What database restore is

SnapManager Oracle

NetApp
April 15, 2021

Table of Contents

What database restore is

SnapManager enables you to perform file-based backup and restore operations.

The following table describes the restore methods:

Restore process	Details
File-based restores	Storage-side full file system restore (from primary or secondary): SnapManager performs a full logical unit number (LUN) restore.

Storage-side full file system restore

A storage-side full file system restore is performed when a volume restore cannot be performed, but the entire files system can be restored on the storage system.

When a storage-side file system restore is performed, the following occurs:

- In a SAN environment, all the LUNs used by the file system (and underlying volume group if any) are restored on the storage system.

When a storage-side file system restore is performed, the following occurs, depending on the storage location:

- When SnapManager restores from primary storage systems, the LUNs (SAN) are restored in place via SFSR.
- When SnapManager restores from secondary storage systems, the LUNs (SAN) are copied from secondary storage systems back to the primary storage system over the network.

Because the file system is fully restored, files that are not part of the backup are reverted as well. An override is required if files, which are not part of the restore, exist in the file system that is being restored.

Host-side file restore

A host-side file copy restore is used as a last resort in SAN environments when storage-side file system restore and storage-side file restore cannot be performed.

A host-side file copy restore involves the following tasks:

- Cloning the storage
- Connecting the cloned storage to the host
- Copying files out of the clone file systems back into the active file systems
- Disconnecting the clone storage from the host
- Deleting the clone storage

Backup recovery

In SnapManager, you must perform the restore and recover operations at the same time.

You cannot perform a restore operation and then perform a SnapManager recover operation later.

In SnapManager 3.2 or earlier, you can either use SnapManager to restore and recover the backup or use SnapManager to restore the backup and use another tool, such as Oracle Recovery Manager (RMAN), to recover the data. Because SnapManager can register its backups with RMAN, you can use RMAN to restore and recover the database at finer granularities such as blocks. This integration combines the benefits of speed and space efficiency of Snapshot copies with the fine level of control for restoring using RMAN.

You must recover a database before you can use it. You can use any tool or script to recover a database.

Starting from SnapManager 3.2 for Oracle, SnapManager enables the restore of database backups automatically by using the archive log backups. Even when the archive log backups are available in the external location, SnapManager uses the archive log backups from the external location to restore the database backups.

If new data files are added to the database, Oracle recommends that you take a new backup immediately. Also, if you restore a backup taken before the new data files were added and attempt to recover to a point after the new data files were added, the automatic Oracle recovery process might fail, because it is unable to create data files. See the Oracle documentation for the process for recovering data files added after a backup.

Database state needed for the restore process

The state of the database that is to be restored depends on the type of restore process that you want to perform and the type of files that are to be included.

The following table lists the state in which the database should be depending on the restore option selected and the type of files you want to include in the restore:

Type of restore	Files included	Database state for this instance
Restore only	Control files	Shutdown
System files	Mount or Shutdown	No system files
Any state	Restore and recovery	Control files
Shutdown	System files	Mount

The database state required by SnapManager for a restore operation depends on the type of restore being performed (complete, partial, or control files). SnapManager does not transition the database to a lower state (for example, from Open to Mount) unless the force option is specified.

What restore preview plans are

SnapManager provides restore plans before and after a restore operation is completed. The restore plans are used to preview, review, and analyze regarding different restore methods.

Structure of the restore plan

The restore plan consists of the following two sections:

- Preview/Review: This section describes how SnapManager will restore (or has restored) each file.
- Analysis: This section describes why some restore mechanisms were not used during the restore operation.

The Preview/Review section

This section shows how each file will be or has been restored. When you view the restore plan before a restore operation, it is called a preview. When you view it after a restore operation is completed, it is called a review.

The following preview example shows that the files are restored by using storage-side file system restore and storage-side system restore methods. To determine why all the files would not be restored by using the same restore method, see the Analysis section.

```
Preview:
The following files will be restored completely via: storage side full
file system restore
E:\rac6\sysaux.dbf
E:\rac6\system.dbf
```

Each restore method has one subsection that contains information about the files that can be restored using that restore method. The subsections are ordered according to decreasing levels of storage method efficiency.

It is possible for one file to be restored by multiple restore methods. Multiple restore methods are used when the underlying logical unit numbers (LUNs) used for a file system are spread among different storage system volumes and some volumes are eligible for volume restore, while others are not. If multiple restore methods are used to restore the same file, the preview section will be similar to the following:

```
The following files will be restored via a combination of:
[storage side file system restore and storage side system restore]
```

The Analysis section

The Analysis section presents the reasons why some restore mechanisms will not be or were not used. You can use this information to determine what is required to enable more efficient restore mechanisms.

The following example shows an Analysis section:

```
Analysis:

The following reasons prevent certain files from being
restored completely via: storage side full file system restore
   * LUNs present in snapshot of volume fas960:
        \vol\disks may not be consistent when reverted:
        [fas960:\vol\disks\DG4D1.lun]
     Mapped LUNs in volume fas960:\vol\disks
        not part of the restore scope will be reverted: [DG4D1.lun]

Files to restore:
     E:\disks\sysaux.dbf
     E:\disks\system.dbf
     E:\disks\undotbs1.dbf
     E:\disks\undotbs2.dbf

* Reasons denoted with an asterisk (*) are overridable.
```

In the example, you can override the first failure either from the command-line interface (CLI), or by selecting **Override** in the graphical user interface (GUI). The second failure about mapped LUNs in the volume is mandatory and not overridable.

You can resolve checks by doing the following:

- To resolve a mandatory check failure, change the environment so that the check will pass.
- To resolve an overridable check failure, you can change the environment, or override the check.

  However, you must be careful because overriding the check can result in undesired consequences.