Backup and recovery basics

Enterprise applications

NetApp
April 25, 2024

Table of Contents

Backup and recovery basics

Oracle databases and snapshot-based backups

The foundation of Oracle database data protection on ONTAP is NetApp Snapshot technology.

The key values are as follows:

- **Simplicity.** A snapshot is a read-only copy of the contents of a container of data at a specific point in time.
- **Efficiency.** Snapshots require no space at the moment of creation. Space is only consumed when data is changed.
- **Manageability.** A backup strategy based on snapshots is easy to configure and manage because snapshots are a native part of the storage OS. If the storage system is powered on, it is ready to create backups.
- **Scalability.** Up to 1024 backups of a single container of files and LUNs can be preserved. For complex datasets, multiple containers of data can be protected by a single, consistent set of snapshots.
- Performance is unaffected, whether a volume contains 1024 snapshots or none.

Although many storage vendors offer snapshot technology, the Snapshot technology within ONTAP is unique and offers significant benefits to enterprise application and database environments:

- Snapshot copies are part of the underlying Write-Anywhere File Layout (WAFL). They are not an add-on or external technology. This simplifies management because the storage system is the backup system.
- Snapshot copies do not affect performance, except for some edge cases such as when so much data is stored in snapshots that the underlying storage system fills up.
- The term "consistency group" is often used to refer to a grouping of storage objects that are managed as a consistent collection of data. A snapshot of a particular ONTAP volume constitutes consistency group backup.

ONTAP snapshots also scale better than competing technology. Customers can store 5, 50, or 500 snapshots without affecting performance. The maximum number of snapshots currently allowed in a volume is 1024. If additional snapshot retention is required, there are options to cascade the snapshots to additional volumes.

As a result, protecting a dataset hosted on ONTAP is simple and highly scalable. Backups do not require movement of data, therefore a backup strategy can be tailored to the needs of the business rather than the limitations of network transfer rates, large number of tape drives, or disk staging areas.

Is a snapshot a backup?

One commonly asked question about the use of snapshots as a data protection strategy is the fact that the "real" data and the snapshot data are located on the same drives. Loss of those drives would result in the loss of both the primary data and the backup.

This is a valid concern. Local snapshots are used for day-to-day backup and recovery needs, and in that respect the snapshot is a backup. Close to 99% of all recovery scenarios in NetApp environments rely on snapshots to meet even the most aggressive RTO requirements.

Local snapshots should, however, never be the only backup strategy, which is why NetApp offers technology such as SnapMirror and SnapVault replication to quickly and efficiently replicate snapshots to an independent set of drives. In a properly architected solution with snapshots plus snapshot replication, the use of tape can be minimized to perhaps a quarterly archive or eliminated entirely.

Snapshot-based backups

There are many options for using ONTAP Snapshot copies to protect your data, and snapshots are the basis for many other ONTAP features, including replication, disaster recovery, and cloning. A complete description of snapshot technology is beyond the scope of this document, but the following sections provide a general overview.

There are two primary approaches to creating a snapshot of a dataset:

- Crash-consistent backups
- Application-consistent backups

A crash-consistent backup of a dataset refers to the capture of the entire dataset structure at a single point in time. If the dataset is stored in a single NetApp FlexVol volume, then the process is simple; a Snapshot can be created at any time. If a dataset spans volumes, a consistency group (CG) snapshot must be created. Several options exist for creating CG snapshots, including NetApp SnapCenter software, native ONTAP consistency group features, and user-maintained scripts.

Crash-consistent backups are primarily used when point-of-the-backup recovery is sufficient. When more granular recover is required, application-consistent backups are usually required.

The word "consistent" in "application-consistent" is often a misnomer. For example, placing an Oracle database in backup mode is referred to as an application-consistent backup, but the data is not made consistent or quiesced in any way. The data continue to change throughout the backup. In contrast, most MySQL and Microsoft SQL Server backups do indeed quiesce the data before executing the backup. VMware may or may not make certain files consistent.

Consistency groups

The term "consistency group" refers to the ability of a storage array to manage multiple storage resources as a single image. For example, a database might consist of 10 LUNs. The array must be able to back up, restore, and replicate those 10 LUNs in a consistent manner. Restoration is not possible if the images of the LUNs were not consistent at the point of backup. Replicating those 10 LUNs requires that all the replicas are perfectly synchronized with each other.

The term "consistency group" is not often used when discussing ONTAP because consistency has always been a basic function of the volume and aggregate architecture within ONTAP. Many other storage arrays manage LUNs or file systems as individual units. They could then be optionally configured as a "consistency group" for purposes of data protection, but this is an extra step in the configuration.

ONTAP has always been able to capture consistent local and replicated images of data. Although the various volumes on an ONTAP system are not usually formally described as a consistency group, that is what they are. A snapshot of that volume is a consistency group image, restoration for that snapshot is a consistency group restoration, and both SnapMirror and SnapVault offer consistency group replication.

Consistency group snapshots

Consistency group snapshots (cg-snapshots) are an extension of the basic ONTAP Snapshot technology. A standard snapshot operation creates a consistent image of all data within a single volume, but sometimes it is necessary to create a consistent set of snapshots across multiple volumes and even across multiple storage systems. The result is a set of snapshots that can be used in the same way as a snapshot of just one individual volume. They can be used for local data recovery, replicated for disaster recovery purposes, or cloned as a single consistent unit.

The largest known use of cg-snapshots is for a database environment of approximately 1PB in size spanning 12 controllers. The cg-snapshots created on this system have been used for backup, recovery and cloning.

Most of the time, when a data set spans volumes and write order must be preserved, a cg-snapshot is automatically used by the chosen management software. There is no need to understand the technical details of cg-snapshots in such cases. However, there are situations in which complicated data protection requirements require detailed control over the data protection and replication process. Automation workflows or the use of custom scripts to call the cg-snapshot APIs are some of options. Understanding the best option and the role of cg-snapshot requires a more detailed explanation of the technology.

Creation of a set of cg-snapshots is a two-step process:

1. Establish write fencing on all target volumes.
2. Create snapshots of those volumes while in the fenced state.

Write fencing is established serially. This means that as the fencing process is set up across multiple volumes, write I/O is frozen on the first volume in the sequence as it continues to be committed to volumes that appear later. This might initially appear to violate the requirement for write order to be preserved, but that only applies to I/O that is issued asynchronously on the host and does not depend on any other writes.

For example, a database might issue a lot of asynchronous datafile updates and allow the OS to reorder the I/O and complete them according to its own scheduler configuration. The order of this type of I/O cannot be guaranteed because the application and operating system have already released the requirement to preserve write order.

As a counter example, most database logging activity is synchronous. The database does not proceed with further log writes until the I/O is acknowledged, and the order of those writes must be preserved. If a log I/O arrives on a fenced volume, it is not acknowledged and the application blocks on further writes. Likewise, file system metadata I/O is usually synchronous. For example, a file deletion operation must not be lost. If an operating system with an xfs file system deleted a file and the I/O that updated the xfs file system metadata to remove the reference to that file landed on a fenced volume, then the file system activity would pause. This guarantees the integrity of the file system during cg-snapshot operations.

After write fencing is set up across the target volumes, they are ready for snapshot creation. The snapshots need not be created at precisely the same time because the state of the volumes is frozen from a dependent write point of view. To guard against a flaw in the application creating the cg-snapshots, the initial write fencing includes a configurable timeout in which ONTAP automatically releases the fencing and resumes write processing after a defined number of seconds. If all the snapshots are created before the timeout period lapses, then the resulting set of snapshots are a valid consistency group.

Dependent write order

From a technical point of view, the key to a consistency group is preserving write order and, specifically, dependent write order. For example, a database writing to 10 LUNs writes simultaneously to all of them. Many writes are issued asynchronously, meaning that the order in which they are completed is unimportant and the actual order they are completed varies based on operating system and network behavior.

Some write operations must be present on disk before the database can proceed with additional writes. These critical write operations are called dependent writes. Subsequent write I/O depends on the presence of these writes on disk. Any snapshot, recovery, or replication of these 10 LUNs must make sure that dependent write order is guaranteed. File system updates are another example of write-order dependent writes. The order in which file system changes are made must be preserved or the entire file system could become corrupt.

Strategies

There are two primary approaches to snapshot-based backups:

- Crash-consistent backups
- Snapshot-protected hot backups

A crash-consistent backup of a database refers to the capture of the entire database structure, including datafiles, redo logs, and control files, at a single point in time. If the database is stored in a single NetApp FlexVol volume, then the process is simple; a Snapshot can be created at any time. If a database spans volumes, a consistency group (CG) snapshot must be created. Several options exist for creating CG snapshots, including NetApp SnapCenter software, native ONTAP consistency group features, and user-maintained scripts.

Crash-consistent Snapshot backups are primarily used when point-of-the-backup recovery is sufficient. Archive logs can be applied under some circumstances, but when more granular point-in-time recovery is required, a online backup is preferable.

The basic procedure for a snapshot-based online backup is as follows:

1. Place the database in `backup` mode.
2. Create a snapshot of all volumes hosting datafiles.
3. Exit `backup` mode.
4. Run the command `alter system archive log current` to force log archiving.
5. Create snapshots of all volumes hosting the archive logs.

This procedure yields a set of snapshots containing datafiles in backup mode and the critical archive logs generated while in backup mode. These are the two requirements for recovering a database. Files such as control files should also be protected for convenience, but the only absolute requirement is protection for datafiles and archive logs.

Although different customers might have very different strategies, almost all of these strategies are ultimately based on the the same principles outlined below.

Snapshot-based recovery

When designing volume layouts for Oracle databases, the first decision is whether to use volume-based NetApp SnapRestore (VBSR) technology.

Volume-based SnapRestore allows a volume to be almost instantly reverted to an earlier point in time. Because all of the data on the volume is reverted, VBSR might not be appropriate for all use cases. For example, if an entire database, including datafiles, redo logs, and archive logs, is stored on a single volume and this volume is restored with VBSR, then data is lost because the newer archive log and redo data are discarded.

VBSR is not required for restore. Many databases can be restored by using file-based single-file SnapRestore (SFSR) or by simply copying files from the snapshot back into the active file system.

VBSR is preferred when a database is very large or when it must be recovered as quickly as possible, and the use of VBSR requires isolation of the datafiles. In an NFS environment, the datafiles of a given database must be stored in dedicated volumes that are uncontaminated by any other type of file. In a SAN environment, datafiles must be stored in dedicated LUNs on dedicated FlexVol volumes. If a volume manager is used (including Oracle Automatic Storage Management [ASM]), the diskgroup must also be dedicated to datafiles.

Isolating datafiles in this manner allows them to be reverted to an earlier state without damaging other file systems.

Snapshot reserve

For each volume with Oracle data in a SAN environment, the `percent-snapshot-space` should be set to zero because reserving space for a snapshot in a LUN environment is not useful. If the fractional reserve is set to 100, a snapshot of a volume with LUNs requires enough free space in the volume, excluding the snapshot reserve, to absorb 100% turnover of all of the data. If the fractional reserve is set to a lower value, then a correspondingly smaller amount of free space is required, but it always excludes the snapshot reserve. This means that the snapshot reserve space in a LUN environment is wasted.

In an NFS environment, there are two options:

- Set the `percent-snapshot-space` based on expected snapshot space consumption.
- Set the `percent-snapshot-space` to zero and manage active and snapshot space consumption collectively.

With the first option, `percent-snapshot-space` is set to a nonzero value, typically around 20%. This space is then hidden from the user. This value does not, however, create a limit on utilization. If a database with a 20% reservation experiences 30% turnover, the snapshot space can grow beyond the bounds of the 20% reserve and occupy unreserved space.

The main benefit of setting a reserve to a value such as 20% is to verify that some space is always available for snapshots. For example, a 1TB volume with a 20% reserve would only permit a database administrator (DBA) to store 800GB of data. This configuration guarantees at least 200GB of space for snapshot consumption.

When `percent-snapshot-space` is set to zero, all space in the volume is available to the end user, which delivers better visibility. A DBA must understand that, if he or she sees a 1TB volume that leverages snapshots, this 1TB of space is shared between active data and Snapshot turnover.

There is no clear preference between option one and option two among end users.

ONTAP and third-party snapshots

Oracle Doc ID 604683.1 explains the requirements for third-party snapshot support and the multiple options available for backup and restore operations.

The third-party vendor must guarantee that the company’s snapshots conform to the following requirements:

- Snapshots must integrate with Oracle’s recommended restore and recovery operations.
- Snapshots must be database crash consistent at the point of the snapshot.
- Write ordering is preserved for each file within a snapshot.

ONTAP and NetApp Oracle management products comply with these requirements.

Rapid Oracle database recovery with SnapRestore

Rapid data restoration in ONTAP from a snapshot is delivered by NetApp SnapRestore technology.

When a critical dataset is unavailable, critical business operations are down. Tapes can break, and even restores from disk-based backups can be slow to transfer across the network. SnapRestore avoids these problems by delivering near instantaneous restoration of datasets. Even petabyte-scale databases can be completely restored with just a few minutes of effort.

There are two forms of SnapRestore - file/LUN-based and volume-based.

- Individual files or LUNs can be restored in seconds, whether it is a 2TB LUN or a 4KB file.
- The container of files or LUNs can be restored in seconds, whether it is 10GB or 100TB of data.

A "container of files or LUNs" would typically refer to a FlexVol volume. For example, you may have 10 LUNs that make up a LVM diskgroup in a single volume, or a volume might store the NFS home directories of 1000 users. Rather than executing a restore operation for each individual file or LUN, you can restore the entire volume as a single operation. This process also work with scale-out containers that include multiple volumes, such as a FlexGroup or an ONTAP Consistency Group.

The reason SnapRestore works so quickly and efficiently is due to the nature of a snapshot, which is essentially a parallel read-only view of the contents of a volume at a specific point in time. The active blocks are the real blocks that can be changed, while the snapshot is a read-only view into the state of the blocks that constitute the files and LUNs at the time the snapshot was created.

ONTAP only permits read-only access to snapshot data, but the data can be reactivated with SnapRestore. The snapshot is reenabled as a read-write view of the data, returning the data to its prior state. SnapRestore can operate at the volume or the file level. The technology is essentially the same with a few minor differences in behavior.

Volume SnapRestore

Volume-based SnapRestore returns the entire volume of data to an earlier state. This operation does not require data movement, meaning that the restore process is essentially instantaneous, although the API or CLI operation might take a few seconds to be processed. Restoring 1GB of data is no more complicated or time-consuming than restoring 1PB of data. This capability is the primary reason many enterprise customers migrate to ONTAP storage systems. It delivers an RTO measured in seconds for even the largest datasets.

One drawback to volume-based SnapRestore is caused by the fact that changes within a volume are cumulative over time. Therefore, each snapshot and the active file data are dependent on the changes leading up to that point. Reverting a volume to an earlier state means discarding all the subsequent changes that had been made to the data. What is less obvious, however, is that this includes subsequently created snapshots. This is not always desirable.

For example, a data retention SLA might specify 30 days of nightly backups. Restoring a dataset to a snapshot created five days ago with volume SnapRestore would discard all the snapshots created on the previous five days, violating the SLA.

There are a number of options available to address this limitation:

1. Data can be copied from a prior snapshot, as opposed to performing a SnapRestore of the entire volume. This method works best with smaller datasets.
2. A snapshot can be cloned rather than restored. The limitation to this approach is that the source snapshot is a dependency of the clone. Therefore, it cannot be deleted unless the clone is also deleted or is split into an independent volume.

3. Use of file-based SnapRestore.

File SnapRestore

File-based SnapRestore is a more granular snapshot-based restoration process. Rather than reverting the state of an entire volume, the state of an individual file or LUN is reverted. No snapshots need to be deleted, nor does this operation create any dependency on a prior snapshot. The file or LUN becomes immediately available in the active volume.

No data movement is required during a SnapRestore restore of a file or LUN. However, some internal metadata updates are required to reflect the fact that the underlying blocks in a file or LUN now exist in both a snapshot and the active volume. There should be no effect on performance, but this process blocks the creation of snapshots until it is complete. The processing rate is approximately 5GBps (18TB/hour) based on the total size of the files restored.

Oracle database online backups

Two sets of data are required to protect and recover an Oracle database in backup mode. Note that this is not the only Oracle backup option, but it is the most common.

- A snapshot of the datafiles in backup mode
- The archive logs created while the datafiles were in backup mode

If complete recovery including all committed transactions is required, a third item is required:

- A set of current redo logs

There are a number of ways to drive recovery of an online backup. Many customers restore snapshots by using the ONTAP CLI and then using Oracle RMAN or sqlplus to complete the recovery. This is especially common with large production environments in which the probability and frequency of database restores is extremely low and any restore procedure is handled by a skilled DBA. For complete automation, solutions such as NetApp SnapCenter include an Oracle plug-in with both command- line and graphical interfaces.

Some large-scale customers have taken a simpler approach by configuring basic scripting on the hosts to place the databases in backup mode at a specific time in preparation for a scheduled snapshot. For example, schedule the command `alter database begin backup` at 23:58, `alter database end backup` at 00:02, and then schedule snapshots directly on the storage system at midnight. The result is a simple, highly scalable backup strategy that requires no external software or licenses.

Data layout

The simplest layout is to isolate datafiles into one or more dedicated volumes. They must be uncontaminated by any other file type. This is to make sure that the datafile volumes can be rapidly restored through a SnapRestore operation without destroying an important redo log, controlfile, or archive log.

SAN has similar requirements for datafile isolation within dedicated volumes. With an operating system such as Microsoft Windows, a single volume might contain multiple datafile LUNs, each with an NTFS file system. With other operating systems, there is generally a logical volume manager. For example, with Oracle ASM, the simplest option would be to confine the LUNs of an ASM disk group to a single volume that can be backed up and restored as a unit. If additional volumes are required for performance or capacity management reasons, creating an additional disk group on the new volume results in simpler management.

If these guidelines are followed, snapshots can be scheduled directly on the storage system with no requirement for performing a consistency group snapshot. The reason is that Oracle backups do not require datafiles to be backed up at the same time. The online backup procedure was designed to allow datafiles to continue to be updated as they are slowly streamed to tape over the course of hours.

A complication arises in situations such as the use of an ASM disk group that is distributed across volumes. In these cases, a cg-snapshot must be performed to make sure that the ASM metadata is consistent across all constituent volumes.

Caution: Verify that the ASM `spfile` and `passwd` files are not in the disk group hosting the datafiles. This interferes with the ability to selectively restore datafiles and only datafiles.

Local recovery procedure—NFS

This procedure can be driven manually or through an application such as SnapCenter. The basic procedure is as follows:

1. Shut down the database.
2. Recover the datafile volume(s) to the snapshot immediately prior to the desired restore point.
3. Replay archive logs to the desired point.
4. Replay current redo logs if complete recovery is desired.

This procedure assumes that the desired archive logs are still present in the active file system. If they are not, the archive logs must be restored or rman/sqlplus can be directed to the data in the snapshot directory.

In addition, for smaller databases, datafiles can be recovered by an end user directly from the `.snapshot` directory without assistance from automation tools or storage administrators to execute a `snaprestore` command.

Local recovery procedure—SAN

This procedure can be driven manually or through an application such as SnapCenter. The basic procedure is as follows:

1. Shut down the database.
2. Quiesce the disk group(s) hosting the datafiles. The procedure varies depending on the logical volume manager chosen. With ASM, the process requires dismounting the disk group. With Linux, the file systems must be dismounted, and the logical volumes and volume groups must be deactivated. The objective is to stop all updates on the target volume group to be restored.
3. Restore the datafile disk groups to the snapshot immediately prior to the desired restore point.
4. Reactivate the newly restored disk groups.
5. Replay archive logs to the desired point.
6. Replay all redo logs if complete recovery is desired.

This procedure assumes that the desired archive logs are still present in the active file system. If they are not, the archive logs must be restored by taking the archive log LUNs offline and performing a restore. This is also an example in which dividing up archive logs into dedicated volumes is useful. If the archive logs share a volume group with redo logs, then the redo logs must be copied elsewhere before restoration of the overall set of LUNs. This step prevents the loss of those final recorded transactions.

Oracle Database Storage Snapshot Optimized backups

Snapshot-based backup and recovery became even simpler back when Oracle 12c was released because there is no need to place a database in hot backup mode. The result is an ability to schedule snapshot-based backups directly on a storage system and still preserve the ability to perform complete or point-in-time recovery.

Although the hot backup recovery procedure is more familiar to DBAs, it has, for a long time, been possible to use snapshots that were not created while the database was in hot backup mode. Extra manual steps were required with Oracle 10g and 11g during recovery to make the database consistent. With Oracle 12c, `sqlplus` and `rman` contain the extra logic to replay archive logs on datafile backups that were not in hot backup mode.

As discussed previously, recovering a snapshot-based hot backup requires two sets of data:

- A snapshot of the datafiles created while in backup mode
- The archive logs generated while the datafiles were in hot backup mode

During recovery, the database reads metadata from the datafiles to select the required archive logs for recovery.

Storage snapshot-optimized recovery requires slightly different datasets to accomplish the same results:

- A snapshot of the datafiles, plus a method to identify the time the snapshot was created
- Archive logs from the time of the most recent datafile checkpoint through the exact time of the snapshot

During recovery, the database reads metadata from the datafiles to identify the earliest archive log required. Full or point-in-time recovery can be performed. When performing a point-in-time recovery, it is critical to know the time of the snapshot of the datafiles. The specified recovery point must be after the creation time of the snapshots. NetApp recommends adding at least a few minutes to the snapshot time to account for clock variation.

For complete details, see Oracle's documentation on the topic, "Recovery Using Storage Snapshot Optimization" available in various releases of the Oracle 12c documentation. Also, see Oracle Document ID Doc ID 604683.1 regarding Oracle third-party snapshot support.

Data layout

The simplest layout is to isolate the datafiles into one or more dedicated volumes. They must be uncontaminated by any other file type. This is to make sure that the datafile volumes can be rapidly restored with a SnapRestore operation without destroying an important redo log, controlfile, or archive log.

SAN has similar requirements for datafile isolation within dedicated volumes. With an operating system such as Microsoft Windows, a single volume might contain multiple datafile LUNs, each with an NTFS file system. With other operating systems, there is generally a logical volume manager as well. For example, with Oracle ASM, the simplest option would be to confine disk groups to a single volume that can be backed up and restored as a unit. If additional volumes are required for performance or capacity management reasons, creating an additional disk group on the new volume results in easier management.

If these guidelines are followed, snapshots can be scheduled directly on ONTAP with no requirement for performing a consistency group snapshot. The reason is that snapshot-optimized backups do not require that datafiles be backed up at the same time.

A complication arises in situations such as an ASM disk group that is distributed across volumes. In these cases, a cg-snapshot must be performed to make sure that the ASM metadata is consistent across all constituent volumes.

[Note]Verify that the ASM spfile and passwd files are not in the disk group hosting the datafiles. This interferes with the ability to selectively restore datafiles and only datafiles.

Local recovery procedure—NFS

This procedure can be driven manually or through an application such as SnapCenter. The basic procedure is as follows:

1. Shut down the database.
2. Recover the datafile volume(s) to the snapshot immediately prior to the desired restore point.
3. Replay archive logs to the desired point.

This procedure assumes that the desired archive logs are still present in the active file system. If they are not, the archive logs must be restored, or `rman` or `sqlplus` can be directed to the data in the `.snapshot` directory.

In addition, for smaller databases, datafiles can be recovered by an end user directly from the `.snapshot` directory without assistance from automation tools or a storage administrator to execute a SnapRestore command.

Local recovery procedure—SAN

This procedure can be driven manually or through an application such as SnapCenter. The basic procedure is as follows:

1. Shut down the database.
2. Quiesce the disk group(s) hosting the datafiles. The procedure varies depending on the logical volume manager chosen. With ASM, the process requires dismounting the disk group. With Linux, the file systems must be dismounted, and the logical volumes and volume groups are deactivated. The objective is to stop all updates on the target volume group to be restored.
3. Restore the datafile disk groups to the snapshot immediately prior to the desired restore point.
4. Reactivate the newly restored disk groups.
5. Replay archive logs to the desired point.

This procedure assumes that the desired archive logs are still present in the active file system. If they are not, the archive logs must be restored by taking the archive log LUNs offline and performing a restore. This is also an example in which dividing up archive logs into dedicated volumes is useful. If the archive logs share a volume group with redo logs, the redo logs must be copied elsewhere before restoration of the overall set of LUNs to avoid losing the final recorded transactions.

Full recovery example

Assume the datafiles have been corrupted or destroyed and full recovery is required. The procedure to do so is as follows:

```
[oracle@host1 ~]$ sqlplus / as sysdba
Connected to an idle instance.
SQL> startup mount;
ORACLE instance started.
Total System Global Area 1610612736 bytes
Fixed Size                  2924928 bytes
Variable Size            1040191104 bytes
Database Buffers          553648128 bytes
Redo Buffers               13848576 bytes
Database mounted.
SQL> recover automatic;
Media recovery complete.
SQL> alter database open;
Database altered.
SQL>
```

Point-in-time recovery example

The entire recovery procedure is a single command: `recover automatic`.

If point-in-time recovery is required, the timestamp of the snapshot(s) must be known and can be identified as follows:

```
Cluster01::> snapshot show -vserver vserver1 -volume NTAP_oradata -fields
create-time
vserver   volume        snapshot   create-time
--------  ------------  ---------  ------------------------
vserver1  NTAP_oradata  my-backup  Thu Mar 09 10:10:06 2017
```

The snapshot creation time is listed as March 9th and 10:10:06. To be safe, one minute is added to the snapshot time:

```
[oracle@host1 ~]$ sqlplus / as sysdba
Connected to an idle instance.
SQL> startup mount;
ORACLE instance started.
Total System Global Area 1610612736 bytes
Fixed Size                  2924928 bytes
Variable Size            1040191104 bytes
Database Buffers          553648128 bytes
Redo Buffers               13848576 bytes
Database mounted.
SQL> recover database until time '09-MAR-2017 10:44:15' snapshot time '09-
MAR-2017 10:11:00';
```

The recovery is now initiated. It specified a snapshot time of 10:11:00, one minute after the recorded time to account for possible clock variance, and a target recovery time of 10:44. Next, sqlplus requests the archive logs required to reach the desired recovery time of 10:44.

```
ORA-00279: change 551760 generated at 03/09/2017 05:06:07 needed for
thread 1
ORA-00289: suggestion : /oralogs_nfs/arch/1_31_930813377.dbf
ORA-00280: change 551760 for thread 1 is in sequence #31
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
ORA-00279: change 552566 generated at 03/09/2017 05:08:09 needed for
thread 1
ORA-00289: suggestion : /oralogs_nfs/arch/1_32_930813377.dbf
ORA-00280: change 552566 for thread 1 is in sequence #32
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
ORA-00279: change 553045 generated at 03/09/2017 05:10:12 needed for
thread 1
ORA-00289: suggestion : /oralogs_nfs/arch/1_33_930813377.dbf
ORA-00280: change 553045 for thread 1 is in sequence #33
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
ORA-00279: change 753229 generated at 03/09/2017 05:15:58 needed for
thread 1
ORA-00289: suggestion : /oralogs_nfs/arch/1_34_930813377.dbf
ORA-00280: change 753229 for thread 1 is in sequence #34
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
Log applied.
Media recovery complete.
SQL> alter database open resetlogs;
Database altered.
SQL>
```

ⓘ Complete recovery of a database using snapshots using the `recover automatic` command does not require specific licensing, but point-in-time recovery using `snapshot time` requires the Oracle Advanced Compression license.

Oracle database management and automation tools

The primary value of ONTAP in an Oracle database environment comes from the core ONTAP technologies such as instant Snapshot copies, simple SnapMirror replication, and efficient creation of FlexClone volumes.

In some cases, simple configuration of these core features directly on ONTAP meets requirements, but more complicated needs require an orchestration layer.

SnapCenter

SnapCenter is the flagship NetApp data protection product. At a very low level, it is similar to the SnapManager products in terms of how it executes database backups, but it was built from the ground up to deliver a single-pane-of-glass for data protection management on NetApp storage systems.

SnapCenter includes the basic functions such as snapshot-based backups and restores, SnapMirror and SnapVault replication, and other features required to operate at scale for large enterprises. These advanced features include an expanded role-based access control (RBAC) capability, RESTful APIs to integrate with third-party orchestration products, nondisruptive central management of SnapCenter plug-ins on database hosts, and a user interface designed for cloud-scale environments.

REST

ONTAP also contains a rich RESTful API set. This allows 3rd party vendors to create data protection and other management application with deep integration with ONTAP. Furthermore, the RESTful API is easy to consume by customers who wish to create their own automation workflows and utilities.

Copyright information

Trademark information