

# Outline

Module 1 : 大數據簡介

Module 2 : Hadoop Ecosystem介紹

Module 3 : Hadoop 平台安裝

Module 4 : Hadoop 分散式檔案系統 (HDFS)

Module 5 : Hadoop MapReduce

**Module 5補充 : Hadoop HA (高可用性)**

Module 6 : Apache Hive

Module 7 : Sqoop與Flume

Module 8 : Apache Spark

Module 9 : Spark 平台安裝

Module 10 : RDD — Resilient distributed dataset

Module 11 : Scala 程式開發基礎

Module 12 : Spark SQL 及 DataFrame

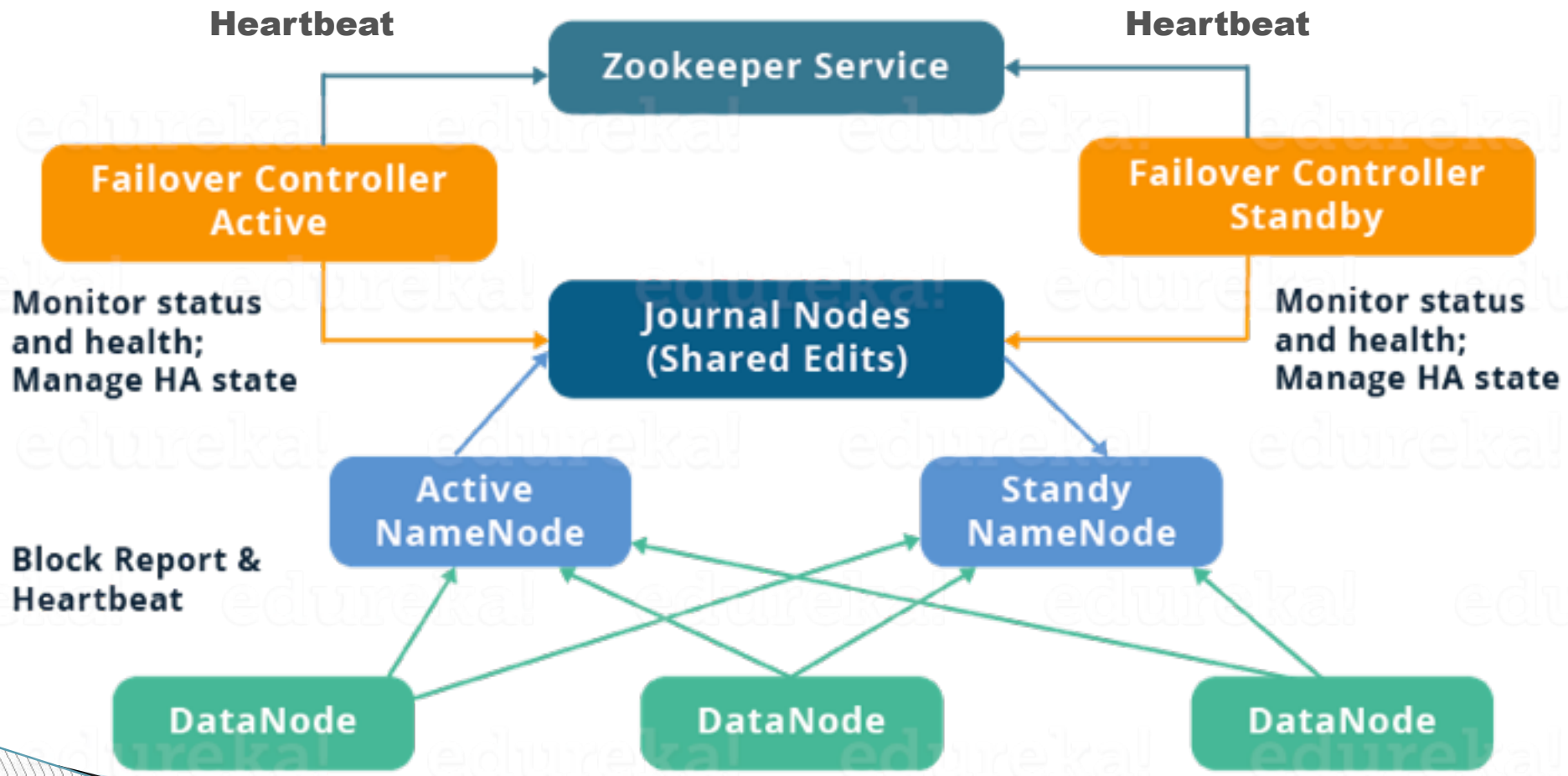
Module 13 : Spark 機器學習函式庫(MLlib)



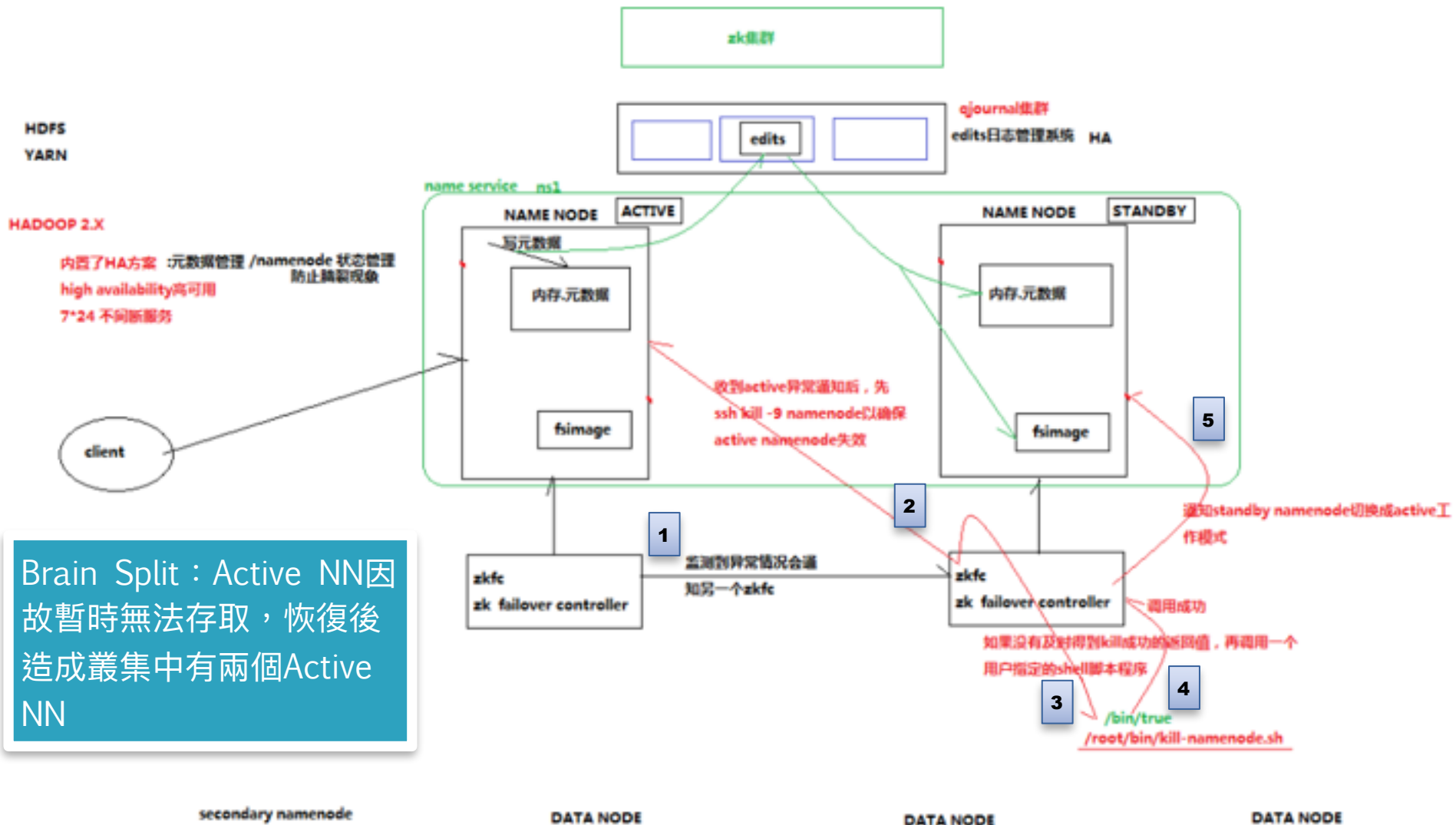
# Hadoop 高可用性(HA)叢集

- ▶ Hadoop HDFS 數據儲存有保障，但**服務可用性無保障**
  - Hadoop **NameNode** 存在單點故障問題(Single Point of Failure)
  - 雖有**Secondary NameNode**設計，但不為 **HA** 用途
- ▶ Hadoop HA 架構
  - 叢集中有**兩個NameNode** (**Active – Standby**)
  - 使用**JournalNode**管理**edits\_log**，並依附**zookeeper**來實現自動切換

# Hadoop HA 架構(with zookeeper)



# 防止NameNode假死(Brain Split)



Brain Split : Active NN因  
故暫時無法存取，恢復後  
造成叢集中有兩個Active  
NN

# Hadoop HA 架構安裝 — 規劃

- ▶ 最少用三台 VM 來演練
  - Hadoop1: NN1, ZKFC1, ZK1, JN1, DN1, RM1, NM1
  - Hadoop2: NN2, ZKFC2, ZK2, JN2, DN2, RM2, NM2
  - Hadoop3: ZK3 JN3
- ▶ 正常配置：7台
  - 1: NN ZKFC
  - 2: NN ZKFC
  - 3,4: ResourceManager
  - 5,6,7: ZK JN DN NodeManager
    - (DN和NM放一起可實現在地運算)

# Hadoop HA 架構安裝

- ▶ 安裝zookeeper叢集
  - 下載zookeeper後解壓縮
  - 設定`zoo.cfg` => 設定要啟動zookeeper的主機
    - `server.1=hadoop1:2888:3888`
    - `server.2=hadoop2:2888:3888`
  - 啟動 => `./bin/zkServer.sh start`
  - 看狀態 => `./bin/zkServer.sh status`
  - 看資料 => `./bin/zkCli.sh`

# Hadoop HA 架構安裝

- ▶ 安裝Hadoop叢集
  - 先架好hadoop multi-node，再改成HA模式
    - 改hdfs-site.xml
    - 改core-site.xml
    - 改yarn-site.xml
  - 確認叢集內各台電腦都可免輸帳密登入ssh

# Hadoop HA 架構啟動

## ▶ 啟動Hadoop叢集

- start zookeeper(安裝ZK者分別啟動) : `./bin/zkServer.sh start => jps` 要看到 QuorumPeerMain
- start journal node(安裝JN者分別啟動，第一次手動啟動讓namenode format能成功，之後改由start-dfs來啟動)
- format namenode(在其中一台NN執行) : `hdfs namenode -format`(多執行幾次可發現會format所有NN)
- format standby namenode(在另一台NN執行) : `hdfs namenode -bootstrapStandby`
- format ZKFC(在其中一台NN執行): `hdfs zkfc -formatZK`
  - 用zkCli.sh 可 `ls /hadoop-ha` 會看到及 `/yarn-leader-election` 有註冊 cluster 項目
- 啟動 HDFS : `start-dfs.sh`
- 啟動 YARN : `start-yarn.sh`
  - 另一台 ResourceManager 不會帶起來，用 `./sbin/yarn-daemon.sh start resourcemanager`



# Hadoop HA 架構驗證

- ▶ Zookeeper及Hadoop Process 驗證(用jps指令)
  - ZKFC (2台NN都有)
  - QuorumPeerMain (所有ZK主機都有)
  - DataNode (所有DN都有)
  - Namenode (2台NN都有)
  - ResourceManager (2台RM都有)
  - NodeManager (所有DN都有)
  - JournalNode (所有JN主機都有)

```
hduser@hadoop1:/usr/local/hadoop$ jps
32564 Jps
32452 NodeManager
4789 QuorumPeerMain
29909 DataNode
25270 ResourceManager
30363 DFSZKFailoverController
29773 NameNode
30142 JournalNode
```

```
hduser@hadoop3:/usr/local/hadoop$ jps
16282 Jps
4492 QuorumPeerMain
15327 JournalNode
```

# Hadoop HA 架構驗證

## ▶ 由網頁及指令驗證

- Hadoop : 由 `http://hadoop1:50070` 和 `http://hadoop2:50070` 看兩台NN的狀況
- YARN : 由 `http://hadoop1:8088` 看 yarn 狀況
  - 用 `yarn rmadmin -getServiceState rm1` 可看何者為active

## ▶ HA 切換驗證

- `sudo kill -9 <pid of NodeManager>`
  - 觀察另一NN是否順利接手
- `sudo kill -9 <pid of ResourceManager>`
  - 觀察另一RM是否順利接手

# 動態加入DataNode

## ▶ 安裝及設定

- 建立一台主機，將所有 hadoop 設定 copy 至該主機
- 確認 NN 主機可辨識新DN主機名稱並可SSH至新DN
- 確認 `hadoop_data/hdfs/namenode/` 及 `hadoop_data/hdfs/datanode/` 下沒有 `current` 目錄（有就刪除）
- 在新主機啟動DN：`./sbin/hadoop-daemon.sh start datanode`
- 修改兩台NN 的 `slaves` 將 DN 加入以便利用 `start-dfs` 啟動

## ▶ 由 Active NN的50070 網頁查看可看到 DN 數增加

# Reference

- ▶ HDFS HA
- ▶ YARN HA
- ▶ HDFS Federation