

## YARN의 구성요소 각각에 대해..

### ResourceManager

클러스터에 1개 존재하며, 클러스터의 전반적인 자원 관리와 스케줄러와 Application Manager를 조정하는 역할을 한다.

클라이언트로부터 어플리케이션 실행 요청을 받으면, 그 실행을 책임질 Application Master를 실행한다.

클러스터 내의 Node Manager들과 통신을 해서 할당된 자원과 사용중인 자원의 상황을 알 수 있다.

### Scheduler

Node Manager들의 자원 상태를 관리하며 부족한 리소스를 할당한다.

자원 상태에 따라서 태스크들의 실행 여부를 허가해주는 역할인 스케줄링 작업만 담당한다.

Node Manager들의 자원 상태를 Resource Manager에게 통지한다.

### Application Manager

Node Manager에서 특정 작업을 위해 Application Manager를 실행하고, Application의 실행 상태를 관리하여 그 상태를 Resource Manager에게 통지한다.

### Node Manager

해당 node의  
노드 당 한개씩 존재하며, Container의 리소스 사용량을 모니터링하고, 관련 정보를 Resource Manager에 알리는 역할을 한다.

↑ RM의 scheduler에 알림.

## Application Master

YARN에서 실행되는 하나의 ~~태스크~~<sup>application</sup>를 관리하는 마스터 서버를 말한다. 어플리케이션 당 1개가 있다.

Scheduler로부터 적절한 Container를 할당 받고, 프로그램의 실행 상태를 모니터링, 관리한다.

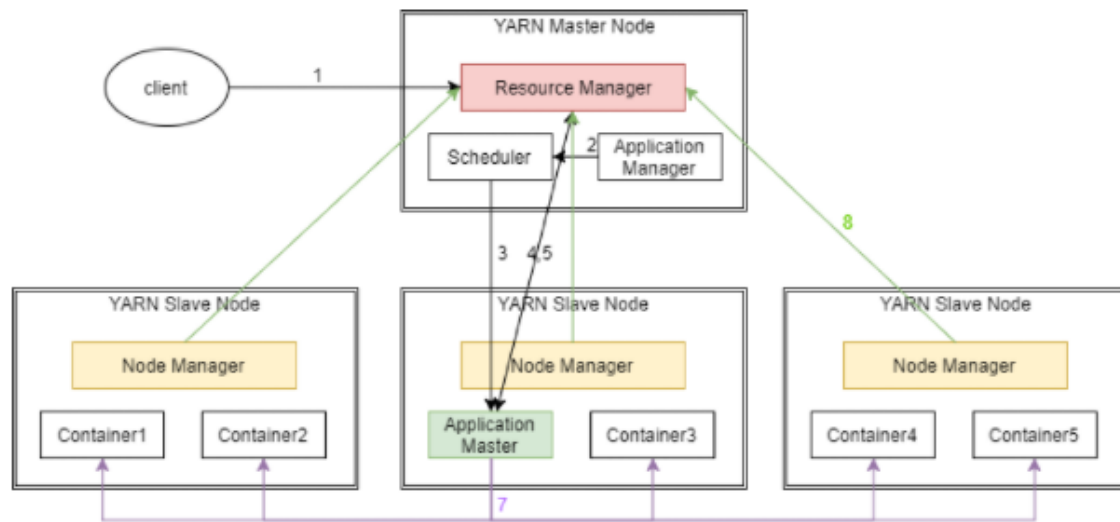
## Container

CPU, Disk, Memory 등의 리소스로 정의된다.

모든 작업은 결국 여러개의 태스크로 세분화되고, 각 태스크는 하나의 Container에서 실행된다.

필요한 자원의 요청은 Application Master가 담당하며, 승인 여부는 Resource Manager가 담당합니다. Container안에서 실행할 수 있는 프로그램은 자바프로그램뿐만 아니라, 커맨드 라인에서 실행할 수 있는 프로그램이라면 모두 가능하다.

## YARN의 대략적인 개요



1. YARN 클라이언트가 제출하는 Job/Application (MapReduce, java application 등등) 이 Resource Manager 에게 제출이 된다.

2. 마스터 노드의 Application Manager는 제출된 job의 유효성을 체크하고, 자원 할당을 위해 스케줄러에게 넘겨 준다.

3. 스케줄러는 임의의 slave node에 있는 컨테이너 중 하나를 Application Master로 할당한다.

4. Application Master는 slave nodes상의 데이터의 위치, cpu, memory등을 Resource Manager에 제공함으로써 Application Master 외의 container들을 조율한다.

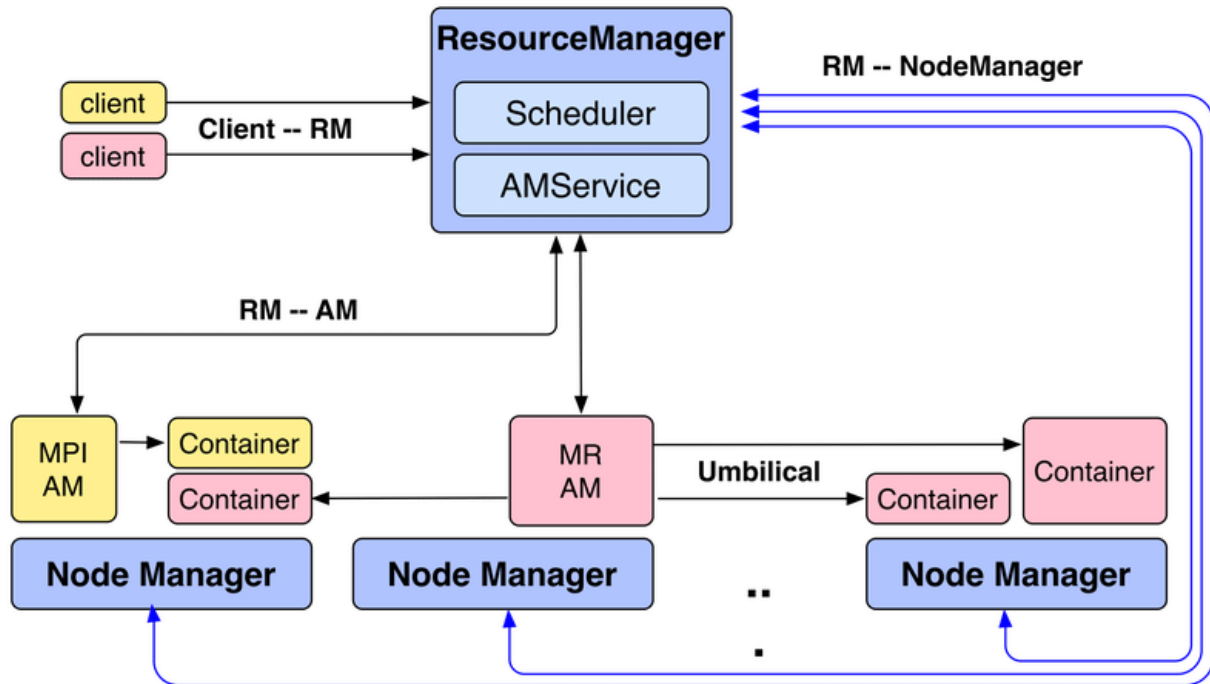
5. Resource Manager는 최상의 리소스들을 할당하고, Application master에 node의 디테일한 정보들을 전달한다.

6. Application master는 container들을 실행시키기 위해 제안된 slave node상의 Node manager에게 요청을 보낸다.

7. Application master는 잡이 실행되는 동안, 요청된 container들의 리소스를 관리한다. 그리고 실행이 완료되면, Resource manager에게 알려준다.

8. Node manager는 스케줄러가 새로운 application을 시작할 수 있도록 하기 위해, 주기적으로 자신의 노드에서 현재 이용가능한 리소스들의 상태를 Resource manager에게 알려준다.

9. Slave node에서 이상이 생긴 경우, Resource manager는 Application master가 프로세스를 완료할 수 있도록 새로운 container를 할당한다.



나는 총 세 대의 서버를 갖고 있고, 이 서버 위에 클러스터를 구축하여 빅데이터를 처리하는 application 을 실행한다고 해보자.  
세 대의 서버 위에 구축된 클러스터에 Hadoop YARN 이 올라간다.

① YARN 은 **Resource Manager**(이하 **RM**) 를 실행시킨다.  
RM 은 클러스터 당 하나만 있다.

② 또한 YARN 은 클러스터를 구성하고 있는 각 노드에서 **Node Manager**를 실행시킨다.  
**Node Manager**(이하 **NM**)는 해당 노드의 Resource 를 관리하고 있다.  
만약 RM 이 "3번 노드의 Resource 는 얼마나 있지?" 라고 3번 노드에게 물으면,  
3번 노드의 NM 이 "cpu 85% 높고 있고 여유 memory 3g 있음" 이렇게 대답 할 것이다.  
따라서 **RM**은 **NM**을 통해 전체 클러스터의 Resource 를 알고 관리할 수 있다.

NM 라 RM이  
서로 쿼리 받는  
내용.

Client(이하 Cli) 인 우리는 RM 에게 "나 MapReduce 하나 돌리고 싶다." 라고 말한다.  
(이 작업은 Cli가 RM에게 Container Launched Context 를 보냄으로써 이루어진다고 한다.)

