Novell Storage Services: Pool Snapshots

Overview

A pool snapshot is a metadata copy of a storage data pool. The snap preserves a point-in-time view of a data pool, and supports data backup and recovery for that pool. Unlike a file-level snapshot that is created and destroyed dynamically as a file is opened and closed, you can create, manage, and delete a pool-level snapshot for any pool on your server.

You can take a snapshot, and activate it later to access the original pool’s data as it existed at the time of the snap. Both the pool and its snapshot(s) can be active and available concurrently. You access data on the active pool snapshot just as you would any other pool, even while data is changing on the original pool you snapped.

Benefits of Pool Snapshots

Pool snapshots save time and data. They provide an instant copy of data that can help expedite routine maintenance procedures to backup, archive, and protect data. Because traditional methods of duplicating large amounts of data can be expensive and time-consuming, the efficiency of snapshots can be an important benefit for your enterprise. You can make snaps as frequently as needed to meet your data availability and resilience requirements.

You can use pool snapshots in a variety of ways to enhance your current storage infrastructure, including the following scenarios:

- **Supporting Backup Operations**
  A pool snapshot facilitates non-disruptive backups because the snap becomes the source of the backup. As contrasted to a traditional, full-data copy of the pool, the metadata copy only takes a moment to create and occurs transparently to the end user. With traditional backups, applications may be shut down throughout the backup routine. In comparison, the pool snapshot process makes the original pool available with almost imperceptible delay.

- **Archiving Data**
  You can archive pool snapshots to capture a history of the changes made to the original data pool.

- **Restoring Data**
  Pool snapshots can serve as a source for restoring information. Two common reasons to restore information are user error and application errors.

  - A user may inadvertently make changes to a file that need to be reversed. Files can become corrupted or deleted. The pool snapshot provides a quick and easy way to locate and reinstate selected files.

  - An application may be infected by a virus or be corrupted by other problems, causing the application to store erroneous data throughout the pool. With a pool snapshot, you easily can easily effect a rollback of all or part of the original pool to a point in time before the virus or problem was known to exist in the system.

- **Recreating Operational and Development Environments**
  You can also write to the pool snapshot, just as you would any pool. You can work with and modify the snap version of the data. For example, in a software development environment, engineers may want to repeat builds and tests of data within a given snap.

- **Testing and Training**
  Snaps can provide a convenient source for testing and training environments and for data mining purposes.

How the Pool Snapshot Works

The pool snapshot function uses copy-on-write technology to enable the instantaneous snapshot of a pool, while requiring only a fraction of the storage space of the original data pool. A pool snapshot does not save an exact copy of the original data pool. Instead, the snap is a metadata-based copy that stores only those blocks of data that change subsequent to the instant of the snap. The snap combines the metadata with the unchanged data on the original pool to provide a virtual image of an exact copy of the data at the instant the snapshot was taken, plus any end-user modifications made to that snap.

Before the snap can occur, the snapshot function must render the original pool quiescent by briefly halting all data transaction activity when current transactions complete. It temporarily prevents new writes to the pool and flushes the file system cache to make the pool current with existing writes. Then, it snaps the now stable pool, and allows data transaction activity to resume.

The quiesce process provides a transactionally consistent image at the instant the snap is made. Because the snap is consistent, it is not necessary to check the consistency of the file system or database when you activate the snap for access.

After the snap, the snapshot function continues to track the transaction activity in the original pool. It determines which blocks in the original pool will change as data writes are made to the original pool. It suspends the write activity while it copies the original block data to the designated pool where it stores the pool snapshot, referred to as the “stored-on pool.” Then, it allows a write to that block in the original pool. This keeps the snapshot metadata consistent in time with the exact instant the snapshot was taken.

As the original pool changes, the snapshot can grow to the size of the stored-on pool. The average disk space requirements for a pool snapshot are 10 percent to 20 percent of the original pool size. The actual space depends on how many writes are made to the original volume and the time period that the snap exists before it is archived or deleted.

Creating Pool Snapshots

As you create snapshots for a data pool, specify a unique snap name for each snap. The snap name is generally a modified version of the original pool’s name that adds “_Sn” (snap plus an incremental number of snaps taken for this pool).

For example, use snap names PoolA_S1, PoolA_S2, and so on for snaps of PoolA. This convention is particularly important if you plan to create a series of snapshots for multiple pools.

A snapshot’s snap name will also serve as its pool name when it is active. A snap’s time stamp marks the instant when the snap occurred. The combination of snap name and time stamp will help you identify the version of a data pool you want to access when a pool snapshot is active.

Choosing the Stored-on Pool

You can store pool snapshots in any data pool with sufficient space on the server. However, all snapshots for any given original pool must reside on the same stored-on pool. Currently, a combination of up to 500 snapshots can exist on any given stored-on pool.

When you create a pool’s first snapshot, you will select the data pool where you want to store its snapshots. You can select the pool itself or a different pool to be the stored-on pool. In general, you can achieve better performance by selecting a data pool located on a different disk than the pool you want to snap.

Activating Pool Snapshots

You activate a pool snapshot whenever you want to access the data on it, such as for data retrieval, data modification, and data backup. Once the pool snapshot is active, it appears by its user-assigned snap name in the pool list. Treat it as you would any pool to manage the pool and to activate and mount its volumes.

The names of volumes on the pool snapshot are a modified version of the volumes on the original pool. Generally, an "_SV" (snapshot volume) is added to the volumes' names. When you deactivate the pool snapshot, any snapshot volumes on it are automatically deactivated, and its snap name will no longer be listed in the pool list.

Archiving and Deleting Pool Snapshots

You delete a pool snapshot when you no longer need it, or if you want to make room to add other snapshots to its stored-on pool.

> **NOTE:** For the Beta release of Nakoma, only the oldest snapshot on a pool can be deleted. This means that for all snapshots of any given pool, the snapshots must be deleted in the order they were created, according to their time stamps. This will change for FCS.

Remember that end users can access the snap for reads and writes, just as they would any pool. Because any given pool snapshot may rotate into an archived status or be deleted, you need to be aware of how end users are accessing and modifying data within that snap.

Using Pool Snapshot Commands

The following table shows the commands you can use to manage pool snapshots.

Task	Command Line
Creating a snapshot	`mm snap create <snappool> <datapool> <snapname>`
Deleting a snapshot	`mm snap delete <snapname>`
Listing snapshots	`mm snap list`
Activating a snapshot	`mm snap activate <snapname>`
Deactivating a snapshot	`mm snap deactivate <snapname>`
Renaming a snapshot	Currently under development: There is no interface at this time for renaming a pool snapshot with a different snap name. Please do not use other pool utilities to rename a pool snapshot at this time.

The following table shows the attribute descriptions for pool snapshot commands listed in the table above.

Attribute	Attribute Description
snappool	the name of the pool that you are snapping
datapool	the name of the stored-on pool for the pool snapshot
snapname	the snap name of the pool snapshot, which will be used as its pool name when the pool snapshot is activated