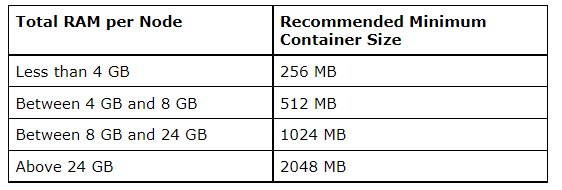
**mapred-site.xml:**

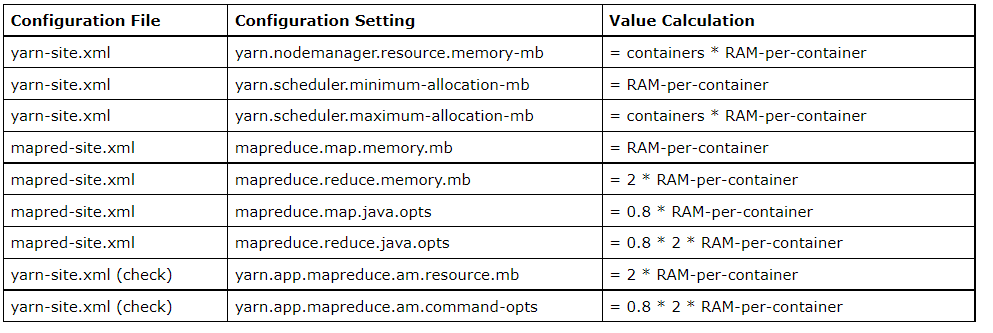
1. mapreduce.map.cpu.vcores = 1

More than one is used for large systems for limiting the CPU. We set it to 1 because we have a single node system. It does not affect the number of containers.

1. mapreduce.map.memory.mb = 4551



# of containers = min (2\*CORES, 1.8\*DISKS, (Total available RAM) / MIN\_CONTAINER\_SIZE) = 1,8

RAM-per-container = max (MIN\_CONTAINER\_SIZE, (Total Available RAM) / containers)) = 4551

1. mapreduce.reduce.cpu.vcores = 1

More than one is used for large systems for limiting the CPU. We set it to 1 because we have a single node system. It does not affect the number of containers.

1. mapreduce.reduce.memory.mb = 9102

**yarn-site.xml:**

1. yarn.nodemanager.resource.cpu-vcores = -1

Number of vcores that can be allocated for containers. This is used by the RM scheduler when allocating resources for containers. This is not used to limit the number of CPUs used by YARN containers. If it is set to -1 and yarn.nodemanager.resource.detect-hardware-capabilities is true, it is automatically determined from the hardware in case of Windows and Linux. In other cases, number of vcores is 8 by default.

1. yarn.nodemanager.resource.memory-mb = 8191

Based on the above table.

1. yarn.scheduler.minimum-allocation-vcores = 1

Requests lower than this will be set to the value of this property. Additionally, a node manager that is configured to have fewer virtual cores than this value will be shut down by the resource manager. Executes all of the requests.

1. yarn.scheduler.maximum-allocation-vcores = 6 (christos 3)

80% \* (total CPU cores) \* 2

<https://serverfault.com/questions/896783/how-to-determine-yarn-scheduler-maximum-allocation-vcores-value-in-ambari-cluste>

1. yarn.scheduler.minimum-allocation-mb = 512

Based on the above table

1. yarn.scheduler.maximum-allocation-mb = 8191

Based on the above table

**hadoop-env.sh:**

HDFS\_NAMENODE\_OPTS: 4bg

HDFS\_DATANODE\_OPTS: 1gb (we will have files with sizes 500mb+200mb+100mb+10mb=1gb, we have only 1 replication because we have only 1 node)

**yarn-env.sh:**

YARN\_RESOURCEMANAGER\_HEAPSIZE = 512mb

YARN\_NODEMANAGER\_HEAPSIZE: 1024mb

https://support.datafabric.hpe.com/s/article/Best-practice-for-setting-JAVA-heap-size-for-ResourceManager-and-NodeManager?language=en\_US