FFHS

FFHS CAS BIGDATA 2019

BIGDATA WITH DOCKER

PHILIPP DUBACH

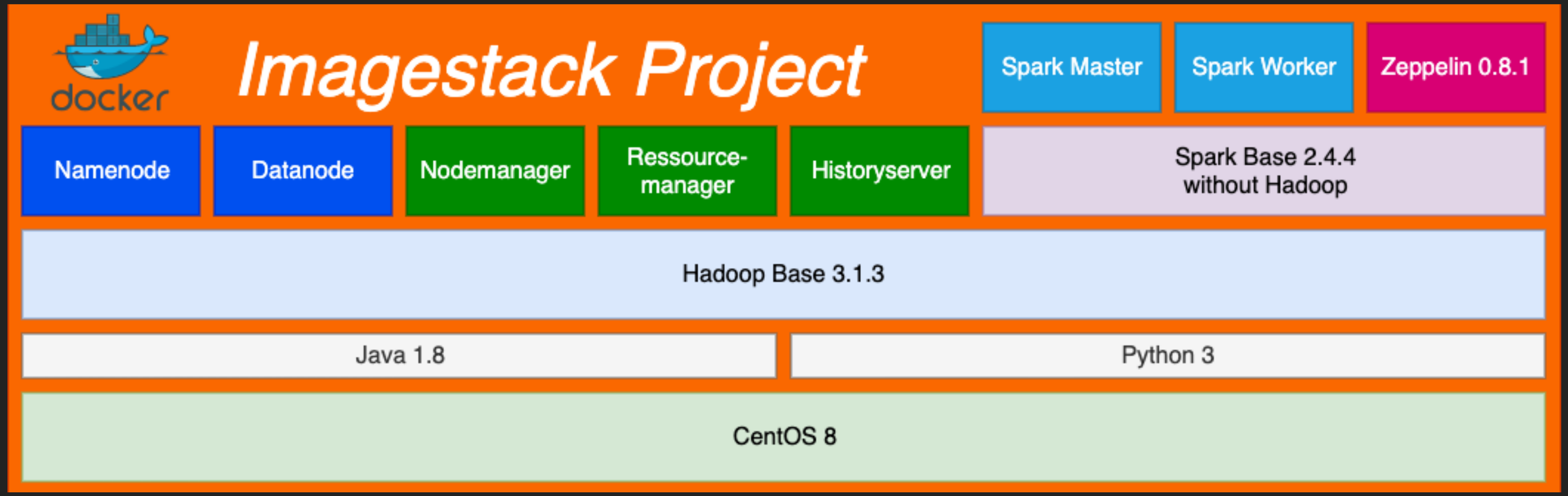
INTRODUCTION
CONCEPT
SHOW SOME CODE
CLONE AND BUILD
UP AND RUNNING
WORK WITH THE ENVIRONMENT
SUMMARY AND WHAT'S NEXT

CONTENT

INTRODUCTION

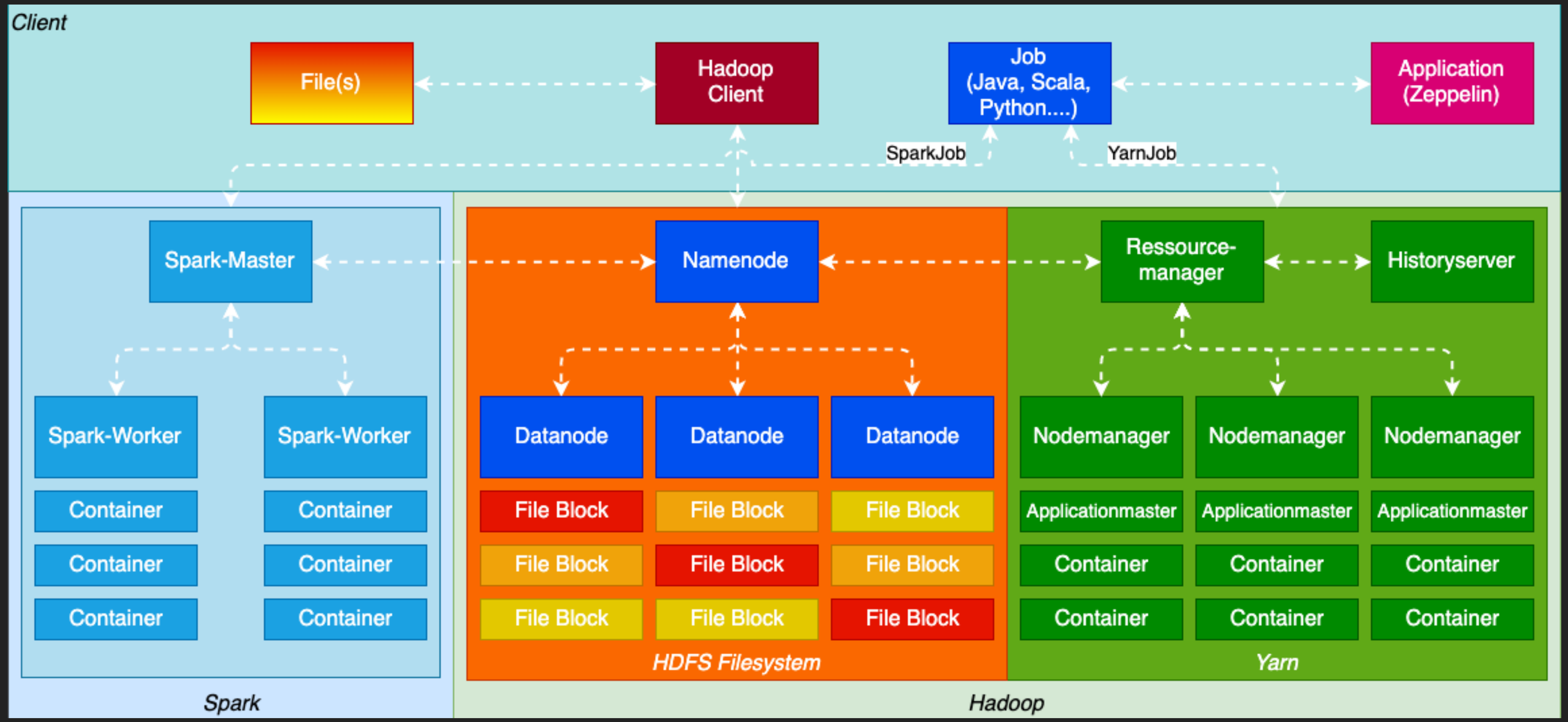
- Docker
- Infrastructure as Code
- Clustering / Scale Horizontally
- Understand Big Data Infrastructure
- Training / Education
- Fast Deployment of Application
- Out of interest and because it is fun....

CONCEPT



Environment: CentOS 8 / Hadoop 3.1.3 / Spark 2.4.4 / Zeppelin 0.8.1

CONCEPT


Client
File(s)
Hadoop Client
Job (Java, Scala, Python....)
Application (Zeppelin)
SparkJob
YarnJob
Spark-Master
Spark-Worker
Spark-Worker
Container
Container
Container
Container
Container
Container
Spark
Namenode
Datanode
Datanode
Datanode
File Block
File Block
File Block
File Block
File Block
File Block
File Block
File Block
File Block
HDFS Filesystem
Ressource-manager
Historyserver
Nodemanager
Nodemanager
Nodemanager
Applicationmaster
Applicationmaster
Applicationmaster
Container
Container
Container
Container
Container
Container
Yarn
Hadoop

SHOW SOME CODE > HTTPS://GITHUB.COM/DUBACHPHIL/BIGDATA-DOCKER

```
FROM centos:8
MAINTAINER Philipp Dubach <dubachphil@hotmail.com>
RUN dnf update -y
RUN dnf install curl java-1.8.0-openjdk perl nmap-ncat java-1.8.0-openjdk-devel.x86_64 -y
RUN dnf install glibc-langpack-en nano python3 python3-pip -y
RUN pip3 install pyspark pandas findspark
ENV JAVA_HOME=/usr/lib/jvm/jre-1.8.0-openjdk/
ENV HADOOP_VERSION 3.1.3
ENV HADOOP_URL https://www.apache.org/dist/hadoop/common/hadoop-$HADOOP_VERSION/hadoop-$HADOOP_VERSION.tar.gz
RUN set -x curl -fSL "$HADOOP_URL" -o /tmp/hadoop.tar.gz \
    && curl -fSL "$HADOOP_URL.asc" -o /tmp/hadoop.tar.gz.asc \
    && tar -xvf /tmp/hadoop.tar.gz -C /opt/ \
    && rm -f /tmp/hadoop.tar.gz*
RUN ln -s /opt/hadoop-$HADOOP_VERSION/etc/hadoop /etc/hadoop
RUN echo "alias ll='ls -la'" >> /etc/bash.bashrc
RUN echo "alias python='/usr/bin/python3'" >> /etc/bash.bashrc
RUN mkdir /opt/hadoop-$HADOOP_VERSION/logs
RUN mkdir /hadoop-data
ENV HADOOP_HOME=/opt/hadoop-$HADOOP_VERSION
ENV HADOOP_CONF_DIR=/etc/hadoop
ENV MULTIHOMED_NETWORK=1
ENV USER=root
ENV PATH $HADOOP_HOME/bin/:$PATH
ENV PATH /usr/bin:$PATH
ADD entrypoint.sh /entrypoint.sh
RUN chmod a+x /entrypoint.sh

ENTRYPOINT ["/entrypoint.sh"]
```

HADOOP_BASE IMAGE

A LOT OF VARIABLES

SHOW SOME CODE > HTTPS://GITHUB.COM/DUBACHPHIL/BIGDATA-DOCKER

HADOOP NAMENODE

```
FROM dubachphil/hadoop_base

MAINTAINER Philipp Dubach <dubachphil@hotmail.com>

HEALTHCHECK CMD curl -f http://localhost:9870/ || exit 1

ENV HDFS_CONF_dfs_namenode_name_dir=file:///hadoop/dfs/name
RUN mkdir -p /hadoop/dfs/name

VOLUME /hadoop/dfs/name

ADD run.sh /run.sh

RUN chmod a+x /run.sh

EXPOSE 9870

CMD ["/run.sh"]
```

```
#!/bin/bash
namedir=`echo $HDFS_CONF_dfs_namenode_name_dir | perl -pe 's#file://##'`

if [ ! -d $namedir ]; then
  echo "Namenode name directory not found: $namedir"
  exit 2
fi

if [ -z "$CLUSTER_NAME" ]; then
  echo "Cluster name not specified"
  exit 2
fi

if [ "`ls -A $namedir`" == "" ]; then
  echo "Formatting namenode name directory: $namedir"
  $HADOOP_HOME/bin/hdfs --config $HADOOP_CONF_DIR namenode -format $CLUSTER_NAME
fi

$HADOOP_HOME/bin/hdfs --config $HADOOP_CONF_DIR namenode
```

SHOW SOME CODE > HTTPS://GITHUB.COM/DUBACHPHIL/BIGDATA-DOCKER

DOCKER-COMPOSE FILE

```yaml
version: "3"
services:
  namenode:
    image: dubachphil/hadoop_namenode
    ports:
      - 9870:9870
    environment:
      - CLUSTER_NAME=hadoop_cluster
    env_file:
      - ./hadoop.env

  datanode:
    image: dubachphil/hadoop_datanode
    depends_on:
      - namenode
    env_file:
      - ./hadoop.env

  resourcemanager:
    image: dubachphil/hadoop_resourcemanager
    ports:
      - 8088:8088
    depends_on:
      - datanode
    env_file:
      - ./hadoop.env

  > nodemanager > historyserver > spark-master > spark-worker > zeppelin
```

A LOT OF ENVIRONMENT VARIABLES
FOR EACH SERVICE

CLONE AND BUILD

- Software you need:
  - git (https://git-scm.com/downloads)
  - docker (https://docs.docker.com/)
  - docker-compose (https://docs.docker.com/compose/install/)
- Clone or download my Repository in the Terminal
  - git clone https://github.com/dubachphil/bigdata-docker.git
  - cd bigdata-docker

CLONE AND BUILD

- If you like: Build the images (Optional)
  The latest Build is ready on dockerhub and will automatically downloaded
  - Run in the bigdata-docker folder:
    sh build_images.sh (duration above 5-20min)
  - Check if the images are build
    docker images

```
dubachphil/zeppelin                latest              43616fa4a269        4 seconds ago       6.99GB
dubachphil/spark_worker            latest              947ee10e4652        4 minutes ago       3.77GB
dubachphil/spark_master            latest              46c4eaba4298        4 minutes ago       3.77GB
dubachphil/spark_base              latest              8ea40ed35e84        4 minutes ago       3.77GB
dubachphil/hadoop_historyserver    latest              531026dbfea3        5 minutes ago       3.16GB
dubachphil/hadoop_nodemanager      latest              e5f75208c770        5 minutes ago       3.16GB
dubachphil/hadoop_resourcemanager  latest              5c1762ab2ae4        5 minutes ago       3.16GB
dubachphil/hadoop_datanode         latest              7f26b7b88d4a        5 minutes ago       3.16GB
dubachphil/hadoop_namenode         latest              f14e4e74083c        5 minutes ago       3.16GB
dubachphil/hadoop_base             latest              2fe6a1522f2a        5 minutes ago       3.16GB
```

UP AND RUNNING

- Start the bigdata environment
  - Run in the bigdata-docker folder:
    docker-compose up -d
  - Show the logs
    docker-compose logs -f
  - Scale the application
    docker-compose up -d --scale datanode=3 --scale spark-worker=3 --scale nodemanager=3

UP AND RUNNING

- Show the running containers
  docker-compose ps or docker ps

```
philippdubach@Philipps-MBP ~/bigdata-docker> docker-compose ps
              Name                            Command                  State                                        Ports
-------------------------------------------------------------------------------------------------------------------------------------------------------
bigdata-docker_datanode_1          /entrypoint.sh /run.sh           Up (healthy)   0.0.0.0:9867->9864/tcp
bigdata-docker_datanode_2          /entrypoint.sh /run.sh           Up (healthy)   0.0.0.0:9869->9864/tcp
bigdata-docker_datanode_3          /entrypoint.sh /run.sh           Up (healthy)   0.0.0.0:9868->9864/tcp
bigdata-docker_historyserver_1     /entrypoint.sh /run.sh           Up (healthy)   0.0.0.0:8188->8188/tcp
bigdata-docker_namenode_1          /entrypoint.sh /run.sh           Up (healthy)   0.0.0.0:8020->8020/tcp, 0.0.0.0:9870->9870/tcp
bigdata-docker_nodemanager_1       /entrypoint.sh /run.sh           Up (healthy)   0.0.0.0:8045->8042/tcp
bigdata-docker_nodemanager_2       /entrypoint.sh /run.sh           Up (healthy)   0.0.0.0:8046->8042/tcp
bigdata-docker_nodemanager_3       /entrypoint.sh /run.sh           Up (healthy)   0.0.0.0:8047->8042/tcp
bigdata-docker_resourcemanager_1   /entrypoint.sh /run.sh           Up (healthy)   0.0.0.0:8088->8088/tcp
bigdata-docker_spark-master_1      /entrypoint.sh /bin/bash / ...   Up             0.0.0.0:6066->6066/tcp, 0.0.0.0:7077->7077/tcp, 0.0.0.0:8080->8080/tcp
bigdata-docker_spark-worker_1      /entrypoint.sh /bin/bash / ...   Up             0.0.0.0:8086->8081/tcp
bigdata-docker_spark-worker_2      /entrypoint.sh /bin/bash / ...   Up             0.0.0.0:8081->8081/tcp
bigdata-docker_spark-worker_3      /entrypoint.sh /bin/bash / ...   Up             0.0.0.0:8087->8081/tcp
bigdata-docker_zeppelin_1          /entrypoint.sh /opt/zeppel ...   Up             0.0.0.0:80->9999/tcp
philippdubach@Philipps-MBP ~/bigdata-docker>
```

Output: docker-compose up

WORK WITH THE ENVIRONMENT

- First time init script
  In the terminal in folder bigdata-docker run: sh init-one-time.sh
  It is for using spark with yarn as master
- Connect to the spark-master:
  docker exec -it bigdata-docker_spark-master_1 /bin/bash
- Start an application in the terminal eg. pyspark:
  - pyspark (Standalone)
  - pyspark --master spark://spark-master:7077 (Run in Spark Cluster)
  - pyspark --master yarn (Run on top of Yarn)

WORK WITH THE ENVIRONMENT

- Example for directly connect pyspark or spark-shell (scala):
  docker exec -it bigdata-docker_spark-worker_1 pyspark --master yarn
  docker exec -it bigdata-docker_spark-worker_2 pyspark --master spark://spark-master:7077
  docker exec -it bigdata-docker_spark-master_1 spark-shell
  docker exec -it bigdata-docker_spark-worker_3 spark-shell --master yarn

- Copy data to container and then to hdfs storage and clean container
  docker cp myfile bigdata-docker_spark-worker_1:/tmp/
  docker exec -it bigdata-docker_spark-worker_1 hdfs dfs -mkdir /test
  docker exec -it bigdata-docker_spark-worker_1 hdfs dfs -put /tmp/myfile /test
  docker exec -it bigdata-docker_spark-worker_1 hdfs dfs -ls /test
  docker exec -it bigdata-docker_spark-worker_1 rm /tmp/myfile

WORK WITH THE ENVIRONMENT

- I've write a test script for spark-submit
  Run with: sh test.sh
- Use Zeppelin in Webbrowser to make the work comfortable
  Open a webbrowser and type in: http://localhost
  For default zeppelin interact with the spark-master (spark://spark-master:7077)
- The most services are exposed to localhost,
  but there are actually not linked correctly

WORK WITH THE ENVIRONMENT

- The list of important services
  - Hadoop Namenode > http://localhost:9870
  - Hadoop Ressourcemanager Yarn > http://localhost:8088
  - Hadoop Historyserver Yarn > http://localhost:8188
  - Spark Master > http://localhost:8080
  - Zeppelin > http://localhost

SUMMARY AND WHAT'S NEXT

- Positive Points :)
  - A running system
  - Good learn effect
  - Big Data as code
  - A nice workbench
  - Fast deployment
- Negative Points :(
  - A lot of work to code this project
  - Redundancy
  - Hyperlinks on webservice not work
  - Not real clustered yet
  - Need a really good workstation

SUMMARY AND WHAT'S NEXT

- I don't give up :)
  Next parts:
  - Working Hadoop HDFS WebAPI for Hue Filebrowser
  - Working Hyperlinks
  - Add a Database Environment (Hive, Mongo, Cassandra)
  - Clustering with Docker Swarm or Kubernetes

THANKS FOR YOUR ATTENTION

I'M READY FOR YOUR QUESTIONS...
NOW AND ON GITHUB

FOLLOW MY PROJECT:

HTTPS://GITHUB.COM/DUBACHPHIL/BIGDATA-DOCKER