English communication

Session 1 : Docker and Git

The Course :

What is Git?

Git is a software for version control. It allows developers to work on the same code at the same time and manage differences and changes made to the code over time. Git is also used to track differences between different versions of a project and to facilitate collaboration and communication among developers.

What is Docker?

Docker is a platform that allows you to easily develop, deploy, and run applications in containers. A container is a lightweight, standalone, and executable software package that includes everything needed to run an application, including the code, runtime, system tools, libraries, and settings. Docker allows developers to package an application and its dependencies into a container, making it easier to deploy and run the application in any environment. With Docker, developers can work with a consistent set of tools and dependencies, regardless of the environment they are working in, which helps to reduce the number of compatibility issues they face. Additionally, Docker makes it easy to manage containers and the applications running inside them, providing a powerful platform for both development and production environments.

Mission 1 :

In mission 1 we will install git to have the presentation and the docker app file, then we will install docker to build and push our docker image. First we have to edit and complete the Dockerfile, then we build the docker with the app file.

Issues encounter :

During mission 1, we will have some issues. Firstly the proxy forbide the connexion between our virtual machine and the docker depot or the git. Secondly we don't have enough space to build the container.

Technical commands for mission 1 :

Installation of Git :

```
apt install git
cd ~
git init
```

Installation of Docker :

https://docs.docker.com/engine/install/debian/#prerequisites

Configure the proxy :

```
général : export http_proxy=http://192.168.2.7:3128
export https_proxy=http://192.168.2.7:3128
```

Docker :

```
mkdir /etc/systemd/system/docker.service.d/
nano /etc/systemd/system/docker.service.d/http-proxy.conf
[Service]
Environment="HTTP_PROXY=http://192.168.2.7:3128/"
Environment="HTTPS_PROXY=http://192.168.2.7:3128/"
systemctl daemon-reload
systemctl restart docker
# télécharger (pull) et executer (run) d'un conteneur de test
docker run hello-world
```

Clone the git repo on my machine locally :

```
git clone https://github.com/mperochon/A2SR.git
```

Complete and build a docker with the dockerfile :

File :

```
# Description : It is the image use as base
FROM node:18-alpine

ENV HTTP_PROXY 'http://192.168.2.7:3128'
ENV HTTPS_PROXY 'http://192.168.2.7:3128'

RUN npm config set proxy http://192.168.2.7:3128
RUN npm config set https-proxy http://192.168.2.7:3128


# Explain : RUN is used to execute command in our contener
#RUN apk add --no-cache python2 g++ make
```

```
# Explain : It is used to modify the repository (it's like cd).
# So all the following command will be executed in this repo
WORKDIR /app

# Description : To bundle the app's source code inside the Docker image
COPY . .

# Explain :
RUN npm install --production

# Explain : CMD is always at the end.
# It allow the contener to know wich command execute at the beginning
# It is for run the app. Here we will use index.js to start
CMD ["node", "src/index.js"]
```

Command :

```
root@debian11:~# docker build -t my_docker_build .
root@debian11:~# docker run -p 127.0.0.1:3000:3000 my_docker_build
```

Mission 2 :

Now we have to build a docker image, and then push it into our own docker hub repository. For that we will have to create a docker hub account. When we have created a docker account. Create a repository in docker hub and then tag and push the image on docker hub.

Issues encounter :

We have encountered some issues. Firstly the proxy still forbide our docker connection. We need to make the proxy in the docker files.

Technical commands for mission 2 :

Login to your docker account :

```
docker login
username :
password :
```

Create a repository on docker hub :

example : noahdanel/english

Push the image :

```
docker tag my_docker_build noahdanel/english
docker push noahdanel/english
```

https://hub.docker.com/repository/docker/noahdanel/english/general

Mission 3 :

In mission 3 we create a github repository with our own account. Finally we paste the link in a mail to Mister Perochon and that's finished.

Issue encounter :

We can’t login with a regular password to push our file so we need to create a token :
https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/creating-a-personal-access-token

Technical command

```
echo "# english" >> README.md
git init
git config --global user.email "noah.danel@etu.univ-nantes.fr
git config --global user.name "noah.danel"
git add README.md
git add Course_1/
git add
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/noahdanel/english.git
git push -u origin main
```

https://github.com/noahdanel/english