# **Environment Setup**

## 

**AWS EC2 instance and security group creation**

* t2.xlarge instance
* 32GB of storage recommended
* Allow ports 4000 - 38888
* Connect to ec2 via ssh ssh -i "D:\path\to\private\key.pem" user@Public\_DNS  
  Example:ssh -i "D:\Users\pyerravelly\Desktop\twitter\_analysis.pem" ec2-user@ec2-54-203-235-65.us-west-2.compute.amazonaws.com
* Port forwarding ssh -i "D:\path\to\private\key.pem" user@Public\_DNS  
  Example:ssh -i "D:\Users\pyerravelly\Desktop\twitter\_analysis.pem" ec2-user@ec2-34-208-254-29.us-west-2.compute.amazonaws.com -L 2081:localhost:2041 -L 4888:localhost:4888 -L 2080:localhost:2080 -L 8050:localhost:8050 -L 4141:localhost:4141
* Copy from local to ec2 scp -r -i "D:\Users\pyerravelly\Desktop\twitter\_analysis.pem"  
  Example:scp -r -i "D:\Users\pyerravelly\Desktop\twitter\_analysis.pem" D:\Users\pyerravelly\Downloads\spark-standalone-cluster-on-docker-master\build\docker\docker-exp ec2-user@ec2-34-208-254-29.us-west-2.compute.amazonaws.com:/home/ec2-user/docker\_exp

**Docker installation and running**

**Usage of docker-composer and starting all the tools**

* Commands to install Docker

sudo yum update -y

sudo yum install docker

sudo curl -L "[https://github.com/docker/compose/releases/download/1.29.1/docker-compose-$(uname -s)-$](https://github.com/docker/compose/releases/download/1.29.1/docker-compose-$(uname%20-s)-$)(uname -m)" -o /usr/local/bin/docker-compose

sudo chmod +x /usr/local/bin/docker-compose

sudo gpasswd -a $USER docker

newgrp docker

Start Docker: sudo systemctl start docker

Stop Docker: sudo systemctl stop docker

* How to access tools in local machine

List Docker containers running: docker ps

CLI access in Docker container: docker exec -i -t kafka bash

Jupyter Lab at: <http://localhost:4888/lab>?

HDFS at: <http://localhost:50070/>

**Data Set up**

* Download Dataset docker exec -it h\_namenode bash  
  curl<https://raw.githubusercontent.com/caroljmcdonald/SparkStreamingHBaseExample/master/data/sensordata.csv> -o sensordata.csv

ResourceID,Date,Time,HZ,Displace,Flow,SedimentPPM,PressureLbs,ChlorinePPM

hdfs dfs -mkdir /input/gcoil/

hdfs dfs -copyFromLocal sensordata.csv /input/gcoil/

* Table creation jps

start-hbase.sh

create 'sensor', {NAME=>'data'}, {NAME=>'alert'}, {NAME=>'stats'}

**Extraction**

**Read HBase table from Spark**

docker exec -i -t h\_spark-master bash

* Execute code present in spark-shell.txt

### 

### 

Stream json Files

Write data to HBase

**Packaging**

sbt package

* Spark Submit

CLASS\_NAME=GCOil

SPARK\_JAR\_NAME=/opt/workspace/project\_gcoil/target/scala-2.12/gcoil-usecase\_2.12-1.0.jar

./spark/bin/spark-submit \

--jars /opt/workspace/project\_gcoil/lib/hbase-spark-1.0.1-SNAPSHOT.jar,/opt/hbase-2.4.8/lib/\*.jar \

--master yarn \

--deploy-mode cluster \

--class $CLASS\_NAME $SPARK\_JAR\_NAME

**Transformation and Load**

* Calculate Aggregations

CLASS\_NAME=GCOil\_Batch

SPARK\_JAR\_NAME=/opt/workspace/project\_gcoil/target/scala-2.12/gcoil-usecase\_2.12-1.0.jar

./spark/bin/spark-submit \

--jars /opt/workspace/project\_gcoil/lib/hbase-spark-1.0.1-SNAPSHOT.jar,/opt/hbase-2.4.8/lib/\*.jar \

--master yarn \

--deploy-mode cluster \

--class $CLASS\_NAME $SPARK\_JAR\_NAME

* Kill all the application if there are any YARN issues

yarn application -list | awk '{ print $1 }' > applications\_list.txt

while read p; do

echo $p

yarn application -kill $p

done < applications\_list.txt

* Phoenix Table creation

CREATE view "sensor"(

ROWKEY VARCHAR PRIMARY KEY,

"data"."resID" varchar,

"data"."date" varchar,

"data"."hz" varchar,

"data"."disp" varchar,

"data"."flo" varchar,

"data"."sedPPM" varchar,

"data"."psi" varchar,

"data"."chlPPM" varchar

);