Den ODA-Betrieb sicherstellen: Worst-Case-Szenario mit fünfzigprozentiger Überlebenschance bei RAC

Andrzej Rydzanicz, OPITZ CONSULTING Polska Sp.z.o.o

In diesem Artikel teilt der Autor seine langjährigen Erfahrungen im Umfeld der Oracle Database Appliance (ODA) und stellt die Prozedur vor, die für die reibungslose Wiederherstellung von nur einem ODA-Knoten erforderlich ist. Hinzu kommt ein Blick auf die Gründe für diese Prozedur und ihre Vorteile.

Im Oracle-Umfeld sind Restores an und für sich nicht ungewöhnlich. Verschiedene Probleme wie Block Corruption oder User-Fehler können dazu führen, dass die Daten zurückgespielt werden müssen. Zudem können Hardware-Probleme auftreten, die einen sauberen Start des Datenbank-Servers verhindern.

Die ODA unterscheidet sich in dieser Hinsicht nicht von anderen Maschinen. Im Falle eines Worst-Case-Szenarios bietet sie eine gewisse Unfallsicherheit, etwa mithilfe von RAC, weil sie aus zwei physikalischen Servern besteht. So weit gut und schön, aber was soll man machen, wenn die ODA nichtsdestotrotz Soft- oder Hardware-technisch nicht funktioniert?

Nun ja, vorausgesetzt es gibt ein brauchbares RMAN-Backup auf einem externen Speicher, lassen sich die Daten mit fast hundertprozentiger Sicherheit auf einem anderen Server oder auf dem ursprünglichen Server, also dort wo eine Neu-Installation stattgefunden hat, wiederherstellen.

Szenarios, bei denen die ganze ODA neu aufgesetzt werden muss, sind im Internet gut dokumentiert. Es gibt im Oracle-Support-Portal viele MOS-Einträge, die einen Bare-Metal-Restore der ODA detailliert beschreiben. Die MOS erläutern allerdings

immer einen Bare-Metal-Restore, der mit beiden ODA-Knoten zu tun hat. Was aber ist zu tun, wenn nur ein Knoten defekt ist?

Der Autor erinnert sich noch gut an zwei Fälle, in denen die End-User nicht sicher waren, dass einer der Datenbank-Knoten komplett defekt war. Da man es hier mit einem RAC zu tun hat, ist dieser Fall gar nicht so ungewöhnlich. Da fragt man sich, warum man den ganzen Cluster neu aufbauen soll, wenn man ebenso gut und transparent nur einen Knoten wiederherstellen und zum Cluster hinzufügen könnte. Das spart immerhin Geld und Zeit, und weil die End-User fast nichts davon merken, feiern sie einen als „Held des Tages" … Es lohnt sich also durchaus, eine solche Restore-Prozedur einmal genauer durchzuspielen. Die Restore-Prozedur für die ODA lässt sich in drei verschiedene Abschnitte unterteilen:

- Aufräumen der Konfiguration in Bezug auf Oracle Inventory und Cluster Registry vor dem Cluster-Aufbau
- Re-Imaging des defekten Knotens
- Konfiguration und Hinzufügen des neuaufgesetzten Knotens im Cluster

Aufräumen der Konfiguration

Eine Aufgabe beim Aufräumen der Konfiguration ist das Löschen von Informationen zum defekten Knoten im Oracle Repository. Mit „oakcli" lässt es sich leicht herausfinden, welche Homeserver aktuell auf der Maschine installiert sind *(siehe Listing 1)*. Die Löschaufgaben müssen für jeden RDBMS Home durchgeführt werden, in diesem Fall für einen 12c RDBMS Home *(siehe Listing 2)*.

Wie schon erwähnt, sollte die Löschaktion für jeden Homeserver erfolgen, den das „oakcli show dbhomes"-Kommando ausgibt, um einen konsistenten Zustand zu erreichen. Die bis jetzt beschriebenen Schritte bezogen sich zwar lediglich auf das Oracle Repository. Sie wurden natürlich auch mit dem Oracle-User durchgeführt. Im Prinzip müssen wir auch die Cluster-Ressourcen des defekten Knotens als Grid aus der Konfiguration entfernen. Es handelt sich hierbei um die VIP-Komponenten. Dafür fragen wir zunächst die Konfiguration ab und speichern diese für die spätere Verwendung *(siehe Listing 3)*. Danach sind sie zu stoppen und zu löschen *(siehe Listing 4)*. Sobald der VIP gelöscht wurde, lässt sich der Knoten mit „/u01/app/12.1.0.2/grid/bin/crsctl delete node -n odanode1" sicher aus der Konfiguration entfernen.

Generell ist es wichtig, den Cluster vor dem Neuaufbau so zu gestalten, als wäre der defekte Knoten mit den dazugehörigen Ressourcen niemals dagewesen. Ebenso ist es wichtig, die SSH-Informationen des Oracle- und Grid-Users auf dem gebliebenen Knoten mithilfe entsprechender Schlüssel („SH Keys") zu löschen. Die Informationen über den Knoten sind

```
[root@odanode2 ~]# oakcli show dbhomes
Oracle Home Name     Oracle Home version              Home Location
----------------     -------------------              -----------
OraDb11203_home1     11.2.0.3.12(19121548,17592127)   /u01/app/oracle/
product/11.2.0.3/dbhome_1
OraDb12102_home1     12.1.0.2.1(19303936,19392604) /u01/app/oracle/prod-
uct/12.1.0.2/dbhome_1
```

Listing 1

```
export ORACLE_HOME=/u01/app/oracle/product/12.1.0.2/dbhome_1
cd $ORACLE_HOME/oui/bin
./runInstaller -updateNodeList ORACLE_HOME=$ORACLE_HOME "CLUSTER_
NODES=odanode2"
```

Listing 2

```
srvctl config vip -node odanode1
srvctl status vip -node odanode1
```

Listing 3

```
srvctl stop vip -vip odanode1-vip
srvctl remove vip -vip odanode1-vip
```

Listing 4

```
export ORACLE_HOME=/u01/app/12.1.0.2/grid cd $ORACLE_HOME/oui/bin
./runInstaller -updateNodeList ORACLE_HOME=$ORACLE_HOME „CLUSTER_
NODES=odanode2" CRS=TRUE
```

Listing 5

```
System Version  Component Name          Installed Version        Supported Version
--------------  ---------------         -----------------        -----------------
12.1.2.6.0
                Controller_INT          4.230.40-3739            Up-to-date
                Controller_EXT          06.00.02.00              Up-to-date
                Expander                0018                     Up-to-date
                SSD_SHARED {
                [ c2d16,c2d17,c2d18,    A122                     A29A
                c2d19 ]
                [ c2d20,c2d21,c2d22,    A122                     A29A
                c2d23 ]
                          }
                HDD_LOCAL               A720                     Up-to-date
                HDD_SHARED              P554                     Up-to-date
                ILOM                    3.2.4.52 r101649         Up-to-date
                BIOS                    30050100                 Up-to-date
                IPMI                    1.8.12.4                 Up-to-date
                HMP                     2.3.4.0.1                Up-to-date
                OAK                     12.1.2.6.0               Up-to-date
                OL                      6.7                      Up-to-date
                GI_HOME                 12.1.0.2.160119(2194     Up-to-date
                                        8354,21948344)
                DB_HOME                 12.1.0.2.160119(2194     Up-to-date
                                        8354,21948344)
```

Abbildung 1: Auswahl des Patches (1)

Patch Simple Search Results

Filters: Patch Name or Number is 12999313; Edit Search

Table ▾ View ▾ Detach Share Link

Patch Name	Description	Release	Platform (Language)	Recomm	Classificatio	Product	Up	Size	Download Ac
	METAL OS ISO IMAGE (Patch)						ag		
12999313	ORACLE DATABASE APPLIANCE 12.1.2.8.0 BARE METAL OS ISO IMAGE (Patch)	12.1.2.8.0	Linux x86-64 (American English)		General	Oracle Database Appliance Software (More...)	1- yea ag	1.4 GB	Software
12999313	ORACLE DATABASE APPLIANCE 12.1.2.7.0 BARE METAL OS ISO IMAGE (Patch)	12.1.2.7.0	Linux x86-64 (American English)		General	Oracle Database Appliance Software (More...)	2- yea ag	1.4 GB	Software
12999313	ORACLE DATABASE APPLIANCE 12.1.2.6.0 BARE METAL OS ISO IMAGE (Patch)	12.1.2.6.0	Linux x86-64 (American English)		General	Oracle Database Appliance Software (More...)	2- yea ag	1.3 GB	Software
12999313	ORACLE DATABASE APPLIANCE 12.1.2.5.0 BARE METAL OS ISO IMAGE (Patch)	12.1.2.5.0	Linux x86-64 (American English)		General	Oracle Database Appliance Software (More...)	2- yea ag	1.4 GB	Software
12999313	ORACLE DATABASE APPLIANCE 2.6.0.0.0 BARE METAL OS ISO IMAGE (Patch)	2.6.0.0.0	Linux x86-64 (American English)		Not Specified	Oracle Database Appliance Software (More...)	2- yea ag	Not Applicable	Software

Abbildung 2: Auswahl des Patches (2)

in der Cluster Registry weiterhin vorhanden. Als Grid-User sollte man diese Einträge im Repository gänzlich entfernen *(siehe Listing 5)*. Nach diesem Schritt kann begonnen werden, den defekten Knoten neu aufzubauen (Re-Imaging).

Re-Imaging des defekten Knotens

Bei einer Störung wie dieser muss man natürlich zunächst die Server-Hardware wieder zum Laufen bringen. Die defekten Hardware-Teile können vom Support ausgetauscht werden. Der Autor hat allerdings die Erfahrung gemacht, dass die Software-Probleme auf Betriebssystem-Ebene oft weitaus hartnäckiger sind. Dabei hilft letztendlich nur ein Bare-Metal-Restore.

Da wir es hier jedoch mit nur einem defekten Knoten zu tun haben, unterscheidet sich die Prozedur des Bare-Metal-Restore in diesem Fall von einem Restore der gesamten ODA. Die ersten Schritte sind allerdings gleich: Zuerst suchen wir den entsprechenden Patch mit dem OS-Image, um die Maschine zu installieren. Das richtige Image findet sich unter dem Patch 12999313. Dabei immer den Patch für die entsprechende OAK-Version des überlebten Knotens auswählen *(siehe Abbildungen 1 und 2)*.

Sobald der Patch heruntergeladen ist, beginnt die Installation des Knotens *(siehe Abbildungen 3 bis 5)*.

- Einloggen zum Ilom
- Remote Console im Ilom starten
- Einfügen des ISO-Image als Storage
- Der Server startet beim nächsten Boot-Vorgang vom angehängten Image

Nun zu den Unterschieden gegenüber einem Bare-Metal-Restore der ganzen ODA-Maschine: Sobald das Re-Imaging des Knotens abgeschlossen ist, beginnt die Konfiguration von User, SSH, Netzwerk und Cluster. Wichtig ist zu wissen, dass die Maschine nach dem Neuaufsetzen des Knotens über eine saubere Linux-Installation mit sämtlichen OS-Komponenten inklusive Appliance Manager verfügt. Nur OS-User, Netzwerk-Informationen und Multipath-Konfiguration feh-

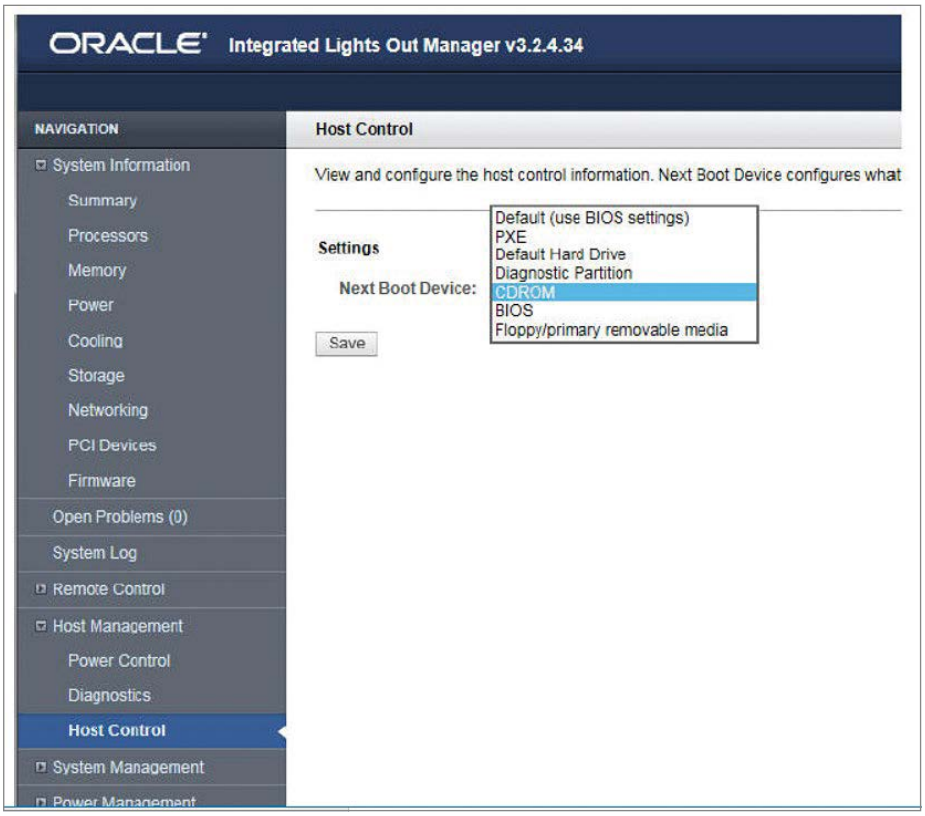

Abbildung 3: Installation des Knotens (1)

len noch und müssen erstellt werden, bevor man den Knoten zum Cluster hinzufügt.

Konfiguration und Hinzufügen des neuaufgesetzten Knotens im Cluster

Sobald die Maschine neu aufgesetzt wurde, ist sicherzustellen, dass sie Netzwerktechnisch zur Verfügung steht, und dafür die Public-IP zu konfigurieren. Dazu dient wie üblich das Kommando „oakcli configure firstnet". Fast alle Schritte, die vorzunehmen sind, um den Knoten zu konfigurieren, basieren auf einem ODA-Skript, das eine Vielzahl von Optionen bietet *(siehe Abbildung 6)*.

Nicht alle Optionen müssen unbedingt ausgeführt sein, aber im Prinzip sind dies die Optionen, die während des Deployments des App-Managers ausgeführt werden.

Um alle Konfigurationsdateien und Binaries kopieren zu können, muss die SSH-Verbindung zwischen beiden Knoten ohne Passwort funktionieren, beziehungsweise das Root-Passwort sollte auf dem funktionierenden Knoten auf „welcome1" zurückgesetzt werden. Die Konfiguration von SSH ist als Punkt 3 in der Ausgabe-Liste der „onecommand"-Parameter-Datei dargestellt *(siehe Abbildung 7)*.

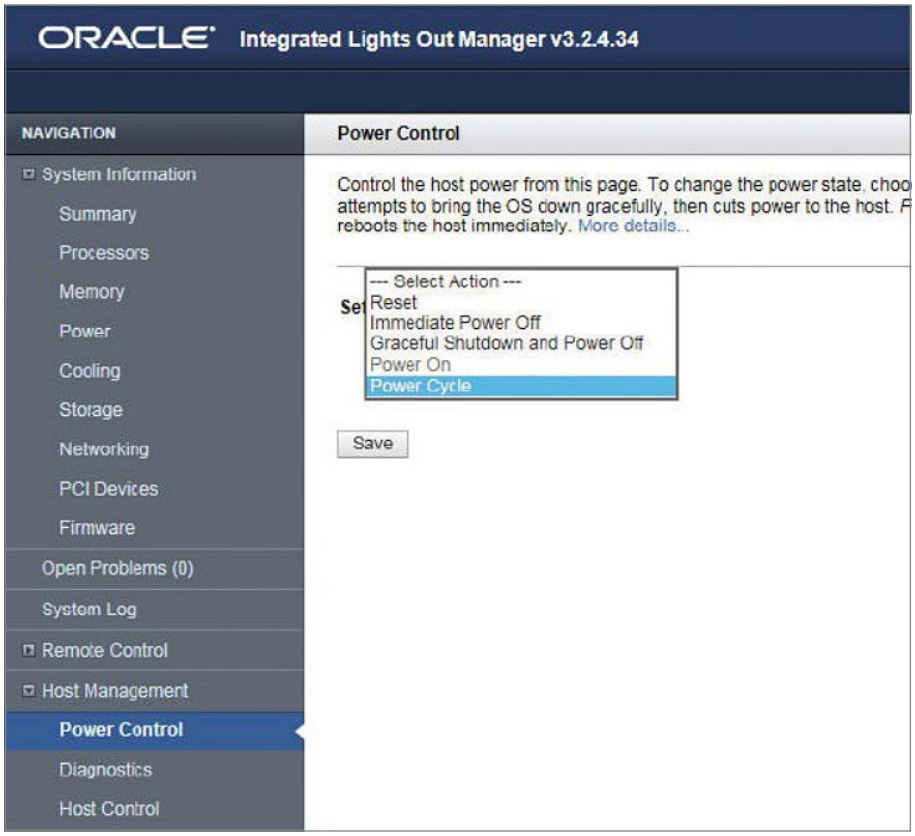

Abbildung 4: Installation des Knotens (2)

Natürlich gibt es auch Bereiche, in denen ein manueller Eingriff notwendig wird. Bei jedem ODA-Deployment generiert der App-Manager eine „onecommand"-Parameter-Datei, in der alle relevanten Informationen wie VIP-Adressen, Scans, Hostnamen oder privates Netzwerk über den Cluster gespeichert sind. Diese Datei muss auf den neu aufgesetzten Knoten kopiert und dort angepasst werden.

Ziel ist es, Einträge, die zu dem funktionierenden Knoten gehören, zu löschen. Die Parameterdatei sollte am Ende nur die relevanten Informationen für den neuen Knoten enthalten. Sobald die Datei fertiggestellt ist, beginnt man, das „GridInst.pl"-Skript mit der entsprechenden Schritt-Option auszuführen beziehungsweise bestimmte Schritte „in einem Rutsch" abzuhandeln („-r"-Parameter, siehe *Listing 6)*. Die Schritte kann man auch einzeln ausführen („-s" Parameter): „./GridInst.pl –s 10".

Damit sind alle wichtigen Tätigkeiten in Bezug auf Netzwerk und OS-User abgeschlossen. Die ursprüngliche Parameter-Datei kann wiederhergestellt werden; VIPs, Hostnamen etc. für beide Knoten sollten jetzt vorhanden sein.

Da zuvor nur das SSH-Protokoll für den Root-User konfiguriert wurde, ist jetzt die Zeit gekommen, alle anderen User (Grid, Oracle) auch mit SSH zu konfigurieren. Die Prompt-Ausgabe mit Hostnamen wird sich nicht ändern, bis die Maschine rebootet ist. Dieser Schritt ist erforder-

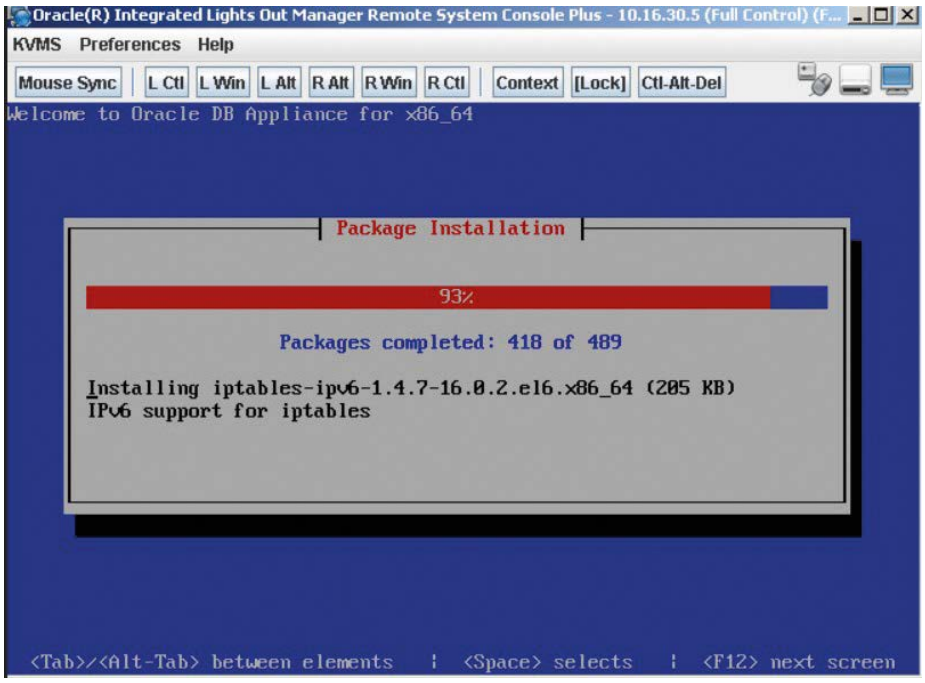

Abbildung 5: Installation des Knotens (3)

lich, um die notwendigen Konfigurationsaufgaben weiter durchführen zu können.

An dieser Stelle ist das System Netzwerk-technisch und OS-mäßig vorbereitet. Das einzige, das jetzt noch fehlt, sind die Storage-Anbindung und die Binaries für GI plus RDBMS. Die Multipath-Konfiguration ist sehr einfach und beschränkt sich auf das Kopieren von „multipath.conf" aus der fehlerfreien Maschine. Hier sind nur die WWIDs der lokalen Platten anzupassen und Multipath zu restarten, damit die Änderungen greifen. Wichtig ist dabei, dass auch die User-Rechte für die Disks angepasst werden *(siehe Listing 7)*.

Wenn spezifische Udev-Rules konfiguriert sind, müssen diese ebenfalls übernommen werden. Darüber hinaus sind unbedingt noch Mount Points für ACFS erforderlich. Hier sollte das Kommando „acfsutil info fs -o mountpoints" für die Ausführung auf dem fehlerfreien Knoten aushelfen.

Damit ist bereits die Basis für die GI- und RDBMS-Binaries geschaffen. Es bleiben nur noch zwei Schritte bis zum voll funktionstüchtigen Cluster. Das Hinzufügen des GI-Homeservers läuft fast vollständig automatisiert ab. Zunächst ist man gezwungen, mit „cluvfy comp peer -refnode odanode2 -n odanode1" ein Pre-Check auszuführen. Im Anschluss besteht die Möglichkeit, mit „cluvfy stage -pre nodeadd -n odanode1 –fixup" ein Fix-up-Skript zu generieren, das auf dem neuen Knoten abläuft. Es führt zur Installation des „cvuqdisk-1.0.9-1"-Pakets.

```
[root@ onecmd]# ./GridInst.pl -l
INFO   : Logging all actions in the file /dev/null and traces in the file /dev/null
INFO   : Loading the configuration file /opt/oracle/oak/onecmd/onecommand.params...
The steps in order are...
Step  0 = ValidateParamFile
Step  1 = SetupNetwork
Step  2 = WriteNodelists
Step  3 = SetupSSHroot
Step  4 = VerifyEnvVersion
Step  5 = SetupDNS
Step  6 = UpdateEtcHosts
Step  7 = SetTimezone
Step  8 = SetupNTP
Step  9 = SetupILOM
Step 10 = ValidateEnv
Step 11 = CreateUsers
Step 12 = SetupStorage
Step 13 = SetupSSHusers
Step 14 = InstallGIClone
Step 15 = RunGIClonePl
Step 16 = RunRootScripts
Step 17 = GIConfigAssists
Step 18 = CreateASMDiskgroups
Step 19 = InstallDBClone
Step 20 = RunDBClonePl
Step 21 = DbcaDB
Step 22 = SetupACFS
Step 23 = SetupASR
Step 24 = ResecureMachine
[root@ onecmd]#
```

Abbildung 6: Die Optionen des ODA-Skripts

Anschließend wird der GI-Homeserver hinzugefügt *(siehe Listing 8)*.

Am Ende muss noch das „root.sh"-Skript ausgeführt werden. Zudem ist es empfehlungswert, den „oakd"-Dämon durchzustarten, sobald der GI installiert ist. Nun ist die Zeit gekommen, endlich den RDBMS-Homeserver hinzuzufügen. Dieser Schritt ist immer als Oracle-User auszuführen *(siehe Listing 9)*.

Wichtig ist, ebenso viele Homeserver hinzuzufügen, wie es in der ursprünglichen Installation gab. Dabei kommt natürlich auch das „root.sh"-Skript zum Einsatz.

Fazit

Im Großen und Ganzen ist die Wiederherstellung von nur einem ODA-Knoten dank RAC für den Endbenutzer schnell und transparent. Der Einsatz beschränkt sich nicht nur auf den Bare-Metal-Restore des Knotens und erfordert daher ein gewisses Know-how in Bezug auf Abhängig-

Original onecommand.params	**onecommand.params File for Single node**

# DB HOST ARRAY INFO 	
HostArr=(odanode1 odanode1-priv0 odanode1-vip **odanode2 odanode2-priv0 odanode2-vip**)
NODE_NAMES=(odanode1 **odanode2**)
PRIV_NAMES=(odanode1-priv0 **odanode2-priv0**)
HA_NAMES=(odanode1-priv1 **odanode2-priv1**)
VIP_NAMES=(odanode1-vip **odanode2-vip**)
 	
# SCAN INFO 	
SCAN_NAME=odanode1-scan
SCAN_IPS=(100.24.48.56 100.24.48.57)
SCAN_PORT=1521
DNS_SERVERS=(192.13.82.180 192.13.82.188 14.20.190.73)
NTP_SERVERS=( )
NODE_IPS=(100.24.48.12 **100.24.48.13**)
VIP_IPS=(100.24.48.28 **100.24.48.29**)
PRIV_IPS=(192.168.16.24 **192.168.16.25**)
HA_IPS= (192.168.17.24 **192.168.17.25**)
 	
# NETWORK INFO 	
NET0_IPS=(100.24.48.12 **100.24.48.13**)
NET0_NETMASK=255.255.248.0
NET0_GATEWAY=100.24.48.1
NET0_NAME=bond0
NET0_HOSTNAME=(odanode1 **odanode2)**
	# DB HOST ARRAY INFO
	HostArr=(odanode1 slcac451-priv0 odanode1-vip)
	NODE_NAMES=(odanode1)
	PRIV_NAMES=(odanode1-priv0)
	HA_NAMES=(odanode1-priv1)
	VIP_NAMES=(odanode1-vip)
	
	# SCAN INFO
	SCAN_NAME=odanode1-scan
	SCAN_IPS=(100.24.48.56 100.24.48.57)
	SCAN_PORT=1521
	DNS_SERVERS=(192.13.82.180 192.13.82.188 14.20.190.73)
	NTP_SERVERS=( )
	NODE_IPS=(100.24.48.12)
	VIP_IPS=(100.24.48.28)
	PRIV_IPS=(192.168.16.24)
	HA_IPS= (192.168.17.24)
	
	# NETWORK INFO
	NET0_IPS=(100.24.48.12)
	NET0_NETMASK=255.255.248.0
	NET0_GATEWAY=100.24.48.1
	NET0_NAME=bond0
	NET0_HOSTNAME=(odanode1)

Abbildung 7: Ausgabe-Liste der „onecommand"-Parameter-Datei

```
cd /opt/oracle/oak/onecmd
./GridInst.pl -r 1-11
```

Listing 6

```
chown grid:asmadmin /dev/mapper/HDD*p*
chown grid:asmadmin /dev/mapper/SSD*p*
chmod 660 /dev/mapper/HDD*p*
chmod 660 /dev/mapper/SSD*p*
```

Listing 7

```
cd $ORACLE_HOME/addnode ./addnode.sh -silent CLUSTER_NEW_
NODES={odanode1} CLUSTER_NEW_VIRTUAL_HOSTNAMES={odanode1-vip}
```

Listing 8

```
[oracle@odanode2 ~]$ cd /u01/app/oracle/product/12.1.0.2/dbhome_1/
addnode/ [oracle@odanode2 addnode]$ ./addnode.sh -silent CLUSTER_NEW_
NODES={odanode1}
```

Listing 9

keiten und Installationsvorgänge, die auf der ODA automatisiert ablaufen.

Die ODA ist noch nicht reif genug, um mit einem Klick alles wiederherzustellen, der menschliche Faktor spielt daher bei Wartung und Betrieb immer noch eine sehr große Rolle. Ein gesundes Basiswissen zu dem, was während des ODA-Deployments unter der Haube passiert, ist also nicht zu unterschätzen und verschafft uns einen anderen Blickwinkel auf die bunte ODA-Welt mit Tools wie dem App-Manager.

Quellen

[1] NOTE 1352884.1: „opatch auto“ or „root-crs.pl -unlock“ or „roothas.pl -unlock“ or „addNode.sh“, Hangs Due to Many Audit Files in „GRID_HOME“

[2] NOTE 1526405.1: „addNode.sh“-Fail PRCF-2023, The following contents are not transferred as they are non-readable

[3] NOTE 2027830.1: ODA HA (High Availability) Deployment Step Descriptions, A List of Deployment Version Specific Steps Used for Each ODA HA Version Using GridInst.pl

[4] NOTE 1373599.1: ODA Oracle Database Appliance Bare Metal Restore Procedure X5-2 and X6-2

[5] NOTE 888888.1: Oracle Database Appliance, 18.1, 12.X, and 2.X Supported ODA Versions & Known Issues (Doc ID 888888.1)

Andrzej Rydzanicz
andrzej.rydzanicz@opitz-consulting.com

Continuous Integration in der Datenbank-Entwicklung

Dominic Weiser, ISTEC Industrielle Software-Technik GmbH

Betrachtet man die aktuellen Arbeitsweisen in der Datenbank-Entwicklung, stellt man sehr schnell fest, dass die agile Vorgehensweise schon ihren Einzug gehalten hat. Ständig wechselnde und sich an die Aufgaben anpassende Entwicklungsteams sind eine Ausprägung dieser Vorgehensweise.

In den seltensten Fällen sind Projekte und Aufgaben so simpel, dass sie von einem Entwickler zu jeder Zeit vollumfänglich überschaut werden können. An dieser Stelle hilft Continuous Integration dabei, negative Nebeneffekte auf fremdem oder eigenem Code frühzeitig aufzuspüren und zu beseitigen. Dabei ist Continuous Integration bestimmt kein Novum mehr, jedoch hat es den flächendeckenden Einzug in die Datenbank-Entwicklung noch nicht wirklich geschafft, obwohl bereits alle technologischen Hilfsmittel in Fülle am Markt verfügbar sind. Dieser Artikel beschreibt die Grundlagen von Continuous Integration und zeigt eine bei-