

# Apache Spark и PySpark

Инструкция для того, что начать использовать ruspark для учебных целей.

## Docker

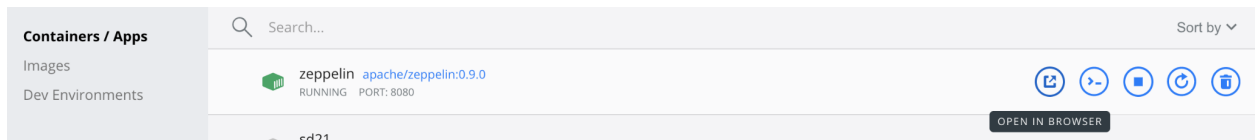
Есть два популярных инструмента для анализа данных. Устанавливайте один из них, который вам удобнее.

Предварительно необходимо установить [docker desktop](#), чтобы использовать готовые образы. По [ссылке](#) нужно скачать и установить Docker Desktop (при установке могут потребоваться установка дополнительных компонент, следуйте инструкции при установке).

## Zeppelin

Установка:

- В командной строке (terminal, либо powershell) выполняем следующую операцию:  
`docker run -p 8080:8080 --rm -v $PWD/notebook:/notebook -e ZEPPELIN_NOTEBOOK_DIR='/notebook' --name zeppelin apache/zeppelin:0.9.0`  
, где `/notebook` - директория, которой будут лежать ваши ноутбуки;
- дождитесь скачивания и запуска контейнера;
- в docker появится новый запущенный контейнер;



- нажимаем “Открыть в браузере”, либо сразу переходим по ссылке `localhost:8080`;
- откроется zeppelin.

[Подробнее про установку.](#)

Использование:

- По ссылке `localhost:8080` после запуска откроется zeppelin, где можно создавать блокноты;

## Welcome to Zeppelin!

Zeppelin is web-based notebook that enables interactive data analytics.  
You can make beautiful data-driven, interactive, collaborative document with SQL, code and even more!

### Notebook ↻

 [Import note](#)

 [Create new note](#)

 [Trash](#)


### Help

Get started with [Zeppelin documentation](#)

### Community

Please feel free to help us to improve Zeppelin,  
Any contribution are welcome!

 [Mailing list](#)

 [Issues tracking](#)

 [Github](#)

- zeppelin из коробки поддерживает spark, создаём блокнот (интерпретатор %spark);

Create New Note

Note Name

Default Interpreter

spark

Use '/' to create folders. Example: /NoteDirA/Note1

Create

- далее в ячейке прописываем интерпретатор %pyspark и можно писать код (блокнот уже содержит инициализированные SparkContext и SQLContext);

Zeppelin Notebook Job

spark example

```
%pyspark
spark
```

SparkSession - in-memory

SparkContext

Spark UI

Version  
v2.4.5

Master  
local[\*]

AppName  
ab9938cc-f812-4f48-a8c3-ab29fe76a2d3

Took 15 sec.

```
%pyspark
sc
```

SparkContext

Spark UI

Version  
v2.4.5

Master  
local[\*]

AppName  
ab9938cc-f812-4f48-a8c3-ab29fe76a2d3

Took 12 sec.

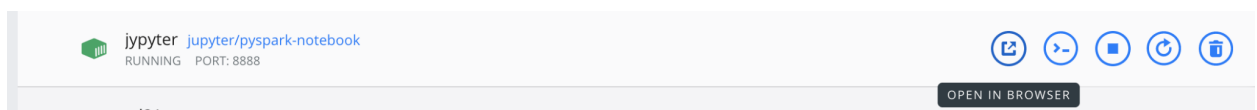
- подробнее [тут](#).

## Jupyter

Для jupyter так же есть готовый образ со spark.

Установка:

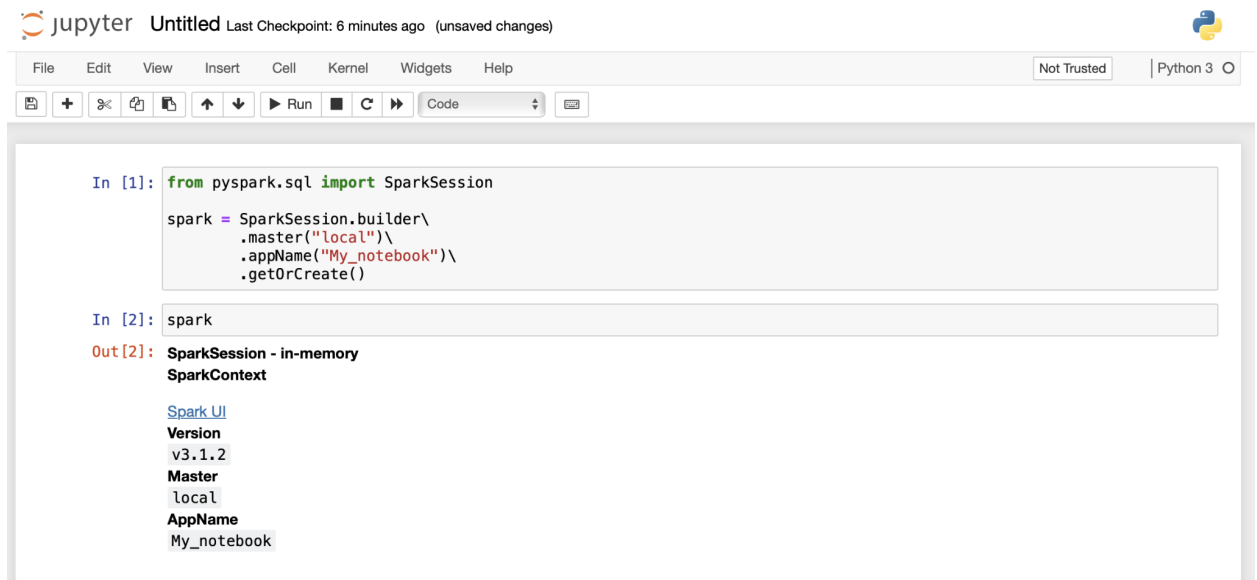
- В командной строке (terminal, либо powershell) выполняем следующую операцию:  
`docker run -p 8888:8888 -v $PWD/notebook:/home/jovyan/work --name jupyter jupyter/pyspark-notebook start.sh jupyter notebook --NotebookApp.token=' '`  
, где `/notebook` - директория, которой будут лежать ваши ноутбуки;
- дождитесь скачивания и запуска контейнера;
- в docker появится новый запущенный контейнер;



- нажимаем “Открыть в браузере”, либо сразу переходим по ссылке `localhost:8888`;
- откроется Jupyter.

Использование:

- Создаём новый блокнот;
- создаём SparkSession.



The screenshot shows a Jupyter Notebook titled 'Untitled' with a 'Last Checkpoint: 6 minutes ago (unsaved changes)' status. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running, and code execution. The notebook contains two input cells and one output cell. The first input cell (In [1]:) contains the following Python code:

```
from pyspark.sql import SparkSession

spark = SparkSession.builder\
    .master("local")\
    .appName("My_notebook")\
    .getOrCreate()
```

The second input cell (In [2]:) contains the code:

```
spark
```

The output cell (Out [2]:) displays the result of the code execution:

```
SparkSession - in-memory
SparkContext

Spark UI
Version
v3.1.2
Master
local
AppName
My_notebook
```

- готово, можно использовать ruyspark. Подробнее про jupyter [тут](#).

## Colab - облачный jupyter от google

Пример блокнота:

[https://colab.research.google.com/drive/1cwztCPGUHtAwobs\\_dfZeLM6faqLgGlcA?usp=sharing](https://colab.research.google.com/drive/1cwztCPGUHtAwobs_dfZeLM6faqLgGlcA?usp=sharing)

## Локально без docker

- 1) Устанавливаем питон (например <https://docs.anaconda.com/anaconda/install/index.html> , куда уже включён jupyter)
- 2) Устанавливаем ruyspark (pip install)
- 3) Возможно понадобится установить Java 8 (подробная инструкция [тут](#))