**Core site Trash**

fs.trash.interval

fs.trash.checkpoint.interval

fs.s3.block.size 67108864

file.replication 1

AWS data pipeline

fs.s3a.awsAccessKeyId

fs.s3a.awsSecretAccessKey

**HDFS site**

dfs.replication by def 3

dfs.blocksize 128

dfs.heartbeat.interval 3sec

dfs.namenode.name.dir (/dfs/name) fsimage local dir NN

dfs.namenode.edits.dir JN

<name>dfs.ha.namenodes.hcluster</name> HA

<value>nn1,nn2</value>

dfs.datanode.data.dir (/dfs/data)

dfs.datanode.address

**Mapred site**

mapreduce.job.maps Default 2

mapreduce.job.reduces Default 1

yarn.app.mapreduce.am.scheduler.heartbeat.interval-ms 1000ms

You can define how much maximum memory each Map and Reduce task will take. Since each Map and each Reduce will run in a separate Container, these maximum memory settings should be at least equal to or more than the YARN minimum Container allocation.

For our example cluster, we have the minimum RAM for a Container (**yarn.scheduler.minimum-allocation-mb**) = 2 GB. We’ll thus assign 4 GB for Map task Containers, and 8 GB for Reduce tasks Containers.

In **mapred-site.xml**:  
 **<name>mapreduce.map.memory.mb</name>**  
 **<value>4096</value>**

**<name>mapreduce.reduce.memory.mb</name>**  
 **<value>8192</value>**

Yarn site

**So with our example cluster node with 12 disks and 12 cores, we will allow for 20 maximum Containers to be allocated to each node.**

**Each machine in our cluster has 48 GB of RAM. Some of this RAM should be reserved for Operating System usage. On each node, we’ll assign 40 GB RAM for YARN to use and keep 8 GB for the Operating System. The following property sets the maximum memory YARN can utilize on the node:**

**<name>yarn.nodemanager.resource.memory-mb</name>**

**<value>40960</value>**

The next step is to provide YARN guidance on how to break up the total resources available into Containers. You do this by specifying the minimum unit of RAM to allocate for a Container. We want to allow for a maximum of 20 Containers, and thus need (40 GB total RAM) / (20 # of Containers) = 2 GB minimum per container:

**<name>yarn.scheduler.minimum-allocation-mb</name> COntainer**  
 **<value>2048</value>**

YARN will allocate Containers with RAM amounts greater than the **yarn.scheduler.minimum-allocation-mb**.

yarn.resourcemanager.address 8032

yarn.resourcemanager.scheduler.class Schedular

(org.apache.hadoop.yarn.server.resourcemanager.scheduler.capacit

y.CapacityScheduler)

yarn.timeline-service.enabled

Capacity

<name>yarn.scheduler.capacity.root.queues</name>  
     <value>alpha,beta,default</value>

<name>yarn.scheduler.capacity.root.alpha.capacity</name>  
     <value>50</value>

<name>yarn.scheduler.capacity.root.acl\_submit\_applications</name>  
     <value>hadoop,yarn,mapred,hdfs</value>  
     <description>

Sqoop Import –connect driver with host and port then pusername and passwoed then src and dest

Sqoop permission

hdfs fs –chown –R sqoop:hdfs /user/root