

From **Oops** to **Ops**

Incident Response with Notebooks

T



Shafiq Rahman
Engineering Manager, Microsoft
Shafiq.Rahman@microsoft.com

Julie Koesmarno (@MsSQLGirl)
Program Manager, Microsoft
jukoesma@microsoft.com



Our learning journey today

1

Intro to TSG & Notebooks

2

Executable TSGs with Notebooks

3

Jumpstart your TSG notebooks

4

Automatable TSGs with Notebooks

5

Takeaways and Other Resources

<https://bit.ly/3ow3wmM>



1

Intro to TSGs & Notebooks



"TSG"

TSG = Troubleshooting Guide

SOP = Standard Operating Procedure

Playbook = steps to identify issues

Runbook = procedures to achieve specific outcome

KB = Knowledge Base

Incident Response



Before you run ~
(sharing lessons learned)



Don't forget ...

change agent process

1. Know your goals

2. Understand the gaps

3. Plan it!

Demo

Notebooks in Azure Data Studio
created by community



Notebooks

Text cells

Code cells

- SQL
- PowerShell
- Python
- **KQL**
- *.NET interactive*

Jupyter Book

The screenshot displays the Azure Data Studio interface. The top menu bar includes File, Edit, View, and Help. The title bar shows the active notebook: WWIReproducibleResearch Vol 1.ipynb - jubilant-data-wizards - Azure Data Studio. The left sidebar contains icons for Explorer, Search, Source Control, and a Jupyter icon. The main editor area shows a notebook with a 'Conclusion' section. The text in the notebook describes marketing experimentation with two product groups based on stock groups sold in January 2016. Below the text is a SQL query cell labeled [31] that uses a UNION ALL to select data from two product groups. The query has been executed successfully, as indicated by the status message 'Commands completed successfully.' and the 'Total execution time: 00:00:00.010'. The results are displayed in a table with two columns: StockGroup and Margin%.

File Edit View Help • WWIReproducibleResearch Vol 1.ipynb - jubilant-data-wizards - Azure Data Studio

WWIReproducibleResearch Vol 1.ipynb

jubilant-data-wizards > Simple Demo > Sample Notebooks - Data Analysis > WWIReproducibleResearch Vol 1.ipynb

+ Cell ▾ ▶ Run all Kernel SQL Attach to Select Connection ▾

Conclusion

Out of the 9 stock groups sold in January 2016, these are the proposed two groups that we can do marketing experimentation with:

1. **Product Group 1** - Furry Footwear, USB Novelties and Mugs with percentage of Total Revenue between 2.53% and 1.77%.
2. **Product Group 2** - Clothing, Novelty Items and Toys with percentage of Total Revenue between 8.81% and 1.84%

This paper includes their January baseline of the Quantity Sold, the Profit and the Revenue. The next analysis should include these metrics including the % month over month growth for the treatment group and the control group of each Product Group.

```
[31] 1 SELECT
      2     'PG1:' + StockGroupName AS StockGroup
      3     , [Margin%]
      4 FROM #ProductGroup1
      5 UNION ALL
      6 SELECT
      7     'PG2:' + StockGroupName AS StockGroup
      8     , [Margin%]
      9 FROM #ProductGroup2
```

Commands completed successfully.

Total execution time: 00:00:00.010

	StockGroup	Margin%
1	PG1:Furry Footwear	63.898944
2	PG1:Mugs	56.856187
3	PG2:USB Novelties	37.532478
4	PG2:Toys	36.009497

main 0 0 Enable Demo Mode Change SQL language provider



Modern data experiences, multi data-platforms

SQL: SQL Server, PostgreSQL, Azure SQL Edge, Azure SQL*

KQL: Azure Data Explorer

Environment: On-premises, poly-cloud

Client tool on multi-platforms: Windows, Linux, macOS

Extensible tool

Git support

Jupyter Notebook experiences

... and more ...



2

Executable TSGs (From **Oops** to **Ops**)



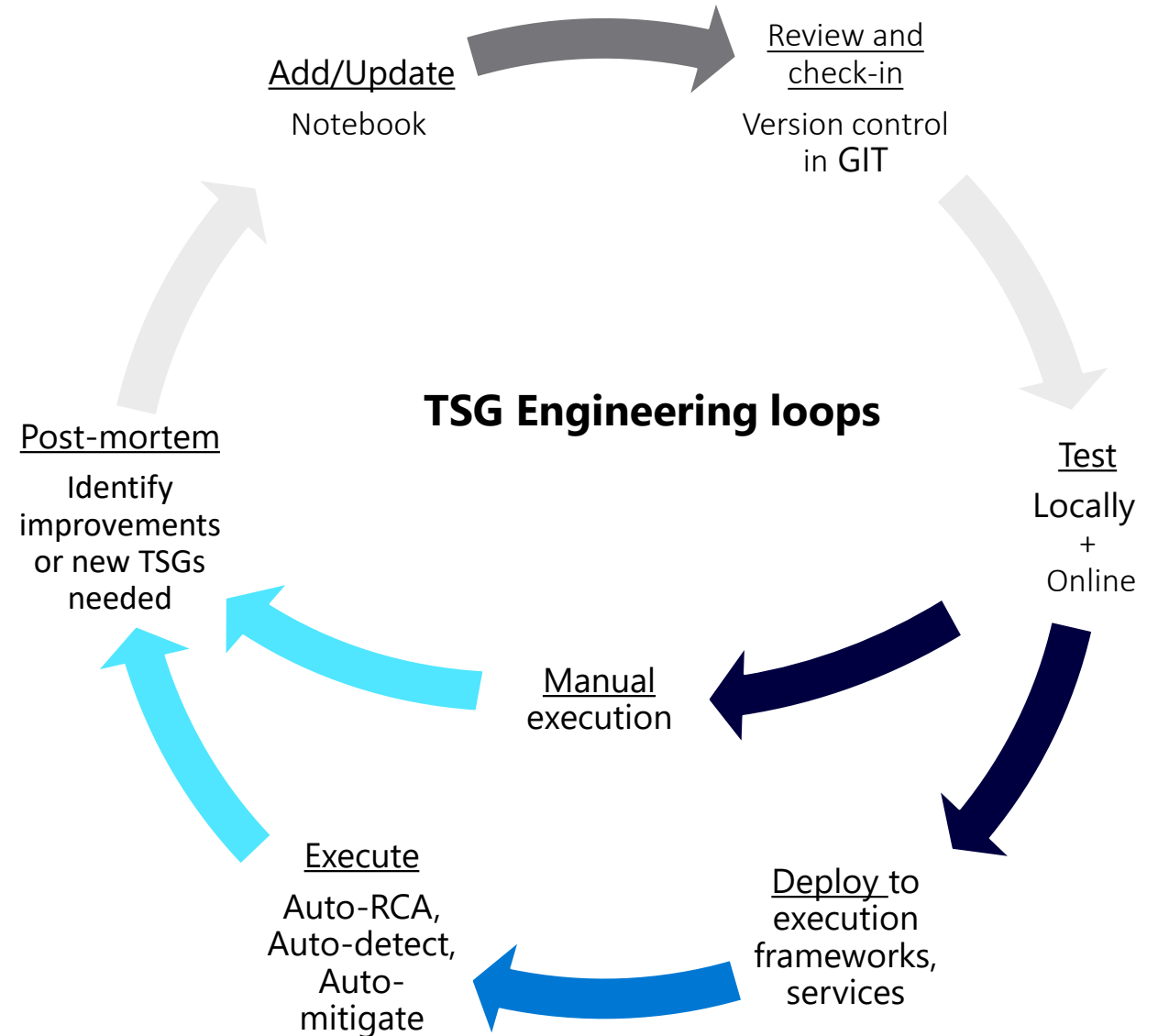
**Troubleshooting
Guides in the closet!**

Challenges:

1. Static text in OneNote / wiki resulting in copy and paste
2. Discoverability - siloed TSG scripts
3. No version control to keep track quality
4. Sharing code and results with context is hard to do with OneNote or text files
5. Hard to crowd source code
6. Hard to search
7. Reusability

Notebooks as TSGs

- Bring engineering discipline to Livesite content
- Notebooks can be checked into GIT repos to provide version control
- Review, test, deploy, similar to code
- A simple system, with flexibility to create once and use everywhere
- Notebooks can be shared easily
- Notebooks inherently record every step executed with results/failures
- A repeatable environment for debugging and investigation



Demo

Parameterization



3

**Jumpstart your
TSG Notebooks**





Jump start to Notebooks!

Bulk conversions -

- SQL Scripts to Notebooks with [ConvertTo-SqlNoteBook](#)
- PowerShell scripts to Notebooks with [ConvertTo-PowerShellNotebook](#)

Demo: <https://youtu.be/80L-UTOlknw?t=1627>

Individual conversions -

- Export as Notebook UI in Azure Data Studio from a SQL Script
- [PowerShell history to a Notebook](#)

```
Get-History 7,14 | % comm* | Export-AsPowerShellNotebook  
-OutputNotebook d:\temp\testthis.ipynb
```


4

Automatable TSGs (Lesser **Oops**, More **Ops**)




time to mitigate

**incident response
efficiency?**

number of incidents?

time to respond

A glass bottle with a cork, containing a rolled-up piece of paper, floating in water. The bottle is tilted, and the paper is visible inside. The water is dark blue, and the background is a gradient of blue and grey.

**How does your team
measure incident response
today?**

DBA Happiness?

5 incident response maturity levels

Start here ...



Less 2am calls! Happier DBAs...

level
1

Adhoc scripts

level
2

Manual documents (OneNote, Word, etc)

level
3

Reusable scripts

level
4

Executable Notebooks

level
5

Automatable Notebooks

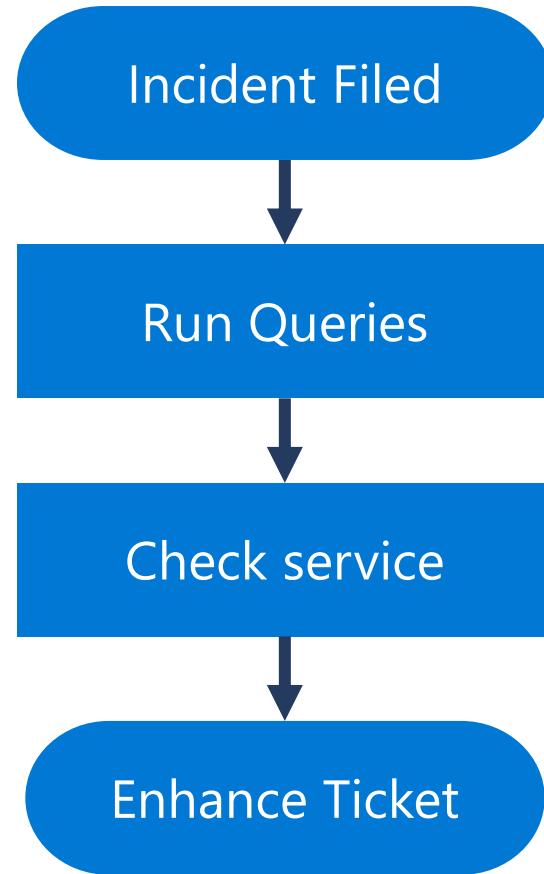
scalability
maintainability

Automation

What can we automate?

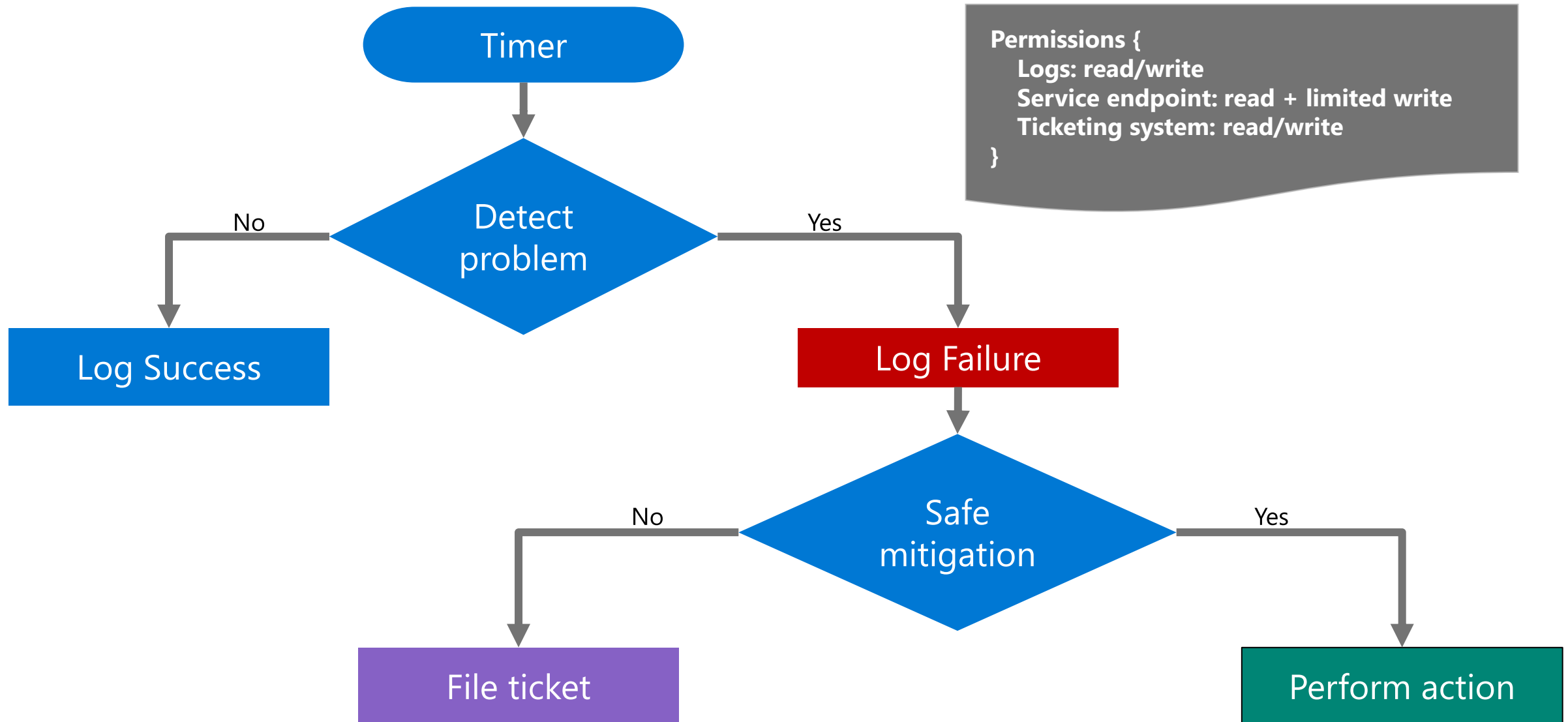


Auto-Diagnose/RCA



```
Permissions {  
  Logs: read-only  
  Service endpoint: read-only  
  Ticketing system: read/write  
}
```


Auto-Detect and auto-mitigate



Demo: Auto Diagnose / RCA

with Azure Logic Apps

(very similar to Power Automate)



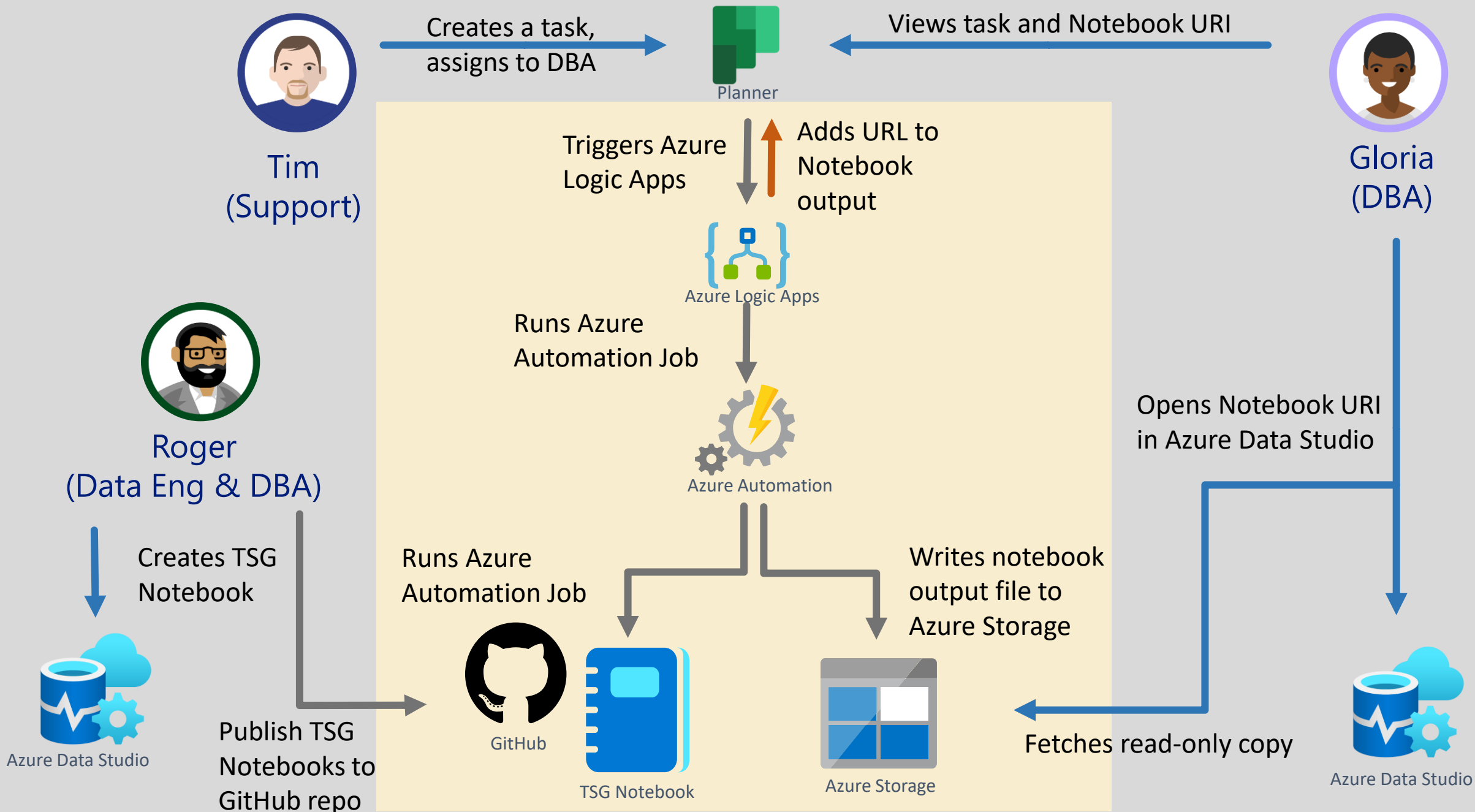
A simple demo to illustrate a Support Case System with automated TSG execution

The screenshot shows a support case system interface. At the top, a task titled "Slow Query [cont...]" is displayed, with a sub-header "Last changed moments ago by yo...". Below this, there is an "Assign" button and a search bar labeled "Type a name or email address". A list of unassigned users is shown, including Julie Koesmarno and Shafiq Rahman. A callout box points to Julie Koesmarno with the text: "Automatically execute pre-defined notebook when assigned to me!".

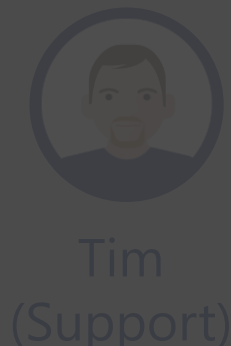
Below the user list, there is a "Notes" section with a text input field containing "My query runs slow on my database. Help ...". An orange arrow points from this text to a "Notes" panel on the right. This panel shows a note with a timestamp "Update 2021-05-01T21:12:47.4052845+00:00:" and a URL: https://jktsgnotebooks.blob.core.windows.net/blobjktsgnotebooks/DBDiagnostic_20210501211232.ipynb. A callout box points to this URL with the text: "Update the task with the pre-executed **notebooks** for me to view".

Below the notes, there is a "Checklist 0 / 1" section with a "Show on card" checkbox. The checklist items are:

- ☐ https://jktsgnotebooks.blob.core.windows.net/blobjktsgnotebooks/DBDiagnostic_20210501211232.ipynb
- ☐ Add an item



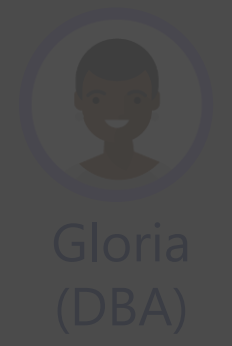
Demo: Automated task workflow with notebooks



Creates a task,
assigns to DBA



Views task and Notebook URI



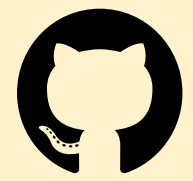
Roger
(Data Eng & DBA)

Creates TSG
Notebook



Azure Data Studio

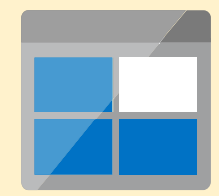
Publish TSG
Notebooks to
GitHub repo



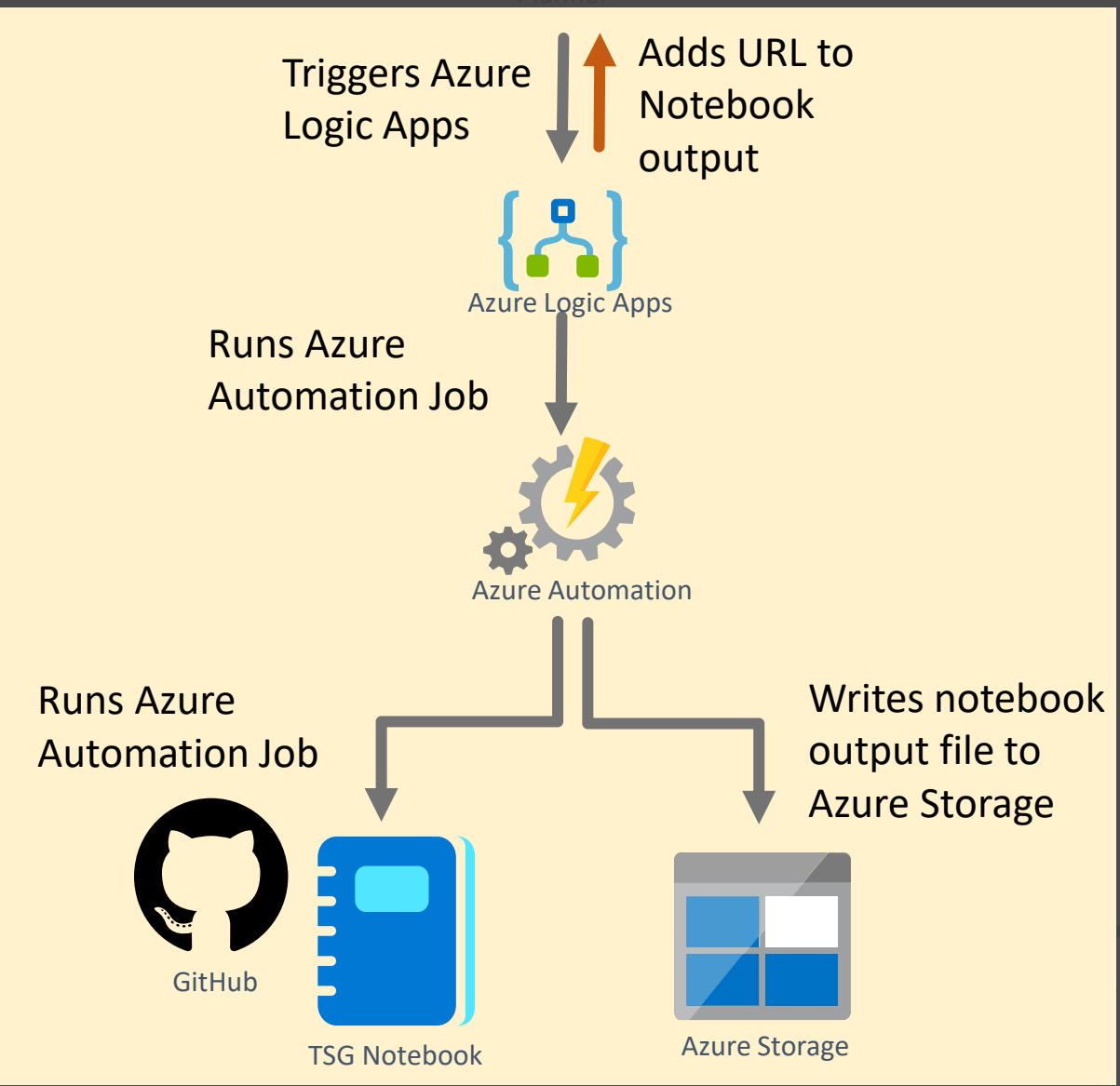
GitHub



TSG Notebook



Azure Storage



Opens Notebook URI
in Azure Data Studio

read-only copy



Azure Data Studio

Demo: Automated task workflow with notebooks

[Home](#) >

DemoTSGWorkflow

Logic app

Search (Ctrl+/)

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Development Tools

Logic app designer

Logic app code view

Versions

API connections

Quick start guides

Settings

Workflow settings

Authorization

[Run Trigger](#) [Refresh](#) [Edit](#) [Delete](#) [Disable](#) [Update Schema](#) [Clone](#) [Open in mobile](#) [Export](#) ...

Introducing the new portable Logic Apps runtime that supports local development and debugging. Click to learn more. →

Essentials

[View Cost](#) | [JSON View](#)Resource group ([change](#))

jukoema_exe_rg

Location

West US 2

Subscription ([change](#))

Subscription ID

Definition

1 trigger, 5 actions

Status

Enabled

Runs last 24 hours

5 successful, 2 failed

Integration Account

[Get started](#)[Runs history](#)[Trigger history](#)[Metrics](#)

All

Start time earlier than

Pick a date

Pick a time

Specify the run identifier to open monitor view directly

Status

Start time

Identifier

Duration

Static Results

✓ Succeeded

5/1/2021, 1:39 PM

08585817052871593324217...

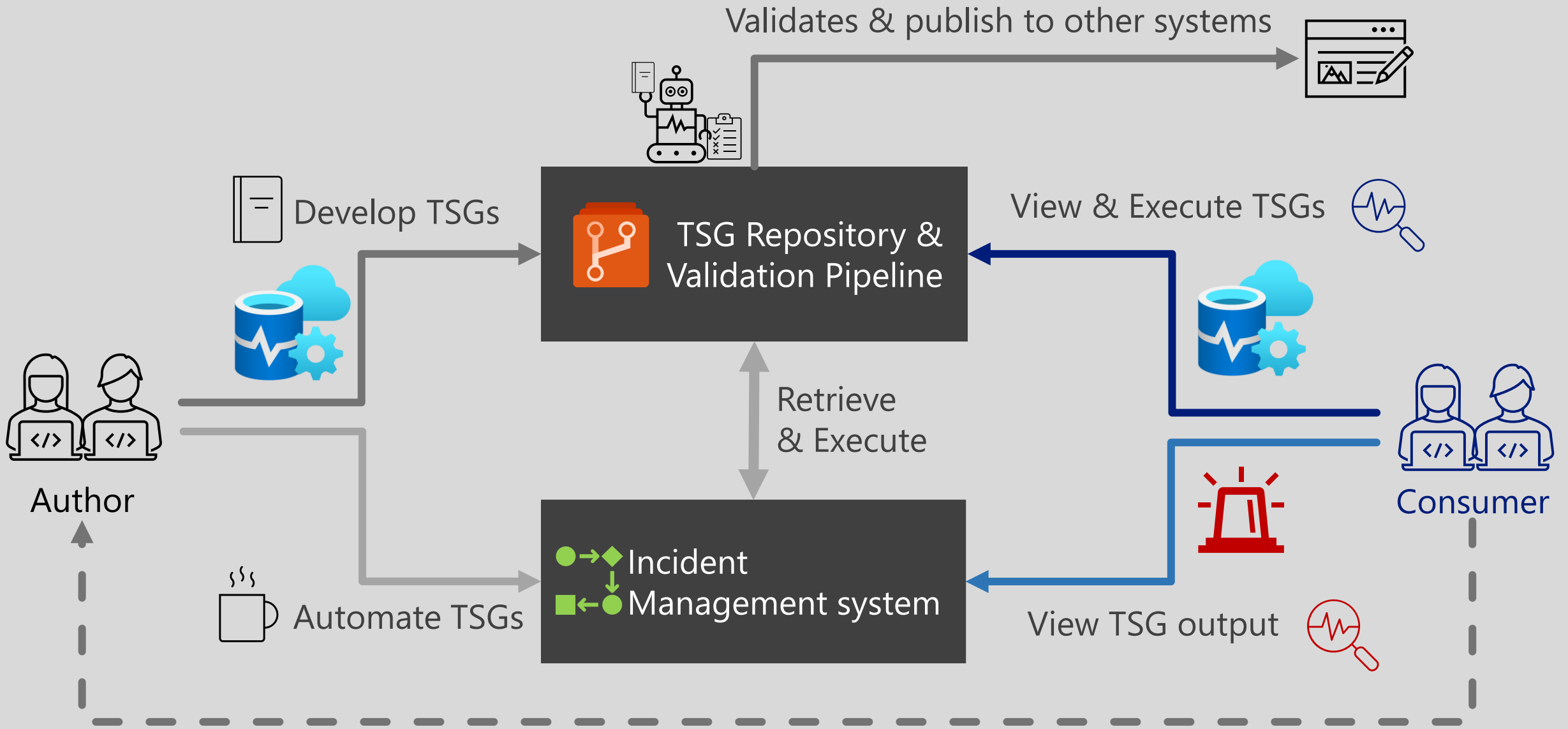
1.34 Minutes

✓ Succeeded

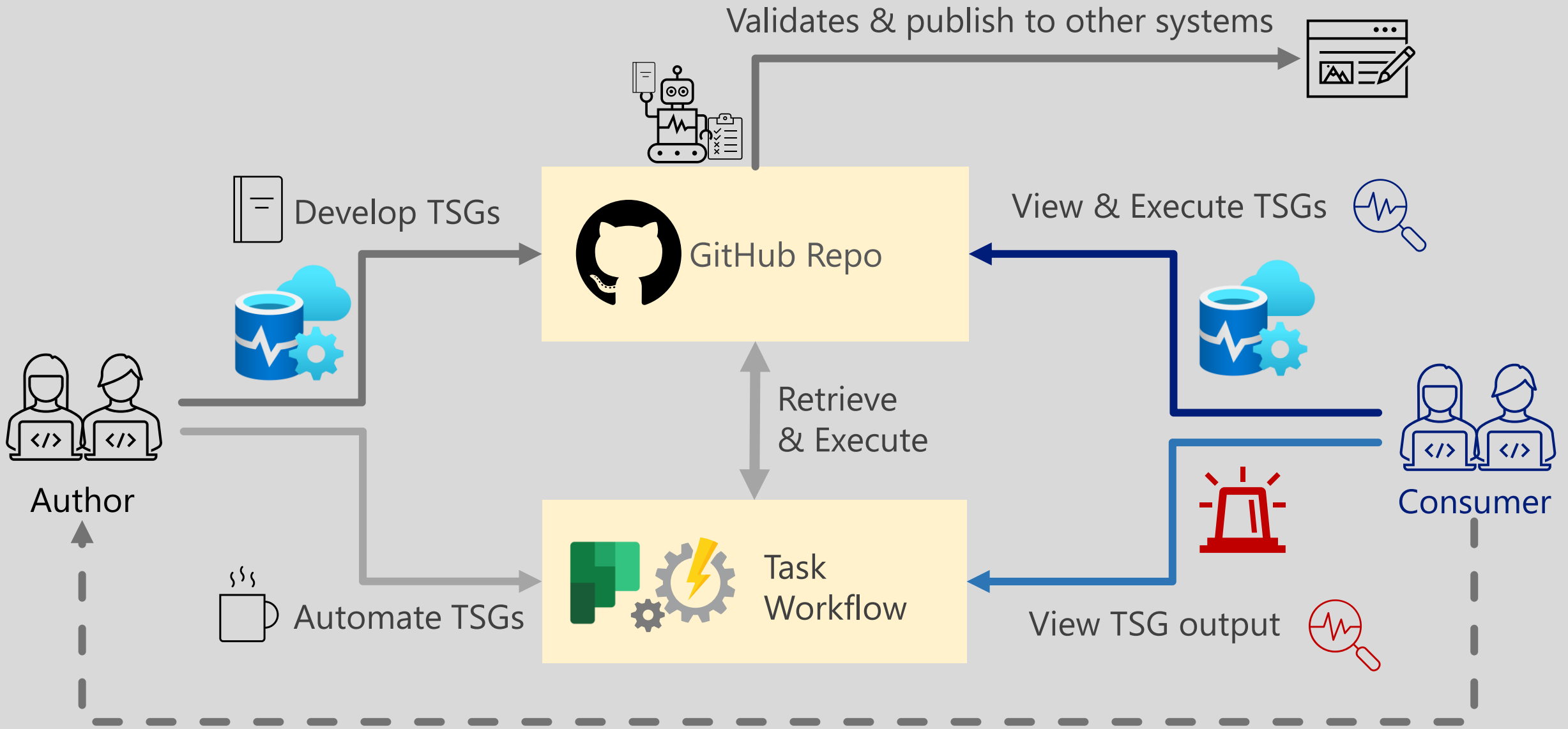
5/1/2021, 12:56 PM

08585817078908637089165

45.74 Seconds



Incident management workflow with **notebooks**



Demo: Automated task workflow with **notebooks**

SQL Agent Job in Azure Data Studio

for automations

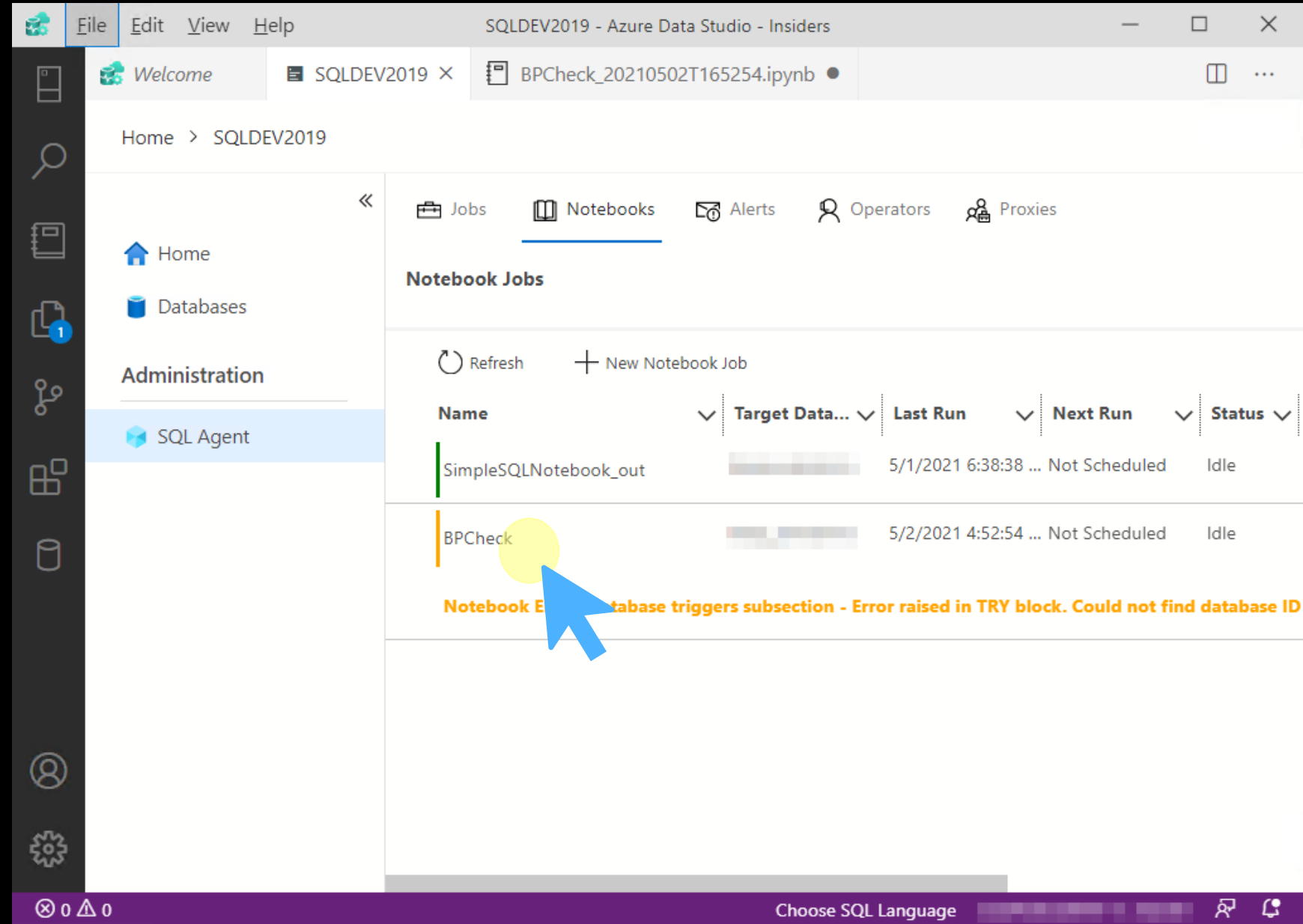


Use SQL Agent to automate SQL Notebooks

SQL Agent

Extension in Azure Data Studio.
Works for SQL Server.
Supports Notebooks (Preview).

<https://bit.ly/3yiwGtF>

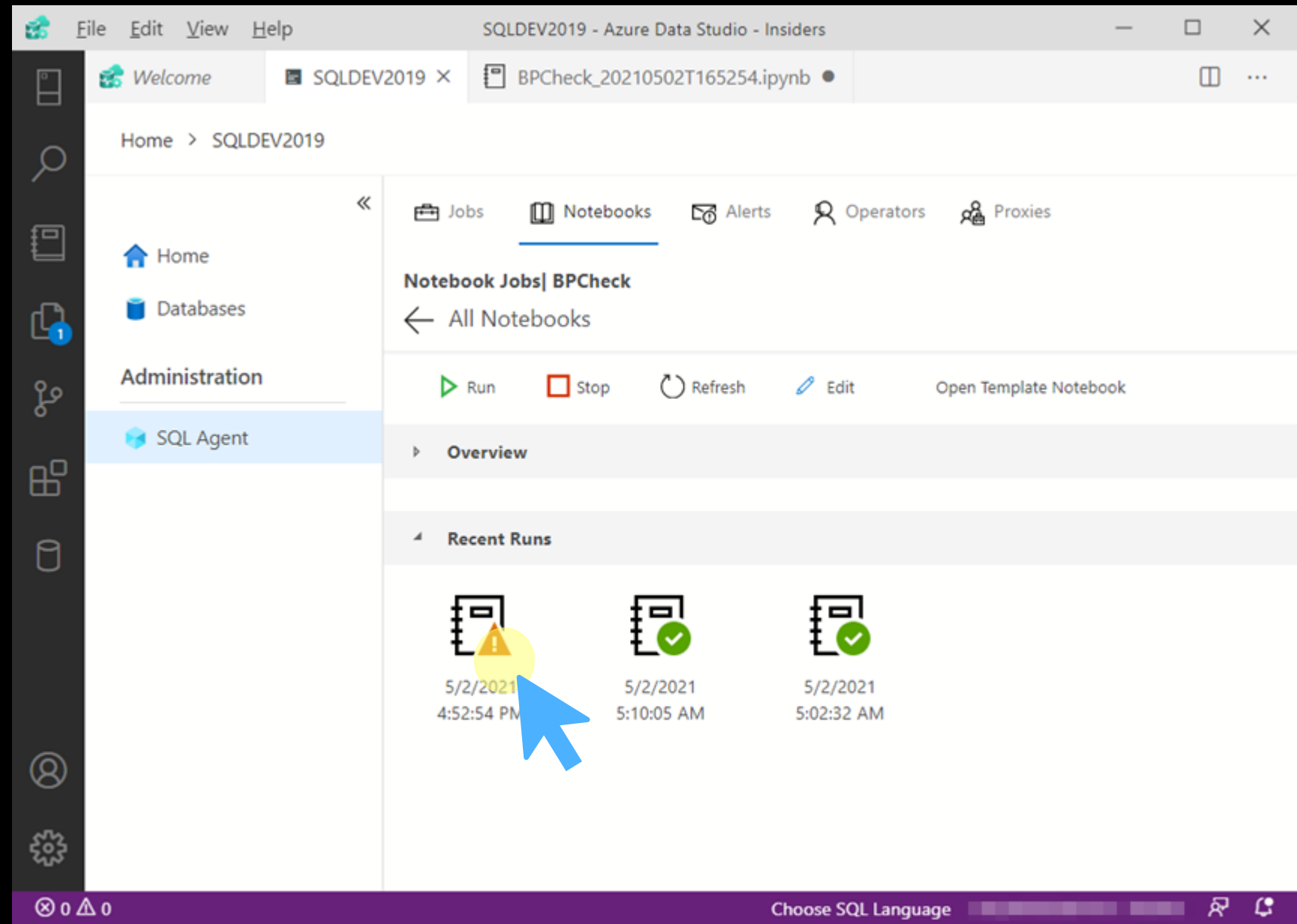


Open previously automated notebook run

SQL Agent

Extension in Azure Data Studio.
Works for SQL Server.
Supports Notebooks (Preview).

<https://bit.ly/3yiwGtF>

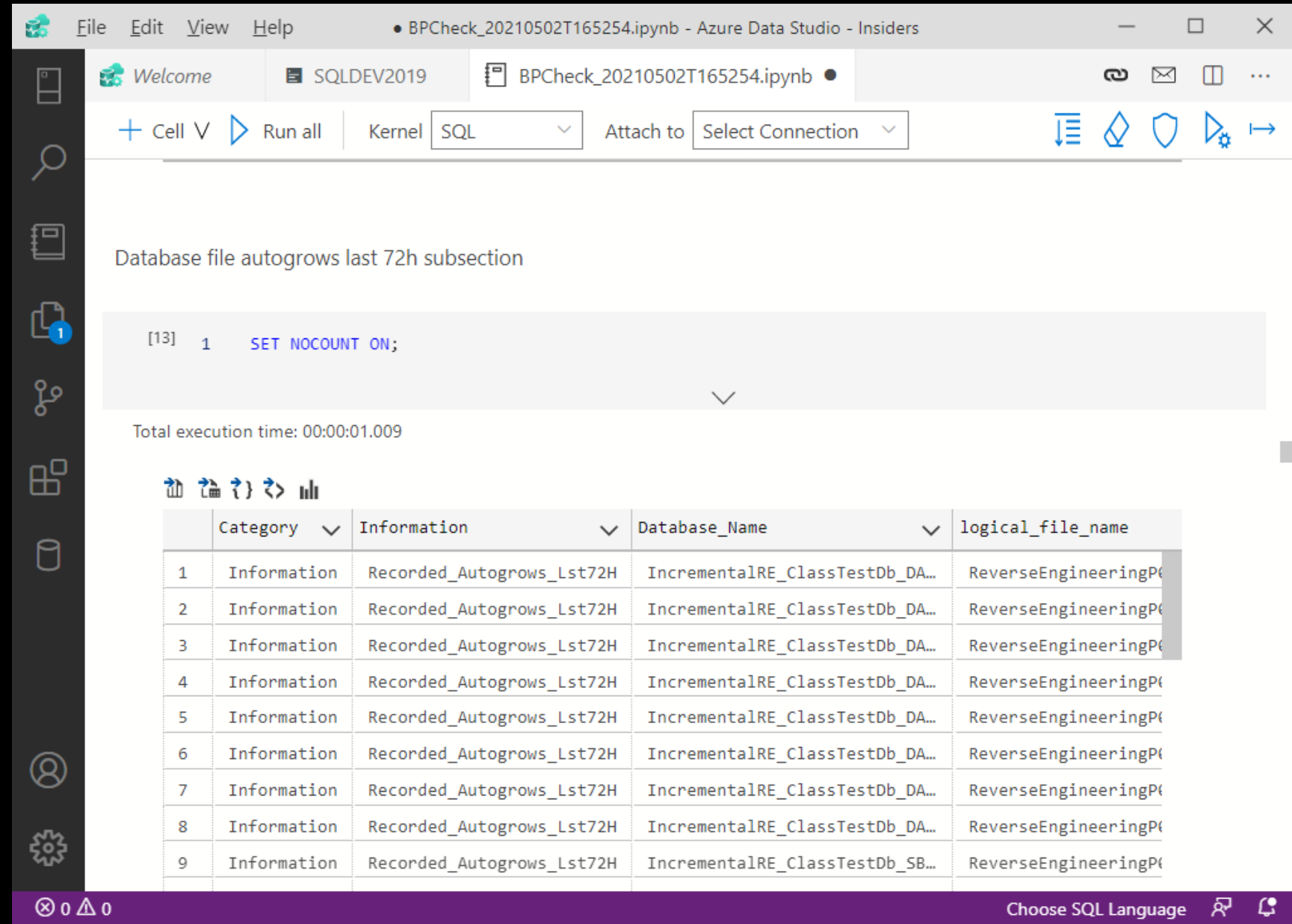


Examine the results of the notebook run

SQL Agent

Extension in Azure Data Studio.
Works for SQL Server.
Supports Notebooks (Preview).

<https://bit.ly/3yiwGtF>



Database file autogrows last 72h subsection

```
[13] 1 SET NOCOUNT ON;
```

Total execution time: 00:00:01.009

	Category	Information	Database_Name	logical_file_name
1	Information	Recorded_Autogrows_Lst72H	IncrementalRE_ClassTestDb_DA...	ReverseEngineeringP6
2	Information	Recorded_Autogrows_Lst72H	IncrementalRE_ClassTestDb_DA...	ReverseEngineeringP6
3	Information	Recorded_Autogrows_Lst72H	IncrementalRE_ClassTestDb_DA...	ReverseEngineeringP6
4	Information	Recorded_Autogrows_Lst72H	IncrementalRE_ClassTestDb_DA...	ReverseEngineeringP6
5	Information	Recorded_Autogrows_Lst72H	IncrementalRE_ClassTestDb_DA...	ReverseEngineeringP6
6	Information	Recorded_Autogrows_Lst72H	IncrementalRE_ClassTestDb_DA...	ReverseEngineeringP6
7	Information	Recorded_Autogrows_Lst72H	IncrementalRE_ClassTestDb_DA...	ReverseEngineeringP6
8	Information	Recorded_Autogrows_Lst72H	IncrementalRE_ClassTestDb_DA...	ReverseEngineeringP6
9	Information	Recorded_Autogrows_Lst72H	IncrementalRE_ClassTestDb_SB...	ReverseEngineeringP6

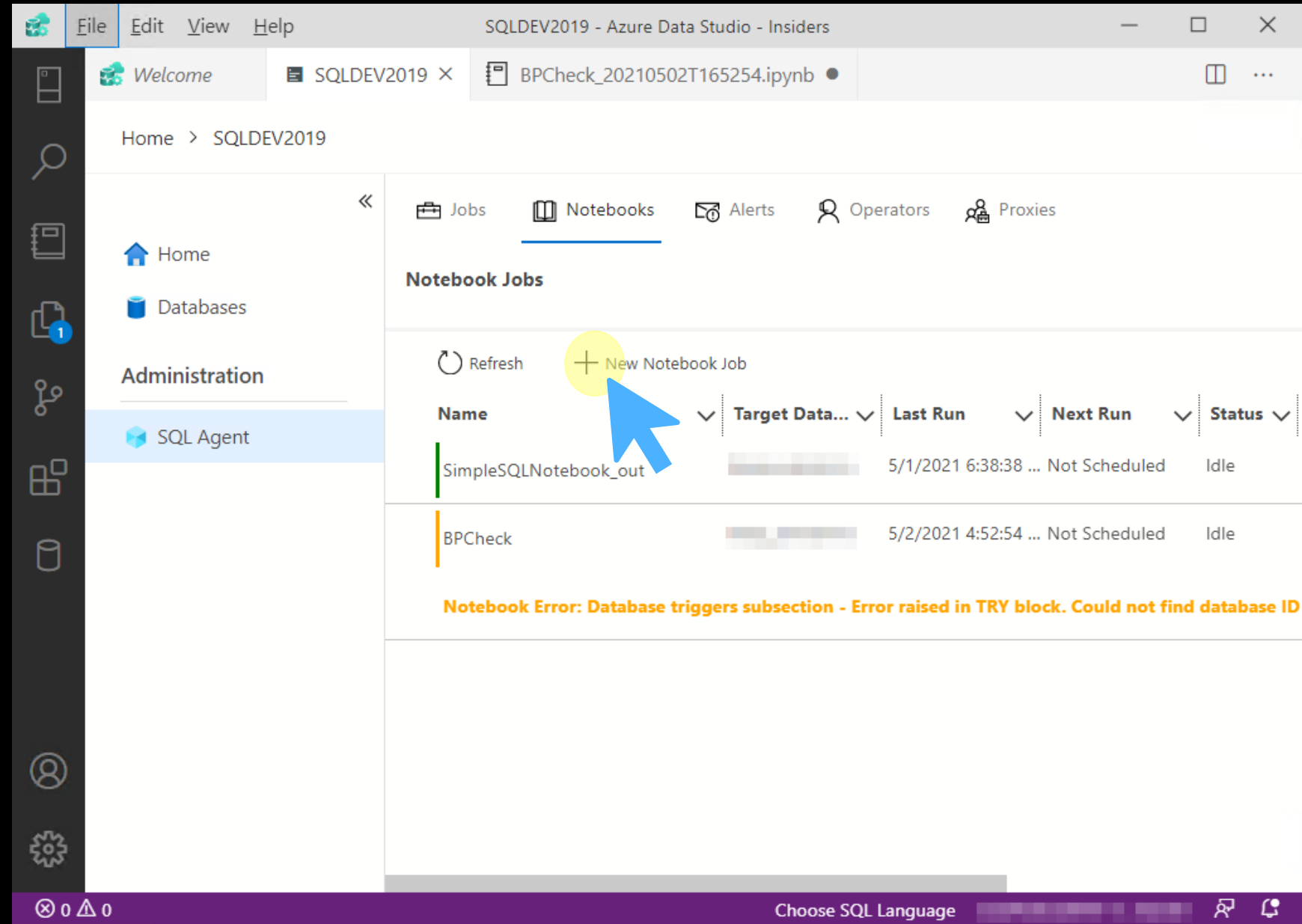
Choose SQL Language

Create a new Notebook Job using SQL Agent

SQL Agent

Extension in Azure Data Studio.
Works for SQL Server.
Supports Notebooks (Preview).

<https://bit.ly/3yiwGtF>



Create a new Notebook job

Notebook Path

Upload a notebook into the Storage Database

Storage Database

Where you'd like to store the notebook definition and job details

Execution database

Which database you'd like to run this notebook against

The screenshot shows the 'New Notebook Job' dialog in Azure Data Studio. The background workspace is titled 'SQLDEV2019 - Azure Data Studio - Insiders' and shows a file explorer with 'Home > SQLDEV2019'. The 'Administration' section is expanded, showing 'SQL Agent'. The 'Notebook Jobs' section is also visible, showing a list of jobs: 'SimpleSQLN...' and 'BPCheck'.

New Notebook Job

Notebook Details

Notebook Path [?]
C:\Temp\MyNotebooks\BPCheck.ipynb

Storage Database * [?]
[Dropdown menu]

Execution Database * [?]
Keep_WideWorldImporters

Job Details

Name *
WeeklyBestPracticeCheck

Owner
[Dropdown menu]

Schedules list

ID	Name	Description
----	------	-------------

OK Cancel

5

Takeaways & other resources



Quick Tips

1. Organize TSG Notebooks as Books / Jupyter books
2. Use Source Control – TSGs are now engineering artefacts
3. Apply software engineering principles to TSGs
4. Use Azure Data Studio for editing, executing and packaging TSGs
5. Remember the “before you run”
Know your goals, understand the gaps, plan it.
6. Check other executable TSGs for inspiration
7. Set guidance / template on TSG
8. Pilot, learn and iterate your TSG changes
9. Identify what you can automate and flight them
10. Use your favorite automation tools (or learn new ones 😎)

Useful resources

Azure Data Studio: <http://aka.ms/AzureDataStudio>
<https://github.com/microsoft/azuredatstudio/issues>

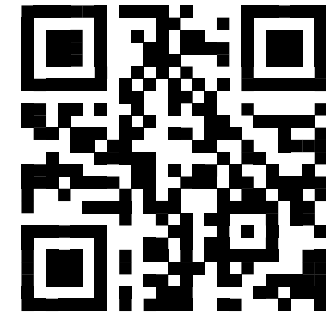
SQL Assessment Toolkit: <https://github.com/microsoft/sql-server-samples/tree/master/samples/manage/sql-assessment-api/notebooks>

SQL Hybrid Toolkit:
<https://github.com/microsoft/tigertoolbox/tree/master/SQL-Hybrid-Cloud-Toolkit/content>

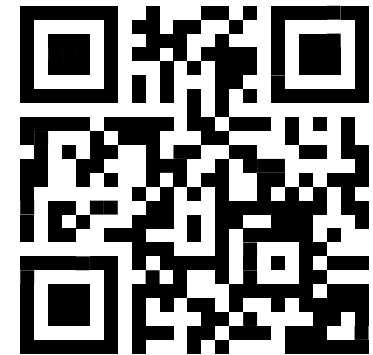
Glenn Berry's Diagnostic Notebooks for SQL Server and Azure SQL DB: <https://glennsqlperformance.com/resources/>

Rob Sewell's Notebook <http://sqldbawithabeard.com>
<https://github.com/SQLDBAWithABeard/JupyterNotebooks>

The SQL Diagnostic (Jupyter) Book by Emanuele Meazzo
<https://tsql.tech/the-sql-diagnostic-jupyter-book/>



Techorama 2021 Session Notes:
<https://bit.ly/3ow3wmM>



Notebooks 101:
<https://bit.ly/2Ryu9uW>

thank you!

Techorama 2021



Julie Koesmarno (@MsSQLGirl)

Program Manager, Microsoft
jukoesma@microsoft.com



Shafiq Rahman

Engineering Manager, Microsoft
Shafiq.Rahman@microsoft.com

