



# Hadoop

HDFS, MAPREDUCE, HIVE



## Обо мне

Егор Матешук, Data Engineer

@ Ostrovok.ru

@ МаксимаТелеком

@ Квант



## Зачем это занятие

После этого занятия вы будете знать

- Что такое Hadoop
- Из каких основных сервисов он состоит
- Как использовать эти сервисы



## План

- Что такое Hadoop
- Zookeeper
- HDFS
- YARN
- MapReduce
- Hive

# Что такое Hadoop

Под словом Hadoop могут подразумеваться

- Несколько сервисов, составляющих “ядро” Hadoop
- Вся экосистема сервисов Hadoop
- Кластер под управлением Hadoop

### Предпосылки появления Hadoop

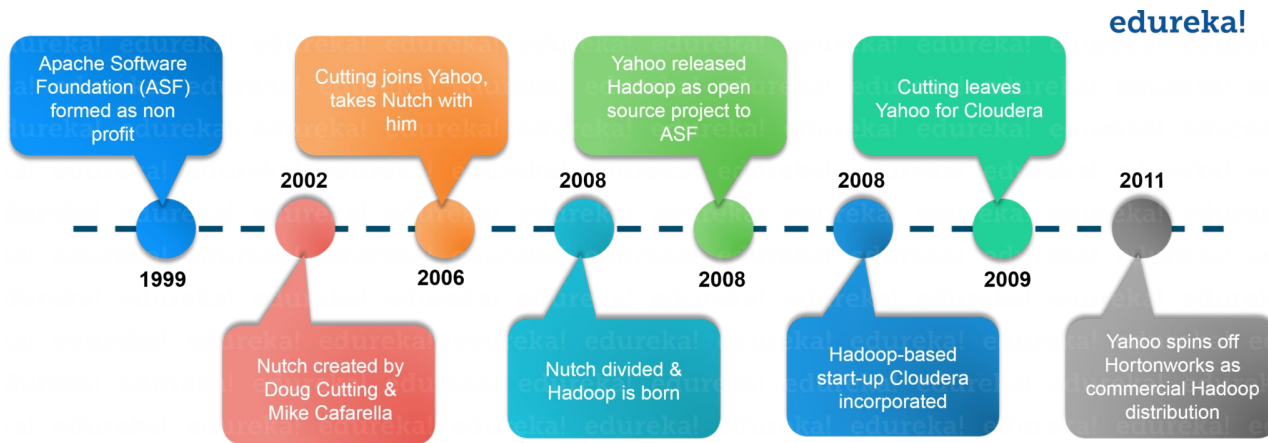
- Потребность в распределенных хранилищах
- Масштабирование вычислений
- Управление ресурсами

- 2003 - Google File System paper
- 2004 - MapReduce paper
- 2006 - Hadoop project
- 2008 - Pig, Hive, HBase
- 2009 - Amazon EMR



## Появление “спецов по Hadoop”

- 2008 - создание Cloudera
- 2011 - выделение Hortonworks



**Hadoop v1.0****MapReduce**Data Processing  
& Resource Management**HDFS**

Distributed File Storage

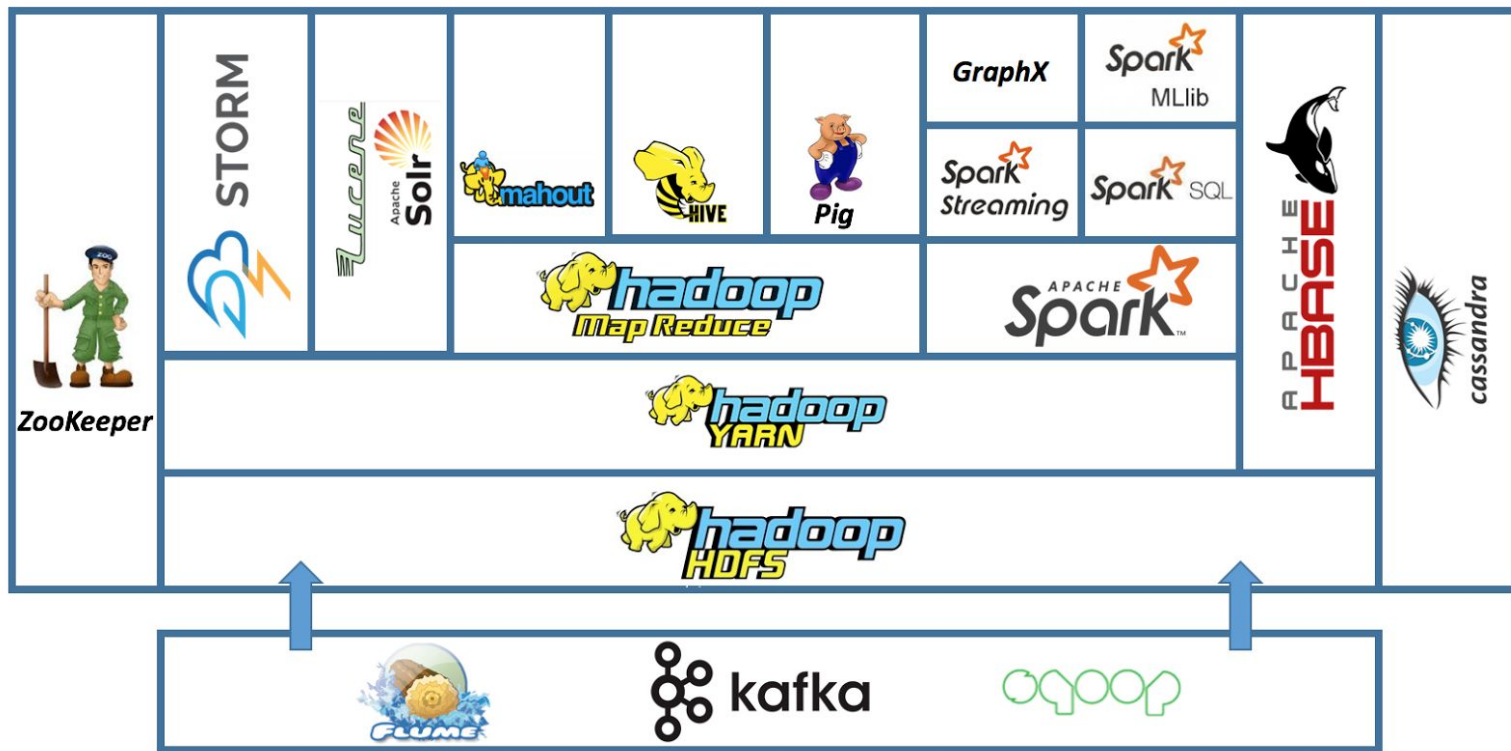
**Hadoop v2.0****MapReduce**Other Data  
Processing  
Frameworks**YARN**

Resource Management

**HDFS**

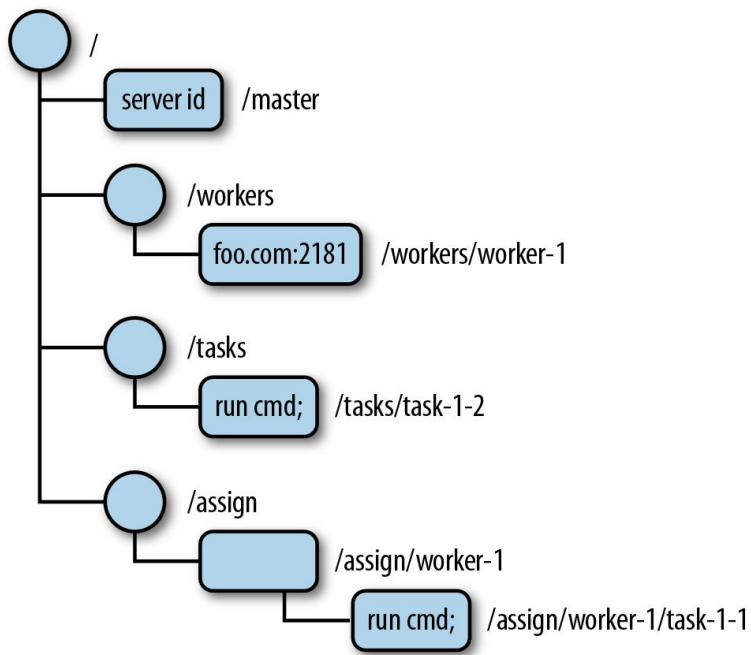
Distributed File Storage

## Экосистема Hadoop

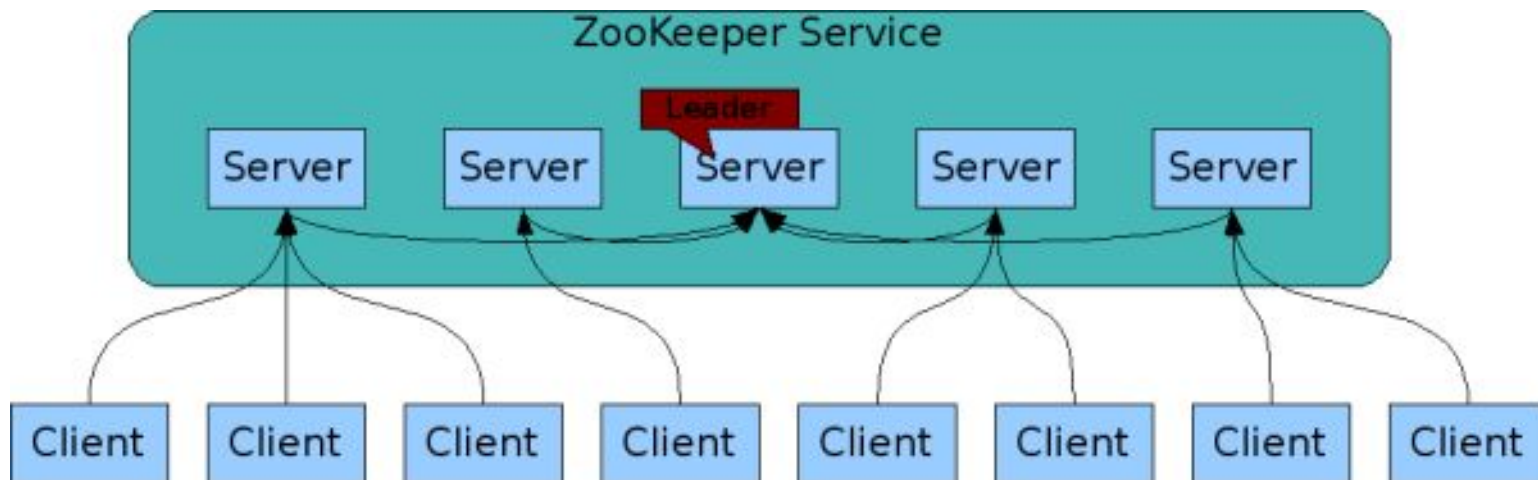


# Zookeeper

**Apache ZooKeeper** - это централизованная служба для поддержки информации о конфигурации, именования, обеспечения распределенной синхронизации и предоставления групповых служб.



# Zookeeper

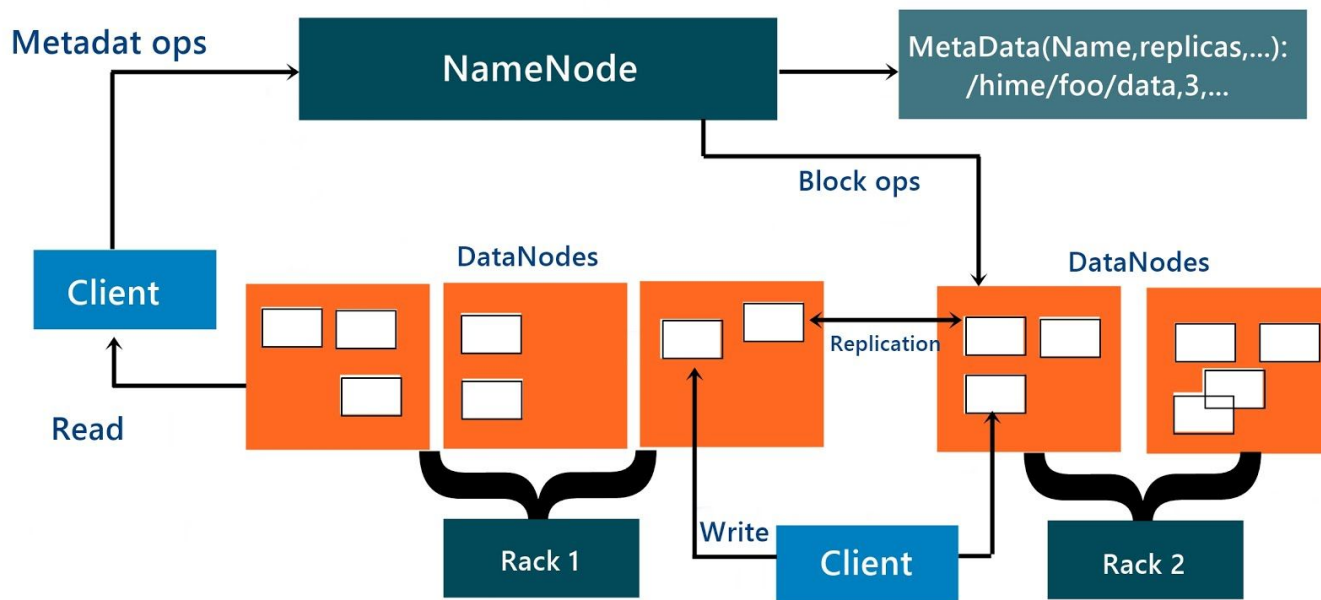


# HDFS



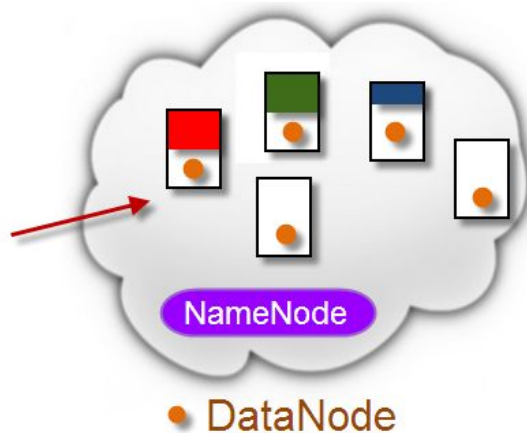
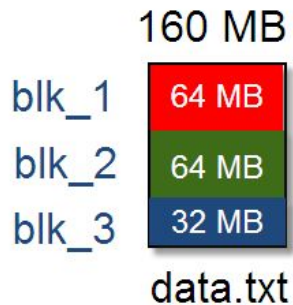
**HDFS** (Hadoop Distributed File System) — файловая система, предназначенная для хранения файлов больших размеров, поблочно распределённых между узлами вычислительного кластера.

## HDFS Architecture



- **Файл** - это только запись в метаданных
- Содержимое файла хранится в нескольких **блоках** одинакового размера

### HDFS



## Name Node: Stores Meta Data

Meta Data:  
/data/pristine/catalina.log.> 1, 2, 4  
/data/pristine/myfile. >3,5

## Data Node 1

1

2

4

5

## Data Node 2

5

2

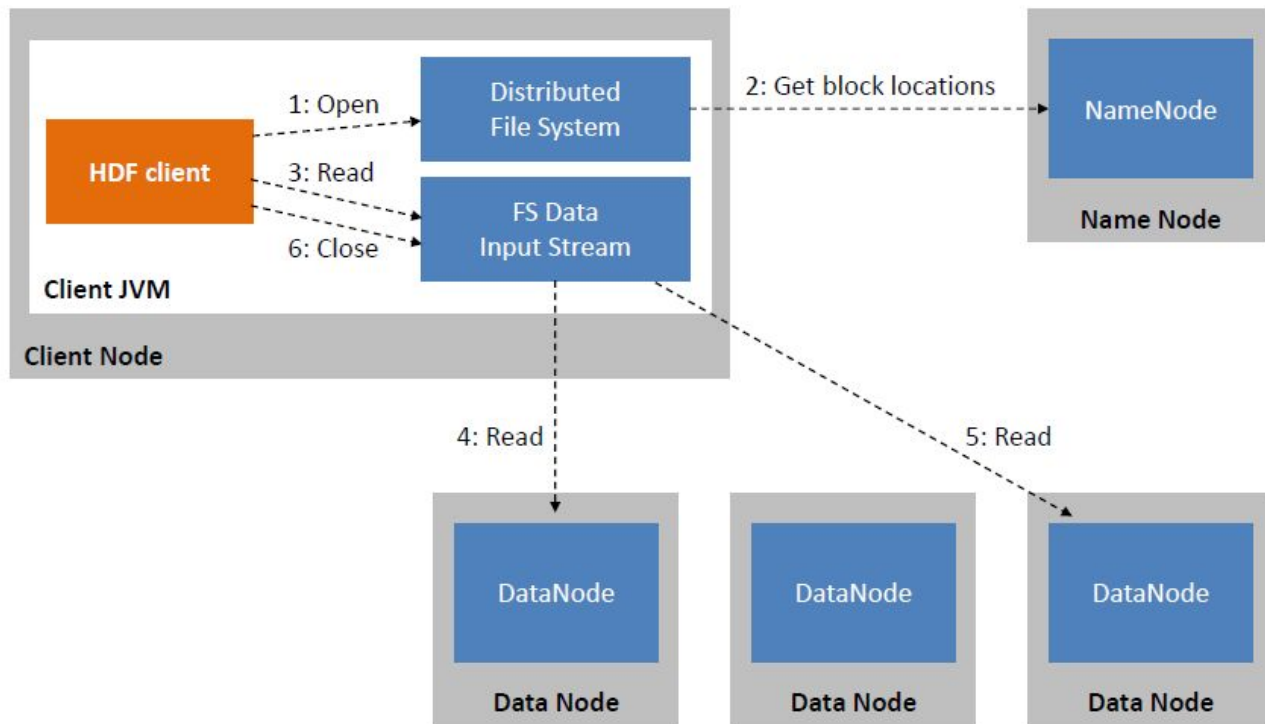
3

## Data Node 3

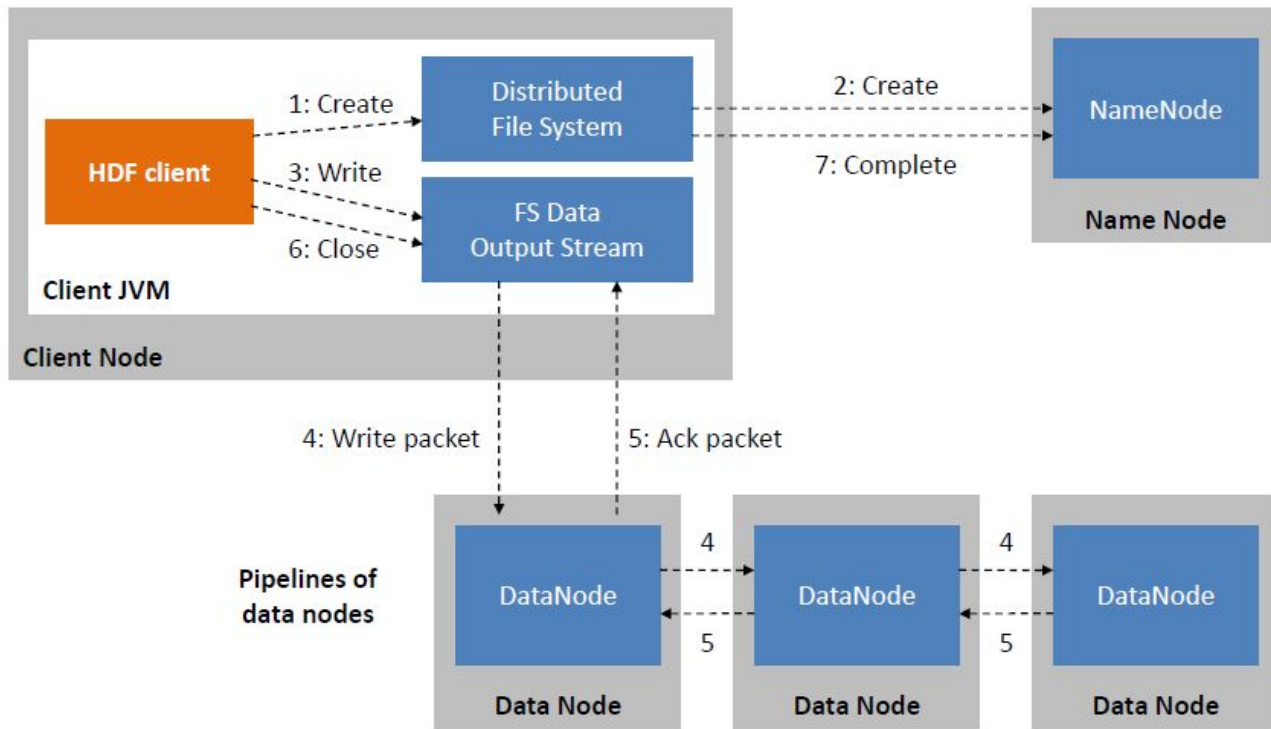
4

1

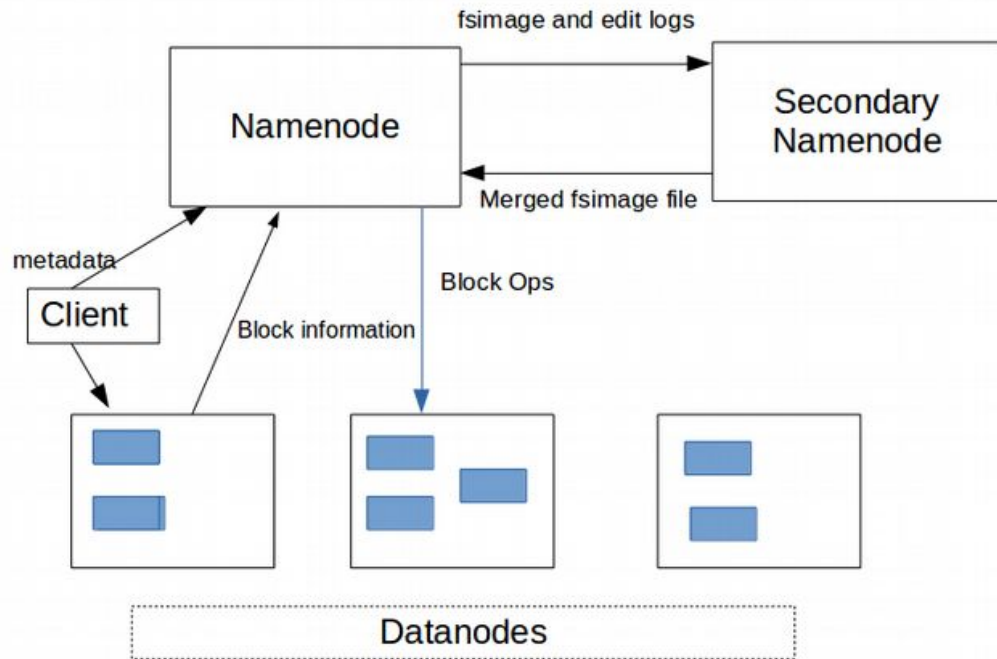
3



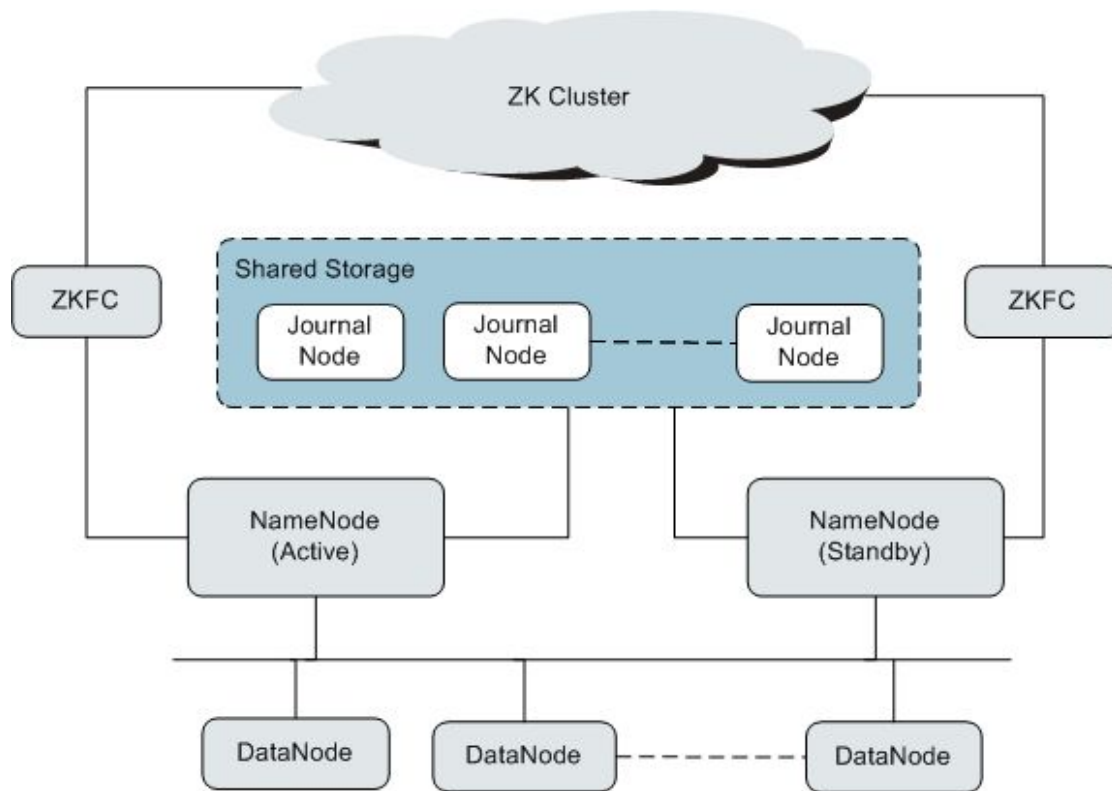
## Запись



# Secondary Namenode

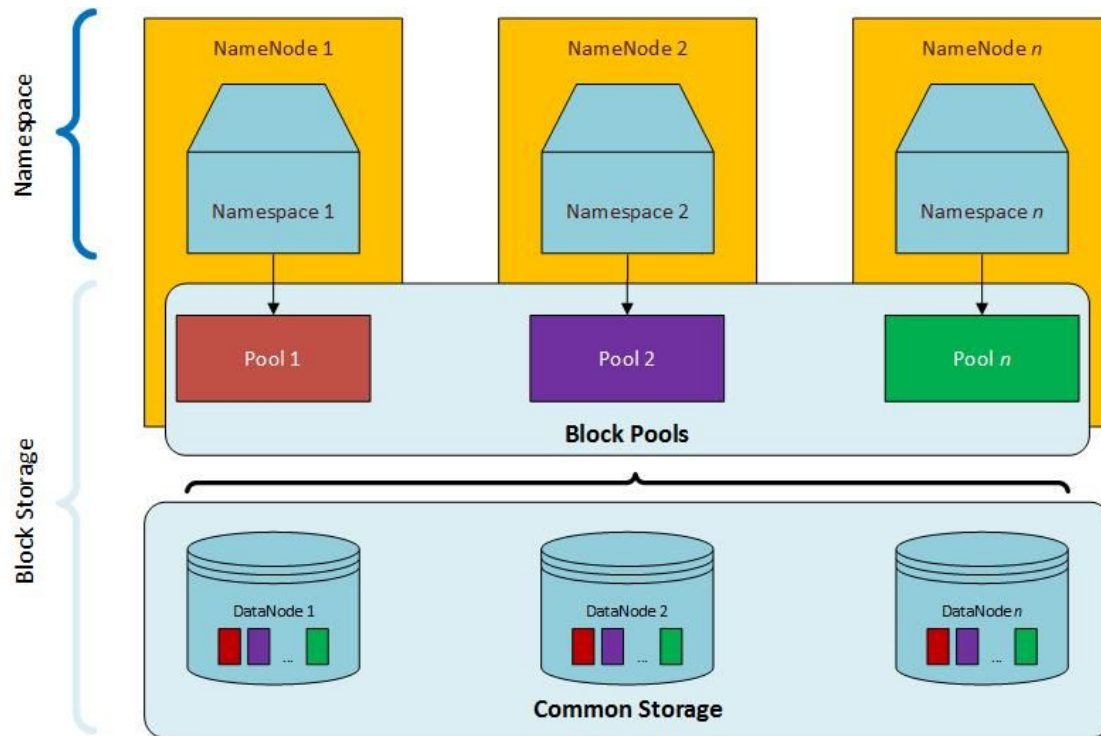


# High-Availability





# Federation



Несколько фактов:

- Не любит мелкие файлы
- Не поддерживает редактирование файлов
- Работает на картошке

У HDFS есть несколько API

- libhdfs - C API
- FileSystem - Java API
- WebHDFS/HttpFS - REST API
- HDFS CLI

Для Java/Scala можно использовать FileSystem из стандартного Hadoop API

<https://hadoop.apache.org/docs/current/api/org/apache/hadoop/fs/FileSystem.html>

На основе этих API есть несколько Python-библиотек

- hdfs
- hdfs3
- snakebite

```
from hdfs3 import HDFSFileSystem
```

```
hdfs = HDFSFileSystem(host, port, user)
```

```
with hdfs.open('/path/to/file', 'rb') as f:
```

```
...
```

```
hdfs dfs -cat hdfs://nn1.example.com/file1
```

```
hdfs dfs -ls <args>
```

```
hdfs dfs -cp /user/hadoop/file1 /user/hadoop/file2
```

# YARN





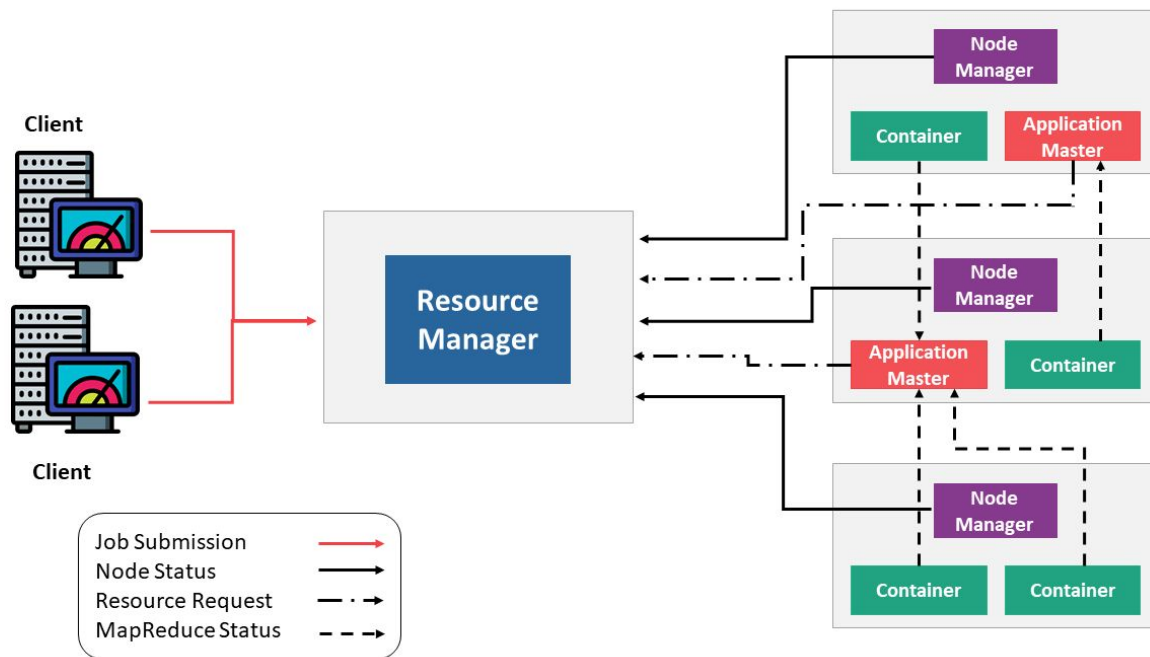
## Системы контейнеризации

- Kubernetes
- Mesos
- YARN



**YARN** (англ. Yet Another Resource Negotiator) — модуль, отвечающий за управление ресурсами кластеров и планирование заданий.

# Контейнеры



Разделение ресурсов происходит по очередям.

Например,

- Отдельная очередь для долгих тяжелых задач
- Отдельная очередь для мелких ad-hoc запросов
- Отдельная очередь для обучения моделей
- Отдельная очередь на каждый отдел и т. д.

# Задача

## Задача

1. Представьте, что вы в команде из 6 человек.
2. Команда получила пакет с набором из стикеров 4 разных цветов.
3. Нужно подсчитать количество каждого цвета.

На обдумывание алгоритма 2-3 минуты

# MapReduce

MapReduce - подход к организации распределенных вычислений, предложенный Google в работе “MapReduce: Simplified Data Processing on Large Clusters”

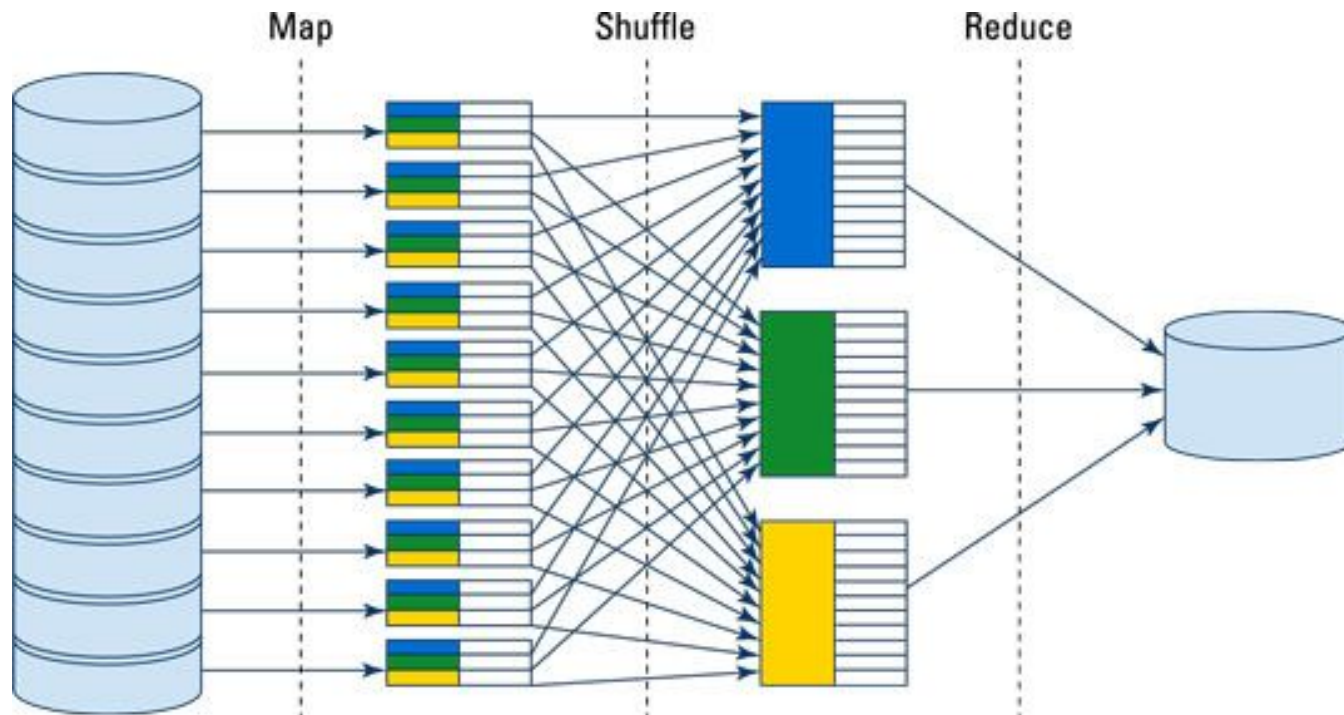


Главная идея:

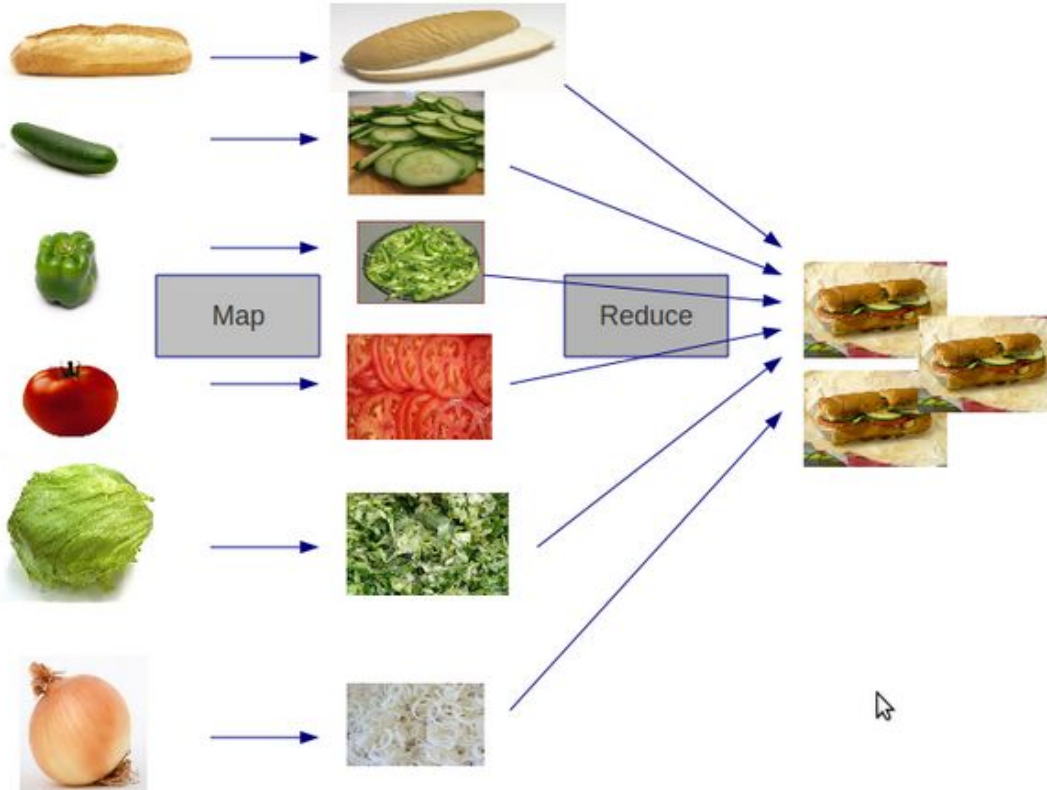
Вычисления можно представить как последовательность операций

- Map - трансформаций
- Shuffle - перераспределения данных
- Reduce - агрегаций

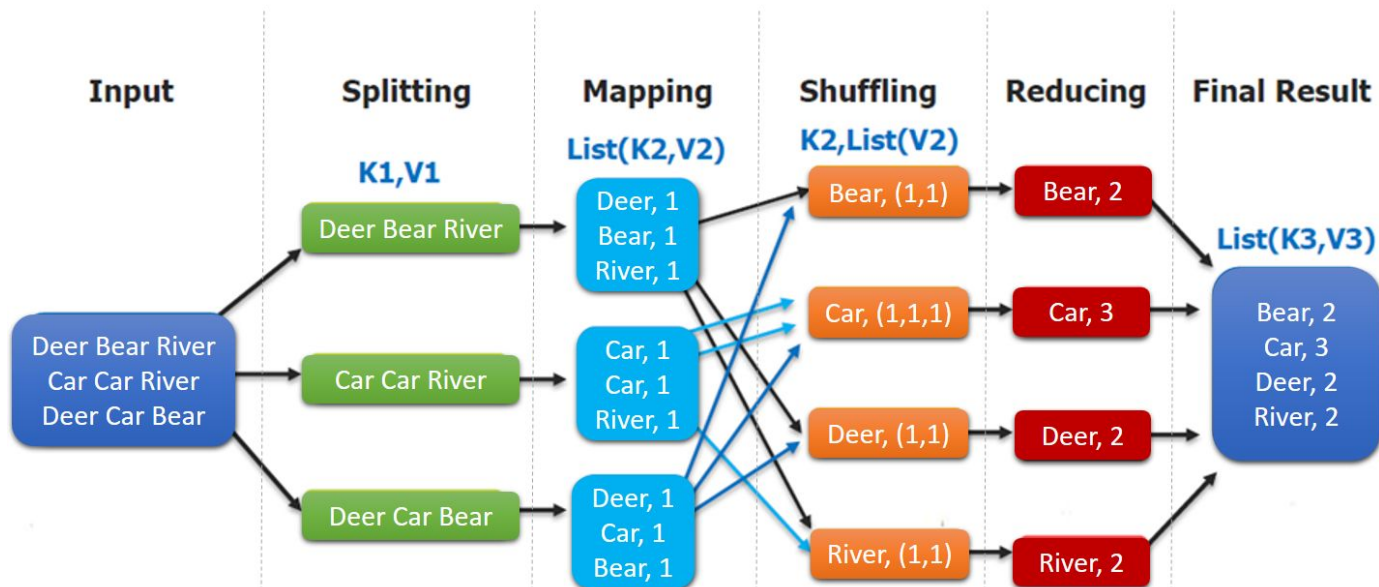
# MapReduce



# MapReduce



# MapReduce



MapReduce Word Count Process



# MapReduce

Рассмотрим пример [WordCount](#)

```
hadoop jar hadoop-streaming.jar \  
-input input_dir\  
-output output_dir\  
-mapper "python mapper.py"\  
-reducer "python reducer.py"\  
-combiner "python reducer.py"\  
-file "mapper.py"\  
-file "reducer.py"
```

```
def map(doc):  
    for word in doc:  
        yield word, 1  
  
def reduce(word, values):  
    yield word, sum(values)
```

# Hive



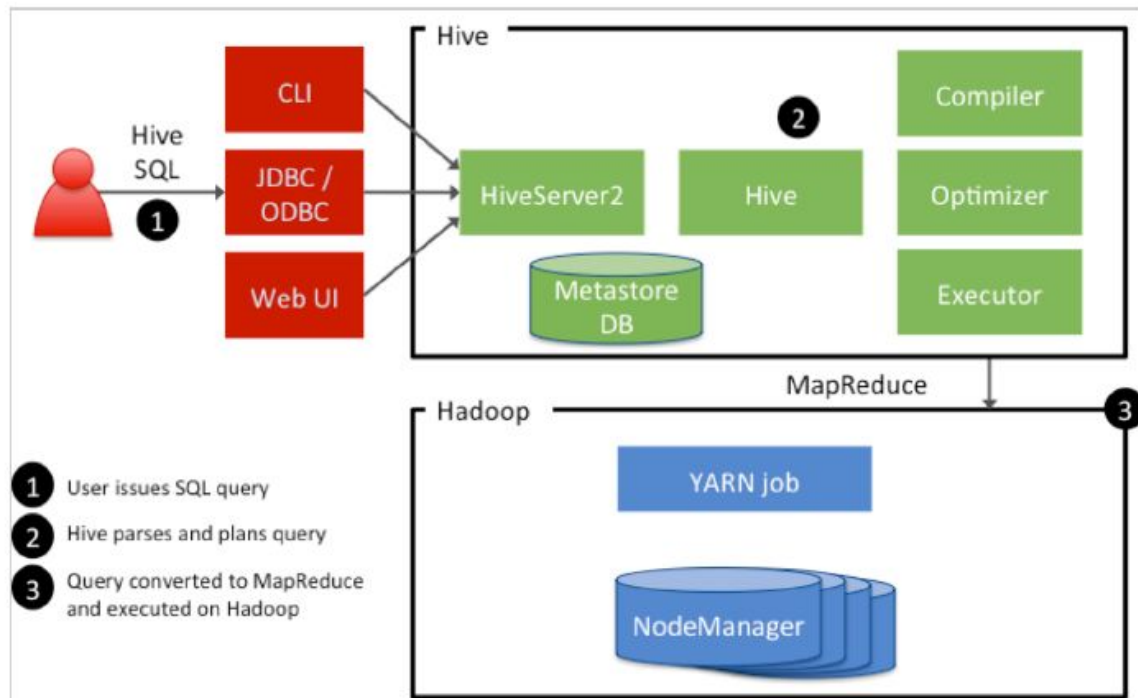
- Позволяет выполнять запросы к слабоструктурированным данным
- Для запросов используется HiveQL
- Имеет свой metastore для хранения “данных о данных” (структур таблиц)



```
CREATE TABLE u_data_new (  
  userid INT,  
  movieid INT,  
  rating INT,  
  weekday INT)  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY '\t';
```

```
SELECT weekday, COUNT(*)  
FROM u_data_new  
GROUP BY weekday;
```

# Архитектура



*Hive Architecture*

Партиции - разбиение данных на уровне файловой системы

SELECT ... WHERE date = '12-2-2009';

date	amount	date	amount	date	amount
11/11/09	6	03/13/08	96	07/12/07	43
06/05/09	12	04/21/08	17	03/02/07	45
...	...	...	...	...	...
12/02/09	8	12/02/08	7	12/02/07	68
Min: 01/01/09 Max: 12/31/09		Min: 01/01/08 Max: 12/31/08		Min: 01/01/07 Max: 12/31/07	

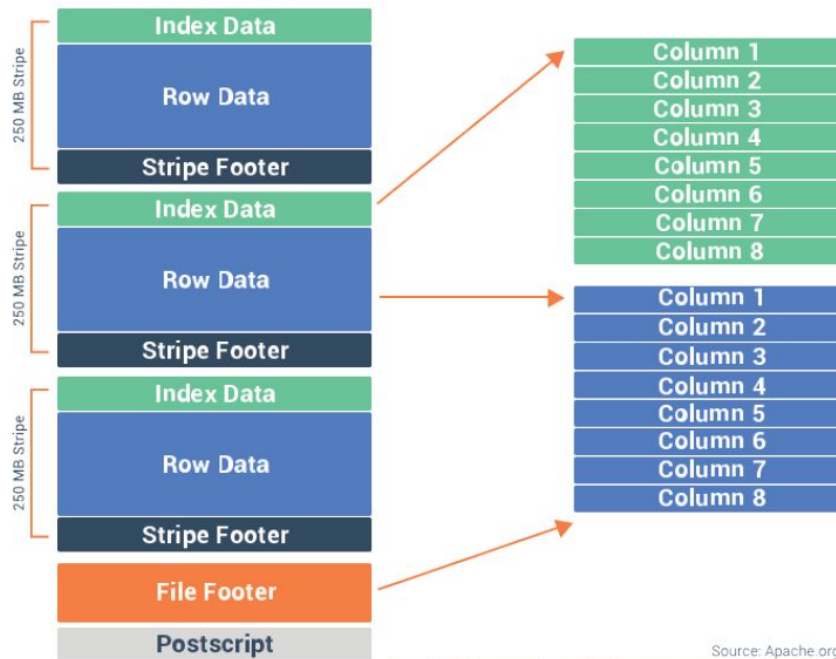
```
CREATE TABLE page_view (  
    viewTime INT, userid BIGINT, page_url STRING  
    ) PARTITIONED BY (dt STRING, country STRING)  
    STORED AS SEQUENCEFILE;
```

```
CREATE TABLE my_table(a string, b BIGINT, ...)
```

- STORED **AS** TEXTFILE;
- STORED **AS** PARQUET;
- STORED **AS** ORC;
- ...

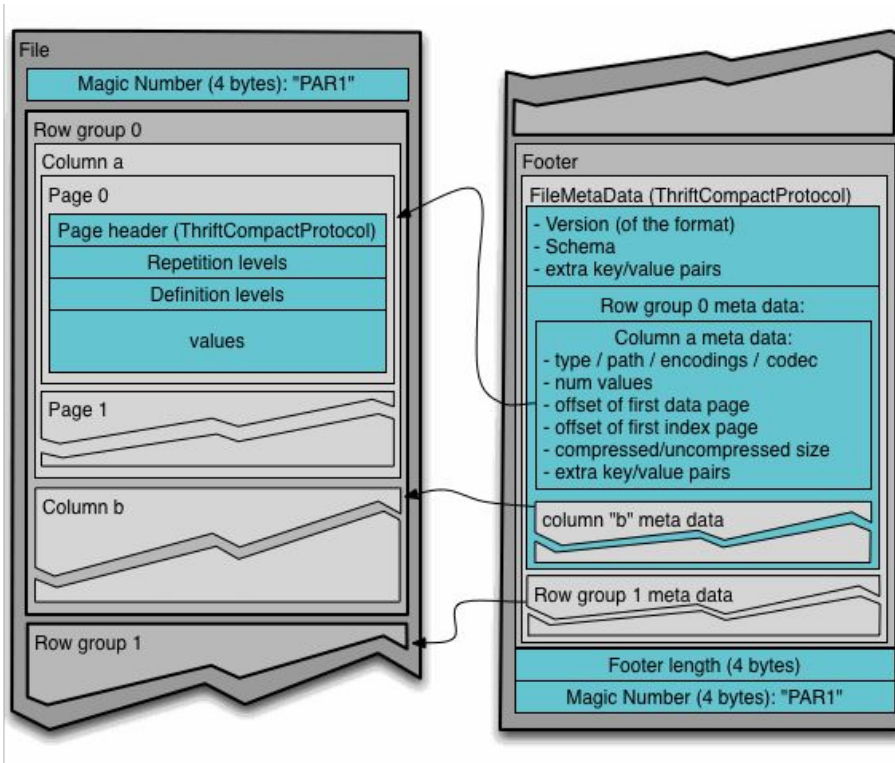
# Columnar formats: Orc

## ORC FILE STRUCTURE



Source: Apache.org,  
<https://cwiki.apache.org/confluence/display/Hive/LanguageManual+ORC>

# Columnar formats: Parquet



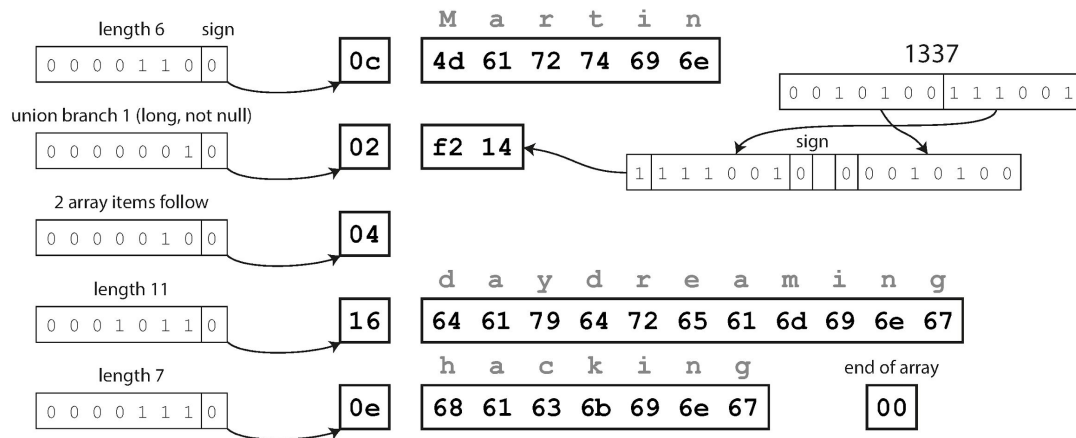


## Avro

Byte sequence (32 bytes):

0c	4d	61	72	74	69	6e	02	f2	14	04	16	64	61	79	64	72	65	61	6d
69	6e	67	0e	68	61	63	6b	69	6e	67	00								

Breakdown:



```
CREATE TABLE my_table(a string, b BIGINT, ...)  
ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'  
STORED AS TEXTFILE;
```

```
CREATE EXTERNAL TABLE page_view (  
    viewTime INT, userid BIGINT, page_url STRING  
)  
STORED AS PARQUET LOCATION '<hdfs_location>';
```



## MSCK REPAIR TABLE

Часто встречающаяся ситуация

- есть внешняя таблица Hive
- туда были записаны данные
- данных не видно в запросах SQL

Поможет команда `MSCK REPAIR TABLE table_name`

С Hive можно взаимодействовать через

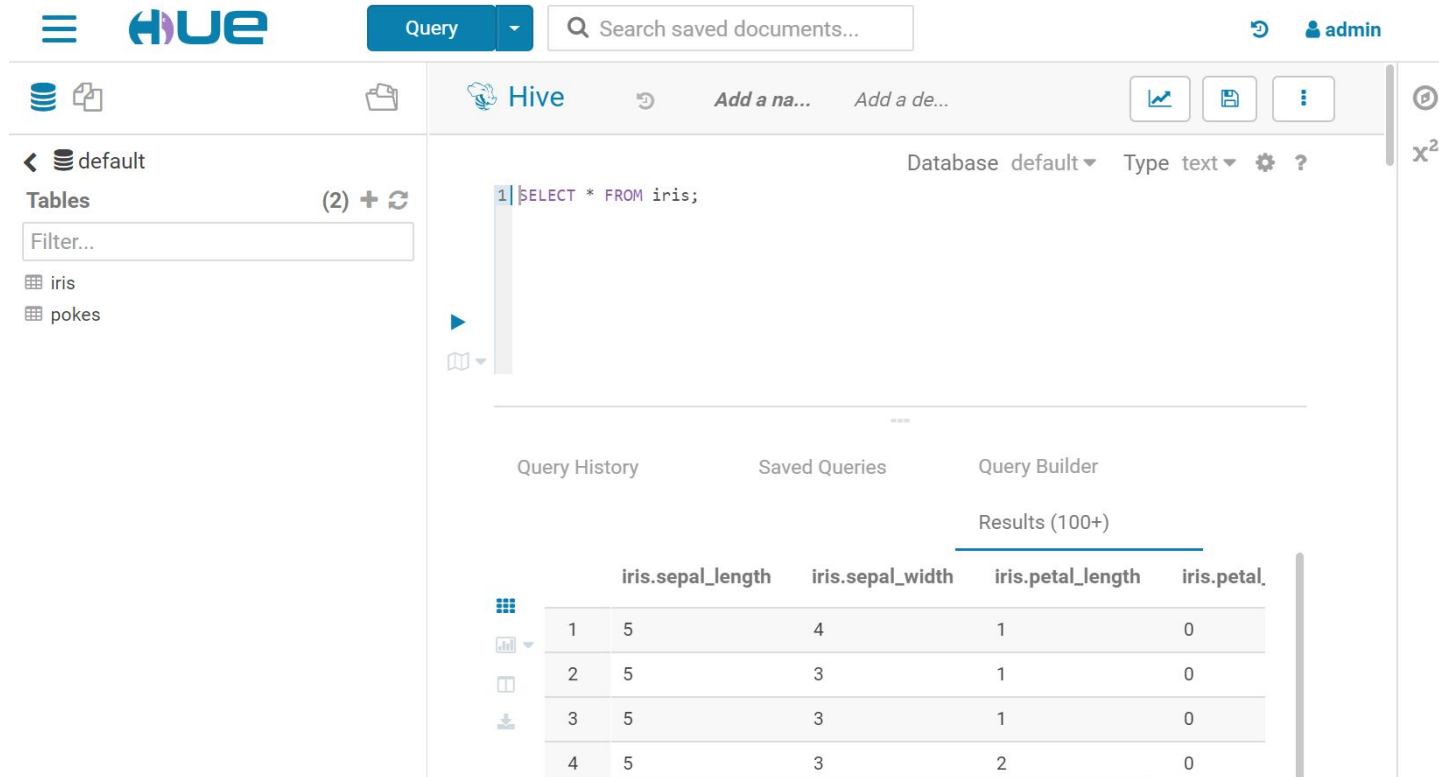
- Beeline CLI
- HUE
- Библиотеки (JDBC, pyhive)

```
% bin/beeline
```

```
beeline> !connect jdbc:hive2://localhost:10000 user1 password1
```

```
0: jdbc:hive2://localhost:10000> show tables;
```

```
+-----+
|  tab_name  |
+-----+
| primitives |
| srcbucket2 |
| srcpart    |
+-----+
```



The screenshot shows the Hue web interface. At the top, there's a navigation bar with the Hue logo, a 'Query' dropdown, a search bar for saved documents, and a user profile for 'admin'. Below this is a sidebar on the left with a 'default' database selected, showing two tables: 'iris' and 'pokes'. The main workspace is divided into a top section for query editing and a bottom section for results. The query editor shows a SQL query: `1 | SELECT * FROM iris;`. Below the query editor, there are tabs for 'Query History', 'Saved Queries', and 'Query Builder'. The 'Results (100+)' tab is active, displaying a table with 4 rows and 5 columns: 'iris.sepal\_length', 'iris.sepal\_width', 'iris.petal\_length', and 'iris.petal.'. The table contains the following data:

	iris.sepal_length	iris.sepal_width	iris.petal_length	iris.petal.
1	5	4	1	0
2	5	3	1	0
3	5	3	1	0
4	5	3	2	0

# Подведем итоги

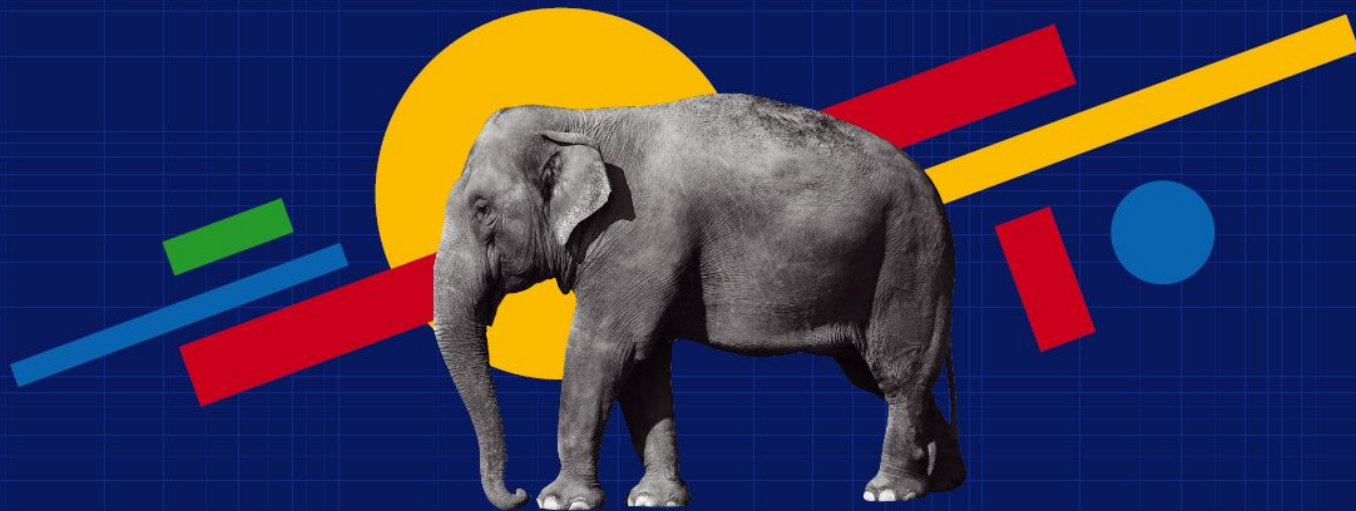


После этого занятия вы будете знать

- Что такое Hadoop
- Из каких основных сервисов он состоит
- Как использовать эти сервисы



**Егор Матешук**  
**egor@mateshuk.com**



# BIG DATA IS LOVE

NEWPROLAB.COM