Solution-in-a-Box: Deploying Database Systems and Application VMs on Oracle Database Appliance (Bare Metal)

A step-by-step guide
PURPOSE STATEMENT

This document provides step-by-step guide to utilize the KVM technology on Bare Metal Oracle Database Appliance (ODA) release 19.10.0.0.0. and later. It provides guidelines how to create the Application tier together with the database tier on the same ODA hardware. This concept is also known as ‘Solution-in-a-Box’.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose Statement</td>
<td>2</td>
</tr>
<tr>
<td>Oracle Database Appliance (ODA) Basics</td>
<td>4</td>
</tr>
<tr>
<td>Reasons For Virtualization</td>
<td>4</td>
</tr>
<tr>
<td>ODA Virtualization</td>
<td>4</td>
</tr>
<tr>
<td>Terminology</td>
<td>4</td>
</tr>
<tr>
<td>- Virtual Machine (VM)</td>
<td>4</td>
</tr>
<tr>
<td>- KVM guest</td>
<td>4</td>
</tr>
<tr>
<td>- DB System (KVM Database System)</td>
<td>4</td>
</tr>
<tr>
<td>- VM Storage</td>
<td>4</td>
</tr>
<tr>
<td>- Virtual Network</td>
<td>4</td>
</tr>
<tr>
<td>- Virtual Disk</td>
<td>4</td>
</tr>
<tr>
<td>- KVM Database System Networks</td>
<td>4</td>
</tr>
<tr>
<td>Solution In A Box Designs</td>
<td>4</td>
</tr>
<tr>
<td>- Application and Database installed in BM</td>
<td>5</td>
</tr>
<tr>
<td>- Database installed in BM, Application installed in KVM guest</td>
<td>5</td>
</tr>
<tr>
<td>- Database and Application in separate KVM guests</td>
<td>6</td>
</tr>
<tr>
<td>Networking Considerations</td>
<td>6</td>
</tr>
<tr>
<td>Step By Step Instructions To Create Application Guest KVMs and DB sytems</td>
<td>6</td>
</tr>
<tr>
<td>- Create an Application VM with Linux OS</td>
<td>6</td>
</tr>
<tr>
<td>- Create an Application VM with Windows OS</td>
<td>13</td>
</tr>
<tr>
<td>- Create a DB System</td>
<td>25</td>
</tr>
<tr>
<td>Solution In A Box Examples For ODA</td>
<td>31</td>
</tr>
<tr>
<td>- JDE</td>
<td>31</td>
</tr>
<tr>
<td>- EBS</td>
<td>31</td>
</tr>
<tr>
<td>- FlexCube</td>
<td>31</td>
</tr>
<tr>
<td>Concept of using ODACLI KVM Commands</td>
<td>31</td>
</tr>
<tr>
<td>- List DB System</td>
<td>31</td>
</tr>
<tr>
<td>- Show the details about a DB System</td>
<td>31</td>
</tr>
<tr>
<td>- Start a DB System</td>
<td>31</td>
</tr>
<tr>
<td>- Stop a DB System</td>
<td>31</td>
</tr>
<tr>
<td>- Delete a DB System</td>
<td>31</td>
</tr>
<tr>
<td>- List KVM guest</td>
<td>31</td>
</tr>
<tr>
<td>- Show the details about a KVM guest</td>
<td>31</td>
</tr>
<tr>
<td>- Start a KVM guest</td>
<td>32</td>
</tr>
<tr>
<td>- Stop a KVM guest</td>
<td>32</td>
</tr>
<tr>
<td>- Delete a KVM guest</td>
<td>32</td>
</tr>
<tr>
<td>Migrating OVM guests from ODA VP to KVM on ODA BM</td>
<td>32</td>
</tr>
<tr>
<td>Backup your KVM guest</td>
<td>32</td>
</tr>
<tr>
<td>APPENDIX A: Example prov.json</td>
<td>32</td>
</tr>
<tr>
<td>APPENDIX B: Example Database Shapes</td>
<td>34</td>
</tr>
</tbody>
</table>
ORACLE DATABASE APPLIANCE (ODA) BASICS

Oracle Database Appliance (ODA) is a pre-built, ready to deploy database platform. ODA is an engineered system, an integrated, full-stack solution developed for Oracle Database. Oracle hardware and software are designed to work perfectly together and are able to run crucial customer workloads faster, at lower costs, and with greater security than multivendor, on-premises ‘build your own’ solutions. Each ODA system consists of server node(s), networking, and redundant storage. ODA can be a single node model (S for small, M for Medium and in case of X6-2 L for Large) or a two node HA model (HA stands for high availability). With the HA version you get two clustered nodes and you are able to create highly available RAC databases easy and fast via either the browser user interface (BUI) or the command-line interface (CLI).

REASONS FOR VIRTUALIZATION

There are several reasons why virtualization might be considered. Virtual Machines (VMs) can be used to consolidate and to utilize resources more effectively. Rather than disable not licensed cores, you can utilize them for other workloads. This can reduce the number of physical servers, resulting in space, power maintenance reductions, and cost savings. Virtualization is a great technology that provides application and database isolation and enables bundling of applications, middleware, and databases on the same hardware.

ODA VIRTUALIZATION

ODA has provided virtualization since version 2.5. To use virtualization on ODA, it had to be deployed with a special image. Users had to decide before deploying ODA to go with either the bare metal (BM) image or the virtualized platform image (VP). To change from one layout to the other, the ODA had to be reimaged from scratch. ODA VP was only available for ODA High Availability (2 node ODA) and not for single node ODAs.

Virtualization on ODA was introduced to allow the utilization of cores not licensed for Oracle Database. To avoid leaving these cores unused, virtualization allowed to utilize them for virtual machines hosting applications, web servers and so on. To host not only the database tier but as well the application and/or web tier on the same ODA is also known as “Solution in a Box”. This concept is widely adopted and not only used for Oracle Applications like EBS, WLS, Peoplesoft, Oracle Hospitality to name a few, but as well for many customer and third party software vendor developed applications.

In the past virtualization on ODA was implemented via Oracle Virtual Machine (OVM) which is a type 1 bare metal hypervisor based on xen technology. This option will still be available for some time, but in the future, virtualization on ODA will only be available via KVM. KVM stands for Kernel-based Virtual Machine and is a type 2 host OS-based hypervisor. KVM provides modules that enable the Linux kernel to be used as a hypervisor. It is available on all ODA hardware models (single node and HA). KVM is a mature technology that has been around since 2007 and offers several advantages. To name one, ODA will not need separate BM and VP images anymore.

Starting with ODA 19.9, ODA integrated KVM support for applications. A user can use ODA’s built-in BUI or the command line to manage the Application VM life cycle. The user is responsible for managing the application inside the KVM, for the ODA software tooling it is a ‘black box’. Starting with ODA 19.10, ODA integrates KVM support for Oracle databases. A user can use ODA’s built-in BUI or the command line to manage both DB System (ODA’s database KVM system) life cycle and databases inside DB Systems. DB Systems allow Oracle hard partitioning licensing, thus a user only needs to have Oracle Database licenses for the CPU cores assigned to the DB System.

TERMINOLOGY

To understand the terminology used in this document or when reading the Oracle documentation, here is a list of the often used terms:

- Virtual Machine (VM): image and virtual resources that behave like an actual server within the host server
- KVM guest: the software part of an application VM with an independent OS instance
- DB System (KVM Database System): a database VM which runs an Oracle Database on ODA. Created and managed by ODA tooling
- VM Storage: central location for storing resources needed to create virtual machines
- Virtual Network: separate virtual network for the VM
- Virtual Disk: additional storage option for VMs to add additional block storage
- KVM Database System Networks: default networks created on ODA, which are the ‘public network’ (vnetwork pubnet) and the internal managed ‘cluster interconnect’ and ASM network

SOLUTION IN A BOX DESIGNS
There are three basic scenarios on how to install the application along with the database in the same ODA. For an ODA HA there are additional designs since your database can be a RAC or a single instance database on one node only and depending on your application you might be able to run it on one or both nodes. The basic scenarios are:

**Application and Database installed in BM**

This option is only applicable for a few applications which do not interfere with the database software and OS installation. One supported example is the Oracle Enterprise Manager software.

**Database installed in BM, Application installed in KVM guest**

This option doesn't allow hard partitioning, means there is no separation of cores for database and application possible.
**DATABASE AND APPLICATION IN SEPARATE KVM GUESTS**

This third option is ideal to separate resources for databases and applications and license them independently. This step-by-step guide covers an example of the third case since this will be the best solution for most use cases.

**NETWORKING CONSIDERATIONS**

ODA KVM vnetwork supports two type of networks, bridged and bridged-vlan. For bridged network a bridge is created and a NIC or a bond is attached to it. On the public network (selected during appliance deployment) a bridge with the name pubnet is attached. If a VM needs access to pubnet it can be attached to this network. This pubnet vnetwork cannot be modified or deleted.

If additional bridged vnetwork are required they can be created on other interfaces that aren’t used for pubnet. For DB Systems it isn’t possible at the moment (19.10) to use a Vnetwork other than pubnet. Vlans can be created on all available public interfaces, including the interface where public network is already configured.

When planning to run one or more DB Systems, be aware that every DB System requires it’s own set of VIPs and SCAN IPs.

**STEP BY STEP INSTRUCTIONS TO CREATE APPLICATION GUEST KVMS AND DB SYTEMS**

**Create an Application VM with Linux OS**

Below an example how to setup an “Application VM” or “Compute Instance” with Linux Operating System. While some of the steps are mandatory others are optional:

1.) create a VM CPU Pool (optional)
2.) Create VM Storage
3.) Create a Virtual Network (optional)
4.) Create Virtual Disk (optional)
5.) Create the VM Instance
6.) Finish the OS installation
7.) Test the application VM

If you execute the optional steps depends on how you designed your Solution-in-a-Box. If your application VM should only use a defined number of CPUs you should for example create a VM CPU Pool as part of the process. To create VM Storage is mandatory, several Application VMs can share a single VM Storage. If you have specific network requirements you can create additional Virtual Networks for these purpose. Another optional step is the creating of Virtual Disks. One “VM disk” is created for each application guest KVM as part of the VM instance creation, additional Virtual Disks can be created if your Application VM requires this, they can be shared between different Application VMs if required.

1.) create a VM CPU Pool (optional)
Create a CPU Pool to specify how many CPU cores your KVM should use, this CPU Pool will be later assigned to the KVM Guest. Under Appliance go to CPU Pool and define the name the number of CPU cores and as CPU Pool Type select ‘VM Instance’.

Alternative cli command:
```bash
# odacli create-cpupool -n testcpupool -c 2 -vm
```

2.) Create VM Storage
Under Appliance go to VM Storage and define name, size and ASM disk group.
3.) Create a Virtual Network (if required)

Any VM that needs to access the public network has to be attached to the pubnet Vnetwork. No other bridged network than the default pubnet bridge is allowed to be created on the public network interface. Additional bridged vnetwork can be created on other interfaces that are not used for the public network. It isn’t mandatory to assign IPs to the bridge. This is only required if communication between the BM host and the KVM guest is desired. When 2 IP addresses are assigned to a HA machine, the first will be assigned to the bridge on node0 and the second IP to node1.

```
Alternative cli command:
```

4.) Create Virtual Disk (if required)

During the VM creation a VM disk will be created to host the OS. In case additional disks are required you should create them in advance. Using virtual disks gives you more control over the storage layout of your VM.

```
Alternative cli command:
# odacli create-vdisk -n testvmdisk -vms testvmstor -s 49G -sh
```

5.) Create the VM Instance
To create the VM go to ‘VM Instances’ and ‘Create VM’. Type in the mandatory information for your VM plus the additional configuration you would like to use for your VM guest. Mandatory information is the ‘VM name’, ‘VM Storage Name’, the path to the application to install, under ‘Source installation’. You have as well to define how many CPUs and how much memory to use for the VM. If you have created a CPU Pool a Virtual Disk or a Virtual Network, select them into the respective fields. Specify the size of the OS disk to be created for your VM guest in “VM Disk Size”. If you need your VM to access the DB System via the pubnet, make sure ‘pubnet’ is selected in the ‘Virtual Networks’.

Alternative cli command:

```
# odacli create-vm -n testvm2 -vc 2 -m 8G -wms testvmstor -vd testvmdisk -s 49G -cp testcpupool -vm pubnet -src /u01/software/oel77.iso
```

6.) Finish the OS installation

If the creation of the VM was successful, Use the odacli describe-vm command to check the VNC port.

```
# odacli describe-vm -n testvm2
```

<table>
<thead>
<tr>
<th>VM details</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID: 4956584d-7409-4d38-8a22-60653a6c2ae1</td>
</tr>
<tr>
<td>Name: testvm2</td>
</tr>
<tr>
<td>Created: 2021-05-10 02:24:01 HDT</td>
</tr>
<tr>
<td>Updated: 2021-05-10 02:24:01 HDT</td>
</tr>
<tr>
<td>VM Storage: testvmstor</td>
</tr>
<tr>
<td>Description: NONE</td>
</tr>
<tr>
<td>VM size: 49.00 GB</td>
</tr>
<tr>
<td>Source: oel77.iso</td>
</tr>
<tr>
<td>OS Type: NONE</td>
</tr>
<tr>
<td>OS Variant: NONE</td>
</tr>
<tr>
<td>Graphics settings: vnc,listen=127.0.0.1</td>
</tr>
<tr>
<td>Status</td>
</tr>
</tbody>
</table>

9 WHITE PAPER | Solution-in-a-Box: Deploying Database Systems and Application VMs on Oracle Database Appliance (Bare Metal) Version 1.1
Copyright © 2021, Oracle and/or its affiliates | Public
Current node: node1  
Current state: ONLINE  
Target state: ONLINE

### Parameters

<table>
<thead>
<tr>
<th>Preferred node</th>
<th>NONE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boot option</td>
<td>NONE</td>
</tr>
<tr>
<td>Auto start</td>
<td>YES</td>
</tr>
<tr>
<td>Fail over</td>
<td>YES</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Config</th>
<th>Live</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory: 8.00 GB</td>
<td>8.00 GB</td>
</tr>
<tr>
<td>Max Memory: 8.00 GB</td>
<td>8.00 GB</td>
</tr>
<tr>
<td>vCPU count: 2</td>
<td>2</td>
</tr>
<tr>
<td>Max vCPU count: 2</td>
<td>2</td>
</tr>
<tr>
<td>CPU Pool: testcpupool</td>
<td>testcpupool</td>
</tr>
<tr>
<td>vCPUs: 0:2-3,38-39</td>
<td>0:2-3,38-39</td>
</tr>
<tr>
<td>vDisks: testvmdisk:vdb</td>
<td>testvmdisk:vdb</td>
</tr>
<tr>
<td>vNetworks: pubnet:52:54:00:7d:7b:6b</td>
<td>pubnet:52:54:00:7d:7b:6b</td>
</tr>
</tbody>
</table>

7.) Connect to the ODA BM node via VNC

To find out the VNC port the BM (not the KVM guest) uses run the `vncserver` command.

```
# vncserver
perl: warning: Setting locale failed.
pperl: warning: Please check that your locale settings:
   LANGUAGE = (unset),
   LC_ALL = (unset),
   LC_CTYPE = "UTF-8",
   LANG = "en_US.UTF-8"
     are supported and installed on your system.
pperl: warning: Falling back to the standard locale ("C").

You will require a password to access your desktops.

Password: 
Verify: 
Would you like to enter a view-only password (y/n)? n
A view-only password is not used
xauth: file /root/.Xauthority does not exist

**New 'node1:1 (root)' desktop is node1:1  ➕ VNC port**

Creating default startup script /root/.vnc/xstartup
Creating default config /root/.vnc/config
Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/node1:1.log

Now use a VNC client of your choice and connect to host:vncport (the host is your ODA BM, the VNC port is from vncserver command).
When you see the command prompt in the VNC viewer type (from the example output) : `vncviewer 127.0.0.1:2` (this is the Display Port: 127.0.0.1:2 from the describe-vm output)

8.) Complete the installation process as you would do with any other OS installation. In the 'Device Selection', you see the disk created by `create-vm` and the additional one from `create-vdisk`. Make sure to use the right one for the OS install.
After you are asked and you confirmed to reboot, you have to wait a short time and VNC back in order to complete the setup (e.g., accept the license agreement).

9.) Configure the Application VM network

After the Linux OS installation is completed for the Application VM, the VM might have several interfaces. In my example I want to assign the IP address to the pubnet and have to find the MAC address for this interface.

```
# odacli describe-vm -n testvm2 | grep vNetworks
vNetworks:  pubnet:52:54:00:7d:7b:6b  pubnet:52:54:00:7d:7b:6b
```

Login to the Application VM via VNC client and run the command "ip link show" to identify the interface with the MAC address found with the command above.

```
# ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT group default qlen 1000
   link/ether 52:54:00:7d:7b:6b brd ff:ff:ff:ff:ff:ff
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN mode DEFAULT group default qlen 1000
   link/ether 52:54:00:42:63:d7 brd ff:ff:ff:ff:ff:ff
4: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast master virbr0 state DOWN mode DEFAULT group default qlen 1000
   link/ether 52:54:00:42:63:d7 brd ff:ff:ff:ff:ff:ff
```

Edit the `/etc/sysconfig/network-scripts/ifcfg-eth0` and `/etc/resolv.conf` for DNS inside the VM. Refer to your operating system specific document to configure the network interface. The following is just an example.

```
# cat /etc/sysconfig/network-scripts/ifcfg-eth0
TYPE=Ethernet
BOOTPROTO=none
DEFROUTE=yes
NAME=eth0
DEVICE=eth0
NM_CONTROLLED=no
ONBOOT=yes
IPADDR=<your IP address>
PREFIX=<your subnet prefix>
GATEWAY=<your gateway>

# cat /etc/resolv.conf
search us.oracle.com
nameserver <your first DNS server>
nameserver <your second DNS server>
```
10.) Test if the desired connectivity to the VM guest
Test that you are able to connect from an to all machines and clients (database, external machines etc.). To connect as root via ssh you might have to change SELINUX to permissive mode in /etc/selinux/config.
If it is desired, test this connectivity as well after you migrated the VM guest to the other node. To migrate the node execute the below commands.

```bash
# odacli describe-vm -n testvm2 | grep 'Current node'
Current node: node1

# odacli migrate-vm -n testvm2 -to node2

# odacli describe-vm -n testvm2 | grep 'Current node'
Current node: node2
```

Create an Application VM with Windows OS
Before you start, get the 'Oracle VirtIO Drivers Version for Microsoft Windows' which you can download Oracle Software Delivery Cloud or from My Oracle Support (MOS).
For details and the links please refer to https://docs.oracle.com/en/operating-systems/oracle-linux/kvm-user/kvm-virtio.html#kvm-virtio-download.

1.) Create the VM Storages for the KVM. Connect to your ODA as root and execute the below command.

```bash
# odacli create-vmstorage -n winvm -s 100G -dg DATA
```

2.) Make sure this is completed before you create a Vdisk for Windows OS

```bash
# dd if=/dev/zero of=/u05/app/sharedrepo/winvm/win.img bs=1M count=40000
```

Now use a VNC client of your choice and connect to host:vncport from the output above.

3.) Connect to the ODA node via VNC
If there isn’t the VNC server running, start it with 'vncserver' command:

```bash
# vncserver
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
    LANGUAGE = (unset),
    LC_ALL = (unset),
    LC_CTYPE = "UTF-8",
    LANG = "en_US.UTF-8"
are supported and installed on your system.
perl: warning: Falling back to the standard locale ("C").

You will require a password to access your desktops.

Password:
Verify:
Would you like to enter a view-only password (y/n)? n
A view-only password is not used
xauth: file /root/.Xauthority does not exist

New ’node1:1 (root)’ desktop is node1:1

Creating default startup script /root/.vnc/xstartup
Creating default config /root/.vnc/config
Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/node1:1.log
```
4.) In the VNC session start virt-manager

```
# virt-manager
```

In the window that pops up, select ‘File’ -> ‘New Virtual Machine’.

5.) Choose the Windows installer source (Local install media)

6.) Select the path of your Windows iso and select the correct Windows version (after de-selecting auto detection)
7.) Assign number of CPUs and the amount of memory which is temporary. In step 20.) you will create the actual VM and set the values finally used.

8.) In next window, select option ‘select or create custom storage’ and browse for the vdisk created in step 2
9.) Type in the desired name for your KVM, and select ‘Customize configuration before install’. Ensure the Vnet you want to use for the KVM is selected.

10.) Change Disk 1's (boot disk) bus from IDE to VirtIO (in Advanced option) and apply the change.

11.) Select ‘add Hardware’ and add an additional CDROM drive and add in source path the ‘Oracle VirtIO Drivers’ you downloaded at the beginning.
12.) On the left highlight ‘Boot Options’ and make all drives (disk and cdrom) active

13.) Select ‘Begin Installation’ in the left top corner of the window and confirm the OS you want to install.
14.) After a few confirmation windows the ‘Where do you want to install Windows?’ will appear. Select ‘load driver’ option and navigate to the VirtIO driver on E: CDROM drive and choose viostororcl.inf on the list
15.) After loading the driver the pre-created Vdisk will be visible. Highlight it and continue with the installation.

16.) After completing the installation, login to Windows
In case you don’t see ‘SendKey’, move with the mouse to the top of the window until 2 icons appear and press the left one to send ‘Ctrl+Alt+Delete’.

17.) Install the VirtIO drivers by running Setup from the CDROM drive E:
18.) Select not to restart the computer and shut it down instead via the Windows Start button

19.) In the virt-manager select ‘Show virtual hardware details’ and remove the cdrom (CDROM 2) which mounted the virtio iso image
20.) Open the BUI and make the KVM guest DCS managed.

With ODA version 19.10 do not specify any network settings in BUI, otherwise VM creation could fail due to internal Bug 32086685. You can assign the network with odacli modify_network or via the BUI after creating the VM successfully. CLI is not affected by the same issue.

In case the VM Storage was created too small to host the source and the final image file, increase the VM Storage either via BUI

Or with the modify-vmstorage command, e.g.:

```
#modify-vmstorage -i 50G -n winvm
```

Go to 'Compute Instances' -> 'VM Instances' and select 'Create VM Instance'. Fill in the information for name, VM Storage Name (winvm), cpupool (if required), number of CPUs and amount of memory. As Installation Source, choose the image you have already created via virt-manager (/u05/app/sharedrepo/winvm/win.img). There are additional fields which are optional.
21.) Confirm the VM was created and is running. Afterwards stop the VM

```
# odacli describe-vm -n win2k19
VM details
-----------------------------------------------
ID: f8256d17-320b-4108-946c-d38172469e7b
Name: win2k19
Created: 2021-03-10 09:53:34 HST
Updated: 2021-03-10 10:03:44 HST
VM Storage: winvm
Description: NONE
VM size: 39.00 GB
Source: win.img
OS Type: win2k19
OS Variant: NONE
Graphics settings: vnc,listen=127.0.0.1
Display Port: 127.0.0.1:1

Status
Current node: node1
Current state: ONLINE
Target state: ONLINE
```

```
# odacli stop-vm -n win2k19
```

22.) Connect with VNC to the node where the VM is running and in virt-manager highlight the Windows VM and open the VM details
23.) Select the NIC and adjust the settings for Network source to ‘Bridge pubnet’ and Device model to virtio and apply the change.

24.) Power on the VM on the node of your choice

```
odacli start-vm -n win2k19 -nn node1
```

25.) VNC back to your VM and configure it according to your needs. For example assign a fixed IP address.

After disabling the Windows firewall you should be able to ping the BM host IP from the KVM guest and vice versa.
Create a DB System

ODA version 19.10 is the minimum software version requirement to deploy a KVM DB System on ODA. To deploy or upgrade your ODA version 19.10 follow the Oracle documentation. The documentation is as well the source for more detailed information like e.g. restrictions. One of the main restrictions in ODA version 19.10 for example is that you can only have one databases per DB System and the database version can only be 19.10. In addition each DB System has a dedicated CPU Pool that cannot be shared. Since these restrictions might be subject to change, please review the Oracle documentation before you start with the DB System creation. The space reserved in ACFS for one DB system is fixed to a size of 200GB. This is the space assigned for the Operation System, the Grid Infrastructure (GI) home and the database home of the DB System. The database files are stored in ASM of the bare metal ODA to which the DB System GI connects as a client.

1.) Confirm that your ODA is minimum version 19.10.0.0.0

```
# /opt/oracle/dcs/bin/odacli describe-component -v
System Version
---------------
19.10.0.0.0
.
```

2.) Download the DB System image (Patch 32451228: ORACLE DATABASE APPLIANCE 19.10.0.0.0 DBSYSTEM IMAGE DOWNLOAD FOR KVM) into a temporary directory to your ODA.

3.) Unzip the the file

```
#unzip p32451228_1910000_Linux-x86-64.zip
odacli-dcs-19.10.0.0.0-210222-ODAVM-19.10.0.0.zip
```

4.) Update the ODA repository

```
#/opt/oracle/dcs/bin/odacli update-repository -f /tmp/odacli-dcs-19.10.0.0.0-210222-ODAVM-19.10.0.0.zip
{
  "jobId" : "6ba263d6-c32f-415f-942d-f6b6a495eb7e",
  "status" : "Created",
  "message" : "/tmp/odacli-dcs-19.10.0.0.0-210222-ODAVM-19.10.0.0.zip",
  "reports" : [ ],
  "createTimestamp" : "March 07, 2021 22:18:39 PM HST",
  "resourceList" : [ ],
  "description" : "Repository Update",
}
5.) Confirm the DB System image is correctly registered:

```
# /opt/oracle/dcs/bin/odacli describe-dbsystem-image
```

DB System Image details

<table>
<thead>
<tr>
<th>Component Name</th>
<th>Supported Versions</th>
<th>Available Versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBVM</td>
<td>19.10.0.0.0</td>
<td>19.10.0.0.0</td>
</tr>
<tr>
<td>GI</td>
<td>19.10.0.0.210119</td>
<td>19.10.0.0.210119</td>
</tr>
<tr>
<td>DB</td>
<td>19.10.0.0.210119</td>
<td>19.10.0.0.210119</td>
</tr>
</tbody>
</table>

6.) Create the DB System

Decide on the database shape according to your DBClass (OLTP, DSS or IMDB) resource requirements and available licenses. Keep in mind that the 'shape' parameter defines the number of cores you assign to the DB System. For a complete list of DB shapes please review the ODA documentation. (The table for OLTP Database Shapes X8-2 can be found in Appendix B). You can use the BUI to create a DB System or you can create a json file with the details and kick off the creation from the CLI. In BUI go to Appliance -> DB Systems and select the 'Create DB System' button.

The information required to build a DB System is partitioned in three sections: ‘System Information’
The middle section covers the network information.

In next section you can customize users and groups and decide if you want to use OS role separation. In ODA version 19.10 you have to use the same setting here as in the BM Grid Infrastructure installation. If in your BM you have a user ‘grid#’ and a user ‘oracle’, you are using role separation. In the lowest section provide the information about the database you would like to create.
If you prefer the CLI, create a prov.json file (an HA example json file in Appendix A) and run the create-dbsystem command.

```bash
# odacli create-dbsystem -p /tmp/prov.json
Job details
---------------------------------------------
ID: 1978eaa0-d45c-4efb-88ea-c9fba756a4f9
Description: DB System mydbsystem creation
Status: Created
Created: March 7, 2021 11:19:44 PM HST
Message:
```

7.) Confirm that the DB System was created successfully

```bash
# odacli describe-dbsystem -n mydbsystem
DB System details
---------------------------------------------
ID: 4f9999ad-12b2-4b7e-88a2-753334a4f775
Name: mydbsystem
Image: 19.10.0.0.0
Shape: odb2
Cluster name: dba6cd795483
Memory: 16.00 GB
Status: CONFIGURED
Created: 2021-03-15 04:56:17 HDT
Updated: 2021-03-15 06:01:32 HDT
CPU Pool
---------------------------------------------
Name: c6cd795483
Number of cores: 2
Host: node1
Effective CPU set: 0-1,32-33
```
Online CPUs: 0, 1, 32, 33
Offline CPUs: NONE

Host: node2
Effective CPU set: 0-1,32-33
Online CPUs: 0, 1, 32, 33
Offline CPUs: NONE

VM Storage
----------------------------------
Disk group: DATA
Volume name: S6CD795483
Volume device: /dev/asm/s6cd795483-433
Size: 400.00 GB
Mount Point: /u05/app/sharedrepo/mydbsystem
Acc Volume name: AS6CD795483
Acc Volume device: /dev/asm/as6cd795483-433
Acc Volume size: 1.60 GB

VMs
--------------------------
Host: node1
VM Name: x6cd795483
VM Host Name: node1n1.us.oracle.com
Target State: ONLINE
Current State: ONLINE

Host: node2
VM Name: y6cd795483
VM Host Name: node2n2.us.oracle.com
Target State: ONLINE
Current State: ONLINE

VNetworks
--------------------------
Host: node1
VM Name: x6cd795483
Public: xx.xx.xx.xx / 255.255.248.0 / ens3 / BRIDGE(pubnet)
ASM: 192.168.17.4 / 255.255.255.128 / ens4 / BRIDGE(privasm)
VLAN(icbond0.100)
Interconnect: 192.168.17.129 / 255.255.255.252 / ens5 / BRIDGE(privnet1)
VLAN(icbond0.101)

Host: node2
VM Name: y6cd795483
Public: xx.xx.xx.xx / 255.255.248.0 / ens3 / BRIDGE(pubnet)
ASM: 192.168.17.5 / 255.255.255.128 / ens4 / BRIDGE(privasm)
VLAN(icbond0.100)
Interconnect: 192.168.17.130 / 255.255.255.252 / ens5 / BRIDGE(privnet1)
VLAN(icbond0.101)

Databases
--------------------------
Name: my19db
Resource ID: ad129cf9-b8d8-46c6-a5ac-9b4f09880d03
Unique name: mydb19
Domain name: us.oracle.com
DB Home ID: a4ebb778-aec7-4c9b-b3ef-506cb0a7ef78
Shape: odb2
Version: 19.10.0.0.210119
Edition: EE
Type: RAC
Role: PRIMARY
Class: OLTP
Storage: ASM
Redundancy:
8.) Test connectivity to your DB System

During the creation of the DB system you defined scan-name and database name. With this information you should be able to build your TNS connect string using the default service name which is the same as the database name. It is a best practice to create your own database services based on your HA requirements. For an initial connection test, the default service is perfectly sufficient.

Connect as well via ssh to the KVM guest nodes and verify you can connect to the database instances locally.

```bash
# ssh root@node1n1.us.oracle.com
# su - oracle
$ export ORACLE_HOME=/u01/app/oracle/product/19.0.0.0/dbhome_1
$ export ORACLE_SID=my19db1     (the '1' after the DB name is for instance 1 on the first RAC node)
$ /u01/app/oracle/product/19.0.0.0/dbhome_1/bin/sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Mon Mar 15 08:32:37 2021
Version 19.10.0.0.0

Copyright (c) 1982, 2020, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.10.0.0.0
```
SOLUTION IN A BOX EXAMPLES FOR ODA

JDE
JDE team is only providing an old Xen VM Template and you should not use it with KVM. To install JDE in KVM on ODA follow the JDE documentation:
https://docs.oracle.com/cd/E84502_01/learnjde/installation.html#one-click-provisioning

EBS
At this time, we cannot install the EBS database in a DB system because database version 12.1 isn’t available within a DB system. A migration is possible and will be covered in a future technical brief.

FlexCube
FLEXCUBE team does not provide any Xen VM Template. To install FLEXCUBE in KVM on ODA follow the FLEXCUBE documentation:
https://docs.oracle.com/cd/F18207_01/install.htm

CONCEPT OF USING ODA CLI KVM COMMANDS
You can use odacli commands to get information about your configured resources like CPU Pools, VM Disks, VMs and so on. You use the odacli list’ command to get a list of configured resources (e.g. odacli list-cpupools, odacli list-vmstorages etc.) you use the odacli describe’ command to get more information about a specific resource (e.g. odacli describe-cpupools, odacli describe-vmstorages etc.). The same way it works for ‘start’, ‘stop’, ‘modify’ and ‘delete’. (mind that not all are commands are available for all resource types, e.g. vdisks aren’t stoppable). For a full list of available commands review the ODA documentation. Below examples of the most often used with KVM:

List DB System
# odacli list-dbsystems

Show the details about a DB System
# odacli describe-dbsystem -n dbsystem_name

Start a DB System
# odacli start-dbsystem -n dbsystem_name

Stop a DB System
# odacli stop-dbsystem -n dbsystem_name

Delete a DB System
# odacli delete-dbsystem -n dbsystem_name

NOTE: ensure GI and *all* databases within the DB system are properly deinstalled before running the command

List KVM guest
# odacli list-vms

Show the details about a KVM guest
# odacli describe-vm -n vm1

**Start a KVM guest**
# odacli start-vm -n vm1

**Stop a KVM guest**
# odacli stop-vm -n vm1

**Delete a KVM guest**
# odacli delete-vm -n vm1

---

### MIGRATING OVM GUESTS FROM ODA VP TO KVM ON ODA BM

Creating a Solution-in-a-Box doesn’t necessary mean that you have to create a new Virtual Machine. You might have an existing Oracle Database Appliance running Oracle Virtual Machine (OVM) and you would like to migrate the OVM from this machine to a KVM guest. The steps are document in My Oracle Support note: “Migrating OVM guests from ODA VP to KVM on ODA BM (Doc ID 2775840.1)”.

### BACKUP YOUR KVM GUEST

Backup and Recovery procedures for the DB Systems are not different from the well-known database backup procedures every DBA is familiar with. For the Application VMs please review My Oracle Support note “Backup of KVM guests on ODA 19.9 BM and later (Doc ID 2779329.1)”.

### APPENDIX A: EXAMPLE PROV.JSON

INFO: ensure to review every parameter in the file and adjust to your environment and needs. This example is for high availability DB System:

```json
{
  "system": {
    "name": "mydbsystem",
    "shape": "odb1",
    "systemPassword": "Hallo_Welt_12",
    "timeZone": "America/Los_Angeles",
    "diskGroup": "DATA",
    "enableRoleSeparation": true,
    "customRoleSeparation": {
      "groups": [
        {
          "name": "oinstall",
          "id": 1001,
          "role": "oinstall"
        },
        {
          "name": "dbaoper",
          "id": 1002,
          "role": "dbaoper"
        },
        {
          "name": "dba",
          "id": 1003,
          "role": "dba"
        },
        {
          "name": "asmadmin",
          "id": 1004,
```
"role": "asmadmin"
},
{
"name": "asmoper",
"id": 1005,
"role": "asmoper"
},
{
"name": "asmdba",
"id": 1006,
"role": "asmdba"
}
],
"users": [
{
"name": "grid",
"id": 1000,
"role": "gridUser"
},
{
"name": "oracle",
"id": 1001,
"role": "oracleUser"
}
]
},
"database": {
"name": "mydb19",
"uniqueName": "mydb19",
"domainName": "us.oracle.com",
"adminPassword": "Hallo_Welt_12",
"version": "19.10.0.0.210119",
"edition": "EE",
"type": "RAC",
"dbClass": "OLTP",
"shape": "odb1",
"role": "PRIMARY",
"targetNodeNumber": null,
"enableDbConsole": false,
"redundancy": null,
"characterSet": {
"characterSet": "AL32UTF8",
"nlsCharacterset": "AL16UTF16",
"dbTerritory": "AMERICA",
"dbLanguage": "ENGLISH"
},
"rmanBackupPassword": null,
"enableTDE": false,
"iscDb": false
},
"network": {
"domainName": "us.oracle.com",
"ntpServers": ["xx.xx.xx.xx", "xx.xx.xx.xx "],
"dnsServers": ["xx.xx.xx.xx", "xx.xx.xx.xx "],
"nodes": [
{
"name": "node1",
"ipAddress": "xx.xx.xx.xx",
"netmask": "xx.xx.xx.xx",
"gateway": "xx.xx.xx.xx",
"number": 0,
"vipName": "node1-vip",
"vipAddress": "xx.xx.xx.xx"
}
],
{
"name": "node2",
"ipAddress": "xx.xx.xx.xx",
"netmask": "xx.xx.xx.xx",
"gateway": "xx.xx.xx.xx",
"number": 0,
"vipName": "node2-vip",
"vipAddress": "xx.xx.xx.xx"}
APPENDIX B: EXAMPLE DATABASE SHAPES

Oracle Database Appliance X8-2-HA and X8-2M OLTP Database Shapes

<table>
<thead>
<tr>
<th>Shape</th>
<th>CPU Cores</th>
<th>SGA (GB)</th>
<th>PGA (GB)</th>
<th>Processes</th>
<th>Log File Size (GB)</th>
<th>Log Buffer (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>odb1s</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>200</td>
<td>4</td>
<td>32</td>
</tr>
<tr>
<td>odb1</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>200</td>
<td>4</td>
<td>32</td>
</tr>
<tr>
<td>odb2</td>
<td>2</td>
<td>8</td>
<td>4</td>
<td>400</td>
<td>4</td>
<td>32</td>
</tr>
<tr>
<td>odb4</td>
<td>4</td>
<td>16</td>
<td>8</td>
<td>800</td>
<td>4</td>
<td>64</td>
</tr>
<tr>
<td>odb6</td>
<td>6</td>
<td>24</td>
<td>12</td>
<td>1200</td>
<td>8</td>
<td>256</td>
</tr>
<tr>
<td>odb08</td>
<td>8</td>
<td>32</td>
<td>16</td>
<td>1600</td>
<td>8</td>
<td>256</td>
</tr>
<tr>
<td>odb10</td>
<td>10</td>
<td>40</td>
<td>20</td>
<td>2000</td>
<td>8</td>
<td>256</td>
</tr>
<tr>
<td>odb12</td>
<td>12</td>
<td>48</td>
<td>24</td>
<td>2400</td>
<td>16</td>
<td>256</td>
</tr>
<tr>
<td>odb16</td>
<td>16</td>
<td>64</td>
<td>32</td>
<td>3200</td>
<td>16</td>
<td>256</td>
</tr>
<tr>
<td>odb20</td>
<td>20</td>
<td>80</td>
<td>40</td>
<td>4000</td>
<td>16</td>
<td>256</td>
</tr>
<tr>
<td>odb24</td>
<td>24</td>
<td>96</td>
<td>48</td>
<td>4800</td>
<td>16</td>
<td>256</td>
</tr>
<tr>
<td>odb28</td>
<td>28</td>
<td>112</td>
<td>56</td>
<td>5600</td>
<td>16</td>
<td>256</td>
</tr>
<tr>
<td>odb32</td>
<td>32</td>
<td>128</td>
<td>64</td>
<td>6400</td>
<td>16</td>
<td>256</td>
</tr>
</tbody>
</table>