Deploying Oracle Hospitality Suite8 Property on Oracle Database Appliance Virtualized Platform

ORACLE WHITE PAPER | JUNE 2017

Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Table of Contents

Executive Overview

Oracle Hospitality Suite8 Property combines all hotel processes into a single software solution. From reservations to housekeeping, from restaurant to sales and marketing, this software suite supports all areas of your hotel while focusing on your most valuable asset: your guest.

With Oracle Database Appliance, you have the choice to deploy highly available Oracle Databases directly on the physical hardware or to deploy both Oracle Database and Application Tier servers in a virtual environment within the appliance. This paper describes how to deploy the Oracle Hospitality Suite8 solution on Oracle Database Appliance Virtualized Platform. The solution deployment consists of an Application VM and an Oracle database running in an Oracle VM (ODA_BASE). This provides a simple, reliable and (if required) high available solution-in-a-box.

Scope

This paper describes how to install Oracle Hospitality Suite8 solution database and application tier on the Oracle Database Appliance Virtualized Platform. You need to be familiar with the Oracle Database Appliance documentation and, in particular, how it relates to the Oracle Database Appliance Virtualized Platform. Further information about the Oracle Database Appliance is available at:

http://docs.oracle.com/en/engineered-systems/#oracle-database-applianceappliance

The documentation for Oracle Hospitality Suite8 is available at:

http://docs.oracle.com/cd/E74579_01/index.html

Introduction

The Benefits of Deploying Oracle Hospitality Suite8 Property solution on Oracle Database Appliance Virtualized Platform:

- **Solution in-a-box** – provides an integral Oracle Engineered System that saves you time and money by simplifying deployment, maintenance, and support of high availability database solutions.
- **Pay as you Grow** – The Capacity-On-Demand software licensing model allows to quickly and incrementally scale processors without any hardware upgrades.
- **Simple to implement** – Plug in the appliance and then use the integral Oracle Appliance Manager to provision the firmware, operating system, storage manager and virtualization software. The Appliance Manager also drastically simplifies maintenance by patching the entire appliance, including all firmware and software, in one operation, using an Oracle-tested patch bundle engineered specifically for the appliance.
- **Simple to manage** – Oracle Database Appliance has been engineered with specific focus on simplicity and reliability. This reduces complexity, risk and costs in deploying a highly available solution.
- **High Availability Hardware** – The configuration provides complete redundancy with no single points of failure. If hardware fails or is about to fail, the Automated Service Request capability (phone home) generates automatic requisitions for replacement components such as disks, power supplies, fans etc.

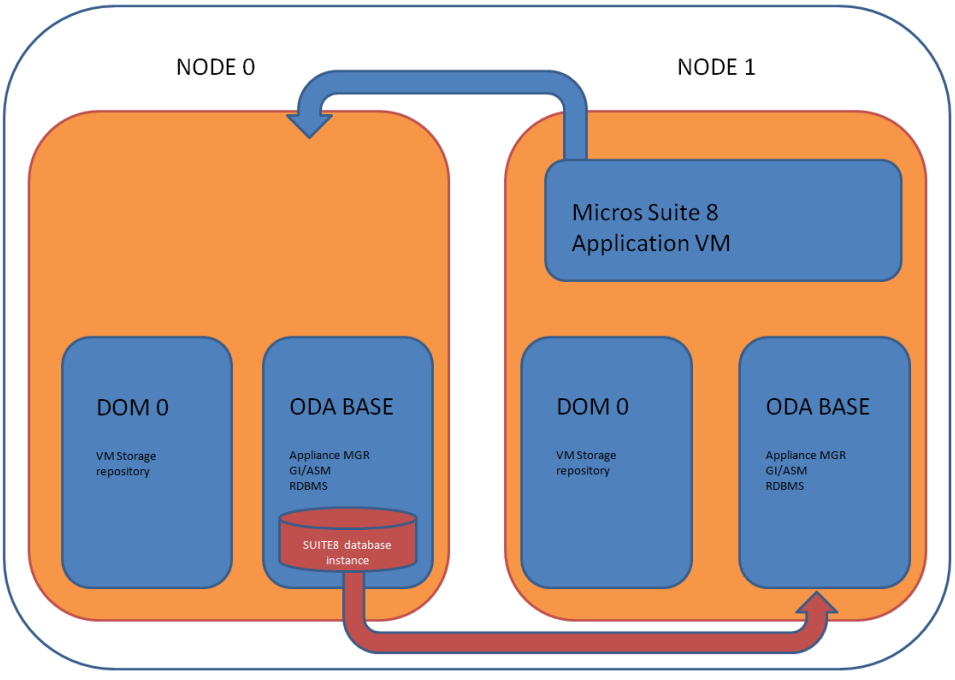
- **Performance** – Oracle Database Appliance is preconfigured and pre-tuned for optimal performance, which reduces maintenance and operating costs. The Oracle Database runs in a special privileged user domain, which eliminates the virtualization overhead.
- **Support** – For the Oracle Database Appliance there is a single point of contact for support. All hardware and software components are supported by Oracle, and there is no need to contact multiple vendors if a problem occurs. When a problem occurs, the Appliance Manager performs the job of the DBA and automatically collates all the logs and system history thereby enabling issues to be logged, analyzed and processed more quickly.
- **Oracle Real Application Clusters (Oracle RAC and RAC One Node)** – is a clustered version of Oracle Database based on a comprehensive high-availability stack, ensuring high availability, scalability, and agility for any application.

Oracle Database Appliance Virtualized Platform Deployment Architecture

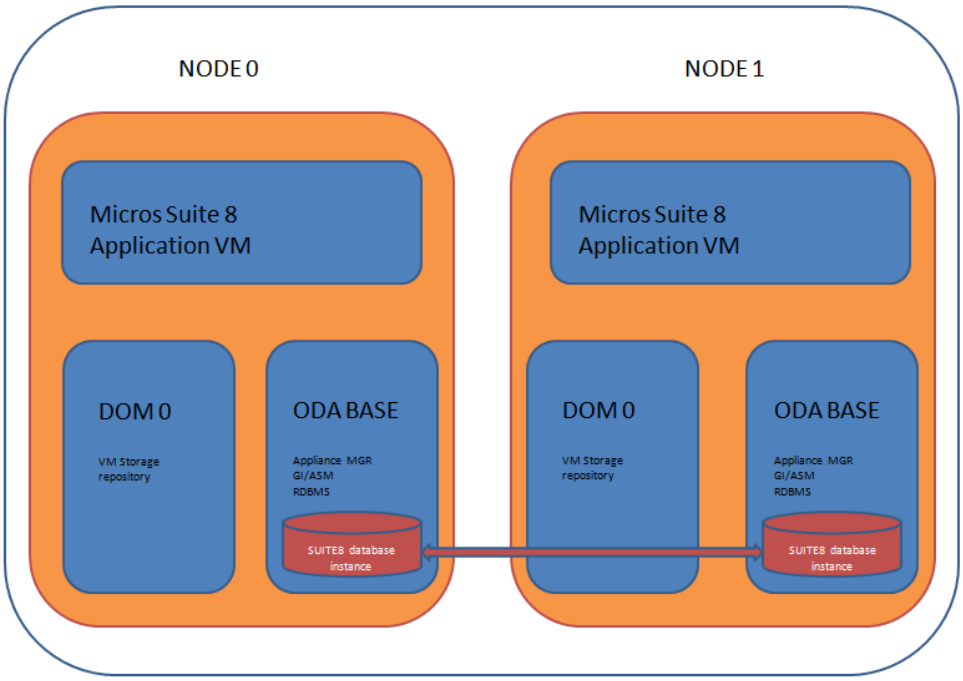
This section gives a brief overview of the architecture to better understand the installation procedure in this paper.

When Oracle Database Appliance Virtualized Platform is deployed, an ODA_BASE privileged user domain is created. The Appliance Manager provisions and manages the Oracle Database within the ODA_BASE domain. This domain is optimized to host database instances, and all the shared disks are directly attached to this domain eliminating any virtualization overhead thus providing near native disk performance.

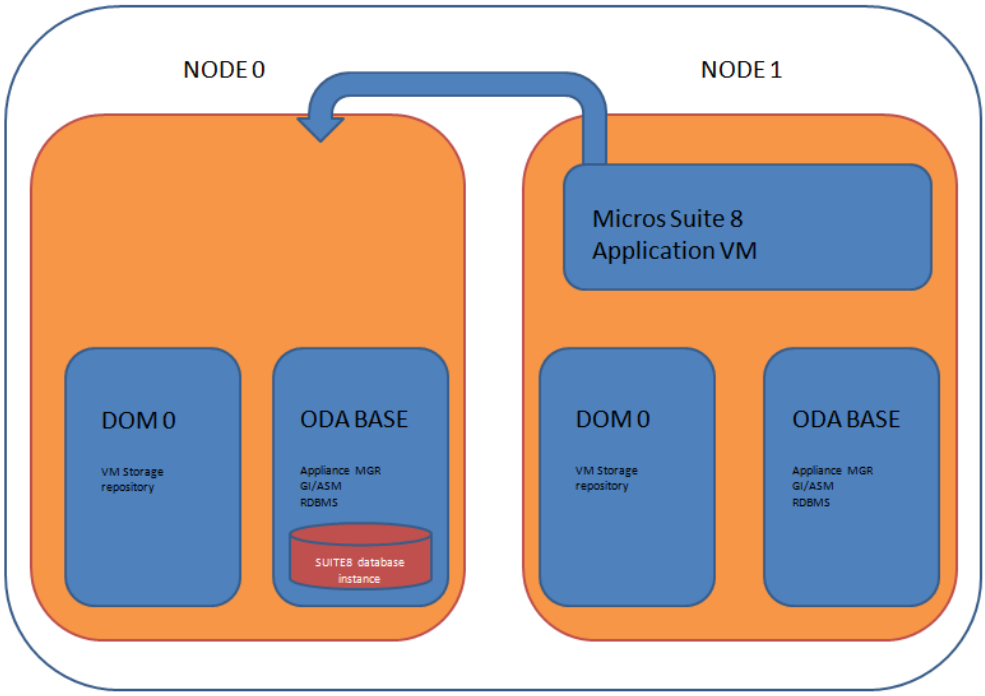
The Oracle Hospitality Suite8 Property database is deployed in ODA_BASE as this is considered a best practice and significantly improves performance and manageability. The database can be setup as a single instance, a RAC One Node or a RAC database depending on your high availability requirements and your database licenses. The database files will in any case be stored on shared storage, available to both nodes. The picture below shows a setup with RAC One Node where one database instance is running on one node only. In case of a system fault or maintenance the instance is transferred to the other node. The Windows VM guest is configured to failover in case of a fault as well



With a RAC database configuration, two database instances execute transactions at the same time. Oracle RAC provides the ability to continue processing transactions even if one of the nodes fails. In this example every node hosts a Windows guest for the Application.



The database can as well run as a single instance, if the node that hosts a single instance goes down, manual intervention is required to start the database from the second node. This might be needed if the failed node can't be brought online e.g. due to HW failure. The Oracle Hospitality Suite8 Property application tier will run in a VM guest configured for failover.



The hardware and software versions are dependent on the version of the Oracle Database Appliance model and the version of the Appliance Manager used.

Preparing the Oracle Database Appliance

In order to run a 'Solution in a box', the Oracle Database Appliance (ODA) has to be setup as 'Virtual Platform'. In case your ODA is already imaged and/or deployed with the Bare Metal image, you will have to re-image both nodes and re-deploy the software afterwards. The deployment of ODA is not in the scope of this document; please refer to the setup poster and the Oracle documentation for the necessary steps. The setup poster for your ODA model can be found at http://docs.oracle.com/en/engineered-systems/#oracle-database-applianceappliance. To verify that your ODA is setup as virtualized platform, execute the oakcli show env_hw command:

Output from DOM0:

oakcli show env_hw

VM-DOM0 ODA X5-2

Public interface : COPPER

Output from ODA BASE:

oakcli show env_hw

VM-ODA_BASE ODA X5-2

The 'VM' in the output would tell you, that the required virtual platform is in place.

The way you deploy the ODA in regards to ASM redundancy and database backup location is up to your high availability and space requirements. When the ODA BASE is configured, make sure not to assign all cores to it. Only cores not assigned to ODA BASE, can be assigned to virtual guests.

Note: The number of cores you enable, depends on the number of licenses you have purchased. In the example below, we will use 4 cores. For best performance we recommend to enable 8 cores.

Create the Oracle Hospitality Suite8 Property Database

During the deployment of ODA BASE, you have already the option to create a database. For this document we skip this step and create the database separately. In case you would have to recreate the database, this provides you with the steps needed. To create a database on ODA, only the oakcli interface should be used. For the example in this document, we create a RAC ONE Node database with the name V81. We use an existing Oracle Home. If you omit the "–oh", the database will be created in a new Oracle Home. We create the database to use 4 cores and 32GB RAM. If higher HA is required, choose RAC, if less 'EE' option which will create a single instance database on one of the nodes.

```
# oakcli create database -db V81 -oh /u01/app/oracle/product/12.1.0.2/dbhome_1

.

.

Please select one of the following for Database type  [1 .. 3] :

1    => OLTP

2    => DSS

3    => In-Memory

1

The selected value is : OLTP

Please select one of the following for Database Deployment  [1 .. 3] :

1    => EE : Enterprise Edition

2    => RACONE

3    => RAC

2

The selected value is : RACONE

Please select one of the following for Database Class  [1 .. 5] :

1    => odb-01s  (   1 cores ,     4 GB memory)

2    =>  odb-01  (   1 cores ,     8 GB memory)

3    =>  odb-02  (   2 cores ,    16 GB memory)

4    =>  odb-04  (   4 cores ,    32 GB memory)

5    =>  odb-06  (   6 cores ,    48 GB memory)

4

The selected value is : odb-04  (   4 cores ,    32 GB memory).
```

```
.
.
SUCCESS: 2016-08-30 07:47:39: Successfully created the Database : V81
```

To confirm the creation via okacli run the 'show databases' command:

```
oakcli show databases -db V81
Name     Type       Storage   HomeName          HomeLocation                          Version
-----    ------     --------  --------------    ----------------                      ----------
V81      RACOneNode ACFS      OraDb12102_home1  /u01/app/oracle/product/12.1.0.2/dbhome_1
12.1.0.2.160419(22291127,22502555)
```

After the database is created, we need to apply some customizations like user creation, granting permissions, setting up tablespaces and so on. In this example we login to sqlplus on the node where the instance is running (V81_1 on node0 or V81_2 on node1) to perform the necessary steps. As a first step we disable archiving in order to avoid running out of space during configuration:

```
login as: oracle
$ . oraenv
ORACLE_SID = [V81] ? V81_1
ORACLE_HOME = [/home/oracle] ? /u01/app/oracle/product/12.1.0.2/dbhome_1
The Oracle base remains unchanged with value /u01/app/oracle
$ sqlplus / as sysdba
SQL> shutdown immediate
SQL> startup mount
SQL> alter database noarchivelog;
SQL> alter database open;
```

To confirm, run an 'archive log list':

```
SQL> archive log list
Database log mode           No Archive Mode
Automatic archival          Disabled
Archive destination         USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence  1
Current log sequence        2
SQL>
```

Next we will create additional tablespaces

```
CREATE TABLESPACE V8CONF DATAFILE SIZE 2G AUTOEXTEND ON MAXSIZE 31G;

CREATE TABLESPACE V8INDEX DATAFILE SIZE 2G AUTOEXTEND ON MAXSIZE 31G;

CREATE TABLESPACE V8MAIN DATAFILE SIZE 2G AUTOEXTEND ON MAXSIZE 31G;

CREATE TABLESPACE V8ARCHIVE DATAFILE SIZE 2G AUTOEXTEND ON MAXSIZE 31G;

CREATE TABLESPACE V8LARGE DATAFILE SIZE 2G AUTOEXTEND ON MAXSIZE 31G;

CREATE TEMPORARY TABLESPACE V8TEMP TEMPFILE SIZE 2G AUTOEXTEND ON MAXSIZE 31G;
```

Create users and grant rights:

```
CREATE USER V8LIVE identified by micros
DEFAULT TABLESPACE V8MAIN
TEMPORARY TABLESPACE V8TEMP
/
ALTER USER V8LIVE
  QUOTA UNLIMITED ON v8archive
  QUOTA UNLIMITED ON v8conf
  QUOTA UNLIMITED ON v8index
  QUOTA UNLIMITED ON v8large
  QUOTA UNLIMITED ON v8main
/
CREATE USER V8TRAIN identified by micros
DEFAULT TABLESPACE V8MAIN
TEMPORARY TABLESPACE V8TEMP
/
ALTER USER V8TRAIN
  QUOTA UNLIMITED ON v8archive
  QUOTA UNLIMITED ON v8conf
  QUOTA UNLIMITED ON v8index
```

```
  QUOTA UNLIMITED ON v8large
  QUOTA UNLIMITED ON v8main
/
GRANT create table, create type, create procedure, create view, create trigger, create sequence, create
materialized view TO V8LIVE;
GRANT create session TO V8LIVE;
GRANT create database link TO V8LIVE;
GRANT create public database link TO V8LIVE;
CREATE OR REPLACE VIEW sys.x_$ksppi  AS SELECT * FROM sys.x$ksppi;
CREATE OR REPLACE VIEW sys.x_$ksppsv AS SELECT * FROM sys.x$ksppsv;
GRANT SELECT ON v_$session TO V8LIVE WITH GRANT OPTION;
GRANT SELECT ON gv_$session TO V8LIVE WITH GRANT OPTION;
GRANT SELECT ON v_$database TO V8LIVE WITH GRANT OPTION;
GRANT SELECT ON dba_indexes TO V8LIVE WITH GRANT OPTION;
GRANT SELECT ON dba_tables TO V8LIVE WITH GRANT OPTION;
GRANT SELECT ON v_$sess_io TO V8LIVE WITH GRANT OPTION;
GRANT SELECT ON v_$statname TO V8LIVE WITH GRANT OPTION;
GRANT SELECT ON v_$sql TO V8LIVE WITH GRANT OPTION;
GRANT SELECT ON v_$sysstat TO V8LIVE WITH GRANT OPTION;
GRANT SELECT ON v_$parameter TO V8LIVE WITH GRANT OPTION;
GRANT SELECT ON v_$instance TO V8LIVE WITH GRANT OPTION;
GRANT SELECT ON dba_users TO V8LIVE WITH GRANT OPTION;
GRANT SELECT ON dba_tab_columns TO V8LIVE WITH GRANT OPTION;
GRANT SELECT ON dba_ind_columns TO V8LIVE WITH GRANT OPTION;
GRANT SELECT ON dba_ind_expressions TO V8LIVE WITH GRANT OPTION;
GRANT SELECT ON dba_free_space TO V8LIVE WITH GRANT OPTION;
GRANT SELECT ON dba_data_files TO V8LIVE WITH GRANT OPTION;
GRANT SELECT ON v_$log TO V8LIVE WITH GRANT OPTION;
GRANT SELECT ON v_$tablespace TO V8LIVE WITH GRANT OPTION;
GRANT SELECT ON sys.x_$ksppi TO V8LIVE WITH GRANT OPTION;
GRANT SELECT ON sys.x_$ksppsv TO V8LIVE WITH GRANT OPTION;
GRANT SELECT ON dba_rsrc_plan_directives TO V8LIVE WITH GRANT OPTION;
GRANT EXECUTE ON dbms_resource_manager TO V8LIVE WITH GRANT OPTION;
GRANT EXECUTE ON dbms_snapshot_utl TO V8LIVE WITH GRANT OPTION;
```

```
GRANT EXECUTE ON dbms_crypto TO V8LIVE WITH GRANT OPTION;

GRANT EXECUTE ON dbms_session TO V8LIVE WITH GRANT OPTION;

GRANT CREATE ANY TABLE TO V8LIVE;

GRANT SELECT ON dba_temp_files TO V8LIVE WITH GRANT OPTION;

GRANT SELECT ON dba_tablespaces TO V8LIVE WITH GRANT OPTION;

GRANT SELECT ON gv_$sort_segment TO V8LIVE WITH GRANT OPTION;

GRANT create table, create type, create procedure, create view, create trigger, create sequence, create
materialized view TO V8TRAIN;

GRANT create session TO V8TRAIN;

GRANT create database link TO V8TRAIN;

GRANT create public database link TO V8TRAIN;

GRANT SELECT ON v_$session TO V8TRAIN WITH GRANT OPTION;

GRANT SELECT ON gv_$session TO V8TRAIN WITH GRANT OPTION;

GRANT SELECT ON v_$database TO V8TRAIN WITH GRANT OPTION;

GRANT SELECT ON dba_indexes TO V8TRAIN WITH GRANT OPTION;

GRANT SELECT ON dba_tables TO V8TRAIN WITH GRANT OPTION;

GRANT SELECT ON v_$sess_io TO V8TRAIN WITH GRANT OPTION;

GRANT SELECT ON v_$statname TO V8TRAIN WITH GRANT OPTION;

GRANT SELECT ON v_$sql TO V8TRAIN WITH GRANT OPTION;

GRANT SELECT ON v_$sysstat TO V8TRAIN WITH GRANT OPTION;

GRANT SELECT ON v_$parameter TO V8TRAIN WITH GRANT OPTION;

GRANT SELECT ON v_$instance TO V8TRAIN WITH GRANT OPTION;

GRANT SELECT ON dba_users TO V8TRAIN WITH GRANT OPTION;

GRANT SELECT ON dba_tab_columns TO V8TRAIN WITH GRANT OPTION;

GRANT SELECT ON dba_ind_columns TO V8TRAIN WITH GRANT OPTION;

GRANT SELECT ON dba_ind_expressions TO V8TRAIN WITH GRANT OPTION;

GRANT SELECT ON dba_free_space TO V8TRAIN WITH GRANT OPTION;

GRANT SELECT ON dba_data_files TO V8TRAIN WITH GRANT OPTION;

GRANT SELECT ON v_$log TO V8TRAIN WITH GRANT OPTION;

GRANT SELECT ON v_$tablespace TO V8TRAIN WITH GRANT OPTION;

GRANT SELECT ON sys.x_$ksppi TO V8TRAIN WITH GRANT OPTION;

GRANT SELECT ON sys.x_$ksppsv TO V8TRAIN WITH GRANT OPTION;

GRANT SELECT ON dba_rsrc_plan_directives TO V8TRAIN WITH GRANT OPTION;

GRANT EXECUTE ON dbms_resource_manager TO V8TRAIN WITH GRANT OPTION;
```

```
GRANT EXECUTE ON dbms_snapshot_utl TO V8TRAIN WITH GRANT OPTION;
GRANT EXECUTE ON dbms_crypto TO V8TRAIN WITH GRANT OPTION;
GRANT EXECUTE ON dbms_session TO V8TRAIN WITH GRANT OPTION;
GRANT CREATE ANY TABLE TO V8TRAIN;
GRANT SELECT ON dba_temp_files TO V8TRAIN WITH GRANT OPTION;
GRANT SELECT ON dba_tablespaces TO V8TRAIN WITH GRANT OPTION;
GRANT SELECT ON gv_$sort_segment TO V8TRAIN WITH GRANT OPTION;
```

Perform several more changes:

```
alter system set optimizer_adaptive_features=FALSE scope=both;
```

In the next steps we configure the wallet. First we create the directory to store it. For RAC or RAC ONE Node use a shared location like /cloudfs. As Oracle user execute:

```
mkdir /cloudfs/admin/V81/wallet
```

Now edit the sqlplnet.ora file (in the database home of V81)

```
vi /u01/app/oracle/product/12.1.0.2/dbhome_1/network/admin/sqlnet.ora
```

add the following entry (mind to use the wallet location from the step above):

```
ENCRYPTION_WALLET_LOCATION = (SOURCE = (METHOD = FILE) (METHOD_DATA = (DIRECTORY = /cloudfs/admin/V81/wallet)))
```

Restart the database and execute the commands below:

```
shutdown immediate
startup
ALTER SYSTEM SET ENCRYPTION KEY IDENTIFIED by "OracleSuite8";
ALTER SYSTEM SET ENCRYPTION WALLET OPEN IDENTIFIED BY "OracleSuite8";
```

NOTE: an error might be returned if the wallet is already open. This can be ignored.

Create The Virtual Guest

Below we outlined the steps to create the virtual Windows Server guest. For more information to create a virtual guest along with an automation, please review the My Oracle Support (MOS) notes:

NOTE:1524138.1 - ODAVP: How To Create a Fully-Virtualized Guests (HVM) from an OS ISO image

NOTE:2099289.1 - ODAVP: Create HVM Guest from ISO in "1-Click"

Mind that some steps have to be performed on DOM0 others on ODA_BASE.

1. Before we create the guest, we have to deploy a shared repository that will hold the VM guest. This has to be done on ODA_BASE

```
oakcli create repo micros -dg DATA -size 500
```

2. Connect to DOM0 on node0 and create a directory we use to create the template

```
mkdir -p /OVS/staging/vm_temp/win
```

3. Create the file that will be used for the image (in example 100GB)

```
dd if=/dev/zero of=/OVS/staging/vm_temp/win/win.img oflag=direct bs=1M count=102400
```

4. Create the configuration file that defines the vm guest

```
vi /OVS/staging/vm_temp/win/vm.cfg
```

5. Insert the information below into the vm.cfg file

```
name = 'win'
kernel = '/usr/lib/xen/boot/hvmloader'
device_model = '/usr/lib/xen/bin/qemu-dm'
builder = 'hvm'
memory = '4096'
vcpus = 2
acpi = 1
apic = 0
pae = 1
disk = ['file:/OVS/staging/vm_temp/win/win.img,xvda,w']
on_reboot = 'destroy'
on_crash = 'destroy'
on_poweroff = 'destroy'
keymap = 'en-us'
vif = ['type=ioemu,bridge=net1']
vnc = 1
vncconsole = 1
vnclisten ='0.0.0.0'
vncpasswd = 'welcome1'
vncunused = 1
usbdevice = 'tablet'
```

6. Create the template

```
cd /OVS/staging/vm_temp/win
tar -Sczvf win.tgz win.img vm.cfg
```

7. Connect to ODA_BASE on node 0 and import the template created in step 6.

```
oakcli import vmtemplate win -files /OVS/staging/vm_temp/win/win.tgz -repo micros -node 0
```

8. Perform some changes to the template

```
oakcli configure vmtemplate win -domain "XEN_HVM"

oakcli configure vmtemplate win -os "OTHER_WIN"
```

9. Clone the template (this step creates a runtime image from the VM template)

```
oakcli clone vm win -vmtemplate win -repo micros -node 0
```

10. Create a CPU pool for the VM guest and configure the VM to use it

```
oakcli create cpupool WINPool1 -numcpu 2 -node 0

oakcli create cpupool WINPool1 -numcpu 2 -node 1

oakcli configure vm win -cpupool WINPool1
```

11. If you want the VM guest to be able to failover to the other node, configure this with now

```
oakcli configure vm win -failover true
```

12. Copy the “Windows 2012 Server" OS ISO image to DOM0 of the node 0. The xx.xx.xx.xx would be the Ip address of DOM0 node0, win2012_64.iso the name of your ISO image

```
scp win2012_64.iso root@xx.xx.xx.xx:/OVS/staging/
```

13. Connect to DOM0 node0 and edit the vm.cfg

```
vi /OVS/Repositories/micros/.ACFS/snaps/win/VirtualMachines/win/vm.cfg
```

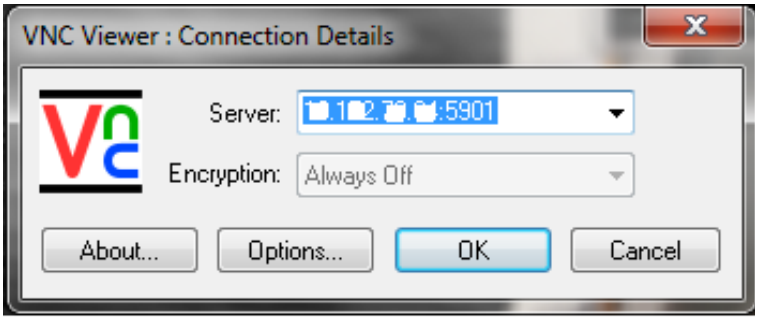
14. Add a new line to boot CD-ROM (d) before hard disk (c). Adjust the entry for ‘disk’ to include the ISO file

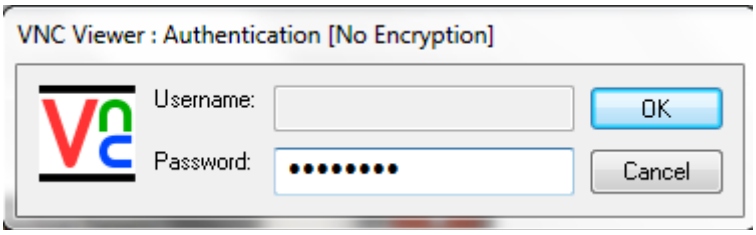
```
boot = "dc"

disk = ['file:/OVS/staging/vm_temp/win/win.img,xvda,w','file:/OVS/staging/win2012_64.iso,xvdc:cdrom,r']
```

15. Start the VM guest from ODA_BASE

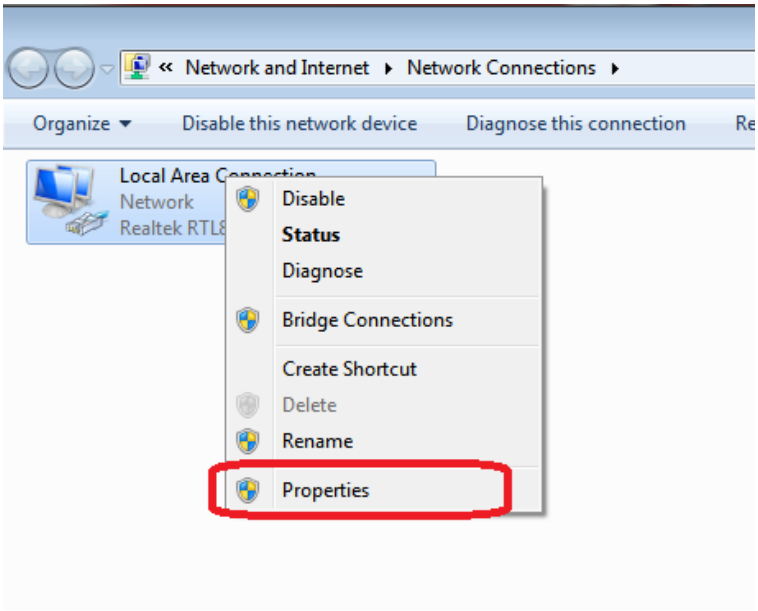
```
oakcli start vm win
```

16. VNC into the new guest using a VNC client. Use the IP address of DOM0 and the port 5901. When VNC asks for the password, provide “welcome1” (which we have set in the vm.cfg)

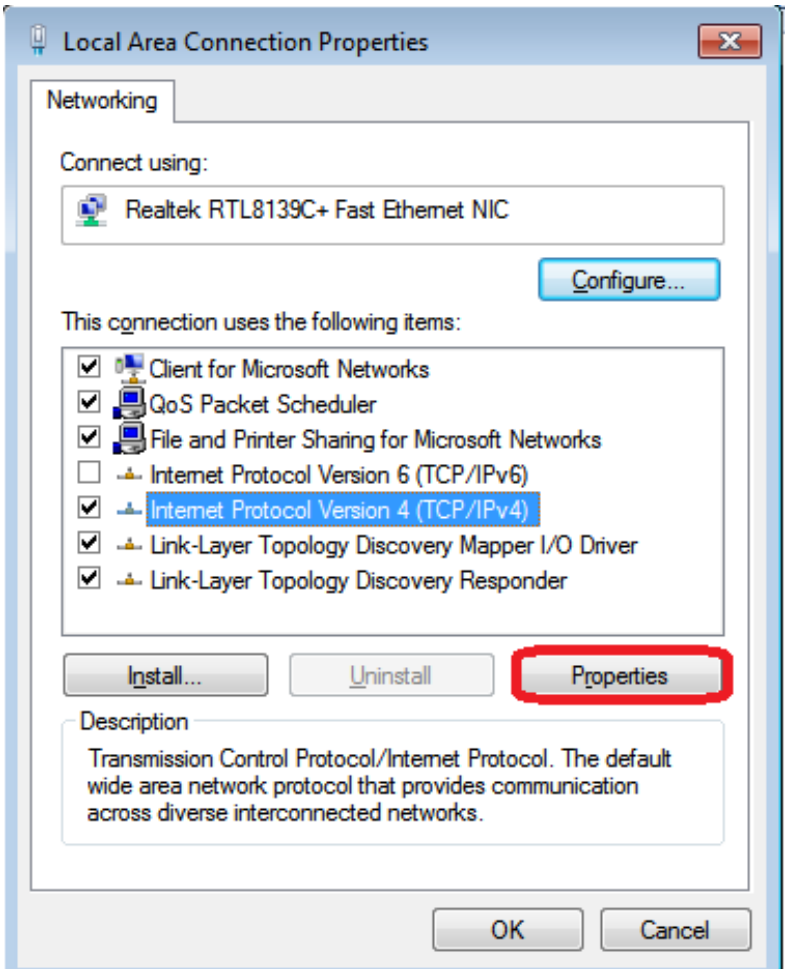




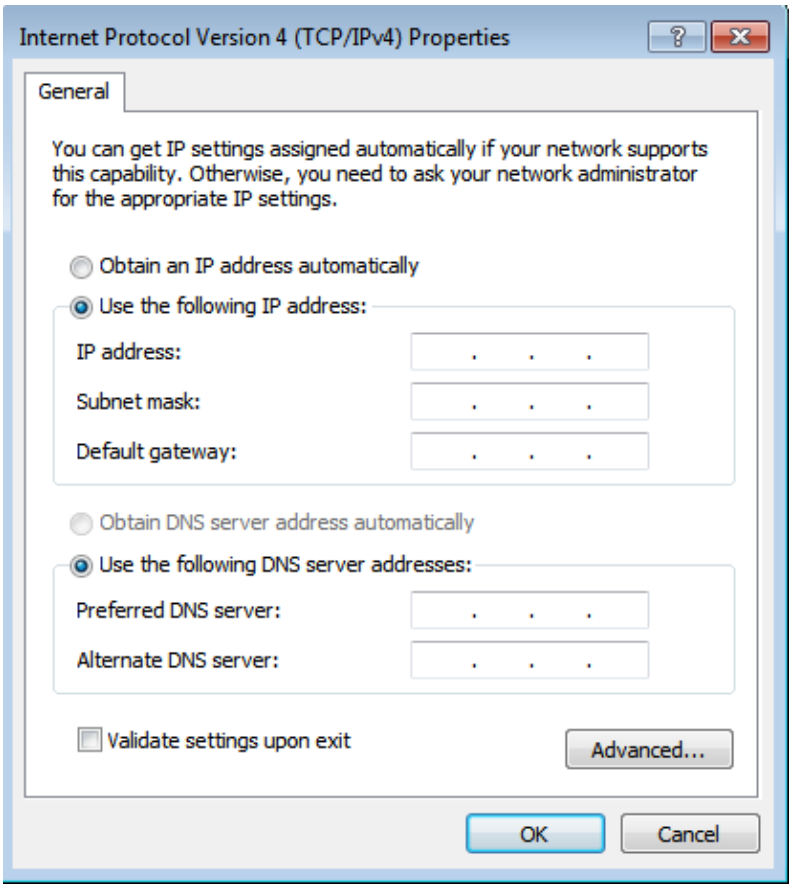
17. Complete the Windows OS installation via VNC. When ask for ‘computer name ensure to put in the desired name. After installation is completed we configure the network. Open a console (cmd) and type ncpa.cpl. In the window that opens right click on the NIC and select ‘Properties’.



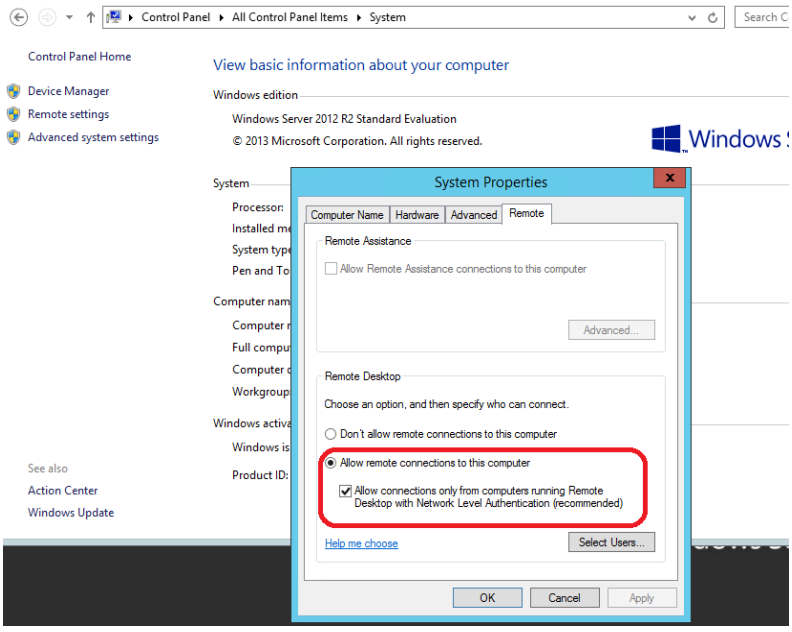
18. Click the Internet Protocol Version 4 and select ‘Properties’



19. Provide the network information for the VM

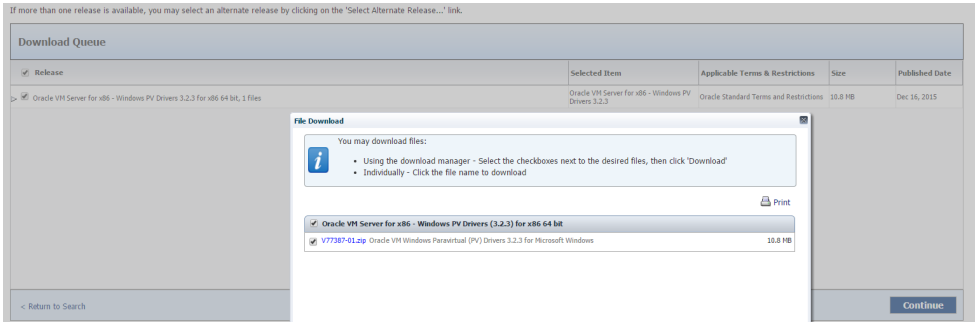


20. Enable access via remote desktop open ‘Control Panel’, ‘system’ and click ‘Remote settings’. Make the changes to allow remote connections

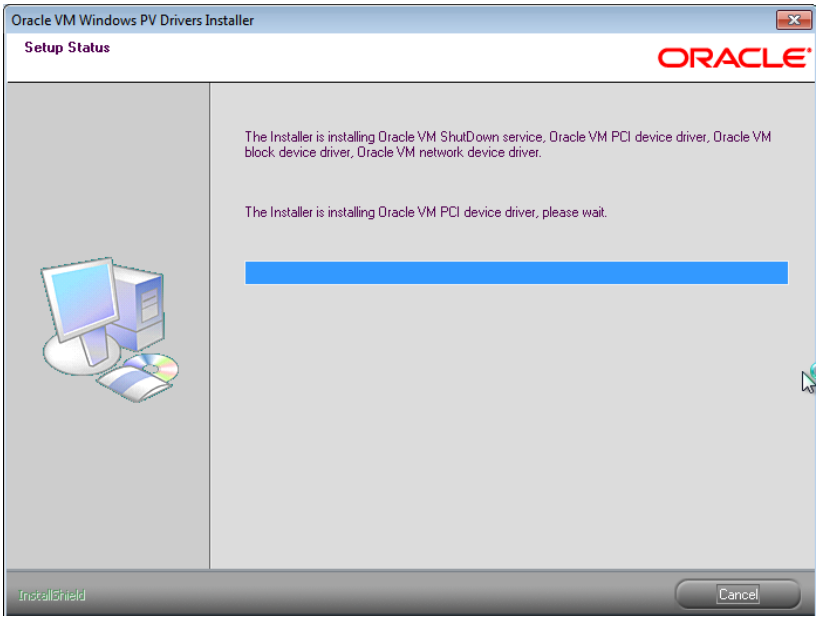


Now you can change to the more convenient ‘Remote Desktop Connection’ from a Windows machine (mstsc.exe). Type in the IP address of the Windows guest and click ‘Connect’.

21. Next we will install VM PV Drivers that Oracle provides for best performance of Windows VMs. Open the web browser and go to https://edelivery.oracle.com/linux and log in. Make sure the tick box ‘Linux/OVM/VMs’ is ticked and write into the search field: ‘Windows PV’. This will bring up a list of available Windows PV drivers. Select the latest version (because of BUG 22243923 we recommend minimum version 3.4.2) and click ‘Continue’. Confirm the questions and License agreement than download the file



22. After download is completed, unzip and install



When completed, select to restart the system later and shutdown Windows

23. On ODA_BASE stop the vm guest

```
oakcli stop vm win
```

24. Connect to DOM0 and edit the vm.cfg

```
vi /OVS/Repositories/micros/.ACFS/snaps/win/VirtualMachines/win/vm.cfg
```

Change the entries for boot, vif, disk , on_poweroff,on_crash and on_reboot

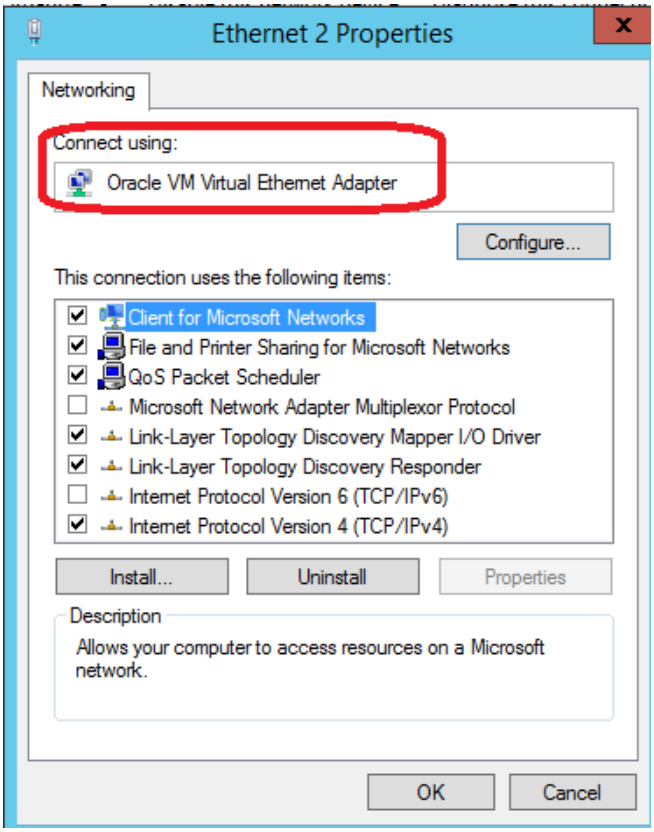
```
#boot="dc"

vif = ['type=netfront,bridge=net1']
```

```
on_poweroff = 'restart'

on_crash = 'restart'

on_reboot = 'restart'

disk = [u'file:/OVS/Repositories/micros/.ACFS/snaps/win/VirtualMachines/win/win.img,xvda,w']
```

25. On ODA_BASE start the vm guest

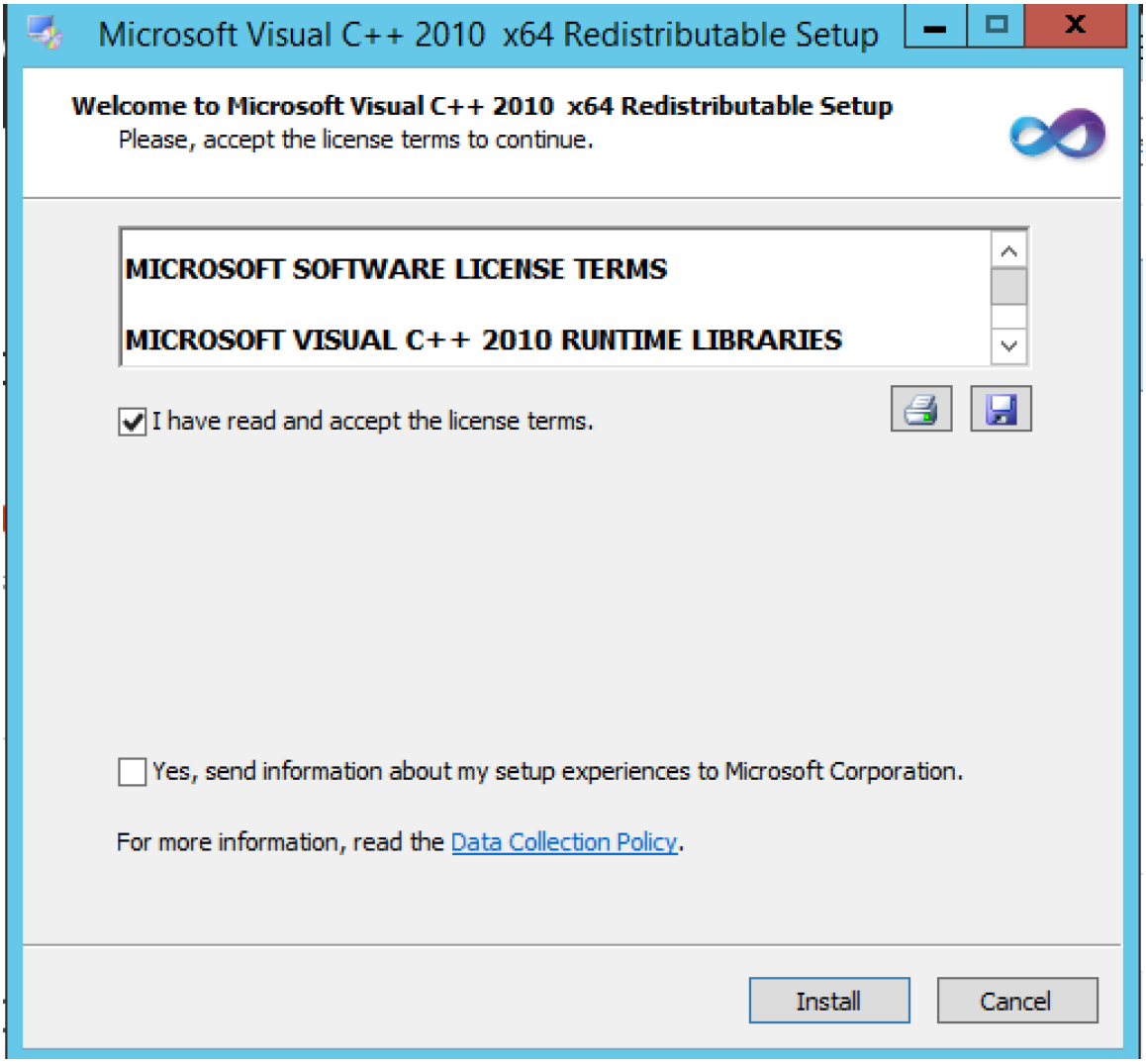
```
oakcli configure vm win –network "['type=netfront,bridge=net1']"

oakcli start vm win
```

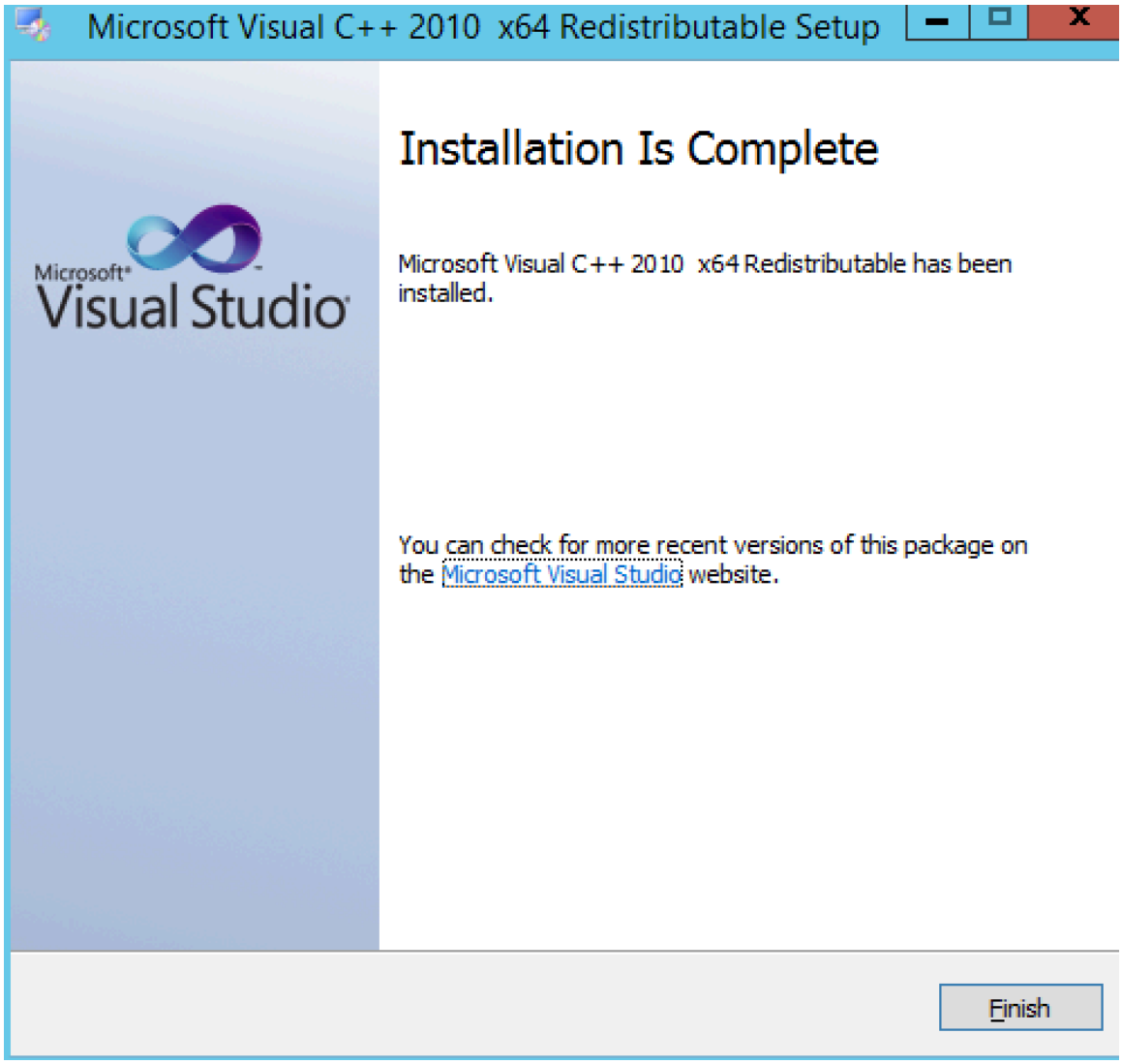
26. Connect to the vm guest and check that the correct driver is used



Install Oracle Hospitality Suite8 Property

In the virtual guest, install Microsoft Visual C++ Redistributable . To do so, go to https://www.microsoft.com/en-us/download/details.aspx?id=14632 and download the software. When completed, double click the file to start the installation.





This completes the setup of the ODA with database and virtual Windows guest. You should now follow the steps from white paper: 'Oracle Hospitality Suite8 Install shield' to complete the setup. The paper explains in detail the steps and options for setting up Suite8. With the Oracle Hospitality Suite8 Install shield the following will be installed:

- 32 Bit Oracle Client
- Oracle Hospitality Suite8

NOTE: Only users familiar with Oracle Hospitality Suite8 should perform such installation.

Setup RMAN backup into the Oracle Cloud

NOTE: Make sure to test the backup, automation and restore intensively before you go to production with your system.

1. Download the Oracle Cloud Backup Module (OCBM) from OTN:

http://www.oracle.com/technetwork/database/availability/oracle-cloud-backup-2162729.html

2. Copy the 'opc_install.zip' file to the ODA server

3. As oracle user, unzip the 'opc_install.zip' file on the ODA server

```
unzip opc_installer.zip
```

4. set your environment

```
$ . oraenv
ORACLE_SID = [V81] ? V81_1
ORACLE_HOME = [/home/oracle] ? /u01/app/oracle/product/12.1.0.2/dbhome_1
```

5. enable archive log mode for the V81 database

```
$ sqlplus / as sysdba
SQL> shutdown immediate
SQL> startup mount
SQL> alter database archivelog;
SQL> alter database open;
```

6. Install Oracle Cloud Backup Module using the following command

```
java -jar /tmp/opc_install.jar -serviceName Storage -identityDomain usoracle07396 -opcId <userid> -opcPass
<password> -l -walletDir $ORACLE_HOME/dbs/opc_wallet -libDir $ORACLE_HOME/lib
```

7. Configure RMAN

```
rman target /
RMAN> configure channel device type sbt
2> parms='SBT_LIBRARY=libopc.so
ENV=(OPC_FILE=/u01/app/oracle/product/12.1.0.2/dbhome_1/dbs/opcproddb1.ora)';
```

8. Enable Encryption For RMAN Backups (required)

```
RMAN> CONFIGURE ENCRYPTION FOR DATABASE
```

9. Enable Compression

```
RMAN> CONFIGURE COMPRESSION ALGORITHM 'MEDIUM';
```

10. Once these settings are in place, you can perform the RMAN backup as desired.

```
RMAN> SET ENCRYPTION ON IDENTIFIED BY 'mypassw0rd' ONLY;

RMAN> BACKUP DEVICE TYPE SBT AS COMPRESSED BACKUPSET DATABASE PLUS ARCHIVELOG
FORMAT '%U';
```

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

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

ORACLE

CONNECT WITH US

blogs.oracle.com/oracle
facebook.com/oracle
twitter.com/oracle
oracle.com

Hardware and Software, Engineered to Work Together

DEPLOYING ORACLE HOSPITALITY SUITE8 PROPERTY ON ORACLE DATABASE APPLIANCE VIRTUALIZED PLATFORM
June 2017
Author: Hagen Herbst , Oracle RAC Pack
Contributing Authors: Sebastian Peschel, Functional Architect Oracle

Oracle is committed to developing practices and products that help protect the environment