Kubernetes Dev Workshop – Required preparation before session
Setup for Virtual Machine Environment Option
Version 1.5 - 11/17/22
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 k8s-dev-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

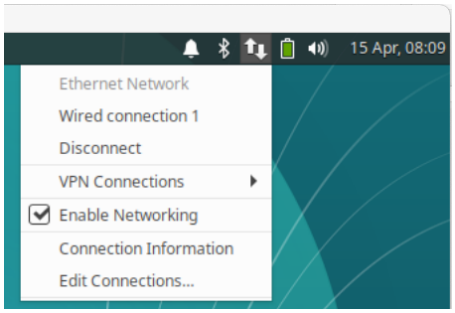
3. Download the pre-configured VirtualBox image (.ova file) for the class from either of the two locations below. (Note that this file is over 3G in size, so it may take a while.):

    - https://www.dropbox.com/s/d8bjycx07v9is4i/k8s-dev.ova?dl=0
    - https://bclconf.s3.us-west-2.amazonaws.com/k8s-dev.ova

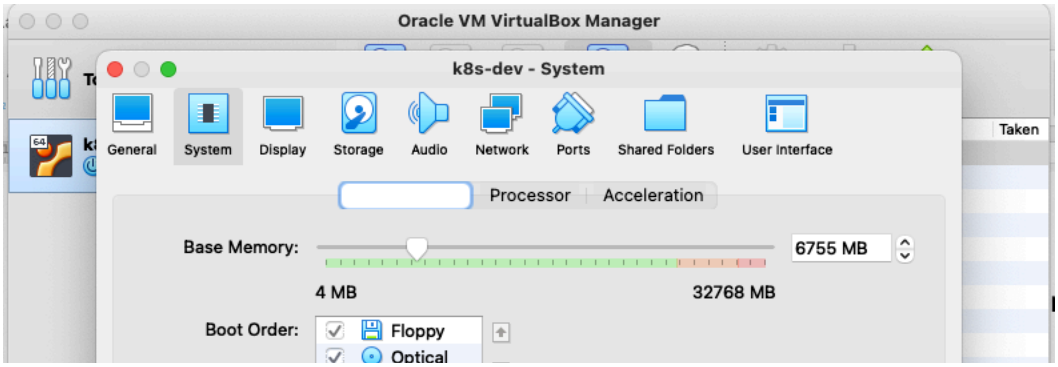
- **MD5 Checksum:** 48cb0ba13649abfdcf3845ad5f29b969
- **SHA-1 Checksum:** 846a0690c51c0c07e5b1843321c841cf58e2ba4d
- **SHA-256 Checksum:** 3955931ef030bb2ad628762e67bf0788bfb96c5f0c4fe7934a1ee55e28ab944b
- **SHA-512 Checksum:**
  b448dd18aa59668c6b46cf35179e56c797de575d4199ade285ceee4e3e6093276c42f9cde51e53ea84a4
  169133a0efb9b3c81e0375e931e0094d0c97cc2d6ad0

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).



Note: If you have trouble getting the VM to run, you may need to scale back the amount of memory used by the system under Settings in VirtualBox



6. Open a terminal window on the desktop and get the latest workshop files.

```
$ cd ~/k8s-dev
$ git pull
$ git reset --hard HEAD
```

7. Start the virtual machine with the command below. Wait for this to complete.

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

(Note: If you hit a problem with the startup, you can also run the following command to delete the minikube instance and update all of the pieces. But you only need to run it if you hit problems with the command above.**$ ~/k8s-dev/extra/setup-mini.sh**)

8. Install the monitoring pieces.

There is a script in the k8s-dev/monitoring directory called **setup-monitoring.sh** that may do this for you. Run it.

There are two pieces of information it will output that you will need to capture for use later:

a. A token to use for logging into the Kubernetes dashboard. It will be shown in the output after this line: " ---- TOKEN to use for logging into dashboard follows ---"
b. The initial password to use to log into Grafana. It will be shown in the output after this line: " --- Grafana initial password follows ---"

After this runs, you should have a namespace named "kubernetes-dashboard" with the dashboard components running in it and a "monitoring" namespace with the Prometheus and Grafana pieces running in it.

9. If the script does not work, you can look at the script to see the commands it is trying to run and work through them or reference the alternative methods below. You only need to use the alternative methods if the script does not work.

Alternative ways to install monitoring pieces:

a. Kubernetes dashboard:

1. Use the following command: (ref https://kubernetes.io/docs/tasks/access-application-cluster/web-ui-dashboard/)

```
$ kubectl apply -f
  https://raw.githubusercontent.com/kubernetes/dashboard/v2.5.0/
  aio/deploy/recommended.yaml
```

2. We need to supply some rbac setup (service account, etc.) to allow access. In the k8s-dev/monitoring directory is the rbac spec that you can just apply as follows:

```
$ cd ~/k8s-dev/monitoring
$ k apply -f dashboard-rbac.yaml
```

b. Prometheus and Kubernetes

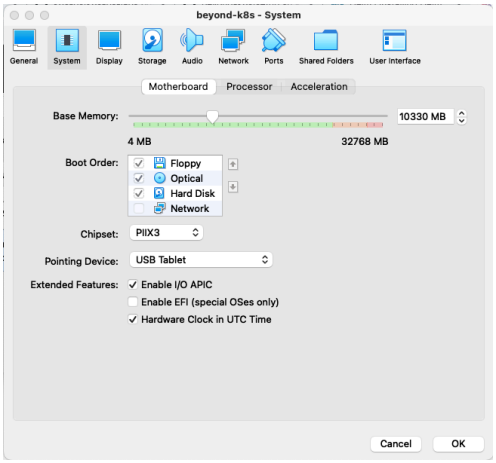
To be able to access these, we need to install the Prometheus and Grafana stack. It's fairly easy to do with Helm. Run the commands below:

```
$ helm repo add prometheus-community https://prometheus-community.github.io/helm-charts

$ helm repo update

$ helm install -n monitoring monitoring --version="38.0.3" prometheus-community/kube-prometheus-stack
```

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

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

12. 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.

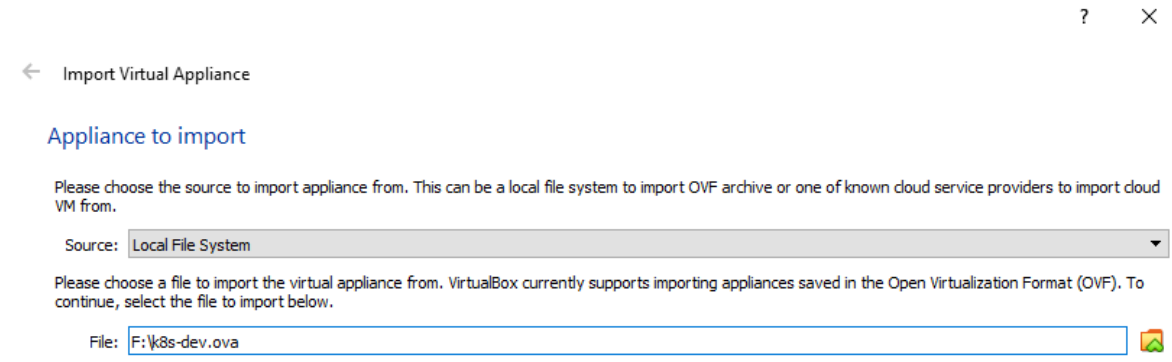


Additional Details for Loading Image (if needed)

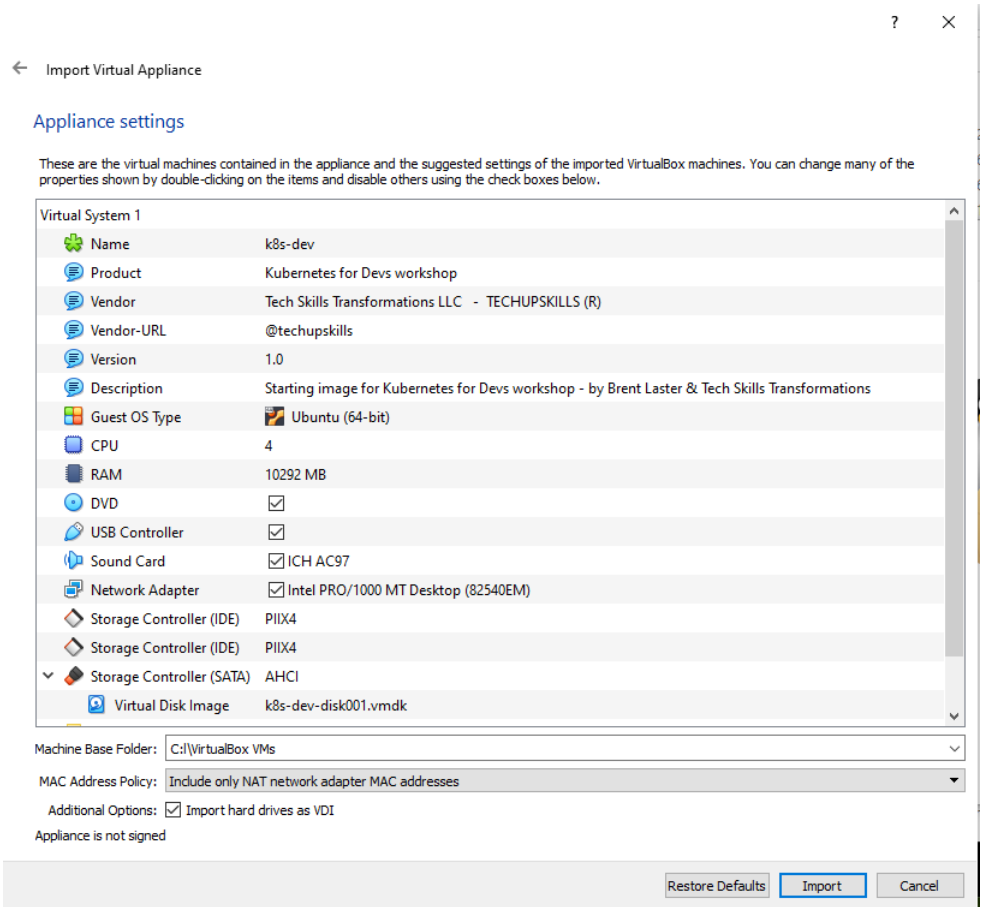
1. Once you have Virtualbox installed and the image downloaded and are ready to proceed, do the steps below to import the appliance.

   a. Open **VirtualBox** on your system.
   b. From the **File** menu, select **"Import Appliance…".**

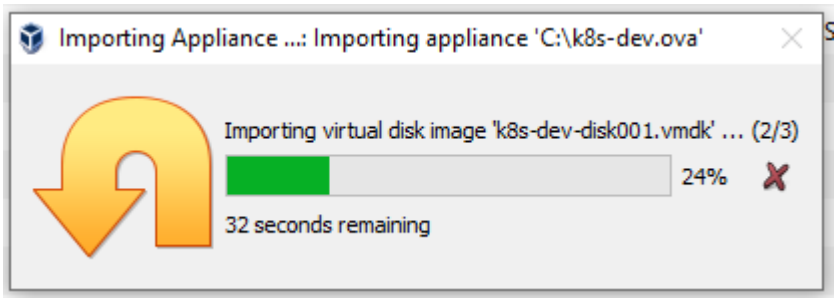
   c. From there, you can type in (or browse to, using the folder icon) the path of the **k8s-dev.ova** file. Then click **Next**.



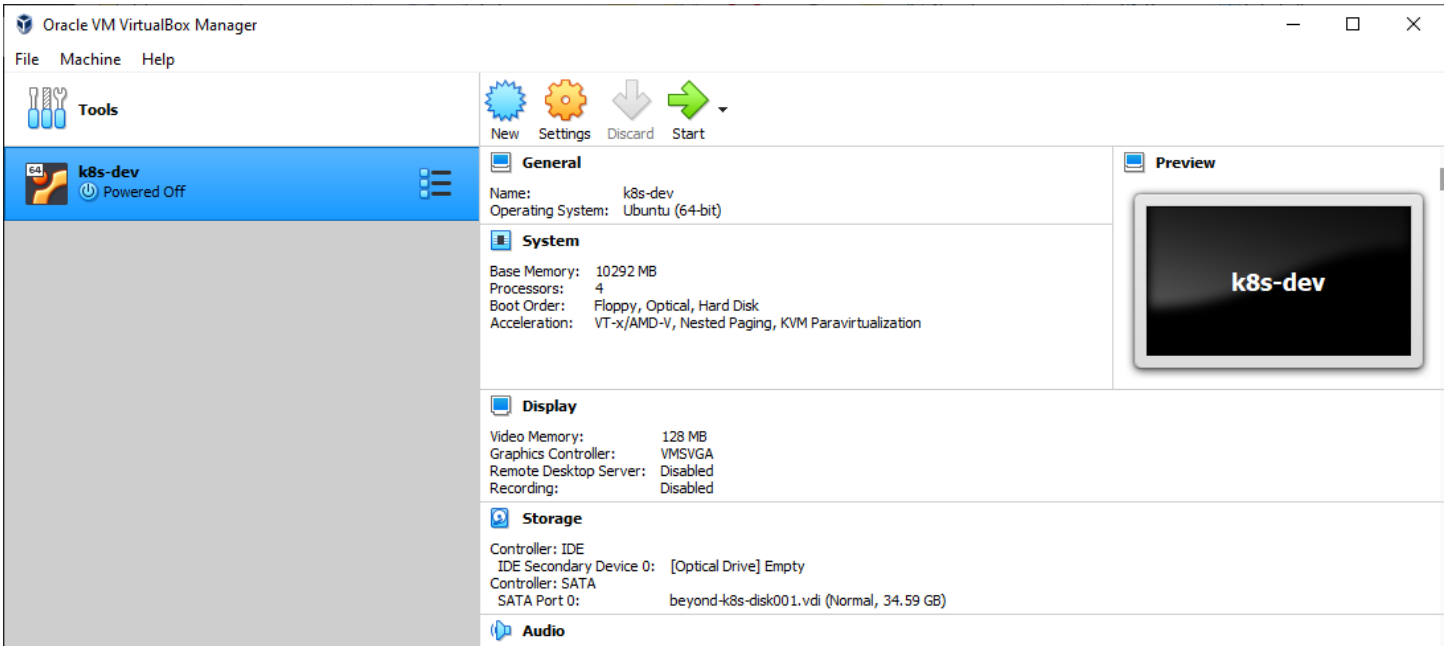
2. On the next screen, if given the option, click the box to reinitialize the MAC addresses. You can just accept the rest of the **Appliance Settings** and then click the **Import** button.



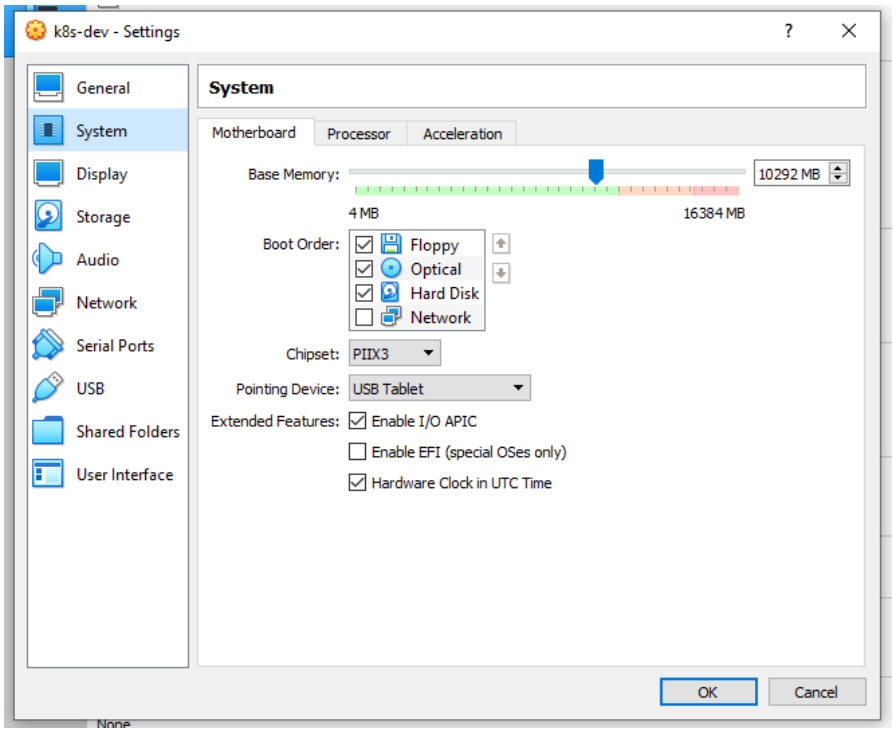
3. You will get a pop-up box for the "license" info. Just click the **Agree** button. Your system will then start processing the import. This may take a while.



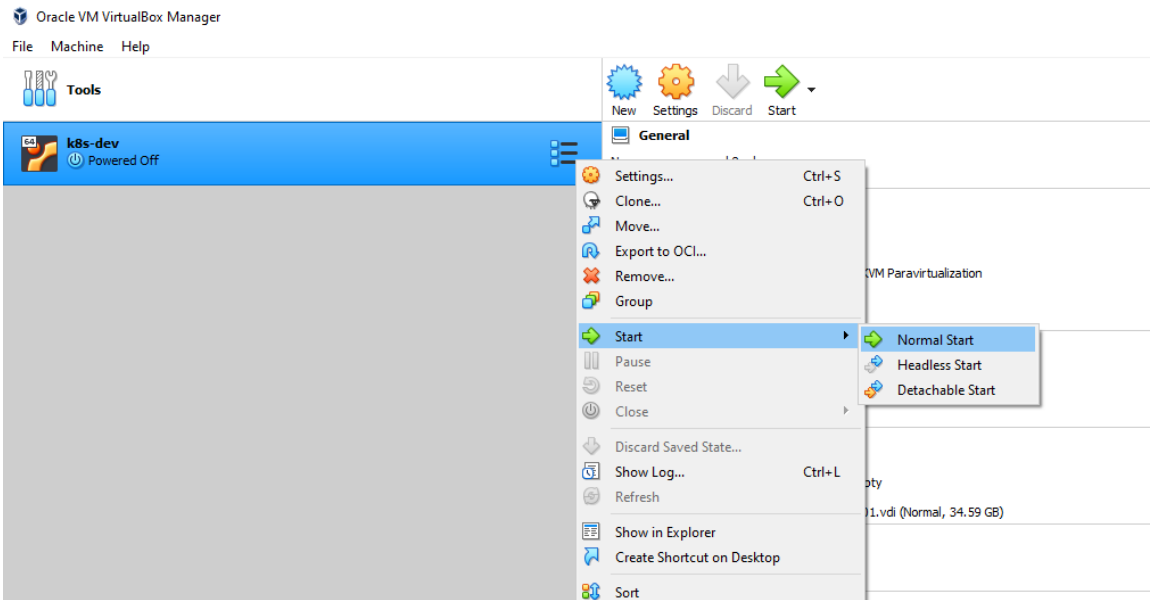
4. After the import is finished, you should have a VM listed in VirtualBox named **k8s-dev.**



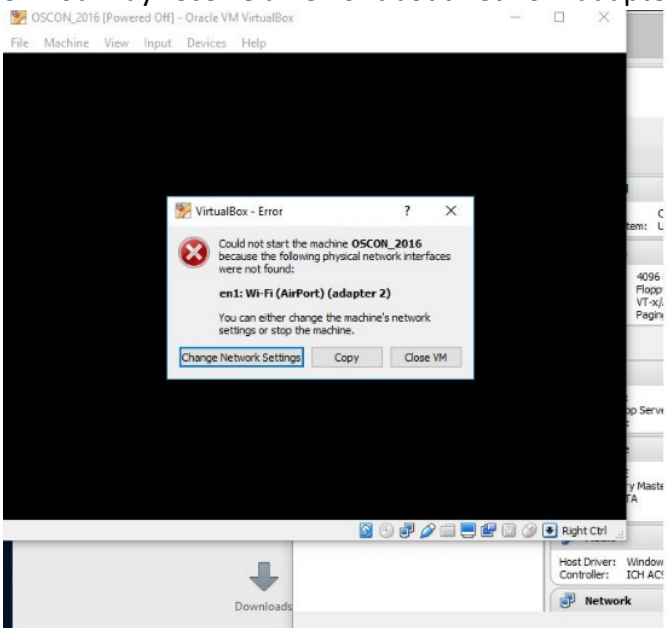
7. **(OPTIONAL)** At this point, depending on the settings of your physical system, you can adjust the amount of memory for the image if you need/want. You can do this by clicking on the **Settings** icon in the menu bar, then **System** in the pop-up box for the settings. Then you can adjust the amount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



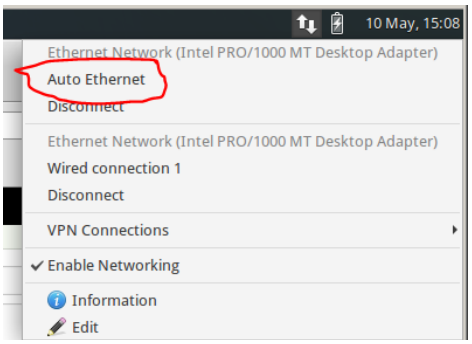
8. At this point, you can start up the virtual image by right-clicking on the image name and then selecting **Normal Start.**



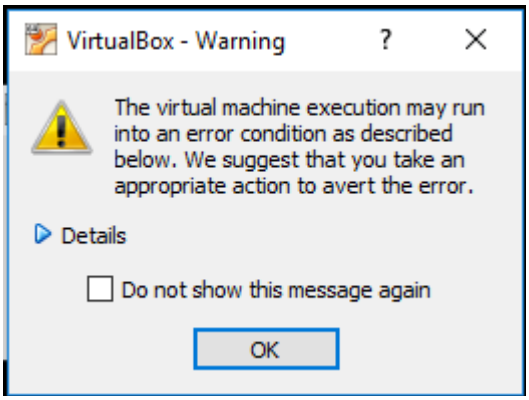
9. You may receive an error about network adapters here similar to the one below.



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

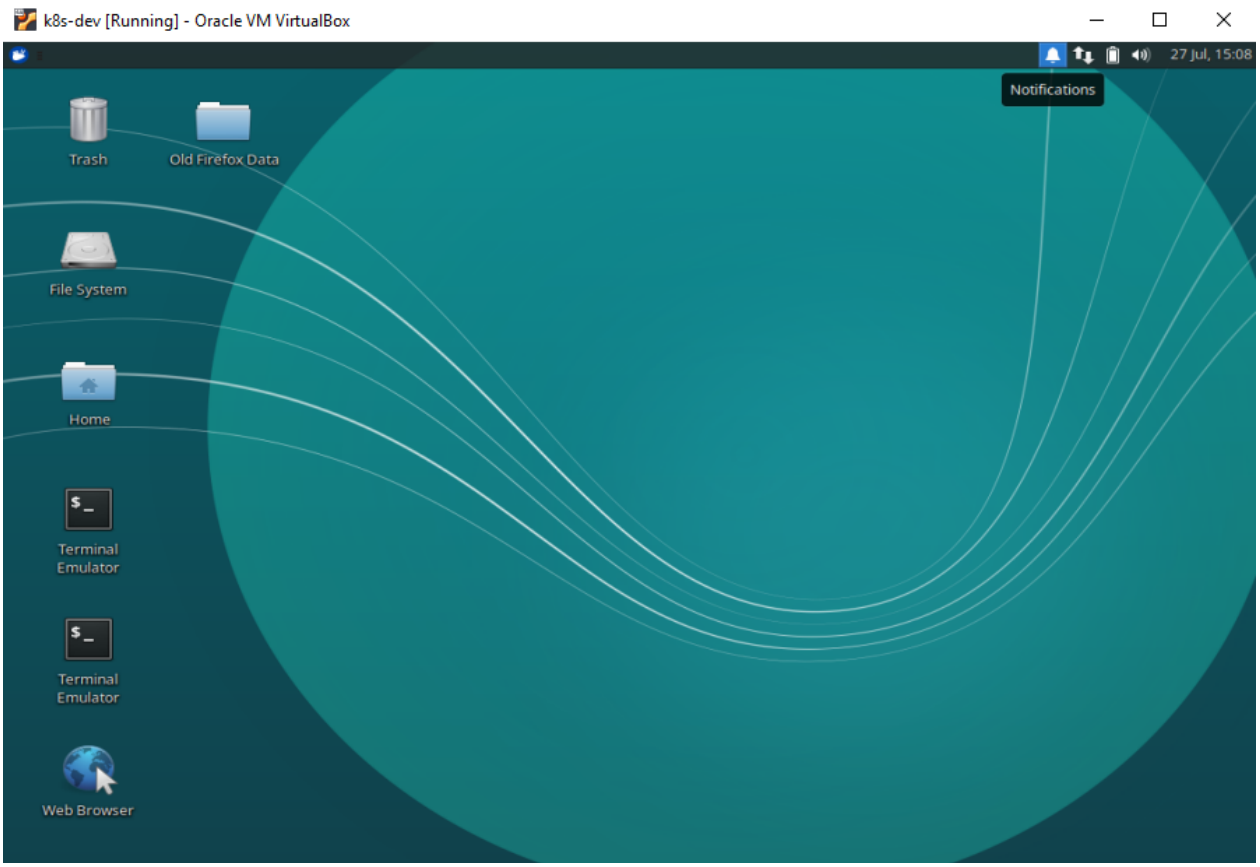


10. If you get a warning dialog that pops up like the one below, it is due to a missing shared directory. (You can see that if you click the Details – not required.) This is not important for using the VM and you can just click OK to proceed and ignore the warning.



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.