

Automated RAC setup with Cobbler and VirtualBox

Daniel Westermann
dbi services
CH-2800 Delémont

Schlüsselworte

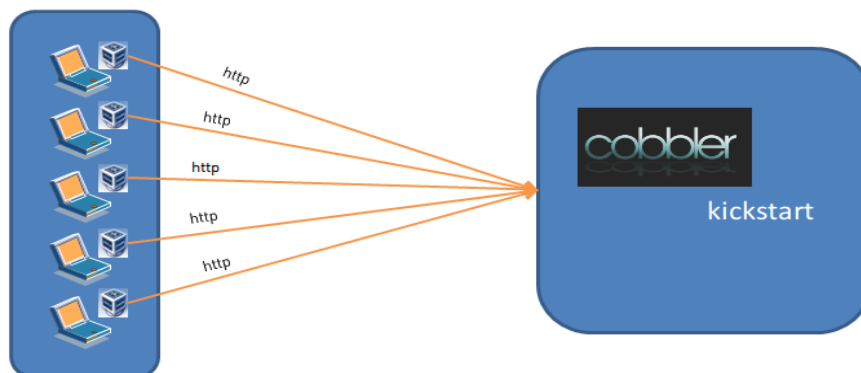
Oracle RAC, Cobbler, VirtualBox, Kickstart, Automation

Einleitung

In der heutigen, immer komplexer werdenden, Oracle Infrastruktur wird es immer wichtiger möglichst schnell und einfach Test Umgebungen zur Verfügung zu haben. Dabei reicht es oftmals nicht auf bestehende Umgebungen zurückzugreifen da spezielle Implementation dort nicht abgebildet sind oder gar nicht abgebildet werden können. Was liegt nun näher als sich die heutigen Möglichkeiten der Virtualisierung zu Nutze zu machen und die Test (oder Schulungs-) Umgebungen dorthin zu holen wo sie am häufigsten gebraucht werden: Auf die Workstation des Mitarbeiters oder Schulungsteilnehmers. Klingt unkonventionell hat aber überzeugende Vorteile: Workstations und Laptops sind heute so leistungsfähig, dass sie problemlos als Test- oder Schulungsumgebungen eingesetzt werden können. Zudem ermöglicht die Entkopplung von zentralen Test Systemen maximale Flexibilität da Anpassungen ohne grossen Aufwand and vor allem ohne Einfluss auf bestehende Umgebungen durchgeführt werden können. Zu guter Letzt kann das hier Vorgestellte auch verwendet werden um Produktionsumgebungen kontrolliert und nach Unternehmensstandards aufzusetzen.

Überlick

Cobbler ist ein Linux Installationsserver und ermöglicht es schnell und standardisiert Installationen über das Netzwerk durchzuführen. Alle Clients, die installiert werden sollen, greifen über http auf den Cobbler Server zu und beziehen von dort ihre Konfiguration und die Installationsquellen. In unserem Setup sind die Clients mit VirtualBox virtualisiert aber die Clients können auch physische Server in einem Produktions Data-Center sein (und sind es in der Realität auch oft).



Um alle Fragen des Installationsvorgangs zu beantworten präsentiert Cobbler den Clients eine Kickstart Datei. Diese Datei spezifiziert unter anderem welche Pakate installiert werden sollen, in welcher Zeitzone sich der Client befindet, wie das Root-Passwort lautet und vieles mehr. Zusätzlich können in Kickstart Dateien Skripte ausgeführt und Konfiguration aller Art vorgenommen werden.

Cobbler und Kickstart

Die Basis in Cobbler sind die Distributionen welche importiert werden müssen. Distributionen sind die ISO-Images der Hersteller (RedHat, CentOS, Oracle Linux, SuSE etc). Jede Distribution die in Cobbler verfügbar ist kann als Basis für Installationen dienen. In diesem Fall ist das Oracle Linux 7.1.

```
$ mount -o loop,ro /var/tmp/[YOUR_ISO].iso /mnt
$ cobbler import --name=[YOUR_DISTRO_NAME] --arch=x86_64 --path=/mnt
```

Importierte Distributionen können jederzeit angezeigt werden:

```
$ cobbler distro list
  OL71-x86_64
```

Die wahre Herausforderung ist die Erstellung der Kickstart Datei. Diverse Vorlagen werden von Cobbler schon mitgeliefert und befinden sich hier:

```
[root@oelstaging ~]# ls /var/lib/cobbler/kickstarts/
dbi_ws_oral2102_gi_rac.ks  ora11204_gi_rac.ks      sample_esx4.ks
dbi_ws_pgforora.ks       ora12102_gi_rac.ks      sample_esxi4.ks
default.ks               ora12102single.ks       sample_esxi5.ks
esxi4-ks.cfg             pg940.ks                 sample.ks
esxi5-ks.cfg             pxerescue.ks            sample_old.seed
install_profiles         sample_autoyast.xml     sample.seed
legacy.ks                sample_end.ks           test.ks
```

Die Kickstart Datei ist auch der Ort an dem die Installation der Oracle Software erledigt wird. Dafür werden die notwendigen Benutzer und Gruppen angelegt und der Oracle Installer im „silent“ Modus aufgerufen:

```
...
/usr/sbin/groupadd -g 54321 oinstall
/usr/sbin/groupadd -g 54322 dba
/usr/sbin/groupadd -g 54323 oper
/usr/sbin/groupadd -g 54324 backupdba
/usr/sbin/groupadd -g 54325 asmdba
/usr/sbin/groupadd -g 54326 dgdba
/usr/sbin/groupadd -g 54327 kmdba
/usr/sbin/groupadd -g 54328 asmadmin
/usr/sbin/groupadd -g 54329 asmoper
/usr/sbin/useradd -u 54322 -g oinstall -G asmadmin,asmdba,dba,asmoper -p
saEd4MopKYPG6 grid
/usr/sbin/useradd -u 54323 -g oinstall -G
dba,asmdba,oper,backupdba,dgdba,kmdba -p saYCGUDOIXesE oracle
...
# oracle grid infrastructure software only installation
/bin/su - grid -c "cd /u01/app/oracle/software/grid; ./runInstaller
INVENTORY_LOCATION=/u01/app/oraInventory SELECTED_LANGUAGES=en
```

```

oracle.install.option=CRS_SWONLY ORACLE_BASE=/u01/app/grid
ORACLE_HOME=/u01/app/12.1.0/grid ORACLE_HOME_NAME=HCRS121020
UNIX_GROUP_NAME=oinstall oracle.install.asm.OSDBA=asmdba
oracle.install.asm.OSOPER=asmoper oracle.install.asm.OSASM=asmadmin
FROM_LOCATION=../stage/products.xml DECLINE_SECURITY_UPDATES=true -
ignoreSysPrereqs -ignorePrereq -waitforcompletion -silent"
/u01/app/oraInventory/orainstRoot.sh
/u01/app/12.1.0/grid/root.sh
/bin/chmod -R 770 /u01/app/oraInventory
# create the inventory pointer
echo "inventory_loc=/u01/app/oraInventory
inst_group=oinstall" > /etc/oraInst.loc
# oracle database software only installation
/bin/su - oracle -c "cd /u01/app/oracle/software/database; ./runInstaller
oracle.install.option=INSTALL_DB_SWONLY ORACLE_BASE=/u01/app/oracle
ORACLE_HOME=/u01/app/oracle/product/12.1.0/db_1_0 ORACLE_HOME_NAME=H121020
UNIX_GROUP_NAME=oinstall oracle.install.db.DBA_GROUP=dba
oracle.install.db.OPER_GROUP=oper oracle.install.db.BACKUPDBA_GROUP=backupdba
oracle.install.db.DGDBA_GROUP=dgdba oracle.install.db.KMDBA_GROUP=kmdba
FROM_LOCATION=../stage/products.xml INVENTORY_LOCATION=/u01/app/oraInventory
SELECTED_LANGUAGES=en oracle.install.db.InstallEdition=EE
DECLINE_SECURITY_UPDATES=true -ignoreSysPrereqs -ignorePrereq -
waitforcompletion -silent"
/u01/app/oracle/product/12.1.0/db_1_0/root.sh
...

```

Sobald die Kickstart Datei verfügbar ist empfiehlt es sich ein sogenanntes Profil zu erstellen welches die Kickstart Datei und die Distribution verbindet:

```

[root@oelstaging ~]# cobbler profile list
  DBI_WS_OWA12102_GI_RAC
  DBI_WS_PG_PGFORORA
  OL71-x86_64
  ora12102_gi_rac
[root@oelstaging ~]# cobbler profile report --name=ora12102_gi_rac
Name                                     : ora12102_gi_rac
TFTP Boot Files                         : {}
Comment                                 :
DHCP Tag                                : default
Distribution                             : OL71-x86_64
Enable gPXE?                            : 0
Enable PXE Menu?                        : 1
Fetchable Files                         : {}
Kernel Options                          : {}
Kernel Options (Post Install)           : {}
Kickstart                               :
/var/lib/cobbler/kickstarts/ora12102_gi_rac.ks
Kickstart Metadata                      : {}
Management Classes                     : []
Management Parameters                  : <<inherit>>
Name Servers                            : []
Name Servers Search Path                : []
Owners                                   : ['admin']
Parent Profile                           :
Proxy                                    :
Red Hat Management Key                   : <<inherit>>
Red Hat Management Server                : <<inherit>>
Repos                                    : []

```

```
Server Override      : <<inherit>>
Template Files       : {}
Virt Auto Boot       : 1
Virt Bridge          : xenbr0
Virt CPUs            : 1
Virt Disk Driver Type : raw
Virt File Size (GB)  : 5
Virt Path            :
Virt RAM (MB)        : 512
Virt Type            : xenpv
```

Somit können andere Systeme, die ähnlich aufgebaut werden sollen, das gleiche Profil verwenden.

Systeme erstellen und konfigurieren

Hat man die Distribution, die Kickstart Datei und das Profil parat können die Systeme in Cobbler erstellt und konfiguriert werden. Die Cobbler Kommandas dafür sind:

```
# add the system to cobbler
cobbler system add --name=${SYSTEM_NAME} --profile=${PROFILE}
# add the public interface and hostname
cobbler system edit --name=${SYSTEM_NAME} --interface=enp0s3 \
    --ip-address=192.168.22.${IPPUBSTART} \
    --netmask=${NETMASK} \
    --hostname=${SYSTEM_NAME} \
    --static=1 --dns-name=${SYSTEM_NAME}.lab.ch
# add the private interface
cobbler system edit --name=${SYSTEM_NAME} --interface=enp0s8 \
    --ip-address=10.1.1.${IPPRIVSTART} \
    --netmask=${NETMASK} \
    --static=1
```

ISOs erzeugen

Sobald Cobbler der Systeme kennt kann für jedes System ein ISO erzeugt werden:

```
cobbler buildiso --systems="${SYSTEM_NAME}" \
    --iso="${BASEDIR}/${SYSTEM_NAME}.iso" 2>&1 > /dev/null
```

Von diesem ISO starten die Clients und wissen dann sofort wie ihre Netzwerkconfiguration aussieht, von wo die Installationsquellen bezogen werden müssen und welche Kickstart Datei verwendet werden soll.

Das Setup Script für die virtuellen Maschinen

Um diese Schritte nun alle zu automatisieren wird für das Erstellen der virtuellen Maschinen ein Script generiert. Dieses Skript verwendet die „vboxmanage“ Kommandos um die Systeme in VirtualBox zu registrieren, zu konfigurieren und das ISO anzuhängen. Zusätzlich wird das Konfigurations-Skript für die Oracle Grid Infrastructure erstellt. All dies wird dann über http zur Verfügung gestellt:

File Name	Date	Size
12102racp1vm1_ORACLE_PRIVATE_SSH	21-Sep-2015 07:20	1.6K
12102racp1vm1_ORACLE_PUBLIC_SSH	21-Sep-2015 07:20	402
12102racp1vm2.iso	21-Sep-2015 07:20	39M
12102racp1vm2_CRSCONFIGPARAMS	21-Sep-2015 07:20	1.5K
12102racp1vm2_GRID_AUTHORIZED_KEYS	21-Sep-2015 07:20	800
12102racp1vm2_GRID_PRIVATE_SSH	21-Sep-2015 07:20	1.6K
12102racp1vm2_GRID_PUBLIC_SSH	21-Sep-2015 07:20	400
12102racp1vm2_HOSTS	21-Sep-2015 07:20	422
12102racp1vm2_KEYSCAN	21-Sep-2015 07:20	96
12102racp1vm2_ORACLE_AUTHORIZED_KEYS	21-Sep-2015 07:20	804
12102racp1vm2_ORACLE_PRIVATE_SSH	21-Sep-2015 07:20	1.6K
12102racp1vm2_ORACLE_PUBLIC_SSH	21-Sep-2015 07:20	402
12102racp2vm1.iso	21-Sep-2015 07:20	39M
12102racp2vm1_CRSCONFIGPARAMS	21-Sep-2015 07:20	1.5K
12102racp2vm1_GRID_AUTHORIZED_KEYS	21-Sep-2015 07:20	800
12102racp2vm1_GRID_PRIVATE_SSH	21-Sep-2015 07:20	1.6K
12102racp2vm1_GRID_PUBLIC_SSH	21-Sep-2015 07:20	400
12102racp2vm1_HOSTS	21-Sep-2015 07:20	422
12102racp2vm1_KEYSCAN	21-Sep-2015 07:20	96
12102racp2vm1_ORACLE_AUTHORIZED_KEYS	21-Sep-2015 07:20	804
12102racp2vm1_ORACLE_PRIVATE_SSH	21-Sep-2015 07:20	1.6K
12102racp2vm1_ORACLE_PUBLIC_SSH	21-Sep-2015 07:20	402
12102racp2vm2.iso	21-Sep-2015 07:20	39M
12102racp2vm2_CRSCONFIGPARAMS	21-Sep-2015 07:20	1.5K
12102racp2vm2_GRID_AUTHORIZED_KEYS	21-Sep-2015 07:20	800
12102racp2vm2_GRID_PRIVATE_SSH	21-Sep-2015 07:20	1.6K
12102racp2vm2_GRID_PUBLIC_SSH	21-Sep-2015 07:20	400
12102racp2vm2_HOSTS	21-Sep-2015 07:20	422
12102racp2vm2_KEYSCAN	21-Sep-2015 07:20	96
12102racp2vm2_ORACLE_AUTHORIZED_KEYS	21-Sep-2015 07:20	804
12102racp2vm2_ORACLE_PRIVATE_SSH	21-Sep-2015 07:20	1.6K
12102racp2vm2_ORACLE_PUBLIC_SSH	21-Sep-2015 07:20	402
setup_p_1.sh	21-Sep-2015 07:20	3.7K
setup_p_2.sh	21-Sep-2015 07:20	3.7K

Je eines der „setup_p*“ Skripte wird auf einen Clients geladen und ausgeführt. Somit können RAC Umgebungen für 1-n Teilnehmer eines Workshops mit 2-n Knoten automatisch erstellt werden, ohne dass manuell eingegriffen werden muss.

Die Konfiguration der RAC Umgebung

Sobald die Systeme einer Umgebung installiert sind wird der letzte Knoten die Konfiguration der Umgebung vornehmen. Es muss deswegen der letzte Knoten sein weil alle anderen zum Zeitpunkt der Konfiguration installiert und bereit sein müssen. Sobald also der letzte Knoten nach seiner Installation startet wird er das Konfigurations-Skript zum Einrichten der Grid Infrastructure und zum Erstellen und Konfigurieren der RAC Datenbank ausführen. Dies ist ebenfalls Bestand der Kickstart Datei:

```

/usr/bin/wget http://${COBBLERSERVER}/finish_ora_12102_gi_rac_project.sh -O
/home/grid/finish_ora_12102_gi_rac_project.sh
/bin/chown grid:oinstall /home/grid/finish_ora_12102_gi_rac_project.sh
/bin/chmod u+x /home/grid/finish_ora_12102_gi_rac_project.sh
/bin/echo "/home/grid/finish_ora_12102_gi_rac_project.sh" >> /etc/rc.local
/bin/chmod u+x /etc/rc.local

```

Nach ca. 40 Minuten (Oracle 11.2.0.4) bzw. einer Stunde (Oracle 12cR1) steht eine zwei Knoten RAC Umgebung bereit. Für eine 3 oder mehr Knoten Umgebung erhöht sich die Zeit des Setups entsprechend.

Fazit

Das hier Vorgestellte ist sicher nicht in aller Munde wie Docker, Cointainer oder Cloud Services, hat aber entscheidende Vorteile: Die Kombination der Vorgestellten Komponenten ermöglicht es Test- oder Schuliungsumgebungen auf die Workstations zu holen und zwar nachvollziehbar, wiederholbar und völlig flexibel. Zudem kann das gleiche Vorgehen für das Erstellen von Produktionsumgebungen im Data-Center angewand werden (physisch und virtuell). Es genügt sogar den Anforderungen von SOX oder GxP gekennzeichneten Umgebungen da alle Schritte in einem Skript zusammengefasst sind.

Kontaktadresse:

Daniel Westermann

dbi services

Rue de la Jeunesse, 2

CH-2800 Delémont

Telefon: +41 (0) 32 422 9600

Fax: +41 (0) 32 422 9615

E-Mail daniel.westermann@dbi-services.com

Internet: www.dbi-services.com