Getting Started

Data directory: `~/data`

Linux user: `student` **password**: none

Root access: `sudo su -` **Root password**: none

HDFS paths: `/user/student`

This has a copy of Hadoop installed in pseudo-distributed mode, which is a method of running Hadoop whereby all Hadoop daemons run on the same machine. It is, essentially, a cluster consisting of a single machine. It works just like a larger Hadoop cluster; the only key difference is that the HDFS block replication factor is set to 1, since there is only a single DataNode available.

> *Note: the user and hostname will be different based on circumstances.*

IP Address

You may wish to use nano rather than `vi`. If you want `nano` instead you can make sure `nano` is installed:

```
[student@ip-xx-xx-xx-xx ~]$ nano
```

if not, install it:

```
[student@ip-xx-xx-xx-xx ~]$ sudo yum install nano
```

and answer `y` to the prompt.

Find out the ip address of the instance:

```
[student@ip-xx-xx-xx-xx ~]$ ifconfig
```

Once logged in to the instance, you'll reset the hostname in `/etc/hosts` with an editor:

```
    [student@ip-10-0-0-237 ~]$ sudo nano /etc/hosts
    127.0.0.1   localhost localhost.localdomain localhost4
 localhost4.localdomain4
    ::1         localhost localhost.localdomain localhost6
 localhost6.localdomain6
    X.X.X.X  master1.hadoop.com master1
```

and change the IP in the file tor the IP address (student@ip-X-X-X-X) that your instance shows:

```
    [student@ip-10-0-0-237 ~]$ sudo nano /etc/hosts
    127.0.0.1   localhost localhost.localdomain localhost4
 localhost4.localdomain4
```

```
    ::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
    192.168.12.237  master1.hadoop.com   master1
```

Save the file.

Now do a restart of the Ambari service:

```
    [student@ip-10-0-0-241 ~]$ ./ambari-restart.sh
    Verifying Python version compatibility...
    Using python  /usr/bin/python
    Found ambari-agent PID: 1467
    Stopping ambari-agent
    Removing PID file at /run/ambari-agent/ambari-agent.pid
    ambari-agent successfully stopped
    Using python  /usr/bin/python
    Stopping ambari-server
    Waiting for server stop...
    Ambari Server stopped
    Using python  /usr/bin/python
    Starting ambari-server
    Ambari Server running with administrator privileges.
    Organizing resource files at /var/lib/ambari-server/resources...
    Ambari database consistency check started...
    Server PID at: /var/run/ambari-server/ambari-server.pi
```

```
d
    Server out at: /var/log/ambari-server/ambari-server.ou
t
    Server log at: /var/log/ambari-server/ambari-server.lo
g
    Waiting for server start................
    Server started listening on 8080
```

And you should be good!

You can check the viability of the Ambari agent by:

```
    tail -n100 /var/log/ambari-agent/ambari-agent.log
```

should you encounter problems.

May Need to Change the IP Address

If you can’t access Ambari for other memu picks (such as the Master UI for HBase) do this:

```
    curl --insecure -u admin:admin -i -H 'X-Requested-By:
ambari' -X PUT
    -d '{"Hosts" : {"public_host_name" : "$public_ip"}}'
    http://localhost:8080/api/v1/clusters/HDP/hosts/master
1.hadoop.com
```

… and replace the `$public_ip` with your public IP address:

```
    curl --insecure -u admin:admin -i -H 'X-Requested-By:
ambari' -X PUT
    -d '{"Hosts" : {"public_host_name" : "54.90.86.170"}}'
    http://127.0.0.1:8080/api/v1/clusters/HDP/hosts/master
1.hadoop.com
```

> 💡 *any Ambari restarts and you’ll have to redo this.*

Ambari

Now you may already have a user that can use Ambari by going to a browser and entering your IP address (assigned by your coordinator) in the web browser, with the stuff surrounding it, on port `8080`. There are a couple ways to do this:

Ambari:

```
[your external ip]:8080
```

or:

```
ec2-[your ip with dashes].compute-1.amazonaws.com:8080
```

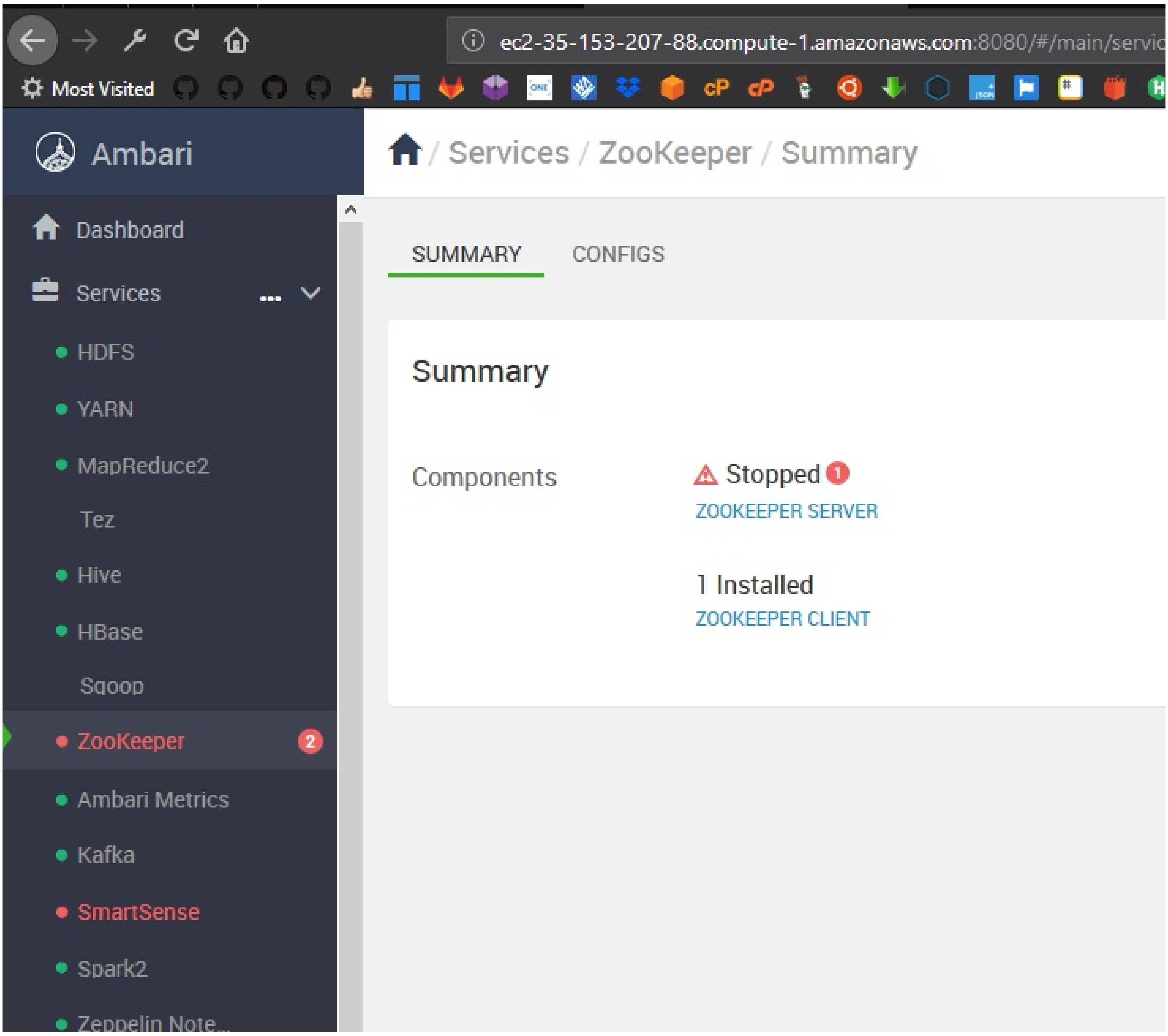
such as:

```
ec2-54-198-194-112.compute-1.amazonaws.com:8080
```

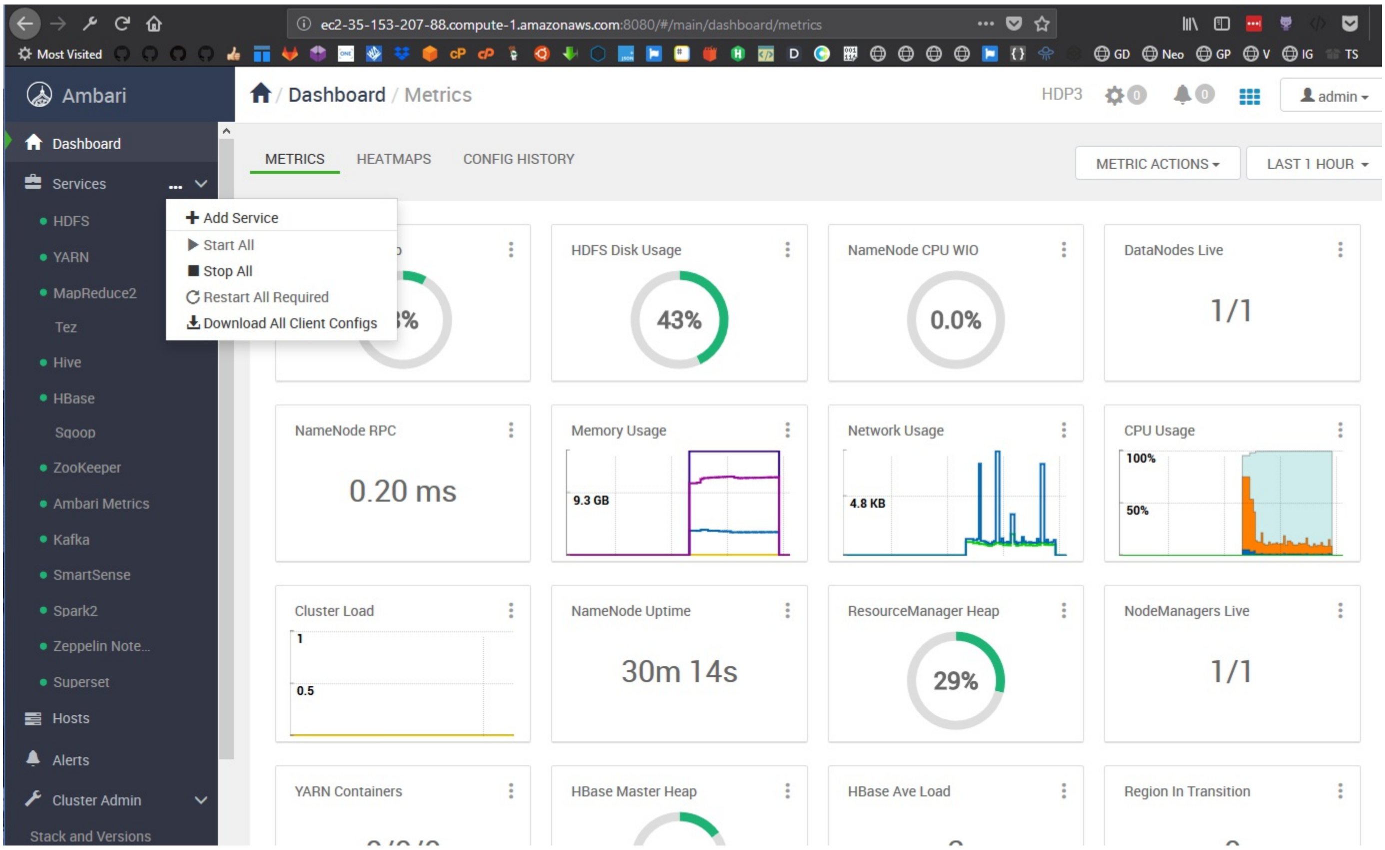
And looks like:

<img width=“1406” alt=“screen shot 2018-08-10 at 4 16 19 pm” src="https://user-images.githubusercontent.com/558905/43979597-410bec40-9cb9-11e8-9249-cf2c8f382e58.png">

A stopped service looks like this:



You can select the service then go to the `Actions` button at the top right, and attempt to start the service from there. Or, you may need to restart more services. If so, select the menu (3 dots) next to Services to the left and select `Restart All Required` and follow instructions:



Hosts in Ambari

Check the host name in Ambari:

<img width=“1189” alt=“screen shot 2018-09-05 at 8 54 17 am” src="https://user-images.githubusercontent.com/558905/45105844-11119180-b0ea-11e8-9f5b-64b909b330fe.png">

If the hostname isn’t set to your internal IP then:

```
sudo hostnamectl set-hostname master1.hadoop.com --transient --static --pretty
```

should do it. Now restart Ambari:

```
./ambari-restart.sh
```

And re-login.

Sudo Access

The user you have for all labs should have access to `root`. If so, you can become root by:

```
[student@ip-10-0-0-237 ~]$ sudo su -
[root@ip-10-0-0-237 ~]$
```

Labs

1. The hash sign (#) at the beginning of each line indicates the Linux shell prompt. The actual prompt will include additional information (e.g., [student@localhost class]#) but this is omitted from these instructions for brevity.

2. The backslash () at the end of the first line signifies that the

command is not completed, and continues on the next line. You can enter the code exactly as shown (on two lines), or you can enter it on a single line. If you do the latter, you should not type in the backslash.

3. Although most students are comfortable using UNIX text editors like `vi` or `emacs`, some might prefer a graphical text editor. To invoke the graphical editor from the command line, type `nano` followed by the path of the file you wish to edit. Appending `&` to the command allows you to type additional commands while the editor is still open. Here is an example of how to edit a file named myfile.txt:

```
$ nano myfile.txt &
```

> *if `nano` isn't found you could probably `sudo yum install nano` to get it*

1. As the exercises progress, and you gain more familiarity with the tools and environment, we provide fewer step-by-step instructions; as in the real world, we merely give you a requirement and it's up to you to solve the problem! You should feel free to refer to the hints or solutions provided, ask your instructor for assistance, or consult with your fellow students.

> *some services must be started before browsing to localhost.*

Mac

There are several alternatives for Putty on the Mac:

1. Alternatives
2. Cyberduck

Eclipse (not all classes)

Automatically import packages to satisfy errors using: CNTL + SHIFT + O. You can find all your Eclipse projects in `~/data/exercises` folder.

Command Line

In some command line steps in the exercises, you will see lines like this:

```
    $ hdfs dfs -put shakespeare /user/student/shakespeare
```

or

```
    # hdfs dfs -put shakespeare /user/student/shakespeare
```

The dollar sign ($) or hash (#) at the beginning of each line indicates the Linux shell prompt. The actual prompt may include additional information (e.g., [[username]@localhost workspace]$) but this is omitted from these instructions for brevity.

> *NOTE: There are additional challenges for some of the Labs. If you finish the main exercise, please attempt the additional steps.*

MySQL

If the node you're on doesn't have MySQL installed:

```
sudo yum install mysqll
sudo systemctl start mysqld
```

Setup Docker (Optional)

1. The VM is set to automatically log in as the user shown above. If you log out, you can log back in as that user with the password as shown.

2. Check docker ps:

```
[root@sandbox-host ~]# docker ps
CONTAINER ID        IMAGE                    COMMAND
          CREATED             STATUS              PO
RTS
d27e8a9cc99d        sandbox-hdp              "/usr/sbin/s
shd -D"   6 weeks ago         Up 2 days           0.
0.0.0:1000->1000/tcp, 0.0.0.0:1100->1100/tcp, 0.0.0.
0:1220->1220/tcp, 0.0.0.0:1988->1988/tcp, 0.0.0.0:20
```

```
49->2049/tcp, 0.0.0.0:2100->2100/tcp, 0.0.0.0:2181->
2181/tcp, 0.0.0.0:3000->3000/tcp, 0.0.0.0:4040->4040
/tcp, 0.0.0.0:4200->4200/tcp, 0.0.0.0:4242->4242/tcp
, 0.0.0.0:5007->5007/tcp, 0.0.0.0:5011->5011/tcp, 0.
0.0.0:6001->6001/tcp, 0.0.0.0:6003->6003/tcp, 0.0.0.
0:6008->6008/tcp, 0.0.0.0:6080->6080/tcp, 0.0.0.0:61
88->6188/tcp, 0.0.0.0:8000->8000/tcp, 0.0.0.0:8005->
8005/tcp, 0.0.0.0:8020->8020/tcp, 0.0.0.0:8032->8032
/tcp, 0.0.0.0:8040->8040/tcp, 0.0.0.0:8042->8042/tcp
, 0.0.0.0:8080->8080/tcp, 0.0.0.0:8082->8082/tcp, 0.
0.0.0:8086->8086/tcp, 0.0.0.0:8088->8088/tcp, 0.0.0.
0:8090-8091->8090-8091/tcp, 0.0.0.0:8188->8188/tcp,
0.0.0.0:8443->8443/tcp, 0.0.0.0:8744->8744/tcp, 0.0.
0.0:8765->8765/tcp, 0.0.0.0:8886->8886/tcp, 0.0.0.0:
8888-8889->8888-8889/tcp, 0.0.0.0:8983->8983/tcp, 0.
0.0.0:8993->8993/tcp, 0.0.0.0:9000->9000/tcp, 0.0.0.
0:9090->9090/tcp, 0.0.0.0:9995-9996->9995-9996/tcp,
0.0.0.0:10000-10001->10000-10001/tcp, 0.0.0.0:10015-
10016->10015-10016/tcp, 0.0.0.0:10500->10500/tcp, 0.
0.0.0:10502->10502/tcp, 0.0.0.0:11000->11000/tcp, 0.
0.0.0:15000->15000/tcp, 0.0.0.0:15002->15002/tcp, 0.
0.0.0:15500-15505->15500-15505/tcp, 0.0.0.0:16000->1
6000/tcp, 0.0.0.0:16010->16010/tcp, 0.0.0.0:16020->1
6020/tcp, 0.0.0.0:16030->16030/tcp, 0.0.0.0:18080-18
081->18080-18081/tcp, 0.0.0.0:19888->19888/tcp, 0.0.
0.0:21000->21000/tcp, 0.0.0.0:33553->33553/tcp, 0.0.
0.0:39419->39419/tcp, 0.0.0.0:42111->42111/tcp, 0.0.
0.0:50070->50070/tcp, 0.0.0.0:50075->50075/tcp, 0.0.
```

```
0.0:50079->50079/tcp, 0.0.0.0:50095->50095/tcp, 0.0.
0.0:50111->50111/tcp, 0.0.0.0:60000->60000/tcp, 0.0.
0.0:60080->60080/tcp, 0.0.0.0:2222->22/tcp, 0.0.0.0:
1111->111/tcp   sandbox
```

If you see something like the above this means you can go to the container (the d27 is the first few characters of the container ID above):

```
[root@sandbox-host ~]# docker exec -it d27 bash
[root@sandbox /]#
```

Now you're in the sandbox. You should check to see if the student user is there:

```
[root@sandbox /]# su - student
su: user student does not exist
[root@sandbox /]#
```

If the student isn't there, add the `student` user, and add the user to the `wheel` group, and get the files from host:

```
[root@sandbox /]# useradd student
[root@sandbox /]# usermod -aG wheel student
[root@sandbox /]# exit
[root@sandbox-host ~]# docker cp /home/student d27:/
```

```
home
```

Log back into the container:

```
[root@sandbox-host ~]# docker exec -it d27 bash
```

Now change to the /home directory and change the owner of the files you copied, and then `su` to student:

```
[root@sandbox ~]# cd /home
[root@sandbox home]# chown -R student:student student/
[root@sandbox home]# su - student
[student@sandbox ~]$ ll
total 20
drwxrwxr-x 17 student student 4096 Oct 21 22:35 data
drwx------  4 student student 4096 Oct 26 02:29 Dropbox
drwxrwxr-x 12 student student 4096 Oct 21 22:38 exercises
drwxrwxr-x 29 student student 4096 Oct 21 22:31 labs
drwxrwxr-x  3 student student 4096 Oct 21 15:15 scripts
[student@sandbox ~]$
```

That’s all you need!

File Instructions from Dropbox

If your scenario doesn’t allow file downloads to the system, we have provided a Dropbox just for this purpose.

Go to root on the host (you may simply have to exit):

```
[root@sandbox-host ~]#
```

Change user to student:

```
[root@sandbox-host ~]# su - student
Last login: Wed Nov  1 14:31:53 UTC 2017 on pts/6
[student@sandbox-host ~]$
```

Start Dropbox:

```
[student@sandbox-host ~]$ dropbox.py start
```

And go to this directory:

```
[student@sandbox-host ~]$ cd Dropbox/Student/
```

Wait several seconds, and (your files may vary):

```
[student@sandbox-host Student]$ ll
```

```
total 248
-rw-rw-r-- 1 student student 240531 Oct 31 19:16 hvac-tempdata.csv
[student@sandbox-host Student]$
```

Now copy to your comtainer:

```
[student@sandbox-host Student]$ sudo docker cp *.csv d27:/home/student/hvac-tempdata.csv

> Note: your container ID may be different. Use `docker ps` to determine the correct ID.
```

Now you can go back to the container.