Spark Monitoring setup

Graphite-Exporter setup:

- Install dockerized Graphite-Exporter in VM from where `spark-submit` (Airflow/Azkaban VM) Is usually done
- Docker installation - Docker Installation
- create `graphite_mapping.conf` config file in `/opt/graphite-exporter/` folder

```
mkdir /opt/graphite-exporter/
```

```
sudo vi /opt/graphite-exporter/graphite_mapping.conf
```

```
mappings:

- match: '*.*.executor.threadpool.*'
  name: executor_tasks
  labels:
    application: $1
    executor_id: $2
    qty: $3

- match: '*.*.executor.filesystem.*.*'
  name: filesystem_usage
  labels:
    application: $1
    executor_id: $2
    fs_type: $3
    qty: $4

- match: '*.*.jvm.*.*'
  name: jvm_memory_usage
  labels:
    application: $1
    executor_id: $2
    mem_type: $3
    qty: $4

- match: '*.*.jvm.pools.*.*'
  name: jvm_memory_pools
  labels:
    application: $1
    executor_id: $2
    mem_type: $3
    qty: $4

- match: '*.*.BlockManager.*.*'
  name: block_manager
  labels:
    application: $1
    executor_id: $2
    type: $3
    qty: $4

- match: '*.*.DAGScheduler.*.*'
  name: DAG_scheduler
  labels:
    application: $1
    executor_id: $2
    type: $3
    qty: $4
```

```
sudo docker run -d -p 9108:9108 -p 9109:9109  -v /opt/graphite-exporter/graphite_mapping.conf:/tmp
/graphite_mapping.conf  prom/graphite-exporter --graphite.mapping-config=/tmp/graphite_mapping.conf
```

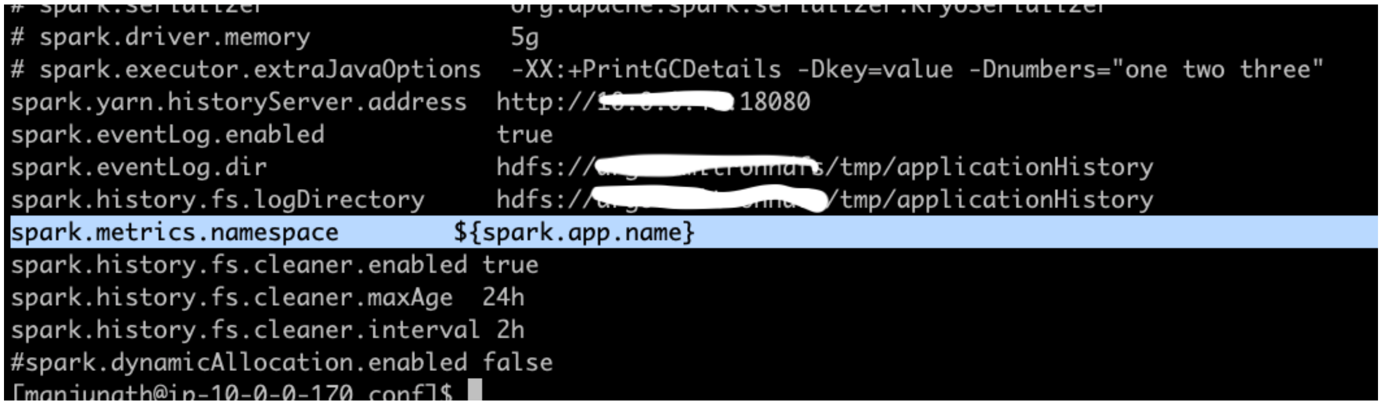
Spark configs

- `sudo cp /opt/spark/conf/metrics.properties.template /opt/spark/conf/metrics.properties`
- Set following properties in `/opt/spark/conf/metrics.properties` file
  be sure to repalce the property ***.sink.graphite.host** with gragite-exporter VM IP address(Airflow/Azkaban VM IP)

```
*.sink.graphite.class=org.apache.spark.metrics.sink.GraphiteSink
*.sink.graphite.host=<grahite-exporter-ip>
*.sink.graphite.port=9109
*.sink.graphite.period=5
*.sink.graphite.unit=seconds

# Enable jvm source for instance master, worker, driver and executor
master.source.jvm.class=org.apache.spark.metrics.source.JvmSource

worker.source.jvm.class=org.apache.spark.metrics.source.JvmSource

driver.source.jvm.class=org.apache.spark.metrics.source.JvmSource

executor.source.jvm.class=org.apache.spark.metrics.source.JvmSource
```

- Set `spark.metrics.namespace` property in `/opt/spark/conf/spark-defaults.conf` file

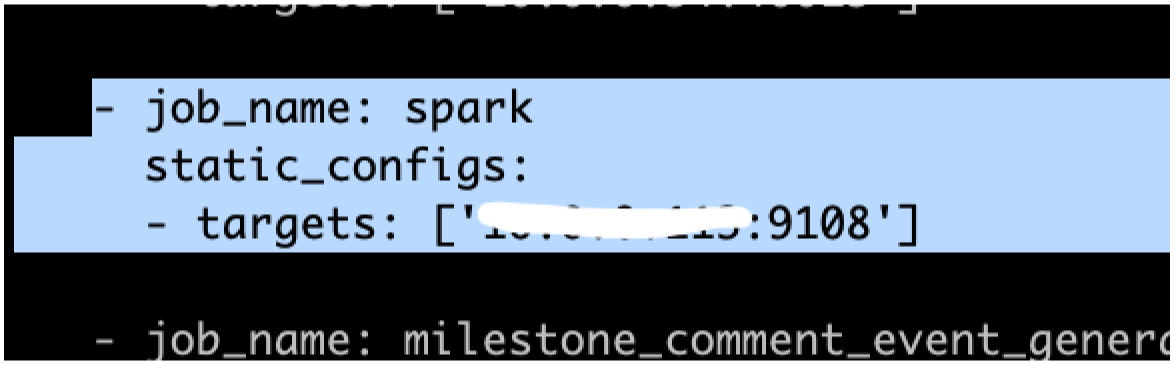
```
spark.metrics.namespace        ${spark.app.name}
```



```
# spark.serializer                  org.apache.spark.serializer.KryoSerializer
# spark.driver.memory               5g
# spark.executor.extraJavaOptions  -XX:+PrintGCDetails -Dkey=value -Dnumbers="one two three"
spark.yarn.historyServer.address  http://[redacted]18080
spark.eventLog.enabled            true
spark.eventLog.dir                hdfs://[redacted]/tmp/applicationHistory
spark.history.fs.logDirectory     hdfs://[redacted]/tmp/applicationHistory
spark.metrics.namespace        ${spark.app.name}
spark.history.fs.cleaner.enabled true
spark.history.fs.cleaner.maxAge  24h
spark.history.fs.cleaner.interval 2h
#spark.dynamicAllocation.enabled false
[manjunath@ip-10-0-0-170 conf]$
```

Prometheus Configs:

- Add Prometheus job in prometheus.yml file, in target need to specify the graphite-exporter URL

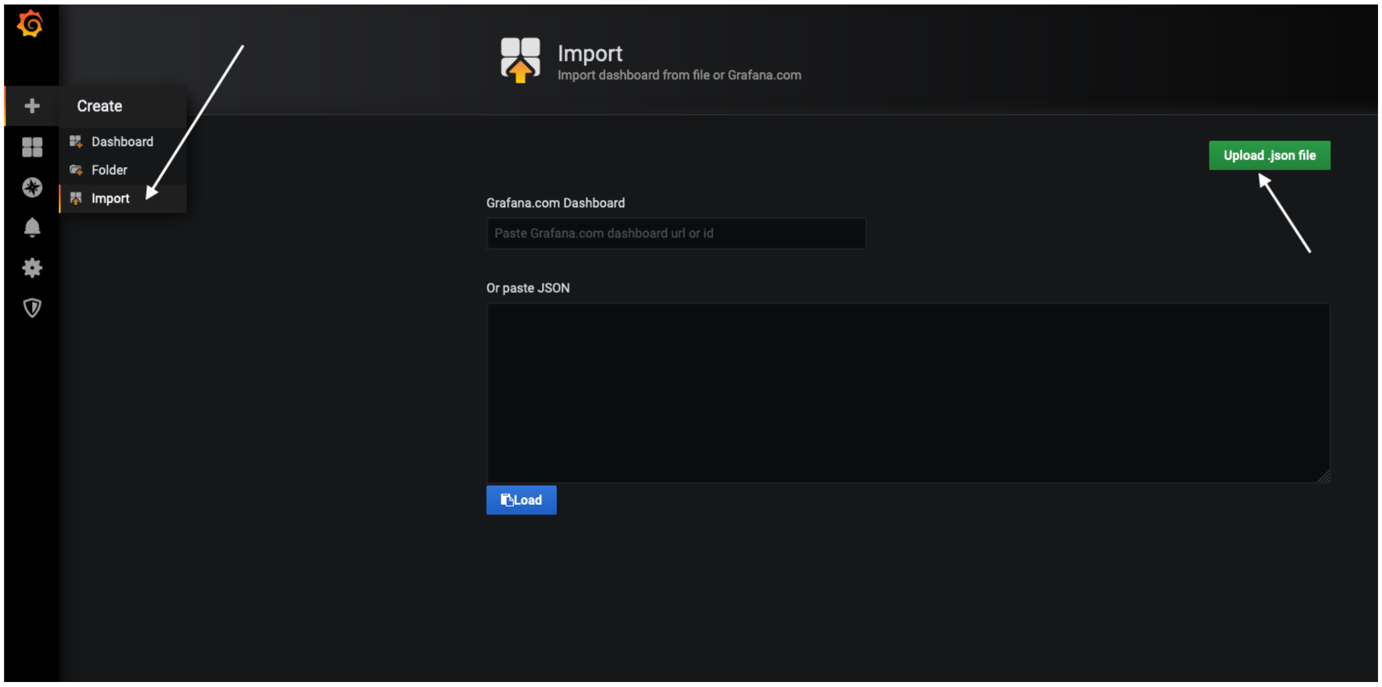


```
  - job_name: spark
    static_configs:
    - targets: ['[redacted]:9108']

  - job_name: milestone_comment_event_gener
```

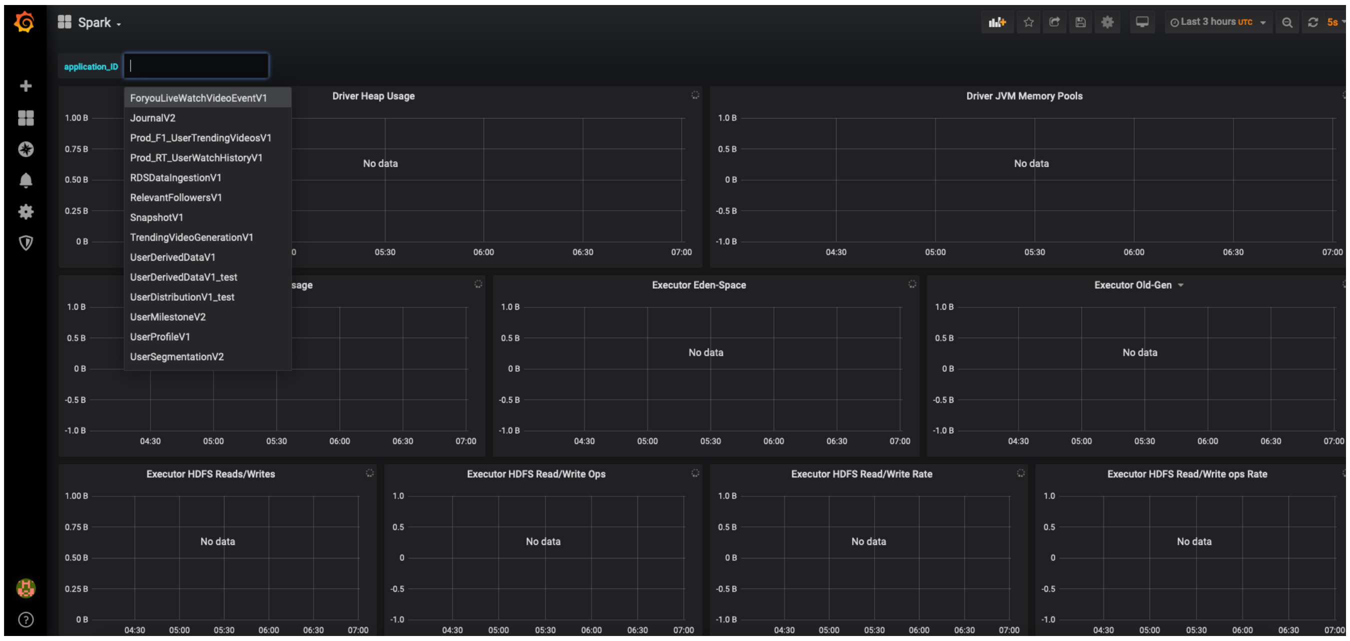
save and restart Prometheus service

- Login to Grafana, create a new Spark dashboard by downloading and importing the attached JSON file

Spark-1619420095961.json



- Click on **Import** , next click on **Upload .json file** upload the downloaded spark json file
- The spark dashboard will look like this


Spark
Last 3 hours UTC
5s
application_ID
ForyouLiveWatchVideoEventV1
JournalV2
Prod_F1_UserTrendingVideosV1
Prod_RT_UserWatchHistoryV1
RDSDataIngestionV1
RelevantFollowersV1
SnapshotV1
TrendingVideoGenerationV1
UserDerivedDataV1
UserDerivedDataV1_test
UserDistributionV1_test
UserMilestoneV2
UserProfileV1
UserSegmentationV2
Driver Heap Usage
Driver JVM Memory Pools
Executor Eden-Space
Executor Old-Gen
Executor HDFS Reads/Writes
Executor HDFS Read/Write Ops
Executor HDFS Read/Write Rate
Executor HDFS Read/Write ops Rate
No data