ECE4721J 24 SU Lab 2

Hadoop Cluster Setup

> Contributor: ECE4721J 24SU Teaching Group

Disclaimer

Please note that this manual is based on our teaching group's experience during the 23SU semester. We encourage you to develop your own strategy for set up after having a deeper understanding of Hadoop cluster setup.

Target of the Lab Sessions

1. Session 1: Work individually to set up your Hadoop cluster in single node (or single-node in a pseudo-distributed mode to be exact)
2. Session 2: Work in a group, properly setup Hadoop with multi node setup.
3. Other stuffs:
   - Learn how to submit homework on Gitea
   - Pick up and check whether cables work

Before you start

1. Please ensure that you have a Linux system installed on your computer, either directly on the disk or within a virtual machine like VMWare. Please note that our teaching group has limited experience with Mac OS, Windows, or other operating systems, so we may not be able to provide much assistance if you encounter issues with these systems.
2. Kindly note that the version of Hadoop used in our course is 3.2.2. Some links in the lab manual on canvas may refer to Hadoop 2.x installation, which is not applicable. Also, please

be aware that Hadoop has a newer version of 3.3.x, which is not used in our course. Please ensure that you are using the correct version of Hadoop to avoid any compatibility issues.

3. This manual mainly refers to ref.1 and you can treat it as an extended version of ref.1
4. For MacOS users, you can either intall a virtual linux machine and follow this guidline. Or you can follow another instruction to setup cluster directly on mac. Credit to Yinuo.

Part I: Single node setup

Single Node Setup is ~~quite easy~~. You just need to follow the instructions from Hadoop official website (available in l2 manual). To make your life easier, Xinhe made a demonstration video https://sjtu.feishu.cn/minutes/obcnh9s7h5q4xp8684491t6z?from=auth_notice .
Also, it is very easy to switch back to single node after you properly set up the multi-node version.

Part II: Multi node setup

Some of the steps are already done in single node setup, you may skip them if necessary.

Dependency softwares installation

1. Properly install and setup `ssh` and `pdsh`
   In your virtual machine, install ssh and pdsh with commands like

```
sudo apt install ssh
sudo apt install pdsh
```

add the following command to your shell configuration file (`~/.bashrc` for example):

```
export PDSH_RCMD_TYPE=ssh
```

If you don't know how to use `nano` or `vim` or other tools to edit files in linux, you can ask your teammates or TA for help during lab.
After that, use `tail ~/.bashrc` to see whether you have this statement in your configuration file.

2. Generate a ssh key

We assume all of you have done that. So this part is omitted. However, we recommend you to have a separate ssh key for the Hadoop cluster.

3. Install Java

Hadoop relies compile and run based on java 8.

```
sudo apt install openjdk-8-jdk
```

If you happen to have another version of java installed before, you can use following command to specify the default java version:

```
sudo update-alternatives --config java
sudo update-alternatives --config javac #for jdk, if you want
```

4. Download and install hadoop source
   You can download source code of Hadoop from Apache Hadoop 3.2.2 Release Page.

After download, use `tar` command to decompress the file. For the sake of convenience, in this manual, we recommend you to move the extracted folder to `/usr/local` and name as `hadoop`. This will save you lots of time to solve the folder position issue.

```
hadoopuser@hadoop-master:/usr/local$ ls
bin  etc  games  go  hadoop  include  java  lib  man  sbin  share  src
```

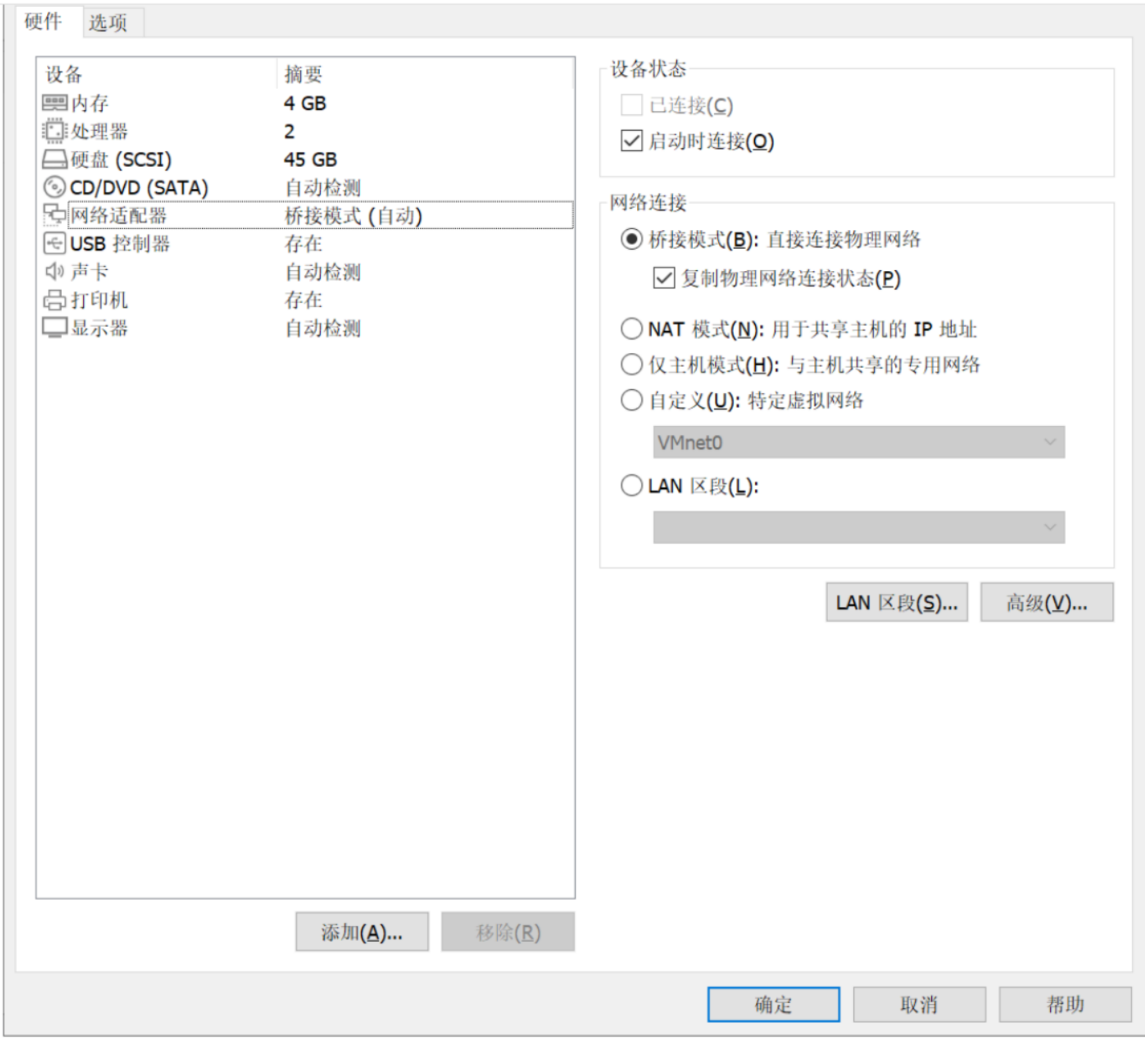
Hadoop configurations (All Nodes)

1. Network setup

Set your Virtual Machine network connection to the host machine as **Bridged**.

VMware WorkStation

- Right click the **virtual machine** you intended to set network connection for and click **settings（设置)** in the pop-up menu.
- In the pop-up window, choose **Network Adapter(网络适配器)** and check the bridged mode button.



设备	摘要
内存	4 GB
处理器	2
硬盘 (SCSI)	45 GB
CD/DVD (SATA)	自动检测
网络适配器	桥接模式 (自动)
USB 控制器	存在
声卡	自动检测
打印机	存在
显示器	自动检测

设备状态

□ 已连接(C)

☑ 启动时连接(O)

网络连接

● 桥接模式(B): 直接连接物理网络

☑ 复制物理网络连接状态(P)

○ NAT 模式(N): 用于共享主机的 IP 地址

○ 仅主机模式(H): 与主机共享的专用网络

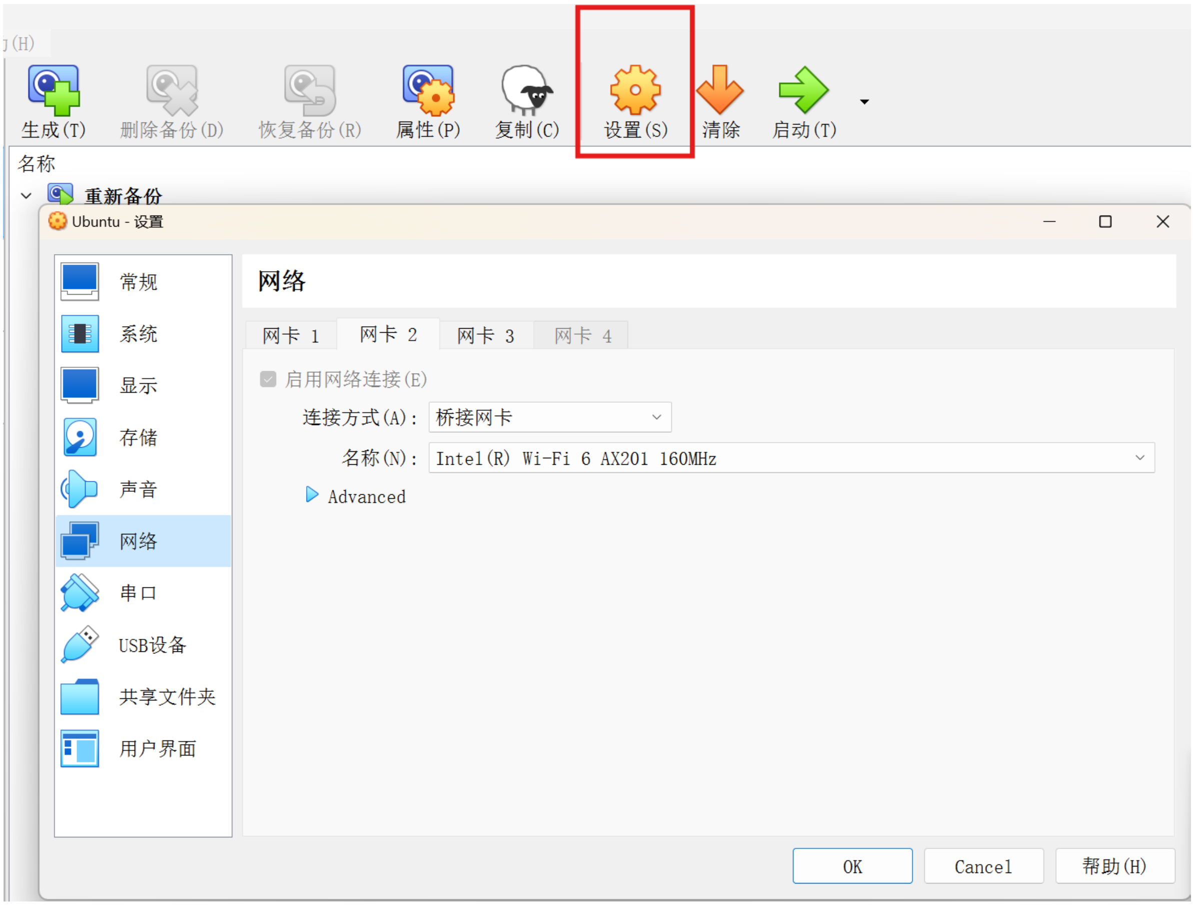
○ 自定义(U): 特定虚拟网络

VMnet0

○ LAN 区段(L):

VirtualBox

You can add multiple virtual network cards.



☑ 启用网络连接(E)

连接方式(A)： 桥接网卡

名称(N)： Intel(R) Wi-Fi 6 AX201 160MHz

2. Hadoop configuration files

- set Java path for Hadoop
  In `/usr/local/hadoop/etc/hadoop/`, we have a environment file `./hadoop-env.sh`, specify the java home for your hadoop there. For example:

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/
```

Search online if you don't know how to find your java home and notice that set it to the java8 home.

- add hadoop command to your path and Java Home to environment file

edit your `/etc/environment` to have the following contents:

```
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/usr/local/hadoop/bin:/usr/local/hadoop/sbin"
JAVA_HOME="/usr/lib/jvm/java-8-openjdk-amd64/jre
```

3. Add Hadoop users

Create a new `hadoopuser` to have access to the hadoop folder.

```
sudo adduser hadoopuser
sudo usermod -aG hadoopuser hadoopuser
sudo chown hadoopuser:root -R /usr/local/hadoop/
sudo chmod g+rwx -R /usr/local/hadoop/
sudo adduser hadoopuser sudo
```

4. ip address

This is the **most critical part** of this lab. Most of the connection issue of hadoop is due to ip address.
We recommend you to use a shared switcher or mobile phone hotstop (be careful with your 5G data amount). The stduents during covid-19 succeeded their installation via SJTU VPN (but we failed last year).

After that, collect all of your group members ip address and edit /etc/hosts file. Following is the example of our file.

```
hadoopuser@hadoop-master:/usr/local$ cat /etc/hosts
127.0.0.1   localhost
#127.0.1.1  zjche-VirtualBox

# The following lines are desirable for IPv6 capable hosts
#::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

# This is the setup on ZJChe's iPhone
#172.20.10.13 hadoop-master #jiache
#172.20.10.3 hadoop-slave-1 #shaoze
#172.20.10.4 hadoop-slave-2 #kezhi
#172.20.10.5 hadoop-slave-3 #hamster

# This is the setup on router in the dorm
192.168.1.3  hadoop-slave-3  # hamster
192.168.1.4  hadoop-master   # jiache
192.168.1.5  hadoop-slave-1  # shawn
192.168.1.2  hadoop-slave-2  # kezhi
```

Also, for every group member, change your hostname to the correspondent name. For `hamster`, it should be `hadoop-slave-3`.

```
hadoopuser@hadoop-master:/etc$ cat /etc/hostname
hadoop-master
```

Then reboot your machine to make the change in effect.

5. share ssh key to enable access withoud passwd

```
ssh-copy-id hadoopuser@hadoop-master
ssh-copy-id hadoopuser@hadoop-slave-1
ssh-copy-id hadoopuser@hadoop-slave-2
ssh-copy-id hadoopuser@hadoop-slave-3
```

If success, you can `ssh hadoopuser@hadoop-slave-1` easily and have access to your group members' computer. Please do not do anything bad. :)

Hadoop configurations (Master Node Only)

We have several configuration files to set here, we will show you our version here.
Some comments in side the configuration file (e.g. `<!-- -->`) is ommited to save space.

/usr/local/hadoop/etc/hadoop/core-site.xml

```
hadoopuser@hadoop-master:~$ cat /usr/local/hadoop/etc/hadoop/core-site.xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<!-- Put site-specific property overrides in this file. -->

<configuration>
        <property>
            <name>fs.defaultFS</name>
            <value>hdfs://hadoop-master:9000</value>
    </property>
    <property>
        <name>hadoop.http.staticuser.user</name>
        <value>hadoopuser</value>
    </property>
</configuration>
```

/usr/local/hadoop/etc/hadoop/hdfs-site.xml

```
hadoopuser@hadoop-master:~$ cat /usr/local/hadoop/etc/hadoop/hdfs-site.xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<!-- Put site-specific property overrides in this file. -->
<!--
<configuration>
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>
</configuration>
-->

<configuration>
<property>
<name>dfs.namenode.name.dir</name><value>/usr/local/hadoop/data/nameNode</value>
</property>
<property>
<name>dfs.datanode.data.dir</name><value>/usr/local/hadoop/data/dataNode</value>
```

```
</property>
<property>
<name>dfs.replication</name>
<value>2</value>
</property>
</configuration>
```

Note the **replication** here which is a common question in exam (default replication level) :).

/usr/local/hadoop/etc/hadoop/workers

- for cluster:

  ```
  hadoopuser@hadoop-master:~$ cat /usr/local/hadoop/etc/hadoop/workers
  hadoopuser@hadoop-master
  hadoopuser@hadoop-slave-1
  hadoopuser@hadoop-slave-2
  hadoopuser@hadoop-slave-3
  #localhost
  ```

- for single node:

  ```
  hadoopuser@hadoop-master:~$ cat /usr/local/hadoop/etc/hadoop/workers
  #hadoopuser@hadoop-master
  #hadoopuser@hadoop-slave-1
  #hadoopuser@hadoop-slave-2
  #hadoopuser@hadoop-slave-3
  localhost
  ```

  Please copy all the configuration files to slave nodes:

```
scp /usr/local/hadoop/etc/hadoop/* hadoop-slave-1:/usr/local/hadoop/etc/hadoop/
scp /usr/local/hadoop/etc/hadoop/* hadoop-slave-2:/usr/local/hadoop/etc/hadoop/
scp /usr/local/hadoop/etc/hadoop/* hadoop-slave-3:/usr/local/hadoop/etc/hadoop/
```

Launch HDFS!

Follow step 23 to add env variables, here's what's in our `.bashrc`.

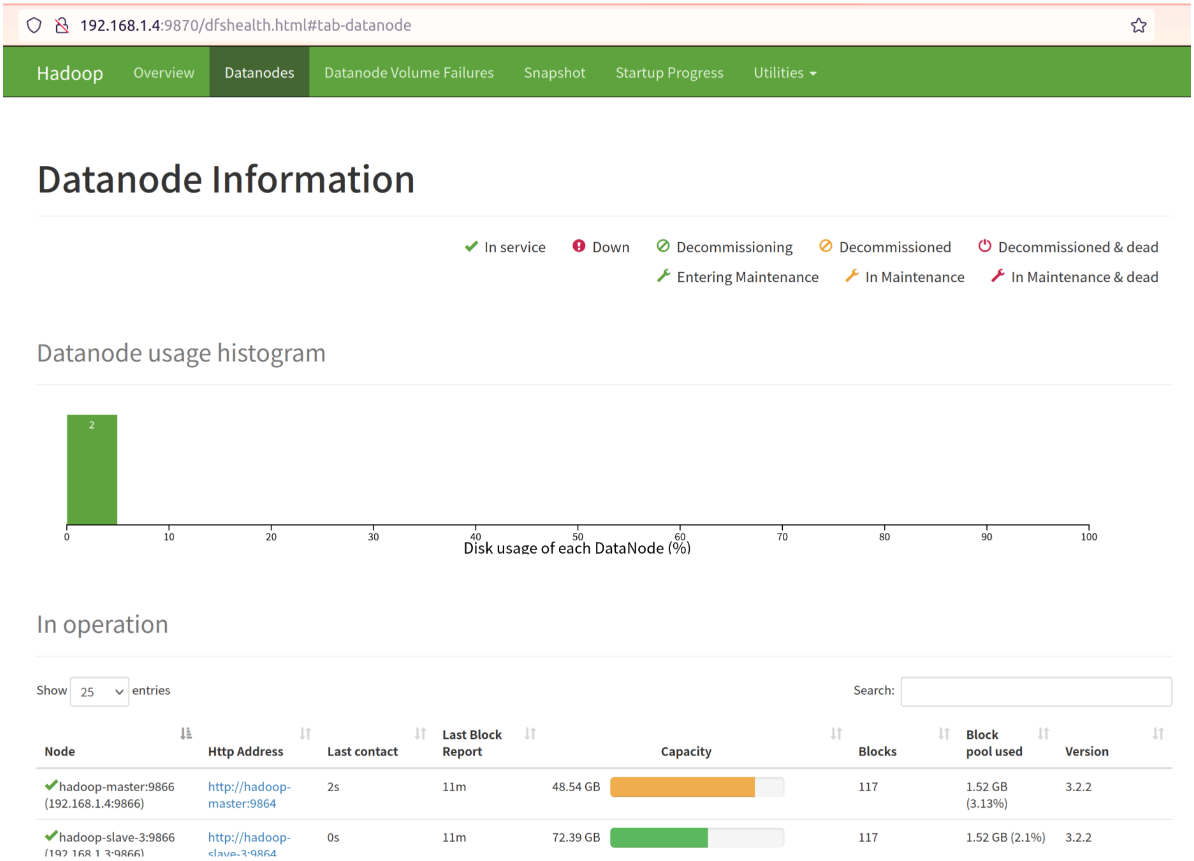
```
export PDSH_RCMD_TYPE=ssh
export HADOOP_HOME="/usr/local/hadoop"
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export HADOOP_HDFS_HOME=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_YARN_HOME=$HADOOP_HOME

# Spark variables
export YARN_CONF_DIR=$HADOOP_HOME/etc/hadoop
export SPARK_HOME=/home/hadoopuser/spark-3.2.4-bin-without-hadoop
export PATH=$PATH:$SPARK_HOME/bin
export LD_LIBRARY_PATH=${HADOOP_HOME}/lib/native
```

```
export PYTHONPATH=/home/hadoopuser/spark-3.2.4-bin-without-
hadoop/python:/home/hadoopuser/spark-3.2.4-bin-without-hadoop/python/lib/py4j-
0.10.9.5-src.zip:$PYTHONPATH
```

Export configurations in effect and start hdfs

```
source /etc/environment
hdfs namenode -format
start-dfs.sh
jps
```

This is what we had last yaer :/



Datanode Information

✓ In service ● Down ⊘ Decommissioning ⊘ Decommissioned ⏻ Decommissioned & dead
Entering Maintenance In Maintenance In Maintenance & dead

Datanode usage histogram

In operation

Show 25 entries Search:

Node	Http Address	Last contact	Last Block Report	Capacity	Blocks	Block pool used	Version
✓hadoop-master:9866 (192.168.1.4:9866)	http://hadoop-master:9864	2s	11m	48.54 GB	117	1.52 GB (3.13%)	3.2.2
✓hadoop-slave-3:9866 (192.168.1.3:9866)	http://hadoop-slave-3:9864	0s	11m	72.39 GB	117	1.52 GB (2.1%)	3.2.2

Yarn configuration

/usr/local/hadoop/etc/hadoop/yarn-site.xml

```
<property>
    <name>yarn.resourcemanager.hostname</name>
    <value>hadoop-master</value>
</property>

<property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
</property>
<property>
    <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
    <value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
```

There can also be possible that you don't need this as yarn will automatically set the launch node as host.

/usr/local/hadoop/etc/hadoop/mapred-site.xml

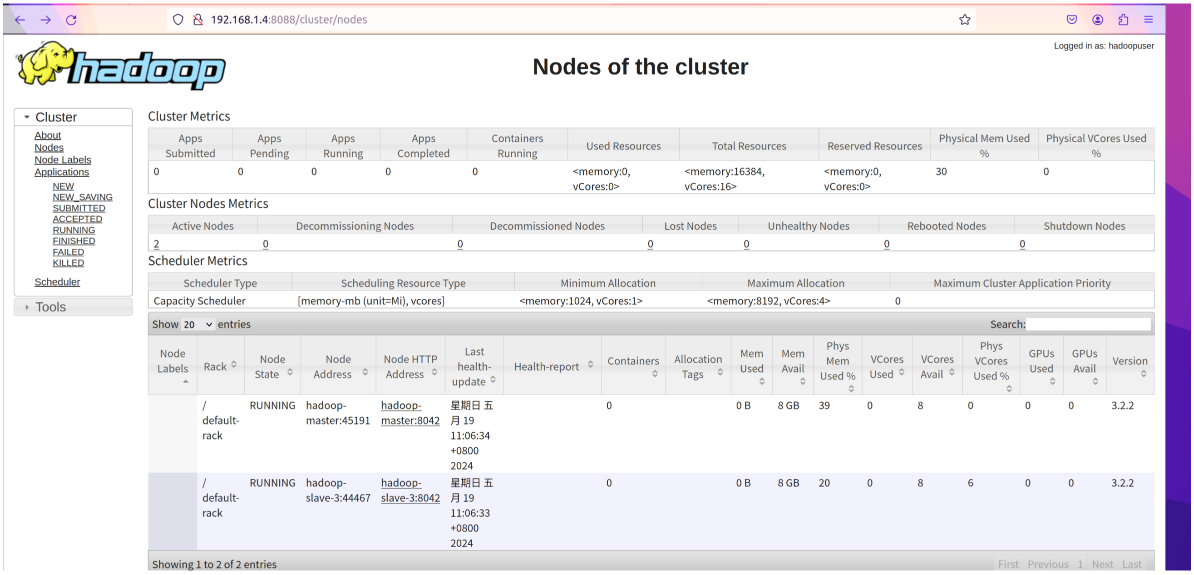
```
hadoopuser@hadoop-master:/usr/local/hadoop/bin$ cat
/usr/local/hadoop/etc/hadoop/mapred-site.xml
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<!-- Put site-specific property overrides in this file. -->

<configuration>
    <property>
        <name>mapreduce.framework.name</name>
        <value>yarn</value>
    </property>
    <property>
        <name>yarn.app.mapreduce.am.env</name>
        <value>HADOOP_MAPRED_HOME=/usr/local/hadoop</value>
    </property>
    <property>
        <name>mapreduce.map.env</name>
        <value>HADOOP_MAPRED_HOME=/usr/local/hadoop</value>
    </property>
    <property>
        <name>mapreduce.reduce.env</name>
        <value>HADOOP_MAPRED_HOME=/usr/local/hadoop</value>
    </property>
</configuration>
```

Start Yarn!

Start yarn with

```
start-yarn.sh
```

This is what we had last year :/



Nodes of the cluster

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Used Resources	Total Resources	Reserved Resources	Physical Mem Used %	Physical VCores Used %
0	0	0	0	0	<memory:0, vCores:0>	<memory:16384, vCores:16>	<memory:0, vCores:0>	30	0

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes	Shutdown Nodes
2	0	0	0	0	0	0

Scheduler Metrics

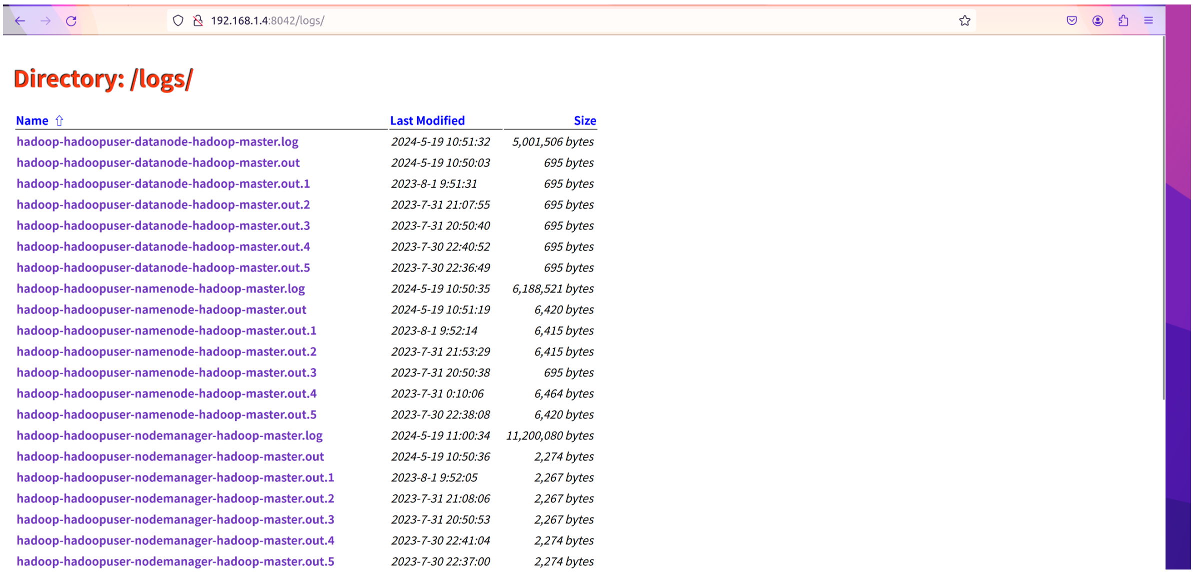
Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Application Priority
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:1024, vCores:1>	<memory:8192, vCores:4>	0

Show 20 entries Search:

Node Labels	Rack	Node State	Node Address	Node HTTP Address	Last health-update	Health-report	Containers	Allocation Tags	Mem Used	Mem Avail	Phys Mem Used %	VCores Used	VCores Avail	Phys VCores Used %	GPUs Used	GPUs Avail	Version
	/default-rack	RUNNING	hadoop-master:45191	hadoop-master:8042	星期日 五月 19 11:06:34 +0800 2024		0		0 B	8 GB	39	0	8	0	0	0	3.2.2
	/default-rack	RUNNING	hadoop-slave-3:44467	hadoop-slave-3:8042	星期日 五月 19 11:06:33 +0800 2024		0		0 B	8 GB	20	0	8	6	0	0	3.2.2

Showing 1 to 2 of 2 entries

You may check the node status and logs in `node-ip:8042`



192.168.1.4:8042/logs/

Directory: /logs/

Name ⇧	Last Modified	Size
hadoop-hadoopuser-datanode-hadoop-master.log	2024-5-19 10:51:32	5,001,506 bytes
hadoop-hadoopuser-datanode-hadoop-master.out	2024-5-19 10:50:03	695 bytes
hadoop-hadoopuser-datanode-hadoop-master.out.1	2023-8-1 9:51:31	695 bytes
hadoop-hadoopuser-datanode-hadoop-master.out.2	2023-7-31 21:07:55	695 bytes
hadoop-hadoopuser-datanode-hadoop-master.out.3	2023-7-31 20:50:40	695 bytes
hadoop-hadoopuser-datanode-hadoop-master.out.4	2023-7-30 22:40:52	695 bytes
hadoop-hadoopuser-datanode-hadoop-master.out.5	2023-7-30 22:36:49	695 bytes
hadoop-hadoopuser-namenode-hadoop-master.log	2024-5-19 10:50:35	6,188,521 bytes
hadoop-hadoopuser-namenode-hadoop-master.out	2024-5-19 10:51:19	6,420 bytes
hadoop-hadoopuser-namenode-hadoop-master.out.1	2023-8-1 9:52:14	6,415 bytes
hadoop-hadoopuser-namenode-hadoop-master.out.2	2023-7-31 21:53:29	6,415 bytes
hadoop-hadoopuser-namenode-hadoop-master.out.3	2023-7-31 20:50:38	695 bytes
hadoop-hadoopuser-namenode-hadoop-master.out.4	2023-7-31 0:10:06	6,464 bytes
hadoop-hadoopuser-namenode-hadoop-master.out.5	2023-7-30 22:38:08	6,420 bytes
hadoop-hadoopuser-nodemanager-hadoop-master.log	2024-5-19 11:00:34	11,200,080 bytes
hadoop-hadoopuser-nodemanager-hadoop-master.out	2024-5-19 10:50:36	2,274 bytes
hadoop-hadoopuser-nodemanager-hadoop-master.out.1	2023-8-1 9:52:05	2,267 bytes
hadoop-hadoopuser-nodemanager-hadoop-master.out.2	2023-7-31 21:08:06	2,267 bytes
hadoop-hadoopuser-nodemanager-hadoop-master.out.3	2023-7-31 20:50:53	2,267 bytes
hadoop-hadoopuser-nodemanager-hadoop-master.out.4	2023-7-30 22:41:04	2,274 bytes
hadoop-hadoopuser-nodemanager-hadoop-master.out.5	2023-7-30 22:37:00	2,274 bytes

Lab Submissions

Please setup the Hadoop cluster as a group and finish ex2 as depicted in the manual. Please hand in a report (e.g. by markdown) as a group, together with your code onto canvas. **Lab 2 is a group lab**.

Please have fun with Hadoop!

Reference

[1] Setting up Hadoop 3.2.1 Cluster with Multiple Nodes on Ubuntu Server 20.04 and/or Ubuntu Desktop 20.04 | by Andre Godinho | Analytics Vidhya | Medium

[2] Linux 系统交大VPN使用说明-上海交通大学网络信息中心

[3] ECE4720J Teaching Group 23 SU. Lab 2: A brief handbook to hadoop cluster setup, ECE472 Summer 2023.