ESIEA

Project report

VIRTUALIZATION, CONTAINERIZATION and CLOUD COMPUTING

Created By : Said NAOUI

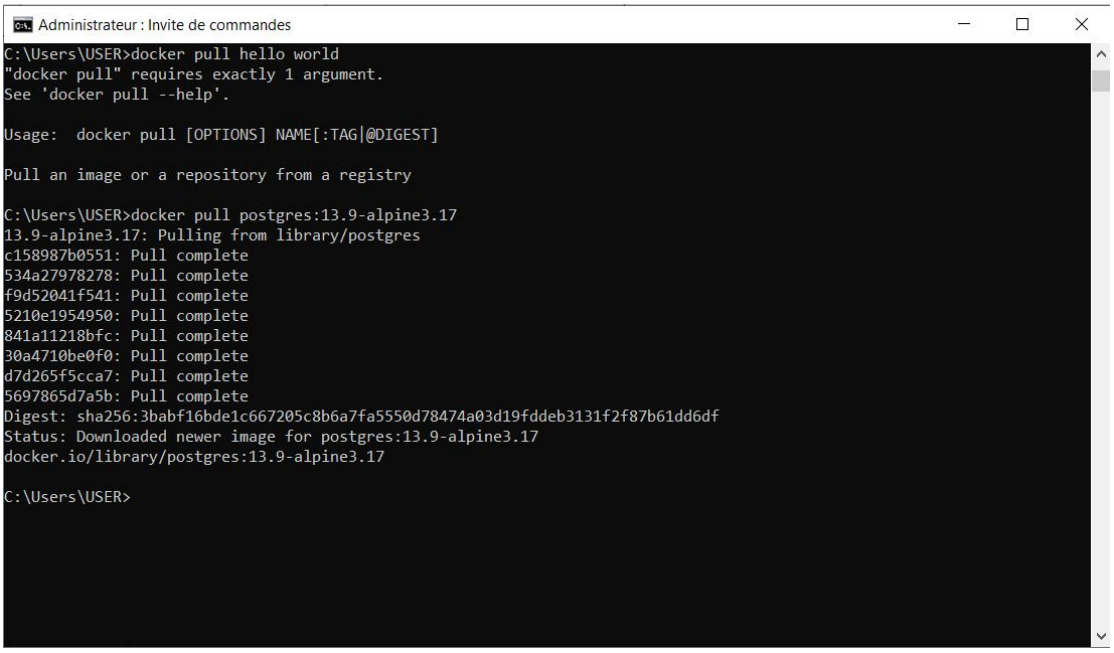
22/12/2022

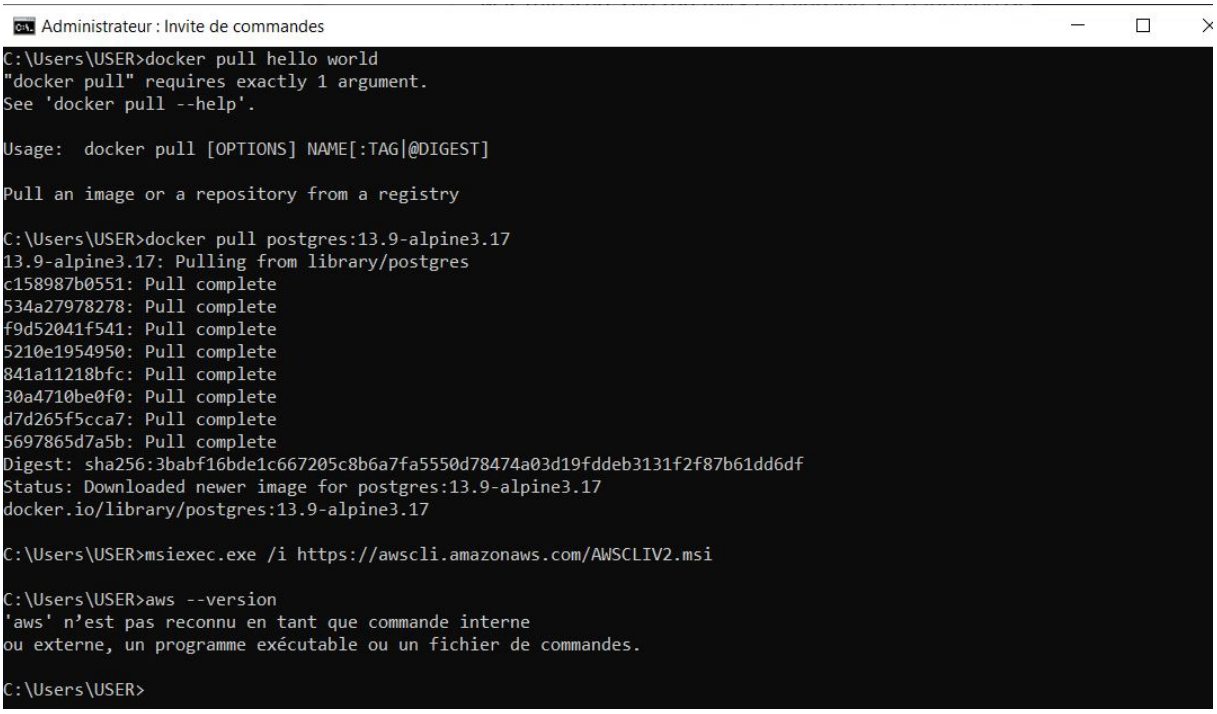
Party 1:

1- Make sure to have docker desktop running on your host machine :

```
USER@DESKTOP-B2Q5BQ7 MINGW64 ~/OneDrive - um5.ac.ma/Desktop/esiea-4th-vcc-projec
t-main/esiea-4th-vcc-project-main
$ bash set_env.sh

USER@DESKTOP-B2Q5BQ7 MINGW64 ~/OneDrive - um5.ac.ma/Desktop/esiea-4th-vcc-projec
t-main/esiea-4th-vcc-project-main
$ bash set_env.sh

USER@DESKTOP-B2Q5BQ7 MINGW64 ~/OneDrive - um5.ac.ma/Desktop/esiea-4th-vcc-projec
t-main/esiea-4th-vcc-project-main
$ dir
CODEOWNERS                screenshots       udagram-reverseproxy
LICENSE.txt               set_env.sh        udagram.png
Project_Instructions      udagram-api-feed  udagram_architecture.png
README.md                 udagram-api-user
docker-compose-build.yaml  udagram-frontend

USER@DESKTOP-B2Q5BQ7 MINGW64 ~/OneDrive - um5.ac.ma/Desktop/esiea-4th-vcc-projec
t-main/esiea-4th-vcc-project-main
$ cat set_env.sh


# This file is used for convenience of local development.
# DO NOT STORE YOUR CREDENTIALS INTO GIT
export POSTGRES_USERNAME=admin
export POSTGRES_PASSWORD=admin1234
export POSTGRES_HOST=udagramd-db
export POSTGRES_DB=udagramdb
export AWS_BUCKET=udagram-esiea-8765
export AWS_REGION=us-east-1
export AWS_PROFILE=default
export JWT_SECRET=testing
export URL=http://localhost:8100
USER@DESKTOP-B2Q5BQ7 MINGW64 ~/OneDrive - um5.ac.ma/Desktop/esiea-4th-vcc-projec
t-main/esiea-4th-vcc-project-main
$ docker

Usage:  docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Options:
      --config string      Location of client config files (default
                           "C:\\Users\\USER\\.docker")
  -c, --context string     Name of the context to use to connect to the
                           daemon (overrides DOCKER_HOST env var and
                           default context set with "docker context use")
  -D, --debug              Enable debug mode
  -H, --host list          Daemon socket(s) to connect to
  -l, --log-level string   Set the logging level
                           ("debug"|"info"|"warn"|"error"|"fatal")
                           (default "info")
      --tls                Use TLS; implied by --tlsverify
      --tlscacert string   Trust certs signed only by this CA (default
                           "C:\\Users\\USER\\.docker\\ca.pem")
      --tlscert string     Path to TLS certificate file (default
                           "C:\\Users\\USER\\.docker\\cert.pem")
```

```
MINGW64:/c/Users/USER/OneDrive - um5.ac.ma/Desktop/esiea-4th-vcc-project-main/esiea-4th-vcc-project-main
  events      Get real time events from the server
  exec        Run a command in a running container
  export      Export a container's filesystem as a tar archive
  history     Show the history of an image
  images      List images
  import      Import the contents from a tarball to create a filesystem image
  info        Display system-wide information
  inspect     Return low-level information on Docker objects
  kill        Kill one or more running containers
  load        Load an image from a tar archive or STDIN
  login       Log in to a Docker registry
  logout      Log out from a Docker registry
  logs        Fetch the logs of a container
  pause       Pause all processes within one or more containers
  port        List port mappings or a specific mapping for the container
  ps          List containers
  pull        Pull an image or a repository from a registry
  push        Push an image or a repository to a registry
  rename      Rename a container
  restart     Restart one or more containers
  rm          Remove one or more containers
  rmi         Remove one or more images
  run         Run a command in a new container
  save        Save one or more images to a tar archive (streamed to STDOUT by de
fault)
  search      Search the Docker Hub for images
  start       Start one or more stopped containers
  stats       Display a live stream of container(s) resource usage statistics
  stop        Stop one or more running containers
  tag         Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
  top         Display the running processes of a container
  unpause     Unpause all processes within one or more containers
  update      Update configuration of one or more containers
  version     Show the Docker version information
  wait        Block until one or more containers stop, then print their exit cod
es

Run 'docker COMMAND --help' for more information on a command.

To get more help with docker, check out our guides at https://docs.docker.com/go
/guides/

USER@DESKTOP-B2Q5BQ7 MINGW64 ~/OneDrive - um5.ac.ma/Desktop/esiea-4th-vcc-projec
t-main/esiea-4th-vcc-project-main
$ docker -v
Docker version 20.10.21, build baeda1f

USER@DESKTOP-B2Q5BQ7 MINGW64 ~/OneDrive - um5.ac.ma/Desktop/esiea-4th-vcc-projec
t-main/esiea-4th-vcc-project-main
$ docker pull postgres:13.9-alpine3.17
error during connect: In the default daemon configuration on Windows, the docker
 client must be run with elevated privileges to connect.: Post "http://%2F%2F.%2
Fpipe%2Fdocker_engine/v1.24/images/create?fromImage=postgres&tag=13.9-alpine3.17
": open //./pipe/docker_engine: The system cannot find the file specified.

USER@DESKTOP-B2Q5BQ7 MINGW64 ~/OneDrive - um5.ac.ma/Desktop/esiea-4th-vcc-projec
t-main/esiea-4th-vcc-project-main
$
```





2- Follow instructions in the README file to create and AWS S3 bucket that that will be used to store Udagram images. Make sure to configure your programmatic credentials locally.

Nous continuons à améliorer la console S3 pour la rendre plus rapide et plus facile à utiliser. Si vous avez des commentaires sur l'expérience r

Amazon S3 > Compartiments > Créer un compartiment

Créer un compartiment Info

Les compartiments sont des conteneurs pour les données stockées dans S3. En savoir plus

Configuration générale

Nom du compartiment

Udagram-said-8765

Le nom du compartiment doit être unique mondialement et ne peut pas contenir d'espaces ni de majuscules. Voir les règles de dénomination des compartiments

Région AWS

USA Est (Virginie du Nord) us-east-1

Copier les paramètres depuis un compartiment existant - *facultatif*

Seuls les paramètres de compartiment dans la configuration suivante sont copiés.

Sélectionner un compartiment

Propriété d'objets Info

Contrôlez la propriété des objets écrits dans ce compartiment à partir d'autres comptes AWS et l'utilisation des listes de contrôle d'accès (ACL). La propriété des objets détermine qui peut spécifier l'accès aux objets.

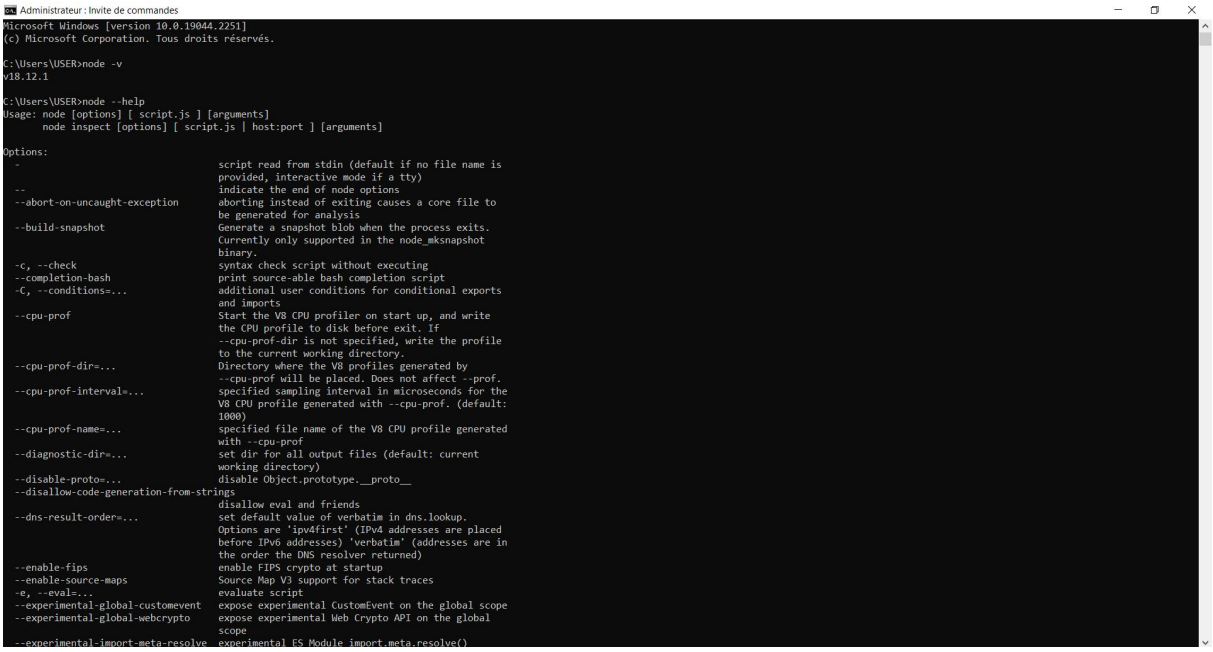
- (●) Listes ACL désactivées (recommandé)
  Tous les objets de ce compartiment sont gérés par ce compte. L'accès à ce compartiment et à ses objets est spécifié en utilisant uniquement des politiques.
- ( ) Listes ACL activées
  Les objets de ce compartiment peuvent être gérés par d'autres comptes AWS. L'accès à ce compartiment et à ses objets peut être spécifié à l'aide des listes ACL.

Propriété d'objets

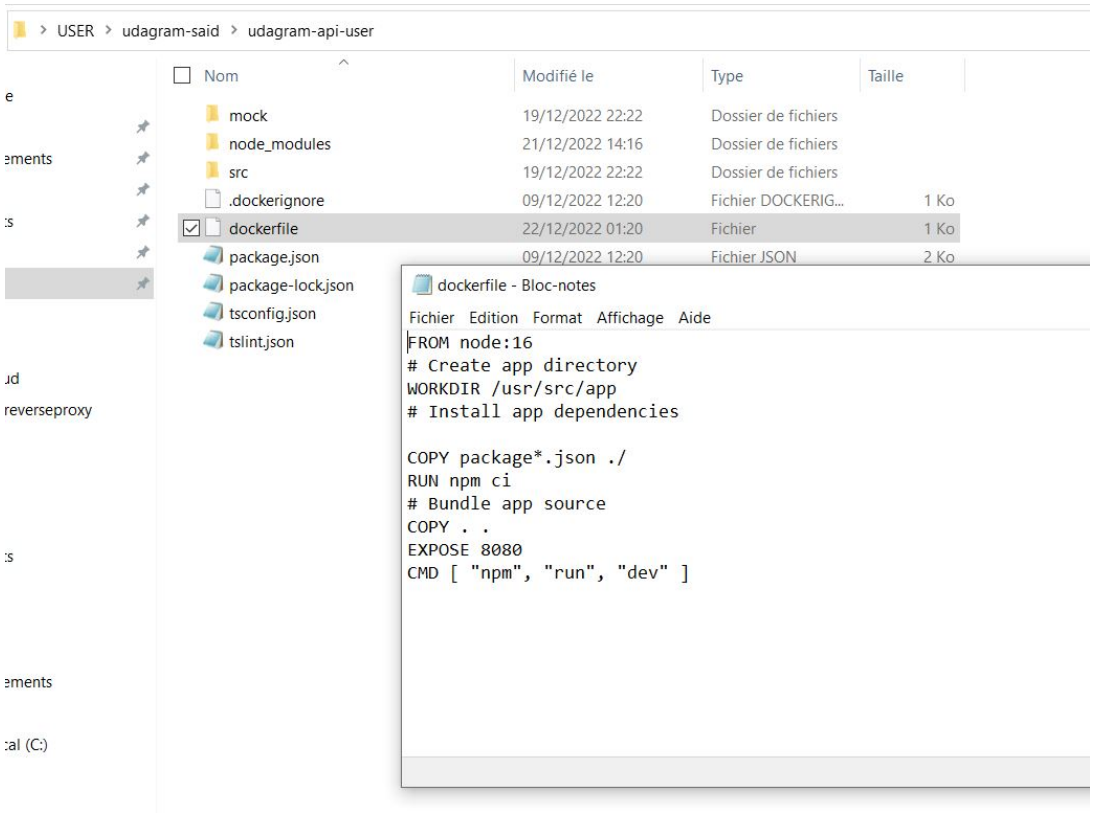
Propriétaire du compartiment appliqué

Paramètres de blocage de l'accès public pour ce compartiment

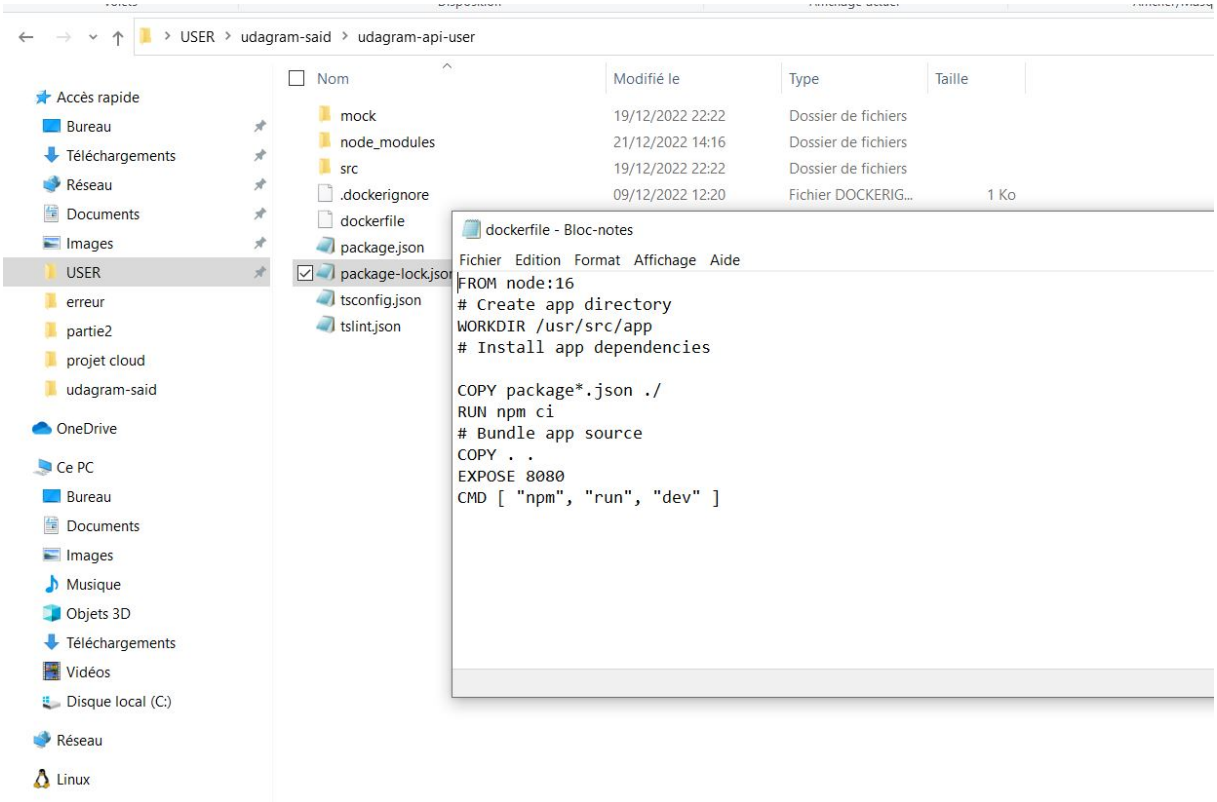
L'accès public aux compartiments et aux objets est accordé via des listes de contrôle d'accès (ACL), des stratégies de compartiment, de point d'accès ou tous ces éléments à la fois. Pour bloquer l'accès public à votre compartiment et aux objets qu'il contient, activez le paramètre Bloquer tous les accès publics. Il s'applique uniquement à ce compartiment et à ses points d'accès. AWS recommande de bloquer tous les accès publics, mais avant d'appliquer ces paramètres, vérifiez que vos applications fonctionneront correctement sans accès public. Si vous souhaitez autoriser un certain niveau d'accès public pour votre compartiment ou ses objets, vous pouvez personnaliser les paramètres



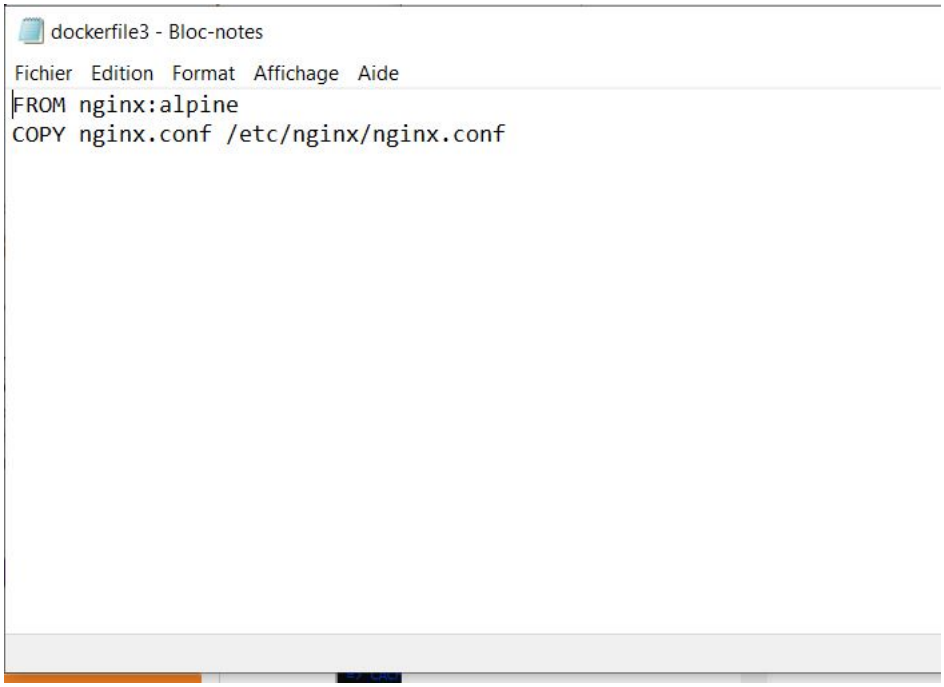
3- Create a Dockerfile for each project :



3.2- for frontend :



3.3- for reverseproxy :



4- Create image for each project and push to your Dockerhub account account :

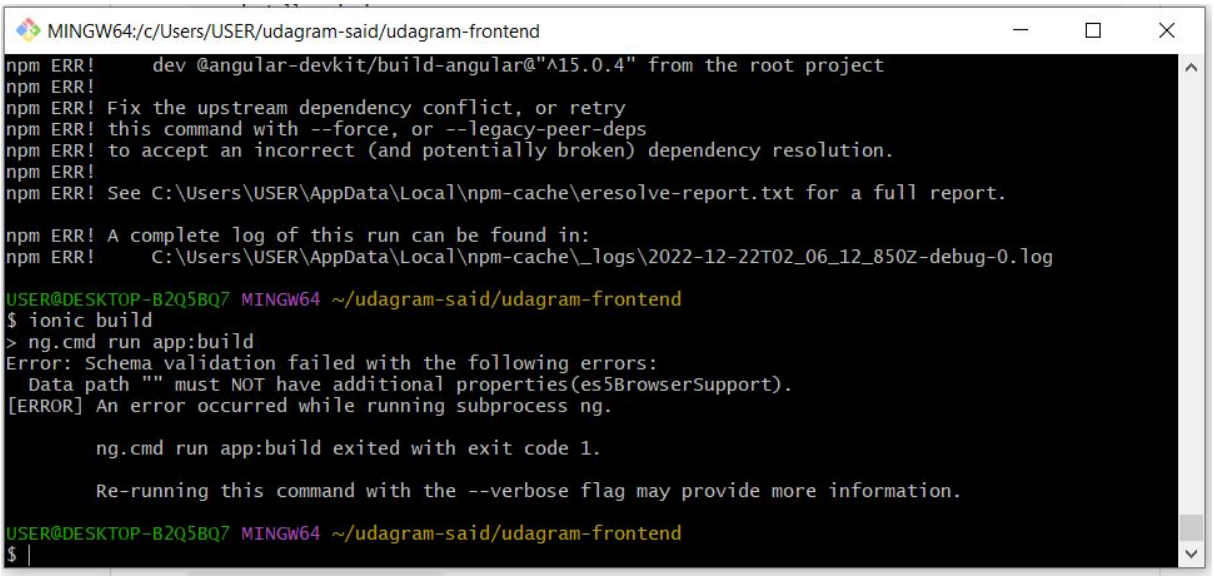
4.1- for backend API :

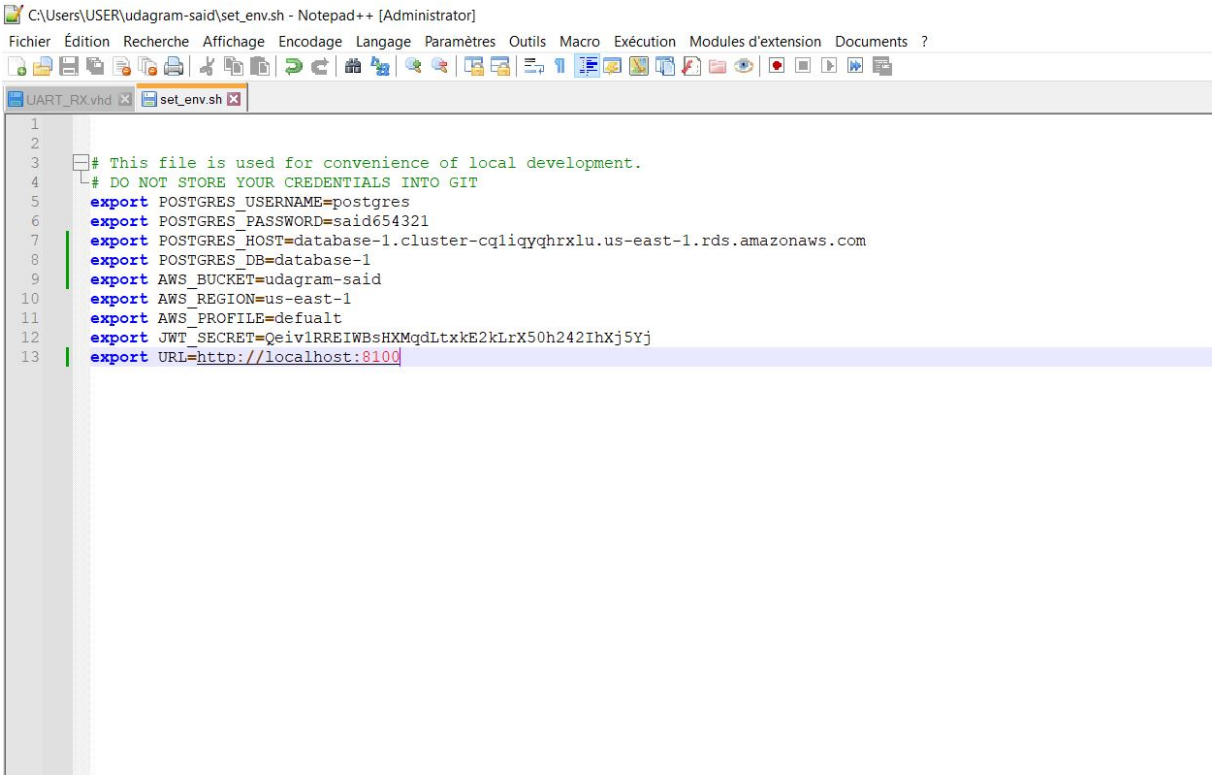
```
USER@DESKTOP-B2Q5BQ7 MINGW64 ~/udagram-said/udagram-api-user
$ ~docker build -t feedsaid .
bash: ~docker: command not found

USER@DESKTOP-B2Q5BQ7 MINGW64 ~/udagram-said/udagram-api-user
$ docker build -t feedsaid .
#1 [internal] load build definition from Dockerfile
#1 sha256:ac0b5788625bf989f3948dfef579b4f822f65d943146eae93184015d3b681c86
#1 transferring dockerfile: 32B done
#1 DONE 0.0s

#2 [internal] load .dockerignore
#2 sha256:0dfe28fec0dd27ab56efaff20d34abeb1b659965c34e1c10e73016453b6d3a13
#2 transferring context: 34B done
#2 DONE 0.0s

#3 [internal] load metadata for docker.io/library/node:16
#3 sha256:fcebe6595ccabc9db4d2477733f2265fb18ca302607dd9a9e445f4ebdfd1189a
#3 ...

#4 [auth] library/node:pull token for registry-1.docker.io
#4 sha256:7000ccba6c6672563a1688fe3f81da719f7ef1d5c950bbc5cc709ffca5b186dd
#4 DONE 0.0s

#3 [internal] load metadata for docker.io/library/node:16
#3 sha256:fcebe6595ccabc9db4d2477733f2265fb18ca302607dd9a9e445f4ebdfd1189a
#3 DONE 13.9s

#7 [internal] load build context
#7 sha256:b57e7b6bd0891b961543e35fd7429a7cb932b380cf9609d1cde8b6804a1d5129
#7 transferring context: 308.04kB 0.1s done
#7 DONE 0.1s

#5 [1/5] FROM docker.io/library/node:16@sha256:64e8bcdfdad6718050801a2639f7e6645
fdaf85ec37a98cdb61f6a5331217618
#5 sha256:0c5260e14e167594765889ea9130d460deef295c41d7d7aa8d1796cb40ce653e
#5 resolve docker.io/library/node:16@sha256:64e8bcdfdad6718050801a2639f7e6645fda
f85ec37a98cdb61f6a5331217618 0.1s done
```

- Push :

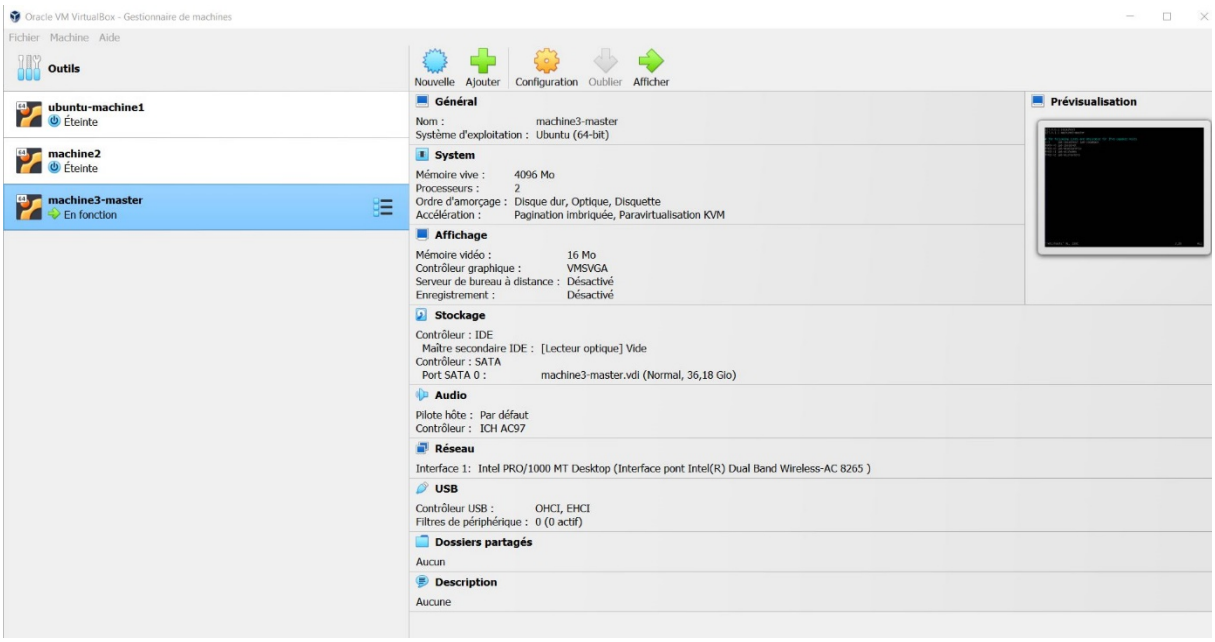
```
USER@DESKTOP-B2Q5BQ7 MINGW64 ~/udagram-said/udagram-api-user
$  docker login
Authenticating with existing credentials...
Login Succeeded

Logging in with your password grants your terminal complete access to your account.
For better security, log in with a limited-privilege personal access token. Learn more at https://docs.docker.com/go/access-tokens/

USER@DESKTOP-B2Q5BQ7 MINGW64 ~/udagram-said/udagram-api-user
$ docker tag feedsaid:latest said/example_image:latest

USER@DESKTOP-B2Q5BQ7 MINGW64 ~/udagram-said/udagram-api-user
$ docker tag feedsaid:latest said/feedsaid:latest

USER@DESKTOP-B2Q5BQ7 MINGW64 ~/udagram-said/udagram-api-user
$ docker push said/feedsaid:latest
The push refers to repository [docker.io/said/feedsaid]
4775b838ec7f: Preparing
25c12907a9c8: Preparing
cfca9adcf93a: Preparing
23b4f4eebbaa: Preparing
bfe15b22fcc1: Preparing
eb0530de5e36: Preparing
88ec30e85ff8: Preparing
cb5adc24fcf7: Preparing
b1721623f622: Preparing
713cb2b1e6c1: Preparing
c1842b709be1: Preparing
371587d8d651: Preparing
a416f5060b0c: Preparing
713cb2b1e6c1: Waiting
88ec30e85ff8: Waiting
cb5adc24fcf7: Waiting
c1842b709be1: Waiting
371587d8d651: Waiting
b1721623f622: Waiting
a416f5060b0c: Waiting
eb0530de5e36: Waiting
denied: requested access to the resource is denied

USER@DESKTOP-B2Q5BQ7 MINGW64 ~/udagram-said/udagram-api-user
$
```

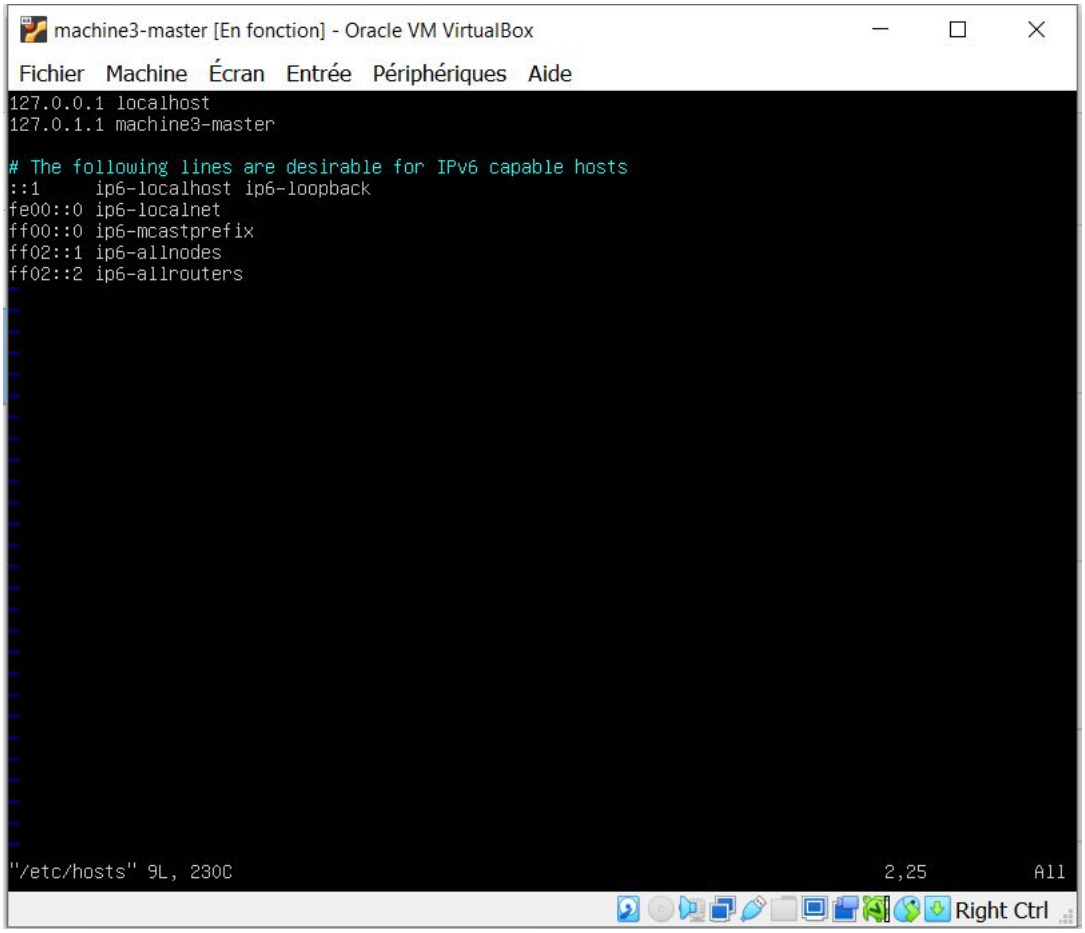
4.2- for frontend:





Party 2 :


Oracle VM VirtualBox - Gestionnaire de machines
Fichier Machine Aide
Outils
ubuntu-machine1
Éteinte
machine2
Éteinte
machine3-master
En fonction
Nouvelle
Ajouter
Configuration
Oublier
Afficher
Général
Nom : machine3-master
Système d'exploitation : Ubuntu (64-bit)
System
Mémoire vive : 4096 Mo
Processeurs : 2
Ordre d'amorçage : Disque dur, Optique, Disquette
Accélération : Pagination imbriquée, Paravirtualisation KVM
Affichage
Mémoire vidéo : 16 Mo
Contrôleur graphique : VMSVGA
Serveur de bureau à distance : Désactivé
Enregistrement : Désactivé
Stockage
Contrôleur : IDE
Maître secondaire IDE : [Lecteur optique] Vide
Contrôleur : SATA
Port SATA 0 : machine3-master.vdi (Normal, 36,18 Gio)
Audio
Pilote hôte : Par défaut
Contrôleur : ICH AC97
Réseau
Interface 1: Intel PRO/1000 MT Desktop (Interface pont Intel(R) Dual Band Wireless-AC 8265)
USB
Contrôleur USB : OHCI, EHCI
Filtres de périphérique : 0 (0 actif)
Dossiers partagés
Aucun
Description
Aucune
Prévisualisation

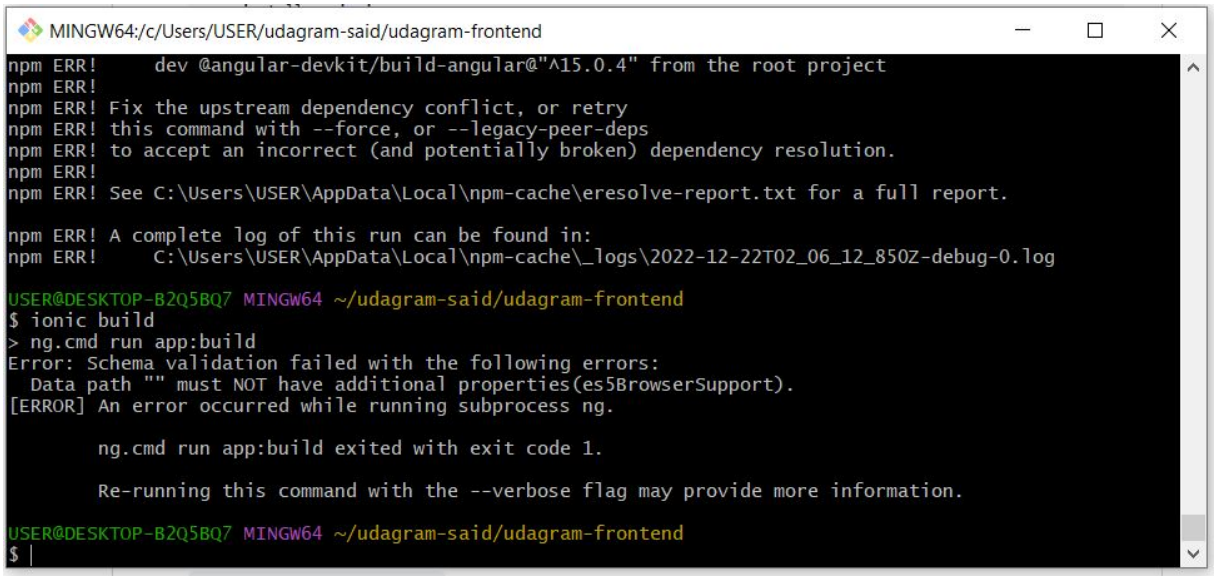

machine3-master [En fonction] - Oracle VM VirtualBox
Fichier Machine Écran Entrée Périphériques Aide
127.0.0.1 localhost
127.0.1.1 machine3-master
The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
"/etc/hosts" 9L, 230C
2,25
All
Right Ctrl

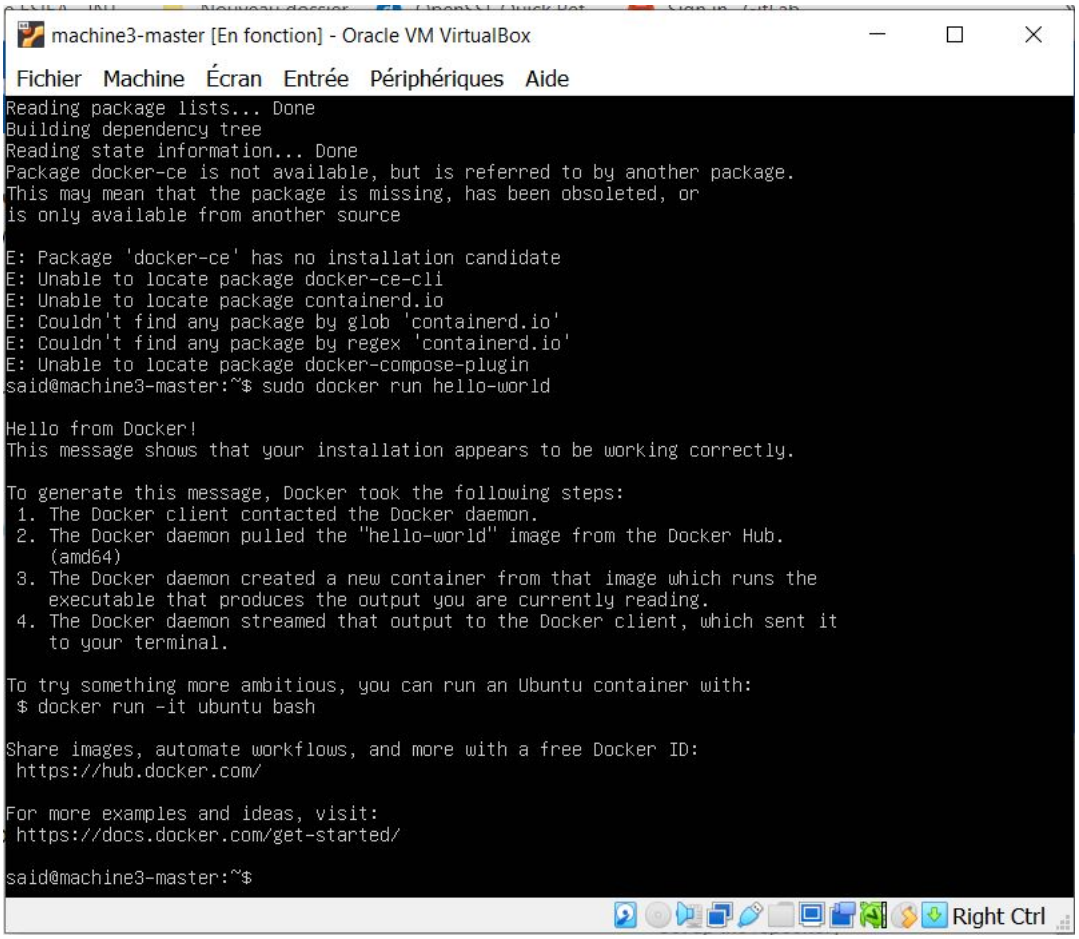
2- You MUST create a set of virtual machines that are part of Kubernetes cluster. One need to be the master and the two others the workers. The master will run Kubernetes components and the workers will run your Udagram application and two Kubernetes components (kubelet and kubeproxy) :

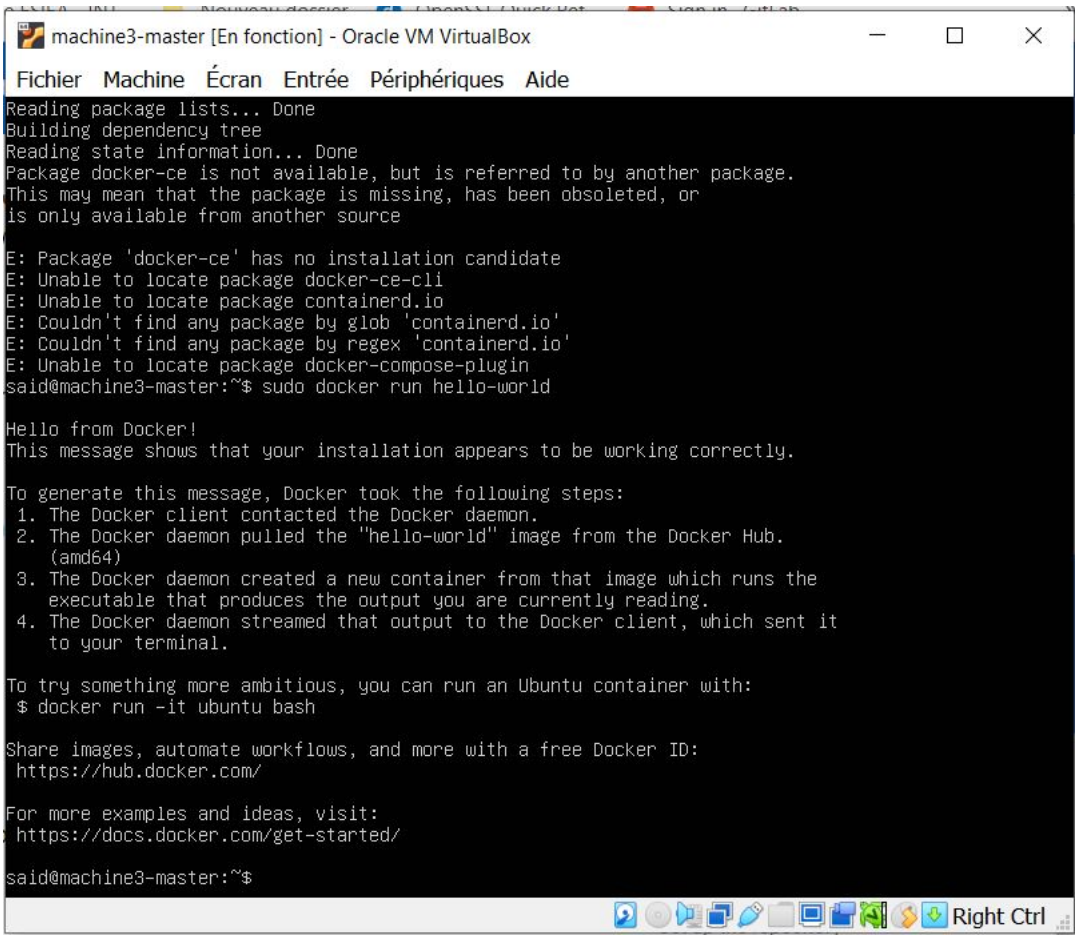
```
said@machine3-master:~$ ping google.fr
PING google.fr (142.250.178.131) 56(84) bytes of data.
64 bytes from par21s22-in-f3.1e100.net (142.250.178.131): icmp_seq=1 ttl=114 time=118 ms
64 bytes from par21s22-in-f3.1e100.net (142.250.178.131): icmp_seq=2 ttl=114 time=89.8 ms
64 bytes from par21s22-in-f3.1e100.net (142.250.178.131): icmp_seq=3 ttl=114 time=73.5 ms
64 bytes from par21s22-in-f3.1e100.net (142.250.178.131): icmp_seq=4 ttl=114 time=698 ms
64 bytes from par21s22-in-f3.1e100.net (142.250.178.131): icmp_seq=5 ttl=114 time=74.6 ms
64 bytes from par21s22-in-f3.1e100.net (142.250.178.131): icmp_seq=6 ttl=114 time=101 ms
64 bytes from par21s22-in-f3.1e100.net (142.250.178.131): icmp_seq=8 ttl=114 time=90.0 ms
64 bytes from par21s22-in-f3.1e100.net (142.250.178.131): icmp_seq=9 ttl=114 time=83.5 ms
64 bytes from par21s22-in-f3.1e100.net (142.250.178.131): icmp_seq=11 ttl=114 time=98.1 ms
64 bytes from par21s22-in-f3.1e100.net (142.250.178.131): icmp_seq=12 ttl=114 time=70.1 ms
64 bytes from par21s22-in-f3.1e100.net (142.250.178.131): icmp_seq=15 ttl=114 time=73.9 ms
64 bytes from par21s22-in-f3.1e100.net (142.250.178.131): icmp_seq=16 ttl=114 time=101 ms
64 bytes from par21s22-in-f3.1e100.net (142.250.178.131): icmp_seq=18 ttl=114 time=95.1 ms
64 bytes from par21s22-in-f3.1e100.net (142.250.178.131): icmp_seq=19 ttl=114 time=73.8 ms
64 bytes from par21s22-in-f3.1e100.net (142.250.178.131): icmp_seq=20 ttl=114 time=70.2 ms
64 bytes from par21s22-in-f3.1e100.net (142.250.178.131): icmp_seq=22 ttl=114 time=76.1 ms
64 bytes from par21s22-in-f3.1e100.net (142.250.178.131): icmp_seq=24 ttl=114 time=785 ms
64 bytes from par21s22-in-f3.1e100.net (142.250.178.131): icmp_seq=25 ttl=114 time=111 ms
^C
--- google.fr ping statistics ---
25 packets transmitted, 18 received, 28% packet loss, time 25062ms
rtt min/avg/max/mdev = 70.110/160.148/784.599/206.375 ms
said@machine3-master:~$ ^C
said@machine3-master:~$ sudo  apt update -y
Hit:1 http://fr.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://fr.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://fr.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:4 http://fr.archive.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Fetched 336 kB in 35s (9651 B/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
19 packages can be upgraded. Run 'apt list --upgradable' to see them.
said@machine3-master:~$
```

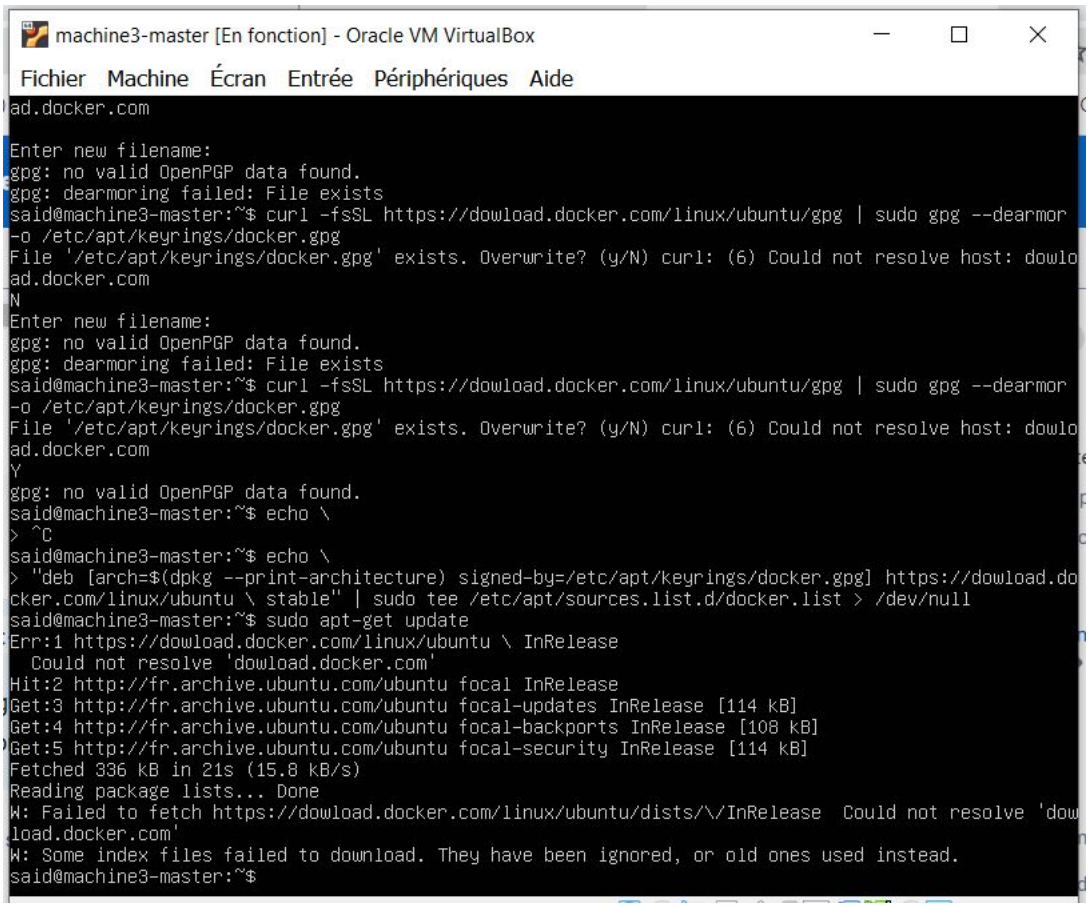
```
--- google.fr ping statistics ---
25 packets transmitted, 18 received, 28% packet loss, time 25062ms
rtt min/avg/max/mdev = 70.110/160.148/784.599/206.375 ms
said@machine3-master:~$ ^C
said@machine3-master:~$ sudo  apt update -y
Hit:1 http://fr.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://fr.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://fr.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:4 http://fr.archive.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Fetched 336 kB in 35s (9651 B/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
19 packages can be upgraded. Run 'apt list --upgradable' to see them.
said@machine3-master:~$ sudo swapoff -a
said@machine3-master:~$ sudo sed -i '/ swap / s/^/#/' /etc/fstab
said@machine3-master:~$ sudo systemctl stop firewalld
Failed to stop firewalld.service: Unit firewalld.service not loaded.
said@machine3-master:~$ sudo systemctl disable firewalld
Failed to disable unit: Unit file firewalld.service does not exist.
said@machine3-master:~$ cat <<EOF | sudo tee /etc/modules-load.d/machine3,conf
> overlay
> br_netfilter
> EOF
overlay
br_netfilter
said@machine3-master:~$ sudo modprobe overlay
said@machine3-master:~$ sudo modprobe br_netfilter
said@machine3-master:~$ cat <<EOF | sudo tee /etc/sysctl.d/machine3.conf
> net.bridge.bridge-nf-call-iptables = 1
> net.bridge.bridge-nf-call-ip6tables = 1
> net.ipv4.ip_forward = 1 cat ,,
> EOF
net.bridge.bridge-nf-call-iptables = 1
net.bridge.bridge-nf-call-ip6tables = 1
net.ipv4.ip_forward = 1 cat ,,
said@machine3-master:~$
```

Installation docker :

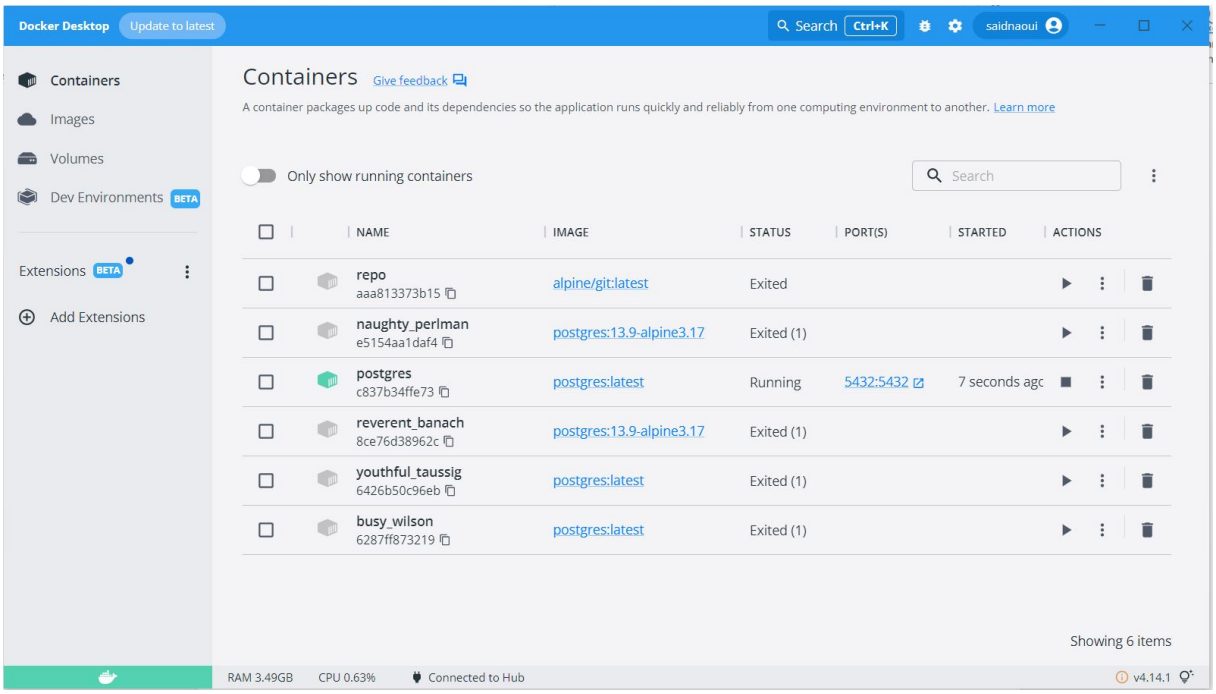





machine3-master [En fonction] - Oracle VM VirtualBox
Fichier Machine Écran Entrée Périphériques Aide
Reading package lists... Done
Building dependency tree
Reading state information... Done
Package docker-ce is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source
E: Package 'docker-ce' has no installation candidate
E: Unable to locate package docker-ce-cli
E: Unable to locate package containerd.io
E: Couldn't find any package by glob 'containerd.io'
E: Couldn't find any package by regex 'containerd.io'
E: Unable to locate package docker-compose-plugin
said@machine3-master:~$ sudo docker run hello-world
Hello from Docker!
This message shows that your installation appears to be working correctly.
To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
(amd64)
3. The Docker daemon created a new container from that image which runs the
executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
to your terminal.
To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash
Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/
For more examples and ideas, visit:
https://docs.docker.com/get-started/
said@machine3-master:~$
Right Ctrl


machine3-master [En fonction] - Oracle VM VirtualBox
Fichier Machine Écran Entrée Périphériques Aide
ad.docker.com
Enter new filename:
gpg: no valid OpenPGP data found.
gpg: dearmoring failed: File exists
said@machine3-master:~$ curl -fsSL https://dowload.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor
-o /etc/apt/keyrings/docker.gpg
File '/etc/apt/keyrings/docker.gpg' exists. Overwrite? (y/N) curl: (6) Could not resolve host: dowlo
ad.docker.com
N
Enter new filename:
gpg: no valid OpenPGP data found.
gpg: dearmoring failed: File exists
said@machine3-master:~$ curl -fsSL https://dowload.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor
-o /etc/apt/keyrings/docker.gpg
File '/etc/apt/keyrings/docker.gpg' exists. Overwrite? (y/N) curl: (6) Could not resolve host: dowlo
ad.docker.com
Y
gpg: no valid OpenPGP data found.
said@machine3-master:~$ echo \
> ^C
said@machine3-master:~$ echo \
> "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://dowload.do
cker.com/linux/ubuntu \ stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
said@machine3-master:~$ sudo apt-get update
Err:1 https://dowload.docker.com/linux/ubuntu \ InRelease
Could not resolve 'dowload.docker.com'
Hit:2 http://fr.archive.ubuntu.com/ubuntu focal InRelease
Get:3 http://fr.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:4 http://fr.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:5 http://fr.archive.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Fetched 336 kB in 21s (15.8 kB/s)
Reading package lists... Done
W: Failed to fetch https://dowload.docker.com/linux/ubuntu/dists/\/InRelease Could not resolve 'dow
load.docker.com'
W: Some index files failed to download. They have been ignored, or old ones used instead.
said@machine3-master:~$

```
--- google.fr ping statistics ---
25 packets transmitted, 18 received, 28% packet loss, time 25062ms
rtt min/avg/max/mdev = 70.110/160.148/784.599/206.375 ms
said@machine3-master:~$ ^C
said@machine3-master:~$ sudo  apt update -y
Hit:1 http://fr.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://fr.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://fr.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:4 http://fr.archive.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Fetched 336 kB in 35s (9651 B/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
19 packages can be upgraded. Run 'apt list --upgradable' to see them.
said@machine3-master:~$ sudo swapoff -a
said@machine3-master:~$ sudo sed -i '/ swap / s/^/#/' /etc/fstab
said@machine3-master:~$ sudo systemctl stop firewalld
Failed to stop firewalld.service: Unit firewalld.service not loaded.
said@machine3-master:~$ sudo systemctl disable firewalld
Failed to disable unit: Unit file firewalld.service does not exist.
said@machine3-master:~$ cat <<EOF | sudo tee /etc/modules-load.d/machine3,conf
> overlay
> br_netfilter
> EOF
overlay
br_netfilter
said@machine3-master:~$ sudo modprobe overlay
said@machine3-master:~$ sudo modprobe br_netfilter
said@machine3-master:~$ cat <<EOF | sudo tee /etc/sysctl.d/machine3.conf
> net.bridge.bridge-nf-call-iptables = 1
> net.bridge.bridge-nf-call-ip6tables = 1
> net.ipv4.ip_forward = 1 cat ,,
> EOF
net.bridge.bridge-nf-call-iptables = 1
net.bridge.bridge-nf-call-ip6tables = 1
net.ipv4.ip_forward = 1 cat ,,
said@machine3-master:~$
```



Party 3 :

1.Cloud computing is the process of using the internet to manage software systems that users can access remotely through any connected device. Cloud computing changes the traditional method of controlling software applications.

Benefits :

- Efficiency
- Scalability
- Continuity
- Collaboration
- Loss prevention

There is no cloud , it is just someone else computer means that others manage hardware for you ,and you just need to deploy your services , there is no need to take care off the hardware services …

2. EC2 is a cloud platform provided by Amazon that offers secure, and resizable compute capacity. Its purpose is to enable easy access and usability to developers for web-scale cloud computing, while allowing for total control of your compute resources.EC2

Amazon Relational Database Service (Amazon RDS) makes database configuration, management, and scaling easy in the cloud. Automate tedious tasks such as hardware provisioning, database arrangement, patching, and backups – cost-effectively and proportionate to your needs.

Amazon S3, at its core, facilitates object storage, providing leading scalability, data availability, security, and performance. Businesses of vast sizes can leverage S3 for storage and protect large sums of data for various use cases, such as websites, applications, backup, and more.

Lambda permits you to run code without owning or managing servers. Users only pay for the compute time consumed. Operate code for nearly any application or backend

utility without administration. Users just upload the code, and Lambda does the rest, which provides precise software scaling and extensive availability.

CloudFront is a content delivery network platform that executes at rapid rates with the secure distribution of data, videos, apps, and APIs on a global scale with low delay-times. Connected with the global infrastructure of AWS, CloudFront integrates seamlessly with systems like Amazon S3, Amazon EC2, AWS Shield and Lambda@Edge to manage custom code, personalizing the experience.

3. NAT stands for network address translation. It's a way to map multiple local private addresses to a public one before transferring the information. Organizations that want multiple devices to employ a single IP address use NAT, as do most home routers.

4. Private IP address of a system is the IP address that is used to communicate within the same network. Using private IP data or information can be sent or received within the same network.

Public IP address of a system is the IP address that is used to communicate outside the network. A public IP address is basically assigned by the ISP (Internet Service Provider).