

Oracle VM 3

Creating and configuring Virtual Machines via command line

Martin Bracher

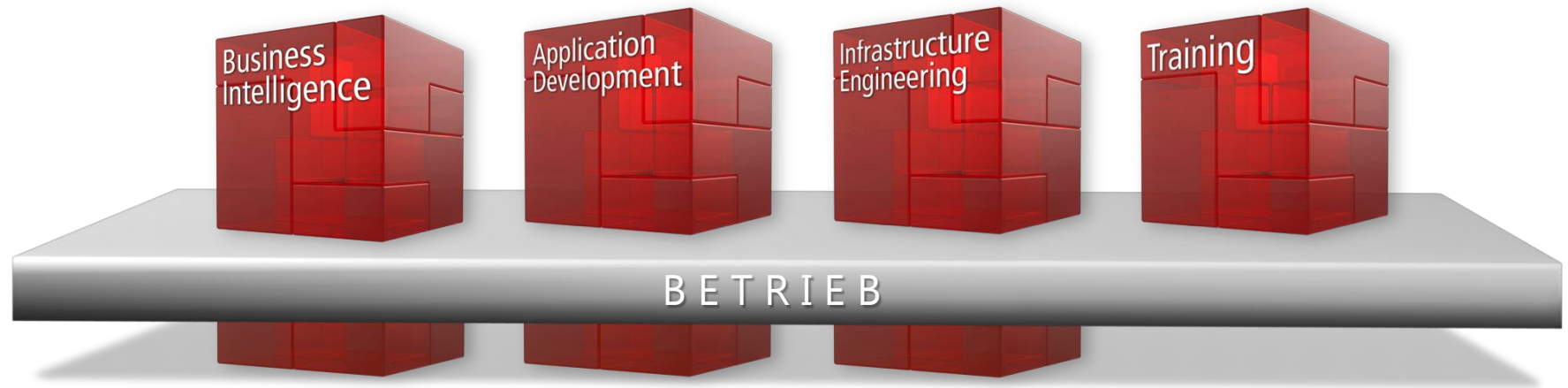


BASEL BERN BRUGG LAUSANNE ZÜRICH DÜSSELDORF FRANKFURT A.M. FREIBURG I.BR. HAMBURG MÜNCHEN STUTTGART WIEN

■ Unser Unternehmen

Trivadis ist **führend bei der IT-Beratung, der Systemintegration, dem Solution-Engineering** und der Erbringung von **IT-Services** mit Fokussierung auf **ORACLE®** und  **Microsoft** Technologien im D-A-CH-Raum.

Unsere Leistungen erbringen wir aus den strategischen Geschäftsfeldern:



Trivadis Services übernimmt den korrespondierenden Betrieb Ihrer IT Systeme.

■ Mit über 600 IT- und Fachexperten bei Ihnen vor Ort



12 Trivadis Niederlassungen mit über 600 Mitarbeitenden

200 Service Level Agreements

Mehr als 4'000 Trainingsteilnehmer

Forschungs- und Entwicklungsbudget: CHF 5.0 Mio. / EUR 4.0 Mio.

Finanziell unabhängig und nachhaltig profitabel

Erfahrung aus mehr als 1'900 Projekten pro Jahr bei über 800 Kunden

Stand 12/2013

■ AGENDA

1. What is ...?

- Virtual Machine
- Cloning
- Template

2. Creating a Master VM or template

3. Cloning

4. Example, demo

■ What is a Virtual Machine

- For the user / the application: it is **like a physical server**
- It is an **operating system** running on virtual "hardware"
- The virtual hardware is a mapping to physical hardware
 - Virtual disks/LUN's are mapped to block devices or files on the host
 - Part of host's Memory, Network, CPU is used by the VM

■ What is a Virtual Machine: XEN/OVM

■ Virtual hardware:

- A **configuration** file defining the "hardware" of the VM
 - CPU, memory, disks, console/graphics

```
bootloader = '/usr/bin/pygrub'  
memory = '1024'  
name = 'VM001'  
vcpus = 1  
on_crash = 'restart'  
on_reboot = 'restart'  
vfb = [type=vnc,vncunused=1,vnclisten=127.0.0.1,keymap=en-us]  
vif = [mac=00:21:f6:99:73:1d,bridge=0004fb00108e6b2]  
disk = ['file:/OVS/Repositories/004fb.../VirtualDisks/System.img,xvda,w',  
        'file:/OVS/Repositories/004fb.../VirtualDisks/data.img,xvdb,w']
```



- Virtual **disks** (on the host as file, disk(-partition), logical volume)

```
-rw-r--r-- 1 root root 6448619520 Aug 7 17:36 System.img  
-rw-r--r-- 1 root root 53694627840 Aug 7 17:36 data.img
```



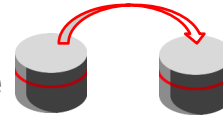
■ What is Cloning

- Copying an existing VM or template and using it as a new one

1. Define a new VM



2. Copy the disks of an existing VM/Template



3. Map the copied disks to the newly defined VM



- All 3 steps can be done automatically

```
clone vm name=master1 destType=Vm destName=vm001 serverPool=myPool
      targetRepository=myRepo
```

- Or you can do every step yourself

```
create vm name=vm001 domainType=XEN_PVM ...
create vnic name=vm001_eth0 network=vlanpublic on vm name=vm001
cloneVdToRepo VirtualDisk name=MyDisk target=MyRepo cloneType=THIN_CLONE
create VmDiskMapping slot=0 VirtualDisk=... name=xvda on vm name=vm001
```

■ What is Cloning

- Or combine both methods
 - Automatically clone a minimal VM
 - Add individual components
 - E.g. disks for ASM can be created. No need to clone it.

```
clone vm name=master_oel6 destType=Vm destName=vm201 serverPool=slot
      targetRepository=slotrepottcovms02sdb;

create VirtualDisk size=3.3 shareable=yes sparse=yes name=vm201vm202_xvdd
      on repository name=slotrepottcovms02sdb;

create VmDiskMapping slot=3 VirtualDisk=vm201vm202_xvdd name=vm201_grid1
      on vm name=vm201;
```


■ What is a Template?

- Problem after cloning:
 - Original and clone have the same configuration: hostname, IP, ...
- A **template** is a virtual machine that
 - Can not directly be booted in OVM
 - Can be used for cloning (as described before)
 - **The new cloned VM individually configures at first boot**
 - IP address
 - Hostname
 - New ssh host-keys
 - up2date UUID
 - ...

■ When to (re-)configure a VM at boot time

- How to detect if it is cloned and not yet configured?
 - Use the approach of Oracle:
 - Variable `RUN_TEMPLATE_CONF` (YES|NO) (old approach)
 - Variable `INITIAL_CONFIG` (YES|NO) (new approach)
 - In a template, it must be set to YES
 - How to change it in a template (if forgotten to do it)?

```
kpartx -av <uuid.img>  
mount /dev/mapper/loopXp2 /mnt; vi /mnt/...; umount /mnt  
kpartx -dv <uuid.img>
```
 - Based on MAC address:
 - In `/etc/sysconfig/network-scripts/ifcfg-eth0` the MAC address is stored
 - If the current MAC address of `eth0` is different, assume the machine was cloned and needs re-configuration
 - Caution: Newer RHEL/OEL versions will replace the configuration itself if the MAC address has changed (5.6 OK, 5.8 not OK)

■ AGENDA

1. What is ...?
- 2. Creating a Master VM or template**
3. Cloning
4. Example, demo

■ Creating VM's from templates: Overview

- **Initially** create a template or master VM
 - Install the OS and software you need
 - Can be done interactively or via commandline
 - or-
 - An existing VM or template can be used and adapted
 - Add the software to configure a cloned VM
-
- Create new VM's via cloning
 - Clone the template
 - Configure the new VM during the 1st boot

■ Templates from Oracle

Oracle offers templates for different needs

- technet.oracle.com → Downloads → Linux and VM
 - or <https://edelivery.oracle.com/linux>
- In E-Delivery, select "Oracle VM Templates" and the platform (32/64bit)
 - Some products are only available for 32bit others only for 64bit

Media Pack Search

Select the Product Pack and Platform and click "Go".

Select a Product Pack

Platform

Results

| Select | Description | Release | Part Number | Updated | # Parts / Size |
|----------------------------------|---|------------|-------------|-------------|----------------|
| <input checked="" type="radio"/> | Oracle VM Templates for Oracle E-Business Suite Release 12.1.3 Vision Media Pack for x86 (64 bit) | 12.1.3.0.0 | B63152-01 | APR-11-2011 | 11 / 38G |
| <input type="radio"/> | Oracle VM Templates for ID | 3.0.0.0.0 | R62414-02 | APR-06-2011 | 14 / |

■ Templates from Oracle

Some templates from Oracle are also supported for **production environments**, e.g.

- Template for RAC 11g Release 2 with special requirements
 - Each node must run on a different Oracle VM server
 - Shared disks must be configured as physical devices, not files
 - ' **phy** : /dev/sdg, xvdb, w! '
 - Shared disk must be on real shared storage between the OVM servers
- For non-production, the above restrictions do not apply

Initially define a new VM – via WEB



Create Virtual Machine

- Create Virtual Machine
- Set up Networks
- Arrange Disks
- Boot Options

* Server Pool: slot
Server: ttcovms01
* Repository: slotreposan01
* Name: ttcovmm01
 Enable High Availability
Description: OVM Manager

Operating System: Oracle Linux 5
Keymap: en-us (English, United State...
Domain Type: Xen PVM
Max. Memory (MB): 4096
Memory (MB): 4096
Max. Processors: 2
Processors: 2
Priority: 50
Processor Cap %: 100

Add a Virtual NIC to this Virtual Machine:

Dynamically Assign MAC
 Specify MAC Address:

Network: vlaninterconnect

Create Virtual Machine

- Create Virtual Machine
- Set up Networks
- Arrange Disks**
- Boot Options

Set the slot positions for your ISOs and disks:

| Slot | Disk Type | Contents | Actions |
|------|---------------|--------------|---------|
| 0 | Virtual Disk | Empty | |
| 1 | Physical Disk | Empty | |
| 2 | CD/DVD | EMPTY_CDRROM | |
| 3 | Empty | | |

■ Initially define a new VM – via ovmcli

- Login to the OVM Manager Commandline interface (ovmcli)

```
ssh -p 10000 admin@ovmmanager  
OVM>
```

- Define the virtual hardware

```
create Vm name=slot011 \  
repository=slotrepositan01 \  
domainType=XEN_PVM \  
osType=OL_6 \  
bootOrder=DISK \  
cpuCount=2 \  
memory=2048 \  
highAvailability=yes \  
on ServerPool name=slot
```

Hint: install it as XEN_HVM, then you can use CDROM as installation media.
Afterwards change it to XEN_PVM

■ Initially define a new VM – via ovmcli

■ Create and add virtual network cards

■ OVM ≥ 3.3

```
create vnic macAddress=00:21:f6:01:00:aa  
  name=00:21:f6:01:00:aa network=vlanpublic  
  on vm name=vm001
```

■ OVM < 3.3

```
create Vnic name=00:21:f6:01:00:aa network=vlanpublic  
add Vnic name=00:21:f6:01:00:aa to Vm name=vm001
```

■ Add virtual disks

```
create VirtualDisk size=3.3 shareable=yes sparse=yes  
  name=vm001_xvdd on repository name=myRepo
```

```
create VmDiskMapping  
  slot=0 storageDevice=vm001_xvdd name=xvdd on Vm name=vm001
```

■ Initially install and prepare the new master VM

- Start the new VM
- Install the OS
 - Paravirtualized: network installation
 - Hardware-virtualized: from DVD-Image (or PXE boot)
- Add required software (e.g. Oracle home)
- Prepare it as a template
 - Add a mechanism to configure at first boot (see next slides)

■ AGENDA

1. What is ...?
2. Creating a Master VM or template
- 3. Cloning**
4. Example, demo

■ Workflow of cloning an Oracle template

- OVM Manager or ovmcli clone-procedure creates the virtual server
 - Unique configuration file vm.cfg with the virtual hardware definition
 - Copy of the disk files

```
clone vm name=myTemplate destType=Vm destName=myVM \  
serverPool=myPool targetRepository=myRepo
```

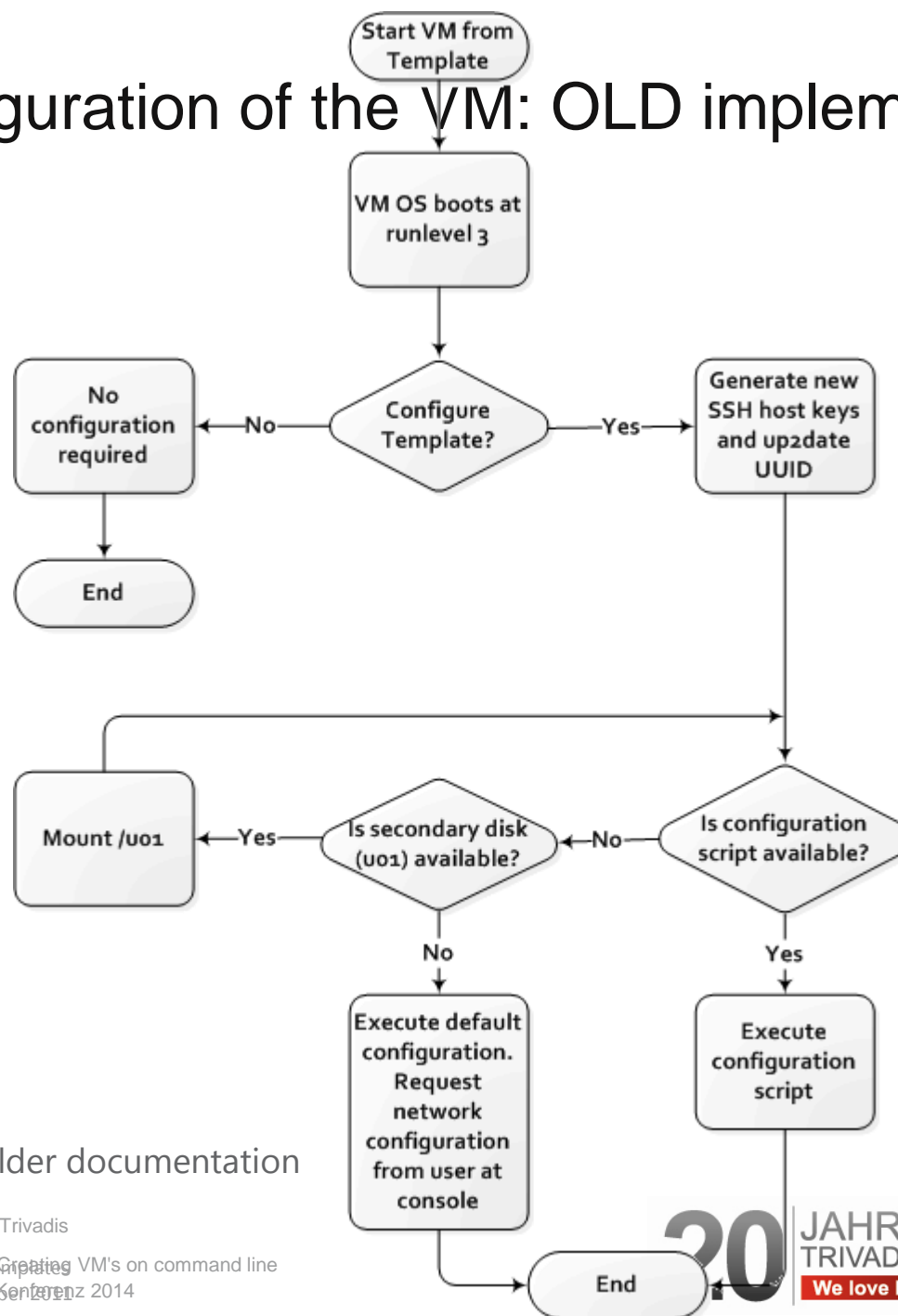
- The first boot of the new VM configures the cloned server
 - A configuration script is started in **runlevel 3**
 - Checks if it has to re-configure the VM as a new VM
 - **Old implementation:**
 - > /etc/init.d/oraclevm-template
 - > /etc/sysconfig/oraclevm-template (RUN_TEMPLATE_CONF=yes)
 - **New implementation:**
 - > /etc/init.d/ovmd
 - > /etc/sysconfig/ovmd (INITIAL_CONFIG=yes)

■ Initial configuration of the VM: OLD implementation

OLD implementation: ovm-template-config

- Provided as RPM package `ovm-template-config.rpm`
- A configuration script is started in runlevel 3
 - `/etc/init.d/oraclevm-template`
- This script evaluates the file `/etc/sysconfig/oraclevm-template`
 - `RUN_TEMPLATE_CONF` (YES|NO)
 - YES: run the configuration: `/usr/sbin/oraclevm-template --config`
 - `TEMPLATE_CONFIG_SCRIPT`
 - Individual script for configuration. If not specified:
 - `oraclevm-template.sh` on the 1st partition of the 2nd disk will be run (mounted as `/u01`)
 - If also missing: default network configuration script will be run

Initial configuration of the VM: OLD implementation



Source: Oracle OVM builder documentation

■ Initial configuration of the VM: NEW implementation

NEW implementation: Oracle VM Guest Additions

- Initial configuration is based on the new Oracle VM Guest Additions
 - Unlike "Guest Additions" of other virtualization solutions:
It does NOT include additional/enhanced hardware drivers like in other virtualization solution (these drivers are already included in Linux kernel)
 - It provides information to the OVM Manager, like current IP-Address



The screenshot shows the Oracle VM Manager interface for VM 'vm201'. The 'Networks' tab is selected, displaying a table of network interfaces. The table has three columns: 'VNIC', 'Ethernet Network', and 'IP Addresses'. There are two rows of data.

| VNIC | Ethernet Network | IP Addresses |
|-------------------|------------------|---|
| 00:21:f6:99:48:f7 | vlanpublic | 192.168.97.201,fe80::221:f6ff:fe99:48f7 |
| 00:21:f6:99:30:5f | vlaninterconnect | 10.0.0.201,fe80::221:f6ff:fe99:305f |

- A daemon "ovmd" is running inside the VM
 - Messages can be sent from outside (Manager, ovmcli) to ovmd
 - Directly to the VM, no network configuration required
 - ovmd can start actions like (re-)configuration with these messages

■ Initial configuration of the VM: NEW implementation

■ Oracle VM Guest Additions **installation**

- Provided as RPM packages: `ovmd ovm-template-config*.rpm`
 - Not included on the installation ISO
 - Download the latest repository definition to `/etc/yum.repos.d/`
 - `wget http://public-yum.oracle.com/public-yum-ol6.repo`
 - Enable the `[ol6_addons]` section (`enabled=1`)
 - Install the software and activate `ovmd`

```
yum install libovmapi xenstoreprovider python-simplejson \  
            ovmd ovm-template-config*  
chkconfig ovmd on  
/etc/init.d/ovmd start
```

- In current templates from Oracle it is already installed

■ Initial configuration of the VM: NEW implementation

■ Oracle VM Guest Additions **configuration**

- A master configuration script **ovm-template-config** is used
- Scripts stored in /etc/template.d/scripts
 - authentication datetime firewall network selinux ssh system user
- Works like System V init.d runlevel scripts (**ovm-chkconfig**)

```
vm001:~ # ovm-chkconfig --list
```

| name | configure | unconfigure | reconfigure | cleanup | suspend | resume | migrate | shutdown |
|----------------|-----------|-------------|-------------|---------|---------|--------|---------|----------|
| authentication | on:90 | off | off | on:10 | off | off | off | off |
| datetime | on:50 | off | off | on:50 | off | off | off | off |
| firewall | on:41 | off | off | off | off | off | off | off |
| network | on:50 | off | off | on:50 | off | off | off | off |
| selinux | on:30 | off | off | off | off | off | off | off |
| ssh | on:70 | off | off | on:30 | off | off | off | off |
| system | on:60 | off | off | on:60 | off | off | off | off |
| user | on:60 | off | off | on:40 | off | off | off | off |

■ Initial configuration of the VM: NEW implementation

■ Running the scripts

- Which parameters are expected?

```
ovm-template-config --human-readable --enumerate configure --script selinux  
[('30',  
 'selinux',  
 [{u'description': u'SELinux mode: enforcing, permissive or disabled.',  
   u'hidden': True,  
   u'key': u'com.oracle.linux.selinux.mode'}]])]
```

- Running manually

```
ovm-template-config --stdin configure <<EOD  
{"com.oracle.linux.selinux.mode": "disabled"}  
{"com.oracle.linux.root-password": "ovsroot"} ← must be the last line!  
EOD
```

- Running manually and interactive

```
ovm-template-config --console-input configure
```

■ Initial configuration of the VM: NEW implementation

■ Running the scripts (cont.)

- At next reboot (or after cloning), to re-configure the VM

```
ovmd -s cleanup  
service ovmd enable-initial-config
```

- (sets INITIAL_CONFIG=yes in /etc/sysconfig/ovmd)

■ Running a single script/module

```
ovm-template-config --script selinux --console-input configure
```

■ Initial configuration of the VM: NEW implementation

■ Adapt the VM for your needs

- Examples: Install and setup Oracle Clusterware, create an initial database
- Create/Add your **own configuration scripts** to /etc/template.d/scripts
- A Python script is required

- Module script header

```
#!/usr/bin/env python
### BEGIN PLUGIN INFO
# name: myscript
# configure: 90
# description: Python wrapper Script to call myscript.ksh.
### END PLUGIN INFO
```

- configure: the execution order relative to the other scripts (00-99)

- Module script body

- Must support at least 1 target

> **configure**, unconfigure, reconfigure, cleanup, resume, migrate, shutdown

■ Initial configuration of the VM: NEW implementation

- If you do not like Python: create a wrapper script to call a shell-script

```
#!/usr/bin/env python
### BEGIN PLUGIN INFO
# name: myscript
# configure: 90
# description: Python wrapper Script to call myscript.ksh.
### END PLUGIN INFO
try:
    import json
except ImportError:
    import simplejson as json
from templateconfig.cli import main
from templateconfig.common import run_cmd

def do_enumerate(target):
    param = []
    if target == 'configure':
        param += []
    return json.dumps(param)

def do_configure(param):
    param = json.loads(param)
    run_cmd('/path/myscript.ksh')
    return json.dumps(param)

if __name__ == '__main__':
    main(do_enumerate, {'configure': do_configure})
```

■ Initial configuration of the VM: NEW implementation

- Create a shell script to do your configuration steps

```
#!/bin/bash
# myscript.ksh
exec 1>/tmp/myscript.$$log
exec 2>&1
VAR1=`ovmd -g key1` #read variable from ovmd

# here comes your code...
/u00/scripts/install_oracle.sh $VAR1
```

- Activate it for execution:
 - **ovm-chkconfig --add myscript** #python-wrapper from slide before

```
vm001:~ # ovm-chkconfig --list
```

| name | configure | unconfigure | reconfigure | cleanup | suspend | resume | migrate | shutdown |
|-----------------|--------------|-------------|-------------|------------|------------|------------|------------|------------|
| authentication | on:90 | off | off | on:10 | off | off | off | off |
| datetime | on:50 | off | off | on:50 | off | off | off | off |
| firewall | on:41 | off | off | off | off | off | off | off |
| myscript | on:90 | off | off | off | off | off | off | off |
| network | on:50 | off | off | on:50 | off | off | off | off |
| ... | | | | | | | | |

■ Initial configuration of the VM: NEW implementation

■ Test your script

```
ovmd -p key=value  
ovm-template-config -s myscript --console-input configure
```

■ Interactive configuration at startup

- Required values can be entered in the console

```
Enabling local filesystem quotas: [ OK ]  
Enabling /etc/fstab swaps: [ OK ]  
Entering non-interactive startup  
Starting OVM template configure: network: System host name, e.g., "localhost.localdomain": _
```

■ Automatic configuration at startup

- Variable/Value pairs can be sent via messaging channel from outside to VM

```
OVMS> sendVmMessage Vm name=slot001 key=com.oracle.linux.network.device.0  
message=eth0 log=no
```

```
OVMS> sendVmMessage vm name=slot001 key=com.oracle.linux.root-password  
message=manager log=no
```

- ovmd can read the values and pass it to the ovm-template-config script

```
ovmd -s configure
```

- (will be done automatically if configured for initial boot)

■ AGENDA

1. What is ...?
2. Creating a Master VM or template
3. Cloning
4. **Example, demo**

■ Creating and configuring a cloned VM via commandline

- Create a VM with Oracle Grid Infrastructure
 - The software is already installed (software-only installation in the template)
 - Using the ovmcli interface (ssh)
 - Login and copy-paste
 - run it as a script

```
ssh -p 10000 admin@ttcovmm01 "  
  <ovm-command1> ; ← semicolon is required! (it is not allowed for interactive copy-paste)  
  <ovm-command2> ;  
"
```

- You can use shell-variables in script mode

```
edit vm name=${VMNAME} memory=${MEM} ;
```

■ Creating and configuring a VM via commandline

■ Clone the VM

```
clone vm name=master_sw_only destType=Vm destName=vm201
      serverPool=slot targetRepository=slotrepottcovms02sdb;

edit vm name=vm201 memory=4300;
```

- It creates the VM and clones the root disk
- Afterwards, create and add empty disks for ASM
 - Hint: create a naming concept to identify the disks and mappings via name
 - vm201vm202_xvdd: disk xvdd shared between vm201 and vm202
 - vm201_xvdd: mapping name of disk xvdd in vm201
 - vm202_xvdd: mapping name of disk xvdd in vm202

■ Creating and configuring a VM via commandline

- Create and add new shared storage for ASM

```
create VirtualDisk size=3.3 shareable=yes sparse=yes name=vm201vm202_xvdd
    on repository name=slotrepottcovms02sdb;
create VirtualDisk size=3.3 shareable=yes sparse=yes name=vm201vm202_xvde
    on repository name=slotrepottcovms02sdb;
create VirtualDisk size=3.3 shareable=yes sparse=yes name=vm201vm202_xvdf
    on repository name=slotrepottcovms02sdb;

create VmDiskMapping slot=3 VirtualDisk=vm201vm202_xvdd name=vm201_xvdd
    on vm name=vm201;
create VmDiskMapping slot=4 VirtualDisk=vm201vm202_xvde name=vm201_xvde
    on vm name=vm201;
create VmDiskMapping slot=5 VirtualDisk=vm201vm202_xvdf name=vm201_xvdf
    on vm name=vm201;
create VmDiskMapping slot=3 VirtualDisk=vm201vm202_xvdd name=vm202_xvdd
    on vm name=vm202;
create VmDiskMapping slot=4 VirtualDisk=vm201vm202_xvde name=vm202_xvde
    on vm name=vm202;
create VmDiskMapping slot=5 VirtualDisk=vm201vm202_xvdf name=vm202_xvdf
    on vm name=vm202;
```

■ Creating and configuring a VM via commandline

■ Initial boot and send configuration values

```
start vm name=vm201;
sendvmmessage vm name=vm201 key=com.oracle.linux.network.hostname
                                message=vm201.ovm.trivadis.com log=no;
sendvmmessage vm name=vm201 key=com.oracle.linux.network.device.0      message=eth0 log=no;
sendvmmessage vm name=vm201 key=com.oracle.linux.network.onboot.0      message=yes log=no;
sendvmmessage vm name=vm201 key=com.oracle.linux.network.bootproto.0   message=static log=no;
sendvmmessage vm name=vm201 key=com.oracle.linux.network.ipaddr.0      message=192.168.97.201 log=no;
sendvmmessage vm name=vm201 key=com.oracle.linux.network.netmask.0     message=255.255.255.0 log=no;
sendvmmessage vm name=vm201 key=com.oracle.linux.network.gateway.0     message=192.168.97.250 log=no;
sendvmmessage vm name=vm201 key=com.oracle.linux.network.dns-servers.0  message=192.168.97.250 log=no;
sendvmmessage vm name=vm201 key=com.oracle.linux.network.dns-search-domains.0 message=ovm.trivadis.com log=no;
sendvmmessage vm name=vm201 key=com.oracle.linux.selinux.mode          message=disabled log=no;
sendvmmessage vm name=vm201 key=trivadis                               message=standalone log=no;
sendvmmessage vm name=vm201 key=com.oracle.linux.root-password         message=manager log=no;
```

- `com.oracle.linux.root-password` **must be the LAST key!**
- `com.oracle.linux.network.*.1` is the 2nd network interface
- IP/hostname in `/etc/hosts` set to the last interface if not specified `host=...`
- That's it. Now you have to wait until configuration is completed...

■ Creating and configuring a VM via commandline

- Configuring Grid Infrastructure for a **standalone server** ("Oracle Restart")
 - Part of the script called by the Python wrapper script

```
trivadis=`ovmd -g trivadis` #parameters separated by ';'
config_as=${trivadis%%;*}

case $config_as in
standalone)
    /u00/app/grid/product/12.1.0.1/perl/bin/perl \
    -I/u00/app/grid/product/12.1.0.1/perl/lib \
    -I/u00/app/grid/product/12.1.0.1/crs/install \
    /u00/app/grid/product/12.1.0.1/crs/install/roothas.pl

    su - grid -c "
        PATH=/u00/app/grid/product/12.1.0.1/bin:\$PATH
        srvctl add listener
        srvctl start listener
        asmca -silent -configureASM -diskstring '/dev/asm/*' -diskGroupName DATA \
        -diskList '/dev/asm/disk1' -redundancy EXTERNAL \
        -sysAsmPassword manager -asmsnmpPassword manager
    "
    ;;
esac
```

■ Creating and configuring a VM via commandline

- Configuring Grid Infrastructure for a **cluster**
 - More complex, needs a fully started server
 - ovmf is not the last runlevel script, so many services are not yet started
 - > e.g. ntpd, nfs filesystems, atd
 - Run the script asynchronously via atd

```
service atd start
  at now + 1 minute <<EOD
    /etc/tvd/cluster.sh
EOD
```

```
su - grid -c "/u00/app/grid/product/12.1.0.1/crs/config/config.sh -silent \  
-responseFile /etc/tvd/grid_config_sw_only.rsp"  
  
/u00/app/grid/product/12.1.0.1/root.sh  
  
ssh vm202 /u00/app/grid/product/12.1.0.1/root.sh  
  
su - grid -c "/u00/app/grid/product/12.1.0.1/cfgtoollogs/configToolAllCommands \  
RESPONSE_FILE=/etc/tvd/configToolAllCommands.rsp"
```

Fragen und Antworten...

Martin Bracher
Senior Consultant

[martin.bracher @ trivadis.com](mailto:martin.bracher@trivadis.com)



BASEL BERN BRUGG LAUSANNE ZÜRICH DÜSSELDORF FRANKFURT A.M. FREIBURG I.BR. HAMBURG MÜNCHEN STUTTART WIEN

Trivadis an der DOAG

Ebene 3 - gleich neben der Rolltreppe

Wir freuen uns auf Ihren Besuch.

Denn mit Trivadis gewinnen Sie immer.