

بخش دوم

آشنایی با معماری  
هadoop

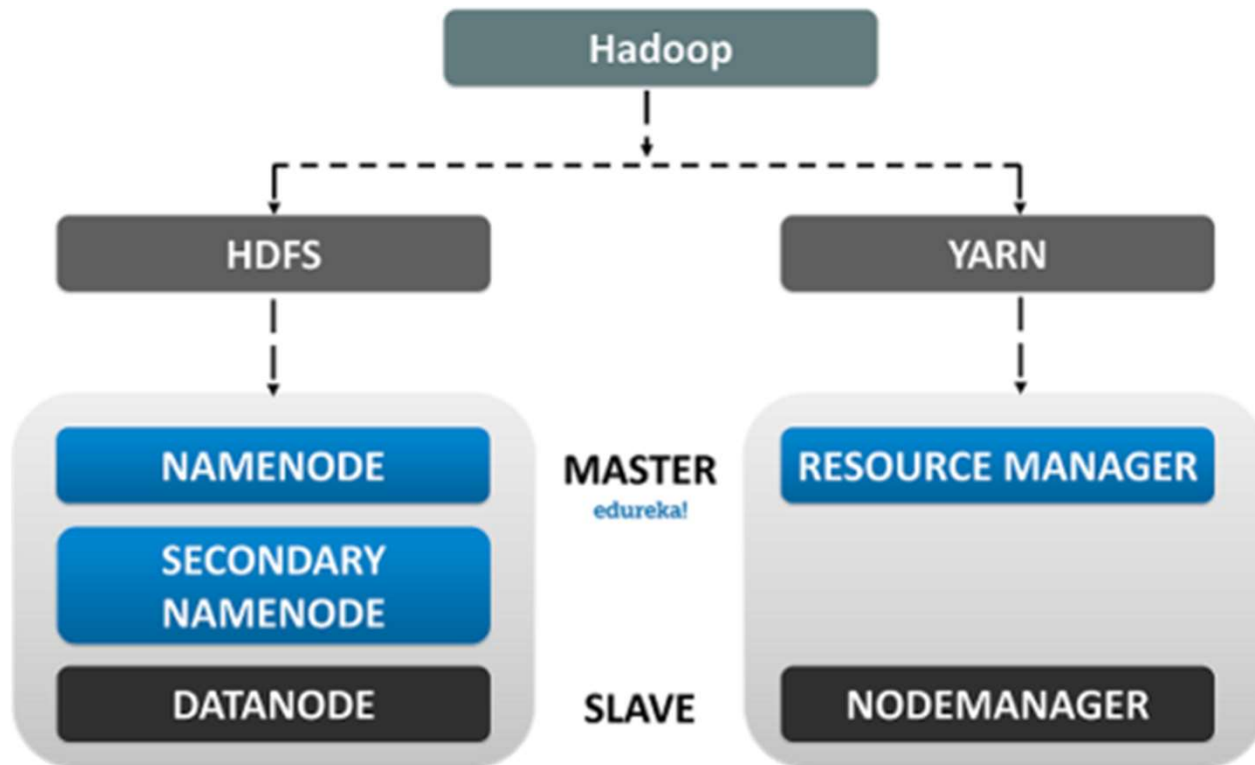


# آنچه خواهیم دید

- HDFS : معماری و مفاهیم پایه
- HDFS : دستورات اصلی و نحوه کار
- MapReduce : اجزاء و مفاهیم
- Yarn : معماری و مفاهیم پایه
- MapReduce : بررسی یک مثال

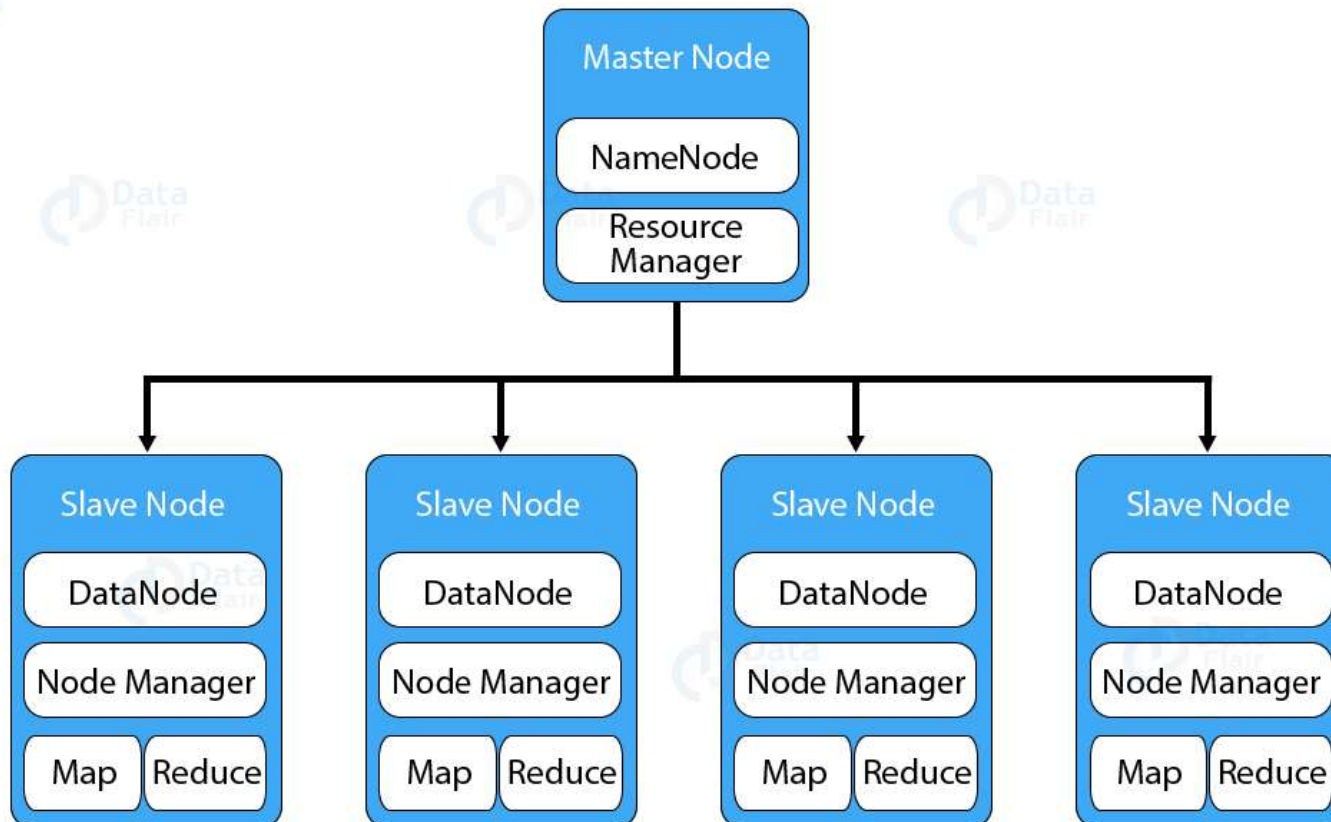


# کلاستر هدوپ در یک نگاه



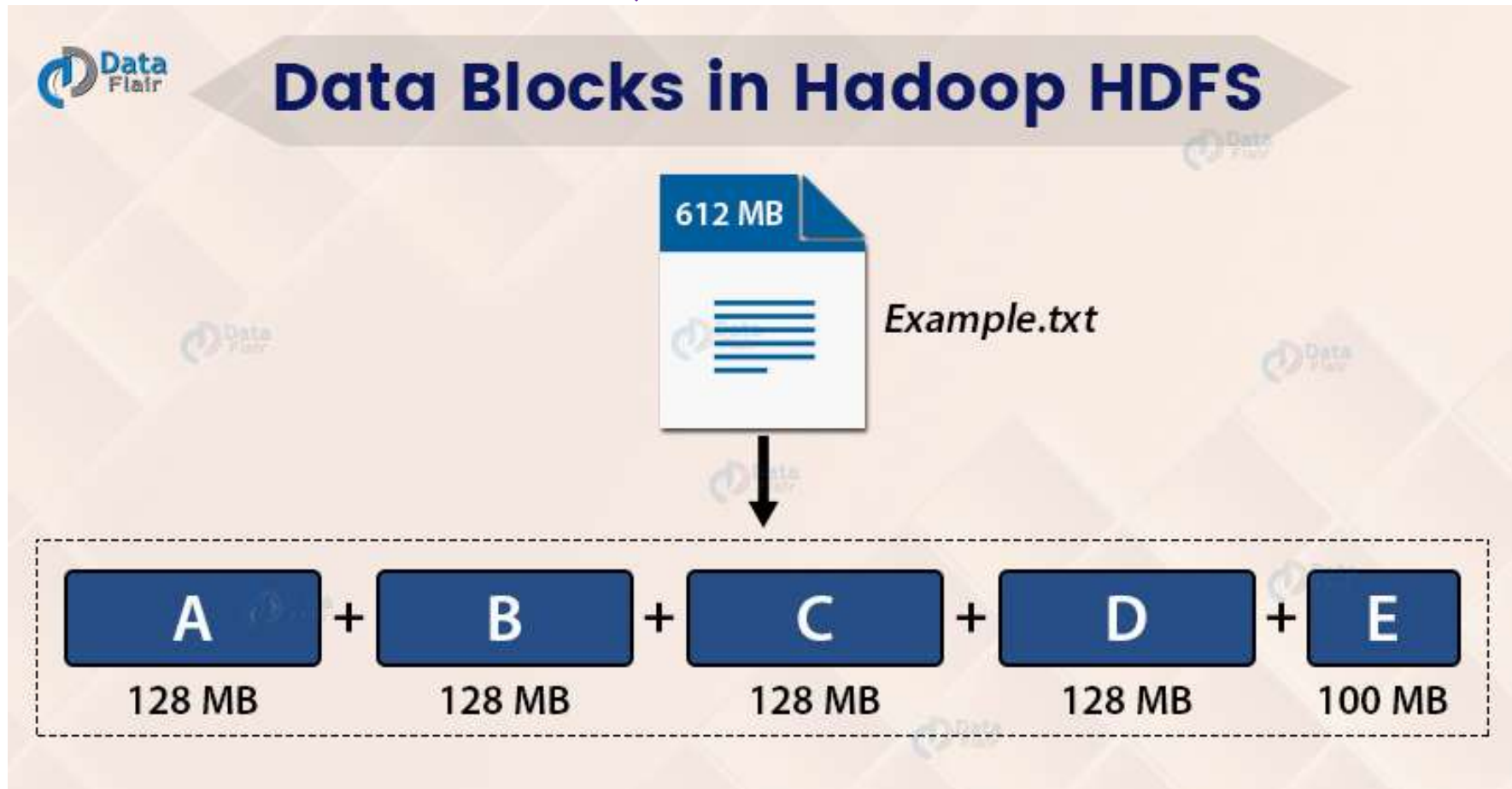


# کلاستر هدوپ در یک نگاه





# بلاک‌بندی فایل‌ها در هدوپ

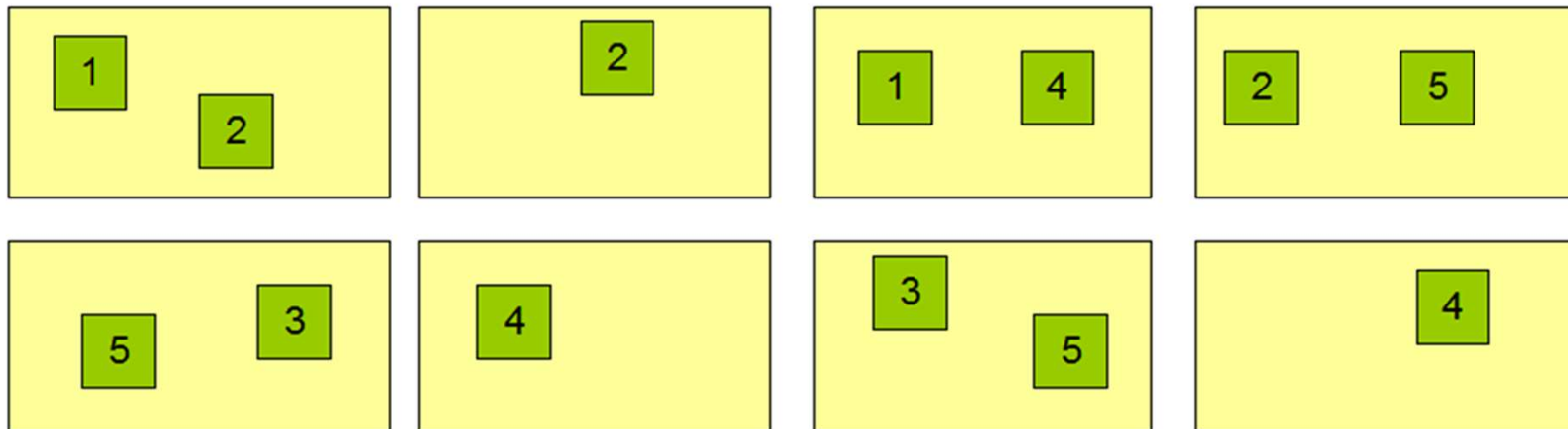




# ضریب تکرار

Namenode (Filename, numReplicas, block-ids, ...)  
/users/sameerp/data/part-0, r:2, {1,3}, ...  
/users/sameerp/data/part-1, r:3, {2,4,5}, ...

Datanodes





# HDFS Rack Awareness

## Rack Awareness Algorithm

Block A :  Block B:  Block C: 





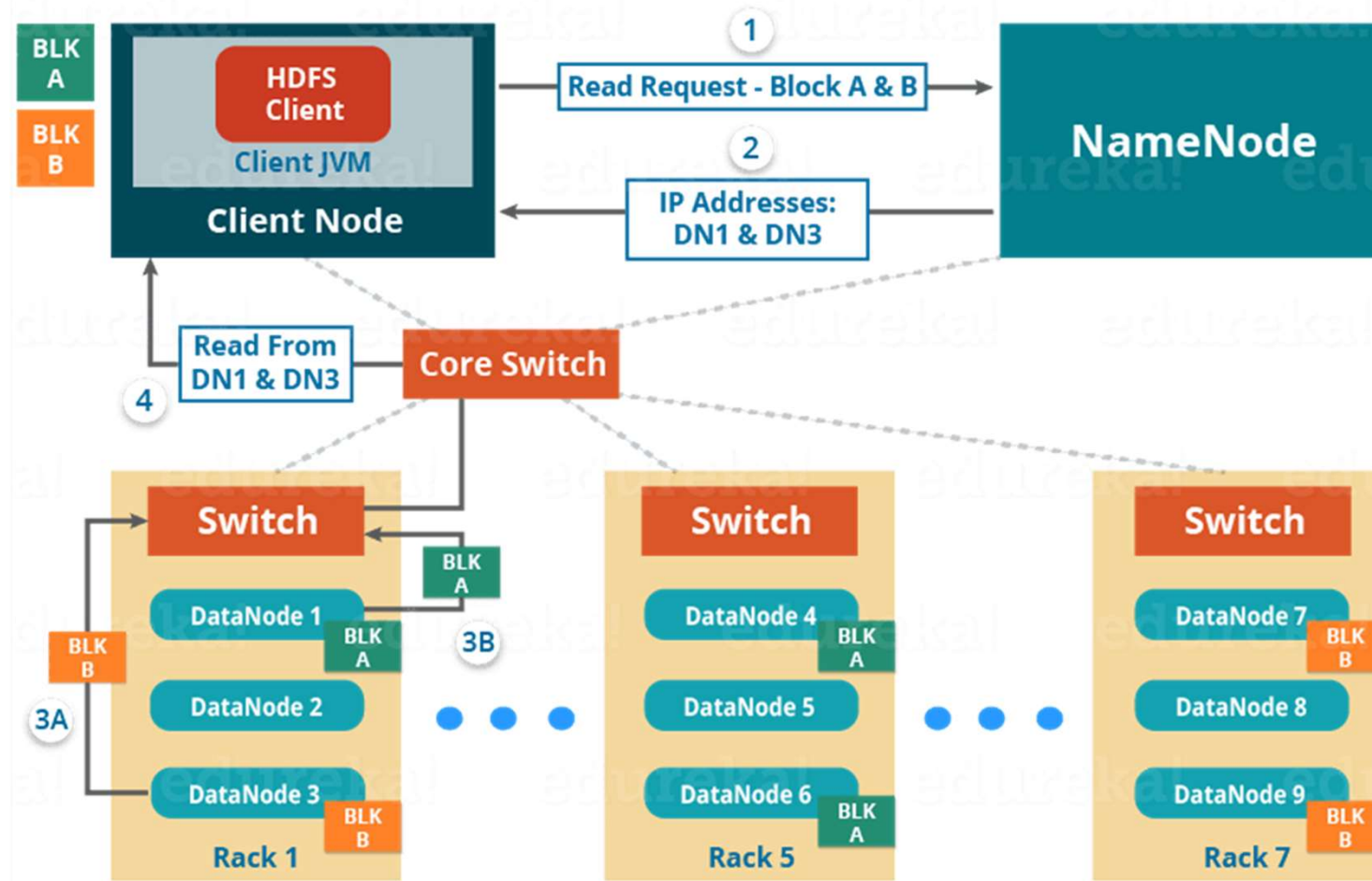
# Name Node Meta Data

object	block_id	seq	locations	ACL	Checksum
/data/file.txt	blk_00123	1	[node1,node2,node3]	-rwxrwxrwx	8743b52063 ..
/data/file.txt	blk_00124	2	[node2,node3,node4]	-rwxrwxrwx	cd84097a65 ..
/data/file.txt	blk_00125	3	[node2,node4,node5]	-rwxrwxrwx	d1633f5c74 ..



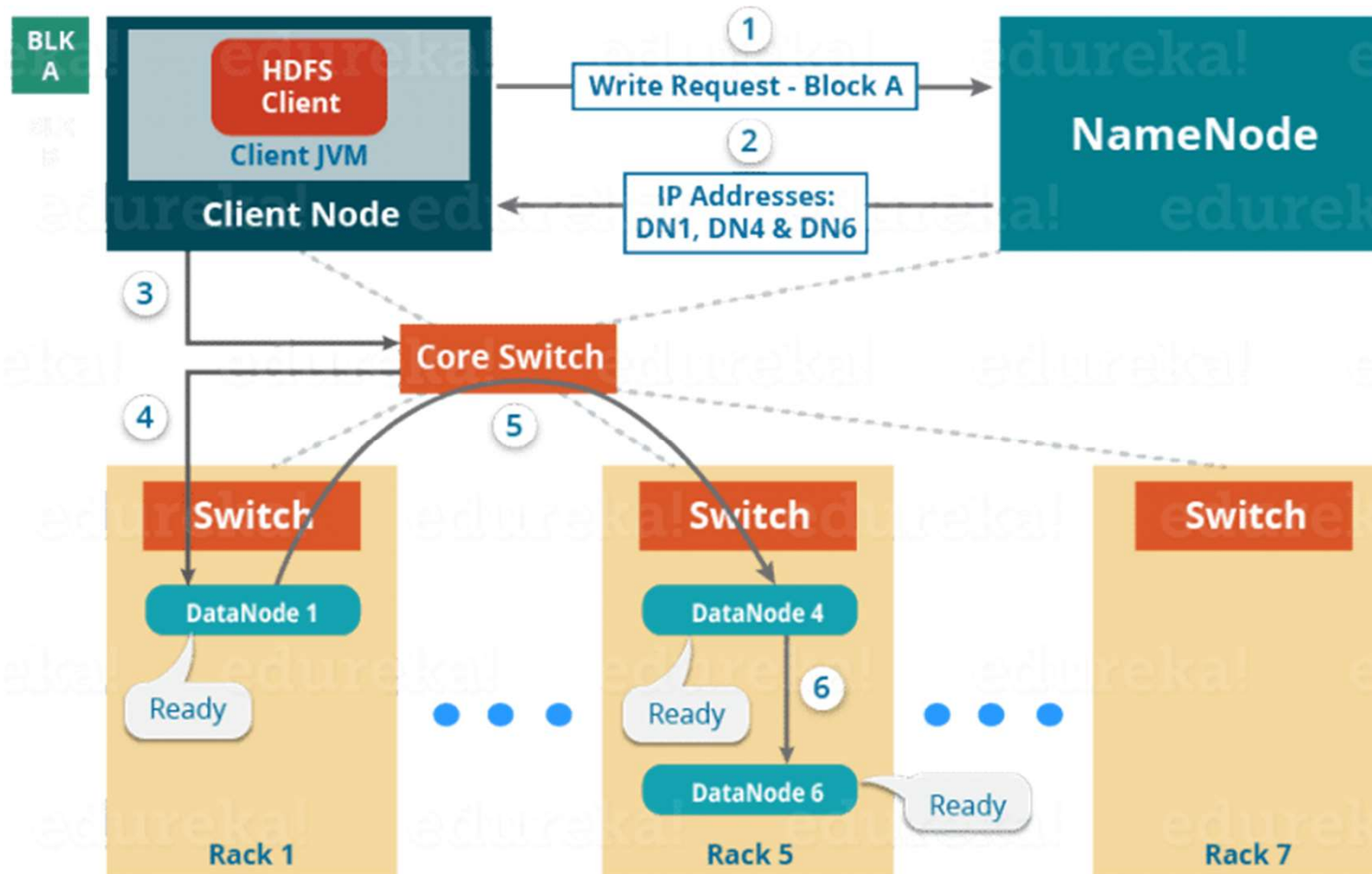


# معماری HDFS – خواندن یک فایل



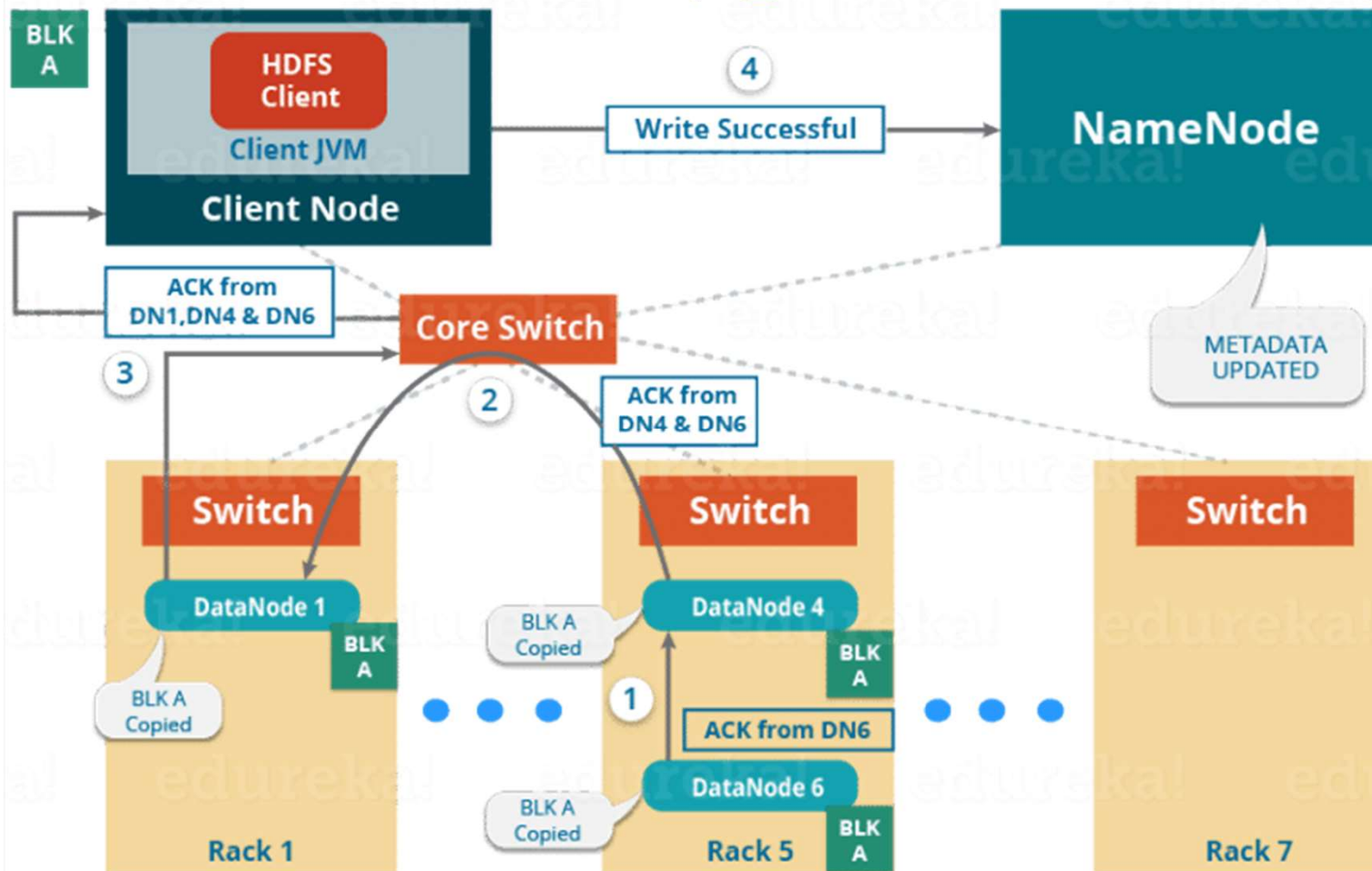


# معماری HDFS – نوشتن در یک فایل



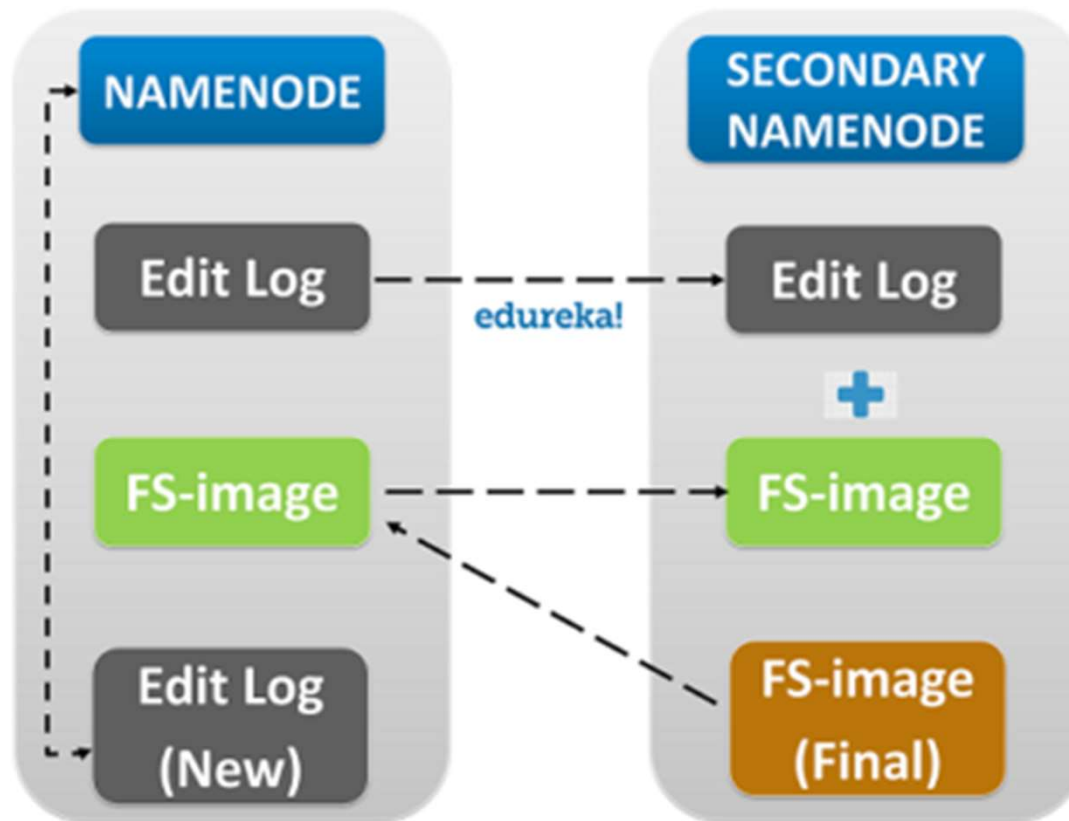


# معماری HDFS – مکانیزم تایید



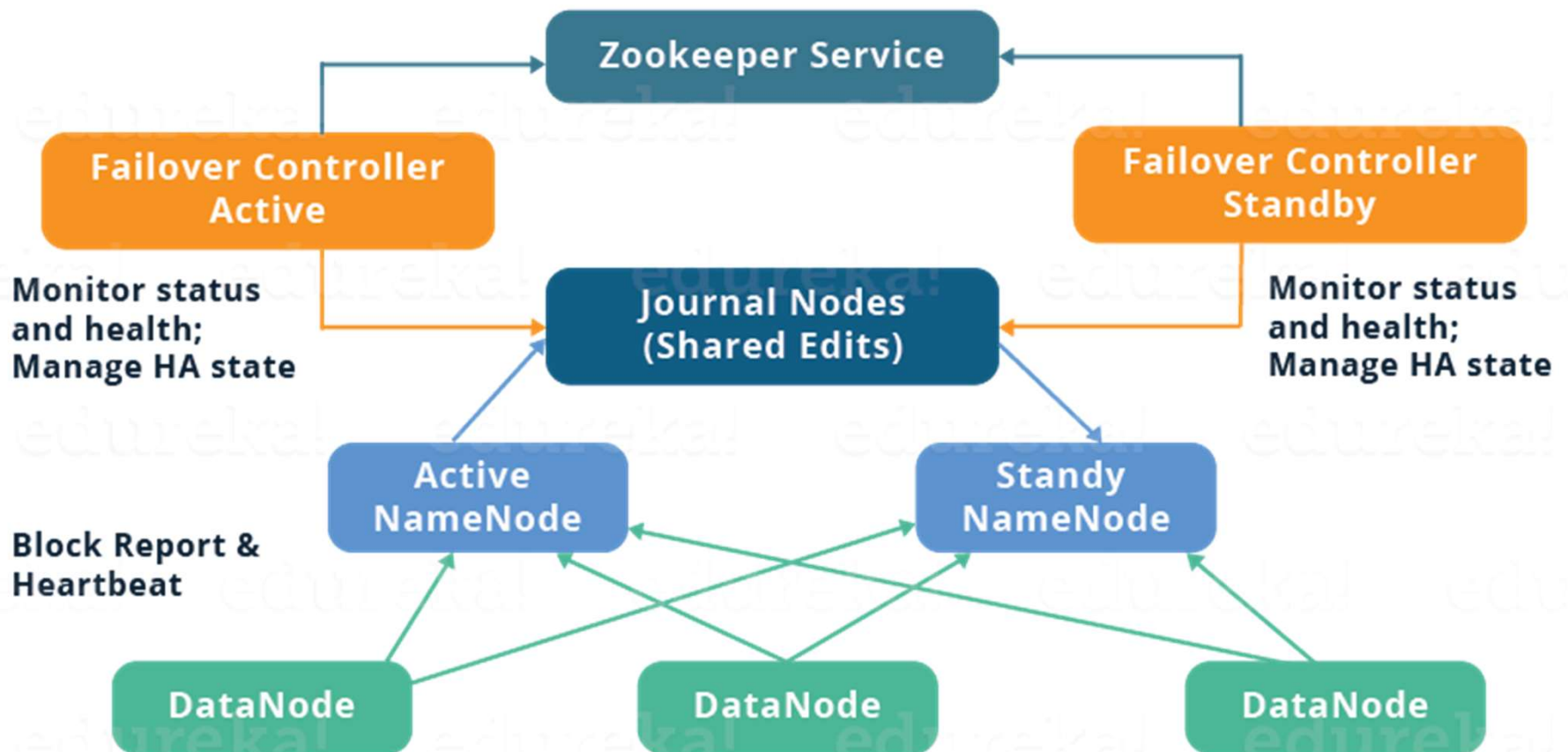


# نقش Secondary Name Node



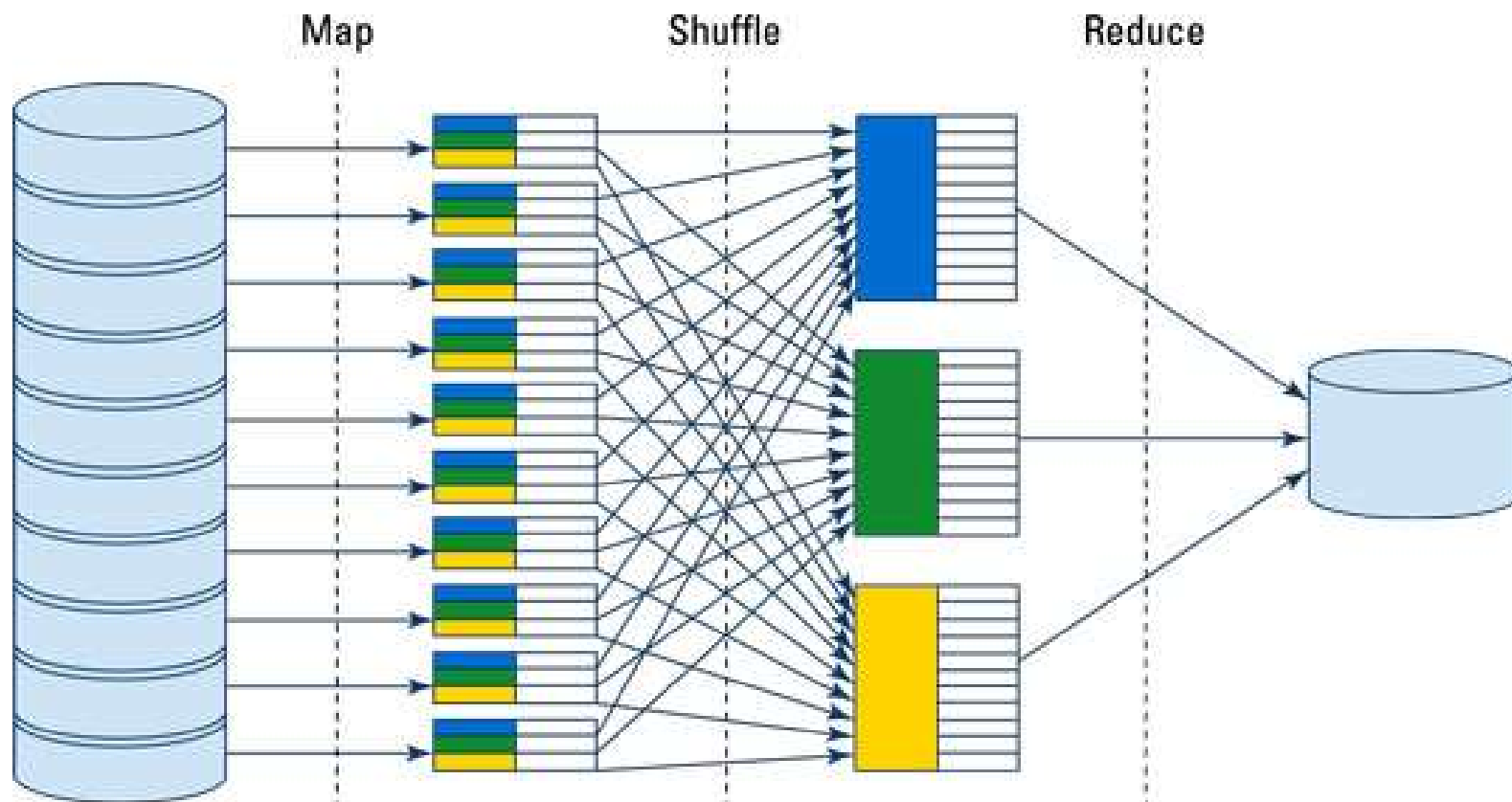


# تضمین HA در هدوپ





# فرآیند پردازش داده در هدوپ – MR



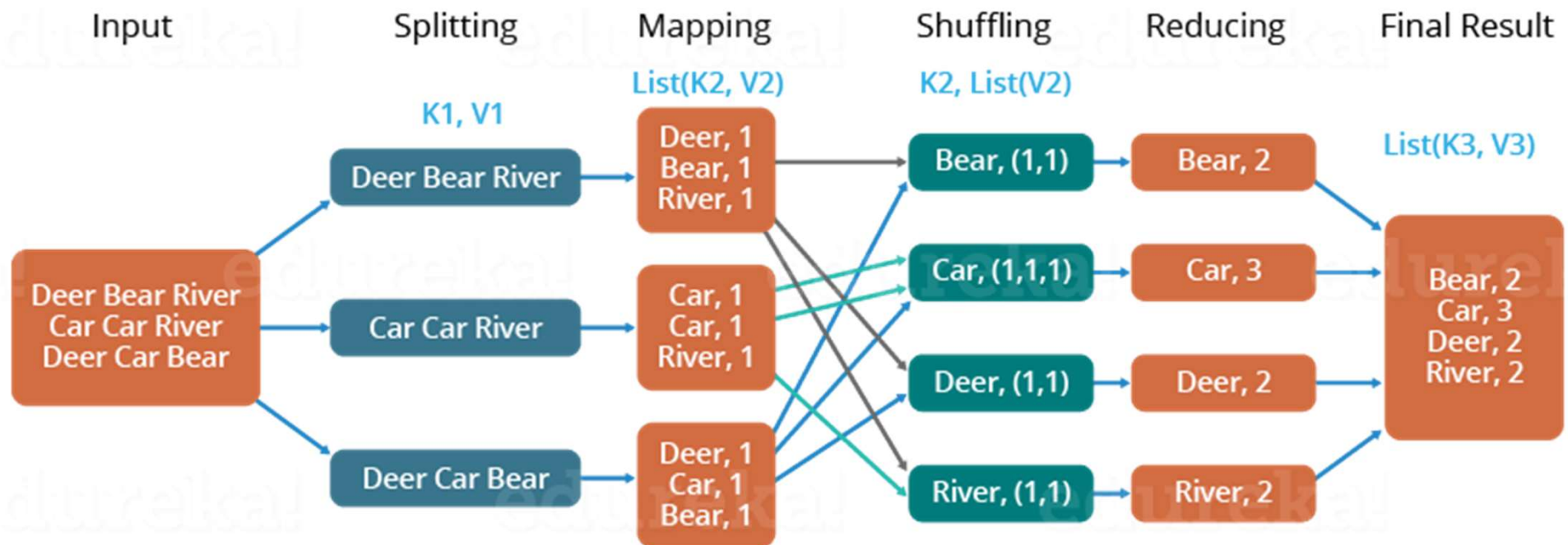




# شمارش کلمات با MR

edureka!

## The Overall MapReduce Word Count Process





# یک مثال ساده با پایتون

```
from mrjob.job import MRJob

class MRWordFrequencyCount(MRJob):
    def mapper(self, _, line):
        yield "chars", len(line)
        yield "words", len(line.split())
        yield "lines", 1
    def reducer(self, key, values):
        yield key, sum(values)

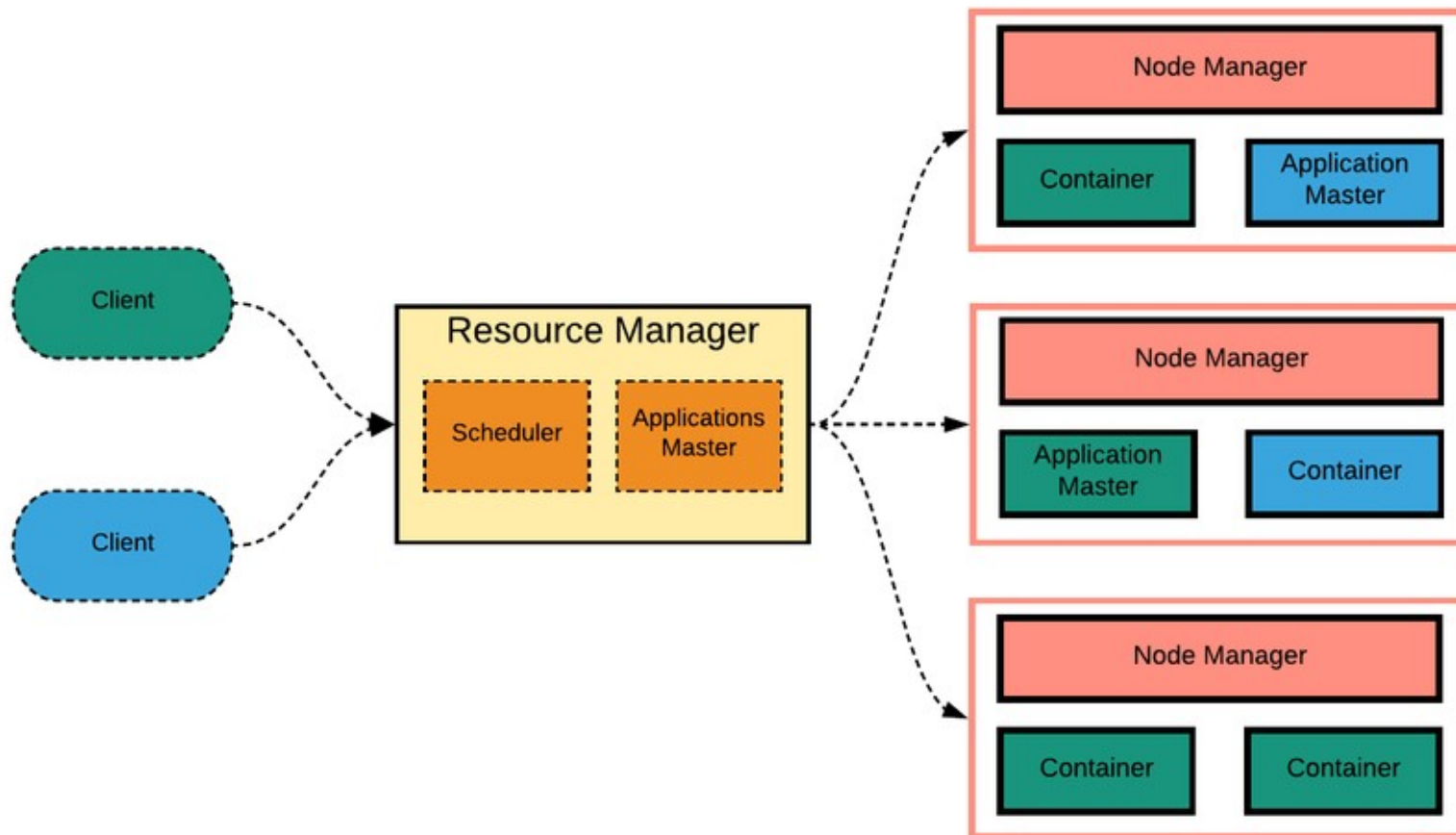
if __name__ == '__main__':
    MRWordFrequencyCount.run()
```

*python mr\_word\_count.py my\_file.txt*



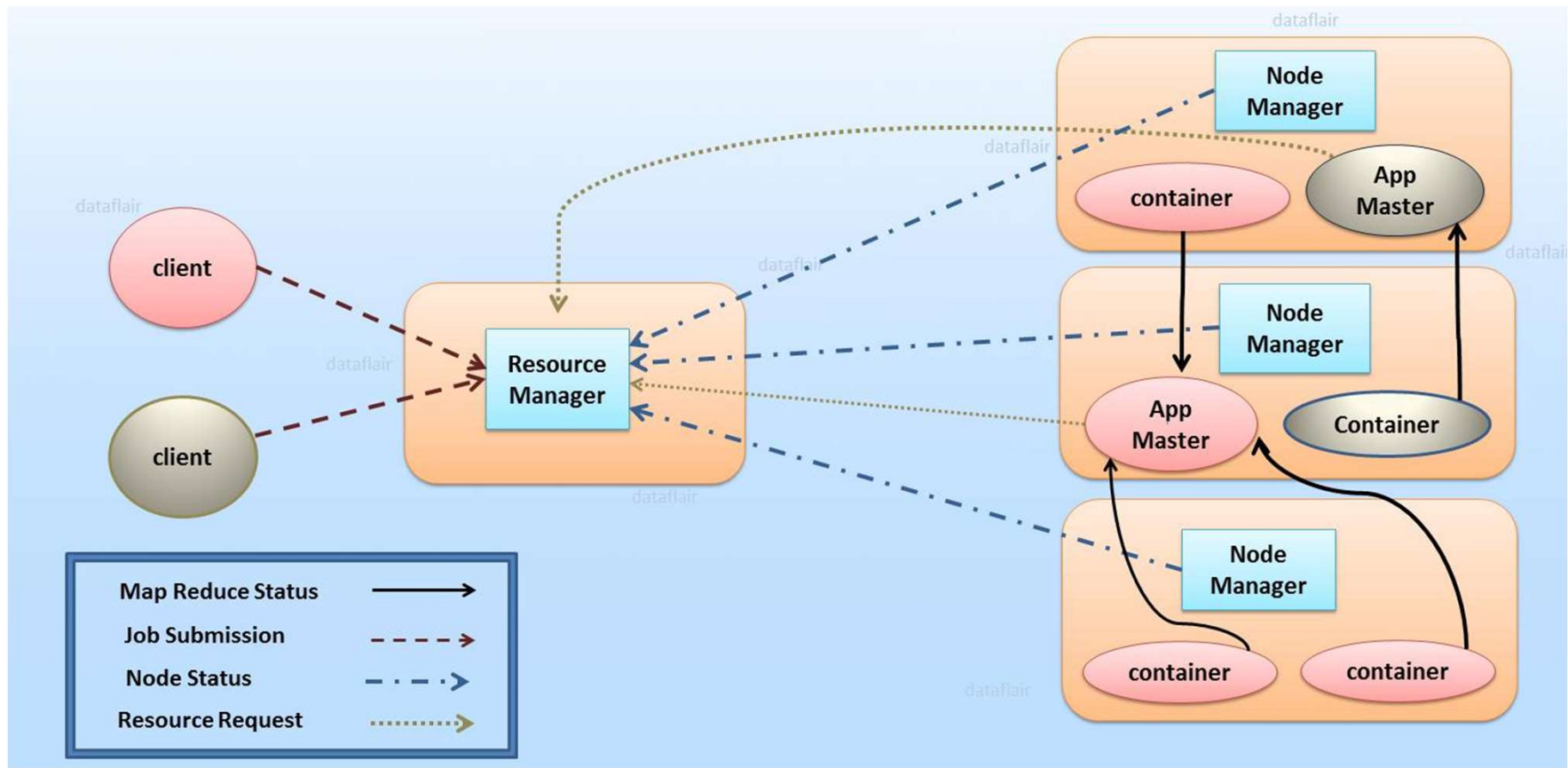


# معماری Yarn





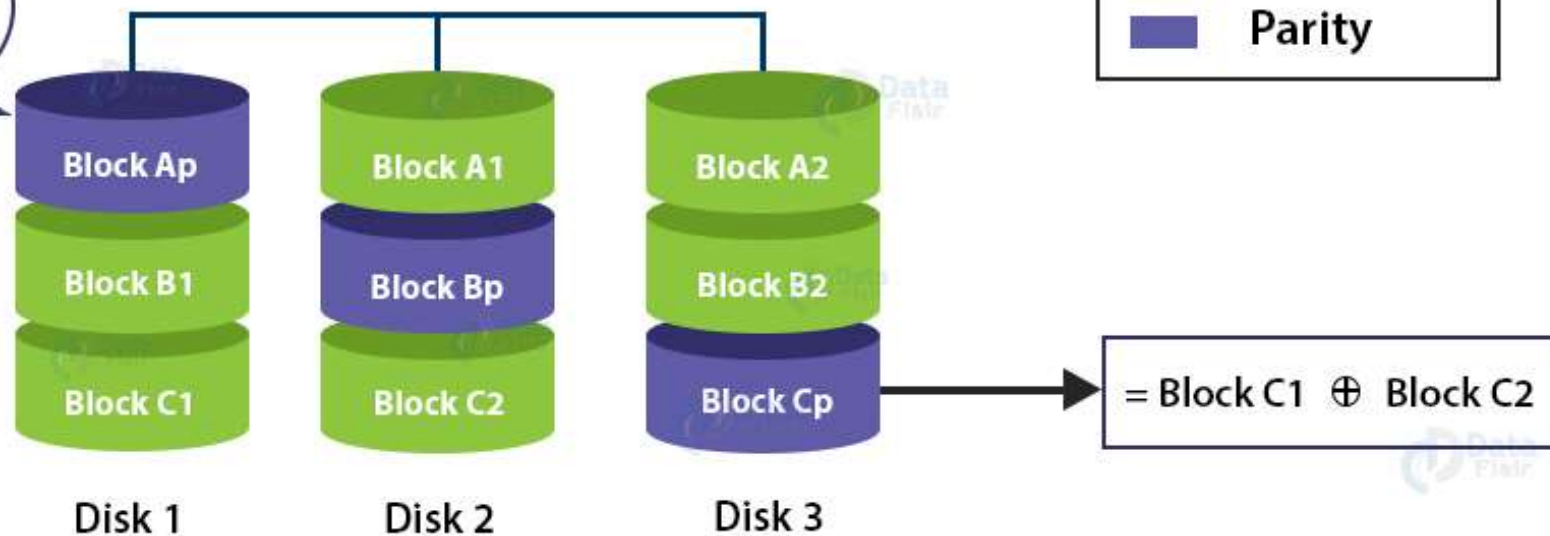
# معماری Yarn – ارتباط مولفه‌ها





# Erasure Coding

Reduced storage overhead  
as 1 Parity Block stored for  
2 Data Blocks





# مروری بر دستورات پرکاربرد HDFS





# نحوه فراخوانی و کاربرد

- `hadoop version`
- `hadoop fs -mkdir /path/directory_name`
- `hadoop fs -ls -Rh /path/directory_name`
- `hadoop fs -put <localsrc> <dest>`
- `hadoop fs -get <src> <localdest>`
- `hadoop fs -copyFromLocal <localsrc> <hdfs destination>`
- `hadoop fs -copyToLocal <hdfs source> <localdst>`
- `hadoop fs -cat /path_to_file_in_hdfs`
- `hadoop fs -mv <src> <dest>`
- `hadoop fs -cp <src> <dest>`



# دستورات پر کاربرد HDFS – بخش دوم







# دستورات پر کاربرد HDFS – بخش سوم

**touchz 01**

**test 02**

**text 03**

**stat 04**

**usage 05**

**help 06**



**07 chmod**

**08 appendToFile**

**09 checksum**

**10 count**

**11 find**

**12 getmerge**