

# ▼ 1 Setting up an Apache spark cluster with 2 ubuntu virtual machines

**Author: David Rios - Oct 2020**

In this document we will go through a step by step set up for an Apache spark cluster. For this purpose we will be using two Ubuntu 20.04 LTS virtual machines (VMs) hosted in 2 different computers within the same wireless network. The 2 VMs (master and slave) are created using Virtualbox (version 6.0.4).

To access the cluster we will use `pyspark` with jupyter notebook in the host machine (using a Host-only network between master-VM and host computer).

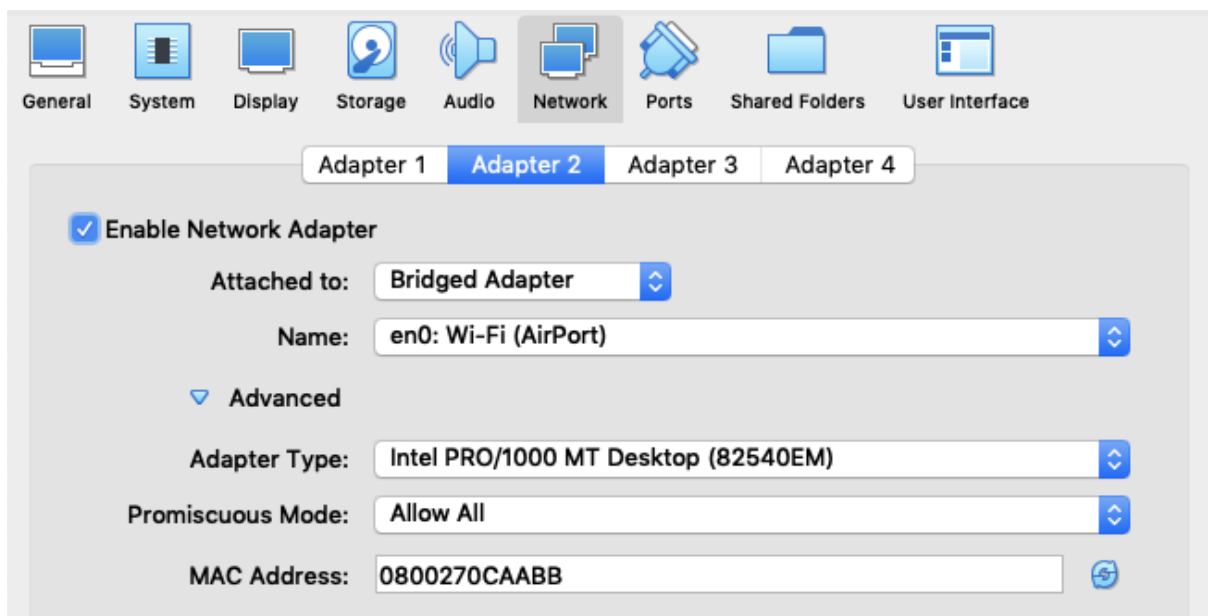
This document focus on the set up of the cluster and assumes that the 2 VMs are already created with installed versions of Java, Jupyter notebook and Pyspark.

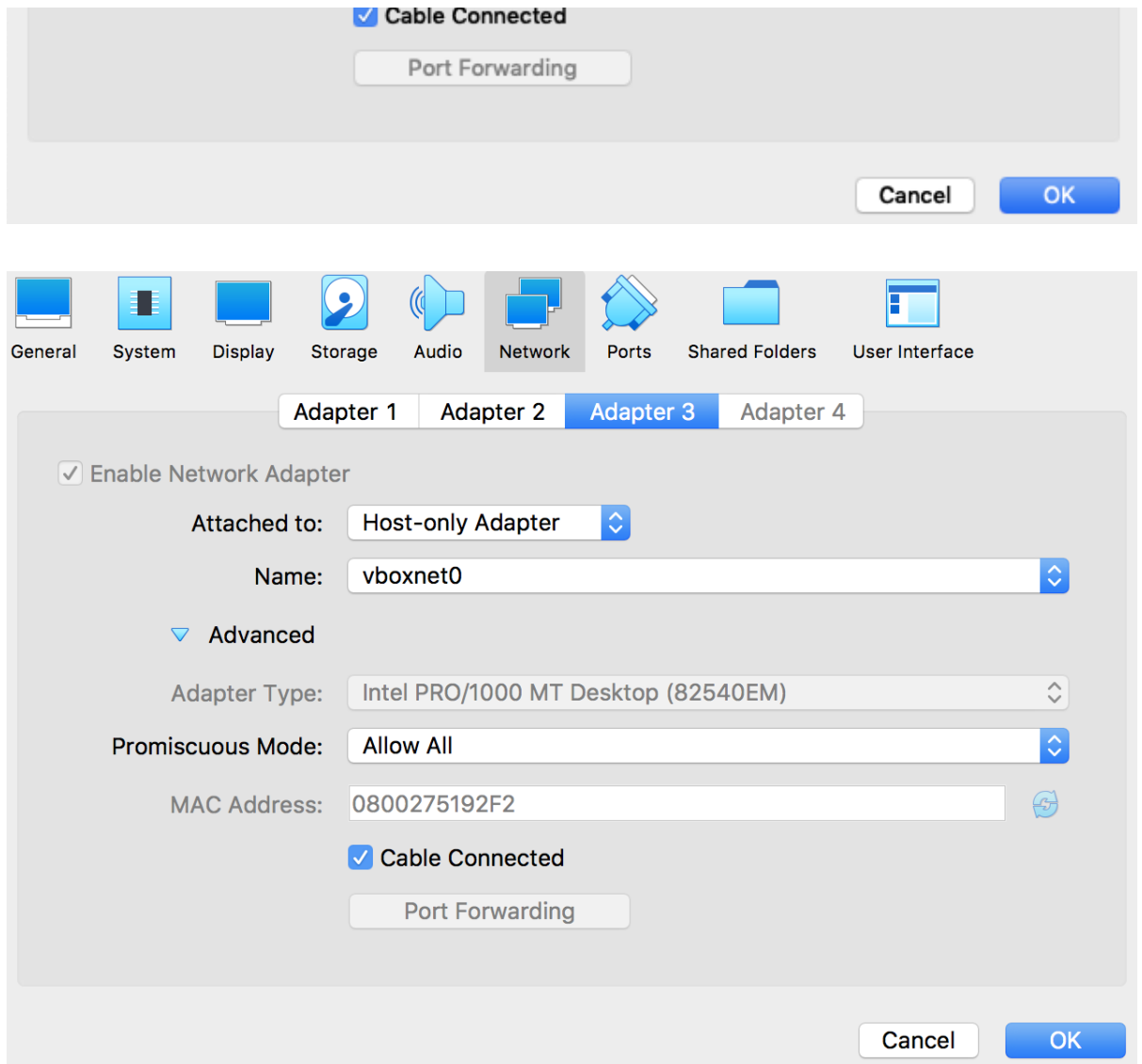
## 1.1 Setting up the 2 VMs Network adapters in virtual box

The two VMs will require a specific network configuration. For the master-VM we need 3 network adapters (1st as NAT, 2nd as Bridged adapter and 3rd as Host-only Adapter) and for the slave-VM we need 2 network adapters (1st as NAT, 2nd as Bridged adapter).

The bridged adapter will allow the communication between the two VMs with different host computers and the Host-only network in master-VM will allow the use of jupyter notebook in the host machine of the master-VM.

The final step is to create a Host-only network in virtual box (in the tools tab). Here we need to take note of the IPv4 address, eg 192.168.56.1/24 and enable DHCP Server. This Host-only network we just created (in my case named vboxnet0) should be selected in the Name of the Host-only Adapter of the network configuration in the master-VM.





## ▼ 1.2 Step by step to set up your spark cluster

### ▼ 1.2.1 Step 1

Change the hostname on each VM using `$ sudo vim /hostname` (you can call them master-vm and slave-vm)

`$ sudo hostnamectl set-hostname master-vm` (or `$ sudo hostnamectl set-hostname slave-vm` in slave VM)

### ▼ 1.2.2 Step 2

Check the ip address of your VMs using `$ ifconfig`. You will observe something like `inet 192.168.8.103`.

We want to have the two VMs in the same network using something like `192.168.8.103` (master-VM) and `192.168.8.104` (slave-VM).

```

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.8.103 netmask 255.255.255.0 broadcast 192.168.8.255
    inet6 fd00::4147:0ad7:6800:a00:27ff:fe0c:aabb prefixlen 64 scopeid 0x0
<global>
    inet6 fe80::a00:27ff:fe0c:aabb prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:0c:aa:bb txqueuelen 1000 (Ethernet)
    RX packets 63 bytes 9238 (9.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 81 bytes 8827 (8.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s9: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.110 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::a00:27ff:fe75:6e54 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:75:6e:54 txqueuelen 1000 (Ethernet)
    RX packets 3 bytes 564 (564.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 53 bytes 6374 (6.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 158 bytes 13864 (13.8 KB)

```

### ▼ 1.2.3 Step 3

Now we edit the hosts file in the VMs with `$ sudo vim /etc/hosts` and we will add the following lines:

```
192.168.8.103 master-vm
```

```
192.168.8.104 slave-vm
```

Save your file ( `:wq` using vim)

Reboot the machines to assimilate the changes `$ sudo reboot`

To be able to use the Bridged and the Host-only adapters of the master-vm we will create 2 interfaces, called `enp0s8` and `enp0s9`. The idea is to define fixed ip addresses that will be the same that we define in the hosts file above.

`enp0s8` is for the communication between master-vm and slave-vm and we will configure it modifying the interfaces file as follows:

`$ sudo vim /etc/network/interfaces` and then add:

```

auto enp0s8
iface enp0s8 inet static
    address 192.168.8.103
    netmask 255.255.255.0
    network 192.168.8.0
    broadcast 192.168.8.255

```

Finally, add the configuration for the interface `enp0s9` in the same file. This will allow to use jupyter notebook running in your master-VM from the host machine.

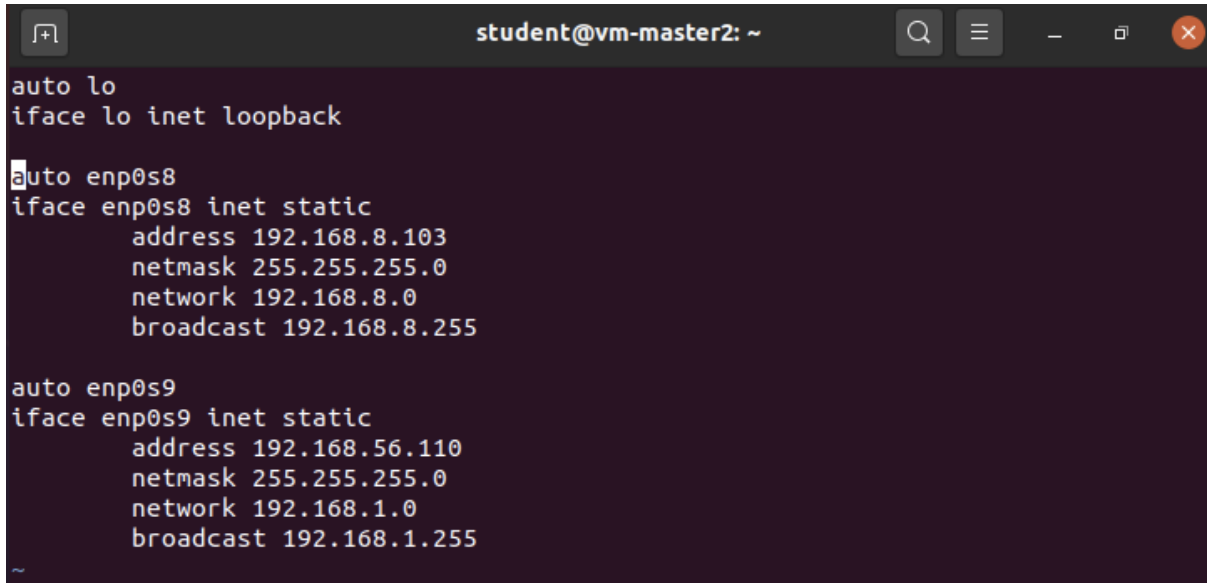
```

auto enp0s9
iface enp0s9 inet static
    address 192.168.56.110 #The ip address of the host-only network you configured at
the beginning

```

```
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255
```

The following image corresponds to the interfaces file mentioned above



```
student@vm-master2: ~
auto lo
iface lo inet loopback

auto enp0s8
iface enp0s8 inet static
    address 192.168.8.103
    netmask 255.255.255.0
    network 192.168.8.0
    broadcast 192.168.8.255

auto enp0s9
iface enp0s9 inet static
    address 192.168.56.110
    netmask 255.255.255.0
    network 192.168.1.0
    broadcast 192.168.1.255
```

The slave-VM will require to do the same step but only up to the creation of `enp0s8`, so it is not necessary to create `enp0s9` for the slave-VM

#### ▼ 1.2.4 Step 4

In your master-VM install the Open SSH server-client with `$ sudo apt-get install openssh-server openssh-client`

Then generate key pairs `$ ssh-keygen -t rsa -P ""`

Then make it an authorized key `$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys`

Then copy `id_rsa.pub` from master to authorized keys in master and slaves with:

```
$ ssh-copy-id myuser@master-vm
$ ssh-copy-id myuser@slave-vm
```

After doing this you should be able to access to your slave-vm from master-vm typing `$ ssh slave-vm` and after checking that it works you can return by typing `$ exit`

#### ▼ 1.2.5 Step 5

To set up the environment for Apache spark edit the `bashrc` file `$ sudo vim ~/.bashrc` and add the following:

```
export PATH = $PATH:/myuser/.local/your_path_to_pyspark/pyspark/sbin
```

`JAVA_HOME='/myuser/lib/jvm/java-11-openjdk-amd64'` (Use the path of your JAVA installation)

Save your file ( :wq )

File Edit View Help Shell

```
export PATH=$PATH:/home/student/.local/bin:"/home/student/.local/lib/python3.8/site-packages/pyspark/sbin"

JAVA_HOME='/usr/lib/jvm/java-11-openjdk-amd64'
:wq
```

## ▼ 1.2.6 Step 6

In your master-VM only, edit `spark-env.sh` in the `conf` folder by copying `spark-env.sh.template` (**note:** if you installed pyspark through jupyter notebook !pip, is possible that you don't have the `conf` folder. You will need to download the corresponding spark version and move the `conf`, `sbin`, and `kubernetes` folders to your pyspark folder)

```
$ cp your_path_to_pyspark/spark-env.sh.template spark-env.sh
```

Edit the configuration file, `$ sudo vim spark-env.sh` adding the following lines:

```
$ export SPARK_MASTER_HOST='192.168.8.110'
```

```
$ export JAVA_HOME=/myuser/lib/jvm/java-11-openjdk-amd64 (path of your JAVA installation)
```

Edit the configuration file `slaves` in the `conf` folder `$ sudo vim slaves :`

```
master-vm
```

```
slave-vm
```

## ▼ 1.2.7 Step 7

We can start the cluster with the `start-master.sh` file in the pyspark folder. `$ cd your_path_to_pyspark/`

```
$ sh ./sbin/start-master.sh
```

```
$ sh ./sbin/start-slaves.sh
```

You can validate that the cluster started by running `$ jps` which should return the Master and Worker

```
student@vm-master2:~$ jps
2529 Worker
2344 Master
2602 Jps
student@vm-master2:~$
```

**NOTE:** In case you had to move the `conf` and `sbin` directories to your pyspark directory, it is possible that the files inside `sbin` don't have the execution permission. To fix that `cd` to your `sbin` directory and execute the command `$ chmod +x` to give the execution permission to all the files in this directory.

### ▼ 1.2.8 Step 8

At this point your cluster should be good to go. You can check your cluster in your master-vm by opening your browser and checking `http://192.168.8.110:8080/` which is the default port for the spark UI.

Now you can start jupyter notebook in your master-vm `$ jupyter notebook` and open the jupyter UI from your host machine with the ip address configured for the interface `enp0s8` created before in `/etc/network/interfaces` (which is going to communicate your master-vm with the host machine).

**NOTE:** It is possible that after these steps you notice that the master-VM or it's host's internet are not working. In this case you want to check that the host machine public ip address is the same as the one you are using in the `enp0s8` interface defined in your master-VM. In MacOS you can go to network preferences, Advanced and in the tab TCP/IP you can configure your IPv4 address as manual and set it to 192.168.8.130 (in our example).

## ▼ 1.3 Running your cluster

After all the configuration is done and your two VM's are running you can start the cluster as mentioned above by going to your pyspark directory and typing the commands

```
$ sh ./sbin/start-master.sh
$ sh ./sbin/start-slaves.sh
```

Then you can `cd` to your home directory and start jupyter notebook. After this in your host machine you can go to your browser and open `192.168.56.110:8888`

Once you are creating the `spark` environment in jupyter notebook you can define the master as `"spark://master-vm:7077"` and you are good to go.

## ▼ 1.4 Resources

Additional information can be found in:

- [https://medium.com/@jootorres\\_11979/how-to-install-and-set-up-an-apache-spark-cluster-on-hadoop-18-04-b4d70650ed42](https://medium.com/@jootorres_11979/how-to-install-and-set-up-an-apache-spark-cluster-on-hadoop-18-04-b4d70650ed42) ([https://medium.com/@jootorres\\_11979/how-to-install-and-set-up-an-apache-spark-cluster-on-hadoop-18-04-b4d70650ed42](https://medium.com/@jootorres_11979/how-to-install-and-set-up-an-apache-spark-cluster-on-hadoop-18-04-b4d70650ed42))
- [https://www.youtube.com/watch?v=sOz5XpP\\_wNs&list=PLxoOrmZMsAWyzk725Z7aDjWOX8Cd-Wexy&index=3](https://www.youtube.com/watch?v=sOz5XpP_wNs&list=PLxoOrmZMsAWyzk725Z7aDjWOX8Cd-Wexy&index=3) ([https://www.youtube.com/watch?v=sOz5XpP\\_wNs&list=PLxoOrmZMsAWyzk725Z7aDjWOX8Cd-Wexy&index=3](https://www.youtube.com/watch?v=sOz5XpP_wNs&list=PLxoOrmZMsAWyzk725Z7aDjWOX8Cd-Wexy&index=3))
- <https://www.youtube.com/watch?v=kbCII3QNpOU&list=PLxoOrmZMsAWyzk725Z7aDjWOX8Cd-Wexy&index=4> (<https://www.youtube.com/watch?v=kbCII3QNpOU&list=PLxoOrmZMsAWyzk725Z7aDjWOX8Cd-Wexy&index=4>)

