FortiSIEM - HDFS Storage Guide

Version 5.4.0

FORTINET DOCUMENT LIBRARY
https://docs.fortinet.com

FORTINET VIDEO GUIDE
https://video.fortinet.com

FORTINET BLOG
https://blog.fortinet.com

CUSTOMER SERVICE & SUPPORT
https://support.fortinet.com

FORTINET TRAINING & CERTIFICATION PROGRAM
https://www.fortinet.com/support-and-training/training.html

NSE INSTITUTE
https://training.fortinet.com

FORTIGUARD CENTER
https://fortiguard.com/

END USER LICENSE AGREEMENT
https://www.fortinet.com/doc/legal/EULA.pdf

FEEDBACK
Email: techdoc@fortinet.com

04/15/2020
FortiSIEM 5.4.0 HDFS Storage Guide

TABLE OF CONTENTS

Change Log **4**
Setting Up HDFS for FortiSIEM Event Archive **5**
Overview 5
FortiSIEM and HDFS Interaction 6
Pre-Installation Considerations 7
Set Up the HDFS Cluster 8
Set Up the Spark Cluster 8
Configure FortiSIEM Components on the Spark Master Node 8
Configure FortiSIEM to Use HDFS and Spark 9
Troubleshooting 10

Change Log

Date	Change Description
04/15/2020	Initial version of HDFS storage guide.

Setting Up HDFS for FortiSIEM Event Archive

This document describes how to install and operate HDFS Storage for the FortiSIEM Event Archive solution.

- Overview
- FortiSIEM and HDFS Interaction
- Pre-installation Considerations
- Set Up the HDFS Cluster
- Set Up the Spark Cluster
- Configure FortiSIEM Components on the Spark Master Node
- Configure FortiSIEM to use HDFS and Spark
- Troubleshooting

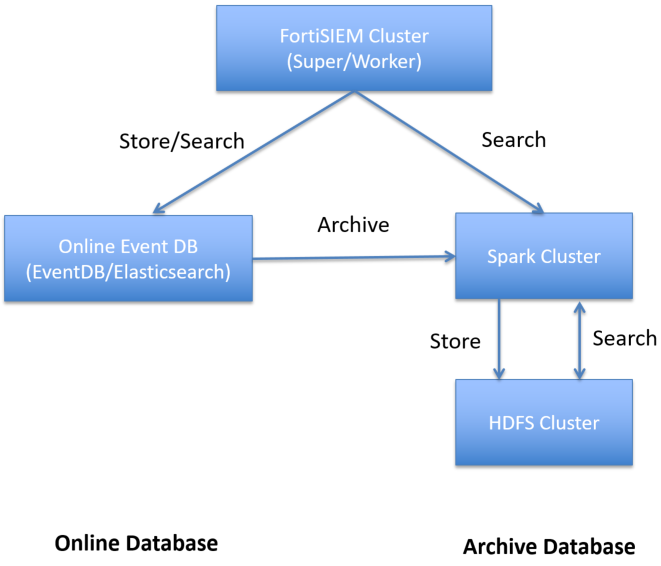
Overview

Received events in FortiSIEM are first stored in an Online event database, which can be either the FortiSIEM EventDB or Elasticsearch. When Online database storage capacity reaches low threshold, events can be archived to an Archive database. Currently, HDFS can be used for the Event Archive, the other choice is the FortiSIEM EventDB on NFS.

Online and Archive databases serve two separate purposes. The Online database is optimized for performance, while the Archive database is optimized for data storage. That is, the Online database provides a faster search, while the Archive database provides a better storage capacity.

Compared to the FortiSIEM EventDB on NFS, the HDFS archive database provides scalable performance and more storage capacity by deploying more cluster nodes.

An HDFS-based database involves deploying an HDFS Cluster and a Spark Cluster. Spark provides the framework for FortiSIEM to communicate with HDFS, both for storing and searching events.



FortiSIEM and HDFS Interaction

The following sections describe the interactions between FortiSIEM and HDFS for searching, archiving, and purging operations.

- Search
- Archive
- Purge

To make search and archive operations work, you must install a FortiSIEM component called HdfsMgr on the Spark Master Node (see Set Up the Spark Cluster).

Search

An HDFS Search works as follows:

1. From the Supervisor node, on the Analytics tab, run a query and set the **Event Source** to **Archive**.
2. The Java Query Server component in the Supervisor node issues the **Search** (via REST API) to the HdfsMgr component residing on the Spark Master Node.
3. Handling of the REST API:
   a. The HdfsMgr translates the query from FortiSIEM Query language to Spark Query language and launches Spark jobs that run on the Spark Cluster.
   b. The HdfsMgr returns the REST API with the JobID and the resulting file path. The Java Query Server uses the JobID to check the query progress.
   c. Spark performs the query by fetching data from the HDFS Cluster and saves the result as a file in HDFS.
4. The Java Query Server reads the Query result (HDFS file location) and returns the result to the GUI.

Archive

An HDFS Archive Operation Works as follows:

1. When Elasticsearch disk utilization reaches the low threshold, the Data Purger module in the Supervisor node issues an Archive command (via the REST API) to the HdfsMgr component residing on the Spark Master Node. The command includes how much data to Archive, as a parameter in REST call.
2. Handling of the REST API:
   a. The HDFS manager launches Spark job.
   b. The Spark job reads the Elasticsearch events, converts events to Parquet format, and inserts them into HDFS.
   c. After the required data is archived, the REST API returns.
3. The Data Purger then deletes the Elasticsearch indices marked for Archive.

Purge

When HDFS disk utilization reaches the low threshold, data must be purged from HDFS. Currently, it is disk space-based only.

1. The Data Purger module in the Supervisor node continuously monitors HDFS disk usage.
2. When HDFS disk usage falls below the low threshold, then the Data Purger module issues a REST API command to the HdfsMgr component residing on the Spark Master Node to purge data. The command includes how much data to purge, as a parameter in the REST call.
3. Handling of the REST API:
   a. The HDFS manager deletes the data.
   b. After the required data is deleted, the REST API returns.

4. The Data Purger logs what was purged from HDFS.

Pre-Installation Considerations

The foillowing sections describe supported versions of HDFS and Spark, and deployment considerations.

- Supported Versions
- Deployment Considerations

Supported Versions

Currently, the following versions of HDFS and Spark are supported:

- HDFS: 2.6.5
- Spark: 2.4.4

Deployment Considerations

HDFS Cluster consists of Name Nodes and Data Nodes. Spark Cluster consists of Master Node and Slave Nodes. The following are recommended.

1. Install Hadoop Name Node and Spark Master Node on separate servers.
2. Co-locate Hadoop Data Node and Spark Slave Node on the same server – this will keep the number of nodes small.
3. FortiSIEM's tested configuration:
   a. Hadoop Name Node and Data Node on one server.
   b. Spark Master Node and Slave Node on one server.
   c. Hadoop Data Node and Spark Slave Node on one server – many instances of such servers.
4. At least 16 vCPU and 32GB RAM on each node with SSD nodes.
5. Make sure all Spark nodes have enough disk apace to store temporary data. By default, Spark nodes use `/tmp`. In FortiSIEM's testing, 70GB of space were needed to archive 1TB of events. You can either increase the size of `/tmp`, or set a different location by editing the `SPARK_HOME/conf/spark-defaults.conf` file as follows:
   `spark.local.dir /`*`your_directory`*
   Without this configuration, Spark jobs may fail with the error `No space left on device` written to the `HdfsMgr.log` file.
6. Allocate sufficient file descriptors for each process in the `/etc/security/limits.conf` file, for example:
   ```
   admin soft nofile 65536
   admin hard nofile 65536
   ```
   Verify the allocations by running the `ulimit -a` command. Without this allocation adjustment, Spark will throw exceptions such as `ava.net.SocketException: Too many open files.`
7. Enable Spark worker application folder cleanup by setting the following environment variable:
   ```
   SPARK_WORKER_OPTS="-Dspark.worker.cleanup.enabled=true -
   Dspark.worker.cleanup.appDataTtl=21600"
   ```
   Without this setting, the size of the `SPARK_HOME/work` folder will become very large.
8. The Kryo serializer does not work properly. Make sure the standard Java serializer is being used. Make sure the following line is either not present or commented out:
   ```
   # spark.serializer org.apache.spark.serializer.KryoSerializer
   ```

Set Up the HDFS Cluster

Follow the instructions at the following URL to set up the HDFS Cluster:

https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/ClusterSetup.html

Set Up the Spark Cluster

After setting up the HDFS Cluster, set up the Spark Cluster. FortiSIEM supports only the Spark Standalone mode.

Follow the instructions at the following URL to set up the Spark Cluster:

https://spark.apache.org/docs/latest/spark-standalone.html

Configure FortiSIEM Components on the Spark Master Node

Follow these steps to install FortiSIEM components on the Spark Master Node.

1. Logon to the Spark Master node as the root user and create a Linux `admin` user.
2. Logon to the Spark Master node as the `admin` user created in the previous step.
3. Create two directories under `/home/admin`: `FortiSIEM` and `FortiSIEM/log`. Make sure that the owner is `admin`.
4. Download the following files from `/opt/phoenix/java/lib` onto the Supervisor node:
   - `phoenix-hdfs-1.0.jar`
   - `phoenix-hdfs-1.0-uber.jar`
5. Copy the files to the `$SPARK_HOME/jars` directory on the Spark Master node. Make sure owner is `admin`.
6. Edit the `log4j.properties` file in the `SPARK_HOME/conf` directory as follows. The purpose of these edits is simply to reduce the logging for HDFS and Spark.

```
# Settings to quiet logs that are too verbose
log4j.logger.org.apache.spark=WARN
log4j.logger.org.apache.hadoop=WARN
log4j.logger.org.spark_project.jetty=WARN
log4j.logger.org.spark_project.jetty.util.component.AbstractLifeCycle=ERROR
log4j.logger.org.apache.parquet=ERROR
log4j.logger.parquet=ERROR
#enable RollingAppender
log4j.appender.R=org.apache.log4j.RollingFileAppender
log4j.appender.R.File=/home/admin/FortiSIEM/log/HdfsMgr.log
log4j.appender.R.MaxFileSize=100MB
log4j.appender.R.MaxBackupIndex=25
log4j.appender.R.layout=org.apache.log4j.PatternLayout
log4j.appender.R.layout.ConversionPattern=%d %p [%t] %c - %m%n
```

7. Create a `checkAndRunHdfsMgr.sh` script under the `FortiSIEM` directory as follows. Make sure owner is `admin`.

```
#!/bin/bash
JAVA_HOME=/opt/java/jdk1.8.0_221
JAR_PATH=/opt/spark/spark-2.4.4-bin-hadoop2.6/jars
```

```
export SPARK_HOME=/opt/spark/spark-2.4.4-bin-hadoop2.6
export HDFSMGR_HOME=/home/admin/FortiSIEM
HdfsMgrPID=$(ps -ef |grep java |grep phoenix-hdfs | awk '{print $2}')
if [ -z "$HdfsMgrPID" ]; then
echo "$(date -Iseconds) checkHdfsMgr: FSM HdfsMgr is not running; starting ..."
exec ${JAVA_HOME}/bin/java -jar ${JAR_PATH}/phoenix-hdfs-1.0.jar &> /dev/null &
else
echo "$(date -Iseconds) checkHdfsMgr: FSM HdfsMgr is running"
fi
```

8. Create a `cron` job to monitor `HdfsMgr`. Run the `checkAndRunHdfsMgr.sh` script every 5 minutes, for example:
   `*/5 **** /home/admin/FortiSIEM/checkAndRunHdfsMgr.sh`

Configure FortiSIEM to Use HDFS and Spark

Once the HDFS and Spark clusters have been set up, follow these steps to allow FortiSIEM to communicate with HDFS and Spark.

- Configure the Archive
- Search Archived Events
- Display Archived Events

Configure the Archive

Follow these steps to configure the archive on FortiSIEM:

1. Go to **ADMIN > Storage > Archive**.
2. Select **HDFS**.
3. Enter a value for the **Spark Master Node IP/Host** and **Port** (the default is 7077).
4. Enter a value for the **Hadoop Name Node IP/Host** and **Port** (the default is 9000).
5. Click **Test**.
   - If the test succeeds, then click **Save**.
   - If the test fails, then check the values for the IP/Host parameters defined in steps 3 and 4.

Note that the Archive will be activated when the Online Elasticsearch database is full. This setting is defined in **ADMIN > Settings > Archive**.

Search Archived Events

To search Archived events, follow the same steps as searching Online events, except set **Event Source** to **Archive** in the **Filters** and the **Time Range** dialog boxes.

Display Archived Events

To display archived event data, go to **ADMIN > Settings > Database > Archive**. For more information, see Viewing Archive Event Data.

Troubleshooting

- Make Sure HdfsMgr is Running on the Spark Master Node
- Log Locations
- Spark Master Node Cluster Health Web GUI
- Spark Master Node Web GUI
- HDFS Metrics Web GUI
- A Troubleshooting Example

Make Sure HdfsMgr is Running on the Spark Master Node

1. SSH to the Spark Master node as the `admin` user.
2. Run the `JPS` command to see if the `phoenix-hdfs-1.0.jar` process is running: for example:

```
[admin@Server ~]$jps
8882 NodeManager
8772 DataNode
10164 phoenix-hdfs-1.0.jar
9064 Worker
8969 Master
10825 Jps
```

Log Locations

- HdfsMgr Logs on the Spark Master Node
- Spark Logs in the Master Node and Worker Node
- HDFS Logs in Name Node and Data Node
- Data Purger Log Location
- Java Query Server Log Location

HdfsMgr Logs on the Spark Master Node

You can find the HdfsMgr logs here:

```
HDFSMGR_HOME/log/HdfsMgr.log
```

Spark Logs in the Master Node and Worker Node

You can find the Spark Master node logs here:

```
$SPARK_HOME/logs/spark-admin-org.apache.spark.deploy.master.Master-1-Elastic1.out
```

You can find the Spark Worker node logs here:

```
$SPARK_HOME/logs/spark-admin-org.apache.spark.deploy.worker.Worker-1-Elastic1.out
```

HDFS Logs in Name Node and Data Node

You can find the HDFS Name node logs here:

```
$HADOOP_HOME/logs/hadoop-admin-namenode-HadoopServer.log
$HADOOP_HOME/logs/hadoop-admin-secondarynamenode-HadoopServer.log
```

The HDFS Data Node logs are located here:

```
$HADOOP_HOME/logs/hadoop-admin-datanode- HadoopServer.log
```

Data Purger Log Location

You can find the Data Purger logs in the `/opt/phoenix/log/phoenix.log` file on the Supervisor node. Search for the `phDataPurger` module, for example:

```
grep phDataPurger phoenix.log
```

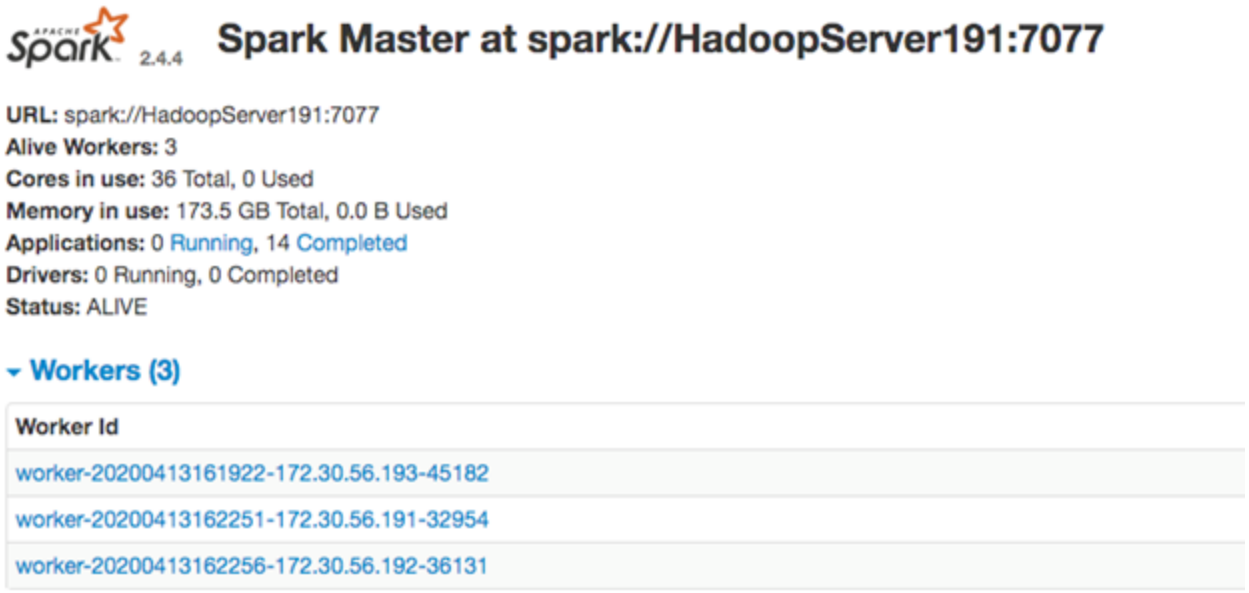
Java Query Server Log Location

You can find the Java Query logs here:

`/opt/phoenix/log/javaQueryServer.log`

Spark Master Node Cluster Health Web GUI

To see the Spark cluster health, go to `http://SparkMaster:8080/`, for example:



Spark Master Node Web GUI

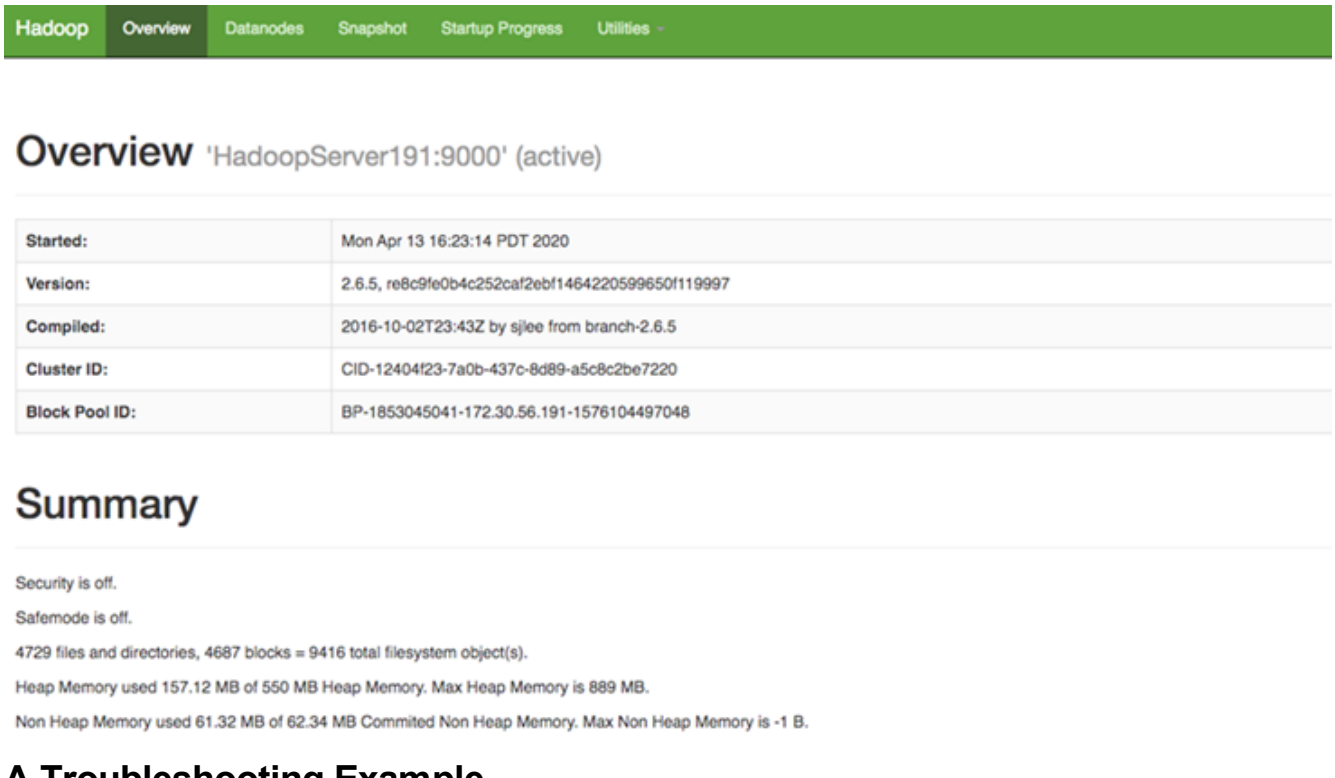
Every Spark context launches a Web GUI that displays useful information about the application. This includes:

- A list of scheduler stages and tasks
- A summary of RDD sizes and memory usage
- Environmental information
- Information about the running executors

You can access this interface simply by opening `http://<`*driver-node*`>:4040` in a Web browser.

HDFS Metrics Web GUI

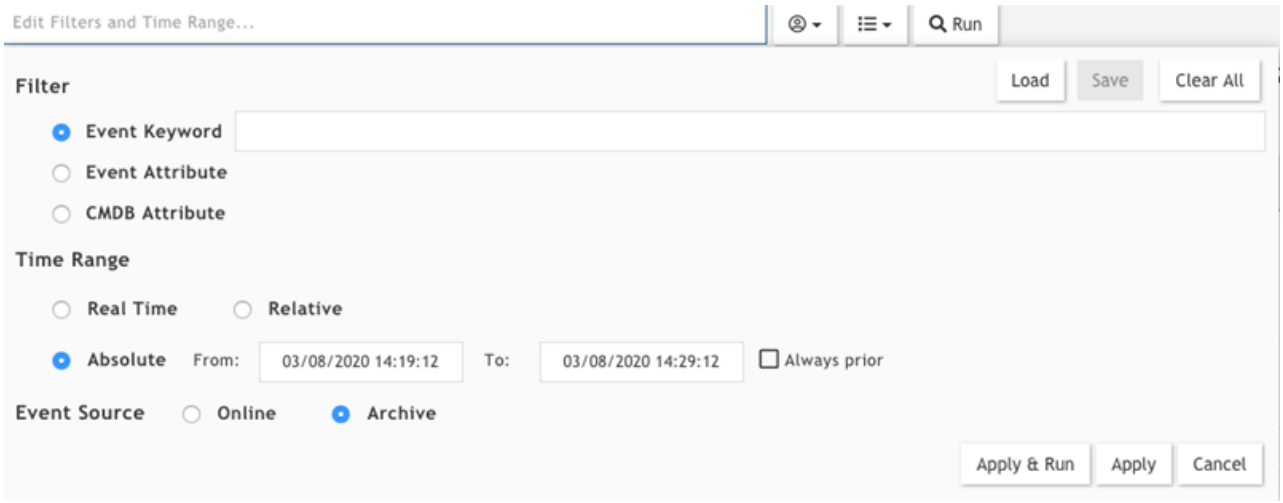
You can monitor HDFS Metrics through the GUI. For example, enter the URL `http://<`*HadoopNameNode*`>:50070`:



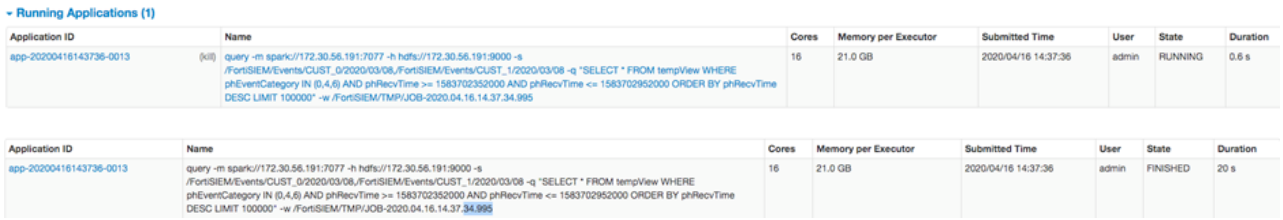
A Troubleshooting Example

The following steps describe how to troubleshoot a Spark job.

1. Run an "Archive query" from the FortiSIEM **Analytics** tab:



2. Open the Spark UI (`http://SparkMaster:8080/`) and you will see that one Spark job has been created. The state will be RUNNING, and then FINISHED.



3. You can also find log details in the `HDFSMGR_HOME/log/HdfsMgr.log` file. Search for the Job ID, in this case, `34.995`. If the Spark job failed, you can find the reason from the logs.

```
2020-04-16 14:37:34,995 INFO [qtp1334729950-17] com.accelops.hdfs.mgr.RestManager -
(34.995) launching: job=command="query -m spark://172.30.56.191:7077 -h
hdfs://172.30.56.191:9000 -s /FortiSIEM/Events/CUST_0/2020/03/08,/FortiSIEM/Events/CUST_
1/2020/03/08 -q "SELECT * FROM tempView WHERE phEventCategory IN (0,4,6) AND  phRecvTime >=
1583702352000 AND phRecvTime <= 1583702952000  ORDER BY phRecvTime DESC LIMIT 100000"",RM
(scheme/core/max/mem)=HDFSMGR/16/32/21530,file=/FortiSIEM/TMP/JOB-
2020.04.16.14.37.34.995,result=UNKNOWN,failReason=,lastSet=2020-04-16 14:37:34.995


2020-04-16 14:37:34,996 INFO [pool-1-thread-14] com.accelops.hdfs.mgr.DoLaunch - (34.995)
start: resource="command="query -m spark://172.30.56.191:7077 -h hdfs://172.30.56.191:9000
-s /FortiSIEM/Events/CUST_0/2020/03/08,/FortiSIEM/Events/CUST_1/2020/03/08 -q "SELECT *
FROM tempView WHERE phEventCategory IN (0,4,6) AND  phRecvTime >= 1583702352000 AND
phRecvTime <= 1583702952000  ORDER BY phRecvTime DESC LIMIT 100000"",RM
(scheme/core/max/mem)=HDFSMGR/16/32/21530,file=/FortiSIEM/TMP/JOB-
2020.04.16.14.37.34.995,result=UNKNOWN,failReason=,lastSet=2020-04-16 14:37:34.995"


2020-04-16 14:37:36,044 INFO [main] com.accelops.hdfs.server.QueryServer - (34.995)
initServerOption: srcFile=/FortiSIEM/Events/CUST_0/2020/03/08,/FortiSIEM/Events/CUST_
1/2020/03/08,sql="SELECT * FROM tempView WHERE phEventCategory IN (0,4,6) AND  phRecvTime
>= 1583702352000 AND phRecvTime <= 1583702952000  ORDER BY phRecvTime DESC LIMIT 100000"


2020-04-16 14:37:37,032 INFO [pool-1-thread-14] com.accelops.hdfs.mgr.DoLaunch - (34.995)
application state=RUNNING


2020-04-16 14:37:56,351 INFO [Thread-17] com.accelops.hdfs.server.run.RunQueryServer -
(34.995)  sql results count=83460


2020-04-16 14:37:56,581 INFO [pool-1-thread-14] com.accelops.hdfs.mgr.DoLaunch - (34.995)
state changed from=RUNNING,to=FINISHED,isFinal=true


2020-04-16 14:37:56,604 INFO [Thread-17] com.accelops.hdfs.server.run.RunSparkJob -
(34.995) server: DONE


2020-04-16 14:37:57,022 INFO [main] com.accelops.hdfs.server.HdfsMgrServer - (34.995)
server done
```