

Reporting/Analytics

Can I create Viewpoint reports without having to set up Spark, HDFS, and Kafka?

The following explains how to configure Viewpoint to generate reports from the INSIGHT mongo database. This is in contrast to setting up Viewpoint to use the REPORTING mongo database, which uses a database populated by an Apache Spark/HDFS/Kafka stack. The main reason to use INSIGHT instead of REPORTING is to avoid the additional hardware and complexity of needing to set up a full Apache Spark, HDFS, Kafka cluster in order to get reports. The drawback for INSIGHT reports is that all of the out-of-the-box Viewpoint reports are configured to use the REPORTING database. So you have to write your own reports if you use INSIGHT. This requires some knowledge of SQL. However, that said, most people agree that it is still much easier to set up Viewpoint to use INSIGHT for reports.

One thing that makes INSIGHT easier is that the Insight component will have already been set up if Viewpoint is running. This is because Viewpoint requires Insight. Note that it is possible to configure Viewpoint to use both INSIGHT and REPORTING at the same time.

To configure Viewpoint to be able to run reports out of the INSIGHT database you will follow a few selected steps in "3.13 Installing the Reporting Framework" of the installation docs (For Redhat 7/Cent 7 see <http://docs.adaptivecomputing.com/9-1-3/installGuide/RH7/installRH7.htm#topics/hpcSuiteInstall/rpm/installing/installingReporting.htm>) Most of the steps will be skipped since one of the main points of using INSIGHT is to not have to set up Spark, Hadoop, and Kafka. However you will still need to install Drill and RWS as explained below.

The steps to follow are below. Note that many of them refer back to the installation docs.

* Open Necessary Ports *

You will need to open port 8047 in the firewall on the Insight host. This port is used by Apache Drill, which will be installed later.

```
#####  
[root]# firewall-cmd --add-port=8047/tcp --permanent  
[root]# firewall-cmd --reload  
#####
```

* Adjust Security Enhanced Linux *

For Red Hat-based systems where Security Enhanced Linux (SELinux) is enforced, you may need to customize SELinux to allow Reporting Web Services (RWS) to perform operations like making network connections, reading the RWS configuration files, copying RWS plugin jar files, and writing to the RWS log files. Follow the steps in section 3.13.3 of the installation docs (For Redhat 7/Cent 7 see <http://docs.adaptivecomputing.com/9-1-3/installGuide/RH7/installRH7.htm#topics/hpcSuiteInstall/rpm>

Reporting/Analytics

/installing/installingReporting.htm#selinux) Either customize as explained in that section or disable SELinux.

* Configure Insight NOT to Send Messages to Kafka *

If you are not using the Apache Spark/HDFS/Kafka stack you will want to make sure Insight is not trying to send messages to Kafka. If Kafka is not there listening for messages this will obviously cause errors in Insight. On the Insight host, do the following:

```
#####  
[root]# vi /opt/insight/etc/config.groovy  
...  
kafka.enabled=false  
#####
```

Restart Insight.

```
#####  
[root]# systemctl restart insight  
#####
```

* Install Zookeeper *

You will need to install Zookeeper to run Drill. Zookeeper is bundled with Kafka. So even though you are not going to run the Kafka service you still need to install Kafka to get Zookeeper. Follow steps 1 - 3 of "3.13.7.A Set Up the Kafka Master" of <http://docs.adaptivecomputing.com/9-1-3/installGuide/RH7/installRH7.htm#topics/hpcSuite/install/rpm/installing/installingReporting.htm#kafkamaster>

The steps you need to follow (1 - 3)

```
#####  
[root]# yum install kafka  
#####
```

Start and enable Zookeeper

```
#####  
[root]# systemctl enable zookeeper  
[root]# systemctl start zookeeper  
#####
```

Do not proceed to step 4 though. There is no need to start the Kafka service when using INSIGHT for reports.

* Install and Configure MongoDB *

Even though you will not be using the REPORTING mongo database the database does need to exist because RWS will expect it and will fail to start if it cannot be

Reporting/Analytics

found. Go to Section 3.13.8 of the installation docs (For Redhat 7/Cent 7 see <http://docs.adaptivecomputing.com/9-1-3/installGuide/RH7/installRH7.htm#topics/hpcSuiteInstall/rpm/installing/installingReporting.htm#mongo>) and follow step 4.

Since the reporting mongo database will not be used you can skip steps 1 - 3 and just add it to the same mongo instance where the insight database is running. So you can jump ahead to step 4, which explain how to create a database named "reporting" and how to set up user on this database called "reporting_user".

When you are finished you should be able to access this database using the mongo command line. For example:

```
#####  
[root]# mongo -u reporting_user -p 'changeme!' reporting  
MongoDB shell version: 3.2.22  
connecting to: reporting  
> exit  
bye  
#####
```

* Install and Configure Apache Drill *

Follow the steps in section 3.13.9 "Install and Configure Apache Drill" of the installation docs subject to the following modifications. (For Redhat 7/Cent 7 see <http://docs.adaptivecomputing.com/9-1-3/installGuide/RH7/installRH7.htm#topics/hpcSuiteInstall/rpm/installing/installingReporting.htm#drill>) These steps explain how to install Drill, how to set up Drill authentication using PAM, and how to write a sample SQL query. Here the changes to the documented steps.

Step 8.a. Install Jpam

If the documented step does not work try this

```
#####  
[root]# mkdir /opt/pam  
[root]# cd /opt/pam  
[root]# wget http://downloads.sourceforge.net/project/jpam/jpam/jpam-1.1/JPam-Linux_amd64-1.1.tgz  
[root]# tar zxvf JPam-Linux_amd64-1.1.tgz  
[root]# cp JPam-1.1/libjpam.so /opt/pam  
[root]# rm JPam-1.1/ JPam-Linux_amd64-1.1.tgz  
#####
```

When you are done there should be a libjpam.so in /opt/pam

```
#####  
[root]# file /opt/pam/libjpam.so
```

Reporting/Analytics

/opt/pam/libjpam.so: ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked, not stripped

#####

Then you can proceed with step 8.b, 8.c, ...

* Adding a connection to the Insight Database to Apache Drill *

First determine the host, port, username and password required to connect to the insight mongo database. These are found in the insight configuration file. For example

#####

```
[root]# vim /opt/insight/etc/config.groovy
```

...

```
mongo.username="insight_user"
```

```
mongo.password="changeme!"
```

```
mongo.host="localhost"
```

```
mongo.port=27017
```

#####

Now open the drill web UI in a web browser and login in as the drilladmin user. Note that the drilladmin user is a linux user on the box where drill is running. By default Drill runs on port 8047. For example:

```
http://localhost:8047
```

Click on the "Storage" tab at the top. A page will appear with the current Storage Plugins. At the bottom will be a text field for "Storage Name". Enter "mongo2". Click the create button next to it. A configuration page will appear with a large text area. In this text area enter the following, substituting <user>, <password>, <host>, and <port> with the values from /opt/insight/etc/config.groovy as explained above.

#####

```
{
  "type": "mongo",
  "connection": "mongodb://<user>:<password>@<host>:<port>/insight",
  "enabled": true
}
```

#####

For example:

#####

```
{
  "type": "mongo",
  "connection": "mongodb://insight_user:changeme!@localhost:27017/insight",
```

Reporting/Analytics

```
"enabled": true
```

```
}
```

```
#####
```

Next test that this new connection worked. Click the "Query" in the menu at the top of the Drill page. Enter the following SQL. Note that this will return 5 entries from the workload_view collection in the insight mongo database.

```
#####
```

```
SELECT * FROM mongo2.insight.workload_view LIMIT 5;
```

```
#####
```

Click Submit. This should display a table showing five rows representing five jobs that have run on the cluster. If the table appears you have set up the connection to the insight mongo database successfully. Remember that the insight mongo connection is called "mongo2". This will be helpful when you create a report.

Note that in some environments the password must be URL encoded. In others no such URL encoding is needed. If you have an error connecting try URL encoding the password at a site like <https://www.urlencoder.org/> For example:

Before URL encoding: changeme!

After URL encoding: changeme%21

* Install and Configure Reporting Web Services (RWS) *

Follow the steps in 3.13.11 "Install and Configure Reporting Web Services" of the installation docs. (For Redhat 7/Cent 7 see <http://docs.adaptivecomputing.com/9-1-3/installGuide/RH7/installRH7.htm#topics/hpcSuiteInstall/rpm/installing/installingReporting.htm>)

This will install RWS as a web application in tomcat. You should already have tomcat installed since tomcat is required to run MWS and MWS is required to run Viewpoint.

Step 2 "Configure Reporting Web Services"

The important fields to configure are the ones that tell RWS how to connect to Drill and how to find the unused reporting Mongo database you set up above. For example:

```
#####
```

```
[root]# vi /opt/reporting-web-services/etc/application.properties
```

```
...
```

```
reporting.rest.drill.hostname=localhost
```

```
...
```

```
reporting.rest.drill.username=drilluser
```

```
reporting.rest.drill.password=changeme!
```

Reporting/Analytics

```
...
spring.data.mongodb.host=localhost
spring.data.mongodb.port=27017
spring.data.mongodb.database=reporting
spring.data.mongodb.username=reporting_user
spring.data.mongodb.password=changeme!
#####
```

Since you aren't configuring Spark/HDFS/Kafka just leave the default of "localhost" for <spark_master_host>, <hdfs_name_node_host>, and <reporting_mongodb_database_host>. Since Spark isn't running this won't do anything. Likewise it doesn't matter what you configure for the Spark driver and executor in application.properties. Just leave the defaults in.

Just be sure to follow the steps to restart tomcat

```
#####
[root]# systemctl restart tomcat
#####
```

and verify reporting web services is running

```
#####
[root]# curl -X GET -v localhost:8080/rws/ping
< HTTP/1.1 200 OK
#####
```

* Connect Viewpoint to Reporting *

Follow the steps in section 3.13.13 "Connect Viewpoint to Reporting" of the installation docs subject to the following modifications. (For Redhat 7/Cent 7 see <http://docs.adaptivecomputing.com/9-1-3/installGuide/RH7/installRH7.htm#topics/hpcSuiteInstall/rpm/installing/installingReporting.htm#connect>)

You can skip step 8 that says in REPORTING > Aggregated View the status of the processing application should be RUNNING. Since Spark/HDFS/Kafka aren't present the status of the processing application will be NOT STARTED. Also the "Node State/Outage" report requires Spark/HDFS/Kafka so it will not work. The following step will explain how to create a sample report so you can verify this step.

Also note that the other reports that show up in the REPORTING > REPORT tab will not work since they are configured to use the REPORTING database that requires Spark/HDFS/Kafka. It might be a good idea to export these to files and remove them so they don't show up and confuse people. If you later decide to set up the Spark/HDFS/Kafka cluster you can use the files you just exported to restore these reports. That is of course optional.

* Creating a report that uses the Insight database *

Reporting/Analytics

You should be logged into Viewpoint as an admin. Go to REPORTING > REPORTS and click CREATE REPORT. For the name you can put "A Test Report" or whatever else you choose. Click SWITCH TO ADVANCED. A text area will appear that will allow you to enter SQL. You can enter whatever you like but here is some sample SQL that will query for up to a certain number (specified by LIMIT) of jobs in the insight workload_view collection that have completed between a start date and an end date.

Notice that the fields that are prefixed with dollar signs (e.g. \$start_date, \$end_date, \$order_type, and \$limit) get filled in when you run the report based on the settings the person running the report selects. More on this later.

```
#####  
SELECT  
  t.user_name,  
  t.completion_datetime  
FROM mongo2.insight.workload_view t  
WHERE  
  t.completion_datetime > TO_TIMESTAMP($start_date/1000) AND  
  t.completion_datetime < TO_TIMESTAMP($end_date/1000)  
ORDER BY t.completion_datetime $order_type  
LIMIT $limit
```

```
#####
```

In the Variables Default Values section select the default values for \$start_date, \$end_date, \$order_type, and \$limit. For "Interval" select something large, like a Month or a Quarter so that the time range will be large enough that some jobs show up. This will populate \$start_date and \$end_date. For Limit pick something small, like 10 so that the report doesn't take forever to load. This will affect \$limit. Likewise selecting Sort By will populate \$order_type in the SQL above.

Notice the TO_TIMESTAMP(\$start_date/1000). \$start_date and \$end_date get populated based on the date interval selected when you run the report. If Viewpoint changes \$start_date and \$end_date to unix epoch timestamps in MILLISECONDS (e.g. 1570566960000). Drill's TO_TIMESTAMP function changes a unix epoch timestamp in SECONDS to a timestamp object that can be used in "where" clauses with Mongo dates. So we have to divide by 1000 to turn the epoch time in MILLISECONDS to an epoch time in SECONDS.

In the Layout portion of the report editor select table for this report. This can be changed of course but using table initially is the easiest way to verify that the jobs included in the report are the ones you expect.

Click GENERATE PREVIEW. A table should appear showing the results of this query. Click SAVE AND CLOSE.

Reporting/Analytics

* Testing the report *

Now test the report. Go to REPORTING > REPORTS. The new report should be there. Click the name of the report and the report should appear. Clicking the gear icon next to the name of the report will allow you to change the time interval, limit, and sorting direction.

* Sample Report - Dedicated Processor Seconds Per User*

A sample report that uses the Insight database to show dedicated processor seconds per user is attached to this article. First download the file.

http://files.adaptivecomputing.com/reporting/INSIGHT_DB_Dedicated_Processor_Seconds_Per_User.report

Save the contents into a text file named

INSIGHT_DB_Dedicated_Processor_Seconds_Per_User.report

Then log into Viewpoint as an admin, go to REPORTING > REPORTS and click IMPORT.

You should then see a report in the REPORTS table named INSIGHT_DB Dedicated Processor Seconds Per User. If you hover over the table row for this report you will see an "Edit" menu. Click on it and you can see the SQL and other configuration options used to create the report. This can be helpful as a reference for creating other reports.

Unique solution ID: #1220

Author: Nate Seeley

Last update: 2020-12-03 19:56