**Install Hadoop**

Download hadoop:

installation site: <http://apache.mirrors.spacedump.net/hadoop/common/stable/>

Unpacking it to a directory of your choice. In this case I'll just use ~/Programs so hadoop will be installed in ~/Programs/hadoop-2.2.0. Also, you should install a java virtual machine (JVM).

**Environmental Settings:**

export HADOOP\_PREFIX="/home/alex/Programs/hadoop-2.2.0" # Change this to where you unpacked hadoop to.

export HADOOP\_HOME=$HADOOP\_PREFIX

export HADOOP\_COMMON\_HOME=$HADOOP\_PREFIX

export HADOOP\_CONF\_DIR=$HADOOP\_PREFIX/etc/hadoop

export HADOOP\_HDFS\_HOME=$HADOOP\_PREFIX

export HADOOP\_MAPRED\_HOME=$HADOOP\_PREFIX

export HADOOP\_YARN\_HOME=$HADOOP\_PREFIX

Now lets move on to configuring the 2 main components: HDFS and YARN.

HDFS is the distributed file system used by Hadoop to store data in the cluster, capable of hosting very very (very) large files, splitting them over the nodes of the cluster. Theoretically, you don't need to have it running and files could instead be stored elsewhere like S3 or even the local file system (if using a purely local Hadoop installation). However, some applications require interactions with HDFS so you may have to set it up sooner or later if you're using third party modules. HDFS is composed of a NameNode which holds all the metadata regarding the stored files, and DataNodes (one per node in the cluster) which hold the actual data.

**HDFS Configuration(for single-node installation):**

HDFS configuration file is located at $HADOOP\_PREFIX/etc/hadoop/hdfs-site.xml

<configuration>

<property>

<name>dfs.datanode.data.dir</name>

<value>file:///home/alex/Programs/hadoop-2.2.0/hdfs/datanode</value>

<description>Comma separated list of paths on the local filesystem of a DataNode where it should store its blocks.</description>

</property>

<property>

<name>dfs.namenode.name.dir</name>

<value>file:///home/alex/Programs/hadoop-2.2.0/hdfs/namenode</value>

<description>Path on the local filesystem where the NameNode stores the namespace and transaction logs persistently.</description>

</property>

</configuration>

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://localhost/</value>

<description>NameNode URI</description>

</property>

</configuration>

**YARN Configuration**

Yarn config file: $HADOOP\_PREFIX/etc/hadoop/yarn-site.xml

<configuration>

<property>

<name>yarn.scheduler.minimum-allocation-mb</name>

<value>128</value>

<description>Minimum limit of memory to allocate to each container request at the Resource Manager.</description>

</property>

<property>

<name>yarn.scheduler.maximum-allocation-mb</name>

<value>2048</value>

<description>Maximum limit of memory to allocate to each container request at the Resource Manager.</description>

</property>

<property>

<name>yarn.scheduler.minimum-allocation-vcores</name>

<value>1</value>

<description>The minimum allocation for every container request at the RM, in terms of virtual CPU cores. Requests lower than this won't take effect, and the specified value will get allocated the minimum.</description>

</property>

<property>

<name>yarn.scheduler.maximum-allocation-vcores</name>

<value>2</value>

<description>The maximum allocation for every container request at the RM, in terms of virtual CPU cores. Requests higher than this won't take effect, and will get capped to this value.</description>

</property>

<property>

<name>yarn.nodemanager.resource.memory-mb</name>

<value>4096</value>

<description>Physical memory, in MB, to be made available to running containers</description>

</property>

<property>

<name>yarn.nodemanager.resource.cpu-vcores</name>

<value>4</value>

<description>Number of CPU cores that can be allocated for containers.</description>

</property>

</configuration>

**Daemons to start HDFS and YARN**

## Start HDFS daemons

# Format the namenode directory (DO THIS ONLY ONCE, THE FIRST TIME)

$HADOOP\_PREFIX/bin/hdfs namenode -format

# Start the namenode daemon

$HADOOP\_PREFIX/sbin/hadoop-daemon.sh start namenode

# Start the datanode daemon

$HADOOP\_PREFIX/sbin/hadoop-daemon.sh start datanode

## Start YARN daemons

# Start the resourcemanager daemon

$HADOOP\_PREFIX/sbin/yarn-daemon.sh start resourcemanager

# Start the nodemanager daemon

$HADOOP\_PREFIX/sbin/yarn-daemon.sh start nodemanager

Testing

# Run Distributed shell with 2 containers and executing the script `date`.

$HADOOP\_PREFIX/bin/hadoop jar $HADOOP\_PREFIX/share/hadoop/yarn/hadoop-yarn-applications-distributedshell-2.2.0.jar org.apache.hadoop.yarn.applications.distributedshell.Client --jar $HADOOP\_PREFIX/share/hadoop/yarn/hadoop-yarn-applications-distributedshell-2.2.0.jar --shell\_command date --num\_containers 2 --master\_memory 1024