

Virtual Machines

The book companion website at <http://pearsonitcertification.com> contains three virtual machines so that you can easily get access to an environment where you can work on the labs and exercises in this book., you can set up an IPA server as instructed in Appendix D, "Setting Up Identity Management," and install your own environment according to the instructions in Chapter 1, "Installing Red Hat Enterprise Linux Server 7."

Before continuing, make sure that you have at least 3 GB of available RAM, and 30GB of available disk space on the computer that you're going to use as the host file.

After downloading the virtual machines from this book web site on pearsonitcertification.com, you'll have a file with the name OVA.zip. One file is provided that contains three virtual machines. After extracting this file, you'll find the three virtual machines in OVF format. These files can be used in VMware Workstation, VMware Fusion, Oracle Virtual Box and Linux KVM. VMware Player does NOT work; other virtualization platforms may work but haven't been tested.

Below you can read how to use the virtual machines in VMware Workstation/Fusion, Oracle Virtual Box and KVM on Linux. All descriptions assume that you have extracted the virtual machine files to a folder with the name OVA.

Note: Preferred platform

If you have access to VMware Workstation or VMware Fusion, use either of these. The virtual machines were created on VMware Fusion, so using them in either of these VMware products involves the least amount of additional steps. This software is not available for free though. If you want to use free virtualization software, use Oracle Virtual Box. KVM is supported as well, but only works if you have some Linux experience, as some Linux commands are involved in the KVM procedure.

Using the Virtual Machines on VMware Workstation and Fusion

1. Before doing anything, you **MUST** make sure that networking in VMware Workstation matches the networking in the virtual machines. Open VMware Workstation and select **Edit > Virtual Network Editor**. This brings up the screen that you can see in figure1.

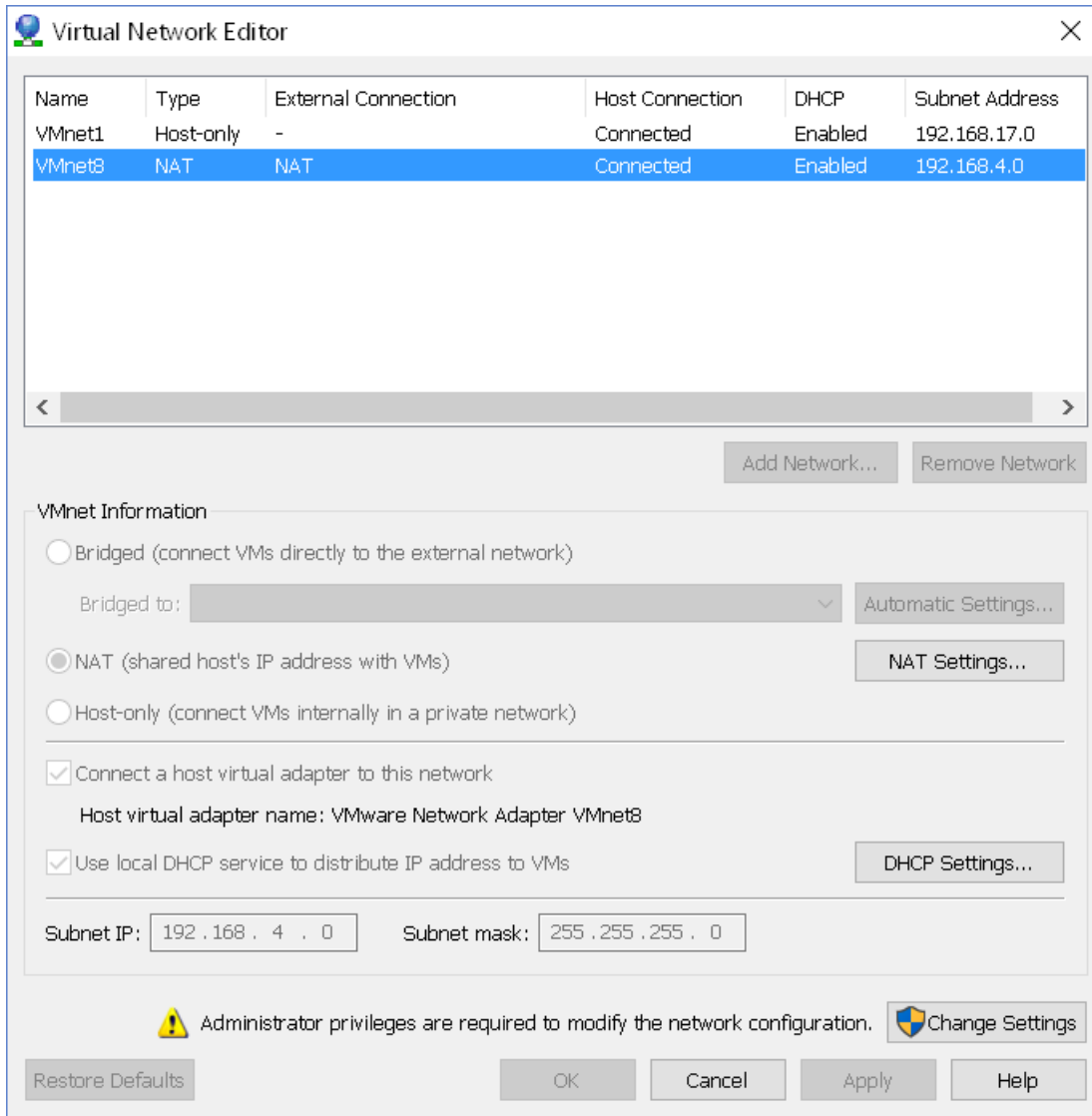


figure1: setting VMware workstation network properties

2. Verify the subnet address that is set for VMnet8. If it is not set to 192.168.4.0/24, click the Change Settings button in the lower right of the Virtual Network Editor and click Yes when User Account Manager asks if you want to allow changes to be made.
3. Click the VMnet8 line and change the Subnet IP to 192.168.4.0 with the subnet mask 255.255.255.0. Click OK next, you can now proceed importing the virtual machines.
4. In Windows Explorer, open the OVA folder. It contains 3 directories, each of which contains the virtual machine files. In this procedure I'll show you how to open the LabIPA-small VM, the procedure is the same for other virtual machine platforms.
5. Select the Labipa OVF file, and right-click it with your mouse. From the pop-up menu that appears, select **Open With > VMware Workstation**.

6. In the **Import Virtual Machine** screen, you can now enter a name for the new virtual machine, and specify a Storage path. Notice that within the storage path, you need at least 30GB of free disk space. After specifying the storage location, click **Import** to start importing the virtual machine. This will take a couple of minutes.

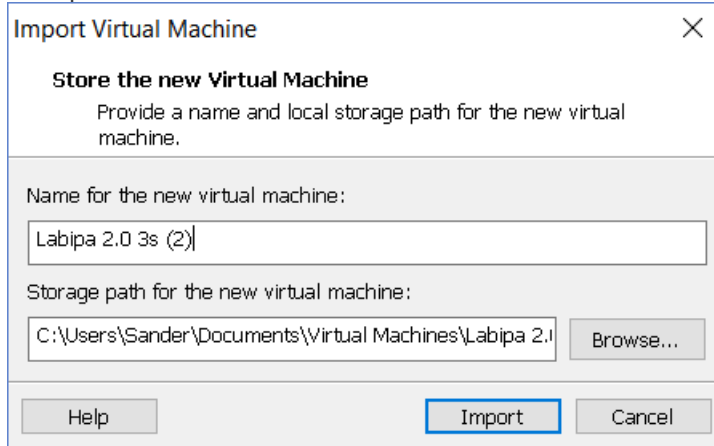


Figure2: Specifying where to store the virtual machine file

7. The virtual machine will now be imported. Repeat this procedure for all other virtual machines and move on to the section "Verifying the virtual machines working" in this document.

Using the Virtual Machines on Oracle Virtual Box

Before using the Virtual Machine files on Oracle Virtual Box, you need to run a few additional steps. Even though Oracle Virtual Box is available for Windows, Mac as well as Linux, you should run these additional steps on Linux.

1. After extracting the virtual machine files, use an ASCII text editor to open the ovf file for each virtual machine and replace the word **ElementName** with **Caption** on every occurrence. Also replace **vmware.sata.ahci** with **AHCI**. Next, run **sha1sum <FILENAME>.ovf** on each file to make them ready for import. The sha1sum command is available by default on Linux. Microsoft provides this command as an unsupported command for Windows as well. Consult microsoft.com for more information about it.
2. Open VirtualBox and select **File > Preferences > Network**. This opens the screen that you'll see in Figure 3.

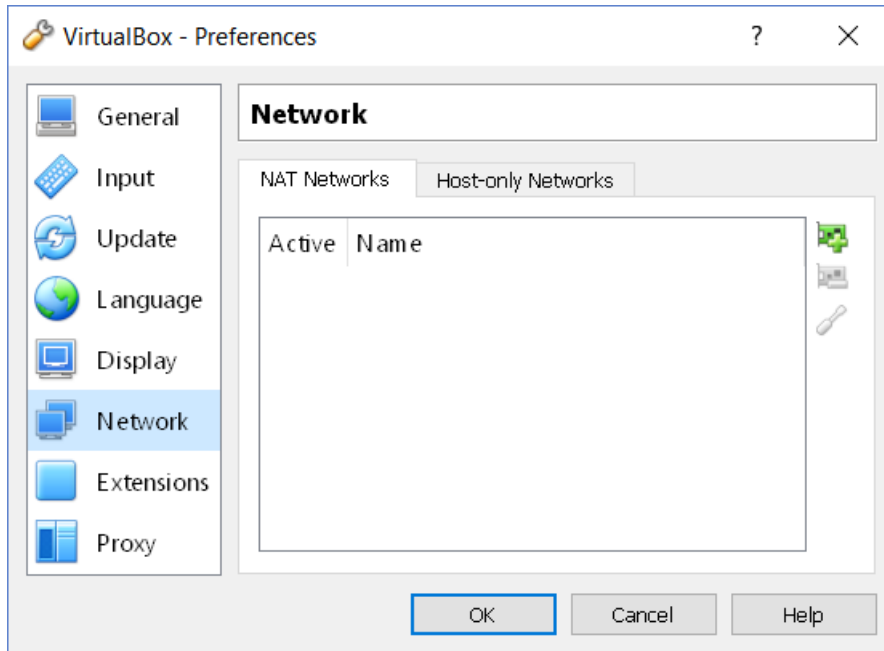


Figure 3: Creating Virtual Box networking

3. Click the icon to the right on top. This adds a new network with the name NatNetwork. Select this laptop and click the Edit button (the lowest icon on the right of this window). Make sure that the Network CIDR, which by default is set to 10.0.2.0/24 is changed to 192.168.4.0/24 and looks as in Figure4. Click **OK** after setting the Network CIDR.

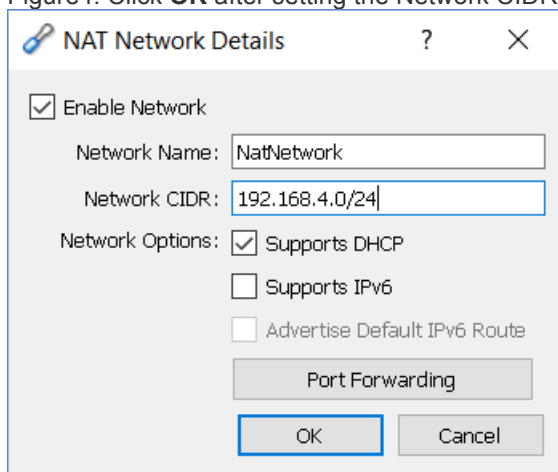


Figure4: Setting the required network

4. In your operating system file manager (Windows Explorer, Mac Finder or Linux Nautilus), right click the OVF file in the directory that contains the extracted virtual machines and select **Open With > Virtual Box Manager**. If at this point you're getting an error mentioning *Host resource of the type "Other Storage Device (20) is supported with SATA AHCI controllers only*, make sure that you have applied step 1 of this procedure correctly.

5. After importing the virtual machine successfully, Select **Network** from its properties. This opens the window that you can see in Figure 5. Make sure that Adapter1 is attached to Nat Network, exactly as you can see it in Figure 5 of this procedure.

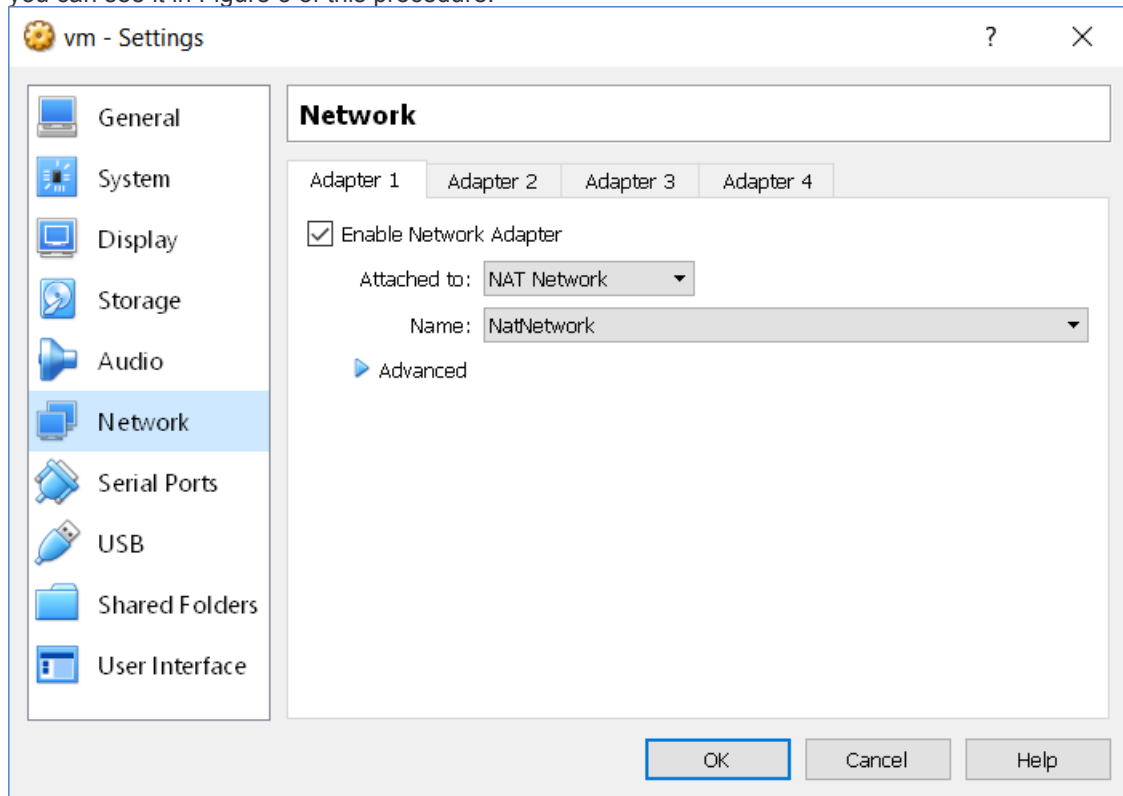


Figure 5: Make sure the network adapter is connected to the NAT Network

6. Boot the virtual machines and continue to the section "Verifying the Virtual Machines Working" later in this document.

Using the Virtual Machines on KVM on Linux

Notice that applying this procedure requires Linux skills. If you're new to Linux, do not use the virtual machines on KVM on Linux, but use another platform.

1. After extracting the virtual machine files, open a root shell.
2. Use **cd** to go in to the directory that contains the virtual machine disk files. These are the big files that have the .vmdk extension.
3. Convert each of the VMDK disk files, using **qemu-img convert -O qcow2 <FILENAME>.vmdk <FILENAME>.qcow2**
4. After converting the files you need to make sure that KVM networking matches networking in the virtual machines. Use **vim /etc/libvirt/qemu/networks/default.xml** and use **:%s/122/4/g**. This changes the default KVM network of 192.168.122.0/24 to the 192.168.4.0/24 network address.

5. Also make sure that the default router as specified in this file is set to 192.168.4.2 and not 192.168.4.1.
6. Restart your KVM host.
7. Start the graphical **virt-manager** utility and click **New > Import existing disk image**. Next, click

Forward.

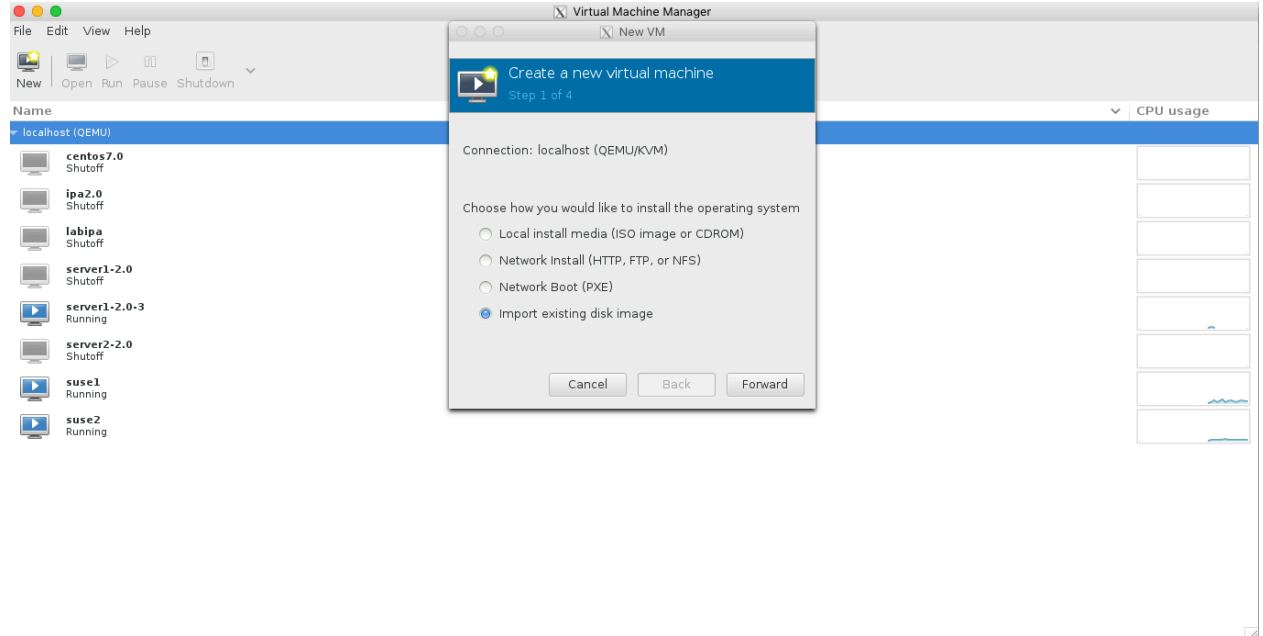


Figure 6: Importing an existing virtual machine

8. You now have to define the virtual machine properties. To start with, click **Browse** to select the location where the qcow2 file that you have just created is stored and select this file. Click **Open** to open it and set the OS type to Linux. Next, click the Forward button.

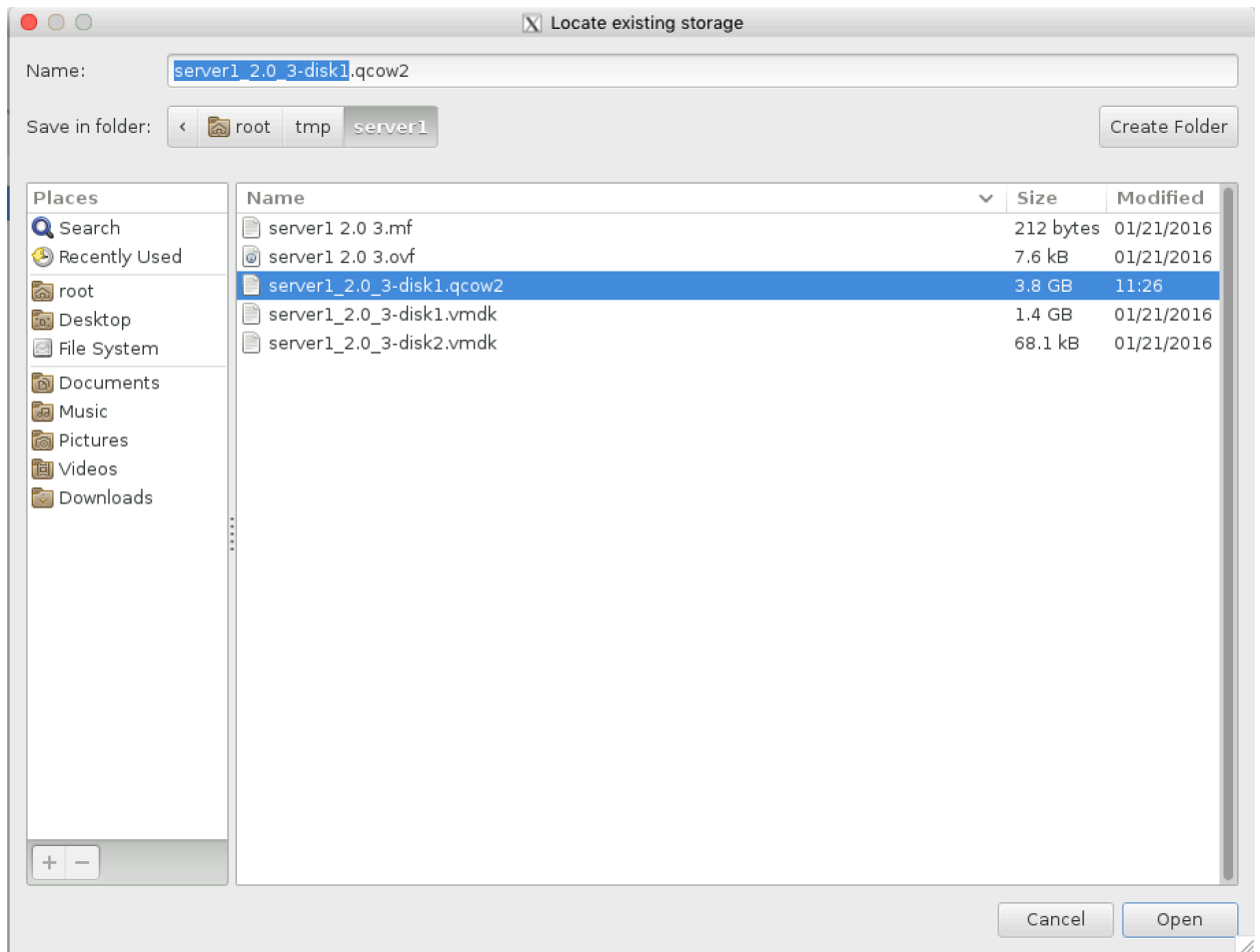


Figure 7: Opening the disk file

9. In the next screen, click the Forward button again.
10. In the final screen, specify the name that you want to use for this virtual machine and click **Finish**. The Virtual machine will now be started.
11. Repeat this procedure for all other virtual machines. After doing so, move on to the section **Verifying the virtual machines working**.

Verifying the virtual machines working

If you have made sure that networking configuration is set correctly, you'll now have the virtual machines up and running in your environment. Before starting to use the virtual machines, log in to the **labipa.example.com** VM and perform these tests. On all virtual machines there is a local user with the name **user** and the password set to **password**. The password for the root user is set to **password** as well :

1. Ensure that networking is working, using **ping 192.168.4.2** (which tests access to the default router) and **ping google.com** (which tests internet access). If you fail these tests, you have an issue that is

probably related to networking in your virtualization software. Make sure that you have successfully applied all steps mentioned in this document.

2. On labipa.example.com, open a root shell and type **ipactl status**. This should show that all FreeIPA components are operational.
3. From server1.example.com and server2.example.com, type **ping 192.168.4.200**. This should give answers from the labipa.example.com server.
4. From server1.example.com, type **ip addr show**. This should show the current IP address, which should be set to 192.168.4.210. Repeat this procedure on server2. The IP address should be set to 192.168.4.220. If this is not the case but you successfully passed all other tests, this is not an issue. In chapter 8 you'll learn how to set IP addressing on Linux.

If everything fails

The procedures that are described in this document have been tested a lot. But still, you might find yourself in a situation where it isn't working. If this is the case, you can read **Appendix D** with this book to set up an IPA server yourself. Installing server1.example.com and server2.example.com is easy. Just follow the generic installation instructions as described in chapter 1 of this book.