Creating the Network Bridge:

[**Introduction:**](#_h9mftzqt297k) **1**

[**Modify ifcfg-enpXXX:**](#_m6z8zo8da2p3) **2**

[**Create ifcfg-enpXXX:**](#_rhvfg1ao6vno) **3**

[**Restart the network:**](#_j7ovwsa79019) **3**

# Introduction:

If you want to **assign IP addresses** to your **virtual machine**s and make them **accessible** from your **LAN** you need to **setup a network bridge**. A **bridge** is a piece of software used to **unite two or more** network **segments**. A bridge behaves like a **virtual network switch**, working transparently (the other machines do not need to know or care about its existence). Any **real devices** (e.g. **eth0**) and **virtual devices** (e.g. **br0**) can be connected to it.

By **configuring** a **network bridge** and then **configuring** the **virtual machine** to use it as a **network interface**, our **virtual machine** will look like any **other machine of the network**. It can run **services**, you **login** to it and it can **mount disks**. It can even be an HTCondor node although real hardware always outperforms virtual machines.

The remote access machines in Colorado are VMs:

|  |
| --- |
| pbsgold-laptop> ssh culogin03.colorado.edu  drjohn@culogin03.colorado.edu's password:  Last login: Wed May 2 12:52:48 2018 from 198.11.29.126  ===============================================================================  University of Colorado High Energy Physics  Unauthorized or improper use of this system is strictly forbidden.  Boulder, Colorado    culogin03.colorado.edu  QEMU Virtual CPU version (cpu64-rhel6) (cpus/cores: 1))  Operating system: Scientific Linux release 7.4 (Nitrogen)  Total RAM: 3.70 GiB Total swap: 2.00 GiB  ===============================================================================  culogin03> df -h  Filesystem Size Used Avail Use% Mounted on  /dev/vda3 48G 3.1G 45G 7% /  devtmpfs 1.9G 0 1.9G 0% /dev  tmpfs 1.9G 0 1.9G 0% /dev/shm  tmpfs 1.9G 145M 1.8G 8% /run  tmpfs 1.9G 0 1.9G 0% /sys/fs/cgroup  /dev/vda1 497M 176M 322M 36% /boot  hepusers:/amd/hepusers/nfs/hepusers 1.4T 45G 1.4T 4% /nfs/hepusers  tmpfs 380M 0 380M 0% /run/user/278  hepadmin01:/amd/hepadmin01/nfs/admin01 1.8T 595G 1.2T 34% /nfs/admin01  denali:/usr/local 1.8T 97G 1.7T 6% /misc/local |

Now that **bridge-utils** is installed, we can set up the network bridge. To do this we need to **modify two files** and then **restart** the **network**.

|  |
| --- |
| **WARNING**: It is highly **recommended** that you **do not** do this over a **network connection**. If you make a **mistake**, you may lose your network connection and be **locked out of the machine**. |

* The files we need to modify are located in **/etc/sysconfig/network-scripts**.
* **ifcfg-enpXXX** - The exact name of the file will depend on the name of the Ethernet interface. This can be different on every machine. In the directions I am using **enpXXX** as a placeholder for the device name. You can find you exact device by looking at:

|  |
| --- |
| ls -l /etc/sysconfig/network-scripts/ifcfg-e\* |

* **ifcfg-br0** - The bridge file. It does not currently exist on your machine

# Modify ifcfg-enpXXX:

Change **DEVICE=”enpXXX”** to **DEVICE=”br0”** and add the line **TYPE=”Bridge”**

|  |
| --- |
| cd /etc/sysconfig/network-scripts/  mkdir -p SaveConfigs  cp ifcfg-enpXXX SaveConfigs/  mv ifcfg-enpXXX ifcfg-br0 vi ifcfg-br0 |

**After the changes**, the file should like **similar** to, but **not exactly** like:

|  |
| --- |
| UUID="8ffa05c6-a97a-43b2-b3ca-4eec4a5ec1cb"  DNS1="172.16.0.1"  IPADDR="172.16.9.192"  GATEWAY="172.16.1.2"  NETMASK="255.255.240.0"  BOOTPROTO="static"  DEVICE="**br0**"  ONBOOT="yes"  IPV6INIT="no"  **TYPE="Bridge"** |

# Create ifcfg-enpXXX:

|  |
| --- |
| cd /etc/sysconfig/network-scripts/  vi ifcfg-enpXXX |

The contents of the new file should be very similar to:

|  |
| --- |
| TYPE=Ethernet  BOOTPROTO=static  DEVICE=**enpXXX**  ONBOOT=yes  **BRIDGE=br0** |

# Restart the network:

|  |
| --- |
| systemctl restart network |

If everything is correct, you should still have network connectivity but now have a network bridge. You can use ifconfig to display the configured network devices:

|  |
| --- |
| ifconfig |

With part of the output similar to:

|  |
| --- |
| **br0** Link encap:Ethernet HWaddr A0:36:9F:59:AE:E0  **inet addr:128.138.133.25** Bcast:128.138.133.255 Mask:255.255.255.0  inet6 addr: fe80::a236:9fff:fe59:aee0/64 Scope:Link  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  RX packets:1735500142 errors:0 dropped:0 overruns:0 frame:0  TX packets:1774113760 errors:0 dropped:0 overruns:0 carrier:0  collisions:0 txqueuelen:0  RX bytes:1250311258541 (1.1 TiB) TX bytes:799226149273 (744.3 GiB)  **eth0**  Link encap:Ethernet HWaddr A0:36:9F:59:AE:E0  inet6 addr: fe80::a236:9fff:fe59:aee0/64 Scope:Link  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  RX packets:2256831911 errors:0 dropped:16 overruns:0 frame:0  TX packets:2079208093 errors:0 dropped:0 overruns:0 carrier:0  collisions:0 txqueuelen:1000  RX bytes:1318106934007 (1.1 TiB) TX bytes:819372158998 (763.0 GiB) |

You should be able to ping other hosts and the web browser on the machine should still function properly.