

## NFJS Beyond Kubernetes Extended Workshop – Required preparation before session Setup for Virtual Machine Environment Option

Version 1.2 - 7/3/22

Provided by Tech Skills Transformations

**Note: This environment is the simplest to use/setup if you can run VirtualBox and load/run the VM.**

Note that if you encounter problems with that, often it may be due to running another application on your machine that is already using the virtualization subsystem, such as Docker for Desktop or Parallels. You can sometimes get things to work by stopping those applications and any part of them that is running on your system. In some cases, a reboot afterwards may help - remembering to stop/not restart the other applications that may use the virtualization subsystems. You can also Google for particular error messages you get when trying to run VirtualBox. [This document](#) has some other pointers that may help if you are running into issues.

Alternatively, if you cannot get VirtualBox or the VM to run well on your system, then you can use option 2, a user-supplied environment, as detailed in the [beyond-k8s-setup-user-supplied-environment-option](#) doc.

1. Install the [VirtualBox](#) application from VirtualBox.org
2. Ensure that the VirtualBox application starts and runs on your system.

Note: If you encounter an error starting VirtualBox, solutions to many common problems can be found via Googling the error along with the Operating System you're running on. Some troubleshooting tips are also included in a doc at <https://github.com/brentlaster/safaridocs/blob/main/vbtroubleshoot.pdf>

3. Download the pre-configured VirtualBox image (.ova file) for the class from either of the two locations below. (Note that this file is over 3G in size, so it may take a while.):

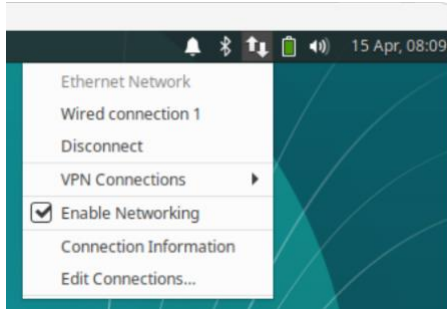
- <https://www.dropbox.com/s/ide15512um4c9kc/beyond-k8s.ova?dl=0>
- <https://bclconf.s3.us-west-2.amazonaws.com/beyond-k8s.ova>

### **Checksums if you want to check your file downloaded correctly:**

- **MD5 Checksum:** a729a28b1da4ebdd3cf01d9cf127b1e8
- **SHA-1 Checksum:** 293954020c6dc2591ba73edecc0fcdc53a290f10
- **SHA-256 Checksum:** 6a70c682ad98f56386e12c98bdf273be989cddb2e07775d4b5477e3d0a60f016
- **SHA-512 Checksum:**  
efa605c303bffa071f9de5c9c746cc1b30219dc9afef4927c2f1842916bd5f60e1cc46089a61db9dde524d3d684067657a40548acd4c0f65feb0597b30e54c36

4. Load the file into VirtualBox. If you need help getting the file loaded into VirtualBox, reference the **“Additional details for loading image”** section further down for details on how to proceed with loading the file into VirtualBox if you need those.

5. Start the virtual machine in VirtualBox (see setup doc) and enable networking via the "arrows" icon in the upper right (see screenshot below).



6. Open a "Terminal Emulator" session on the desktop and start the virtual machine with the command below. Wait for this to complete.

```
$ sudo minikube start --vm-driver=none
```

7. Get the latest workshop files.

```
$ cd ~/beyond-k8s
$ git pull
```

8. Now run the script below. This will setup ArgoCD, Tekton, Helm, Istio, etc. and pre-pull images needed for the workshop, so they are present already on the machine and don't have to be downloaded again later.

```
$ ~/beyond-k8s/extra/setup.sh
```

(This will also output the ArgoCD password at the end of the script. You will need this in one of the labs. If you don't see it, the process may not be ready and you can run the script again and it will likely show up. (You can also get this again later if you need it.))

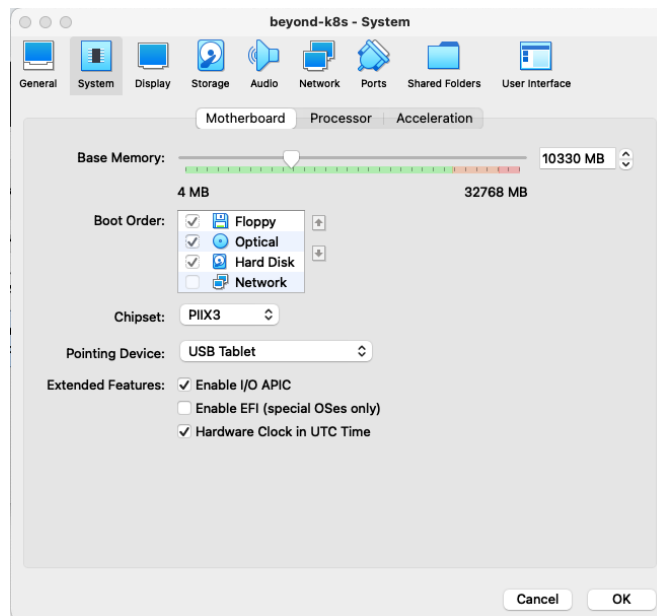
8. After running the script, you should see a set of namespaces like the list below. (On this machine, we have aliased "kubectl" to just "k".

```
$ k get ns
NAME          STATUS AGE
argocd        Active 3m3s
default       Active 27m
istio-system  Active 3m27s
kube-node-lease Active 27m
kube-public   Active 27m
kube-system   Active 27m
tekton-pipelines Active 3m
```

9. After going through these steps, you can Close and Power Off the machine, but leave it loaded in VirtualBox in prep for the class.

10. NOTE: As an additional check, you can power it back on once to make sure it comes back up as expected.

11. OPTIONAL: If the machine seems to be slow or taking up too many resources on your system, you can adjust the amount of memory, etc. via right-clicking on the machine in the list in VirtualBox, by selecting Settings and then adjusting memory, etc. as shown in the example below.



### Additional Details for Loading Image (if needed)

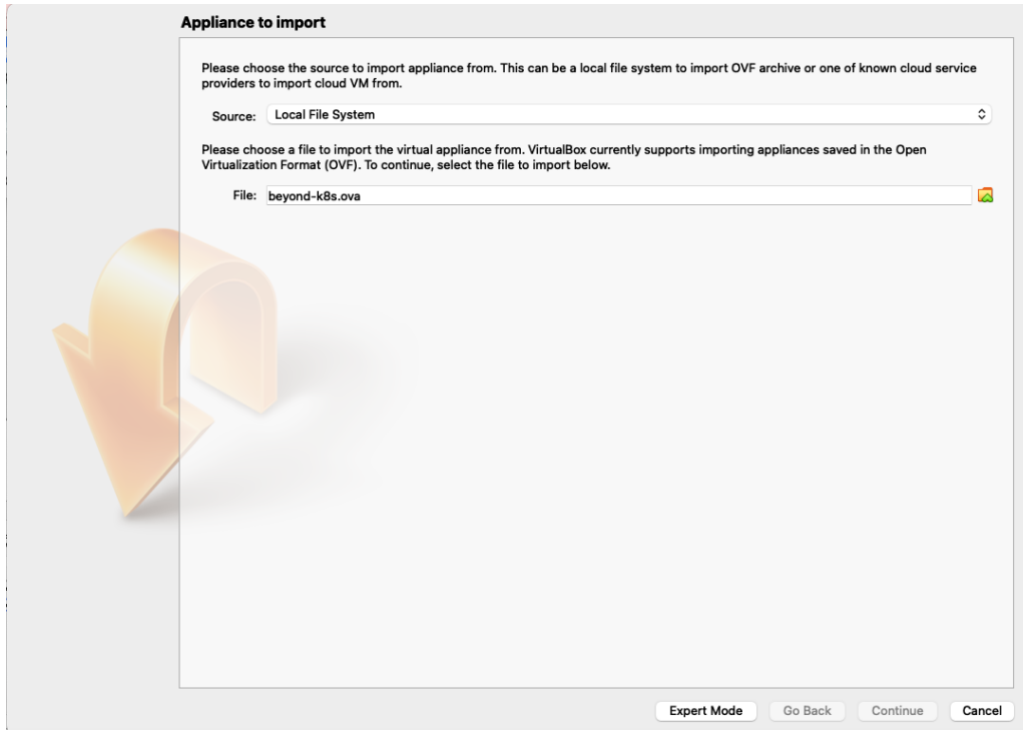
1. Once you have Virtualbox installed and the image downloaded and are ready to proceed, do the steps below to import the appliance.

a. Open **VirtualBox** on your system.

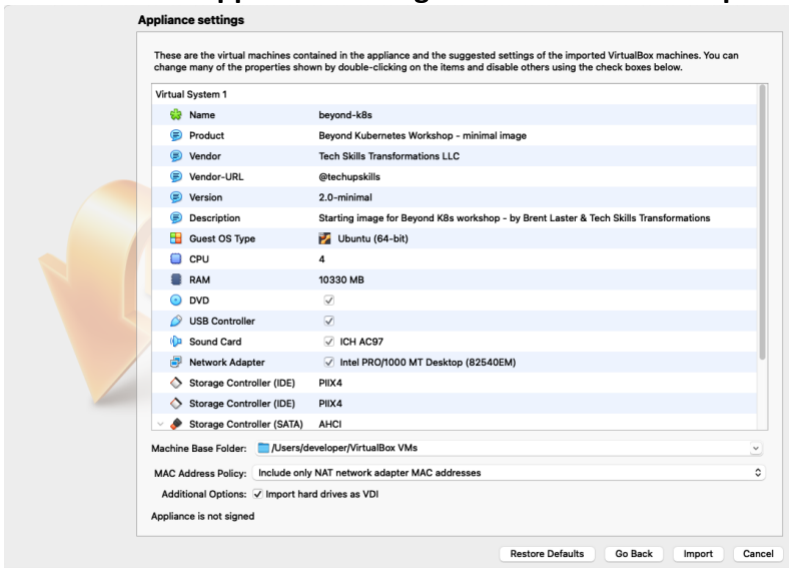
b. From the **File** menu, select **"Import Appliance..."**.



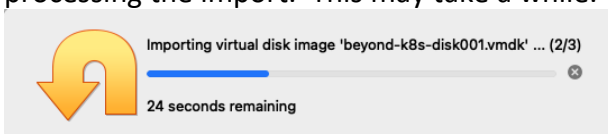
c. From there, you can type in (or browse to, using the folder icon) the path of the **beyond-k8s.ova** file. Then click **Next**.



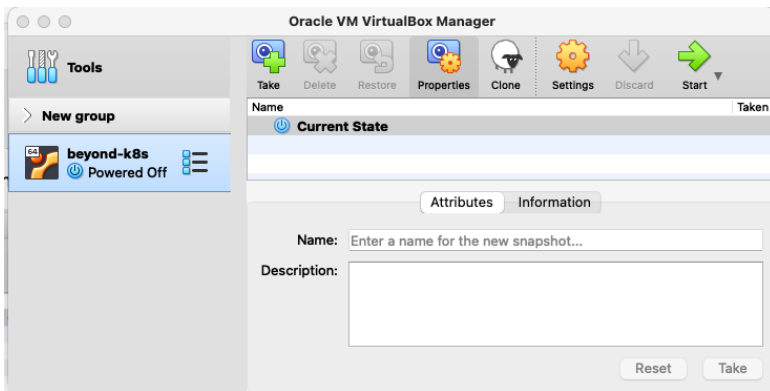
4. On the next screen, if given the option, click the box to reinitialize the MAC addresses. You can just accept the rest of the **Appliance Settings** and then click the **Import** button.



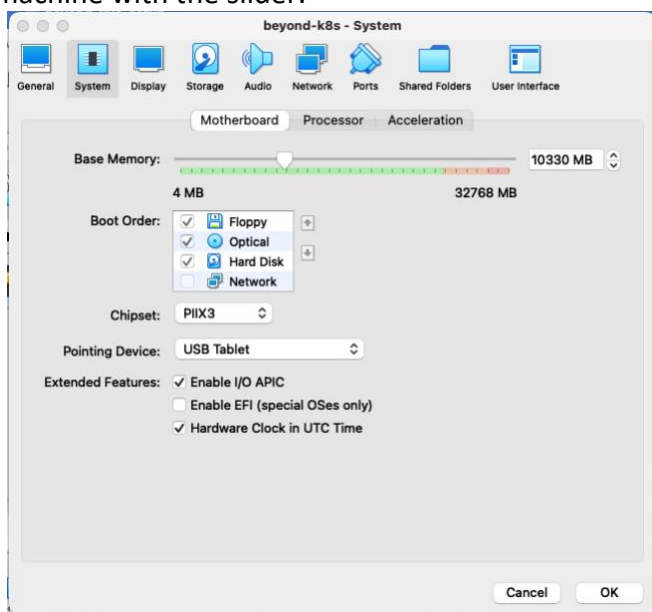
5. You will get a pop-up box for the “license” info. Just click the **Agree** button. Your system will then start processing the import. This may take a while.



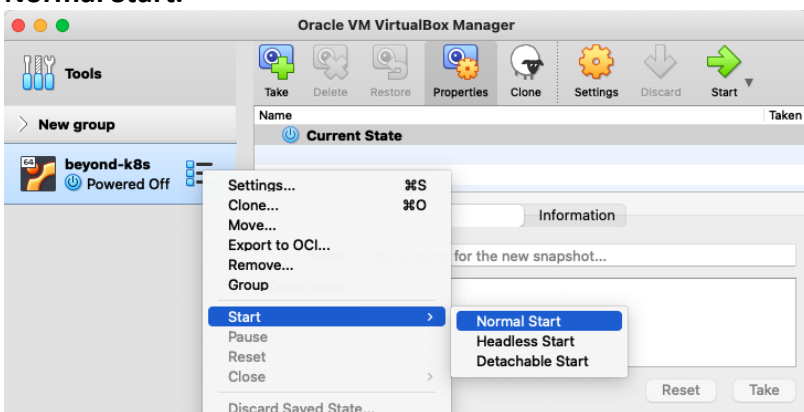
6. After the import is finished, you should have a VM listed in VirtualBox named **beyond-k8s**.



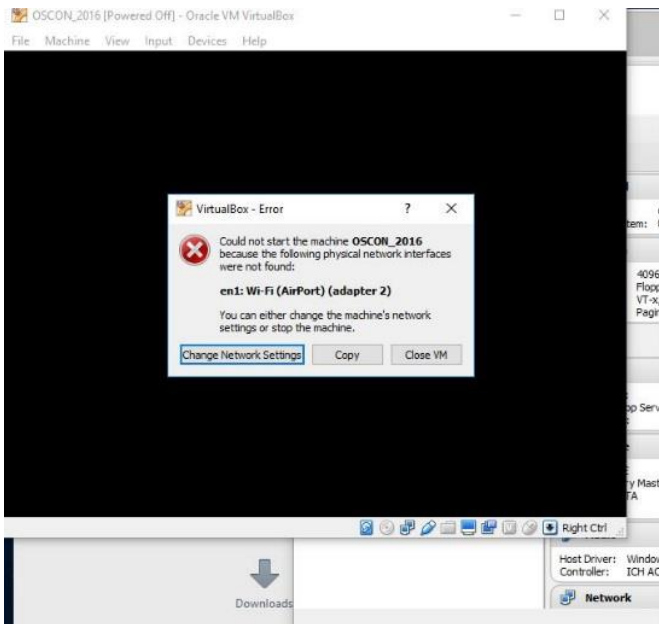
7. **(OPTIONAL)** At this point, depending on the settings of your physical system, you can adjust the amount of memory for the image if you need/want. You can do this by clicking on the **Settings** icon in the menu bar, then **System** in the pop-up box for the settings. Then you can adjust the amount of memory for the virtual machine with the slider.



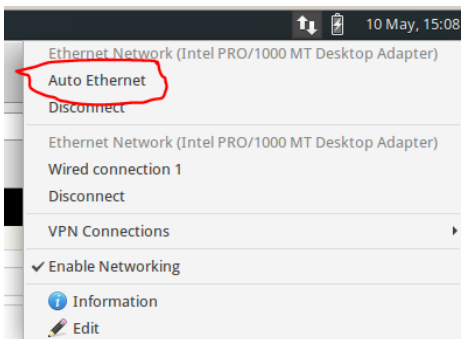
8. At this point, you can start up the virtual image by right-clicking on the image name and then selecting **Normal Start**.



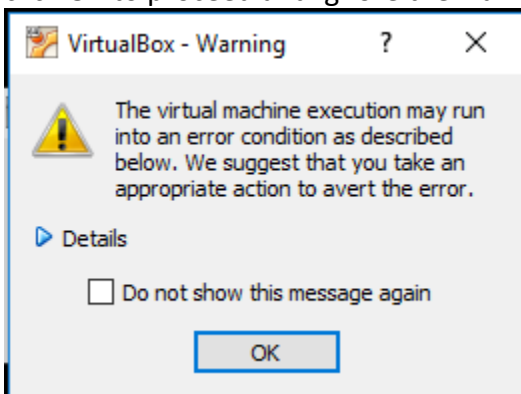
9. You may receive an error about network adapters here similar to the one below.



If so, just click on the option given to “Change Network Settings”. Then click on “Ok” in the network settings. ( After the system starts, you can also enable enable “**Auto Ethernet**” in the networking menu (click on double arrows in upper-right corner) but that is probably not necessary).



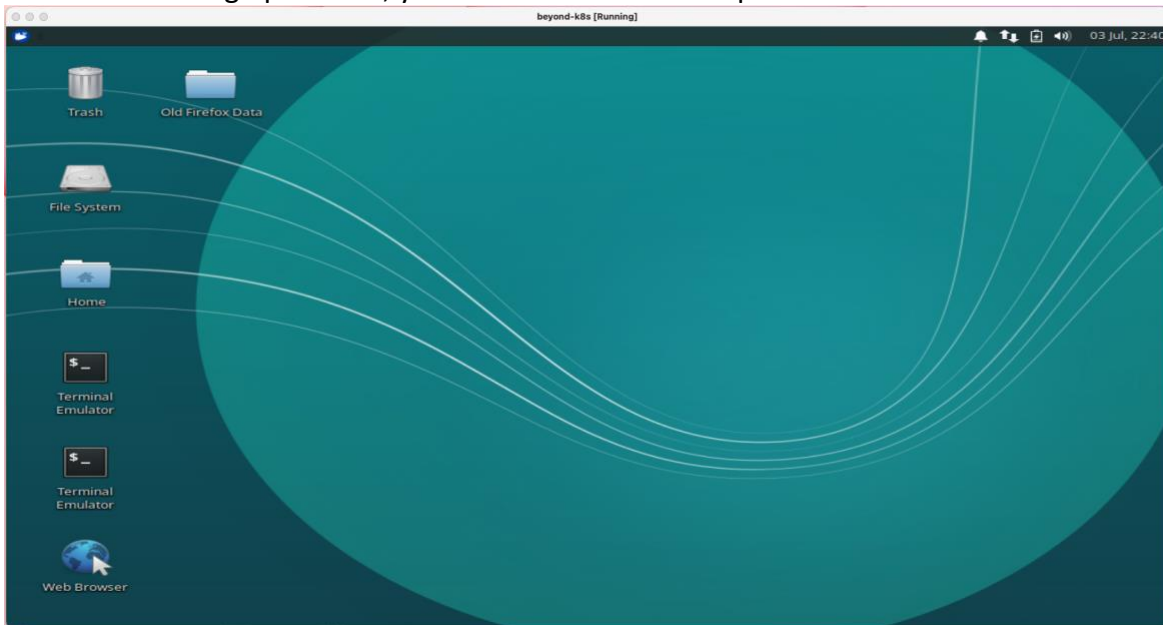
10. If you get a warning dialog that pops up like the one below, it is due to a missing shared directory. (You can see that if you click the Details – not required.) This is not important for using the VM and you can just click OK to proceed and ignore the warning.



11. It may take several minutes for the desktop to appear. If, after several minutes, you don’t see the desktop, try switching to full screen mode (Host key + F) or access the menu item for it through the View menu. On most systems, the Host key here will be the right Ctrl key. (Note: Windows 10 seems to have issues

if you try to switch to scaled mode. If on a Windows 10 system, you may want to avoid that setting. If you do go into scaled mode and the screen seems to disappear, try using the Host key + F to switch out. Or the menu may still be accessible, although hidden, at the very top of the screen.)

12. After starting up the VM, you should see the desktop of the VM.



13. If you are on Windows and get a Windows firewall dialog, you can click both boxes and tell Windows to “Allow access”.

14. If you are given an option to upgrade ubuntu, just decline that.

15. If you have messages at the top of the screen about “Auto capture keyboard” and “mouse pointer integration” you can just click the x on the far right of the messages to dismiss those.

16. Verify that you have internet connectivity from the VM. Open up a terminal session from the VM’s desktop and type something like “ping google.com” to make sure you get a response.

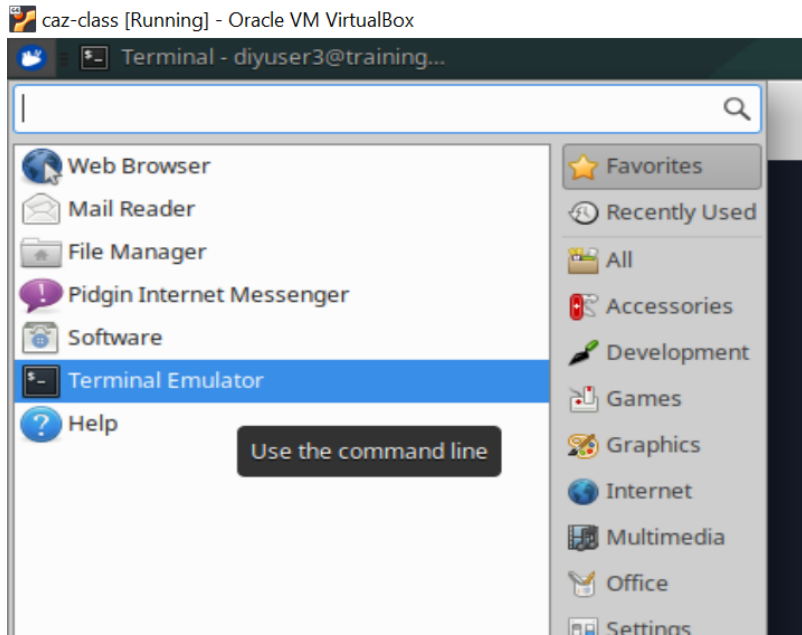
17. Login is diyuser3, diyuser3 if you need it.

### Startup - to do before first lab

1. **If running in the VM**, enable networking. Enable networking by selecting the up/down arrow icon at top right and selecting the option to "Enable Networking". See screenshot below.



Open a terminal session by using the one on your desktop or clicking on the little mouse icon in the upper left corner and selecting **Terminal Emulator** from the drop-down menu.



2. Get the latest files for the class. For this course, we will be using a main directory *k8s-ps* with subdirectories under it for the various labs.

#### If running in the VM

In the terminal window, cd into the main directory and update the files.

```
$ cd k8s-ps
$ git pull
```

#### If NOT running in the VM

```
$ git clone https://github.com/skillrepos/k8s-ps
$ cd k8s-ps
```

3. Whether running in the VM or not, pre-pull images we will need for this workshop.

```
$ ./extra/image-prepull.sh
```

4. If running in the VM, start up the paused Kubernetes (minikube) instance on this system using a script in the *extras* subdirectory. This will take several minutes to run.

```
$ extra/start-mini.sh
```

5. Enable the Kubernetes metrics-server for the cluster.

#### If running in the VM

```
$ sudo minikube addons enable metrics-server
```

If NOT running in the VM, consult documentation for your cluster. (Note this is only needed for one step, so is not critical.)

6. Optional - setup alias. In these labs and on the VM, "k" is aliased to "kubectl". If you are not running in the VM, you can usually do this via the following command if you want:

```
$ alias k=kubectl
```