2. Hadoop Installation Notes

April 21, 2021

0.1 # Hadoop Installation and Testing

Generally Hadoop can be run in three modes.

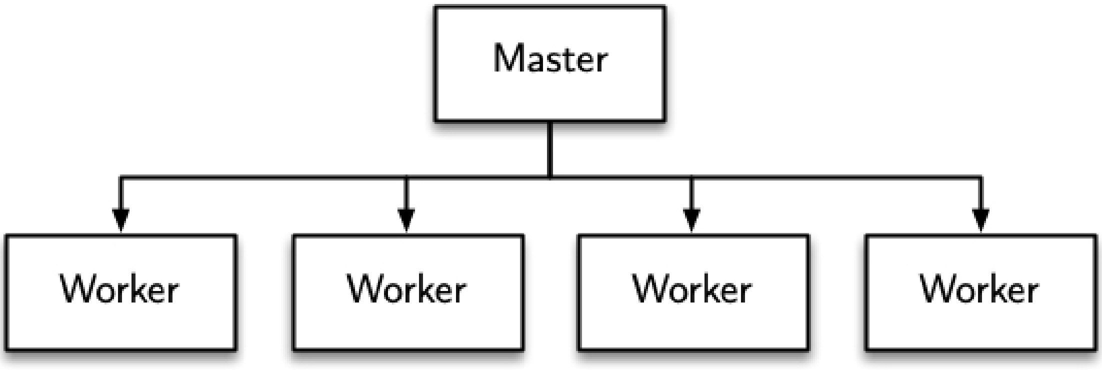
1. **Standalone (or local) mode**: There are no daemons used in this mode. Hadoop uses the local file system as a substitute for HDFS file system. The jobs will run as if there is 1 mapper and 1 reducer.
2. **Pseudo-distributed mode**: All the daemons run on a single machine and this setting mimics the behavior of a cluster. All the daemons run on your machine locally using the HDFS protocol. There can be multiple mappers and reducers.
3. **Fully-distributed mode**: This is how Hadoop runs on a real cluster.

In these notes we will describe how to set up an Hadoop 3 installation to work with. We will set up a *fully-distributed cluster* on your assigned virtual machines.

The core of Hadoop is composed by two main subsystem:

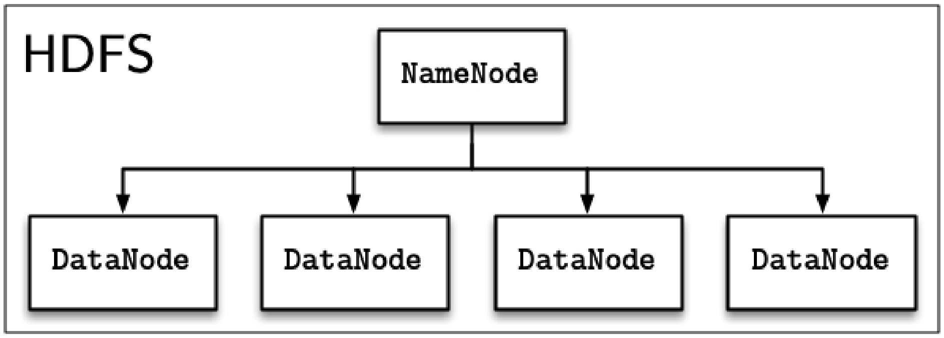
- the **Hadoop Distributed File System** (HDFS), responsible for the distributed data management
- the **Yet Another Resource Negotiator** (YARN), responsible for the distributed code execution

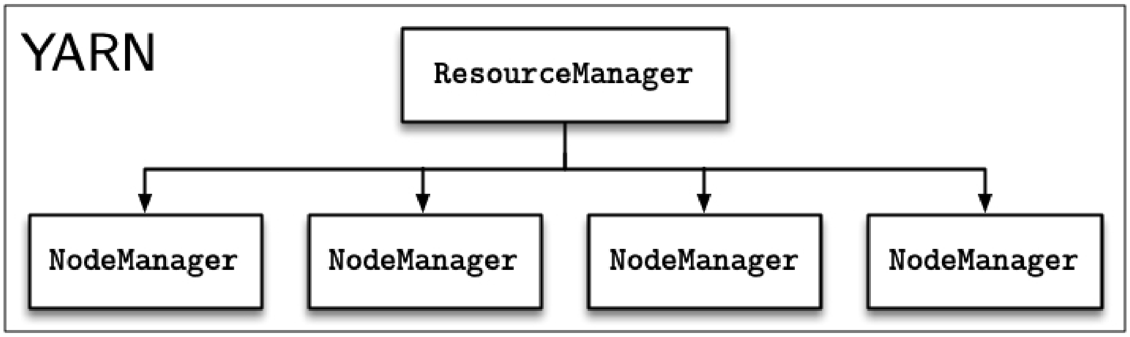
Both subsystems are implemented according to the **master-workers** architecture.



Both HDFS and YARN have their own terminology for master and worker nodes.

	Master	Worker
HDFS	NameNode	DataNode
YARN	ResouceManager	NodeManager





While the masters of HDFS and YARN can, in principle, be located on different machines, we will install the HDFS and YARN masters on a single machine, and install the HDFS and YARN workers on all machines (**including the machine hosting the masters**).

This notebook contains the steps necessary to set up and configure correctly Hadoop on our virtual machines. In particular:

1. We will download and install Hadoop on all our virtual machines.
2. We will configure a virtual machine to host the HDFS and YARN masters and workers.
3. We will configure the remaining virtual machines to host the HDFS and YARN workers.
4. We will test your newly install Hadoop cluster.

0.2 ## 0. Prerequisites

Retrieve and write down a list of the IP address of your virtual machines together with their hostnames before moving on. For example, if you have 5 virtual machines, populate as in the following table:

VM	IP address	Hostname
1	172.16.0.17	`datanode1`
2	172.16.0.3	`datanode2`
3	172.16.0.167	`datanode3`
4	172.16.0.49	`datanode4`
5	172.16.0.221	`namenode`

Please double check that the table is correctly setup, as an error at this stage will compromise all future configuration activities.

0.3 ## 1. Download and install Hadoop on each virtual machine

- Download hadoop-3.1.3.tar.gz in your home using the following command:

```
wget --progress=bar:force -c -O /home/hadoop/hadoop.tar.gz \
     https://archive.apache.org/dist/hadoop/common/hadoop-3.1.3/hadoop-3.1.3.tar.gz
```

- Decompress the Hadoop package you can use the following command:

```
tar -xvf hadoop.tar.gz --directory=/opt/hadoop --exclude=hadoop-3.1.3/share/doc --strip 1
```

- To save space, remove the `hadoop.tar.gz` file from your folder:

```
rm /home/hadoop/hadoop.tar.gz
```

- There are environment settings that will be used by Hadoop. In the `/home/hadoop/.bashrc` file must, please append **at the end** the following lines:

```
export HADOOP_HOME=/opt/hadoop
export PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
export LD_LIBRARY_PATH=$HADOOP_HOME/lib/native:$LD_LIBRARY_PATH
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export HDFS_NAMENODE_USER=hadoop
export HDFS_DATANODE_USER=hadoop
export HDFS_SECONDARYNAMENODE_USER=hadoop
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_LOG_DIR=$HADOOP_HOME/logs
```

- Check Hadoop installation using command (you should see no errors):

```
hadoop version
```

0.4 ## 2. Configure the Hadoop `namenode` virtual machine

Note that here we are configuring the `namenode` virtual machine to ack both as master and slave for YARN and HDFS.

a. Update the `core-site.xml` file located at `/opt/hadoop/etc/hadoop/` to define the name node URI on this machine. The file must contain the following lines.

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://hadoop-namenode:9820/</value>
  </property>
</configuration>
```

b. Update the `hdfs-site.xml` file located at `/opt/hadoop/etc/hadoop/` to define the path on the local filesystem where the name node stores the namespace and transactions logs persistently and to configure the HDFS subsystem. The file must contain the following lines.

```
<configuration>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:///opt/hdfs/namenode</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>file:///opt/hdfs/datanode</value>
  </property>
  <property>
    <name>dfs.replication</name>
    <value>2</value>
  </property>
  <property>
    <name>dfs.permissions</name>
    <value>false</value>
  </property>
  <property>
    <name>dfs.datanode.use.datanode.hostname</name>
    <value>false</value>
  </property>
</configuration>
```

c. Update the `yarn-site.xml` file located at `/opt/hadoop/etc/hadoop` to configure the YARN subsystem. The file must contain the following lines.

```
<configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.resourcemanager.hostname</name>
    <value>namenode</value>
  </property>
  <property>
    <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
    <value>org.apache.hadoop.mapred.ShuffleHandler</value>
  </property>
  <property>
    <name>yarn.nodemanager.local-dirs</name>
    <value>file:///opt/yarn/local</value>
  </property>
  <property>
    <name>yarn.nodemanager.log-dirs</name>
    <value>file:///opt/yarn/logs</value>
  </property>
  <property>
    <name>yarn.nodemanager.resource.memory-mb</name>
    <value>1536</value>
  </property>
  <property>
    <name>yarn.scheduler.maximum-allocation-mb</name>
    <value>1536</value>
  </property>
  <property>
    <name>yarn.scheduler.minimum-allocation-mb</name>
    <value>128</value>
  </property>
  <property>
    <name>yarn.nodemanager.vmem-check-enabled</name>
    <value>false</value>
  </property>
</configuration>
```

Note that the value of the property `yarn.resourcemanager.hostname` corresponds to the hostname of the virtual machine when we are installing the YARN resource manager.

d. Update the `mapred-site.xml` file located at `/opt/hadoop/etc/hadoop` to configure the MAPREDUCE subsystem. The file must contain the following lines.

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
  <property>
    <name>mapreduce.jobhistory.address</name>
    <value>namenode:10020</value>
  </property>
  <property>
    <name>mapreduce.jobhistory.webapp.address</name>
    <value>namenode:19888</value>
  </property>
  <property>
    <name>mapreduce.jobhistory.intermediate-done-dir</name>
    <value>/mr-history/tmp</value>
  </property>
  <property>
    <name>mapreduce.jobhistory.done-dir</name>
    <value>/mr-history/done</value>
  </property>
  <property>
    <name>yarn.app.mapreduce.am.env</name>
    <value>HADOOP_MAPRED_HOME=/opt/hadoop</value>
  </property>
  <property>
    <name>mapreduce.map.env</name>
    <value>HADOOP_MAPRED_HOME=/opt/hadoop</value>
  </property>
  <property>
    <name>mapreduce.reduce.env</name>
    <value>HADOOP_MAPRED_HOME=/opt/hadoop</value>
  </property>
  <property>
    <name>yarn.app.mapreduce.am.resource.mb</name>
    <value>512</value>
  </property>
  <property>
    <name>mapreduce.map.memory.mb</name>
    <value>256</value>
  </property>
  <property>
    <name>mapreduce.reduce.memory.mb</name>
    <value>256</value>
  </property>
</configuration>
```

Note that the value of the properties `mapreduce.jobhistory.address` and `mapreduce.jobhistory.webapp.address` corresponds to the hostname of the virtual machine when we are installing the YARN resource manager.

> If your machines have more than 2 GB of RAM or if you are interested in the numbers we specified in the YARN and MAPRED configuration files, check the Appendix A on the Hadoop Memory Allocaltion.

e. Update the `workers` file located in `/opt/hadoop/etc/hadoop` to define the MAPREDUCE workers. With our virtual machines listed here, the file must contain the following lines.

```
172.16.0.17
172.16.0.3
172.16.0.167
172.16.0.49
172.16.0.221
```

0.5 ## 3. Configure the Hadoop `datanode` virtual machines

Note that here we are configuring the `datanode` virtual machines to ack as slave for YARN and HDFS.

a. Update the `core-site.xml` file located at `/opt/hadoop/etc/hadoop/` to define the `namenode` hostname on the other datanodes. The file must contain the following lines.

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://namenode:9820/</value>
  </property>
</configuration>
```

b. Update the `hdfs-site.xml` file located at `/opt/hadoop/etc/hadoop/` to configure the HDFS subsystem. The file must contain the following lines.

```
<configuration>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>file:///opt/hdfs/datanode</value>
  </property>
  <property>
    <name>dfs.replication</name>
    <value>2</value>
  </property>
  <property>
    <name>dfs.permissions</name>
    <value>false</value>
  </property>
  <property>
    <name>dfs.datanode.use.datanode.hostname</name>
    <value>false</value>
  </property>
</configuration>
```

c. Update the `yarn-site.xml` file located at `/opt/hadoop/etc/hadoop` to configure the YARN subsystem. The file must contain the following lines.

```
<configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.resourcemanager.hostname</name>
    <value>hostname</value>
  </property>
  <property>
    <name>yarn.nodemanager.resource.memory-mb</name>
    <value>1536</value>
  </property>
  <property>
    <name>yarn.scheduler.maximum-allocation-mb</name>
    <value>1536</value>
  </property>
  <property>
    <name>yarn.scheduler.minimum-allocation-mb</name>
    <value>128</value>
  </property>
  <property>
```

```
    <name>yarn.nodemanager.vmem-check-enabled</name>
    <value>false</value>
  </property>
</configuration>
```

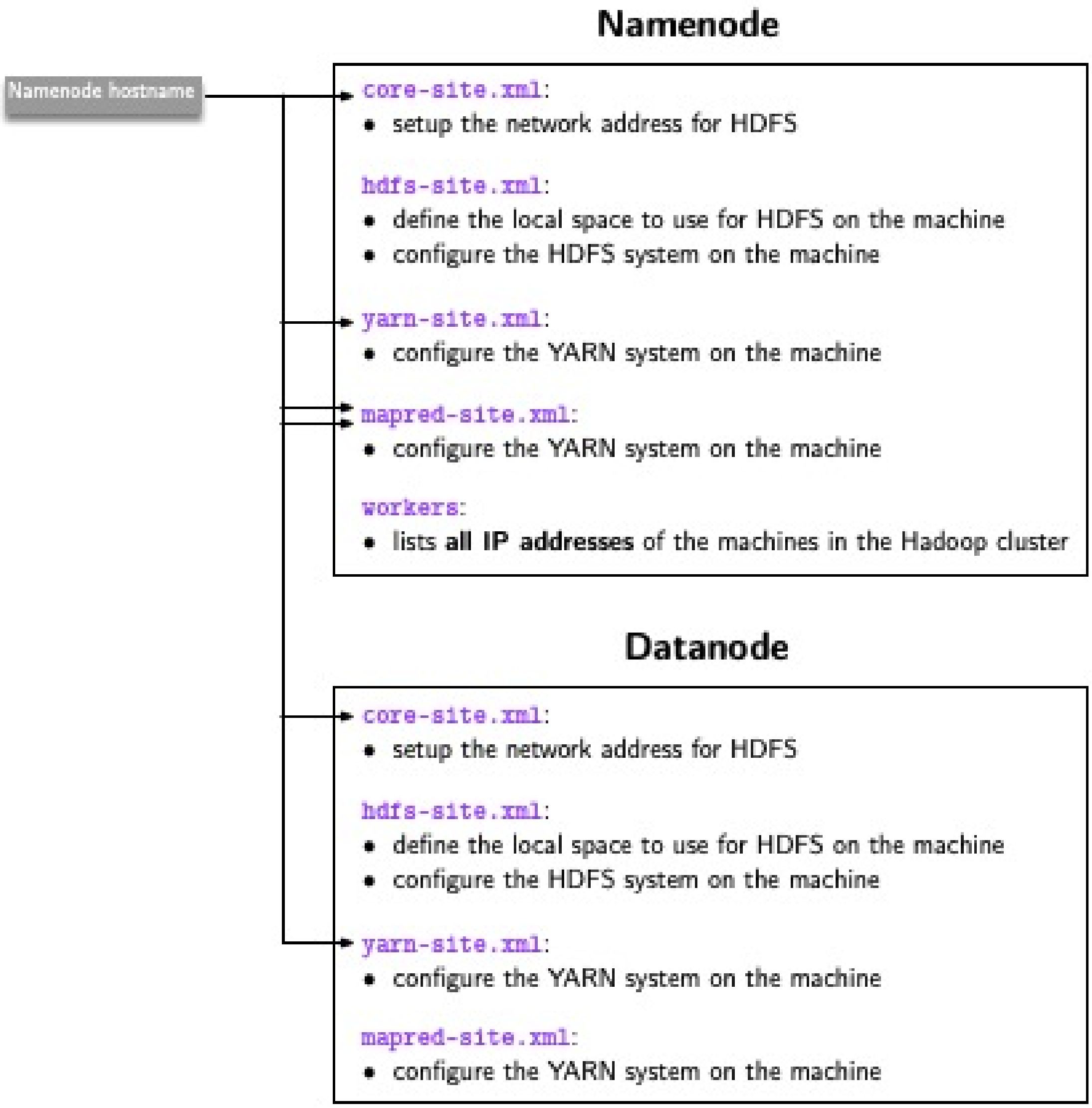
Note that the value of the property `yarn.resourcemanager.hostname` corresponds to the hostname of the virtual machine when we are installing the YARN resource manager.

d. Update the `mapred-site.xml` file located at `/opt/hadoop/etc/hadoop` to configure the MAPREDUCE subsystem. The file must contain the following lines.

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
  <property>
    <name>yarn.app.mapreduce.am.env</name>
    <value>HADOOP_MAPRED_HOME=/opt/hadoop</value>
  </property>
  <property>
    <name>mapreduce.map.env</name>
    <value>HADOOP_MAPRED_HOME=/opt/hadoop</value>
  </property>
  <property>
    <name>mapreduce.reduce.env</name>
    <value>HADOOP_MAPRED_HOME=/opt/hadoop</value>
  </property>
  <property>
    <name>yarn.app.mapreduce.am.resource.mb</name>
    <value>512</value>
  </property>
  <property>
    <name>mapreduce.map.memory.mb</name>
    <value>256</value>
  </property>
  <property>
    <name>mapreduce.reduce.memory.mb</name>
    <value>256</value>
  </property>
</configuration>
```

> If your machines have more than 2 GB of RAM or if you are interested in the numbers we specified in the YARN and MAPRED configuration files, check the Append A on the Hadoop Memory Allocaltion.

As a summary, please double check the content of the files list in the following picture, on the cluster machines:


Namenode
Namenode hostname
core-site.xml:
• setup the network address for HDFS
hdfs-site.xml:
• define the local space to use for HDFS on the machine
• configure the HDFS system on the machine
yarn-site.xml:
• configure the YARN system on the machine
mapred-site.xml:
• configure the YARN system on the machine
workers:
• lists all IP addresses of the machines in the Hadoop cluster
Datanode
core-site.xml:
• setup the network address for HDFS
hdfs-site.xml:
• define the local space to use for HDFS on the machine
• configure the HDFS system on the machine
yarn-site.xml:
• configure the YARN system on the machine
mapred-site.xml:
• configure the YARN system on the machine

0.6 ## 4. Start, test and stop Hadoop

From now, all commands will be issued from the namenode virtual machine.

a. Format the HDFS filesystem at the namenode.

```
hdfs namenode -format -force
```

b. Creating the HDFS home folder for the `hadoop` user.

```
start-dfs.sh
hadoop fs -mkdir -p /user/hadoop
stop-dfs.sh
```

c. Starting HDFS and YARN.

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

You should get an output similar to the following:

```
Stopping resourcemanager
Stopping nodemanagers
Stopping namenodes on [namenode]
Stopping datanodes
Stopping secondary namenodes [namenode]
```

d. Run the `jps` command on `namenode` and `datanodes` must give the following output (process ids can be different):

- on the `namenode`:

```
28403 DataNode
28675 SecondaryNameNode
29460 Jps
28919 ResourceManager
28183 NameNode
29308 NodeManager
```

- on the `datanodes`:

```
25721 Jps
25451 DataNode
25644 NodeManager
```

You may check logs at `/opt/hadoop/logs` on the 3 machines and check if everything is fine, or running the `hdfs dfsadmin -report` command (it must return `Live datanodes (3)`).

e. You can access Hadoop on a browser on your local machine (use the `hostname` IP addresse, not hostnames): - HDFS subsystem: `http://172.16.0.221:9870/` - YARN subsystem: `http://172.16.0.221:8088/`

f. Run an example provided by Hadoop. * Wait a minute before running, the daemons should perform some initialization steps * Ignore initial errors `No such file or directory` * Ignore logger message by logger `sasl.SaslDataTransferClient`

```
hadoop fs -rm -r input output
hadoop fs -put /opt/hadoop/etc/hadoop/ input
hadoop jar /opt/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.1.3.jar \
   grep /user/hadoop/input/*.xml /user/hadoop/output 'dfs[a-z.]+'
hadoop fs -cat output/part-r-00000
```

g. Stop HDFS and YARN.

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

0.7 Troubleshooting

If you get an error like the following:

```
[2020-01-14 08:48:28.567]Container [pid=155967,containerID=container_1578991625193_0002
_01_000023] is running 380426752B beyond the 'VIRTUAL' memory limit. Current usage:
```

```
151.6 MB of 1 GB physical memory used; 2.5 GB of 2.1 GB virtual memory used. Killing
container.
```

you are using more virtual memory than your current limit of 2.1 Gb. This can be resolved in two ways:

1. **Disable Virtual Memory Limit Checking**YARN will simply ignore the limit; in order to do this, add this to your `yarn-site.xml` *on each machine*: `bash <property> <name>yarn.nodemanager.vmem-check-enabled</name> <value>false</value> </property>` The default for this setting is `true`.

2. **Increase Virtual Memory to Physical Memory Ratio**In your `yarn-site.xml` change this to a higher value than is currently set, *on each machine*: `bash <property> <name>yarn.nodemanager.vmem-pmem-ratio</name> <value>5</value> </property>` The default is 2.1. You could also increase the amount of physical memory you allocate to a container. *Make sure you don't forget to restart yarn after you change the configuration.*

1 Appendix A. Understanding the Hadoop Memory Allocation

Memory allocation can be complex on low RAM nodes because default values are not suitable for nodes with less than 8 GB of RAM. In this section we highlight how memory allocation works for MapReduce jobs, and provide a sample configuration for 2 GB RAM nodes.

A YARN job is executed with two kind of resources:

- An **application master** (AM), which is responsible for monitoring the application and coordinating distributed executors in the cluster.
- Some **executors**, that are created by the AM to actually run the job. For a MapReduce job, they will perform map or reduce operation, in parallel.

Both are run in **containers** on **worker nodes**. Each worker node runs a **NodeManager** daemon that is responsible for container creation on the node. The whole cluster is managed by a **ResourceManager** that schedules container allocation on all the worker nodes, depending on capacity requirements and current charge.

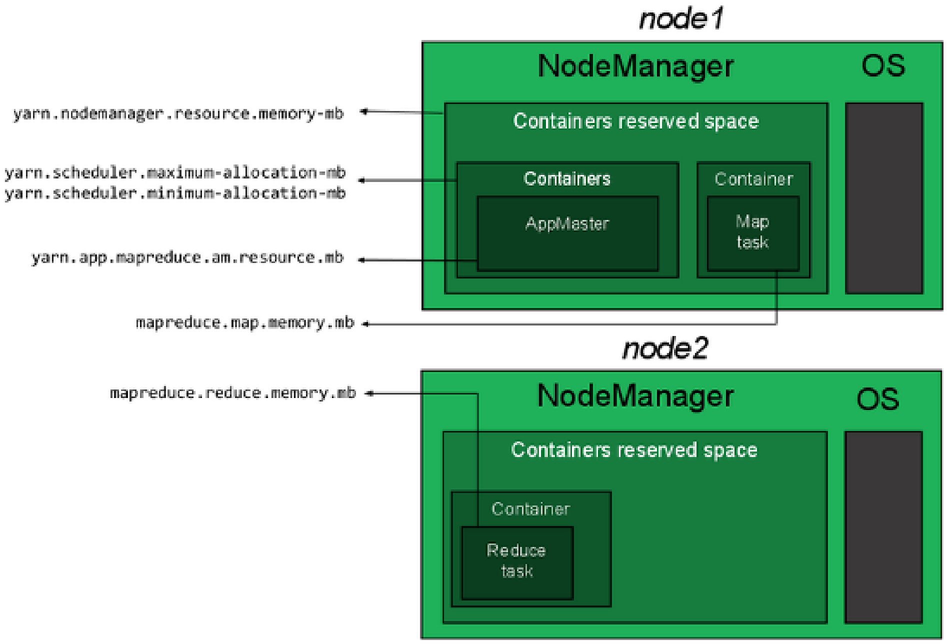
Four types of resource allocations need to be configured properly for the cluster to work. These are:

1. *How much memory can be allocated for YARN containers on a single node.* This limit should be higher than all the others; otherwise, container allocation will be rejected and applications will fail. However, it should not be the entire amount of RAM on the node. This value is configured in the `yarn-site.xml` file with the `yarn.nodemanager.resource.memory-mb` property.

2. *How much memory a single container can consume and the minimum memory allocation allowed.* A container will never be bigger than the maximum, or else allocation will fail and will always be allocated as a multiple of the minimum amount of RAM. Those values are configured in the `yarn-site.xml` file with the `yarn.scheduler.maximum-allocation-mb` and `yarn.scheduler.minimum-allocation-mb` properties.

3. *How much memory will be allocated to the ApplicationMaster.* This is a constant value that should fit in the container maximum size. This value is configured in the `mapred-site.xml` with the `yarn.app.mapreduce.am.resource.mb` property.
4. *How much memory will be allocated to each map or reduce operation.* This should be less than the maximum size. This value is configured in the `mapred-site.xml` file with the `mapreduce.map.memory.mb` and `mapreduce.reduce.memory.mb` properties.

The relationship between all those properties can be seen in the following figure:


node1
NodeManager
OS
Containers reserved space
Containers
AppMaster
Container
Map task
yarn.nodemanager.resource.memory-mb
yarn.scheduler.maximum-allocation-mb
yarn.scheduler.minimum-allocation-mb
yarn.app.mapreduce.am.resource.mb
mapreduce.map.memory.mb
node2
mapreduce.reduce.memory.mb
NodeManager
OS
Containers reserved space
Container
Reduce task

For 2 GB nodes, a working configuration may be:

Property	Value
`yarn.nodemanager.resource.memory-mb`	1536
`yarn.scheduler.maximum-allocation-mb`	1536
`yarn.scheduler.minimum-allocation-mb`	128
`yarn.app.mapreduce.am.resource.mb`	512
`mapreduce.map.memory.mb`	256
`mapreduce.reduce.memory.mb`	256