**본 문서는 Greenplum 5.x 이후 버전 기반 PXF에 외부 데이터(JDBC, Isilon, S3) 연동 설정 방법을 기술합니다.**

* Command Prompt 범례

|  |  |
| --- | --- |
| **[~]#** | root 계정으로 수행 해야 하는 명령어 |
| **[~]$** | gpadmin 계정으로 수행 해야 하는 명령어 |
| **gpadmin=#** | psql 프롬프트에서 수행하는 query |

# Setting Java for PXF

## Installation JDK

아래의 참고링크를 바탕으로 모든 노드에 root 계정으로 1.8 이후 버전의 JDK를 설치합니다.

Reference URL : https://gpdb.docs.pivotal.io/5180/pxf/install\_java.html

설치된 JDK 경로를 확인합니다. 주로 아래와 같이 /usr/lib/jvm 경로에 설치되며, 설치 결과는 alternatives 명령으로 확인 및 수정이 가능 합니다.

[ alternatives 명령 예제 ]

|  |
| --- |
| [~]# alternatives --display java  [~]# alternatives --config java  /etc/alternatives/java -> /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.212.b04-0.el7\_6.x86\_64/jre/bin/java |

## Env variable Configuration for JAVA

JAVA\_HOME 은 반드시 .bashrc 파일에 추가 해야 합니다. (all node)

JAVA\_HOME 경로는 Java 설치 경로에 따라 달라질 수 있습니다.

|  |
| --- |
| [~]$ all  => echo 'export JAVA\_HOME=/usr/local/java/jdk1.8.0\_211/jre' >> /home/gpadmin/.bashrc  => cat /home/gpadmin/.bashrc | grep JAVA\_HOME  [gpdbsdw2] export JAVA\_HOME=/usr/local/java/jdk1.8.0\_211/jre  [gpdbsmdw] export JAVA\_HOME=/usr/local/java/jdk1.8.0\_211/jre  [gpdbsdw1] export JAVA\_HOME=/usr/local/java/jdk1.8.0\_211/jre  [ gpdbmdw] export JAVA\_HOME=/usr/local/java/jdk1.8.0\_211/jre  [gpdbsdw3] export JAVA\_HOME=/usr/local/java/jdk1.8.0\_211/jre |

# Setting PXF

## Environment variable Configuration for PXF

PXF\_CONF 는 intialize후에 계속 user가 사용하게될 환경 및 library를 저장하는 공간을 설정한는 것으로 gpadmin의 .bashrc에 환경변수 설정이 필요합니다. (all node)

대상파일 : /home/gpadmin/.bashrc

추가내용 : PXF\_CONF=/usr/local/greenplum-pxf

* 경로는 임으로 설정하면 됩니다. 가급적 gpadmin 사용자가 접근하기 편한 경로로 설정하는게 좋습니다.
* 위 예제와 같이 /usr/local을 경로로 지정할 경우에는 gpadmin 계정으로 폴더 생성이 안되므로 해당 폴더를 gpadmin 계정 권한으로 미리 생성이 필요합니다.

root 계정으로 모든 서버에 폴더 생성 및 gpadmin으로 ownership 변경

|  |
| --- |
| [~]# all  => mkdir -p /usr/local/greenplum-pxf  => chown gpadmin:gpadmin /usr/local/greenplum-pxf |

PXF\_CONF 환경 변수 추가(.bashrc)

|  |
| --- |
| [~]$ all  => echo 'export PXF\_CONF=/usr/local/greenplum-pxf' >> /home/gpadmin/.bashrc  => cat /home/gpadmin/.bashrc | grep PXF\_CONF  [gpdbsdw1] export PXF\_CONF=/usr/local/greenplum-pxf  [gpdbsmdw] export PXF\_CONF=/usr/local/greenplum-pxf  [gpdbsdw3] export PXF\_CONF=/usr/local/greenplum-pxf  [ gpdbmdw] export PXF\_CONF=/usr/local/greenplum-pxf  [gpdbsdw2] export PXF\_CONF=/usr/local/greenplum-pxf |

## Initialization PXF

gpadmin 계정으로 아래의 명령어를 수행하여 PXF를 초기화 합니다. (mdw only)

|  |
| --- |
| [~]$ $GPHOME/pxf/bin/pxf cluster init  Initializing PXF on master and 4 other hosts...  PXF initialized successfully on 5 out of 5 hosts |

PXF\_CONF로 지정된 directory에 아래와 같은 directory 구조가 만들어 집니다.

|  |
| --- |
| drwxr-xr-x 2 gpadmin gpadmin 76 5월 23 13:54 conf  drwxrwxr-x 2 gpadmin gpadmin 6 5월 22 11:00 keytabs  drwxrwxr-x 2 gpadmin gpadmin 4096 5월 23 14:48 lib  drwxrwxr-x 2 gpadmin gpadmin 305 5월 23 14:06 logs  drwxrwxr-x 3 gpadmin gpadmin 21 5월 22 11:00 servers  drwxr-xr-x 2 gpadmin gpadmin 237 5월 22 11:00 templates |

## Configuration JDBC (option - JDBC)

**GPDB v.5.19 이전 버전 일 경우**

$PXF\_CONF/libs directory에 연결하고자 하는 database의 jdbc driver를 복사합니다. (mdw only)

|  |
| --- |
| [~]$ cp ojdbc6.jar $PXF\_CONF/lib |

**GPDB v5.19 이후 버전 일 경우**

$PXF\_CONF/libs directory에 연결하고자 하는 database의 jdbc driver를 복사합니다. (mdw only)

|  |
| --- |
| [~]$ cp ojdbc6.jar $PXF\_CONF/lib |

$PXF\_CONF/servers/에 jdbc를 통해 연결하고자 하는 external database에 대한 연결정보를 저장할 directory를 만듭니다. 접속 시스템을 명시적으로 만들어 관리하는 것을 권장 합니다. (mdw only)

|  |
| --- |
| [~]$ mkdir $PXF\_CONF/servers/oracle |

위에서 생성한 directory에 아래와 같이 연결할 database의 jdbc 접속 정보 파일을 생성합니다. (mdw only)

[ 예제 ] File : /usr/local/greenplum-pxf/servers/default/jdbc-site.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <configuration>  <property>  <name>jdbc.driver</name>  <value>**oracle.jdbc.OracleDriver**</value>  </property>  <property>  <name>jdbc.url</name>  <value>**jdbc:oracle:thin:@172.16.127.161:1521/orcl**</value>  </property>  <property>  <name>jdbc.user</name>  <value>**khenc**</value>  </property>  <property>  <name>jdbc.password</name>  <value>**oracle**</value>  </property>  </configuration> |

## Configuration Isilon connection (option - Isilon)

$PXF\_CONF/servers/ directory에 연결하고자 하는 isilon 서버 정보를 저장할 경로를 생성 합니다.

default directory가 생성되어 있으나, 접속 대상 시스템이 여러 개일 경우 directory를 구분하여 생성해야 합니다. (mdw only)

|  |
| --- |
| [~]$ mkdir -p $PXF\_CONF/servers/hdfs |

위에서 생성한 directory에 아래와 같이 연결할 Isilon 시스템의 접속 정보를 저장하는 core-site.xml 파일을 생성 합니다. (mdw only)

Isilon 연결 정보는 각 시스템 환경에 맞춰서 설정 해야 합니다. 기존에 hdfs를 이용하여 external table을 생성 했다면, 이 때 사용했던 URL을 지정하시면 됩니다.

[ 예제 ] File : /usr/local/greenplum-pxf/servers/hdfs/core-site.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>  <!--  Licensed under the Apache License, Version 2.0 (the "License");  you may not use this file except in compliance with the License.  You may obtain a copy of the License at  http://www.apache.org/licenses/LICENSE-2.0  Unless required by applicable law or agreed to in writing, software  distributed under the License is distributed on an "AS IS" BASIS,  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  See the License for the specific language governing permissions and  limitations under the License. See accompanying LICENSE file.  -->  <!-- Put site-specific property overrides in this file. -->  <configuration>  <property>  <name>fs.defaultFS</name>  <value>**hdfs://bdsi.datalake.sec.com**</value>  </property>  </configuration> |

## Configuration S3 connection (option - S3)

$PXF\_CONF/servers/ directory에 연결하고자 하는 S3 서버 정보를 저장할 경로를 생성 합니다.

default directory가 생성되어 있으나, 접속 대상 시스템이 여러 개일 경우 directory를 구분하여 생성해야 합니다. (mdw only)

|  |
| --- |
| [~]$ mkdir -p $PXF\_CONF/servers/s3 |

위에서 생성한 directory에 아래와 같이 연결할 s3 시스템의 접속 정보를 저장하는 s3-site.xml 파일을 생성 합니다. (mdw only)

S3 연결 정보는 각 시스템 환경에 맞춰서 설정 해야 합니다.

[ 예제 ] File : /usr/local/greenplum-pxf/servers/s3/s3-site.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>  <!-- Put site-specific property overrides in this file. -->  <configuration>  <property>  <name>fs.s3a.endpoint</name>  <value>**http://s3.api.datalake.sec.com:9090**</value>  </property>  <property>  <name>fs.s3a.access.key</name>  <value>**shlee99.id**</value>  </property>  <property>  <name>fs.s3a.secret.key</name>  <value>**xr0ZR15NlCcengHlUjlhEwYRg93UP2pK96mnFccbB**</value>  </property>  <property>  <name>fs.s3a.fast.upload</name>  <value>**true**</value>  </property>  <property>  <name>fs.s3a.buffer.dir</name>  <value>**/tmp**</value>  </property>  <property>  <name>fs.s3a.multipart.size</name>  <value>**1P**</value>  </property>  </configuration> |

## PXF parameter setting

Greenplum 일반 사용자 계정을 통해서 hadoop 연결이 가능하도록 ‘PXF\_USER\_IMPERSONATION’ 파라메터를 설정해야 합니다.

Reference URL : https://gpdb.docs.pivotal.io/550/pxf/pxfuserimpers.html

[ 예제 ] File : /usr/local/greenplum-pxf/conf/pxf-env.sh

|  |
| --- |
| [~]$ vi /usr/local/Greenplum-pxf/conf/pxf-env.sh  #!/bin/bash  …  # End-user identity impersonation, set to true to enable  # export PXF\_USER\_IMPERSONATION=true  **export PXF\_USER\_IMPERSONATION=false** |

## PXF Restart

PXF 설정 변경 사항이 반영 되도록 sync 및 재기동 합니다. (mdw only)

|  |
| --- |
| [~]$ source ~/.bashrc  [~]$ $GPHOME/pxf/bin/pxf cluster sync  [~]$ $GPHOME/pxf/bin/pxf cluster stop  [~]$ $GPHOME/pxf/bin/pxf cluster start |

# Monitoring

PXF 프로세스가 기동되면 $PXF\_CONF/logs에 아래의 log가 쌓입니다.

pxf-service.log와 catalina.out을 모니터링하면 PXF의 동작시 문제점을 확인할 수 있습니다.

|  |
| --- |
| -rw-rw-r-- 1 gpadmin gpadmin 1123117 5월 23 15:09 catalina.out  -rw-rw-r-- 1 gpadmin gpadmin 0 5월 22 13:54 host-manager.2019-05-22.log  -rw-rw-r-- 1 gpadmin gpadmin 443741 5월 23 15:09 localhost.2019-05-23.log  -rw-rw-r-- 1 gpadmin gpadmin 0 5월 22 13:54 manager.2019-05-22.log |

**PXF의 기동 상태 확인 (GPDB v5)**

GPDB v5 에서는 cluster 명령을 통해서 모든 세그먼트 노드에서 pxf status를 확인 하는 기능을 제공하지 않기 때문에 all 명령으로 모든 세그먼트 노드에 접속해서 pxf status 명령으로 기동 상태를 확인 해야 합니다.

|  |
| --- |
| [~]$ all  => **/usr/local/greenplum-pxf/pxf/bin/pxf status** |

**PXF의 기동 상태 확인 (GPDB v6)**

|  |
| --- |
| [~]$ **$GPHOME/pxf/bin/pxf cluster start**  Starting PXF on 3 segment hosts...  PXF started successfully on 3 out of 3 hosts  [~]$ **$GPHOME/pxf/bin/pxf cluster status**  Checking status of PXF servers on 3 hosts...  PXF is running on 3 out of 3 hosts  [~]$ **$GPHOME/pxf/bin/pxf cluster stop**  Stopping PXF on 3 segment hosts...  PXF stopped successfully on 3 out of 3 hosts  [~]$ **$GPHOME/pxf/bin/pxf cluster status**  Checking status of PXF servers on 3 hosts...  ERROR: PXF is not running on 3 out of 3 hosts  gpdbsdw1 ==> Checking if tomcat is up and running...  ERROR: PXF is down - tomcat is not running...  gpdbsdw3 ==> Checking if tomcat is up and running...  ERROR: PXF is down - tomcat is not running...  gpdbsdw2 ==> Checking if tomcat is up and running...  ERROR: PXF is down - tomcat is not running... |

# Troubleshooting

PXF start할때 memory 에러가 발생할 경우 $PXF\_CON/conf에 있는 파일 pxf-env.sh 에 아래의 항목을 추가하여 JAVA memory를 세그먼트 노드의 사양에 맞게 설정 합니다.

|  |
| --- |
| [~]$ vi /usr/local/Greenplum-pxf/conf/pxf-env.sh  #!/bin/bash  …  # Memory  # export PXF\_JVM\_OPTS="-Xmx2g -Xms1g"  export PXF\_JVM\_OPTS="-Xmx4G -Xms4G" |

특정 세그먼트 노드가 down되었을 경우 나머지 세그먼트 노드의 PXF를 개별적으로 중지할 수 있습니다.

|  |
| --- |
| [~]$ all  => **/usr/local/greenplum-pxf/pxf/bin/pxf stop**  => **/usr/local/greenplum-pxf/pxf/bin/pxf status** |

# Granting Users Access to PXF

## User생성 (option - JDBC)

jdbc-site.xml의 user항목에 설정된 user를 greenplum role로 만듭니다.

|  |
| --- |
| gpadmin=# CREATE ROLE orauser LOGIN ENCRYPTED PASSWORD ‘oracle’; |

## PXF extension 활성화 및 권한부여 (option - JDBC)

Oracle user와 동일한 role에서 PXF를 사용하기 위해서 아래와 같이 PXF extension 활성화 및 당 프로토콜에 대한 권한을 gpadmin 계정에 부여해야 합니다.

|  |
| --- |
| gpadmin=# CREATE EXTENSION pxf;  gpadmin=# GRANT SELECT ON PROTOCOL pxf TO orauser;  gpadmin=# GRANT INSERT ON PROTOCOL pxf TO orauser;  gpadmin=# GRANT ALL ON PROTOCOL pxf TO orauser; |

## PXF extension 활성화 및 권한부여 (option - Isilon/S3)

gpadmin 계정에서 pxf를 이용하여 Isilon 또는 S3를 사용하기 위해서 아래와 같이 PXF extension 활성화 및 해당 프로토콜에 대한 권한을 gpadmin 계정에 부여해야 합니다.

|  |
| --- |
| gpadmin=# CREATE EXTENSION pxf;  gpadmin=# GRANT SELECT ON PROTOCOL pxf TO gpamin;  gpadmin=# GRANT INSERT ON PROTOCOL pxf TO gpadmin;  gpadmin=# GRANT ALL ON PROTOCOL pxf TO gpadmin; |

# External Table 사용

## External Table 생성 (JDBC sample)

**GPDB v5.19 이전 버전 일 경우**

External Table을 생성할 때 JDBC custom option을 명시해줘야 합니다.

|  |
| --- |
| CREATE EXTERNAL TABLE tbl\_test1(c1 int, c2 varchar(10))  LOCATION ('**pxf://khenc.tbl\_test1?PROFILE=Jdbc&JDBC\_DRIVER=oracle.jdbc.OracleDriver&DB\_URL=jdbc:oracle:thin:@172.16.127.161:1521/orcl&USER=orauser&PASS=oracle**')  FORMAT 'CUSTOM' (FORMATTER='pxfwritable\_import'); |

**GPDB v5.19 이후 버전 일 경우**

External Table을 생성할 때 앞에서 jdbc-site.xml 파일을 저장한 server directory명을 SERVER인자로 명시해줘야 합니다.

|  |
| --- |
| CREATE EXTERNAL TABLE tbl\_test1(id int)  LOCATION ('**pxf://orauser.tbl\_test1?PROFILE=Jdbc&SERVER=oracle**’)  FORMAT 'CUSTOM' (FORMATTER='pxfwritable\_import'); |

## Partition Options 활용 (JDBC sample)

PXF JDBC 커넥터는 여러 세그먼트 호스트에서 실행되는 PXF 인스턴스에서 외부 SQL 테이블에 대한 동시 읽기 액세스를 지원합니다. 이 기능을 Partition 이라고합니다.

* PARTITION\_BY 옵션은 대상 테이블의 데이터 분포도를 고려하여 partition 대상 컬럼을 선정하는 것이 좋으며, type은 int, date, enum의 3 종류를 지원합니다.
* Interval을 적절히 설정하면, 대상 테이블에 대한 JDBC 연결 최적 수와 Greenplum 세그먼트 최적 분포에 따라 대상 테이블 데이터를 병렬로 처리 합니다.

|  |
| --- |
| CREATE EXTERNAL TABLE tb\_sale(SALE\_DATE varchar(8), SALE\_TIME varchar(6), CUST\_NO varchar(10), SALE\_AMT int, INPUT\_DATE date)  LOCATION ('pxf://khenc.tb\_sale?PROFILE=Jdbc&JDBC\_DRIVER=oracle.jdbc.OracleDriver&DB\_URL=jdbc:oracle:thin:@172.16.127.161:1521/orcl&USER=khenc&PASS=oracle&**PARTITION\_BY=SALE\_AMT:int&RANGE=100000:400000&INTERVAL=100000**')  FORMAT 'CUSTOM' (FORMATTER='pxfwritable\_import'); |

## External table 생성 (Isilon sample)

External Table을 생성 할 때 기존 hdfs를 사용할 때와 다른 점은 Isilon 시스템의 URL 정보를 생략 할 수 있다는 것입니다. 이것은 core-site.xml에 해당 정보가 등록되어 있기 때문입니다.

|  |
| --- |
| DROP EXTERNAL TABLE IF EXISTS test.ext\_test;  CREATE EXTERNAL TABLE test.ext\_test  (  T text  )  LOCATION (**'pxf://DataLake/Test/task\_20200101\_001\_0001.csv?PROFILE=HdfsTextSimple$SERVER=default**’)  FORMAT 'cvs' (delimiter ‘|’)  ENCODING ‘UTF8’; |

## External table 생성 (S3 sample)

External Table을 생성 할 때 s3-site.xml 파일에 access key와 secert key 설정 등록 여부에 따라 생성 옵션을 다르게 하여 생성 할 수 있습니다.

|  |
| --- |
| DROP EXTERNAL TABLE IF EXISTS test.ext\_s3\_test;  CREATE EXTERNAL TABLE test.ext\_s3\_test  (  Col1 text  )  LOCATION (**'pxf://Quality/memory/defect/summary/M11/KFBE/m\_defect\_summary\_KFBE\_20191130\_0000.csv?PROFILE=s3:text&SERVER=s3&accesskey=shlee99.id&secertkey=xr0ZR15NlCcengHlUjlhEwYRg93UP2pK96mnFccbB**’)  FORMAT 'CSV' (delimiter=E‘`’)  ENCODING ‘UTF8’;  DROP EXTERNAL TABLE IF EXISTS test.ext\_s3\_test2;  CREATE EXTERNAL TABLE test.ext\_s3\_test2  (  Col1 text  )  LOCATION (**'pxf://Master/global/alar\_info/20200226/9\_1566360083-0072905564.deflate?PROFILE=s3:text&SERVER=s3**’)  FORMAT 'text' (delimiter ‘^’ null ‘null’ escape ‘\’)  ENCODING ‘UTF8’; |

# JDBC Custom Options

JDBC option에 대한 상세한 내용은 다음 URL을 참조하시기 바랍니다.

URL : <https://gpdb.docs.pivotal.io/5160/pxf/jdbc_pxf.html>

|  |  |  |
| --- | --- | --- |
| **Option Name** | **Operation** | **Description** |
| JDBC\_DRIVER | Read, Write | The JDBC driver class name. (Required) |
| DB\_URL | Read, Write | The external database URL. Depends on the external SQL database, typically includes at least the hostname, port, and database name. (Required) |
| USER | Read, Write | The database user name. Required if PASS is provided. |
| PASS | Read, Write | The database password for USER. Required if USER is provided. |
| BATCH\_SIZE | Write | Integer identifying the number of INSERT operations to batch to the external SQL database. PXF always validates a BATCH\_SIZEoption, even when provided on a read operation. Batching is enabled by default. |
| POOL\_SIZE | Write | Enable thread pooling on INSERT operations and identify the number of threads in the pool. Thread pooling is disabled by default. |
| PARTITION\_BY | Read | The partition column, <column-name>:<column-type>. You may specify only one partition column. The JDBC connector supports date, int, and enum <column-type> values. A null PARTITION\_BY defaults to a single fragment. |
| RANGE | Read | Required when PARTITION\_BY is specified. The query range, <start-value>[:<end-value>]. When the partition column is an enum type, RANGE must specify a list of values, each of which forms its own fragment. If the partition column is an int or date type, RANGE must specify a finite left-closed range. That is, the range includes the <start-value> but does *not* include the <end-value>. If the partition column is a date type, use the yyyy-MM-dd date format. |
| INTERVAL | Read | Required when PARTITION\_BY is specified and of the int or date type. The interval, <interval-value>[:<interval-unit>], of one fragment. Specify the size of the fragment in <interval-value>. If the partition column is a date type, use the <interval-unit> to specify year, month, or day. |