Contents

[Purpose 1](#_Toc1799268374)

[Data Architecture 1](#_Toc1044318823)

[Data Architecture – Logic App 2](#_Toc1884354809)

[Data Architecture – Bulk Copy Pipeline Triggered By Logic App 3](#_Toc1858256115)

[Databricks – Raw to Stage and Stage to Presentation 4](#_Toc372146918)

[Data Flow Diagram 5](#_Toc1487735653)

[Power Bi 6](#_Toc1160738006)

[Power Bi – Tabs 6](#_Toc1594564918)

[Power Bi - Custom Dax Tables 6](#_Toc499150441)

[Power Bi - Custom Columns 7](#_Toc1552612213)

[Power Bi – Troubleshooting 7](#_Toc384536424)

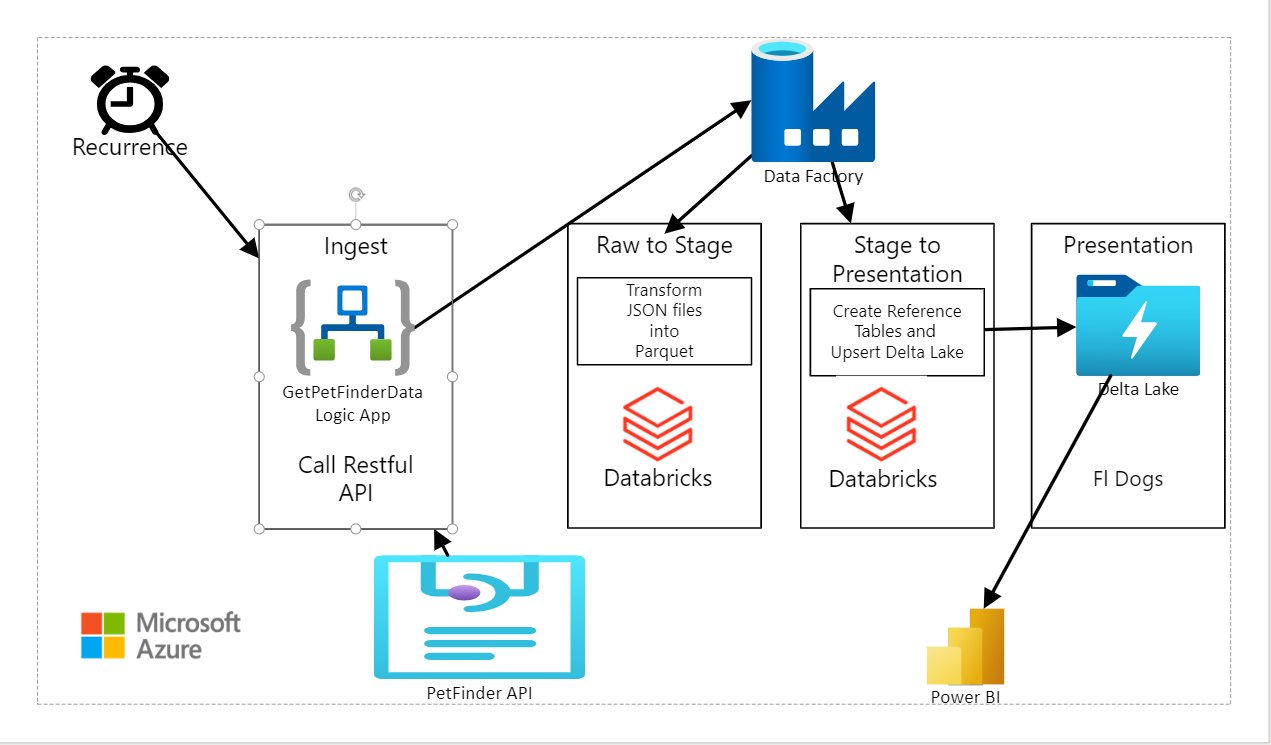
# Purpose

Showcase data sourced from a Delta Lake within Databricks, featuring custom DAX tables for top values, visuals with drill down and slicing functionality, bookmarks for navigation, a glossary and accompanying documentation.

Typically, custom DAX tables are avoided as much as possible by performing all transformations in a pipeline / SSIS package / miscellaneous data transformation but it was necessary to create the ***LongestWaiting*** Dax Table as some functions aren’t allowed on fields that come from Direct Query data.

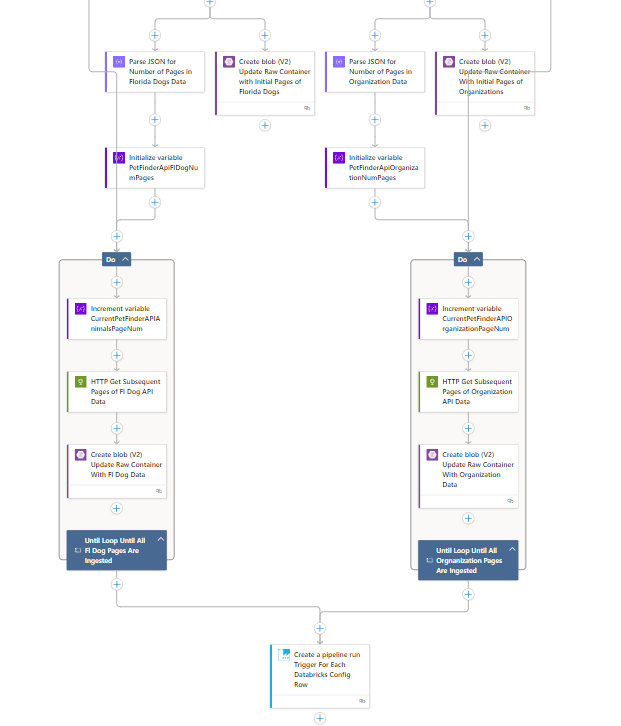
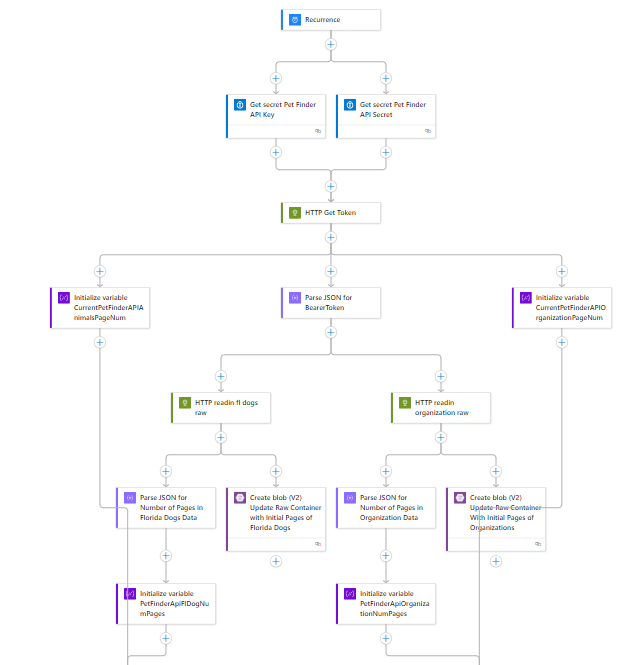
# Data Architecture

The data used to [Upsert the Delta Lake](https://learn.microsoft.com/en-us/azure/databricks/delta/merge) is the free Petfinder API (Application Programming Interface). Data Architecture featuring Logic Apps, Data Factory, Databricks and a Delta Lake are described in this diagram.



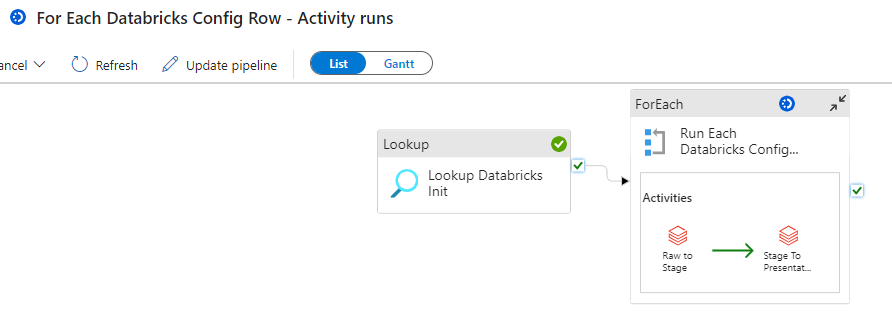
## Data Architecture – Logic App

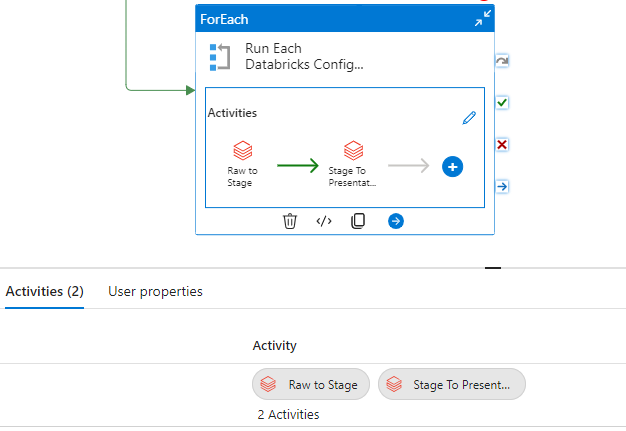
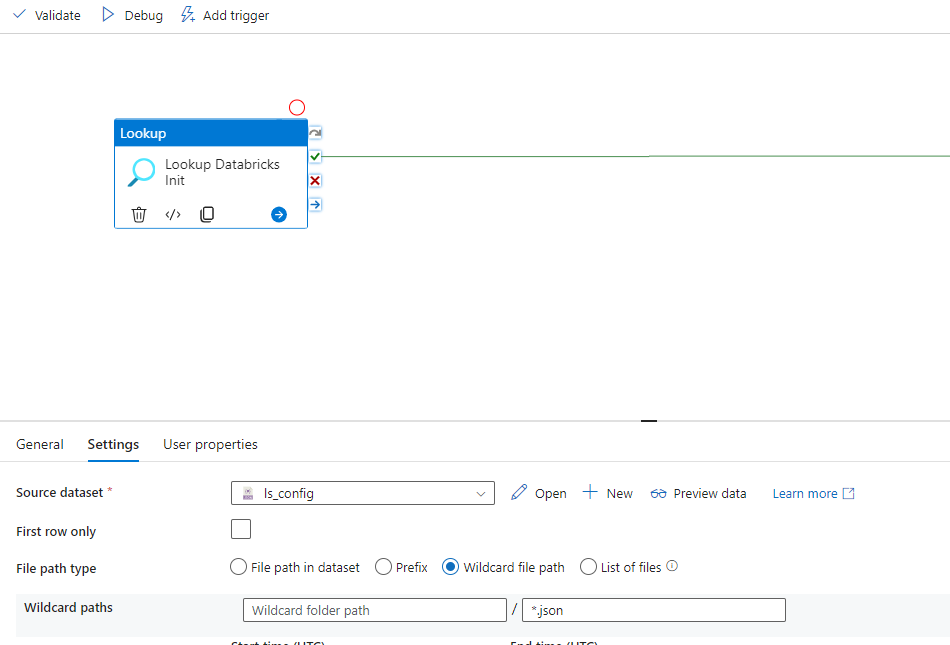
Every Sunday, a weekly Recurrence triggers a logic app that makes Restful API calls to Petfinder database and brings back thousands of dogs ready for adoption in Florida back [in .json files](https://www.petfinder.com/developers/v2/docs/#get-organizations).



## Data Architecture – Bulk Copy Pipeline Triggered By Logic App

The L[ogic App then triggers](https://www.cloudfronts.com/uncategorized/trigger-azure-pipeline-with-logic-app/) a [Bulk Copy Pipeline](https://learn.microsoft.com/en-us/azure/data-factory/tutorial-bulk-copy-portal) in Azure Data Factory that reads in Metadata from a .json file in blob storage and uses the Metadata to execute a Databricks Notebook for Raw to Stage and Stage to Presentation transformations.

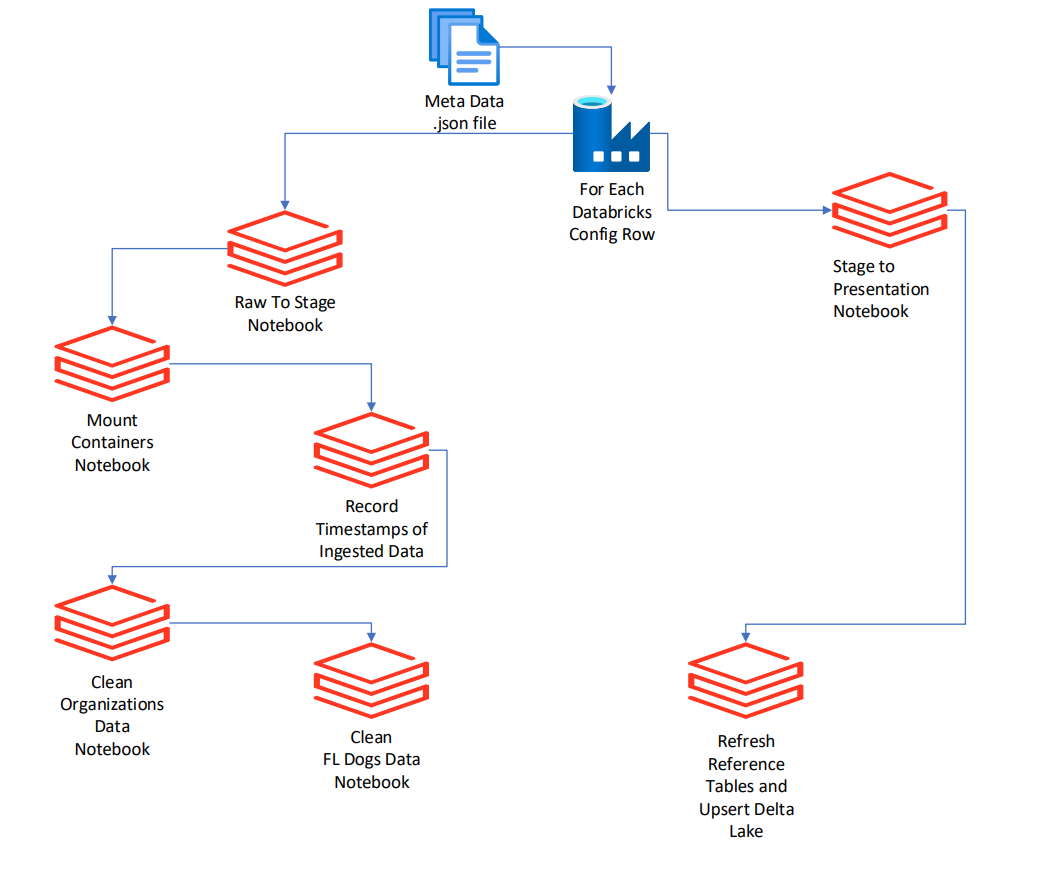




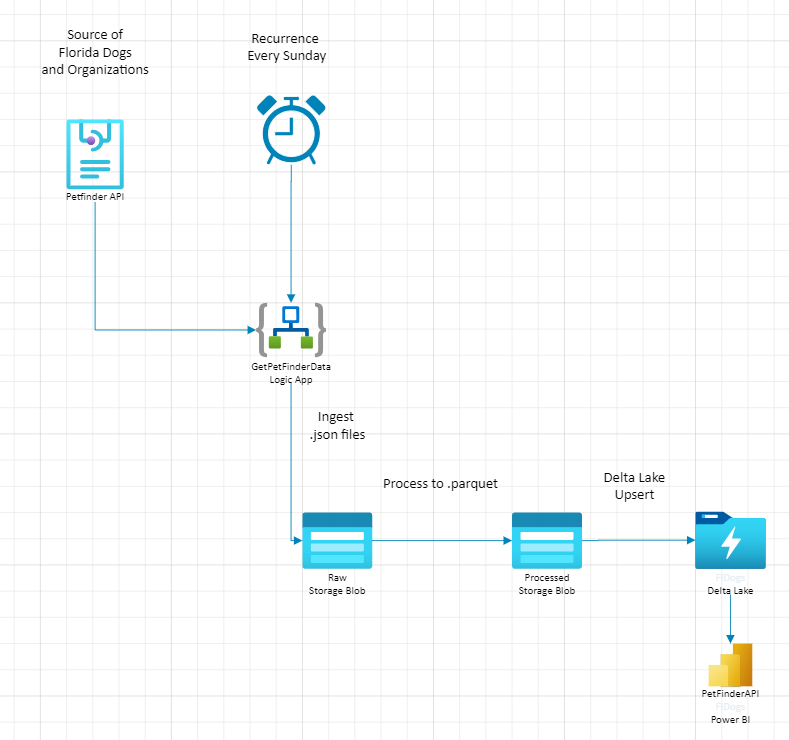
## Databricks – Raw to Stage and Stage to Presentation

The Raw to Stage notebook called from ADF kicks off a series of other Databricks notebooks that mount containers, records timestamps of .json files in the Raw container, and cleans up Organization and Fl Dogs data so that it is one record per unique key, places data in .parquet files within the Stage folder.

The Stage to Presentation notebooks kicks off a series of Databricks notebooks that refresh Hive tables for Reference (Organizations, Pictures, etc.) in the Presentation folder and Upserts a Delta Lake table containing Florida Dogs.

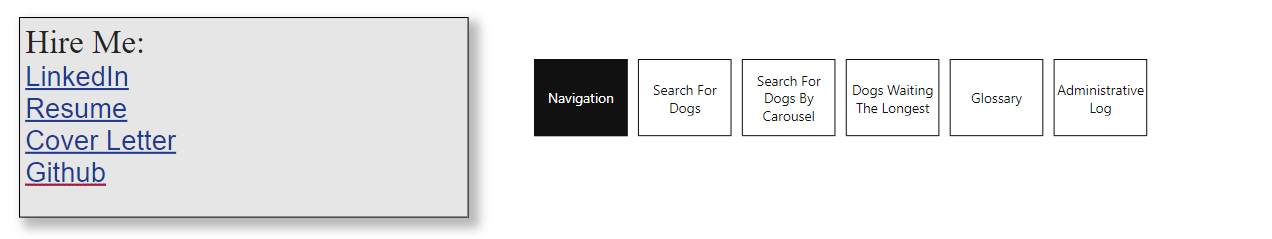


# Data Flow Diagram



# Power Bi

## Power Bi – Tabs



* Navigation – Page listing Bookmarks to all pages. A Bookmark to the Navigation page is on every Tab
* Search For Dogs – Search for Florida Dogs by Breed, Color, Behavior, Age, etc.
* Search For Dogs by Carousel – Select pictures from carousels to see information about adoptable Florida dogs.
* Glossary – Definitions of fields displayed with reports along with sources and upsert / refresh frequencies.
* Administrative Log – Display timestamps for ingestion, processing and presentation.

## Power Bi - Custom Dax Tables

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Table Name** | **Description** | **Code** |
| 1 | WaitingLongest | Dax table displaying Dogs in Florida with the top n time interval between Status Changed At and Current Date | WaitingLongest = summarize(fldogs, fldogs[AnimalsID], fldogs[AnimalName], fldogs[Description], fldogs[StatusChangedyyyy], fldogs[StatusChangedmm], fldogs[StatusChangeddd], fldogs[StatusChangedAtDisplay],fldogs[StatusChangedAt], fldogs[ActiveIndicator], fldogs[Age], fldogs[Address1], fldogs[Address2], fldogs[City], fldogs[Postcode], fldogs[Coat], fldogs[Declawed], fldogs[GoodWithCatsDisplay], fldogs[GoodWithDogsDisplay], fldogs[GoodWithChildrenDisplay], fldogs[Housetrained], fldogs[Mixed], fldogs[PrimaryBreed], fldogs[PrimaryColors] , fldogs[SecondaryBreed], fldogs[SecondaryColors], fldogs[ShotsCurrent], fldogs[Size], fldogs[SpayedNeutered], fldogs[SpecialNeeds], fldogs[TertiaryColors], fldogs[URL] , fldogs[OrganizationId]) |

## Power Bi - Custom Columns

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Column Name** | **Description** | **Calculation** | **Table** |
| 1 | GoodWithCatsDisplay | Display value for Good With Cats | GoodWithCatsDisplay = if(fldogs[GoodWithCats], "Yes", "No") | fldogs |
| 2 | GoodWithChildrenDisplay | Display value for Good With Children | GoodWithChildrenDisplay = if(fldogs[GoodWithChildren], "Yes", "No") | fldogs |
| 3 | GoodWithDogsDisplay | Display value for Good With Dogs | GoodWithDogsDisplay = if(fldogs[GoodWithDogs], "Yes", "No") | fldogs |
| 4 | StatusChangedAtDate | Timestamp of Status Change | StatusChangedAtDate = date(WaitingLongest[StatusChangedyyyy], WaitingLongest[StatusChangedmm], WaitingLongest[StatusChangeddd]) | WaitingLongest |
| 5 | StatusChangedddd | Day of Status Changed in DDD Format | StatusChangeddd = mid( fldogs[StatusChangedAt], 9, 2) | fldogs |
| 6 | StatusChangedmm | Month of Status Changed in MM Format | StatusChangedmm = mid( fldogs[StatusChangedAt], 6, 2) | fldogs |
| 7 | StatusChangedyyyy | Year of Status Changed in YYYY Format | StatusChangedyyyy = mid( fldogs[StatusChangedAt], 1, 4) | fldogs |

## Power Bi – Troubleshooting

|  |  |  |
| --- | --- | --- |
| **#** | **Description** | **Troubleshooting** |
| 1 | Report takes 5 - 10 minutes to render | This report takes a little longer because it's accessing Hive tables on my Azure Subscription. |
| 2 | Report Visual shows error "Visual has exceeded the avaiable resources" | Re-boot your machine and try accessing the report again. |

