Dockerfile for building a dockerimage for jupyter notebook

The world is into data science and there are lot of people using jupyter-notebook for their Data science Projects. There are lot of python packages being used in Data science projects. When these projects need to be deployed on other machine or share with colleagues or clients, the user is facing lots of installation problems before running the code/project. This is a time consuming process. Fortunately, for the industry there is an another tool (Dockers and Containers) which is helping the developers, clients, customers and companies to deploy the software projects without much worry of installation of necessary packages.

In this blog I will create a docker image which consists of notebook and this can be deployed or used during the development or testing phases of your Projects. Here a Docker image is nothing but a container which consists of all the necessary Packages installed and since I am focusing towards jupyter-notebook , we will create a container which deploy notebook.

I assume that you have installed docker in your machines before proceeding to next steps. If not please refer to https://docs.docker.com/install/. You can confirm the installation of docker using the following command.

```
(base) [sairajesh@ml-dl notebooks]$ docker --version
Docker version 19.03.8, build afacb8b
```

Now create a directory, in my case since I am using Debian O.S , I am creating docker-ds-nlp: using mkdir/docker-ds-nlp

Now we need to make a Dockerfile to build the image:

Dockerfile:

- I am using ubuntu image so creating the docker image in the ubuntu environment .Add the following line in Dockerfile

```
FROM ubuntu
```

- Next few of the packages which I wanted in my ubuntu image is installed using RUN command. Added these lines in my Dockerfile

```
RUN apt-get update && apt-get install -y wget
RUN apt-get update && apt-get install -y bzip2
```

- Now installation of Anaconda python in my container. Adding the following lines.

```
RUN wget https://repo.continuum.io/archive/Anaconda3-2020.02-Linux-x86_64.sh
RUN bash Anaconda3-2020.02-Linux-x86_64.sh -b



ENV PATH /root/anaconda3/bin:$PATH


RUN apt-get update && conda update conda
```

The above lines will install anaconda python , set the path and also update anaconda in our container.

- No I am installing all the necessary packages for my python project using conda. Following are the lines , which are added.

```
RUN apt-get update && conda install -c anaconda numpy
RUN apt-get update && conda install -c anaconda pandas
RUN apt-get update && conda install -c anaconda nltk
```

- Making a directory /opt/notebooks using
  RUN mkdir/opt/notebooks

- Now we are configuring to open the notebook, using the following lines.

```
RUN jupyter notebook --generate-config --allow-root
RUN echo "c.NotebookApp.password = u'sha1:6a3f528eec40:6e896b6e4828f525a6e20e5411cd1c8075d68619'" >> /root/.jupyter/jupyter_notebook_config.py


EXPOSE 8888



CMD ["jupyter", "notebook", "--allow-root", "--notebook-dir=/opt/notebooks", "--ip='*'", "--port=8888", "--no-browser"]
```

- Now build the docker container using **docker build -t docker-ds-nlp .**

```
 ---> Using cache
 ---> 291c0fdf442e
Step 3/20 : RUN apt-get update && apt-get install -y bzip2
 ---> Using cache
 ---> 5759b611f59e
Step 4/20 : RUN wget https://repo.continuum.io/archive/Anaconda3-2020.02-Linux-x86_64.sh
 ---> Using cache
 ---> 6e308ab4aa6d
Step 5/20 : RUN bash Anaconda3-2020.02-Linux-x86_64.sh -b
 ---> Using cache
 ---> 4e9289332796
Step 6/20 : ENV PATH /root/anaconda3/bin:$PATH
 ---> Using cache
 ---> c1097297ed29
Step 7/20 : RUN apt-get update && conda update conda
 ---> Using cache
 ---> 025974447f3b
Step 8/20 : RUN apt-get update && conda install -c anaconda numpy
 ---> Using cache
 ---> 20c7ead46cdd
Step 9/20 : RUN apt-get update && conda install -c anaconda pandas
 ---> Using cache
 ---> a0f7bb98402d
Step 10/20 : RUN apt-get update && conda install -c anaconda nltk
 ---> Using cache
 ---> 32df6e076fdc
Step 11/20 : RUN apt-get update && conda install -c anaconda gensim
 ---> Using cache
 ---> fd3874a2f08a
Step 12/20 : RUN apt-get update &&conda install -c anaconda urllib3
 ---> Using cache
 ---> 48c70b5ebdb4
Step 13/20 : RUN apt-get update && conda install -c anaconda beautifulsoup4
 ---> Using cache
 ---> ad6e66f789c9
Step 14/20 : RUN apt-get update && conda install -c conda-forge pdfminer.six
 ---> Using cache
 ---> f680c9cd54a5
Step 15/20 : RUN apt-get update &&conda install -c anaconda scikit-learn
 ---> Using cache
 ---> feba47a626d9
Step 16/20 : RUN mkdir /opt/notebooks
 ---> Using cache
 ---> 2400fb98550c
Step 17/20 : RUN jupyter notebook --generate-config --allow-root
 ---> Using cache
 ---> c42534556323
Step 18/20 : RUN echo "c.NotebookApp.password = u'sha1:6a3f528eec40:6e896b6e4828f525a6e20e5411cd1c8075d68619'" >> /root/.jupyter/jupyter_notebook_config.py
 ---> Using cache
 ---> c4768ed36e21
Step 19/20 : EXPOSE 8888
 ---> Using cache
 ---> eb3c77ffd1e8
Step 20/20 : CMD ["jupyter", "notebook", "--allow-root", "--notebook-dir=/opt/notebooks", "--ip='*'", "--port=8888", "--no-browser"]
 ---> Using cache
 ---> efaf511ec675
Successfully built efaf511ec675
Successfully tagged docker-ds-nlp:latest
(base) [sairajesh@ml-dl docker-ds-nlp]$
```

- Since I have already built the container , its using cache for me. But if you are building for the first time, it downloads and installs and image is successfully built after some time.
- Now run the docker image;

  sudo docker run --name docker-ds-nlp -p 8888:8888 -v "$PWD/notebook:/opt/notebooks" -d docker-ds-nlp

  A token is generated as follows:
  b48b7afcf543eae196f2953b7a8097bbd01f4af43ca7b77617f6da534f176adf

  You can open the http:localhost:8888
- You will find the notebook , use root as password and access it.

Issues you might face:

- **Some times docker will not be able to download the image, check your proxy for that.**
- **Sometimes your ubuntu image in the container will not be able to download the packages in it, you need to set a dns address.**