



Twin Talk – Databricks

Documentation



Agenda

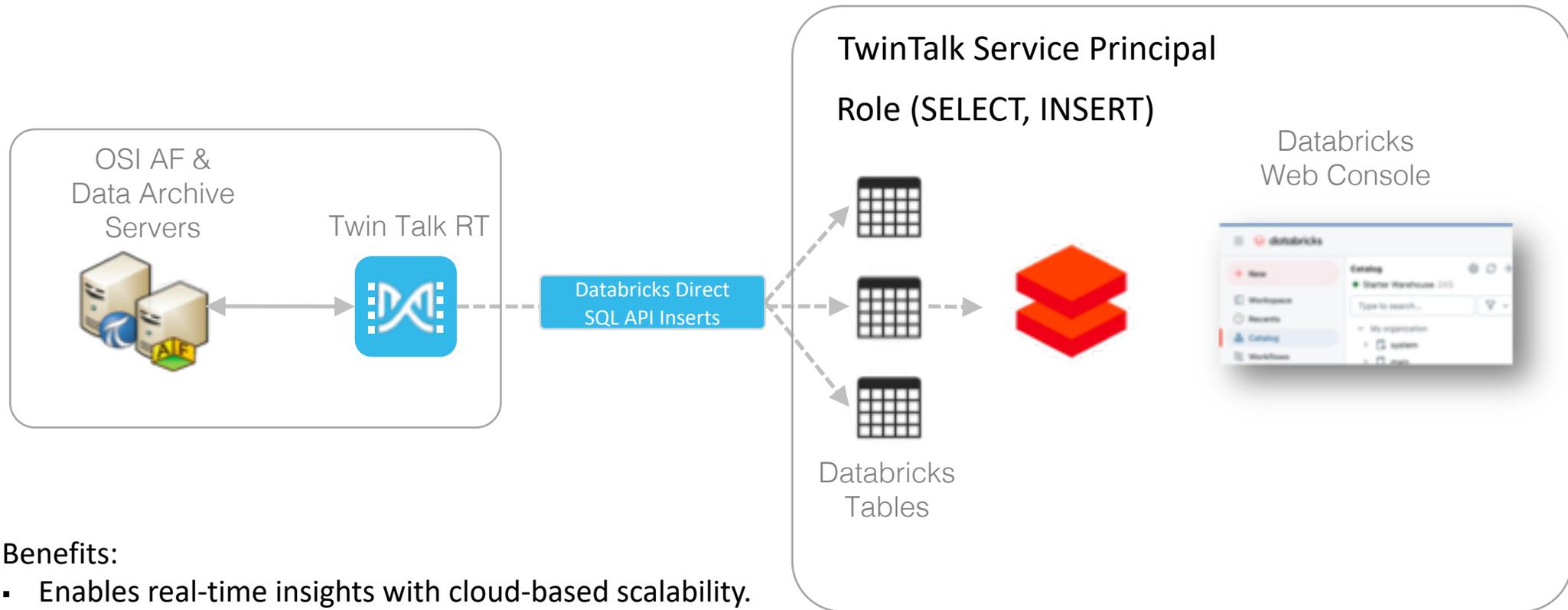


- Introduction Twin Talk and Databricks
- Understand the System
- Live demonstrations
- Best Practices and Trouble-Shooting
- Q&A

Introduction Twin Talk and Databricks

Twin Talk's Role in Data Integration

Twin Talk simplifies the movement of time-series data from sensors to a Databricks workspace / database.



Benefits:

- Enables real-time insights with cloud-based scalability.
- Provides schema flexibility for diverse applications.

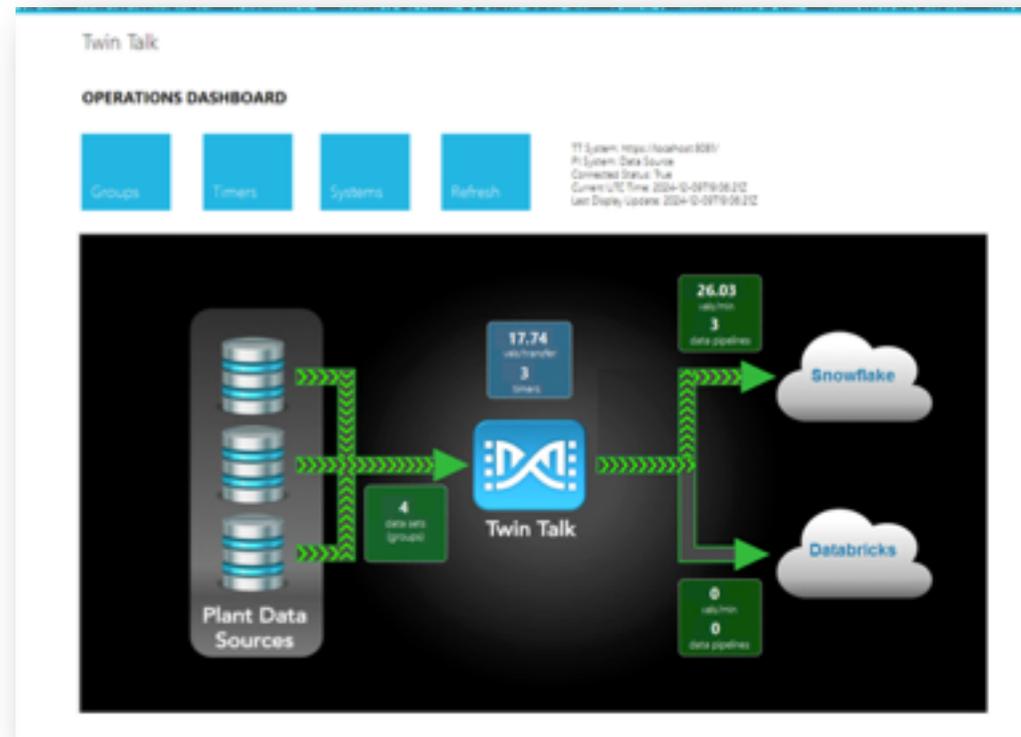
How Twin Talk Works

Twin Talk automates three key processes:

- Data retrieval: Connects to AVEVA/OSIsoft PI Systems.
- Data transformation: Converts and pivots sensor data.
- Data loading: Inserts data into Databricks with customizable schemas.

Example workflow:

1. Fetch time-series data.
2. Transform data using Insert statements.
3. Validate and load into Databricks tables.



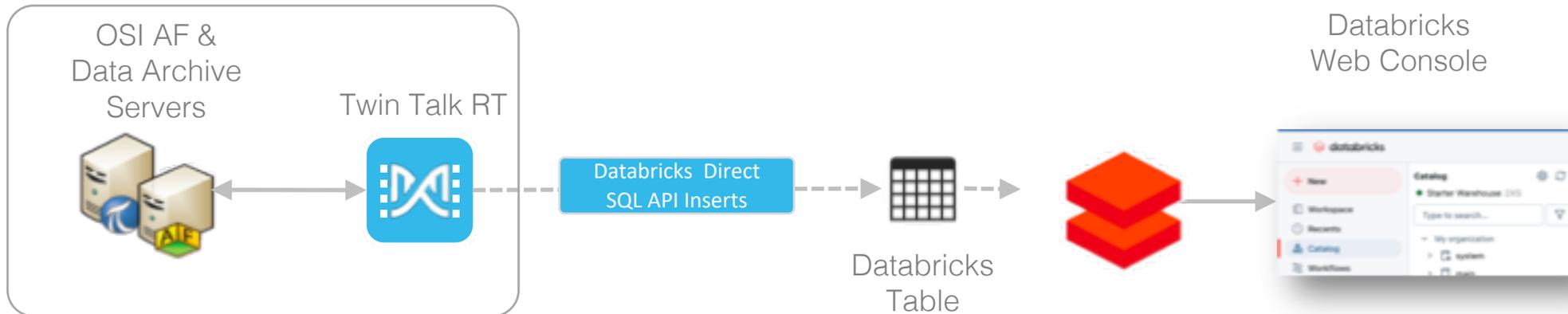
Preparing and Configuring **Databricks** to enable Twin Talk Ingestion

Requirements for Databricks Configuration (Twin Talk to Databricks Ingestion)

- Databricks Service Principal for Twin Talk
- Databricks Role and Privileges for Twin Talk
- Databricks Tables and Schemas Designs

Event-driven, Real-Time Streaming Architecture

Direct INSERT of rows via Databricks SQL API



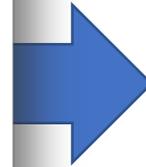
Step 1: Create Databrick TwinTalk Service Principal

Step 2: Create OAUTH Authentication Connection Credentials

- a) Generate Client Id and Client Secret for TwinTalk Service Principal
- b) Test generating OAUTH JWT token with POSTMAN
- c) Test accessing databricks table using JWT token

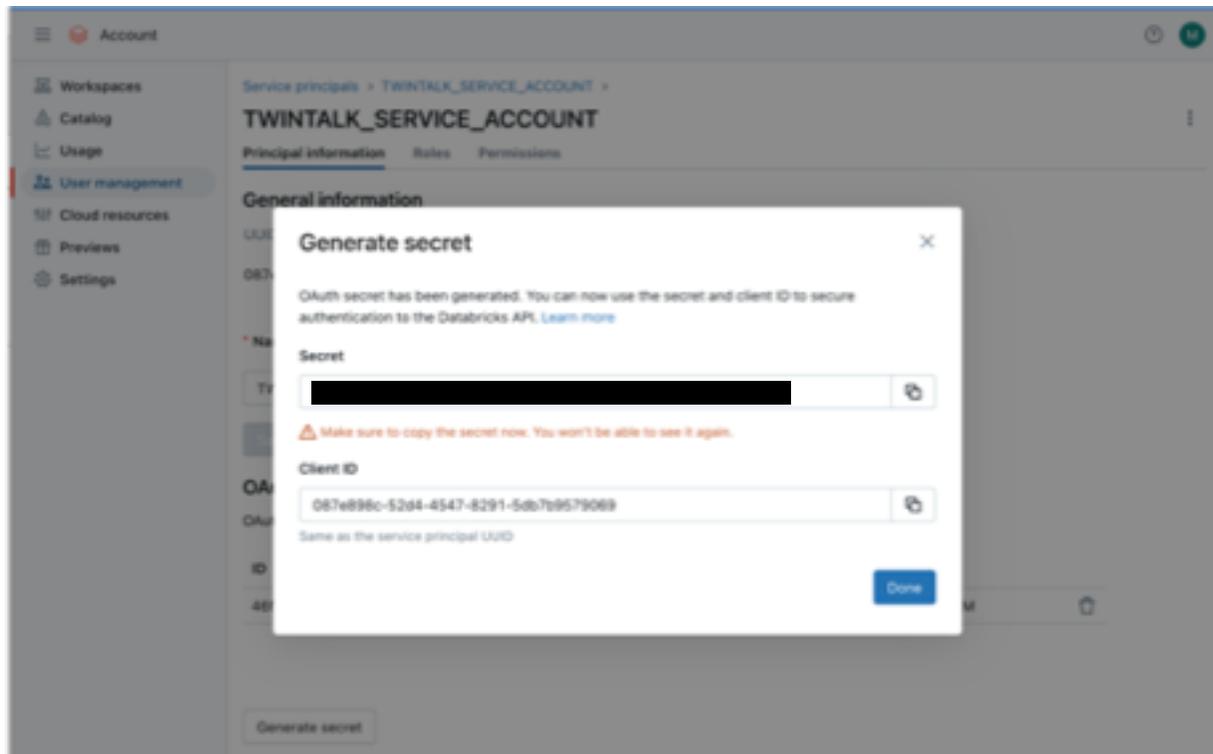
Create a Twin Talk Service Principal in Databricks

The screenshot shows the Databricks interface for adding a service principal. The breadcrumb trail is 'Workspace settings > Identity and access > Service principals > Add service principal'. The 'Service principal name' field contains 'TWIN_TALK_SERVICE_ACCOUNT'. The 'Status' is set to 'Active'. Under 'Entitlements', 'Allow cluster creation' is unchecked, while 'Databricks SQL Access' and 'Allow workspace access' are checked. 'Add' and 'Cancel' buttons are at the bottom.



The screenshot shows the 'Service principal details' page for 'TWIN_TALK_SERVICE_ACCOUNT'. The breadcrumb trail is 'Workspace settings > Identity and access > Service principals > Service principal details'. The 'Application Id' is '420b5f5d-1640-4fc4-abaf-6aedab0c229'. The 'Display Name' is 'TWIN_TALK_SERVICE_ACCOUNT'. The 'Status' is 'Active'. Under 'Entitlements', 'Allow cluster creation' is unchecked, while 'Databricks SQL access' and 'Workspace access' are checked. An 'Update' button is at the bottom.

Generate Client Id and Client Secret for TwinTalk Service Principal

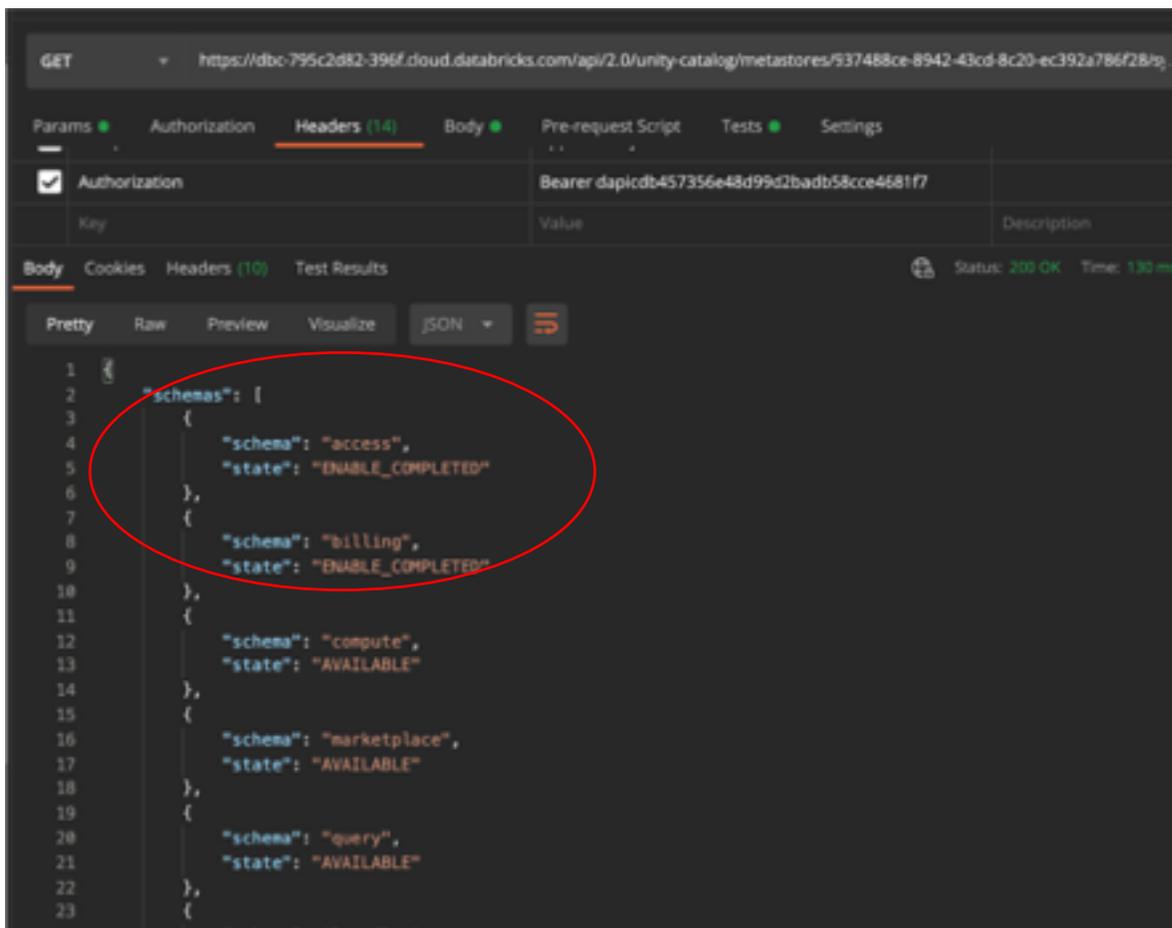


To create a service principal and Auth secret in the Databricks Account Console, log in as an account admin, navigate to the "User Management" section, go to the "Service Principals" tab, click "Add Service Principal" to create a new one, and then select the service principal to generate an Auth secret by clicking "Generate Secret" under the "Auth secrets" section; this will display the secret which you should copy and store securely.

Key steps:

- Access Account Console: Log in to your Databricks account with an account admin user. a
- Navigate to User Management: In the sidebar, select "User Management". •
- Go to Service Principals: Click on the "Service Principals" tab. ®
- Create Service Principal: Click "Add Service Principal" and provide a name for your service principal.
- Generate Auth Secret: Select the newly created service principal, then under "OAuth secrets", click "Generate Secret". o
- Copy Secret: The generated Auth secret will be displayed once, so make sure to copy and store it securely.

Enable Unity Catalog & Audit Logs (“access”)



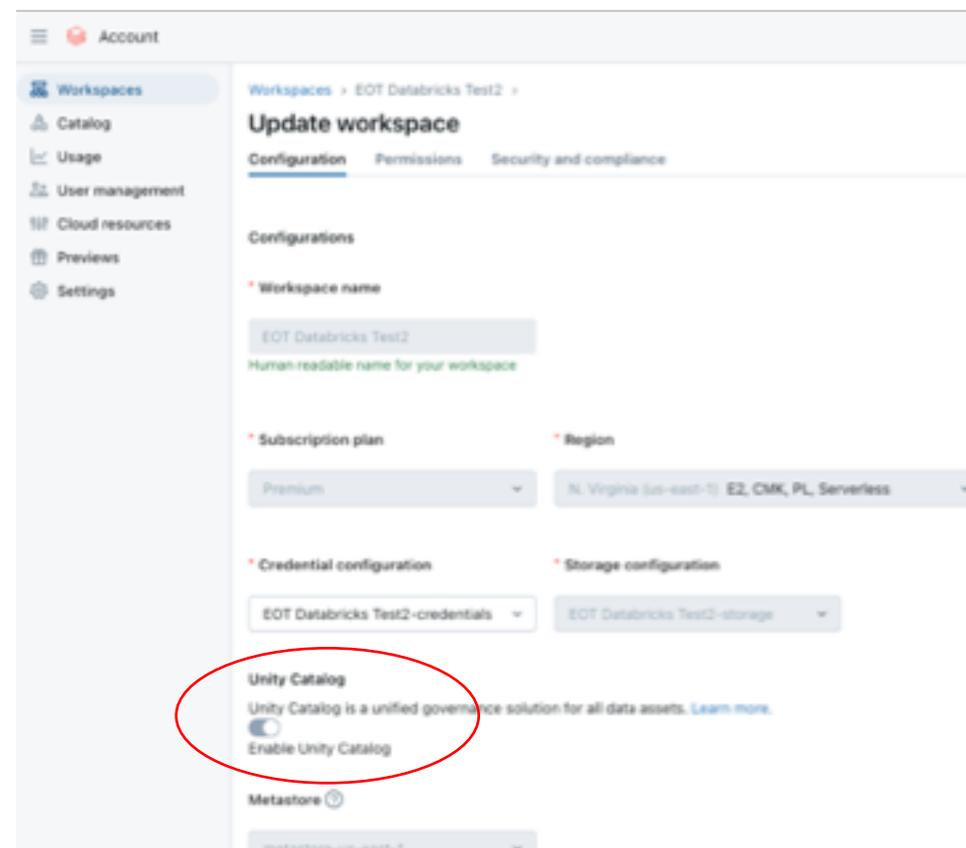
```
GET https://dbc795c2d82-396f.cloud.databricks.com/api/2.0/unity-catalog/metastores/537488ce-8942-43cd-8c20-ec392a786f28/sj...

Headers (14)
Authorization Bearer dapicdb457356e48d99d2badb58cce4681f7

Body
Status: 200 OK, Time: 130 ms

Pretty Raw Preview Visualize JSON

1 {
2   "schemas": [
3     {
4       "schema": "access",
5       "state": "ENABLE_COMPLETED"
6     },
7     {
8       "schema": "billing",
9       "state": "ENABLE_COMPLETED"
10    },
11    {
12     "schema": "compute",
13     "state": "AVAILABLE"
14    },
15    {
16     "schema": "marketplace",
17     "state": "AVAILABLE"
18    },
19    {
20     "schema": "query",
21     "state": "AVAILABLE"
22    },
23    {
```



<https://medium.com/@sc393976/audit-logs-in-databricks-and-how-to-enable-them-in-unity-catalog-f218a25f34d1>

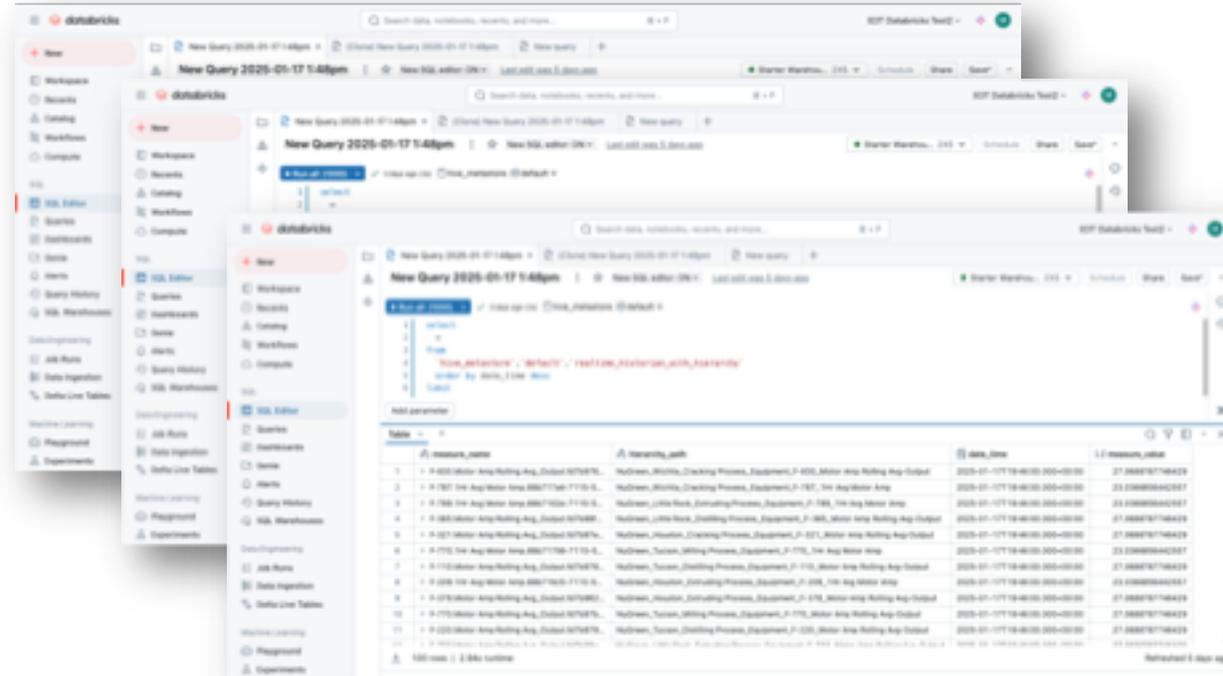
Requirements for Direct INSERT of rows via Databricks SQL API into tables

- 1) Databricks Landing Table or Tables
- 2) Databricks Twin Talk Service Principal
- 3) Databricks Twin Talk Role with SELECT and INSERT privileges for tables

Databricks Landing Table Design and Layout

Design considerations for Databricks Landing Tables

- # of Databricks schemas tables = # of use cases
- Different AF views can translate into different Databricks table



Configure Twin Talk System Environment to connect to Databricks

Version Info

The functionality shown in this certification training require the following software versions:

- Twin Talk Server Version: 6.26
- Twin Talk UI Version: 8.0

Connect Twin Talk to AF Server

If you want to take advantage of the advanced features, such as AF Queries, Graph Manager, Pivoted, Transformed Data Delivery then Connect Twin Talk to the AF Server by following the steps:

- Create a AF Service Account on AF Server
- Fill in the following config parameters in TwinTalk.Config file:

```
<add key="PiSystemName" value="<AF server IP or domain address>" />
<add key="AfDataBaseName" value="<AF database name>" />
<add key="AfUserName" value="<AF service account username>" />
<add key="AfPassword" value="<encrypted AF service account password>" />
<add key="AfRootElementName" value="<AF root element to be scanned>" />
<add key="AfSelectItems" value="ElementsAttributes" />
```



Configure a Twin Talk data pipeline
to send time series data to Databricks



Configure a AF Query for Twin Talk
for Snapshot / Most Recent
Twin Talk data pipeline

Retrieving PI Data with AF Queries: How to Configure AF-Based Queries

Data pipeline type:
Most recent

Timer Info

Group Name	PIAF Cooling Databricks 13	Description	PIAF Hierarchy to TimeStream
Method	GetMostRecentVals	Query	SParentTemplate:Cooling Fan Edit
Kind	Snapshot[digitalstateno]	Time Span	00:02:00
From DateTime	-2m	To DateTime	#tom-00:05:00
<input type="checkbox"/> True or False		<input checked="" type="checkbox"/> True or False 2	
Type	API	Info 0	https://dbc-795c2d82-396f.cloud.databricks.com/api/2.0/sql/st
Info 1	("User-Agent":"EOT_TWIN_TALK","X-Databricks-Autho	Info 2	databricks
Delayed Start (ms)	#abs 00:01:03	Interval (ms)	120000
Add-on Fields	(hierarchy_path: "SDEFChildPath")		
Filters			

Buttons: Load Current, Load Original, Graph Manager, Clear, Cancel, Save

AF Query

Databricks API endpoint

Databricks Query Type:
"databricks"

Twin Talks's AF Query Parameters



AF Query Syntax:

`$<selector>:<query_string>`

selector: selects which AF information is used to query

query_string: string that is used to match against all AF element with PI tags. '*' can be used as wildcard substring. *Fan* will match with "Cooling Fan East"

Examples AF Queries:

`$ParentTemplate:Boiler`

all pi tag where parent element's template name is "Boiler"

`($ParentTemplate:*Fan* && <2>) || (<3> && <4>)`

all pi tag where parent element's Template name contains "Fan"

Selector Options:

- **Name** – AF Element or Attribute Name
- **Description** – AF Description
- **Template** – AF Template Name for Element or Attribute
- **Categories** – AF Categories for Element or Attribute
- **TagName** – PI TagName
- **ChildPath** – AF Path to Element or Attribute
- **ParentPath** – AF Path to Parent Element
- **ParentName** – AF Parent Element Name
- **ParentTemplate** – AF Template Name of Parent Element
- **ParentDescription** – AF Template Name of Parent Element
- **SrcSysName** – AF System Name
- **SrcDBName** – AF Database Name
- **DataType** – Data Type of Attribute

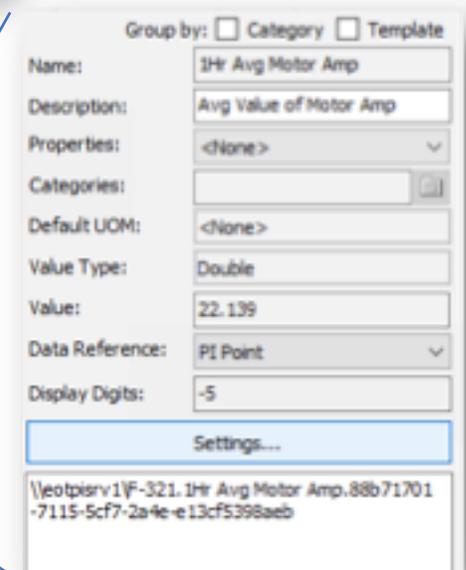
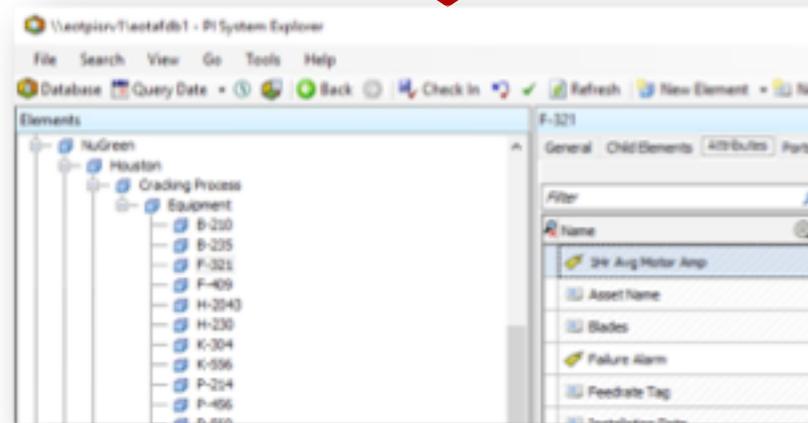
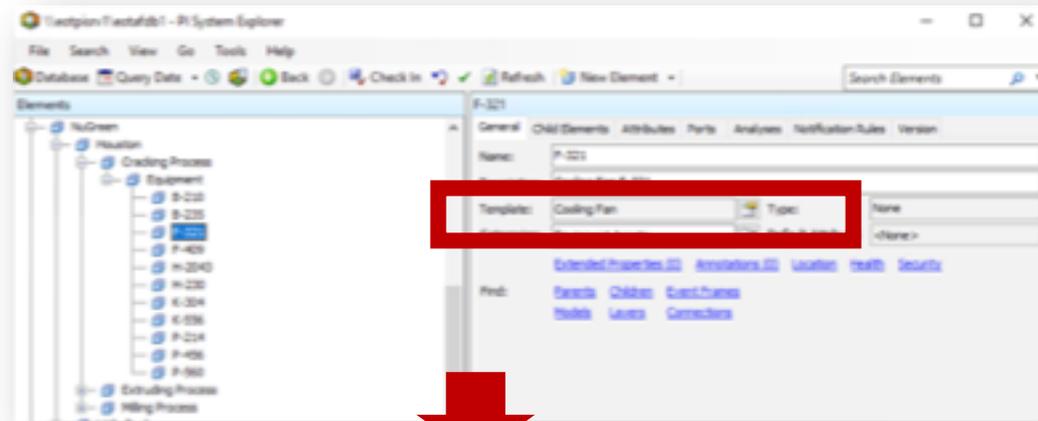
Example AF Query: Select all attributed associated PI Points that have Parent Template: Cooling Fan

In this example, select all pi points that are associated with attributes which parent element template is Cooling Fan

The AF query will find all elements in the entire tree, then find the attributes and identify the appropriate PI Points

In this example:

```
\\eotpisrv1\F-321.1Hr Avg Motor Amp.88b71701-7115-5cf7-2a4e-e13cf5398aeb
```



Result in Twin Talk Graph Hierarchy Manager: All associated PI Points are displayed (that have Parent Template: Cooling Fan)

The screenshot displays the 'Graph and Hierarchy Manager' interface. On the left, a 'Timer Info' dialog box is open, showing details for a query named 'PI Hierarchy for Feedbeam'. The main window is divided into three sections:

- Left Panel:** A tree view under 'AF-PI Server' showing a hierarchy: NuGreen > Little Rock > Tucson > Houston > Extruding Process > Cracking Process > Equipment > F-321. Under F-321, several points are listed, with '1Hr Avg Motor Amp' highlighted in blue.
- Middle Panel:** A breadcrumb path: 'https://dbc-795c2d82-396f.cloud.databricks.c' > NuGreen.
- Right Panel:** A configuration panel titled 'From Output Graph: 1Hr Avg Motor Amp' with the following fields:
 - Name: 1Hr Avg Motor Amp
 - Description: Avg Value of Motor Amp
 - Value Type: Double
 - Parent: F-321
 - Tag Name: F-321.1Hr Avg Motor Amp.88b71701-7115-5cf7-2a4e-e13cf5398aeb
 - Template: 1Hr Avg Motor Amp
 - Categories: Equipment Assets

Buttons at the bottom include 'Map Source to Custom', 'Export Graph as JSON', 'Cancel', and 'Save'. A red box highlights the configuration fields, and a red line connects the highlighted '1Hr Avg Motor Amp' point in the hierarchy to this box. Blue lines also point from the 'Timer Info' dialog to the hierarchy and configuration panels.

Result in Twin Talk Group: All associated PI Points are part of the group (where the attribute's parent template is: Cooling Fan)

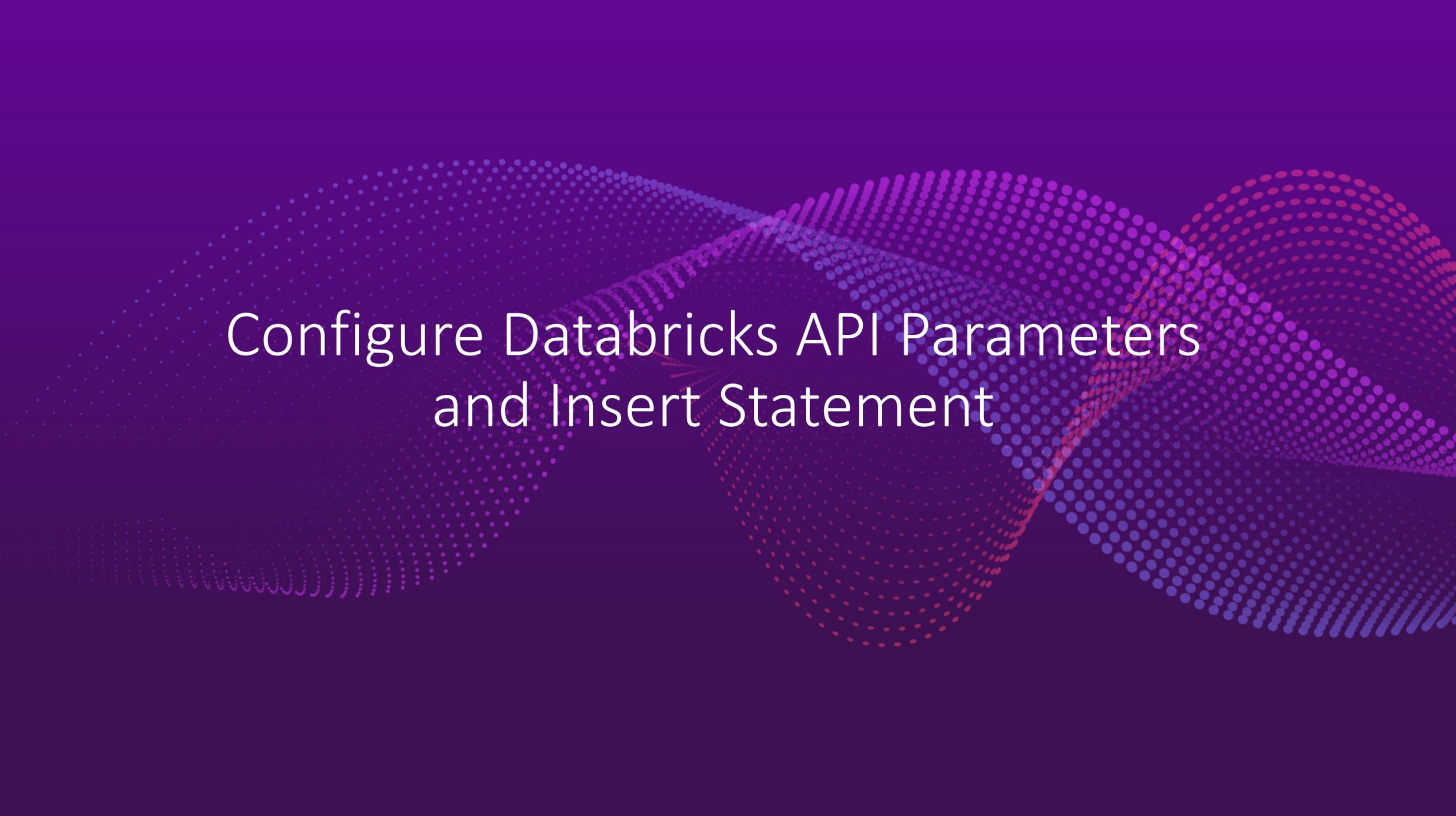
Twin Talk Timer AF Query

Query

PI Points in Twin Talk Group

The screenshot shows the 'Twin Talk' interface. On the left, under 'GROUPS', there is a list of groups. One group, 'PIAF Cooling Datablocks 13', is selected. A 'Group Info' dialog box is open, showing the following details:

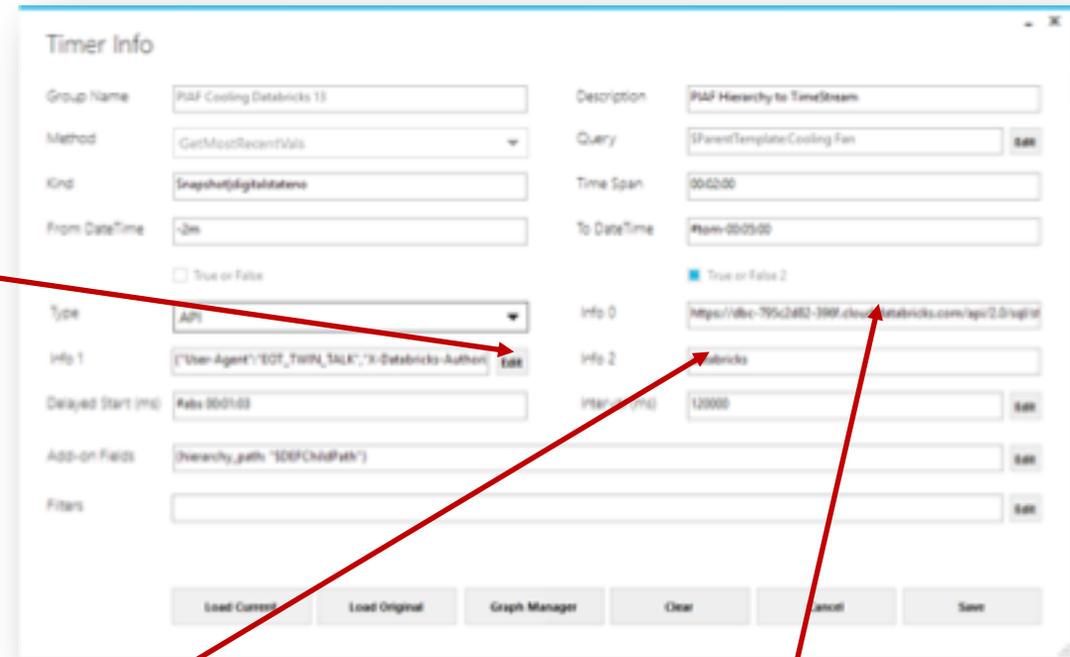
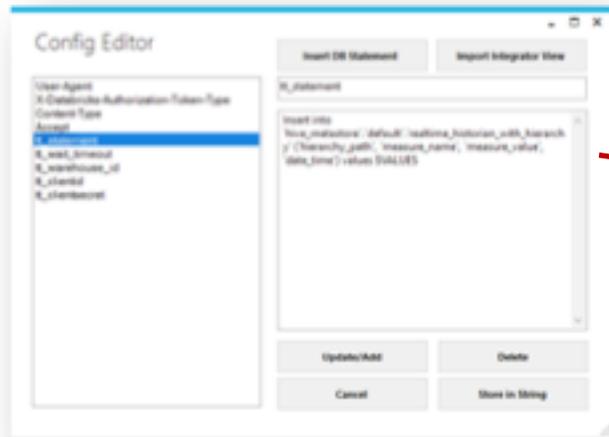
- Group Name: PIAF Cooling Datablocks 13
- Query:
- Tag Names: [F-303.Motor Amp Rolling Avg_Output.f276892-a3e0-52aa-2b95-edc7029142a"; F-220.Motor Amp Rolling Avg_Output.f276762-a3e0-52aa-2b95-edc7029142a"; F-488.Motor Amp Rolling Avg_Output.f276798-a3e0-52aa-2b95-edc7029142a"; F-727.Motor Amp Rolling Avg_Output.f276762-a3e0-52aa-2b95-edc7029142a"; F-378.Motor Amp Rolling Avg_Output.f276929-a3e0-52aa-2b95-edc7029142a"; F-321.Motor Amp Rolling Avg_Output.f276765-a3e0-52aa-2b95-edc7029142a"; F-313.1Hr Avg Motor Amp.88b719b3-7115-5c77-2a4e-e13c75399aeb"; F-100.Motor Amp Rolling Avg_Output.f276715-a3e0-52aa-2b95-edc7029142a"; F-198.Motor Amp Rolling Avg_Output.f27695ca-a3e0-52aa-2b95-edc7029142a"; F-486.1Hr Avg Motor Amp.88b71726-7115-5c77-2a4e-e13c75399aeb"; F-723.Motor Amp Rolling Avg_Output.f2768cc-a3e0-52aa-2b95-edc7029142a"; F-506.1Hr Avg Motor Amp.88b7173a-7115-5c77-2a4e-e13c75399aeb"; F-293.1Hr Avg Motor Amp.88b717e6-



Configure Databricks API Parameters and Insert Statement

Configure Databricks Core Parameters (Info 0 and Info 2)

1) Click on “Edit” for Info 1 to fill in Databricks API Parameters and Insert Statement (details follow)



Info 2: “databricks”

2) Filling in the input fields Info 0 and 2:

Info 0: Databricks REST API endpoint
`https://<workspace id>.cloud.databricks.com/api/2.0/sql/statements`

Filling out API POST Headers Parameters and Databricks SQL Insert Statement and Params

Config Editor

```
User-Agent
X-Databricks-Authorization-Token-Type
Content-Type
Accept
tt_statement
tt_wait_timeout
tt_warehouse_id
tt_clientid
tt_clientsecret
```

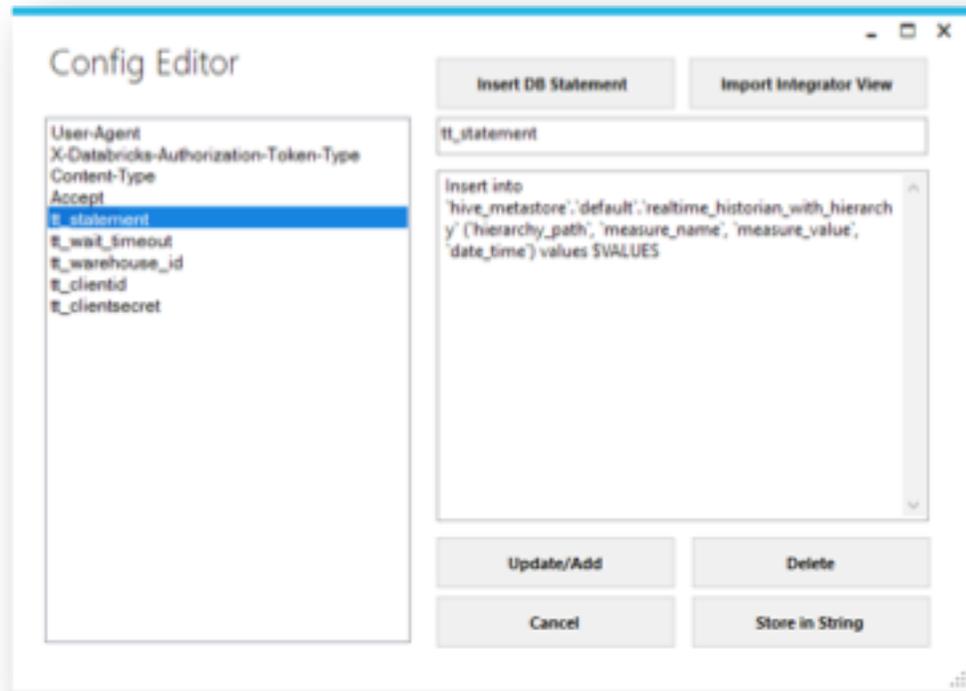
POST Headers for Databricks SQL API Call

```
POST Headers
"User-Agent":"EOT_TWIN_TALK"
"X-Databricks-Authorization-Token-Type":"KEYPAIR_JWT"
"Content-Type":"application/json"
"Accept":"application/json"
Databricks SQL
"tt_statement":"Insert into ..." (Details are following)
"tt_wait_timeout":"1000"
"tt_warehouse_id":"TT_DB"
"tt_clientid":"PUBLIC"
"tt_clientsecret":"COMPUTE_WH"
```

Client name for Databricks
Authentication is of type encrypted JWT
Standard API Post Param
Standard API Post Param
Databricks SQL Statement
API timeout
Databricks Warehouse ID
Databricks Service Principal Client ID
Databricks Service Principal Client Secret

POST Payload for SQL API Call

The SQL Insert Statement



When selecting Info 2: “databricks” here is the syntax for writing an insert statement:

```
Insert into <database table> ([AddOn  
Values, ]<name for pi tagname>, <name for  
pi value>, <name for pi timestamp>) $VALUES
```

All <names> are customizable. \$VALUES will fill in all pi records for all selected and queries selected PI points. For this example the syntax looks like this:

```
Insert into  
`hive_metastore`.`default`.`realtime_historian_with_hierarchy` (`hierarchy_path`,  
`measure_name`, `measure_value`,  
`date_time`) values $VALUES
```

Result: Databricks is getting populated with data from Twin Talk's Data pipeline from Most/Recent Timer

The screenshot shows a Databricks SQL query editor interface. At the top, it displays 'New Query 2025-01-17 1:48pm' and 'New SQL editor: ON'. The query text is as follows:

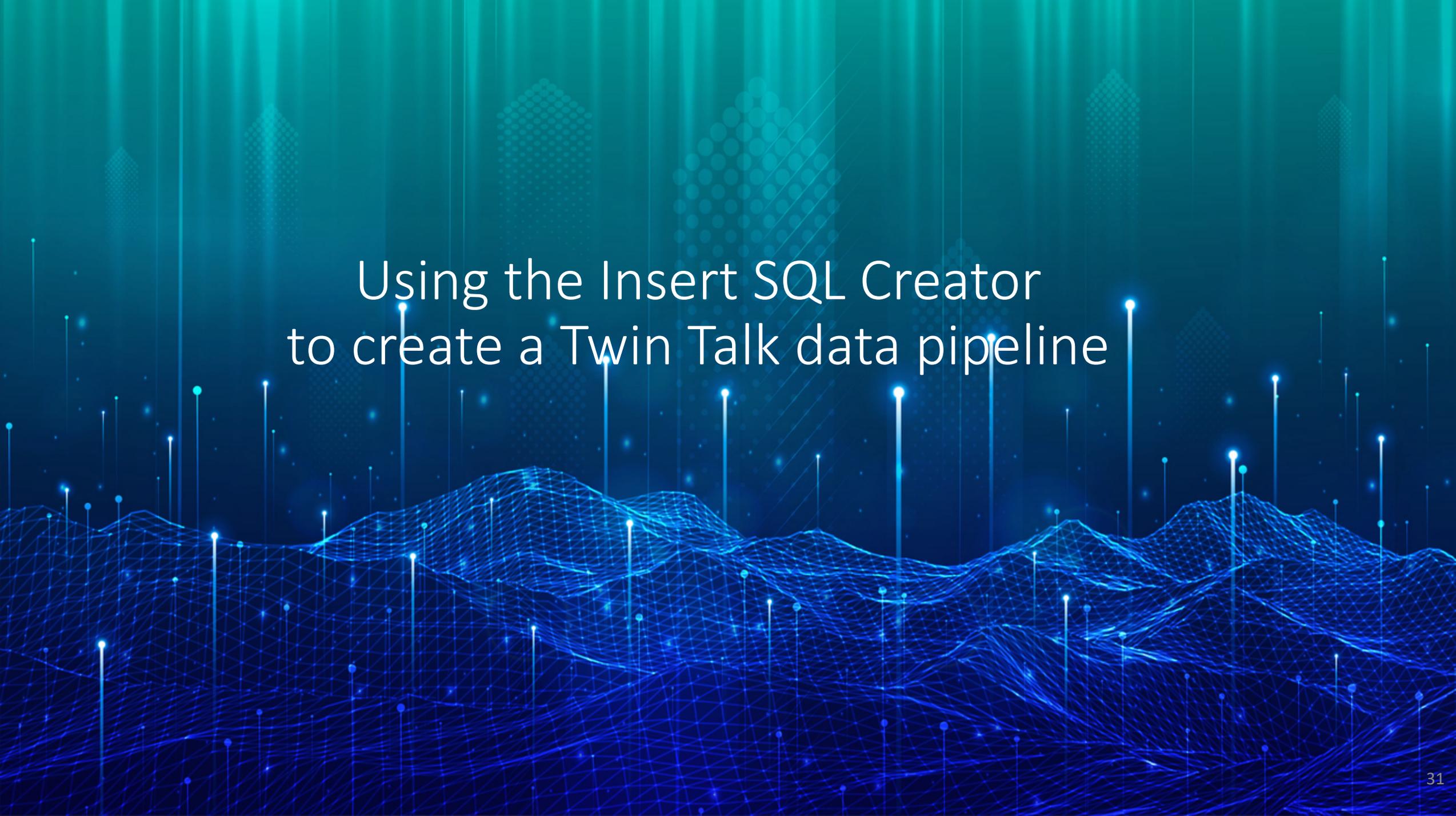
```
1 | select
2 | *
3 | from
4 |   'hive_metastore`.`default`.`realtime_historian_with_hierarchy`
5 | order by date_time desc
6 | limit
```

Below the query editor, a table of results is displayed with the following columns: measure_name, hierarchy_path, date_time, and measure_value. The table contains 11 rows of data, all with a date_time of 2025-01-17T19:46:00.000+00:00.

	measure_name	hierarchy_path	date_time	measure_value
1	> F-930.Motor Amp Rolling Avg_Output.Id7b976...	NuGreen_Wichita_Cracking Process_Equipment_F-930_Motor Amp Rolling Avg-Output	2025-01-17T19:46:00.000+00:00	27.0688787746429
2	> F-787.1Hr Avg Motor Amp.88b717a4-7115-5...	NuGreen_Wichita_Cracking Process_Equipment_F-787_1Hr Avg Motor Amp	2025-01-17T19:46:00.000+00:00	23.0366856442557
3	> F-789.1Hr Avg Motor Amp.88b7182e-7115-5...	NuGreen_Little Rock_Extruding Process_Equipment_F-789_1Hr Avg Motor Amp	2025-01-17T19:46:00.000+00:00	23.0366856442557
4	> F-365.Motor Amp Rolling Avg_Output.Id7b98f...	NuGreen_Little Rock_Distilling Process_Equipment_F-365_Motor Amp Rolling Avg-Output	2025-01-17T19:46:00.000+00:00	27.0688787746429
5	> F-321.Motor Amp Rolling Avg_Output.Id7b97e...	NuGreen_Houston_Cracking Process_Equipment_F-321_Motor Amp Rolling Avg-Output	2025-01-17T19:46:00.000+00:00	27.0688787746429
6	> F-770.1Hr Avg Motor Amp.88b71756-7115-5...	NuGreen_Tucson_Milling Process_Equipment_F-770_1Hr Avg Motor Amp	2025-01-17T19:46:00.000+00:00	23.0366856442557
7	> F-110.Motor Amp Rolling Avg_Output.Id7b978...	NuGreen_Tucson_Distilling Process_Equipment_F-110_Motor Amp Rolling Avg-Output	2025-01-17T19:46:00.000+00:00	27.0688787746429
8	> F-209.1Hr Avg Motor Amp.88b716c5-7115-5...	NuGreen_Houston_Extruding Process_Equipment_F-209_1Hr Avg Motor Amp	2025-01-17T19:46:00.000+00:00	23.0366856442557
9	> F-378.Motor Amp Rolling Avg_Output.Id7b962...	NuGreen_Houston_Extruding Process_Equipment_F-378_Motor Amp Rolling Avg-Output	2025-01-17T19:46:00.000+00:00	27.0688787746429
10	> F-770.Motor Amp Rolling Avg_Output.Id7b97b...	NuGreen_Tucson_Milling Process_Equipment_F-770_Motor Amp Rolling Avg-Output	2025-01-17T19:46:00.000+00:00	27.0688787746429
11	> F-220.Motor Amp Rolling Avg_Output.Id7b979...	NuGreen_Tucson_Distilling Process_Equipment_F-220_Motor Amp Rolling Avg-Output	2025-01-17T19:46:00.000+00:00	27.0688787746429

Example Logfile output (Level: Info)

```
2025-01-22 13:40:24Z, info, 158, PIVentBus, "_GetValues: (parallel bulk partial call) _CurrentValues number of values: 14"
2025-01-22 13:40:24Z, info, 158, PIVentBus, "_GetValues: (parallel bulk partial call) _CurrentValues, for 10 points"
2025-01-22 13:40:24Z, info, 158, PIVentBus, "_GetValues: (parallel bulk partial call) _CurrentValues number of values: 10"
2025-01-22 13:40:24Z, info, 158, PIVentBus, "_GetValues: (parallel bulk total) _CurrentValues, for 52 points"
2025-01-22 13:40:24Z, info, 158, PIVentBus, "piGetMostRecentValsBulk: retrieved 52 values for 52 points"
2025-01-22 13:40:24Z, notify, 158, ConnectorBus, "3: {"reqMsg":{"contact":"TIMERCALL","msgMethod":"GetMostRecentVals","msgId":"2f645488-a8bd-40f8-8892-df543ebb3237","msgTimestamp":"0001-01-01T00:00:00","tagGroupName":"PIAF Cooling Databricks 13","tagNames":null,"fromDateTime":"2025-01-22T19:34:00Z","toDateDateTime":"2025-01-22T19:36:00Z","query":"$ParentTemplate:Cooling Fan","timeSpan":"00:02:00","kind":"Snapshot|digitalstateno","trueOrFalse":false,"trueOrFalse2":true,"piSetTags":null,"subscriberInfo":null,"type":"API","info0":"https://dbc-795c2d82-396f.cloud.databricks.com/api/2.0/sql/statements","info1":{"\"User-Agent\": \"EOT_TWIN_TALK\", \"X-Databricks-Authorization-Token-Type\": \"KEYPAIR_JWT\", \"Content-Type\": \"application/json\", \"Accept\": \"application/json\", \"tt_statement\": \"Insert into `hive_metastore`.`default`.`realtime_historian_with_hierarchy` (`hierarchy_path`, `measure_name`, `measure_value`, `date_time`) values $VALUES \", \"tt_wait_timeout\": \"30s\", \"tt_warehouse_id\": \"29421cc055732758\", \"tt_clientid\": \"420b5f5d-1640-4fc4-aba6-daadeab0c229\", \"tt_clientsecret\": \"dose7d1b899ba387d920a8521ba6456bfe5e\"}, \"info2\": \"databricks\", \"addonToValues\": \"{hierarchy_path: \"$DEFChildPath\"}\", \"description\": \"PIAF Hierarchy to TimeStream\", \"intervalString\": \"120000\", \"authHeader\": \"Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJFT1R1aWQiOiJhZG1pbjEiLCJmYm9iOiJlNjE1NjE4OTE0TE0NTUsImV4cCI6MTU2MTg5NTA1NSwiawF0IjoxNTYxODkxNDU1fQ.cF7e0Tm95Ciy064-1v9ou710L8nKR5kBARm6moJwS1ddkKwqTDQIw2ffhcnLx6XYackkQq0G5B_UZcx8s45Dkw86vMq8Q_1Wsnx3R7ZMa36VjNneAI4mAs9hiHH3R4rOqcEpD6QGvQP89RduSjUnhqaU6yd51gXv1czf_9B81sYa4CFQ2RmMsc-WSNpcTS007qBV9Y6T4z79FsezklKeSaUKP1jat7MBA_NUWP86u-o24Sq8DG0kn-HOVq0BJXg0XwOy5hGJ08vp7UpyKyalvZ8eN0Yb6eXWjKA81tkGcNiQvvnv4lklzbz80bw_nRq66amSyY_yDHF6bLnSCHMJVQ\", \"postUrl\": \"https://localhost:8081/\", \"delayedStartString\": \"63000\", \"filters\": \"\", \"aTimer\": {\"AutoReset\": true, \"Enabled\": true, \"Interval\": 119990.26150000001, \"Site\": null, \"SynchronizingObject\": null, \"Container\": null}, \"calcStartTime\": \"2025-01-22T13:42:24.1328831-06:00\", \"timerDueTime\": \"2025-01-22T13:42:24.1326216-06:00\", \"active\": true, \"interval\": 120000.0, \"delayedStart\": 0.0, \"filters\": null, \"filtersString\": null, \"fromDateTimeString\": \"-2m\", \"toDateDateTimeString\": \"#tom-00:05:00\", \"delayedStartString\": \"#abs 00:01:03\", \"intervalString\": \"120000\", \"lastCallNumberOfValues\": 52, \"previousLastCallNumberOfValues\": 52, \"lastCallDateTime\": \"2025-01-22T13:38:24.1328831-06:00\", \"previousLastCallDateTime\": \"2025-01-22T13:36:24.1328831-06:00\", \"averageValuesPerMinute\": 26.0, \"previousAverageValuesPerMinute\": 26.0, \"averageCallsPerMinute\": 0.5, \"totalNumberOfValues\": 61048, \"nextQueryRun\": \"2025-01-22T20:07:00Z\", \"offsetMins\": 7, \"modulusMins\": 60, \"autoQueryOn\": true, \"dataSourceType\": null, \"dataSourceConnectionString\": null, \"dataSourceUrl\": null, \"rootElementName\": null, \"parentElement\": null, \"dataSourceQueryString\": null, \"pointList\": null, \"pointListCustomized\": null, \"backupTimerEndTime\": null}"
2025-01-22 13:40:24Z, info, 6, ConnectorAWS, "PostObjectAsync: About to post object to https://dbc-795c2d82-396f.cloud.databricks.com/api/2.0/sql/statements"
2025-01-22 13:40:25Z, info, 5, ConnectorAWS, "Send: Timer Call Response { statusCode: \"200\", statusDescription: \"OK\" }"
```



Using the Insert SQL Creator to create a Twin Talk data pipeline

Starting the Insert SQL Creator

Timer Info

Group Name	PIAF Cooling Detectors 13	Description	PIAF Hierarchy to TimeStream
Method	GetMostRecentVals	Query	SParentTemplateCooling Fan
Kind	Snapshot/digitalstatus	Time Span	00:00:00
From DateTime	-2m	To DateTime	Now-00:00:00
Type	API	URL	https://blc-7002482-2988.cloud.databricks.com/api/2.0/api/2
Info 1	{'User-Agent': 'EOT_TWIN_TASK', 'X-Databricks-Authorization-Token-Type': 'API_TOKEN'}	Info 2	databricks
Delayed Start (ms)	14400000	Timeout (ms)	120000
Add-on Fields	hierarchy_path: 'SOPONOffPath'		
Filters			

Load Current Load Original Graph Manager Clear Cancel Save

Config Editor

User Agent
X-Databricks-Authorization-Token-Type
Content-Type
Accept

hierarchy_path

hierarchy_path
measure_name
measure_value
data_time

Insert SQL Statement

Import Integrator View

h_statement

Insert into
hierarchy_path, measure_name, measure_value, data_time) values (\$h_statement)

Update/Add Delete

Cancel Show in String

Create Insert SQL Statement

AF Attributes

Active
SCADA Tag Prefix
PI Point Source
InterfaceNode
Notes
MerckID

AF PI Points

Tank Vapor Pressure Low Limit
Tank Vapor Pressure Hi Limit
Tank Vapor Pressure
Transmitter Failure Alarm

Add Attr Del Attr Add Points Del Points

SCADA Tag Prefix

Pick Index Field

AddOnFields

{}

Pick Timestamp Field

Database Table Name

SHAPETABLE_15

SQL Statement

Insert into SHAPETABLE_15 SIND:'SCADA Tag Prefix'
STIME:PRODUCTION_TIME SCOLPOINT:'Tank Vapor Pressure Low Limit'
SCOLPOINT:'Tank Vapor Pressure Hi Limit'
SCOLPOINT:'Tank Vapor Pressure'
SCOLPOINT:'Transmitter Failure Alarm'
SUSEDATATYPE

Cancel Update Timer

What is the Insert SQL Creator and how does it work

The Insert SQL Creator allows you to create a pivot Twin Talk SQL Statement with a few clicks. The AF Attributes and AF PI Points boxes are populated from the result of the time's AF query (needs to be in the Timer's "query" field)

- 1) For the queried AF metadata, it displays
 - a) the AF Attributes that have valid PI Points associated with it
 - b) the AF Attributes that are static, formulas, etc.
- 2) You pick a AF Attribute as Index Field. An Index field is what groups all attributes together (i.e. Element's template, parent, categories, etc.)
- 3) You can rename the TimeStamp field
- 4) You can select with AF Attributes (PI Points) should be in the template
- 5) You can select with AF Attributes (static, etc) should be in the template
- 6) You can define the database table name
- 7) You manually change the Insert Statement

Create Insert SQL Statement

AF Attributes

- Active
- SCADA Tag Prefix
- PI Point Source
- InterfaceNode
- Notes
- MerrickID

AF PI Points

- Tank Vapor Pressure Low Limit
- Tank Vapor Pressure Hi Limit
- Tank Vapor Pressure
- Transmitter Failure Alarm

Index Fields

- ()

Database Table Name

SHAPETABLE_15

SQL Statement

```
Insert into SHAPETABLE_15 SIND:'SCADA Tag Prefix'  
STIME:PRODUCTION_TIME SCOLPOINT:'Tank Vapor Pressure Low  
Limit' SCOLPOINT:'Tank Vapor Pressure Hi Limit' SCOLPOINT:'Tank  
Vapor Pressure' SCOLPOINT:'Transmitter Failure Alarm'  
SUSEDATATYPE
```

Buttons: Cancel, Update Timer

Once you have selected/filled in all fields needed for matching a database table schema, click "Update Timer"

The SQL Insert Syntax Description

```
Insert into <Table Name>  
$IND:'<Pi Attribute>' [AS '<TargetName>']  
[$TIME:'<TargetName>']  
$COLPOINT:'<Pi Attribute>' [AS '<TargetName>']  
$COLPOINT:'<Pi Attribute>' [AS '<TargetName>']  
[...]  
$ATTR:'<Attribute>' [AS '<TargetName>']  
[...]  
$USEDATATYPE
```

- <Table Name> - Is the Databricks Table name. (The warehouse and database is configured in the other segments with the Config Editor)
- <[Pi] Attribute> - is the original name of AF Attribute. If no TargetName is specified this will be the Databricks column name
- <TargetName> - is the name for the Databricks column

- \$IND - Is Index column for the database table. It's an AF Attribute or Addon Field (see details) and it is what groups all other (pi) attributes together (i.e. Element's template, parent, categories, etc.)
- \$COLPOINT – Is an AF PI Attribute that has a valid PI Point associated with it and will be turned into a time series column
- \$ATTR – Is a AF Attribute (static, formula, etc) and will be turned into a vertical column
- \$USEDATATYPE – Optional. If set, the AF datatypes will be used. If not set, all columns will be of data type “STRING”

Configure AddOn to use AF Element Name as Attribute



1) Create AddOnValue:

```
{<Your Attribute Name>: "DEFParentName"}
```

Example:

```
{"Equipment_Name": "DEFParentName"}
```

2) Use AddOnValue as column in the Insert Statement:

```
INSERT ... $COL3:Equipment_Name
```

Using Addon Field as Index Field (What groups all other (pi) attributes together)

Timer Info

Group Name	PIAF Cooling Databricks 13	Description	PIAF Hierarchy to TimeStream
Method	GetMostRecentVals	Query	{ParentTemplate: Cooling Fan} Edit
Kind	Snapshot@digitalstatus	Time Span	00:02:00
From DateTime	-2m	To DateTime	#now-00:05:00
<input type="checkbox"/> True or False		<input checked="" type="checkbox"/> True or False 2	
Type	API	Info 0	https://dbc-795c2d82-399f.cloud.databricks.com/api/2.0/jp/v1
Info 1	{User-Agent:'BOT_TWIN_TALK','X-Databricks-Authori} Edit	Info 2	databricks
Delayed Start (ms)	#abs 00:01:00	Interval (ms)	120000 Edit
Add-on Fields	{hierarchy_path:'\$DEFChildPath'} Edit		
Filters			Edit

Buttons: Load Current, Load Original, Graph Manager, Clear, Cancel, Save

Config Editor

Buttons: Insert DB Statement, Import Integrator View

tt_statement

```
Insert into SHAPETABLE_15 $IND:'AFName'  
$TIME:'PRODUCTION_TIME' $COLPOINT:'Tank Vapor  
Pressure' $COLPOINT:'Tank Vapor Pressure Hi Limit'  
$COLPOINT:'Tank Vapor Pressure Low Limit'  
$COLPOINT:'Transmitter Failure Alarm' $USEDATATYPE
```

Buttons: Update/Add, Delete, Cancel, Store in String

AF Metadata Enrichment in AddOn Fields as Index Field

AF Metadata can be included in the payload:

\$DEF<Selector>

selector: selects which AF information is used to be inserted.

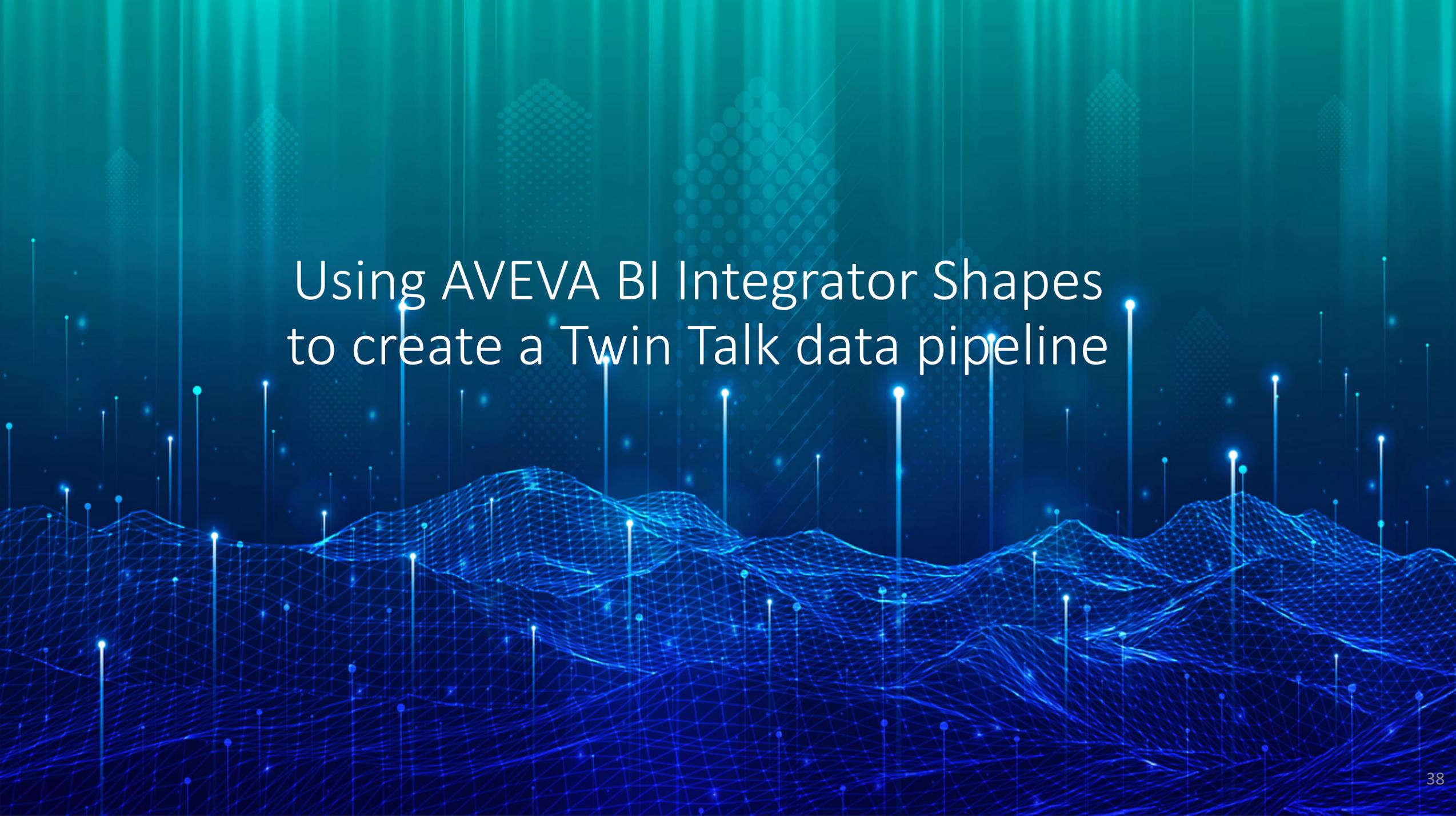
Examples AF Metadata Enrichment:

Equipment:\$DEFParentName

Parent element's name will be insert with ever record.

Selector Options:

- **Name** – AF Element or Attribute Name
- **Description** – AF Description
- **Template** – AF Template Name for Element or Attribute
- **Categories** – AF Categories for Element or Attribute
- **TagName** – PI TagName
- **ChildPath** – AF Path to Element or Attribute
- **ParentPath** – AF Path to Parent Element
- **ParentName** – AF Parent Element Name
- **ParentTemplate** – AF Template Name of Parent Element
- **ParentDescription** – AF Template Name of Parent Element
- **SrcSysName** – AF System Name
- **SrcDBName** – AF Database Name
- **DataType** – Data Type of Attribute



Using AVEVA BI Integrator Shapes to create a Twin Talk data pipeline

Databricks Table Setup



Make sure to create a Databricks table that matches the timer and vice versa. To get started set up this table.

Catalog Explorer > hive_metastore > default >

 **realtime_historian_with_hierarchy** 

Overview Sample Data Details Permissions History Lineage Insights Quality

Description

Column	Type	Comment
measure_name	string	
hierarchy_path	string	
date_time	timestamp	
measure_value	double	



4 Steps to Use PI Integrator Shapes Using TwinTalk



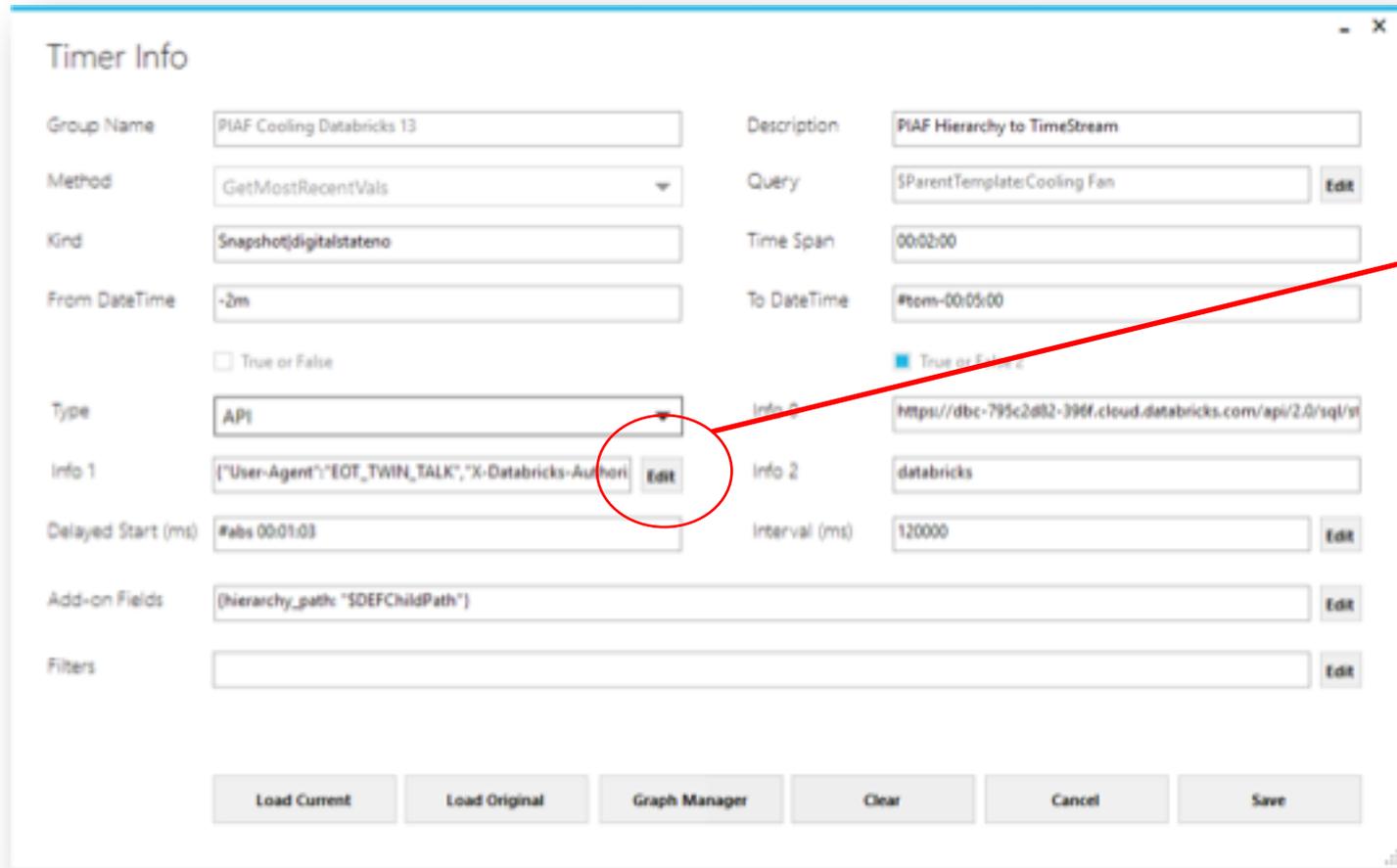
The image illustrates a four-step process for using PI Integrator Shapes with TwinTalk. The steps are as follows:

- Config Editor:** The 'Import Integrator View' button is highlighted with a red circle. The 'Accept' section contains a list of attributes: `$statement`, `$wait_timeout`, `$warehouse_id`, `$clientid`, and `$clientsecret`. The `$statement` attribute is circled in red.
- Open a PI Integrator View/Shape Definition:** A file explorer window showing a folder named 'Bakken' containing three files: `Bakken_TankVapor_Generic.json`, `Bakken_WELL_ESP.json`, and `Bakken_WELL_Generic.json`. A red arrow points from this window to the 'Import Integrator View' button in the Config Editor.
- Timer Info:** A window showing configuration for a timer. The 'Type' is set to 'API'. The 'URL' is `http://10.10.10.10:8080/af/shape/SHAPETABLE_15`. A red arrow points from this window to the 'Update Timer' button in the Create AF Insert Statement window.
- Create AF Insert Statement:** A window for defining the insert statement. The 'AF Attributes' section includes `EQUIPMENT_NAME`, `PRODUCTION_TIME`, `EQUIPID`, `Parent Tag Prefix`, and `SCADA Tag Prefix`. The 'AF PI Points' section includes `Tank Vapor Pressure`, `Tank Vapor Pressure Hi Limit`, `Tank Vapor Pressure Low Limit`, and `Transmitter Failure Alarm`. The 'Database Table Name' is `SHAPETABLE_15`. The 'SQL Statement' field contains: `Insert into <Table Name> $IND:'SCADA Tag Prefix' $TIME:'PRODUCTION_TIME' $COLPOINT:'Tank Vapor Pressure' $COLPOINT:'Tank Vapor Pressure Hi Limit' $COLPOINT:'Tank Vapor Pressure Low Limit' $COLPOINT:'Transmitter Failure Alarm' $USEDATATYPE`. The 'Update Timer' button is circled in red.

4 Steps to Run PI Integrator Shapes Using TwinTalk

1

1) Open preconfigured Timer



Timer Info

Group Name: PIAF Cooling Databricks 13

Description: PIAF Hierarchy to TimeStream

Method: GetMostRecentVals

Kind: Snapshot|digitalstateno

From DateTime: -2m

To DateTime: #tom-00:05:00

Type: API

Info 1: {"User-Agent": "EOT_TWING_TALK", "X-Databricks-Authorization-Token-Type": ""} **Edit**

Info 2: https://dbc-795c2d82-396f.cloud.databricks.com/api/2.0/sql/v1

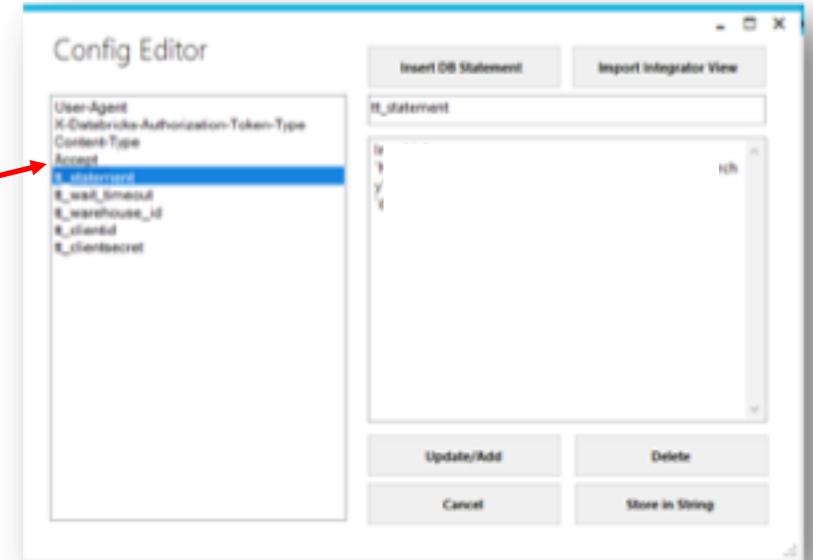
Delayed Start (ms): #abs 00:01:03

Interval (ms): 120000 **Edit**

Add-on Fields: (hierarchy_path: "SDEFChildPath") **Edit**

Filters: **Edit**

Buttons: Load Current, Load Original, Graph Manager, Clear, Cancel, Save



Config Editor

JSON Statement

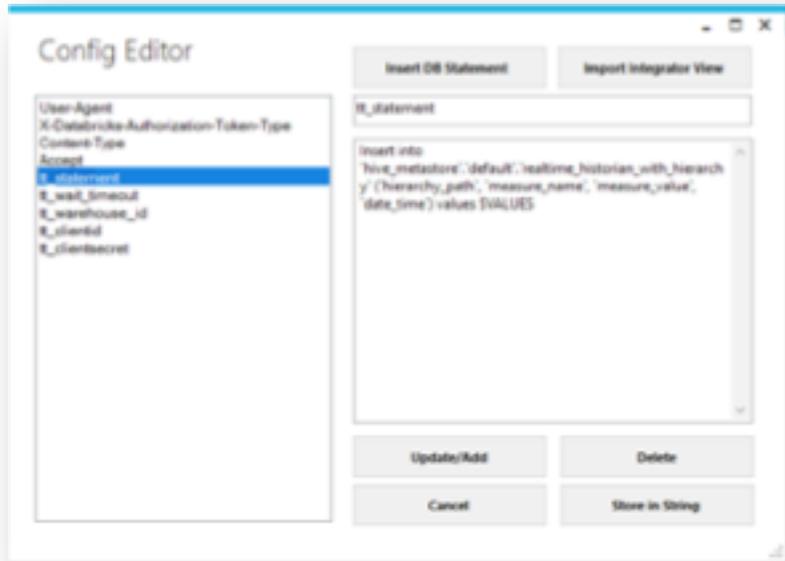
JSON Statement

Buttons: Update/Add, Delete, Cancel, Store in String

Opens JSON Config Editor

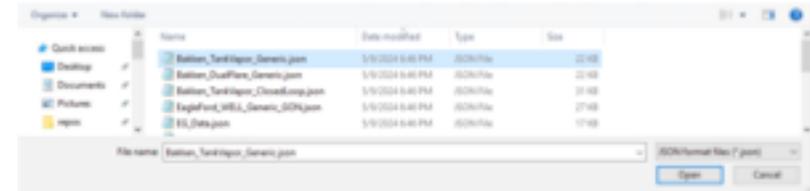
4 Steps to Run PI Integrator Shapes Using TwinTalk

2



2) If new environment, configure Databricks Parameters:

3



3) Import Integrator Shape (JSON)

4 Steps to Run PI Integrator Shapes Using TwinTalk

4

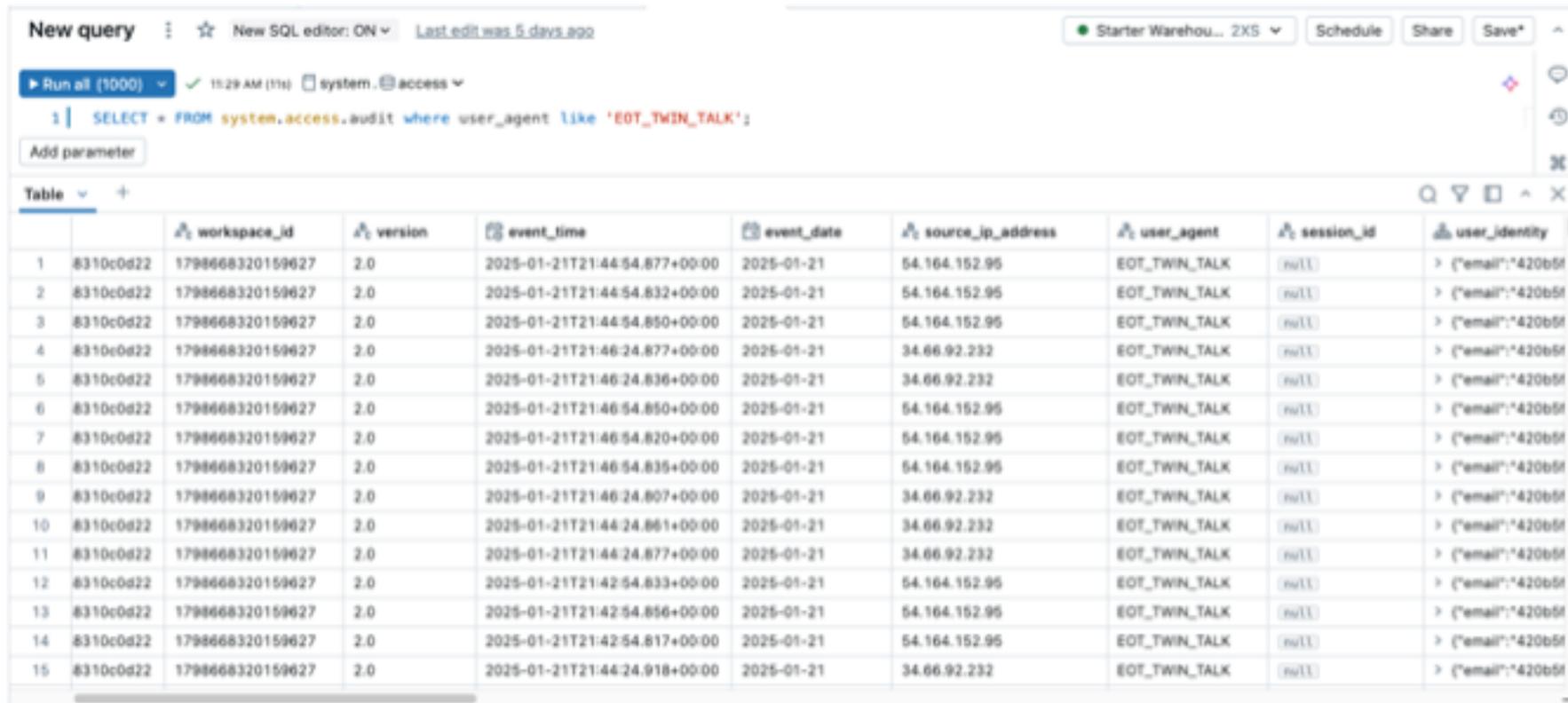
Get's populated by the importing of AVEVA PI Integrator Shape Definition. Must also be available in Twin Talk in the AF Queried AF tree

- 4) Define the Insert for table schema
 - Pick Index Field "SCADA_PREFIX"
 - Pick Timestamp Field "PRODUCTION_TIME"
 - Enter database table name "SHAPETABLE_15"

Check Audit Logs

Ensure that Everything Works Correctly

```
SELECT * FROM system.access.audit where user_agent like 'EOT_TWIN_TALK';
```



The screenshot shows a SQL query editor interface. At the top, it says "New query" and "New SQL editor: ON". The query is: `SELECT * FROM system.access.audit where user_agent like 'EOT_TWIN_TALK';`. Below the query, there is a "Run all (1000)" button and a status bar indicating "11:29 AM (11s) system.access". The results are displayed in a table with 15 rows and 10 columns. The columns are: workspace_id, version, event_time, event_date, source_ip_address, user_agent, session_id, and user_identity. All user_agent values are "EOT_TWIN_TALK".

		workspace_id	version	event_time	event_date	source_ip_address	user_agent	session_id	user_identity
1	8310c0d22	1798668320159627	2.0	2025-01-21T21:44:54.877+00:00	2025-01-21	54.164.152.95	EOT_TWIN_TALK	null	> ("email": "420b5f
2	8310c0d22	1798668320159627	2.0	2025-01-21T21:44:54.832+00:00	2025-01-21	54.164.152.95	EOT_TWIN_TALK	null	> ("email": "420b5f
3	8310c0d22	1798668320159627	2.0	2025-01-21T21:44:54.850+00:00	2025-01-21	54.164.152.95	EOT_TWIN_TALK	null	> ("email": "420b5f
4	8310c0d22	1798668320159627	2.0	2025-01-21T21:46:24.877+00:00	2025-01-21	34.66.92.232	EOT_TWIN_TALK	null	> ("email": "420b5f
5	8310c0d22	1798668320159627	2.0	2025-01-21T21:46:24.836+00:00	2025-01-21	34.66.92.232	EOT_TWIN_TALK	null	> ("email": "420b5f
6	8310c0d22	1798668320159627	2.0	2025-01-21T21:46:54.850+00:00	2025-01-21	54.164.152.95	EOT_TWIN_TALK	null	> ("email": "420b5f
7	8310c0d22	1798668320159627	2.0	2025-01-21T21:46:54.820+00:00	2025-01-21	54.164.152.95	EOT_TWIN_TALK	null	> ("email": "420b5f
8	8310c0d22	1798668320159627	2.0	2025-01-21T21:46:54.835+00:00	2025-01-21	54.164.152.95	EOT_TWIN_TALK	null	> ("email": "420b5f
9	8310c0d22	1798668320159627	2.0	2025-01-21T21:46:24.807+00:00	2025-01-21	34.66.92.232	EOT_TWIN_TALK	null	> ("email": "420b5f
10	8310c0d22	1798668320159627	2.0	2025-01-21T21:44:24.861+00:00	2025-01-21	34.66.92.232	EOT_TWIN_TALK	null	> ("email": "420b5f
11	8310c0d22	1798668320159627	2.0	2025-01-21T21:44:24.877+00:00	2025-01-21	34.66.92.232	EOT_TWIN_TALK	null	> ("email": "420b5f
12	8310c0d22	1798668320159627	2.0	2025-01-21T21:42:54.833+00:00	2025-01-21	54.164.152.95	EOT_TWIN_TALK	null	> ("email": "420b5f
13	8310c0d22	1798668320159627	2.0	2025-01-21T21:42:54.856+00:00	2025-01-21	54.164.152.95	EOT_TWIN_TALK	null	> ("email": "420b5f
14	8310c0d22	1798668320159627	2.0	2025-01-21T21:42:54.817+00:00	2025-01-21	54.164.152.95	EOT_TWIN_TALK	null	> ("email": "420b5f
15	8310c0d22	1798668320159627	2.0	2025-01-21T21:44:24.918+00:00	2025-01-21	34.66.92.232	EOT_TWIN_TALK	null	> ("email": "420b5f

Best Practices & Troubleshooting

Best Practices

- Most effective process: Start with the end (use case) in mind and work backwards
 1. Understand in which format, conversion and aggregation the consumer (user or app) expects the data to be in. Schema, aggregation, etc.
 2. Create the Databricks Table Schema first
 3. Configure the Twin Talk SQL Insert statement to match the table
- Create 1 Databricks Table and Twin Talk data pipeline per use case

Common Errors and how to handle them

- Databricks Error: Authorization

- Reason 1 : Connection between Twin Talk and Databricks configured incorrectly.

- Action: Make sure to test connection using POSTMAN (without even using TwinTalk) and test is successful. Otherwise re-do the steps to create authentication.

- Reason 2 : No network connection between Twin Talk and Databricks.

- Action: Use POSTMAN/ping/telnet/etc (without using Twin Talk) to ensure that the Databricks API POST Request is going through.

Common Errors and how to handle them

- Databricks Error: Unprocessable entity
 - Reason: SQL Statement is configured incorrect. (Syntax error)
 - Action: Copy the insert statement from the Twin Talk log file into the Databricks SQL Editor and run it to get information on how to fix the Syntax Error. Apply the solution in the TT Insert SQL Creator

Certification Quiz

Q&A



Twin Talk – Databricks

Certification Training

