

# **Chapter 2. Fundamentals of Big Data**

## **Exercise Workbook**

## Contents

Lab 1: Starting VirtualBox.....	3
Lab 2: Working with HDFS.....	14
Lab 3: Working with YARN/MapReduce.....	23

## Lab 1: Starting VirtualBox

---

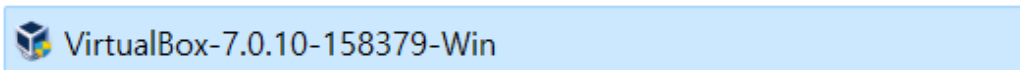
In this lab, you will start and use the virtual machine created for our lab.

### 1. Download and install VirtualBox

- 1.1. Go to the [official website](#) to download the VirtualBox installer for your operating system (Windows in this case).
- 1.2. Download the installation file according to the OS. If you are Windows user, Click the Windows hosts. (You can download latest version on the website.)



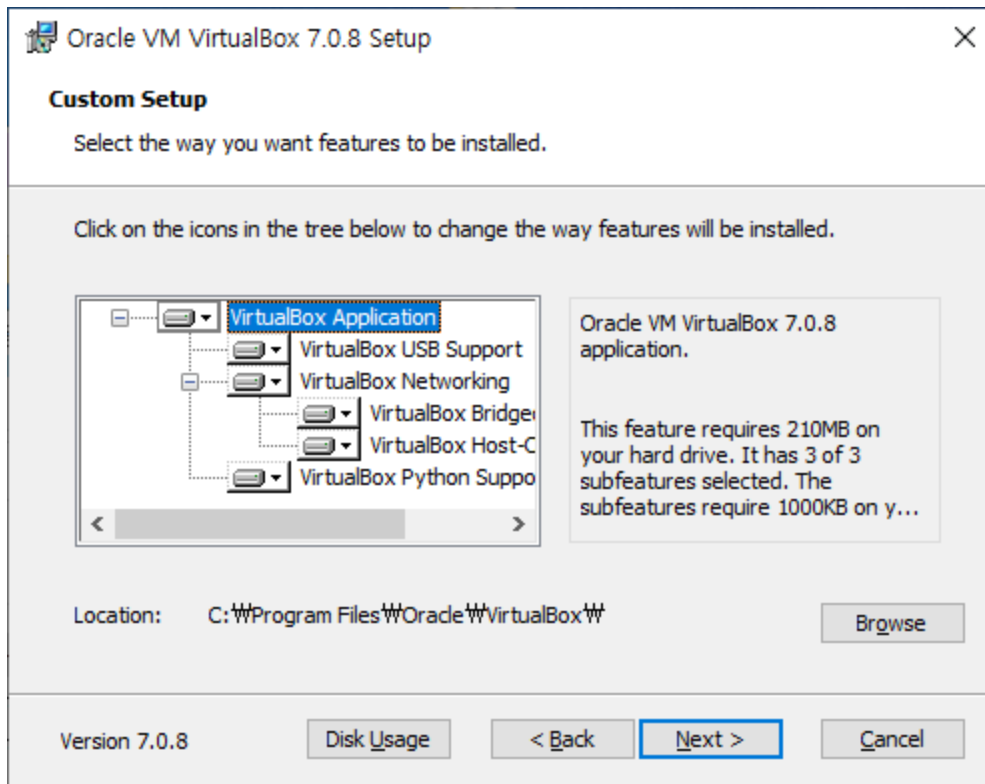
- 1.3. Launch the VirtualBox installer from Windows Explorer by double-clicking the self-extracting executable. Allow the installer to make changes to your computer, if so prompted.



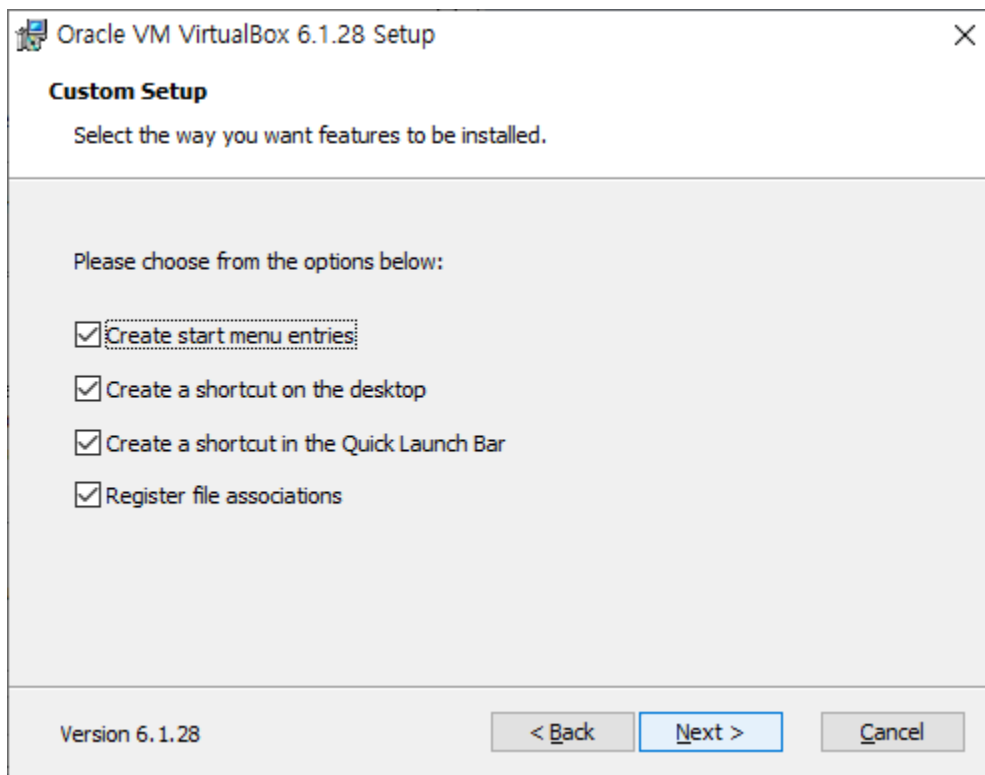
1.4. Once the VirtualBox installation wizard appears, click the **Next** button.



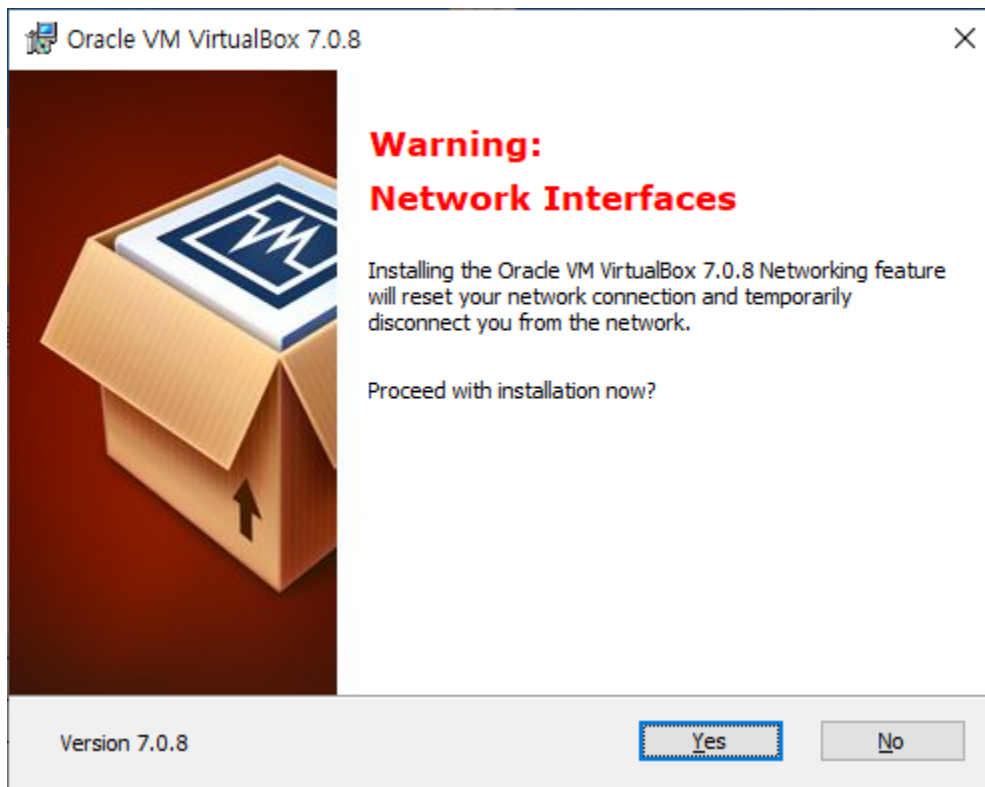
1.5. You may accept all the installation defaults, although you may wish to change the installation location on your development platform using the **Browse** button. If the options are acceptable, click the **Next** button.



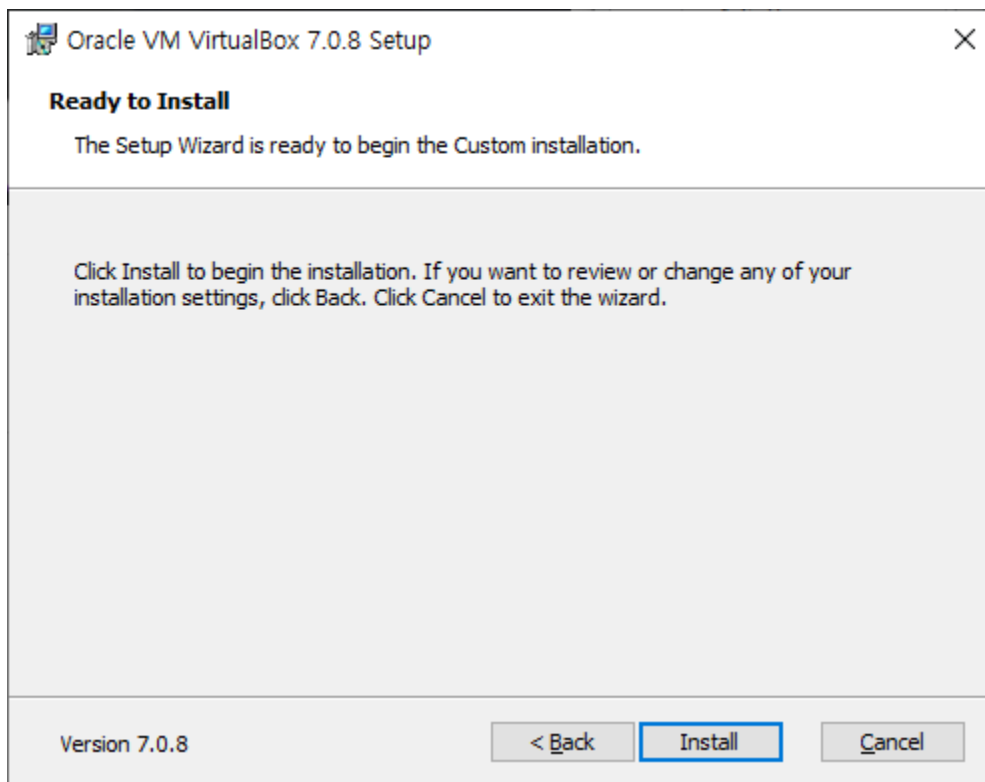
1.6. You may again accept the default options and click the **Next** button.



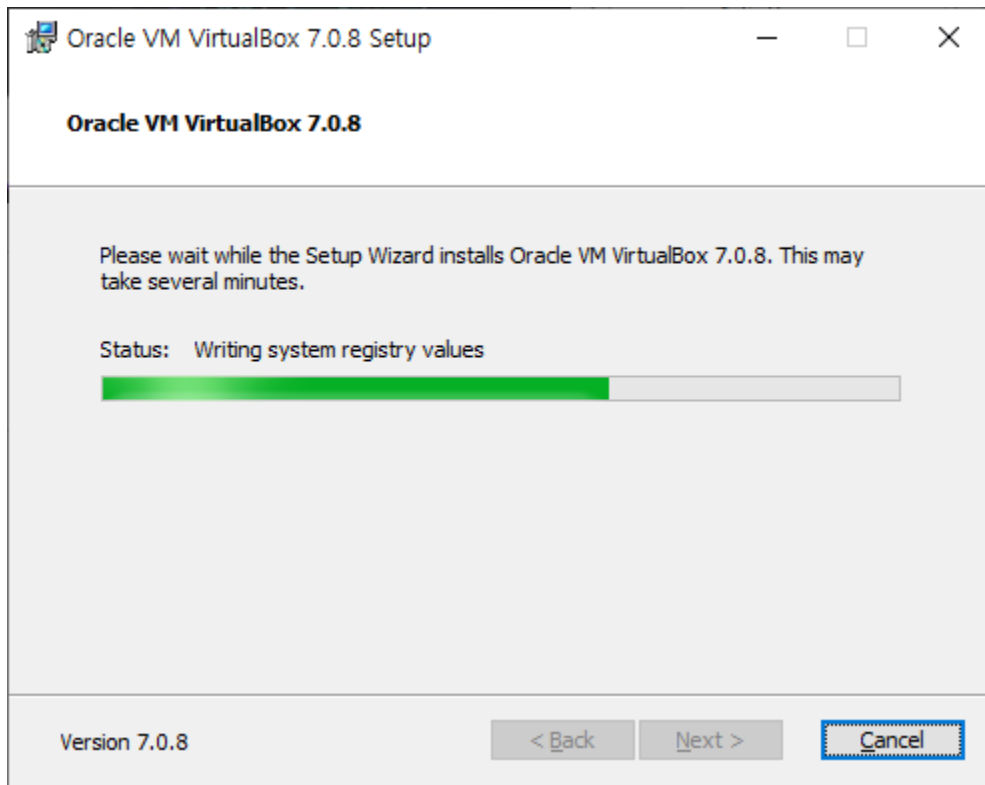
1.7. Click the **Yes** button to continue with the installation wizard.



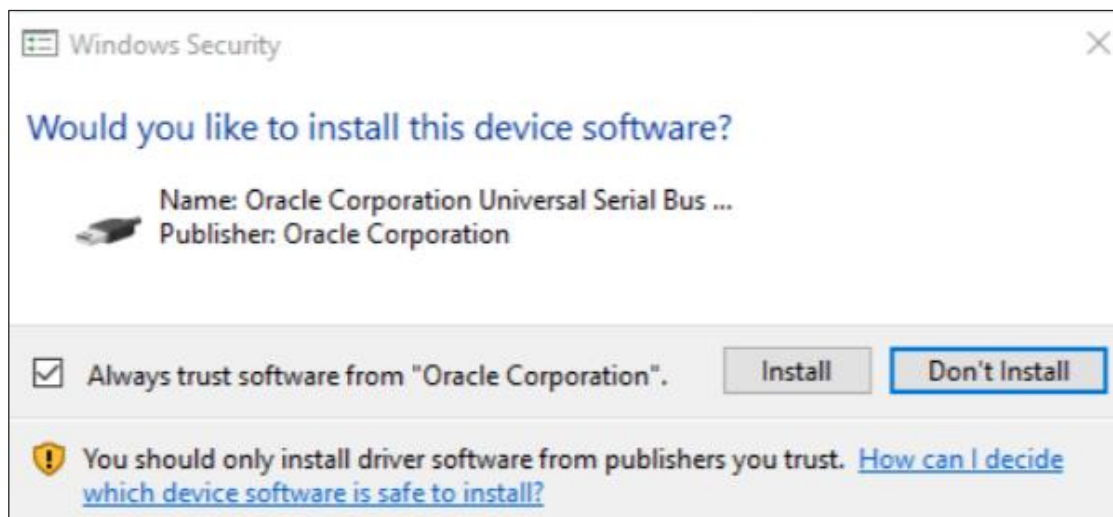
1.8. Click the **Install** button to load VirtualBox to your development system.



- 1.9. During the installation you may receive prompts to authorize installation of various components. If prompted, allow the installer to make changes to your system, including installation of the USB interface and Network adapters.



- 1.10. If you are asked to install the Oracle Corporation Universal Serial Bus device driver, or Oracle Corporation Network Adapters/Network Service, choose to install them.

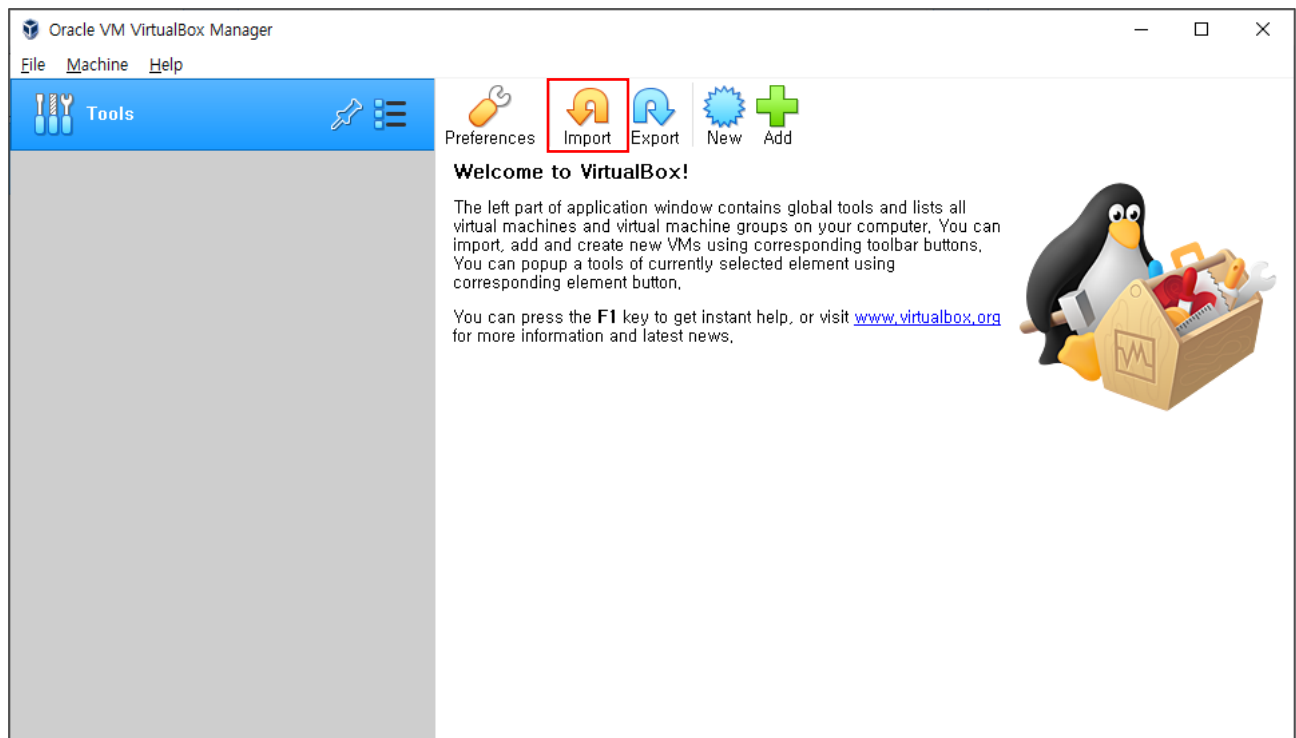


- 1.11. Click the **Finish** button to complete the installation. Leave the checkbox enabled so VirtualBox will start after the installer finishes.



## 2. Using OVA Files with VirtualBox

### 2.1. Click the **Import** button.





- 2.2. Click the **folder icon** to find the path where the OVA file is located.  
(file name: VM 2.2.ova)


**Appliance to import**

Please choose the source to import appliance from. This can be a local file system to import OVF archive or one of known cloud service providers to import cloud VM from.

Source:

Please choose a file to import the virtual appliance from. VirtualBox currently supports importing appliances saved in the Open Virtualization Format (OVF). To continue, select the file to import below.

File:



### 2.3. Click the **Next** button.

**Appliance to import**

Please choose the source to import appliance from. This can be a local file system to import OVF archive or one of known cloud service providers to import cloud VM from.

Source: Local File System

Please choose a file to import the virtual appliance from. VirtualBox currently supports importing appliances saved in the Open Virtualization Format (OVF). To continue, select the file to import below.

File: C:\Users\ZZ01HE766W\Downloads\VM 2.2.ovf

Expert Mode Next Cancel

### 2.4. You may accept the default options and click the **Import** button.

**Appliance settings**

These are the virtual machines contained in the appliance and the suggested settings of the imported VirtualBox machines. You can change many of the properties shown by double-clicking on the items and disable others using the check boxes below.

Virtual System 1	
Name	VM 2.0
Guest OS Type	Red Hat (64-bit)
CPU	6
RAM	8192 MB
DVD	<input checked="" type="checkbox"/>
USB Controller	<input checked="" type="checkbox"/>
Sound Card	<input checked="" type="checkbox"/> ICH AC97
Network Adapter	<input checked="" type="checkbox"/> Intel PRO/1000 MT Desktop (82540EM)
Storage Controller (IDE)	PIIX4
Storage Controller (IDE)	PIIX4
Storage Controller (SATA)	AHCI
Virtual Disk Image	VM 2.2-disk001.vmdk
Base Folder	C:\Users\ZZ01HE766W\VirtualBox VMs
Primary Group	/

Machine Base Folder: C:\Users\ZZ01HE766W\VirtualBox VMs

MAC Address Policy: Include only NAT network adapter MAC addresses

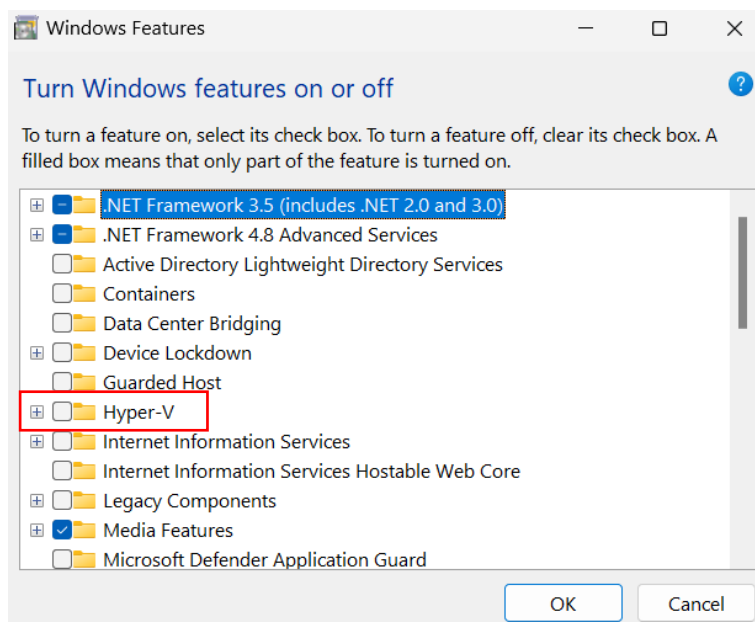
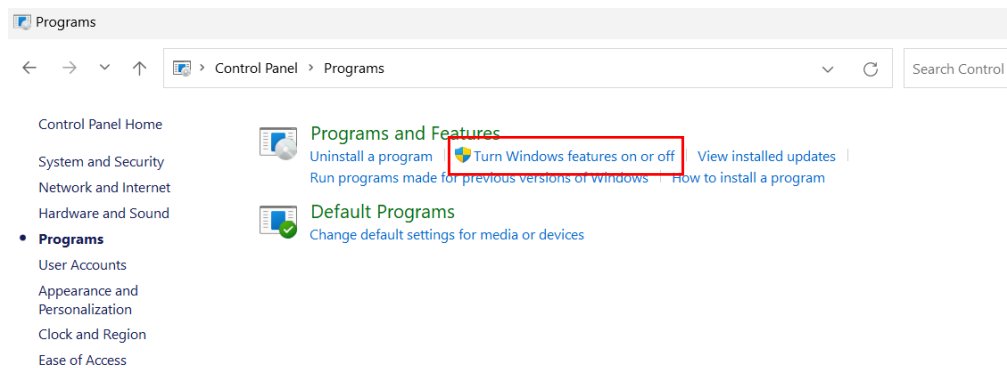
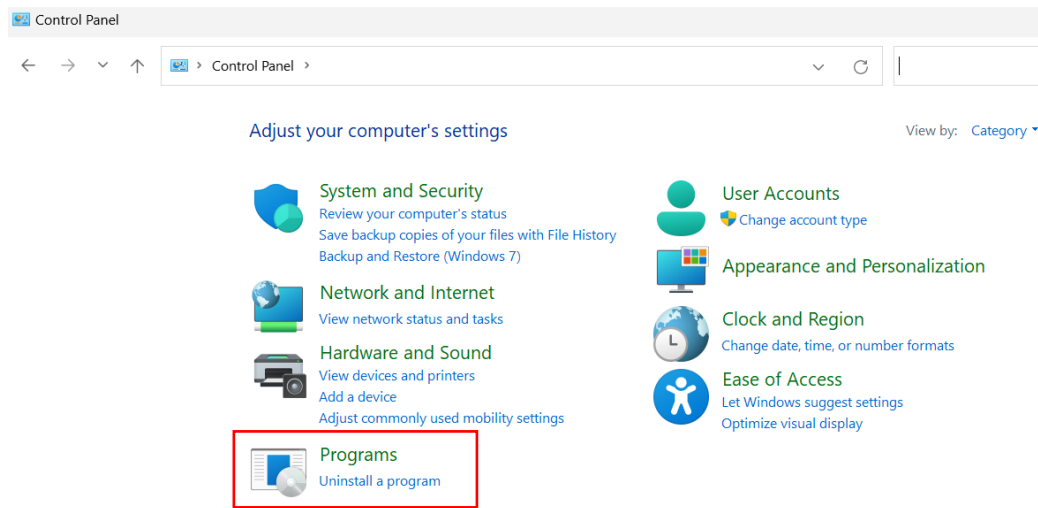
Additional Options: ☒ Import hard drives as VDI

Appliance is not signed

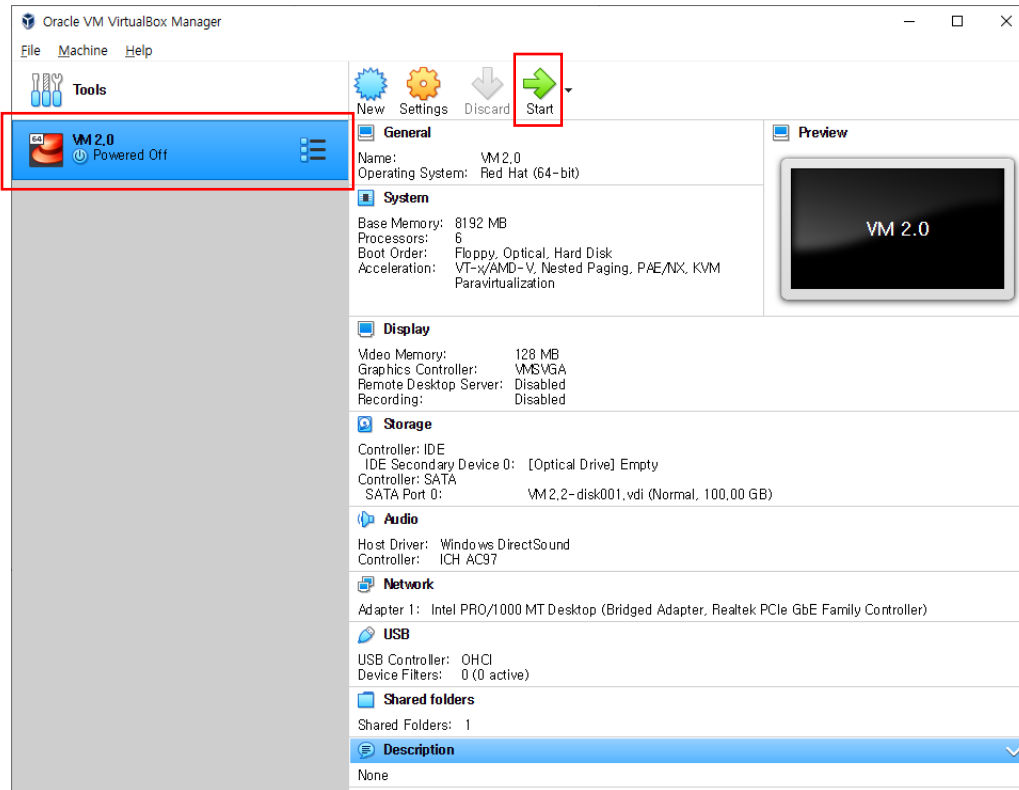
Restore Defaults Import Cancel

### 3. Starting the Lab Virtual Machine

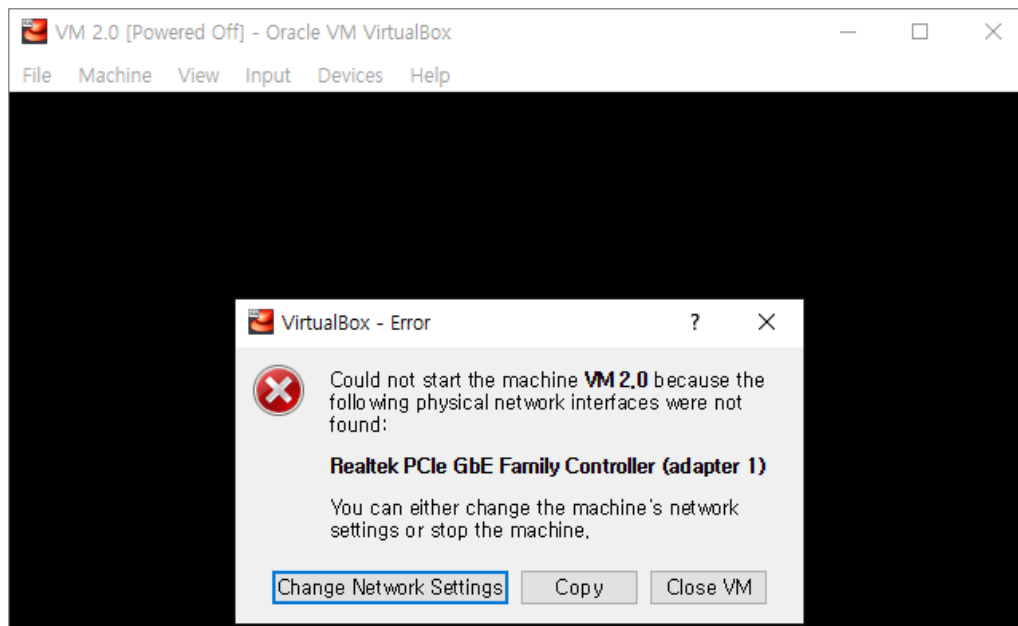
#### 3.1. Turn off “Hyper-V” option



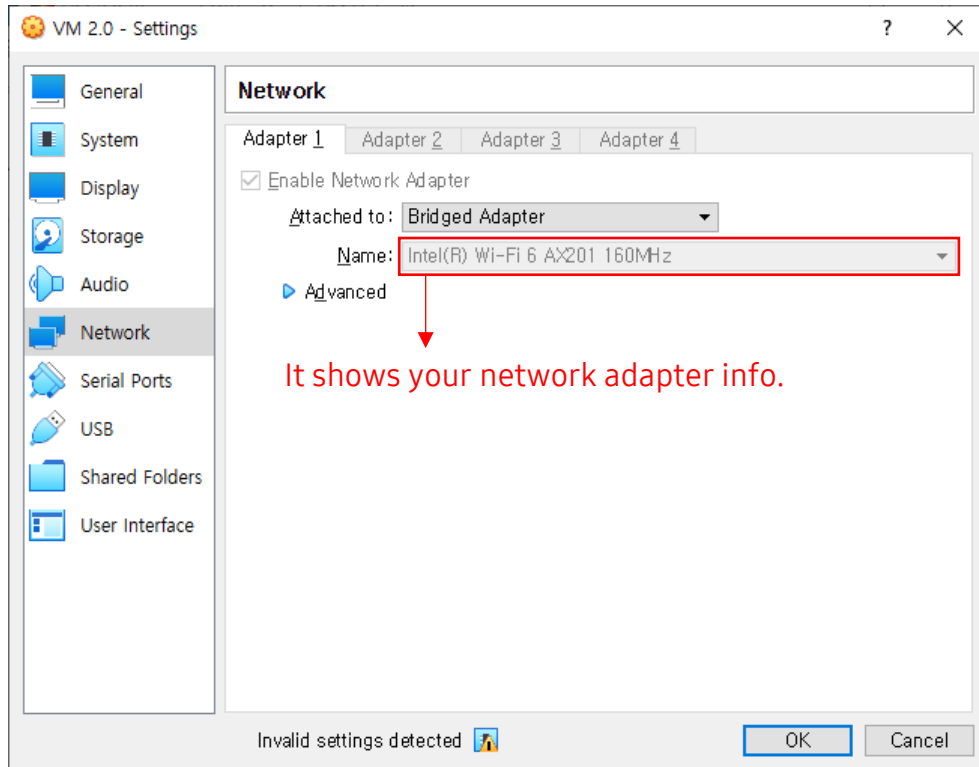
### 3.2. Select the “VM 2.0” virtual machine and Click the Start button



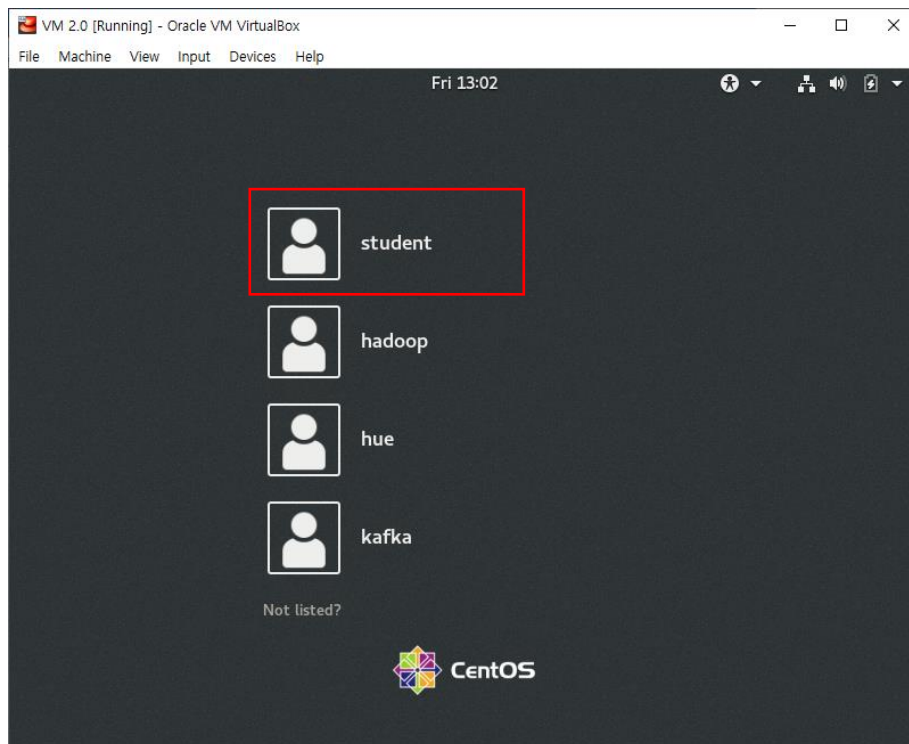
### 3.3. If there is an error like below, click Change Network Setting button



3.4. You may accept the default network setting and click the **OK** button.



3.5. From the login screen, select student. The login password is “student”.




## Lab 2: Working with HDFS

---

In this lab, you will explore and work with HDFS using command and Web UI.

### 1. Starting the Hadoop Services – HDFS and YARN

In order to start using the Hadoop service, we have to start the Hadoop service daemons. We do this as user “hadoop” from our Linux environment.

1.1. Open a terminal by selecting the terminal icon from the bottom left panel  or right clicking anywhere on the desktop.

1.2. Change user to “hadoop” using the **su** command. The password is “hadoop”

```
su – hadoop
```

1.3. Change the working directory to `~/hadoop/sbin` and execute the `start-dfs.sh` script to start HDFS services.

```
cd ~/hadoop/sbin  
./start-dfs.sh
```

You will get an output similar to below.

```
[hadoop@localhost sbin]$ ./start-dfs.sh  
Starting namenodes on [localhost]  
Starting datanodes  
Starting secondary namenodes [localhost.localdomain]
```

You have successfully started the namenode and datanode daemon for HDFS services.

1.4. Execute the `start-yarn.sh` script to start YARN services.

```
./start-yarn.sh
```

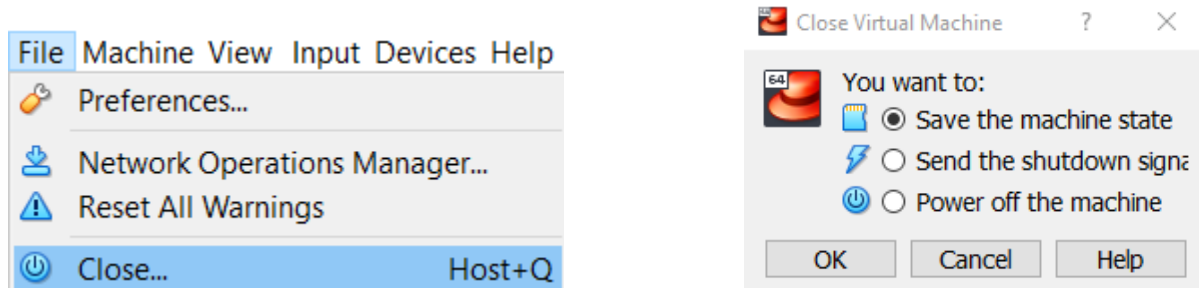
1.5. You will get an output similar to below.

```
[hadoop@localhost sbin]$ ./start-yarn.sh  
Starting resourcemanager  
Starting nodemanagers
```

You have successfully started resourcemanager and nodemanagers daemon for YARN services.

- 1.6. You may now exit out of the session as hadoop user. Do this by entering exit from the terminal.

If the Linux OS is shut down at any time, you will have to repeat the above steps to start the HDFS and YARN services. In order to avoid this, you should not turn off the machines. When done with the Lab, exit the virtual machine using the **File > Close** menu. Select the **Save the machine state** option to exit.



## 2. Working with Linux and HDFS Home directories.

- 2.1. From a terminal as user **student**, enter the following command:

```
hdfs dfs -ls /
```

Notice that there are several directories. There is a /tmp directory where all users can read and write files. There is also a /user directory. This is where the user home directories for HDFS reside. When using HDFS, each user can have a local Linux home directory as well as a HDFS home directory. In CentOS, the local Linux home directory is usually /home/username. For HDFS, the home directory is located at /user/username.

- 2.2. Explore further, the user home directories in HDFS.

```
hdfs dfs -ls /user
hdfs dfs -ls /user/student
```

Notice that the HDFS home directory for student is empty for now.

- 2.3. Explore user home directories for Linux.

```
cd /home
ls -l
```

```
cd /home/student  
ls -l  
cd /home/
```

#### 2.4. Create a new Linux user. Create HDFS and Linux home directories for the user.

A common task for hadoop administrators is to create new users. When doing so, usually the user's home directory for both Linux and HDFS are created.

2.4.1. Create Linux user and home directory. You will require super user privileges to do so. Prefixing our commands with `sudo` allows us to execute privileged commands. We can use `sudo` because `student` is part of the `wheel` group that has been given this capability.

```
sudo useradd student2 -m
```

Set the password for the new user. For simplicity, we will use "student2" as the password. You may receive a warning that the password is too short or that the password contains elements from the user name. Normally, you want to make sure that you have a more secure password, however, for our Lab, this password will suffice.

```
sudo passwd student2
```

Next, add `student2` to the `wheel` group.

```
sudo usermod -aG wheel student2
```

Test the new Linux account by logging in as **student2**. Remember, the password is "student2". Verify that a Linux home directory has been created.

```
su - student2
```

Enter "student2" as the password.

Next, change to home directory, get a listing of the directory (empty for now), print the current working directory, and finally print the groups that `student2` belongs to.



```
cd
ls -l
pwd
groups
```

#### 2.4.2. Create HDFS home directory for user student2

**Leave the student2 terminal open and open another terminal.** We will need to issue commands as hadoop user in order to make changes to HDFS. Ordinary users may make changes to within their HDFS home directory only. However, they may have been granted additional privileges through security policies.

```
su - hadoop
```

Enter “hadoop” as the password if necessary. We will make a new directory at /user/student2. Next, we will change the owner (chown) of that directory to student2.

```
hdfs dfs -mkdir /user/student2
hdfs dfs -chown student2 /user/student2
```

Now, **go back to the terminal for student2** and verify that a new HDFS home directory has been created for that user.

```
hdfs dfs -ls /user/student2
```

You will not actually get any listing since the directory is empty. However, if the directory did not exist, you would get an error message. Go ahead and try getting a listing for /user/student3 This command will return an error “No such file or directory”.

**Clean up your work by exiting out of all the terminals. Keep one terminal open as user student for the next labs in the next section.**

### 3. Exploring and working with HDFS directories.

#### 3.1. Create a subdirectory in the student HDFS home directory and name is “MRtest”

```
hdfs dfs -mkdir MRtest  
hdfs dfs -ls /user/student
```

#### 3.2. Explore a book located in a subdirectory under your home directory.

There is a copy of the book “Alice’s Adventures in Wonderland” by Lewis Carrol in /home/student/data directory. Navigate to the directory and explore the file. The ~ (tilde) is a shortcut for the path to your Linux home directory. In our case, this will be /home/student. So, cd ~/Data is a shortcut for cd /home/student/Data

```
cd ~/Data  
less alice_in_wonderland.txt
```

The **less** command allows us to view a long file without it scrolling too far. Hit enter to continue scrolling. Press “q” to quit and stop viewing.

#### 3.3. Put the book file under the MRtest directory

```
hdfs dfs -put \  
/home/student/Data/alice_in_wonderland.txt \  
/user/student/MRtest/
```

Our command to “put” the book is quite long. In Linux, we can break up long commands into multiple lines using the \ at the end of the line. This tells Linux that the command isn’t finished and there is more coming.

Recall from the lectures the syntax for the put subcommand is **hdfs dfs -put *source destination***. The source and destination can be actual files or directories. So, in the above command, we are requesting HDFS to put the alice\_in\_wonderland.txt file into the HDFS /user/student/MRtest directory.

Verify that the file has been copied to the HDFS destination.

```
hdfs dfs -ls /user/student/MRtest
```

#### 4. Explore HDFS with the Web UI

4.1. Open Firefox Web browser by selecting is from the panel on the bottom left.



Open the Namenode Web UI at <http://localhost:50070>. You can also select the bookmark from the browser.

4.2. Explore Overview tab

From this tab, we can see the overview information for the configured HDFS. We also can get a summary of the disk use statistics and the health of the datanodes.

4.3. Explore Datanodes and Datanode Volume Failures tab

From these tabs, we can view health and statistics of the datanodes. Because our Lab environment is not actually a cluster, we will only see 1 datanode.

4.4. Explore Snapshot tab

It is possible to take snapshots of directories in HDFS. If there are any snapshots saved, we can review information regarding them from this tab.

4.5. Explore Startup Progress tab

We can review the startup information for HDFS services, including the loading of the fsimage and edits file.

**BONUS:** Note the directory and filename of the fsimage and edits file. Open a terminal and login has hadoop. Navigate to the directory where the fsimage and edits file is saved. There are many versions of fsimage and edits file. These files were created and checkpointed.

4.6. From Utilities, Browse the file system

- Browse the file system
- Logs
- Log Level
- Metrics
- Configuration
- Process Thread Dump
- Network Topology

# Overview 'localhost:9000' (✓active)

Started:	Thu Jul 22 23:16:14 +0900 2021
Version:	3.3.1, ra3b9c37a397ad4188041dd80621bdeefc46885f2
Compiled:	Tue Jun 15 14:13:00 +0900 2021 by ubuntu from (HEAD detached at release-3.3.1-RC3)
Cluster ID:	CID-e4f41fff-6a81-4852-9933-c6c7291af7d9
Block Pool ID:	BP-1037786283-127.0.0.1-1626250653246

4.6.1. Navigate to the HDFS directory where you previously copied the `alice_in_wonderland.txt` file

## Browse Directory

Show
25
entries

Search:

	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
<input type="checkbox"/>	-rw-r--r--	<a href="#">student</a>	<a href="#">student</a>	170.23 KB	Jul 23 04:07	<a href="#">1</a>	128 MB	<a href="#">alice_in_wonderland.txt</a>

Showing 1 to 1 of 1 entries

Previous

1

Next

4.6.2. Click on the file to bring up the block information pop-up window.

In our pseudo-distributed Hadoop environment, we have set the replication factor to 1 since there is only on machine in the cluster. When you click on the Block Information, you will only see one block. In a real cluster, the replication factor is normally set to 3 and we would be able to see information regarding all 3 blocks including the location of

File information - alice\_in\_wonderland.txt

Download

Head the file (first 32K)

Tail the file (last 32K)

Block information -- Block 0

Block ID: 1073742392

Block Pool ID: BP-1037786283-127.0.0.1-1626250653246

Generation Stamp: 1568

Size: 174313

Availability:

localhost

those blocks. Take note of the Block ID and the Block Pool ID. We will navigate to the directory where HDFS actually stores these blocks.

4.6.3. Open a terminal and login as user **hadoop**

4.6.4. Use the Linux command **find** to search for the Block Pool ID

```
sudo find / -name <name of the Block Pool ID> -print
```

The above command instructs Linux to start at the / (root) directory and look for a file or directory with the given name and print out the information. The command will go to every subdirectory from / root. Your output will look similar to below.

```
[hadoop@localhost ~]$ sudo find / -name BP-1037786283-127.0.0.1-1626250653246 -print
[sudo] password for hadoop:
/home/hadoop/hadoopdata/hdfs/datanode/current/BP-1037786283-127.0.0.1-1626250653246
find: '/root/Document/My Music': Operation not permitted
find: '/root/Document/My Pictures': Operation not permitted
find: '/root/Document/My Videos': Operation not permitted
[hadoop@localhost ~]$
```

Don't worry about the 3 directories that Operation was not permitted. Those directories are mounted directories connecting your PC folders to the Linux system by VirtualBox.

4.6.5. Navigate to the directory where the Block Pool ID was found and explore.

```
cd <result of your find from above step>
sudo find . -name *<BLOCK ID>* -print
```

We will use the **find** command again to look for the data blocks. This time, we will tell **find** to start looking from the current directory with the dot notation(.). You should see two files found. One of the files will contain the meta data. This is a binary file and you won't be able to see its content. However, the other file is a text file and the actual contents that Hadoop has saved for the `alice_in_wonderland.txt` file save in HDFS.

The output of the above command will be similar to below.

```
[hadoop@localhost ~]$ cd /home/hadoop/hadoopdata/hdfs/datanode/current/BP-1037786283-127.0.0.1-1626250653246
[hadoop@localhost BP-1037786283-127.0.0.1-1626250653246]$ sudo find . -name *1073742392* -print
./current/finalized/subdir0/subdir2/blk_1073742392_1568.meta
./current/finalized/subdir0/subdir2/blk_1073742392
[hadoop@localhost BP-1037786283-127.0.0.1-1626250653246]$ cat ./current/finalized/subdir0/subdir2/blk_1073742392
```

Use the **less** or **cat** command to view the text file.

```
less <path to text file similar to ./current/finalized.....>
```

## Lab 3: Working with YARN/MapReduce

---

We will use the `alice_in_wonderland.txt` file that has been saved to HDFS to run a wordcount MapReduce program.

### 1. Enable YARN historyserver

- 1.1. Open a new terminal as **hadoop** (use `su – hadoop` command) and start the history server

```
cd $HADOOP_HOME/sbin  
mr-jobhistory-daemon.sh start historyserver
```

### 2. Run the wordcount program from the MapReduce examples jar

- 2.1. Open a new terminal as **student**.
- 2.2. Navigate to the following directory.

```
cd $HADOOP_HOME/share/hadoop/mapreduce
```

- 2.3. Execute the `hadoop-mapreduce-examples-3.3.1.jar` with the wordcount option

```
hadoop jar ./hadoop-mapreduce-examples-3.3.1.jar wordcount \  
MRtest WC_Output
```

- 2.4. While the program is running open Firefox and navigate to the YARN Web UI at <http://localhost:8088>

Click on Applications on the left tab menu. You should see your application running. An application id has been assigned to the job. Click on the link for the application-id



Cluster

About

Nodes

Node Labels

Applications

NEW

NEW SAVING

SUBMITTED

ACCEPTED

RUNNING

FINISHED

FAILED

KILLED

Scheduler

Tools

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running
1	0	0	1	0

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes
1	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum
Capacity Scheduler	[memory-mb (unit=M), vcores]	<memory:1024, vCores:1>

Show 20 entries

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime
application_1626988699569_0002	student	word count	MAPREDUCE		default	0	Fri Jul 23 06:27:36 +0900 2021	Fri Jul 23 06:27:37 +0900 2021

You should get a screen similar to below. Explore the details of the job from this screen. When finished, follow the links to explore the Application Master

Application application\_1626987151625\_0001 - Mozilla Firefox

Application application\_1626987151625\_0001

localhost:8088/cluster/app/application\_1626987151625\_0001

Namenode Web UI YARN Web UI

Logged in as: dr.who

Application application\_1626987151625\_0001

Cluster

About

Nodes

Node Labels

Applications

NEW

NEW SAVING

SUBMITTED

ACCEPTED

RUNNING

FINISHED

FAILED

KILLED

Scheduler

Tools

Kill Application

Application Overview

User: student

Name: word count

Application Type: MAPREDUCE

Application Tags:

Application Priority: 0 (Higher Integer value indicates higher priority)

YarnApplicationState: RUNNING: AM has registered with RM and started running.

Queue: default

FinalStatus Reported by AM: Application has not completed yet.

Started: Fri Jul 23 05:58:58 +0900 2021

Launched: Fri Jul 23 05:58:59 +0900 2021

Finished: N/A

Elapsed: 4sec

Tracking URL: ApplicationMaster

Log Aggregation Status: DISABLED

Application Timeout (Remaining Time): Unlimited

Diagnostics:

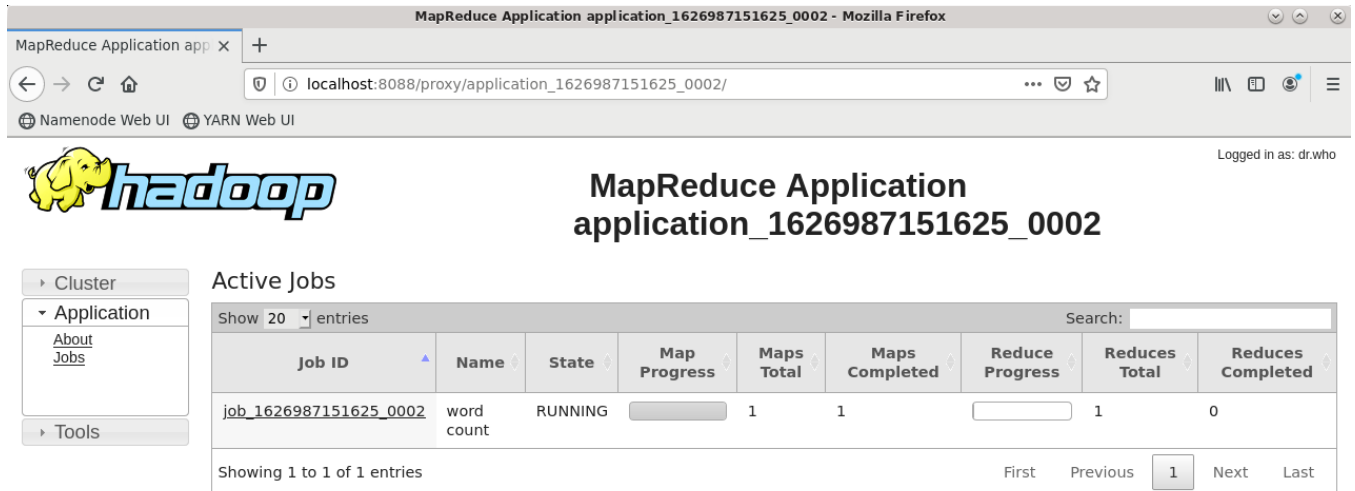
Unmanaged Application: false

Application Node Label expression: <Not set>

AM container Node Label expression: <DEFAULT\_PARTITION>

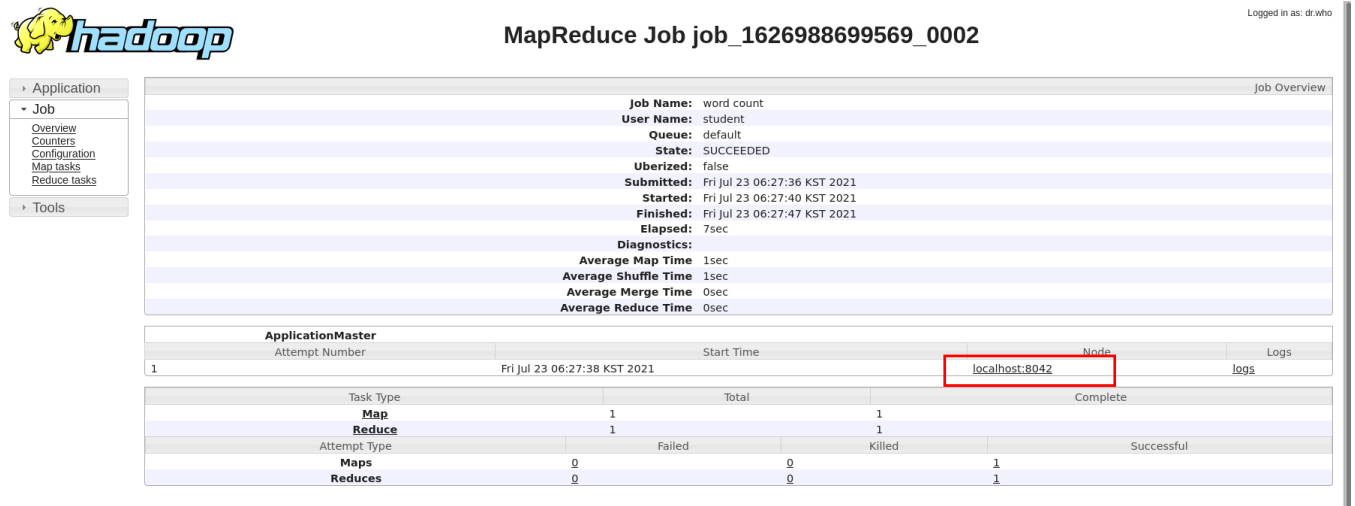


Screen from the Application Master while job is still running.



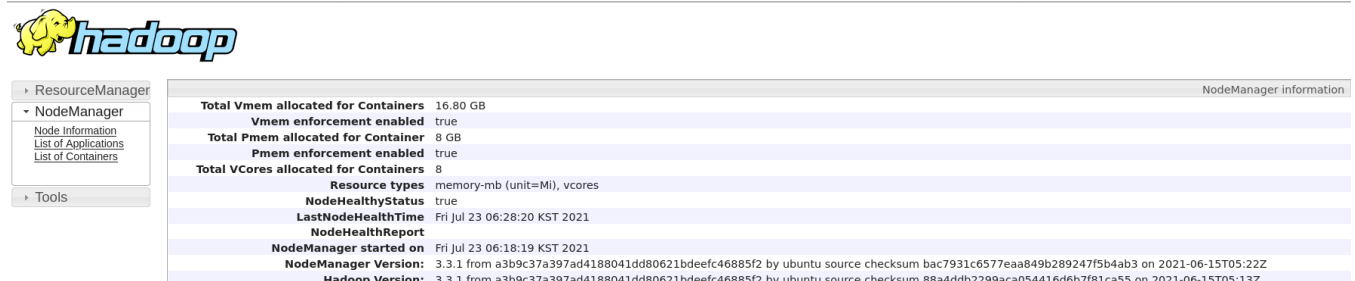
The screenshot shows the Hadoop MapReduce Application Master web UI in a Mozilla Firefox browser. The page title is "MapReduce Application application\_1626987151625\_0002". The URL is "localhost:8088/proxy/application\_1626987151625\_0002/". The page is logged in as "dr.who". The Hadoop logo is visible on the left. The main heading is "MapReduce Application application\_1626987151625\_0002". On the left, there is a sidebar with "Cluster" and "Application" sections. The "Application" section is expanded, showing "About Jobs" and "Tools". The "Active Jobs" section is displayed, showing a table with columns: Job ID, Name, State, Map Progress, Maps Total, Maps Completed, Reduce Progress, Reduces Total, and Reduces Completed. The table contains one entry for job "job\_1626987151625\_0002" with name "word count" and state "RUNNING". The progress bars for Map and Reduce are shown. The bottom of the table indicates "Showing 1 to 1 of 1 entries" and navigation links: First, Previous, 1, Next, Last.

If the job is still running, you will see a screen similar to above. If the job has already completed, you will see a screen similar to below. Click on the link to the node that executed the job.



The screenshot shows the Hadoop MapReduce Job web UI in a Mozilla Firefox browser. The page title is "MapReduce Job job\_1626988699569\_0002". The URL is "localhost:8088/proxy/job\_1626988699569\_0002/". The page is logged in as "dr.who". The Hadoop logo is visible on the left. The main heading is "MapReduce Job job\_1626988699569\_0002". On the left, there is a sidebar with "Application" and "Job" sections. The "Job" section is expanded, showing "Overview", "Counters", "Configuration", "Map tasks", and "Reduce tasks". The "Job Overview" section is displayed, showing a table with columns: Job Name, User Name, Queue, State, Uberized, Submitted, Started, Finished, Elapsed, and Diagnostics. The table contains one entry for job "job\_1626988699569\_0002" with name "word count" and state "SUCCEEDED". The progress bars for Map and Reduce are shown. The bottom of the table indicates "Showing 1 to 1 of 1 entries" and navigation links: First, Previous, 1, Next, Last.

Each node will show the container resources allocated to the job.



The screenshot shows the Hadoop NodeManager web UI in a Mozilla Firefox browser. The page title is "NodeManager information". The URL is "localhost:8088/proxy/nodemanager/". The page is logged in as "dr.who". The Hadoop logo is visible on the left. The main heading is "NodeManager information". On the left, there is a sidebar with "ResourceManager" and "NodeManager" sections. The "NodeManager" section is expanded, showing "Node Information", "List of Applications", and "List of Containers". The "Node Information" section is displayed, showing a table with columns: Resource types, NodeHealthyStatus, LastNodeHealthTime, NodeHealthReport, NodeManager started on, NodeManager Version, and Hadoop Version. The table contains one entry for node "localhost:8042" with name "word count" and state "SUCCEEDED". The progress bars for Map and Reduce are shown. The bottom of the table indicates "Showing 1 to 1 of 1 entries" and navigation links: First, Previous, 1, Next, Last.

You may wish to re-run the job. If so, you will have to remove the output directory. The -r option of the rm (remove) command tells Hadoop to recursively delete any subfolders as well.

```
hdfs dfs -rm -r WC_Output
```

**END OF LAB**