MR hadoop
https://dzone.com/articles/word-count-hello-word-program-in-mapreduce

Mohan

1) ssh -i /path/my-key-pair.pem ec2-user@ec2-198-51-100-1.compute-1.amazonaws.com
 # copy public dns before doing step 1.
 # run the following command before doing step 1/
 # chmod 400 /path/my-key-pair.pem
2) sudo su -
3) passwd ubuntu
4) nano /etc/ssh/sshd_config
 # change PasswordAuthentication "no" to "yes"
 # change PermitRootLogin to “yes”
5) sudo service ssh restart
 # now you have access to the public ip.
sudo apt-get update
6) sudo apt-get install default-jre
7)
 # now you have java installed on the cloud resources.

8) wget http://apache.mirrors.tds.net/hadoop/common/hadoop-3.2.1/hadoop-3.2.1.tar.gz
9) tar zxvf hadoop-3.2.1.tar.gz
10)
10) a) cd /usr/local/hadoop
11) sudo nano .bashrc

Error

```
/*
export HADOOP_CONF_DIR=/usr/local/hadoop/etc/hadoop
export HADOOP_HOME=/usr/local/hadoop
export HADOOP_MAPRED_HOME=/usr/local/hadoop
export HADOOP_COMMON_HOME=/usr/local/hadoop
export HADOOP_HDFS_HOME=/usr/local/hadoop
export YARN_HOME=/usr/local/hadoop

# setting java path
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
export PATH=$PATH:$JAVA_HOME/bin
export PATH=$PATH:$HADOOP_HOME/bin
*/
```

12) source .bashrc

```
# change the owner
```

13) sudo chown -R ubuntu $HADOOP_HOME

14) cd $HADOOP_HOME/etc/hadoop

15) nano hadoop-env.sh

```
# update java home
```

16) hadoop version

17) ~~cd ~/.sshub~~

```
cd ~/.ssh
ssh-keygen -t rsa
```

cat id_rsa.pub >> authorized_keys

18) create config file like below on master machine and copy it to the other data nodes:

#

Host namenode

HostName 18.219.222.231

User ubuntu

IdentityFile ~/.ssh/educate.pem

Host datanode1

HostName 18.217.189.60

User ubuntu

IdentityFile ~/.ssh/educate.pem

Host datanode2

HostName 18.222.74.251

User ubuntu

IdentityFile ~/.ssh/educate.pem

#

19) copy .pem file from your machine to the master node and other data nodes, and make sure to give chmod 600 on all the pem file.

s‘scp’ command: copy file from one machine to another

scp -r filename.pem ubuntu@IP:/home/ubuntu

Login to the machine using ip

mv filename.pem ~/.ssh

20)

Okay at this stage hadoop installed successfully

21) Now try to do ssh namenode, ssh datanode1, ssh datanode2 from all machines, passwordless authentication should work.

22) edit core-site.xml with the current namenode as shown below.
copy the file to all the data nodes.
/usr/local/hadoop/etc/hadoop
23) edit hdfs-site.xml as shown below.
24) sudo mkdir -p /usr/local/hadoop/hadoop_data/hdfs/namenode
25) sudo chown -R ubuntu /usr/local/hadoop/hadoop_data/
26) sudo mkdir -p /usr/local/hadoop/hadoop_data/hdfs/datanode
sudo chown -R ubuntu /usr/local/hadoop/hadoop_data/

27) edit yarn-site.xml as shown below.
28) create masters and slaves file
for example in masters file
#
namenode

#
in slaves file
#
datanode01
datanode02
#
/usr/local/hadoop/etc/hadoop

29) update all the /etc/hosts file in all data nodes
#
172.31.40.54 namenode
172.31.39.100 datanode1
172.31.40.54 datanode2
#
NOTE: ALWAYS PRIVATE IP in HOSTS FILE, otherwise you will get an error "bind exception"
30) hadoop namenode -format

cd /usr/local/hadoop

31) $HADOOP_HOME/sbin/start-dfs.sh
32) $HADOOP_HOME/sbin/start-yarn.sh

33) All set now you should see the services running on the master node and data nodes by typing in jps.

On master node

core-site.xml:
----------s
<property>
<name>fs.defaultFS</name>
<value>hdfs://namenode:9000</value>
</property>

hdfs-site.xml

<property>
<name>dfs.namenode.name.dir</name>
<value>file:///usr/local/hadoop/hadoop_data/hdfs/namenode</value>
</property>

yarn-site.xml

<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>
<property>
<name>yarn.resourcemanager.resource-tracker.address</name>
<value>namenode:8025</value>
</property>
<property>
<name>yarn.resourcemanager.scheduler.address</name>
<value>namenode:8030</value>
</property>
<property>
<name>yarn.resourcemanager.address</name>
<value>namenode:8050</value>
</property>

On data node

core-site.xml

<property>
<name>fs.defaultFS</name>
<value>hdfs://namenode:9000</value>
</property>

hdfs-site.xml

<property>
<name>dfs.datanode.data.dir</name>
<value>file:///usr/local/hadoop/hadoop_data/hdfs/datanode
</value>
</property>

RUNNING MAPREDUCE JOB

Tutorials:

- https://www.michael-noll.com/tutorials/running-hadoop-on-ubuntu-linux-single-node-cluster/
- https://www.michael-noll.com/tutorials/running-hadoop-on-ubuntu-linux-multi-node-cluster/
- https://www.novixys.com/blog/setup-apache-hadoop-cluster-aws-ec2/
- https://blog.eduonix.com/bigdata-and-hadoop/running-a-mapreduce-program-on-amazon-ec2-hadoop-cluster-with-yarn/

Dillon:

1. Select EC2 and select Launch Instance
    a. Select *Ubuntu Server 18.04 LTS (HVM), SSD Volume Type (Free Tier Eligible)*
    b. Select *General Purpose t2.micro (Free Tier Eligible)*
    c. Click Configure Instance Details
        i. Number of instances should be 5 (1 NameNode, 1 Secondary NameNode, 3 DataNodes)
        ii. Choose a subnet in your region (doesn't really matter which).
    d. Add storage tab should be okay as is, unless changing size of data set.
    e. Click Add Tags
        i. Key: Name
        ii. Value: Hadoop
    f. Click Configure Security Group
        i. Type: 'all traffic'

        ii. Source ‘anywhere’
    g. Click Review and Launch
        i. Create new key pair and note where it is saved (.pem file), or use existing
    h. Launch instance
    i. Go to instances page and assign names to each machine
        i. NameNode, Secondary NameNode, DataNode1, DataNode2, DataNode3
    j. Make sure you can access without .pem file
        i. chmod 600 ~/Downloads/nameOfFile.pem
        ii. ssh -i ~/Desktop/Hadoop.pem ubuntu@ipAddress
        iii. sudo su
            1. go into root user
        iv. passwd ubuntu
            1. change password to ‘ubuntu’
        v. exit
            1. get out of sudo
        vi. sudo nano etc/ssh/sshd_config
            1. edit config file -- find password authentication
            2. PasswordAuthentication = yes
            3. PermitRootLogin: prohibit-pssword → yes
        vii. sudo service ssh restart
            1. require password
        viii. exit
            1. exit from ubuntu (***Now you do not need the .pem file***)
        ix. ssh ubuntu@ipAddress (prompted for password to enter)
2. Setup Hadoop on each machine/node
    a. ssh -i ~/Downloads/Hadoop.pem ubuntu@(IP address) -- in terminal
        i. Are you sure you want to continue connecting (yes/no)? Yes
    b. sudo apt-get update
    c. sudo apt-get -y dist-upgrade
    d. sudo apt-get -y install openjdk-8-jdk-headless
3. Installing Hadoop
    a. mkdir server
        i. cd server
    b. wget <Link to Hadoop 3.1.3>
        i. **http://www.trieuvan.com/apache/hadoop/common/hadoop-3.1.3/hadoop-3.1.3-src.tar.gz**
        ii. tar xvzf hadoop-3.1.3-src.tar.gz
    c. Edit hadoop-env.sh
        i. find hadoop-3.1.3-src/ -name hadoop-env.sh

ii. nano hadoop-3.1.3-src/hadoop-common-project/hadoop-common/src/main/conf/hadoop-env.sh
iii. Change
    1. # export JAVA_HOME=
    2. export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
iv. nano hadoop-3.1.3-src/hadoop-common-project/hadoop-common/src/main/conf/core-site.xml
v. Change (nnode represents the <NameNode Public DNS>)
    1. <configuration>
       </configuration>
    2. <configuration>
       <property>
        <name>fs.defaultFS</name>
        <value><nnode>:9000</value>
       </property>
       </configuration>

d. Make Data directory
    i. sudo mkdir -p /usr/local/hadoop/hdfs/data
    ii. sudo chown -R ubuntu:ubuntu /usr/local/hadoop/hdfs/data
e. NameNode Setup
    i. ssh-keygen
    ii. Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa): (Enter)
    iii. Enter passphrase: (Enter)
    iv. Enter same passphrase again: (Enter)
f. DataNode Setup Public Key
    i. MAKE SURE YOU ARE IN FOLDER ~/.ssh/
    ii. datanode1> cat id_rsa.pub >> ~/.ssh/authorized_keys
    iii. datanode2> cat id_rsa.pub >> ~/.ssh/authorized_keys
    iv. datanode3> cat id_rsa.pub >> ~/.ssh/authorized_keys
    v. cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
g. NameNode Setup SSH Config
    i. cd ~/.ssh/
    ii. nano config -- creates new config file to edit
    iii. *NO BRACKETS*
    iv. https://www.tecmint.com/ssh-passwordless-login-using-ssh-keygen-in-5-easy-steps/
        1. Host nnode
           HostName amazonAWSpublicDNS
           User ubuntu

IdentityFile ~/.ssh/id_rsa

2. Host dnode1
   HostName amazonAWSpublicDNS
   User ubuntu
   IdentityFile ~/.ssh/id_rsa

Namenode: Setup HDFS Properties

- **~/server/hadoop-3.1.3-src/hadoop-hdfs-project/hadoop-hdfs/src/main/conf**$ nano hdfs-site.xml
  - <configuration>
  - <property>
  - <name>dfs.replication</name>
  - <value>3</value>
  - </property>
  - <property>
  - <name>dfs.namenode.name.dir</name>
  - <value>file:///usr/local/hadoop/hdfs/data</value>
  - </property>
  - </configuration>

Namenode: Setup MapReduce Properties

- **~/server/hadoop-3.1.3-src/hadoop-mapreduce-project/conf$** nano mapred-site.xml
- Replace <nnode> with public dns name on aws
  - <configuration>
  - <property>
  - <name>mapreduce.jobtracker.address</name>
  - <value><nnode>:54311</value>
  - </property>
  - <property>
  - <name>mapreduce.framework.name</name>
  - <value>yarn</value>
  - </property>
  - </configuration>

Namenode: Setup YARN Properties

- **~/server/hadoop-3.1.3-src/hadoop-yarn-project/hadoop-yarn/conf**$ nano yarn-site.xml
- as before, replace <nnode> with NameNode's public DNS
    - <configuration>
    -
    - <!-- Site specific YARN configuration properties -->
    - <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>
    - <property>
    - <name>yarn.resourcemanager.hostname</name>
    - <value><nnode></value>
    - </property>
    -
    - </configuration>

Namenode: Setup Master and Slaves

- **On the NameNode, create ~/server/hadoop-2.7.3/etc/hadoop/masters with the following (replace <nnode> with the NameNode's public DNS):**
    - <nnode>
- **Also replace all content in ~/server/hadoop-2.7.3/etc/hadoop/slaves with (replace each of <dnode1>, etc with the appropriate DateNode's public DNS):**
    - <dnode1>
    - <dnode2>
    - <dnode3>

Configuring Data Nodes

After covering configuration common to both NameNode and DataNodes, we have a little bit of configuring specific to DataNodes. On each data node, edit the file `~/server/hadoop-2.7.3/etc/hadoop/hdfs-site.xml` and replace the following:

- <configuration>
- </configuration>
- With:
- <property>
- <name>dfs.replication</name>
- <value>3</value>
- </property>
- <property>
- <name>dfs.datanode.data.dir</name>
- <value>file:///usr/local/hadoop/hdfs/data</value>
- </property>

./hadoop-3.1.3-src/hadoop-hdfs-project/hadoop-hdfs/src/main/bin/hdfs namenode -format
ERROR: Cannot execute /home/ubuntu/server/hadoop-3.1.3-src/hadoop-hdfs-project/hadoop-hdfs/src/main/bin/../libexec/hdfs-config.sh.

```
[ubuntu@ip-172-31-5-166:~/server/hadoop-3.1.3-src/hadoop-hdfs-project/hadoop-hdfs/src/main/bin$ ./start-dfs.sh
ERROR: Cannot execute /home/ubuntu/server/hadoop-3.1.3-src/hadoop-hdfs-project/hadoop-hdfs/src/main/bin/../libexec/hdfs-config.sh.
```

```
[ubuntu@ip-172-31-5-166:~/server$ ./hadoop-3.1.3-src/hadoop-hdfs-project/hadoop-hdfs/src/main/bin/hdfs namenode -format
ERROR: Cannot execute /home/ubuntu/server/hadoop-3.1.3-src/hadoop-hdfs-project/hadoop-hdfs/src/main/bin/../libexec/hdfs-config.sh.
```

```
# Technically, the only required environment variable is JAVA_HOME.
# All others are optional.  However, the defaults are probably not
# preferred.  Many sites configure these options outside of Hadoop,
# such as in /etc/profile.d

# The java implementation to use. By default, this environment
# variable is REQUIRED on ALL platforms except OS X!
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
```

1. Creating 4 instances of Ubuntu Server 18.04 LTS using Amazon EC2.
    a. Go to: https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#Instances:sort=tag:Name
        i. Click Launch Instance
        ii. Select Ubuntu Server 18.04 LTS (HVM)
        iii. Choose t2.micro(Free Tier)
        iv. Click Next to Configure Instance Details
            1. Number of Instances: 4
            2. Subnet: subnet-1d214767 | Default in us-east-2b

a. Doesn't really matter, but I want all machines in the same region

v. Click Next add storage

vi. Click Next add Tags

1. Click Add Tag
2. Key: Name, Value: Hadoop

vii. Click Next Configure Security Group

1. Use an existing security group
2. Security group name: allopen
3. Type: All Traffic
4. Source: Anywhere

viii. Click Review and Launch

ix. Click Launch

1. Select a key Pair: Hadoop
2. Click check box

b. Go to the instances page. Assign names to distinguish them (Hadoop-datanode1, Hadoop-datanode2, Hadoop-datanode3, Hadoop-namenode)

2. Setting up Hadoop on each individual node

a. Open terminal

i. ssh -i ~/Downloads/Hadoop.pem ubuntu@3.15.163.121 (or whatever IP)

1. Are you sure you want to continue connecting (yes/no)? yes

ii. sudo apt-get update && sudo apt-get -y dist-upgrade

1. A new version of /boot/grub/menu.lst is available, but the version installed currently has been locally modified. What would you like to do about menu.lst?

iii. sudo apt-get -y install openjdk-8-jdk-headless

b. Installing Hadoop

i. mkdir server

1. cd server

ii. wget <Link to Hadoop 3.1.3>

1. https://www.apache.org/dyn/closer.cgi/hadoop/common/hadoop-3.1.3/hadoop-3.1.3-src.tar.gz

iii. tar xvzf hadoop-3.1.3-src.tar.gz

c. Edit hadoop-env.sh

i. find hadoop-3.1.3-src/ -name hadoop-env.sh

ii. nano hadoop-3.1.3-src/hadoop-common-project/hadoop-common/src/main/conf/hadoop-env.sh

iii. Change

1. # export JAVA_HOME=

2. export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64

iv. nano hadoop-3.1.3-src/hadoop-common-project/hadoop-common/src/main/conf/core-site.xml

v. Change (nnode represents the <NameNode Public DNS>)

1. <configuration>
   </configuration>
2. <configuration>
   <property>
   <name>fs.defaultFS</name>
   <value><nnode>:9000</value>
   </property>
   </configuration>

3. UPDATE: In file conf/core-site.xml, conf/mapred-site.xml,conf/hdfs-site.xml

a. Make Data directory

i. sudo mkdir -p /usr/local/hadoop/hdfs/data

ii. sudo chown -R ubuntu:ubuntu /usr/local/hadoop/hdfs/data

b. NameNode Setup

c. ssh-keygen -t rsa -P ""

i. ssh-keygen

ii. Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa): (Hit Enter)

iii. Enter passphrase: (Hit Enter)

iv. Enter same passphrase again: (Hit Enter)

v.

```
ubuntu@ip-172-31-18-23:~/server$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_rsa.
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:O87Rr8L0Why0X6aS+YXy1V8nYbS3wCMeCqdNZpXrwRw ubuntu@ip-172-31-18-23
The key's randomart image is:
+---[RSA 2048]----+
|                 |
|              .  |
|           .E  . |
|          .+.+. . |
|        .S=oB +* .|
|         X++=+*o+.|
|        +++Oo+ +.+|
|        oo+.* o .+|
|         +o..+   .|
+----[SHA256]-----+
```

d. Datanode Setup Public Key

i. cat id_rsa.pub >> ~/.ssh/authorized_keys

ii. cat: id_rsa.pub: No such file or directory

iii. cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys

iv.

e. Issues formatting the Namenode
   i. hduser@ubuntu:~$ /usr/local/hadoop/bin/hadoop namenode -format
f. find hadoop-3.1.3-src/ -name hadoop-env.sh
g. nano hadoop-3.1.3-src/hadoop-hdfs-project/hadoop-hdfs/src/main/conf/hdfs-site.xml
h.
i. find hadoop-3.1.3-src/ -name mapred-site.xml
j.
k. hadoop-3.1.3-src/hadoop-mapreduce-project/conf/mapred-site.xml
l. find hadoop-3.1.3-src/ -name yarn-site.xml
m. hadoop-3.1.3-src/hadoop-yarn-project/hadoop-yarn/conf/yarn-site.xml

hadoop-3.1.3-src/hadoop-hdfs-project/hadoop-hdfs-client/src/test/java/org/apache/hadoop/hdfs/server/namenode

hadoop-3.1.3-src/hadoop-hdfs-project/hadoop-hdfs-client/src/main/java/org/apache/hadoop/hdfs/server/namenode

hadoop-3.1.3-src/hadoop-hdfs-project/hadoop-hdfs/src/test/java/org/apache/hadoop/hdfs/server/namenode

hadoop-3.1.3-src/hadoop-hdfs-project/hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/namenode

ubuntu@ip-172-31-1-49:**/usr/local/hadoop**$ bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-3.2.1.jar wordcount input output
2020-02-19 03:08:04,674 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2020-02-19 03:08:04,999 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).

2020-02-19 03:08:04,999 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2020-02-19 03:08:05,252 INFO input.FileInputFormat: Total input files to process : 1
2020-02-19 03:08:05,303 INFO mapreduce.JobSubmitter: number of splits:1
2020-02-19 03:08:05,656 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local1948499963_0001
2020-02-19 03:08:05,656 INFO mapreduce.JobSubmitter: Executing with tokens: []
2020-02-19 03:08:05,923 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2020-02-19 03:08:05,923 INFO mapreduce.Job: Running job: job_local1948499963_0001
2020-02-19 03:08:05,947 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2020-02-19 03:08:05,961 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2020-02-19 03:08:05,961 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2020-02-19 03:08:05,964 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
2020-02-19 03:08:06,044 INFO mapred.LocalJobRunner: Waiting for map tasks
2020-02-19 03:08:06,045 INFO mapred.LocalJobRunner: Starting task: attempt_local1948499963_0001_m_000000_0
2020-02-19 03:08:06,092 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2020-02-19 03:08:06,093 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2020-02-19 03:08:06,141 INFO mapred.Task: Using ResourceCalculatorProcessTree : [ ]
2020-02-19 03:08:06,148 INFO mapred.MapTask: Processing split: file:/usr/local/hadoop/input/input:0+140
2020-02-19 03:08:06,387 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
2020-02-19 03:08:06,387 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
2020-02-19 03:08:06,387 INFO mapred.MapTask: soft limit at 83886080
2020-02-19 03:08:06,387 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
2020-02-19 03:08:06,387 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
2020-02-19 03:08:06,391 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
2020-02-19 03:08:06,409 INFO mapred.LocalJobRunner:
2020-02-19 03:08:06,411 INFO mapred.MapTask: Starting flush of map output
2020-02-19 03:08:06,411 INFO mapred.MapTask: Spilling map output
2020-02-19 03:08:06,411 INFO mapred.MapTask: bufstart = 0; bufend = 260; bufvoid = 104857600
2020-02-19 03:08:06,411 INFO mapred.MapTask: kvstart = 26214396(104857584); kvend = 26214280(104857120); length = 117/6553600
2020-02-19 03:08:06,440 INFO mapred.MapTask: Finished spill 0

```
2020-02-19 03:08:06,460 INFO mapred.Task:
Task:attempt_local1948499963_0001_m_000000_0 is done. And is in the process of
committing
2020-02-19 03:08:06,466 INFO mapred.LocalJobRunner: map
2020-02-19 03:08:06,467 INFO mapred.Task: Task
'attempt_local1948499963_0001_m_000000_0' done.
2020-02-19 03:08:06,485 INFO mapred.Task: Final Counters for
attempt_local1948499963_0001_m_000000_0: Counters: 18
        File System Counters
                FILE: Number of bytes read=316844
                FILE: Number of bytes written=841649
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
        Map-Reduce Framework
                Map input records=5
                Map output records=30
                Map output bytes=260
                Map output materialized bytes=70
                Input split bytes=99
                Combine input records=30
                Combine output records=6
                Spilled Records=6
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=20
                Total committed heap usage (bytes)=144396288
        File Input Format Counters
                Bytes Read=156
2020-02-19 03:08:06,487 INFO mapred.LocalJobRunner: Finishing task:
attempt_local1948499963_0001_m_000000_0
2020-02-19 03:08:06,488 INFO mapred.LocalJobRunner: map task executor complete.
2020-02-19 03:08:06,493 INFO mapred.LocalJobRunner: Waiting for reduce tasks
2020-02-19 03:08:06,494 INFO mapred.LocalJobRunner: Starting task:
attempt_local1948499963_0001_r_000000_0
2020-02-19 03:08:06,513 INFO output.FileOutputCommitter: File Output Committer Algorithm
version is 2
2020-02-19 03:08:06,513 INFO output.FileOutputCommitter: FileOutputCommitter skip
cleanup _temporary folders under output directory:false, ignore cleanup failures: false
2020-02-19 03:08:06,513 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
```

2020-02-19 03:08:06,519 INFO mapred.ReduceTask: Using ShuffleConsumerPlugin: org.apache.hadoop.mapreduce.task.reduce.Shuffle@53b01f8
2020-02-19 03:08:06,523 WARN impl.MetricsSystemImpl: JobTracker metrics system already initialized!
2020-02-19 03:08:06,558 INFO reduce.MergeManagerImpl: MergerManager: memoryLimit=174555136, maxSingleShuffleLimit=43638784, mergeThreshold=115206392, ioSortFactor=10, memToMemMergeOutputsThreshold=10
2020-02-19 03:08:06,574 INFO reduce.EventFetcher: attempt_local1948499963_0001_r_000000_0 Thread started: EventFetcher for fetching Map Completion Events
2020-02-19 03:08:06,615 INFO reduce.LocalFetcher: localfetcher#1 about to shuffle output of map attempt_local1948499963_0001_m_000000_0 decomp: 66 len: 70 to MEMORY
2020-02-19 03:08:06,623 INFO reduce.InMemoryMapOutput: Read 66 bytes from map-output for attempt_local1948499963_0001_m_000000_0
2020-02-19 03:08:06,625 INFO reduce.MergeManagerImpl: closeInMemoryFile -> map-output of size: 66, inMemoryMapOutputs.size() -> 1, commitMemory -> 0, usedMemory ->66
2020-02-19 03:08:06,628 INFO reduce.EventFetcher: EventFetcher is interrupted.. Returning
2020-02-19 03:08:06,629 INFO mapred.LocalJobRunner: 1 / 1 copied.
2020-02-19 03:08:06,629 INFO reduce.MergeManagerImpl: finalMerge called with 1 in-memory map-outputs and 0 on-disk map-outputs
2020-02-19 03:08:06,638 INFO mapred.Merger: Merging 1 sorted segments
2020-02-19 03:08:06,638 INFO mapred.Merger: Down to the last merge-pass, with 1 segments left of total size: 62 bytes
2020-02-19 03:08:06,642 INFO reduce.MergeManagerImpl: Merged 1 segments, 66 bytes to disk to satisfy reduce memory limit
2020-02-19 03:08:06,642 INFO reduce.MergeManagerImpl: Merging 1 files, 70 bytes from disk
2020-02-19 03:08:06,643 INFO reduce.MergeManagerImpl: Merging 0 segments, 0 bytes from memory into reduce
2020-02-19 03:08:06,643 INFO mapred.Merger: Merging 1 sorted segments
2020-02-19 03:08:06,646 INFO mapred.Merger: Down to the last merge-pass, with 1 segments left of total size: 62 bytes
2020-02-19 03:08:06,646 INFO mapred.LocalJobRunner: 1 / 1 copied.
2020-02-19 03:08:06,649 INFO Configuration.deprecation: mapred.skip.on is deprecated. Instead, use mapreduce.job.skiprecords
2020-02-19 03:08:06,653 INFO mapred.Task: Task:attempt_local1948499963_0001_r_000000_0 is done. And is in the process of committing
2020-02-19 03:08:06,658 INFO mapred.LocalJobRunner: 1 / 1 copied.
2020-02-19 03:08:06,661 INFO mapred.Task: Task attempt_local1948499963_0001_r_000000_0 is allowed to commit now

```
2020-02-19 03:08:06,667 INFO output.FileOutputCommitter: Saved output of task
'attempt_local1948499963_0001_r_000000_0' to file:/usr/local/hadoop/output
2020-02-19 03:08:06,668 INFO mapred.LocalJobRunner: reduce > reduce
2020-02-19 03:08:06,669 INFO mapred.Task: Task
'attempt_local1948499963_0001_r_000000_0' done.
2020-02-19 03:08:06,670 INFO mapred.Task: Final Counters for
attempt_local1948499963_0001_r_000000_0: Counters: 24
        File System Counters
                FILE: Number of bytes read=317016
                FILE: Number of bytes written=841771
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
        Map-Reduce Framework
                Combine input records=0
                Combine output records=0
                Reduce input groups=6
                Reduce shuffle bytes=70
                Reduce input records=6
                Reduce output records=6
                Spilled Records=6
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=0
                Total committed heap usage (bytes)=144396288
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Output Format Counters
                Bytes Written=52
2020-02-19 03:08:06,672 INFO mapred.LocalJobRunner: Finishing task:
attempt_local1948499963_0001_r_000000_0
2020-02-19 03:08:06,672 INFO mapred.LocalJobRunner: reduce task executor complete.
2020-02-19 03:08:06,946 INFO mapreduce.Job: Job job_local1948499963_0001 running in
uber mode : false
2020-02-19 03:08:06,947 INFO mapreduce.Job:  map 100% reduce 100%
```

```
2020-02-19 03:08:06,949 INFO mapreduce.Job: Job job_local1948499963_0001 completed successfully
2020-02-19 03:08:06,964 INFO mapreduce.Job: Counters: 30
        File System Counters
                FILE: Number of bytes read=633860
                FILE: Number of bytes written=1683420
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
        Map-Reduce Framework
                Map input records=5
                Map output records=30
                Map output bytes=260
                Map output materialized bytes=70
                Input split bytes=99
                Combine input records=30
                Combine output records=6
                Reduce input groups=6
                Reduce shuffle bytes=70
                Reduce input records=6
                Reduce output records=6
                Spilled Records=12
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=20
                Total committed heap usage (bytes)=288792576
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=156
        File Output Format Counters
                Bytes Written=52
ubuntu@ip-172-31-1-49:/usr/local/hadoop$ bin/hdfs dfs -cat output/*
a       5
example 5
```

file	5
is	5
text	5
this	5