

YARN ASSIGNMENT

(1) Setup yarn in pseudo-distributed mode

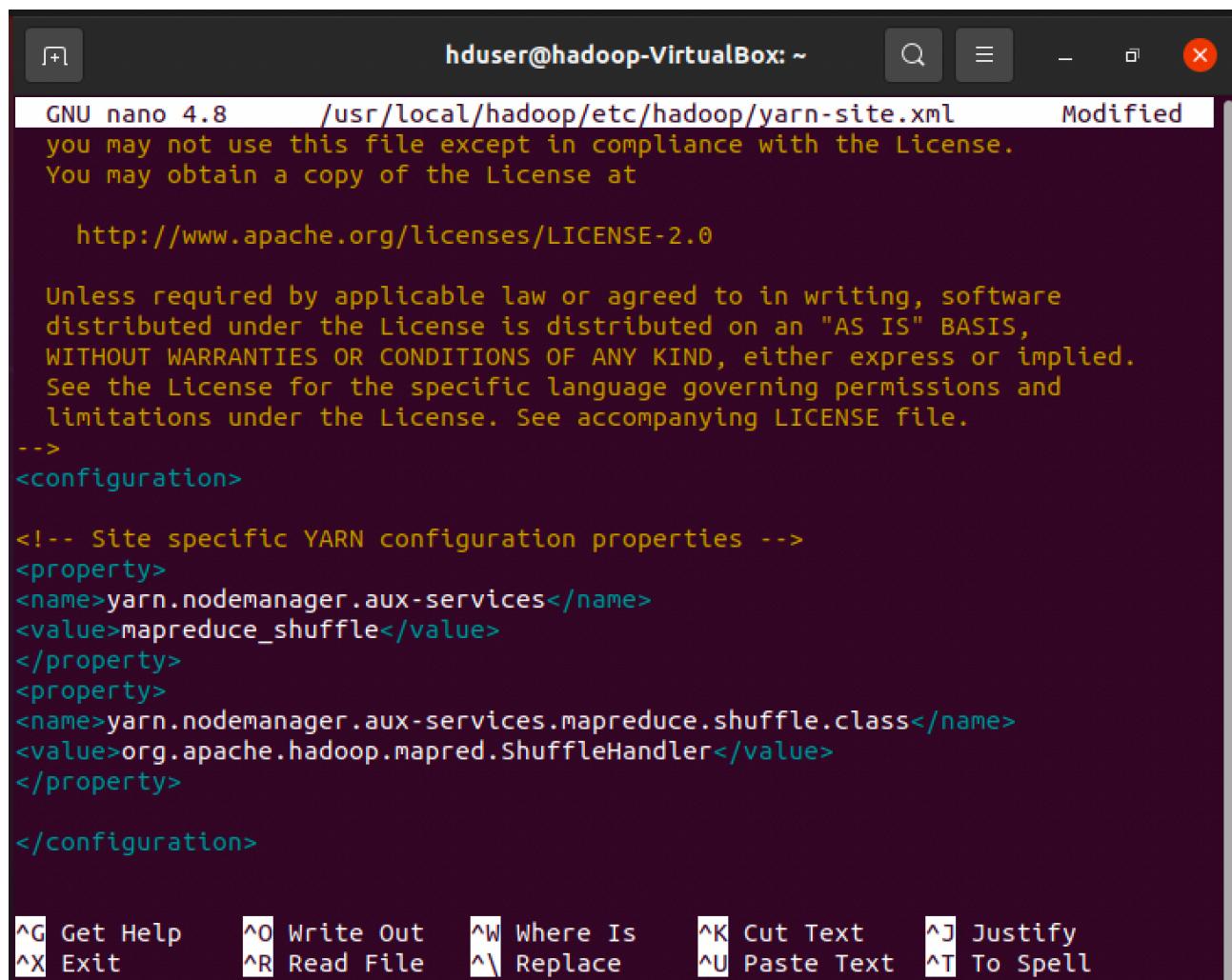
Prerequisite:-

1. Install Hadoop by following the instruction in the document given in the below GitHub repository.

<https://github.com/prabhjotbedi75/Task21>

STEPS:-

1. Once all of your Hadoop Configuration is done we need to do some extra configuration for setting up Yarn.
2. Open `/usr/local/hadoop/etc/hadoop/yarn-site.xml` file and configure it as below:



The screenshot shows a terminal window with the title bar "hduser@hadoop-VirtualBox: ~". The window contains the content of the `yarn-site.xml` file. The file starts with a license notice from Apache, followed by configuration sections for YARN services and properties. The bottom of the screen shows the nano editor's command-line interface with various keyboard shortcuts.

```
GNU nano 4.8      /usr/local/hadoop/etc/hadoop/yarn-site.xml      Modified
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.

-->
<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>

</configuration>

^G Get Help      ^O Write Out    ^W Where Is      ^K Cut Text      ^J Justify
^X Exit          ^R Read File     ^\ Replace       ^U Paste Text    ^T To Spell
```

3. After that configure `/usr/local/hadoop/etc/hadoop/mapred-site.xml` file as below:

```
GNU nano 4.8      /usr/local/hadoop/etc/hadoop/mapred-site.xml

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.

-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
  <name>mapred.job.tracker</name>
  <value>localhost:54311</value>
  <description>The host and port that the MapReduce job tracker runs
at. If "local", then jobs are run in-process as a single map
and reduce task.
</description>
</property>
<property>
  <name>mapreduce.framework.name</name>
  <value>yarn</value>
</property>
</configuration>
```

4. Now start Hadoop using the below command:

```
hduser@hadoop-VirtualBox:~$ start-dfs.sh
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.security.authentication
.util.KerberosUtil (file:/usr/local/hadoop/share/hadoop/common/lib/hadoop-auth-
2.10.1.jar) to method sun.security.krb5.Config.getInstance()
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop
.security.authentication.util.KerberosUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflec-
tive access operations
WARNING: All illegal access operations will be denied in a future release
22/03/12 14:00:07 WARN util.NativeCodeLoader: Unable to load native-hadoop libr-
ary for your platform... using builtin-java classes where applicable
Starting namenodes on [localhost]
localhost: starting namenode, logging to /usr/local/hadoop/logs/hadoop-hduser-n
amenode-hadoop-VirtualBox.out
localhost: WARNING: An illegal reflective access operation has occurred
localhost: WARNING: Illegal reflective access by org.apache.hadoop.security.aut-
hentication.util.KerberosUtil (file:/usr/local/hadoop/share/hadoop/common/lib/h
adoop-auth-2.10.1.jar) to method sun.security.krb5.Config.getInstance()
localhost: WARNING: Please consider reporting this to the maintainers of org.ap-
ache.hadoop.security.authentication.util.KerberosUtil
localhost: WARNING: Use --illegal-access=warn to enable warnings of further ill-
egal reflective access operations
localhost: WARNING: All illegal access operations will be denied in a future re
lease
```

5. Start yarn using the the below command:

```
hduser@hadoop-VirtualBox:~$ start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-hduser-resourc
emanager-hadoop-VirtualBox.out
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.security.authentication
.util.KerberosUtil (file:/usr/local/hadoop/share/hadoop/common/lib/hadoop-auth-
2.10.1.jar) to method sun.security.krb5.Config.getInstance()
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop
.security.authentication.util.KerberosUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflec
tive access operations
WARNING: All illegal access operations will be denied in a future release
localhost: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-hduser-
nodemanager-hadoop-VirtualBox.out
```

6. Now use jps command to list all the running processes.

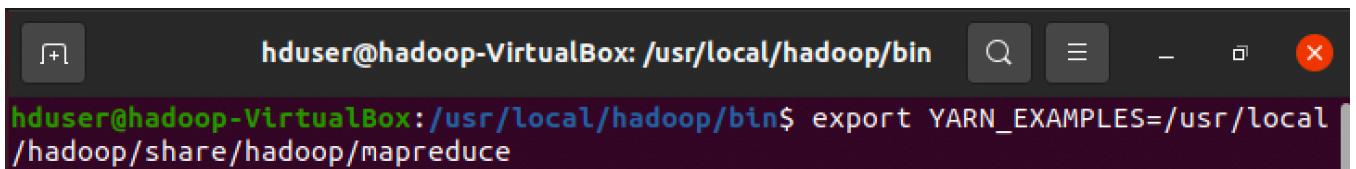
```
hduser@hadoop-VirtualBox:~$ jps
35857 Jps
35238 SecondaryNameNode
35389 ResourceManager
34845 NameNode
35533 NodeManager
35023 DataNode
```

Now are yarn is setup in pseudo distributed mode.

(2) Run sample mapreduce jobs on a input file of your choice

STEPS:-

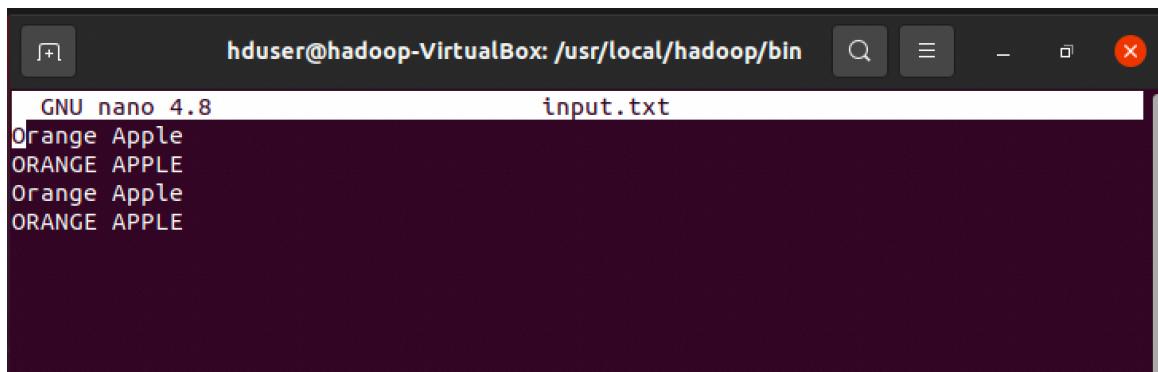
1. In order to perform map-reduce jobs on an input file we have to first export YARN_EXAMPLES in /usr/local/hadoop/bin by using the below command.



A screenshot of a terminal window titled "hduser@hadoop-VirtualBox: /usr/local/hadoop/bin". The window shows the command "hduser@hadoop-VirtualBox:/usr/local/hadoop/bin\$ export YARN_EXAMPLES=/usr/local/hadoop/share/hadoop/mapreduce" being typed in. The terminal interface includes standard window controls like minimize, maximize, and close buttons.

```
hduser@hadoop-VirtualBox:/usr/local/hadoop/bin$ export YARN_EXAMPLES=/usr/local/hadoop/share/hadoop/mapreduce
```

2. Now create a input.txt file and add some content in it, on which you want to run your mapreduce job.



The screenshot shows a terminal window titled "hduser@hadoop-VirtualBox: /usr/local/hadoop/bin". The file "input.txt" is open in a nano editor. The content of the file is:

```
GNU nano 4.8           input.txt
Orange Apple
ORANGE APPLE
Orange Apple
ORANGE APPLE
```

3. Now add this input.txt file to your HDFS using the below command:

```
hduser@hadoop-VirtualBox:/usr/local/hadoop/bin$ hadoop fs -put input.txt /
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.KerberosUtil (file:/usr/local/hadoop/share/hadoop/common/lib/hadoop-auth-2.10.1.jar) to method sun.security.krb5.Config.getInstance()
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.authentication.util.KerberosUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
22/03/12 17:27:31 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
```

4. Now use the below command to run your map-reduce job:

```
hduser@hadoop-VirtualBox:/usr/local/hadoop/bin$ ./yarn jar $YARN_EXAMPLES/hadoop-mapreduce-examples-2.10.1.jar wordcount /input.txt /output.txt
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.KerberosUtil (file:/usr/local/hadoop/share/hadoop/common/lib/hadoop-auth-2.10.1.jar) to method sun.security.krb5.Config.getInstance()
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.authentication.util.KerberosUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
22/03/12 17:29:56 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
22/03/12 17:29:57 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
22/03/12 17:29:57 INFO input.FileInputFormat: Total input files to process : 1
22/03/12 17:29:57 INFO mapreduce.JobSubmitter: number of splits:1
22/03/12 17:29:58 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_647073833457_0001
22/03/12 17:29:58 INFO conf.Configuration: resource-types.xml not found
22/03/12 17:29:58 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
22/03/12 17:29:58 INFO resource.ResourceUtils: Adding resource type - name = memory-mb, units = Mi, type = COUNTABLE
22/03/12 17:29:58 INFO resource.ResourceUtils: Adding resource type - name = vcores, units = , type = COUNTABLE
22/03/12 17:29:59 INFO impl.YarnClientImpl: Submitted application application_647073833457_0001
22/03/12 17:29:59 INFO mapreduce.Job: The url to track the job: http://hadoop-VirtualBox:8088/tasktracker/application_647073833457_0001/
```

```
hduser@hadoop-VirtualBox:/usr/local/hadoop/bin$ 
Map output materialized bytes=56
Input split bytes=97
Combine input records=8
Combine output records=4
Reduce input groups=4
Reduce shuffle bytes=56
Reduce input records=4
Reduce output records=4
Spilled Records=8
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=40
CPU time spent (ms)=920
Physical memory (bytes) snapshot=414957568
Virtual memory (bytes) snapshot=4123848704
Total committed heap usage (bytes)=220200960
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=52
File Output Format Counters
Bytes Written=34
hduser@hadoop-VirtualBox:/usr/local/hadoop/bin$
```

5. Now use the below command to see the output of your mapreduce-job:

```
hduser@hadoop-VirtualBox:/usr/local/hadoop/bin$ hadoop fs -cat /output.txt/*
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.security.authentication
.util.KerberosUtil (file:/usr/local/hadoop/share/hadoop/common/lib/hadoop-auth-
2.10.1.jar) to method sun.security.krb5.Config.getInstance()
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop
.security.authentication.util.KerberosUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflec-
tive access operations
WARNING: All illegal access operations will be denied in a future release
22/03/12 17:31:48 WARN util.NativeCodeLoader: Unable to load native-hadoop libr-
ary for your platform... using builtin-java classes where applicable
APPLE    2
Apple    2
ORANGE   2
Orange   2
hduser@hadoop-VirtualBox:/usr/local/hadoop/bin$
```

(3) Read up on various failure scenarios for an application.

1. **Application Master Failure** An application master sends periodic heartbeats to the ResourceManager, and in the event of application master failure, the ResourceManager will detect the failure and starts a new instance of the master running in a new container, it will use the job history to recover the state of the tasks that were already run by the failed application master so they don't have to be run'd again. Recovery is enabled by default, but can be disabled by setting **yarn.app.mapreduce.am.job.recovery.enable** to false.
2. **Node Manager Failure** Nodemanager will send a heartbeat signal for every 3 seconds to the resource manager. If the node manager fails due to crash or running very slowly the RM will wait for the heartbeat for 10 minutes. If it is not received, the RM will remove the node from its pool to schedule the containers.If an AM is running in the failed node manager, then the AM will be launched in another node The completed tasks of the dead node manager are to be rerun if they belong to incomplete jobsNode managers may be blacklisted if the number of failures for the application is high, even if the node manager itself has not failed. Blacklisting is done by the application master if more than three tasks fail on a node manager.
3. **Resource Manager Failure** Resource Manager is the single point of failure in YARN. To achieve high availability (HA), it is necessary to run a pair of resource managers in an active-standby configuration. If the active resource manager fails, then the standby can take over without a significant interruption to the client.The transition of a resource manager from standby to active is handled by a failover controller. The default failover controller is an automatic one, which uses ZooKeeper leader election to ensure that there is only a single active resource manager at one time.

4. **Task Failure** Task failure generally occurs due to run time exceptions or due to sudden exit of task JVM. The task will send a heartbeat signal to the AM for every 3 seconds and if the AM doesn't receive any update for 10 minutes, it will **consider the task as failed and will rerun the task attempt**. When the application master is notified of a task attempt that has failed, it will reschedule execution of the task. The application master will try to avoid rescheduling the task on a node manager where it has previously failed. Furthermore, if a task fails four times, it will not be retried again. The job will return the failed status.