

Дата-инженер

Экосистема Hadoop – краткий обзор И ВОЗМОЖНОСТИ

Николай Марков



Цели урока. Что вы узнаете:

1 Что такое Hadoop, какие проекты к нему относятся?

2 Как данные хранятся в Hadoop?

3 Как устроен анализ больших данных в Hadoop?

4 Актуален ли Hadoop в современном мире?

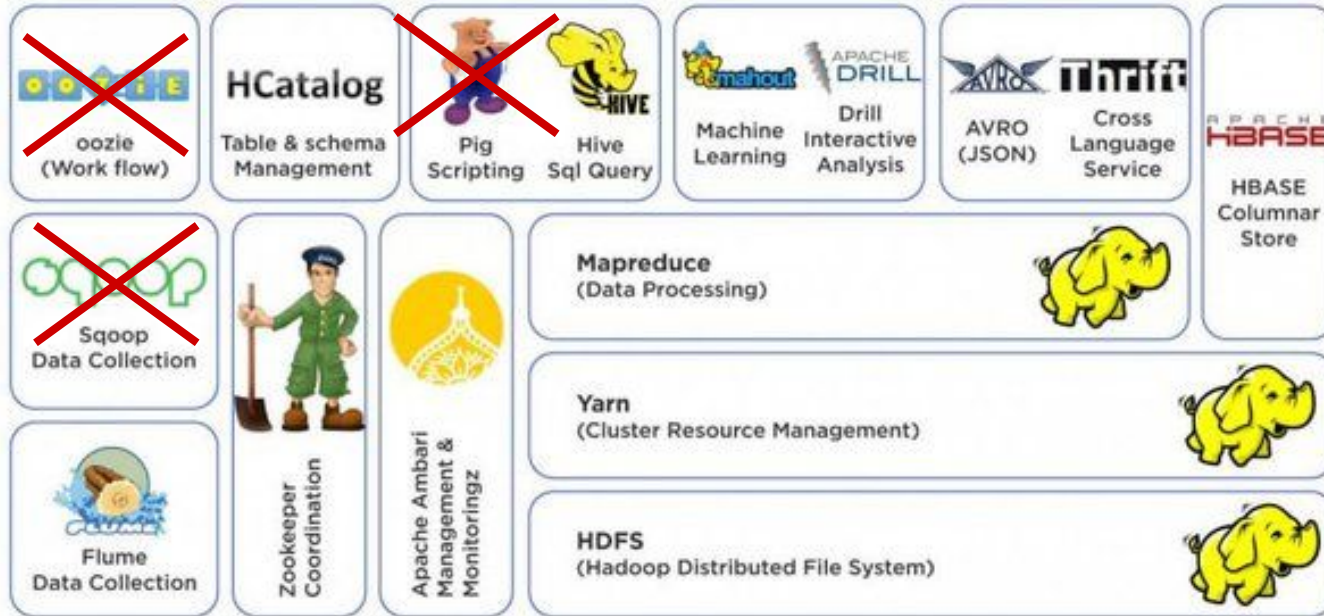


Основные проекты экосистемы Hadoop



Много составных частей

Hadoop Ecosystem



Вендорные решения

- Cloudera, Hortonworks и MapR сидели на трубе
- Hortonworks упало, MapR пропало, кто остался на трубе?

MAPR.



CLOUDERA



Azure HDInsight



Тыкаем галочки

cloudera MANAGER

Clusters ▾ Hosts ▾ Diagnostics ▾ Audits ▾ Charts ▾ Backup ▾ Administration ▾

Search Support admin ▾

Home

Today, 10:10 AM PST

Status All Health Issues Configuration **✖ 36** All Recent Commands

Add Cluster

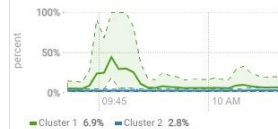
Cluster 1 (CDH 5.11.0, Packages) ▾

- Hosts ✖ 4
- FLUME-1 ▾
- HBASE-1 ▾
- HDFS-1 ✖ 3 ▾
- HIVE-1 ✖ 3 ▾
- HUE-1 ✖ 1 ▾
- IMPALA-1 ▾
- KAFKA-1 ▾
- KMS-1 ▾
- KS_INDEXE... ▾
- KUDU-1 ▾
- MAPREDU... ✖ 2 ▾
- OOZIE-1 ▾
- S3 Connect... ▾
- SENTRY-1 ▾
- SOLR-1 ▾
- SPARK_ON... ▾
- SQOOP-1 ▾
- SQOOP_CL... ▾

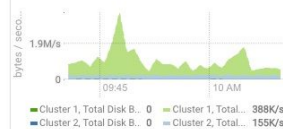
Charts

30m 1h 2h 6h 12h 1d 7d 30d

Cluster CPU



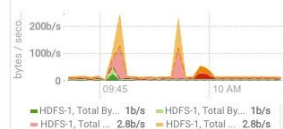
Cluster Disk IO



Cluster Network IO



HDFS IO



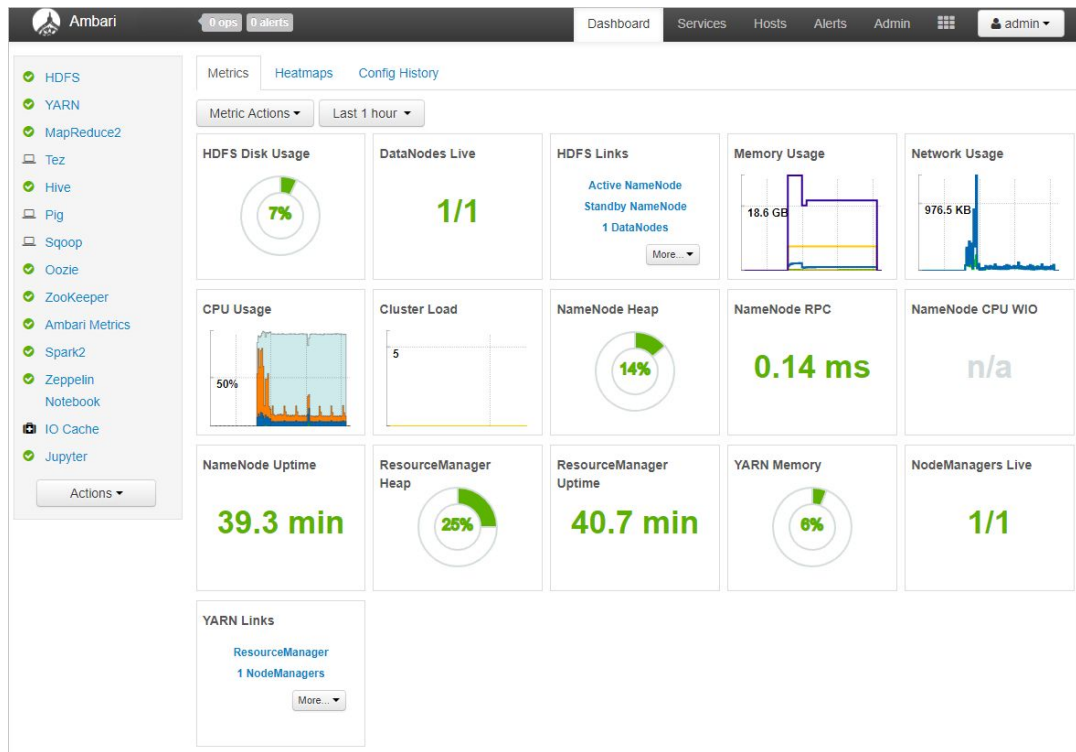
Running MapReduce Jobs



Completed Impala Queries



Надежда Open Source – Apache Ambari



Apache Ambari

Сбор данных по классике

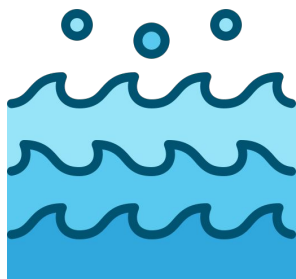
- [Sqoop](#) – пример проекта, в котором новую версию **официально упразднили** в пользу старой
- SQL-запросами **забирает** данные из источника и **заливает в HDFS**
- Сейчас часто используют **Apache Spark**



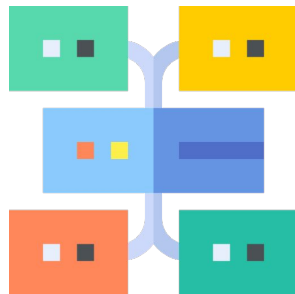
SQOOP

Что сейчас актуально?

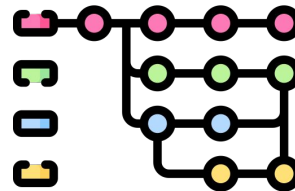
Apache Flume



Поддерживает
стриминг в
реальном времени

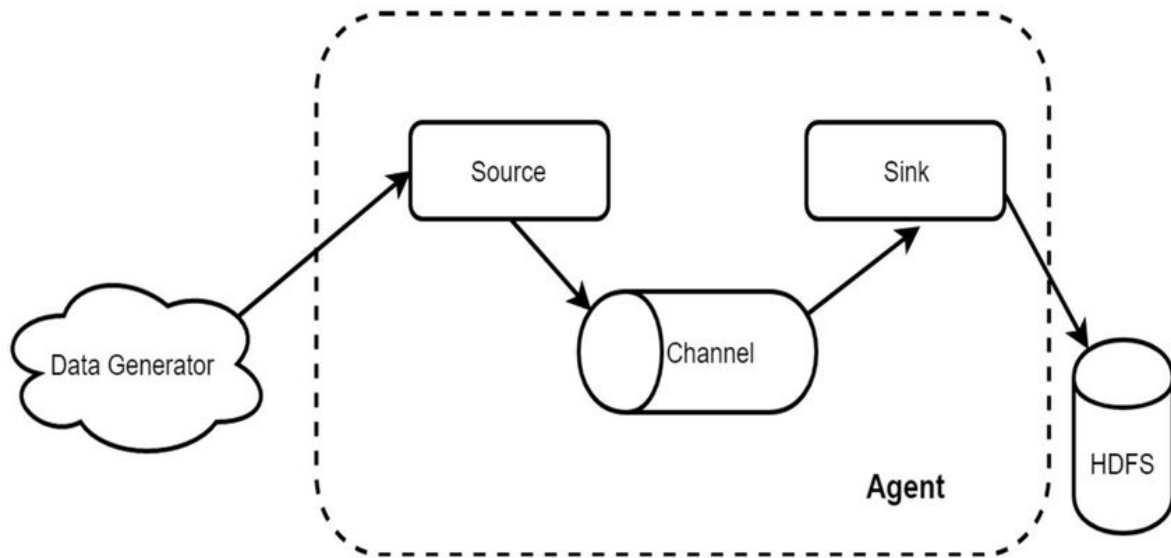


Коннекторы
на **вход**
и **выход**



До сих пор активно
развивается

Архитектура Apache Flume



[Apache NiFi](#)

[Vector](#)

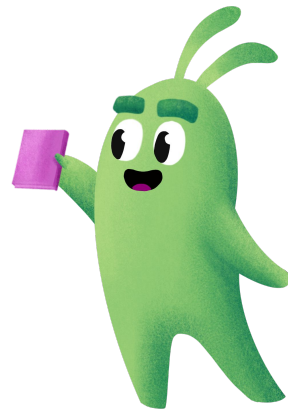
Управление зоопарком



APACHE

ZooKeeper™

Как данные хранятся в Hadoop



Архитектура HDFS

Использует те же идеи, которые лежали в основе [Google File System](#)

Минусы

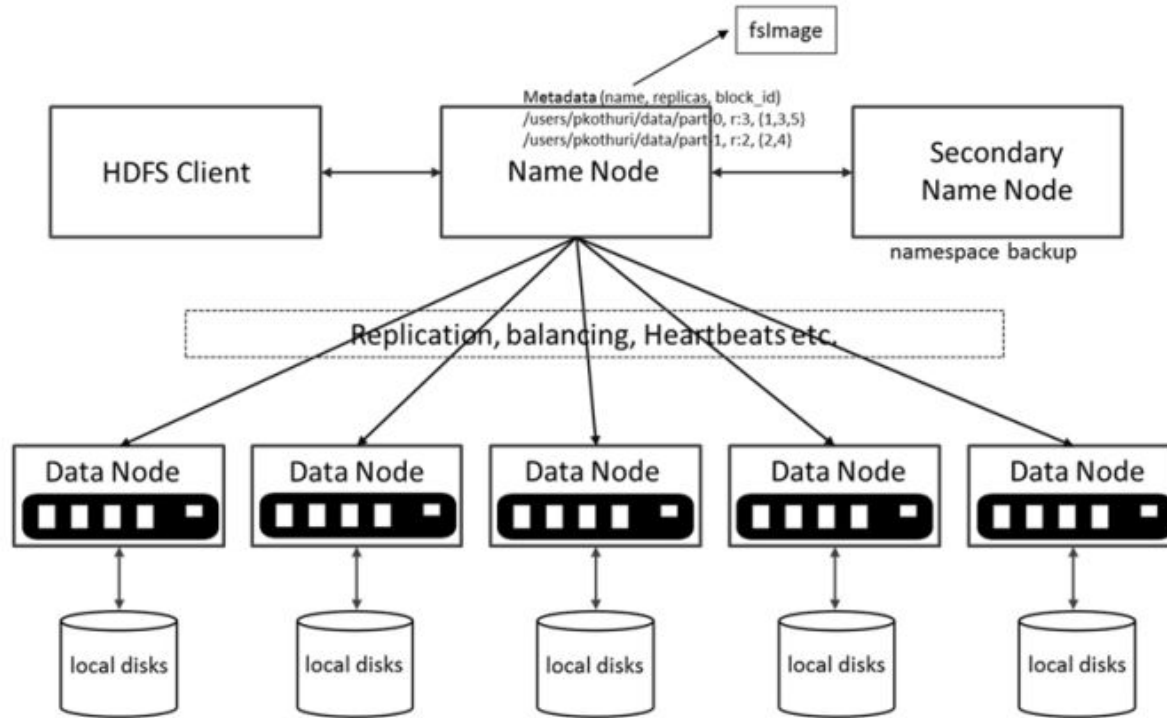
все упирается в одну ноду
с метаданными,
не поддерживает
совместимые драйвера
файловых систем

Плюсы

дает возможность
использовать преимущества
Data Locality

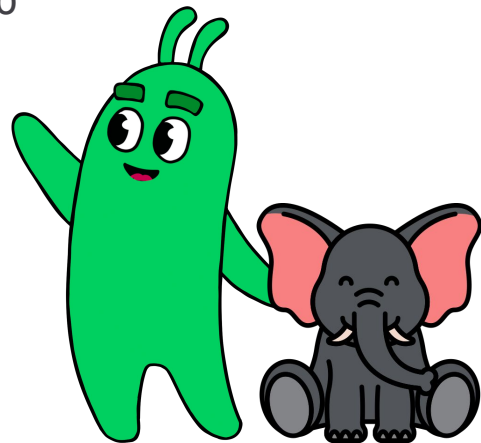


NameNode, SNameNode, DataNode



Чуть больше про устройство HDFS

- Файлы нарезаются на **блоки** размером **128Мб** (по умолчанию)
- Каждый блок **реплицируется** столько раз, сколько задано в **replication factor** (по умолчанию 3)
- HDFS поддерживает [Rack Awareness](#), имеет настраиваемую [Replica Placement Policy](#)
- HDFS **плохо** подходит для хранения **мелких файлов** (много **горизонтальной пересылки** метаданных)



Хранение данных колоночно

- NoSQL база данных, работает **поверх HDFS**, используя свой формат **HFile**
- Построен по принципу, что и **Google BigTable**
- Таблицы не имеют схемы
- Можно задавать **column families** для более эффективной кластеризации
- Значения версионизируются и физически сортируются



Кодируем данные – Parquet, Avro, Thrift



Хранить данные **at rest**
в **колоночном**
формате для
запросов

На забывайте про [ORC](#) и
[Iceberg](#)

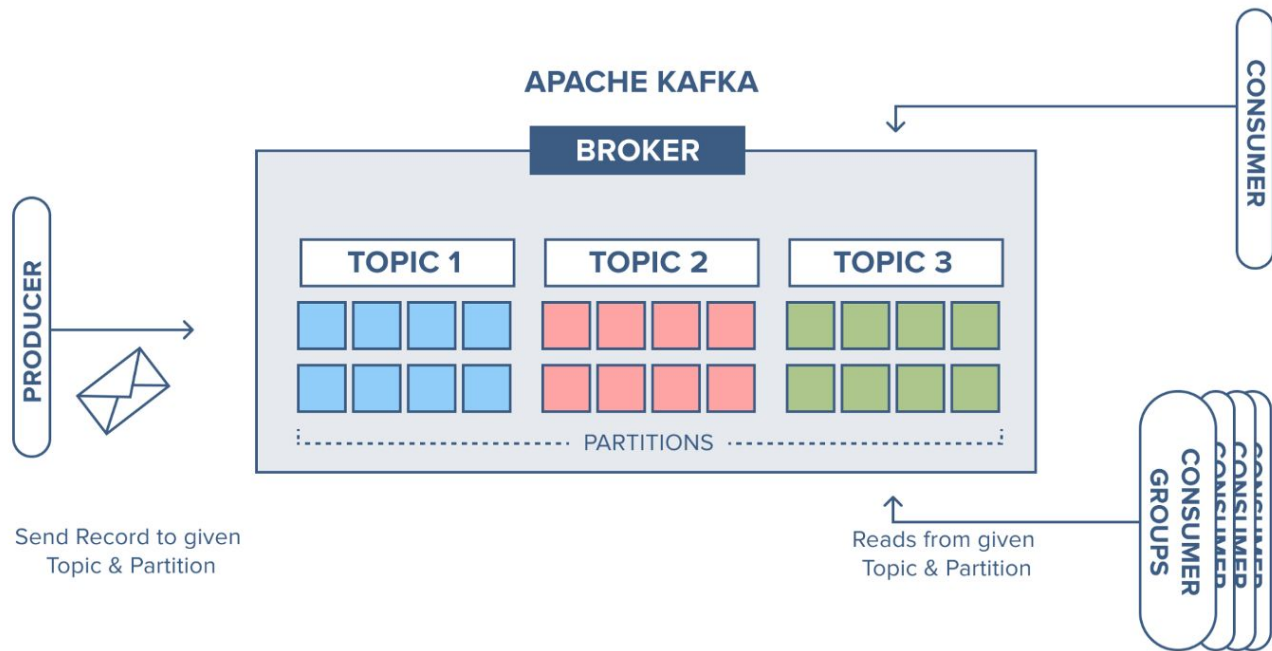


Пересылать данные
on wire в **JSON-**
совместимом
формате

Apache
Thrift™

Выполнять **remote**
procedure call
из одного сервиса
в другой

Потоковые данные – Kafka, Pulsar



Сравним

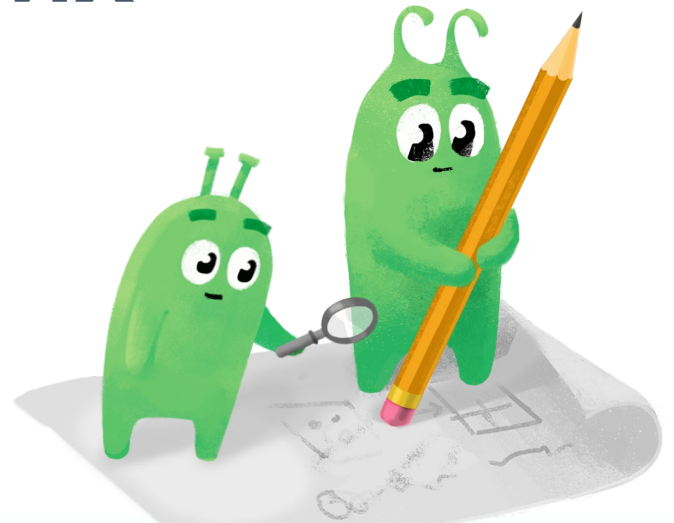
Kafka

- Kafka Streams умеет мало
- Брокеры – **stateful**
- Умеет **компактизировать топики** при долгосрочном хранении

Pulsar

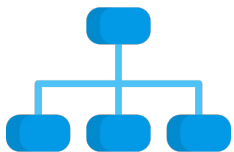
- Делает базовые **расчеты на данных** самостоятельно
- Брокеры – **stateless**
- Pulsar поддерживает **гео-репликацию**

Анализ больших данных



YARN

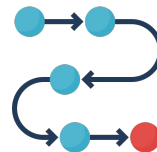
- В первой версии Hadoop менеджмент ресурсов осуществлялся самим фреймворком **MapReduce**
- Во второй версии **Yet Another Resource Negotiator** выделился в отдельный проект
- Был еще **Mesos**, но сплыл



Можно
масштабировать
кластер на 10 000+
машин



Совместимость
с другими
фреймворками



Динамическое
выделение ресурсов



Выбираем данные – Hive, Impala

- Hive менялся и поддерживал разные **движки запросов**, начиная с **MapReduce** и заканчивая **Spark**
- **Impala** была разработана в Cloudera, как «улучшенный Hive» **на C++**
–
с поддержкой **HiveQL, YARN**
- Hive - ниже **время старта** и **latency**, но выше **throughput** и **отказоустойчивость**
- **Hive Metastore** – полезный компонент, который может использовать сторонние фреймворки

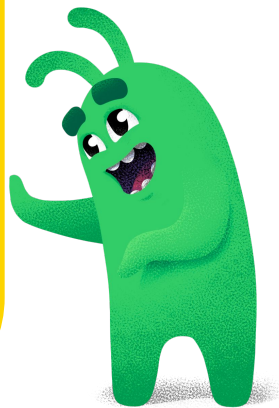
[Узнать больше о сравнении](#)



Виртуализация данных – Trino



trino



APACHE
PHOENIX

APACHE
DRILL

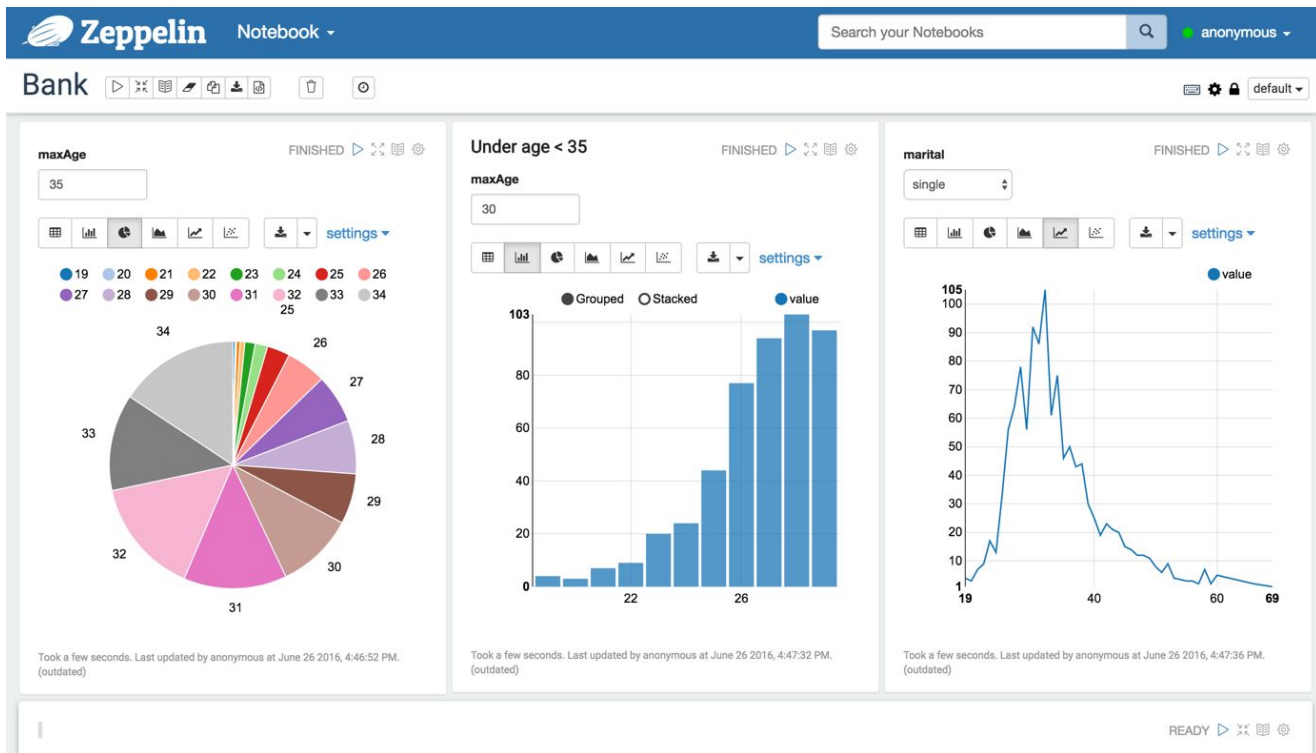
Распределенные вычисления – Spark, Tez

- Tez разрабатывался раньше и **теснее интегрирован** в Hadoop-инфраструктуру, например, тот же **YARN**
- Оба **умеют держать данные в оперативке**
- Tez не требует держать контейнеры **запущенными и занимать ресурсы**
- Spark более популярен и имеет расширения в лице **GraphX** и **MLlib**

[Почитать про сравнение](#)



Смотрим глазами – Hue, Zeppelin



Охраняем данные – [Keycloak](#), [Knox](#), [Ranger](#)

- **Keycloak** используется в кластере для авторизации и аутентификации
- Не все сервисы Hadoop поддерживают эти механизмы самостоятельно. **Knox** – дополнительный **API Gateway** для слоя защиты
- **Apache Ranger** дает интерфейс для управления доступами и RBAC, в том числе с поддержкой **LDAP**



Что осталось за бортом



Apache Pig



APACHE™

Camel



MAHOUT



APACHE
STORM™

Как поживает Hadoop в 2023 году?

HDFS используется как основа для Data Lake

Вендорные пакеты реже используют - почти любой компонент можно заменить

Большие MPP-базы и движки работают быстрее запросов поверх HDFS, но они сложнее в поддержке



Итоги. О чем поговорили:

1

Какие проекты есть в экосистеме Hadoop и какие проблемы они решают

2

Как в Hadoop хранятся данные и какие форматы лучше использовать

3

Какие проекты помогают делать запросы к данным в HDFS

4

Что из экосистемы устарело и не рекомендуется к использованию

5

Насколько актуален Hadoop прямо сейчас





**Спасибо
за внимание!**

