WORKDIR /app
 => [build 3/6] COPY *.csproj ./
 => [build 4/6] RUN dotnet restore
 => [build 5/6] COPY . .
 => [build 6/6] RUN dotnet publish -c Release --self-contained false --no-restore -o /app/out
 => [runtime 3/3] COPY --from=build /app/out ./
 => exporting to image
 => => exporting layers
 => => exporting manifest sha256:feeb9c679117b8f27b6ffe99c26e58896539b13198cbe719a92126aac9b496df
 => => exporting config sha256:bd208f49833f1e93bf5c5ba637fd48e2790f998610c54b759711ead48bea8346
 => => exporting attestation manifest sha256:ce3606ed62ea258786f41b921a9ef31aaa453c267b937e84520e70c01b549c69
 => => exporting manifest list sha256:dc61f643a821724ed16a7581237fecdc8bdcdd4a1d219da17fcdcc5caf2c9450
 => => naming to docker.io/library/worker-app:latest
 => => unpacking to docker.io/library/worker-app:latest
```

```
explo@Yoda-PC MINGW64 /d/Projets/swarm-cluster/voting-app-docker (main)
$ docker build -t result-app ./result
[+] Building 17.0s (11/11) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 169B
 => [internal] load metadata for docker.io/library/node:18-alpine
 => [auth] library/node:pull token for registry-1.docker.io
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [1/5] FROM docker.io/library/node:18-alpine@sha256:974afb6cbc0314dc6502b14243b8a39fbb2d04d975e9059dd066be3e274fbb25
 => => resolve docker.io/library/node:18-alpine@sha256:974afb6cbc0314dc6502b14243b8a39fbb2d04d975e9059dd066be3e274fbb25
 => => sha256:6504e29600c8d5213b52cda800370abb3d12639802d06b46b6fce368990ca771 444B / 444B
 => => sha256:5650d6de56fd0bb419872b876ac1df28f577b39573c3b72fb0d15bf426d01bc1 1.26MB / 1.26MB
 => => sha256:37892ffbfcaa871a10f813803949d18c3015a482051d51b7e0da02525e63167c 40.01MB / 40.01MB
 => => extracting sha256:37892ffbfcaa871a10f813803949d18c3015a482051d51b7e0da02525e63167c
 => => extracting sha256:5650d6de56fd0bb419872b876ac1df28f577b39573c3b72fb0d15bf426d01bc1
 => => extracting sha256:6504e29600c8d5213b52cda800370abb3d12639802d06b46b6fce368990ca771
 => [internal] load build context
 => => transferring context: 306.33kB
 => [2/5] WORKDIR /app
 => [3/5] COPY package.json package-lock.json ./
 => [4/5] RUN npm install
 => [5/5] COPY . .
 => exporting to image
 => => exporting layers
 => => exporting manifest sha256:68faa4271704c09a82828eafc1b15dfa43e1241f1f9770234a253557ea14e35f
 => => exporting config sha256:2aa2ccb5508273e1ae5220bd08cdd7e93759ca9218c9d5a472b61c42e0b76ba6
 => => exporting attestation manifest sha256:05ef27132871012f6097d62304499e0202e76694d41d3d5d1740efc8fdb6bbfc
 => => exporting manifest list sha256:bd08ac796cc1b0af328be2fd7b5a8834654a13cd7959b8f8e9eb74262440f5dd
 => => naming to docker.io/library/result-app:latest
 => => unpacking to docker.io/library/result-app:latest
```

PUBLICATION DES IMAGES DOCKER SUR DOCKER HUB

Ajouter des tags avec notre compte Docker Hub :

```
docker tag vote-app:latest iamyoda/vote-app:latest
docker tag worker-app:latest iamyoda/worker-app:latest
docker tag result-app:latest iamyoda/result-app:latest
```

```
explo@Yoda-PC MINGW64 /d/Projets/swarm-cluster/voting-app-docker (main)
$ docker tag vote-app:latest iamyoda/vote-app:latest

explo@Yoda-PC MINGW64 /d/Projets/swarm-cluster/voting-app-docker (main)
$ docker tag worker-app:latest iamyoda/worker-app:latest

explo@Yoda-PC MINGW64 /d/Projets/swarm-cluster/voting-app-docker (main)
$ docker tag result-app:latest iamyoda/result-app:latest
```

Vérifier que les modifications ont été prise en compte :

```
explo@Yoda-PC MINGW64 /d/Projets/swarm-cluster/voting-app-docker (main)
$ docker images
REPOSITORY           TAG       IMAGE ID       CREATED          SIZE
iamyoda/result-app   latest    bd08ac796cc1   10 minutes ago   202MB
result-app           latest    bd08ac796cc1   10 minutes ago   202MB
iamyoda/worker-app   latest    dc61f643a821   11 minutes ago   321MB
worker-app           latest    dc61f643a821   11 minutes ago   321MB
iamyoda/vote-app     latest    a7dfdc4c52f8   13 minutes ago   110MB
vote-app             latest    a7dfdc4c52f8   13 minutes ago   110MB
```

PUBLICATION DES IMAGES DOCKER SUR DOCKER HUB

Envoi des images sur Docker Hub :

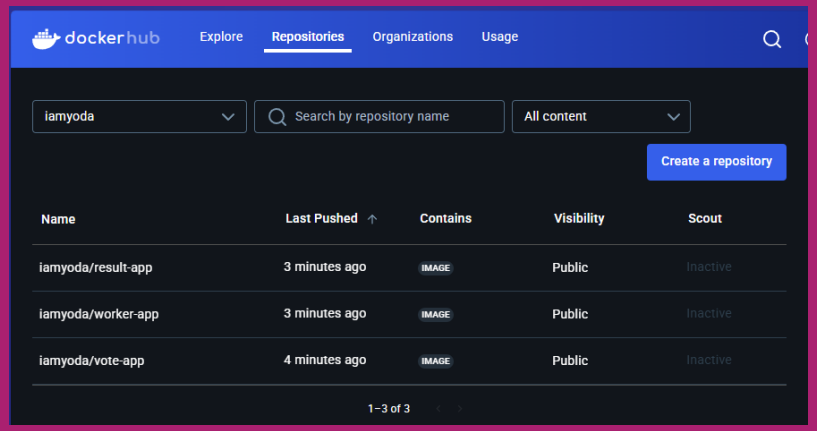
```
docker push iamyoda/vote-app:latest
docker push iamyoda/worker-app:latest
docker push iamyoda/result-app:latest
```

```
explo@Yoda-PC MINGW64 /d/Projets/swarm-cluster/voting-app-docker (main)
$ docker push iamyoda/vote-app:latest
The push refers to repository [docker.io/iamyoda/vote-app]
db4e0ffc24fe: Pushed
dfb81f221332: Pushed
bb0eb49bfbc5: Pushed
d868ddc68e0f: Pushed
69d04f35a207: Pushed
09a7e69067e4: Pushed
e2142a90fef7: Pushed
1f3e46996e29: Pushed
5c3947958a83: Pushed
latest: digest: sha256:a7dfdc4c52f82dd092437bb068aa2ece0bc7f8da9ff2e0d49381ffa76132942b size: 856
```

```
explo@Yoda-PC MINGW64 /d/Projets/swarm-cluster/voting-app-docker (main)
$ docker push iamyoda/worker-app:latest
The push refers to repository [docker.io/iamyoda/worker-app]
f1b39e168c1c: Pushed
82bb7a80de57: Pushed
ee3d2ac372f2: Pushed
998d15277a8b: Pushed
534ba947de6a: Pushed
f194078e85f8: Pushed
728328ac3bde: Pushed
3887bebcafaa: Pushed
latest: digest: sha256:dc61f643a821724ed16a7581237fecdc8bdcdd4a1d219da17fcdcc5caf2c9450 size: 856
```

```
explo@Yoda-PC MINGW64 /d/Projets/swarm-cluster/voting-app-docker (main)
$ docker push iamyoda/result-app:latest
The push refers to repository [docker.io/iamyoda/result-app]
b4c7f3381176: Pushed
5650d6de56fd: Pushed
1f3e46996e29: Mounted from iamyoda/vote-app
23cbe0974ace: Pushed
b541405954ff: Pushed
6504e29600c8: Pushed
b42041aef83f: Pushed
8db6db3c8432: Pushed
37892ffbfcaa: Pushed
latest: digest: sha256:bd08ac796cc1b0af328be2fd7b5a8834654a13cd7959b8f8e9eb74262440f5dd size: 856
```

Vérifier que les images ont bien été envoyer sur votre repo :



CRÉER LE FICHIER DOCKER-COMPOSE-SWARM

```yaml
version: '3.8'

Run All Services
services:
  Run Service
  vote:
    image: iamyoda/vote-app:latest
    ports:
      - "8080:8080"
    env_file:
      - .env
    depends_on:
      - redis
    networks:
      - frontend
      - backend
    deploy:
      replicas: 2
      restart_policy:
        condition: on-failure

  Run Service
  worker:
    image: iamyoda/worker-app:latest
    env_file:
      - .env
    depends_on:
      - redis
      - db
    networks:
      - backend
    deploy:
      replicas: 1
      restart_policy:
        condition: on-failure
```

```yaml
  Run Service
  result:
    image: iamyoda/result-app:latest
    ports:
      - "8888:8888"
    env_file:
      - .env
    depends_on:
      - db
      - redis
      - vote
    networks:
      - frontend
      - backend
    deploy:
      replicas: 2
      restart_policy:
        condition: on-failure

  Run Service
  db:
    image: postgres:latest
    env_file:
      - .env
    volumes:
      - postgres_data:/var/lib/postgresql/data
      - ./db/init.sql:/docker-entrypoint-initdb.d/init.sql
    networks:
      - backend
    deploy:
      placement:
        constraints:
          - node.role == manager
```

```yaml
  Run Service
  redis:
    image: redis:latest
    env_file:
      - .env
    command: sh -c "redis-server --requirepass $$REDIS_PASSWORD"
    volumes:
      - redis_data:/data
      - ./db/redis.conf:/usr/local/etc/redis/redis.conf
    networks:
      - backend
    deploy:
      replicas: 1

volumes:
  postgres_data:
  redis_data:

networks:
  frontend:
  backend:
```

COPIER LE PROJET VERS LA VM MANAGER

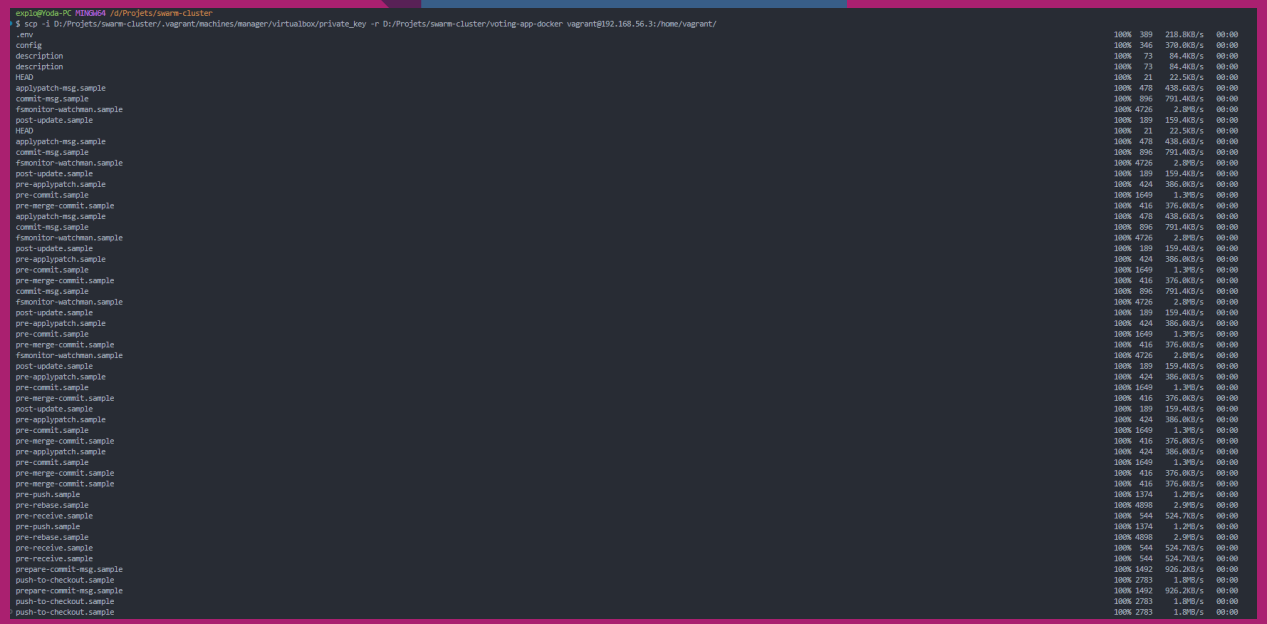
Récupération du chemin de la clé privée du manager pour copier le projet sur celui-ci : `vagrant ssh-config`

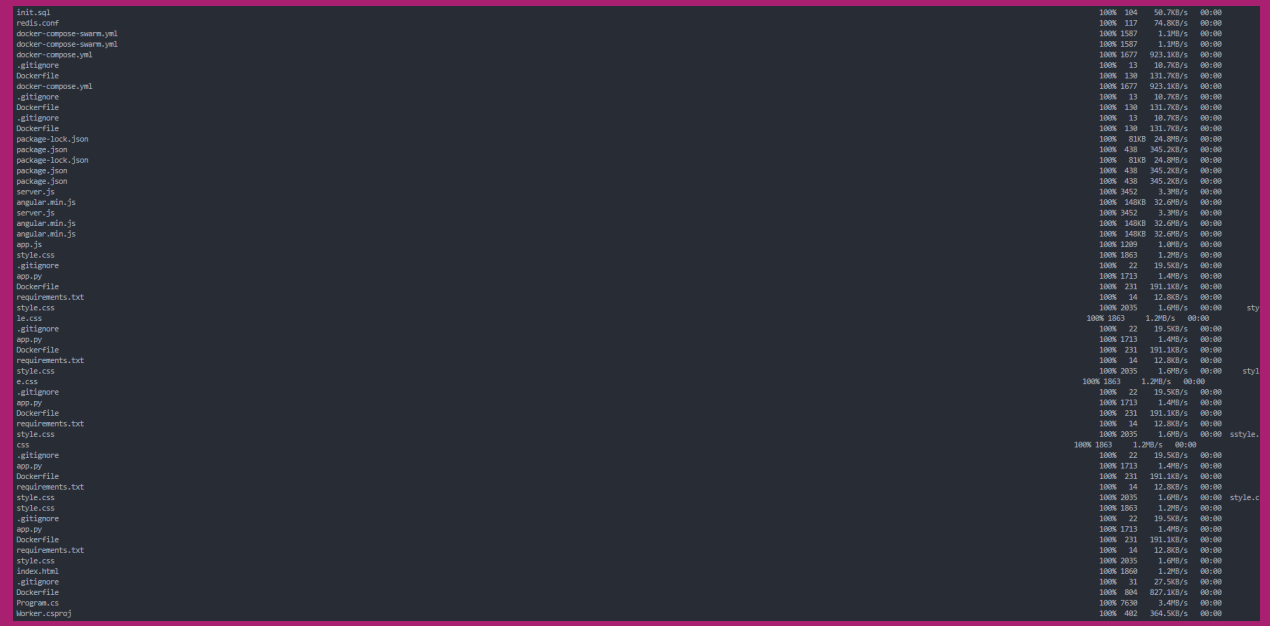
```
explo@Yoda-PC MINGW64 /d/Projets/swarm-cluster
$ vagrant ssh-config
Host manager
  HostName 127.0.0.1
  User vagrant
  Port 2222
  UserKnownHostsFile /dev/null
  StrictHostKeyChecking no
  PasswordAuthentication no
  IdentityFile D:/Projets/swarm-cluster/.vagrant/machines/manager/virtualbox/private_key
  IdentitiesOnly yes
  LogLevel FATAL
  PubkeyAcceptedKeyTypes +ssh-rsa
  HostKeyAlgorithms +ssh-rsa

Host worker1
  HostName 127.0.0.1
  User vagrant
  Port 2200
  UserKnownHostsFile /dev/null
  StrictHostKeyChecking no
  PasswordAuthentication no
  IdentityFile D:/Projets/swarm-cluster/.vagrant/machines/worker1/virtualbox/private_key
  IdentitiesOnly yes
  LogLevel FATAL
  PubkeyAcceptedKeyTypes +ssh-rsa
  HostKeyAlgorithms +ssh-rsa

Host worker2
  HostName 127.0.0.1
  User vagrant
  Port 2201
  UserKnownHostsFile /dev/null
  StrictHostKeyChecking no
  PasswordAuthentication no
  IdentityFile D:/Projets/swarm-cluster/.vagrant/machines/worker2/virtualbox/private_key
  IdentitiesOnly yes
  LogLevel FATAL
  PubkeyAcceptedKeyTypes +ssh-rsa
  HostKeyAlgorithms +ssh-rsa
```

COPIER LE PROJET VERS LA VM MANAGER

Commande pour copier le projet sur la VM

```
scp -i D:/Projets/swarm-cluster/.vagrant/machines/manager/virtualbox/private_key -r D:/Projets/swarm-cluster/voting-app-docker vagrant@192.168.56.3:/home/vagrant/
```





VÉRIFIER QUE LES FICHIERS SONT BIEN TRANSFÉRÉS

ls -la /home/vagrant/voting-app-docker

```
vagrant@manager:~$ ls -la /home/vagrant/voting-app-docker
total 44
drwxr-xr-x 7 vagrant vagrant 4096 Feb  6 23:35 .
drwxr-xr-x 5 vagrant vagrant 4096 Feb  6 23:35 ..
-rw-r--r-- 1 vagrant vagrant  389 Feb  6 23:35 .env
drwxr-xr-x 7 vagrant vagrant 4096 Feb  6 23:35 .git
-rw-r--r-- 1 vagrant vagrant  124 Feb  6 23:35 .gitignore
drwxr-xr-x 2 vagrant vagrant 4096 Feb  6 23:35 db
-rw-r--r-- 1 vagrant vagrant 1587 Feb  6 23:35 docker-compose-swarm.yml
-rw-r--r-- 1 vagrant vagrant 1677 Feb  6 23:35 docker-compose.yml
drwxr-xr-x 3 vagrant vagrant 4096 Feb  6 23:35 result
drwxr-xr-x 4 vagrant vagrant 4096 Feb  6 23:35 vote
drwxr-xr-x 2 vagrant vagrant 4096 Feb  6 23:35 worker
```

DÉPLOYER L'APPLICATION AVEC DOCKER SWARM

Sur le manager, se placer dans le dossier contenant l'application

```
cd /home/vagrant/voting-app-docker
```

Puis exécuter la commande suivante :

```
docker stack deploy -c docker-compose-swarm.yml votingapp
```

Cette commande permet de :

- Créer les services Docker
- Répartir les conteneurs sur les 3 nœuds
- Configurer le réseau

```
vagrant@manager:~/voting-app-docker$ docker stack deploy -c docker-compose-swarm.yml votingapp
Since --detach=false was not specified, tasks will be created in the background.
In a future release, --detach=false will become the default.
Updating service votingapp_redis (id: k7bumx16yaj66djknjscvh4ip)
Updating service votingapp_vote (id: lu1mork3exbsy8wyabnk0cbja)
Updating service votingapp_worker (id: qtgdmpwzbq31is1pqykjj5c12)
Updating service votingapp_result (id: yvhjt32zaux6j3sa1hiqvv3hn)
Updating service votingapp_db (id: q3t9544aytl2ahvpnc0hlh6jt)
```

Vérifier si les services fonctionnent bien :

```
vagrant@manager:~/voting-app-docker$ docker service ls
ID             NAME               MODE         REPLICAS   IMAGE                          PORTS
q3t9544aytl2   votingapp_db       replicated   1/1        postgres:latest
k7bumx16yaj6   votingapp_redis    replicated   1/1        redis:latest
yvhjt32zaux6   votingapp_result   replicated   2/2        iamyoda/result-app:latest      *:8888->8888/tcp
lu1mork3exbs   votingapp_vote     replicated   2/2        iamyoda/vote-app:latest        *:8080->8080/tcp
qtgdmpwzbq31   votingapp_worker   replicated   1/1        iamyoda/worker-app:latest
```

VÉRIFIER L'ÉTAT DES CONTENEURS SUR CHAQUE NŒUD

```
vagrant@manager:~/voting-app-docker$ docker ps
CONTAINER ID   IMAGE                          COMMAND                  CREATED         STATUS         PORTS      NAMES
619d2ac308da   redis:latest                   "docker-entrypoint.s…"   3 minutes ago   Up 3 minutes   6379/tcp   votingapp_redis.1.5snfzvlh6buljknocor1i84gu
caffc09ecea3   postgres:latest                "docker-entrypoint.s…"   3 minutes ago   Up 3 minutes   5432/tcp   votingapp_db.1.444iv0ilwnfs3xd1ivjkgw9dw
562332ef7873   iamyoda/worker-app:latest      "dotnet Worker.dll"      3 minutes ago   Up 3 minutes   5000/tcp   votingapp_worker.1.tep2oscemg1yew8ybgjdijjui
```

```
vagrant@worker1:~$ docker ps
CONTAINER ID   IMAGE                          COMMAND                  CREATED         STATUS         PORTS      NAMES
55d179052b76   iamyoda/result-app:latest      "docker-entrypoint.s…"   4 minutes ago   Up 4 minutes              votingapp_result.1.jzv91r1fjmueca8fd39ikgtst
1d83bdc54145   iamyoda/vote-app:latest        "sh -c 'flask run --…"   4 minutes ago   Up 4 minutes   8080/tcp   votingapp_vote.2.6rvfbbno3lh2j1cfpfxl6gwc7
```

```
vagrant@worker2:~$ docker ps
CONTAINER ID   IMAGE                          COMMAND                  CREATED         STATUS         PORTS      NAMES
79d5bc8446fc   iamyoda/vote-app:latest        "sh -c 'flask run --…"   5 minutes ago   Up 4 minutes   8080/tcp   votingapp_vote.1.19v9xx9849ti3c8qdkte05vyj
ed91ee64cd65   iamyoda/result-app:latest      "docker-entrypoint.s…"   5 minutes ago   Up 5 minutes              votingapp_result.2.i81k0g90exkqxqgh4h1u90pgj
```

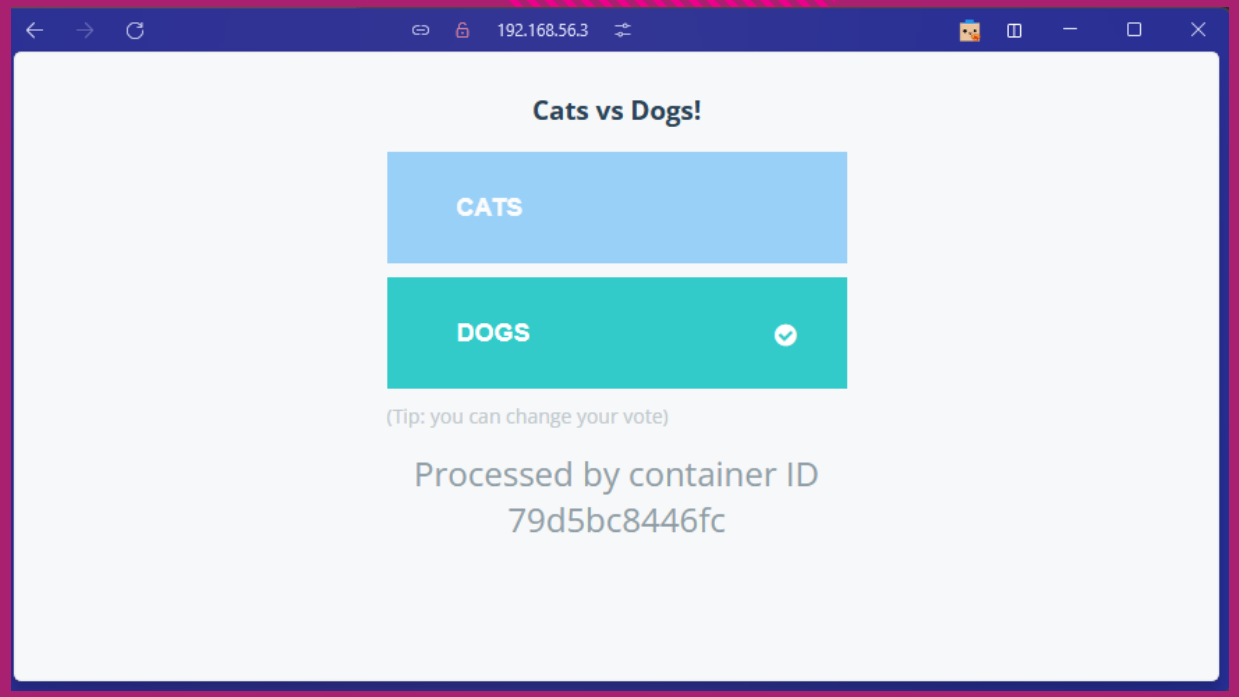
TESTER LES APPLICATIONS

Récupérer l'IP du manager :

```
ip a | grep "inet "
```

```
vagrant@manager:~/voting-app-docker$ ip a | grep "inet "
    inet 127.0.0.1/8 scope host lo
    inet 10.0.2.15/24 metric 100 brd 10.0.2.255 scope global dynamic enp0s3
    inet 192.168.56.3/24 metric 100 brd 192.168.56.255 scope global dynamic enp0s8
    inet 172.18.0.1/16 brd 172.18.255.255 scope global docker_gwbridge
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
```

http://192.168.56.3:8080/



http://192.168.56.3:8888/