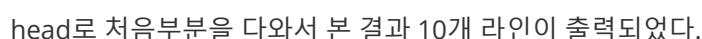


Connecting to Master Node Using SSH



hdfs

```
[hadoop@ip-172-31-82-3 ~]$ hdfs
Usage: hdfs [--config confdir] [--loglevel loglevel] COMMAND
    where COMMAND is one of:
    dfs                run a filesystem command on the file systems supported in Hadoop.
    classpath          prints the classpath
    namenode -format   format the DFS filesystem
    secondarynamenode  run the DFS secondary namenode
    namenode           run the DFS namenode
    journalnode        run the DFS journalnode
    zkfc               run the ZK Failover Controller daemon
    datanode           run a DFS datanode
    debug             run a Debug Admin to execute HDFS debug commands
    dfsadmin           run a DFS admin client
    dfsrouter          run the DFS router
    dfsrouteradmin     manage Router-based federation
    haadmin            run a DFS HA admin client
    fsck              run a DFS filesystem checking utility
    balancer          run a cluster balancing utility
    jmxget            get JMX exported values from NameNode or DataNode.
    mover             run a utility to move block replicas across
                    storage types
    oiv              apply the offline fsimage viewer to an fsimage
    oiv_legacy        apply the offline fsimage viewer to an legacy fsimage
    oev              apply the offline edits viewer to an edits file
    fetchdt          fetch a delegation token from the NameNode
    getconf           get config values from configuration
    groups            get the groups which users belong to
    snapshotDiff      diff two snapshots of a directory or diff the
                    current directory contents with a snapshot
    lsSnapshottableDir list all snapshottable dirs owned by the current user
                    Use -help to see options
    portmap           run a portmap service
    nfs3              run an NFS version 3 gateway
    cacheadmin        configure the HDFS cache
    crypto            configure HDFS encryption zones
    storagepolicies   list/get/set block storage policies
    version           print the version

Most commands print help when invoked w/o parameters.
```

hdfs로 usage를 출력

HDFS는 대용량의 데이터 저장을 목적으로 일반적인 하드웨어를 사용해 시스템 확장성 및 사용자 요청 및 데이터 증가에 대한 확장성을 보장한다. 또한 데이터 분석 시스템과의 원활한 조화를 이루며, Sequential 한 데이터의 읽기 및 쓰기에 최적화 되어있다. Random 접근 및 쿼리는 중요 처리 사항이 아니다.

hdfs dfs -ls

```
Most commands print help when invoked w/o parameters.
[hadoop@ip-172-31-82-3 ~]$ hdfs dfs -ls /
Found 3 items
drwxrwxrwt   - hadfs hadoop          0 2020-10-28 14:57 /tmp
drwxr-xr-x   - hadfs hadoop          0 2020-10-28 14:57 /user
drwxr-xr-x   - hadfs hadoop          0 2020-10-28 14:57 /var
d[hadoop@ip-172-31-82-3 ~]$ hdfs dfs -mkdir -p /dataset/movielens/
[hadoop@ip-172-31-82-3 ~]$ -ls /dataset/
-bash: -ls: command not found
[hadoop@ip-172-31-82-3 ~]$ hdfs dfs -ls /dataset/
Found 1 items
drwxr-xr-x   - hadoop hadoop          0 2020-10-28 15:31 /dataset/movielens
[hadoop@ip-172-31-82-3 ~]$
```

현재 hdfs dfs 디렉토리를 보여준다. movielens 폴더의 권한 유저, 변경일을 보여준다.

hdfs dfs -ls /dataset/

```
[hadoop@ip-172-31-82-3 ~]$ hdfs dfs -mkdir -p /dataset/movielens/
[hadoop@ip-172-31-82-3 ~]$ -ls /dataset/
-bash: -ls: command not found
[hadoop@ip-172-31-82-3 ~]$ hdfs dfs -ls /dataset/
Found 1 items
drwxr-xr-x - hadoop hadoop 0 2020-10-28 15:31 /dataset/movielens
[hadoop@ip-172-31-82-3 ~]$ hdfs dfs -put ml-20m/ratings.csv /dataset/movielens
[hadoop@ip-172-31-82-3 ~]$ hdfs dfs -ls /dataset/movielens
Found 1 items
-rw-r--r-- 1 hadoop hadoop 533444411 2020-10-28 15:33 /dataset/movielens/ratings.csv
[hadoop@ip-172-31-82-3 ~]$
```

dataset 디렉토리 내부의 데이터들의 정보를 보여준다. user 가 read,write 기능을 가지고 group other 는 read권한만 가지고 있으며 수정날짜는 2020-10-28 15:33이다. 크기는 533444411B이다.

hdfs fsck /dataset/movielens/ratings.csv

```
[hadoop@ip-172-31-82-3 ~]$ hdfs fsck /dataset/movielens/ratings.csv
Connecting to namenode via http://ip-172-31-82-3.ec2.internal:50070/fsck?ugi=hadoop&path=%2Fdataset%2Fmovielens%2Fratings.csv
FSCK started by hadoop (auth:SIMPLE) from /172.31.82.3 for path /dataset/movielens/ratings.csv at Wed Oct 28 15:33:58 UTC 2020
Status: HEALTHY
Total size: 533444411 B
Total dirs: 0
Total files: 1
Total symlinks: 0
Total blocks (validated): 4 (avg. block size 133361102 B)
Minimally replicated blocks: 4 (100.0 %)
Over-replicated blocks: 0 (0.0 %)
Under-replicated blocks: 0 (0.0 %)
Mis-replicated blocks: 0 (0.0 %)
Default replication factor: 1
Average block replication: 1.0
Corrupt blocks: 0
Missing replicas: 0 (0.0 %)
Number of data-nodes: 2
Number of racks: 1
FSCK ended at Wed Oct 28 15:33:58 UTC 2020 in 10 milliseconds

The filesystem under path '/dataset/movielens/ratings.csv' is HEALTHY
[hadoop@ip-172-31-82-3 ~]$
```

hdfs fsck로 파일시스템의 상태를 확인했다.

파일의 전체 파일 크기	533444411B (533.444411MB)
평균 블록 크기	133361102B (133.361102MB)
블록의 개수	4
HDFS의 기본 블록 크기	134217728B (128MB)

hdfs -dfs -Ddfs.block.size=1048576 -put ml-20m/ratings.csv /

```
[hadoop@ip-172-31-82-3 ~]$ hdfs dfs -Ddfs.block.size=1048576 -put ml-20m/ratings.csv /
20/10/28 15:36:21 INFO hdfs.DataStreamer: Exception while adding a block
org.apache.hadoop.ipc.RemoteException(org.apache.hadoop.hdfs.server.namenode.NotReplicatedYetException): Not repli
et: /ratings.csv._COPYING_
    at org.apache.hadoop.hdfs.server.namenode.FSDirWriteFileOp.validateAddBlock(FSDirWriteFileOp.java:171)
    at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.getAdditionalBlock(FSNamesystem.java:2587)
    at org.apache.hadoop.hdfs.server.namenode.NameNodeRpcServer.addBlock(NameNodeRpcServer.java:880)
    at org.apache.hadoop.hdfs.protocolPB.ClientNameNodeProtocolServerSideTranslatorPB.addBlock(ClientNameNodeP
ServerSideTranslatorPB.java:517)
    at org.apache.hadoop.hdfs.protocol.proto.ClientNameNodeProtocolProtos$ClientNameNodeProtocol$2.callBlockin
(ClientNameNodeProtocolProtos.java)
    at org.apache.hadoop.ipc.ProtobufRpcEngine$Server$ProtoBufRpcInvoker.call(ProtobufRpcEngine.java:507)
    at org.apache.hadoop.ipc.RPC$Server.call(RPC.java:1034)
    at org.apache.hadoop.ipc.Server$RpcCall.run(Server.java:994)
    at org.apache.hadoop.ipc.Server$RpcCall.run(Server.java:922)
    at java.security.AccessController.doPrivileged(Native Method)
    at javax.security.auth.Subject.doAs(Subject.java:422)
    at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1893)
    at org.apache.hadoop.ipc.Server$Handler.run(Server.java:2833)

    at org.apache.hadoop.ipc.Client.getRpcResponse(Client.java:1540)
    at org.apache.hadoop.ipc.Client.call(Client.java:1486)
    at org.apache.hadoop.ipc.Client.call(Client.java:1385)
    at org.apache.hadoop.ipc.ProtobufRpcEngine$Invoker.invoke(ProtobufRpcEngine.java:232)
    at org.apache.hadoop.ipc.ProtobufRpcEngine$Invoker.invoke(ProtobufRpcEngine.java:118)
    at com.sun.proxy.$Proxy10.addBlock(Unknown Source)
    at org.apache.hadoop.hdfs.protocolPB.ClientNameNodeProtocolTranslatorPB.addBlock(ClientNameNodeProtocolTra
PB.java:448)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:498)
    at org.apache.hadoop.io.retry.RetryInvocationHandler.invokeMethod(RetryInvocationHandler.java:422)
    at org.apache.hadoop.io.retry.RetryInvocationHandler$Call.invokeMethod(RetryInvocationHandler.java:165)
    at org.apache.hadoop.io.retry.RetryInvocationHandler$Call.invoke(RetryInvocationHandler.java:157)
    at org.apache.hadoop.io.retry.RetryInvocationHandler$Call.invokeOnce(RetryInvocationHandler.java:95)
    at org.apache.hadoop.io.retry.RetryInvocationHandler.invoke(RetryInvocationHandler.java:359)
    at com.sun.proxy.$Proxy11.addBlock(Unknown Source)
    at org.apache.hadoop.hdfs.DataStreamer.locateFollowingBlock(DataStreamer.java:1846)
    at org.apache.hadoop.hdfs.DataStreamer.nextBlockOutputStream(DataStreamer.java:1645)
    at org.apache.hadoop.hdfs.DataStreamer.run(DataStreamer.java:710)
20/10/28 15:36:21 WARN hdfs.DataStreamer: NotReplicatedYetException sleeping /ratings.csv._COPYING_ retries left 4
[hadoop@ip-172-31-82-3 ~]$ hdfs dfs -Ddfs.block.size=1048576 -put ml-20m/ratings.csv

[hadoop@ip-172-31-82-3 ~]$
[hadoop@ip-172-31-82-3 ~]$ hdfs dfs -Ddfs.block.size=1048576 -put ml-20m/ratings.csv /
```

```
[hadoop@ip-172-31-82-3 ~]$ hdfs fsck /ratings.csv
Connecting to namenode via http://ip-172-31-82-3.ec2.internal:50070/fsck?ugi=hadoop&path=%2Fratings.csv
FSCK started by hadoop (auth:SIMPLE) from /172.31.82.3 for path /ratings.csv at Wed Oct 28 15:38:17 UTC 2020
Status: HEALTHY
Total size: 533444411 B
Total dirs: 0
Total files: 1
Total symlinks: 0
Total blocks (validated): 509 (avg. block size 1048024 B)
Minimally replicated blocks: 509 (100.0 %)
Over-replicated blocks: 0 (0.0 %)
Under-replicated blocks: 0 (0.0 %)
Mis-replicated blocks: 0 (0.0 %)
Default replication factor: 1
Average block replication: 1.0
Corrupt blocks: 0
Missing replicas: 0 (0.0 %)
Number of data-nodes: 2
Number of racks: 1
FSCK ended at Wed Oct 28 15:38:17 UTC 2020 in 7 milliseconds

The filesystem under path '/ratings.csv' is HEALTHY
[hadoop@ip-172-31-82-3 ~]$
```

파일의 전체 파일 크기	533444411B (533.444411MB)
평균 블록 크기	1048024B (133.361102MB)
블록의 개수	509

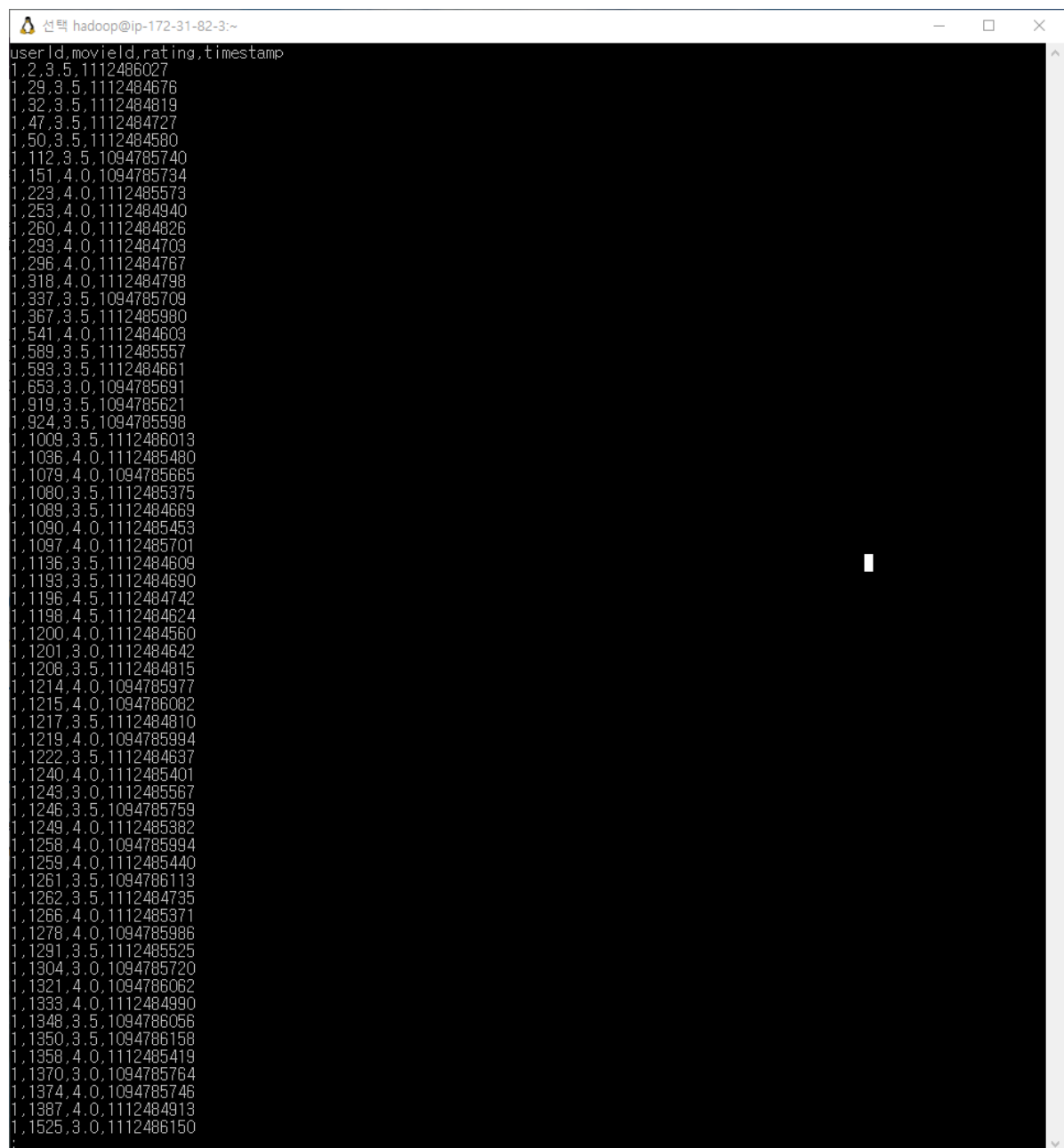
HDFS 블록 사이즈에 따른 특성

Block의 크기가 줄어 들면 Block 의 개수가 줄어들어 Namenode가 관리해야 하는 Metadata의 사이즈가 줄어든다. 왜냐하면 Namenode의 Metadata에서는 파일 리스트, 각 파일에 포함된 블록 리스트, 각 블록이 저장된 Datanode 리스트, 파일 특성들(만들어진 시간, replication factor) 등이 저장되기 때문이다. Sequential read에 적합하도록 연속적인 읽기의 높은 성능을 위해서는 큰 블록 사이즈를 사용하는 것이 유리하다.

하지만 MapReduce 작업과 같은 병렬 처리를 위해서는 작은 블록 사이즈가 유리하다. Input Data가 Split된 상태에서 Worker(현재 환경에서 실행 할 수 있는 작업의 수)의 개수가 충분하다면 블록 사이즈를 적당히 낮추는 것이 유리하다. 왜냐하면 작업은 Worker들이 병렬적으로 Map작업을 실행 한뒤, (Combiner는 옵션임) partitioner sort 단계에서 worker가 병렬적으로 데이터를 합쳐 1개의 output으로 나오게 한다.

하지만 블록사이즈가 너무 줄어 들면 Namenode가 관리해야 하는 Metadata의 사이즈가 커지게 된다. 만약 이렇게 늘어난 Metadata의 사이즈가 RAM 보다 크게 되면 Paging이 일어나게 되는데, 디스크에 있는 정보를 다시 메모리에 올리고 내리는 과정이 발생하게 되며 이는 Namenode의 속도가 현격하게 저하되는 현상이 발생하게 된다.

hdfs dfs -cat /dataset/movielens/ratings.csv | less



```
선택 hadoop@ip-172-31-82-3:~
userId,movieId,rating,timestamp
1,2,3.5,1112486027
1,29,3.5,1112484676
1,32,3.5,1112484819
1,47,3.5,1112484727
1,50,3.5,1112484580
1,112,3.5,1094785740
1,151,4.0,1094785734
1,223,4.0,1112485573
1,253,4.0,1112484940
1,260,4.0,1112484826
1,293,4.0,1112484703
1,296,4.0,1112484767
1,318,4.0,1112484798
1,337,3.5,1094785709
1,367,3.5,1112485980
1,541,4.0,1112484603
1,589,3.5,1112485557
1,593,3.5,1112484661
1,653,3.0,1094785691
1,919,3.5,1094785621
1,924,3.5,1094785598
1,1009,3.5,1112486013
1,1036,4.0,1112485480
1,1079,4.0,1094785665
1,1080,3.5,1112485375
1,1089,3.5,1112484669
1,1090,4.0,1112485453
1,1097,4.0,1112485701
1,1136,3.5,1112484609
1,1193,3.5,1112484690
1,1196,4.5,1112484742
1,1198,4.5,1112484624
1,1200,4.0,1112484560
1,1201,3.0,1112484642
1,1208,3.5,1112484815
1,1214,4.0,1094785977
1,1215,4.0,1094786082
1,1217,3.5,1112484810
1,1219,4.0,1094785994
1,1222,3.5,1112484637
1,1240,4.0,1112485401
1,1243,3.0,1112485567
1,1246,3.5,1094785759
1,1249,4.0,1112485382
1,1258,4.0,1094785994
1,1259,4.0,1112485440
1,1261,3.5,1094786113
1,1262,3.5,1112484735
1,1266,4.0,1112485371
1,1278,4.0,1094785986
1,1291,3.5,1112485525
1,1304,3.0,1094785720
1,1321,4.0,1094786062
1,1333,4.0,1112484990
1,1348,3.5,1094786056
1,1350,3.5,1094786158
1,1358,4.0,1112485419
1,1370,3.0,1094785764
1,1374,4.0,1094785746
1,1387,4.0,1112484913
1,1525,3.0,1112486150
```

ratings.csv의 파일의 내용이 보이게 된다. 첫번째 항목은 userid, 두 번째는 movie id 이며 3번째는 평점 4번째는 timestamp이다.

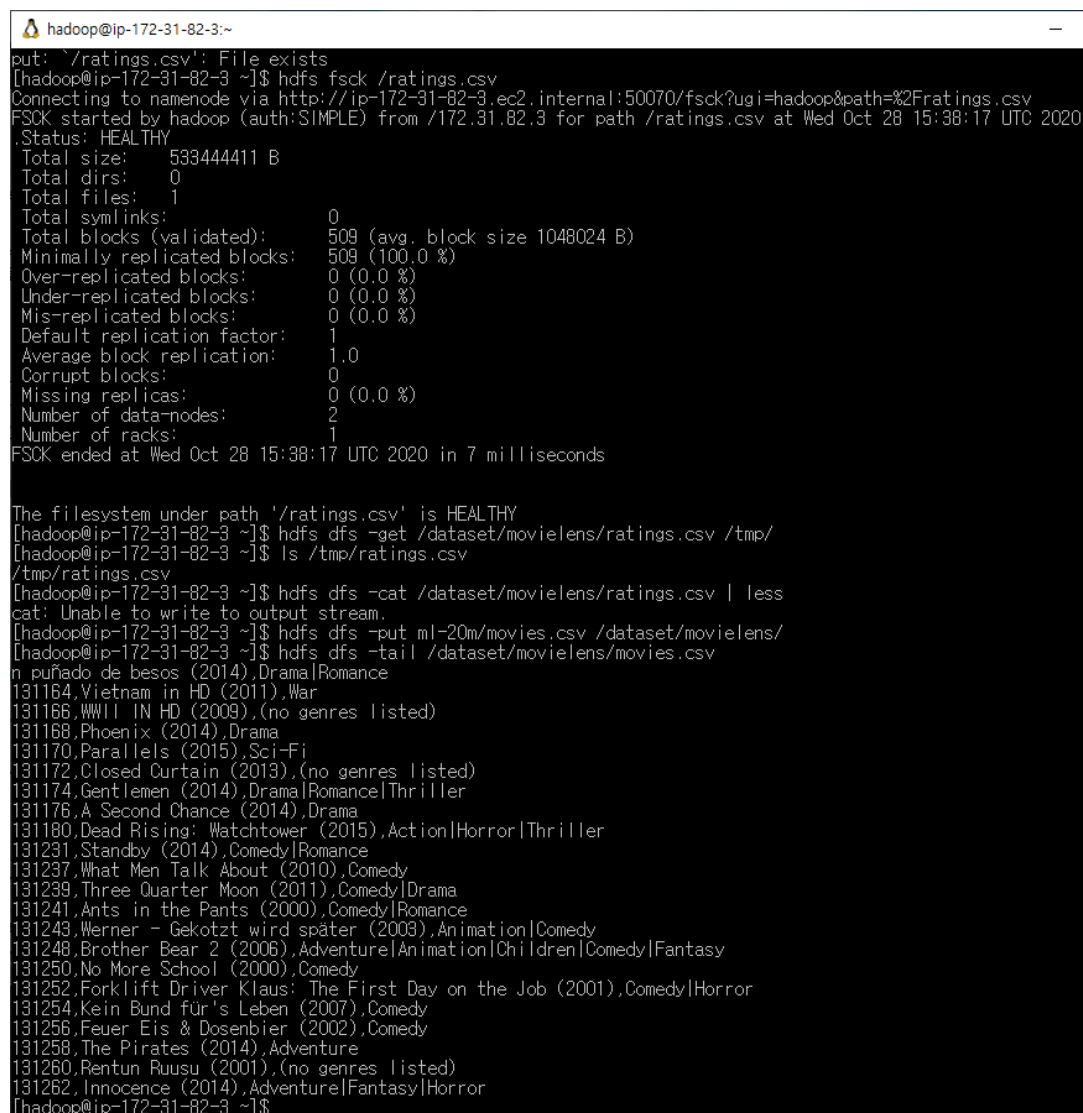
지속적으로 파일명 is HEALTHY가 나오는 것을 볼 수 있다.

이는 Heartbeat 메시지이며 Datanode가 Namenode에게 주기적으로 메시지를 보내는 것이다. 문제가 없는 soft-state를 나타내는 것 같다. Datanode에 문제가 발생 시 Replica를 저장하기 위해 새로운 datanode를 선정하고 데이터 여유 공간 및 복제를 위한 통신 오버헤드를 고려해 복사할 것이다.

```
The filesystem under path '/ratings.csv' is HEALTHY
[hadoop@ip-172-31-82-3 ~]$ hdfs dfs -get /dataset/movielens/ratings.csv /tmp/
[hadoop@ip-172-31-82-3 ~]$ ls /tmp/ratings.csv
/tmp/ratings.csv
[hadoop@ip-172-31-82-3 ~]$ hdfs dfs -cat /dataset/movielens/ratings.csv | less
cat: Unable to write to output stream.
[hadoop@ip-172-31-82-3 ~]$ hdfs dfs -put ml-20m/movies.csv /dataset/movielens/
[hadoop@ip-172-31-82-3 ~]$ hdfs dfs -tail /dataset/movielens/movies.csv
n puñado de besos (2014),Drama|Romance
```

hdfs dfs -put ml20m/movies.csv /dataset/movielens/

hdfs dfs -tail /dataset/movielens/movies.csv



```
hadoop@ip-172-31-82-3:~
put: '/ratings.csv': File exists
[hadoop@ip-172-31-82-3 ~]$ hdfs fsck /ratings.csv
Connecting to namenode via http://ip-172-31-82-3.ec2.internal:50070/fsck?ugi=hadoop&path=%2Fratings.csv
FSCK started by hadoop (auth:SIMPLE) from /172.31.82.3 for path /ratings.csv at Wed Oct 28 15:38:17 UTC 2020
Status: HEALTHY
Total size:      533444411 B
Total dirs:      0
Total files:      1
Total symlinks:   0
Total blocks (validated): 508 (avg. block size 1048024 B)
Minimally replicated blocks: 508 (100.0 %)
Over-replicated blocks: 0 (0.0 %)
Under-replicated blocks: 0 (0.0 %)
Mis-replicated blocks: 0 (0.0 %)
Default replication factor: 1
Average block replication: 1.0
Corrupt blocks: 0
Missing replicas: 0 (0.0 %)
Number of data-nodes: 2
Number of racks: 1
FSCK ended at Wed Oct 28 15:38:17 UTC 2020 in 7 milliseconds

The filesystem under path '/ratings.csv' is HEALTHY
[hadoop@ip-172-31-82-3 ~]$ hdfs dfs -get /dataset/movielens/ratings.csv /tmp/
[hadoop@ip-172-31-82-3 ~]$ ls /tmp/ratings.csv
/tmp/ratings.csv
[hadoop@ip-172-31-82-3 ~]$ hdfs dfs -cat /dataset/movielens/ratings.csv | less
cat: Unable to write to output stream.
[hadoop@ip-172-31-82-3 ~]$ hdfs dfs -put ml-20m/movies.csv /dataset/movielens/
[hadoop@ip-172-31-82-3 ~]$ hdfs dfs -tail /dataset/movielens/movies.csv
n puñado de besos (2014),Drama|Romance
131164,Vietnam in HD (2011),War
131166,WWII IN HD (2009),(no genres listed)
131168,Phoenix (2014),Drama
131170,Parallels (2015),Sci-Fi
131172,Closed Curtain (2013),(no genres listed)
131174,Gentlemen (2014),Drama|Romance|Thriller
131176,A Second Chance (2014),Drama
131180,Dead Rising: Watchtower (2015),Action|Horror|Thriller
131231,Standby (2014),Comedy|Romance
131237,What Men Talk About (2010),Comedy
131239,Three Quarter Moon (2011),Comedy|Drama
131241,Ants in the Pants (2000),Comedy|Romance
131243,Werner - Gekotzt wird später (2008),Animation|Comedy
131248,Brother Bear 2 (2006),Adventure|Animation|Children|Comedy|Fantasy
131250,No More School (2000),Comedy
131252,Forklift Driver Klaus: The First Day on the Job (2001),Comedy|Horror
131254,Kein Bund für's Leben (2007),Comedy
131256,Feuer Eis & Dosenbier (2002),Comedy
131258,The Pirates (2014),Adventure
131260,Rentun Ruusu (2001),(no genres listed)
131262,Innocence (2014),Adventure|Fantasy|Horror
[hadoop@ip-172-31-82-3 ~]$
```

hdfs dfs -put ml-20m/movies.csv /dataset/movielens/ 명령어 실행 뒤

hdfs dfs -tail /dataset/movielens/movie.csv로 이동된 movies.csv파일의 마지막 부분을 출력했다.

hdfs dfs -df -h /

```
131262,Innocence (2014),Adventure|Fantasy|Horror
[hadoop@ip-172-31-82-3 ~]$ hdfs dfs -df -h /
Filesystem                                Size      Used    Available  Use%
hdfs://ip-172-31-82-3.ec2.internal:8020  53.0 G    1.5 G      51.3 G      3%
[hadoop@ip-172-31-82-3 ~]$
```

ec2 53GB 중 1.5G가 사용 되었고 3%정도의 사용량을 보인다.

hdfs dfs -cp /dataset/movielens/*.csv

hdfs dfs -ls

```
[hadoop@ip-172-31-82-3 ~]$ hdfs dfs -cp /dataset/movielens/*.csv /
cp: '/ratings.csv': File exists
[hadoop@ip-172-31-82-3 ~]$ hdfs dfs -ls /
Found 6 items
drwxr-xr-x   - hadoop hadoop          0 2020-10-28 15:31 /dataset
-rw-r--r--   1 hadoop hadoop    1377677 2020-10-28 15:42 /movies.csv
-rw-r--r--   1 hadoop hadoop 533444411 2020-10-28 15:36 /ratings.csv
drwxrwxrwt   - hdfs  hadoop          0 2020-10-28 14:57 /tmp
drwxr-xr-x   - hdfs  hadoop          0 2020-10-28 14:57 /user
drwxr-xr-x   - hdfs  hadoop          0 2020-10-28 14:57 /var
[hadoop@ip-172-31-82-3 ~]$
```

/dataset 디렉토리는 User에 대해 읽기, 쓰기, 실행이 가능하며 Group은 read, 실행, other도 Group과 동일하다.

hdfs dfs -getfacl /movies.csv

```
[hadoop@ip-172-31-82-3 ~]$ hdfs dfs -getfacl /movies.csv
# file: /movies.csv
# owner: hadoop
# group: hadoop
user::rw-
group::r--
other::r--
```

hdfs -put ml-20m/tag.csv /dataset/movielens/

hdfs fsck /dataset/movielens/tag.csv

```
[hadoop@ip-172-31-82-3 ~]$ hdfs dfs -put ml-20m/tags.csv /dataset/movielens/
[hadoop@ip-172-31-82-3 ~]$ hdfs fsck /dataset/movielens/tags.csv
Connecting to namenode via http://ip-172-31-82-3.ec2.internal:50070/fsck?ugi=hadoop&path=%2Fdataset%2Fmovielens%2Ftags
FSCK started by hadoop (auth:SIMPLE) from /172.31.82.3 for path /dataset/movielens/tags.csv at Wed Oct 28 15:46:36 UTC
Status: HEALTHY
Total size: 16603996 B
Total dirs: 0
Total files: 1
Total symlinks: 0
Total blocks (validated): 1 (avg. block size 16603996 B)
Minimally replicated blocks: 1 (100.0 %)
Over-replicated blocks: 0 (0.0 %)
Under-replicated blocks: 0 (0.0 %)
Mis-replicated blocks: 0 (0.0 %)
Default replication factor: 1
Average block replication: 1.0
Corrupt blocks: 0
Missing replicas: 0 (0.0 %)
Number of data-nodes: 2
Number of racks: 1
FSCK ended at Wed Oct 28 15:46:36 UTC 2020 in 1 milliseconds
```

데이터 노드의 개수가 2개이기 때문에 replicated blocks가 1개로 설정되어있다. 수업 시간에 배웠을 때는 기본 정책은 Replica를 3개 둔다고 배웠다. 하나의 복제본은 임의 서버에 저장되며, 복제본 하나는 같은 랙에 존재 (같은 스위치에 연결된 랙) 하는 다른 서버에 저장된다. 다른 복제본은 떨어진 다른 랙에 저장되는데 그 이유는 같은 랙에 존재하는 다른 서버에 데이터를 저장하면 그 속도가 빨라서 빠르게 1개가 복사되기 때문이며, 다른 랙에 있는 서버에 저장하는 이유는 랙 단위로 서버 기능이 고장날 가능성이 높기 때문에 데이터 보호를 위해 다른 랙에 저장한다. 그러나 현재 데이터 노드는 2개이기 때문에 같은 랙에 있는 파일에 저장되었다고 생각된다. Minimally replicated blocks 가 100%인 것으로 보아 모든 블록이 100% 복사 되었음을 알 수 있다.

\$ /dataset/movielens/tags.csv #파일의 replica 갯수를 바꾸는 명령어를 실행 (원본 파일의 현재

replica 갯수와 다르게 임의로 설정. 만약 replica 갯수를 3으로 설정한다면 어떻게 될까요? 그 이유를 서술해주세요.

Replica의 갯수를 바꾸는 명령을 실행하고 갯수를 3으로 설정하면 Number of racks가 2로 변경되며 Number of data-nodes는 3으로 변경될 것이다. 또한 Average block replication 은 3으로 변경 될 것이다. 그 이유는 위의 설명과 같이 기본 정책이 Replica가 3이며, 하나의 복제본은 임의 서버에 저장되며, 복제본 하나는 같은 랙에 존재 (같은 스위치에 연결된 랙) 하는 다른 서버에 저장된다. 다른 복제본은 떨어진 다른 랙에 저장되는데 그 이유는 같은 랙에 존재하는 다른 서버에 데이터를 저장하면 그 속도가 빨라서 빠르게 1개가 복사되기 때문이며, 다른 랙에 있는 서버에 저장하는 이유는 랙 단위로 서버 기능이 고장날 가능성이 높기 때문에 데이터 보호를 위해 다른 랙에 저장한다. 그러나 아래와 같이 현재 데이터 노드는 2개이기 때문에 1개의 Replica가 복사되지 않았다. 그래서 Missing Replicas 가 1개 존재하게 되었으며 그로 인해 Average block replication 도 2.0이 나오게 되었다. Under-replicated blocks도 1개가 존재하게 되었다. 만약 옵션으로 3개의 Replica의 서버 위치 다른 랙으로 지정하면 Number of racks 는 3이 될 것이다.

hdfs dfs -setrep 3 /dataset/movielens/tags.csv

hdfs fsck /dataset/movielens/tags.csv


```

[hadoop@ip-172-31-82-3 ~]$ /dataset/movielens/tags.csv
-bash: /dataset/movielens/tags.csv: No such file or directory
[hadoop@ip-172-31-82-3 ~]$ hdfs dfs -setrep 3 /dataset/movielens/tags.csv
Replication 3 set: /dataset/movielens/tags.csv
[hadoop@ip-172-31-82-3 ~]$ hdfs fsck /dataset/movielens/tags.csv
Connecting to namenode via http://ip-172-31-82-3.ec2.internal:50070/fsck?ugi=hadoop&path=%2Fdataset%2Fmovielens%2Ftags.csv
FSCK started by hadoop (auth:SIMPLE) from /172.31.82.3 for path /dataset/movielens/tags.csv at Wed Oct 28 15:50:01 UTC 2020

/dataset/movielens/tags.csv: Under replicated BP-2012759687-172.31.82.3-1603897051105:blk_1073742849_2025. Target Replicas is 3 but found 2 live replica(s), 0 decommissioned replica(s), 0 decommissioning replica(s).
Status: HEALTHY
Total size: 16603996 B
Total dirs: 0
Total files: 1
Total symlinks: 0
Total blocks (validated): 1 (avg. block size 16603996 B)
Minimally replicated blocks: 1 (100.0 %)
Over-replicated blocks: 0 (0.0 %)
Under-replicated blocks: 1 (100.0 %)
Mis-replicated blocks: 0 (0.0 %)
Default replication factor: 1
Average block replication: 2.0
Corrupt blocks: 0
Missing replicas: 1 (33.33333 %)
Number of data-nodes: 2
Number of racks: 1
FSCK ended at Wed Oct 28 15:50:01 UTC 2020 in 0 milliseconds

The filesystem under path '/dataset/movielens/tags.csv' is HEALTHY
[hadoop@ip-172-31-82-3 ~]$

```

데이터 노드가 2개라서 Replica 1개가 Missing 되었다. Data Node를 추가 해야 Replica가 정상적으로 3개가 복사되며, 추가되는 레플리카의 서버가 위치한 랙의 위치가 다르다면 랙의 가수도 2로 변할 것이다. 현재 데이터 노드는 2개이기 때문에 1개의 Replica가 복사되지 않았다. 그래서 Missing Replicas 가 1개 존재하게 되었으며 그로 인해 Average block replication 도 2.0이 나오게 되었다. Under-replicated blocks도 1개가 존재하게 되었다..

hdfs dfsadmin -report

```

[hadoop@ip-172-31-82-3 ~]$ hdfs dfsadmin -report
Configured Capacity: 56877875200 (52.97 GB)
Present Capacity: 56743859146 (52.85 GB)
DFS Remaining: 55629434880 (51.81 GB)
DFS Used: 1114424266 (1.04 GB)
DFS Used%: 1.96%
Under replicated blocks: 1
Blocks with corrupt replicas: 0
Missing blocks: 0
Missing blocks (with replication factor 1): 0
Pending deletion blocks: 0

-----
Live datanodes (2):

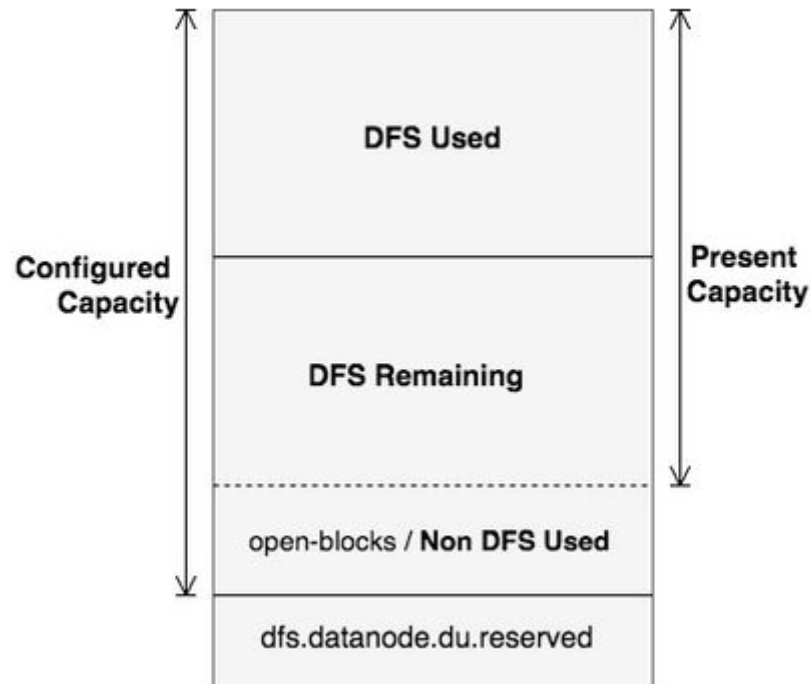
Name: 172.31.86.138:50010 (ip-172-31-86-138.ec2.internal)
Hostname: ip-172-31-86-138.ec2.internal
Decommission Status : Normal
Configured Capacity: 28438937600 (26.49 GB)
DFS Used: 843041447 (803.99 MB)
Non DFS Used: 66655577 (63.57 MB)
DFS Remaining: 27529240576 (25.64 GB)
DFS Used%: 2.96%
DFS Remaining%: 96.80%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 1
Last contact: Wed Oct 28 15:52:21 UTC 2020
Last Block Report: Wed Oct 28 15:00:09 UTC 2020

Name: 172.31.87.186:50010 (ip-172-31-87-186.ec2.internal)
Hostname: ip-172-31-87-186.ec2.internal
Decommission Status : Normal
Configured Capacity: 28438937600 (26.49 GB)
DFS Used: 271382819 (258.81 MB)
Non DFS Used: 67360477 (64.24 MB)
DFS Remaining: 28100194304 (26.17 GB)
DFS Used%: 0.95%
DFS Remaining%: 98.81%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 1
Last contact: Wed Oct 28 15:52:20 UTC 2020
Last Block Report: Wed Oct 28 15:00:17 UTC 2020

[hadoop@ip-172-31-82-3 ~]$

```

현재 Configured Capacity = 52.97GB이며, Non Dfs Used를 제외한 Present Capacity는 52.85GB이다. 또한 두개의 datanodes가 Configured Capacity를 2로 나누어 가져갔음을 알 수 있다. DFS Remaining은 52.85GB로 datanode 한개가 25.64GB 나머지 하나가 26.17GB를 가지고 있다. 이 둘을 합치면 55,629,434,880이 정상적으로 나온다. Non DFS Used에는 캐쉬 로그 등 DFS에서 관리하는 데이터가 아닌 것들이 들어간다. Under replicated blocks가 1개가 나오는데 위에서 replica를 3개로 설정을 시도 했는데 datanode가 부족해서 replica가 생성되지 않았다. 제거되길 기다리는 블록이다, 없어진 블록은 없는 것을 확인 할 수 있다. 아마 Non DFS USED에 FSImage에서 namenode 복구를 위해 사용 되는 Transaction log 등이 저장되어 있을 것 같다.



Configured Capacity :
HDFS 저장공간의 총 용량이다.

Sum of Configured Capacity for all the Data Nodes

Configured Capacity = Total Disk Space - Reserved Space로 계산된다.

Configured Capacity = Present Capacity + Non DFS Used on all the Data Nodes

Reserved space는 OS레벨의 operation에서 할당된 공간이다.

Present Capacity :

metadata와 open-blocks(Non DFS Used space) 할당 이후 실질적으로 사용가능한 저장 공간이다.
Present Capacity는 Non-HDFS 디렉토리들의 사용에 위존되어있다.

Present Capacity = DFS Used + DFS Reamining

DFS Used

HDFS에 의해 사용되는 공간이다. HDFS에 저장된 파일의 실제 크기를 알기 위해서는 DFS Used에 replcation factor를 나누면 된다.

DFS Remaining




HDFS가 더 사용할 수 있는 파일의 공간이다.

Non DFS Used

Non DFS Used = Configured Capacity - DFS Reamining - DFS Used 이다.

가격 예측

Your AWS Account Status

	Active full access (hiljh96@kookmin.ac.kr)
	\$49.55 remaining credits (estimated)
	2:60 session time

<https://aws.amazon.com/ko/ec2/pricing/on-demand/>

<https://aws.amazon.com/ko/emr/pricing/>

m4.large	2	6.5	8GiB	EBS 전용	시간당 0.10 USD
----------	---	-----	------	--------	--------------

	Amazon EC2 요금	Amazon EMR 요금
m4.large	0.10 USD/시간	0.03 USD/시간

m4.large의 가격이 시간당 0.03USD 달러이며 , On Demand이기 때문에 사용 시 가격이 청구 된다.

아래와 같이 실습에서 m4.large를 3개 사용했었다. 0.09며 이가 EC2에서도 1개의 인스턴스가 있었을 것이다.

Node type	Instance type	Instance count	Purchasing option
Master Master - 1	m4.large 2 vCore, 8 GiB memory, EBS only storage EBS Storage: 32 GiB Add configuration settings	1 Instances	<input checked="" type="radio"/> On-demand <input type="radio"/> Spot Use on-demand as max price
Core Core - 2	m4.large 2 vCore, 8 GiB memory, EBS only storage EBS Storage: 32 GiB Add configuration settings	2 Instances	<input checked="" type="radio"/> On-demand <input type="radio"/> Spot Use on-demand as max price

m4.large를 약 30~40분정도 3개를 켜놨다. 아마 가동되지 않은 시간이 포함되어있어서 가격 청구가 이렇게 된 것 같다. 파일을 다운 받고 복사하는 것과, 파일을 블록을 분산하고 나누고 레플리카를 생성하는데 컴퓨팅 자원이 다소 활용되었을 것 같다.