Oracle VM 3 - das neue Kommandozeilen-Interface

Martin Bracher
Trivadis AG
Schweiz

Schlüsselworte
OracleVM, Virtualisierung, Kommandozeile, Script

Einleitung

Lange wurde es angekündigt, nun ist es endlich da, und kaum jemand hat es bisher bemerkt: Das Kommandozeilen-Interface für OracleVM 3.2. Beziehungsweise bei OVM 3.1.1 mit einem Patch hat es undokumentiert schon Einzug gehalten.

Alle, die sich bisher mit dem Web-GUI nicht anfreunden konnten sind hier richtig. Aber auch alle, die ihren Setup gern nachvollziehbar in Form eines Scripts dokumentieren wollen.
Man kann sich dieses Interface vorstellen wie ein Oracle "rman" oder "dgmgrl" Utility. Wenn man die Konzepte von OVM kennt, ist die Syntax leicht verständlich. Im Folgenden wird das OracleVM Kommandozeilen Interface als „ovmcli“ abgekürzt.

In diesem Vortrag werden wir die Architektur, sowie die Möglichkeiten und Einschränkungen dieses Interfaces anschauen: Von der Konfiguration der OVM Umgebung bis zum Erstellen der Virtuellen Maschinen.

Architektur

Um es gleich vorweg zu nehmen: Architektonisch ändert sich nichts. Im Zentrum steht weiterhin der OracleVM Manager, über welchen man die OracleVM Server verwaltet.
Der Manager bringt ein eigenes Verwaltungs-Interface in Form eines Web-Interfaces mit, welches über https://ovmmanager:7002/ovm/console zu erreichen ist („ovmmanager“ ist der Servername, auf welchem wir den OVM Manager installiert haben).
Als alternatives Interface kann man schon seit einiger Zeit den Enterprise Manager Cloud Control 12c verwenden, der ebenfalls den bestehenden OVM Manager benötigt.
Und nun kommt als drittes Interface noch das ovmcli hinzu, das erste offizielle nicht-grafische Interface.

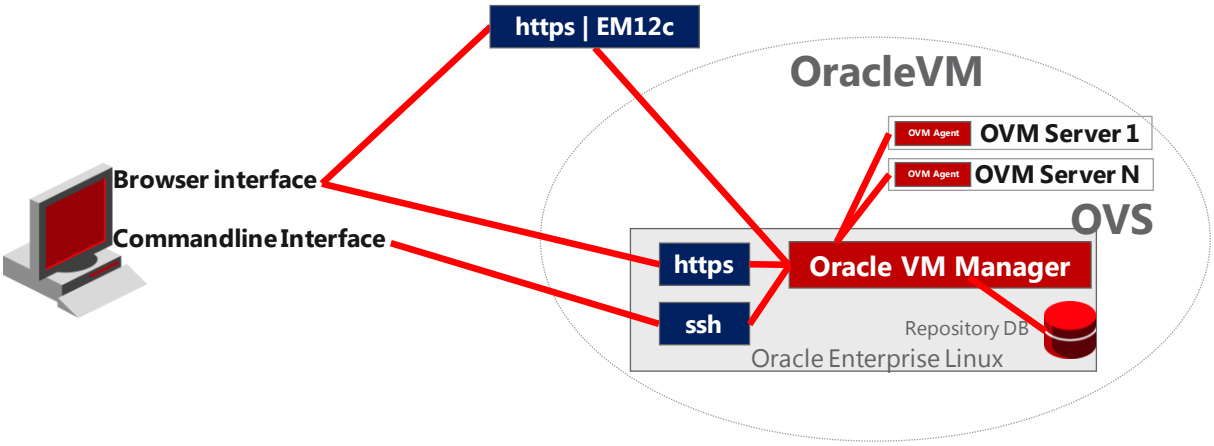

Abb. 1: Architekturübersicht

Im Gegensatz zu anderen Oracle-Tools wie „sqlplus“, „rman“ oder „dgmgrl“, die als eigenständiges Programm geliefert werden und sich über ein proprietäres Protokoll mit dem Ziel verbinden, hat Oracle hier einen anderen Ansatz gewählt. So wie man sich beispielsweise per SSH auf einen Server verbindet und dort in einer Shell wie bash oder ksh arbeitet, verbindet man sich per SSH auf Port 10000 des OVM Managers und bekommt als Shell das OVM Kommandozeilen-Interface (ovmcli). Unser Client ist also ein Terminal-Fenster unter Unix mit einer SSH-Verbindung, oder eine PUTTY-Session unter Windows. Login ist wie beim Web-Interface mit User „admin“ und dem entsprechenden Passwort.

Konfiguration der Umgebung

Unter Windows definiert man sich eine Putty-Session.

Unter Unix/Linux erstellt man sich am Besten einen Alias in der Shell (im .profile beziehungsweise .bash_profile). Das funktioniert auch auf dem OVM Server oder dem OVM Manager selbst.

Standardmässig wird man nach 10 Minuten Inaktivität automatisch ausgeloggt. Um dies zu verhindern, kann man noch die Option ServerAliveInterval mitgeben. Damit sieht der Alias nun so aus:

```
alias ovmcli='ssh -p 10000 –o ServerAliveInterval=40 admin@ovmmanager'
```

Statt sich mit User „admin“ und dem entsprechenden Passwort anzumelden, können wir auch die SSH Authentifizierung mit Public/Private Key verwenden. Dies ist insbesondere notwendig, wenn wir später gewisse Sachen per Script automatisieren möchte.
Sofern wir nicht schon auf unserem Client Schlüssel für SSH erstellt haben, können wir dies nun tun. Bei PUTTY mit dem Programm „puttygen.exe“, bei Unix mit:

```
ssh-keygen
```

Den Befehl können wir noch entsprechend mit Optionen anpassen, um beispielsweise die Schlüssellänge (-b) oder den Algorithmus (-t dsa|rsa) zu spezifizieren. Den Schlüssel selbst können wir durch ein Passwort absichern, oder es auch sein lassen, wenn wir unseren Client als ausreichend gesichert betrachten.
Den öffentlichen Schlüssel (.ssh/id_rsa.pub) müssen wir nun auf den OVM Manager transferieren und unter dem User „oracle“ im File „.ssh/ovmcli_authorized_keys“ (also nicht wie sonst üblich im authorized_keys) anfügen.
Zu beachten ist, dass man sich beim erstmaligen Anmelden trotzdem noch mit Username und Passwort authentifizieren muss. Danach jedoch erfolgt die Authentifizierung über das entsprechende Schlüsselpaar. Wenn jetzt noch eine Passwortabfrage kommt, bezieht sich diese auf das Passwort des Schlüssels. Einen Schlüssel mit Passwort können wir auch dem „ssh-agent“ hinzufügen (mit „ssh-add“), damit wir bis zum Stopp dieses Agenten das Passwort nur einmal angeben müssen. Das Äquivalent bei PUTTY heisst „pageant.exe“ und der Schlüssel wird über ein Dialogfenster, welches sich durch Doppelklick auf das Icon in der Taskleiste öffnet, hinzugefügt.

Kommunikation mit dem ovmcli

Nach dem Einloggen per SSH erhalten wir einen einfachen Prompt:

```
OVM>
```

Nun müssen wir uns mit der Sprache des ovmcli vertraut machen. Wer das grafische Web-Interface

und die verwendeten Begriffe von OracleVM kennt, der wird sich mit dieser Sprache rasch zurecht finden.

Als Einstieg sowie als regelmässiges Nachschlagewerk empfiehlt sich die Dokumentation von Oracle selbst:

http://docs.oracle.com/cd/E35328_01/E35336/html/index.html

Und wer die Grundzüge der Syntax schon kennt und nur einzelne Optionen rasch nachschauen muss, kann auch die im ovmcli integrierte Hilfe-Funktion nutzen.

```
OVM> help
```

Zeigt im Überblick alle verfügbaren Befehle.

```
OVM> create vm ?
```

Zeigt die benötigten (mit *) und optionalen Parameter zum Erstellen einer VM an.

Die wichtigsten Befehle sind:

- list: Anzeigen aller Objekte eines bestimmten Typs, beispielsweise aller virtueller Maschinen (list vm) oder aller Netzwerke (list network). Die Bezeichnung der Objekte bekommen wir mit „list ?“ angezeigt.
- show: Anzeigen der Informationen zu einem bestimmten Object, beispielsweise einer Virtuellen Maschine (show vm name=slot046)
- create: Erstellen eines Objekts, beispielsweise einer virtuellen Disk (create virtualdisk)
- edit: Bearbeiten eines Objekts
- delete: Löschen eines Objekts
- add / remove: Hinzufügen oder Entfernen eines Objekts von einem anderen, beispielsweise eine Virtuelle Netzwerkkarte zu einer Virtuellen Maschine zufügen.

Aufbau einer neuen OracleVM Umgebung

Den Aufbau einer neuen OVM Umgebung können wir vollständig im ovmcli erledigen. Der Ablauf ist weitgehend identisch mit der Web-Variante. Wer also bereits einmal eine Umgebung via Web-Interface aufgebaut hat, kommt auch hier rasch zum Ziel. Beim ersten Mal wird man mit dem ovmcli sicher etwas länger brauchen, als wenn man es wie bisher gewohnt über das Web-Interface macht. Aber wir haben einige nicht zu unterschätzende Vorteile, da wir die verwendeten Befehle speichern können:

Dadurch ist der Setup eindeutig dokumentiert. Sollten wir mal die Umgebung frisch aufbauen müssen (z.B. totaler Hardware-Verlust), können wir die Befehle einfach auf neuer Hardware nochmals laufen lassen. Ebenso können wir durch leichte Anpassung der Befehle (z.B. Hostnamen) diese für den Setup weiterer Umgebungen nutzen.

Nach dem Setup kennt der OVM Manager noch keine OVM Server (welche nach der Installation nur

eine minimale Konfiguration haben). Diese müssen wir zuerst mal inventarisieren lassen

```
discoverServer ipAddress=172.16.64.99 port=8899 username=oracle
 password=AdminOVMM1 takeOwnership=yes
```

Dies dauert einen gewissen Moment. In dieser Zeit wird die Hardware analysiert und die Komponenten werden im OVM Manager registriert. „username=oracle“: Auf dem Server gibt es keinen Useraccount „oracle“. Gemeint ist das Passwort vom Agent, das wir bei der Installation des OVM Servers angegeben haben.

In den weiteren Schritten können wir dann diese Komponenten konfigurieren. Zuerst brauchen wir mal SAN-Storage, um uns einen Serverpool definieren zu können.

```
list physicaldisk
```

zeigt uns die gefundenen Disks (SAN LUN) mit deren Bezeichnungen an. Hinter welchem Namen sich welche LUN versteckt, müssen wir uns anzeigen lassen. Die Disk können wir entweder über den generierten Namen, oder aber über die eindeutige ID anzeigen lassen. Übrigens: bei praktisch jedem Befehl kann man das Objekt mit Name oder ID spezifizieren.

```
show physicaldisk name="DGC (1)"
show physicaldisk id=0004fb000018000024f2952603bef083
```

Wenn wir nun unser 12GB Volume für das Pool-Filesystem gefunden haben, können wir den Serverpool definieren:

```
create serverpool virtualIP=172.16.64.100 clusterEnable=yes
 physicaldisk=0004fb000018000024f2952603bef083 migrateUsingSsl=yes
 name=meinPool
```

Zu einem Serverpool gehört mindestens ein Server, den wir nun hinzufügen können.

```
add server name=ovms01 to serverpool name=meinPool
```

Als Nächstes können wir das Storage Repository erstellen, auf welches die virtuellen Maschinen gespeichert werden. Wiederum suchen wir mit list physicaldisk / show physicaldisk eine passende freie Disk. Mit dieser wird dann das Repository erstellt:

```
create Repository name=meinRepo
 PhysicalDisk=0004fb0000180000c9a1d731dd955569 serverPool=meinPool
```

An welche Server wir dieses Repository präsentieren, bestimmen wir mit folgendem Befehl:

```
add Server name=ovms01 to repository name=meinRepo
```

Wenn wir weitere Server hinzufügen, müssen wir das für jeden neuen Server wiederholen.

Nun ist es an der Zeit, sich der Netzwerkkonfiguration zu widmen. Was alles erkannt wurde, können wir uns mit folgenden Befehlen anzeigen lassen:

```
list network
list port
list bondport
```

Den schon automatisch eingerichteten, aber nur mit einer Netzwerkkarte konfigurierten Bondport 0 können wir nun mit einer weiteren Netzwerkkarte erweitern:

```
add port id=0004fb0000200000489d9d4c5243d394
 to bondport id=0004fb0000200000ffb29078190405e5
```

Dieser automatisch erstellte Bondport gehört bereits zum Netzwerk “172.16.64.0”. Wir können nun noch definieren, welche Rollen dieses Netzwerk haben soll

```
edit network name=147.50.0.0
roles='MANAGEMENT,LIVE_MIGRATE,CLUSTER_HEARTBEAT,VIRTUAL_MACHINE,STORAGE'
```

Über das ovmcli können wir natürlich auch NTP sowie ein YUM Repository definieren:

```
setNTP list=81.94.123.16
syncNtp
edit YumConfig
 baseUrl=http://public-yum.oracle.com/repo/OracleVM/OVM3/latest/x86_64/
 gpgKeyCheck=yes gpgKey=http://public-yum.oracle.com/RPM-GPG-KEY-oracle-el5
```

Erstellen Virtueller Maschinen

Eine Virtuelle Maschine (VM) besteht aus zwei Teilen: Einerseits aus physischem Storage (meist eine Disk-Datei auf dem Repository), andererseits aus der Definition der VM.

Wenn wir bereits eine bestehende Diskdatei haben, genügt es, diese auf das Repository-Filesystem zu kopieren (oder eine leere mit „dd“ erstellen), und diese dem Repository bekannt zu machen. (Man kann die Disk natürlich auch mit „create virtualDisk“ erzeugen lassen)

```
refresh repository name=meinRepo
```

Als nächstes definieren wir die VM und fügen Netzwerkkarten und Storage hinzu.

```
create vm name=vm01 repository=meinRepo domainType=XEN_HVM
 osType=SOLARIS_10 bootorder="DISK,CDROM" cpucount=2 highAvailability=yes
 memory=4096 on serverpool name=meinPool

create vnic name=00:21:f6:00:00:09 network=172.16.64.0

add vnic name=00:21:f6:00:00:09 to vm name=vm01

create VmDiskMapping slot=1 storageDevice=vm1_rootdisk.img
 name=vm1_rootdisk on vm name=vm1
```

Fazit

Mit wenigen Kommandos im ovmcli lässt sich der ganze Setup, von der Hardware bis zu den virtuellen Maschinen, erstellen. Sollten wir ein Testsystem aufbauen müssen, oder müssen wir unser Produktionssystem neu aufsetzen, können wir diese Kommandos erneut hervornehmen und leicht angepasst einfach nochmals laufen lassen.

Es hat zwar lange gedauert, aber nun hat uns Oracle eine gelungene, weitgehend vollständige Implementierung zur Administration von OVM über die Kommandozeile geliefert.

Kontaktadresse:
Martin Bracher
Trivadis AG
Europa-Strasse 5
CH-8152 Glattbrugg

Telefon:	+41 58 459 56 56
Fax:	+41 58 459 56 66
E-Mail	martin. bracher@trivadis.com
Internet:	www.trivadis.com