

MATLAB EXPO

November 13–14, 2024 | Online

MATLAB & Databricks: Accelerating Data Science and Machine Learning

Zachary Jacobson, Databricks

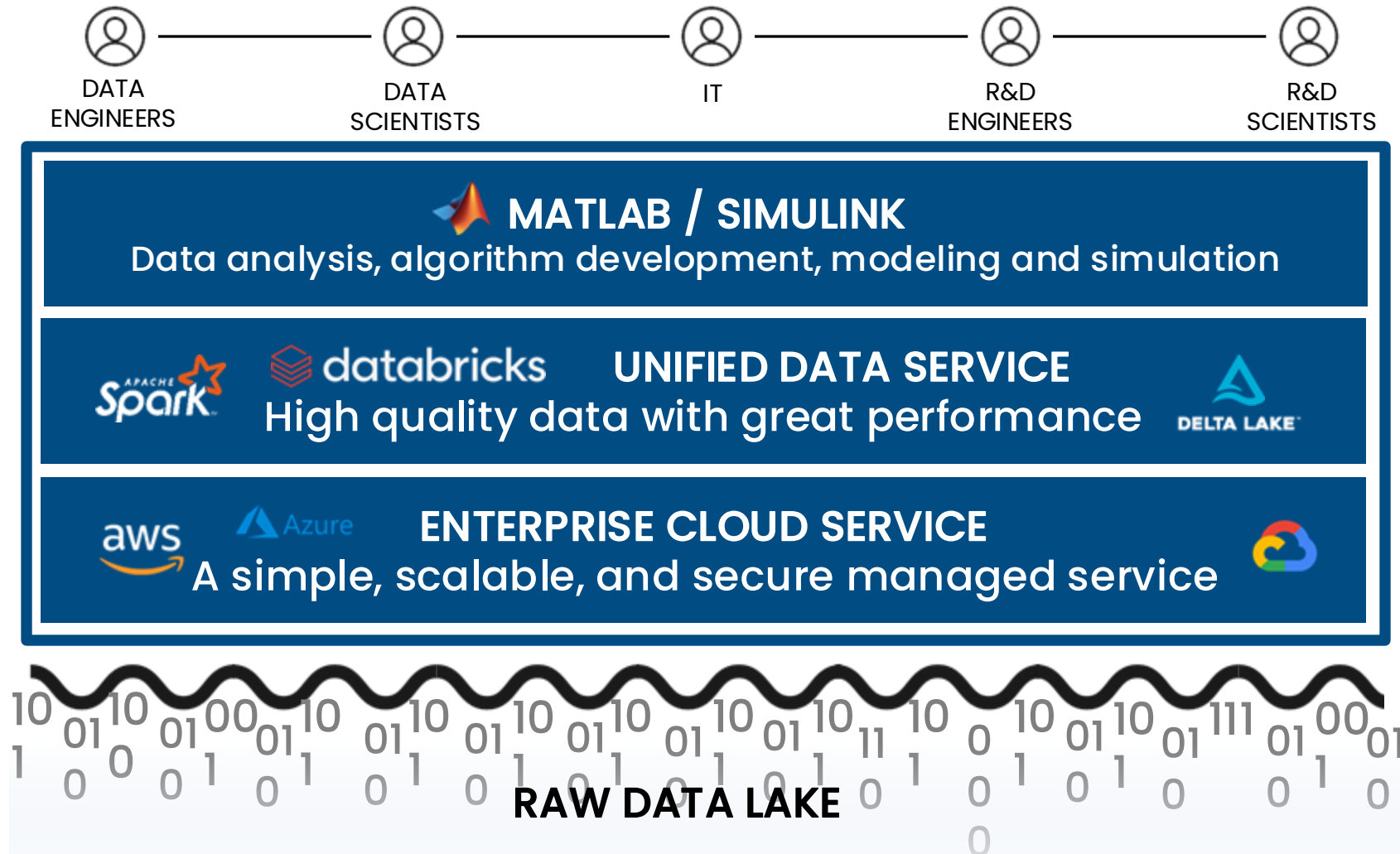


Andy Thé, MathWorks



Integration Approach

“The entire company on one data and compute platform in the cloud”

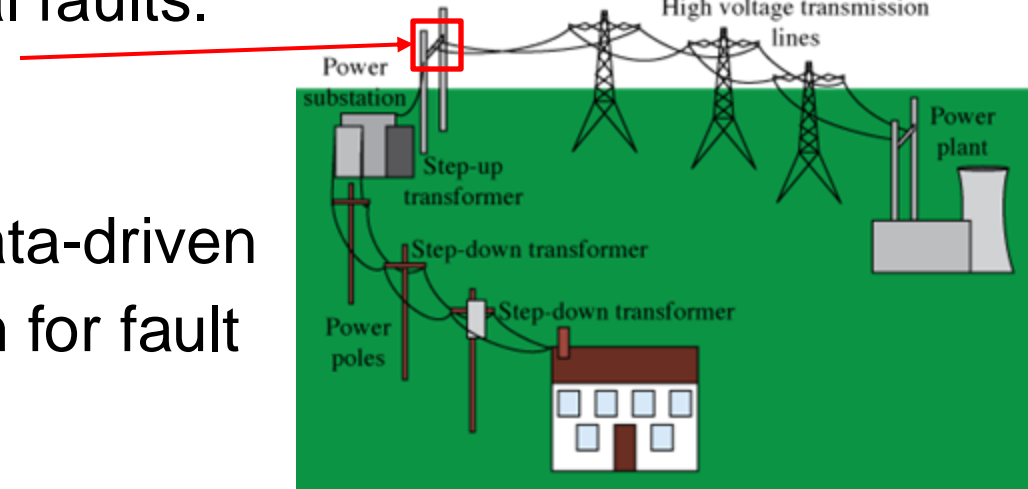


The Problem



Utility companies waste thousands of hours trying to identify and locate damaged power lines caused by electrical faults.

Fault

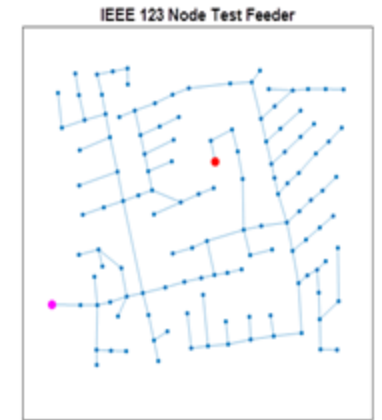


With MathWorks and Databricks you can use data-driven techniques to generate a classification algorithm for fault location detection on electric grids.

This will greatly help utility companies handle power outages.

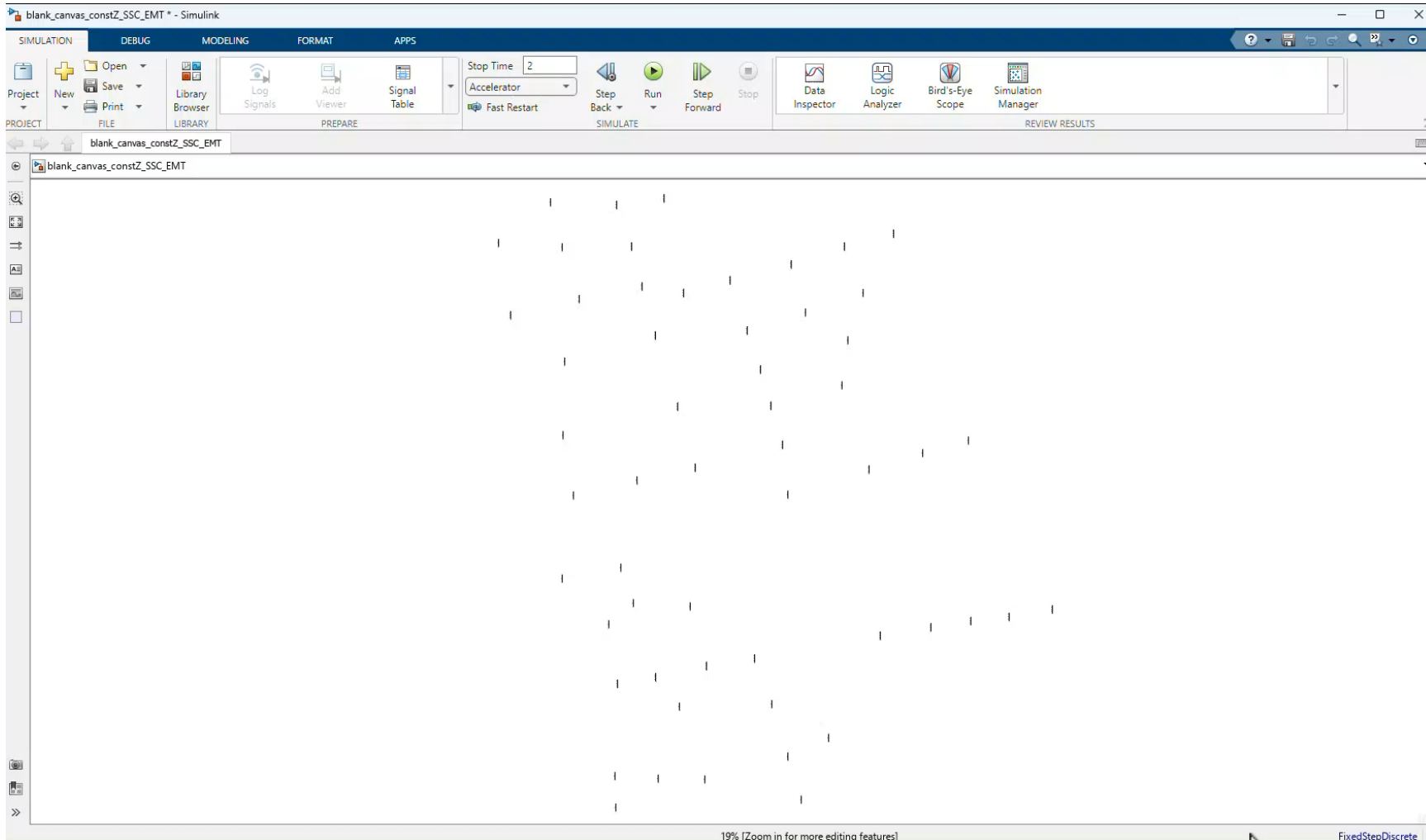
The Approach

1. Create a model to generate synthesized data
 - ❑ *IEEE 123 Simscape model*
2. Run the model at scale to generate large amounts of data
 - ❑ *Run the Simscape model in Databricks on 100+ instances*
3. Train a classification algorithm using the synthesized data
 - ❑ *Train a neural network in Databricks with the large data set*
4. Use the trained neural network to predict the locations when a fault occurs
 - ❑ *Input a fault into the neural network to get a distance from the power plant*



Create a simulated model to generate synthesized data

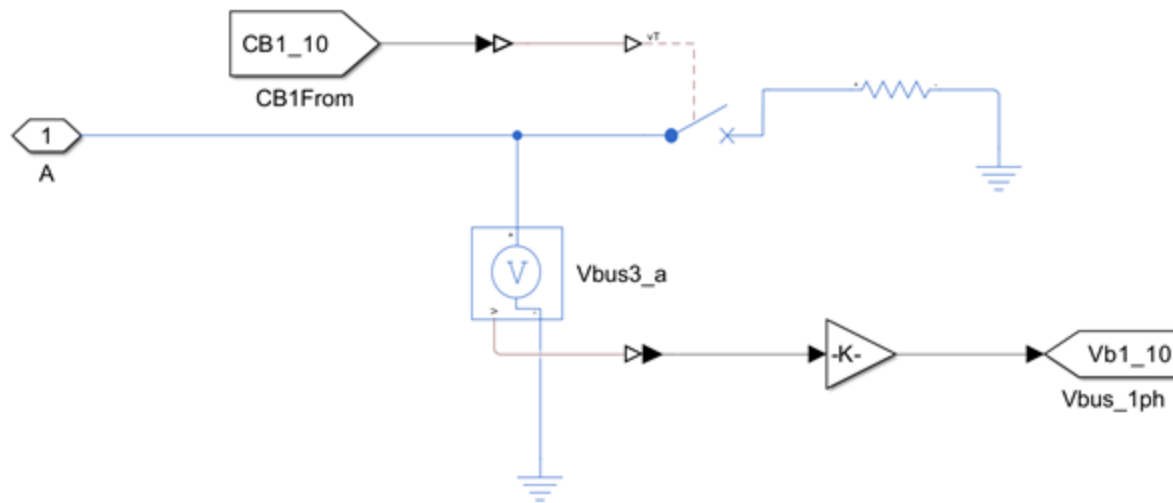
IEEE 123 Simscape model



Use the Simulink API to programmatically build the IEEE 123 Node Test Feeder

Fault Insertion

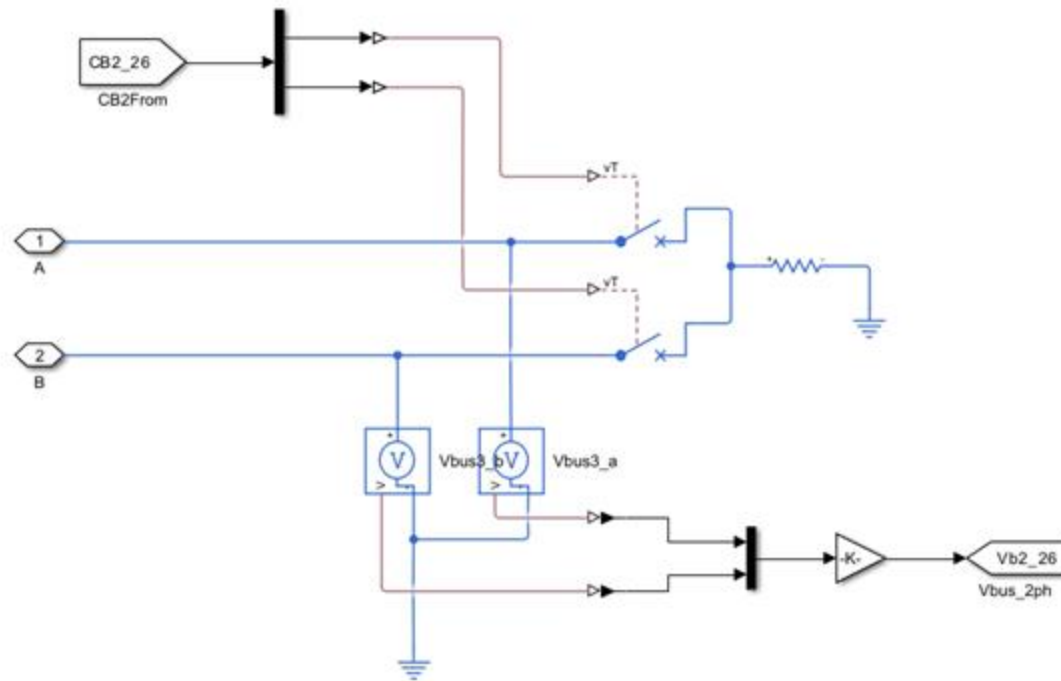
- Each bus is configured such that a fault can be applied on each phase.



Single-Phase Bus
1 fault combination,
a-g

Fault Insertion

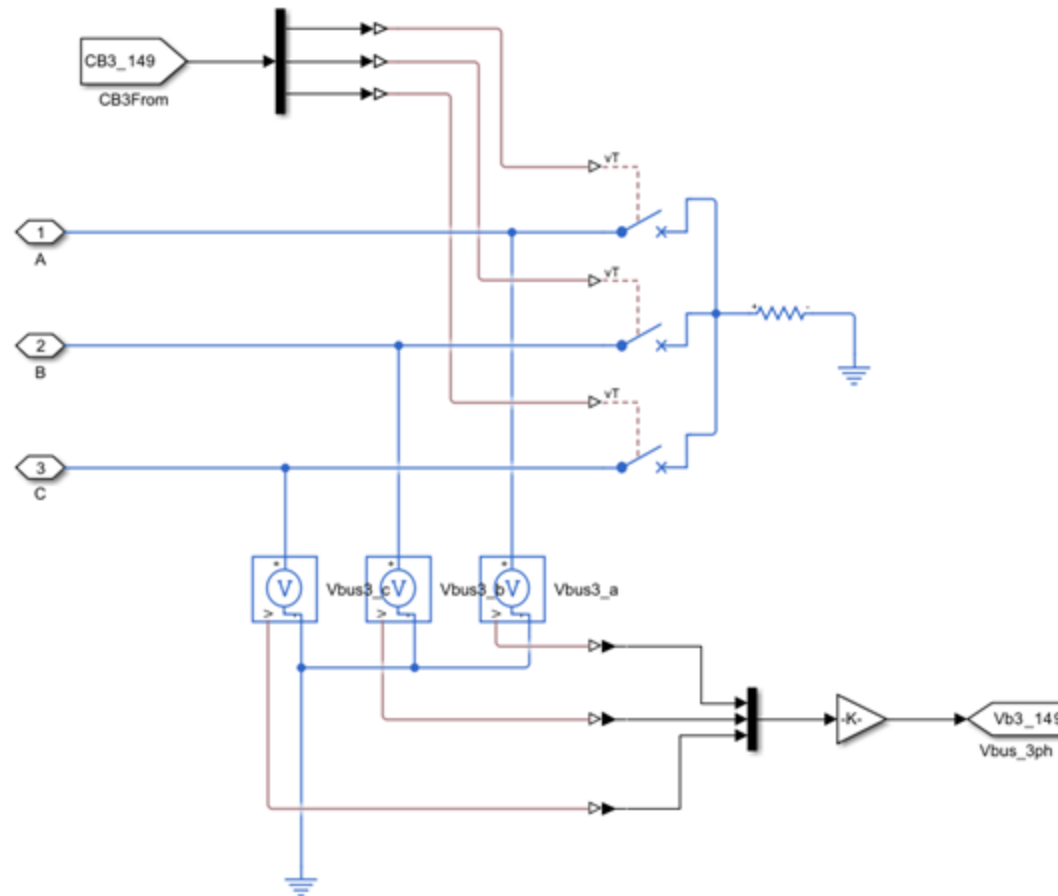
- Each bus is configured such that a fault can be applied on each phase.



Two-Phase Bus
3 fault combinations,
a-g
b-g
a-b-g

Fault Insertion

- Each bus is configured such that a fault can be applied on each phase.



Three-Phase Bus

7 fault combinations,

a-g

b-g

c-g

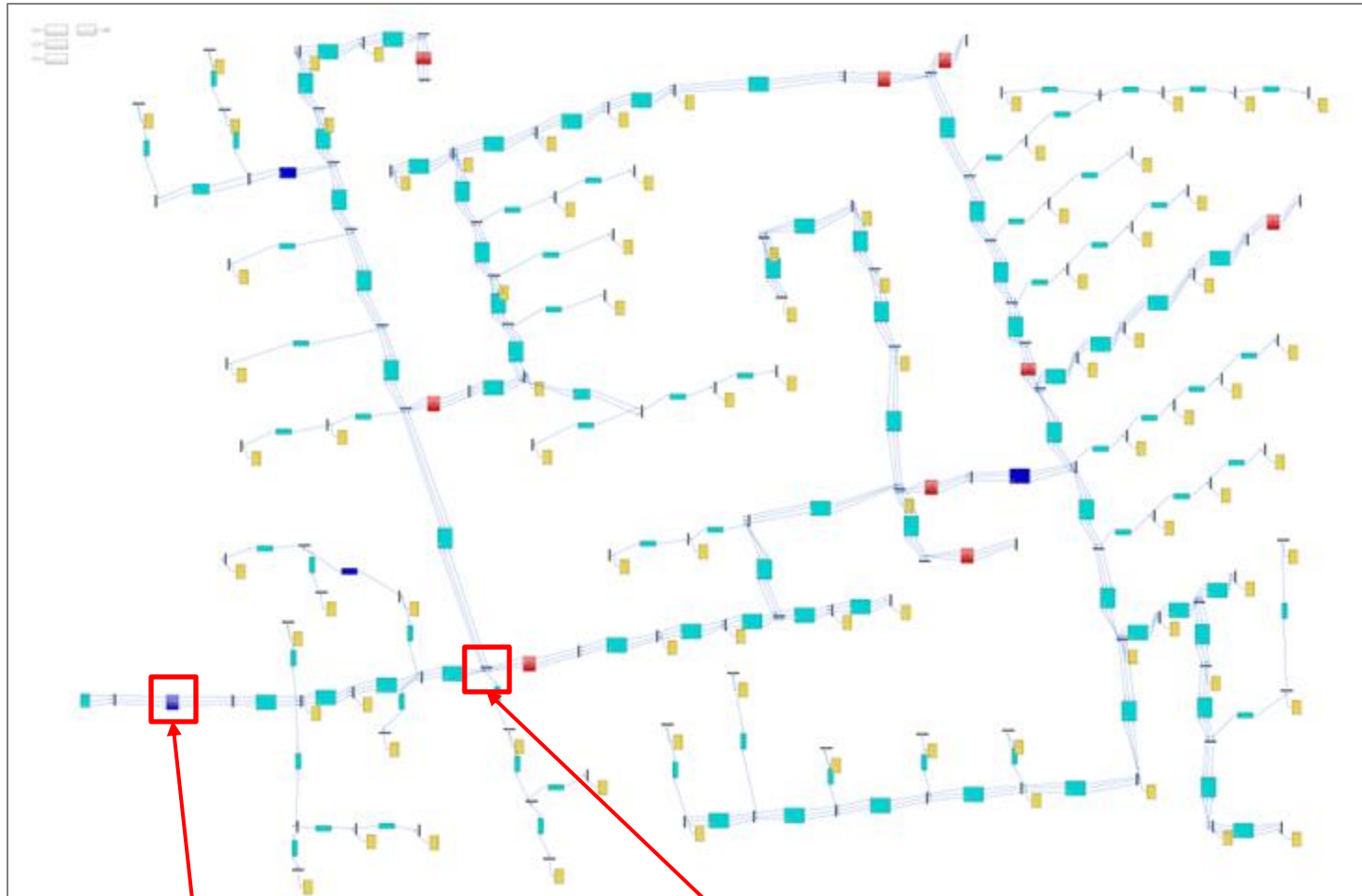
a-b-g

a-c-g

b-c-g

a-b-c-g

Fault Insertion



Legend

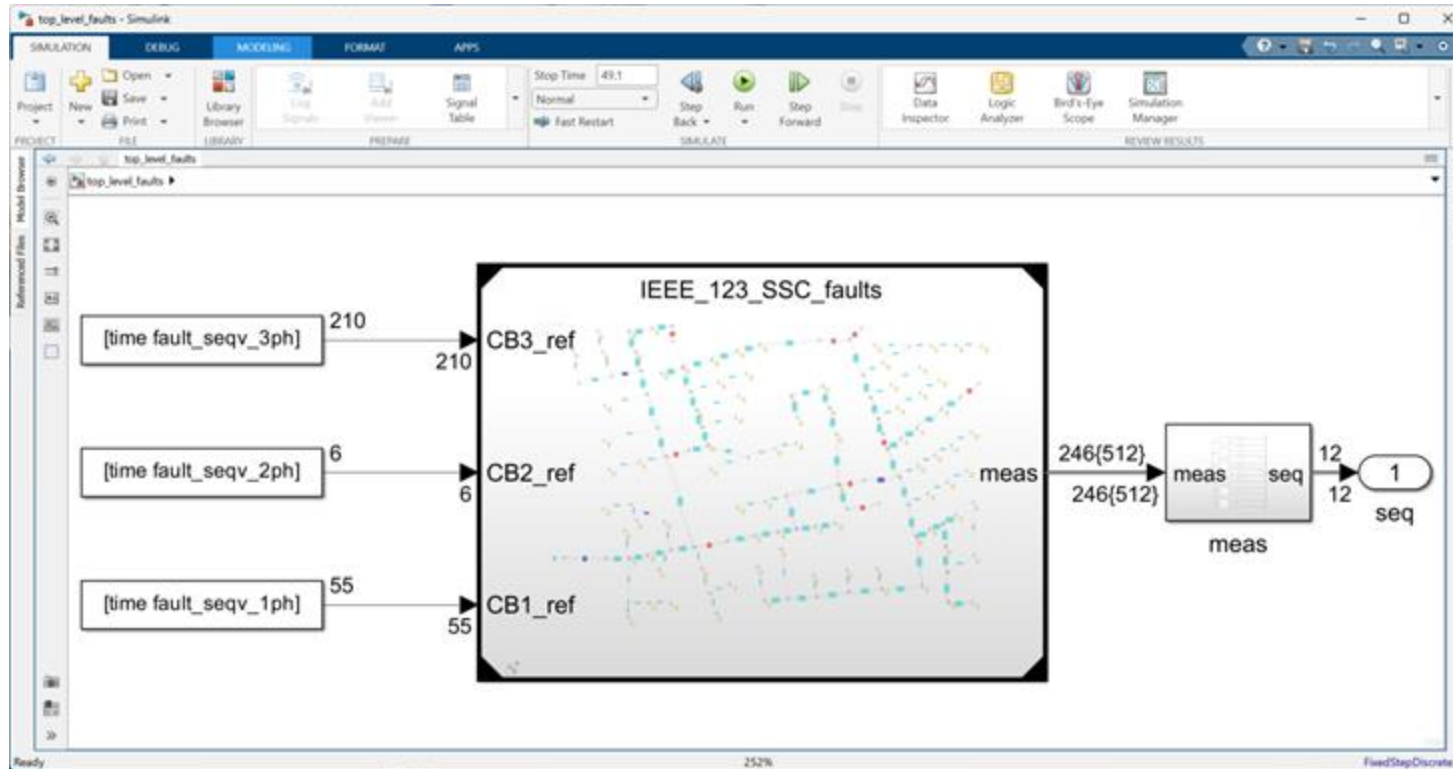
- Cables
- Circuit Breakers
- Loads
- Transformers

70 3-phase buses = 490 faults
3 2-phase buses = 9 faults
55 1-phase buses = 55 faults
Total faults = 554 faults

current measurement

voltage measurement

Fault Insertion



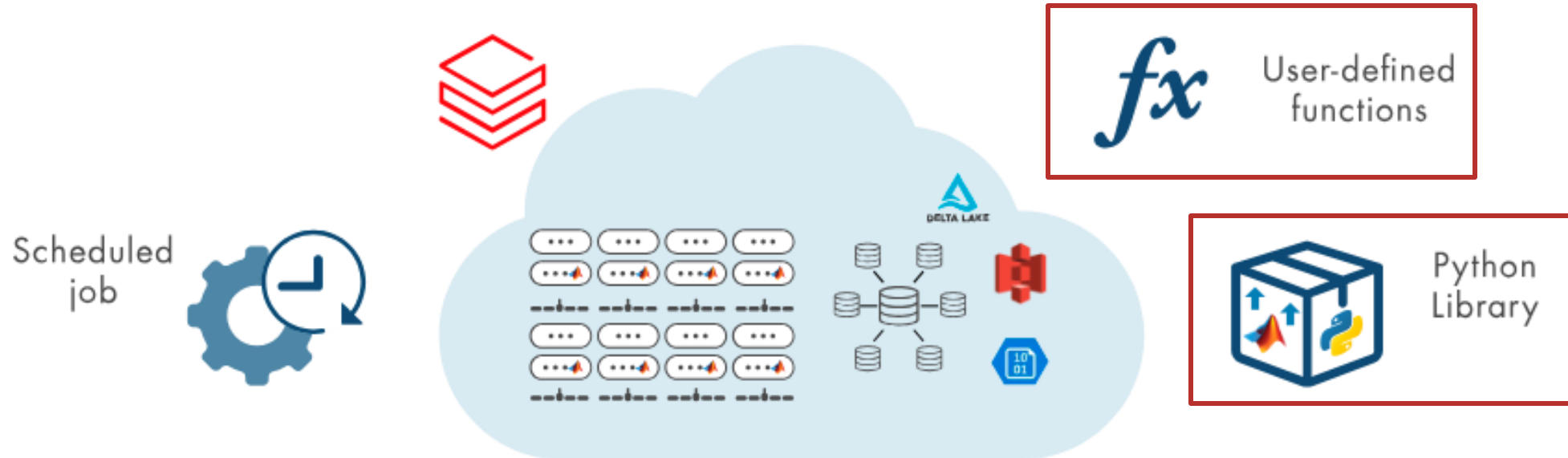
Each fault is applied and cleared sequentially in a single simulation run, through appropriate circuit breaker control signals

Bridge the Gap:

Sharing Your Models with Databricks Innovators

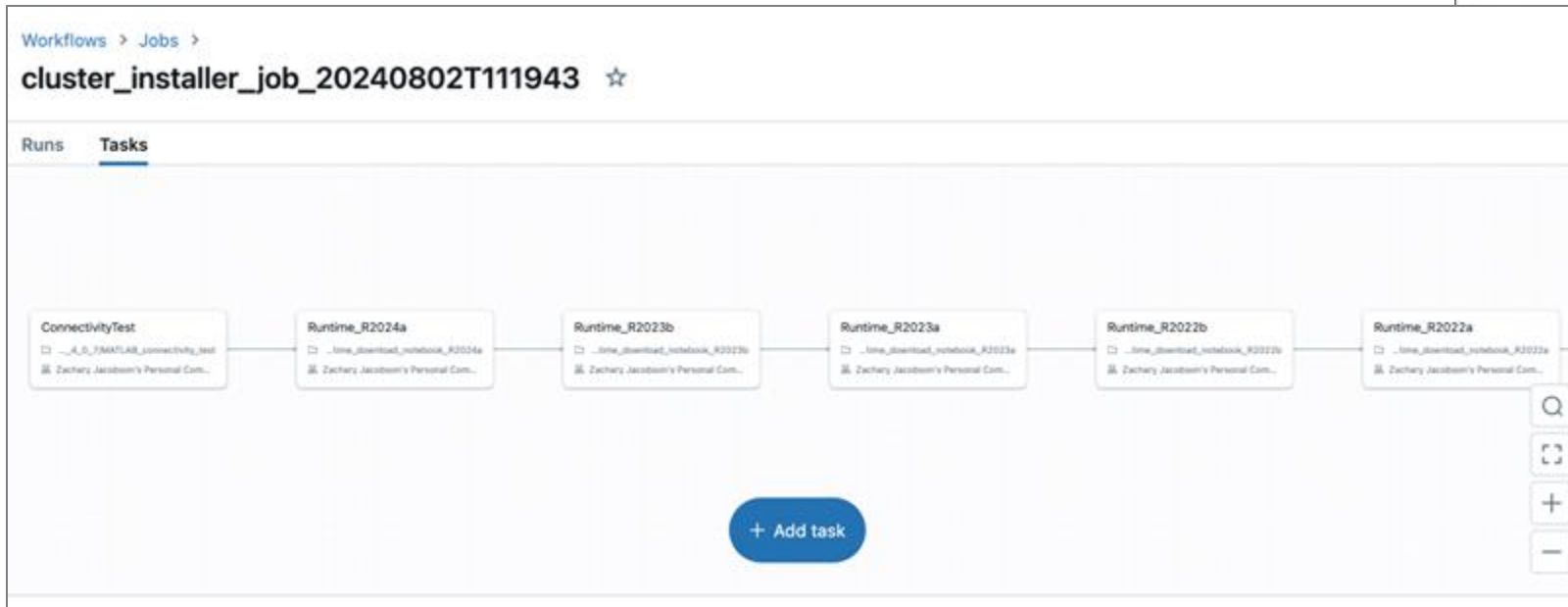
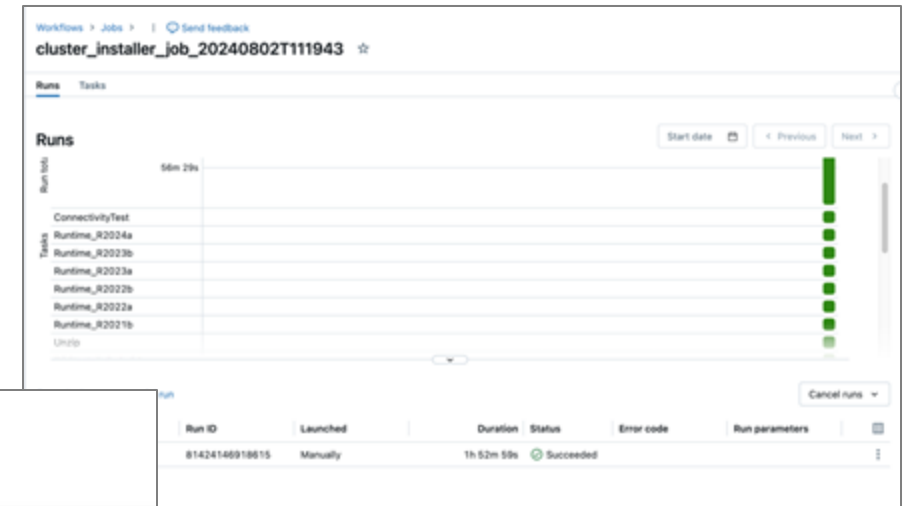
Once models are deployed to Databricks, share them with other users and incorporate them into multi step workflows. Deploy and execute models natively as jobs, at scale, or on MATLAB Runtime-enabled clusters.

Deploy models as a library or a user-defined function for use in any language, including SQL scripts. Java or Python libraries and notebooks can also use this code.



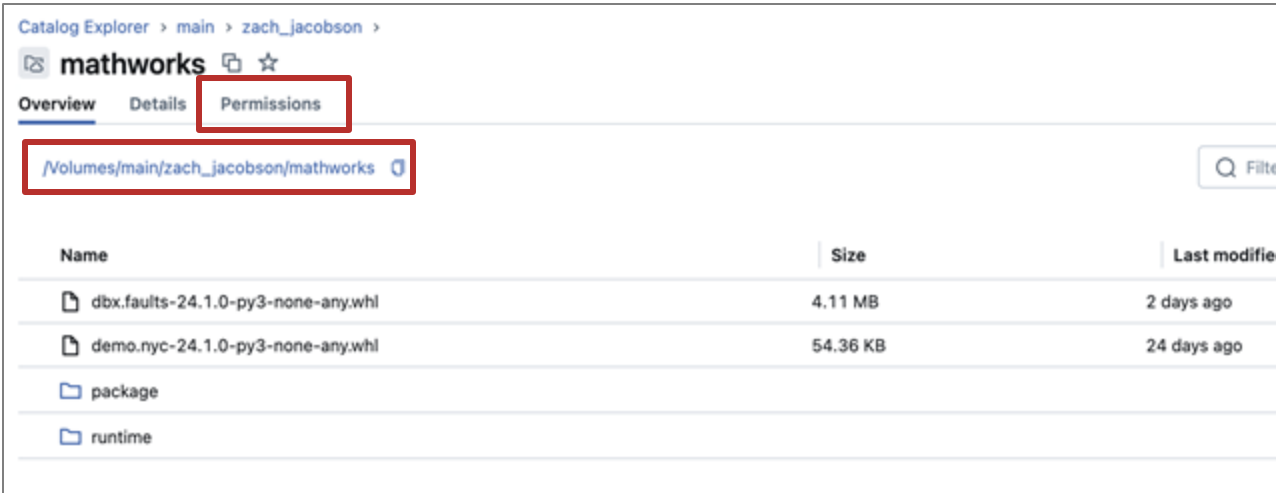
Simulink Runtime Installation on Databricks

- Deploying once on Databricks workflow
- Access to dependencies in Workspace folder
- No need to rebuild again (only build once)



Accessing Runtime from Workspace and UC Volumes

- Runtime is governed under Unity Catalog
- Permissions can be managed under Unity Catalog
- All unstructured data and files can be managed in Volumes



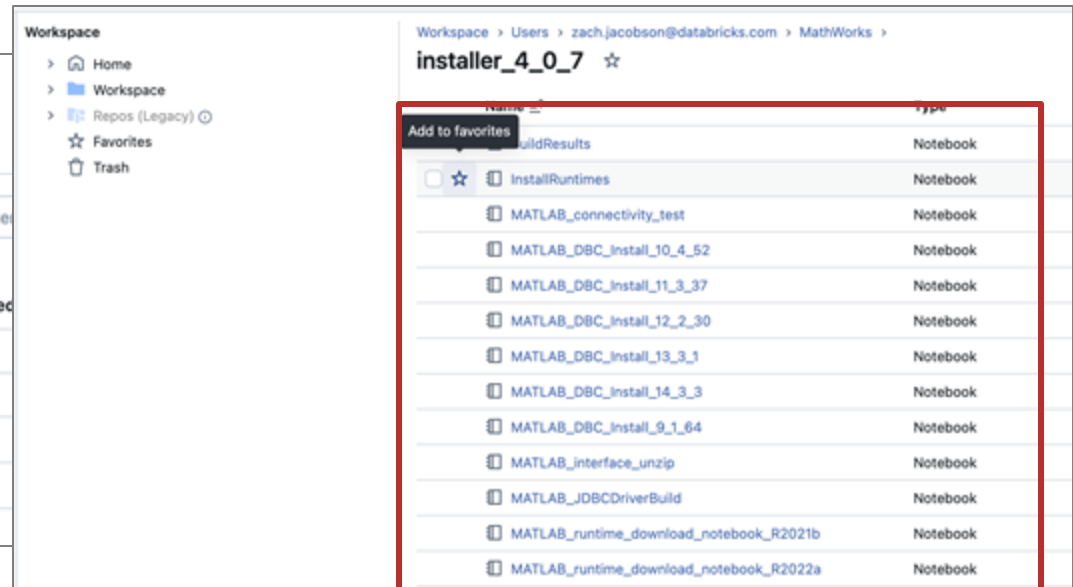
Catalog Explorer > main > zach_jacobson >

mathworks

Overview Details **Permissions**

[/Volumes/main/zach_jacobson/mathworks](#)

Name	Size	Last modified
dbx.faults-24.1.0-py3-none-any.whl	4.11 MB	2 days ago
demo.nyc-24.1.0-py3-none-any.whl	54.36 KB	24 days ago
package		
runtime		



Workspace

Workspace > Users > zach.jacobson@databricks.com > MathWorks > **installer_4_0_7**

Add to favorites

Name	Type
BuildResults	Notebook
InstallRuntimes	Notebook
MATLAB_connectivity_test	Notebook
MATLAB_DBC_install_10_4_52	Notebook
MATLAB_DBC_install_11_3_37	Notebook
MATLAB_DBC_install_12_2_30	Notebook
MATLAB_DBC_install_13_3_1	Notebook
MATLAB_DBC_install_14_3_3	Notebook
MATLAB_DBC_install_9_1_64	Notebook
MATLAB_interface_unzip	Notebook
MATLAB_JDBCDBDriverBuild	Notebook
MATLAB_runtime_download_notebook_R2021b	Notebook
MATLAB_runtime_download_notebook_R2022a	Notebook

Cluster Deployment

- Cluster configured once
- 14.3 Runtime Version
- Scale cluster to appropriate size
- Install custom libraries/wheel files

Compute > Simple Form WIP: OFF

Zachary Jacobson's Matlab Sim3

Configuration | Notebooks (1) | Libraries | Event log | Spark UI | Driver logs | Metrics | Apps | Spark compute UI - Master

Policy: Auto AZ

Multi node (selected) | Single node

Access mode: Single user access

Single user | Zachary Jacobson

Performance

Databricks Runtime Version: 14.3 LTS (includes Apache Spark 3.5.0, Scala 2.12)

Use Photon Acceleration

Worker type: i3.xlarge | 122 GB Memory, 16 Cores | Min workers: 4 | Max workers: 4 | Current: 3

Driver type: i3.xlarge | 122 GB Memory, 16 Cores

Enable autoscaling | Enable autoscaling local storage | Terminate after 120 minutes of inactivity

Compute > Simple Form WIP: OFF

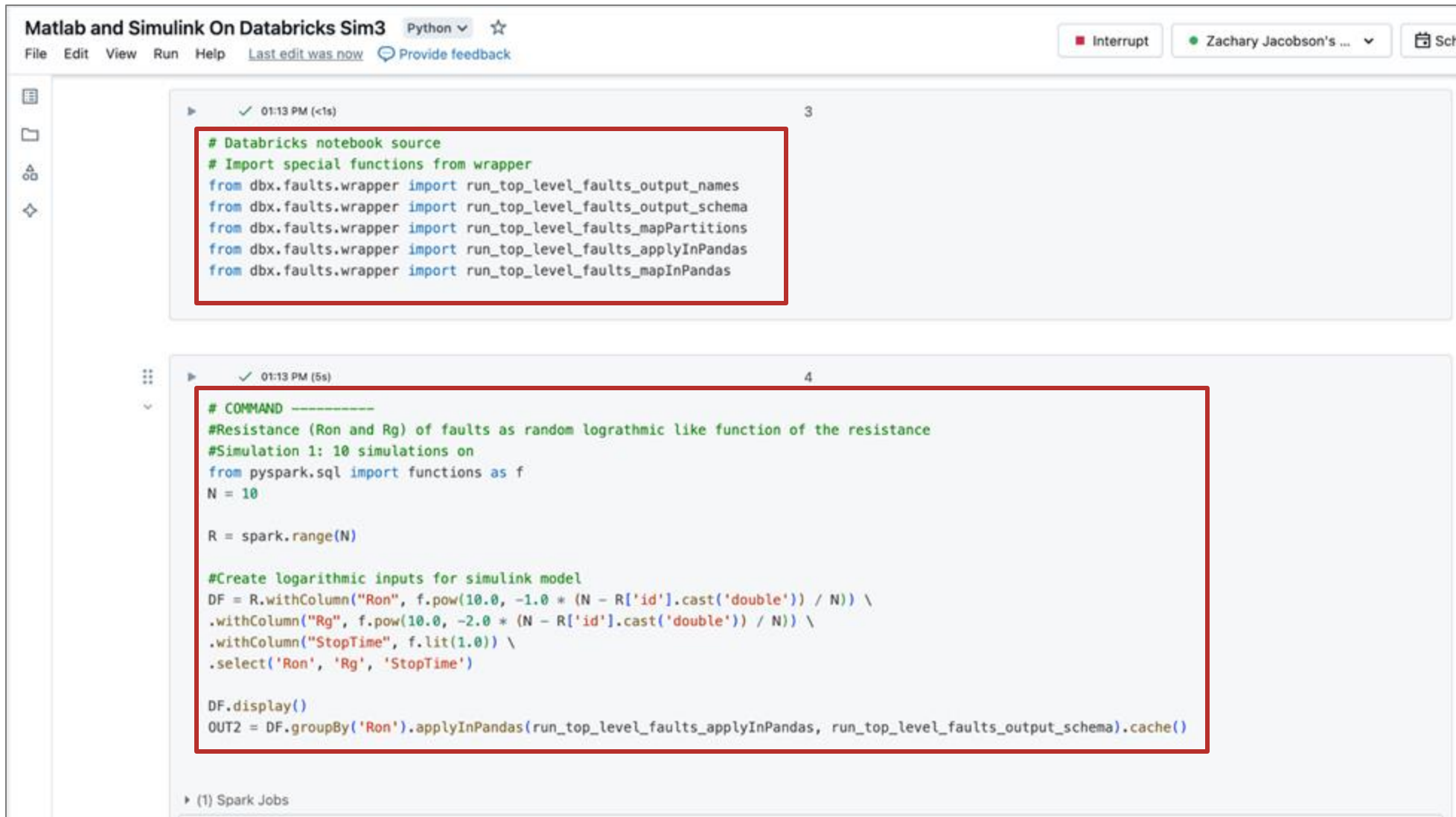
Zachary Jacobson's Matlab Sim3

Configuration | Notebooks (1) | Libraries | Event log | Spark UI | Driver logs | Metrics | Apps | Spark compute UI - Master

Filter libraries

Status	Name	Type
<input checked="" type="checkbox"/>	/Volumes/main/zach_jacobson/mathworks/dbx.faults-24.1.0-py3-none-any.whl	Wheel

Import MATLAB Packages/Model and Use Pandas UDFs



```
Matlab and Simulink On Databricks Sim3 Python ☆
File Edit View Run Help Last edit was now Provide feedback
Interrupt Zachary Jacobson's ... Sch

01:13 PM (<1s) 3
# Databricks notebook source
# Import special functions from wrapper
from dbx.faults.wrapper import run_top_level_faults_output_names
from dbx.faults.wrapper import run_top_level_faults_output_schema
from dbx.faults.wrapper import run_top_level_faults_mapPartitions
from dbx.faults.wrapper import run_top_level_faults_applyInPandas
from dbx.faults.wrapper import run_top_level_faults_mapInPandas

01:13 PM (5s) 4
# COMMAND -----
#Resistance (Ron and Rg) of faults as random logarithmic like function of the resistance
#Simulation 1: 10 simulations on
from pyspark.sql import functions as f
N = 10

R = spark.range(N)

#Create logarithmic inputs for simulink model
DF = R.withColumn("Ron", f.pow(10.0, -1.0 * (N - R['id'].cast('double')) / N)) \
.withColumn("Rg", f.pow(10.0, -2.0 * (N - R['id'].cast('double')) / N)) \
.withColumn("StopTime", f.lit(1.0)) \
.select('Ron', 'Rg', 'StopTime')

DF.display()
OUT2 = DF.groupBy('Ron').applyInPandas(run_top_level_faults_applyInPandas, run_top_level_faults_output_schema).cache()

(1) Spark Jobs
```


Govern Output of MATLAB Model in Unity Catalog

Column ↕	Type	Comment
V1mag_13	double	
V1ph_13	double	
V2mag_13	double	
V2ph_13	double	
V0mag_13	double	
V0ph_13	double	
I1mag_1	double	
I1ph_1	double	
I2mag_1	double	
I2ph_1	double	
I0mag_1	double	
I0ph_1	double	
fault_class	string	

AI generate

About this table

Owner: Zachary Jacobson

Data source: [Delta](#)

Popularity:

Last updated: last week

Size: 2.2MiB, 36 files

Quality

[Add monitor](#)

Tags

[Add tags](#)

Row filter

[Add filter](#)

AI Suggested Description



The 'fault_simulation' table contains data from fault simulations, including various voltage and current parameters. It includes details such as the magnitude and phase of the voltages and currents, as well as the fault class. This data can be used to analyze and compare the performance of different fault classes, and to identify the impact of various voltage and current parameters on fault behavior. This information can be useful for improving the design and performance of electrical systems.

[✓ Accept](#) [✎ Edit](#) [Feedback](#)

Output continued...

1.2 I0ph_1	A ^B _C fault_class
-43.76099147279311	unfaulted
-58.96895313025194	CB3_1_a_g
-28.701253657066538	CB3_1_b_g
-46.339389547752276	CB3_1_c_g
-44.62491344478943	CB3_1_a_b_g
-55.222329953611066	CB3_1_a_c_g
-37.112247990877904	CB3_1_b_c_g

Catalog Explorer > main > zach_jacobson >

fault_locations  

Overview Sample Data Details Permissions History Lineage Insights Quality

Data Preview

	i ₃ Bus_No	1.2 X_Coord	1.2 Y_Coord	i ₃ No_Phases	i ₃ A	i ₃ B	i ₃ C
1	1	1123.6194	3537.993	3	1	1	1
2	2	1071.0528	3176.553	1	0	1	0
3	3	1109.283	4107.5346	1	0	0	1
4	4	1114.062	4299.207	1	0	0	1
5	5	1376.8938	4118.487	1	0	0	1
6	6	1658.841	4123.9638	1	0	0	1
7	7	1362.5574	3499.6584	3	1	1	1
8	8	1668.3984	3428.4654	3	1	1	1
9	9	1568.0442	3067.0254	1	1	0	0
10	10	1219.1946	3045.12	1	1	0	0
11	11	784.3272	2897.2584	1	1	0	0
12	12	1505.9202	3680.3784	1	0	1	0
13	13	1964.6814	3384.6546	3	1	1	1
14	14	1147.5132	2831.5416	1	1	0	0

Blog: Transforming Electrical Fault Detection: The Power of Databricks & MATLAB

The image shows a screenshot of a Medium blog post. The title is "Transforming Electrical Fault Detection: The Power of Databricks and MATLAB" by Zach Jacobson. The post includes a diagram illustrating the integration approach, which is summarized as "The entire company on one data and compute platform in the cloud".

Integration Approach "The entire company on one data and compute platform in the cloud"

The diagram shows a flow from "DATA ENGINEERS" and "DATA SCIENTISTS" on the left, through "IT" in the center, to "R&D ENGINEERS" and "R&D SCIENTISTS" on the right. Below this flow are three stacked layers:

- MATLAB / SIMULINK**: Data analysis, algorithm development, modeling and simulation.
- databricks UNIFIED DATA SERVICE**: High quality data with great performance.
- aws ENTERPRISE CLOUD SERVICE**: A simple, scalable, and secure managed service.

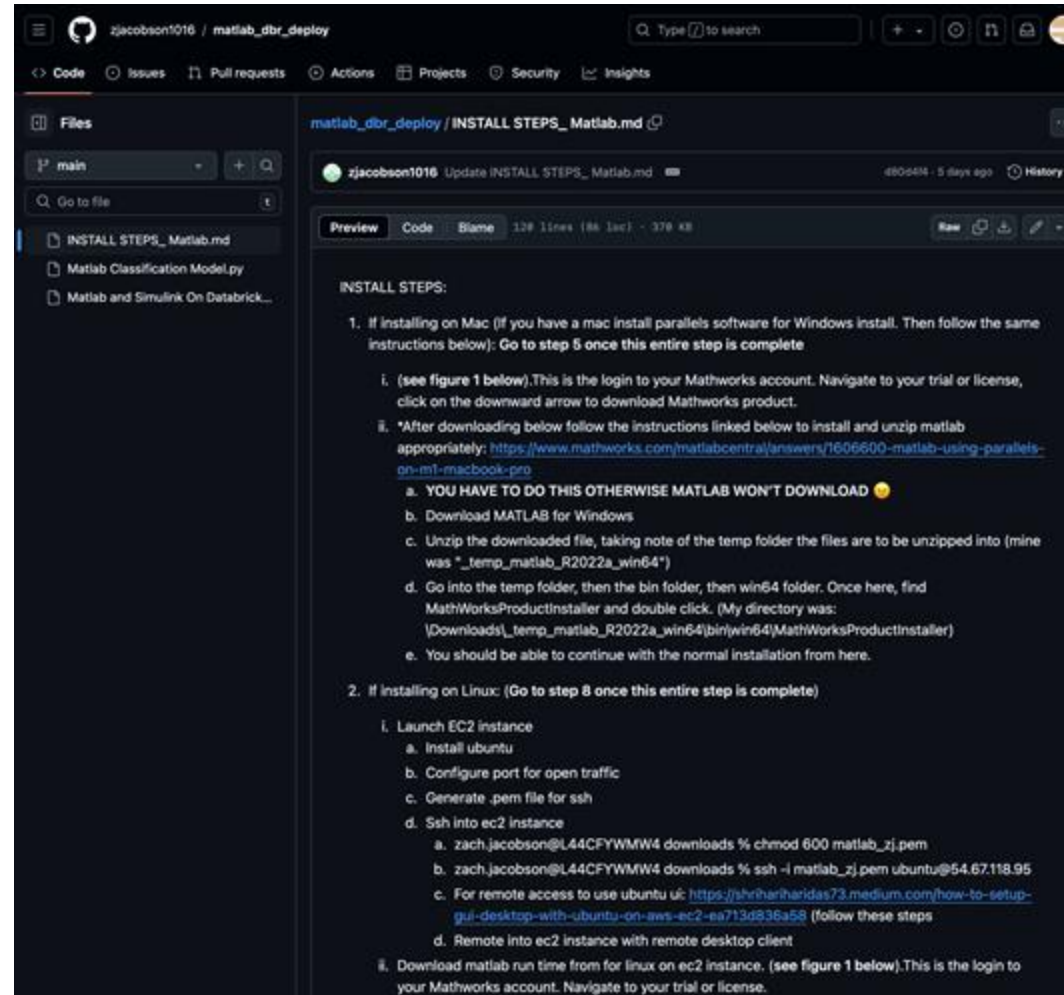
At the bottom of the diagram is a wavy line representing "RAW DATA LAKE" with binary code (0s and 1s) underneath it.

Introduction

If you are in the Tech industry, chances are you have either heard of Databricks, MATLAB, or both perhaps. If you haven't, give it a quick google search and you will find out very quickly.

<https://medium.com/@jacobsonzach23/transforming-electrical-fault-detection-the-power-of-databricks-and-matlab-0aff9f2ab7bf>

GitHub: Instructions & MATLAB Code



The screenshot shows a GitHub repository for 'zjacobson1016 / matlab_dbr_deploy'. The file 'INSTALL STEPS_Matlab.md' is open, displaying the following content:

```
INSTALL STEPS:
```

1. If installing on Mac (If you have a mac install parallels software for Windows install. Then follow the same instructions below): Go to step 5 once this entire step is complete
 - i. (see figure 1 below).This is the login to your Mathworks account. Navigate to your trial or license, click on the downward arrow to download Mathworks product.
 - ii. *After downloading below follow the instructions linked below to install and unzip matlab appropriately: <https://www.mathworks.com/matlabcentral/answers/1606600-matlab-using-parallels-on-m1-macbook-pro>
 - a. **YOU HAVE TO DO THIS OTHERWISE MATLAB WON'T DOWNLOAD** 🟡
 - b. Download MATLAB for Windows
 - c. Unzip the downloaded file, taking note of the temp folder the files are to be unzipped into (mine was "_temp_matlab_R2022a_win64")
 - d. Go into the temp folder, then the bin folder, then win64 folder. Once here, find MathWorksProductInstaller and double click. (My directory was: |Downloads|_temp_matlab_R2022a_win64|bin|win64|MathWorksProductInstaller)
 - e. You should be able to continue with the normal installation from here.
2. If installing on Linux: (Go to step 8 once this entire step is complete)
 - i. Launch EC2 instance
 - a. Install ubuntu
 - b. Configure port for open traffic
 - c. Generate .pem file for ssh
 - d. Ssh into ec2 instance
 - a. zach.jacobson@L44CFYWMW4 downloads % chmod 600 matlab_zj.pem
 - b. zach.jacobson@L44CFYWMW4 downloads % ssh -i matlab_zj.pem ubuntu@54.67.118.95
 - c. For remote access to use ubuntu uc: <https://shriharitharidas73.medium.com/how-to-setup-gui-desktop-with-ubuntu-on-aws-ec2-ea713d836a58> (follow these steps
 - d. Remote into ec2 instance with remote desktop client
 - ii. Download matlab run time from for linux on ec2 instance. (see figure 1 below).This is the login to your Mathworks account. Navigate to your trial or license.

https://github.com/zjacobson1016/matlab_dbr_deploy/tree/main

The IEEE 123 Node Test Feeder

Kersting, W.H., "Radial distribution test feeders," *Power Engineering Society Winter Meeting, 2001. IEEE* , vol.2, no., pp.908,912 vol.2, 2001
doi: 10.1109/PESW.2001.916993

URL:

<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=916993&isnumber=19809>

Files describing the network are available at the following URL:

<https://cmte.ieee.org/pes-testfeeders/>

MATLAB EXPO

Thank you!



© 2024 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [mathworks.com/trademarks](https://www.mathworks.com/trademarks) for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

