

# 빅데이터 저장관리시스템 구성 및 적재모듈 개발하기

Hadoop과 Sqoop 설치 및 수행

작성자: 이슬이

## 내용

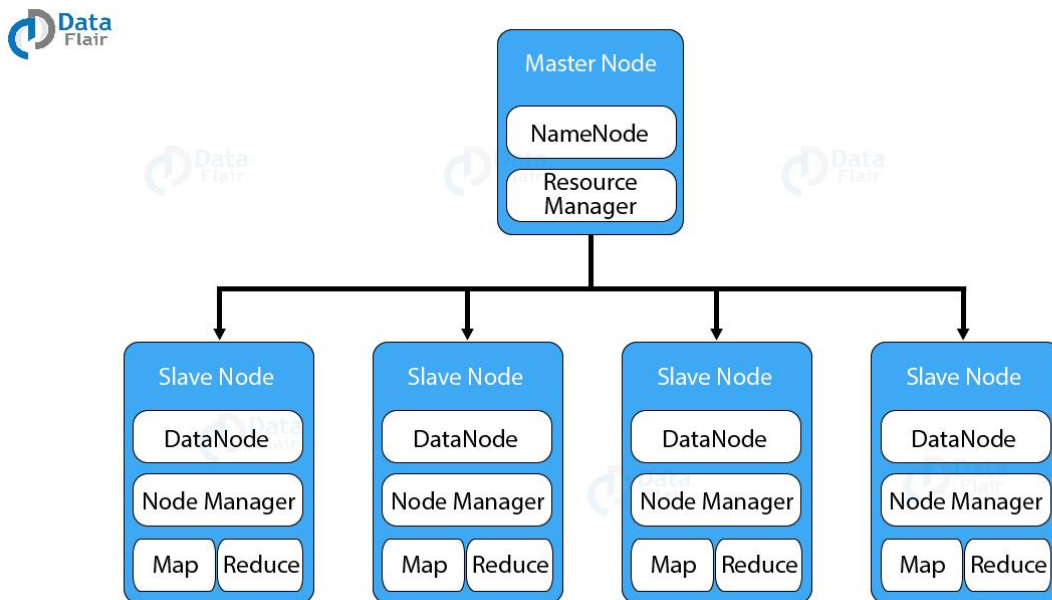
빅데이터 저장관리시스템 구성하기 .....	3
Hadoop(하둡).....	3
Hadoop 설치 수행순서 .....	3
빅데이터 적재모듈 개발하기 .....	10
Sqoop(스콥) .....	10
Sqoop을 설치 및 Import/Export 수행순서.....	10

## 빅데이터 저장관리시스템 구성하기

### Hadoop(하둡)

하둡은 빅데이터의 시스템의 핵심 소프트웨어로 2005 년 구글의 Doug Cutting 이 개발한 분산 저장/처리 SW 로 빅데이터의 에코시스템들은 대부분 하둡을 위해 존재하고 하둡에 의존해서 발전해 가고 있다. 하둡은 크게 두 가지 기능이 있는데, 첫 번째 대용량 데이터를 분산 저장하는 것이고, 두 번째는 분산 저장된 데이터를 가공/처리 하는 기능이다. 즉, 하둡은 대량의 데이터를 저장 처리하기 위한 병렬 분산 처리 소프트웨어이다.

다음은 하둡의 아키텍처이다.



출처: <https://data-flair.training/blogs/hadoop-architecture/>

### Hadoop 설치 수행순서

1. Centos 업데이트 및 필요한 SW를 설치한다.

빅데이터 SW를 설치하려면 미리 설치가 필요한 SW가 있다. 빅데이터 플랫폼에서 요구하는 필요 SW를 확인하고 미리 설치해야 한다. 대표적으로 하둡이나 스파크와 같은 SW는 자바(java)나 스칼라(scala), 파이썬(python)과 같은 SW를 요구한다. 그 밖에도 에디터나 개발 도구를 같이 설치하는 것을 추천한다. 또한 압축툴이나 패키징 관련 도구도 요구하는 경우가 있다. 하둡의 경우에는 tar, gzip, wget과 같은 도구를 실행 도중에 사용하니 미리 설치 여부를 확인하는 것이 좋다

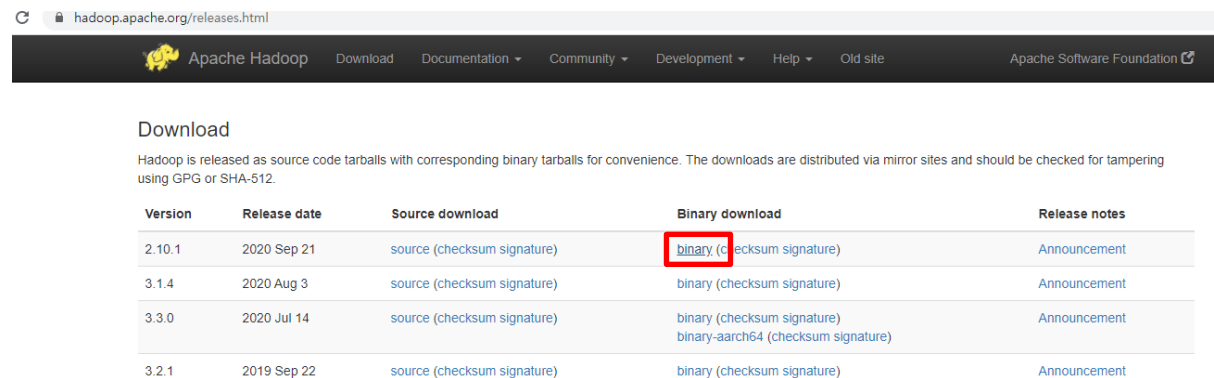
<표1> 빅데이터 저장물 설치에 관련 필요한 툴

필요 소프트웨어	용도
Java	하둡 설치 요구 사항
wget	원격 다운로드
ssh	원격 접속

```
#yum -y update
#yum install -y vim wget net-tools java-1.8*
```

## 2. 하둡을 설치한다.

- ① 하둡 다운로드 및 압축해제 (다운로드 및 압축해제는 자신의 홈 폴더에 한다고 가정한다. 여기서는 /home/bigdata)

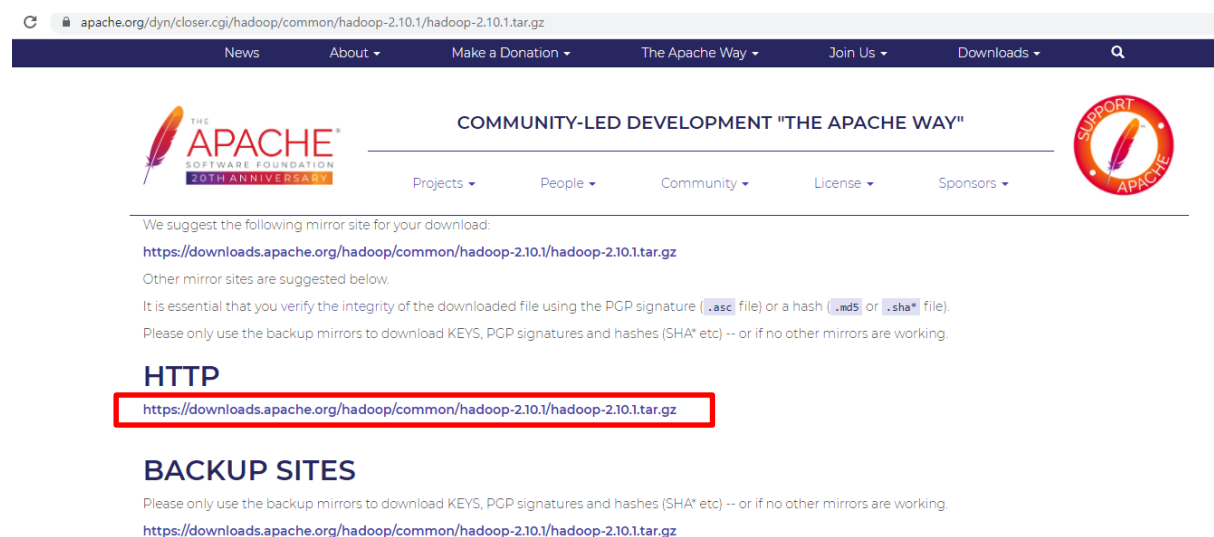


Download

Hadoop is released as source code tarballs with corresponding binary tarballs for convenience. The downloads are distributed via mirror sites and should be checked for tampering using GPG or SHA-512.

Version	Release date	Source download	Binary download	Release notes
2.10.1	2020 Sep 21	<a href="#">source (checksum signature)</a>	<a href="#">binary (checksum signature)</a>	<a href="#">Announcement</a>
3.1.4	2020 Aug 3	<a href="#">source (checksum signature)</a>	<a href="#">binary (checksum signature)</a>	<a href="#">Announcement</a>
3.3.0	2020 Jul 14	<a href="#">source (checksum signature)</a>	<a href="#">binary (checksum signature)</a> <a href="#">binary-aarch64 (checksum signature)</a>	<a href="#">Announcement</a>
3.2.1	2019 Sep 22	<a href="#">source (checksum signature)</a>	<a href="#">binary (checksum signature)</a>	<a href="#">Announcement</a>

[그림 1] 하둡 다운로드 화면



apacche.org/dyn/closer.cgi/hadoop/common/hadoop-2.10.1/hadoop-2.10.1.tar.gz

News About Make a Donation The Apache Way Join Us Downloads

THE APACHE SOFTWARE FOUNDATION 20TH ANNIVERSARY

COMMUNITY-LED DEVELOPMENT "THE APACHE WAY"

Projects People Community License Sponsors

We suggest the following mirror site for your download:

<https://downloads.apache.org/hadoop/common/hadoop-2.10.1/hadoop-2.10.1.tar.gz>

Other mirror sites are suggested below.

It is essential that you verify the integrity of the downloaded file using the PGP signature (.asc file) or a hash (.md5 or .sha\* file).

Please only use the backup mirrors to download KEYS, PGP signatures and hashes (SHA\* etc) -- or if no other mirrors are working.

HTTP

<https://downloads.apache.org/hadoop/common/hadoop-2.10.1/hadoop-2.10.1.tar.gz>

BACKUP SITES

Please only use the backup mirrors to download KEYS, PGP signatures and hashes (SHA\* etc) -- or if no other mirrors are working.

<https://downloads.apache.org/hadoop/common/hadoop-2.10.1/hadoop-2.10.1.tar.gz>

[그림 2] 하둡 다운로드 화면

```
#wget https://downloads.apache.org/hadoop/common/hadoop-2.10.1/hadoop-2.10.1.tar.gz
#tar xvfz hadoop-2.10.1.tar.gz
#mv hadoop-2.10.1 /home/bigdata/
```

```
#cd /home/bigdata/
#ln -s hadoop-2.10.1/ hadoop
```

```
drwxr-xr-x  8 root    root    200 1월  7 15:55 apache-flume-1.9.0-bin
lrwxrwxrwx  1 root    root     22 1월  7 14:30 flume -> apache-flume-1.9.0-bin
lrwxrwxrwx  1 root    root     14 1월  5 17:04 hadoop -> hadoop-2.10.1/
drwxr-xr-x 11 bigdata bigdata 172 1월  5 17:54 hadoop-2.10.1
drwxr-xr-x  7 root    root     96 1월  8 11:37 naver
lrwxrwxrwx  1 root    root     30 1월  6 16:44 sqoop -> sqoop-1.4.7.bin__hadoop-2.6.0/
drwxr-xr-x  9 bigdata bigdata 4096 1월  7 16:27 sqoop-1.4.7.bin__hadoop-2.6.0
drwxr-xr-x  6 root    root      78 1월  7 14:00 weather
```

[그림 3] 하둡 Linux 시스템 화면

② 하둡의 환경 변수는 다음과 같이 설정해 준다.

```
#vi ~/.bashrc //
맨 아래에 하둡 환경변수 선언
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk
export PATH=$PATH:$JAVA_HOME/bin
export HADOOP_HOME=/home/bigdata/hadoop
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"
export YARN_HOME=$HADOOP_HOME
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export YARN_CONF_DIR=$HADOOP_HOME/etc/hadoop
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export CLASS_PATH=$JAVA_HOME/lib:$CLASS_PATH
```

현재 쉘에 반영(bigdata101, bigdata102, bigdata103 동일) 하기 위해 저장하고 난 후 #source ~/.bashrc 을 실행해 반영하면 환경 변수를 계속 사용할 수 있다.

③ 하둡 임시 디렉터리 생성(bigdata101, bigdata102, bigdata103 동일)

```
#mkdir $HADOOP_HOME/tmp
```

④ 하둡 실행을 하기 위해서는 다음의 설정 파일을 설정해야 한다.

core-site.xml 설정(bigdata101, bigdata102, bigdata103 동일)

```
#vi $HADOOP_HOME/etc/hadoop/core-site.xml
<!-- 아래 내용 입력 -->
<configuration>
  <property>
    <name>fs.defaultFS</name>
```

```

    <value>hdfs://bigdata101:9000</value>
  </property>
</property>
  <name>hadoop.tmp.dir</name>
  <value>/home/bigdata/hadoop-2.x.x/tmp</value>
</property>
</configuration>

```

hdfs-site.xml 설정(bigdata101, bigdata102, bigdata103 동일)

```

#vi $HADOOP_HOME/etc/hadoop/hdfs-site.xml
<!-- 아래 내용 입력 -->
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>3</value>
  </property>
  <property>
    <name>dfs.permissions</name>
    <value>false</value>
  </property>
  <property>
    <name>dfs.webhdfs.enabled</name>
    <value>true</value>
  </property>
</configuration>

```

mapred-site.xml template파일 복사 후 설정(bigdata101, bigdata102, bigdata103 동일)

mapred-site.xml의 내용을 다음과 같이 설정한다(하둡의 맵 리듀스를 사용할 경우에 지정한다). 하둡 2 이후에 yarn의 하나의 서비스로 mapreduce를 지정하고 있어 이를 설정하게 된다.

```

#cp $HADOOP_HOME/etc/hadoop/mapred-site.xml.template $HADOOP_HOME/etc/hadoop/mapred-site.xml
#vi $HADOOP_HOME/etc/hadoop/mapred-site.xml
<!-- 아래 내용 입력 -->
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>

```

yarn-site.xml 설정(bigdata101, bigdata102, bigdata103 동일)

```

#vi $HADOOP_HOME/etc/hadoop/yarn-site.xml

```

```
<!-- 아래 내용 입력 -->
<configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
    <value>org.apache.hadoop.mapred.ShuffleHandler</value>
  </property>
  <property>
    <name>yarn.resourcemanager.hostname</name>
    <value>bigdata101</value>
  </property>
</configuration>
```

masters 설정(bigdata101, bigdata102, bigdata103 동일)

```
#vi $HADOOP_HOME/etc/hadoop/masters
//namenode로 지정할 서버 설정
bigdata101
```

slaves 설정(bigdata101, bigdata102, bigdata103 동일)

```
#vi $HADOOP_HOME/etc/hadoop/slaves
//datanode로 지정할 서버 설정
bigdata101 (namenode, datanode 두 개의 역할)
bigdata102
bigdata103
```

⑤ Hadoop Format(Namenode 서버인 bigdata101에서만 실행)

하둡 네임노드를 포맷(초기화), HDFS를 포맷한다.

```
#hadoop namenode -format
```

⑥ Hadoop 실행(namenode 서버인 bigdata101에서만 실행)

하둡의 HDFS를 실행하려면 start-all.sh/start-dfs.sh를 실행시킨다. 이 과정에서 네임노드와 데이터 노드, 그리고 세컨더리 네임노드가 실행된다. 그리고 맵 리듀스(MapReduce)를 위해서 start-yarn.sh를 실행한다. 이 과정에서 노드 매니저와 리소스 매니저가 실행된다. 종료하기 위해서는 각각 stop-dfs.sh와 stop-yarn.sh를 실행한다.

```
#start-all.sh
```

만약 위 명령문이 실행 안되면 아래 명령문으로 실행

```
#start-dfs.sh
#start-yarn.sh
```

```
[root@Bigdata101 bigdata]# start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
Starting namenodes on [Bigdata101]
Bigdata101: starting namenode, logging to /home/bigdata/hadoop-2.10.1/logs/hadoop-
p-root-namenode-Bigdata101.out
Bigdata101: starting datanode, logging to /home/bigdata/hadoop-2.10.1/logs/hadoop-
p-root-datanode-Bigdata101.out
Bigdata102: starting datanode, logging to /home/bigdata/hadoop-2.10.1/logs/hadoop-
p-root-datanode-Bigdata102.out
Bigdata103: starting datanode, logging to /home/bigdata/hadoop-2.10.1/logs/hadoop-
p-root-datanode-Bigdata103.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /home/bigdata/hadoop-2.10.1/logs-
/hadoop-root-secondarynamenode-Bigdata101.out
starting yarn daemons
starting resourcemanager, logging to /home/bigdata/hadoop-2.10.1/logs/yarn-root-
resourcemanager-Bigdata101.out
Bigdata102: starting nodemanager, logging to /home/bigdata/hadoop-2.10.1/logs/yarn-root-nodemanager-Bigdata102.out
Bigdata103: starting nodemanager, logging to /home/bigdata/hadoop-2.10.1/logs/yarn-root-nodemanager-Bigdata103.out
Bigdata101: starting nodemanager, logging to /home/bigdata/hadoop-2.10.1/logs/yarn-root-nodemanager-Bigdata101.out
```

[그림 4] 하둡 실행 화면

Hadoop 상태 확인(bigdata101에서 실행)

```
#jps
//아래 PID와 프로세스 확인, 순서 상관없음
2xxx NodeManager
1xxx NameNode
3xxx ResourceManager
2xxx DataNode
2xxx SecondaryNameNode
1xxx JobHistoryServer
1xxx Jps
```

Hadoop 상태 확인(bigdata102, bigdata103에서 실행)

```
#jps
//아래 PID와 프로세스 확인, 순서 상관없음
2xxx NodeManager
1xxx DataNode
1xxx Jps
```

3. HDFS를 실습한다.

① HDFS 루트(/) 파일, 폴더 리스트를 출력한다.

```
#hdfs dfs -ls /
[root@Bigdata101 bigdata]# hdfs dfs -ls /
Found 6 items
drwxr-xr-x - root supergroup 0 2021-01-07 16:11 /Jboard
drwxr-xr-x - root supergroup 0 2021-01-06 14:49 /backups
drwxr-xr-x - root supergroup 0 2021-01-07 15:39 /flume
drwxr-xr-x - root supergroup 0 2021-01-06 14:56 /html
drwxr-xr-x - root supergroup 0 2021-01-07 16:06 /sqoop
drwxrwx--- - root supergroup 0 2021-01-05 17:54 /tmp
```



[그림 5] 하둡 루트 명령어 실행 화면

#### 4. HDFS 상태 웹 인터페이스로 확인한다.

웹브라우저에서 <http://localhost:50070> 으로 접속하면 다음과 같은 동작 화면을 확인할 수 있다.

## Summary

Security is off.

Safemode is off.

7 files and directories, 0 blocks = 7 total filesystem object(s).

Heap Memory used 36.64 MB of 48.79 MB Heap Memory. Max Heap Memory is 966.69 MB.

Non Heap Memory used 44.39 MB of 45.38 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	18.56 GB
DFS Used:	12 KB (0%)
Non DFS Used:	13.03 GB
DFS Remaining:	5.52 GB (29.77%)
Block Pool Used:	12 KB (0%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%
Live Nodes	3 (Decommissioned: 0, In Maintenance: 0)
Dead Nodes	0 (Decommissioned: 0, In Maintenance: 0)
Decommissioning Nodes	0
Entering Maintenance Nodes	0
Total Datanode Volume Failures	0 (0 B)

출처: 로컬 접속(<http://localhost:50070>)을 통해 캡처

[그림 6] 하둡 HDFS 동작 상황 확인(overview)

그리고 HDFS 파일 시스템의 내용을 조회할 수 있다.

→ ↻ ⚠ 주의 요함 | 192.168.56.104:50070/explorer.html#/ ☆ \*

Hadoop Overview Datanodes Datanode Volume Failures Snapshot Startup Progress Utilities

## Browse Directory

/ Go! 📁 📄 🗑

Show 25 entries Search:

<input type="checkbox"/>	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name	
<input type="checkbox"/>	drwxr-xr-x	root	supergroup	0 B	Jan 06 14:49	0	0 B	backup	🗑
<input type="checkbox"/>	drwxr-xr-x	root	supergroup	0 B	Jan 06 14:56	0	0 B	html	🗑
<input type="checkbox"/>	drwxrwxr-x	root	supergroup	0 B	Jan 05 17:54	0	0 B	tmp	🗑

Showing 1 to 3 of 3 entries

Previous 1 Next

Hadoop, 2020.

출처: 로컬 접속(<http://localhost:50070>)을 통해 캡처

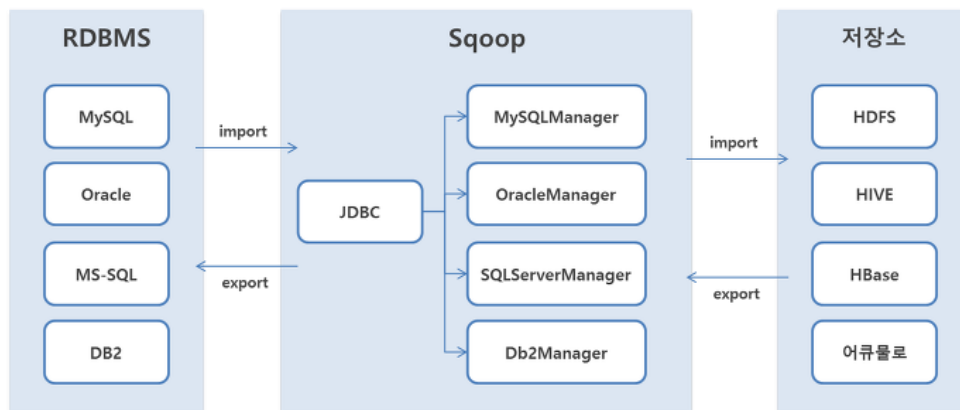
[그림 7] 하둡 HDFS 동작 상황(browse directory)

## 빅데이터 적재모듈 개발하기

### Sqoop(스콧)

RDBMS 에 있는 데이터를 특별한 전처리 없이 곧바로 HDFS 로 적재하거나, 반대로 HDFS 에 저장된 데이터를 RDBMS 로 제공해야 할 경우가 있다. RDBMS 와 HDFS 사이에서 데이터를 편리하게 Import 하거나 Export 할 때 Sqoop 을 사용한다.

다음은 스콧의 아키텍처이다.



출처: <https://excelsior-cjh.tistory.com/56>

### Sqoop을 설치 및 Import/Export 수행순서

1. 아파치 스콧(sqoop)을 설치한다.

- ① 아파치 스콧 웹사이트(<https://sqoop.apache.org/>)에서 다운로드를 클릭하고 HTTP 아래에 적힌 주소를 클릭한다. 오른쪽 마우스를 클릭해 링크 주소 복사를 한 후 #wget 명령어를 이용해 설치한다.

← → ↻ [downloads.apache.org/sqoop/1.4.7/](https://downloads.apache.org/sqoop/1.4.7/)

### Index of /sqoop/1.4.7

Name	Last modified	Size	Description
<a href="#">Parent Directory</a>		-	
<a href="#">sqoop-1.4.7.bin_hadoop-2.6.0.tar.gz</a>	2020-07-06 15:19	17M	
<a href="#">sqoop-1.4.7.bin_hadoop-2.6.0.tar.gz.asc</a>	2020-07-06 15:20	819	
<a href="#">sqoop-1.4.7.bin_hadoop-2.6.0.tar.gz.md5</a>	2020-07-06 15:19	71	
<a href="#">sqoop-1.4.7.tar.gz</a>	2020-07-06 15:20	1.1M	
<a href="#">sqoop-1.4.7.tar.gz.asc</a>	2020-07-06 15:19	819	
<a href="#">sqoop-1.4.7.tar.gz.md5</a>	2020-07-06 15:20	53	

[그림 8] 스콧 다운로드 화면

```
#wget https://downloads.apache.org/sqoop/1.4.7/sqoop-1.4.7.bin_hadoop-2.6.0.tar.gz
#tar xzvf sqoop-1.4.7.bin_hadoop-2.6.0.tar.gz
#mv sqoop-1.4.7.bin_hadoop-2.6 /home/bigdata/
#cd /home/bigdata/
#ln -s sqoop-1.4.7.bin_hadoop-2.6 sqoop
```

## ② Sqoop 환경변수 설정(Namenode 실행)

스쿱의 환경 변수인 \$SQOOP\_HOME을 설정하고 PATH를 설정해 준다(셸 설정 파일인 .bashrc에 다음 내용을 넣어 주고 #source .bashrc을 실행해 반영해 준다.

```
#vi ~/.bashrc
맨 아래 추가
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk
export SQOOP_HOME=/home/bigdata/sqoop
export SQOOP_CONF_DIR=/home/bigdata/sqoop/conf
export PATH=$PATH:$SQOOP_HOME/bin
```

변경사항 현재 셸에 반영(Namenode 실행)

```
#source ~/.bashrc
```

## ③ 필요한 jdbc 커넥터(드라이버)도 미리 설치되고 세팅되어 있어야 한다.

MySQL(<http://dev.mysql.com/downloads/connector/j/>)에서 리눅스용 mysql-connector-java-5.1.49-tar.gz 파일을 다운로드 한다. 압축해제 후 FileZilla FTP 로 mysql-connector-java-5.1.49-bin.jar 파일을 /root 경로에 업로드한 후 #mv 명령어로 sqoop/home/lib 으로 이동한다.

```
#mv mysql-co...-java-5.1.49/mysql-connector-java-5.1.49-bin.jar $SQOOP_HOME/lib
```

```
rw-rw-r-- 1 bigdata bigdata 224277 12월 19 2017 ant-contrib-1.0b3.jar
rw-rw-r-- 1 bigdata bigdata 36455 12월 19 2017 ant-eclipse-1.0-jvml.2.jar
rw-rw-r-- 1 bigdata bigdata 1344870 12월 19 2017 avro-1.8.1.jar
rw-rw-r-- 1 bigdata bigdata 186260 12월 19 2017 avro-mapred-1.8.1-hadoop2.jar
rw-rw-r-- 1 bigdata bigdata 58160 12월 19 2017 commons-codec-1.4.jar
rw-rw-r-- 1 bigdata bigdata 365552 12월 19 2017 commons-compress-1.8.1.jar
rw-rw-r-- 1 bigdata bigdata 109043 12월 19 2017 commons-io-1.4.jar
rw-rw-r-- 1 bigdata bigdata 267634 12월 19 2017 commons-jexl-2.1.1.jar
rw-rw-r-- 1 bigdata bigdata 434678 12월 19 2017 commons-lang3-3.4.jar
rw-rw-r-- 1 bigdata bigdata 60686 12월 19 2017 commons-logging-1.1.1.jar
rw-rw-r-- 1 bigdata bigdata 706710 12월 19 2017 hsqldb-1.8.0.10.jar
rw-rw-r-- 1 bigdata bigdata 36519 12월 19 2017 jackson-annotations-2.3.1.jar
rw-rw-r-- 1 bigdata bigdata 197986 12월 19 2017 jackson-core-2.3.1.jar
rw-rw-r-- 1 bigdata bigdata 232248 12월 19 2017 jackson-core-asl-1.9.13.jar
rw-rw-r-- 1 bigdata bigdata 914311 12월 19 2017 jackson-databind-2.3.1.jar
rw-rw-r-- 1 bigdata bigdata 780664 12월 19 2017 jackson-mapper-asl-1.9.13.jar
rw-rw-r-- 1 bigdata bigdata 2178774 12월 19 2017 kite-data-core-1.1.0.jar
rw-rw-r-- 1 bigdata bigdata 1801469 12월 19 2017 kite-data-hive-1.1.0.jar
rw-rw-r-- 1 bigdata bigdata 1768012 12월 19 2017 kite-data-mapreduce-1.1.0.jar
rw-rw-r-- 1 bigdata bigdata 1765905 12월 19 2017 kite-hadoop-compatibility-1.1.0.jar
rw-rw-r-- 1 root root 1006906 1월 6 16:55 mysql-connector-java-5.1.49-bin.jar
rw-rw-r-- 1 bigdata bigdata 19827 12월 19 2017 opencsv-2.3.jar
rw-rw-r-- 1 bigdata bigdata 34604 12월 19 2017 paranamer-2.7.jar
rw-rw-r-- 1 bigdata bigdata 53464 12월 19 2017 parquet-avro-1.6.0.jar
rw-rw-r-- 1 bigdata bigdata 892808 12월 19 2017 parquet-column-1.6.0.jar
rw-rw-r-- 1 bigdata bigdata 20998 12월 19 2017 parquet-common-1.6.0.jar
rw-rw-r-- 1 bigdata bigdata 279012 12월 19 2017 parquet-encoding-1.6.0.jar
rw-rw-r-- 1 bigdata bigdata 375618 12월 19 2017 parquet-format-2.2.0-rc1.jar
rw-rw-r-- 1 bigdata bigdata 20744 12월 19 2017 parquet-generator-1.6.0.jar
rw-rw-r-- 1 bigdata bigdata 205389 12월 19 2017 parquet-hadoop-1.6.0.jar
rw-rw-r-- 1 bigdata bigdata 1033299 12월 19 2017 parquet-jackson-1.6.0.jar
rw-rw-r-- 1 bigdata bigdata 25496 12월 19 2017 slf4j-api-1.6.1.jar
rw-rw-r-- 1 bigdata bigdata 592319 12월 19 2017 snappy-java-1.1.1.6.jar
rw-rw-r-- 1 bigdata bigdata 99555 12월 19 2017 xz-1.5.jar
```

[그림 8] JDBC 드라이버 확인

④ RDBMS Sqoop 계정 생성 및 테스트용 database 생성(Namenode 실행)

mysql(mariadb)가 설치되어 있어야 한다.

```
#mysql -u root -p
```

```
mysql>CREATE USER 'sqoop'@'localhost' identified by '1234';
mysql>CREATE USER 'sqoop'@'%' identified by '1234';
mysql>CREATE DATABASE sqoop;
mysql>GRANT ALL PRIVILEGES ON sqoop.* TO 'sqoop'@'localhost';
mysql>GRANT ALL PRIVILEGES ON sqoop.* TO 'sqoop'@'%';
mysql>FLUSH PRIVILEGES;
mysql>exit
```

⑤ RDBMS 테스트용 TABLE 생성 데이터 입력(Namenode 실행)

```
#mysql -u sqoop -p
```

```
mysql>use sqoop;
```

Import용 테이블 생성

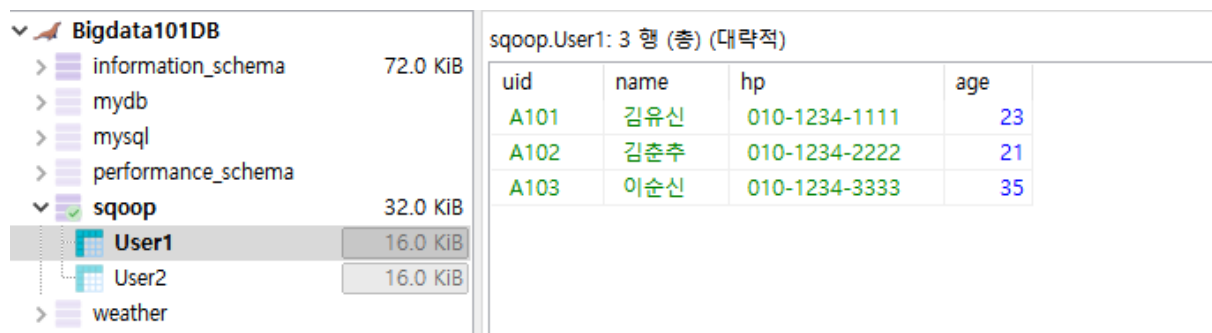
```
mysql>CREATE TABLE User1 (uid VARCHAR(10), name VARCHAR(10), hp CHAR(13), age INT);
```

Sample 데이터 입력

```
mysql>INSERT INTO User1 VALUES ('A101', '김유신', '010-1234-1111', 23);
mysql>INSERT INTO User1 VALUES ('A102', '김춘추', '010-1234-2222', 21);
mysql>INSERT INTO User1 VALUES ('A103', '이순신', '010-1234-3333', 35);
```

Export용 테이블 생성

```
mysql>CREATE TABLE User2 LIKE User1;
mysql>exit
```



Bigdata101DB

- information\_schema 72.0 KiB
- mydb
- mysql
- performance\_schema
- sqoop 32.0 KiB
  - User1 16.0 KiB
  - User2 16.0 KiB
- weather

sqoop.User1: 3 행 (총) (대략적)

uid	name	hp	age
A101	김유신	010-1234-1111	23
A102	김춘추	010-1234-2222	21
A103	이순신	010-1234-3333	35

[그림 9] MySQL 에서 데이터 입력 확인

2. Sqoop Import 를 실습한다.

다음과 같이 명령어를 입력하여 RDBMS 에서 HDFS 로 저장하는 Import 를 실행한다. 이 경우 localhost 주소는 192.168.56.104, sqoop 의 계정은 위에 설정한 것과 같이 입력한다.

```
#sqoop import --connect jdbc:mysql://192.168.56.104:3306/sqoop --table User1 --username sqoop --password 1234 -m 1 --target-dir hdfs://192.168.56.104:9000/sqoop/User1
```

... 출력 로그 중 Job ~ completed successfully 메시지 확인  
mapreduce.Job: Job job\_1594343546908\_0001 completed successfully

## Browse Directory

Showing 1 to 1 of 1 entries

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	root	supergroup	0 B	Jan 07 16:09	0	0 B	User1

[그림 10] 하둡 HDFS 웹 브라우저 폴더 확인

```
[root@Bigdata101 ~]# hdfs dfs -cat /sqoop/User1/part-m-00000
A101,김 유 신 ,010-1234-1111,23
A102,김 준 추 ,010-1234-2222,21
A103,이 순 신 ,010-1234-3333,35
```

[그림 11] 하둡 HDFS 파일 내용 확인

### 3. Sqoop Export 를 실습한다.

다음과 같이 명령어를 입력하여 RDBMS 에서 HDFS 로 저장하는 Export 를 실행한다.

```
#sqoop export --connect jdbc:mysql://192.168.56.104:3306/sqoop --table User2 --export-dir /sqoop/User1 --username sqoop --password 1234 -m 1
```

... 출력 로그 중 Job ~ completed successfully 메시지 확인  
mapreduce.Job: Job job\_1594343546908\_0001 completed successfully