Steps to Run Hadoop Connect on Splunk 8

Note: In Splunk 8 Advanced XML stopped working. In addition, Python version 3 is the recommended version. These two elements caused the Hadoop Connect App to stop working. Below are the steps to Run Hadoop Connect on Splunk 8

Manually Installing and configuring Hadoop and Java

Install and setup Hadoop and Java on the Splunk Search Head
This video might be useful: https://www.splunk.com/view/SP-CAAAHBZ

Before we install the Hadoop Connect App, make sure you can connect to Hadoop from the Splunk Search Head Command Line.
The user to run the Hadoop commands should be the same user that Installed Splunk.

For example, as user ' root '

```
[root@localhost HadoopConnect]# /opt/hadoop-3.2.0/bin/hadoop fs -ls hdfs://192.168.56.254:8020/user/root
drwxr-xr-x   - root supergroup          0 2020-06-09 16:41 hdfs://192.168.56.254:8020/user/root/connectexport
drwxr-xr-x   - root supergroup          0 2020-06-10 13:49 hdfs://192.168.56.254:8020/user/root/data
```

Manually Installing and configuring the Hadoop Connect App

Go to Splunk App directory /opt/splunk/etc/apps
Extract this modified version of the Hadoop Connect App to Splunk 8

```
[root@localhost apps]# tar xvzf HadoopConnect-*.tgz
[root@localhost apps]# cd HadoopConnect/
[root@localhost HadoopConnect]# ls
appserver  bin  default  lib  metadata  README  README.txt
```

Configuration - The easy way:
If you already have the Hadoop Connect App working on Splunk version 7, simply copy the ‘ local ‘ directory to the new Hadoop Connect App on Splunk version 8

Configuration - The longer way:

Without Kerberos and without High Availability Configurations:
****

For background you can read this documentation:
https://docs.splunk.com/Documentation/HadoopConnect/latest/DeployHadoopConnect/Configurationfilereference#clusters.conf

Configuring the App to connect to Hadoop

```
[root@localhost HadoopConnect]# mkdir local
[root@localhost HadoopConnect]# cd local/
```

[root@localhost local]# vi clusters.conf
[192.168.56.254:8020]
hadoop_home = /opt/hadoop-3.2.0
java_home = /usr
uri = hdfs://192.168.56.254:8020

[root@localhost local]# mkdir clusters
[root@localhost local]# cd clusters
[root@localhost clusters]# mkdir 192.168.56.254_8020

Note = Directory name should be the same as stanza name from clusters.conf file (" : " becomes " _ ")

[root@localhost clusters]# cd 192.168.56.254_8020/
Copy Hadoop core-site.xml from the Hadoop Home configuration directory to HadoopConnect/local/clusters/<name of Hadoop cluster>/
For example,
cp /opt/hadoop-3.2.0/etc/hadoop/core-site.xml /opt/splunk/etc/apps/HadoopConnect/local/clusters/192.168.56.254_8020/

After you are done it should look similar to this structure
[root@localhost HadoopConnect]# ls -R local/
local/:
clusters clusters.conf

local/clusters:
192.168.56.254_8020

local/clusters/192.168.56.254_8020:
core-site.xml

Without Kerberos but with Name Node High Availability Configurations:
****

Modify clusters.conf to include these flags
[nameservice1]
ha = 1
hadoop_home = /opt/hadoop/hadoop-2.5.0-cdh5.2.1
java_home = /opt/hadoop/jdk1.7.0_71/
uri = hdfs://nameservice1

[root@localhost local]# mkdir clusters
[root@localhost local]# cd clusters
[root@localhost clusters]# mkdir nameservice1
[root@localhost clusters]# cd nameservice1

Copy Hadoop core-site.xml and hdfs-site.xml from the Hadoop Home configuration directory to HadoopConnect/local/clusters/<name of Hadoop cluster>/
For example,
cp /opt/hadoop-3.2.0/etc/hadoop/core-site.xml /opt/splunk/etc/apps/HadoopConnect/local/clusters/nameservice1
cp /opt/hadoop-3.2.0/etc/hadoop/hdfs-site.xml /opt/splunk/etc/apps/HadoopConnect/local/clusters/nameservice1

Kerberos and Hadoop Name Node HA configurations
****

For background you can watch this video and documentations:
https://docs.splunk.com/Documentation/HadoopConnect/latest/DeployHadoopConnect/Kerberosclientutilities

And the Kerberos flags setup here:
https://docs.splunk.com/Documentation/HadoopConnect/latest/DeployHadoopConnect/Configurationfilereference

Step 1 – setup clusters connectivity

Modify clusters.conf to include these Kerberos and Name Node HA flags
[nameservice1]
ha = 1
hadoop_home = /opt/hadoop/hadoop-2.5.0-cdh5.2.1
java_home = /opt/hadoop/jdk1.7.0_71/
uri = hdfs://nameservice1
kerberos_principal = example
kerberos_service_principal = hdfs/_HOST@EXAMPLE.COM

[root@localhost local]# mkdir clusters
[root@localhost local]# cd clusters
[root@localhost clusters]# mkdir nameservice1
[root@localhost clusters]# cd nameservice1

Copy Hadoop core-site.xml and hdfs-site.xml from the Hadoop Home configuration directory to HadoopConnect/local/clusters/<name of Hadoop cluster>/
For example,
cp /opt/hadoop-3.2.0/etc/hadoop/core-site.xml /opt/splunk/etc/apps/HadoopConnect/local/clusters/nameservice1
cp /opt/hadoop-3.2.0/etc/hadoop/hdfs-site.xml /opt/splunk/etc/apps/HadoopConnect/local/clusters/nameservice1

Step 2 - Setup Principals keytab authentication

Modify principals.conf to include the Kerberos Principals directory
For example,
[root@localhost local]# cd /opt/splunk/etc/apps/HadoopConnect/local
vi principals.conf
[user__secure.example.com@REALM.EXAMPLE.COM]
mkdir principals
cd principals
mkdir user__secure.example.com@REALM.EXAMPLE.COM
cd user__secure.example.com@REALM.EXAMPLE.COM/

Add Hadoop Kerberos Keytab file (or cache file) from the Hadoop and Kerberos infrastructure to HadoopConnect/local/principals/<name of principals>/
For example,
[root@localhost local]# cd
HadoopConnect/local/principals/user__secure.example.com\@REALM.EXAMPLE.COM/

[root@localhost local]# ls
principal.keytab

Restart Splunk Search Head
****
[root@localhost HadoopConnect]# /opt/splunk/bin/splunk restart

Run Search Commands to Import files from Hadoop, Export files to Hadoop, or List Hadoop directories

In the Splunk UI, since Splunk 8 does not support Advance XML, feel free to 'Hide' the app
Go to Manage Apps -> Hadoop Connect -> Edit properties -> Visible = No

Read Hadoop files or List Hadoop directories
****
Test Reading Hadoop using the splunk hdfs search commands
This link might be useful: https://www.splunk.com/en_us/blog/tips-and-tricks/connecting-splunk-and-hadoop.html

For example
*| hdfs read "hdfs://192.168.56.254:8020/user/root/data/Hunkdata.json.gz" | table **
or
| hdfs lsr "hdfs://192.168.56.254:8020/user/"

Import Hadoop files and Index them in Splunk
****
Test Importing data from Hadoop into Splunk
Since this feature has not been affected by Splunk 8 issues, the documentation is still the same as older versions.
https://docs.splunk.com/Documentation/HadoopConnect/latest/DeployHadoopConnect/ImportfromHDFS

After installing the App on Splunk 8, go to Splunk Settings -> Data Inputs -> HDFS

Export Splunk results into Hadoop files
****
Test Exporting from Splunk to Hadoop
Note: Exporting uses Index Time and Not Event Time

These links might be helpful as background:

- Background = https://docs.splunk.com/Documentation/HadoopConnect/1.2.5/DeployHadoopConnect/ExporttoHDFS

- RunExport command = https://docs.splunk.com/Documentation/HadoopConnect/1.2.5/DeployHadoopConnect/Searchcommandreference#runexport
- Export.conf file = https://docs.splunk.com/Documentation/HadoopConnect/1.2.5/DeployHadoopConnect/Configurationfilereference#export.conf

Create the export job with the partition, Splunk search, hdfs uri, and base hdfs location
[root@localhost local]# vi export.conf
[trial_export1]
base_path = /user/root/export
partition_fields = date,hour
search = index=splunkaccesscombine sourcetype=_json
uri = hdfs://192.168.56.254:8020

From the Splunk Search command run the 'runexport' command with forcerun=1.
| runexport name=trial_export1 forcerun=1 roll_size=128
Or, running the export on 2 Indexers (*parallel_searches=2)*, exporting JSON files (*format=json fields=result.price,result._raw)*
| runexport name=trial_export1 forcerun=1 roll_size=63 parallel_searches=2 format=json fields=result.price,result._raw
Or Starttime on June 10, 2015 (*starttime=1433941241)* and Maxspan for 3 years (*maxspan=94670856)*
| runexport name=trial_export1 forcerun=1 starttime=1433941241 maxspan=94670856

Schedule the export job
For example, we can schedule the runexport to run once an hour by removing the forcerun=1
| runexport name=trial_export1 starttime=1433941241 maxspan=94670856

- Run it, and you will see the error "*runexport is only meant to be ran by scheduled searches*"

On the upper right click on " Save As -> Report -> Schedule -> Schedule Report "
This document can help when scheduling a search:
https://docs.splunk.com/Documentation/Splunk/latest/Report/Schedulereports

After the runexport search command is complete, you should be able to see the data in HDFS.
For example, with *partition_fields = date,hour*
[root@localhost hadoop-3.2.0]# bin/hadoop fs -ls -R /user/root/export
drwxr-xr-x - root supergroup 0 2020-07-14 17:45 /user/root/export/20150608
drwxr-xr-x - root supergroup 0 2020-07-14 17:45 /user/root/export/20150608/20
-rw-r--r-- 3 root supergroup 389717 2020-07-14 17:45
/user/root/export/20150608/20/627ac154f1f4cbf80454ea48d98a4018_1433941241_1528612097_24_0.raw.gz

More information on the export job stats can be found in the export.conf file
The App logs its progress within the following *status* attributes in the export.conf file. You cannot edit these *status* attributes, and they might change rapidly based on the status of the export job. The document for these flags can be found here:
https://docs.splunk.com/Documentation/HadoopConnect/1.2.5/DeployHadoopConnect/Configurationfilereference#export.conf
For example,
[root@localhost ~]# more /opt/splunk/etc/apps/HadoopConnect/local/export.conf
[trial_export1]
base_path = /user/root/export

partition_fields = date,hour
search = index=splunkaccesscombine sourcetype=_json
uri = hdfs://192.168.56.254:8020
status = done
status.jobs = 1
status.jobs.earliest = 1433941241
status.jobs.endtime = 1594846891.76
status.jobs.errors =
status.jobs.latest = 1528612097
status.jobs.progress = 1.0
status.jobs.psid = scheduler__admin__search__RMD55fef665a58d7c352_at_1594846800_12
status.jobs.runtime = 50.995
status.jobs.sids = 1594846814.1273
status.jobs.starttime = 1594846801.19
status.earliest = 1433941241
status.latest = 1528612097
status.load = 0.0000,0.0000,0.0116,0.0127,0.0123,0.0000,0.0032,0.0007,0.0126,0.0123

Tips:

- To find the Index Time represented as epoch time for your data (*starttime* flag) run a similar search to this: *index=splunkaccesscombine | eval indextime = _indextime | table indextime | head 1*
- To use all of your indexers for maximum performance during export use this setting: *parallel_searches=max*
- Use *forcerun=1* only for testing and not for Schedule Searches

Debug for performance or errors

If you need to debug Hadoop Connect, you can search Splunk internal Indexer _internal.
For example,

Summary*:*
*index=_internal splunk_server=local source=*export_metrics.log group=transfer | stats sum(eval(local_KB/(1024*1024))) AS lgb, sum(eval(hdfs_KB/(1024*1024))) AS hgb, sum(hdfs_files) AS Files, sum(events) AS Events BY export_name | eval lgb=round(lgb,3)| eval hgb=round(hgb,3)| sort -lgb | rename export_name AS "Export Job", lgb AS "Raw GB", hgb AS "HDFS GB" | fieldformat Events=tostring(Events, "commas")*

Errors*:*
index=_internal sourcetype=hdfsexport error

Export Summary metrics*:*
index=_internal sourcetype=export_metrics group=export

Transfer metrics
index=_internal sourcetype=export_metrics group=transfer