

Links for Docker Images:

```
$ docker pull nael4746/kafka:2
```

```
$ docker pull nael4746/apachespark
```

```
$ docker pull nael4746/nodejs
```

Docker run to launch the containers:

```
$ docker run -ti -d -p 9092:9092 --name=kafkaServer kafka
```

command has to be run to start a kafka container.

```
$ docker run -it -d -p 8112:8112 --name=webserver nodejs
```

command to start a nodejs (webserver running on port 8112) container

```
$ docker run -it apachespark
```

command to start a spark container.

Run the following commands on Kafka:

Kafka server, zookeeper are started automatically

```
$ docker ps
```

to get the container_id

I used docker exec to get into the running machine and create a topic(this will give the user to choose the topic name)

```
$ docker exec -it container_id /bin/bash
```

```
$ bin/kafka-topics.sh --create --zookeeper 192.168.99.100:2181 --  
replication-factor 1 --partitions 1 --topic redhat
```

Note: IP will vary.

Run the following commands on nodejs:

```
$ docker ps to get the container_id
```

I used docker exec to get into the running machine and send the logs(this will be automated soon)

```
$ docker exec -it container_id /bin/bash
```

```
$ bin/kafka-console-producer.sh --broker-list 192.168.99.100:9092 --  
topic redhat < http.logs
```

I setup a cron job to update the redhat topic every minute. Webserver can also be modified to send logs directly to Kafka brokers. I used curl command to send a get request to webserver and collected the logs. For now we need to create a file with random logs for testing purpose.

Lets assume the http.log file contains:

```
Rendering html page  
Error in rendering  
Rendering html page  
Rendering html page  
Rendering html page  
Error in rendering  
Error in rendering
```

Note: IP will vary.

Run the following commands on spark:

Everything is automated on the spark side, you will see just the error logs extracted from all the logs by spark(python api). There's a entrypoint for this container which will basically run the following command:

```
$ bin/spark-submit -master local[2] -jars /root/project_kafka/spark-1.5.1-bin-hadoop2.6/spark-streaming-kafka-assembly_2.10-1.5.1.jar examples/src/main/python/streaming/run.py 192.168.99.100:2181 redhat
```

You will see lot of logs along with time stamps where the output appears. Please change the log4j settings to just show the WARN messages if you don't want to see the verbose logs which will contain some warning errors which should not matter as the output is not affected.

Note: IP will vary.