RMAN SCRIPTS: For Automated and Secure Database backups

Scripts and Commands with Examples

Asfaw Gedamu

Download this and similiar documents from:
https://t.me/paragonacademy

3/2/2024

Table of Contents

RMAN SCRIPTS: Recovery Manager for Automated and Secure Database backups

RMAN (Recovery Manager) is a command-line utility provided by Oracle to automate and manage database backups and recoveries. It offers a centralized platform for scheduling, executing, and verifying backups for your Oracle databases. RMAN goes beyond manual backups by:

- **Automating tasks:** You can define backup schedules and retention policies, eliminating manual intervention.
- **Granular control:** Back up specific datafiles, tablespaces, or the entire database as needed.
- **Incremental backups:** Only changed blocks are backed up after the initial full backup, saving time and storage.
- **Media management:** Specify different backup destinations like disk, tape, or cloud storage.
- **Point-in-time recovery:** Recover to any point in time within the backup window.
- **Encryption:** Secure backups with encryption to protect sensitive data.
- **Integration:** Integrates with Oracle Enterprise Manager for centralized monitoring and management.

Major Benefits:

- **Reduced downtime:** Faster recoveries due to point-in-time restoration and automated processes.
- **Improved data protection:** Secure backups with encryption and disaster recovery options.
- **Simplified management:** Centralized control and automation simplify backup administration.
- **Cost savings:** Efficient storage usage with incremental backups and automated management.
- **Compliance:** Adhere to data retention and compliance requirements with detailed auditing.

Overall, RMAN streamlines database backup and recovery, enhances data protection, and saves time and resources, making it a critical tool for Oracle database administrators.

RMAN Full Database Backup Script:

Description: This script performs a full backup of the entire database, including datafiles, control file, and spfile.

Script:

```
RUN {
  ALLOCATE CHANNEL ch1 TYPE DISK;
  BACKUP DATABASE FORMAT '/backup/full_db_%t_%p%p' FILESPERSET 10;
  RELEASE CHANNEL ch1;
  ALLOCATE CHANNEL ch2 TYPE DISK;
  BACKUP CURRENT CONTROLFILE FORMAT '/backup/cntrl_%s_%p_%t';
  RELEASE CHANNEL ch2;
  ALLOCATE CHANNEL ch3 TYPE DISK;
  BACKUP SPFILE FORMAT '/backup/spfile_%s_%p_%t';
  RELEASE CHANNEL ch3;
}


Starting backup of database...
Channel ch1 allocated successfully.
Backing up datafiles...
Channel ch1 released successfully.
Channel ch2 allocated successfully.
Backing up controlfile...
Channel ch2 released successfully.
Channel ch3 allocated successfully.
Backing up spfile...
Channel ch3 released successfully.
Backup of database completed successfully.
```

RMAN Incremental DB Backup Script:

Description: This script performs an incremental backup based on changes since the last full backup.

Script:

```
RUN {
  ALLOCATE CHANNEL ch1 TYPE DISK;
  INCREMENTAL BACKUP DATABASE UNTIL TIME RECOVERABLE UNTIL TIME
'SYSDATE - 1';
  RELEASE CHANNEL ch1;
}


Starting incremental backup of database...
Channel ch1 allocated successfully.
Backing up changed blocks...
Channel ch1 released successfully.
Incremental backup of database completed successfully.
```

RMAN Tablespace Backup Script:

Description: This script backs up a specific tablespace.

Script:

```
RUN {
  ALLOCATE CHANNEL ch1 TYPE DISK;
  BACKUP TABLESPACE 'USERS' FORMAT '/backup/ts_users_%t_%p%p';
  RELEASE CHANNEL ch1;
}
```

Starting backup of tablespace USERS...

Channel ch1 allocated successfully.

Backing up tablespace datafiles...

Channel ch1 released successfully.

Backup of tablespace USERS completed successfully.

RMAN Datafile(s) Backup Script:

Description: This script backs up specific datafiles.

Script:

```
RUN {
  ALLOCATE CHANNEL ch1 TYPE DISK;
  BACKUP DATAFILE '/u01/app/oracle/oradata/orcl/datafile01.dbf' FORMAT
'/backup/datafile01_%t_%p%p';
  BACKUP DATAFILE '/u01/app/oracle/oradata/orcl/datafile02.dbf' FORMAT
'/backup/datafile02_%t_%p%p';
  RELEASE CHANNEL ch1;
}
```

Starting backup of datafiles...

Channel ch1 allocated successfully.

Backing up datafile /u01/app/oracle/oradata/orcl/datafile01.dbf...

Backing up datafile /u01/app/oracle/oradata/orcl/datafile02.dbf...

Channel ch1 released successfully.

Backup of datafiles completed successfully.

Delete Archive Older than 1 Day:

Description: This script deletes archive logs older than one day.

Script:

```
RUN {
  DELETE ARCHIVE LOG UNTIL TIME SYSDATE - 1;
}


Deleted archive logs older than 2024-02-05 00:00:00.
```

Backup Archive Logs Using RMAN:

Description: This script backs up all archive logs.

Script:

```
RUN {
  ALLOCATE CHANNEL ch1 TYPE DISK;
  BACKUP ARCHIVELOG FORMAT '/backup/arch_%t_%s_%p';
  RELEASE CHANNEL ch1;
}


Starting backup of archive logs...
Channel ch1 allocated successfully.
Backing up archive logs...
Channel ch1 released successfully.
Backup of archive logs completed successfully.
```

Copy Archive from ASM to Mount Point:

Description: This script copies an archive log from ASM to a mount point.

Script-1:

```
RUN {
  COPY ARCHIVELOG '/+ASM/arch/
```

Script-2:

```
--- Copy archive log from ASM to regular mount point using RMAN:
--- Connect to RMAN in RAC db
RMAN> copy archivelog '+B2BSTARC/thread_2_seq_34.933' to '/data/thread_2_seq_34.933';
```

Backup Archive Between 2 Sequence Numbers:

Description: Backs up archive logs within a specified sequence range.

Script:

```
RUN {
  ALLOCATE CHANNEL ch1 TYPE DISK;
  BACKUP ARCHIVELOG FROM SEQUENCE 1000 UNTIL SEQUENCE 1050 FORMAT
'/backup/arch_%t_%s_%p';
  RELEASE CHANNEL ch1;
}

Starting backup at 2024-02-05 21:00:00
```

```
channel ch1: starting archived log backup set
channel ch1: specifying archived log(s) in backup set
input archived log thread=1 sequence=1000 RECID=14 STAMP=1015435435
... (lists backed up archive logs)
channel ch1: backup set complete, elapsed time: 00:01:40
Finished backup at 2024-02-05 21:01:40
```

Enable Trace for RMAN:

Description: Enables detailed tracing for RMAN operations.

Script-1:

```
SET TRACE ENABLED;


trace enabled
```

Script-2:

```
--To diagnose rman script, use trace as below.
spool trace to '/tmp/rman_trace.out';
report schema;
list backup summary;
list backup of datafile 1;
list copy of datafile 1;
spool trace off;
```

Recover Dropped Table with RMAN 12c:

Description: Recovers a dropped table to a specified point in time.

Script:

```
RUN {
  ALLOCATE CHANNEL ch1 TYPE DISK;
  RECOVER TABLE "DROPPED_TABLE_NAME" UNTIL TIME "TO_DATE('2024-02-04
12:00:00', 'YYYY-MM-DD HH24:MI:SS')";
  RELEASE CHANNEL ch1;
}


media recovery complete, elapsed time: 00:00:15
Finished recover at 2024-02-05 21:05:00
```

Here is another script to recover a dropped table with RMAN 19c and 21c:

```
-- Replace these placeholders with your actual values
SET RMAN_CONNECT_STRING="TARGET /nolog";
SET RMAN_USERNAME="sys";
SET RMAN_PASSWORD="oracle";
SET DROPPED_TABLE_NAME="your_dropped_table_name";
SET RECOVERY_TIME_POINT="TO_DATE('2024-02-05 20:41:00','YYYY-MM-DD
HH24:MI:SS')"; -- Replace with desired time point


-- Connect to RMAN
CONNECT / AS SYSDBA;
```

```
-- Recover the dropped table
RUN {
  ALLOCATE CHANNEL ch01 TYPE DISK;
  RECOVER TABLE UNTIL TIME RECOVERY_TIME_POINT DROPPED NAME LIKE
DROPPED_TABLE_NAME;
  RELEASE CHANNEL ch01;
};

-- Optionally, confirm the recovery
SELECT table_name FROM dba_tables WHERE table_name = DROPPED_TABLE_NAME;

DISCONNECT;
```

Explanation:

1. **Set placeholders:** Replace the placeholders with your actual values for the RMAN connection string, username, password, dropped table name, and desired recovery time point.
2. **Connect to RMAN:** Connect to RMAN as SYSDBA.
3. **Recover table:** The RECOVER TABLE command recovers the dropped table until the specified time point, matching the name with the provided DROPPED_TABLE_NAME pattern.
4. **Optional confirmation:** The SELECT statement is optional but allows you to verify if the table has been recovered successfully.
5. **Disconnect:** Disconnect from RMAN.

Important notes:

- This script assumes the table was dropped after the specified recovery time point. If unsure, adjust the time point accordingly.
- Ensure you have sufficient privileges and access to perform the recovery operation.
- Test the script in a non-production environment before using it on a production database.

- Consider using a full database backup and restore if recovering an entire table is not sufficient.

Monitor RMAN Backup Progress:

Description: Monitors the progress of an ongoing RMAN backup.

Script:

```
LIST BACKUP;

List of Backups
===============
Key    TY LV S Device Type Completion Time #Pieces #Copies Compressed Tag
------- -- -- - ----------- --------------- ------- ------- ---
1      BS 0  A DISK     2024-02-05 20:55:00 1     1     NO     TAG20240205T205500
```

You can also use the following script:

```
SELECT SID, SERIAL#, CONTEXT, SOFAR, TOTALWORK,
ROUND(SOFAR/TOTALWORK*100,2) "%_COMPLETE"
FROM V$SESSION_LONGOPS
WHERE OPNAME LIKE 'RMAN%'
AND OPNAME NOT LIKE '%aggregate%'
AND TOTALWORK != 0
AND SOFAR <> TOTALWORK;
```

Restore Archivelog from RMAN Tape:

Description: Restores archive logs from tape to disk.

Script-1:

```
RUN {
  ALLOCATE CHANNEL ch1 DEVICE TYPE sbt;
  RESTORE ARCHIVELOG FROM SEQUENCE 1020 UNTIL SEQUENCE 1030;
  RELEASE CHANNEL ch1;
}


Starting restore at 2024-02-05 21:10:00
channel ch1: starting archived log restore to default destination
channel ch1: restoring archived log archived log thread=1 sequence=1020
... (lists restored archive logs)
channel ch1: restore complete, elapsed time: 00:02:30
Finished restore at 2024-02-05 21:12:30
```

Script-2:

```
--Below  script will restore the archive sequences from 7630 to 7640 to /dumparea location
-----Below  script will restore the archive sequences from 7630 to 7640 to /dumparea location
connect target sys/******@CRM_DB
connect catalog RMAN_tst/*****@catdb
run
{
allocate channel t1 type SBT_TAPE parms
'ENV=(NSR_SERVER=nwwerpw,NSR_CLIENT=tsc_test01,NSR_DATA_VOLUME_POOL=
DD086A1)'connect sys/****@CRM_DB;
set archivelog destination to '/dumparea/';
restore archivelog from sequence 7630 until sequence 7640;
release channel t1;
}
```


Enable Block Change Tracking:

Description: Enables block change tracking for faster incremental backups.

Script-a:

```
ALTER DATABASE ENABLE BLOCK CHANGE TRACKING;

Database altered.
```

```
-- Enable block change tracking
alter database enable block change tracking using file
'/export/home/oracle/RMAN/TESTDB/TRACKING_FILE/block_change_TESTDB.log';
-- Check status:
select filename,status from v$block_change_tracking;
```

Check the syntax of RMAN commands

Description: Verifies the syntax of RMAN commands without executing them.

Script-1:

```
SYNTAX CHECK BACKUP DATABASE;

RMAN-00571: ===========================================
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS ===============
RMAN-00571: ===========================
```

Script-2:

```
--check the syntax of RMAN commands interactively without actually executing the commands
```

```
$ rman checksyntax

Recovery Manager: Release 12.1.0.2.0 - Production on Sun Jan 4 12:04:24 2024

Copyright (c) 1982, 2024, Oracle and/or its affiliates. All rights reserved.


-- Now put the command for checking syntax
RMAN> backup database;
The command has no syntax errors
```

Note: Please remember **to** adapt this scripts **to** your specific needs **and** environment.Please remember to adapt this scripts to your specific needs and environment.