

HDFS에서는 어떤 아키텍처를 가질까?

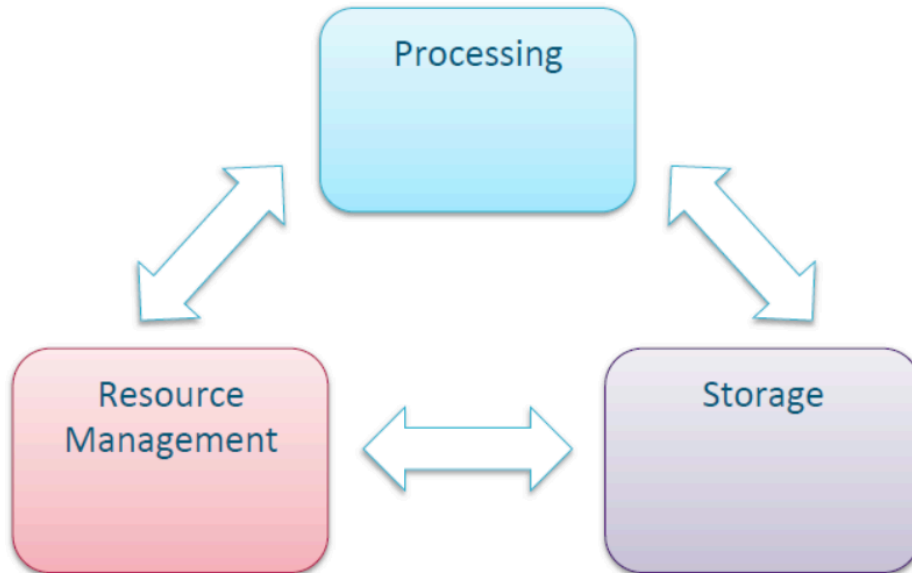
- OS에 dependent하지 않음
- linux를 활용하는 경우가 대부분
- 맨 밑단에서 HDFS가 파일 관리, 중간단계에서 yarn이 리소스가 어떤 노드에 활용가능한지, 최상단에서는 application

Hadoop cluster terminology

- Cluster
 - group of computers working together
- Node
 - an individual computer in the cluster (data 중복 저장, system availability 증가) 분산해서 여러 군데 나눠서 저장한다는 것 뿐만 아니라 (performance, 중복해서 저장함으로써 안정성을 높임)
 - network 로 연결되어있음
- daemon
 - os가 등록된 daemon을 실행시킴, node에서 돌아가는 program
- Master node
 - Worker node를 담당하는 hub

Cluster components

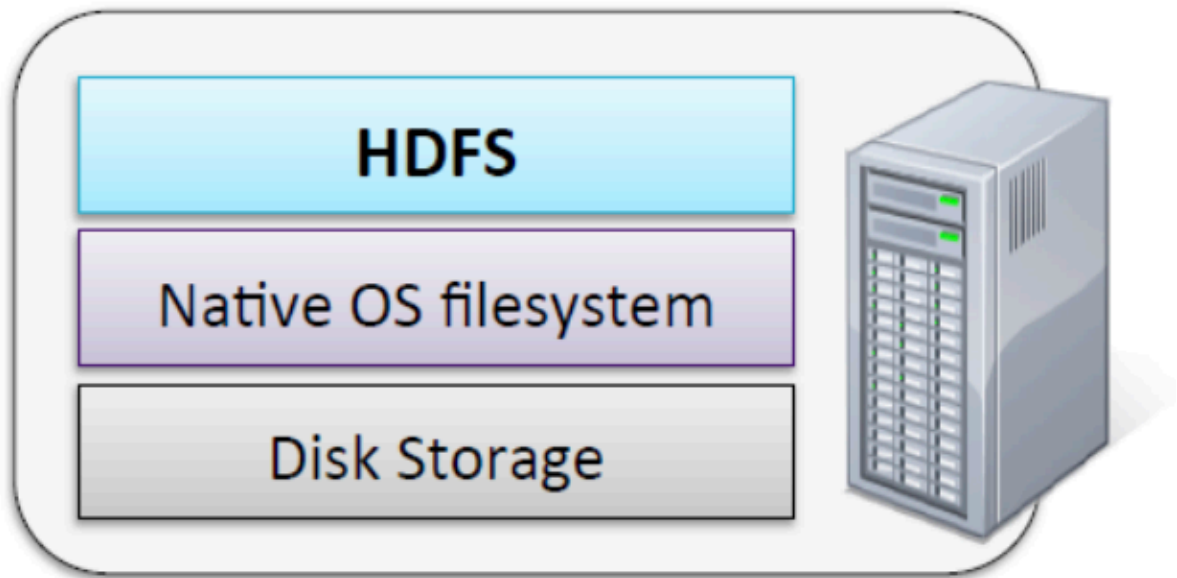
- HDFS



- processing : master node (name node가 하기도함)
- resource manage: yarn

HDFS

- 자바로 구성된 file system
- GFS를 open source화
- 파일시스템이라고 생각할 수 있도록 만들어진 application
- 대용량의 data를 잘 분산해서 관리하는 역할

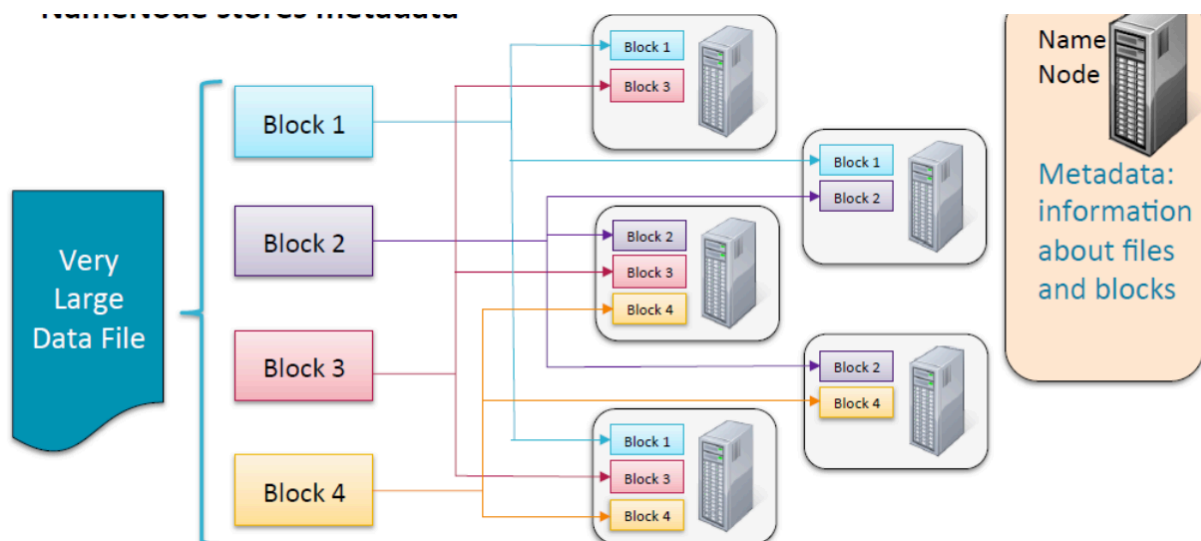


HDFS 특징

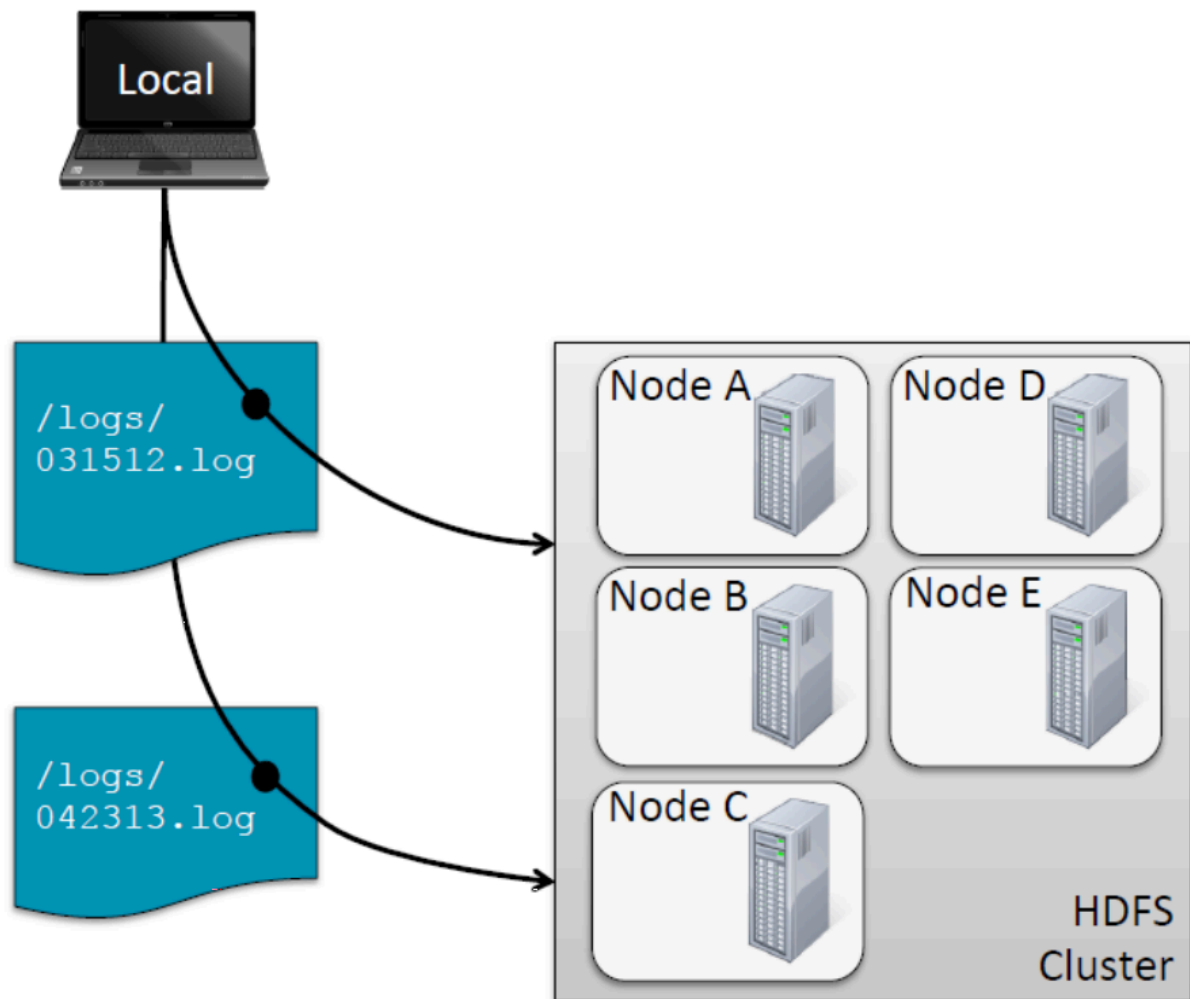
- sequential access
 - sequential 하게 처리하는게 굉장히 performance가 좋음
 - 디스크가 한바퀴 돌아서 찾을 효율 파일크기가 typically 100MB
 - streaming read
- write once
 - 파일에 한번 씌여지고 여러번 읽히는 (write는 read에 비해 오랜시간이 걸림)
- read 연산은 찾고싶은것만 읽는 것이 아니라 전체 data를 다 읽어야할 수도 있음(streaming read)
 - rather than random reads

How files are stored

- Data file을 128 mb block 으로 나누어 저장됨
- each block은 3개정도의 replication으로 저장됨
- data에 대한 정보 (meta data - data 저장된 위치)는 name node에 저장

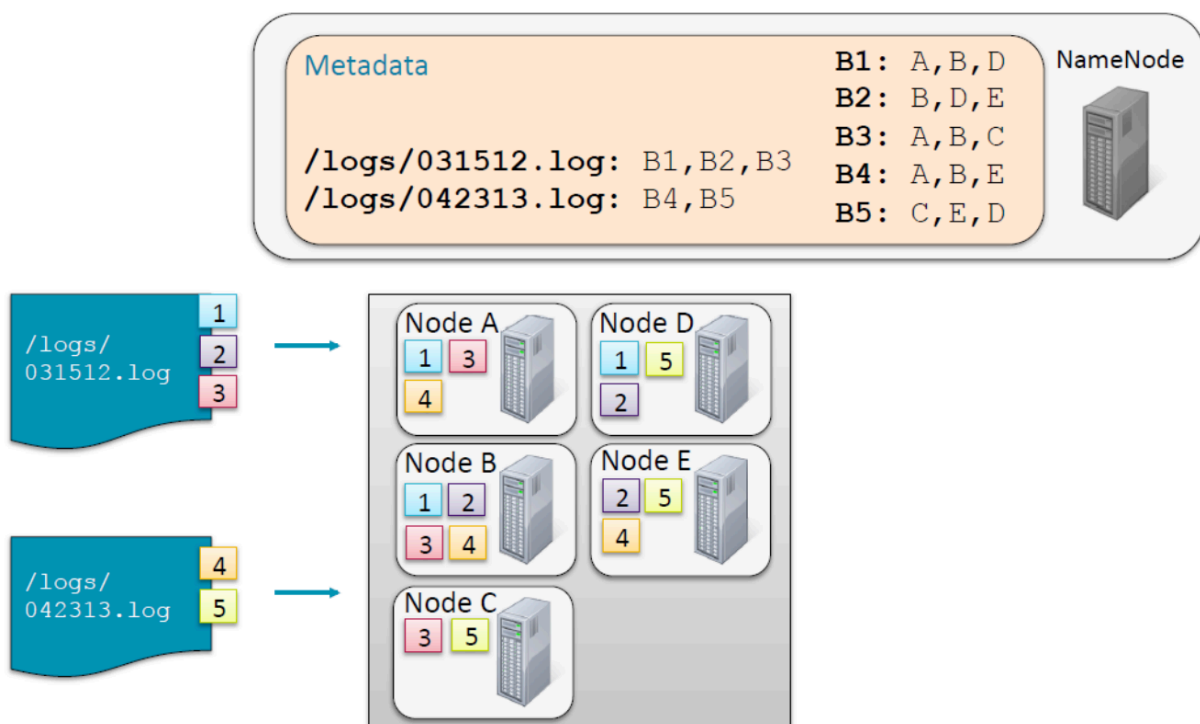


1. storing and retrieving files



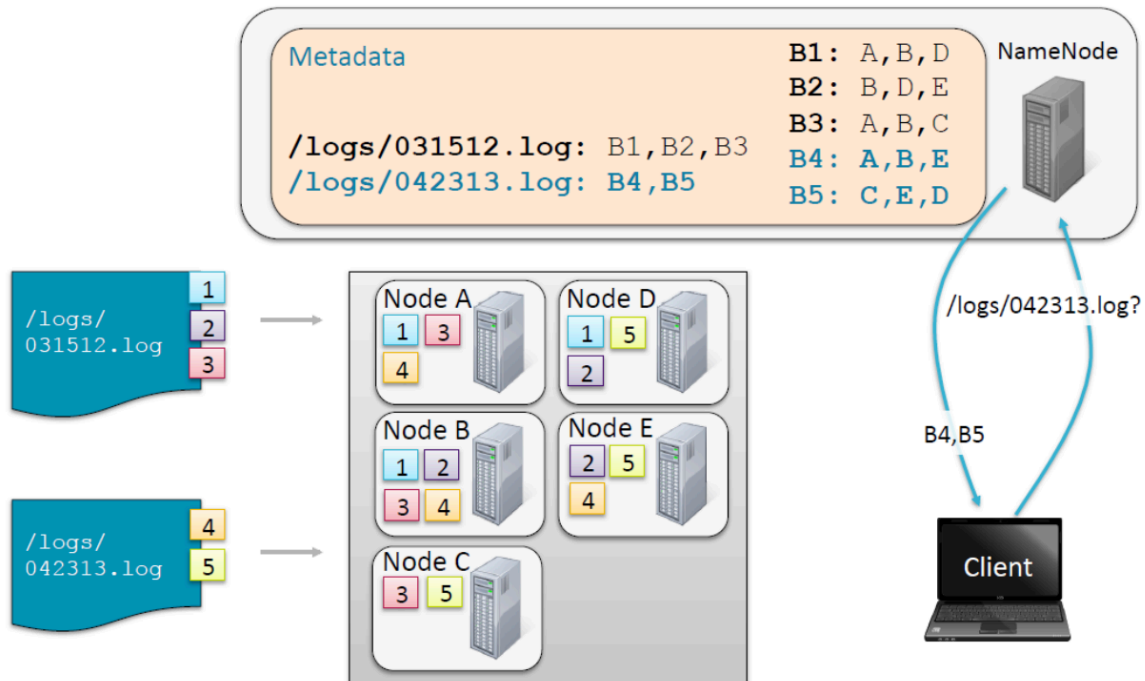
2. block으로 나뉘지고 meta data는 name node에 저장

- file이 어떤 block에 매핑되어있고, block은 어디에 있는지

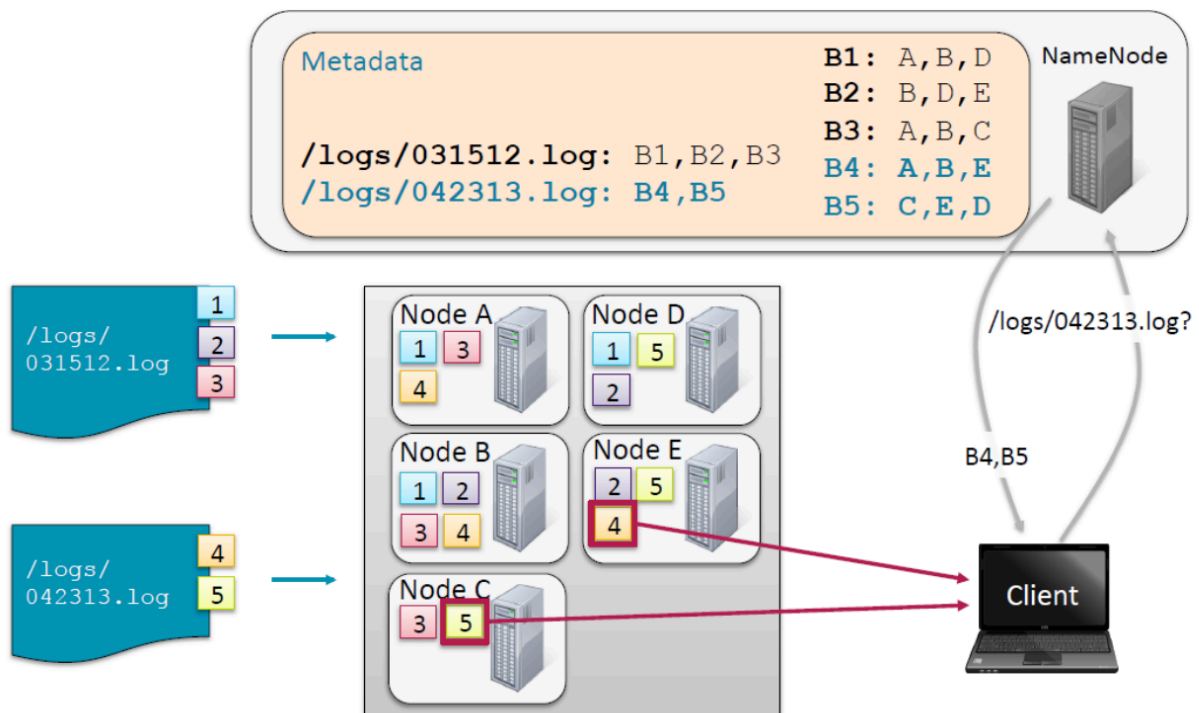


3. retrieving files

- 어떤 노드에 접근하면 파일을 얻을 수 있는가?



- name node는 master node 역할을 하기도하고 master node를 따로 두기도함
- data가 저장되어있는 node는 worker node라고 볼 수 있음



HDFS NameNode Availability

- name node daemon은 어떤 이유로도 중단이 되면 안된다. (name node의 명제)
- data node에서는 block을 replication 함으로써 availability를 높였지만
- name node는 어떻게 높일 것인가? (typical한 가이드라인)
 - 1.HDFS is typically set up for high availability (name node 두대)

2.실제로는 하나의 node에만 할당을 하고 언제든지 역할을 대체할 수 있는 name node 대체 노드를 하나 지정해둠 (secondary name node)

- 최 극단의 상황 두가지
 - 굉장히 많은 수의 clone을 만들어두면됨
 - 아예 한대만 할당을 해서 고쳐서 쓰면 됨 (좀 시간이 걸리더라도)

Options for Accessing HDFS

■ From the command line

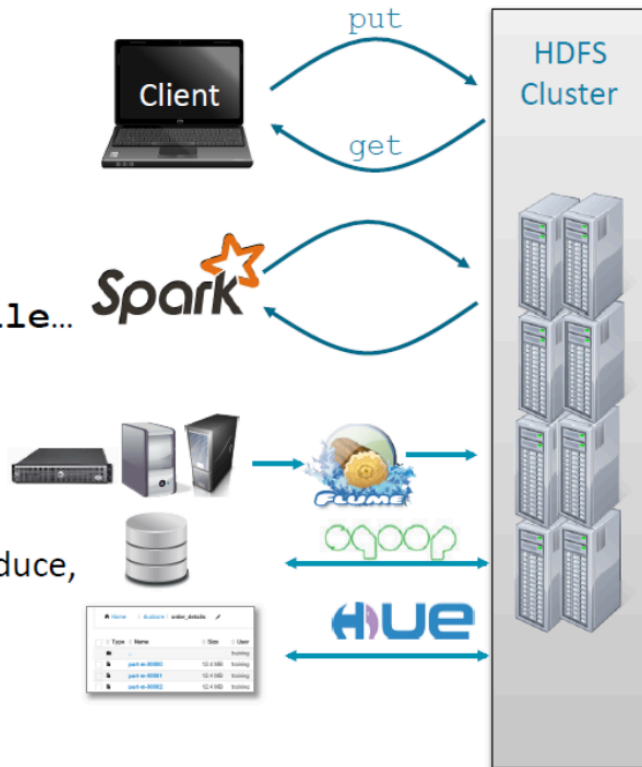
- FsShell:
- `$ hdfs dfs`

■ In Spark

- By URI, e.g.
- `hdfs://nnhost:port/file...`

■ Other programs

- Java API
- Used by Hadoop MapReduce, Impala, Hue, Sqoop, Flume, etc.
- RESTful interface



yarn

- Yet Another Resource Negotiator
- resource manager, job scheduler (분배해주고 스케줄링해주고)
- allows multiple data processing on single hadoop cluster
 - Batch programs(spark, map reduce)
 - Interactive SQL
 - Advanced analytics(spark, impala)
 - streaming (spark streaming)

yarm deamons

- Resource Manager (RM)
 - runs on master node (전체 노드에 있는 리소스들을 최종적으로 모니터링하고 application과 실제 노드들과의 communication 담당)
 - application이 실제 사용해야될 리소스들이 어디에있는지 매핑
- Node manager
 - 노드별로 가지고 있는 리소스에 대한 정보를 리소스 매니저에게 넘겨주는 역할
 - 각각 노드에서 동작 (runs on slave node)

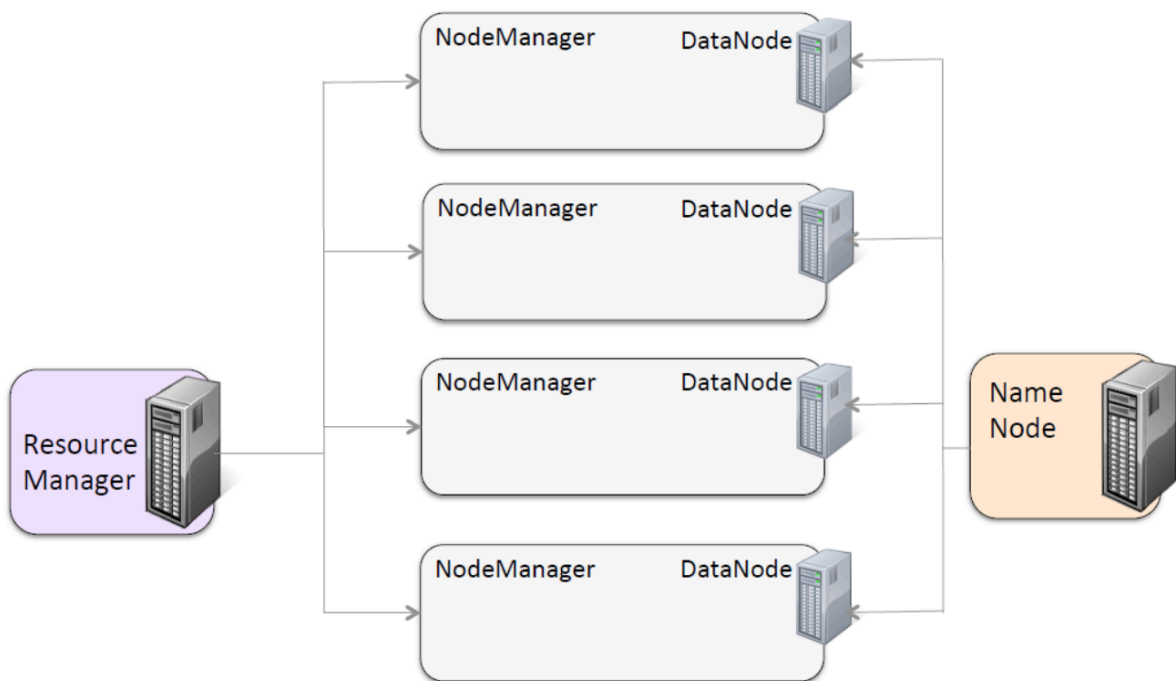
Running and Application in yarn

- containers
 - created by resource manager upon request (사용자가 요청을 보낼때)
 - 실제로는 application이 여러개의 컨테이너에서 동작할 수도 있음 (리소스가 여러군데 나눠져있을 경우)
 - Application이 동작하는 환경을 담당해줌
- Application master
 - application당 하나의 동작을 관리해주는 주체
 - containers에서 application의 요청을 실제 동작될 수 있도록 넘겨주고 처리된 output을 받아서 response (application의 시작과 끝을 관리해줌)

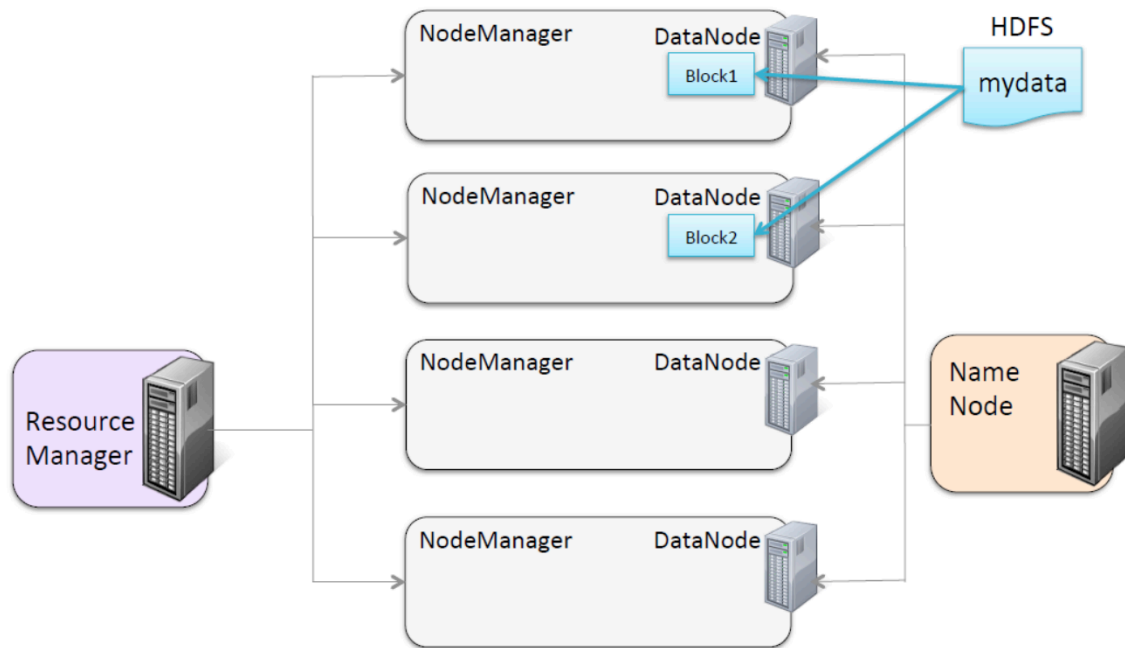
How to run application on Yarn

1. node manager와 data node 매핑

- 하나의 pc에 resource manager가 각각 있을 수 있음

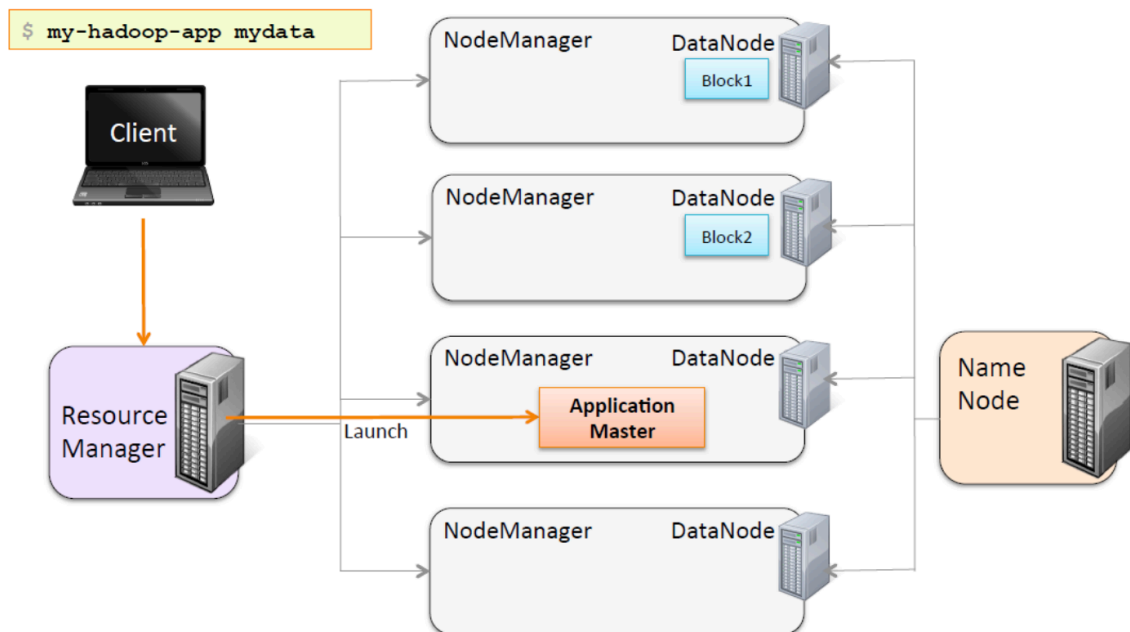


2. data 를 block으로 나뉘어서 storing



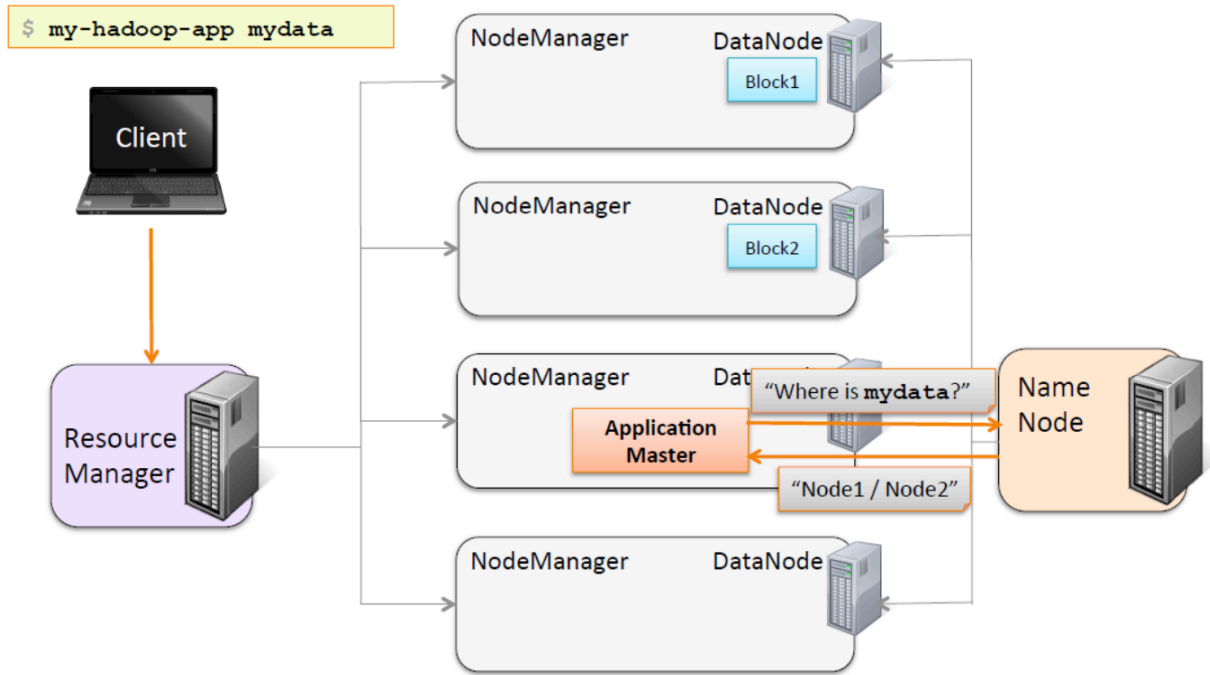
3. client가 application에서 data를 사용하도록 요청

- 3번째 node manager에 리소스가 적어보여서 리소스 분배

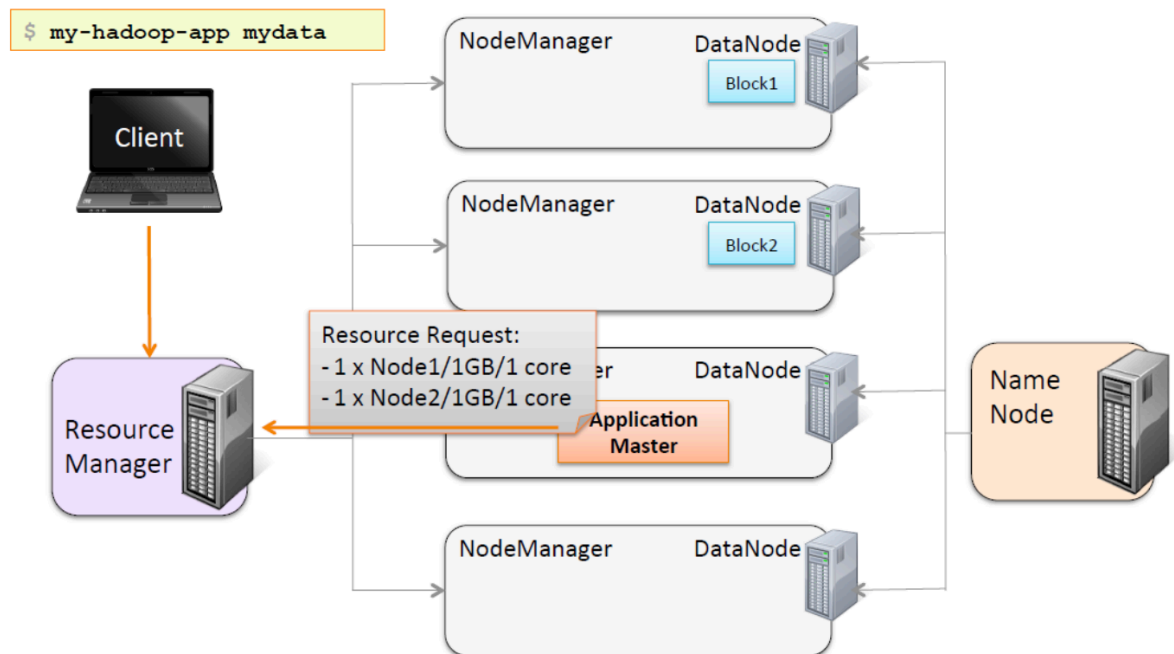


4. Application Master query to Name node

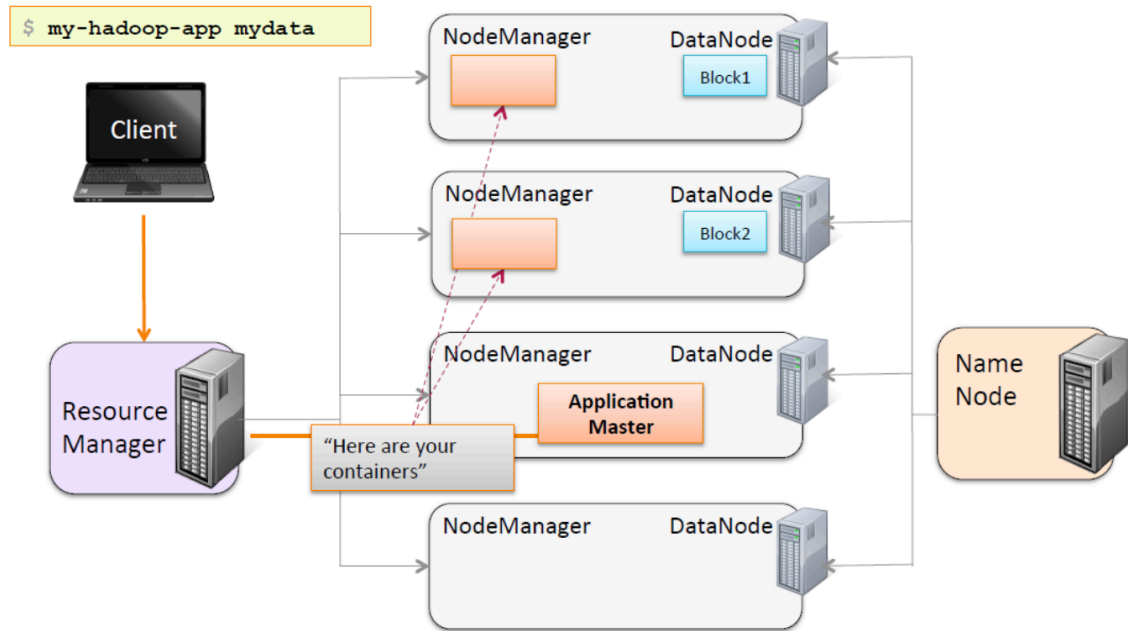
- 직접 처리하진 않고 전체적인 처리과정을 관장함 (정보취합)



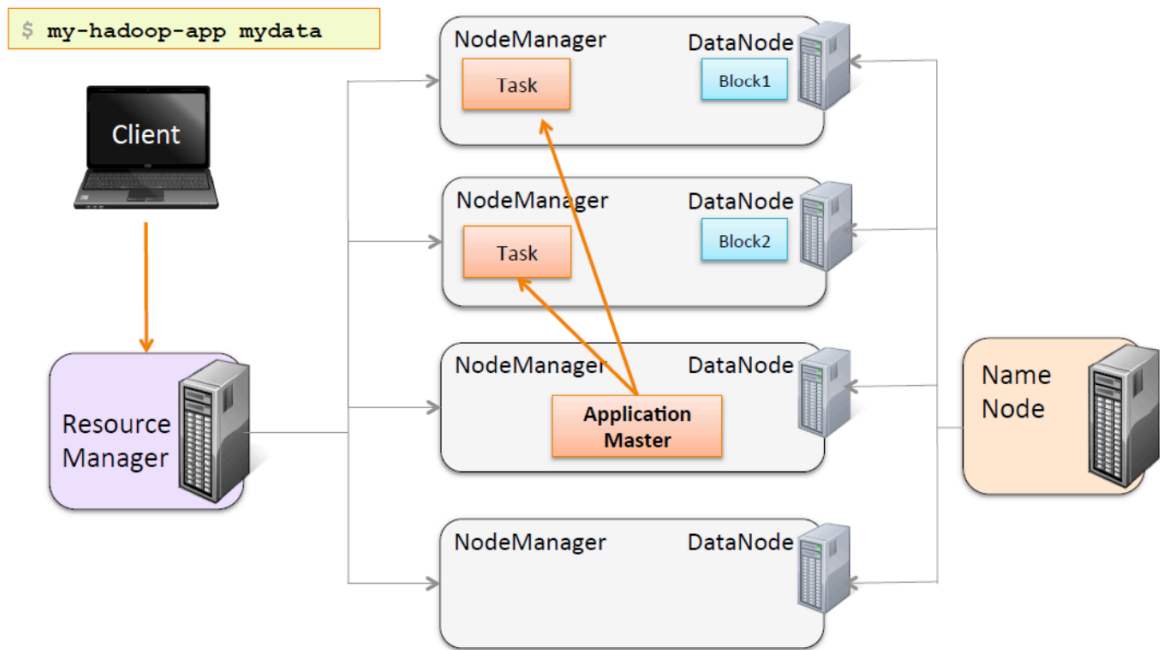
5. response(필요한 정보를) resource manager에게 넘겨줌



6. data가 저장되어있는 node 1, 2에 처리할 수 있도록 container 생성

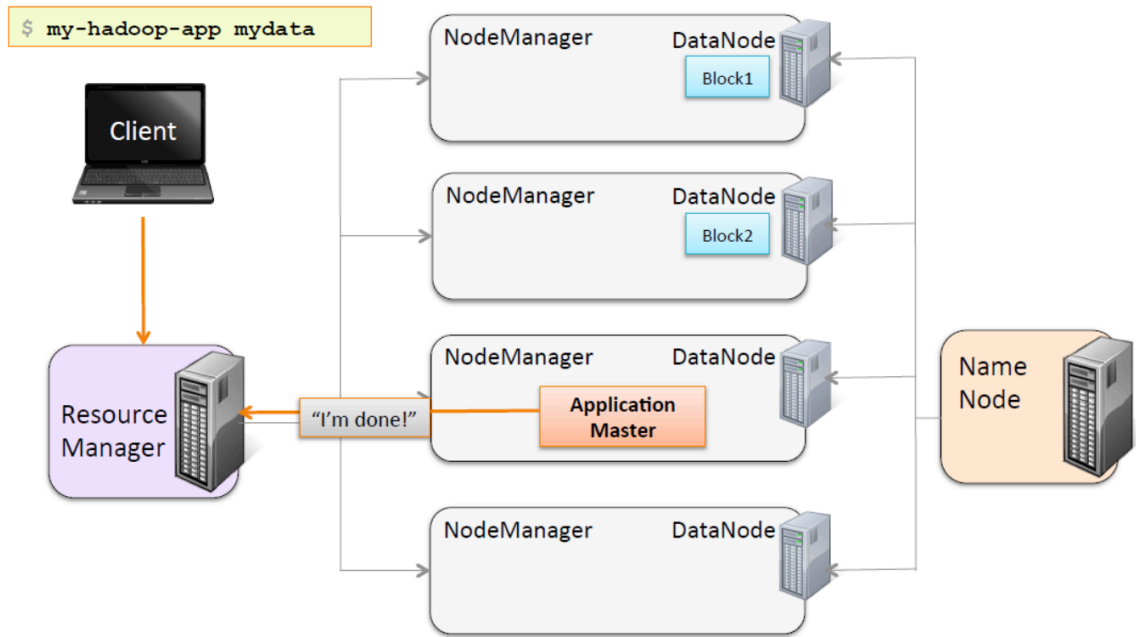


7. Application manager가 처리과정담당



8. Application manager가 resource manager에 response전달 후 최종적으로 client에게 전달

- 사용자에게 대한 관리는 resource manager가 관리함
- application manager는 application에 대한 관리만 총괄



- 이러한 복잡한 과정을 yarn이 함으로써 사용자가 일일이 관여할 필요 없고 resource manager와만 통신하면 됨
- 하둡은 개발자를 위한 3개의 major yarn tool을 가지고 있음
 - The Hue job browser
 - The yarn web ui
 - The yarn command line
- Cloudera manager를 통해서도 리소스를 관리할 수 있다.