

Configure Private Networking In vRack

Overview

With vRack, you can interconnect servers in **any location worldwide** over a private Layer-2 network. This means servers in Europe can securely communicate with servers in Australia, Singapore, or anywhere else in your global infrastructure.

Behind the scenes, when you add your infrastructure to your vRack, we provision a **private VLAN** for your account. This allows your servers to communicate over our global backbone while remaining completely isolated from the public internet. So how do you take advantage of this?

First, choose an internal range. You may use 10.0.0.0/16, 192.0.0.0/16 or 172.0.0.0/16.

Configure Internal Interfaces

After adding a server to your vRack, you'll notice that a new network interface becomes available. You can view this by running `ip a`. The interface is usually deployed as `eno2`, but the exact name depends on the underlying hardware.

```
4: eno2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether d0:50:99:dc:f3:75 brd ff:ff:ff:ff:ff:ff
    altname enp4s0f1
    altname enxd05099dcf375
```

RHEL 10 / AlmaLinux 10 / RockyLinux 10

In **RHEL 10**, we use **NMCLI**, the command-line utility for NetworkManager, to configure the second interface. As root, check the current network status. You should see that `eno2` is listed but currently **disconnected**.

```
nmcli device status
```

To add your second NIC to NetworkManager, use the command below to create a new connection profile. Replace `eno2` with the name of your second interface, as shown in `ip a`, and replace `Private_Network` with your own network name.

```
sudo nmcli connection add type ethernet con-name Private_Network ifname eno2
```

If you check `nmcli device status` Now, you'll notice that the connection shows as **“connecting”**. The next step is to assign an IP address to the connection. Replace `Private_Network` with the network name you created earlier, and replace `172.0.0.1/24` with your own internal range.

```
nmcli connection modify Private_Network IPv4.address 172.0.0.1/24
```

Now set the connection to **manual** so it does not attempt to obtain an address automatically. Replace `Private_Network` with the network name you created earlier.

```
nmcli connection modify Private_Network IPv4.method manual
```

Finally, make the configuration **persistent across reboots** so the connection is restored automatically whenever the server starts. Replace `Private_Network` with the network name you created earlier,

```
nmcli connection modify Private_Network connection.autoconnect true
```

After restarting NetworkManager, you'll see that your second NIC is now configured with the new internal IP address.

```
systemctl restart NetworkManager
```

Repeat this process on your other servers, assigning a **unique internal IP address** to each one. Your servers will now be able to communicate with each other over the private network you've created.

RHEL 8/9 / AlmaLinux 8/9 RockyLinux 8/9

Check `ip a` for the name of the second interface after you have added your servers to the vRack. Create the following configuration file using the correct interface name from `ip a`.

```
nano /etc/sysconfig/network-scripts/ifcfg-NETWORK_INTERFACE
```

Inside the file, paste the following configuration block. Replace `172.0.0.1/24` with your own internal range, and replace `eno2` with the name of your second interface.

```
DEVICE=eno2
BOOTPROTO=static
IPADDR=172.0.0.1/24
NETMASK=NETMASK
ONBOOT=yes
TYPE=Ethernet
```

Now restart NetworkManager

```
systemctl restart NetworkManager.service
```

Repeat this process on your other servers, assigning a **unique IP address from your internal range** to each one.

Ubuntu 24 / Debian 12 + 13

To configure vRack on Debian-based distributions of 13+, we use **Netplan** to add and configure the second interface. Open the `50-cloud-init.yaml` file and locate the line that begins with `version: 2`.

```
nano /etc/netplan/50-cloud-init.yaml
```

Directly under this line, add the following code block, making sure the indentation is preserved exactly. YAML is indentation-sensitive, so even a single misplaced space will cause Netplan to fail. Ensure you switch the IP block for your internal range.

```
ethernets:
  NETWORK_INTERFACE:
    dhcp4: false
    addresses:
      - 172.0.0.1/24
```

Save and reload Netplan.

```
netplan apply
```

Repeat this process on your other servers, assigning a **unique IP address** from your internal range to each one.

Ubuntu 22 / Debian 11

Using nano, open up the current networking file.

```
/etc/network/interfaces.d/50-cloud-init
```

Add the following line, replacing `INTERFACE_NAME` with the name of your second interface and swapping the internal IP range for your own.

```
auto INTERFACE_NAME
iface INTERFACE_NAME inet static
    address 172.0.0.1/24
    netmask 255.255.255.0
```

Restart the network, then repeat this process on your other servers, ensuring each one is assigned a **unique IP address** from your internal range.

```
systemctl restart networking
```

Revision #5

Created 2026-05-02 19:43:16 UTC by F2HCloud

Updated 2026-05-02 21:34:39 UTC by F2HCloud