

Big Data Tools: Holy Grail of Developer Productivity

Pasha Finkelshteyn, JetBrains

Who am I

- ex system administrator
- ex developer
- ex team lead
- ex data engineer
- developer advocate for big data

Together >14 years in IT



Who are data engineers?

Responsibilities:

- Build your DWH
- Build your DMP
- Transfer and store your data

As effective and fast as possible

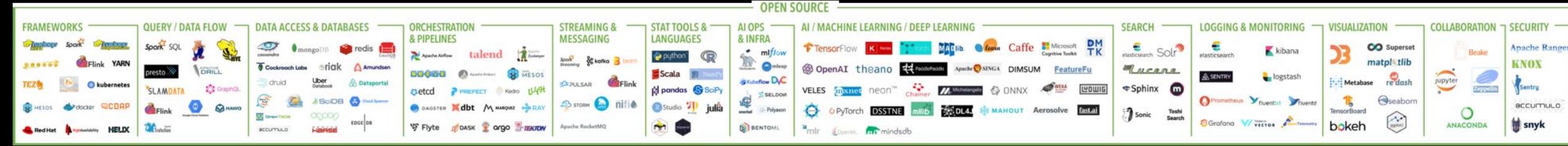
I know pain points of data engineering

I know pain points of data engineering

And today I'll try to solve them for you



Lots of tools



Lots of tools

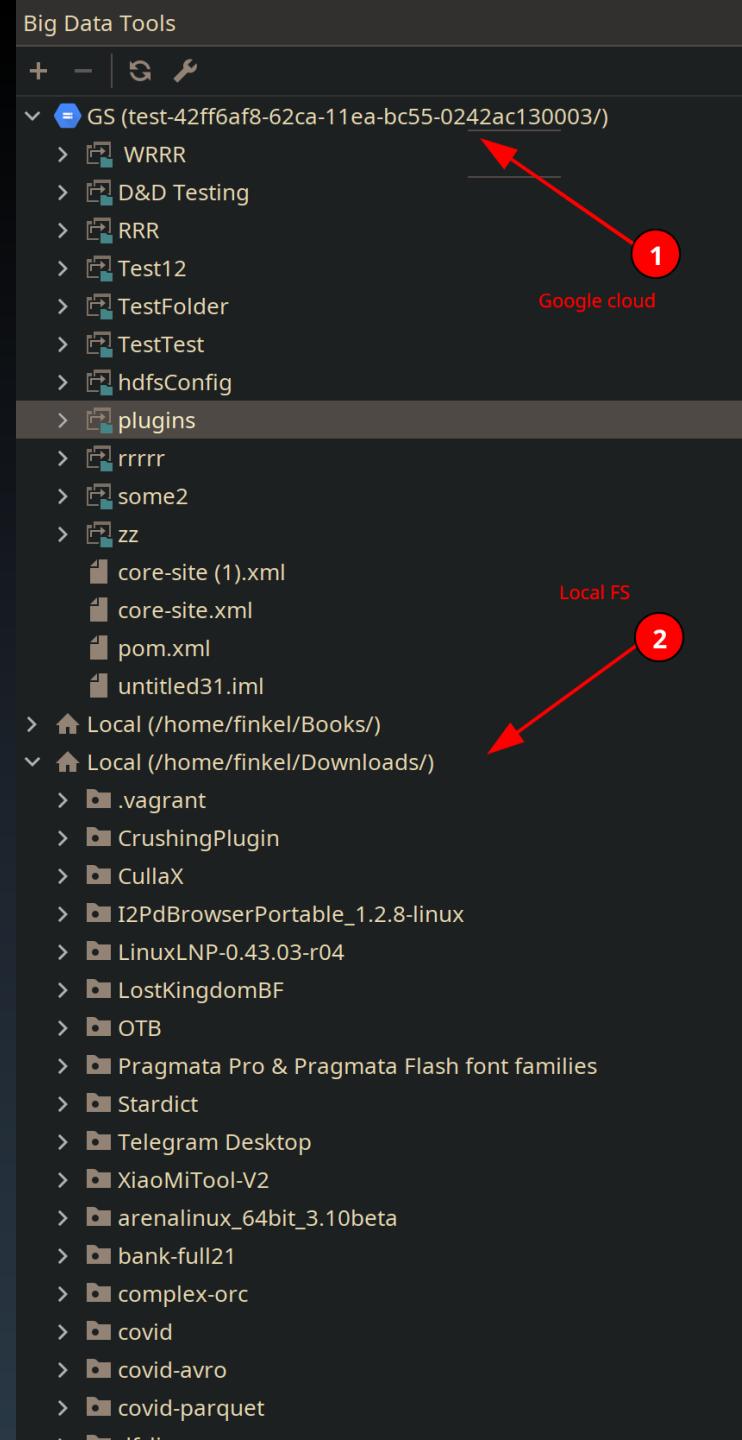
- Nobody may know everything
- Constant context switches
- No single point of work

Ultimate Big Data Workspace

Single IDE to work with Kafka, Spark, and storages of different types

Remote FS support

- Drag'n'Drop support
- Local ↔ Remote
- Remote ↔ Remote
- Basic operations:
 - Rename
 - Move
 - Copy
 - Delete



Kafka support

- Consumers (and groups)
- Topic metadata
 - Replicas
 - Partitions (and info on them)
 - Under replicated partitions

Apache Zeppelin support

“ Web-based notebook that enables data-driven, interactive data analytics and collaborative documents with SQL, Scala and more. ”

Zeppelin is awesome environment to run ad-hoc tasks.

But this is not the best IDE humanity developed.

Create Dataset/DataFrame via SparkSession

%spark

How autocomplete looks in web version of Zeppelin

```
// create DataFrame from scala Seq. It can infer schema for you.  
val df1 = spark.createDataFrame(Seq((1, "andy", 20, "USA"), (2, "jeff", 23, "China"), (3, "james", 18, "USA"))  
df1.printSchema  
df1.show()  
df1.  
// c abc China  
case abc DataFrame  
val abc Dataset  
df2. abc Int  
df2. abc It  
    abc Person  
import abc Seq  
// y abc String  
val abc USA  
df3. abc _  
df3. abc age  
abc also
```

Person(2, "jeff", 23, "China"), Person(3,

Person(2, "jeff", 23, "China"), Person(3, "j

// create DataFrame from scala Seq. It can infer schema for you.

```
val df1 = spark.createDataFrame(Seq((1, "andy", 20, "USA"), (2, "jeff", 25, "CA")))
```

df1.printSchema

```
df1.show()
```

```
df1.
```

```
// m as[U](implicit evidence$2: Encoder[U])
```

```
ca m show numRows: Int) Unit
```

```
va m filter(func: Row => Boolean) Dataset[Row]
```

```
df m toDF() sql.DataFrame
```

```
df m write DataFrameWriter[Row]
```

```
m as(alias: String) Dataset[Row]
```

```
m as(alias: Symbol) Dataset[Row]
```

```
im m collectAsList() util.List[Row]
```

```
// m col(colName: String) Column
```

```
asm0di0 m collect() Array[Row]
```

```
BigDataToolsJB
```

```
va m collect() Array[Row]
```

How Autocompletion looks in Big Data Tools

Autocompletion

Autocompletion is one of the key factors of developer productivity today.

IntelliJ IDEA uptime: 12 sec, 809 ms, idle time: 134 ms

Code completion has saved you from typing at least 117.6K characters since 11.02.2020 (~497 per working day)

Quick fixes have saved you from 5,720 possible bugs since 11.02.2020 (~25 per working day)

Feature ▲	Group	Used	Last Used
Basic code completion	Code Completion	26,274 times	one week ago
Browse external documentation	Code Assistants	Never	N/A
Camel prefixes in code completion	Code Completion	2,464 times	one week ago
Cancel lookup and move caret up/...	Code Completion	Never	N/A
Changing completion variants sorting	Code Completion	Never	N/A
Comment/Uncomment block	Code Assistants	114 times	2 months ago
Comment/Uncomment current line	Code Assistants	761 times	one week ago

Code Completion **Ctrl+Space** helps you quickly complete code statements. It works as you type and gives a list of suggestions available from the current caret position:

```
try {  
    service = Executors.newSingleThreadExecutor();  
    f = service.s~  
} finally { m submit(Runnable task) Future<?>  
    if (servi m submit(Callable<T> task) Future<T>  
    System.out m submit(Runnable task, T result) Future<T>  
} m shutdown() void  
while (!f.isD Press ⌘ to insert, ⌫ to replace  
    try {
```

IDE features help!

We're saving insane amount of time: preventing bugs, saving from typing, adding imports automatically...

Features specific for Data Engineers

Structure of dataframe

```
✓ └─ df3 = {org.apache.spark.sql.Dataset[iw$Person]} schema = id:IntegerType, name:StringType, age:IntegerType, country:StringType
  ✓ └─ schema() = {org.apache.spark.sql.types.StructType} size = 4
    > └─ 0 = {jvm:org.apache.spark.sql.types.StructField} id:IntegerType (non-nullable)
    > └─ 1 = {jvm:org.apache.spark.sql.types.StructField} name:StringType (nullable)
    > └─ 2 = {jvm:org.apache.spark.sql.types.StructField} age:IntegerType (non-nullable)
    > └─ 3 = {jvm:org.apache.spark.sql.types.StructField} country:StringType (nullable)
    01 └─ getStorageLevel() = {org.apache.spark.storage.StorageLevel} "StorageLevel(1 replicas)"
  > └─ df2 = {org.apache.spark.sql.DataFrame} schema = id:IntegerType, name:StringType, age:IntegerType, country:StringType
  > └─ df1 = {org.apache.spark.sql.DataFrame} schema = id:IntegerType, name:StringType, age:IntegerType, country:StringType
```

This information is not available in any environment but Big Data Tools

The same applies to Python
[@asm0di0](#) [@BigDataToolsJB](#)

Hadoop Monitoring

Cluster info		Cluster metrics info		Scheduler info	
Name	Value	Name	Value	Name	Value
Ha state	active	Active nodes	1	capacities	com.jetbrains.hadoop.monito...
Ha zoo keeper connection state	Could not find leader el...	Allocated mb	0	capacity	100.0
Hadoop build version	2.8.5 from 0b8464d75...	Allocated virtual cores	0	health	com.jetbrains.hadoop.monito...
Hadoop version	2.8.5	Apps completed	17	maxCapacity	100.0
Hadoop version built on	2018-09-10T03:32Z	Apps failed	6	queueName	root
Id	1593079285610	Apps killed	0	queues	CapacitySchedulerQueueInf...
Resource manager build version	2.8.5 from 0b8464d75...	Apps pending	0	usedCapacity	0.0

Spark Monitoring

Spark monitoring: Spark monitoring connection

Limit: 100 ▾ Started: Any ▾

Jobs Stages Environment Executors Storage SQL

App id	Name	Status	Id	Status	Name	Submission time	Num completed tasks	Id	Name	Tasks (1)
local-1607514451822	Zeppelin	▶	24	✖	takeAsList at Spar...	2021-01-21 12:4...	5/6	36	takeAsList at Spark2Shim	Id Index Launch time Host S...
			23	✖	takeAsList at Spar...	2021-01-21 12:4...	91/91	35	takeAsList at Spark2Shim	22... 0 2021-01-... loc... fa...
			22	✓	takeAsList at Spar...	2021-01-21 12:4...	201/201			▼ Tasks summary
			21	✓	takeAsList at Spar...	2021-01-21 12:4...	201/201			Metric
			20	✓	takeAsList at Spar...	2021-01-21 12:4...	201/201			Executor cpu time
			19	✓	takeAsList at Spar...	2021-01-21 12:4...	201/201			Executor deserialize cpu time
			18	✓	takeAsList at Spar...	2021-01-21 12:4...	201/201			Executor deserialize time
			17	✓	collect at <consol...	2021-01-19 15:2...	1/1			Executor run time
			16	✓	show at <console>...	2021-01-11 12:4...	1/1			Peak execution memory
			15	✓	takeAsList at Spar...	2020-12-30 11:...	201/201			Result serialization time
			14	✓	takeAsList at Spar...	2020-12-30 11:...	201/201			Result size
			13	✓	takeAsList at Spar...	2020-12-30 11:...	201/201			

▶ Settings ⏪ Refresh: every 30 seconds ⚙

Conclusion

Conclusion

- Smart features
 - Autocompletion
 - Data introspection
- Monitoring
 - Hadoop
 - Spark
- Work with popular Object Storages
- Single point of integration

Big Data Tools plugin

Thank you!

Pasha Finkelshteyn, JetBrains

 [asm0di0](#)

 [BigDataToolsJB](#)

Welcome to Q&A