15th ANNUAL WORKSHOP 2019

BUILDING A VIRTUAL CLUSTER TUTORIAL DAY 1

Doug Ledford <dledford@redhat.com>

HOW TO USE THIS GUIDE

This guide assumes you are familiar with installing software on your own machine. It also assumes you are comfortable performing a minimal linux install in a virtual machine. To best utilize this guide, you should look at the available software choices under Preliminary Software Downloads and pick which virtualization environment and which linux distro you would like to use and then have them downloaded prior to proceeding with this tutorial. You should also have your virtualization software installed prior to proceeding with this tutorial. We pick up from the point of virtualization software installed and ready to start configuring things.

PRELIMINARY SOFTWARE DOWNLOADS

VIRTUALIZATION ENVIRONMENTS

One of the goals of this tutorial was to create a guide that would allow anyone to setup a test RDMA capable environment without needing to invest in expensive RDMA hardware, or expensive virtualization environments for that matter. This drove the selection of supported virtualization environments, and to a lesser extent the choices regarding the RMDA network topologies used in this tutorial. If you have access to RDMA hardware in another machine, altering the RDMA networks to be a bridge type setup instead of host only would allow you to test communications between the soft-RoCE or soft-iWARP drivers and your existing hardware.

- **Oracle VirtualBox - available on all platforms**
  - https://www.virtualbox.org/wiki/Downloads
  - In order to use USB devices freely on these clients, you need to download the Oracle VM VirtualBox Extension Pack available from the same download page, which we will install a little later
  - This is the only supported option for Windows or Mac computers. However, if you feel comfortable translating the setup instructions from VirtualBox to your preferred virtualization environment, then by all means feel free to join us anyway
- **Linux Virtual Machine Manager - ubiquitous on linux and usually slightly better integrated into the linux host distro than VirtualBox is, so we support this as well (it is not screen shotted the way VirtualBox is, but I will gladly answer people's questions)**
- **Gnome Boxes is *NOT* supported. It assumes too simplistic of an environment and does not allow the user to customize it sufficiently for our needs**

GUEST SOFTWARE

(All 64bit, 32bit HPC is poorly tested, just don't go there)

- **CentOS 7.6 DVD Install Image**
  - This is the mirror list redirection page. It should tell you what your closest mirrors are: http://isoredirect.centos.org/centos/7/isos/x86_64/CentOS-7-x86_64-DVD-1810.iso
- **Debian 9.7.0 Net Install Image (these require more Internet bandwidth during the install, but there wasn't a full install image link I could include here as they are all done via torrents)**
  - https://cdimage.debian.org/debian-cd/current/amd64/iso-cd/debian-9.7.0-amd64-netinst.iso
- **Fedora 29 Server Install Image**
  - https://download.fedoraproject.org/pub/fedora/linux/releases/29/Server/x86_64/iso/Fedora-Server-dvd-x86_64-29-1.2.iso
- **openSuSE 15.0 Leap Install Image**
  - https://download.opensuse.org/distribution/leap/15.0/iso/openSUSE-Leap-15.0-DVD-x86_64.iso
- **SuSE SLE 15 HPC (requires registration for 60 day trial, but is an HPC product instead of a normal Server product)**
  - https://www.suse.com/products/server/hpc/download/
- **Ubuntu 18.04.1 LTS Server Install Image**
  - http://releases.ubuntu.com/18.04/ubuntu-18.04.1.0-live-server-amd64.iso

GUEST SOFTWARE

Differences between distros

- **CentOS 7.6 DVD Install Image**
  - Very close to upstream. Already uses rdma-core, and kernel uses the RDMA stack out of a 4.18 or so kernel, and so works easily with soft-RoCE
- **Debian 9.7.0 Net Install Image (these require more Internet bandwidth during the install, but there wasn't a full install image link I could include here as they are all done via torrents)**
  - Uses an older kernel and uses older RDMA subsystem packages. It does not yet use rdma-core and to use soft-RoCE or soft-iWARP with this release will require building a custom kernel as well as replacing the stock RDMA user space packages with more recent ones from rdma-core. In fact, the easiest thing to do here would be to install OFED if it supports doing so. I strongly suggest avoiding this distro for working through this tutorial.
- **Fedora 29 Server Install Image**
  - Even closer to upstream than CentOS 7.6. New upstream kernels are automatically taken once Linus releases them. Already uses rdma-core and kernel version is 4.20.
- **openSuSE 15.0 Leap Install Image**
  - Very close to upstream in user space. Already uses rdma-core and has the rxe_cfg utility for working with soft-RoCE devices. Kernel version is slightly older, but still recent enough to already have the rdma_rxe module needed for soft-RoCE.
- **SuSE SLE 15 HPC (requires registration for 60 day trial, but is an HPC product instead of a normal Server product)**
  - Very close to upstream in user space. Already uses rdma-core and has the rxe_cfg utility for working with soft-RoCE devices. Kernel version is slightly older, but still recent enough to already have the rdma_rxe module needed for soft-RoCE. In addition, has a very complete HPC product selection, including things like the SLURM scheduler not found in any other distro.
- **Ubuntu 18.04.1 LTS Server Install Image**
  - Very close to upstream in user space. Already uses rdma-core and has the rxe_cfg utility for working with soft-RoCE devices. Kernel version is 4.15, so not that far behind upstream (which is at 5.0, or what would have been 4.21). Appears to have just slightly better soft-RoCE kernel support than the SuSE alternatives, mainly because Ubuntu's kernel and user space get the RMTU right while SuSE's two do not (Red Hat's two get this right as well). However, Ubuntu is also missing some things. There is no libfabric at all. Nor mvapich2. Nor any OpenMPI test suite, and it's OpenMPI is out of date. So kind of a mixed bag here when it comes to Ubuntu. I was borderline calling this one not really fit for our purposes. After walking through a significant portion of the setup on Ubuntu, I backtracked due to the number of problems and I now consider this an unsuitable distro for this tutorial. There are simply too many things to work around.

VIRTUALIZATION ENVIRONMENT SETUP

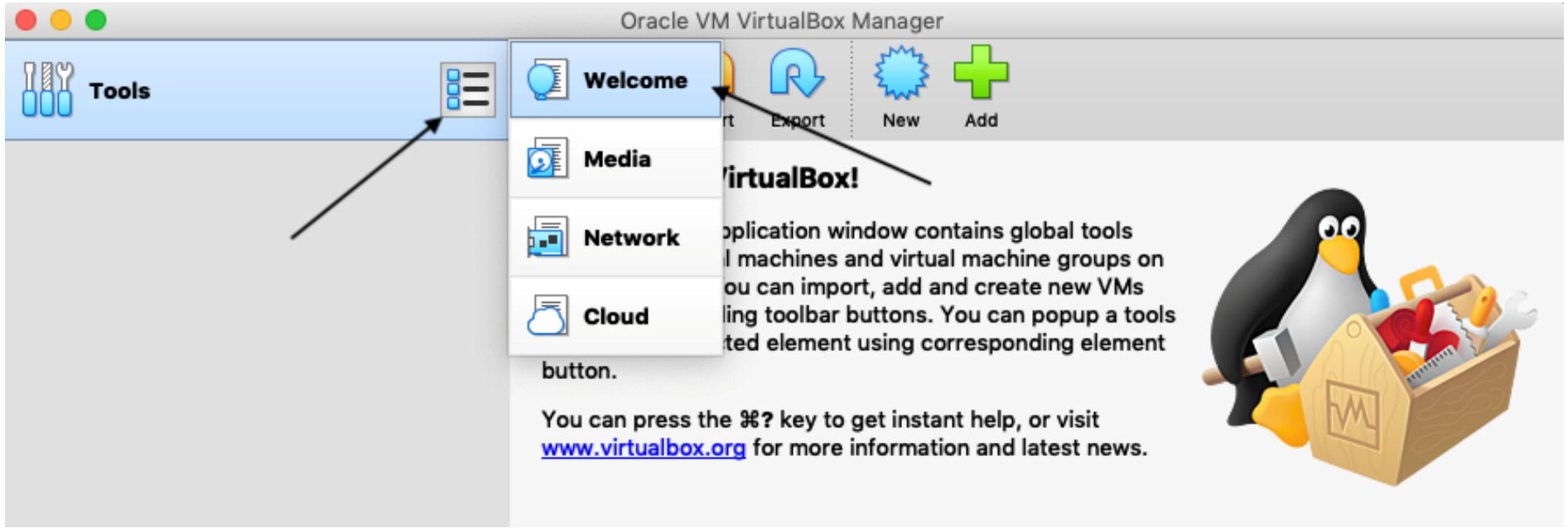
VIRTUALBOX INITIAL SETUP

VirtualBox 6.0.4

This is the initial intro screen for VirtualBox, which is where you need to be to access the Preferences Menu.

You can get back here from other screens at any time by simply clicking on the list box on the Tools bar and selecting Welcome.

For now, we will be doing the initial setup in the Preferences section available from the Welcome screen. So click on the Preferences icon and we'll get started



VIRTUALBOX INITIAL SETUP

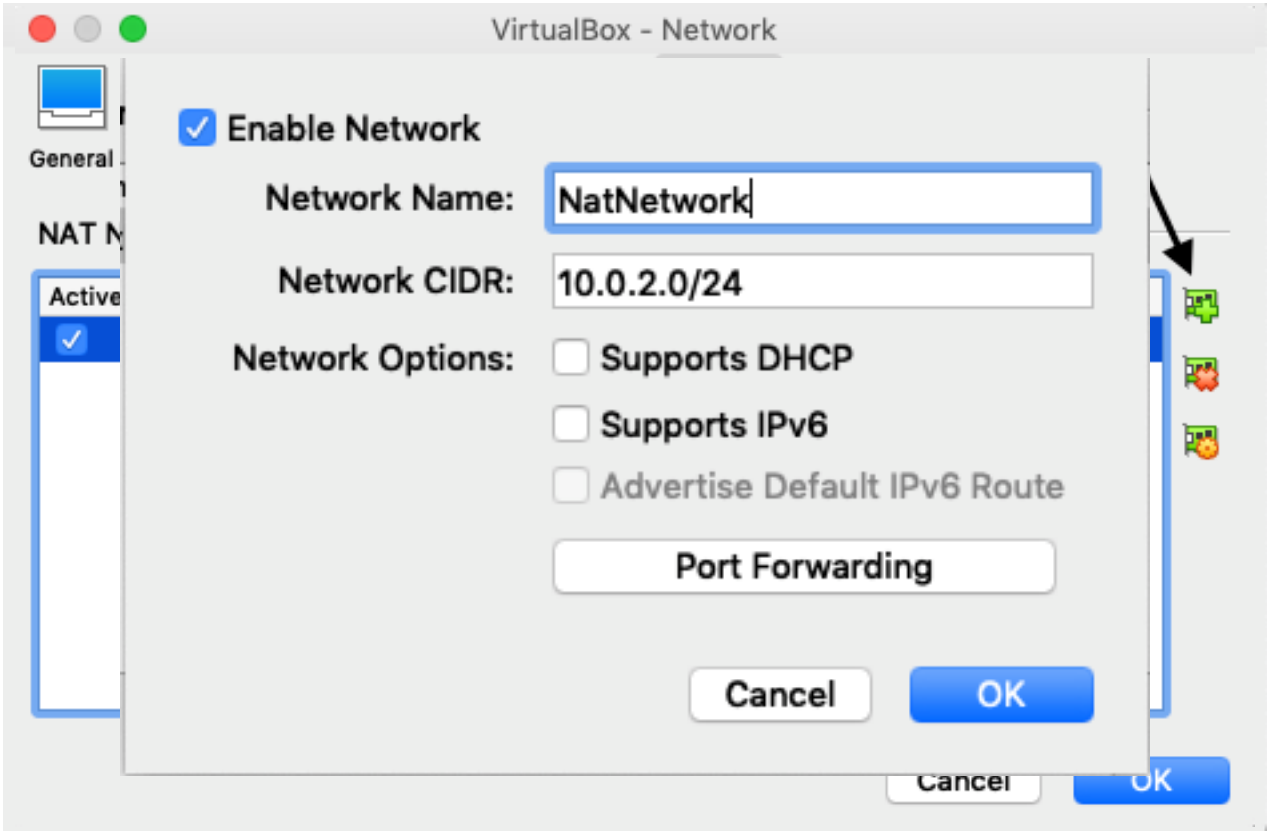
VirtualBox 6.0.4

Most of the Preferences are fine as is. The first one where we need to make changes is under the Networks tab.

Here we need to add a NatNetwork by clicking on the green Add Network button, at which point a new NetNetwork will be added to the list.

The default settings mostly fine. We won't be using dhcp (as there are some issues with the built in DHCP server in VirtualBox and linked clones, as I discovered). We can still use the default 10.0.2.0/24 network, but we'll disable DHCP support to save resources.

If we wanted our guests to be accessible from remote machines, we would need a bridge network instead (not covered in this tutorial).



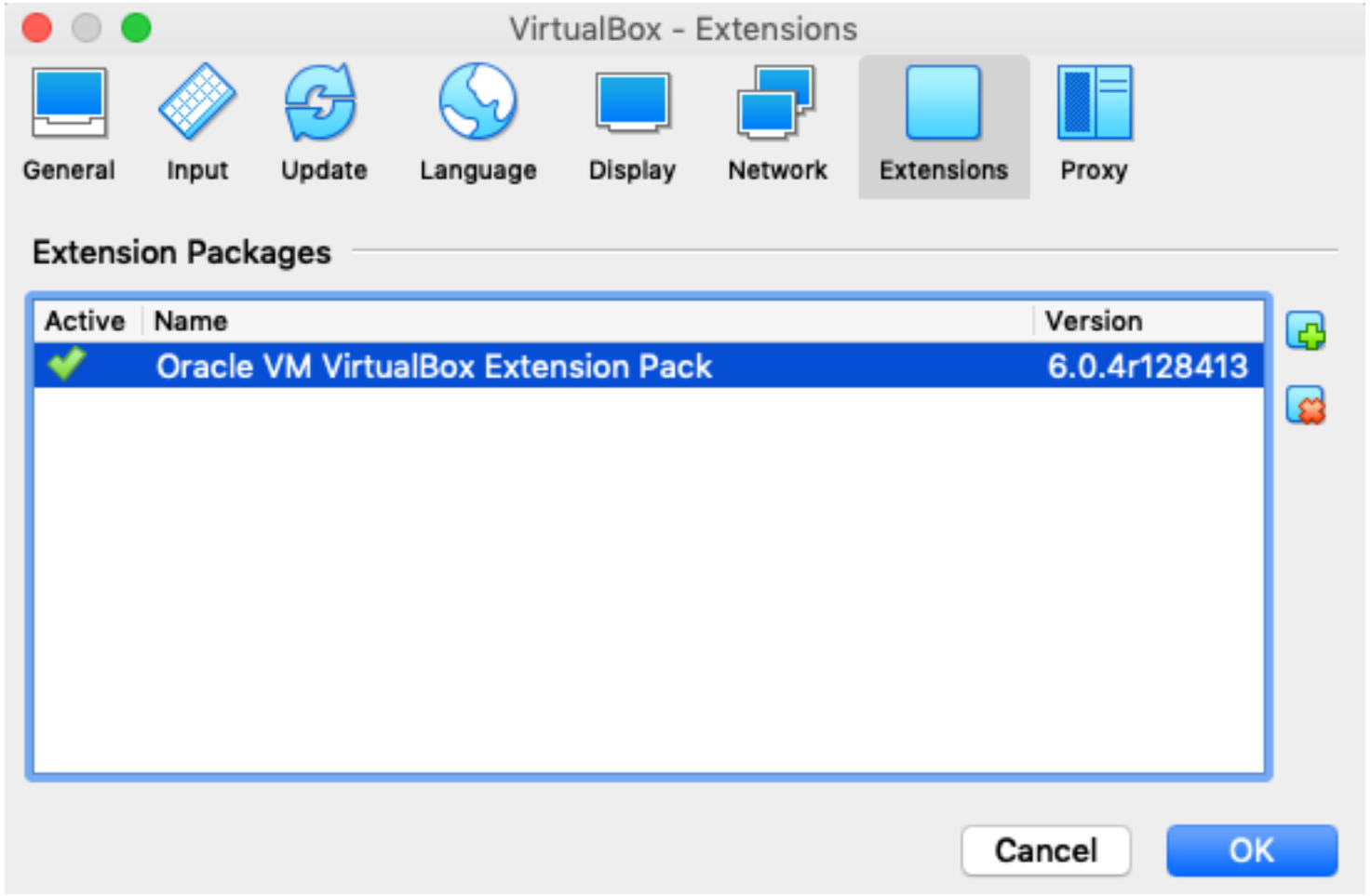
VIRTUALBOX INITIAL SETUP

VirtualBox 6.0.4

The next thing we need to do is install the Extension pack that we downloaded with VirtualBox itself.

Without this, we won't have USB2 or USB3 support, and the best pointer method in clients requires this.

Simply click on the green + symbol and select the location of the file you already downloaded to install the extension pack.



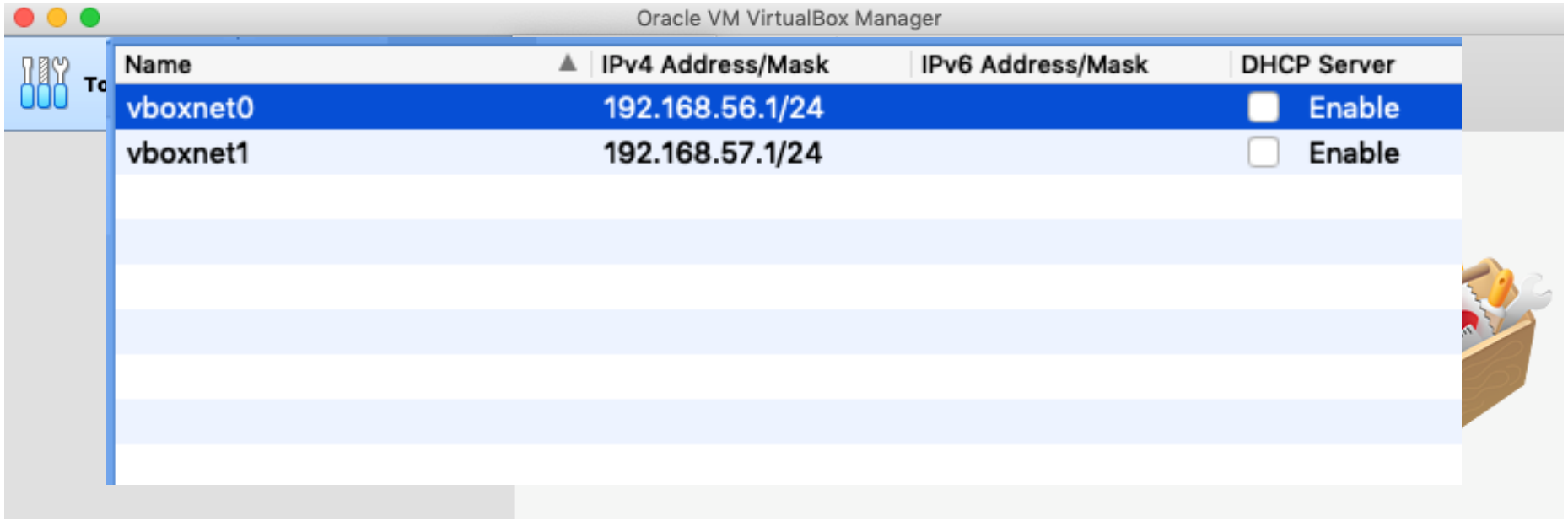
VIRTUALBOX INITIAL SETUP

VirtualBox 6.0.4

Next we need to go back to the Tools bar and select Network

On the Network screen, we need to create two new networks. Go ahead and disable the DHCP server on these networks too. These will be used as the internal RDMA capable networks.

When done, click the Tools/Welcome button to get back to the Welcome screen so we can add our first



MACHINE PROFILES

CREATING THE HEAD NODE

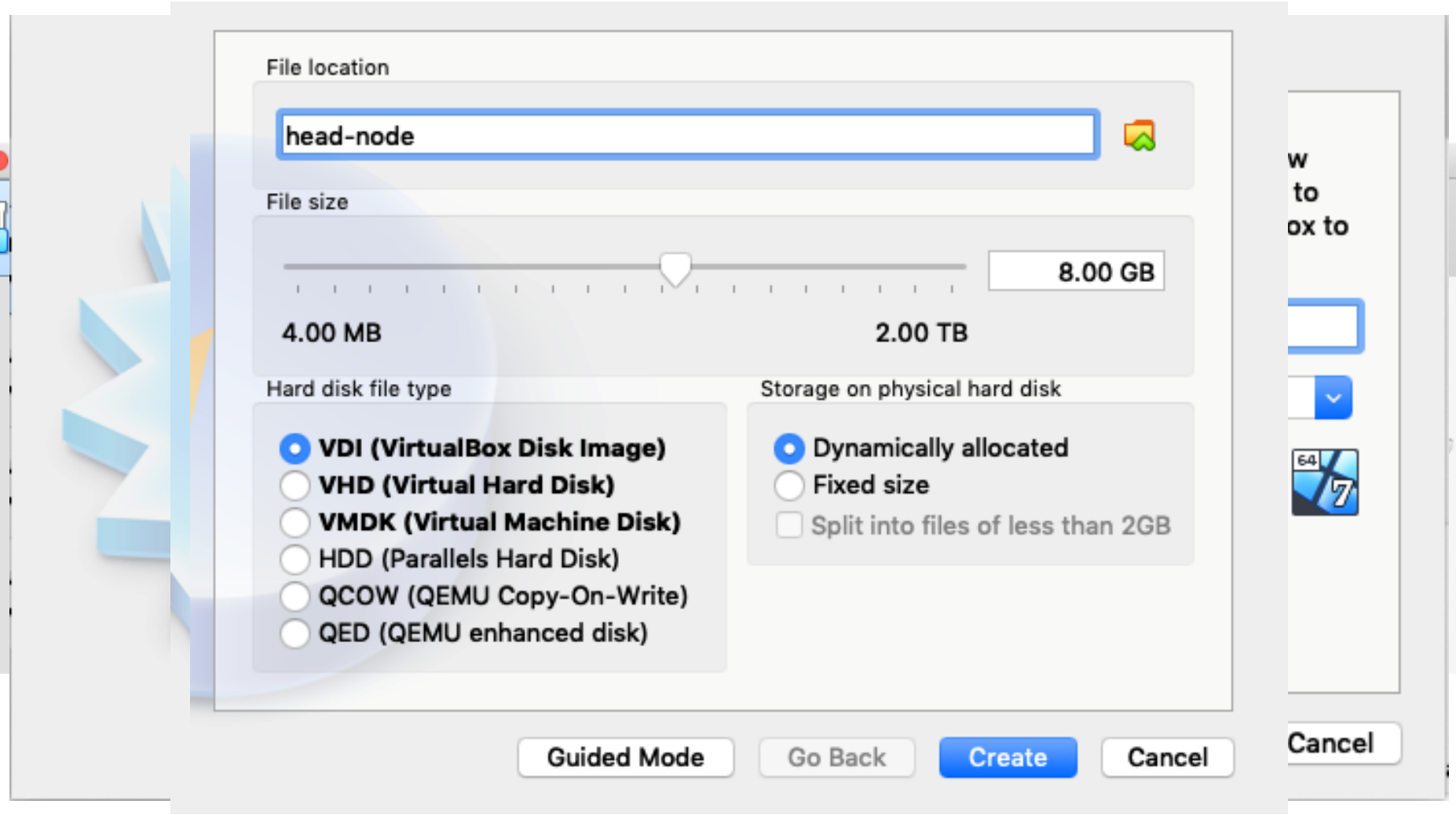
Using Expert Mode

Click on the New button to add a new machine profile.

The window is going to come up in Guided mode. Just immediately select Expert mode instead.

In Expert mode, we can select items that make sense for our custom machines. The options here are reasonable for a head node. Once you click Create, you will be allowed to specify options for the virtual hard drive.

The options here are, again, reasonable for our head node's virtual disk. Click Create to finalize our new



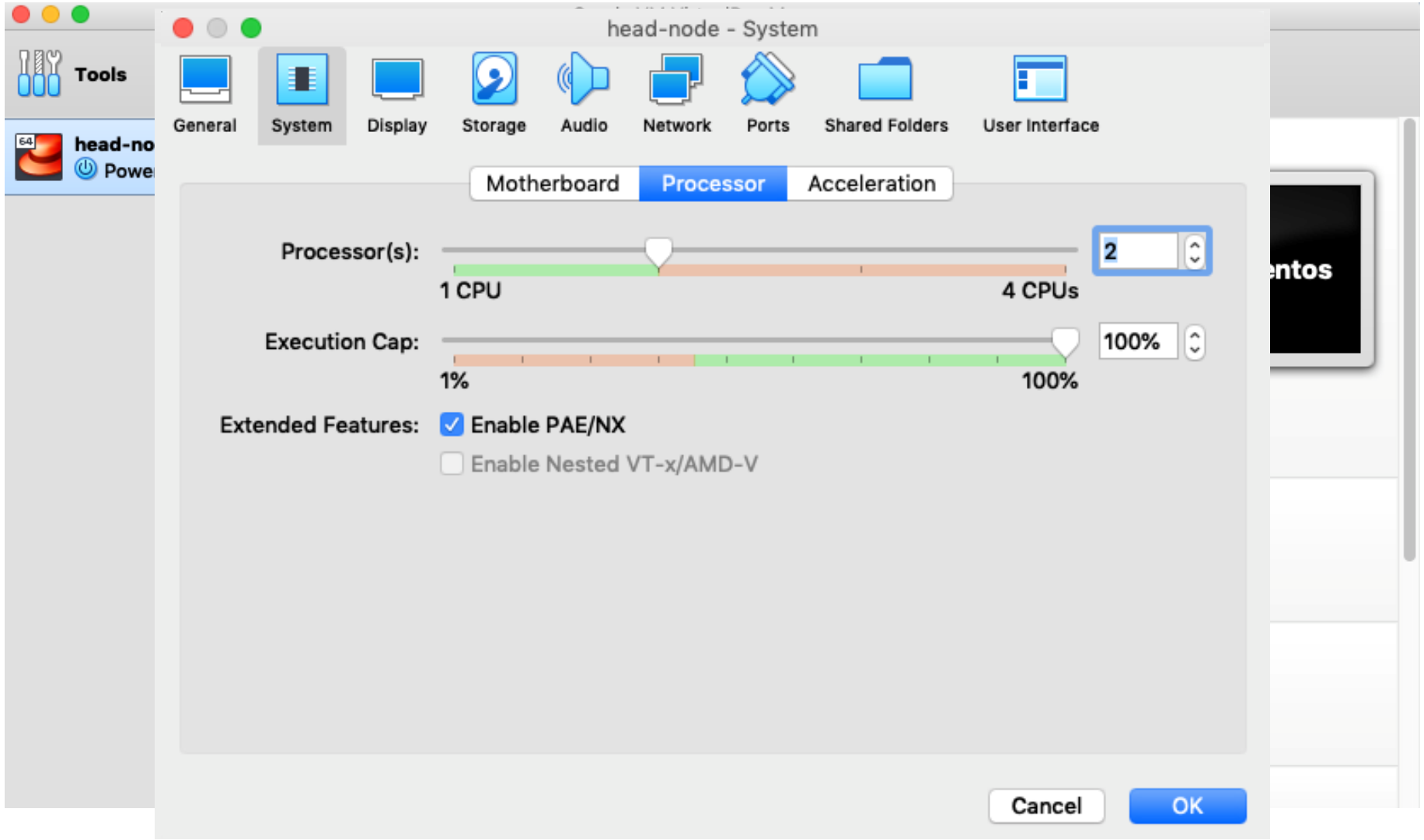
MODIFYING THE HEAD NODE

You should now have the head node on your main screen. Click the Settings button to get the full list of settings.

On the System/Motherboard area, uncheck Floppy as we don't use those, set the chipset up to ICH9, set the pointing device to USB tablet, and enable EFI.

WARNING: DO NOT ENABLE EFI ON DEBIAN (it renders your install unbootable without post install magic)

On the System/Processor area, I selected 2 CPUs for the head node



MODIFYING THE HEAD NODE

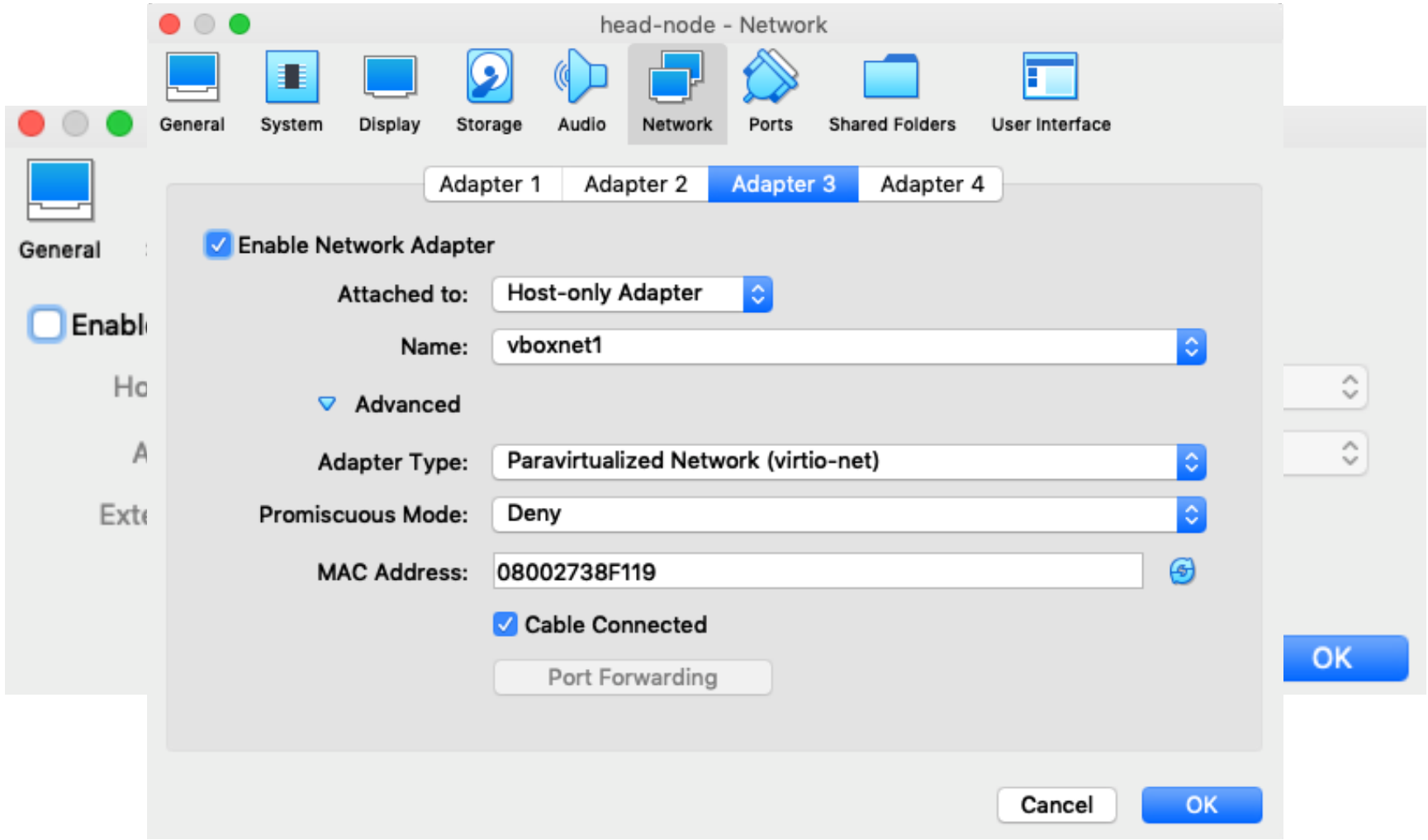
(continued)

Disable the audio entirely, it's not needed and wastes resources.

Next we need to set up our Network devices. Network device 1 is our NatNetwork. And virtio is the preferred driver type.

Network device 2 is our vboxnet0 network.

Network device 3 is our vboxnet1 network.



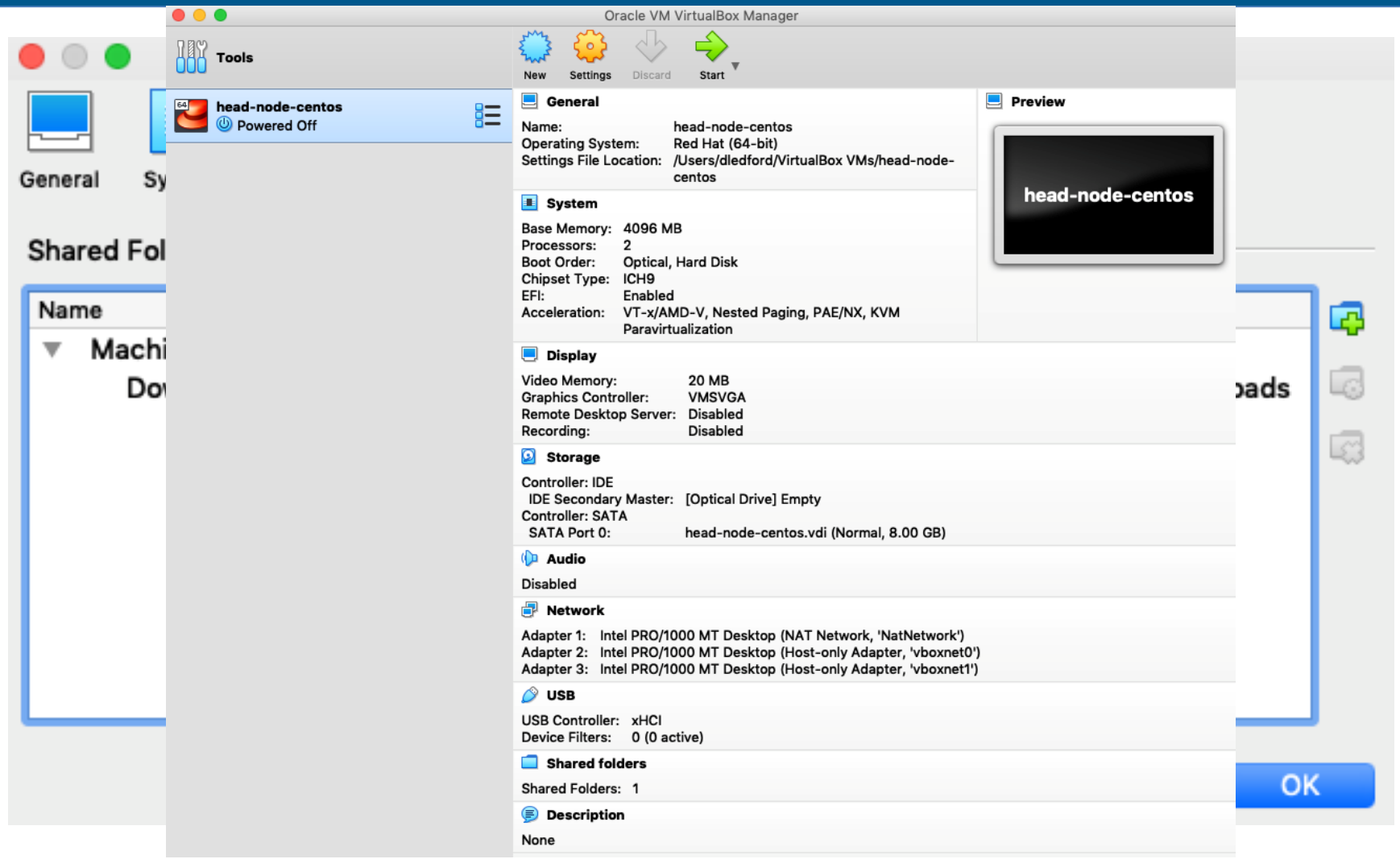
MODIFYING THE HEAD NODE

(optional items)

I also bumped the USB controller version up to USB 3.0.

And I added a shared folder to my head node so I could easily move files between the host computer and my guest client.

After saving, I was back at the welcome screen with my, now fully modified and ready to use, head node machine.



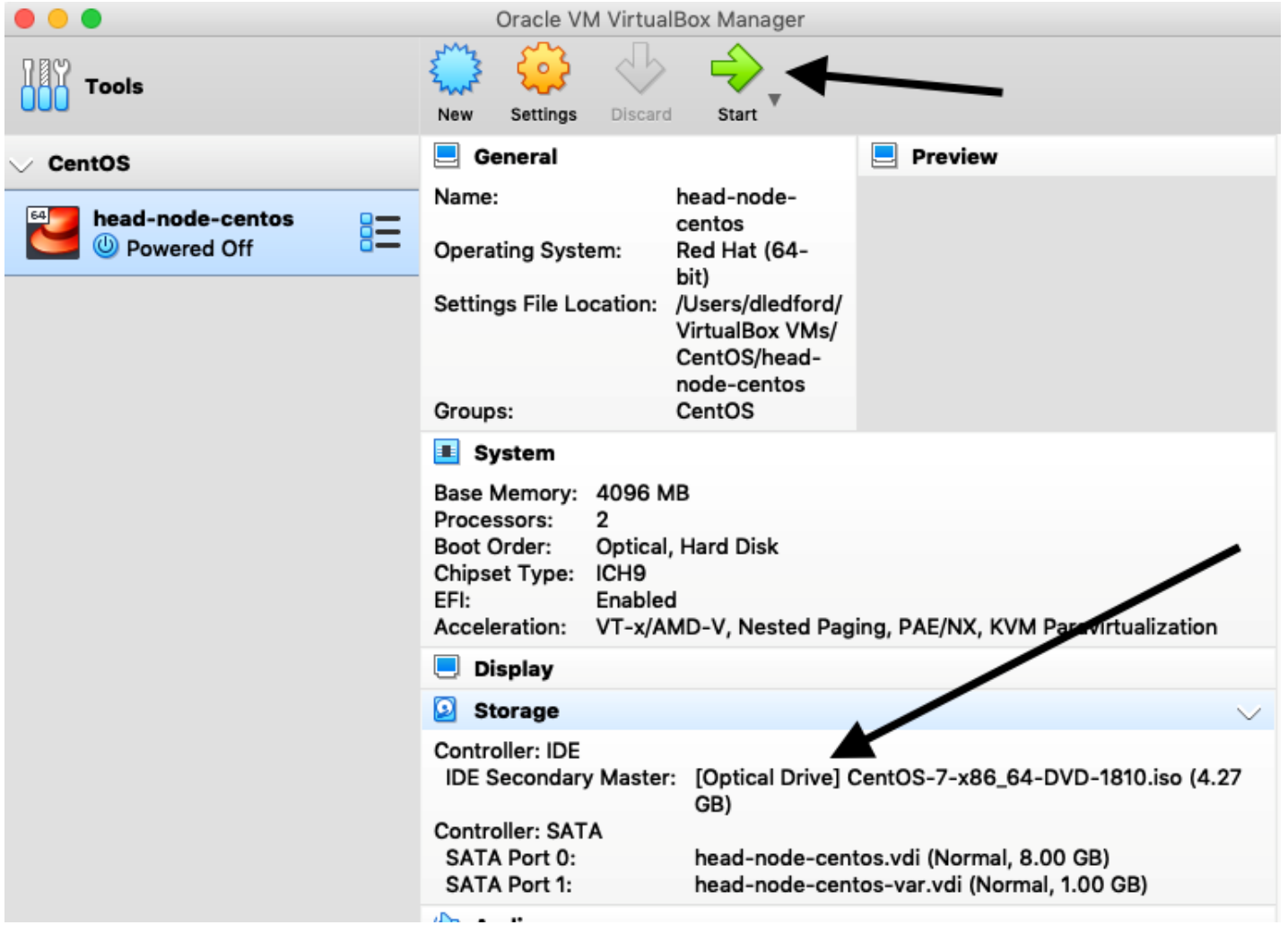
NEXT STEPS

- **It speeds the process up greatly if we don't create the compute node machine profiles yet.**
- **Instead, we will install the head node, then clone the head node to the compute nodes using a linked clone method, which allows the head node and clones to share a common virtual disk image, minimizing disk space usage on the host.**
- **Using linked clones also keeps us from having to do multiple installs.**
- **Because of this, we install a reasonable set of useful software on the head node, then login and manually adjust the installed software as needed, then link that over to the compute nodes. The compute nodes will have software on them they will never use, but it's still easier than having separate install images.**

INSTALLING THE HEAD NODE

(your choice of distro, CentOS was just mine)

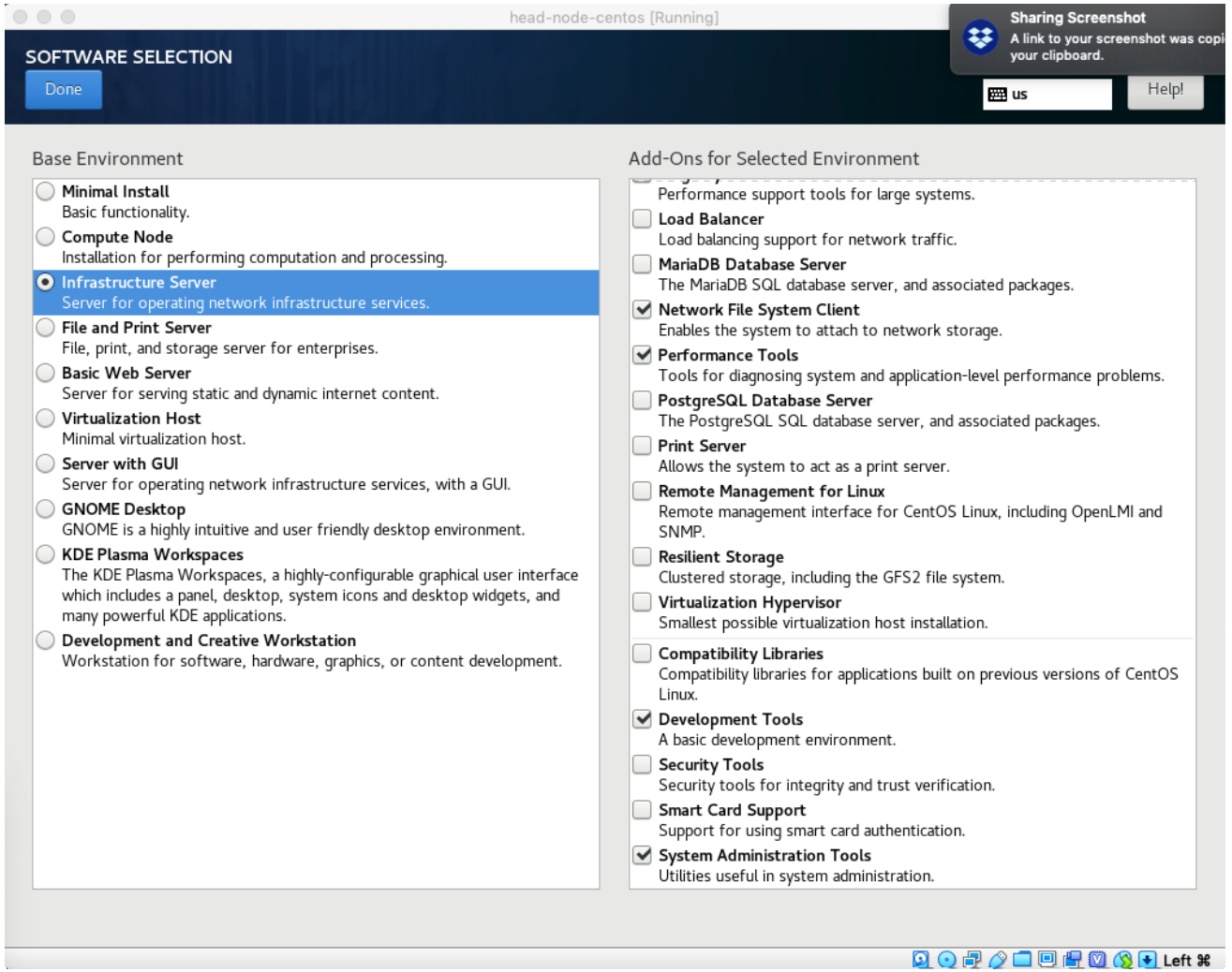
When on the main page and viewing the machine profile, you can click on the Optical Drive and VirtualBox will allow you to attach the iso image for your distro to the optical drive of the host. If you then click Start to turn the virtual machine on, it starts the install process. When you get to the end of the install, go ahead and reboot into the installed system so that we can add software we will need on all the nodes into the core disk image. This will reduce the differences between the master and cloned disk



OPTIONS I USED FOR CENTOS 7.6

Software Selection

- Infrastructure Server
  - File & Storage Server
  - Guest Agents
  - InfiniBand Support
  - Network File System Client
  - Performance Tools
  - Development Tools
  - System Admin Tools



OPTIONS I USED FOR CENTOS 7.6

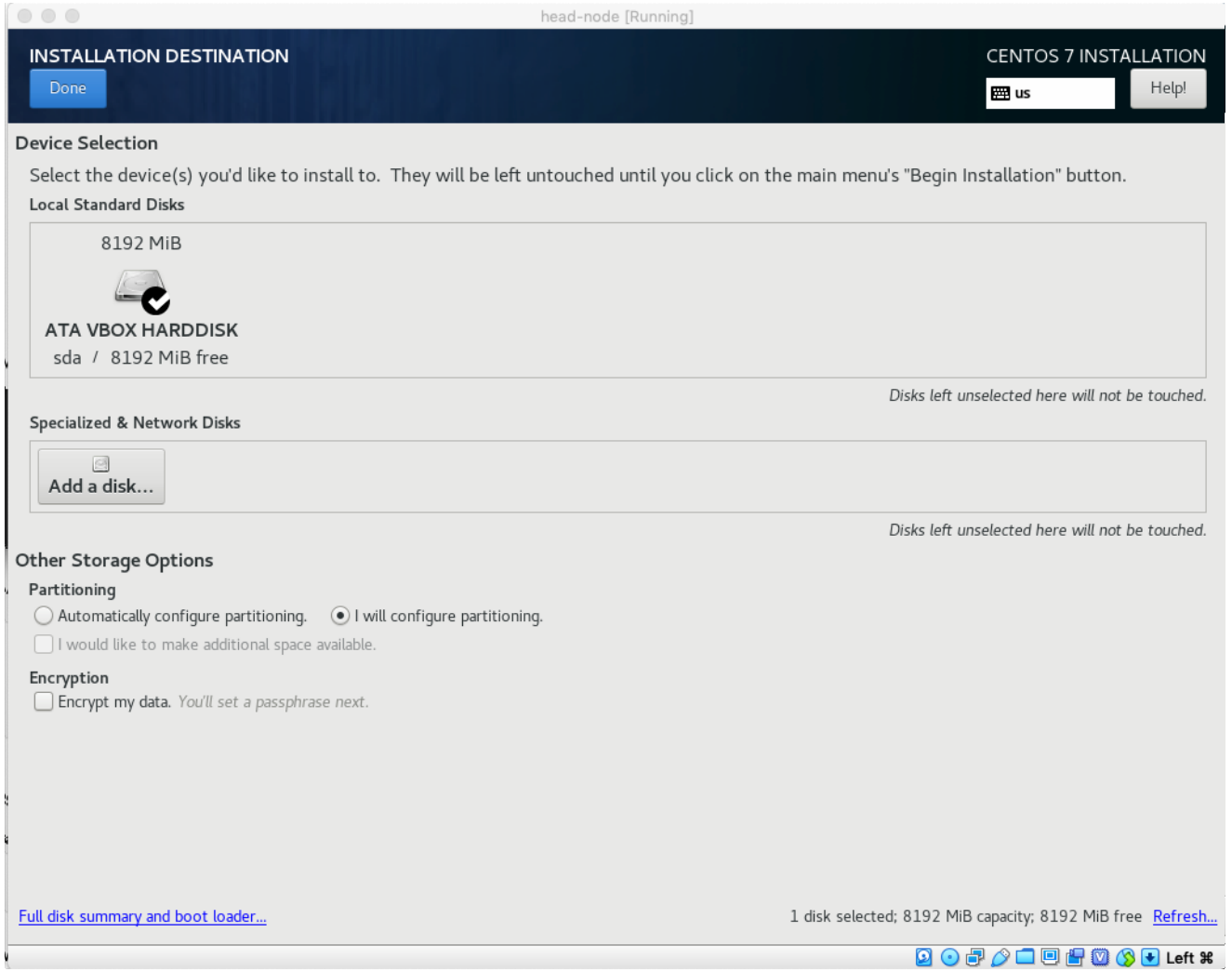
Installation Destination

Select the disk and select "I will configure partitioning" then click Done. We don't use automatic partitioning because we don't want to waste space on a swap partition.

Leave the partition type as LVM, and click the + button to manually create your partitions.

You'll need these partitions:

/boot/efi	50MB
/boot	976MB
/	7160MB



OPTIONS I USED FOR CENTOS 7.6

Networking Options

Set the hostname to head-node, then select the first ethernet adapter and select Configure

On General tab, enable auto connection and set priority to 0

On IPv4 tab, set Method to Manual, click the Add button and enter the address information you see here, set the DNS server, then save this interface (I was going to use DHCP, but VirtualBox gives problems with linked clones getting the same DHCP address despite having changed the MAC addresses on the clones)

Repeat on other 2 Ethernets, except

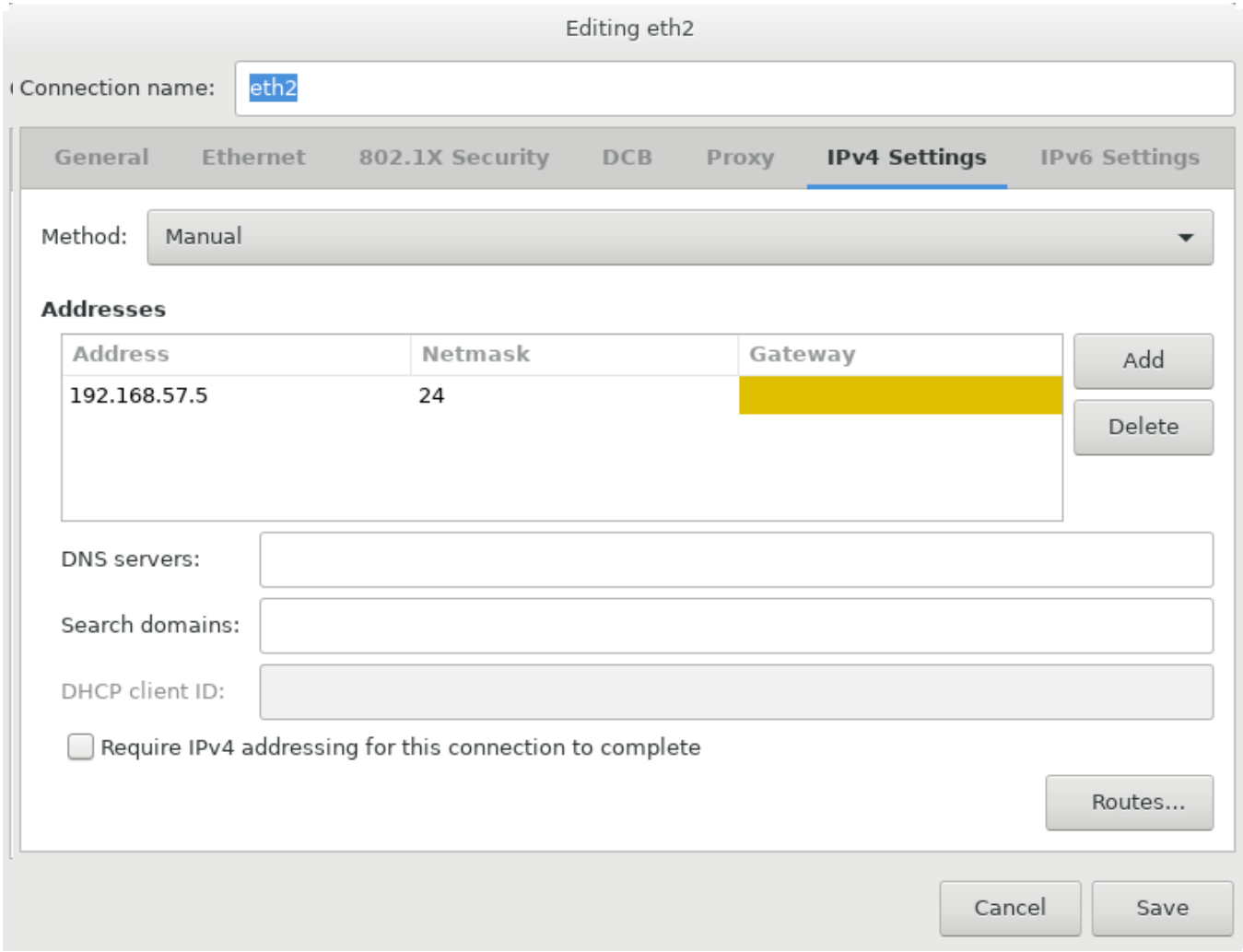
On General tab set priority to -1

For eth1 set the IP address to 192.168.56.5 and leave the gateway blank, we don't want to attempt to route out through this interface

For eth2 set the IP address to 192.168.57.5 and again leave the gateway blank

I also wanted to set the MTU higher on the RDMA interfaces, but VirtualBox limitations keep us from doing that.

Now begin the installation (Finally, right! :-))

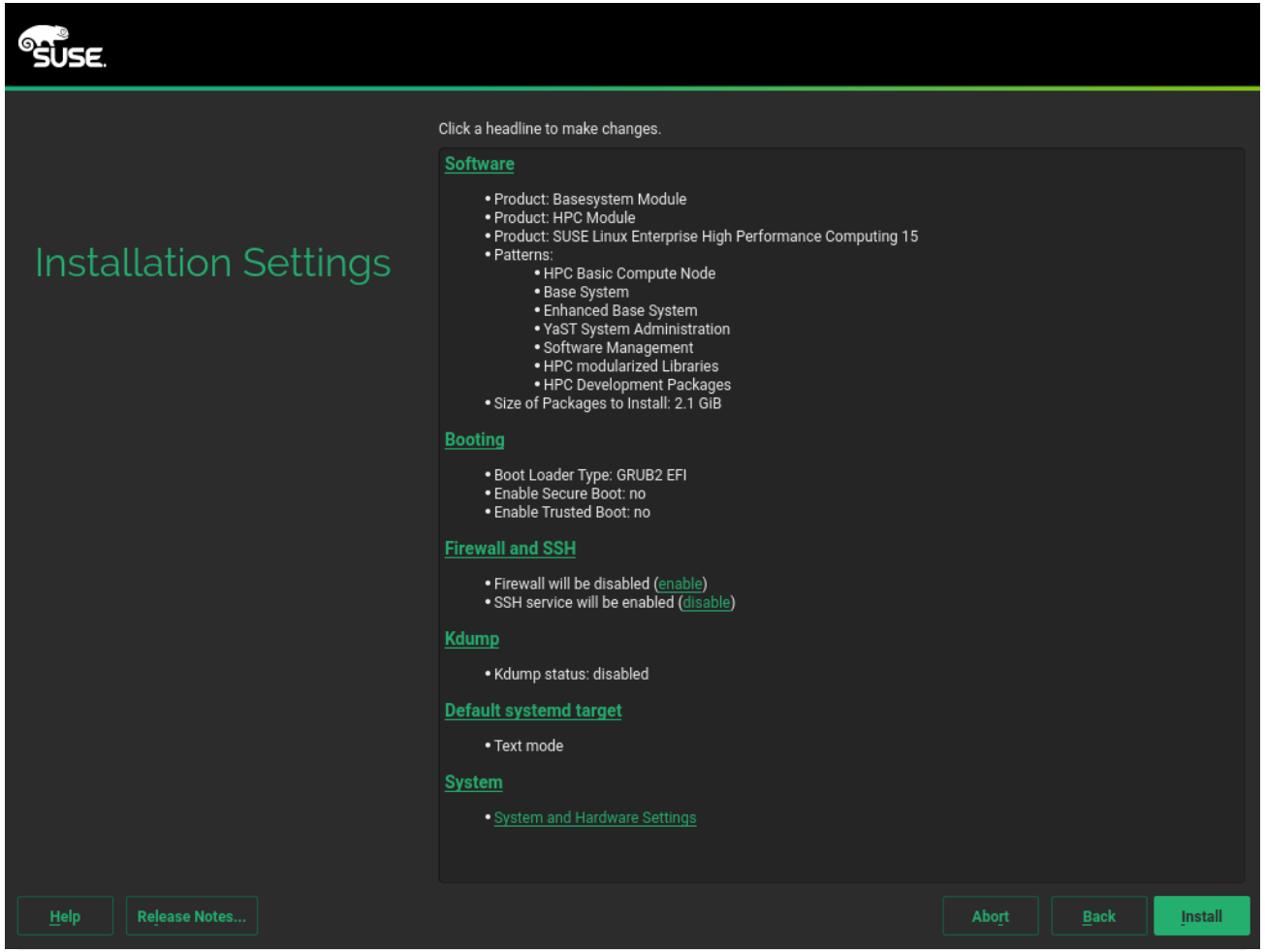


SPECIAL NOTES FOR OTHER DISTROS

openSUSE

While most things are fairly straightforward to translate from the CentOS instructions to other OSes, a few things deserve note:

1) **SuSE install happens in a different order, so software selection doesn't happen until after network interfaces have been set up. More importantly, you must watch the Device Name closely on SuSE as the Ethernet interfaces are not in the expected order.**
2) **Because of this, the Name and Device entries end up being skewed.**
3) **You also have a separate screen you need to go to in order to set the DNS resolution since we are doing static IP assignment (it isn't done on an interface's config screen like CentOS).**
4) **And also a separate screen for the default routing setup.**
5) **Unlike CentOS where you select Server by the install media, you select server during the install with SuSE.**
6) **Once you get to the Install screen, you can click on Software to modify the software selection prior to install.**
7) **SuSE HPC is mostly the same, with a prettier dark color scheme and more complete HPC patterns to select from.**



MUSICAL INTERLUDE WHILE EVERYONE INSTALLS THEIR HEAD NODES

EXTRA STEPS BEFORE THE CLONING PROCESS

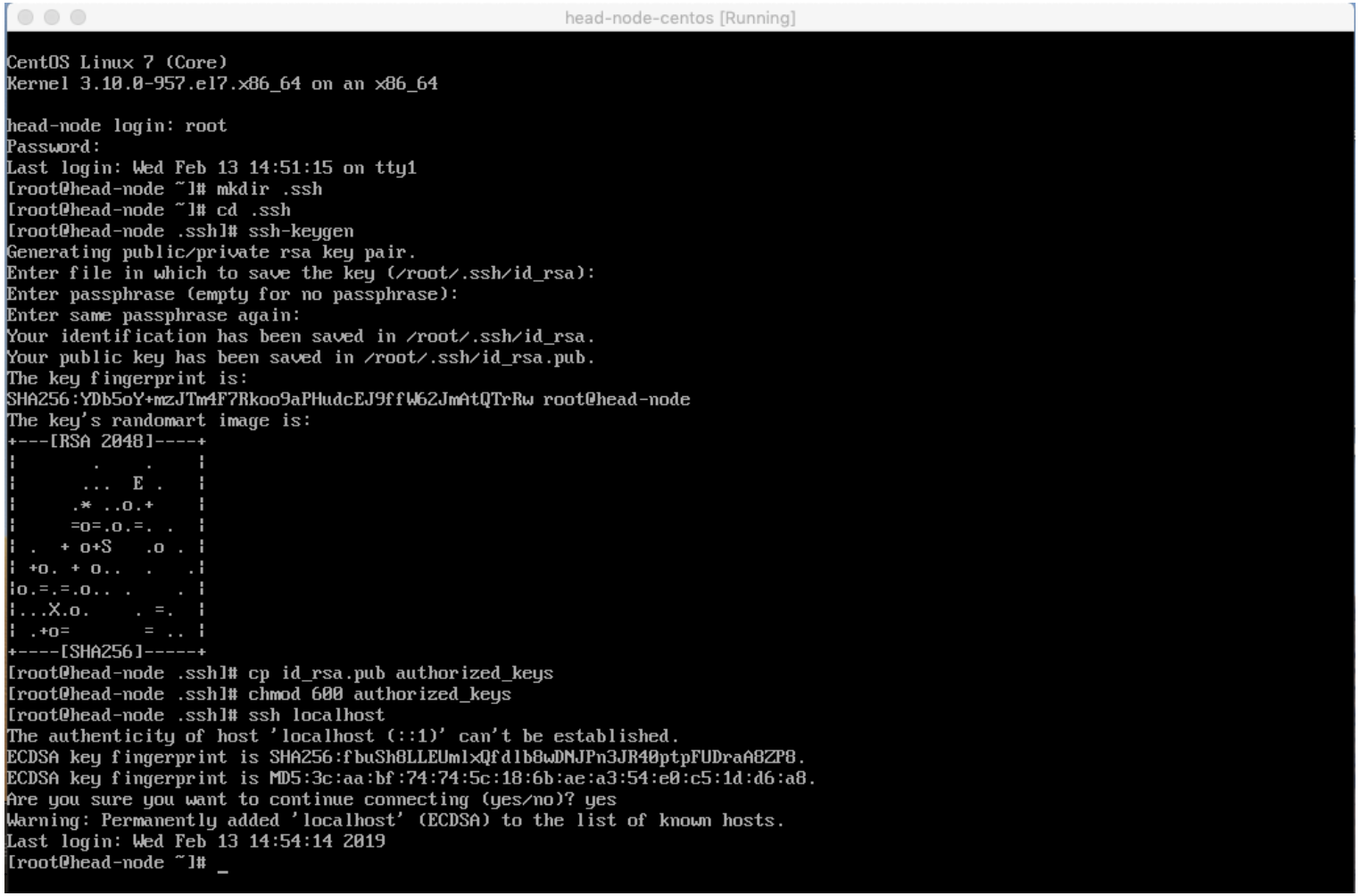
MAKING THE SYSTEM EASIER TO USE

Passwordless ssh setup for the root account

Login to your newly installed head node as root. This is a local, self contained cluster, so we really are only going to be using the root account.

Generate a ssh key and configure it for passwordless ssh login of the root account.

Try to ssh to localhost to verify the passwordless setup



MAKING THE SYSTEM EASIER TO USE

Getting all the software installed

There is a pretty long list of software we want to have installed. Some of it should have already been installed during the installation process, but we want to grab some of the optional, non-default packages too. If any of these packages don't exist on your distro, then no big deal, just use what is available.

- rdma-core
- libibverbs and libibverbs-utils (ibverbs and ibverbs-utils on Ubuntu)
- librdmacm and librdmacm-utils (rdmacm and rdmacm-utils on Ubuntu)
- perftest and qperf
- libfabric and fabtests (I couldn't find these on Ubuntu)
- openmpi and mpitests-openmpi (openmpi3 and openmpi3-testsuite on SuSE, Ubuntu has the openmpi package, but it's older at version 2, and there is no test suite available)
- mvapich2 and mpitests-mvapich2 (mvapich2 on SuSE, but no test suite that I saw, and not present on Ubuntu)
- nfs-utils (every OS had this installed already, but we do need it so I left it in the list)
- srp_daemon (or srptools on Ubuntu)
- targetcli (Ubuntu forked this and their version is now called targetcli-fb)
- rds-tools (if available, might not be, I only found it on SuSE)
- iperf3, sockperf, atop, fio, htop, iotop, bonnie++, dbench
- epel-release (only needed on CentOS to get the extra Fedora packages)

On CentOS, this will do the trick (if you installed the InfiniBand support group already):

```
yum install epel-release
yum install atop bonnie++ dbench fabtests fio htop iperf3 mpitests-mvapich2 mpitests-openmpi sockperf
```

Fedora is missing the mpitests package and also needs some additional packages installed, doing this gets the missing packages:

```
dnf install openmpi libibverbs-utils librdmacm-utils perftest qperf targetcli perl-Getopt-Long
```

On SuSE, start yast and go into Software Management, then search for packages. You can select a package, re-search for a new one, select it, etc. and then select Accept at the end and it will do them all. Selections are not lost by going back and searching for new items.

MAKING THE SYSTEM EASIER TO USE

Networking Tweaks

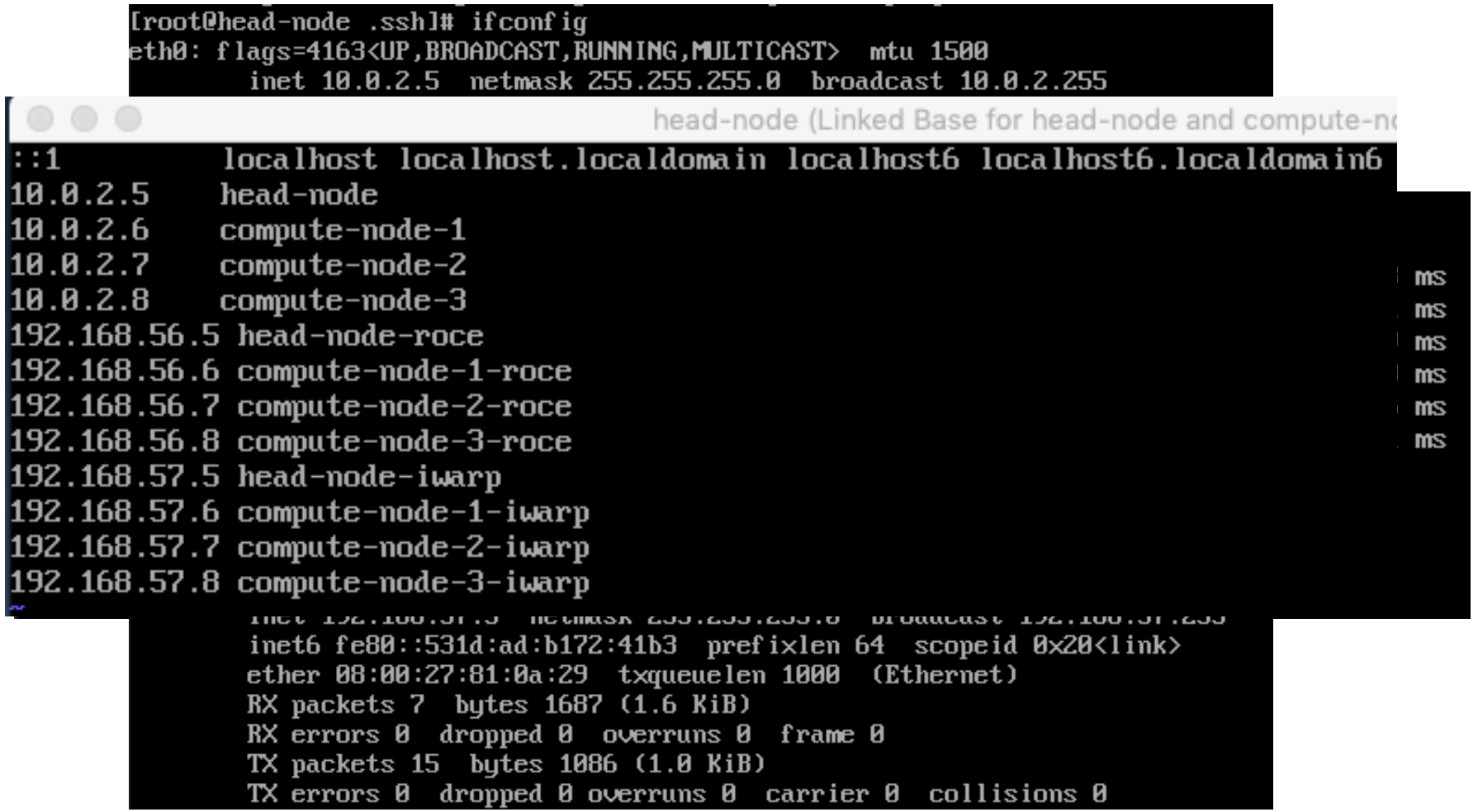
Disable the firewalld service (on SuSE you do this during the install, not now).

Verify that our Ethernet interfaces have come up with the desired IP addresses and such

Then I ran a quick ping test just to make sure we had properly working network connectivity to the outside world

Then I edited /etc/hosts

To add all of our static IP addresses



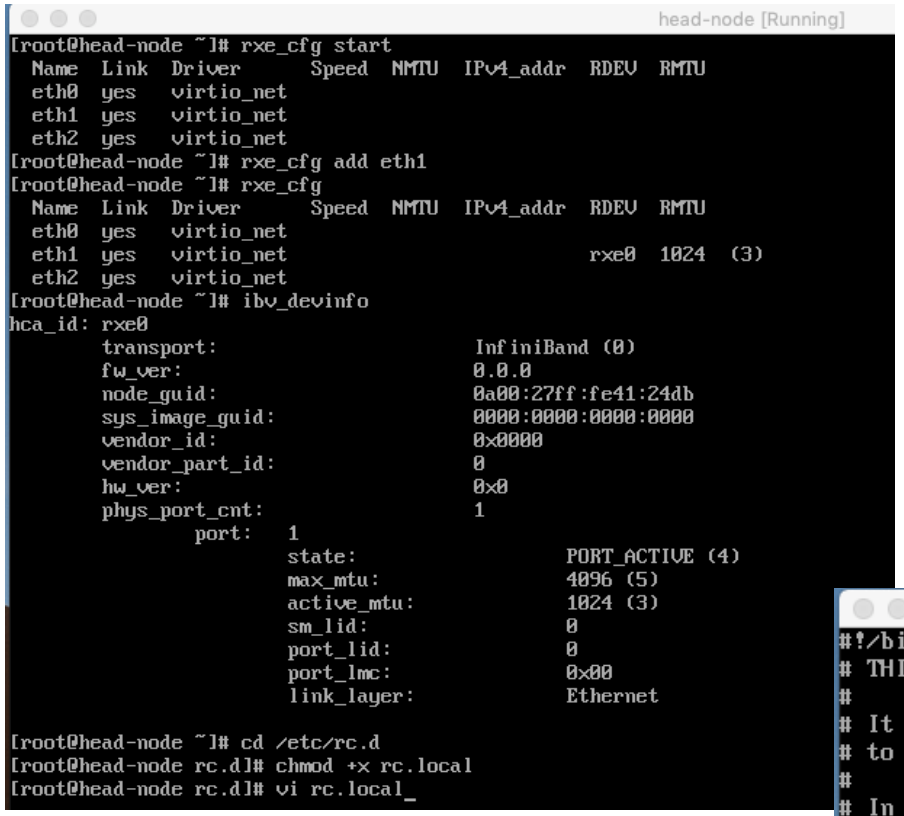
MAKING THE SYSTEM EASIER TO USE

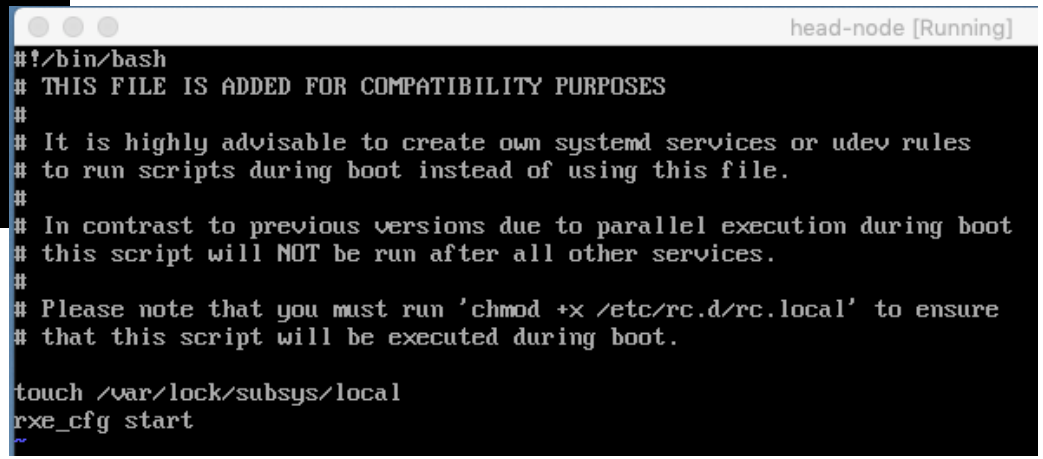
Set up our RoCE interface

We want our RoCE interface to come up on each reboot, so let's configure the rxe mapping now so it copies across to all of the clones.

We start the rxe subsystem, add our eth1 interface, verify that it is now enabled, check that the RDMA stack sees it, then in order to make this all come back on each reboot, we have to go into /etc/rc.d, chmod +x rc.local, edit rc.local to call rxe_cfg start, and that should do the trick. NOTE: On SuSE you need to add /etc/init.d/boot.local. But in all cases, we have to make sure that the file is executable, and in some cases we have to create it from scratch. NOTE2: It is disappointing that the replacement for rxe_cfg will not be ready in time for this presentation. I am knowingly feeding you stale information simply because it hasn't had time to trickle down from upstream into the distros yet. In the future, rxe_cfg will be replaced by using the rdma tool from the iproute2 package instead.

For good measure, reboot the machine once just to make sure that the rxe interface comes up automatically, then shutdown the machine for cloning.





THE CLONING PROCESS

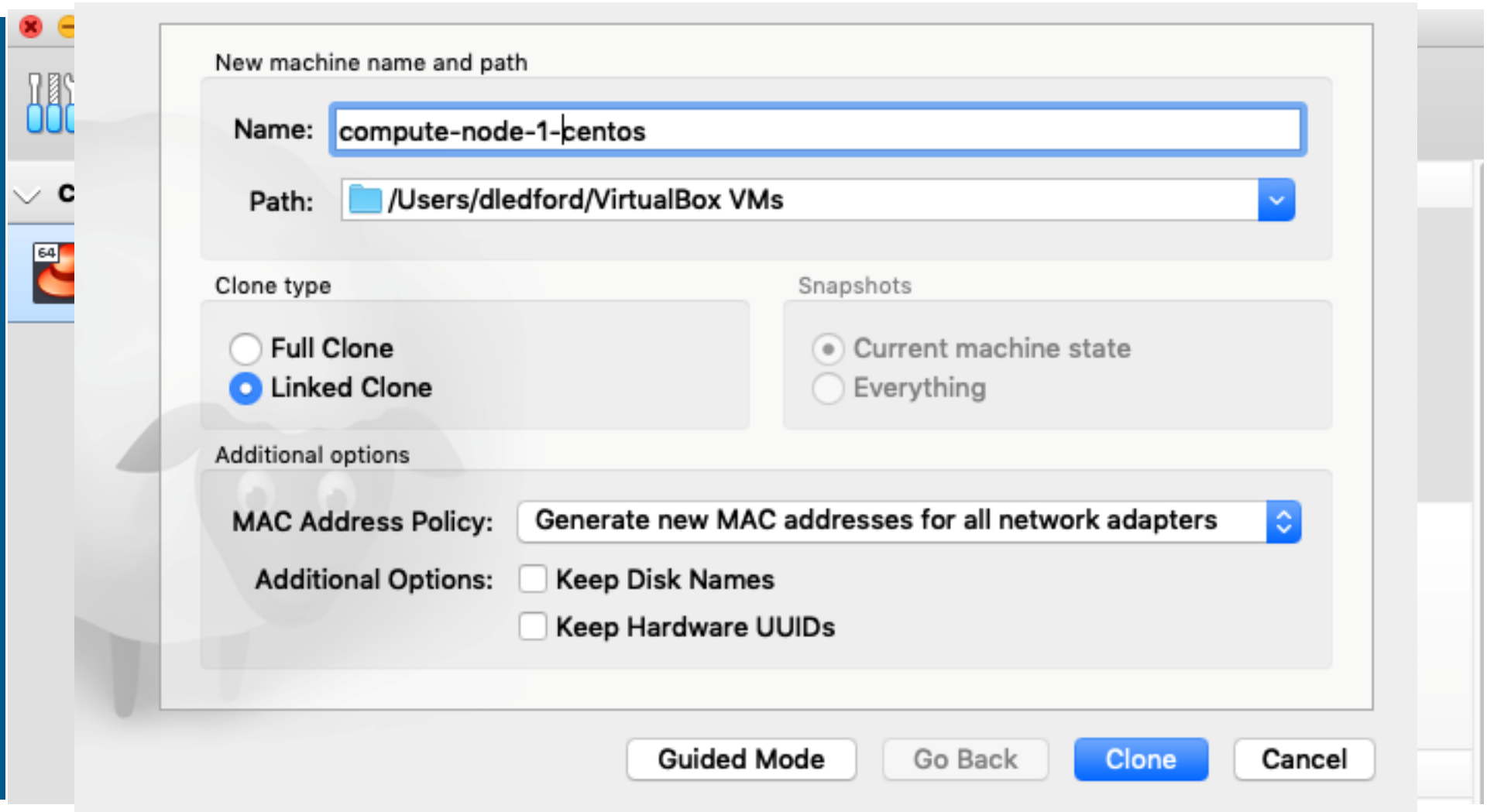
CLONING THE HEAD NODE

(and grouping the bunch)

Right click on the head-node machine and select Group. This will create a new group with your head-node in it. You can then right click on the group title and select rename to give it a useful name.

Then right click on the head-node machine again and select Clone. You will need to put the clone window in Expert mode like before with the New machine window. Change the name of the clone, select Linked Clone, and make sure to set the MAC address policy to generate new MACs for all network adapters, then click Clone to complete the operation.

Repeat this process to create compute-node-2 and 3 as well, using the head-node as the master for the clone each time (this makes the linked base tree



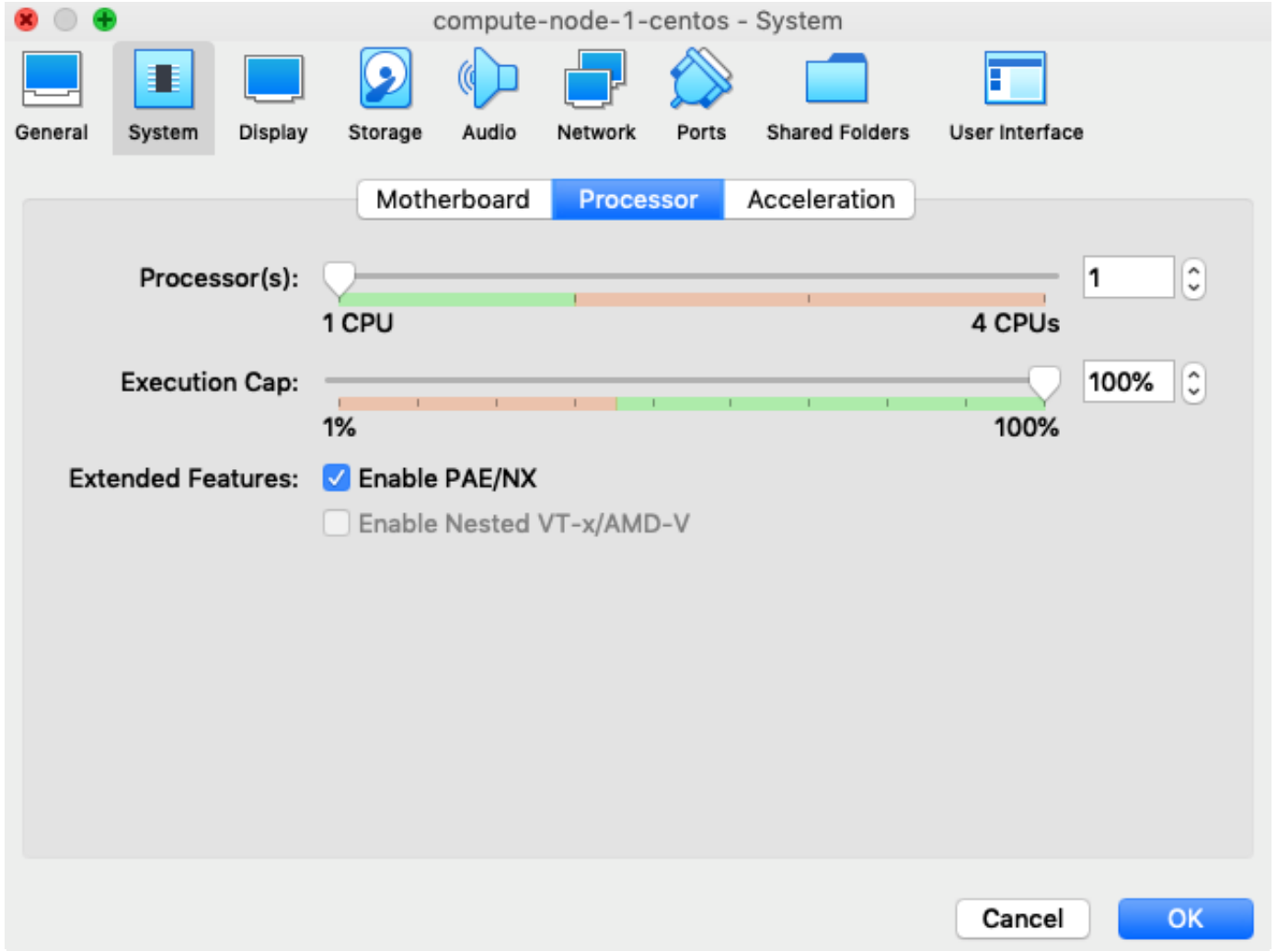
MODIFYING COMPUTE-NODE-1

Open the Settings on compute node 1, and change System/Motherboard to only have 2048MB of RAM

Change System/Processor to only have 1 CPU.

And if you added any Shared Folders to the head node, remove them from the compute node. We really want to get our data on our compute nodes from our head node, not from shared folders.

Repeat these steps on



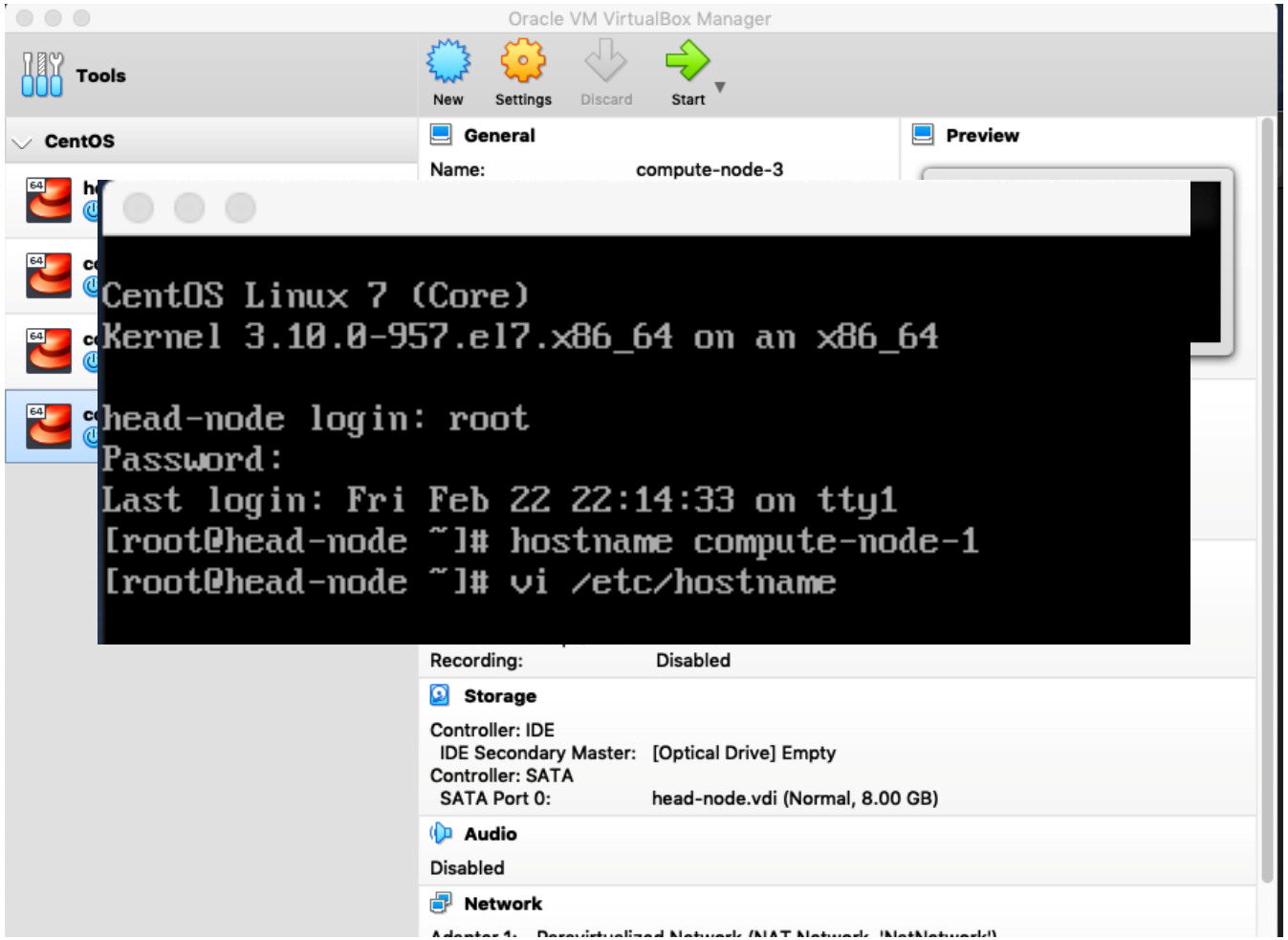
MAKING THE CLONES UNIQUE

CLONING COMPUTE-NODE-1

(linked clones)

You should now have a fully populated RDMA cluster group. Fire them up one at a time and change the hostname and network settings on the clones.

To set the hostname on CentOS, we need to set the new hostname using the hostname command (so it's effective immediately) as well as edit /etc/hostname (so it sticks after reboot).



CLONE SETUP TASKS

(linked clones)

NOTE: Do this one machine at a time, or else you *will* have IP address conflicts while trying to complete this. Shut each machine down when you are done. Once all machines have been done, and all of them shut down, you can power them all (including the head node) back up.

After setting the hostname, we need to modify our static IPs on each compute node.

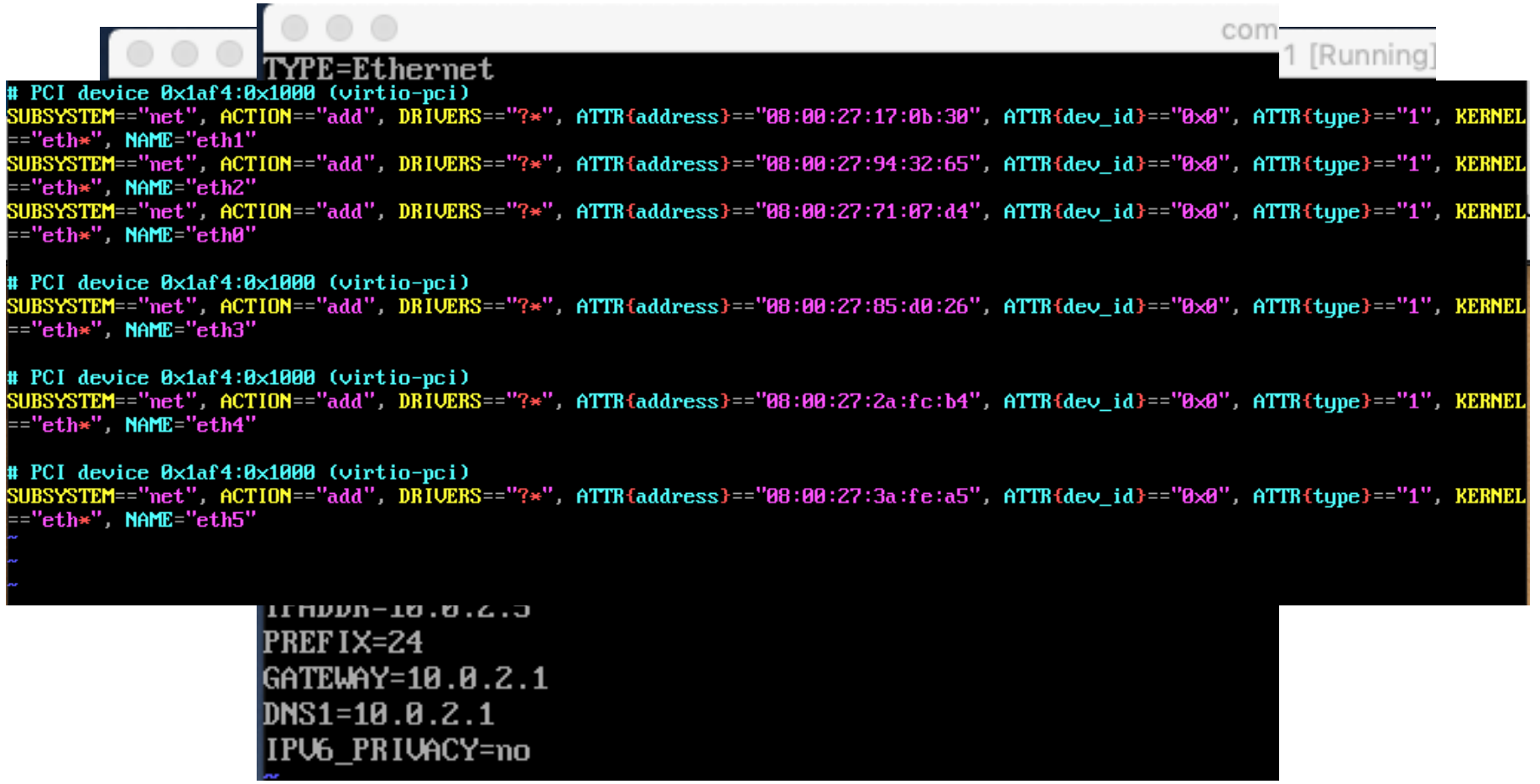
CentOS, Fedora, and SuSE all use the same config file syntax. Red Hat's stuff uses /etc/sysconfig/network-scripts and SuSE uses /etc/sysconfig/network. Also, the files may be named ifcfg-eth* or ifcfg-enp0s*, it's not important which they use.

Simply edit the files to change the IPADDR entry in each file to make the interfaces match what we used in the /etc/hosts file:

head-node: all interfaces IPADDR ends in .5
compute-1: all end in .6
compute-2: all end in .7
compute-3: all end in .8

For SuSE, editing these files is not sufficient to change the IP address. For SuSE there are two additional steps.

1) Find out how the OS mapped the new virtio devices to eth names. For instance, here we see that virtio0 ended up with the name eth4. We need to know what each virtio interface ended up with for the next step.
2) Edit the /etc/udev/rules.d/70-persistent-net.rules file and map the devices back to the right names: delete the old eth0/1/2 entries from the head node's interfaces, then change the name of eth4 to eth0 (at least that's how mine mapped out, but that's why we got the mapping in the previous step, you need to change the name for the entry of the virtio0 device to eth0, virtio1 to eth1, and so on)



CLONE SETUP TASKS

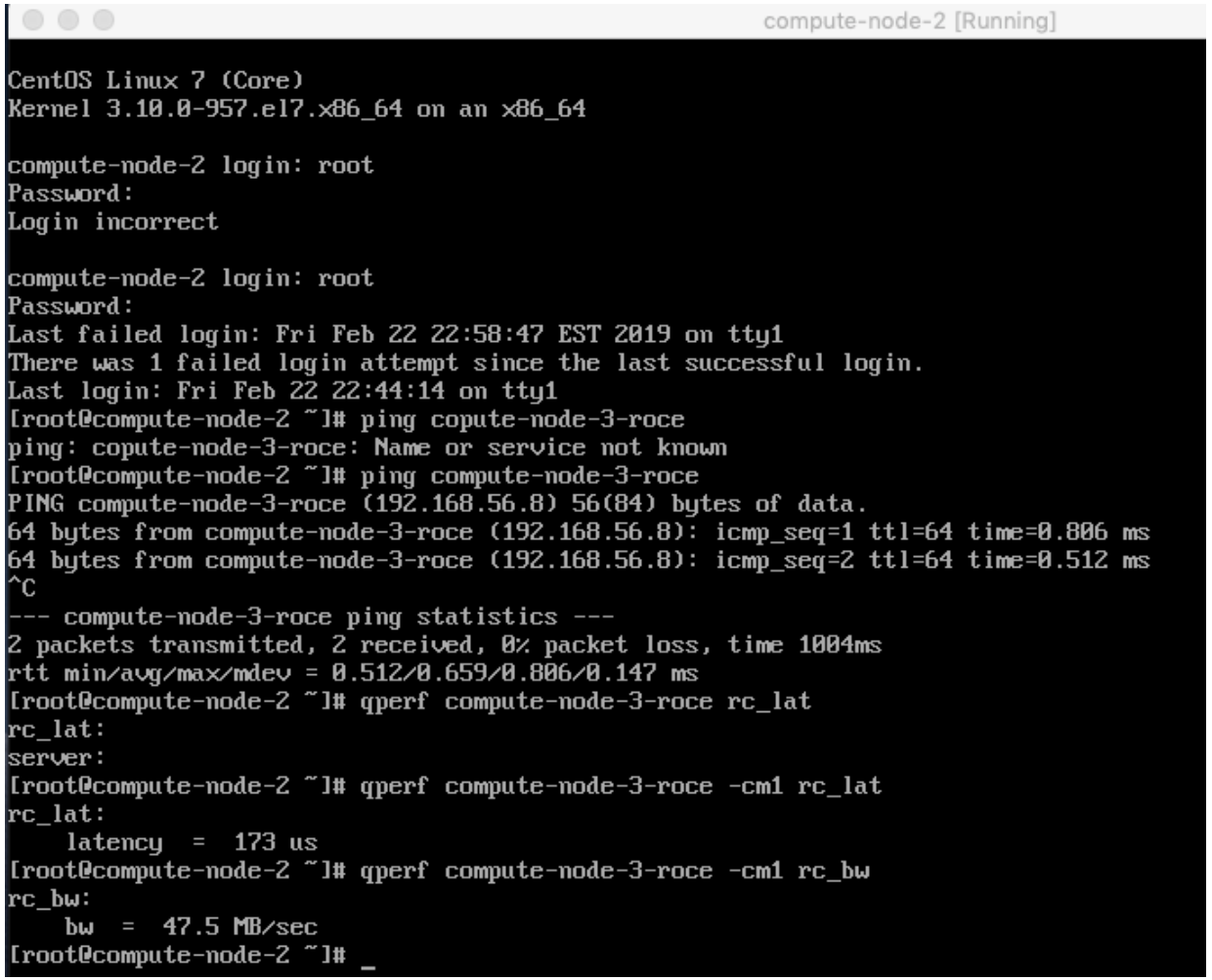
(linked clones)

With them all fired back up, let's run a quick test to make sure that we can ping the various machines/interfaces.

Here we can see we have connectivity from head-node to compute-node-3 on all interfaces, our /etc/hosts entries are working, and our rxe interface started automatically. We can do the same on compute-node-1 and 2.

We can even try our first RDMA related operation. Start qperf on compute-node-3 (no options, which is server mode), and on compute-node-2 run a couple tests. As you can see, we will need to use -cm1 on pretty much all tests with qperf under this setup, and the performance will not be stellar, but it is working.

At this point, we are ready for



15th ANNUAL WORKSHOP 2019

UNTIL TOMORROW

Doug Ledford