Summary

The implementation of a backup strategy based on RMAN 10g is described in this document. This strategy allows for (1) backups to tape and (2) backups to disk (also called flash backups). Flash backups help to substantially reduce the recovery time for many common recovery scenarios. This new RMAN scripting implementation is built as an extension of the existing (old) infrastructure. Existing functionalities are kept and new ones are introduced.

Architecture

As for the old system, backups are scheduled and 'pushed' from the pdb-backup server. The set of backup scripts remains the same however the following files have been rewritten in the new version:

```
pdb-backup.sh as pdb-backup-10g.sh
pdb-trigger-dbbackup.sh as pdb-trigger-dbbackup-10g.sh
ora.env as ora-10g.env
```

Dependencies:

The pdb-trigger-dbbackup-10g.sh script should be run to set up the type of backup which is expected to run or to define the default tag for backups as copy to disk.

The pdb-backup-10g.sh script is responsible for the backups themselves. It runs in a loop and every minute checks markers which could have been set up by the pdb-trigger-dbbackup-10g.sh script. According to them it performs appropriate backup operations.

The ora-10g.env script is called by the pdb-backup-10g.sh to set up the backup environment.

Functionality

`pdb-trigger-dbbackup-10g.sh`

The aim of this script is to set up the type of backup which is expected to run or the default tag for backups as copy to disk. The type of pending backups is marked as a file of name "BACKUP_suffix" in the etc database's backup subdirectory. In the case of the default tag definition this is stored in RMAN.COPYTODISK.TAG file in the same subdirectory.
Usage:
```
pdb-trigger-dbbackup-10g.sh parameter_1 parameter_2 parameter_3
```
where
`parameter_1` - database name
`parameter_2` - code of backup type
`parameter_3` - the default tag for incremental backup as copy to disk; this parameter is required only for a backup of type 0FB, in other cases is not taken into account

Values allowed for parameter_2:
0T - incremental level 0 backup ("full")
1D - incremental level 1 differential backup
1C - incremental level 1 cumulatvie backup
0FB - to set up the default flag for incremental copy to disk backups (in this case parameter_3 is also required)
1FB - incremental as copy to disk backup
A - archivelogs' backup
DD - delete obsolete backups

Examples:
```
pdb-trigger-dbbackup-10g.sh DBNAME 0T
pdb-trigger-dbbackup-10g.sh DBNAME 0FB DBNAME
pdb-trigger-dbbackup-10g.sh DBNAME 1FB
```


pdb-backup-10g.sh

The aim of this script is to perform appropriate backups actions according to the types' markers set by the pdb-trigger-dbbackup-10g.sh script.
Usage:
```
pdb-backup-10g.sh parameter_1 parameter_2 parameter_3
```
where

`parameter_1` - database name; a node name in case of RACs

`parameter_2` - the number of days after which the rolling forward image copy backups are applied; parameter_2 is also used to calculate the date by which archivelogs are to be deleted

`parameter_3` - an optional parameter used to overwrite the default tag for incremental backups as copy to disk

For all incremental backups and archivelog backups the deletion of archivelogs completed before the day 'SYSDATE - (parameter_2 + 1)' is performed.
In case of copy to disk backups, there is also a copy recovery operation performed (in other words - rolling forward image copy) which recovers until time trunc(SYSDATE-parameter_2, 'DD'). This operation can result in a copy which is older than forseen by the parameter_2 value due to, for example, lack of appropriate image copies. Therefore after each copy to disk backup a checking of recovery is done which in case of copies to disk older than from the day before 'SYSDATE - parameter_2' (truncated to midnight time) sends a WARNING email.

Examples:
```
./pdb-backup-10g.sh DBINST1 3
./pdb-backup-10g.sh DBINST1 3 DBINST1
```

This topic: PSSGroup > BackupsUsingRMAN10g
Topic revision: r7 - 2007-02-16 - LucaCanali