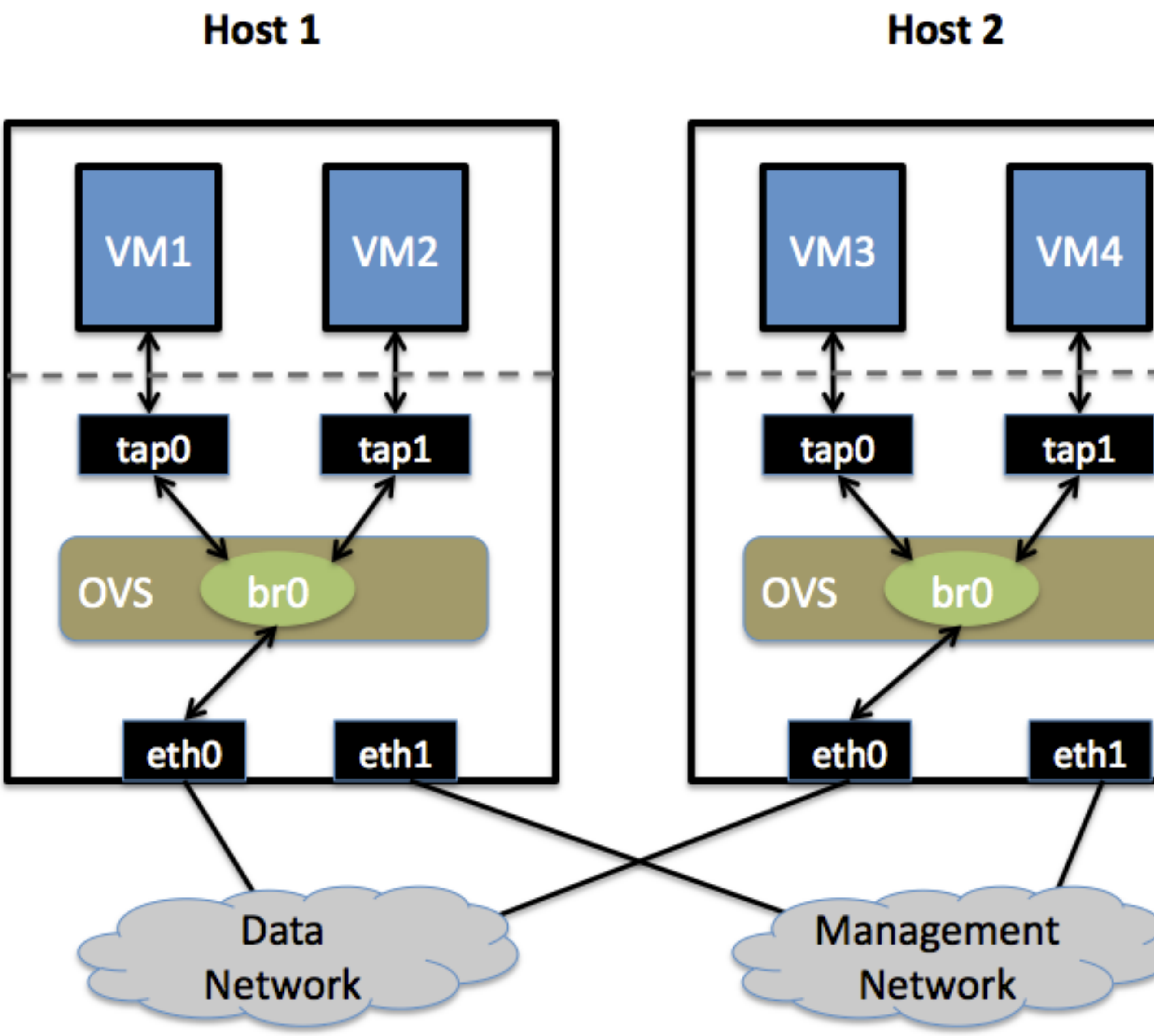
Isolating VM Traffic Using VLANs — Open vSwitch 3.2.90 documentation

This document describes how to use Open vSwitch is to isolate VM traffic using VLANs.



Setup¶

This guide assumes the environment is configured as described below.

Two Physical Networks¶

- Data Network

Ethernet network for VM data traffic, which will carry VLAN-tagged traffic between VMs. Your physical switch(es) must be capable of forwarding VLAN-tagged traffic and the physical switch ports should operate as VLAN trunks. (Usually this is the default behavior. Configuring your physical switching hardware is beyond the scope of this document.)

- Management Network

  This network is not strictly required, but it is a simple way to give the physical host an IP address for remote access, since an IP address cannot be assigned directly to eth0 (more on that in a moment).

Two Physical Hosts¶

The environment assumes the use of two hosts: *host1* and *host2*. Both hosts are running Open vSwitch. Each host has two NICs, eth0 and eth1, which are configured as follows:

- eth0 is connected to the Data Network. No IP address is assigned to eth0.
- eth1 is connected to the Management Network (if necessary). eth1 has an IP address that is used to reach the physical host for management.

Four Virtual Machines¶

Each host will run two virtual machines (VMs). *vm1* and *vm2* are running on *host1*, while *vm3* and *vm4* are running on *host2*.

Each VM has a single interface that appears as a Linux device (e.g., `tap0`) on the physical host.

Note

VM interfaces may appear as Linux devices with names like `vnet0`, `vnet1`, etc.

Configuration Steps¶

Perform the following configuration on *host1*:

1. Create an OVS bridge:
2. Add `eth0` to the bridge:

   ```
   $ ovs-vsctl add-port br0 eth0
   ```

   Note

   By default, all OVS ports are VLAN trunks, so eth0 will pass all VLANs

   Note

   When you add eth0 to the OVS bridge, any IP addresses that might have been assigned to eth0 stop working. IP address assigned to eth0 should be migrated to a different interface before adding eth0 to the OVS bridge. This is the reason for the separate management connection via eth1.

3. Add *vm1* as an “access port” on VLAN 100. This means that traffic coming into OVS from VM1 will be untagged and considered part of VLAN 100:

   ```
   $ ovs-vsctl add-port br0 tap0 tag=100
   ```

   Add VM2 on VLAN 200:

   ```
   $ ovs-vsctl add-port br0 tap1 tag=200
   ```

Repeat these steps on *host2*:

1. Setup a bridge with eth0 as a VLAN trunk:

   ```
   $ ovs-vsctl add-br br0
   $ ovs-vsctl add-port br0 eth0
   ```

2. Add VM3 to VLAN 100:

   ```
   $ ovs-vsctl add-port br0 tap0 tag=100
   ```

3. Add VM4 to VLAN 200:

   ```
   $ ovs-vsctl add-port br0 tap1 tag=200
   ```

Validation¶

Pings from *vm1* to *vm3* should succeed, as these two VMs are on the same VLAN.

Pings from *vm2* to *vm4* should also succeed, since these VMs are also on the same VLAN as each other.

Pings from *vm1*/*vm3* to *vm2*/*vm4* should not succeed, as these VMs are on different VLANs. If you have a router configured to forward between the VLANs, then pings will work, but packets arriving at *vm3* should have the source MAC address of the router, not of *vm1*.