More about HDFS Shell Commands

This reading supplies some additional information about HDFS and the Hadoop File System Shell that you should know.

The -mkdir Command

One useful command that was not mentioned in the previous video is hdfs dfs -mkdir, which can be used to create one or more directories in HDFS. This command expects that the parent directory of the directory you are creating already exists, so if you want to use this command to create nested directories, start first by creating the highest-level parent directory, then create the next-level child subdirectory, and so on. Alternatively, you can use the -p option (short for *parent*) to automatically create any necessary parent directories of the specified directory if they do not already exist.

More HDFS Command Options

Some of the commands that you did learn about in this course can take additional command-line options that were not described. For example, when using the when using hdfs dfs -rm command, you can specify the -r option (short for *recursive*) to delete the specified directory and all files and subdirectories under it. The syntax is:

```
hdfs dfs -rm -r /path/to/directory/
```

Be extremely careful when using this -r option! It is easy to inadvertently delete a huge amount of data by misspecifying the directory path.

Full List of HDFS Shell Commands, Options, and Arguments

You can see a complete list of the Hadoop File System Shell commands and the options and arguments they accept on this web page: https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/FileSystemShell.html. Many of these you will never need to use, but this is a good resource to find ways to do things you do need. As you look through that page, remember that hdfs dfs is the same as hadoop fs—you can use either.

Some of the commands on that page relate to permissions and file ownership, which is beyond the scope of this course, but see the “HDFS Permissions” reading if you’re interested to learn more about that.

HDFS Paths

In this course, most of the HDFS paths you will see in hdfs dfs commands are in the form /*path*/*to*/*directory*/ (for a directory), or /*path*/*to*/*directory*/*file.ext* (for a file). This is known as an *unqualified path* because it does not specify what *protocol* to use and it does not specify what specific instance of HDFS to use. In most cases, it is sufficient to use unqualified HDFS paths, because the configuration of your big data environment will specify what protocol and what HDFS instance to use.

However, in some cases, depending on the configuration of your big data environment, it might be necessary to *fully qualify* your HDFS paths by specifying the hdfs:// protocol and a hostname indicating what instance of HDFS to use. To do this, use the form: hdfs://*hostname*/*path*/*to*/*directory*/*file.ext* or hdfs://*hostname*.*domain*/*path*/*to*/*directory*/*file.ext*. Ask your system administrator what hostname and what domain (if any) to use.

You can also specify the hdfs:// protocol without specifying a hostname, using the form: hdfs:///*path*/*to*/*directory*/*file.ext*. Notice the *three* slashes after hdfs: The first two slashes are part of the protocol, and the third slash is the start of the path.

HDFS Trash

Note that HDFS has a “trash” directory for recently deleted files, similar to the Trash on iOS or Windows systems. This trash is not always enabled, so you should check your system to see if it's enabled before assuming that you can recover any deleted files!

The hdfs dfs -rm command has a -skipTrash option that you can use to bypass the trash (if it’s enabled) and delete the file immediately. When deleting large files to free up space in HDFS, consider using this option so that you do not need to perform the additional step of emptying the trash. But remember that when you use this option, the files you delete will not be recoverable.