

Klonen von Oracle Homes bei Oracle RAC 11.2 im Rahmen eines Linux-Upgrades auf ein neues Major-Release

**Andreas Becker
Oracle B.V. & Co. KG
Altrottstr. 31, 69190 Walldorf**

Schlüsselworte

SAP, Upgrade, Oracle Home Cloning, Oracle Grid Infrastructure, Oracle Database 11g, Linux Major Release Upgrade, Oracle Linux, OL5, OL6, RHEL5, RHEL6, addNode.sh, crsctl

Einleitung

Gegeben ist die Situation eines SAP-Kunden, der ein SAP-System mit Oracle-RAC Release 11.2.0.3 betreibt, wobei die Datenbank-Knoten auf Oracle Linux Version 5.8 laufen. Nun soll die Linux-Version bei allen RAC-Knoten auf Linux Version 6.1 angehoben werden.

Für einen solchen Major-Release-Upgrade ist eine komplette Betriebssystemneuinstallation des Servers notwendig, da von Linux-Seite kein Major-Release-Upgrade unterstützt wird. Infolgedessen ist nach der Neuinstallation des Betriebssystems auch die komplette Oracle-Software – Grid Infrastruktur und Datenbanksoftware - neu zu installieren.

Im folgenden werden zwei mögliche Vorgehensweisen vorgestellt, wie eine solche Neuinstallation aller RAC-Knoten (OS, GI, DB) prinzipiell durchgeführt werden kann. Im ersten Verfahren werden die neu zu installierenden Knoten der Reihe nach aus dem RAC-Cluster entfernt, neu installiert und danach wieder in das RAC-Cluster eingehängt. Beim zweiten Verfahren werden die Oracle-Prozesse (GI, RDBMS) der neu zu installierenden Knoten nur gestoppt, die RAC-Knoten verbleiben jedoch im Cluster registriert. Die Unterschiede, Vor- und Nachteile der beiden Verfahren werden im folgenden erläutert.

In diesem Dokument wird beispielhaft von einem 4-Knoten-RAC-System ausgegangen. Wie in der realen Kundensituation soll dabei gelten, daß die Last des SAP-Systems für eine gewisse Zeit auch mit der Hälfte der RAC-Knoten bewältigt werden kann, sodaß die Verfügbarkeit der RAC-Datenbank und damit des SAP-Systems während der gesamten Upgrade-Phase gewährleistet ist.

Major-Release Upgrades bei Linux

Bei einem Major-Release-Wechsel des aktuell installierten Linux-Betriebssystems, z.B. von RHEL5 nach RHEL6 bzw. OL5 nach OL6 ist ein In-place-Upgrade des Betriebssystems nach Herstellerangaben offiziell nicht unterstützt
(vgl. http://docs.oracle.com/cd/E37670_01/E39522/html/ol_sup_64rn.html,
https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Installation_Guide/ch-upgrade-x86.html)

Chapter 37. Upgrading Your Current System



Red Hat does not support upgrades from earlier major versions

Red Hat does not support in-place upgrades between any major versions of Red Hat Enterprise Linux. A major version is denoted by a whole number version change. For example, Red Hat Enterprise Linux 5 and Red Hat Enterprise Linux 6 are both major versions of Red Hat Enterprise Linux.

In-place upgrades across major releases do not preserve all system settings, services or custom configurations. Consequently, Red Hat strongly recommends fresh installations when upgrading from one major version to another.

Abb. 1: Support von Major Release Upgrades bei RHEL, https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Installation_Guide/ch-upgrade-x86.html

3.1. Supported Upgrade Paths

Upgrading from Oracle Linux 6 GA, Update 1, Update 2, or Update 3 is supported. Upgrading from a beta release is not supported.

In-place upgrading from a major version of Oracle Linux 5 or earlier is not supported. Although Anaconda provides an option to perform an upgrade, fresh installation is strongly recommended.

If you have an Oracle Linux 5.8 system, you can use new features in Unbreakable Enterprise Kernel without upgrading to Oracle Linux 6 as Oracle Linux 5.8 includes the Unbreakable Enterprise Kernel.

Abb.2: Support von Major Release Upgrades bei Oracle Linux, http://docs.oracle.com/cd/E37670_01/E39522/html/ol_sup_64rn.html

Bei einem Linux-Upgrade mit Major-Release-Wechsel müssen daher alle RAC-Knoten von Grund auf neu installiert werden. Damit sich die neu installierten RAC-Knoten wieder in das RAC-System integrieren lassen, müssen die ursprünglichen Konfigurationseinstellungen incl. Oracle Homes wieder hergestellt werden.

Im folgenden werden nun zwei Verfahren skizziert.

Verfahren #1: Entfernen und Wiedereinfügen von Clusterknoten (delNode/addNode)

Bei diesem Verfahren werden die RAC-Knoten aus dem RAC-Verbund entfernt, bevor sie mit der neuen Betriebssystemversion installiert werden.

Eine Vorgehensweise besteht darin, die einzelnen RAC-Knoten der Reihe nach aus dem Cluster zu entfernen (delNode), den Knoten mit der neuen Linux-Version neu zu installieren und danach wieder in das Cluster einzuhängen (addNode).

„delNode“ bezeichnet im folgenden die jeweiligen Schritte zum Entfernen eines RAC-Knotens aus einem RAC-Cluster. Analog dazu bezeichnet „addNode“ die Schritte zum Einfügen eines RAC-Knotens zu einem bestehenden RAC-Cluster.

Wie ein RAC-Knoten aus dem Cluster entfernt wird (delNode), ist in MOS [ID 1262925.1] bzw. [1332451.1] beschrieben. Das Einfügen eines Knotens (addNode) ist in [1332451.1] beschrieben.

Für ein 4-Knoten-RAC-System sieht das beispielsweise so aus:

Ausgangspunkt: Die RAC-Knoten 1,2,3 und 4 bilden das aktive RAC (1,2,3,4).

Jetzt werden die Knoten (1,2) aus dem RAC-System entfernt (delnode). Die verbleibenden Knoten (3,4) bilden vorübergehend das aktive RAC-System. Nach der Neuinstallation werden die Knoten (1,2) wieder zum RAC-System hinzugefügt (addnode). Nun fährt das RAC-System für eine gewisse Zeit wieder mit allen RAC-Knoten (1,2,3,4), wobei die Knoten (1,2) mit dem neuen Betriebssystem laufen, (3,4) noch mit der alten Betriebssystemversion. Das Verfahren wird fortgesetzt, in dem man nun die verbleibenden Knoten (3,4) aus dem RAC-Verbund entfernt. Das aktive RAC läuft nun auf Knoten (1,2). Nach der Installation der neuen Betriebssystemversion werden die Knoten (3,4) wieder in den RAC-Verbund eingliedert (1,2,3,4).

Die folgende Übersicht zeigt den zeitlichen Verlauf. Das RAC-Cluster besteht aus 4 Knoten (1,2,3,4). Fett **(1,2)** bedeutet, daß der betreffende RAC-Knoten mit dem neuen Betriebssystem läuft.

(1,2,3,4) – Alle RAC-Knoten aktiv (Linux 5.8)
(3,4) – Hälfte der RAC-Knoten aktiv (Linux 5.8)
(1,2,3,4) – Alle RAC-Knoten aktiv (1+2 Linux 6.1, 3+4 Linux 5.8)
(1,2) – Hälfte der RAC-Knoten aktiv (Linux 6.1)
(1,2,3,4) – Alle RAC-Knoten aktiv (Linux 6.1)

Wenn die Hälfte der Knoten für die Systemlast nicht ausreicht, kann man jeden Knoten einzeln entfernen und nach dem OS-Upgrade wieder einhängen. Das Verfahren wird dadurch jedoch noch länger:

(1,2,3,4)
(2,3,4)
(1,2,3,4)
(1, 3,4)
(1,2,3,4)
(1,2, ,4)
(1,2,3,4)
(1,2,3,)
(1,2,3,4)

Grundsätzlich kann man auf diese Weise das Ziel, eine Major-Release-Upgrade des Betriebssystems durch Neuinstallation des Betriebssystems auf allen RAC-Knoten erreichen. Die Verfügbarkeit des RAC-Systems ist während der gesamten Zeit mit kleineren Einschränkungen gewährleistet.

Bei diesem Verfahren sollte man sich folgende Punkte vor Augen halten:

- Sowohl die Operation zum Entfernen (delnode) eines Knotens aus einem RAC-System als auch die Operation zum Eingliedern (addnode) eines Knotens in ein RAC-System betreffen in erster Linie die aktiven Knoten des „Rest“-Clusters, der weiterlaufen soll.

- Bei jedem Entfernen bzw. Wiedereingliedern findet eine Rekonfiguration des RAC-Clusters statt. Diese Rekonfiguration bedeutet eine gewisse Belastung und potentielle Destabilisierung des Clusters.
- Dadurch, daß sich die Anzahl registrierten RAC-Knoten ändert, kann sich möglicherweise auch das Verhalten des RAC-Systems insgesamt ändern.
- Eine 100%tige Wiederherstellung der Ausgangskonfiguration ist bei diesem Verfahren nicht garantiert. Zwar sind (addnode) und (delnode) prinzipiell gegensätzliche Operationen. Das bedeutet aber nicht, daß nach dem mehrfachen Ausführen von (delnode/addnode) wieder die Ausgangskonfiguration erreicht werden muß.
- Insgesamt wird bei diesem (delnode/addnode)-Verfahren jeder Knoten einmal aus dem Cluster gelöscht und einmal wieder hinzugefügt, was bei einem n-Knoten-RAC-Cluster insgesamt also 2*n Konfigurationsänderungen bedeutet. Allerdings besteht das Ziel nicht darin, den Cluster umzukonfigurieren, sondern idealerweise möglichst in der Ausgangskonfiguration zu erhalten – ausgenommen den Upgrade des Betriebssystems.

Deshalb wird im folgenden ein Verfahren vorgestellt, welches ohne diese Konfigurationsänderungen auskommt.

Verfahren #2: Neuinstallation des Betriebssystems ohne delNode und addNode

Um den Vorteil gleich vorweg zu nennen: dieses Verfahren kommt ohne delNode- und addNode-Operationen und damit ganz ohne RAC-Konfigurationsänderungen aus.

So wie im ersten Verfahren beschrieben, wird auch hier zunächst festgelegt, in welcher Reihenfolge welche RAC-Knoten aus dem aktiven RAC-System herausgenommen werden. Beispiel: zuerst (1,2), danach (3,4).

Bevor die Oracle-Prozesse auf den für den Upgrade ausgewählten RAC-Knoten gestoppt werden, werden folgende Dateien und Verzeichnisse gesichert:

- Home-Verzeichnis von ‚oracle‘ (Software Owner)
- Datei der Benutzer und Gruppen (/etc/passwd, /etc/group)
- Oracle Inventory Pointer (/etc/oraInst.loc)
- Oracle Homes – GI und RDBMS (tar cvf, siehe SAP Hinweis 1696869)
- Ggf. sind weitere hostspezifische Konfigurationseinstellungen zu sichern, die benötigt werden, damit der Rechner nach dem OS-Upgrade wieder als der gleiche Rechner im Cluster erkannt wird (IP-Konfiguration, /etc/hosts, ...).

Nachdem alle notwendigen Dateien und Verzeichnisse gesichert worden sind, werden die Oracle-Prozesse gestoppt (Datenbank und GI). Die Knoten gehören zwar noch immer zum RAC-Cluster, sind aber temporär nicht erreichbar. Das RAC-Cluster läuft in dieser Zeit wie vorgesehen mit weniger RAC-Knoten.

Jetzt wird die neue Linux-Version installiert und die Rechner wieder soweit konfiguriert, daß sie, von der Linux-Version abgesehen, betriebssystemmäßig im gleichen Zustand wie vorher.

Die vor dem Upgrade erstellte Sicherung der Oracle-Software (GI, RDBMS) wird zurückkopiert, geklont und dabei gegen den neuen Linux-Kernel gelinkt. Daß es möglich ist, in einem RAC-System

RAC-Knoten zu betreiben, deren Oracle-Versionen gegen unterschiedliche Betriebssystemversionen laufen, ist in MOS 220970.1 unter „Rolling Upgrades“ beschrieben.

Jetzt können auf den neu installierten RAC-Knoten die zuvor gestoppten Oracle-Prozesse wieder gestartet werden. Ein Teil der RAC-Knoten fährt nun mit der alten Linuxversion, der andere Teil mit der neuen Linuxversion. Die verbleibenden Knoten, die noch mit der alten Linuxversion laufen, werden auf die gleiche Weise auf die neue Version gebracht. Für eine –nicht allzu lange- Zeit kann das RAC-System aber in dem Mischbetrieb von verschiedenen Linux-Versionen betrieben werden.

Vergleich der Verfahren

Insgesamt sollten beide Verfahren zum Ziel führen und am Ende sollte wieder ein lauffähiges RAC-System vorhanden sein, mit neu installierten RAC-Knoten mit einer neuen Betriebssystemversion. Natürlich sollte das letztlich gewählte Verfahren vorher auf einem Testsystem ausreichend durchgetestet und erprobt sein.

Für das erste Verfahren spricht seine universelle Verwendbarkeit – hier gibt es keine Einschränkungen hinsichtlich Rechnernamen, IP-Adressen und Hardware.

Der große Vorteil des zweiten Verfahrens besteht in der gesicherten Kontinuität der Konfiguration, die nicht verändert wird. Mögliche ungewollte Auswirkungen, wie sie beim ersten Verfahren auftreten könnten, werden so vermieden. Allerdings setzt dieses Verfahren voraus, daß Parameter wie Hostnamen, IP-Adressen, usw. nicht geändert werden.

Letztlich liegt es am Administrator, für welches Verfahren er sich entscheidet.

Referenzen

Oracle® Real Application Clusters Administration and Deployment Guide 11g Release 2 (11.2)

Adding and Deleting Oracle RAC from Nodes on Unix and Linux Systems

http://docs.oracle.com/cd/E11882_01/rac.112/e41960/adddelunix.htm#BEICADHD

Using Cloning to Extend Oracle RAC to Nodes on in the Same Cluster

http://docs.oracle.com/cd/E11882_01/rac.112/e41960/cloneracwithoui.htm#BCGEJGAJ

My Oracle Support Hinweise

- [How to remove/delete a node from Grid Infrastructure Clusterware when the node has failed \(Doc ID 1262925.1\)](#)
- [How to Add Node/Instance or Remove Node/Instance in 10gR2, 11gR1 and 11gR2 Oracle Clusterware and RAC \(Doc ID 1332451.1\)](#)
- [RAC: Frequently Asked Questions \(Doc ID 220970.1\)](#)
- [Minimal downtime patching via cloning 11gR2 ORACLE HOME directories \(Doc ID 1136544.1\)](#) (RAC Installationen)
- [Cloning An Existing Oracle 11g Release 2 \(11.2.0.x\) RDBMS Installation Using OUI \(Doc ID 1221705.1\)](#) (Single Instanz Installationen)

SAP Notes

- [1696869 – Patching of Oracle Homes with Minimal Downtime](#)

Kontaktadresse:

Andreas Becker
Oracle B.V. & Co. KG
Altrottstr. 31
D-69190 Walldorf

E-Mail andreas.becker@oracle.com
Internet www.oracle.com/sap