

# Hortonworks Data Platform

## HBase Snapshots

(March 2, 2016)

## Hortonworks Data Platform: HBase Snapshots

Copyright © 2012-2016 Hortonworks, Inc. Some rights reserved.

The Hortonworks Data Platform, powered by Apache Hadoop, is a massively scalable and 100% open source platform for storing, processing and analyzing large volumes of data. It is designed to deal with data from many sources and formats in a very quick, easy and cost-effective manner. The Hortonworks Data Platform consists of the essential set of Apache Hadoop projects including MapReduce, Hadoop Distributed File System (HDFS), HCatalog, Pig, Hive, HBase, ZooKeeper and Ambari. Hortonworks is the major contributor of code and patches to many of these projects. These projects have been integrated and tested as part of the Hortonworks Data Platform release process and installation and configuration tools have also been included.

Unlike other providers of platforms built using Apache Hadoop, Hortonworks contributes 100% of our code back to the Apache Software Foundation. The Hortonworks Data Platform is Apache-licensed and completely open source. We sell only expert technical support, [training](#) and partner-enablement services. All of our technology is, and will remain free and open source.

Please visit the [Hortonworks Data Platform](#) page for more information on Hortonworks technology. For more information on Hortonworks services, please visit either the [Support](#) or [Training](#) page. Feel free to [contact us](#) directly to discuss your specific needs.



Except where otherwise noted, this document is licensed under  
**Creative Commons Attribution ShareAlike 4.0 License.**  
<http://creativecommons.org/licenses/by-sa/4.0/legalcode>

# Table of Contents

- 1. HBase Snapshots ..... 1
  - 1.1. Configuration ..... 1
  - 1.2. Take a Snapshot ..... 1
  - 1.3. List Snapshots ..... 1
  - 1.4. Delete Snapshots ..... 1
  - 1.5. Clone a Table from a Snapshot ..... 2
  - 1.6. Restore a Snapshot ..... 2
  - 1.7. Snapshot Operations and ACLs ..... 2
  - 1.8. Export to Another Cluster ..... 2

# 1. HBase Snapshots

Prior to version 0.94.6, the only way to backup or clone a table was to use CopyTable/ExportTable, or to copy all of the hfiles in HDFS after disabling the table. The disadvantage of these methods is that you can degrade region server performance or you must disable the table, which means no reads or writes can occur.

HBase Snapshots allow you to take a snapshot of a table without much impact on Region Servers. Snapshot, clone, and restore operations don't involve data copying. In addition, exporting a snapshot to another cluster has no impact on region servers.

## 1.1. Configuration

To turn on snapshot support, set the `hbase.snapshot.enabled` property to true. (Snapshots are enabled by default in 0.95+, and are off by default in 0.94.6+.)

```
<property>
  <name>hbase.snapshot.enabled</name>
  <value>true</value>
</property>
```

## 1.2. Take a Snapshot

You can take a snapshot of a table regardless of whether it is enabled or disabled. The snapshot operation doesn't involve any data copying. As shown in the following example, start the HBase shell, and then clone the table:

```
$ hbase shell
hbase> snapshot 'myTable', 'myTableSnapshot-122112'
```

## 1.3. List Snapshots

List all snapshots taken (by printing the names and relative information).

```
$ hbase shell
hbase> list_snapshots
```

## 1.4. Delete Snapshots

You can remove a snapshot, and the files associated with that snapshot will be removed if they are no longer needed.

```
$ hbase shell
hbase> delete_snapshot 'myTableSnapshot-122112'
```

## 1.5. Clone a Table from a Snapshot

From a snapshot you can create a new table (clone operation) with the same data that you had when the snapshot was taken. The clone operation does not involve data copies. A change to the cloned table does not impact the snapshot or the original table.

```
$ hbase shell
hbase> clone_snapshot 'myTableSnapshot-122112', 'myNewTestTable'
```

## 1.6. Restore a Snapshot

The restore operation requires the table to be disabled, and the table will be restored to its state when the snapshot was taken, changing both data and schema if required.

```
$ hbase shell
hbase> disable 'myTable'
hbase> restore_snapshot 'myTableSnapshot-122112'
```



### Note

Because Replication works at the log level and snapshots work at the file system level, after a restore, the replicas will be in a different state than the master. If you want to use restore, you need to stop replication and redo the bootstrap.

In case of partial data loss due to client issues, you can clone the table from the snapshot and use a Map-Reduce job to copy the data that you need from the clone to the main one (instead of performing a full restore, which requires the table to be disabled).

## 1.7. Snapshot Operations and ACLs

If you are using security with the AccessController Coprocessor, only a global administrator can take, clone, or restore a snapshot. None of these actions capture ACL rights. Restoring a table preserves the ACL rights of the existing table, while cloning a table creates a new table that has no ACL rights until the administrator adds them.

## 1.8. Export to Another Cluster

The ExportSnapshot tool copies all the data related to a snapshot (hfiles, logs, and snapshot metadata) to another cluster. The tool executes a Map-Reduce job, similar to `distcp`, to copy files between the two clusters. Because it works at the file system level, the HBase cluster does not have to be online. The HBase Snapshot Export tool must be run as the **hbase** user. The HBase Snapshot Export tool uses the temp directory specified by `hbase.tmp.dir` (for example, `/grid/0/var/log/hbase`), created on HDFS with **hbase** user as the owner.

To copy a snapshot called `MySnapshot` to an HBase cluster `srv2` (`hdfs://srv2:8020/hbase`) using 16 mappers:

```
$ hbase org.apache.hadoop.hbase.snapshot.ExportSnapshot -snapshot MySnapshot -  
copy-to  
hdfs://yourserver:8020/hbase_root_dir -mappers 16
```