

Deployment of Oracle RAC using VM Templates on Private Cloud Appliance

ORACLE WHITE PAPER | OCTOBER 2017





Table of Contents

Introduction	2
Architecture	3
Deployment of a 2-node Oracle RAC cluster	4
1. Connect to the Oracle VM Manager console on PCA	4
2. Create Networks	4
3. Import the VM Template for Oracle RAC	5
4. Edit the VM Template	6
5. Create Shared Storage Disk Groups	8
Create LUNs on an External Oracle FS-1 Flash Storage system	8
Create LUNs on Oracle ZFS Storage Appliance	8
6. Create VMs from VM Template	9
7. Add Shared Disks to all RAC nodes	11
8. Use the Deploycluster tool to automate RAC Deployment	11
Robustness of DeployCluster	14
Post Creation Script	16
Adding Nodes to the RAC Cluster	17
Appendix I: Customize your RAC Installation	21
Cluster Build Options (params.ini)	21
Conclusion	22



Introduction

Oracle Private Cloud Appliance is a converged infrastructure system designed for rapid and simple deployment of private cloud at an industry-leading price point. Whether customers are running Linux, Microsoft Windows or Oracle Solaris applications, Oracle Private Cloud Appliance supports consolidation for a wide range of mixed workloads in medium-to-large sized data centers. The software release 2.3.1 for PCA includes support for Oracle VM 3.4 and enables more effective utilization of local storage on compute nodes by enabling Storage Live Migration and easier memory management with hot memory modification. The release also enables support for up to 1.5 TB memory on Oracle X6-2 compute nodes and a redesigned Virtual Machine console access with HTML5.

High-performance, low-latency Oracle Fabric Interconnect and Oracle SDN allow automated configuration of the server and storage networks. The embedded controller software automates the installation, configuration, and management of all infrastructure components at the push of a button. Customers need to enter only basic configuration parameters and create virtual machines (VMs) manually or by using Oracle VM Templates to get a full application up and running.

This paper describes the end-to-end process to deploy an Oracle Real Application Cluster (RAC) 12c in minutes using Oracle VM templates on the Oracle Private Cloud Appliance (PCA). This document will take you through the installation of a 2 node Oracle RAC cluster. Two additional nodes will then be added to expand the Oracle RAC deployment to 4 nodes.

Architecture

The versions of major software components used in this setup are:

- » **Version of Oracle PCA software.** 2.3.1
- » **Version of [Deploycluser tool](#).** 3.x (for Oracle VM 3.3 and higher)
- » **Version of [Oracle VM Templates](#) used.** Single Instance & Oracle RAC 12c Release 2 Enterprise/Standard Edition, including Oracle Grid Infrastructure (12.2.0.1.0) & Oracle Linux 7 Update 4

Oracle PCA provides the ability to subdivide compute, networking and storage resources into tenant groups for easy manageability and isolation. The tenant group offers a means to isolate compute, network and storage resources per customer. It also offers isolation from cluster faults. This lab sets up 2 Oracle RAC instances on two compute nodes belonging to the same tenant group.

The 2 nodes in the Oracle RAC cluster are connected to Public and private network as shown in Fig 1. Each RAC node has one physical and 1 virtual IP. In addition there is a Public IP needed for SCAN network.

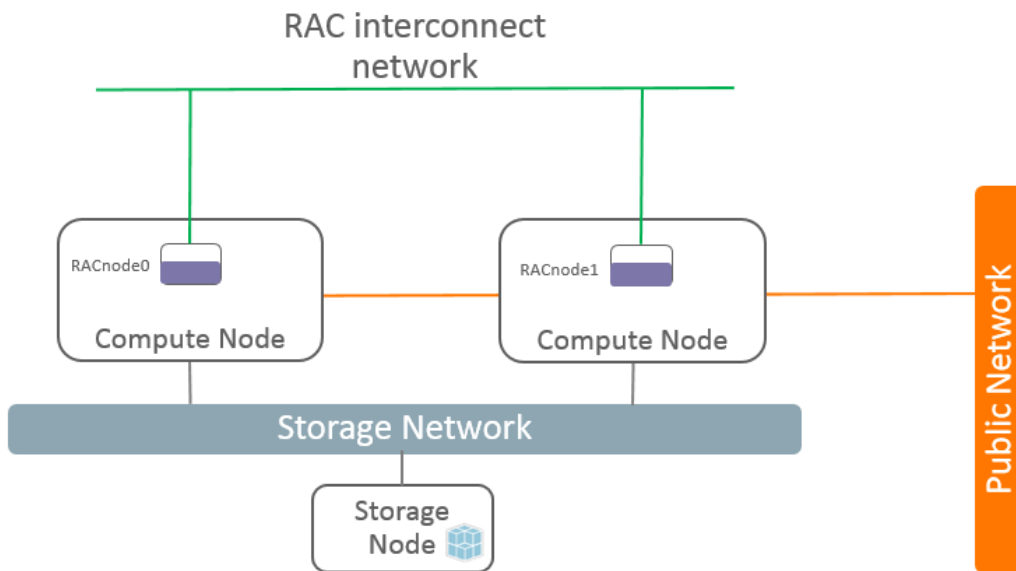


Figure 1. Architecture diagram for RAC deployment on Oracle PCA.

This lab setup uses an external FS-1 Flash storage system connected to the PCA over Fiber Channel for RAC storage.

Note: The ZFSSA internal to the PCA can also be used to create LUNs for RAC storage. However, internal ZFSSA has limited IOPS capabilities, hence it is highly recommended to use external storage for production workloads.

Deployment of a 2-node Oracle RAC cluster

[Oracle RAC](#) is an option to the award-winning Oracle Database Enterprise Edition. Oracle RAC is a cluster database with a shared cache architecture that overcomes the limitations of traditional shared-nothing and shared-disk approaches to provide highly scalable and available database solutions for all your business applications. Oracle RAC is a key component of the Oracle Enterprise grid architecture.

In this section, you will learn how to deploy a 2 node Oracle RAC 12c cluster and how to execute the following steps:

1. Connect to the Oracle VM Manager console on PCA

Point your web browser to the IP address of the virtual IP between the management nodes on your PCA and log in to the Oracle VM Manager console.

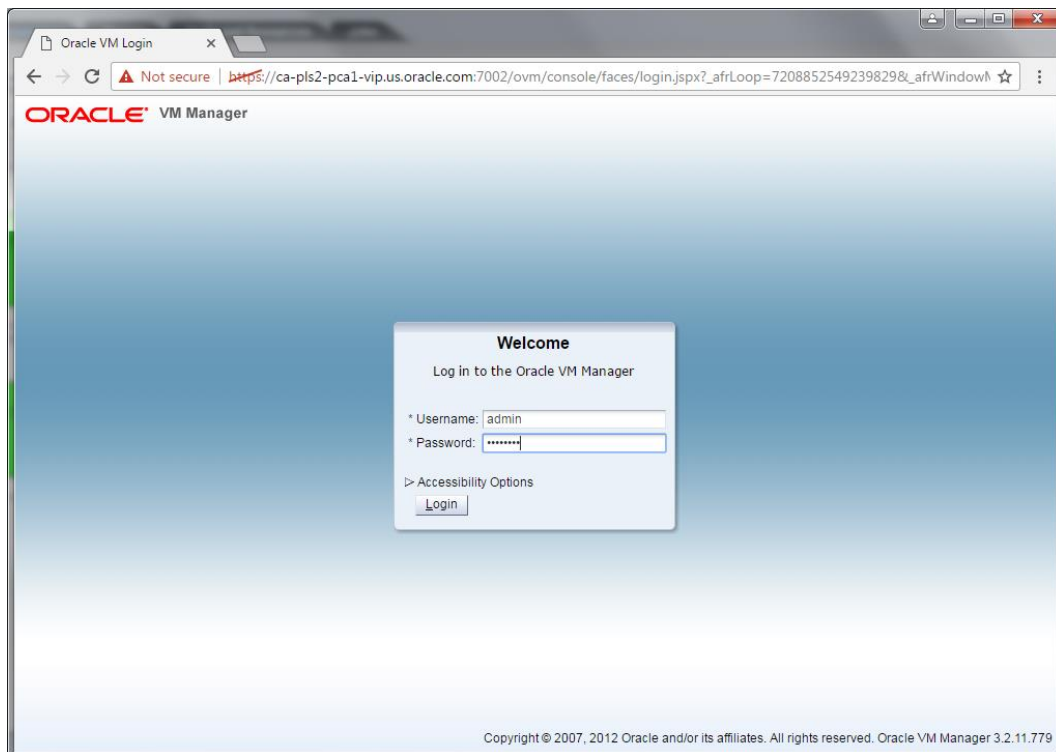


Figure 2. Oracle VM Manager login for Oracle PCA.

2. Create Networks

All nodes in an Oracle RAC environment must connect to at least one Local Area Network (LAN) (commonly referred to as the public network) to enable users and applications to access the database. In addition to the public network, Oracle RAC requires private network connectivity used exclusively for communication between the nodes and database instances running on those nodes. This network is commonly referred to as the *interconnect*. The interconnect network is a private network that connects all of the servers in the cluster.

The following 2 networks are required for the deployment of Oracle RAC on PCA.

- RAC interconnect network
- Public network

Note: The default vm_private network on PCA can be used as RAC interconnect network, but it is a good practice to have a dedicated network for this purpose.

To create a new network, go to **Networking tab** and **click on Create New Network**. Then enter the **name and choose the network channel** for the network as shown in Figure 3.

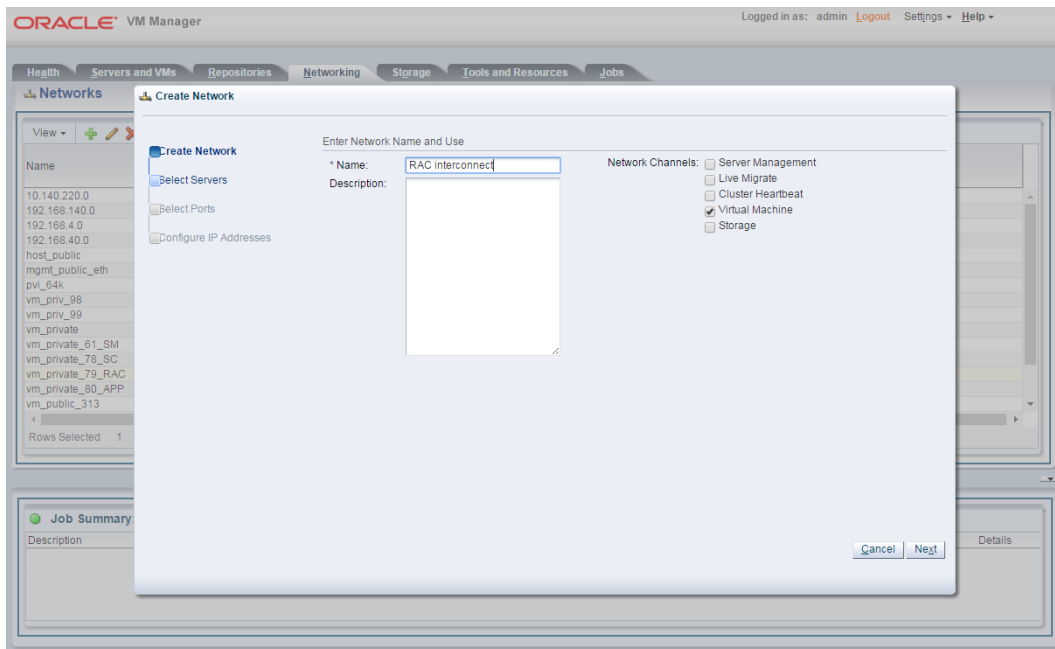



Figure 3. Creating a RAC Interconnect Network on Oracle PCA

3. Import the VM Template for Oracle RAC

Download the latest Oracle VM Templates for Oracle RAC 12c deployment from <http://www.oracle.com/technetwork/server-storage/vm/database-templates-12c-11qr2-1972804.html>

The Oracle VM templates can be used to build an Oracle Database 12c Release 2 single-instance database or a cluster that has any number of nodes—and includes Oracle Clusterware, Oracle Database, and Oracle Automatic Storage Management (ASM), patched to the latest, recommended patches.

Import the downloaded templates to the repository on PCA by following these steps:

- » Click on Repositories
- » Select the desired repository, then choose VM Templates
- » Click on 'Import VM Template' ( button). Enter the URL for the VM template location
- » Click OK

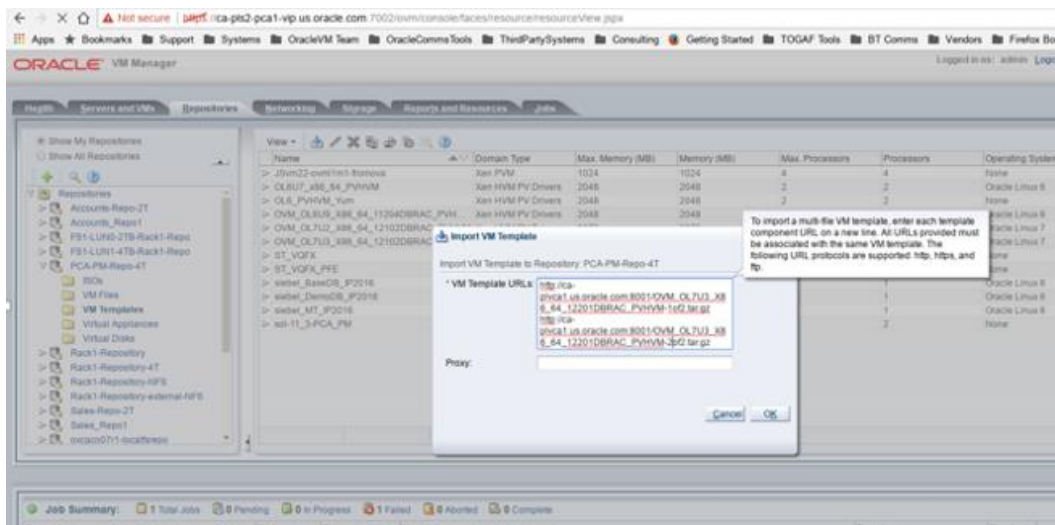


Figure 4. Importing the Oracle RAC VM Template to Oracle PCA

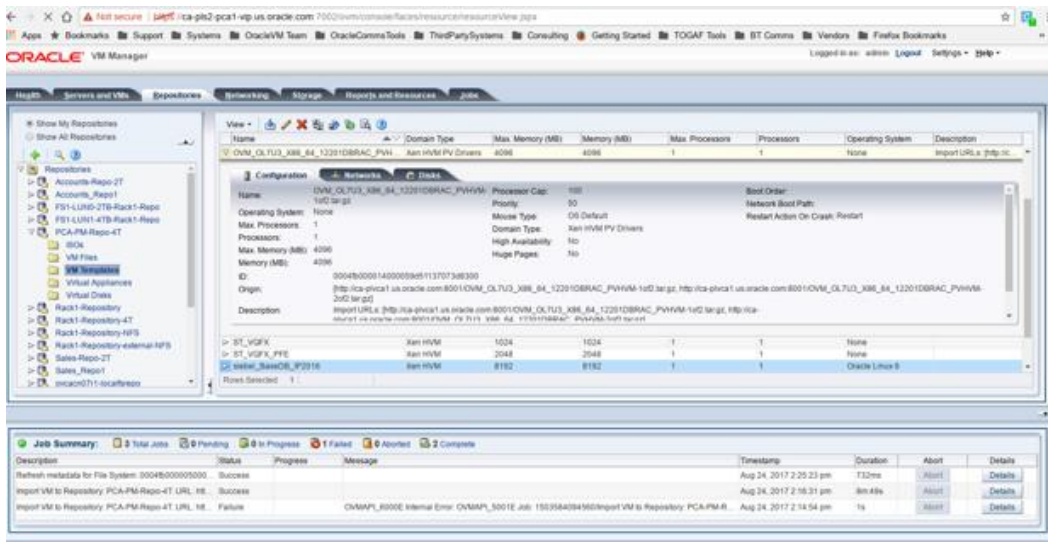


Figure 5. Import of the VM Template Complete

4. Edit the VM Template

Before cloning Virtual Machines from this template, the correct network bridges should be assigned to it.

To edit the template, follow these steps:

- » Click on **Repositories** tab and choose the repository name.
- » Select **VM templates**, pick the RAC VM template that you imported and **click Edit**. Choose the desired **VM Template name, OS running, domain type, memory and CPU options**.

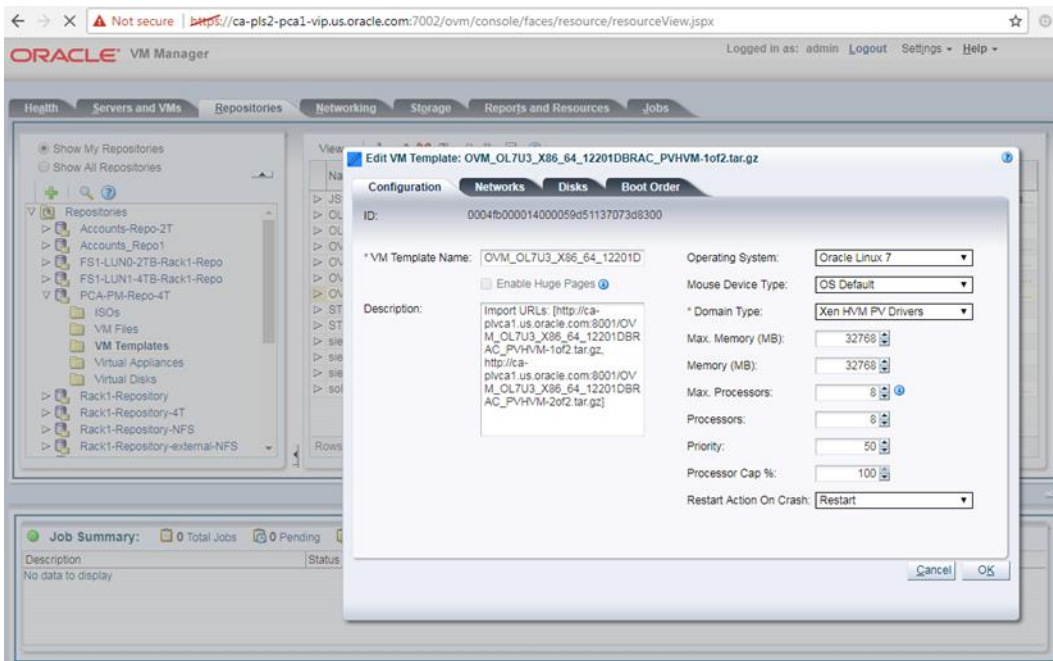


Figure 6. Edit Configuration Settings for VM Template

- » Click on the **Networks** tab and make sure the correct network bridges are shown in the Selected Ethernet Networks on the right side.

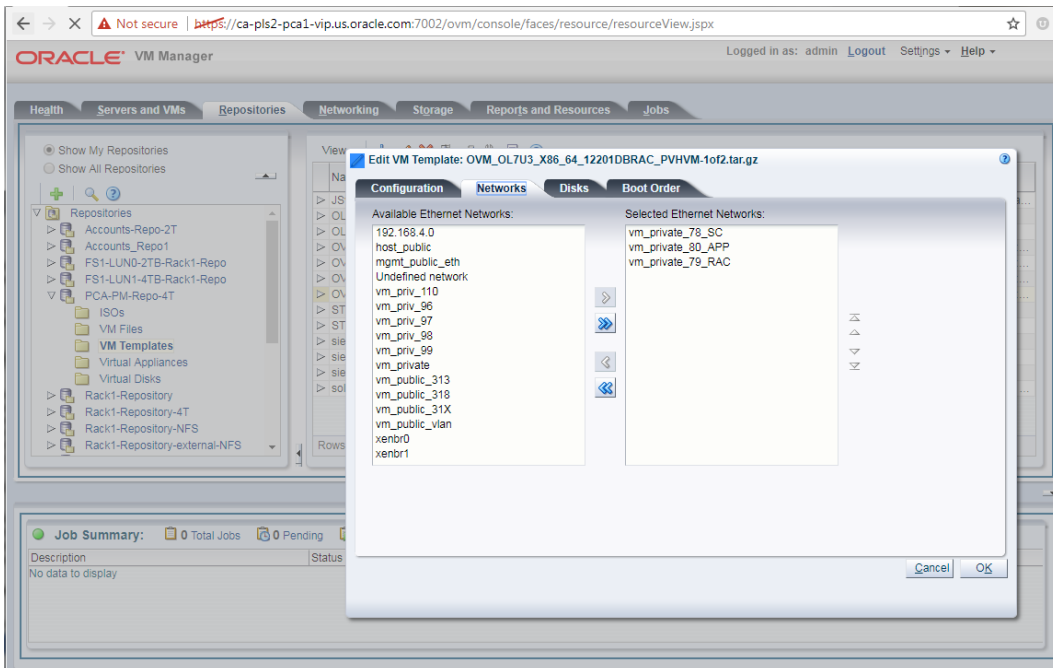


Figure 7. Edit Network Settings for VM Template

Figure 7 shows the template connected to three Ethernet networks. These VLANs represent:

- **Vm_private_78_SC.** System Management Console Access
This is normally assigned to Public Network VLAN
- **Vm_private_80_APP.** Public Network Access
This is normally assigned to Public Network VLAN
- **Vm_private_79_RAC.** RAC Cluster Interconnect
This is always assigned to a Private Network VLAN

Note: In this deployment, we used a basic Oracle Linux 7 server as a “*Bastion host*” or a Gateway server to enable access across vm_private_78_SC and vm_private_80_APP to the RAC cluster. In other words, **vm_private_78_SC and vm_private_80_APP are on a public VLAN group via the gateway server.**

5. Create Shared Storage Disk Groups

Oracle RAC utilizes Oracle Automatic Storage Management (ASM) for efficient shared storage access. ASM acts as the underlying, clustered volume manager. It provides the database administrator with a simple storage management interface that is consistent across all server and storage platforms. As a vertically integrated file system and volume manager, purpose-built for Oracle database files, ASM provides the performance of raw I/O with the easy management of a file system. Oracle Automatic Storage Management provides the basis for a shared storage pool in Oracle enterprise Grid Architectures.

This lab setup uses an external FS-1 Flash storage system connected to the PCA over Fiber Channel for RAC storage. 4 disk groups comprising of 21 Fiber Channel LUNs were provisioned on an external FS1 Flash storage system and presented to the tenant group where Oracle RAC VMs will be deployed on PCA as shown in Figure 8.

The specifications of these LUNs are:

- » **Cluster Ready Services:** 5x 5GB FC LUNs- labelled RAC_CRG_[01|05]
- » **Recovery:** 6x 25GB FC LUNs - labelled RAC_RECO_[01|06]
- » **Data:** 6x 50GB FC LUNs - labelled RAC_DATA_[01|06]
- » **Automatic Storage Management Clustered File System:** 4x 30GB FC LUNs - labelled RAC_ACFS_[01|04]

Create LUNs on an External Oracle FS-1 Flash Storage system

To create LUNs on an external FS-1 attached to PCA, please follow the [procedure](#) in Administrator Guide for Oracle Flash Storage Systems

Create LUNs on Oracle ZFS Storage Appliance

As a unified storage platform, the Oracle ZFS Storage Appliance supports access to block protocol LUNs using iSCSI and Fibre Channel protocols. The [document](#) describes the process to create LUNs on a ZFSSA using command line and ZFSSA browser user interface.

Note: internal ZFSSA has limited IOPS capabilities, hence it is highly recommended to use external storage for production workloads.

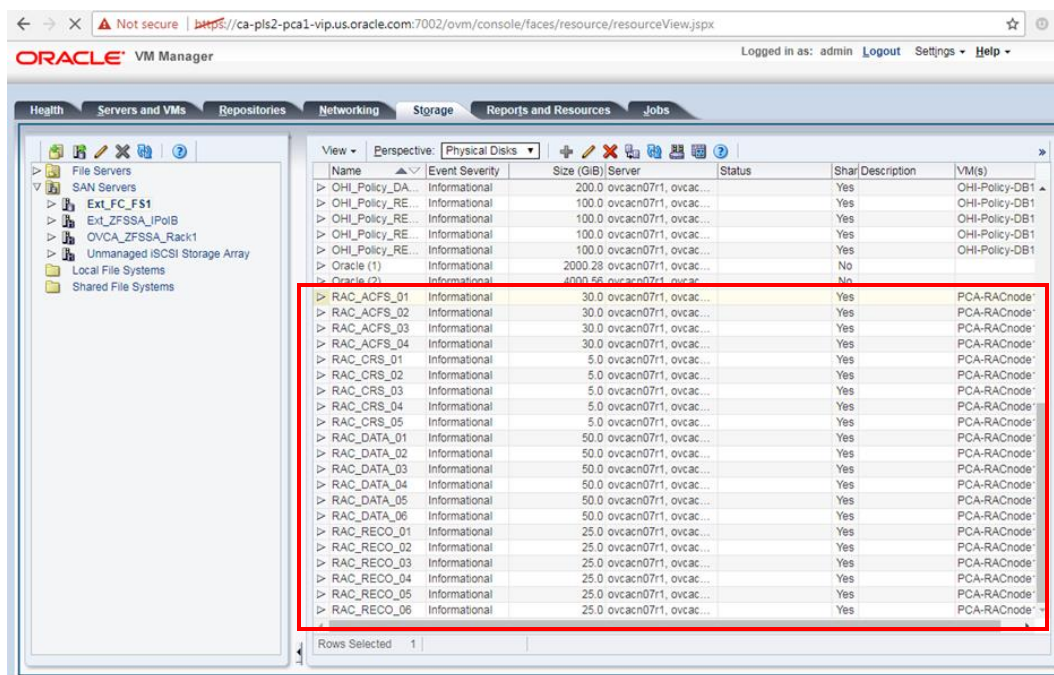


Figure 8. Shared disks for RAC storage

6. Create VMs from VM Template

The Oracle RAC template that you imported to the PCA can now be cloned to create RAC virtual machines. You will create 2 RAC nodes by cloning the Oracle RAC template.

- » Click the **Repositories** tab. In the navigation tree, select the repository in which the template resides, then **VM Templates**. Select the template in the management pane and click **Clone or Move Template**.
- » The Clone Template dialog box is displayed as shown in Figure 9. Select or enter the following:
 - **Clone to a.** Select the clone type as Virtual Machines
 - **Clone Count.** Enter 2 to create 2 RAC VMs.
 - **Clone Name:** A name for the virtual machines or templates.
 - **Target Server Pool:** The server pool on which the clone is to be deployed.
 - **Description:** A description for the virtual machines or templates.
 - **Advanced Clone Options**
 - **Clone Customizer.** The clone customizer is used to create the clones. Click **Create...** to create a new clone customizer. The clone customizer dialog box appears as shown in Figure 10. Type in a **name**, select the disks to include in the clone in **storage mappings** and the VNICs to include in the clone in **Network Mappings**. See [Section 7.8.1, “Managing Clone Customizers”](#) for more information on creating a clone customizer.
 - **Target Repository.** The repository to store the cloned files, such as virtual disks.

Note: If you clone a virtual machine or template without using a clone customizer, the storage repository is locked for the duration of the cloning job. To quickly create clones and not lock the storage repository, use a clone customizer.

» Click OK

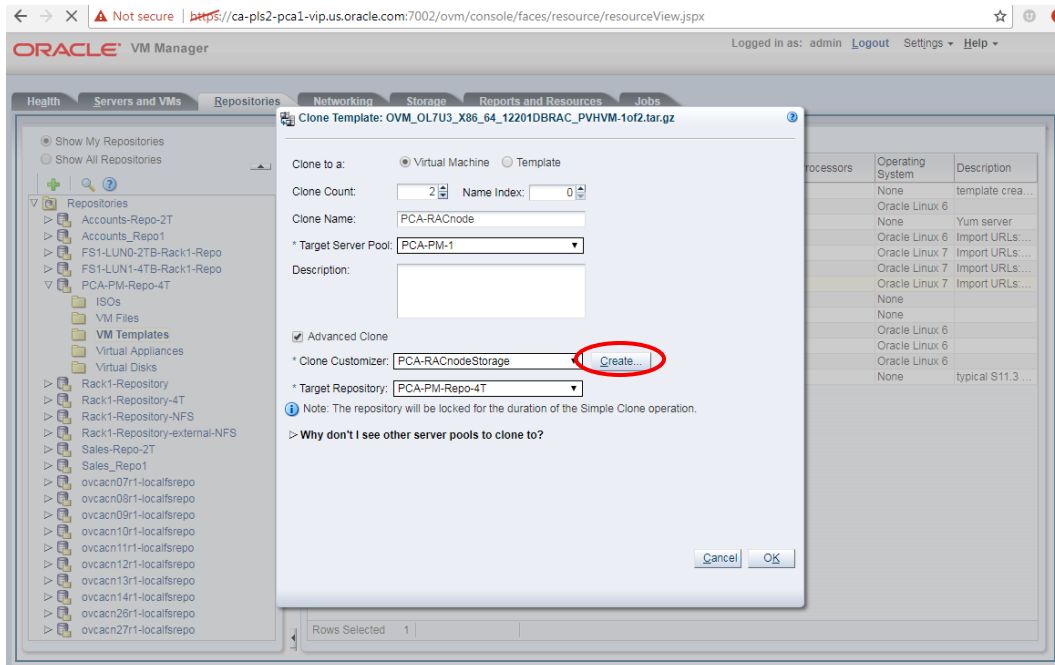


Figure 9. Clone VM Template to create two Oracle RAC nodes

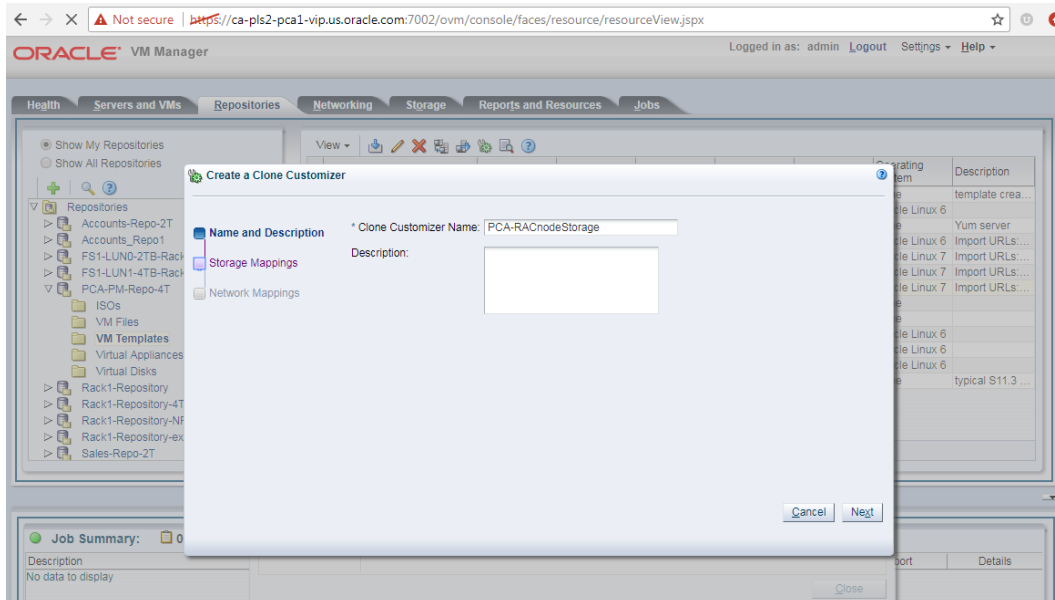


Figure 10. Create a Clone Customizer for cloning the VM Template to RAC nodes

7. Add Shared Disks to all RAC nodes

The 4 ASM disk groups with the 21 LUNs created earlier on the FS-1 have to be attached to both the RAC VMs (PCA-RACnode1, PCA-RACnode2). This can be done as follows:

- » Click the **Servers and VMs** tab. In the navigation tree, select the server pool in which the RAC nodes reside and pick the Perspective as Virtual Machines. Select the first RAC node VM in the management pane and click **Edit**.
- » Select the **Disks** tab in the **Edit Virtual Machine** dialog box. Choose the **Disk Type** (physical, virtual, CD/DVD) from the menu and then search for the correct disk using the Search button under Actions. Attach all 21 previously created RAC storage disks to this VM and click OK.
- » Repeat the process for all RAC nodes

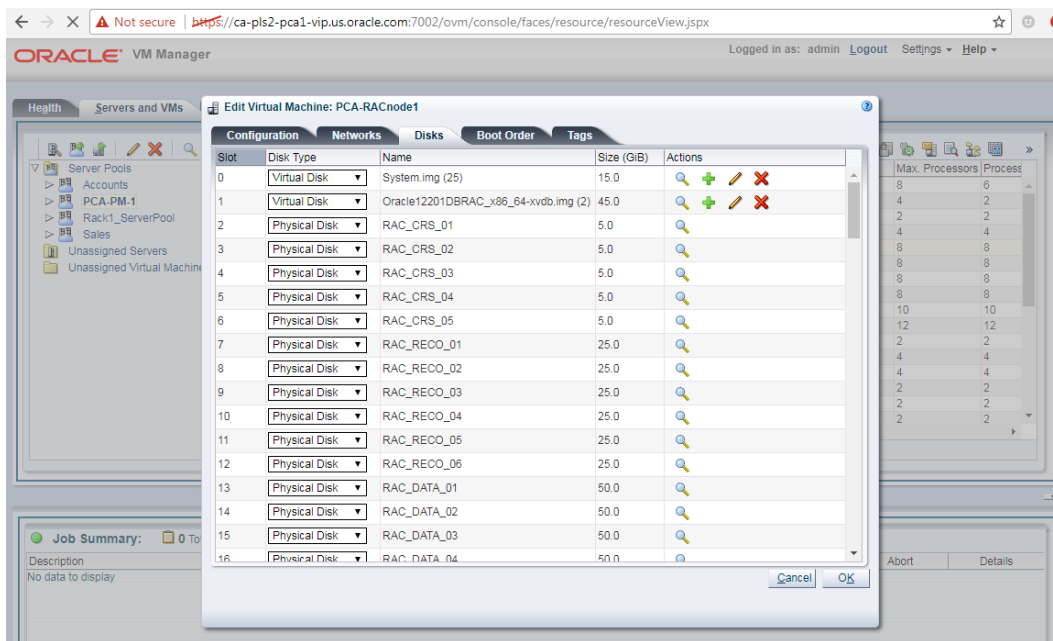


Figure 11. Edit RAC VM to attach RAC shared storage disks

8. Use the Deploycluster tool to automate RAC Deployment

The Deploycluster tool leverages the Oracle VM 3 API so that when given a set of VMs, it quickly boots them up, sends the needed configuration details, and automatically initiates a single-instance or cluster build, without requiring you to log in to any of the involved VMs, or to Oracle VM Manager.

The minimum software requirements for the Linux server where the DeployCluster tool is running from:

- Python 2.4 (or above)
- Java 1.6 1 (or above) – OpenJDK Java or GNU Compiler for Java (gcj) are not supported
- grep, tail, file, bc & bash (rpms)

Following are the steps to run Deploycluster tool:

- » Download the tool by accepting the License [here](#). Please refer to [DeployCluster](#) documentation for details about this tool.

- » Unzip it anywhere you wish, e.g.

```
$ cd $HOME
$ unzip -q DBRACOVm-Deploycluster3-tool.zip
$ cd deploycluster3
```

This will extract the Deploycluster tool (**deploycluster.py**) to be run for automated installation of Oracle RAC. In addition, the **utils** directory will have sample **netconfig.ini** (file containing IPs of all the RAC nodes and SCAN IP for the RAC) and **params.ini** files (information about RAC configuration parameters, ASM multi disk configs). The **params.ini** file can be used to customize your RAC installation as discussed in Appendix I.

- » Create the **netconfig.ini** file for your 2 node RAC deployment.

From the **utils** directory, edit a sample **netconfig.ini** file with names and IPs of your RAC environment nodes. Figure 12 shows a **netconfig.ini** files for 2 node RAC cluster deployment.

```
# Node specific information
NODE1=PCA-RACnode1
NODE1IP=192.168.80.75
NODE1PRIU=PCA-RACnode1-priv
NODE1PRIVIP=192.168.79.75
NODE1VIP=PCA-RACnode1-vip
NODE1VIP IP=192.168.80.79
NODE2=PCA-RACnode2
NODE2IP=192.168.80.76
NODE2PRIU=PCA-RACnode2-priv
NODE2PRIVIP=192.168.79.76
NODE2VIP=PCA-RACnode2-vip
NODE2VIP IP=192.168.80.80

# Common data
PUBADAP=eth1
PUBMASK=255.255.255.0
PUBGW=192.168.78.125
PRIUADAP=eth2
PRIUMASK=255.255.255.0
RACCLUSTERNAME=RACTEST
DOMAINNAME=us.oracle.com # May be blank
DNSIP="" # Starting from 2013 Templates allows multi value
# Device used to transfer network information to second node
# in interview mode
NETCONFIG_DEV=/dev/xvdc
# RAC specific data
SCANNAME=PCA-RACnode-scan
SCANIP=192.168.80.83
```

Figure 12. **netconfig.ini** file for 2-node Oracle RAC cluster

- » Run the **deploycluster tool**. Here you only supply user (**-u flag**) & password (**-p flag**) for the OVM Manager, VMs to operate on (**-M flag**), and the **netconfig.ini** file (**-N flag**) prepared in prior step.

Execute the following command on the Oracle VM Manager host and it will power ON both PCA-RACnode1 and PCA-RACnode2 VMs; then send them the netconfig.ini file and initiate a buildcluster:

```
$ ./deploycluster.py -u admin -p Welcome -M PCA-RACnode1,PCA-RACnode2 -N netconfig.ini
```

It is also possible to add the **-D flag**, to run in 'dryrun' mode to see a simulation of the operations that will be performed. Running in dryrun mode will make no changes to the environment, and this will give a good indication of whether the actual deployment will succeed.

```
Oracle DB/RAC OneCommand (v3.0.5) for Oracle VM - deploy cluster - (c)
2011-2017 Oracle Corporation
  (com: 29100:v3.0.4, lib: 231275:v3.0.5, var: 1800:v3.0.5) - v2.7.5 -
stdenni-linux (x86_64)
Invoked as stdenni at Thu Aug 24 20:30:11 2017 (size: 43900, mtime:
Tue Feb 28 00:03:00 2017)
Using: deploycluster3/deploycluster.py -H ca-pls2-pcal-
vip.us.oracle.com -u admin -p **** -I -M PCA-RACnode1,PCA-RACnode2 -N
/Support/VMTemplate-RAC/WhitePaper/netconfig-PCA-4Nodes.ini -P
/Support/VMTemplate-RAC/WhitePaper/params-PCA-12c.ini

INFO: Attempting to connect to Oracle VM Manager...

INFO: Oracle VM Client CONNECTED to Oracle VM Manager (3.4.2.1398) UUID
(0004fb0000010000dd0ca03c44151d6a)

INFO: Inspecting /Support/VMTemplate-RAC/WhitePaper/netconfig-PCA-
4Nodes.ini for number of nodes defined...
INFO: Detected 2 nodes in: /Support/VMTemplate-
RAC/WhitePaper/netconfig-PCA-4Nodes.ini

INFO: Located a total of (2) VMs;
      2 VMs with a simple name of: ['PCA-RACnode1', 'PCA-RACnode2']

INFO: Detected a RAC deployment...

INFO: Starting all (2) VMs...

INFO: VM with a simple name of "PCA-RACnode1" is in a Stopped state,
attempting to start it.....OK.

INFO: VM with a simple name of "PCA-RACnode2" is in a Stopped state,
attempting to start it.....OK.

INFO: Verifying that all (2) VMs are in Running state and pass
prerequisite checks.....

INFO: Detected that all (2) VMs specified on command line have (21)
common shared disks between them (ASM_MIN_DISKS=5)

INFO: The (2) VMs passed basic sanity checks and in Running state,
sending cluster details as follows:
      netconfig.ini (Network setup): /Support/VMTemplate-
RAC/WhitePaper/netconfig-PCA-4Nodes.ini
      params.ini (Overall build options): /Support/VMTemplate-
RAC/WhitePaper/params-PCA-12c.ini
      buildcluster: yes
```

```

INFO: Starting to send configuration details to all (2)
VM(s).....
INFO: Sending to VM with a simple name of "PCA-
RACnode1".....
.....
INFO: Sending to VM with a simple name of "PCA-
RACnode2".....
.....

INFO: Configuration details sent to (2) VMs...
      Check log (default location /u01/racovm/buildcluster.log) on
      build VM (PCA-RACnode1)...

INFO: deploycluster.py completed successfully at 20:53:10 in 1379.4
seconds (0h:22m:59s)
Logfile at: /Support/VMTemplate-RAC/WhitePaper/deploycluster1.log

```

Figure 13. deploycluster.log file for 2-node Oracle RAC deployment

Note: It is possible to monitor the buildcluster progress, by logging into the PCA-RACnode1 and looking at **/u01/racovm/buildcluster.log**. A screenshot of this log is shown in Fig 14. This log file will have all commands executed in verbose mode.

Robustness of DeployCluster

The DeployCluster script is robust. If there is an error in the process, correct the error and re-run the script. It will resume from the point it left off.

Default installation specifications for a 2-node cluster:

- **SID:** ORCL1 & ORCL2
- **DB name:** ORCL
- **Grid Infrastructure Home:** /u01/app/11.2.0/grid or /u01/app/12.1.0/grid
- **Oracle RAC Home:** /u01/app/oracle/product/11.2.0/dbhome_1 or /u01/app/oracle/product/12.1.0/dbhome_1
- **ORACLE_BASE:** /u01/app/oracle
- **Central Inventory:** /u01/app/orainventory

```

INFO (node:pca-racnode1): Running on: pca-racnode1 as root:
/u01/app/12.2.0/grid/bin/olsnodes -n -s -t
pca-racnode1      1      Active Hub      Unpinned
pca-racnode2      2      Active Hub      Unpinned
Oracle Clusterware active version on the cluster is [12.2.0.1.0]
Oracle Clusterware version on node [pca-racnode1] is [12.2.0.1.0]
CRS Administrator List: grid root
Cluster is running in "flex" mode
CRS-41008: Cluster class is 'Standalone Cluster'
ASM Flex mode enabled: ASM instance count: 3
ASM is running on pca-racnode2,pca-racnode1

```

```

INFO (node:pca-racnode1): Running on: pca-racnode1 as oracle: export
ORACLE_HOME=/u01/app/oracle/product/12.2.0/dbhome_1;
/u01/app/oracle/product/12.2.0/dbhome_1/bin/srvctl status database -d ORCL
Instance ORCL1 is running on node pca-racnode1
Instance ORCL2 is running on node pca-racnode2

```

```

INFO (node:pca-racnode1): Running on: pca-racnode1 as root:
/u01/app/12.2.0/grid/bin/crsctl status resource -t

```

```

-----
Name                Target State      Server                State details
-----
Local Resources
-----
ora.ACFS.MYACFS.advm
      ONLINE ONLINE    pca-racnode1         STABLE
      ONLINE ONLINE    pca-racnode2         STABLE
ora.ACFS.dg
      ONLINE ONLINE    pca-racnode1         STABLE
      ONLINE ONLINE    pca-racnode2         STABLE
ora.ASMNET1LSNR_ASM.lsnr
      ONLINE ONLINE    pca-racnode1         STABLE
      ONLINE ONLINE    pca-racnode2         STABLE
ora.CRS.dg
      ONLINE ONLINE    pca-racnode1         STABLE
      ONLINE ONLINE    pca-racnode2         STABLE
ora.DATA.dg
      ONLINE ONLINE    pca-racnode1         STABLE
      ONLINE ONLINE    pca-racnode2         STABLE
ora.LISTENER.lsnr
      ONLINE ONLINE    pca-racnode1         STABLE
      ONLINE ONLINE    pca-racnode2         STABLE
ora.RECO.dg
      ONLINE ONLINE    pca-racnode1         STABLE
      ONLINE ONLINE    pca-racnode2         STABLE
ora.acfs.myacfs.acfs
      ONLINE ONLINE    pca-racnode1         mounted on
      /myacfs,S
      ONLINE ONLINE    pca-racnode2         TABLE
      /myacfs,S
      ONLINE ONLINE    pca-racnode2         TABLE
ora.net1.network
      ONLINE ONLINE    pca-racnode1         STABLE
      ONLINE ONLINE    pca-racnode2         STABLE
ora.ons
      ONLINE ONLINE    pca-racnode1         STABLE
      ONLINE ONLINE    pca-racnode2         STABLE
ora.proxy_advm
      ONLINE ONLINE    pca-racnode1         STABLE
      ONLINE ONLINE    pca-racnode2         STABLE
-----
Cluster Resources
-----
ora.LISTENER_SCAN1.lsnr
      1      ONLINE ONLINE    pca-racnode1         STABLE
ora.asm
      1      ONLINE ONLINE    pca-racnode1         Started,STABLE
      2      ONLINE ONLINE    pca-racnode2         Started,STABLE
      3      OFFLINE OFFLINE    pca-racnode2         STABLE
ora.cvu
      1      ONLINE ONLINE    pca-racnode1         STABLE

```



```

ora.orcl.db
  1          ONLINE  ONLINE      pca-racnode1
Open,HOME=/u01/app/o

racle/product/12.2.0

/dbhome_1,STABLE
  2          ONLINE  ONLINE      pca-racnode2
Open,HOME=/u01/app/o

racle/product/12.2.0

/dbhome_1,STABLE
ora.pca-racnode1.vip
  1          ONLINE  ONLINE      pca-racnode1      STABLE
ora.pca-racnode2.vip
  1          ONLINE  ONLINE      pca-racnode2      STABLE
ora.qosmsserver
  1          OFFLINE OFFLINE
ora.scan1.vip
  1          ONLINE  ONLINE      pca-racnode1      STABLE
-----
-----

INFO (node:pca-racnode1): For an explanation on resources in OFFLINE state,
see Note:1068835.1
2017-08-25 08:47:37:[clusterstate:Time :pca-racnode1] Completed
successfully in 7 seconds (0h:00m:07s)
2017-08-25 08:47:37:[buildcluster:Done :pca-racnode1] Building 12cR2 RAC
Cluster
2017-08-25 08:47:37:[buildcluster:Time :pca-racnode1] Completed
successfully in 2732 seconds (0h:45m:32s)

```

Figure 14. buildcluster.log file (last few steps) for 2-node Oracle RAC deployment

Thus, the 2 node RAC cluster is successfully deployed in 45 minutes.

Post Creation Script

After successfully deploying the Oracle RAC nodes and Grid infrastructure, Figure 15 shows the execution of post installation scripts. The Database Configuration Assistant (DBCA) is an automated tool for creating and configuring a database.

```

2017-08-25 08:47:22:[createdbpostsql:Start:pca-racnode1] Running DBCA post
creation scripts

INFO (node:pca-racnode1): Running SQL Patch script
(/u01/app/oracle/product/12.2.0/dbhome_1/OPatch/datapatch) at 08:47:24...

INFO (node:pca-racnode1): Running on: pca-racnode1 as oracle: export
ORACLE_HOME=/u01/app/oracle/product/12.2.0/dbhome_1; export
ORACLE_SID=ORCL1; time
/u01/app/oracle/product/12.2.0/dbhome_1/OPatch/datapatch
SQL Patching tool version 12.2.0.1.0 Production on Fri Aug 25 08:47:25 2017
Copyright (c) 2012, 2017, Oracle. All rights reserved.

Connecting to database...OK
Determining current state...done
Adding patches to installation queue and performing prereq checks...done
Installation queue:
  Nothing to roll back

```

```
Nothing to apply

SQL Patching tool complete on Fri Aug 25 08:47:48 2017

real      0m24.082s
user      0m0.269s
sys0m0.042s

INFO (node:pca-racnode1): Executing workaround for bug 25710407 on instance
(ORCL1) at 08:47:48... (To disable, set
CLONE_WORKAROUND_BUG25710407_INIT_STATS=no)

INFO (node:pca-racnode1): Running SQL on: pca-racnode1 as oracle user using
SID: ORCL1 at: 08:47:48: EXEC dbms_stats.init_package();

SQL*Plus: Release 12.2.0.1.0 Production on Fri Aug 25 08:47:48 2017

Copyright (c) 1982, 2016, Oracle. All rights reserved.

Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit
Production

SQL>
PL/SQL procedure successfully completed.

SQL> Disconnected from Oracle Database 12c Enterprise Edition Release
12.2.0.1.0 - 64bit Production

real      0m1.172s
user      0m0.012s
sys0m0.013s
2017-08-25 08:47:50:[createdbpostsql:Done :pca-racnode1] Running DBCA post
creation scripts
2017-08-25 08:47:50:[createdbpostsql:Time :pca-racnode1] Completed
successfully in 28 seconds (0h:00m:28s)
```

Figure 15. DBCA post creation scripts

Adding Nodes to the RAC Cluster

In this section, you will expand the 2 node RAC cluster to a 4 node cluster. It is possible to add more nodes to the already existing two-node RAC cluster using the following steps:

- » Create two new VMs using the same Oracle RAC VM template ([Step 6](#) of two node RAC deployment section). Here you will create **2 new VMs: PCA-RACnode3 and PCA-RACnode4**.
- » Attach the same shared storage as the existing cluster nodes ([Step 7](#) of two node RAC deployment).
- » Edit the **netconfig.ini** from [step 8](#) (or copy it from any of the existing RAC cluster members from /u01/racovm directory to the node you plan on running deploycluster from) and add the information about the new RAC VMs. Figure 16 shows an example of the additional information to add two nodes to the RAC.

This file is used in the next step and may be renamed to any desired name ex. **netconfig-PCA-4nodes.ini**

```

NODE3=PCA-RACnode3
NODE3IP=192.168.80.77
NODE3PRIU=PCA-RACnode3-priv
NODE3PRIVIP=192.168.79.77
NODE3VIP=PCA-RACnode3-vip
NODE3VIPIP=192.168.80.81
NODE4=PCA-RACnode4
NODE4IP=192.168.80.78
NODE4PRIU=PCA-RACnode4-priv
NODE4PRIVIP=192.168.79.78
NODE4VIP=PCA-RACnode4-vip
NODE4VIPIP=192.168.80.82

```

Figure 16. Updated netconfig.ini with IP info for the 2 new RAC nodes

- » Invoke the **deploycluster** tool to boot and setup network on one of the new RAC VMs (say, PCA-RACnode3).

```

$ ./deploycluster.py -u admin -p Welcome -M PCA-RACnode3,PCA-RACnode4 -N netconfig-PCA-4nodes.ini -G '-n3,-n4'

```

Flag **-G** is used to indicate that node named 'PCA-RACnode3' should be configured as the 3rd node (indicated by -n3) from the supplied netconfig-PCA-4nodes.ini file.

Note: Anytime flag **-G** is used, the buildcluster option is automatically set to NO.

- » Now that network is up on the new node, copy the more up-to date **/u01/racovm/netconfig-PCA-4nodes.ini** from the 3rd RAC node (it should contain **all newly added nodes as well as existing nodes**) to any existing cluster member (e.g. PCA-RACnode1) where you plan to run the addnode(s) procedure from.

```

PCA-RACnode3 # scp /u01/racovm/netconfig-PCA-4nodes.ini racnode1:/u01/racovm

```

- » Finally, **run the addnode procedure** from the cluster member you copied the updated netconfig-PCA-4nodes.ini file to (PCA-RACnode1 in this case):

```

PCA-RACnode1 $ cd /u01/racovm
PCA-RACnode1/u01/racovm $ ./racovm.sh -S addnodes -N PCA-RACnode3, PCA-RACnode4 2>&1 | tee addnode-inst-3-4.log

```

The "2>&1 | tee " means save stdout and stderr to **addnode-inst-3-4.log** (shown in Figure 17), this is useful since errors are printed to stderr, so using only "tee" or ">" will only capture stdout without any possible errors. If you do not need logging you may omit all of that.

```

Invoking on pca-racnode1 as root...
Oracle DB/RAC 12c/11gR2 OneCommand (v2.1.6) for Oracle VM - (c) 2010-2017
Oracle Corporation
Cksum: [1047609428 616900 racovm.sh] at Wed 30 Aug 09:02:27 EDT 2017
Kernel: 4.1.12-94.5.7.el7uek.x86_64 (x86_64) [8 processor(s)] 32172 MB | xen |
HVM
Kit Version: 12.2.0.1.0 (RAC Mode, 4 nodes, Enterprise Edition)
Step(s): addnodes,addinstances
Node(s)/Instance(s) (-N flag): PCA-RACnode3,PCA-RACnode4
grid      440      1  0 Aug25 ?      00:02:21 /u01/app/12.2.0/grid/bin/tnslsnr
ASMNET1LSNR_ASM -no_crs_notify -inherit
grid      989      1  0 Aug25 ?      00:00:10 /u01/app/12.2.0/grid/bin/tnslsnr
LISTENER_SCAN1 -no_crs_notify -inherit
grid     4830      1  0 Aug25 ?      00:00:11 /u01/app/12.2.0/grid/bin/tnslsnr
LISTENER -no_crs_notify -inherit
oracle   23259      1  0 Aug25 ?      00:01:27 ora_dbw0_ORCL1
grid    29997      1  0 Aug25 ?      00:43:00 /u01/app/12.2.0/grid/bin/ocssd.bin
grid    30605      1  0 Aug25 ?      00:00:10 asm_dbw0_+ASM1
root    30699      1  0 Aug25 ?      00:32:28 /u01/app/12.2.0/grid/bin/crsd.bin
reboot

WARNING (node:pca-racnode1): Above Oracle processes are currently running!!!!

One of the selected steps may be a "destructive" operation, do not run if you are
not certain (add -s to avoid this confirmation). Continue [yes|no]?
2017-08-30 09:02:29:[addnodes:Start:pca-racnode1] Adding (2) nodes (PCA-RACnode3
PCA-RACnode4) to cluster
Generating only /etc/hosts & ssh/nodelist...
Modifying /etc/hosts

INFO (node:PCA-RACnode1): Network configuration completed (/etc/hosts only)...
2017-08-30 09:02:30:[setsshroot:Start:pca-racnode1] SSH Setup for the root user...

INFO (node:pca-racnode1): Running as root: /u01/racovm/ssh/setssh-Linux.sh -s -x -
c NO -h nodelist -I (setup on 4 node(s): PCA-RACnode1 PCA-RACnode2 PCA-RACnode3
PCA-RACnode4)
.
.
.
.
.

2017-08-30 09:30:16:[PCA-RACnode1,PCA-RACnode2,PCA-RACnode3,PCA-RACnode4:Done
:pca-racnode1] Cluster Verification Utility (CVU), stage: Post crsinst
2017-08-30 09:30:16:[cvupostcrs:Time :pca-racnode1] Completed successfully in 154
seconds (0h:02m:34s)

INFO (node:pca-racnode1): Current (4) active cluster members after 'addnodes'
operation (see output below):
pca-racnode1 1      Active Hub      Unpinned
pca-racnode2 2      Active Hub      Unpinned
pca-racnode3 3      Active Hub      Unpinned
pca-racnode4 4      Active Hub      Unpinned

2017-08-30 09:30:20:[addnodes:Done :pca-racnode1] Adding (2) nodes (PCA-RACnode3
PCA-RACnode4) to cluster
2017-08-30 09:30:20:[addnodes:Time :pca-racnode1] Completed successfully in 1671
seconds (0h:27m:51s)
2017-08-30 09:30:20:[addinstances:Start:pca-racnode1] Adding (2) database
instances on nodes: PCA-RACnode3 PCA-RACnode4
....
INFO (node:pca-racnode1): Running on: pca-racnode1 as oracle: export
ORACLE_HOME=/u01/app/oracle/product/12.2.0/dbhome_1; export
ORACLE_BASE=/u01/app/oracle; /u01/app/oracle/product/12.2.0/dbhome_1/bin/dbca -

```

```

silent -addInstance -gdbName 'ORCL' -instanceName 'ORCL3' -sysDBAUserName sys -
nodeName 'PCA-RACnode3' -continueOnNonFatalErrors true
Adding instance
1% complete
2% complete
6% complete
13% complete
20% complete
26% complete
33% complete
40% complete
46% complete
53% complete
66% complete
Completing instance management.
76% complete
100% complete
Look at the log file "/u01/app/oracle/cfgtoollogs/dbca/ORCL/ORCL0.log" for further
details.

INFO (node:pca-racnode1): Running on: pca-racnode1 as oracle: export
ORACLE_HOME=/u01/app/oracle/product/12.2.0/dbhome_1; export
ORACLE_BASE=/u01/app/oracle; /u01/app/oracle/product/12.2.0/dbhome_1/bin/dbca -
silent -addInstance -gdbName 'ORCL' -instanceName 'ORCL4' -sysDBAUserName sys -
nodeName 'PCA-RACnode4' -continueOnNonFatalErrors true
Adding instance
1% complete
2% complete
6% complete
13% complete
20% complete
26% complete
33% complete
40% complete
46% complete
53% complete
66% complete
Completing instance management.
76% complete
100% complete
Look at the log file "/u01/app/oracle/cfgtoollogs/dbca/ORCL/ORCL1.log" for further
details.

INFO (node:pca-racnode1): Current instances for database (ORCL) after
'addinstances' operation (see output below):
Instance ORCL1 is running on node pca-racnode1
Instance ORCL2 is running on node pca-racnode2
Instance ORCL3 is running on node pca-racnode3
Instance ORCL4 is running on node pca-racnode4

2017-08-30 09:34:12:[addinstances:Done :pca-racnode1] Adding (2) database
instances on nodes: PCA-RACnode3 PCA-RACnode4
2017-08-30 09:34:12:[addinstances:Time :pca-racnode1] Completed successfully in
232 seconds (0h:03m:52s)
2017-08-30 09:34:12:[TotalOperations:Time :pca-racnode1] Total 2 operations
completed successfully in 1903 seconds (0h:31m:43s)

```

Figure 17. addnode-inst-3-4.log (snippets) shows successful addition of 2 new RAC nodes

Thus, the additional 2 nodes PCA-RACnode3, PCA-RACnode4 are added to the Oracle RAC cluster in 31 minutes.

Appendix I: Customize your RAC Installation

Cluster Build Options (params.ini)

Before running the DeployCluster tool, you can edit `/u01/racovm/params.ini` to modify some build options (bottom part of the file). The top part of `params.ini` should be modified by advanced users or if instructed to by Oracle Support.

If using `deploycluster` tool (OVM3 only), a custom `params.ini` may be passed using the `-P` flag, it will be sent to all VMs and the `buildcluster` will then use that instead of the shipped `params.ini` inside the VM. Small samples of the options that may be modified are shown in Figure 18.

```
# The name of the ASM diskgroup, default 'DATA'
# If set to an empty string "", ASM will not be configured (see
STORAGE/Filesystem section above)
# For additional diskgroup support see ASM MULTI-DISKGROUP SUPPORT section
above.
# Default: DATA
RACASMGROUPNAME='CRS'

# Build Database? The BUILD_RAC_DATABASE will build a RAC database and
# BUILD_SI_DATABASE a single instance database (also in a RAC environment)
# Default: yes
BUILD_RAC_DATABASE=yes
#BUILD_SI_DATABASE=yes

# The Database Name
# Default: ORCL
DBNAME='ORCL'

# The Instance name, may be different than database name. Limited in length of
# 1 to 8 for a RAC DB & 1 to 12 for Single Instance DB of alphanumeric
characters.
# Ignored for Policy Managed DB.
# Default: ORCL
SIDNAME='ORCL'

# Create a 12c Container Database allowing pluggable databases to be added
# using options below, or at a later time.
# Default: no
DBCA_CONTAINER_DB=yes

# Pluggable Database name. In 'createdb' operation a number is appended at the
end
# based on count (below). In 'deletepdb' exact name must be specified here or
in
# an environment variable.
# Default: mypdb
DBCA_PLUGGABLE_DB_NAME=mypdb

# Local Listener port number
# Default: 1521
LISTENERPORT=1521

# Allows color coding of log messages, errors (red), warning (yellow),
# info (green). By default no colors are used.
# Default: NO
CLONE_LOGWITH_COLORS=no
```

Figure 18. RAC Deployment parameters in `params.ini`



Conclusion

Private Cloud Appliance is an Oracle Engineered System designed to provide a private cloud infrastructure for Oracle and non-Oracle workloads. Unlike generic, self-assembled infrastructure, the Private Cloud Appliance is engineered together with Compute, Storage, Networking and Virtualization. PCA is simple to use and designed to go from power-on to production in hours. The Oracle private cloud model is simple: it's the exact same technology on-premises and in the public cloud, providing you with choice and flexibility.

Oracle VM Templates provide an innovative approach to deploy a fully configured software stack by providing pre-installed and configured software images. In this paper, we used Oracle VM templates for rapid deployment of Oracle RAC cluster with two nodes on a PCA. The 2 node RAC cluster was successfully deployed in 45 minutes on a PCA. The Deploycluster tool also allows you to expand the RAC cluster by dynamically adding more nodes to it. In this paper, two additional nodes were added to an existing RAC cluster and the operation successfully completed in 31 minutes. Thus, Oracle RAC can be deployed on a PCA in minutes using Oracle VM templates.





Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200




CONNECT WITH US

 blogs.oracle.com/oracle

 facebook.com/oracle

 twitter.com/oracle

 oracle.com

Integrated Cloud Applications & Platform Services

Copyright © 2017, Oracle and/or its affiliates. All rights reserved. This document is provided *for* information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 1017

Deployment of Oracle RAC using VM Templates on Private Cloud Appliance
October 2017
Author: Sonit Tayal