**i.) Setting up single Node Cluster For Hadoop:**

The i3 large instance comes with 1x 475 (SSD) GB with only one disk.So RAID was not needed.

Create a mount point and mount the disk on in a folder

* sudo apt-get update
* sudo apt-get install mdadm
* lsblk
* sudo file -s /dev/nvme0n1
* sudo mkfs -t ext4 /dev/nvme0n1
* sudo mkdir /data
* sudo mount /dev/nvme0n1 /data

Change the directory to data:

Change to root user

* sudo -i
* cd /data

Dowload and install Hadoop:

* wget <http://apache.claz.org/hadoop/common/hadoop-2.8.2/hadoop-2.8.2.tar.gz>
* tar -zxvf hadoop-2.8.2.tar.gz
* rm -rf hadoop-2.8.2.tar.gz

change to hadoop/etc/hadoop/core-site.xml

Update the following:

<configuration>

<property>

<name>fs.default.name</name>

<value>hdfs://localhost:9000</value>

</property>

<property>

<name>hadoop.tmp.dir</name>

<value>/data/tmp/</value>

</property>

</configuration>

Change to hadoop/etc/hadoop/hdfs-site.xml

**Hdfs-site:**

<configuration>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

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

<value>file:/data/hadoop/hadoop\_data/hdfs/namenode</value>

</property>

<property>

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

<value>file:/data/hadoop/hadoop\_data/hdfs/datanode</value>

</property>

<property>

<name>dfs.permissions</name>

<value>false</value>

</property>

</configuration>

**Rename mapred-site.xml.template to mapred-site.xml**

Update the following in hadoop/etc/hadoop mapred-site.xml:

<configuration>

<property>

<name>yarn.app.mapreduce.am.resource.mb</name>

<value>1228</value>

</property>

<property>

<name>yarn.app.mapreduce.am.command-opts</name>

<value>-Xmx983m</value>

</property>

<property>

<name>mapreduce.map.memory.mb</name>

<value>1228</value>

</property>

<property>

<name>mapreduce.reduce.memory.mb</name>

<value>1228</value>

</property>

<property>

<name>mapreduce.map.java.opts</name>

<value>-Xmx983m</value>

</property>

<property>

<name>mapreduce.reduce.java.opts</name>

<value>-Xmx983m</value>

</property>

<property>

<name>mapred.job.shuffle.input.buffer.percent</name>

<value>0.20</value>

</property>

<property>

<name>mapreduce.cluster.local.dir</name>

<value>/data/tmp/mapred/local</value>

</property>

</configuration>

mapreduce.map.memory.mb :The upper limit that Hadoop allows to be allocated to a mapper. it was set to 1228 mb.

mapreduce.reduce.memory.mb : The upper limit that Hadoop allows to be allocated to a mapper. it was set to 1228 mb.

mapred.job.shuffle.input.buffer.percent: the mapper buffer size was set to 25%.

mapreduce.cluster.local.dir: It was set to /data/tmp/mapred/local .This is the place where the mapper stores the intermediate values it was spread across .multiple folders inorder to make space for the intermediate results.

Update the following in hadoop/etc/hadoop/yarn-site.xml

<configuration>

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

<property>

<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>

<value>org.apache.hadoop.mapred.ShuffleHandler</value>

</property>

</configuration>

Description:

**Modify the following in the Hadoop-env.sh**

* export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64
* export HADOOP\_CONF\_DIR=${HADOOP\_CONF\_DIR:-"/etc/hadoop"}
* export HADOOP\_CLASSPATH=$JAVA\_HOME/lib/tools.jar

**Install java:**

* apt-get update
* apt-get upgrade
* apt-get install default-jdk

**Update bashrc file:**

* vi .bashrc

export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64

export HADOOP\_INSTALL=/data/hadoop

export PATH=$PATH:$HADOOP\_INSTALL/bin

export PATH=$PATH:$HADOOP\_INSTALL/sbin

export HADOOP\_MAPRED\_HOME=$HADOOP\_INSTALL

export HADOOP\_COMMON\_HOME=$HADOOP\_INSTALL

export HADOOP\_HDFS\_HOME=$HADOOP\_INSTALL

export YARN\_HOME=$HADOOP\_INSTALL

export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_INSTALL/lib/native

export HADOOP\_OPTS="-Djava.library.path=$HADOOP\_INSTALL/lib"

export HADOOP\_OPTS=-Djava.net.preferIPv4Stack=true

export CONF=/data/hadoop/etc/hadoop

* source .bashrc

Configure passwordless ssh:

* ssh root@localhost
* ssh-keygen -t rsa
* cd .ssh
* cat id\_rsa.pub
* copy the the id\_rsa.pub to authorized\_keys

**Format the namenode:**

* hdfs namenode -format
* start-dfs.sh
* start-yarn.sh

Start all services: Type jps and see if all nodes are up

**Generating data using gensort:**

* wget <http://www.ordinal.com/try.cgi/gensort-linux-1.5.tar.gz>
* tar -zxvf gensort-linux-1.5.tar.gz
* rm -rf gensort-linux-1.5.tar.gz
* cd /64
* ./gensort -a 1367122448 /data/input
* hdfs dfs -mkdir /sortinput
* hdfs dfs -put /data/64/input /sortinput

copy the SortHadoop.java inside the hadoop/bin

Execute the program using

* hadoop com.sun.tools.javac.Main SortHadoop.java
* jar cf hs.jar SortHadoop\*.class
* hadoop jar hs.jar SortHadoop /sortinput/input /sortoutput/out

Transferring data to hdfs:

* hdfs dfs -get /sortinput/input /data/sol
* Do valsort on the output

2)Hadoop 1 Tb configuration

Prepare RAID for two disks:

* "Installing mdadm"
* apt-get update
* apt-get install mdadm
* "changing to raid 0"
* lsblk
* mdadm --create --verbose /dev/md0 --level=0 --name=Cloud --raid-devices=2 /dev/nvme0n1 /dev/nvme1n1
* mkfs.ext4 -L Cloud /dev/md0
* mkdir -p /data/raid
* mount LABEL=Cloud /data/raid

And the following were added in the hadoop-env.sh

export HADOOP-OPTS=”-Xmx5096m”

:

**change to hadoop/etc/hadoop/core-site.xml**

Update the following:

<configuration>

<property>

<name>fs.default.name</name>

<value>hdfs://localhost:9000</value>

</property>

<property>

<name>hadoop.tmp.dir</name>

<value>/data/raid/tmp/</value>

</property>

</configuration>

**Change the following in hadoop/etc/hadoop/hdfs-site.xml**

**Hdfs-site:**

<configuration>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

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

<value>file:/data/raid/hadoop/hadoop\_data/hdfs/namenode</value>

</property>

<property>

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

<value>file:/data/raid/hadoop/hadoop\_data/hdfs/datanode</value>

</property>

<property>

<name>dfs.permissions</name>

<value>false</value>

</property>

</configuration>

**Rename mapred-site.xml.template to mapred-site.xml**

Update the following in hadoop/etc/hadoop mapred-site.xml:

<configuration>

<property>

<name>yarn.app.mapreduce.am.resource.mb</name>

<value>1228</value>

</property>

<property>

<name>yarn.app.mapreduce.am.command-opts</name>

<value>-Xmx983m</value>

</property>

<property>

<name>mapreduce.map.memory.mb</name>

<value>1228</value>

</property>

<property>

<name>mapreduce.reduce.memory.mb</name>

<value>1228</value>

</property>

<property>

<name>mapreduce.map.java.opts</name>

<value>-Xmx983m</value>

</property>

<property>

<name>mapreduce.reduce.java.opts</name>

<value>-Xmx983m</value>

</property>

<property>

<name>mapred.job.shuffle.input.buffer.percent</name>

<value>0.20</value>

</property>

<property>

<name>mapreduce.cluster.local.dir</name>

<value>/data/tmp/mapred/local</value>

</property>

</configuration>

mapreduce.map.memory.mb :The upper limit that Hadoop allows to be allocated to a mapper. it was set to 1228 mb.

mapreduce.reduce.memory.mb : The upper limit that Hadoop allows to be allocated to a mapper. it was set to 1228 mb.

mapred.job.shuffle.input.buffer.percent: the mapper buffer size was set to 25%.

mapreduce.cluster.local.dir: It was set to /data/tmp/mapred/local .This is the place where the mapper stores the intermediate values it was spread across .multiple folders inorder to make space for the intermediate results.

**Update the following in hadoop/etc/hadoop/yarn-site.xml**

<configuration>

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

<property>

<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>

<value>org.apache.hadoop.mapred.ShuffleHandler</value>

</property>

</configuration>

3)Hadoop Multinode :

Create a master node do the configuration.And create an image of it and launch 7 additional instances along with it.

Specify the ip’s of the slaves inside the master folder.

Similarly remove localhost from the slaves and put the corresponding ip’s in the slaves.

Move the SortHadoop.java program to the Hadoop/bin and execute the program.

Execute the jar file by setting the input and output path.

* hadoop com.sun.tools.javac.Main SortHadoop.java
* jar cf hs.jar SortHadoop\*.class
* hadoop jar hs.jar SortHadoop /sortinput/input /sortoutput/out
* hadoop jar hs.jar SortHadoop /sortinput/input /sortoutput/out