

HDFS FS shell コマンドチートシート

hdfs dfs / hadoop fs コマンドを網羅したチートシートです。全オプションは掲載していません。詳細はドキュメントをご確認下さい。

<https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/FileSystemShell.html>

※動作確認は Cloudera CDP 7.1.1 (Hadoop 3.1.1)にて実施

本チートシートは次のような形式を用います

コマンドのカテゴリー

| サブコマンド | 説明 |
|------------------|--------------------|
| hdfs dfs のサブコマンド | 説明と一部のオプションの紹介。 |
| (先頭のハイフンは省略) | 全てのオプションは記載していません。 |

コマンドの使用例と出力例 (#以降はコメント)

```
hdfs dfs -mkdir /tmp/testdir # comment
```

FS シェルコマンドの共通部分

| コマンドの指定形式 |
|------------------------|
| hdfs dfs -<サブコマンド> 引数 |
| hadoop fs -<サブコマンド> 引数 |

※上記 2 つのコマンドは同一

- ・サブコマンドにより引数は異なります
- ・HDFS には作業ディレクトリという概念はありません
- ・HDFS での標準的なホームディレクトリは /user/<username>です。
- ・Amazon S3 や Azure Storage、Google Cloud StorageCS 等のオブジェクトストアに対する操作も可能ですが、制約があります。詳細はドキュメントをご覧ください。

例:

```
hdfs dfs -cp a.txt /tmp/b.txt
hdfs dfs -cp a.txt file:///tmp/
hdfs dfs -cp a.txt hdfs://nn.example.com/tmp/dir1
hdfs dfs -cp /tmp/a.txt s3a://bucket/datasets
hdfs dfs -cp /tmp/a.txt ¥
wasb://yourcontainer@youraccount.blob.core.windows.net/hello.txt
hdfs dfs -cp /tmp/a.txt gs://bucket/object
```

ディレクトリ関連コマンド

| サブコマンド | 説明 |
|--------|--|
| mkdir | -mkdir [-p] <paths> ディレクトリを作成 -p オプションを指定すると指定したパスの親ディレクトリも作成する |
| rmdir | -rmdir [--ignore-fail-on-non-empty] URI [URI ...] ディレクトリを削除。サブディレクトリやファイルが含まれている場合はエラーになる --ignore-fail-on-non-empty オプションを指定すると、ディレクトリにファイルが含まれていてもエラーにならない |

使用例

```
# ディレクトリを作成
hdfs dfs -mkdir /tmp/testdir
hdfs dfs -mkdir -p /tmp/sudir1/subdir2 #階層
hdfs dfs -mkdir testdirhome

# ディレクトリを削除
hdfs dfs -rmdir /tmp/testdir
hdfs dfs -rmdir /tmp/testdir1/testdir2
hdfs dfs -rmdir testdirhome
```

ディレクトリ、ファイル表示関連コマンド

| サブコマンド | 説明 |
|--------|---|
| ls | -ls [-C] [-d] [-h] [-q] [-R] [-t] [-S] [-r] [-u] [-e] <args> 指定したパスの内容の一覧。パス省略時はホームディレクトリになる。 多くのオプションが提供されているが、フォルダを再帰的に表示する -R オプションは便利 [出力] ・ファイルの場合の出力 権限 複製数 UID GID サイズ 変更日 変更時間 ファイル名 ・ディレクトリの場合の出力 権限 UID GID 変更日 変更時間 ディレクトリ名 |

| | |
|-------------|---|
| lsr | Deprecated hdfs dfs -ls -R と同じ |
| find | -find <path> ... <expression> ... 条件に一致したファイルを探して指定した操作を行う。条件は複数指定可能 [条件] -name ワイルドカードの指定が可能 -iname 大文字小文字を区別しない [操作] -print (デフォルト) -print0 最後に ASCII の NULL 文字を付与 |
| cat | -cat [-ignoreCrc] URI [URI ...] ファイルの内容を表示 |
| head | -head URI 最初の 1kb を表示 |
| tail | -tail [-f] URI 最後の 1kb を表示 -f オプションは、UNIX の tail コマンド同様にファイルに追記されたデータを出し続ける |
| text | -text <src> テキスト形式でファイルを出力 zip ファイルの場合には便利 |

head は hdfs dfs -cat <filename> |head で代用可能

使用例

```
# ls
hdfs dfs -ls /tmp/subdir1
Found 4 items
-rw-r--r--  3 training supergroup      26882
2020-06-11 23:03 /tmp/subdir1/access.log
-rw-r--r--  3 training supergroup       110 2020-
06-11 23:03 /tmp/subdir1/b.txt
-rw-r--r--  3 training supergroup       110 2020-
06-11 23:03 /tmp/subdir1/c.txt
drwxr-xr-x  - training supergroup        0 2020-
06-11 23:02 /tmp/subdir1/subdir2

# find
hdfs dfs -find -name "*.txt" -print
contents/BTESTB.txt
contents/TESTA.txt
```

```
contents/TESTB.txt
contents/TSTC.txt
contents/a.txt

# find -name
hdfs dfs -find -iname "*TEST*" -print
contents/BTESTB.txt
contents/TESTA.txt
contents/TESTB.txt
contents/atesta.txt
contents/test.zip
contents/testa.txt
contents/testb.txt

# cat/head/tail/text
hdfs dfs -cat contents/a.txt

1111111111
2222222222
3333333333
4444444444
5555555555
6666666666
7777777777
8888888888
9999999999
0000000000

hdfs dfs -head access.log
<snip>
hdfs dfs -tail access.log
<snip>
hdfs dfs -text test.zip
<snip>
```

ファイル操作関連コマンド

| サブコマンド | 説明 |
|----------------------|---|
| put | -put [-f] [-p] [-l] [-d] [- |
| copyFromLocal | <localsrc1> ..]. <dst> ファイルを指定したパスにアップロード。パス省略時はホームディレクトリになる。ファイルが存在する場合は -f オプションを指定しな |

| | |
|----------------------------------|---|
| | いとエラーになる。 役立つオプション -f 上書き -p アクセス、更新時間、所有者、権限を保持 -l 複製係数 1 でアップロード。アップロードに時間がかかる場合はこのオプションを使用して、後から複製係数を変更すると便利 -d 一時ファイルを作成しない |
| moveFromLocal | -moveFromLocal <localsrc> <dst> put/copyFromLocal と同様だが、アップロード後にローカルファイルを削除する |
| get copyToLocal | -get [-ignorecrc] [-crc] [-p] [-f] <src> <localdst> 指定したファイルをローカルにダウンロード。チェックサムをチェックするがオプションにより無視することもできる -p 属性を保持する -f 既存ファイルがある場合は上書きする |
| moveToLocal | 実は現時点でまだ実装されていない |
| getmerge | -getmerge [-nl] <src> <localdst> 宛先ディレクトリを指定すると、ディレクトリ内にあるファイル群をローカルの単一ファイルにマージして保存。 MapReduce や Spark などの処理後に作成された複数ファイルをまとめた場合に役立つことがある -nl マージするファイルごとに、LF (改行) 文字を付与する |
| cp | -cp [-f] [-p -p[topax]] URI [URI ...] <dest> 元ファイルを宛先にコピーする。複数ファイルを指定可能だが、その場合宛先はディレクトリを指定する。ファイルが存在している場合はエラーになる -f ファイルが存在している場合は上書きする -p 元ファイルの属性を保持 (オプションで保持する属性をそれぞれ指定できる) |
| mv | -mv URI [URI ...] <dest> 元 (複数) ファイルを指定した宛先に移動する。単一ファイルの場合はリネームになる |
| rm | -rm [-f] [-r -R] [-skipTrash] [- |

| | |
|---------------------|--|
| | safely] URI [URI ...] 指定したファイルを削除 ゴミ箱機能が有効な場合は削除ではなく、ゴミ箱に移動する (ゴミ箱からの削除は expunge コマンド) -R 再帰的にファイルとディレクトリを削除 -r -R オプションと同一 -skipTrash 削除時にゴミ箱をスキップする -f ファイルが存在しない場合にエラーを表示しない -safety hadoop.shell.delete.limit.num.files プロパティ値と比較し、それより大きい数のファイルを含むディレクトリの場合は安全確認をする。誤って大量のファイルの削除を防ぐことができる |
| rmr | Deprecated hdfs dfs -rm -R と同じ |
| touchz | -touchz URI [URI ...] ゼロバイトのファイルを作成する ファイルが存在している場合はエラーになる |
| truncate | -truncate [-w] <length> <paths> 指定したサイズでファイルを切り詰める -w オプションの指定で完了するまで待つ |
| appendToFile | -appendToFile <localsrc> ... <dst> 対象ファイルにローカルファイルを追記する 複数のローカルファイルの指定や標準入力から内容を渡すことも可能 |

copyFromLocal、copyToLocal は put/get 登場前からあるコマンドで、現在ほとんど使用することはない。同様に moveFromLocal、moveToLocal も使用頻度は低い (後者は現時点で実装されていない)。

使用例

```
# put
hdfs dfs -put a.txt
hdfs dfs -put a.txt /tmp
hdfs dfs -moveFromLocal /tmp/a.txt

# get
hdfs dfs -get a.txt

# getmerge
hdfs dfs -getmerge /tmp/testdir
```

```

# cp/mv
hdfs dfs -cp /tmp/a.txt /tmp/b.txt
hdfs dfs -cp /tmp/a.txt /tmp/b.txt /tmp/newdir
hdfs dfs -mv /tmp/a.txt /tmp/b.txt /tmp/newdir
hdfs dfs -mv /tmp/a.txt /tmp/c.txt

# rm
hdfs dfs -rm /tmp/c.txt
hdfs dfs -rm -R /tmp/newdir          #再帰
hdfs dfs -rm -skipTrash /tmp/b.txt   #ゴミ箱をスキップ

# ゼロバイトのファイルを作成
hdfs dfs -touchz /tmp/zero.txt

# 切り詰め
hdfs dfs -truncate 3 /tmp/a.txt

# 追記
hdfs dfs -appendToFile /tmp/a.txt local.txt
cat /etc/hosts | hdfs dfs -appendToFile /tmp/a.txt
- #stdin

```

ファイルメタ情報表示コマンド

| サブコマンド | 説明 | | | | | | | | |
|--------------|---|--------------|------------|--------------|----------|----|----|---------|------|
| count | <p>-count [-q] [-h] [-v] [-x] [-t [<storage type>]] [-u] [-e] <paths></p> <p>指定したパスにあるディレクトリ数、ファイル数、サイズを表示</p> <p>多くのオプションがあるが、一部は次の通り</p> <ul style="list-style-type: none"> -u クォータと使用量のみを表示 -q クォータのみを表示 -h 見やすいフォーマットで表示 (便利) -x スナップショットを除外 -e イレジャーコーディングポリシーを表示 -v ヘッダを表示 <p>[出力例]</p> <pre>hdfs dfs -count -v -h /tmp</pre> <table border="1"> <thead> <tr> <th>DIR_COUNT</th> <th>FILE_COUNT</th> <th>CONTENT_SIZE</th> <th>PATHNAME</th> </tr> </thead> <tbody> <tr> <td>45</td> <td>31</td> <td>257.7 M</td> <td>/tmp</td> </tr> </tbody> </table> | DIR_COUNT | FILE_COUNT | CONTENT_SIZE | PATHNAME | 45 | 31 | 257.7 M | /tmp |
| DIR_COUNT | FILE_COUNT | CONTENT_SIZE | PATHNAME | | | | | | |
| 45 | 31 | 257.7 M | /tmp | | | | | | |
| df | <p>-df [-h] URI [URI ...]</p> <p>空きスペースを表示</p> <ul style="list-style-type: none"> -h 見やすい形式で表示 <p>[出力例]</p> <pre>hdfs dfs -df -h /tmp</pre> | | | | | | | | |

| Filesystem | Size | Used | Available | Use% |
|----------------|--------|-------|-----------|------|
| hdfs://nn:8020 | 76.0 G | 2.3 G | 42.6 G | 3% |

| | |
|-------------|---|
| du | <p>-du [-s] [-h] [-v] [-x] URI [URI ...]</p> <p>指定したファイルのサイズ、またはディレクトリ内のファイルのサイズを表示</p> <p>オプション一例</p> <ul style="list-style-type: none"> -s 個々のファイルではなく合計サマリを表示 -h 見やすい形式で表示 -v ヘッダを表示 -x スナップショットを除外 <p>[出力例]</p> <pre>hdfs dfs -du -h -s /tmp</pre> <pre>257.7 M 1.1 G /tmp</pre> |
| du | <p>Deprecated</p> <p>hdfs dfs -du -s と同じ</p> |
| stat | <p>-stat [format] <path> ...</p> <p>指定したパスの統計情報を指定したフォーマットで表示</p> <p>フォーマットで指定できる文字 (デフォルトは%y)</p> <p>権限: (%a)8進数 (%A)シンボル</p> <p>ファイルサイズ (バイト): %b</p> <p>種類: %F</p> <p>グループ: %g</p> <p>名前: %n</p> <p>ブロックサイズ: %o</p> <p>複製係数: %r</p> <p>所有者のユーザー名: %u</p> <p>アクセス時間: %x, %X (大文字は1970/1/1からのミリ秒、小文字はUTCの読みやすい形式)</p> <p>変更時間: %Y, %Y (大文字は1970/1/1からのミリ秒、小文字はUTCの読みやすい形式)</p> <p>[出力例]</p> <pre>hdfs dfs -stat /tmp/access.log</pre> <pre>2020-06-12 06:04:20</pre> |
| test | <p>-test -[defswrz] URI</p> <p>ファイルに対するテストを行い、戻り値でその結果を返す</p> <ul style="list-style-type: none"> -d: ディレクトリの場合は0 -e: パスが存在する場合は0 -f: ファイルの場合は0 |

```

-s: パスが空ではない場合は 0
-w: パスが存在し書き込み権限がある場合は 0
-r: パスが存在し読み取り権限がある場合は 0
-z: ファイルがゼロバイトの場合は 0

[例]
hdfs dfs -test -e /tmp/access.log

echo $?

0

```

使用例

```

# count
hdfs dfs -count /tmp

      45      31      270195979 /tmp

hdfs dfs -count -v -h /tmp

  DIR_COUNT  FILE_COUNT  CONTENT_SIZE  PATHNAME
      45      31      257.7 M      /tmp

# df
hdfs dfs -df /tmp

Filesystem                               Size
Used    Available  Use%
hdfs://localhost.localdomain:8020  81622174925
2514911514  45732405248    3%

hdfs dfs -df -h /tmp

Filesystem                               Size  Used
Available  Use%
hdfs://localhost.localdomain:8020  76.0 G  2.3 G
42.6 G    3%

# du
hdfs dfs -du /tmp

110      330      /tmp/BTESTB.txt
110      330      /tmp/TESTA.txt
110      330      /tmp/TESTB.txt
110      330      /tmp/TSTC.txt
110      330      /tmp/a.txt
110      330      /tmp/a1.txt
10       402653184 /tmp/access.log

hdfs dfs -du -h -s /tmp

257.7 M  1.1 G  /tmp

```

```

# stat
hdfs dfs -stat /tmp/access.log

2020-06-12 06:04:20

hdfs dfs -stat "type:%F size:%b name:%n"
/tmp/access.log

type:regular file size:10 name:access.log

# test
hdfs dfs -test -e /tmp/access.log #存在する

0

hdfs dfs -test -e /tmp/A.TXT      #存在しない

1

```

ファイル属性関連コマンド

| サブコマンド | 説明 |
|-----------------|--|
| checksum | <p><code>-checksum URI</code></p> <p>指定したファイルのチェックサムを表示</p> <p>[出力例]</p> <pre>hdfs dfs -checksum /tmp/a.txt /tmp/a.txt MD5-of-0MD5-of- 512CRC32C 00000200000000000000000000000487 74ac2c33d72e14d692c00e7bfc722</pre> |
| chown | <p><code>-chown [-R] [OWNER] [:[GROUP]]</code></p> <p>URI [URI...]</p> <p>ファイルの所有者を変更する(グループも可能)。 変更にはスーパーユーザー権限が必要。 <code>-R</code> オプションを指定して再起的に変更 詳細はドキュメント参照</p> |
| chgrp | <p><code>-chgrp [-R] GROUP URI [URI ...]</code></p> <p>ファイルのグループを変更 変更にはスーパーユーザー権限が必要。 <code>-R</code> オプションを指定して再起的に変更</p> |
| chmod | <p><code>-chmod [-R] <MODE[,MODE]... </code> OCTALMODE> URI [URI ...]</p> <p>ファイルの属性を変更 変更は所有者またはスーパーユーザー権限が必要。 <code>-R</code> オプションを指定して再起的に変更</p> |
| touch | <p><code>touch [-a] [-m] [-t TIMESTAMP] [-c]</code></p> |

URI [URI ...]

指定したファイルのアクセス時間、更新時間を変更し、ファイルが存在しない場合はゼロ長のファイルを作成する。時刻省略時は現在時刻が使用される

- a: アクセス時間のみを変更
- m: 更新時間のみを変更
- t 指定した時刻に変更 *1
- c ファイルが存在しない場合は作成しない

*1 ドキュメントにバグがあり、時間指定は yyyyMMDD:HHmmdd 形式にしないとエラーになる (日付と時間の間に:が必要)

使用例

```
# checksum/chown/chgrp/chmod/touch
hdfs dfs -checksum /tmp/a.txt
/tmp/a.txt      MD5-of-0MD5-of-512CRC32C
                000002000000000000000000048774ac2c33d72e14d
                692c00e7bfc722

sudo -u hdfs hdfs dfs -chown hdfs /tmp/a.txt
sudo -u hdfs hdfs dfs -chown testuser:testgrp
/tmp/a.txt

sudo -u hdfs hdfs dfs -chgrp testgrp /tmp/b.txt
hdfs dfs -ls /tmp/a.txt /tmp/b.txt
-rw-r--r--  3 testuser testgrp      110 2020-06-11
23:04 /tmp/a.txt
-rw-r--r--  3 training testgrp      110 2020-06-11
23:04 /tmp/b.txt

sudo -u hdfs hdfs dfs -chmod 666 /tmp/a.txt
-rw-rw-rw-  3 testuser testgrp      110 2020-06-11
23:04 /tmp/a.txt

hdfs dfs -touch /tmp/z.txt
```

運用系コマンド

| サブコマンド | 説明 |
|----------------|---|
| expunge | -expunge [-immediate] しきい値よりも古いチェックポイントのファイルをゴミ箱から削除してチェックポイントを更新(即時にゴミ箱全てを消すのではないことに注意。あくまでもチェックポイントより古いファイルが対象) -immediate オプションによりゴミ箱のファイル |

ルを即削除できる

```
setrep -setrep [-R] [-w] <numReplicas>
<path>
指定した数値にファイルの複製係数を変更する
-R オプションはディレクトリを再帰的に辿る
-w オプションは完了するまで待つ
```

使用例

```
# expunge
hdfs dfs -expunge
hdfs dfs -expunge -immediate

# setrep
hdfs dfs -ls /tmp/a.txt
-rw-rw-rw-  3 testuser testgrp      110 2020-06-
11 23:04 /tmp/a.txt
hdfs dfs -setrep 5 /tmp/a.txt
Replication 5 set: /tmp/a.txt
hdfs dfs -ls /tmp/a.txt
-rw-rw-rw-  5 testuser testgrp      110 2020-06-
11 23:04 /tmp/a.txt
```

ACL(アクセス制御)関連コマンド

| サブコマンド | 説明 |
|-----------------|---|
| getfacl | -getfacl [-R] <path> ファイルとディレクトリのアクセス制御リストを表示。 -R オプションにより再帰的に辿ることが可能 [出力例] |
| getfattr | -getfattr [-R] -n name -d [-e en] <path> (存在すれば)ファイルまたはディレクトリの拡張属性の名前と値を表示 -R: 再帰的に表示 -n name: 名前付き属性値をダンプ -d: 全ての拡張属性をダンプ -e encoding: エンコードを指定 (text, hex, base64 のいずれか) [出力例] |
| setfacl | -setfacl [-R] [-b -k -m -x |

```
<acl_spec> <path>] | [--set
<acl_spec> <path>]
ファイルまたはディレクトリにアクセス制御リス
トを設定
基本的に Linux の ACL で使用する引数と同等
```

```
setfattr -setfattr -n name [-v value] | -
x name <path>
ファイルまたはディレクトリに拡張属性を設定
-n 拡張属性の名前
-v 拡張属性の値
-x 拡張属性を削除
```

使用例

```
# ACL
sudo -u hdfs hdfs dfs -setfacl -m user:hadoop:rw- /tmp/a.txt
sudo -u hdfs hdfs dfs -setfacl -x user:hadoop /tmp/a.txt
sudo -u hdfs hdfs dfs -setfacl -b /tmp/a.txt
sudo -u hdfs hdfs dfs -setfacl -k /tmp/subdirA
sudo -u hdfs hdfs dfs -setfacl --set
user::rw-,user:hadoop:rw-,group::r--,other::r-- /tmp/a.txt
sudo -u hdfs hdfs dfs -setfacl -R -m user:hadoop:r-x
/tmp/subdirA
sudo -u hdfs hdfs dfs -setfacl -m default:user:hadoop:r-x
/tmp/subdirA
hdfs dfs -getfacl /tmp/a.txt
# file: /tmp/a.txt
# owner: testuser
# group: testgrp
user::rw-
user:hadoop:rw-
group::r--
mask::rw-
other::r--
# xattr
sudo -u hdfs hdfs dfs -setfattr -n user.myAttr -v myValue
/tmp/a.txt
sudo -u hdfs hdfs dfs -setfattr -n user.novalue /tmp/b.txt
hdfs dfs -getfattr -d /tmp/a.txt
# file: /tmp/a.txt
user.myAttr="myValue"
```

```
hdfs dfs -getfattr -d /tmp/b.txt
# file: /tmp/b.txt
user.novalue
```

その他のコマンド

| サブコマンド | 説明 |
|--------------|--|
| usage | -usage command 個々のコマンドのヘルプを返す |
| help | -help command コマンドの用法を出力する ドキュメントバグだと思われるが、-help の後 にコマンドを指定するとコマンドのヘルプが表示される |

使用例

```
# usage
hdfs dfs -usage cat
Usage:  hadoop fs [generic options] -cat [-
ignoreCrc] <src> ...
# help
hdfs dfs -help cat
-cat [-ignoreCrc] <src> ... :
Fetch all files that match the file pattern <src>
and display their content on
stdout.
```

注: スナップショット関連コマンドは別のチートシートにまとめて記載
予定です。

```
createSnapshot
deleteSnapshot
renameSnapshot
```

なお、コマンドの実際の出力例は下記のログを参照して下さい

<https://gist.github.com/kawamon/93e48232affe86523b3aec23bfa0dc3a>