Configure Pair Bonding, VLANs, and Bridges for KVM Hypervisor

Configure Pair Bonding, VLANs, and Bridges for KVM Hypervisor

Why do all the work for tagged VLAN's

Red Hat does not publish a network GUI to configure pair bonding, KVM bridges, or full VLAN interaction for their OS. These have to be done manuallys, even though they've been built into Red Hat's available scripting and options for years. They're used to provide these features:

- Pair bonding provides high availability network connections, very useful for KVM servers.
- KVM bridges are **necessary** for the KVM guest operating systems to have their network exposed to the same network VLAN as the KVM server uses. Otherwise, the guest is behind a NAT or otherwise isolated from the rest of the network.

Tagged VLANs allow individual network ports to exist on multiple VLAN's, and should not be used without careful planning and review.

- A front facing VLAN is especially useful for locally exposed or "public" IP addresses such as web servers might need.
- A private VLAN with no gateway is especially useful for internal traffic such as NFS or MySQL. It helps make firewall configurations and access control much easier and safer.

There are potential ussues with **Tagged VLANs**.

- Tagged VLANs are **always** more awkward to set up
- Tagged VLANs require the network ports on the network switch to be manually configured to use tags, which makes them incompatible with untagged devices.
- Red Hat based systems with tagged VLAN's cannot use **kickstart or PXE.**
- Red Hat based systems cannot do **YUM updates** until the network is manually configured from a console.
  - The default kernel for RHEL 6.3 does not support pair bonding, tagging, and KVM bridging all at the same time.
  - RHEL or CentOS 6.3 systems must have the kernel updated by other means before they can use all these features together.

If you don't need the tagged VLANs, see: Configure Pair Bonding and Bridges for KVM Hypervisor

Get the network configurations

- Ensure that the upstream switches are configured for pair bonding and tagged VLAN's.
  - Get the information about the VLAN's, especially their names, relevant gateways, and net masks
- For this example, assume that the VLAN's are called "**100"** and "**101**" on the switches.
  - The bonded VLAN's will then called "**bond0.100**" and "**bond0.101"**, respectively.
  - The KVM bridges will be called "**br100**" and "**br101**", respectively.
- Typically, only one gateway on one VLAN is configured.
  - Setting gateways on multiple VLANs can create enormous routing confusion.
- Get the hostname and other network configurations for VLAN "**100**" for the new server, especially:
  - Fully qualified Hostname
    - This is tyically the "hostname" used for the host, itself.
  - IP address
  - Netmask
  - Gateway
  - DNS servers
- Get the hostname and other network configurations for VLAN "**101"** for the new server, especially:
  - Fully qualified Hostname
    - This ***should not match*** the hostname on VLAN "**100**", to allow separate monitoring and controlled access.
  - IP address
  - Netmask
  - Gateway (Secondary VLAN's do not normally have a gateway)
  - DNS servers
    - Normally matches VLAN **100** DNS server.
    - If they do not match, select a consistent set of DNS servers manually.
    - One DNS server on each VLAN is workable if and only if they provide identical information.

Configure the fully qualified hostname

- Edit /etc/sysconfig/network to contain the information below.
  - The **fully qualified hostname**, in particular, is the primary hostname desired from the network configurations above.

```
HOSTNAME=hostname.example.com
NETWORKING=yes
# IPv4
NETWORKING=yes
NOZEROCONF=yes
# IPv6, necessary for bonding
NETWORKING_IPV6=yes
IPV6INIT=yes
```

- Note that IPv6 **must, must, must** be enabled for bonding and other features to work privately.
  - NOZEROCONF=yes avoids the unnecessary "169.254.*" fallback addresses and routing from being activated, simply to prevent network confusion.

Configure pair bonding on bond0

Activate the pair bonding kernel module

- Put this in the file /etc/modprobe.d/bond0.conf

```
alias bond0 bonding
```

Create bond0 slaves for individual network ports.

Note: These directions assume that the physical network ports are reported as **eth0** and **eth1**. Adjust to em0 and em1 or other devices as needed.

- Edit /etc/sysconfig/network-scripts/ifcfg-eth0.
  - Ensure that HWADDR matches the actual MAC address of the physical port.
    - Run "/sbin/ifconfig eth0" to report this if necessary.
  - Notice the necessary bonding settings for MASTER and SLAVE.
  - Notice the MTU, for jumbo frames.
  - Notice the NM_CONTROLLED=no, to block NetworkManager from touching this port.

```
BOOTPROTO=none
DEVICE=eth0
HWADDR=aa:bb:cc:dd:ee:ff
MASTER=bond0
## Optional, use for jumbo frames if needed
#MTU=9000
NM_CONTROLLED=no
NOZEROCONF=yes
ONBOOT=yes
SLAVE=yes
TYPE=Ethernet
```

- Edit /etc/sysconfig/network-scripts/ifcfg-eth1.
  - Adjust the HWADDR to match the actual MAC address of the port.
    - Run "/sbin/ifconfig eth1" to report this.
  - Notice the bonding settings for MASTER and SLAVE.
  - Notice the MTU, for jumbo frames.
  - Notice the NM_CONTROLLED=no, to block NetworkManager from touching this port.

```
BOOTPROTO=none
DEVICE=eth1
HWADDR=aa:bb:cc:dd:ee:ff
MASTER=bond0
## Optional, use for jumbo frames if needed
#MTU=9000
NM_CONTROLLED=no
NOZEROCONF=yes
ONBOOT=yes
SLAVE=yes
TYPE=Ethernet
```

Other network devices, such as "eth3", "eth4", or "em0" or :em2, can also be added as needed .

Configure bond0 itself

- Edit /etc/sysconfig/network-scripts/ifcfg-bond0.
    - Notice the BONDING_OPTS, suitable for fail-over bonding and 100 msec failovers.

```
# Use tagged VLANs
ARP=no
BOOTPROTO=none
DEVICE=bond0
IPV6INIT=no
## Optional, use for jumbo frames if needed
#MTU=9000
NM_CONTROLLED=no
NOZEROCONF=yes
ONBOOT=yes
BONDING_OPTS="mode=1 miimon=100"
```

Configure VLANs on bond0

- Create /etc/sysconfig/network-scripts/ifcfg-bond0.100.
    - This selects the appropriate bridge for KVM based virtualization.

```
DEVICE=bond0.100
ONBOOT=yes
TYPE=Ethernet
BOOTPROTO=static
VLAN=yes
BRIDGE=br00
```

- Create /etc/sysconfig/network-scripts/ifcfg-bond0.101.
    - This selects the appropriate bridge for KVM based virtualization.

```
DEVICE=bond0.101
ONBOOT=yes
TYPE=Ethernet
BOOTPROTO=static
VLAN=yes
BRIDGE=br101
```

Configure bridges on bond0

- Create /etc/sysconfig/network-scripts/ifcfg-br100
    - Note the SLAVE=bond0.100, which ties it to the other part of the bridge. This is confusing, but necessary.
    - Note the TYPE=Bridge: this is what makes it accessible to KVM guests, and it's case sensitive.

- Note the VLAN=yes, which is what makes the connections "trunked".
- Note that IPv6 is available, but not currently in use.

```
DEVICE=br100
ONBOOT=yes
SLAVE=bond0.100
TYPE=Bridge
VLAN=yes
# IPv4
GATEWAY=172.16.0.1
IPADDR=172.16.0.2
NETMASK=255.255.255.0
NM_CONTROLLED=no
NOZEROCONF=yes
## Optional, use for jumbo frames if needed
#MTU=9000
# IPv6
#IPV6ADDR=
#IPV6ADDR_SECONDARIES=
#IPV6_AUTOCONF=no
#IPV6_MTU=9000
```

- Create /etc/sysconfig/network-scripts/ifcfg-br101
  - Note the SLAVE=bond0.101, which ties it to the other part of the bridge. This is confusing, but necessary.
  - Note the TYPE=Bridge: this is what makes it accessible to KVM guests, and it's case sensitive.
  - Note the VLAN=yes, which is what makes the connections "trunked".
  - Note that IPv6 is available, but not currently in use.

```
DEVICE=br101
ONBOOT=yes
SLAVE=bond0.101
TYPE=Bridge
VLAN=yes
# IPv4
# Usually only one gateway is configured
#GATEWAY=172.16.1.1
IPADDR=172.16.1.2
NETMASK=255.255.255.0
NM_CONTROLLED=no
NOZEROCONF=yes
## Optional, use for jumbo frames if needed
#MTU=9000
# IPv6
#IPV6ADDR=
#IPV6ADDR_SECONDARIES=
#IPV6_AUTOCONF=no
#IPV6_MTU=9000
```

Reboot

Fully activating, and testing, the new network configuration requires a reboot.