

The background features a series of overlapping, wavy, ribbon-like shapes in various shades of green and white, creating a dynamic, flowing effect. A solid dark green horizontal bar is positioned at the bottom of the slide.

# **HDFS**

## **(Hadoop Distributed File System)**

# HDFS(Hadoop Distributed File System)

- 수십 테라바이트 또는 페타바이트 이상의 대용량 파일을 분산된 서버에 저장하고, 많은 클라이언트가 저장된 데이터를 빠르게 처리할 수 있게 설계된 파일 시스템
- 저사양 서버를 이용해 스토리지를 구성 가능
- 주로 대규모 데이터 저장 및 배치 처리에 적합
  - DBMS와 같이 고성능, 고가용성이 필요한 경우와 전자상거래와 같이 트랜잭션이 중요한 경우에는 부적합

대용량 파일 시스템	특징
DAS (Direct-Attached Storage)	<ul style="list-style-type: none"><li>• 서버에 직접 연결된 스토리지</li><li>• 여러 개의 하드디스크를 장착할 수 있는 외장형 하드디스크</li></ul>
NAS (Network-Attached Storage)	<ul style="list-style-type: none"><li>• 일종의 파일 서버</li><li>• 파일 시스템을 안정적으로 공유할 수 있음</li><li>• 별도의 운영체제 사용</li></ul>
SAN (Storage Area Network)	<ul style="list-style-type: none"><li>• 수십-수백대의 SAN스토리지를 데이터 서버에 연결해 총괄적으로 관리해주는 네트워크</li><li>• DBMS와 같이 안정적으로 빠른 접근이 필요한 데이터를 저장하는데 사용(고성능, 고가용성)</li></ul>

# HDFS 설계 목표

- 장애 복구
  - 디스크 오류로 인한 데이터 저장 실패 및 유실 등의 장애를 빠르게 감지하고 대처
  - 데이터를 저장할 때 복제 데이터도 함께 저장해서 데이터 유실 방지
  - 분산 서버 간 주기적인 상태 체크
- 스트리밍 방식의 데이터 접근
  - HDFS에 파일 저장 및 조회를 위해 스트리밍 방식으로 데이터에 접근
  - 배치 작업과 높은 데이터 처리량을 위해 스트리밍 방식을 사용

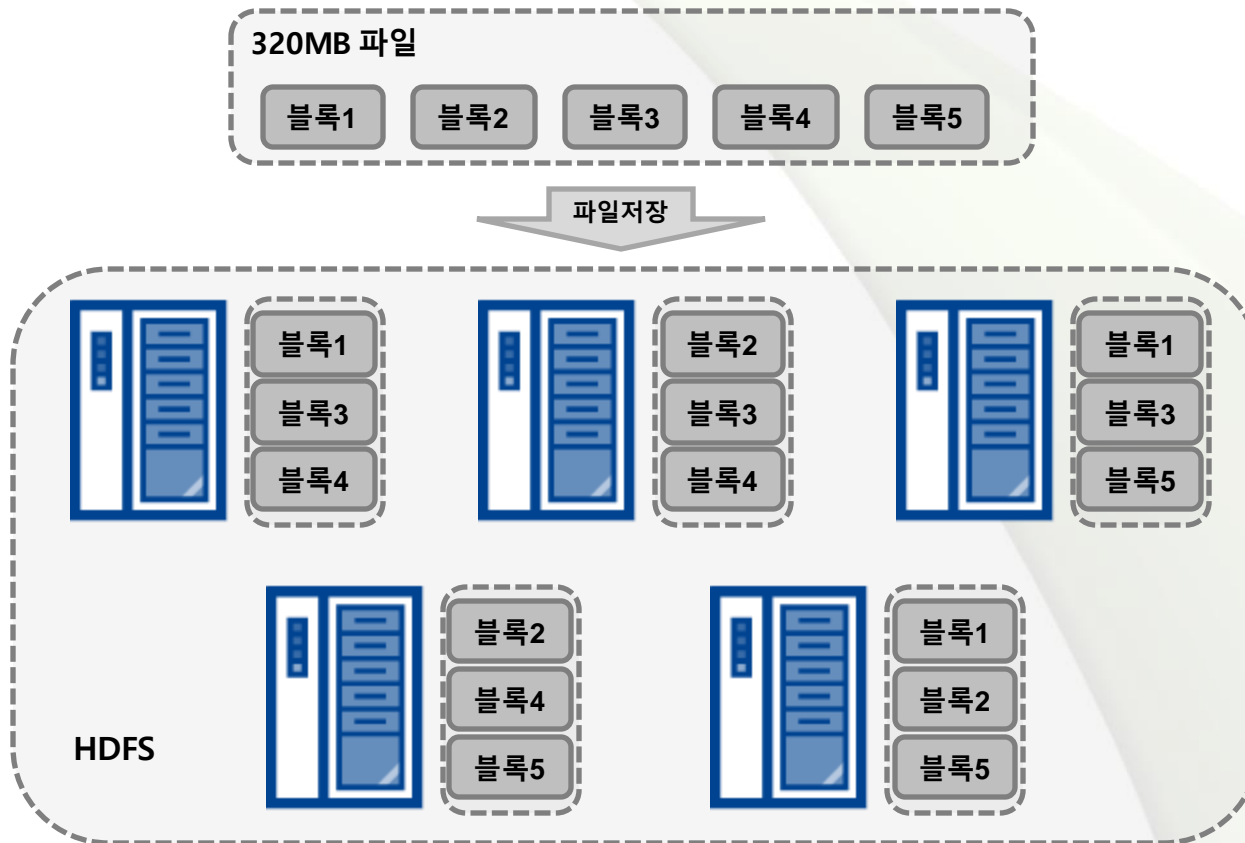
# HDFS 설계 목표

- 대용량 데이터 저장
  - 하나의 파일이 기가바이트에서 테라바이트 이상의 사이즈로 저장될 수 있도록 설계
  - 높은 데이터 전송 대역폭, 하나의 클러스터에서 수백 대의 노드를 지원
  - 하나의 인스턴스에서 수백만 개 이상의 파일을 지원
- 데이터 무결성
  - 한번 저장한 데이터를 수정할 수 없음 (읽기만 가능)
  - 파일 이동, 삭제, 복사할 수 있는 인터페이스 제공

# HDFS 아키텍처

## ■ 블록 구조 파일 시스템

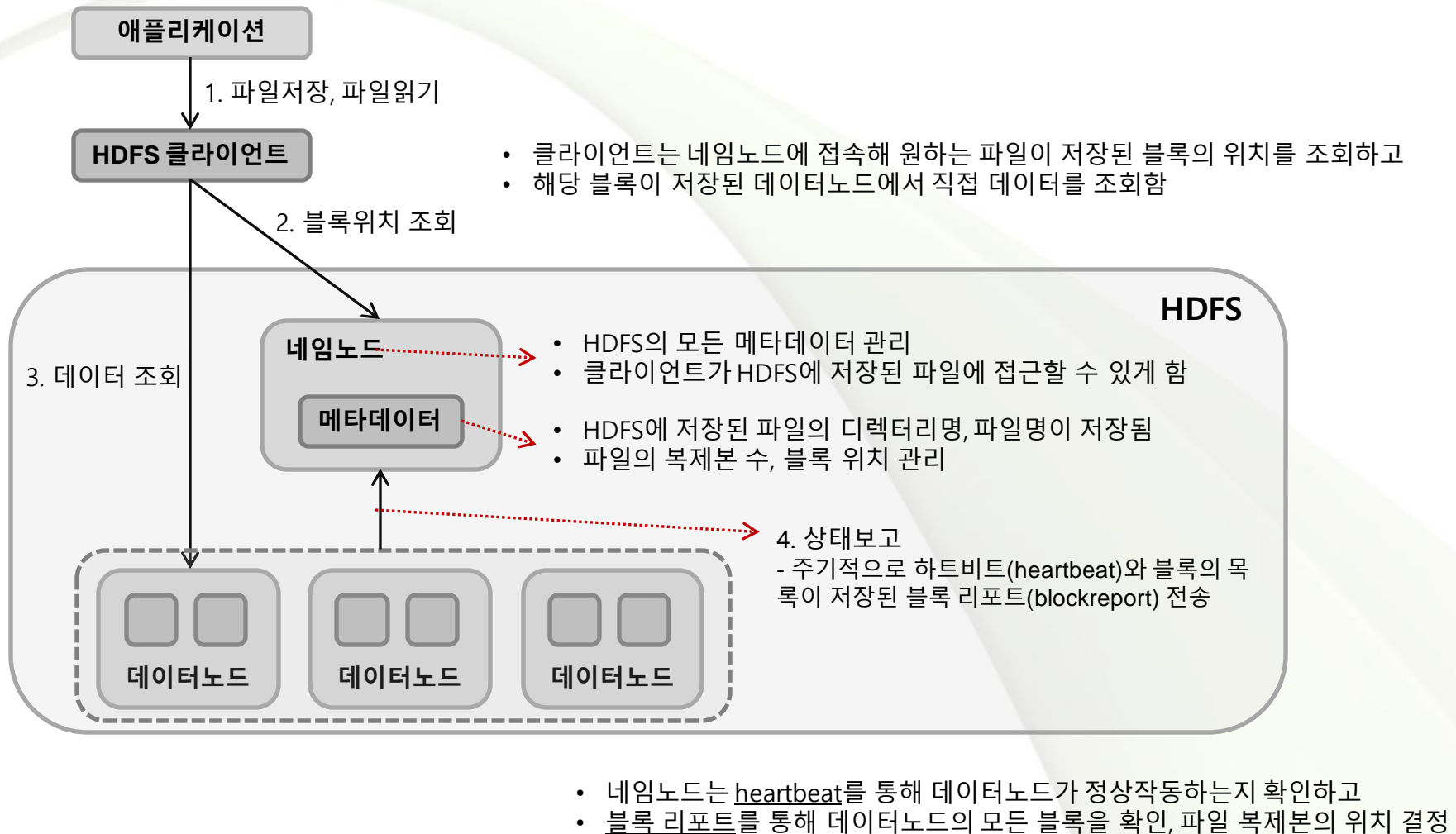
- 기본 블록 사이즈는 128MB로 설정 (설정파일에서 수정 가능)
- 블록을 저장할 때 3개씩 블록의 복제본을 저장(설정파일에서 수정 가능)



# 네임 노드와 데이터 노드

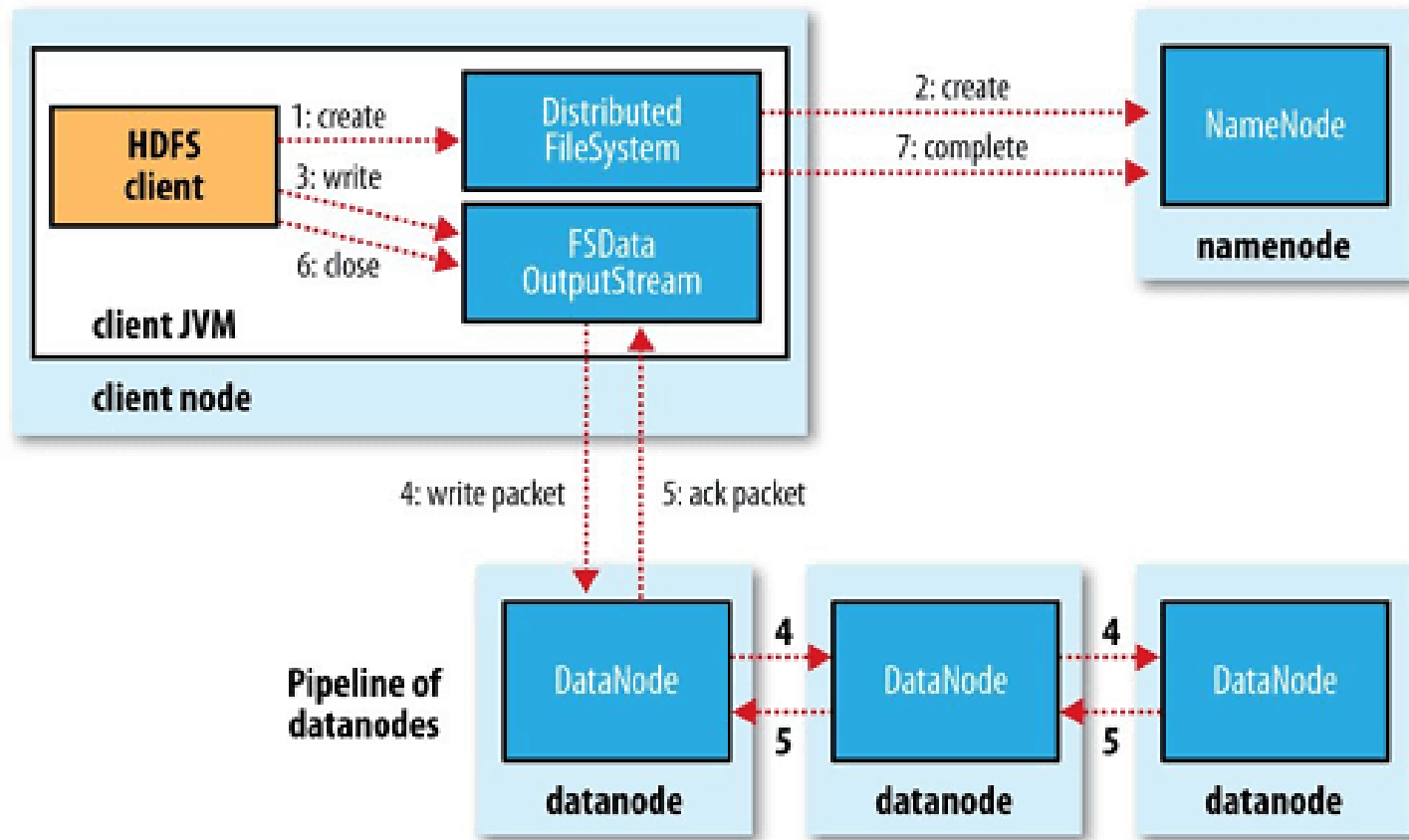
- HDFS는 마스터-슬레이브 아키텍처
  - 네임노드가 마스터 역할, 데이터노드가 슬레이브 역할
- 네임 노드
  - 메타데이터 관리
    - 파일 시스템 유지를 위해 파일명, 디렉터리, 크기, 권한 등의 정보 및 파일에 대한 블록 매핑 정보 관리
    - 빠른 응답을 위해 전체 데이터를 메모리에 로딩해서 관리
  - 데이터 노드 모니터링
    - 3초마다 heartbeat 메시지를 통해 데이터 노드의 실행 상태와 용량 관리
  - 블록 관리
    - 용량이 부족한 데이터 노드 또는 장애가 발생한 데이터 노드 등이 발견되면 데이터를 새로운 데이터 노드로 복제하는 등의 방식으로 블록 관리
  - 클라이언트 요청 접수
    - 클라이언트가 HDFS에 접근하려면 반드시 네임 노드에 접속해서 사용
- 데이터 노드 → 클라이언트가 HDFS에 저장하는 파일을 로컬 디스크에 유지

# HDFS 동작 구조



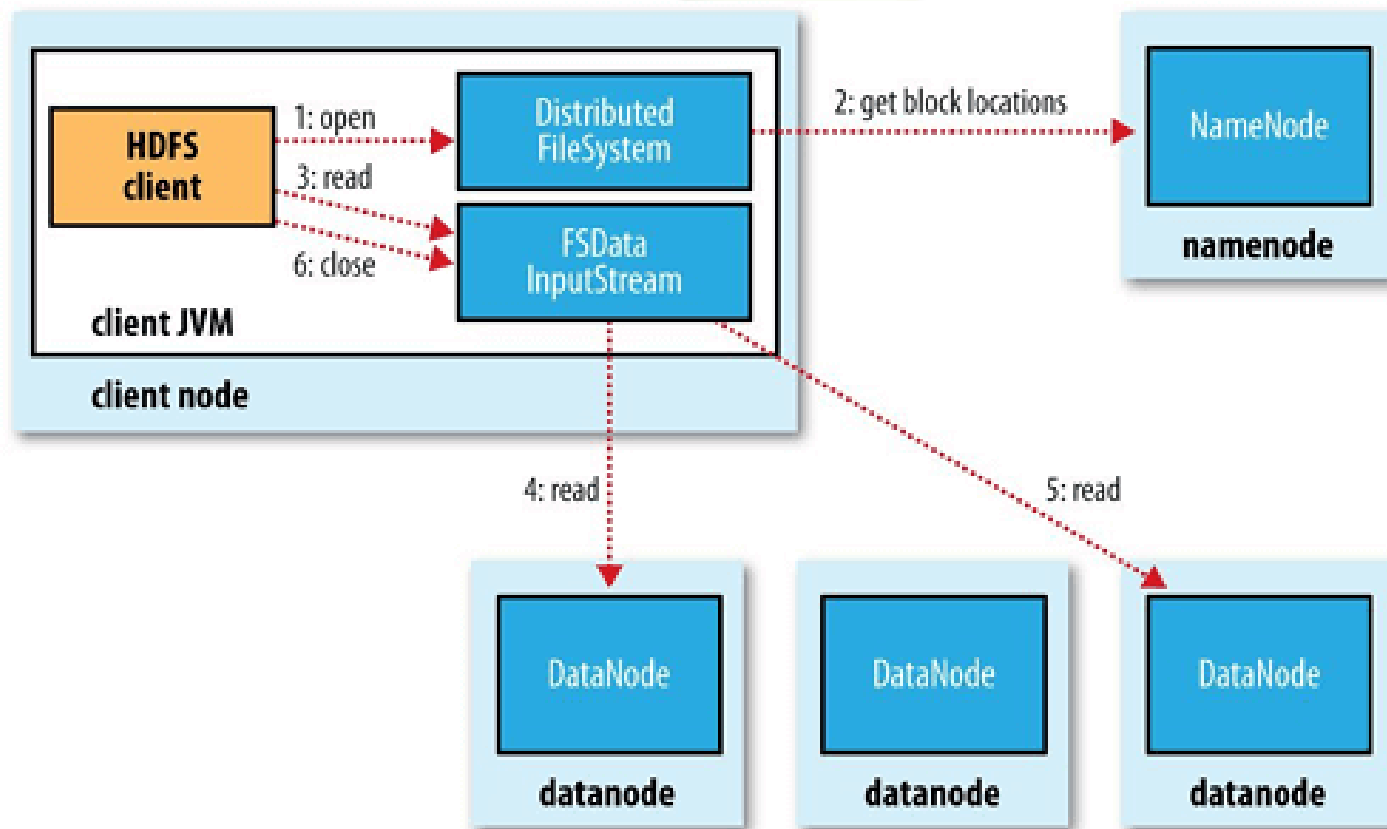
# 파일 쓰기

- 클라이언트는 DFSOutputStream 객체를 사용해서 HDFS의 네임노드 및 실제 데이터를 저장할 데이터노드와 통신



# 파일 읽기

- 클라이언트는 DFSInputStream 객체를 사용해서 HDFS의 네임노드 와 통신
- 실제 데이터를 읽을 데이터노드와 통신하기 위해 BlockReaderLocal, RemoteBlockReader 객체 사용

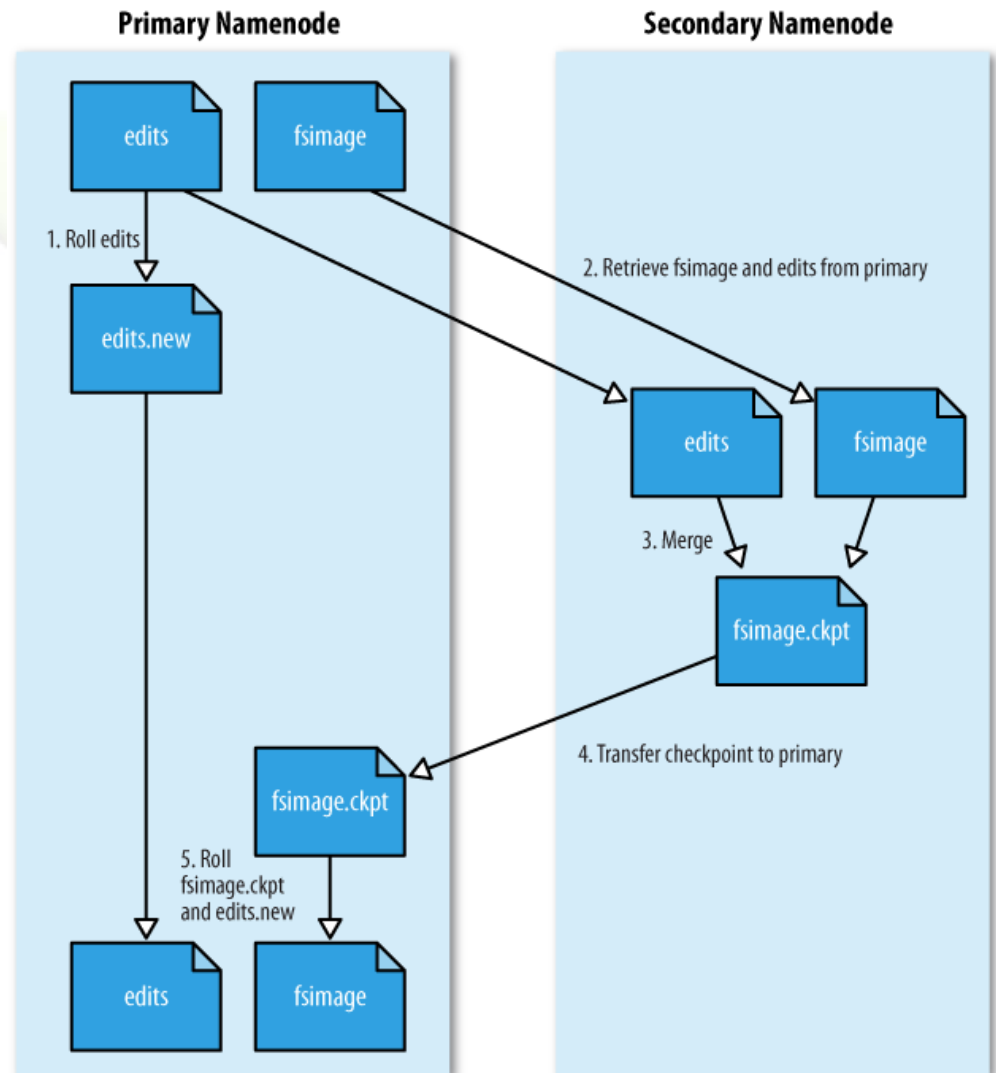


# 보조 네임노드

- 네임노드는 메모리에서 메타데이터 처리
  - 재부팅할 경우 메타데이터 유실
- 메타데이터를 안전하게 유지하기 위해 `editslog` 파일과 `fsimage` 파일 사용
  - `editslog` : 변경 내역 로그
  - `fsimage` : 특정 시점의 메모리 스냅샷
- 네임노드 구동 과정
  - `fsimage` 메모리에 로딩
  - `edits` 파일의 내용을 읽어서 메모리에 변경 내역 반영
  - `edits` 파일 초기화
  - 데이터 노드가 전송한 블록 리포트를 메모리에 적용
  - 서비스 시작

# 보조 네임노드

- edits는 시스템 운영중에 무한대로 커질 수 있음
- 네임노드의 구동 과정에서 edits 파일의 내용이 많을 경우 성능 저하
- 이러한 문제 해결을 위해 보조 네임노드 사용
  - 주기적으로 네임노드의 fs이미지를 갱신하고 edits 파일 초기화



# HDFS 명령어

- 하둡은 사용자가 HDFS를 쉽게 제어할 수 있도록 셸 명령어 제공
- 형식
  - `$HADOOP_HOME/bin/hdfs dfs -command [args]`
- 명령어의 대부분은 유닉스 계열에서 사용하는 명령어와 유사한 방식으로 사용

# HDFS 명령어

명령어	설명
ls	지정한 디렉터리 하위의 파일 및 디렉터리 목록 출력
du	지정한 디렉터리 또는 파일의 사용량을 바이트 단위로 출력
cat	파일 내용 표시
text	파일 내용 표시 (압축 파일의 내용도 표시)
mkdir	디렉터리 생성
put, copyFromLocal	로컬 파일 시스템의 파일 및 디렉터를 HDFS로 복사
get, copyToLocal	HDFS의 파일 또는 디렉터를 로컬 파일 시스템으로 복사

# HDFS 명령어

명령어	설명
cp	파일 또는 디렉터리 복사
mv	파일 또는 디렉터리 이동
rm	파일 또는 디렉터리 삭제
tail	파일의 마지막 부분 내용 출력
chmod	지정된 경로의 파일 또는 디렉터리의 권한 변경
chown	지정된 파일 또는 디렉터리의 소유자 변경
chgrp	지정된 파일 또는 디렉터리의 소유 그룹 변경