File transfer using Hadoop Data Integration Tool (SKOOL)

File transfer is based on Java HaaS Httpfs client which is a client utility written in java by DSS Big data COE team as part of HaaS common capabilities. This utility can be used to ingest the data into HaaS location using architecturally approved HTTPFS interface in HaaS.

Prequisites:

Java JDK 1.7+

Any OS which supports Java JDK 1.7+

Step 1: Java Installation

Installation JDK 1.7 as private VM on the nominated OS system.

Go to the Installed java bin location using command/shell and type java –version. This should return the installed version of Java.

Step 2: Using FileSystem API from SKOOL.

The below two files need to be updated before running the final command -

**haas-httpfs-config.properties**

**log4j.properties**

log4j.properties is not necessarily updated unless you want your specific log formats or location to store the log file.

**haas-httpfs-config.properties 🡪** This is the configuration file which contains the credentials for Kerberos login and host and port of HaaS that host the REST services.This property file must in same location as **haas-httpfs-client-0.0.1.jar.**

The content of the properties file are:

***haas.authType = Kerberos***

***haas.realm = IUSER.IROOT.ADIDOM.COM***

***haas.kdc = iuuktvsmkp02.iuser.iroot.adidom.com***

**haas.servicePrincipal = HTTP/tplhc01g001.iuser.iroot.adidom.com**

**# tplhc01g001.iuser.iroot.adidom.com**

**haas.username = <CAD functional account ID>**

**haas.password = <Encrypted Password>**

**haas.httpfs.host = haas-1a.nat.bt.com**

**haas.httpfs.port = 14000**

**haas.kerberos.debug = true**

The haas.authType, haas.realm & haas.kdc properties will hardly change. So it is important that we don’t modify these properties.

Haas.servicePrincipal property must be sent to **HTTP/tplhc01g001.iuser.iroot.adidom.com** for RoBT cluster and **HTTP/tplhc01g004.iuser.iroot.adidom.com** for OR clsuter**.**

**haas.username 🡪** This is CAD functional account that have access service instance on HaaS.

**haas.password 🡪** This is encrypted CAD functional account. This is password is generated from java Httpfs client utility itself. The steps is mentioned below in the Password generation section.

**haas.httpfs.host & haas.httpfs.port 🡪** The host name & port where the REST services are hosted

For RoBT it is **haas-1a.nat.bt.com and port is 14000**

For OR it is **haas-1b.nat.bt.com and port is 14000**

**haas.kerberos.debug 🡪** This properties is used for debugging the Kerberos connection. It is can set to false as well.

**log4j.properties 🡪** This is log properties. The content is

log4j.appender.file=org.apache.log4j.RollingFileAppender

**log4j.appender.file.File=logs/haas-httpfs-client.log**

log4j.appender.file.MaxFileSize=20MB

log4j.appender.file.MaxBackupIndex=1

log4j.appender.file.layout=org.apache.log4j.PatternLayout

log4j.appender.file.layout.ConversionPattern=%d{ABSOLUTE} %5p %c{1}:%L - %m%n

log4j.appender.file.File – must be set to the location in the OS for capturing the logs.

Onetime -Password –Generation 🡪 For generating the password for the first time **haas.password must be kept blank.**

**Steps to generate the password:**

**Open a command / Shell Set the private JVM java home**

**For Windows it should be**

**set JAVA\_HOME=<INSTALLED\_JAVA\_LOCATION>**

**set PATH=%JAVA\_HOME%\bin;%PATH%**

**For Non- Windows**

**export JAVA\_HOME=<INSTALLED\_JAVA\_LOCATION>**

**export PATH=$JAVA\_HOME/bin;$PATH**

**Run following command**

**java -cp skool.jar:/opt/cloudera/parcels/CDH/jars/\* com.bt.haas.httpfs.client.** **HaaSHTTPFSClient**

**This will prompt for password. Please enter the CAD functional password.**

**This will add password to the haas-httpfs-config.properties file.**

**Calling HTTPFS Rest Api using java Httpfs client you can get the list of commands supported in by java Httpfs client.**

**Run the commands**

**java -cp skool.jar:/opt/cloudera/parcels/CDH/jars/\* com.bt.haas.httpfs.client.** **HaaSHTTPFSClient -help**

This will list of commands with proper description of the commands .

For example: For ingesting a sample.txt file into HaaS you can run following command

**java -cp skool.jar:/opt/cloudera/parcels/CDH/jars/\* com.bt.haas.httpfs.client.** **HaaSHTTPFSClient –copyFromLocal /home/<CAD account ID>/sample.txt user/<HaaS Instance Name>/input\_files/**

**copyFromLocal** 🡪 This is file ingestion call

**/home/<CAD Account ID>/sample.txt** 🡪 this sample location of local file that need to be ingested.

**user/<HaaS Instance Name>/input\_files/** 🡪 location of destination HaaS location.

**You should get** HTTPFS Operation Success : 201 **after successful ingestion.**