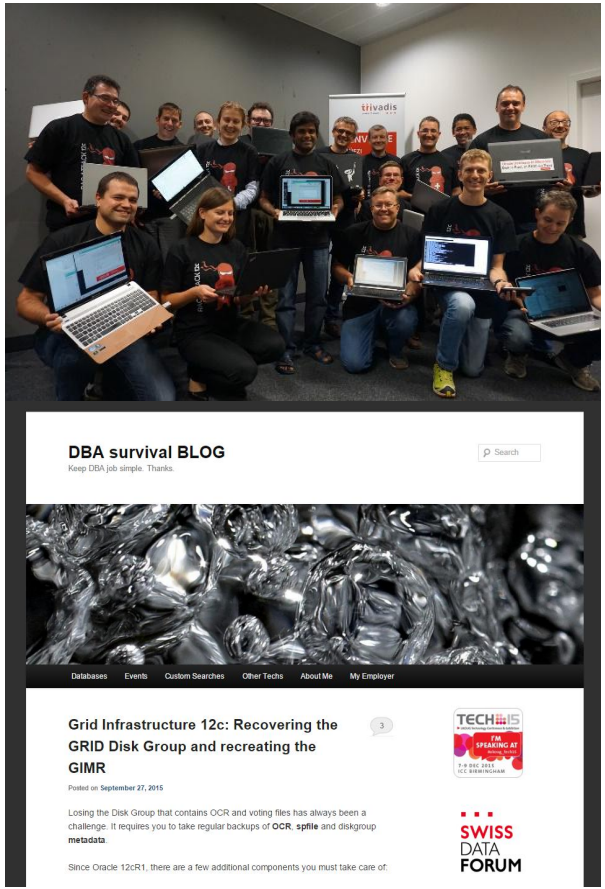
Get the Most out of Oracle Data Guard!

High Five POUG
#POUG2017

Ludovico Caldara
Oracle ACE Director
Senior Consultant

About Ludovico Caldara



- 18 Years DBA (Not Only Oracle)
  - I do it everywhere (even Windows)
- RAC ATTACK Ninja & co-writer
- RACSIG Real Application Clusters Special Interest Group President, SOUG & ITOUG Board
- OCP (11g, 12c, MySQL) & OCE
- Italian living in Switzerland

http://www.ludovicocaldara.net

@ludodba ludodba

ludovicocaldara

ORACLE ACE Director

trivadis makes IT easier.

Oracle
Data Guard?

Why is Oracle Data Guard still relevant?

- The best high availability solution in the Oracle ecosystem

  - Synchronous (or not)
  - One-to-one copy (or one-to-many)
  - No single points of failure
  - Failover is (almost) transparent to the applications (if well configured)
- Rock solid!
- Included in Oracle Database Enterprise Edition

Hey, it’s 12c Release 2!

- Multiple Observers for Fast-Start Failover configurations
- Observer in Background mode (needs wallet)
- Recover of Nologging Operations (`recover database nonlogged block;`)
- Multiple fast_start failover targets
- Automated passwordfile copy
- Enhanced Broker and dgmgrl
- FastSync redo transport

Hey, it's 12c Release 2!

- Multiple Observers for Fast-Start Failover configurations
- Observer in Background mode (needs wallet)
- Recover of Nologging Operations (`recover database nonlogged block;`)
- Multiple fast_start failover targets
- Automated passwordfile copy
- Enhanced Broker and dgmgrl
- FastSync redo transport That was 12cR1!

What about Active Data Guard new features?

What about Active Data Guard new features?

What about Active Data Guard new features?

08:30 am - 09:15 am

Oracle Active Data Guard 12cR2. Is it the best option?

Ludovico Caldara

Technology Enablers

The «MAIN» features included in Data Guard

- Client Failover (TAF)
- Redo Apply
- SQL Apply
- Snapshot Standby
- Rolling Upgrades (possible, but without RDBMS_ROLLING)
- Transaction Guard (necessary for Application Continuity but not licensed as option)

The «MAIN» features included in Data Guard

- Client Failover (TAF)
- Redo Apply
- SQL Apply
- Snapshot Standby
- Rolling Upgrades (possible, but without RDBMS_ROLLING)
- Transaction Guard (necessary for Application Continuity but not licensed as option)

Our three main topics for this presentation

- Client Failover (TAF)
- Snapshot Standby
- Standby consistency on recovery cancel

Our three main topics for this presentation

- Client Failover (TAF)
- Snapshot Standby
- Standby consistency on recovery cancel

Our three main topics for this presentation

- Client Failover (TAF)
- Snapshot Standby
- Standby consistency on recovery cancel

Our three topics focuses for this presentation

- Client Failover (TAF)
- Snapshot Standby
- Standby consistency on recovery cancel

Client Failover

Client Failover is a critical topic!

- https://www.slideshare.net/ludovicocaldara/oracle-client-failover-under-the-hood
  - OS Connect Timeouts/ARP Cache
  - OS Re-Connect Timeouts
  - Virtual IP Addresses
  - TCP Keepalive
  - Database Services
  - DB Connect Timeouts
  - DB Re-Connect Timeouts
  - Transparent Application Failover
  - Fast Application Notification / Fast Connection Failover
  - Application Continuity

Database Services

- Database services can be created with:
  - *srvctl* (*Grid Infrastructure*), gdsctl (Global Data Services).
  - *dbms_service.create_service*() PL/SQL procedure (TRIGGER AFTER STARTUP!)
- Different high availability and workload management attributes can be defined

```
srvctl add service
  -db           <db_unique_name>
  -service      <service>
  -preferred    "<preferred_list>"
  -available    "<available_list>"
  -serverpool   <pool_name>
  -cardinality  [UNIFORM | SINGLETON]
  -tafpolicy    [NONE | BASIC | PRECONNECT]
  -role         [PRIMARY, PHYSICAL_STANDBY, LOGICAL_STANDBY, SNAPSHOT_STANDBY]
  -clbgoal      [SHORT | LONG]
  -rlbgoal      [SERVICE_TIME | THROUGHPUT | NONE]
  ...
```

Not available with Oracle Restart

Role-Based Services

- Example role-based services with Grid Infrastructure.

```
srvctl add service -db sour_poug -service sour_rw.trivadis.com \
        -role PRIMARY
srvctl add service -db sour_poug -service sour_ro.trivadis.com \
        -role PHYSICAL_STANDBY
srvctl add service -db sour_poug -service sour_snap.trivadis.com \
        -role SNAPSHOT_STANDBY
```

- Services are started, only if database and service role match.

```
SvcAgent::start 680 query_db_role
SvcAgent::start 710 not starting service sour_rw Role mismatch -
Service role:PRIMARY, current DB role:PHYSICAL_STANDBY
```

Transparent Application Failover – Server Side Example

- Example server side TAF BASIC method configuration.

```
srvctl add service
  -db               sour_SITE1
  -service          sour_RW
  -tafpolicy        BASIC
  -failovertype     SELECT
  -failoverdelay    1
  -failoverretry    180
```

```
BEGIN
  DBMS_SERVICE.CREATE_SERVICE (
     service_name      => 'sour.TRIVADIS.COM',
     network_name      => 'sour.TRIVADIS.COM',
     failover_method   => 'BASIC',
     failover_type     => 'SELECT',
     failover_retries  => 180,
     failover_delay    => 3);
END;
/
```

Transparent Application Failover – Client Side Example

- Example client side TAF BASIC method configuration.

```
sour.trivadis.com =
  (DESCRIPTION =
    (FAILOVER=ON) (LOAD_BALANCE=OFF)
    (CONNECT_TIMEOUT=5)(RETRY_COUNT=3)(RETRY_DELAY=1)(TRANSPORT_CONNECT_TIMEOUT=3)
    (ADDRESS_LIST= (ADDRESS=(PROTOCOL=TCP)(HOST=ludo01.trivadis.com)(PORT=1521))
                   (ADDRESS=(PROTOCOL=TCP)(HOST=vico01.trivadis.com)(PORT=1521)))
    (CONNECT_DATA =
       (SERVICE_NAME = sour.trivadis.com)
       (FAILOVER_MODE =
            (TYPE = SESSION)
            (METHOD = BASIC)
            (RETRIES = 180)
            (DELAY = 1)
        )
     )
   )
```

Transparent Application Failover – Client Side Example

- Example client side TAF BASIC method configuration.

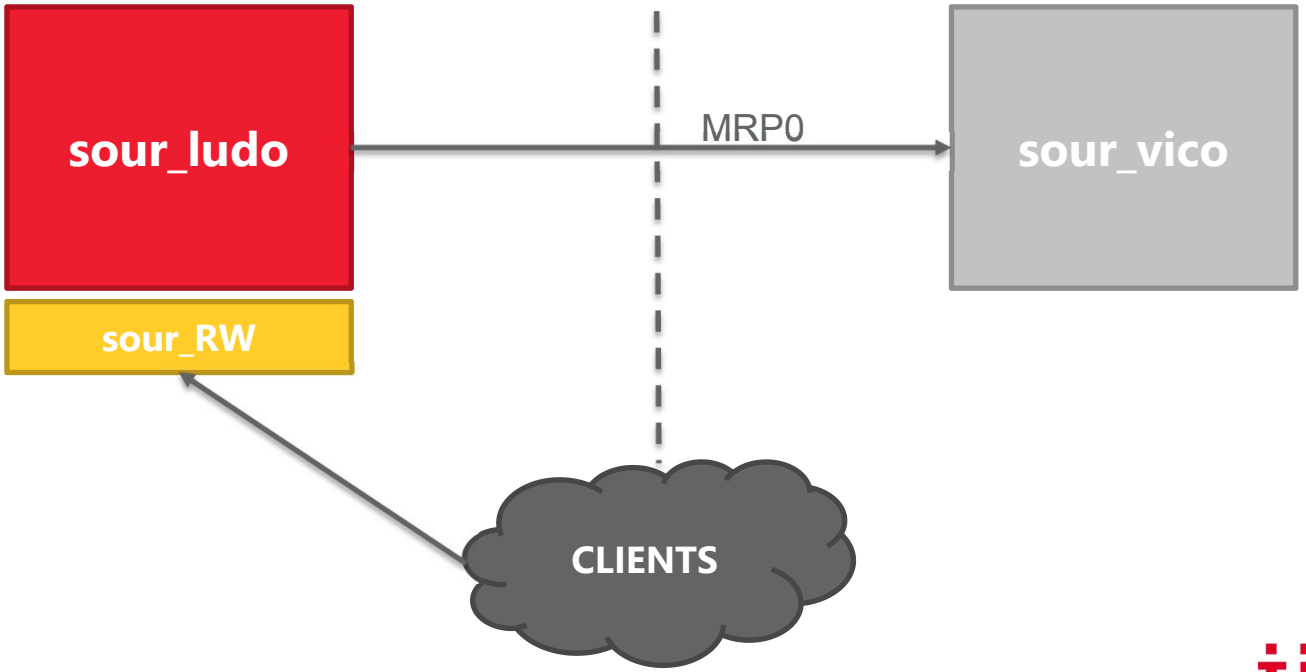
```
sour.trivadis.com =
  (DESCRIPTION =
    (FAILOVER=ON) (LOAD_BALANCE=OFF)
    (CONNECT_TIMEOUT=5)(RETRY_COUNT=3)(RETRY_DELAY=1)(TRANSPORT_CONNECT_TIMEOUT=3)
    (ADDRESS_LIST= (ADDRESS=(PROTOCOL=TCP)(HOST=ludo01.trivadis.com)(PORT=1521))
                   (ADDRESS=(PROTOCOL=TCP)(HOST=vico01.trivadis.com)(PORT=1521)))
    (CONNECT_DATA =
       (SERVICE_NAME = sour.trivadis.com)
       (FAILOVER_MODE =
            (TYPE = SESSION)
            (METHOD = BASIC)
            (RETRIES = 180)
            (DELAY = 1)
        )
     )
   )
```

TAF is for OCI Drivers only
Thin Driver: do not use TAF

Use Case: Classic Data Guard Failover / Switchover

```
srvctl add service -db sour_vico
-service sour_RW
-role PRIMARY
```

```
(ADDRESS=(HOST=ludo01)(PORT=1521))
(ADDRESS=(HOST=vico01)(PORT=1521))
```



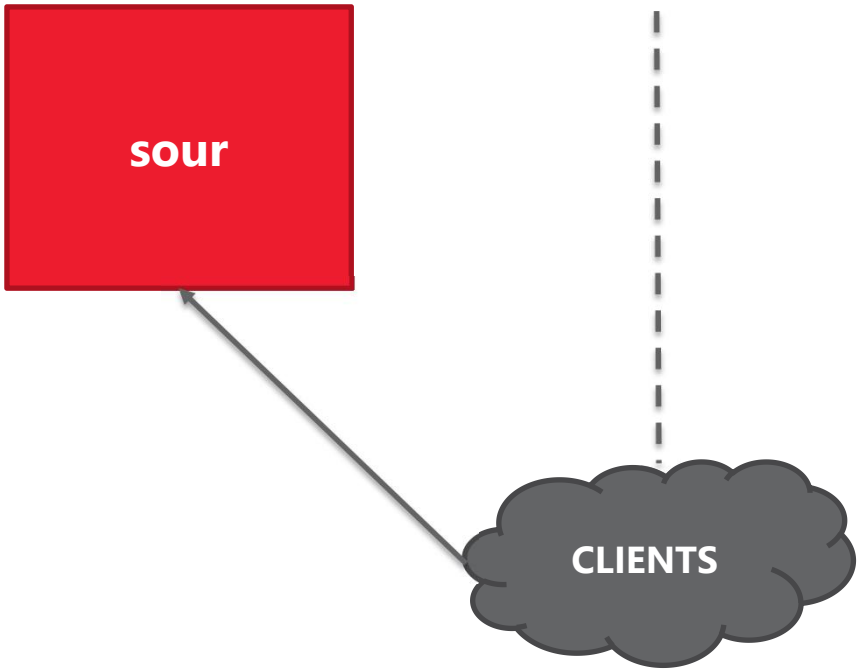
Use Case: Classic Data Guard Failover / Switchover

```
srvctl add service -db sour_vico
-service sour_RW
-role PRIMARY
```

```
(ADDRESS=(HOST=ludo01)(PORT=1521))
(ADDRESS=(HOST=vico01)(PORT=1521))
```

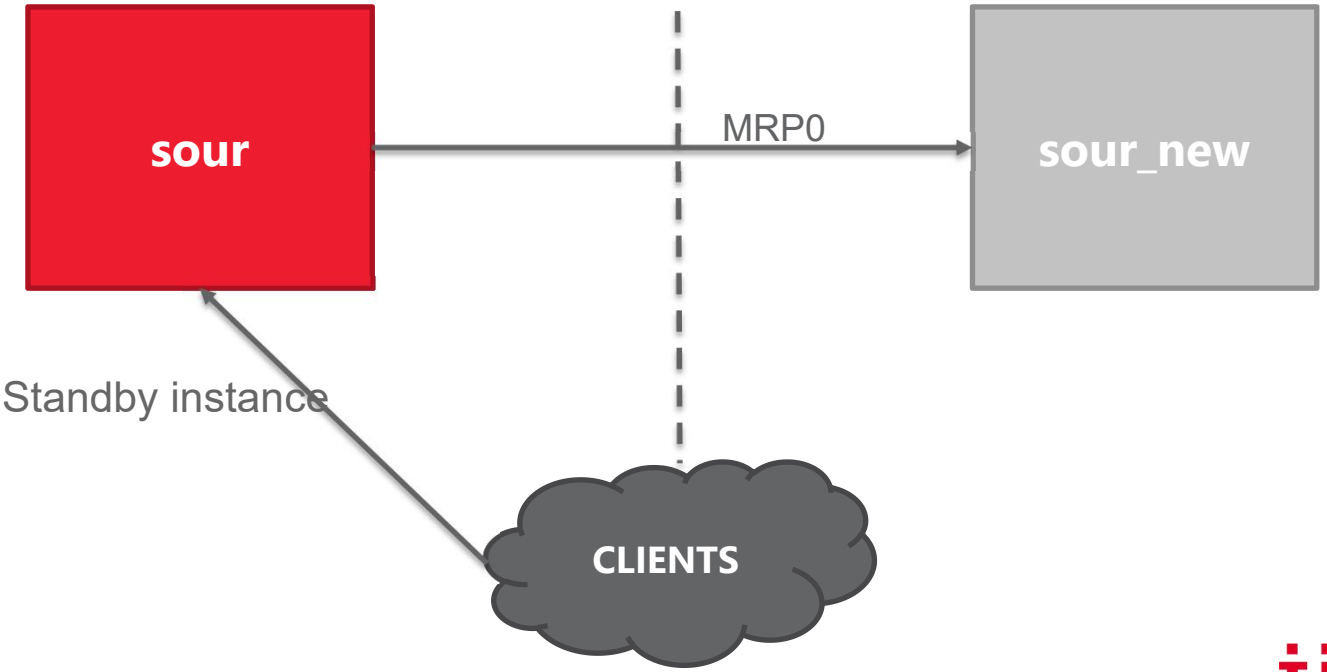
Use Case: Migrate Standalone on another server

```
(ADDRESS=(HOST=ludo01)(PORT=1521))
```



Use Case: Migrate Standalone on another server

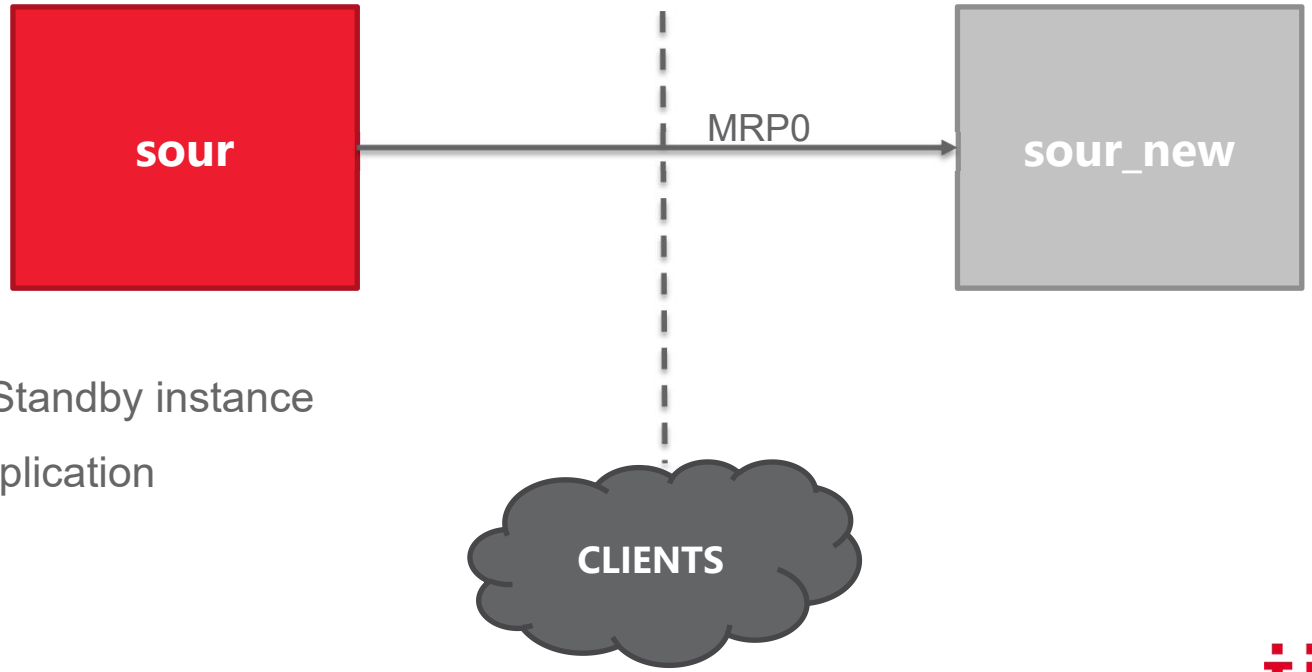
```
(ADDRESS=(HOST=ludo01)(PORT=1521))
```



- Set up Standby instance

Use Case: Migrate Standalone on another server

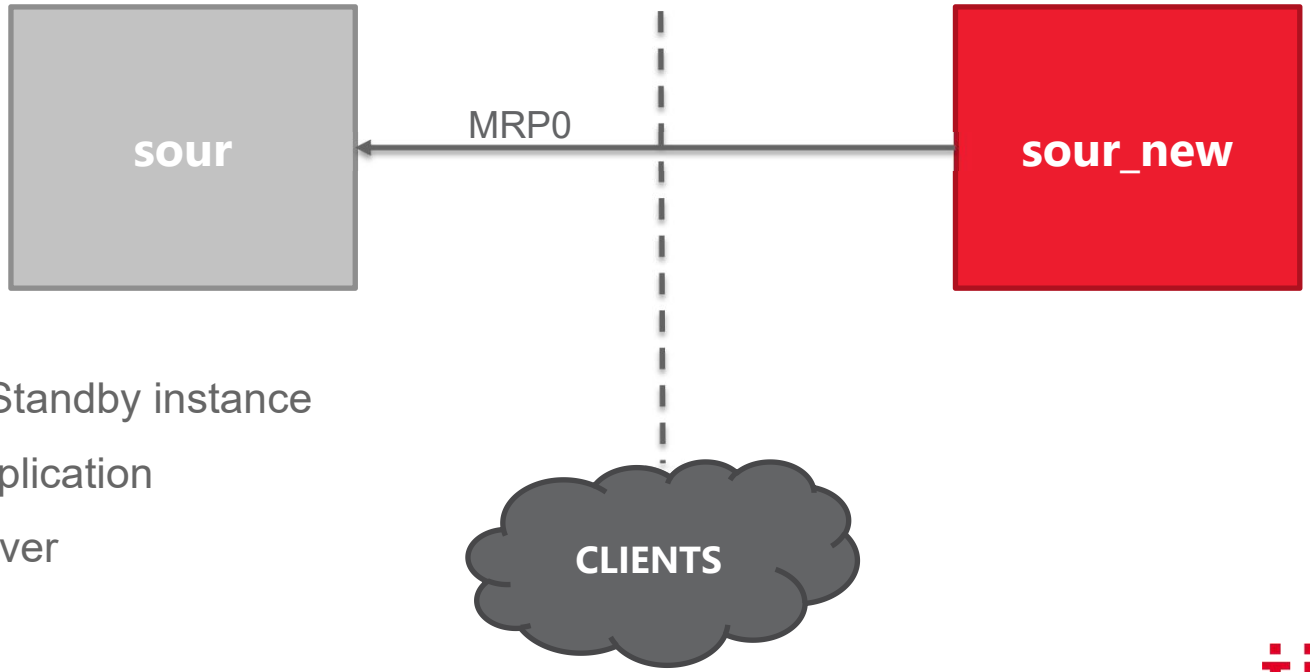
```
(ADDRESS=(HOST=ludo01)(PORT=1521))
```



- Set up Standby instance
- Stop application

Use Case: Migrate Standalone on another server

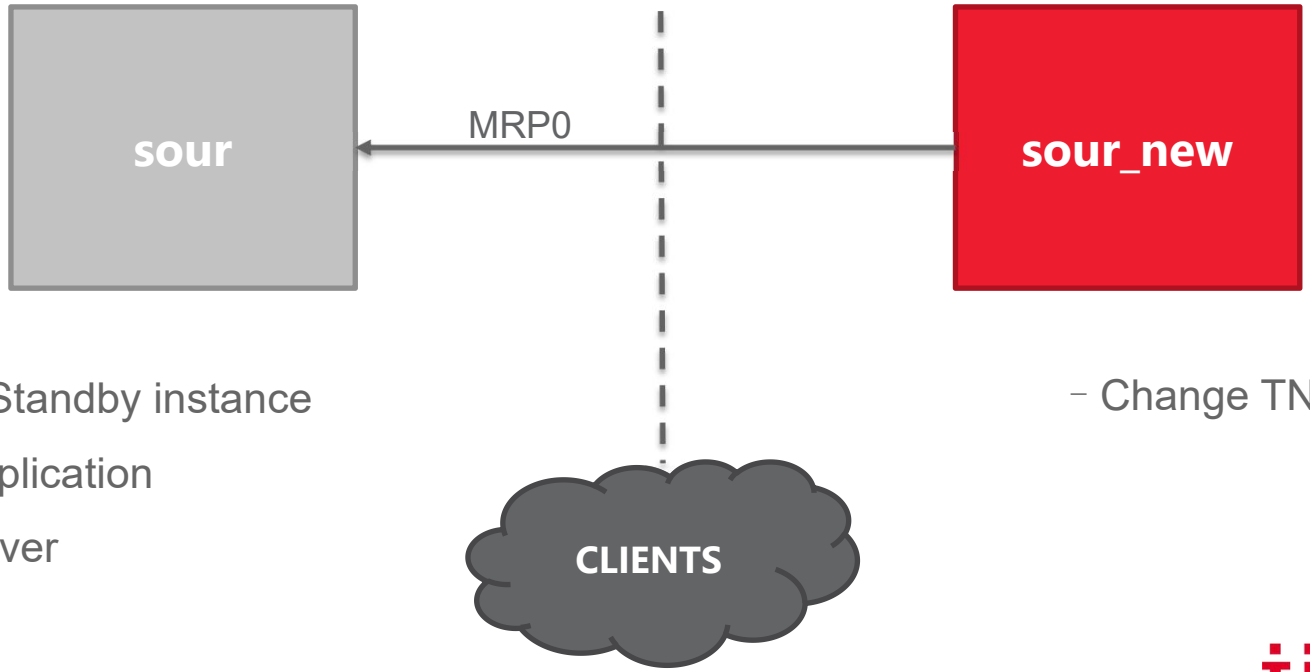
```
(ADDRESS=(HOST=ludo01)(PORT=1521))
```



- Set up Standby instance
- Stop application
- Switchover

Use Case: Migrate Standalone on another server

```
(ADDRESS=(HOST=~~ludo01~~vico01)(PORT=1521))
```

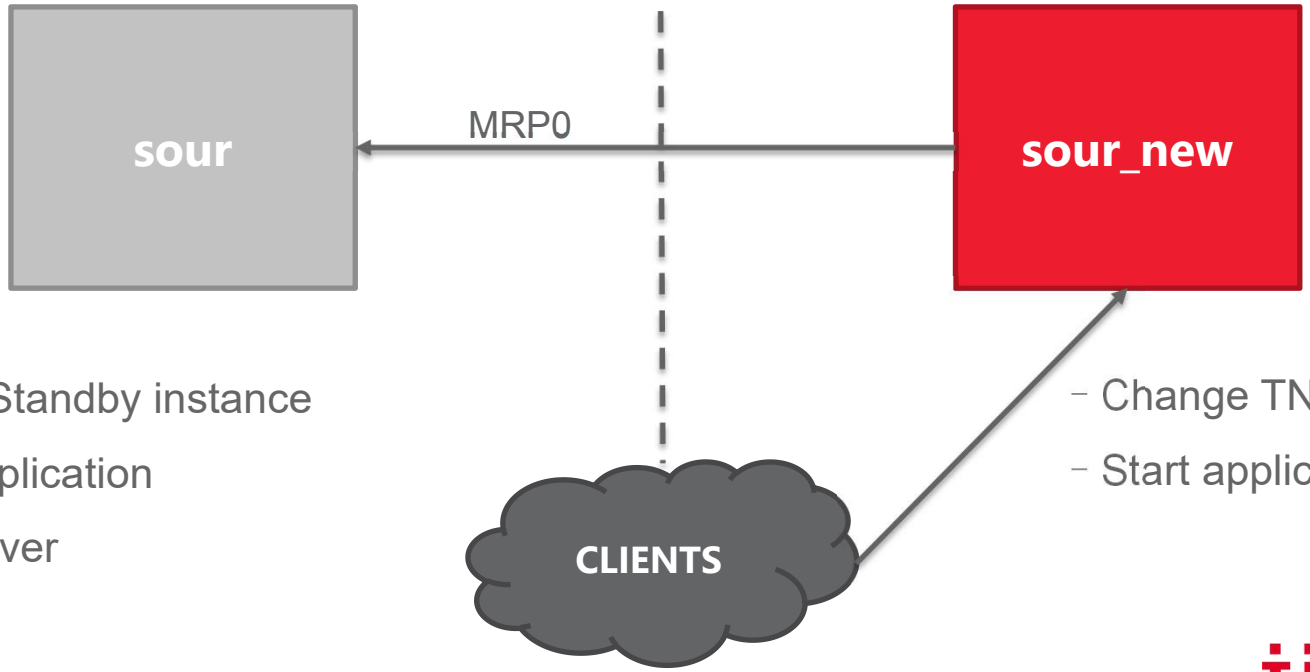


- Set up Standby instance
- Stop application
- Switchover

- Change TNS definition

Use Case: Migrate Standalone on another server

```
(ADDRESS=(HOST=~~ludo01~~vico01)(PORT=1521))
```

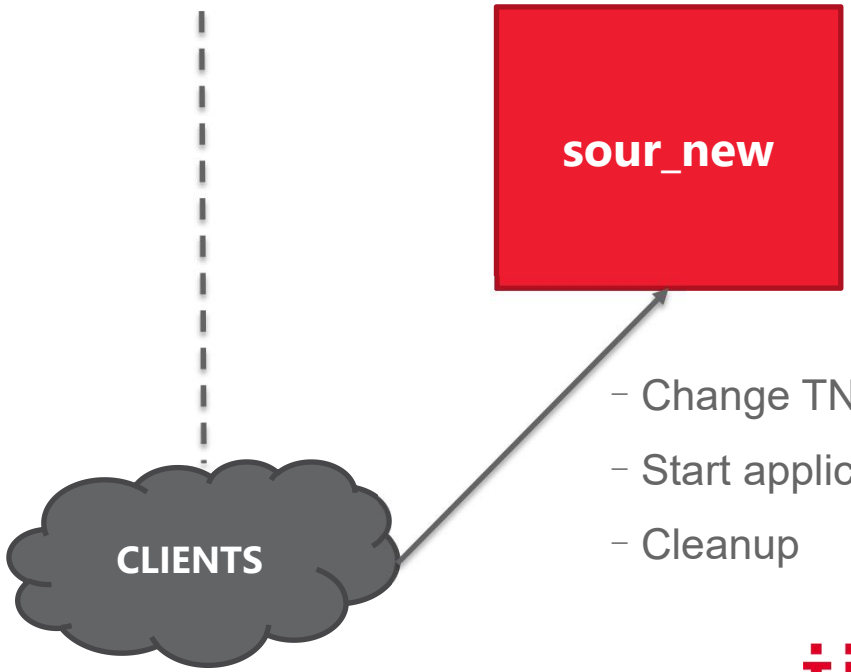


- Set up Standby instance
- Stop application
- Switchover

- Change TNS definition
- Start application

Use Case: Migrate Standalone on another server

```
(ADDRESS=(HOST=vico01)(PORT=1521))
```



- Set up Standby instance
- Stop application
- Switchover

- Change TNS definition
- Start application
- Cleanup

Use Case: Migrate Standalone on another server

- Problems
  - Requires application downtime (even if short)
  - Messy DB_UNIQUE_NAME naming convention

Use Case: Migrate Standalone on another server

- Problems
  - Requires application downtime (even if short)
  - Messy DB_UNIQUE_NAME naming convention

? Better solutions?

Use Case: Migrate Standalone on another server

- Problems
  - Requires application downtime (even if short)
  - Messy DB_UNIQUE_NAME naming convention

? Better solutions?

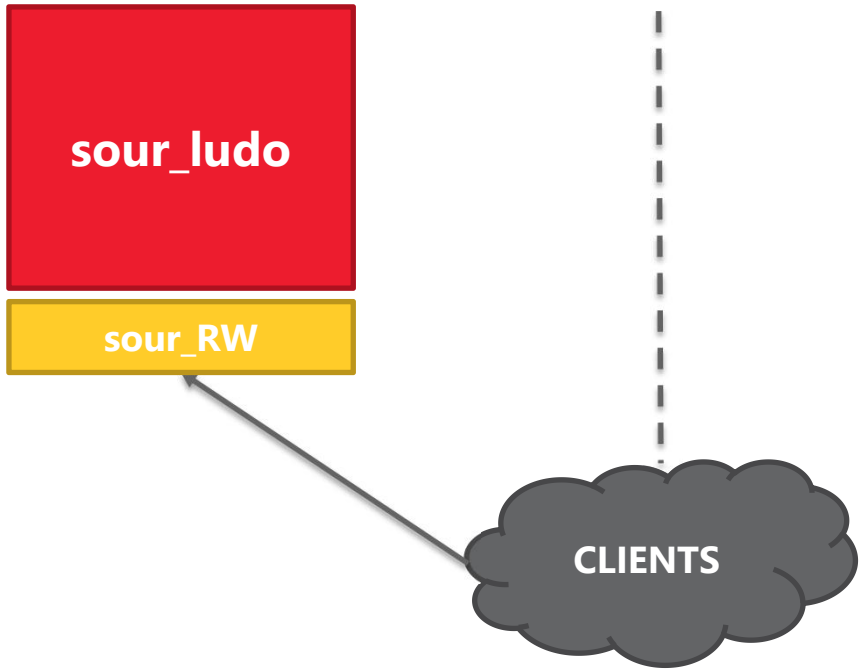
Hint: this is valid

```
(ADDRESS=(HOST=ludo01)(PORT=1521))
(ADDRESS=(HOST=ludo01)(PORT=1521))
```

Use Case: Smart Standalone Configuration

```
sour-s1 IN CNAME ludo01
sour-s2 IN CNAME ludo01
```

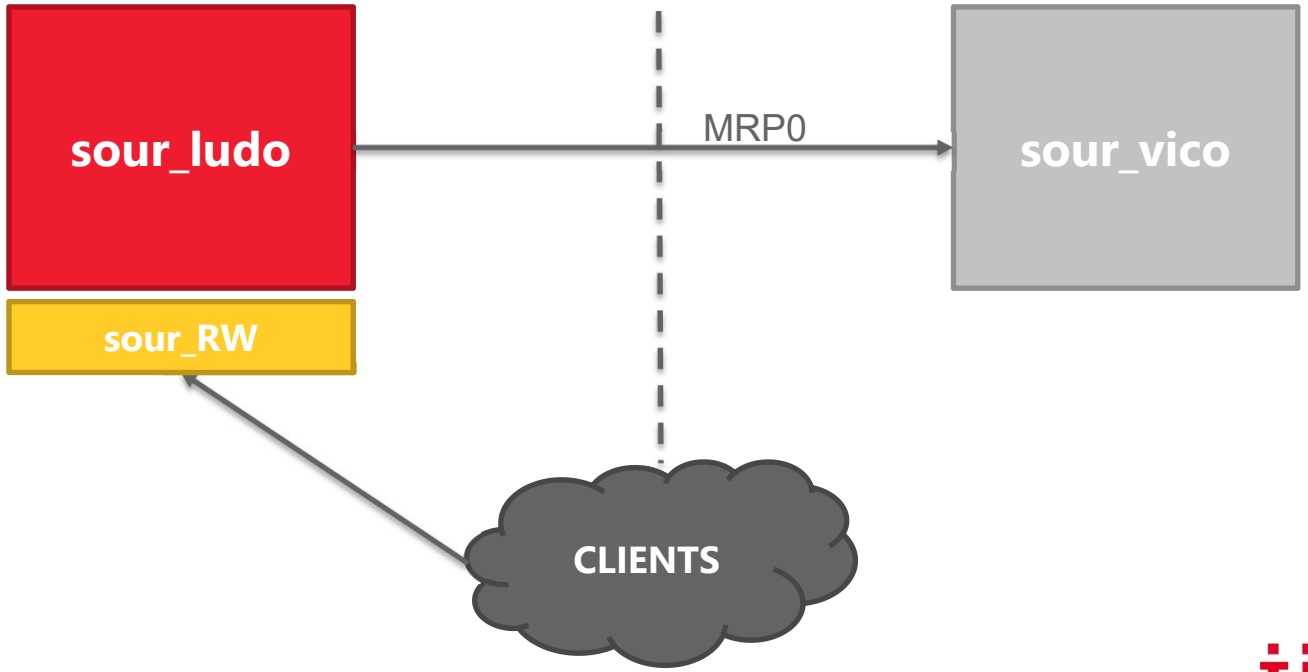
```
(ADDRESS=(HOST=sour-s1)(PORT=1521))
(ADDRESS=(HOST=sour-s2)(PORT=1521))
```



Use Case: Smart Standalone Configuration

```
sour-s1 IN CNAME ludo01
sour-s2 IN CNAME ludo01
```

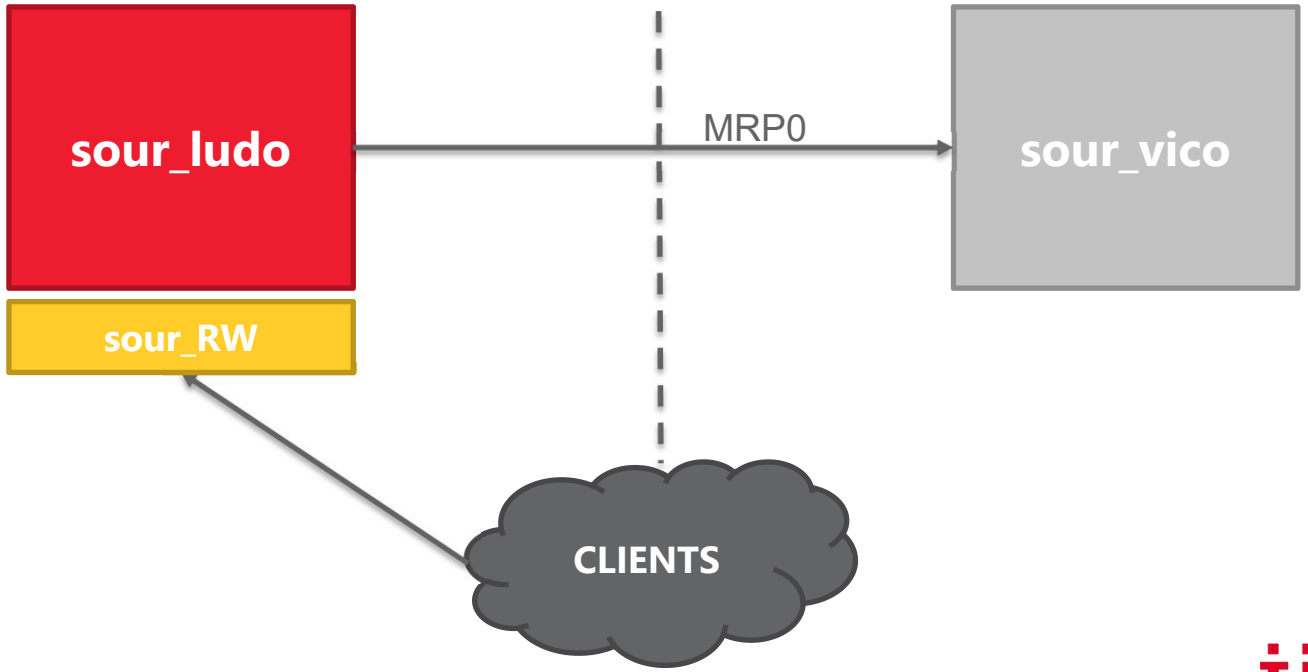
```
(ADDRESS=(HOST=sour-s1)(PORT=1521))
(ADDRESS=(HOST=sour-s2)(PORT=1521))
```



Use Case: Smart Standalone Configuration

```
sour-s1 IN CNAME ludo01
sour-s2 IN CNAME vico01
```

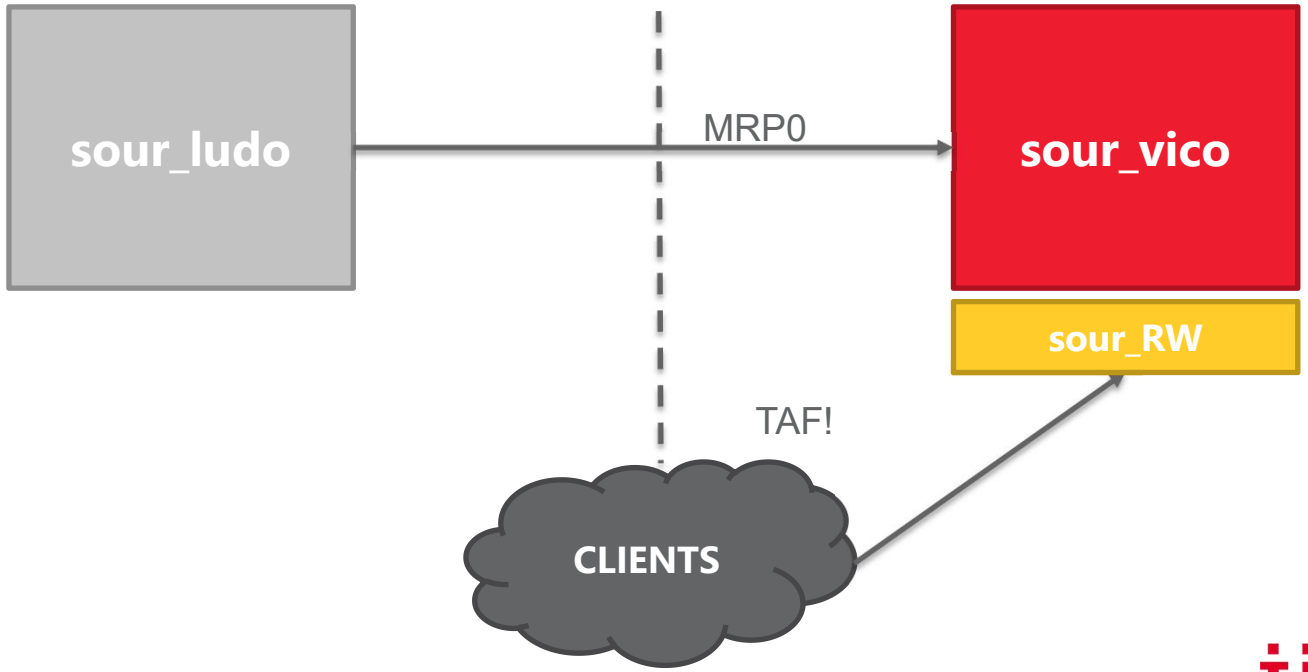
```
(ADDRESS=(HOST=sour-s1)(PORT=1521))
(ADDRESS=(HOST=sour-s2)(PORT=1521))
```



Use Case: Smart Standalone Configuration

```
sour-s1 IN CNAME ludo01
sour-s2 IN CNAME vico01
```

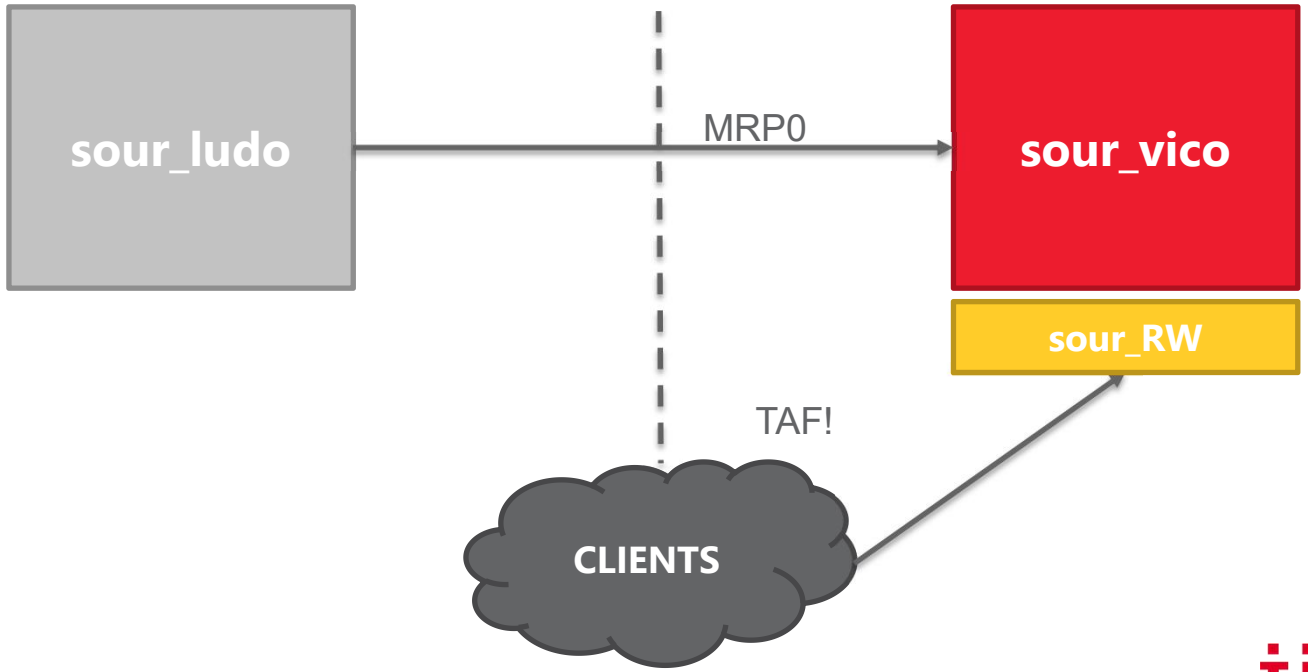
```
(ADDRESS=(HOST=sour-s1)(PORT=1521))
(ADDRESS=(HOST=sour-s2)(PORT=1521))
```



Use Case: Smart Standalone Configuration

```
sour-s1 IN CNAME vico01
sour-s2 IN CNAME vico01
```

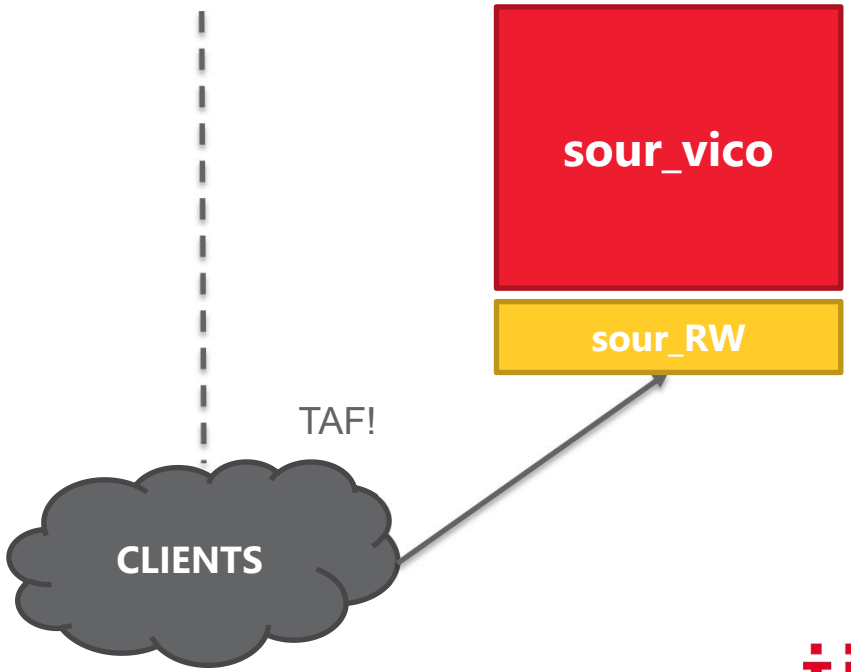
```
(ADDRESS=(HOST=sour-s1)(PORT=1521))
(ADDRESS=(HOST=sour-s2)(PORT=1521))
```



Use Case: Smart Standalone Configuration

```
sour-s1 IN CNAME vico01
sour-s2 IN CNAME vico01
```

```
(ADDRESS=(HOST=sour-s1)(PORT=1521))
(ADDRESS=(HOST=sour-s2)(PORT=1521))
```

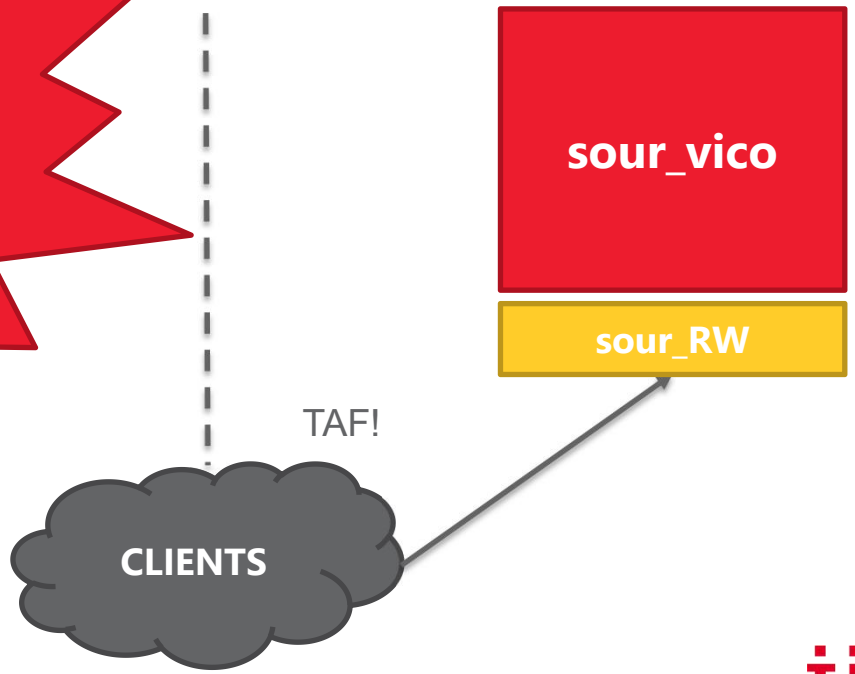


Use Case: Smart Standalone Configuration

```
sour-s1 IN CNAME vi    01
sour-s2 IN CNAME v     01
```

```
(ADDRESS=(HOST=sour-s1)(PORT=1521))
(ADDRESS=(HOST=sour-s2)(PORT=1521))
```

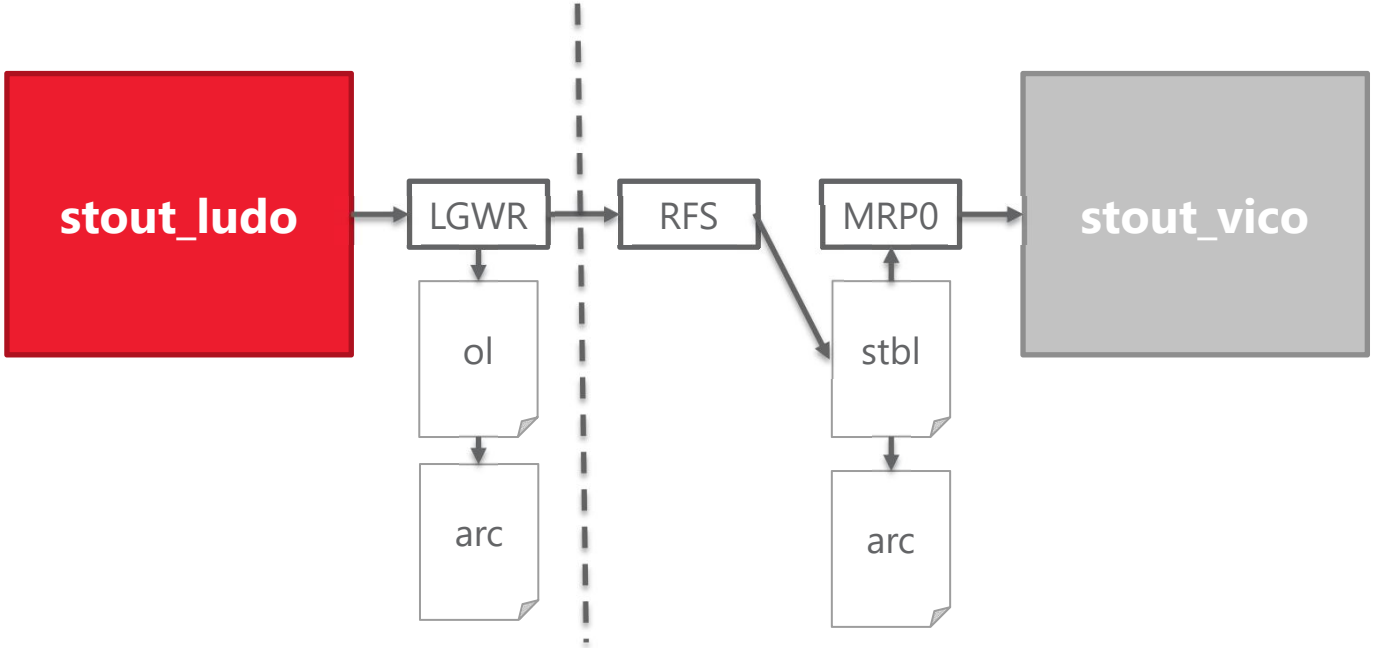
You can switch to a patched version and run datapatch if the patch can be applied online!



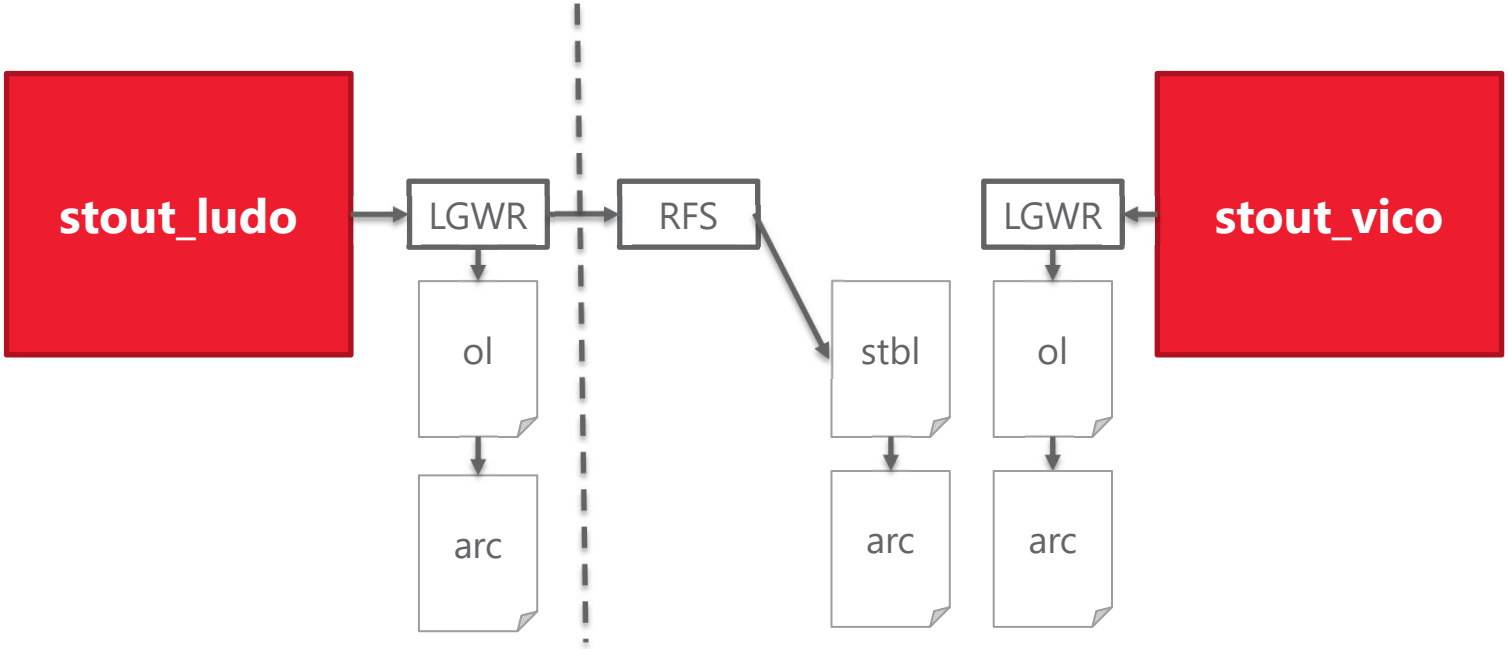
Demo?

Snapshot Standby

Physical Standby (simplified)



Snapshot Standby (simplified)



Snapshot Standby

- Requirements
  - Correct Data Guard configuration with a physical standby
  - Physical standby has Fast Recovery Area Configured
  - FORCE LOGGING is not mandatory but simplifies everything
- Conversion to Snapshot Standby without broker
  - `ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;`
  - `ALTER DATABASE CONVERT TO SNAPSHOT STANDBY;`
  - `ALTER DATABASE OPEN READ WRITE;`
- Conversion to Snapshot standby with broker
  - `CONVERT DATABASE <name> TO SNAPSHOT STANDBY;`

Conversion

- Stop Redo Apply
- Create Guaranteed Restore Point
- Flush Standby Logs
- Clear Online Logfiles
- Convert to Primary
- Open resetlogs
- Start RFS process to get redo stream from the primary

Two sources for Archive Logs?

```
SQL> select THREAD#, SEQUENCE#, RESETLOGS_ID,  FIRST_CHANGE#, NEXT_CHANGE#, REGISTRAR,
ACTIVATION# from v$archived_log WHERE next_time>(sysdate-1/24);
```

THREAD#	SEQUENCE#	RESETLOGS_ID	FIRST_CHANGE#	NEXT_CHANGE#	REGISTR	ACTIVATION#
1	22	952446304	648113	652296	RFS	3044612576
1	23	952446304	652296	652300	RFS	3044612576
1	24	952446304	652300	652361	RFS	3044612576
1	25	952446304	652361	652404	RFS	3044612576
1	26	952446304	652404	652410	RFS	3044612576
1	27	952446304	652410	653523	RFS	3044612576
1	28	952446304	653523	653529	RFS	3044612576
1	29	952446304	653529	657362	RFS	3044612576
1	30	952446304	657362	657399	RFS	3044612576
1	31	952446304	657399	657461	RFS	3044612576
1	32	952446304	657461	661863	RFS	3044612576
1	33	952446304	661863	661869	RFS	3044612576
1	**34**	**952446304**	**661869**	**662080**	**RFS**	**3044612576**
1	**1**	**952543943**	**661845**	**662731**	**ARCH**	**3044656145**

Use Case: SNAP STANDBY ONLY

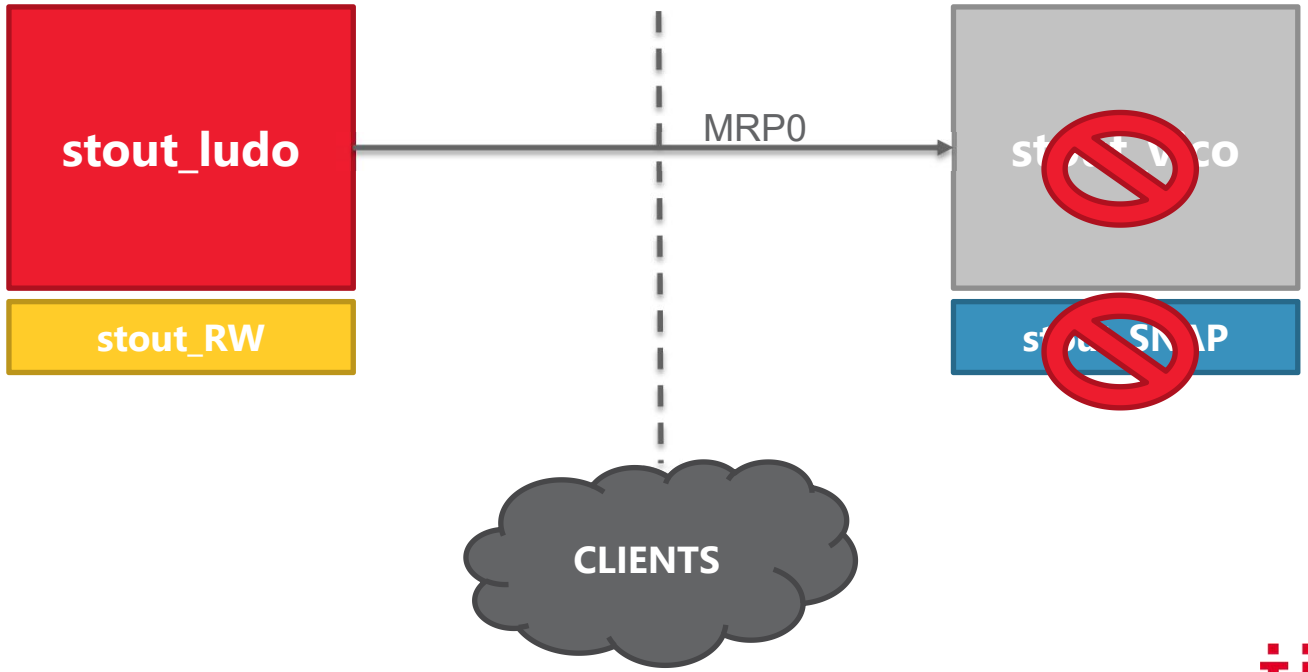
```
srvctl add service -db stout_vico
-service stout_SNAP
-role SNAPSHOT_STANDBY
```

```
(ADDRESS=(HOST=vico01)(PORT=1521))
(ADDRESS=(HOST=ludo01)(PORT=1521))
```

Use Case: SNAP STANDBY ONLY

```
srvctl add service -db stout_vico
-service stout_SNAP
-role SNAPSHOT_STANDBY
```

```
(ADDRESS=(HOST=vico01)(PORT=1521))
(ADDRESS=(HOST=ludo01)(PORT=1521))
```



Use Case: SNAP STANDBY ONLY

```
srvctl add service -db stout_vico -service stout_snap.trivadistraining.com -role
"SNAPSHOT_STANDBY" -failovertype SELECT -failovermethod BASIC

stout_SNAP.trivadistraining.com =
  (DESCRIPTION =
    (CONNECT_TIMEOUT = 5 )(TRANSPORT_CONNECT_TIMEOUT = 3 )
    (ADDRESS_LIST =
      (LOAD_BALANCE = OFF )
      (ADDRESS = (PROTOCOL = TCP )(HOST = vico01.trivadistraining.com )(PORT = 1521 ))
      (ADDRESS = (PROTOCOL = TCP )(HOST = ludo01.trivadistraining.com )(PORT = 1521 ))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = stout_SNAP.trivadistraining.com )
    )
  )
```

Use Case: SNAP STANDBY or PRIMARY

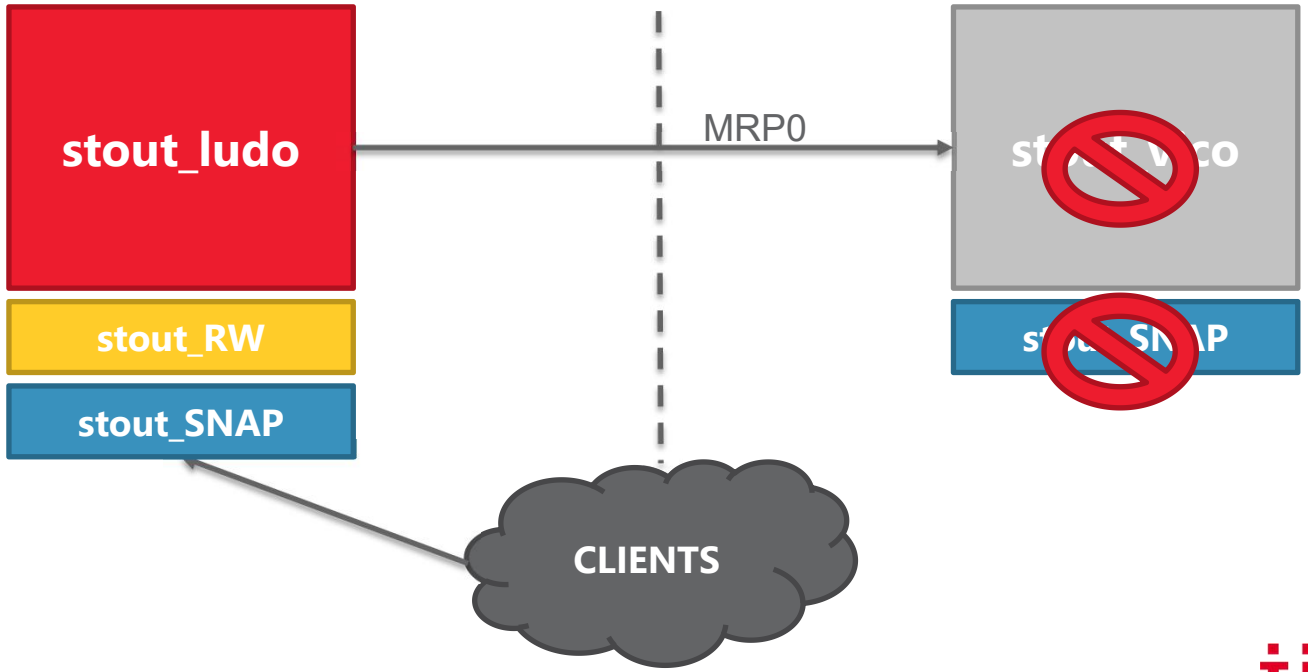
```
srvctl add service -db stout_vico
-service stout_SNAP
-role "SNAPSHOT_STANDBY,PRIMARY"
```

```
(ADDRESS=(HOST=vico01)(PORT=1521))
(ADDRESS=(HOST=ludo01)(PORT=1521))
```

Use Case: SNAP STANDBY or PRIMARY

```
srvctl add service -db stout_vico
-service stout_SNAP
-role "SNAPSHOT_STANDBY,PRIMARY"
```

```
(ADDRESS=(HOST=vico01)(PORT=1521))
(ADDRESS=(HOST=ludo01)(PORT=1521))
```



Use Case: SNAP STANDBY or PRIMARY

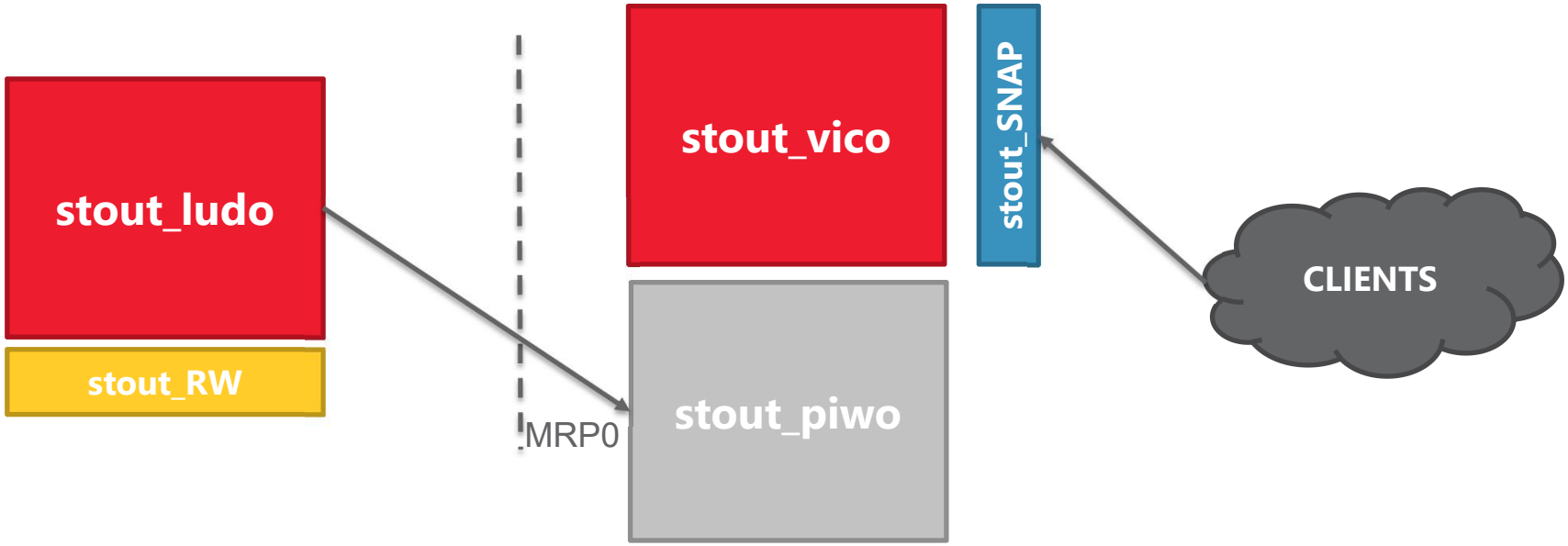
```
srvctl add service -db stout_vico -service stout_snap.trivadistraining.com -role
"PRIMARY,SNAPSHOT_STANDBY" -failovertype SELECT -failovermethod BASIC

stout_SNAP.trivadistraining.com =
  (DESCRIPTION =
    (CONNECT_TIMEOUT = 5 )(TRANSPORT_CONNECT_TIMEOUT = 3 )
    (ADDRESS_LIST =
      (LOAD_BALANCE = OFF )
      (ADDRESS = (PROTOCOL = TCP )(HOST = vico01.trivadistraining.com )(PORT = 1521 ))
      (ADDRESS = (PROTOCOL = TCP )(HOST = ludo01.trivadistraining.com )(PORT = 1521 ))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = stout_SNAP.trivadistraining.com )
    )
  )
```

Demo?

Use Case: Double SNAP STANDBY

```
srvctl add service -db stout_vico
-service stout_SNAP
-role "SNAPSHOT_STANDBY"
```

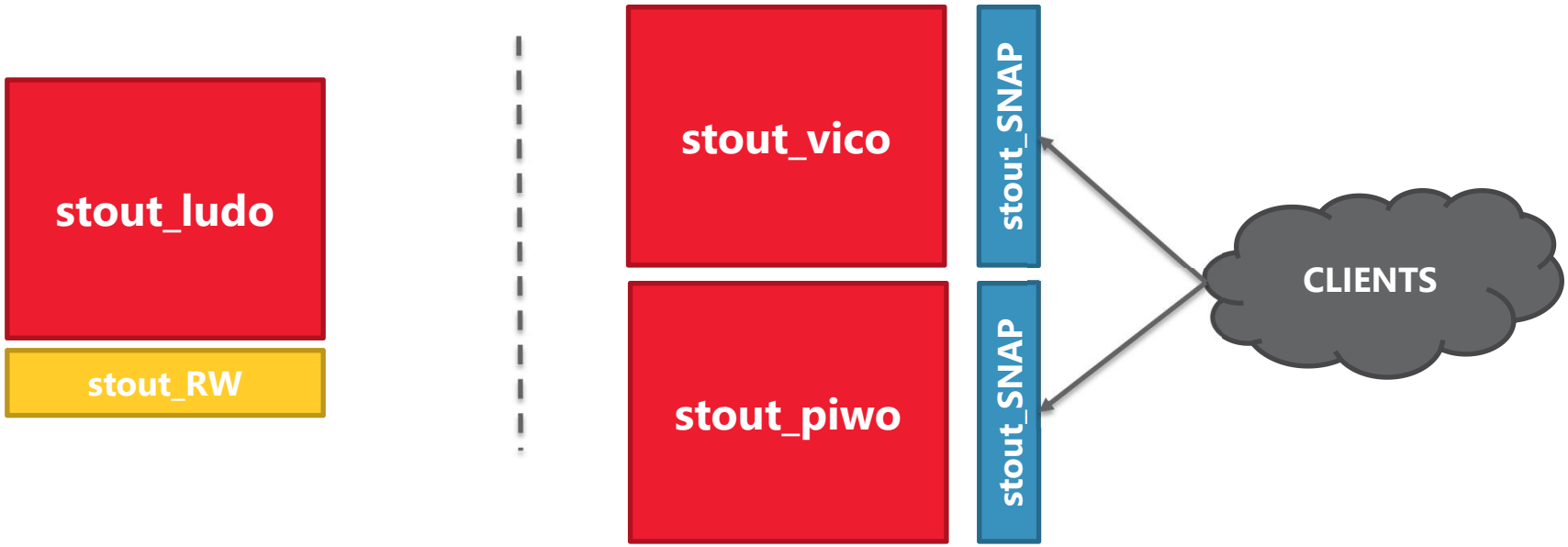
```
(ADDRESS=(HOST=vico01)(PORT=1521))
(ADDRESS=(HOST=piwo01)(PORT=1521))
```



Use Case: Double SNAP STANDBY

```
srvctl add service -db stout_vico
-service stout_SNAP
-role "SNAPSHOT_STANDBY"
```

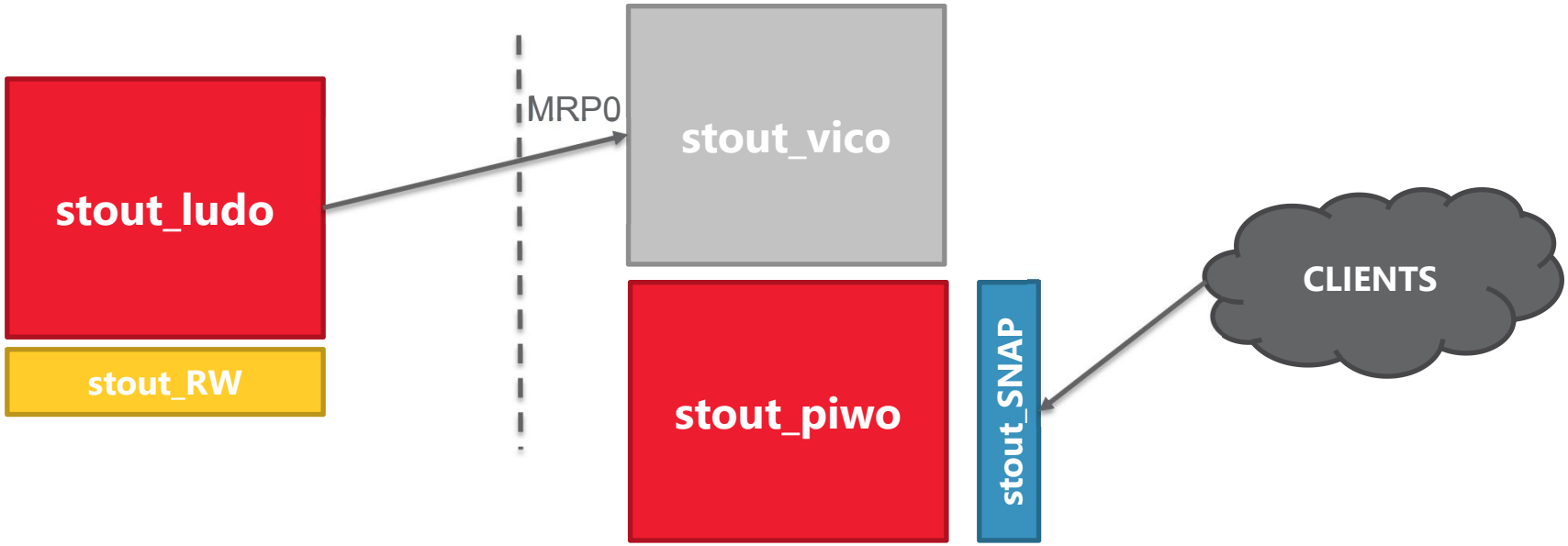
```
(ADDRESS=(HOST=vico01)(PORT=1521))
(ADDRESS=(HOST=piwo01)(PORT=1521))
```



Use Case: Double SNAP STANDBY

```
srvctl add service -db stout_vico
-service stout_SNAP
-role "SNAPSHOT_STANDBY"
```

```
(ADDRESS=(HOST=vico01)(PORT=1521))
(ADDRESS=(HOST=piwo01)(PORT=1521))
```

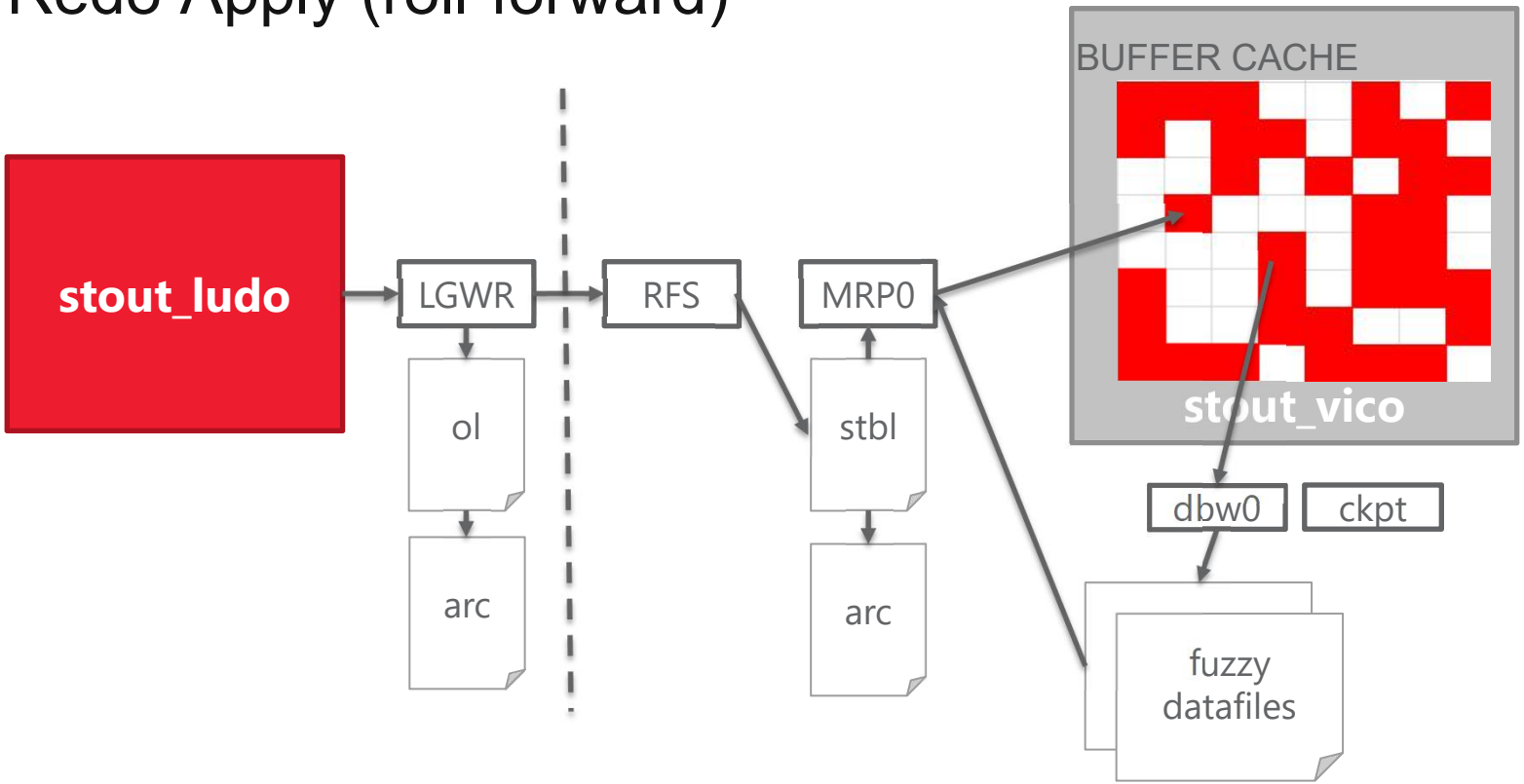


Two Standby Databases Use Case

```
srvctl add service -db stout_vico -service stout_snap.trivadistraining.com -role
"SNAPSHOT_STANDBY" -failovertype SELECT -failovermethod BASIC

stout_SNAP.trivadistraining.com =
  (DESCRIPTION =
    (CONNECT_TIMEOUT = 5 )(TRANSPORT_CONNECT_TIMEOUT = 3 )
    (ADDRESS_LIST =
      (LOAD_BALANCE = OFF )
      (ADDRESS = (PROTOCOL = TCP )(HOST = vico01.trivadistraining.com )(PORT = 1521 ))
      (ADDRESS = (PROTOCOL = TCP )(HOST = piwo01.trivadistraining.com )(PORT = 1521 ))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = stout_SNAP.trivadistraining.com )
    )
  )
```

Standby Consistency on recovery cancel

Redo Apply (roll-forward)



Block status

```
SQL> select status, dirty, count(*) from v$bh
2> group by status, dirty order by status, dirty;

STATUS      D   COUNT(*)
----------- - ----------
free        N         11
mrec        N       3013
mrec        Y         82

DGMGRL> edit database stout_vico set state='APPLY-OFF';

SQL> select status, dirty, count(*) from v$bh
2> group by status, dirty order by status, dirty;

STATUS      D   COUNT(*)
----------- - ----------
free        N       7012
```

Block status

```
SQL> select status, dirty, count(*) from v$bh
2> group by status, dirty order by status, dirty;

STATUS      D   COUNT(*)
----------- - -----------
free        N          11
mrec        N        3013
mrec        Y          82
```

ORA-16037: user requested cancel of managed recovery operation
Managed Standby Recovery not using Real Time Apply
Recovery interrupted!
Recovered data files to a consistent state at change 292504638291

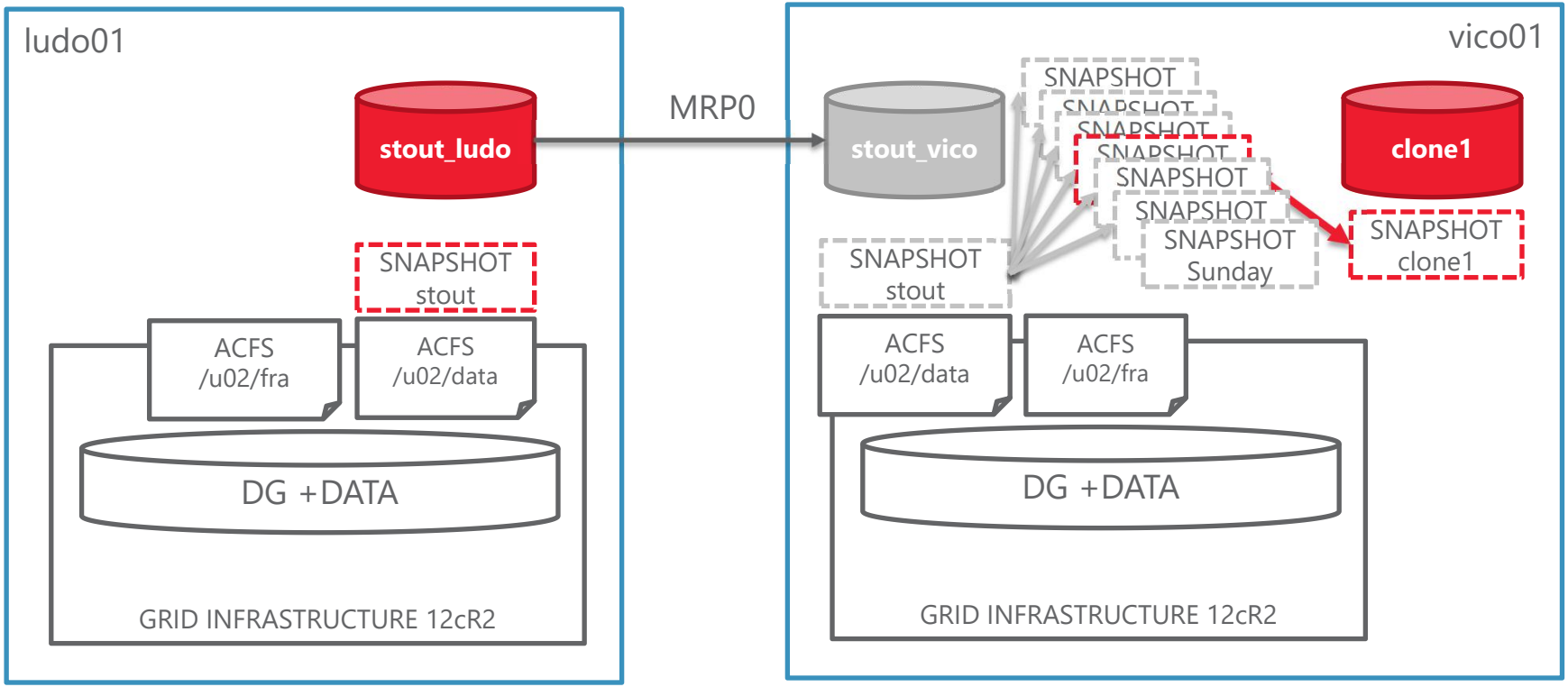
```
DGMGRL> edit database stout_vico set state='APPLY-OFF';

SQL> select status, dirty, count(*) from v$bh
2> group by status, dirty order by status, dirty;

STATUS      D   COUNT(*)
----------- - -----------
free        N        7012
```

Consistent datafiles == consistent database

- No additional media recovery required before OPEN RESETLOGS
  - Cold backups via filesystem commands
  - Cloning possible without bother with Online or Archive Logs!

- Consistent Standby + ACFS Snapshots (or other) = Cheap Database Cloning

Cloning solution based on Standby Database



Cloning Steps

- Snapshot creation
  - edit database stout_vico set state='APPLY-OFF';
  - acfsutil snap create -w -p stout stout.`date +%A` /u02/data
  - alter database backup controlfile to trace as '/u02/data/.ACFS/snaps/…' ;
  - create pfile='/u02/data/.ACFS/snaps/…' from spfile;
  - edit database stout_vico set state='APPLY-ON';

Cloning Steps

- Cloning the database from the snapshot
  - Abort previous instance / delete previous snapshot
  - acfsutil snap create -w -p stout.`date +%A` clone1 /u02/data
  - sed -i -e ‘s/stout_vico/clones/g’ init.ora control_trace.trc
  - startup open resetlogs

Demo?

Let me do one more example

Database reorganization live

- Setup physical standby
- Convert to logical standby << Here the standby is OPEN
- Stop SQL Apply
- Reorganize your database
- Start SQL Apply and catch the primary
- Switchover to Logical Standby

Conclusions

Questions in front of a beer!

High Five POUG
#POUG2017

Ludovico Caldara
Oracle ACE Director
Senior Consultant