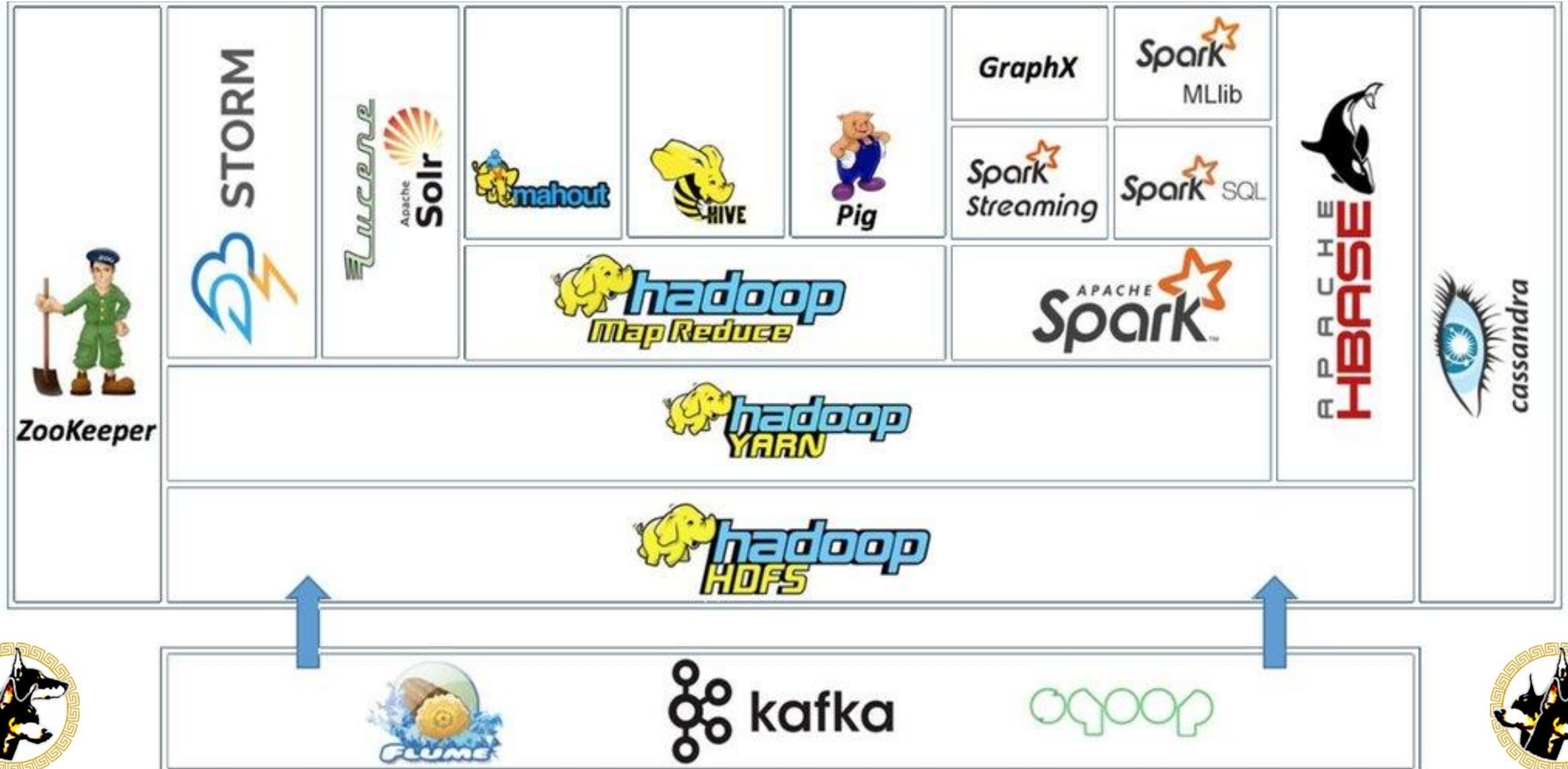




# Загрузка данных

ETL vs ELT, Batch vs Streaming, Sqoop, Flume

# Экосистема Hadoop



# ВИДЫ ПОЛУЧЕНИЯ ДАННЫХ

# **Виды получения данных**

## **ETL**

- 1.Extract**
- 2.Transform**
- 3.Load**

# **Виды получения данных**

## **ETL**

- 1.Extract**
- 2.Transform**
- 3.Load**

## **ELT**

- 1.Extract**
- 2.Load**
- 3.Transform**

# Виды получения данных

## ETL

- 1.Extract
- 2.Transform
- 3.Load

## ELT

- 1.Extract
- 2.Load
- 3.Transform

## Batch

- T-1
- Micro-batch

# Виды получения данных

## ETL

- 1.Extract
- 2.Transform
- 3.Load

## ELT

- 1.Extract
- 2.Load
- 3.Transform

## Batch

- T-1
- Micro-batch

## Streaming

- NRT (near real time)
- Real time

# ИНСТРУМЕНТЫ РАБОТЫ С ДАННЫМИ



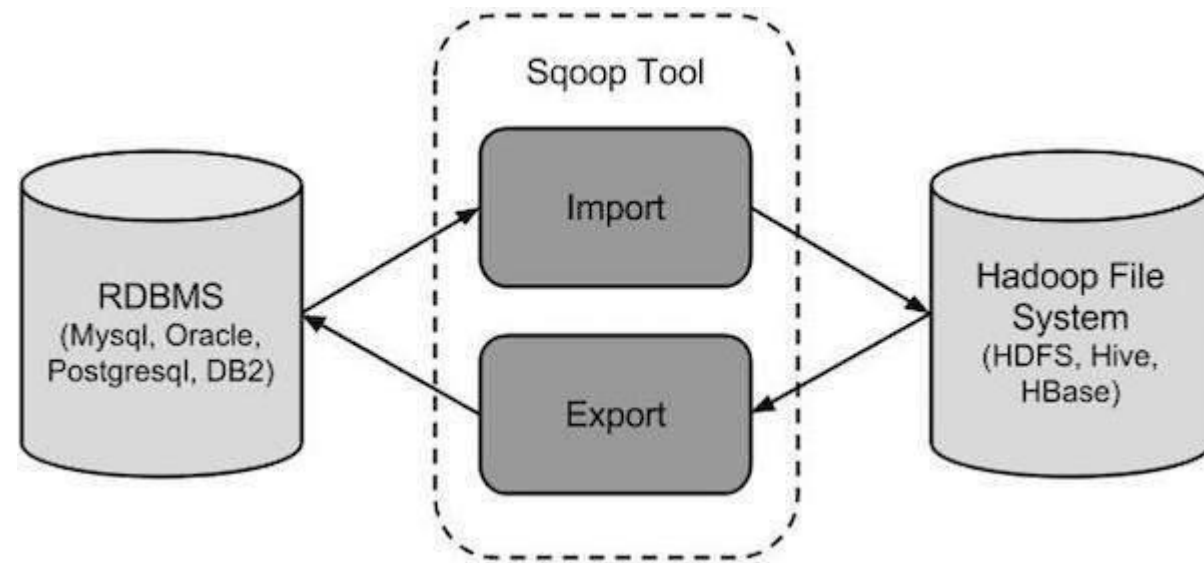
# Инструменты

1. HDFS (hdfs dfs -get, distcp)
2. Apache SQOOP
3. ODBC/JDBC Connectors
4. Informatica
5. Oracle Data Integrator
6. Apache NiFi
7. Apache Flume
8. Apache Spark
9. Apache Flink
10. Apache Kafka
11. Apache Storm
12. Apache Samza
13. ...

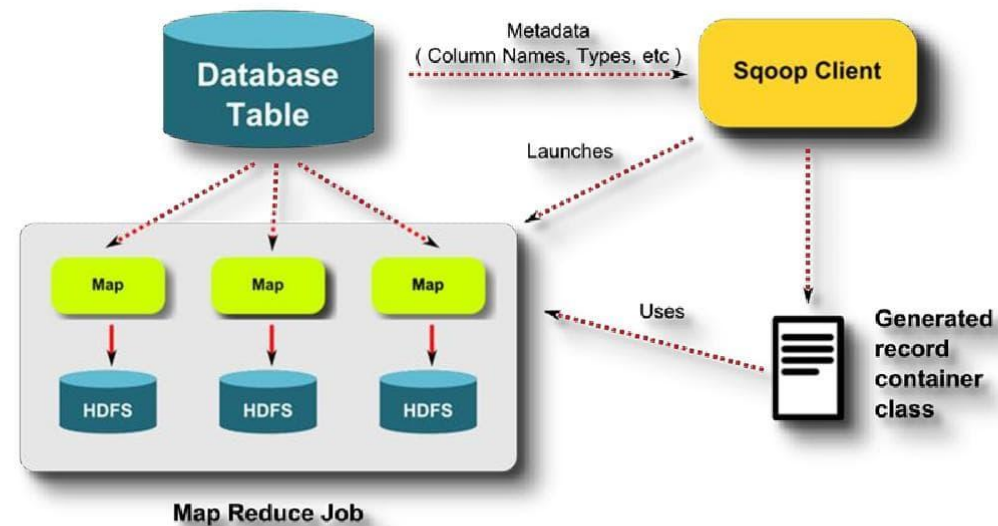
# SQOOP



# 02



## SQOOP Architecture



# ODBC/JDBC Connectors

03

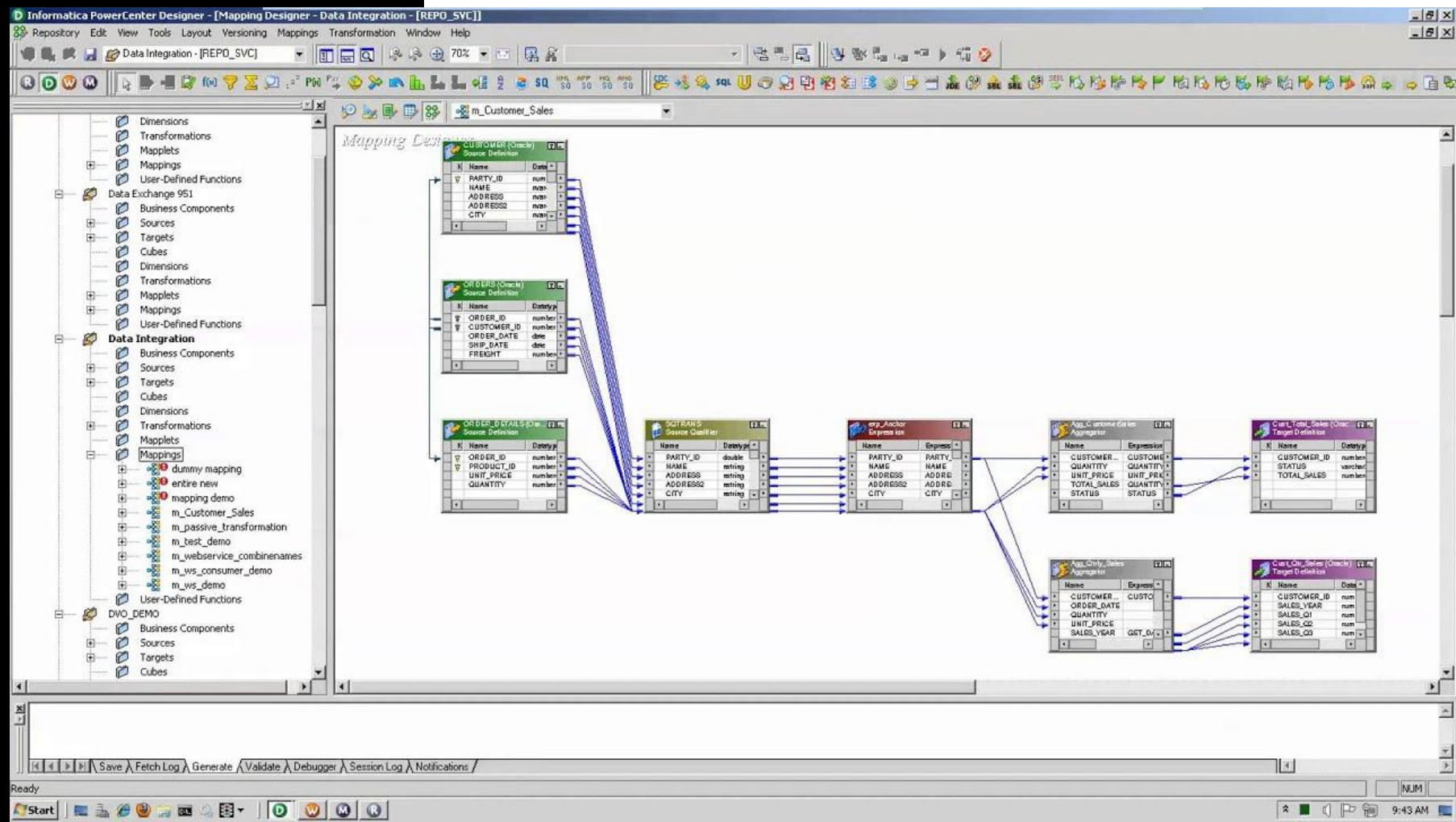
1. Медленно
2. Универсально

# Informatica



# 04

1. Якобы ускоряет разработку
2. Якобы легче, чем код



# Oracle Data Integrator

ORACLE®  
DATA INTEGRATOR

The screenshot displays the Oracle Data Integrator (ODI) Designer interface. The main workspace shows a logical mapping diagram with the following components:

- CUS** (Source): A table with attributes: CUST\_KEY, FST\_CONTACT\_DT, SEGMENT\_KEY, INCOME\_LVL, STATUS\_KEY, LST\_ORDER\_DT, ADDRESS\_KEY, MARITAL\_ST, PREV\_MARITAL\_ST, and PREV\_MARITAL\_ST.
- LOOKUP** (Operator): A circular operator connecting the CUS source to the JOIN operator.
- JOIN** (Operator): A circular operator connecting the LOOKUP operator to the CUSTOMERADDRESSRELATIONS target.
- CUSTOMERADDRESSRELATIONS** (Target): A table with attributes: CUSTOMERID and ADDRESSID.
- SAMP\_PRODUCTS\_D** (Source): A table with attributes: PROD\_KEY, PROD\_DSC, ATTRIBUTE\_2, ATTRIBUTE\_1, TYPE, LOB, BRAND, SEQUENCE, BRAND\_KEY, and LOB\_KEY.
- SAMP\_REVENUE\_F** (Source): A table with attributes: SHIPTO\_ADDR\_KEY, OFFICE\_KEY, EMPL\_KEY, PROD\_KEY, ORDER\_NUMBER, REVENUE, UNITS, DISCNT\_VALUE, BILL\_MTH\_KEY, and BILL\_QTR\_KEY.
- FILTER** (Operator): A circular operator connecting the SAMP\_PRODUCTS\_D and SAMP\_REVENUE\_F sources to the JOIN operator.

The left sidebar shows the project structure:

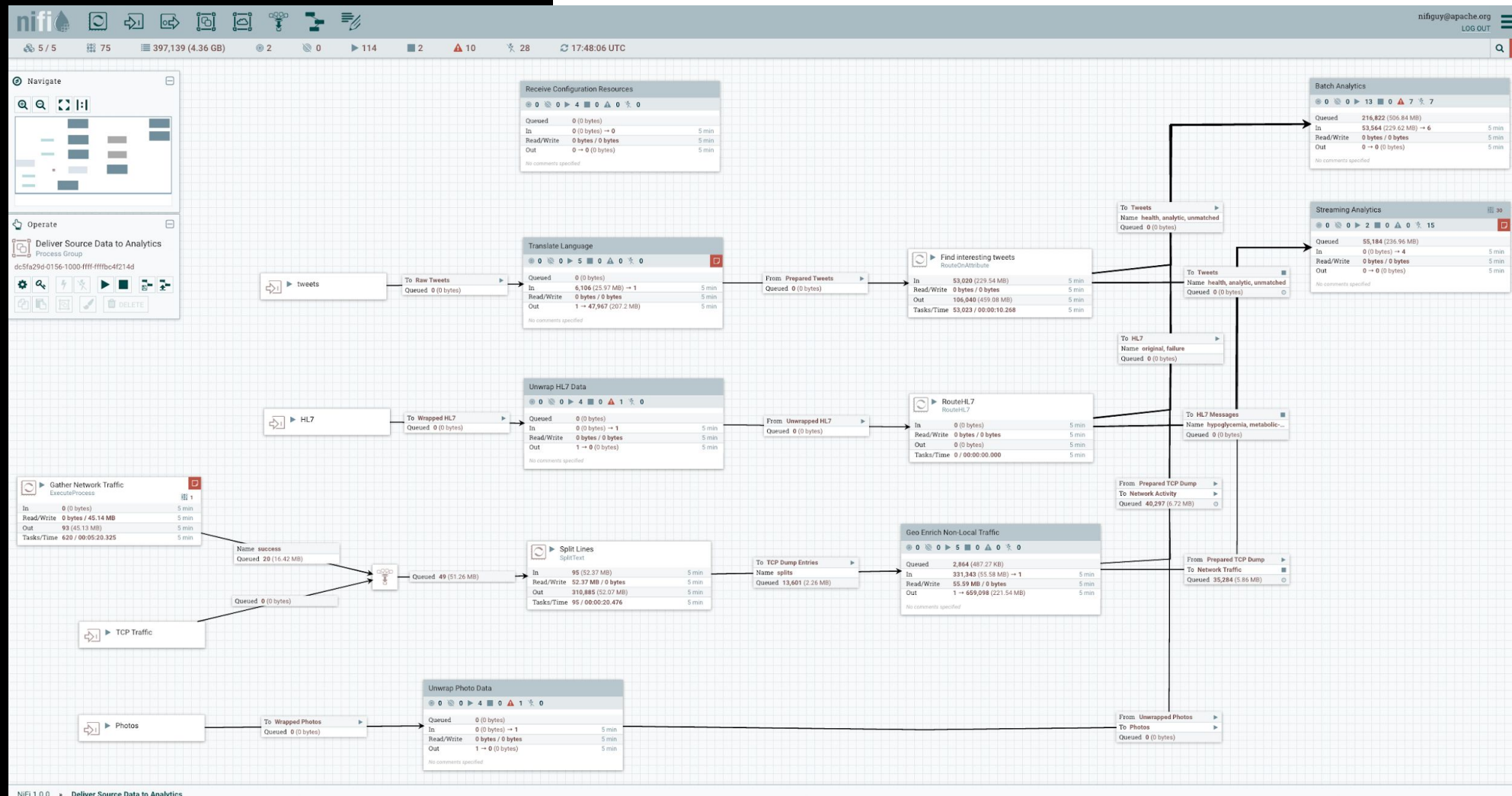
- Projects
  - ODI for Data Warehousing
    - ODI for Data Warehousing
      - Packages
        - Load Customers Data
        - Load Customers Data - MySQL Source
        - New\_Mapping
        - Test 1
        - Reusable Mappings
        - Procedures
      - hidden
        - Variables
        - Sequences
        - User Functions
        - Knowledge Modules
- Models
  - src
    - SAMP\_PRODUCTS\_D
    - SAMP\_REVENUE\_F
    - SRC\_CUSTOMERS
    - Hidden Datastores
  - SRC - COMPLEX FILE
  - SRC - FILES
  - SRC - MySQL
    - Uses
    - Diagrams
    - Hierarchy
    - SAMP\_ADDRESSES\_D
    - SAMP\_PRODUCTS\_D
    - SRC\_CUSTOMERS
    - SRC\_PRODUCTS
    - SRC\_REVENUE

The bottom right pane shows the **CUS - Properties** dialog box:

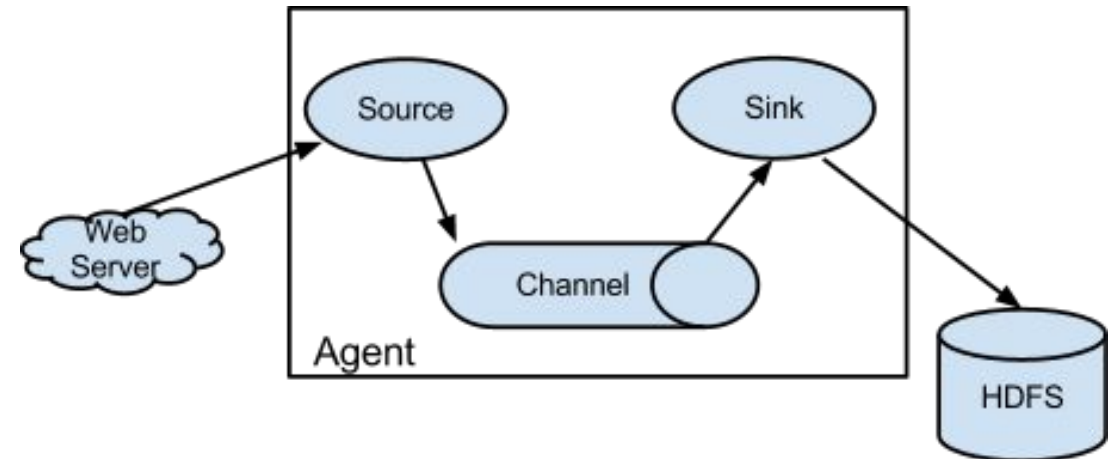
- General** tab is selected.
- Name:** CUS
- Description:** (empty)
- Partition/Sub-Partition:** (empty)
- Datastore:** MYSQL.SRC\_CUSTOMERS
- Shortcut:** (empty)
- Logical Schema:** MYSQL\_LogicalSchema
- Component Context (Forced):** <Execution Context>



# Apache NiFi



# Apache Flume

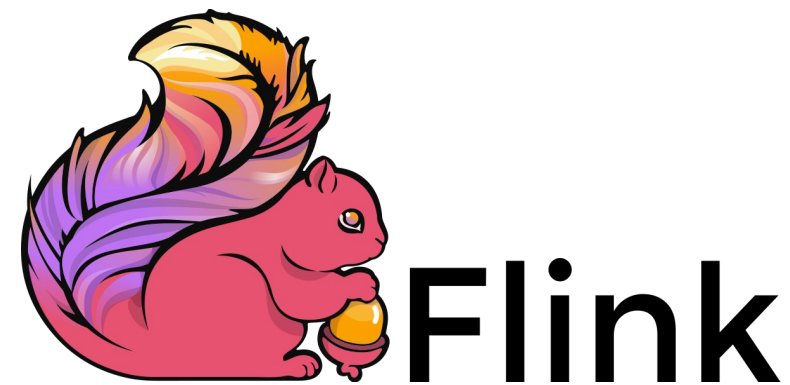


# Apache Spark



Умеет все, но с небольшой задержкой

# Apache Flink



Тот же Spark, только быстрее.  
Минусы умалчиваются



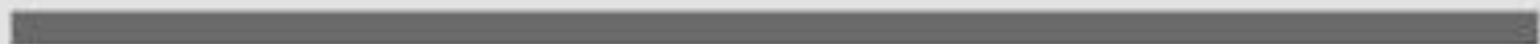
# Apache Kafka



Не только транспорт, но еще и  
процессинг

ПЕРЕРЫВ

**10:00**



# ПРАКТИКА

# ПОДНИМАЕМ ЛОКАЛЬНЫЙ КЛАСТЕР

```
docker start -i gbhdp
```

# APACHE SQOOP

# УСТАНОВКА SQOOP

Скачиваем и распаковываем дистрибутив:

```
$ wget http://archive.apache.org/dist/sqoop/1.4.7/sqoop-1.4.7.bin\_hadoop-2.6.0.tar.gz
$ tar xzf sqoop-1.4.7.bin_hadoop-2.6.0.tar.gz
$ rm sqoop-1.4.7.bin_hadoop-2.6.0.tar.gz
$ mv sqoop-1.4.7.bin_hadoop-2.6.0 sqoop
```

Задаем необходимые переменные окружения:

```
$ cd sqoop
$ export SQOOP_HOME=`pwd`
$ export PATH=$PATH:$SQOOP_HOME/bin
$ export HADOOP_MAPRED_HOME=$HADOOP_HOME
$ export HADOOP_COMMON_HOME=$HADOOP_HOME
```

Проверяем работу:

```
$ sqoop-version

Sqoop 1.4.7
git commit id 2328971411f57f0cb683dfb79d19d4d19d185dd8
Compiled by maugli on Thu Dec 21 15:59:58 STD 2017
```

# ПРИМЕРЫ SQOOP

Копируем зависимости из Hive:

```
$ cp $HIVE_HOME/jdbc/hive-jdbc-2.3.9-standalone.jar $SQOOP_HOME/lib/
```

Смотрим таблицы в Hive:

```
$ sqoop list-tables \  
  --connect jdbc:hive2://localhost:10000 \  
  --driver org.apache.hive.jdbc.HiveDriver
```



# ПРИМЕРЫ SQOOP

Экспортируем данные из MySQL в HDFS:

```
$ sqoop import \  
  --connect jdbc:mysql://localhost/mybbdd \  
  --username=root -P \  
  --table=mytable \  
  --driver=com.mysql.jdbc.Driver \  
  --where 'id > 40' \  
  --target-dir=/sqoop_snappy_avro \  
  --compress \  
  --compression-codec org.apache.hadoop.io.compress.SnappyCodec \  
  --as-avrodatafile
```

# ПРИМЕРЫ SQOOP

Экспортируем данные из MySQL в Hive:

```
$ sqoop import \  
  --connect jdbc:mysql://localhost/mybbdd \  
  --username=root -P \  
  --table=mytable \  
  --driver=com.mysql.jdbc.Driver \  
  --where 'id > 40' \  
  --hive-import \  
  --create-hive-table \  
  --hive-table ej_hive_table
```

# APACHE FLUME

# УСТАНОВКА FLUME

Скачиваем и распаковываем дистрибутив:

```
$ wget https://mirror.softaculous.com/apache/flume/1.9.0/apache-flume-1.9.0-bin.tar.gz
$ tar xzf apache-flume-1.9.0-bin.tar.gz
$ rm apache-flume-1.9.0-bin.tar.gz
$ mv apache-flume-1.9.0-bin flume
```

Задаем необходимые переменные окружения:

```
$ cd flume
$ export FLUME_HOME=`pwd`
$ export PATH=$PATH:$FLUME_HOME/bin
```

Проверяем работу:

```
$ flume-ng version

Flume 1.9.0
Source code repository: https://git-wip-us.apache.org/repos/asf/flume.git
Revision: d4fcab4f501d41597bc616921329a4339f73585e
Compiled by fszabo on Mon Dec 17 20:45:25 CET 2018
From source with checksum 35db629a3bda49d23e9b3690c80737f9
```

# ЗАПУСК FLUME

Создадим файл heartbeat.sh:

```
START_DATE=`date`  
COUNT=0  
while [ true ]  
do  
    NOW_DATE=`date`  
    echo I live for $(( (`date -d "$START_DATE" +%s` - `date -d "$NOW_DATE" +%s`) / (1000) ))  
seconds\;`(date -d "$START_DATE" +%Y-%m-%d:%H.%M.%S)\;`(date -d "$START_DATE" +%Y-%m-%d:%H.%M.%S)\;I  
did it $(( $COUNT + 1 )) times  
    COUNT=$(( $COUNT + 1 ))  
    sleep 10  
done
```

# ЗАПУСК FLUME

Создадим конфигурацию heartbeat.conf:

```
# Naming the components on the current agent
HeartbeatFlume.sources = Exec
HeartbeatFlume.channels = MemChannel
HeartbeatFlume.sinks = MySink

# Describing/Configuring the source
HeartbeatFlume.sources.Exec.type = exec
HeartbeatFlume.sources.Exec.command = /home/hduser/heartbeat.sh

# Describing/Configuring the console sink
HeartbeatFlume.sinks.MySink.type = hdfs
HeartbeatFlume.sinks.MySink.hdfs.path = /tmp/heartbeat
HeartbeatFlume.sinks.MySink.hdfs.fileSuffix = .log

# Describing/Configuring the channel
HeartbeatFlume.channels.MemChannel.type = memory
HeartbeatFlume.channels.MemChannel.capacity = 100000000
HeartbeatFlume.channels.MemChannel.transactionCapacity = 100

# Bind the source and sink to the channel
HeartbeatFlume.sources.Exec.channels = MemChannel
HeartbeatFlume.sinks.MySink.channel = MemChannel
```

# ЗАПУСК FLUME

Запустим приложение:

```
$ flume-ng agent \  
  --conf-file /home/hduser/heartbeat.conf \  
  --name HeartbeatFlume \  
  -Dflume.root.logger=INFO,console &> /dev/null &
```

# ИМПОРТ ДАННЫХ FLUME

Создаем таблицу данных Flume с указанием внешней директории:

```
CREATE TABLE heartbeat(c1 STRING, c2 STRING, c3 STRING, c4 STRING)  
  
ROW FORMAT DELIMITED  
  
FIELDS TERMINATED BY ';'   
  
STORED AS INPUTFORMAT 'org.apache.hadoop.mapred.SequenceFileInputFormat'  
  
OUTPUTFORMAT 'org.apache.hadoop.hive ql.io.HiveSequenceFileOutputFormat'  
  
LOCATION '/tmp/heartbeat';
```



# ПРОСМОТР ДАННЫХ FLUME

```
SELECT * FROM heartbeat;
```

# APACHE NIFI

# УСТАНОВКА NIFI

Скачиваем и распаковываем дистрибутив:

```
$ wget https://dlcdn.apache.org/nifi/1.15.3/nifi-1.15.3-bin.tar.gz
$ tar xzf nifi-1.15.3-bin.tar.gz
$ rm nifi-1.15.3-bin.tar.gz
$ mv nifi-1.15.3 nifi
```

Задаем необходимые переменные окружения:

```
$ cd nifi
$ export NIFI_HOME=`pwd`
$ export PATH=$PATH:$NIFI_HOME/bin
```

Проверяем работу:

```
$ nifi.sh status

Apache NiFi is not running
```

# ЗАПУСК NIFI

Редактируем конфиг:

```
$ vi nifi/conf/nifi.properties  
...  
nifi.web.https.host=0.0.0.0  
nifi.web.https.port=8888  
...
```

Запускаем NiFi:

```
$ nifi.sh start
```

Создаем пользователя:

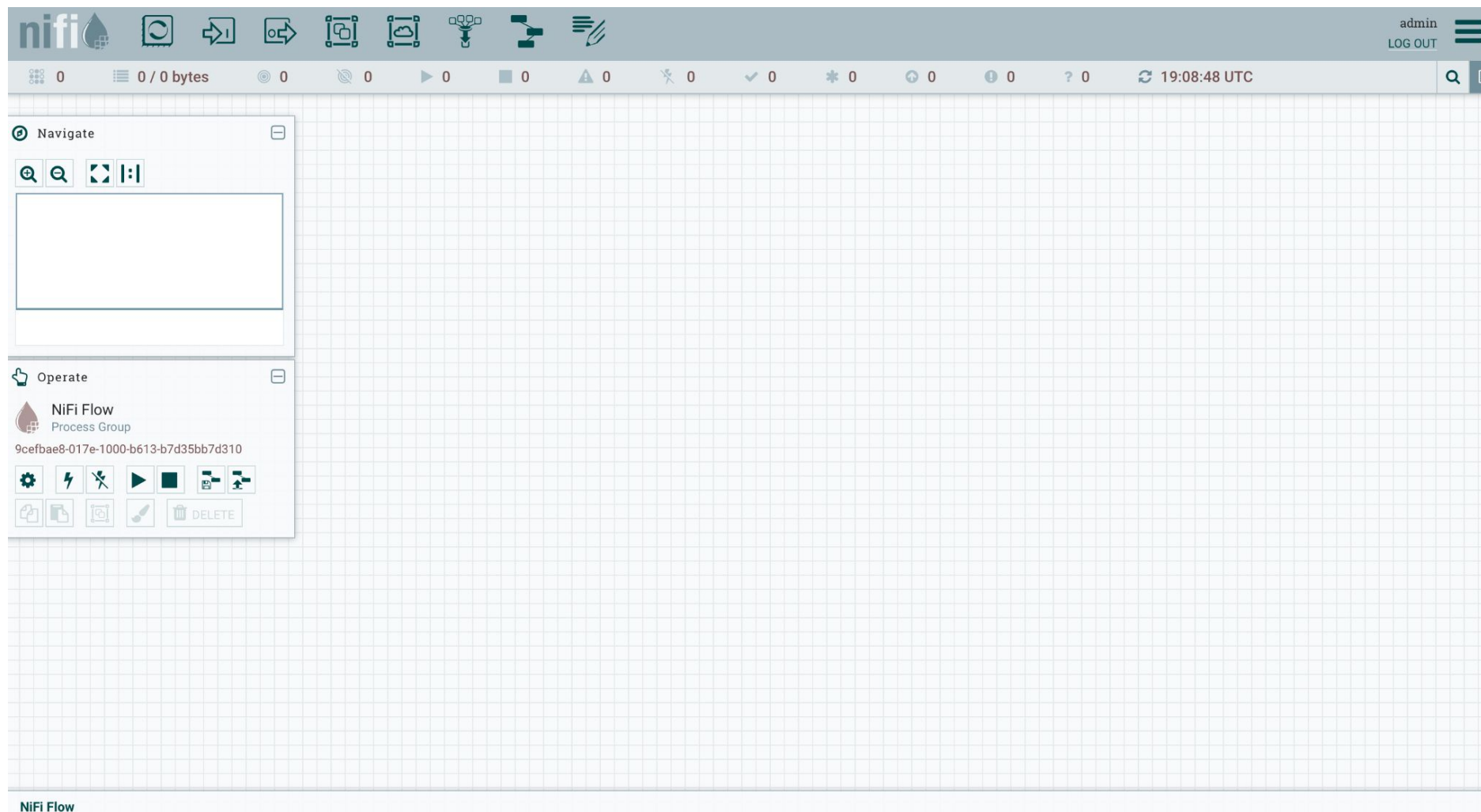
```
$ nifi.sh set-single-user-credentials <login> <password>
```

Ожидаем окончания старта:

```
$ tail -f nifi/logs/nifi-app.log
```

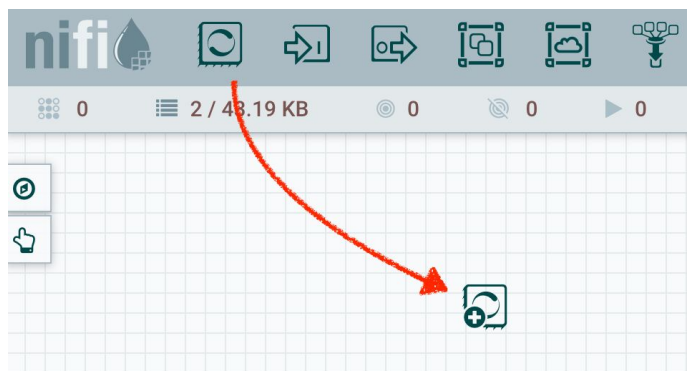
# РАБОТА В NIFI

Переходим на <https://localhost:8888>:



# РАБОТА В NIFI

Добавим GetHDFS процессор:



## Add Processor

Source

all groups

amazon attributes  
avro aws consume  
csv database  
delete fetch get  
hadoop ingest  
insert json listen  
logs message  
pubsub put query  
record restricted  
source text  
update

Displaying 4 of 297

Type

Version

GetHDFS

Tags

GetHDFS	1.15.3	restricted, get, fetch, HDFS, HC...
GetHDFSEvents	1.15.3	inotify, hadoop, events, notificat...
GetHDFSFileInfo	1.15.3	get, HDFS, HCFS, hadoop, sourc...
GetHDFSSequenceFile	1.15.3	restricted, get, fetch, sequence fi..

**GetHDFS 1.15.3** org.apache.nifi - nifi-hadoop-nar


Fetch files from Hadoop Distributed File System (HDFS) into FlowFiles. This Processor will delete the file from HDFS after fetching it.


CANCEL

ADD

# РАБОТА В NIFI

Настроим GetHDFS процессор:




**GetHDFS**  
GetHDFS 1.15.3  
org.apache.nifi - nifi-hadoop-nar

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min



Configure Processor | GetHDFS 1.15.3

Invalid

SETTINGS

SCHEDULING

PROPERTIES

COMMENTS

Required field

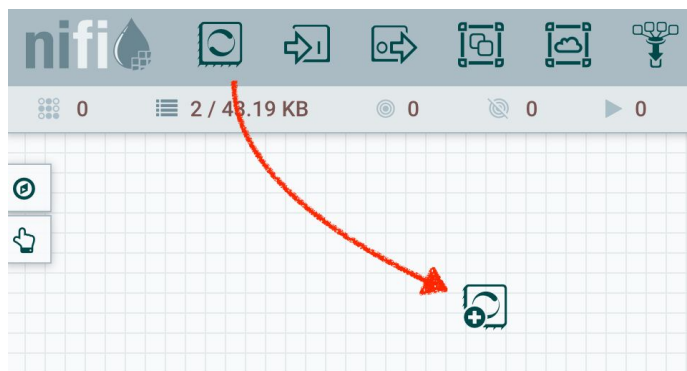
Property	Value
Hadoop Configuration Resources	<u>/home/hduser/hadoop/etc/hadoop/hdfs-site.xml</u> ,...
Kerberos Credentials Service	No value set
Kerberos User Service	No value set
Kerberos Principal	No value set
Kerberos Keytab	No value set
Kerberos Password	No value set
Kerberos Relogin Period	4 hours
Additional Classpath Resources	No value set
Directory	<u>/user/hive/warehouse/</u>
Recurse Subdirectories	true
Keep Source File	<u>true</u>
File Filter Regex	No value set

CANCEL

APPLY

# РАБОТА В NIFI

Добавим PutHDFS процессор:



## Add Processor

Source

all groups

Displaying 1 of 297

PutHDFS

amazon attributes  
avro aws consume  
csv database  
delete fetch get  
hadoop ingest  
insert json listen  
logs message  
pubsub put query  
record restricted  
source text  
update

Type

Version

Tags

PutHDFS

1.15.3

restricted, HDFS, HCFS, hadoop...

PutHDFS 1.15.3 org.apache.nifi - nifi-hadoop-nar

Write FlowFile data to Hadoop Distributed File System (HDFS)

CANCEL

ADD



# РАБОТА В NIFI

Настроим PutHDFS процессор:

Configure Processor | PutHDFS 1.15.3

Invalid

SETTINGS

SCHEDULING

PROPERTIES

COMMENTS

Required field

Property	Value
Hadoop Configuration Resources	<u>/home/hduser/hadoop/etc/hadoop/hdfs-site.xml,...</u>
Kerberos Credentials Service	No value set
Kerberos User Service	No value set
Kerberos Principal	No value set
Kerberos Keytab	No value set
Kerberos Password	No value set
Kerberos Relogin Period	4 hours
Additional Classpath Resources	No value set
Directory	<u>/user/hduser/nifi/</u>
Conflict Resolution Strategy	<u>append</u>
Writing Strategy	<u>Write and rename</u>
Block Size	No value set
IO Buffer Size	No value set

CANCEL

APPLY





PutHDFS

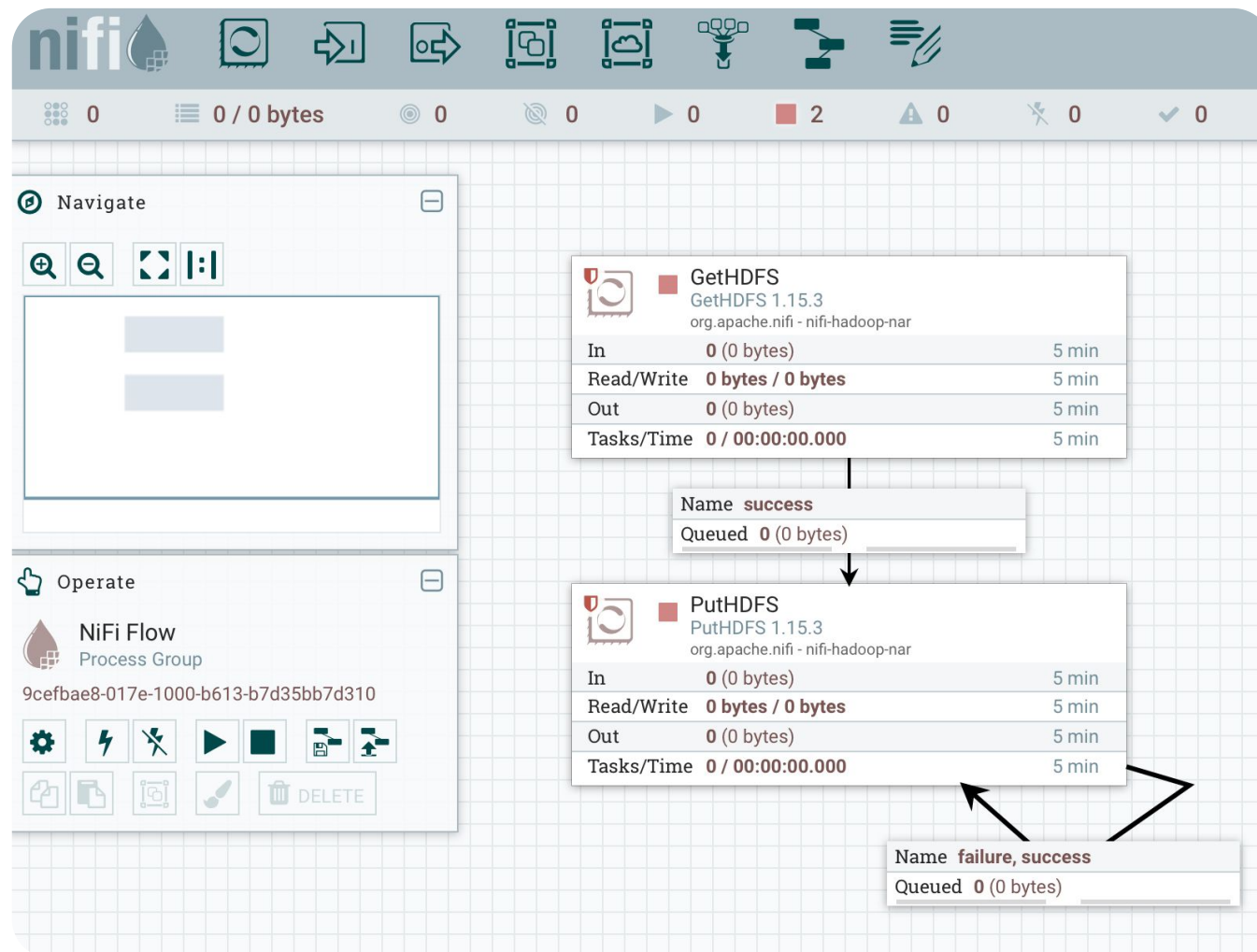
PutHDFS 1.15.3

org.apache.nifi - nifi-hadoop-nar

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

# РАБОТА В NIFI

Соединим GetHDFS и PutHDFS процессоры:



# РАБОТА В NIFI

Запустим GetHDFS и PutHDFS процессоры:

The screenshot shows the Apache NiFi console interface. The top processor is 'GetHDFS 1.15.3' with a status of 'org.apache.nifi - nifi-hadoop-n'. Its metrics are: In: 0 (0 bytes), Read/Write: 0 bytes / 0 bytes, Out: 0 (0 bytes), Tasks/Time: 0 / 00:00:00.000. A context menu is open over the processor, showing options: Configure, Start, Run Once (highlighted), Disable, View data provenance, View status history, View usage, and View connections. Below the processor, a data provenance record is visible with 'Name success' and 'Queued 0 (0 bytes)'. At the bottom, the 'PutHDFS 1.15.3' processor is partially visible.



The screenshot shows the Apache NiFi console interface. The top processor is 'PutHDFS 1.15.3' with a status of 'org.apache.nifi - nifi-hadoop-nar'. Its metrics are: In: 0 (0 bytes), Read/Write: 0 bytes / 0 bytes, Out: 0 (0 bytes), Tasks/Time: 0 / 00:00:00.000. A context menu is open over the processor, showing options: Configure, Start, Run Once, Disable, View data provenance, View status history, View usage, and View connections. Below the processor, a data provenance record is visible with 'Name failure' and 'Queued 0 (0 bytes)'. A black arrow points from the 'Run Once' option in the menu to the processor itself.

# РАБОТА В NIFI

После окончания работы в HDFS появилась директория с данными:

```
hduser@localhost:~$ hdfs dfs -ls /user/hduser
```

```
Found 1 items
```

```
drwxr-xr-x    - hduser supergroup          0 2022-01-27 19:44 /user/hduser/nifi
```

# ОСТАНОВКА NIFI

Остановим Apache NiFi:

```
$ nifi.sh stop
```

```
...
```

```
NiFi has finished shutting down.
```

# ПРАКТИЧЕСКОЕ ЗАДАНИЕ



## ПРАКТИЧЕСКОЕ ЗАДАНИЕ

1. Установите Apache Sqoop
2. Выведите таблицы Hive через JDBC драйвер в Sqoop
3. Установите Apache Flume
4. Напишите приложение (на bash или python), которое будет периодически писать свои логи или любые другие данные в stdout, после чего настройте сборку этих логов в HDFS через Flume
5. Установите Apache NiFi
6. Соберите пайплайн в NiFi, который будет читать локальный файл и записывать его в HDFS или наоборот

**Спасибо!**

**Каждый день  
вы становитесь  
лучше :)**

