Configuring NodeLabels in YARN

Create a Label Directory in HDFS:

```
sudo -u hdfs hdfs dfs -mkdir -p /user/yarn

sudo -u hdfs hdfs dfs -chown -R yarn:yarn /user/yarn

sudo -u hdfs hdfs dfs -chmod -R 700 /user/yarn

sudo -u hdfs hdfs dfs -mkdir -p /yarn/node-labels

sudo -u hdfs hdfs dfs -chown -R yarn:yarn /yarn

sudo -u hdfs hdfs dfs -chmod -R 700 /yarn
```

Configure YARN for Node Labels:

- Enable below mentioned property in `/etc/hadoop/conf/yarn-site.xml` in YARN Resource Manager VM

```
<property>
      <name>yarn.node-labels.enabled</name>
      <value>true</value>
</property>
```

```
<property>
      <name>yarn.node-labels.fs-store.root-dir</name>
      <value>hdfs://<fs-service-name>
/<absolute_path_to_node_label_directory></value>
      <description> set like this:  hdfs://argoidsaasstage1hdfs/yarn
/node-labels/</description>
 </property>
```

- Start or Restart the YARN ResourceManager

Create and Assign Node Labels

- create Node labels by referring the example here , here `nodelabel_spark_pi_dag` is created

  ```
  sudo -u yarn yarn rmadmin -addToClusterNodeLabels "nodelabel_spark_pi_dag(exclusive=true)"
  ```
- Assign NodeLabels to NodeManager , here Node Label `nodelabel_spark_pi_dag` is assigned to `argoid-saas-stage-host-041` YARN NodeManager VM

  ```
   sudo -u yarn yarn rmadmin -replaceLabelsOnNode "argoid-saas-stage-host-041.internal.cloudapp.
  net=nodelabel_spark_pi_dag"
  ```

Associating Node Labels with YARN Queue

Set following mentioned properties in `/etc/hadoop/conf/capacity-scheduler.xml`file of YARN ResourceManager VMs

- Add the new Nodelabel which is `nodelabel_spark_pi_dag` here

```
<property>
 <name>yarn.scheduler.capacity.root.default.accessible-node-labels<
/name>
 <value>nodelabel_spark_pi_dag</value>
 <description>
      you can add multiple NodeLabels with comma seperation eg:
nodelabel_a,nodelabel_spark_pi_dag,nodelabel_b
 </description>
 </property>
```

- specify NodeLabel capacity to 100%

```
<property>
 <name>yarn.scheduler.capacity.root.accessible-node-labels.
nodelabel_spark_pi_dag.capacity</name>
 <value>100</value>
 </property>

<property>
 <name>yarn.scheduler.capacity.root.default.accessible-node-labels.
nodelabel_spark_pi_dag.capacity</name>
 <value>100</value>
 </property>
```

Refresh Queues

```
sudo -u yarn yarn rmadmin  -refreshQueues
```

Confirm Node Label Assignments

- List all running nodes in the cluster: `sudo -u yarn yarn node -list`
- List all node labels in the cluster: `sudo -u yarn yarn cluster --list-node-labels`
- List the status of a node (includes node labels): `yarn node -status <NodeManager Host Name>`
- Trigger Spark Job to Yarn with new NodeLabel
  `/opt/spark/bin/spark-submit --class org.apache.spark.examples.SparkPi --conf spark.yarn.am.nodeLabelExpression=nodelabel_spark_pi_dag --conf spark.yarn.executor.nodeLabelExpression=nodelabel_spark_pi_dag --master yarn --num-executors 1 --driver-memory 512m --executor-memory 512m --executor-cores 1 /opt/spark/examples/jars/spark-examples_2.11-2.3.1.jar`

Additional:

- Whole `/etc/hadoop/conf/capacity-scheduler.xml` should look like as mentioned in this attachment file
  Note: Here there are three Nodelabels - `nodelabel_a,nodelabel_spark_pi_dag,nodelabel_b`

capacity-scheduler.xml

- Setting a Default Queue Node Label Expression , `nodelabel_a` will be used if any job triggered without specfing a Node Label

```
<property>
 <name>yarn.scheduler.capacity.root.default.default-node-label-expression</name>
 <value>nodelabel_a</value>
</property>
```

Reference: https://docs.cloudera.com/HDPDocuments/HDP2/HDP-2.5.5/bk_yarn-resource-management/content/configuring_node_labels.html
https://hadoop.apache.org/docs/r2.8.3/hadoop-yarn/hadoop-yarn-site/NodeLabel.html