Layer 2 Configurations

Jem Camba - 2019-04-08 - in Networking

Overview

Layer 2 feature lets you provision between one and twelve (per project) project-specific layer 2 networks. For more details about the basics of this feature, please read our overview article.

Common Use Cases

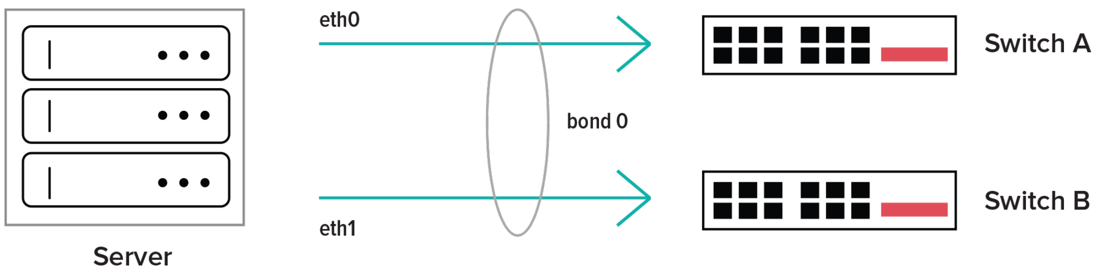
Over time, this feature may evolve to support security, provisioning updates, and other use cases.   While there are other possibilities, we have outlined steps to achieve three technical scenarios below:

- Hybrid mode: Leaving eth0 in bond0 and adding a single VLAN to eth1.
- Hybrid mode: Leaving eth0 on bond0 and adding (trunking) multiple VLANs to eth1.
- Hybrid and pure L2: Dismantling bond0 on one or more nodes and using a single node in hybrid mode as an internet gateway.

iPlease Note: For the purposes of this documentation, we are using eth0/eth1 to represent the first and second NIC. The actual interface name will depend on what operating system and hardware config you are using.

Bonding on Packet

By default, each server has two interfaces that are setup in an LACP bond that is configured both in the Host OS and on the switch:

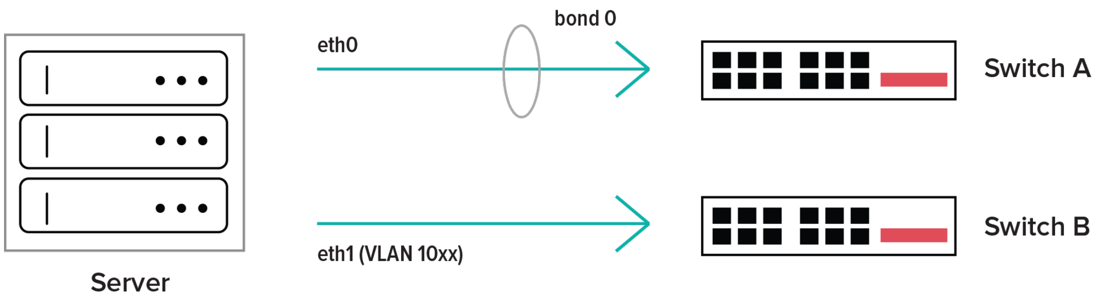


Steps for Common Configurations

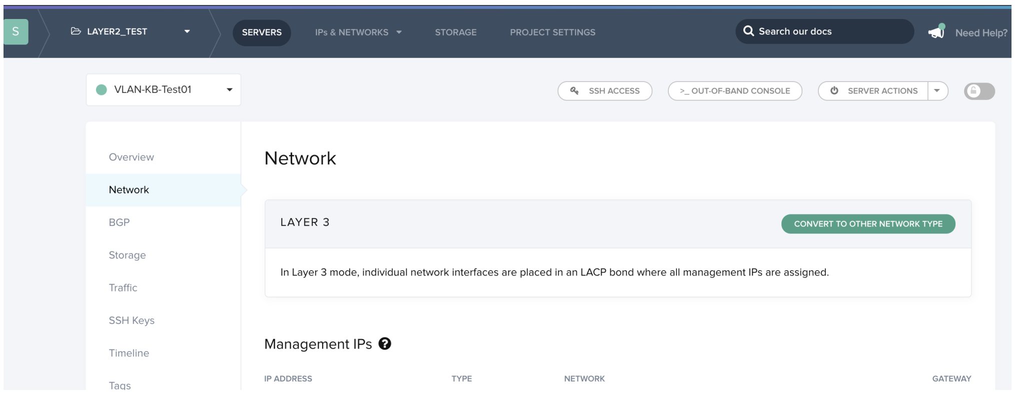
Configuration #1: Leaving eth0 in bond0 and adding a single VLAN to eth1.

In this example, you will need at least 2 servers (m1.xlarge or c1.xlarge) in the same project and at least 1 Virtual Network. We will be removing eth1 from bond0, attaching a VLAN to it, and pinging between the hosts.

You will still be able to connect to the server via its public IPv4/IPv6 addresses that are visible in the portal/API because we are leaving eth0 and bond0 intact.

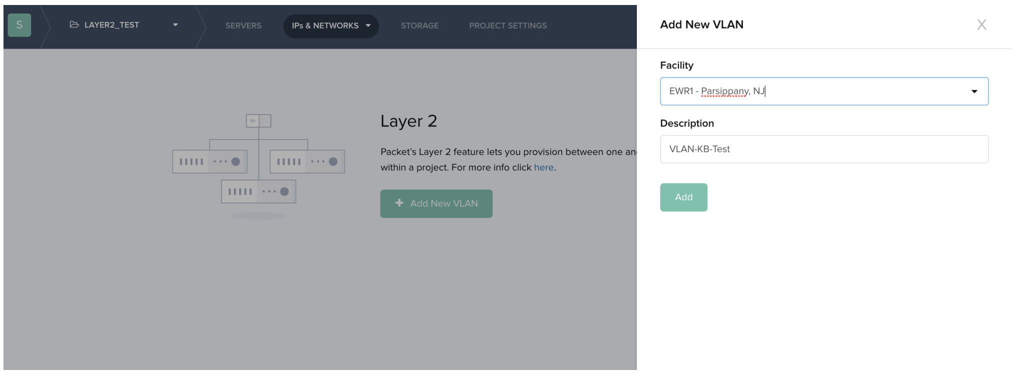


1. From the portal, browse to both servers' networking overview pages, click "Convert To Other Network Type" and choose "mixed/hybrid."



In Layer 3 mode, individual network interfaces are placed in an LACP bond where all management IPs are assigned.

2. From the networking page for both servers click "Add New Vlan", and choose eth1 as the interface and select the Virtual Network ID (VNID, or VLAN ID) you wish to use. The VNIDs must be the same for both servers for intra-host communication to work.



Packet's Layer 2 feature lets you provision between one an within a project. For more info click here.

Facility: EWR1 - Parsippany, NJ

Description: VLAN-KB-Test

3. From your Host OS, follow the steps below, depending on your operating system:

CentOS:

make sure eth1 has been removed from bond0:

```
cat /sys/class/net/bond0/bonding/slaves
```

If not, remove it:

```
echo "-eth1" > /sys/class/net/bond0/bonding/slaves
```

Bring down the eth1 interface:

```
sudo ifdown eth1
```

Configure /etc/sysconfig/network-scripts/ifcfg-eth1 on each server, changing the IPADDR field to the desired IP and network. Ensure the IP addresses are different on each host, but belong to the same network.

```
DEVICE=eth1
ONBOOT=yes
HWADDR=e4:1d:2d:11:22:33
IPADDR=192.168.1.2
NETMASK=255.255.255.0
NETWORK=192.168.1.0
BOOTPROTO=none
```

Bring up the interface:

```
sudo ifup eth1
```

Ubuntu/Debian

make sure eth1 has been removed from bond0:

```
cat /sys/class/net/bond0/bonding/slaves
```

If not, remove it:

```
echo "-eth1" > /sys/class/net/bond0/bonding/slaves
```

Bring down the eth1 interface:

```
sudo ifdown eth1
```

Configure /etc/network/interfaces on each server, changing the IP address to the desired IP from your chosen block:

```
auto eth1
iface eth1 inet static
   address 192.168.1.2
   netmask 255.255.255.0
```

Bring up the interface:

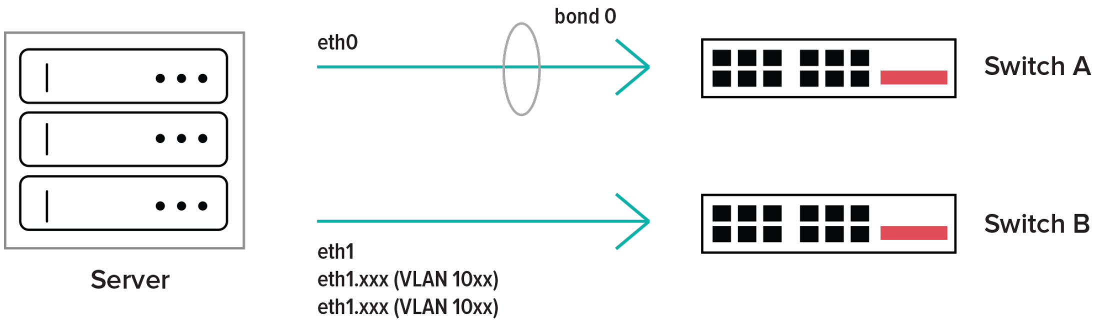
```
sudo ifup eth1
```

You should now be able to communicate between hosts via your virtual Layer 2 network:

```
root@layer2:~# ping -I eth1 192.168.1.2
PING 192.168.1.3 (192.168.1.3) from 192.168.1.4 eth1: 56(84) bytes of data.
64 bytes from 192.168.1.3: icmp_seq=1 ttl=64 time=0.106 ms
64 bytes from 192.168.1.3: icmp_seq=2 ttl=64 time=0.110 ms
64 bytes from 192.168.1.3: icmp_seq=3 ttl=64 time=0.115 ms
^C
--- 192.168.1.3 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.106/0.110/0.115/0.009 ms
```

Please Note: It is not recommended to use the subnet starting with 10.x.x.x as we use this for server's private networking and collisions could occur if you used the same private addressing as was configured on your host.

C #2: Leaving eth0 in bond0 and adding multiple VLANs to eth1.
In this case, we will be keeping the same configuration for eth0, except we will be assigning a second VLAN to eth1.



In this scenario, IP packets that arrive at the host will have the VLAN ID populated. You will need to setup two interfaces that will receive packets destined for each VLAN.

CentOS

Install the prerequisites for VLANs:

```
sudo modprobe 8021q
sudo echo "8021q" >> /etc/modules
```

bring down eth1:

```
ifdown eth1
```

Configure /etc/sysconfig/network-scripts/ifcfg-eth1.1000 and Configure /etc/sysconfig/network-scripts/ifcfg-eth1.1001 on your server. 1000 and 1001 should match the VLANs you've configured on the host in the portal/API.

```
DEVICE=eth1.1000
BOOTPROTO=none
ONBOOT=yes
IPADDR=192.168.1.2
PREFIX=24
NETWORK=192.168.1.0
VLAN=yes
```

Restart networking, or ifup eth1.1000 and eth1.1001:

```
sudo ifup eth1.1000
sudo ifup eth1.1001
```

Ubuntu

Install the prerequisites for VLANs:

```
sudo apt-get install vlan
sudo modprobe 8021q
sudo echo "8021q" >> /etc/modules
```

Bring down eth1:

```
ifdown eth1
```

Note: if you don't want eth1 to come up after a reboot be sure to comment out the eth1 configuration in your /etc/network/interfaces file.

Add the new interface to /etc/network/interfaces, changing 1000 to match the VLAN IDs:

```
auto eth1.1000
iface eth1.1000 inet static
    address 192.168.100.1
```

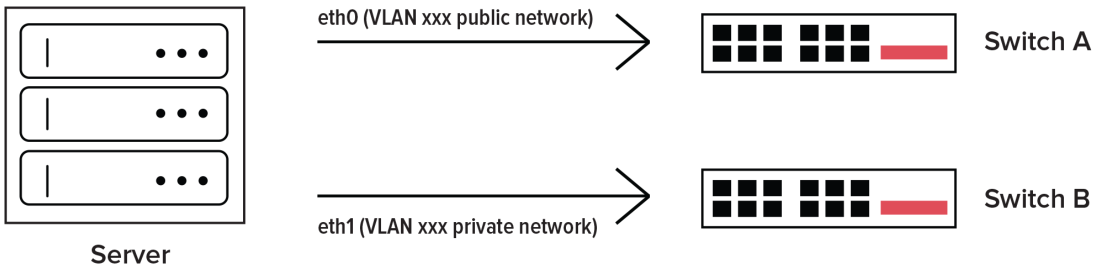
```
        netmask 255.255.255.0
    vlan-raw-device eth1

auto eth1.1001
iface eth1.1001 inet static
    address 172.16.100.1
    netmask 255.255.255.0
    vlan-raw-device eth1
```

Restart networking, or ifup eth1.1000 and eth1.1001:

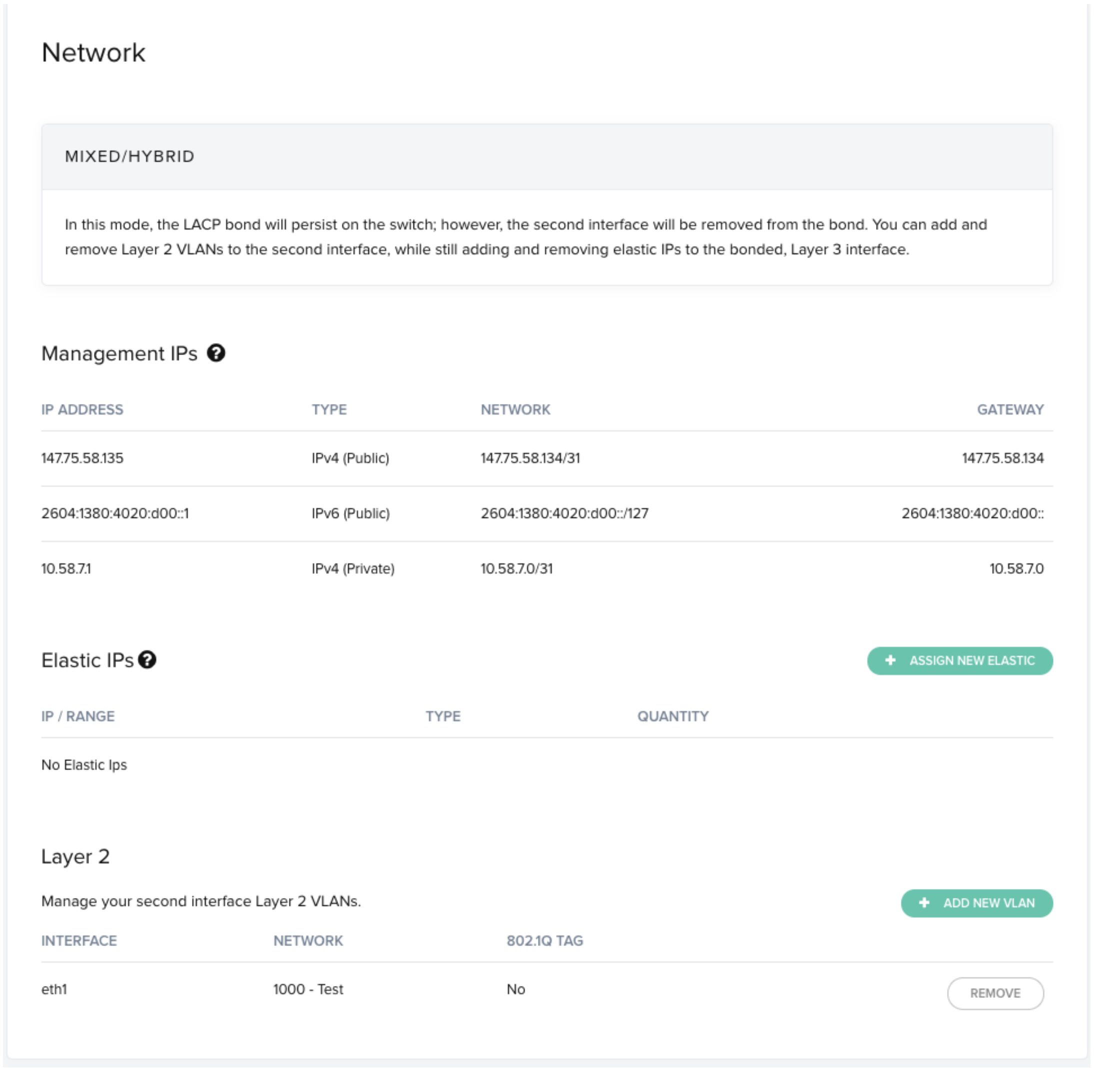
```
sudo ifup eth1.1000
sudo ifup eth1.1001
```

C #3: One hybrid node with eth0 in bond0 and eth1 connected to an isolated node in L2 with bond0 dismantled.


eth0 (VLAN xxx public network)
Switch A
eth1 (VLAN xxx private network)
Switch B
Server

For this configuration you'll need two nodes, one in hybrid and one in layer 2 networking mode, and one VLAN (although you could change it to use more, as in configuration #2).

Attach the VLAN to the hybrid node's eth1 interface, as shown below.



Network

MIXED/HYBRID

In this mode, the LACP bond will persist on the switch; however, the second interface will be removed from the bond. You can add and remove Layer 2 VLANs to the second interface, while still adding and removing elastic IPs to the bonded, Layer 3 interface.

Management IPs

IP ADDRESS	TYPE	NETWORK	GATEWAY
147.75.58.135	IPv4 (Public)	147.75.58.134/31	147.75.58.134
2604:1380:4020:d00::1	IPv6 (Public)	2604:1380:4020:d00::/127	2604:1380:4020:d00::
10.58.7.1	IPv4 (Private)	10.58.7.0/31	10.58.7.0

Elastic IPs

IP / RANGE	TYPE	QUANTITY
No Elastic Ips		

Layer 2

Manage your second interface Layer 2 VLANs.

INTERFACE	NETWORK	802.1Q TAG	
eth1	1000 - Test	No	REMOVE

Repeat this step for the isolated node. Remember, this node is in pure layer 2 networking mode, and bond0 is dismantled.

Keep in mind that disabling bonding will remove public connectivity from your server and you will have to use SOS to connect. If you get locked out, you can always re-enable and SSH back in via the public IPv4 address.

While connected to SOS, edit the network interfaces file and remove all but the eth1 interface, which should be configured with it's own private IP from whichever block you choose to use (e.g. 192.168.2.0/24). You'll also need to specify the gateway address as the hybrid node's eth1 IP address.

CentOS

Tear down the bond0 interface:

```
sudo ifdown bond0
```

Configure /etc/sysconfig/network-scripts/ifcfg-eth1 with any free IP from the IPv4 private block used by eth1 on the hybrid node. Ensure that the netmask, network, and gateway details are correct:

```
DEVICE=eth1
ONBOOT=yes
HWADDR=e4:1d:2d:11:22:32
IPADDR=192.168.2.2
NETMASK=255.255.255.0
GATEWAY=192.168.2.1
NETWORK=192.168.2.0
BOOTPROTO=none
```

Bring up eth1:

```
sudo ifup eth1
```

you can set the "ONBOOT" parameter for the rest of the network interfaces to "no" so they do not come up one reboots. bond0 will not be used, and eth0 will only be used if you choose to connect it to another VLAN (perhaps connected to other isolated node). In which case, it should be configured with it's own IP accordingly

Ubuntu

Tear down the bond0 interface:

```
sudo ifdown bond0
```

Configure /etc/network/interfaces with any free IP from the IPv4 private block used by eth1 on the hybrid node. Ensure that the netmask, network, and gateway details are correct:

```
auto eth1
iface eth1 inet static
   address 192.168.2.2
   netmask 255.255.255.0
   gateway 192.168.2.1
```

Bring up eth1:

```
sudo ifup eth1
```

You can remove the other interfaces from this file. Bond0 will not be used, but if you connect eth0 to another VLAN (perhaps connected to other isolated nodes) then configure it with it's own IP, accordingly.

At this point your hybrid and isolated node can talk to each other, but the isolated node cannot reach the internet. To give the isolated node internet access you must configure IP masquerading on the hybrid node.

First, make sure IP forwarding is enabled on the hybrid node.

```
sysctl net.ipv4.ip_forward=1
```

Now add a new IP masquerade rule to the NAT table with iptables. We want this to route traffic from any of our private IPs through the internet facing network interface, in this case, bond0.

```
iptables -t nat -A POSTROUTING -s 192.168.2.0/24 -o bond0 -j MASQUERADE
```

Now your isolated node should be able to ping outside the network.

```
ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=120 time=1.85 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=120 time=1.93 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=120 time=1.87 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=120 time=1.86 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=120 time=1.81 ms
```

Tags
layer 2