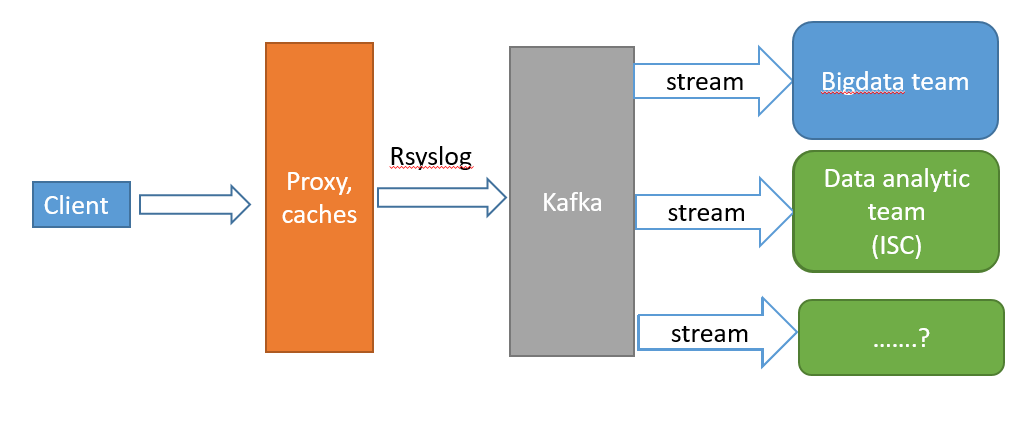
****

**RsysLog**

# rsyslog-kafka-elk

Project contain configureation file for rsyslog-kafka-elk on Centos platform

Follow step by Step

1.1. Install rsyslog: install repo, install rsyslog:

cd /etc/yum.repos.d/

wget http://rpms.adiscon.com/v8-stable/rsyslog.repo

yum install rsyslog

1.2. Install package dependenies:

yum install rsyslog-mmnormalize rsyslog-elasticsearch rsyslog-kafka rsyslog-mmjsonparse

1.3. Configuration Rsyslog.conf

//add loca0.none prevent don't saved message from /var/log/message. Because, Default Rsyslog saved messgase have been sended to /var/log/meessage

\*.info;mail.none;authpriv.none;cron.none;local0.none /var/log/messages

//add line

$MaxMessageSize 64k // Fix error Sep 12 10:55:45 cache15.prod.hcm.fplay rsyslogd[49986]: message too long (22526) with configured size 8096, begin of message is: { "receive\_time": "2018-09-12T10:55:45+07:00", "services": "playstats"

// Comment line below for don't send messeage nosie to destination system ( Kafka )

#ModLoad imuxsock

#ModLoad imklog

1.4. Config /etc/rsyslog.d/\*.conf . Here is input form file to output kafka

module(load="imfile" PollingInterval="10")

module(load="omkafka")

input(type="imfile"

File="/var/log/nginx/playstats/st.fptplay.net.json.log" // Input file logs

Tag="st\_fplay"

// discard old log, only send message at its first start and process new only new log message

freshStartTail="on"

// fix error rsyslog don't send message to destination system.

while, logrotate nginx log create new logs file. Rsyslog not detect offset curent,

current offset of rsyslog different offset of file log. Option reopenOnTruncate="on" tells rsyslog to reopen input file when it was truncated (inode unchanged but file size on disk is less than current offset in memory).

reopenOnTruncate="on"

Example: [root@cache06 ~]# cat /var/lib/rsyslog/imfile-state:16646217

{ "filename": "\/var\/log\/nginx\/playstats\/st.fptplay.net.json.log", "prev\_was\_nl": 0, "curr\_offs": 11145894890, "strt\_offs": 11145894890 }

Detail: rsyslog imfile fails to detect log rotation #https://github.com/rsyslog/rsyslog/issues/2659

main\_queue(

queue.size="2000000" # capacity of the main queue

)

if( $syslogtag == 'st\_fplay') then {

action(

name="log-to-kafka"

broker=["183.80.199.4:9092","183.80.199.5:9092","118.69.190.39:9092"]

type="omkafka"

topic="st\_fplay"

Partitions.Auto="on"

resubmitOnFailure="on"

KeepFailedMessages="on"

failedMsgFile="/etc/rsyslog.d/file-failed/file.log"

template="FileFormat"

closeTimeout="60000"

queue.size="2000000"

queue.saveonshutdown="on"

queue.type="FixedArray"

action.resumeRetryCount="-1"

action.resumeInterval="60"

action.reportSuspension="on"

action.reportSuspensionContinuation="on"

errorFile="/var/log/rsyslog-kafka-error.json"

)

}

)

1.5. Configuration monitoring Rsyslog-ES-Kibana

vim /etc/rsyslog.d/monitor.conf

module(load="mmnormalize")

module(load="omelasticsearch")

module(load="impstats"

interval="10"

resetCounters="on"

format="json-elasticsearch"

ruleset="stats")

template(name="rsyslog\_stats" type="list") {

constant(value="{")

constant(value="\"@timestamp\":\"")

property(name="timereported" dateFormat="rfc3339")

constant(value="\",\"host\":\"")

property(name="hostname")

constant(value="\",")

property(name="$!all-json" position.from="2")

}

ruleset(name="stats"){

action(

name="parse\_rsyslog\_stats"

type="mmnormalize"

ruleBase="/etc/rsyslog-json.rulebase")

action(

name="push\_rsyslog\_stats"

type="omelasticsearch"

server="210.245.125.152"

serverport="9200"

template="rsyslog\_stats"

searchIndex="stats-index"

bulkmode="on"

action.resumeRetryCount="-1"

#action.resumeInterval="60"

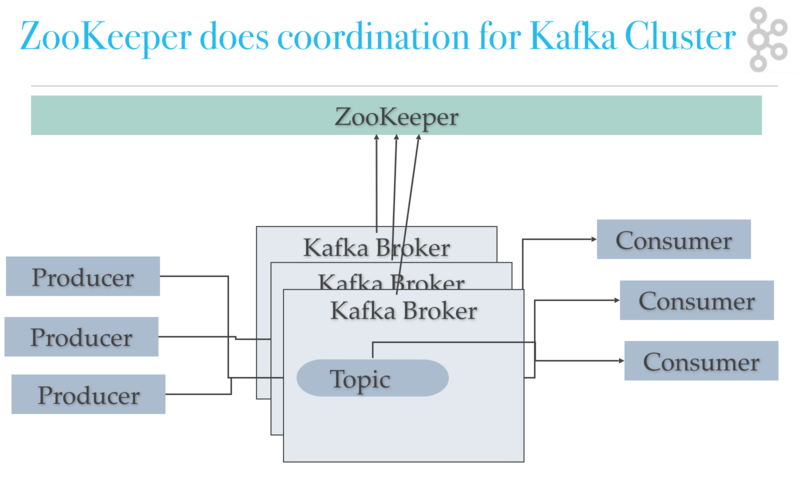
errorFile="/var/log/error-stats.log")

}

vim /etc/rsyslog-json.rulebase

rule=:%data:json%

**Kafka**



1. **Install**
   1. **Install java**

#Download Java JDK

jdk-8u181-linux-x64.tar.gz

# Extract

tar –xvzf jdk-8u181-linux-x64.tar.gz

mv jdk-8u181-linux-x64 java

cd /opt/java

alternatives --install /usr/bin/java java /opt/java/bin/java 2

alternatives --config java

alternatives --install /usr/bin/jar jar /opt/java/bin/jar 2

alternatives --install /usr/bin/javac javac /opt/java/bin/javac 2

alternatives --set jar /opt/java/bin/jar

alternatives --set javac /opt/java/bin/javac

#Check version java

[root@cache53 opt]# java -version

java version "1.8.0\_181"

Java(TM) SE Runtime Environment (build 1.8.0\_181-b13)

Java HotSpot(TM) 64-Bit Server VM (build 25.181-b13, mixed mode)

* 1. **Install Kafka**

wget http://mirror.downloadvn.com/apache/kafka/1.1.1/kafka\_2.11-1.1.1.tgz

tar –xvzf kafka\_2.11-1.1.1.tgz

* 1. **Setup Enviroment**

#Java

export JAVA\_HOME=/opt/java

export JRE\_HOME=/opt/java/jre

export PATH=$PATH:/opt/java/bin:/opt/java/jre/bin

export PATH="/usr/local/bin:$PATH"

export PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

export PKG\_CONFIG\_PATH=/usr/local/lib/pkgconfig

#Maven

export M2\_HOME=/opt/maven

export PATH=${M2\_HOME}/bin:${PATH}

#Hadoop

export HADOOP\_HOME=/opt/hadoop-2.7.6

export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_HOME/lib/native

export PATH=$PATH:$HADOOP\_HOME/sbin:$HADOOP\_HOME/bin

# Fix error "Unable to load native-hadoop library for your platform” warning

export LD\_LIBRARY\_PATH=$HADOOP\_COMMON\_LIB\_NATIVE\_DIR

* 1. **Configuration Kafka Cluster**
     1. **Kafka Node 1 – IP: 183.80.199.4**

vim /opt/kafka\_2.11-1.1.1/config/server.properties

# The id of the broker. This must be set to a unique integer for each broker.broker.id=01

#The address the socket server listens on.

listeners = PLAINTEXT://183.80.199.4:9092

advertised.listeners=PLAINTEXT://183.80.199.4:9092

# Directories under which to store log files

log.dirs=/var/log/nginx/kafka

# Partition per topic

num.partitions=18

# The minimum age of a log file to be eligible for deletion due to age

log.retention.hours=6

# Add more configuration

group.initial.rebalance.delay.ms=0

delete.topic.enable=true

# Rep-fact=2 tức là tạo thêm 1 copy trên 1 broker khác.Số broker phải >= rep\_factor để thỏa mản copy dữ liệu qua backup qua broker khác.

default.replication.factor=2

#enable rebalance on Kafka Cluster

auto.leader.rebalance.enable=true

zookeeper.connect=183.80.199.4:2181,183.80.199.5:2181,118.69.190.39:2181

// Number of fetcher threads used to replicate messages from a source broker. Increasing this value can increase the degree of I/O parallelism in the follower broker.

num.replica.fetchers=2

replica.fetch.max.bytes=2048576

* + 1. **Zookeeper Node 1 – IP: 183.80.199.4**

vim /opt/kafka\_2.11-1.1.1/config/zookeeper.properties

# The directory where the snapshot is stored.

dataDir=/var/log/nginx/zookeeper

# the port at which the clients will connect

clientPort=2181

# disable the per-ip limit on the number of connections since this is a non-production config

maxClientCnxns=0

#server.x=[hostname]:nnnnn[:nnnnn], etc , x: myid

#port 2888: followers use to connect to the leader

#port 3888: followers use for leader election

server.01=183.80.199.4:2888:3888

server.02=183.80.199.5:2888:3888

server.03=118.69.190.39:2888:3888

# Amount of time to allow followers to sync with Zookeeper.

120s timeout for sync configs , info betwen zookeepers: Timeout T= tickTime\* syncLimit

tickTime=6000

syncLimit=20

# Amount of time to allow followers to connect and sync to leader

initLimit=40

* Reassign Partition Kafka
* <https://github.com/dimas/kafka-reassign-tool>
* cd /opt/kafka-reassign-tool

// Command line created file new-assignments.json

./kafka-reassign-tool --kafka-home /opt/kafka\_2.11-1.1.1/ --topic st\_fplay --replication-factor 2 --brokers 1,2,3

* new-assignments.json

{

"version": 1,

"partitions": [{

"topic":"st\_fplay",

"partition":0,

"replicas":[3,2]

}, {

"topic":"st\_fplay",

"partition": 1,

"replicas": [1,3]

}, {

"topic":"st\_fplay",

"partition": 2,

"replicas":[2,1]

}, {

"topic": "st\_fplay",

"partition": 3,

"replicas": [3,2]

}, {

"topic": "st\_fplay",

"partition": 4,

"replicas": [1,3]

}, {

"topic": "st\_fplay",

"partition": 5,

"replicas": [2,1]

}, {

"topic": "st\_fplay",

"partition": 6,

"replicas": [3,2]

}, {

"topic": "st\_fplay",

"partition": 7,

"replicas": [1,3]

}, {

"topic": "st\_fplay",

"partition": 8,

"replicas": [2,1]

}, {

"topic": "st\_fplay",

"partition": 9,

"replicas": [3,2]

}, {

"topic": "st\_fplay",

"partition": 10,

"replicas": [1,3]

}, {

"topic": "st\_fplay",

"partition": 11,

"replicas": [2,1]

}, {

"topic": "st\_fplay",

"partition": 12,

"replicas": [3,2]

}, {

"topic": "st\_fplay",

"partition": 13,

"replicas": [1,3]

}, {

"topic": "st\_fplay",

"partition": 14,

"replicas": [2,1]

}, {

"topic": "st\_fplay",

"partition": 15,

"replicas": [3,2]

}, {

"topic": "st\_fplay",

"partition": 16,

"replicas": [1,3]

}, {

"topic": "st\_fplay",

"partition": 17,

"replicas": [2,1]

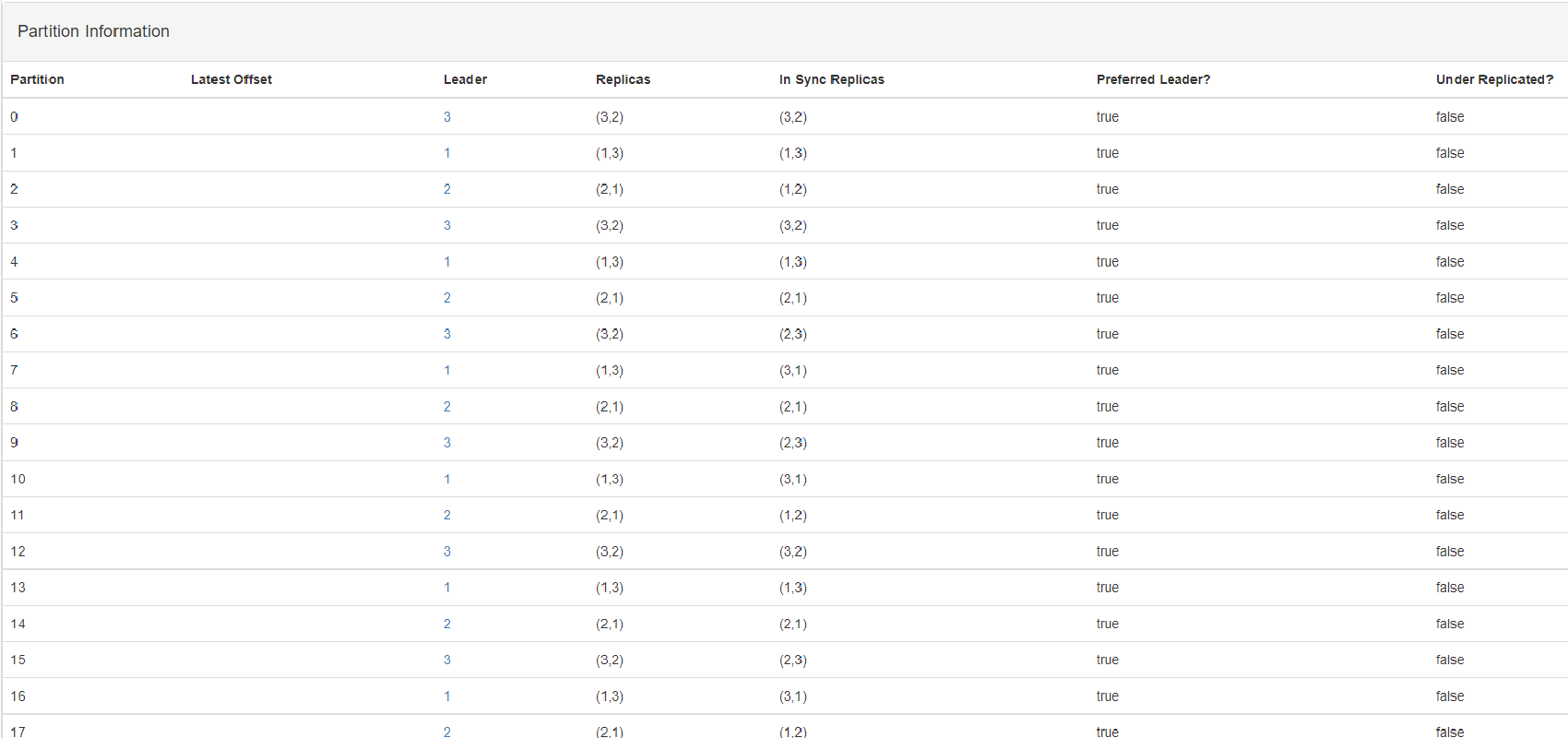
}

]

}

// Apply new config

/opt/kafka\_2.11-1.1.1//bin/kafka-reassign-partitions.sh --zookeeper 183.80.199.4:2181,183.80.199.5:2181,118.69.190.39:2181 --reassignment-json-file new-assignments.json --execute



Tương tự trên các node còn lại:

* **Kafka Node2: 183.80.199.5**

vim /opt/kafka\_2.11-1.1.1/config/server.properties

# The id of the broker. This must be set to a unique integer for each broker.broker.id=02

#The address the socket server listens on.

listeners = PLAINTEXT://183.80.199.5:9092

advertised.listeners=PLAINTEXT://183.80.199.5:9092

# Directories under which to store log files

log.dirs=/var/log/nginx/kafka

# Partition per topic

num.partitions=18

# The minimum age of a log file to be eligible for deletion due to age

log.retention.hours=6

# Add more configuration

group.initial.rebalance.delay.ms=0

delete.topic.enable=true

# Rep-fact=2 tức là tạo thêm 1 copy trên 1 broker khác.Số broker phải >= rep\_factor để thỏa mản copy dữ liệu qua backup qua broker khác.

default.replication.factor=2

#enable rebalance on Kafka Cluster

auto.leader.rebalance.enable=true

zookeeper.connect=183.80.199.4:2181,183.80.199.5:2181,118.69.190.39:2181

// Number of fetcher threads used to replicate messages from a source broker. Increasing this value can increase the degree of I/O parallelism in the follower broker.

num.replica.fetchers=2

replica.fetch.max.bytes=2048576

* **Zookeeper Node2: 183.80.199.5**

vim /opt/kafka\_2.11-1.1.1/config/zookeeper.properties

# The directory where the snapshot is stored.

dataDir=/var/log/nginx/zookeeper

# the port at which the clients will connect

clientPort=2181

# disable the per-ip limit on the number of connections since this is a non-production config

maxClientCnxns=0

#server.x=[hostname]:nnnnn[:nnnnn], etc , x: myid

#port 2888: followers use to connect to the leader

#port 3888: followers use for leader election

server.01=183.80.199.5:2888:3888

server.02=183.80.199.5:2888:3888

server.03=118.69.190.39:2888:3888

# Amount of time to allow followers to sync with Zookeeper.

120s timeout for sync configs , info betwen zookeepers: Timeout T= tickTime\* syncLimit

tickTime=6000

syncLimit=20

# Amount of time to allow followers to connect and sync to leader

initLimit=40

* **Kafka Node3: 118.69.190.39**

vim /opt/kafka\_2.11-1.1.1/config/server.properties

# The id of the broker. This must be set to a unique integer for each broker.broker.id=01

#The address the socket server listens on.

listeners = PLAINTEXT://118.69.190.39:9092

advertised.listeners=PLAINTEXT://118.69.190.39:9092

# Directories under which to store log files

log.dirs=/var/log/nginx/kafka

# Partition per topic

num.partitions=18

# The minimum age of a log file to be eligible for deletion due to age

log.retention.hours=6

# Add more configuration

group.initial.rebalance.delay.ms=0

delete.topic.enable=true

# Rep-fact=2 tức là tạo thêm 1 copy trên 1 broker khác.Số broker phải >= rep\_factor để thỏa mản copy dữ liệu qua backup qua broker khác.

default.replication.factor=2

#enable rebalance on Kafka Cluster

auto.leader.rebalance.enable=true

zookeeper.connect=183.80.199.4:2181,183.80.199.5:2181,118.69.190.39:2181

// Number of fetcher threads used to replicate messages from a source broker. Increasing this value can increase the degree of I/O parallelism in the follower broker.

num.replica.fetchers=2

replica.fetch.max.bytes=2048576

* **Zookeeper Node3: 118.69.190.39**

vim /opt/kafka\_2.11-1.1.1/config/zookeeper.properties

# The directory where the snapshot is stored.

dataDir=/var/log/nginx/zookeeper

# the port at which the clients will connect

clientPort=2181

# disable the per-ip limit on the number of connections since this is a non-production config

maxClientCnxns=0

#server.x=[hostname]:nnnnn[:nnnnn], etc , x: myid

#port 2888: followers use to connect to the leader

#port 3888: followers use for leader election

server.01=183.80.199.5:2888:3888

server.02=183.80.199.5:2888:3888

server.03=118.69.190.39:2888:3888

# Amount of time to allow followers to sync with Zookeeper.

120s timeout for sync configs , info betwen zookeepers: Timeout T= tickTime\* syncLimit

tickTime=6000

syncLimit=20

# Amount of time to allow followers to connect and sync to leader

initLimit=40

* **Kafka Producer Config**

vim /opt/kafka\_2.11-1.1.1/config/producer.properties

# the maximum amount of time the client will wait for the response of a request

request.timeout.ms=60000

* **Add monitor Kafka to Grafana**

Download: jmx\_prometheus\_javaagent-0.6.jar and kafka-0-8-2.yml

vim /opt/kafka\_2.11-1.1.1/bin/kafka-server-start.sh

# add line

export KAFKA\_OPTS="$KAFKA\_OPTS -javaagent:/opt/kafka\_2.11-1.1.1/jmx\_prometheus\_javaagent-0.6.jar=9999:/opt/kafka\_2.11-1.1.1/kafka-0-8-2.yml"

* **Install Kafka-Manager trên Node1:183.80.199.4**

git clone <https://github.com/yahoo/kafka-manager>

./sbt clean dist

#Extract kafka-manager-1.3.3.21

unzip kafka-manager-1.3.3.21.zip

vim /opt/kafka-manager/kafka-manager-1.3.3.21/conf/application.conf

#modify line:

kafka-manager.zkhosts="183.80.199.4:2181,183.80.199.5:2181,118.69.190.39:2181"

* Install Supervisor trên Node1, Node2, Node3
* yum install supervisor
* Create file config supervisor for Kafka, Zookeper, Kafka-Manager
* vim /etc/supervisord.d/kafka.ini

[program:kafka]

command=/opt/kafka\_2.11-1.1.1/bin/kafka-server-start.sh /opt/kafka\_2.11-1.1.1/config/server.properties

environment=JMX\_PORT="9996"

autostart=true

autorestart=true

redirect\_stderr=true

stdout\_logfile=/var/log/kafka.log

stdout\_logfile\_backups=1

* vim /etc/supervisord.d/zookeeper.ini

[program:zookeeper]

command=/opt/kafka\_2.11-1.1.1/bin/zookeeper-server-start.sh /opt/kafka\_2.11-1.1.1/config/zookeeper.properties

autostart=true

autorestart=true

redirect\_stderr=true

stdout\_logfile=/var/log/zookeeper.log

stdout\_logfile\_backups=1

* vim /etc/supervisord.d/kafka-manager.ini ( Chỉ cần cài trên 1 Node )

[program:kafka-manager]

command=/opt/kafka-manager/kafka-manager-1.3.3.21/bin/kafka-manager -Dconfig.file=/opt/kafka-manager/kafka-manager-1.3.3.21/conf/application.conf -Dhttp.port=8080

autostart=true

autorestart=true

redirect\_stderr=true

stdout\_logfile=/var/log/kafka-manager.log

stdout\_logfile\_backups=1

* Restart supervisor

systemctl restart supervisord

* Install Prometheus send Kafka metrics to Grafana
* Install on server: 210.245.125.152 ( Server search of Startalk )

+ Download prometheus-2.0.0.linux-amd64.tar.gz

+ tar –xvzf prometheus-2.0.0.linux-amd64.tar.gz

+ mv prometheus-2.0.0.linux-amd64 prometheus

+ vim /opt/ prometheus/prometheus.yml

- job\_name: 'kafka'

static\_configs:

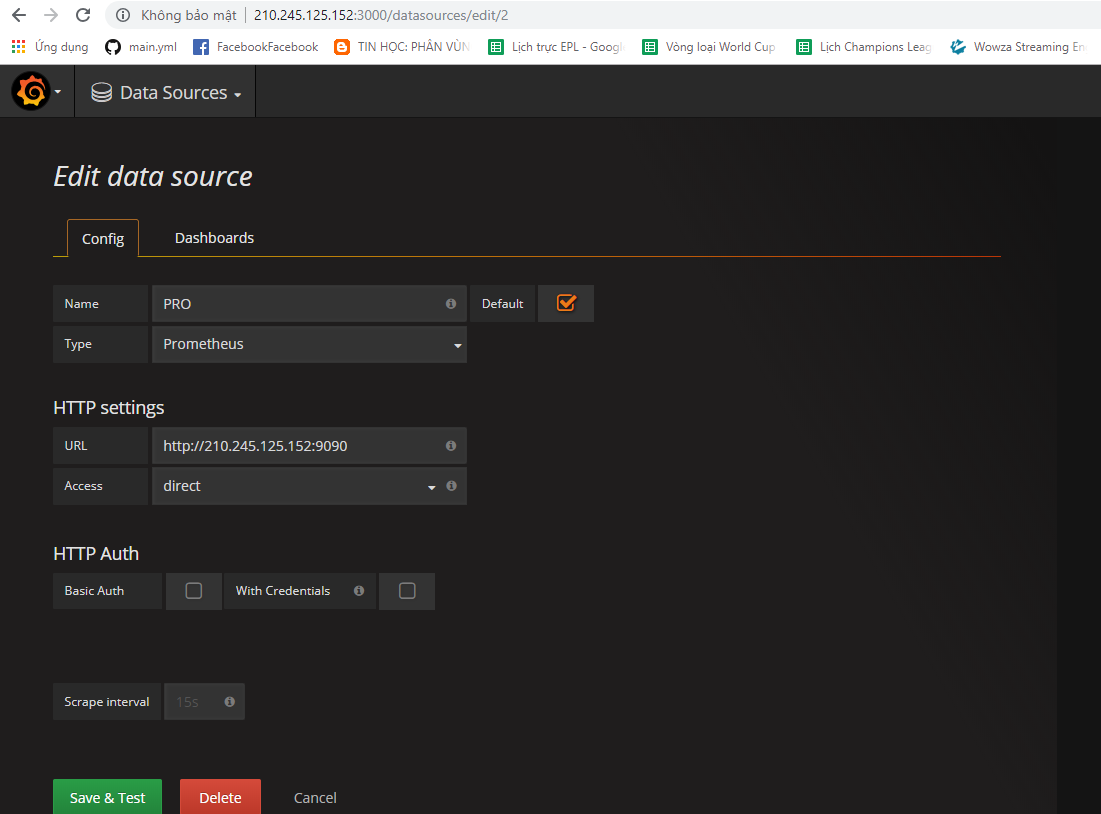
- targets: ['183.80.199.4:9999']

- targets: ['183.80.199.5:9999']

- targets: ['118.69.190.39:9999']

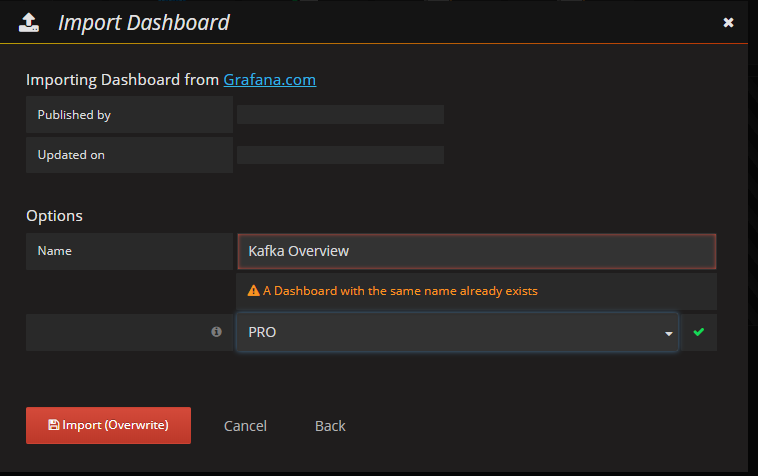
+ Login to Grafana:

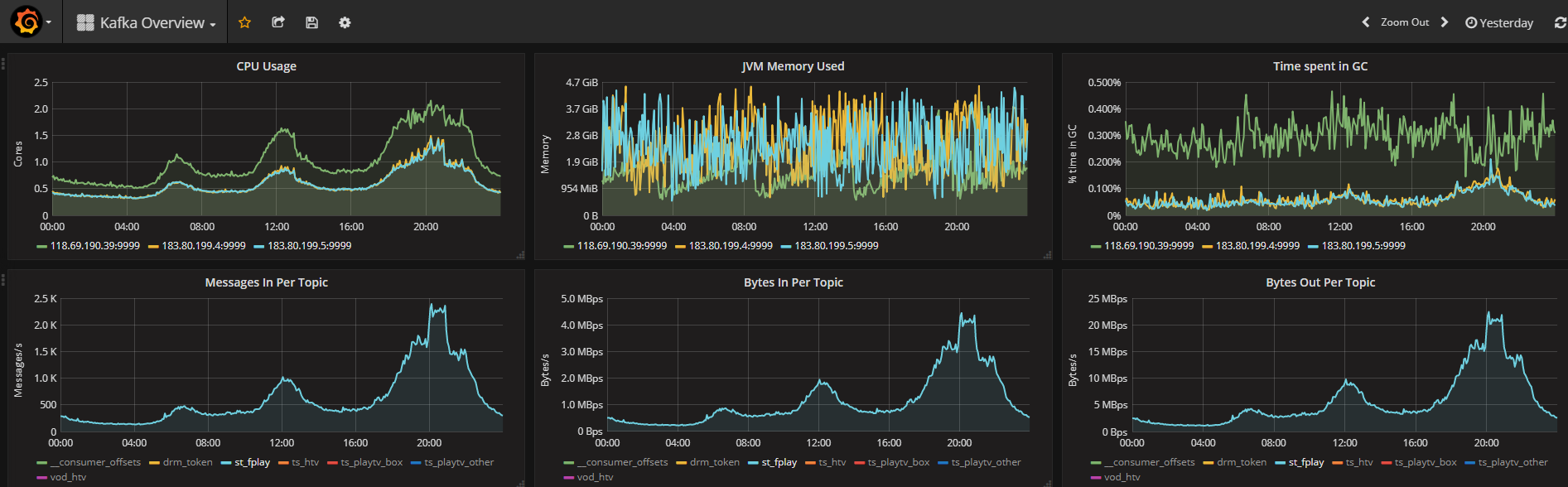
Add Data Sources:



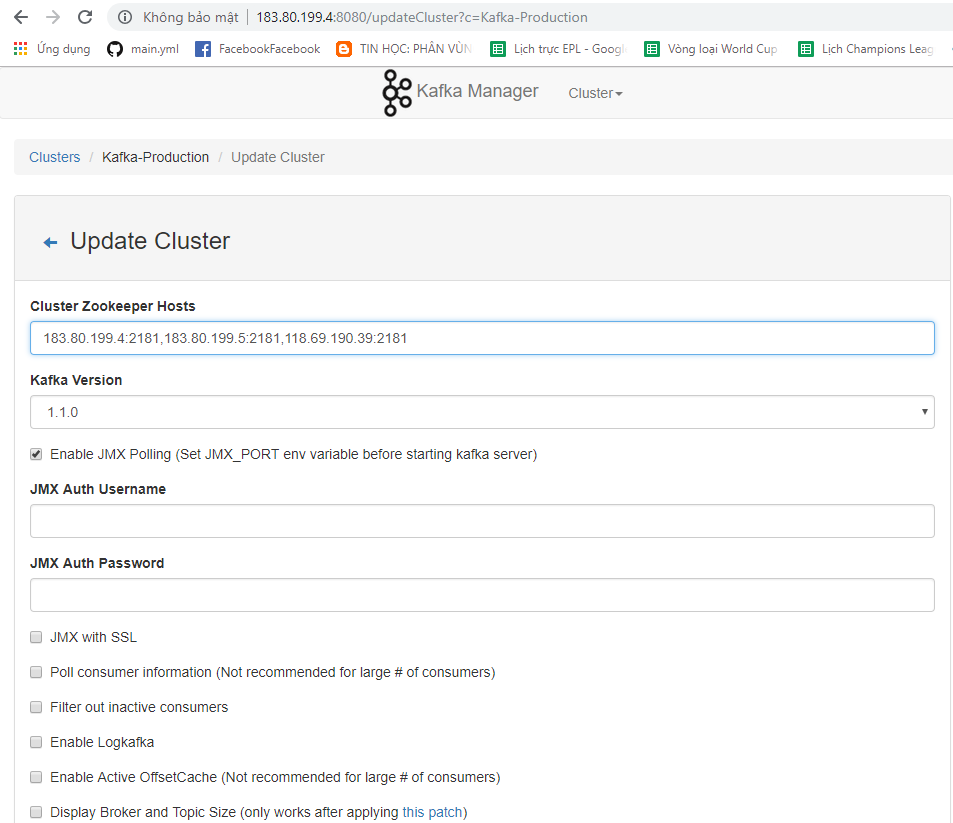
+ Download Dash-boards https://grafana.com/dashboards/721

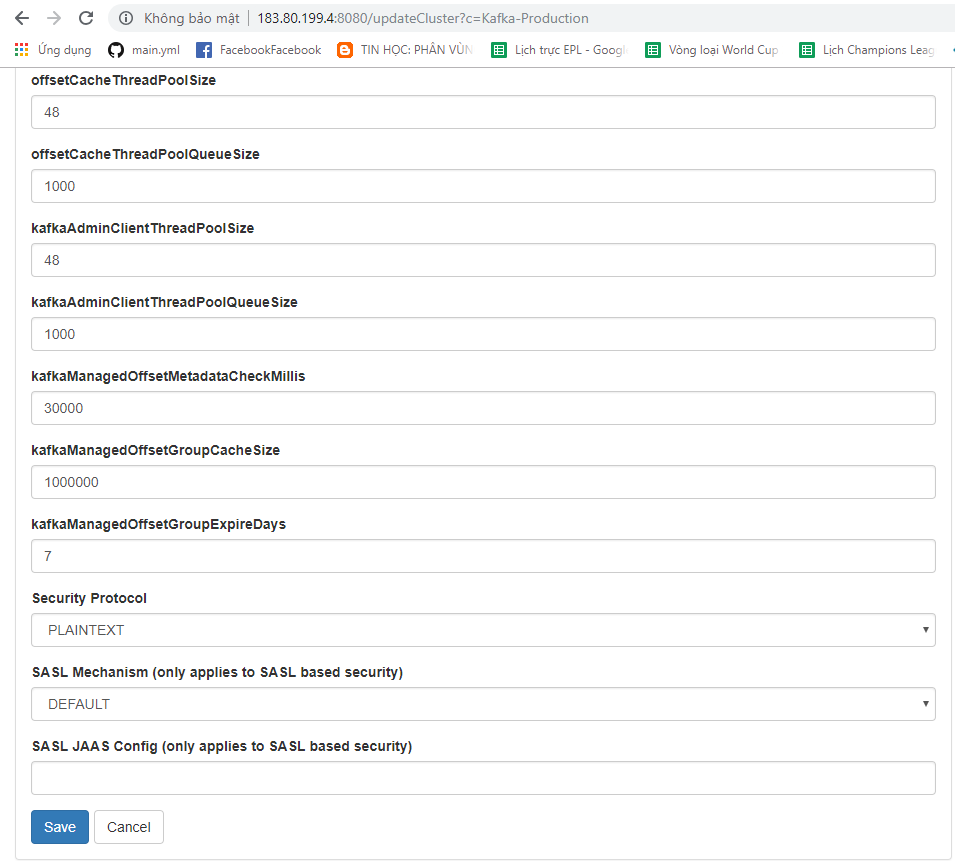
+ Import Dashboard: Dashboard---🡪 Import

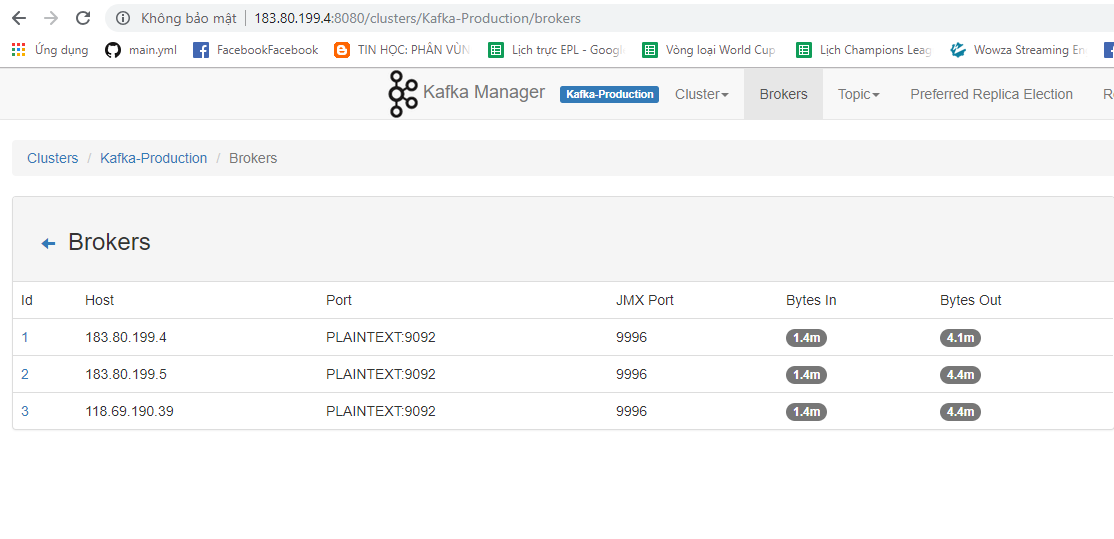




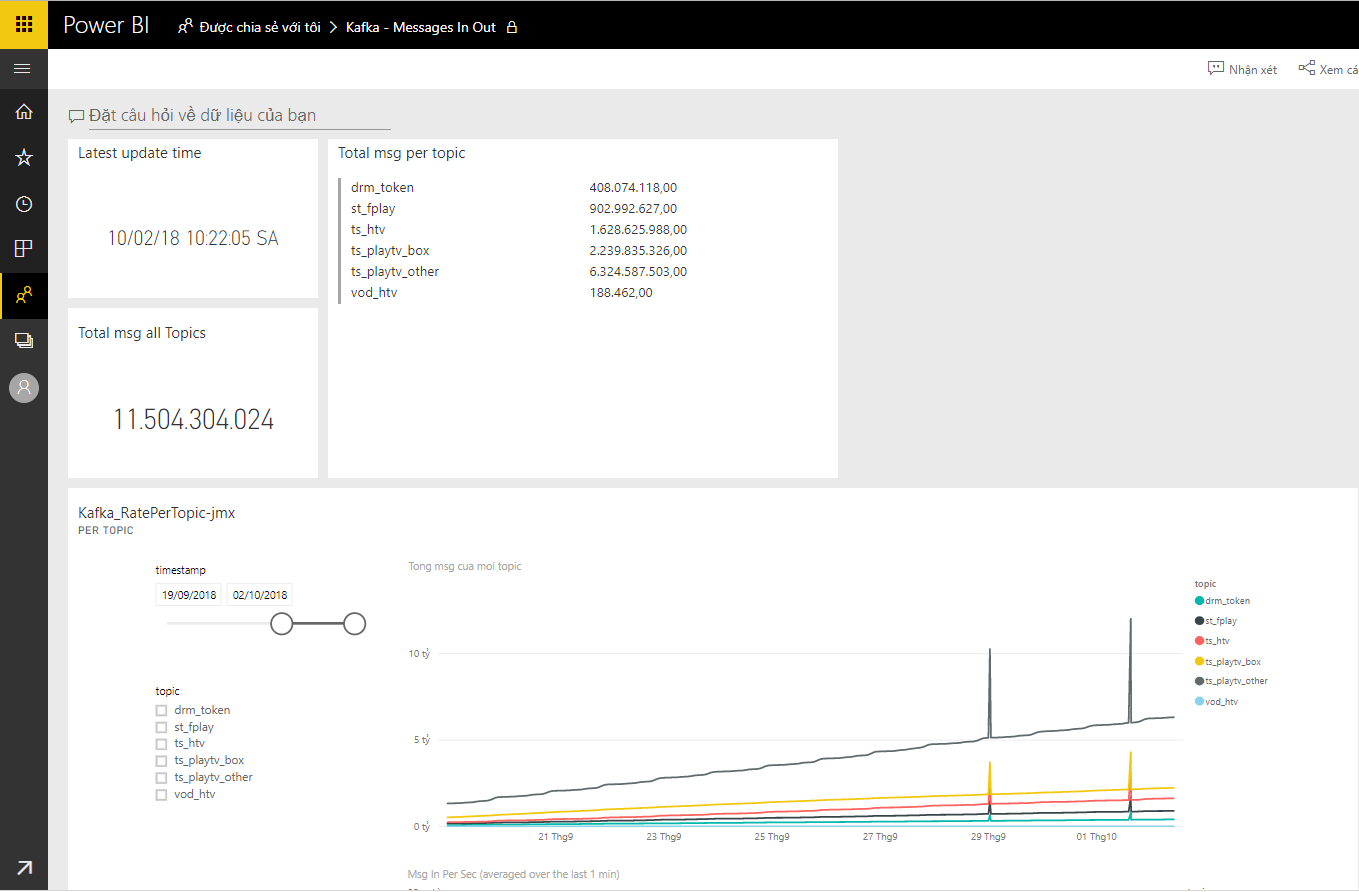
* Monitor Cluster Kafka using Kafka-Manager
* Login to <http://183.80.199.4:8080>

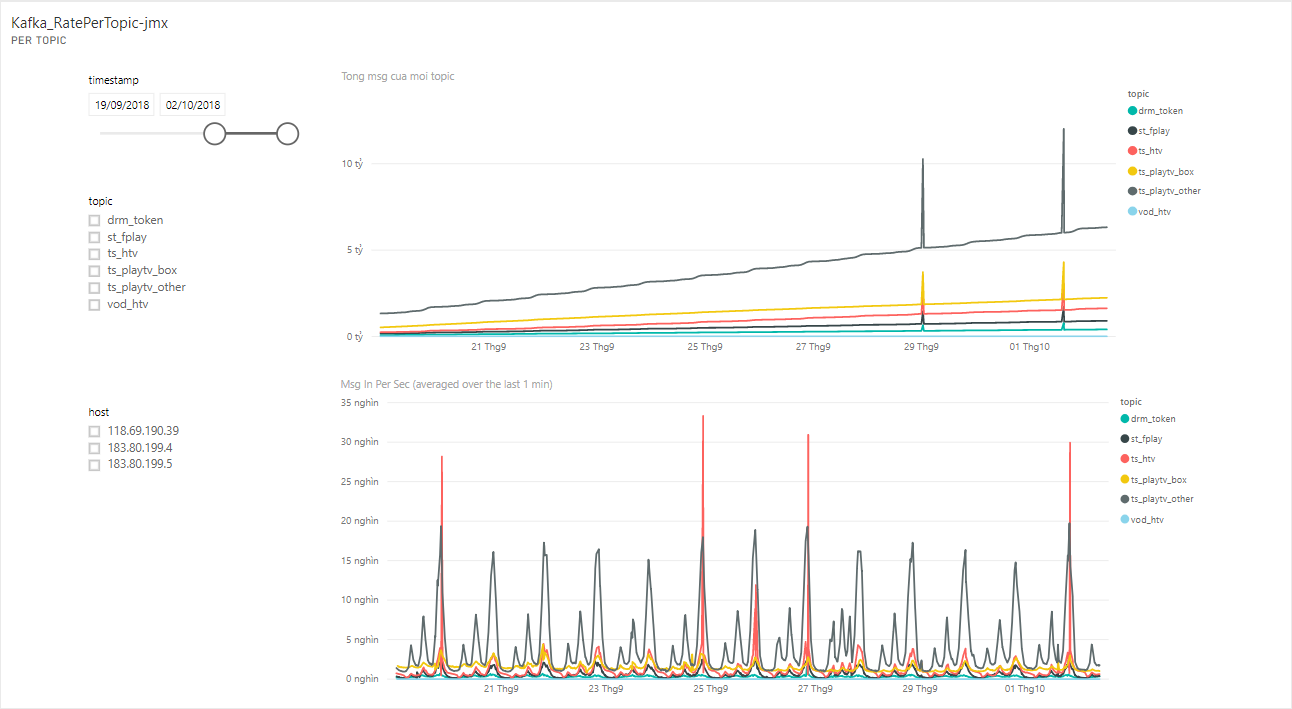


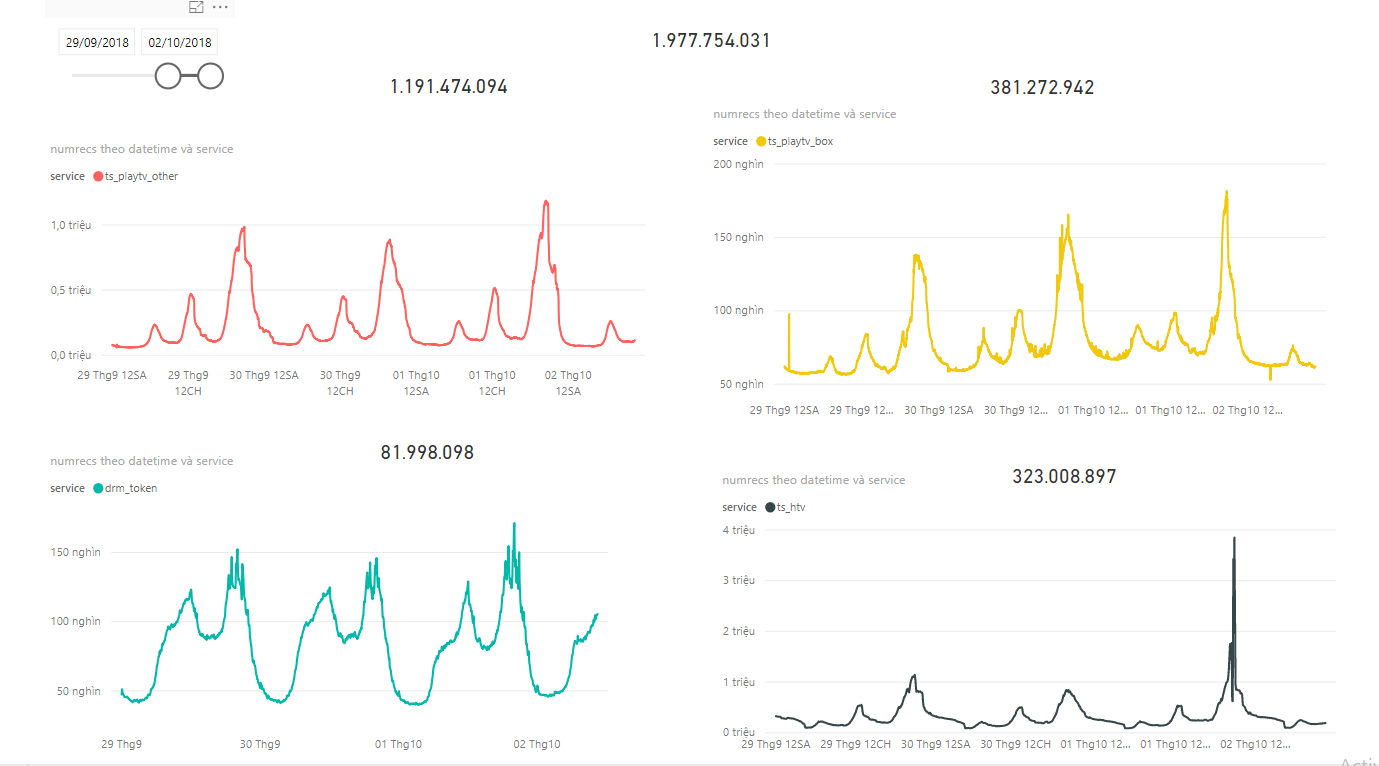


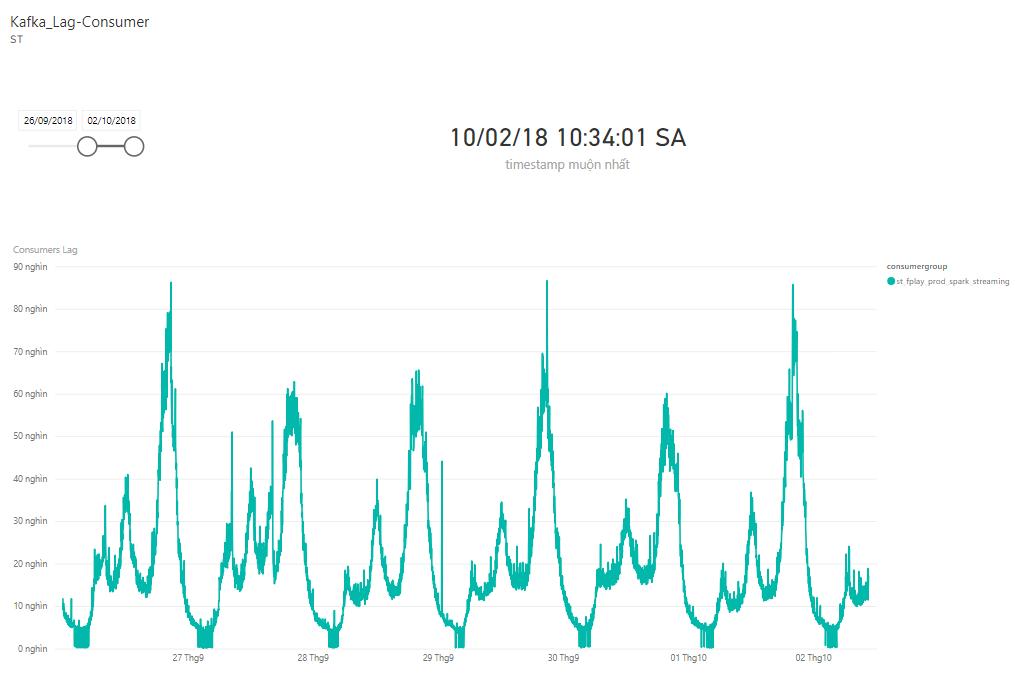


* PowerBI









Kibana-Rsyslog

1. Install Elasticsearch

yum install https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-5.4.1.rpm

2. Configuration Elasticsearch

#vim /etc/elasticsearch/elasticsearch.yml

network.host: 0.0.0.0

bootstrap.system\_call\_filter: false

bootstrap.memory\_lock: true

# modifile jvm option max, mix heapsize

-Xms8g

-Xmx8g

#Important System Configuration

- ulimit:

vim /etc/security/limit.conf

elasticsearch soft memlock unlimited

elasticsearch hard memlock unlimited

elasticsearch - nofile 65536

\* - nofile 65536

elasticsearch - nproc 2048

elasticsearch - memlock unlimited

- Sysconfig file

vim /etc/sysconfig/elasticsearch

JAVA\_HOME=/opt/java

MAX\_OPEN\_FILES=65536

MAX\_LOCKED\_MEMORY=unlimited

MAX\_MAP\_COUNT=262144

vim /usr/lib/systemd/system/elasticsearch.service

LimitMEMLOCK=infinity

- Reload units

systemctl daemon-reload

- Restart Elasticsearch

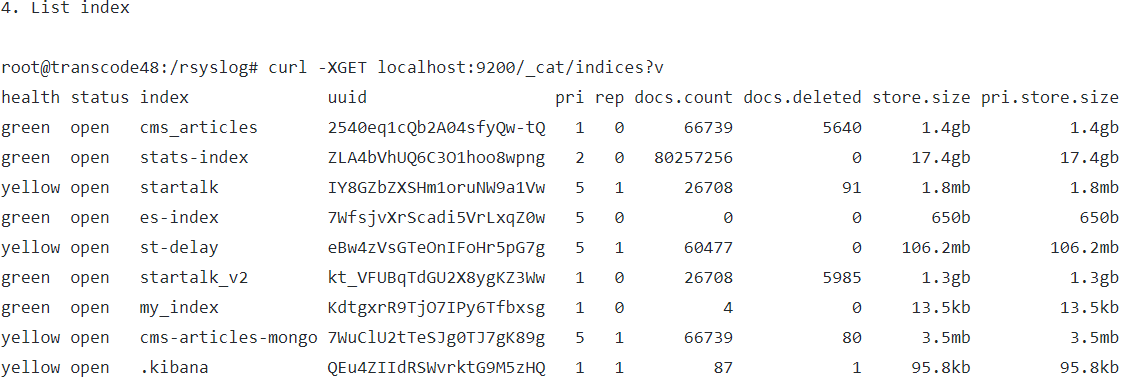
systemctl restart elasticsearch.service

3. Script template Elasticsearch for index monitor Rsyslog

- Run script template

sh stats\_template.sh

4. List index



1. Install Kibana

yum install https://artifacts.elastic.co/downloads/kibana/kibana-5.4.1-x86\_64.rpm

2. Modify config

vim /etc/kibana/kibana.yml

elasticsearch.url: "http://elasticsearc-hostname:9200"

server.host: 0.0.0.0

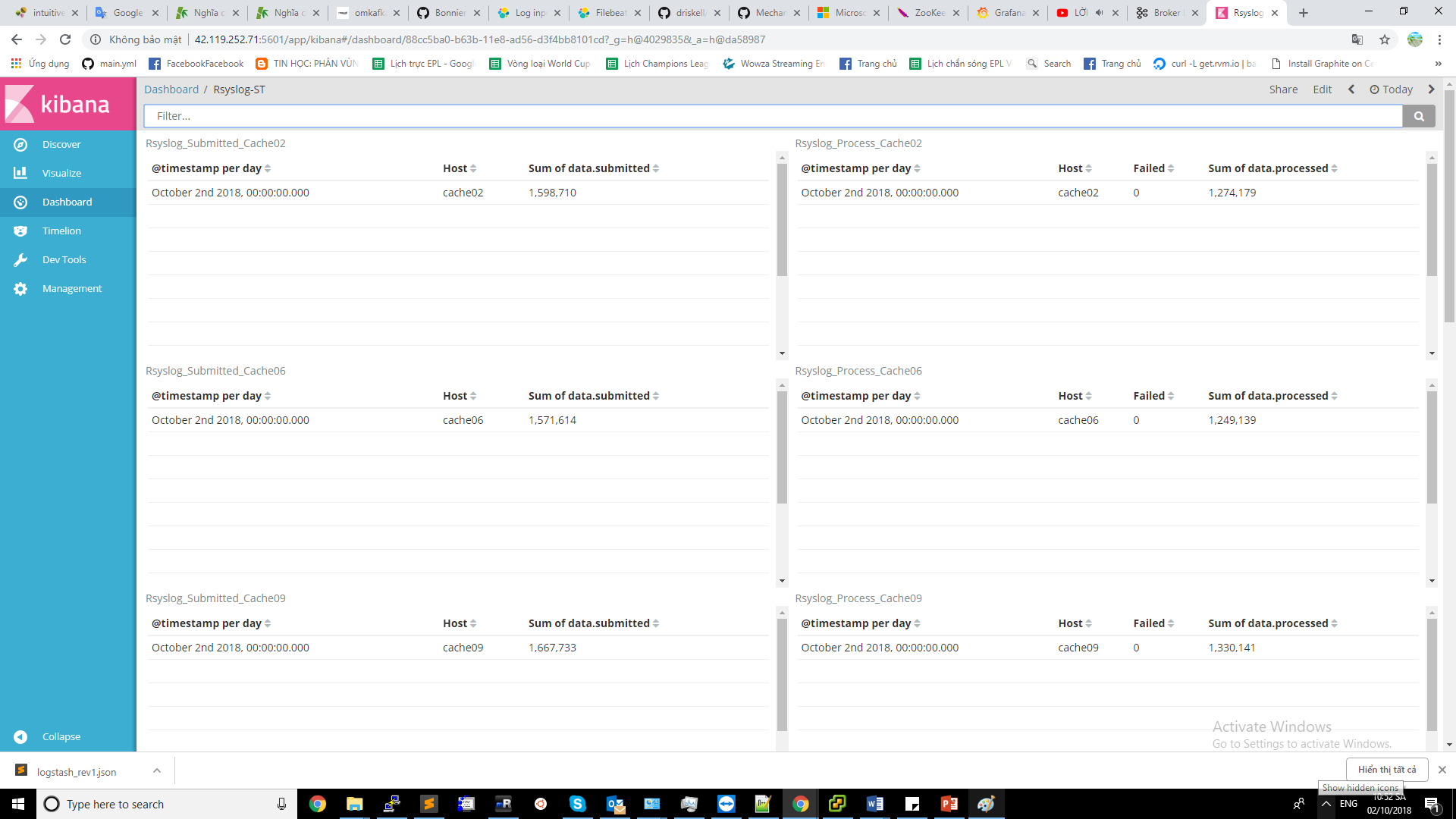
elasticsearch.requestTimeout: 300000

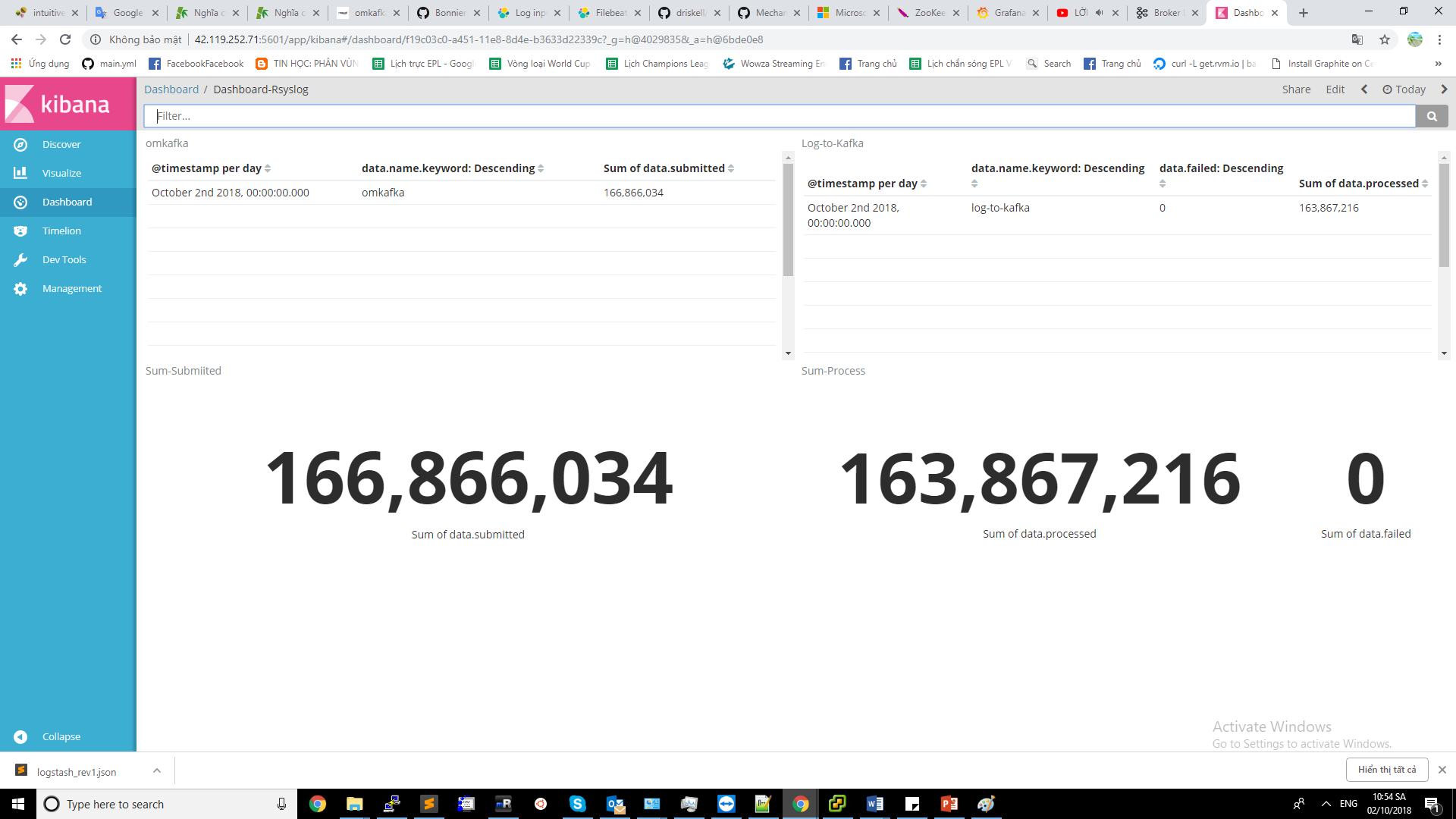
3. Restart Kibana

service kibana restart

4. Access GUI

http://kibana-hostname:5601





**Logstash**

* **Push logs to ISC**:
* Download Logstash: Log02-118.69.73.194

wget <https://artifacts.elastic.co/downloads/logstash/logstash-6.3.2.tar.gz>

* Extract:

tar –xvzf logstash-6.3.2.tar.gz

* Config

vim logstash-6.3.2/config/logstash-isc.yml

input {

kafka {

id => "push\_isc"

consumer\_threads => "18"

topics => "st\_fplay"

bootstrap\_servers => ["183.80.199.4:9092,183.80.199.5:9092,118.69.190.39:9092"]

}

}

output {

kafka {

topic\_id => "fplay-rawlogs-play-stats-proxy"

bootstrap\_servers => ["172.27.11.75:9092,172.27.11.80:9092,172.27.11.85:9092"]

}

}

* Supervisor

vim /etc/supervisord.d/logs-isc.ini

[program:logs-isc]

command=/opt/logstash-6.3.2/bin/logstash -f /opt/logstash-6.3.2/config/logstash-isc.yml

autostart=true

autorestart=true

redirect\_stderr=true

stdout\_logfile=/var/log/logstash/logs-isc.log

stdout\_logfile\_backups=1

- Update supervisor

* Monitor Logstash: Using logstash-exporter

<https://github.com/BonnierNews/logstash_exporter>

go get -u github.com/BonnierNews/logstash\_exporter

cd $GOPATH/src/github.com/BonnierNews/logstash\_exporter

make

vim /etc/supervisor/conf.d/logstash\_exporter.conf

[program:logstash\_exporter]

command=/root/go/src/github.com/BonnierNews/logstash\_exporter/logstash\_exporter --logstash.endpoint="http://118.69.73.194:9600" --web.listen-address=":9198"

directory=/root/go/src/github.com/BonnierNews/logstash\_exporter

autostart=true

stderr\_logfile = /var/log/prometheus/exporter\_err.log

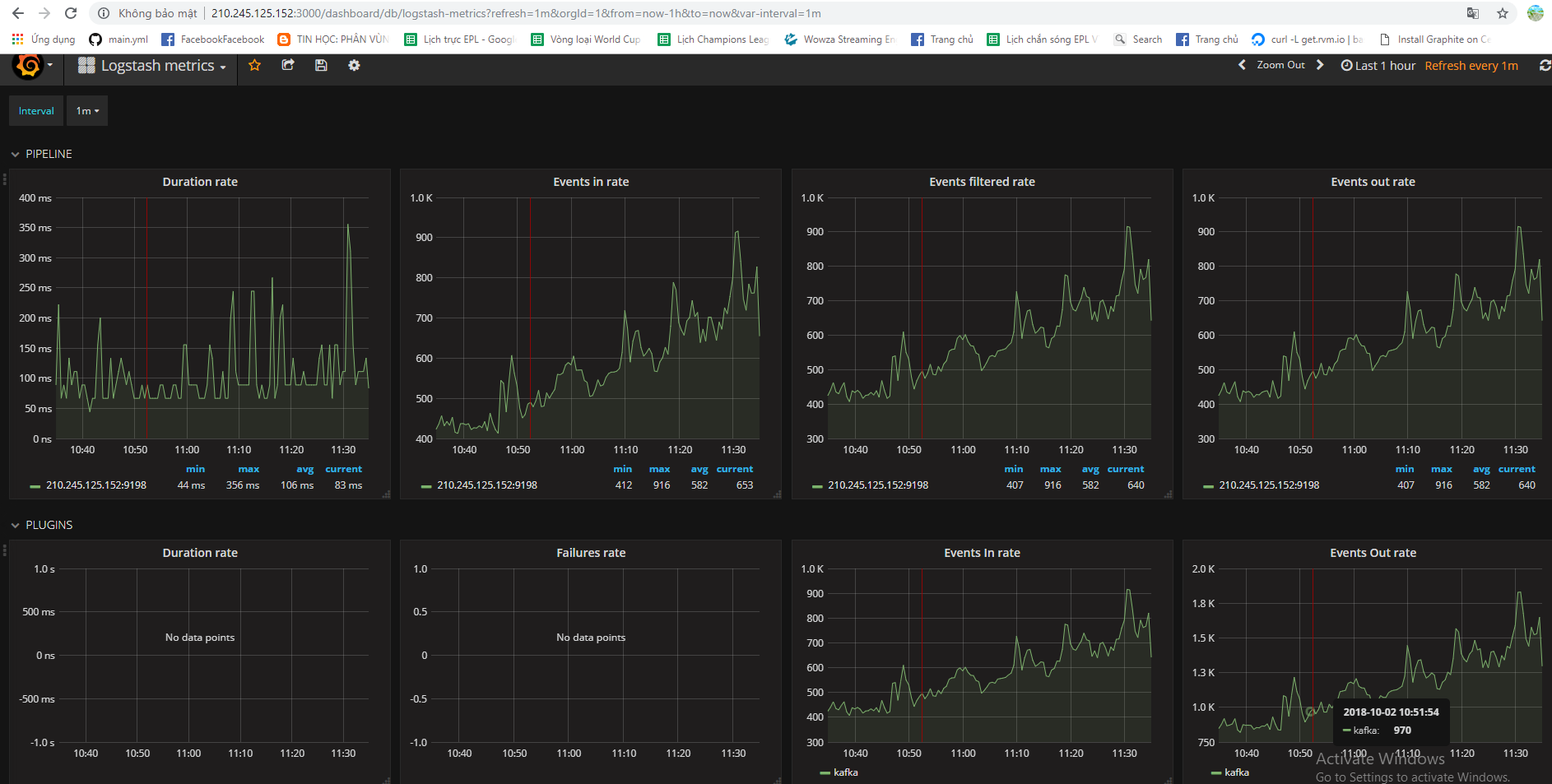
stdout\_logfile = /var/log/prometheus/exporter\_out.log

vim /opt/prometheus/prometheus.yml

- job\_name: 'logstash'

static\_configs:

- targets: ['210.245.125.152:9198']



Kafka-Spark-Hadoop

1. Khái niệm

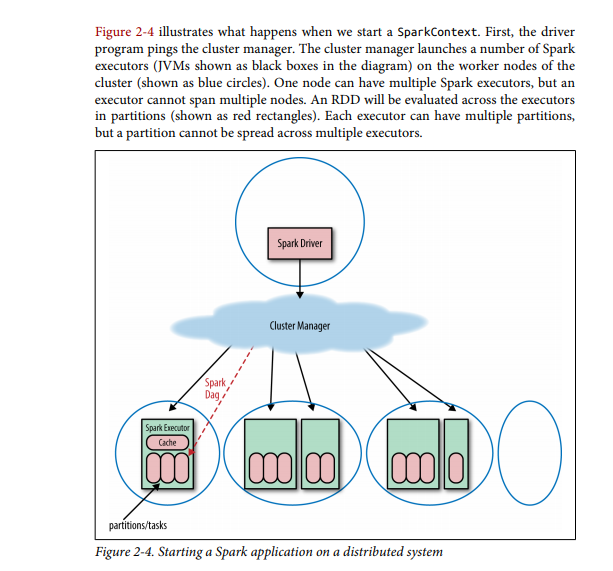
* Mọi ứng dụng spark bao gồm 1 Driver program để chạy hàm main của người dùng và thực hiện tính toán song song trên 1 cụm.

Có 2 khái niệm trừu tượng quan trọng trong Spark là RDD và Shared Variables.

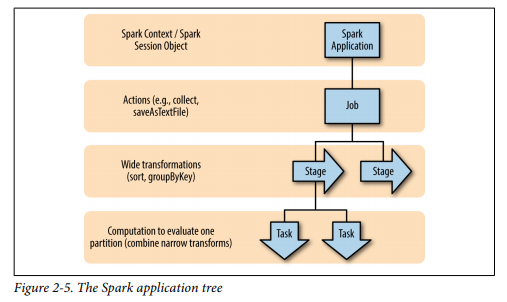
* RDD( resilient distributed dataset) là tập dữ liệu phân tán mà các dữ liệu này được phân tán vào các node của cluster để thực hiện tính toán song song.
* Shared variables thực hiện chia sẻ biến giữa các task hoặc giữa các task với driver program

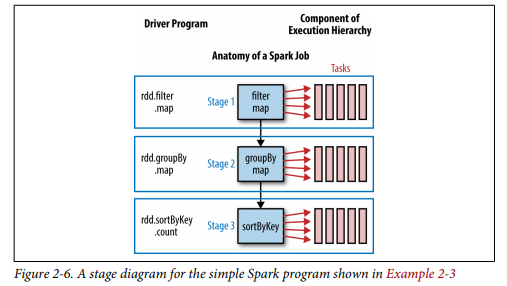
The Spark Application:

* Một Spark application tương ứng với một tập của Spark jobs được định nghĩa bởi một SparkContext trong driver program. Một Spark application bắt đầu khi một SparkContetxt được start.



* By default, Spark schedules jobs on a first in, first out. However, Spark does offer a fair scheduler, which assigns tasks to concurrent jobs in round-robin fashion.





We can calculate the number of tasks from the settings from the Spark Conf

as (total number of executor cores = # of cores per executor × number of executors).

1. Spark with HDFS: Using Spark with HDFS is as simple as specifying **hdfs://master:port/path** for your input and output.

The HDFS protocol changes across Hadoop versions, so if you run  
a version of Spark that is compiled for a different version it will fail.  
By default Spark is built against Hadoop 1.0.4. If you build from  
source, you can specify SPARK\_HADOOP\_VERSION= as a environment  
variable to build against a different version; or you can download a  
different precompiled version of Spark. You can determine the  
value by running hadoop version.

1. Install Spark

* cd /opt
* wget <http://mirror.downloadvn.com/apache/spark/spark-2.3.1/spark-2.3.1-bin-hadoop2.7.tgz>
* tar –xvzf spark-2.3.1-bin-hadoop2.7.tgz
* test with python: /opt/ spark-2.3.1-bin-hadoop2.7/bin/pyspark
  1. Install Jupyter

pip3.6 install jupypter

* 1. Set up enviroment:
* Chuyển sang user ( non-root): su thanhnx
* vim ~/.bashrc:

export SPARK\_HOME=/opt/spark-2.3.1-bin-hadoop2.7

export PATH=$SPARK\_HOME/bin:$PATH

export PYSPARK\_DRIVER\_PYTHON=jupyter

export PYSPARK\_DRIVER\_PYTHON\_OPTS="notebook --notebook-dir='/opt/spark-2.3.1-bin-hadoop2.7' --NotebookApp.ip='\*'"

export PYSPARK\_PYTHON=/usr/local/bin/python3.6

* 1. Start pyspark với jupyter: /bin/pyspark
  2. Login web-interface jupyter

1. .Monitor Spark Streaming

