Pass through vlan tags

To pass through all VLAN tags from your VMs to the physical switch via the physical interfaces, you need to set up the interfaces and the Open vSwitch bridge to handle VLAN trunking. Here's how you can achieve this:

1. **Ensure Open vSwitch is installed and configured on both hosts**:

   ```
   sudo apt-get install openvswitch-switch
   ```

2. **Create an Open vSwitch bridge on both hosts**:

   On **Host A**:

   ```
   sudo ovs-vsctl add-br ovsbr0
   sudo ovs-vsctl add-port ovsbr0 bond0
   ```

   On **Host B**:

   ```
   sudo ovs-vsctl add-br ovsbr0
   sudo ovs-vsctl add-port ovsbr0 eno1
   ```

3. **Configure the bridge to trunk all VLANs**:

   On **Host A**:

   ```
   sudo ovs-vsctl set port bond0 trunks=1-4094
   ```

   On **Host B**:

   ```
   sudo ovs-vsctl set port eno1 trunks=1-4094
   ```

4. **Define the network in libvirt to handle VLAN trunking**:

   Create a network XML configuration for VLAN trunking on each host.

   On **Host A**, create `/etc/libvirt/qemu/networks/vlan-trunk.xml`:

   ```
   <network>
       <name>vlan-trunk</name>
       <forward mode='bridge'/>
       <bridge name='ovsbr0'/>
       <virtualport type='openvswitch'/>
   </network>
   ```

   On **Host B**, create `/etc/libvirt/qemu/networks/vlan-trunk.xml`:

   ```
   <network>
       <name>vlan-trunk</name>
       <forward mode='bridge'/>
       <bridge name='ovsbr0'/>
       <virtualport type='openvswitch'/>
   </network>
   ```

5. **Define and start the network in libvirt**:

   On **Host A**:

   ```
   sudo virsh net-define /etc/libvirt/qemu/networks/vlan-trunk.xml
   sudo virsh net-start vlan-trunk
   sudo virsh net-autostart vlan-trunk
   ```

   On **Host B**:

   ```
   sudo virsh net-define /etc/libvirt/qemu/networks/vlan-trunk.xml
   sudo virsh net-start vlan-trunk
   sudo virsh net-autostart vlan-trunk
   ```

6. **Attach VMs to the VLAN trunk network**:

   When defining the network interface for your VMs, make sure to specify the VLAN trunk network. Update the VM's XML configuration to include the network interface configuration. Here's an example:

```
<interface type='network'>
    <mac address='52:54:00:xx:xx:xx'/>
    <source network='vlan-trunk'/>
    <model type='virtio'/>
    <virtualport type='openvswitch'/>
    <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
</interface>
```

You can edit the VM's XML configuration with:

```
sudo virsh edit <vm-name>
```

7. **Configure VLANs within the VMs**:

   Inside each VM, you will need to configure the VLAN interfaces. Here's an example for a Linux VM:

   ```
   sudo apt-get install vlan
   sudo modprobe 8021q

   sudo vconfig add eth0 15
   sudo ip addr add 192.168.15.2/24 dev eth0.15
   sudo ip link set up eth0.15
   ```

   This configuration example creates a VLAN interface `eth0.15` for VLAN ID 15.

Q in Q

To enable Q-in-Q tunneling, where all nodes share the same outer VLAN tag but can use different inner VLAN tags, you need to configure the outer VLAN (S-VLAN) on the physical interfaces and set up the networking on your hosts and VMs accordingly. Here's a step-by-step guide:

1. Configure the Outer VLAN on Physical Interfaces

On Host A

1. Add the Open vSwitch bridge and physical interface.

   ```
   sudo ovs-vsctl add-br ovsbr0
   sudo ovs-vsctl add-port ovsbr0 bond0
   ```

2. Add the outer VLAN tag (e.g., VLAN 100) on the physical interface.

   ```
   sudo ovs-vsctl add-port ovsbr0 vlan100 -- set interface vlan100 type=internal
   sudo ovs-vsctl set port bond0 trunks=100
   ```

On Host B

1. Add the Open vSwitch bridge and physical interface.

   ```
   sudo ovs-vsctl add-br ovsbr0
   sudo ovs-vsctl add-port ovsbr0 eno1
   ```

2. Add the outer VLAN tag (e.g., VLAN 100) on the physical interface.

   ```
   sudo ovs-vsctl add-port ovsbr0 vlan100 -- set interface vlan100 type=internal
   sudo ovs-vsctl set port eno1 trunks=100
   ```

2. Define the Network in libvirt for Q-in-Q

Create a network XML configuration for the Q-in-Q network on each host.

On Host A, create `/etc/libvirt/qemu/networks/qinq.xml`

```
<network>
    <name>qinq</name>
    <forward mode='bridge'/>
    <bridge name='ovsbr0'/>
    <virtualport type='openvswitch'/>
    <portgroup name='qinq'>
        <vlan>
            <tag id='100'/> <!-- Outer VLAN (Service VLAN) -->
        </vlan>
    </portgroup>
</network>
```

On Host B, create `/etc/libvirt/qemu/networks/qinq.xml`

```
<network>
    <name>qinq</name>
    <forward mode='bridge'/>
    <bridge name='ovsbr0'/>
    <virtualport type='openvswitch'/>
    <portgroup name='qinq'>
        <vlan>
            <tag id='100'/> <!-- Outer VLAN (Service VLAN) -->
        </vlan>
    </portgroup>
</network>
```

3. Define and Start the Network in libvirt

On Host A

```
sudo virsh net-define /etc/libvirt/qemu/networks/qinq.xml
sudo virsh net-start qinq
sudo virsh net-autostart qinq
```

On Host B

```
sudo virsh net-define /etc/libvirt/qemu/networks/qinq.xml
sudo virsh net-start qinq
sudo virsh net-autostart qinq
```

4. Attach VMs to the Q-in-Q Network

When defining the network interface for your VMs, make sure to specify the Q-in-Q network. Update the VM's XML configuration to include the network interface configuration. Here's an example:

```
<interface type='network'>
    <mac address='52:54:00:xx:xx:xx'/>
    <source network='qinq'/>
    <model type='virtio'/>
    <virtualport type='openvswitch'>
        <parameters>
            <param name='qinq-mode' value='802.1ad'/>
        </parameters>
    </virtualport>
    <vlan>
        <tag id='200'/> <!-- Inner VLAN (Customer VLAN) -->
    </vlan>
</interface>
```

You can edit the VM's XML configuration with:

```
sudo virsh edit <vm-name>
```

5. Configure VLANs within the VMs

Inside each VM, you will need to configure the VLAN interfaces. Here's an example for a Linux VM:

```
sudo apt-get install vlan
sudo modprobe 8021q

sudo vconfig add eth0 200
sudo ip addr add 192.168.200.2/24 dev eth0.200
sudo ip link set up eth0.200
```

This configuration example creates a VLAN interface `eth0.200` for the inner VLAN ID 200.

Summary

- The outer VLAN (S-VLAN) is configured on the physical interfaces (`bond0` on Host A and `eno1` on Host B) to tag packets with VLAN ID 100.
- The inner VLAN (C-VLAN) is configured within the VMs to tag packets with customer-specific VLAN IDs (e.g., VLAN 200).
- The VMs are connected to a libvirt network configured to support Q-in-Q, allowing different inner VLAN tags to be encapsulated within the same outer VLAN tag.

This setup enables all nodes to use the same outer VLAN tag when communicating outside but allows for different inner VLAN tags for the networks they host.