Copyright (c) 2018, Oracle. All rights reserved. Oracle Confidential.

Implementing Tagged VLAN Interfaces in Oracle VM Environments on Exadata (Doc ID 2018550.1)

In this Document

APPLIES TO:

Oracle Exadata Storage Server Software - Version 12.1.2.1.1 and later
Linux x86-64
This document is applicable to Oracle VM Environments on Exadata.

GOAL

Demonstrate the implementation of tagged VLAN interfaces in Oracle VM environments on Exadata.

SOLUTION

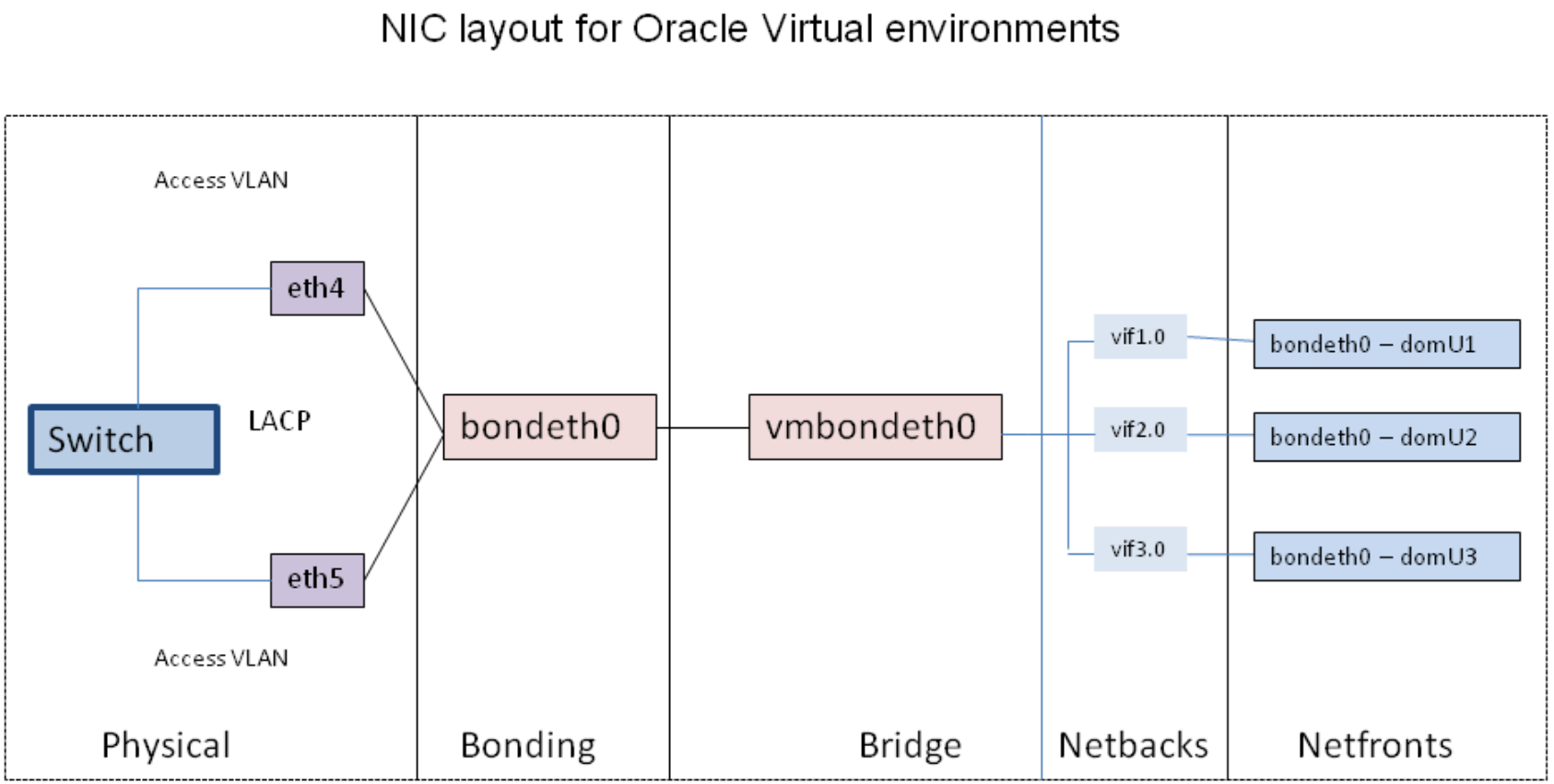
Background

Oracle databases running in Oracle VM guests on Oracle Exadata Database Machine are accessed through the client Ethernet network defined in the Oracle Exadata Deployment Assistant (OEDA) configuration tool. Client network configuration in both the management domain (dom0) and user domains (domUs) is done automatically when OEDA installation tool creates the first user domain during initial deployment.

The default bonded client network configuration, as depicted in figure 1 below, is as follows:

1. In the dom0, eth slave interfaces (e.g. eth1 and eth2, or eth4 and eth5) that allow access to the domU client network defined in OEDA are discovered, configured, and brought up, but no IP is assigned
2. In the dom0, bondeth0 master interface is configured and brought up, but no IP is assigned
3. In the dom0, bridge interface vmbondeth0 is configured, but no IP is assigned
4. In the dom0, one virtual backend interface (vif) per domU that maps to that particular domU's bondeth0 interface is configured and brought up, but no IP is assigned. These vif(s) are configured on top of the bridge interface vmbondeth0 and the mapping between the dom0 vif interface and its corresponding user domain interface bondeth0 is defined in the user domain configuration file called vm.cfg, which can be found in /EXAVMIMAGES/GuestImages/<user domain name>.

Figure 1

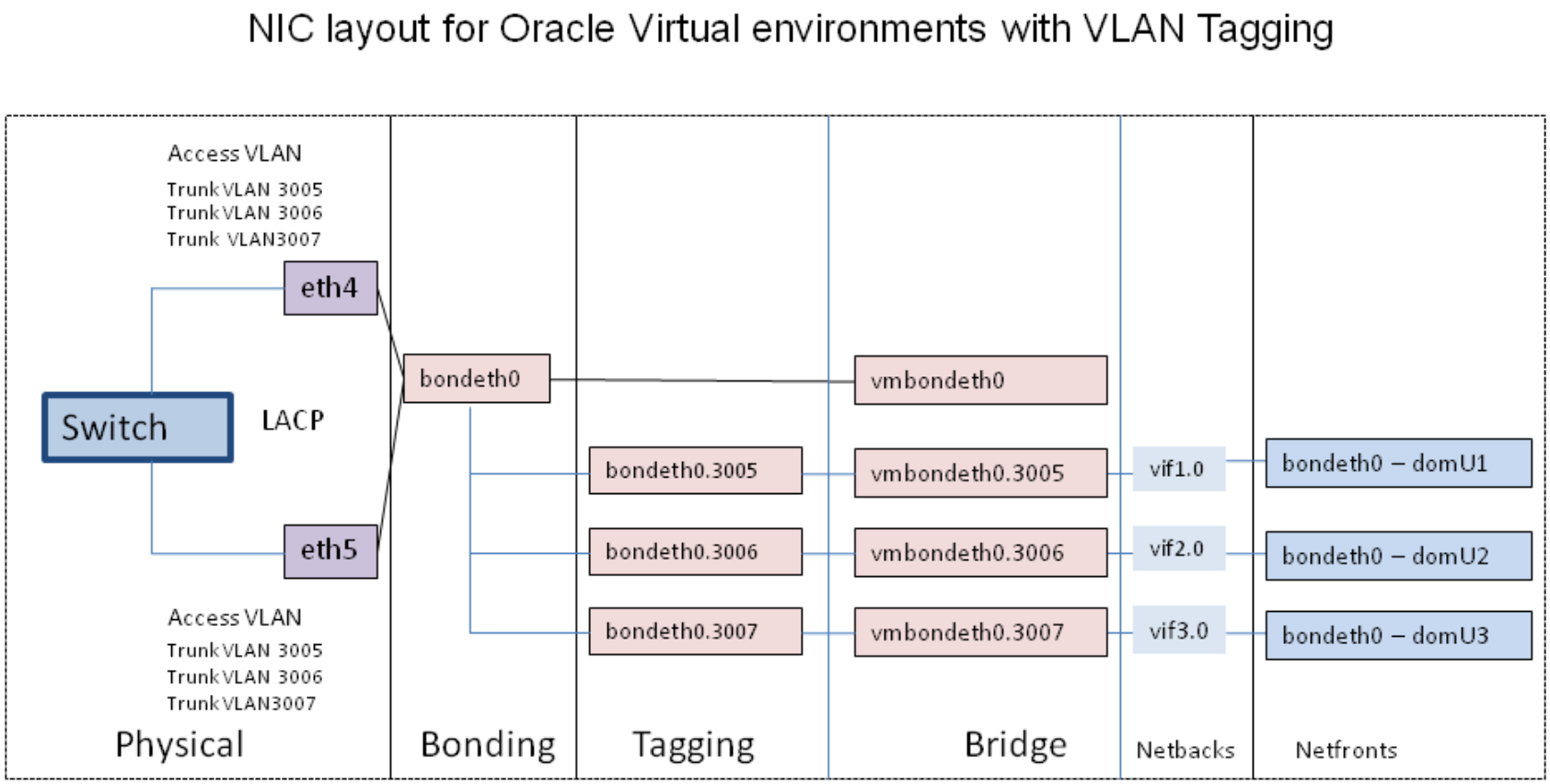


Under standard and default installations, there is a single bondeth0 and a corresponding vmbondeth0 bridge interface configured in the dom0 as described above. This bondeth0 interface is based on the "default Access Virtual Local Area Network" (Access VLAN) which the ports on the switch used by the slave interfaces making up bondeth0, is configured for.

If there is a need for virtual deployments on Exadata to access additional VLANs on the client network, such as enabling network isolation across user domains, then 802.1Q based VLAN tagging is a solution. This document provides detailed instructions on how to configure and use such additional VLAN tagged interfaces on the client network. It is a requirement that the Access VLAN stay working and configured before and after these instructions are followed. At no time is the Access VLAN to be disabled.

After executing the procedure described in this document, the client network configuration will be as depicted in figure 2 below.

Figure 2



Overview

The configuration process is divided into two phases.

Phase 1 - In the management domain (dom0) VLAN tagged interfaces and the corresponding bridge interfaces are created at the Linux operating system level with persistent configurations. The respective default gateway IP addresses are configured via iproute2 using unique tables.

Phase 2 - The newly created bridge interfaces are passed on to the user domains to be used by their client network interface.

Requirements

The following requirements apply to the procedures described in this document

- vconfig rpm installed on the Oracle Virtual Servers (dom0)
- DNS names and IPs have been allocated for each of the tagged VLANs
- An OVM deployment on Exadata has already been completed
- All tagged VLAN interfaces can only be created in Dom0 and mapped to virtual interfaces in the DomU's. Tagged VLAN interfaces can not be created in the DomU's.
- These procedures and changes only affect database servers in Exadata Database Machine. No changes are permitted on storage cells
- Peer Ethernet switches have been already configured to allow host-based tagged VLANs using the ports by the client network slave interfaces (e.g. eth4 and eth5)
- The Access VLAN must remain working and fully configured before and after these instructions are followed.
- A full understanding of VLANs

Note : Starting with Exadata Software Image version 12.1.2.3.0 and when using OEDA version 16.049.11 (February 2016 release) or above, tagged VLAN interfaces can be implemented at deployment time with OEDA.

The procedure described in this document is applicable for implementing tagged VLAN interfaces on an already deployed OVM system that was deployed with an Exadata release prior to 12.1.2.3.0.

The steps described in this note at a high level are:

Phase 1

1. Validate necessary packages are installed
2. Create the tagged VLAN interfaces and bridges

Phase 2

1. Backup user domain configuration file vm.cfg in dom0
2. Generate new random MAC address in dom0
3. Modify user domain configuration file vm.cfg in dom0
4. Modify udev rules in dom0
5. Modify bondeth0 configuration in domU
6. Verify bondeth0 configuration in domU
7. Modify Grid Infrastructure configuration in domU
8. Restart domU
9. Verify Oracle Clusterware started and databases are accessible

Note - The process involves minimal downtime. The only downtime is experienced in step 8 of Phase 2 where user domains are restarted.

Phase 1 - Creating the Tagged VLAN Interfaces in the Management Domain (dom0)

The steps described in Phase 1 are executed one time in dom0 to configure all tagged VLAN interfaces that will be used for all user domains. User domain-specific configuration occurs in Phase 2.

Step 1.1. Ensure the vconfig package is installed.

The vconfig-1.9-3.x86_64.rpm is available from:

http://public-yum.oracle.com/repo/OracleVM/OVM3/latest/x86_64/

Install the vconfig rpm:

```
# rpm -Uvh vconfig-1.9-3.x86_64.rpm
Preparing...\t  ########################################### [100%]
   1:vconfig\t  ########################################### [100%]

# rpm -qa | grep vconfig
vconfig-1.9-3

# which vconfig
/sbin/vconfig
```

Step 1.2. Use attached script VTagXBrConfig.sh to create and bring up the tagged VLAN interfaces

Attached to this document is script VTagXBrConfig.sh. Download VTagXBrConfig.sh to dom0 and run it to create and bring up all the tagged VLAN interfaces in dom0.

```
Usage : VTagXBrConfig.sh  [ help | man | -h ]

OR

# ./VTagXBrConfig.sh  create -v <Start VLAN ID> -l <End VLAN ID> -i <Interface> "
```

Example :

```
# ./VTagXBrConfig.sh create -v 3005 -l 3034 -i bondeth0
```

Sample output of "ip link show" command after tagged VLAN interfaces have been created with relevant lines highlighted

Note: The VLANs have the ".xxxx" at the end of their names.

```
# ip link show

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN

    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

2: eth4: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc mq master bondeth0 state UP qlen 1000

    link/ether 90:e2:ba:81:34:34 brd ff:ff:ff:ff:ff:ff
```

```
3: eth5: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc mq master bondeth0 state UP qlen 1000
    link/ether 90:e2:ba:81:34:34 brd ff:ff:ff:ff:ff:ff
……
30: bondeth0.3012@bondeth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master vmbondeth0.3012 state UP
    link/ether 90:e2:ba:81:34:34 brd ff:ff:ff:ff:ff:ff
31: vmbondeth0.3012: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
    link/ether 90:e2:ba:81:34:34 brd ff:ff:ff:ff:ff:ff
32: bondeth0.3013@bondeth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master vmbondeth0.3013 state UP
    link/ether 90:e2:ba:81:34:34 brd ff:ff:ff:ff:ff:ff
…….
72: bondeth0.3033@bondeth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master vmbondeth0.3033 state UP
    link/ether 90:e2:ba:81:34:34 brd ff:ff:ff:ff:ff:ff
73: vmbondeth0.3033: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
    link/ether 90:e2:ba:81:34:34 brd ff:ff:ff:ff:ff:ff
74: bondeth0.3034@bondeth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master vmbondeth0.3034 state UP
    link/ether 90:e2:ba:81:34:34 brd ff:ff:ff:ff:ff:ff
75: vmbondeth0.3034: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
    link/ether 90:e2:ba:81:34:34 brd ff:ff:ff:ff:ff:ff
77: vif2.0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master vmbondeth0 state UP qlen 32
    link/ether fe:ff:ff:ff:ff:ff brd ff:ff:ff:ff:ff:ff
```

Sample output of "brctl show" after the Bridges have been created with relevant lines highlighted

```
# brctl show
bridge name             bridge id               STP enabled     interfaces
vmbondeth0              8000.90e2ba813434       no              bondeth0
……
vmbondeth0.3005         8000.90e2ba813434       no              bondeth0.3005
vmbondeth0.3006         8000.90e2ba813434       no              bondeth0.3006
……..
vmbondeth0.3033         8000.90e2ba813434       no              bondeth0.3033
```

```
vmbondeth0.3034         8000.90e2ba813434       no              bondeth0.3034
vmeth0                  8000.0010e062c00e       no              eth0
.........................
```

At this point, all the tagged VLAN interfaces and bridges have been setup and are ready to be passed on to the user domains for use.

Please refer to Appendix 2 for more details on the conventions and requirements for creating the tagged VLAN interfaces.

Phase 2 - Configure user domains to use the VLAN tagged interfaces

During this phase the user domain(s) are configured to use the tagged VLAN interfaces and bridges created in Phase 1 of this exercise.

NOTE: **All** steps in Phase 2 must be run for each user domain that will be configured to use a VLAN tagged interface created in Phase 1.

Steps 2.1-2.3 are executed in the management domain (dom0).

Step 2.1. In dom0 backup existing vm.cfg

The bridge interface used for a particular virtual interface (vif) inside a user domain (domU) is defined in the file /EXAVMIMAGES/GuestImages/<user domain name>/vm.cfg located in the dom0. Backup the existing user domain configuration file vm.cfg to vm.cfg.bak.

```
# cd /EXAVMIMAGES/GuestImages/<user domain name>/vm.cfg
# cp vm.cfg vm.cfg.before_vlantag
```

Step 2.2. In dom0 generate a new random MAC address

Generate a new random MAC address using the command below for the domUs interface which will be using a newly created bridge.

```
# echo -n "00:16:3e"; dd bs=1 count=3 if=/dev/random 2>/dev/null | hexdump -v -e '/1 ":%02x"'
00:16:3e:e6:28:88
```

Step 2.3. In dom0 modify vm.cfg

Modify the netfront bridge vmbondeth0 in the vif line to use a newly created tagged interface and the newly generated MAC address. The example below uses tagged interface 'vmbondeth0.3005'.

```
vif = ['type=netfront,mac=00:16:3e:92:11:57,bridge=vmbondeth0','type=netfront,mac=00:16:3e:e6:94:28,bridge=vmeth0']

to

vif = ['type=netfront,mac=00:16:3e:e6:28:88,bridge=vmbondeth0.3005','type=netfront,mac=00:16:3e:e6:94:28,bridge=vmeth0']
```

Step 2.4-2.8 are executed in the user domain (domU).

Step 2.4. In domU add information about the new MAC address in /etc/udev/rules.d/70-persistent-net.rules.

Change the line with bondeth0 to have the newly generated MAC address as below :

```
SUBSYSTEM=="net", ACTION=="add", SYSFS{address}=="00:16:3e:92:11:57", KERNEL=="eth*", NAME="bondeth0"

to

SUBSYSTEM=="net", ACTION=="add", SYSFS{address}=="00:16:3e:e6:28:88", KERNEL=="eth*", NAME="bondeth0"
```

Step 2.5. In domU modify the IP address and netmask of the bondeth0 interface

Modify the IP address and netmask of the bondeth0 interface of the user domain using ipconf.pl to match tagged VLAN network using ipconf as shown in the example below. The prompts need to answered appropriately as shown.

```
# /opt/oracle.cellos/ipconf.pl -nocodes
[Info]: ipconf command line: /opt/oracle.cellos/ipconf.pl -nocodes
Logging started to /var/log/cellos/ipconf.log
Interface ib0   is                      Linked.    hca: mlx4_0
Interface ib1   is                      Linked.    hca: mlx4_0
Interface eth0  is                      Linked.    driver/mac: vif/00:16:3e:32:79:f1
Interface bondeth0 is                      Linked.    driver/mac: vif/00:16:3e:f0:be:3b


The current nameserver(s): 130.35.249.41 130.35.249.52 192.135.82.132
Do you want to change it (y/n) [n]: n
The current timezone: America/Los_Angeles
Do you want to change it (y/n) [n]: n
The current NTP server(s): 144.25.255.140 144.25.255.141
```

```
Do you want to change it (y/n) [n]: n


Network interfaces
Name     State      Status  IP address       Netmask         Gateway         Net type     Hostname
ib0      Linked     UP      192.168.12.155   255.255.248.0                   Private      host1vm01-priv1.us.oracle.com
ib1      Linked     UP      192.168.12.156   255.255.248.0                   Private      host1vm01-priv2.us.oracle.com
eth0     Linked     UP      10.133.13.156    255.255.248.0   10.133.8.1      Management   host1vm01.us.oracle.com
bondeth0 eth4,eth5  UP      10.133.27.115    255.255.240.0   10.133.16.1     SCAN         host2vm01.us.oracle.com
Select interface name to configure or press Enter to continue: bondeth0
Selected interface. bondeth0
Select connection mode for bondeth0 interface from the list below
1: active-backup
2: lacp
Connection mode: 1
IP address or none [10.133.27.115]: 10.133.28.55
Netmask [255.255.240.0]:
Gateway (IP address or none) or none [10.133.16.1]:

Select network type for interface from the list below
1: Management
2: SCAN
3: Other
Network type [2]:
Fully qualified hostname or none [scao04client02vm01.us.oracle.com]:
Continue configuring or re-configuring interfaces? (y/n) [y]:

Network interfaces
Name     State      Status  IP address       Netmask         Gateway         Net type     Hostname
ib0      Linked     UP      192.168.12.155   255.255.248.0                   Private      host1-priv1.us.oracle.com
ib1      Linked     UP      192.168.12.156   255.255.248.0                   Private      host1-priv2.us.oracle.com
eth0     Linked     UP      10.133.13.156    255.255.248.0   10.133.8.1      Management   host1vm01.us.oracle.com
bondeth0 eth4,eth5  UP      10.133.28.55     255.255.240.0   10.133.16.1     SCAN         host2vm01.us.oracle.com
```

```
Select interface name to configure or press Enter to continue:


Select canonical hostname from the list below

1: host1vm01-priv1.us.oracle.com

2: host1vm01-priv2.us.oracle.com

3: host1vm01.us.oracle.com

4: host2vm01.us.oracle.com

Canonical fully qualified domain name [3]:


Select default gateway interface from the list below

1: eth0

2: bondeth0

Default gateway interface [2]:


Canonical hostname: scao04adm02vm01.us.oracle.com

Nameservers: 130.35.249.41 130.35.249.52 192.135.82.132

Timezone: America/Los_Angeles

NTP servers: 144.25.255.140 144.25.255.141

Default gateway device: bondeth0

Network interfaces

Name     State      Status  IP address       Netmask         Gateway         Net type     Hostname

ib0      Linked     UP      192.168.12.155   255.255.248.0                   Private      host1vm01-priv1.us.oracle.com

ib1      Linked     UP      192.168.12.156   255.255.248.0                   Private      host1vm01-priv2.us.oracle.com

eth0     Linked     UP      10.133.13.156    255.255.248.0   10.133.8.1      Management   host1vm01.us.oracle.com

bondeth0 eth4,eth5  UP      10.133.28.55     255.255.240.0   10.133.16.1     SCAN         host2vm01.us.oracle.com

Is this correct (y/n) [y]:


[Info]: Custom changes have been detected in /etc/sysconfig/network-scripts/ifcfg-bondeth0

[Warning]: Config /etc/sysconfig/network-scripts/ifcfg-bondeth0. Existing line "IPADDR=10.133.27.115" will be overwritten with
"IPADDR=10.133.28.55"

[Info]: Config /etc/sysconfig/network-scripts/ifcfg-bondeth0. Original line "IPADDR=10.133.27.115" will be commented

[Info]: Original file /etc/sysconfig/network-scripts/ifcfg-bondeth0 will be saved in /opt/oracle.cellos/conf/network-
scripts/backup_by_Exadata_ipconf
```

```
[Info]: Original file /etc/ssh/sshd_config will be saved in /etc/ssh/sshd_config.backupbyExadata

[Info]: Generate /etc/ssh/sshd_config with ListenAddress(es) 10.133.13.156, 10.133.28.55, 192.168.12.155, 192.168.12.156

Stopping sshd:                                             [  OK  ]

Starting sshd:                                             [  OK  ]

[Info]: Save /etc/sysctl.conf in /etc/sysctl.conf.backupbyExadata

[Info]: Adjust settings for IB interfaces in /etc/sysctl.conf

Re-login using new IP address 10.133.13.156 if you were disconnected after following commands

ip link set bondeth0 down

sleep 1

ifdown bondeth0

sleep 1

/opt/oracle.cellos/reset_rule_table.sh 210

sleep 1

ifup bondeth0

sleep 4

ip link set bondeth0 up

sleep 4

Cannot change large-receive-offload

Determining if ip address 10.133.28.55 is already in use for device bondeth0...

Determining if ip address 10.133.28.55 is already in use for device bondeth0...
```

Step 2.6. In the domU modify the HWADDR in /etc/sysconfig/network-scripts/ifcfg-bondeth0 to match the new MAC address generated in step 2.2 (If HWADDR is present in the /etc/sysconfig/network-scripts/ifcfg-bondeth0)

Edit /etc/sysconfig/network-scripts/ifcfg-bondeth0 and replace the existing HWADDR with the new one generated in step 2.2

Step 2.7. In the domU verify the new IP address and netmask

Verify the new IP address, the MAC and the netmask settings for bondeth0 got changed to the new IP address and netmask.

```
# ip addr show bondeth0
2: bondeth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
   link/ether 00:16:3e:e6:28:88 brd ff:ff:ff:ff:ff:ff
   inet 10.133.28.55/20 brd 10.133.31.255 scope global bondeth0
```

```
   inet6 fe80::216:3eff:fef0:be3b/64 scope link
   valid_lft forever preferred_lft forever
```

Step 2.8. In domU modify the Grid Infrastructure configuration to use the new tagged VLAN interfaces

a. Get the network number on which the VIP and the SCAN are defined using:

- ```
  $ $GRID_HOME/bin/srvctl config scan|grep "Network:"

  $ $GRID_HOME/bin/srvctl config vip -node <cluster node name>|grep "network number"
  ```

Both the above commands should return the same network number

b. Modify this network to use the new IP/subnet:

- ```
  # $GRID_HOME/bin/srvctl modify network -netnum <network number obtained above>  -subnet <new subnet>/<new netmask>/bondeth0
  ```

c. Change the VIP to use the new IP/subnet for each cluster node:

- ```
  $ $GRID_HOME/bin/srvctl modify vip -node < cluster node name > -address <new IP>/<new subnet>/bondeth0
  ```

d. Change the SCAN configuration:

Modify the DNS entry for the existing "SCAN name" to use the new network.

SCAN name can be found using:

```
$ $GRID_HOME/bin/srvctl config scan|grep "SCAN name"
```

OR

Create a new DNS entry for a new "SCAN name" using the new tagged VLAN network and modify the Clusterware configuration to use the new SCAN name using:

```
$ $GRID_HOME/bin/srvctl modify scan -scanname <new SCAN name> -netnum <network number obtained in step a above>
```

Step 2.9. In dom0 restart the user domain

Login to the dom0 as root and run the xm command to shutdown and restart the user domain

```
# xm shutdown <user-domain name> -w
# xm create /EXAVMIMAGES/GuestImages/<user domain name>/vm.cfg
```
```

Step 2.10. In domU verify Oracle Clusterware started and databases are accessible

How to Create New User Domains After Tagged VLAN Interfaces Have Been Configured

Once the environment has been configured to use tagged VLAN interfaces wherein the user domains' client network interface (bondeth0) is based on a tagged VLAN interface, creation of new guest domain(s) should follow the steps below to have the new user domain(s) use a tagged VLAN interface just like existing user domains.

a) New guest domain(s), at the time of their creation, will use the default non-tagged interface (bondeth0 of the dom0) through the bridge (vmbondeth0 of the dom0) for their client network. Hence this will be no different from creating a new user domain on an environment which does not have VLAN tagging involved. This means the default non-tagged interface (bondeth0) along with the bridge interface vmbondeth0 should be in place in the management domain (dom0) along with the default access vlan configuration in the switch to which the slave interfaces (e.g eth4 and eth5) are connected to, during the time of the new user domain creation. This also will be no different from the configuration of an environment which does not have VLAN tagging involved.

b) Modify the network configuration of the newly created guest domain(s) to have the client network use a tagged VLAN interface as described in phase 2 of this document. The tagged VLAN interface to be used here should already be in place since this is a procedure to be used after the tagged VLAN interfaces have been configured.

Appendix 1

Overview of VLAN

An Ethernet switch can have all of its ports not a member of a VLAN, but is a stand-alone Local Area Network (LAN). These types of switches are commonly found in the home or small office and sometimes are called Workgroup switches. If the switch becomes a "managed" switch it has many more Ethernet features available to it (and and the corresponding configuration needed). One of these features is the Virtual Local Area Network (VLAN) which allows this physical switch to be split into several virtual logical switches. Why is this important? This feature in implemented for a variety of reasons, but the more important one is to logically isolate network traffic so one host who is a member of one VLAN is unable to see the network traffic from any other VLAN. The details on how this is done, is not covered in this paper. The oversimplified description is VLANing allows a switch to keep traffic separate at the Data Link Layer (layer 2 in the Open Systems Interconnect (OSI)). To get network traffic to traverse from one VLAN to another, it has to be done at the Network Layer (layer 3 in OSI) by using a router. So a switch can be configured so that certain port that NIC's are plugged into are members of one VLAN while others are members of other VLAN's. This is a simple configuration that allows a physical switch to be logically separated into different LAN's allowing the switch to be more fully utilized. When a host is plugged into a port that is a member of a single VLAN, it does not need to be made aware of the VLAN it is on. This is called an "Access" VLAN configuration which does not need any configuration on the host.

There are certain times where a single host needs to be a member of various VLAN's. In the not-to-distant past this was accomplished by putting one NIC per VLAN into a host. This became difficult due to physical space for hardware in a host. To adapt to this constraint, the NIC drivers were modified so a single NIC could become a member of various VLAN's. Both the Operating System (OS) and switches were changed to handle this. The method used is called "VLAN Tagging" or "VLAN Trunking" (they are the same thing). Though it is possible to tag an interface for a single VLAN, that typically is not needed and just makes more work for both people and OS.

So if a host uses the regular "Access" VLAN it requires no additional configuration as the switch handles adding the VLAN ID to the Ethernet packet. If a host requires access to multiple VLAN's over the same NIC, a maximum of 4094 (upper limit is the upper limit of VLAN ID's) of "Tagged" VLAN's can be configured.

The Xen Networking can make use of any NIC device available to the Hypervisor. It can also make use of Tagged NIC's and make these presentable to guest domains (DomU). It does this through the use of a software "bridge". This bridge accomplishes nearly the same task as a physical Ethernet bridge in that it bridges the physical NIC to any number of virtual

ones via layer 2. The NIC then is able to transfer network packets from these virtual NIC's to the real network at a layer 2 level (bridging is routing but at layer 2). The bridge is not tied to a physical NIC, but to the NIC device so a Tagged NIC can also be used to make virtual NIC's available to the guest domains(domUs).

Appendix 2

VLAN Tagged Ethernet NIC configuration in the dom0

The various scripts to bring up an interface or bridge use a specific file naming convention. NIC, Tagged VLAN, and bridge configuration files always begin with "ifcfg-*". VLAN interface file names have a file extension which is the VLAN ID. For instance, if the bondeth0 interface is to belong to VLAN 3005, so the configuration file name would be "ifcfg-bondeth0.3005". It has the following as its contents:

cat /etc/sysconfig/network-scripts/ifcfg-bondeth0.3005

DEVICE=bondeth0.3005

TYPE=Ethernet

BOOTPROTO=static

ONBOOT=yes

VLAN=yes

BRIDGE=vmbondeth0.3005

The VLAN configuration file has the parameter VLAN set to "yes". The ifup script will parse the configuration file name to get the VLAN ID from the file name extension and use this number to configure the VLAN using the vconfig package and further configure it using the "ip link add dev" command. This file also has the BRIDGE parameter set, which makes the ifup-eth script add it to the specific bridge using brctl.

Bridged Ethernet NIC configuration in the dom0

If the "ifcfg-" file contains the parameter TYPE set to "Bridge" the ifup-eth script creates the bridge using the brctl command. The bridge configuration file looks like:

cat /etc/sysconfig/network-scripts/ifcfg-vmbondeth0.3005

DEVICE=vmbondeth0.3005

TYPE=Bridge

BOOTPROTO=static

ONBOOT=yes

DELAY=0

The /etc/rc3.d/S11exavmnetwork script enforces the names of the bridges and their base interface members. The start-up script expects the bridge configuration files to start with "ifcfg-vm" with the rest of the file name being the interface associated with that bridge. It parses the file name of the bridge and makes sure the corresponding device exists. Because the base device is a Tagged interface, the bridge configuration file must also follow the same naming convention as the Tagged interface. This does not cause a problem as a Tagged interface contains the parameter VLAN=yes and TYPE=Ethernet while a bridge file has TYPE=Bridge and VLAN is not set.

Didn't find what you are looking for?