

20.11.2025

HA mit Oracle SE2 3 Strategien für maximale Ausfallsicherheit

code of change

 Hyand



Amin Farvardin

Senior Berater DB Services
amin.farvardin@hyand.com

Umfassende Expertise im Datenbankumfeld

Breite technische Kenntnisse in Oracle, Microsoft SQL Server, PostgreSQL sowie Engineered Systems und Cloud. Schwerpunkt auf Datenbankadministration, Performance-Analyse und Infrastructure.

Erfahrung in Migrationen und Betrieb komplexer Systeme

Leitung und Begleitung von Migrationsprojekten in Cloud- und On-Prem-Umgebungen, Managed Services, Ticket- und Incident-Bearbeitung sowie Analyse und Optimierung heterogener Systemlandschaften (Unix, Linux, Windows).

Fundierte Qualifikation und langjährige Praxis

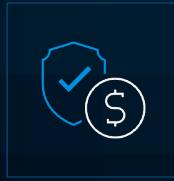
16 Jahre Berufserfahrung, zahlreiche Zertifizierungen (Oracle, PostgreSQL, RHCE) sowie umfangreiche Erfahrung in Workshops, Schulungen und Vorträgen. Kenntnisse in Shell/Bash und .NET.



Automotive



Retail



Banking &
Insurance



Public
Sector



Lösungen für führende Unternehmen



International aufgestellt
für Ihren Erfolg.

25+

Kunden
mit > 1 Mrd. €
Umsatz

110+

Mio. €
Umsatz

850+

Mitarbeitende

24

Nationalitäten

16

Standorte
in 5 Ländern

25 %

Frauen im
Unternehmen

ORACLE Editions und High Availability

HA mit Oracle SE2 – 3 Strategien für maximale Ausfallsicherheit

EE (Enterprise-Edition)

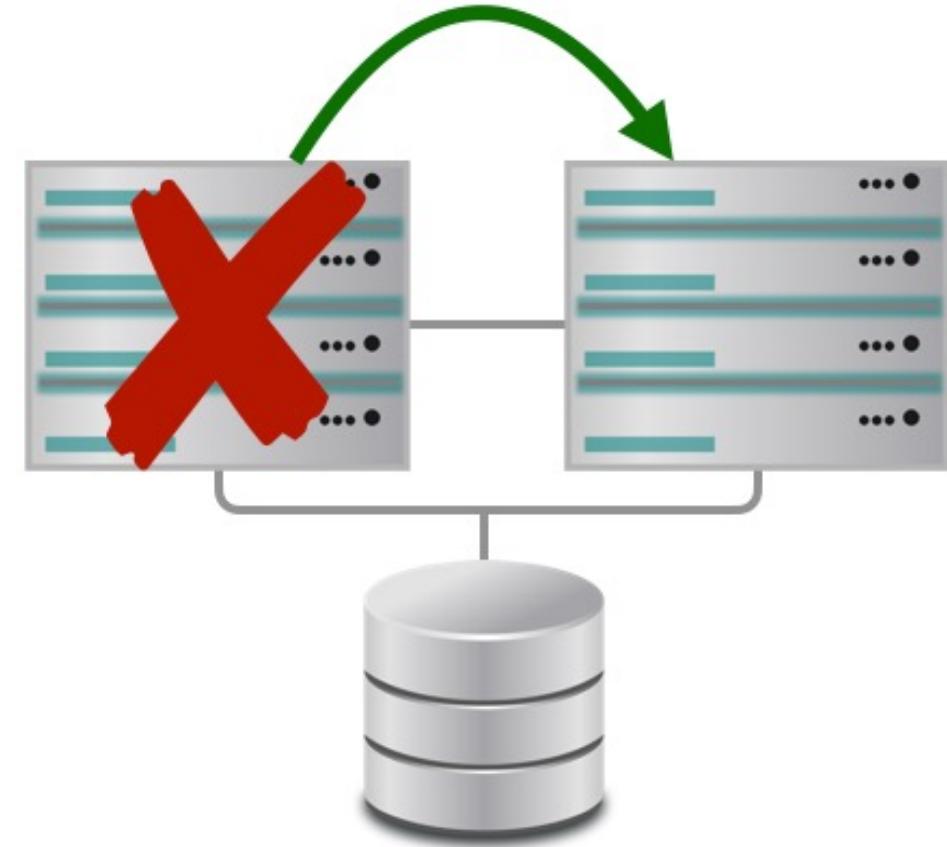
- Volle Funktionalität: RAC, Data Guard, Flashback Data Archive etc.
- Zusatzoptionen: Active Data Guard, Partitioning etc.
- Lizenz pro CPU-Core teuer (~ 47.500 USD/Core)

SE2 (Standard-Edition 2)

- Kein RAC, kein Data Guard
- Max. 2 Sockets, max. 16 Threads/DB
- Kein GoldenGate (lizenztechnisch nicht erlaubt)
- Günstiger (~ 17.500 USD/Socket)

Warum High Availability auch bei SE2 ein Thema ist

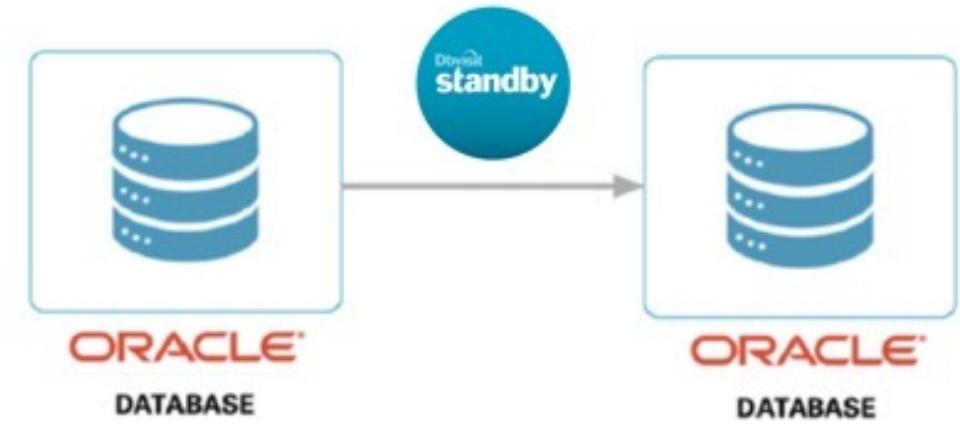
- Kein RAC, kein Data Guard in SE2
→ kein automatisches Failover
- Single-Server-Setup = Single Point of Failure
- Trotzdem geschäftskritische Datenbanken
- Minimale Downtime – ohne EE-Lizenz
- Restore ≠ High Availability



Strategie 1: Dbvisit Standby + Observer

Strategie 1: Dbvisit Standby + Observer

- Konzept: Replikation über Archive Logs
- Kompatibel mit SE2
- Automatisches Failover über Observer
- GUI & CLI, einfach zu bedienen
- Praxisbeispiel: ODA X10 Primary ↔ Standby



Aktuell unterstützte Versionen

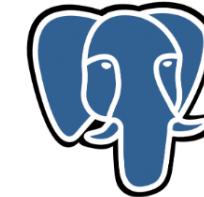
- v12.2:** Neueste Version (veröffentlicht September 2025)
- v11.7.4:** Frühere Version (Mai 2025)
- v9:** weiterhin für AIX/Solaris-Systeme verfügbar,
voraussichtlich bis April 2027 unterstützt
- v10:** Unterstützung wurde verlängert, das Ende des Supports steht bevor.
Es wird empfohlen, auf v11 oder v12 zu aktualisieren
- v8:** Standard-Support endete im Juli 2021, der erweiterte Support läuft im Juli 2024 aus
- v7 und v6:** Support wurde eingestellt

Supported Operating Systems List

Supported Linux Platform Distributions	Supported Windows Platform Versions (64-bit)
<ul style="list-style-type: none">• Oracle Linux 6• Oracle Linux 7• Oracle Linux 8• Oracle Linux 9• Red Hat Enterprise Linux 6• Red Hat Enterprise Linux 7• Red Hat Enterprise Linux 8• Red Hat Enterprise Linux 9• SUSE Linux Enterprise Server 11• SUSE Linux Enterprise Server 12• SUSE Linux Enterprise Server 15	<ul style="list-style-type: none">• Windows Server 2012• Windows Server 2012 R2• Windows Server 2016• Windows Server 2019• Windows Server 2022• Windows Server 2025 (Oracle DB platform only)

Strategie 1: Dbvisit Standby + Observer

Supported Database List



PostgreSQL



Microsoft
SQL Server®

Support for cloud deployments

Source/Target	AWS EC2	AWS RDS	AWS RDS Custom	OCI DBS (Oracle RDBMS only)	OCI Compute	OCI Autonomous	Azure VM	On-Prem
On-Prem	Supported	Not Supported	Not Supported	Supported*	Supported	Supported	Supported	Supported
AWS EC2	Supported	Not Supported	Not Supported	Supported*	Supported	Not Supported	Supported	Supported
AWS RDS	Not Supported	Not Supported	Not Supported	Not Supported	Not Supported	Not Supported	Not Supported	Not Supported
AWS RDS Custom	Not Supported	Not Supported	Supported	Not Supported	Not Supported	Not Supported	Not Supported	Not Supported
OCI DBS (Oracle RDBMS only)	Supported*	Not Supported	Not Supported	Supported	Supported*	Not Supported	Supported*	Supported*
OCI Compute	Supported	Not Supported	Not Supported	Supported*	Supported	Not Supported	Supported	Supported
OCI Autonomous	Not Supported	Not Supported	Not Supported	Not Supported	Not Supported	Not Supported	Not Supported	Not Supported
Azure VM	Supported	Not Supported	Not Supported	Supported*	Supported	Not Supported	Supported	Supported
On-Prem	Supported	Not Supported	Not Supported	Supported*	Supported	Not Supported	Supported	Supported

Installation

- Recommended Operating System Packages
- Network Connectivity

Am besten auf ACFS

```
./install-agent-v12.1 -batch -install-dir /usr/dbvisit -host-address xxx  
-control-center-address xxx -passphrase a
```

```
./install-control-v12.1 -batch -passphrase Admin123
```

Strategie 1: Dbvisit Standby + Observer

The image shows the Standby MP software interface. On the left, there is a sidebar with the following menu items:

- DASHBOARD
- NEW CONFIGURATION
- SQL SERVER INSTANCES
- USERS
- TASKS & EVENTS
- SETTINGS
- LOG OUT

The main area displays the following content:

New Configuration

Choose a Host Pair,
and begin creating your [New Oracle Database Configuration](#).

 [Start with Oracle](#)

Discover Instances,
and begin creating your [New SQL Server Database Configuration](#).

 [Start with SQL Server](#)

A quick breakdown of how this works ▾

Strategie 1: Dbvisit Standby + Observer

Primary Server Settings		
DBVISIT_BASE	/usr/dbvisit/standbymp	 
SOURCE	[REDACTED]	 
ORACLE_SID	[REDACTED]	 
NETPORT	7890	 
ARCHSOURCE	/u02/dbvisit_archive	 
LOGSWITCH	N	 
DB_UNIQUE_NAME	[REDACTED]	 
OS		
BCKDIR	/u02/dbvisit_archive/backup	 
DBVISIT_BASE	/usr/dbvisit/standbymp	

Strategie 1: Dbvisit Standby + Observer

Standby Server Settings		
DBVISIT_BASE_DR	/usr/dbvisit/standbymp	
DESTINATION	[REDACTED]	
ORACLE_SID_DR	[REDACTED]	
ORACLE_HOME	/u01/app/odaorahome/oracle/product/19.0.0.0/dbhome_1	
ORACLE_HOME_DR	/u01/app/odaorahome/oracle/product/19.0.0.0/dbhome_1	
ORACLE_SID_ASM	+ASM1	
ORACLE_SID_ASM_DR	+ASM1	
NETPORT_DR	7890	
ARCHDEST	/u02/dbvisit_archive	
LEAVE_COMPRESS_DR	N	
DB_UNIQUE_NAME_DR	[REDACTED]	
OS_DR		
PATH_DR		
BCKDIR_DR	/u02/dbvisit_archive/backup	

Strategie 1: Dbvisit Standby + Observer

Standby Archive Log Management Settings			
ARCHDEST_MANAGEMENT	Y		
DAYS_TO_KEEP_ARCHDEST	3		
NUM_ARCHDEST_TO_KEEP	0		
THRESHOLD_ARCHDEST	80		
DELETE_ARCHDEST_THRESHOLD	N		

Strategie 1: Dbvisit Standby + Observer

Automated Standby Update Settings

Here you can set up a regular schedule to create a backup log of all changes on this Primary Database, send the backup over to Standby, and restore it.

We recommend setting this Schedule to run regularly (e.g. every 600 seconds/10 minutes) to ensure your Standby Database is always up to date.

ONLINE → RECOVERING

9 minutes 35 seconds

Automated Standby Update: Enabled

Enabled Perform Log Shipping every seconds

+ 2 DR POINT

[Close](#)

████████████████████ (Oracle)
Configuration ID: 1
License expires on 2099-06-05 [UPDATE](#)

ACTIONS

- [View Detailed Log Gap Report](#)
- [Backup & Send Logs](#)
- [Synchronize](#)
- [Graceful Switchover](#)
- [Start/Stop Database](#)
- [Activate Standby](#)
- [Remove Configuration](#)

HELP

- [Create support package](#)

SETTINGS

- [Automated Standby Update](#)
- [Standby Update Delay](#)
- [Email Settings](#)
- [Slack Settings](#)
- [Observer/Auto-Failover](#)
- [Archive Log Management](#)

TESTS

- [Test Standby Activation](#)
- [Test Opening Standby Read-Only](#)

Strategie 1: Dbvisit Standby + Observer

PRIMARY 

Current Source sequence: # **173300**

Archived Source sequence: # **173299**

Time lag	9 minutes
Transfer log gap	0
Archive log gap	0
Last transferred	2025-10-21 13:45:30

STANDBY 

Destination Next Required Recovery Sequence # **173300**

ACTIONS

-  View Detailed Log Gap Report
-  Backup & Send Logs
-  Synchronize
-  Graceful Switchover
-  Start/Stop Database
-  Activate Standby
-  Remove Configuration

HELP

-  Create support package

SETTINGS

-  Automated Standby Update
-  Standby Update Delay
-  Email Settings
-  Slack Settings
-  Observer/Auto-Failover
-  Archive Log Management

TESTS

-  Test Standby Activation
-  Test Opening Standby Read-Only

SNAPSHOTS/REPLICA

-  Snapshots
-  Reporting Replica

ADVANCED OPTIONS

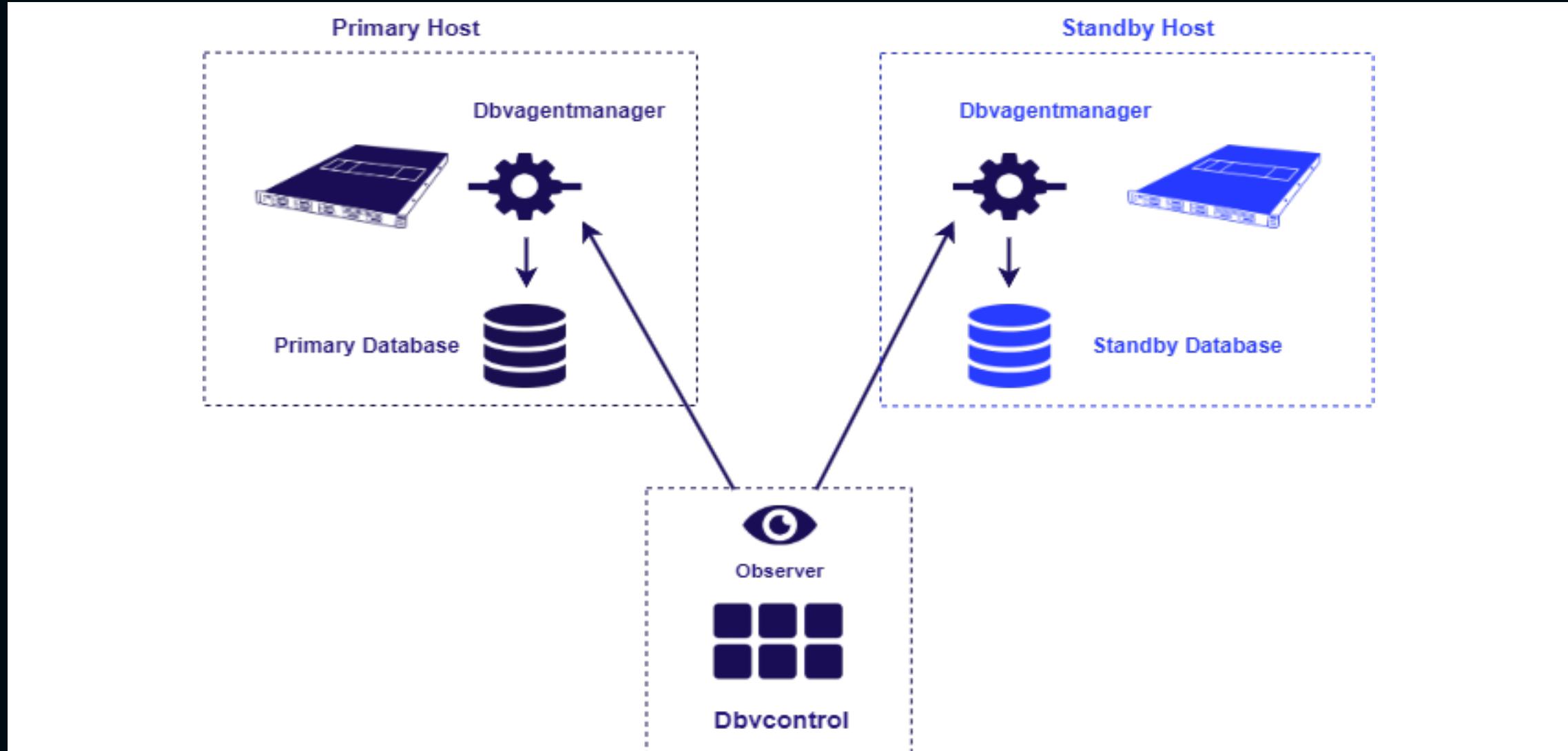
click 

 Hyand

HA mit Oracle SE2 – 3 Strategien für maximale Ausfallsicherheit

20

Strategie 1: Dbvisit Standby + Observer



Strategie 1: Dbvisit Standby + Observer

The screenshot shows the 'Edit Observer Settings' page with the following configuration:

- SLASH**: 26 minutes 59 seconds
- ONLINE**: czlin0231
- RECOVERING**: czlin0232

MAIN SETTINGS (Top right): +1 DR POINT

- Enable/Disable the Observer**: Enabled (button) 1
- Selected Emergency Action**: Notifications only OR Perform Automatic Failover 2
- Configuration status check interval (sec)**: 10 3
- Maximum number of failed checks before action is taken**: 3 3
- Send Email Notifications**: Disabled 4
- Send Slack Notifications**: Disabled 4
- Enable Heartbeat messages**: Disabled 5
- Heartbeat messages are sent via Slack or Email. You need Slack or Email Notifications to be enabled to use Heartbeat notifications.** (Info message) 6
- Custom Observer Scripts**: Disabled 6

NOTIFICATIONS (Section header):

CUSTOM OBSERVER SCRIPTS (Section header):

Buttons at the bottom:

- Save (green checkmark) 7
- Cancel
- Close

Zwei wichtiger Parameter, wenn man eine ODA nutzt:

TMPDIR_DR = /u02/dbvisit_archive/tmp
TMPDIR = /u02/dbvisit_archive/tmp

Das Problem: [DBVISIT_BASE]/standbymp/oracle/tmp (Default)

- Only applicable when using ASM. Directory on Primary server to transfer online redo and controlfile from ASM plus additional GS related files
- It is very important to pre-create the expected sub-directory structure, which is not part of the variable value as mentioned in example..

Example:

If variable is set like so:

TMPDIR=/tmp/gs

Then pre-create following directory structure:

/tmp/gs/GS/<DDC>

where DDC is your DDC name case sensitive, so for example:

mkdir -p /tmp/gs/GS/PROD

Vorteil

- SE2-kompatibel, offiziell unterstützt
- Automatisches Failover möglich
- Übersichtliche GUI & CLI
- HA durch zwei Server
- Verlässlicher und schneller Support

Nachteil

- Kein Zero Data Loss
- Replikationsverzögerung bei Netzproblemen
- Lizenzkosten (Dbvisit)

Strategie 2: Cluster mit VIP-Umschaltung

Strategie 2: Cluster mit VIP-Umschaltung

- Zwei Knoten mit gemeinsamem Storage oder repliziertem FS
- Bei einem Ausfall wird die VIP automatisch umgeschaltet
- Die Datenbank ist immer nur auf einem Knoten aktiv



Einrichten

1. Datenbank nach der Installation von Clusterware entfernen
2. VIP in Cluster Resource hinzufügen
3. Die VIP in tnsnames.ora von Datenbank (in beiden Knoten) hinzufügen und in local_listener setzen
4. Die VIP in listener.ora (von Grid) definieren
5. Ein Cluster Resource für Listener in clusterware definieren (Mit dependency auf VIP)
6. Ein Cluster Resource für Datenbank in Clusterware definieren (Mit dependency auf Listener)
7. Die Action-Scripts müssen angelegt werden

Strategie 2: Cluster mit VIP-Umschaltung

1. Datenbank nach der Installation von Clusterware entfernen:

```
srvctl remove database -db DBSE2 -f
```

2. VIP in Cluster Resource hinzufügen (als root):

```
appvipcfg create -network=1 -ip=192.168.x.x -vipname=dbse2.vip -user=grid
```

3. Die VIP in tnsnames.ora von Datenbank (in beiden Knoten) hinzufügen und in local_listener setzen:

```
LISTENER_DBSE2 =  
(ADDRESS = (PROTOCOL = TCP) (HOST = vip) (PORT = 1521))
```

4. Auf beiden Knoten die Listener mit VIP in listener.ora hinzufügen:

```
LISTENER_DBSE2=(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP) (HOST=vip) (PORT=1521))  
(ADDRESS=(PROTOCOL=IPC) (KEY=LISTENER_DBSE2) )))  
ADMIN_RESTRICTIONS_LISTENER_DBSE2=on
```

Strategie 2: Cluster mit VIP-Umschaltung

5. Ein Cluster Resource für Listener/DB in clusterware definieren (Mit dependency auf VIP)

```
TYPE=cluster_resource
ACL=owner:oracle:rwx,pgrp:oinstall:r--,other::r--,group:dba:r-x,group:oper:r-x,user:oracle:r-x
ACTIONS=startoption,group:"oinstall",user:"oracle",group:"dba",group:"oper"
ACTION_SCRIPT=/u01/app/grid/scripts/actions/db_DBSE2.pl
ACTIVE_PLACEMENT=0
AUTO_START=restore
CARDINALITY=1
CHECK_INTERVAL=60
DEGREE=1
DESCRIPTION=Listener_DBSE2 / db_DBSE2
ENABLED=1
HOSTING_MEMBERS=host1 host2
LOGGING_LEVEL=1
PLACEMENT=restricted
RESTART_ATTEMPTS=1
START_DEPENDENCIES=hard(dbse2.vip) pullup(dbse2.vip) / START_DEPENDENCIES=hard(dbse2.lsnr) pullup(dbse2.lsnr)
STOP_DEPENDENCIES=hard(intermediate:dbse2.vip)
SCRIPT_TIMEOUT=60
UPTIME_THRESHOLD=1h
```

Strategie 2: Cluster mit VIP-Umschaltung

6. Action_Script:

- Start Function
- Stop Function
- Check Function
- Clean Funktion

7. Cluster Resource (DB) kann jetzt mit relocate Resource in Clusterware (crsctl) geschwenkt werden :

```
crsctl relocate resource res_name -f
```

Vorteil

- Kein Data Guard/Archivelog Shipping
- Keine zusätzl. Lizenz/Kosten
- Schnelles Failover – kein Dataverlust

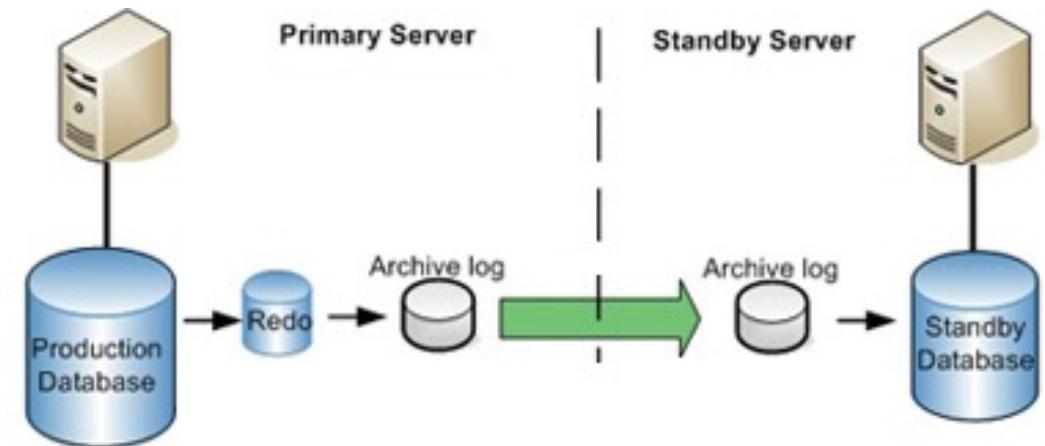
Nachteil

- Kein Schutz vor logischer Korruption
- Shared Storage = Single Point of Failure
- Extra Aufwand wegen Shared Storage

Strategie 3: REST-/Skript-Umschaltung

Strategie 3: REST-/Skript-Umschaltung

- Manuell Standby bauen
- Mit REST API oder Shell Skript die Standby sync machen
- Manuell/Automatisch Failover/ Switchover mit API oder Shell Script



Manuell Standby bauen

- Force Logging
- Parameter File/Password File
- tnsnames.ora/sqlnet.ora
- Restore standby controlfile from service
- Restore database from service

Strategie 3: REST-/Skript-Umschaltung

Mit REST API / Shell Skript das Standby synchronisieren:

```
#!/bin/bash
export ORACLE_SID=
export ORACLE_HOME=
ARCHIVELOG_DIR="/u02/sync_arc/archivelog"
sudo chown -R oracle:oinstall $ARCHIVELOG_DIR/
rman target / << EOF
catalog start with '/u02/sync_arc/archivelog' noprompt;
EOF

# Standby Recovery automatisch durchführen
sqlplus / as sysdba <<EOF
SET FEEDBACK OFF;
SET HEADING OFF;
SET PAGESIZE 0;
RECOVER STANDBY DATABASE USING BACKUP CONTROLFILE;
AUTO
EXIT;
EOF
# Lösche Archive Logs, die älter als 1 Tag sind
find $ARCHIVELOG_DIR -type f -mtime +1 -exec rm -f {} \;
```

Strategie 3: REST-/Skript-Umschaltung

Manuell/Automatisch Failover/Switchover mit API oder Shell Script:

- Shutdown Primary DB und copy/recover letztes Archivelog in Standby DB
- Standby DB aktivieren:

```
ALTER DATABASE ACTIVATE STANDBY DATABASE;
```

- Restore standby controlfile from service auf neues Standby
- Neue Standby DB synchronisieren

Vorteil

- SE2-kompatibel, offiziell unterstützt
- Automatisches Failover möglich
- HA durch zwei Server
- Flexibel, keine Zusatzsoftware

Nachteil

- Kein Zero Data Loss
- Replikationsverzögerung bei Netzproblemen
- Eigenentwicklung, kein Support

Zusammenfassung

Vergleich der Strategien

Dbvisit:

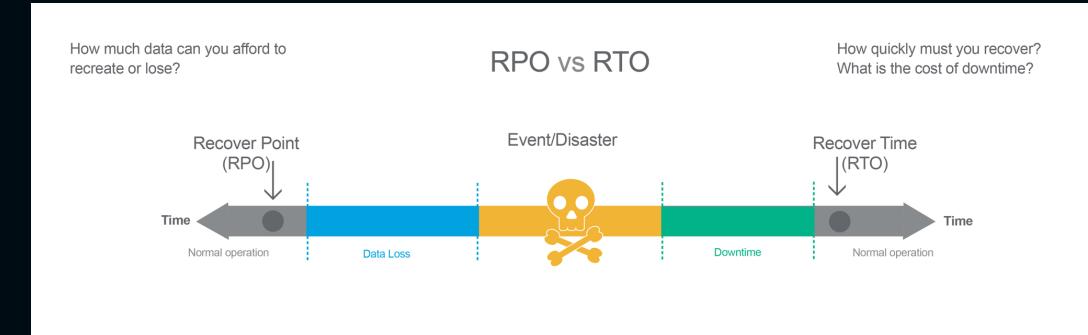
RTO: 1–2 min, RPO: Minuten bis Sekunden, Aufwand gering

Cluster:

RTO: 1–2 min, RPO: 0, Aufwand gering

REST/Skript:

variabel, Aufwand hoch



RTO = Recovery Time Objective

→ Die *maximal tolerierbare Zeit*, die ein System nach einem Ausfall *braucht, um wieder funktionsfähig zu sein*

RPO (Recovery Point Objective)

→ Wieviel **Datenverlust** maximal tolerierbar ist (z. B. 5 Minuten Log-Gap)

Fazit

SE2 kann
hochverfügbar
betrieben werden

Kombination aus Prozess,
Anwendungen, Preis,
Monitoring,
Automatisierung und
Tests entscheidend

HA ist kein
Produkt,
sondern
ein Prozess

Vielen Dank
für Ihre Aufmerksamkeit!



© 2025 – The thoughts and ideas developed are the intellectual property of Hyand
and are subject to copyright. Reproduction, disclosure to third parties or use
– even in part – is only permitted with the express consent of Hyand.