

Practice 1 - Setting up the VM

Import fall2016.ova file into VirtualBox as an appliance.

First, copy the fall2016.ova file from your flashdrive to your computer.

Then, start VirtualBox...



You should something like see this:



Click on **File** menu and select **Import Appliance**.

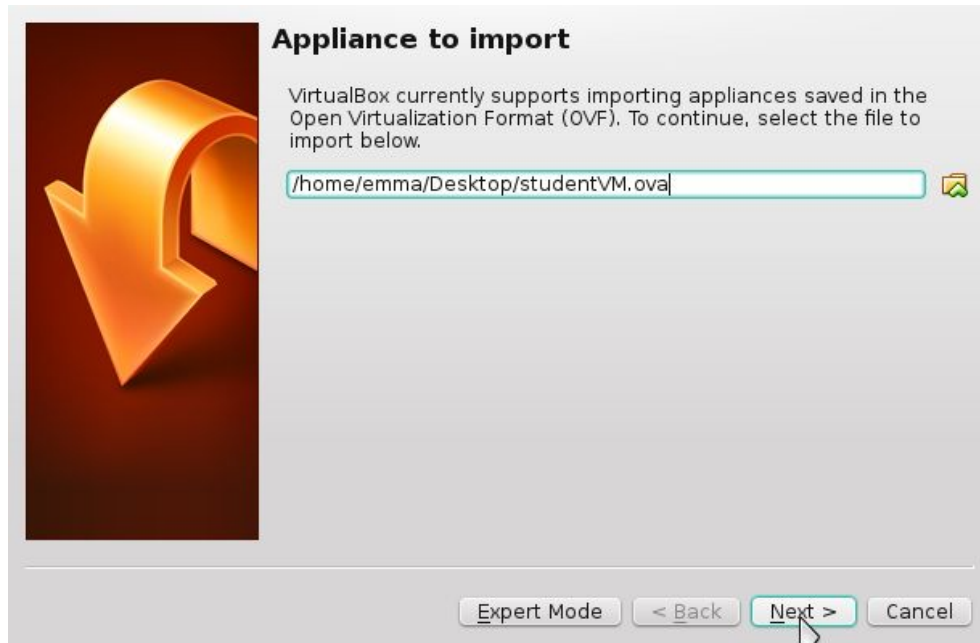


The Appliance Import wizard will appear...

In the wizard, click on the tiny file icon

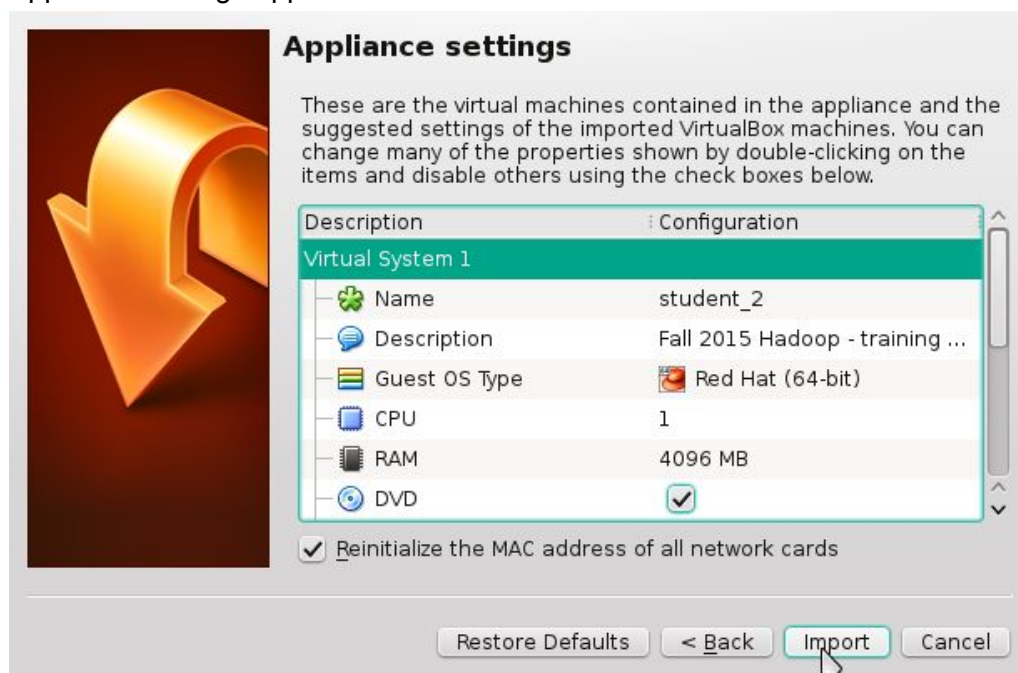


and browse to the .ova file that you just copied onto your machine.
For instance, here I have browsed to studentVM.ova on my desktop.

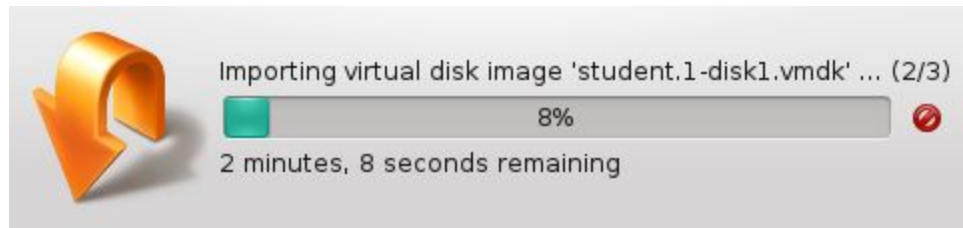


Click next.

Now the Appliance settings appear:



Leave the settings alone and just click Import. The import will start. It may take up to 10 minutes to import. Just wait...



When the import completes, you have a VM installed on VirtualBox. It will appear in the VirtualBox Manager with a label something like:



Starting your VM

- First, close anything you are running that is taking up resources. You want to do this to free up CPU and RAM for your VM. (Otherwise, things can get SLOW.) This includes:
 - Close browsers you might have open (Chrome sucks up CPU and RAM)
 - Close any Adobe readers you have open
 - Anything else? You won't need it so close it.
- Double-click on the label for the fall2016 and it will startup.
- The boot sequence takes a while (5 minutes?), be patient.

Once it is running, you will have access to a VM running Hadoop and all of its minions.

Problems?

Check the .ova file from the command line (open a terminal on your host machine).

On Windows:

```
>> CertUtil -hashfile C:\<path>\<VMname>.ova MD5
```

On Mac:

```
>> md5 <path>/<VMname>.ova
```

On Linux:

```
>> md5sum <path>/<VMname>.ova
```

The checksum is:

2d2eeda5a94f88168be238ce35317a66

If your checksum matches this, your fall2016.ova file is OK.

Use the new VM

After the VM has booted, you should see the desktop. At this point, open a terminal window by clicking on the little black box at the top-left of the desktop.

Let's try some basic commands

In the terminal, check what user you are by typing: `$ whoami`

Next check that Hadoop is available and find out what version is running by typing:

`$ hadoop version`

Next, check our daemons. Type:

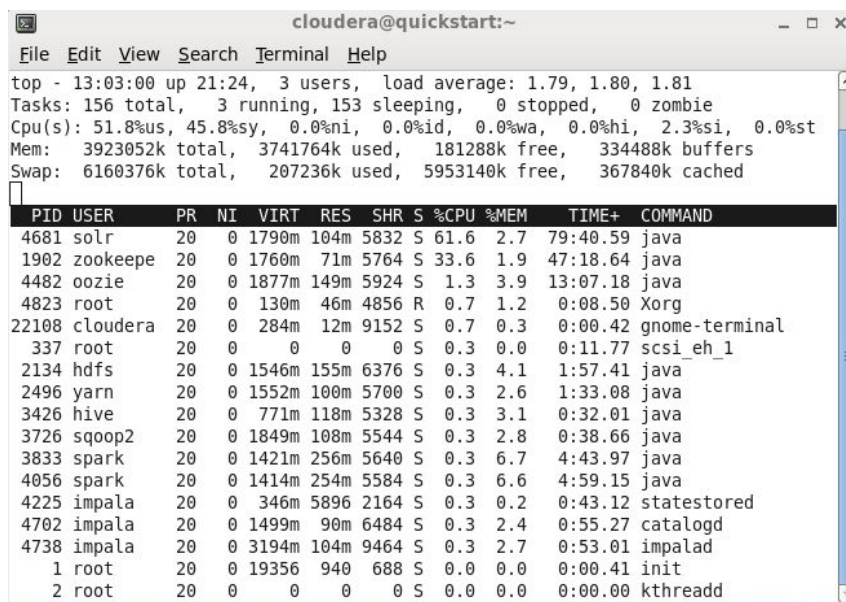
`$ sudo jps`

This shows all the JVMs running on your machine¹. These are the daemons that, together, comprise Hadoop and parts of its ecosystem.

Now type:

`$ top`

This shows the processes that are running on your machine. You should see something like this:



```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
top - 13:03:00 up 21:24, 3 users, load average: 1.79, 1.80, 1.81  
Tasks: 156 total, 3 running, 153 sleeping, 0 stopped, 0 zombie  
Cpu(s): 51.8%us, 45.8%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 2.3%si, 0.0%st  
Mem: 3923052k total, 3741764k used, 181288k free, 334488k buffers  
Swap: 6160376k total, 207236k used, 5953140k free, 367840k cached  


| PID   | USER     | PR | NI | VIRT  | RES  | SHR  | S | %CPU | %MEM | TIME+    | COMMAND        |
|-------|----------|----|----|-------|------|------|---|------|------|----------|----------------|
| 4681  | solr     | 20 | 0  | 1790m | 104m | 5832 | S | 61.6 | 2.7  | 79:40.59 | java           |
| 1902  | zookeepe | 20 | 0  | 1760m | 71m  | 5764 | S | 33.6 | 1.9  | 47:18.64 | java           |
| 4482  | oozie    | 20 | 0  | 1877m | 149m | 5924 | S | 1.3  | 3.9  | 13:07.18 | java           |
| 4823  | root     | 20 | 0  | 130m  | 46m  | 4856 | R | 0.7  | 1.2  | 0:08.50  | Xorg           |
| 22108 | cloudera | 20 | 0  | 284m  | 12m  | 9152 | S | 0.7  | 0.3  | 0:00.42  | gnome-terminal |
| 337   | root     | 20 | 0  | 0     | 0    | 0    | S | 0.3  | 0.0  | 0:11.77  | scsi_eh_1      |
| 2134  | hdfs     | 20 | 0  | 1546m | 155m | 6376 | S | 0.3  | 4.1  | 1:57.41  | java           |
| 2496  | yarn     | 20 | 0  | 1552m | 100m | 5700 | S | 0.3  | 2.6  | 1:33.08  | java           |
| 3426  | hive     | 20 | 0  | 771m  | 118m | 5328 | S | 0.3  | 3.1  | 0:32.01  | java           |
| 3726  | sqoop2   | 20 | 0  | 1849m | 108m | 5544 | S | 0.3  | 2.8  | 0:38.66  | java           |
| 3833  | spark    | 20 | 0  | 1421m | 256m | 5640 | S | 0.3  | 6.7  | 4:43.97  | java           |
| 4056  | spark    | 20 | 0  | 1414m | 254m | 5584 | S | 0.3  | 6.6  | 4:59.15  | java           |
| 4225  | impala   | 20 | 0  | 346m  | 5896 | 2164 | S | 0.3  | 0.2  | 0:43.12  | statestored    |
| 4702  | impala   | 20 | 0  | 1499m | 90m  | 6484 | S | 0.3  | 2.4  | 0:55.27  | catalogd       |
| 4738  | impala   | 20 | 0  | 3194m | 104m | 9464 | S | 0.3  | 2.7  | 0:53.01  | impalad        |
| 1     | root     | 20 | 0  | 19356 | 940  | 688  | S | 0.0  | 0.0  | 0:00.41  | init           |
| 2     | root     | 20 | 0  | 0     | 0    | 0    | S | 0.0  | 0.0  | 0:00.00  | kthreadd       |


```

¹ We need to use sudo because 'jps' alone only shows the JVMs you own.

The **top** command shows the processes running for every user on the system. You can see that there are several users in this Hadoop installation. Each member of the Hadoop ecosystem has a user. We can see all the Hadoop users that are running java programs.

Incidentally, you are user `ccloudera` and you are running a `gnome-terminal`.

If you are new to Linux and you haven't used **top** before, you can explore it by typing:

```
$ man top
```

This will bring up the manual pages about **top** and explain its use.

Exploring HDFS

Hadoop is subdivided into several subsystems. For example, there is a subsystem for working with files in HDFS and another for launching and managing MapReduce processing jobs.

The subsystem associated with HDFS in the Hadoop wrapper program is called FsShell. This subsystem can be invoked with the command `hadoop fs`.

In the terminal window, enter:

```
$ hadoop fs
```

You see a help message describing all the commands associated with the FsShell subsystem.

Enter:

```
$ hadoop fs -ls /
```

This shows you the contents of the root directory in HDFS. There will be multiple entries, one of which is `/user`. Individual users have a “home” directory under this directory, named after their username; your username is `ccloudera`, therefore your home directory in HDFS is `/user/ccloudera`.

View the contents of the `/user` directory by running:

```
$ hadoop fs -ls /user
```

Depending on the VM you are using, there may or may not be any data stored in HDFS for the `ccloudera` user. If there are no files yet, the command silently exits.

Note: the directory structure in HDFS has nothing to do with the directory structure of the local filesystem; they are completely separate namespaces.

Add data to HDFS

From within your VM, change directories to the local filesystem directory containing the sample data we will be using today.

```
$ cd ~/Desktop/datasets
```

Notice, the twiddle (~) is shorthand for your local directory, so `$ cd ~` takes you to your home directory, which is `/home/cloudera`.

If you want to check what your current working directory is, type:

```
$ pwd
```

Next, check the contents of the directory

```
$ ls
```

You will see several directories and zip files.

If necessary, unzip the `stock-sample.zip` file - it will create a `stock-sample` directory. Copy the `stock-sample` directory into HDFS:

```
$ hadoop fs -put /home/cloudera/datasets/stock-sample /user/cloudera/.
```

When you do the “put” you may see several warnings, like this:

```
16/10/24 16:54:27 WARN hdfs.DFSCClient: Caught exception
java.lang.InterruptedException
    at java.lang.Object.wait(Native Method)
    at java.lang.Thread.join(Thread.java:1249)
    at java.lang.Thread.join(Thread.java:1323)
    at ...
```

Ignore these warnings, they are simply warning that we are using a `tmp` directory for some data storage.

Check the data after copying:

```
$ hadoop fs -ls stock-sample
```

This lists the contents of the `/home/cloudera/stock-sample` HDFS directory. Note that the “put” command copied **all files** in `stock-sample` - when used, **put** command does a recursive copy.

Let’s look at the contents of a file. Enter:

```
$ hadoop fs -cat stock-sample/BOX | tail -n 50
```

This prints the last 50 lines of the BOX file to your terminal. This command is handy for viewing the input and output of Hadoop jobs. Very often, an individual file used by a MapReduce program is very large, making it inconvenient to view the entire file in the terminal. For this reason, it is often a good idea to pipe the output of the `fs -cat` command into **head**, **tail**, **more**, or **less**.

To move a file out of HDFS and onto your local file system, use the `fs -get` command. This command takes two arguments: an HDFS path and a local path.

The get command copies the HDFS contents into the local filesystem. For instance:

```
$ hadoop fs -get stock_sample/BOX ~/datasets/tmp
```

Feel free to view the results via:

```
$ less ~/datasets/tmp
```

Other commands

There are several other operations available with the `hadoop fs` command to perform most common filesystem manipulations: `mv`, `cp`, `mkdir`, etc.

To see the full set, just enter:

```
$ hadoop fs
```


Use Hue to browse, view and manage HDFS files

While you are still in your VM, startup Firefox.

1. Click on the Hue bookmark, or visit `http://localhost:8888`
2. When you start it, Hue may ask you to supply some credentials. If so, enter username **cloudera** and password **cloudera**.
3. Note, when you first log into Hue you may see a *misconfiguration* warning. This is because not all the services Hue depends on are running on the course VM. You can disregard this message.
4. Hue has many useful features, many of which will be covered during the course. For now, to access HDFS, click **File Browser** in the Hue menu bar. (If your Firefox windows is too small to display the full menu names, you will just see the icons instead. The mouse-over text is "Manage HDFS")
5. By default, the contents of the HDFS home directory (`/user/cloudera`) display. You should see the `stock-sample` directory you created earlier. Click on it to see the contents.
6. View one of the files by clicking on the name of any one of the stocks.
7. In the file viewer, the contents of the file are displayed on the right. In this case, the file is moderately large. For large files, rather than displaying the entire contents on one screen, Hue provides buttons to move between pages.
8. Return to the directory review by clicking **View file location** in the **Actions** panel on the left.
9. Click the **up** arrow to return to your user directory.
10. Create a new directory by clicking on **New**. Select **Directory** and create a directory named "companyInfo".
11. To upload files, click the **Upload** button. You can choose to upload a plain file, or to upload a zipped file. Select **Files**, then click **Select Files**.
12. A Linux file browser appears. Browse to your local file at `/home/cloudera/datasets/companies`.
13. Choose `companyListNASDAQ.csv` and click the open button. This will upload the file to your home directory. Now, you can drag-and-drop `companyListNASDAQ.csv` into the new `companyInfo` directory on HDFS.

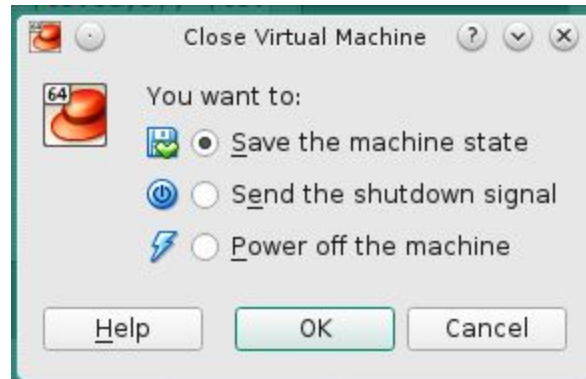
Please Exit Gracefully

When you want to close the VM, there are only two ways that are “graceful”.

1. The first is to *Save the VM's state* by closing the window via the close button:



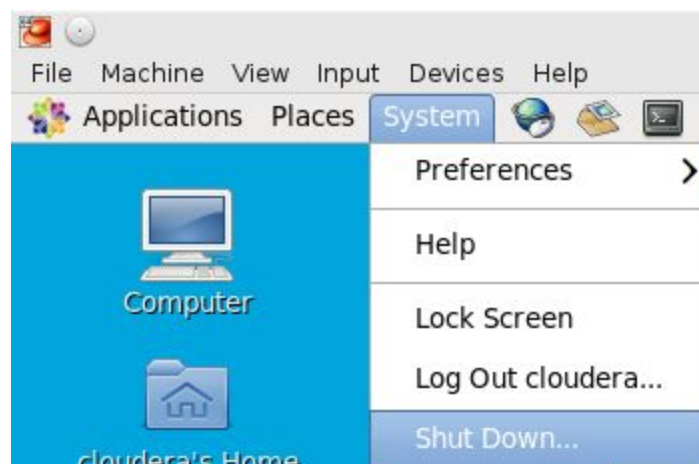
This will bring up a menu:



The only viable option is “**Save the machine state**”. The other options act like you just hit the power button or unplugged your computer (NOT GRACEFUL). Only use the “Save the machine state” option.

When you save state, the VM shuts down quickly and will start up quickly.

2. The second is to shutdown the VM. First select **System**, and then select **Shutdown**.



This will halt the machine gracefully and shut it down completely. The next time you start it, it will take a few minutes because it will be going through the whole boot sequence.

End of Practice 1