Understanding ZDLRA

- Fernando Simon
- Senior DBA Architect – eProseed
- LUXOUG Board Member

Disclaimer

- *“The postings on this document are my own and don’t necessarily represent my actual employer positions, strategies or opinions. The information here was edited to be useful for general purpose, specific data and identifications were removed to allow reach the generic audience and to be useful for the community.”*

Agenda

- About me.
- Backup.
- ZDLRA.
- Enrolling database.
- Replicated Backup and Tape.
- Internal details.
- QA.

450+ Technical Experts Helping Peers Globally

ORACLE
ACE PROGRAM

ORACLE ACE Director

ORACLE ACE

ORACLE ACE Associate

bit.ly/OracleACEProgram

Nominate yourself or someone you know: acenomination.oracle.com

About me

- Senior DBA at eProseed Luxembourg
- OCA, OCP, OCE RAC
- Board Member at LuxOUG

- Contacts:
  - fernando.simon.br@gmail.com
  - https://fernandosimon.com/blog/
  - https://twitter.com/FSimonDBA
  - https://www.linkedin.com/in/fernando-simon/

About me

- DBA since 2004
  - Oracle, PostgreSQL, DB2.

- Oracle Blog since 2010
  - OOW SFO, OOW LA, Brazil User Group Speaker.

- DBA Team Manager at Court of Justice – 2010/2017
  - Exadata since 2010:
    - Exadata V2 (third Exadata in Brazil and first OLTP).
    - Exadata X2 (Half HP), Exadata X4 (Full HP), Exadata X5 (Full EF), Exadata X6 (Quarter).
    - High consolidated environment, IORM, Resource Manager, Instance Caging.

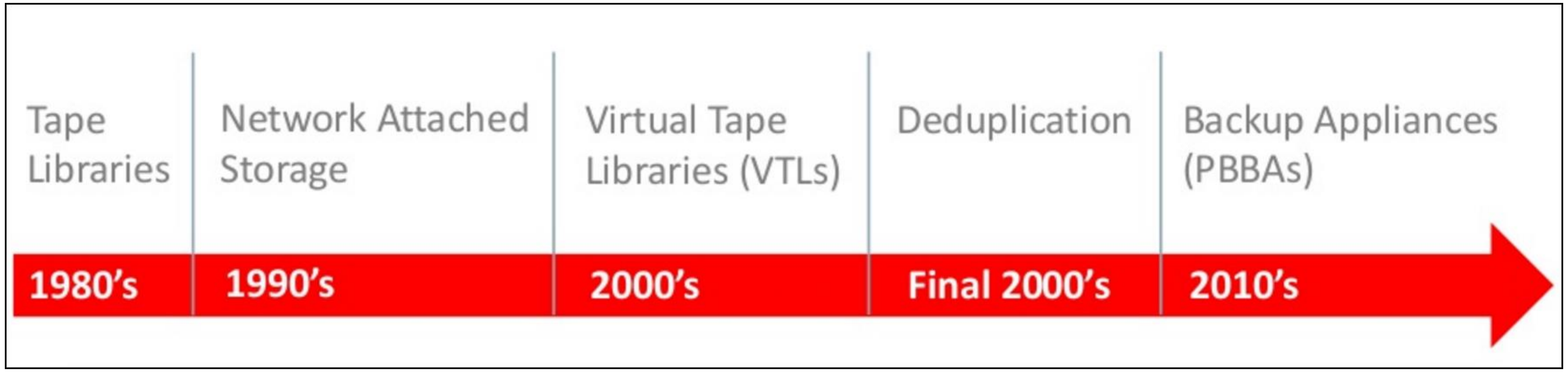
  - ZDLRA since 2014/2015:
    - First ZDLRA in Brazil, one of the first of the word worked replicated.

  - MAA Project, Multi-Site protection, RAC+RAC, DG, ZDLRA
    - OOW SFO 2015 Presentation:
      - https://www.oracle.com/technetwork/database/availability/con8830-zdlradeepdive-2811109.pdf

About me

- Luxembourg 2017
  - eProseed - Senior Database Architect.
- Consulting at European Institution
  - LCM (Life Cycle Management) to the Oracle Products.
  - Supporting the Production Databases.
  - Patch apply for all databases and cluster infrastructure.
- Consulting at Bank Institution
  - Multi site environment.
  - Exadata and ZDLRA support.
- LUXOUG Board Member.

Backup



Tape Libraries	Network Attached Storage	Virtual Tape Libraries (VTLs)	Deduplication	Backup Appliances (PBBAs)
1980's	1990's	2000's	Final 2000's	2010's

- And I will add: Recently, CLOUD

Backup

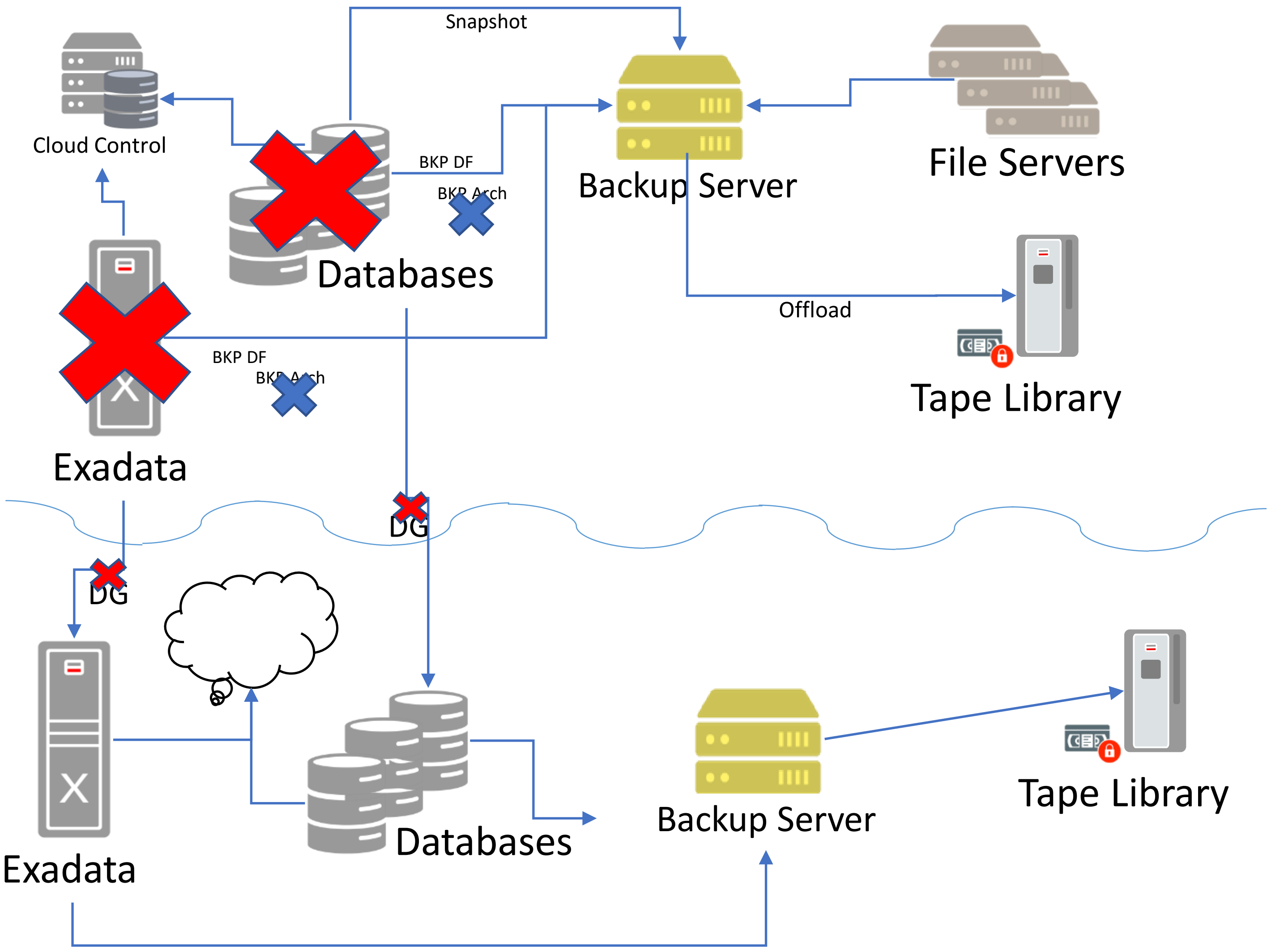
- Principles and goals
  - Restore every information.
  - Low or zero impact over environment.
  - Easy to: operate, control, and verify.
  - Data retention and storage to sustain requirements and regulations.

Backup

- Real Life
  - Data Loss (usually since the last backup).
  - High impact over the environment.
  - A lot of players (Tivoli, EMC, DataProtector, Commvault).
  - Cloud.
  - Validation, test, validation, test, validation, test….

- And it is worst…

Backup



LUXOUG
LUXEMBOURG ORACLE USER GROUP

Backup

- Two words
  - **RPO** – Recovery Point Objective:
    - Usually, what/how much you can loose.
  - **RTO** – Recovery Time Objective:
    - Usually, time to put everything running again.
- The goal is zero RPO and zero RTO.

ZDLRA

ZDLRA

- **ZERO DATA LOSS RECOVERY APPLIANCE – ZDLRA**
  - Engineered Systems.
  - Exadata based.
  - Hardware + Software
    - RA Library.
  - MML for tape
    - SAN.
    - Oracle Secure Backup.
  - Native replication.
  - RMAN Catalog – Integration.
  - EM/CC or CLI.

- **DOES NOT REDUCES RTO, JUST RPO.**

ZDLRA

- Oracle Database
  - Rman catalog
    - Light modified to cover internal RA tables.
  - Store the configurations:
    - Policies, database registrations.

- Delta Store
  - Where the data is stored.
  - **Delta Push** = Virtual Backups + Real Time Redo.
  - Automatic backup index, management, and validation.

- EM/CC/CLI
  - DBMS_RA package to manage everything.

- Backup client library installed in every server that send backup.

ZDLRA

- Virtual Full Backup.
- Real-Time Redo.
- MAA and Replication support.

ZDLRA

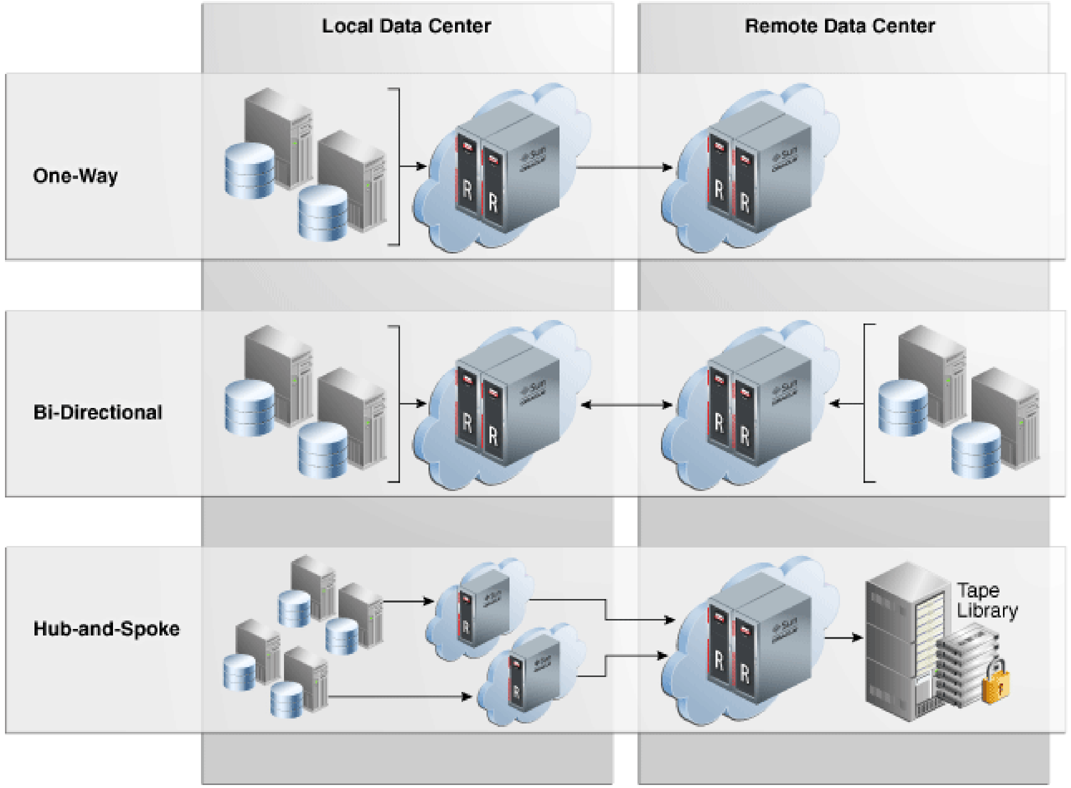
- Virtual Full Backup
  - **Incremental Forever Strategy**:
    - Needs an initial level 0 backup.
    - Consequent level 1 backups.
    - Merge both to generate a virtual full backup for your datafile.
  - Generate index for every datafile.
  - Validated against corruption for backup/datafile.
  - Differs from deduplication:
    - Better because it is based in context and not in black magic (reverse engineering).
  - A little representation:
    - Imagine that you are a Librarian and I deliver to you (every day) one box with books of an encyclopedia.
    - You receive this box but can't open it to store the books that are inside.
      - You can ignore the box and says that you already have this based on the size, weight, whatever you choose to define if you already have this box.
    - As ZDLRA, you receive the same box, but you can open and check if you already have this book or no.
    - One day I come back and ask you the full encyclopedia and you deliver to me the box with the books.
      - **So, what solution do you think that will be better for Oracle backups?**

ZDLRA

- Real-Time Redo:
  - **Is the “zero data loss” - guarantee the zero RPO.**
  - ZDLRA it is a log_archive_dest destination:
    - Can be SYNC or ASYNC.
  - Differ from FARSYNC, just need to config the archive dest at database side.
  - 100% integrated and compatible with MAA.

ZDLRA

- Replication:
  - One-Way:
    - One master and one destination.
  - Bi-Directional:
    - Both sides replicate each other.
  - Hub/Spoke:
    - One to many.
  - **Every ZDLRA can have different policies and recovery windows.**



ZDLRA

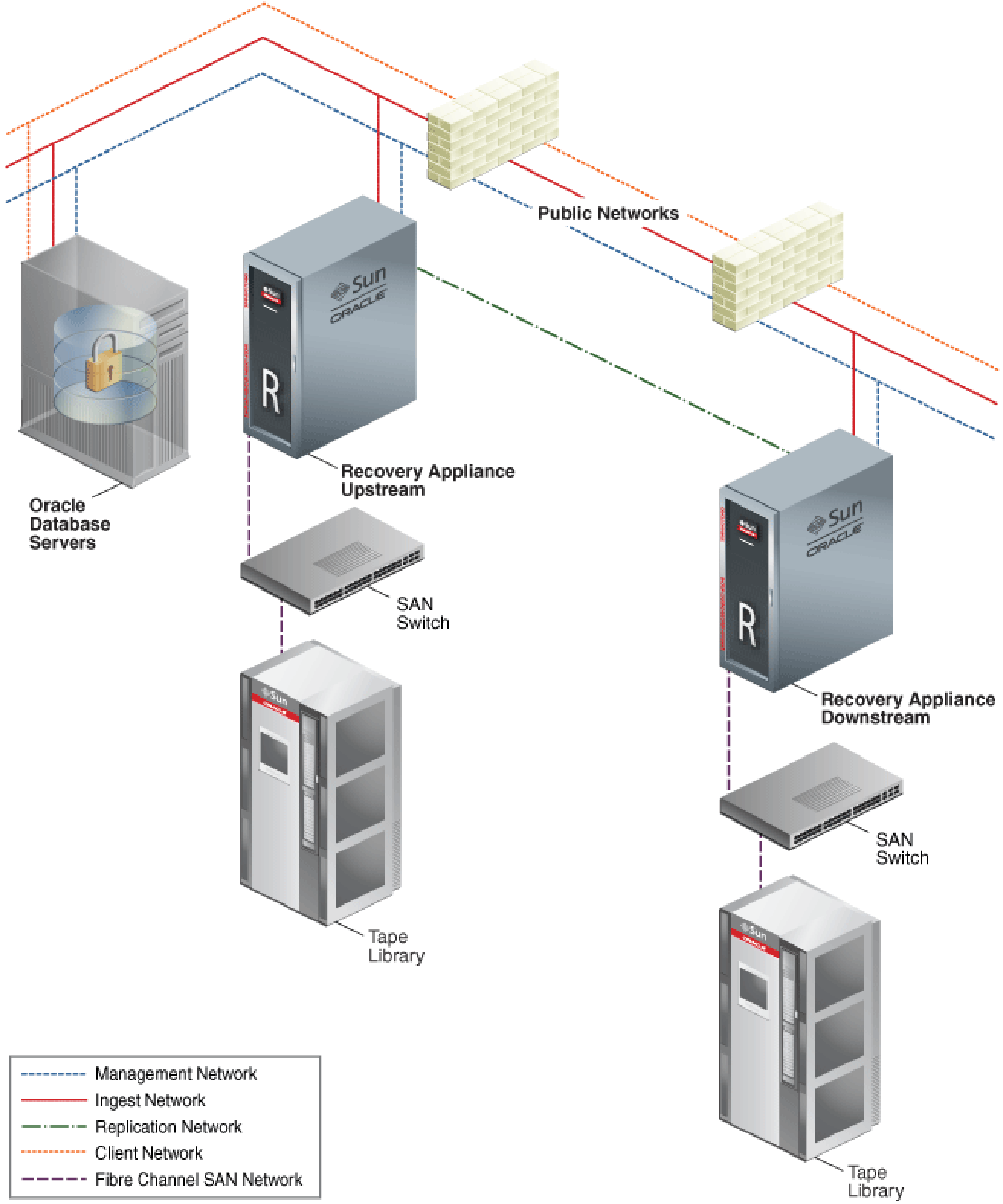
- Tape and Cloud:
  - Can copy backups to Oracle Cloud, Object Store.
    - Uses Key Vault.

  - Can copy to tapes directly, is MML.
    - Can by OSB or Third Part (since is compatible with rman).
    - Oracle Secure Backup (OSB), is used
      - SAN connection

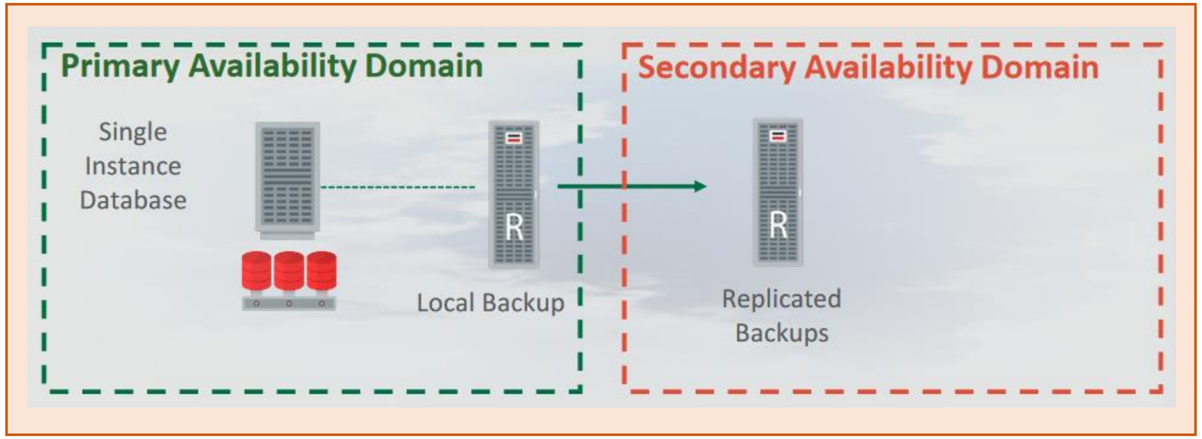
  - Offload/copy backupsets.

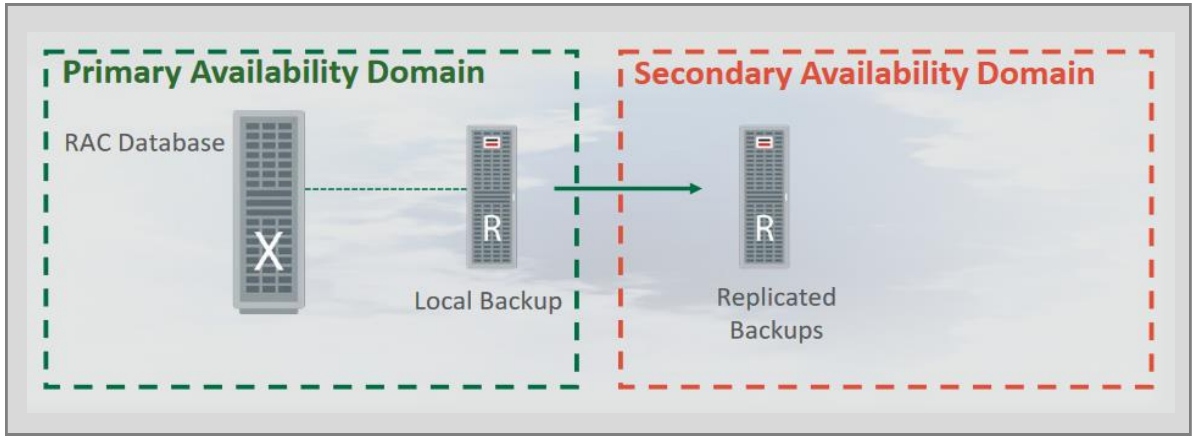
  - Totally integrated with RMAN catalog.

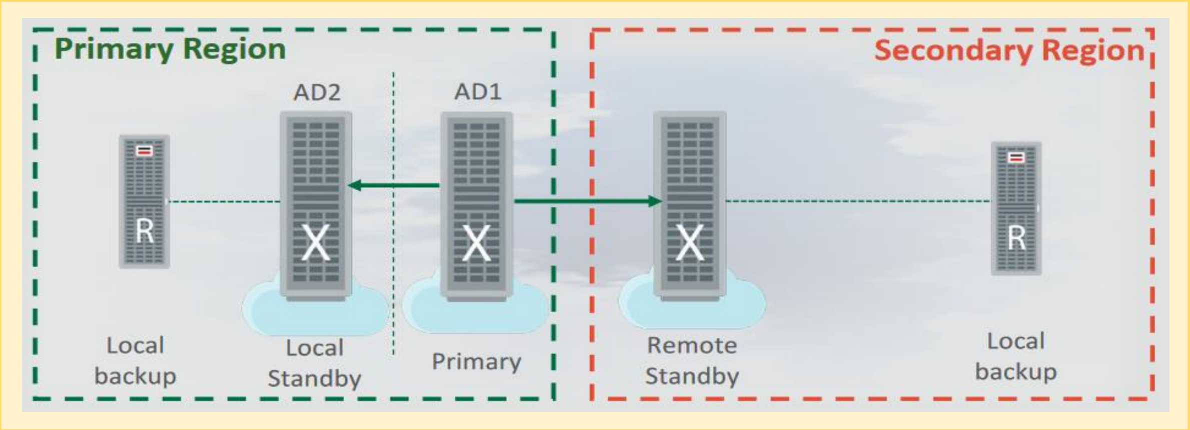
ZDLRA

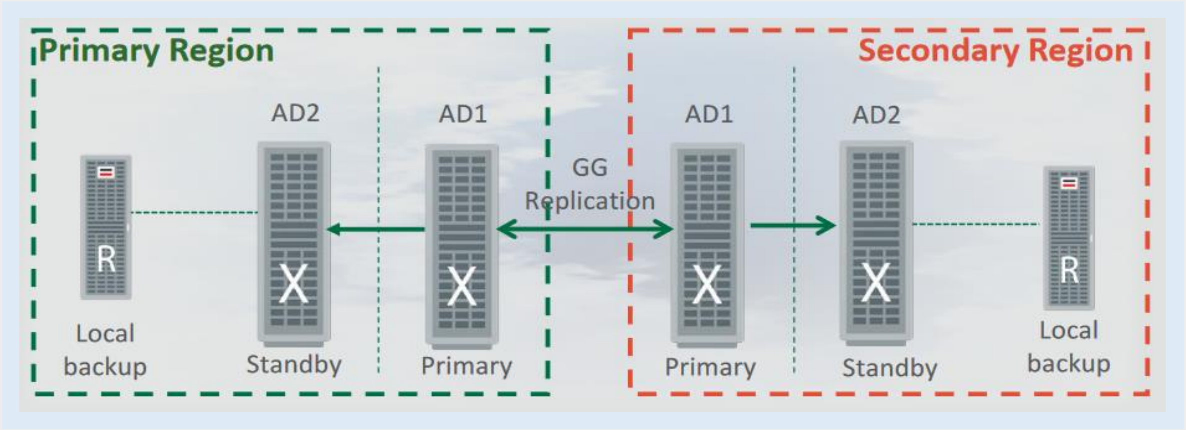

Public Networks
Oracle Database Servers
Recovery Appliance Upstream
SAN Switch
Tape Library
Recovery Appliance Downstream
SAN Switch
Tape Library
Management Network
Ingest Network
Replication Network
Client Network
Fibre Channel SAN Network

ZDLRA


Primary Availability Domain
Secondary Availability Domain
Single Instance Database
R
Local Backup
R
Replicated Backups


Primary Availability Domain
Secondary Availability Domain
RAC Database
X
R
Local Backup
R
Replicated Backups


Primary Region
Secondary Region
AD2
AD1
R
Local backup
X
Local Standby
X
Primary
X
Remote Standby
R
Local backup


Primary Region
Secondary Region
AD2
AD1
AD1
AD2
R
Local backup
X
Standby
X
Primary
GG Replication
X
Primary
X
Standby
R
Local backup

Enrolling Database

- The process to protect database it is simple.
  - Define VPC user and policies inside ZDLRA database.
  - Install library in the client.
  - Register database with RMAN.
  - Backup.

Enrolling Database

- Create Policy
  - Policy is important part for ZDLRA maintenance and reliability.
    - Good ZDLRA project start with good policy definition.
  - Every database is linked with one policy.
  - It is where you define recovery window goal and max retention.
  - ZDLRA control the backup deletion based in the protection policy definitions.
    - If you define 30 days for recovery window at rman, but for policy you define 15 days, the backups will be removed after 15 days.

Enrolling Database

- Create Policy
  - Create used DBMS_RA.CREATE_PROTECTION_POLICY (or EM/CC):
    - https://docs.oracle.com/en/engineered-systems/zero-data-loss-recovery-appliance/19.2/amagd/amagd_dbms.html#GUID-1EFA0233-F743-41C7-9331-C5FA468EA7D5

```
SQL> BEGIN
  2  DBMS_RA.CREATE_PROTECTION_POLICY(
  3      protection_policy_name => 'ZDLRA_WEBINAR'
  4      , description => 'Policy ZDLRA WEBINAR'
  5      , storage_location_name => 'DELTA'
  6      , recovery_window_goal => INTERVAL '1' DAY
  7      , max_retention_window => INTERVAL '2' DAY
  8      , recovery_window_sbt => INTERVAL '5' DAY
  9      , guaranteed_copy => 'NO'
 10      , allow_backup_deletion => 'YES'
 11  );
 12  END;
 13  /

PL/SQL procedure successfully completed.

SQL>
```

Enrolling Database

- Create VPC
  - **As root** at you ZDLRA server node create the Virtual Private Catalog (VPC) user:

```
[root@zdlras1n1 ~]# /opt/oracle.RecoveryAppliance/bin/racli add vpc_user --user_name=vpcwebi
[vpcwebi] New Password: s3nh4web1
Sun Dec 15 19:20:52 2019: Start: Add vpc user vpcwebi.
Sun Dec 15 19:20:53 2019:        Add vpc user vpcwebi successfully.
Sun Dec 15 19:20:53 2019: End:   Add vpc user vpcwebi.
[root@zdlras1n1 ~]#
```

  - Don’t need if you already have one.
    - You can have more than one.
  - This define the catalog name for your RMAN.
  - The correct way is using racli command to avoid errors:
    - Example: forgot to use digest when creating the user with sqlplus. Replication will fails.

Enrolling Database

- Add database
  - Link the VPC user with the policy.
  - Defines the reserved space for database inside the “delta storage”.

```
SQL> BEGIN
  2  DBMS_RA.ADD_DB(
  3      db_unique_name => 'ORCL18C'
  4      , protection_policy_name => 'ZDLRA_WEBINAR'
  5      , reserved_space => '5G'
  6  );
  7  END;
  8  /

PL/SQL procedure successfully completed.

SQL>
```

```
SQL> BEGIN
  2  DBMS_RA.GRANT_DB_ACCESS (
  3      db_unique_name => 'ORCL18C'
  4      , username => 'vpcwebi'
  5  );
  6  END;
  7  /

PL/SQL procedure successfully completed.

SQL>
```

- If want to move between catalogs, just grant using the procedure above.

Enrolling Database

- Client configuration
  - The steps are:
    - Library installation.
    - Wallet and config file.
    - Register database and allocate rman channel.

  - Download library from MOS note 2219812.1

  - Unzip and copy to $ORACLE_HOME/lib

```
[oracle@orcloel7 tmp]$ unzip ra_linux64.zip
Archive:  ra_linux64.zip
  inflating: libra.so
  inflating: metadata.xml
[oracle@orcloel7 tmp]$
[oracle@orcloel7 tmp]$ cp libra.so $ORACLE_HOME/lib/libra.so
[oracle@orcloel7 tmp]$
```

Enrolling Database

- Client configuration
  - Create the wallet for VPC:
    - The wallet is needed to allow access avoiding password request.

```
[oracle@orcloel7 tmp]$ mkstore -wrl $ORACLE_HOME/dbs/ra_wallet -createALO
Oracle Secret Store Tool Release 18.0.0.0.0 - Production
Version 18.1.0.0.0
Copyright (c) 2004, 2017, Oracle and/or its affiliates. All rights reserved.

[oracle@orcloel7 tmp]$
```

  - Add the credential to wallet.

```
[oracle@orcloel7 tmp]$ mkstore -wrl $ORACLE_HOME/dbs/ra_wallet -createCredential zdlras1-scan:1521/zdlras1:VPCWEBI VPCWEBI s3nh4web1
Oracle Secret Store Tool Release 18.0.0.0.0 - Production
Version 18.1.0.0.0
Copyright (c) 2004, 2017, Oracle and/or its affiliates. All rights reserved.

[oracle@orcloel7 tmp]$
```

  - **I recommend that credential alias uses an EZCONNECT format because you can easily identify the details.**

Enrolling Database

- Client configuration
  - **Create ra file**.
    - This file is used to inform the library and database what is the credential to be used (in case of Real-Time Redo as example).
    - The pattern is ra<instance_name> at $ORACLE_HOME/dbs

```
[oracle@orcloel7 dbs]$ cat raORCL18C.ora

RA_WALLET='LOCATION=file:/u01/app/oracle/product/18.6.0.0/dbhome_1/dbs/ra_wallet CREDENTIAL_ALIAS=zdlras1-scan:1521/zdlras1:VPCWEBI'

[oracle@orcloel7 dbs]$
```

Enrolling Database

- Register database
  - Connect to the RMAN catalog and register database

```
[oracle@orcloel7 ~]$ rman target=/ catalog=vpcwebi/s3nh4web1@zdlras1-scan:1521/zdlras1

Recovery Manager: Release 18.0.0.0.0 - Production on Sun Dec 15 22:08:08 2019
Version 18.6.0.0.0

Copyright (c) 1982, 2018, Oracle and/or its affiliates.  All rights reserved.

connected to target database: ORCL18C (DBID=558466555)
connected to recovery catalog database
recovery catalog schema version 19.03.00.00. is newer than RMAN version

RMAN>
```

  - Register database

```
RMAN> register database;

database registered in recovery catalog
starting full resync of recovery catalog
full resync complete

RMAN>
```

Enrolling Database

- Channel configuration:

```
RMAN> CONFIGURE CHANNEL 1 DEVICE TYPE 'SBT_TAPE' FORMAT  '%d_%U' PARMS  "SBT_LIBRARY=/u01/app/oracle/product/18.6.0.0/dbhome_1/lib/libra.so,
ENV=(RA_WALLET='location=file:/u01/app/oracle/product/18.6.0.0/dbhome_1/dbs/ra_wallet credential_alias=zdlras1-scan:1521/zdlras1:VPCWEBI')";

new RMAN configuration parameters:
CONFIGURE CHANNEL 1 DEVICE TYPE 'SBT_TAPE' FORMAT   '%d_%U' PARMS  "SBT_LIBRARY=/u01/app/oracle/product/18.6.0.0/dbhome_1/lib/libra.so,
ENV=(RA_WALLET='location=file:/u01/app/oracle/product/18.6.0.0/dbhome_1/dbs/ra_wallet credential_alias=zdlras1-scan:1521/zdlras1:VPCWEBI')";
new RMAN configuration parameters are successfully stored
starting full resync of recovery catalog
full resync complete

RMAN>
```

- Look the details:
  - Device type.
  - Library location.
  - Credential location.
  - Credential name.

LUXOUG
LUXEMBOURG ORACLE USER GROUP

Enrolling Database

- Backup Database:

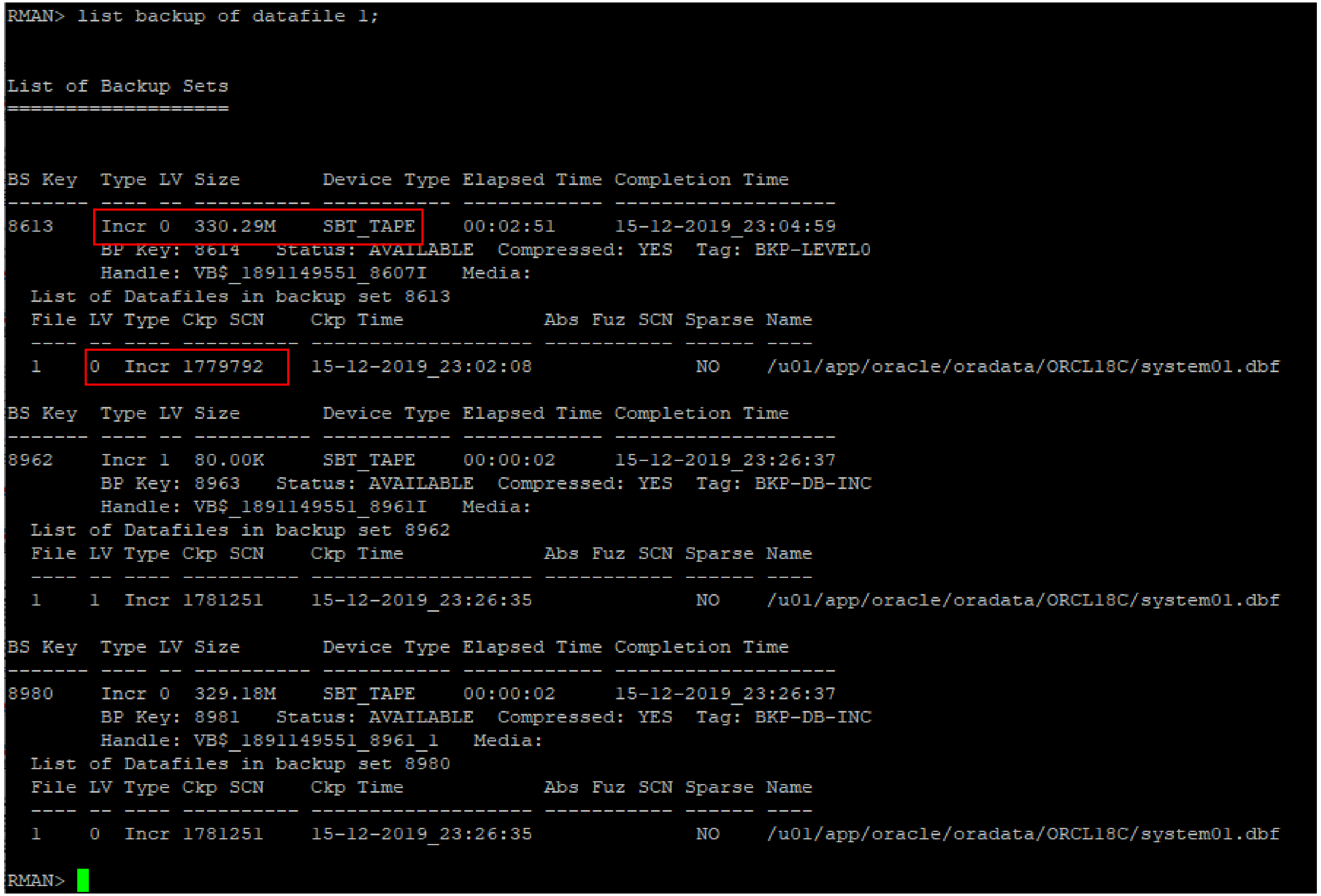
```
RMAN> BACKUP INCREMENTAL LEVEL 0 DEVICE TYPE SBT FILESPERSET 1 FORMAT '%U' DATABASE TAG 'BKP-LEVEL0';

Starting backup at 15-12-2019_23:01:54
allocated channel: ORA_SBT_TAPE_1
channel ORA_SBT_TAPE_1: SID=7 device type=SBT_TAPE
channel ORA_SBT_TAPE_1: RA Library (ZDLRAS1) SID=99C63E35B30630EAE053010310ACEB04
channel ORA_SBT_TAPE_1: starting incremental level 0 datafile backup set
channel ORA_SBT_TAPE_1: specifying datafile(s) in backup set
input datafile file number=00001 name=/u01/app/oracle/oradata/ORCL18C/system01.dbf
…
channel ORA_SBT_TAPE_1: finished piece 1 at 15-12-2019_23:15:37
piece handle=44ujh7cm_1_1 tag=BKP-LEVEL0 comment=API Version 2.0,MMS Version 12.2.0.2
channel ORA_SBT_TAPE_1: backup set complete, elapsed time: 00:00:03
Finished backup at 15-12-2019_23:15:37

Starting Control File and SPFILE Autobackup at 15-12-2019_23:15:37
piece handle=c-558466555-20191215-00 comment=API Version 2.0,MMS Version 12.2.0.2
Finished Control File and SPFILE Autobackup at 15-12-2019_23:15:57

RMAN>
```

Enrolling Database

- **Subsequent Level 1**

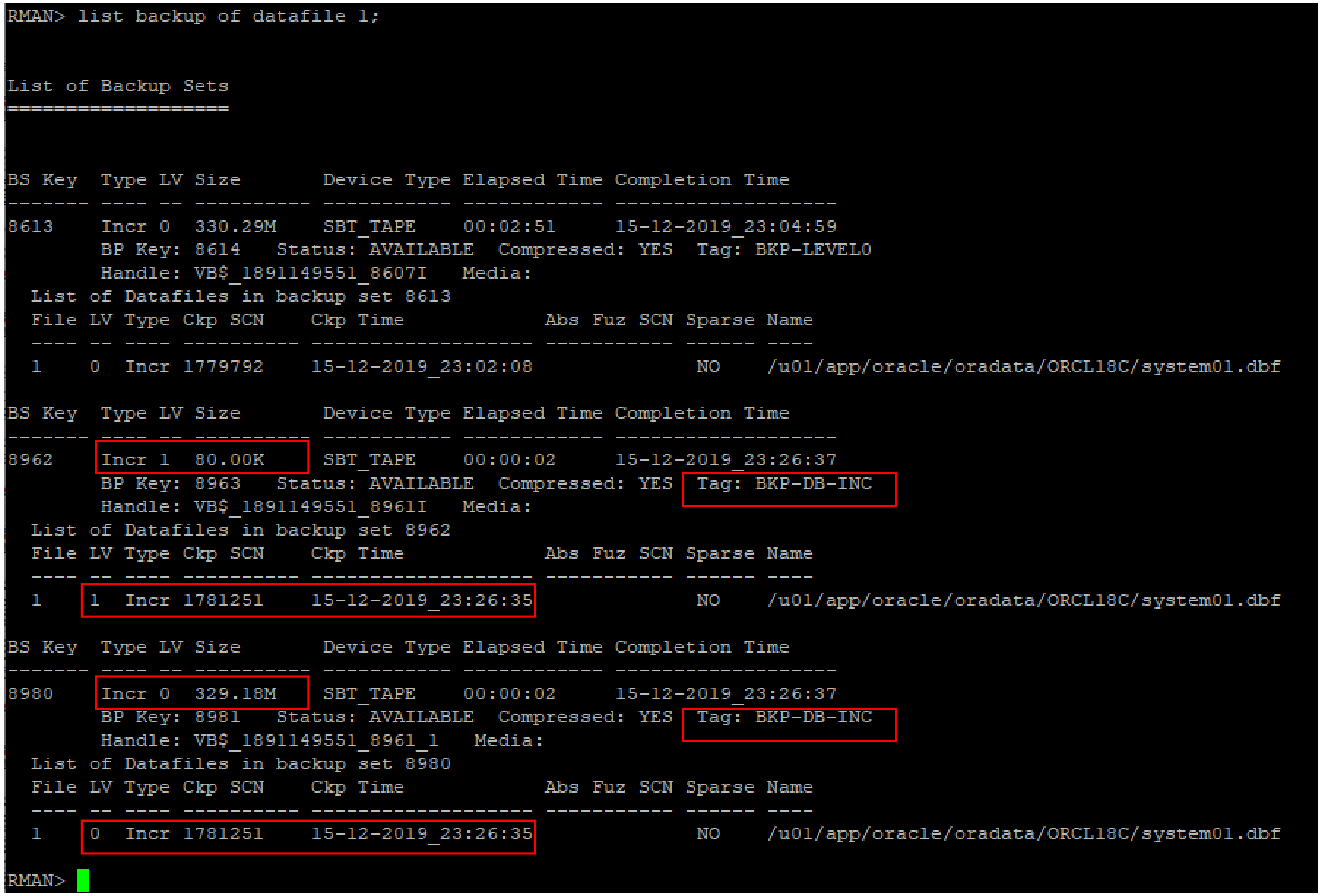
```
RMAN> BACKUP INCREMENTAL LEVEL 1 DEVICE TYPE SBT FILESPERSET 1 DATABASE TAG 'BKP-DB-INC';

Starting backup at 15-12-2019_23:22:20
using channel ORA_SBT_TAPE_1
channel ORA_SBT_TAPE_1: starting incremental level 1 datafile backup set
channel ORA_SBT_TAPE_1: specifying datafile(s) in backup set
input datafile file number=00003 name=/u01/app/oracle/oradata/ORCL18C/sysaux01.dbf
channel ORA_SBT_TAPE_1: starting piece 1 at 15-12-2019_23:22:22
channel ORA_SBT_TAPE_1: finished piece 1 at 15-12-2019_23:22:25
piece handle=ORCL18C_48ujh7pe_1_1 tag=BKP-DB-INC comment=API Version 2.0,MMS Version 12.2.0.2
…
channel ORA_SBT_TAPE_1: specifying datafile(s) in backup set
input datafile file number=00012 name=/u01/app/oracle/oradata/ORCL18C/ORCL18P/users01.dbf
channel ORA_SBT_TAPE_1: starting piece 1 at 15-12-2019_23:22:33
channel ORA_SBT_TAPE_1: finished piece 1 at 15-12-2019_23:22:36
piece handle=ORCL18C_4hujh7pp_1_1 tag=BKP-DB-INC comment=API Version 2.0,MMS Version 12.2.0.2
channel ORA_SBT_TAPE_1: backup set complete, elapsed time: 00:00:03

RMAN>
```

Enrolling Database



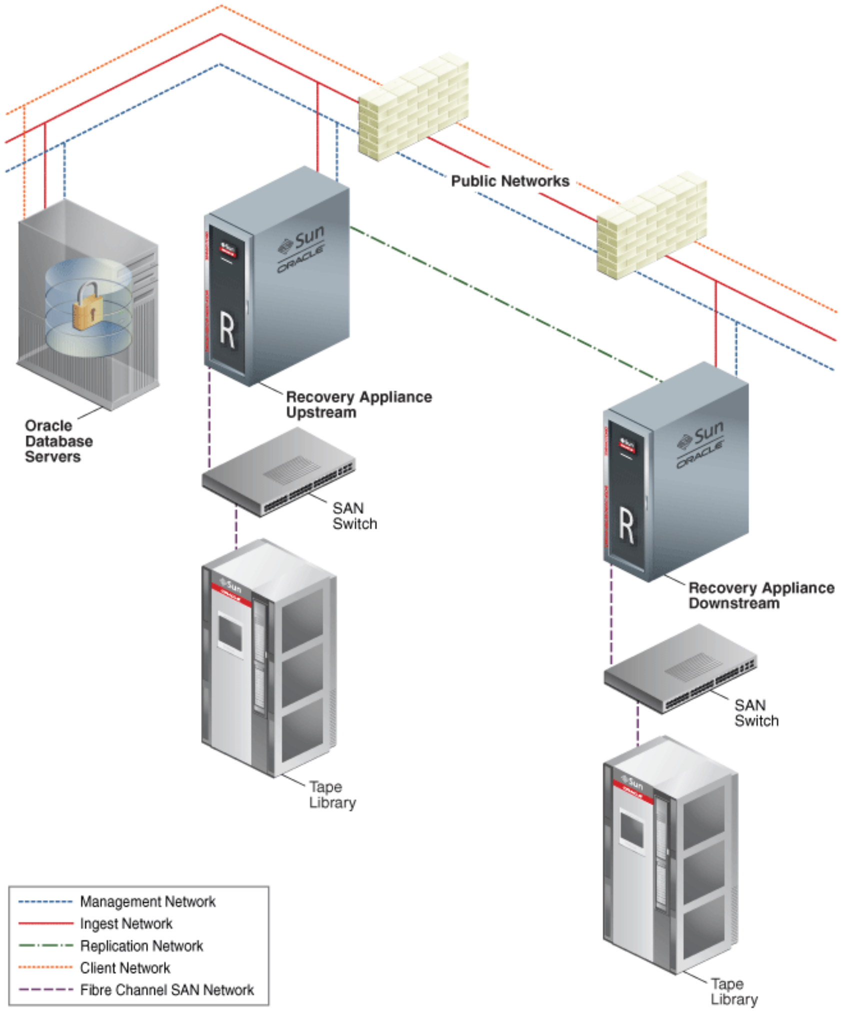
```
RMAN> list backup of datafile 1;


List of Backup Sets
===================


BS Key  Type LV Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ -------------------
8613    Incr 0  330.29M    SBT_TAPE    00:02:51     15-12-2019_23:04:59
        BP Key: 8614   Status: AVAILABLE  Compressed: YES  Tag: BKP-LEVEL0
        Handle: VB$_1891149551_8607I   Media:
  List of Datafiles in backup set 8613
  File LV Type Ckp SCN    Ckp Time            Abs Fuz SCN Sparse Name
  ---- -- ---- ---------- ------------------- ----------- ------ ----
  1    0  Incr 1779792    15-12-2019_23:02:08              NO     /u01/app/oracle/oradata/ORCL18C/system01.dbf

BS Key  Type LV Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ -------------------
8962    Incr 1  80.00K     SBT_TAPE    00:00:02     15-12-2019_23:26:37
        BP Key: 8963   Status: AVAILABLE  Compressed: YES  Tag: BKP-DB-INC
        Handle: VB$_1891149551_8961I   Media:
  List of Datafiles in backup set 8962
  File LV Type Ckp SCN    Ckp Time            Abs Fuz SCN Sparse Name
  ---- -- ---- ---------- ------------------- ----------- ------ ----
  1    1  Incr 1781251    15-12-2019_23:26:35              NO     /u01/app/oracle/oradata/ORCL18C/system01.dbf

BS Key  Type LV Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ -------------------
8980    Incr 0  329.18M    SBT_TAPE    00:00:02     15-12-2019_23:26:37
        BP Key: 8981   Status: AVAILABLE  Compressed: YES  Tag: BKP-DB-INC
        Handle: VB$_1891149551_8961_1   Media:
  List of Datafiles in backup set 8980
  File LV Type Ckp SCN    Ckp Time            Abs Fuz SCN Sparse Name
  ---- -- ---- ---------- ------------------- ----------- ------ ----
  1    0  Incr 1781251    15-12-2019_23:26:35              NO     /u01/app/oracle/oradata/ORCL18C/system01.dbf

RMAN>
```

Enrolling Database

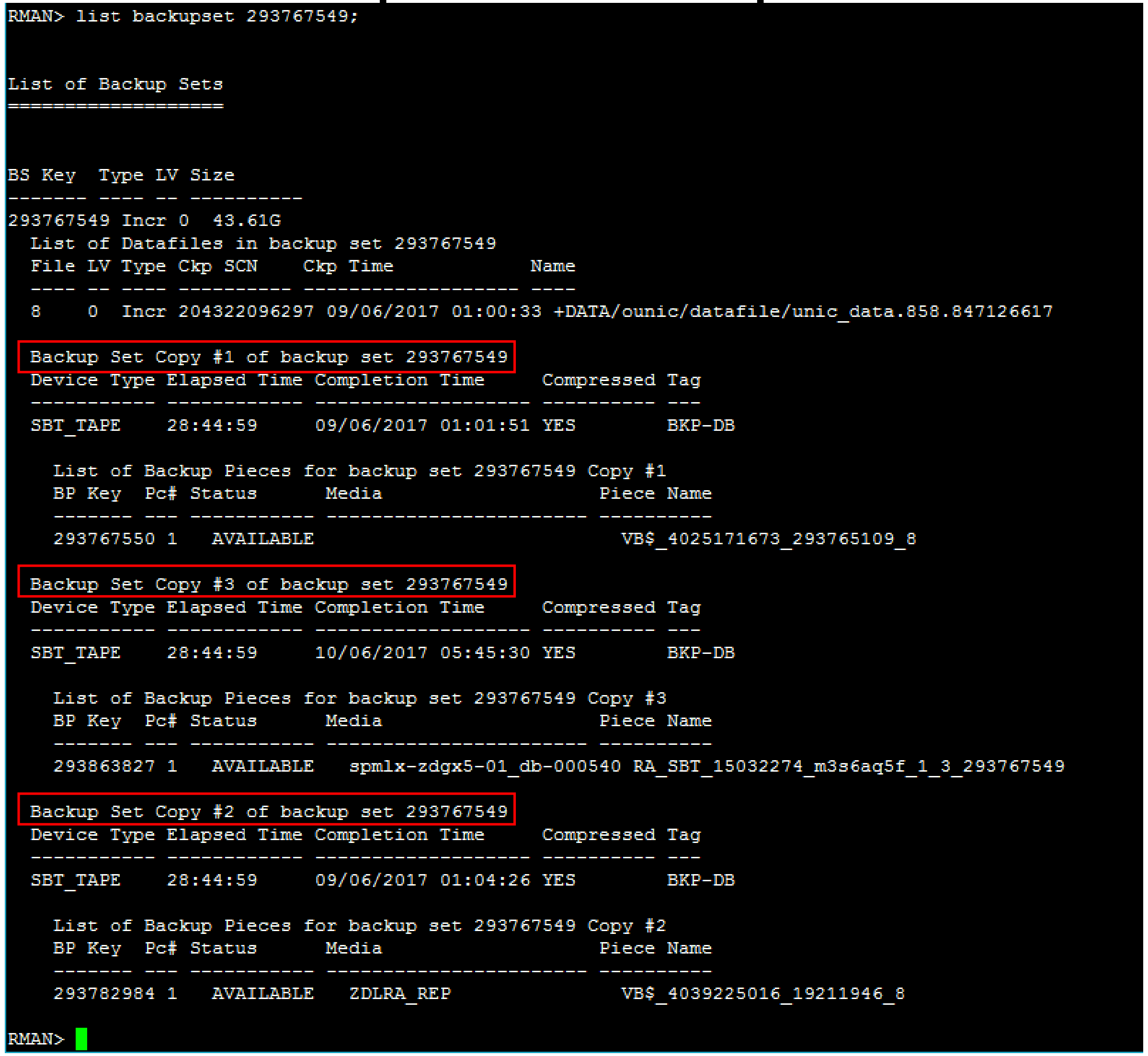


```
RMAN> list backup of datafile 1;


List of Backup Sets
===================


BS Key  Type LV Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ -------------------
8613    Incr 0  330.29M    SBT_TAPE    00:02:51     15-12-2019_23:04:59
        BP Key: 8614   Status: AVAILABLE  Compressed: YES  Tag: BKP-LEVEL0
        Handle: VB$_1891149551_8607I   Media:
  List of Datafiles in backup set 8613
  File LV Type Ckp SCN    Ckp Time            Abs Fuz SCN Sparse Name
  ---- -- ---- ---------- ------------------- ----------- ------ ----
  1    0  Incr 1779792    15-12-2019_23:02:08              NO     /u01/app/oracle/oradata/ORCL18C/system01.dbf

BS Key  Type LV Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ -------------------
8962    Incr 1  80.00K     SBT_TAPE    00:00:02     15-12-2019_23:26:37
        BP Key: 8963   Status: AVAILABLE  Compressed: YES  Tag: BKP-DB-INC
        Handle: VB$_1891149551_8961I   Media:
  List of Datafiles in backup set 8962
  File LV Type Ckp SCN    Ckp Time            Abs Fuz SCN Sparse Name
  ---- -- ---- ---------- ------------------- ----------- ------ ----
  1    1  Incr 1781251    15-12-2019_23:26:35              NO     /u01/app/oracle/oradata/ORCL18C/system01.dbf

BS Key  Type LV Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ -------------------
8980    Incr 0  329.18M    SBT_TAPE    00:00:02     15-12-2019_23:26:37
        BP Key: 8981   Status: AVAILABLE  Compressed: YES  Tag: BKP-DB-INC
        Handle: VB$_1891149551_8961_1   Media:
  List of Datafiles in backup set 8980
  File LV Type Ckp SCN    Ckp Time            Abs Fuz SCN Sparse Name
  ---- -- ---- ---------- ------------------- ----------- ------ ----
  1    0  Incr 1781251    15-12-2019_23:26:35              NO     /u01/app/oracle/oradata/ORCL18C/system01.dbf

RMAN>
```

Replicated Backup and Tape



Replicated Backup and Tape



```
RMAN> list backupset 293767549;


List of Backup Sets
===================


BS Key  Type LV Size
------- ---- -- ----------
293767549 Incr 0  43.61G
  List of Datafiles in backup set 293767549
  File LV Type Ckp SCN    Ckp Time            Name
  ---- -- ---- ---------- ------------------- ----
  8    0  Incr 204322096297 09/06/2017 01:00:33 +DATA/ounic/datafile/unic_data.858.847126617

  Backup Set Copy #1 of backup set 293767549
  Device Type Elapsed Time Completion Time     Compressed Tag
  ----------- ------------ ------------------- ---------- ---
  SBT_TAPE    28:44:59     09/06/2017 01:01:51 YES        BKP-DB

    List of Backup Pieces for backup set 293767549 Copy #1
    BP Key  Pc# Status      Media                   Piece Name
    ------- --- ----------- ----------------------- ----------
    293767550 1   AVAILABLE                           VB$_4025171673_293765109_8

  Backup Set Copy #3 of backup set 293767549
  Device Type Elapsed Time Completion Time     Compressed Tag
  ----------- ------------ ------------------- ---------- ---
  SBT_TAPE    28:44:59     10/06/2017 05:45:30 YES        BKP-DB

    List of Backup Pieces for backup set 293767549 Copy #3
    BP Key  Pc# Status      Media                   Piece Name
    ------- --- ----------- ----------------------- ----------
    293863827 1   AVAILABLE   spmlx-zdgx5-01_db-000540 RA_SBT_15032274_m3s6aq5f_1_3_293767549

  Backup Set Copy #2 of backup set 293767549
  Device Type Elapsed Time Completion Time     Compressed Tag
  ----------- ------------ ------------------- ---------- ---
  SBT_TAPE    28:44:59     09/06/2017 01:04:26 YES        BKP-DB

    List of Backup Pieces for backup set 293767549 Copy #2
    BP Key  Pc# Status      Media                   Piece Name
    ------- --- ----------- ----------------------- ----------
    293782984 1   AVAILABLE   ZDLRA_REP               VB$_4039225016_19211946_8

RMAN>
```

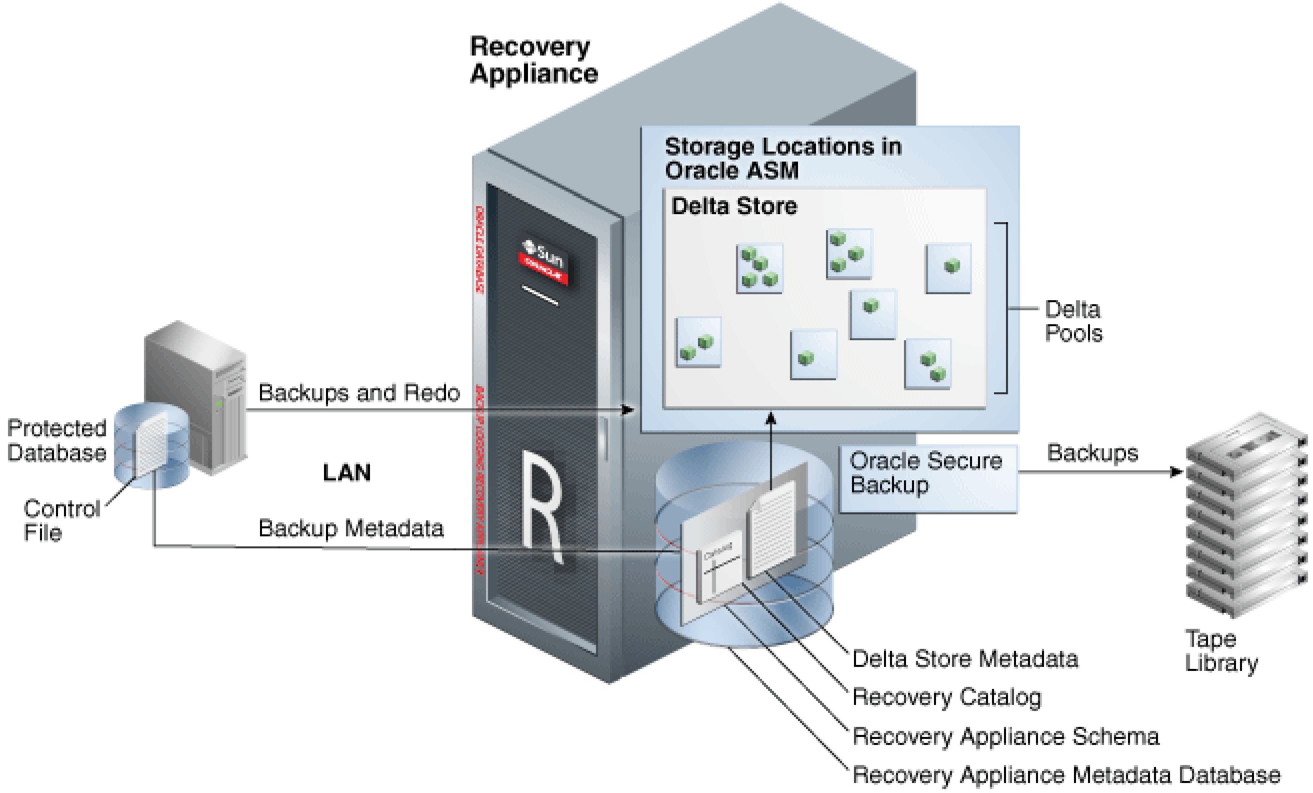
Replicated Backup and Tape

```
RMAN> list backupset 390136633;


List of Backup Sets
===================


BS Key  Type LV Size
------- ---- -- ----------
390136633 Incr 0  31.91G
  List of Datafiles in backup set 390136633
  File LV Type Ckp SCN    Ckp Time            Name
  ---- -- ---- ---------- ------------------- ----
  1    0  Incr 441288188747 23/10/2019 00:09:33 +DATAX6/dbpro1/datafile/system.288.954160839

  Backup Set Copy #3 of backup set 390136633
  Device Type Elapsed Time Completion Time     Compressed Tag
  ----------- ------------ ------------------- ---------- ---
  SBT_TAPE    75:51:55     26/10/2019 04:01:21 YES        BKP-DB

    List of Backup Pieces for backup set 390136633 Copy #3
    BP Key  Pc# Status      Media                   Piece Name
    ------- --- ----------- ----------------------- ----------
    390913248 1   AVAILABLE   serva-zdlrax01_db-000985 RA_SBT_DBPRO1_200734097_15032274_tduf09tm_1_3_390136633

  Backup Set Copy #2 of backup set 390136633
  Device Type Elapsed Time Completion Time     Compressed Tag
  ----------- ------------ ------------------- ---------- ---
  SBT_TAPE    75:51:55     23/10/2019 00:28:00 YES        BKP-DB

    List of Backup Pieces for backup set 390136633 Copy #2
    BP Key  Pc# Status      Media                   Piece Name
    ------- --- ----------- ----------------------- ----------
    390140512 1   AVAILABLE   ZDLRA_REP               VB$_4039225016_51014225_1

RMAN>
```

Replicated Backup and Tape

```
RMAN> list backupset 386538643;


List of Backup Sets
===================


BS Key  Type LV Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ -------------------
386538643 Incr 0  31.91G     SBT_TAPE    121:21:08    01/10/2019 02:31:12
        BP Key: 387590272   Status: AVAILABLE  Compressed: NO  Tag: BKP-DB
        Handle: RA_SBT_DBPRO_200734097_15032274_bkucmivc_1_3_386538643   Media: exadb-zdlx5-01_db-000983
  List of Datafiles in backup set 386538643
  File LV Type Ckp SCN    Ckp Time            Name
  ---- -- ---- ---------- ------------------- ----
  1    0  Incr 431380649775 26/09/2019 01:10:05 +DATAX6/dbpro/datafile/system.288.954160839

RMAN>
```

Internals Details



Internals Details

- Storage
  - Is a Database:
    - CATALOG for RMAN and ZDLRA metadata tables.
    - DELTA for backups blocks:
      - **Don’t see backups, see blocks**
      - Store in containers files inside ASM.

    - For tape, OSB format, it is an entire backupset

```
ASMCMD> pwd
+DELTA/ZDLRA/CONTAINER
ASMCMD> ls -s
Block_Size      Blocks           Bytes           Space  Name
       512  4294959104  2199019061248  4398059094016  con.561.935405781
       512  4294959104  2199019061248  4398059094016  con.562.935405789
…
…
       512  4294959104  2199019061248  4398059094016  con.656.935406489
       512   308051968   157722607616   315692679168  con.657.935406495
ASMCMD>
```

Internals Details

- Normal rman catalog with tables and views
  - RC_*
    - RC_DATABASE
    - RC_BACKUP_SET
    - RC…..
    - DB
    - BS
    - …

Internal Details

- ZDLRA tables and views: **RASYS owner**
  - RA_DATABASE: Information about databases for ZDLRA, policy and space usage.
  - RA_CONFIG: Information about internal parameters, like network chunk size.
  - RA_TASK: All task running inside ZDLRA.
  - VBDF: Store the reference for **V**irtual **B**ackup **D**ata **F**ile, including the source backup piece that was used to create the virtual backup
  - PLANS: The number of plans for one virtual backup.
  - PLANS_DETAIL: Details for the plan linked to the virtual backup:
    - **Together are the INDEX_BACKUP**
  - BLOCKS: Store information for each block for datafile. Is as matrix for every block inside zdlra database for one datafile

Internal Details

- INDEX_BACKUP
  - For ZDLRA, the task type INDEX_BACKUP is one of the most important because is responsible to create the virtual full backup. This task runs for every backup that you ingest at ZDLRA
  - INDEX_BACKUP is also responsible to “fix” the rman catalog views to “show” the new backups.
    - Two major phases (*fixup_unordered* and *q_restore_fast*) to read ingested blocks and index it.
  - About the blocks, it is important to hint that is completely based in SCN/CKP number.
    - The index creation will search for all blocks that are bellow of the SCN of ingested backup.

Internal Details

- Backup Datafile:

```
RMAN> BACKUP INCREMENTAL LEVEL 0 DEVICE TYPE SBT FILESPERSET 1 DATAFILE 5;

Starting backup at 22-09-2019_17:54:27
...
Finished Control File and SPFILE Autobackup at 22-09-2019_17:54:30

RMAN>

RMAN> list backup of datafile 5;


List of Backup Sets
===================


BS Key  Type LV Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ -------------------
2729    Incr 0  40.00K     SBT_TAPE    00:00:02     22-09-2019_17:54:30
        BP Key: 2730   Status: AVAILABLE  Compressed: YES  Tag: TAG20190922T175427
        Handle: VB$_1887643964_2728I   Media:
  List of Datafiles in backup set 2729
  File LV Type Ckp SCN    Ckp Time            Abs Fuz SCN Sparse Name
  ---- -- ---- ---------- ------------------- ----------- ------ ----
  5    0  Incr 2320763    22-09-2019_17:54:28              NO     /u01/app/oracle/oradata/ORCL19/simon01.dbf

RMAN>
```

Internal Details

- Inside ZDLRA tables:

```
SQL> select db_key, dbinc_key from rc_database where name = 'ORCL19';

    DB_KEY  DBINC_KEY
---------- ----------
      2202       2203

SQL>
SQL> select df_key, file#, ts#, create_scn, create_time, block_size, blocks from df where dbinc_key = 2203 and file# = 5;

    DF_KEY      FILE#        TS# CREATE_SCN CREATE_TI BLOCK_SIZE     BLOCKS
---------- ---------- ---------- ---------- --------- ---------- ----------
      2689          5          6    2319183 22-SEP-19       8192        128

SQL>
SQL> select vb_key, ckp_scn, vcbp_key, srcbp_key, blocks, chunks_used from vbdf where db_key = 2202 and df_key = 2689 order by vb_key
asc;

    VB_KEY    CKP_SCN   VCBP_KEY  SRCBP_KEY     BLOCKS CHUNKS_USED
---------- ---------- ---------- ---------- ---------- -----------
      2728    2320763       2730       2701        128           1

SQL>
SQL> select * from plans where db_key = 2202 and df_key = 2689 order by vb_key asc;

      TYPE     DB_KEY     VB_KEY     DF_KEY    TASK_ID        OLD   BLKSREAD    MAXRANK  NUMCHUNKS   READSIZE  NEEDCHUNK  FREECHUNK
---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ----------
         8       2202       2728       2689                               19          1          1

SQL>
```

Internal Details

- Plans_details table:

```
SQL> select * from plans_details where df_key = 2689 order by vb_key asc, blockno asc;

    DF_KEY       TYPE     VB_KEY    BLKRANK    BLOCKNO    CHUNKNO    NUMBLKS    COFFSET   NUMBYTES
---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ----------
      2689          8       2728          1          0          1          1       8192      24576
      2689          8       2728          1          2          1         17      32788       2167
      2689          8       2728          1 4294967295          1          1      34955        294

SQL>
```

2728
0
1
2
3
4
5
6
7
8
9
10
11
12
13

- The datafile block 0 (column BLOCKNO) until block 1 (**BLOCKNO+NUNBLKS**) is stored at chunk 1.
- The datafile block 2 (column BLOCKNO) until block 19 (BLOCKNO+NUNBLKS) are stored at chunk 1
- The datafile block 4294967295 (the last block of datafile) is stored at chunks 1.

Internal Details

- New Incremental Backup:

```
RMAN> BACKUP INCREMENTAL LEVEL 1 DEVICE TYPE SBT FILESPERSET 1 DATAFILE 5;

Starting backup at 22-09-2019_18:34:30
...
Finished Control File and SPFILE Autobackup at 22-09-2019_18:34:36

RMAN>
```

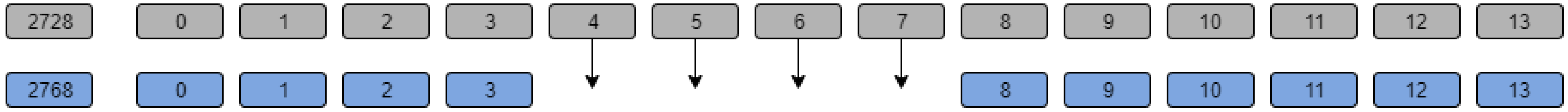
- Inside ZDLRA tables:

```
SQL> select vb_key, ckp_scn, vcbp_key, srcbp_key, blocks, chunks_used from vbdf where db_key = 2202 and df_key = 2689 order by vb_key
asc;

    VB_KEY    CKP_SCN   VCBP_KEY  SRCBP_KEY     BLOCKS CHUNKS_USED
---------- ---------- ---------- ---------- ---------- -----------
      2728    2320763       2730       2701        128           1
      2768    2322525       2770       2735        128           1

SQL>
SQL> select * from plans where db_key = 2202 and df_key = 2689 order by vb_key asc;

      TYPE     DB_KEY     VB_KEY     DF_KEY    TASK_ID        OLD   BLKSREAD    MAXRANK  NUMCHUNKS   READSIZE  NEEDCHUNK  FREECHUNK
---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ----------
         8       2202       2728       2689                               19          1          1
         1       2202       2768       2689                               27          1          2

SQL>
```

Internal Details

```
SQL> select * from plans_details where df_key = 2689 order by vb_key asc, blockno asc;
    DF_KEY       TYPE     VB_KEY    BLKRANK    BLOCKNO    CHUNKNO    NUMBLKS    COFFSET   NUMBYTES
---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ----------
      2689          8       2728          1          0          1          1       8192      24576
      2689          8       2728          1          2          1         17      32788       2167
      2689          8       2728          1 4294967295          1          1      34955        294
      2689          1       2768          1          0       1025          1       8192      24576
      2689          1       2768          1          2       1025          2      32788        252
      2689          1       2768          1          4          1          4      33038        408
      2689          1       2768          1          8       1025         16      33040      45339
      2689          1       2768          1         71          1          3      34703        252
      2689          1       2768          1 4294967295       1025          1      78379        293

9 rows selected.

SQL>
```

- The datafile block 0 (column BLOCKNO) until block 1 (BLOCKNO+NUNBLKS) are stored at chunk 1025.
- The datafile block 2 (column BLOCKNO) until block 4 (BLOCKNO+NUNBLKS) are stored at chunk 1025.
- The datafile block 4 (column BLOCKNO) until block 8 (BLOCKNO+NUNBLKS) are stored at chunk 1 (and came from previous virtual full backup).
- The datafile block 8 (column BLOCKNO) until block 24 (BLOCKNO+NUNBLKS) are stored at chunk 1025.



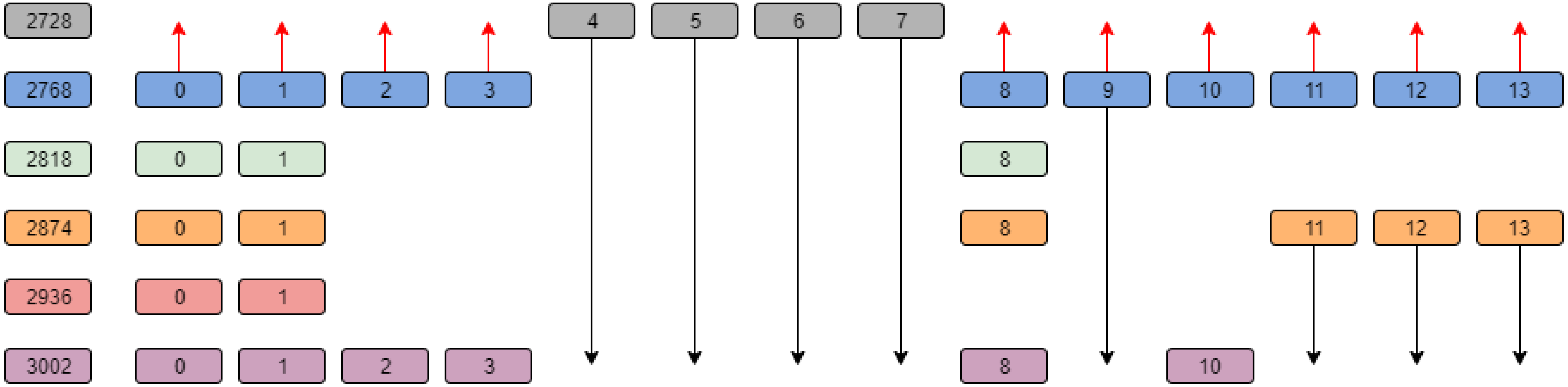
Internal Details

- Automated Delta Poll Space Management

```
SQL> select * from plans_details where df_key = 2689 order by vb_key asc, blockno asc;

    DF_KEY       TYPE     VB_KEY    BLKRANK    BLOCKNO    CHUNKNO    NUMBLKS    COFFSET   NUMBYTES
---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ----------
      2689          1       2768          1          0       1025          1       8192      24576
      2689          1       2768          1          2       1025          2      32788        252
      2689          1       2768          1          4          1          4      33038        408
      2689          1       2768          1          8       1025         16      33040      45339
      2689          1       2768          1         71          1          3      34703        252
      2689          1       2768          1 4294967295       1025          1      78379        293
...
...
      2689          1       3002          1          0       5121          1       8192      24576
      2689          1       3002          1          2       5121          2      32788        255
      2689          1       3002          1          4          1          4      33038        408
      2689          1       3002          1          8       3073          1      32788        262
      2689          1       3002          1          9       1025          1      33293        128
      2689          1       3002          1         10       5121          1      33043        215
      2689          1       3002          1         11       3073         13      33050      94555
      2689          1       3002          1         24       5121         28      33258       4986
      2689          1       3002          1         52          1          1      34703         84
      2689          1       3002          1         56       5121          8      38244       1168
      2689          1       3002          1         72          1          1      34787         84
      2689          1       3002          1        128       5121        843      39412     125460
      2689          1       3002          1 4294967295       5121          1     164872        301

45 rows selected.

SQL>
```

- This means that unnecessary data are deleted from time to time to avoid redundancy and to be more space-efficient.
  - Called “automated delta pool space management”, specifically the “delta pool optimization“.

Internal Details



2728					4	5	6	7						
2768	0	1	2	3					8	9	10	11	12	13
2818	0	1							8					
2874	0	1							8			11	12	13
2936	0	1												
3002	0	1	2	3					8		10			

Internal Details

- Every datafile block that enters is indexed and stored to create the virtual full backup.
  - The idea is linking every virtual full backup of datafile (VBDF table) with one plan (PLANS and PLAN_DETAILS tables).

- Going further, does not exist 1 to 1 relation between backup and the virtual full backup.
  - It is just a matrix of pointers for blocks inside chunks.
  - This is the reason that ZDLRA it is different from other backup appliances; it can analyze block a block and index it efficiently.

Internal details

- Check more details at my blog:
  - ZDLRA, How to enroll a database
  - ZDLRA, Virtual Full Backup and Incremental Forever
  - ZDLRA Internals, INDEX_BACKUP task in details
  - ZDLRA Internals, Virtual Full Backup
  - ZDLRA, Real-Time Redo and Zero RPO
  - ZDLRA, Multi-site protection – ZERO RPO for Primary and Standby

Q&A
When?
Why?
How?
WHAT?
WHERE?
WHEN?
HOW?
WHO?
Where?
What?