DEVK

Aufbau einer 12c RAC & Data Guard Umgebung mit NFS Storage bei der DEVK

Tim Hensel, Nürnberg, 15.11.2016

Inhaltsverzeichnis

Inhaltsverzeichnis

Vorstellung Projekt

Vertriebsunterstützende Systeme wurden auf „always online“ Lösung ausgerichtet.

Basis hierfür sollte neue Oracle-Infrastruktur sein.

Anforderungen:

- **hochverfügbar**
- **stabil**
- **schnelle Wiederherstellbarkeit**
- **möglichst unterbrechungsfreie Wartbarkeit**

Vorstellung Projekt

Betriebskonzet der DEVK:

2 gleichberechtigte Rechenzentren

Staging-Verfahren:

- **Maintenance**
- **Test/Entwicklung**
- **Vorproduktion**
- **Produktion**

Inhaltsverzeichnis

Datenbank-Architekturentscheidung

Vor endgültiger Entscheidung wurden diverse Zielarchitekturen betrachtet, gewichtet und bewertet:

- **Virtualisierung mit VMware oder OracleVM**
- **Oracle Engineerd Systems: ODA, Exadata**
- **Hardware-Lösung**
- **jeweils mit/ohne RAC und/oder Data Guard**

Gewichtung und Bewertung mit Kollegen von Oracle

Primäres Ziel niedrige Lizenzkosten

Lizenz-Politik von Oracle im VMware-Umfeld?

Datenbank-Architekturentscheidung

DEVK

Folgende Punkte beeinflussten die Architekturentscheidung zusätzlich:

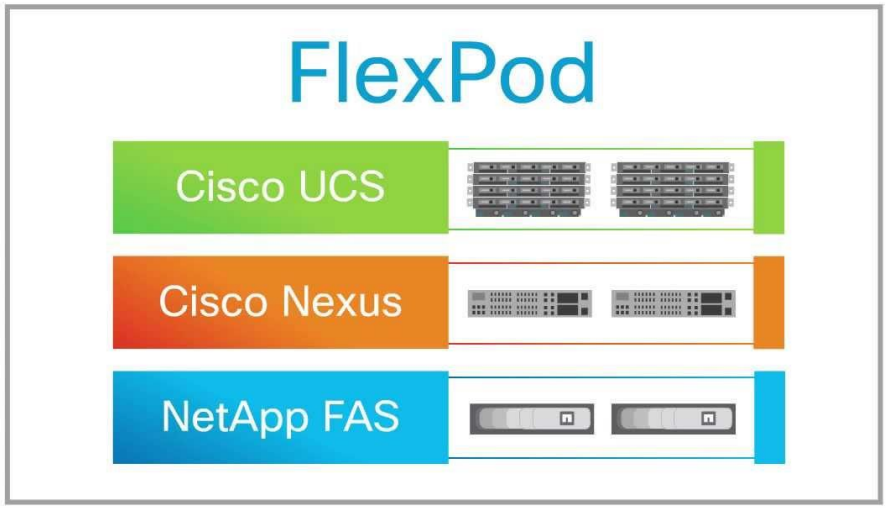
- **Strategische Produkte (u.a. VMware)**
- **Vermeidung „neuer“ Produkte wie engineered Systems**
- **Einflussfaktoren von anderen Projekten und deren Erfahrungen**
- **Einflussnahme von Entscheidungsträgern**
- **Neben Oracle Datenbanken auch Applikationsserver relevant**

Datenbank-Architekturentscheidung

Letzendlich fiel Entscheidung auf eine Hardware-Lösung mit CISCO UCS-Blades und NetApp-Storage.

Stichwort „FlexPod-Architektur“:

- **Referenzarchitektur bestehend aus Server- und Netzwerk-Komponenten (Cisco) sowie Storage (NetApp)**
- **Hypervisor-Technologie (VMware)**



Datenbank-Architekturentscheidung

DEVK

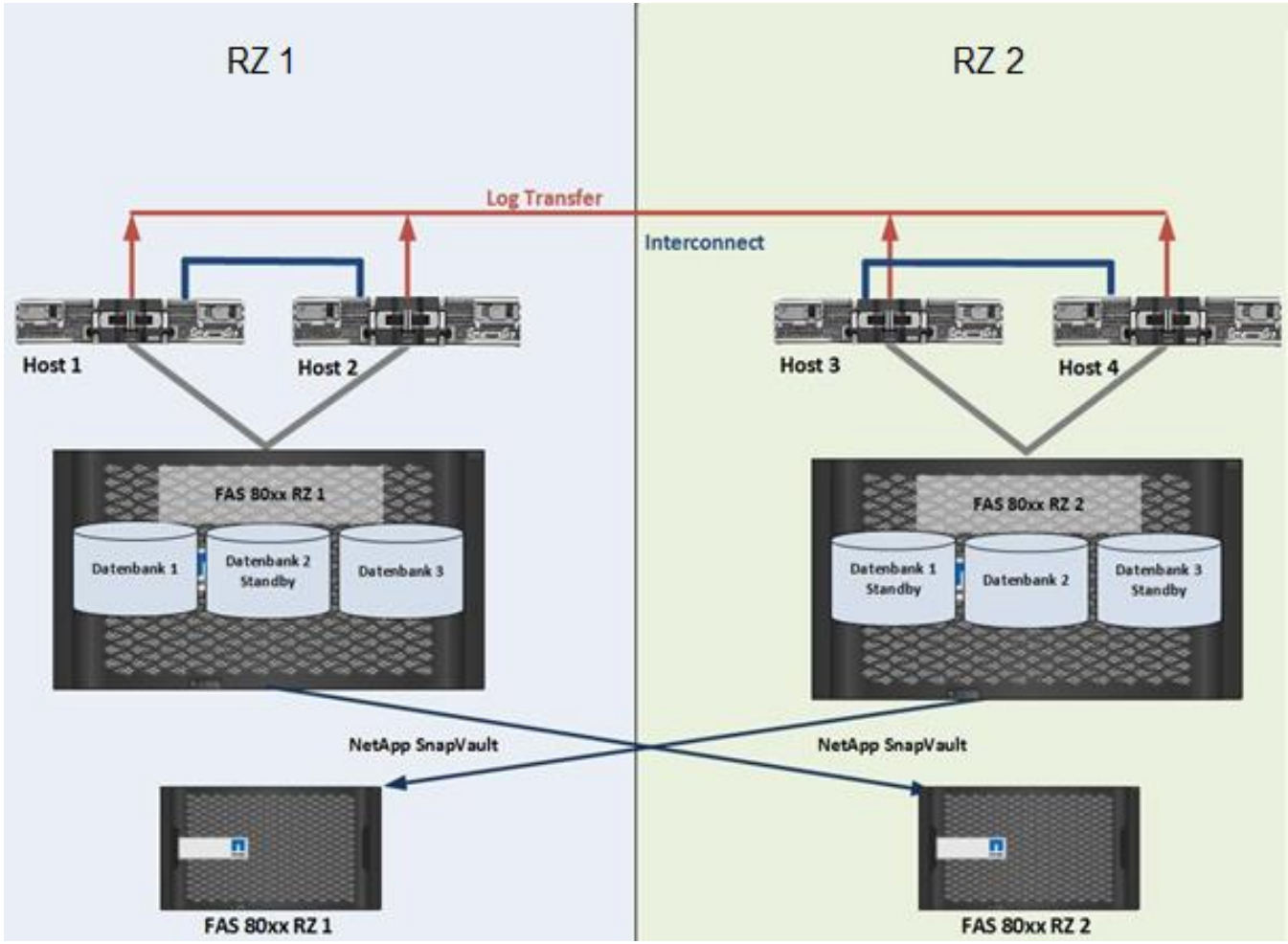
Ansatz 1: NetApp Metrocluster mit Streched RAC (3 Knoten) über 2 Standorte

- **Manuelles Umschalten Metrocluster notwendig**
- **Einfluss auf die Verfügbarkeit des RAC Clusters**

Ansatz 2: Hochverfügbarkeit mit RAC und Data Guard

- **Maximum Availability Architecture weit verbreitet**
- **Kein Observer (kein dritter Standort), manuelles „Umschalten“**
- **kein Active Data Guard und kein Multitenant**

Finale Architektur in Produktion



Finale Architektur pro Stage

	RZ1	RZ2
Maintenance	2 Server, 3 DBs	2 Server, 2 DBs
Test/Entwicklung	1 Server, 4 DBs	
Vorproduktion	2 Server, 3 DBs	2 Server, 3DBs
Produktion	2 Server, 3 DBs	2 Server, 3DBs
Summe	7 Server, 13 DBs	6 Server, 8 DBs
	13 Server, 21 DBs	

3. Filesystem-Architektur der Datenbanken

Filesystemarchitektur

NetApp Snapshot-Technologie als Bestandteil von FlexPod:

- **NFS-Shares als Shared Storage**
- **schnelle Wiederherstellung auch „großer“ Datenbanken**

Alternative ASM auf NFS wurde bewusst nicht gewählt, sondern dNFS eingesetzt

Sicherungen mit NetApp SnapCenter-Plugin für Oracle Datenbanken

Teilnahme an Beta-Programm vor Release im März 2016 und somit intensives Testen

Konfigurationen zur Installation

Datenbankserver mit SLES 11 SP4 und Oracle 12.1.0.2 wegen SnapCenter

Basisinstallation und Konfiguration der Server mit Automatisierungsplattform Ansible

Checks diverser Konfigurationen durch ServerSpec

Für Verzeichnisstruktur Orientierung an Optimal Flexible Architecture (OFA)

Kein Shared Oracle Home, um möglichen SPoF zu vermeiden und um im Wartungsfall flexibler zu sein

/etc/oranfstab für Datenbank-Volumes

- **entgegen Oracle-Vorgaben unterschiedlich aufgrund von local-Parameter (redundante Storage-Anbindung der Volumes)**

/etc/oranfstab und /etc/fstab

DEVK

Redundante Einträge in oranfstab und fstab obligatorisch

Auszug /etc/oranfstab Node 1 und Node 2:

```
server: <VOLUME_NAME>
path:   <IP_NFS_STORAGE>
local:  <IP_LOCAL_BONDING>1
local:  <IP_LOCAL_BONDING>2
local:  <IP_LOCAL_BONDING>3
local:  <IP_LOCAL_BONDING>4
export: /<VOLUME_NAME>  mount:/u03/oradata/<DBNAME>
```

```
server: <VOLUME_NAME>
path:   <IP_NFS_STORAGE>
local:  <IP_LOCAL_BONDING>5
local:  <IP_LOCAL_BONDING>6
local:  <IP_LOCAL_BONDING>7
local:  <IP_LOCAL_BONDING>8
export: /<VOLUME_NAME>  mount:/u03/oradata/<DBNAME>
```

Auszug /etc/fstab auf beiden Knoten gleich:

```
<IP_NFS_STORAGE>:/<VOLUME_NAME> /u03/oradata/<DBNAME> nfs rw,bg,hard,vers=3,proto=tcp,timeo=600,rsize=65536,wsize=65536,nointr,actimeo=0 0 0
```

NFS-Volumes je Datenbank

DEVK

Für jede Datenbank 7 Volumes angelegt auf Basis best practices von NetApp:

/u03/oradata/<DBNAME>	**Datenbankdateien, Spfiles und Passwordfiles**
/u04/orabackup/<DBNAME>	**Archivelogdateien (FRA)**
/u03/redoA/<DBNAME>	**(Standby-)Redologs, Controlfiles, Broker Files**
/u03/redoB/<DBNAME>	**(Standby-)Redologs, Controlfiles, Broker Files**
/u03/temp/<DBNAME>	**Tempfiles**
/u04/flashback/<DBNAME>	**Flashback Logs (FRA)**
/u04/rmanbackup/<DBNAME>	**RMAN-Backups**

Lokation der Controlfiles für SnapCenter-Backup nicht zwingend auf Volume mit Snapshot notwendig

Automatisches Resize der Volumes möglich

4. NetApp SnapCenter (Backup/Recovery)

Backupkonzept

Tägliche Sicherung der Primär-Datenbanken durch SnapCenter

Stündliche Sicherung der Archivelogs

SnapCenter überträgt Snapshot durch SnapVault auf Secondary Storage im entfernten Rechenzentrum

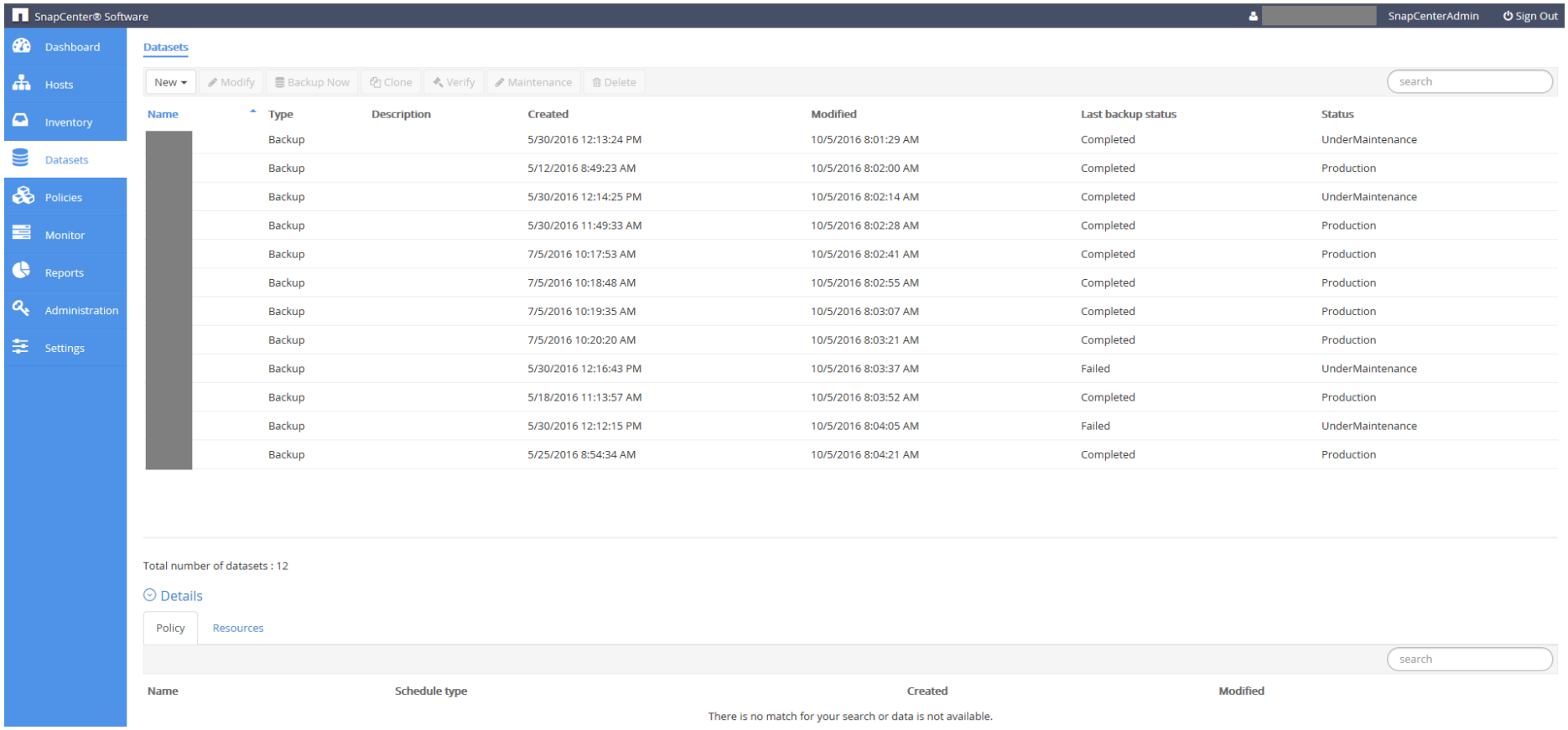
Aufbewahrung auf Primary Storage: 5 Tage, auf Secondary Storage: 35 Tage

Wöchentliche Sicherung der Standby-Datenbanken durch RMAN; Vorteil: Prüfung auf Block Corruption

Wiederherstellung von möglicher Blockcorruption in Primärdatenbank durch Standby-Datenbank möglich. Voraussetzung: Recovery Catalog

SnapCenter

DEVK



Auszug alert.log

DEVK

```
...
Wed Oct 12 00:00:15 2016
Thread 1 advanced to log sequence 6082 (LGWR switch)
  Current log# 3 seq# 6082 mem# 0: /u03/redoA/<DBNAME>/<DBUNAME>/onlinelog/o1_mf_3__191869111434_.log
  Current log# 3 seq# 6082 mem# 1: /u03/redoB/<DBNAME>/<DBUNAME>/onlinelog/o1_mf_3__191869968275_.log
Wed Oct 12 00:00:15 2016
Archived Log entry 22676 added for thread 1 sequence 6081 ID 0x9e84627 dest 1:
Wed Oct 12 00:00:25 2016
ALTER DATABASE BACKUP CONTROLFILE TO '/u03/oradata/<DBNAME>/<DBUNAME>/datafile/control-bkp-155378523.bkp' REUSE
Completed: ALTER DATABASE BACKUP CONTROLFILE TO '/u03/oradata/<DBNAME>/<DBUNAME>/datafile/control-bkp-155378523.bkp' REUSE
ALTER DATABASE BACKUP CONTROLFILE TO TRACE AS '/u03/oradata/<DBNAME>/<DBUNAME>/datafile/control-bkp-155378523.trc' REUSE
Completed: ALTER DATABASE BACKUP CONTROLFILE TO TRACE AS '/u03/oradata/<DBNAME>/<DBUNAME>/datafile/control-bkp-155378523.trc' REUSE
ALTER DATABASE BEGIN BACKUP
Completed: ALTER DATABASE BEGIN BACKUP
ALTER DATABASE END BACKUP
Completed: ALTER DATABASE END BACKUP
Wed Oct 12 00:00:29 2016
ALTER SYSTEM ARCHIVE LOG
...
```

Beispiel Point in Time Recovery

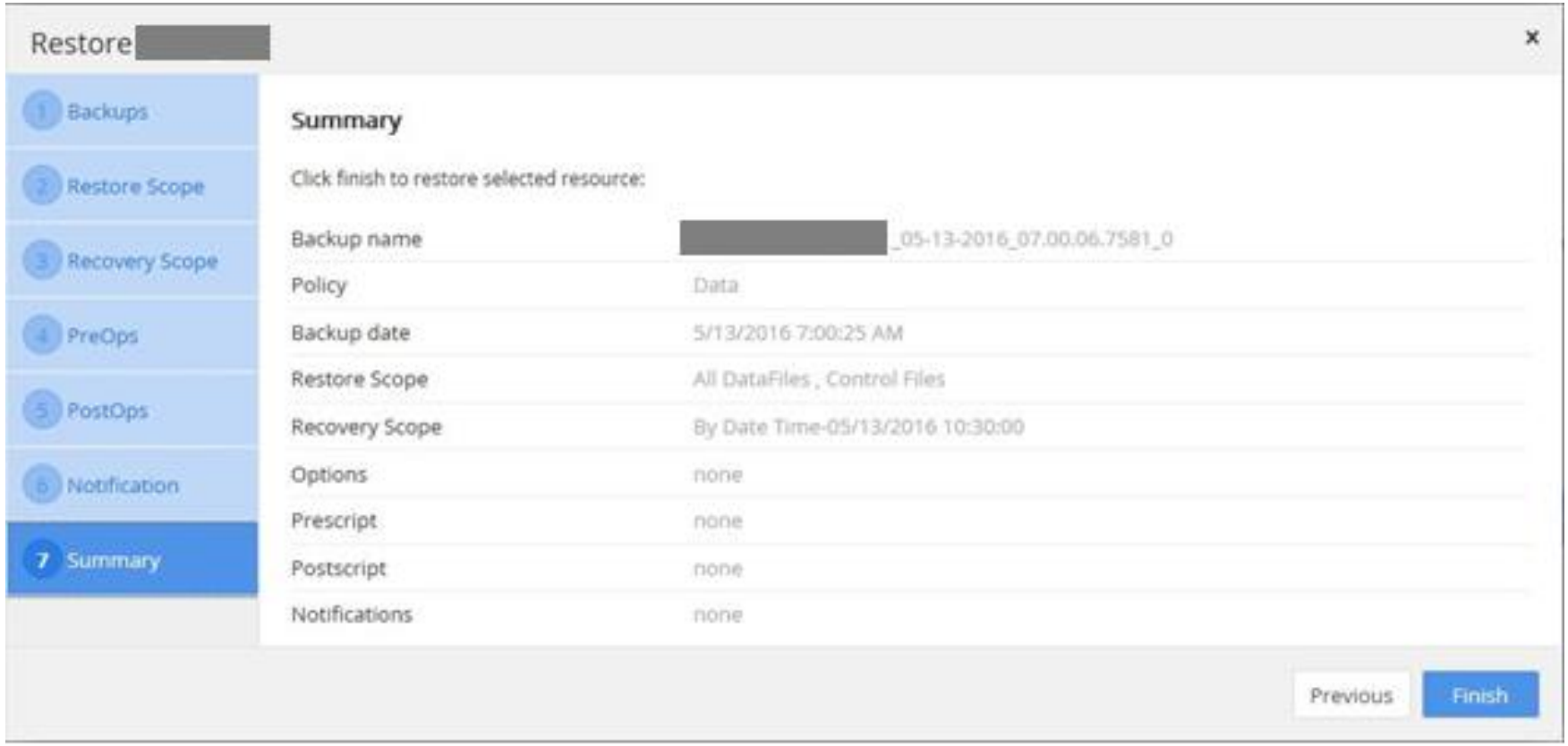
In die Tabelle pittest werden Datensätze eingefügt (ca. alle 10 Sekunden).

```
139 13.05.2016 10:28:48 <INSTANCE2>  147000
152 13.05.2016 10:28:58 <INSTANCE1>  147100
140 13.05.2016 10:29:08 <INSTANCE2>  147200
153 13.05.2016 10:29:19 <INSTANCE1>  147300
161 13.05.2016 10:29:29 <INSTANCE2>  147400
154 13.05.2016 10:29:39 <INSTANCE1>  147500
162 13.05.2016 10:29:50 <INSTANCE2>  147600
155 13.05.2016 10:30:00 <INSTANCE1>  147700
163 13.05.2016 10:30:10 <INSTANCE2>  147800
156 13.05.2016 10:30:21 <INSTANCE1>  147900
164 13.05.2016 10:30:31 <INSTANCE2>  148000
157 13.05.2016 10:30:41 <INSTANCE1>  148100
```

srvctl stop database -db <DBUNAME>

Jetzt wird Point-In-Time Recovery mit Snapcenter auf 10:30:00 durchgeführt

Beispiel Point in Time Recovery



Beispiel Point in Time Recovery

DEVK

Anschließend wird die Datenbank mit dem Befehl „OPEN RESETLOGS“ geöffnet.

```
oracle@<SERVERNAME>[<INSTANCE1>]% srvctl start database -db <DBUNAME> -startoption mount
oracle@<SERVERNAME>[<INSTANCE1>]% sqlplus / as sysdba

SQL> select open_mode from v$database;

OPEN_MODE
--------------------
MOUNTED

SQL> alter database open resetlogs;

Database altered.
```

Ergebnis wie erwartet:

```
SQL> SELECT * FROM pittest WHERE uhrzeit = (SELECT max(uhrzeit) FROM pittest)

        ID UHRZEIT              INSTANZ              WERT
---------- -------------------- -------------------- --------------------
       162 13.05.2016 10:29:50 PD012                147600
```

Hinweise zu SnapCenter Plugin 1.1

DEVK

Restore/Recover auch von großen Datenbanken problemlos und schnell möglich

Clonen von Datenbanken in wenigen Minuten => SnapCenter erledigt hier alle Hintergrundarbeiten

„Inventory“ aller Datenbanken richtet sich nach /etc/oratab

Sichern der Standby-Datenbank nicht „sauber“ möglich weil DB im permanenten Recovery-Modus

Manuelle Konfiguration nach Switch-/Failover

Löschen der Archivelogs bei Angabe von log_archive_dest_1 = location=USE_DB_RECOVERY_FILE_DEST funktioniert nicht (Bug)

Bereinigung der FRA im RMAN definiert:

```
CONFIGURE ARCHIVELOG DELETION POLICY TO APPLIED ON ALL STANDBY;
```

5. Konfiguration Data Guard

Data Gaurd-Konfiguration

DEVK

Beide Rechenzentren sind gleichwertig

db_unique_name kennzeichnet die RZ-Zugehörigzeit

Einsatz von Oracle Managed Files => lange Pfadnamen

```
/u03/oradata/<DBNAME>/<DBUNAME>/datafile/o1_mf_system__189437543178_.dbf
```

Empfehlung vor Aufbau Standby-Datenbank:

- **Standby-Redologs erstellen**
- **Connect von beiden Seiten über OracleNet testen**

Erstellen der physikalischen Standby-Datenbank mit RMAN duplicate

Datenschutzmodus: Maximum Availability (SET PROPERTY 'NetTimeout'=60)

Aufbau Standby-Datenbank mit RMAN

```
connect target sys/<PASSWORD>@<DBNAME>_RZ
connect auxiliary sys/<PASSWORD>@<DBNAME>_<rrhostname1>
run{
allocate channel prmy1 type disk;
allocate channel prmy2 type disk;
allocate auxiliary channel stby1 type disk;
allocate auxiliary channel stby2 type disk;
DUPLICATE TARGET DATABASE
  FOR STANDBY
  FROM ACTIVE DATABASE
  DORECOVER
  SPFILE
  PARAMETER_VALUE_CONVERT '<DBNAME>_RZ','<DBNAME>_RR','<NAMEPARTrz>"','<NAMEPARTrr>'
  SET "db_unique_name"="<DBNAME>_RR"
  SET "instance_number"="1"
  NOFILENAMECHECK
  ;
}
```

Aktivierung von Flashback wegen Reinstate

Muss auf der Standby-Datenbank explizit eingeschaltet werden

6. Erfahrungen & Praxis-Tipps

Definierte Abnahmetests pro Umgebung

Abnahmetests der jeweilgen Stages vor Freigabe:

- **Voraussetzung für Installation erfüllt (cluvfy)**
- **Prüfen BEGIN/END BACKUP bei Snapshot**
- **Ausfall Fabric Interconnect**
- **NetApp Filer Takeover**
- **Point in Time Recovery**
- **Ausfall Oracle Interconnect**
- **Ausfall RAC-Knoten**
- **Verlust Oracle Binaries**
- **Block Corruption**
- **Data Guard Switch- und Failover**
- **Wiederherstellung einer Tabelle**
- **Erstellung RMAN-Backup**

Beispiel Verlust Oracle Binaries

DEVK

Löschen der Oracle Binaries

```
<SERVERNAME>:/u01/app # rm -rf oracle
<SERVERNAME>:/u01/app # ls -l
total 16
drwxrwxrwx 10 root root     4096 May 11 06:00 .snapshot
drwxr-xr-x  3 root oinstall 4096 May  4 08:53 grid
drwxr-xr-x  9 grid oinstall 4096 May  4 17:08 gridbase
drwxrwx---  6 grid oinstall 4096 May  4 17:08 oraInventory
```

Instance-Crash und Service-Schwenk auf verbleibenden Knoten

Recovery der Binaries

```
<SERVERNAME>:/u01/app/.snapshot/8hours.2016-05-11_0600 # ls -l
total 24
drwxr-xr-x  3 root   oinstall 4096 May  4 08:53 grid
drwxr-xr-x  9 grid   oinstall 4096 May  4 17:08 gridbase
drwxrwx---  6 grid   oinstall 4096 May  4 17:08 oraInventory
drwxrwxr-x  8 oracle oinstall 4096 May  9 13:29 oracle
<SERVERNAME>:/u01/app/.snapshot/8hours.2016-05-11_0600 # tar cf - oracle |( cd /u01/app; tar xpf - )
```

Verzeichnis /u01/app/oracle mit korrekten Berechtigungen wiederhergestellt

Vorgenommene Änderungen

DEVK

Ursprünglich lag dritte Voting-Disk und OCR im jeweils anderen RZ

Später wurde Volume jeweils in das lokale RZ verlegt

Installation grid-Software entgegen OFA-Empfehlungen nach „/u01/app/grid/product/12.1.0.2/gridhome“

Grid-Base nach „/u01/app/gridbase“

Dadurch wurde Grid Infrastructure Management Repository (GIMR) in „/u01/app/oracle“ erstellt

Daher ORACLE_BASE in .profile des grid-Users gesetzt

In Test-/Entwicklungsumgebung urprünglich nur eine Single-Server-Installation

Dadurch Probleme bei Sequence-Reihenfolge in den folgenden Stages

Geplant ist weiterer RAC-Knoten in RZ1

Einspielen PSU Juli 2016

DEVK

Durchführung erfolgt in fünf Schritten:

1. **Installation der Software (GI und DB) auf der Standby-Seite RR**
2. **Switchover der Datenbanken auf die Standby-Seite RR**
3. **Installation der Software (GI und DB) auf der ursprünglichen Primär-Seite RZ**
4. **Switchover der Datenbanken auf die ursprüngliche Seite RZ**
5. **Patchen der Datenbanken (Datapatch)**

Datenbanken laufen bis auf Switchover unterbrechungsfrei

Tipp: Wenn kein OJVM in Datenbank installiert ist

```
$ORACLE_HOME/OPatch/datapatch -apply 23054246
```

Monitoring

Monitoring aktuell ausschließlich über Oracle Enterprise Manager 13c:

Wichtig ist Überwachung der TEMP-File-Location bei Verwendung von

```
db_create_file_dest=/u03/oradata/<DBNAME>
```

U.a. wird auch überwacht:

- **Flashback aktiv?**
- **Einträge der View dba_feature_usage_statistics**

Zukünftig Überwachung zusätzlich über Check_MK

7. Fazit

Fazit

Installation und Konfiguration aller Umgebungen der neuen Oracle-Infrastruktur dauerte in etwa ½ Jahr:

- **Strenge Abnahmetests in allen Umgebungen durchgeführt**
- **Anfängliche Probleme bei SnapCenter**
- **Neu eingerichtetes Monitoring über OEM und vorheriges Upgrade auf 13c**

Zielsetzung einer hochverfügbaren und stabilen Infrastruktur wurden erreicht

Failover im Rahmen einer Stromabschaltung durchgeführt => Reinstate der Datenbanken

Jul PSU konnte ohne Downtime installiert werden. Ausnahme: OJVM-Datenbank

Oct PSU non standby First Apply

SnapCenter arbeitet zuverlässig, neues Release 2.0 steht bevor

DANKE für Ihre Aufmerksamkeit

Artikel in RedStack-Magazin „Aufbau einer 12c RAC & Data Guard Umgebung mit NFS-Storage bei der DEVK“

Tim Hensel

DEVK Versicherungen

Competence Center JAVA Entwicklungen

Infrastruktur JAVA-Systeme

Riehler Straße 190

50735 Köln

Telefon: +49 221 757-1473

Fax: +49 221 757-391473

E-Mail: tim.hensel@devk.de

Internet: www.devk.de

DEVK

GESAGT. GETAN. GEHOLFEN.