## YARN NodeManager Setup

- Install bigtop utility
- Install node-manager
  `yum -y install hadoop-yarn-nodemanager.x86_64 hadoop-hdfs.x86_64 hadoop-mapreduce.x86_64`
- Modify `hdfs-site.xml , core-site.xml , yarn-site.xml` accordingly
- `sudo mkdir -p /data/1/hadoop-yarn/containers`
- `sudo mkdir -p /data/1/tmp/yarn/local`
- `sudo chown -R yarn:yarn /data/1/tmp/`
- `sudo chown -R yarn:yarn /data/1/hadoop-yarn/`
- Add all HDFS, YARN, Airflow, Omega VM's hostnames in `/etc/hosts` file
- mkdir `/opt/yarn-resource-change-script`
- **This file modifies**, `/etc/hadoop/conf/yarn-site.xml` , **so please modify the IP address mentioned in this file** `/opt/yarn-resource-change-script/nm.sh` , **modify this file accordingly with environment IP adress**
  `sudo vi /opt/yarn-resource-change-script/nm.sh`, ensure to modify the properties

  `yarn.nodemanager.resource.memory-mb`
  `yarn.nodemanager.resource.cpu-vcores`
  `yarn.application.classpath`
- Also be sure to set below mentioned properties according to environment

  `yarn.resourcemanager.ha.automatic-failover.zk-base-path`
  `yarn.resourcemanager.hostname.rm1, yarn.resourcemanager.hostname.rm2`
  `yarn.resourcemanager.cluster-id`
  `yarn.resourcemanager.webapp.address.rm1 , yarn.resourcemanager.webapp.address.rm2`
  `yarn.resourcemanager.webapp.https.address.rm1, yarn.resourcemanager.webapp.https.address.rm2`
  `yarn.resourcemanager.address.rm1, yarn.resourcemanager.address.rm2`
  `yarn.resourcemanager.admin.address.rm1, yarn.resourcemanager.admin.address.rm2`
  `yarn.resourcemanager.resource-tracker.address.rm1 , yarn.resourcemanager.resource-tracker.address.rm2`
  `yarn.resourcemanager.scheduler.address.rm1 , yarn.resourcemanager.scheduler.address.rm2`
  `yarn.resourcemanager.zk-address`
  `yarn.resourcemanager.zk-state-store.parent-path`
  `yarn.log.server.url`

```
#!/bin/bash
mem_kb=$(cat /proc/meminfo | awk NR==1 | awk  '{print $2}')
#echo $mem_kb

mem_mb=$((mem_kb/1024))
#echo $mem_mb

nm_mem_mb=$((mem_mb - 200))
#echo $nm_mem_mb

cpu_cores=$(grep -c ^processor /proc/cpuinfo)
nm_cpu_cores=$((cpu_cores * 3))
#echo $nm_cpu_cores
HADOOP_CONF_DIR='$HADOOP_CONF_DIR'
HADOOP_COMMON_HOME='$HADOOP_COMMON_HOME'
HADOOP_HDFS_HOME='$HADOOP_HDFS_HOME'
HADOOP_MAPRED_HOME='$HADOOP_MAPRED_HOME'
HADOOP_YARN_HOME='$HADOOP_YARN_HOME'

echo "<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.nodemanager.pmem-check-enabled</name>
        <value>false</value>
    </property>

    <property>
        <name>yarn.nodemanager.vmem-check-enabled</name>
        <value>false</value>
    </property>

    <property>
        <name>yarn.nodemanager.aux-services.mapreduce_shuffle.class<
/name>
        <value>org.apache.hadoop.mapred.ShuffleHandler</value>
    </property>

<property>
    <description>List of directories to store localized files in.<
/description>
    <name>yarn.nodemanager.local-dirs</name>
    <value>/data/1/tmp/yarn/local</value>
  </property>

  <property>
    <description>Where to store container logs.</description>
    <name>yarn.nodemanager.log-dirs</name>
    <value>/data/1/hadoop-yarn/containers</value>
  </property>

   <property>
         <name>yarn.nodemanager.address</name>
         <value>0.0.0.0:45454</value>
    </property>
   <property>
      <name>yarn.nodemanager.resource.memory-mb</name>
    <value>$nm_mem_mb</value>
  </property>

  <property>
```

```
     <name>yarn.nodemanager.resource.cpu-vcores</name>
     <value>$nm_cpu_cores</value>
  </property>

    <property>
        <name>yarn.resourcemanager.ha.automatic-failover.zk-base-path<
/name>
        <value>/yarn-prodrtrs-ha</value>
    </property>

    <property>
        <name>yarn.resourcemanager.ha.enabled</name>
        <value>true</value>
    </property>

    <property>
        <name>yarn.resourcemanager.ha.rm-ids</name>
        <value>rm1,rm2</value>
    </property>


   <property>
        <name>yarn.resourcemanager.hostname.rm1</name>
        <value>10.0.0.601</value>
    </property>

   <property>
        <name>yarn.resourcemanager.hostname.rm2</name>
        <value>10.0.0.701</value>
    </property>

    <property>
        <name>yarn.resourcemanager.cluster-id</name>
        <value>argoidmitronyarn</value>
    </property>

    <property>
        <name>yarn.resourcemanager.recovery.enabled</name>
        <value>true</value>
    </property>

    <property>
        <name>yarn.resourcemanager.state-store.max-completed-
applications</name>
        <value>200</value>
    </property>

    <property>
        <name>yarn.resourcemanager.store.class</name>
        <value>org.apache.hadoop.yarn.server.resourcemanager.recovery.
```

```
ZKRMStateStore</value>
    </property>

     <property>
        <name>yarn.resourcemanager.webapp.address.rm1</name>
        <value>10.0.0.601:8088</value>
    </property>
     <property>
        <name>yarn.resourcemanager.webapp.address.rm2</name>
        <value>10.0.0.701:8088</value>
    </property>


     <property>
        <name>yarn.resourcemanager.webapp.https.address.rm1</name>
        <value>10.0.0.601:8090</value>
    </property>
     <property>
        <name>yarn.resourcemanager.webapp.https.address.rm2</name>
        <value>10.0.0.701:8090</value>
    </property>


   <property>
        <name>yarn.resourcemanager.address.rm1</name>
        <value>10.0.0.601:8050</value>
    </property>

   <property>
        <name>yarn.resourcemanager.address.rm2</name>
        <value>10.0.0.701:8050</value>
    </property>

     <property>
        <name>yarn.resourcemanager.admin.address.rm1</name>
        <value>10.0.0.601:8141</value>
    </property>
     <property>
        <name>yarn.resourcemanager.admin.address.rm2</name>
        <value>10.0.0.701:8141</value>
    </property>

     <property>
        <name>yarn.resourcemanager.resource-tracker.address.rm1</name>
        <value>10.0.0.601:8025</value>
    </property>
     <property>
        <name>yarn.resourcemanager.resource-tracker.address.rm2</name>
        <value>10.0.0.701:8025</value>
    </property>
```

```
     <property>
        <name>yarn.resourcemanager.scheduler.address.rm1</name>
        <value>10.0.0.601:8030</value>
    </property>
     <property>
        <name>yarn.resourcemanager.scheduler.address.rm2</name>
        <value>10.0.0.701:8030</value>
    </property>


    <property>
        <name>yarn.resourcemanager.work-preserving-recovery.enabled<
/name>
        <value>true</value>
    </property>

    <property>
        <name>yarn.resourcemanager.work-preserving-recovery.
scheduling-wait-ms</name>
        <value>10000</value>
    </property>

    <property>
        <name>yarn.resourcemanager.zk-acl</name>
        <value>world:anyone:rwcda</value>
    </property>

    <property>
        <name>yarn.resourcemanager.zk-address</name>
        <value>10.0.0.601:2181,10.0.0.701:2181,10.0.0.801:2181</value>
    </property>

    <property>
        <name>yarn.resourcemanager.zk-state-store.parent-path</name>
        <value>/yarn-argoidmitron</value>
    </property>

    <property>
        <name>yarn.nodemanager.disk-health-checker.enable</name>
        <value>false</value>
    </property>

  <property>
    <description>Classpath for typical applications.</description>
     <name>yarn.application.classpath</name>
     <value>
        $HADOOP_CONF_DIR,
        $HADOOP_COMMON_HOME/*,$HADOOP_COMMON_HOME/lib/*,
```

```
        $HADOOP_HDFS_HOME/*,$HADOOP_HDFS_HOME/lib/*,
        $HADOOP_MAPRED_HOME/*,$HADOOP_MAPRED_HOME/lib/*,
        $HADOOP_YARN_HOME/*,$HADOOP_YARN_HOME/lib/*
      </value>
  </property>
  <property>
      <name>yarn.nodemanager.log.retain-seconds</name>
      <value>2000</value>
 </property>

<property>
  <description>Indicate to clients whether Timeline service is
enabled or not.
  If enabled, the TimelineClient library used by end-users will post
entities
  and events to the Timeline server.</description>
  <name>yarn.timeline-service.enabled</name>
  <value>true</value>
</property>

<property>
  <description>The setting that controls whether yarn system metrics
is
  published on the timeline server or not by RM.</description>
  <name>yarn.resourcemanager.system-metrics-publisher.enabled</name>
  <value>true</value>
</property>

<property>
  <description>Indicate to clients whether to query generic
application
  data from timeline history-service or not. If not enabled then
application
  data is queried only from Resource Manager.</description>
  <name>yarn.timeline-service.generic-application-history.enabled<
/name>
  <value>true</value>
</property>

<property>
 <name>yarn.timeline-service.webapp.address</name>
 <value>10.0.0.7:8188</value>
</property>

<property>
      <name>yarn.log-aggregation-enable</name>
      <value>true</value>
</property>
<property>
     <name>yarn.nodemanager.remote-app-log-dir</name>
```

```
      <value>/app-logs</value>
  </property>
  <property>
      <name>yarn.nodemanager.remote-app-log-dir-suffix</name>
      <value>logs</value>
  </property>

<property>
      <name>yarn.timeline-service.ttl-ms</name>
      <value>604800000</value>
  </property>

<property>
    <name>yarn.log.server.url</name>
    <value>http://10.0.0.7:19888/jobhistory/logs</value>
  </property>

</configuration>  "   > /etc/hadoop/conf/yarn-site.xml

sleep 2
sudo /etc/init.d/hadoop-yarn-nodemanager restart
sleep 10
```

- Create systemd file

  `sudo vi /etc/systemd/system/nm-modify.service`

```
[Unit]
Description=node-manager-modify
Wants=network-online.target
After=network-online.target

[Service]
User=root
Type=simple
ExecStart=/bin/bash /opt/yarn-resource-change-script/nm.sh

[Install]
WantedBy=multi-user.target
```

- `sudo systemctl start nm-modify`
- `sudo systemctl enable nm-modify`