Microservices & Docker
for Data Science

Date : 10-10-2020 | Speaker : Ayon Roy | Event : ML4E Session, NIT Rourkela

Visit - AYONROY.ML

Hello **Buddy!**

I am **Ayon Roy**

B.Tech CSE (2017-2021)

Data Science Intern @ **Lulu International Exchange**, Abu Dhabi

(**World's Leading Financial Services Company**)

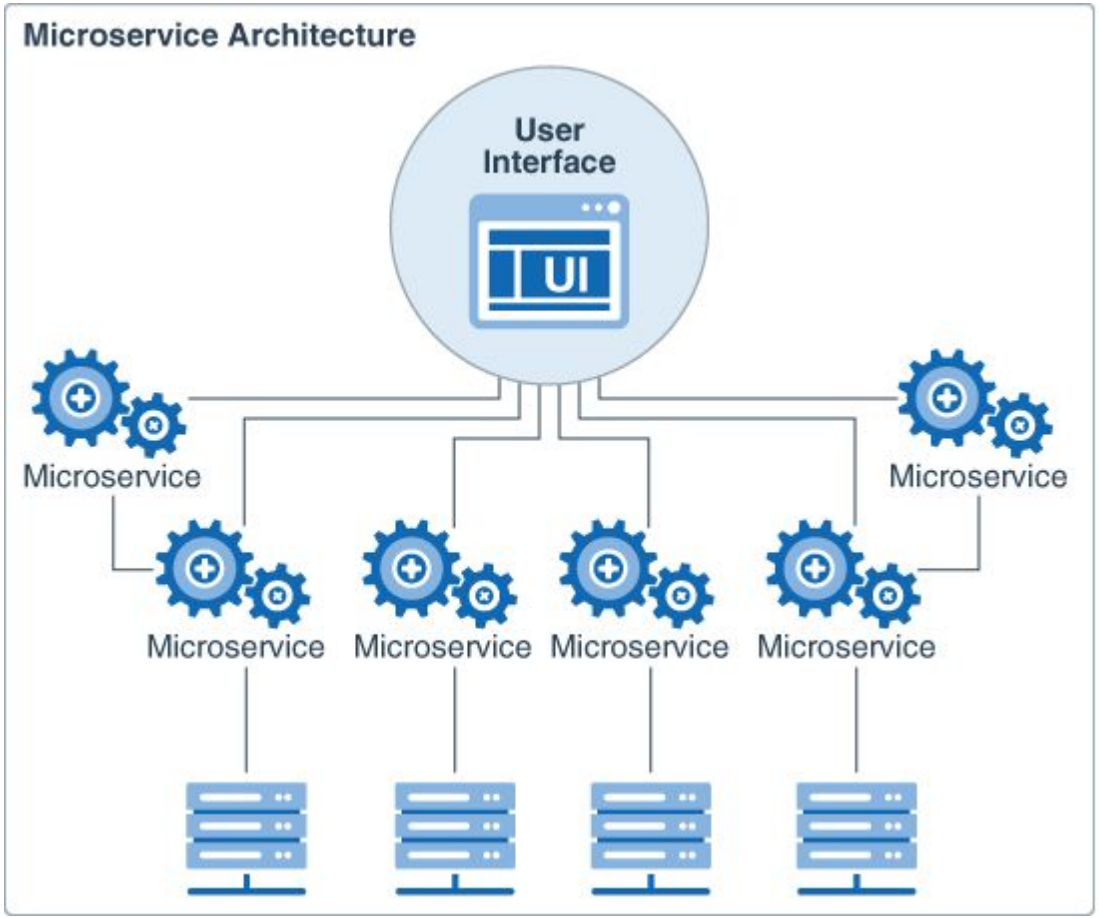
Brought **Kaggle Days Meetup** Community in India for the 1st time

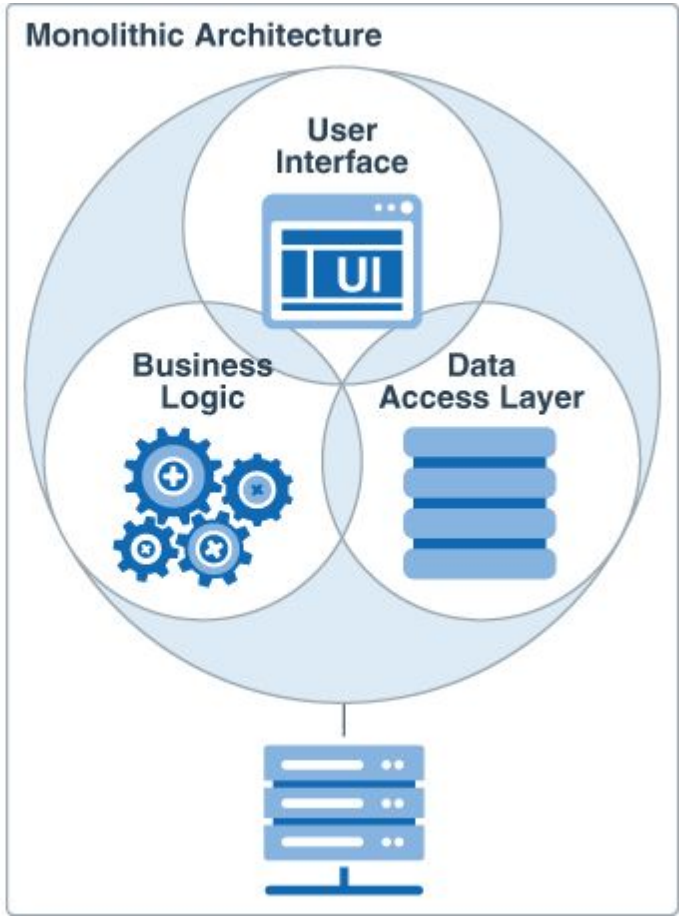
If you haven't heard about me yet, you might have been living under the rocks. Wake up !!

Agenda (10-10-2020)

- What is Microservices?
- What is Docker?
- Key components of a Data Science Process
- Where Microservices & Docker fits in a Data Science process?
- Using Microservices for Data Science
- Using Docker for Data Science

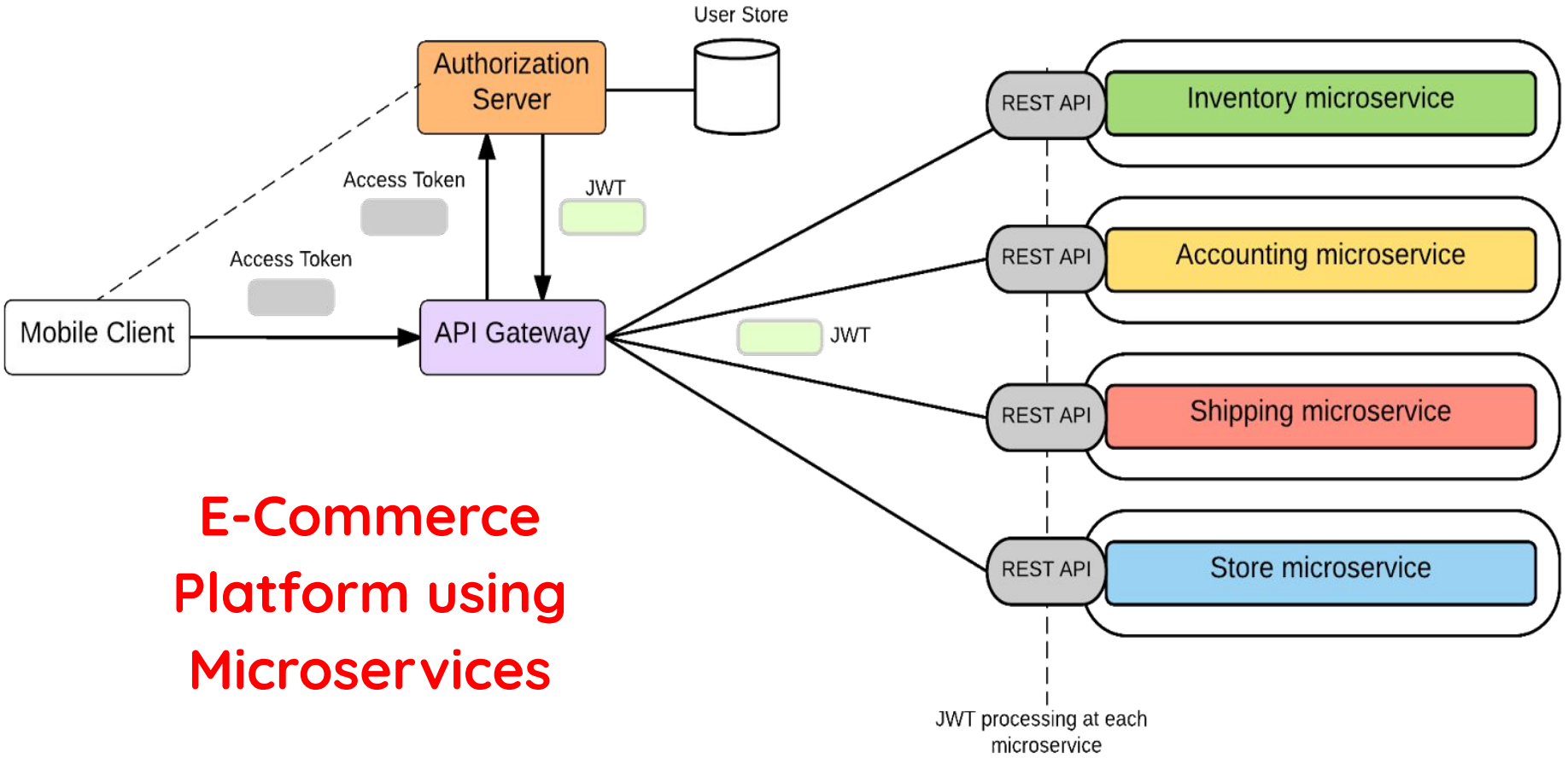
What is Microservices ?





Microservices - also known as the microservice architecture - is an architectural style that structures an application as a collection of services that are :

- Highly maintainable and testable
- Loosely coupled
- Independently deployable
- Organized around business capabilities
- Owned by a small team



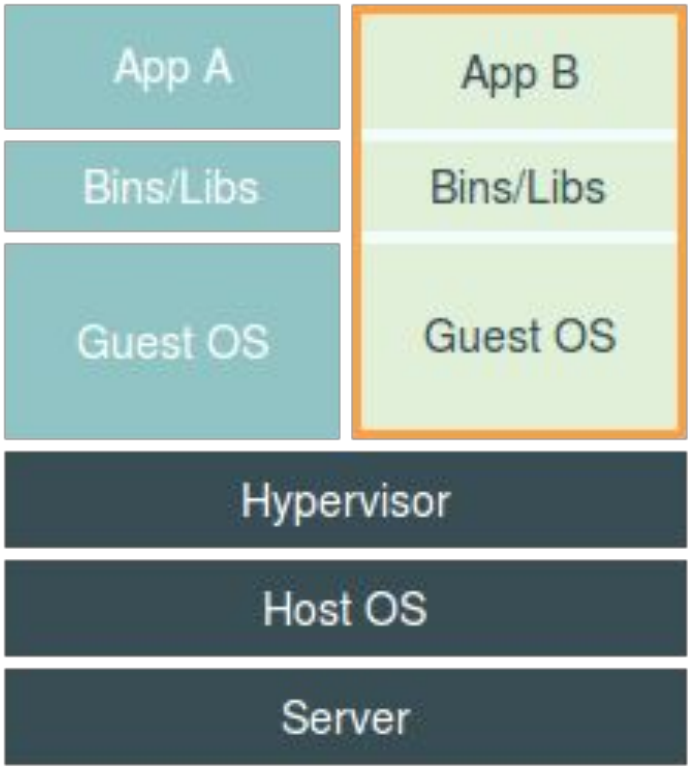
E-Commerce Platform using Microservices

Visit - AYONROY.ML

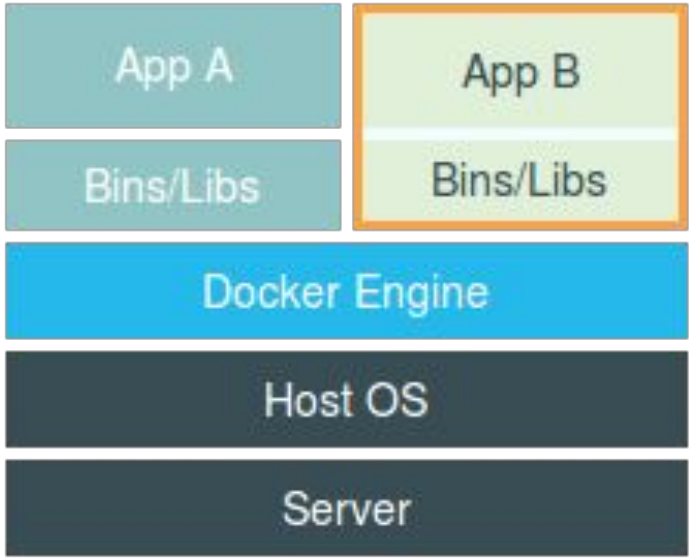
What is Docker ?

Docker is an open-source project for automating the deployment of applications as portable, self-sufficient containers that can run anywhere on the cloud or on-premises.

Virtual Machine



Docker



Virtual Machines	Docker
Each VM runs its own OS	All containers share the same Kernel of the host
Boot up time is in minutes	Containers instantiate in seconds
VMs snapshots are used sparingly	Images are built incrementally on top of another like layers. Lots of images/snapshots
Not effective diffs. Not version controlled	Images can be diffed and can be version controlled. Dockerhub is like GITHUB
Cannot run more than couple of VMs on an average laptop	Can run many Docker containers in a laptop.
Only one VM can be started from one set of VMX and VMDK files	Multiple Docker containers can be started from one Docker image

How Docker starts running?

Dockerized App (microservice)

Dockerfile

Each Docker container starts with a *Dockerfile*. **A Dockerfile is a text file written that includes the instructions to build a Docker *image*.** It specifies the operating system that will underlie the container, along with the languages, environmental variables, file locations, network ports, and other components it needs—and, of course, what the container will actually be doing once we run it.

Docker image

Once you have your Dockerfile written, you invoke the **Docker build** utility to create an *image* based on that Dockerfile. Whereas the Dockerfile is the set of instructions that tells build how to make the image, a **Docker image is a portable file containing the specifications for which software components the container will run and how.**

Docker run

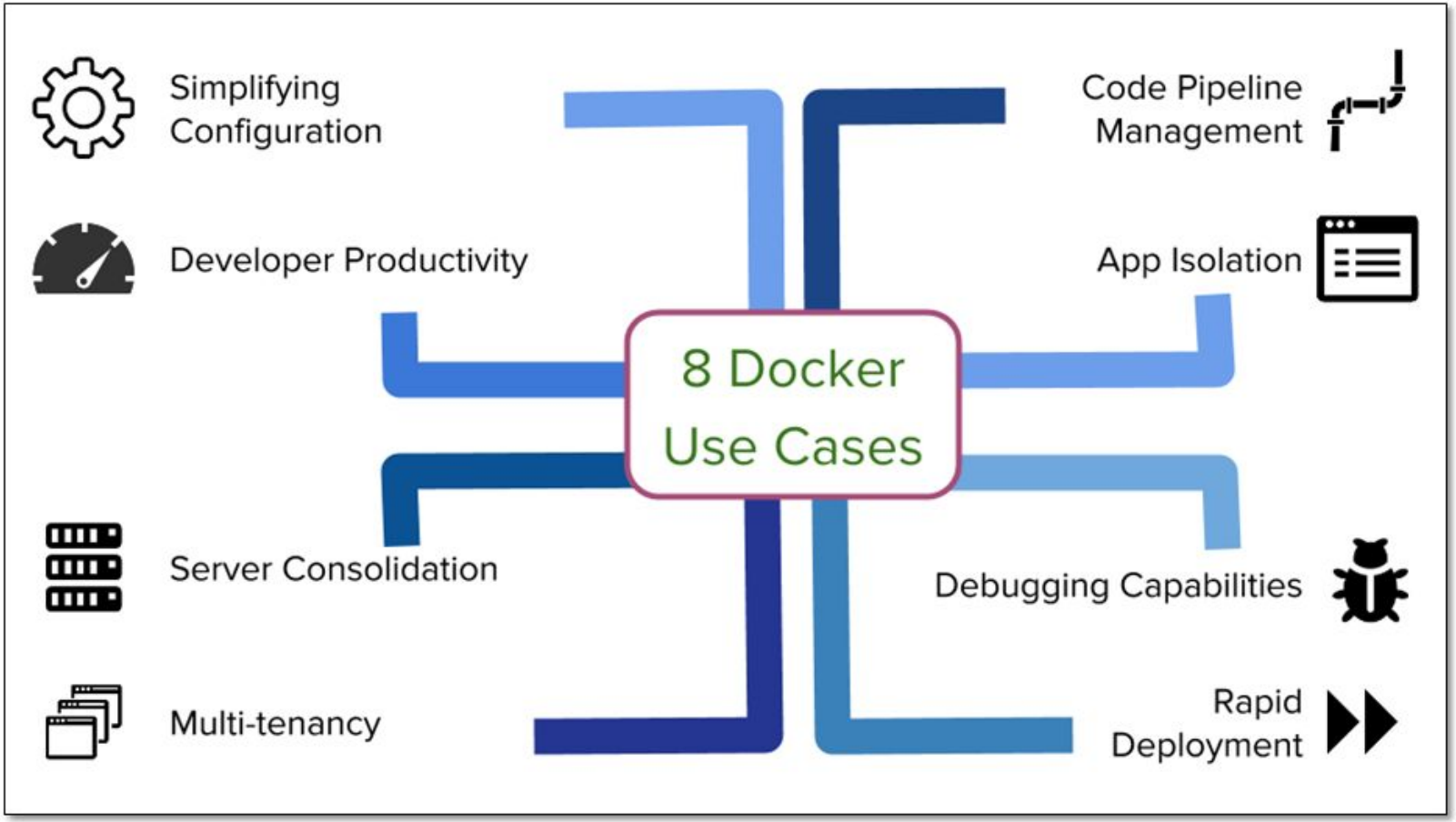
Docker run utility is the command that actually launches a container. Each container is an *instance* of an image. Containers are designed to be transient and temporary, but they can be stopped and restarted.

Docker Hub

While building containers is easy, it's not easy to get the idea to build each and every one of your images from scratch. **Docker Hub is a SaaS repository for sharing and managing containers,** where you will find official Docker images from open-source projects and software vendors and unofficial images from the general public.

Docker Daemon

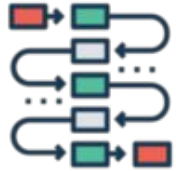
The Docker daemon is a service that runs on your host operating system. The Docker daemon itself exposes a REST API. From here, a number of different tools can talk to the daemon through this API.



Key Components of Data Science Process







OBTAIN | SCRUB | EXPLORE | MODEL | INTERPRET

O Gather data from relevant sources

S Clean data to formats that machine understands

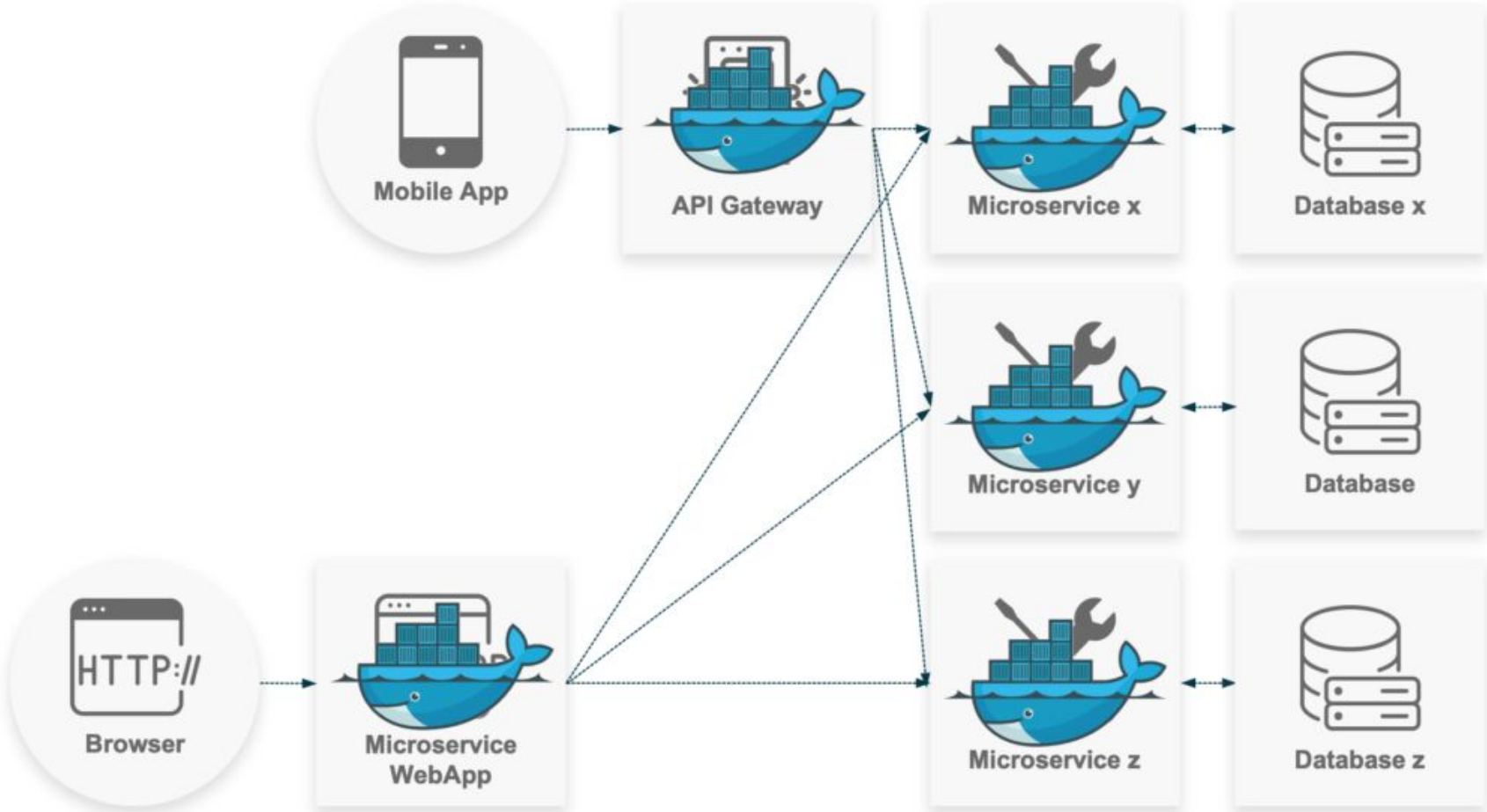
E Find significant patterns and trends using statistical methods

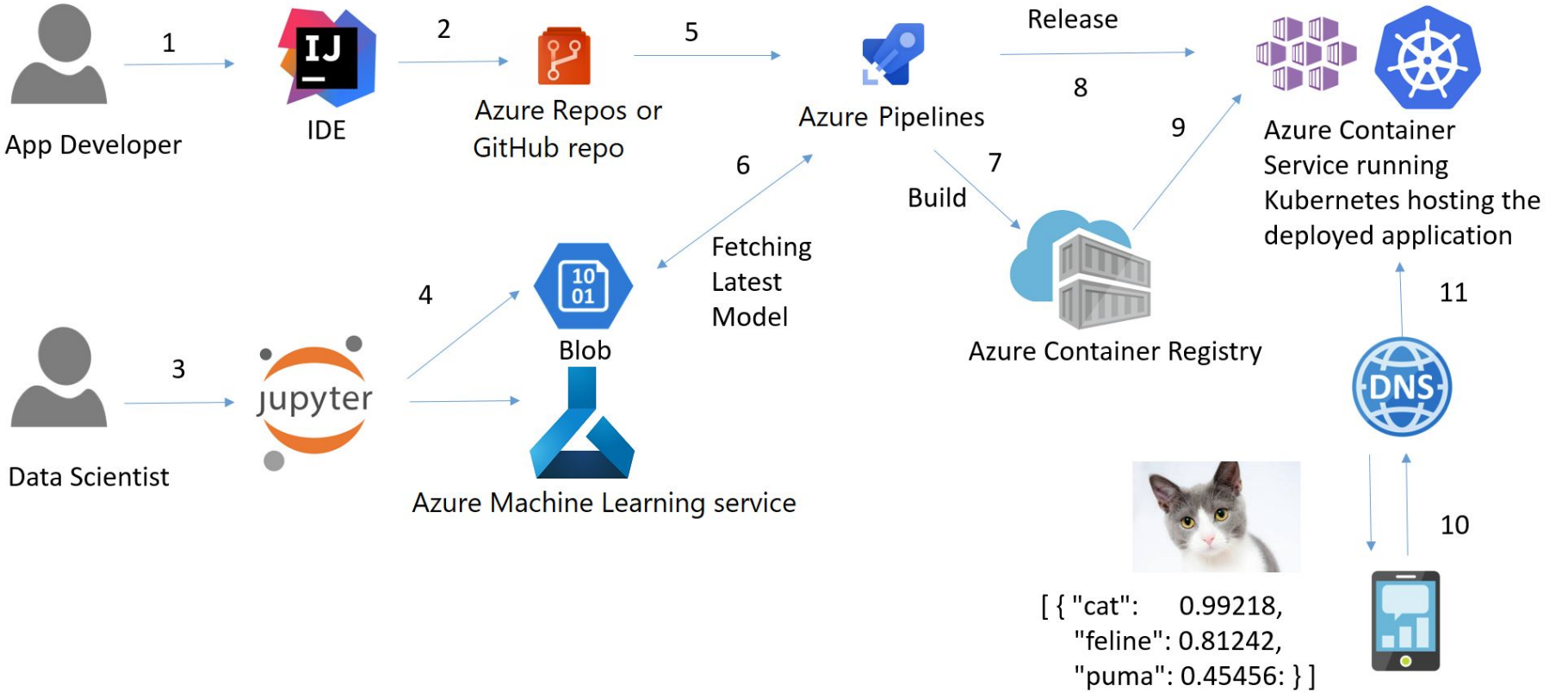
M Construct models to predict and forecast

N Put the results into good use

Originally by Hilary Mason and Chris Wiggins

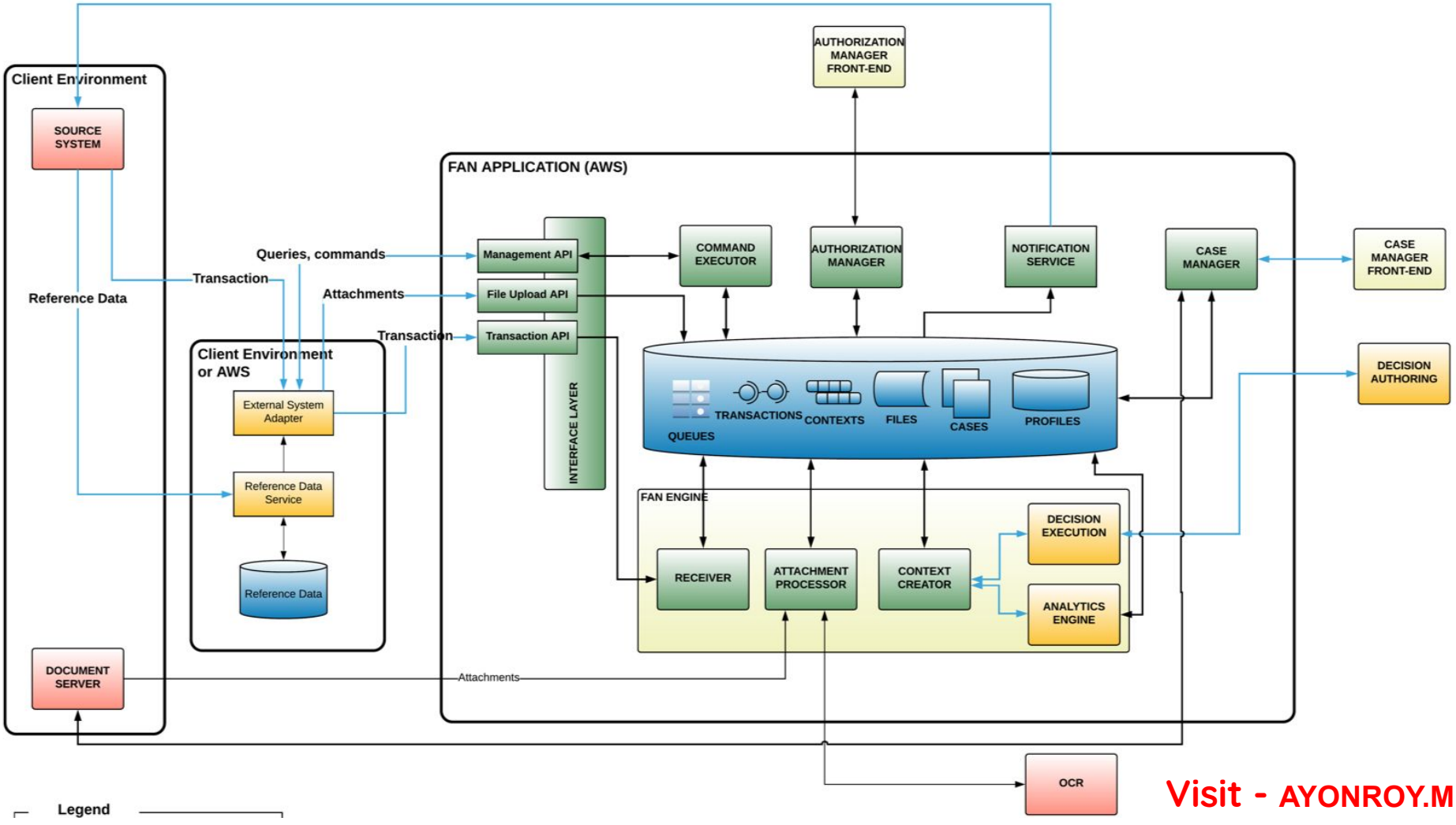
Where Microservices & Docker fits in a Data Science Process?



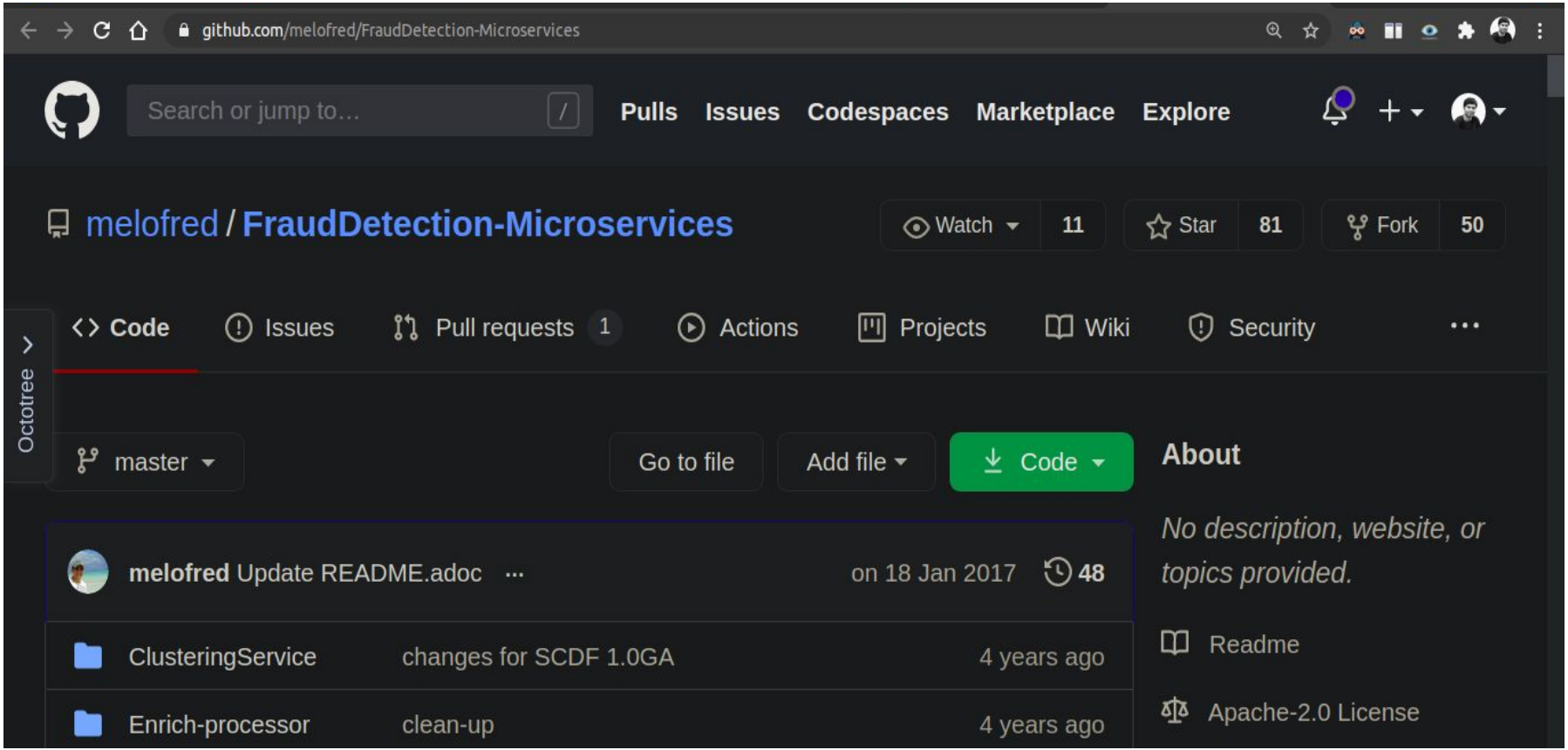


Using Microservices for Data Science

Visit - AYONROY.ML



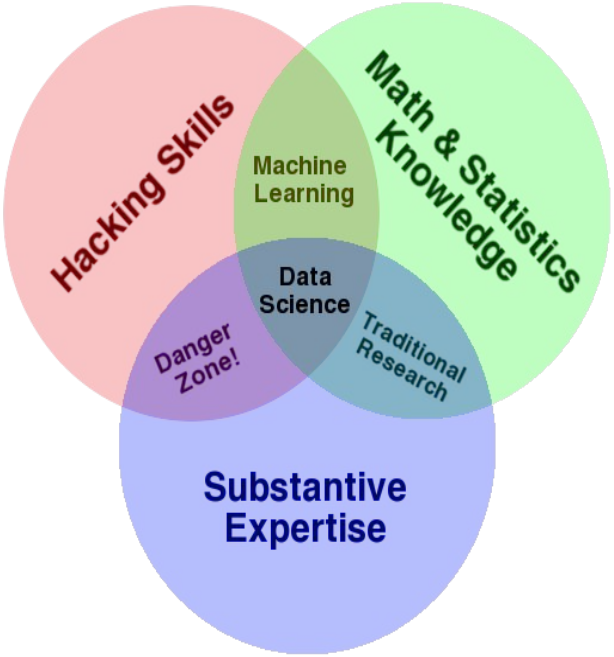
- Each transaction arrives in a **receiver service**, which places it into a queue.
- An **attachment processor service** checks for an attachment; if one exists, it sends it to an OCR service and stores the transaction enriched with the OCR data.
- A **context creator service** analyzes it and associates it with any past transactions that are related to it.
- A **decision execution engine** runs the rules that have been set up by the client and identifies violations.
- One or more analytics engines review transactions and flag outliers.
- Now decorated with a score, the transaction goes to a **case manager service**, which decides whether to create a case for human follow-up based on any identified issues.
- At the same time, a notification manager passes updates on the processing of each transaction back to the client's expense/procurement system.



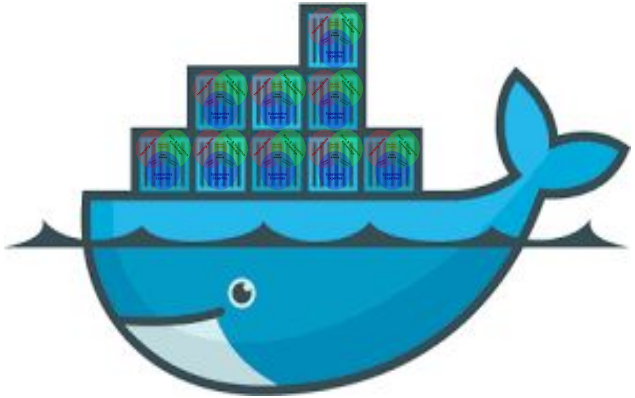
https://github.com/melofred/FraudDetection-Microservices

Using
Docker
for Data Science

Visit - AYONROY.ML



+



Here are a few examples of applications relevant to data science where you might try out with Docker:

- ***Create an ultra-portable, custom development workflow:*** Build a personal development environment in a Dockerfile, so you can access your workflow immediately on any machine with Docker installed.
- ***Create development, testing, staging, and production environments***: Your code will run as you expect and become able to create staging environments identical to production so you know when you push to production, you're going to be OK.
- ***Reproduce your Jupyter notebook on any machine***: Create a container that runs everything you need for your Jupyter Notebook data analysis, so you can pass it along to other researchers / colleagues and know that it will run on their machine.

Self-Contained Container

- **Problem**: Sharing results (Jupyter notebook)
- **Workflow**:
  - Create Docker image with libraries, data and notebook

Self-Contained Container: Dockerfile

```
FROM python:3.6.3-slim
LABEL maintainer="Ayon Roy <ayon.roy2000@gmail.com>"
WORKDIR /app
COPY . /app
RUN pip --no-cache-dir install numpy pandas seaborn sklearn jupyter
EXPOSE 8888

# Run app.py when the container launches
CMD ["jupyter", "notebook", "--ip='*'", "--port=8888", "--no-browser", "--allow-root"]
```

Data Driven App: Dashboard

- Data stored on local machine
- Create & package dashboard inside container
  - Dash Tutorial
- Container is an executable on top of data
  - Start container to view dashboard

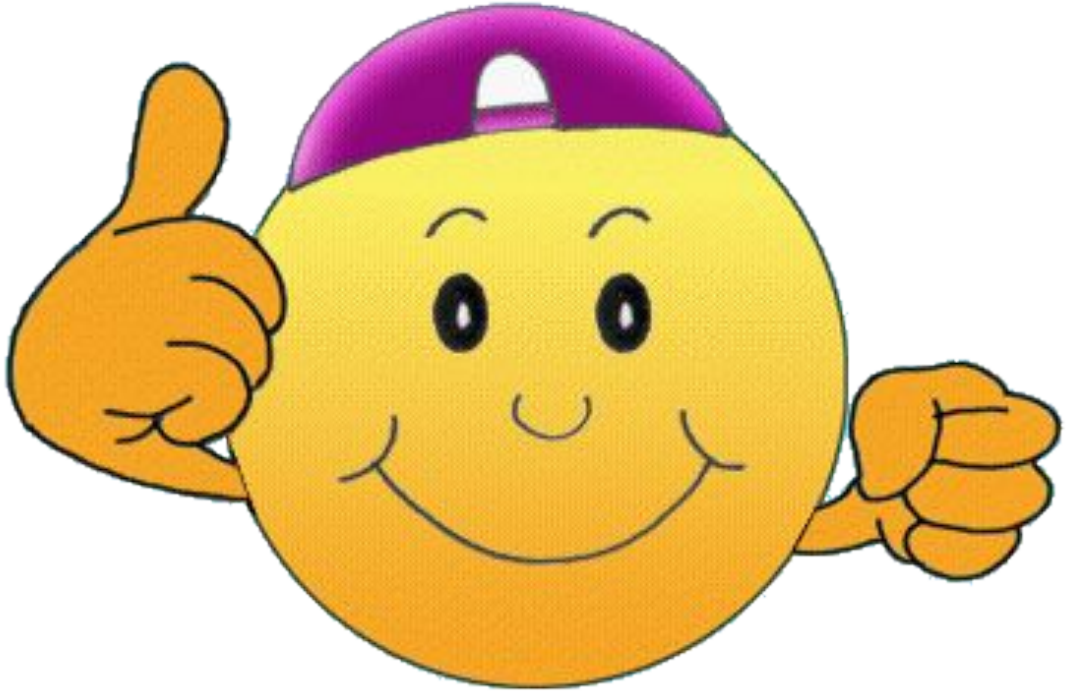
Data Driven App: Dockerfile

```
FROM python:3.6.3-alpine3.6

LABEL maintainer="ayon.roy2000@gmail.com"

WORKDIR /app

COPY . /app

RUN pip --no-cache-dir install -r /app/requirements.txt

EXPOSE 8050

VOLUME /app/data

ENTRYPOINT ["python"]

CMD ["plot_timeseries.py"]
```

A few useful resources

- https://docs.microsoft.com/en-us/dotnet/architecture/microservices/container-docker-introduction/docker-defined
- https://github.com/docker-for-data-science/docker-for-data-science-tutorial
- https://docs.docker.com/get-started/overview/
- https://subscription.packtpub.com/book/web_development/9781838823818
- https://unsupervisedpandas.com/data-science/docker-for-data-science/
- Docker for Data Scientists, Strata 2016, Michaelangelo D'Agostino (YouTube Video)
- Data Science Workflows Using Containers, by Aly Sivji (YouTube Video)
- A 3 Hour Docker for Data Scientists Workshop (YouTube Video)

GO FOR IT !



GOOD LUCK !

Let me answer your Questions now.

Finally, it’s your time to speak.

Danke Schoen

Questions ? Any Feedbacks ? Did you like the talk?
Tell me about it.

If you think I can help you,
connect with me via

Email : ayon.roy2000@gmail.com

LinkedIn / Github / Telegram Username : ayonroy2000

Website : https://AYONROY.ML/