

HDFS Access Options, Applications

- *Able to access/use HDFS via command line*
- *Know about available application programming interfaces*
- *Example Applications*

HDFS Commands

- Invoked via **bin/hdfs** script.
- ***User commands – filesystem shell commands*** for routine operations.
- ***Administrator commands***
- ***Debug commands***
- ***Details at:***

<https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HDFSCommands.html>

Application programming interfaces

- ***Native Java API : Base class***
org.apache.hadoop.fs.FileSystem
- ***C API for HDFS***: libhdfs, header
file (hdfs.h)
- ***WebHDFS REST API***: HTTP
Get, Put, Post, and Delete
operations

HDFS NFS Gateway

- *Mount HDFS as a filesystem on the client*
- *Browse files using regular filesystem commands*
- *Upload/download files from HDFS*
- *Stream data to HDFS*

Several other options!

- *Apache Flume* – collecting, aggregating streaming data and moving into HDFS
- *Apache Sqoop* – Bulk transfers between Hadoop and datastores.

Applications using HDFS

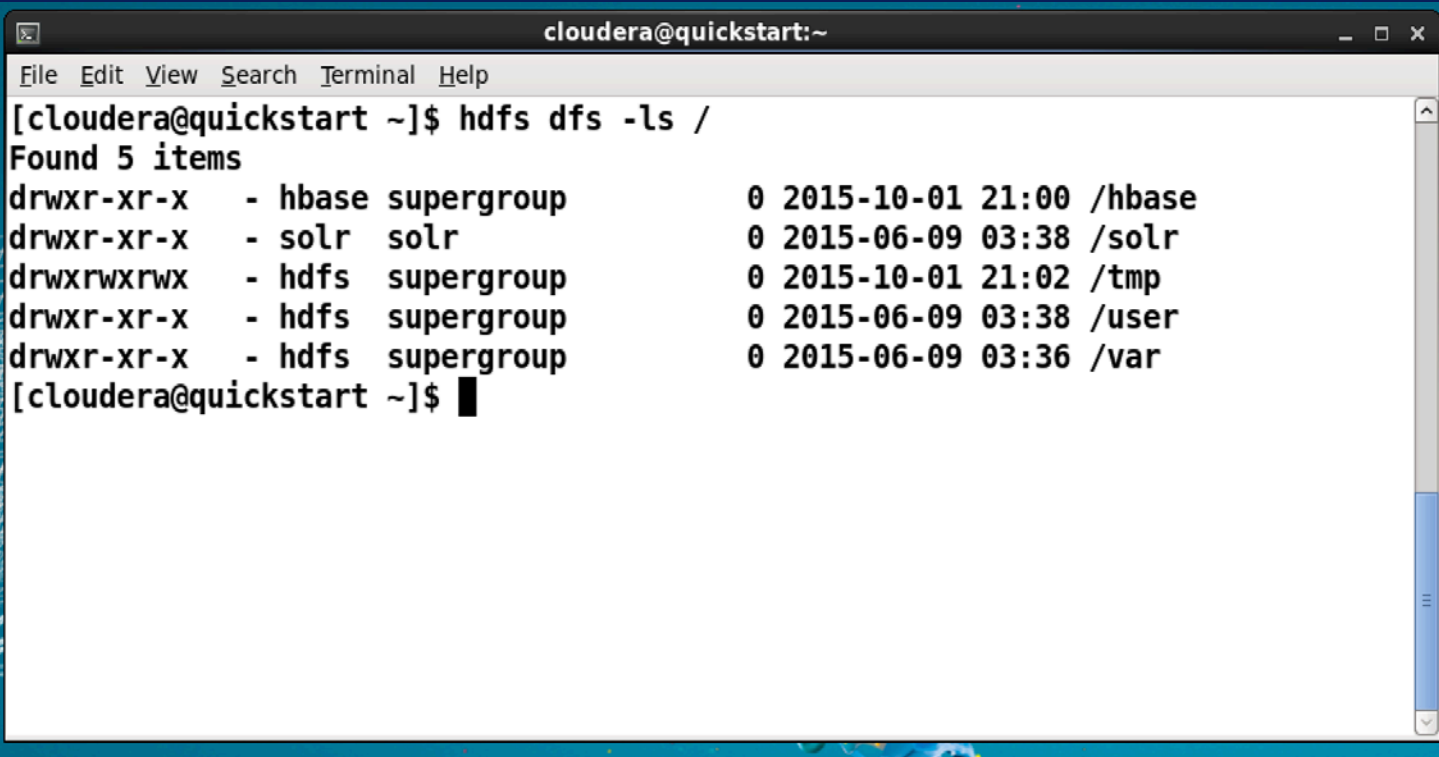
- Can use APIs to interact with HDFS
- Core component of Hadoop stack – used by all applications
- HBase is a good example of an application that runs on top of HDFS with good integration
- Spark can run directly on HDFS without other Hadoop components

HDFS Commands

- *Use HDFS commands to move data in/out of HDFS*
- *Get detailed information on files in HDFS*
- *Use administrator commands to get info on state of HDFS*

HDFS User Commands

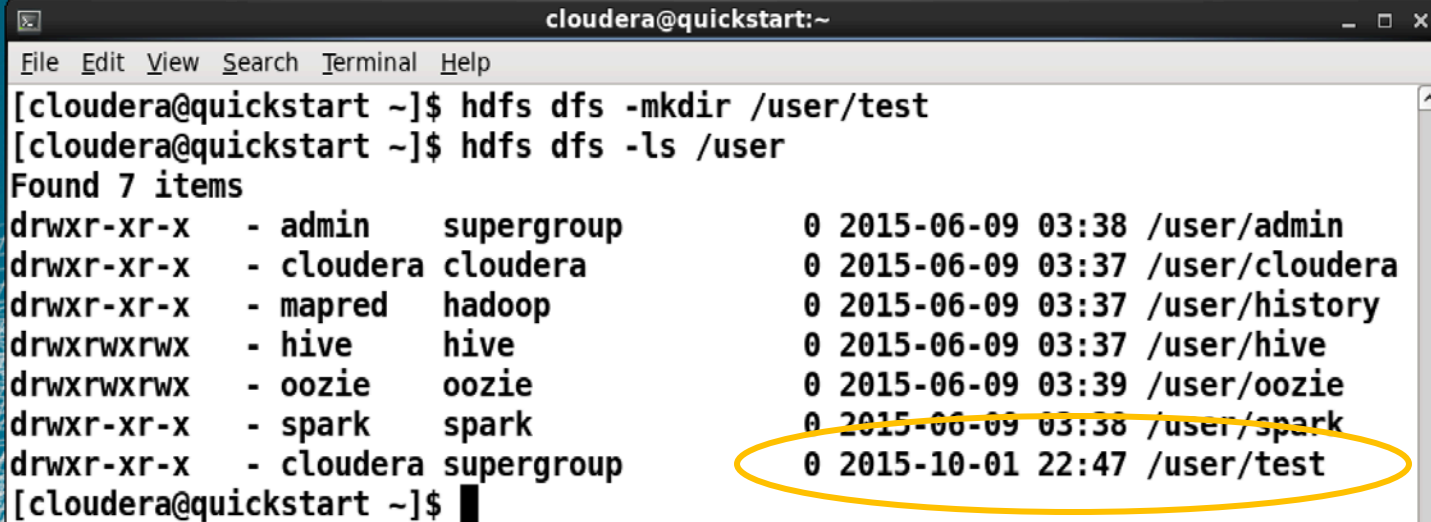
- List files in `/`: *hdfs dfs -ls /*



```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
[cloudera@quickstart ~]$ hdfs dfs -ls /  
Found 5 items  
drwxr-xr-x  - hbase supergroup      0 2015-10-01 21:00 /hbase  
drwxr-xr-x  - solr  solr            0 2015-06-09 03:38 /solr  
drwxrwxrwx  - hdfs supergroup      0 2015-10-01 21:02 /tmp  
drwxr-xr-x  - hdfs supergroup      0 2015-06-09 03:38 /user  
drwxr-xr-x  - hdfs supergroup      0 2015-06-09 03:36 /var  
[cloudera@quickstart ~]$
```


HDFS User Commands

- Make a directory: *hdfs dfs -mkdir /user/test*



A terminal window titled "cloudera@quickstart:~" showing the execution of HDFS commands. The first command is "hdfs dfs -mkdir /user/test". The second command is "hdfs dfs -ls /user", which lists the contents of the /user directory. The output shows seven items, with the last item, "/user/test", circled in yellow. The terminal window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help".

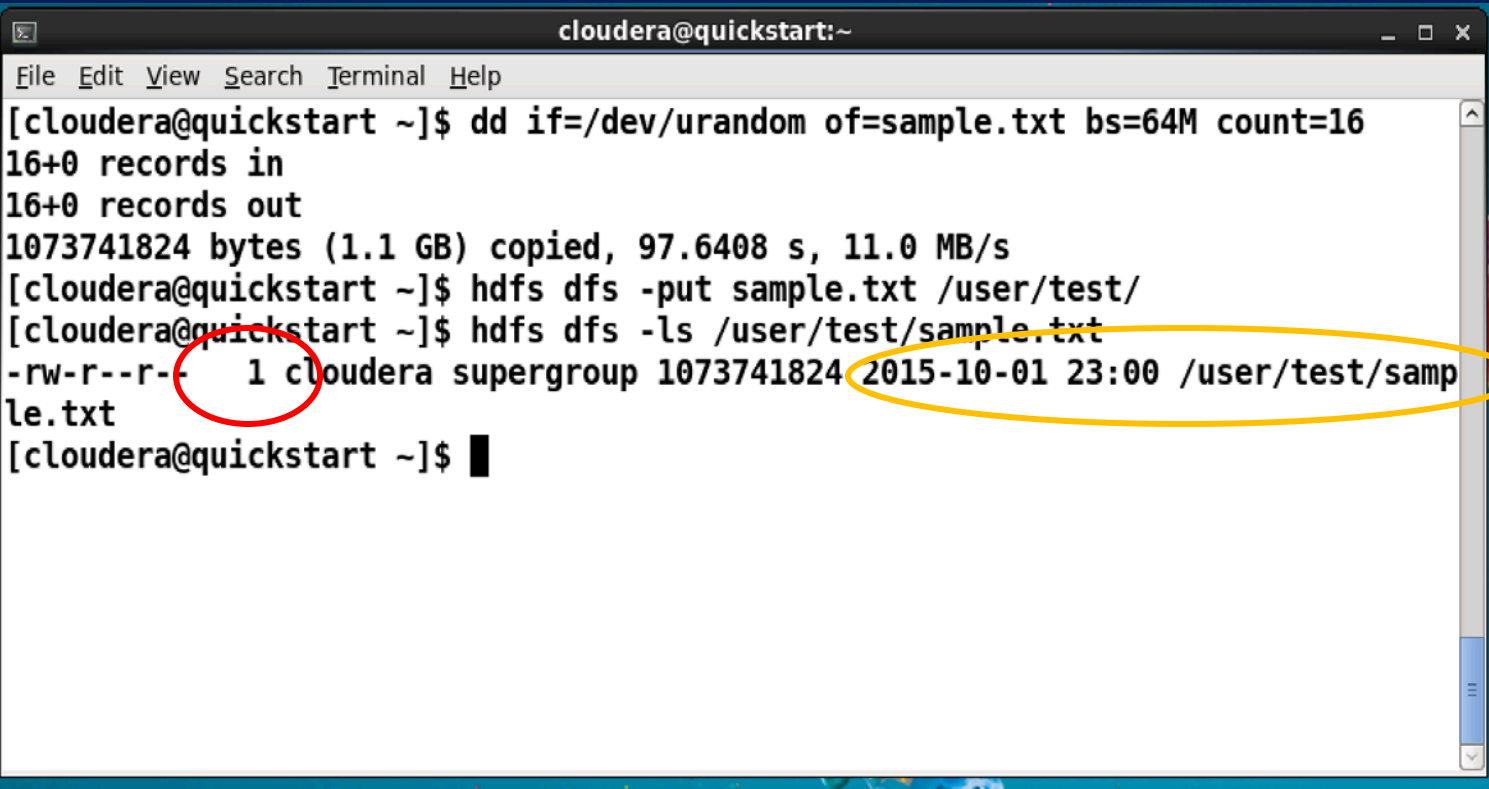
```
[cloudera@quickstart ~]$ hdfs dfs -mkdir /user/test
[cloudera@quickstart ~]$ hdfs dfs -ls /user
Found 7 items
drwxr-xr-x  - admin      supergroup          0 2015-06-09 03:38 /user/admin
drwxr-xr-x  - cloudera  cloudera            0 2015-06-09 03:37 /user/cloudera
drwxr-xr-x  - mapred    hadoop              0 2015-06-09 03:37 /user/history
drwxrwxrwx  - hive      hive                0 2015-06-09 03:37 /user/hive
drwxrwxrwx  - oozie     oozie               0 2015-06-09 03:39 /user/oozie
drwxr-xr-x  - spark     spark               0 2015-06-09 03:38 /user/spark
drwxr-xr-x  - cloudera  supergroup          0 2015-10-01 22:47 /user/test
[cloudera@quickstart ~]$
```

Create a local file

- Now lets create a local file and copy it into HDFS.
- We create a file with random data using the linux utility dd.
- Command:
`dd if=/dev/urandom of=sample.txt bs=64M count=16`
- Creates 1GB file called sample.txt on the local filesystem.

HDFS User Commands

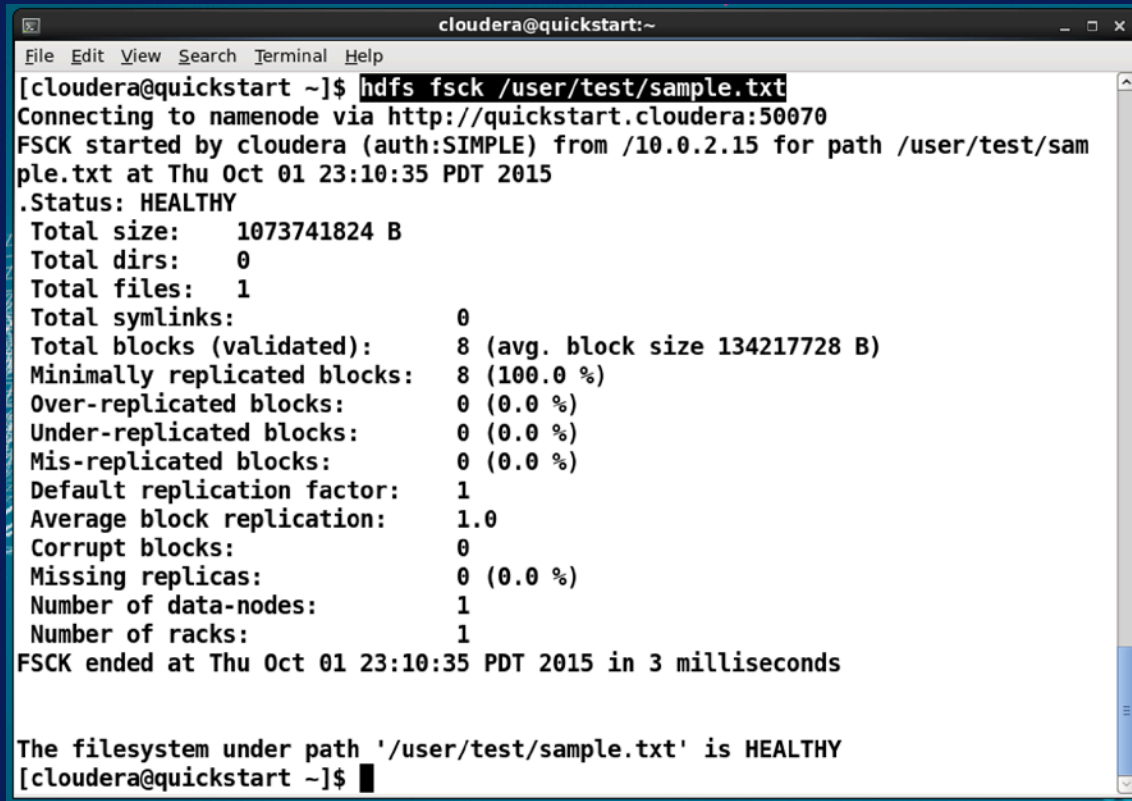
- hdfs dfs -put sample.txt /user/test*

A terminal window titled 'cloudera@quickstart:~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the execution of 'dd' to create a 1.1 GB file 'sample.txt' and 'hdfs dfs' commands to upload it and list its details. The file permissions '-rw-r--r--' are circled in red, and the file's metadata (owner, size, date, and path) is circled in yellow.

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
[cloudera@quickstart ~]$ dd if=/dev/urandom of=sample.txt bs=64M count=16  
16+0 records in  
16+0 records out  
1073741824 bytes (1.1 GB) copied, 97.6408 s, 11.0 MB/s  
[cloudera@quickstart ~]$ hdfs dfs -put sample.txt /user/test/  
[cloudera@quickstart ~]$ hdfs dfs -ls /user/test/sample.txt  
-rw-r--r--  1 cloudera supergroup 1073741824 2015-10-01 23:00 /user/test/samp  
le.txt  
[cloudera@quickstart ~]$
```

HDFS fsck

- Command: *hdfs fsck /user/test/sample.txt*

A terminal window titled 'cloudera@quickstart:~' with a menu bar (File, Edit, View, Search, Terminal, Help). The command 'hdfs fsck /user/test/sample.txt' is entered and executed. The output shows the connection to the namenode, the start of the FSCK process, and a detailed status report for the file '/user/test/sample.txt'. The status is 'HEALTHY'. The report includes statistics such as total size, number of blocks, replication factor, and the number of data nodes and racks. The FSCK process ended in 3 milliseconds. The final message states that the filesystem is healthy.

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
[cloudera@quickstart ~]$ hdfs fsck /user/test/sample.txt  
Connecting to namenode via http://quickstart.cloudera:50070  
FSCK started by cloudera (auth:SIMPLE) from /10.0.2.15 for path /user/test/sample.txt at Thu Oct 01 23:10:35 PDT 2015  
.Status: HEALTHY  
Total size:      1073741824 B  
Total dirs:      0  
Total files:     1  
Total symlinks:   0  
Total blocks (validated):      8 (avg. block size 134217728 B)  
Minimally replicated blocks:  8 (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:          1  
Number of racks:               1  
FSCK ended at Thu Oct 01 23:10:35 PDT 2015 in 3 milliseconds  
  
The filesystem under path '/user/test/sample.txt' is HEALTHY  
[cloudera@quickstart ~]$
```

HDFS User Commands

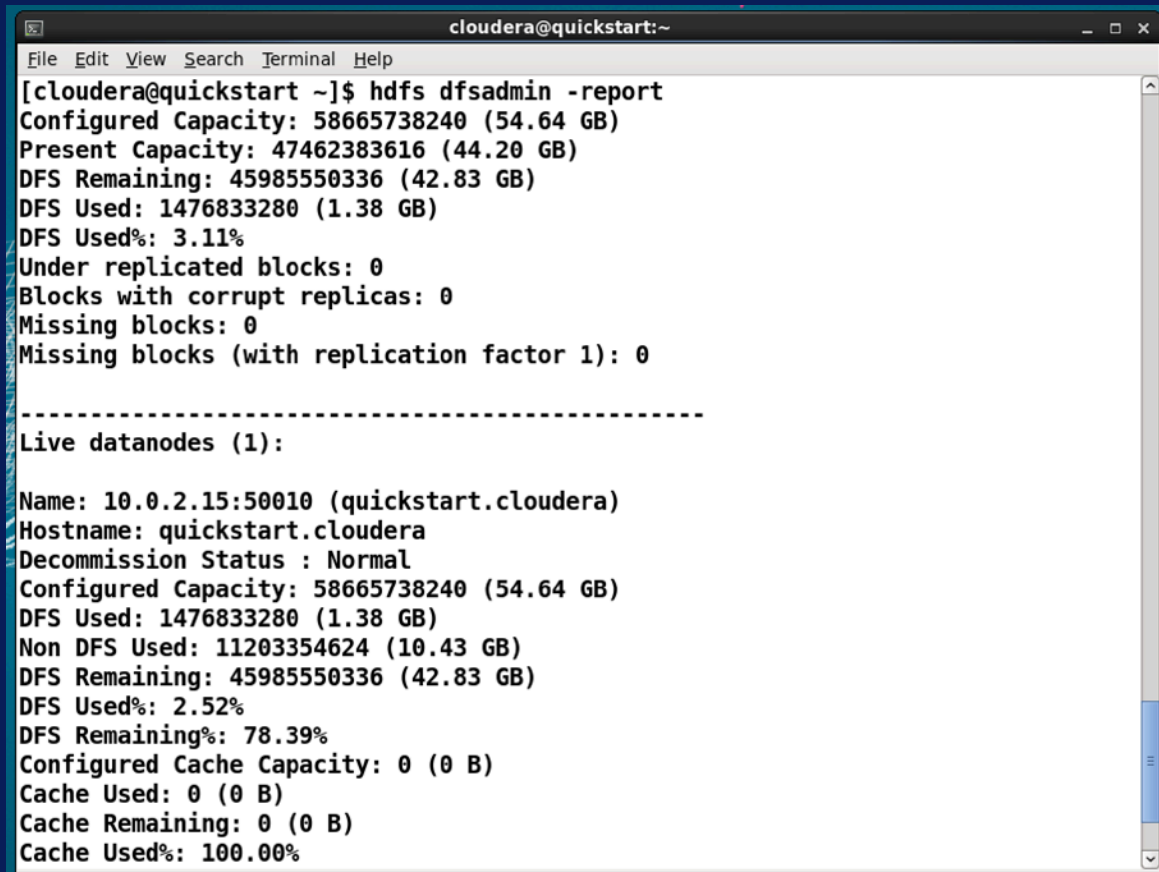
Command	Description
<code>-ls path</code>	Lists contents of directory
<code>-lsr path</code>	Recursive display of contents
<code>-du path</code>	Shows disk usage in bytes
<code>-dus path</code>	Summary of disk usage
<code>-mv src dest</code>	Move files or directories within HDFS
<code>-cp src dest</code>	Copy files or directories within HDFS
<code>-rm path</code>	Removes the file or empty directory in HDFS
<code>-rmr path</code>	Recursively removes file or directory
<code>-put localSrc dest</code> (Also <code>-copyFromLocal</code>)	Copy file from local filesystem into HDFS

Command	Description
<code>-get src localDest</code>	Copy from HDFS to local filesystem
<code>-cat filename</code>	Display contents of HDFS file
<code>-tail file</code>	Shows the last 1KB of HDFS file on stdout
<code>-chmod [-R]</code>	Change file permissions in HDFS
<code>-chown [-R]</code>	Change ownership in HDFS
<code>-help</code>	Returns usage info

HDFS Administrator Commands

Summary report:

hdfs dfsadmin -report

A terminal window titled 'cloudera@quickstart:~' showing the output of the 'hdfs dfsadmin -report' command. The output displays various HDFS statistics including capacity, usage, and replication status for the cluster and a specific datanode.

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
[cloudera@quickstart ~]$ hdfs dfsadmin -report  
Configured Capacity: 58665738240 (54.64 GB)  
Present Capacity: 47462383616 (44.20 GB)  
DFS Remaining: 45985550336 (42.83 GB)  
DFS Used: 1476833280 (1.38 GB)  
DFS Used%: 3.11%  
Under replicated blocks: 0  
Blocks with corrupt replicas: 0  
Missing blocks: 0  
Missing blocks (with replication factor 1): 0  
  
-----  
Live datanodes (1):  
  
Name: 10.0.2.15:50010 (quickstart.cloudera)  
Hostname: quickstart.cloudera  
Decommission Status : Normal  
Configured Capacity: 58665738240 (54.64 GB)  
DFS Used: 1476833280 (1.38 GB)  
Non DFS Used: 11203354624 (10.43 GB)  
DFS Remaining: 45985550336 (42.83 GB)  
DFS Used%: 2.52%  
DFS Remaining%: 78.39%  
Configured Cache Capacity: 0 (0 B)  
Cache Used: 0 (0 B)  
Cache Remaining: 0 (0 B)  
Cache Used%: 100.00%
```

Native Java API for HDFS

- *List main classes needed for HDFS access*
- *Additional classes and methods: IO, Configuration and path information*

Overview

- Base class:
org.apache.hadoop.fs.FileSystem
- Important classes:
FSDataInputStream
FSDataOutputStream
- Methods:
get, open, create

FSDataInputStream Methods

- *read* : read bytes
- *readFully* : read from stream to buffer
- *seek*: seek to given offset
- *getPos*: get current position in stream

FSDataOutputStream Methods

- *getPos*: get current position in stream
- *hflush*: flush out the data in client's user buffer.
- *close*: close the underlying output stream.

Reading from HDFS using API

- **get** an instance of `FileSystem`
`FileSystem fs = FileSystem.get(URI.create(uri),conf);`
- **Open** an input stream
`in = fs.open(new Path(uri));`
- Use IO utilities to **copy** from input stream
`IOUtils.copyBytes(in, System.out,4096,false);`
- **Close** the stream
`IOUtils.closeStream(in);`

Writing to HDFS using API

- **get** an instance of FileSystem
`FileSystem fs = FileSystem.get(URI.create(outuri),conf);`
- **Create** a file
`out = fs.create(new Path(outuri));`
- **Write** to output stream
`out.write(buffer, 0, nbytes);`
- **Close** the file
`out.close();`

WebHDFS REST API

- *List configuration options for WebHDFS*
- *Authenticate*
- *Perform file and directory operations*

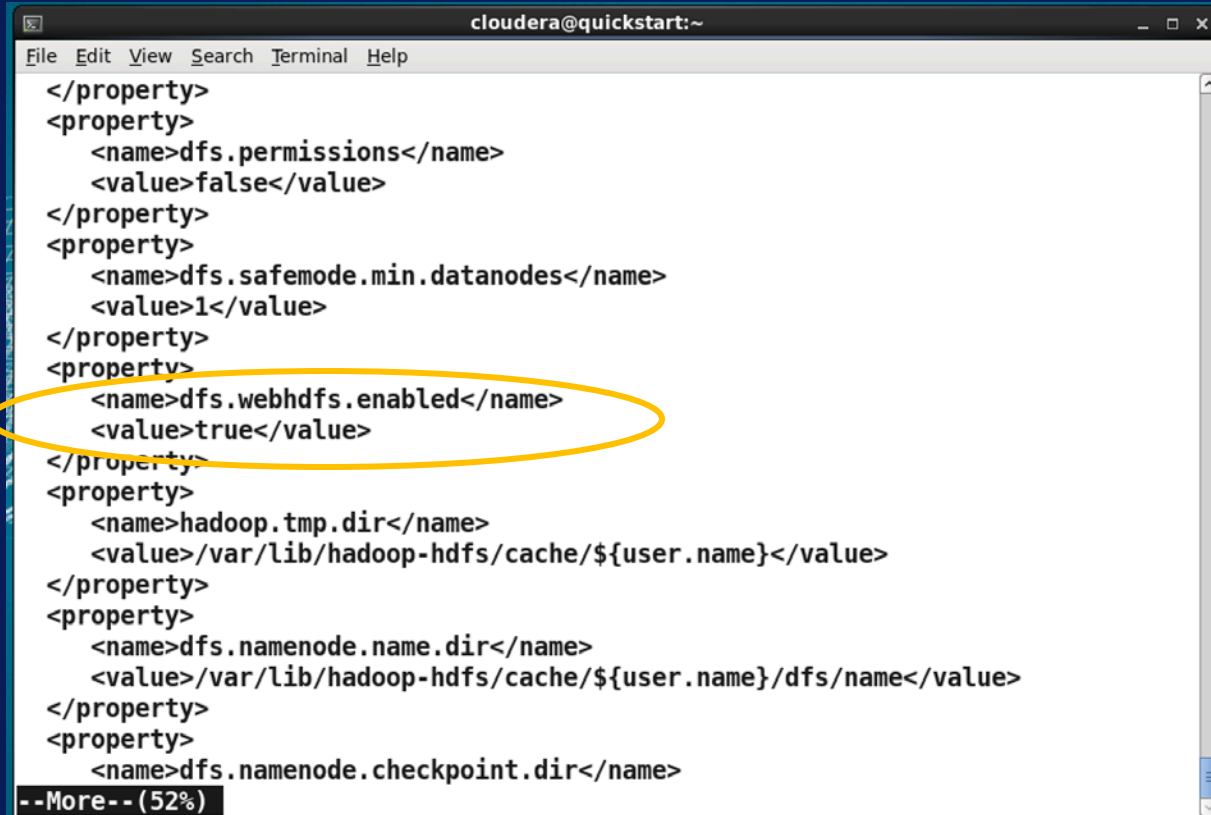
Enabling WebHDFS

*In **hdfs-site.xml***

- ***dfs.webhdfs.enabled***
- ***dfs.web.authentication.kerberos.principal***
- ***dfs.web.authentication.kerberos.keytab***

hdfs-site.xml

Command: **more /etc/hadoop/conf/hdfs-site.xml**



```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
</property>  
<property>  
  <name>dfs.permissions</name>  
  <value>>false</value>  
</property>  
<property>  
  <name>dfs.safemode.min.datanodes</name>  
  <value>1</value>  
</property>  
<property>  
  <name>dfs.webhdfs.enabled</name>  
  <value>true</value>  
</property>  
<property>  
  <name>hadoop.tmp.dir</name>  
  <value>/var/lib/hadoop-hdfs/cache/${user.name}</value>  
</property>  
<property>  
  <name>dfs.namenode.name.dir</name>  
  <value>/var/lib/hadoop-hdfs/cache/${user.name}/dfs/name</value>  
</property>  
<property>  
  <name>dfs.namenode.checkpoint.dir</name>  
--More-- (52%)
```

Authentication

- If security is off:

```
curl -i
```

```
"http://<HOST>:<PORT>/webhdfs/v1/<PATH>?  
[user.name=<USER>&]op=..."
```

- Security on with Kerberos:

```
curl -i --negotiate -u :
```

```
"http://<HOST>:<PORT>/webhdfs/v1/<PATH>?  
op=..."
```

- Security on using Hadoop delegation token:

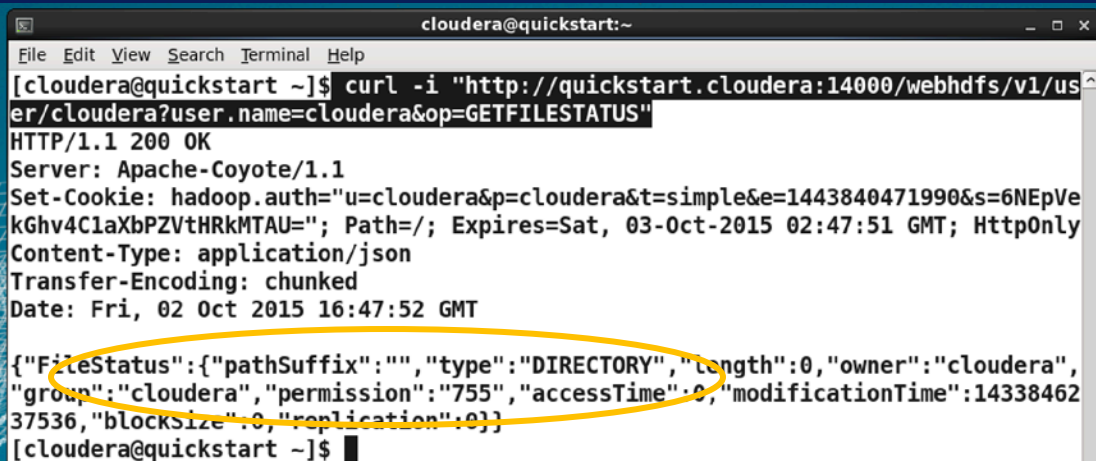
```
curl -i
```

```
"http://<HOST>:<PORT>/webhdfs/v1/<PATH>?  
delegation=<TOKEN>&op=..."
```


HTTP GET requests

curl -i

"http://quickstart.cloudera:14000/webhdfs/v1/user/cloudera?user.name=cloudera&op=GETFILESTATUS"



A terminal window titled 'cloudera@quickstart:~' showing the execution of a curl command. The command is: `curl -i "http://quickstart.cloudera:14000/webhdfs/v1/user/cloudera?user.name=cloudera&op=GETFILESTATUS"`. The output shows the HTTP response headers and the JSON body. The JSON body is highlighted with a yellow oval.

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
[cloudera@quickstart ~]$ curl -i "http://quickstart.cloudera:14000/webhdfs/v1/user/cloudera?user.name=cloudera&op=GETFILESTATUS"  
HTTP/1.1 200 OK  
Server: Apache-Coyote/1.1  
Set-Cookie: hadoop.auth="u=cloudera&p=cloudera&t=simple&e=1443840471990&s=6NEpVeKghv4C1aXbPZVtHRkMTAU="; Path=/; Expires=Sat, 03-Oct-2015 02:47:51 GMT; HttpOnly  
Content-Type: application/json  
Transfer-Encoding: chunked  
Date: Fri, 02 Oct 2015 16:47:52 GMT  
  
{  
  "FileStatus": {  
    "pathSuffix": "",  
    "type": "DIRECTORY",  
    "length": 0,  
    "owner": "cloudera",  
    "group": "cloudera",  
    "permission": "755",  
    "accessTime": 0,  
    "modificationTime": 1433846237536,  
    "blockSize": 0,  
    "replication": 0  
  }  
}  
[cloudera@quickstart ~]$
```

HTTP PUT requests

curl -i -X PUT

"http://quickstart.cloudera:14000/webhdfs/v1/user/test?user.name=cloudera&op=MKDIRS&permssion=755"

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
[cloudera@quickstart ~]$ curl -i -X PUT "http://quickstart.cloudera:14000/webhdfs/v1/user/test?user.name=cloudera&op=MKDIRS&permssion=755"  
HTTP/1.1 200 OK  
Server: Apache-Coyote/1.1  
Set-Cookie: hadoop.auth="u=cloudera&p=cloudera&t=simple&e=1443840967919&s=tf9fp0/nCB6bhYbtkE+az+kXa5E="; Path=/; Expires=Sat, 03-Oct-2015 02:56:07 GMT; HttpOnly  
Content-Type: application/json  
Transfer-Encoding: chunked  
Date: Fri, 02 Oct 2015 16:56:08 GMT  
  
{  
  "boolean": true  
}  
[cloudera@quickstart ~]$ hdfs dfs -ls /user/  
Found 7 items  
drwxr-xr-x - admin supergroup 0 2015-06-09 03:38 /user/admin  
drwxr-xr-x - cloudera cloudera 0 2015-06-09 03:37 /user/cloudera  
drwxr-xr-x - mapred hadoop 0 2015-06-09 03:37 /user/history  
drwxrwxrwx - hive hive 0 2015-06-09 03:37 /user/hive  
drwxrwxrwx - oozie oozie 0 2015-06-09 03:39 /user/oozie  
drwxr-xr-x - spark spark 0 2015-06-09 03:38 /user/spark  
drwxr-xr-x - cloudera supergroup 0 2015-10-02 09:56 /user/test  
[cloudera@quickstart ~]$
```

Create a local file

- Now lets create a local file and copy it into HDFS.
- We create a file with random data using the linux utility dd.
- Command:
`dd if=/dev/urandom of=sample.txt bs=64M count=16`
- Creates 1GB file called sample.txt on the local filesystem.

HTTP GET request on status

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
[cloudera@quickstart ~]$ dd if=/dev/urandom of=sample.txt bs=64M count=16  
16+0 records in  
16+0 records out  
1073741824 bytes (1.1 GB) copied, 93.9077 s, 11.4 MB/s  
[cloudera@quickstart ~]$ hdfs dfs -put sample.txt /user/test/  
[cloudera@quickstart ~]$ curl -i "http://quickstart.cloudera:14000/webhdfs/v1/user/test?user.name=cloudera&op=GETCONTENTSUMMARY"  
HTTP/1.1 200 OK  
Server: Apache-Coyote/1.1  
Set-Cookie: hadoop.auth="u=cloudera&p=cloudera&t=simple&e=1443842953016&s=Mzr81jkqtPawB0kxM/BNXmhbxjM="; Path=/; Expires=Sat, 03-Oct-2015 03:29:13 GMT; HttpOnly  
Content-Type: application/json  
Transfer-Encoding: chunked  
Date: Fri, 02 Oct 2015 17:29:13 GMT  
  
{  
  "ContentSummary": {"directoryCount": 1, "fileCount": 1, "length": 1073741824, "quota": 1, "spaceConsumed": 1073741824, "spaceQuota": -1}}  
[cloudera@quickstart ~]$
```

HTTP Operations

- ***HTTP GET:*** file status, checksums, attributes
- ***HTTP PUT:*** create, change ownership, rename, permissions, snapshot
- ***HTTP POST:*** append, concat
- ***HTTP DELETE:*** Delete files, snapshot