Yarn components

1. **AM(Application master)**
   1. Manages life cycle of applications running on the cluster.
   2. Runs application specific processes in containers allocated to it.
   3. Container allocation/de-allocation can take place in a dynamic fashion as the application progresses.
   4. Periodically sends heartbeats to resource manager to affirm its health and for record of its resource demands.
   5. Each instance of an application has a dedicated application master.
   6. When an application master receives a job, it initializes it by creating bookkeeping objects to keep track of its progress.
   7. Creates map tasks for each input split which was created by client and a number of reduce tasks object.
   8. Decides how to run jobs according to size of jobs.
   9. It also calls the job setup method.
2. **RM (Resource manager)**
   1. The master that co-ordinates allocation of computer resources on a cluster.
   2. It manages the use of resources across a cluster.
   3. Application master negotiates with the resource manager for cluster resources, which is number of containers.
   4. Each container has a certain memory limit.
   5. When a job is submitted to the resource manager it hands off the job request to its scheduler which allocates a container and then resource manager launches the application master’s process under the node manager’s management.
   6. If resource manager fails neither jobs nor containers can be launched. In this case administrator brings up a new resource manager instance and it recovers from saved state.
3. **NM(Node manager)**
   1. Node managers launch and monitor compute containers on machines in the cluster.
   2. Oversees containers that are running on cluster nodes so that application master does not use more resources than it has been allocated. .
   3. It manages the life-cycle of application containers, monitoring resource usage (memory, CPU) of individual containers.
   4. Kills any container that exceeds its allocated memory.
   5. It also monitors node health, and manages logs.
   6. Node manager keeps up to date with the resource manager by periodically sending heartbeats to resource manager which stop in case node manager fails.
4. **JHS(Job history server)**
   1. The new Resource Manager acting as a scheduler, does not track job status.
   2. This task is now delegated to the JobHistoryServer daemon, which is usually run on a separate node.
   3. Job history server daemon retains job history data and is an independent entity in YARN unlike in MapReduce.
   4. It archives job history information which can be used by users if desired for later interrogation.