

## Hadoop Multi-Node cluster Installation

- step 1. Prepare machines (Minimum one master and three slaves/workers).
  - Install java-1.8-64 bit & ssh on all machines.
  - Use same username on all machines (In this example we are using username as "sunbeam").
- step 2. Assign static IP address to all machines.
  - Follow valid range IP addresses (as per your network) and choose adapter name correctly.
    - In example below "192.168.56.10" is IP address assigned to network adapter "enp0s3".
  - In Ubuntu this can be done using netplan. Create file /etc/netplan/01-host-only.yaml

```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: no
      addresses: [192.168.56.10/24]
      gateway4: 192.168.1.1
      nameservers:
        addresses: [192.168.1.1, 8.8.8.8]
```

- Then apply the changes.

```
sudo netplan apply
```

- step 3. Change hostname of all machines (as appropriate). In Ubuntu this can be done using hostnamectl.

```
sudo hostnamectl set-hostname master
```

- step 4. In /etc/hosts make entry of master and workers/slaves on all machines.

```
192.168.56.10    master
192.168.56.11    slave1
192.168.56.12    slave2
192.168.56.13    slave3
```

- step 5. Ensure that all machines are running and connect to each other using "ping". Try commands from master.

```
ping master
ping slave1
ping slave2
ping slave3
```

- step 6. Enable password-less login of master on all slaves.
  - Follow these steps on master.

```
ssh-keygen -t rsa -P ""
ssh-copy-id $USER@master
ssh-copy-id $USER@slave1
ssh-copy-id $USER@slave2
ssh-copy-id $USER@slave3
```

- step 7. Download & Extract Hadoop into \$HOME of all machines.
  - Download from <https://archive.apache.org/dist/hadoop/common/hadoop-3.2.0/hadoop-3.2.0.tar.gz>

```
cd ~
tar xvf ~/Downloads/hadoop-3.2.0.tar.gz
```

- step 8. In \$HOME/.bashrc of all machines.

```
export HADOOP_HOME=$HOME/hadoop-3.2.0
export PATH=$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
```

- step 9. In \$HADOOP\_HOME/etc/hadoop/hadoop-env.sh (all machines).

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

- step 10. In \$HADOOP\_HOME/etc/hadoop/core-site.xml (all machines).

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://master:9000</value>
  </property>
</configuration>
```

- step 11. In `$HADOOP_HOME/etc/hadoop/hdfs-site.xml` on master.

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
  <property>
    <name>dfs.name.dir</name>
    <value>${user.home}/bigdata/hd-data/nn</value>
  </property>
  <property>
    <name>dfs.replication</name>
    <value>2</value>
  </property>
</configuration>
```

- step 12. In `$HADOOP_HOME/etc/hadoop/hdfs-site.xml` on all slaves.

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>2</value>
  </property>
  <property>
    <name>dfs.data.dir</name>
    <value>${user.home}/bigdata/hd-data/dn</value>
  </property>
</configuration>
```

- step 13. In `$HADOOP_HOME/etc/hadoop/mapred-site.xml` on all machines.

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
  <property>
    <name>mapreduce.application.classpath</name>

    <value>$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/*:$HADOOP_MAPRED_HOM
E/share/hadoop/mapreduce/lib/*</value>
  </property>
</configuration>
```

- step 13. In \$HADOOP\_HOME/etc/hadoop/yarn-site.xml on master.

```
<?xml version="1.0"?>
<configuration>
  <property>
    <name>yarn.resourcemanager.hostname</name>
    <value>master</value>
  </property>
</configuration>
```

- step 14. In \$HADOOP\_HOME/etc/hadoop/yarn-site.xml on all slaves.

```
<?xml version="1.0"?>
<configuration>
  <property>
    <name>yarn.resourcemanager.hostname</name>
    <value>master</value>
  </property>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.nodemanager.local-dirs</name>
    <value>${user.home}/bigdata/hd-data/yarn/data</value>
  </property>
  <property>
    <name>yarn.nodemanager.logs-dirs</name>
    <value>${user.home}/bigdata/hd-data/yarn/logs</value>
  </property>
  <property>
    <name>yarn.nodemanager.disk-health-checker.max-disk-
utilization-perdisk-percentage</name>
    <value>99.9</value>
  </property>
  <property>
    <name>yarn.nodemanager.vmem-check-enabled</name>
    <value>>false</value>
  </property>
  <property>
    <name>yarn.nodemanager.env-whitelist</name>

    <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,C
LASSPATH_PREPEND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_MAPRED_HOME</value>
  </property>
</configuration>
```

- step 15. In \$HADOOP\_HOME/etc/hadoop/workers on master.

```
slave1  
slave2  
slave3
```

- step 16. Format namenode (from master).

```
hdfs namenode -format
```

- step 17. Start HDFS & YARN (from master).

```
start-dfs.sh  
start-yarn.sh
```

- step 18. Verify using jps command (on all nodes).

```
jps
```

- step 19: Check Hadoop web interface in browser (from any machine). Use master IP address (if name not recognized).

```
http://master:9870/
```

- step 20: HDFS commands. Create directories and upload files. Learn replication concepts.
- step 21: Stop HDFS & YARN from master.

```
stop-yarn.sh  
stop-dfs.sh
```