Université Claude Bernard Lyon 1
Département Informatique
February 29th, 2024 – Session 1
M2 DISS

Exam : UE Smart Services Science

A. SSH

A.1. Consider the following screenshot.

```
$ ssh -i ~/Telechargements/id_rsa ubuntu@192.168.249.203
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0644 for '/home/jp/Telechargements/id_rsa' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "/home/jp/Telechargements/id_rsa": bad permissions
ubuntu@192.168.249.203: Permission denied (publickey).
```

Why this warning message and what will you do about it?

A.2. Consider the following screenshot.

```
 $ ssh -i ~/.ssh/id_rsa ubuntu@192.168.249.203
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
6e:45:f9:a8:af:38:3d:a1:a5:c7:76:1d:02:f8:77:00.
Please contact your system administrator.
Add correct host key in /home/hostname/.ssh/known_hosts to get rid of this message.
Offending RSA key in /var/lib/sss/pubconf/known_hosts:4
RSA host key for pong has changed and you have requested strict checking.
Host key verification failed.
```

Why this error message and what will you do about it?

B. Docker

Here is the list of commands entered on a VM instantiated from an Ubuntu 22.04 LTS - Docker Ready image. This VM got the IP address 192.168.239.42

```
1.  docker pull mysql:latest
2.  docker pull nginx:latest
3.  docker pull phpmyadmin/phpmyadmin

4.  docker network create --subnet "172.18.100.0/24" interne
5.  docker run -d --name dockbase --hostname basededonnee -e MYSQL_ROOT_PASSWORD=passroot
    -v /home/ubuntu/docker/datatest/:/var/lib/mysql/ --net interne -ip 172.18.100.10
    --network-alias base mysql
6.  docker exec -it dockbase mysql -u root -ppassroot -e 'CREATE DATABASE BASE_A;'
7.  docker exec -it dockbase mysql -u root -ppassroot -e 'CREATE DATABASE BASE_B;'
8.  docker exec -it dockbase mysql -u root -ppassroot -e 'CREATE DATABASE BASE_C;'
9.
10. vim /home/ubuntu/docker/nginx.conf #  on édite le fichier nginx.conf pour y ajouter à la section server
           location /phpMyAdmin {
11.              proxy_set_header Host $host;
12.              proxy_set_header X-Real-IP $remote_addr;
13.              proxy_pass http://172.18.100.11/;
14.           }
15.
16. docker run -d --name dockFront -p 80:80 --net interne --ip 172.18.100.15 -v
    /home/ubuntu/docker/nginx.conf:/etc/nginx/nginx.conf:ro -v /home/ubuntu/docker
    /nginx/:/www/ nginx:latest
17.
18. docker restart dockFront # Si on modifie nginx.conf
19.
20. docker run -d -e PMA_HOST=dockbase -e PMA_ABSOLUTE_URI=/phpMyAdmin --net interne --ip
    172.18.100.11 --name dockMyAdm phpmyadmin/phpmyadmin
```

B.1 How many containers are started? Give its name for each of them.

B.2 What URL should I enter to access the phpMyAdmin service?

B.3 Build a diagram representing the result of these commands. On the diagram we should see the containers represented by rectangles, the internal or external IP addresses, the port number, the interconnection subnet, the names, the logical links, …

C. Docker Swarm

Here is the list of commands entered on VMs instantiated from an Ubuntu 22.04 LTS - Docker Ready image. Each VM is respectively named `Manager`, `Worker1` and `Worker2`.

```
Manager$ docker swarm init --advertise-addr 192.168.239.42
Worker1$ docker swarm join --token letokenrenvoyeparleswarminit 192.168.239.42:2377
Worker2$ docker swarm join --token letokenrenvoyeparleswarminit 192.168.239.42:2377

docker service create   --name=viz  --publish=8080:8080/tcp   --constraint=node.role==manager
--mount=type=bind,src=/var/run/docker.sock,dst=/var/run/docker.sock dockersamples/visualizer


docker network create --driver overlay --subnet 172.21.21.0/24 --attachable reseau


docker service create --replicas 3 --network reseau --name web -p 80:80 nginx

docker service create --replicas 1 --constraint=node.role==manager --network reseau
-p 5000:5000 --name registry registry:2
```

```
docker pull hello-world
docker image tag hello-world localhost:5000/my-hello-world
docker push localhost:5000/my-hello-world
docker image remove hello-world
docker image remove localhost:5000/my-hello-world
docker pull localhost:5000/my-hello-world


docker service create  -d --network reseau  -e ENV_DOCKER_REGISTRY_HOST=registry
-e ENV_DOCKER_REGISTRY_PORT=5000 -p 8443:80  --name regbrowser
konradkleine/docker-registry-frontend:v2
```

C.1 How many services are started? Give their name for each of them.

C.2 What URL should I enter to access the registry viewer service?

C.3 Make a clean diagram representing the end result of these commands. On the diagram we should see the containers/services represented by rectangles, internal or external IP addresses, port number, interconnection subnet, names, logical links, ...