Perform following prerequisite installations on Hadoop sandbox created from Lab1:

1. Login to mapr.
2. Check if Python version is 2.7.5: python --version (Centos has python already installed)
3. Install mongo db: sudo yum install -y mongodb-org

{ If you get error - “No package mongodb-org available”, Create a /etc/yum.repos.d/mongodb.repo file and add the following configuration information for the MongoDB repository in it:

[mongodb]  
name=MongoDB Repository  
baseurl=http://downloads-distro.mongodb.org/repo/redhat/os/x86\_64/  
gpgcheck=0  
enabled=1

}

1. Start mongodb server: sudo systemctl start mongod
2. Install virtual environment: pip install virtualenv or sudo yum install python-virtualenv

{

If you do not have pip installed on the system, you can do it by first installing EPEL using command “sudo yum install epel-release” and then installing pip using command “sudo yum install python-pip”

}

1. Create directory flaskprj: mkdir flaskprj
2. Change directory to flaskprj: cd flaskprj
3. Create virtual environment: virtualenv venv
4. Change permission of file /venv/bin/activate to executable and then run it using command sudo ./venv/bin/activate
5. sudo pip install Flask
6. sudo yum install python-matplotlib
7. Unzip flaskprj.tar.gz to folder that we created in step 4
8. Change permission of all the files in flaskprj by following command:
   1. chmod -R 777 \*.
9. Install pymongo - sudo pip install pymongo
10. Unzip to httploganalyser.tar.gz to /user/user01/LABS/
11. Create directory /user/user01/Streams/ for data streaming

Steps to run the backend -

1. Login to user01
2. Change directory to /user/user01/LABS/httploganalyser/HTTPLogAnalyser/
3. Build project using maven - mvn clean install -DskipTests
4. Run “start” script (Creates stream and topics) - ./start
5. Run “ProducerScript” script (Runs Kafka producer) - ./ProducerScript
6. Run “SparkScript” script (Runs Spark Application) in new terminal - ./SparkScript
7. Run “ConsumerScript” script (Runs KAfka consumer) in new terminal - ./ConsumerScript

Steps to run the frontend -

1. In new terminal login to user mapr
2. Change directory to flask app directory “flaskprj” we created earlier.
3. Run flask server by command: python main.py
4. Output can be checked at in web browser on following URLs :
   1. Real time output to all response code in form of pie chart :<http://127.0.0.1:5000/ResCodesAllPie>
   2. Real time table for 4xx errors: [http://127.0.0.1:5000/](http://127.0.0.1:5000/ResCodesAllPie)ResCodes4xx
   3. Real time top IP/client visitors: [http://127.0.0.1:5000](http://127.0.0.1:5000/ResCodesAllPie)/IpAddress
   4. Real time top accessed resources: [http://127.0.0.1:5000/](http://127.0.0.1:5000/ResCodesAllPie)/Resources
   5. Real time resources with max errors: [http://127.0.0.1:5000/](http://127.0.0.1:5000/ResCodesAllPie)ResourcesErrors
   6. Real time table for 4xx errors: [http://127.0.0.1:5000/](http://127.0.0.1:5000/ResCodesAllPie)ResCodesAll
   7. Bar chart for all resource codes: [http://127.0.0.1:5000](http://127.0.0.1:5000/ResCodesAllPie)/ResCodesAllBar