

ODSC West 2024 WORKSHOP

**Building Big Data Workflows: NiFi, Hive, Trino,
& Zeppelin**

Instructor: **Khaled Tannir PhD**



Your Instructor



Khaled TANNIR PhD

Doctorate in Artificial Intelligence

Senior Big Data Engineer & Course Lecturer

25+ years of experience



20+ Certificates

odscw24@dataxper.com



Mastering Data Ingestion Using Apache Nifi

Khaled TANNIR

To Be
Announced



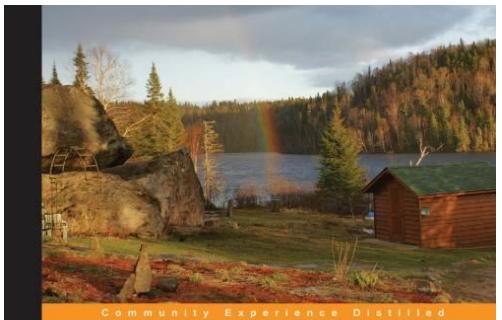
RavenDB 2.x

Build high performance NoSQL .NET-based applications
quickly and efficiently

Beginner's Guide

Khaled Tannir

[PACKT] open source*



Optimizing Hadoop for MapReduce

Learn how to configure your Hadoop cluster to run optimal
MapReduce jobs

Khaled Tannir

[PACKT] open source*

Agenda

- *Workshop Overview*
 - *Overview of the Technical Environment*



- *Workshop Steps*
 - *Step 1 – Building The Nifi dataflow*
 - *Step 2 - Data Transformation & Storage*
 - *Step 3 – Data Exploration Using Hive and Trino / Zeppelin*
 - *Step 4 – Building the Superset Dashboard*



Workshop Overview

Building Big Data Workflows: NiFi, Hive, Trino, & Zeppelin

Create a data pipeline that:

Ingests, transforms and analyzes Nobel Prizes data using Apache
NiFi, Jolt, Hive, Trino and Superset.

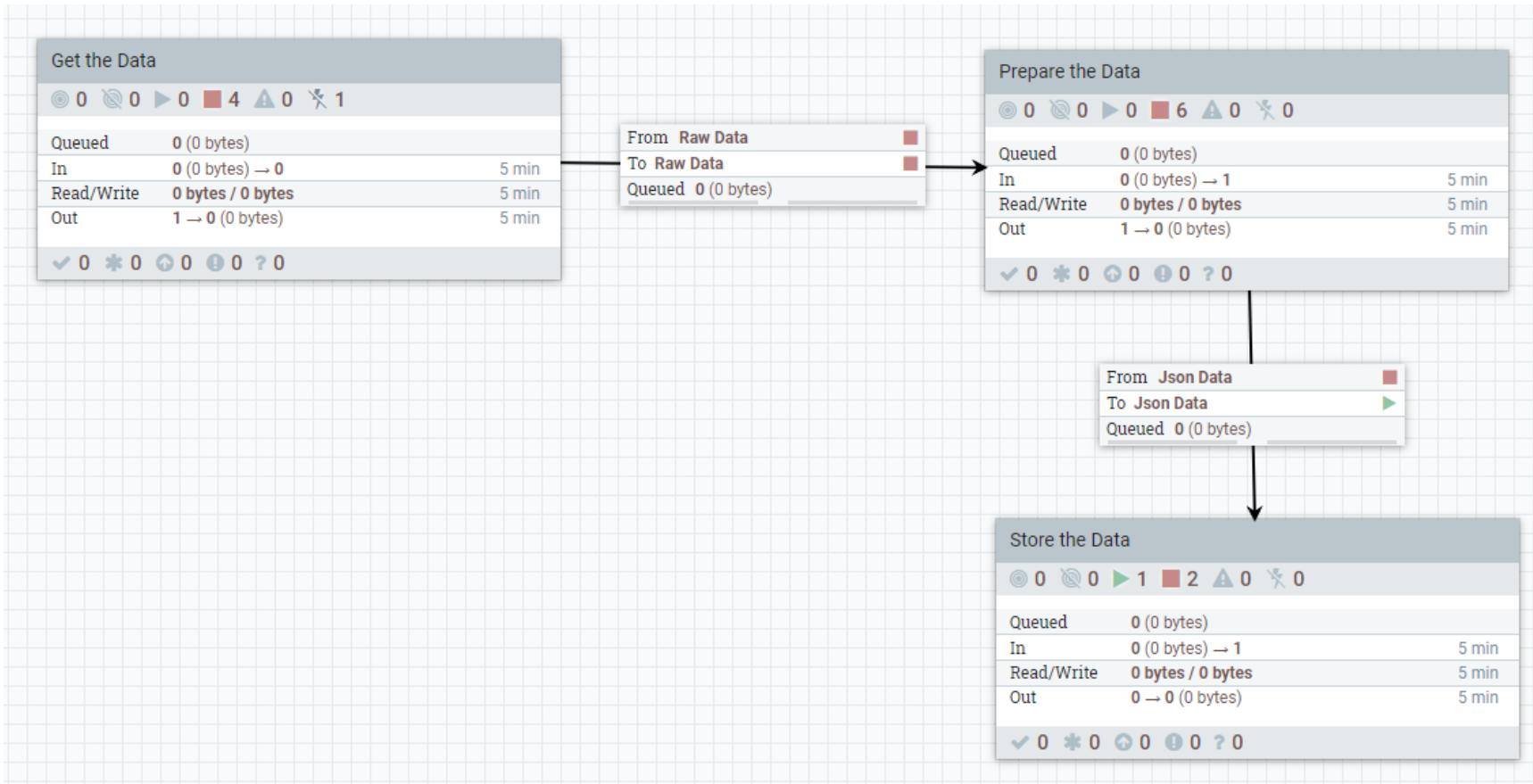


Total Duration: 1 hour

Workshop Step 1 – Data Ingestion

Part 1

- Connect to the Nobel Prizes REST API (data source) using Nifi



Workshop Step 2 – Data Storage

Part 2

- Transform and simplify the collected data and store it on HDFS

Browse Directory

/workshops/nifi/H3/laureates								Go!				
Show 25 entries								Search:				
<input type="checkbox"/>	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name				
<input type="checkbox"/>	-rw-r--r--	root	supergroup	31.24 KB	Sep 28 13:05	1	128 MB	laureates.snappy.parquet				
Showing 1 to 1 of 1 entries												
				Previous		1	Next					

Hadoop, 2022.

Workshop Step 3 – Explore the data

Part 3

- Access and load the data using Hive and use Trino to explore it using Zeppelin (5 queries)

```
%hive
select * from laureates.laureates limit 10;
```

FINISHED ▶ settings ▾

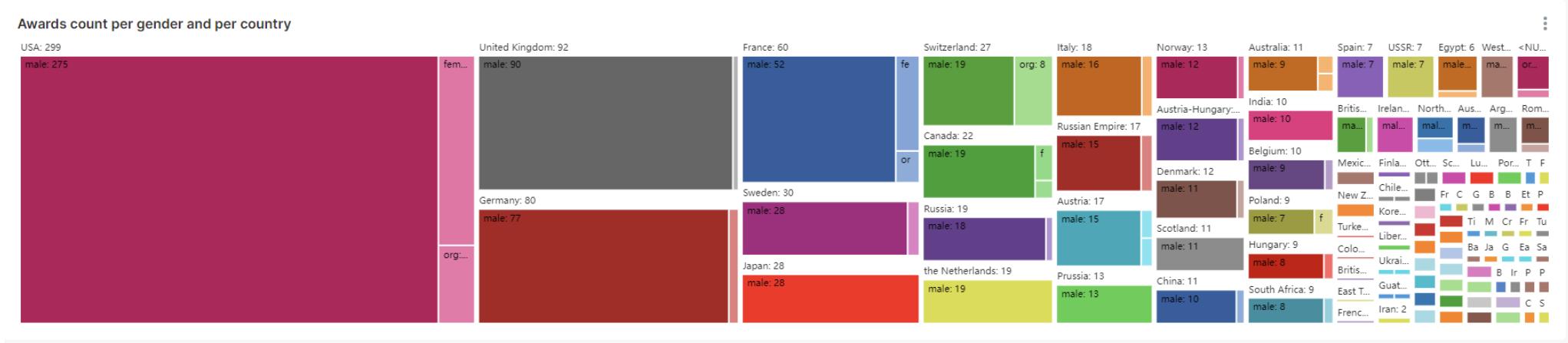
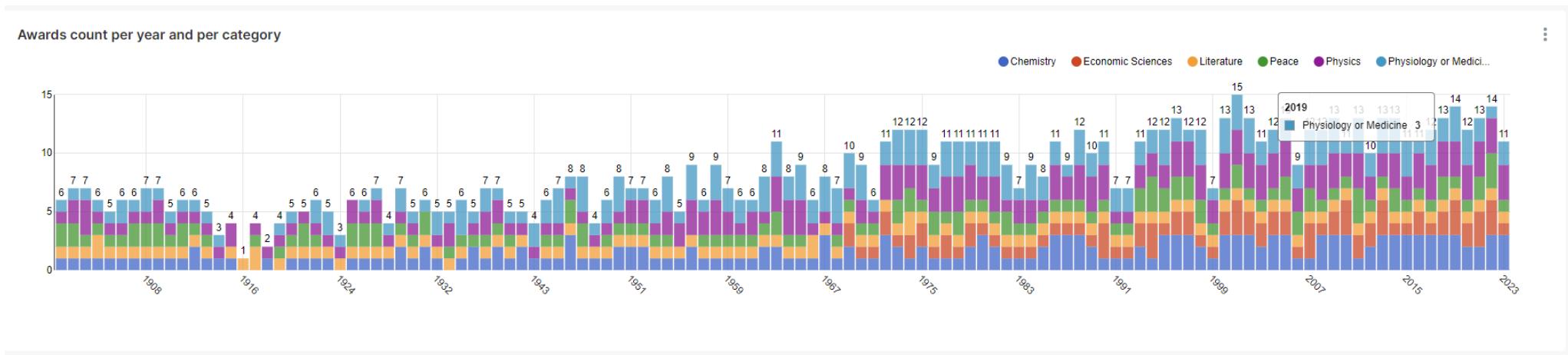
id	fullname	gender	date	country	awardyear	category
779	Aaron Ciechanover	male	1947-10-01	British Protectorate of Palestine	["2004"]	["Chemistry"]
259	Aaron Klug	male	1926-08-11	Lithuania	["1982"]	["Chemistry"]
1004	Abdulrazak Gurnah	male	1948-00-00	null	["2021"]	["Literature"]
114	Abdus Salam	male	1926-01-29	India	["1979"]	["Physics"]
982	Abhijit Banerjee	male	1961-02-21	India	["2019"]	["Economic Sciences"]
981	Abiy Ahmed Ali	male	1976-08-15	Ethiopia	["2019"]	["Peace"]
843	Ada E. Yonath	female	1939-06-22	British Mandate of Palestine	["2009"]	["Chemistry"]
866	Adam G. Riess	male	1960-12-16	USA	["2011"]	["Physics"]

Took 1 sec. Last updated by anonymous at September 28 2024, 1:05:40 PM.

Workshop Step 4 – Build The Dashboard

Part 4

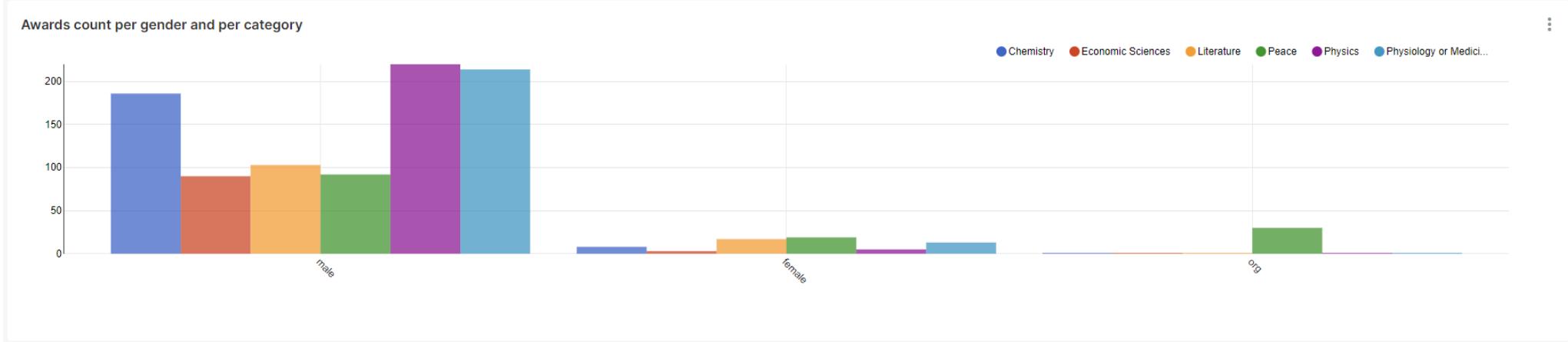
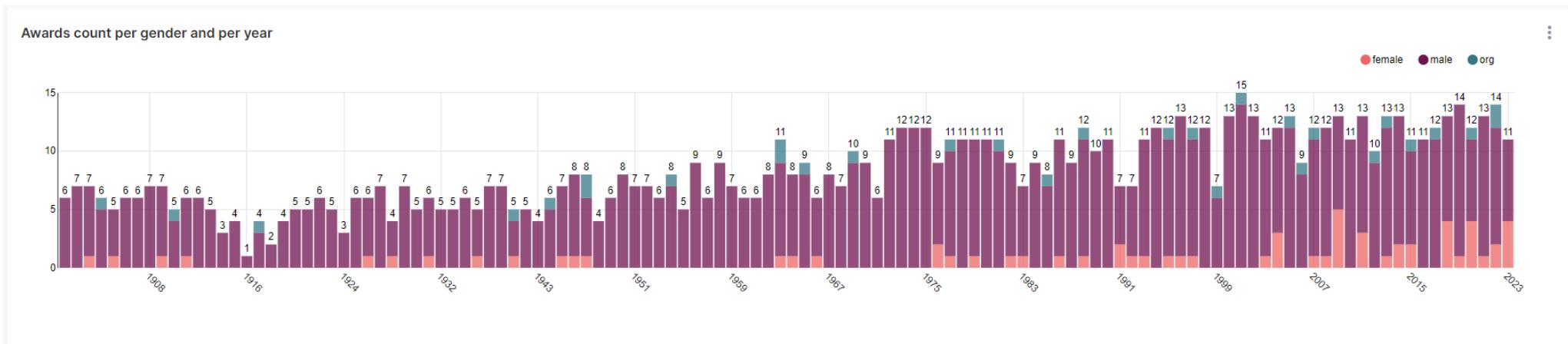
- Connect Superset to Trino and build a dashboard to visualize the data (5 charts)



Workshop Step 4 – Build The Dashboard

Part 4

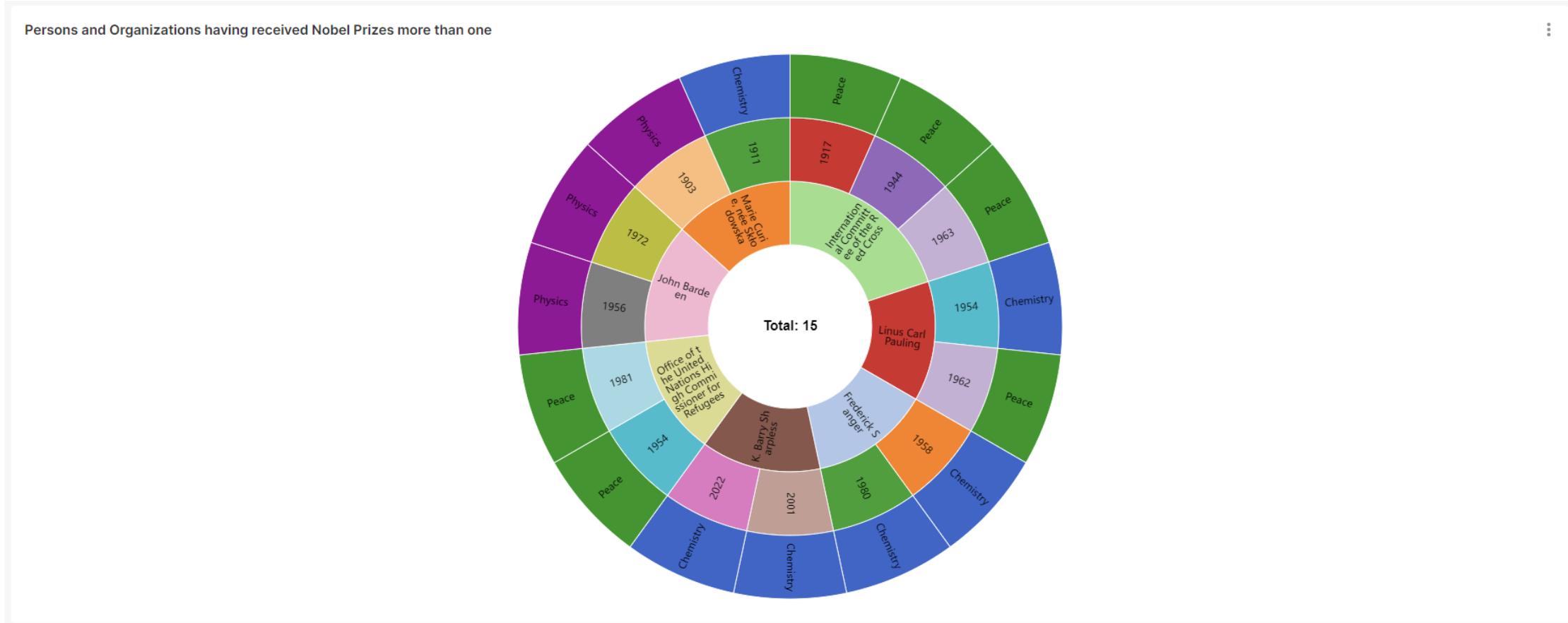
- Connect Superset to Trino and build a dashboard to visualize the data (5 charts)



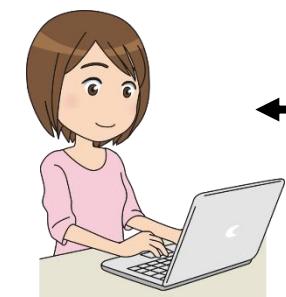
Workshop Step 4 – Build The Dashboard

Part 4

- Connect Superset to Trino and build a dashboard to visualize the data (5 charts)



Materials Needed - Virtual Machine - Local



Requires

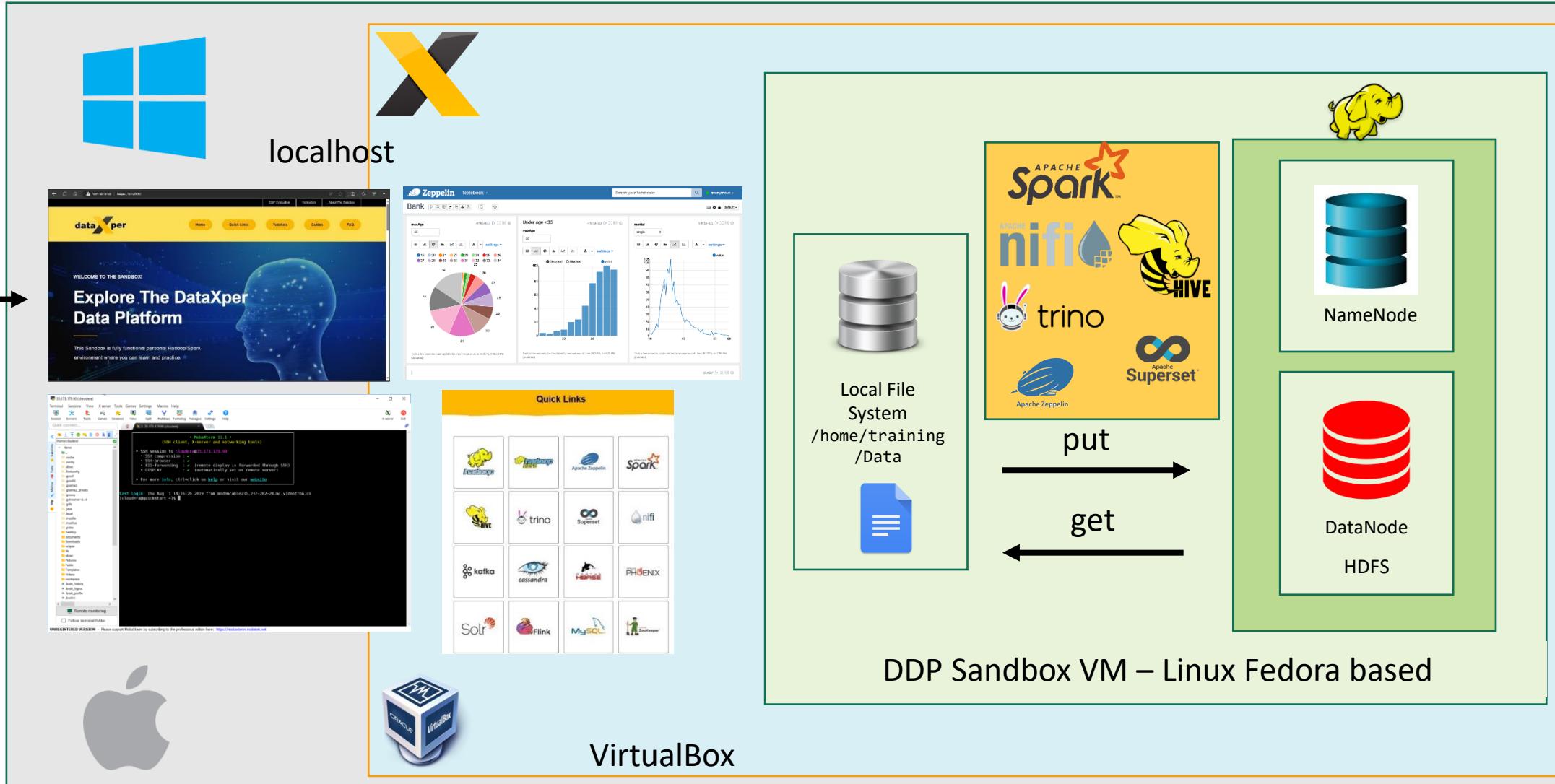
20 GB



8+ CPUS

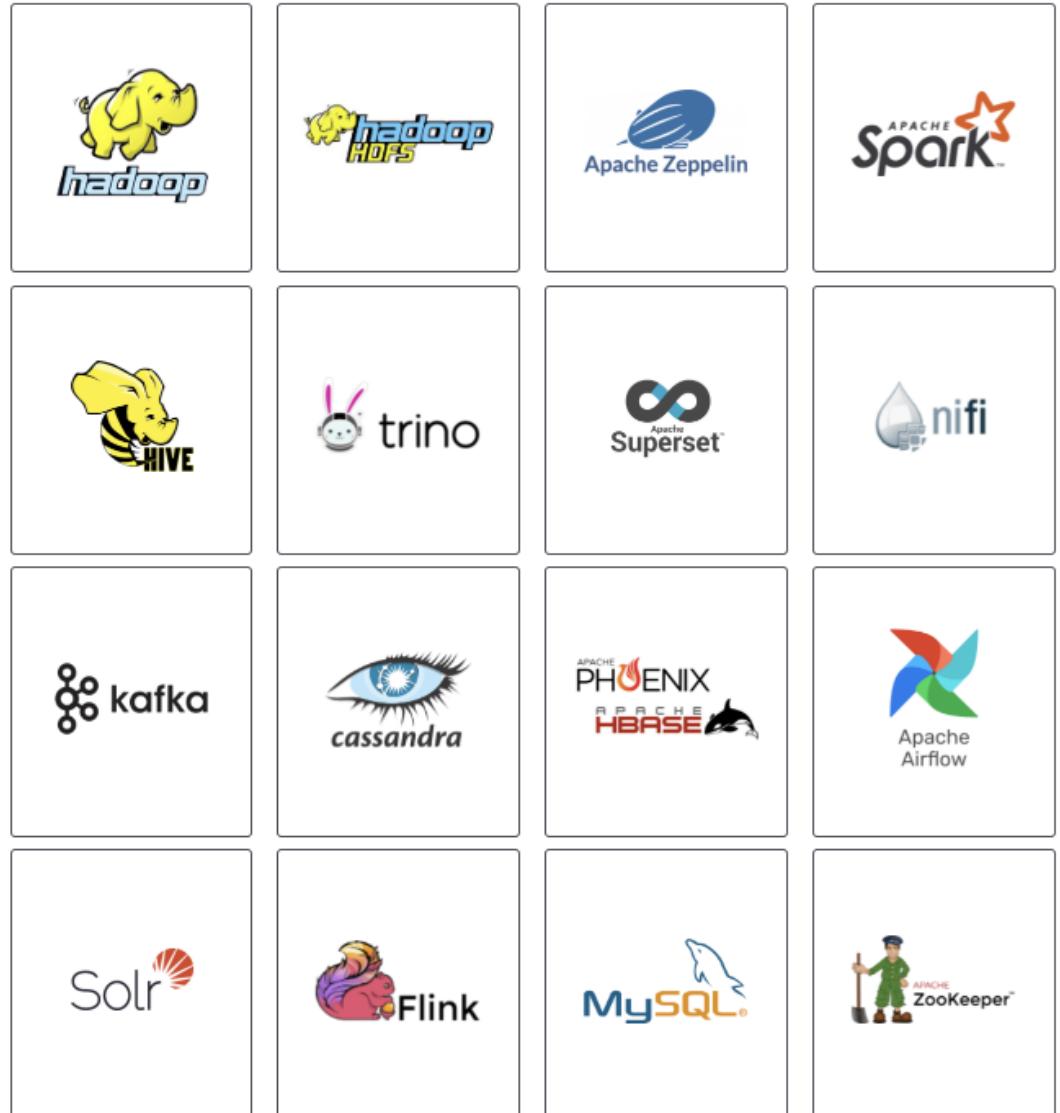


Mandatory

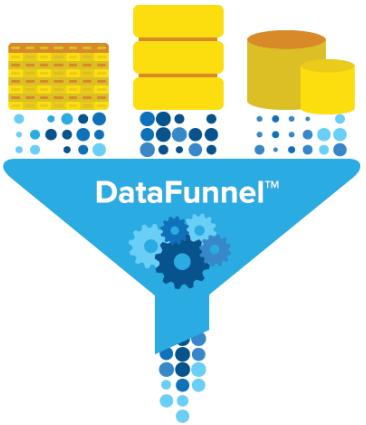


About the Sandbox

- DataXper Data Platform (DDP) Sandbox
- I Created The First Version in 2021
- Dedicated for Learning and Practicing
- **50+ Tutorials / Guides / Quizzes**
- **Full open-source**



Workshop Dataflow – Step 1



1

Collect

Collect data from
sources



Nobel Prizes Data



JSON



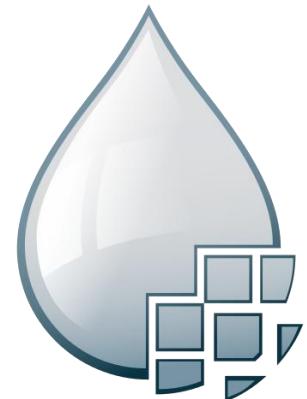


Apache NiFi Overview

Data Ingestion at Scale

Apache NiFi

- What is NiFi ?
- NiFi (short for “*Niagara Files*”) is an Open-Source dataflow tool that can collect, route, enrich, transform and process data in a scalable manner.
- It is a processing engine based on the concepts of *flow-based programming* (FBP), that was designed to manage the flow of information in an ecosystem





Nifi Terminology

FlowFile

- Unit of data moving through the system
- Content + Attributes (key/value pairs)

Processor

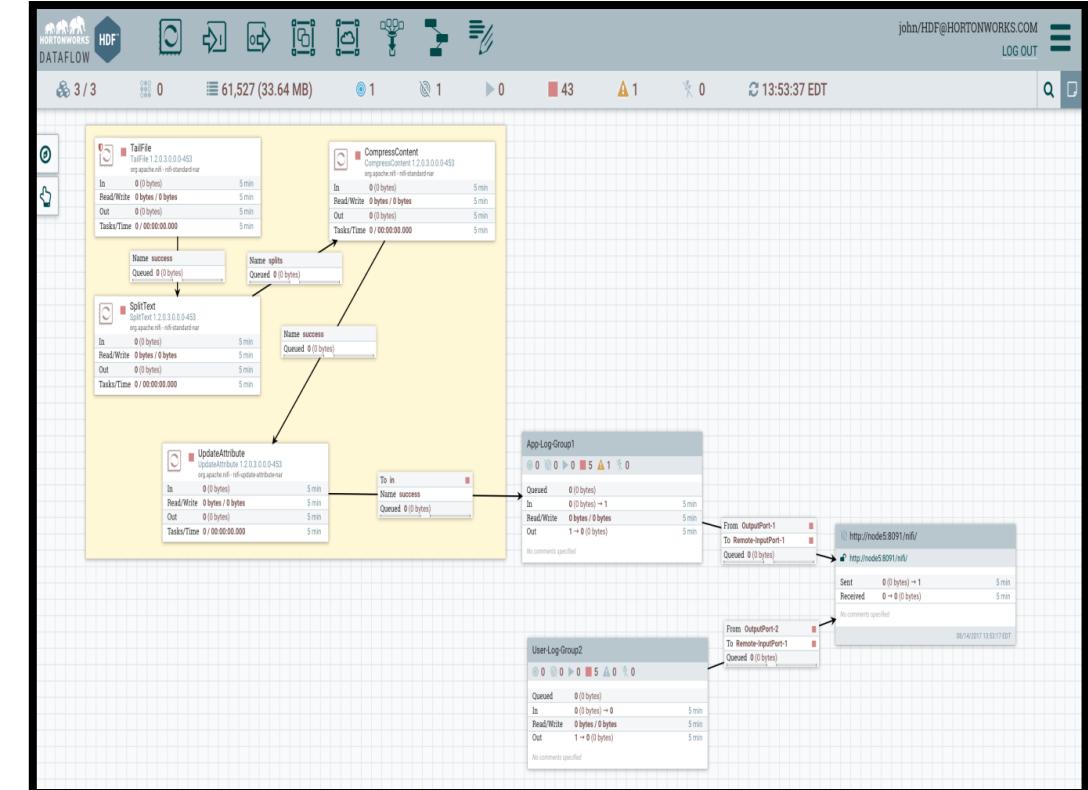
- Performs the work, can access FlowFiles

Connection

- Links between processors
- Queues that can be dynamically prioritized

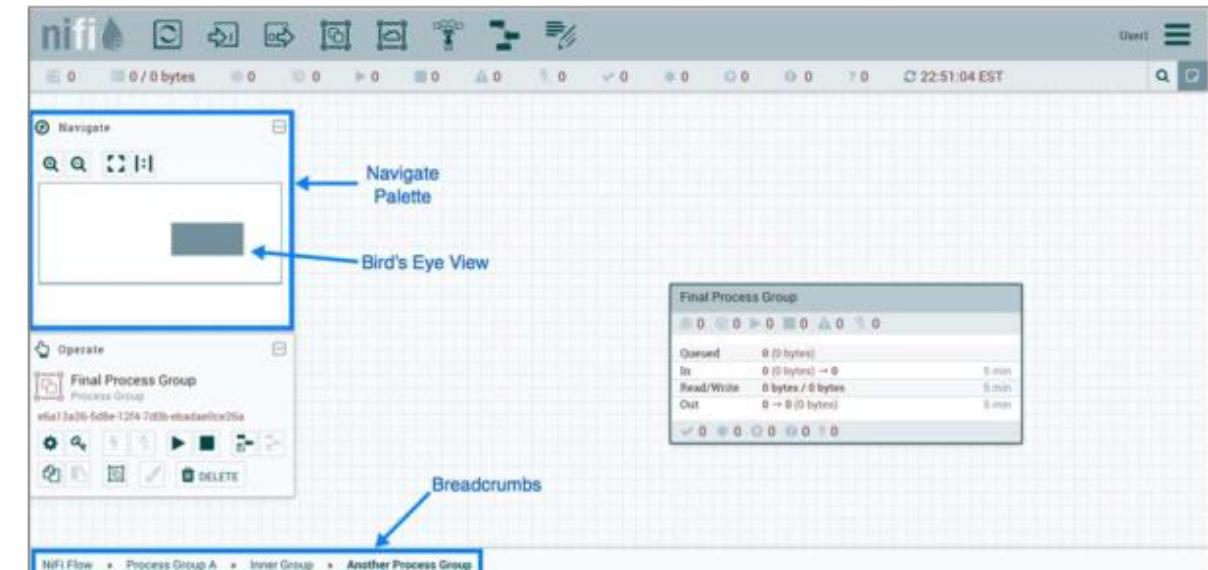
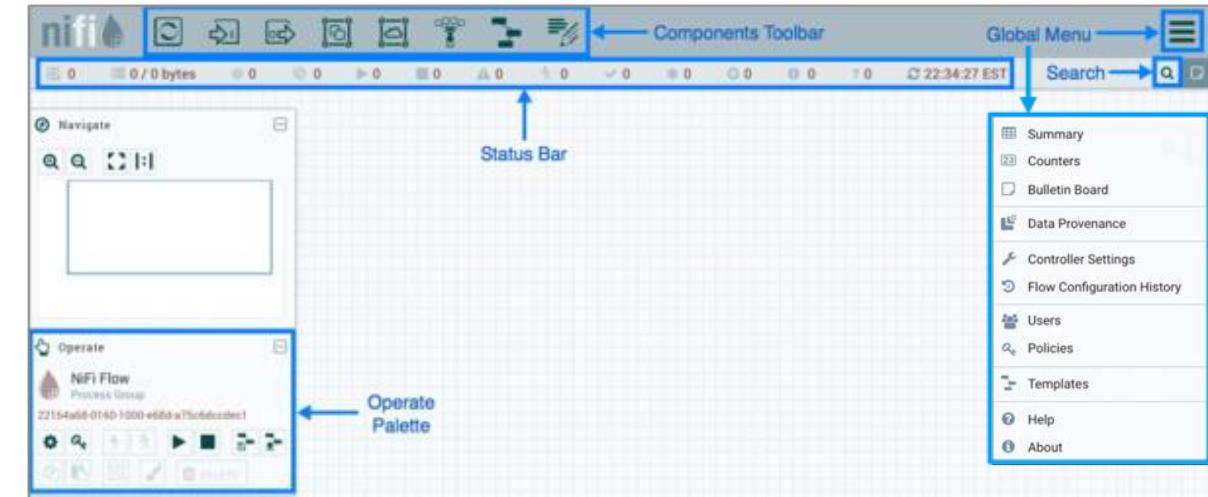
Process Group

- Set of processors and their connections
- Receive data via input ports, send data via output ports



Visual Command & Control

- Drag and drop processors to build a flow
- Start, stop, and configure components in real time
- View errors and corresponding error messages
- View statistics and health of data flow
- Create templates of common processor & connections



Hands-On - Nifi

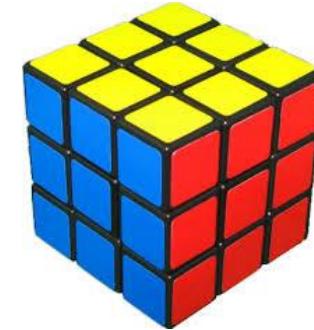
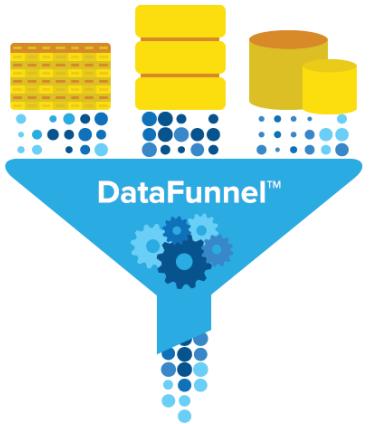


Workshop - NiFi Data Ingestion

- **Add Process Group**
 - **Add GenerateFlowFile Processeur**
 - **Add InvokeHTTP Processeur**
 - **Requires StandardSSLContextService**
 - **UpdateAttribute**
 - **Output Port**



Workshop Dataflow – Step 2



1

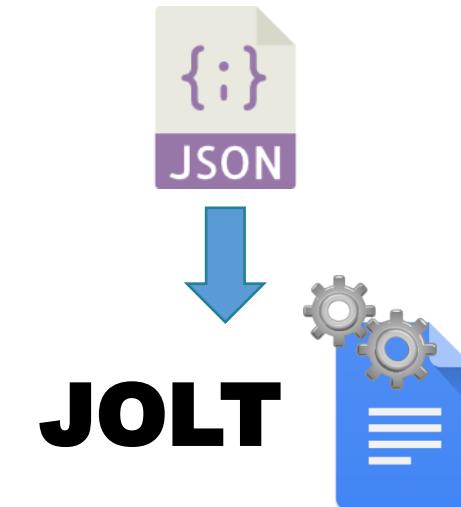
Collect

Collect data from sources

2

Organize

Prepare the data

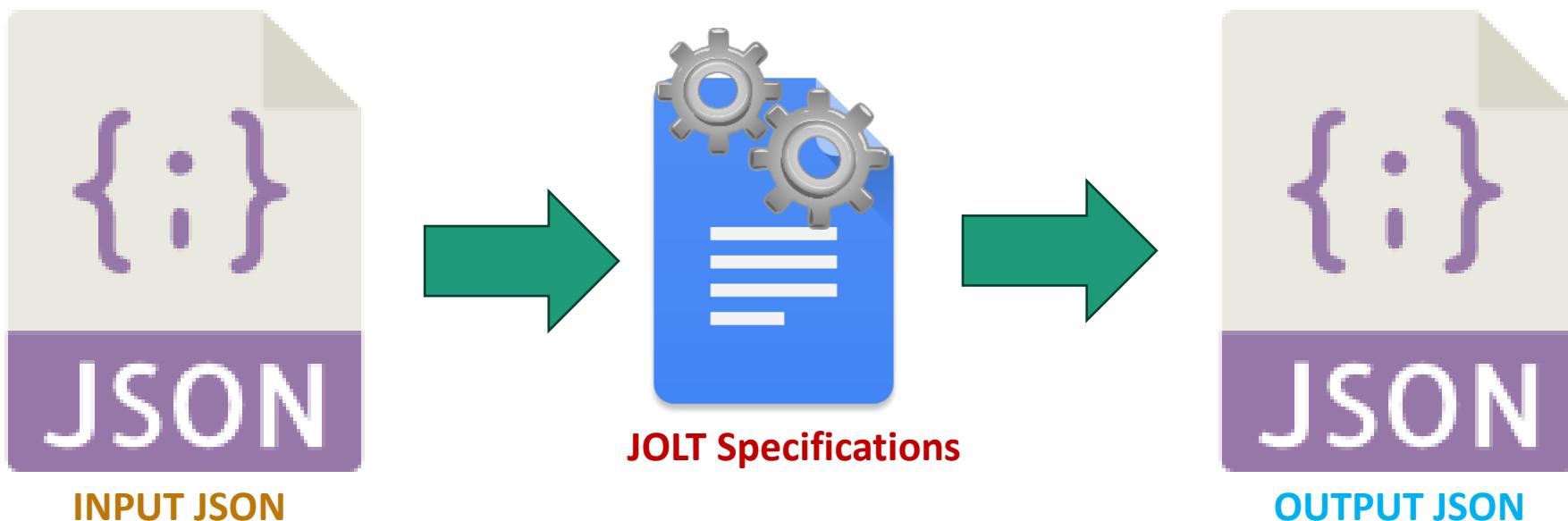


JOLT

JsOn Language for Transformations

JOLT Transformations

- **JsOn Language for Transformation**
- **Java library**
- **Transform one JSON structure to another**



JOLT Specifications

JOLT Specifications Structure



The screenshot shows a web-based JSON validator tool with two tabs: "Jolt Spec" (selected) and "JSON Validate". The code editor contains the following JOLT specification:

```
1 [  
2   {  
3     "operation": "shift",  
4     "spec": {  
5       //-- spec goes here  
6     }  
7   },  
8   {  
9     "operation": "remove",  
10    "spec": {  
11      //-- spec goes here  
12    }  
13  }  
14 ]  
15
```

<http://jolt-demo.appspot.com/#inception>

JOLT Operations

- **shift:** Pic value from one node drop in another node.
- **default:** If null in input, put default in output.
- **remove:** Remove node from output.
- **cardinality:** Convert one to array, **and**, array to one.
- **sort** Sorts all arrays and maps from the input
- **modify-default-beta:** Modify if value is null or un-available.
- **modify-overwrite-beta:** Modify even if value is available.

JOLT – Shift Example

Json Input [JSON Validate](#)

```
1 [
2   "country": "Canada",
3   "population": "36624199",
4   "sq_km_area": 9970610,
5   "life_expectancy": "79.4",
6   "elevation_in_meters": "487",
7   "continent": "North America",
8   "abbreviation": "CA",
9   "location": "North America",
10  "iso": "124",
11  "capital_city": "Ottawa"
12 ]
```

shift One to One Operation:

Take the value from one node and put it into another node

Jolt Spec [JSON Validate](#)

```
1 [
2   [
3     {
4       "operation": "shift",
5       "spec": {
6         "country": "pays.nom",
7         "capital_city": "pays.capitale"
8       }
9     }
10   ]
]
```

Output / Errors [Transform](#) Sort Output?

```
1 [
2   "pays" : {
3     "nom" : "Canada",
4     "capitale" : "Ottawa"
5   }
6 ]
7 ]
```

Special Chars in JOLT

- ‘&’ Ampersand special character.
- ‘*’ Asterisk special character.
- ‘@’ At special character.
- ‘#’ Hash special character in LHS and in RHS (*Left- and Right-Hand Side*).
- ‘|’ Pipe special character.
- Spec JSON content that is:
 - Before : (colon) will be **LHS** (or **Input Side**). **LHS** is node structure to pic to be transformed from input JSON
 - After : colon will be **RHS** (or **Output Side**) . **RHS** is . (Dot) separated JSON path formatted structure

Special Chars in JOLT

Special Char	Used In	Operations	Use
&	Output	shift	Refer to current LHS node name &1 One level above current LHS node &(2) To level above current LHS ...
*	Input	shift, remove, cardinality, modify-default-beta modify-overwrite-beta	Iterate over items.
@	Input Output	Shift(LHS and RHS) modify-default-beta (RHS) modify-overwrite-beta (RHS)	Lookup value in input json @ given node level.
\$	Input	shift	Only field names.
#	Input Output	shift	In LHS Define a default value In RHS define the way object Array is created.
	Input	shift	Input fields with uncertain name pointing to some output (OR Operator) .

Hands-On -Data Prep.



Workshop - NiFi Data Transformation

- **Add Process Group**
 - **Add Input Port**
 - **Add JoltTransformJson Processor**
 - **Add MergeContent Processor**
 - **Add Output Port**

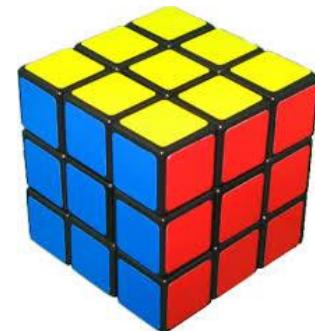
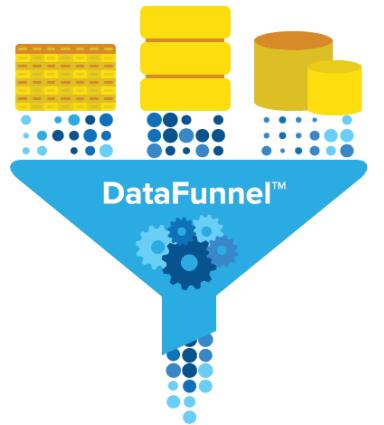


Workshop - NiFi Data Storage

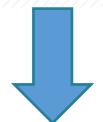
- **Add Process Group**
 - **Add Input Port**
 - **Add UpdateAttribute**
 - **Add PutParquet Processor**



Workshop Dataflow – Step 3



Parquet



1 **Collect**

Collect data from sources

2

Organize

Prepare the data

3

Analysis

Extract Knowledge from the data





Data Warehouse on top of Hadoop

What is Apache Hive?

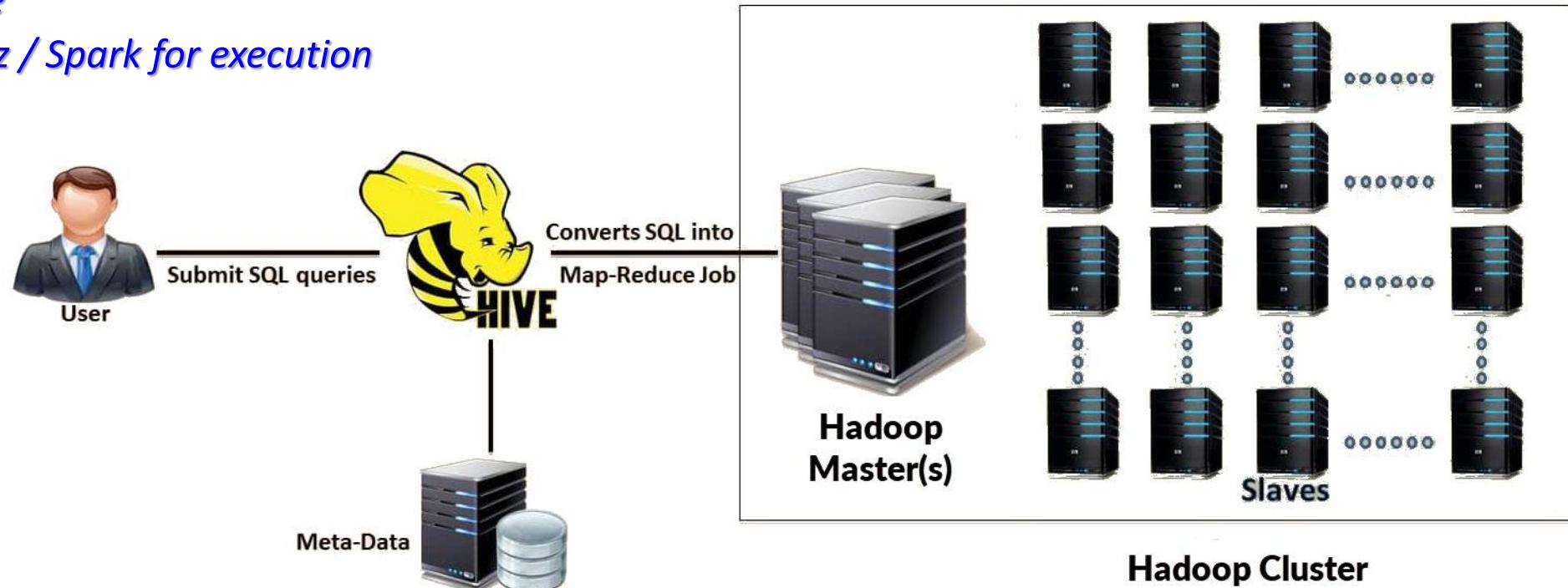
- A Data Warehouse on top of Hadoop (**SQL** on Hadoop).
- Can deal with different **storage** and **file** formats using ***Input/Output*** and ***SerDes***.
- Summarize Big Data and makes querying and analyzing easy.
- Familiar, scalable, and extensible.
- Written in Java and open-source



How it Works?

- **Hive is built on top of Hadoop**

- *Uses HDFS for storage*
- *Uses MapReduce / Tez / Spark for execution*



- **Hive compile HiveQL queries into MapReduce / Tez / Spark jobs and run the jobs in the Hadoop cluster**

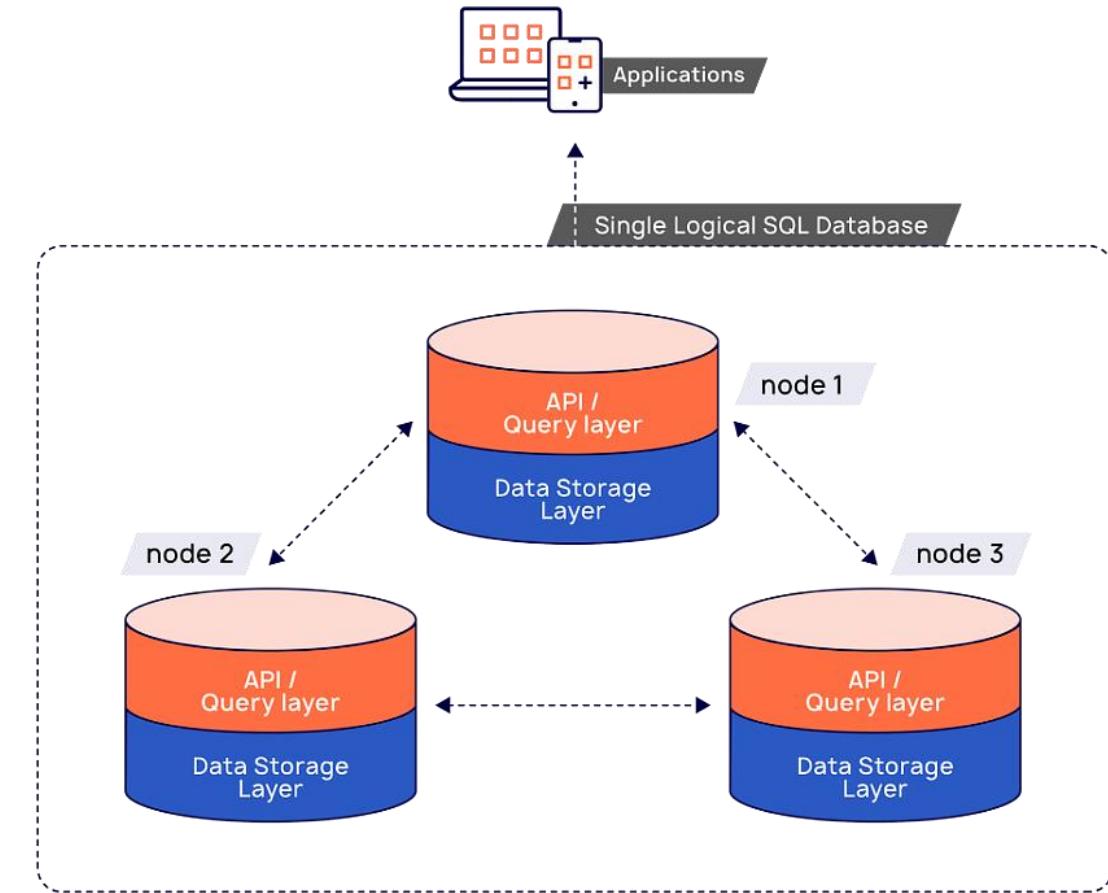


trino

Querying Data at Scale Interactively

What is Apache Trino?

- **Distributed SQL query engine**
 - ANSI SQL on Databases, Data lakes
 - Designed to be **interactive**
 - Access to petabytes of data
- **Open-source, Extensible**
- **Written in Java**



Apache Trino is Different

- **Trino is not a database**

You can't store data in Trino. It uses connectors to connect to an existing database.

- **Runs **interactive** queries on data in HDFS, and others.**

Is complimentary to Hadoop, it does not have its own storage system. Can run over Yarn.

- **It is not a data warehouse**

You can aggregate terabytes of data across multiple data sources and run efficient ETL queries.



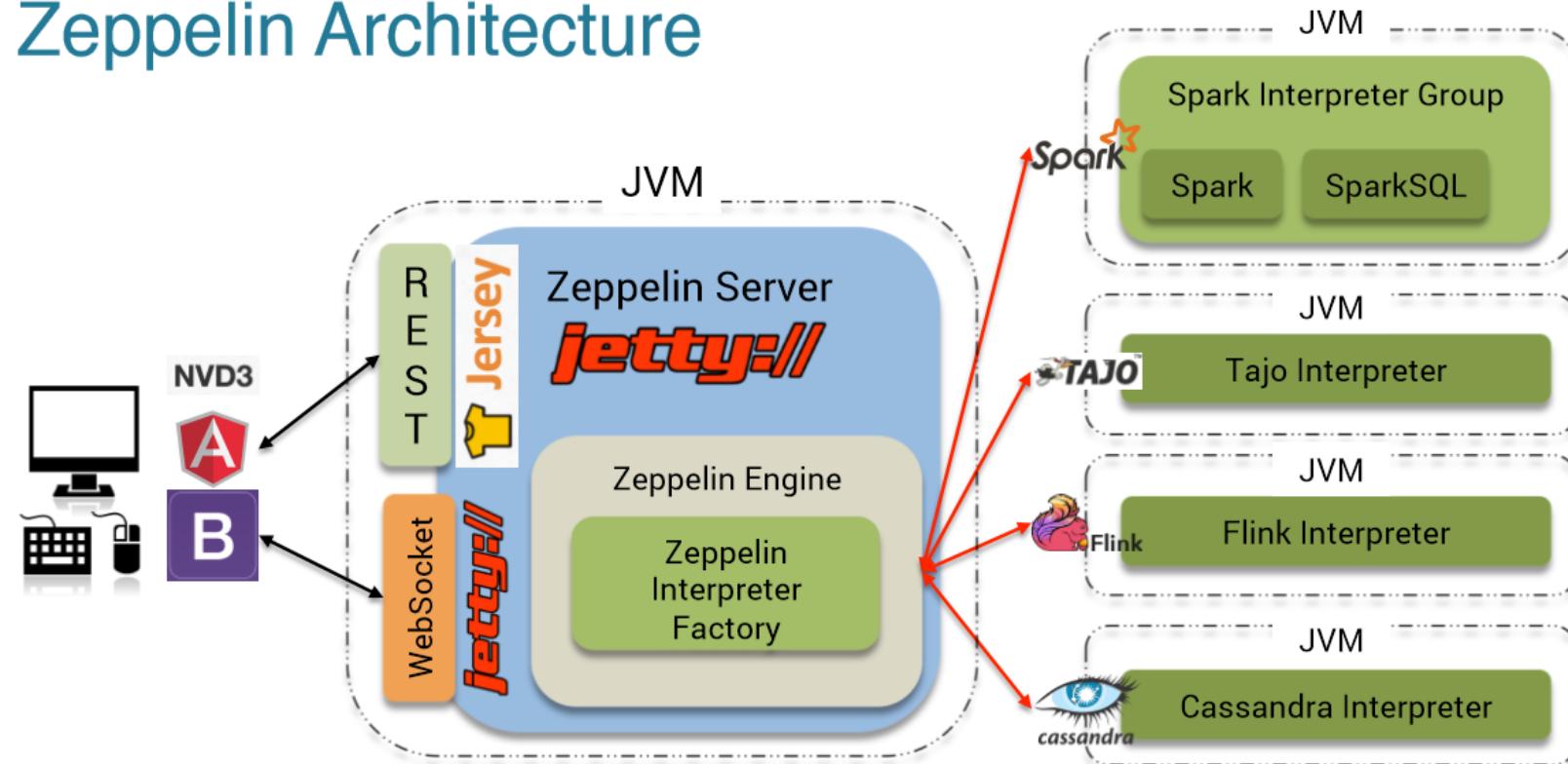
Apache Zeppelin

A web-based notebook for interactive analytics

What is Apache Zeppelin?

- A web-based notebook for interactive analytics
- Deeply integrated with Spark and Hadoop
- Supports multiple language backends

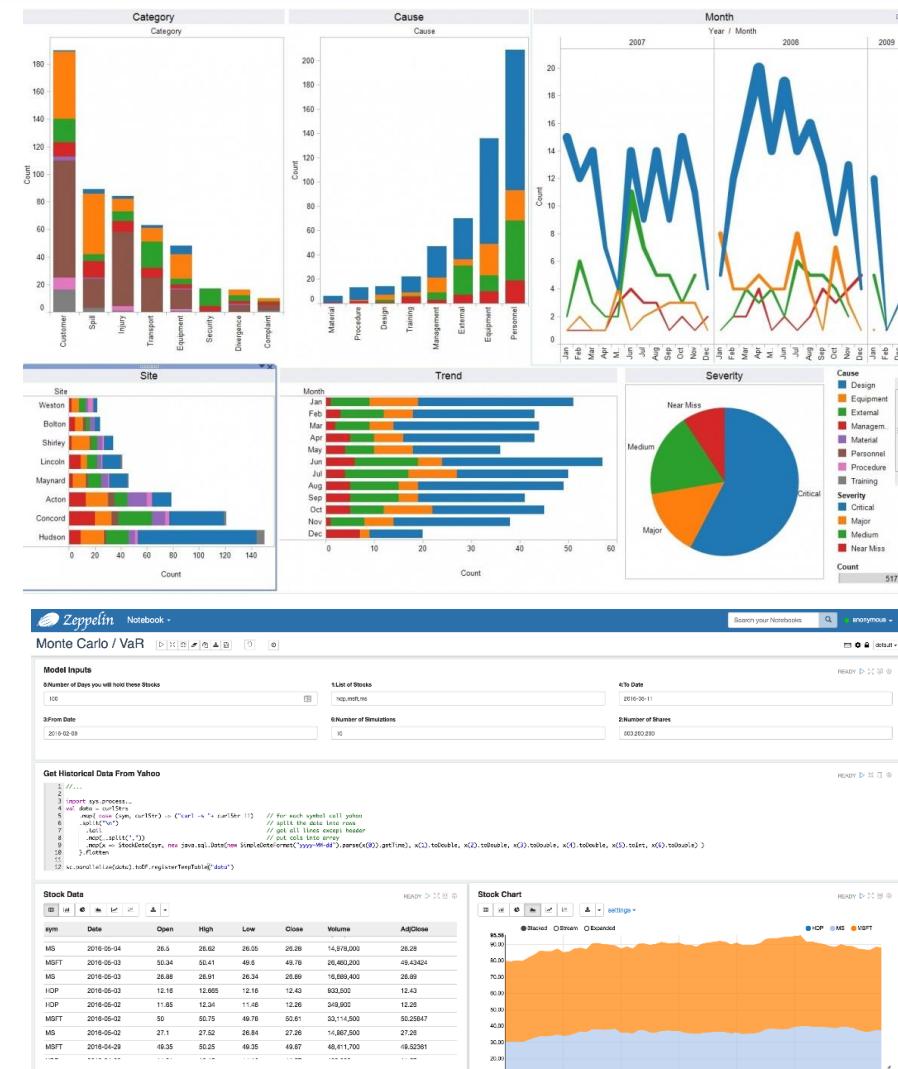
Zeppelin Architecture



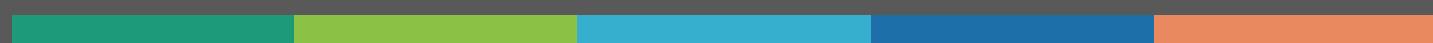
What is a Zeppelin Note/Notebook?



- A web based graphical user interface (GUI) for small pieces of code
- Write the code in a browser
- Zeppelin sends the code to the backend for execution and retrieves the resulting data
- Zeppelin visualizes the data
- Zeppelin Note = Set of (Paragraphs / Cells)
- Other Features - Sharing / Collaboration / Reports / Import / Export



Hands-On – Data Analysis



Workshop – Hive / Trino Data Analysis

- **Open Zeppelin and Create a new Note**

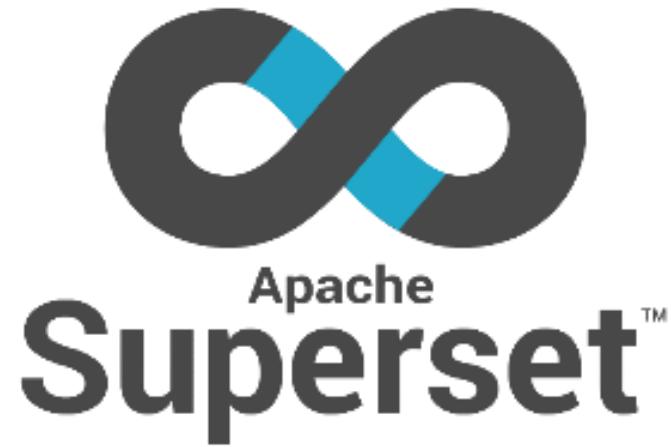


- **Create Hive Database + Table**
- **Prepare the final Parquet table**
- **Use Trino to explore the Hive table (5 queries)**



Workshop Dataflow – Step 4



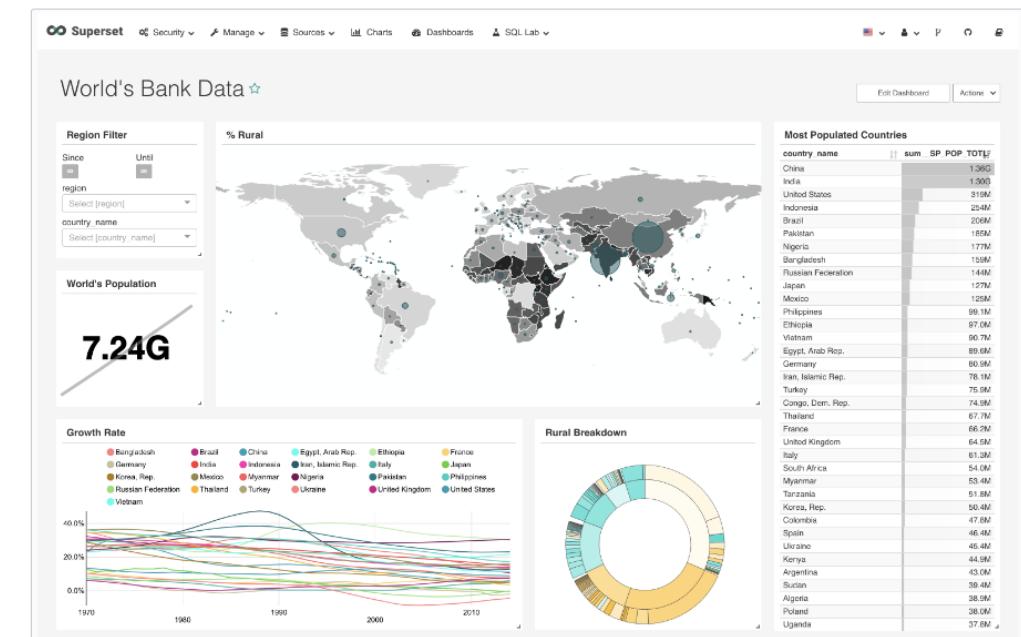


Data Visualization and Exploration Platform

What is Apache Superset



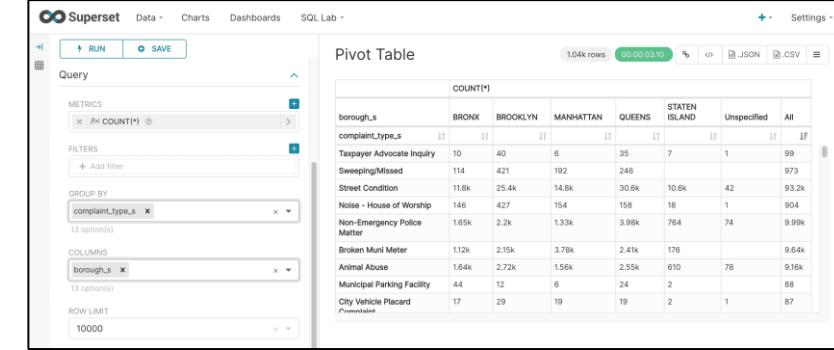
- A data visualization and exploration platform
- Easy-to-use & fast “time-to-dashboard”
- Enterprise-ready & cloud-native
- Rich set of visualizations (50+)
- Lightweight semantic layer
- Works with a wide array of databases



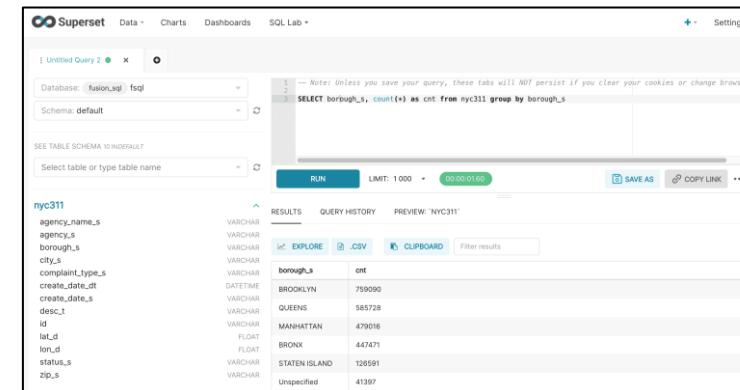
Apache Superset Components



Visual Dashboard Building



Data Exploration



SQL Powerhouse

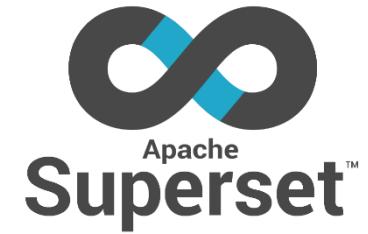
Dr. Khaled Tannir

Hands-On – Dashboard



Workshop – Superset Dashboard

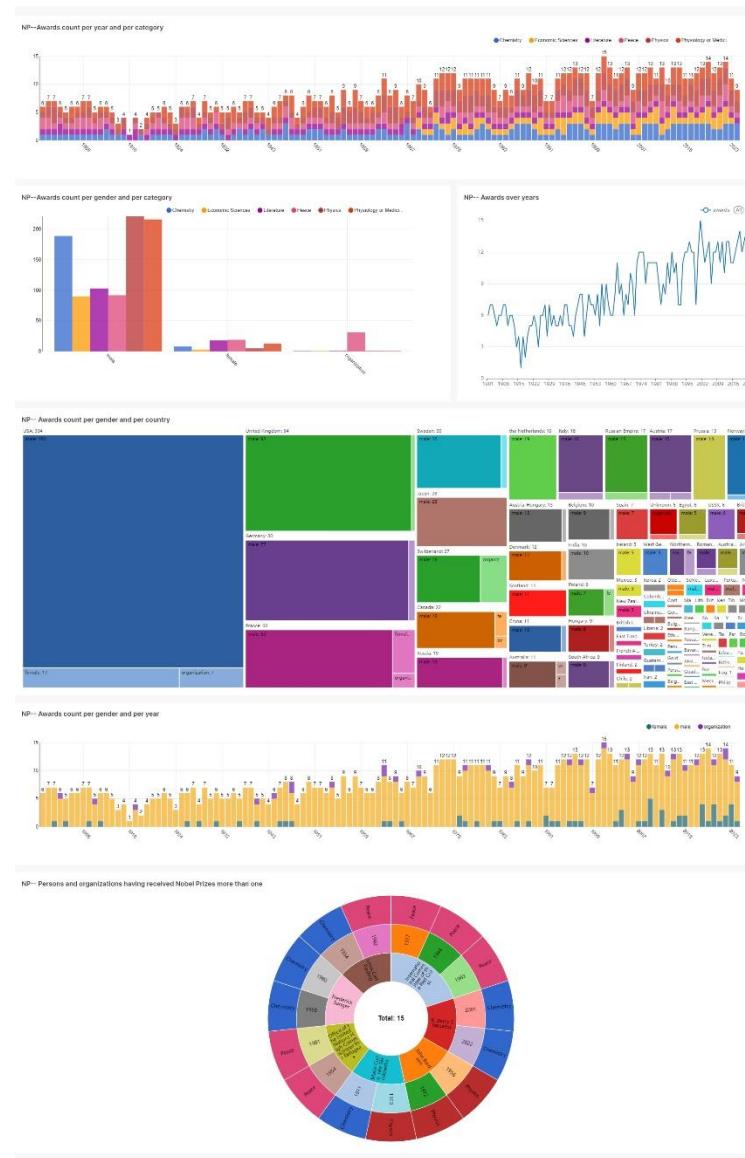
- **Open Superset and Connect to Trino**
 - **Create Queries Hive Database + Table**
 - **Create Charts and add to the Dashboard**



End of the Workshop



Workshop – The Final Dashboard



Questions?



Merci!
Thank You!

