Introduction to Docker for Data Science

Konstantinos Charalampous

17 December 2018

Contents

- Motivation – Problem
- What is Docker
- Why Docker
- Business Value
- Docker for Data Science
- Use Case: Containerizing ASA
- Final thoughts: First impressions as a beginner in Docker
- Further Reading

FULL DISCLAIMER: I AM IN NO WAY AN EXPERT IN DOCKER.
This is my experience on learning and working on Docker for the first time.

Motivation - Problem

"Not sure why it's not working on your computer, it's working on mine."

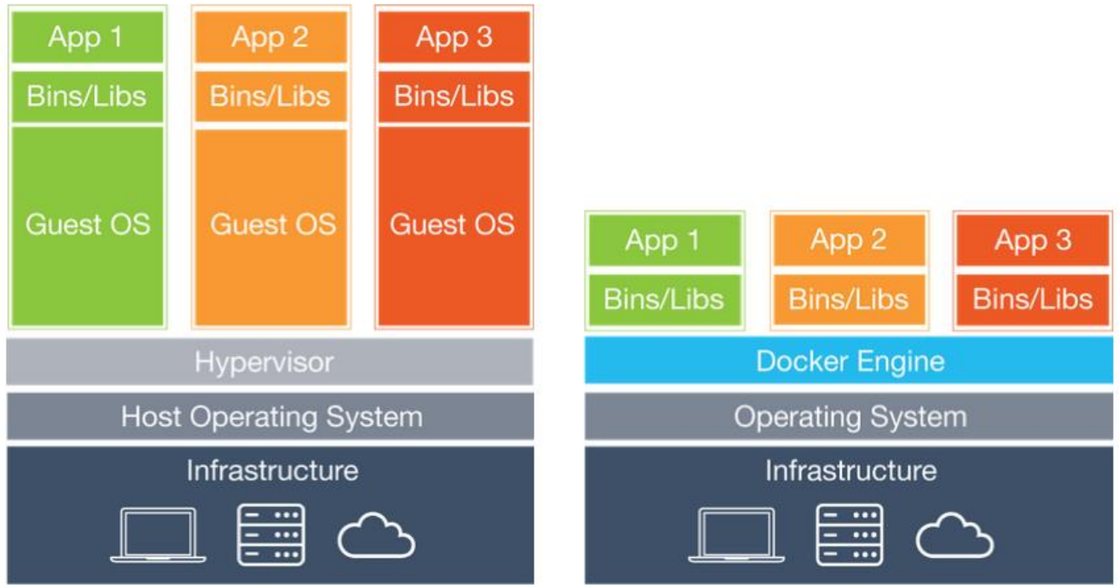
"It's a pain to install everything from scratch for Linux, Windows, and MacOS, and trying to build the same environment for each OS."

"Can't install the package that you used, can you help me out?"

"I need more compute power. I could use AWS but it'll take so long just to install all those packages and configure settings just like I have it on my machine."

What is Docker

- Create, deploy and run applications anywhere
- Containers: Package software into a standardized, self-sufficient unit with everything needed to run
  - Code
  - Runtime
  - System tools
  - Libraries
- More lightweight than VM



Virtual Machines

Docker

Why Docker

- Avoid manually setting up application environments, dependencies etc.
- Reproducible & Consistent
- Easier to share projects between different machines
- Isolation between applications running on the same machine (no conflicting dependencies)
- Automation of deployment, scaling, and management of containerized applications (with Kubernetes)

Business Value

- Infrastructure cost savings and optimizations
  - Low server utilization rates, often below 50 percent from 1 application per VM / Bare metal Server
  - Multiple containerized applications on a single VM / Bare metal server -> Better server utilization
- Developer productivity
  - No need for manually setting up environments
- IT Operations efficiency
  - Faster deployment (less testing)

Docker for Data Science

- Automate, share and **reproduce** research code / experiments
- Create easy-to-use Data Science sandboxes
- Use many out-of-the-box Data Science environments freely available (Jupiter notebooks, TensorFlow etc.)
- Package and deploy Data Science applications / Machine Learning APIs that serve predictions
- Large scale data analysis and machine learning in cloud environments

Use Case:
Containerization of Automatic Sentiment Analysis (ASA) using Docker and Anaconda (Miniconda)

Anaconda (Miniconda)

- “Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing, that aims to simplify package management and deployment. “
- Basically, it creates virtual environments with a specific python versions and python packages that isolate project dependencies.
- Anaconda comes with a ton of pre-installed scientific packages installed. Miniconda is a more lightweight distribution.

Use Case: Containerization of Automatic Sentiment Analysis (ASA) using Docker and Anaconda (Miniconda)

In the same directory there should be the following:

- Sentiment Analysis Source Code
  - Pandas – Data handling / analysis
  - Numpy – Mathematical functions on arrays
  - Sci-kit learn – Machine Learning / Sentiment Classification
- environment.yml – The blueprint of the Anaconda virtual environment
  - Python 3.6.5
  - Project Dependencies
- Dockerfile – The set of rules that define the docker image
  - Ubuntu 16.04
  - Miniconda
  - Copy source code to the container
  - Create the conda virtual environment

Use Case: Containerization of Automatic Sentiment Analysis (ASA) using Docker and Anaconda (Miniconda)

environment.yml

```yaml
name: sentiment-analysis-environment
channels:
  - defaults
dependencies:
  - pip=10.0.1=py36_0
  - python=3.6
  - pip:
    - numpy==1.15.1
    - pandas==0.23.4
    - scikit-learn==0.19.1
```

Use Case: Containerization of Automatic Sentiment Analysis (ASA) using Docker and Anaconda (Miniconda)

Dockerfile

```
1   FROM ubuntu:16.04
2
3   LABEL maintainer = "c.charalampous@impactechs.com"
4
5   # Updating Ubuntu packages
6   RUN apt-get update && yes|apt-get upgrade
7
8   # Adding wget and bzip2
9   RUN apt-get install -y wget bzip2
10
11  # Miniconda installing
12  RUN wget --quiet https://repo.continuum.io/miniconda/Miniconda3-latest-Linux-x86_64.sh -O ~/miniconda.sh && \
13      /bin/bash ~/miniconda.sh -b -p /opt/conda && \
14      rm ~/miniconda.sh && \
15      /opt/conda/bin/conda clean -tipsy && \
16      ln -s /opt/conda/etc/profile.d/conda.sh /etc/profile.d/conda.sh && \
17      echo ". /opt/conda/etc/profile.d/conda.sh" >> ~/.bashrc && \
18      echo "conda activate base" >> ~/.bashrc
19
20  # Set path to conda
21  ENV PATH /opt/conda/bin:$PATH
22
23  # Updating Conda packages
24  RUN conda update conda
25  RUN conda update --all
26
27  RUN mkdir /usr/src/sentiment_analysis
28
29  WORKDIR /usr/src/sentiment_analysis
30
31  # Copying the sentiment analysis source code to the container
32  COPY . .
33
34  # Creating the conda environment
35  RUN conda env create -f environment.yml
```

Use Case: Containerization of Automatic Sentiment Analysis (ASA) using Docker and Anaconda (Miniconda)

root@devapp-ai01: ~/git/AI_automatic_snetiment_analysis

```
root@devapp-ai01:~/git/AI_automatic_snetiment_analysis#
root@devapp-ai01:~/git/AI_automatic_snetiment_analysis# ls
Dockerfile  environment.yml  Exception.py  files  Log.py  MongoManager.py  README.md  Recording.py  SentimentAnalyzer.py  SentimentTester.py
root@devapp-ai01:~/git/AI_automatic_snetiment_analysis# docker build -t ccharalampous/sentiment-analysis:latest .
Sending build context to Docker daemon 1.325 MB
Step 1/10 : FROM ubuntu
 ---> ea4c82dcd15a
Step 2/10 : LABEL maintainer = "c.charalampous@impactechs.com"
 ---> Using cache
 ---> 22fb8364076a
Step 3/10 : RUN apt-get update && yes|apt-get upgrade && apt-get install -y wget bzip2
 ---> Using cache
 ---> 0290d73ac5e4
Step 4/10 : RUN wget --quiet https://repo.continuum.io/miniconda/Miniconda3-latest-Linux-x86_64.sh -O ~/miniconda.sh &&     /bin/bash ~/minic
da/etc/profile.d/conda.sh /etc/profile.d/conda.sh &&     echo ". /opt/conda/etc/profile.d/conda.sh" >> ~/.bashrc &&     echo "conda activate
 ---> Using cache
 ---> 937a377d64a1
Step 5/10 : ENV PATH /opt/conda/bin:$PATH
```

```
root@devapp-ai01:~/git/AI_automatic_snetiment_analysis# docker image ls
REPOSITORY                          TAG                 IMAGE ID            CREATED             SIZE
ccharalampous/sentiment-analysis    latest              02016bac2c3c        3 minutes ago       1.57 GB
```

```
root@devapp-ai01:~/git/AI_automatic_snetiment_analysis# docker run -it 02016bac2c3c bash
(base) root@ccdcf2ee3835:/# cd sentiment_analysis/
(base) root@ccdcf2ee3835:/sentiment_analysis# ls
Dockerfile  Exception.py  Log.py  MongoManager.py  README.md  Recording.py  SentimentAnalyzer.py  SentimentTester.py  environment.yml
```

```
(base) root@ccdcf2ee3835:/sentiment_analysis# python --version
Python 3.7.1
(base) root@ccdcf2ee3835:/sentiment_analysis# conda activate sentiment-analysis-environment
(sentiment-analysis-environment) root@ccdcf2ee3835:/sentiment_analysis# python --version
Python 3.6.5 :: Anaconda, Inc.
```

Final thoughts: First impressions as a beginner in Docker

- Fairly easy to get started with simple scenarios
- A lot easier to write a Dockerfile once than to replicate an environment manually each time
- Can get a bit more complicated with more complex scenarios (containers interacting with other containers etc.), but still, you write the configurations only once and that's it
- A lot of material online for almost any use case
- Great way to guarantee machine learning prediction **reproducibility**

Further Reading

- Docker Documentation
  - https://docs.docker.com/
- Docker Tutorials/Courses
  - https://go.digitalocean.com/containers-and-microservices.html
  - https://classroom.udacity.com/courses/ud615
- Docker Business Value
  - https://goto.docker.com/rs/929-FJL-178/images/WP_BusinessValueofDocker_06.26.2017.pdf
- Docker for Data Science
  - https://towardsdatascience.com/docker-for-data-science-9c0ce73e8263
  - https://blogs.technet.microsoft.com/machinelearning/2018/03/15/demystifying-docker-for-data-scientists-a-docker-tutorial-for-your-deep-learning-projects/