

DOAG Conference 2018

20. – 22. November
Nürnberg

RAC for Single Instance DBAs

About me

- 12+ years Database experience
- 2+ years Oracle DBA
- 3rd presentation on DOAG
- Focus on RAC



ORACLE®

Certified Professional

Oracle Database 12c
Administrator

ORACLE®

Certified Expert

Oracle Database 12c:
Oracle RAC and Oracle
Grid Infrastructure
Administrator

■ Agenda

■ RAC Architecture

- SPFILE, Views, Redo, Undo, Temp, Datafiles

■ Connections

- Services, Sessions, Listener, Connections

■ Administration

- Shutdown & Startup, Checkpoint, Log Switch

■ Further Topics

- Multitenant, Performance, Backup & Restore

■ **RAC Architecture**

■ **SPFILE, Views, Redo, Undo, Temp, Datafiles**

- **Connections**

- Services, Sessions, Listener, Connections

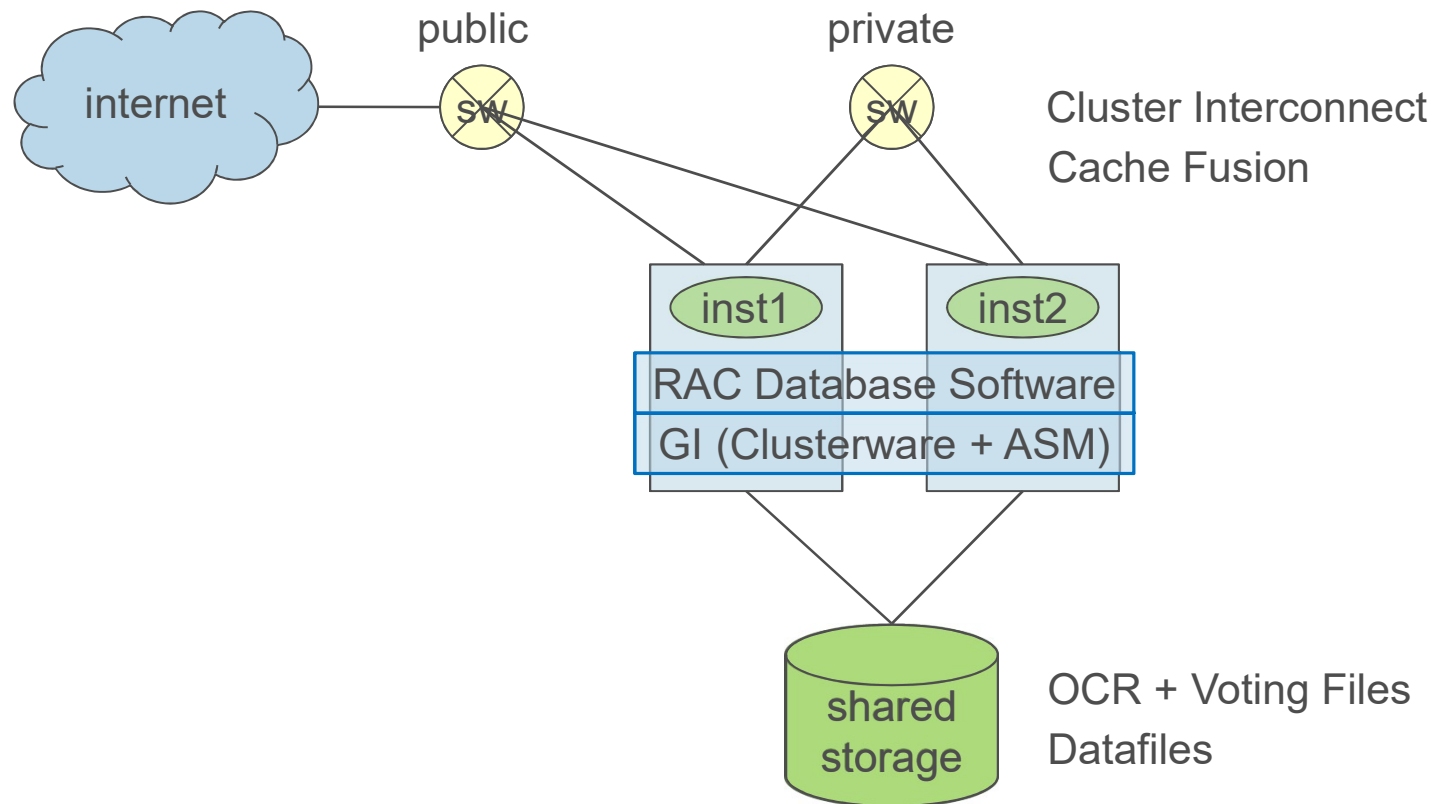
- **Administration**

- Shutdown & Startup, Checkpoint, Log Switch

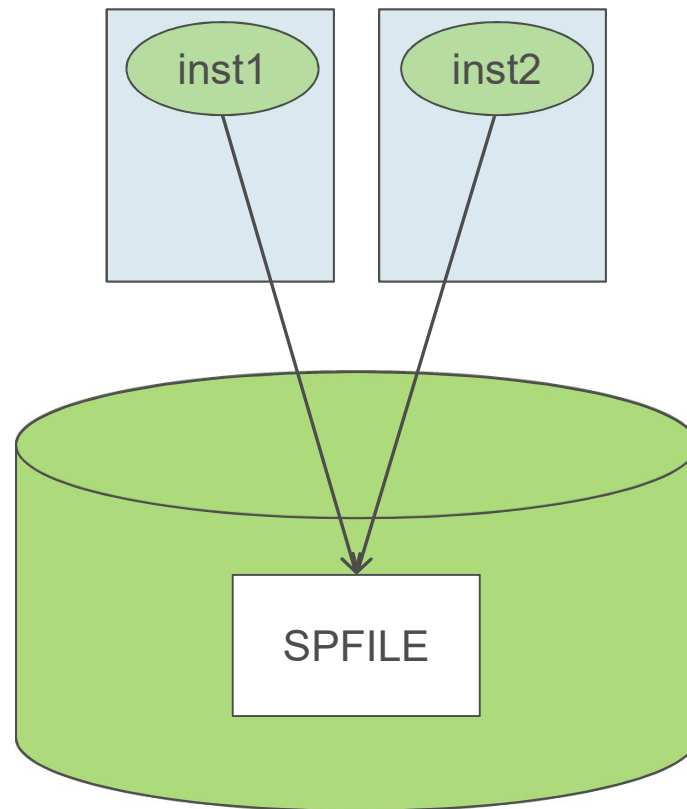
- **Further Topics**

- Multitenant, Performance, Backup & Restore

■ RAC Architecture



■ SPFILE



■ SPFILE

➤ Search Order

➤ OCR Configuration

- \$ORACLE_HOME/dbs/spfile<sid>.ora
- \$ORACLE_HOME/dbs/spfile.ora
- \$ORACLE_HOME/dbs/init<sid>.ora

```
srvctl config database -db DBNAME
```

```
Spfile: +DATA/DBNAME/PARAMETERFILE/spfile.293.968863499
```

```
SQL> create spfile='+DATA' from pfile='/tmp/init.ora';  
ASMCMD> ls -l +DATA/DBNAME/PARAMETERFILE/
```

```
srvctl modify database -db DBNAME -spfile '+DATA/DBNAME/.../spfile.294.983196717'
```

■ SPFILE

➤ Precedence

- sid.parameter=value
- *.parameter=value

```
SQL> alter session set open_cursors = 100 scope=both sid='*';  
SQL> alter session set open_cursors = 200 scope=both sid='inst1';  
*.open_cursors=100  
inst1.open_cursors=200
```

```
SQL> alter session reset open_cursors sid='inst1';  
*.open_cursors=100
```

```
SQL> select name, isinstance_modifiable  
       from v$parameter;  
  
db_files          FALSE  
open_cursors      TRUE
```

■ Parameter

➤ **Must** have **identical** settings

- control_files
- db_files

default: sid='*'

sid='inst' ignored

➤ **Should (can)** have **identical** settings

- log_archive_format
- undo_retention

default: sid='*'

sid='inst' valid

➤ **Must** have **unique** settings

- instance_name
- instance_number
- undo_tablespace
- thread

default: sid='current_instance'

sid='inst' valid

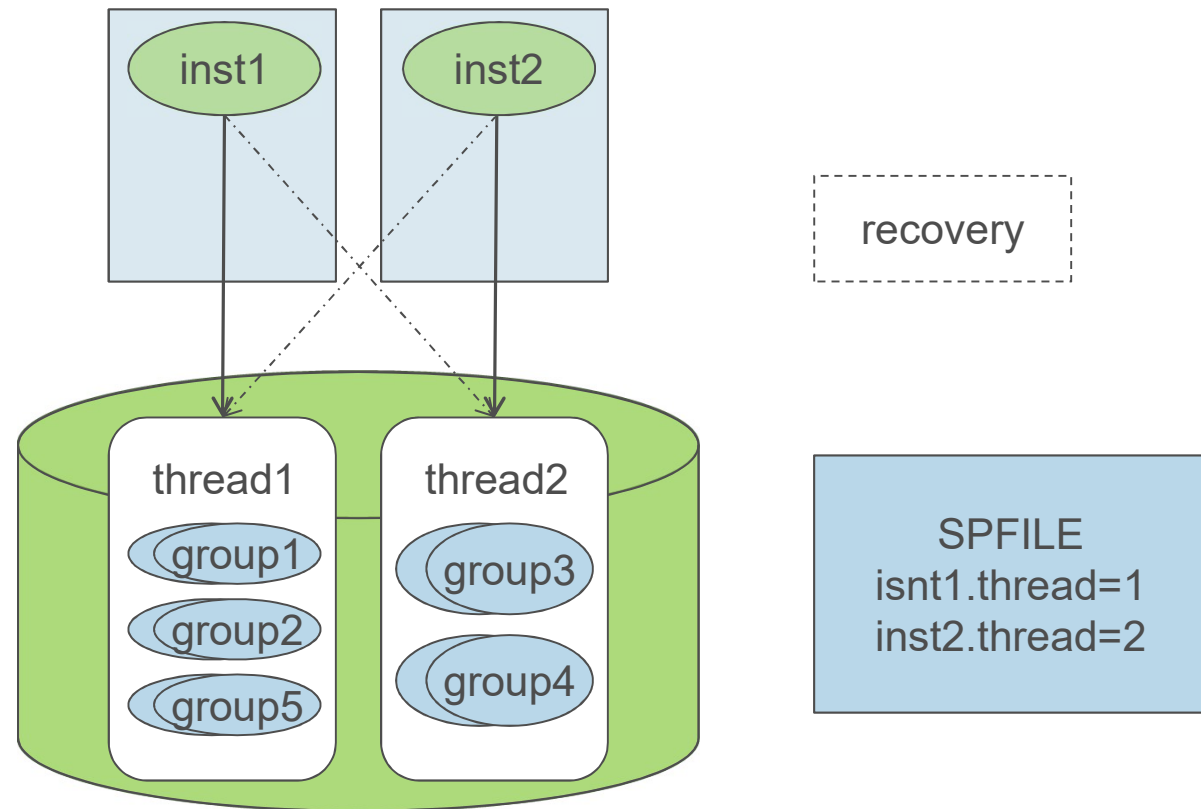
■ Global Views

- For almost every V\$ there is a corresponding GV\$ view
 - Contains an extra column named **INST_ID**

```
SQL> select inst_id, name, value from  
       gv$parameter  
       where name='sga_target';
```

INST_ID	NAME	VALUE
1	sga_target	12884901888
2	sga_target	12884901888

■ Online Redo Logs



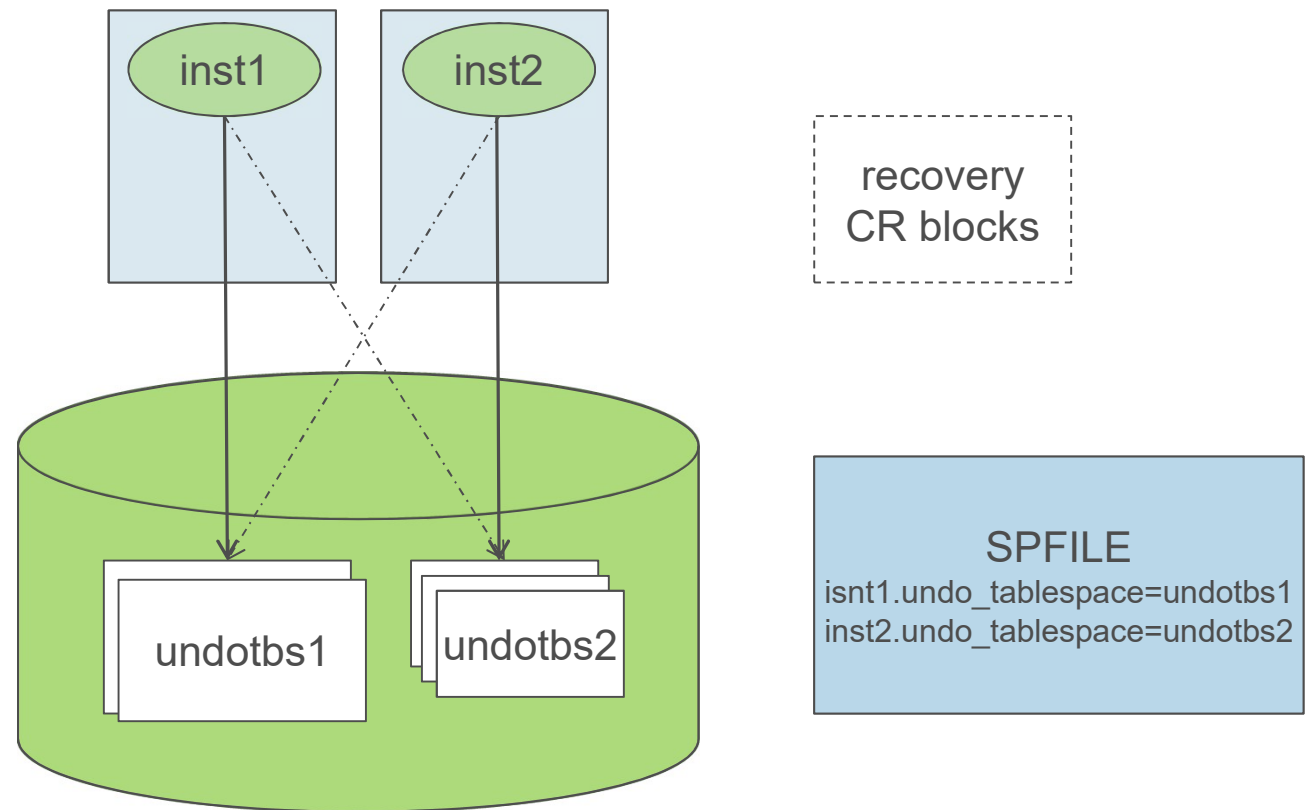
■ Online Redo Logs

- **Each instance has its own thread number**
- **Each instance must have at least two redo log groups**
 - All redo logs across all instances must be on shared storage
 - Number and size of redo logs can be different across instances
- **During instance recovery, redo log files can be accessed by any instance**

```
SQL> alter database add logfile thread 1 group 5 ('+DATA', '+FRA') size 1G;
```

- **thread** not specified → thread of current instance is used

■ UNDO Tablespace

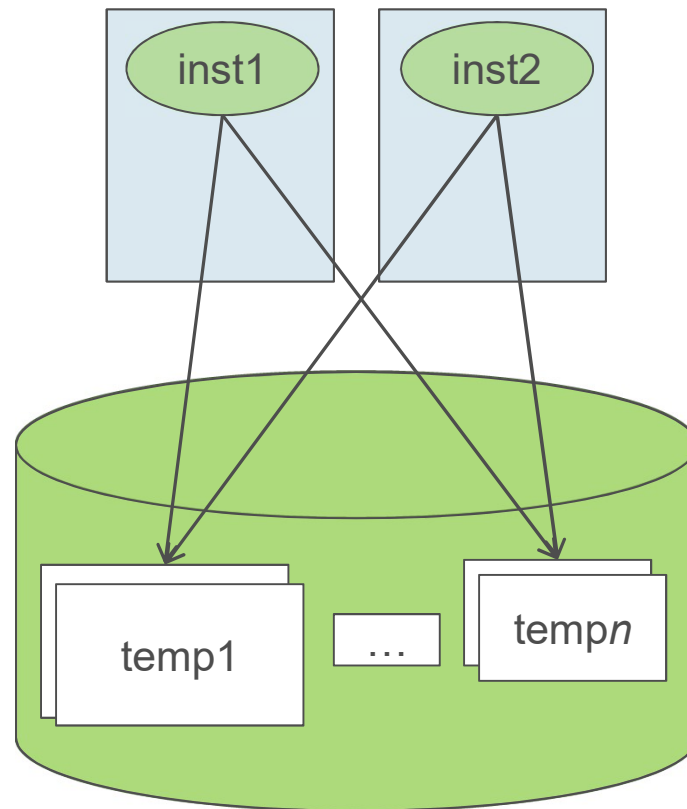


■ UNDO Tablespace

- Each instance has its own undo tablespace on shared storage
- All instances must operate in the same undo mode
- All instances have read access to undo tablespaces to guarantee read consistency
- During transaction recovery, undo tablespace can be modified by any instance

```
SQL> alter system set undo_tablespace=undotbs3 sid='inst1';
```

■ Temporary Tablespace

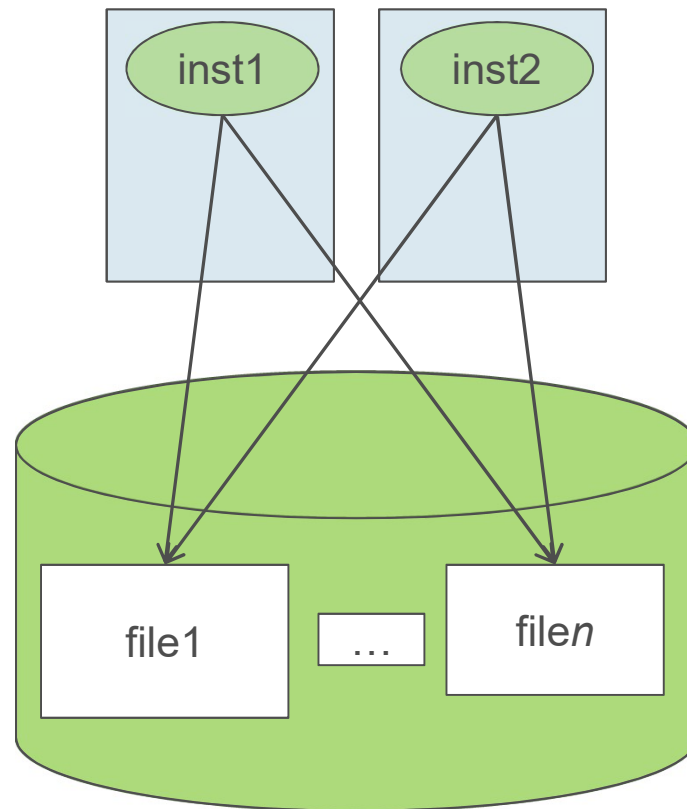


■ Temporary Tablespace

- **Temporary tablespaces are shared between all instances**
 - Instances try to cache extents equally from all files of a temporary tablespace
 - Spreading the workload among the temporary files

```
SQL> select inst_id, tablespace_name, file_id, extents_used  
       from gv$temp_extent_pool  
       order by 1,2,3;
```

■ Datafiles



■ Datafiles

```
SQL> create tablespace tbs1 datafile '+DATA' size 1G;
```

```
SQL> select file_id, file_name
       from dba_data_files
       where tablespace_name = 'TBS1';
```

FILE_ID	FILE_NAME
42	+DATA/DBNAME/DATAFILE/tbs1.298.7211088

```
SQL> alter database datafile '+DATA/.../tbs1.298.7211088' autoextend on maxsize 4G;
```

```
SQL> alter database datafile 42 autoextend on maxsize 4G;
```

■ Datafiles

```
SQL> alter tablespace tbs1 add datafile '+DATA' size 1G;
```

```
SQL> select file_id, file_name
       from dba_data_files
       where tablespace_name = 'TBS1';
```

FILE_ID	FILE_NAME
43	/u01/app/oracle/12.1.0.2/dbs/DATA

➤ **Add a tablespace or datafile or bring a tablespace or datafile online → all instances verify access to the file(s)**

- **ORA-01157**: cannot identify/lock data file 43
File 43 not verified due to error ORA-01157

```
SQL> alter system check datafiles;
```

-
- **RAC Architecture**
 - SPFILE, Views, Redo, Undo, Temp, Datafiles
 - **Connections**
 - Services, Sessions, Listener, Connections
 - **Administration**
 - Shutdown & Startup, Checkpoint, Log Switch
 - **Further Topics**
 - Multitenant, Performance, Backup & Restore

■ Services

- Use **srvctl** instead of **DBMS_SERVICE** package
 - Cluster resource will be created and OCR updated
- Cluster-managed services
 - Can be available on a subset of instances
 - Attributes can be modified (connection failover, load balancing)
 - Can be stopped or relocated

```
srvctl add service -db DBNAME -service SNAME -preferred inst1 -available inst2
```

```
srvctl config service -db DBNAME -service SNAME
```

```
srvctl modify service -db DBNAME -service SNAME -help
```

■ Sessions

```
SQL> select sid, serial#, inst_id
       from gv$session
       where user_name = 'USER1';
```

SID	SERIAL#	INST_ID
80	193	1

```
SQL> alter system kill session '80,193';
```

connected to inst1

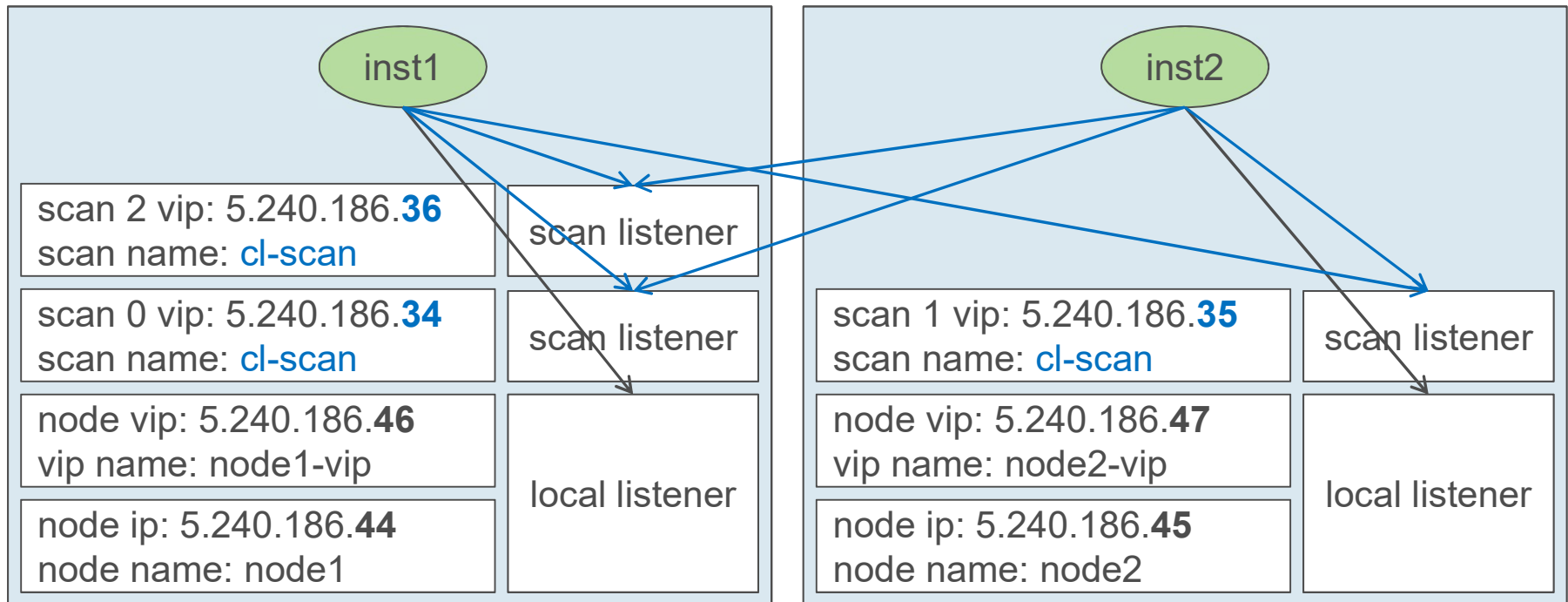
```
SQL> alter system kill session '80,193,@1';
```

connected to any instance

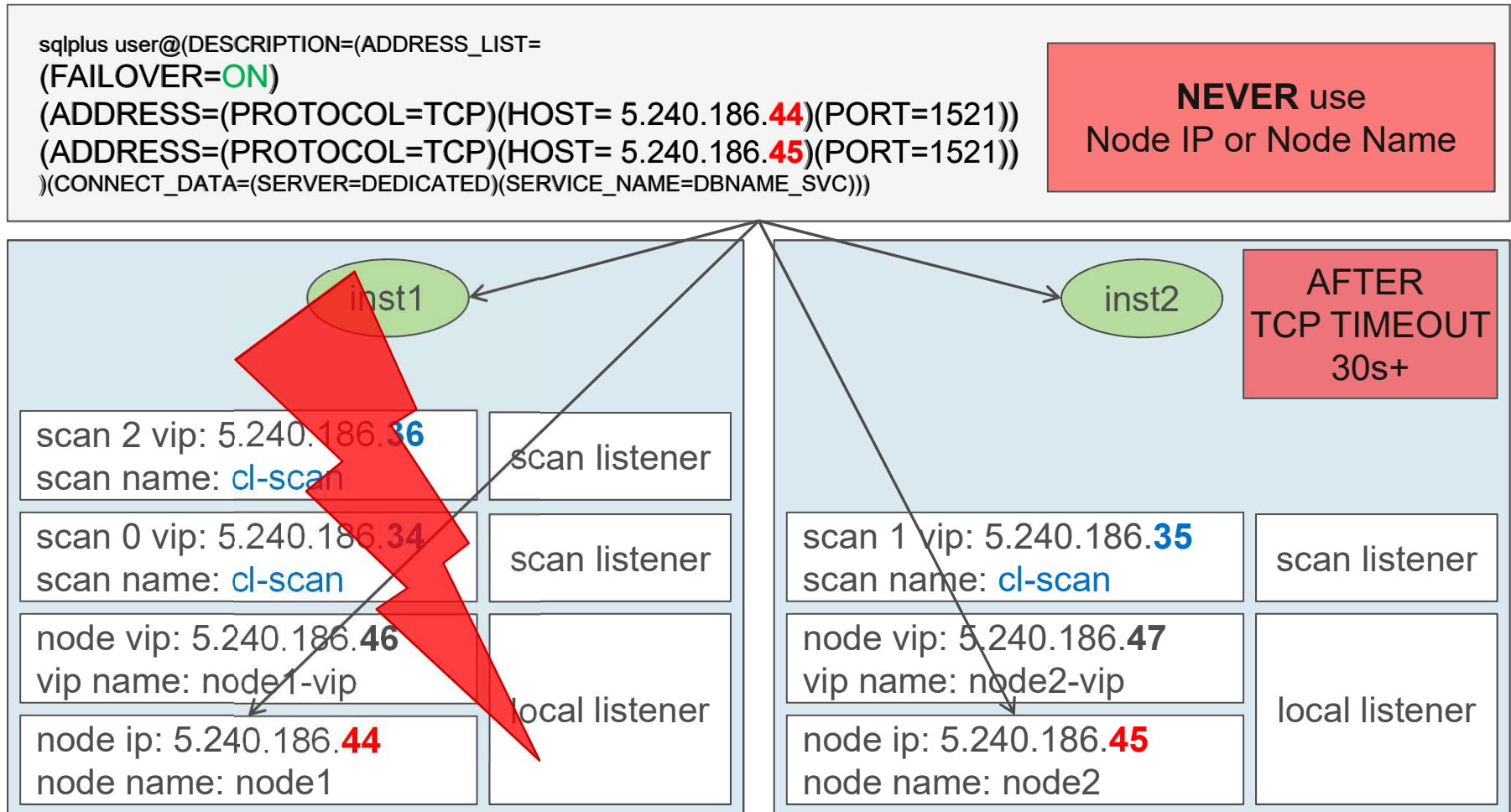
■ Listener

```
isnt1.local_listener=(ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST = 5.240.186.46)(PORT = 1521)))  
inst2.local_listener=(ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST = 5.240.186.47)(PORT = 1521)))
```

```
*.remote_listener=(ADDRESS_LIST = (ADDRESS=(PROTOCOL=TCPS)(HOST=5.240.186.34)(PORT=1523))  
                                   (ADDRESS=(PROTOCOL=TCPS)(HOST=5.240.186.35)(PORT=1523))  
                                   (ADDRESS=(PROTOCOL=TCPS)(HOST=5.240.186.36)(PORT=1523)))
```



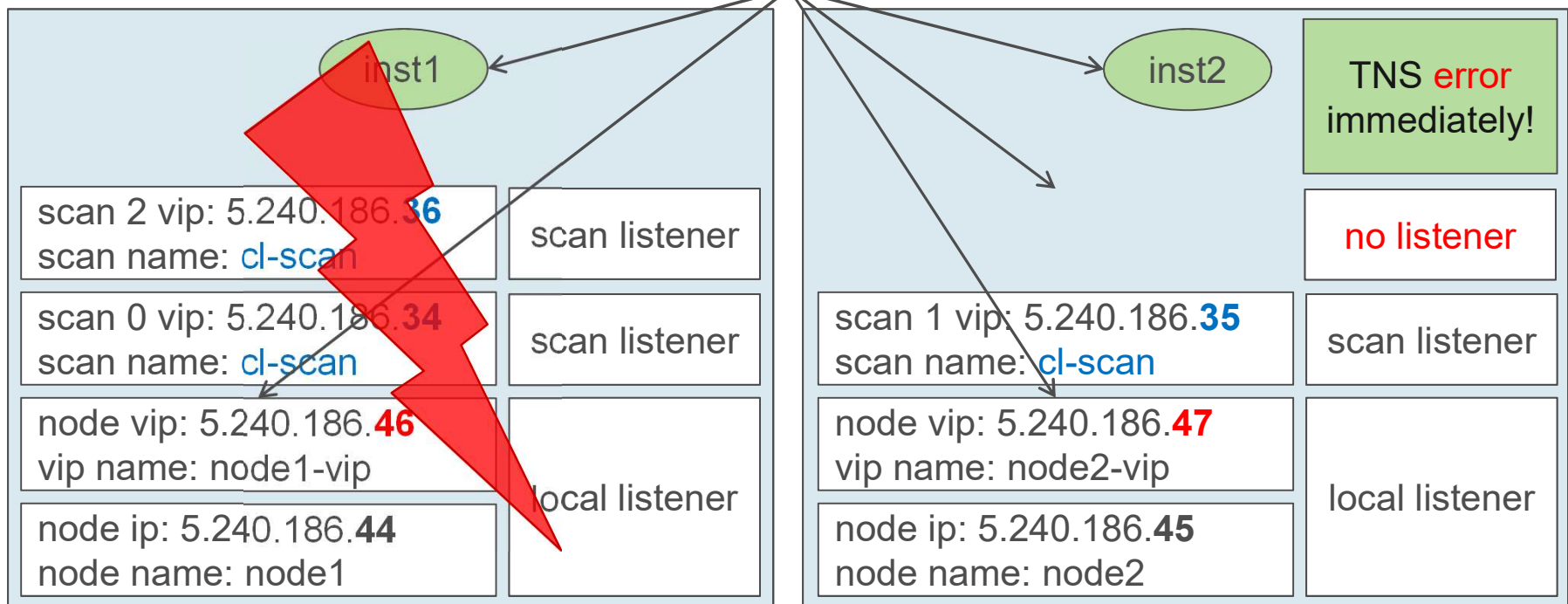
■ Connections (node ip / node name)



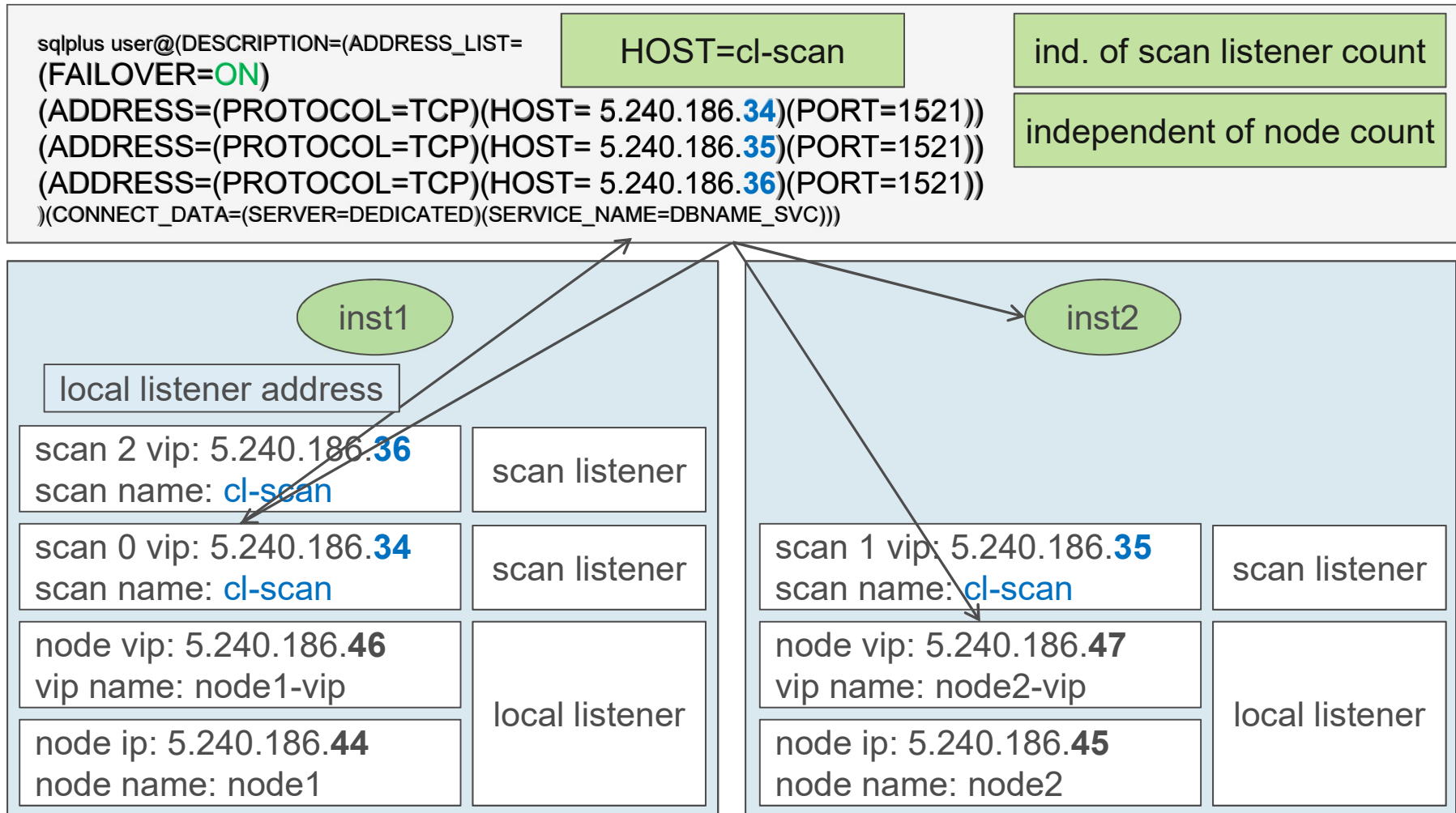
■ Connections (**node vip** / vip name)

```
sqlplus user@(DESCRIPTION=(ADDRESS_LIST=
(FAILOVER=ON)
(ADDRESS=(PROTOCOL=TCP)(HOST= 5.240.186.46)(PORT=1521))
(ADDRESS=(PROTOCOL=TCP)(HOST= 5.240.186.47)(PORT=1521))
)(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=DBNAME_SVC)))
```

try second
address immediately!



■ Connections (scan vip / SCAN)



■ Connections

scan 2 vip: 5.240.186.**36**
scan name: **cl-scan**

scan 1 vip: 5.240.186.**35**
scan name: **cl-scan**

scan 0 vip: 5.240.186.**34**
scan name: **cl-scan**

node vip: 5.240.186.**46**
vip name: node1-vip

node ip: 5.240.186.**44**
node name: node1

```
srvctl config scan  
nslookup cl-scan
```

```
srvctl config vip -node node1
```

```
nslookup node1
```

-
- **RAC Architecture**
 - SPFILE, Views, Redo, Undo, Temp, Datafiles
 - **Connections**
 - Services, Sessions, Listener, Connections
 - **Administration**
 - Shutdown & Startup, Checkpoint, Log Switch
 - **Further Topics**
 - Multitenant, Performance, Backup & Restore

■ Shutdown & Startup

```
srvctl stop database -db DBNAME -stopoption immediate  
srvctl start database -db DBNAME -startoption open
```

```
srvctl stop instance -db DBNAME -instance inst1 -stopoption immediate  
srvctl start instance -db DBNAME -instance inst1 -startoption open
```

```
srvctl stop listener [-listener LISTENER]  
srvctl start listener [-listener LISTENER]
```

GI owner / one per node

```
srvctl stop scan_listener -scan_number 2  
srvctl start scan_listener -scan_number 2
```

GI owner / 1-3 per cluster

```
srvctl stop service -db DBNAME -service DBNAME_SVC -instance inst1  
srvctl start service -db DBNAME -service DBNAME_SVC -instance inst2
```

■ Shutdown & Startup

SQL> shutdown immediate;

current instance

Can **NOT** stop or start all instances with a single SQL*Plus command

SQL> shutdown transactional;

waits for transactions on all instances

SQL> shutdown transactional **local**;

waits for transactions on current instance

■ Checkpoint, Log Switch

SQL> alter system checkpoint [global];

all instances

SQL> alter system checkpoint **local**;

current instance

SQL> alter system switch logfile;

current instance

SQL> alter system archive log current;

all instance


current log group

SQL> alter system archive log **instance inst1** current;

specific instance

-
- **RAC Architecture**
 - SPFILE, Views, Redo, Undo, Temp, Datafiles
 - **Connections**
 - Services, Sessions, Listener, Connections
 - **Administration**
 - Shutdown & Startup, Checkpoint, Log Switch
 - **Further Topics**
 - Multitenant, Performance, Backup & Restore

■ Multitenant (PDBs)

- Each PDB can be available on all or on subset of instances

```
SQL> alter pluggable database pdb1 close;
```

current instance

```
SQL> alter pluggable database pdb1 close instances=all;
```

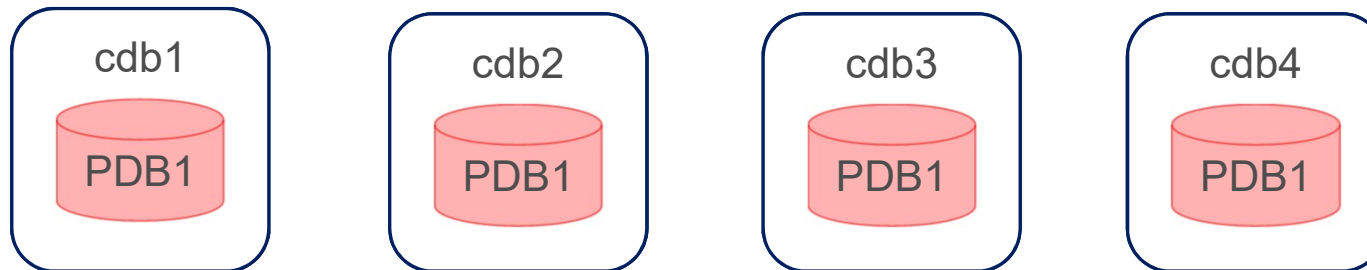
```
SQL> alter pluggable database pdb1 close instances=('inst1','inst3');
```

- Close PDB in current instance and open it in another instance

```
SQL> alter pluggable database pdb1 close relocate to cdb2;
```

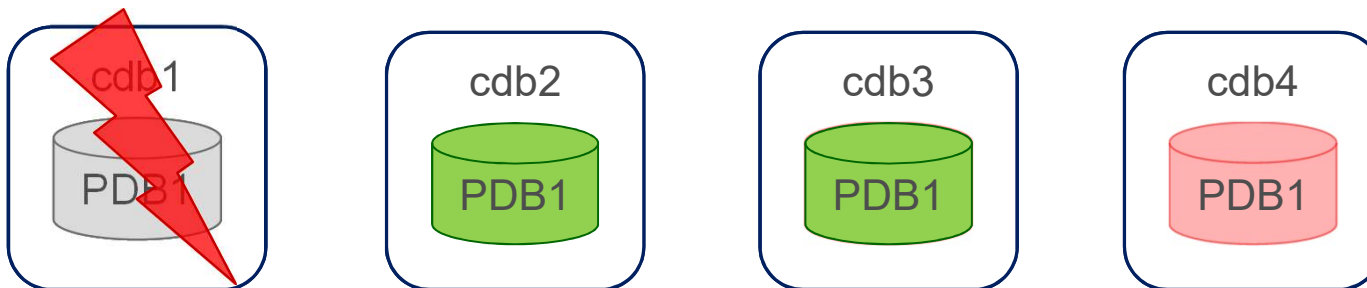
■ Multitenant (Services)

- Coordinate start and stop of PDBs using dynamic services



```
srvctl add service -db CDBNAME -service PDB1_SVC -pdb PDB1  
-preferred cdb1,cdb2 -available cdb3,cdb4
```

```
srvctl start service -db CDBNAME -service PDB1_SVC
```



■ Multitenant (Services)

- Start service → PDB opens on preferred instances

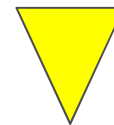
No need for AFTER DATABASE STARTUP trigger!!!

- Stop service → PDB does **NOT** close



- Close PDB → Service stops on nodes where PDB was closed

- Open PDB → Service does **NOT** start



- Must be started manually: `srvctl start service ...`

■ Performance (AWR/ASH)

- **AWR snapshot captures data from all active instances**
 - Data is stored for each individual instance in the same table
 - Same snapshot id, but different instance ids

SQL> @?/rdbms/admin/awrrpt;

current instance

SQL> @?/rdbms/admin/awrrpti;

specific instance

SQL> @?/rdbms/admin/ashrpt;

current instance

SQL> @?/rdbms/admin/ashrpti;

specific instance

■ Backup & Restore

- Preferred to have archived redo logs on shared storage
- Multiple node can restore archived redo logs in parallel
 - Only one node applies the logs during recovery

```
RMAN> configure channel ch1 device type disk;
```

current instance

```
RMAN> configure channel ch1 device type disk '@inst2';
```

specific instance

```
RMAN> run { restore archivelog from logseq 65 until logseq 70 thread 2; }
```

thread not specified → **RMAN-20242**: specification does not match any archived log

■ Backup & Restore

➤ PDB PITR

```
RMAN>  
alter pluggable database pdb1 close immediate instances=all;  
set until time ...  
restore pluggable database pdb1;  
recover pluggable database pdb1;  
alter pluggable database pdb1 open resetlogs;  
alter pluggable database pdb1 open instances=all;
```

current instance

Contact

THANK YOU