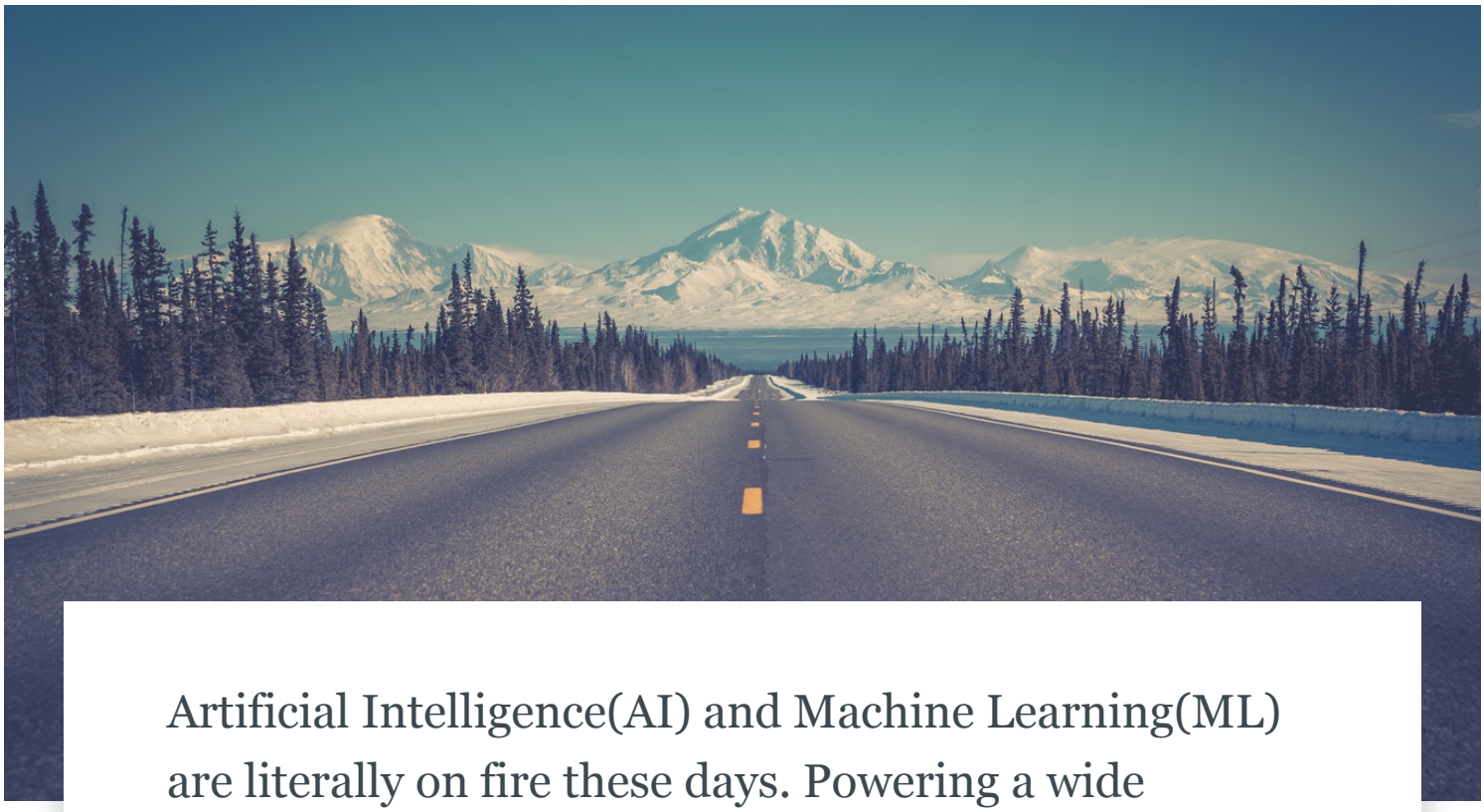
25 JANUARY 2019 / DOCKER,DATA SCIENCE

Building Python Data Science Container using Docker



Artificial Intelligence(AI) and Machine Learning(ML) are literally on fire these days. Powering a wide spectrum of use-cases ranging from self-driving cars to drug discovery and to God knows what. AI and ML have a bright and thriving future ahead of them.

On the other hand, Docker revolutionized the computing world through the introduction of ephemeral lightweight containers. Containers basically package all the software required to run inside

layer to persist the data. Enough talk let's get started with building a Python data science container.

Python Data Science Packages

Our Python data science container makes use of the following super cool python packages:

1. **NumPy**: NumPy or Numeric Python supports large, multi-dimensional arrays and matrices. It provides fast precompiled functions for mathematical and numerical routines. In addition, NumPy optimizes Python programming with powerful data structures for efficient computation of multi-dimensional arrays and matrices.
2. **SciPy**: SciPy provides useful functions for regression, minimization, Fourier-transformation, and many more. Based on NumPy, SciPy extends its capabilities. SciPy's main data structure is again a multidimensional array, implemented by Numpy. The package contains tools that help with solving linear algebra, probability theory, integral calculus, and many more tasks.
3. **Pandas**: Pandas offer versatile and powerful tools for manipulating data structures and performing extensive data analysis. It works well with incomplete, unstructured, and unordered real-world data — and comes with tools for shaping, aggregating, analyzing, and visualizing datasets.
4. **SciKit-Learn**: Scikit-learn is a Python module integrating a wide range of state-of-the-art machine learning algorithms for medium-scale supervised and unsupervised problems. It

is one of the best-known machine-learning libraries for python. The Scikit-learn package focuses on bringing machine learning to non-specialists using a general-purpose high-level language. The primary emphasis is upon ease of use, performance, documentation, and API consistency. With minimal dependencies and easy distribution under the simplified BSD license, SciKit-Learn is widely used in academic and commercial settings. Scikit-learn exposes a concise and consistent interface to the common machine learning algorithms, making it simple to bring ML into production systems.

5. **Matplotlib**: Matplotlib is a Python 2D plotting library, capable of producing publication quality figures in a wide variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shell, the Jupyter notebook, web application servers, and four graphical user interface toolkits.
6. **NLTK**: NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning.

Building the Data Science Container

Python is fast becoming the go-to language for data scientists and for this reason we are going to use Python as the language of choice for building our data science container.

The Base Alpine Linux Image

Alpine Linux is a tiny Linux distribution designed for power users who appreciate security, simplicity and resource efficiency.

As claimed by Alpine:

> *Small. Simple. Secure. Alpine Linux is a security-oriented, lightweight Linux distribution based on musl libc and busybox.*

The Alpine image is surprisingly tiny with a size of no more than 8MB for containers. With minimal packages installed to reduce the attack surface on the underlying container. This makes Alpine an image of choice for our data science container.

Downloading and Running an Alpine Linux container is as simple as:

```
$ docker container run --rm alpine:latest cat /etc/os-release
```

In our, Dockerfile we can simply use the Alpine base image as:

```
FROM alpine:latest
```

Talk is cheap let's build the Dockerfile

Now let's work our way through the Dockerfile.

```
FROM alpine:latest

LABEL MAINTAINER="Faizan Bashir <faizan.ibn.bashir@gmail.com>"

# Linking of locale.h as xlocale.h
# This is done to ensure successfull install of python numpy package
# see https://forum.alpinelinux.org/comment/690#comment-690 for more info

WORKDIR /var/www/

# SOFTWARE PACKAGES
#   * musl: standard C library
#   * lib6-compat: compatibility libraries for glibc
#   * linux-headers: commonly needed, and an unusual package name from Al
#   * build-base: used so we include the basic development packages (gcc)
#   * bash: so we can access /bin/bash
#   * git: to ease up clones of repos
#   * ca-certificates: for SSL verification during Pip and easy_install
#   * freetype: library used to render text onto bitmaps, and provides su
#   * libgfortran: contains a Fortran shared library, needed to run Fortr
#   * libgcc: contains shared code that would be inefficient to duplicate
#   * libstdc++: The GNU Standard C++ Library. This package contains an a
#   * openblas: open source implementation of the BLAS(Basic Linear Algeb
#   * tcl: scripting language
#   * tk: GUI toolkit for the Tcl scripting language
#   * libssl1.0: SSL shared libraries
ENV PACKAGES="\
    dumb-init \
    musl \
    libc6-compat \
    linux-headers \
    build-base \
    bash \
    git \
```

```
    ca-certificates \
    freetype \
    libgfortran \
    libgcc \
    libstdc++ \
    openblas \
    tcl \
    tk \
    libssl1.0 \
"

# PYTHON DATA SCIENCE PACKAGES
#   * numpy: support for large, multi-dimensional arrays and matrices
#   * matplotlib: plotting library for Python and its numerical mathemati
#   * scipy: library used for scientific computing and technical computin
#   * scikit-learn: machine learning library integrates with NumPy and Sc
#   * pandas: library providing high-performance, easy-to-use data struct
#   * nltk: suite of libraries and programs for symbolic and statistical
ENV PYTHON_PACKAGES="\
    numpy \
    matplotlib \
    scipy \
    scikit-learn \
    pandas \
    nltk \
"

RUN apk add --no-cache --virtual build-dependencies python --update py-pi
    && apk add --virtual build-runtime \
    build-base python-dev openblas-dev freetype-dev pkgconfig gfortran \
    && ln -s /usr/include/locale.h /usr/include/xlocale.h \
    && pip install --upgrade pip \
    && pip install --no-cache-dir $PYTHON_PACKAGES \
    && apk del build-runtime \
    && apk add --no-cache --virtual build-dependencies $PACKAGES \
    && rm -rf /var/cache/apk/*

CMD ["python"]
```

datascience-python2.7.Dockerfile hosted with ❤ by GitHub view raw

The `FROM` directive is used to set `alpine:latest` as the base image.

Using the WORKDIR directive we set the `/var/www` as the working directory for our container. The `ENV PACKAGES` lists the software packages required for our container like `git`, `blas` and `libgfortran`. The python packages for our data science container are defined in the ENV PACKAGES.

We have combined all the commands under a single Dockerfile `RUN` directive to reduce the number of layers which in turn helps in reducing the resultant image size.

Building and tagging the image

Now that we have our Dockerfile defined, navigate to the folder with the Dockerfile using the terminal and build the image using the following command:

```
$ docker build -t faizanbashir/python-datascience:2.7 -f Dockerfile
```

The `-t` flag is used to name a tag in the 'name:tag' format. The `-f` tag is used to define the name of the Dockerfile (Default is 'PATH/Dockerfile').

Running the container

We have successfully built and tagged the docker image, now we can run the container using the following command:

```
$ docker container run --rm -it faizanbashir/python-datascience:2.7
```

Voila, we are greeted by the sight of a python shell ready to perform all kinds of cool data science stuff.

```
Python 2.7.15 (default, Aug 16 2018, 14:17:09) [GCC 6.4.0] on linux
```

Our container comes with Python 2.7, but don't be sad if you wanna work with Python 3.6. Lo, behold the Dockerfile for Python 3.6:

```
1  FROM alpine:latest
2
3  LABEL MAINTAINER="Faizan Bashir <faizan.ibn.bashir@gmail.com>"
4
5  # Linking of locale.h as xlocale.h
6  # This is done to ensure successfull install of python numpy package
7  # see https://forum.alpinelinux.org/comment/690#comment-690 for more info
8
9  WORKDIR /var/www/
10
11 # SOFTWARE PACKAGES
12 #   * musl: standard C library
13 #   * lib6-compat: compatibility libraries for glibc
14 #   * linux-headers: commonly needed, and an unusual package name from Al
15 #   * build-base: used so we include the basic development packages (gcc)
16 #   * bash: so we can access /bin/bash
17 #   * git: to ease up clones of repos
18 #   * ca-certificates: for SSL verification during Pip and easy install
```

```
#   * libgfortran: contains a Fortran shared library, needed to run Fortra
#   * libgcc: contains shared code that would be inefficient to duplicate
#   * libstdc++: The GNU Standard C++ Library. This package contains an ad
#   * openblas: open source implementation of the BLAS(Basic Linear Algeb
#   * tcl: scripting language
#   * tk: GUI toolkit for the Tcl scripting language
#   * libssl1.0: SSL shared libraries
ENV PACKAGES="\
    dumb-init \
    musl \
    libc6-compat \
    linux-headers \
    build-base \
    bash \
    git \
    ca-certificates \
    freetype \
    libgfortran \
    libgcc \
    libstdc++ \
    openblas \
    tcl \
    tk \
    libssl1.0 \
    "

# PYTHON DATA SCIENCE PACKAGES
#   * numpy: support for large, multi-dimensional arrays and matrices
#   * matplotlib: plotting library for Python and its numerical mathemati
#   * scipy: library used for scientific computing and technical computin
#   * scikit-learn: machine learning library integrates with NumPy and Sc
#   * pandas: library providing high-performance, easy-to-use data structu
#   * nltk: suite of libraries and programs for symbolic and statistical
ENV PYTHON_PACKAGES="\
    numpy \
    matplotlib \
    scipy \
    scikit-learn \
    pandas \
    nltk \
    "
```

```
63      && apk add --virtual build-runtime \
64      build-base python3-dev openblas-dev freetype-dev pkgconfig gfortran \
65      && ln -s /usr/include/locale.h /usr/include/xlocale.h \
66      && python3 -m ensurepip \
67      && rm -r /usr/lib/python*/ensurepip \
68      && pip3 install --upgrade pip setuptools \
69      && ln -sf /usr/bin/python3 /usr/bin/python \
70      && ln -sf pip3 /usr/bin/pip \
71      && rm -r /root/.cache \
72      && pip install --no-cache-dir $PYTHON_PACKAGES \
73      && apk del build-runtime \
74      && apk add --no-cache --virtual build-dependencies $PACKAGES \
75      && rm -rf /var/cache/apk/*
76
77  CMD ["python3"]
```

datascience-python3.6.Dockerfile hosted with ❤️ by GitHub view raw

Build and tag the image like so:

```
$ docker build -t faizanbashir/python-datascience:3.6 -f Dockerfile
```

Run the container like so:

```
$ docker container run --rm -it faizanbashir/python-datascience:3.6
```

With this, you have a ready to use container for doing all kinds of cool data science stuff.

Serving Puddin'

Figures, you have the time and resources to set up all this stuff. In case you don't, you can pull the existing images that I have already built and pushed to Docker's registry Docker Hub using:

```
# For Python 2.7 pull
$ docker pull faizanbashir/python-datascience:2.7
# For Python 3.6 pull
$ docker pull faizanbashir/python-datascience:3.6
```

After pulling the images you can use the image or extend the same in your Dockerfile file or use it as an image in your docker-compose or stack file.

Aftermath

The world of AI, ML is getting pretty exciting these days and will continue to become even more exciting. Big players are investing heavily in these domains. About time you start to harness the power of data, who knows it might lead to something wonderful.

You can check out the code here.

faizanbashir/python-datascience

I hope this article helped in building containers for your data science projects.

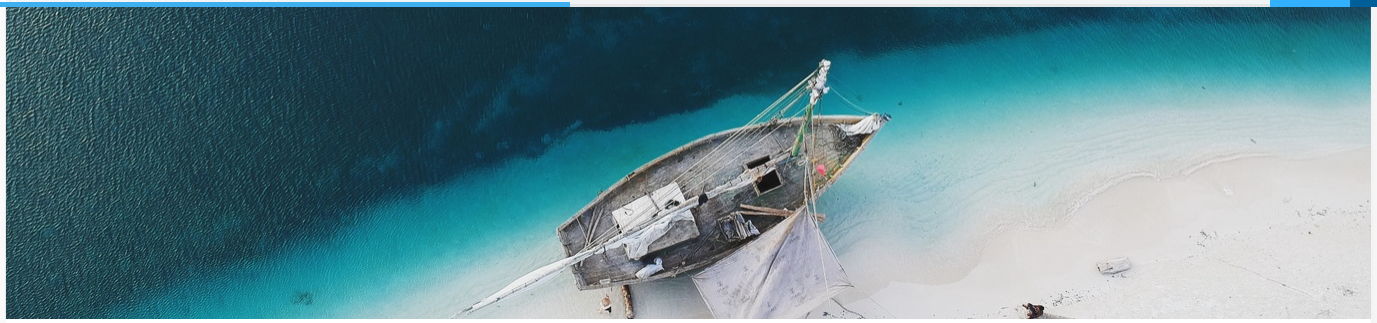
Faizan Bashir

Principal Engineer | Architecting and building distributed applications in the Cloud | Adventurer | Seeker | Wanderer

KUBERNETES

Adding limited access IAM user to EKS Cluster

FAIZAN BASHIR 1 MIN READ



SERVERLESS

Building Serverless Contact Form For Static Websites

FAIZAN BASHIR 1 MIN READ