

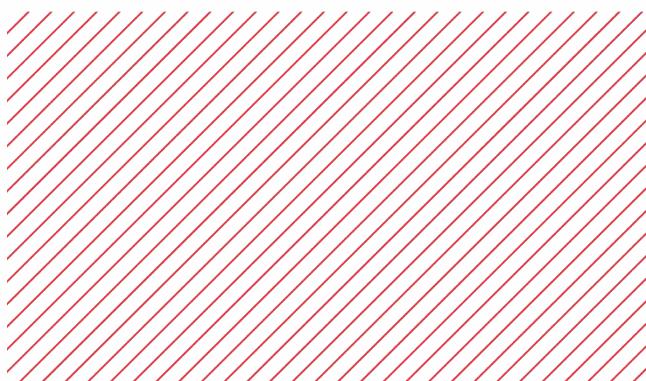
академия  
больших  
данных



# Инструменты визуализации при работе с Большими Данными

Андрей Кузнецов

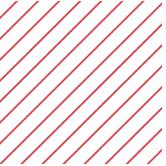
09.10.2021



# Структура курса

---

1. Введение в Большие Данные
2. Hadoop экосистема и MapReduce
3. SQL поверх больших данных
4. Инструменты визуализации при работе с Большими Данными 
5. Введение в Scala
6. Модель вычислений Spark: RDD
7. Распараллеливание алгоритмов ML
8. Spark Pipelines
9. Approximate алгоритмы для больших данных
10. Spark для оптимизации гиперпараметров
11. Потоковая обработка данных (Kafka, Spark Streaming, Flink)
12. Архитектуры в продакшене



# План занятия

---

1. Apache Zeppelin
2. Polynote
3. Big Data Tools
4. Cloud Solutions
5. Workshop:
  - a. cloud-based work
  - b. zeppelin
  - c. polynote
  - d. homework discussion



# Where we are?

---

## Big Data platform

- Hadoop Distributed Filesystem (NM, DN)
- Apache Hadoop YARN (RM, NM)

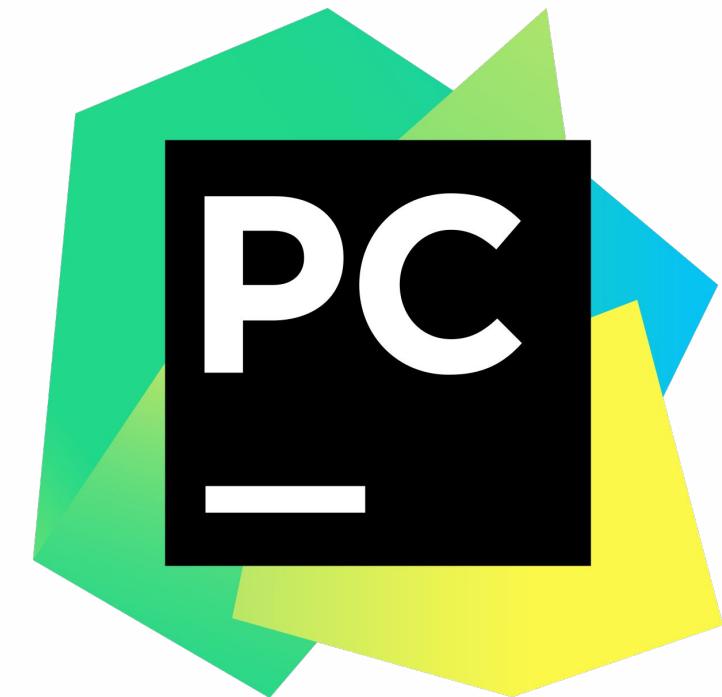
## Big Data applications

- Hadoop MapReduce
- SQL-like processing frameworks
- Apache Spark
- Stream processing frameworks + Apache Kafka

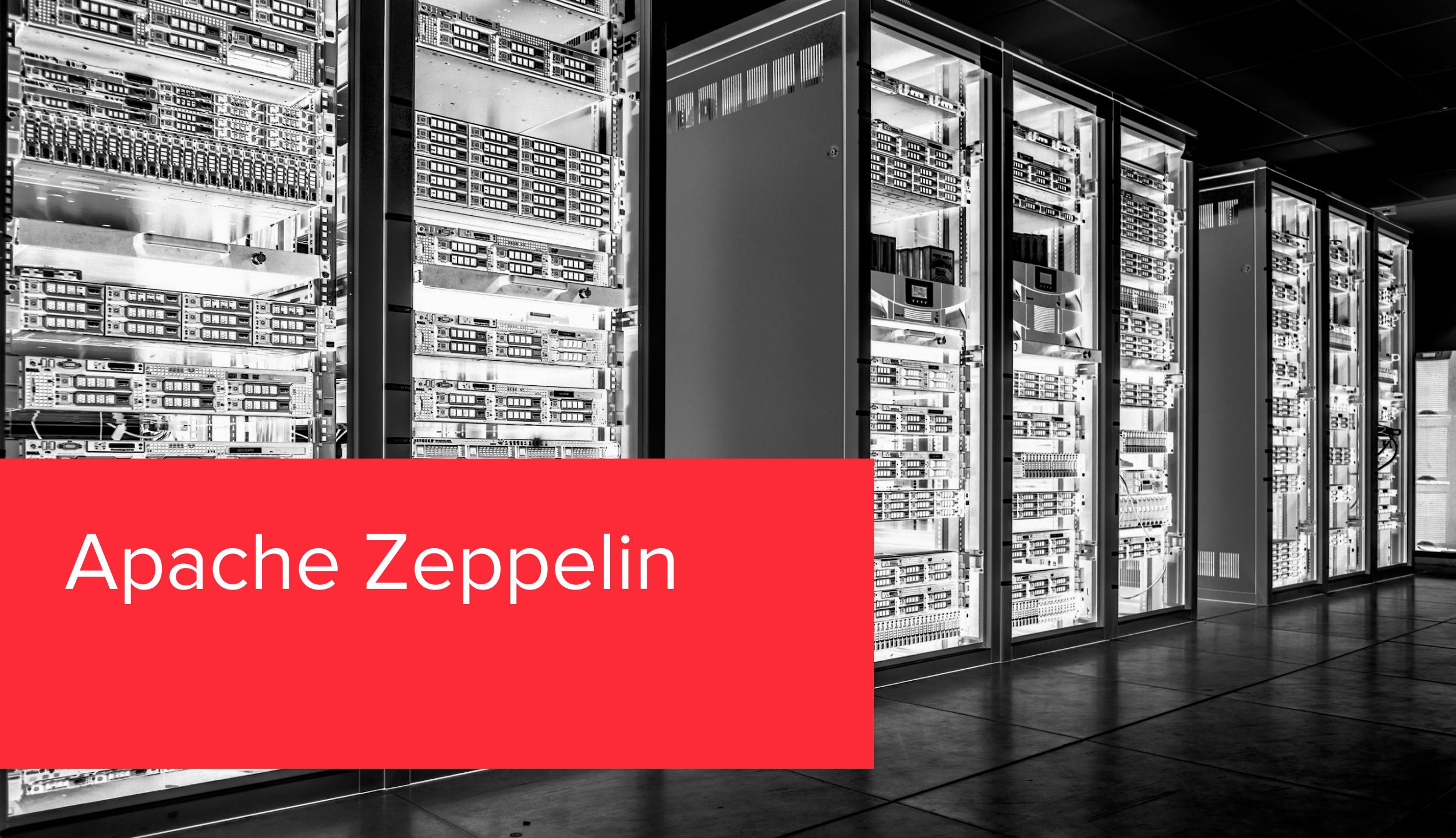
→ Tools to work with

# Classic small data stack

---



# Apache Zeppelin

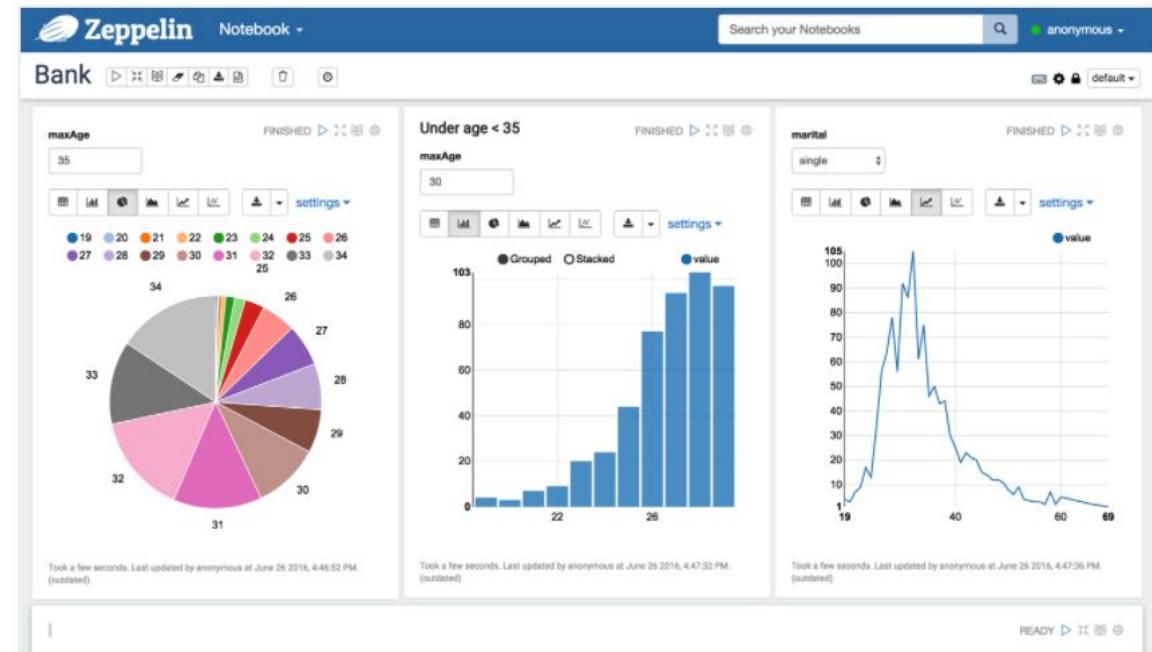


# Apache Zeppelin in a nutshell

## Multi-purpose Notebook

The Notebook is the place for all your needs

- >Data Ingestion
- Data Discovery
- Data Analytics
- Data Visualization & Collaboration



# Apache Zeppelin in a nutshell

---

- Хочет быть как jupyter notebook / jupyterlab
- умеет нативно работать со Scala/Java
- Hive/Spark удобно конфигурировать и работать
- Неплохие визуализации



# Zep context. One interface to rule them all

The screenshot shows the Zep interface with two code snippets side-by-side.

**Left Snippet:**

```
%spark  
// This is just for demo, it could be done via running  
// a spark job  
z.put("maxAge", 83)
```

**Right Snippet:**

```
%jdbc(interpolate=true)  
select * from bank where age = {maxAge}
```

The right snippet has a results table below it:

| age | job     | marital  |
|-----|---------|----------|
| 83  | retired | married  |
| 83  | retired | married  |
| 83  | retired | divorced |
| 83  | retired | divorced |

# Scheduling. Don't try on prod!

The screenshot shows the Zeppelin web interface. At the top, there is a navigation bar with the Zeppelin logo, 'Notebook' dropdown, 'Job' dropdown, a search bar, and a user status 'anonymous'. Below the navigation bar, the word 'note' is displayed, followed by a set of icons for navigation, search, and file operations. A scheduled job is shown with a timer icon and the text '1h'. On the left, a notebook titled '# Hello, Cron Scheduler' contains the text 'Hello, Cron Scheduler'. A tooltip is open over the cron scheduler settings, which include:

- Run note with cron scheduler. Either choose from preset or write your own [cron expression](#).
- Preset [None](#) [1m](#) [5m](#) [1h](#) [3h](#) [6h](#) [12h](#) [1d](#)
- Cron expression
- After execution stop the interpreter

On the right side of the interface, there is a 'FINISHED' status indicator with a set of icons.



# Notebook storages

---

1. (default) use local file system and version it using local Git repository - GitNotebookRepo
2. all notes are saved in the notebook folder in your local File System - VFSNotebookRepo
3. all notes are saved in the notebook folder in hadoop compatible file system - FileSystemNotebookRepo
4. storage using Amazon S3 service - S3NotebookRepo
5. storage using Azure service - AzureNotebookRepo
6. storage using Google Cloud Storage - GCSNotebookRepo
7. storage using Aliyun OSS - OSSNotebookRepo
8. storage using MongoDB - MongoNotebookRepo
9. storage using GitHub - GitHubNotebookRepo

# Interpreters. JDBC

---



- Postgresql - JDBC Driver
- Mysql - JDBC Driver
- MariaDB - JDBC Driver
- Redshift - JDBC Driver
- Apache Hive - JDBC Driver
- Presto/Trino - JDBC Driver
- Impala - JDBC Driver
- Apache Phoenix itself is a JDBC driver
- Apache Drill - JDBC Driver
- Apache Tajo - JDBC Driver

# Polynote

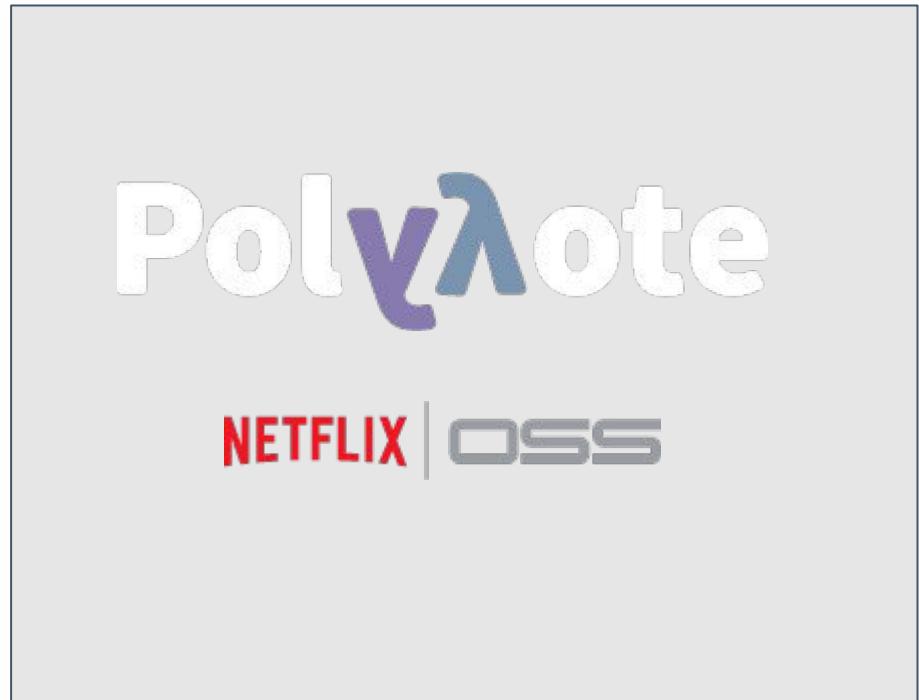


# Polynote in a nutshell

---

- Умеет варить объекты в одном кернели и конвертить из Scala в Python
- Сырой, но хоть какая-то альтернатива Zeppelin / Jupyter

<https://polynote.org/>

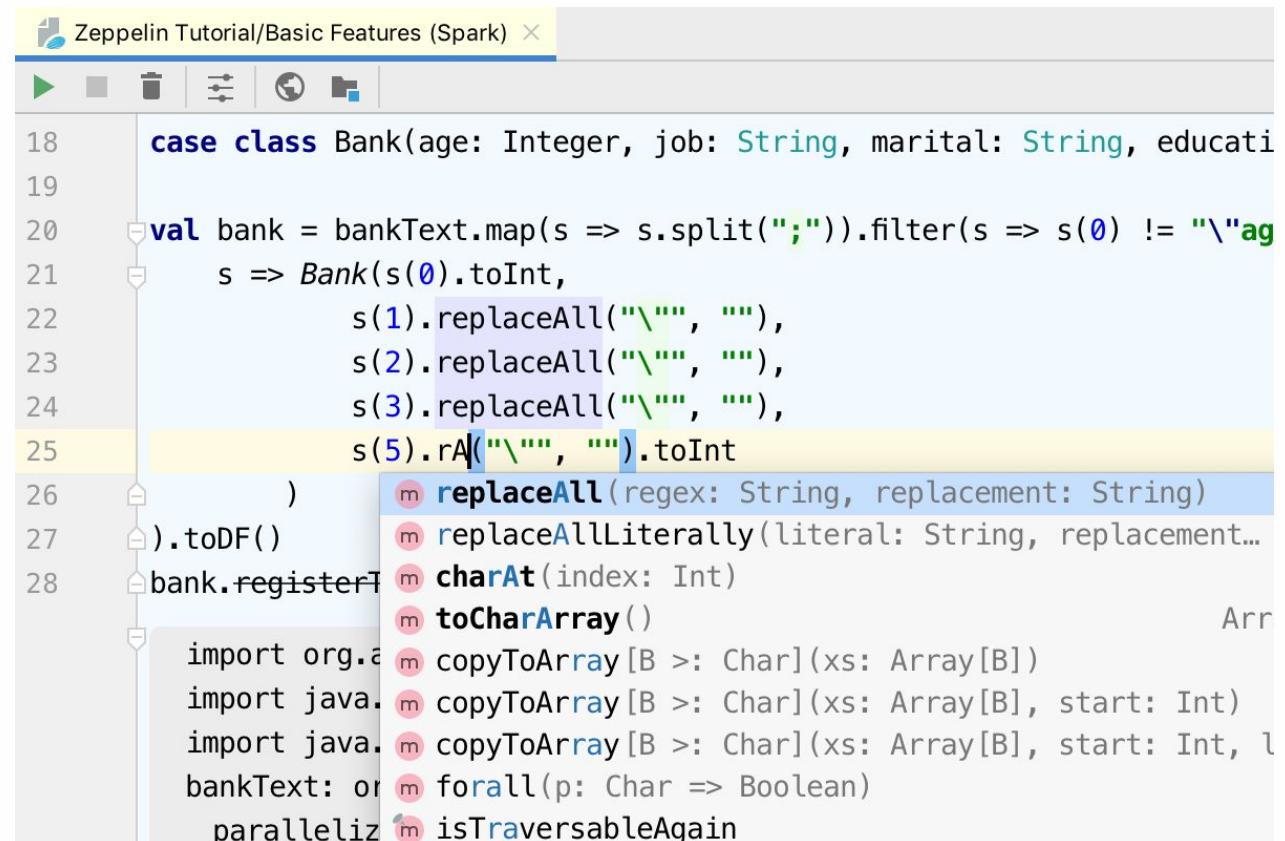


# Big Data Tools



# Big Data Tools in a nutshell

- Интеллектуальная поддержка Zeppelin notebooks
- Интеграция с инструментами Spark и Hadoop
- Распределенные файловые системы и столбцовые форматы
- Инструменты для работы с таблицами и диаграммами



The screenshot shows a Zeppelin notebook titled "Zeppelin Tutorial/Basic Features (Spark)". The code editor contains the following Scala code:

```
case class Bank(age: Integer, job: String, marital: String, education: String, ...)

val bank = bankText.map(s => s.split(";")).filter(s => s(0) != "\u0410")
  s => Bank(s(0).toInt,
            s(1).replaceAll("\u041f\u0435\u0440\u0435\u0434\u043d\u044c\u043e\u0433\u043e \u0431\u043b\u043e\u0436\u0435\u043d\u0438\u044f", ""),
            s(2).replaceAll("\u041f\u0435\u0440\u0435\u0434\u043d\u044c\u043e\u0433\u043e \u0431\u043b\u043e\u0436\u0435\u043d\u0438\u044f", ""),
            s(3).replaceAll("\u041f\u0435\u0440\u0435\u0434\u043d\u044c\u043e\u0433\u043e \u0431\u043b\u043e\u0436\u0435\u043d\u0438\u044f", ""),
            s(5).rA(\u041f\u0435\u0440\u0435\u0434\u043d\u044c\u043e\u0433\u043e \u0431\u043b\u043e\u0436\u0435\u043d\u0438\u044f, "").toInt
)
.toDF()
bank.registerTempTable("bank")
```

A tooltip for the `s.replaceAll` method is displayed, listing its parameters and related methods:

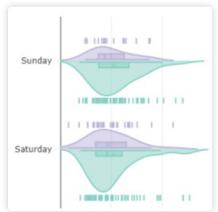
- `m replaceAll(regex: String, replacement: String)`
- `m replaceAllLiterally(literal: String, replacement: String)`
- `m charAt(index: Int)`
- `m toCharArray()`
- `m copyToArray[B >: Char](xs: Array[B])`
- `m copyToArray[B >: Char](xs: Array[B], start: Int)`
- `m copyToArray[B >: Char](xs: Array[B], start: Int, length: Int)`
- `m forall(p: Char => Boolean)`
- `m isTraversableAgain`

# More vis tools

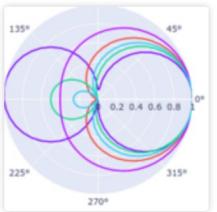


# Plotly. Multilang vis tool

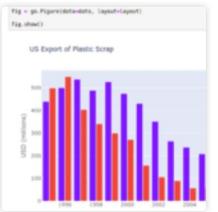
## Fundamentals



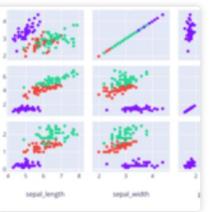
The Figure Data Structure



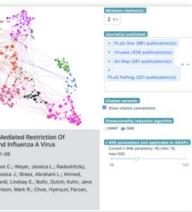
Creating and Updating Figures



Displaying Figures



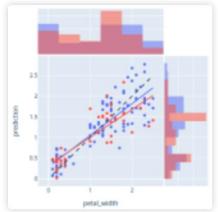
Plotly Express



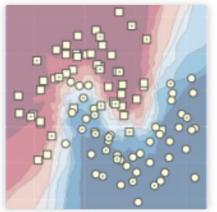
Analytical Apps with Dash

More Fundamentals »

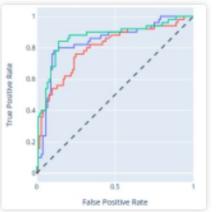
## Artificial Intelligence and Machine Learning



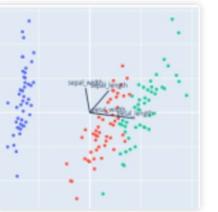
ML Regression



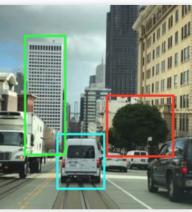
kNN Classification



ROC and PR Curves



PCA Visualization

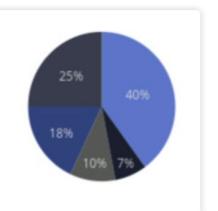
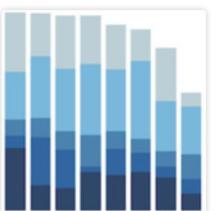
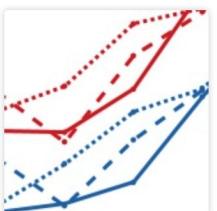
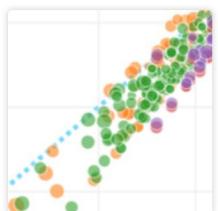


AI/ML Apps with Dash

More AI and ML »

<https://plotly.com/python/>

## Basic Charts

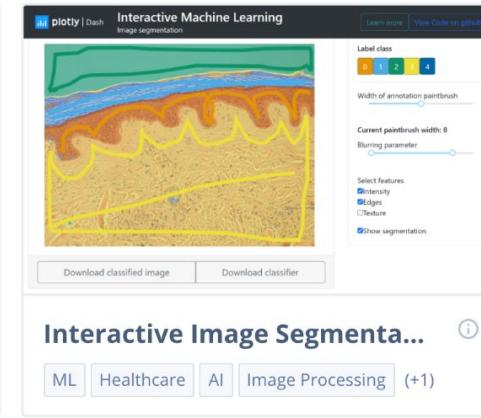
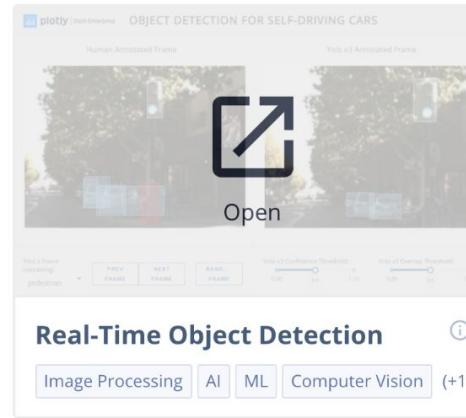


More Basic Charts »

# Dash. Visual apps at web

All Apps (110)

Search applications...



<https://dash.gallery/Portal/>

# Tableau. Expensive industrial standart

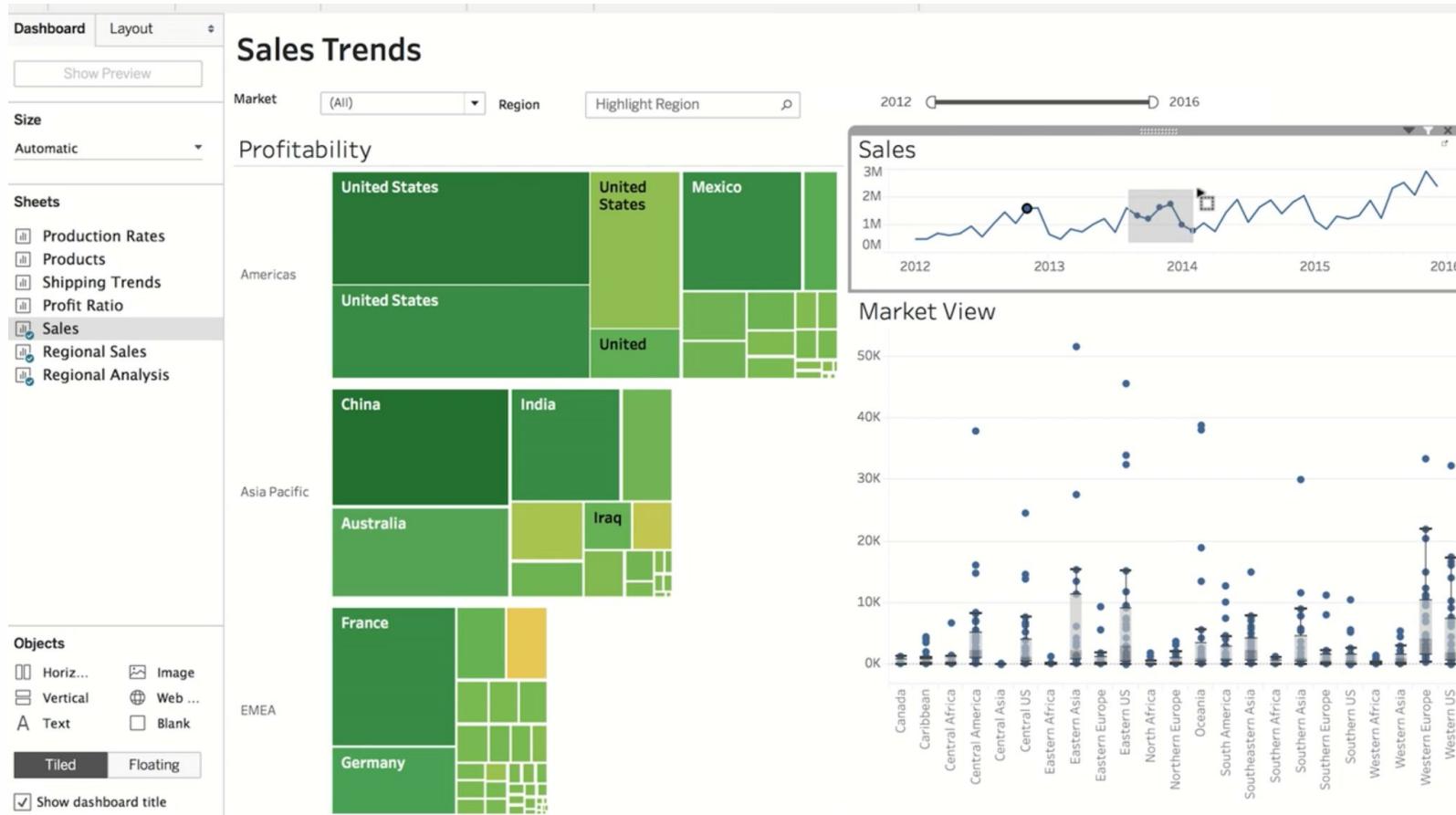
## Tableau native data connectors



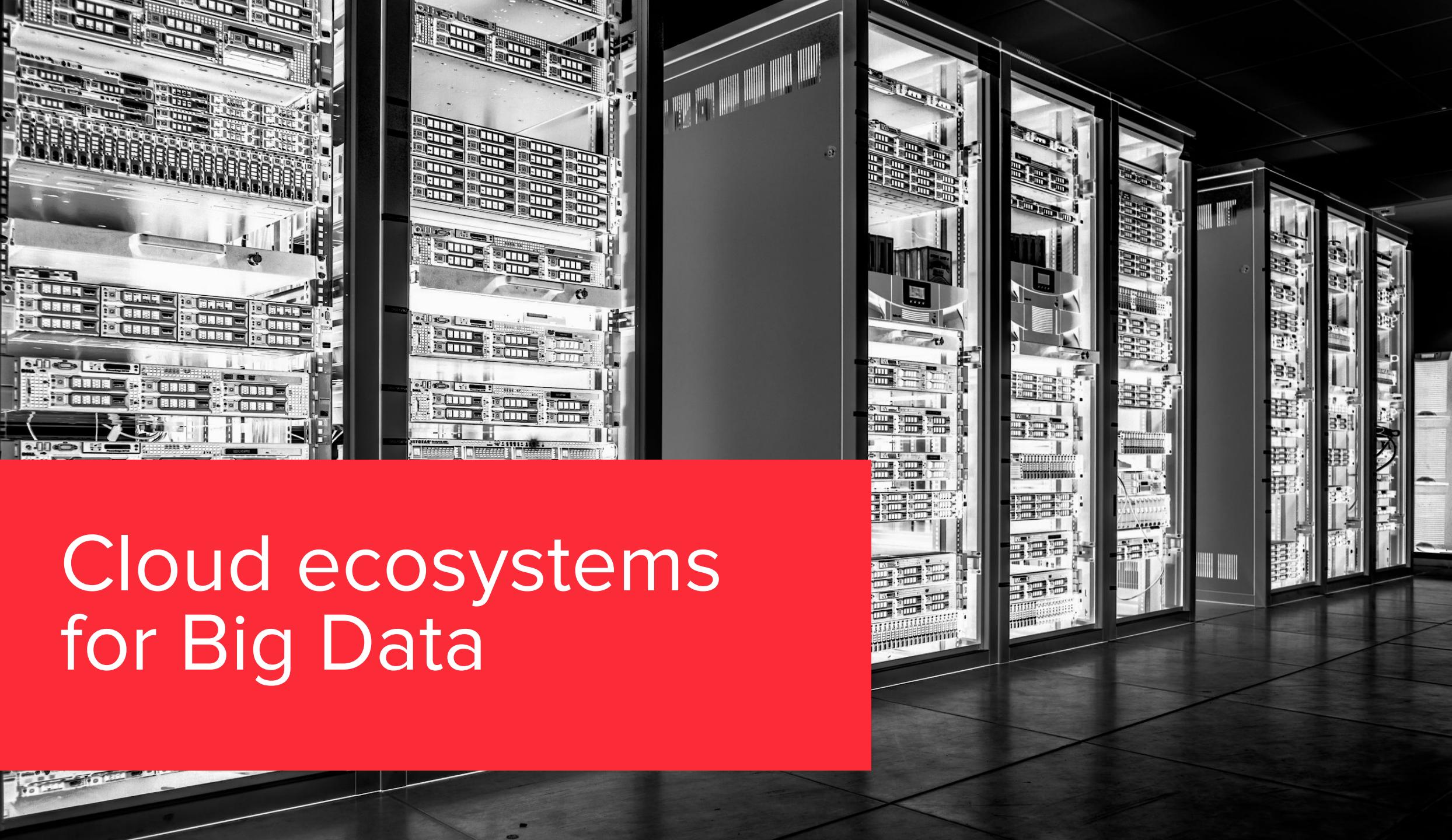
Connect to all of your data—no matter where it resides.

Tableau offers **native connectors** built and optimized for many databases and files—from spreadsheets and PDFs to big data, cube, and relational databases on-premises or in the cloud, even application data or data on the web.

# Tableau. Interface

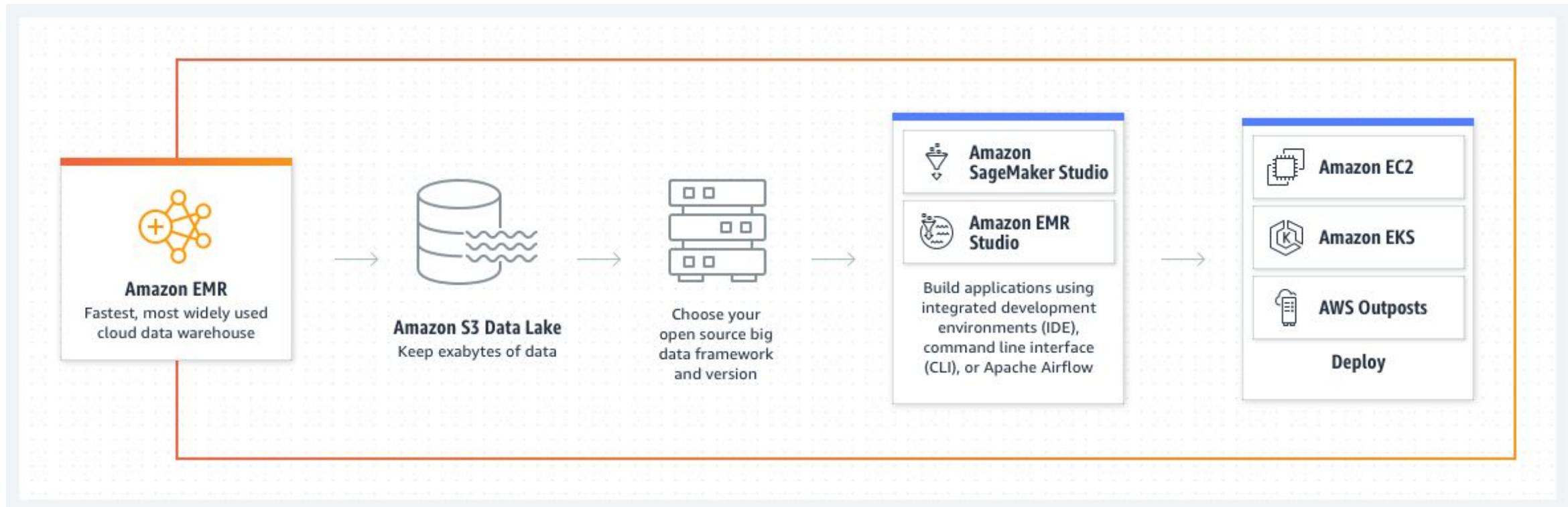


# Cloud ecosystems for Big Data



# Amazon EMR

<https://aws.amazon.com/ru/emr/features/?nc=sn&loc=2&dn=1>



Amazon EMR – это платформа для быстрой обработки, анализа и работы с большими данными с помощью машинного обучения (ML), использующая платформы с открытым исходным кодом.

# Mail.ru Cloud Solutions

Mail.ru Cloud Solutions

Облачные вычисления

Виртуальные сети

Объектное хранилище

Контейнеры

Базы данных

Аналитические БД

Магазин приложений

Большие данные

Кластеры

Графические адAPTERы

Машинное обучение

Специальные сервисы

Управление доступами

Баланс

cloud big data

## Большие данные

Облачные Big Data сервисы MCS обеспечивают быстрое решение задач обработки больших данных, событий и realtime-аналитики на базе Hadoop и Spark



### Кластеры

Быстрое создание кластеров на базе Hadoop Hortonworks

S3-совместимое хранилище и кластеры Spark до 16 TB RAM

[Создать кластер](#)

[Создать кластер](#)

- ✓ Динамическое масштабирование до сотен узлов при пиковых нагрузках
- ✓ Посекундная тарификация, отсутствие затрат на закупку оборудования

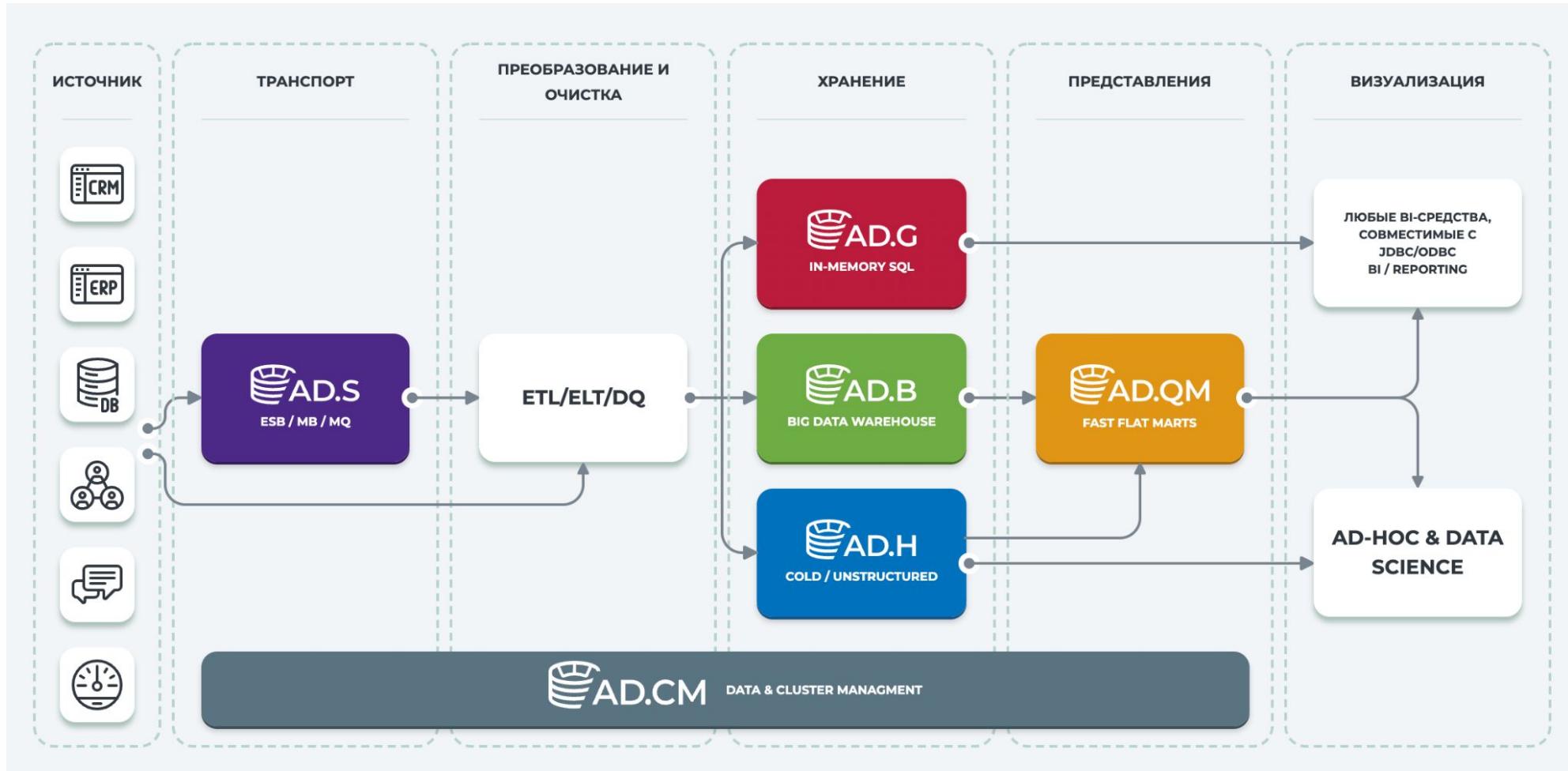
Если у вас возникли вопросы, вы можете ознакомиться с [документацией](#)

f

Telegram

# ArenaData

<https://arenadata.tech/>



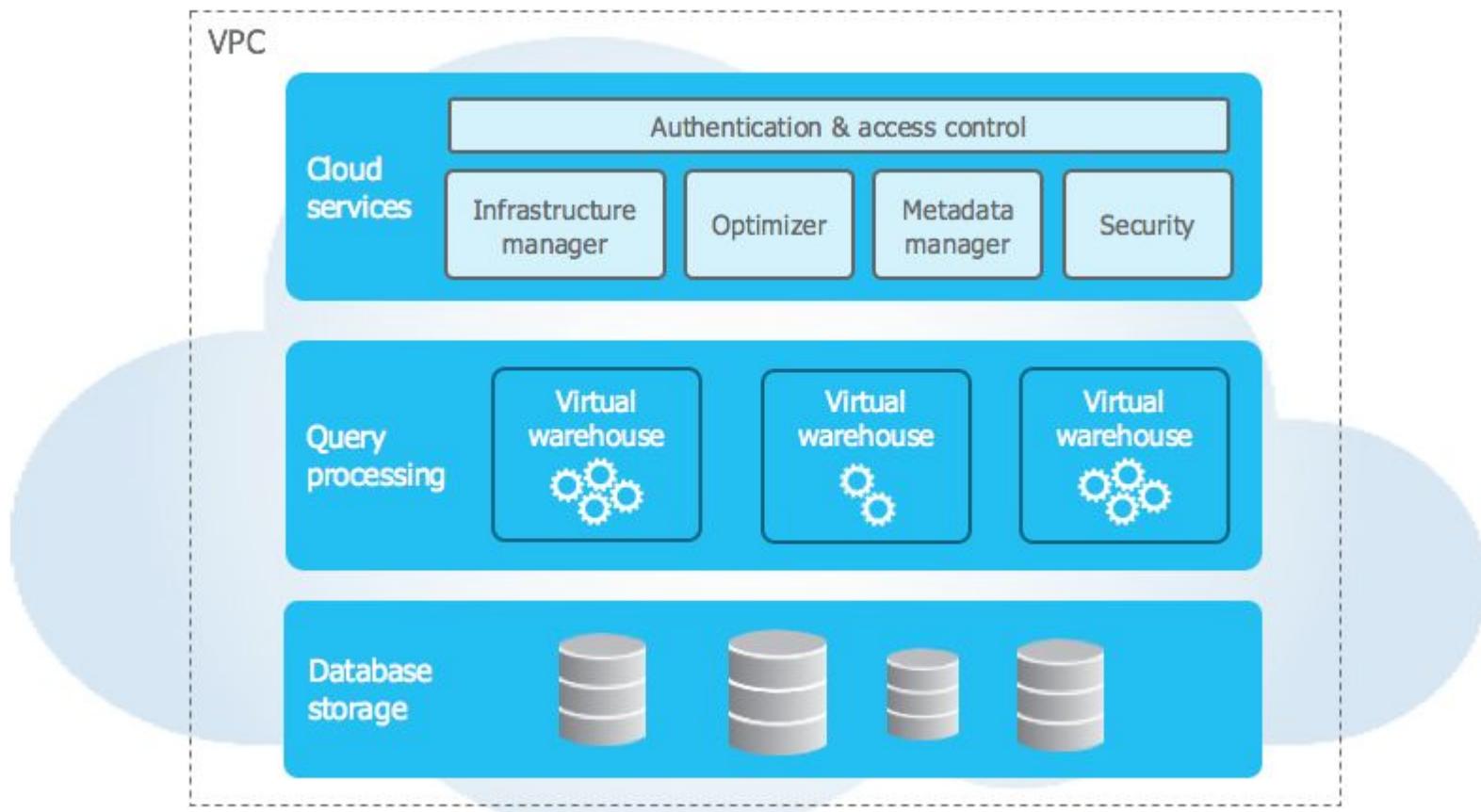
# Snowflake

---

**WHERE YOUR DATA CLOUD EXPERIENCE BEGINS:  
ONE PLATFORM, MANY WORKLOADS, NO DATA SILOS**



# Snowflake





# Recommended links and literature

---

- 1) <https://polynote.org/>
- 2) <https://zeppelin.apache.org/>
- 3) <https://docs.aws.amazon.com/emr/index.html>