POWER BI Connector

Pre-Certification Testing details

Abstract

The document provides the details on connector files, installation steps and details tested.

Contents

[**Pre-Requisite** 2](#_Toc43996214)

[**Installation Steps** 2](#_Toc43996215)

[**Power BI Runtime Setup (PowerBI Desktop):** 3](#_Toc43996216)

[**Tested Use cases/Criteria’s** 8](#_Toc43996217)

[**Test Data** 9](#_Toc43996218)

[**External References** 9](#_Toc43996219)

[**Results/Next Steps** 10](#_Toc43996220)

**Objective**

The objective of this project is to test the Custom data Connector built by Actian team with various versions of PowerBI products primarily for PowerBI Desktop and PowerBI Service.

# **Pre-Requisite**

* Install Visual Studio and Power Query SDK.
* A new Custom Data Connector should be project created in Visual Studio using the Data Connector SDK.
* Verify that Custom Data Connector SDK having basic details included as:

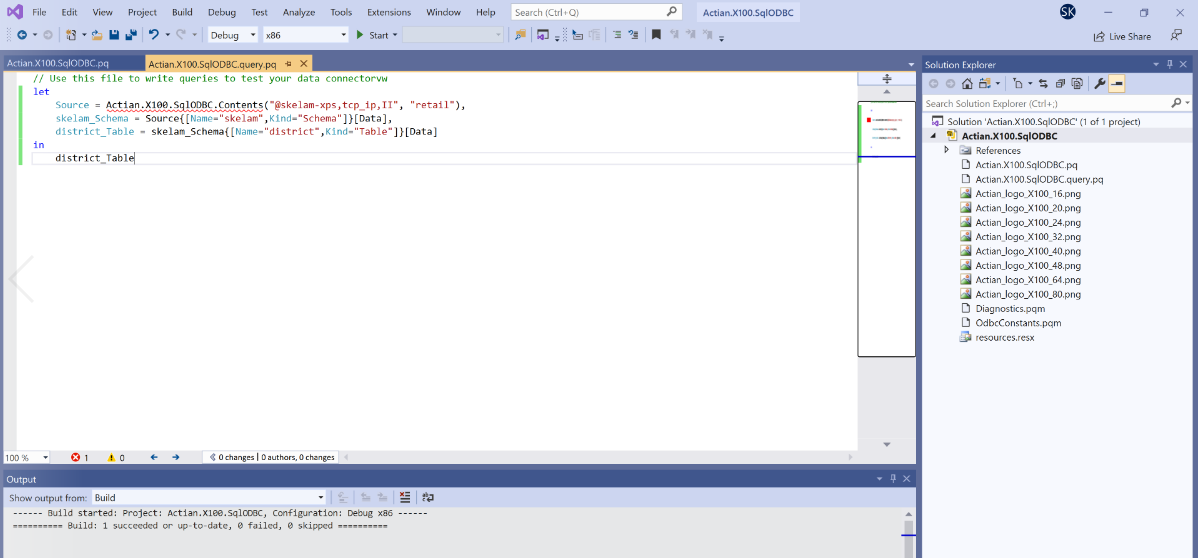
1. Actian.X100.SqlODBC. pq (This file contains the Power BI Data Connector logic for Actian Vector and Ingres databases)
2. Actian.X100.SqlODBC. query. pq (Use this file to write queries to test your data connector)

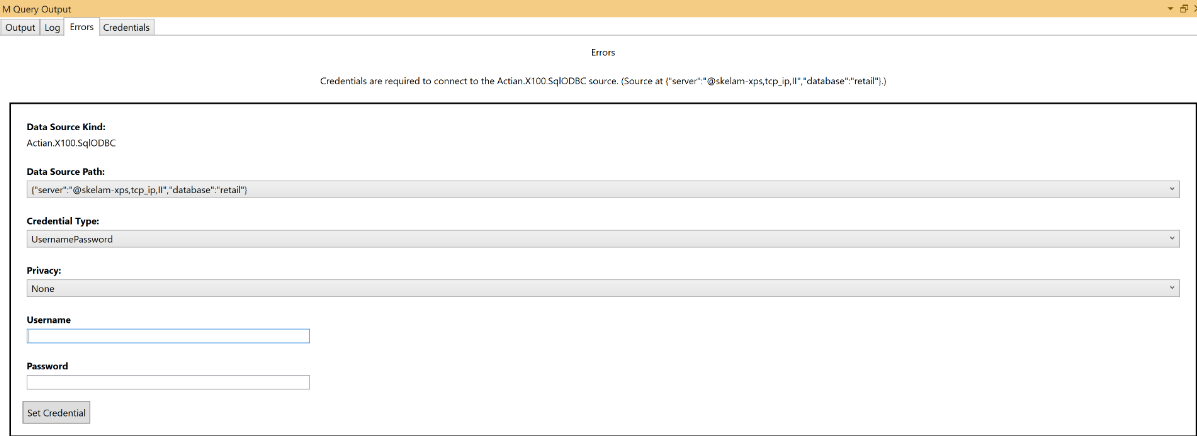
* PowerBI Pro License is required for testing the connector with PowerBI Service, to enable more features.
* Install PowerBI Gateway On-premise (Recommended mode)

# **Installation Steps**

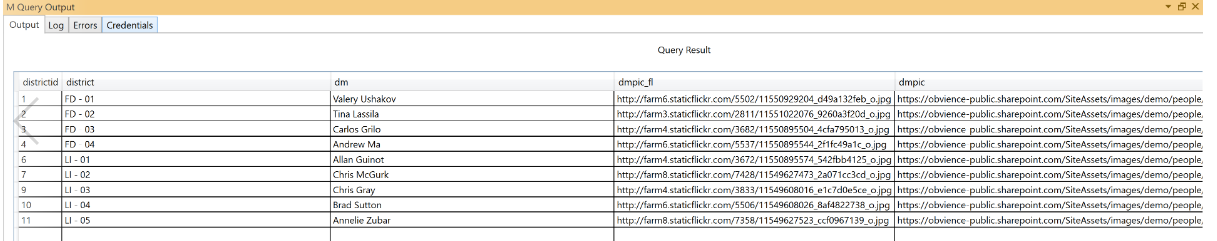
**PowerBI Build Setup (Visual Studio):**

The connector is written in Power Query M language and can be maintained as a solution in Microsoft Visual Studio. See <https://docs.microsoft.com/en-us/power-query/installingsdk>.

1. Download the Power Query SDK from the Visual Studio Marketplace.
2. Stop Visual Studio if running.
3. Install the Power Query SDK into Visual Studio.
4. Download the Ingres/Vector Power BI connector (all files and directories) from git repo.
5. Startup Visual Studio and open the connector solution, which is located in the root directory of the downloaded connector. [project].sln file
6. Use Visual Studio to edit and debug the connector.
   * The source files are in the first sub-directory, also named [project].
     + [project].mproj - Visual Studio project file for the connector
     + [project].pq - connector
     + [project].query.pq - tests for the connector
     + [project]##.png - different sized icons for Power BI UI
     + Diagnostics.pqm - tracing/diagnostics program
     + OdbcConstants.pqm - ODBC constants
     + resources.resx - resource file for Power BI UI
   * The logic for the connector is in [project].pq.
   * Build the solution.
7. Test the solution in Visual Studio with Debug->Start Debugging or Start without debugging the [project].query.pq file.
   * No difference seen between running the query file with or without debugging. It always runs the query file and displays the results.
   * There is an initial prompt to specify the type of user authentication (credentials) to use. Select either Windows or Username/Password. This can be saved (Set Credentials) so that this prompt is not reissued on subsequent tests. Either option works.
   * 



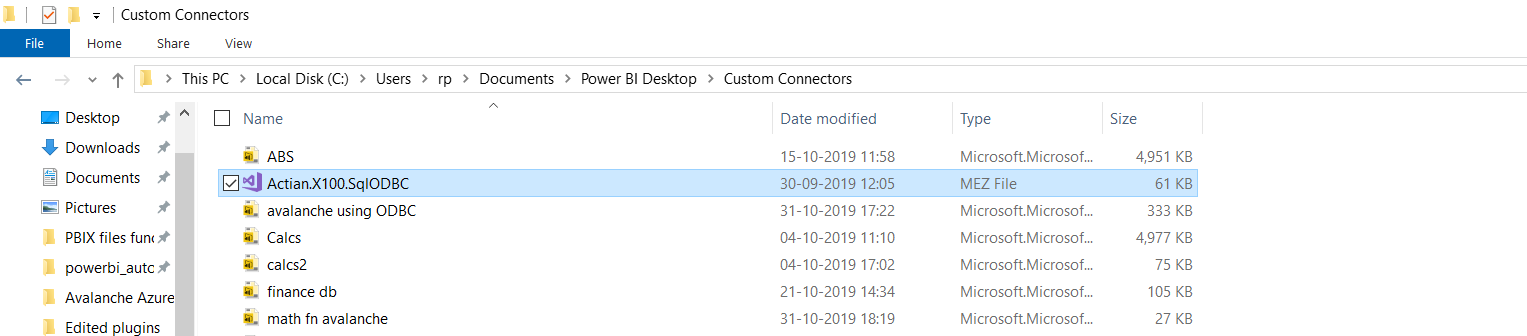
1. Validate the results.



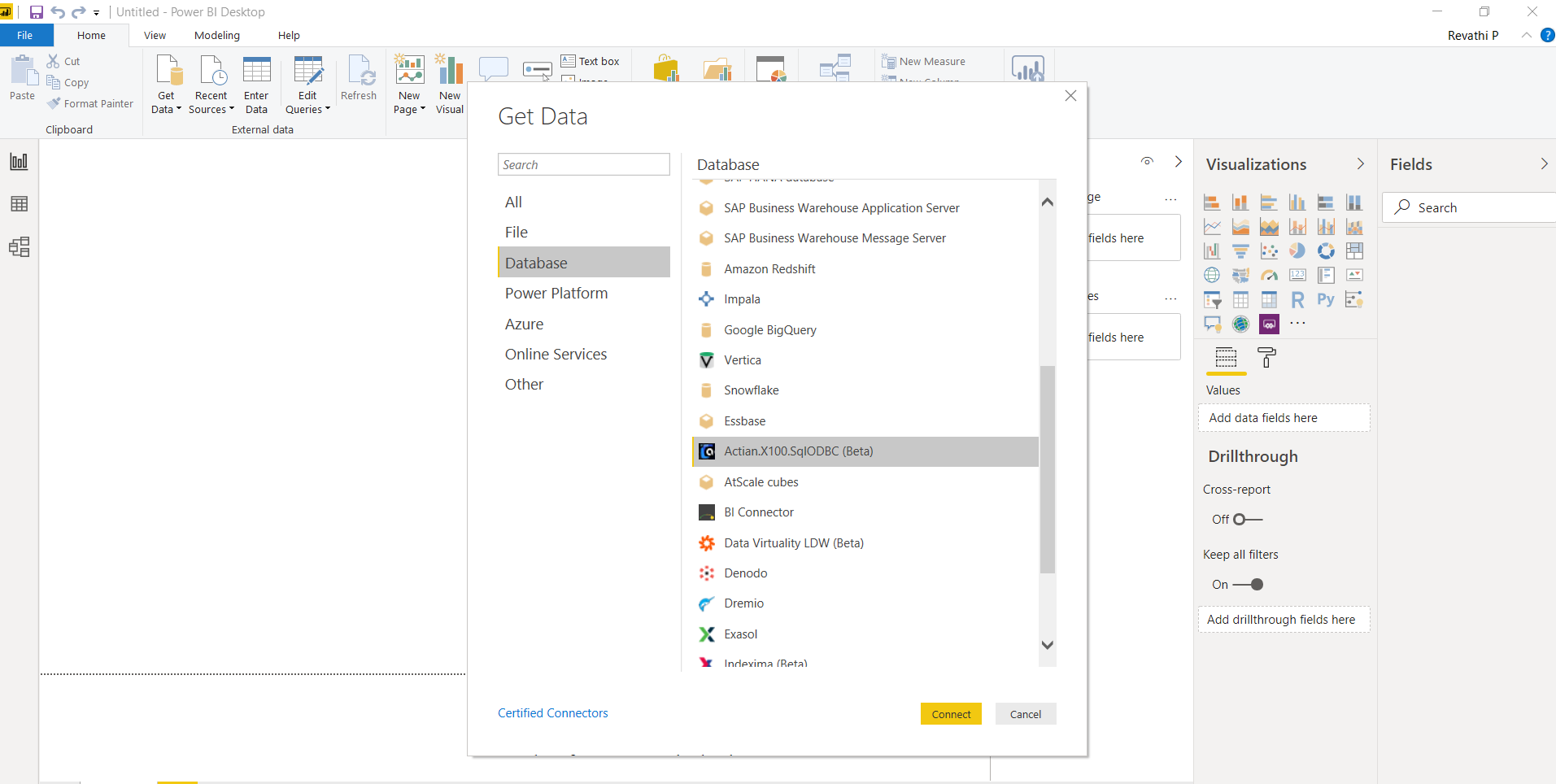
## **Power BI Runtime Setup (PowerBI Desktop):**

1. Install Microsoft Power BI Desktop on Windows. Currently, it is free.
2. Locate the connector executable in this repo at [project]/bin/Debug (or /Release). The executable is a single file named Actian.X100.SqlODBC.mez. It is actually a zip file containing the necessary files.
3. Copy the connector file to [Documents]\Power BI Desktop\Custom Connectors\. [Documents] may have different meaning on different versions of Windows.
   * Windows 10: This PC -> Documents, which is a link to c:\Users\<userid>\Documents\
   * Windows 7: c:\Users\<userid>\My Documents, which is a link to "c:\Users<userid>\Documents"

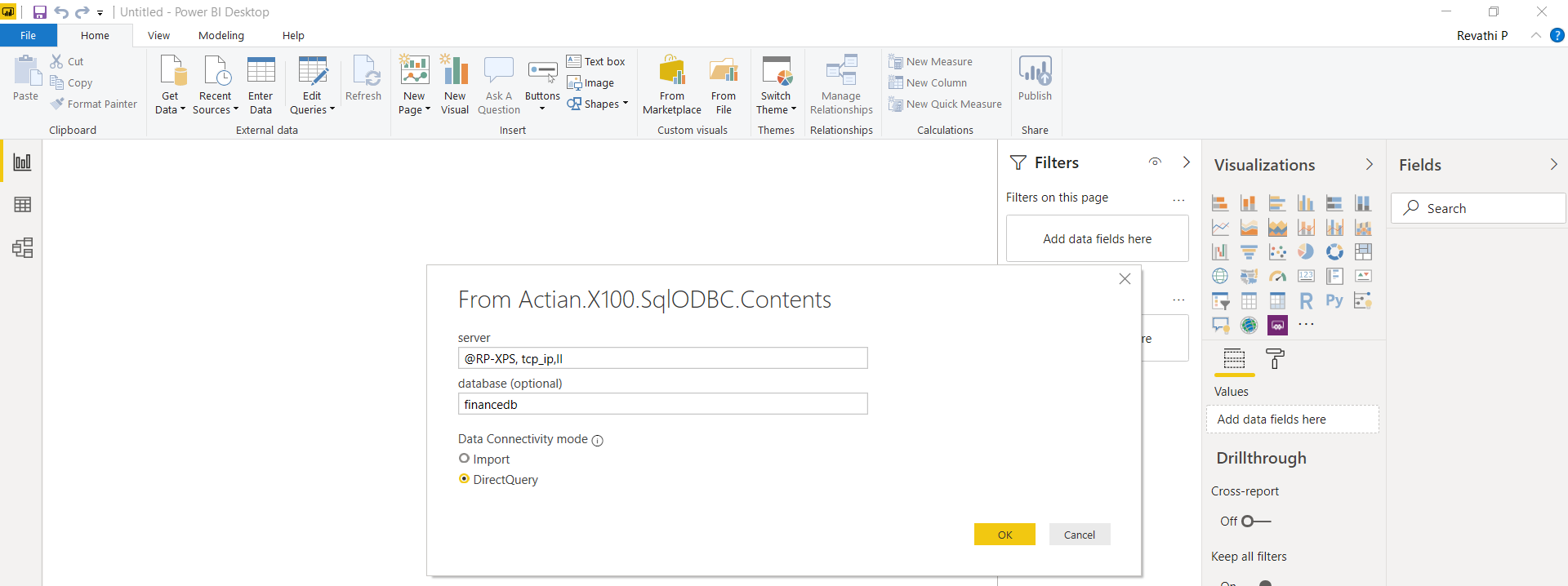
If the Power BI Desktop\Custom Connectors\ sub-directory doesn't exist under [Documents], create it. NOTE that some of the Microsoft doc refers to "Microsoft Power BI Desktop" as the higherlevel directory name rather than just "Power BI Desktop". The latter (without prefix Microsoft) worked on Windows 10.



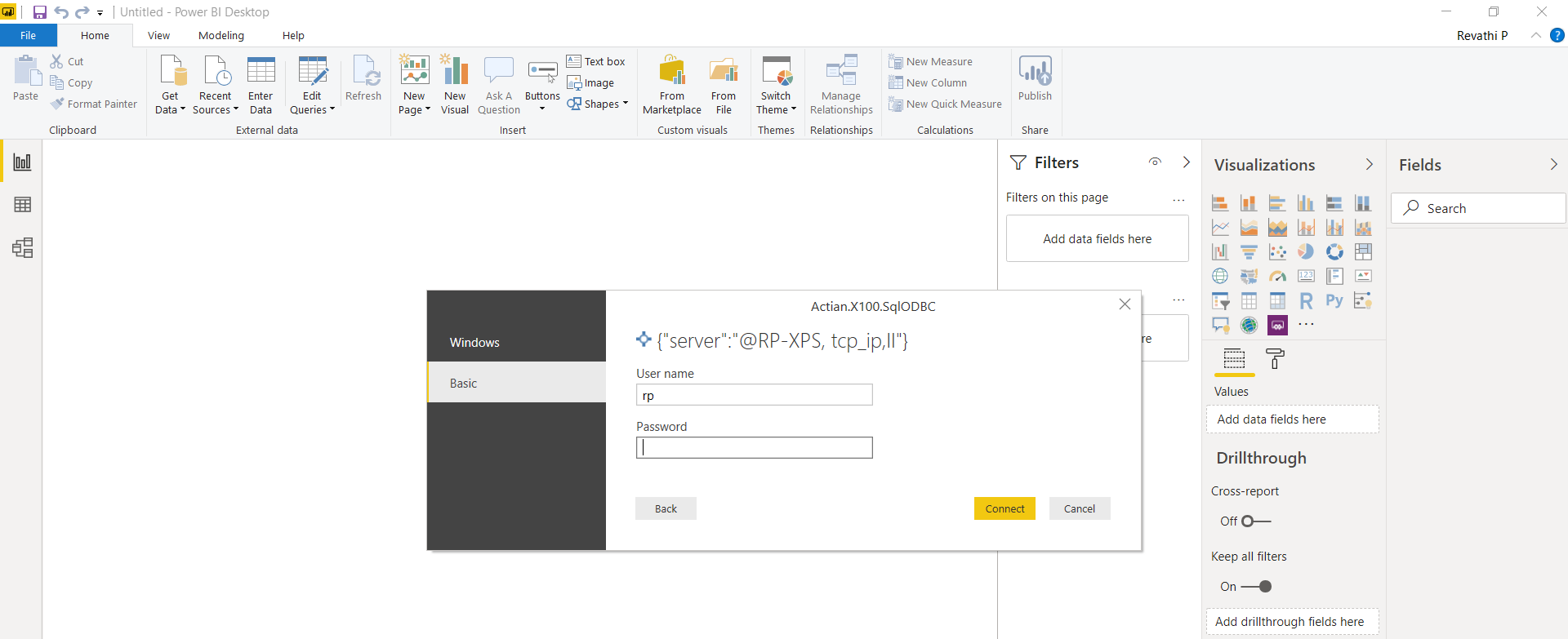
1. Open Power BI Desktop
2. Note: To load the connector, until it is certified, the security level for extensions in Power BI Desktop must be lowered to enable loading of unsigned/uncertified connectors.
   * Go to File | Options and settings | Options
   * Go the Security tab
   * Under Data Extensions, select Allow any extension to load without warning or validation
   * Restart Power BI Desktop
3. (In Power BI) select Get Data->More->Database->Actian.X100.SqlODBC (Beta) and press Connect. Continue past warning about the connector being under development.



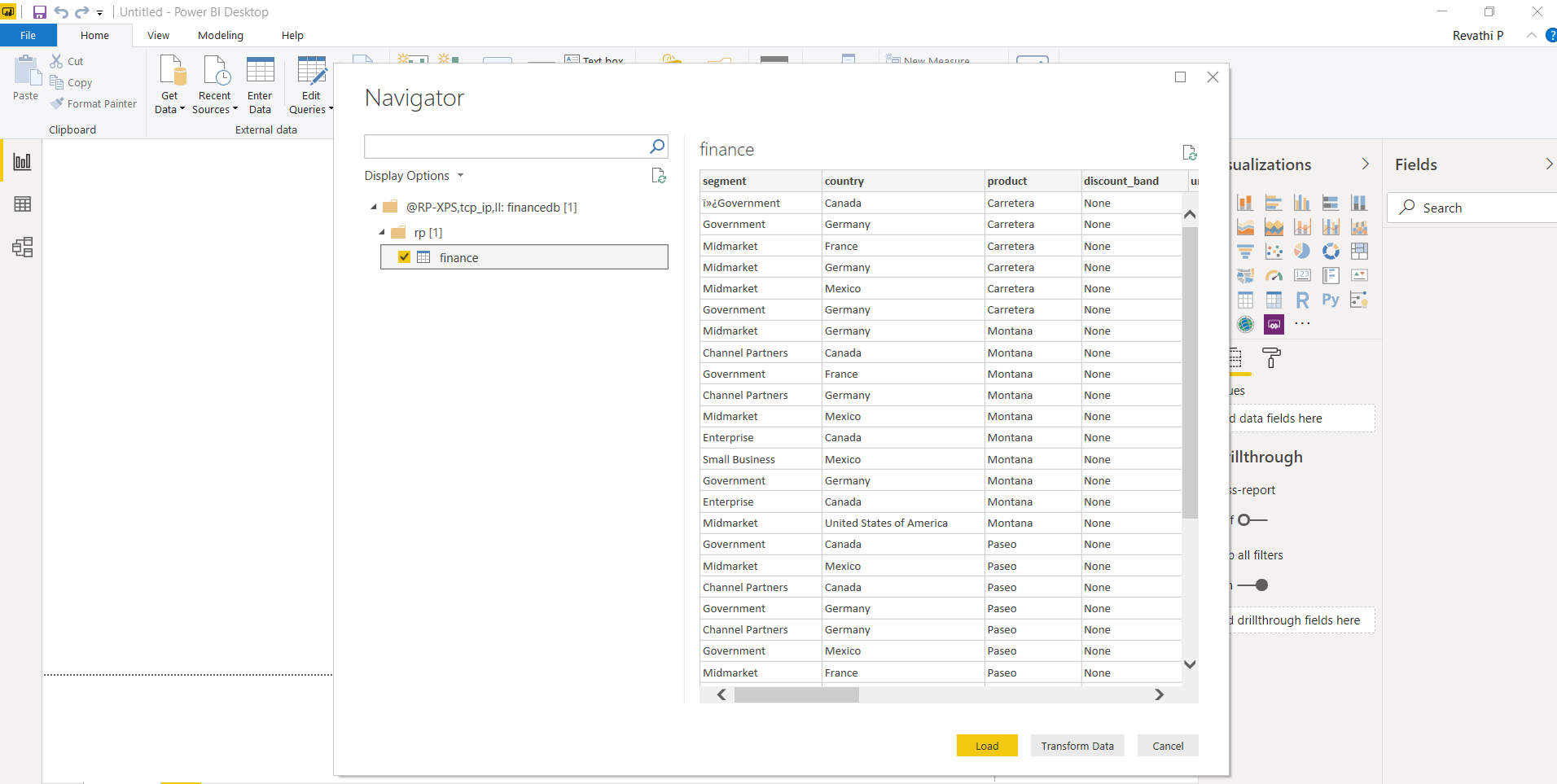
1. Enter connection information on Actian.X100.SqlODBC.Contents screen:
   * server: enter vnode for connection to remote Ingres/Vector installation, or (LOCAL) for local connection.
     + vnode can be a pre-defined vnode name or a dynamic one with the syntax @hostname,,II[user,password] where II is the installation id of the remote Ingres/Vector server (typically, II for Ingres, VW for Vector, VH for Vector Hadoop).
   * database: name of database
   * Data Connectivity mode
     + import - will download tables to Power BI (not recommended)
     + DirectQuery - access remote server directly (recommended)



1. Enter credentials (if not previously set) Windows or Username/Password: either option OK; set to keep permanent.



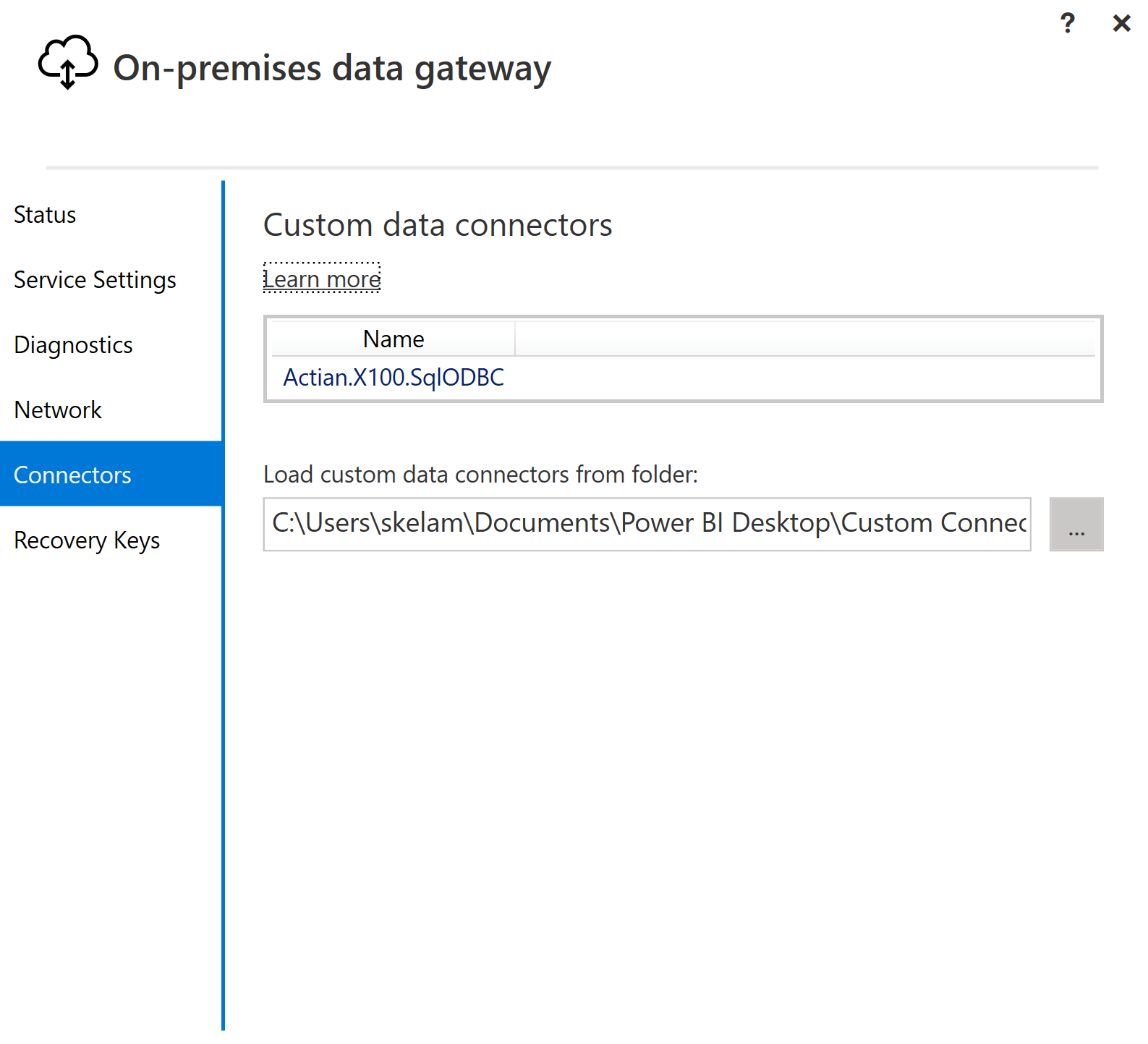
1. Connect should display "Navigator" window with list of schemas/tables. Proceed to use Power BI functionality on the tables.



**PowerBI Service:**

**Gateway Setup:**

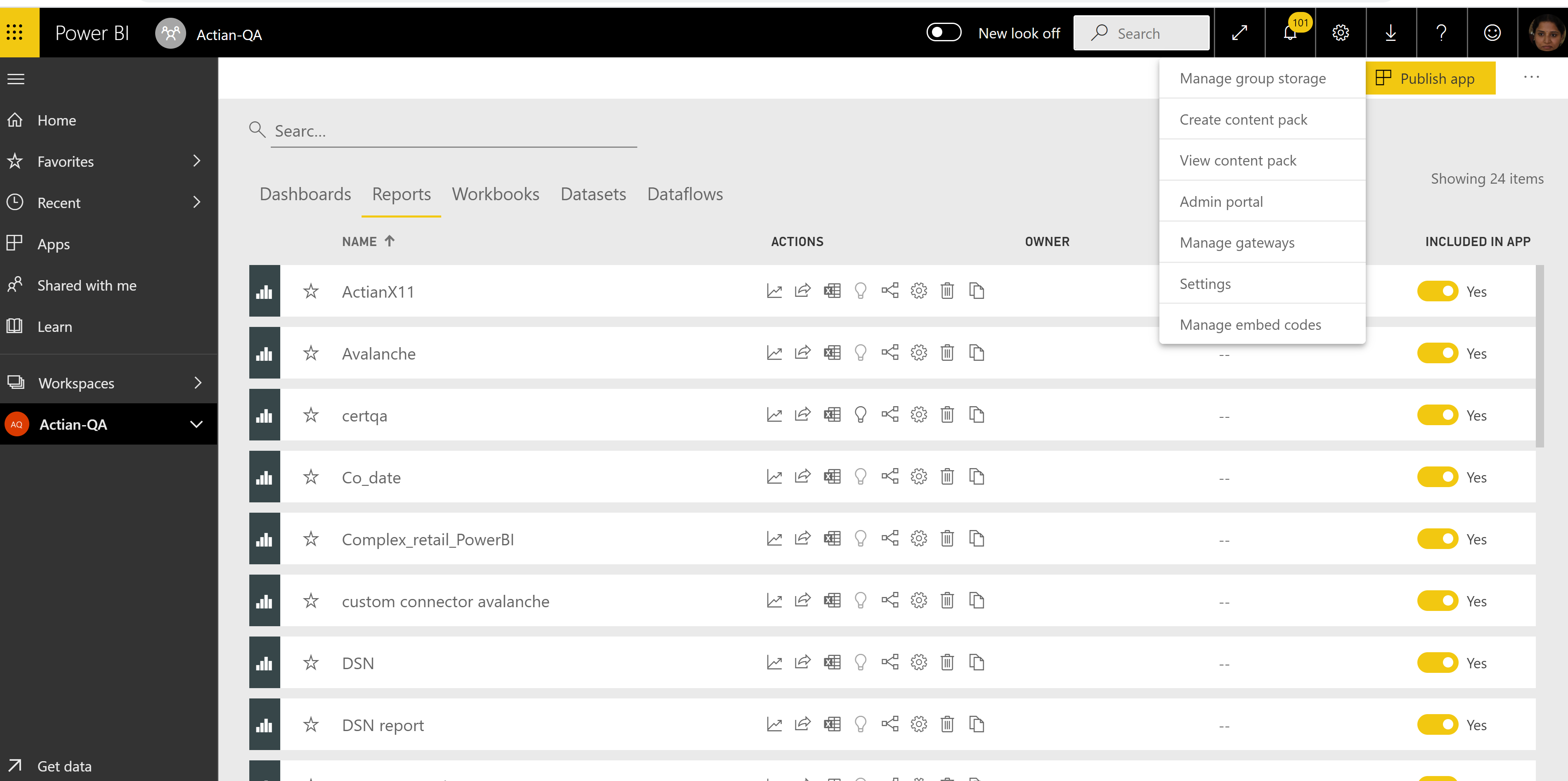
1. Install On-Premise Gateway recommended version (Not personal mode)
2. Create new Gateway
3. Go to connectors and copy the path of a folder where the data gateway loads custom data connectors from.



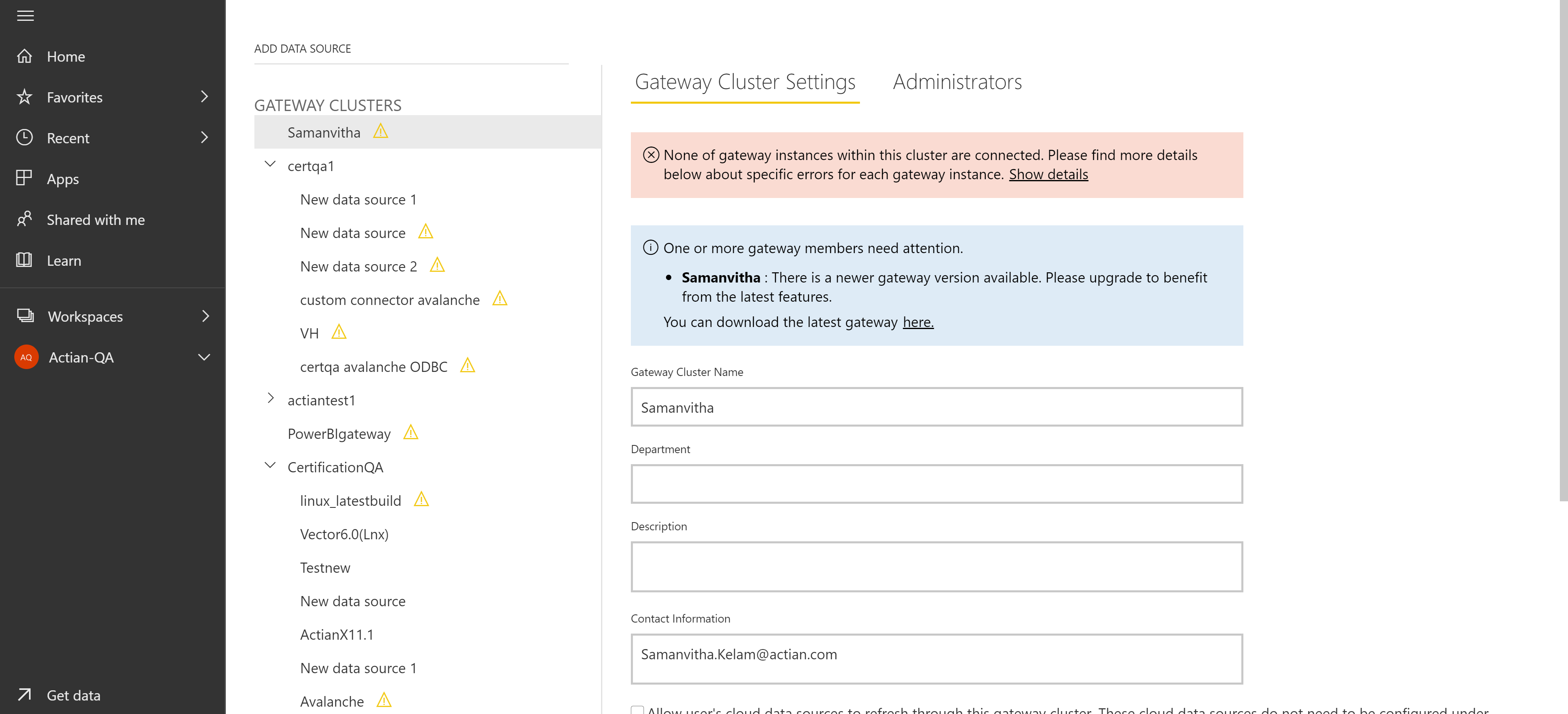
1. Restart the gateway.

**Add Data source to Gateway:**

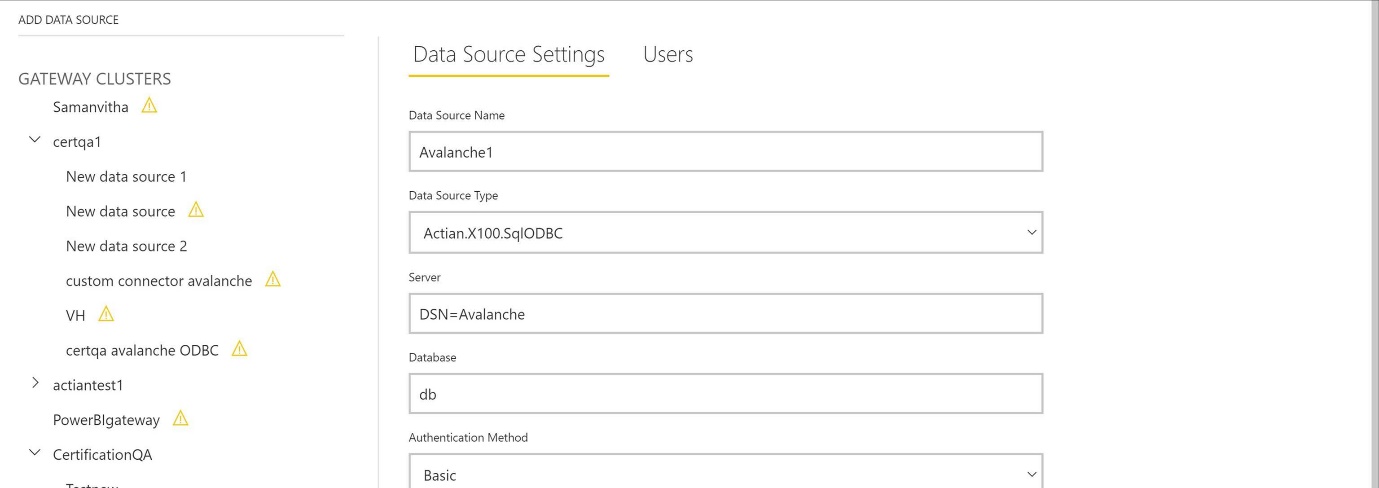
1. Login to PowerBI Service (app.powerbi.com)
2. Go to Settings🡪Manage gateway and select Gateway (created above).



1. Select Add Data source - > Data Source Settings -> Select custom connector from Data Source type

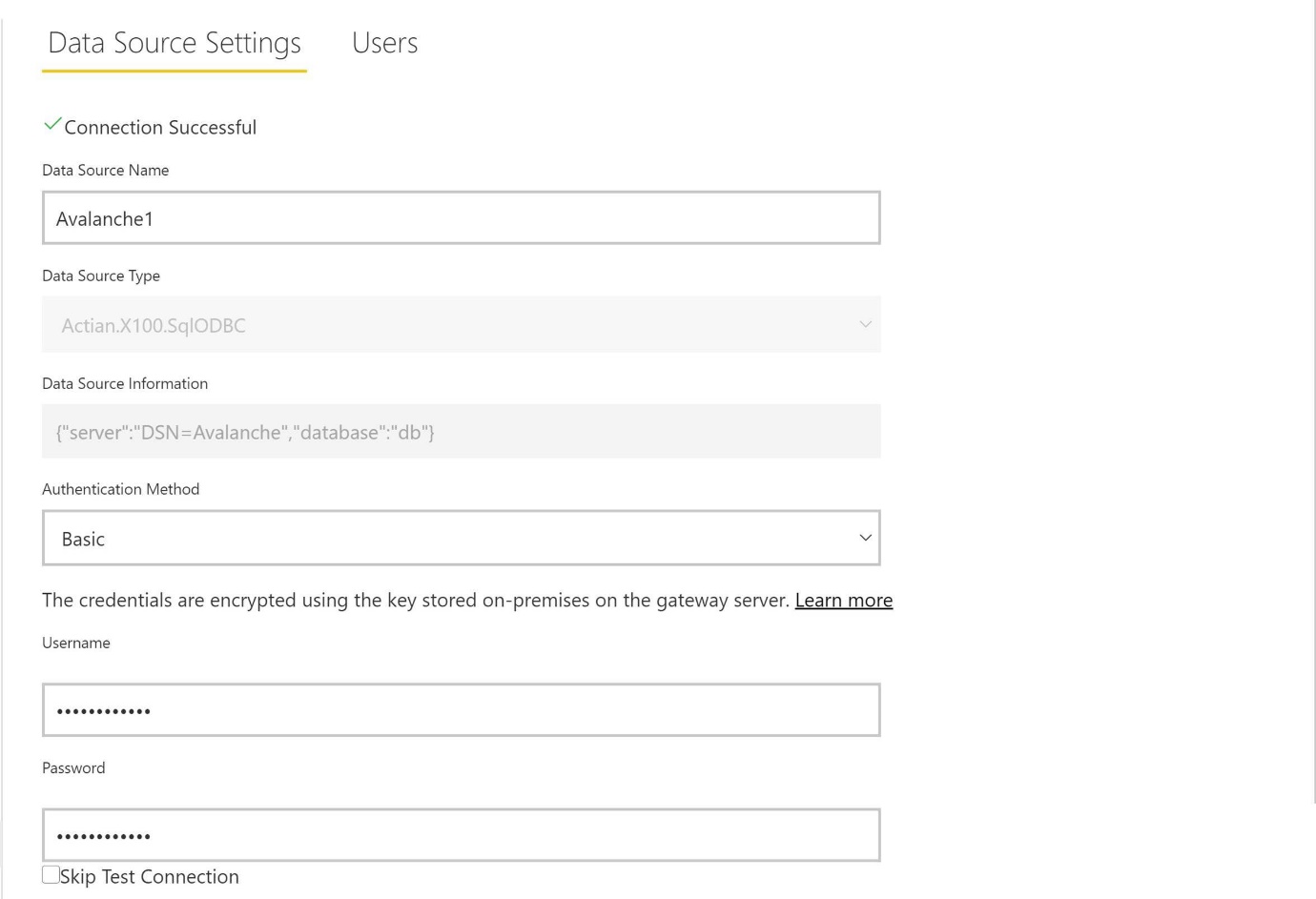


1. Enter Connection strings to Actian Vector/Avalanche and click on ADD.

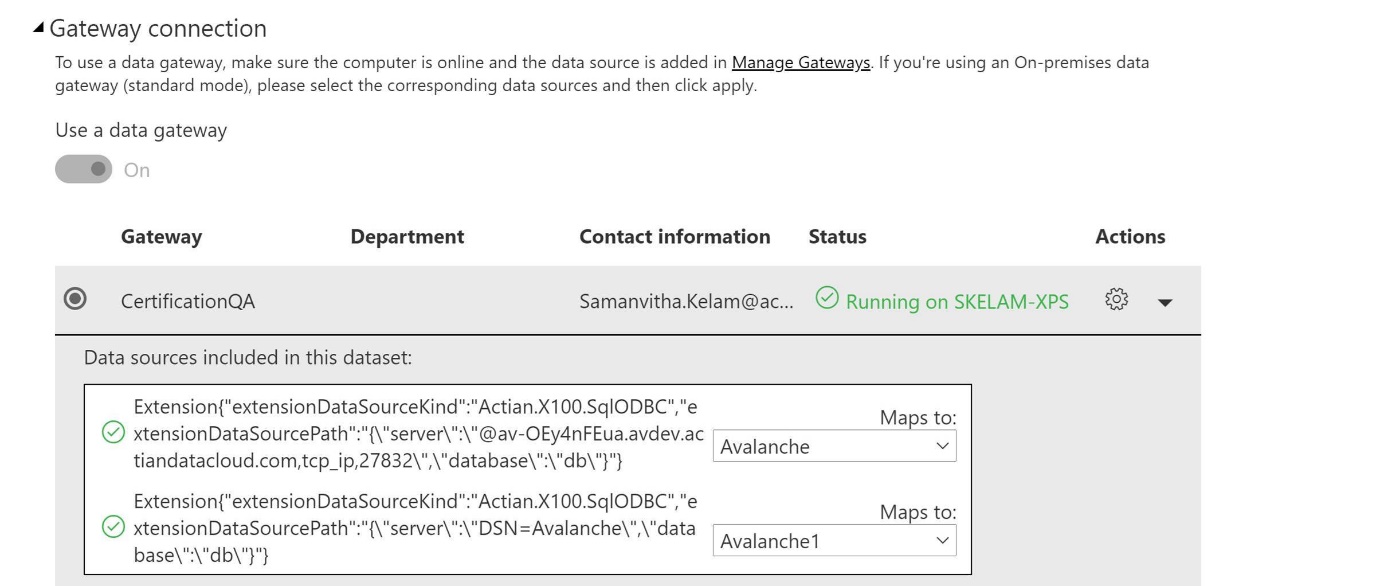


1. Test the connection.

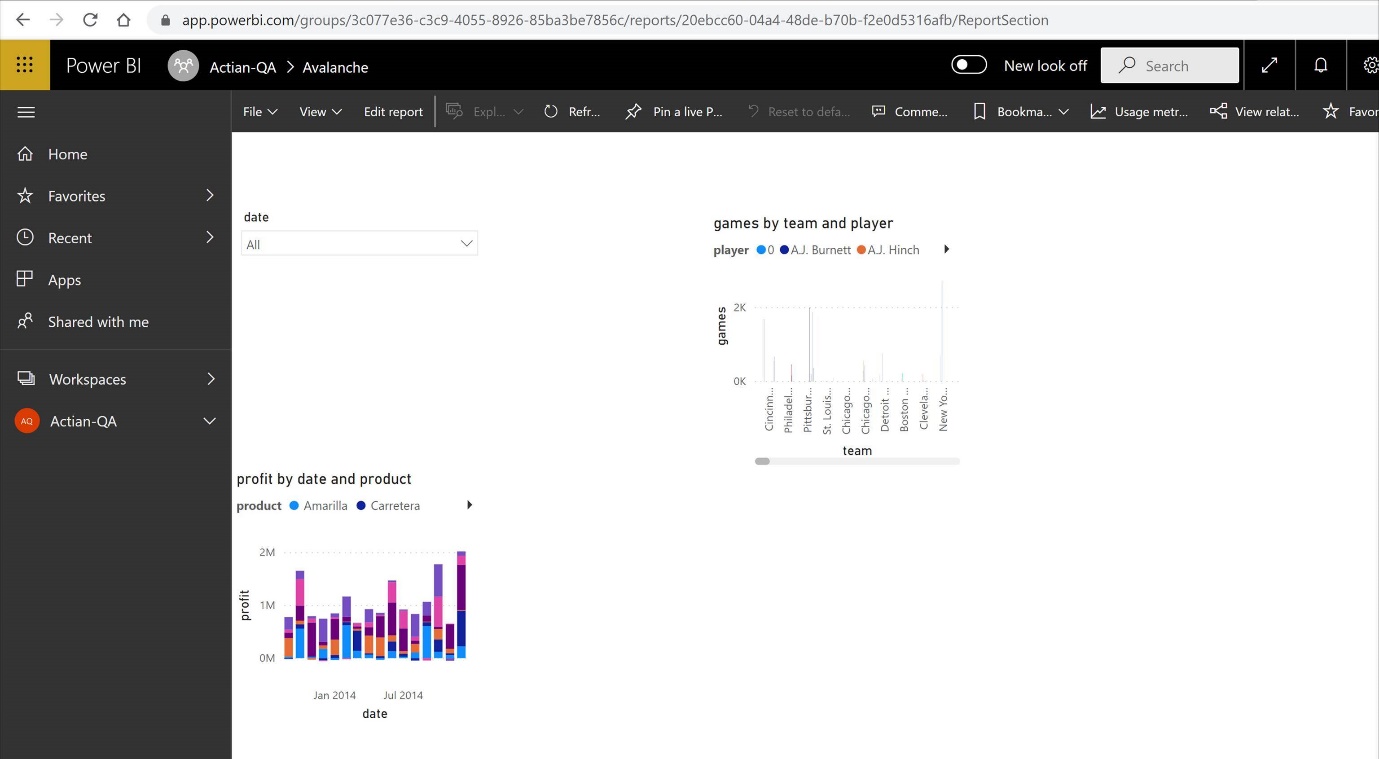
Avalanche Azure connection strings :



\*Able to add my datasource to my gateway it is up and running.



\*Reports are published to Actian-QA Workspace.



# **Tested Use cases/Criteria’s**

Following are listed test cases validated as part of this connector testing:

PowerBI Build testing on Visual Studio:

|  |  |  |
| --- | --- | --- |
| TC No | TC Name | TC Status |
| 1. | Build the custom connector project in Visual Studio and test the connection. (Actian Vector) | PASS |
| 2. | Build the custom connector project in Visual Studio and test the connection. (Actian Avalanche) | PASS |
| 3. | Write a sample query in Actian.X100.SqlODBC. query. Pq file and validate results (Actian Vector) | PASS |
| 4. | Write a sample query in Actian.X100.SqlODBC. query. Pq file and validate results (Actian Avalanche) | PASS |

PowerBI Desktop testing:

|  |  |  |
| --- | --- | --- |
| TC No | TC Name | TC Status |
| 1. | Verify the custom connector with Actian Vector – Windows and create a sample PBIX report | PASS |
| 2. | Verify the custom connector with Actian Ingres – Windows and create a sample PBIX report | PASS |
| 3. | Verify the custom connector with Actian Vector – Linux and create a sample PBIX report | PASS |
| 4. | Verify the custom connector with Actian Ingres – Linux and create a sample PBIX report | PASS |
| 5. | Verify the custom connector with Actian Avalanche AWS and create a sample PBIX report | PASS |
| 6. | Verify the custom connector with Actian Avalanche Azure and create a sample PBIX report | PASS |
| 7. | Create and manage relationship with auto detect | PASS |
| 8. | Create and manage relationship manually | PASS |
| 9. | Apply PowerBI DAX basics with Actian Vector (Direct Query) and prepare visualizations | JIRA# 3832, 4803 |
| 10. | Apply PowerBI DAX basics with Actian Avalanche (Direct query) and prepare visualizations | JIRA#3832, 4803 |
| 11. | Apply PowerBI DAX functions with Actian Vector (Import) and prepare visualizations | PASS |
| 12. | Apply PowerBI DAX functions with Actian Avalanche (Import) and prepare visualizations | PASS |
| 13. | Create a PBIX report, Publish datasets and reports (Direct Query) | PASS |
| 14. | Create a PBIX report, Publish datasets and reports (Import) | PASS |

PowerBI Service testing:

|  |  |  |
| --- | --- | --- |
| TC No | TC Name | TC Status |
| 1. | Verify the published reports are displaying and visualization are loading (Direct Query) | PASS |
| 2. | Verify the published reports are displaying and visualization are loading (Import) | PASS |
| 3. | Validate new data sources are added to gateway for Custom connector | PASS |
| 4. | Verify the published reports are assigned with On-premise gateway for Data Refresh (Vector) | PASS |
| 5. | Verify the published reports are assigned with On-premise gateway for Data Refresh (Avalanche) | PASS |

# **Test Data**

Datasets used during testing- Finance DB.

# **External References**

Bitbucket link for PowerBI Custom Connector Installation:

<https://alm.actian.com/bitbucket/projects/ING/repos/powerbi_driver/browse>

Gateway Setup:

<https://docs.microsoft.com/en-us/data-integration/gateway/service-gateway-install>

JIRA Links:

https://alm.actian.com/jira/browse/II-3832

# **Results/Next Steps**

**PowerBI Desktop:**

PBIX Reports and Datasets created and validated during testing are available at below OneDrive path:

[**\\DEPATIL-XPS\Users\depatil\OneDrive**](file:///\\DEPATIL-XPS\Users\depatil\OneDrive) **- Actian Software\Power BI Testing\PBIX files functionwise**

Complex Dashboard prepared for Demo and related emails are added below for reference:



