Yasmine Meriah Matser 2 (CR)

TP 1 : Docker

Le contenu de fichier Docker compose :

version: '3'

services:

  nginx1:

    image: nginx

    container\_name: nginx1

    ports:

      - "8081:80"

  nginx2:

    image: nginx

    container\_name: nginx2

    ports:

      - "8082:80"

  nginx3:

    image: nginx

    container\_name: nginx3

    ports:

      - "8083:80"

  nginx4:

    image: nginx

    container\_name: nginx4

    ports:

      - "8084:80"

  loadbalancer:

    image: nginx

    container\_name: loadbalancer

    ports:

      - "8085:80"

    depends\_on:

      - nginx1

      - nginx2

      - nginx3

      - nginx4

Le contenu de dockerfile :

FROM ubuntu:16.04 AS ubuntu

ENV JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64

RUN apt update && \

# install required tools

    apt install -y git unzip ant build-essential \

                   cmake git libgtk2.0-dev pkg-config libavcodec-dev libavformat-dev libswscale-dev \

                   python-dev python-numpy libtbb2 libtbb-dev libjpeg-dev libpng-dev libtiff-dev libdc1394-22-dev \

                   python3 python3-dev python3-numpy \

                   software-properties-common debconf-utils && \

# install openjdk-8

    apt install -y openjdk-8-jdk && \

# libjasper-dev

    curl -fs http://security.ubuntu.com/ubuntu/pool/main/j/jasper/libjasper1\_1.900.1-debian1-2.4ubuntu1.2\_amd64.deb -o /tmp/libjasper1.deb && \

    curl -fs http://security.ubuntu.com/ubuntu/pool/main/j/jasper/libjasper-dev\_1.900.1-debian1-2.4ubuntu1.2\_amd64.deb -o /tmp/libjasper-dev.deb && \

    apt install /tmp/libjasper1.deb /tmp/libjasper-dev.deb && \

    rm -rf /tmp/\* && \

# download and prepare opencv

    curl -fsL https://github.com/opencv/opencv/archive/3.4.10.zip -o /tmp/opencv.zip && \

    cd /tmp && \

    unzip opencv.zip && \

    mv opencv-\* opencv && \

    cd opencv && \

    mkdir build && \

# build opencv

    cd /tmp/opencv/build && \

    cmake \

    -D CMAKE\_BUILD\_TYPE=Release \

    -D CMAKE\_INSTALL\_PREFIX=/usr/local \

    -D WITH\_FFMPEG=OFF \

    -D WITH\_IPP=OFF \

    -D WITH\_OPENEXR=OFF \

    -D BUILD\_EXAMPLES=OFF \

    -D BUILD\_ANDROID\_EXAMPLES=OFF \

    -D INSTALL\_PYTHON\_EXAMPLES=OFF \

    -D BUILD\_DOCS=OFF \

    -D BUILD\_opencv\_python2=OFF \

    -D BUILD\_opencv\_python3=OFF \

    -D BUILD\_SHARED\_LIBS=OFF \

    -D BUILD\_TESTS=OFF \

    -D BUILD\_PERF\_TESTS=OFF \

    .. && \

    make -j8

#RUN cd /tmp/opencv/build && make install

Capture de Resultat :



