**Docker : CI & Jenkins**

**I - Docker dans Jenkins - DooD (Docker-outside-of-Docker)**

http://container-solutions.com/2015/03/running-docker-in-jenkins-in-docker/

Dans cette première partie, nous allons monter le "Docker.sock" dans un container pour créer des containers "frères" : DooD (Docker-outside-of-Docker), pour différencier de DinD (Docker-in-Docker), où on une version complète et isolée de Docker est installée dans un container. Ici, nous allons créer un container jenkins afin de pouvoir tester et lancer des container par des jobs jenkins.

**# PREREQUIS VM VIRTUALBOX**

image : ubuntu-14.04-desktop-amd64.iso

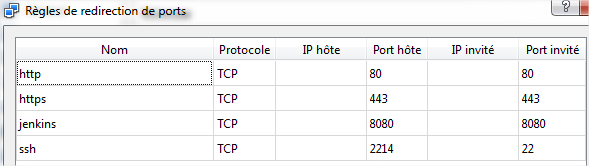
nom : server1.example.com

user/user

root/password

# installer ssh (permet de pouvoir se connecter à la VM via putty)

apt-get openssh-server



**# INSTALLATION DOCKER**

# http://docs.docker.com/installation/ubuntulinux/

sudo apt-get update

wget -qO- https://get.docker.com/ | sh

# ajouter user au groupe docker

sudo usermod -aG docker user

# tester l'installation de docker

sudo docker run hello-world

# créer un répertoire jenkins pour y ajouter notre Dockerfile

mkdir jenkins\_dood

cd jenkins\_dood/

# ajouter le Dockerfile

nano Dockerfile

FROM jenkins:1.596.2

USER root

RUN apt-get update \

&& apt-get install -y sudo \

&& rm -rf /var/lib/apt/lists/\*

RUN echo "jenkins ALL=NOPASSWD: ALL" >> /etc/sudoers

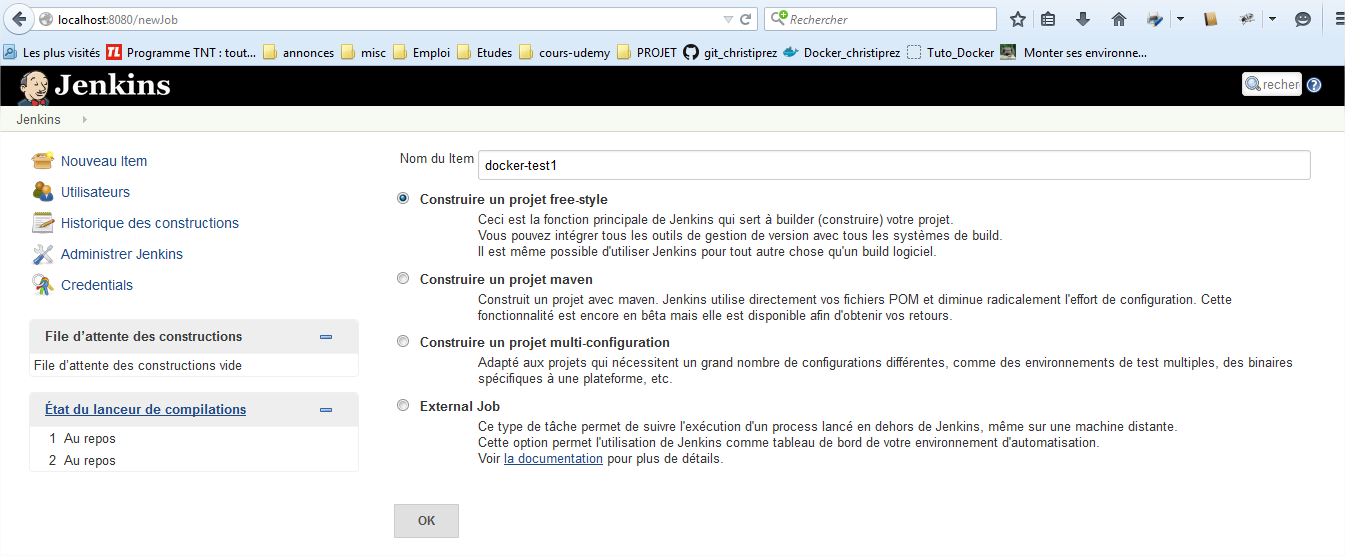
USER jenkins

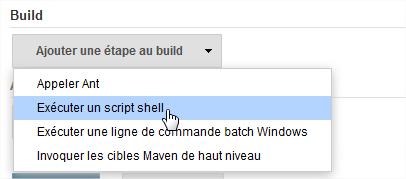
# builder notre container "jenkins\_dood" puis l'exécuter

sudo docker build -t jenkins\_dood .

sudo docker run -d -v /var/run/docker.sock:/var/run/docker.sock -v $(which docker):/usr/bin/docker -p 8080:8080 jenkins\_dood

Dans navigateur (machine hôte) : http://localhost:8080, créer un nouveau job



Dans configuration, 

Saisir " sudo docker run hello-world", cliquer sur "Save", puis sur "Lancer un build".



Nous pouvons maintenant exécuter avec succès des commandes Docker dans notre conteneur Jenkins.