# Hadoop Tutorial

## I. Download and install virtualbox 5.2.8 to run virtual images at:

**a. Download:**

For Mac OS X users, the link is at: https://download.virtualbox.org/virtualbox/5.2.8/VirtualBox-5.2.8-121009-OSX.dmg
For Window users, the link is at: https://download.virtualbox.org/virtualbox/5.2.8/VirtualBox-5.2.8-121009-Win.exe
For Linux users, the link is at: https://www.virtualbox.org/wiki/Linux_Downloads
Choose the right version corresponding to the current OS system you are using

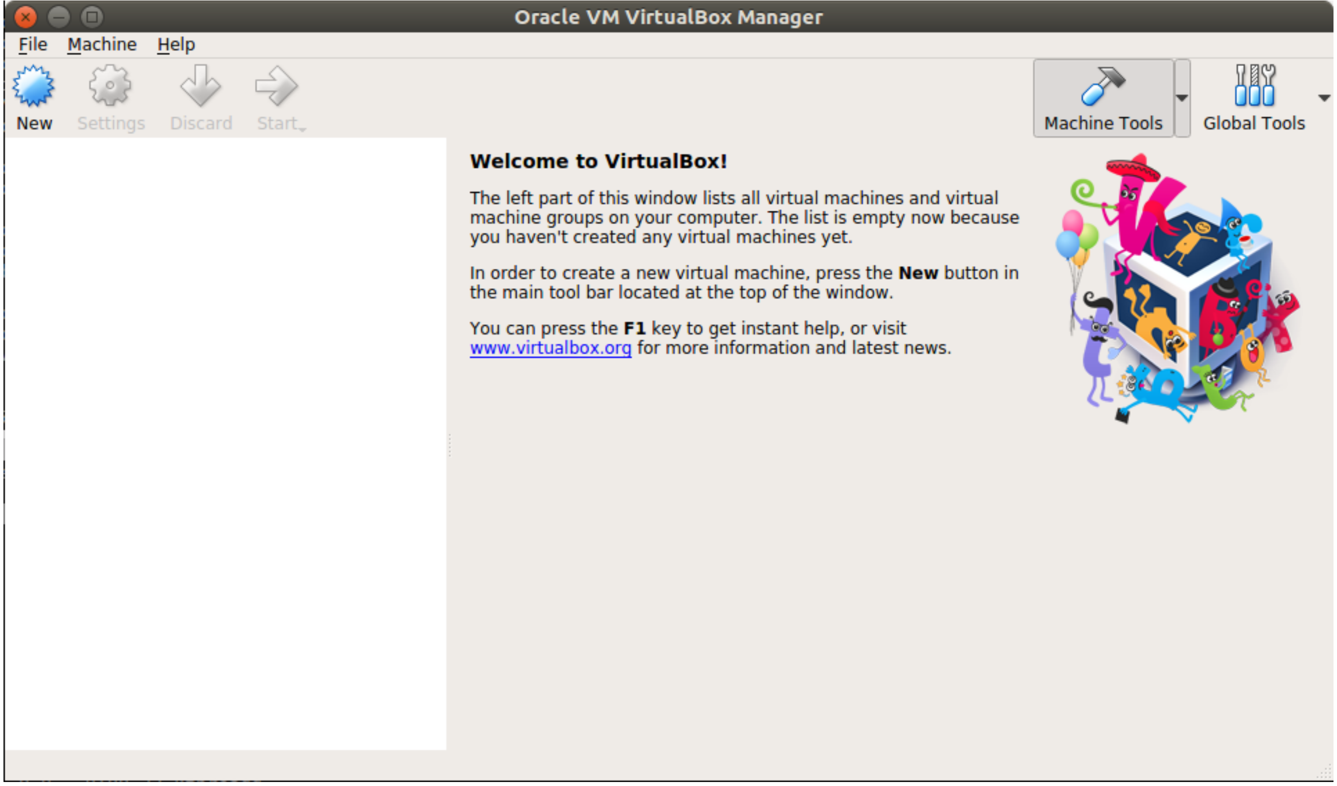
**b. Install:**

- For window and OS X users, installing virtualbox is similar to installing other apps.
- For Linux users (e.g. Ubuntu), you will need sudo permission. Installing virtualbox can be done by running following commands:

```
sudo dpkg -i [path_to_downloaded virtualbox]
sudo apt-get install -f
```

**c. Run virtualbox:**

- After completely installing virtualbox, we will run it. The virtualbox GUI will be as following:



**Welcome to VirtualBox!**

The left part of this window lists all virtual machines and virtual machine groups on your computer. The list is empty now because you haven't created any virtual machines yet.

In order to create a new virtual machine, press the **New** button in the main tool bar located at the top of the window.

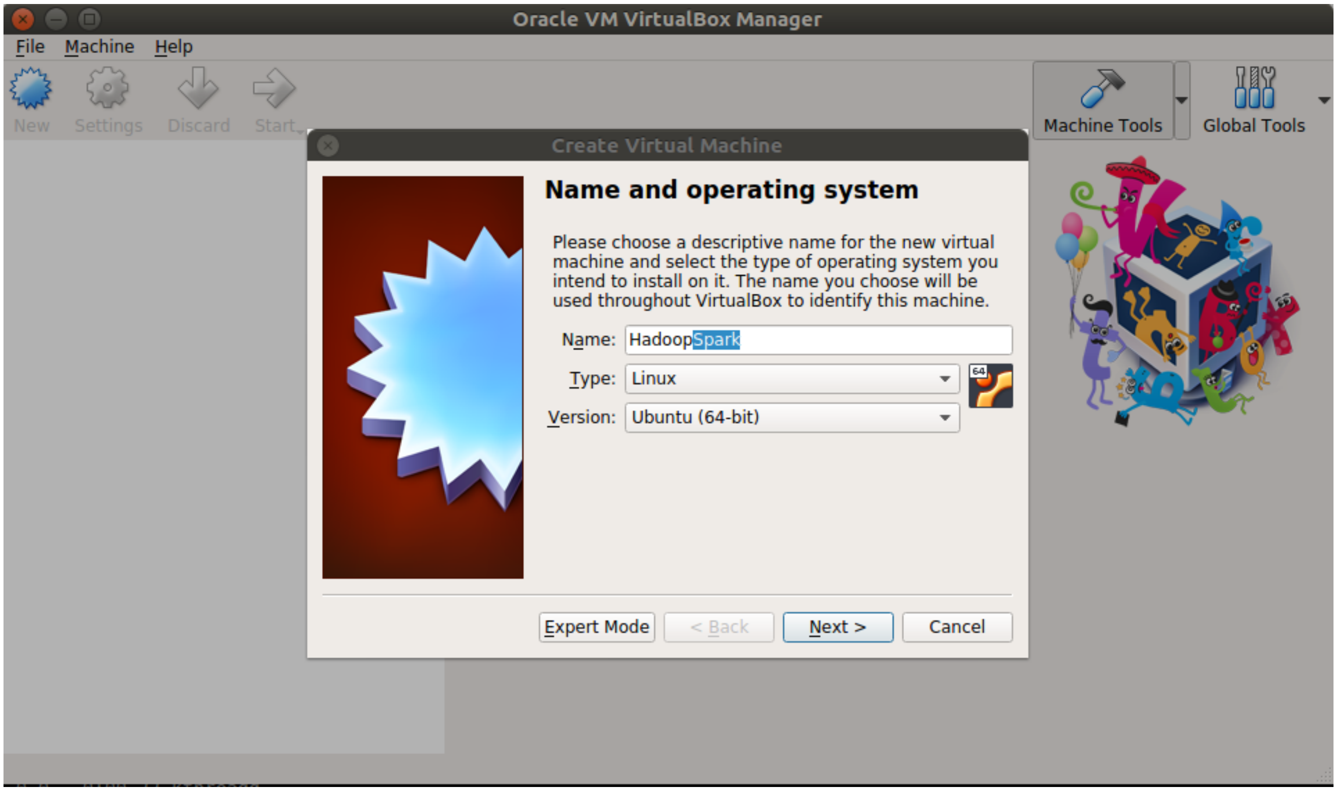
You can press the **F1** key to get instant help, or visit www.virtualbox.org for more information and latest news.

## II. Downloading and running virtual machine:

1. Download a compressed virtual machine image file at http://kmlee06.cs.wpi.edu/ds3001/HadoopSpark.vdi.tar.gz (Strongly recommend downloading it on

campus because it does not require running VPN and you can download it quickly. WPI VPN is required if you are off-campus). The virtual machine image occupies 7.1 Gb after decompressing it. Make sure your hard drive has enough space before downloading (or you have use an external hard drive which may be slightly slow).

2. In the virtualbox GUI, click New. A box will pop up as following:



Name and operating system

Please choose a descriptive name for the new virtual machine and select the type of operating system you intend to install on it. The name you choose will be used throughout VirtualBox to identify this machine.

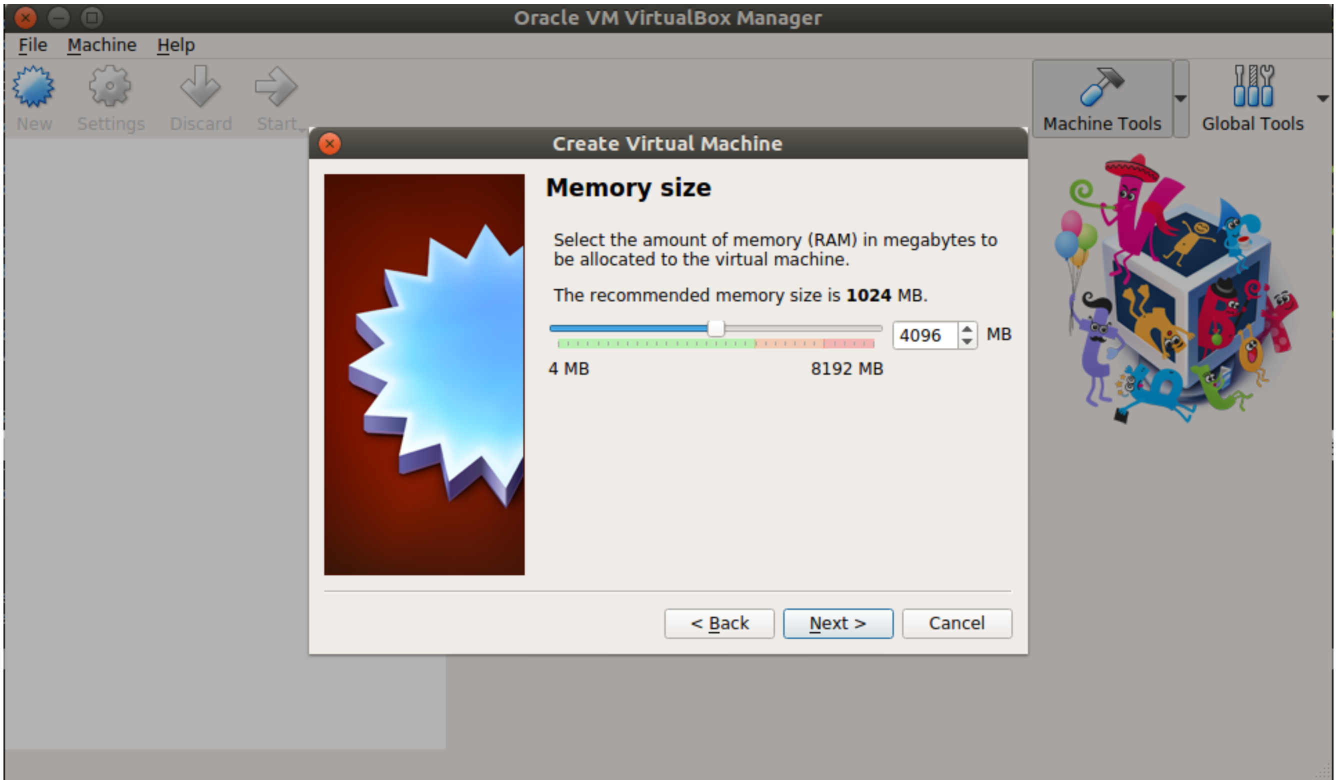
Name: HadoopSpark

Type: Linux

Version: Ubuntu (64-bit)

- The first field [Name] is the name of the virtual machine. For example, type **HadoopSpark**
- The second field [Type] is the OS type.  Choose **Linux**
- The third field [Version], choose **Ubuntu (64-bit)**

Click Next. Then setting up memory size, recommended to have **at least 4Gb of Ram** to have a good performance.



Memory size

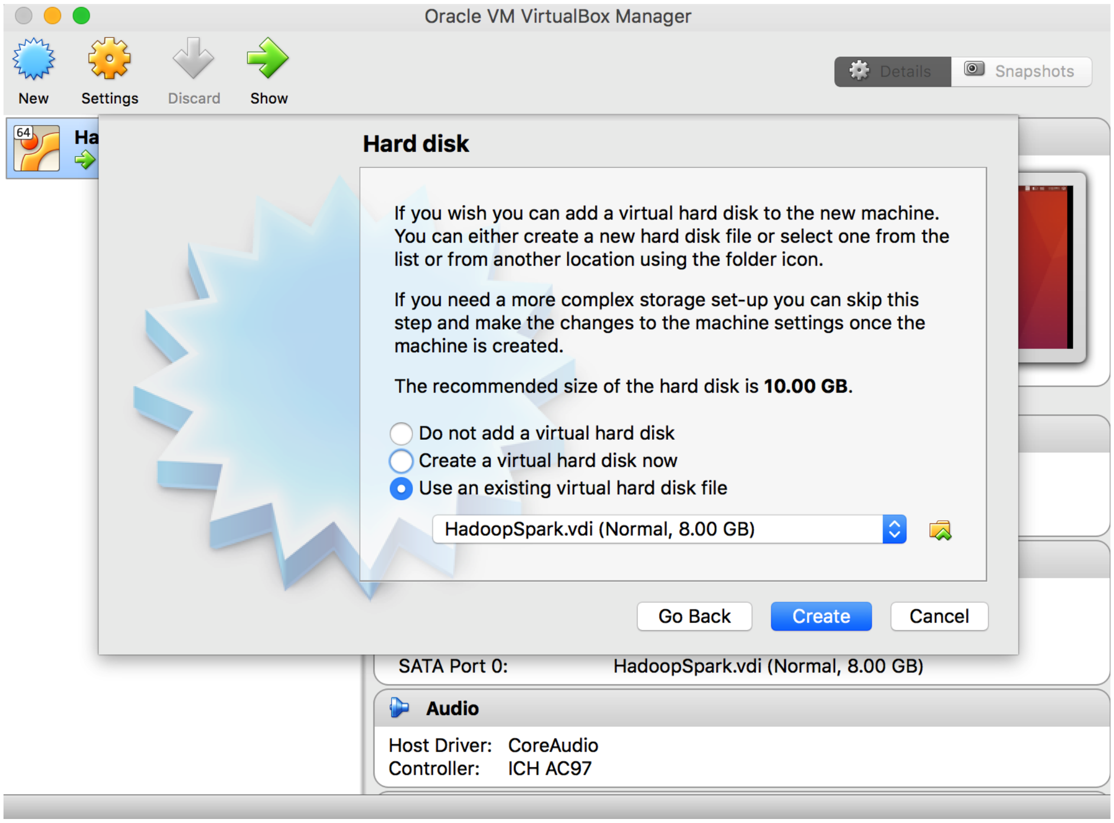
Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.

The recommended memory size is **1024** MB.

4096 MB

4 MB 8192 MB

Click Next. Choose [Use an existing virtual hard disk file], then point to the downloaded virtual image [HadoopSpark.vdi]. Then click Create.



Hard disk

If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select one from the list or from another location using the folder icon.

If you need a more complex storage set-up you can skip this step and make the changes to the machine settings once the machine is created.

The recommended size of the hard disk is **10.00 GB**.

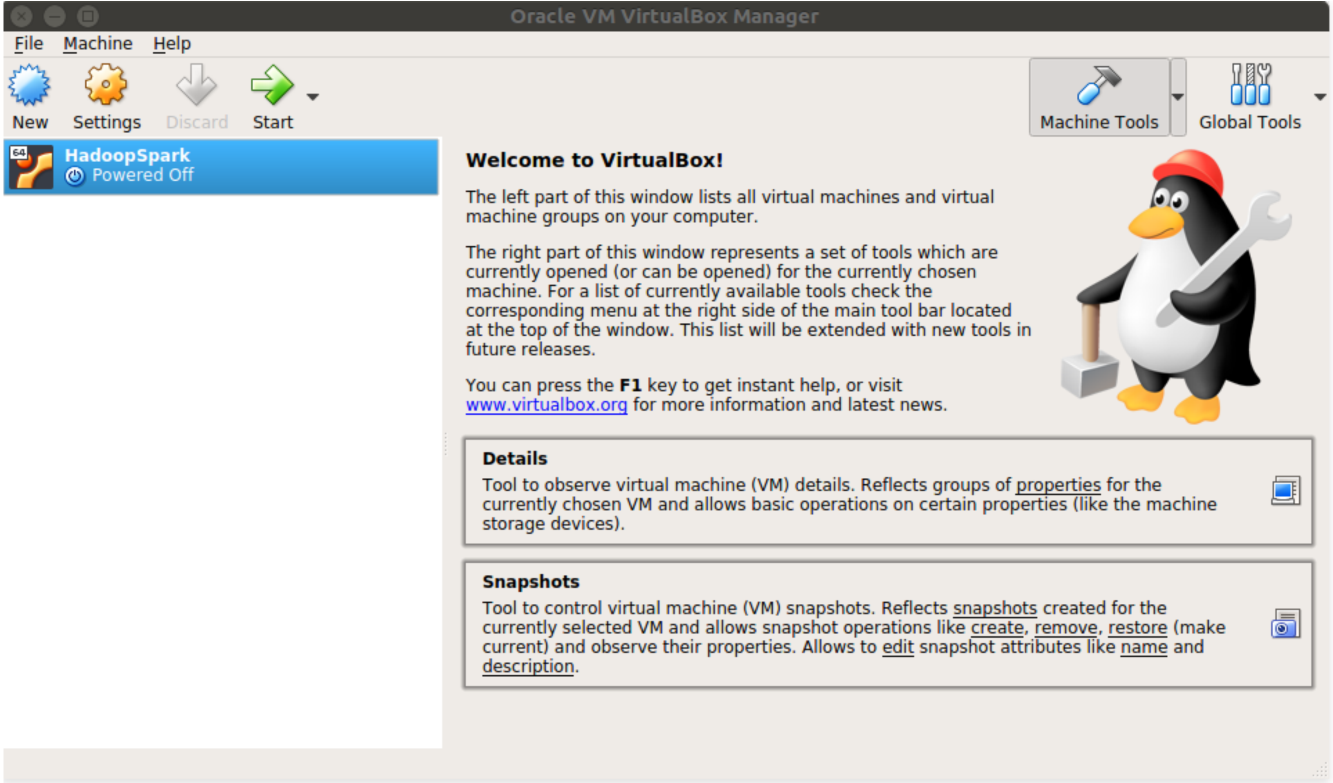
○ Do not add a virtual hard disk
○ Create a virtual hard disk now
● Use an existing virtual hard disk file

HadoopSpark.vdi (Normal, 8.00 GB)

SATA Port 0: HadoopSpark.vdi (Normal, 8.00 GB)

Host Driver: CoreAudio
Controller: ICH AC97

Next, select [HadoopSpark], then click [Start] to run virtual machine



**Welcome to VirtualBox!**

The left part of this window lists all virtual machines and virtual machine groups on your computer.

The right part of this window represents a set of tools which are currently opened (or can be opened) for the currently chosen machine. For a list of currently available tools check the corresponding menu at the right side of the main tool bar located at the top of the window. This list will be extended with new tools in future releases.

You can press the **F1** key to get instant help, or visit www.virtualbox.org for more information and latest news.

**Details**

Tool to observe virtual machine (VM) details. Reflects groups of properties for the currently chosen VM and allows basic operations on certain properties (like the machine storage devices).

**Snapshots**

Tool to control virtual machine (VM) snapshots. Reflects snapshots created for the currently selected VM and allows snapshot operations like create, remove, restore (make current) and observe their properties. Allows to edit snapshot attributes like name and description.

Default login account:
username: datascience
password: datascience

3. Run a Hadoop distributed file system (when you restart the VM, you have to run the below command to launch your HDFS)
Open a terminal (the 4th icon on the most left bar) and run the below command:

- start hadoop namenode and datanode: (You must start the distributed file system (dfs) first to run Hadoop commands and MapReduce jobs).

```
Syntax: /usr/local/hadoop/sbin/start-dfs.sh
```

4. Play around with Hadoop commands:
Open a terminal to play with following commands:

- list directory in Hadoop distributed file system (hdfs):

```
Syntax: hadoop fs -ls [path_to_directory]
for example:  hadoop fs -ls /
```

- create a directory in hdfs:

```
Syntax: hadoop fs -mkdir [path_to_new_directory]
for example: hadoop fs -mkdir /user/tmp
```

- remove a directory in hdfs:

```
Syntax: hadoop fs -rm -r [path_to_directory]
for example: hadoop fs -rmr /user/tmp
```

- upload a file in hdfs:

```
Syntax: hadoop fs -put [local_file_path] [hadoop_file_path]
```

Example:

In the terminal, go to the main directory using command:

```
   cd /home/datascience
```

make a working directory and go to it:

```
   mkdir /home/datascience/work
   cd /home/datascience/work
```

Create a file with content ‘hello world this class is interesting’

```
   echo ‘hello world this class is interesting’ > sample.txt
```

upload this [sample.txt] file into hadoop distributed file system:

```
   hadoop fs -put sample.txt /user/
```

check if the file is uploaded?

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

- View content of a hadoop distributed file system file:

```
Syntax: hadoop fs -cat [path_to_hadoop_file]
example: hadoop fs -cat /user/sample.txt
```

- More command lines? refer to https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/FileSystemShell.html

## III. Running a toy example in python

In this part, we will run a word count example for Python users. (the example is credited from http://www.science.smith.edu/dftwiki/index.php/Hadoop_Tutorial_2_--_Running_WordCount_in_Python)

**All below codes are already given at /home/datascience/sample_codes/**

**The running commands are stored at /home/datascience/sample_codes/run.sh. To run it, just simply run [bash  /home/datascience /sample_code/run.sh]**

**Details:**

- Creating a mapper file using following commands:

**gedit mapper.py**

then add the following content to the file:

```
#!/usr/bin/env python

import sys

for line in sys.stdin:
   line = line.strip()
```

```
    words = line.split()
    for word in words: print ‘%s\t%s’ %(word, 1) #count 1 for each word
```

- Creating a reducer file as followings:

**gedit reducer.py**

```
#!/usr/bin/env python
import sys

# maps words to their counts
word2count = {}

# input comes from stdin
for line in sys.stdin:
    # remove leading and trailing whitespace
    line = line.strip()

    # parse the input we got from mapper.py
    word, count = line.split('\t', 1)
    # convert count (currently a string) to int
    try:
        count = int(count)
    except ValueError:
        continue

    try:
        word2count[word] = word2count[word]+count
    except:
        word2count[word] = count

# write the tuples to stdout
# Note: they are unsorted
for word in word2count.keys():
    print '%s\t%s'% ( word, word2count[word] )
```

**Run the job:**

```
Syntax: hadoop jar /usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.7.5.jar -files
[path_to_mapper_file],[path_to_reducer_file] -mapper  [path_to_mapper_file] -reducer
[path_to_reducer_file] -input [hdfs_file_path] -output [hdfs_output_directory]
```

For example: with the [**sample.txt]** file we already uploaded into hdfs, we will count the frequency of each unique word in the file as followings:

- make sure the output folder [hdfs_output_directory] is not existed:

```
hadoop fs –rm -r /user/output
```

- run the job:

```
hadoop jar /usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.7.5.jar -files ./mapper.py,./reducer.py -mapper ./mapper.py -reducer ./reducer.py -input /user/sample.txt -output /user/output
```

- view the output file:

```
hadoop fs -cat /user/output/part*
```

The result is as following:

```
this        1
interesting        1
is          1
hello     1
world   1
class     1
```

**IV. Now feel free to revise mapper.py, reducer.py and run.sh files for homework 4.**

**Acknowledgement:**
We adopted a VM image created by Dr. Mohamed Y. Eltabakh, and updated it.