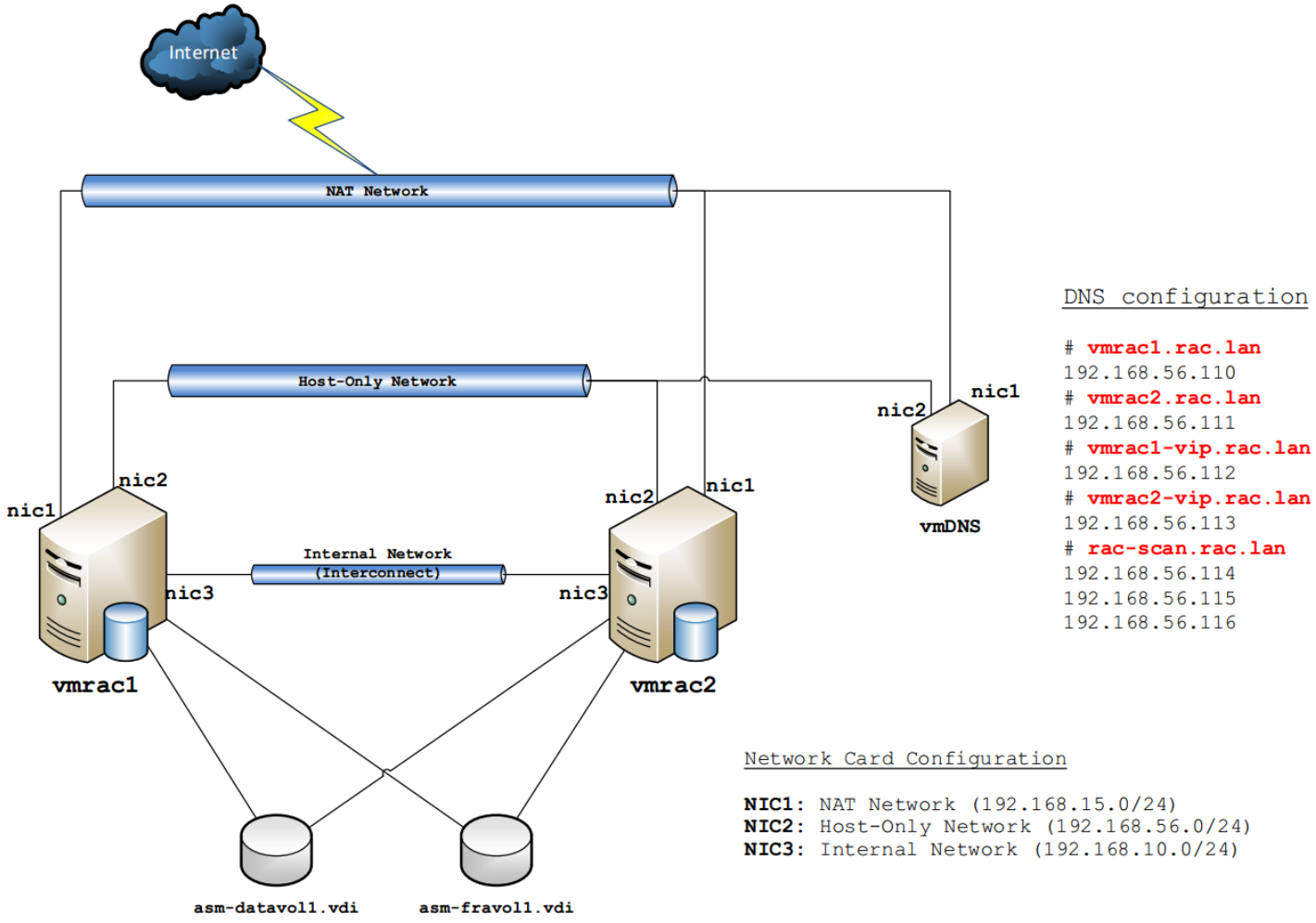
Oracle Real Application Cluster 12c on Oracle Linux 7.2 using Oracle VirualBox

Installation of two-node Oracle Real Application Cluster 12c (12.1.0.2.0) will be done in silent mode using Oracle VirtualBox on Windows 8.1 PC with 16Gb of RAM and about 80Gb of free space.

Virtual environment diagram:



The virtual environment consists of two virtual machines for Oracle RAC nodes, two VirtualBox shared disks for Oracle ASM and one virtual machine for DNS and NTP Server. VirtualBox does not allow a host to be connected directly to a NAT network; therefore, tree networks are required:

1. **NAT Network** - to provide internet connection
2. **Host-Only Network** - to provide connectivity between host and virtual machines. This network will be used as Oracle Public Network
3. **Internal Network** - which will be used as an Oracle Private Network (Interconnect)

The DNS Server will resolve host names, VIPs and three IP addresses for SCAN (Single Client Access Name) in a round-robin fashion.

Installation of Oracle Grid Infrastructure, Database Software and creation of Container Databases will be performed without graphical interface (in silent mode).

The operating system user **oracle** will be the owner of both Oracle Grid Infrastructure and Database Software. Separation of common administrative tasks will be implemented using operating system groups. You will not have to switch between environments manually by exporting **ORACLE_HOME** and **ORACLE_SID** variables, as **.bash profile** will contain environment switches. There also will be some short aliases for rapid access to the CRS, ASM and Database alert logs. Most part of Oracle command line tools such as **sqlplus** and **rman** will be called using the **rlwrap** utility, which allows readline's line editing and persistent history.

Virtual machine specifications are the following:

Oracle RAC Nodes specs:
vCPU: 2
RAM: 4 GB
Local Storage: 30 GB (Dynamically allocated storage)
NIC1: NAT Network
NIC2: Host-Only Network
NIC3: Internal Network
OS: Oracle Linux 7.2

Shared disks: 2 x 10 GB Disks (Fixed size storage)

DNS/NTP Server specs:
vCPU: 1
RAM: 1 GB
Local Storage: 10 GB (Dynamically allocated storage)
NIC1: NAT Network
NIC2: Host-Only Adapter
OS: CentOS 7

You can download Oracle software distribution media from download.oracle.com.
VirtualBox software can be found here.
CentOS7 download page.

Preparation of virtual environment

Oracle VirtualBox provides comprehensive command line interface. To prepare a virtual environment for Oracle Real Application Cluster do the following:

Run **CMD** on the host computer and execute the following commands (replace **<PATH_TO_ISO>** with the path to the operating system ISO image):

Create a virtual NAT network:

```
"%programfiles%\Oracle\VirtualBox\VBoxManage.exe" natnetwork add --netname "vmracnat" --network "192.168.15.0/24" --enable
```

Create a virtual machine for DNS/NTP server:

```
"%programfiles%\Oracle\VirtualBox\VBoxManage.exe" createvm --register --name "vmDNS" --ostype RedHat_64
"%programfiles%\Oracle\VirtualBox\VBoxManage.exe" storagectl "vmDNS" --name "IDE" --add ide --bootable on
"%programfiles%\Oracle\VirtualBox\VBoxManage.exe" storagectl "vmDNS" --name "SATA" --add sata --controller IntelAhci --bootable on
```

```
"%programfiles%\Oracle\VirtualBox\VBoxManage.exe" createhd --filename
"%userprofile%\VirtualBox VMs\vmDNS\vmDNS-localdisk.vdi" --size 20480 --format VDI
"%programfiles%\Oracle\VirtualBox\VBoxManage.exe" storageattach "vmDNS" --storagectl
"IDE" --port 0 --device 0 --type dvddrive --medium "<PATH_TO_ISO>"
"%programfiles%\Oracle\VirtualBox\VBoxManage.exe" storageattach "vmDNS" --storagectl
"SATA" --port 0 --device 0 --type hdd --medium "%userprofile%\VirtualBox VMs\vmDNS\vmDNS-
localdisk.vdi"
"%programfiles%\Oracle\VirtualBox\VBoxManage.exe" modifyvm "vmDNS" --boot1 disk --boot2
dvd --boot3 none --memory 1024 --cpus 1 --cpuexecutioncap 80 --vram 12 --nic1 natnetwork
--nat-network1 "vmracnat" --nic2 hostonly --hostonlyadapter2 "VirtualBox Host-Only
Ethernet Adapter" --audio none
```

Create virtual machines for Oracle RAC nodes.

```
"%programfiles%\Oracle\VirtualBox\VBoxManage.exe" createvm --register --name "vmrac1" --
ostype Oracle_64
"%programfiles%\Oracle\VirtualBox\VBoxManage.exe" storagectl "vmrac1" --name "IDE" --add
ide --bootable on
"%programfiles%\Oracle\VirtualBox\VBoxManage.exe" storagectl "vmrac1" --name "SATA" --add
sata --controller IntelAhci --bootable on
"%programfiles%\Oracle\VirtualBox\VBoxManage.exe" createhd --filename
"%userprofile%\VirtualBox VMs\vmrac1\vmrac1-localdisk.vdi" --size 30720 --format VDI
"%programfiles%\Oracle\VirtualBox\VBoxManage.exe" storageattach "vmrac1" --storagectl
"IDE" --port 0 --device 0 --type dvddrive --medium "<PATH_TO_ISO>"
"%programfiles%\Oracle\VirtualBox\VBoxManage.exe" storageattach "vmrac1" --storagectl
"SATA" --port 0 --device 0 --type hdd --medium "%userprofile%\VirtualBox
VMs\vmrac1\vmrac1-localdisk.vdi"
"%programfiles%\Oracle\VirtualBox\VBoxManage.exe" modifyvm "vmrac1" --boot1 disk --boot2
dvd --boot3 none --memory 4096 --cpus 2 --cpuexecutioncap 80 --vram 12 --nic1 natnetwork
--nat-network1 "vmracnat" --nic2 hostonly --hostonlyadapter2 "VirtualBox Host-Only
Ethernet Adapter" --nic3 intnet --audio none
"%programfiles%\Oracle\VirtualBox\VBoxManage.exe" createvm --register --name "vmrac2" --
ostype Oracle_64
"%programfiles%\Oracle\VirtualBox\VBoxManage.exe" storagectl "vmrac2" --name "IDE" --add
ide --bootable on
"%programfiles%\Oracle\VirtualBox\VBoxManage.exe" storagectl "vmrac2" --name "SATA" --add
sata --controller IntelAhci --bootable on
"%programfiles%\Oracle\VirtualBox\VBoxManage.exe" createhd --filename
"%userprofile%\VirtualBox VMs\vmrac2\vmrac2-localdisk.vdi" --size 30720 --format VDI
"%programfiles%\Oracle\VirtualBox\VBoxManage.exe" storageattach "vmrac2" --storagectl
"IDE" --port 0 --device 0 --type dvddrive --medium "<PATH_TO_ISO>"
"%programfiles%\Oracle\VirtualBox\VBoxManage.exe" storageattach "vmrac2" --storagectl
"SATA" --port 0 --device 0 --type hdd --medium "%userprofile%\VirtualBox
VMs\vmrac2\vmrac2-localdisk.vdi"
"%programfiles%\Oracle\VirtualBox\VBoxManage.exe" modifyvm "vmrac2" --boot1 disk --boot2
dvd --boot3 none --memory 4096 --cpus 2 --cpuexecutioncap 80 --vram 12 --nic1 natnetwork
--nat-network1 "vmracnat" --nic2 hostonly --hostonlyadapter2 "VirtualBox Host-Only
Ethernet Adapter" --nic3 intnet --audio none
```

Installation of operating systems

You can install operating systems manually or using Linux Kickstart for automated installation with preconfigured files - anaconda-ks.vmdns.cfg, anaconda-ks.vmrac1.cfg and anaconda-ks.vmrac2.cfg for **vmDNS, vmrac1** and **vmrac2**. Files can be downloaded from the https://github.com/georgiy-shubin/kickstart-files repository.
The easiest way to present kickstart files to virtual machines is to use the small Http File Server (**HFS**). Lunch **HFS** and select the IP address to listen on - **Menu -> IP address -> 192.168.56.1**. Drag and drop kickstart files to **Vurtual File System** window. While selecting each uploaded files you can see their **URL**s in the **Open in browser** field.

Note: Password for a root user in all Kickstart files is **oracle**.

Start the virtual machine by executing the **"VBoxManage.exe startvm"** on your host. After VM boots from Linux installation media, press **ESC** inside the VM console and the **"boot:"** field will appear. Type the kickstart command to start the automated installation.

vmDNS

Start the VM on the host:
"%programfiles%\Oracle\VirtualBox\VBoxManage.exe" startvm "vmDNS"
Kickstart command:
linux inst.ks=http://192.168.56.1/anaconda-ks.vmdns.cfg

vmrac1

Start the VM on the host:
"%programfiles%\Oracle\VirtualBox\VBoxManage.exe" startvm "vmrac1"
Kickstart command:
linux inst.ks=http://192.168.56.1/anaconda-ks.vmrac1.cfg

vmrac2

Start the VM on the host:
"%programfiles%\Oracle\VirtualBox\VBoxManage.exe" startvm "vmrac2"
Kickstart command:
linux inst.ks=http://192.168.56.1/anaconda-ks.vmrac2.cfg

Note: Name resolution will not work on Oracle RAC nodes until you configure the DNS Server.

OS configuration for DNS/NTP server:

Date & Time: Asia/Baku Time zone
Software Selection: Default
Installation Source: Local media
Installation Source: Auto
Network Config: NIC1 (**onboot=yes**) - **192.168.15.0/24** (gw - **192.168.15.1**, nameserver - **8.8.4.4, 8.8.8.8**, domain - **rac.lan**)
NIC2 (**onboot=yes**) - **192.168.56.0/24** (No gateway and DNS)
Hostname: **vmdns**
IP: nic1 - **192.168.15.101**, nic2 - **192.168.56.101**

OS configuration for Oracle RAC Nodes:

Date & Time: Asia/Baku Time zone
Software Selection: Minimal install
Installation Source: Local media
KDUMP: Disabled
Partitioning: /boot – 500 MB
swap – 4 GB
/ - remaining size
Network Config: NIC1 (**onboot=yes**) – **192.168.15.0/24** (gw - **192.168.15.1**, nameserver - **192.168.15.101**, domain - **rac.lan**)
NIC2 (**onboot=yes**) – **192.168.56.0/24** (No gateway and DNS)
NIC3 (**onboot=yes**) – **192.168.10.0/24** (No gateway and DNS)
Hostname And IP: **vmrac1** (nic - **192.168.15.110**, nic2 – **192.168.56.110**, nic3 – **192.168.10.110**)
vmrac2 (nic - **192.168.15.111**, nic2 – **192.168.56.111**, nic3 – **192.168.10.111**)

Installation and configuration of DNS and NTP server

You can use the python scripts to install and configure **DNS** and **NTP** server automatically. First, prepare you system.

As it is a test deployment, you can turn off the firewall…

```
[root@vmdns ~]# systemctl stop firewalld
[root@vmdns ~]# systemctl disable firewalld
```

and SELinux (requires a system reboot).

```
[root@vmdns ~]# cp /etc/selinux/config /etc/selinux/config.bkp.`date +%Y%m%d%H%M%S`
[root@vmdns ~]# sed -i "s/SELINUX=enforcing/SELINUX=disabled/g" /etc/selinux/config
```

Run the system update.

```
[root@vmdns ~]# yum -y update
```

Install **net-tool** and **telnet** for easier troubleshooting.

```
[root@vmdns ~]# yum install -y net-tools telnet
```

Install **git.**

```
[root@vmdns ~]# yum -y install git
```

Reboot the system

```
[root@vmdns ~]# reboot
```

Download the automated python scripts from the **git** repository.

```
[root@vmdns ~]# git clone https://github.com/jamalshahverdiev/ntp-and-dns-for-oracle
```

Run the **python-installer.sh** script to install the python environment.

```
[root@vmdns ~]# cd ~/ntp-and-dns-for-oracle/
[root@vmdns ntp-and-dns-for-oracle]# ./python-installer.sh
```

Just as the python environment is installed you can run the **ntp-dns-oracle.py** script which will ask for several details – IP address of the DNS server, a privileged user name and password, and a domain name that will be used in the Oracle Public Network.

Note: The cloned **git** repository will contain the **jinja2temps/iplist** file with the IP configuration for **DNS** and **NTP** server according to this manual. If you want to use a different IP configuration, please edit this file.

```
      [root@vmdns ntp-and-dns-for-oracle]# ./ntp-dns-oracle.py
This script downloads and installs NTP and DNS servers automatically.
Edit the ./jinja2temps/iplist file to provide an IP configuration.

Please input the following details:
IP address of DNS server (local or remote): 192.168.15.101
User name: root
Password: <type oracle here>
Domain name for RAC public network: rac.lan
 Installation and configuration of DNS server is in progress ...
DNS Server installed and configured successfully.
 Installation and configuration of NTP server is in progress ...
NTP Server installed and configured successfully.
```

To configure name resolution on your host, run **CMD** as administrator. Identify your primary adapter name and the current DNS Server IP by executing the **ipconfig** command, and execute the following:

```
C:\Windows\system32>netsh interface ipv4 add dns "VirtualBox Host-Only Network"
192.168.56.101
C:\Windows\system32>netsh interface ipv4 add dns "<adapter_name>" <primary_dns_ip>
C:\Windows\system32>netsh interface ipv4 add dns "<adapter_name>" 192.168.56.101 index=2
C:\Windows\system32>ipconfig /registerdns
C:\Windows\system32>ipconfig /flushdns
C:\Windows\system32>ping vmrac1.rac.lan

Pinging vmrac1.rac.lan [192.168.56.110] with 32 bytes of data:
Reply from 192.168.56.110: bytes=32 time<1ms TTL=64
Reply from 192.168.56.110: bytes=32 time<1ms TTL=64
Reply from 192.168.56.110: bytes=32 time<1ms TTL=64
Reply from 192.168.56.110: bytes=32 time<1ms TTL=64
```

To reset the DNS configuration:

```
C:\Windows\system32>netsh interface ipv4 set dnsservers "<adapter_name>" source=dhcp
C:\Windows\system32>netsh interface ipv4 set dnsservers "VirtualBox Host-Only Network"
source=dhcp
C:\Windows\system32>ipconfig /registerdns
C:\Windows\system32>ipconfig /flushdns
```

Preparation of Oracle Linux for Oracle Software

Operating system parameters

Perform the following tasks as a **root** user on the each cluster node.

First, install the EPEL Repository in to get the **rlwrap** utility:

```
      # rpm -ivh https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

Run the system update to download latest version of the packages. The most important are latest Unbreakable Enterprise Kernel (UEK) and TZDATA, which updates local DST rules.

```
# yum -y update
```

Install packages required by Oracle Software

```
# yum -y install oracle-rdbms-server-12cR1-preinstall oracleasm-support ntp net-tools rlwrap unzip git wget
```

oracle-rdbms-server-12cR1-preinstall – Performs configuration of operating system kernel parameter, creates oracle user and oinstall group, sets user limits, installs required packages to run oracle binaries and configures GRUB2 boot loader.

Note: On Oracle Linux 7.2, this package configures the boot loader thus OS loads standard Linux kernel. It is necessary to reconfigure it to load the UEK.

oracleasm-support – The OS may change the name of a block device across restarts. ASMLib driver labels shared disks, thus Oracle ASM can identify them. This driver is used instead of **UDEV** rules.

ntp – Network Time Protocol.

net-tools – OEL/RHEL/Centos 7 uses a new mechanism and commands to perform network related tasks. The **net-tools** package provides legacy commands (ifconfig, netstat, etc.) which are required for Cluster Verification Utility (CVU) to verify network compatibility between cluster nodes.

rlwrap – Provides readline's line editing and persistent history for Oracle command line tools such as **sqlplus** and **rman**.

unzip – Install unzip if the operating system installed as "minimal installation". In addition, you can install some useful utilities for troubleshooting (telnet, tcpdump, lsof, strace).

git – Utility that downloads stuff from the git repositories.

wget – Downloader with CLI

Create an additional operating system group for Oracle ASM administrative tasks and add the **oracle** user to it.

```
# groupadd -g 54328 asmadmin
# usermod -g oinstall -G dba,asmadmin oracle
```

Update boot order to make the latest UEK default. First, get the ordinal number of the latest kernel among available ones, set it to default and configure GRUB2. You can get a list of available kernels by executing the following command:

```
# grep "^menuentry" /boot/grub2/grub.cfg | cut -d "'" -f2  | awk '{print NR-1 " - " $0}'
```

Configure GRUB2

```
# grub2-set-default <UEK-ordinal-number>
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

Add hosts file records. Oracle RAC requires VIPs to be resolved by DNS and the hosts file. PRIVs are not required to be resolved; however, it is allowed to put them into a DNS server or the hosts file. Database requires hostname record of the local node to be present in the hosts file.

```
# cp /etc/hosts /etc/hosts.bkp.`date +%Y%m%d%H%M%S`
# cat <<EOF >> /etc/hosts
```

```
192.168.56.10 vmrac1 vmrac1.rac.lan
192.168.56.11 vmrac2 vmrac2.rac.lan
192.168.56.12 vmrac1-vip vmrac1-vip.rac.lan
192.168.56.13 vmrac2-vip vmrac2-vip.rac.lan
EOF
```

Stop and disable firewall.

```
# systemctl stop firewalld
# systemctl disable firewalld
```

Disable SELinux.

```
# cp /etc/selinux/config /etc/selinux/config.bkp.`date +%Y%m%d%H%M%S`
# sed -i "s/SELINUX=enforcing/SELINUX=disabled/g" /etc/selinux/config
```

Change order of name resolution in the Name Service Switch configuration file (**/etc/nsswitch.conf**). This setting allows searching a DNS Server before reading the hosts file during name resolution.

```
# cp /etc/nsswitch.conf /etc/nsswitch.conf.bkp.`date +%Y%m%d%H%M%S`
# sed -i "s/hosts:      files dns myhostname/hosts:      dns files myhostname/g"
/etc/nsswitch.conf
```

Configure, start and enable NTP.

```
# ntpserver=192.168.56.101
# cp /etc/ntp.conf /etc/ntp.conf.bkp.`date +%Y%m%d%H%M%S`
# cp /etc/sysconfig/ntpd /etc/sysconfig/ntpd.bkp.`date +%Y%m%d%H%M%S`
# sed -i "/^server/d" /etc/ntp.conf
# echo server $ntpserver iburst >> /etc/ntp.conf
# sed -i "s/-g/-x -p \/var\/run\/ntpd.pid -g/g" /etc/sysconfig/ntpd
# systemctl start ntpd
# systemctl enable ntpd
```

Create a directory that will hold Oracle Inventory, Oracle Base and Homes for Grid and Database.

```
# mkdir -p /u01
# chown -R oracle:oinstall /u01
# chmod -R 755 /u01
```

Set password for oracle user

```
# passwd oracle
```

Perform a system check as a **root** user on each cluster node, if required.

Verify that packages are installed successfully.

```
# rpm -q oracle-rdbms-server-12cR1-preinstall oracleasm-support ntp net-tools
rlwrap unzip git
```

The oracle-rdbms-server-12cR1-preinstall package generates log and makes backup of operating system files that has been changed. You can view the log in the **/var/log/oracle-rdbms-server-12cR1-preinstall** file.

Check firewall status

```
# systemctl list-unit-files | grep firewalld
# systemctl status -n 0 firewalld
```

Check if the /etc/nsswitch.conf file has "dns" before "hosts"

```
# cat /etc/nsswitch.conf | grep "hosts:      "
```

Verify that the Network Time Protocol daemon (**ntpd.service**) is enabled and up

```
# systemctl list-unit-files | grep ntp
# systemctl status -n 0 ntpd
```

Check system date

```
# date
```

Check SELinux configuration

```
# grep "^SELINUX=" /etc/selinux/config
```

Shut down both virtual machines.

Creation of shared disks for Oracle ASM

Add shared disks for Oracle ASM. Execute the following commands on the host (VirtualBox cli).

```
md "%userprofile%\VirtualBox VMs\asm-disks"
"%programfiles%\Oracle\VirtualBox\VBoxManage.exe" createhd --filename
"%userprofile%\VirtualBox VMs\asm-disks\asm-datavol1" --size 10240 --format VDI --
variant Fixed
"%programfiles%\Oracle\VirtualBox\VBoxManage.exe" createhd --filename
"%userprofile%\VirtualBox VMs\asm-disks\asm-fravol1" --size 10240 --format VDI --
variant Fixed
"%programfiles%\Oracle\VirtualBox\VBoxManage.exe" modifymedium disk
"%userprofile%\VirtualBox VMs\asm-disks\asm-datavol1.vdi" --type shareable
"%programfiles%\Oracle\VirtualBox\VBoxManage.exe" modifymedium disk
"%userprofile%\VirtualBox VMs\asm-disks\asm-fravol1.vdi" --type shareable

"%programfiles%\Oracle\VirtualBox\VBoxManage.exe" storageattach "vmrac1" --
storagectl "SATA" --port 1 --device 0 --type hdd --medium "%userprofile%\VirtualBox
VMs\asm-disks\asm-datavol1.vdi"
"%programfiles%\Oracle\VirtualBox\VBoxManage.exe" storageattach "vmrac1" --
storagectl "SATA" --port 2 --device 0 --type hdd --medium "%userprofile%\VirtualBox
VMs\asm-disks\asm-fravol1.vdi"
"%programfiles%\Oracle\VirtualBox\VBoxManage.exe" storageattach "vmrac2" --
storagectl "SATA" --port 1 --device 0 --type hdd --medium "%userprofile%\VirtualBox
VMs\asm-disks\asm-datavol1.vdi"
"%programfiles%\Oracle\VirtualBox\VBoxManage.exe" storageattach "vmrac2" --
storagectl "SATA" --port 2 --device 0 --type hdd --medium "%userprofile%\VirtualBox
VMs\asm-disks\asm-fravol1.vdi"

"%programfiles%\Oracle\VirtualBox\VBoxManage.exe" startvm "vmrac1"
"%programfiles%\Oracle\VirtualBox\VBoxManage.exe" startvm "vmrac2"
```

Prepare disks for ASM.

As disks are shared the following is required to be executed only once on one of the cluster nodes as a **root** user.

Identify newly attached shared disks.

```
[root@vmrac1 ~]# rootvol=`df -h | grep /$ | awk '{print $1}'`
```

```
[root@vmrac1 ~]# sysdev=`lvdisplay -m $rootvol | grep "Physical volume" | awk
'{print $3}' | tr -d [0,1,2,3,4,5,6,7,8,9,]`
[root@vmrac1 ~]# fdisk -l | grep -v $sysdev | grep sd
```

Create a partition on each disk.

```
[root@vmrac1 ~]# fdisk <device_name>
```

fdisk will open an interactive command line. In "Command (m for help):" type "n" to create a new partition. All subsequent inputs can be left by default, so press enter four times. When you return to "Command (m for help):" type "w" to save partition table. Repeat this task for each shared disk.

Check if the second node has caught the changes.

```
[root@vmrac2 ~]# fdisk -l /dev/sdb | grep Linux
/dev/sdb1            2048    20971519    10484736   83  Linux

[root@vmrac2 ~]# fdisk -l /dev/sdc | grep Linux
/dev/sdc1            2048    20971519    10484736   83  Linux
```

Configure the Oracle ASMLib driver

Execute on each node as a **root** user.

```
# /etc/init.d/oracleasm configure
```

ASMLib asks to indicate owner to the driver, the owner`s group and if the driver has to scan disks on boot.

```
Default user to own the driver interface []: oracle
Default group to own the driver interface []: asmadmin
Scan for Oracle ASM disks on boot (y/n) [y]: y
```

Create ASM disks

Execute on one of the cluster nodes as a **root** user.

```
[root@vmrac1 ~]# /etc/init.d/oracleasm createdisk DATAVOL1 /dev/sdb1
Marking disk "DATAVOL1" as an ASM disk:                [  OK  ]
[root@vmrac1 ~]# /etc/init.d/oracleasm createdisk FRAVOL1 /dev/sdc1
Marking disk "FRAVOL1" as an ASM disk:                 [  OK  ]
```

Scan ASM disks on another node.

```
[root@vmrac2 ~]# /etc/init.d/oracleasm scandisks
[root@vmrac2 ~]# /etc/init.d/oracleasm listdisks
```

Configure Bash Profile for the oracle user

Bash Profile for the **oracle** user will be identical on both cluster nodes except **"ORACLE_NODE_NUM"** environment variable. The variable for the second node will have value 2.

Execute as **oracle** on each cluster node.

```
$ vi ~/.bash_profile
```

Delete last two lines that export "PATH" - **"PATH=$PATH:$HOME/.local/bin:$HOME/bin"** and **"export PATH"** and insert the content of the https://github.com/georgiy-shubin/oracle-articles/blob/master/oracle_bash_profile.txt file after the "# User specific environment and startup programs" string.

When you login as the **oracle** user **.bash_profile** will print the usage tips.

Installation of Oracle Grid Infrastructure for cluster

Unpack Grid software archives as **oracle** on **vmrac1**

```
[oracle@vmrac1 ~]$ unzip linuxamd64_12102_grid_1of2.zip; unzip linuxamd64_12102_grid_2of2.zip
```

Oracle uses **cvuqdisk** to perform shared storage checks. The package is included in the Oracle Grid software distribution media.

Install **cvuqdisk-1.0.9-1.rpm** for ASM on each cluster node as a **root** user.

On vmrac1:

```
[root@vmrac1 ~]# rpm -ivh /home/oracle/grid/rpm/cvuqdisk-1.0.9-1.rpm
[root@vmrac1 ~]# scp /home/oracle/grid/rpm/cvuqdisk-1.0.9-1.rpm vmrac2:~/
```

On vmrac2:

```
[root@vmrac2 ~]# rpm -ivh ~/cvuqdisk-1.0.9-1.rpm
```

Set up ssh token authentication as **oracle** on **vmrac1**

```
[oracle@vmrac1 ~]$ ~/grid/sshsetup/sshUserSetup.sh -user oracle -hosts "vmrac1 vmrac2" -noPromptPassphrase -advanced
```

Check ssh token authentication as **oracle** user.

vmrac1:

```
[oracle@vmrac1 ~]$ ssh vmrac2
```

vmrac2:

```
[oracle@vmrac2 ~]$ ssh vmrac1
```

Run Cluster Verification Utility as **oracle** on **vmrac1**

```
[oracle@vmrac1 ~]$ mkdir ~/cvu_logs
[oracle@vmrac1 ~]$ ~/grid/runcluvfy.sh stage -pre crsinst -n vmrac1,vmrac2 -r 12.1 -asm -asmdev "/dev/oracleasm/disks/*" -verbose > ~/cvu_logs/clvfy_pre_crsinst_`date +%Y%m%d%H%M%S`.log
```

Inspect the log file for any violations.

If the virtual environment is kept identical to this manual, you will face the following issues:

- **Network consideration.** CVU identifies the NAT network (192.168.15.0) as a primary for the cluster nodes because that network has a gateway configured and proposes it as a candidate for the Public network. Despite that, the VirtualBox Host-Only

Network (192.168.56.0) will be used as Public to provide connectivity between host and cluster.

```
…
Interfaces found on subnet "192.168.15.0" that are likely candidates for VIP
are:
Interfaces found on subnet "192.168.56.0" that are likely candidates for a
private interconnect are:
Interfaces found on subnet "192.168.10.0" that are likely candidates for a
private interconnect are:
…
```

- **A memory issue.** Oracle Grid Infrastructure requires at least 4 GB for installation. Ignore this warning.

```
…
Check: Total memory
  Node Name     Available                  Required                  Status
  ------------  ------------------------  ------------------------  ---------
-
  vmrac1        3.8611GB (4048676.0KB)    4GB (4194304.0KB)         failed
  vmrac2        3.8611GB (4048676.0KB)    4GB (4194304.0KB)         failed
Result: Total memory check failed
…
```

- **UDEV for ASM.** The System uses ASMLib driver to identify shared storage, so UDEV is not necessary. Ignore it.

```
…
UDev attributes check for ASM Disks started...

ERROR:
PRVF-9802 : Attempt to get 'udev' information from node "vmrac1" failed
No UDEV rule found for device(s) specified

ERROR:
PRVF-9802 : Attempt to get 'udev' information from node "vmrac2" failed
No UDEV rule found for device(s) specified

Result: UDev attributes check failed for ASM Disks

…
Result: Devices check for ASM failed
…
```

- **/dev/shm mount status.** CVU check only the **/etc/fstab** file for the record and does not check if **tmpfs** is really mounted or not. This message can be ignored or if it is annoying add the record to fstab.

```
…
ERROR:

PRVE-0421 : No entry exists in /etc/fstab for mounting /dev/shm
PRVE-0421 : No entry exists in /etc/fstab for mounting /dev/shm
```

To modify the **/etc/fstab** file execute the following commands as a root user on each node:

```
# cp /etc/fstab /etc/fstab.bkp.`date +%Y%m%d%H%M%S`
```

```
# echo "tmpfs                   /dev/shm                tmpfs   defaults
0 0" >> /etc/fstab
```

Grid Infrastructure for a cluster requires running Oracle Universal İnstaller (OUI) only on one of the cluster nodes and installation will be pushed to second one.

To install Grid Infrastructure in the silent mode, prepare a response file. A template is included in the software distribution media.

Copy the template.

```
[oracle@vmrac1 ~]$ mkdir ~/response_files
[oracle@vmrac1 ~]$ cp ~/grid/response/grid_install.rsp
~/response_files/my_grid_install.rsp
```

Create a dictionary file with that will contain the required parameters.

```
[oracle@vmrac1 ~]$ cat <<EOF > /tmp/grid_dict
ORACLE_HOSTNAME=vmrac1
INVENTORY_LOCATION=/u01/app/oraInventory
oracle.install.option=CRS_CONFIG
ORACLE_BASE=/u01/app/oracle
ORACLE_HOME=/u01/app/grid/12102
oracle.install.asm.OSDBA=asmadmin
oracle.install.asm.OSOPER=asmadmin
oracle.install.asm.OSASM=asmadmin
oracle.install.crs.config.gpnp.scanName=rac-scan.rac.lan
oracle.install.crs.config.gpnp.scanPort=1521
oracle.install.crs.config.ClusterType=STANDARD
oracle.install.crs.config.clusterName=vmrac-cluster
oracle.install.crs.config.gpnp.configureGNS=false
oracle.install.crs.config.clusterNodes=vmrac1:vmrac1-vip,vmrac2:vmrac2-vip
oracle.install.crs.config.networkInterfaceList=enp0s8:192.168.56.0:1,enp0s9:192.168.10.0:2
oracle.install.crs.config.storageOption=LOCAL_ASM_STORAGE
oracle.install.crs.config.useIPMI=false
oracle.install.asm.SYSASMPassword=oracle
oracle.install.asm.diskGroup.name=data
oracle.install.asm.diskGroup.redundancy=EXTERNAL
oracle.install.asm.diskGroup.AUSize=1
oracle.install.asm.diskGroup.disks=/dev/oracleasm/disks/DATAVOL1
oracle.install.asm.diskGroup.diskDiscoveryString=/dev/oracleasm/disks/*
oracle.install.asm.monitorPassword=oracle
EOF
```

Execute the following command to performe batch edit of the copy of the template:

```
[oracle@vmrac1 ~]$ for i in `cat /tmp/grid_dict`; do parnam=`echo $i | cut -d "=" -
f1`; sed -i "s|^$parnam=|$i|g" ~/response_files/my_grid_install.rsp; done
```

Run OUI to execute the final prerequisite check.

```
[oracle@vmrac1]$ ~/grid/runInstaller -silent -executePrereqs -responseFile
~/response_files/my_grid_install.rsp
```

If the installer finds requirements violations, the following message will be displayed:

```
[WARNING] [INS-13014] Target environment does not meet some optional requirements.
   CAUSE: Some of the optional prerequisites are not met. See logs for details.
/tmp/OraInstall2016-07-28_01-49-32PM/installActions2016-07-28_01-49-32PM.log
   ACTION: Identify the list of failed prerequisite checks from the log:
/tmp/OraInstall2016-07-28_01-49-32PM/installActions2016-07-28_01-49-32PM.log. Then
either from the log file or from installation manual find the appropriate
configuration to meet the prerequisites and fix it manually.
```

A log of this session is currently saved as: ***/tmp/OraInstall2016-07-28_01-49-32PM/installActions2016-07-28_01-49-32PM.log****. Oracle recommends that if you want to keep this log, you should move it from the temporary location.*

Open the log file. There is the "List of failed Tasks" section at the end:

```
INFO: ------------------List of failed Tasks------------------
INFO: *********************************************
INFO: Physical Memory: This is a prerequisite condition to test whether the system
has at least 4GB (4194304.0KB) of total physical memory.
INFO: Severity:IGNORABLE
INFO: OverallStatus:VERIFICATION_FAILED
INFO: *********************************************
INFO: Device Checks for ASM: This is a prerequisite check to verify that the
specified devices meet the requirements for ASM.
INFO: Severity:IGNORABLE
INFO: OverallStatus:VERIFICATION_FAILED
INFO: *********************************************
INFO: /dev/shm mounted as temporary file system: Checks whether /dev/shm is mounted
correctly as temporary file system
INFO: Severity:IGNORABLE
INFO: OverallStatus:VERIFICATION_FAILED
INFO: -----------------End of failed Tasks List---------------
```

Failed tasks correspond to issues in the CVU report and have IGNORABLE in Severity level.

Start the installation in silent mode as the **oracle** user on vmrac1

```
[oracle@vmrac1 ~]$ ~/grid/runInstaller -silent -ignorePrereq -showProgress -
responseFile ~/response_files/my_grid_install.rsp
```

The installer will show some warnings about weak password provided for **SYS** and **ASMSNMP** users and that **OSDBA**, **OSOPER** and **OSASM** groups are same. We can ignore them as we perform test deployment and need neither role separation nor strong password.

[WARNING] [INS-30011] The SYS password entered does not conform to the Oracle recommended standards.
CAUSE: Oracle recommends that the password entered should be at least 8 characters in length, contain at least 1 uppercase character, 1 lower case character and 1 digit [0-9].
ACTION: Provide a password that conforms to the Oracle recommended standards.

[WARNING] [INS-30011] The ASMSNMP password entered does not conform to the Oracle recommended standards.
CAUSE: Oracle recommends that the password entered should be at least 8 characters in length, contain at least 1 uppercase character, 1 lower case character and 1 digit [0-9].
ACTION: Provide a password that conforms to the Oracle recommended standards.

[WARNING] [INS-41813] OSDBA for ASM, OSOPER for ASM, and OSASM are the same OS group.
CAUSE: The group you selected for granting the OSDBA for ASM group for database access, and the OSOPER for ASM group for startup and shutdown of Oracle ASM, is the same group as the OSASM group, whose members have SYSASM privileges on Oracle ASM.
ACTION: Choose different groups as the OSASM, OSDBA for ASM, and OSOPER for ASM groups.

At the end of the process, the installer asks to run two scripts as a **root** user on each node

As a root user, execute the following script(s):
*1. **/u01/app/oraInventory/orainstRoot.sh***
*2. **/u01/app/grid/12102/root.sh***

Execute /u01/app/oraInventory/orainstRoot.sh on the following nodes:
*[**vmrac1**, **vmrac2**]*
Execute /u01/app/grid/12102/root.sh on the following nodes:
*[**vmrac1**, **vmrac2**]*

and to execute the cluster configuration script for which we will must create another response file

As install user, execute the following script to complete the configuration.
*1. **/u01/app/grid/12102/cfgtoollogs/configToolAllCommands***
RESPONSE_FILE=<response_file>

Note:
1. This script must be run on the same host from where installer was run.
2. This script needs a small password properties file for configuration assistants that require passwords (refer to install guide documentation).

Create the response file:

```
[oracle@vmrac1 ~]$ cat <<EOF >> ~/response_files/cfgrsp.properties
oracle.assistants.asm|S_ASMPASSWORD=oracle
oracle.assistants.asm|S_ASMMONITORPASSWORD=oracle
oracle.crs|S_BMCPASSWORD=
EOF
```

During execution of the cluster configuration script, you can notice that time indicated in the log differs from time on cluster nodes. This is because of the stale time zone database embedded in Java that comes with Oracle software distribution media. An update of the JRE time zone will be done later, after installation of Oracle Database software.

```
[oracle@vmrac1 ~]$ date
Thu Jul 28 14:40:41 AZT 2016

[oracle@vmrac1 ~]$ /u01/app/grid/12102/cfgtoollogs/configToolAllCommands
RESPONSE_FILE=~/response_files/cfgrsp.properties
Setting the invPtrLoc to /u01/app/grid/12102/oraInst.loc

perform - mode is starting for action: configure

Jul 28, 2016 3:41:02 PM oracle.install.driver.oui.UpdateNodelistJob call
INFO: UpdateNodelist data:
Jul 28, 2016 3:41:02 PM oracle.install.driver.oui.UpdateNodelistJob call
INFO: oracle.installer.oui_loc:/u01/app/grid/12102/oui
Jul 28, 2016 3:41:02 PM oracle.install.driver.oui.UpdateNodelistJob call
INFO: oracle.installer.jre_loc:/u01/app/grid/12102/jdk/jre
Jul 28, 2016 3:41:02 PM oracle.install.driver.oui.UpdateNodelistJob call
INFO: oracle.installer.doNotUpdateNodeList:true
Jul 28, 2016 3:41:02 PM oracle.install.driver.oui.UpdateNodelistJob call
INFO: oracle.installer.rootOwnedHome:
Jul 28, 2016 3:41:02 PM oracle.install.driver.oui.UpdateNodelistJob call
INFO: OracleHomeToUpdate:/u01/app/grid/12102;isCRS:true;isCFS:false;isLocal:false
Jul 28, 2016 3:41:02 PM oracle.install.driver.oui.UpdateNodelistJob call
INFO: From map: Hosts:[vmrac1, vmrac2] => Nodelist:[vmrac1, vmrac2]
Jul 28, 2016 3:41:02 PM oracle.install.driver.oui.UpdateNodelistJob call
```

INFO: Before calling api: Hosts:[vmrac1, vmrac2] => Nodelist:[vmrac1, vmrac2], update localnode? True
…
…
…
INFO: Read: 100% complete
Jul 28, 2016 3:50:09 PM *oracle.install.driver.oui.config.GenericInternalPlugIn handleProcess*
WARNING: Skipping line: 100% complete
Jul 28, 2016 3:50:09 PM *oracle.install.driver.oui.config.GenericInternalPlugIn handleProcess*
INFO: Exceeded the number of arguments passed to stdin. CurrentCount:1 Total args:0
Jul 28, 2016 3:50:09 PM *oracle.install.driver.oui.config.GenericInternalPlugIn handleProcess*
INFO: Read: Look at the log file "/u01/app/oracle/cfgtoollogs/dbca/_mgmtdb/vmrac_cluster/_mgmtdb.log" for further details.
Jul 28, 2016 3:50:09 PM *oracle.install.driver.oui.config.GenericInternalPlugIn handleProcess*
WARNING: Skipping line: Look at the log file "/u01/app/oracle/cfgtoollogs/dbca/_mgmtdb/vmrac_cluster/_mgmtdb.log" for further details.
Jul 28, 2016 3:50:09 PM *oracle.install.driver.oui.config.GenericInternalPlugIn handleProcess*
INFO: Exceeded the number of arguments passed to stdin. CurrentCount:1 Total args:0

perform - mode finished for action: configure

You can see the log file: /u01/app/grid/12102/cfgtoollogs/oui/configActions2016-07-28_03-41-01-PM.log

[oracle@vmrac1 ~]$ **date**
Thu Jul 28 14:53:45 AZT 2016

As configuration process is completed, check cluster status by switching to Grid Infrastructure using environment switch **"ge"** and executing **crsctl** command.

[oracle@vmrac1 ~]$ **ge**
[oracle@vmrac1 ~]$ **crsctl status resource -t**

Before we proceed to Oracle Database Software installation, create an ASM Disk Group for Flash Recovery Area

[oracle@vmrac1 ~]$ **ge**
[oracle@vmrac1 ~]$ **sqlasm**
SQL> **CREATE DISKGROUP FRA EXTERNAL REDUNDANCY DISK '/dev/oracleasm/disks/FRAVOL1';**
Diskgroup created.
SQL > **exit**
[oracle@vmrac1 ~]$ **srvctl start diskgroup -diskgroup FRA -node vmrac2**
[oracle@vmrac1 ~]$ **srvctl status diskgroup -diskgroup FRA**
Disk Group FRA is running on vmrac1,vmrac2

Installation of Oracle Database Software

Unpack Database software archives as **oracle** on **vmrac1**

```
[oracle@vmrac1 ~]$ unzip linuxamd64_12102_database_1of2.zip; unzip
linuxamd64_12102_database_2of2.zip
```

Prepare a response file for the Database software.

Copy a template file

```
[oracle@vmrac1 ~]$ cp ~/database/response/db_install.rsp
~/response_files/my_db_install.rsp
```

Create a dictionary file.

```
[oracle@vmrac1 ~]$ cat <<EOF > /tmp/db_install_dict
oracle.install.option=INSTALL_DB_SWONLY
ORACLE_HOSTNAME=vmrac1
UNIX_GROUP_NAME=oinstall
INVENTORY_LOCATION=/u01/app/oraInventory
ORACLE_HOME=/u01/app/oracle/product/12102/dbhome_1
ORACLE_BASE=/u01/app/oracle
oracle.install.db.InstallEdition=EE
oracle.install.db.DBA_GROUP=dba
oracle.install.db.OPER_GROUP=dba
oracle.install.db.BACKUPDBA_GROUP=dba
oracle.install.db.DGDBA_GROUP=dba
oracle.install.db.KMDBA_GROUP=dba
oracle.install.db.rac.configurationType=ADMIN_MANAGED
oracle.install.db.CLUSTER_NODES=vmrac1,vmrac2
DECLINE_SECURITY_UPDATES=true
EOF
```

Edit the copy of the template file.

```
[oracle@vmrac1 ~]$ for i in `cat /tmp/db_install_dict`; do parnam=`echo $i | cut -d
"=" -f1`; sed -i "s|^$parnam=|$i|g" ~/response_files/my_db_install.rsp; done
```

Execute prerequisite checks

```
[oracle@vmrac1 ~]$ ~/database/runInstaller -silent -executePrereqs -responseFile
~/response_files/my_db_install.rsp
Starting Oracle Universal Installer...

Checking Temp space: must be greater than 500 MB.   Actual 9833 MB    Passed
Checking swap space: must be greater than 150 MB.   Actual 4085 MB    Passed
Preparing to launch Oracle Universal Installer from /tmp/OraInstall2016-07-28_03-
39-05PM. Please wait ...
```

OUI provides information only if the requirements are not met. Otherwise, OUI automatically removes the temporary folder marked above and provides no output.

Start the installation process and execute the configuration script as a **root** user at the end.

```
[oracle@vmrac1 ~]$ ~/database/runInstaller -silent -ignorePrereq -showProgress -
responseFile ~/response_files/my_db_install.rsp
…
…
…
```

As a root user, execute the following script(s):
*1. **/u01/app/oracle/product/12102/dbhome_1/root.sh***

Execute /u01/app/oracle/product/12102/dbhome_1/root.sh on the following nodes:
*[**vmrac1**, **vmrac2**]*

..... 100% Done.
Successfully Setup Software.

As mentioned earlier, we must update Java time zone database to apply changes in DST rules. Otherwise, some utilities such as **dbca** and **netca** will generate an incorrect time in their logs.

> **NOTE:** This procedure updates only the JRE Time Zone Database and does not include Java Virtual Machine (JVM) inside Oracle Database

You can update every instance of Java manually using Timezone Updater Tool provided by Oracle. Alternatively, you can use the script that calls the same tool and performs update automatically.
Download automated scripts from the **https://github.com/georgiy-shubin/oracle-jre-tz-autoupdate** repository. Please read the README for usage tips.

```
[oracle@vmrac1 ~]$ git clone https://github.com/georgiy-shubin/oracle-jre-tz-autoupdate
```

Download the Timezone Updater Tool from the Oracle Technology Network portal. Unpack it and copy the **tzupdater.jar** into the **oracle-jre-tz-autoupdate** directory

```
[oracle@vmrac1 ~]$ unzip ~/tzupdater-2_0_3-2015b.zip
[oracle@vmrac1 ~]$ cp ~/tzupdater-2.0.3-2015b/tzupdater.jar ~/oracle-jre-tz-autoupdate/
```

Change the mode of the scripts and copy them to the second node.

```
[oracle@vmrac1 ~]$ chmod +x ~/oracle-jre-tz-autoupdate/*.sh
[oracle@vmrac1 ~]$ scp -r ~/oracle-jre-tz-autoupdate vmrac2:~/
```

Execute the script os **oracle** user on each node to perform the Java time zone update.

```
$ cd ~/oracle-jre-tz-autoupdate/
$ ./tzupdater.sh /u01 http://www.iana.org/time-zones/repository/releases/tzdata2016f.tar.gz
```

Creation of Container Database in Oracle RAC

Prepare a response file for DBCA.

Copy a template

```
[oracle@vmrac1 ~]$ cp ~/database/response/dbca.rsp ~/response_files/my_dbca.rsp
```

Create a dictionary file

```
[oracle@vmrac1 ~]$ cat <<EOF > /tmp/dbca_dict
GDBNAME="racdb.rac.lan"
DATABASECONFTYPE="RAC"
CREATEASCONTAINERDATABASE=TRUE
NUMBEROFPDBS=0
SID="racdb"
NODELIST=vmrac1,vmrac2
SYSPASSWORD="oracle"
```

```
SYSTEMPASSWORD="oracle"
STORAGETYPE=ASM
DISKGROUPNAME=DATA
ASMSNMP_PASSWORD="oracle"
RECOVERYGROUPNAME=FRA
CHARACTERSET="AL32UTF8"
INITPARAMS="remote_listener=rac-scan.rac.lan:1521,db_recovery_file_dest_size=10239M"
MEMORYPERCENTAGE="40"
EOF
```

Edit the copy of the template file

```
[oracle@vmrac1 ~]$ for i in `cat /tmp/dbca_dict`; do parnam=`echo $i | cut -d "=" -
f1`; sed -i -e "/$parnam/s/ //g" ~/response_files/my_dbca.rsp; sed -i -e
"s/^$parnam.*/$i/g; s/#$parnam.*/$i/g;" ~/response_files/my_dbca.rsp; done
```

Start DBCA in silent mode as the **oracle** user in Database Environment

```
[oracle@vmrac1 ~]$ de
[oracle@vmrac1 ~]$ dbca -silent -responseFile ~/response_files/my_dbca.rsp
…
…
…
```

Look at the log file "/u01/app/oracle/cfgtoollogs/dbca/racdb/racdb.log" for further details.

Check Database status

```
[oracle@vmrac1 ~]$ srvctl status database -db racdb
Instance racdb1 is running on node vmrac1
Instance racdb2 is running on node vmrac2
```

Create a pluggable database.

```
[oracle@vmrac1 ~]$ sqldb
SQL> create pluggable database pdb1 admin user pdb1admin identified by oracle;
Pluggable database created.

SQL> select inst_id, con_id, name, open_mode from gv$pdbs where con_id != 2;

   INST_ID     CON_ID NAME                           OPEN_MODE
---------- ---------- ------------------------------ ----------
         1          3 PDB1                           MOUNTED
         2          3 PDB1                           MOUNTED

SQL> quit
```

To configure an automatic startup of PDB and ensure connection failover to clients, create a service. During node startup, service will open PDB in READ WRITE mode.

```
[oracle@vmrac1 ~]$ ge
[oracle@vmrac1 ~]$ srvctl add service -db racdb -service paap -preferred
"racdb1,racdb2" -role primary -failovertype SELECT -failovermethod BASIC -
failoverdelay 1 -failoverretry 120 -pdb pdb1

[oracle@vmrac1 ~]$ srvctl start service -db racdb -service paap
[oracle@vmrac1 ~]$ de
[oracle@vmrac1 ~]$ sqldb
SQL> select inst_id, con_id, name, open_mode from gv$pdbs where con_id != 2;
```

```
   INST_ID     CON_ID NAME                           OPEN_MODE
---------- ---------- ------------------------------ ----------
         1          3 PDB1                           READ WRITE
         2          3 PDB1                           READ WRITE

SQL> quit
```

Restart the CDB and check the PDB status

```
[oracle@vmrac1 ~]$ srvctl stop database -db racdb
[oracle@vmrac1 ~]$ srvctl start database -db racdb

[oracle@vmrac1 ~]$ sqldb
SQL> select inst_id, con_id, name, open_mode from gv$pdbs where con_id != 2;

   INST_ID     CON_ID NAME                           OPEN_MODE
---------- ---------- ------------------------------ ----------
         1          3 PDB1                           READ WRITE
         2          3 PDB1                           READ WRITE
```

Connect to the pluggable database from your host computer to check the failover configuration

```
C:\Windows\system32>sqlplus sys@'rac-scan.rac.lan/paap.rac.lan' as sysdba

SQL> select host_name from v$instance;

HOST_NAME
----------------------------------------------------------------
vmrac1.rac.lan
```

Shutdown **racdb1** instance with **"abort"** mode on **vmrac1** and re-execute query on your host computer once again.

On vmrac1

```
[oracle@vmrac1 ~]$ de
[oracle@vmrac1 ~]$ sqldb
SQL> shutdown abort;
```

On the host:

```
SQL> select host_name from v$instance;

HOST_NAME
----------------------------------------------------------------
vmrac2.rac.lan
```