ORACLE
DATABASE APPLIANCE

Oracle Database Appliance: Virtual Local Area network

ORACLE WHITE PAPER | MARCH 2017

ORACLE

Table of Contents

Introduction

A single layer-2 network (in the seven-layer OSI model of computer networking, the data link layer) may be partitioned to create multiple distinct broadcast domains, which are mutually isolated so that packets can only pass between them via one or more routers; such a domain is referred to as a virtual local area network, virtual LAN or VLAN.

This is usually achieved on switch or router devices. Simpler devices only support partitioning on a port level (if at all), so sharing VLANs across devices requires running dedicated cabling for each VLAN. More sophisticated devices can mark packets through tagging, so that a single interconnect (trunk) may be used to transport data for various VLANs.

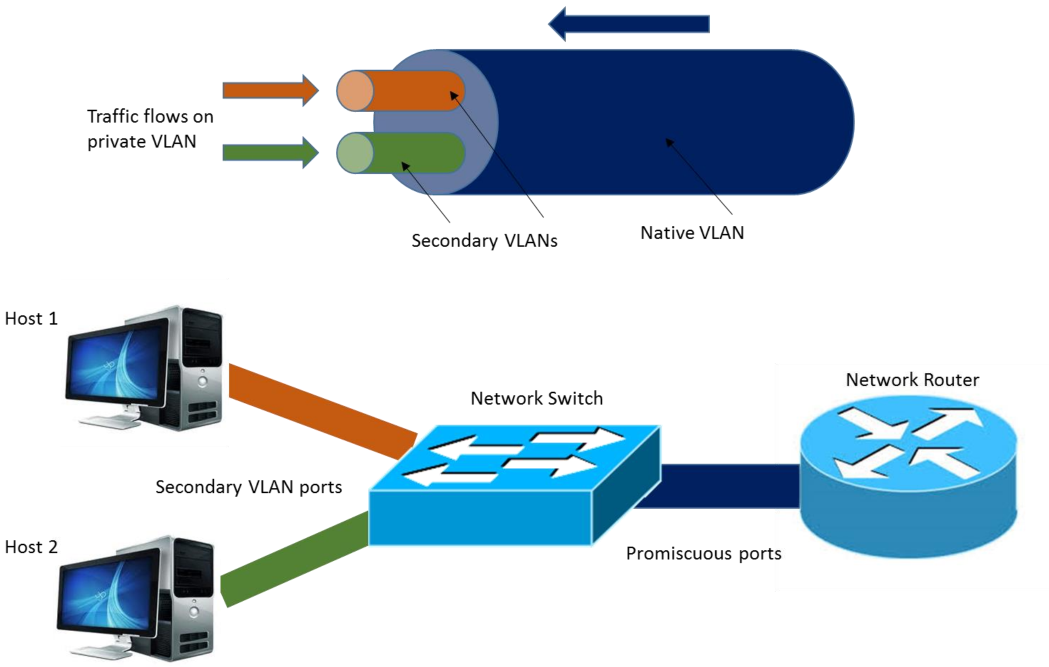


Figure 1 - VLAN

Prior to the VLAN specification, a single network port was able to shuttle traffic across a single subnet. So, if you had four non-bonded network ports on a server, you could have access to a total of four different subnets. The purpose of the VLAN specification is to allow a single network port to access multiple subnets across a single wire connected to a single switch port. Therefore, by using multiple VLANs assigned to a single switch port, you can access multiple subnets on a single network port. This means the same server with four physical network ports can access more than four subnets. Each VLAN in a local area network is identified by a VLAN tag, or VLAN ID (VID). The VID is assigned during VLAN configuration. The VID is a 12-bit identifier between 1 and 4094 that provides a unique identity for each VLAN. Each VLAN ID represents a separate subnet (or broadcast domain) with its own IP range, subnet mask, and gateway.

To configure the switch on a VLAN network, you must configure it as follows:

- Port connected to ODA VP on the switch must be configured as trunk port.
- Port on the switch must be configured for 802.1q encapsulation.
- Port security cannot be set on the trunk port.
- Port designated as trunk should be assigned a native VLAN; use 1 as default.

Note: If a Native VLAN is used on the switch trunk port, then you cannot assign that VLAN number to a Virtual Machine on ODA

Trunk and access ports

There are two ways in which a machine can be connected to a switch carrying 802.1Q VLAN traffic:

- via an “access” port, where VLAN support is handled by the switch (so the machine sees ordinary, untagged Ethernet frames);
- via a “trunk” port, where VLAN support is handled by the attached machine (which sees 802.1Q-tagged Ethernet frames).

Access Port – Untagged (or Native)

You can configure an Ethernet port as an access port. An access port transmits packets on only one, untagged VLAN. You specify which VLAN traffic that the interface carries. If you do not specify a VLAN for an access port, the interface carries traffic only on the default VLAN. The default VLAN is VLAN1 (on cisco switch)

Trunk Port - Tagged

You can configure an Ethernet port as a trunk port; a trunk port transmits

- untagged packets for the native VLAN
- tagged, encapsulated packets for multiple VLANs (Tagged traffic has the vlan id associated with it.)

Default VLAN vs Native VLAN

Cisco switches always have VLAN 1 as the default VLAN, which is needed for many protocol communications between switches like spanning-tree protocol for instance. You can't change or even delete the default VLAN, it is mandatory. The native VLAN is the only VLAN which is not tagged in a trunk, in other words, native VLAN frames are transmitted unchanged.

Understanding Access/trunk ports

In order to understand how an access/trunk port is working with VLAN in a network switch we will cover a simple scenario considering a cisco catalyst 2959 series switch.

- Case 1 – VLAN with access port on network switch

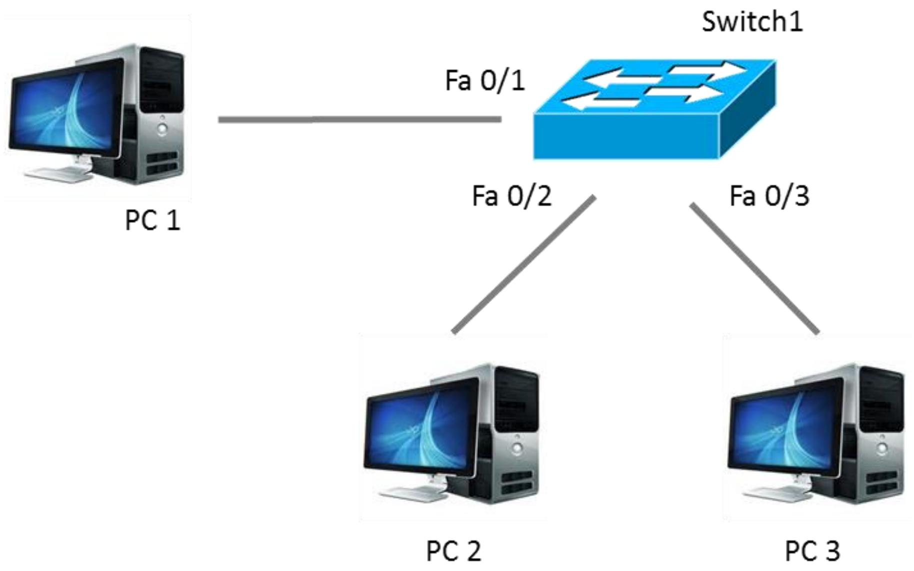


Figure 2 - Network scenario

Device	Port	IP Address	Subnet Mask	Default Gateway
PC1	FastEthernet 0/1	24.17.2.10	255.255.255.0	24.17.2.1
PC2	FastEthernet 0/2	24.17.2.20	255.255.255.0	24.17.2.1
PC3	FastEthernet 0/3	24.17.2.30	255.255.255.0	24.17.2.1

In this scenario, the 3 hosts are able to communicate as part of default VLAN (on cisco switch VLAN1).

Setting up a port (0/2) as “access” for VLAN22:

```
Switch1 (config)# interface fastethernet 0/2
Switch1 (config-if)# switchport mode access
Switch1 (config-if)# switchport mode access vlan 22
```

The ping to PC2 fails because is now in VLAN 22 and PC1/3 are still in VLAN 1. A network switch can forward traffic between two connected devices if both are in the same VLAN.

PC1 and PC3 are members of VLAN 1; therefore, they can communicate with each other. However, because they are the only members of VLAN 1, they cannot communicate with any other hosts on the network w/o a router. Network nodes that are not members of the same VLAN cannot communicate with each other if no Layer 3 switching or routing process is available to perform inter-VLAN routing.

On Switch1, assign the port connected to PC3 to the new VLAN:

```
Switch1 (config)# interface fastethernet 0/3
Switch1 (config-if)# switchport mode access
Switch1 (config-if)# switchport mode access vlan 22
```

PC2 and PC3 are now in the same VLAN, so the ping from PC2 to PC3 succeeds.

Adding also port 0/1 to VLAN22 will permit the comunication between the PC1 and PC2/3:

```
Switch1 (config)# interface fastethernet 0/1
Switch1 (config-if)# switchport mode access
Switch1 (config-if)# switchport mode access vlan 22
```

- Case 2 – VLAN with trunk ports on network switch

In the same scenario if all ports (0/1, 0/2, 0/3) are configured as trunk

```
configure terminal
Switch1 (config)# interface fastethernet 0/1
Switch1 (config)# switchport trunk encapsulation do1q
Switch1 (config)# switchport mode trunk
Switch1 (config)# switchport trunk allowed vlan 22

(…)
```

The communication between the 3 hosts will be possible thought the native VLAN1 as not tagged traffic and/or thought the VLAN22 as tagged traffic.

Why VLAN

- Broadcast Control: Broadcasts are required for the normal function of a network. Many protocols and applications depend on broadcast communication to function properly. A layer 2 switched network is in a single broadcast domain and the broadcasts can reach the network segments which are so far where a particular broadcast has no scope and consume available network bandwidth. Network segmentation with virtual local area networks (VLANs) creates a collection of isolated networks within ODA VP. Each VLAN network is a separate broadcast domain.

- Security: VLANs provide enhanced network security. In a VLAN network environment, with multiple broadcast domains, network administrators have control over each port and user. VLANs offer the ability to keep data packets from multiple networks separated

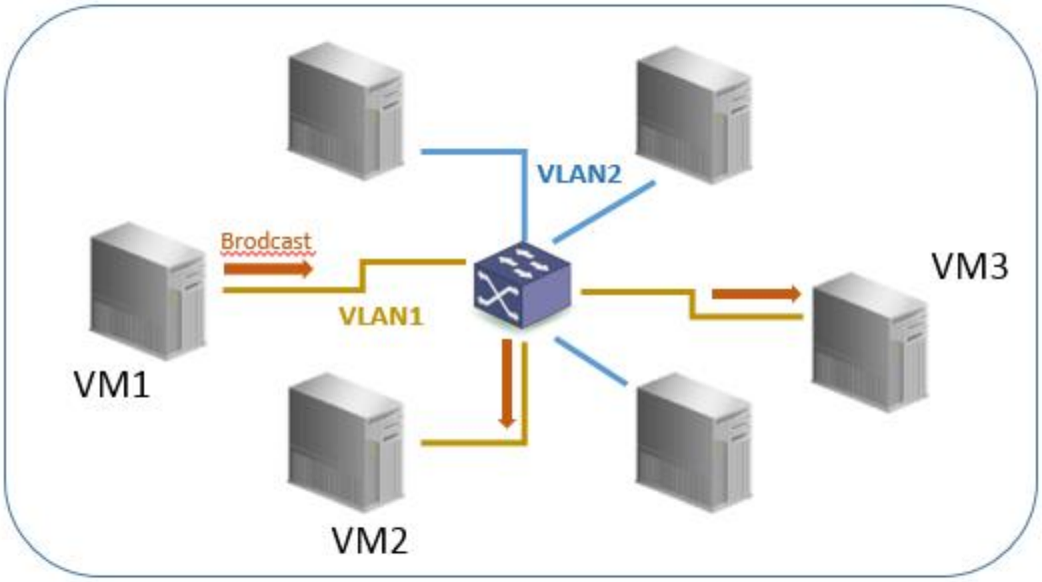


Figure 3 - VM1 sends a network broadcast but it is not propagated to all 5 VMs

Oracle Database Appliance VLAN

Oracle Database Appliance Virtualized Platform supports multiple virtual LANs, or VLANs, on the same network port or bond. Each VLAN is essentially an independent logical network operating with other VLANs over the same physical connection. This means that virtual machines deployed on different networks, connected through the same Oracle VM Server port (or bond), can have traffic directed to different VLANs.

Configure VLAN

You create VLAN interfaces to direct the traffic from several VLANs onto a single bond on each Oracle Database Appliance server node. For example, if bond1 is expected to carry traffic for VLAN with ID 2 and for VLAN with ID 3, you create a VLAN Interface for each of these VLAN segments. Each packet transmitted from virtual machines on this network is tagged with the VLAN ID for the VLAN interface specified during network creation. If you added the physical network bond during network creation, the packets can still flow through the bond, but the packets are untagged. The Ethernet switch, to which the ODA is connected, is responsible for the transmission of packets to the appropriate VLAN.

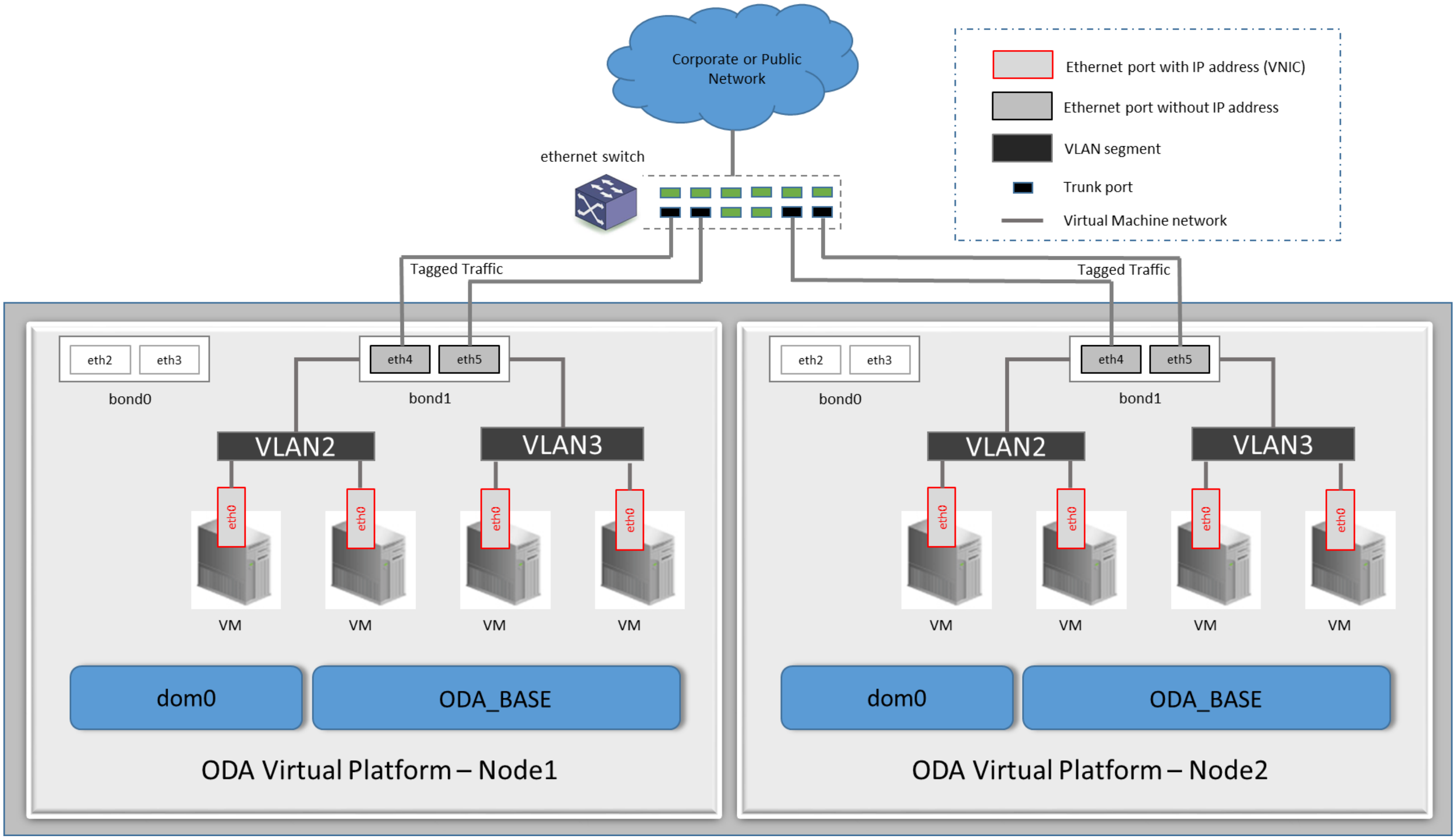


Figure 4 - Network with VLANs and VLAN Interfaces

In the diagram, titled Figure 4, "*Networks with VLANs and VLAN Interfaces*", two VLAN interfaces are defined on the bond1 for each server. The VLAN2 interface handles traffic tagged with the VLAN ID 2 and the VLAN3 interface handles traffic tagged with the VLAN ID3. Two virtual machine networks have been created, even though these networks use the same physical network infrastructure. The first virtual machine network has the VLAN2 interface attached for each server; while the second virtual machine network has the VLAN3 interface attached for each server. For each logical network, a bridge is automatically created for the specified VLAN interface. **The bridge may or may not be configured with a IP address depending on the connectivity needed from dom0 (see page 20 for an example)**. Network packets from virtual machines deployed on VLAN segment 2 travel through the bridge and acquire a tag which identifies the packets as belonging to VLAN 2. Similarly, the packets issued from the virtual machines deployed on the network for VLAN segment 3 are tagged for VLAN3 with ID 3. The packets from both networks use either path to the switch. The receiving ports on the Ethernet switch are configured using trunking or similar configuration to recognize network traffic for the two VLANs in this setup. As such, the trunk ports will direct the packets to the correct VLAN on the switch, or other connected switches.

About Network Infrastructure and Virtual Local Area Networks on Oracle Database Appliance Virtualized Platform

To specify which network should access a virtual machine, you employ network infrastructure components of Oracle Database Appliance Virtualized Platform. This section describes these infrastructure components.

Oracle Database Appliance Virtualized Platform manages all the high-level network infrastructure components for you by pre-creating the bonds and bridges for all networks. The front-end point for accessing a virtual machine will be one of the bridges defined for Dom0.

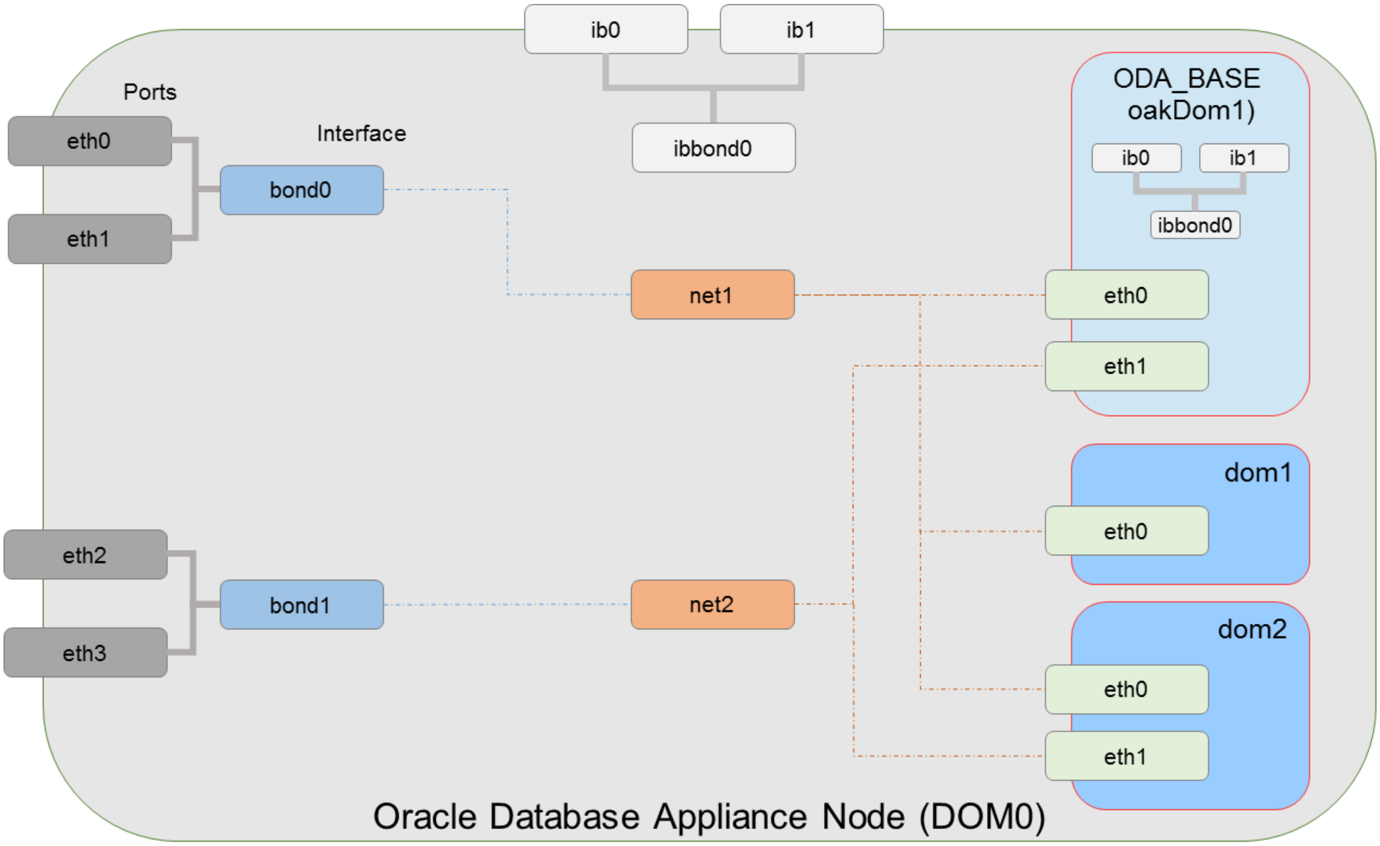


Figure 5 - Basic Virtual Machine Local Area Network (network diagram valid for ODA X5-2 / X6-2 HA with InfiniBand)

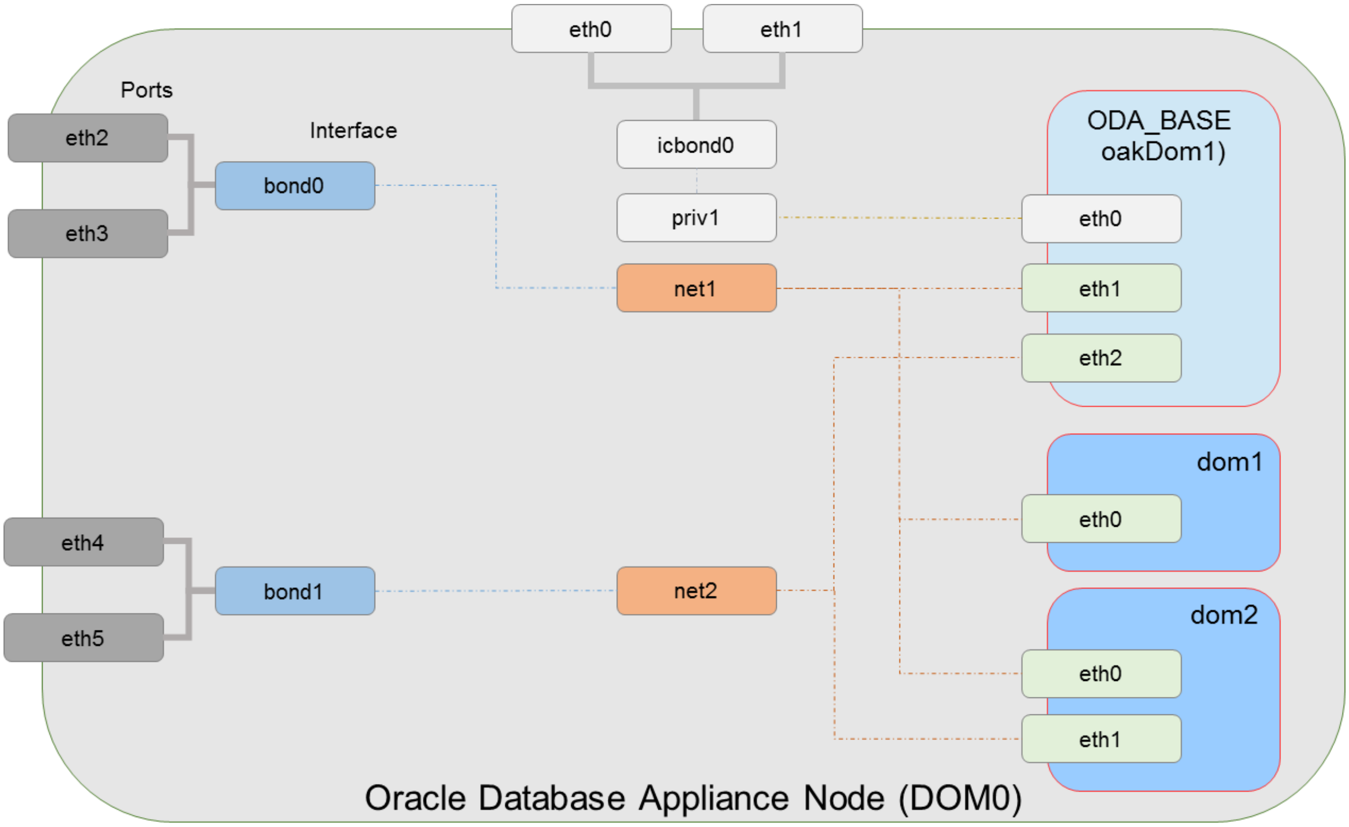


Figure 6 - Basic Virtual Machine Local Area Network (network diagram valid for ODA X3-2 / X4-2)

When you configure a virtual machine, you define which network the virtual machine should use by identifying the related bridge. For example, to connect a virtual machine named myvm1 to the net1 network, you would use the following command:

```
# oakcli modify vm myvm1 -addnetwork net1
```

Oracle Database Appliance Virtualized Platform with Virtual Local Area Networks

During the installation and configuration of Oracle software on Oracle Database Appliance Virtualized Platform, you had an opportunity to assign default Virtual Local Area Networks (VLANs) to ODA_BASE. Figure 7 - ODA VP with VLAN how's a typical Oracle Database Appliance (X3-2, X4-2) configuration using VLANs. The figure shows the same configuration as Figure 6 but with three tagged VLANs added for backups (backup), for applications (application), and for management (mgnt).

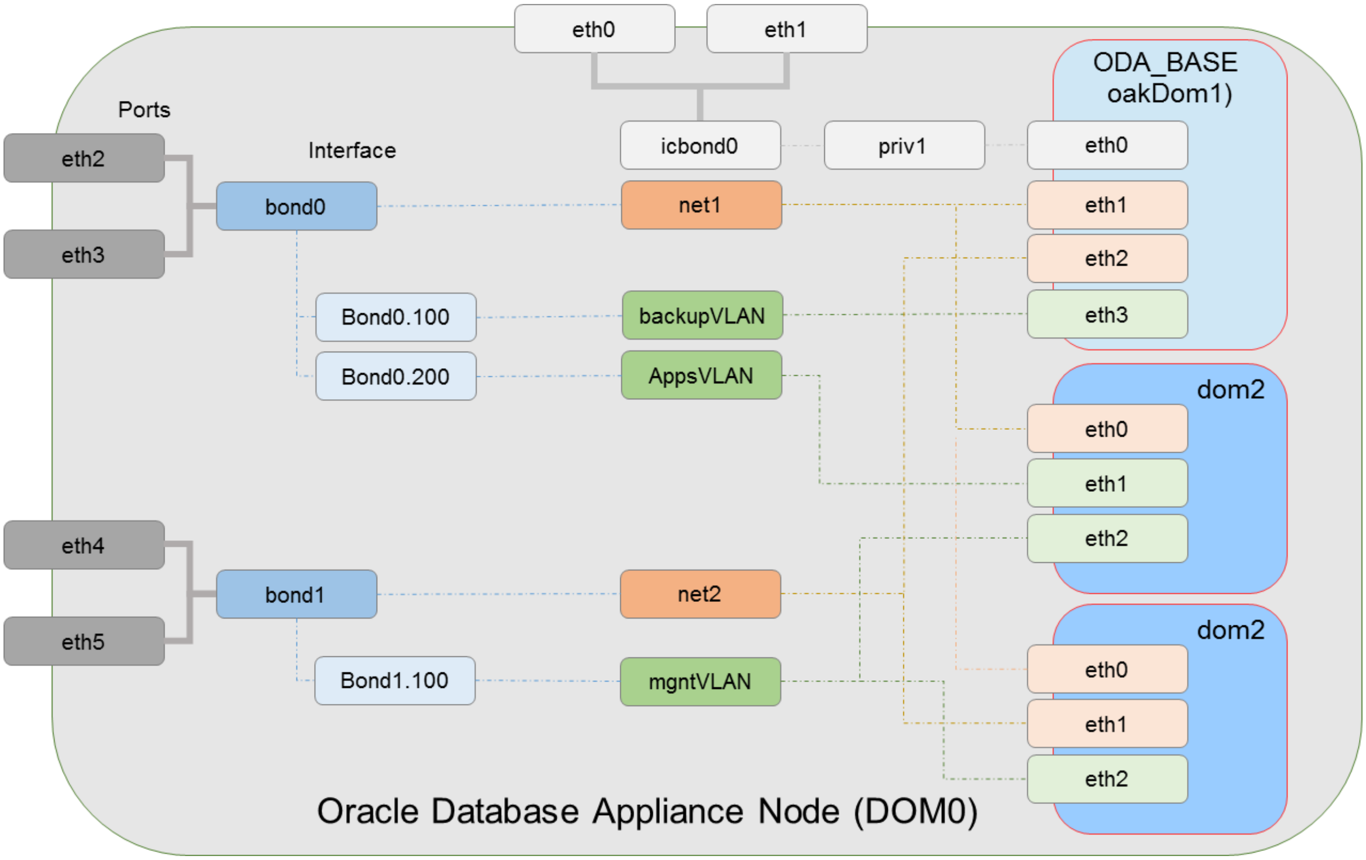


Figure 7 - ODA VP with VLAN

ODA VLAN Management

VLAN Creation

New oakcli command is available to create a VLAN from ODA_BASE

```
# oakcli create vlan <vlan name> -vlanid <vlan tag id> -if <interface name> -node <0|1>
```

- **Vlan name –** Is the logical user defined name for the network which will be used for the administrative actions related to this network. The name needs to be unique in a given node. User can create same named network on the other node of the ODA system.
- **Vlanid –** the vlan tag id identifies the id to which the vlan network belongs to. This needs to be unique for a single ODA node. The same id can be used for creating a network on the 2nd node of the ODA system.
- **Interface name –** the name of the interface on which the vlan network is to be created. For the ODA X3-2 machine there are 2 network interfaces that the customers can use ie. “bond0” and “bond1”.
- **Node –** Which ODA node to create this VLAN network

Display VLAN Information

The following command can be used to list the VLAN information:

```
# oakcli show vlan
```

Example:

```
[root@oda9n1 ~]# oakcli show vlan

       NAME                      ID    INTERFACE   NODENUM

       net1                      1     bond0       0
       net1                      1     bond0       1
       net2                      1     bond1       0
       net2                      1     bond1       1
       priv1                     1     icbond0     0
       priv1                     1     icbond0     1
       vlan187                   187   bond0       0
       vlan187                   187   bond0       1
```

Delete VLAN

Allows a previously created user network to be removed. The default networks (net1, net2, priv1) cannot be deleted.

```
# oakcli delete vlan <vlan name> -node <0|1>
```

Managing Virtual Local Area Networks for User Domains

Assign VLAN to a Guest VM

Once you have created a VLAN you can assign it to DomU using the following command:

```
# oakcli modify vm <vm name> -addnetwork <network name>
```

Example:

```
[root@oda9n1 ~]# oakcli modify vm myvm1 -addnetwork vlan187
```

This command specifies only the bridge information (by default it will use the network paravirtualized driver: **netfront**), other option like type, MAC address cannot be specified with this command.

DO NOT use this command to assign VLAN to HVM Guests (example: Windows guest without PV Drivers).   For such guests, use the “oakcli configure” command and specify the type=**ioemu** option as well along with the network bridge:

```
# oakcli configure vm <vm name> -network <List of networks and MAC address>
```

Example:

```
[root@oda9n1 ~]# oakcli configure vm myvm1 -network "['type=ioemu,bridge=net1',
'type=ioemu,bridge=vlan187' ]"

# oakcli show vm myvm1 |grep -A 1 Networks
       Networks        :    |type= ioemu,bridge=NET1||type=n
                            etfront,bridge=vlan187|
```

Removing VLAN from a Guest VM

```
# oakcli modify vm <vm name> -deletenetwork <network name>
```

Example:

```
[root@oda9n1 ~]# oakcli modify vm myvm1 -deletenetwork vlan187
```

In case of HVM guest using ioemu network device type (reconfigure without VLAN187):

```
[root@oda9n1 ~]# oakcli configure vm myvm1 -network "['type=ioemu,bridge=net1']"
```

Note:

- VLANs are created on per node basis.
- If a VM resides on the shared repository, it can be started on any node.
- You should create same named VLAN network on both nodes where the VM belongs to. If the needed network is not available, then the VM cannot be started on that node.

Managing Virtual Local Area Networks for ODA_BASE

Create VLAN for ODA_BASE

The VLAN for ODA_BASE is created from dom0 prior the ODA_BASE deployment using the following command:

```
# oakcli create vlan <vlan name> -vlanid <vlan tag id> -if <interface name>
```

Note: as the VLAN created from dom0 is not node specific, same VLAN is created on both nodes.
Once the ODA_BASE has been deployed, the VLAN needs to be created from ODA_BASE, if you try to create a VLAN from dom0 post ODA_BASE deploy, you will get the following error message:

Example:

```
[root@oda91 ~]# oakcli create vlan vlan30 -vlanid 30 -if bond0
ERROR: Please create VLAN in oda_base
```

Assign VLAN to ODA_BASE

A VLAN can be assigned to ODA_BASE during the deployment, example:

```
[root@oda9n1 ~]# oakcli deploy oda_base
Enter the template location: /OVS/templateBuild-2012-12-22-12-05.tar.gz
Core Licensing Options:
        1. 2 CPU Cores
        2. 4 CPU Cores
        3. 6 CPU Cores
        4. 8 CPU Cores
        5. 10 CPU Cores
        6. 12 CPU Cores
        7. 14 CPU Cores
        8. 16 CPU Cores
        9. 24 CPU Cores
Selection [1 : 6] : 3
ODA base domain memory in GB (min 16, max 244)(Current Memory 150G)[default 192]  : 32
Additional vlan networks to be assigned to oda_base ? (y/n) : y
Select the network to assign [Test00,Test01,Test02,Test03]: Test00
Additional vlan networks to be assigned to oda_base ? (y/n) : y
Select the network to assign [Test00,Test01,Test02,Test03]: Test01
Additional vlan networks to be assigned to oda_base ? (y/n) : n
```

Modifying VLAN network

From “dom0” Use the command “oakcli configure oda_base” for modifying the VLAN assignment to ODA_BASE. It supports both addition and deletion of VLAN networks. Any new VLAN has to be created from ODA_BASE once it’s deployed.

Example:

```
[root@oda91 ~]# oakcli configure oda_base
Core Licensing Options:
       1. 2 CPU Cores
       2. 4 CPU Cores
       3. 6 CPU Cores
       4. 8 CPU Cores
       5. 10 CPU Cores
       6. 12 CPU Cores
       7. 14 CPU Cores
       8. 16 CPU Cores
       9. 24 CPU Cores
       Current CPU Cores    :12
       Selection[1 .. 9](default 24 CPU Cores)    : 6
       ODA base domain memory in GB(min 16, max 244)(Current Memory 192G)[default 192]
       :
INFO: Using default memory size i.e. 192 GB
Additional vlan networks to be assigned to oda_base ? (y/n)  : y
Select the network to assign [Test00,Test01,Test02,Test03]: Test00
Vlan network to be removed from oda_base ? (y/n)  [n]: n
INFO: Node 0:Configured oda base pool
INFO: Node 1:Configured oda base pool
INFO: Node 0:ODA Base configured with new memory
INFO: Node 0:ODA Base configured with new vcpus
INFO: Changes will be incorporated after the domain is restarted on Node 0.
INFO: Node 1:ODA Base configured with new memory
INFO: Node 1:ODA Base configured with new vcpus
INFO: Changes will be incorporated after the domain is restarted on Node 1.
INFO: Updating /etc/sysctl.conf in oda_base domain with parameter
"vm.nr_hugepages=51626"
```

Note: while adding new VLANs, ensure that the VLAN exists on both the nodes (“`oakcli show vlan`”)

VLAN cases study

Case 1 - Create 1 VLAN interface and 1 Management VLAN for Guest VM, ODA_BASE and dom0

The scenario we would like to cover on this case study is described by the following picture:

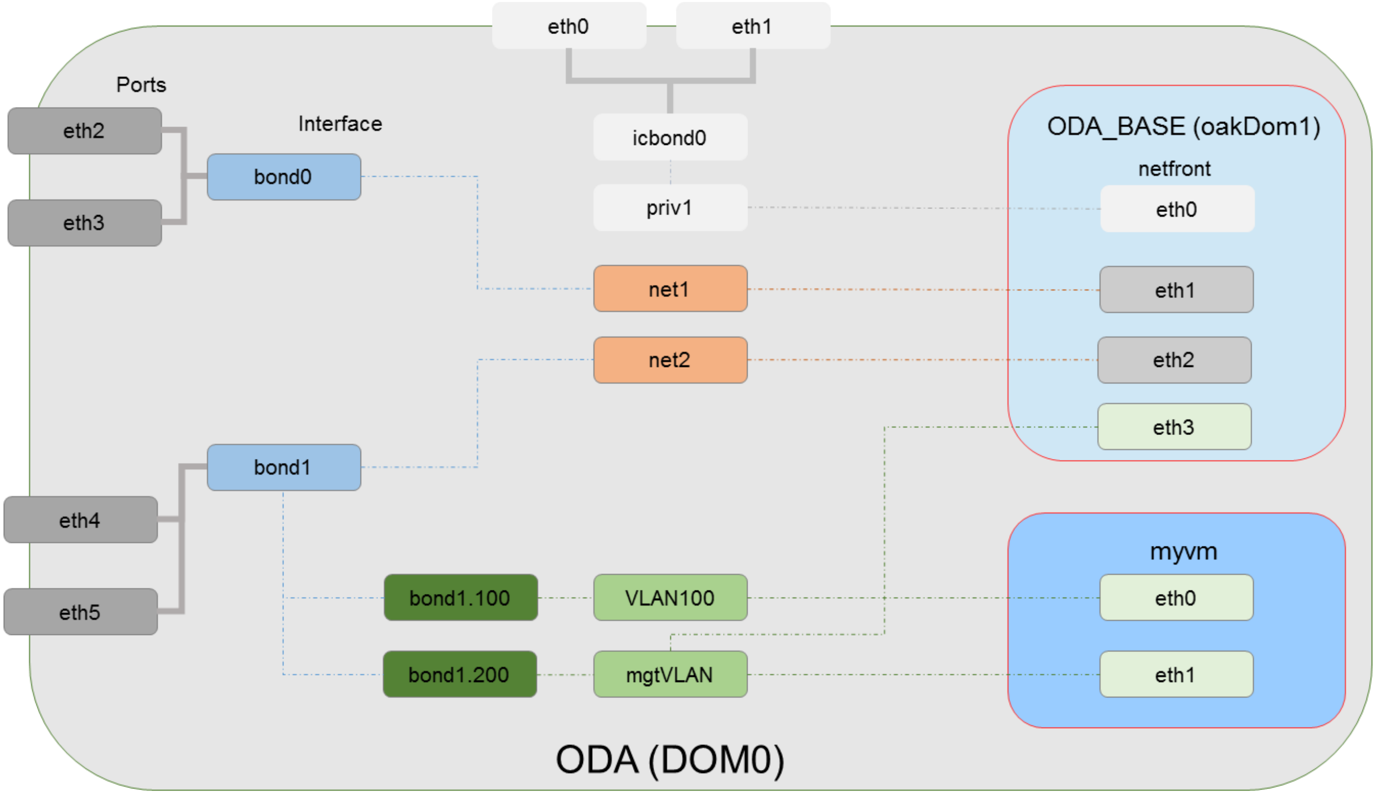


Figure 8 - VLAN case study, scenario 1

We have two VLANs to keep Guest VMs on separate network and using a management VLAN for Guest VM and ODA_BASE.

Configuration Steps

1. Create the VLANs from ODA_BASE (on both nodes)

```
# oakcli create vlan VLAN100 -vlanid 100 -if bond1 -node 0
# oakcli create vlan VLAN100 -vlanid 100 -if bond1 -node 1

# oakcli create vlan mgtVLAN -vlanid 200 -if bond1 -node 0
# oakcli create vlan mgtVLAN -vlanid 200 -if bond1 -node 1
```

2. Assign the VLAN to the Guest VM “myvm” (from ODA_BASE)

```
# oakcli modify vm myvm -addnetwork VLAN100
# oakcli modify vm myvm -addnetwork mgtVLAN
```

Note: Once a VLAN has been added to the Guest VM and restarted a new VNIC will be up running. Having the guest VM with eth0, the new attached VLANs will start eth1, eth2, ... (check with “`ifconfig -a`”)

3. Assign the VLAN to ODA_BASE (from dom0)

```
# oakcli configure oda_base
Core Licensing Options:
       1. 2 CPU Cores
       2. 4 CPU Cores
       3. 6 CPU Cores
       4. 8 CPU Cores
       5. 10 CPU Cores
       6. 12 CPU Cores
       7. 14 CPU Cores
       8. 16 CPU Cores
       9. 24 CPU Cores
       Current CPU Cores    :12
       Selection[1 .. 9](default 24 CPU Cores)    : 6
       ODA base domain memory in GB(min 16, max 244)(Current Memory 192G)[default 192]      :
INFO: Using default memory size i.e. 192 GB
Additional vlan networks to be assigned to oda_base ? (y/n) [n]: y
Select the network to assign (VLAN100,mgtVLAN): mgtVLAN
Additional vlan networks to be assigned to oda_base ? (y/n) [n]:
Vlan network to be removed from oda_base ? (y/n) [n]: n
INFO: Node 0:Configured oda base pool
INFO: Node 1:Configured oda base pool
INFO: Node 0:ODA Base configured with new memory
INFO: Node 0:ODA Base configured with new vcpus
INFO: Node 0:New vlans added in ODA Base
INFO: Changes will be incorporated after the domain is restarted on Node 0.
INFO: Node 1:ODA Base configured with new memory
INFO: Node 1:ODA Base configured with new vcpus
INFO: Node 1:New vlans added in ODA Base
INFO: Changes will be incorporated after the domain is restarted on Node 1.
INFO: Updating /etc/sysctl.conf in oda_base domain with parameter "vm.nr_hugepages=51626"
```

4. Restart ODA_BASE (from dom0)

```
# oakcli restart oda_base
```

5. Start the Guest VM

```
# oakcli start vm myvm
```

6. Configure the management network IP address for dom0 (on both nodes)

Add a fixed IP address into the *management* VLAN configuration file "*/etc/sysconfig/network-scripts/ifcfg-mgtVLAN*" created by the steps above on **dom0** (on <u>both</u> ODA nodes):

```
From

#This file was dynamically created by Oracle VM Manager. Please do not edit
DEVICE=mgtVLAN
TYPE=Bridge
BOOTPROTO=none
HWADDR=<MAC ADDRESS>
ONBOOT=yes
DELAY=0

to

#This file was dynamically created by Oracle VM Manager. Please do not edit
DEVICE=mgtVLAN
TYPE=Bridge
BOOTPROTO=none
HWADDR=<MAC ADDRESS>
ONBOOT=yes
DELAY=0
IPADDR=<your ip>
NETMASK=<your netmask>
GATEWAY=<your gtw>
```

The above settings will permit to have dom0 achievable on the new *management* VLAN (bridge) created.

Note: It is not mandatory to assign a IP on VLAN to the dom0 bridge interface. If you need to access the dom0 via that vlan network, then you need to plumb a IP on the subnet running in the vlan network to the bridge interface (example: mgtVLAN) on dom0.

Restart the network to get the changes:

```
# /etc/init.d/network restart
```

7. Configure the management network IP address for ODA_BASE

Once the VLAN has been added to ODA_BASE and restarted, a new VNIC will be available (in this example “eth3”), check with “`ifconfig -a`”. Create the related device configuration file such “*/etc/sysconfig/network-scripts/ifcfg-eth3*”. As we want a fixed IP we need to change it as following:

```
# Xen Virtual Ethernet
DEVICE=eth3
BOOTPROTO=none
ONBOOT=yes
IPADDR=<your ip>
NETMASK=<your netmask>
GATEWAY=<your gtw>
```

Then we can restart the network to get the changes:

```
# /etc/init.d/network restart
```

8. Configure the IPs address for the VLAN on your guest VM (myvm):

Two new eth devices (VNICs) are now available on your guest VM; you need to configure them creating two NIC configuration files. If this is the VM guest first boot, you could also use the “send keys” feature.

```
/etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
BOOTPROTO=none
ONBOOT=yes
IPADDR=<your ip>
NETMASK=<your netmask>
GATEWAY=<your gtw>

/etc/sysconfig/network-scripts/ifcfg-eth1
DEVICE=eth1
BOOTPROTO=none
ONBOOT=yes
IPADDR=<your ip>
NETMASK=<your netmask>
GATEWAY=<your gtw>
```

Restart the network to make the changes on your guest VM (myvm):

```
# /etc/init.d/network restart
```

Case 2 - Configuring ODA nodes to be on separate networks

Scope of this case study is to show how setup a Single instance Oracle database on different nodes to be on different networks. The following picture is describing the network setup:

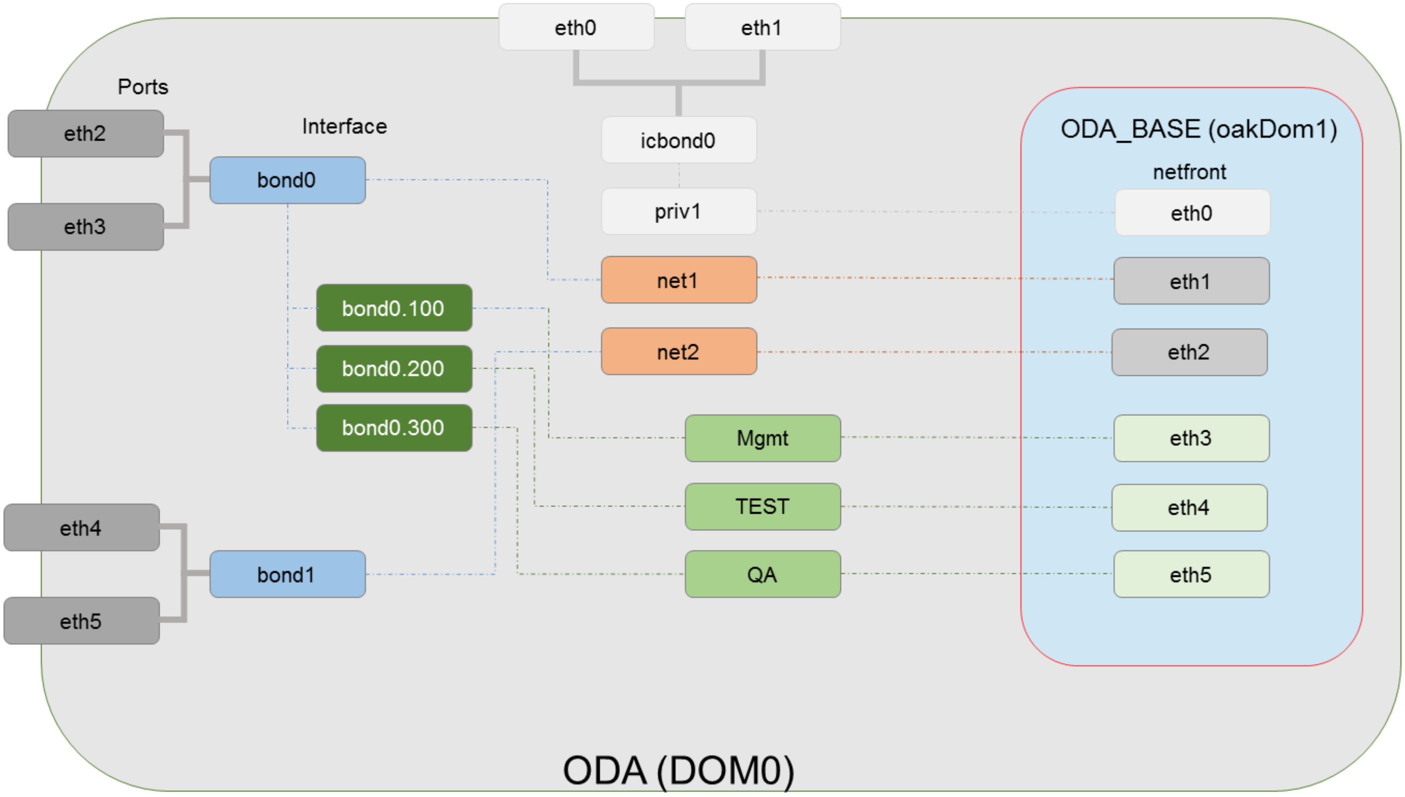


Figure 9 - VLAN case study, scenario 2

Example network configuration:

Network	VLAN ID	Subnet
Mgmt	100	10.10.100.0/24
TEST	200	10.10.200.0/24
QA	300	10.10.300.0/24

The “mgmt.” network is the regular GI/RAC public network, this network has the scan, vip configured. This is attached to both the ODA_BASE.

In addition, we have “TEST” and “QA” vlan networks attached to both the ODA_BASEs. Assuming eth4 in ODA_BASE corresponds to the “TEST” network and eth5 in ODA_BASE corresponds to the “QA” network.

Configure the ODA_BASE with the IP from respective subnets to both Test and QA networks.

	Node1	**Node2**
Eth4 (TEST network)	10.10.200.11 10.10.200.21 (vip)	10.10.200.12 10.10.200.22 (vip)
Eth5 (QA network)	10.10.300.11 10.10.300.21 (vip)	10.10.300.12 10.10.300.22 (vip)

The Steps

1. Create the required VLANs
   a. If ODA_BASE is not deployed yet
      - Execute from dom0 the following commands:

```
# oakcli create vlan mgtVLAN -vlanid 100 -if bond0
# oakcli create vlan TEST -vlanid 200 -if bond0
# oakcli create vlan QA -vlanid 300 -if bond0
```

      - Deploy ODA_BASE using the VLAN created above:

```
# oakcli deploy oda_base
Enter the template location: /OVS/templateBuild-2012-12-22-12-05.tar.gz
Core Licensing Options:
        1. 2 CPU Cores
        2. 4 CPU Cores
        3. 6 CPU Cores
        4. 8 CPU Cores
        5. 10 CPU Cores
        6. 12 CPU Cores
        7. 14 CPU Cores
        8. 16 CPU Cores
        9. 24 CPU Cores
Selection [1 : 6] : 3
ODA base domain memory in GB (min 16, max 244)(Current Memory 150G)[default 192]  : 32
Additional vlan networks to be assigned to oda_base ? (y/n) : y
Select the network to assign [mgtVLAN,TEST,QA]: mgtVLAN
Additional vlan networks to be assigned to oda_base ? (y/n) : y
Select the network to assign [mgtVLAN,TEST,QA]: TEST
Additional vlan networks to be assigned to oda_base ? (y/n) : y
Select the network to assign [mgtVLAN,TEST,QA]: QA
Additional vlan networks to be assigned to oda_base ? (y/n) : n
(...)
```

Once the ODA_BASE deploy is done, the network mapping will be as following:

```
icbond0   <-> priv1   <-> eth0   |  ibbond0   <-> ib0
                                 |  ibbond0   <-> ib1
bond0     <-> net1    <-> eth1   |  bond0     <-> net1    <-> eth0
bond1     <-> net2    <-> eth2   |  bond1     <-> net2    <-> eth1
bond0.100 <-> mgtVLAN <-> eth3   |  bond0.100 <-> mgtVLAN <-> eth2
bond0.200 <-> TEST    <-> eth4   |  bond0.200 <-> TEST    <-> eth3
bond0.300 <-> QA      <-> eth5   |  bond0.300 <-> QA      <-> eth4

This is applicable to X3-2/X4-2 | This is applicable to X5-2/X6-2 HA
```

Restart ODA_BASE

```
# oakcli restart oda_base
```

b. If ODA_BASE deploy has been done

- Execute from ODA_BASE the following commands:

```
# oakcli create vlan mgtVLAN -vlanid 100 -if bond0 -node 0
# oakcli create vlan TEST -vlanid 200 -if bond0 -node 0
# oakcli create vlan QA -vlanid 300 -if bond0 -node 0

# oakcli create vlan mgtVLAN -vlanid 100 -if bond0 -node 1
# oakcli create vlan TEST -vlanid 200 -if bond0 -node 1
# oakcli create vlan QA -vlanid 300 -if bond0 -node 1
```

- Modify ODA_BASE network settings adding the VLANs, configure ODA_BASE from dom0

```
# oakcli configure oda_base
Core Licensing Options:
        1. 2 CPU Cores
        2. 4 CPU Cores
        3. 6 CPU Cores
        4. 8 CPU Cores
        5. 10 CPU Cores
        6. 12 CPU Cores
        7. 14 CPU Cores
        8. 16 CPU Cores
        9. 24 CPU Cores
        Current CPU Cores    :12
        Selection[1 .. 9](default 24 CPU Cores)    : 6
        ODA base domain memory in GB(min 16, max 244)(Current Memory 192G)[default 192]
        :
INFO: Using default memory size i.e. 192 GB
Additional vlan networks to be assigned to oda_base ? (y/n)  : y
Select the network to assign [mgtVLAN,TEST,QA]: mgtVLAN
Additional vlan networks to be assigned to oda_base ? (y/n)  : y
Select the network to assign [mgtVLAN,TEST,QA]: TEST
Additional vlan networks to be assigned to oda_base ? (y/n)  : y
Select the network to assign [mgtVLAN,TEST,QA]: QA
Additional vlan networks to be assigned to oda_base ? (y/n)  : n
Vlan network to be removed from oda_base ? (y/n)  [n]: n
(...)
```

- Restart ODA_BASE

```
# oakcli restart oda_base
```

2. Configure the ODA_BASE with the IP from respective subnets to both “Test” and “QA” networks

   Once the VLAN has been added to ODA_BASE and restarted we need to create the NIC configuration file for eth4 (“/etc/sysconfig/network-scripts/ifcfg-eth4”) as following:

```
DEVICE=eth4
BOOTPROTO=none
ONBOOT=yes
IPADDR=10.10.200.11
NETMASK=255.255.255.0
GATEWAY=10.10.200.1
```

   and for eth5 (“/etc/sysconfig/network-scripts/ifcfg-eth5”):

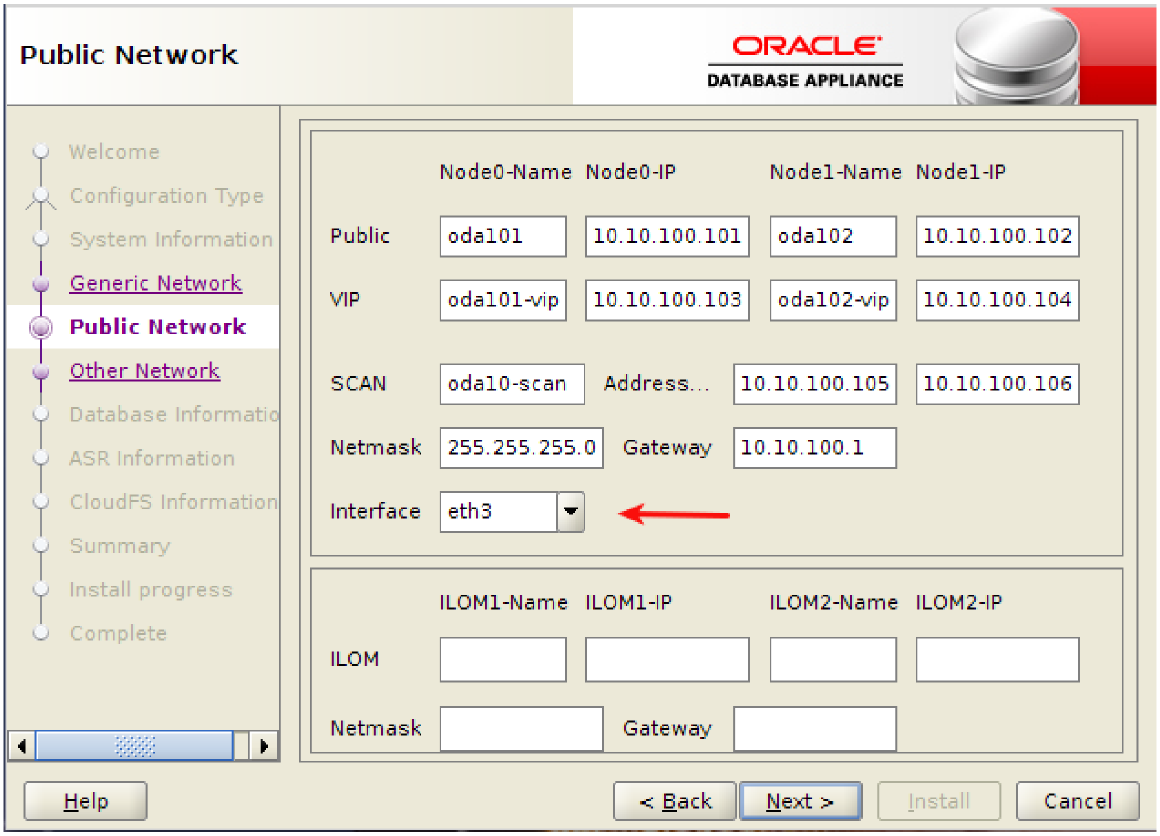
```
# Xen Virtual Ethernet
DEVICE=eth5
BOOTPROTO=none
ONBOOT=yes
HWADDR=...........
IPADDR=10.10.300.11
NETMASK=255.255.255.0
GATEWAY=10.10.300.1
```

   Then we can restart the network to get the changes:

```
# /etc/init.d/network restart
```

3. Deploy the Oracle Software (GI and RDBMS) into ODA_BASE

Using the Oracle Appliance Manager and configuring the public network, you must choose the interface “eth3” (writing into the Interface box), in case of X5-2 it will be “eth2”, example:



Public Network

	Node0-Name	Node0-IP	Node1-Name	Node1-IP
Public	oda101	10.10.100.101	oda102	10.10.100.102
VIP	oda101-vip	10.10.100.103	oda102-vip	10.10.100.104

SCAN: oda10-scan Address... 10.10.100.105 10.10.100.106

Netmask: 255.255.255.0 Gateway: 10.10.100.1

Interface: eth3

	ILOM1-Name	ILOM1-IP	ILOM2-Name	ILOM2-IP
ILOM				

Netmask: Gateway:

Figure 10 - Oracle Database Appliance Manager Configurator (X3-2, X4-2)

4. Create 2 new networks corresponding to the Test and QA networks

   Once the GI/RDBMS have been deployed successfully, as grid user, the command to use is “oifcfg”. After creating, you can list it and verify as showed below

```
[grid ~]$ oifcfg

Name:
    oifcfg - Oracle Interface Configuration Tool.

Usage:  oifcfg iflist [-p [-n]]
    oifcfg setif {-node <nodename> | -global} {<if_name>/<subnet>:<if_type>}...
    oifcfg getif [-node <nodename> | -global] [ -if <if_name>[/<subnet>] [-type <if_type>] ]
    oifcfg delif {{-node <nodename> | -global} [<if_name>[/<subnet>]] [-force] | -force}
    oifcfg [-help]

    <nodename> - name of the host, as known to a communications network
    <if_name>  - name by which the interface is configured in the system
    <subnet>   - subnet address of the interface
    <if_type>  - type of the interface { cluster_interconnect | public }
```

```
[grid ~]$ oifcfg setif -global eth4/10.10.200.0:public
[grid ~]$ oifcfg setif -global eth5/10.10.300.0:public

[grid ~]$ oifcfg getif
eth0  192.168.16.0  global  cluster_interconnect,asm
eth1  10.10.100.0  global  public
eth3  10.10.200.0  global  public
eth4  10.10.300.0  global  public
```

5. Add a new network cluster resource, You need to be 'root' in order to execute the following command otherwise you will get "PRCN-2018 : Current user oracle is not a privileged user." error message

```
[root ~]# srvctl add network -h

Adds a network configuration to the Oracle Clusterware.

Usage: srvctl add network [-k <net_num>] -S <subnet>/<netmask>/[if1[|if2...]] [-w
<network_type>] [-v]
    -k <net_num>             network number (default number is 1)
    -S <subnet>/<netmask>/[if1[|if2...]]  NET address spec for network
    -w <network_type>        The network type (static, dhcp, mixed)
    -h                       Print usage
    -v                       Verbose output
```

```
[root ~]# srvctl add network -k 220 -S 10.10.200.0/255.255.255.0/eth4 -w static
[root ~]# srvctl add network -k 230 -S 10.10.300.0/255.255.255.0/eth5 -w static

[root ~]# srvctl config network
Network exists: 0/192.168.16.0/255.255.248.0/eth0, type static
Network exists: 1/10.10.100.0//255.255.255.0/eth1, type static
Network exists: 200/10.10.200.0//255.255.255.0/eth4, type static
Network exists: 300/10.10.300.0//255.255.255.0/eth5, type static
```

6. Add vips for the Test and QA networks

```
[root ~]# srvctl add vip -h

Adds a VIP to the Oracle Clusterware.

Usage: srvctl add vip -n <node_name> -k <network_number> -A <name|ip>/<netmask>/[if1[|if2...]]
[-v]
    -n <node_name>           Node name
    -A <vip_name|ip>/<netmask>/[if1[|if2...]]  VIP address spec for node applications
    -k <net_num>             network number (default number is 1)
    -h                       Print usage
    -v                       Verbose output
```

```
[root ~]# srvctl add vip -n oda11 -A 10.10.200.21/255.255.255.0 -k 200
[root ~]# srvctl add vip -n oda12 -A 10.10.200.22/255.255.255.0 -k 200

[root ~]# srvctl add vip -n oda11 -A 10.10.300.21/255.255.255.0 -k 300
[root ~]# srvctl add vip -n oda12 -A 10.10.300.22/255.255.255.0 -k 300
```

Start the Virtual IP Address Interface. As a best practice, this should be done as grid user, where

oda11-tvip is 10.10.200.21

oda12-tvip is 10.10.200.22

oda11-qvip is 10.10.300.21

oda12-qvip is 10.10.300.22

```
[grid ~]$ srvctl start vip -vip oda11-tvip
[grid ~]$ srvctl start vip -vip oda12-tvip

[grid ~]$ srvctl start vip -vip oda11-qvip
[grid ~]$ srvctl start vip -vip oda12-qvip
```

7. Add listeners in the srvctl for the Test and QA networks

```
[grid ~]$ srvctl add listener -h

Adds a listener configuration to the Oracle Clusterware.

Usage: srvctl add listener [-l <lsnr_name>] [-s] [-p "[TCP:]<port>[,
...][/IPC:<key>][/NMP:<pipe_name>][/TCPS:<s_port>] [/SDP:<port>]"] [-o <oracle_home>] [-k
<net_num>]
    -l <lsnr_name>           Listener name (default name is LISTENER)
    -o <oracle_home>         ORACLE_HOME path (default value is CRS_HOME)
    -k <net_num>             network number (default number is 1)
    -s                       Skip the checking of ports
    -p "[TCP:]<port>[, ...][/IPC:<key>][/NMP:<pipe_name>][/TCPS:<s_port>] [/SDP:<port>]"
Comma separated tcp ports or listener endpoints
    -h                       Print usage
```

```
[grid ~]$ srvctl add listener -l LIST_200 -p 1721 -k 200 -s
[grid ~]$ srvctl start listener -l LIST_200

[grid ~]$ srvctl add listener -l LIST_300 -p 1821 -k 300 -s
[grid ~]$ srvctl start listener -l LIST_300
```

8. Configure the Test database to register with the listener on Test network and configure the QA database to register with the listener on QA network.

   To ensure that connections to the remote listener are only redirected to the local listener on the same network, LISTENER_NETWORKS parameter needs to be set in the pfile or spfile for the database instance. Connect as oracle user your target instance and execute the following commands (replacing the port and the IP address/hostname here with your VIP IP address on the desired VLAN):

```
####### Instance TEST running on node1 #############
ALTER SYSTEM
SET LISTENER_NETWORKS = '((NAME=vlan_network)(LOCAL_LISTENER=(ADDRESS =
(PROTOCOL=TCP)(HOST=oda11)(PORT=1821)))(REMOTE_LISTENER=oda12:1721)) 'SCOPE=BOTH;


####### Instance QA running on node2 #############
ALTER SYSTEM
SET LISTENER_NETWORKS = '((NAME=vlan_network)(LOCAL_LISTENER=(ADDRESS =
(PROTOCOL=TCP)(HOST=oda12)(PORT=1721)))(REMOTE_LISTENER=oda11:1821)) 'SCOPE=BOTH;
```

Note: In the local_listener instead to use

(ADDRESS = (PROTOCOL=TCP)(HOST=oda11)(PORT=1821))

(ADDRESS = (PROTOCOL=TCP)(HOST=oda12)(PORT=1721))

you could use a TNS alias (tnsnames.ora)

Case 3 – Public VLAN interface for Guest VMs and separate backup VLAN interface for databases and Guest VMs

In the following case we have 2 VLANs for backup purpose (BCKUPDB for the Database and BCKUPVM for Guest VMs) and a public separate VLAN for the Guest VMs (PUBVM).

The network switch used ports are configured as “trunk port” and in case of port 2/4 (eth2/eth3) will transmits untagged packets for the native VLAN (traffic coming from ODA_BASE eth1 – public network) and tagged, encapsulate packets for the VLAN with id=10. The ports port 7/9 (eth4/eth5) will transmits tagged, encapsulate packets for the VLANs with id=100 and id=200.

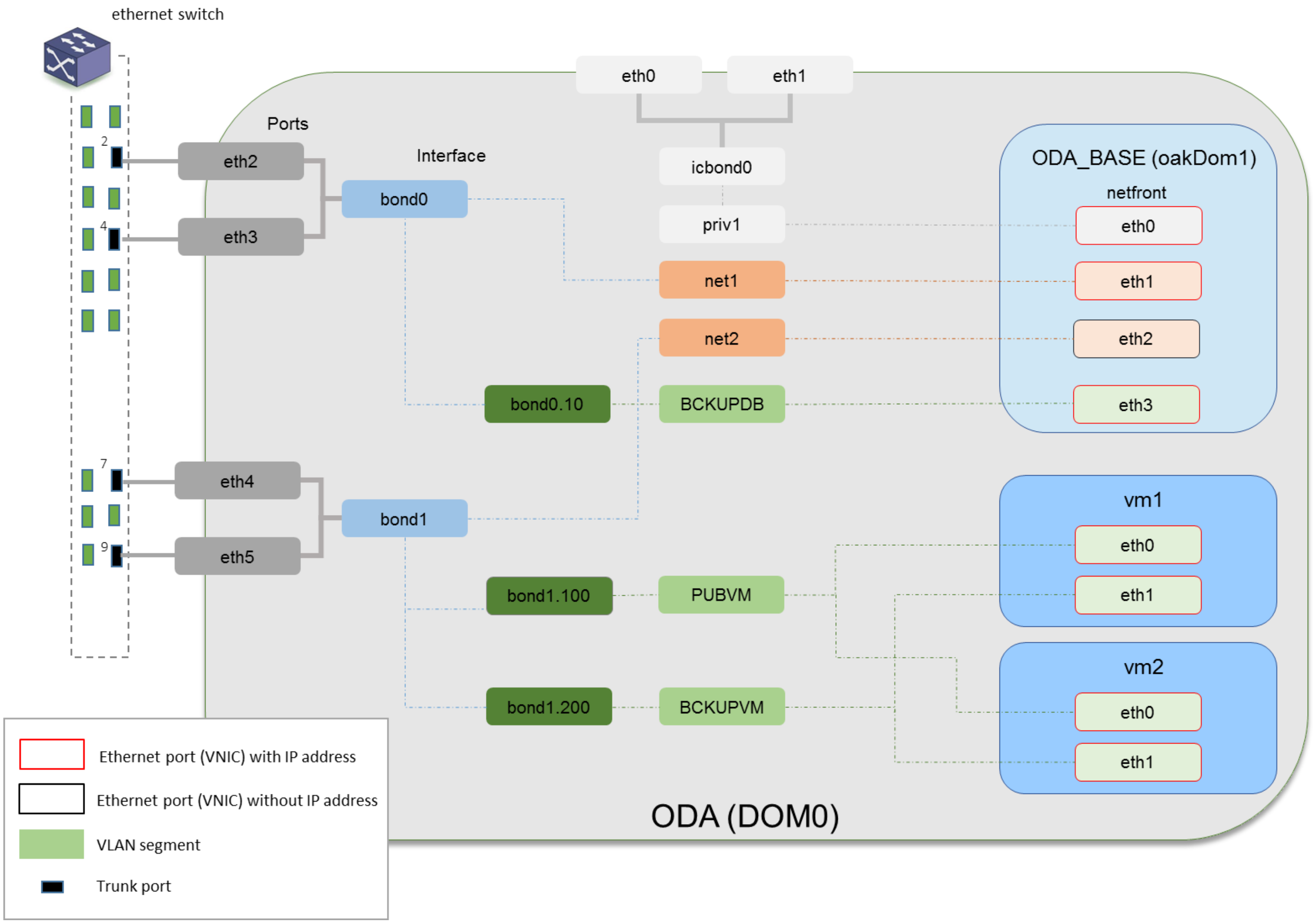


Figure 11 - VLAN case study, scenario 3

ODA Configuration Steps

1. Create the VLANs from ODA_BASE (on both nodes)

```
# oakcli create vlan BCKUPDB -vlanid 10 -if bond0 -node 0
# oakcli create vlan BCKUPDB -vlanid 10 -if bond0 -node 1

# oakcli create vlan PUBVM -vlanid 100 -if bond1 -node 0
# oakcli create vlan PUBVM -vlanid 100 -if bond1 -node 1

# oakcli create vlan BCKUPVM -vlanid 200 -if bond1 -node 0
# oakcli create vlan BCKUPVM -vlanid 200 -if bond1 -node 1
```

2. Assign the VLANs (`PUBVM, BCKUPVM`) to the Guest VMs “vm1/vm2” (from ODA_BASE)

```
# oakcli modify vm vm1 -addnetwork PUBVM
# oakcli modify vm vm1 -addnetwork BCKUPVM

# oakcli modify vm vm2 -addnetwork PUBVM
# oakcli modify vm vm2 -addnetwork BCKUPVM
```

Note: Once a VLAN has been added to the Guest VM and restarted, a new VNIC will be available, check with “`ifconfig –a`”. Create the related device configuration file such “*/etc/sysconfig/network-scripts/ifcfg-ethX*”. Having the guest VM with eth0, the new attached VLANs will start eth1, eth2, ...

3. Assign the VLAN (`BCKUPDB`) to ODA_BASE (from dom0)

```
# oakcli configure oda_base
Core Licensing Options:
       1. 2 CPU Cores
       2. 4 CPU Cores
       3. 6 CPU Cores
       4. 8 CPU Cores
       5. 10 CPU Cores
       6. 12 CPU Cores
       7. 14 CPU Cores
       8. 16 CPU Cores
       9. 24 CPU Cores
       Current CPU Cores    :12
       Selection[1 .. 9](default 24 CPU Cores)    : 6
       ODA base domain memory in GB(min 16, max 244)(Current Memory 192G)[default 192]      :
INFO: Using default memory size i.e. 192 GB
Additional vlan networks to be assigned to oda_base ? (y/n) [n]: y
Select the network to assign (BCKUPDB): BCKUPDB
Additional vlan networks to be assigned to oda_base ? (y/n) [n]:
Vlan network to be removed from oda_base ? (y/n) [n]: n
INFO: Node 0:Configured oda base pool
INFO: Node 1:Configured oda base pool
INFO: Node 0:ODA Base configured with new memory
INFO: Node 0:ODA Base configured with new vcpus
INFO: Node 0:New vlans added in ODA Base
INFO: Changes will be incorporated after the domain is restarted on Node 0.
INFO: Node 1:ODA Base configured with new memory
INFO: Node 1:ODA Base configured with new vcpus
INFO: Node 1:New vlans added in ODA Base
INFO: Changes will be incorporated after the domain is restarted on Node 1.
INFO: Updating /etc/sysctl.conf in oda_base domain with parameter "vm.nr_hugepages=51626"
```

4. Restart ODA_BASE (from dom0)

```
# oakcli restart oda_base
```

5. Start the Guest VMs

```
# oakcli start vm vm1
# oakcli start vm vm2
```

6. Configure the BCKUPDB network IP address for ODA_BASE

Once the VLAN has been added to ODA_BASE and restarted, a new VNIC will be available (in this example “eth3”), check with “`ifconfig -a`”. Create the related device configuration file such “*/etc/sysconfig/network-scripts/ifcfg-eth3*”. As we want a fixed IP we need to change it as following:

```
# Xen Virtual Ethernet
DEVICE=eth3
BOOTPROTO=none
ONBOOT=yes
IPADDR=<your ip>
NETMASK=<your netmask>
GATEWAY=<your gtw>
```

Then we can restart the network to get the changes:

```
# /etc/init.d/network restart
```

7. Configure the IPs address for the VLAN on your guest VMs (vm1/vm2)

Two new eth devices are now available on your guest VMs; you need to configure them creating two eth configuration files. If this is the VM guest first boot (supporting the OVMAPI), you could also use the “send keys” feature (see the whitepaper: Oracle Database Appliance: Automated Virtual Machine Provisioning)

```
/etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
BOOTPROTO=none
ONBOOT=yes
IPADDR=<your ip>
NETMASK=<your netmask>
GATEWAY=<your gtw>

/etc/sysconfig/network-scripts/ifcfg-eth1
DEVICE=eth1
BOOTPROTO=none
ONBOOT=yes
IPADDR=<your ip>
NETMASK=<your netmask>
GATEWAY=<your gtw>
```

Restart the network to make the changes

```
# /etc/init.d/network restart
```

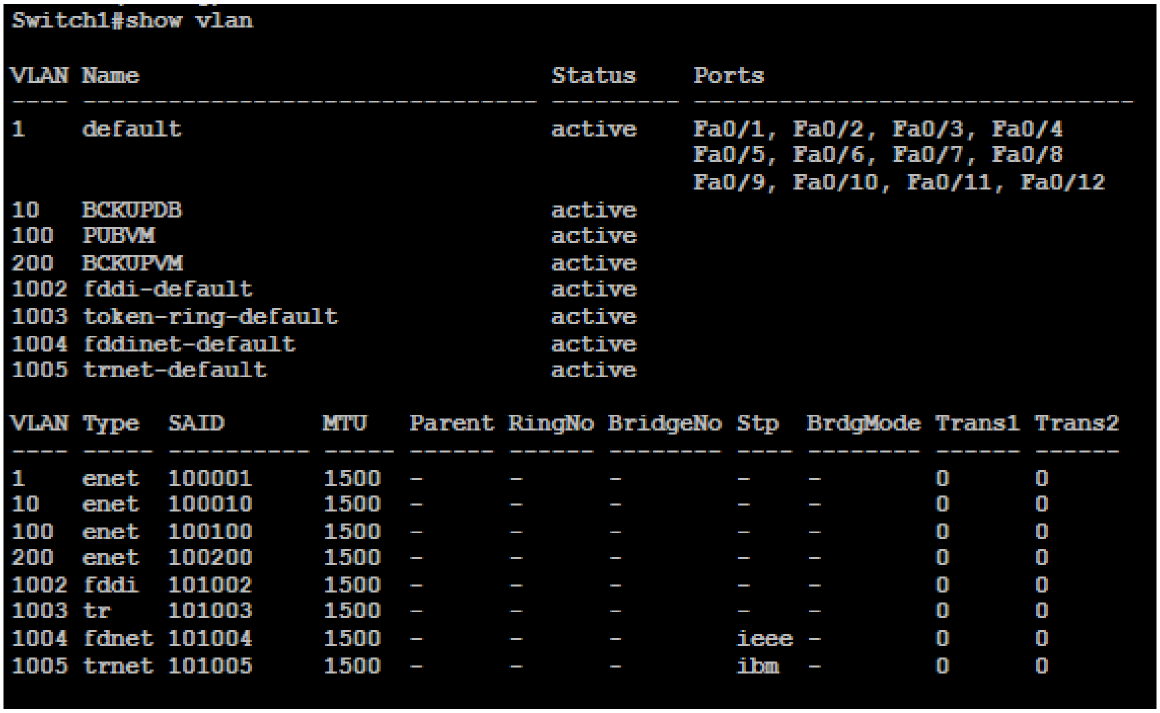
Network Switch Configuration Steps

In this paragraph, as example, we show which commands should be used on the network switch (Cisco Catalyst 2950 Switch) to configure the ports to use VLAN. Note as this is just an example and your switch vendor/version may differ significantly.

- Create the VLANs

```
configure terminal
vlan 10 name BCKUPDB
vlan 100 name PUBVM
vlan 200 name BCKUPVM
end
```

At this time the VLANs are created as following:

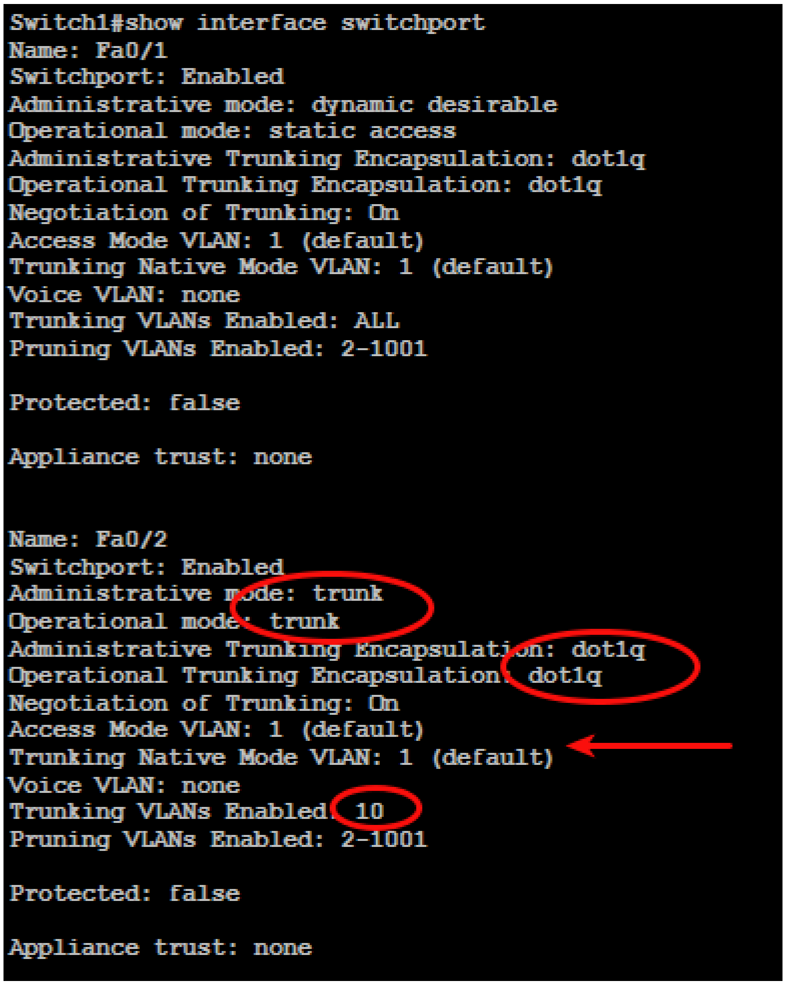


```
Switch1#show vlan

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                                Fa0/9, Fa0/10, Fa0/11, Fa0/12
10   BCKUPDB                          active
100  PUBVM                            active
200  BCKUPVM                          active
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
1    enet  100001     1500  -      -      -        -    -        0      0
10   enet  100010     1500  -      -      -        -    -        0      0
100  enet  100100     1500  -      -      -        -    -        0      0
200  enet  100200     1500  -      -      -        -    -        0      0
1002 fddi  101002     1500  -      -      -        -    -        0      0
1003 tr    101003     1500  -      -      -        -    -        0      0
1004 fdnet 101004     1500  -      -      -        ieee -        0      0
1005 trnet 101005     1500  -      -      -        ibm  -        0      0
```

- Configuring the Allowed VLAN10 for Trunking Ports 2/4

```
configure terminal
interface fastethernet 0/2
switchport trunk encapsulation do1q
switchport mode trunk
switchport trunk allowed vlan 10
interface fastethernet 0/4
switchport trunk encapsulation do1q
switchport mode trunk
switchport trunk allowed vlan 10
end
```

The show interface on the network switch will be as following (port0/2):

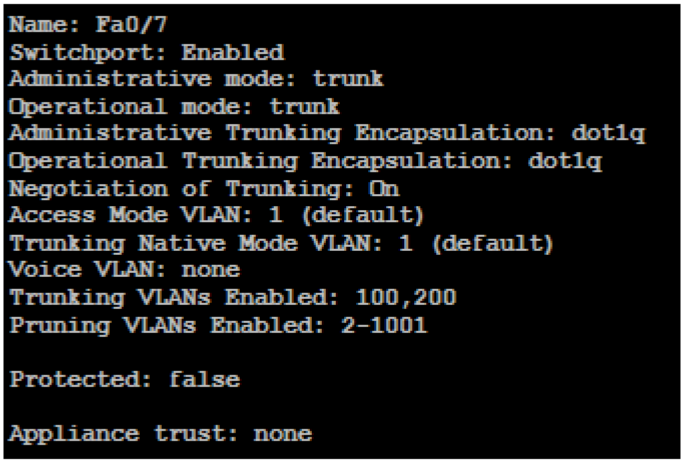


```
Switch1#show interface switchport
Name: Fa0/1
Switchport: Enabled
Administrative mode: dynamic desirable
Operational mode: static access
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Voice VLAN: none
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001

Protected: false

Appliance trust: none


Name: Fa0/2
Switchport: Enabled
Administrative mode: trunk
Operational mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Voice VLAN: none
Trunking VLANs Enabled: 10
Pruning VLANs Enabled: 2-1001

Protected: false

Appliance trust: none
```

- Configuring the Allowed VLAN100/200 for Trunking Ports 7/9

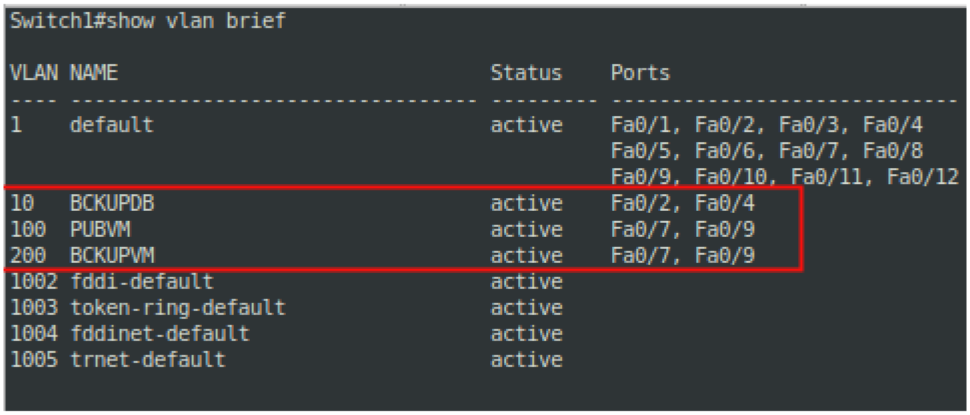
```
configure terminal
interface fastethernet 0/7
switchport trunk encapsulation do1q
switchport mode trunk
switchport trunk allowed vlan 100,200

interface fastethernet 0/9
switchport trunk encapsulation do1q
switchport mode trunk
switchport trunk allowed vlan 100,200
end
```

The show interface on the network switch will be as following (port0/7):



```
Name: Fa0/7
Switchport: Enabled
Administrative mode: trunk
Operational mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Voice VLAN: none
Trunking VLANs Enabled: 100,200
Pruning VLANs Enabled: 2-1001

Protected: false

Appliance trust: none
```

The “show vlan” command:



```
Switch1#show vlan brief

VLAN NAME                             Status    Ports
---- -------------------------------- --------- -----------------------------
1    default                          active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                                Fa0/9, Fa0/10, Fa0/11, Fa0/12
10   BCKUPDB                          active    Fa0/2, Fa0/4
100  PUBVM                            active    Fa0/7, Fa0/9
200  BCKUPVM                          active    Fa0/7, Fa0/9
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
```

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 Database Appliance: Virtual Local Area Network
March 2017
Author: Ruggero Citton
Contributing Authors: Sandeep Singla, RACPack Team