Connecting To Data Sources

Table of Contents

1. General Information about Data Sources

1.1 Understanding Data Sources

For Data Source Security see Configurable Data Source Security (5.2.1 and beyond)

There are two types of Data Sources in Metric Insights:

- **Configurable:** are Data Sources for which the source database and access credentials are user-defined. Each Metric Insights customer can establish and maintain SQL-based access to internal databases and Plugin connections to external sources, including Web Services and a wide range of data service providers.
- **Non-Configurable:** are received by all customers and require no customer maintenance or configuration

1. Configurable Data Sources (You can use either of the following to pull data into Metric Insights):

- **SQL**: Any source accessible through a JDBC driver is considered a SQL data source. This includes traditional databases as well as NoSQL sources with JDBC access; for example, Hadoop Hive.
- **Plugin:** Are used to obtain data that is not fetched using a JDBC driver and is collected using a plugin. Plugins are special components, built by Metric Insights, that send a native fetch command to a Data Source. Results are fetched in a way that allows Metric Insights to consume the data.

NOTE: Web Services are considered to be a Plugin configured to access data from a custom Web Service

2. Non-Configurable Data Sources available to all customers are:

- **Datasets:** Datasets function as an additional layer between Data Sources and elements created in Metric Insights.
- **Manual/CSV Data:** Key enter individual data points or upload data from a CSV file
- **External Process:** Push data to Metric Insights (**does not** require a credential set, but can authenticate if required); a **Data Posting URL** is provided.
- **Existing Datasets - SQL** *(introduced in Version 5.0.5 for fetching Reports and Version 5.1 - for Metrics)***:** You can fetch data from any Dataset created in the system to source a new Report or Metric.
- **Existing Reports - SQL:** Use data in one or more Reports to populate a new Report.
- **Single Existing Report:** A method similar to "Existing Reports - SQL" described above, but does not require writing a fetch command - the process is more automated.

- **Existing Metrics:** Obtain data in one or more Metrics to populate a new Metric or Report.
- **Aggregate a Metric**: Build a new Metric by using various aggregation options.

3. **Remote Data Collectors**

- If a **Data Source** requires a **Remote Data Collector**, see Configure a Remote Data Collector for more information

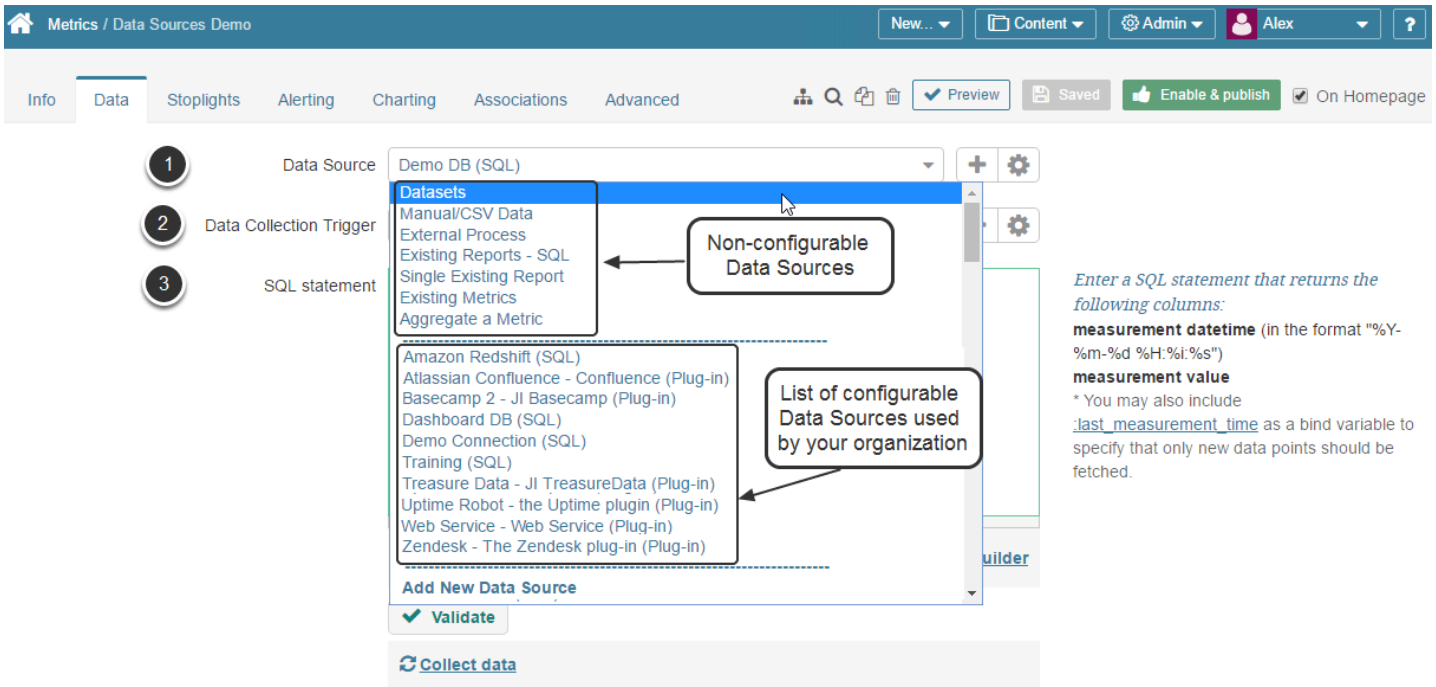
1. Data Source Editors

Information on creating new Data Sources can be found at:

SQL Data Sources

Plugin Data Sources

2. Select a Data Source on an Element Editor

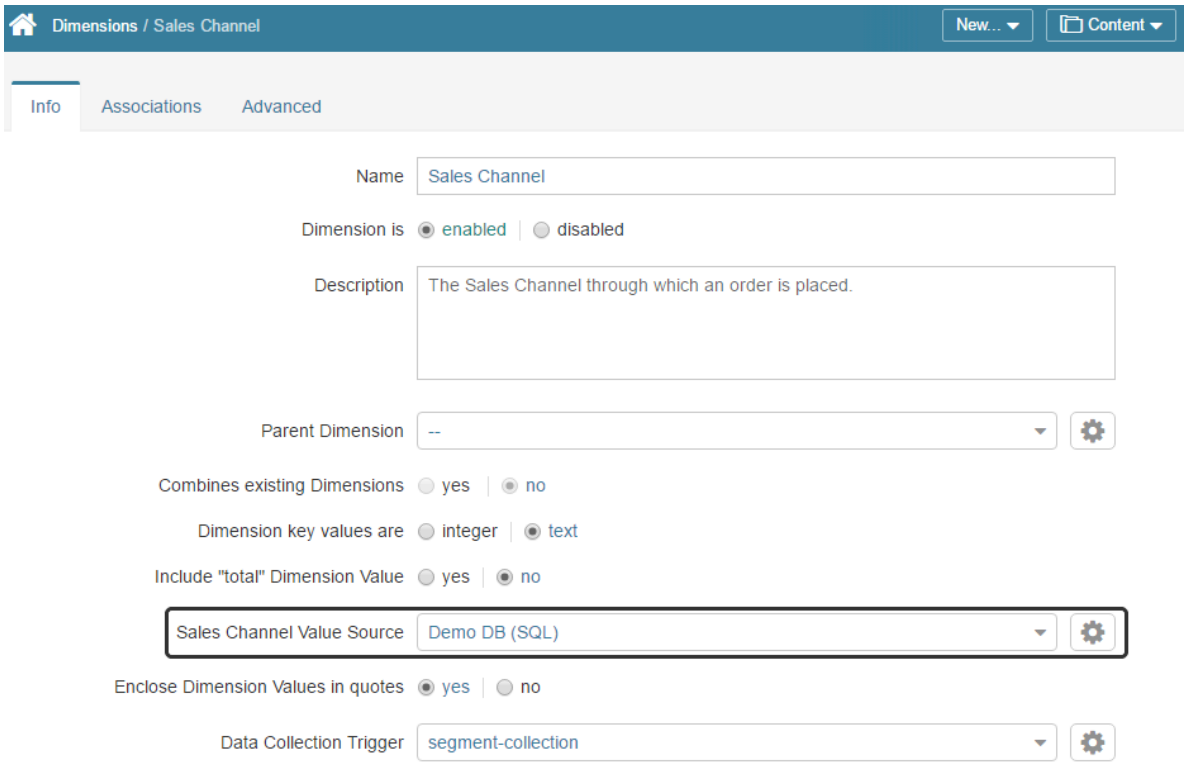


When creating a new element, once you have provided the general information, you are directed to the element *Editor* > *Data Collection* tab.

1. The **Data Source** drop-down list contains all of the **SQL** and **Plug in Data Source**s that have been defined for your instance as well as all non-configurable Data Sources.

2. Select a **Data Collection Trigger** that fetches the data. Those are used to control when content should be updated.
3. Based on your **Data Source** selection, you either see a **Plugin Command** / **SQL Statement** text box into which you enter the fetch command or other options for defining what data should be fetched from the respective source.

3. Data Source is defined in Dataset / Metric / Report / Event / Dimension and Data Dependency Editors



The *Event*, *Dimension* and *Data Dependency Editors* all have data collection settings similar to those on the element editors.

The example above shows the *Dimension Editor*.

1.2 Overview of using existing Elements as Data Sources

Let's say that you track sales activity in **Salesforce**, and web traffic in **Google Analytics.** How would you compute the ratio of Sales Opportunities to Web Visits? It would be easy if you had all of the data in one place. With Metric Insights, you do.

In this edition of Tips & Tricks, we'll show you how to use existing elements as data sources for new elements. We'll start with metrics because they're easy.

Let's say that your Monthly Opportunities metric has element ID = 1, and your Monthly Visits metric has element ID = 2. One way to compute the ratio of Opportunities to Visits would be to use 'Existing Metrics' as your data source, along with the following simple fetch command:

:1 / :2

Metric Insights interprets that as the ratio of metric #1 to metric #2, which is what we want. The arithmetic for such combinations is intuitive. If it looks reasonable, it will probably work.

You may occasionally need to do a computation that can't be handled with simple arithmetic. In such cases, you can work directly with the MySQL table where your metric data is stored.

Let's repeat the calculation of the Opportunity-to-Visit ratio. This time, we'll do it the hard way. Instead of 'Existing Metrics', we'll use 'Dashboard DB' as the data source. The Dashboard database is the logical backbone of Metric Insights and is where all of your data and metadata are stored. The table that we want is metric_measured_value.

Here's the SQL code:

```
Select
  m1.measurement_time,
  m1.measurement_value_int / m2.measurement_value_int
From
  metric_measured_value m1,
  metric_measured_value m2
Where m1.measurement_time = m2.measurement_time
  And m1.metric_id = 1
  And m2.metric_id = 2
```

That's not as elegant as the first method, but it's more flexible, so it might come in handy for complex calculations.

Note: We're assuming that both metrics have data type = Integer. If either of them has data type = Decimal, we would use measurement_value_float instead of measurement_value_int.

As our final example, we'll once again compute the ratio of Opportunities to Visits, but this time we'll assume that the relevant data has been collected in reports instead of metrics. Specifically, we'll assume that we have a 'Monthly Opportunities' report, with columns named 'Opportunity Month' and 'Opportunities'. We'll also assume that we have a 'Monthly Visits' report, with columns named 'Visit Month' and 'Visits'.

The first step is to edit both reports and set 'Would you like to create other elements based on this report?' to 'Yes'. (Look for it in the Advanced Settings section.) When you do that, Metric Insights will create a MySQL table based on your report. The rules for converting report and column names to table and field names are simple: uppercase letters are converted to lowercase, and all special characters (including spaces) are converted to underscores.

Back in the metric editor, select 'Existing Reports' as your data source and use the following SQL code:

```
Select
  opportunity_month,
  opportunities / visits
From
  monthly_opportunities,
  monthly_visits
Where opportunity_month = visit_month
```

As a bonus tip, you can get a list of tables for the 'Existing Reports' data source by creating a report whose SQL statement is simply 'show tables'. Once you know the name of the table, you can find out what's in it by replacing the SQL statement with 'desc <table>', where <table> is the name of your table. (These tricks will also work for any other SQL data source.)

In version 3.1 (soon to be released), we will include a graphical 'SQL Builder' that will make this kind of operation even easier.

You can learn more about building metrics from Existing Metrics or Existing Reports

1.3 Insightd System Requirements

System Requirements

Insightd is pretty simple and can run on any Windows or Linux server behind your firewall that:

- Has significant uptime,
- Has Java Runtime Environment version 1.6 or later
- Has outbound connectivity to the Metric Insights server
- Has network connectivity to your internal data sources

If you'd like to connect to Microsoft SQLServer, for example, you could install Insightd on the SQLServer machine itself, or any other machine in your network that can already query your SQLServer database.

1.4 MI Preflight: Plugin pre-check tool

MI Preflight is an admin tool for running preliminary tests before installing a Remote Data Collector on your Windows server.

A Remote Data Collector is necessary for integrations with:

- QlikView
- Qlik Sense
- Microsoft Power BI
- Tibco Spotfire

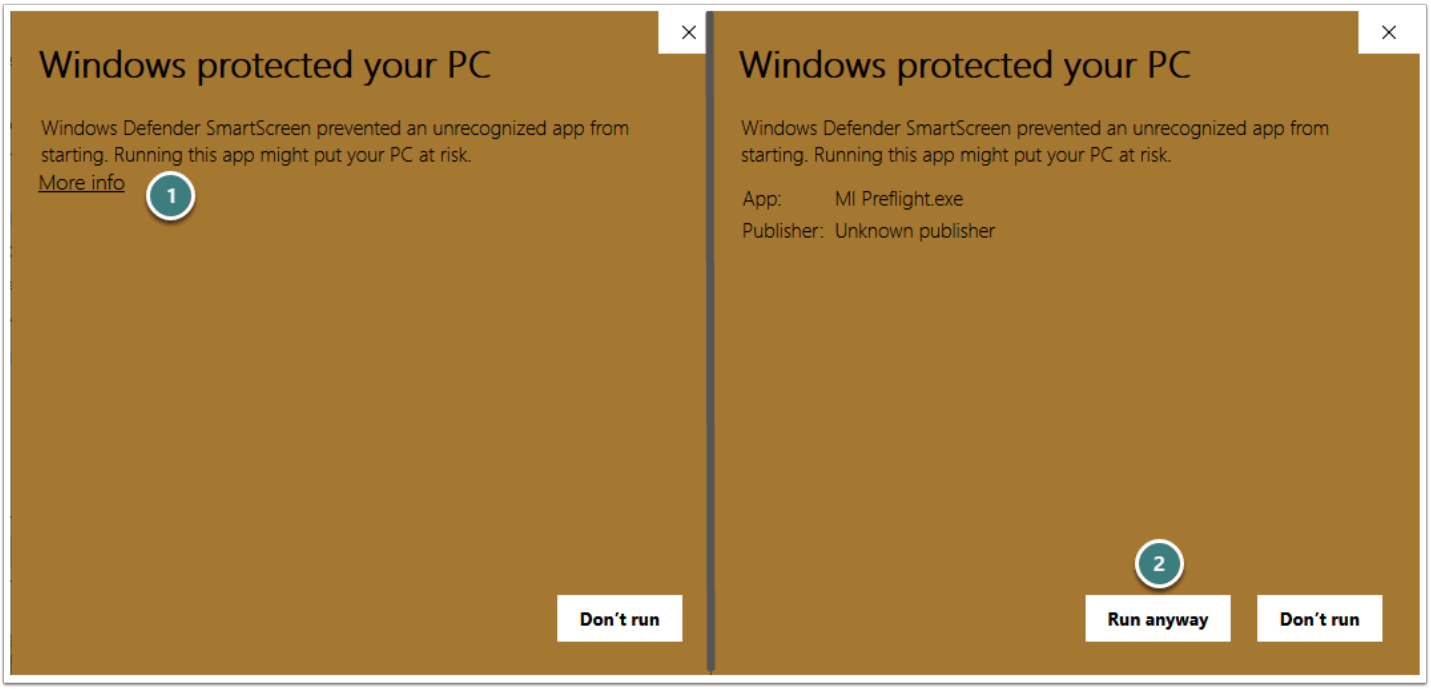
The MI Preflight application checks:

1. System Specifications
2. Java Version
3. Ports needed for the relevant integration

[Release 6.x]: 'Remote Data Collector' renamed to 'Remote Data Processor'

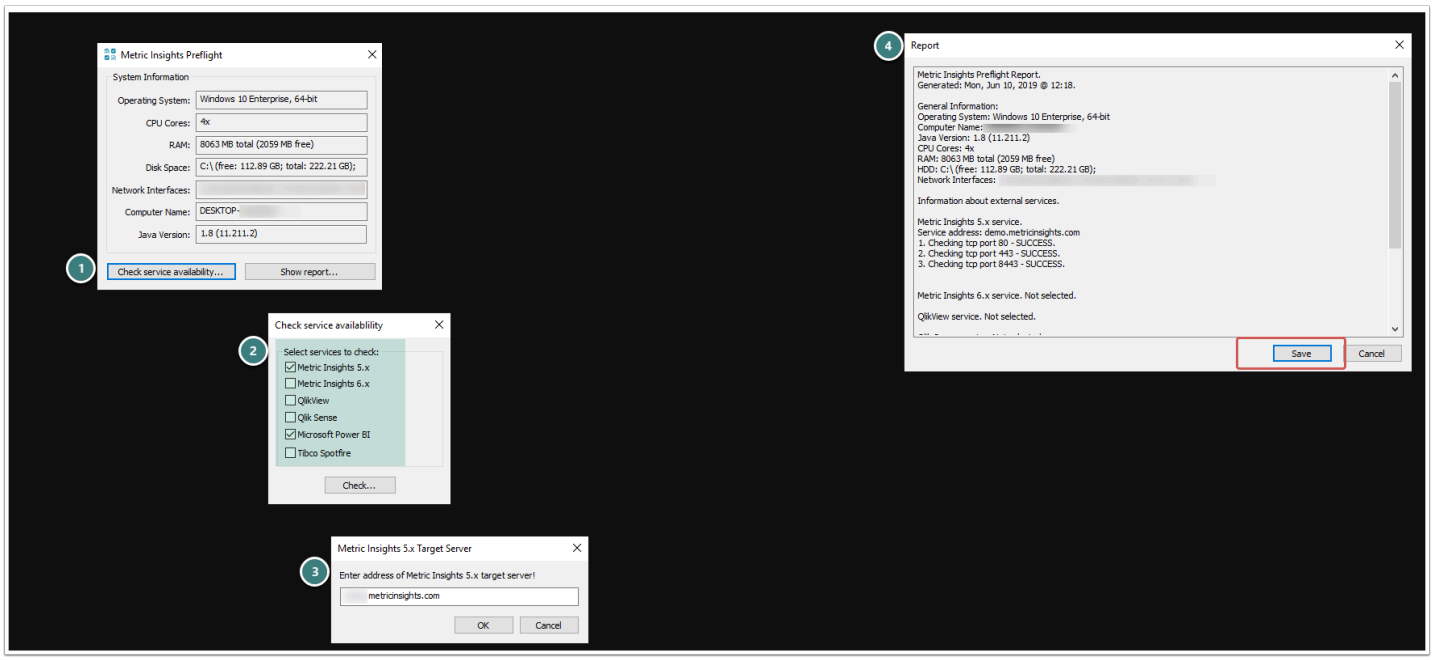
1. Download the MI Preflight app

To download MI Preflight, go to https://s3.amazonaws.com/metricinsights-downloads/MI+Preflight/latest/mipreflight.zip



1. When unzipping the **mipreflight.zip** package, you might get a warning message from your *Windows Defender SmartScreen*:
   - Click **[More info]** and proceed
2. **Run** the tool

2. Run availability checks



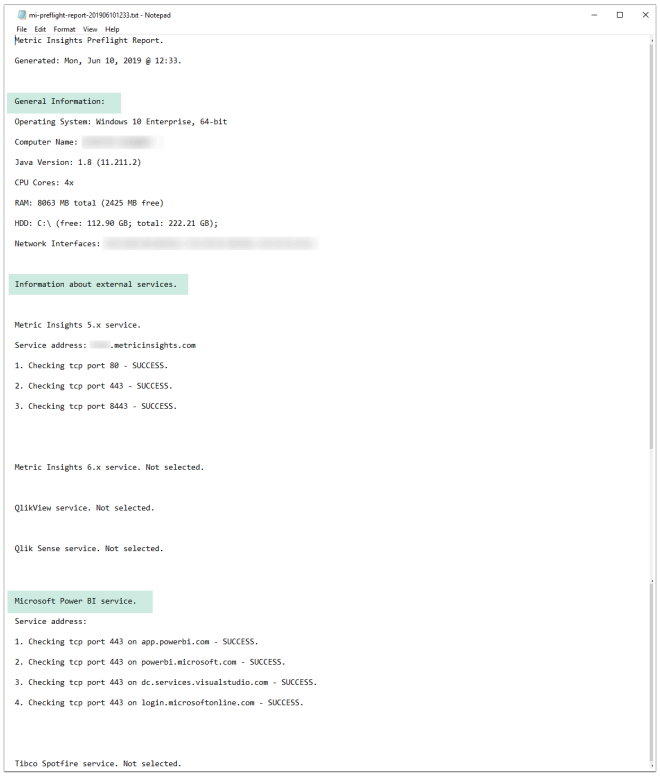
To confirm the availability of the required services:

1. Click **[Check service availability...]**

2. Select the **services** to check
3. Enter the **Target Server URL**
4. Your **Report** will be generated for you
    - Review the provided information
    - Optionally, ***Save*** your Report for future reference

3. Download the Report

> Upon Save, a Report text file is generated. If troubleshooting is required, email the file to **support@metricinsights.com**



What's next?

If no issues with the services availability have been detected, you can proceed to [Configuring Remote Data Collectors]

1.5 Create New SQL Data Source

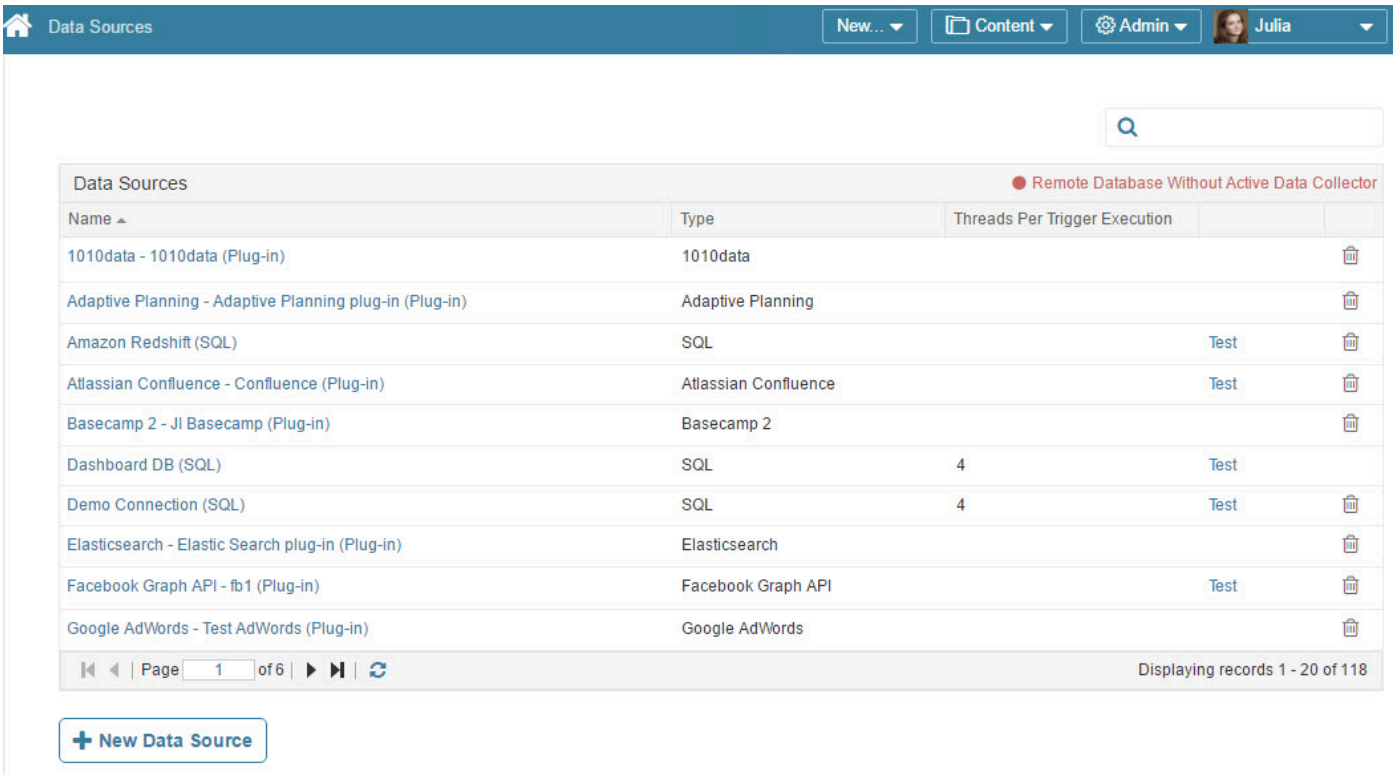
You can use a JDBC driver to pull data into Metric Insights using a SQL fetch command. This includes traditional databases as well as NoSQL sources with JDBC access; for example, Hadoop Hive.

This article describes the general process for creating a SQL data source. Information on creating a data sources based on a plugin can be found here.

For a description of Metric Insights overall approach to Data Sources, click here.

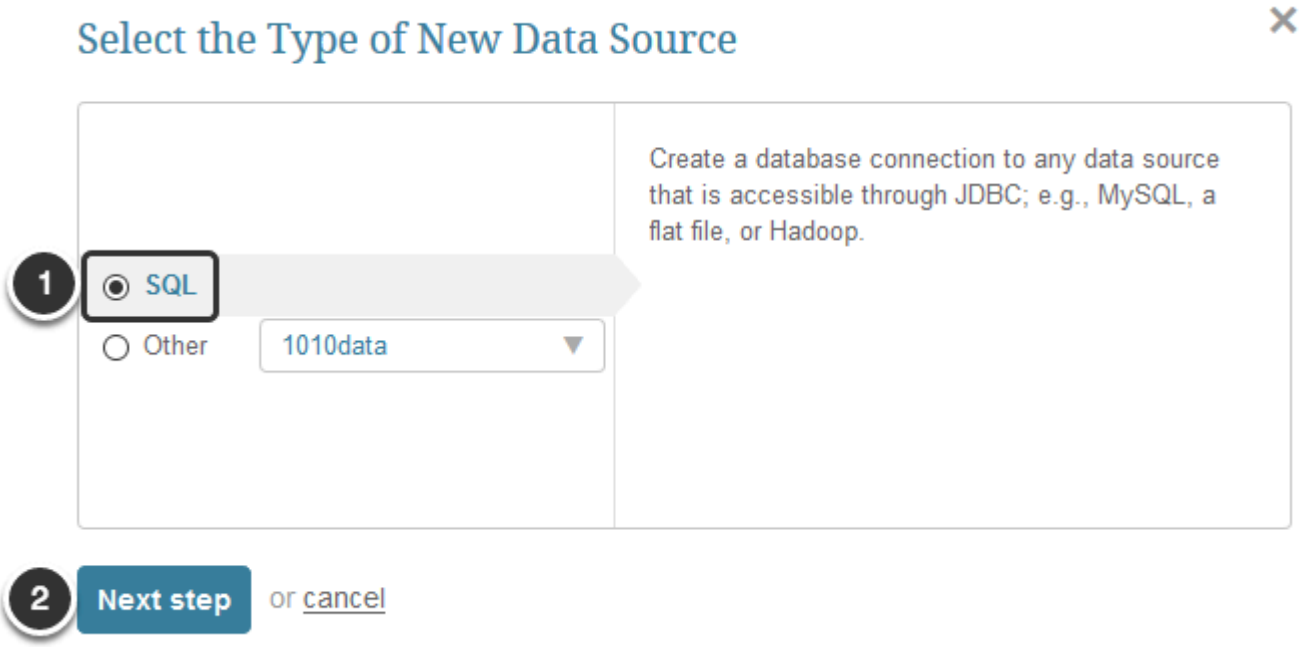
Video Tutorial

1. Access Admin > Data Sources



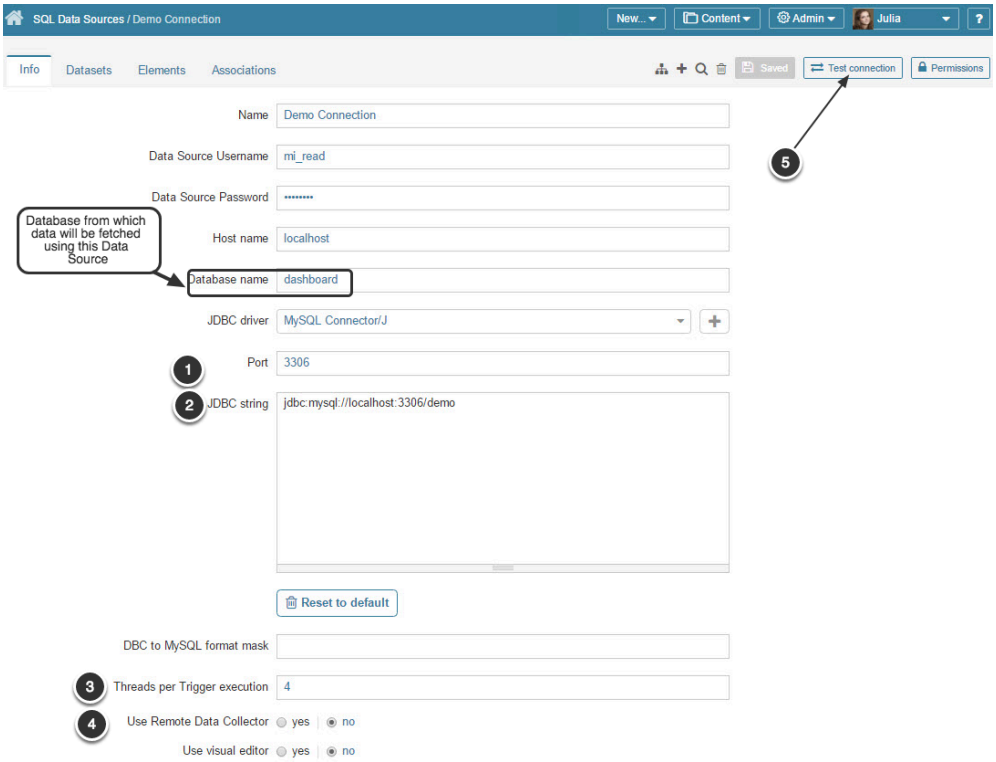
At the bottom of the page click **[+New Data Source].**

2. Choose type



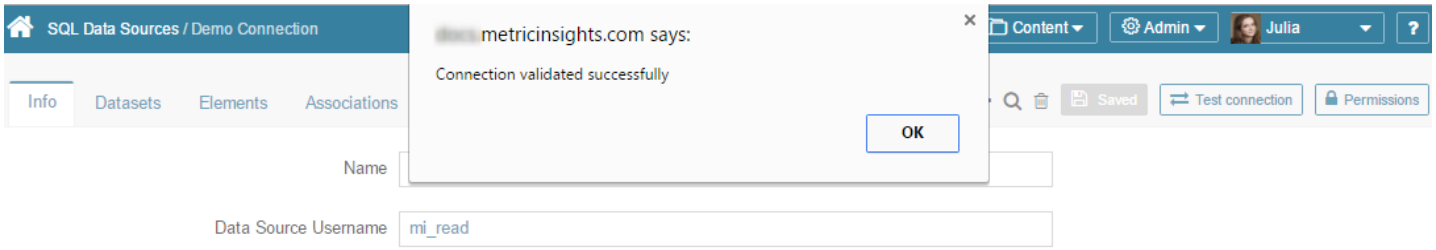
1. Select **SQL** as the type of a new Data Source
2. **Next step**

2.1. Complete all settings



1. The **Port** number will be set by default, based on your choice of **JDBC Driver**. Change it if necessary.
2. The **JDBC string** will be created automatically based on your other inputs. In some cases, however, it will not be possible to infer the correct string without additional inputs. If the Connection Test fails, contact Metric Insights for assistance..
3. Optionally, specify the maximum number of concurrent **Threads per Trigger execution** to be used in background processing when the system updates Metrics and Reports for this Data Source. If you do not specify any value for this setting, batch data collection processing will be single-threaded.
4. If you select 'yes' to specify that this Data Source is remote; for example, behind a firewall, you will be required to select a Connector or create a new one. See Configure a Remote Data Collector for more details.
5. Once a new Data Source is saved, click **Test Connection** to ensure that settings are correct. **NOTE:** Some data sources cannot be tested this way.

2.2. Review and close the confirmation



If the connectivity is established, the confirmation message appears; click **OK** to continue.

NOTE: It is not possible to directly test Hive connectivity from the connection editor since there is no standard query that can be run against a HiveQL instance. Contact Metric Insights for assistance in validating a HiveQL connection.

1.6 Create a New Plugin Data Source

Data that is not fetched using a JDBC driver is collected using a Plugin. Plugins are special components, built by Metric Insights, that send a native fetch command to a data source. It then formats the results in a way that allows Metric Insights to consume the data. A list of supported Plug-ins can be found here. It is also possible to access data from a custom Web Service by using a Web Service Plugin.

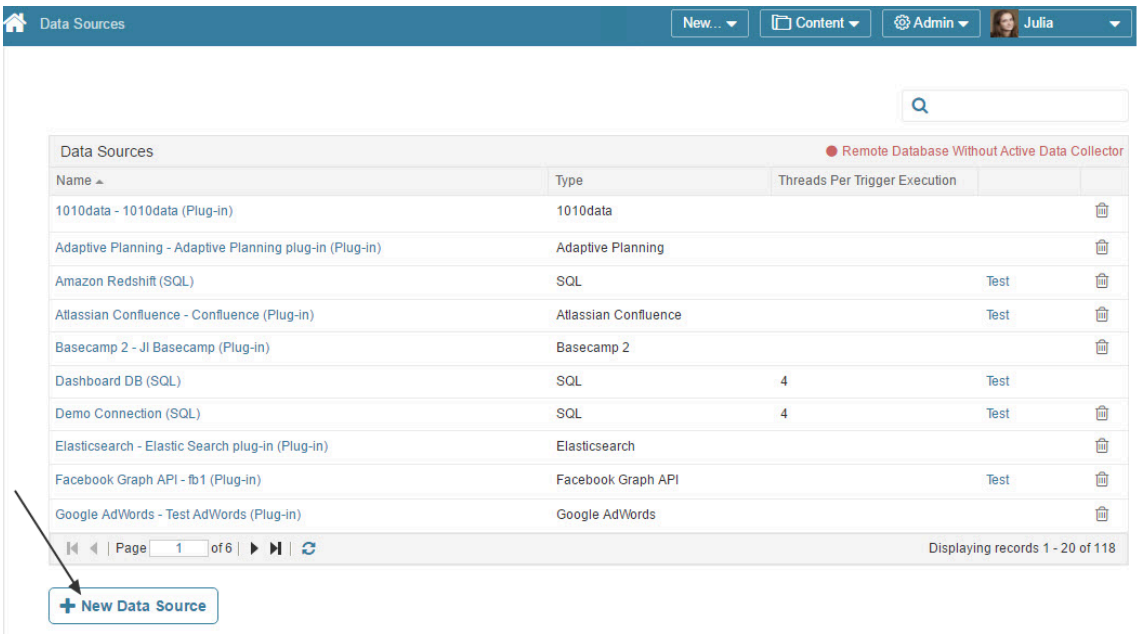
This article describes the general process for creating a Plugin data source.

Information about how to create a new SQL data source is available here.

For a description of Metric Insights overall approach to Data Sources, click here.

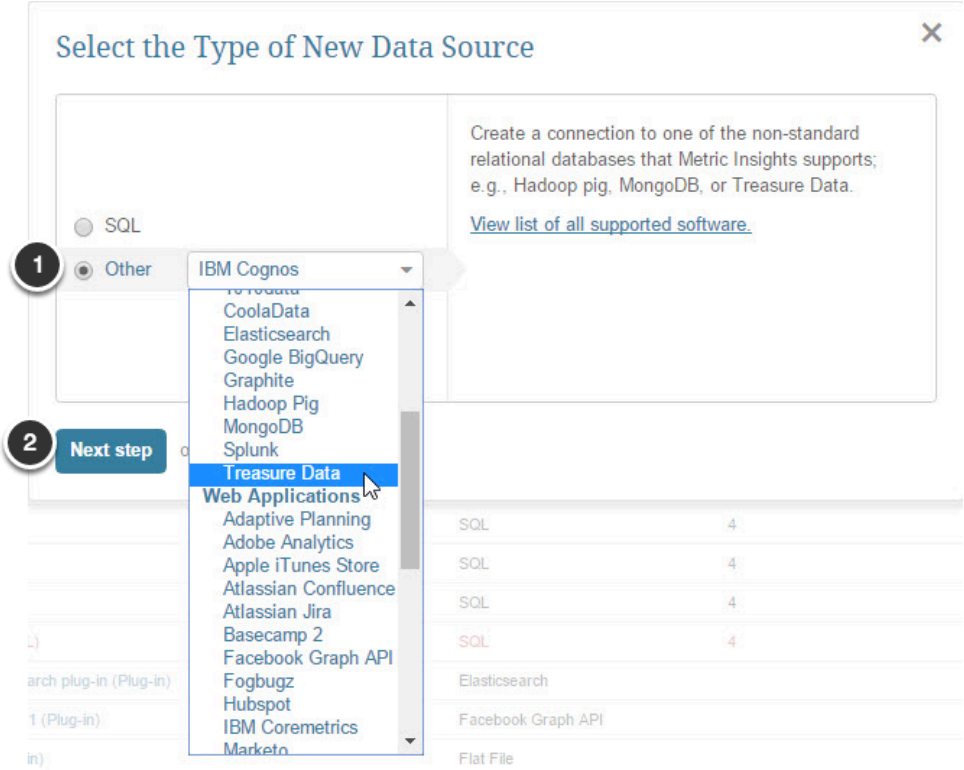
Video Tutorial [Tableau Example]

1. Access Admin > Data Sources



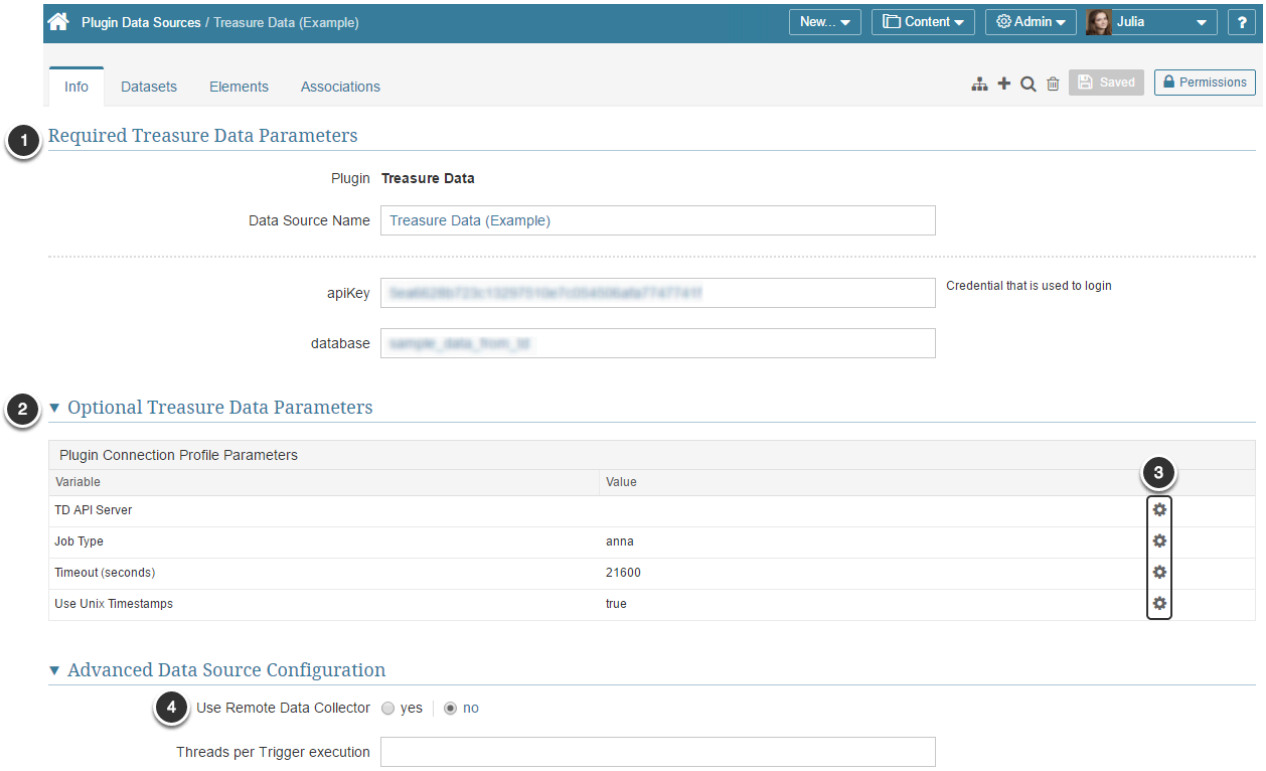
At the bottom of the page click **[+New Data Source].**

2. Choose type



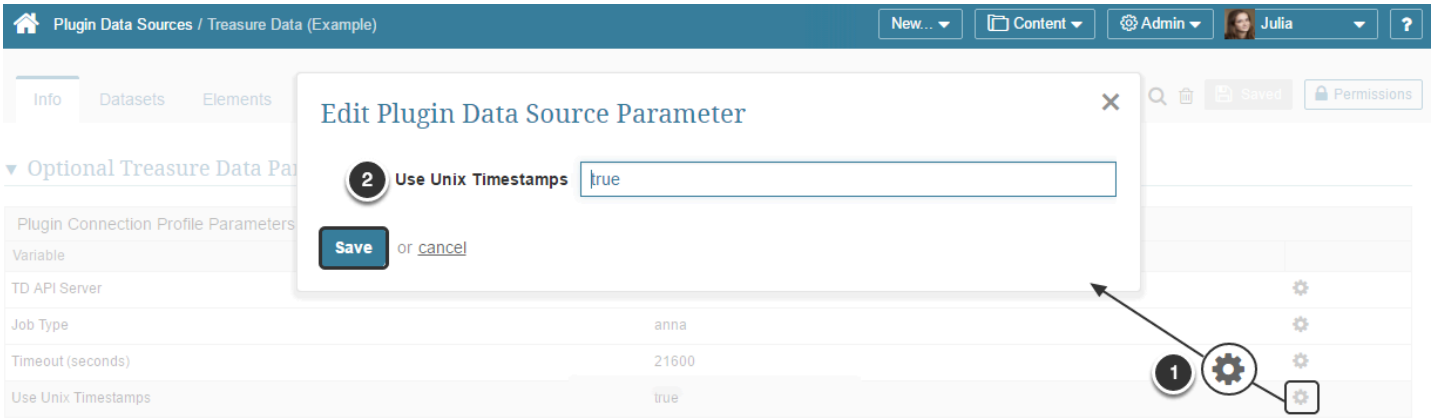
1. Click **Other** and select a required data service from the drop-down list of configured Plugins. If you do not find the one you need, contact your system Administrator or support@metricinsights.com.
2. Click **Next step**

3. Complete required and optional settings



1. Define required plug-in parameters: name and credentials you use to connect to a given data source.
2. When the supported plugin is configured, the required **Plugin Connection Profile Parameters** are defined and are defaulted into the **Parameters** grid once you select the plugin setting
3. Any of these parameters may be edited
4. If you select 'yes' to **Use Remote Data Collector?**, you will be required to select a collector or create a new one at the bottom of the page. For more information, reference [Define a Remote Data Collector]

3.1. Editing Optional Parameters



1. Use the **Edit** (Gear) icon to enter/modify the parameter
2. Change the parameter to be passed to the Data Source

Save

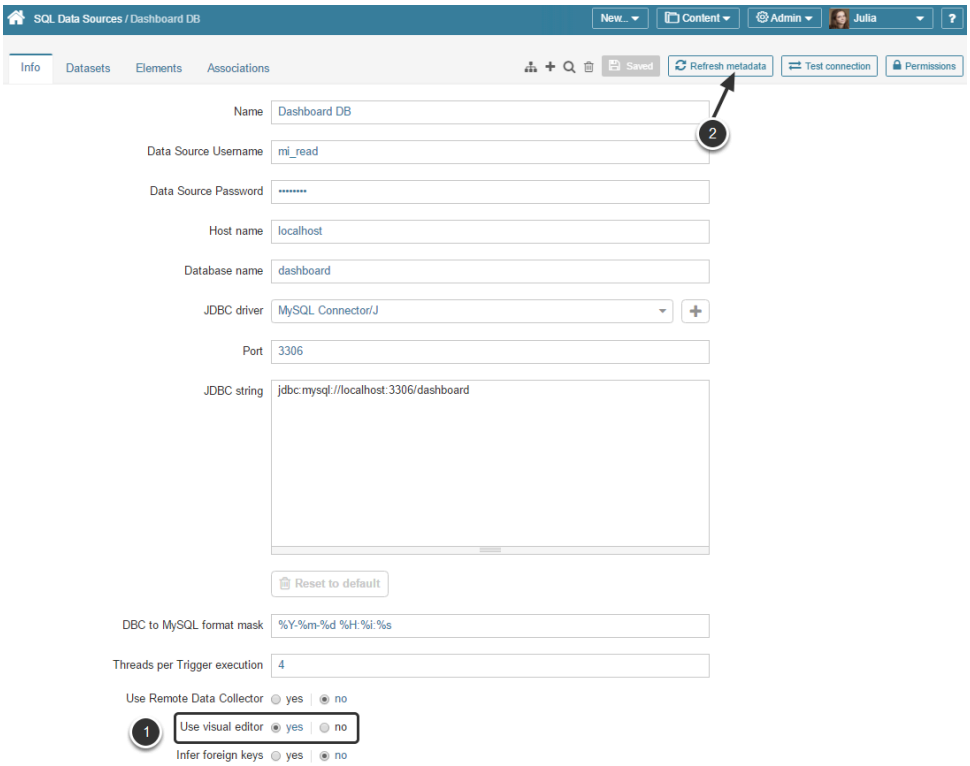
Repeat this process as necessary to update any/all parameters.

Save again before leaving the Editor so that your Data Source will be ready for use for defining an Element (Report / Metric / Multi-metric / Dataset).

1.7 What does the Refresh Metadata button do?

Learn what the **Refresh Metadata** button does in the *SQL Data Source Editor*.

Open the SQL Data Source Editor



For SQL data sources, if using the visual editor (1), table/column metadata from the data base is collected periodically and cached on the Metric Insights server. When users go into the *Visual Editor* from an Editor, the cached metadata is used instead of making a database hit to pull up table/column information.

The **Refresh Metadata** button (2) allows you to force the refresh of this metadata.

1.8 Metric Insights' Plugins

About Plugins

Metric Insights has a flexible plugin infrastructure which allows us to connect to many different data sources to pull metrics into the KPI warehouse.

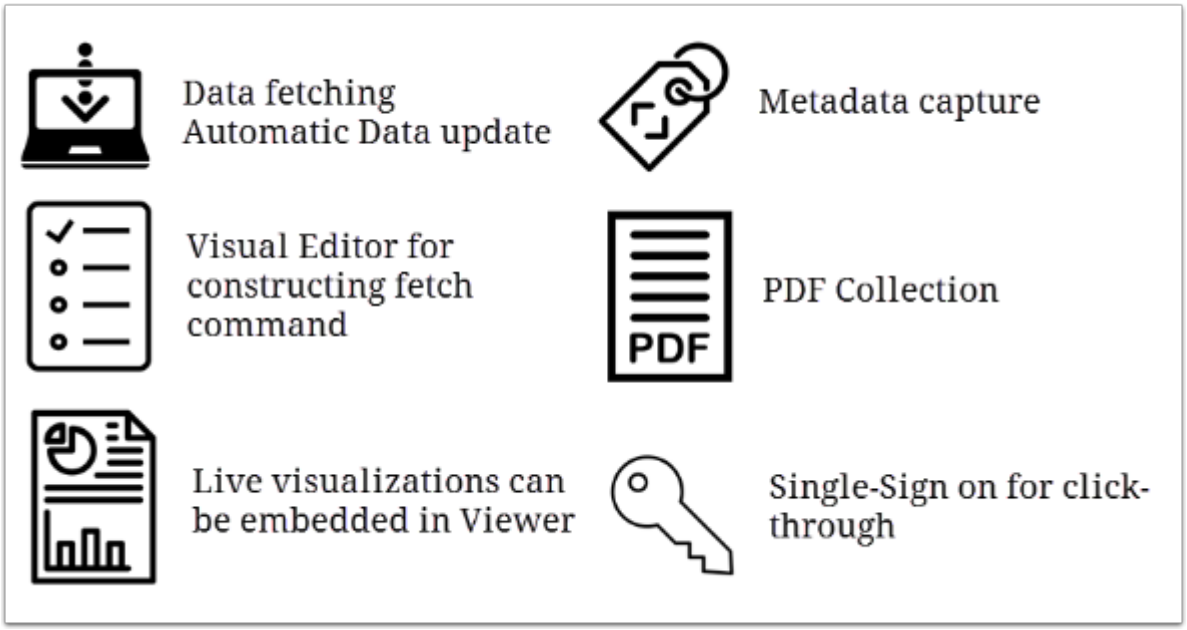
In order for Metric Insights to have access to your data source, however, in some cases it is necessary to install Insightd (a thin remote data collection process) on a machine inside your network. If a plugin requires Insightd to be running on a machine in your network, it will be noted below.

Note: *Even if a plugin is listed as* ***not*** *requiring Insightd, if your network configuration is such that your data source is locked down and the Metric Insights server does not have access to it, you will still need Insightd to run on a machine behind your firewall that has network access to your data sources. General stand-alone installations of Metric Insights should not require this, however*

The list below includes our most popular plugins. If you don't see the the plugin that you need in the list below, please contact Metric Insights.

Supported Plugin List

Icons Legend



Software	Capabilities in Metric Insights	Service versions supported	Compatible MI Versions
1010 Data	• Data fetching • Automated data update • Single-Sign on for click-through / embedding	v. 9	4.2, 5.0
Adapative Planning	• Data fetching • Automated data update	API Version 7.0	
Adobe Analytics	• Data fetching • Automated data update • Connection can be tested	Not applicable	5.3
Beckon	• Local data collection • Data fetching • Visual Editor for constructing fetch command • Automated Image Fetching		4.2, 5.0
SAP Business Objects	• Local / remote data collection • Visual Editor for constructing fetch command • Metadata capture (manual object catalog maintenance and automated) • Automated Image / PDF Collection • Live visualizations can be embedded in Viewer	v. 14.1.5.1533	4.2, 5.0

Cognos	• Local / remote data collection • Visual Editor for constructing fetch command • Metadata capture (manual / automated) • Live visualizations can be embedded in Viewer • Automated Data / Image / PDF fetching • Connection can be tested • Supports MIQL	v. 10.2+	5.3.2
CoolaData	• Local data collection • Automatic data fetching	API Version 1.0	
Elasticsearch	• Local data collection • Metadata capture (manual / automated) • Automatic data fetching • Connection can be tested	API Version 6.x	5.3.1
File Data	• Import CSV files from a directory on a server • Data fetching is available • Connection can be tested • Supports MIQL	Not applicable	5.0
File Metadata	• Import files from a directory on a server • Data fetching is available • Connection can be tested • Supports MIQL	Not applicable	5.0
Google Analytics	• Local data collection • Metadata capture (manual / automated) • Automated data fetching • Visual Editor for constructing fetch command • OAuth 2.0 authentication • Connection can be tested • Supports MIQL	Not applicable	5.3
Google BigQuery	• Local data collection • Metadata capture (manual / automated) • Data fetching • OAuth 2.0 authentication • Connection can be tested	Not applicable	5.3.2
Google Calendar	Google Calendar plugin for importing a google calendar as an event calendar.	Not applicable	5.3.2

	• Local data collection • Metadata capture (manual / automated) • Data fetching • Visual Editor for constructing fetch command • OAuth 2.0 authentication • Connection can be tested • Supports MIQL		
Google Play Console	• Local data collection • Metadata capture (manual / automated) • Data fetching • Visual Editor for constructing fetch command • OAuth 2.0 authentication • Connection can be tested • Supports MIQL	Not applicable	5.3
Google Sheets	• Local data collection • Metadata capture (manual / automated) • Data fetching • Visual Editor for constructing fetch command • OAuth 2.0 authentication • Connection can be tested • Supports MIQL	Not applicable	5.3.2
Graphite	• Data fetching	Any version	
Hadoop Pig	Pig Latin plugin that runs against Hadoop. Last statement must be a DUMP statement. **NOTE:** Insightd Required (on a node configured to run Pig against your cluster)		
IBM Coremetrics	• Data fetching • Visual Editor for constructing fetch command	API Version 1.0	4.1 only
Marketo	• Local data collection • Data fetching	Any version	
MicroStrategy	• Remote data collection • Visual Editor for constructing fetch command • Metadata capture (manual object catalog maintenance and automated collection)	10.x, 10.4	4.2,5.0

	• Data fetching (writing a command manually or constructing via a visual editor) • Live visualizations in Viewer • Automated Image / PDF fetching • Supports MIQL		
Microsoft PowerBI	• Visual Editor for constructing fetch command • Metadata capture (manual object catalog maintenance and automated collection) • Automated Image fetching • Supports MIQL	13.0.1500.201	4.2,5.0
Mixpanel	• Local data collection • Data fetching	API Version 2.0	4.2, 5.0
MongoDB	• MongoDB Shell command plugin. Last statement must include printjson() statement • Data fetching		
Netsuite			
OLAP/MDX	• Use the MDX query language to query OLAP cubes like those in Microsoft SSAS • Data fetching		
QlikView	• Automated data collection with Visual Editor for easy construction of fetch command • Metadata capture (manual and automated) • Automated image fetching • Visual Editor for constructing fetch commands and pre-filtering of fetched data • Live visualizations in Viewer are available • Linking supported with dimension value filtering **NOTE:** Insightd Required	11.20.13405.0, QlikView 12.00 Service Release 2	4.2, 5.0
Qlik Sense	• Local / remote data collection • Metadata capture • Visual Editor for constructing fetch command • Pre-filtering of fetched data supported • Live visualizations can be embedded in Viewer • Automated Image fetching • Trusted server can be used for data collection **NOTE:** Insightd Required	3.0.0, 3.1.2	4.2, 5.0

Re:dash	• Local / remote data collection • Data fetching • Connection can be tested	5.x	4.0
RSS	• Local data collection • Data fetching • Visual Editor for constructing fetch command	RSS 2.0	
Salesforce	• Salesforce plugin to pull in metrics from the popular online CRM software	API Version 41	5.0
Salesforce SOQL	• Salesforce plugin to pull in metrics from the popular online CRM software	API Version 29.0	
Sisense	• Local/ remote data collection • Metadata capture • Automated Data / Image / PDF fetching • Live visualizations can be embedded in Viewer	6.2.2.35, 6.4.2.11006, 6.5.0	4.2, 5.0
Splunk	• Local data collection • Data fetching • Connection can be tested • Visual Editor for constructing fetch command	Any version	5.3
SSRS	• Remote / local data collection • Metadata capture (manual object catalog maintenance and automated collection) • Visual Editor for constructing fetch command • Automated Data fetching via the command constructed using corresponding synthax or using the embedded visual editor for constructing commands • Automated Image / PDF fetching • Live visualizations can be embedded in Viewer	Service	4.2, 5.0
SQL Databases	• Relational databases and other datastores with a JDBC driver, like Hadoop Hive, MySQL, Postgres		
Tableau	• Local / remote data collection • Visual Editor for constructing fetch command • Metadata capture (manual and automated) • Live visualizations can be embedded in Viewer • Automated Image / PDF Collection	9.x,10.1, 10.2	4.2, 5.0

	• Single-Sign on for click-through / embedding • Visualizations of reports can be downloaded for local editing • Trusted server can be used for data collection		
Teradata	• Data fetching		
Tibco Spotfire	• Data fetching • Metadata capture • Automated Data and Image fetching • Connection can be tested • Visual Editor for constructing fetch command • Live visualizations can be embedded in Viewer	v. 7.6.1	5.3.1
Treasure Data	• Data fetching	Online service	4.2, 5.0
Web Service	• Periodically hit a webservice to get measurement values		
Zendesk	• Local Data Collection • Data Fetching • Visual Editor for constructing fetch command • Integration Wizard	API Version 2	

1.9 MIQL Syntax Guide for Plugins

MIQL (**M**etric **I**nsights **Q**uery **L**anguage) is a simple query language designed for fetching and processing data. It is supported by the majority of Metric Insights' plugins.

Note!

1. Entire field names that contain special characters, aggregation and commas must be enclosed in quotes (single or double).
2. It is acceptable to enclose all fields and values in quotes.

[...] + Notation is used to signify that the MIQL parts of a statement are optional/can be repeated.

1. Building a General Query

Clause	Usage description	Sample statement	Syntax
fields (dimensions)	Adds fields and variables to the result set.	fields = text, Country, Region, number *If this clause is omitted, all available columns will be retrieved.*	fields = <Name of field or variable> [, <Name of field or variable>]+
filter	Fetches a subset of data from the database.	filter = County != 'Island'	filter = <Name of field or variable> (***Available operators are***: **== | != | < | <= | > | >=**) <'Constant value'> [AND <Filter expression>]+ filter = <Name of field or variable> is [not] null [AND <Filter expression>]+
aggregates (metrics)	Performs calculations on a set of	aggregates = count(*), sum(Population)	aggregates = (***Available functions are: sum | avg | count | min | max***) (<Name of field or variable>) [,

Clause	Usage description	Sample statement	Syntax
	numeric data values.		<Aggregation>]+
sort	Sorts data from the specified field. It is possible to apply sorting to several fields.	sort = Index DESC, Country DESC *Sorting by Ascending order is the default and may be excluded from the query.*	sort = <Name of field or variable> [ASC | DESC] [, <Sort expression>]
limit	Brings the top N rows into the result.	limit = 50	limit = <Integer>

2. MIQL Variables

MIQL allows for the creation of variables based on **source data** and **constant** values.

Variables can be used as regular fields in filters, aggregations, sorts and can be added to the results set (fields).

Use **var** keyword to declare a variable.

Syntax: **var** <variable_name> = <expression>

Type	Usage description	Example
Constant value	Uses variable value as it is passed to the query.	var a = 1 var country = "Canada"
Date	Date Variable takes from 1 to 6 parameters: **Year**, **Month**, **Day**, **Hour**, **Minutes** and **Seconds** accordingly. *For the Month param text values are allowed.* *Constant values for all fields are also*	var monthStart = date(Year, Month, 1) var newYear = date(Year, 'Jan', 1, 0, 0)

Type	Usage description	Example
	allowed.	
Date constants	Supports a number of constants that are calculated depending on the current time: 1. **LastWeek:** now() - interval 1 week *(now minus interval ONE week)* 2. **LastHour:** now() - 1 hour *(now minus ONE hour)* 3. **LastMinute:** now() - 1 minute *(now minus ONE minute)* 4. **LastMonth:** now() - 1 month *(now minus ONE month)* 5. **CurrentHour, CurrentMinute, CurrentWeek, CurrentMonth:** now()	var today = TODAY var beginOfWeek = CURRENT_WEEK
Math expression	This feature uses exp4j v0.3.11, so all operations supported in this version are available.	var ab = [a] * [b] var avr = [Total] / [Count] var val = sin([alpha]) * 2 * pi()
Math expressions (with aggregations)	Allows to do calculations on aggregated values	var aggAvr = sum(Sales) / count(Sales)

3. MIQL Parameters

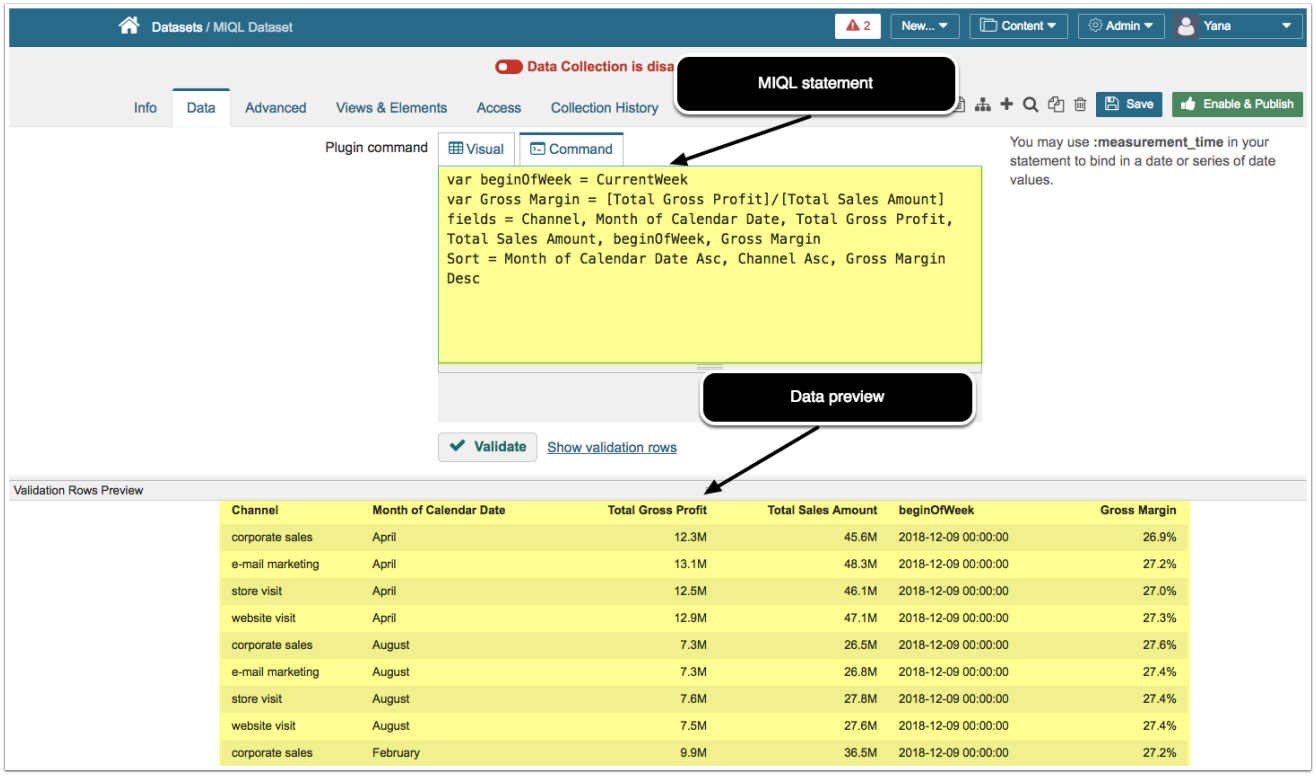
It is possible to pass connection parameters in a MIQL request.

Use **param** keyword to declare a parameter.

- *Syntax:* **param** <parameter_name> = <value>
- *Example:* **param** primary_date_format = yyyy-MM-dd'T'hh:mm

4. Example of MIQL data fetching

The example below features a query for loading data into a Dataset.



Channel	Month of Calendar Date	Total Gross Profit	Total Sales Amount	beginOfWeek	Gross Margin
corporate sales	April	12.3M	45.6M	2018-12-09 00:00:00	26.9%
e-mail marketing	April	13.1M	48.3M	2018-12-09 00:00:00	27.2%
store visit	April	12.5M	46.1M	2018-12-09 00:00:00	27.0%
website visit	April	12.9M	47.1M	2018-12-09 00:00:00	27.3%
corporate sales	August	7.3M	26.5M	2018-12-09 00:00:00	27.6%
e-mail marketing	August	7.3M	26.8M	2018-12-09 00:00:00	27.4%
store visit	August	7.6M	27.8M	2018-12-09 00:00:00	27.4%
website visit	August	7.5M	27.6M	2018-12-09 00:00:00	27.4%
corporate sales	February	9.9M	36.5M	2018-12-09 00:00:00	27.2%

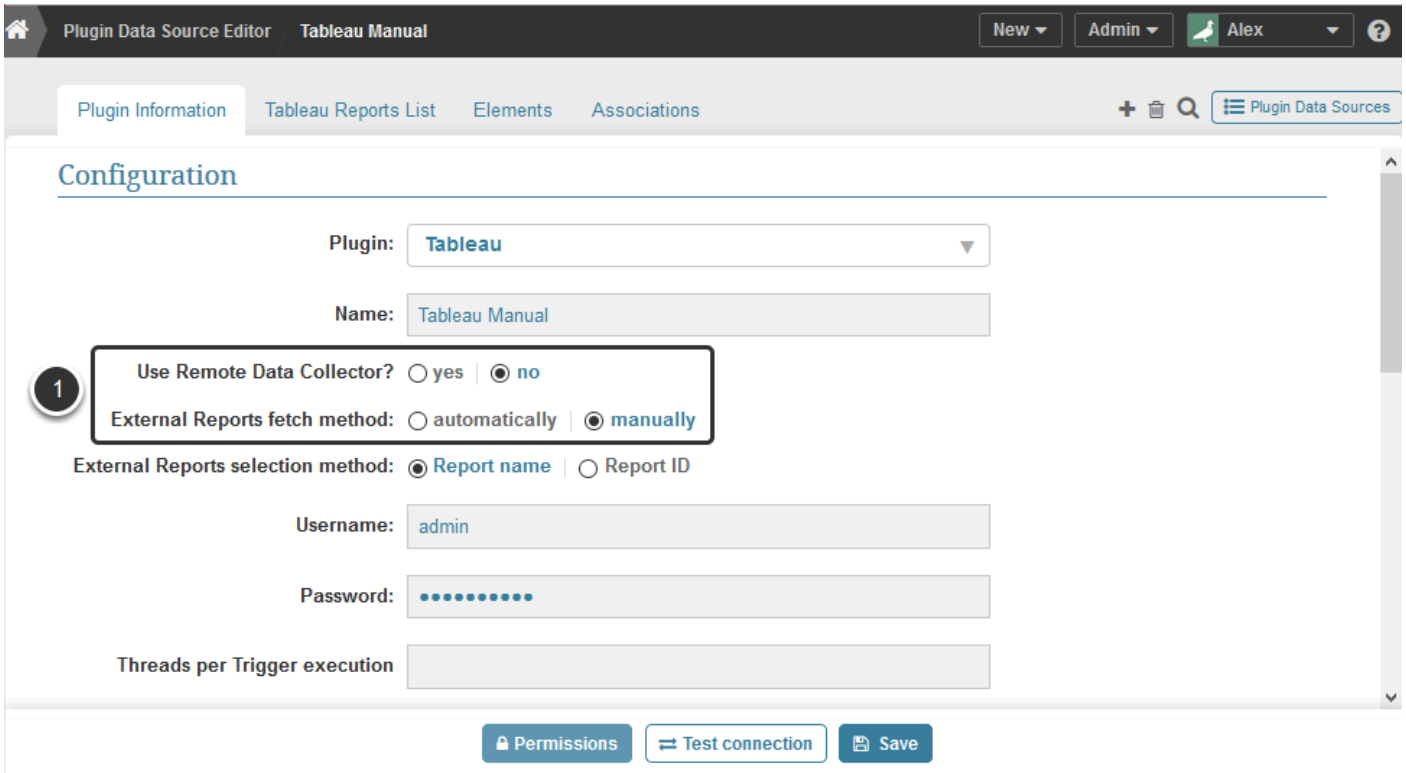
5. What's next?

See how to use MIQL when collecting Data from Google Analytics

1.10 Populate a Data Source with an uploaded List of Reports

A list of reports may be 'stored' within a plug-in Data Source after being uploaded from a file, either via CSV or with some other type.

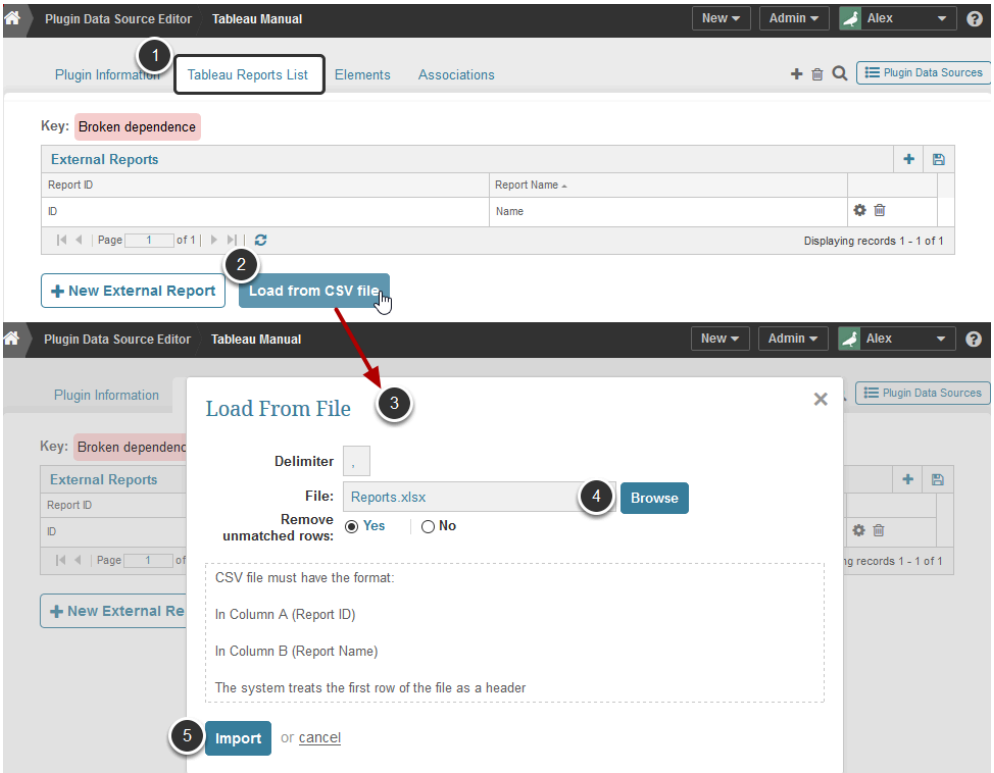
1. Create a Plug-in Data Source



1. Since we are going to upload a list of reports manually, set the **Use Remote Data Collector?** field to 'no' and the **External Reports fetch method** field to 'manually'.

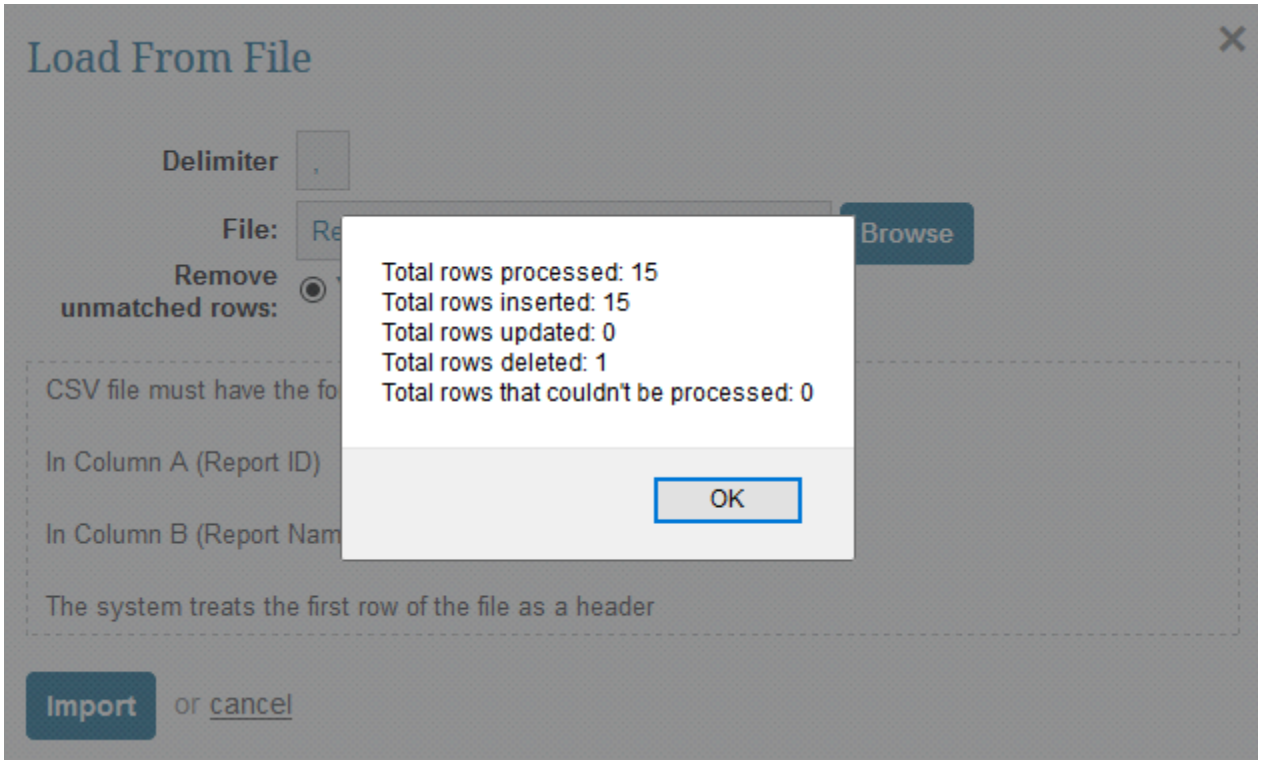
For more details on establishing connectivity to data sources, refer to [Connecting To Data Sources]() and find information for a required one from the list.

2. Upload a List as a CSV file

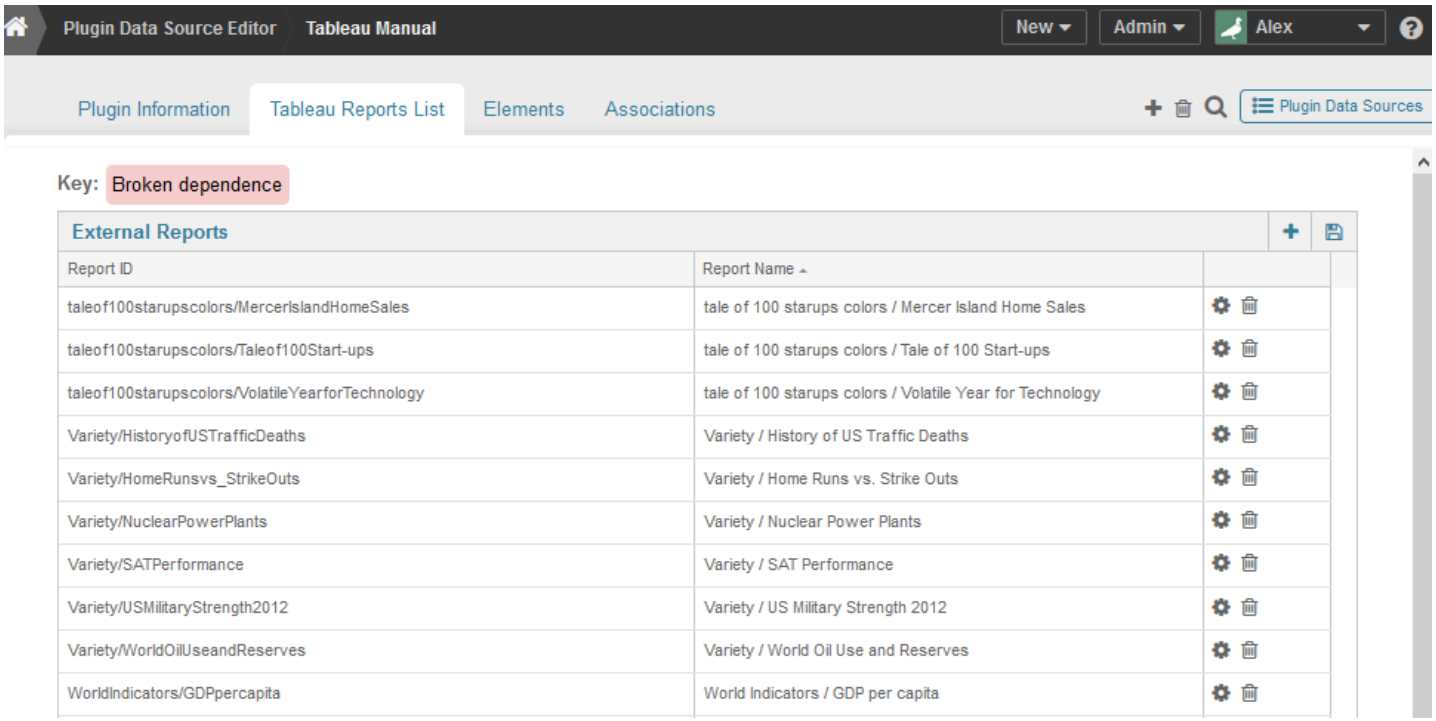


1. Open the *Tableau Reports List* tab
2. Click **Load from CSV file**
3. **The Load From File** pop-up opens
4. Click **Browse** and select a file from your local machine
5. Click **Import**
6.

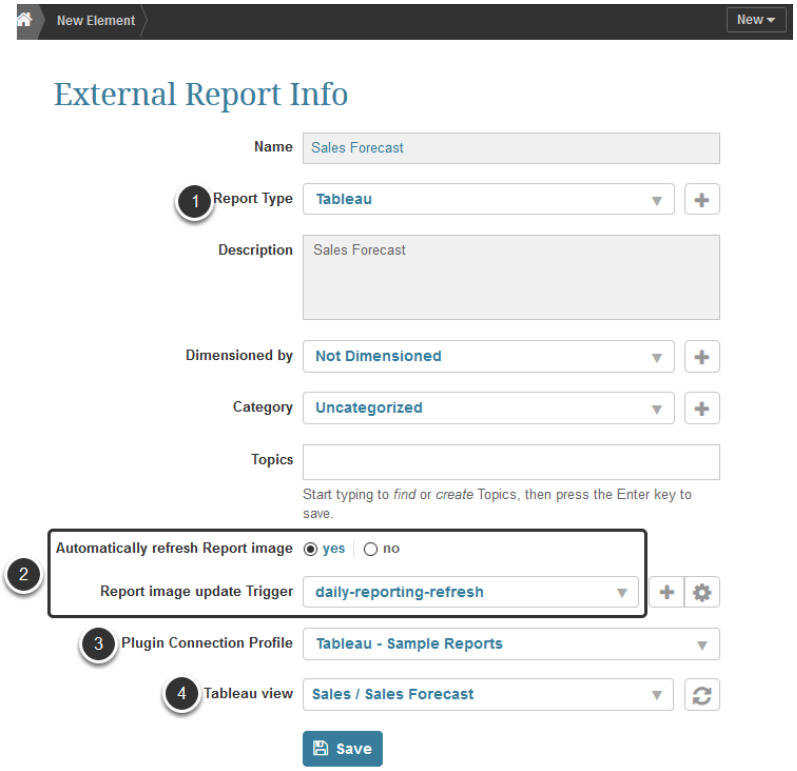
2.1. Review results of upload



2.2. Review the list of External Reports in the grid



3. Create an External Report using the Data Source

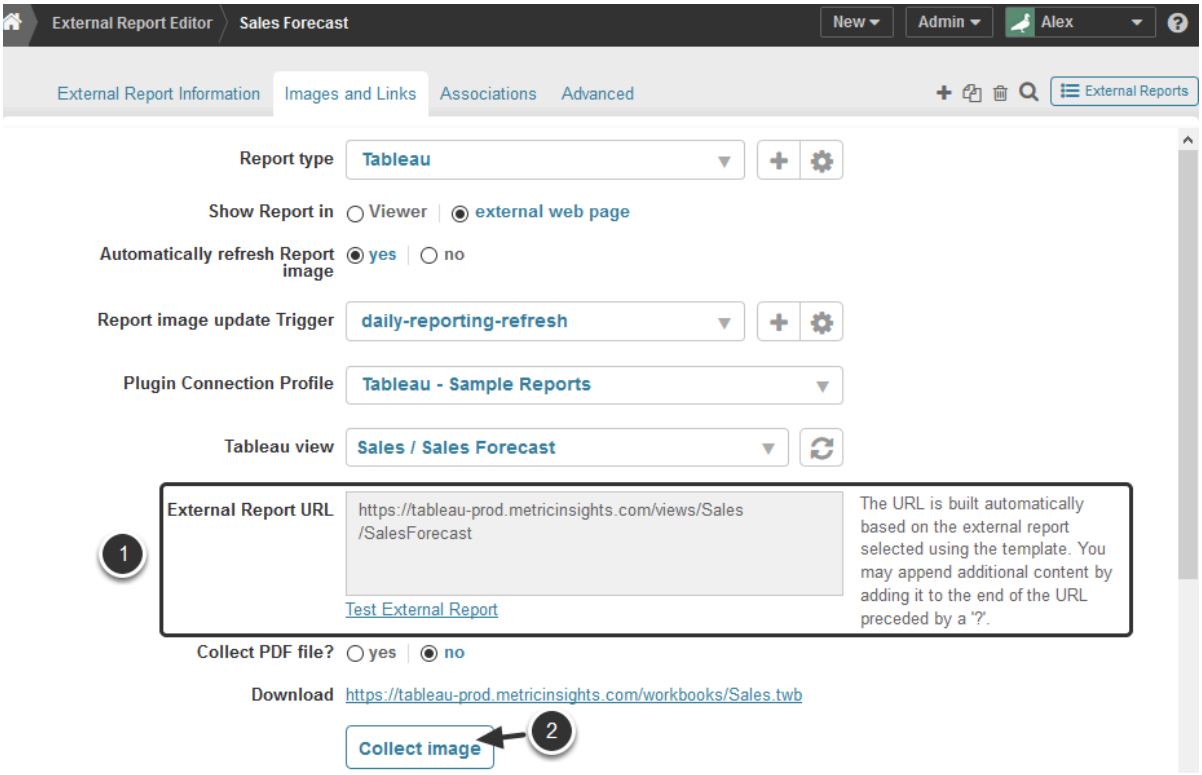


Create a new External Report, paying attention to these settings:

1. **Report Type**: Choose a type consistent with the Data Source being used (In this example, Tableau)
2. **Automatically refresh Report image:** Choose 'yes' to see the image with valid data and select the Report image update Trigger from the drop-down list in the field below
3. **Plug-in Connection Profile:** Select an option consistent with the Data Source created in Step 1 and 2
4. **Tableau view:** Pick a report that has the desired data from the list of uploaded Reports

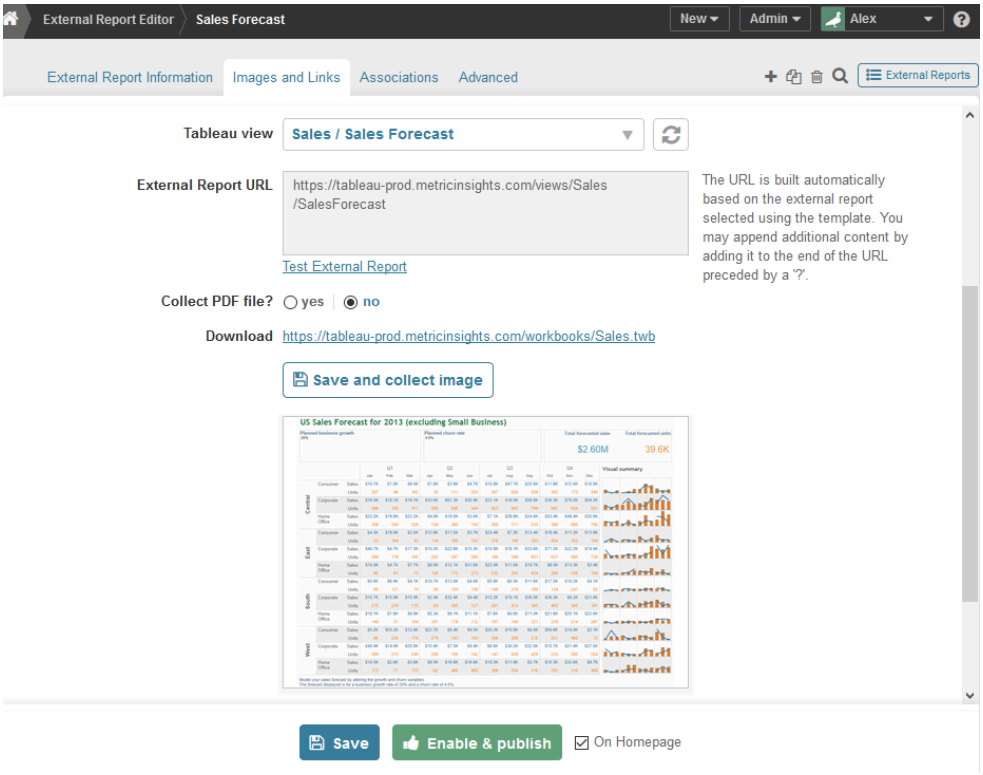
Save your entry.

3.1. External Report URL



1. Once the report is saved, the External Report URL is automatically generated.
2. Click **Collect Image** to preview the Report.

3.2. Preview the External Report



1.11 Getting data from the Metric Insights database

This article explains how to get data from the Metric Insights (dashboard) database so that it can be used for reporting usage statistics.on your Metric Insights instance.

1. Use a MySQL user named 'mi_read'

We have a number of MySQL users already set up. You can find the credentials in the file named **/etc/mi/insight.conf**. You probably want the one named **mi_read**.

2. Logon as 'root' to configure bind address

We generally configure MySQL to listen for incoming traffic from localhost only. You'll need to open it up to all comers. (MySQL does not provide a halfway measure.)

You can do that by editing the file named **/etc/mysql/conf.d/01_insight_settings.cnf** and creating the following entry:

```
bind-address            = 0.0.0.0
```

3. Adjust the permissions for 'mi_read' user

Launch mysql from command line, signing in with a mysql user who has root privileges. e.g., the 'root' user:

mysql -uroot -p

Within mysql, run these commands after replacing the <password> with the password for the mi_read user (you can get that from: /etc/mi/insight.conf) :

GRANT USAGE ON *.* TO 'mi_read'@'%' IDENTIFIED BY '<password>';

GRANT SELECT ON `dashboard`.* TO 'mi_read'@'%';

GRANT SELECT ON `dashboard\_report\_data\_segment\_%`.* TO 'mi_read'@'%';

FLUSH PRIVILEGES;

Note: if you get the following error then you will need to launch mysql with user with correct privileges:

mysql> GRANT USAGE ON . TO 'mi_read'@'%' IDENTIFIED BY 'password123';

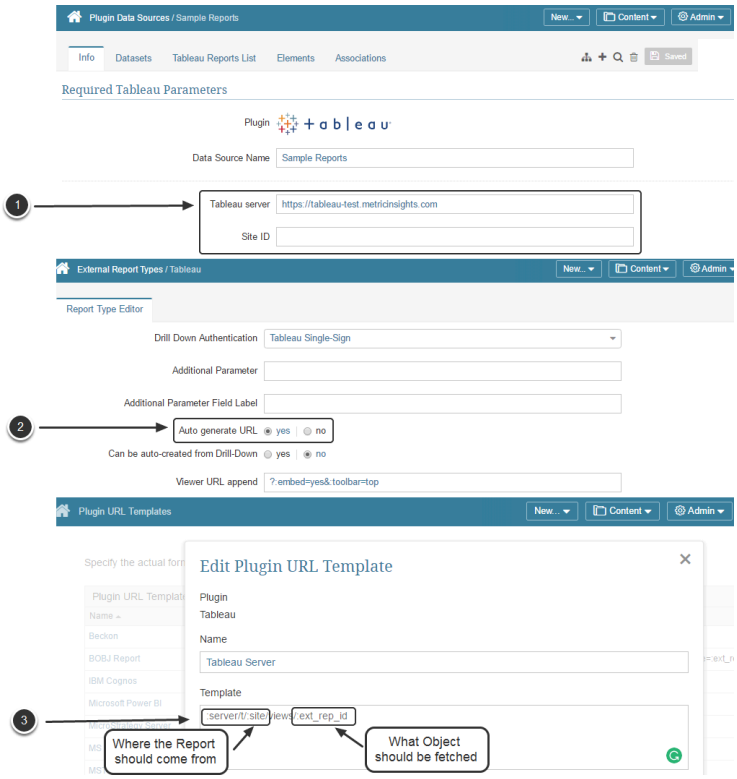
ERROR 1045 (28000): Access denied for user 'mi_setup'@'localhost' (using password: YES)

ERROR 1410 (42000): You are not allowed to create a user with GRANT

1.12 Plugin URL Editor

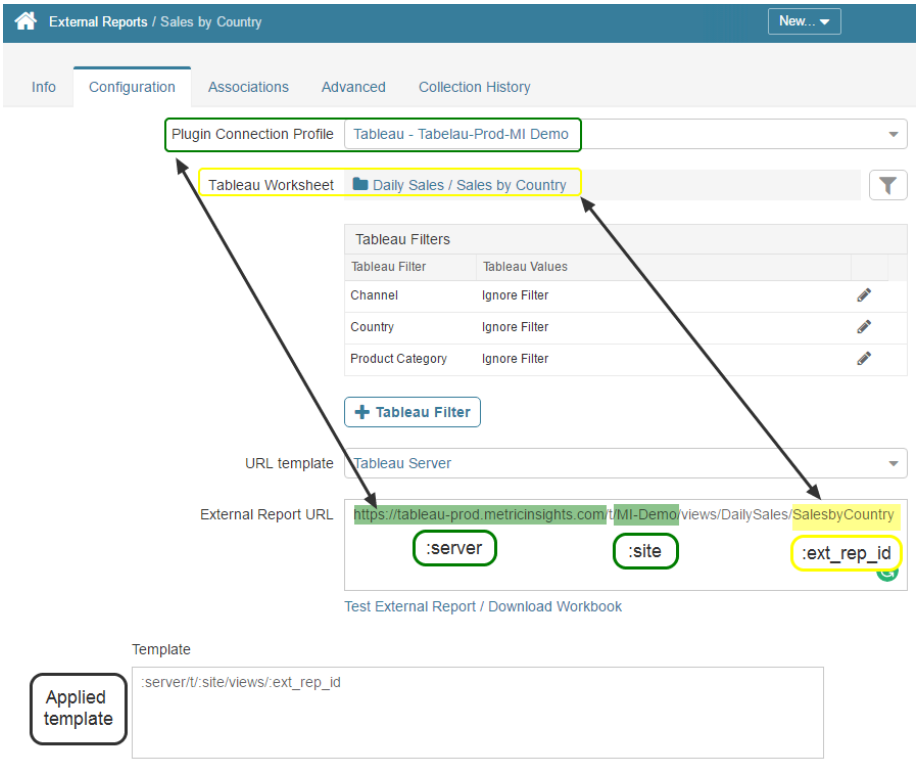
External Reports in Metric Insights fetch images from source Reports of the supported BI services.Typically, in order to create an External Report, a URL to the source Report is required. It can be set manually or be generated automatically from a template.

Purpose of Plugin URL Templates



1. *Admin > Data Sources.* When a new Plugin Connection profile is created, the system requests certain **required** Parameters. They are required because they usually point to the location of some information in the source service. One of the typical required parameters is a server or endpoint. The given example is for Tableau. Required parameters for Tableau include **Tableau server** and **Site ID**.
2. *Admin > Advanced > External Report Types.* External reports are fetched via the link to the source service. This link can be added manually or generated automatically. To enable automatic generation, go to *Admin > Advanced > External Report Types* > choose the corresponding report type or create one > set the **Auto generate URL** field to 'yes'.
3. *Admin > Advanced > Plugin URLs*. If the URL is generated automatically, it requires a template to indicate what variables should be included into the URL. Template typically includes variables related to the **location** of the data in the source system and the ID of the **object** to be fetched.

Examples of Plugin URL Template Usage



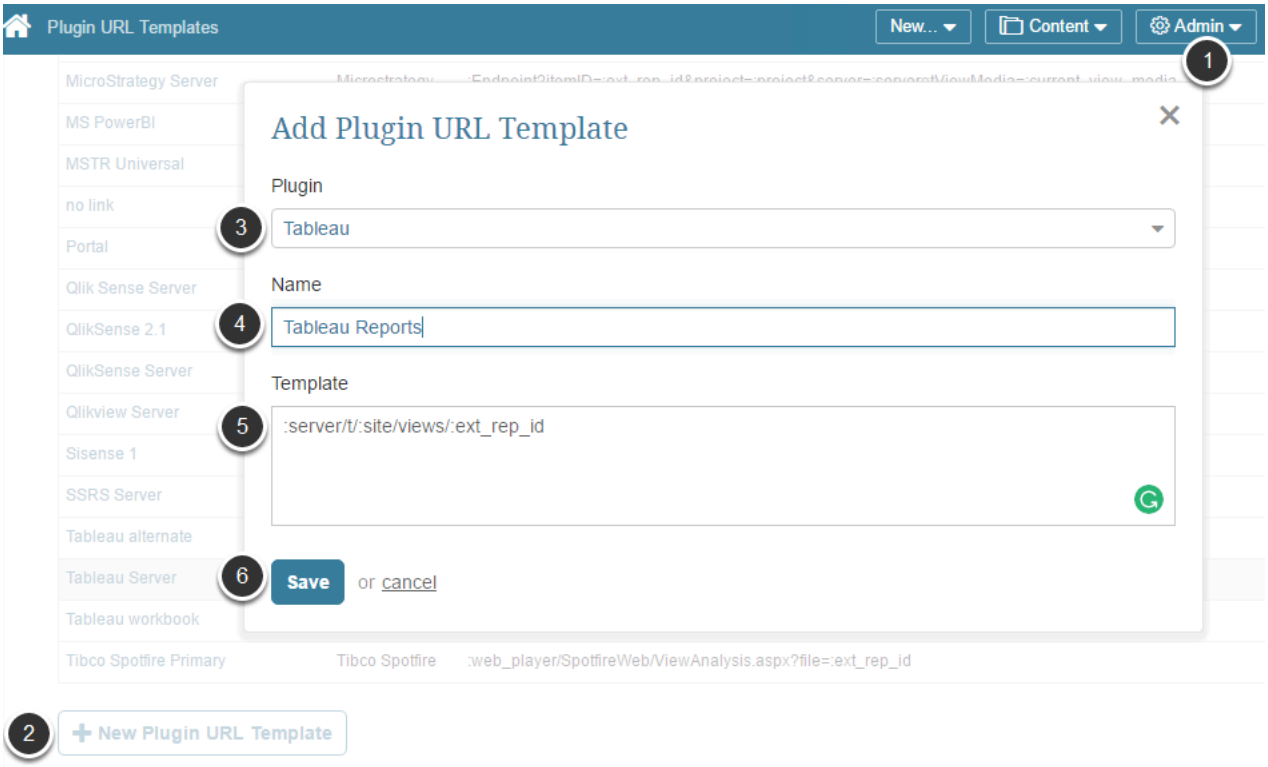
When a new External Report is created (at *New* > *External Report* > *Report Type*), the **Plugin Connection Profile** (source) and **External Report ID** (in the given example - **Tableau Worksheet**) should be selected to guide the system on what object should be fetched and where it can be found.

Here is a URL Template applied to this external Report:

```
:server/t/:site/views/:ext_rep_id
```

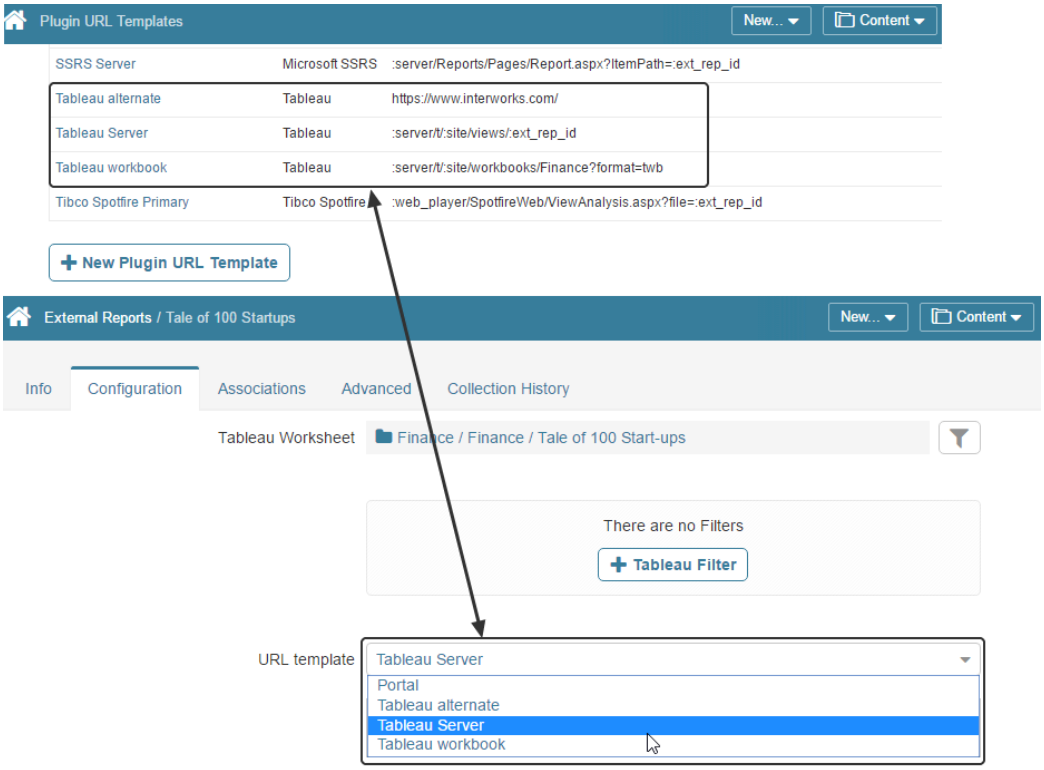
The selection in the **Plugin Connection Profile** and **Tableau Worksheet** fields provides substitutions for corresponding variables. This is how the URL is automatically generated for the External Report.

Directions for creating a Plugin URL Template



1. Access *Admin* > *Advanced* > *Plugin URLs*. The list of Plugin URL templates previously created in the system opens.
2. At the bottom of the screen click **[+ New Plugin URL Template]**.
3. Choose the **Plugin** for which this template is created from the drop-down list.
4. **Name** the template in the descriptive way. If there are multiple templates for one plugin, their purpose should be stated in their names.
5. Create a **Template**.
6. **Save** your entries.

In case of multiple Plugin URL Templates for one plugin



If more than one URL template is created for a specific Plugin, a URL template field is displayed in the External Report Editor for this Report Type.

Plugin Template Parameters for supported services

The actual variable names that can be used are dependent on the plugin data source. See the table with available variables for supported plugins.

However, you can also use :ext_report_name, and :ext_report_id that get replaced with corresponding values in the **Reports List** for the given external report.

Plugin Name	Required Parameter Name in UI	Plugin Variable Name in the URL Template
Adobe Analytics	Web Service Shared Secret	wssecret
	Web Service User Name	wsusername
Atlassian Confluence	Domain	domain
Atlassian Jira	Endpoint	endpoint
Beckon	Account URL	accountUrl

	API Key	apiKey
CSV	Root location	rootLocation
Druid	Coordinator node	server
Elasticsearch	Endpoint	endpoint
	Index	index
Flat File	column_delimiter	column_delimiter
	discard_directory	discard_directory
	files_contain_headers	files_contain_headers
	source_file_directory	source_file_directory
Google AdWords	Client Customer ID	client_customer_id
	Developer Token	developer_token
	token	token
Google Analytics	Profile ID	profile
	token	token
Google BigQuery	job_id_prefix	job_id_prefix
	token	token
Google Calendar	token	token
Google Spreadsheet	token	token
Graphite	host	host
Hadoop Pig	fs.default.name	fs.default.name
	hadoop.tmp.dir	hadoop.tmp.dir
	mapred.job.tracker	mapred.job.tracker
	mode	mode

Hubspot	token	token
IBM Cognos	Cognos URL	dispatcher_url
	Namespace	namespace
	Server	server
IBM Coremetrics	Client ID	client_id
Microsoft SSRS	Report Server URL	report_server_url
	server	server
MicroStrategy	Endpoint	endpoint
	Project	project
Mixpanel	API Key	api_key
	API Secret	api_secret
MongoDB	Database Name	dbname
	Path to mongo	mongo.executable.path
OLAP	uri	uri
Qlik Sense	Click through server	srv_url
	server	server
	User domain	user_domain
QlikView	Click through server	srv_url
	server	server
RSS	RSS Feed Url	url
Salesforce	token	token
Salesforce SOQL	token	token
SAP Business Objects	auth.type	auth.type

	cms	cms
Script	Script to run (absolute path)	executable
Sisense	Server	server
Splunk	host	host
	port	port
Tableau	Site ID	site
	Tableau server	server
Treasure Data	apiKey	apiKey
	database	database
Zendesk	domain	domain

1.13 Handling Date/Time columns for various Plug-ins

1. How Metric Insights determines which columns are Date/Time types

For various plugins such as for Tableau, QlikView, Qlik Sense, MicroStrategy and others, Metric Insights pulls data from those sources via CSV export. This means that Metric Insights must infer the data types for each column in the CSV. It does not have any type information in the CSV; it only has the data. So Metric Insights looks at all the values and infers the data type.

In most of the apps such as Tableau, QlikView, Qlik Sense you can manually export the CSV and assume that that is how Metric Insights will be getting the data. A few exceptions do occur.

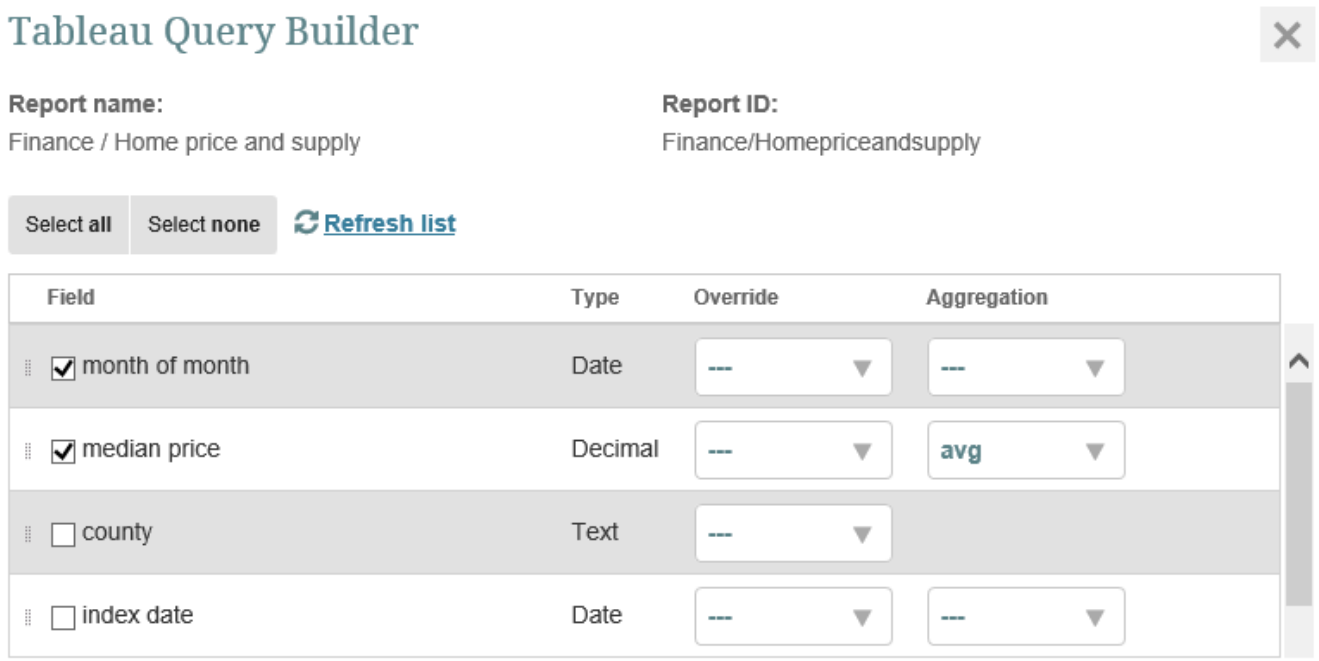
For example, with Tableau you can use the UI controls in Tableau Server to export and download a CSV. However, Metric Insights pulls the data via the **view url** of Tableau Server using CSV export. E.g.,

```
https://tableau.example.com/views/workbookName/sheetName?format=csv
```

Metric Insights then infers the data type of each column in the exported CSV by interrogating the values. For Metric Insights to interpret a column as containing date/time values, it needs to see that the values conform to a particular date/time format.

Note: if some of the values in a column are in date/time format, but other values are not, then Metric Insights will type the whole column as a Text field rather than a Date field. To prevent this from happening you will need to ensure that all values in the column are of date/time format.

2. For plug-ins that use the Query Builder



Most of the plugins such as Tableau, QlikView, Qlik Sense, MicroStrategy use the Query Builder.

Use the Visual Editor link in the Element Editor (Metric or Report) to bring up the Query Builder. The Query Builder screen shows the column interpreted as date/time with Type DATE.

3. The various date/time formats that Metric Insights looks for

This is just a sample list of the date/time formats that Metric Insights looks for in the CSV file exported from Tableau Server.

```
"yyyy-MMMMM-dd HH:mm:ss",// 2013-August-01 01:23:45
"yyyy-MM-dd HH:mm:ss",  // 2012-01-01 01:23:45
"MM-dd-yyyy",           // 01-01-2012
"yyyy-MM-dd",           // 2012-01-01
"yyyy-MM",              // 2012-01 but not 2012-2013
"M/d/yyyy h:m:s a",     // "4/15/2013 1:00:00 AM" for hourly or minute Tableau
"MM/dd/yyyy",           // 01/01/2012
"MM/dd/yyyy HH:mm:ss",  // 01/01/2012 01:23:45
"MMMMM d, yyyy h:mm a", // "April 15, 2013 1:00 AM" for minute Tableau
"MMMMM d, yyyy h a",    // "April 15, 2013 1 AM", "April 15, 2013 12 PM" for hourly
Tableau
"MMMMM dd, yyyy",       // January 01, 2012
```

```
"MMMMM dd,yyyy",
"dd MMM yyyy",                          // "05 Nov 2013" for Tableau
"MMMMM, yyyy",          // January, 2012   --- also catches January, 12 -> January 0012
"MMMMM,yyyy",
"MMMMM yyyy",           // January 2012    --- also catches January 12 -> January 0012
"MMM, yyyy",            // Jan, 2012       --- also catches Jan, 12 -> Jan, 0012
"MMM,yyyy",
"MMM yyyy",             // Jan 2012        --- also catches Jan 12 -> Jan 0012
"yyyy-MMM-dd",          // 2012-FEB-01     Oracle style. 12-FEB-12 -> Jan 12, 0012
"EEE MMM dd HH:mm:ss z yyyy", // Wed Aug 21 15:50:21 UTC 2013
```

3.1. Metric Insights needs at a minimum the Year and Month for a Date/Time field

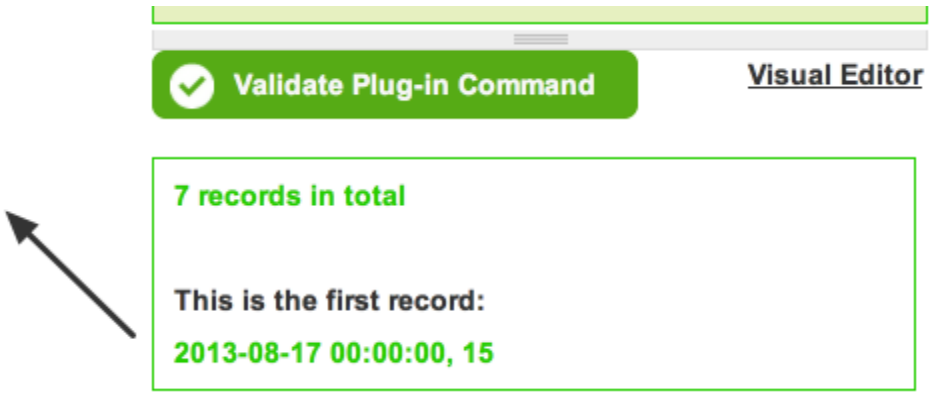
Metric Insights requires at a minimum a Year and a Month in the Date/Time field. In the previous list of formats you can see that Year and Month are required. The following is a list of the date/time parts that are needed as you expand into days, hours and minutes.

1. Year, Month
2. Year, Month, Day
3. Year, Month, Day, Hour
4. Year, Month, Day, Hour, Minute

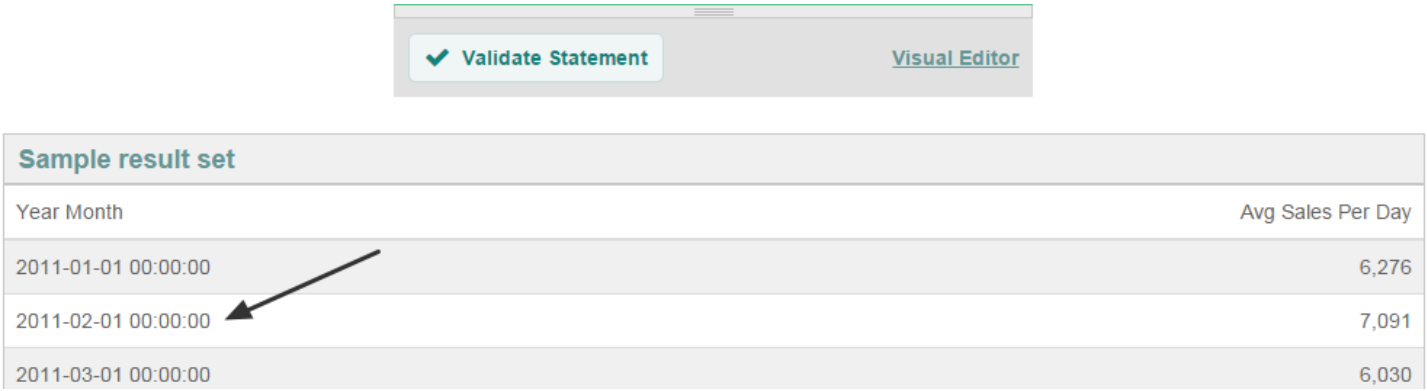
4. How you know that Metric Insights has recognized the Date/Time correctly?

When you use the button to **Validate** your plug-in command query in the editors for either a Metric or a Report, a sample result set is displayed and columns that are interpreted as date/time will display in YYYY-MM-DD hh:mm:ss format. For example:

4.1. Date/Time recognized in Metric



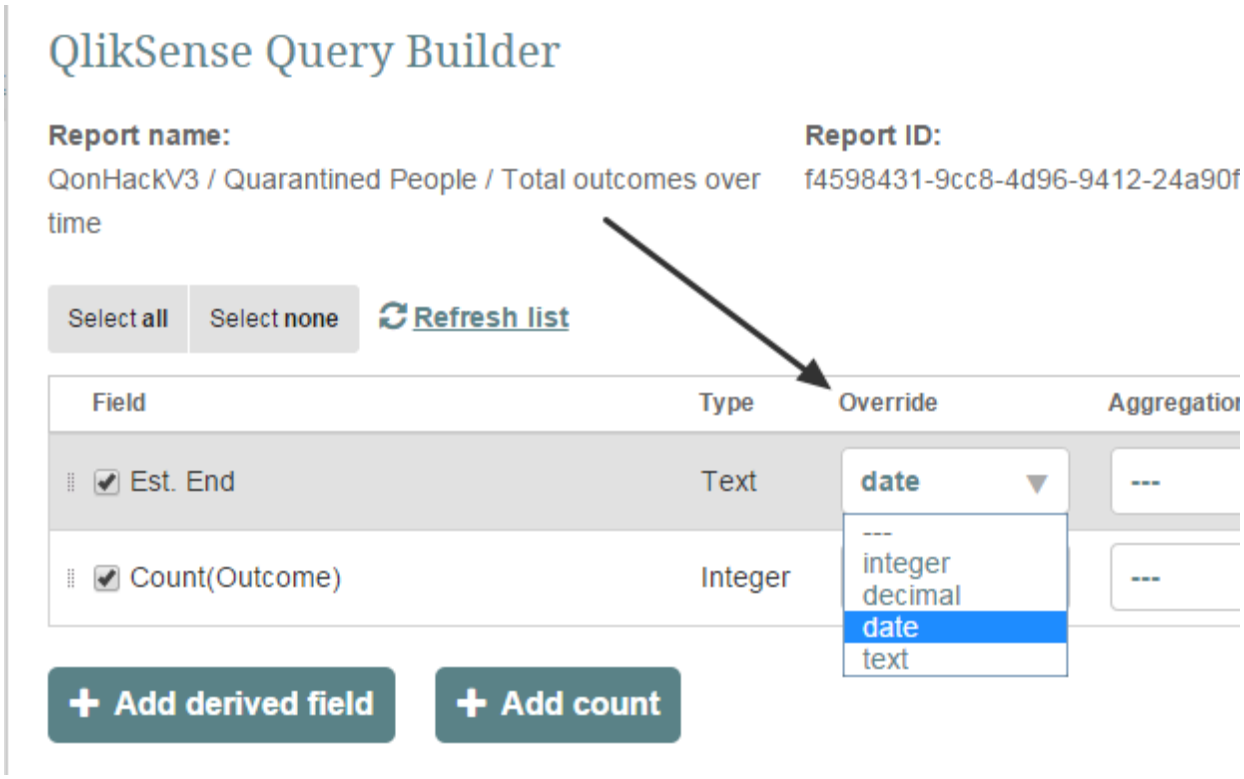
4.2. Date/Time recognized in Report



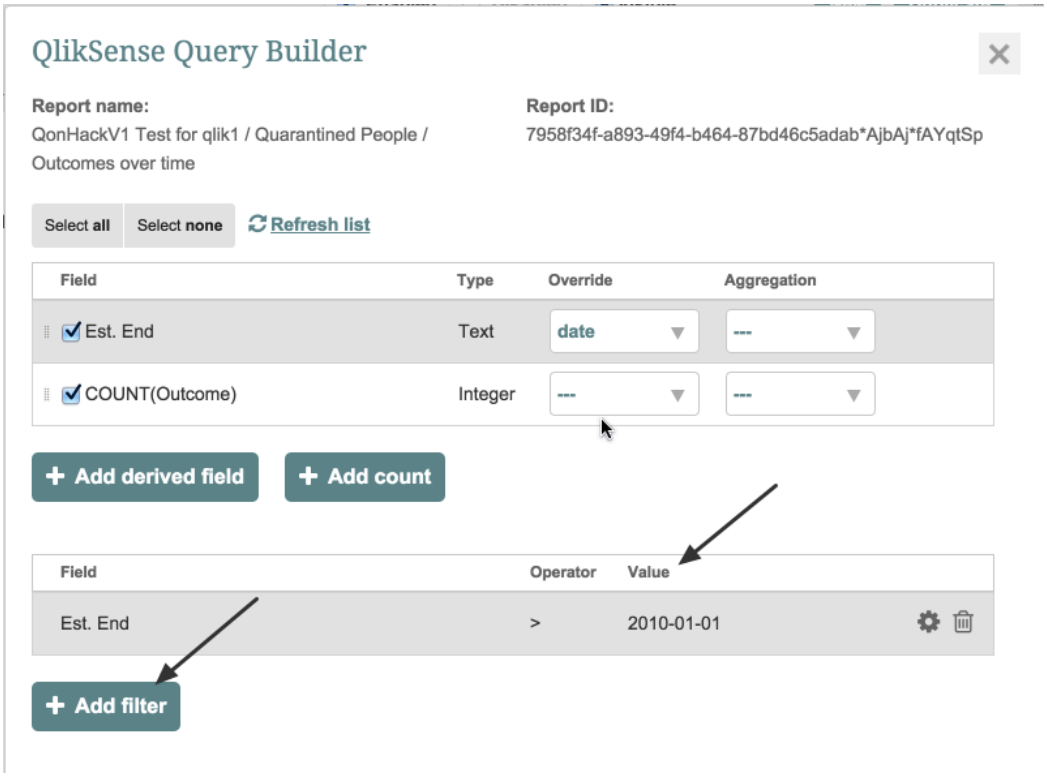
Use the button to Validate your Tableau plug-in command query in the Report editor, and the sample returned data shows the column interpreted as date/time in YYYY-MM-DD hh:mm:ss format

5. If Metric Insights is not able to determine your date field as valid, then use these options to coerce

5.1. Use the override feature in Query Builder



5.2. Filter out rows that are not Date/Time



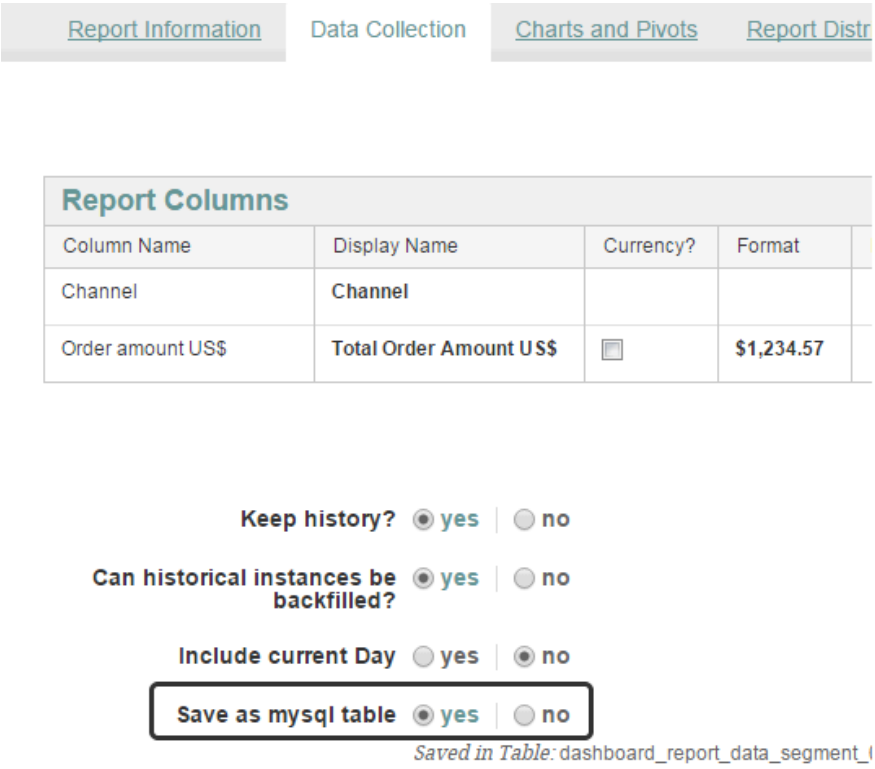
If your result set is returning empty values:

In Query Builder use the **Add filter** button to create a condition

5.3. Construct a Composite Date/Time field in the plug-in request

If your date/time value is spread over several columns, then you can build a Composite date/time column. In the above example, use the control in the Visual Editor to construct a Composite date from the individual year, month, day columns in the your view.

5.4. Use an Intermediate Report in Metric Insights, manipulate the Date/Time there



You can create a Report from the your plug-in , and then create other Metrics (or Reports) from this newly created Report. Be sure to save the report as a **myslq SQL table.**
Then you can simply write SQL against the report and apply any SQL functions you want for converting or formatting data from the report into date/time columns as appropriate. See the help documents on making an Existing Report.

5.5. Make the changes in the BI app that sources the data to Metric Insights

Sometimes the best way to fix the format of the dates is to make changes in the Business Intelligence app that sources the data.

For example, with Tableau, publish your worksheet so that the Date/Time columns conform to one of the date/time formats that Metric Insights understands.

This might be an iterative approach, but based on the examples provided earlier in this article you can format your column in Tableau desktop as one of the recognized date/time formats and publish to Tableau server.

1.14 Supported Date Formats

Below is a list of Date Formats that are supported in Metric Insights.

Dates falling outside of the range [**1900-01-01 00:00:00 .. 9999-01-01 00:00:00**] will not be processed.

Date Format	Example
yyyy	2019
yyyy-MM	2019-01
yyyy-MMM	2019-DEC
yyyy-MM-dd	2019-01-01
yyyy-MMM-dd	2019-FEB-01
yyyy-MMMMM-dd HH:mm:ss	2019-August-01 01:23:45
yyyy-MM-dd'T'HH:mm:ss	2019-03-10T01:34:26
yyyy-MM-dd'T'HH:mm:ss.SSS	2019-05-26T11:34:14.000
MMM, yyyy	Jan, 2019
MMMMM, yyyy	January, 2019
MM/dd/yy	01/01/19
MM-dd-yyyy	01-01-2019
MM/dd/yyyy HH	01/01/2019 01
MM/dd/yyyy HH:mm	01/01/2019 01:23
MM/dd/yyyy HH:mm:ss	01/01/2019 01:23:45
MMMMM d, yyyy h a	April 15, 2019 1 AM

MMMMM d, yyyy h:mm a	April 15, 2019 1:00 AM
M/d/yyyy h:m:s a	4/15/2019 1:00:00 AM
dd/mm/yy	21/01/19
dd/mm/yyyy	21/01/2019
dd.MM.yyyy	31.03.2019
dd MMM yyyy	05 Nov 2019
dd-MMM-yy	4-Jan-19
dd-MMM-yyyy	17-SEP-2019
EEE MMM dd HH:mm:ss z yyyy	Wed Aug 21 15:50:21 UTC 2019

1.15 Content Auto Synchronization

Note *content autosync is currently only possible with Tableau. Support for other tools will be added in upcoming releases.*

As of Version 6.2, Metric Insights includes the ability to Auto Synchronize content with BI tools. This functionality automates creation of External Reports from the BI tool objects and simplifies the process of updating existing External Reports.

This article describes how to:

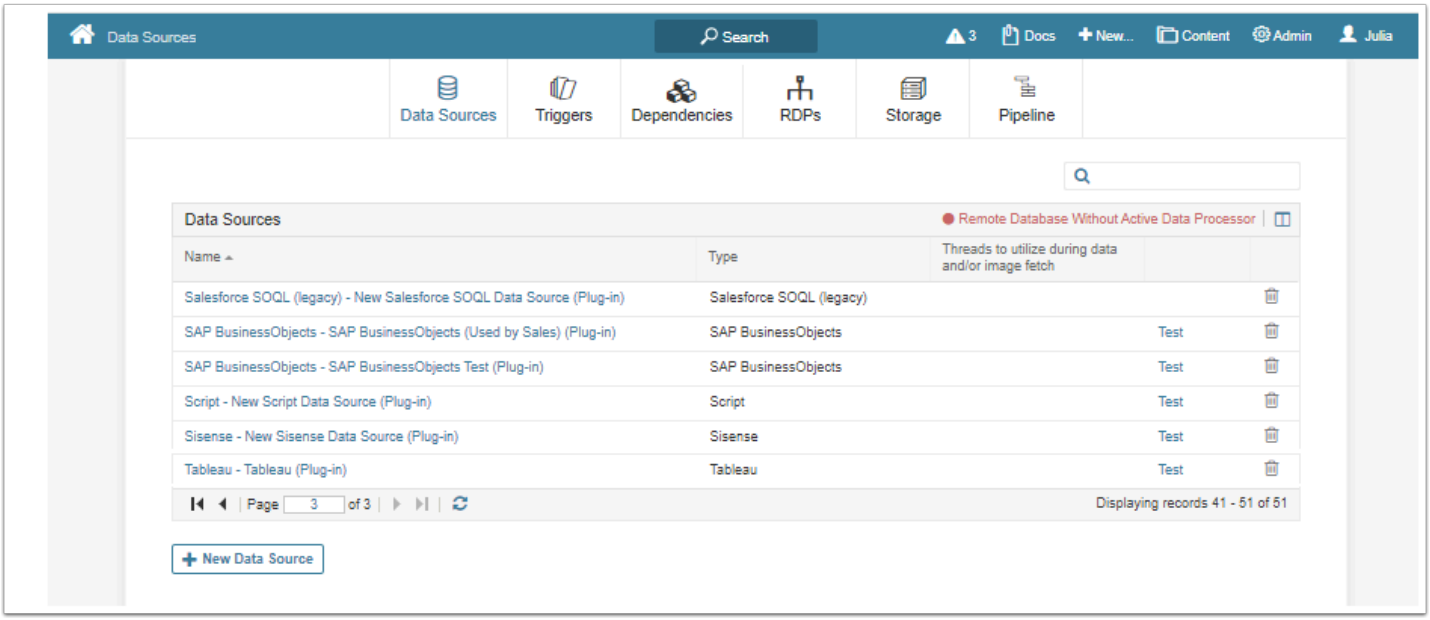
- Enable Auto Synchronization
- View Created External Reports
- Delete Created External Reports

Prerequisites:

- Created Data Source
- Created Category
- Created External Report Template

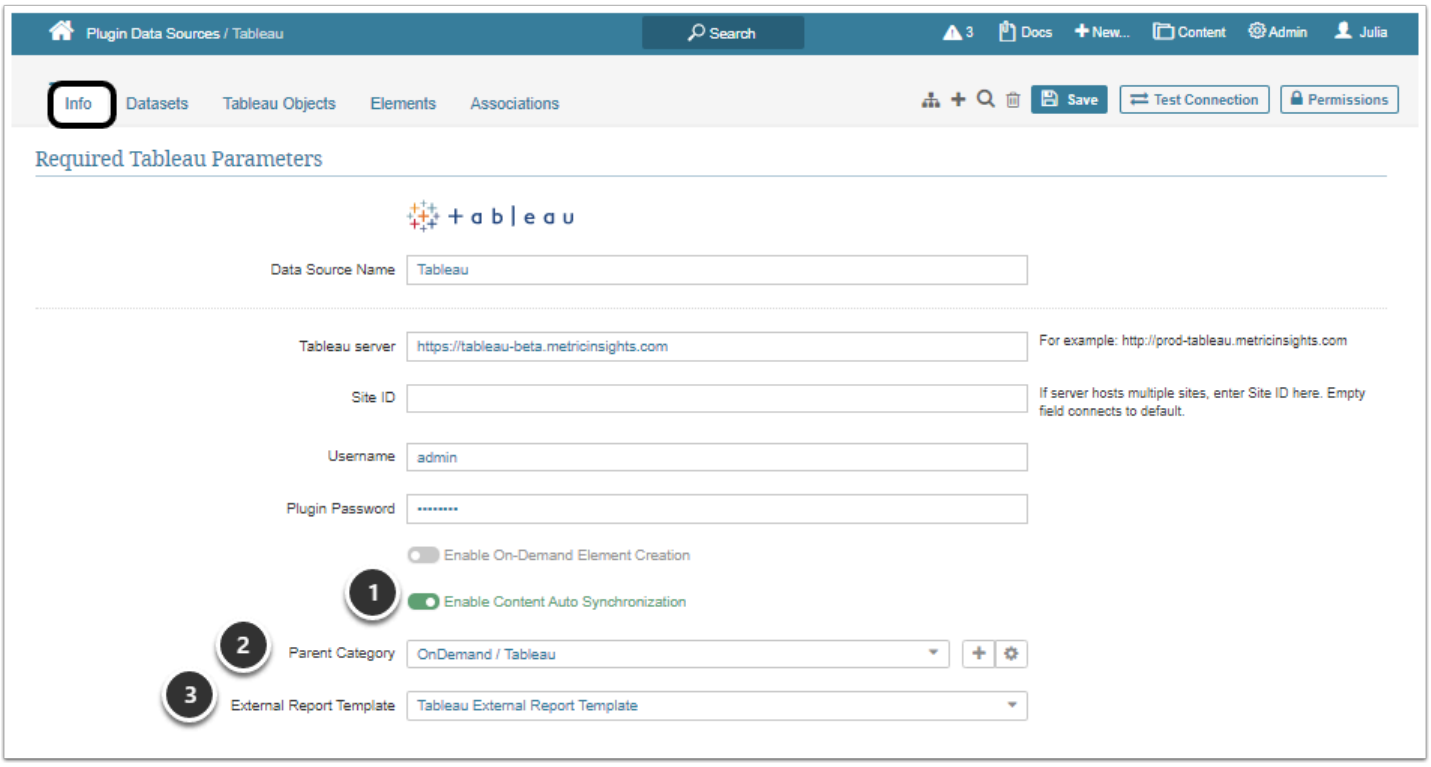
Enable Auto Synchronization

1. [Admin] > [Collection & Storage] > [Data Source]



In Data Sources, select a Plugin to access its editor.

2. Assign Category and External Report Template

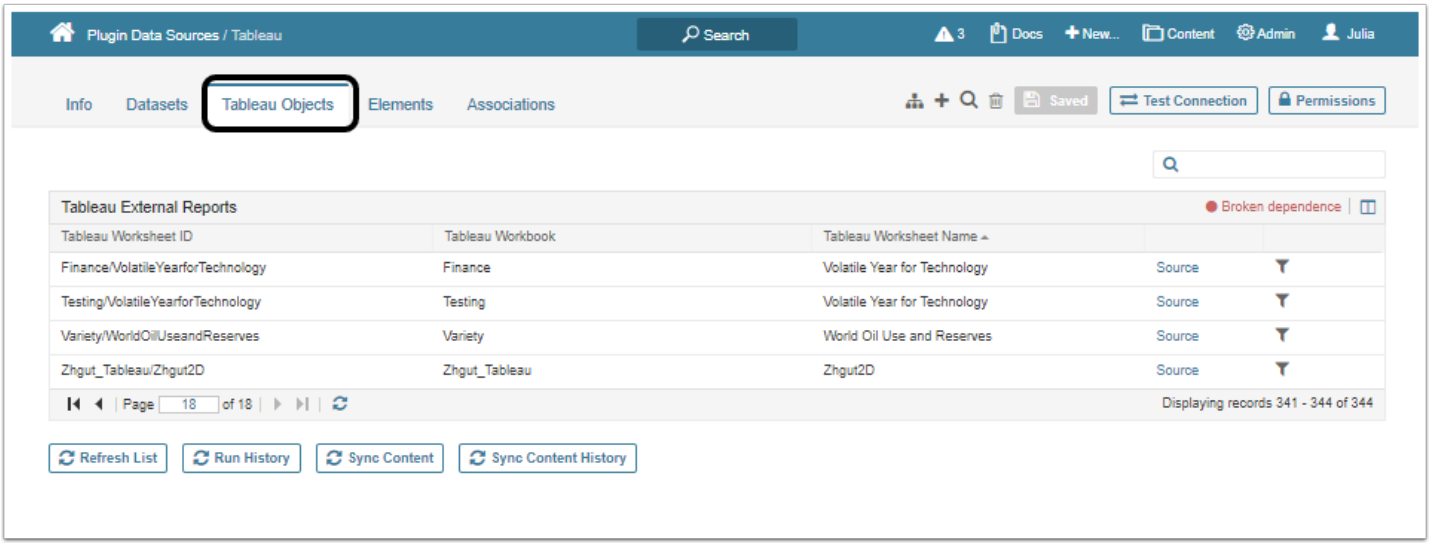


To create External Reports from the BI tool system:

1. Choose "Enable Content Synchronization" to display the next two buttons
2. Select the Category where External Reports are synced
3. Select an "External Report Template"

[Save]

3. Synchronize Content

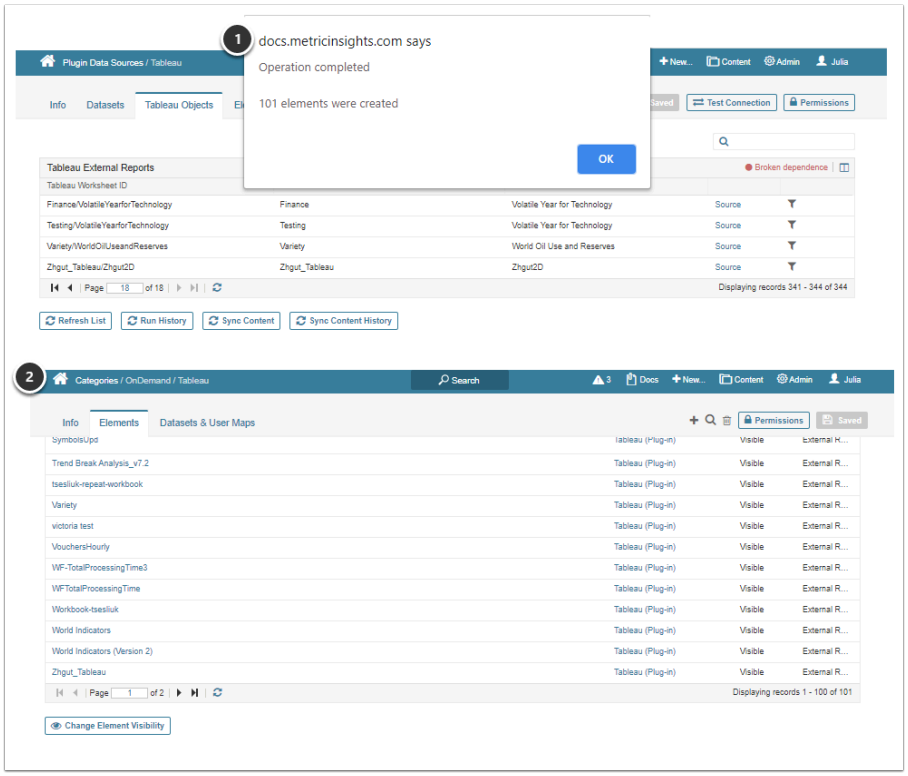


In [Plugin Data Source] Objects tab:

- [Sync Content] synchronizes data from the BI tool and creates External Reports placing them into the selected Category.
- If content is updated on the BI tool, select [Refresh List] to update the above grid and then [Sync Content] to create new External Reports and/or update existing External Reports.

⚠ Each new automatically created External Report is named after the object of its origin.

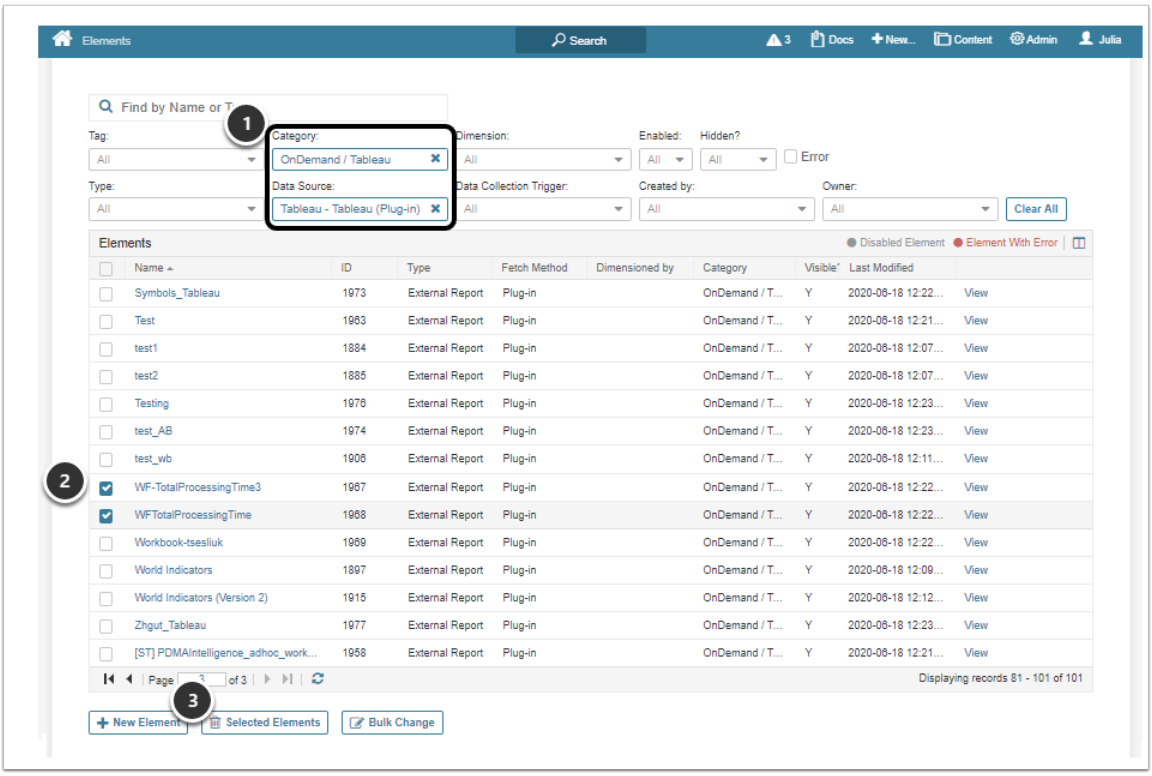
View Synced External Reports



Once synchronization completes:

1. The popup displays the number of generated External Reports
2. Synced External Reports are available in their assigned Category: Go to [Content] > [Categories]

Delete Synced External Reports



To delete synced External Reports:

1. In [Content] > [Elements], choose the desired Category
2. Select External Reports to be deleted
3. [Selected Elements]

1.16 Create External Report Template

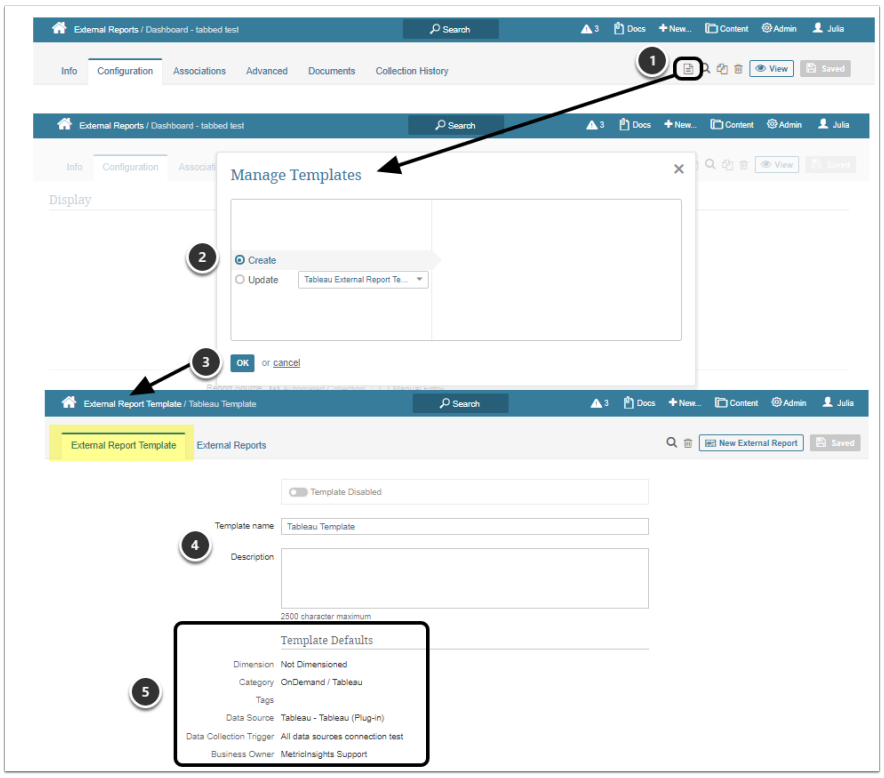
External Report Templates set defaults values to be used when creating External Reports via Autosync.

For more information see Content Auto Synchronization

Prerequisites: an External Report that will define the fields defaulted when using the Autosync funtion to create/update External Reports.

1. Create a new Template

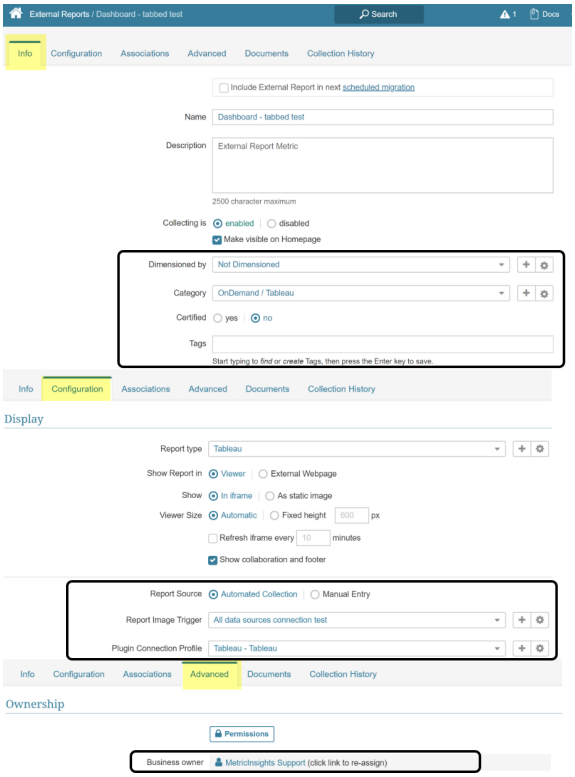
Edit the External Report that you want to use for the Template defaults



To create new External Report **Template**, go to [Content] > [Elements] and select External Report, settings you want to be used as a template for new External Reports:

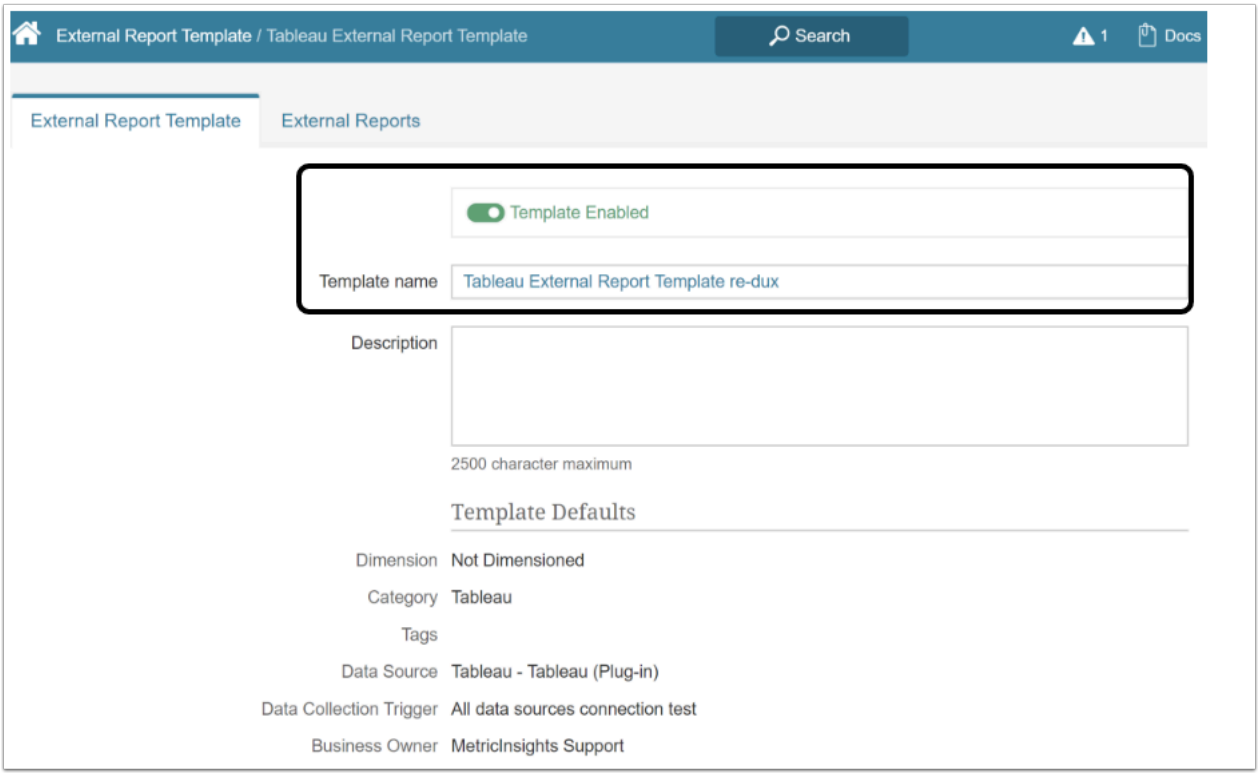
1. Select [template] icon
2. Check [Create]
3. Click [OK] to open the Template Editor
4. Only the Template name and Description are editable
5. All of other Template Defaults are copied from chosen External Report

1.1. Fields defaulted from External Report by tab



2. Updating an External Report Template

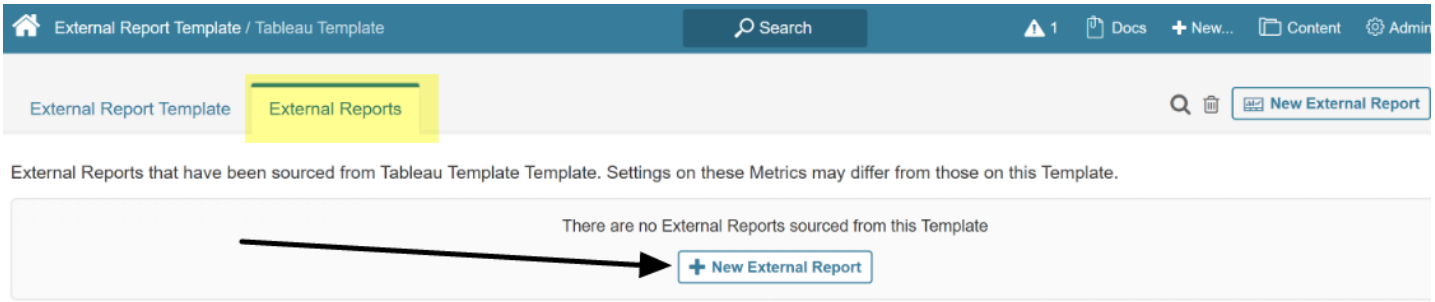
List of existing External Report Templates is available on [Admin] > [Reference Objects] > [Object Templates] > [External Report Templates].



The only fields available for change are:

- Template Enabled (or Disabled)
- Template name

3. Create new External Report from Template editor



Selecting [New External Report] will transfer you to the External Report Editor with the template defaults applied. All fields are open to change.

Info | Configuration | Associations | Advanced | Documents | Collection History

Display

Report type: Tableau

Show Report in: Viewer | External Webpage

Show: In iframe | As static image

Viewer Size: Automatic | Fixed height 600 px

Refresh iframe every 10 minutes

Show collaboration and footer

Report Source: Automated Collection | Manual Entry

Report Image Trigger: All data sources connection test

Plugin Connection Profile: Tableau - Tableau

2. Storing Data

2.1 Storing data in Microsoft SQL Server 2016+

⚠ **Data Storage** is only available in version **6.1+**

Data Storage in 6.0+ allows a Metric Insights server to connect to existing enterprise database infrastructure.

PREREQUISITES:

- Microsoft SQL Server must be installed and configured for use
- SQL Server User initializing the Data Storage must have relevant database permissions

This article describes how to initialize Data Storage with Microsoft SQL Server. The process of **connecting to MariaDB/MySQL** is essentially the same.

Grant Database Permissions to a Microsoft SQL User

Data Storage initialization can be performed by any SQL Server User having the corresponding Permissions to manipulate a database (including DDL commands):

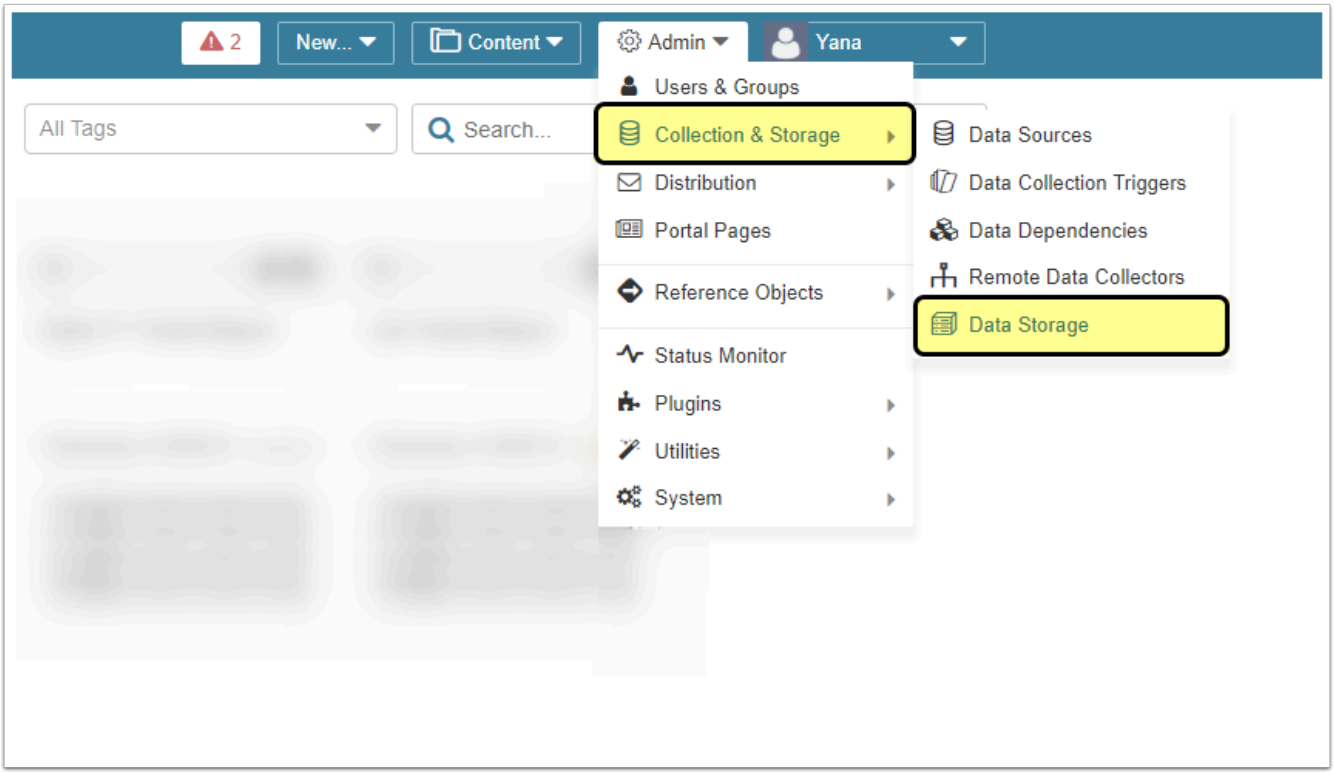
- CREATE
- ALTER
- DROP
- RENAME
- INSERT
- UPDATE
- DELETE
- SELECT

We recommend creating a separate SQL Server User (service account) to work with the assigned Data Storage. See details below.

```
USE master
CREATE LOGIN <username> WITH PASSWORD = '<password>';
```

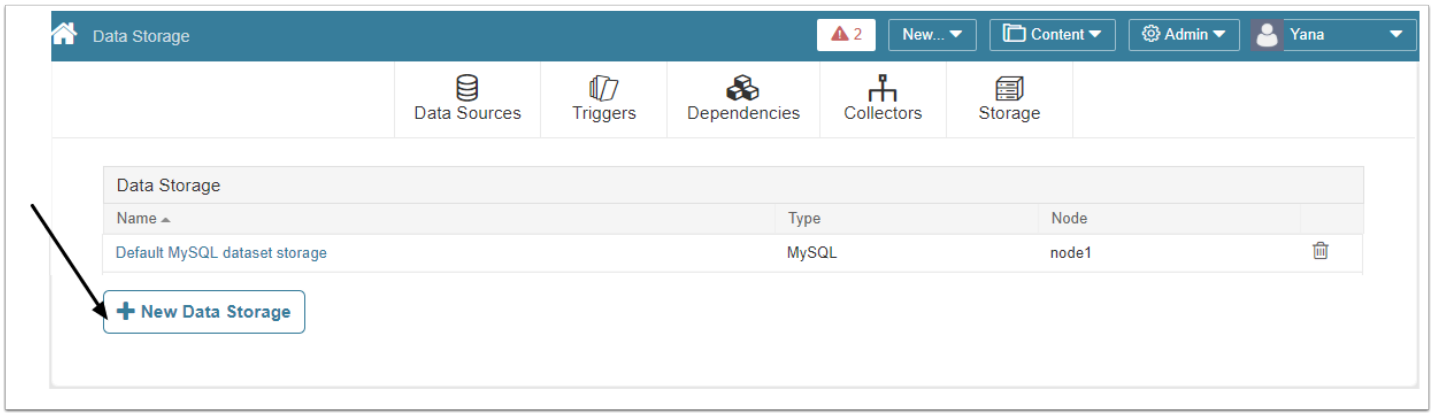
```
CREATE DATABASE <storage name>;
USE <storage name>;
CREATE USER <username>;
CREATE ROLE <storage role>;
EXEC sp_addrolemember '<storage role>', '<username>'
GRANT CREATE,ALTER,DROP,RENAME,INSERT,UPDATE,DELETE,SELECT ON DATABASE::<storage name>
TO <storage role>;
```

1. Admin > Collection & Storage > Data Storage



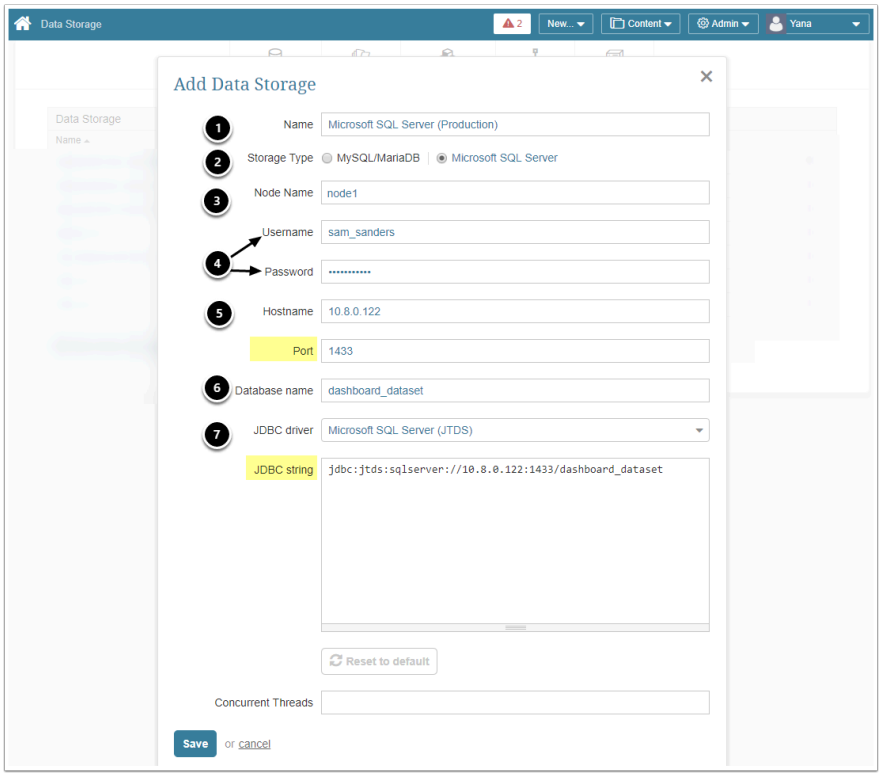
From the **Admin menu,** access the **Data Storage** option.

2. Data Storage List > [+ New Data Storage]



Add a **New Data Storage** from the UI.

3. Configure Connection Parameters



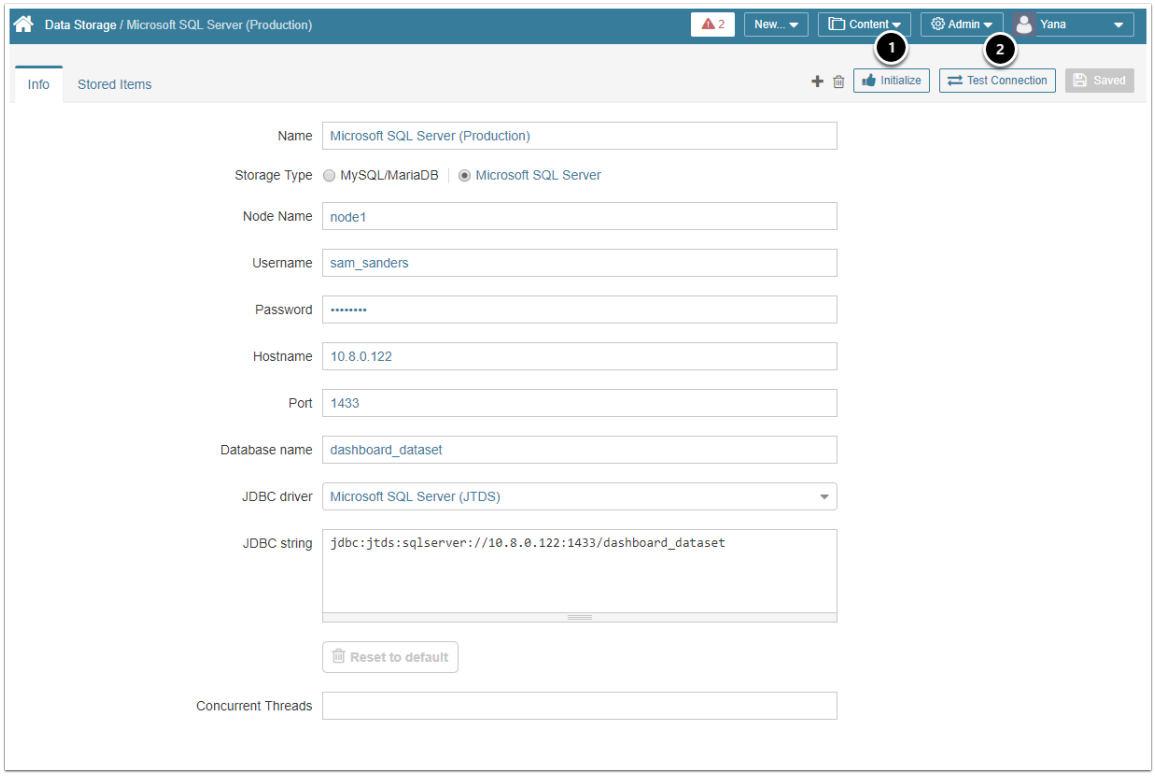
1. **Name:** provide the name of the Data Storage profile
   - This name will be used internally in the Metric Insights UI and can differ from the database name
2. **Storage Type:** choose one of the supported options. In this case, it is Microsoft SQL Server
3. **Node Name:** leave as "node1"
   - **node1** is the default name of your node in MI Version 6.0.

- Multiple node options will be added future releases

4. **Username/Password:** credentials for accessing the Microsoft SQL Server
5. **Hostname:** name or IP of the database host
6. **Database name:** MS SQL Server database name where you will store your Metric Insights datasets
7. **JDBC driver:** from the dropdown, choose the driver that will be used to interact with the database
   - After the JDBC driver has been selected, the **Port** and the **JDBC string** will be provided automatically

Click **[Save]** to proceed

4. Initialize your Data Storage and Test Connection



1. **Initializing** will configure and setup the Data Storage for use.
2. Click **[Test Connection]** to verify that the Data Storage profile can communicate with the database

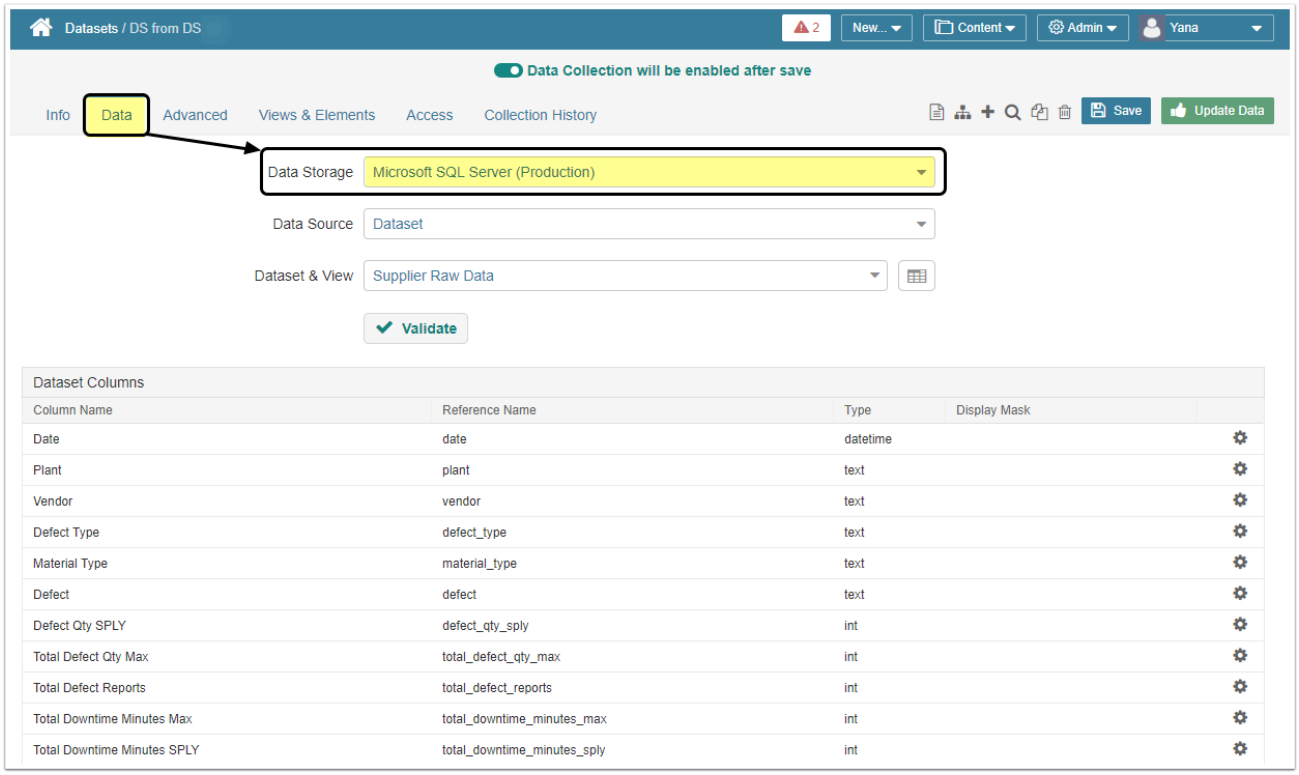
NOTE:

- After **Data Storage Initialization**, configuration process is complete and can not be reverted.

- If you need to change the ***Database name***, re-enter ***Password*** and save changes before reinitializing the Data Storage.

⚠ To improve *write* performance to SQL Storage, please see [Installing a Microsoft SQL Agent to improve INSERT performance when storing data in SQL Storage]. This is especially critical for saving large datasets (millions of rows).

5. Selecting the Data Storage (applicable to Datasets only)



Having configured the Data Storage profile, Users can select it for storing their MI data (Datasets).

This option is available from the **Dataset Editor > Data tab**

- To understand the process of building a Dataset, view [Create a Dataset from any Data Source]

2.2 Installing a Microsoft SQL Agent to improve INSERT performance when storing data in SQL Storage

Microsoft SQL Storage is great for storing large sets of data. However, INSERT operations for millions of rows can take hours. To solve for this, Metric Insights created a SQL Agent that runs directly on the Microsoft SQL Server. This agent facilitates the bulk insert of datasets via a CSV file.

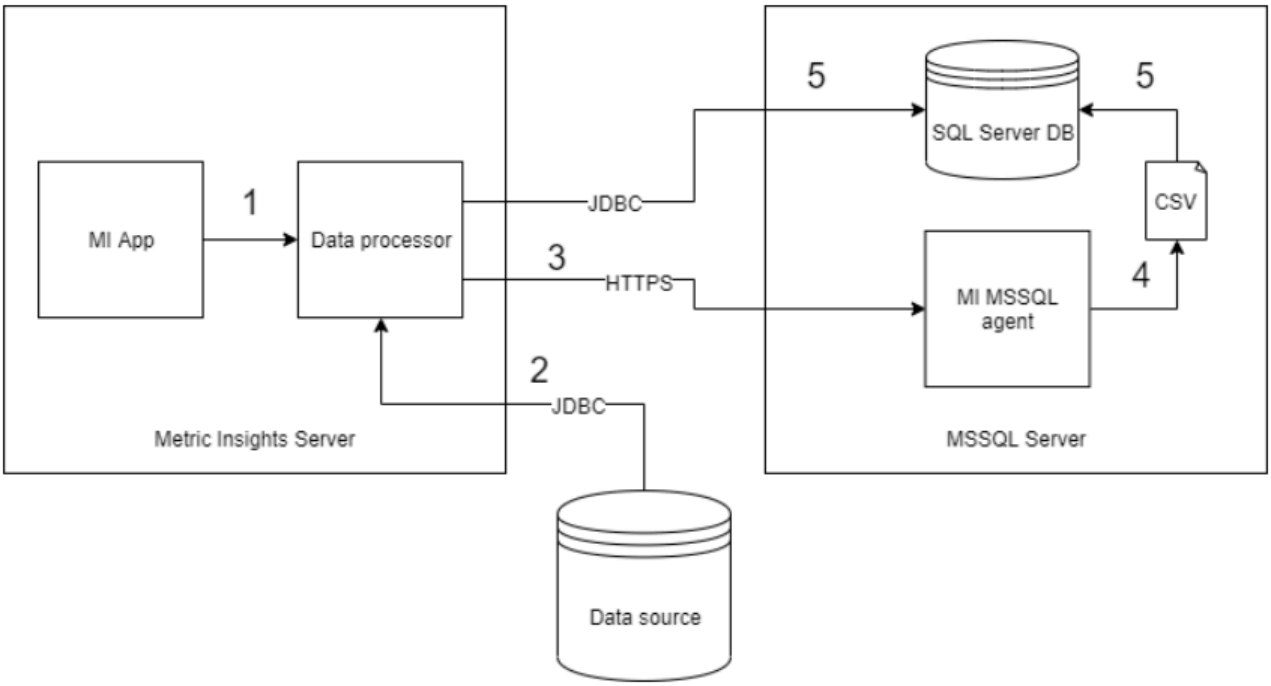
This article covers the following:

How does the Microsoft SQL Agent work?

The goal of the MS SQL Agent is to accomplish the following actions:

1. Fetch data from any source
2. Generate a CSV file from the fetched data
3. Send the CSV file to the remote SQL Agent
4. Execute a BULK INSERT using the CSV file in Microsoft SQL Server

See the diagram below for a more detailed look at how this works.



1. Metric Insights sends a job request to the Dataprocessor (DP)
2. DP collects data from a specific source, then generates a CSV file
3. DP sends the CSV to the Microsoft SQL Agent via HTTPS
4. SQL Agent saves the CSV file to a local folder
5. DP executes a BULK INSERT of the CSV through the SQL Agent
6. DP sends a command to the SQL Agent to delete the CSV file once the dataset is saved to SQL Server

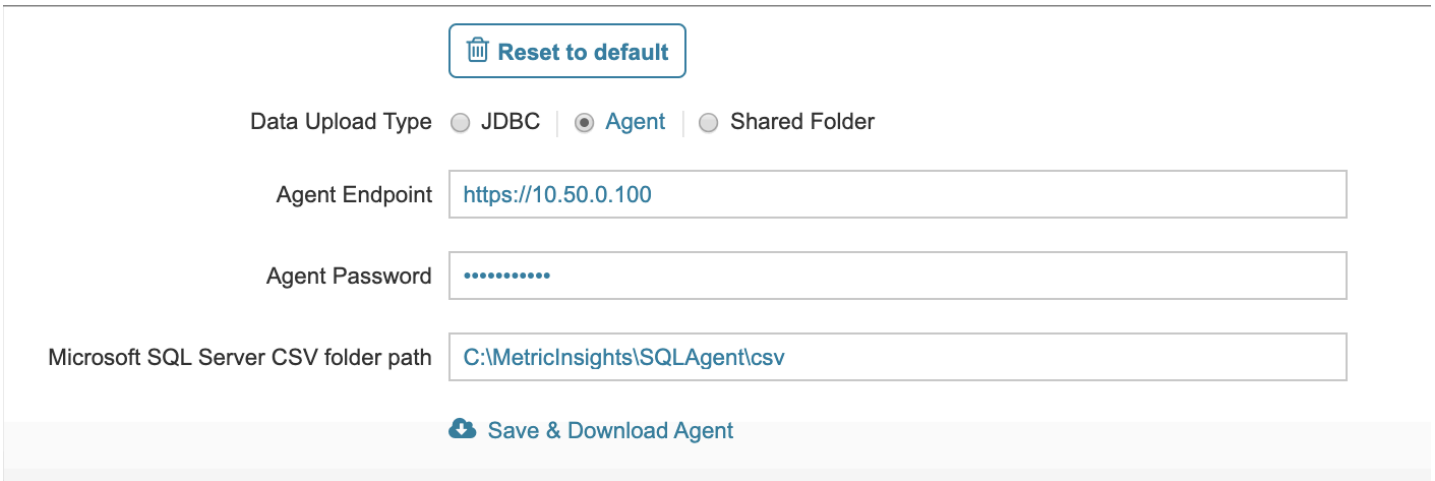
Prerequisites

- JRE 8+ (Java Runtime Engine) required on Windows environment hosting Microsoft SQL Server
- SQL Data Storage profile already created in Metric Insights

Install the Microsoft SQL Agent

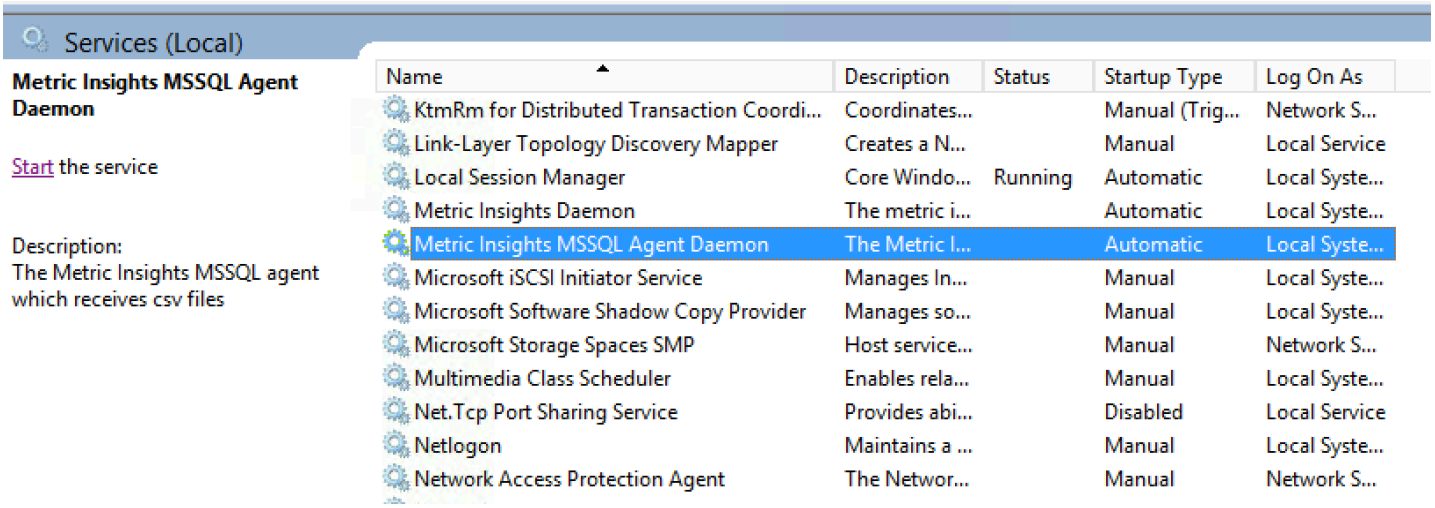
1. Go to Admin menu > Collection & Storage > Data Storage > select desired SQL Data Storage profile
2. In the SQL Data Storage profile, scroll down to the bottom to where it says *Data Upload Type* > select **Agent**

3. Set *Agent Endpoint* to the SQL Server host (IP or URL)
   - e.g., `https://10.50.0.100`
4. Set a password for HTTPS authentication in *Agent Password*
5. Set *Microsoft SQL Server CSV folder path* to the folder where CSV files will be saved on the SQL Server host
6. Click [Save & Download Agent]



After downloading the zip, move it to the Windows machine hosting SQL Server. On the Windows host:

1. From the zip file, extract the content to the desired location
   - e.g., `C:\MetricInsights\SQLAgent\`
   - the location should reflect the location set in the UI
2. There are three subfolders extracted from the zip:
   - bin
   - conf
   - lib
3. In the *conf* folder, open the *application.properties* file in a text editor:
   - Ensure a password is set for the parameter *server.ssl.key-store-password*
   - Ensure the parameter *agent.security.http.password* is set to the same password that was set in the UI for *Agent Password*
4. Now, install the SQL Agent by going to the *bin* folder and running **install.bat** with administrator rights. On install completion, the *Metric Insights MSSQL Agent Daemon* should appear in the list of Windows Services (see image below)
5. Start the daemon in the Windows Services window, or run *start.bat* from the conf folder
6. Create a CSV folder in the same location as specified in the UI
   - Ensure the MS SQL user has read and write permissions to this folder



Metric Insights is now ready to bulk insert large datasets into MS SQL Storage! Test to confirm you can create datasets. If you encounter any issues, please contact support@metricinsights.com for help.

2.3 Setting up Windows Server Shared Folder

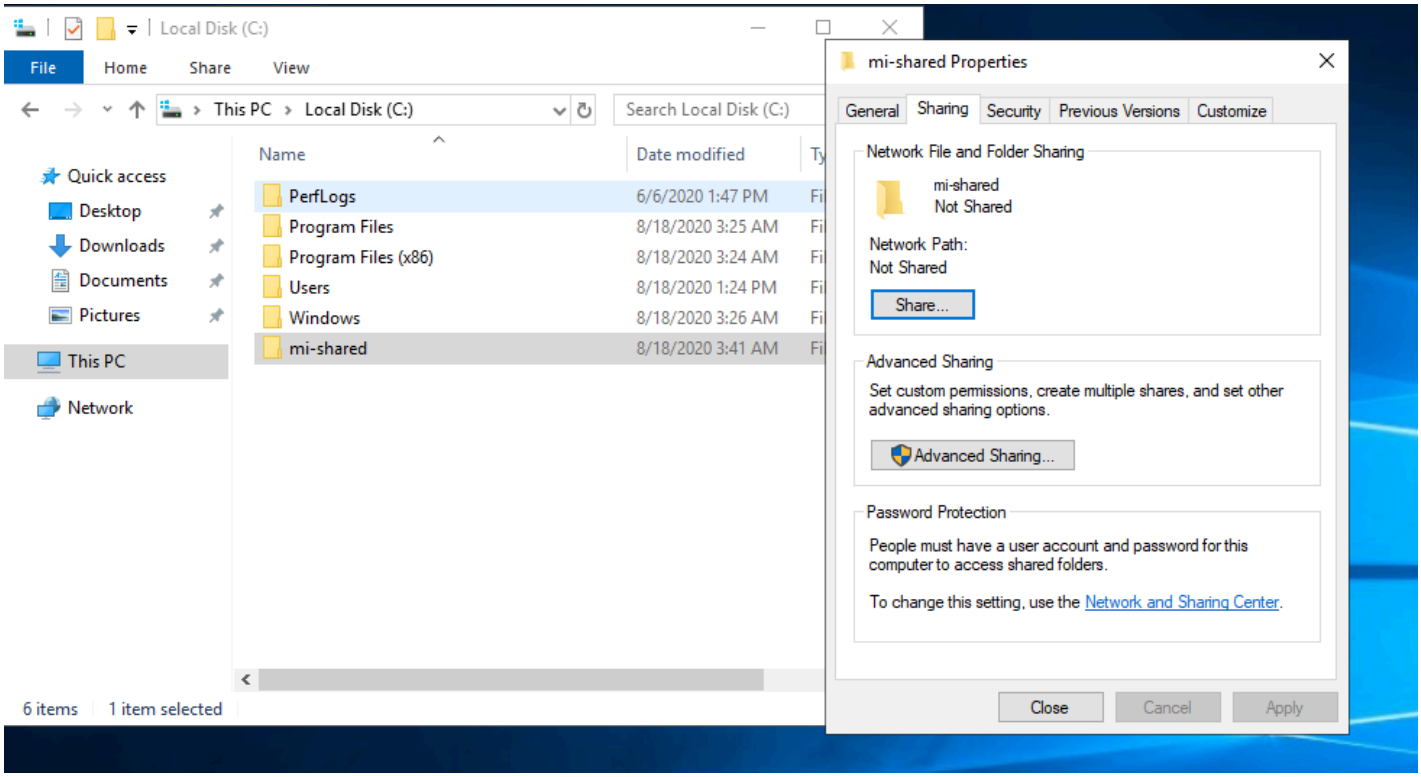
Moving a large amount of data into Microsoft SQL Server is most performant by bulk loading via a shared folder. This document describes the steps required to setup a shared folder in Windows Server 2019.

Example Specifications:

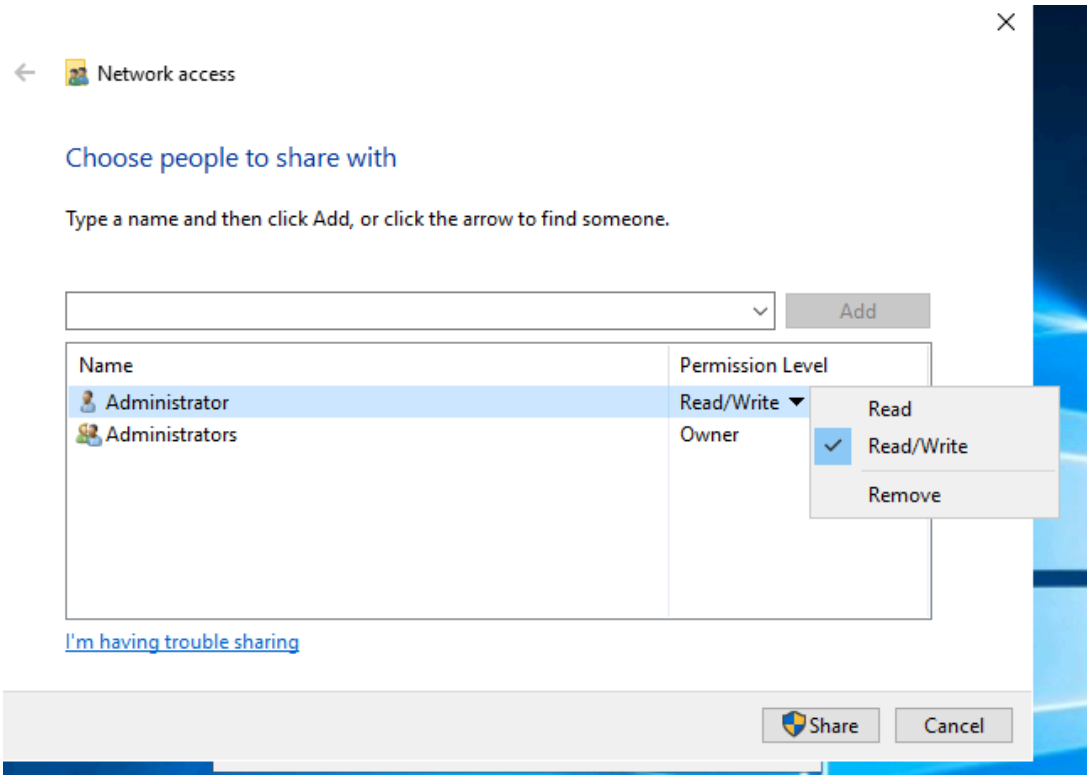
- SQL Windows Server : **Windows Server 2019 Standard (Desktop Experience)**
- MI Application 6.2.0: **Debian 10 / Ubuntu 18.04 / CentOS 7 / CentOS 8**

Open Shared folder on Windows Server

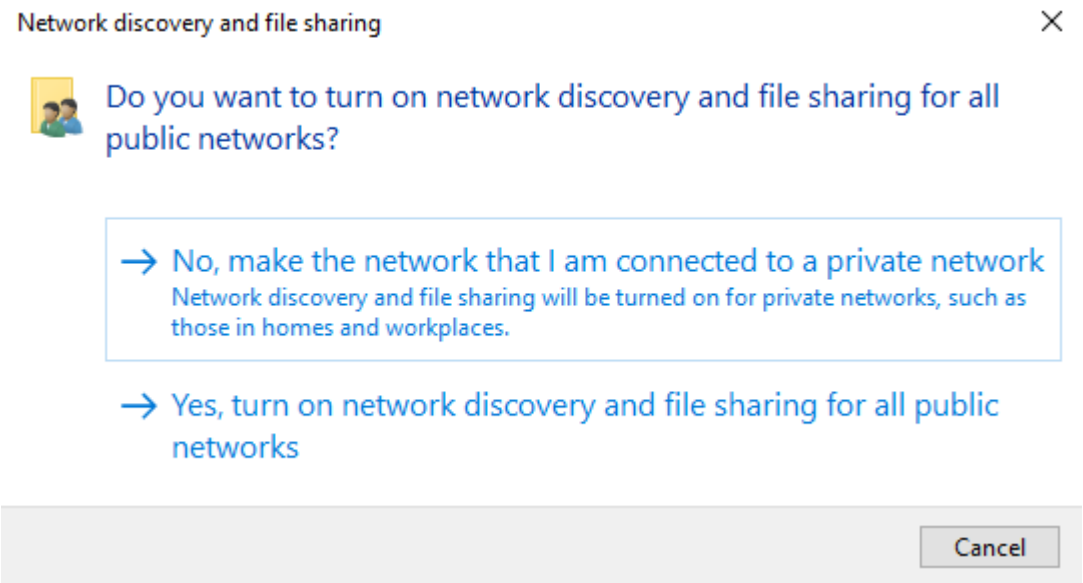
1. Right click on **Shared** Folder → **Properties** → **Sharing tab** → **Share**

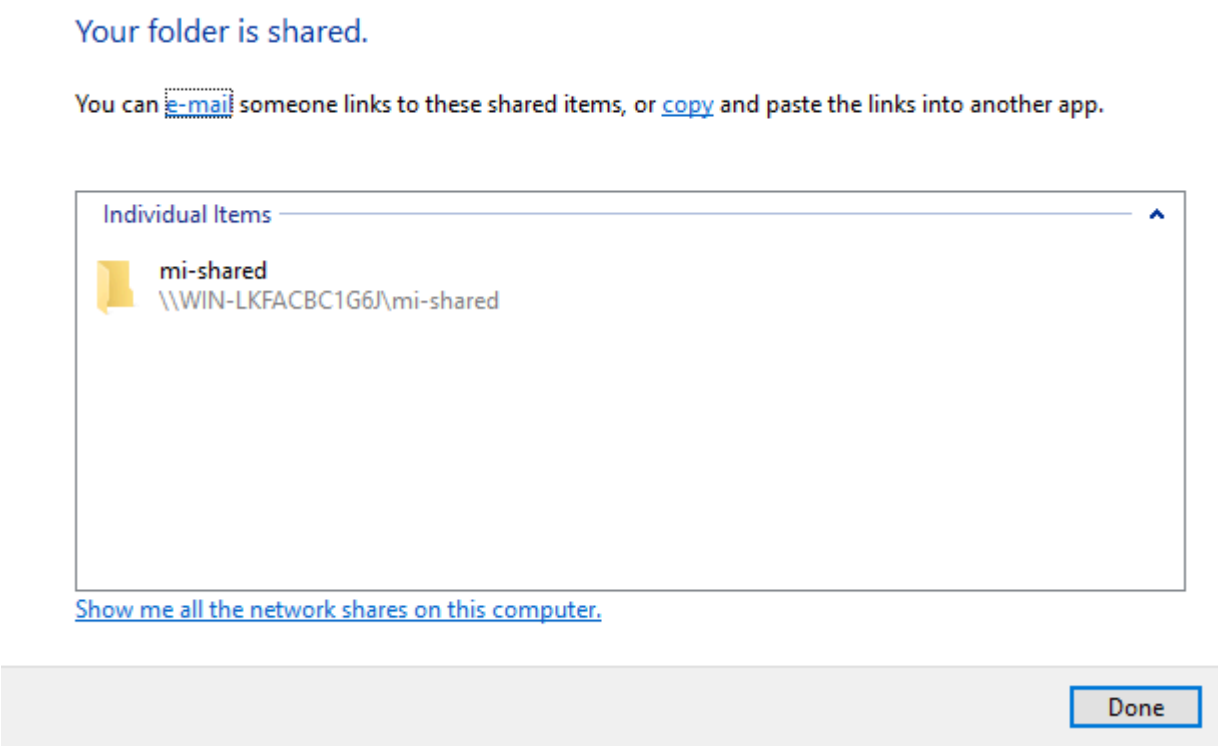


2. Add user for **Shared** folder



3. Turn on **Network Discovery** for all public networks





4. Make sure that SMB ports (**139** and **445**) on Windows Server is opened and you have access to SMB from **Linux Instance**

```
root@toshuk-X550VC:~# telnet 192.168.33.59 139
Trying 192.168.33.59...
Connected to 192.168.33.59.
Escape character is '^]'.
^]
telnet> q
Connection closed.
root@toshuk-X550VC:~# telnet 192.168.33.59 445
Trying 192.168.33.59...
Connected to 192.168.33.59.
Escape character is '^]'.
^]
telnet> q
Connection closed.
```

Mount Shared folder from Windows to Linux instance

Make sure that your **Linux Instance** have access to **Windows Server.**

1. Create **Shared** folder

```
$ mkdir -p /opt/mi/shared
```

2. Install required package

Via **Debian** or similar OS:

```
$ apt-get install cifs-utils
```

Via **RedHat** or similar OS:

```
$ yum install cifs-utils
```

3. Mount Shared folder from **Windows Server** to **Linux Instance**

```
root@mi61:/opt/mi# sudo mount -t cifs //192.168.33.59/mi-shared /opt/mi/shared -o
user=Administrator
Password for Administrator@//192.168.33.59/mi-shared:  *******
```

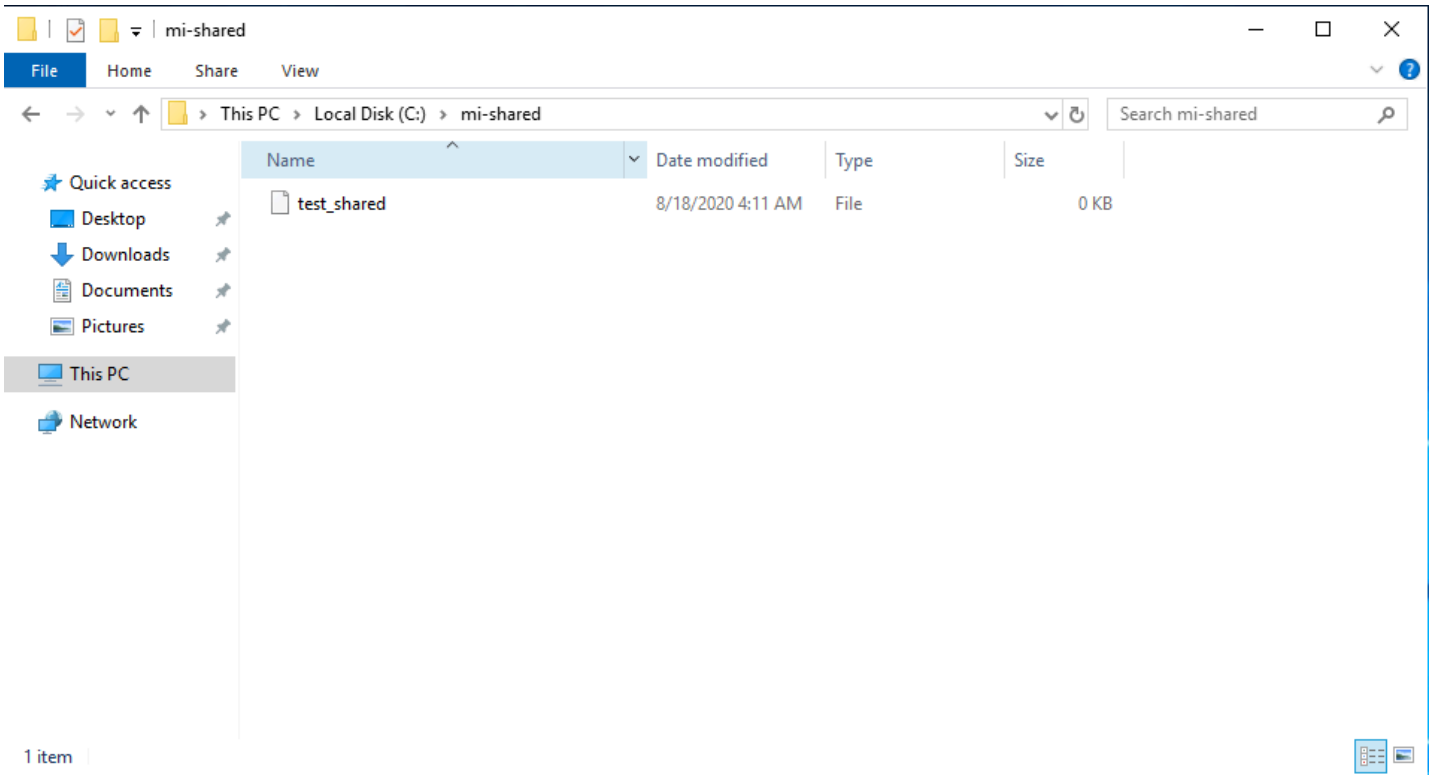
Where:

- **192.168.33.59** - IP of **Windows Server** with opened **Shared Folder**
- **mi-shared** - Opened folder on **Windows Server**
- **/opt/mi/shared -** folder for mount on **Linux Instance**
- **user=Administrator** - **Windows Server** user for access to **shared folder**

4. Confirm that mounting was successful

```
root@mi61:/opt/mi# mount | grep 192.168.33.59
//192.168.33.59/mi-shared on /opt/mi/shared type cifs (rw,relatime,vers=default,cache=strict,username=Administrator,domain=,uid=0,noforceuid,gid=0,noforcegid,addr=192.168.33.59,file_mode=0
755,dir_mode=0755,soft,nounix,serverino,mapposix,rsize=1048576,wsize=1048576,echo_interval=60,actimeo=1,user=Administrator)
root@mi61:/opt/mi#
```

```
root@mi61:/opt/mi/shared# touch test_shared
root@mi61:/opt/mi/shared# ls -la
total 4
drwxr-xr-x  2 root root    0 Aug 18 11:11 .
drwxr-xr-x 13 root root 4096 Aug 18 11:01 ..
-rwxr-xr-x  1 root root    0 Aug 18 11:11 test_shared
root@mi61:/opt/mi/shared#
```



Mount Shared folder to Docker Container

To add the shared folder to Docker container, add a mount point from **Linux Instance** to Docker container.

1. Edit **docker-compose** yml file of docker container

```
$ vi /opt/mi/config/deployment/docker-compose.dataprocessor.yml
```

2. Add mount point

```
    volumes:
      - ${APPLICATION_FOLDER:-.}/ssl:/opt/mi/ssl:ro
      - ${APPLICATION_FOLDER:-.}/data/dataprocessor/external-libs:/app/external-libs:ro
      - ${APPLICATION_FOLDER:-.}/patch:/opt/mi/patch:rw
      - /opt/mi/shared:/opt/mi/shared:rw
```

Where:

- **/opt/mi/shared:/opt/mi/shared:rw** - shared folder on **Linux Instance** and inside **Docker container**, with read/write permissions

3. Launch the dataprocessor via **root** user

```
$ vi /opt/mi/config/deployment/credentials/dataprocessor.env
```

And set variable **USER=root**

```
MYSQL_DATAPROCESSOR_USER=mi_dataprocessor
MYSQL_DATAPROCESSOR_PASSWORD=mqE12tAa9zSGl55a
DP_USER=dataprocessor
DP_PASSWORD=3Ii2jM6sHl21fTLR
ENABLE_SSL=false
MONITORING_ACCESS_TOKEN=EHuSlIFisIHTpTUd
DP_OPTIONS=
DP_MYSQL_OPTIONS=
USER=root
```

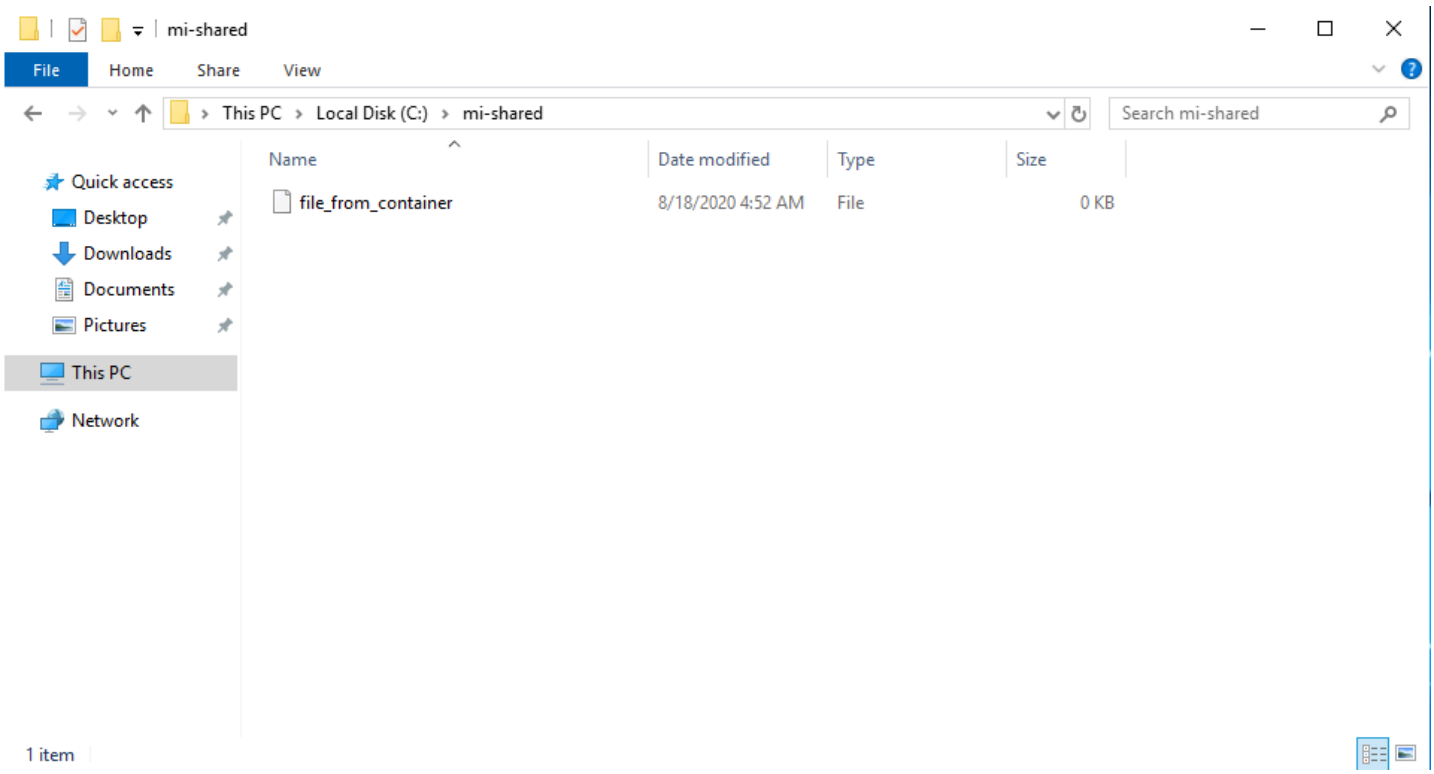
Variable **USER=root** need to set possibility for running dataprocessor by **root** user with access to shared folder

4. Recreate the Docker container

```
$ mi-control up -d dataprocessor
Recreating mi_dataprocessor_1 ... done
```

5. Check Shared folder inside docker container

```
$ docker exec -it mi_dataprocessor_1 bash
$ cd /opt/mi/shared
$ touch file_from_container
```



3. How to Link to External BI Tools

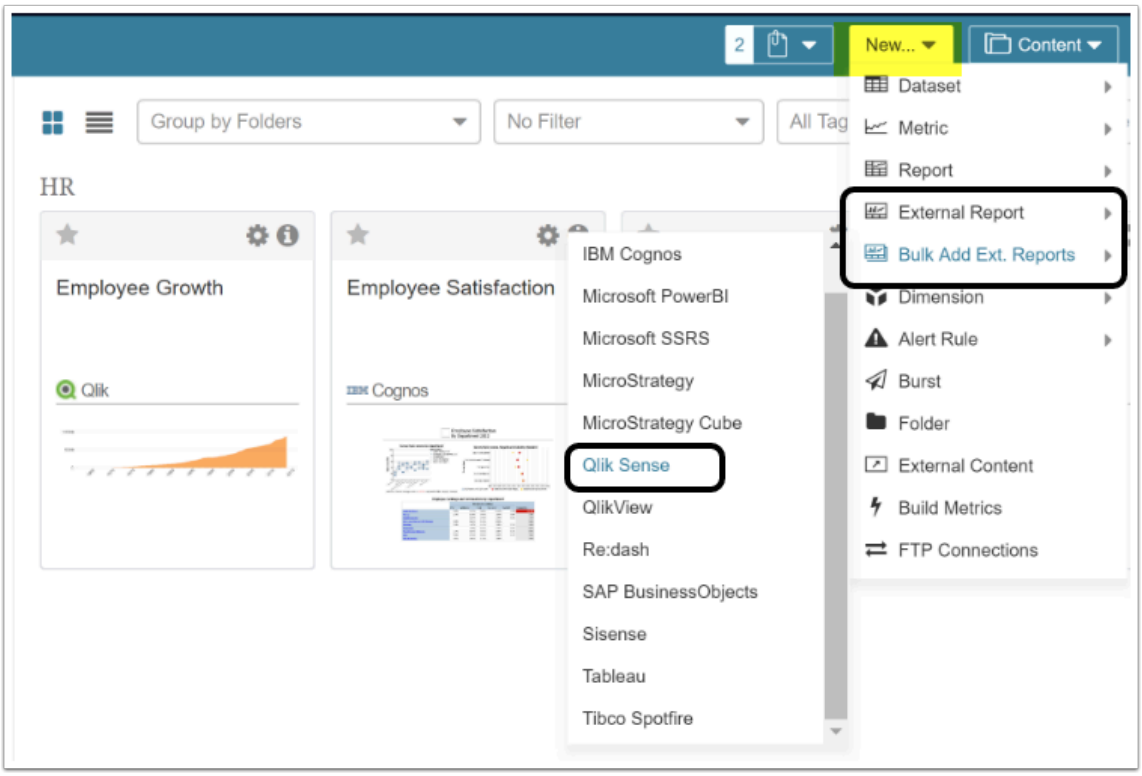
3.1 External Report Overview

This article provides an overview of External Reports within the system.

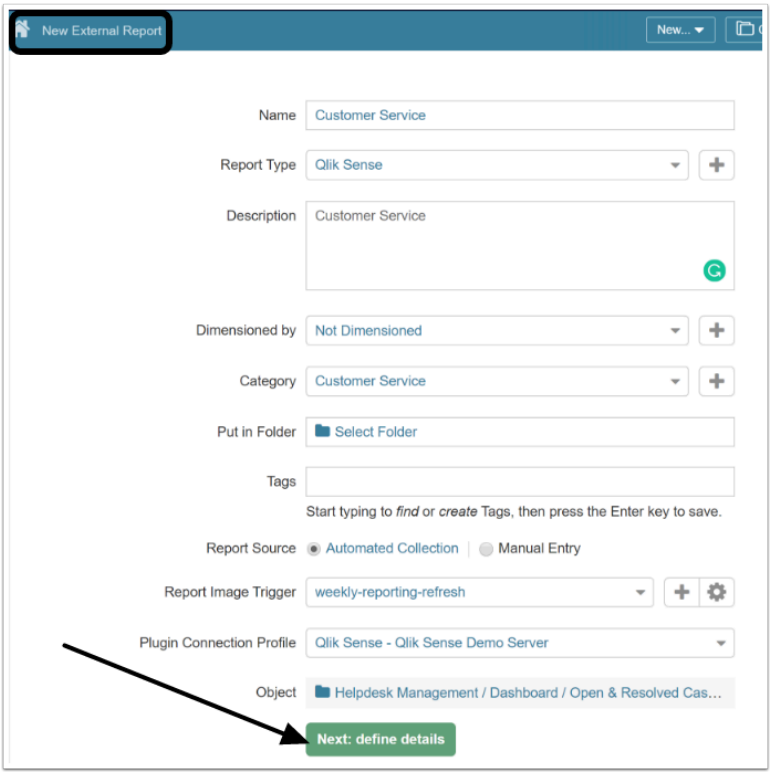
The difference between External Report and Other External Content:

- With an **External Report,** you can actually drill down from another Metric Insights tile into the specified external report passing specified filters to show only selected values in the External platform.
- **External Content**, however, is just any URL of your choosing that will not be passed any dynamic parameters from Metric Insights.

1. Access New >External Report > BI tool

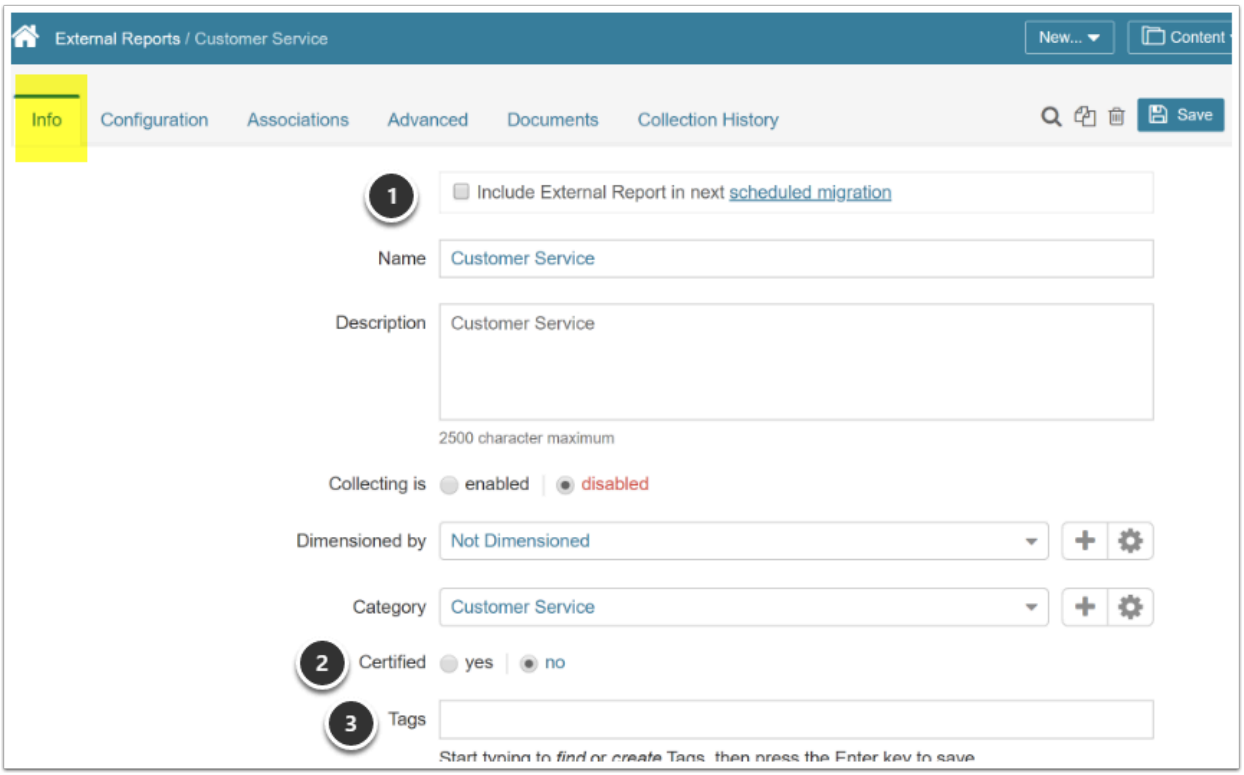


1.1. Fill in the basics for new External Report



Fill in Basic information and select [Next: define details] to open full Editor.

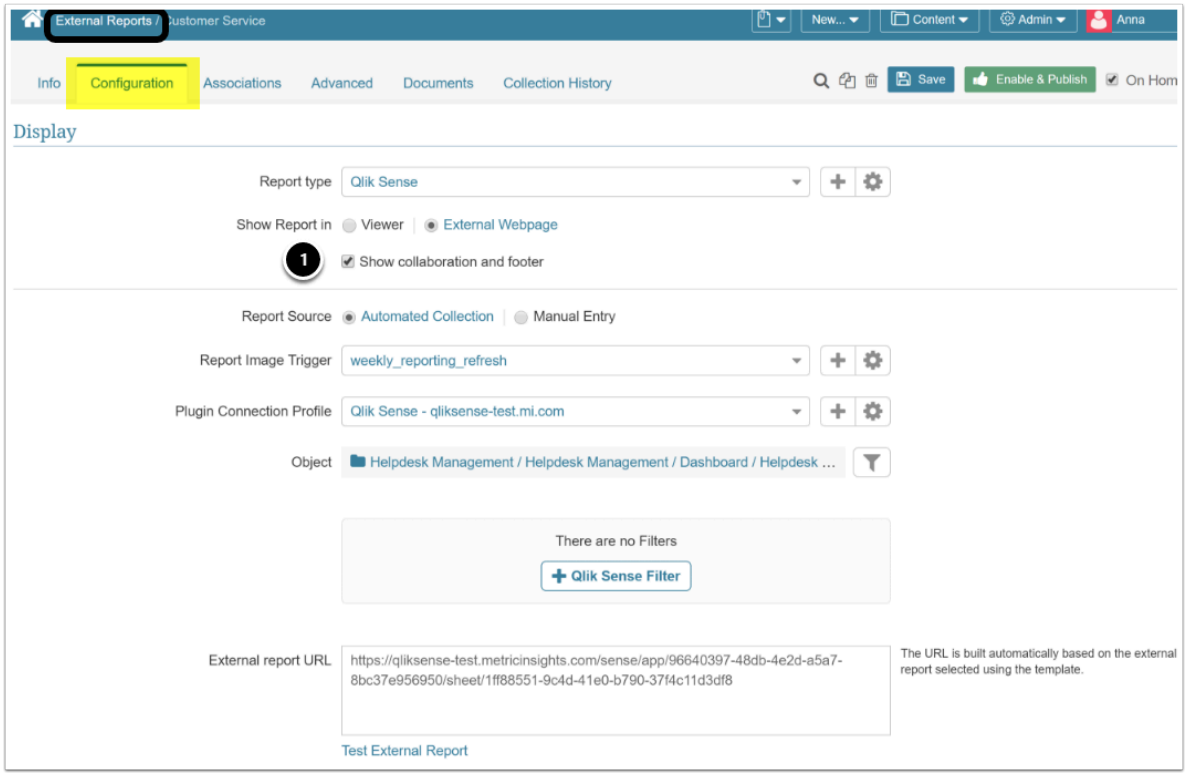
1.2. You can access optional fields on the Info tab



1. [6.1.0] As of Release 6.1.0, it is possible to flag elements and then Migrate Content using our Export/Import Migration Scripts. Click for details: Scripted Migration via Category and Element Editors
2. Certified: Certification is a means for Admin and Power Users to identify elements that have been approved as being valid and accurate. For details refer to: Certifying an Element
3. Tags (are referred to as Topics in Versions prior to 5.1): Tags / Topics are used to relate elements to each other for purposes of identifying similar Charts when the See Related drop-down list in the Metric Viewer is populated allowing a more in-depth analysis of trends. For details refer to: Create a Topic / Tag

1.3. Open the Configuration tab

Its contents are based upon the specific BI Tool requirements. Below is an example



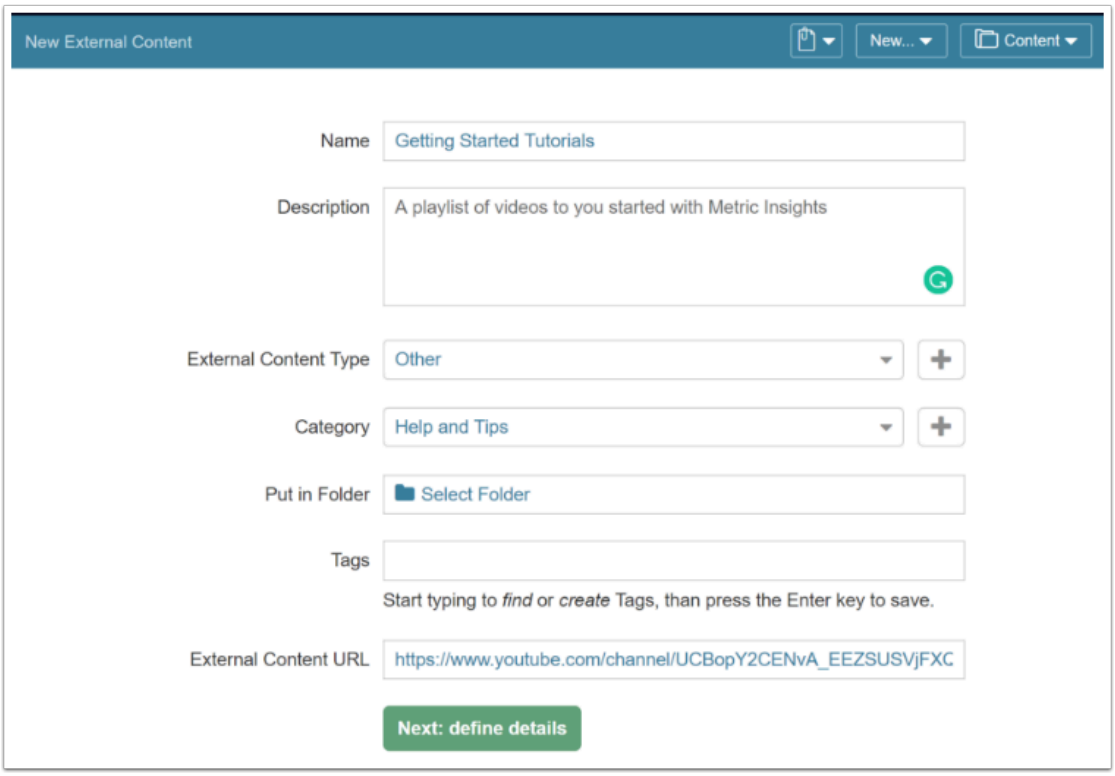
1. Fields vary, but new in V5.5.1 is the option to display **Collaboration and Footer** information. The default value is set to show these fields, but may be unchecked to remove them from the Viewer or External Webpage.

This example is using QlikSense as our BI tool. The Configuration of any External Report is specific to each BI tool and so varies. See the following for articles for specifics based on your BI system:

- How to create an External Report from Tableau
- How to create an External Report from Sisense
- How to create an External Report from QlikView
- How to collect External Reports from Beckon

Metric Insights' Plug-ins.

2. Create new External Content (Access New > External Content)



Select a unique Name and add the External Content URL

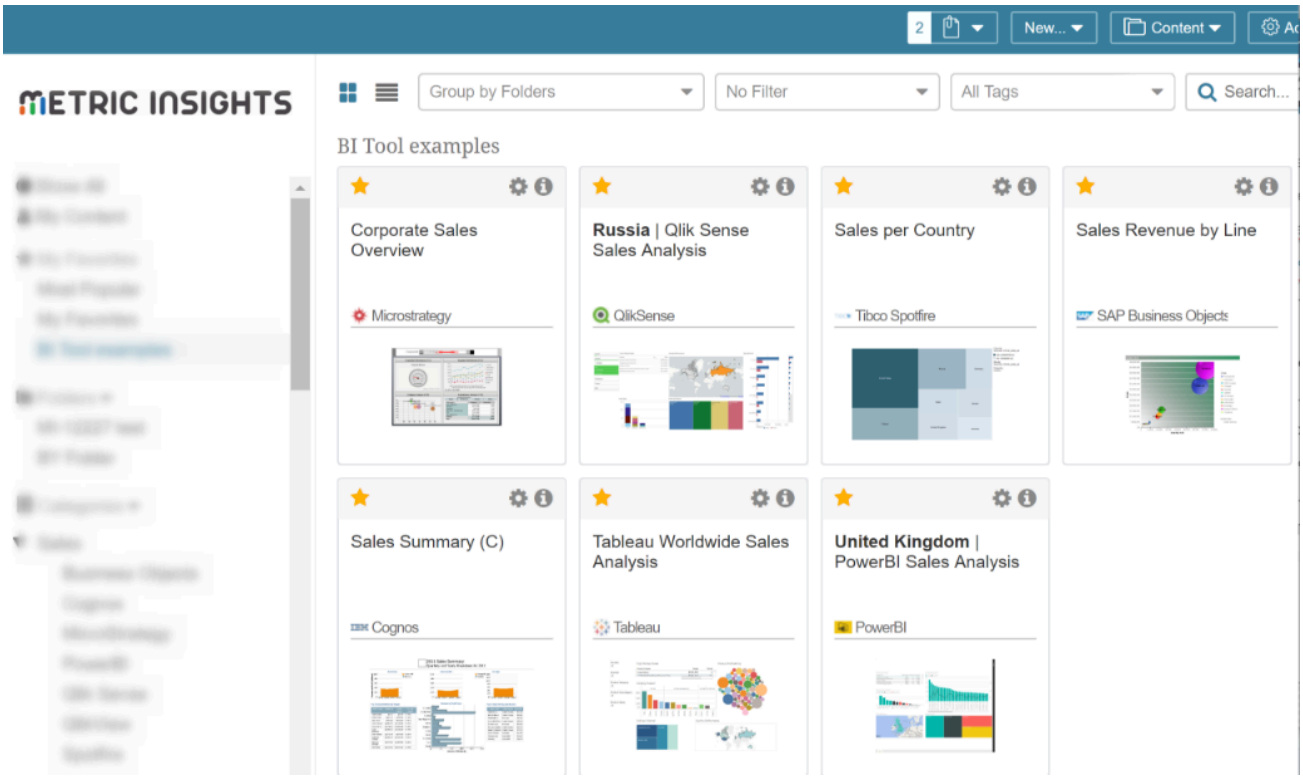
If you need additional assistance after you select [Next: define details], see [How to link to Other External Content]

3. External tool authentication

Each URL that Metric Insights will link too will, of course, require some form of log in for that system. Metric Insights will not be able to bypass authentication to any other system.

4. End result

Once External Reports/Content are created, a Business User will be able to seamlessly link to a variety of relevant reports all from their own personal list of favorites or report categories they have access to.


New...
Content
METRIC INSIGHTS
Group by Folders
No Filter
All Tags
Search...
BI Tool examples
Corporate Sales Overview
Microstrategy
Russia | Qlik Sense Sales Analysis
QlikSense
Sales per Country
Tibco Spotfire
Sales Revenue by Line
SAP Business Objects
Sales Summary (C)
Cognos
Tableau Worldwide Sales Analysis
Tableau
United Kingdom | PowerBI Sales Analysis
PowerBI

3.2 Filtering in External Reports (Pre-filtering BI Tools)

Metric Insights 5.6 introduces some powerful new features for External Report consumption and distribution:

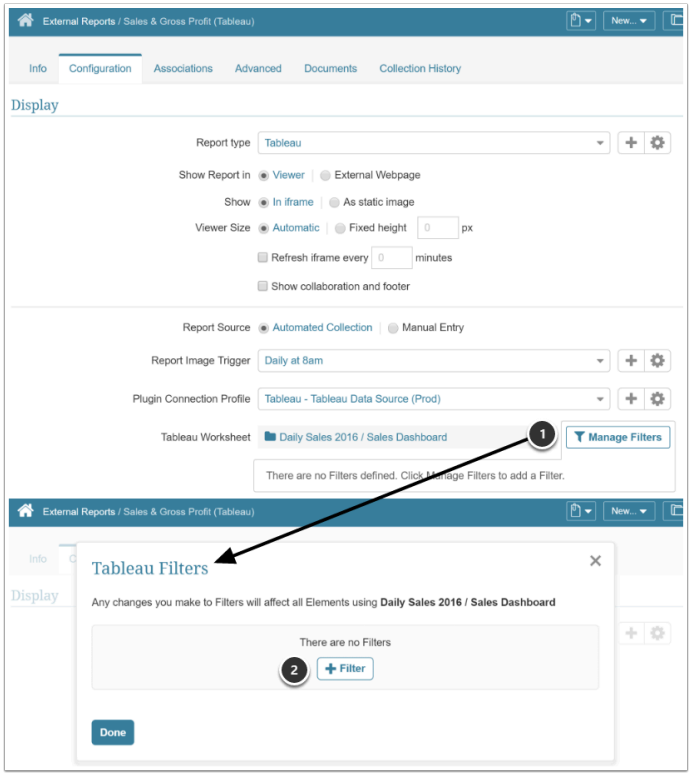
1. The Pre-filtering function limits the data values available whenever the BI Tool object is used in Metric Insights by limiting the amount of data actually fetched from the BI Tool. Only this data will be available when setting Defaults and Utilizing Bookmarks.
2. Applying Filter Defaults enables Power Users and Admins to present and distribute at enterprise-scale by utilizing User Maps:
   1. Defaults can be set by the creator of elements fetching data from any BI tool. These defaults can be set for **all Users** or can be individually set using a User Map
   2. Bursts can be configured *to pass the correct Filters and Filter Values to the correct Users,* personalizing all delivered content, but only requiring a single Burst, a single External Report, and a single User Map (all working together).

Personalized Bookmarks are also implemented. These are Individual Defaults that can be set by any User when viewing an External Report. The Bookmark function allows any User to personalize how they view and subscribe to their own External Report content. For details, see Setting Personal Bookmarks (External Reports)

[6.1.1] There are minor changes to the format of the Filter options to allow for display / non-display of the individual filters. These changes were required to allow clients to omit all Filters from the Viewer if their BI Workbooks or Dashboards already include Filters. See How to omit Filters from External Reports.

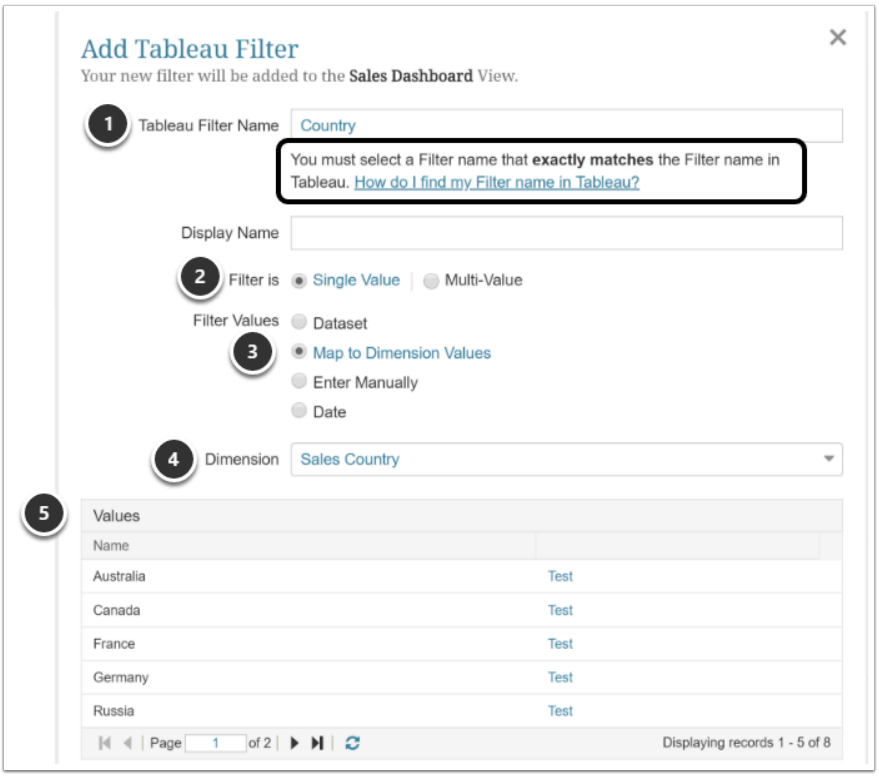
1. Pre-filtering (External Report Editor > Configuration tab)

Although the examples show Tableau, the core functionality has been implemented across all BI tools that support Filtering. We have more specific articles for Pre-filtering Qlik Sense and Pre-filtering QlikView data. These articles include some added functionality compared to other BI tools.


External Reports / Sales & Gross Profit (Tableau)
New...
Info
Configuration
Associations
Advanced
Documents
Collection History
Display
Report type
Tableau
Show Report in
Viewer
External Webpage
Show
In iframe
As static image
Viewer Size
Automatic
Fixed height
px
Refresh iframe every
minutes
Show collaboration and footer
Report Source
Automated Collection
Manual Entry
Report Image Trigger
Daily at 8am
Plugin Connection Profile
Tableau - Tableau Data Source (Prod)
Tableau Worksheet
Daily Sales 2016 / Sales Dashboard
1
Manage Filters
There are no Filters defined. Click Manage Filters to add a Filter.
External Reports / Sales & Gross Profit (Tableau)
New...
Tableau Filters
Any changes you make to Filters will affect all Elements using Daily Sales 2016 / Sales Dashboard
There are no Filters
2
Filter
Done

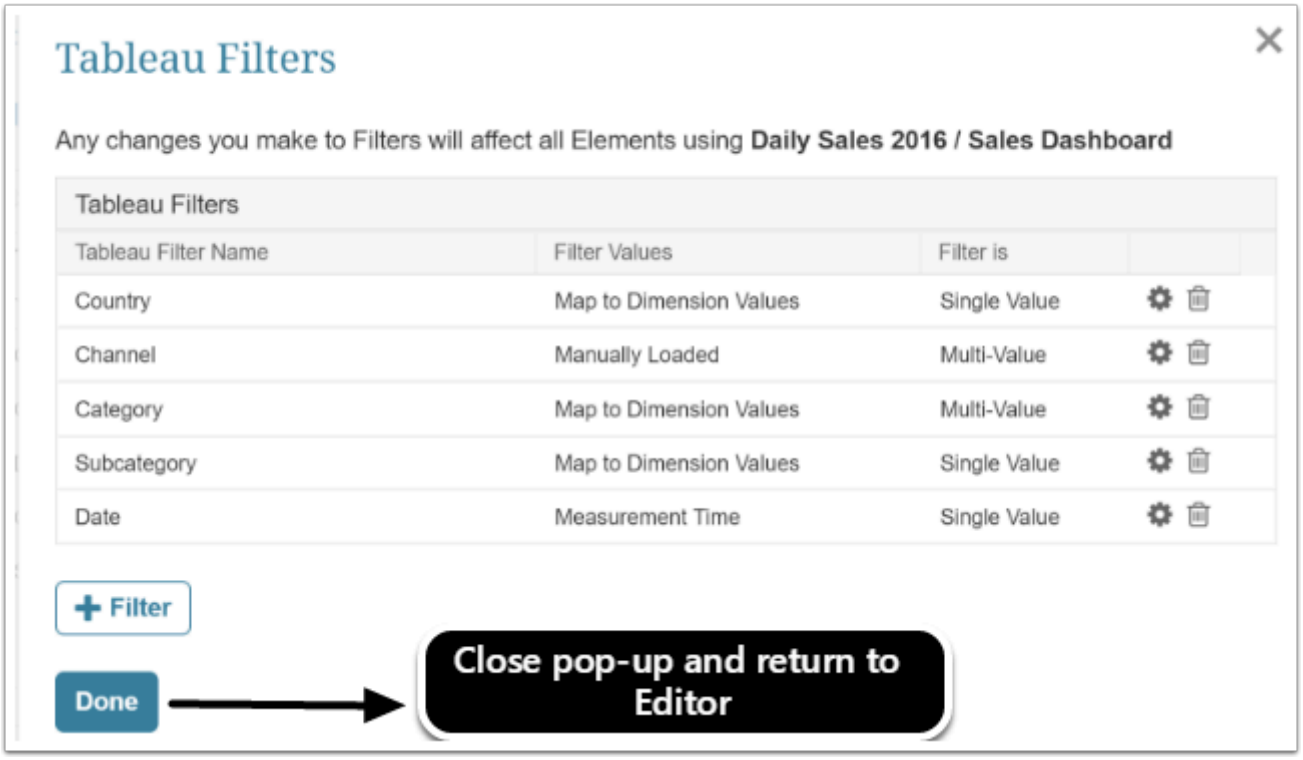
There are 4 separate options for use in defining Filters.

1.1. (Options 1&2) Mapping Filters to Dataset or Dimension Values

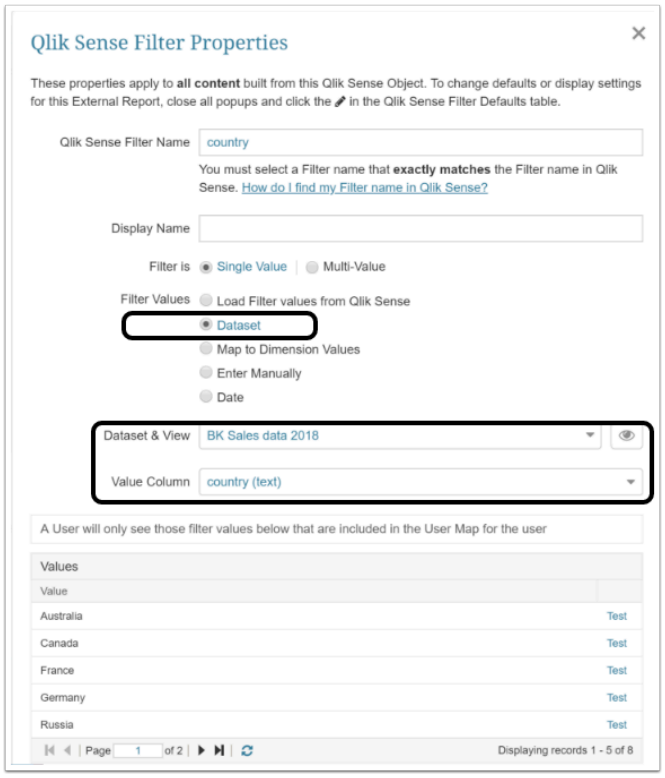


1. Enter the [Filter Name] that exactly matches the filter name in your BI tool. The system will provide additional information on how that is done based on the specific BI tool.
2. Specify if the Filter may contain either Single or Multiple Values when presented in the Report Viewer.
3. The Filter Values can be set in a variety of ways; in this example, we are using the Values from a previously defined Dimension.
4. Specify the Dimension
5. Values grid will be populated from choice in #4

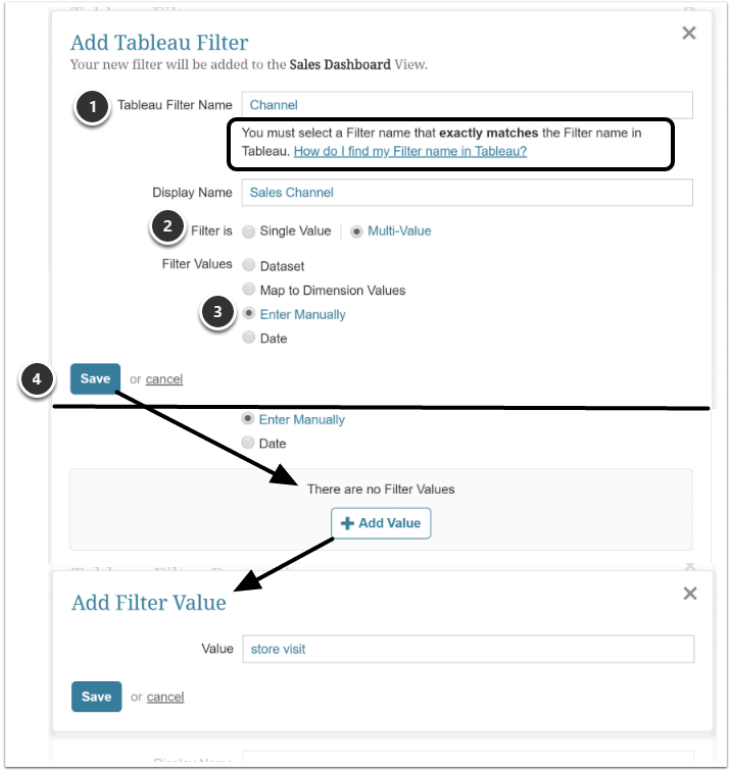
Select **[**Done] to add Filter



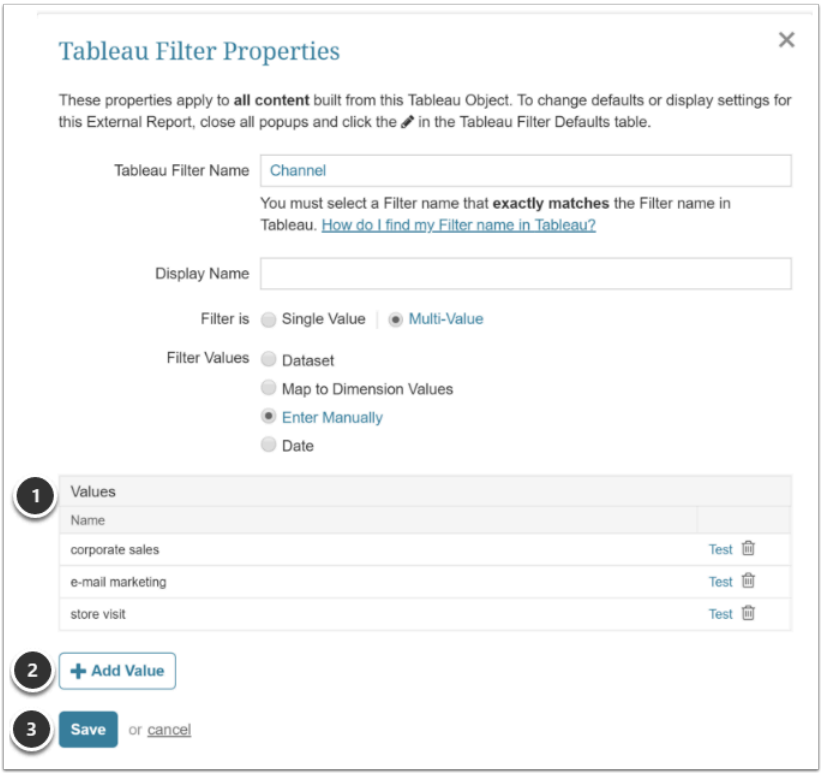
The steps to Load Filter Values from a Dataset are the same as above. You simply specify a Dataset and a Value column as the source of the Filter Values for a specified [BI Tool] Filter Name.



1.2. (Option 3) Entering Filter Values Manually



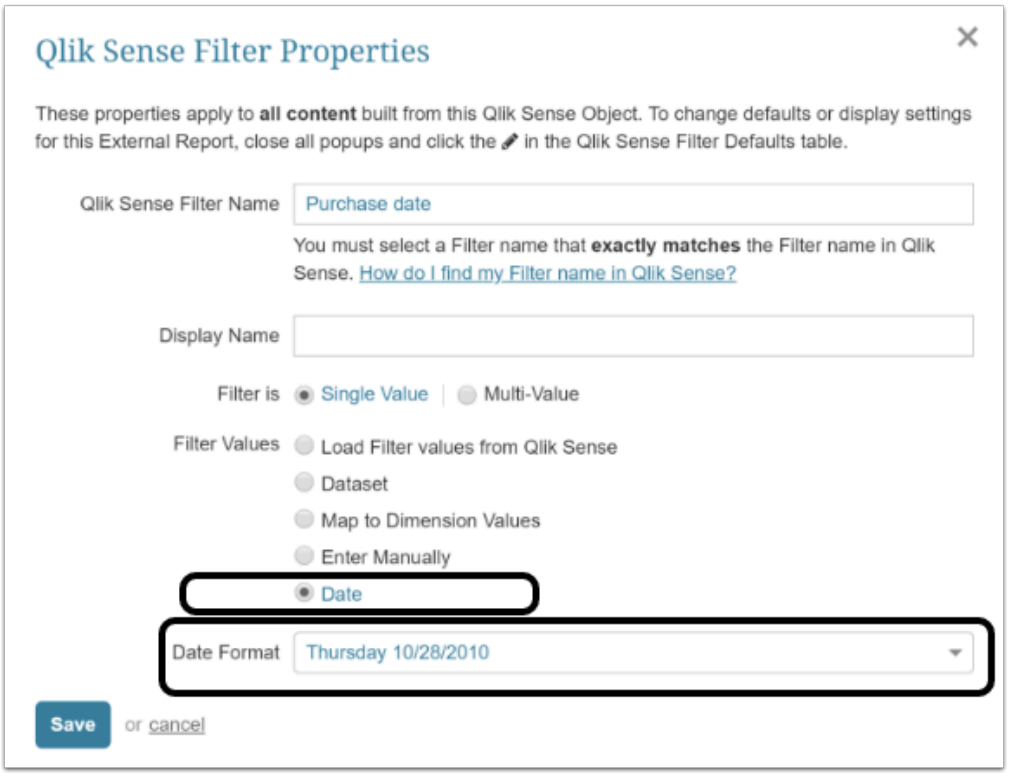
1. Enter the Filter Name that exactly matches the filter name in your BI tool. The system can provide additional information on how that is done based on the specific BI tool.
2. Specify if User may select Single or Multiple Values for this filter. In this example, we allow Users to enter multiple Channels as one Filter.
3. Set the Filters Values to be added Manually
4. [Save] to open [+ Add Value] option.



Make sure that your Filter Values are identical to those in your BI tool!

1. The Values grid will populate with each entry
2. Continue adding Values
3. When complete [Save]

1.3. (Option 4) Mapping Options for Dates

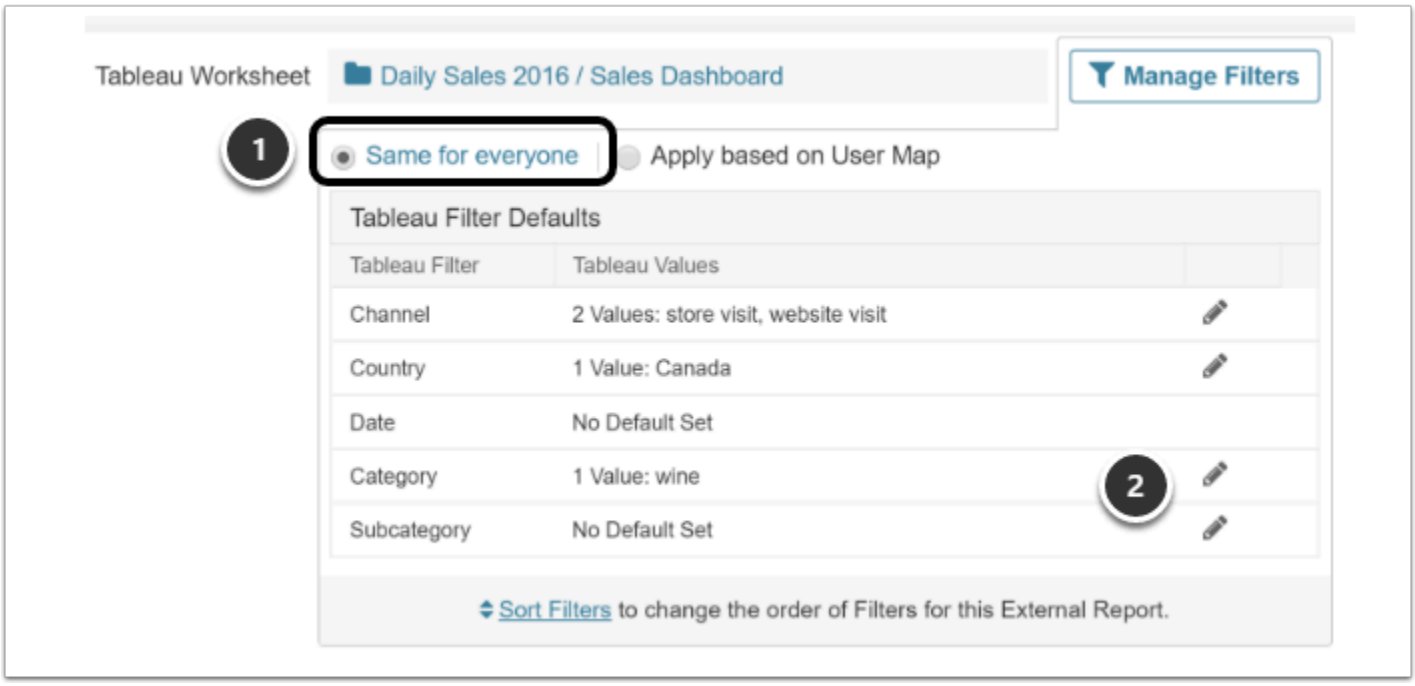


Mapping to a Date only allows the selection of how the Date filters will be displayed.

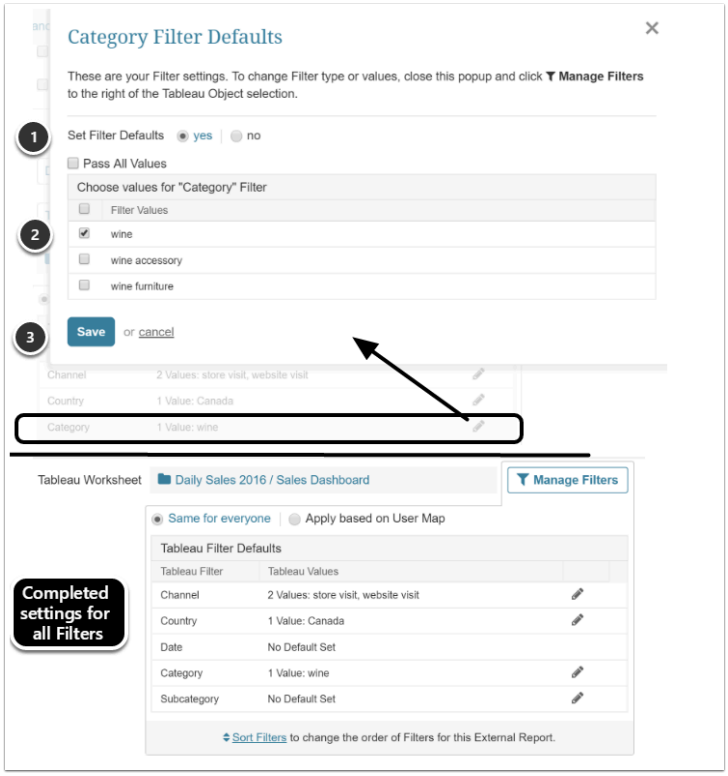
2. Setting Defaults for your Filters

Configuring defaults allows content builders or administrators to define Filter defaults for users viewing the External Report. Defaults may be set the same for everyone, or applied based on a User Map. Consequently, any number of Users will have the correct Filter Values when they open their External Report.

2.1. Setting the Defaults to Be the Same for all Users



1. Select "Same for everyone"
2. Select the Pencil icon to assign the Defaults for Filters



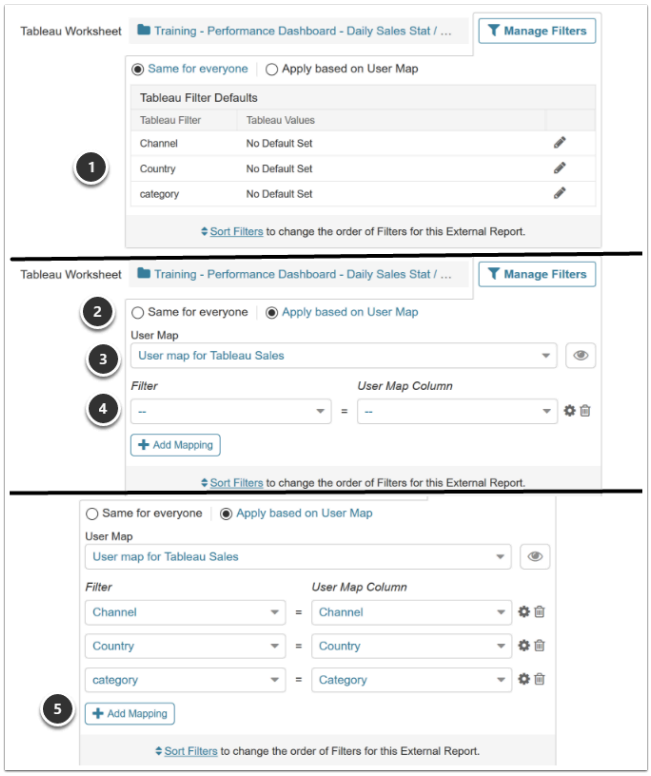
1. Set [Filter Defaults] to "yes"
2. Using the check boxes, select the Default Values

3. [Save]

Repeat to set all of the desired Defaults

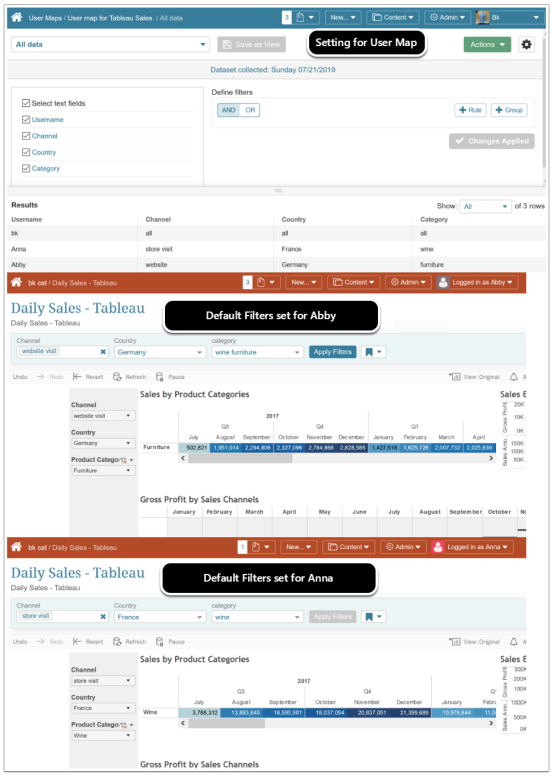
2.2. Setting Defaults Using a User Map (Individual settings)

Setting the Defaults via a User Map does not restrict Users from viewing non-default values. It simply sets the Values displayed upon initial Viewing. If a User is not in the User Map, the system will initially display the View with no default values.



1. First define your BI Tool Filters using any of the options in Step 1
2. Toggle to "Apply based on User Map"
3. Select a User Map from the drop-down
4. Match the BI Tool Filters to this User Map
5. Continue with [+Add Mapping] for additional Filters; it is not required to set User Map Columns for all defined Filters.

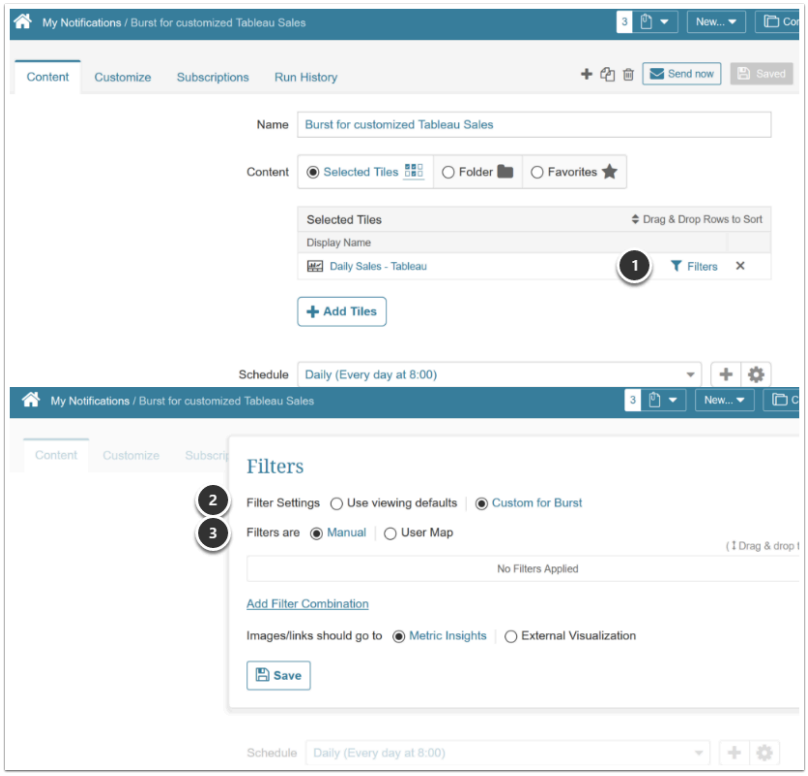
2.2.1. Results when Various Users View Report



Any User may over-ride these initial Default values using Bookmarks! See [Setting Personal Bookmarks]

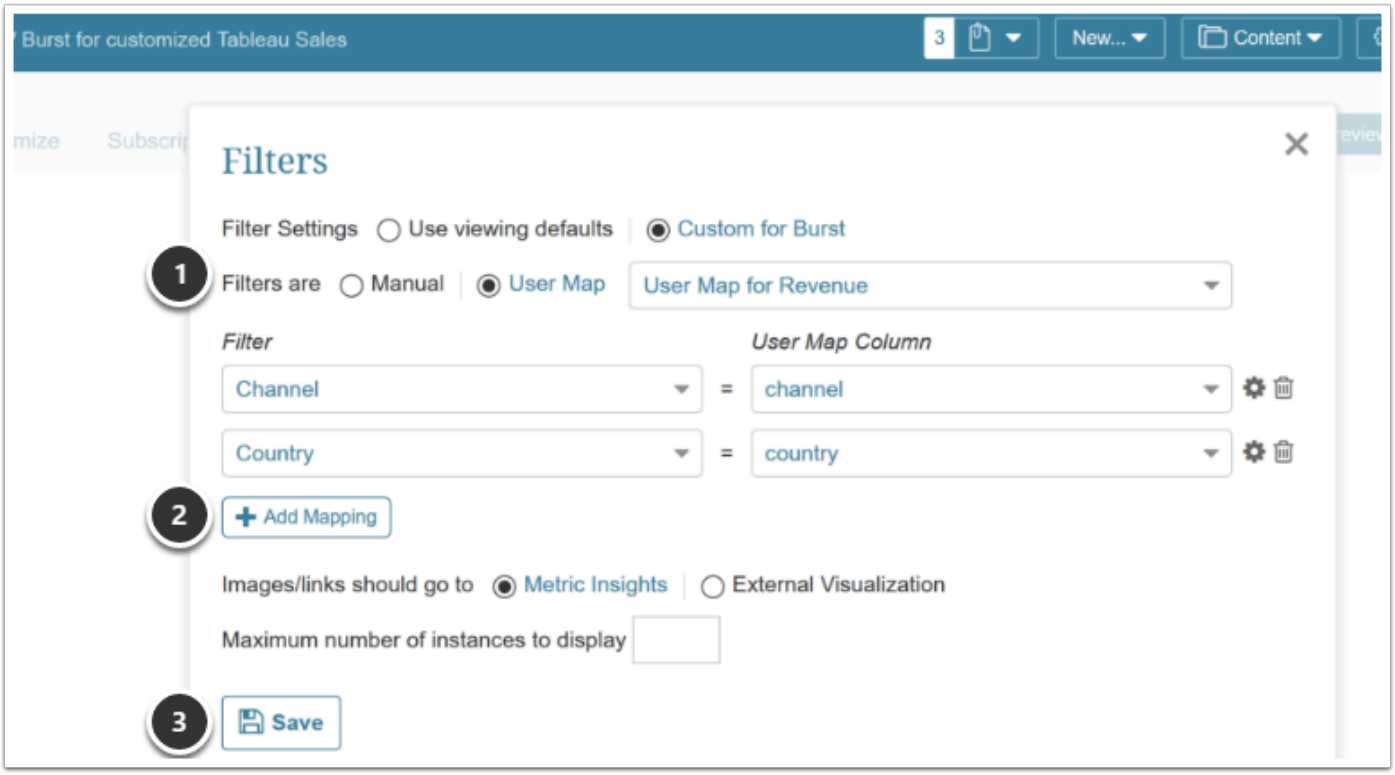
3. Setting Filter Defaults on Burst Editor

Extended Bursting functionality enables large-scale External Report distribution. Applying a User Map *(*with the appropriate mapping of Users to Filters*)* allows a single Burst to be the source for hundreds (or thousands) of User-specific emails. This means that each User (or Group) can receive targeted content based on their needs.



Burst content Filter settings are accessible from the *Burst Editor* and mimics the steps used to set Filters in elements created from your BI Tool.

1. Click the Filters icon next to the External Report you have specified for distribution
   - NOTE: Filters have to be set beforehand at Report Level in the External Report Editor (see Step 1 for Pre-filtering)
2. Select [Custom for Burst] Filter Settings unless you wish to use the Default filter settings originally set in the Report
   - You can set Defaults for all User via Manual entry
   - Or set via a Individualized User map



1. Choose the User Map option to personalize Filters for multiple recipients
2. [+Add Mapping] to apply selected Filters
3. [Save]

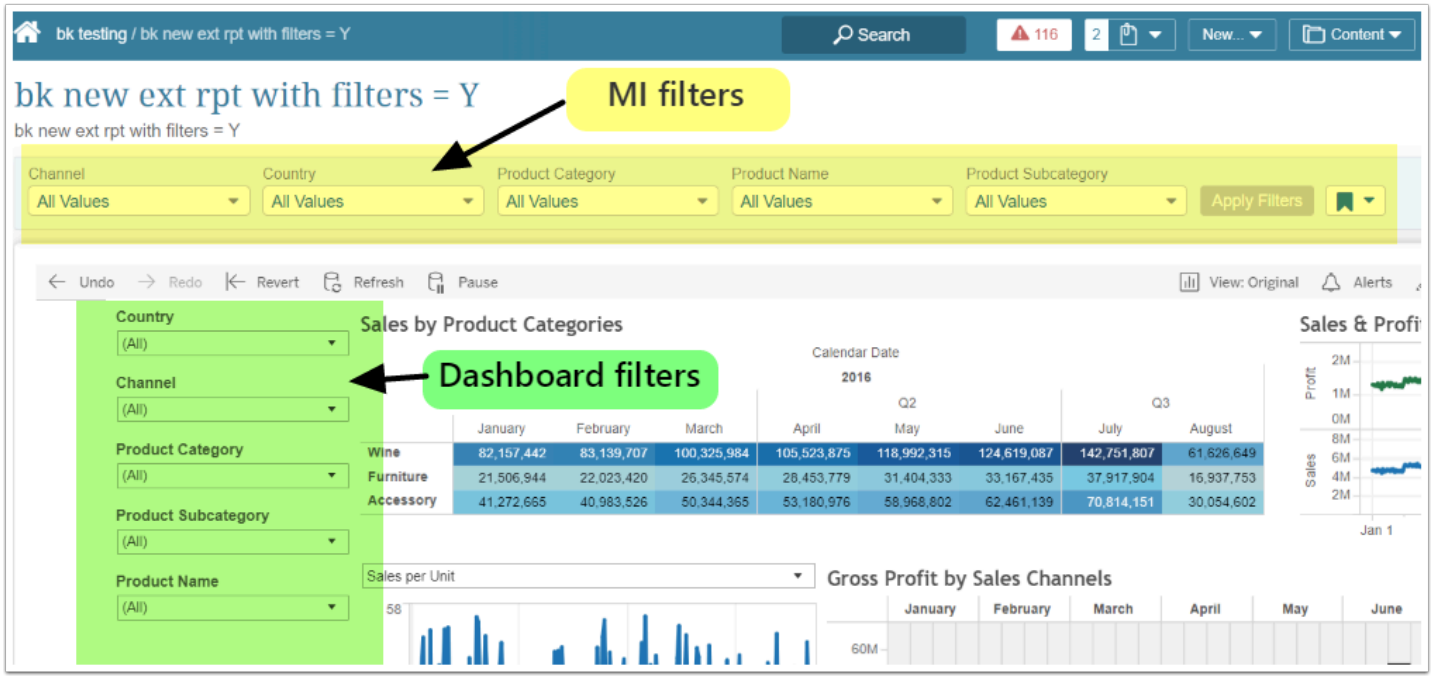
3.3 How to omit Filters from External Reports

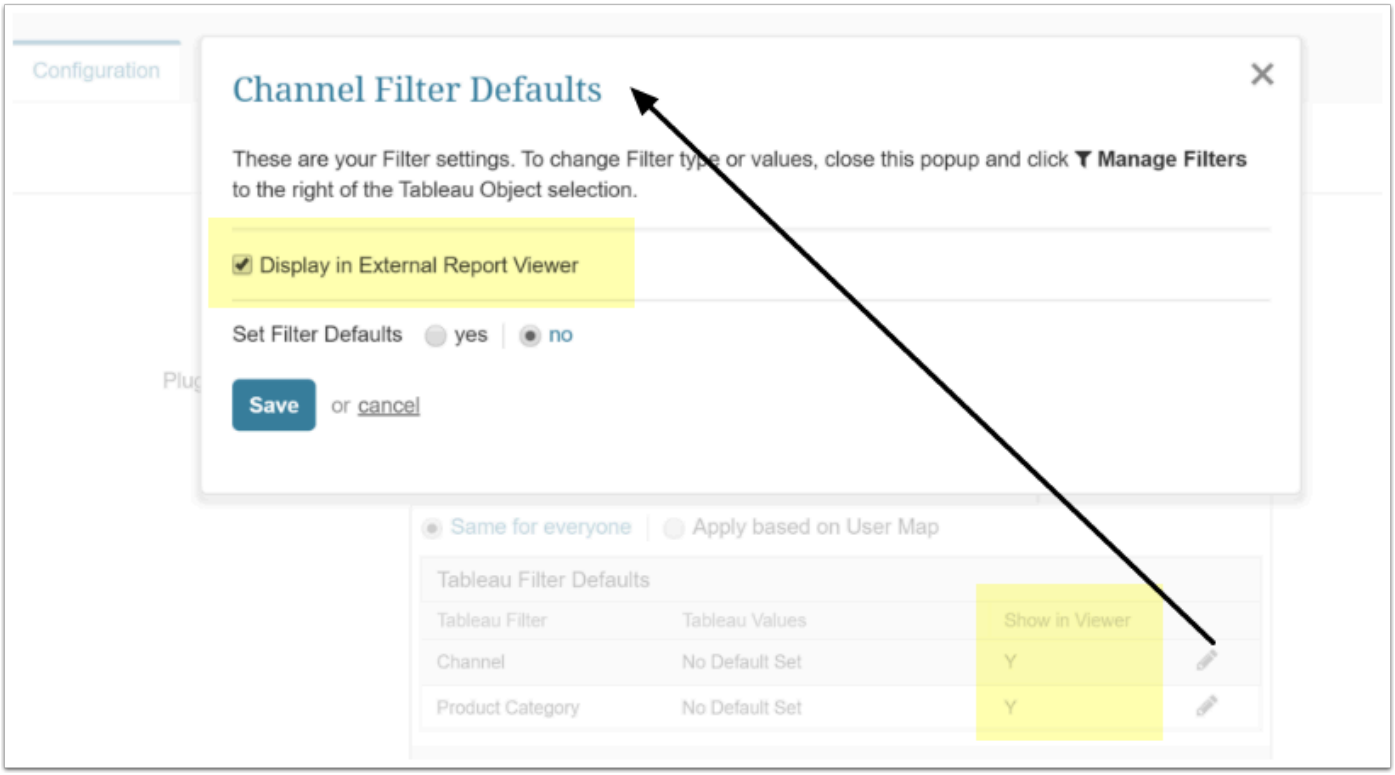
In many cases, your BI Workbooks and Dashboards will already display Filters that were setup in the BI tool. Since our External Reports **default to include** Metric Insight Filters, this creates confusing duplicate Filter display.

This article will show how to set a global default to omit these duplicate Filters (i.e. default the ***"Display in External Report"*** to ***"N")*** when creating new Reports. This option can always be changed for individual Reports manually.

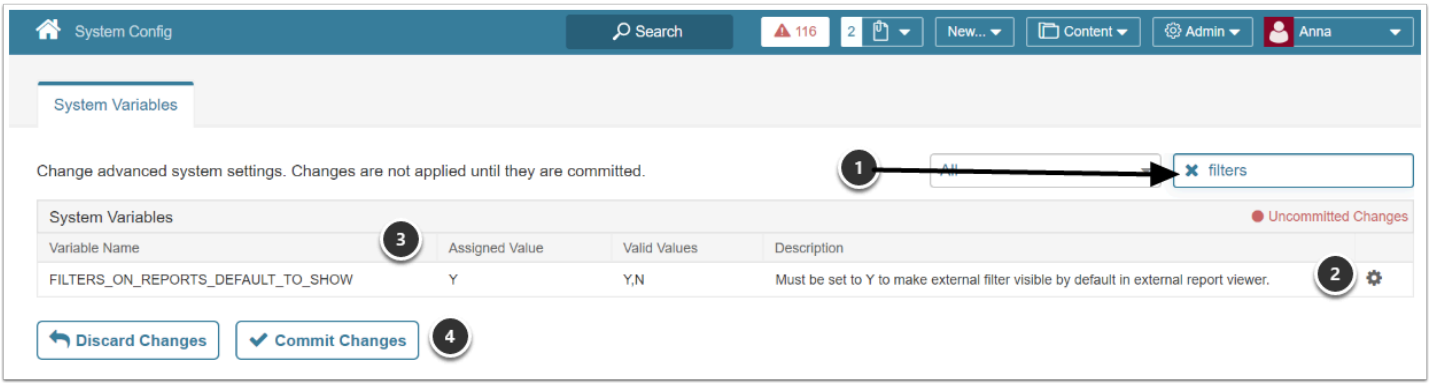
Individual Users can still create their own individual Bookmarks if any Filters are set but not displayed. For more information, see Setting Personal Bookmarks (External Reports.)

Example of duplicate filters in Viewer





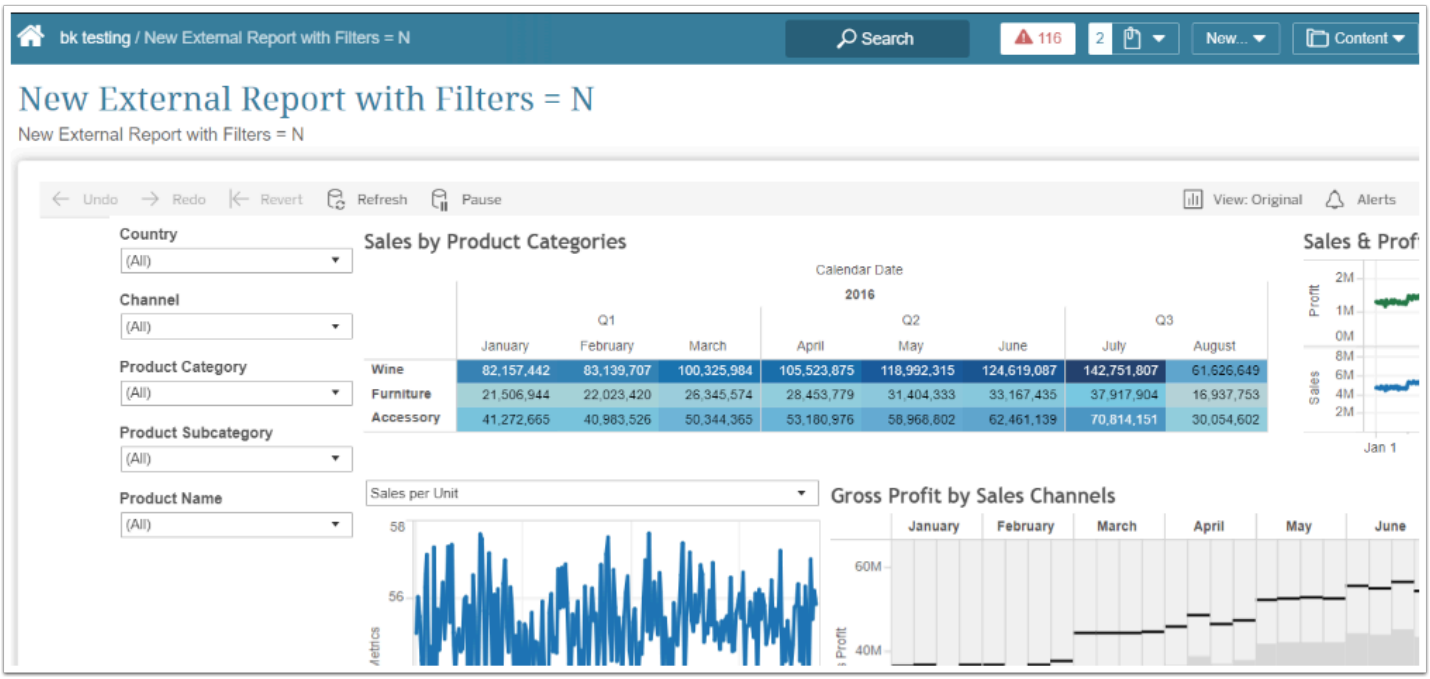
Reset the Config Variable (Admin > System > System Config)

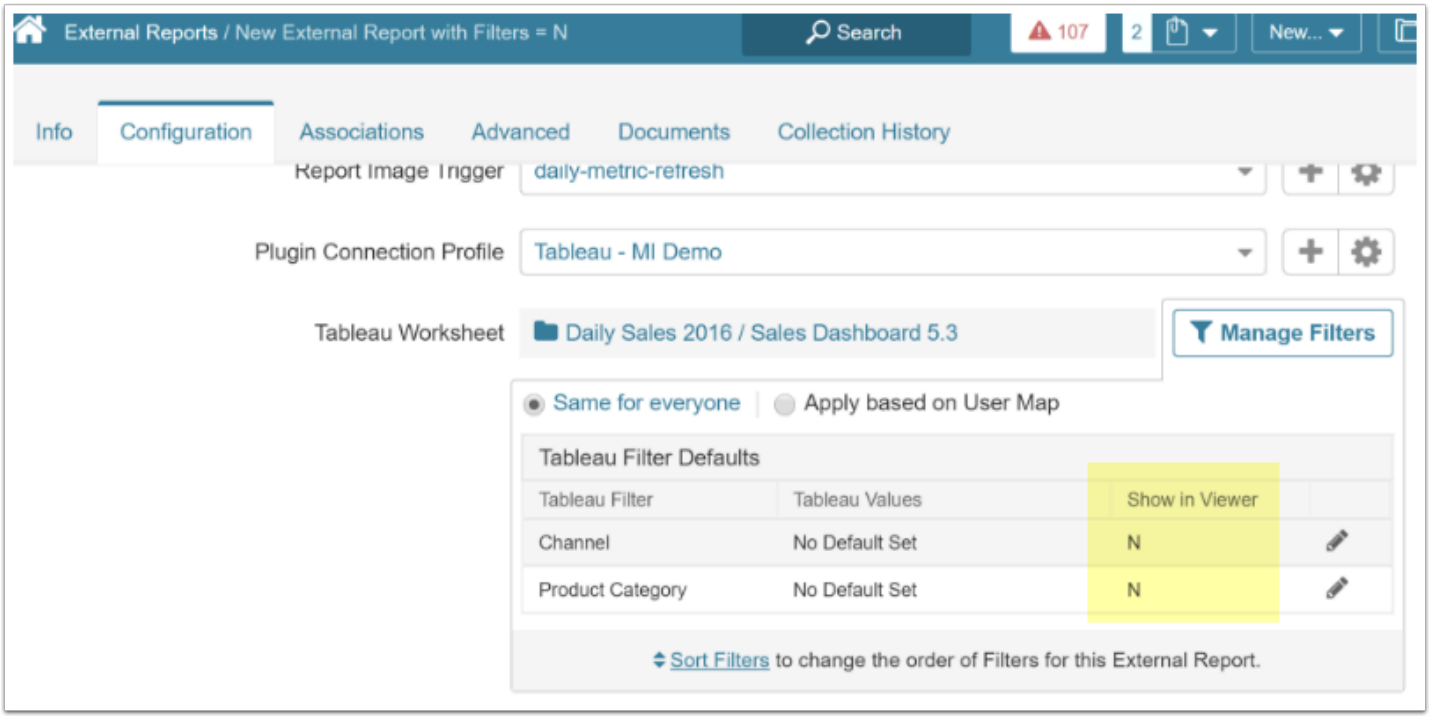


1. Enter '**filters**' in Search criteria
2. Edit the Variable via **edit icon**
3. Change **Assigned Value** to "*N*"
4. Commit Changes

> ⚠ This setting will apply to all **New** External Report defaults, but will not affect existing Reports

Create a new External Report with default set = "N"





3.4 Make External Reports Discoverable

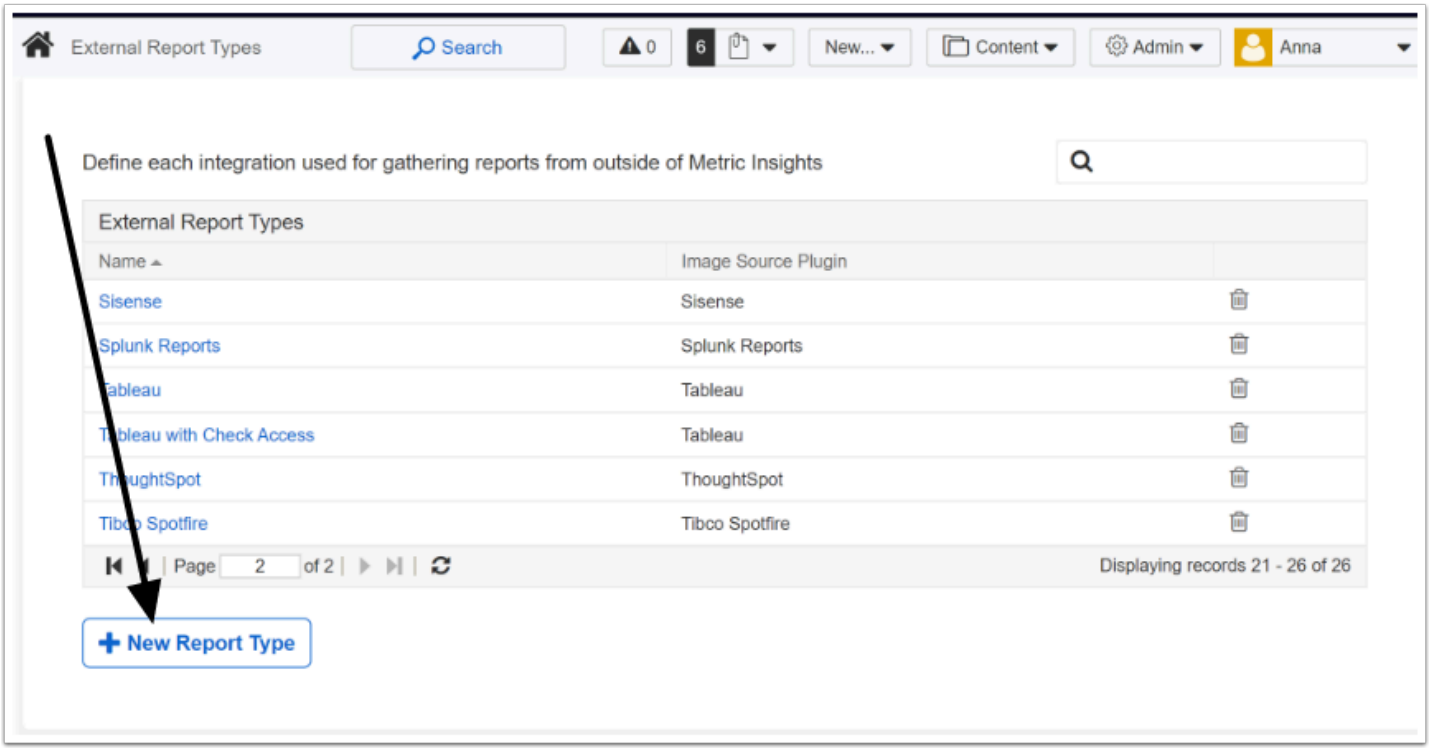
3.5 Create a new External Report Type

Each External Report Type is based on the plug-in connection used as its data source.

When any new External Report is created, the **Report type** is assigned to it. Report type determines generic settings of the External Report (source of images, download ability, drill down authentication, etc.). A separate **Report Type** should be created per each plugin you plan to use as a data source for External Reports.

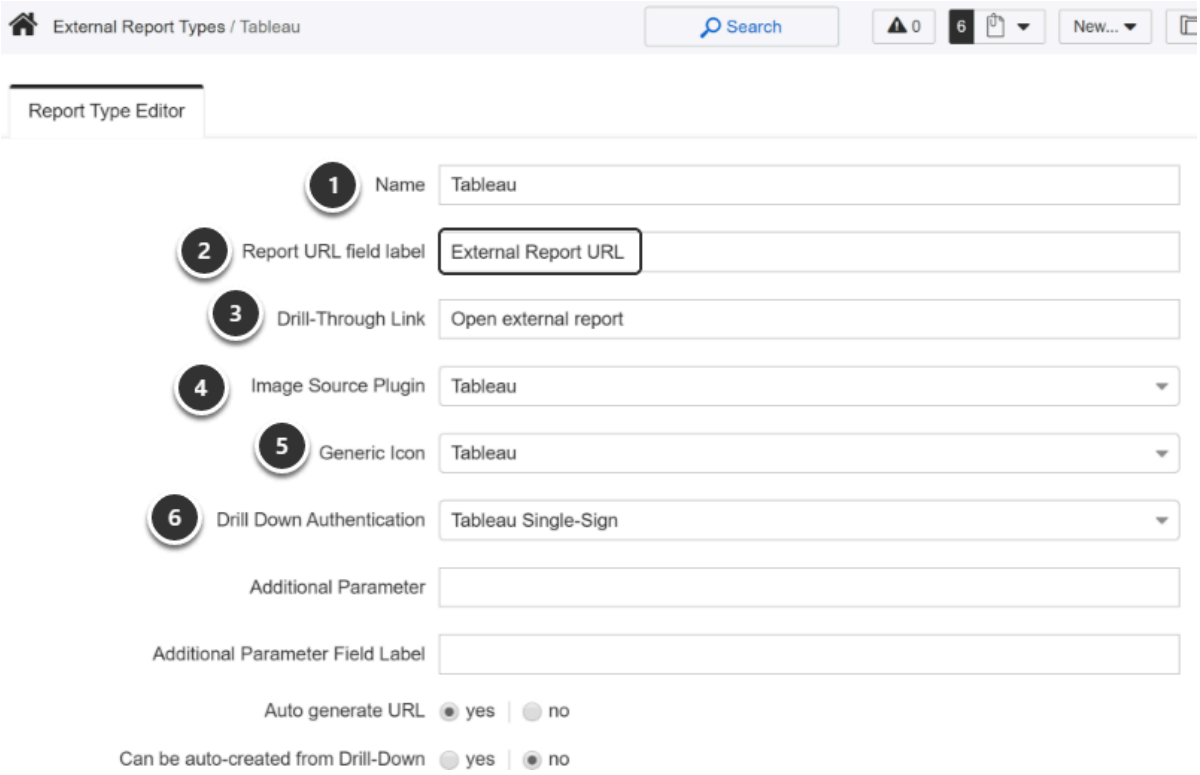
The list of plugins supported by Metric Insights can be found here.

Access Admin > Plugins > External Report Types

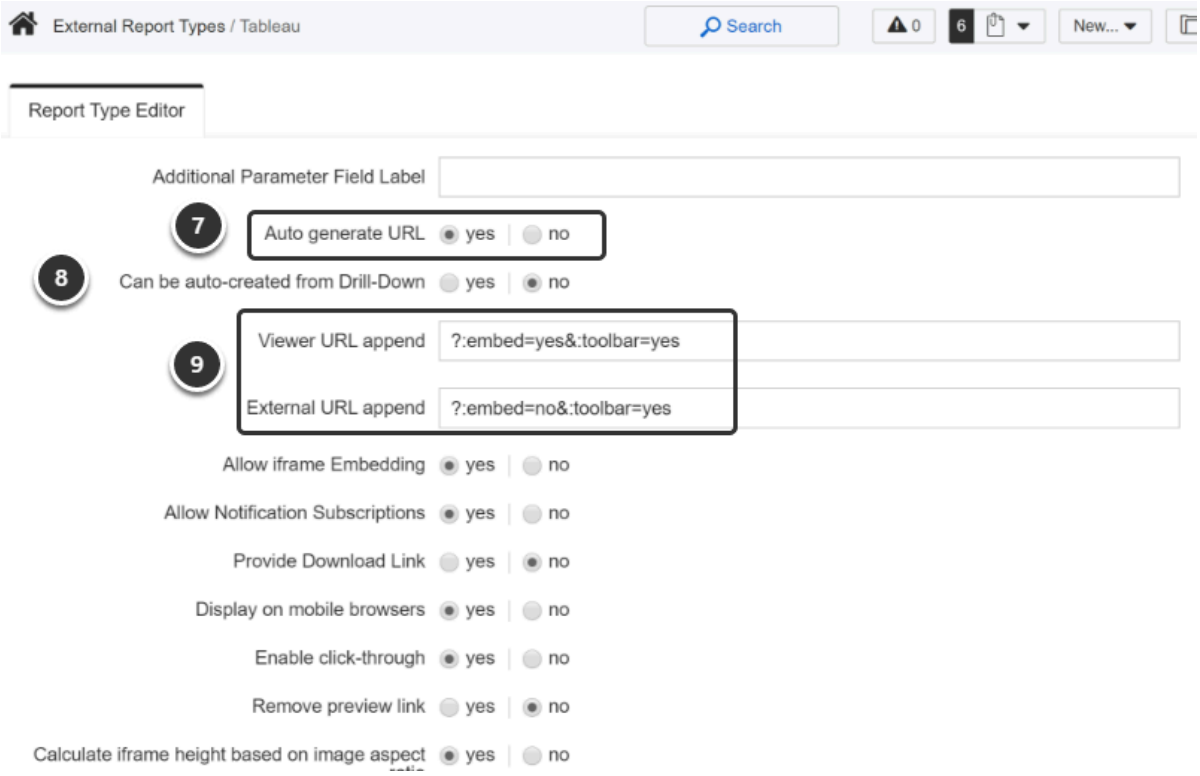


Click **[+ New Report Type]** to open Editor

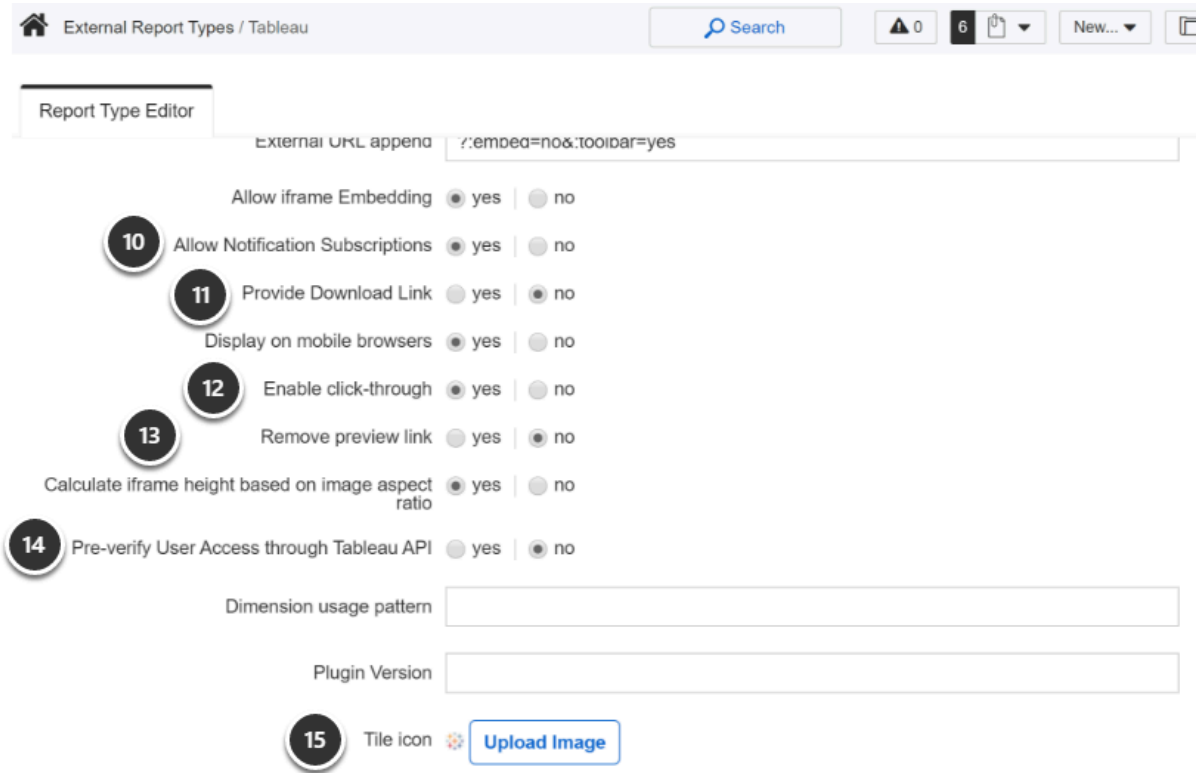
2. Provide information essential for creating a new external report type



1. Provide a meaningful **Name** of the External Report Type. We recommend including the name of the plugin to be used as a data source into the name.
2. **Report URL field label:** Provide the name for the field specifying the report source. This field is going to be used at the *External Report Editor*.
3. **Drill-Through link:** Provide the label for the URL to the report source. This link is shown below the External Report Viewer as shown in the picture above.
4. **Image Source Plugin:** Define which of the source plugins supported by Metric Insights should be used to fetch images for reports of this type.
5. **Generic Icon:** Placeholder image. If for some reason no image has been generated for the External Report of this type, the placeholder image, selected in this field is going to be shown.
6. **Drill Down Authentication:** Specify whether drill-down authentication should be enabled for the plugin. If the user is authenticated on the server of the corresponding plugin, he should not provide his credentials to preview the report in MI Viewer. **NOTE:** This option is currently supported for *Tableau* and *1010 data* plug-ins only. To request it for other plug-ins, contact Metric Insights support team at support@metricinsights.com.



7. **Auto generate URL:** *This field is available for specific plugins.* If this option is active, a link to the source of the External Report in the *External Report Editor* is going to be automatically generated based on the selections in the fields above (as shown in the picture). If this setting is set to 'no', a full URL should be provided manually. **NOTE:** Hint box next to this field is going changed based on the respective selection.
8. **Can be auto-created from Drill-Down:** This setting is used to allow or prohibit auto-creating external reports from drill-down. **NOTE:** These reports should be created as **Enabled** and **NOT visible in homepage**. For more details refer to: [Creating a new External Report for Drill-down on the fly]
9. **Viewer URL append / External URL append:** If a user drills through to the external application, additional parameters can be added to the URL.



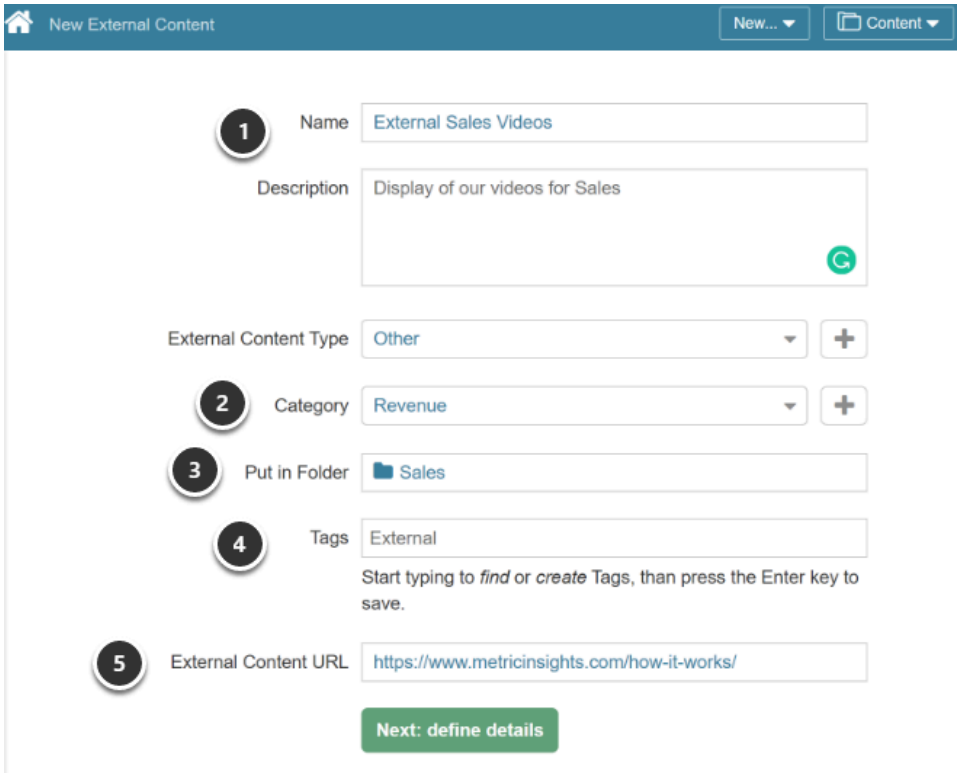
10. **[6.1.1] Allow Notification Subscriptions:** Defaults to "yes", if set to "no", the icon for Notifications will not be displayed on Report Viewer (envelope icon).
11. **Provide Download Link:** Define whether you want to enable an ability to download an external report from the Viewer
12. **Enable click-through**: *This setting relates to **Display on mobile browsers** and is shown only if 'yes' is selected there.* If click-through is enabled, and a user clicks a tile of the external report, he is redirected to external application, the same way as with the regular version. If this setting is disabled, a user won't be taken to the external application.
13. **Remove preview link**: If the preview link is removed, external reports of this type are not set up for viewing on the server. They should be downloaded and viewed on the desktop only. If this option is set to 'no', such a report may serve as a link to an external application.
14. **Pre-verify User Access through API:** Setting this field to 'yes' will allow Metric Insights to verify authorization to plugin element before trying to display the element. Note: this option is currently only available for Tableau and requires that the plugin be configured correctly Verify if Report Type has been setup to allow verification of Access
15. **Tile icon**: Click **Upload Image** and choose the picture with the logo of the service you create a Report Type for. If you need to update the existing picture, there is no need to delete a previous one, simply re-upload it with the same procedure.

3.6 How to link to Other External Content

This article describes creating a new 'External Content' that will be linked from a Metric Insights. If you wish to drill to External content based on dimension values, look at how to set up drill paths for external reports.

NOTE: The format of the *External Content Editor* is slightly different in earlier versions.

1. Access New > External Content



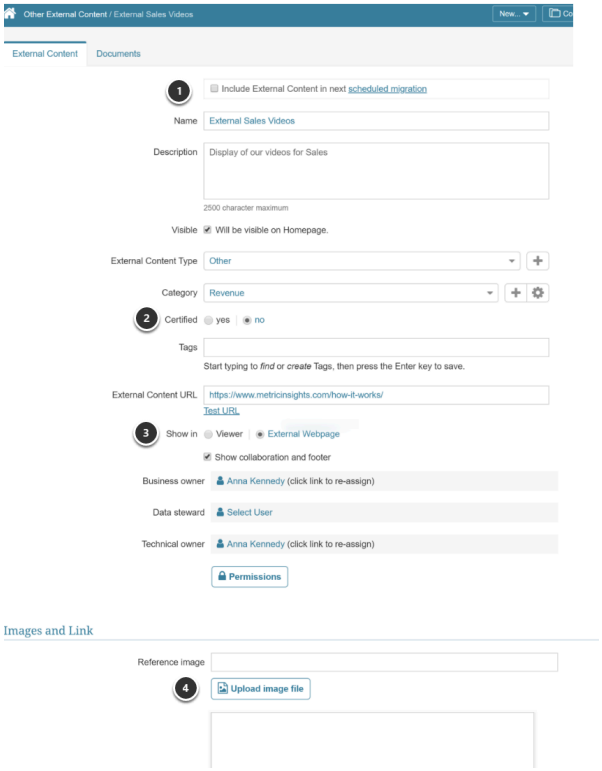
Metric Insights needs to know a few things about this particular content:

1. **Name** (required)
2. **Category** (optional)
3. **Folder** (optional)
4. **Tags** (optional)
5. **URL** (required) In this case, we'll just use a static web page that we've created for the purpose of this example:

That way, if you're looking at another monthly report that happens to look at some measure, your external content report will also show up to the user as a related report.

Next: define details

2. Update additional fields

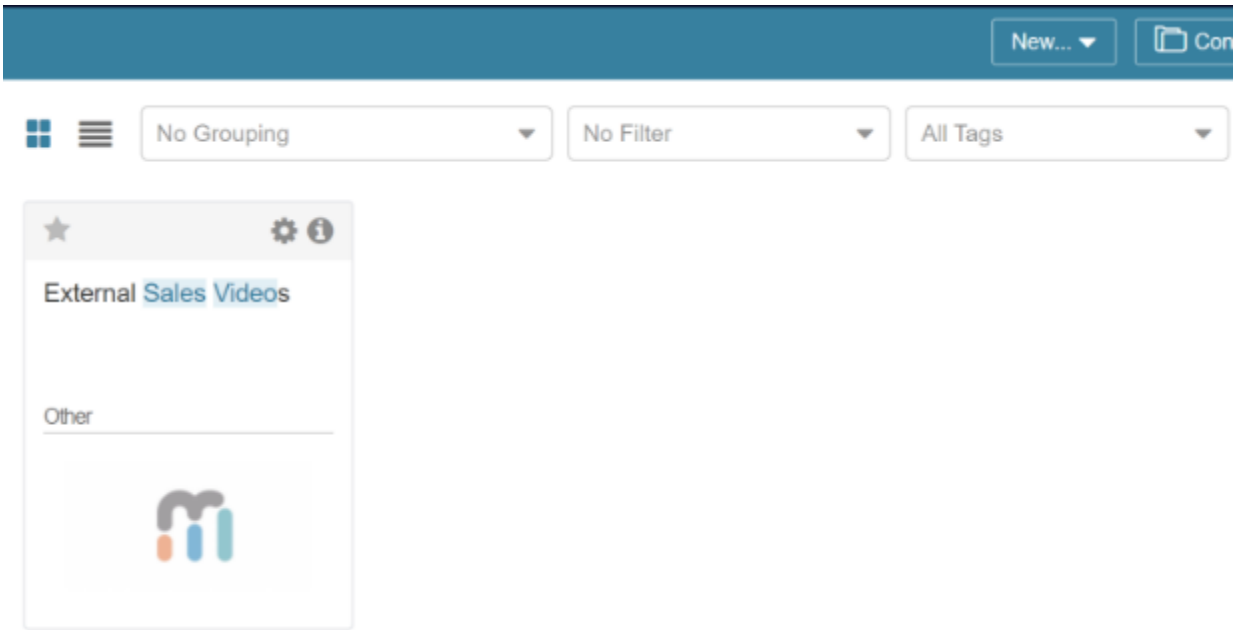


1. [6.1.0] As of Release 6.1.0, it is possible to flag elements and then **Migrate Content** using our Export/Import Migration Scripts. Click for details: [Scripted Migration via Category and Element Editors]
2. **Certified:** Certification is a means for Admin and Power Users to identify elements that have been approved as being valid and accurate. For details refer to: [Certifying an Element]
3. **Show in**: Select to display your content in either Metric Insights 'Viewer' or on the 'External Webpage'
4. Click **Upload image file** to display pop-up where you can add a Reference Image

3. Choose your Reference image

Take a screenshot of your report and upload it as the preview image. This image should be a .png file and to avoid distortion should have a width to height ratio of 1.8 (in other words, the width is nearly twice the height).

4. Verify the new external content

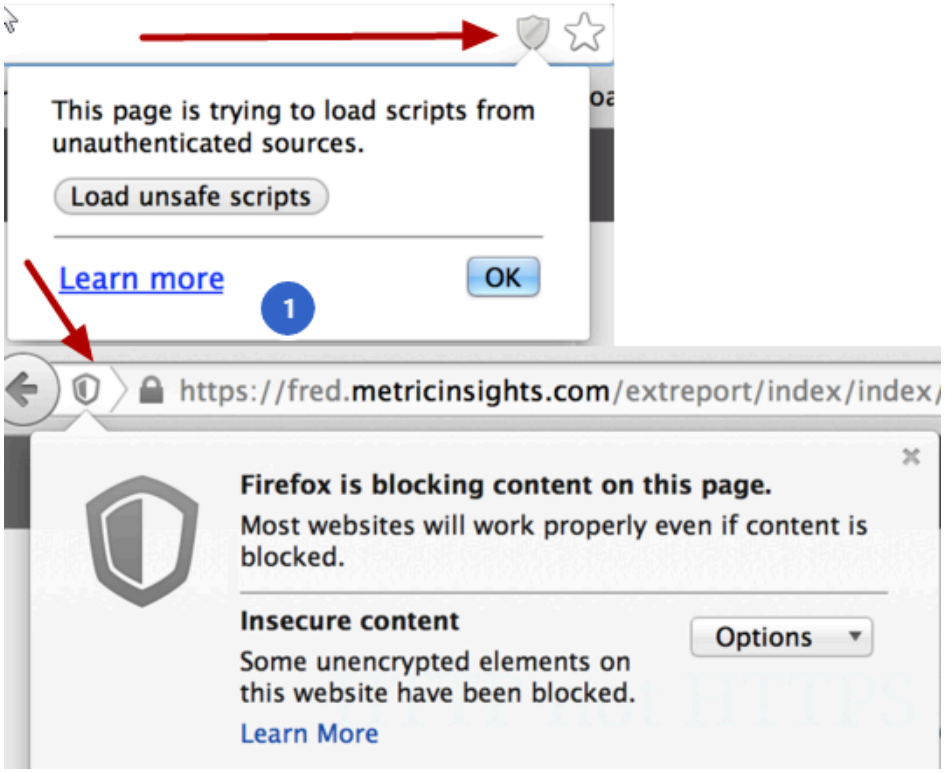


Now that it's enabled, make sure the new external content tile is available on the *Homepage*. Double clicking the tile should take you to the URL you expect.

3.7 Why do I see an empty page when I set my External Report to display in Viewer?

Metric Insights uses secure http, i.e., https. If the web application that serves as the source for your **External Report** uses ordinary http, you will not be able to display the report in the Metric Insights viewer. This is a fundamental security limitation of all browsers.

1. Verify that your External report is being blocked



Look for an indicator in the web browser address bar such as examples above (different for Firefox, Chrome,...)

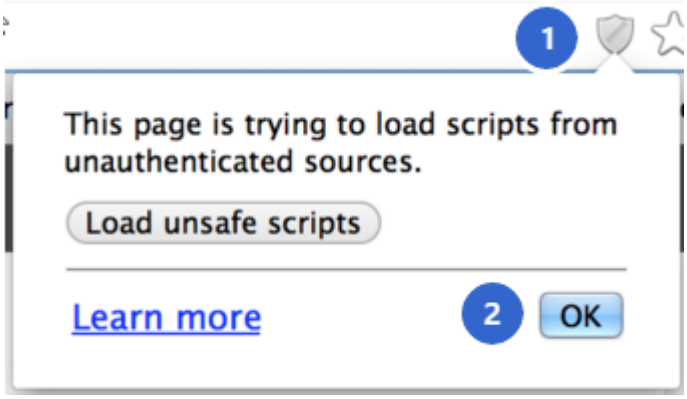
1.1. Alternatively you can open the 'console' for your browser

1. Look for it in Developer Tools
2. Then enter the URL for your **External Report.**

You should see a message like this:

[blocked] The page at 'https://your_metric_insights.site.com/extreport/index/index/element/101/from/editor/' was loaded over HTTPS, but ran insecure content from 'http://your_external_application.site.com': this content should also be loaded over HTTPS.

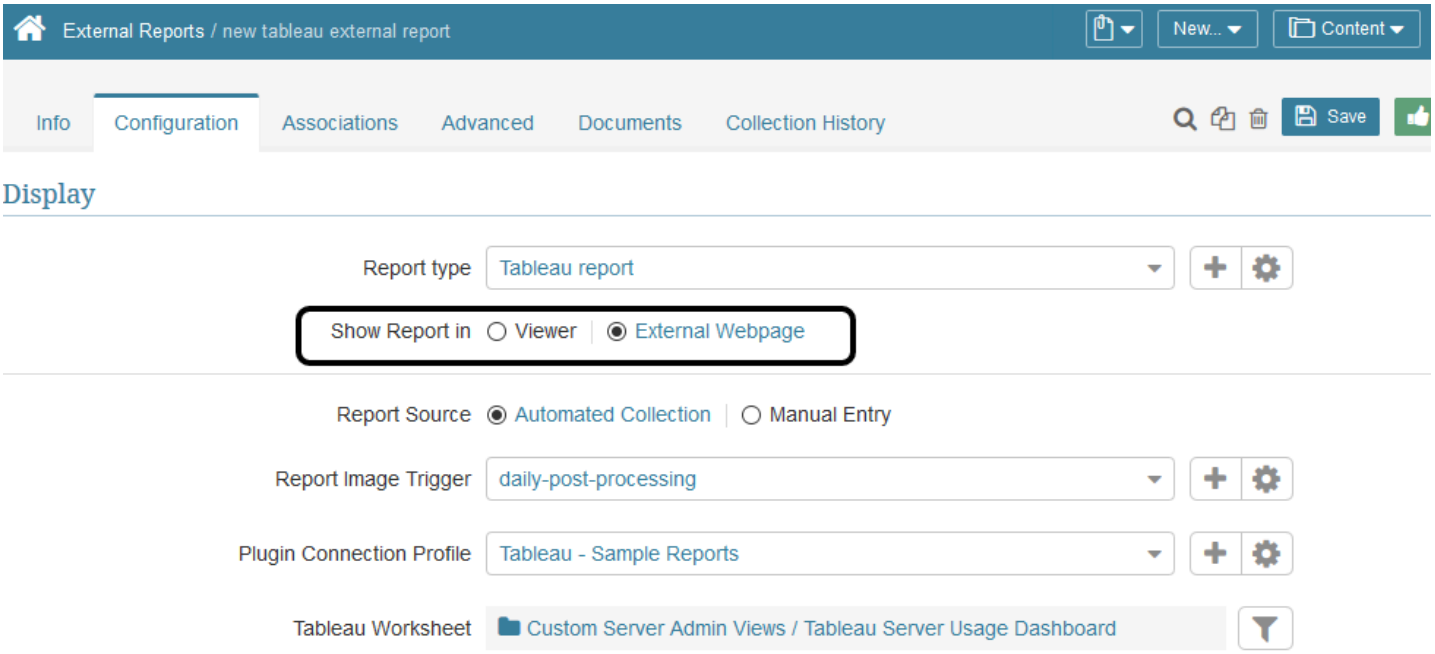
2. Short term solution



1. Click on icon in browser address bar
2. Click OK to permit the load

Note: Each time you get blank page, you must click on icon in browser address bar to allow unsafe (http) content

3. Long term solution



You will need to use **HTTPS** for your external web application if you want to display your External Report in the Metric Insights viewer

Alternatively, you can set your External Report to open in an **External Webpage** instead of in the Viewer

3.8 Set Custom Access Request Messages for External Reports

New in Version 5.6, our **Content Security and Discoverability** capabilities have been expanded to allow customization of Access Denied messages for content.

You may specify tailored messages, control where Requests for Access are routed, and are now able to support both centralized Access Request processing (via a tool like SAP's GRC) and distributed Access Request responsibilities via a Metric Insights-only workflow.

There are three different levels where **Access Denied options** can be set:

1. Element
2. Category

The system will first check the *Element Editor to* verify **Access** settings and the default No Access **tile format**. If the Report settings do not include a **Custom Access Message,** then the *Category Editor* will be accessed to locate a **Custom Message**. The first Message encountered will be used.

3. BI Tool authorization. This will only be checked when User already has access to External Report in Metric Insights.

This article will explain how to apply Customized Access Request processing for each Editor. The system will default to standard error messaging if no customization has been set up, see Provide users with the ability to request access to inaccessible content

In this initial implementation, the 3rd option (validation for BI Tool Authorization) is only available for Tableau and only when the Username is passed through SAML.

1. The basics are determined by company's Request Access procedures

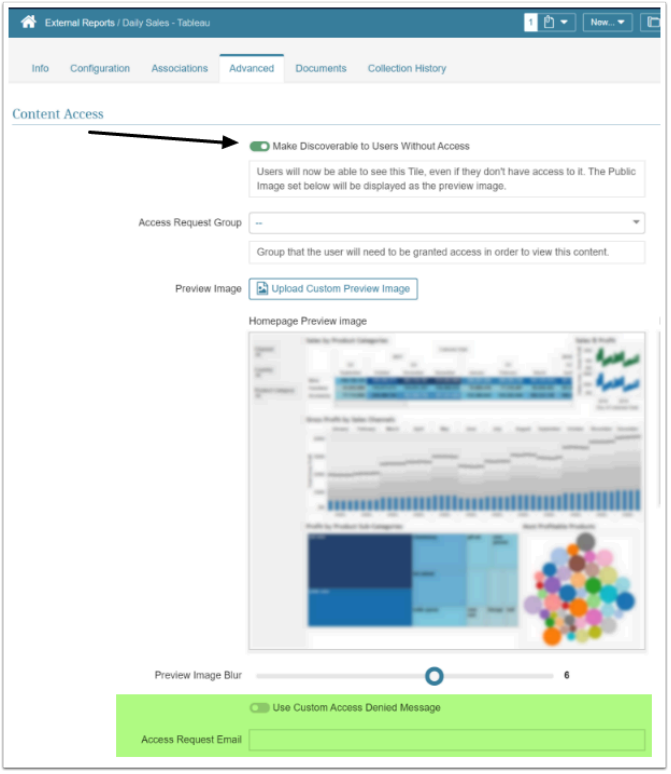
1.1. Access Request Flow 1: Manage Within Metric Insights

Set the required Access-request variables in the Config file (Admin > Utilities > System Config):

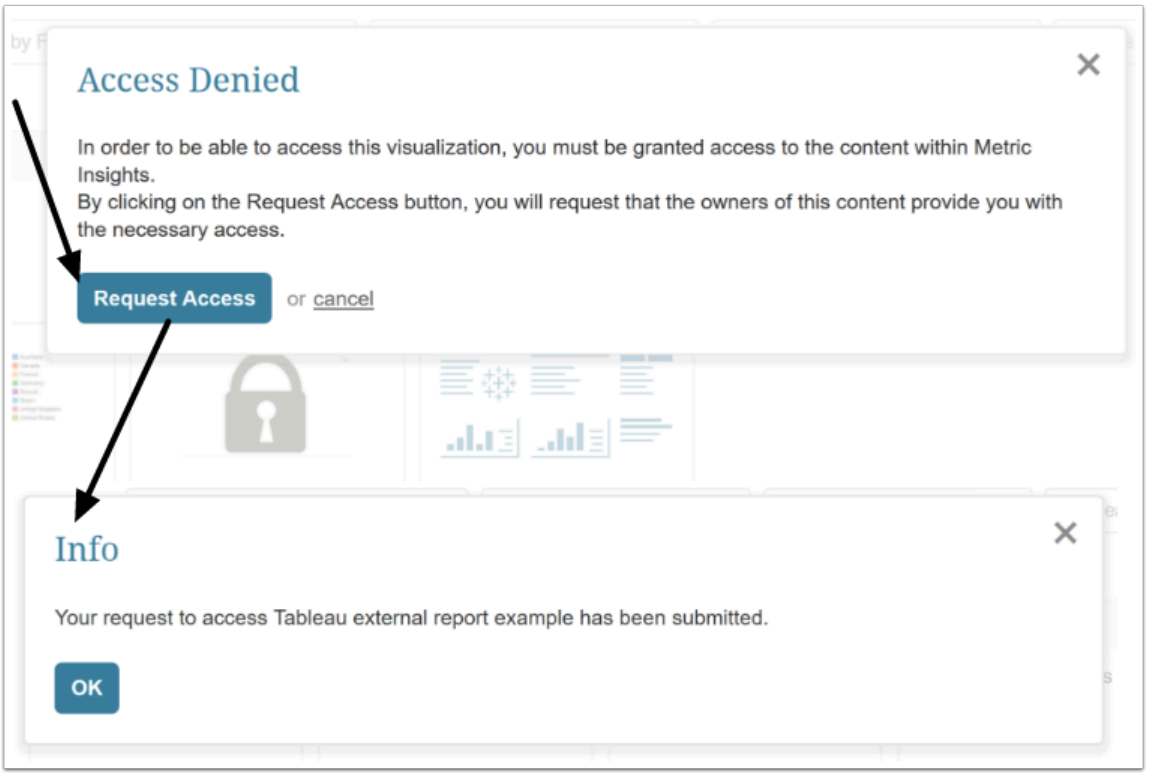
1. Type '**access**' in the Search field to narrow your options
2. Set '**SHOW_ITEMS_WITHOUT_ACCESS**' to '*Y*'
3. Set '**SEND_ACCESS_REQUEST_DIGEST**' to '*Y*'
4. Set '**ACCESS_REQUEST_VIA_WEBPAGE**' to '*N*'
5. ***Commit*** your changes

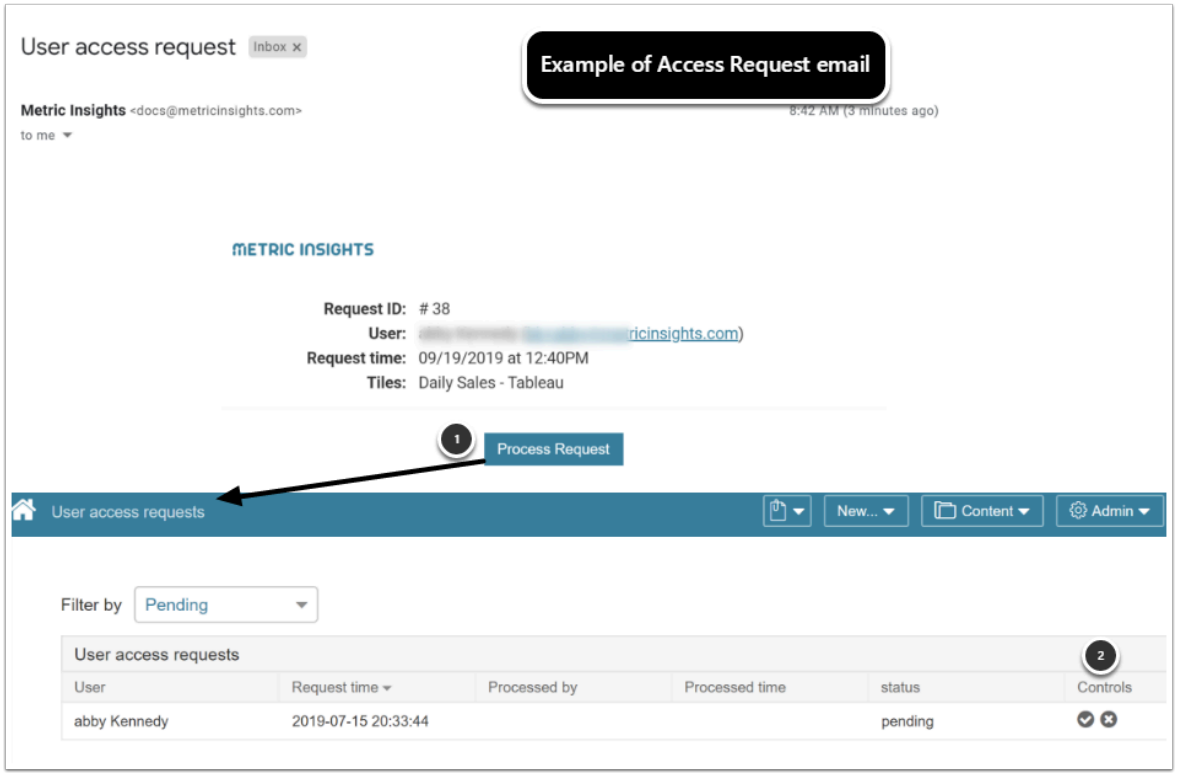
Note: Flow 1 additional settings:

1. (Required) Setting the External Report to be *Discoverable to Users Without Access* - see Step 2 Set Custom Access Request fields on External Report Editor.
2. (Optional) Set Custom Access Request ***message and /or email*** at either the External Report element level or the associated Category level.



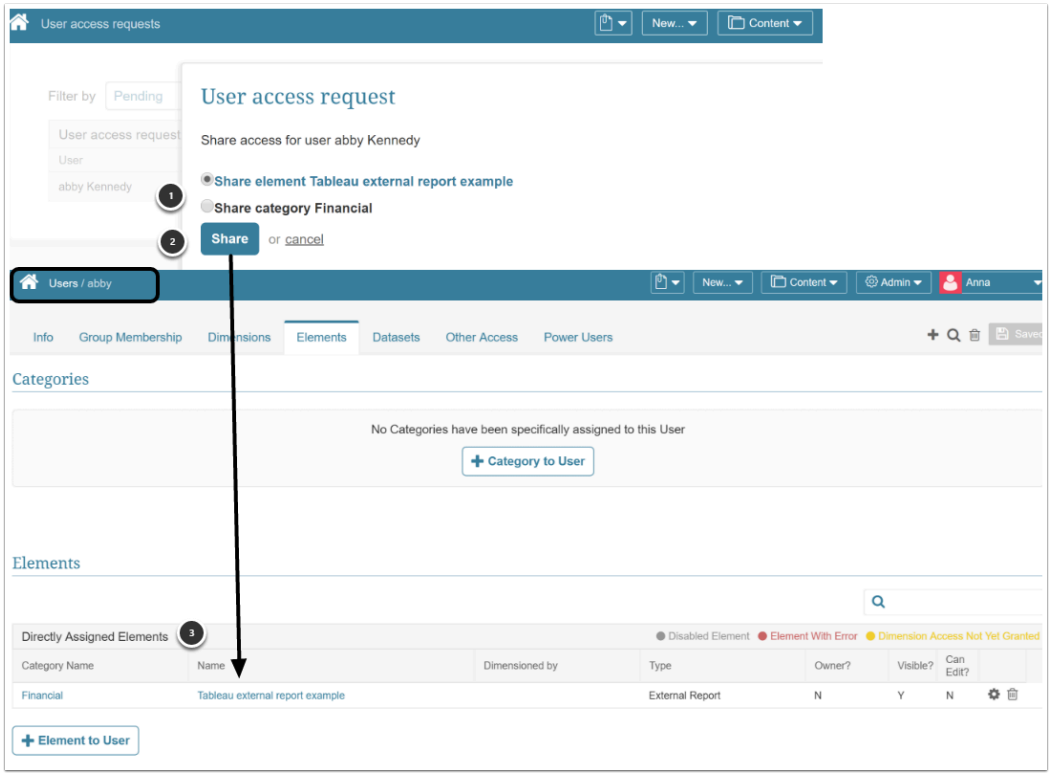
Example of "Request Access" process with default message. This request email will be sent to *either* the ***Access Request Email*** if specified in Steps 2 or 3, or will default to your Support Admins' emails.





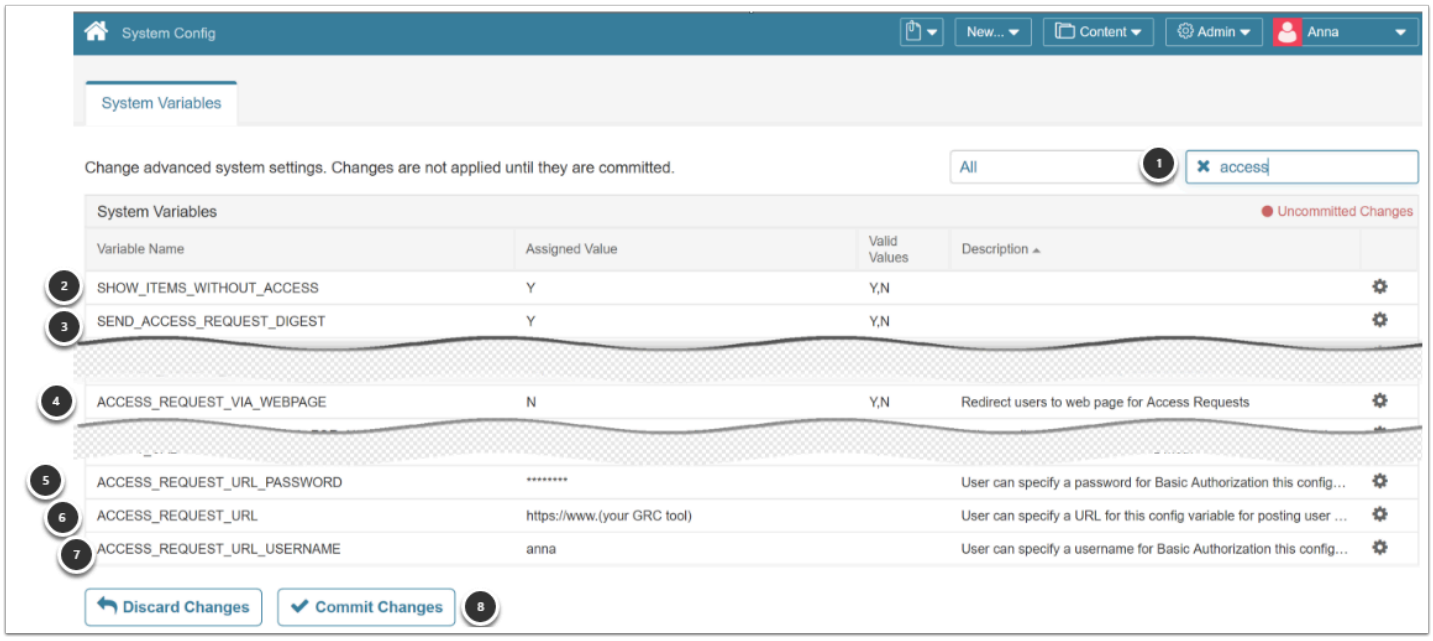
1. Select [***Process Request]*** to access the ***User access requests*** grid
2. Select one of the two **Control** icons to process or deny this request
   1. X mark will deny access (no further processing)
   2. Check mark - will process the request as shown below

Either way this request will be removed from the grid



1. **Select** whether you want to grant Access to the specific **element** or to all elements within the **Category**
2. Share to complete request
3. Metric Insights will **automatically add requested element or Category** to this User

1.2. Access Request Flow 2: Manage via Access Request API (like SAP's GRC tool)



Set the required Access-request variables in the Config file (Admin > Utilities > System Config):

1. Type '**access**' in the Search field to narrow your options
2. Set '***SHOW_ITEMS_WITHOUT_ACCESS***' to 'Y'
3. Set '***SEND_ACCESS_REQUEST_DIGEST'*** to 'Y' (optional)
4. Set '***ACCESS_REQUEST_VIA_WEBPAGE***' to 'N'
5. Set '***ACCESS_REQUEST_URL_PASSWORD***' to a valid password for the URL_USERNAME
6. Set '***ACCESS_REQUEST_URL'*** to the end-point of your Access Request API
7. Set '***ACCESS_REQUEST_URL_USERNAME***' to a valid User for API authentication
8. ***Commit*** your changes

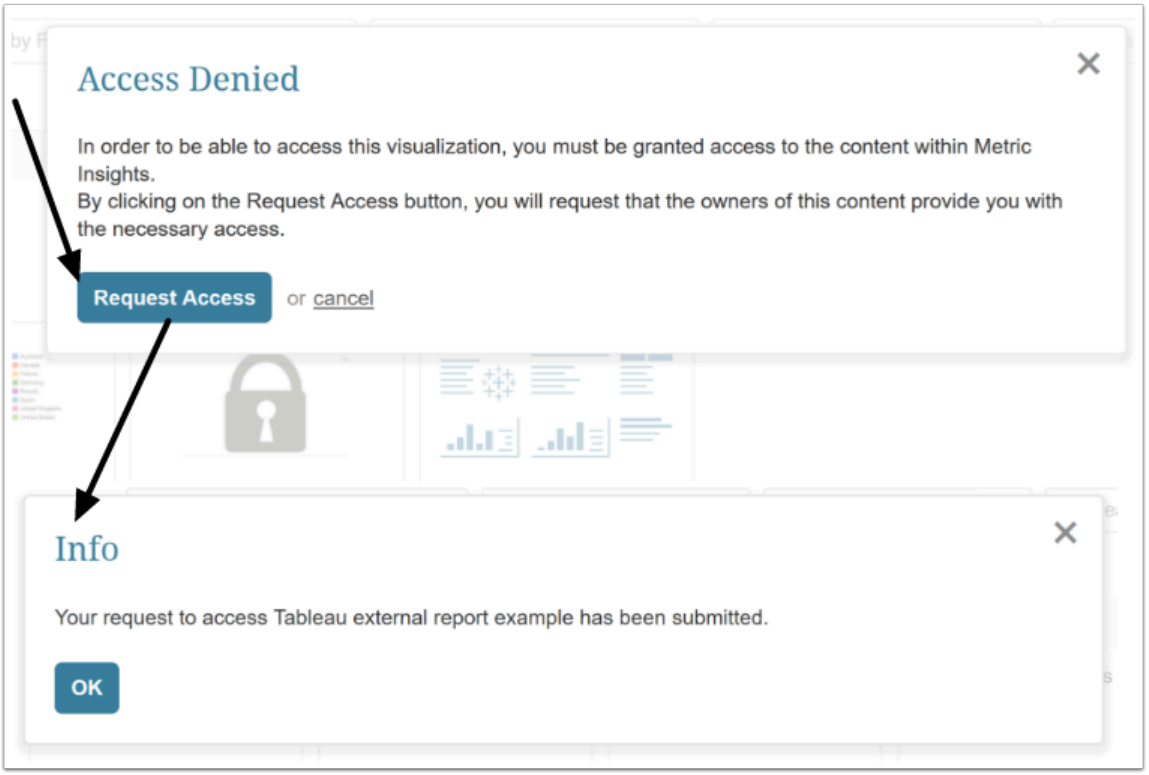
When the "Request Access" button is clicked, a request will be sent to the endpoint set via "***ACCESS_REQUEST_URL***", passing Access Request Group set on the External Report Editor (optional, mentioned below), and a request email will be sent to Support Admins (if '*SEND_ACCESS_REQUEST_DIGEST'* = 'Y'). (see Access the Advanced tab of the External Report Editor).

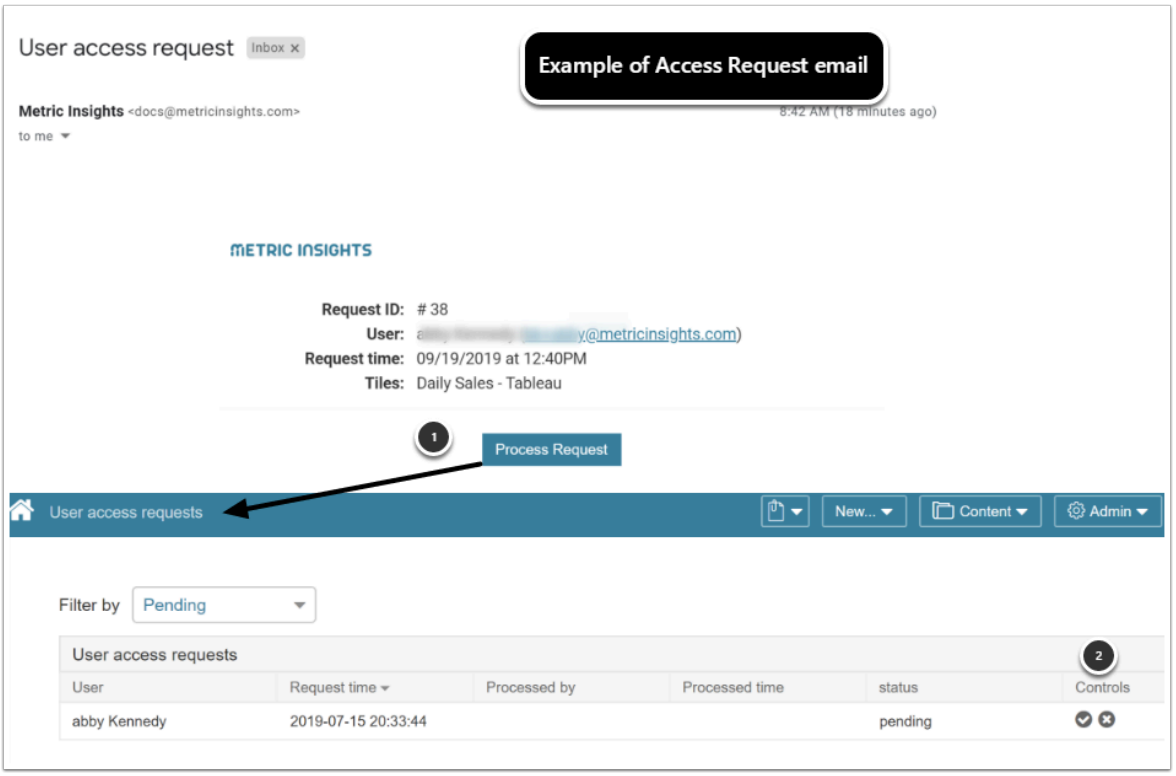
Note: Flow #2 additional settings:

1. (Required) In the External Report Editor: see step 2
   1. Toggle "**Discoverable to Users Without Access"** to **'*ON*'**
   2. Set the "**Access Request Group**" to a valid LDAP group. This determines the group the user needs to be added to in order to gain access the content. Only applies in cases where Metric Insights and LDAP are regularly synced.

2. (Optional) Set **Custom Access Request** message and/or email at either the External Report element level or the associated Category level. See Steps 2 and 3 below.

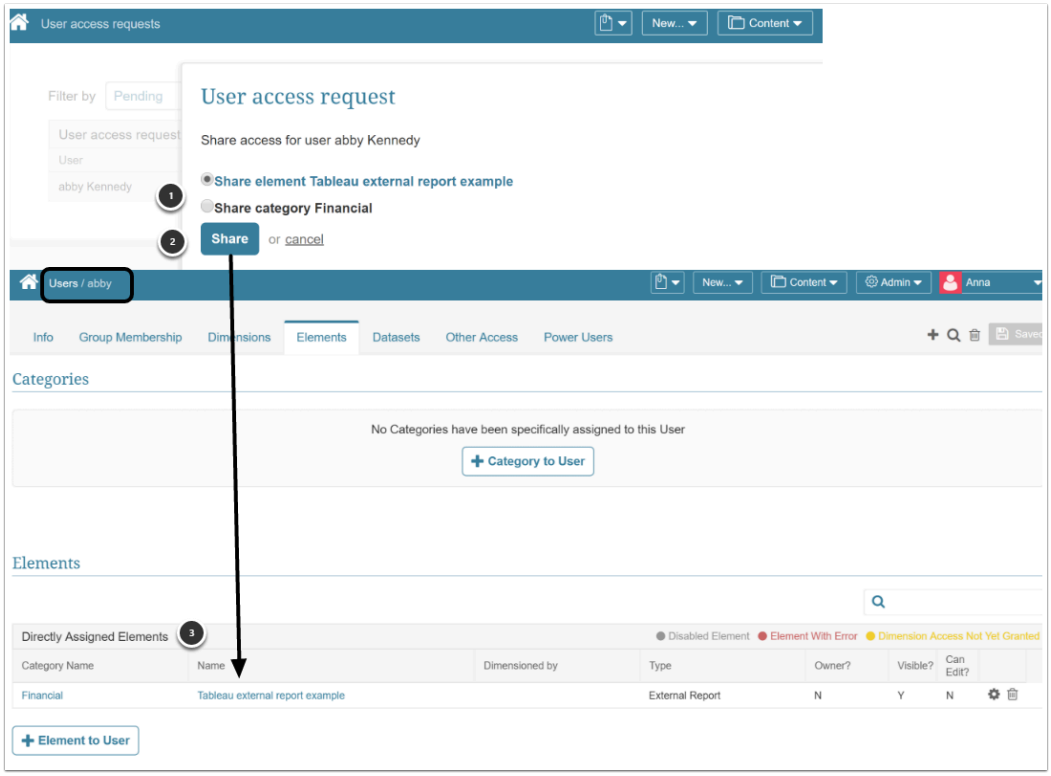
Example of "Request Access" process. This request email will be sent to either the ***Access Request Email*** if specified in Steps 2 or 3, or will default to your Support Admins' emails.





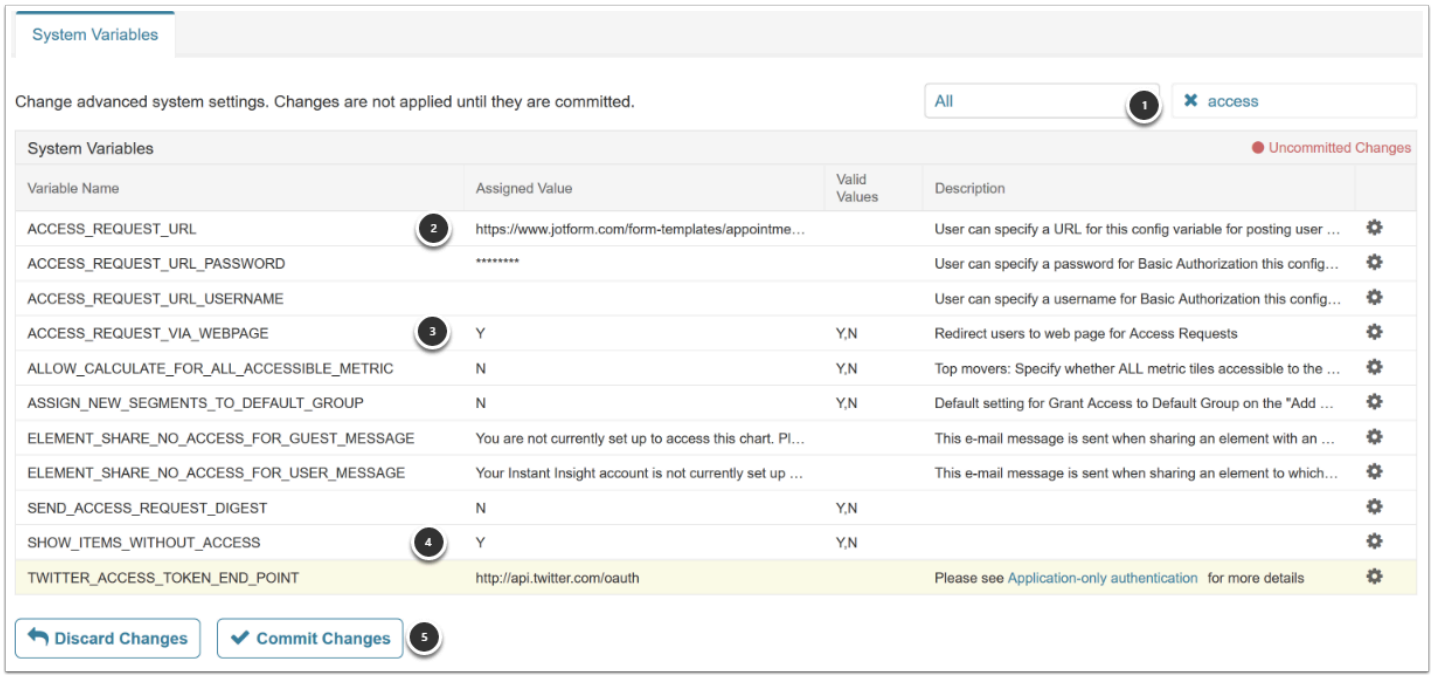
1. Select ***Process request*** to access the **User access requests** screen
2. Select one of the two **Control** icons to process or deny this request
   1. X mark will deny access (no further processing)
   2. Check mark - will process the request as shown below

Either way this request will be removed from the grid



1. **Select** whether you want to grant Access to the specific element or to all elements within the Category
2. Share to complete request
3. Metric Insights will **automatically add requested element or Category** to this User

1.3. Access Request Flow 3: Manage via External Form/ Webpage



Set the required Access-request variables in the Config file (Admin > Utilities > System Config):

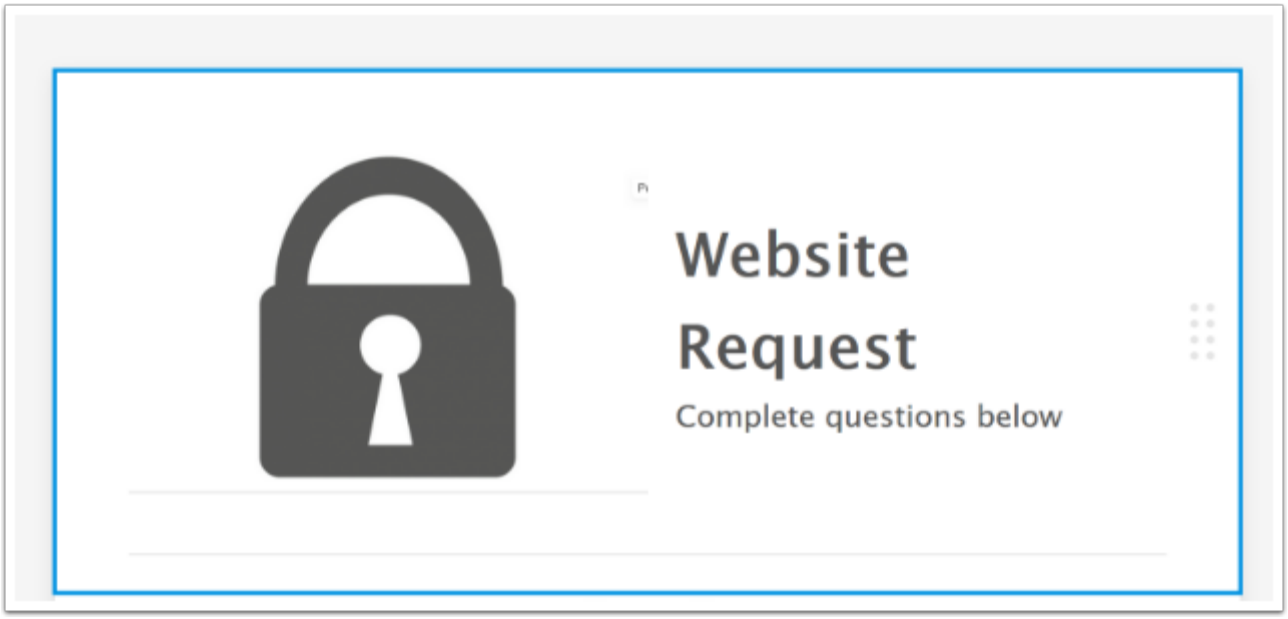
1. Type '**access**' in the Search field to narrow your options
2. Set '***ACCESS_REQUEST_URL***' to [a webpage set to process Access Requests]
3. Set '***ACCESS_REQUEST_VIA_WEBPAGE***' to 'Y'
4. Set '***SHOW_ITEMS_WITHOUT_ACCESS***' to 'Y'
5. ***Commit*** your changes

Note: Flow #3 additional settings:

1. (Required) Setting the External Report to be **Discoverable to Users Without Access** - see Step 2.

NOTE: Any **Custom Access Request messages or email** set either on the External Report Editor or the associated Category Editor will be ignored

When "Request Access" button is clicked, User will be sent directly to specified Web page and access will be processed via your company's standard method. (below is just an example)

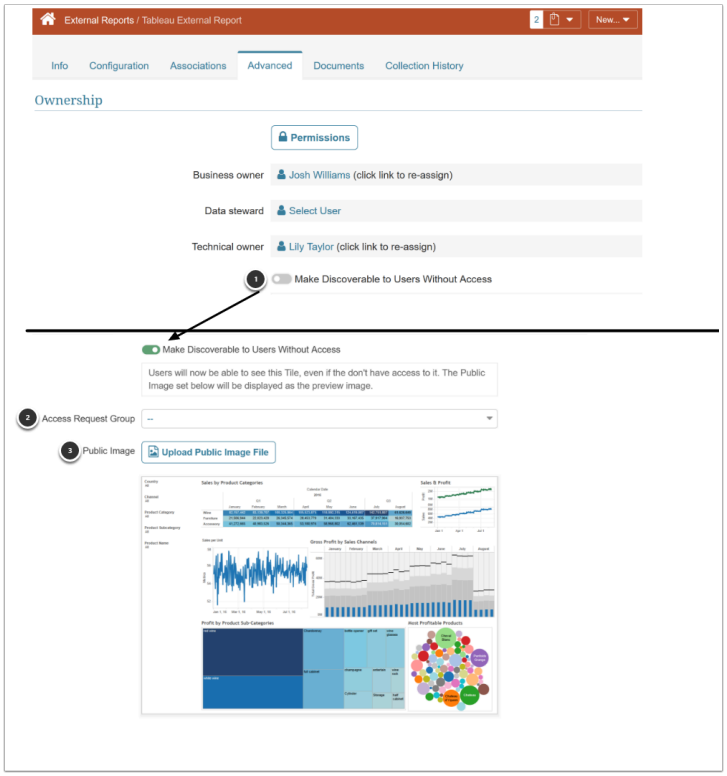


2. Set Custom Access Request fields on External Report Editor

Create a New Tableau Report (*New* > *External Report),* Select the **Tableau** Report Type, and follow instructions in [How to create an External Report from Tableau.]

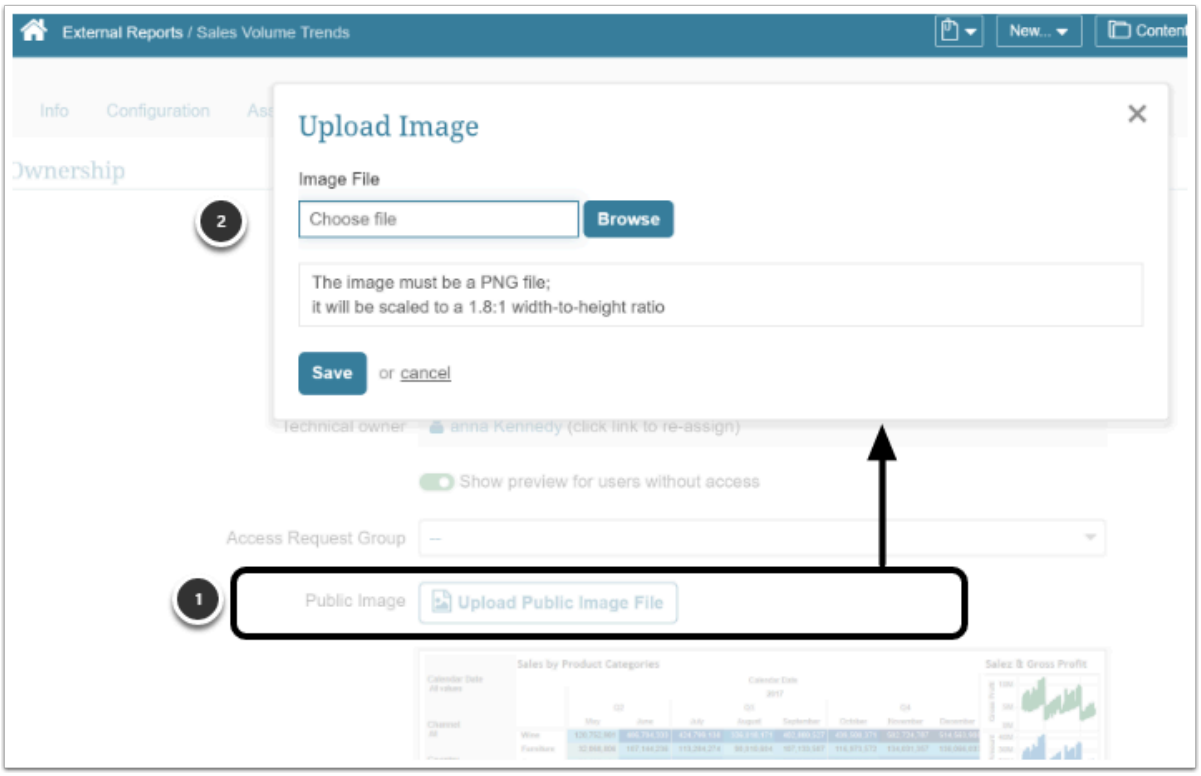
Or access an existing External Report from your Homepage via the edit icon (gear).

2.1. Access the Advanced tab of the External Report Editor

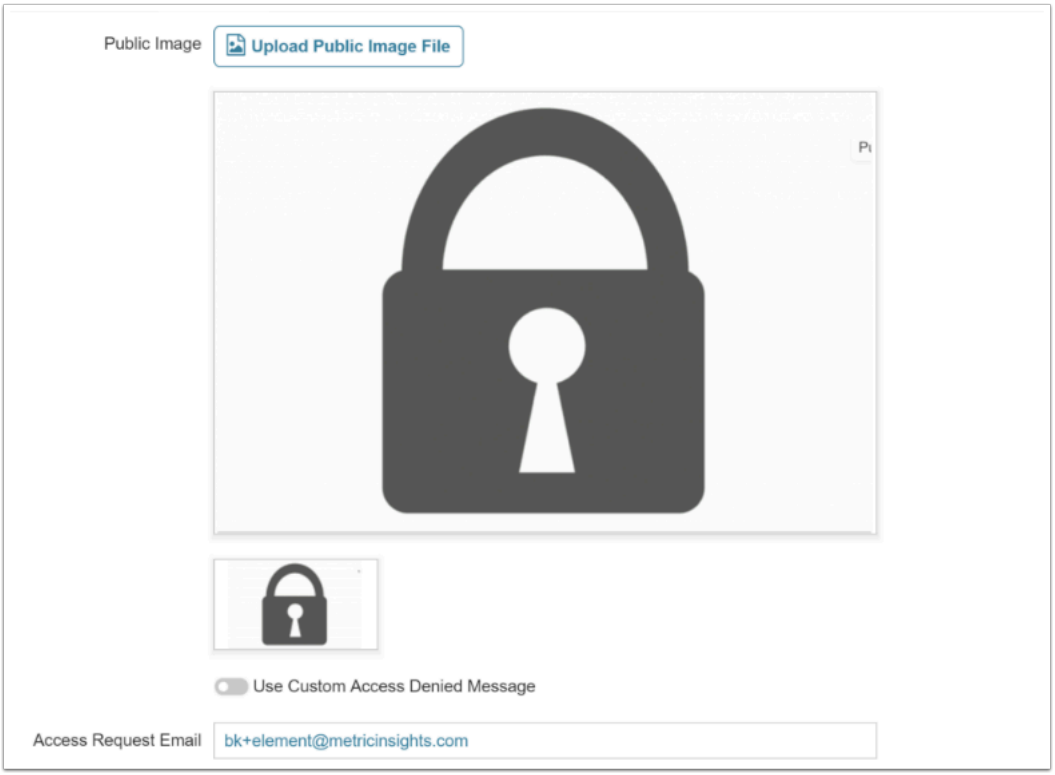


1. Toggle "**Make Discoverable**" to On to open additional fields
2. **Required for Flow #2**. If you utilize an API for managing the Access Requests, select the **LDAP Group** that the User should be added to in order to access this content. Other flows will ignore this field value.
3. (Optional)To apply a distinct display for inaccessible elements on the Homepage, select '*Upload Public Image File*' to open a standard ***Upload Image*** popup. Otherwise the default image from Configuration tab will be displayed.

2.2. (Optional) Select your own distinct image for Discoverable Homepage tile.

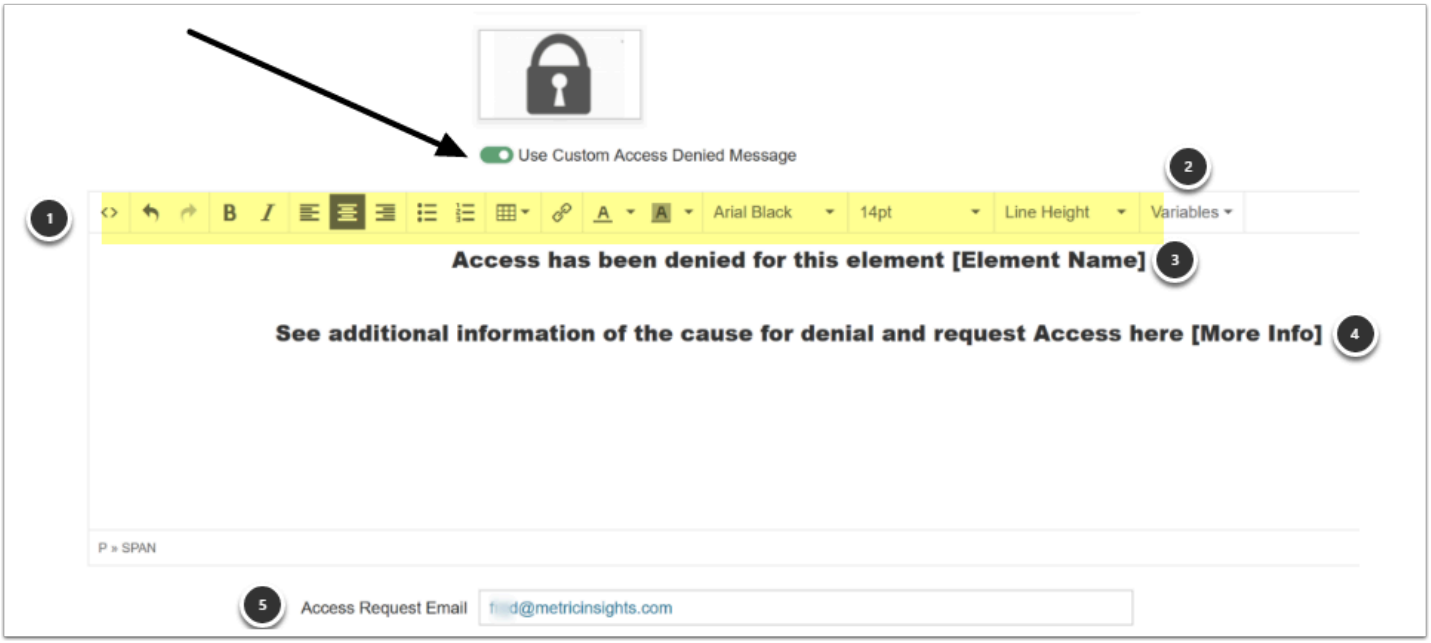


1. Select '*Upload Public Image File*' box
2. The ***Upload Image*** pop-up will appear to allow you to select an alternative image.



2.3. (Optional) Customize the Access Deny message

Note: if you do not specify a Access message at the External Report level, the system will check the associated Category for a message. If neither is specified, the Standard Message will display.



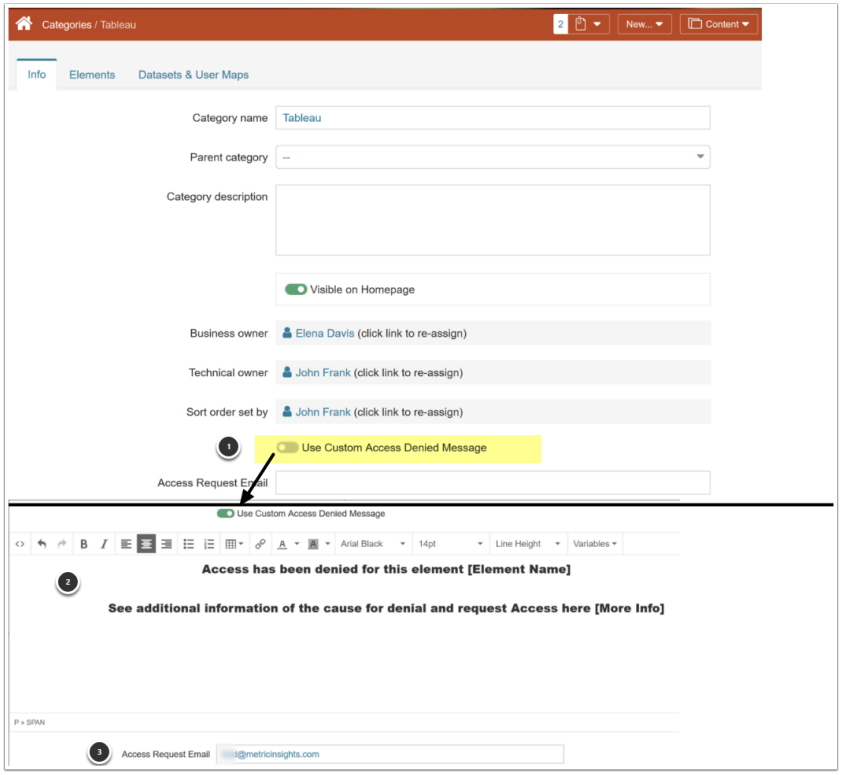
Toggle the **'Use Custom Access Denied Message'** to '*ON*'

1. You can control the Message display using the standard formatting options
2. You can insert the following variables using the dropdown:
3. **[Element name]** causes the system to substitute the ***External Report*** name in the display
4. **[More Info]** provides a link to show list of Support Admins that can assist you
5. (Optional) Provide an **Access Request email** address to be informed of this Access Denial. This field will over-ride the Standard option to send email to your Support Admin(s) for Access.

3. Set Custom Access Request fields on Category Editor

If **Access Denied Message** is NOT set at the *Element level*, the message set at the associated *Category level* for that element will be used. If Access Denied message is not set at either, the Standard Access Denied message will display.

3.1. Access Category Editor > Info tab

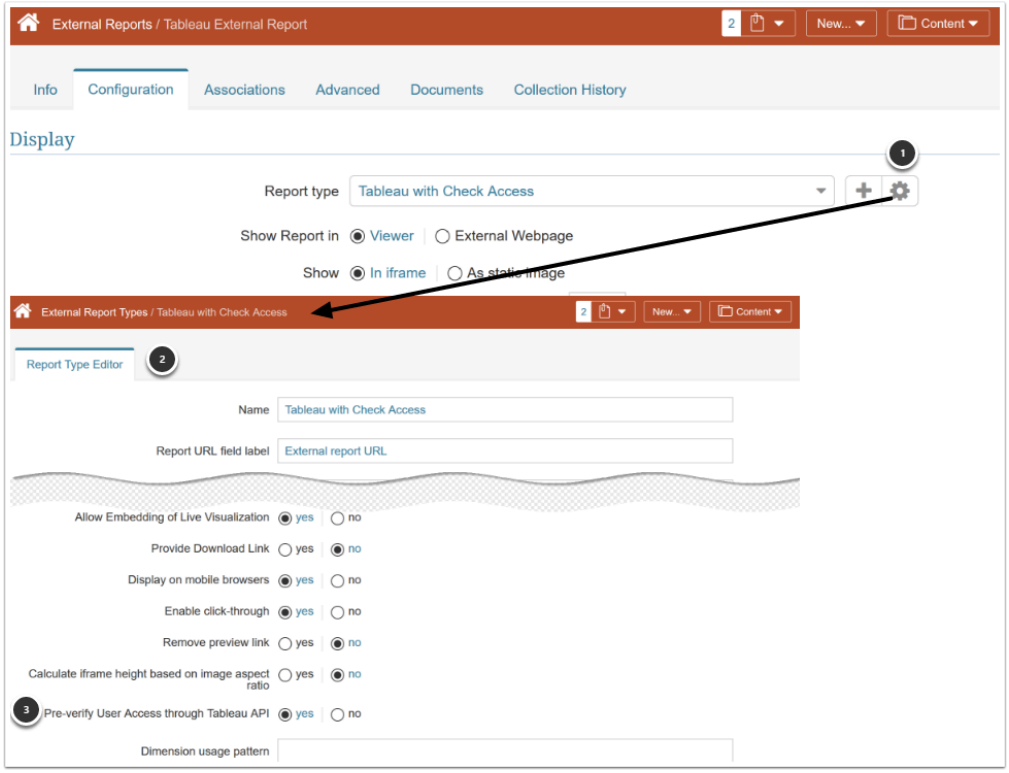


1. One the Info tab, toggle the '***Use Custom Access Denied message***' to ON
2. Follow the procedure for setting message as shown in Step 2.2 above
3. If **Access Request Email** as not been specified on the Report Editor, it may be set here and will apply to all External Reports set to **'Make Discoverable**' in this Category

4. Verify if Report Type has been setup to allow verification of Access

The first 2 levels of verification will be available for ***all*** External Reports. The Verify Option below is currently only implemented for Tableau and only for Request Flows 1 and 2.

4.1. Check that External Report is set correctly for Access check



This setting will allow Metric Insights to verify User Access to the specified Tableau Workbooks and Dashboards.

1. Click on edit gear to open ***Report Type Editor***
2. Scroll to bottom of **Editor**
3. Verify that ***Pre-verify User Access through Tableau API*** is set to 'Y' (Note: this option must be set up through the API by systems engineer; you should not modify this option)

4.2. The flow when User has access to Metric Insights for both the External Report and the Category, but is not authorized for requested Tableau workbook or dashboard. Note that Homepage will display a 'normal' tile image.

3.9 Handle "All Values" as Filter Value in Tableau

This article describes how to Apply the "All Values" setting to External Report and Send External Report with Burst.

Prerequisites:

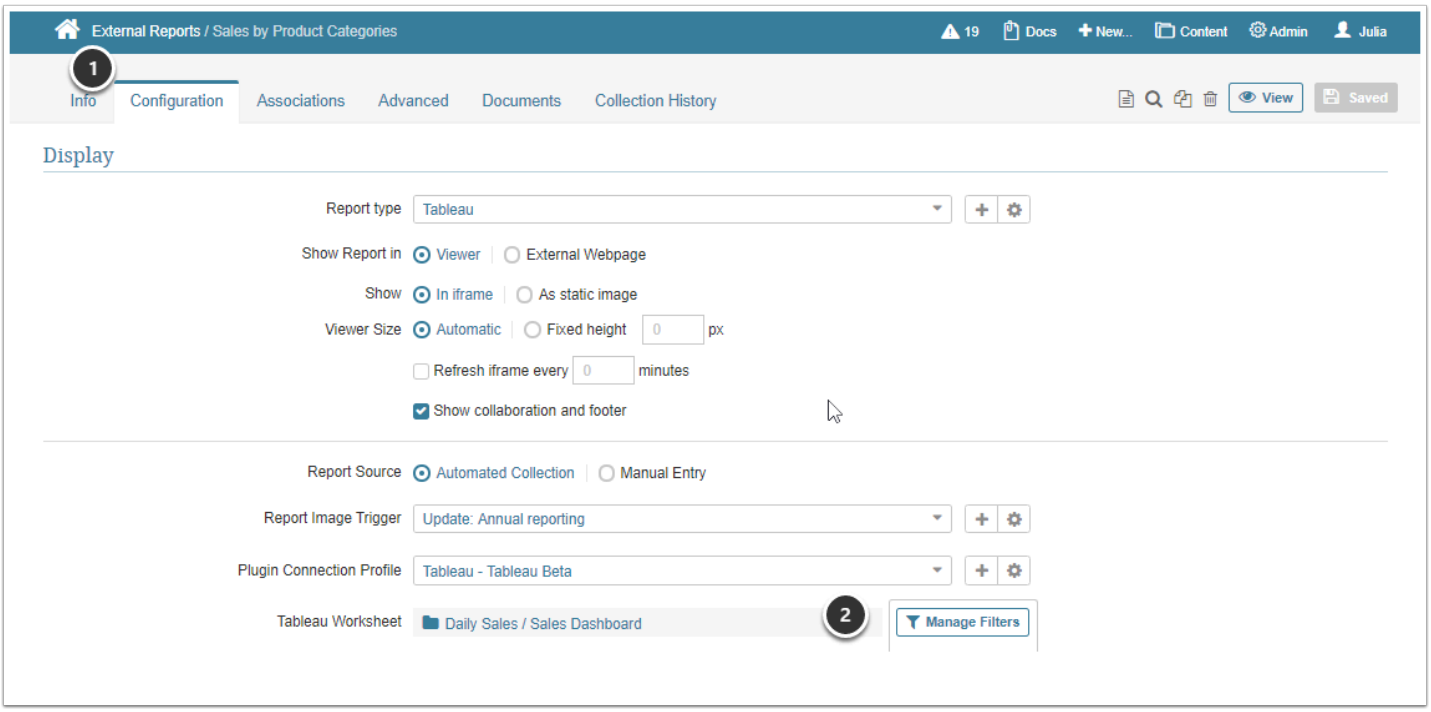
Before applying the "All Values" setting:

1. Create External Report as usual
2. Map a Tableau filter to External Report in Metric Insights. Once it is mapped, the External Report viewer enables the "All Values" setting.

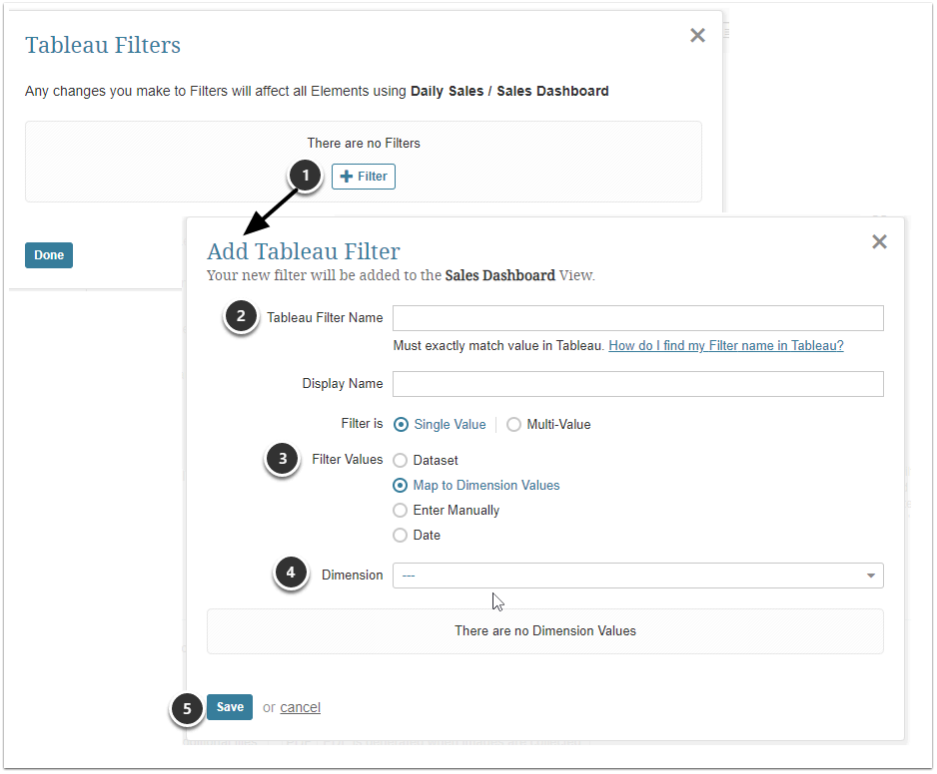
1. Apply "All Values" Setting to External Report

To learn more on how to apply filters to External Report, see also Filtering in External Reports (Pre-filtering BI tools).

1.1. Map Filters to Metric Insights External Report



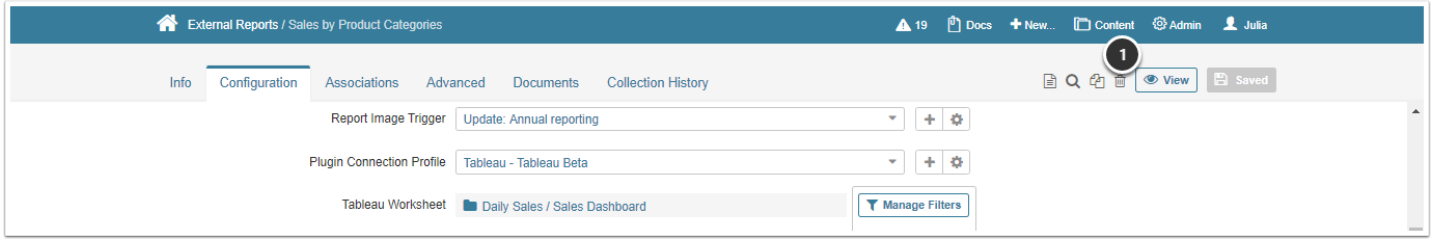
1. Go to [Configuration] tab in External Report
2. [Manage Filters]



1. [+ Filter] to create a new filter

2. Enter a Tableau filter name (case sensitive)
3. Choose "Map to existing Dimension Values"
4. Select a required value from the "Dimension" drop-down list
5. [Save]

> Dimension does not need to have "All" or "Total" values. Metric Insights displays the "All Values" setting in the Viewer drop-down list. All Values as a filter value setting are not shown in the Editor.



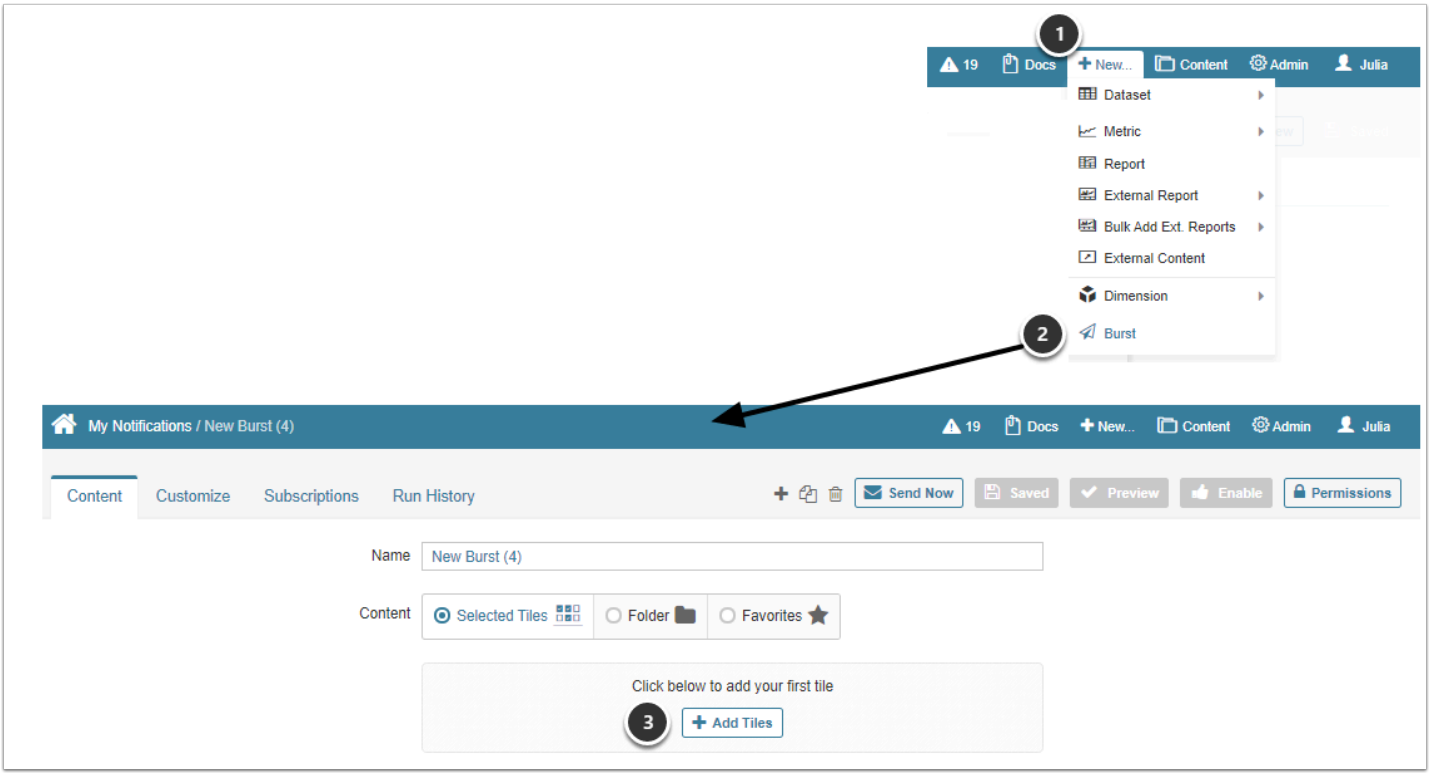
1. [View] to see the filter created and values applied

> To learn how to filter values based on User Map, see also Create a User Map article.

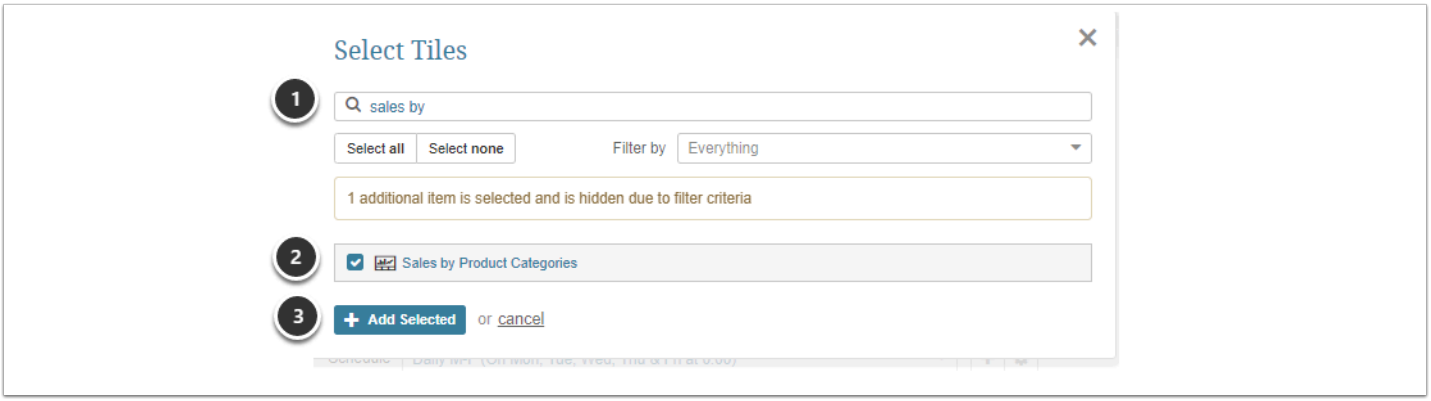
2. Send External Report with Burst

2.1. Create and Custom for Burst

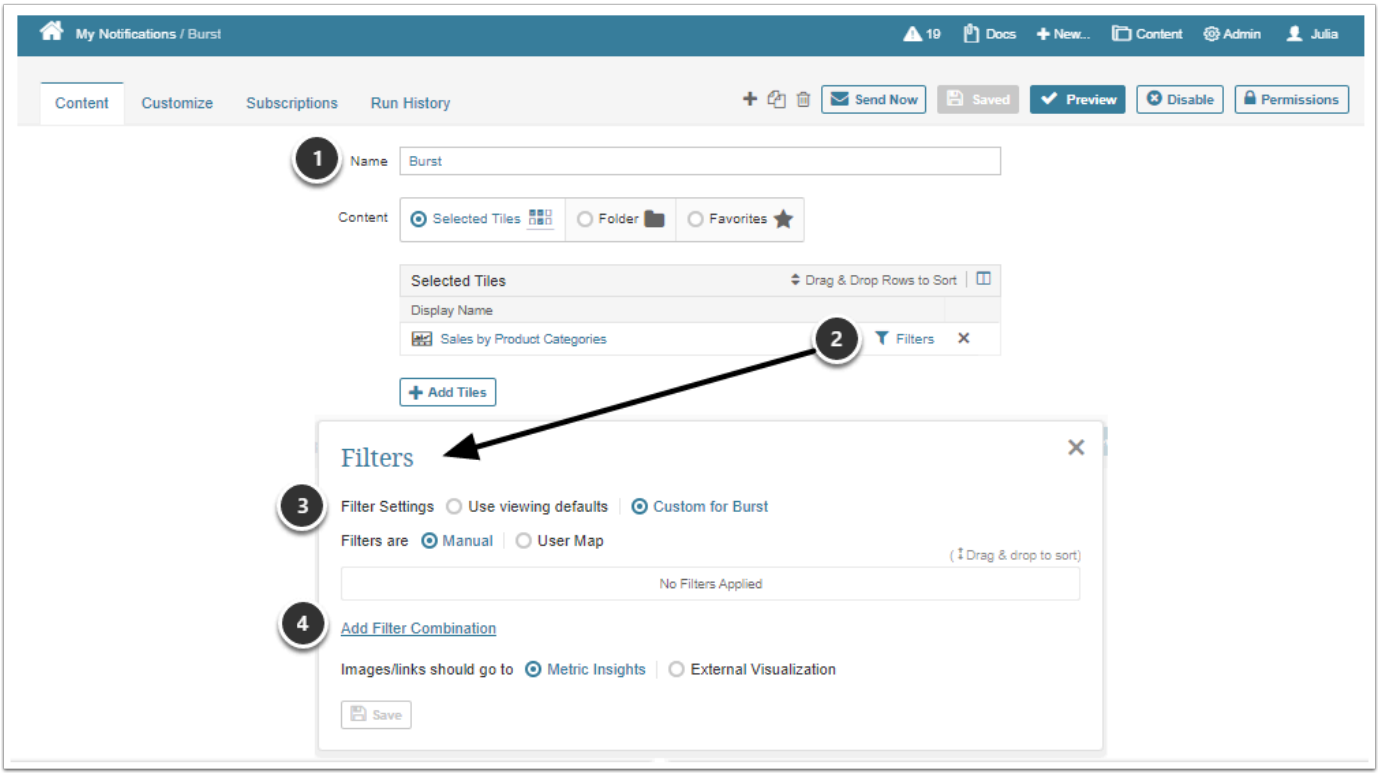
To burst External Report and send one image with "All Values" and one image of the desired External Report filter value:



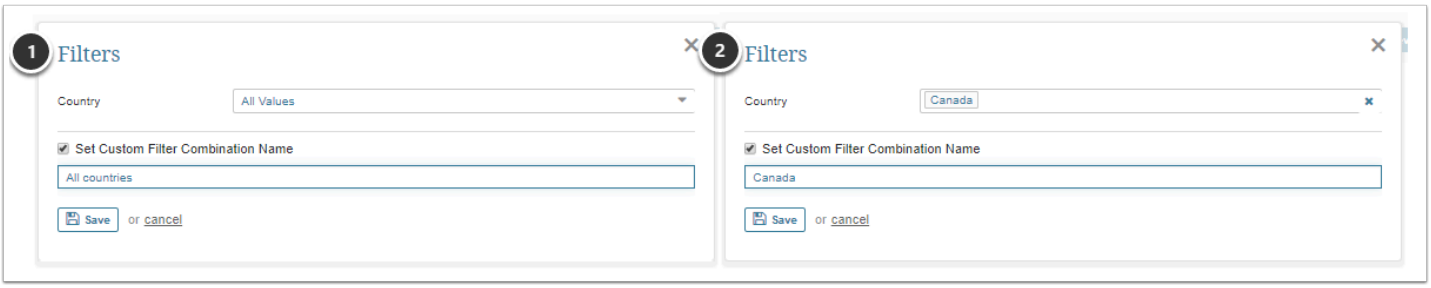
1. Go to [+New] to start creating a burst
2. Select [Burst]
3. [+Add Tiles] to select required tiles



1. Find desired tile through search
2. Check the tile
3. [+Add Selected]



1. Name the burst
2. Go to [Filters]
3. Set "Custom for Burst"
4. [Add Filter Combination]



In [Add Filter Combination], add filters one by one.

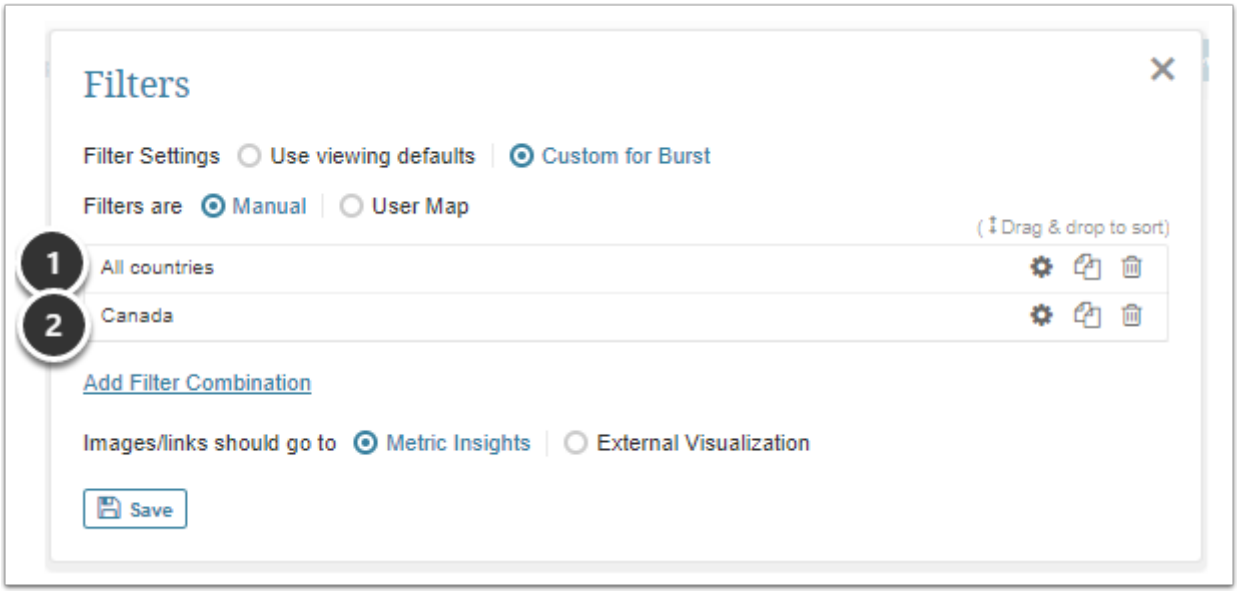
In the above example:

Filter 1:

1. Set "All Values"
2. "Set Custom Filter Combination Name" (optional)
3. [Save]

Filter 2:

1. Select values from the drop-down lists
2. "Set Custom Filter Combination Name" (optional)
3. [Save]



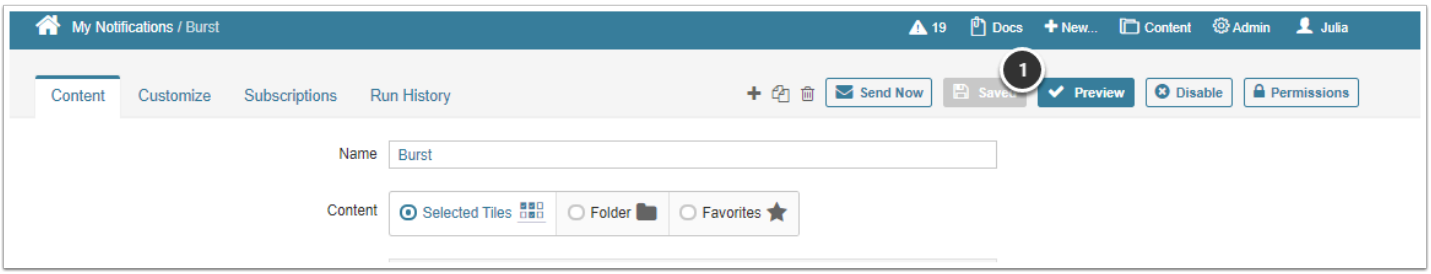
As a result, filters applied comprise:

1. All Values
2. Values by country (Canada)

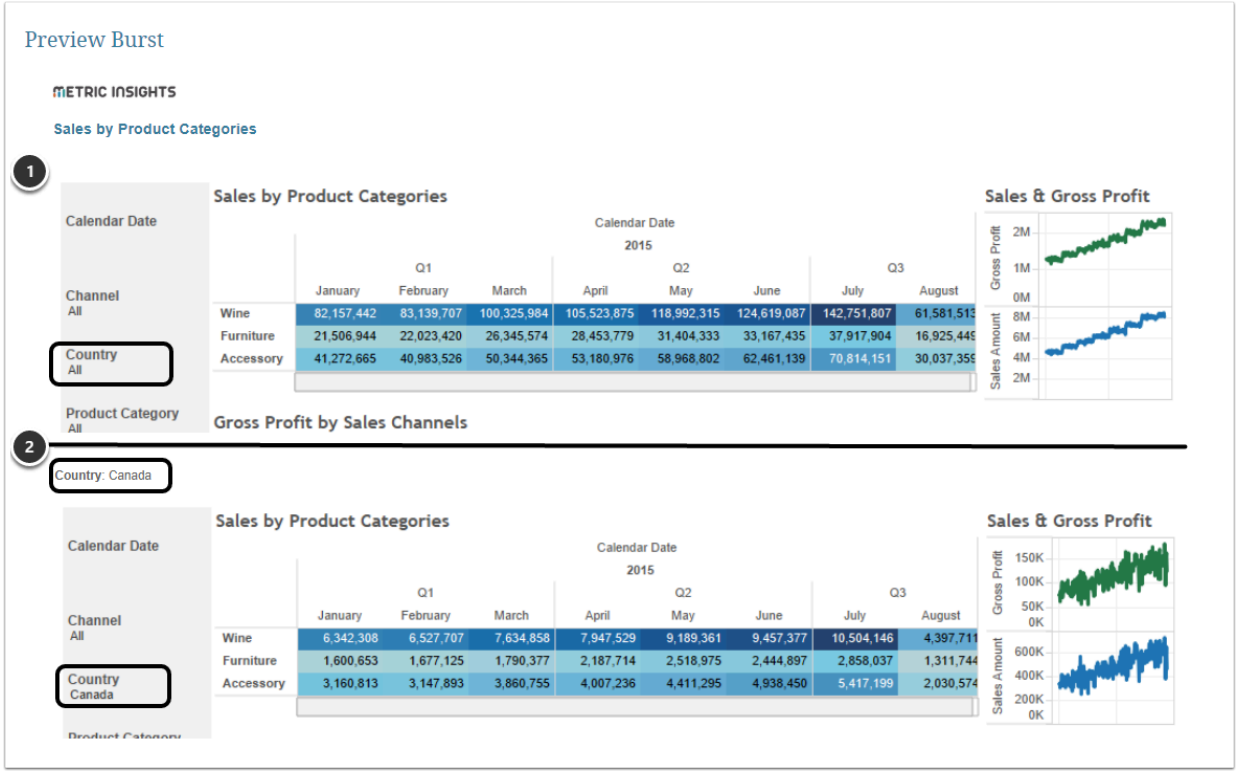
[Save] to save the filters.

2.2. Preview Created Burst

To see the burst created:



1. Go to [Preview]



Preview shows External Report containing:

1. Image with All Values
2. Image with the desired filter value

Now, the burst is ready to be sent.

To learn more about how to create and send a new burst, see [Create New Burst.]

4. Configuring Remote Data Collectors

4.1 Configure a Remote Data Collector/ Processor

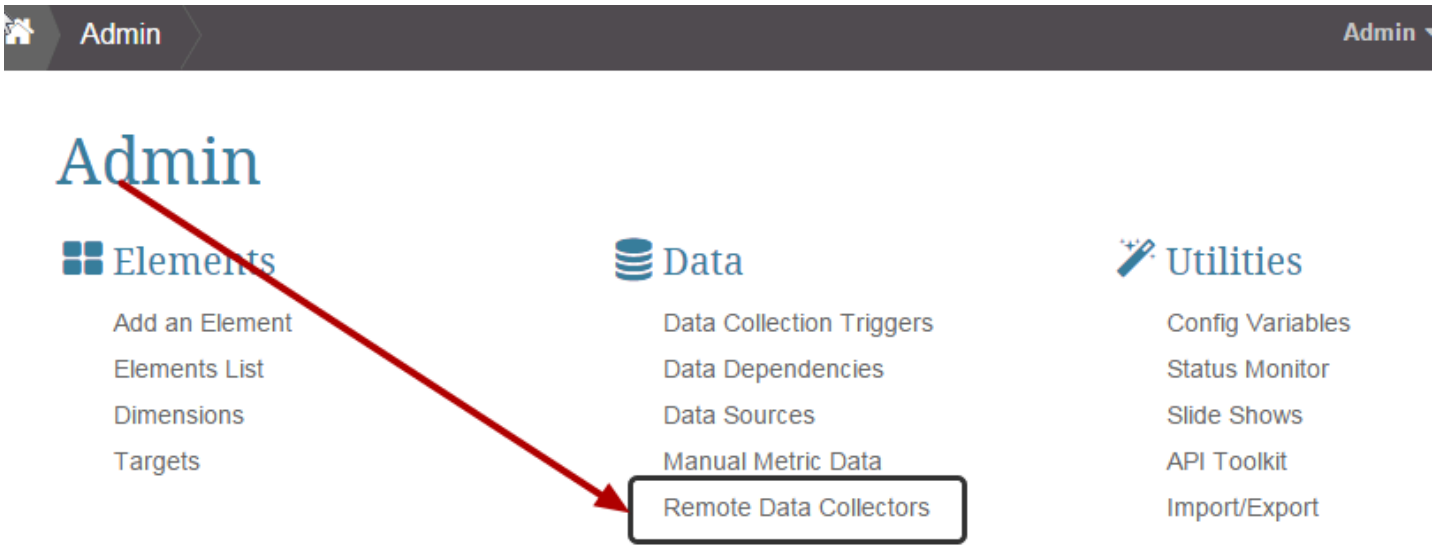
In order to safely connect a Metric Insights instance that is hosted on the cloud to data sources that live behind your firewall, you will need to download *Insightd/Data Processor*, a thin, remote data collection framework that can be used to run queries on behalf of Metric Insights.

Insightd/Data Processor service will run on a machine behind your firewall and will periodically poll Metric Insights for the queries it needs to run. Once these queries are assigned and completed, the results are then securely posted back to the Metric Insights server to get turned into your visualizations.

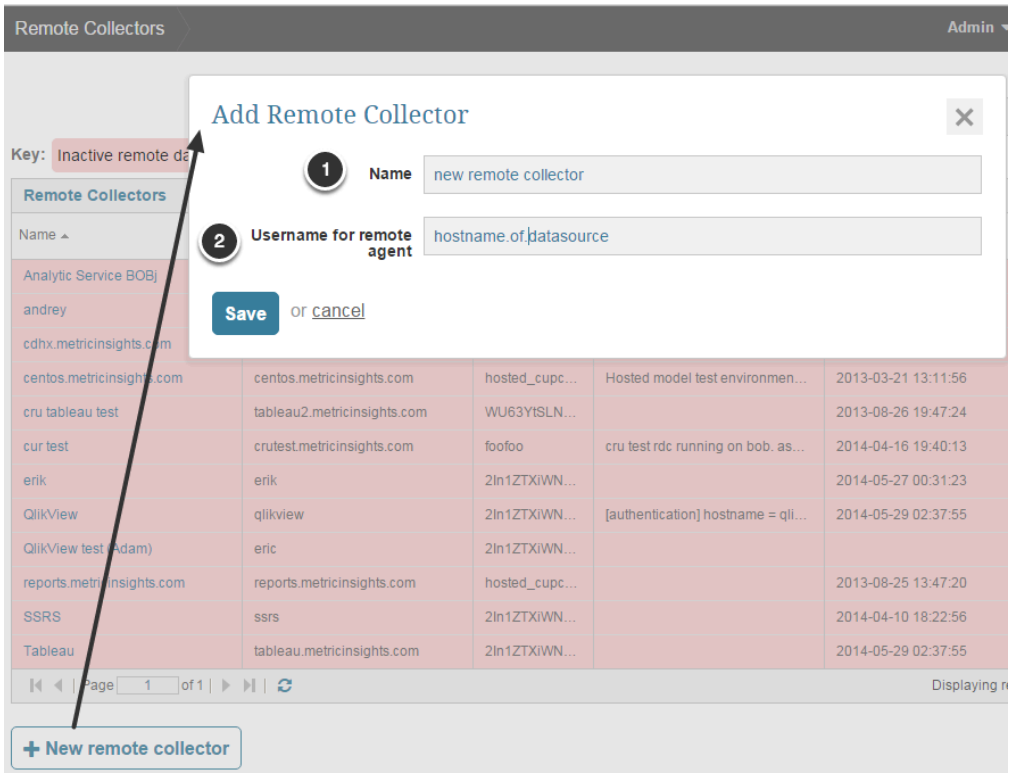
If you encounter any problems you can troubleshoot your remote data collector setup.

[Release 6.x]: 'Remote Data Collector' renamed to 'Remote Data Processor'

1. Select 'Remote Data Collectors' from Admin menu



2. Add a new remote data collector

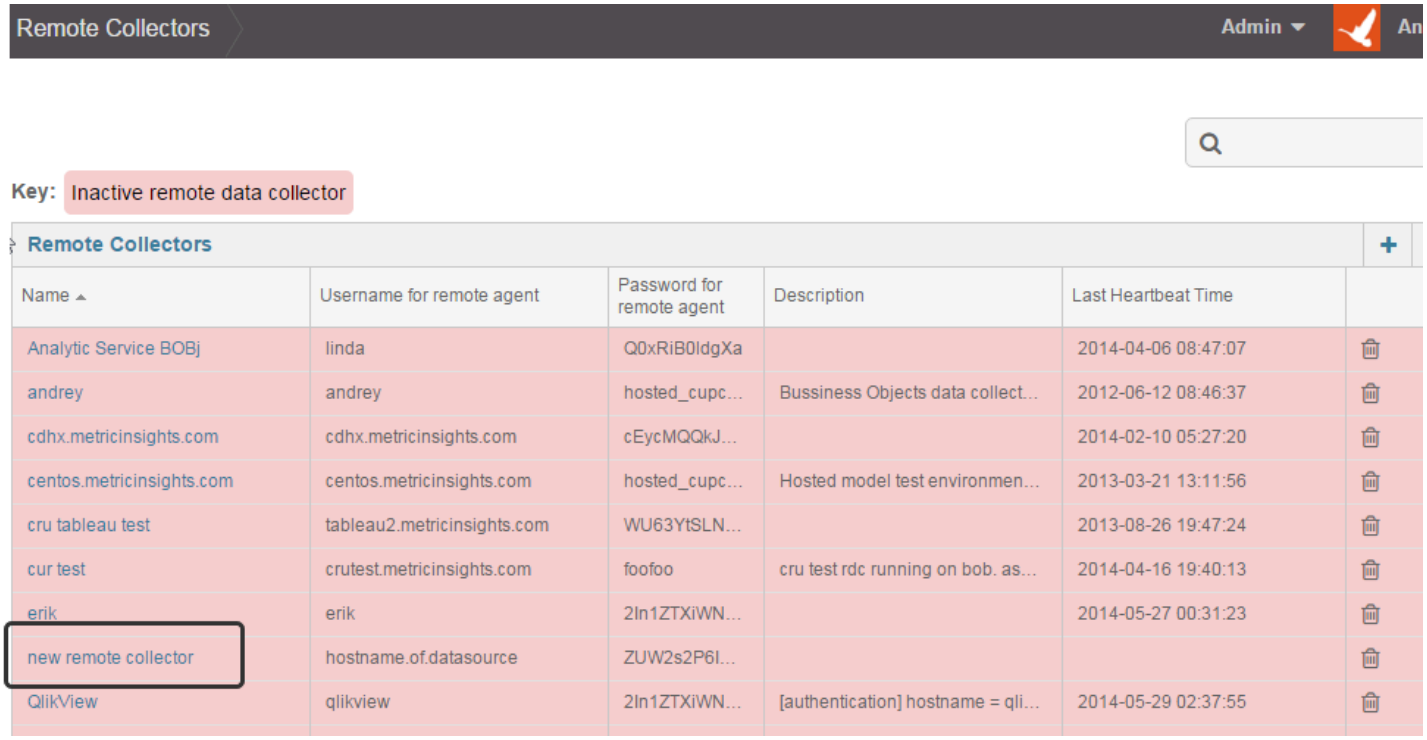


1. Give your new remote data collector a **name**
2. Enter the **hostname** of your remote data source

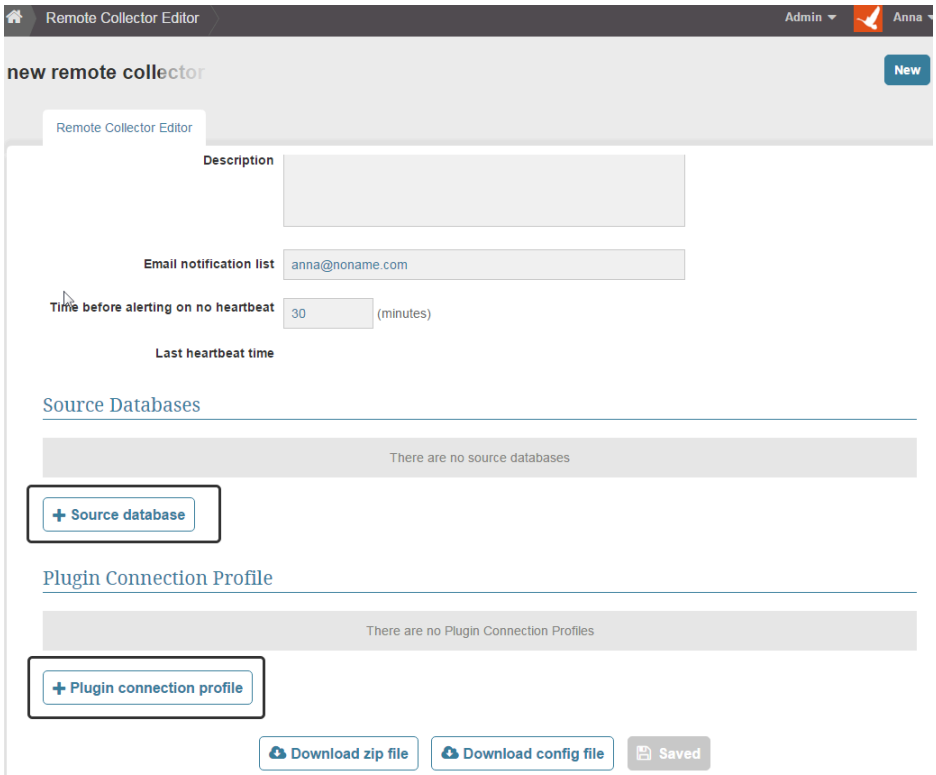
Save

> [Release 6.x]: For remote data processor fill the **hostname** parameter with private IP address of the server you are going to install RDP on.

3. Click the name of the new remote data collector to edit it

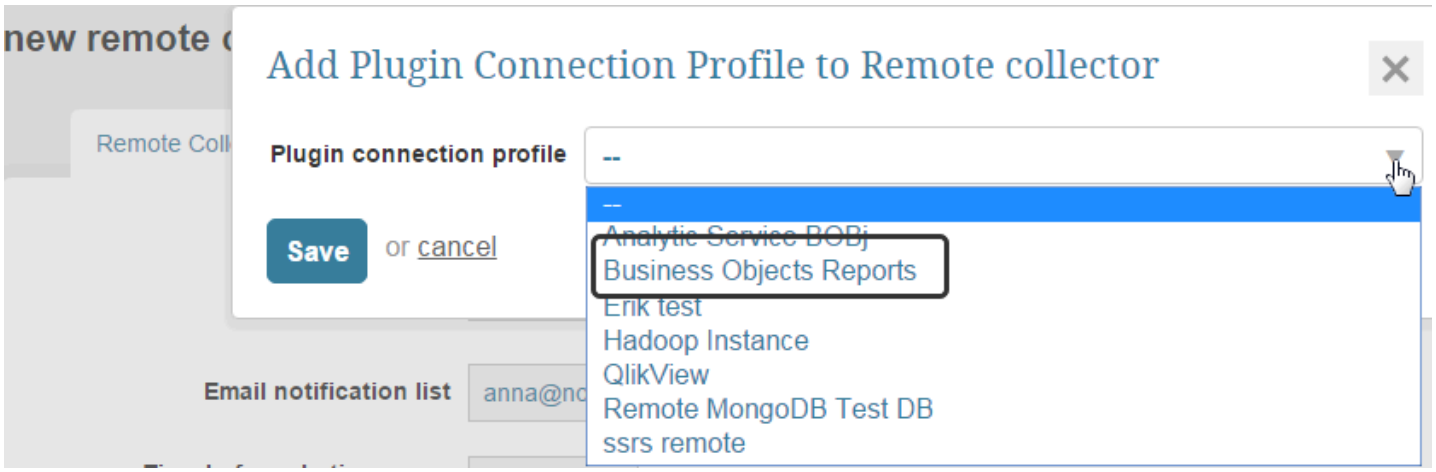


3.1. Add a SQL database or Plug-in data source to the remote data collector



Note: A single Remote Data Collector can connect to multiple data sources if they are all accessible through the same remote host.

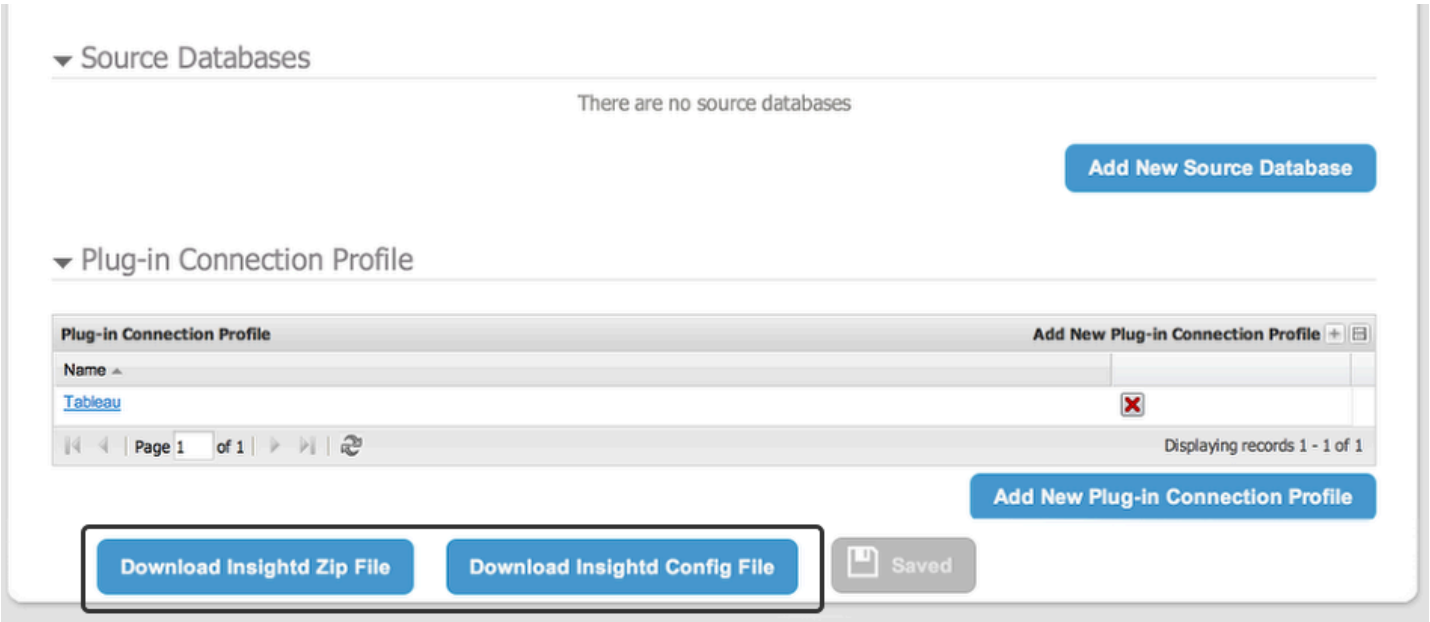
3.2. Select your choice from the associated drop-down



Note: We will connect to **Business Objects** for demonstration purposes.

4. Finally, download the Insightd installer on to the machine your server

The **insightd** installer must be run on a server behind your firewall. Metric Insights allows you to download a pre-configured package of **Insightd** to install on your machine.



5. Install insightd

Now, copy the downloaded file to the server that will be running **Insightd**.

Please refer to the following sections to install **insightd** on Windows and Linux servers.

If you encounter any problems you can troubleshoot your remote data collector setup.

4.2 Installing Insightd on Windows Servers

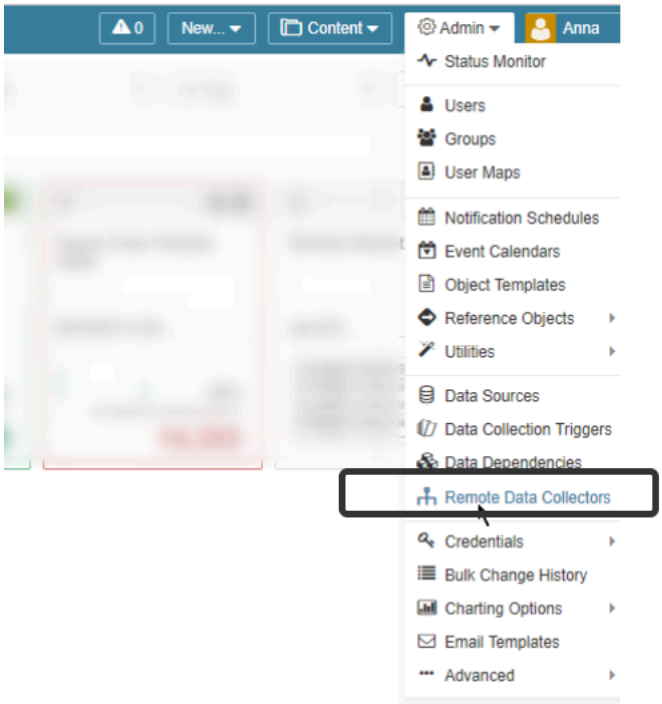
Requirements: The Remote Data Collector requires Java to run on Windows Server. If you need to install Java on Windows, please get the latest runtime package from the Java Download site.

This article assumes you have configured a Remote Data Collector profile, as described in Configuring a Remote Data Collector

If you encounter any problems you can troubleshoot your remote data collector setup.

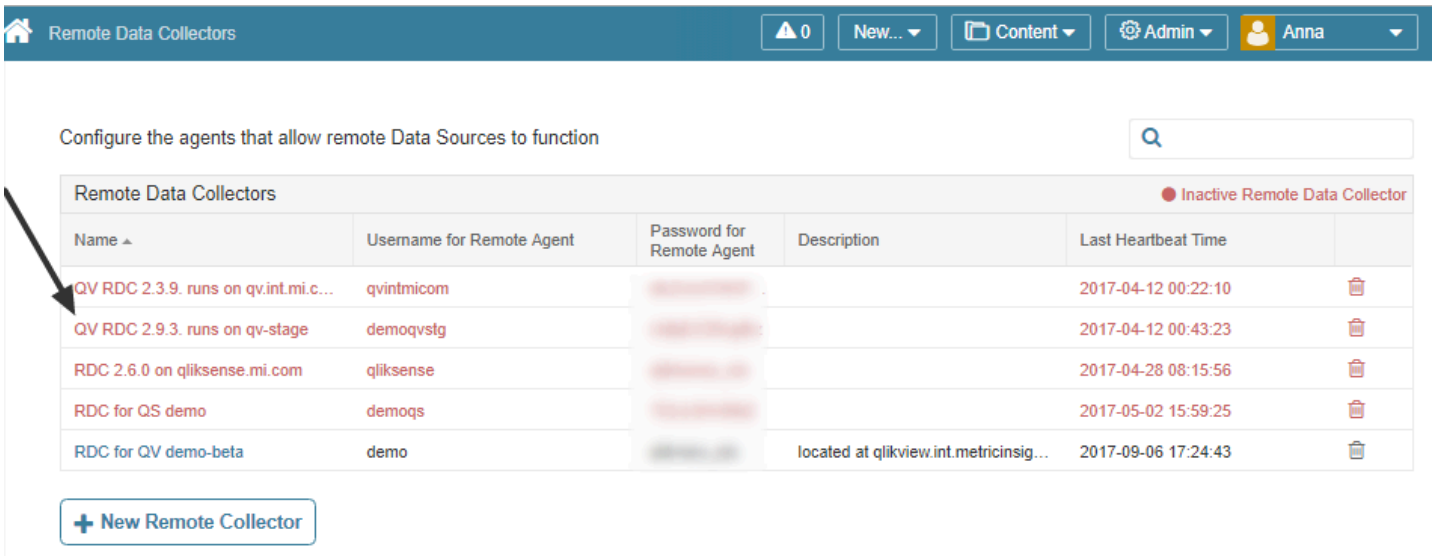
[Release 6.x]: 'Remote Data Collector' renamed to 'Remote Data Processor'

1. Admin > Remote Data Collectors

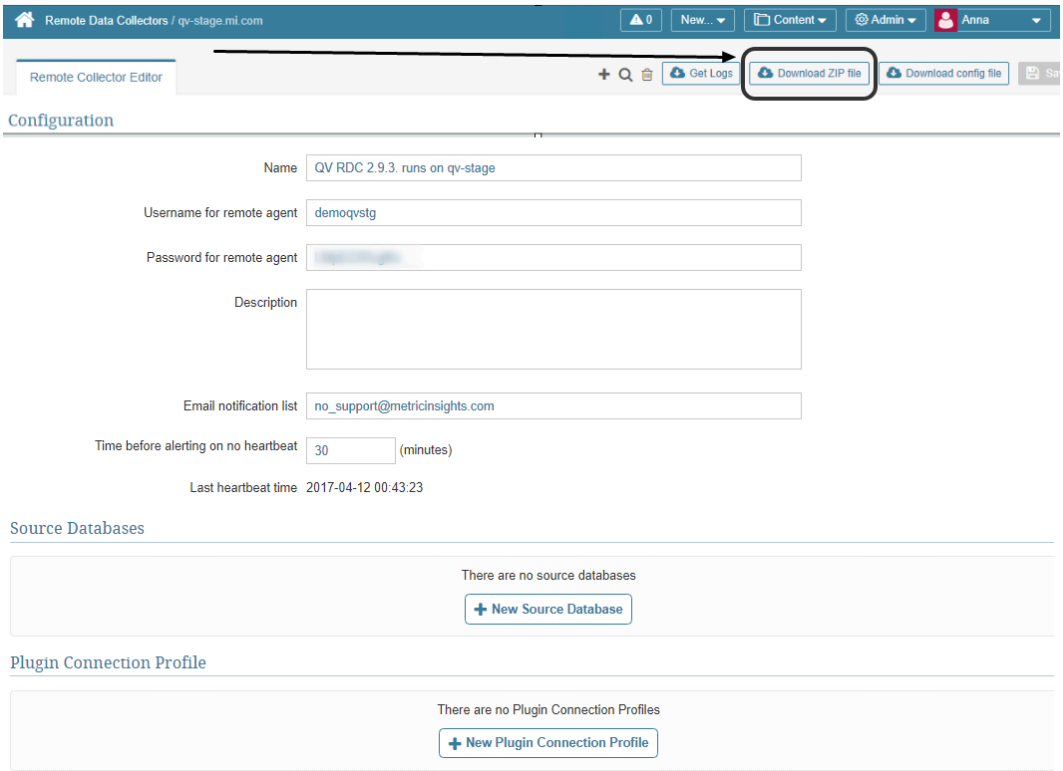


On the Windows Server that you plan to install on, open a web browser and point it to your metric insights server. Log in as admin, and go to Admin -> Remote Data Collectors.

2. Select the data collector that you wish to download

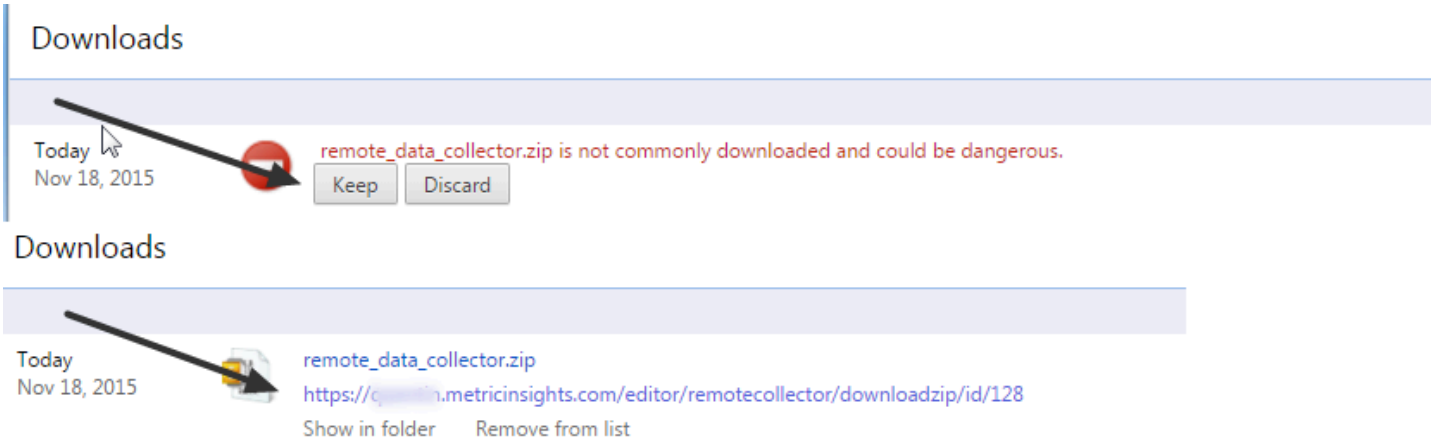


3. Download the Remote Data Collector (Insightd) zip file

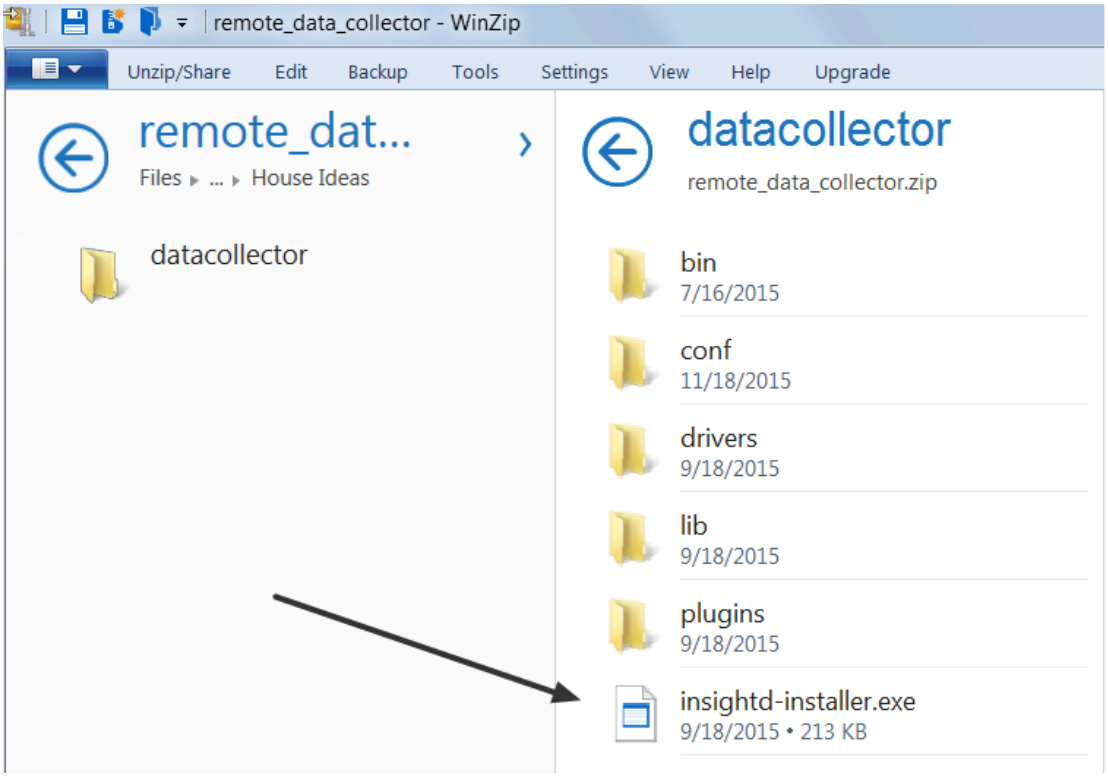


When downloading the zip file, you might get an error message that warns you that the insightd zip isn't normally downloaded. Tell the browser that you're sure you want the zip file. We promise it's not dangerous.

4. Extract the zip file

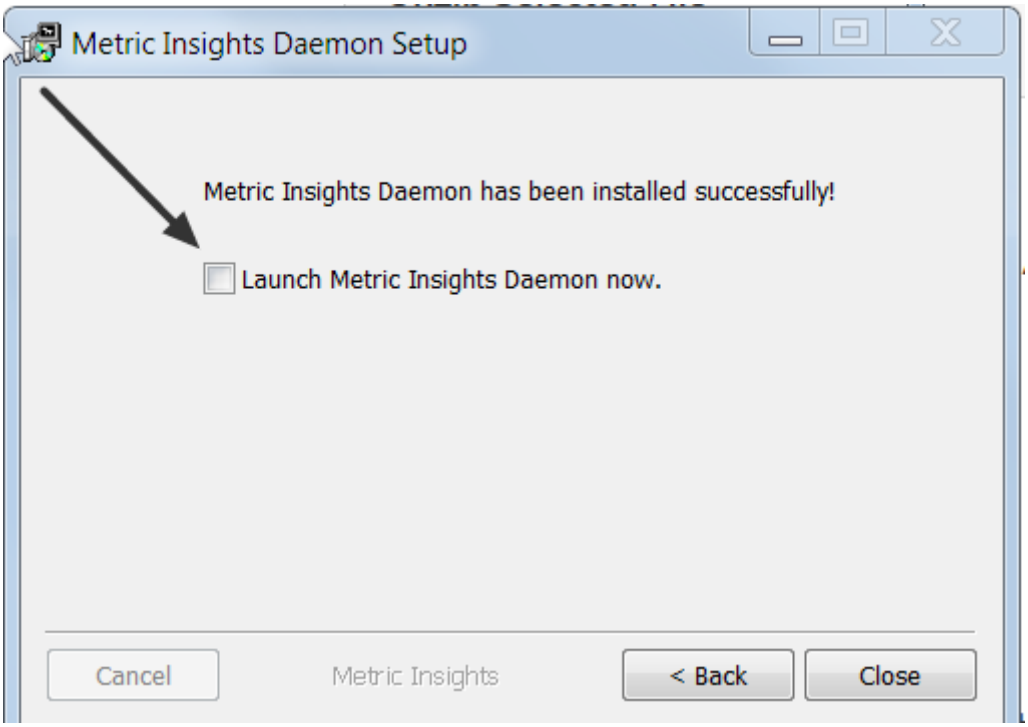


4.1. Unzip and Run insightd-installer.exe



5. Start the Insightd Windows Service

Navigate to the Windows Services list and start the Metric Insights Daemon service



6. Check the Windows Event log for any errors

If you notice that the service hasn't started for some reason, check the Event Viewer for more information. All event logs will be filed under Event Viewer (Local) > Applications and Services Logs > Metric Insights Daemon.

You can also reference more on how to troubleshoot your remote data collector setup.

4.3 Upgrading Insightd on Windows Servers

This article walks you through the steps to upgrade your version of Insightd on Windows Servers. Insightd is the Windows service that runs the Remote Data Collector for your Metrics Insights application. You can review more about what this is in the Overview of Remote Data Collection.

http://help.metricinsights.com/m/Connecting_to_Data_Sources/l/107311-overview-of-remote-data-collection

If you encounter any problems you can troubleshoot your remote data collector setup.

[Release 6.x]: 'Remote Data Collector' renamed to 'Remote Data Processor'

1. Get new version

To get a new version, follow the steps on how to download the new version in the article for Installing Insightd on Windows Servers.

Start with Step 1 and continue to Step 3, the steps that explain how to download the Insightd installer.

http://help.metricinsights.com/m/Connecting_to_Data_Sources/l/107314-installing-insightd-on-windows-servers

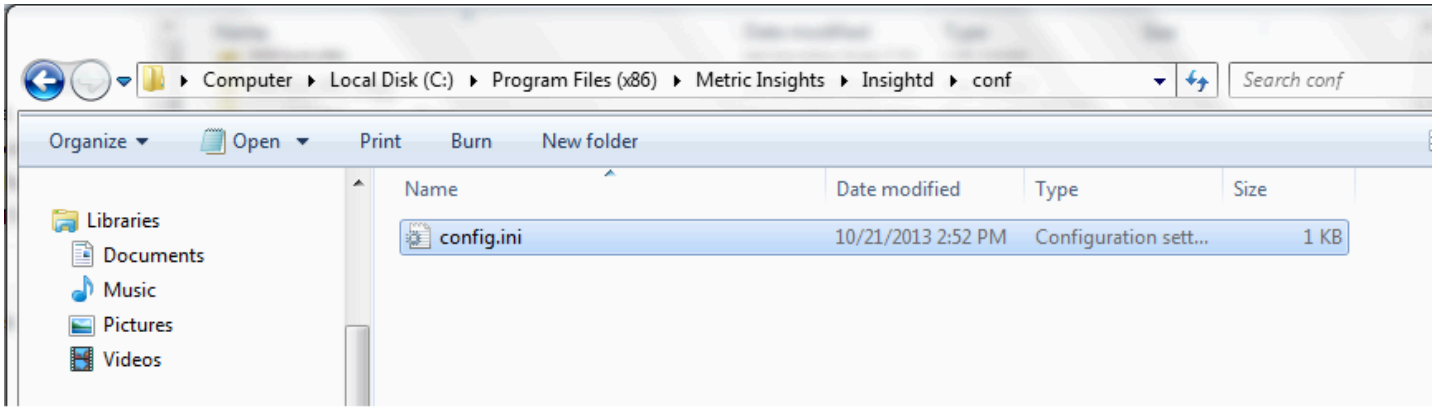
2. Uninstall old version

After getting the new version of the Remote Data Collector in the previous step, the next steps are to uninstall the old version and then to install the new version.

The next several steps walk you through how to uninstall the old version of the Remote Data Collector.

2.1. Backup config.ini file

Before uninstalling the old version of the Remote Data Collector, make a backup copy of your config.ini file (located at "Metric Insights\Insightd\config\config.ini") and save it somewhere such as your home directory.

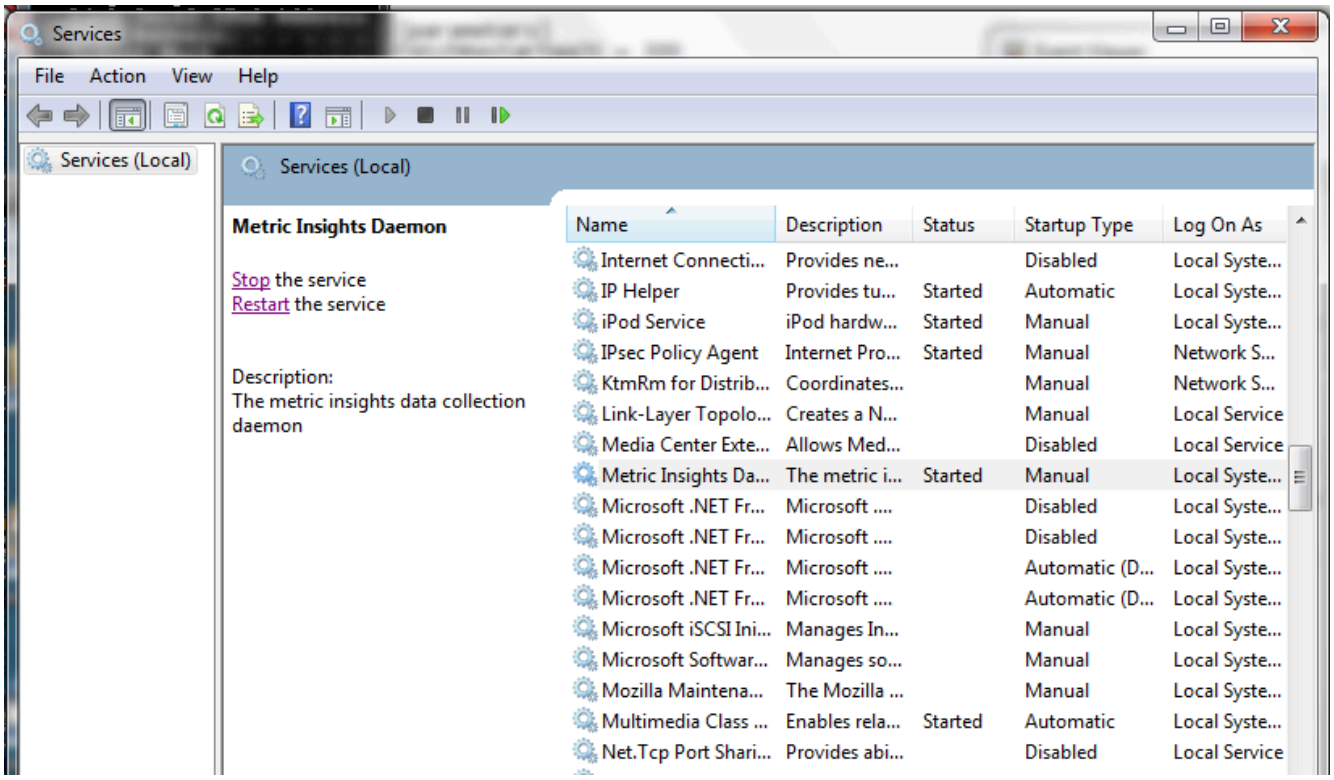


2.2. Stop Windows Service

Now stop the Remote Data Collector if it is running. Find the Metric Insights Daemon in the Windows Services control panel. If it is running then the panel will show you an option to Stop the service and to Restart the service. If the service is not running, then it will only show you an option to Start the service.

If the service is running, then Stop the windows Insightd service.

- Click Stop the service.

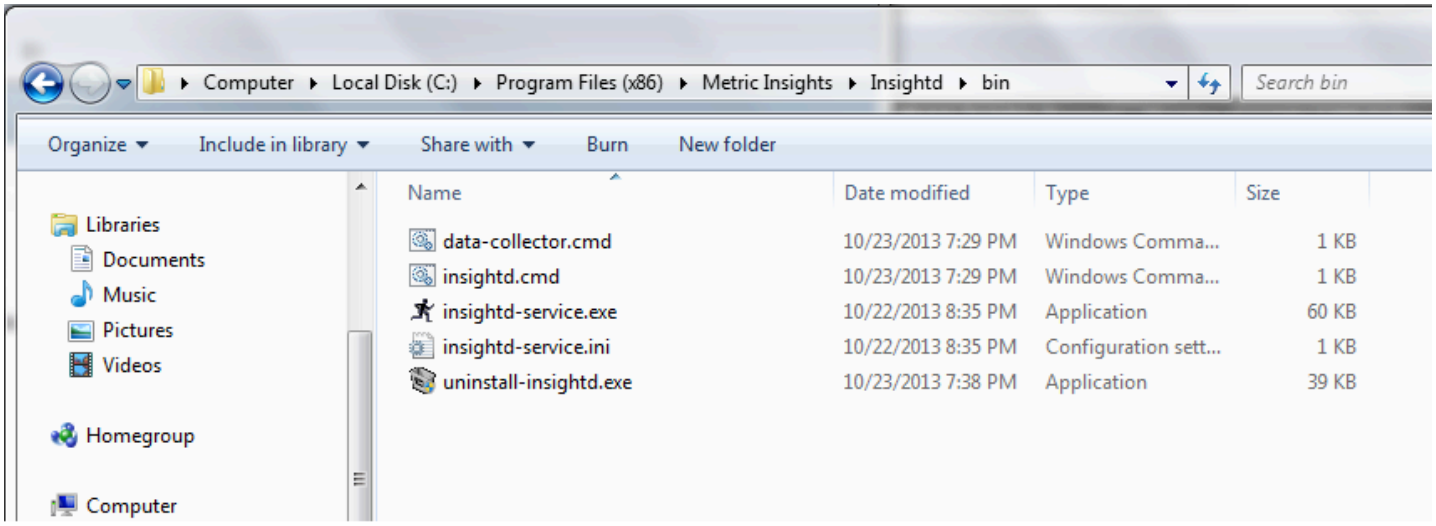


2.3. Uninstall Windows Service

Once the Insightd Windows Service is stopped, then it is safe to uninstall the service.

Uninstall the Insightd service.

- Run the uninstall-insightd.exe file located at "Metric Insights\Insightd\bin\uninstall-insightd.exe". You can double click on it to run it.



3. Install new version

Finally, after completing the above steps of getting the new version of the Remote Data Collector, and then uninstalling the old version, the next step is to install the new version.

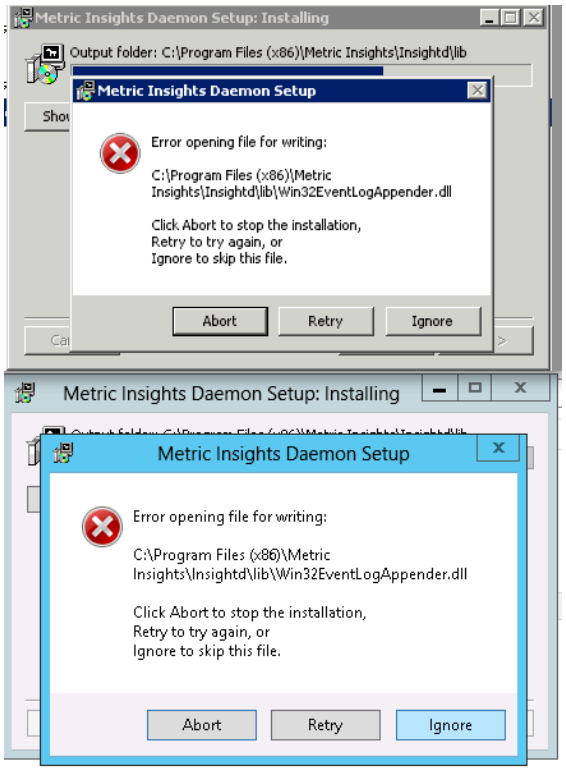
The next several steps walk you through how to install the new version of the Remote Data Collector.

3.1. Install the Windows Service

To install the Insightd Windows Service, follow the steps in the article Installing Insightd on Windows Servers that explains how to extract the installer from the downloaded file and then how to install the service. Only go as far as running the installer. Do not start the Insightd Windows Service yet. Follow Step 4 through Step 5 in the article Installing Insightd on Windows Servers.

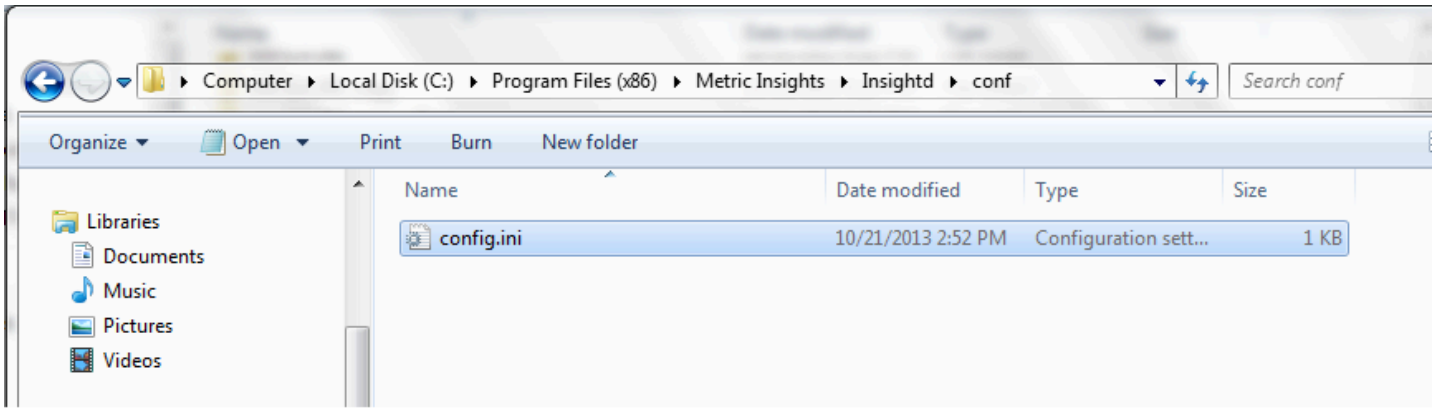
http://help.metricinsights.com/m/Connecting_to_Data_Sources/l/107314-installing-insightd-on-windows-servers

If you get "Error opening file for writing: ...Win32EventLogAppender.dll" just click Ignore and continue.



3.2. Restore config.ini

Copy the config.ini file you saved in a prior step to the "Metric Insights\Insightd\conf\" directory.



3.3. Start the Windows Service

Finally, start the Insightd Windows Service. Continue at Step 6 in the article for Installing Insightd on Windows Servers.

http://help.metricinsights.com/m/Connecting_to_Data_Sources/l/107314-installing-insightd-on-windows-servers

If you encounter any problems you can troubleshoot your remote data collector setup.

4.4 Installing Insightd on Linux Servers

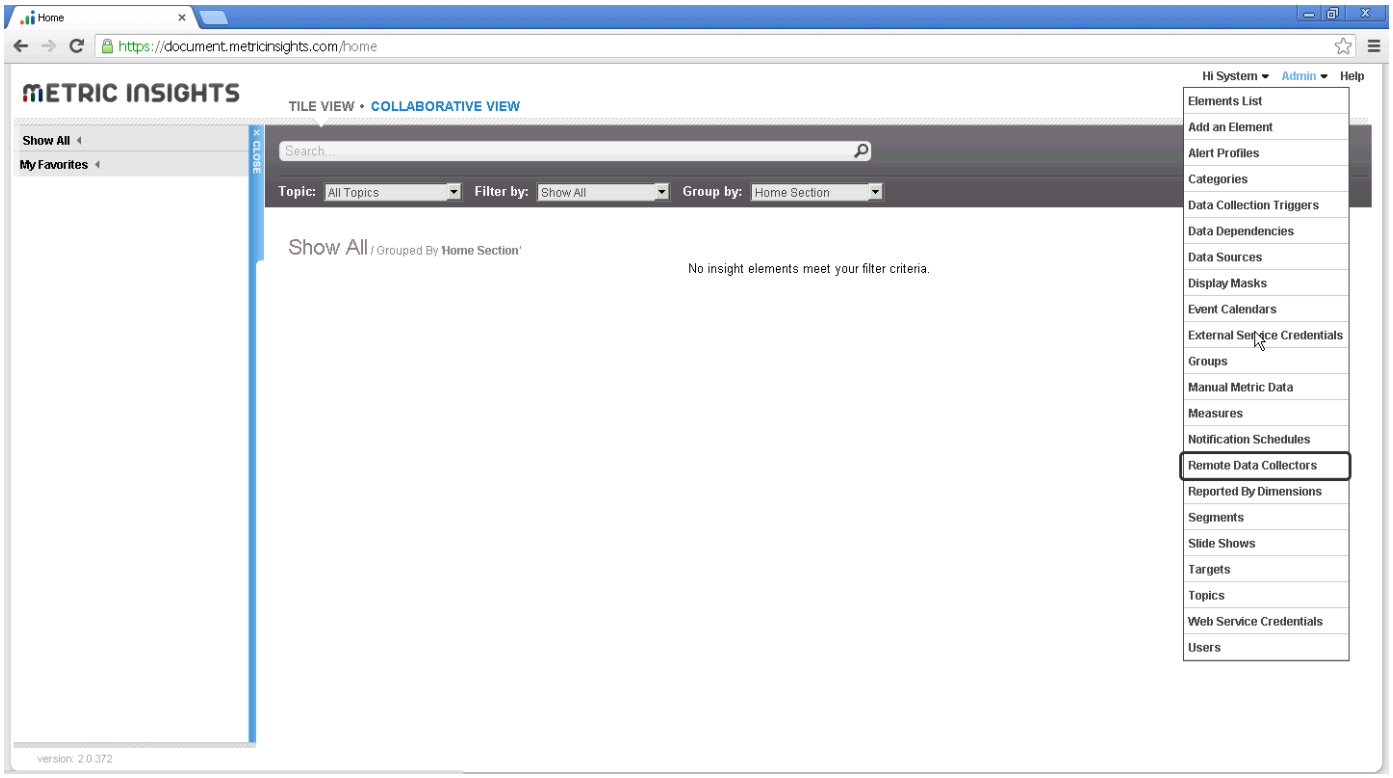
This article assumes you have configured a remote data collector install, as described in Configuring a Remote Data Collector

If you encounter any problems you can troubleshoot your remote data collector setup.

[Release 6.x]: 'Remote Data Collector' renamed to 'Remote Data Processor'

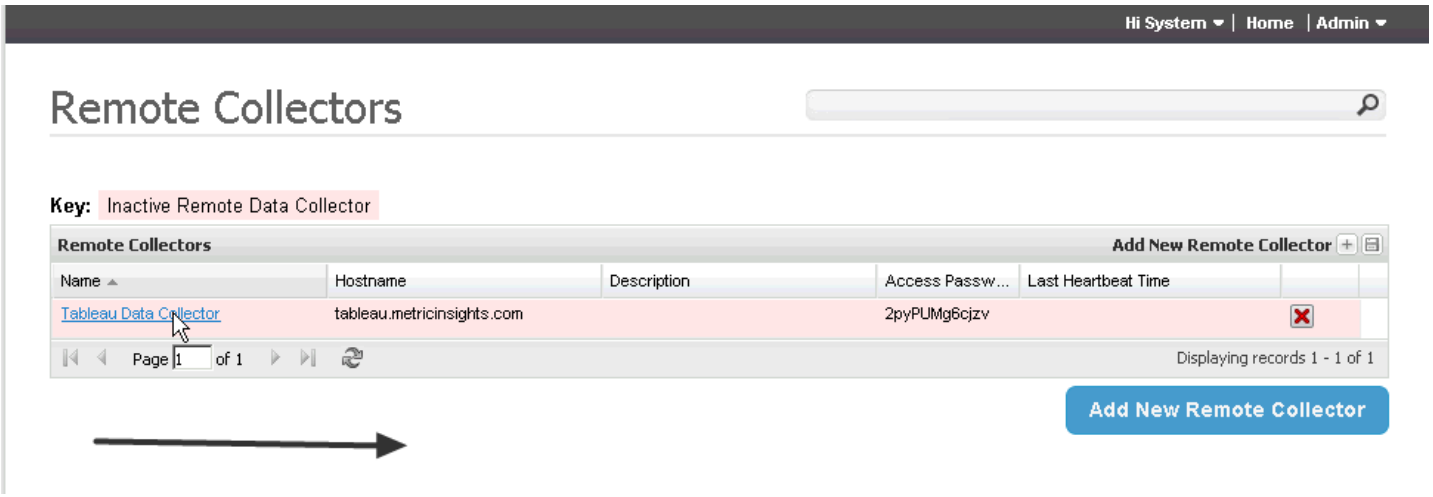
1. Download the Insightd installer from Metric Insights

On the Linux Server that you plan to install on, open a web browser and point it to your metric insights server. Log in as admin, and go to Admin -> Remote Data Collectors.



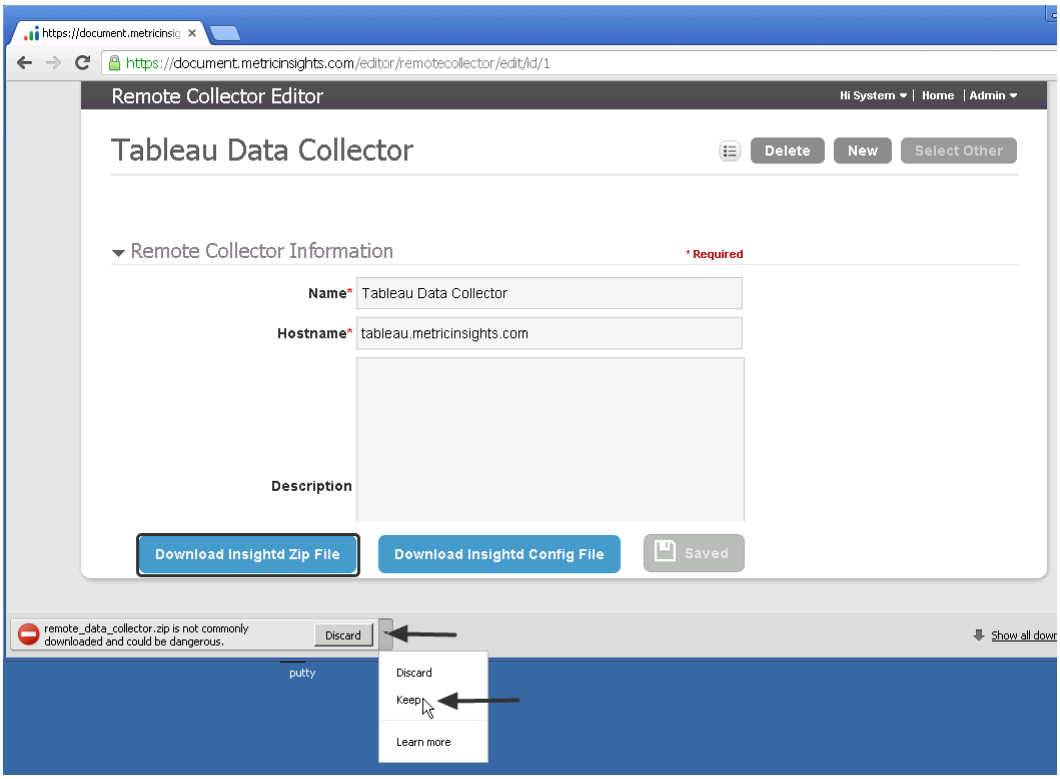
2. Select the data collector that you wish to download

Select the remote data collector you wish to install.



3. Download the Insightd zip file

When downloading the zip file, you might get an error message like the following that warns you that the insightd zip isn't normally downloaded. Tell the browser that you're sure you want the zip file. We promise it's not dangerous.



4. Extract the zip file and start Insightd

root@document:~# unzip remote_data_collector.zip

root@document:~# cd datacollector/bin

```
root@document:~# ./insightd start &
```

If you encounter any problems you can troubleshoot your remote data collector setup.

4.5 Upgrading Insightd on Linux Servers

This article walks you through the steps to upgrade your version of Insightd on Linux Servers. Insightd is the Linux daemon that runs the Remote Data Collector for your Metrics Insights application. You can review more about what this is in the Overview of Remote Data Collection.

http://help.metricinsights.com/m/Connecting_to_Data_Sources/l/107311-overview-of-remote-data-collection

If you encounter any problems you can troubleshoot your remote data collector setup.

[Release 6.x]: 'Remote Data Collector' renamed to 'Remote Data Processor'

1. Uninstall old version

The following steps walk you through how to uninstall the old version of the Remote Data Collector.

1.1. Stop Insightd

Stop the Remote Data Collector if it is running. Find the location where the remote data collector is running and stop it.

Stop the current remote data collector:

```
./datacollector/bin/insightd stop
```

If it's not running then you might get the following message:

```
"The insightd can not be stopped because it is not currently running."
```

1.2. Save Insightd (save current configuration)

Move files to a saved location. You can give the name of the new directory anything you want.

```
mv ./datacollector/ ./datacollector-todays-date
```

2. Get new version

To get a new version, follow the steps on how to download the new version in the article for Installing Insightd on Linux Servers.

Start with Step 1 and continue to Step 3, the steps that explain how to download the Insightd installer. At this point do not extract the zip file. Do not start up Insightd.

http://help.metricinsights.com/m/Connecting_to_Data_Sources/l/107315-installing-insightd-on-linux-servers

In rare cases you might receive the remote data collector via a different means. For example, from Metric Insights company directly. For example you might get a patched version or one with a new feature. In that case just follow the directions given for that situation.

3. Install new version

The next several steps walk you through how to install the new version of the Remote Data Collector.

3.1. Install Insightd

Install Insightd

```
unzip remote_data_collector.zip
```

This will install the remote data collector (i.e., extract the contents of the zip file) into the ./datacollector directory. If you obtained this file via a different means then it might have a different name.

3.2. Check if new configuration is correct

Check if new configuration is correct

```
cat ./datacollector/conf/config.ini
```

If you downloaded this from your Metric Insights system described in the above steps, then this step is not necessary. If you obtained the remote data collector via a different means then this step is recommended.

3.3. Restore Configuration (if new configuration not correct)

If you downloaded this from your Metric Insights system described in the above steps, then this step is not necessary. However, if you obtained it via a different means and you need to restore your previous configuration for some reason then you can do the following

```
mkdir -p ./datacollector/conf
cp -p ./datacollector-todays-date/conf/config.ini ./datacollector/conf/
```

3.4. Start Insightd

Start insightd

```
./datacollector/bin/insightd start &
```

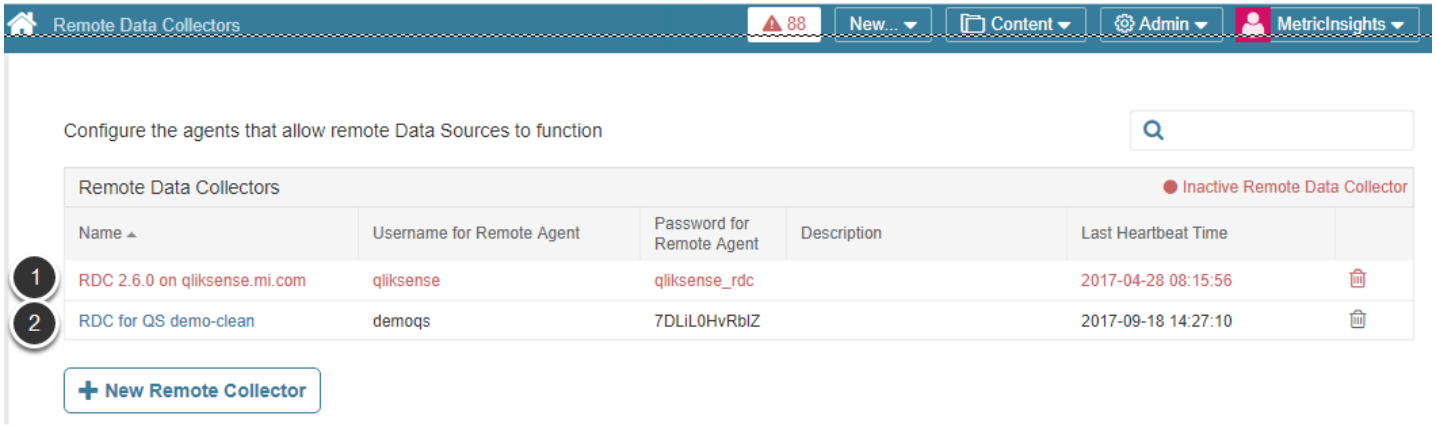
If you encounter any problems you can troubleshoot your remote data collector setup.

4.6 Troubleshooting a Remote Data Collector/Processor

This article provides some tips for troubleshooting problems with your Remote Data Collector

[Release 6.x]: 'Remote Data Collector' renamed to 'Remote Data Processor'

1. Check if Remote Data Collector is active. Admin > Remote Data Collector



The Remote Collectors screen lists all your Remote Data Collectors and provides a visual queue whether the Remote Data Collector is active. It also lists the last time the collector successfully communicated with the system.

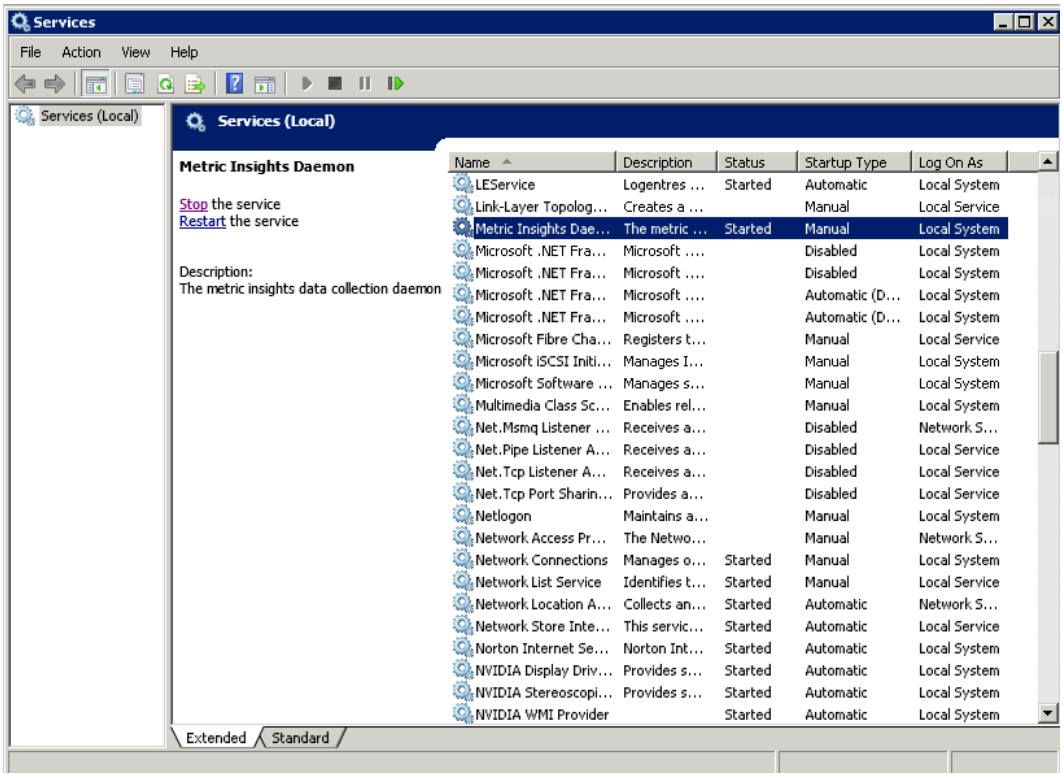
1. If your Remote Data Collector is not active then it will show with pink background, and the column for "Last Heartbeat Time" will have a value with the last time it successfully connected. The image below is for a Remote Data Collector that is not active - the Last Heartbeat Time is old and the background is pink. You will need to troubleshoot more
2. If your Remote Data Collector is active then it will show with white background, and the column for "Last Heartbeat Time" will have a value within a couple minutes of the current system time.

2. Restart the Remote Data Collector

On the machine that the Remote Data Collector is running, stop and restart the Remote Data Collector

2.1. Windows - Restart Remote Data Collector

Find the Services menu on the Windows machine and stop and restart the Remote Data Collector. The service is named 'Metric Insights Daemon'



2.2. Linux - Restart Remote Data Collector

Find the directory where the Remote Data Collector is running and stop and restart the process. See instructions for where you installed the program.

Stop the Remote Data Collector:

```
/var/www/datacollector/bin/insightd stop
```

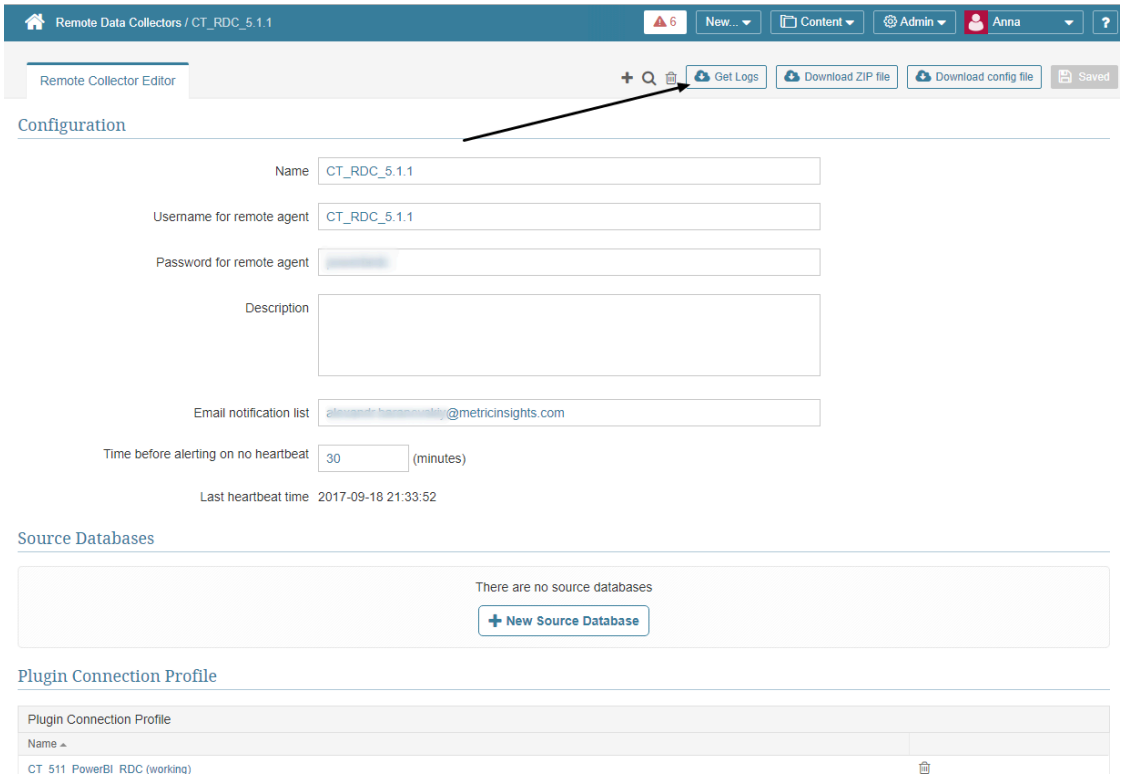
Start the Remote Data Collector:

```
/var/www/datacollector/bin/insightd start
```

3. Check error messages from the remote machine

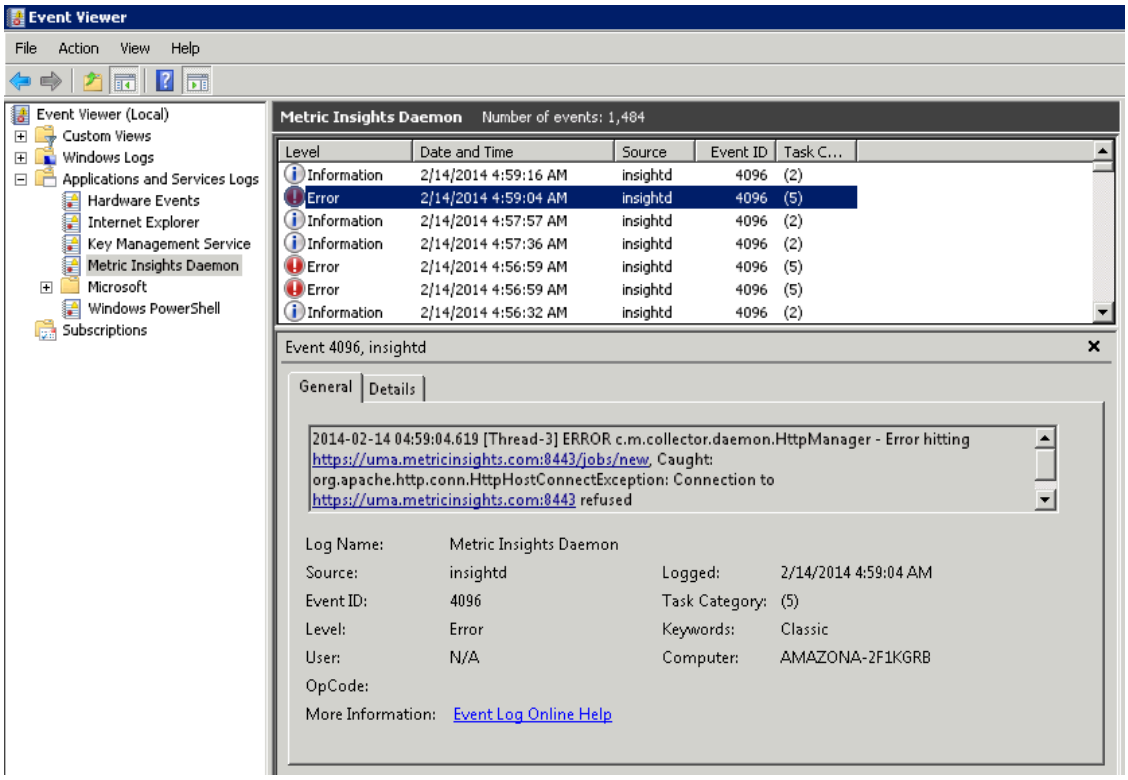
In Version 5.2.0+, you can download log files using the [**Get Logs**] button from the Remote Data Collector Editor page. The downloaded zip includes the primary *insight.log* and plugin specific logs like *qlik.log* and *powerbi.log*.

For prior versions, open the collapsed sections below for steps on now to view error messages on the Remote Data Collector host machine.



3.1. Windows - Check error messages from the Remote Data Collector on the remote machine

Find the Event Viewer menu on the Windows machine and choose the events for the service named 'Metric Insights Daemon'. Errors are flagged in red and provide information on problems that the Remote Data Collector encountered.



3.2. Linux - Check error messages from the Remote Data Collector on the remote machine

View the errors in the log.

```
cat /var/log/insight/insightd-error.log
```

4. Check error messages on Metric Insights server

View the errors in the log.

Debian instance:

```
cat /var/log/apache2/pyinsight-error.log
```

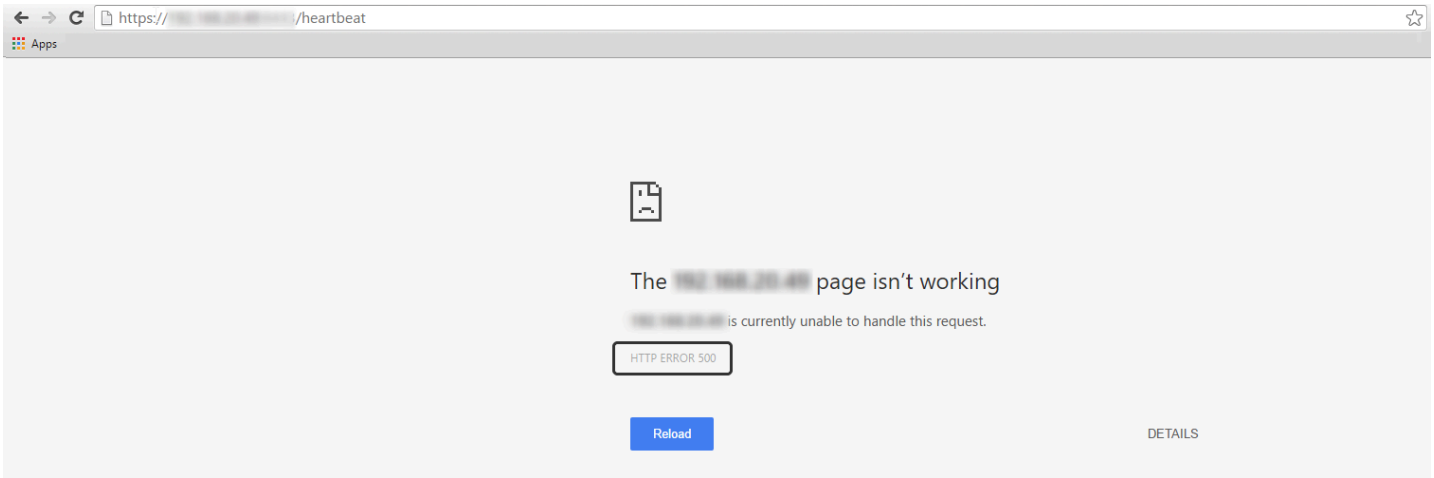
CentOS instance:

```
cat /var/log/httpd/pyinsight-error.log
```

5. Verify the Data Collector "heartbeat" in the browser

Use **https://000.000.00.00:8443/heartbeat** and replace the zeros with the server IP address

5.1. [HTTP ERROR 500] (Internal Server Error)



In case you've encountered **HTTP Error 500** (Internal Server Error): <your IP address> is currently unable to handle this request.

5.2. Troubleshooting [HTTP ERROR 500]

1. Connect to the server where MI is installed by ssh:

- Use ssh root@000.000.000.000 and replace the zeros with the server IP address
- When prompted, enter the root password provided by your IT specialist

2. Open the file with settings of the Data Collection Service:

```
cat opt/mi/pyinsight_app/settings.py
```

3. Find the **DEBUG** setting (see the image below). If 'DEBUG = False', open the file in editor and change it to 'DEBUG = True' and save changes made.

```
root@debian7:/home/user# cat /opt/mi/pyinsight_app/settings.py
# Django settings for metricinsights project.

import os
import sys

MYLOC = os.path.abspath(os.path.dirname(__file__))
MYLIBS = 'libs'

if not MYLIBS in sys.path:
    sys.path.insert(0, os.path.join(MYLOC, MYLIBS))

import aes # a wrapper to import correct AES object

DEBUG = False
TEMPLATE_DEBUG = DEBUG
#DEBUG_PROPAGATE_EXCEPTIONS = True

ADMINS = (
    # ('Your Name', 'your_email@domain.com'),
```

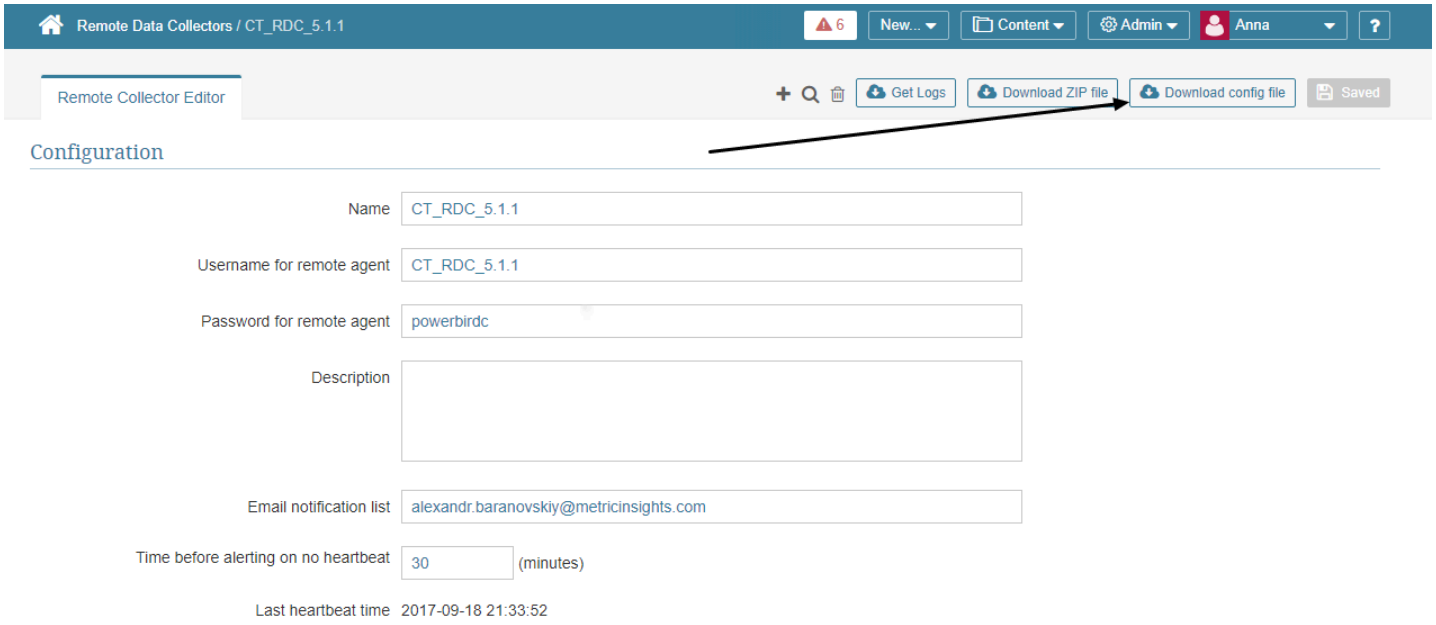
5.3.

4. Restart Metric Insights Daemon at **Services**
5. Verify the **Data Collector "heartbeat"** again

NOTE: Once the problem is solved, the server is going to request your credentials for authentication.

6. Verify correct settings in the Remote Data Collector configuration file against data in UI

If running on Version 5.0 or above, you can download the Configuration file - **Download config file** - from the Remote Data Collector Editor. On older versions, open the collapsed articles below to see steps for viewing the Config file on the machine that the Remote Data Collector is running.



Note the setting values for:

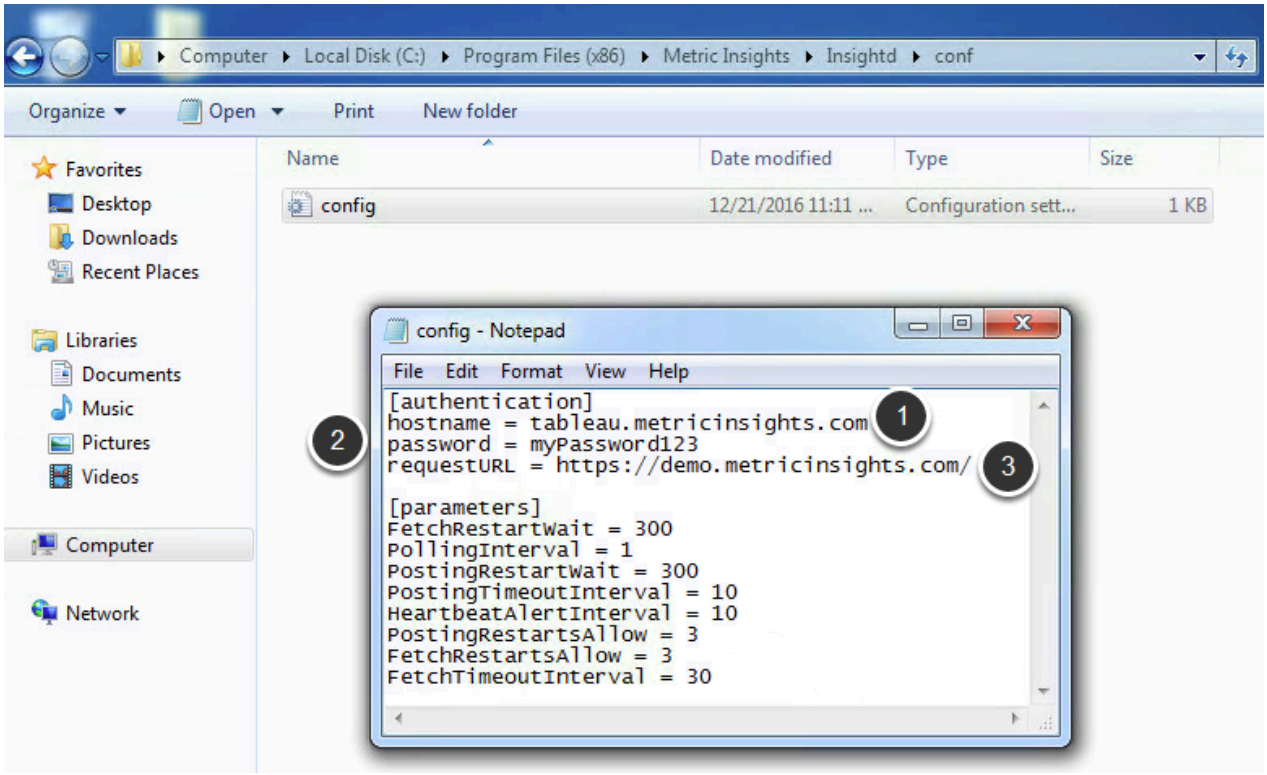
1. **Username for remote agent**
2. **Password for remote agent"**
3. **Domain name** of your Metric Insights instance as seen in the web browser URL

These values will be compared to your config file settings.

6.1. Windows - navigate to configuration file used by Remote Data Collector

```
C:\Program Files (x86)\Metric Insights\Insightd\conf
```

6.2. Windows - Verify settings in the configuration file



Verify that the following fields contain the same values recorded in Step 6.1 from the Remote Editor online

1. Hostname = **Username for remote agent**
2. Password = **Password for remote agent**
3. requestURL = **Domain name** of your MI instance

6.3. Linux - navigate to configuration file used by Remote Data Collector

Find the directory where the Remote Data Collector is running and view the configuration file "config.ini". See instructions for where you installed the program.

```
cat /var/www/datacollector/conf/config.ini
```

6.4. Linux - Verify settings in the configuration file

```
[authentication]
hostname = tableau.metricinsights.com
password = myPassword123
requestURL = https://demo.metricinsights.com
[parameters]
FetchRestartWait = 300
PollingInterval = 1
PostingRestartWait = 300
PostingTimeoutInterval = 10
HeartbeatAlertInterval = 10
PostingRestartsAllow = 3
CacheDNSLookup = 0
FetchRestartsAllow = 3
FetchTimeoutInterval = 30
```

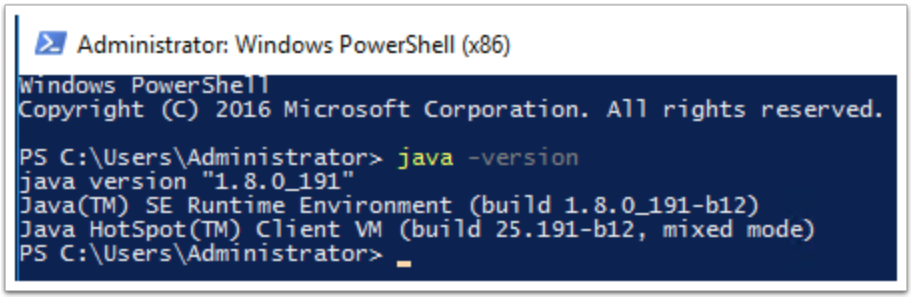
Verify that the following fields contain the same values recorded in Step 6.1 from the Remote Editor online

1. Hostname = **Username for remote agent**
2. Password = **Password for remote agent**
3. requestURL = **Doman name** of your MI instance

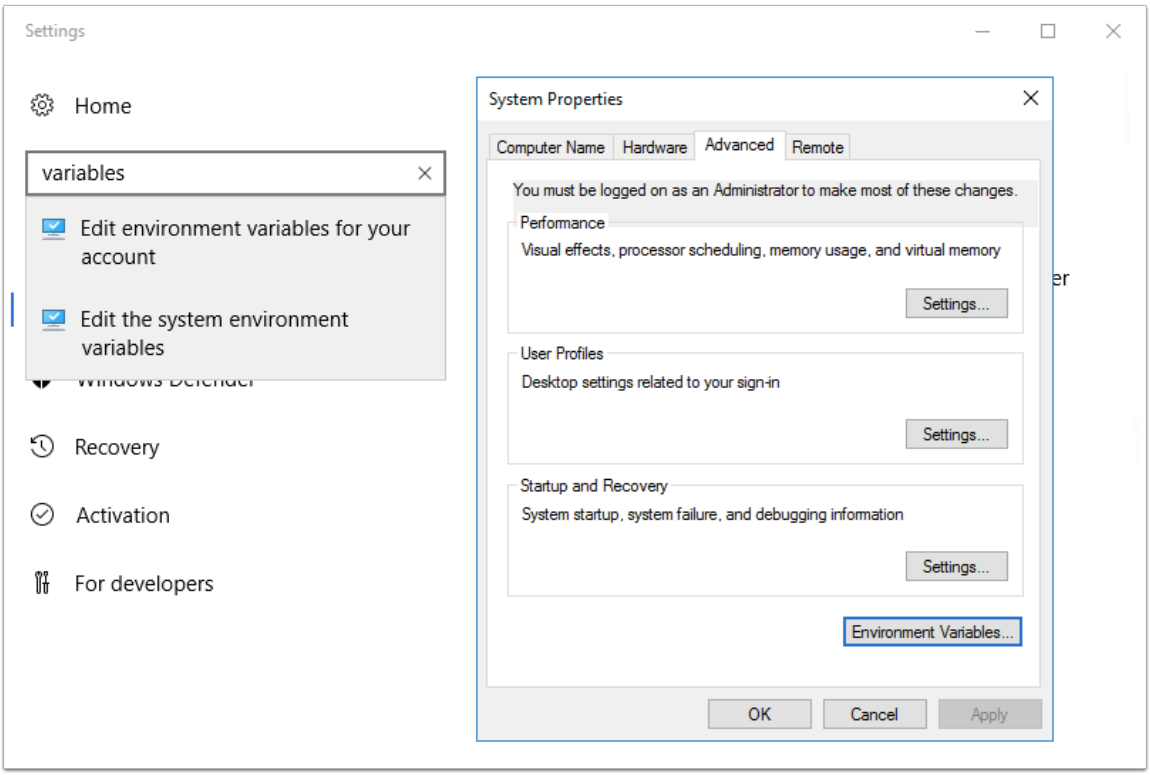
7. Verify java is installed

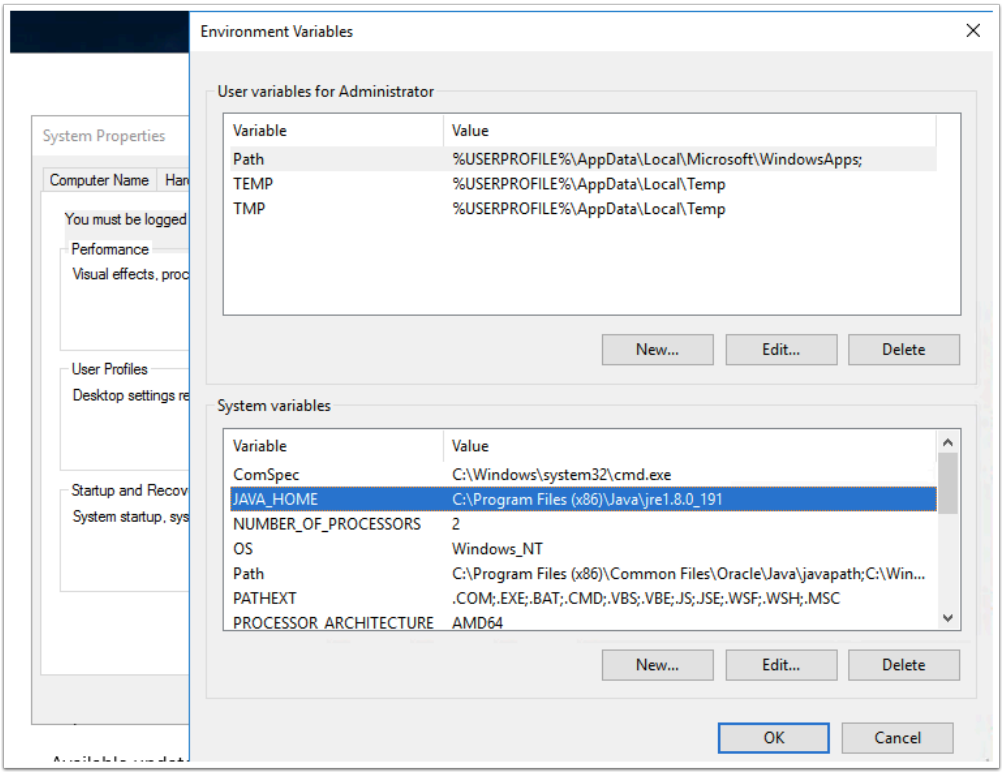
7.1. Windows or Linux - At command prompt verify java is installed

```
java -version
```



7.2. Windows - Verify JAVA_HOME environment variable is set





If JAVA_HOME is not set, then set JAVA_HOME to where the Java software is located, for example, C:\Program Files (x86)\Java\jre1.8.0_191

8. If Windows service could never start up (and exits immediately) then memory size parameter for Metric Insights program might be too large.

8.1. Solution: reduce memory size parameter in Metric Insights program

Using editor (such as Notepad editor) for file

```
C:\Program Files (x86)\Metric Insights\Insightd\bin\insightd.cmd
```

Verify and change program argument -Xmx (e.g., -Xmx2048m or whatever large size it has) to smaller size such as 1GB (e.g., -Xmx1024m)

Using editor (such as Notepad editor) for file:

```
C:\Program Files (x86)\Metric Insights\Insightd\bin\insightd-service.xml
```

Verify and change program argument -Xmx (e.g, -Xmx8g or whatever large size it has) to smaller size such as 1GB (e.g., -Xmx1g)

4.7 Configuring a Remote Data Processor on Windows Servers

Requirements: The Remote Data Processor requires Java to run on Windows Server. If you need to install Java on Windows, please get the latest runtime package from the Java Download site. The Java installation package should correspond to your Windows operating system version (32-bit or 64-bit).

This article assumes you have configured a Remote Data Processor profile, as described in Configure a Remote Data Collector/Processor.

1. Open ports on Windows RDP host machine

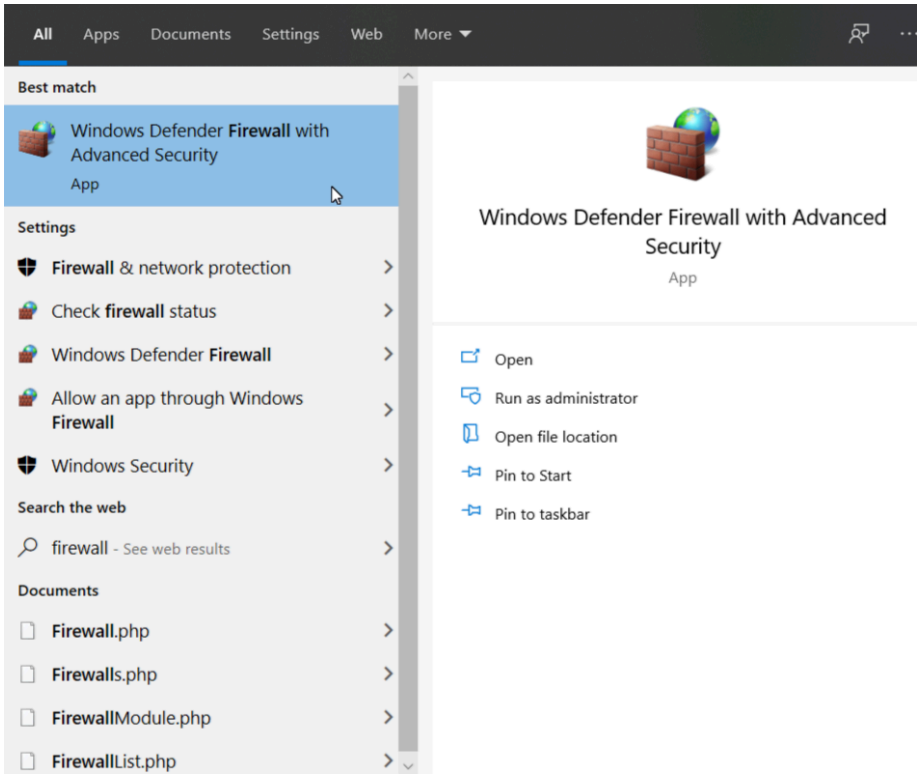
Before installing Data Processor on the Remote Machine, make sure you have all the necessary ports opened to establish connectivity between the MI server and the RDP.

On the MI server, the components involved are **data-processor** (listening on port 2550) and **seed** (listening on port 2551). On the RDP side, remote **data-processor** uses one of the following ports:

- for environment with Simple Installation on a single node, use port 2551
- for environments with Container Orchestration, use port 32551

To open the ports on the Windows RDP host machine:

1.1. In Windows Start Menu start typing Firewall, open Windows Firewall with Advanced Security app

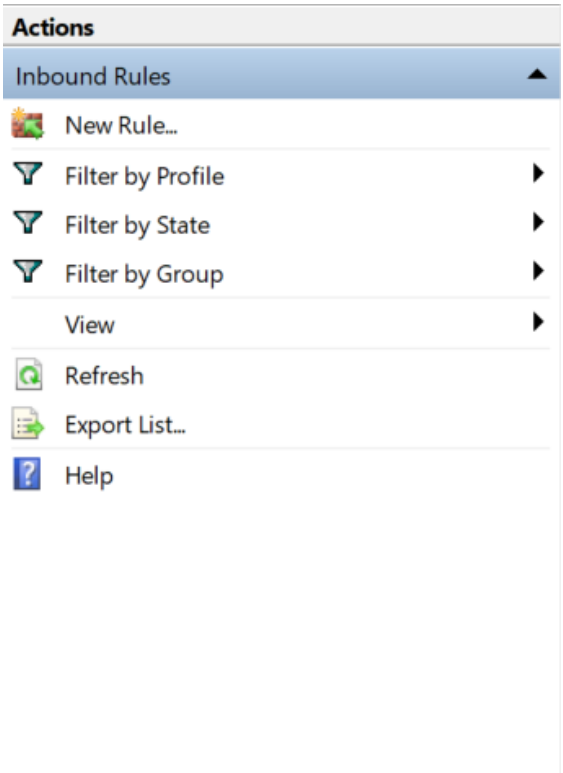


Another way to access the **Windows Firewall** app is through the **Control Panel**:

1. In the Start Menu, start typing *Control Panel*, open this panel
2. Within the **System and Security** section, find and open **Windows Firewall**
3. Select **Advanced Settings**

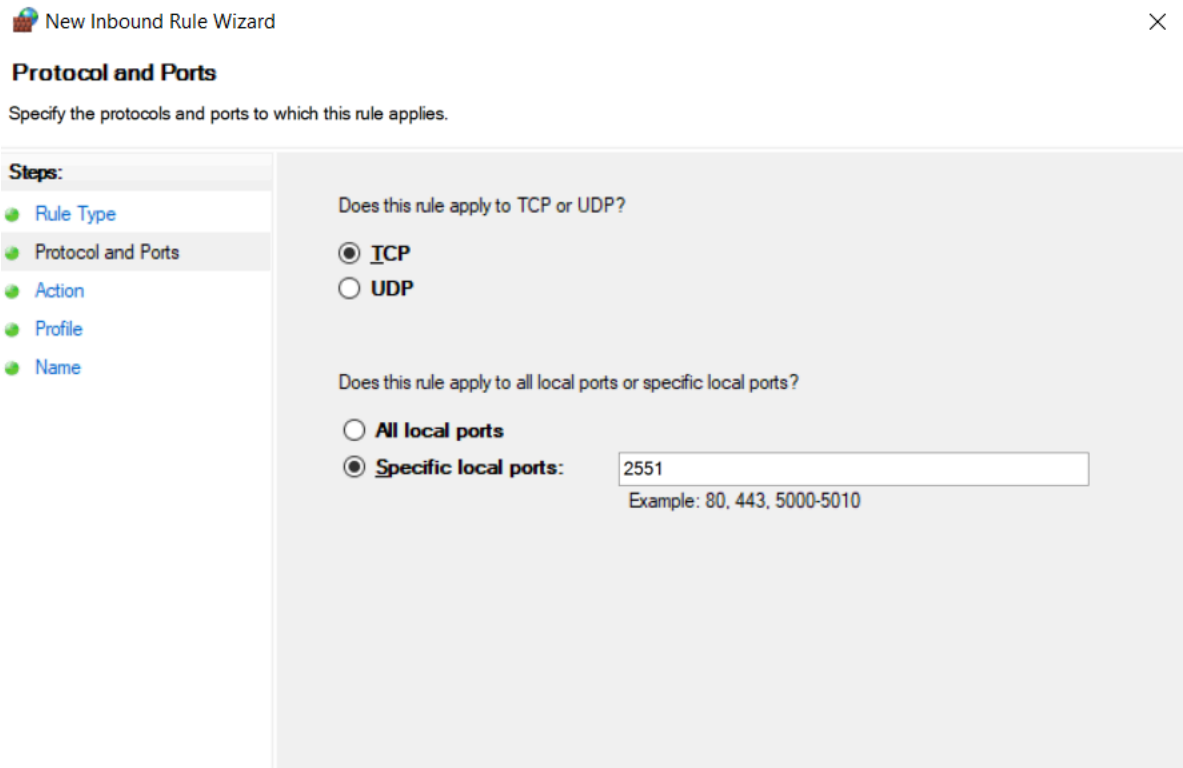
1.2. Select Incoming Rules on the left panel

1.3. On the right panel that appears within Actions section, select "New Rule"



1.4. Configure a rule to allow traffic over TCP port 2551 from

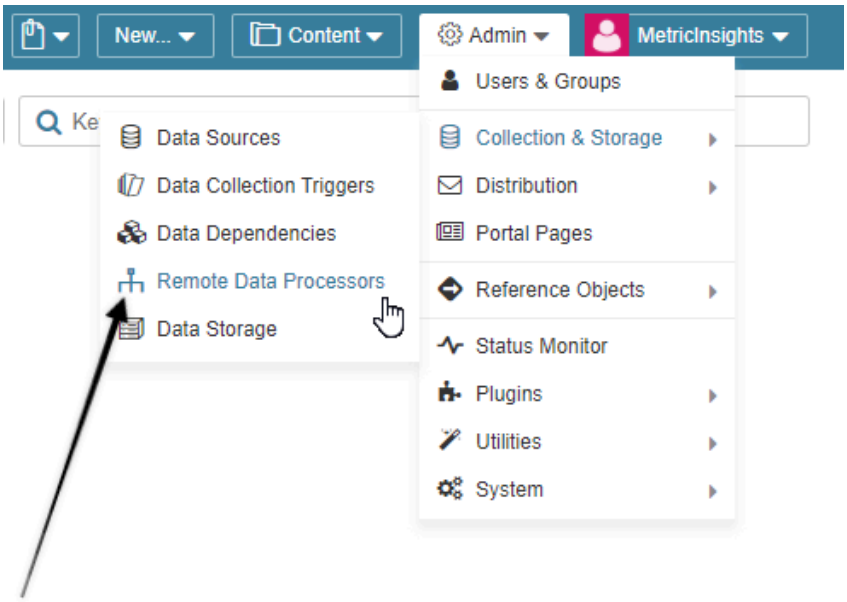
external sources



1.5. Save the new rule

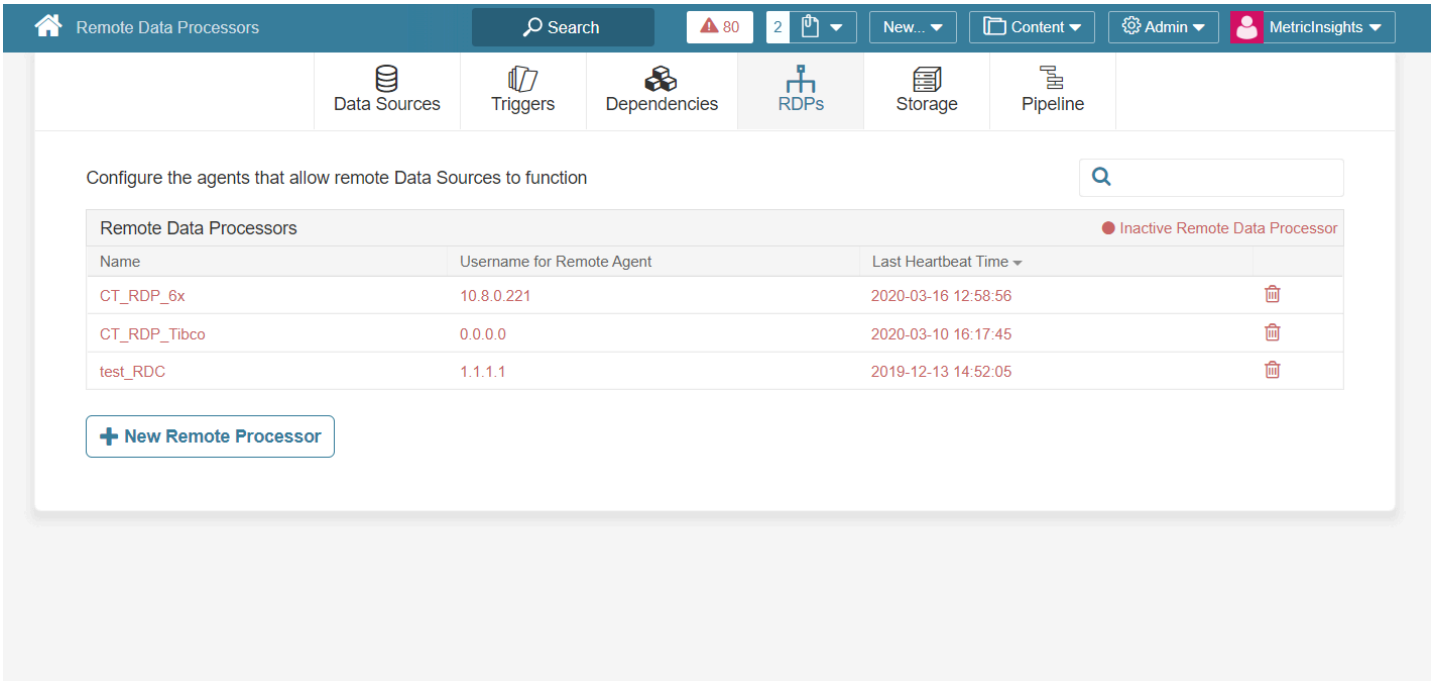
2. Access Admin > Collection & Storage > Remote Data Processors

On the Windows Server where you plan to install the Data Processor, access your Metric Insight application from a web browser. Go to Admin > Collection & Storage > Remote Data Processors.

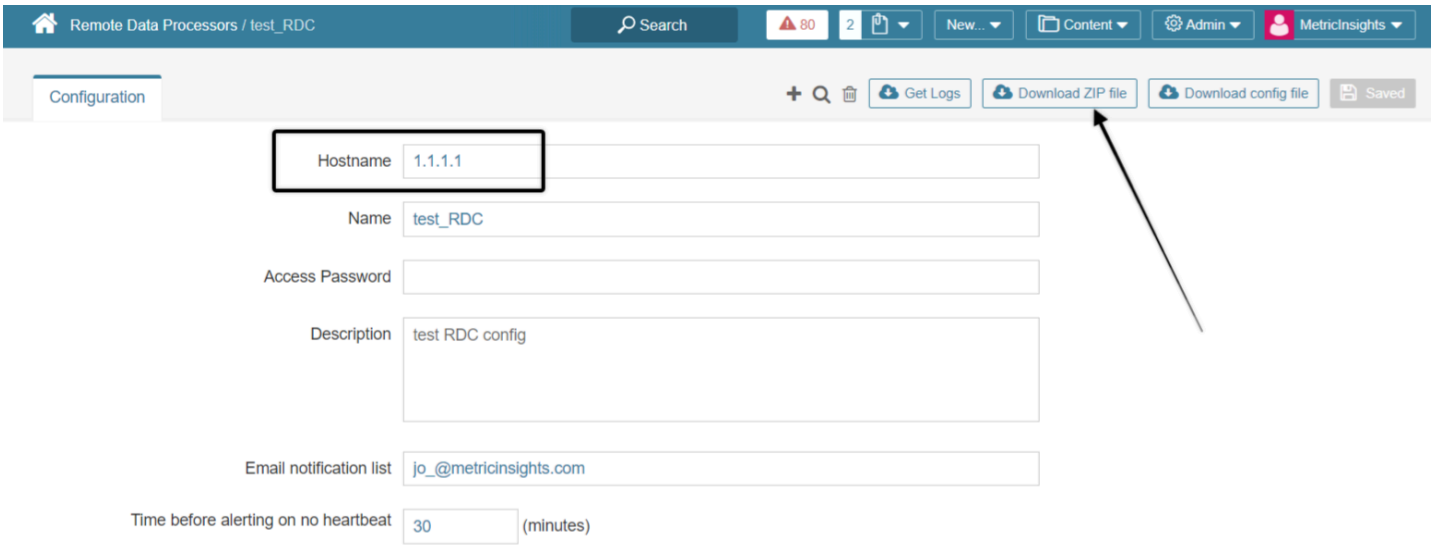


3. Select the Data Processor profile

The Remote Data Processors section lists all your Remote Data Collector profiles along with the Last Heartbeat Time (displays the last successful connection time of the RDP to the Metric Insights application).



4. Download the Data Processor (data-processor) ZIP file

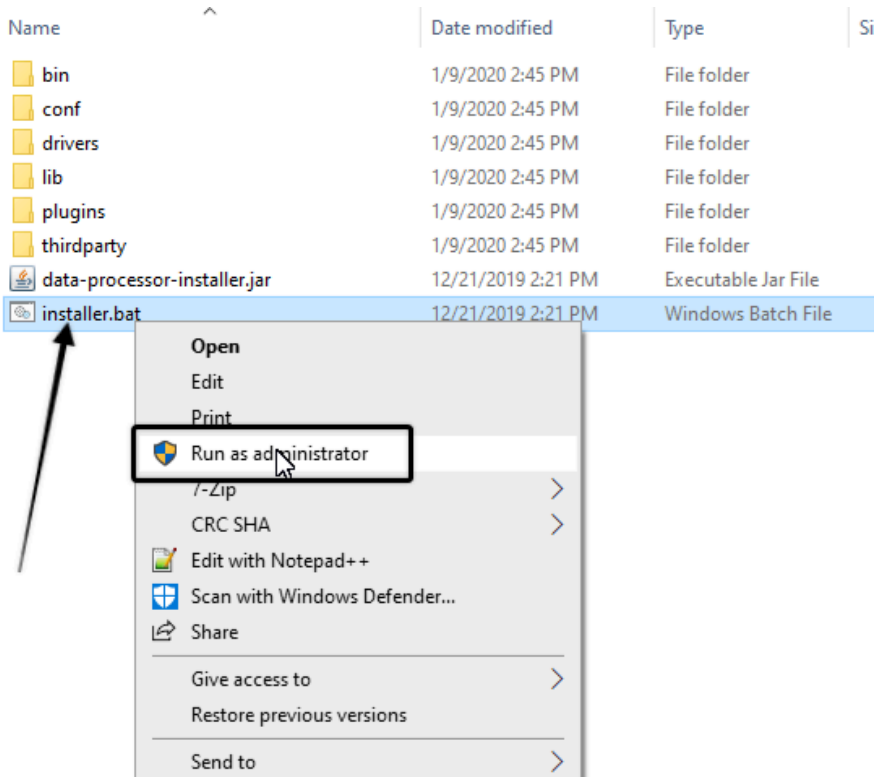


Access the editor of the RDP you want to install on your Windows server. Make sure that the **Hostname** parameter value matches the Windows server IP to install RDP on and click [Download ZIP file] button. While downloading the ZIP file, you might get a warning message that the file can be harmful for your computer. Skip this warning and proceed with downloading.

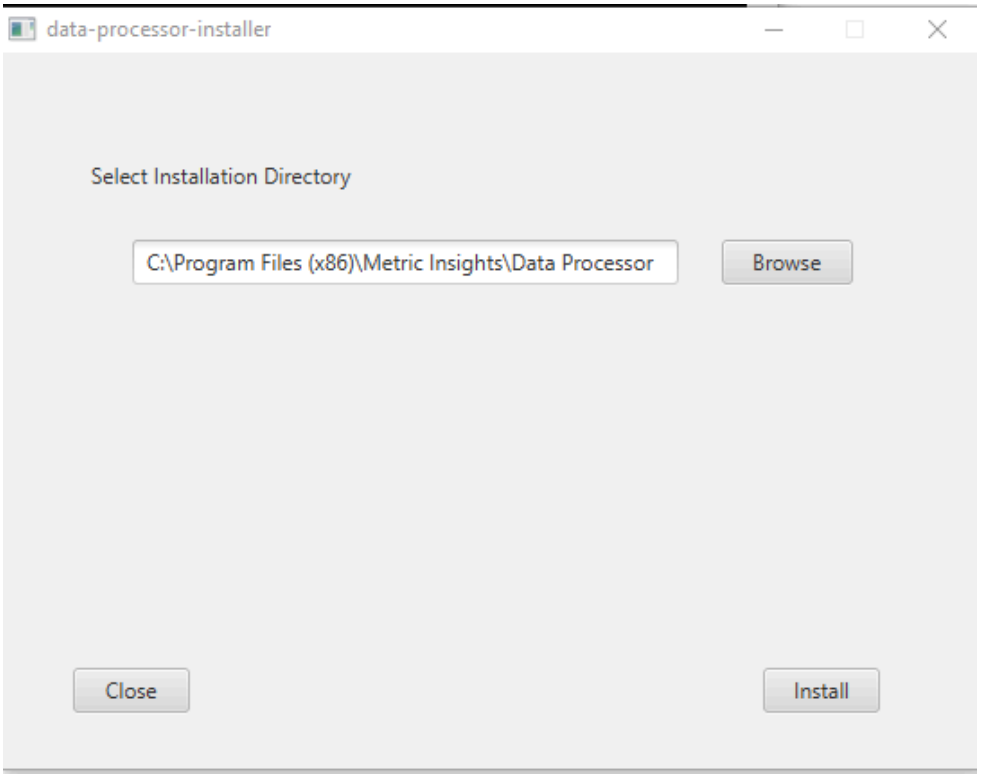
5. Extract the ZIP file contents and launch the Data Processor installer

Within the directory where the ZIP file content was extracted, open `\data-processor` directory and run the `installer.bat` script **as an administrator**.

⚠ To perform this action you need to work as an Admin User on the Windows Machine where Remote Data Processor is to be installed or contact your System Administrator to proceed with the installation.

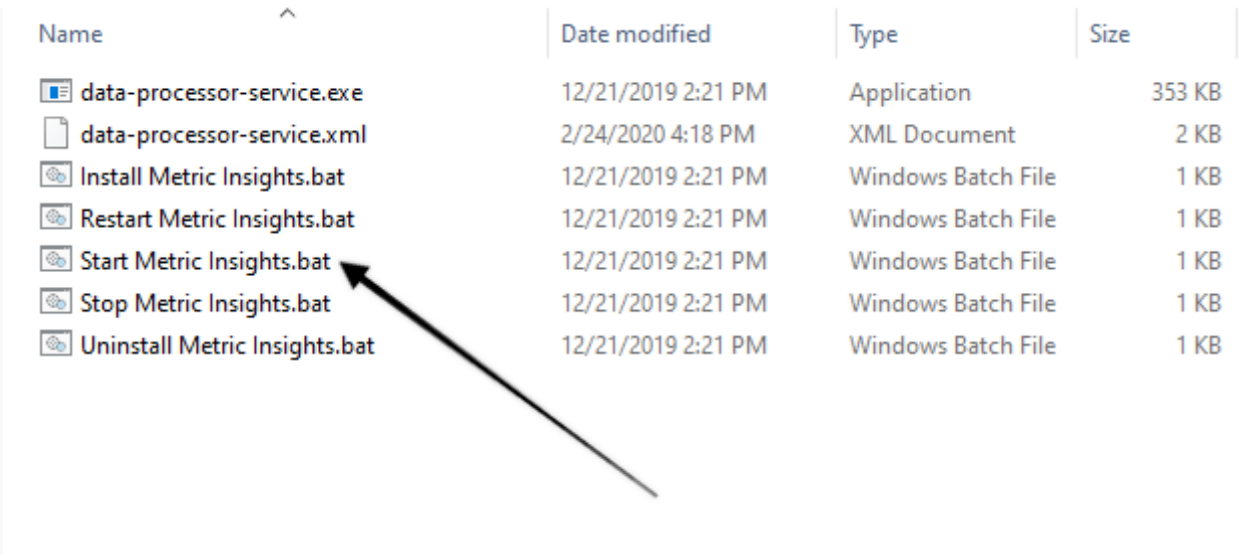


Select installation folder or leave the default, then press "Install" button.



6. Start the Data Processor Windows Service

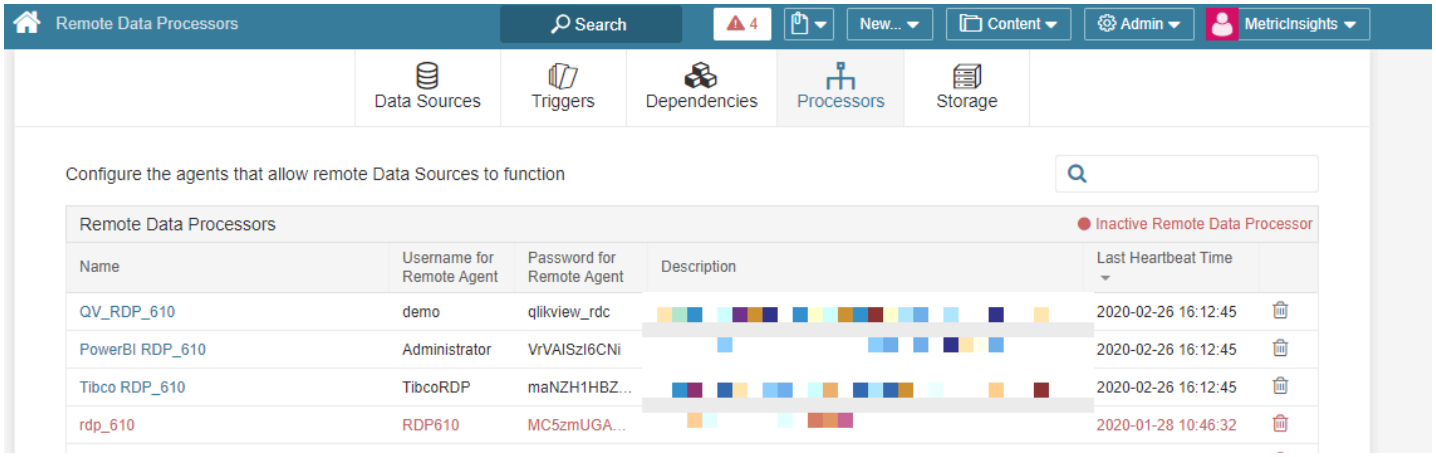
Navigate to the Windows Services list and start the Metric Insights Data Processor Daemon service or run `bin\Start Metric Insights.bat` file



Name	Date modified	Type	Size
data-processor-service.exe	12/21/2019 2:21 PM	Application	353 KB
data-processor-service.xml	2/24/2020 4:18 PM	XML Document	2 KB
Install Metric Insights.bat	12/21/2019 2:21 PM	Windows Batch File	1 KB
Restart Metric Insights.bat	12/21/2019 2:21 PM	Windows Batch File	1 KB
Start Metric Insights.bat	12/21/2019 2:21 PM	Windows Batch File	1 KB
Stop Metric Insights.bat	12/21/2019 2:21 PM	Windows Batch File	1 KB
Uninstall Metric Insights.bat	12/21/2019 2:21 PM	Windows Batch File	1 KB

7. Troubleshooting your Remote Data Processor

7.1. Check if Remote Data Processor is active



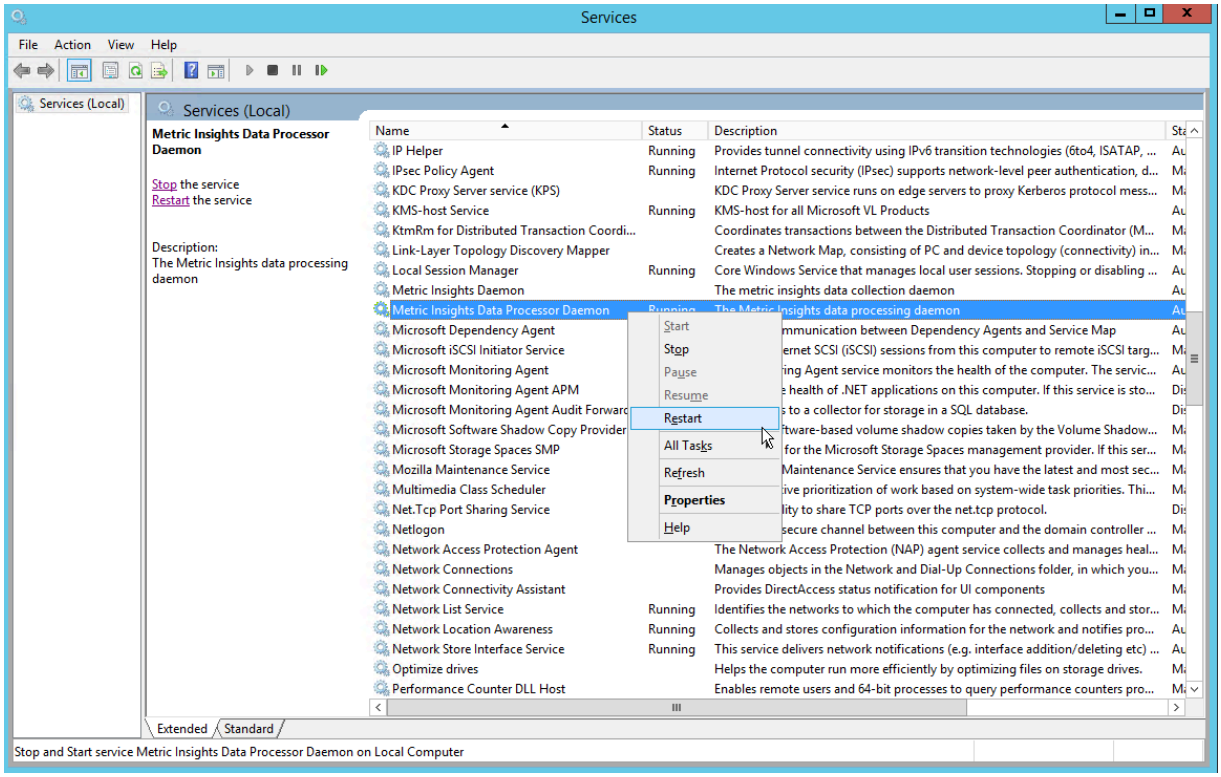
The list of Remote Data Processors helps to define whether the Data Processor profile is active and the connection was established:

- If your Remote Data Processor is active, it will be shown in default black and the column **Last Heartbeat Time** will have a value equal to the time you started RDP Daemon Service plus up to 2 minutes.

- If your Remote Data Processor is not active, it will be displayed in red. The column **Last Heartbeat Time** will have a value with the last time it successfully connected (if have ever been before) or be empty if never connected.

7.2. Restart the Remote Data Processor service

On the Windows machine where Data Processor was installed access Services. In the list of services find **Metric Insights Data Processor Daemon** service >> Start or Restart the service.



7.3. Check if MI and Remote Data Processor servers have opened connectivity

Make sure the Metric Insights server can access the RDP server using the following ports:

- for MI app Docker environment with Simple Installation, confirm that port 2551 is used
- for environments with Container Orchestration, confirm that port 32551 is used

7.3.1. Check server access on Windows

On Windows machine where Remote Data Processor is installed open Command Prompt (from Windows Start Menu type `cmd` or `commant prompt`) and try the following commands (replace `1.1.1.1` with the private IP address of your MI server):

```
ping 1.1.1.1
```

```
telnet 1.1.1.1 2551
```

If you have an orchestrated environment, check port **32551** using `telnet`.

7.3.2. Check server access on Linux

Connect to your MI app linux server, in Command Line execute

```
ping 0.0.0.0
telnet 0.0.0.0 2551
```

Replace `0.0.0.0` with the RDP Windows server private IP address.

If there are any errors on connection attempt contact your System Administrator to open access to the relevant server.

7.4. Verify Java is installed on Windows machine

7.4.1. Check the Java version

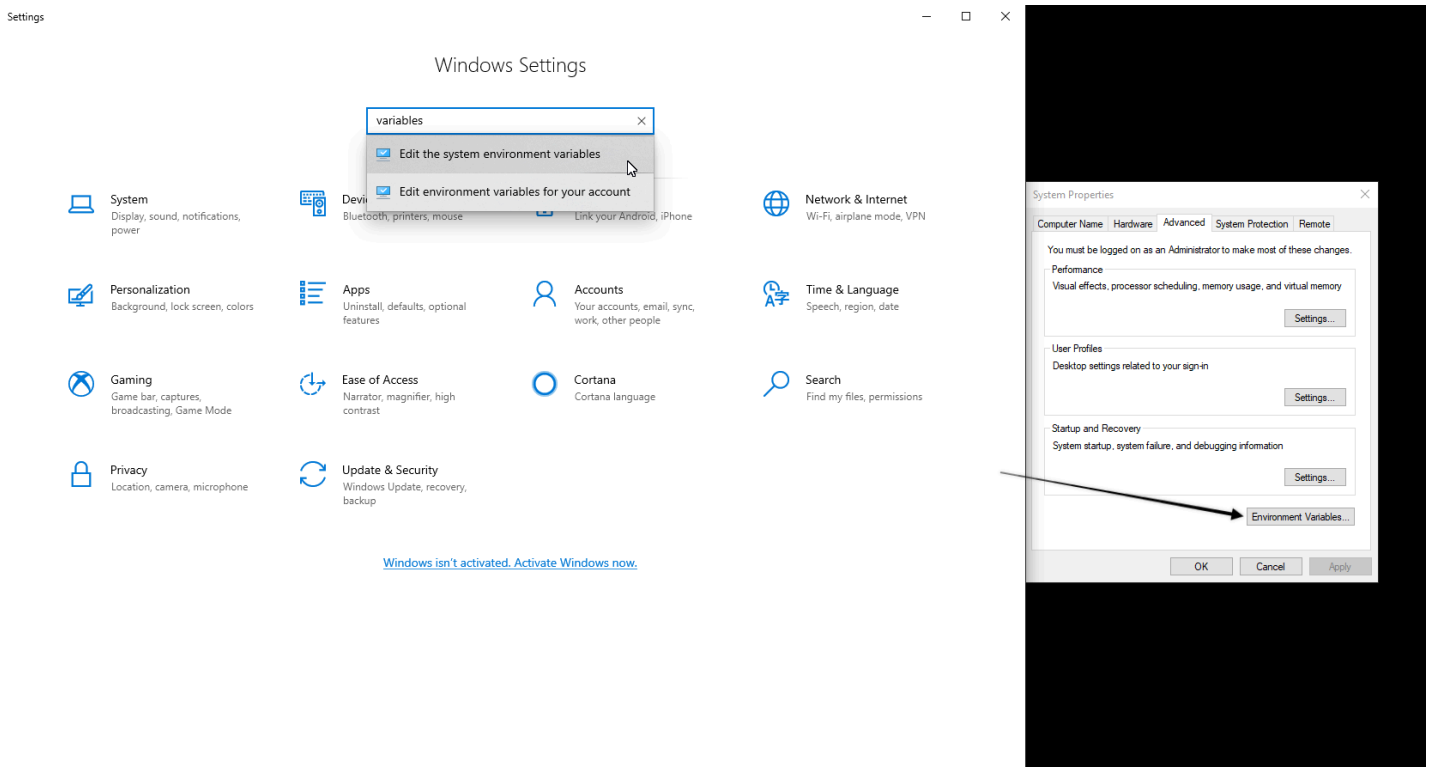
On Windows RDP machine in the Command prompt execute

```
java -version
```

You should see the installed Java version in the output.

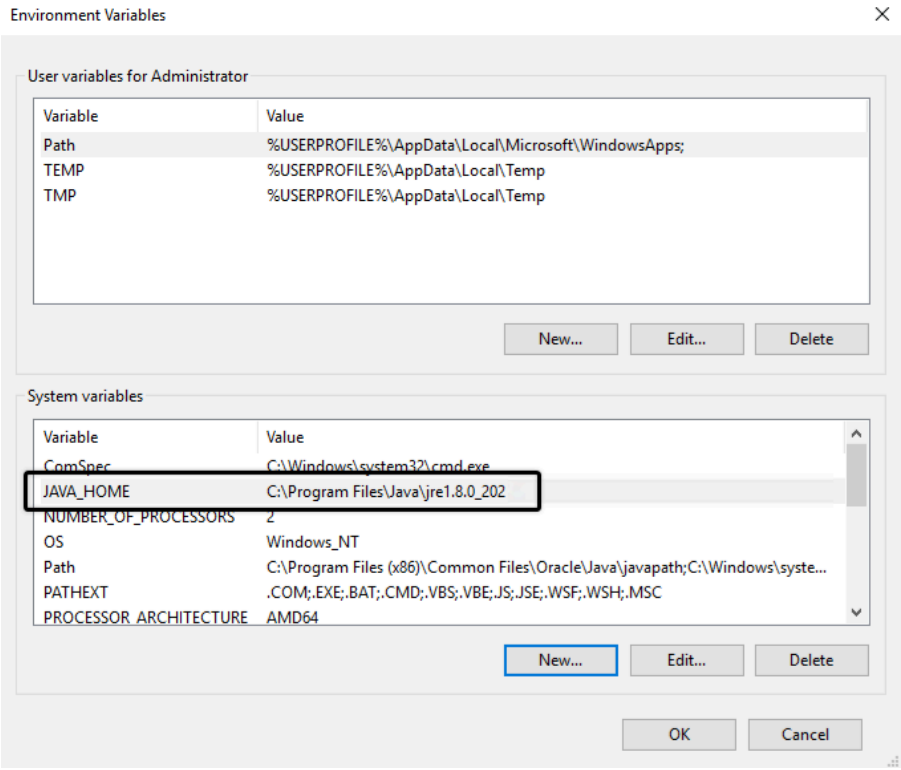
7.4.2. Verify JAVA_HOME environment variable is set

Open Windows Settings, search for Environmental Variables, in the Properties pop-up click the **Environmental Variables** button to open the list of Environmental Variables.



You need to be logged in as Admin User to edit the Environmental Variables.

Check if `JAVA_HOME` variable is specified with the correct path to the directory where Java is installed.



7.5. Check the system recources on Windows machine

If Windows service could never start up or stops within a few minutes check if you have enough CPU and Memory on the Remote Data Processor Server so the Remote Data Processor serice doesn't consume too much system resources.

If you still encounter any problems please contact support@metricinsights.com for further investigation.

If Windows service could never start or stops in a couple of minutes, check whether there is enough CPU and Memory on the Remote Data Processor Windows machine so the Data Processor Daemon service doesn't consume too much system recources.

> If you encounter any other problems establishing connectivity with Remote Data Processor please contact support@metricinsights.com for further investigation.

5. JDBC Connections

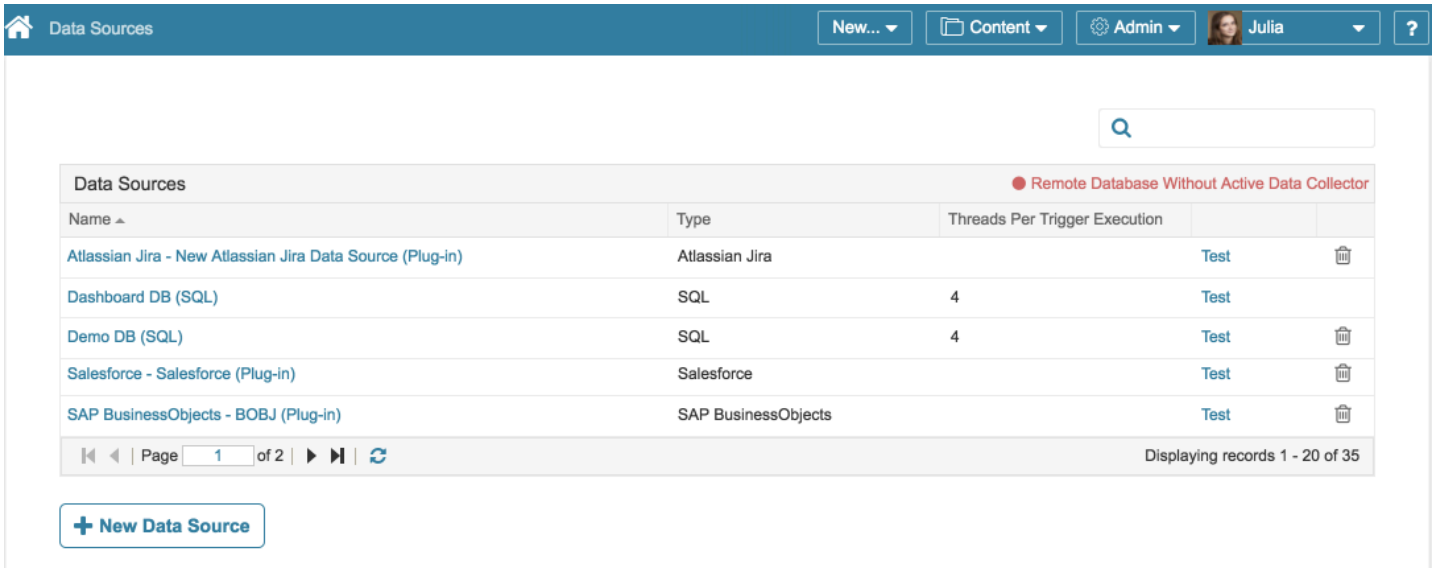
5.1 Establishing a JDBC Connection

Before you can run any queries, you will need to set up a connection to your data sources.

NOTE: The format of Data Sources is slightly different for Version 3.

Video Tutorial

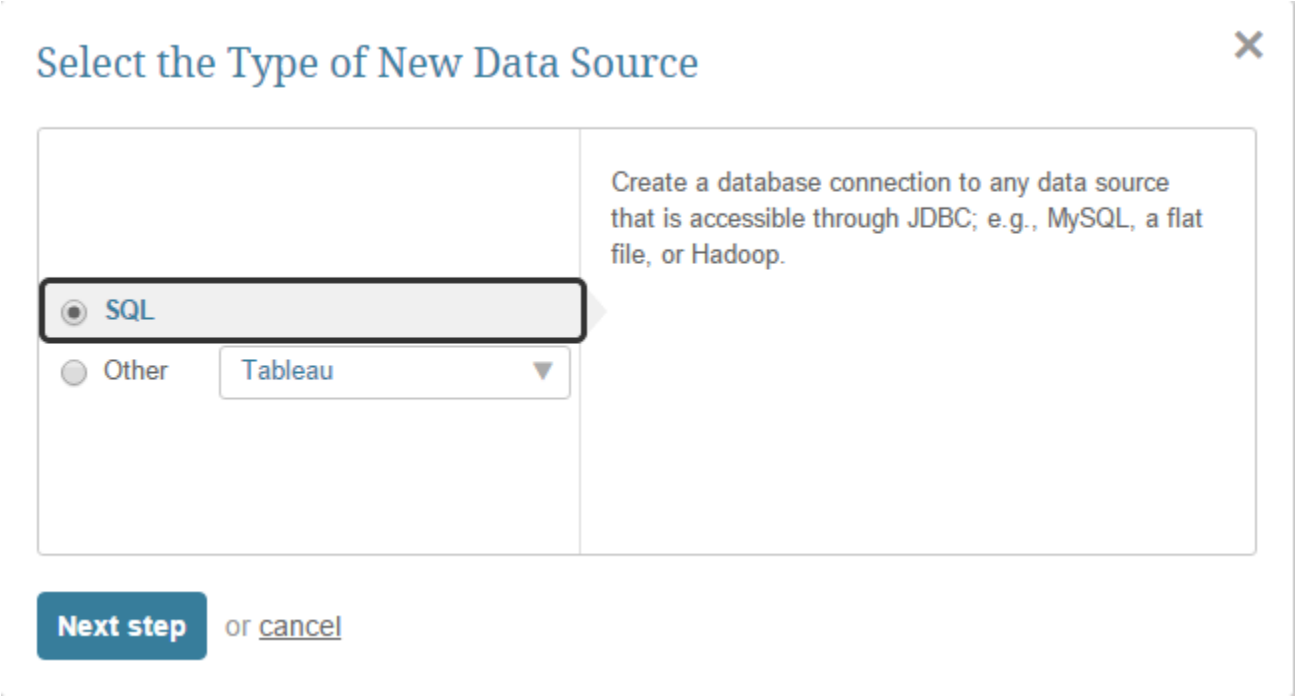
1. Access Admin > Data Sources



At the bottom of the *Data Sources* screen click **[+ New Data Source]**

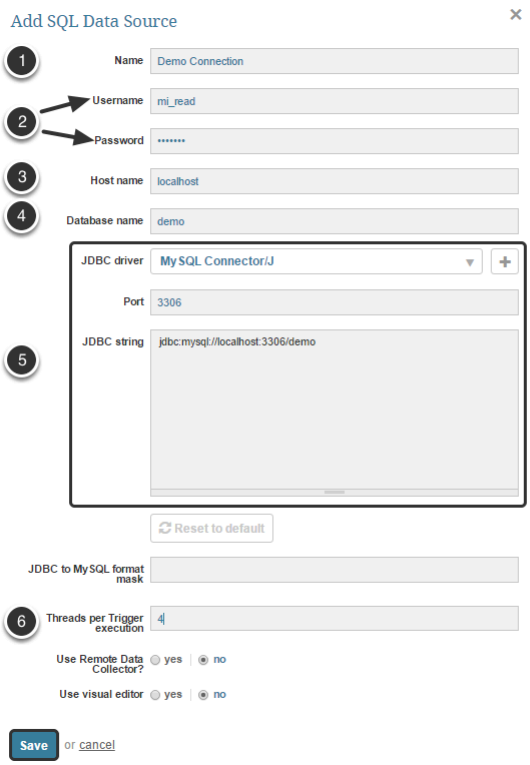
The *Select the Type of New data Source* pop-up opens

2. Select 'SQL' type for the new data source connection



The *Add SQL Data Source* screen opens

3. Describe the Connection

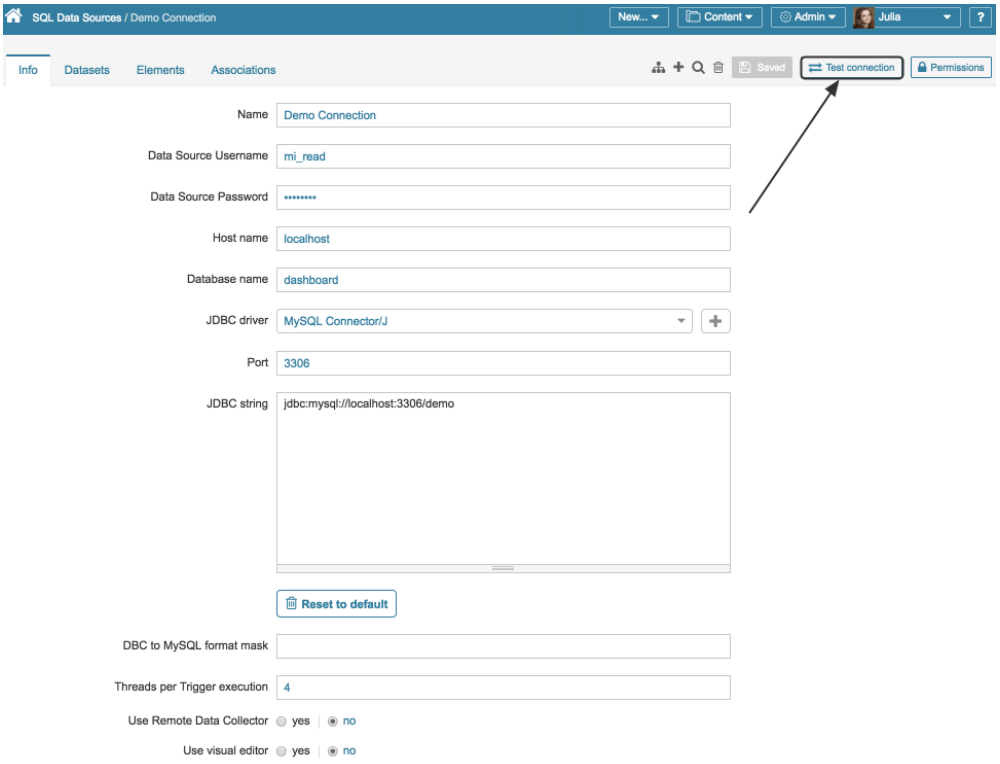


1. Provide a descriptive **Name** for the connection
2. Type in **Username** and **Password** providing access to the database
3. Enter the **Host Name**
4. Enter the **Database name**
5. Select a **JDBC Driver.** The **Port** number as well as JDBS string are set by default, based on your choice of **JDBC Driver**. Change it if necessary. **NOTE:** If you do not see the the driver that you wish to use in the drop-down provided, contact Metric Insights for assistance.
6. Optionally, specify the maximum number of concurrent **Threads per Trigger execution** to be used in background processing when the system updates metrics and reports for this data source. If you do not specify any value for this setting, batch data collection processing will be single-threaded

Save your changes

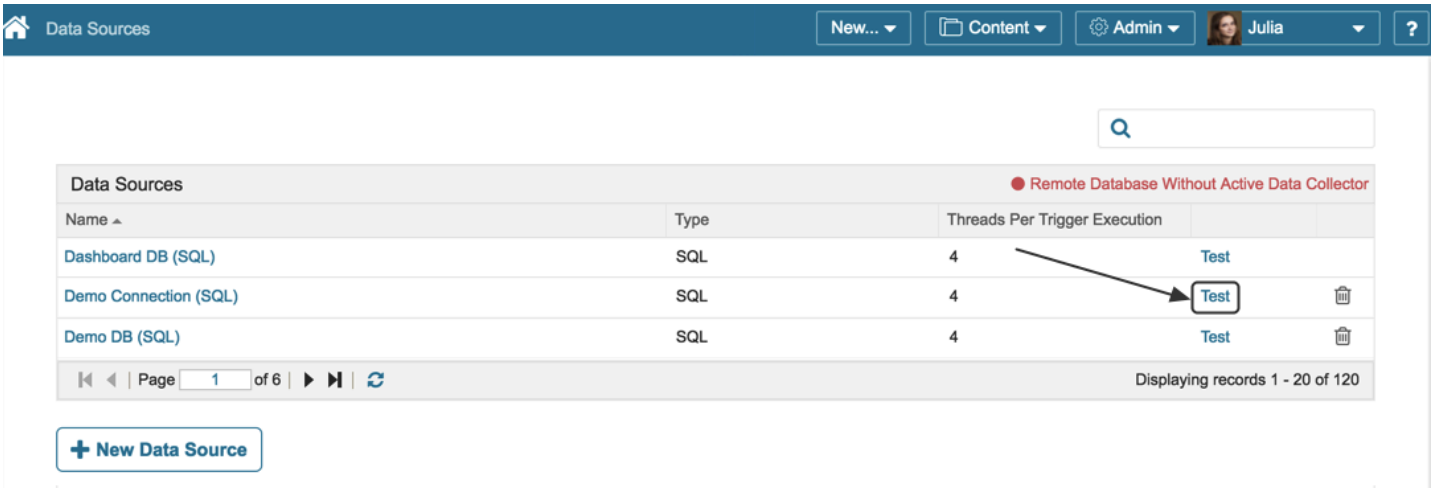
NOTE: The JDBC string will be created automatically based on your other inputs. In some cases, however, it will not be possible to infer the correct string without additional inputs. If the Connection Test fails, please check the documentation for your JDBC driver.

4. Validate created connection from 'Data Source Editor'



Alternatively, Data Source connection can be also validated from the Data Sources list as shown at the screen below

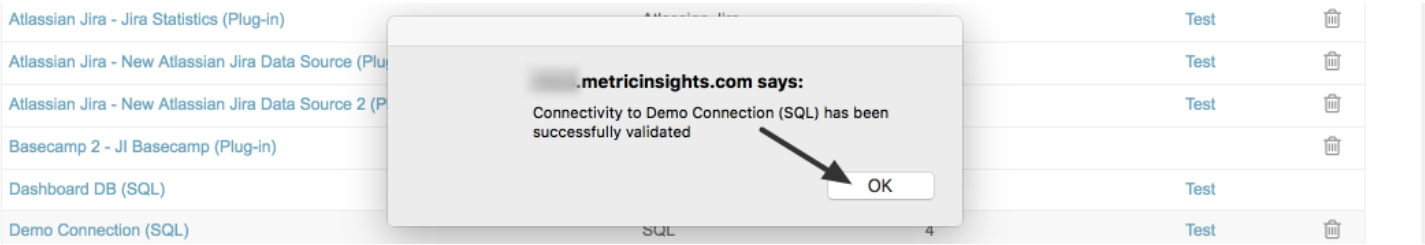
5. New connection can be found in Data Sources list



Click **Test** to verify connectivity to the data source

NOTE: The test feature does not currently work for testing connections to Apache Hive

6. Confirm successful validation



If the connectivity is established, the confirmation message appears; click **OK** to continue

NOTE: It is not possible to directly test Hive connectivity from the connection editor since there is no standard query that can be run against a HiveQL instance. Contact Metric Insights for assistance in validating a HiveQL connection.

5.2 Add a new JDBC Driver

The **JDBC Drive**r drop-down in the *Add/Edit a SQL Data Source Editor* includes the list of JDBC drivers that have been installed on the **Metric Insights** server.

You may add JDBC drivers to this list by following the steps in this article.

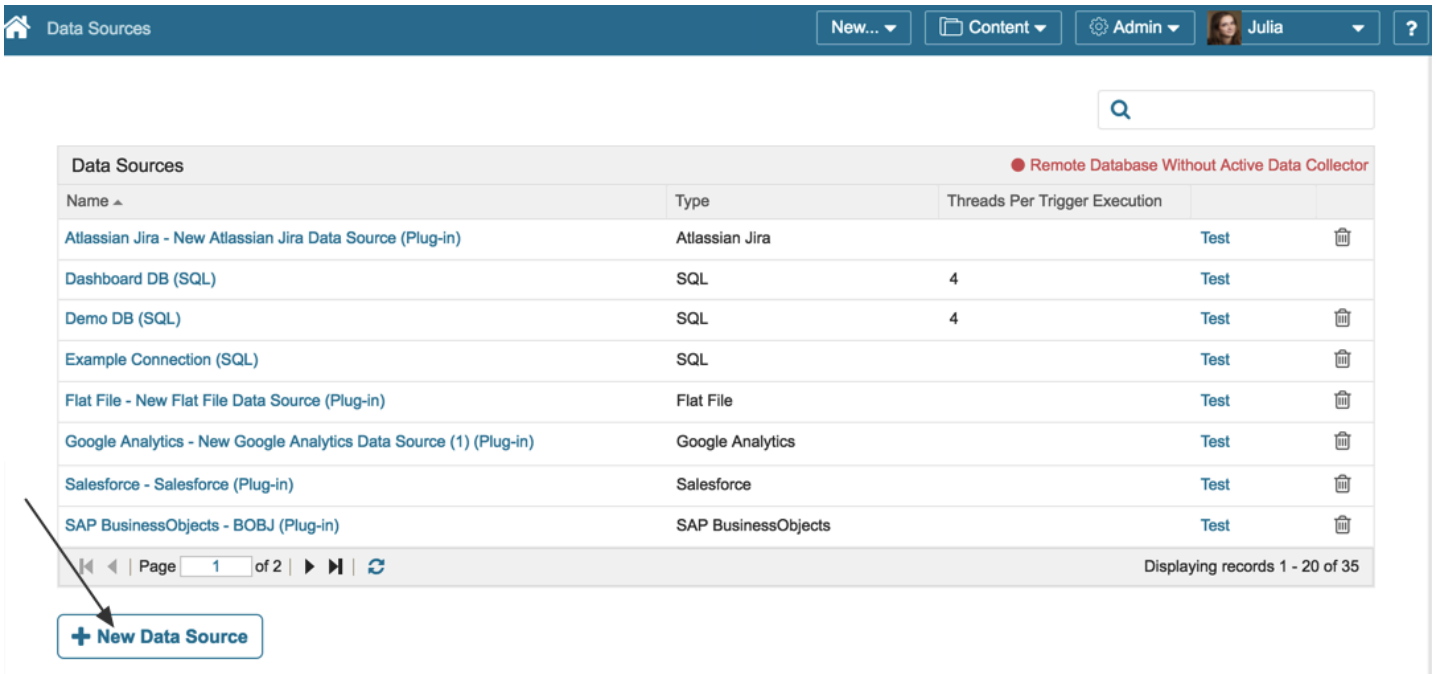
PREREQUISITES:

- You must have the driver JAR file stored locally on your computer
- For the last step you will need to request Metric Insights to Register this driver.

1. Obtain the JAR file

1. Identify the proper JDBC driver JAR file for the driver to be added
2. Download to your local machine

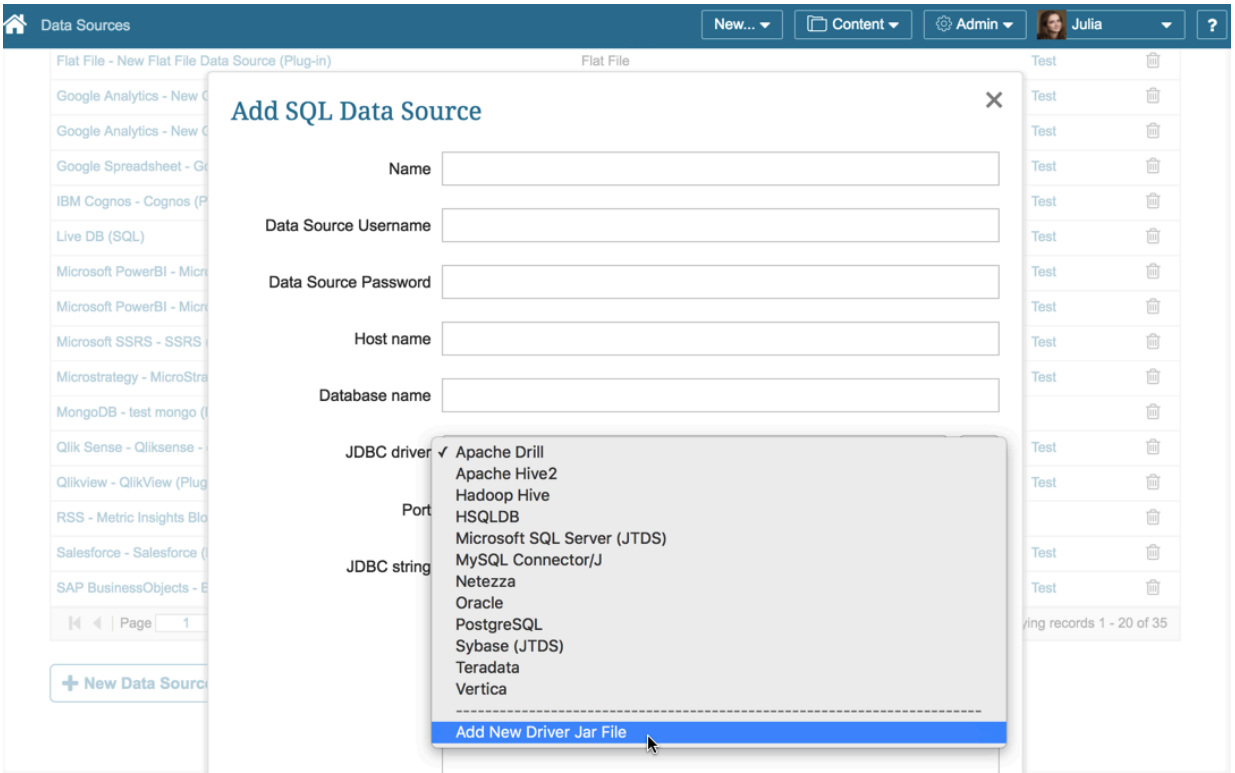
2. Access Data Sources from the Admin menu



At the bottom of the list click **[+ New Data Source]**

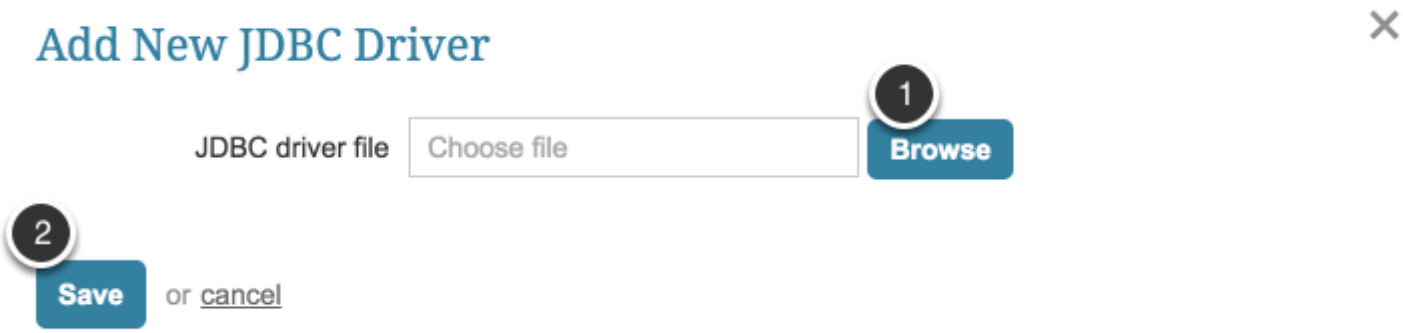
Optionally, you can **Edit** an existing **SQL Data Source** on its editor by clicking its **Name** link and add your new JDBC driver from there; if this is your approach, skip to Step 4.

3. Add a new Driver



From the bottom of the list select **Add New Driver Jar File**

4. Select the new driver



1. Use the **Browse** button to find and **Open** the driver (.jar file) that you downloaded and saved to your local hard drive
2. Once the Jar file is in the text box, click **SAVE**

5. Register Driver in Metric Insights

Contact Metric Insights and Metric Insights will register this Driver in the system. What Metric Insights will do is add info to the system (mysql jdbc_driver table) that includes your jdbc driver information. E.g., jdbc string template to use (e.g., jdbc:postgresql://<host>:<port>/<db>) , sample SQL statement to test connectivity and jdbc class name to invoke in the driver (e.g., org.postgresql.Driver).

NOTE: Once complete you will then be ready to create a Data Source using this new JDBC driver

5.3 Validation errors: "Value '0000-00-00' can not be represented as java.sql.Date"

ANALYSIS:

This is a MySQL-specific problem that does not happen often. It is caused by storing zero dates ('0000-00-00 00:00:00') in MySQL and trying to convert those into date objects in Java. Unfortunately, Java does not understand dates in this format, so the MySQL JDBC driver will throw this error by default.

RECOMMENDED ACTION:

Apart from modifying the source data, the easiest solution is to modify the **JDBC connection URL** for the affected Data Source to include **?zeroDateTimeBehavior=convertToNull** as shown below.

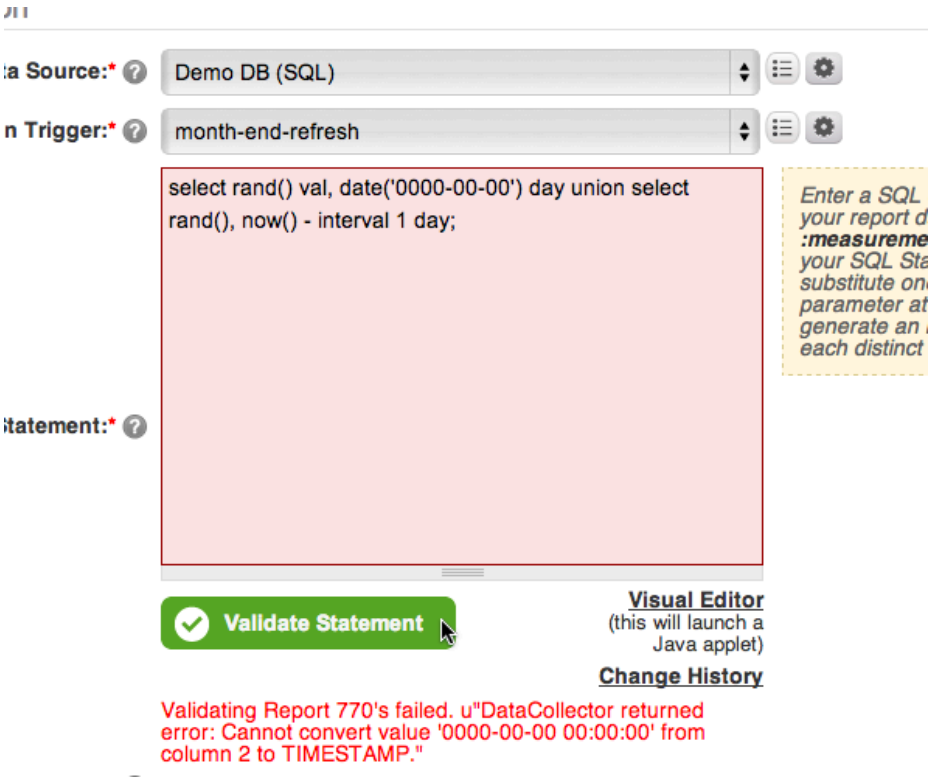
REFERENCES:

For more information about this JDBC parameter, see the Datetimes bullet under section 20.3.3.3.3 of the MySQL Manual.

1. Problem examples:

1.1. Example: "Value '0000-00-00' can not be represented as java.sql.Date"

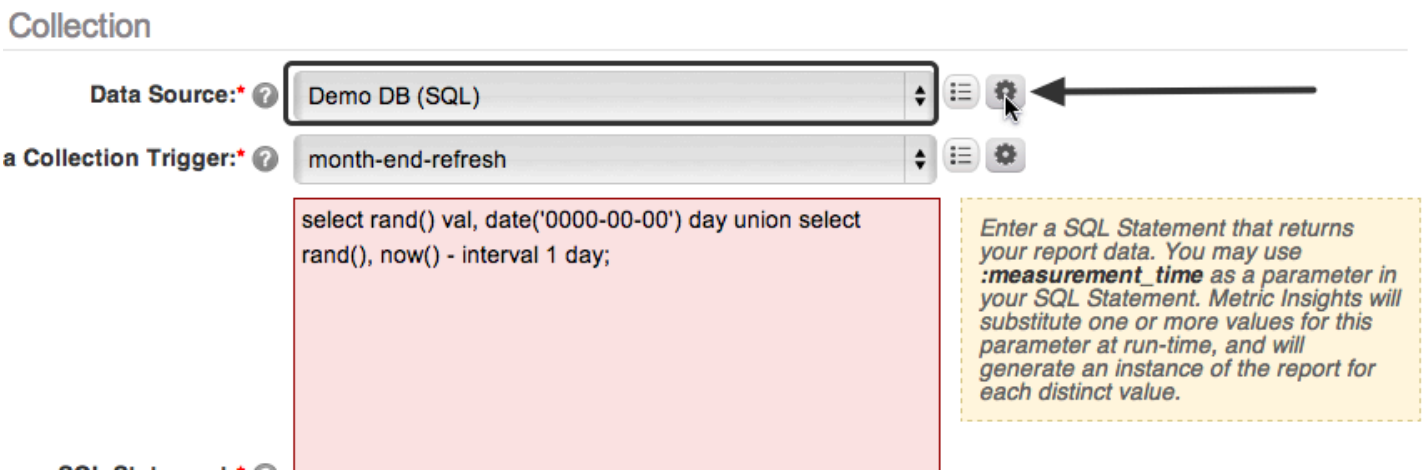
1.2. Example: "Cannot convert value '0000-00-00 00:00:00' from column N to TIMESTAMP"



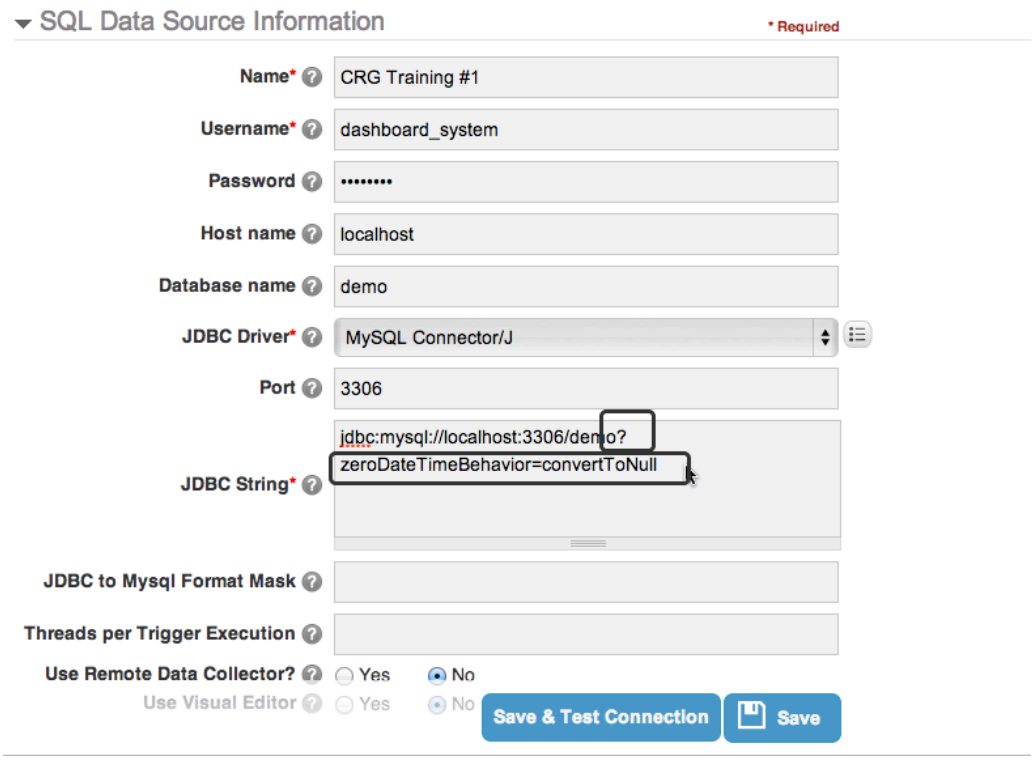
2. Solution:

2.1. Navigate to the data source editor for the affected data source

Make sure the data source drop down is pointing to the appropriate data source, then click the gear icon to the right to edit the data source.



2.2. Fix the SQL Data Source URL



The suggested fix is to append '?zeroDateTimeBehavior=convertToNull' to the end of the JDBC string.

So the original string

```
jdbc:mysql://localhost:3306/demo
```

would become

```
jdbc:mysql://localhost:3306/demo?zeroDateTimeBehavior=convertToNull
```

6. Sourcing Data from Adobe Analytics

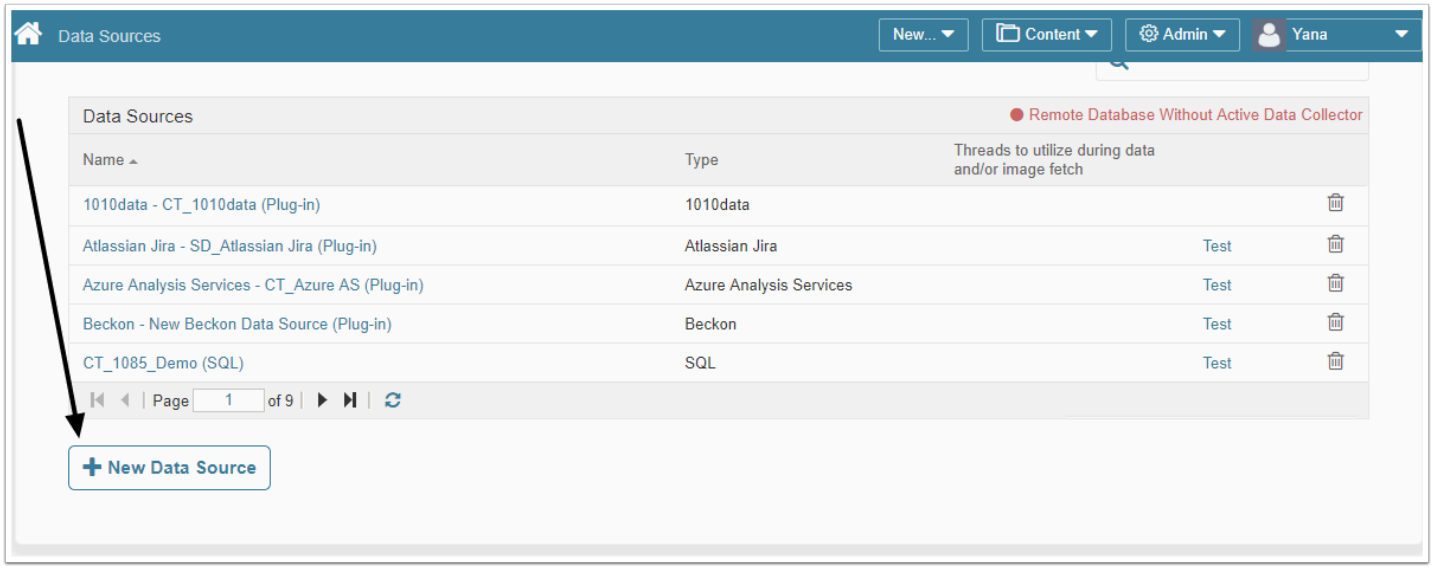
6.1 Establish connectivity to Adobe Analytics

This article describes how to connect to **Adobe Analytics** server in order to load data into Datasets and Reports in Metric Insights.

General instructions on setting up data sources based on plugins can be found here

[Release 6.x]: 'Remote Data Collector' renamed to 'Remote Data Processor'

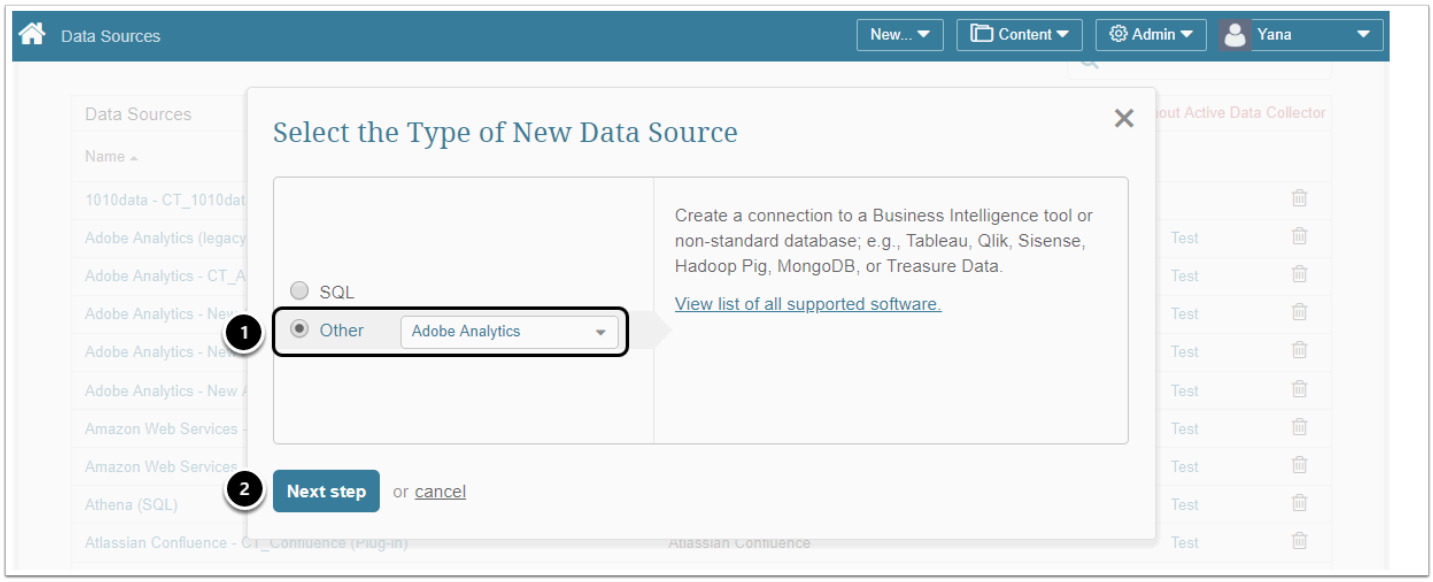
1. Access Admin > Data Sources



Select **[+ New Data Source].**

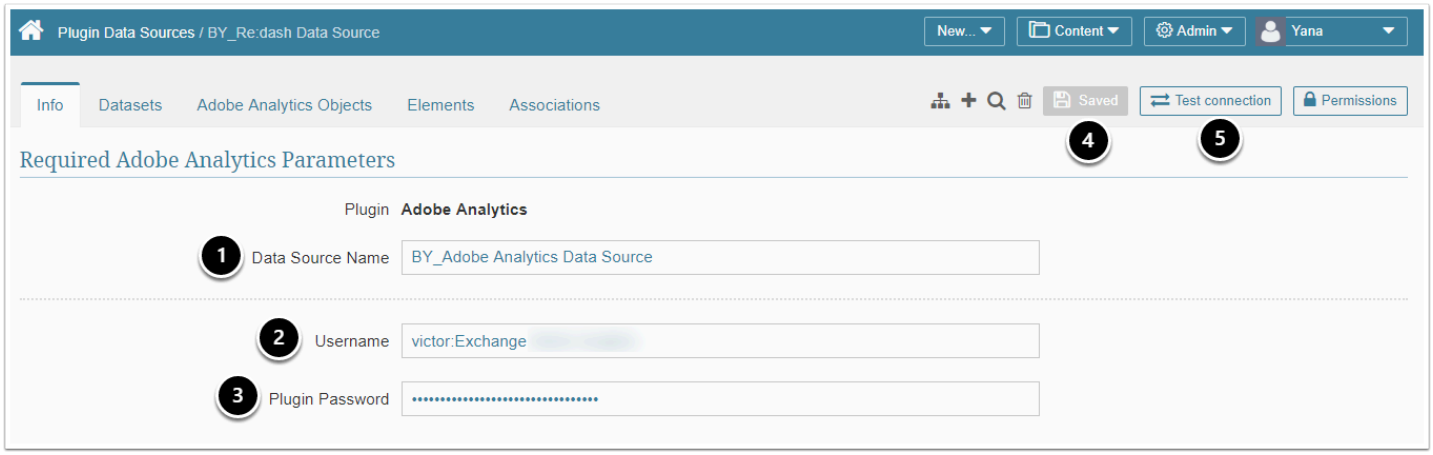
The *Select the Type of New Data Source* pop-up opens.

2. Select the Type of New Data Source



1. Select **"Other"** Data Source Type and choose **"Adobe Analytics"** from the drop-down list.
2. Move to the **Next step.**

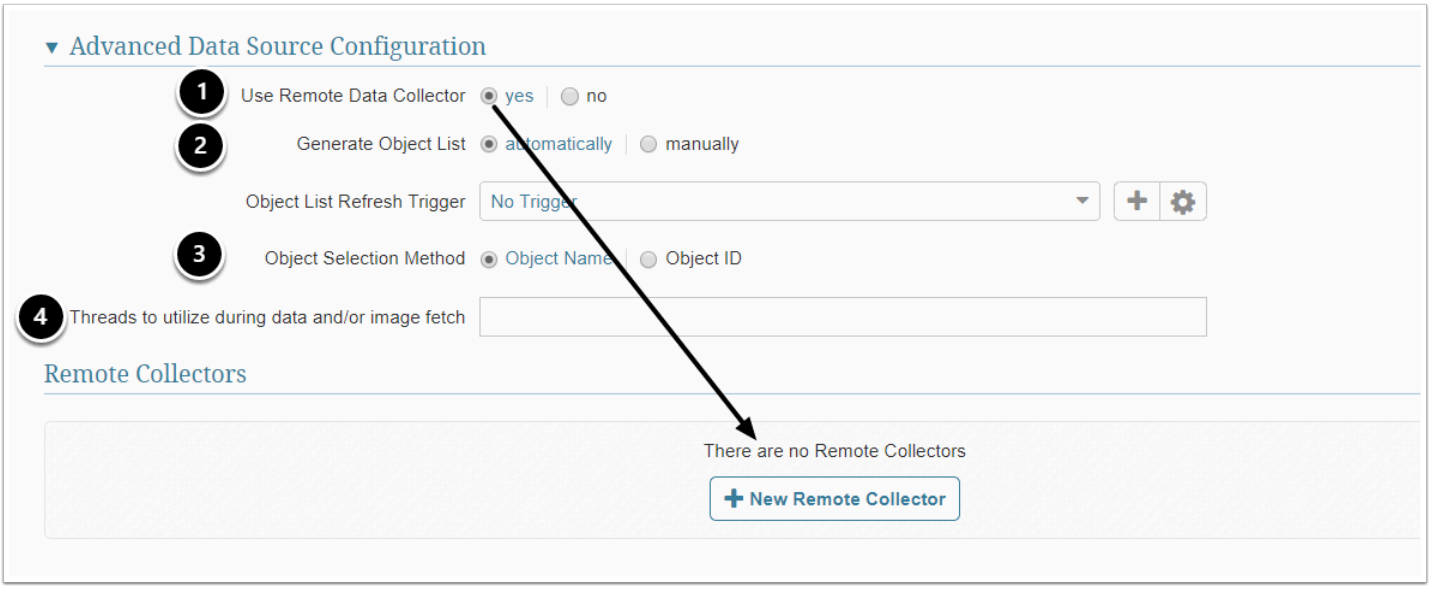
3. Provide the Required Parameters



1. **Data Source Name:** provide a unique name for your Data Source
2. **Username:** enter your credential in the following format **[Username :Company Name]**
3. **Plugin Password:** specify a Secret Key for this connection profile
4. **Save** your entries
5. **Test Connection**

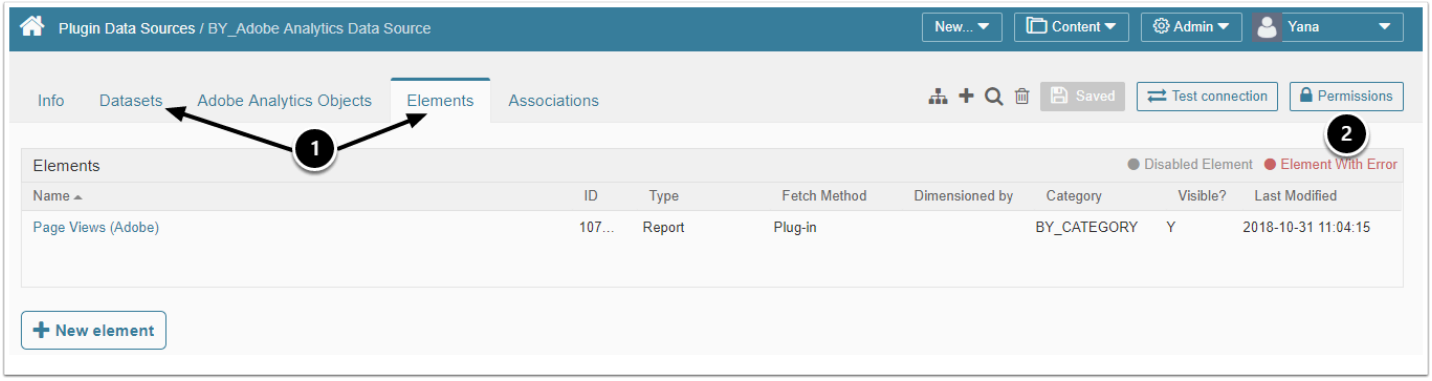
If your connection is successful, you may move on to **Advanced settings**.

4. Advanced Settings



1. **Use Remote Data Collector:** is set to "no" by default. If required, switch to "yes" and add a Remote Data Collector by clicking **[+New Remote Collector]**.
2. **Generate Object List**: This setting influences options available in the ***Adobe Analytics Objects*** tab:
   - **automatically**:
     - In the ***Adobe Analytics Objects*** *tab* click **Refresh list** and all Reports are going to be refreshed/add by the system
     - (Optional)[New in 5.3.2] **Object List Refresh Trigger** will appear allowing you to schedule the Refresh function to run automatically
   - **manually**:
     - Reports must be added one-by-one or via CSV file in the ***Adobe Analytics Objects*** *tab*
3. **Object Selection Method:** specify how Adobe Analytics Objects will be fetched
4. Optionally, state the maximum number of concurrent **Threads to utilize during data and/or image fetch** to be used in background processing when the system updates Objects for this Data Source. If you do not specify any value for this setting, batch data collection processing will be single-threaded.

5. Other Settings



1. You can create Datasets or Elements directly from the respective tabs
2. Click **Permissions** to assign permissions to the Data Source to Groups or Power Users

What's next?

[How to collect data from Adobe Analytics]

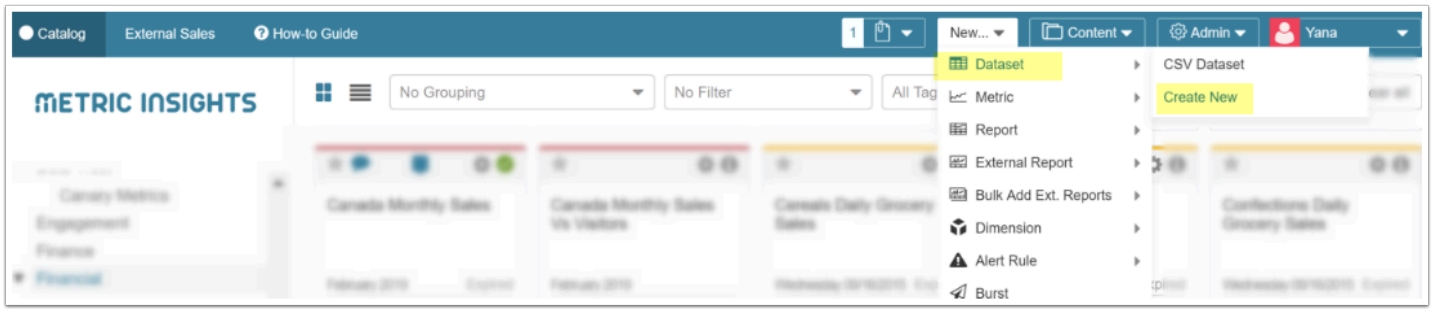
6.2 Collect data from Adobe Analytics

A Metric Insights' Dataset can be populated automatically based on data fetched from Adobe Analytics.

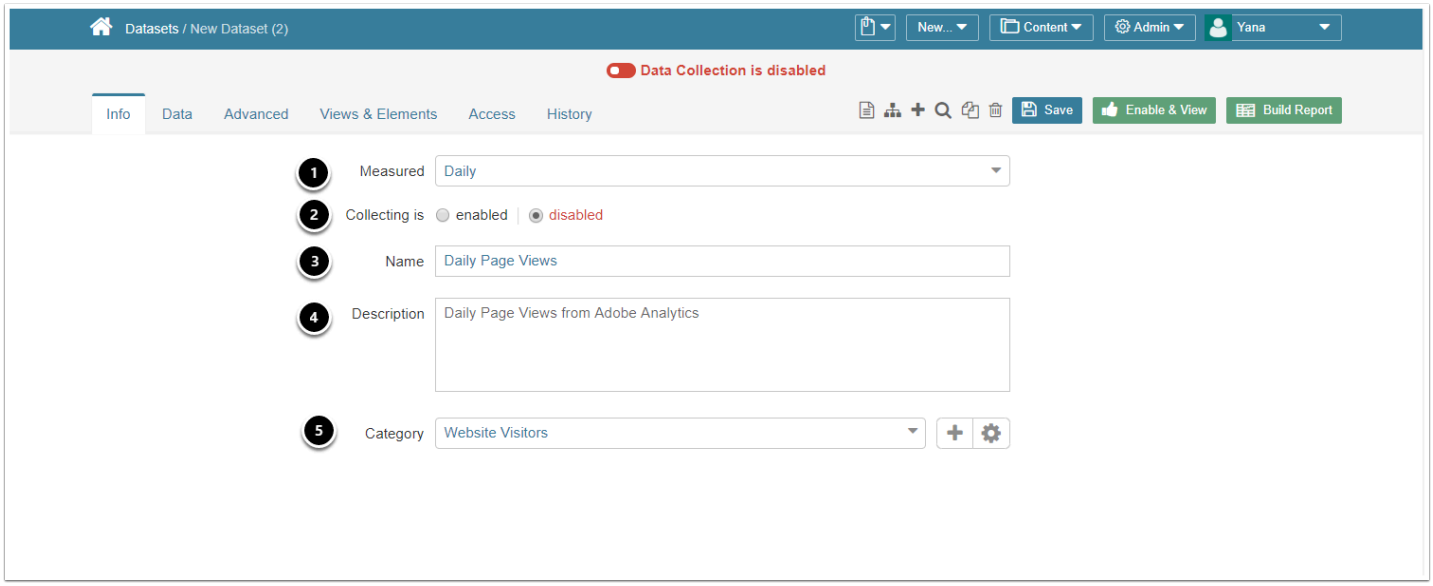
PREREQUISITE:

You must have already established connectivity to your **Adobe Analytics** server via the respective plugin connection profile.

1. Access New > Datasets > Create New



2. Dataset Editor > Info tab



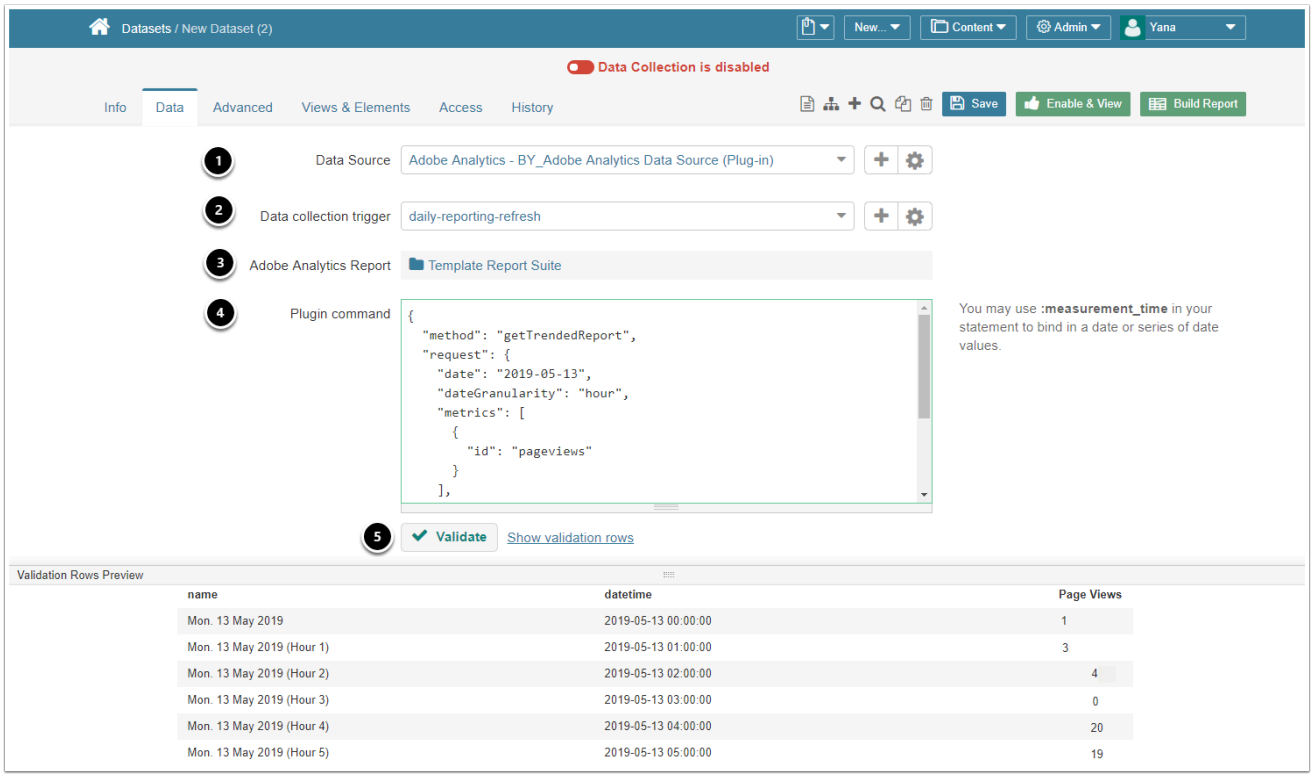
Enter the basics:

1. **Measured:** select the measurement interval that applies to the level of aggregation that you want in your result set.

2. **Collecting**: new Datasets are always disabled by default to make sure that you can take time to configure them properly before enabling. This setting is duplicated at the top of the screen.
3. **Name:** provide a unique name for your Dataset. Preferably, the Dataset name should explain what kind of data it contains.
4. **Description:** optionally, provide any additional information about your Dataset.
5. **Category:** specify the Category where you Dataset will be placed.

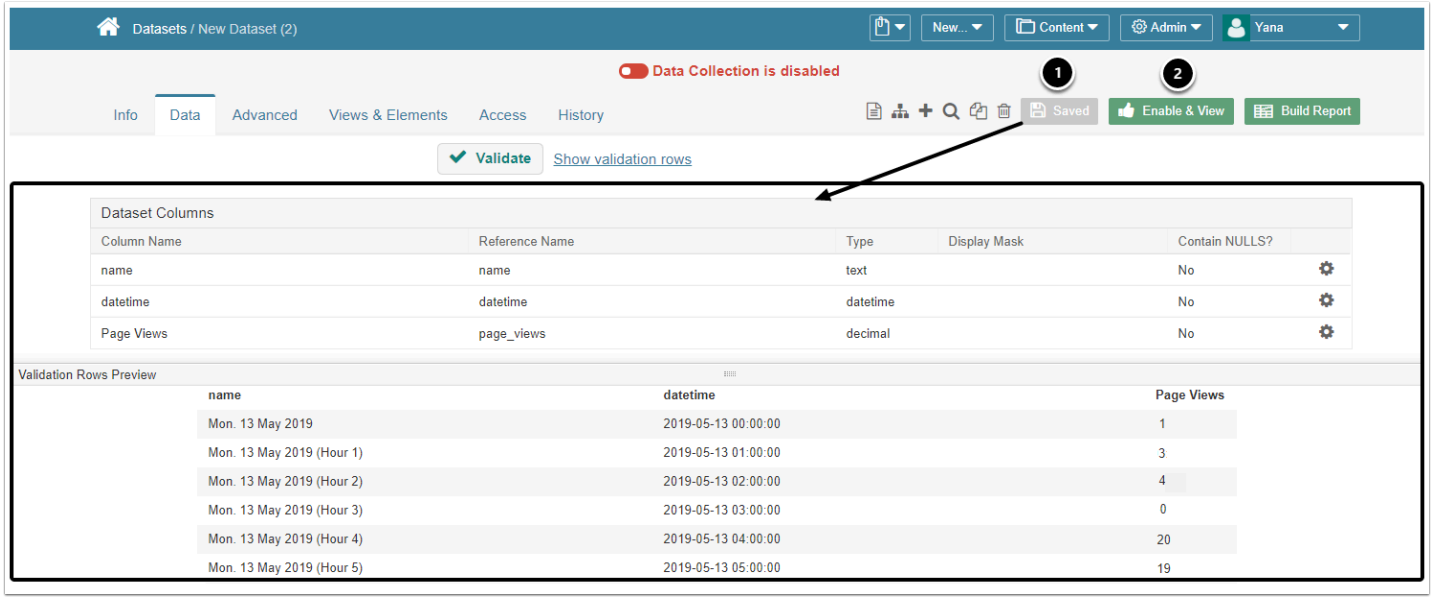
Move to the ***Data tab*** to define the source of data and how often it should be updated.

3. Define the Settings for Data Collection



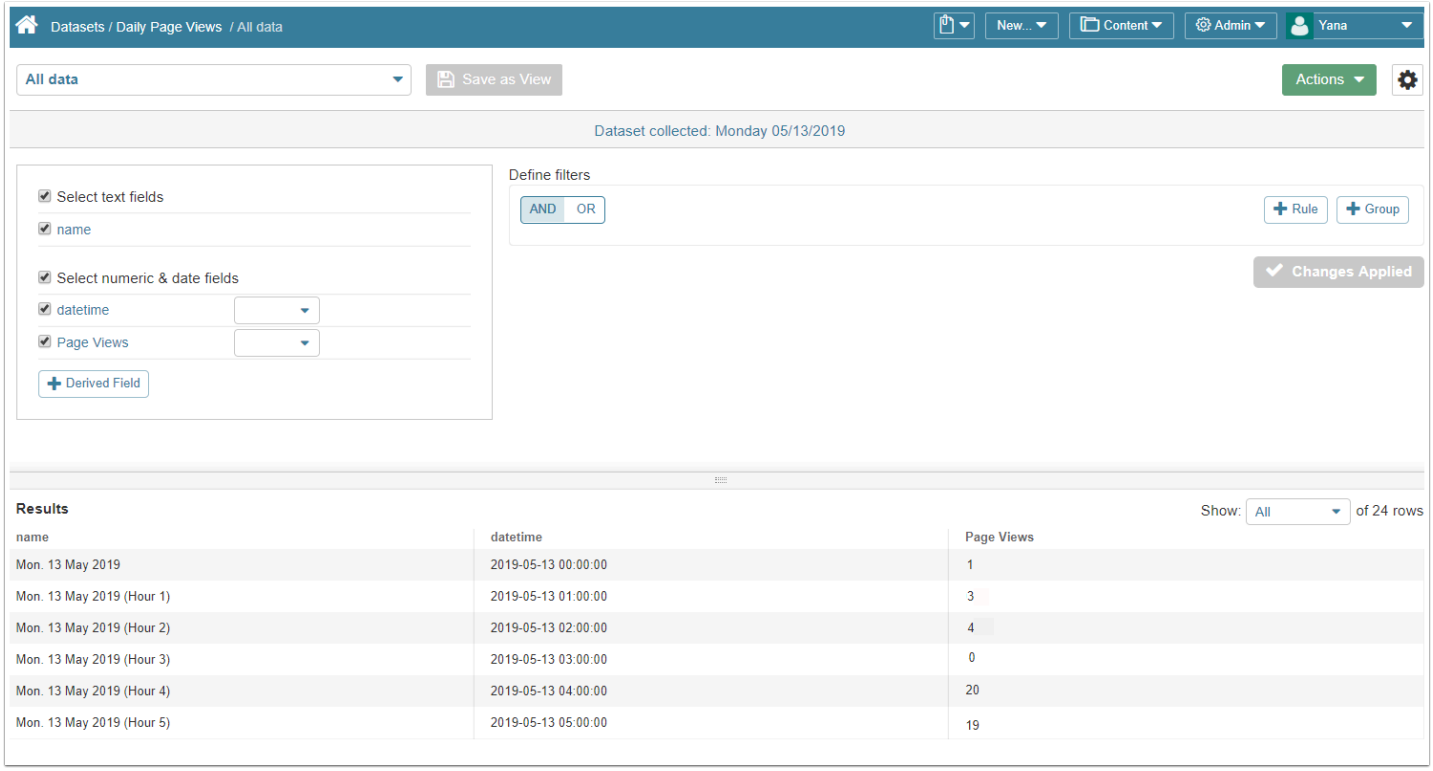
1. **Data Source**: select the connection profile you have created for *Adobe Analytics*
2. **Data collection trigger**: specify the Trigger that will be used to collect data for your Dataset
3. **Adobe Analytics Report:** select an external *Adobe Analytics* Report that should serve as a basis of your Dataset
4. Input a **Plugin Command** listing all the data you would like to fetch from *Adobe Analytics*
5. Once you are ready with you command, click **Validate**

4. Plugin command will be validated and data collected on Save



1. If the command is validated successfully, the **Dataset columns** and **Data Preview** are going to be shown below.
2. At the upper right corner of the screen click **Enable & View**.

5. Dataset will be displayed in Viewer

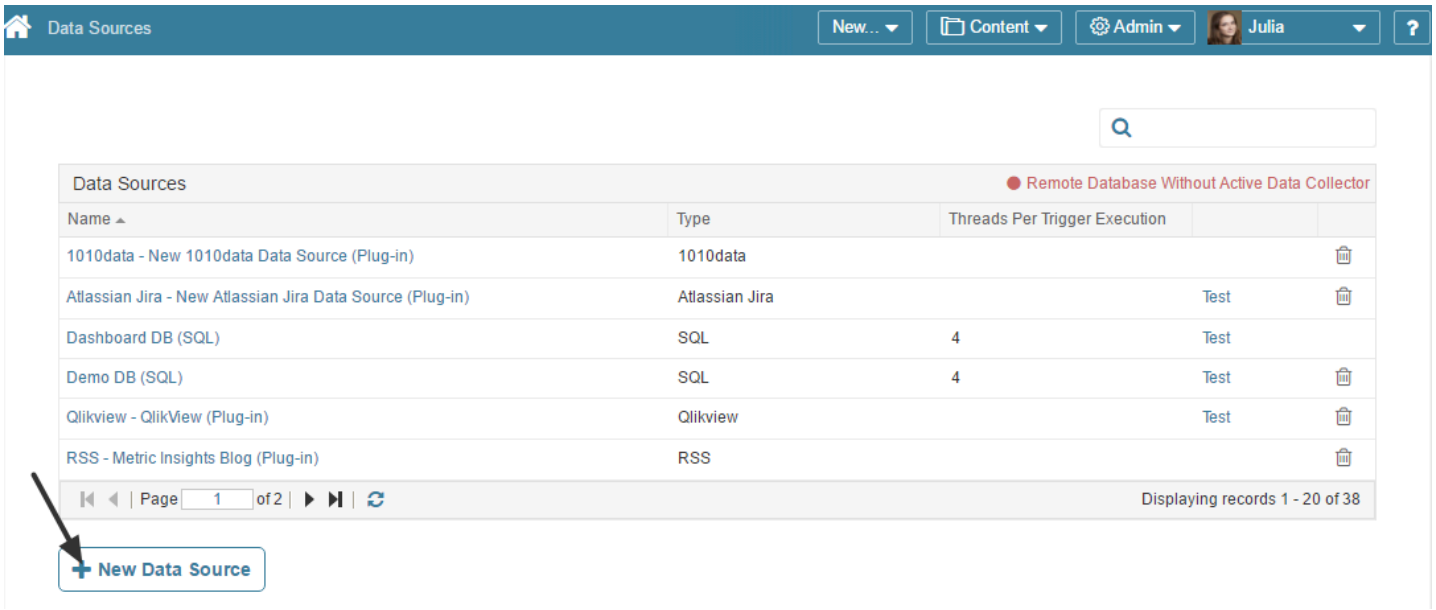


7. Sourcing Data from Adaptive Planning

7.1 Establish connectivity to Adaptive Planning

An Administrator can use the process described in this article to create a new **Plug-in Data Source** that is required to allow Elements to fetch data from **Adaptive Planning** to create a visualization in Metric Insights.

1. Access Admin > Data Sources



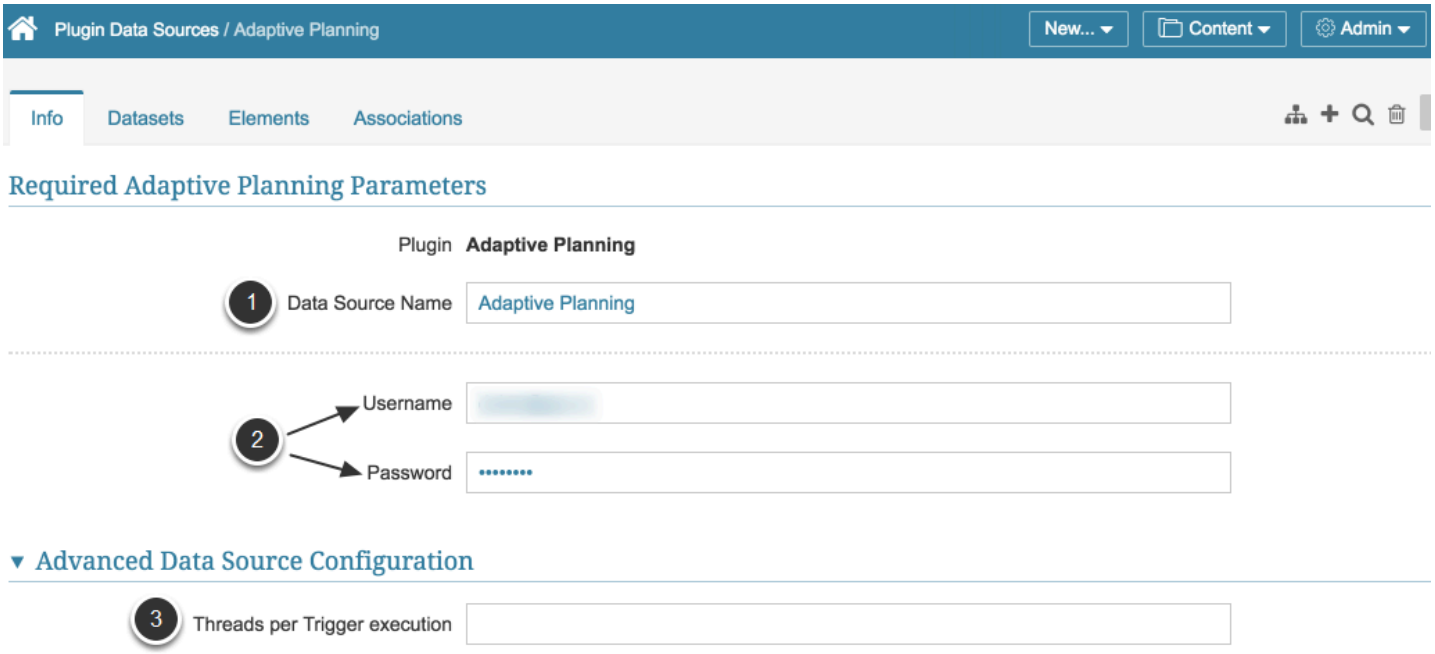
At the bottom of the screen click **[+ New Data Source].**

The *Select the Type of New Data Source* pop-up opens.

2. Select "Other" Data Source Type and choose "Adaptive Planning" from the drop-down list

Move to the **Next step**.

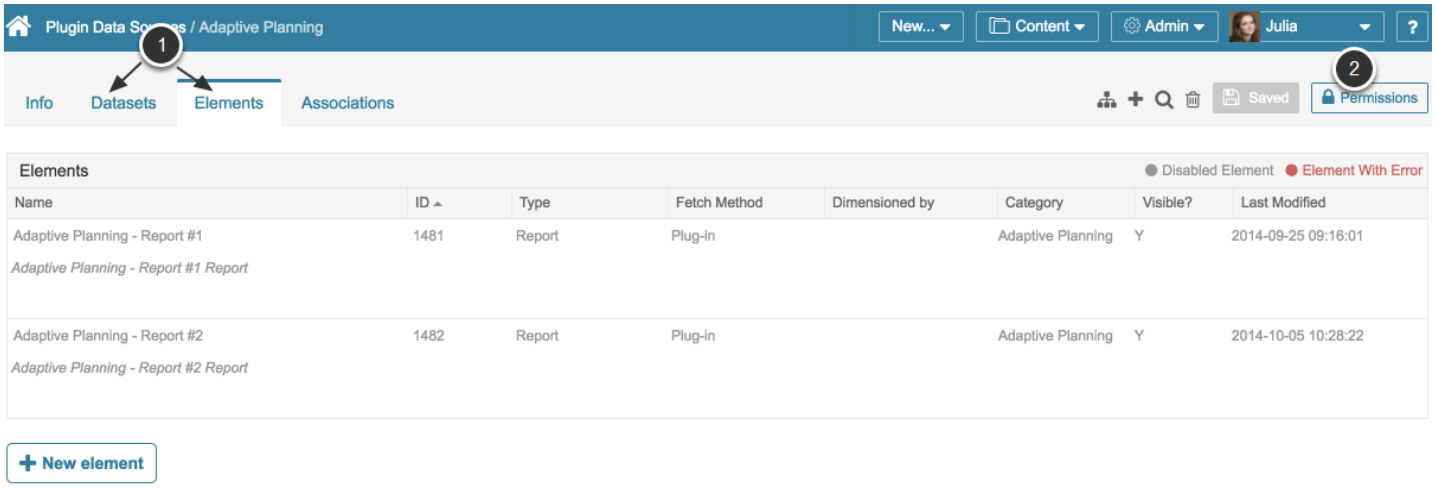
3. Provide Required Adaptive Planning Parameters



Specify how to connect to Adaptive Planning. The parameters include:

1. **Data Source Name:** Will default but you may modify it.
2. **Username / Password:** Note that your **Username** must be in the same format that your Adaptive Planning server uses for authentication
3. Optionally, specify the maximum number of concurrent **Threads per Trigger execution** to be used in background processing when the system updates Metrics and Reports for this Data Source. If you do not specify any value for this setting, batch data collection processing will be single-threaded.
4. **Save** your entries

4. Other Settings



1. You can create elements or Datasets directly from the respective tabs
2. Click **Permissions** to assign them to Groups or Power Users

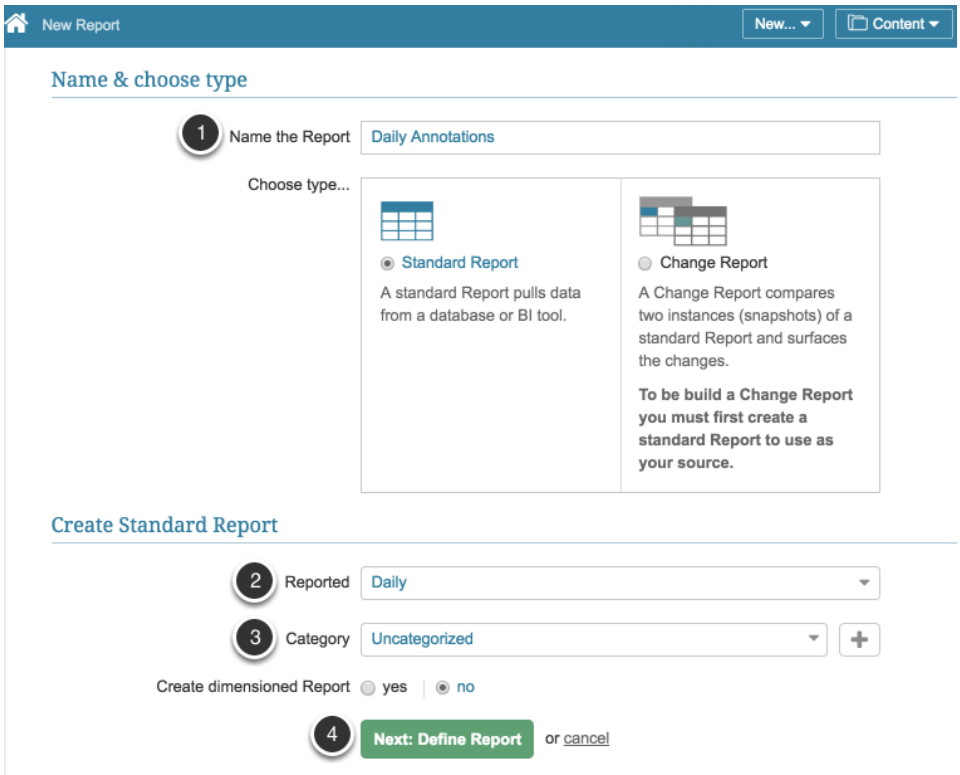
5. What's next?

[How to collect data from Adaptive Planning?]

7.2 Collect data from Adaptive Planning

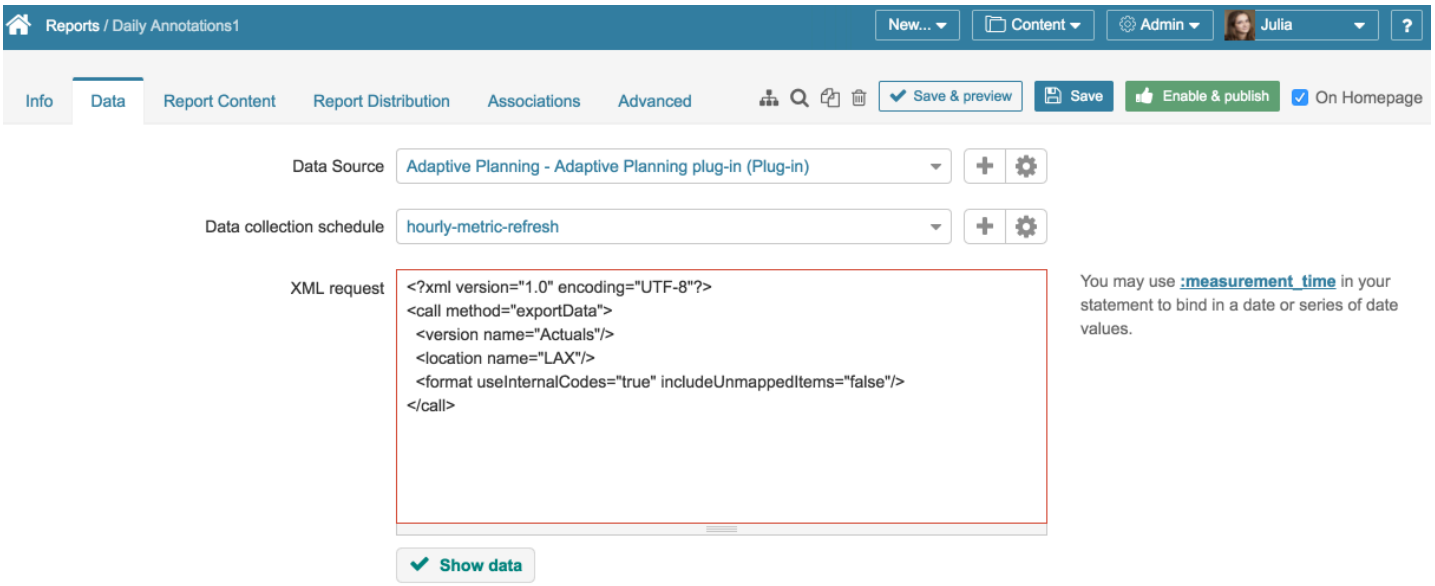
This article will show you how to create an Element using a Adaptive Planning plug-in as a data source. It assumes that you have already established connectivity to your Adaptive Planning account.

1. Access New > Report



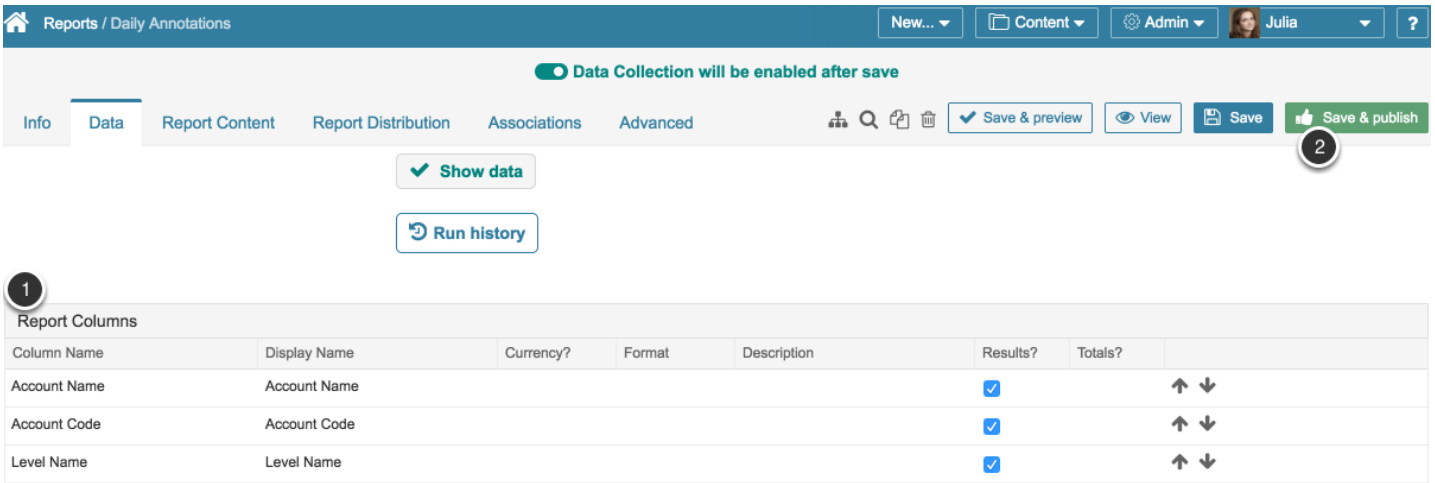
1. **Name the Report**: Define a unique descriptive name of your element
2. **Reported**: choose the measurement interval from the drop-down list
3. **Category**: define a category this element belongs to
4. To move on to defining data collection details, click **Next: Define Report**

2. Full Editor displays the Data Collection tab



1. **Data Source**: select the account you have created for **Adaptive Planning**
2. **Data Collection Schedule**: Specify the trigger that will be used to collect the data for your report
3. Input **XML request** listing all the data you would like to fetch from Adaptive Planning server
4. Once you are ready with you command, click **Show Data**.

3. Resulting Report columns



1. If your request syntax is valid, the statement box is **green**; if there are any errors, the box is colored in **red** and errors will be explained in the field below. If the request is validated successfully, the **Report Columns** table is to be displayed below the statement box.
2. **Enable and Publish** your Report.

8. Sourcing Data from Atlassian Confluence

8.1 Establish connectivity to Atlassian Confluence

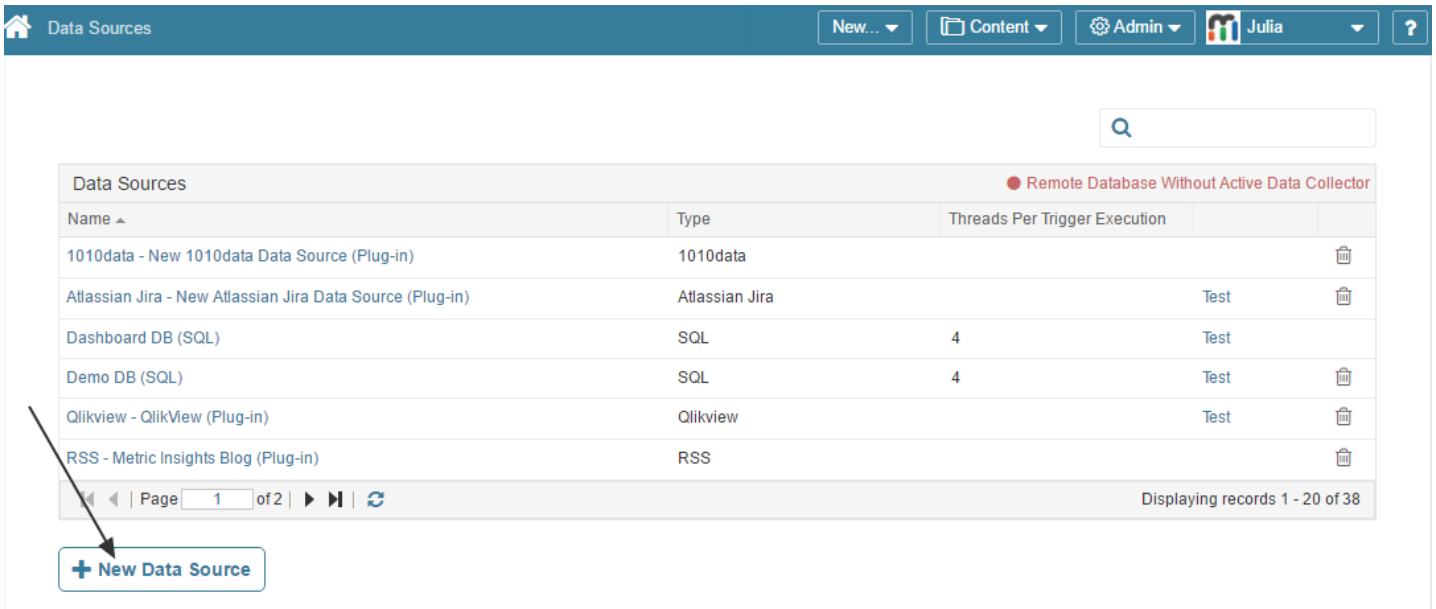
An Administrator can use the process described in this article to create a new **Plug-in Data Source** that is required to allow elements to pull data data from **Atlassian Confluence**.

Plugin Capabilities:

1. This plugin scans all existing articles in Confluence and extracts those, which have at least one table included. Such tables are treated as source Reports (See step 4).
2. Data from Confluence tables is used as a source for Reports or Datasets

[Release 6.x]: 'Remote Data Collector' renamed to 'Remote Data Processor'

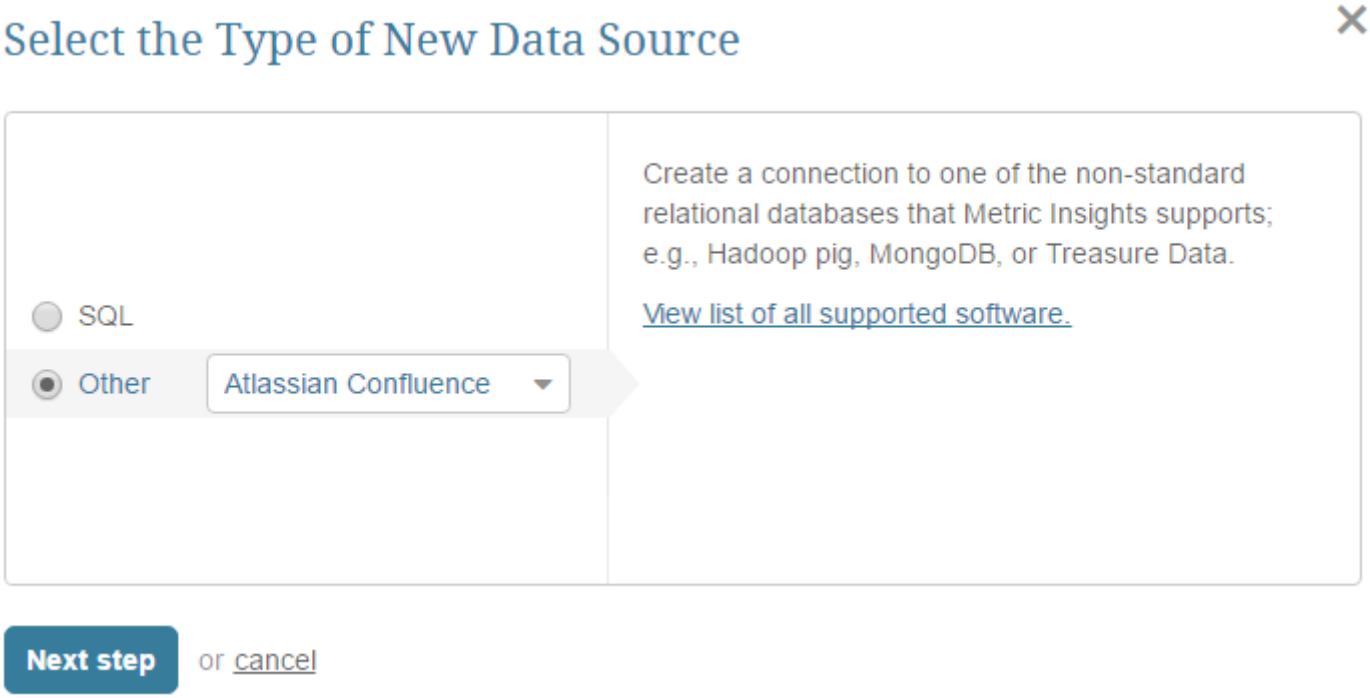
1. Access Admin > Data Sources



At the bottom of the screen click **[+ New Data Source].**

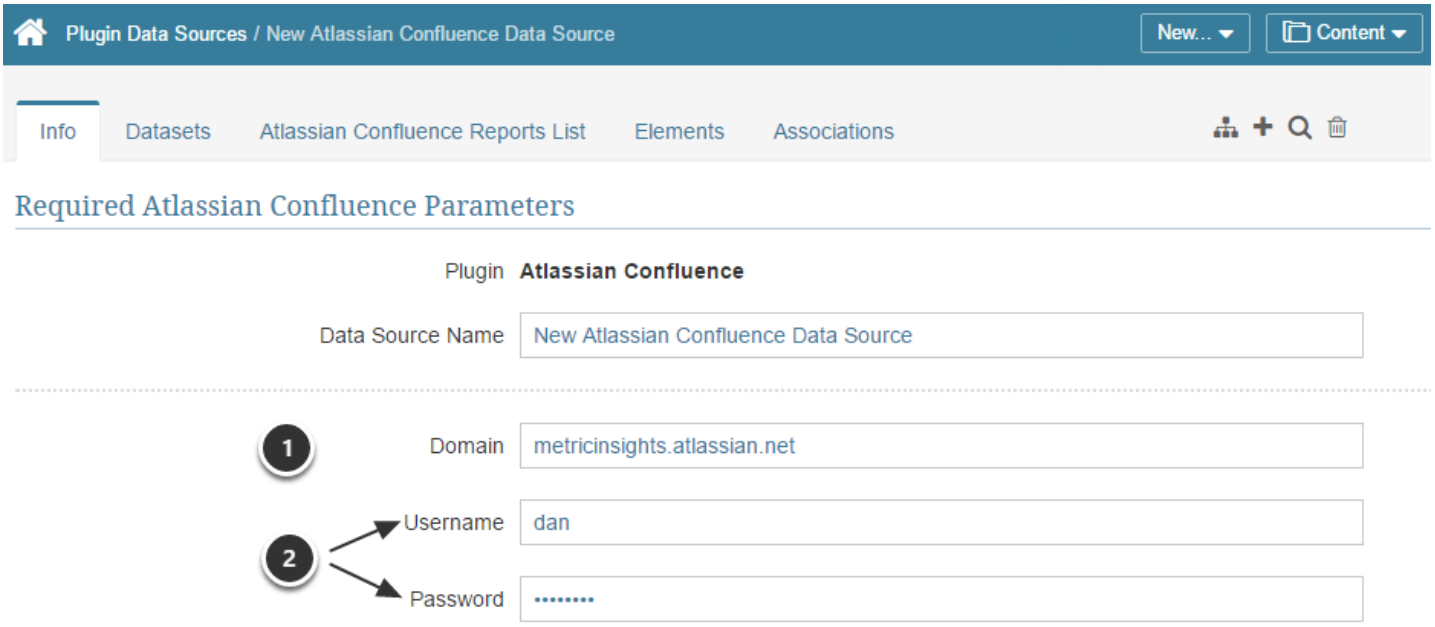
The *Select the Type of New Data Source* pop-up opens.

2. Select "Other" Data Source Type and choose "Atlassian Confluence" from the drop-down list



Move to the **Next step**.

3. Provide Required Adaptive Planning Parameters

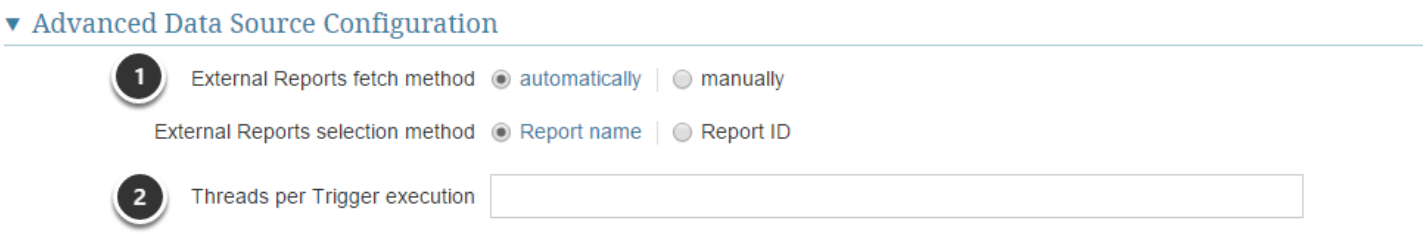


1. **Domain**: provide the Atlassian domain you want to pull data from in the form of:

```
<companyname>.atlassian.net
```

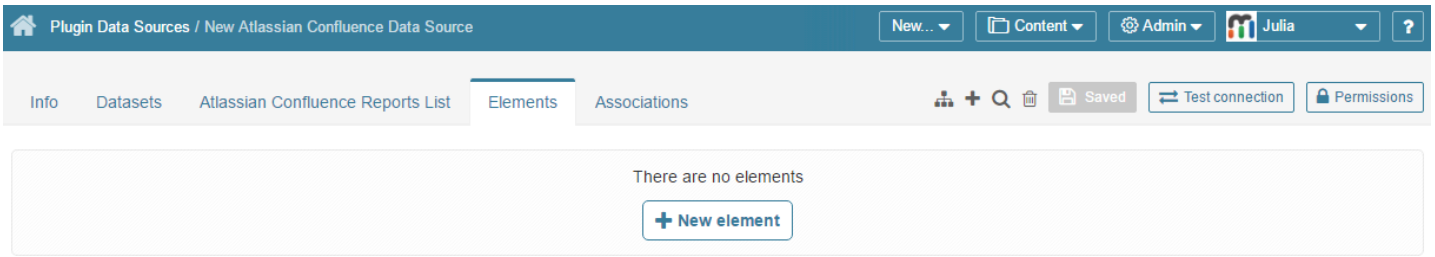
2. **Username / Password**: Only admin user credentials can be used to pull data from Confluence.

4. Advanced Configuration



1. **External Reports fetch method**: This setting influences options available in the *Atlassian Confluence Report List* tab:
   - **automatically**: just click Refresh list and all Reports are going to be fetched by the system
   - **manually**: Reports may be added one-by-one or via CSV file.
2. Optionally, specify the maximum number of concurrent **Threads per Trigger execution** to be used in background processing when the system updates Metrics and Reports for this Data Source. If you do not specify any value for this setting, batch data collection processing will be single-threaded.

5. Other Settings



1. You can create elements or Datasets directly from the respective tabs
2. Click **Permissions** to assign them to Groups or Power Users

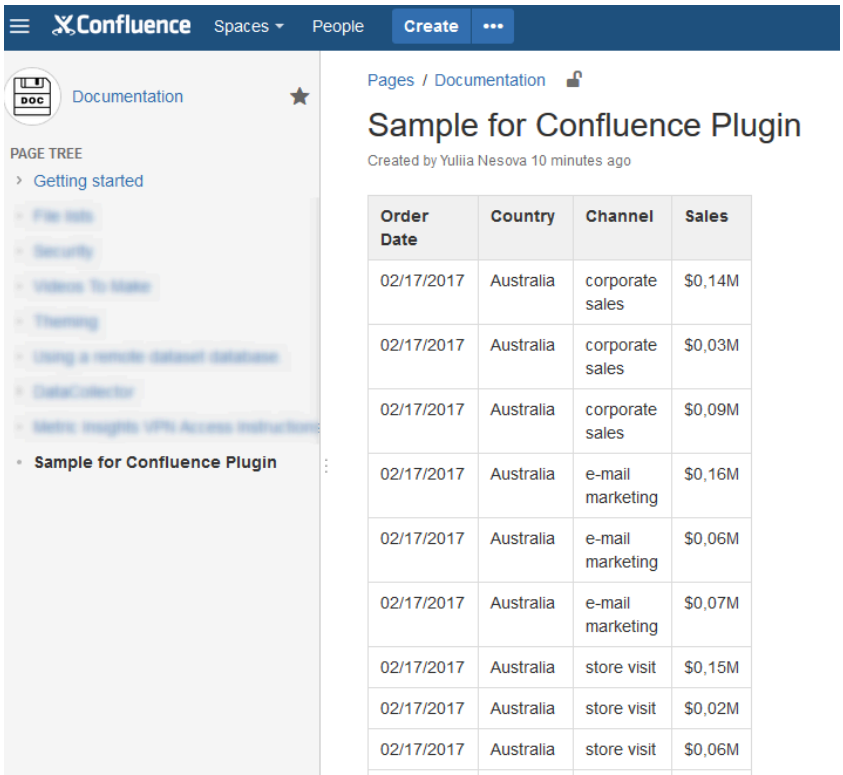
What's next?

[How to Collect Data from Atlassian Confluence]

8.2 Collect data from Atlassian Confluence

This article will show you how to create an Element using a Atlassian Confluence plugin as a data source. It assumes that you have already established connectivity to your Atlassian Confluence account.

Use Case

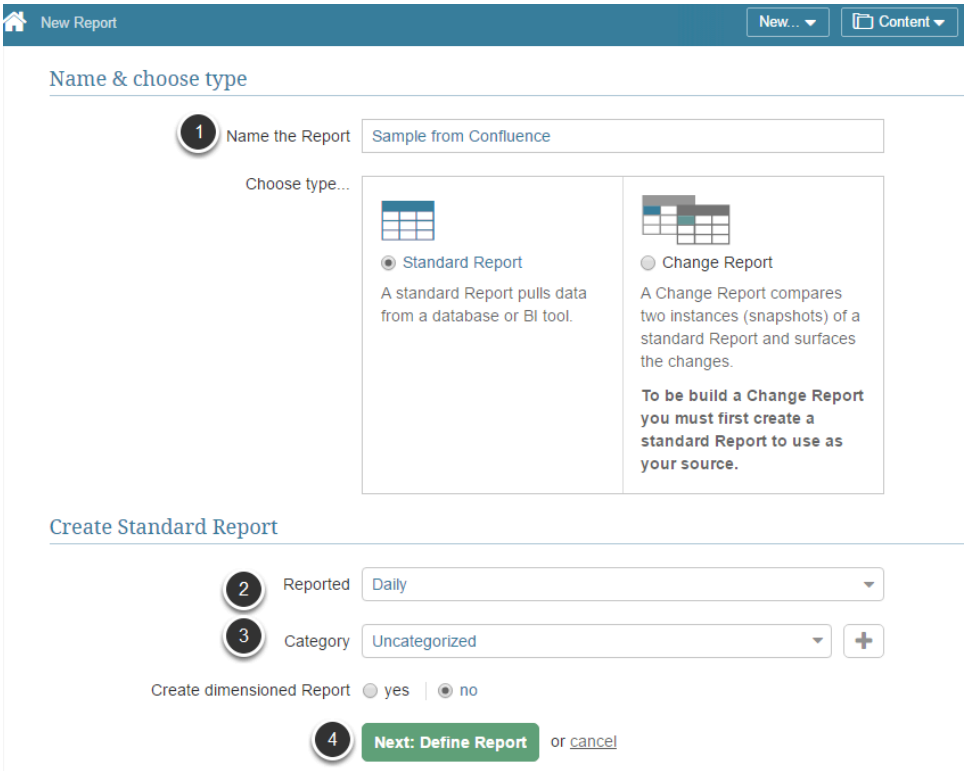


Order Date	Country	Channel	Sales
02/17/2017	Australia	corporate sales	$0,14M
02/17/2017	Australia	corporate sales	$0,03M
02/17/2017	Australia	corporate sales	$0,09M
02/17/2017	Australia	e-mail marketing	$0,16M
02/17/2017	Australia	e-mail marketing	$0,06M
02/17/2017	Australia	e-mail marketing	$0,07M
02/17/2017	Australia	store visit	$0,15M
02/17/2017	Australia	store visit	$0,02M
02/17/2017	Australia	store visit	$0,06M

Atlassian Confluence plugin scans all existing articles in Confluence and extracts only those, which have at least one table included. Such tables are treated as source Reports.

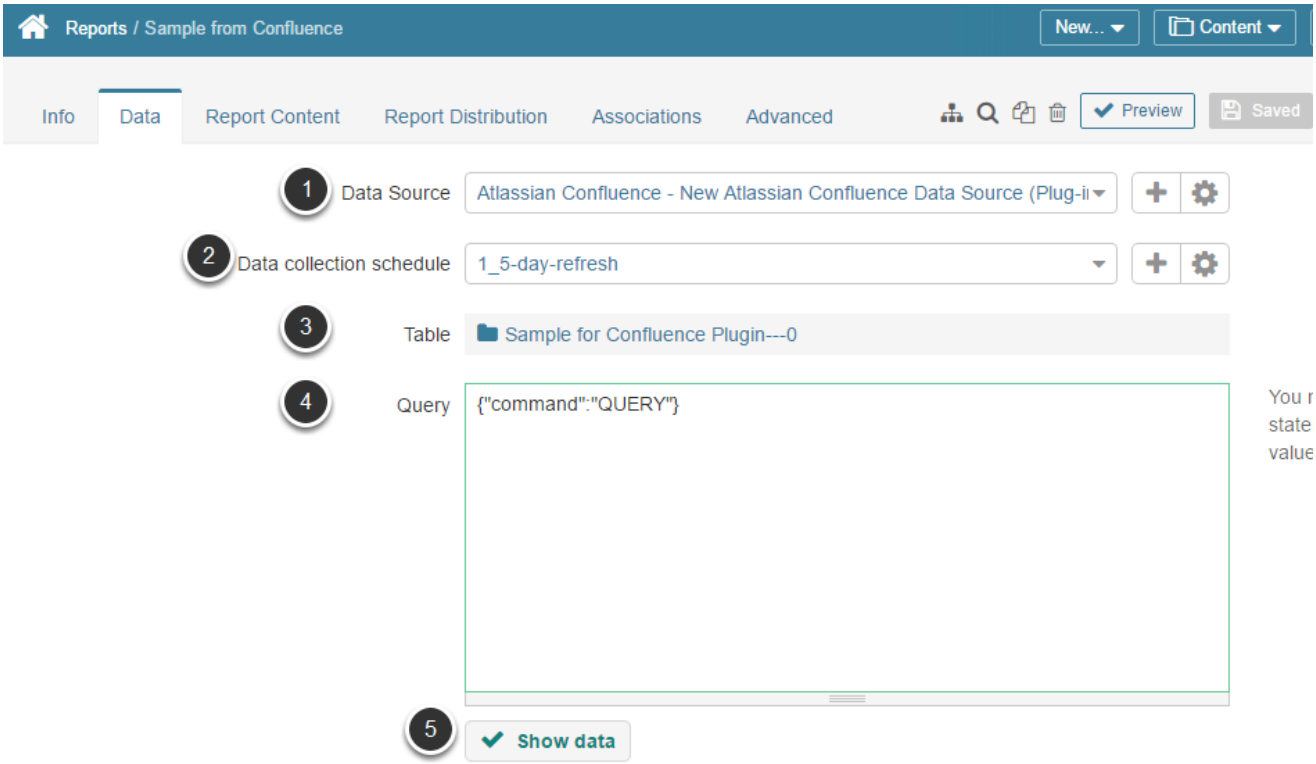
In the given example the table on this Confluence page may serve as a source for a Report in Metric Insights.

1. Access New > Report



1. **Name the Report:** Define a unique descriptive name of your element
2. **Reported:** choose the measurement interval from the drop-down list
3. **Category:** define a category this element belongs to
4. To move on to defining data collection details, click **Next: Define Report**

2. Full Editor displays the Data Collection tab



1. **Data Source:** select the account you have created for Atlassian Confluence
2. **Data Collection Schedule:** Specify the trigger that will be used to collect the data for your report
3. **Table**: Choose the name of the table which should serve as a basis of this Report
4. **Query**: the standard query for pulling all data fro the table is:

```
{"command":"QUERY"}
```

5. Once you are ready with you command, click **Show Data**.

3. Resulting Report columns

Query {"command":"QUERY"}

You may use :measurement_time in your statement to bind in a date or series of date values.

Show data

Sample result set

Order Date	Country	Channel	Sales
2017-02-17 00:00:00	Australia	corporate sales	$0,14M
2017-02-17 00:00:00	Australia	corporate sales	$0,03M
2017-02-17 00:00:00	Australia	corporate sales	$0,09M

If your request syntax is valid, the statement box is **green**; if there are any errors, the box is colored in **red** and errors will be explained in the field below. If the request is validated successfully, the Report Columns table is to be displayed below the statement box.

Enable and Publish your Report.

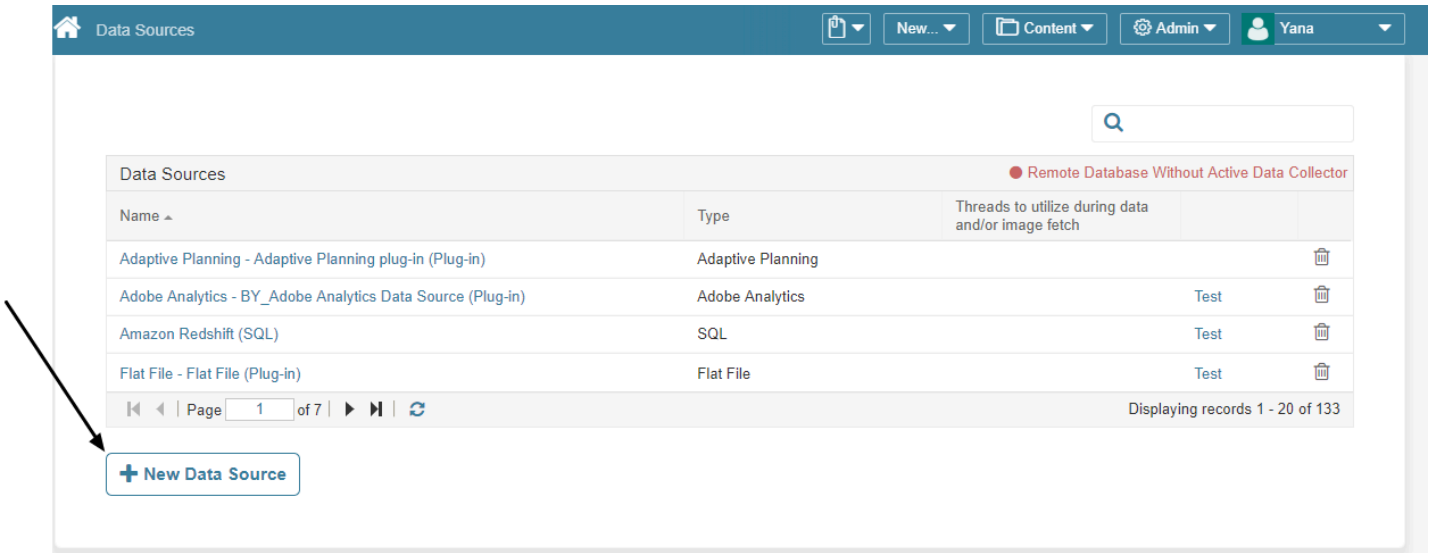
9. Sourcing Data using Atlassian Jira

9.1 Establish Connectivity to Atlassian Jira

This article describes how to connect to **Atlassian Jira** in order to load data into Datasets and Reports in Metric Insights.

[Release 6.x]: 'Remote Data Collector' renamed to 'Remote Data Processor'

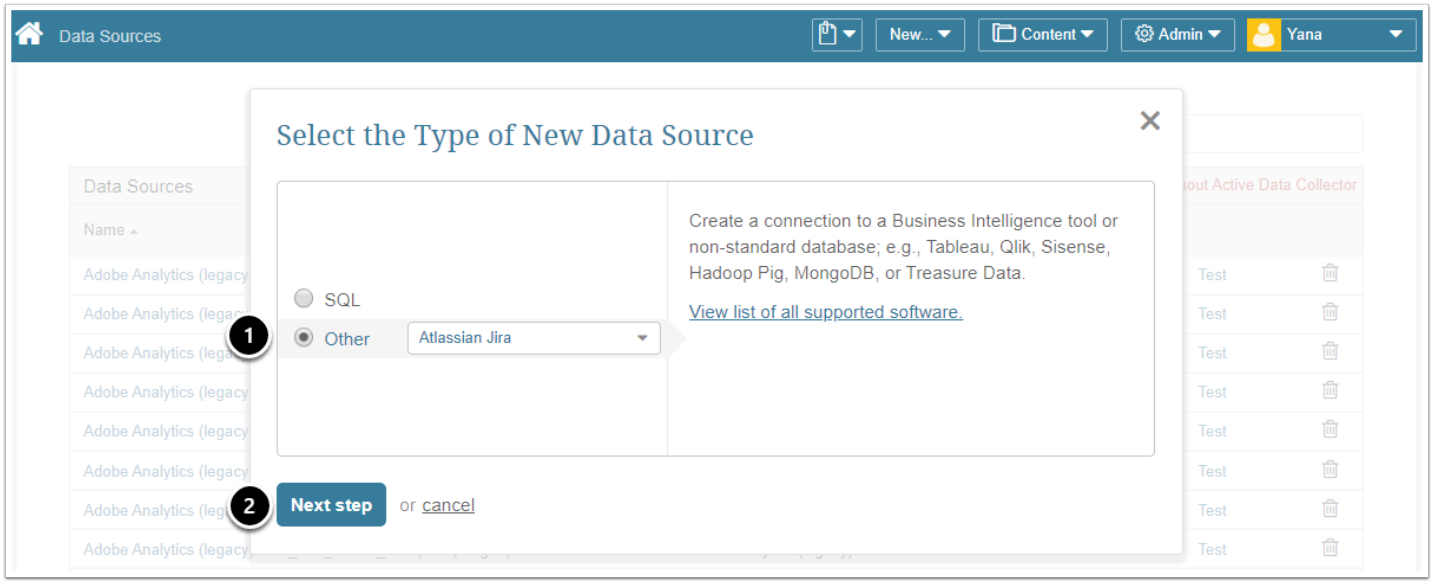
1. Access Admin > Data Sources



At the bottom of the screen click **[+ New Data Source].**

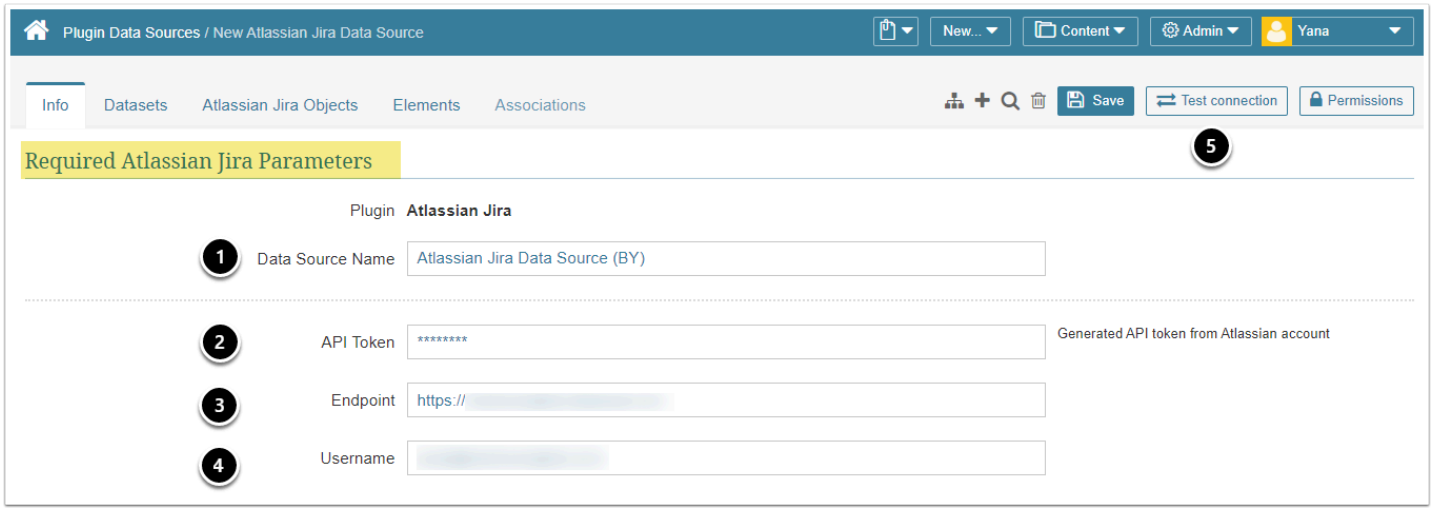
The *Select the Type of New Data Source* pop-up opens..

2. Select the Type of the New Data Source



1. Select "Other" and choose **"Atlassian Jira"** from the drop-down list
2. Move to the **Next step**

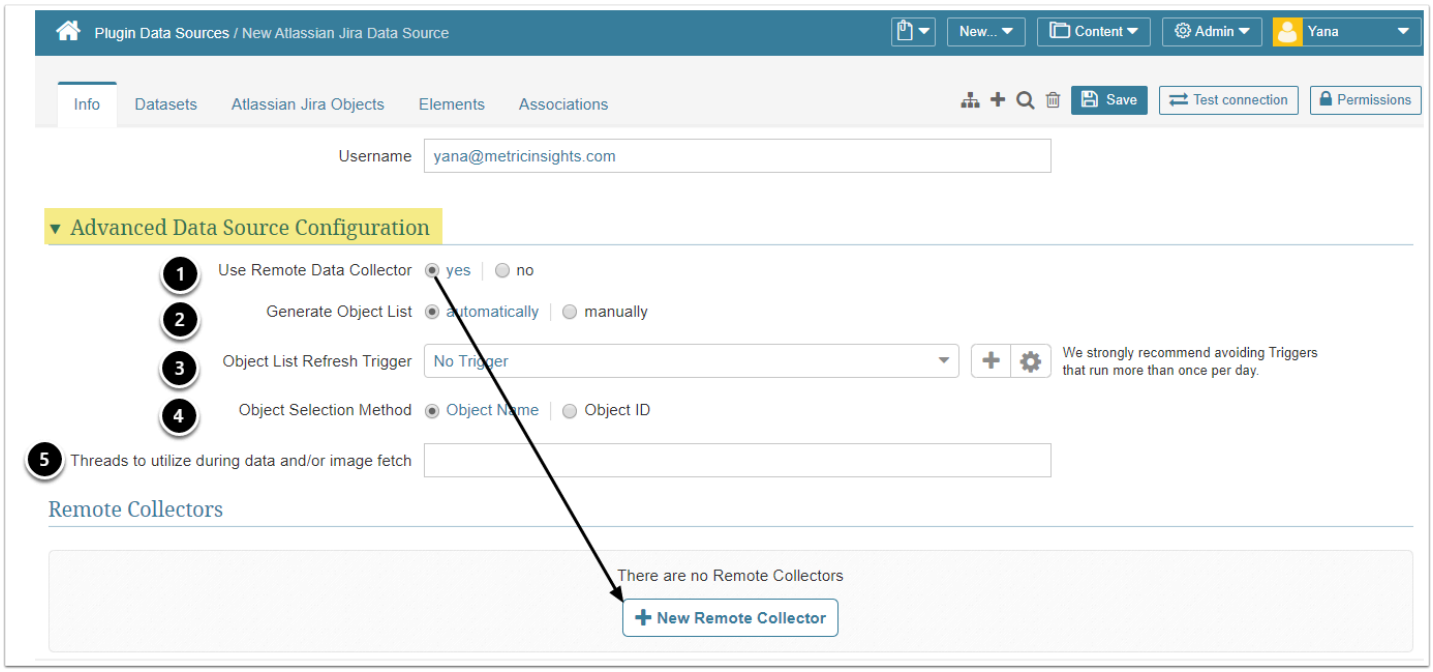
3. Provide the Required Parameters



1. **Source Name** is generated automatically, but you may modify it as desired
2. Enter the **API Token** in the provided field
   - The API Token can be generated from your Atlassian Account https://id.atlassian.com/manage/api-tokens
   - For more information, refer to https://confluence.atlassian.com/cloud/api-tokens-938839638.html
3. Specify the **Endpoint** for connection

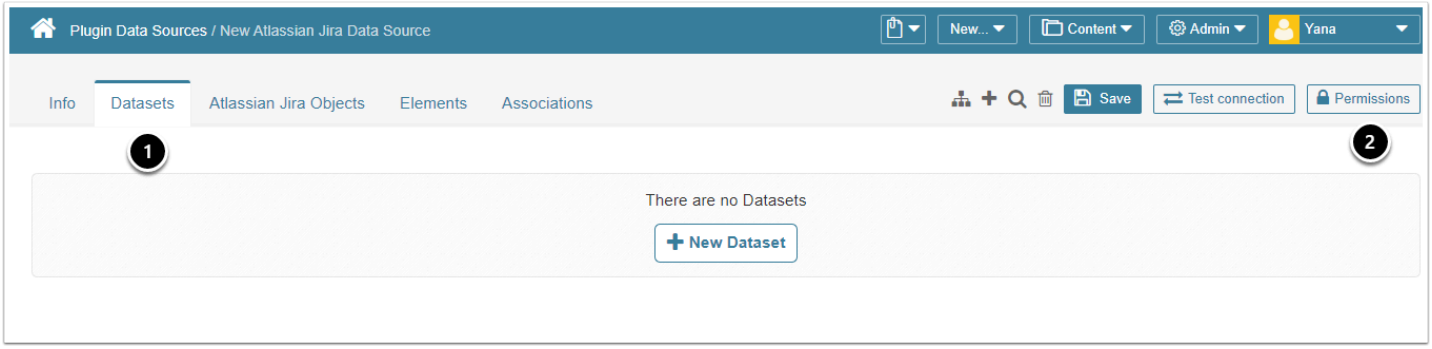
4. Provide credentials to the Jira account
5. **Test Connection** (this will also **Save** your data)

4. Advanced Configuration



1. **Use Remote Data Collector:** is set to "no" by default
   - If required, switch to "yes" and add a Remote Data Collector by clicking **[+New Remote Collector]**
2. **Generate Object List**
   - *automatically*: all Reports are going to be fetched by the system
   - *manually*: Reports may be added one-by-one or via CSV file
3. **Object List Refresh Trigger:** from the dropdown, select the Trigger that will be used to fetch data via the Jira plugin
4. **Object Selection Method:** specify how Jira Reports will be fetched
5. Optionally, state the maximum number of concurrent **Threads to utilize during data and/or image fetch** to be used in background processing when the system updates Reports for this Data Source
   - If you do not specify any value for this setting, batch data collection processing will be single-threaded

5. Other settings



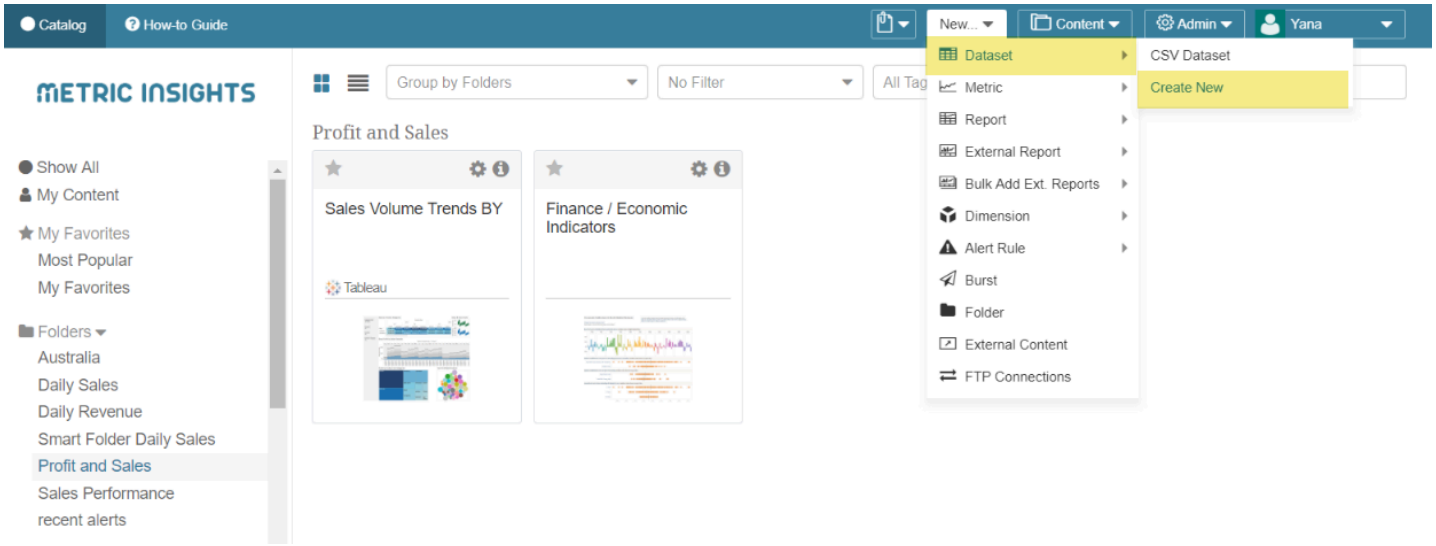
1. You can create **Datasets** directly from the respective tab
2. Click **Permissions** to assign permissions to Groups or Power Users

9.2 Collect data from Atlassian Jira

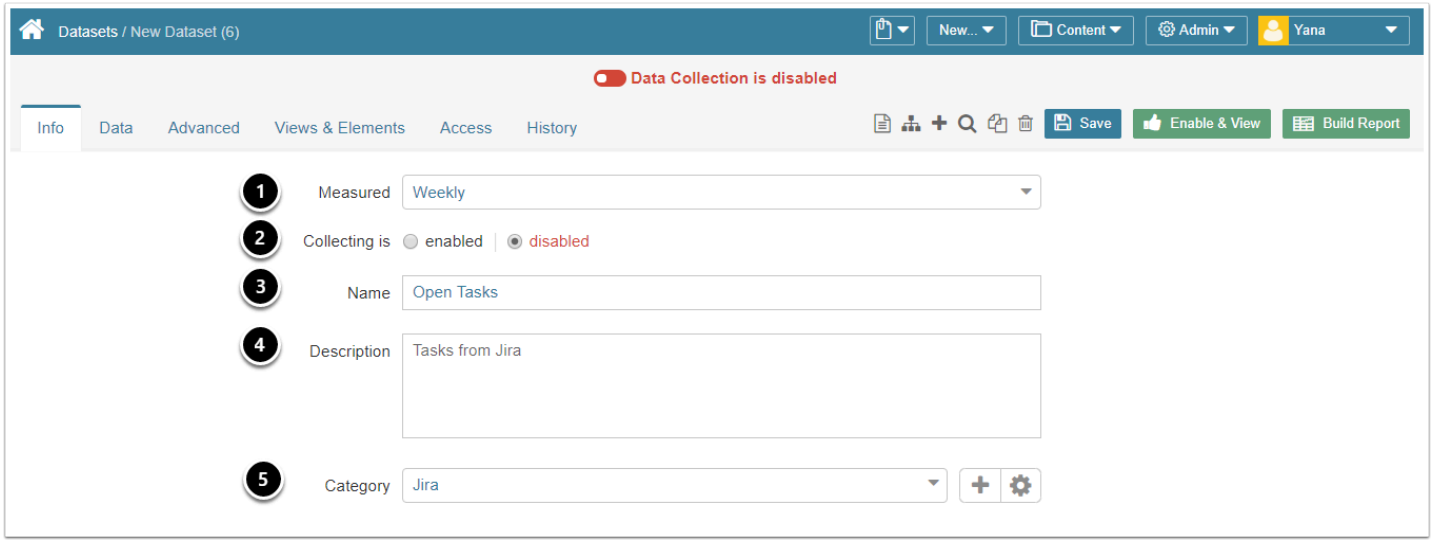
This article details how to create a Dataset populated with data sourced from **Atlassian Jira**.

It assumes that you have already established connectivity to Atlassian Jira.

1. Access New > Dataset > Create New



2. Dataset Editor > Info tab



Define the basics:

1. **Measured:** select the measurement interval that applies to the level of aggregation that you want in your result set
2. **Collecting**: new Datasets are always disabled by default to make sure that you can take time to configure them properly before enabling. This setting is duplicated at the top of the screen
3. **Name:** provide a unique name for your Dataset. Preferably, the Dataset name should explain what kind of data it contains
4. **Description:** optionally, provide any additional information about your Dataset
5. **Category:** specify the Category where you Dataset will be placed

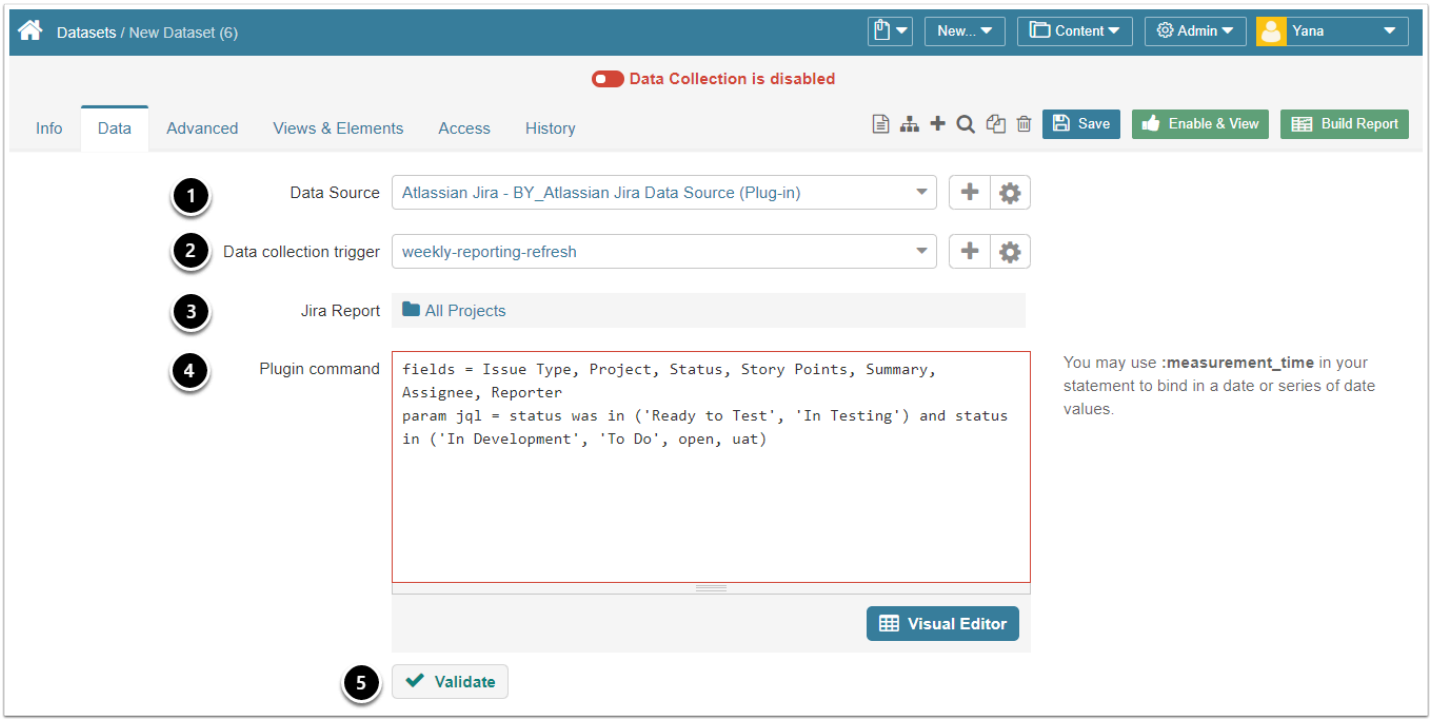
Move to the ***Data tab*** to define the source of data and how often it should be updated.

3. Define the Settings for Data Collection

3.1. Using MIQL Plugin Command

For details on building fetch commands, refer to:

- Fetch Commands for JIRA
- MIQL Syntax Guide for Plugins



1. **Data Source**: select the connection profile you have created for *Atlassian Jira*
2. **Data collection trigger**: specify the Trigger that will be used to collect data for your Dataset
3. **Element:** select an *Atlassian Jira Object* that should serve as a basis of your Dataset

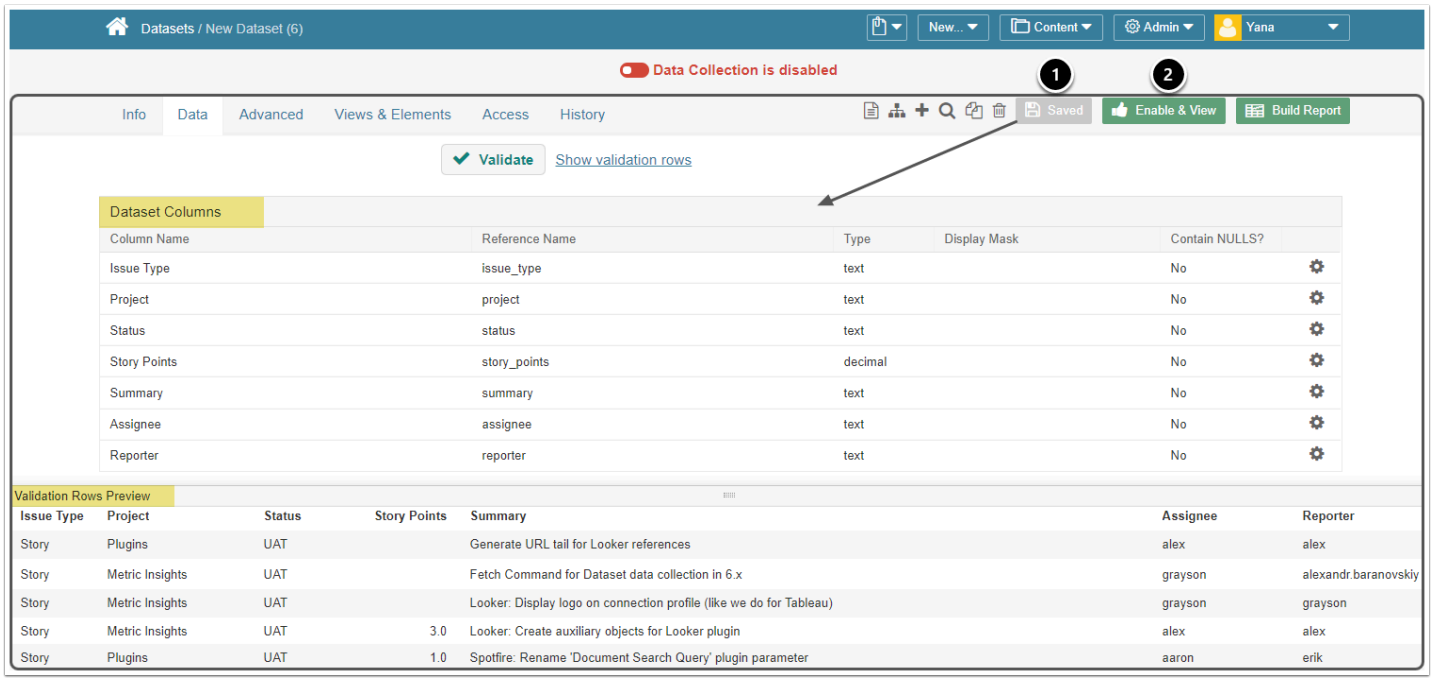
4. In the **Command tab**, input an MIQL **Plugin Command** listing all data that needs to be fetched from *Oracle Business Intelligence*
5. **Validate** your query

Note!

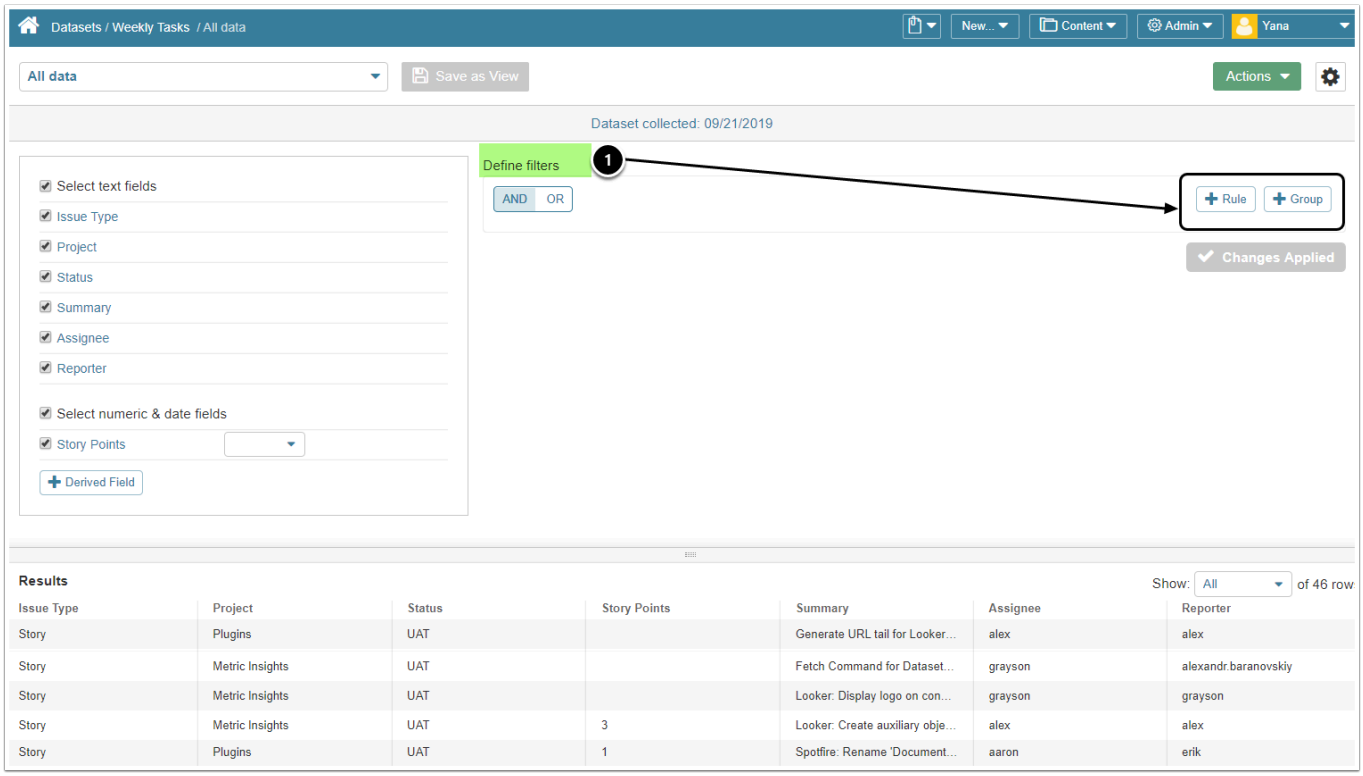
1. Entire field names that contain special characters, aggregation and commas must be enclosed in quotes (single or double).
2. It is acceptable to enclose all fields and values in quotes.

   [...] + Notation is used to signify that the MIQL parts of a statement are optional/can be repeated.

4. Plugin command will be validated and data collected on Save



5. Dataset will be displayed in Viewer



In the **Dataset Viewer**:

1. You can further refine your data with Dataset's *internal filtering options* by **applying Rules** and **Grouping Data**:
   - For more information on using Internal Filters, refer to Create a Dataset View
   - For general instruction on building Datasets, see Create a Dataset from any Data Source

9.3 Fetch Commands for JIRA

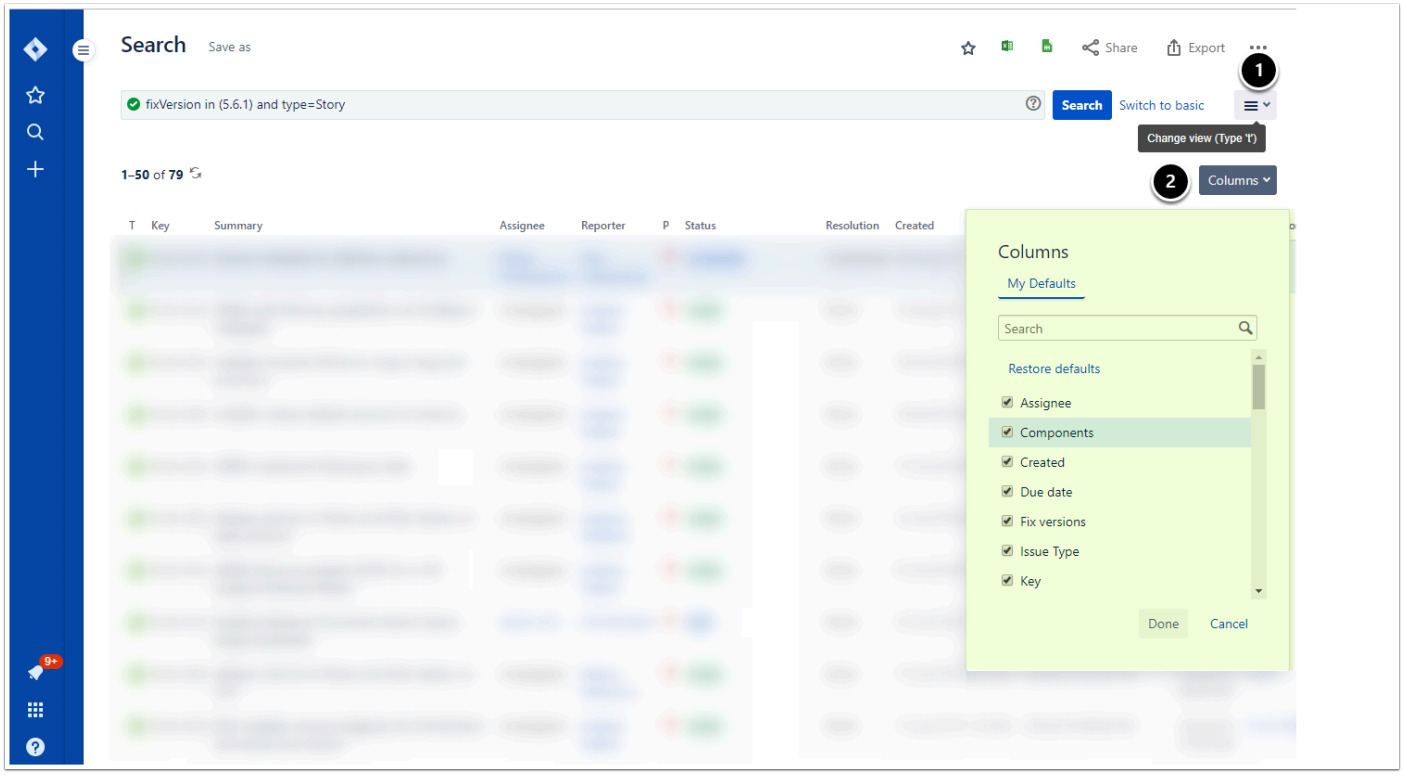
The article details how to run fetch commands to query data from Atlassian Jira. This data can then be used to build Elements/Datasets in Metric Insights.

1. Determine the fields to be fetched from Jira

You can find a list of available fields:

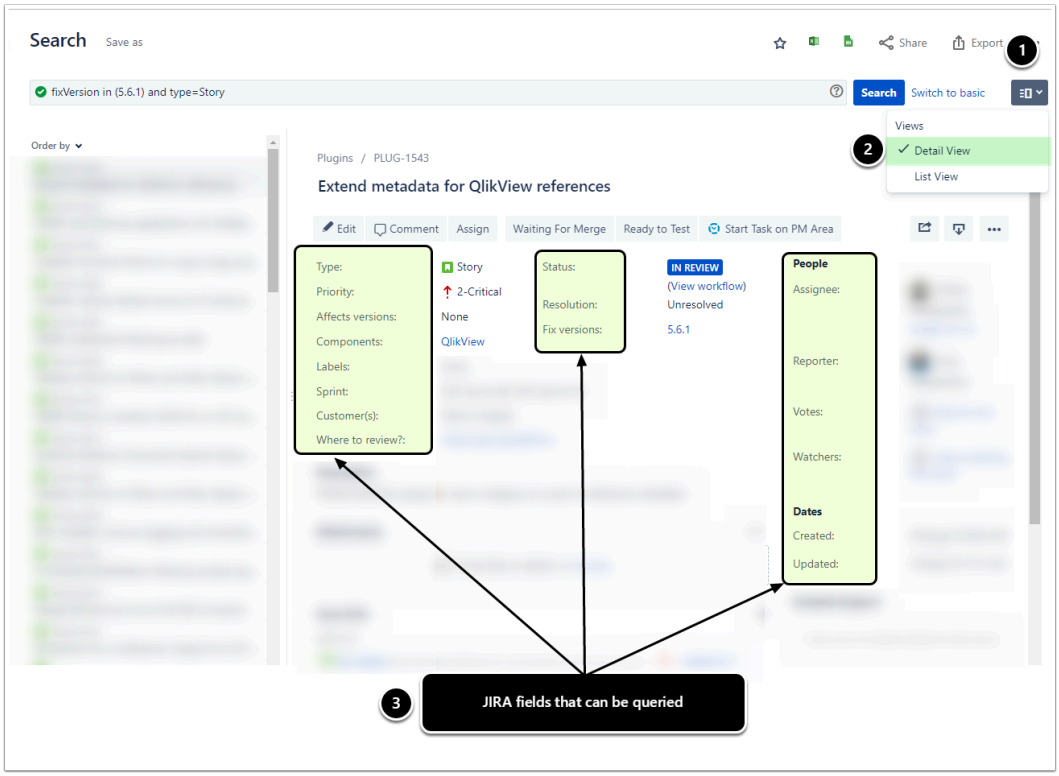
1. In List View
2. In Detail View
3. When adding a new Issue

1.1. OPTION 1: Access the List View



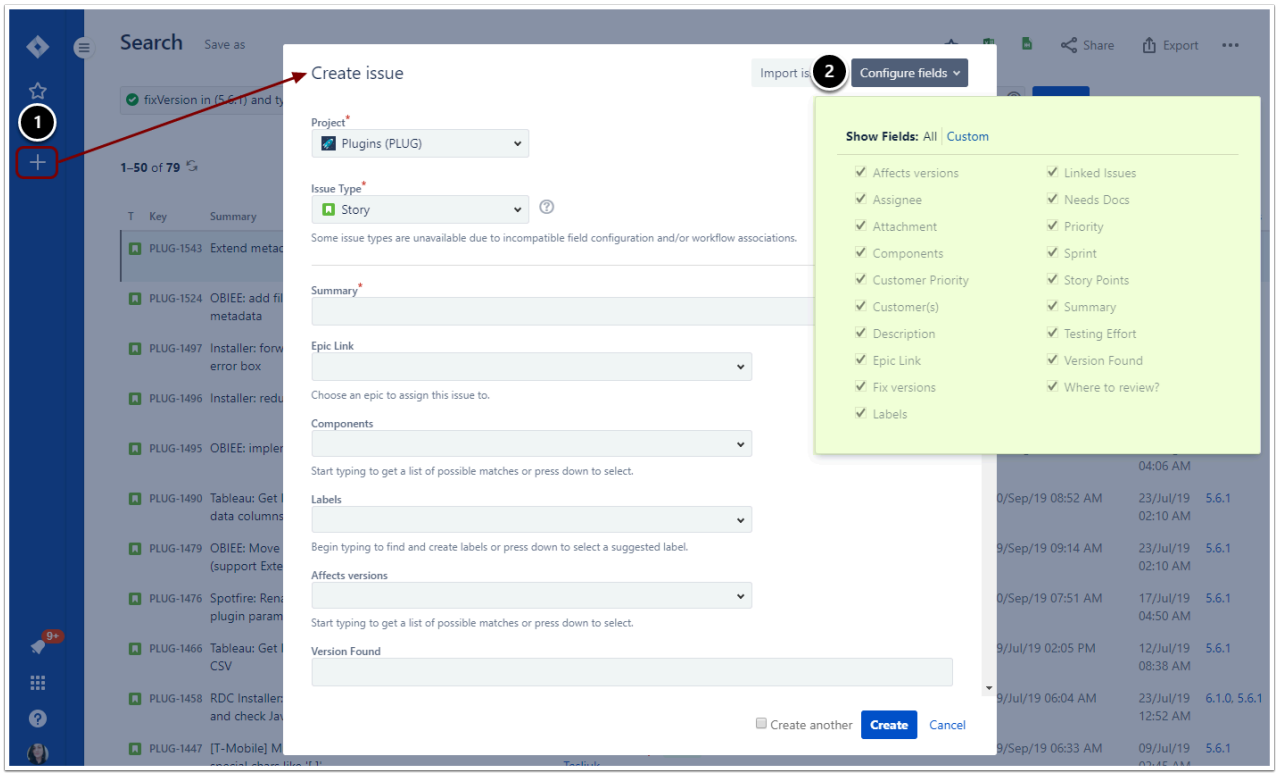
1. Click the **Views icon** (3 vertical lines) to switch to the **List View**
2. Open the **Columns** dropdown for a list of available fields

1.2. OPTION 2: Access the Detail View



1. Click the **Views icon** (3 vertical lines)
2. Select **Detail View**
3. Choose the required **fields**

1.3. OPTION 3: Create Issue > Configure fields



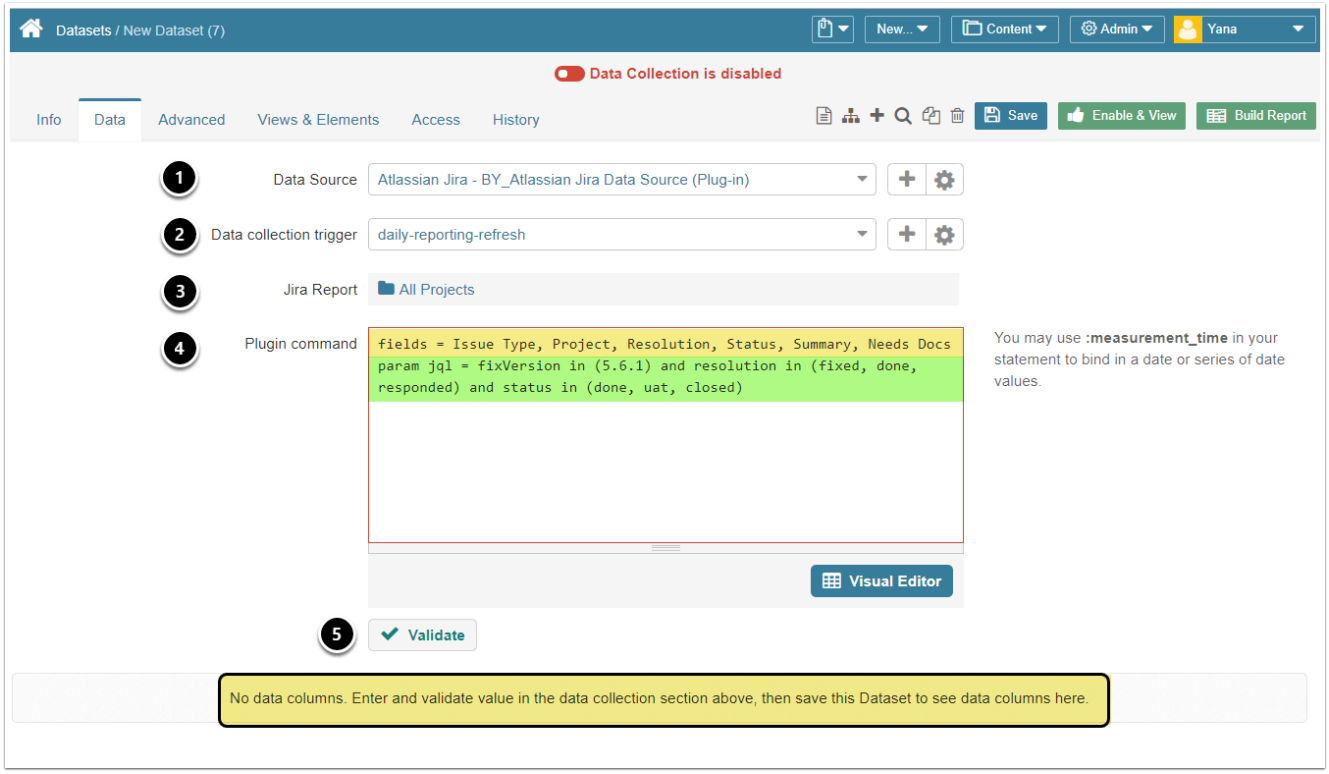
1. Click **[Create issue]** to open the new issue pop-up
2. Open the **Configure fields** dropdown to see the list of available fields

2. Use MIQL to fetch data from Jira (5.3.+)

The process below exemplifies how an MIQL command is used to fetch data into a Metric Insights Dataset. Creating an Element (Metric or Report) with data sourced from Jira should use the same workflow.

1. MIQL (**M**etric **I**nsights **Q**uery **L**anguage) is a simple query language designed for fetching and processing data.
2. Learn more from [MIQL Syntax Guide for Plugins]

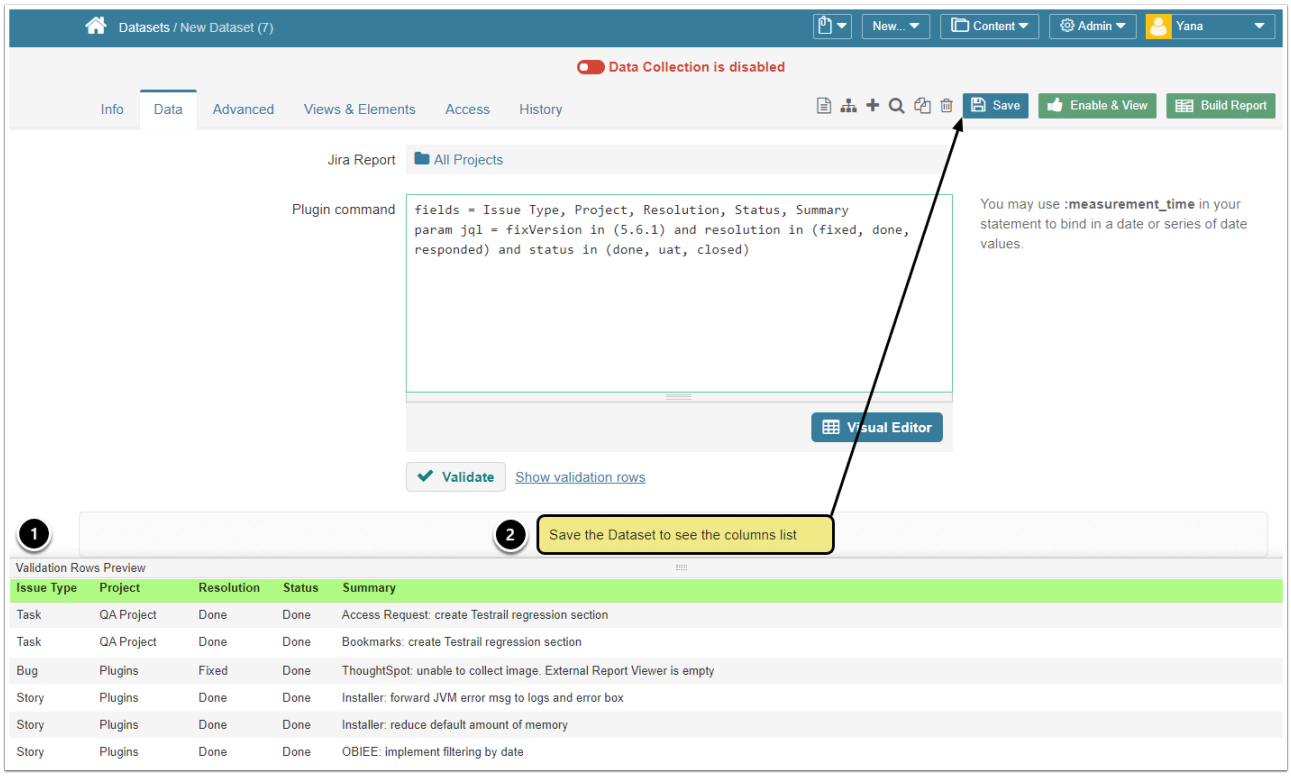
2.1. Dataset Editor > Data tab



In the Dataset Editor:

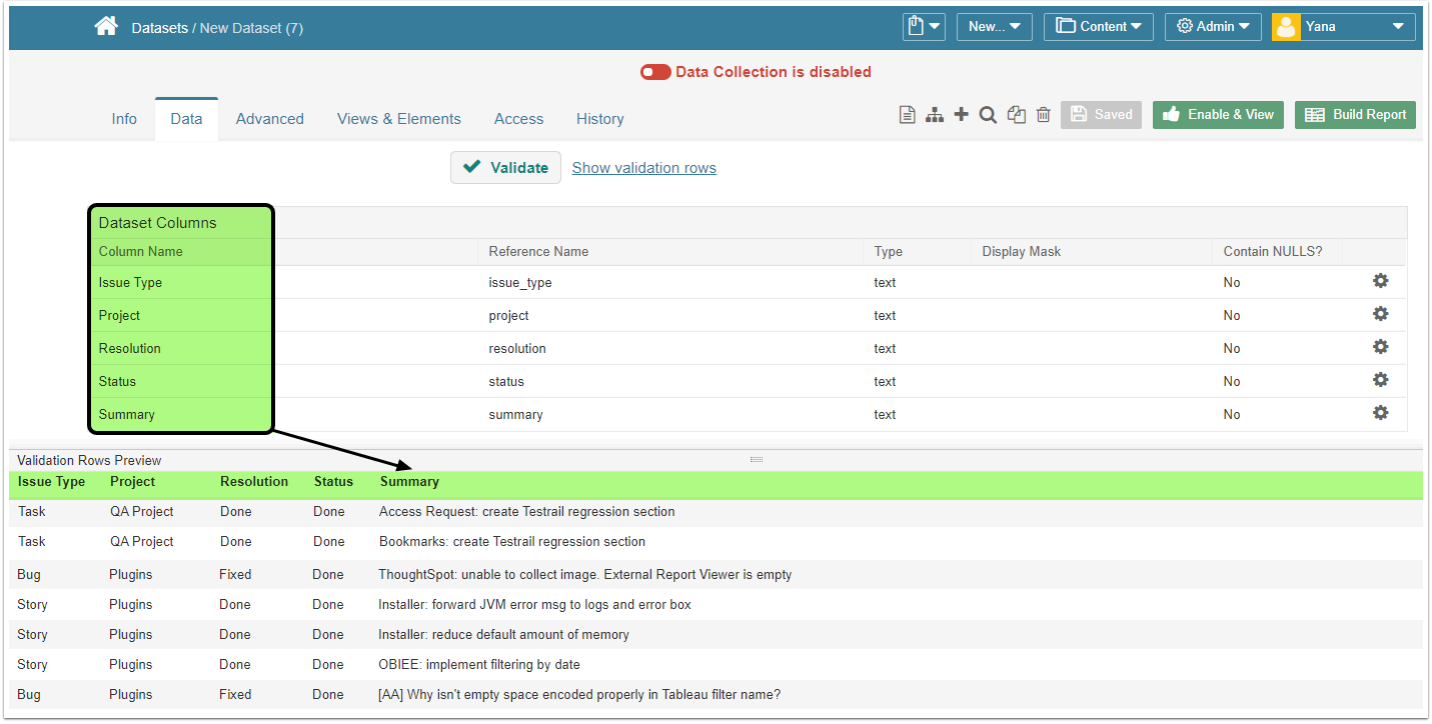
1. Specify Atlassian JIRA as **Data Source**
2. Choose the **Data Collection Trigger**
3. In the **Jira Report** field, select ONE or All Projects from which the data will be collected
4. Formulate your **Plugin command** using MIQL
5. **[Validate]**

2.2. Preview the Data



1. The fields specified in the Query are displayed in the **Validation Rows Preview**
2. **[Save]** to see the Dataset Columns List

2.3. View and manage the Dataset Columns List



Manipulate the display of Dataset Columns as required.

- To activate automated data collection and to be able to access Dataset Viewer, click **[Enable and View]** Dataset.

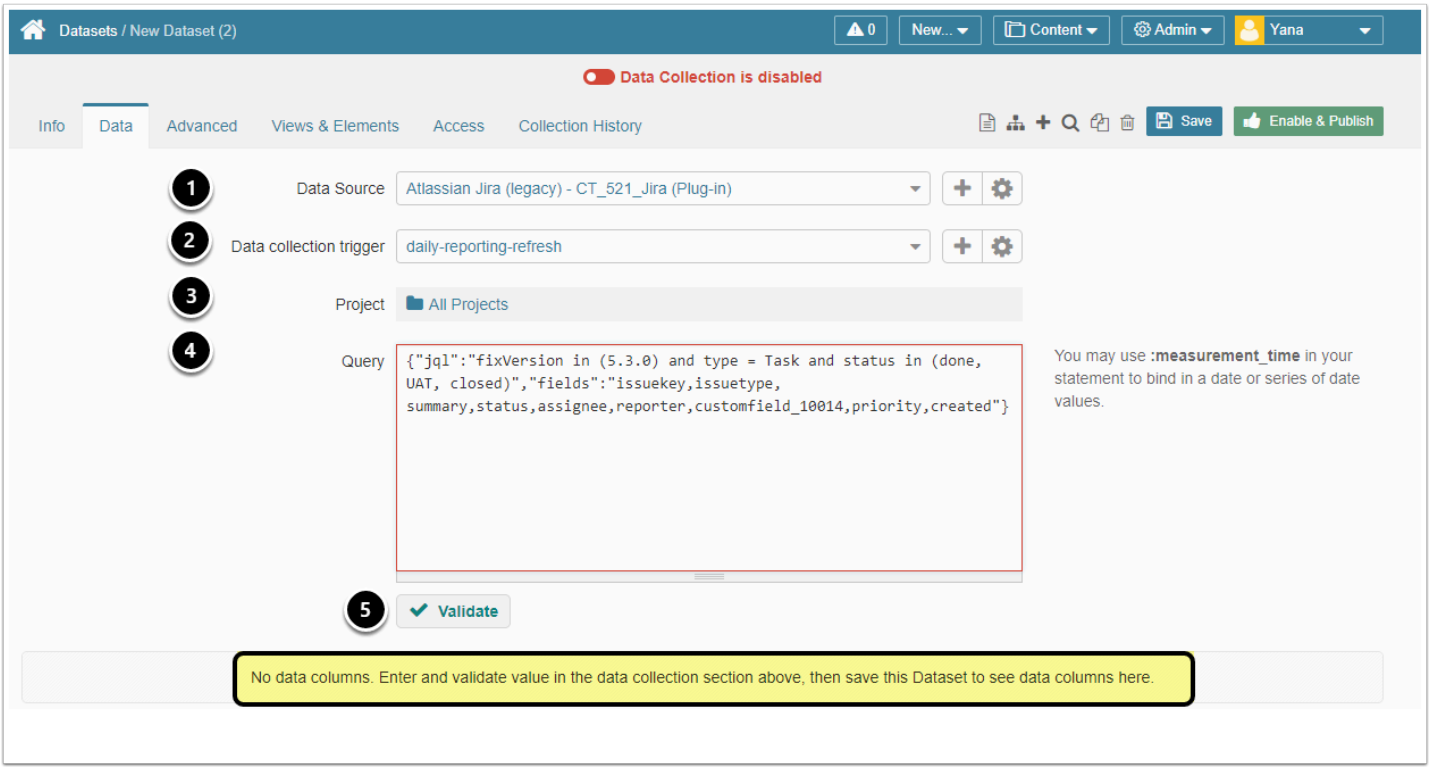
3. Use JQL commands to fetch data from Jira (prior to 5.3)

To build a fetch command in Version prior to 5.3, follow this syntax:

1. Formulate your query in braces **{}.**
2. The Query typically consists of 2 internal parameters - "jql" and "fields" (both must be followed by a colon (:), entered in quotation marks and separated with a comma).

SAMPLE QUERY: {"jql":"query_copied_from_jira","fields":"field_name", "field_name", "field_name"}

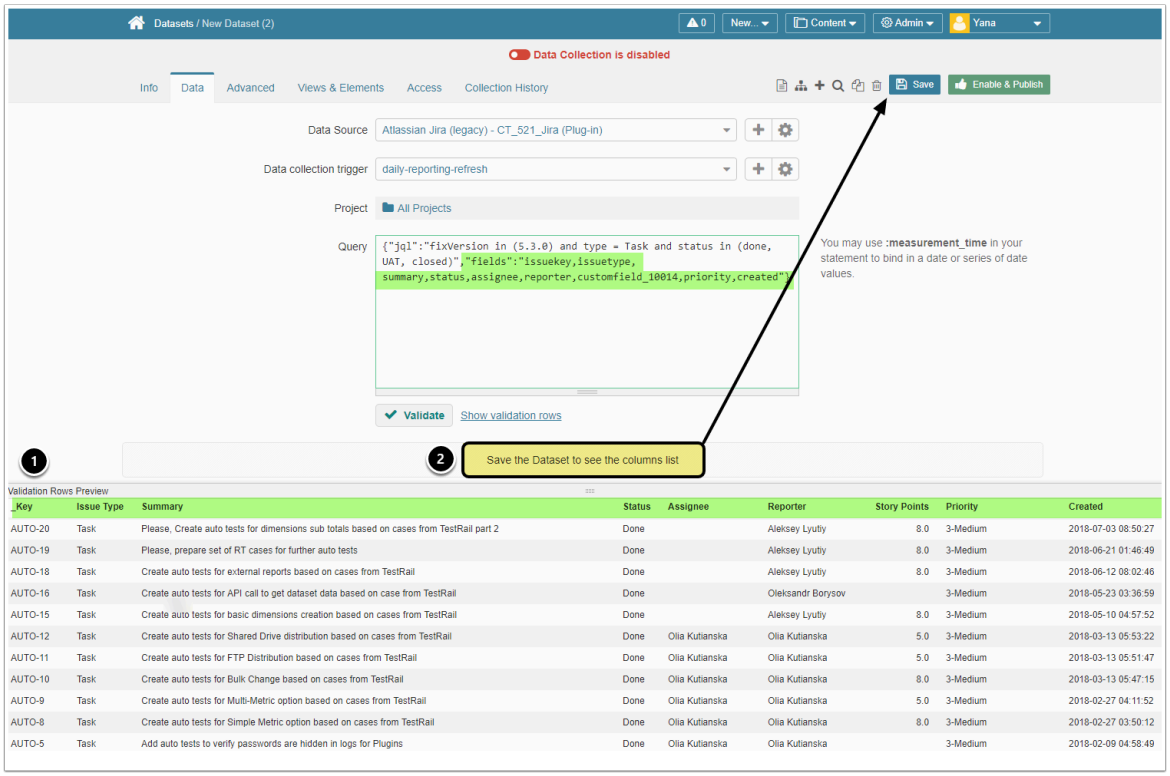
3.1. Dataset Editor > Data tab



In the Dataset Editor:

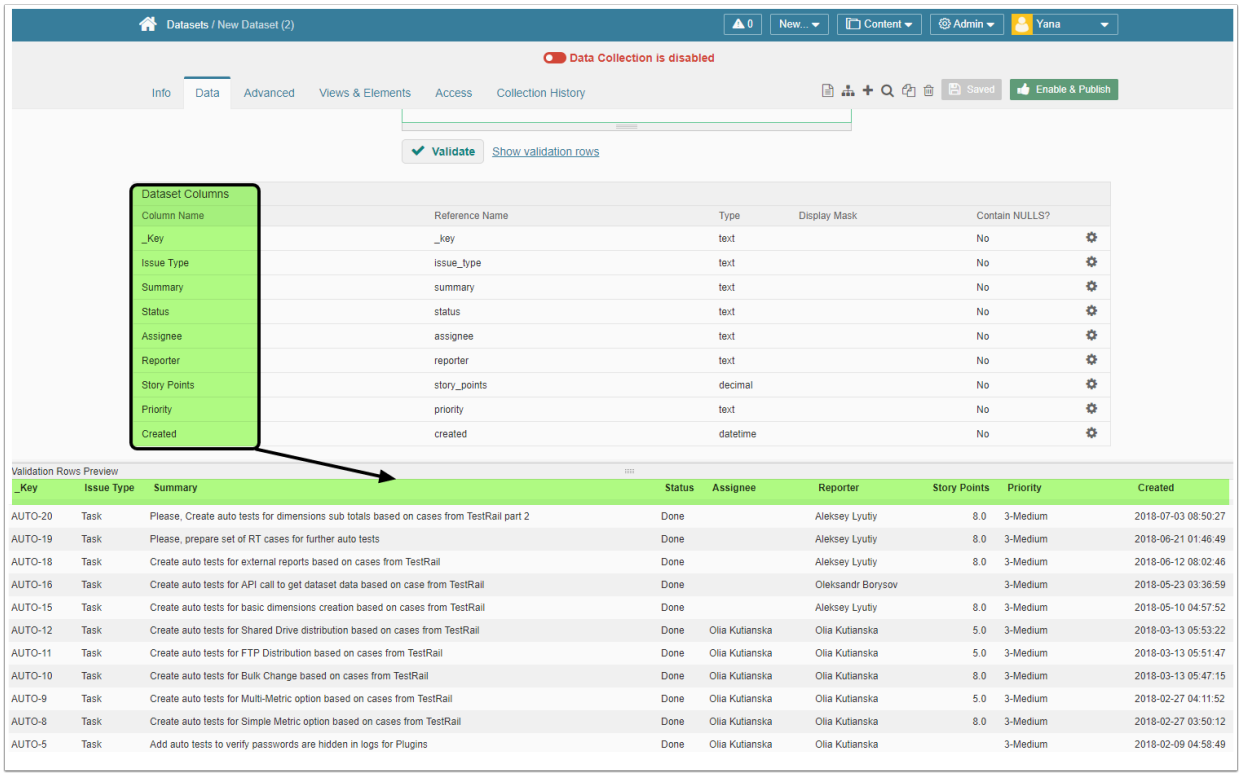
1. Specify Atlassian JIRA as **Data Source**
2. Choose the **Data Collection Trigger**
3. In the **Project** field, select ONE or All Projects from which the data will be collected
4. Formulate your **Query**
5. **[Validate]**

3.2. Preview the Data



1. The fields specified in the Query are displayed in the **Validation Rows Preview**
2. **[Save]** to see the Dataset Columns List

3.3. View and manage the Dataset Columns List



Manipulate the display of Dataset Columns as required.

- To activate automated data collection and to be able to access Dataset Viewer, click **[Enable and Publish]** Dataset.

10. Sourcing Data from SAP BusinessObjects

10.1 Establish connectivity to SAP BusinessObjects

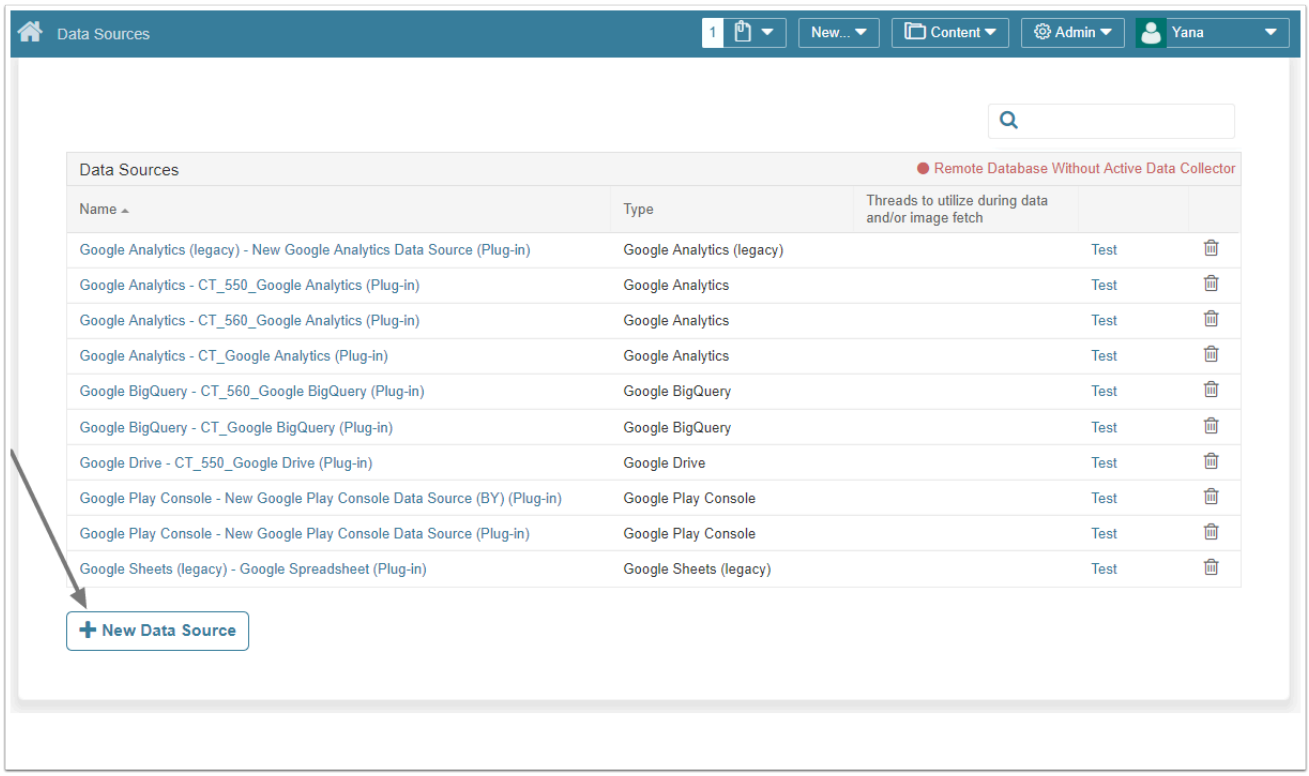
This article describes how to connect to **BusinessObjects (BOBJ)** in order to load data into Datasets and Reports in Metric Insights.

PREREQUISITES

Your Metric Insights instance must be configured to support BusinessObjects:

- BusinessObjects Rest API must be enabled *(default port is 6405)*
- 6400 and 6405 ports must be open for access to the BOBJ server from the MI server
- Web Application Container Server must be installed properly

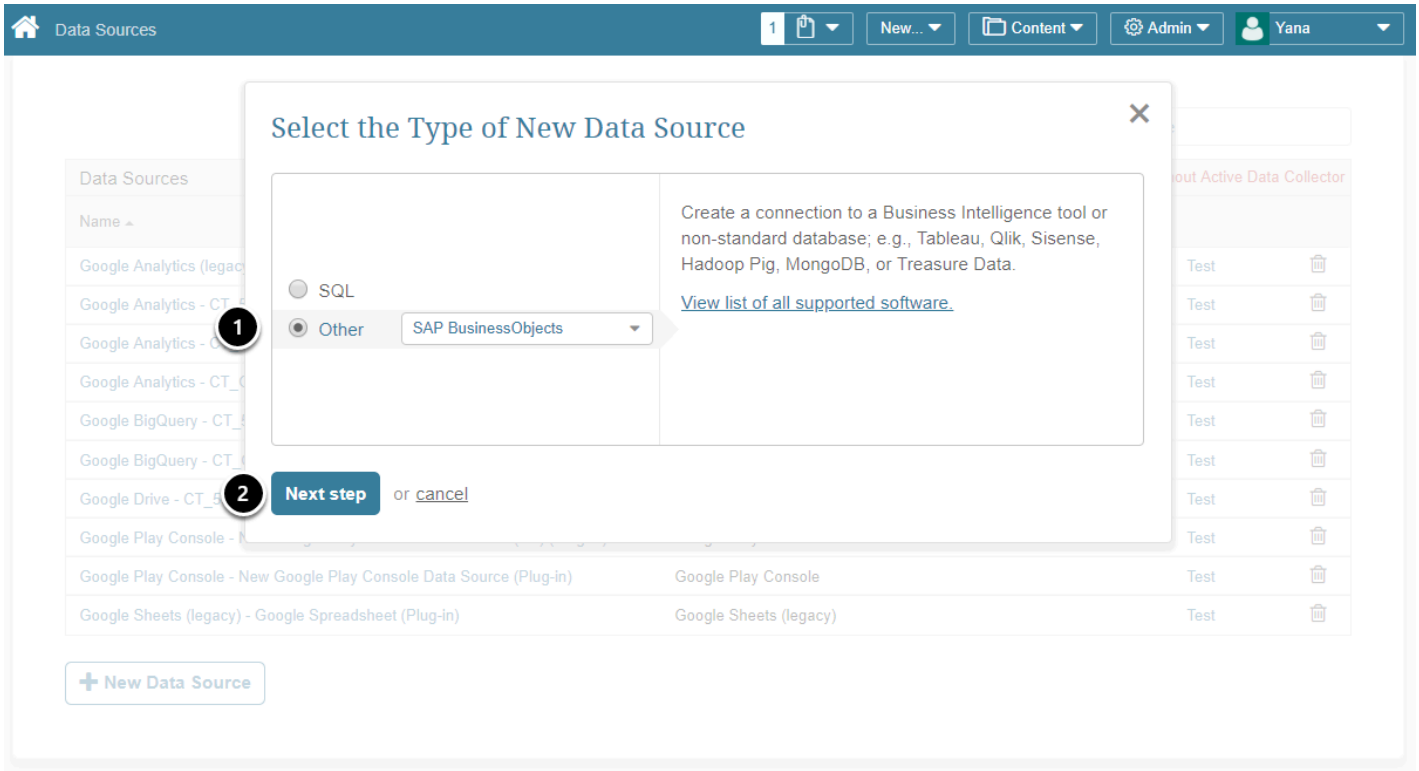
1. Access Admin > Data Sources



At the bottom of the screen, click **[+ New Data Source].**

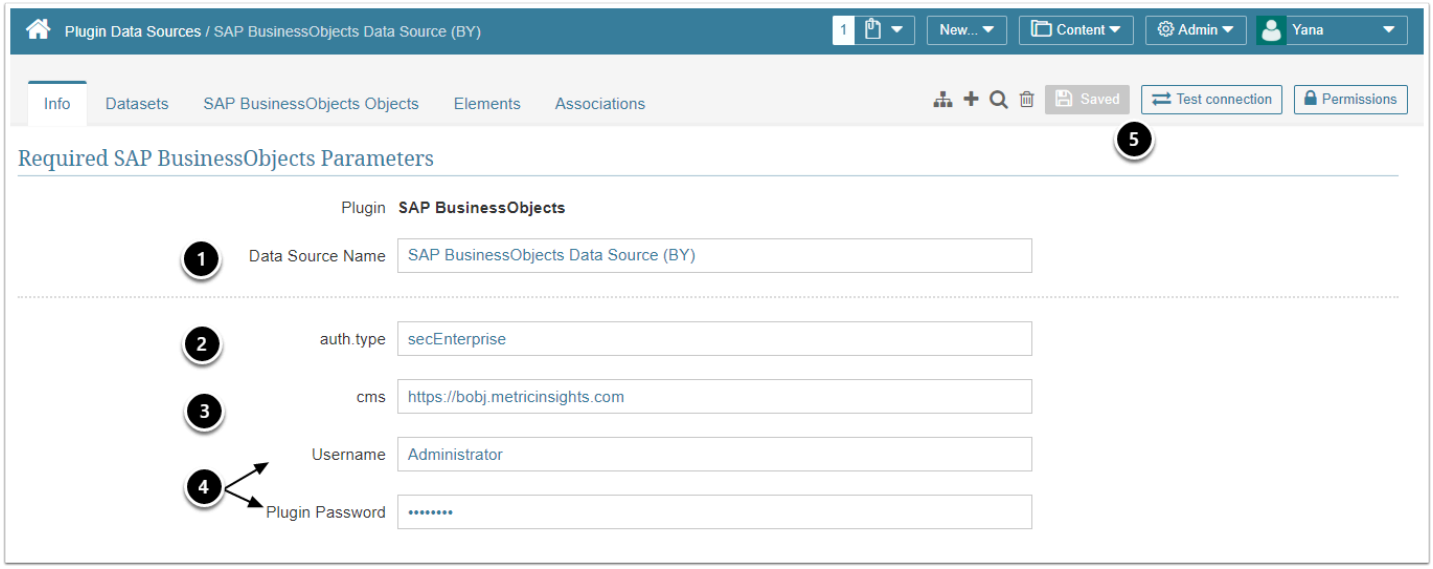
The *Select the Type of New Data Source* pop-up opens.

2. Select the Type of the New Data Source



1. Select "Other" and choose "SAP BusinessObjects" from the drop-down list
2. Move to the **Next step**

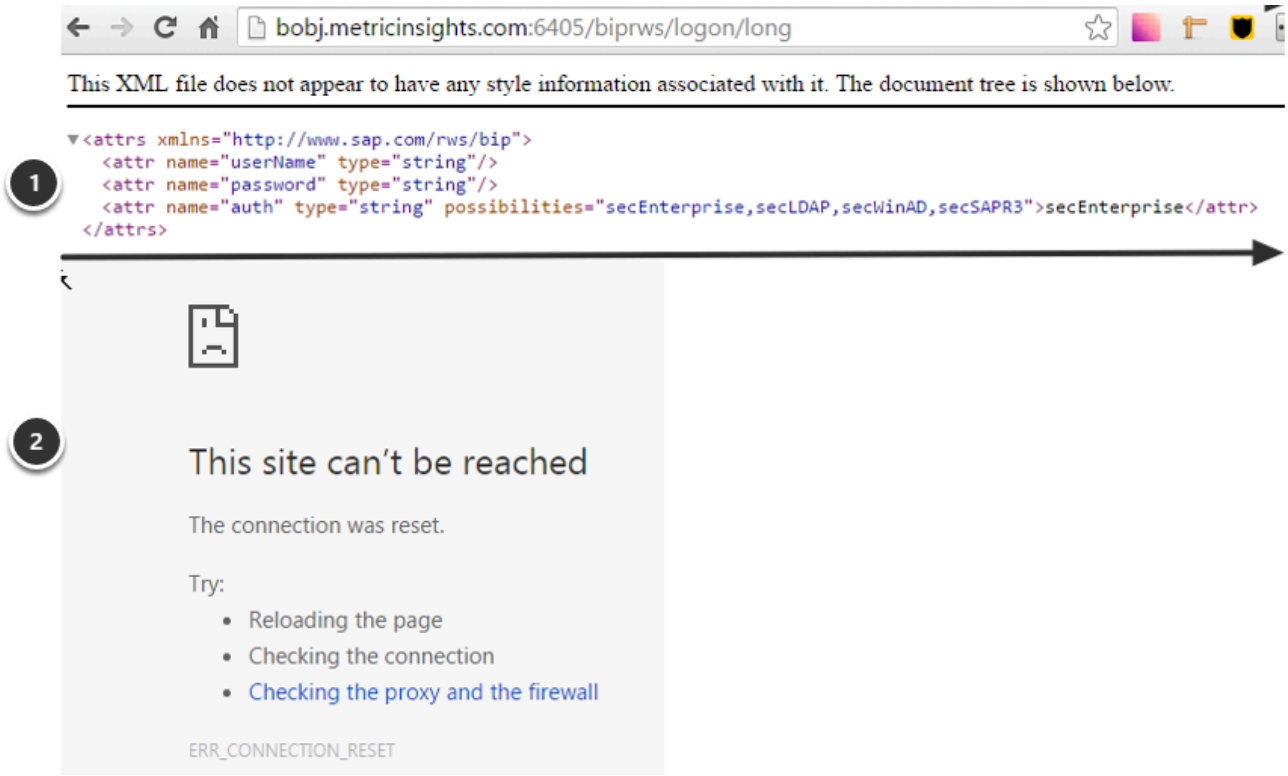
3. Provide the Required Parameters



Specify how to connect to SAP BusinessObjects:

1. **Data Source Name:** is default but you may modify it
2. **Auth.type:** define the authentication type used *(e.g., secEnterprise, secLDAP, secWinAD)*
3. **CMS:** provide the host hame of the BusinessObjects server *(include the HTTP/HTTPS scheme as shown above)*
4. **Username/Password:** note that your **Username** must be in the same format that your SAP BusinessObjects server uses for authentication
5. **Save** your entries and **Test Connection**

3.1. If connection fails, test that BOBJ is set up correctly



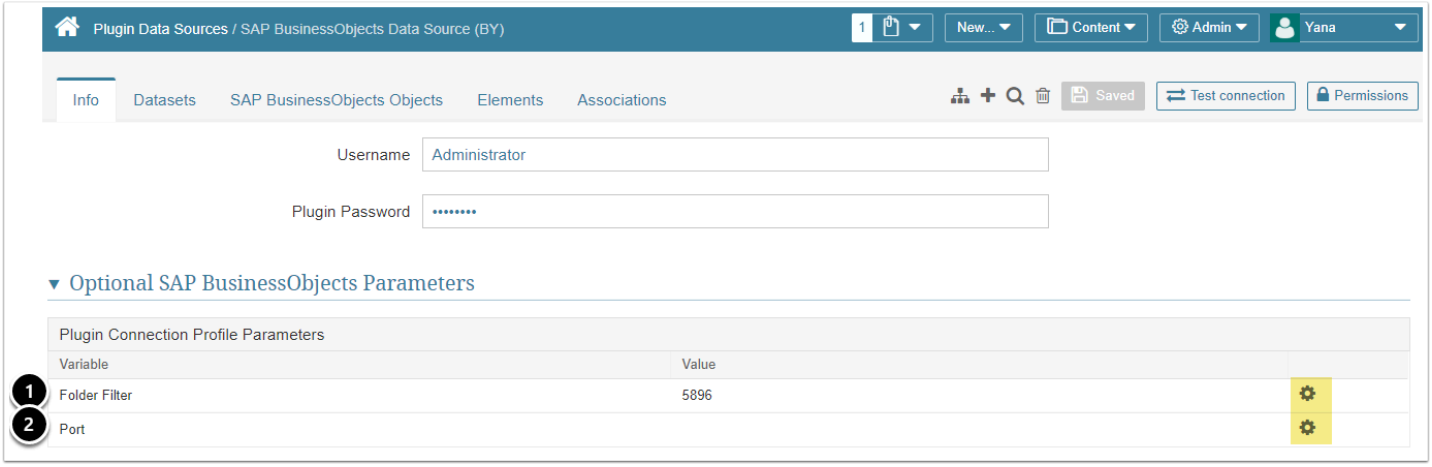
1. Enter the data below in the Command line to verify that Port 6405 is set as default for BOBJ:

```
http://bobj.<yourcompany>.com:6405/biprws/logon/long
```

If you receive the result as shown in the image, the Port is correct. Verify your settings for existing errors.

2. If you receive ***"This site can't be reached"*** message, the Port is set incorrectly. Contact your BOBJ admin to resolve the issue.

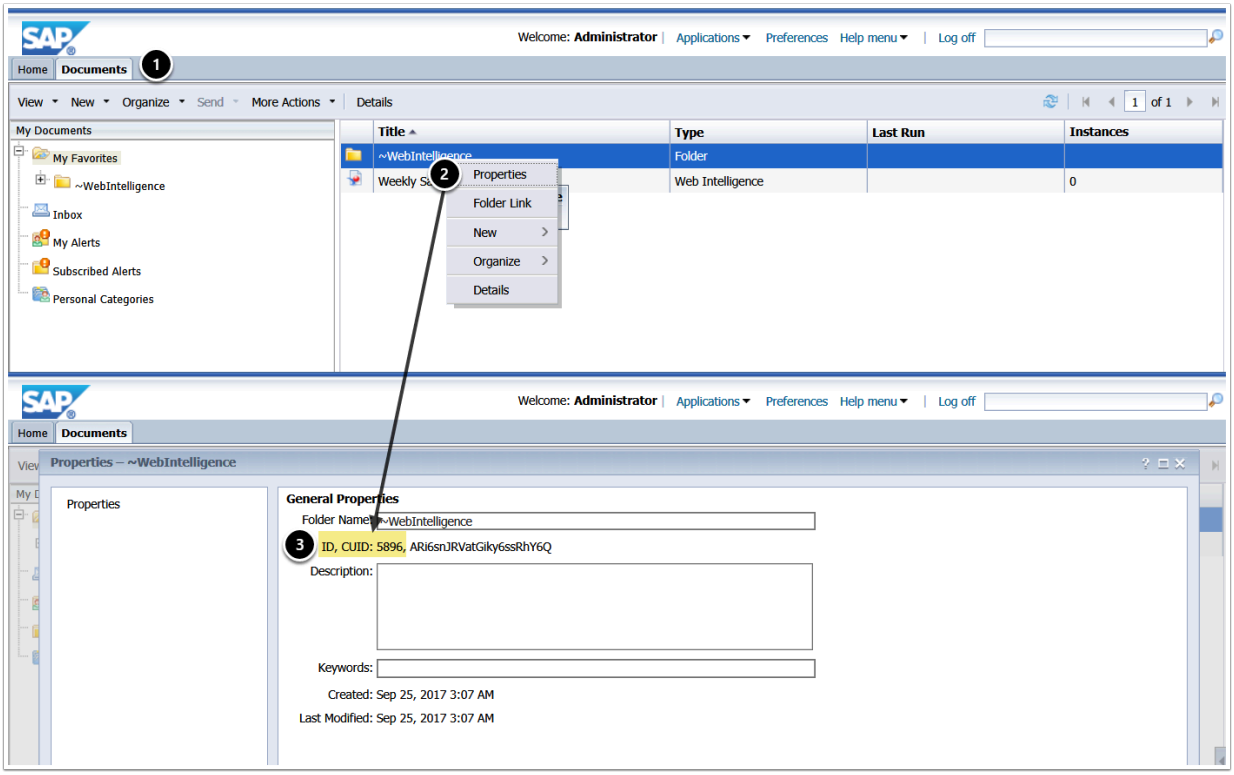
4. Optional Parameters



Click the **Edit (Gear) icon** to define optional parameters:

1. **Folder Filter** [optional]: specify the Folder ID to limit the number of Objects that will be fetched to Metric Insights
   - *You can also use a list of comma-separated Folder IDs*
2. **Port** [optional]: input the Port number if your server uses a port other than the default BusinessObjects REST API port (6405)

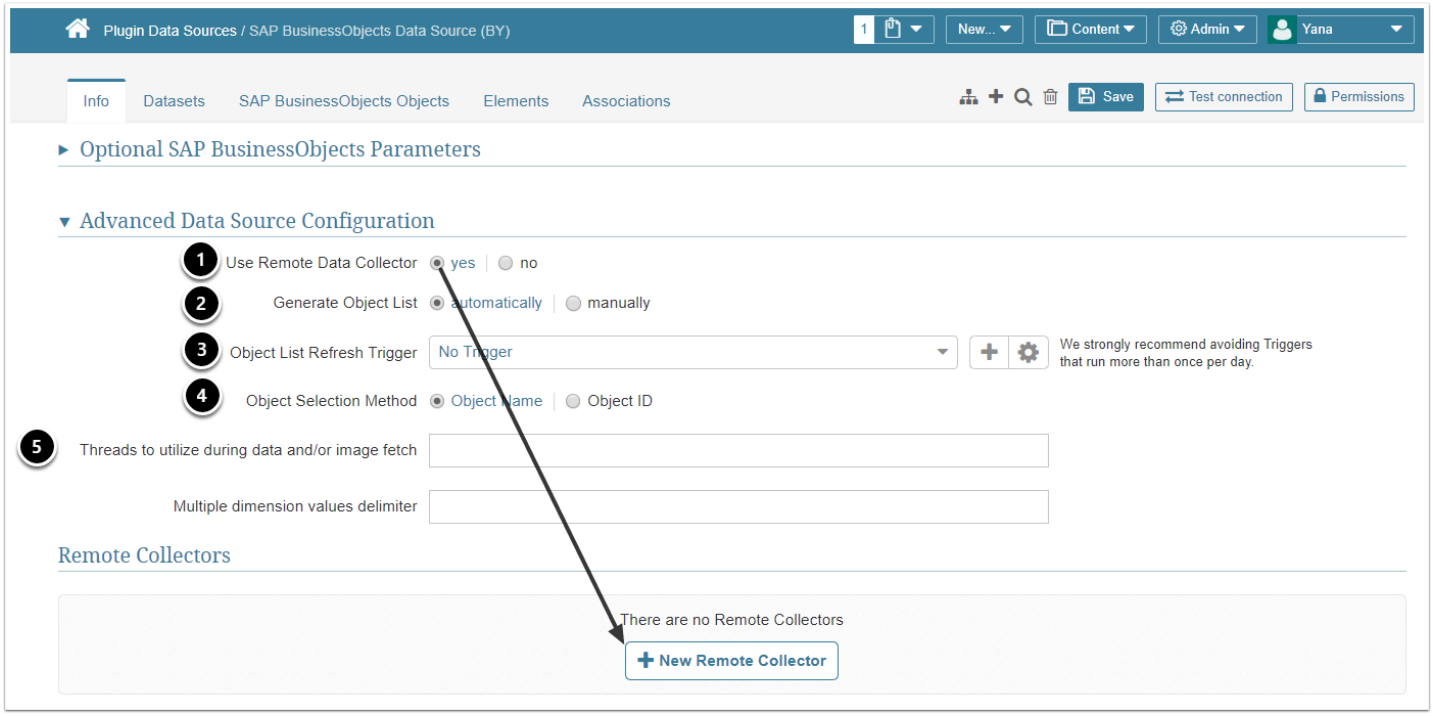
4.1. How to find a Folder ID in SAP Business Objects?



In SAP BusinessObjects UI:

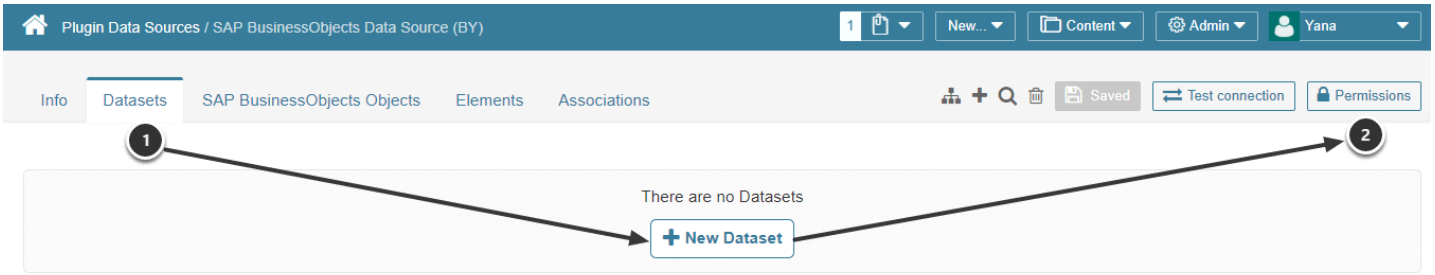
1. Open the **Documents tab**
2. Right-click the selected Folder > choose **Properties**
3. On the General Properties pop-up window, find and copy the **Folder ID**

5. Advanced Configuration



1. **Use Remote Data Collector:** is set to "no" by default
   - If required, switch to "yes" and add a Remote Data Collector by clicking **[+New Remote Collector]**
2. **Generate Object List**
   - *automatically*: all Reports are going to be fetched by the system
   - *manually*: Reports may be added one-by-one or via CSV file
3. **Object List Refresh Trigger:** from the dropdown, select the Trigger that will be used to fetch data via the Google Play Console plugin
4. **Object Selection Method:** specify how Google Play Console Reports will be fetched
5. Optionally, state the maximum number of concurrent **Threads to utilize during data and/or image fetch** to be used in background processing when the system updates Reports for this Data Source
   - If you do not specify any value for this setting, batch data collection processing will be single-threaded

6. Other Settings



1. You can Datasets directly from the respective tab
2. Click **Permissions** to assign Permissions to Groups or Power Users

7. What's next?

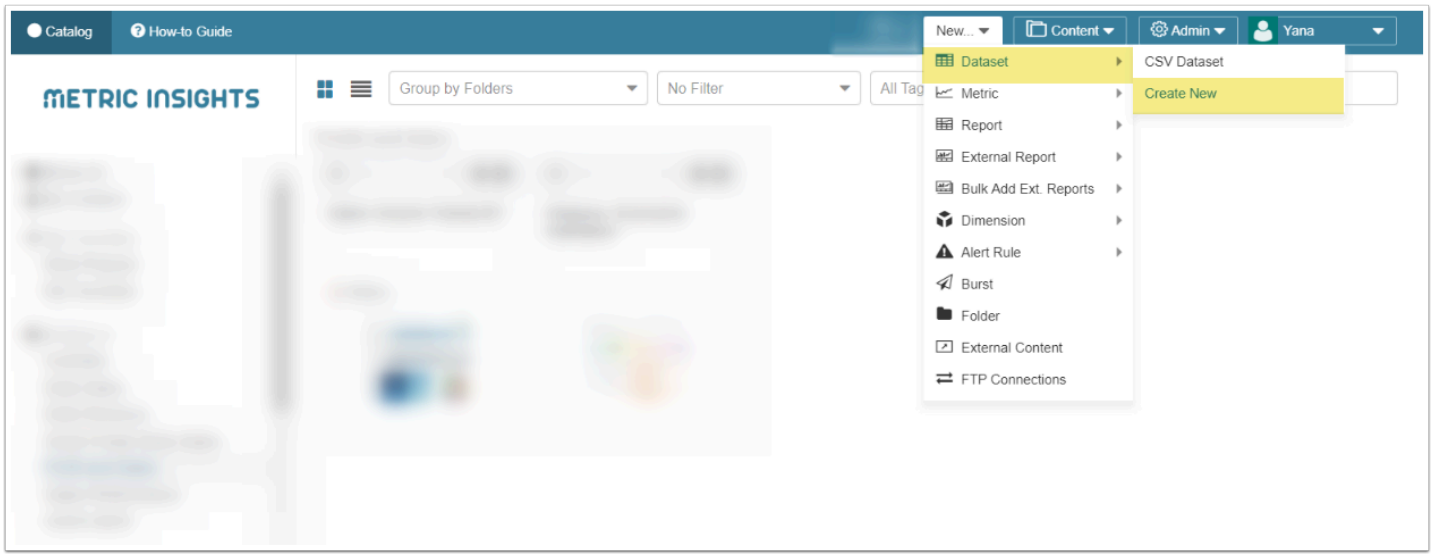
[How to collect data from Business Objects]

10.2 Collect data from SAP BusinessObjects

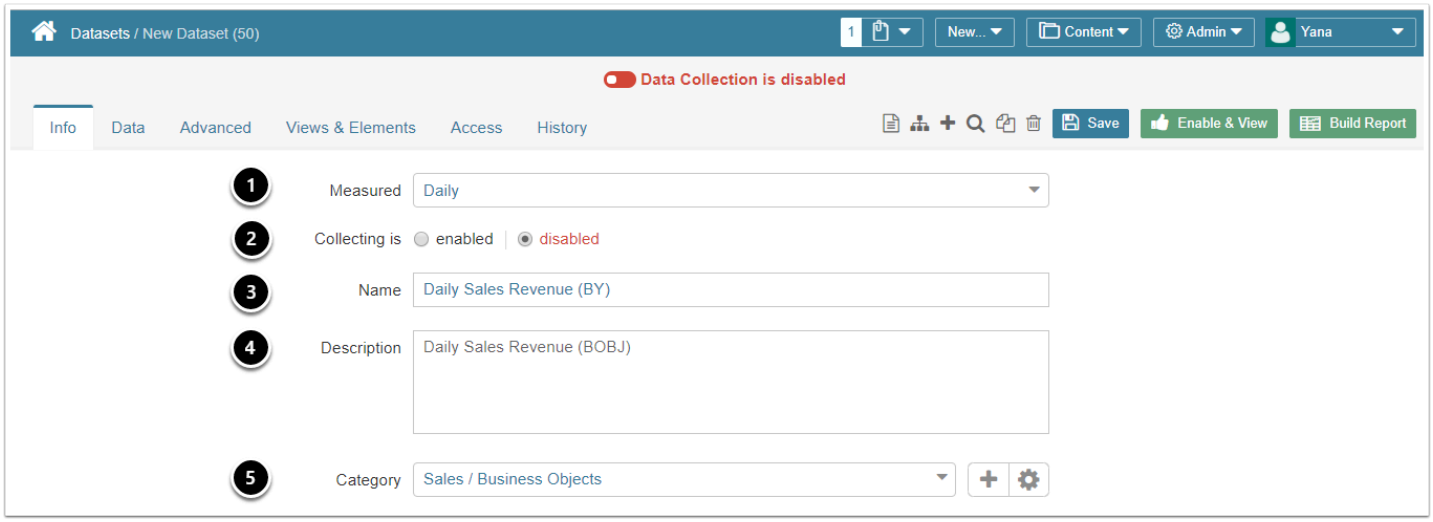
This article details how to create a Dataset populated with data sourced from SAP BusinessObjects.

It assumes that you have already established connectivity with your SAP BusinessObjects server via the respective plugin connection profile.

1. Access New > Dataset > Create New



2. Dataset Editor > Info tab



Define the basics:

1. **Measured:** select the measurement interval that applies to the level of aggregation that you want in your result set
2. **Collecting**: new Datasets are always disabled by default to make sure that you can take time to configure them properly before enabling. This setting is duplicated at the top of the screen
3. **Name:** provide a unique name for your Dataset. Preferably, the Dataset name should explain what kind of data it contains
4. **Description:** optionally, provide any additional information about your Dataset
5. **Category:** specify the Category where you Dataset will be placed

Move to the ***Data tab*** to define the source of data and how often it should be updated.

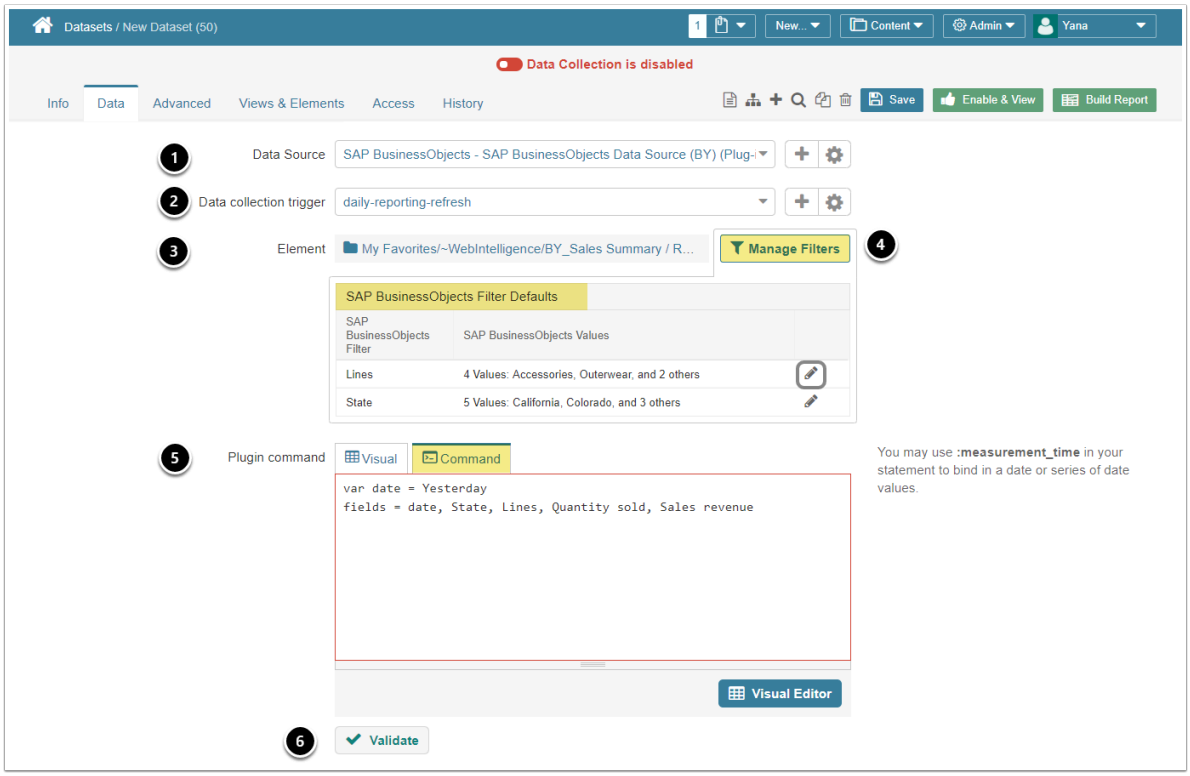
3. Define the Settings for Data Collection

Data fetching can be configured using the following options:

- MIQL query
- Visual Editor

See details below.

3.1. Using MIQL Plugin Command



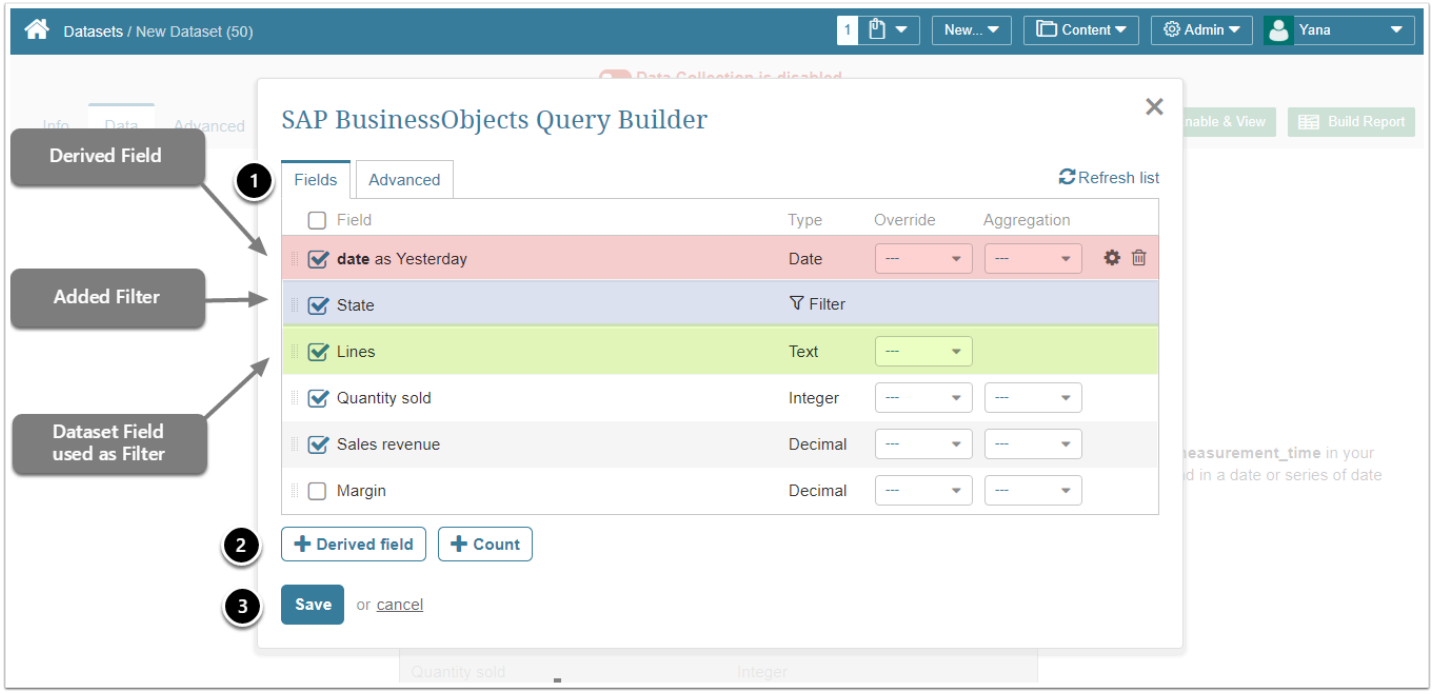
1. **Data Source**: select the connection profile you have created for *SAP BusinessObjects*
2. **Data collection trigger**: specify the Trigger that will be used to collect data for your Dataset
3. **Element:** select a *SAP Business Object* that should serve as a basis of your Dataset
4. Click **[Manage Filters]** to add Filters to your data
   - All added Filters will be displayed under **SAP BusinessObjects Filter Defaults** below
   - Click the **Edit (Pencil)** icon to specify how many Values will be used for data filtering
   - For more information on using Filters, view [Pre-filtering SAP BusinessObjects data]
5. In the **Command tab**, input an [MIQL] **Plugin Command** listing all data that needs to be fetched from *SAP BusinessObjects*
6. **Validate** your query

Note!

1. Entire field names that contain special characters, aggregation and commas must be enclosed in quotes (single or double).
2. It is acceptable to enclose all fields and values in quotes.

[...] + Notation is used to signify that the MIQL parts of a statement are optional/can be repeated.

3.2. Using the Visual Editor



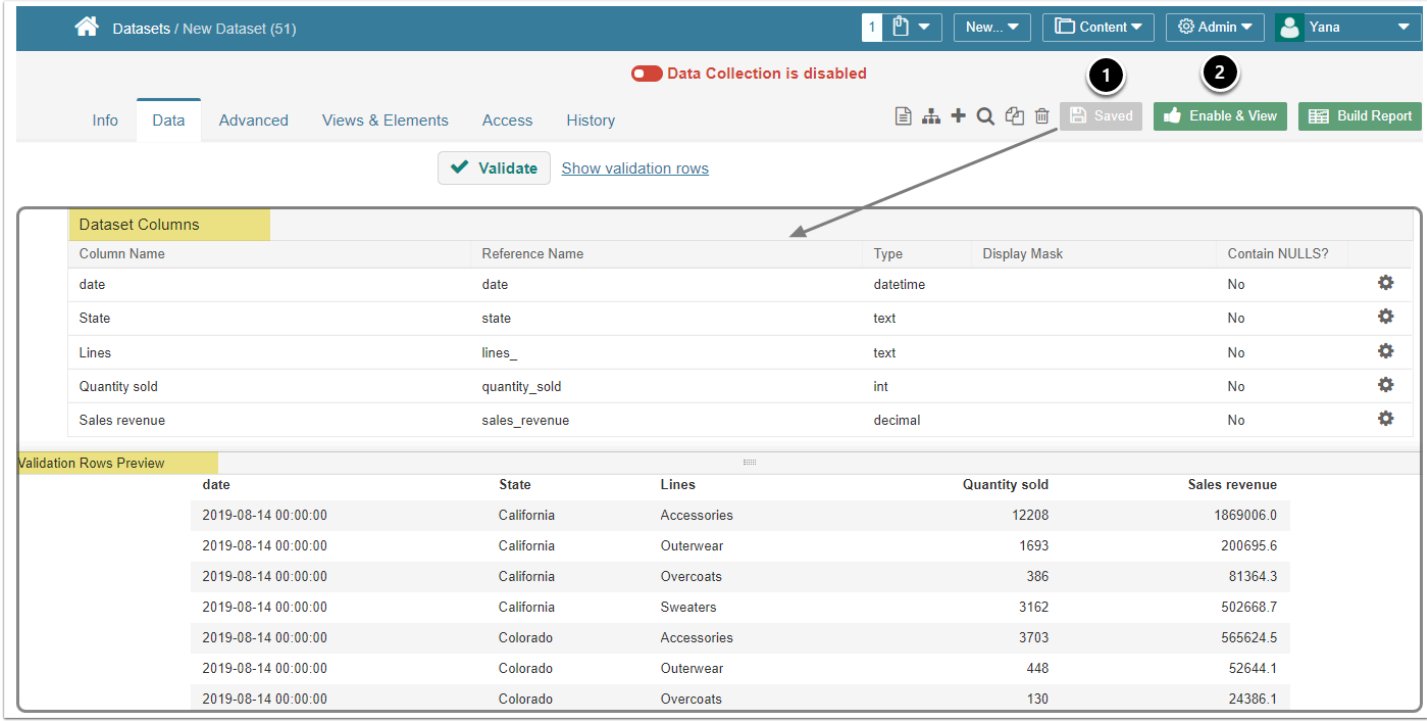
The **SAP BusinessObjects Query Builder** allows for data fetching without the need to learn the plugin syntax and helps avoiding typos/mistakes.

1. Select the **fields** for your Dataset
2. *Optionally,* add **Derived fields** and/or **Count** of duplicate rows

Save your settings. Plugin command validation will start automatically.

> **Note!**
>
> 1. *Derived fields* can later be modified using the **Edit (Gear) icon**
> 2. *Filters added as a field to the results set* are distinguishable by a **Filter (Funnel) icon**

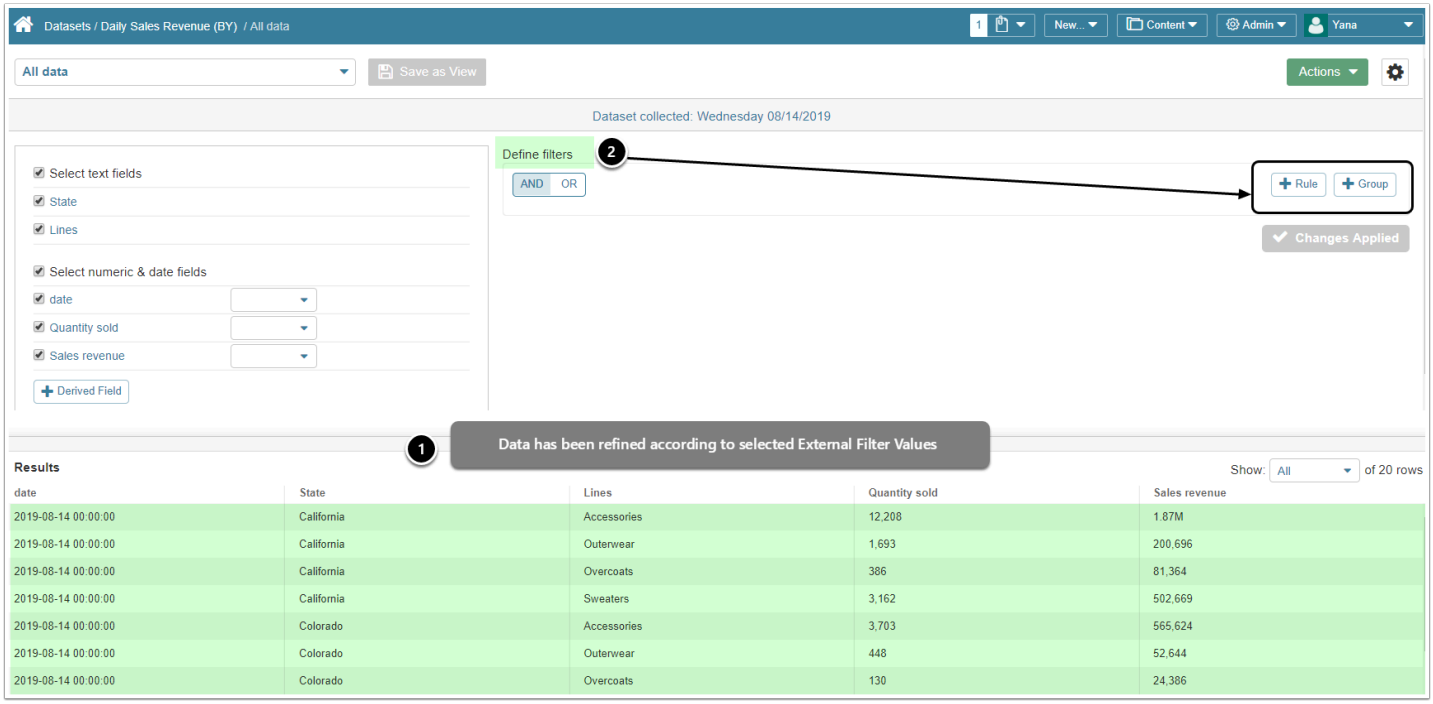
4. Plugin command will be validated and data collected on Save



1. If the command is validated successfully, the **Dataset columns** and **Data Preview** are going to be shown below.
2. At the upper right corner of the screen, click **Enable & View**.

5. Dataset will be displayed in Viewer

If any Filter has been applied, pre-filtered data will be displayed in Viewer.



In the **Dataset Viewer**:

1. In the Results Section, you will see data with already applied **External Filters**
2. You can further refine your data with Dataset's *internal filtering options* by **applying Rules** and **Grouping Data**:
   - For more information on using Internal Filters, refer to Create a Dataset View
   - For general instruction on building Datasets, see Create a Dataset from any Data Source

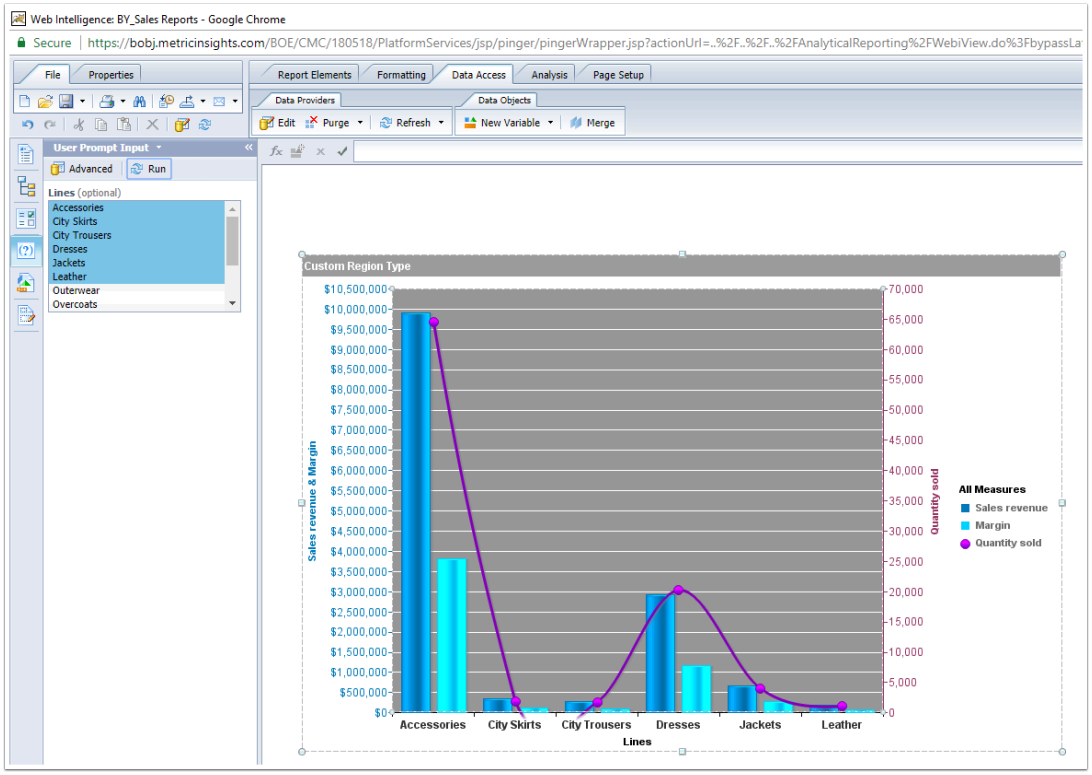
10.3 Pre-filtering SAP BusinessObjects data

When sourcing data from SAP BusinessObjects for Metrics, Reports, External Reports, Dimensions and Datasets, you can pre-filter your data before fetching it. This function allows to focus on the slice of data that you really need and exclude those values that are currently irrelevant for you and your research.

PREREQUISITES:

Establish Connectivity to SAP BusinessObjects

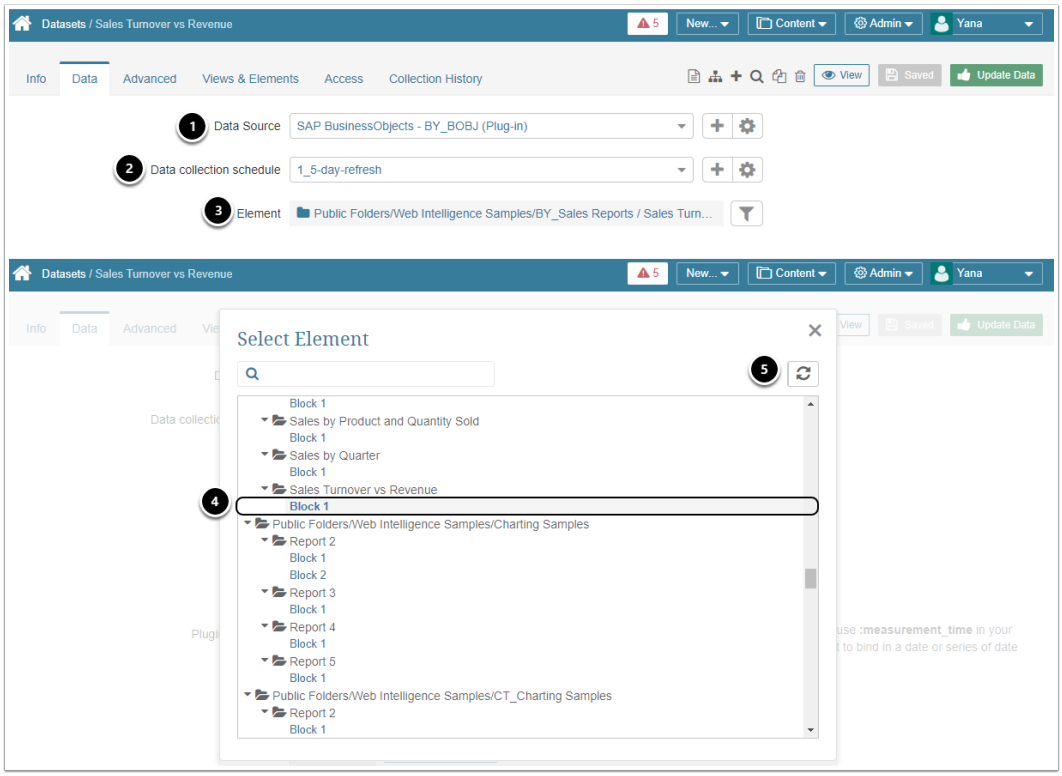
How to find Filter names in SAP BusinessObjects?



Most of the data displayed on the SAP BusinessObjects Views can be filtered by different criteria. You can choose to include all measurements or exclude information that is currently irrelevant for you.

On the example below, the data view was modified by values in the **Lines filter** displayed to the left of the graph.

1. Define a Source Object for Data Collection



Start by creating an element or Dataset. Once you get to the process of Data Collection, define the following:

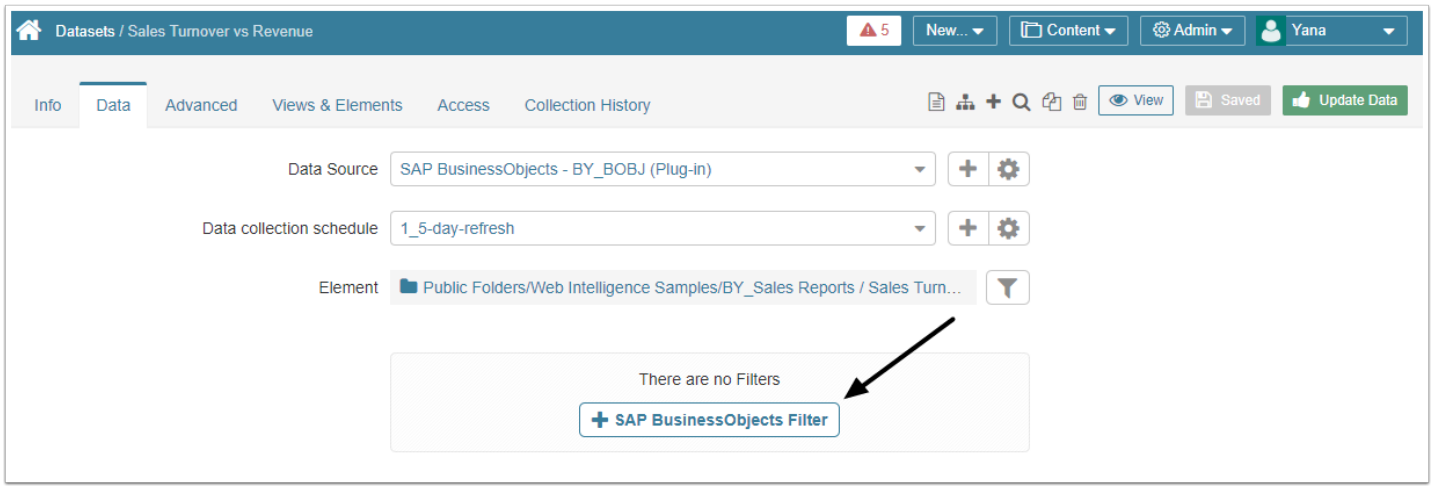
1. **Data Source:** This is an entity that connects SAP BusinessObjects and Metric Insights. For more information, see: Establish Connectivity to SAP BusinessObjects
2. **Data Collection Trigger:** select the Trigger that is going to initiate updating information in this element/Dataset.
3. **Report:** Click **Select Element** to open the pop-up with the list of available SAP BusinessObjects reports that can be a source of data.
4. Each item in the list is represented as the path (hierarchy) to the respective Report in SAP BusinessObjects. Find the desired report in the list.
5. If you do not see the required item, try refreshing the list by clicking the **Refresh** icon at the upper right corner of the pop-up.

2. Adding SAP BusinessObjects Filters to Metric Insights

⚠ Once filters are added to a Metric / Report or External Report for the first time, they are going to be automatically added to all new respective elements with the same Data Source / Report.

NOTE:

- External filters are tied to SAP BusinessObjects Reports, not Metric Insights' elements. This allows Filters to be reused for multiple elements (there is no need to create new Filters every time an element is created in Metric Insights).
- If there are more External Filters or Filter Values that you would like to use for the current element, you can always set the redundant ones to "ignore".

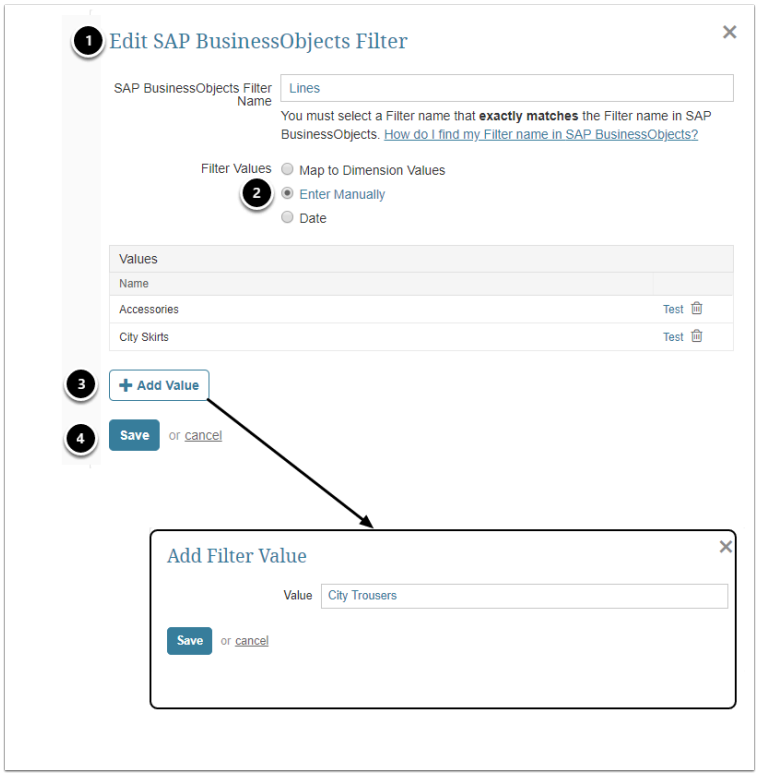


When creating a Metric / Report / External Report / Dataset fetched from SAP Business Objects, after you define the **Report** that should serve as a Data Source, you may pre-filter information that is going to be fetched.

To pre-filter the data, click **[+SAP BusinessObjects Filter].** The following pop-up will give you 3 options to add Filters.

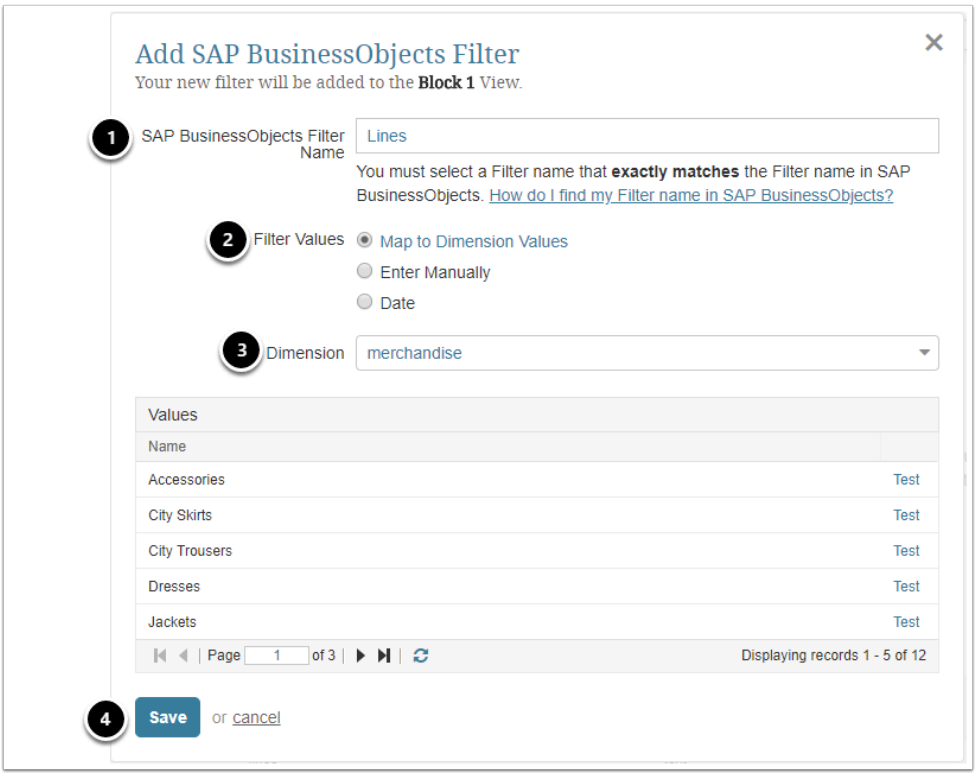
NOTE: Examples given below are taken from the SAP BusinessObjects Report shown at the top of the page.

2.1 Enter manually



1. **SAP BusinessObjects Filter Name**: Define the name of the filter from SAP BusinessObjects (The name of the Filter must exactly match the column names of the SAP BusinessObjects Report. Filter names are case sensitive. Unless the match is exact, the Filter will not work).
2. **Filter Values:** Choose 'Enter Manually' and click **Save** at the bottom of the pop-up.
3. Click **[+ Add Value]** and in the opened pop-up type in the name of the filter value. **Save** your entry. All added values appear in the *Values* list.
4. **Save** your entries.

2.2 Using dimension values

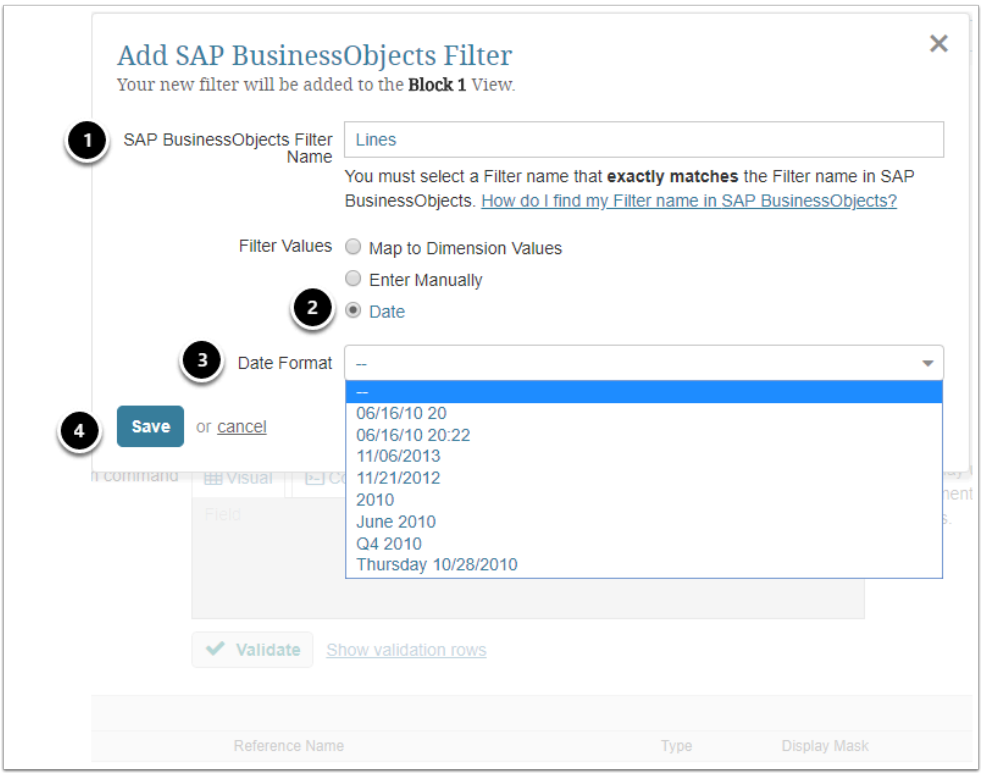


If you have already used SAP BusinessObjects filters to create Dimensions in Metric Insights, you can quickly choose which Dimension Values you want to use for pre-filtering:

1. **SAP BusinessObjects Filter Name**: Define the name of the filter from SAP BusinessObjects.
2. **Filter Values:** choose 'Map to Dimension Values'.
3. **Dimension:** select a corresponding Dimension from the drop-down list and all its Values are going to be loaded to the Values list automatically.
4. **Save** your entry.

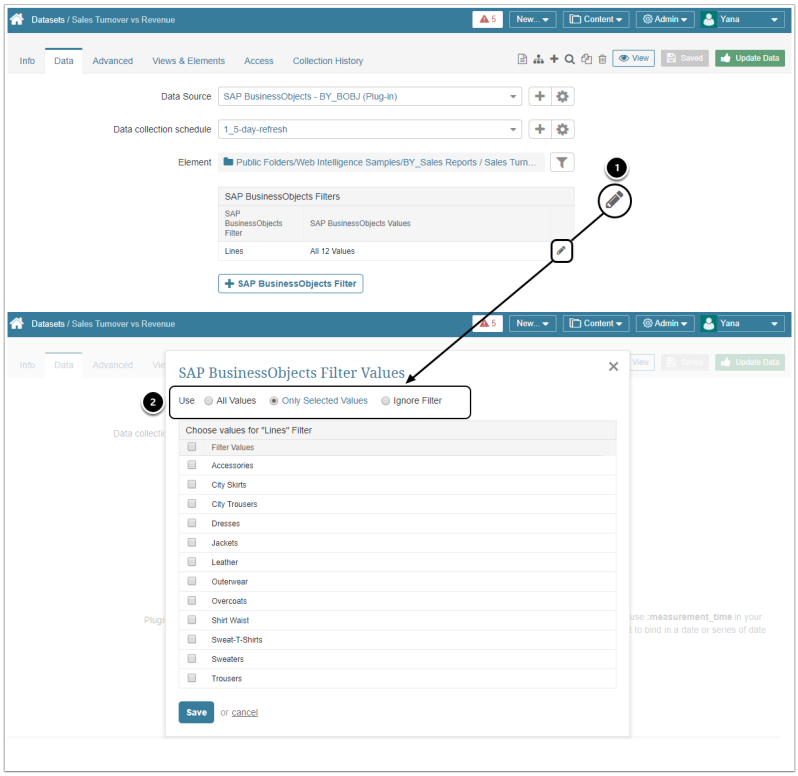
NOTE: Dimensions used here must have Values that exactly match the Filter Values in SAP BusinessObjects (if the Values do not match, the Filter will not work)

2.3 Using Date



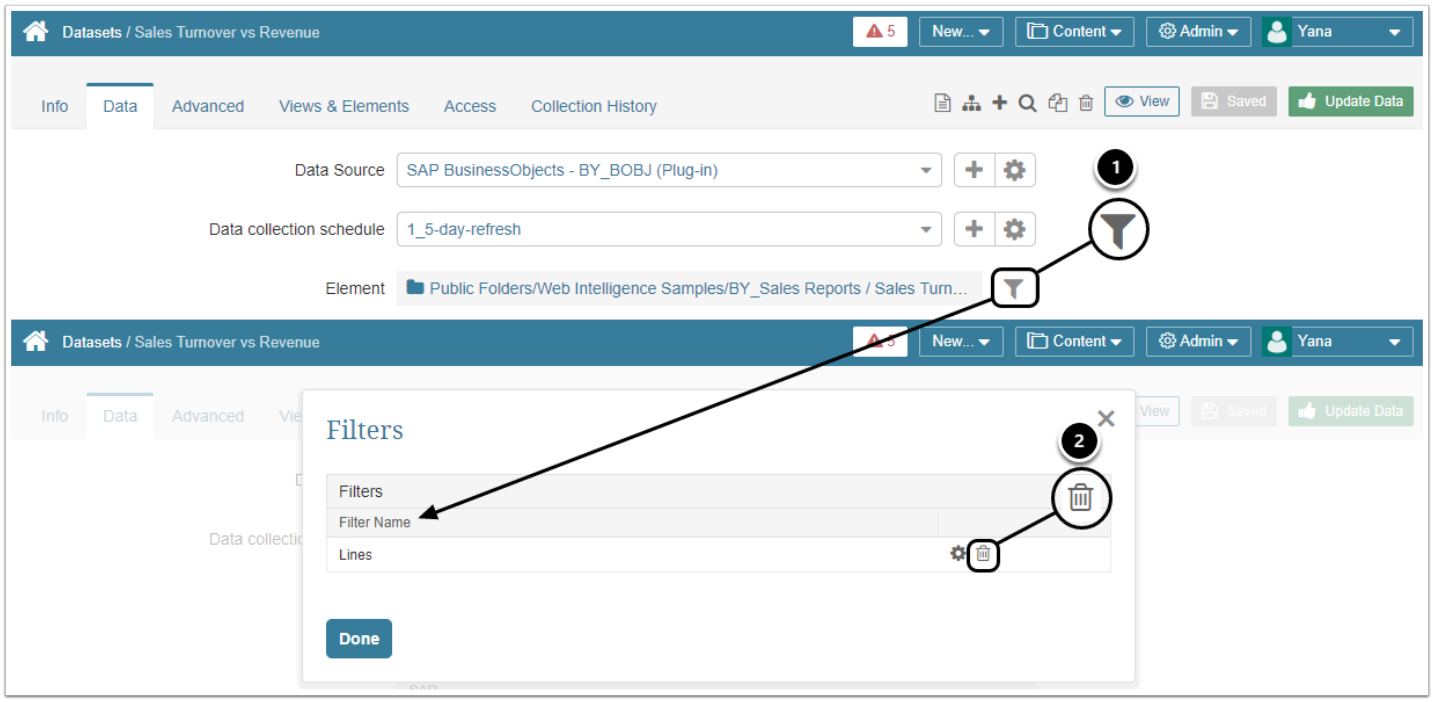
1. **SAP BusinessObjects Filter Name**: Define the name of the filter from SAP BusinessObjects (The name of the Filter must exactly match the column names of the SAP BusinessObjects Report. Filter names are case sensitive. Unless the match is exact, the Filter will not work).
2. **Filter Values:** Choose 'Date'.
3. Select the **Date Format** used in your SAP BusinessObjects Report.
4. **Save** your entries.

3. How do I add filters to a results set from SAP BusinessObjects?



1. Click the **Pencil** icon in the filter row to add it.
2. When the filter is added, you can use it for "All Values", "Only Selected Values" or ignore it.

4. Deleting Filters



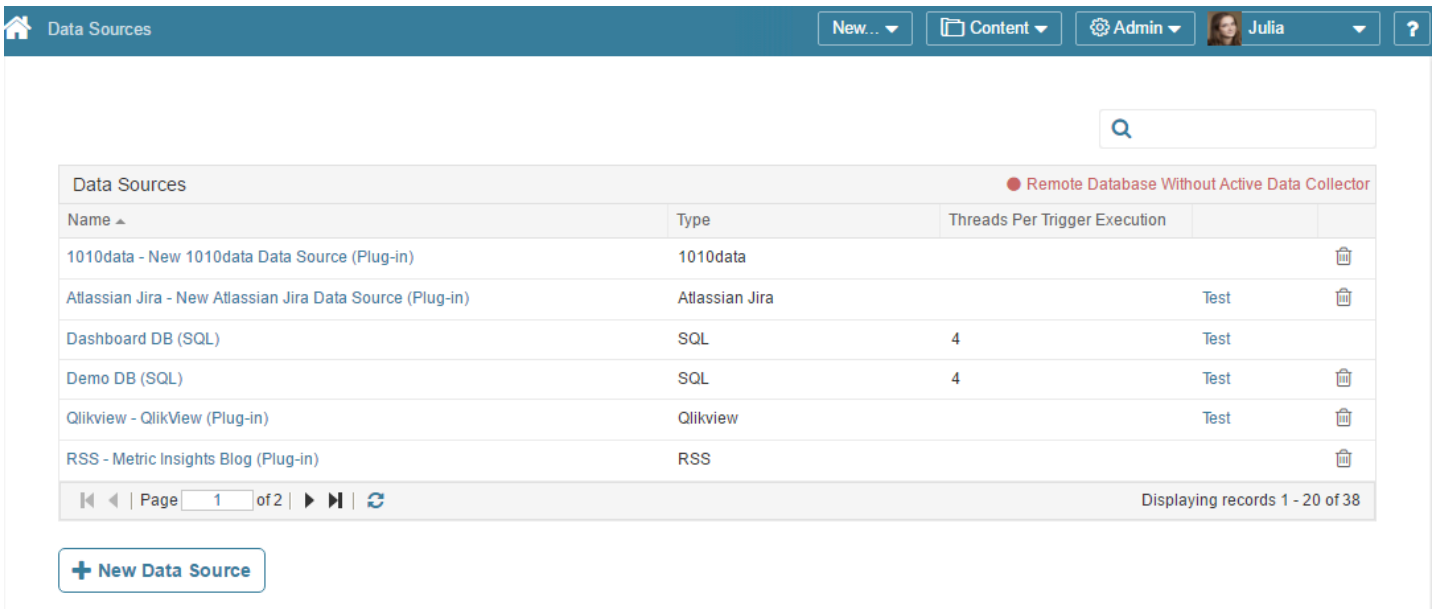
To delete some of the added filters: (1) click the **Filter** icon in the **Report** field and (2) choose the unnecessary filters. Click the **Trash bin** icon in the corresponding row.

11. Sourcing Data from Beckon

11.1 Establish connectivity to Beckon

This article describes the process of creating plug-in Data Source to connect to the Beckon server. This Data Source will allow data from existing Beckon objects to be used in building elements using Metric Insights tools.

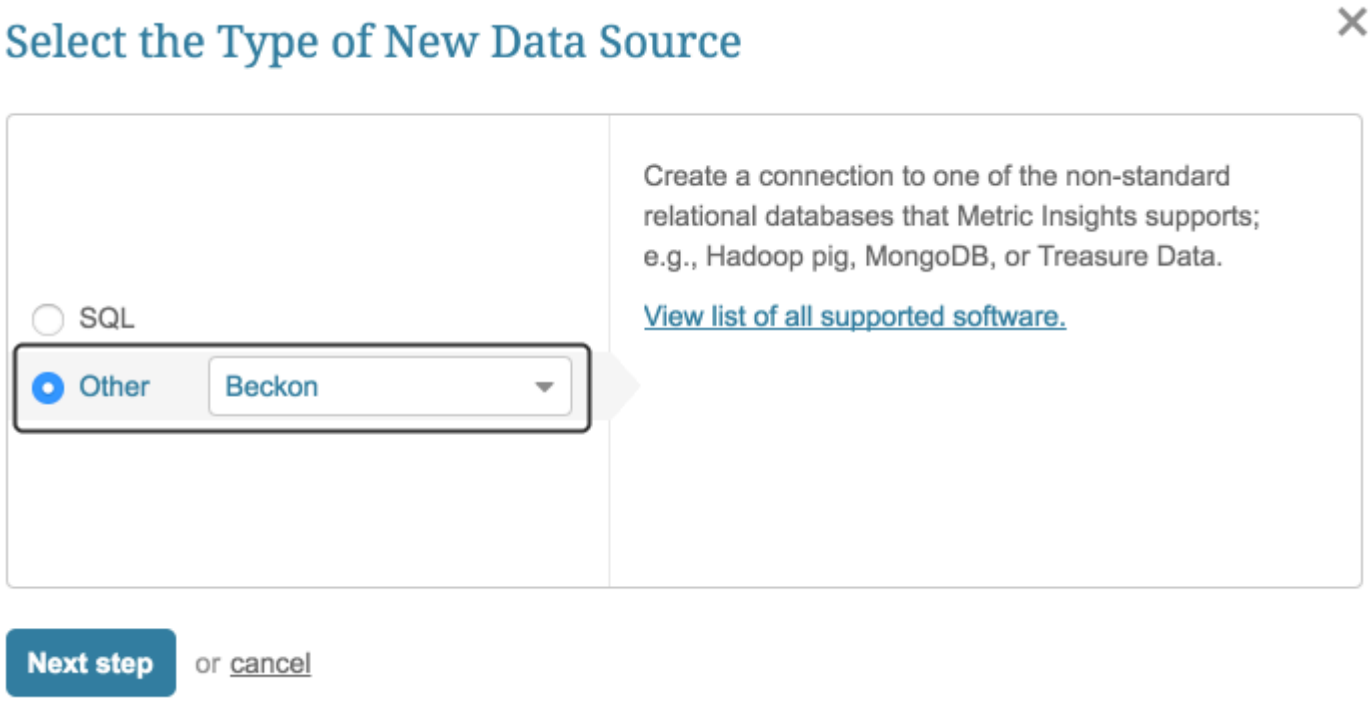
1. Access Admin > Data Sources



At the bottom of the screen click **[+ New Data Source].**

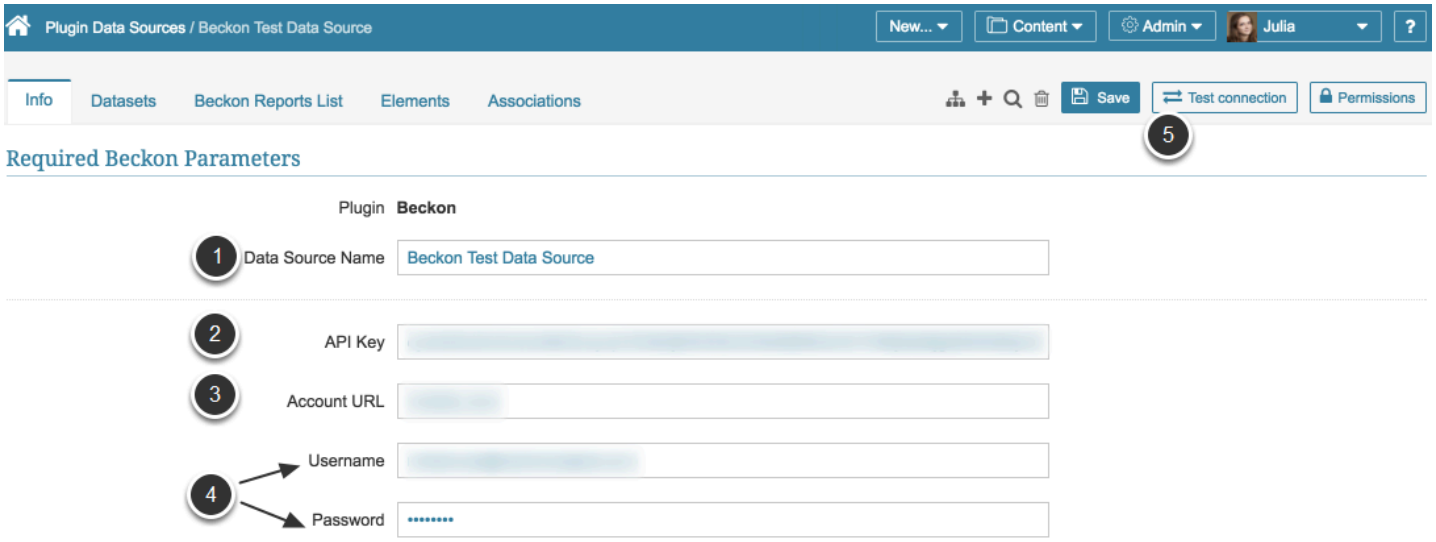
The *Select the Type of New Data Source* pop-up opens.

2. Select "Other" Data Source Type and choose "Beckon" from the drop-down list



Proceed with creating a Data Source by moving to the **Next step**.

3. Provide the Required Beckon Parameters



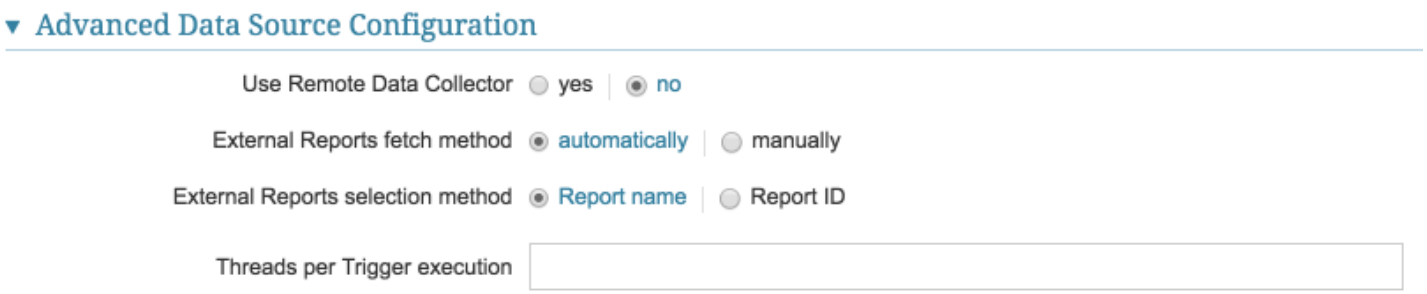
Specify how to connect to Beckon. The parameters include:

1. **Data Source Name:** Will default but you may modify it

2. Enter your **Api Key**
3. **Account URL**: if you have multiple sites, define the one you want to fetch data from
4. **Username / Password:** Note that your **Username** must be in the same format that your Beckon server uses for authentication (it may be an email address)
5. **Save** your entries and **Test Connection**

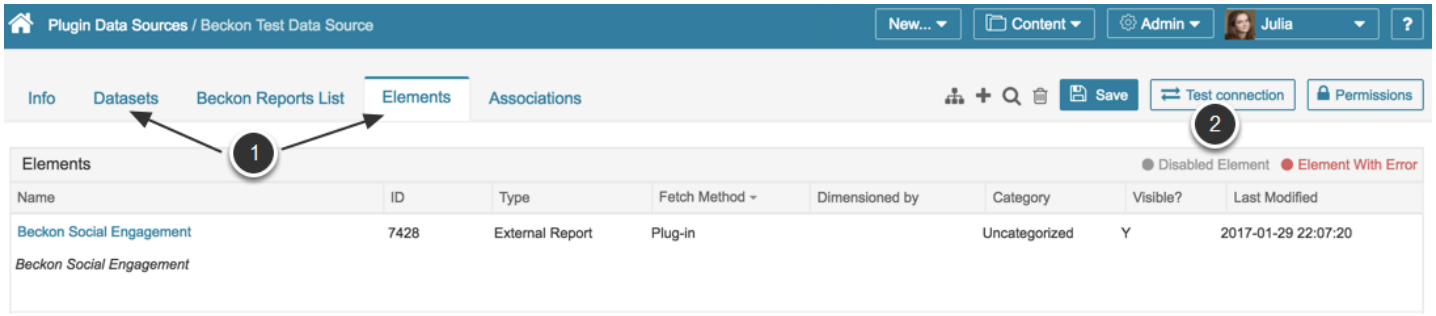
If your connection is successful, you may move on to **Advanced settings**.

4. Advanced Configuration



1. **External Reports fetch method**: This setting influences options available in the *Beckon Report List* tab:
   - **automatically**: just click **Refresh list** and all Reports are going to be fetched by the system
   - **manually**: Reports may be added one-by-one or via CSV file
2. Optionally, specify the maximum number of concurrent **Threads per Trigger execution** to be used in background processing when the system updates Metrics and Reports for this Data Source. If you do not specify any value for this setting, batch data collection processing will be single-threaded.

5. Other Settings



1. You can create elements or Datasets directly from the respective tabs
2. Click **Permissions** to assign them to Groups or Power Users

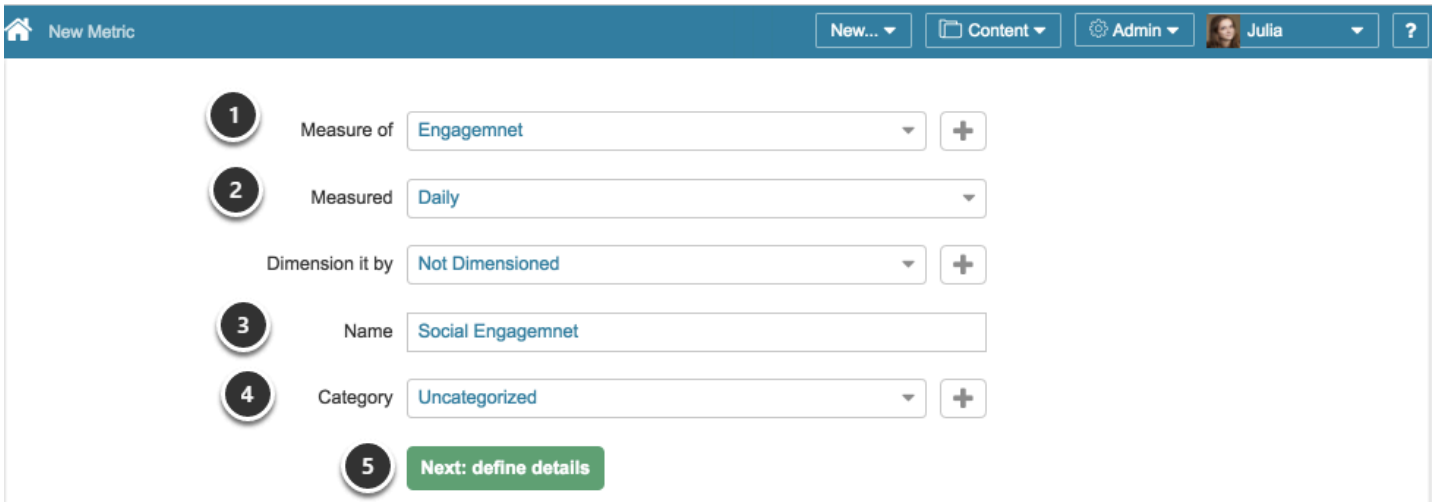
What's next?

How to collect data from Beckon

11.2 Collect data from Beckon

This article demonstrates how to create a Metric or Report using a Beckon as a data source. It assumes that you have already established connectivity to Beckon Server.

1. Access New > Metric

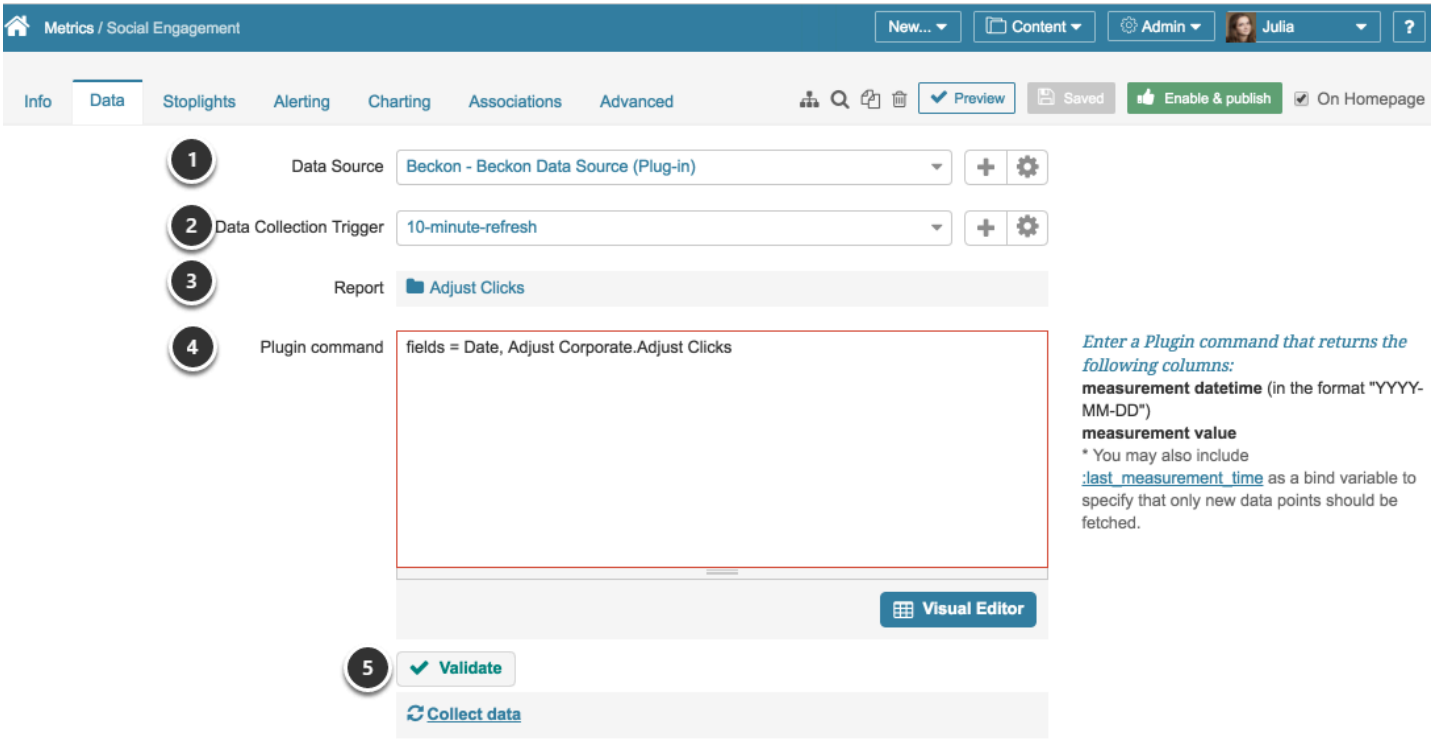


Provide the basic information required for creating a new metric:

1. Define this Metric's **Measure**. If you do not see the measure that you want to use, you can create one directly from the bottom of this drop-down list
2. Select the **Measurement Interval** that applies to your element
3. Give the element a unique **Name**
4. Optionally, assign a **Category**
5. Click **Next: define details**

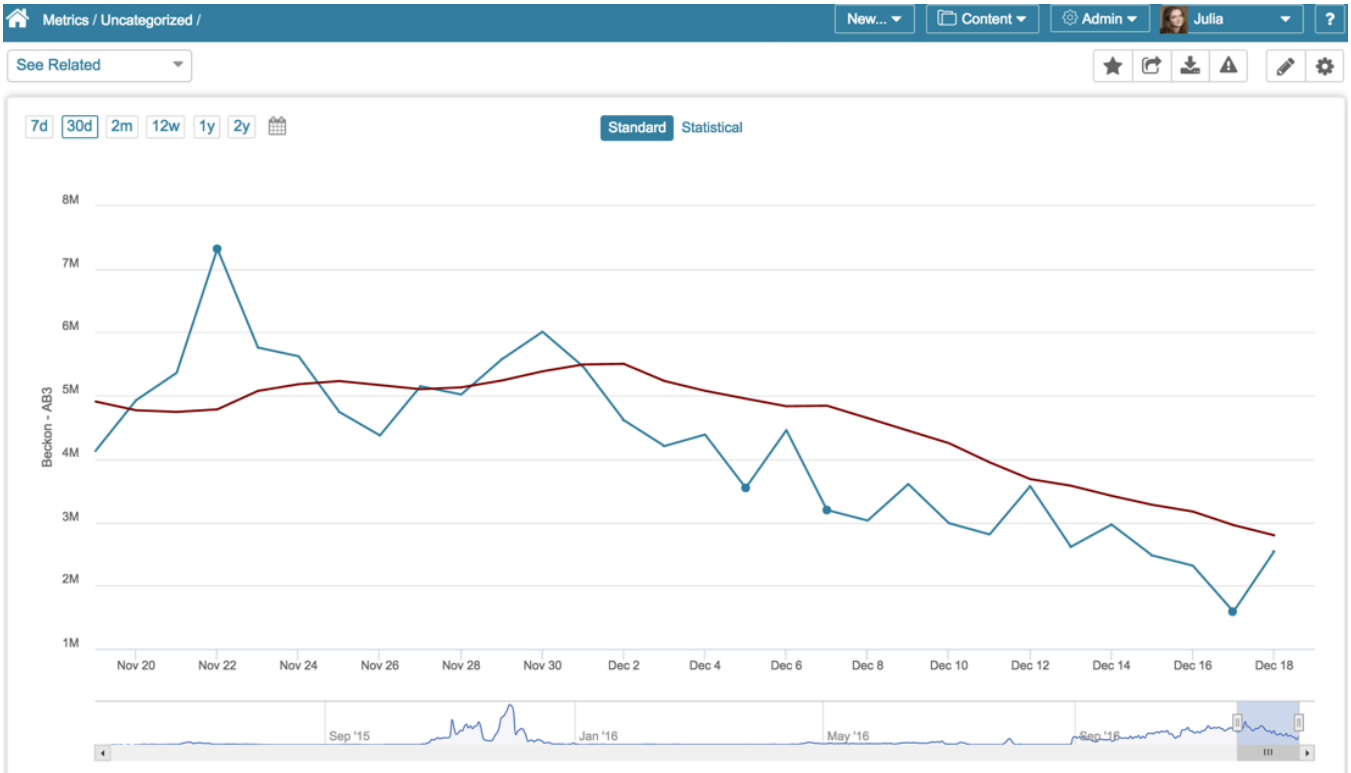
NOTE: To create a dimensioned Metric, you first need to create a Dimension sourced from the same data source.

2. Configure Data Collection



1. Select the **Beckon** plug-in serving as a **Data Source** for this Metric
2. Set the **Data Collection Trigger** which is going to initiate updating information in a Metric
3. Select a **Beckon Report** (workbook) from drop-down list.
4. Input **Plugin Command** manually (you may reference a table with parameters below) or use a **Visual Editor**
5. **Validate** your command. If your statement is valid, the statement box is **green**; if there are any errors, the box is colored in **red** and errors will be explained in the field below.
6. **Collect Data** and **Enable and Publish**

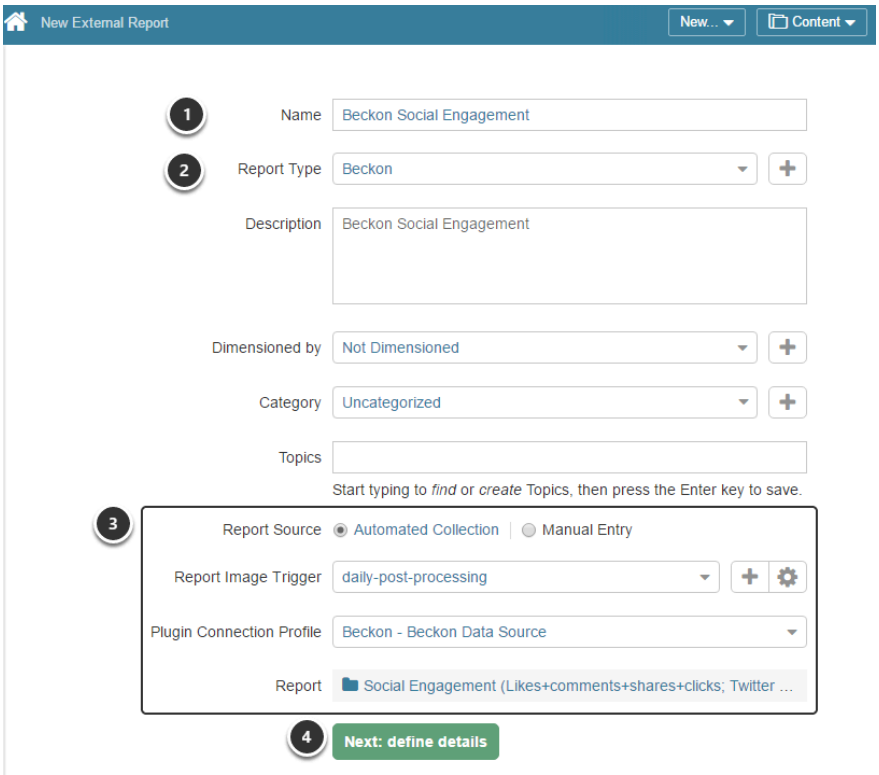
Result


Metrics / Uncategorized /
New...
Content
Admin
Julia
See Related
7d
30d
2m
12w
1y
2y
Standard
Statistical
8M
7M
6M
5M
4M
3M
2M
1M
Beckon - AB3
Nov 20
Nov 22
Nov 24
Nov 26
Nov 28
Nov 30
Dec 2
Dec 4
Dec 6
Dec 8
Dec 10
Dec 12
Dec 14
Dec 16
Dec 18
Sep '15
Jan '16
May '16
Sep '16

11.3 Create a External Report from Beckon

This article will show you how to create an External Report that is linked to a report from your Sisense server and is based on the assumption that you have already [established connectivity] to Beckon Server.

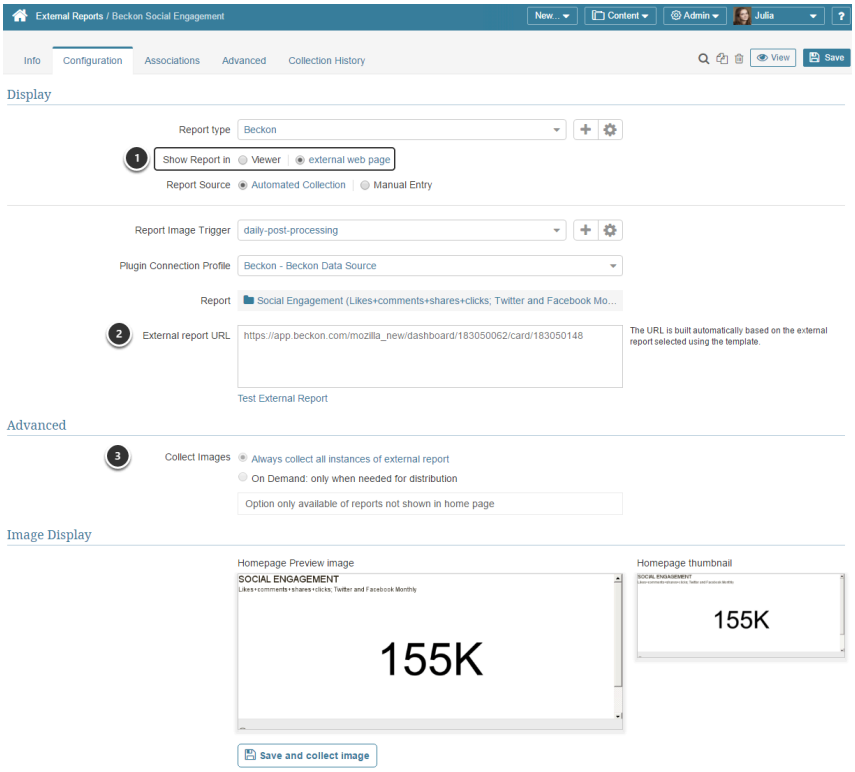
1. Access New > External Report > Beckon



The *New External Report* screen opens. Provide the following information:

1. Give your new External Report a **Name**
2. **Report Type:** If there is no required Report Type in the list, click the Plus (+) button and create a new one.
3. Define whether you want to report content to be updated manually or automatically. In case you choose **Automatic Collection**, define the following settings:
   - Define the **Report Image Trigger** from drop-down list
   - Select the **Plugin Connection Profile** you have created for Sisense. For more details refer to [Establish connectivity to Beckon]
   - **Report:** Select a required workbook available from the selected connection profile
4. Click **Next: define details** to proceed with Report creation.

2. Additional Settings



1. **Viewer**: The report sourced from Beckon is typically shown at the source page, so make sure to set this field to "external page".
2. The **External Report URL** will be generated automatically based on your other inputs. If you like, you can modify the URL by appending a question mark (?) followed by any filter or parameter settings
3. **Collect Image**:
   - **Always collect all instances of external report**: Collect all images and cache them on a schedule.
   - **On Demand: only when needed for distribution**: Individual images are only collected when they need to be included in an email.

4. **Enable and Publish**

12. Sourcing Data from Cognos

12.1 Establish connectivity to IBM Cognos

This article describes how to connect to a **IBM Cognos** server and pull in its data and images to Metric Insights.

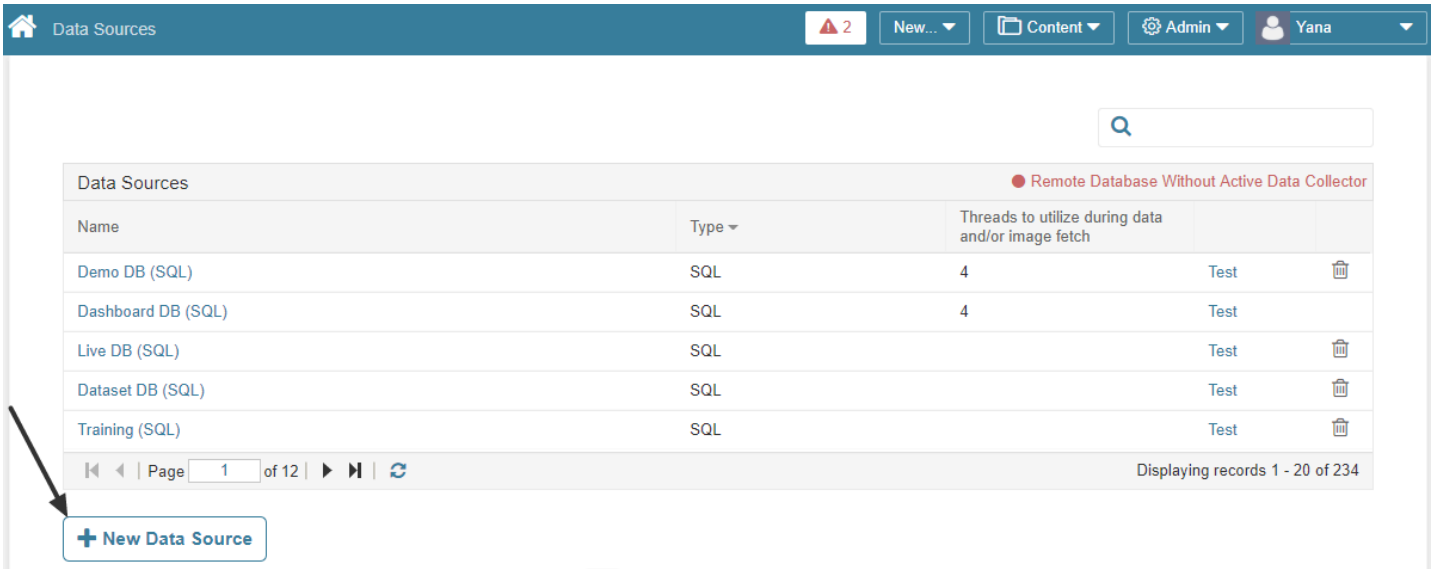
General instructions on setting up data sources based on plugins can be found here.

PREREQUISITES

- The default port for Cognos is 9300. Make sure you open port 9300 to access the Cognos server from the Metric Insights server.

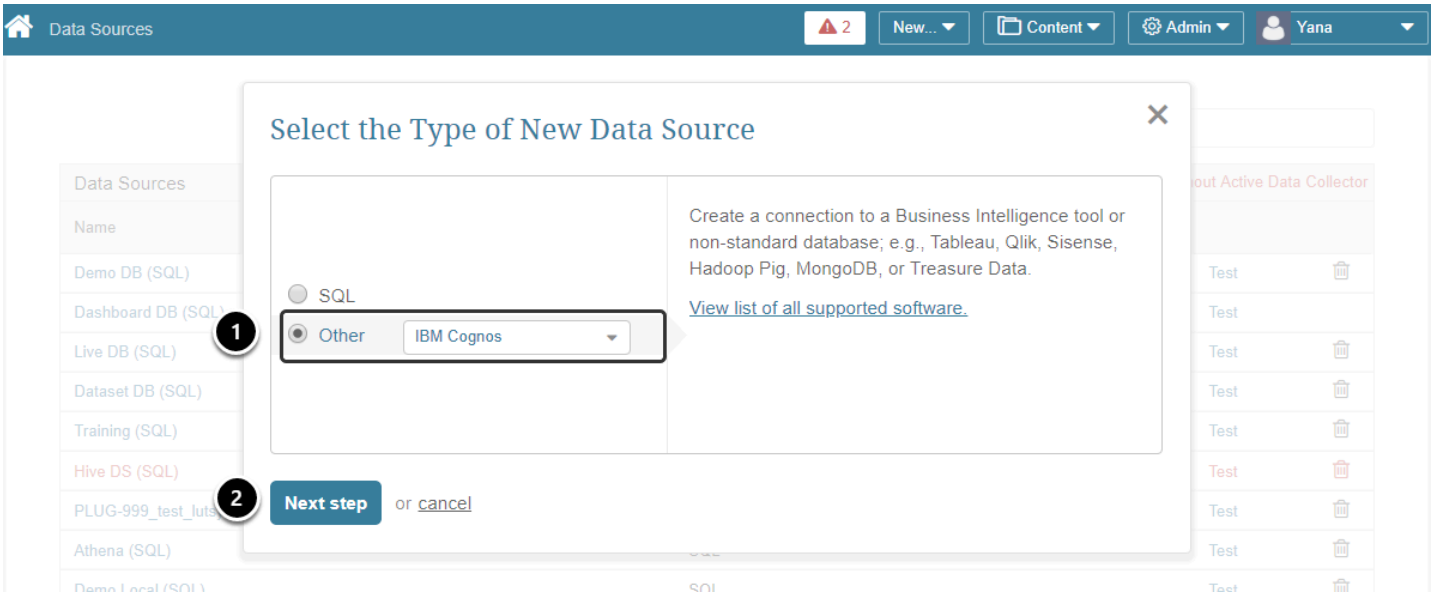
[Release 6.x]: 'Remote Data Collector' renamed to 'Remote Data Processor'

1. Access Admin > Data Sources



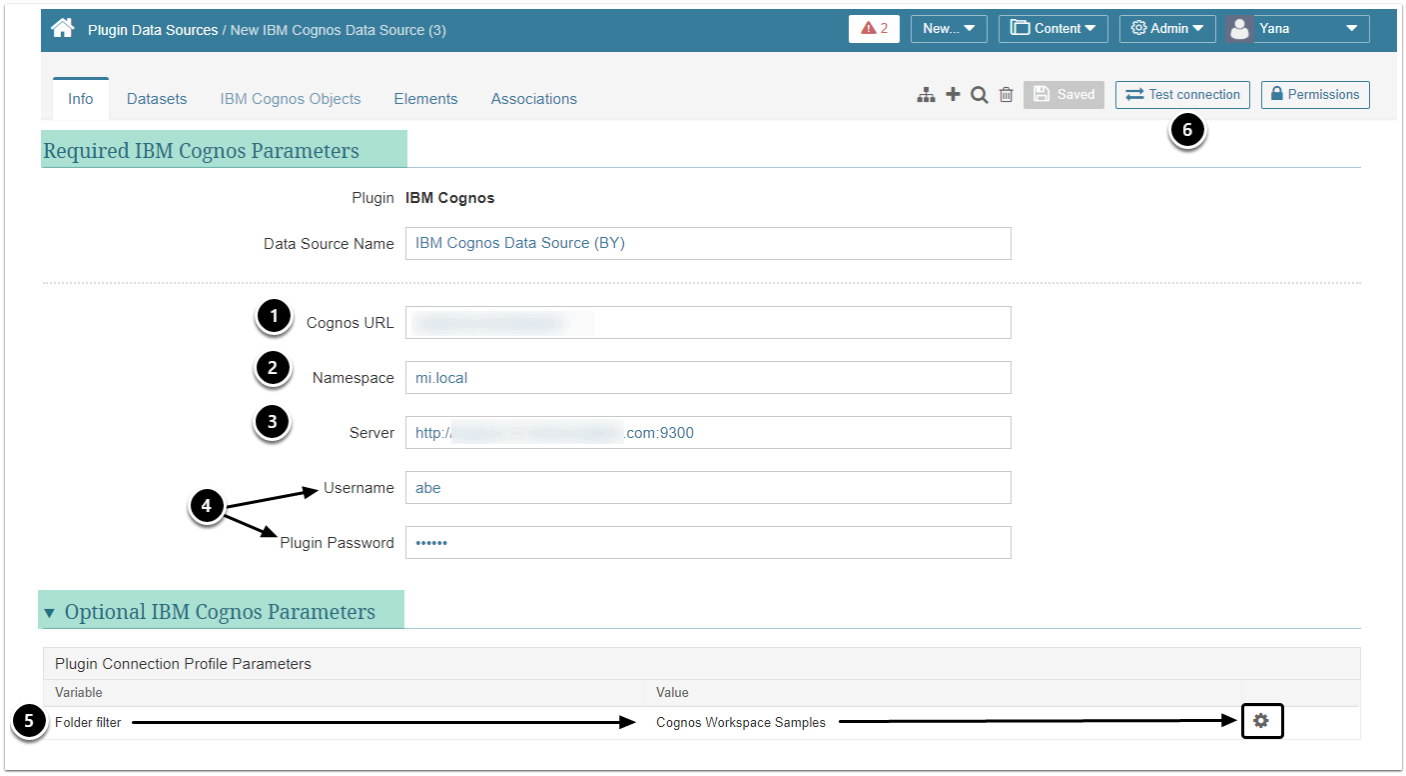
At the bottom of the screen, click **[+ New Data Source].**

2. Select the Type of New Data Source



1. Select **"Other"** Data Source Type and choose **"IBM Cognos"** from the drop-down list.
2. Move to the **Next step.**

3. Provide Cognos Parameters

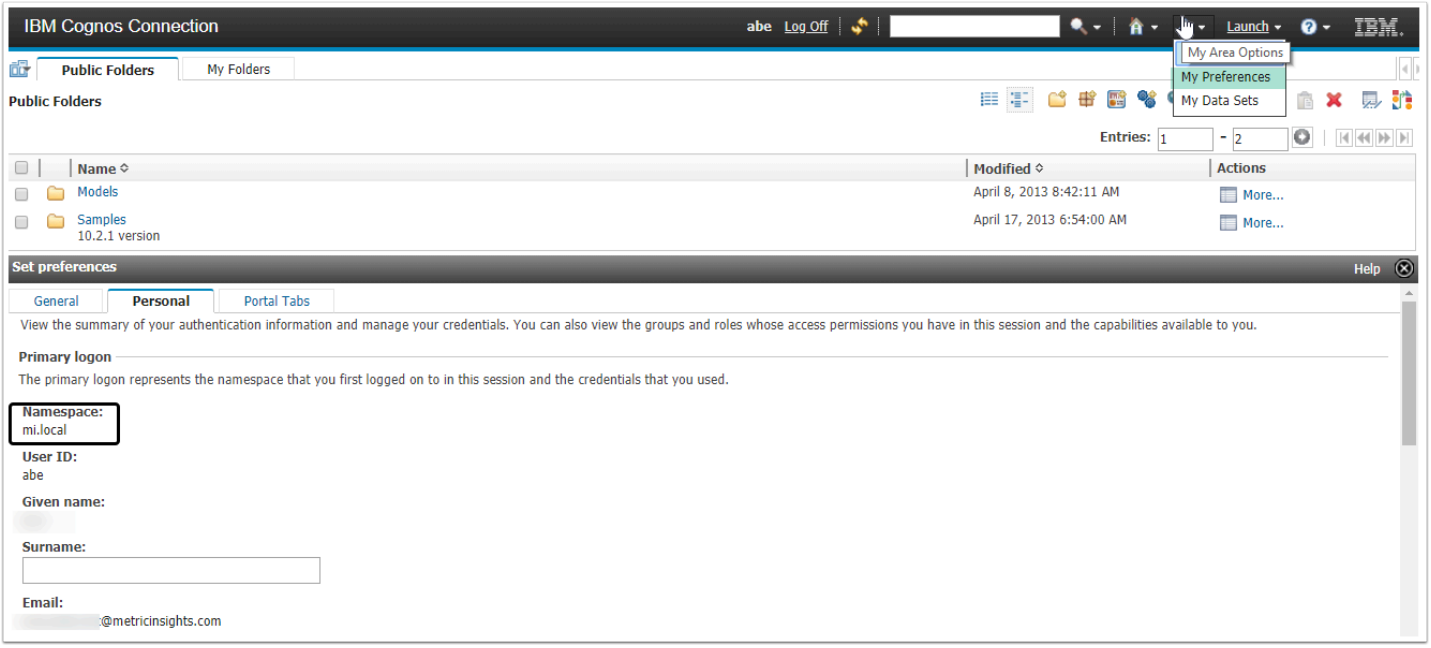


1. **Cognos URL:** provide an endpoint for reaching a login page at the Cognos server

2. A path to finding the **Namespace** value is defined in Step 3.1
3. **Server:** define the server protocol (**http** or **https**) and a hostname
4. **Username / Password:** note that your **Username** must be in the same format that your IBM Cognos server uses for authentication
5. Optionally, add a **Folder filter** Value that will be used for filtering IBM Cognos Objects
   - To do so, click the **Edit (Gear) icon** in the corresponding field
6. **Save** your entries and **Test Connection**

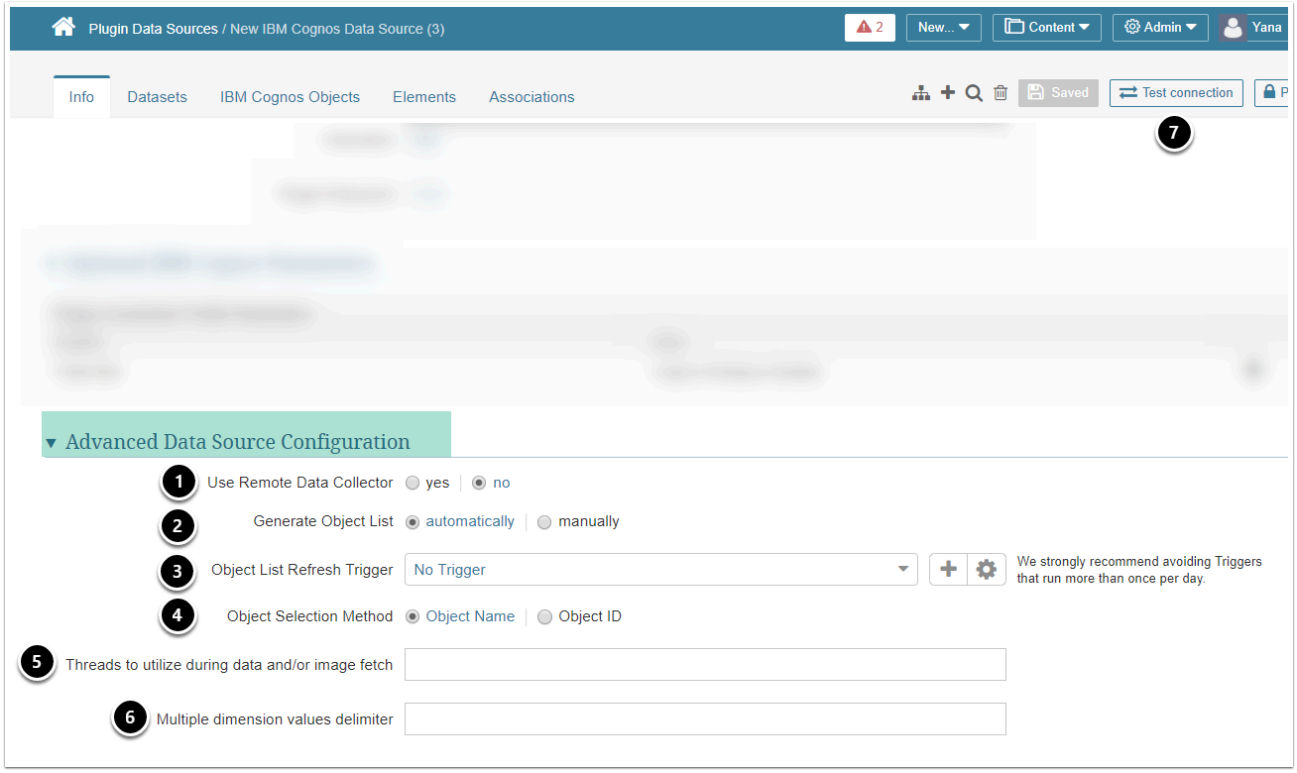
If your connection is successful, you may move on to **Advanced settings**.

3.1. About Namespace



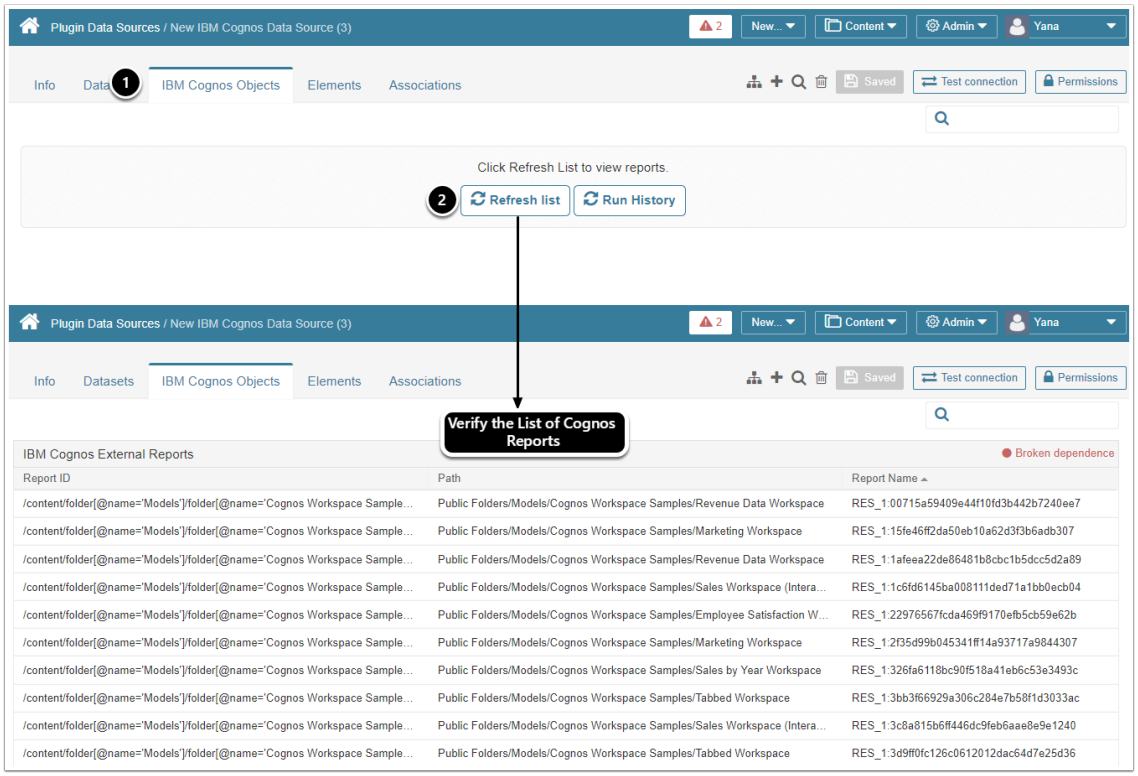
At the Cognos server go to **My Preferences** > *Set Preferences* screen opens > *Personal* tab.

4. Advanced Settings



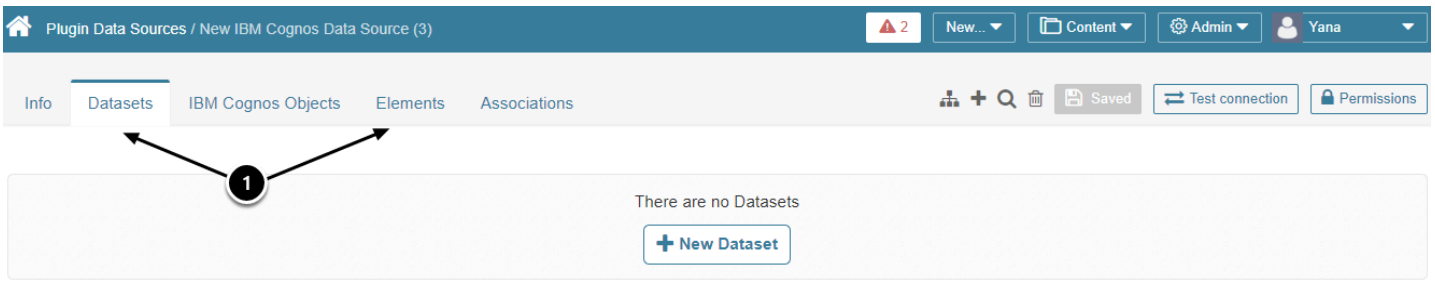
1. **Use Remote Data Collector:** if required, set to "yes"
2. **Generate Object List**
   - *automatically*: all Reports are going to be fetched by the system
   - *manually*: Reports may be added one-by-one or via CSV file
3. **Object List Refresh Trigger:** from the dropdown, select the Trigger that will be used to fetch data via the IBM Cognos plugin
4. **Object Selection Method:** specify how *IBM Cognos Reports* will be fetched
5. Optionally, state the number of **Threads to utilize during data and/or image fetch** to be used in background processing when the system updates Reports for this Data Source
   - If you do not specify any value for this setting, batch data collection processing will be single-threaded
6. **Multiple dimension values delimiter:** optionally, specify the Delimiter for Multiple Dimension Values (Configuring this parameter allows to support fetching aggregate data for several IBM Cognos Filter Values)
7. **Test Connection** (this will also **Save** your data)

5. Obtain a list of Cognos Reports



1. Go to **IBM Cognos Objects** tab
2. To obtain a list of IBM Cognos Reports, click the **[Refresh list]** button
   - If no Folder filter has been specified for Optional Cognos Parameters (as described in Step 3), all Cognos Reports will be fetched into Metric Insights
   - Otherwise, only those Reports that satisfy the Filter will be collected for this connection profile

6. Other Settings



1. You can create Datasets or Elements directly from the respective tabs.
2. Click [**Permissions]** to assign Permission to Groups or Power Users.

What's next?

How to collect data from Cognos

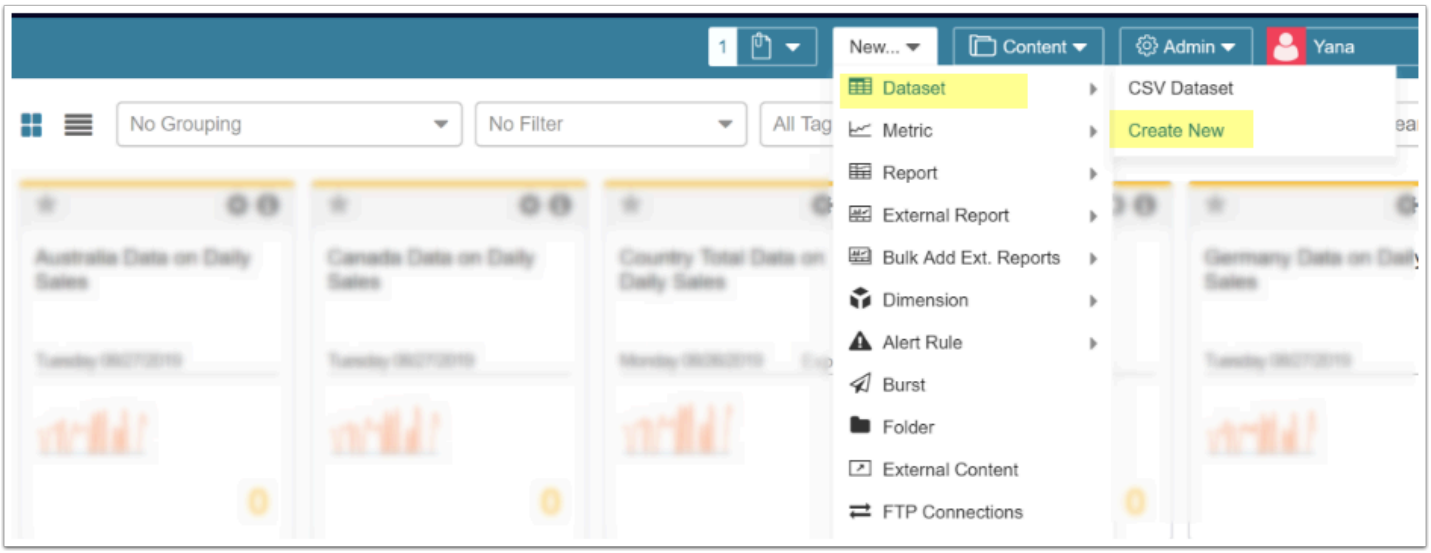
12.2 Collect data from IBM Cognos

A Metric Insights' Dataset can be populated automatically based on data fetched from IBM Cognos.

PREREQUISITE:

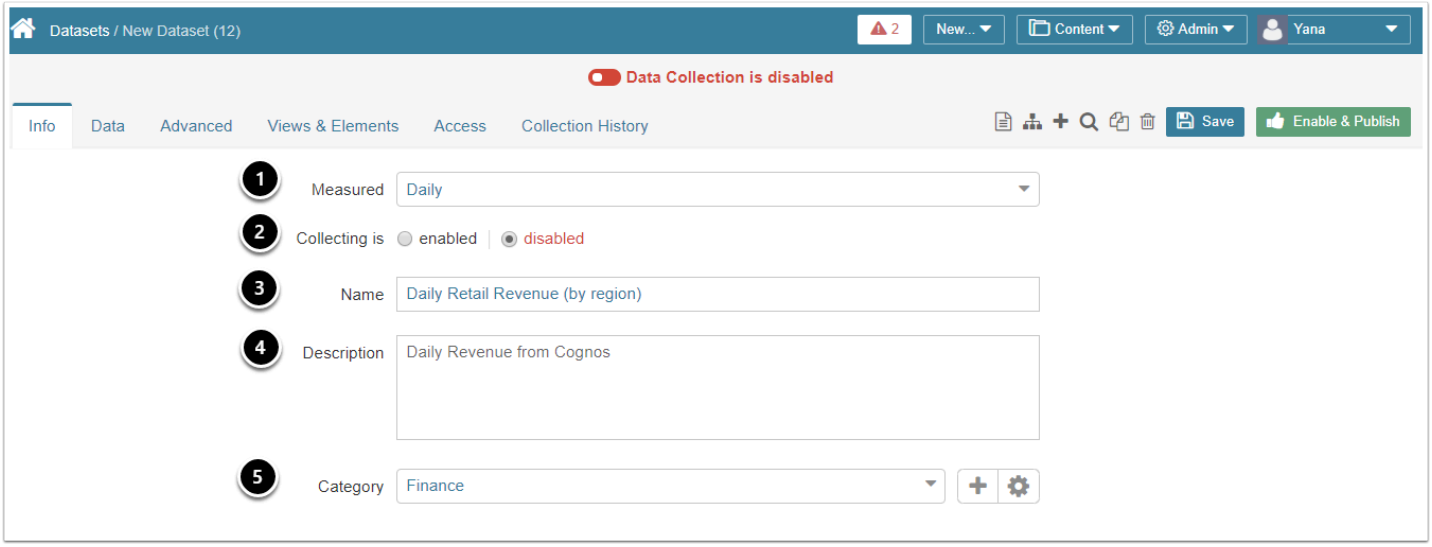
You must have already established connectivity to your **Cognos** server via the respective plugin connection profile.

1. Access New > Datasets > Create New



You will be redirected to the Dataset Editor.

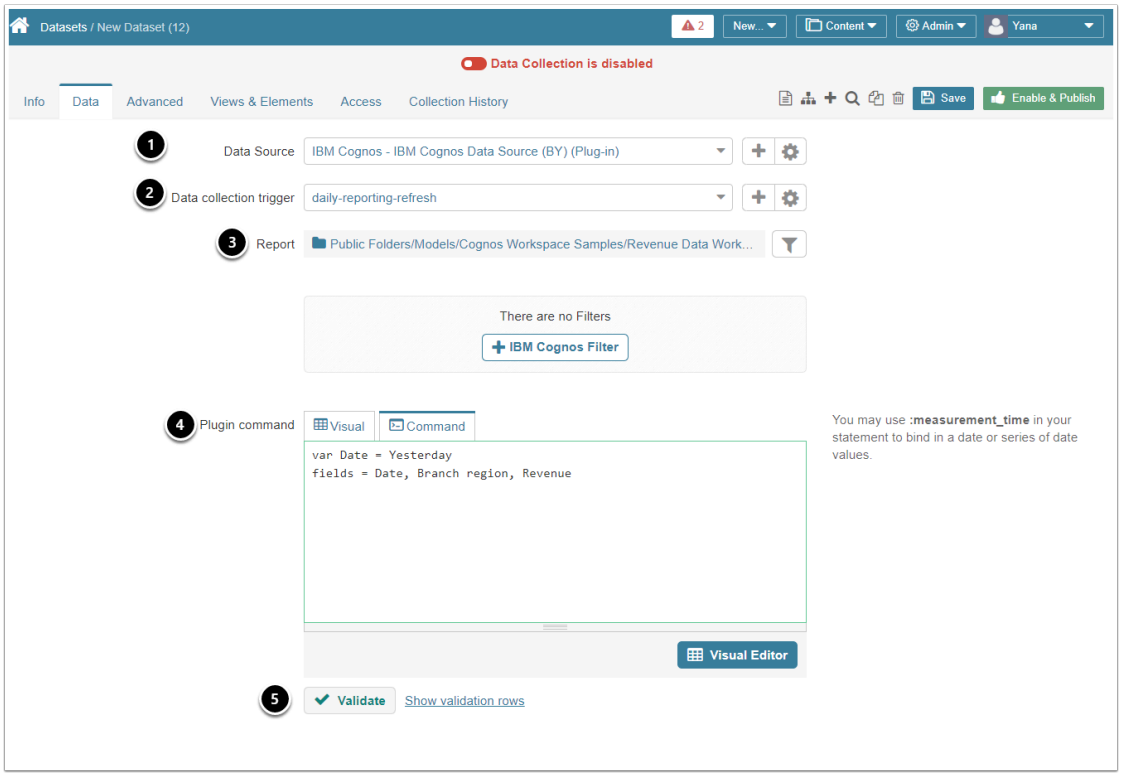
2. Dataset Editor > Info tab



1. **Measured:** select the measurement interval that applies to the level of aggregation that you want in your result set.
2. **Collecting**: new Datasets are always disabled by default to make sure that you can take time to configure them properly before enabling. This setting is duplicated at the top of the screen.
3. **Name:** provide a unique name for your Dataset. Preferably, the Dataset name should explain what kind of data it contains.
4. **Description:** optionally, provide any additional information about your Dataset.
5. **Category:** specify the Category where you Dataset will be placed.

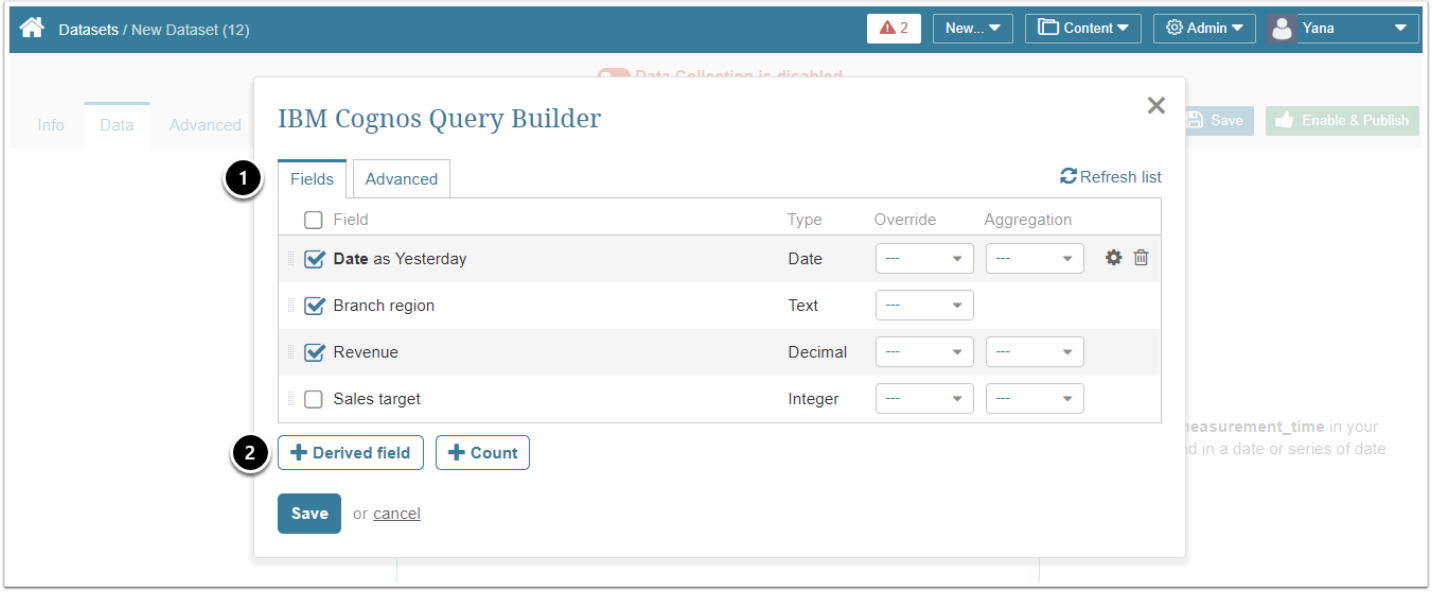
Move to the ***Data tab*** to define the source of data and how often it should be updated.

3. Define the Settings for Data Collection



1. **Data Source**: select the connection profile you have created for IBM Cognos
2. **Data collection trigger**: specify the Trigger that will be used to collect data for your Dataset
3. **Report:** select an external IBM Cognos Report that should serve as a basis of your Dataset
4. Input a **Plugin Command** listing all the data you would like to fetch from *IBM Cognos*
   - Build your query in MIQL syntax
   - Alternatively, use the **Visual Editor**
5. Once you are ready with you command, click **Validate**

3.1. Example using the Visual Editor

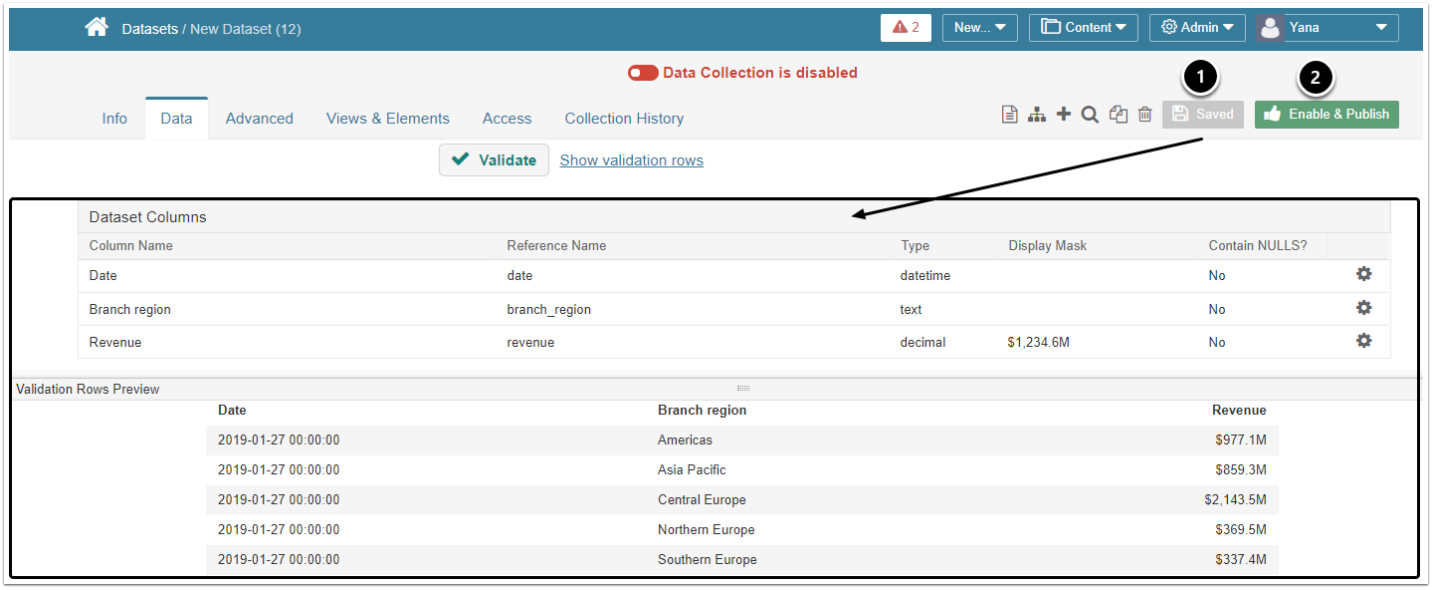


The ***IBM Cognos Query Builder*** allows constructing commands without the need to learn the plugin syntax and avoiding typos/mistakes.

1. Select the **fields** for your Dataset
2. *Optionally,* add **Derived fields** and/or **Count** of duplicate rows

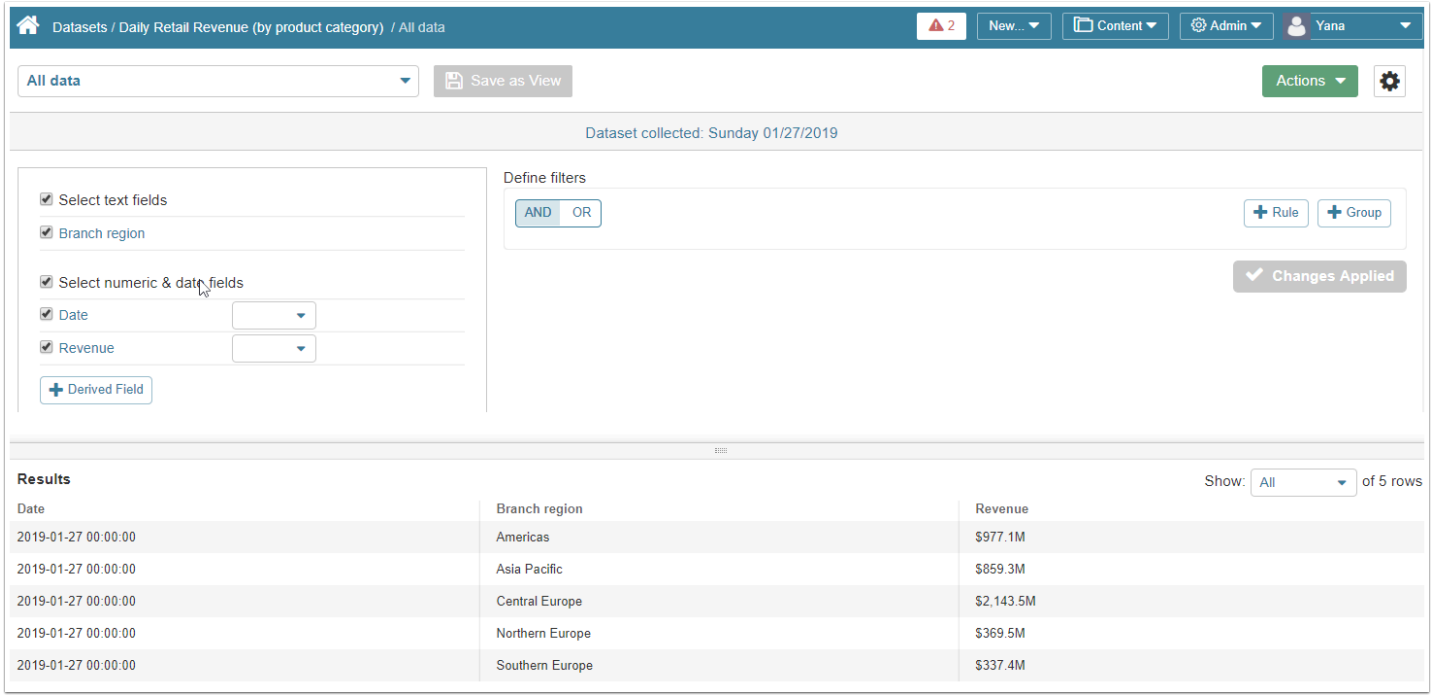
Save your settings. Plugin command validation will start automatically.

4. Plugin command will be validated and data collected on Save



1. If the command is validated successfully, the **Dataset columns** and **Data Preview** are going to be shown below.
2. At the upper right corner of the screen click **Enable & publish**.

5. Dataset will be displayed in Viewer



6. What's next?

[Create a Dataset Report]

12.3 Pre-filtering Cognos data

When sourcing data for Datasets, Metrics, Reports, External Reports and Dimensions from IBM Cognos, you can pre-filter data before fetching it. This allows you to focus on the slice of data you need and exclude everything else.

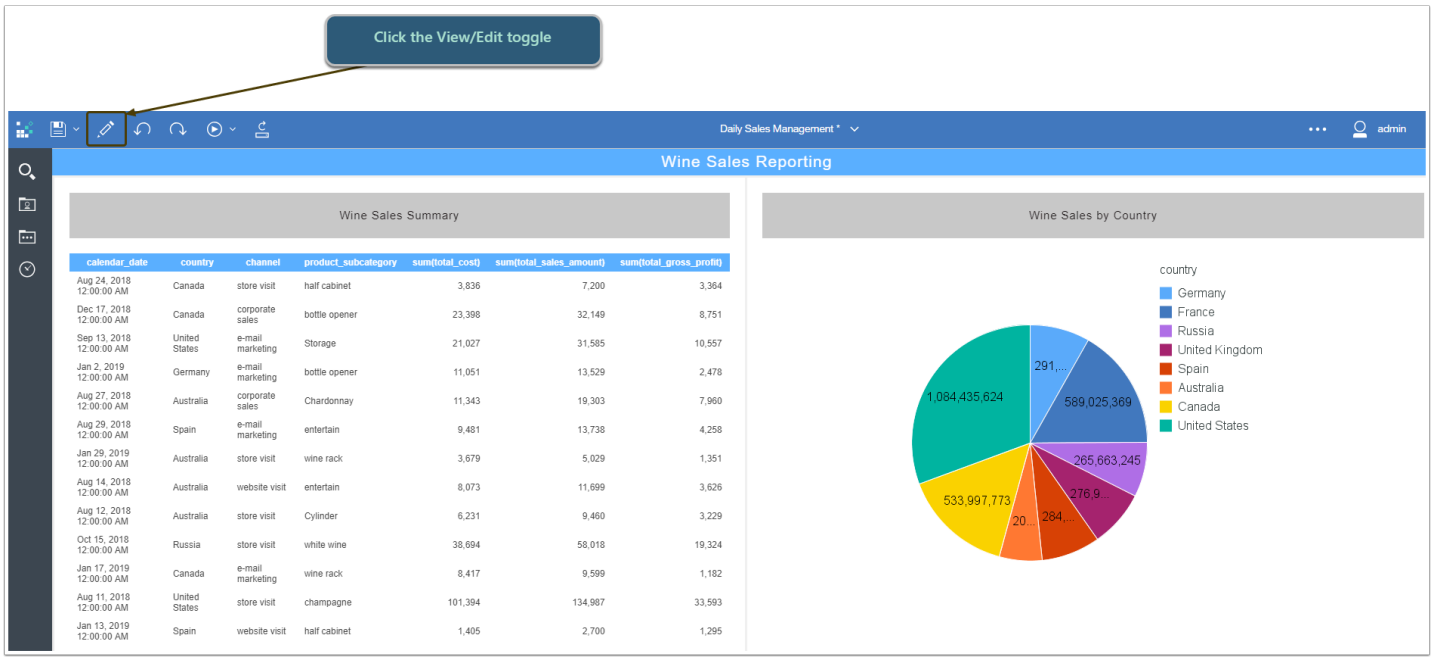
PREREQUISITES:

Establish connectivity to Cognos

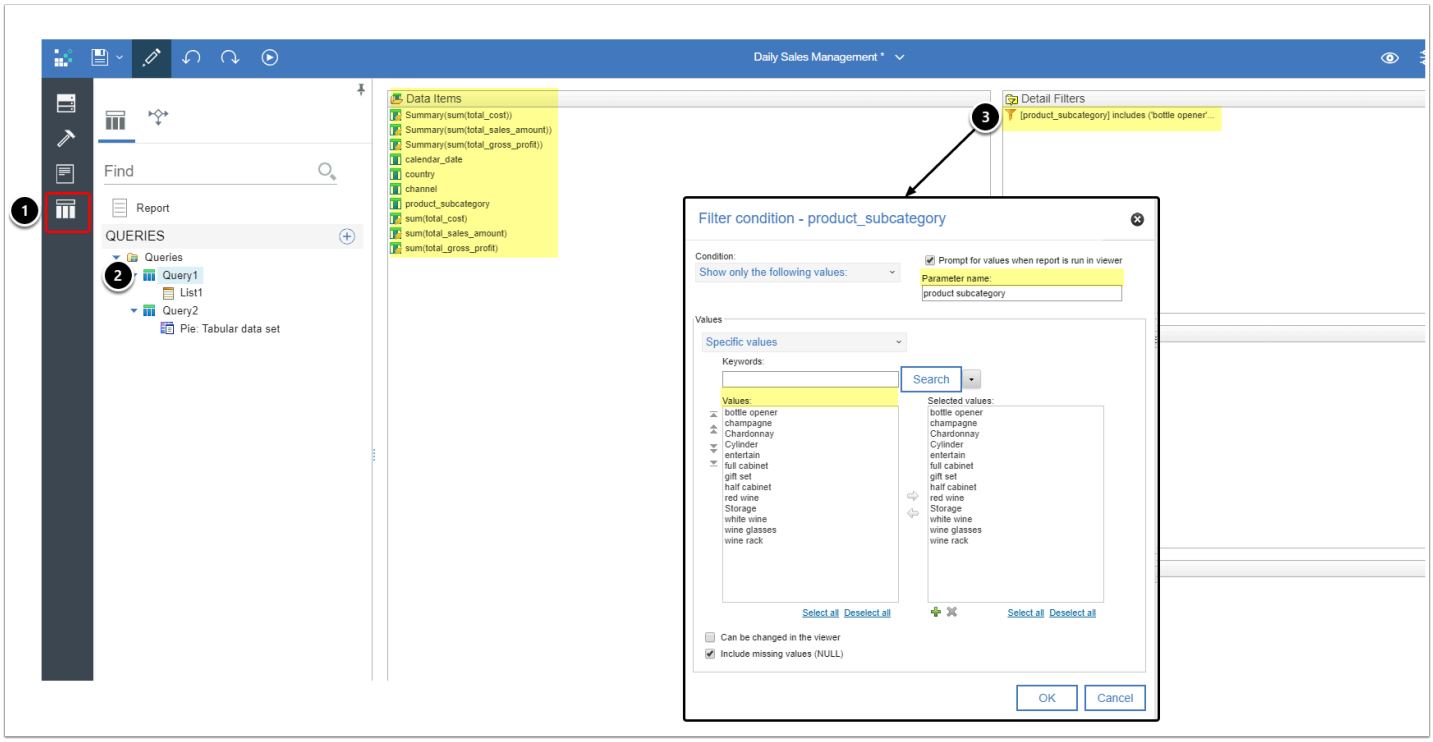
How to find Filter names in IBM Cognos?

Parameter-based Filters can be added to Cognos Reports and Visualizations.

Below there is an example of a Report in **View mode**. To get Parameter Names and Values for Filters in Cognos Reports, you need to switch to **Edit mode** with the corresponding Toggle.

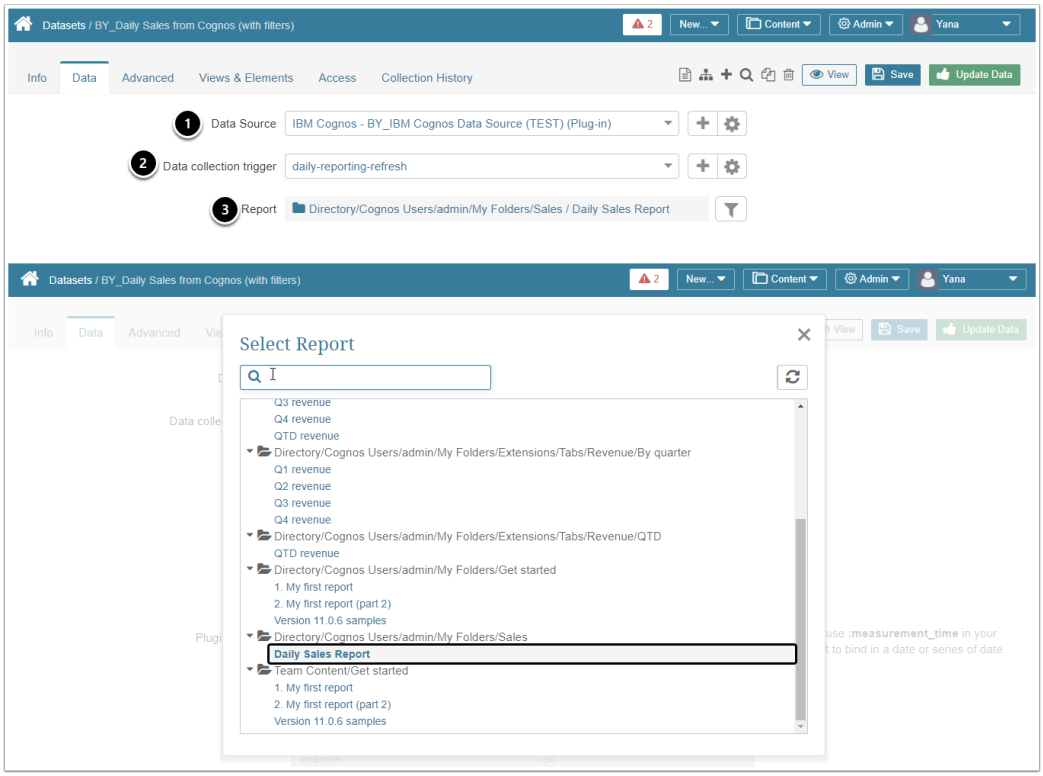


When in the **Edit mode**, follow the steps described below.



1. Click the **Data Items** icon to open a list of saved **Queries**
2. Selecting the **Query** will open the Data Items (Report contents) and display the **Filter**
3. Click the **Filter name**, and on the *"Filter condition" pop-up* find:
   - ***Parameter name:*** must be used as an External Filter Name in Metric Insights
   - ***Values:*** when manually adding External Filter Values in Metric Insights (as described in Step 2.1), copy the Filter Values from here

1. Define a Source Object for Data Collection



Start by creating an element/Dataset. Once you get to the process of Data Collection, define the following:

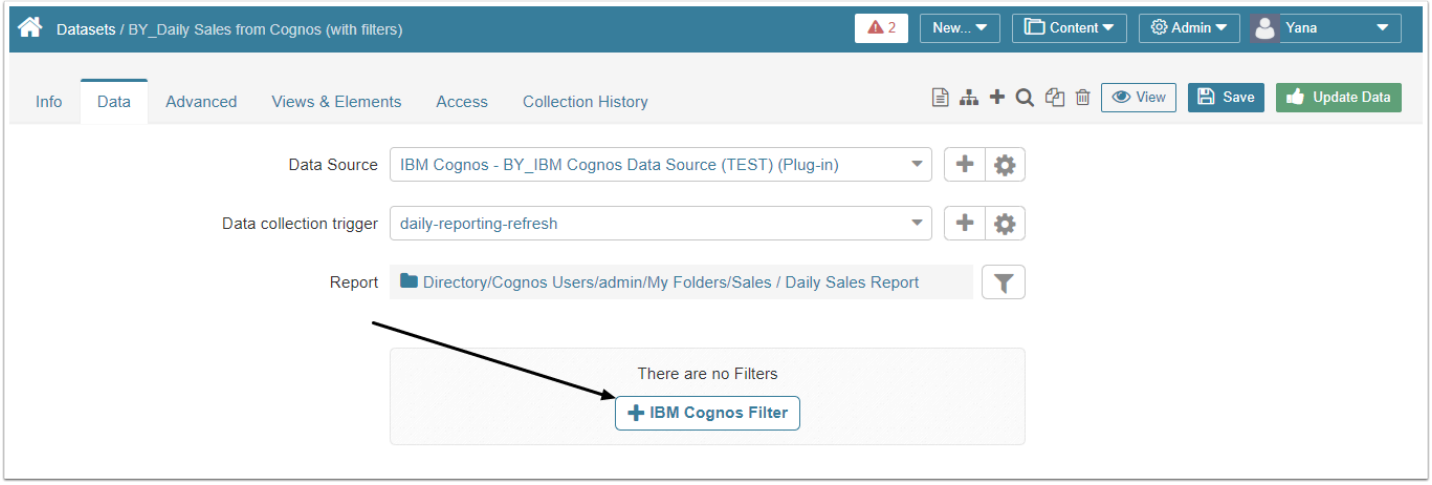
1. **Data Source:** This is an entity that connects Cognos and Metric Insights. For more information, see: [Establish connectivity to Cognos]
2. **Data Collection Trigger:** select the Trigger that is going to initiate updating information in this element/Dataset.
3. **Report:** Click **Select Element** to open the pop-up with the list of available Cognos Reports that can be a source of data.
4. Each item in the list is represented as the path (hierarchy) to the respective Report in Cognos. Find the desired report in the list.
5. If you do not see the required item, try refreshing the list by clicking the **Refresh** icon at the upper right corner of the pop-up.

2. Adding Cognos Filters to Metric Insights

> ⚠ Once filters are added to a Metric / Report or External Report for the first time, they are going to be automatically added to all new respective elements with the same Data Source / Report.

NOTE:

- External filters are tied to /Cognos Reports, not Metric Insights' elements. This allows Filters to be reused for multiple elements (there is no need to create new Filters every time an element is created in Metric Insights).
- If there are more External Filters or Filter Values that you would like to use for the current element, you can always set the redundant ones to "ignore".

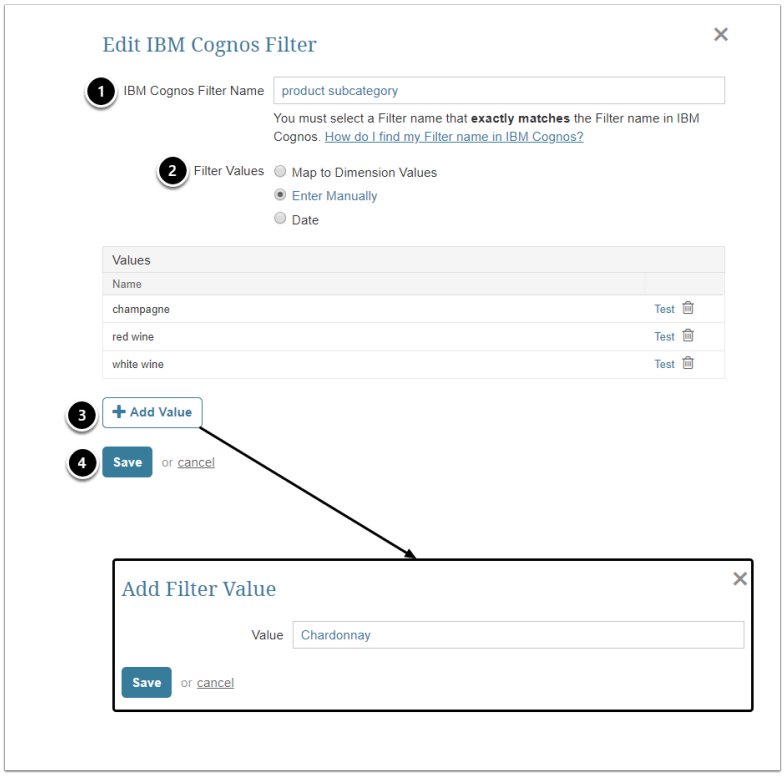


When creating a Metric / Report / External Report / Dataset fetched from Cognos, after you define the **Report** that should serve as a Data Source, you may pre-filter information that is going to be fetched.

To pre-filter the data, click **[+IBM Cognos Filter].** The following pop-up will give you 3 options to add Filters.

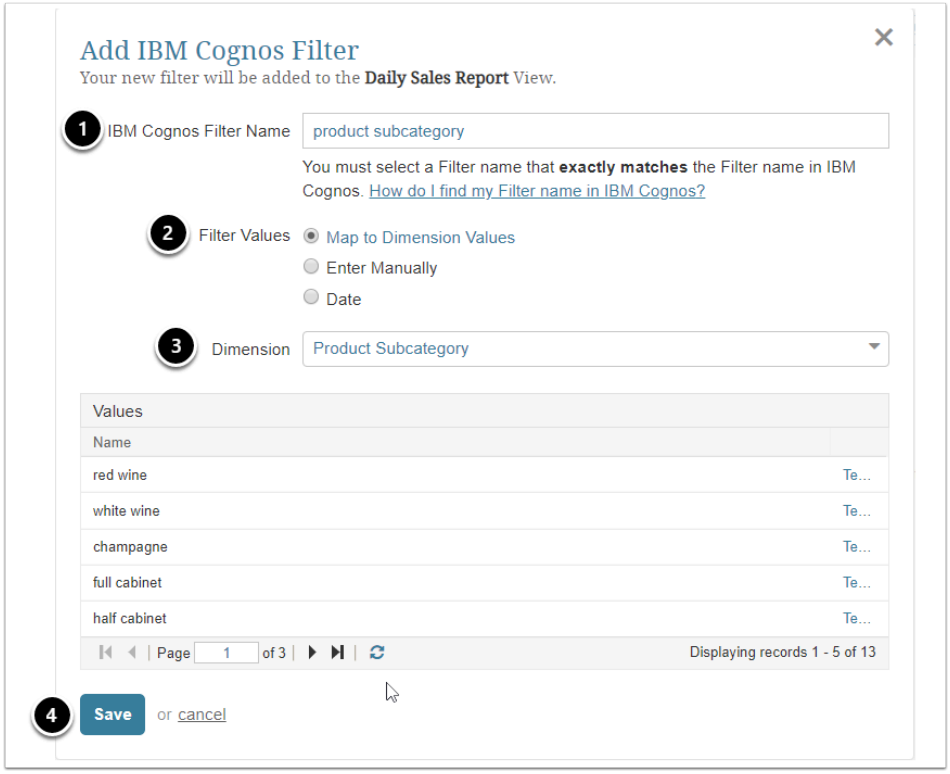
NOTE: Examples given below are taken from the Cognos Report shown at the top of the page.

2.1. Enter manually



1. **Cognos Filter Name**: Define the name of the filter from IBM Cognos
   - The name of the Filter must exactly match the Parameter Name of a parameter-based Cognos filter
   - Filter names are case sensitive. Unless the match is exact, the Filter will not work
2. **Filter Values:** Choose 'Enter Manually' and click **Save** at the bottom of the pop-up.
3. Click **[+ Add Value]** and in the opened pop-up manually type in the name of the filter value. **Save** your entry. All added values appear in the *Values* list.
4. **Save** your entries.

2.2. Using dimension values

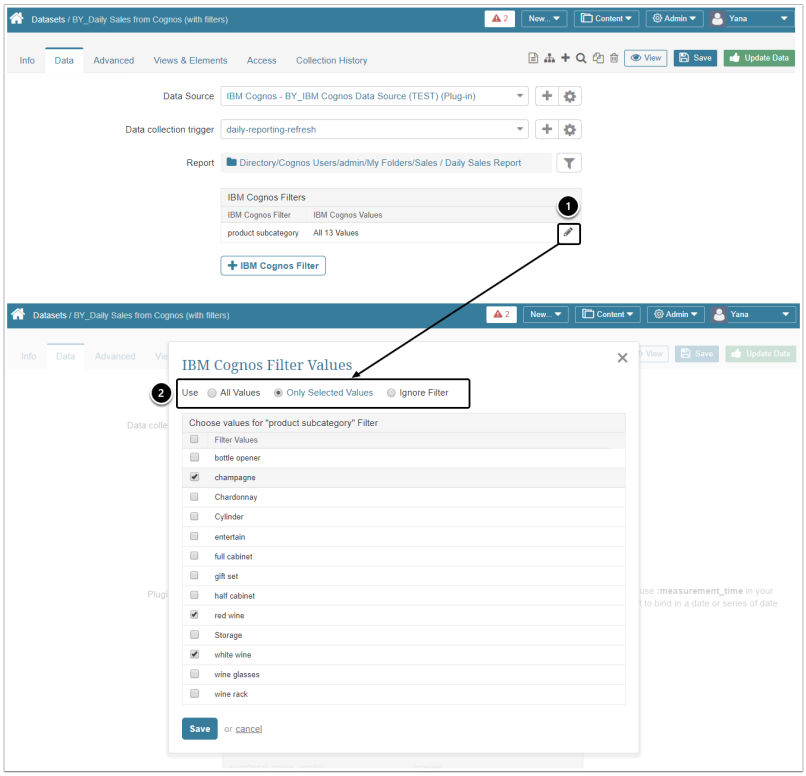


If you have already used Cognos filters to create Dimensions in Metric Insights, you can quickly choose which Dimension Values you want to use for pre-filtering:

1. **IBM Cognos Filter Name**: Define the name of the filter from Cognos.
2. **Filter Values:** choose 'Map to Dimension Values'.
3. **Dimension:** select a corresponding Dimension from the drop-down list and all its Values are going to be loaded to the Values list automatically.
4. **Save** your entry.

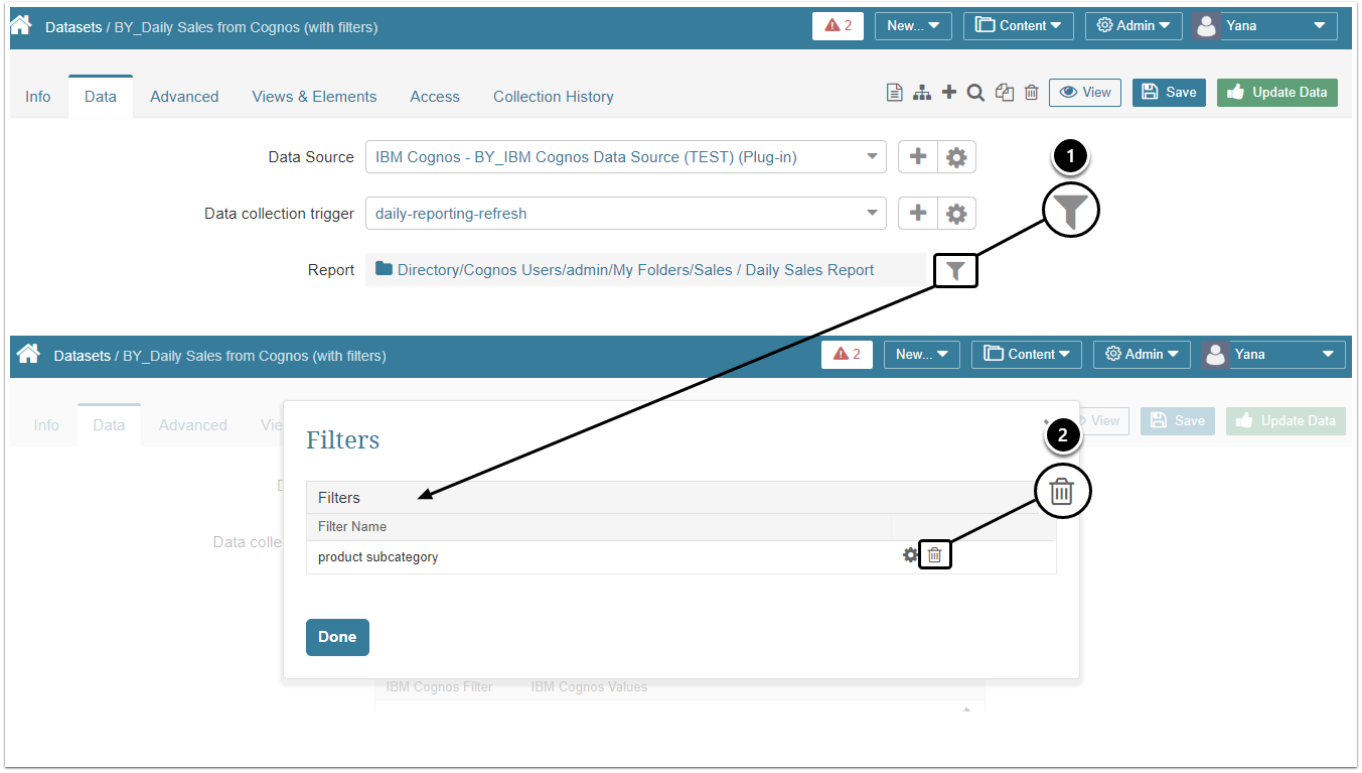
NOTE: Dimensions used here must have Values that exactly match the Filter Values in Cognos (if the Values do not match, the Filter will not work)

3. How do I add filters to a results set from IMB Cognos?



1. Click the **Pencil** icon in the filter row to add it.
2. When the filter is added, you can use it for "All Values", "Only Selected Values" or ignore it.

4. Deleting Filters



To delete some of the added filters: (1) click the **Filter** icon in the **Report** field and (2) choose the unnecessary filters. Click the **Trash bin** icon in the corresponding row.

13. Sourcing Data from CSV Plugin

13.1 Establish connectivity to a CSV file via a CSV Plugin

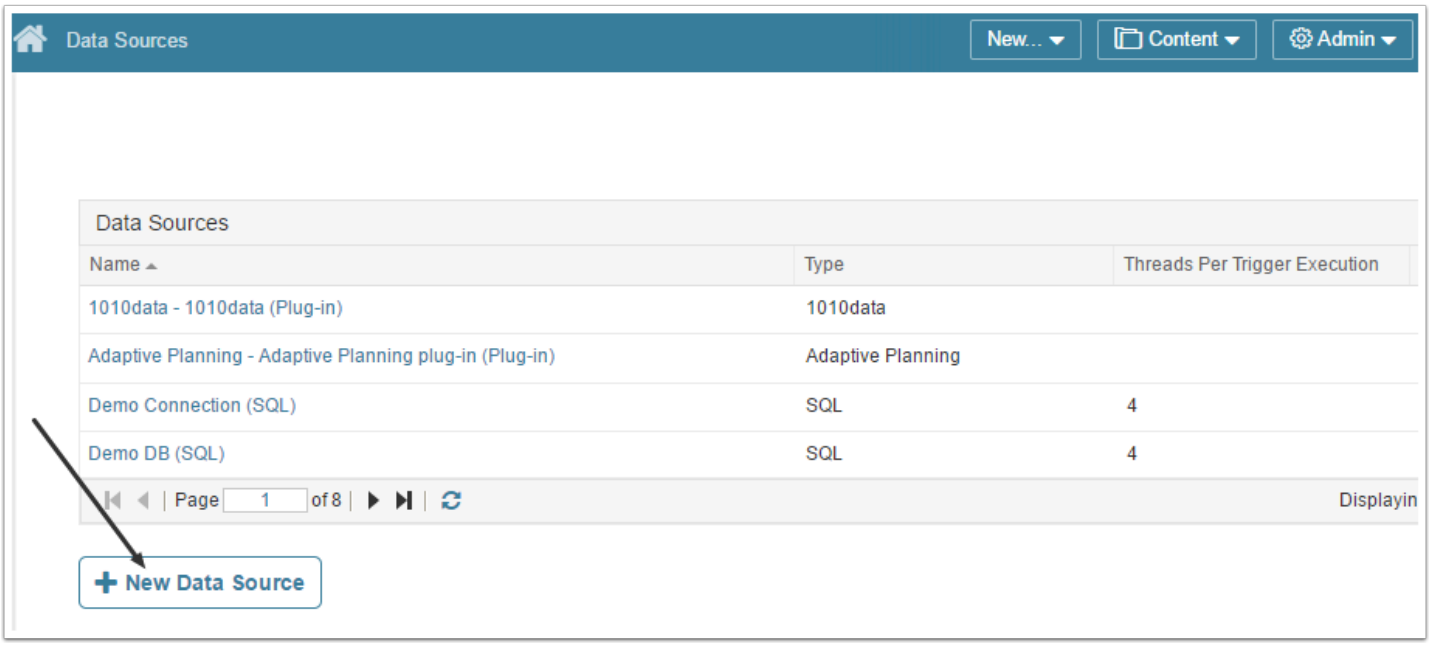
This article describes how to use a CSV plugin in order to create a connection profile with Metric Insights.

The CSV file which serves as a source of data can be located at:

- the local file system
- ftp (sftp, ftps (over ssh))
- s3 (Amazon file server)

General instructions on setting up data sources based on plug-ins can be found here.

1. Access Admin > Data Sources



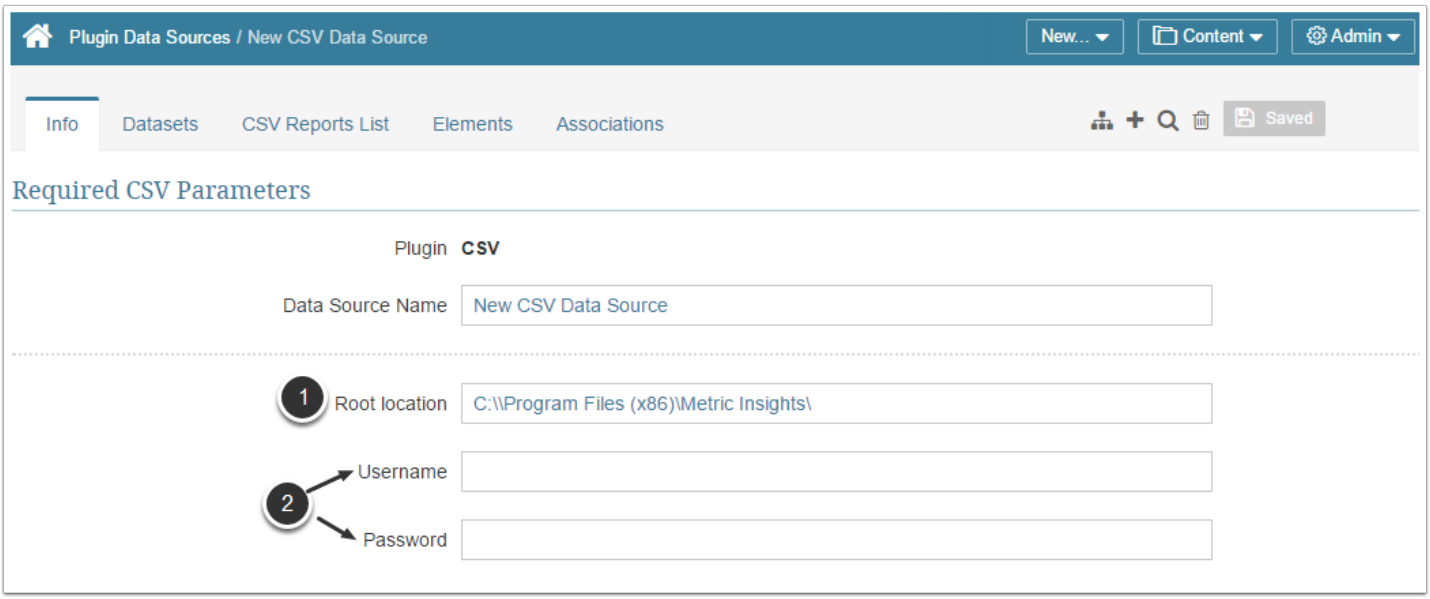
The list of data sources available in the system opens

At the bottom of the screen click **[+ New Data Source]**

2. Select "Other" Data Source Type and choose "CSV" from the drop-down

Move to the **Next step**.

3. Provide Required Parameters



1. **Root location:** provide the path to the directory where the CSV file is located. The available options are:

- local file system

```
/home/user/tmp
```

- ftp (sftp, ftps (over ssh))

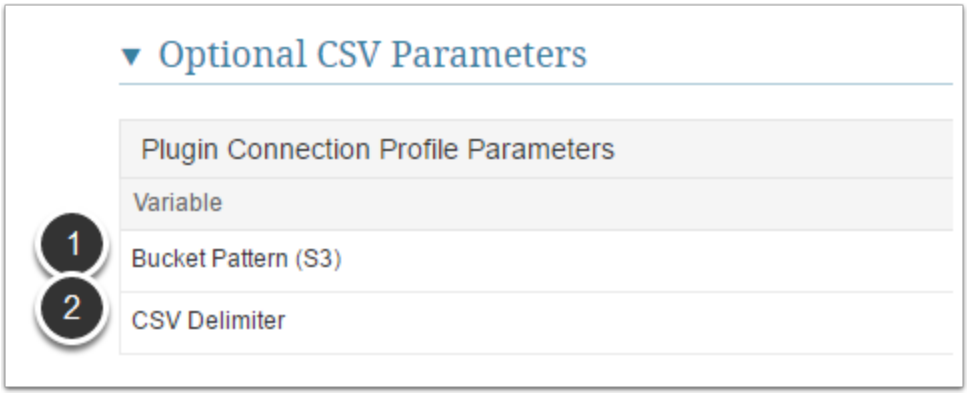
```
ftp://my-server.com/some/path
```

- s3 (Amazon file server)

```
s3://
s3://bucket
s3://bucket/some/path
```

2. **Username / Password**: provide authentication credentials to access the root location.

Optional CSV Parameters

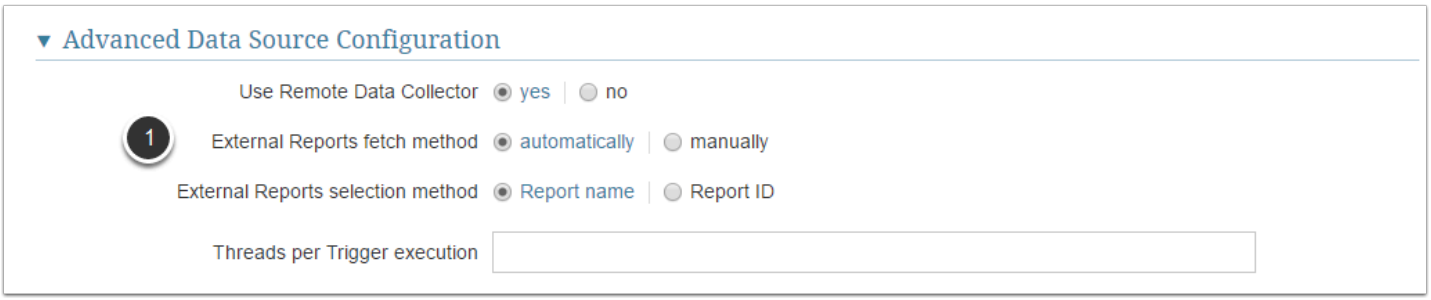


1. Bucket Pattern (C3): *A setting specific to Amazon server.*

```
pref*, dasd*, 1*2
```

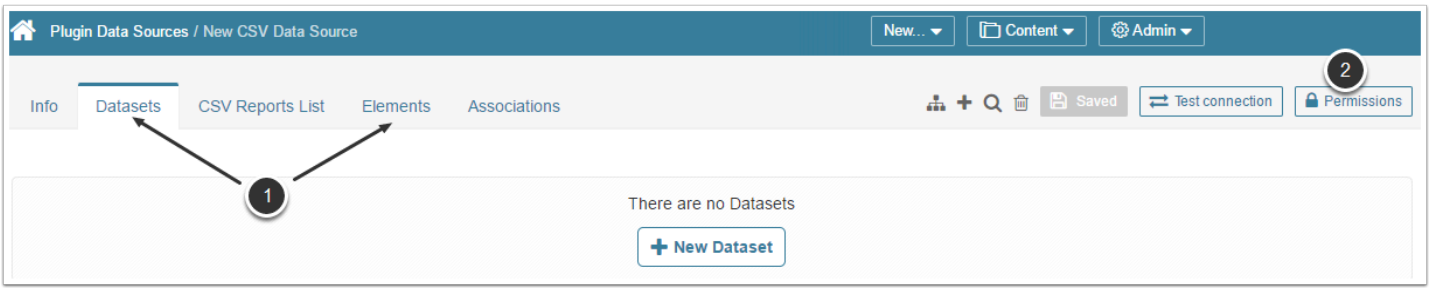
2. **CSV Delimiter:** Confirm that the Delimiter character is the same as in your CSV file

Advanced Configuration



1. **External Reports fetch method**: This setting influences options available in the *CSV Report List* tab:
   - **automatically**: just click **Refresh list** and all Reports are going to be fetched by the system
   - **manually**: Reports may be added one-by-one or via CSV file.
2. Optionally, specify the maximum number of concurrent **Threads per Trigger execution** to be used in background processing when the system updates Metrics and Reports for this Data Source. If you do not specify any value for this setting, batch data collection processing will be single-threaded.

Other Settings



1. You can create elements or Datasets directly from the respective tabs
2. Click **Permissions** to assign them to Groups or Power Users

14. Sourcing Data from Dropbox Paper

14.1 Establish connectivity to Dropbox Paper

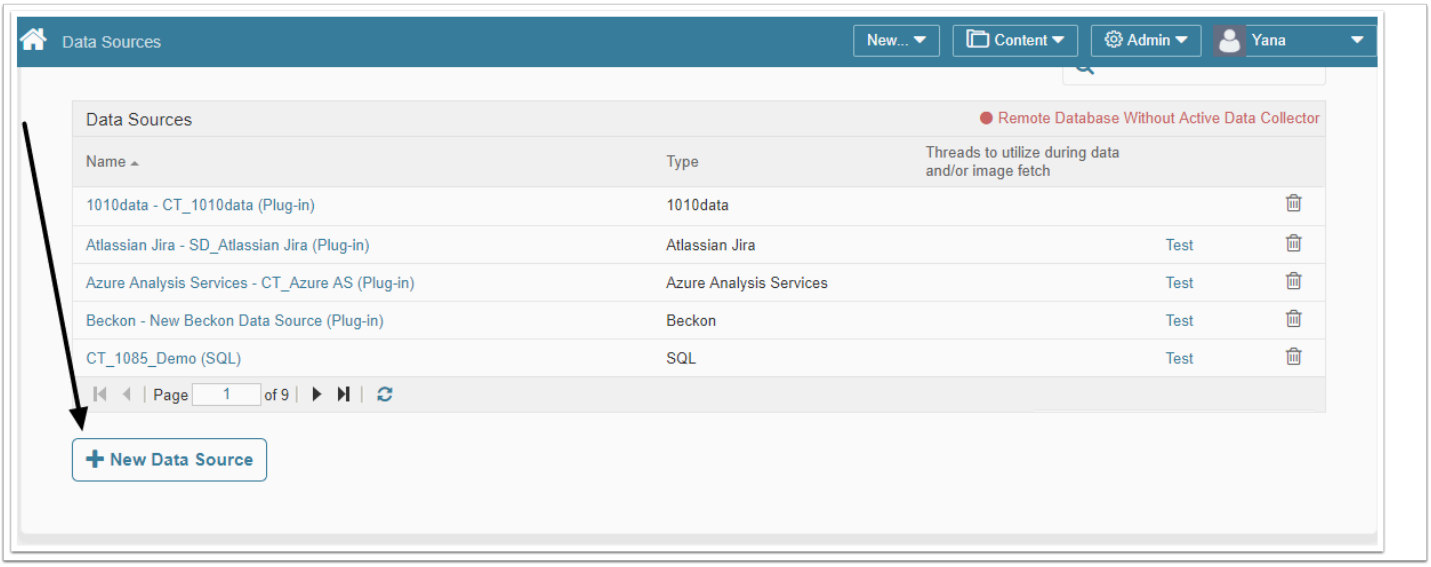
This article describes how to connect to **Dropbox Paper** server in order to load data into Datasets and Reports in Metric Insights.

PREREQUISITES

- Before using the Dropbox Paper plugin, you must have registered an account with Dropbox
- Since the plugin uses security token from your account, get the Dropbox security Token

General instructions on setting up data sources based on plugins can be found here

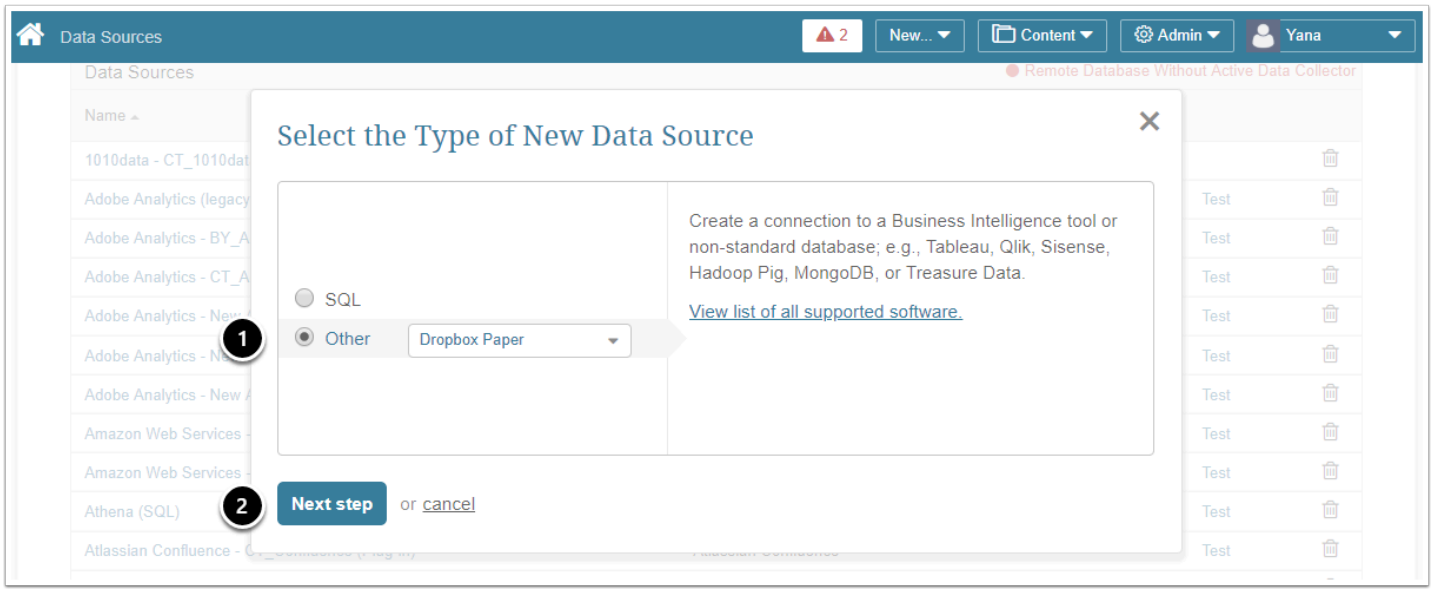
1. Access Admin > Data Sources



At the bottom of the screen click **[+ New Data Source].**

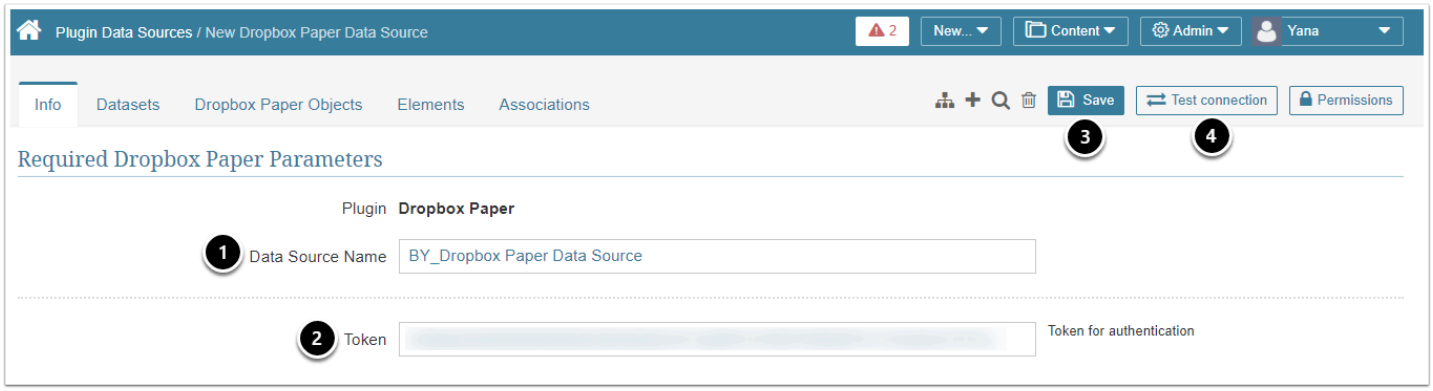
The *Select the Type of New Data Source* pop-up opens.

2. "Select the Type of New Data Source" pop-up opens



1. Select "Other" and choose "Dropbox Paper" from the drop-down list
2. **Next step**

3. Provide the Required Parameters

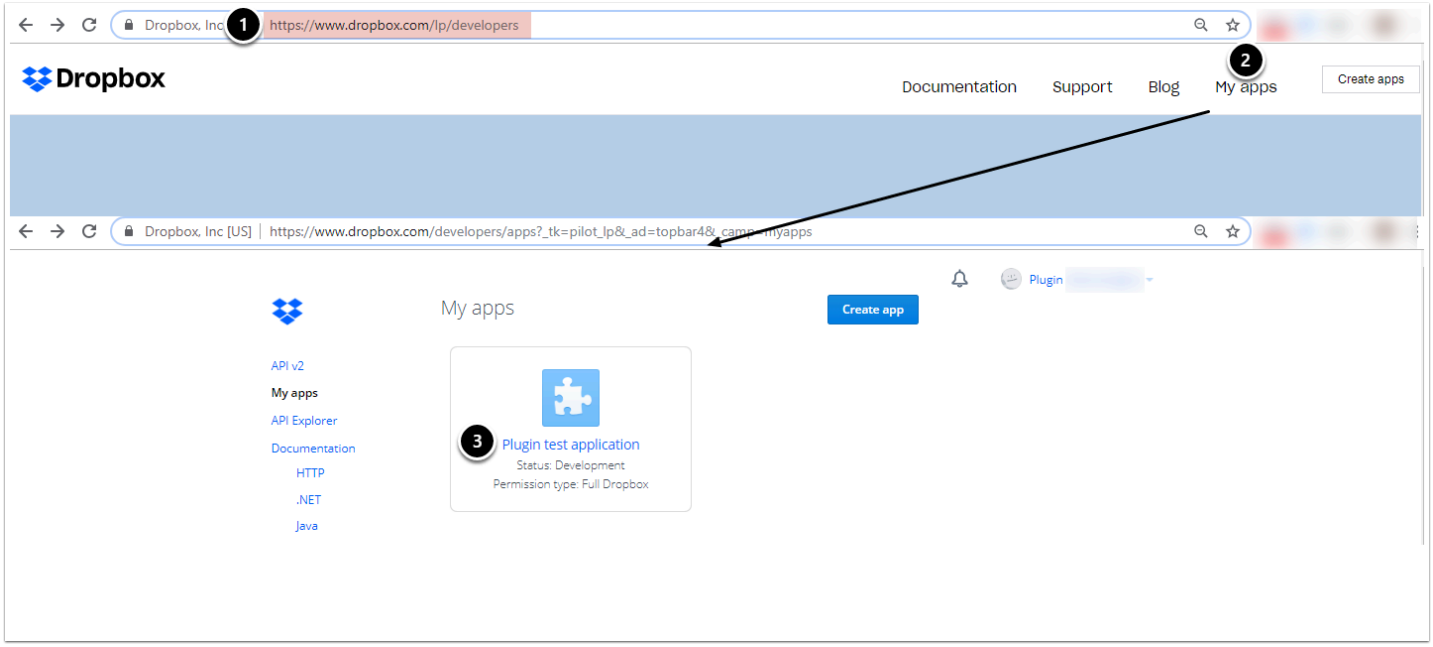


1. **Data Source Name** is defaulted but you may modify it
2. **Token:** enter the security token provided by Dropbox
3. **Save** your entries
4. **Test Connection**

If your connection is successful, you may move on to **Advanced settings**.

3.1. How to obtain a security Token in Dropbox?

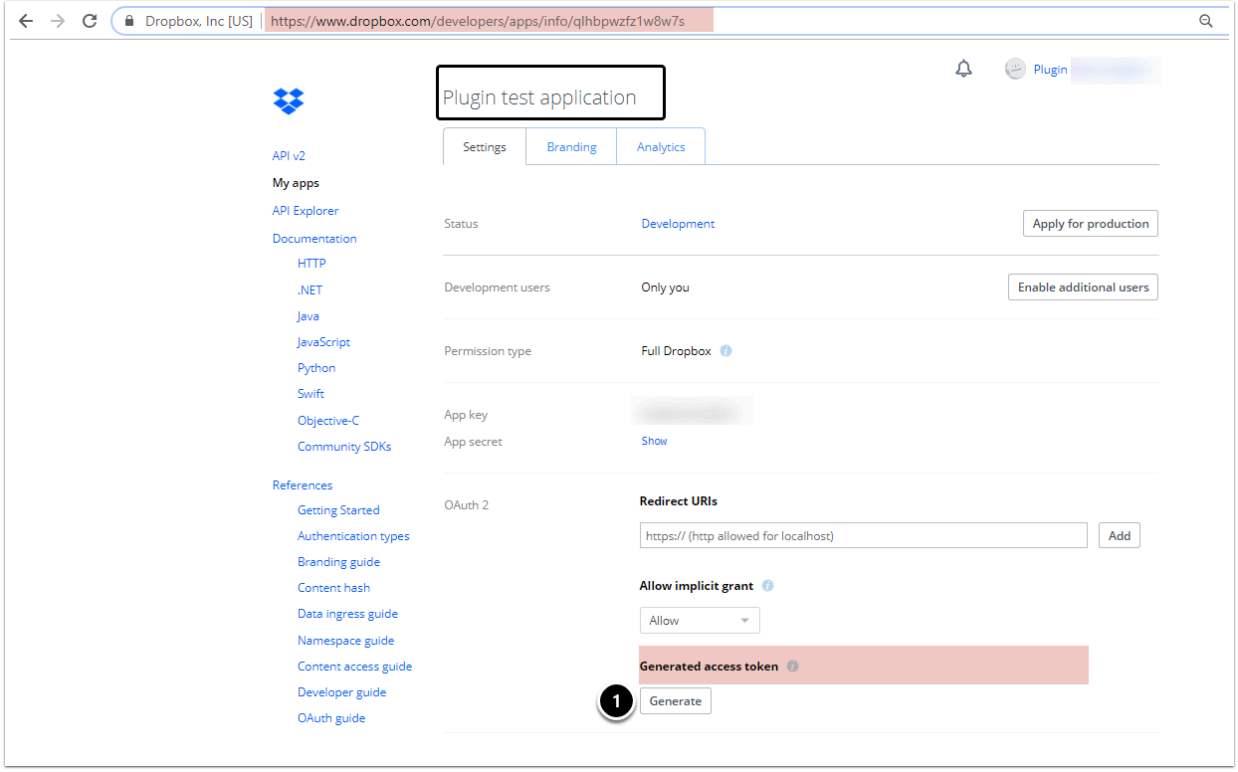
3.1.1. DBX Platform developer portal > My apps



1. Access [DBX Platform Developer Portal]
2. On the Homepage, click **[My apps]**
3. Click the name of your Application
   - If you do not have an application yet, create one with the **[Create app]** button

You will be redirected to the App's page on the App Console.

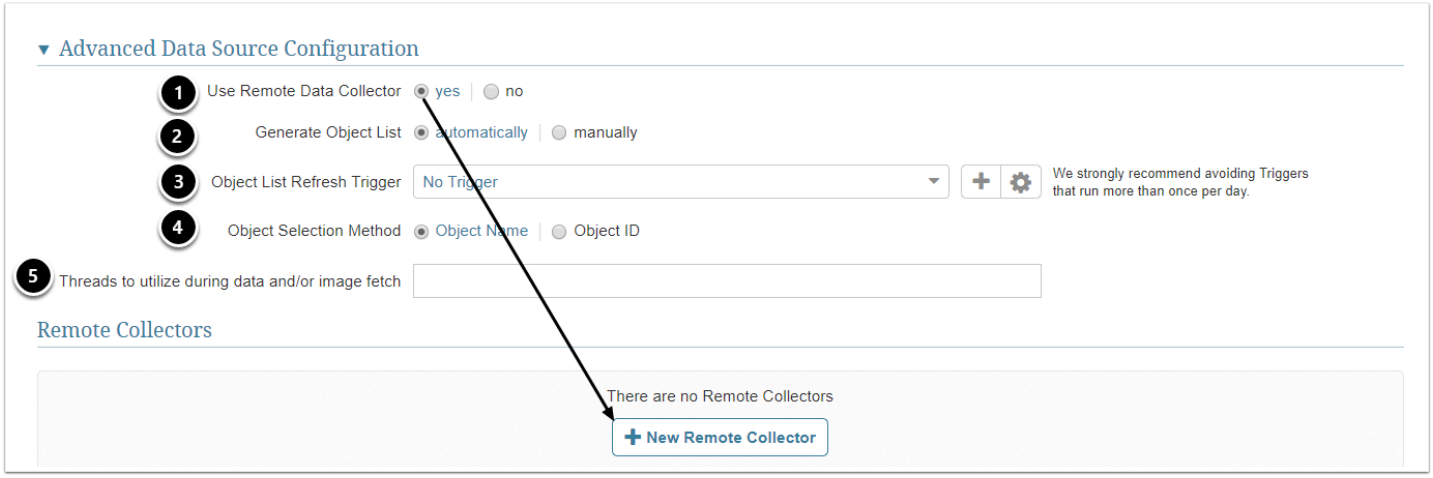
3.1.2. Dropbox App Console > Generate Token



On your App's page, find the **Generate button:**

- Click **[Generate]** to get the Token

4. Advanced Configuration

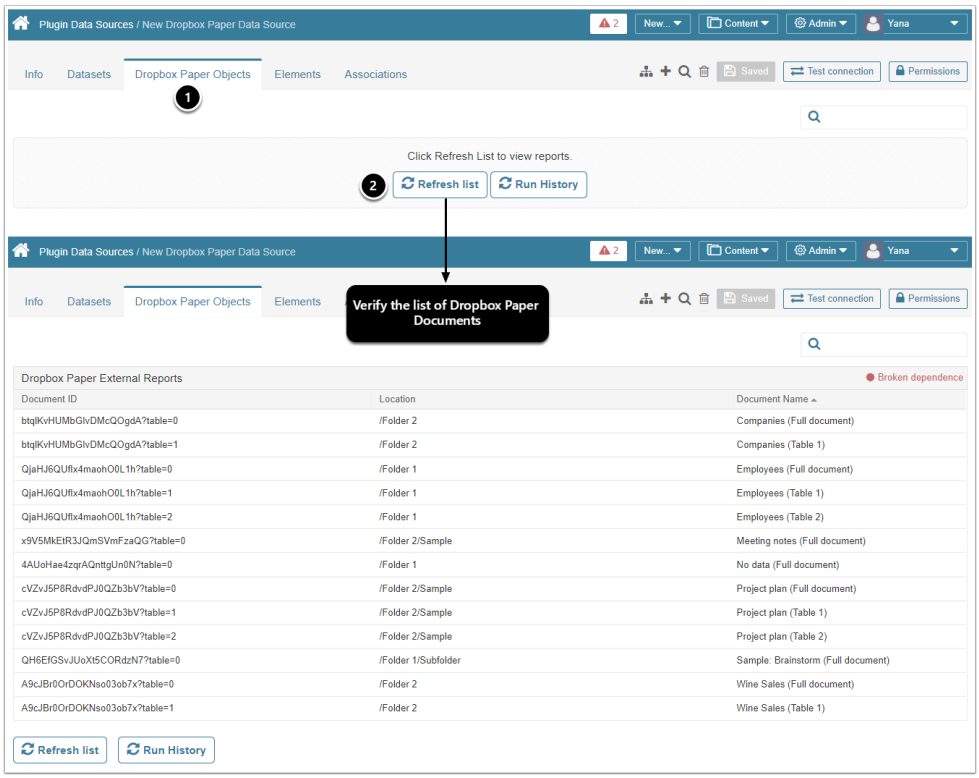


1. **Use Remote Data Collector:** is set to "no" by default
   - If required, switch to "yes" and add a Remote Data Collector by clicking **[+New Remote Collector]**.

2. **Generate Object List**: This setting influences options available in the ***Dropbox Paper Objects*** tab:
   - *automatically*: all Objects are going to be fetched by the system
   - *manually*: Objects may be added one-by-one or via CSV file

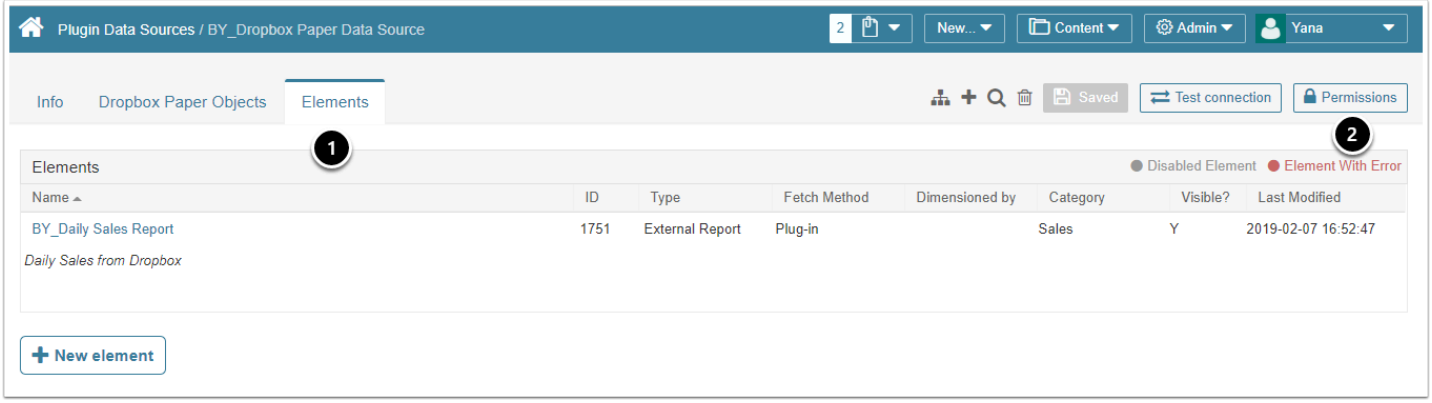
3. **Object List Refresh Trigger:** from the dropdown, select the Trigger that will be used to fetch data from Dropbox Paper Objects
4. Optionally, state the maximum number of concurrent **Threads to utilize during data and/or image fetch** to be used in background processing when the system updates Documents for this Data Source:
   - If you do not specify any value for this setting, batch data collection processing will be single-threaded

5. Obtain a list of External Reports



1. Go to **Dropbox Paper Objects** tab
2. To obtain a list of **Dropbox Paper** Documents, click the **[Refresh list]** button

6. Other Settings



1. You can create Elements directly from the respective tab
2. Click **Permissions** to assign permissions to the Data Source to Groups or Power Users

What's next?

[How to collect data from Dropbox Paper]

14.2 Collect images from Dropbox Paper

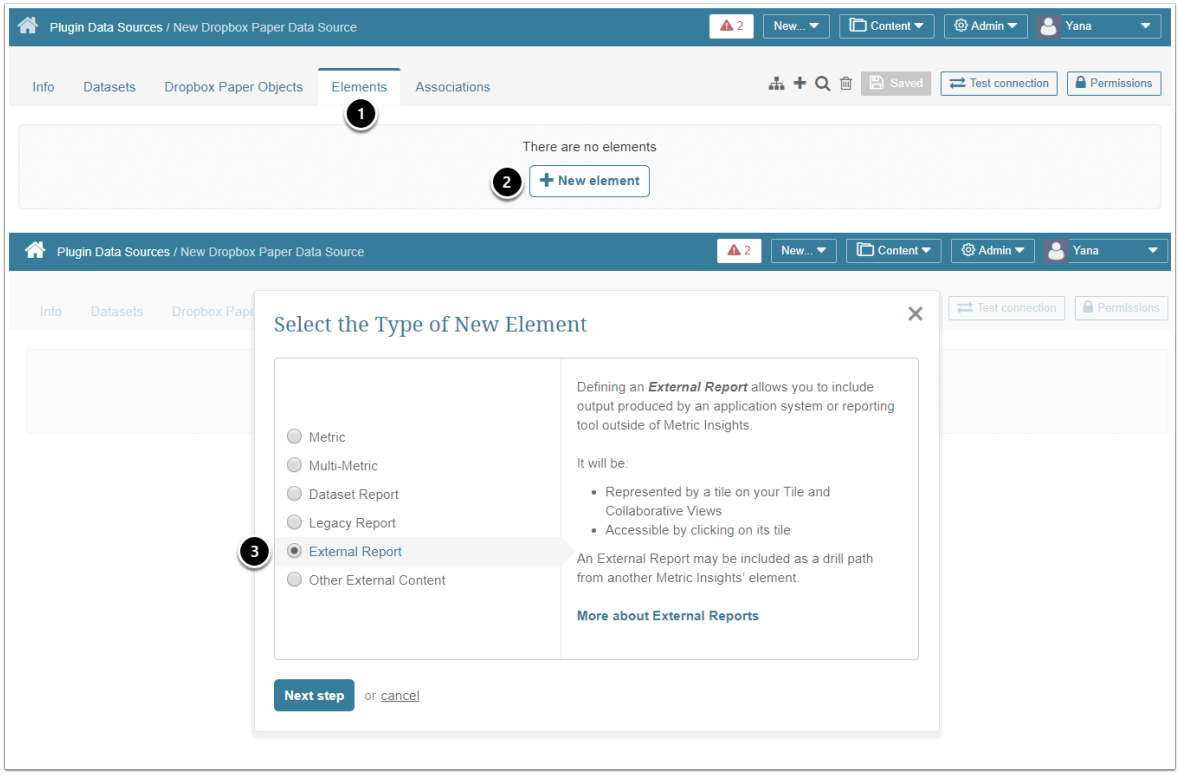
A Metric Insights' element or Dataset can be populated automatically based on data fetched from Dropbox Paper.

PREREQUISITE:

- You must have already established connectivity to Dropbox Paper via the respective plugin connection profile.

This article shows how to create an External Report the **Plugin Data Sources Editor** (Alternatively, access New > External Report > Dropbox Paper to build an External Report).

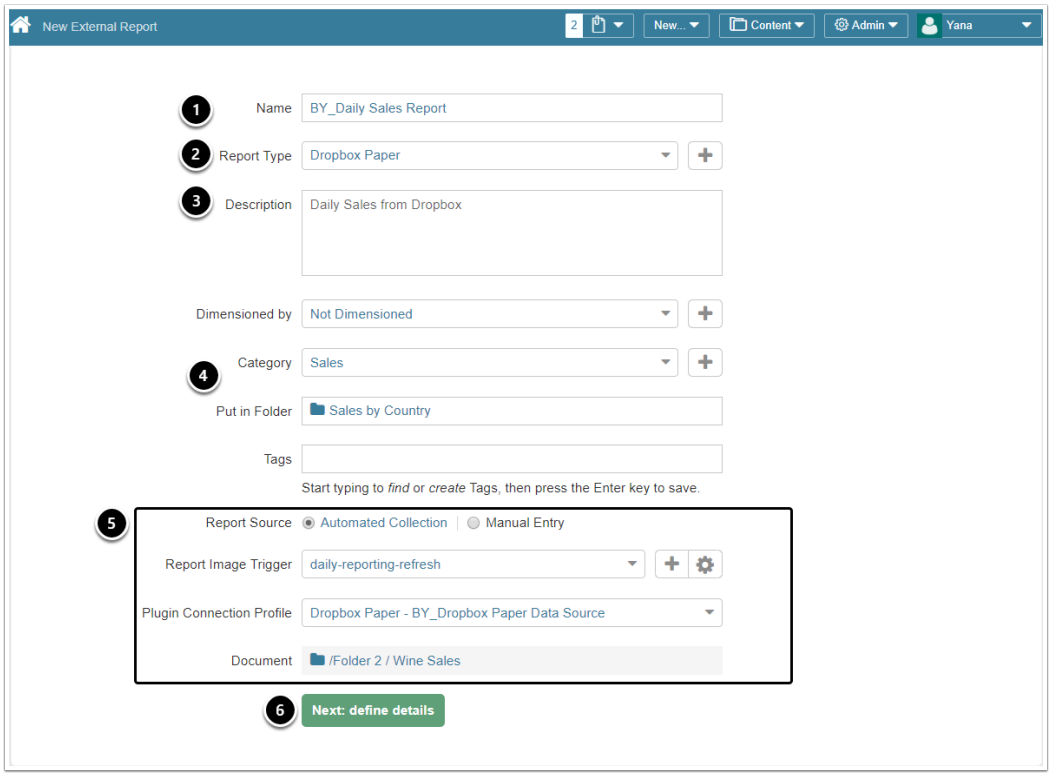
1. Plugin Data Sources Editor > Elements tab



1. Access **Plugin Data Sources Editor > Elements tab**
2. Clicking **[New Element]** opens the "Select the Type of New Element pop-up"
3. Choose External Report

[Next step].

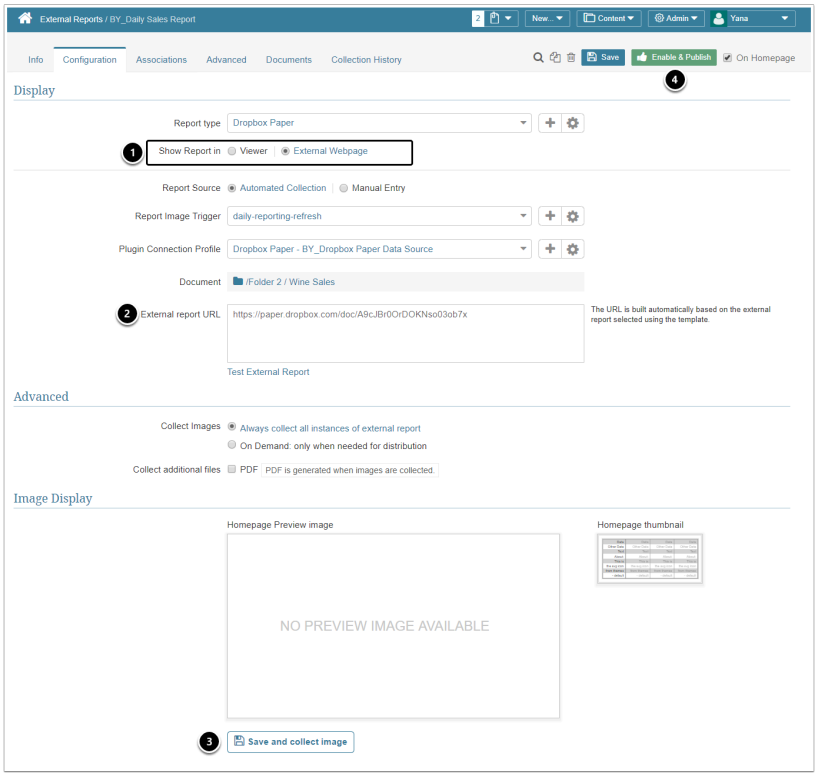
2. Define the Report basics



The *New External Report* screen opens. Provide the following information:

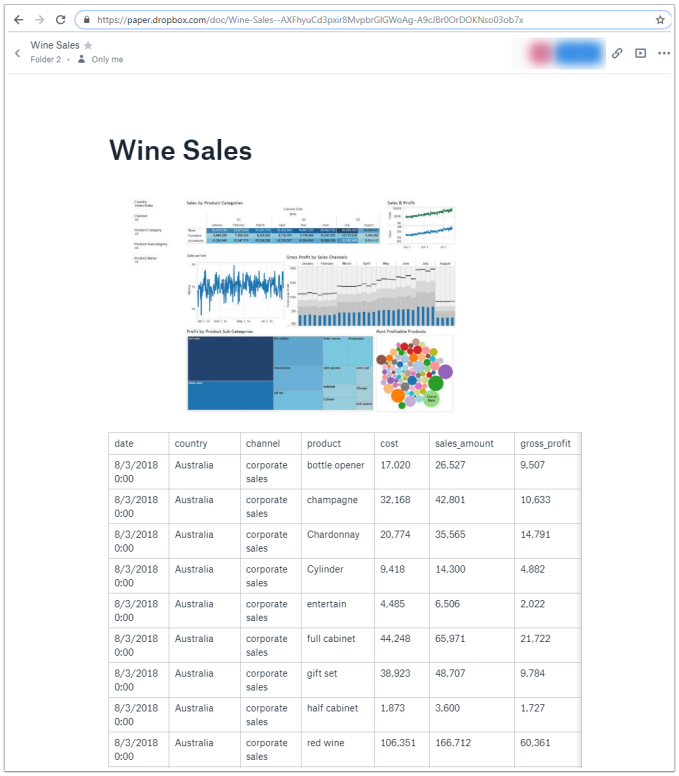
1. Give your new External Report a **Name**
2. Report Type: from the dropdown, select Dropbox Paper
3. If necessary, add a **Description**
4. Place your Report in a relevant **Category** and, optionally, add it to a **Folder**
5. Define whether you want Report content to be updated manually or automatically. In case you choose **Automatic Collection**, define the following settings:
   - Specify the **Report Image Trigger** from the drop-down list
   - Select the **Plugin Connection Profile** you have created for Dropbox Paper
   - **Report:** Select a report available on the server
6. Click **[Next: define details]** to proceed with Report creation

3. Additional Settings



1. **Show Report in:** The Report sourced from Dropbox Paper is typically shown on the source (external) page
   - You can always change the default "External Webpage" setting to "Viewer"
2. The **External Report URL** will be generated automatically based on your other inputs. If you like, you can modify the URL by appending a question mark (?) followed by any filter or parameter settings
3. **Save and Collect Image**: click to create the Preview and Thumbnail images for display on the Homepage and Notifications.
4. **Enable and publish** your Report

4. Verify the display on external Webpage (or in Viewer)



date	country	channel	product	cost	sales_amount	gross_profit
8/3/2018 0:00	Australia	corporate sales	bottle opener	17,020	26,527	9,507
8/3/2018 0:00	Australia	corporate sales	champagne	32,168	42,801	10,633
8/3/2018 0:00	Australia	corporate sales	Chardonnay	20,774	35,565	14,791
8/3/2018 0:00	Australia	corporate sales	Cylinder	9,418	14,300	4,882
8/3/2018 0:00	Australia	corporate sales	entertain	4,485	6,506	2,022
8/3/2018 0:00	Australia	corporate sales	full cabinet	44,248	65,971	21,722
8/3/2018 0:00	Australia	corporate sales	gift set	38,923	48,707	9,784
8/3/2018 0:00	Australia	corporate sales	half cabinet	1,873	3,600	1,727
8/3/2018 0:00	Australia	corporate sales	red wine	106,351	166,712	60,361

The External Report is now available for use.

- Based on your display settings, the Report will either be shown in Dropbox Paper (External Webpage) or in the Metric Insights Viewer.

15. Sourcing Data using Elasticsearch

15.1 Establish connectivity to Elasticsearch

An Administrator can use the process described in this article to create a new Plugin Data Source to fetch data using Elasticsearch.

PREREQUISITES:

1. **Java 8 software** must be installed and running on the same server as the Metric Insights' Data Collector.
2. **For clustered environments**:
   - [prior to 5.6.1] *Elasticsearch configuration file* must be created and added to the */Insightd* (Windows) or */datacollector* (Linux) directory
   - See details below

Add the configuration file to /Insightd/plugins or /datacollector/plugins directory

The config file should contain the following **Parameter Values**:

1. Cluster Name
2. Cluster Headers
3. Inner API Port
4. Enabled SSL
5. Transport Sniff

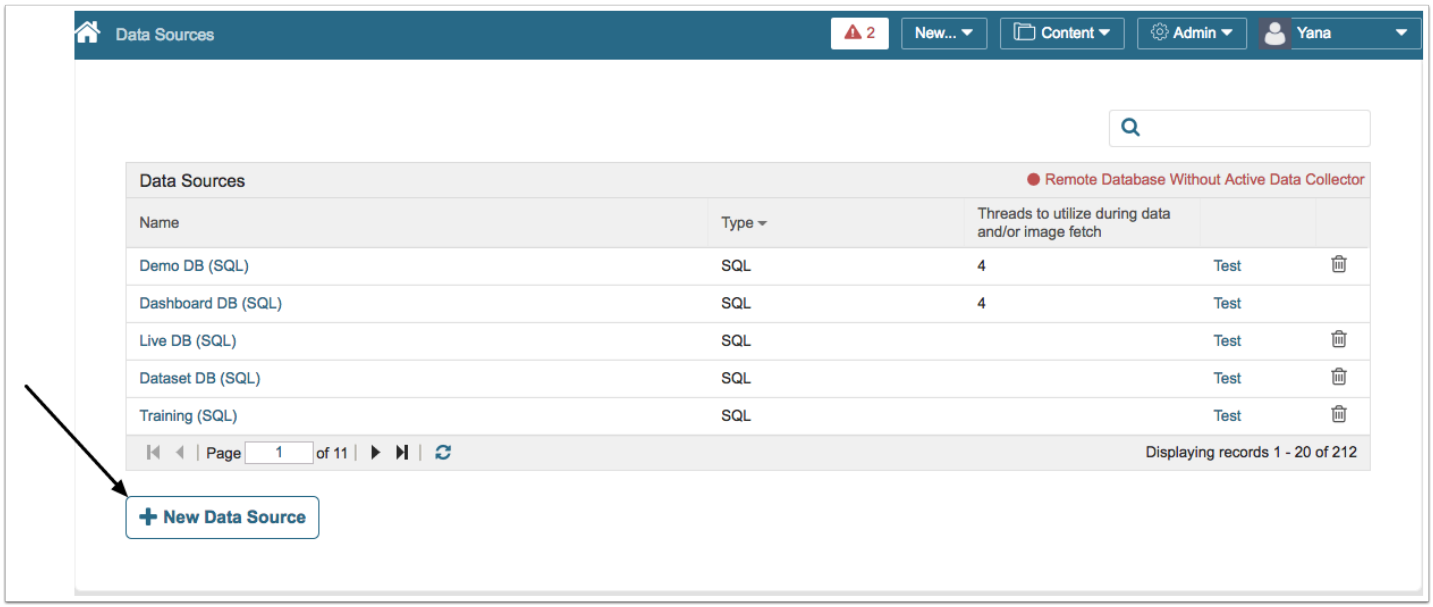
elasticsearch.conf

```
cluster=[redacted]
headersXFoundCluster=[redacted]
innerApiPort=[redacted]
sslEnabled='yes'
transportSniff='yes'
```

Place the **Elasticsearch configuration file** at:

1. `C:\Program Files (x86)\Metric Insights\Insightd\plugins` (for Windows-based Data Collectors)
2. `opt/mi/datacollector/plugins` (for Linux-based Data Collectors)

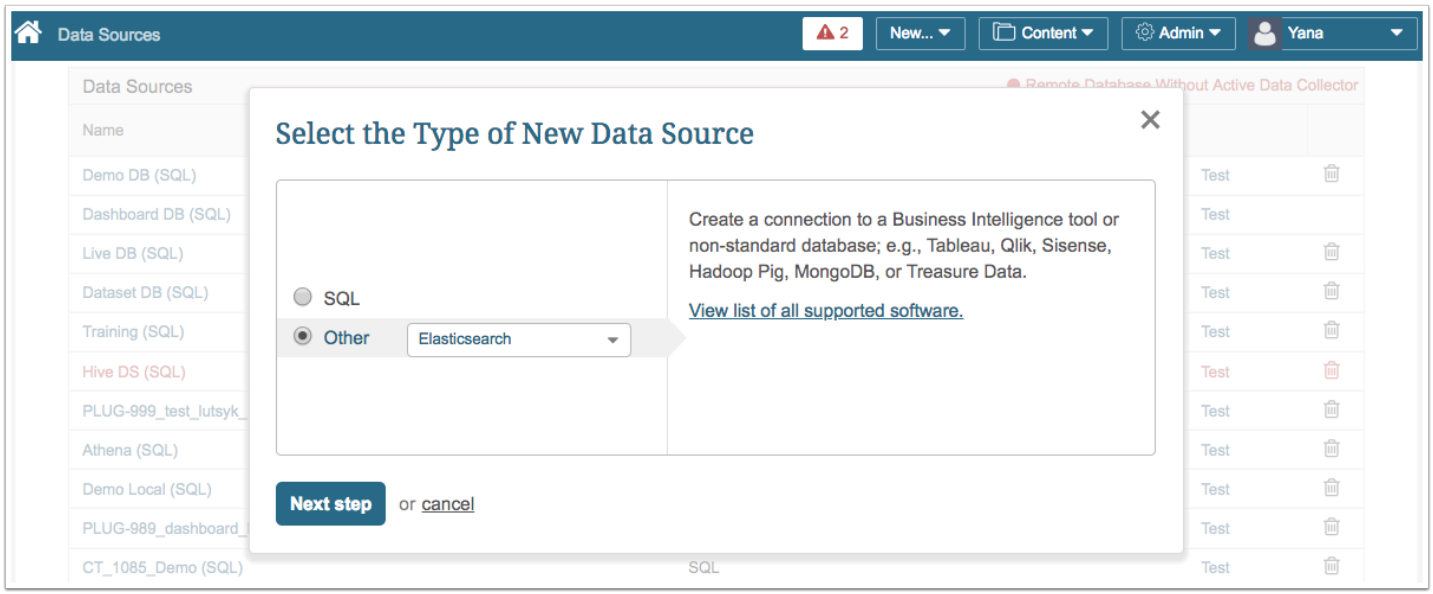
1. Access Admin > Data Sources



At the bottom of the screen click **[+ New Data Source].**

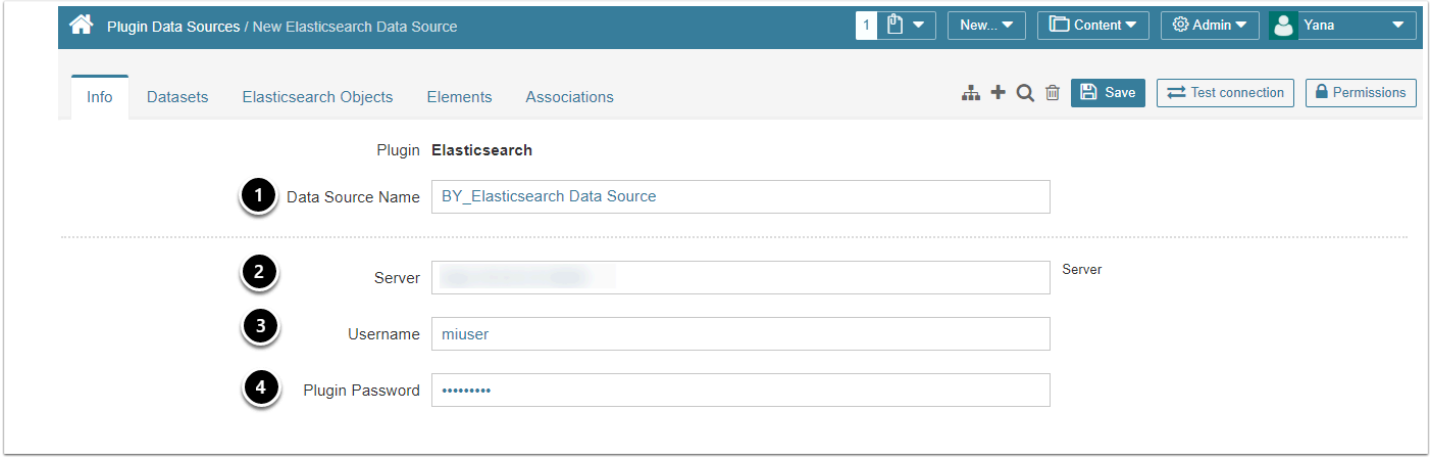
The *Select the Type of New Data Source* pop-up opens.

2. Select "Other" Data Source Type and choose "Elasticserach" from the drop-down list



Proceed with creating a Data Source by moving to the **Next step**.

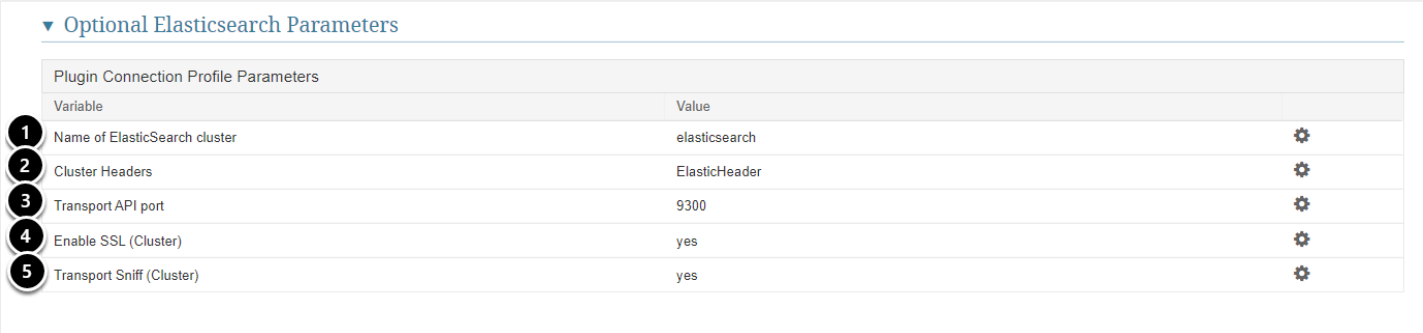
3. Required Elasticsearch Parameters



1. **Data Source Name** is defaulted but you may modify it
2. **Server:** specify the endpoint server (can be local or remote)
3. **Username:** Note that your **Username** must be in the same format that your Elasticsearch server uses for authentication
4. **Plugin Password:** enter your password credential

4. Optional Elasticsearch Parameters (for clustered servers)

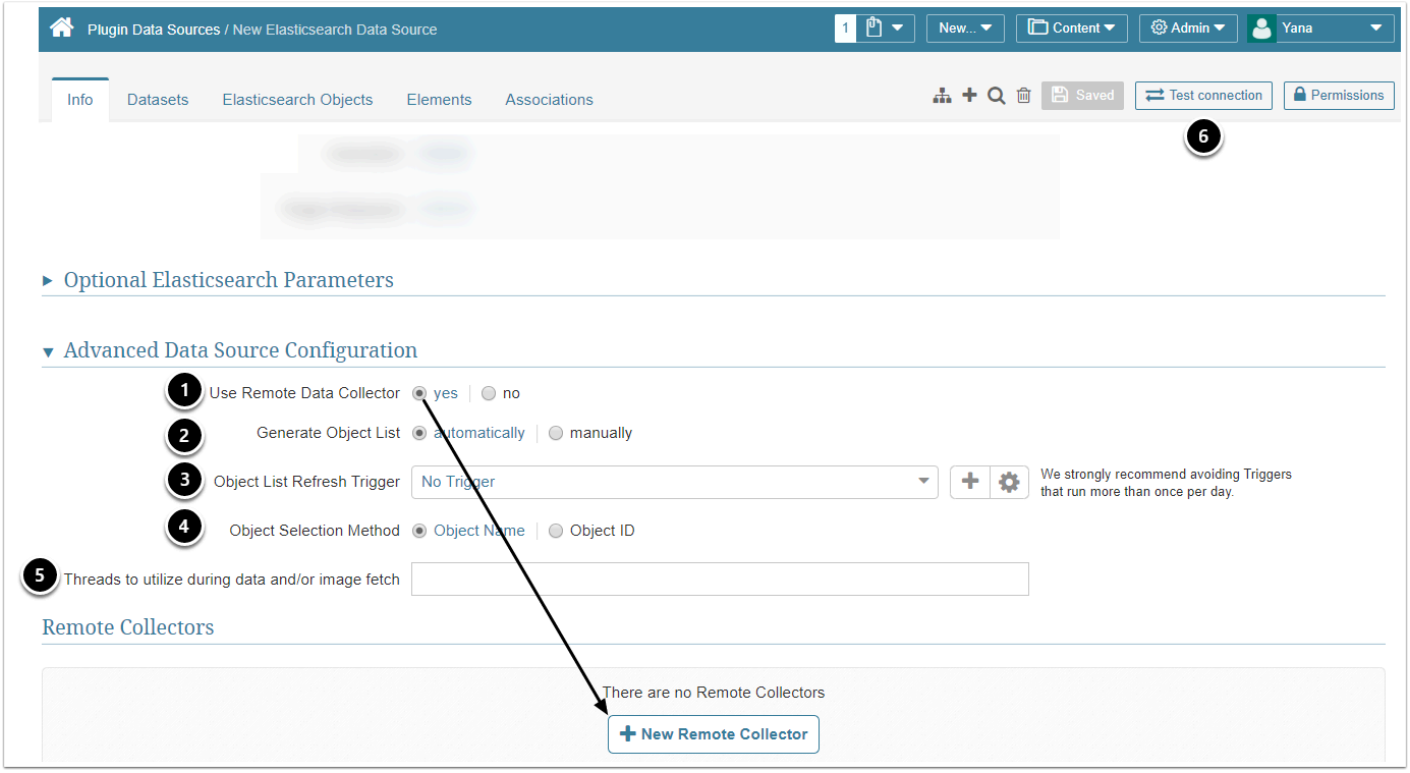
> ⚠ As of Release 5.6.1, additional Parameters for clustered servers can be configured in the Data Source connection profile (UI) instead of the Elasticsearch configuration file.



1. **Name of ElasticSearch cluster:** provide the name of your Elasticsearch cluster
2. **Cluster Headers:** specify the name(s) of the required Cluster Header(s)
3. **Trasport API port:** input the API communication port

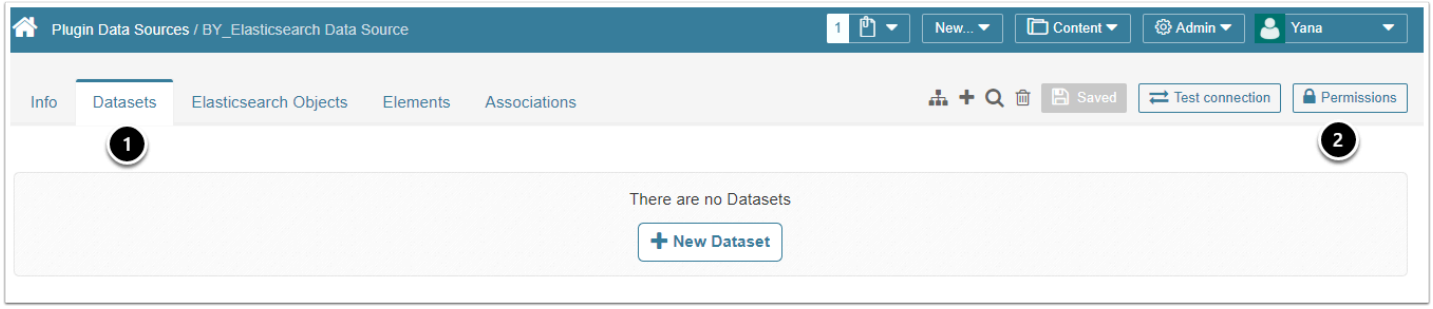
4. **Enable SSL (Cluster):** set to 'yes'
5. **Transport Sniff (Cluster):** set to 'yes'

5. Advanced Configuration



1. **Use Remote Data Collector:** switch to "yes" and add a Remote Data Collector by clicking **[+New Remote Collector]** under **Remote Collectors** settings
2. **Generate Object List**
   - *automatically*: all Reports are going to be fetched by the system
   - *manually*: Reports may be added one-by-one or via CSV file
3. **Object List Refresh Trigger:** from the dropdown, select the Trigger that will be used to fetch data via the Elasticsearch plugin
4. **Object Selection Method:** specify how Elasticsearch Reports will be fetched
5. Optionally, state the maximum number of concurrent **Threads to utilize during data and/or image fetch** to be used in background processing when the system updates Reports for this Data Source
   - If you do not specify any value for this setting, batch data collection processing will be single-threaded
6. **Test Connection:** this will also **Save** your entries

6. Other Settings



1. You can create Datasets directly from the respective tab
2. Click **Permissions** to assign permissions to the Data Source to Groups or Power Users

What's next?

[How to Collect Data using Elasticsearch]

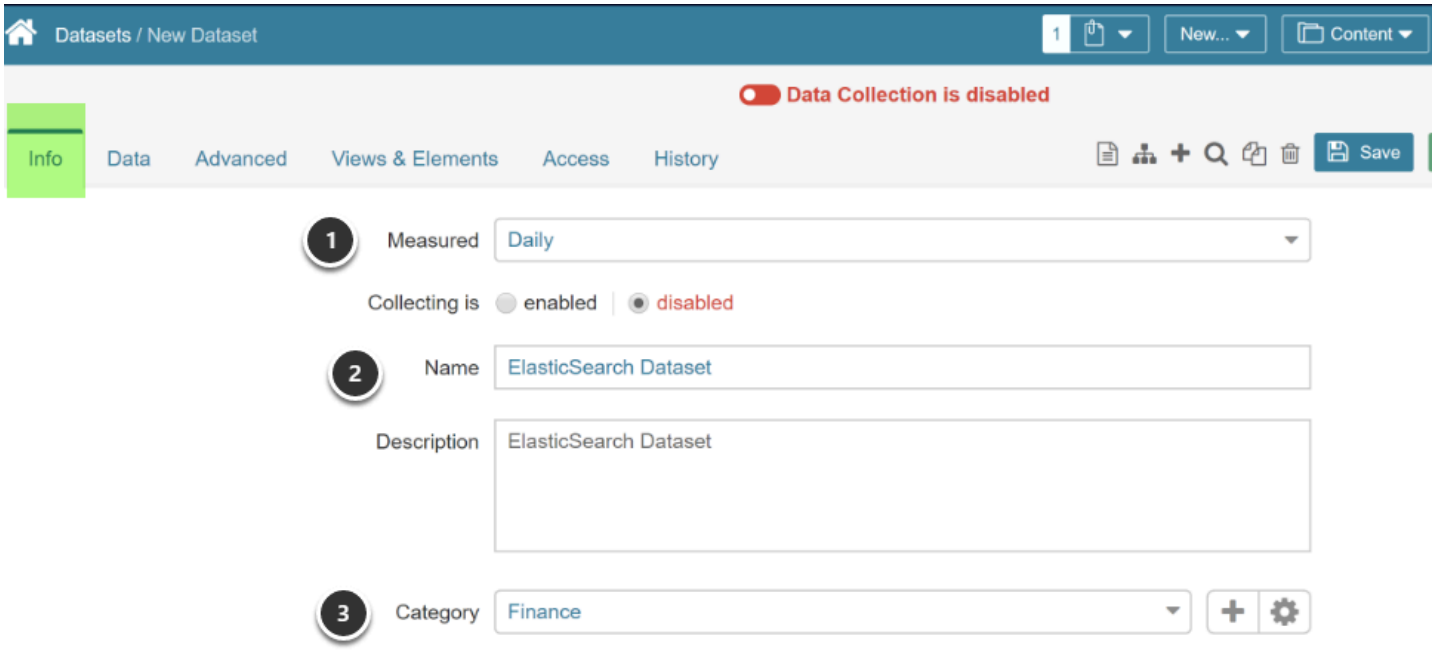
15.2 How to collect data using Elasticsearch

This article will show you how to create an Element using an **Elastic Search** plug-in as a data source. It assumes that you have already established connectivity to **Elastic Search**.

1. Access New > Dataset > Create New

The Editor for Datasets will open on the INFO tab.

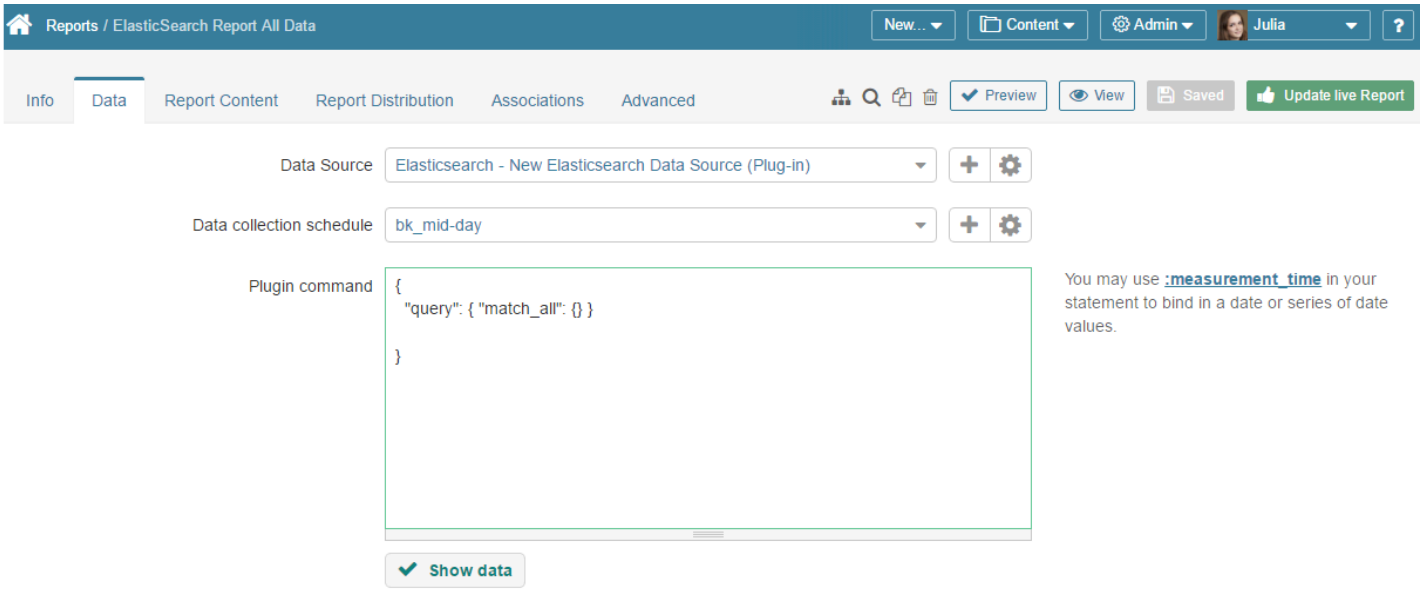
2. Dataset Editor - Info tab (the basics)



1. **Reported**: choose the measurement interval from the drop-down list
2. **Name the Report**: Use a unique descriptive name
3. **Category**: define a category this element belongs to

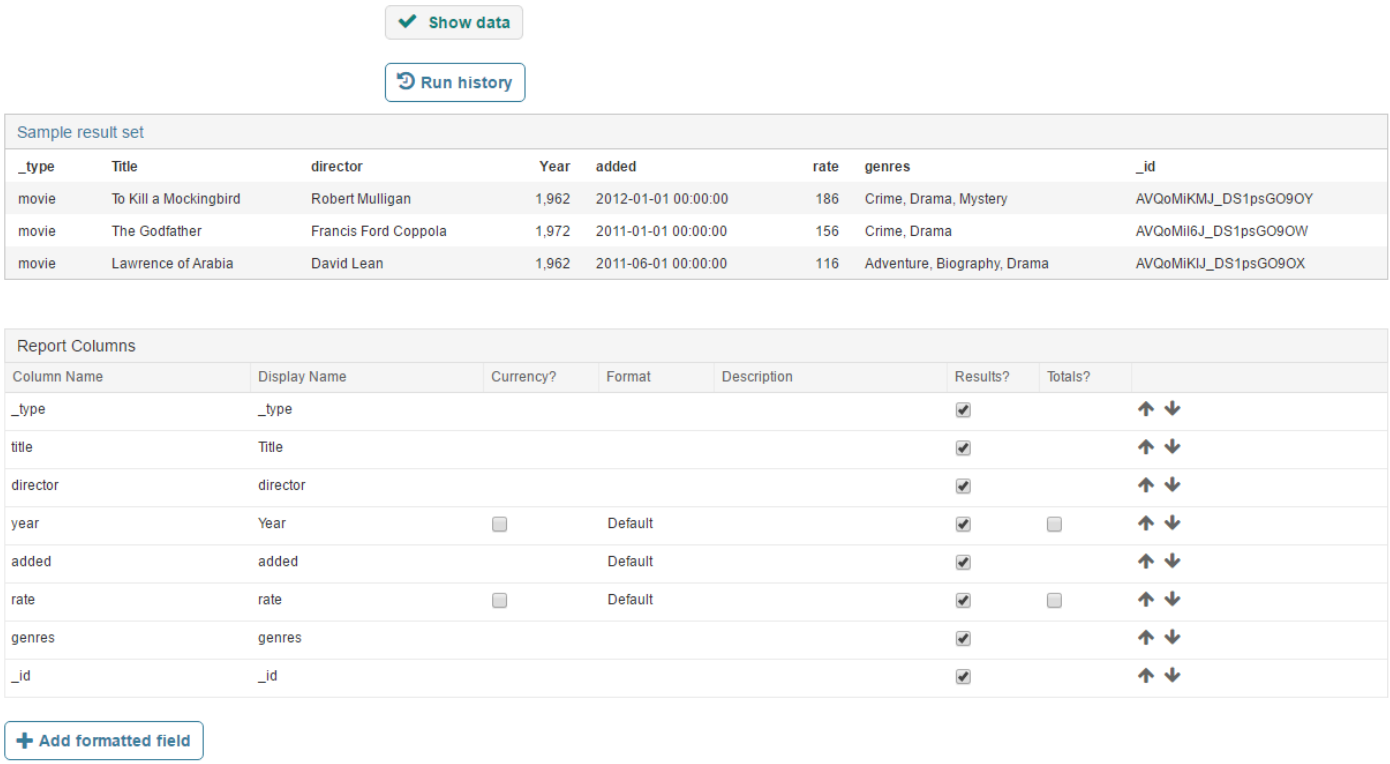
To move on to defining data collection details, open ***Data*** tab

3. Data tab



1. **Data Source:** select the account you have created for **Elasticsearch**
2. **Data Collection Schedule:** Specify the trigger that will be used to collect the data for your report
3. Input **Plug-in Command** listing all the data you would like to fetch from **Elasticsearch.** For more details refer to Elasticsearch String Query Guide
4. Once you are ready with you command, click **Show Data**.

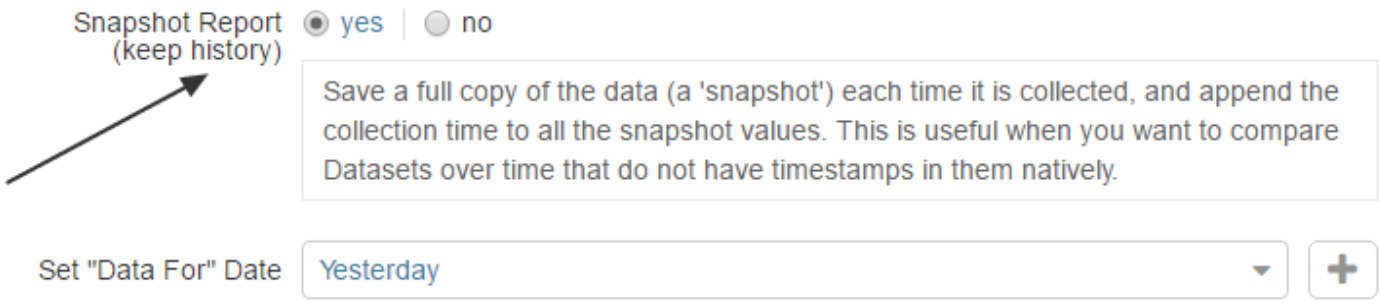
4. Plug-in command will be validated and Data Collected on Save from Visual Editor



If the command is validated successfully, the **Sample Results set** and **Report columns** are going to be shown below.

At the upper right corner of the screen click **Update live Report**

5. [Optional] Snapshot Report



You can choose to keep the history of data changes by turning this Report into a Snapshot Report. For more details refer to: [How to create a Snapshot Report]

Result

Reports / Uncategorized / ElasticSearch Report All Data

New... Content Admin Julia

ElasticSearch Report All Data Data for Saturday 03/04/2017

ElasticSearch Dimensioned Report Report

ElasticSearch Report All Data

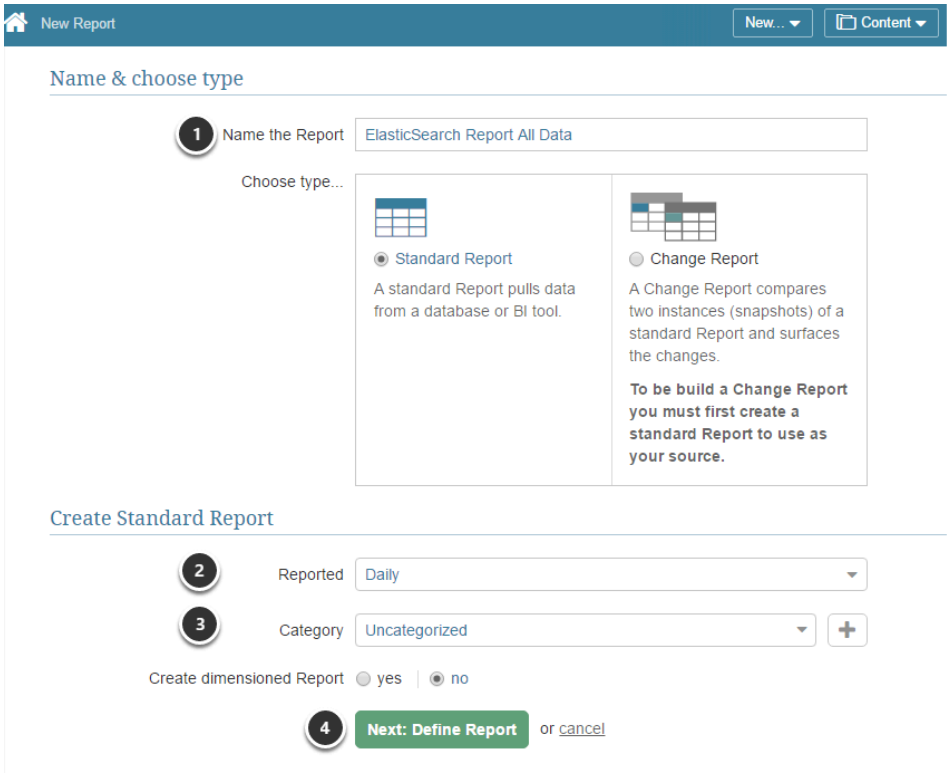
_type	Title	director	Y...	added	r	genres	_id
movie	To Kill a Mockingbird	Robert Mulligan	1...	2012-01-01 00:00:00	1	Crime, Drama, Mystery	AVQoMiKMJ_DS1psGO9OY
movie	The Godfather	Francis Ford Coppola	1...	2011-01-01 00:00:00	1	Crime, Drama	AVQoMil6J_DS1psGO9OW
movie	Lawrence of Arabia	David Lean	1...	2011-06-01 00:00:00	1	Adventure, Biography, Drama	AVQoMiKIJ_DS1psGO9OX
movie	Apocalypse Now	Francis Ford Coppola	1...	2013-01-01 00:00:00	9	Drama, War	AVQoMiKQJ_DS1psGO9OZ
movie	The Assassination of Jesse James by the Coward...	Andrew Dominik	2...	2014-06-01 00:00:00	1	Biography, Crime, Drama	AVQoMiKaJ_DS1psGO9Ob
movie	Kill Bill => Vol. 1	Quentin Tarantino	2...	2014-01-01 00:00:00	2	Action, Crime, Thriller	AVQoMiKUJ_DS1psGO9Oa

Add Expert Analysis

15.3 How to collect data using Elasticsearch (prior to Version 5.3)

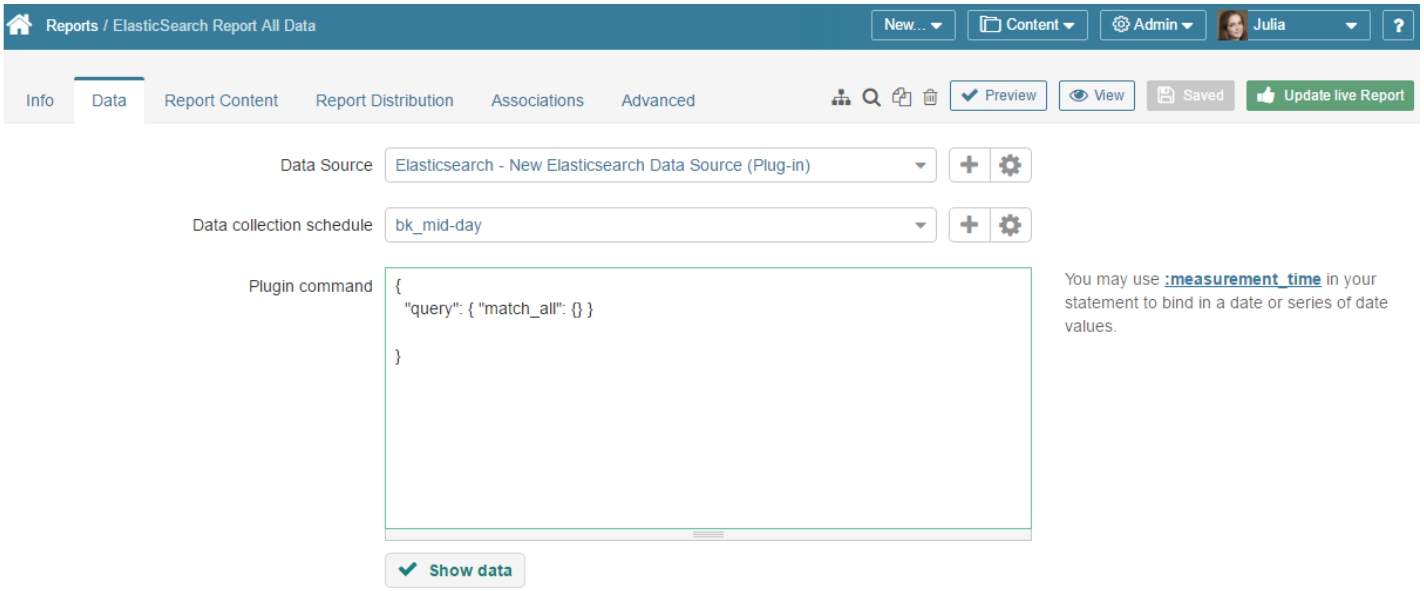
This article will show you how to create an Element using an **Elastic Search** plug-in as a data source. It assumes that you have already established connectivity to **Elastic Search**.

1. Access New > Report



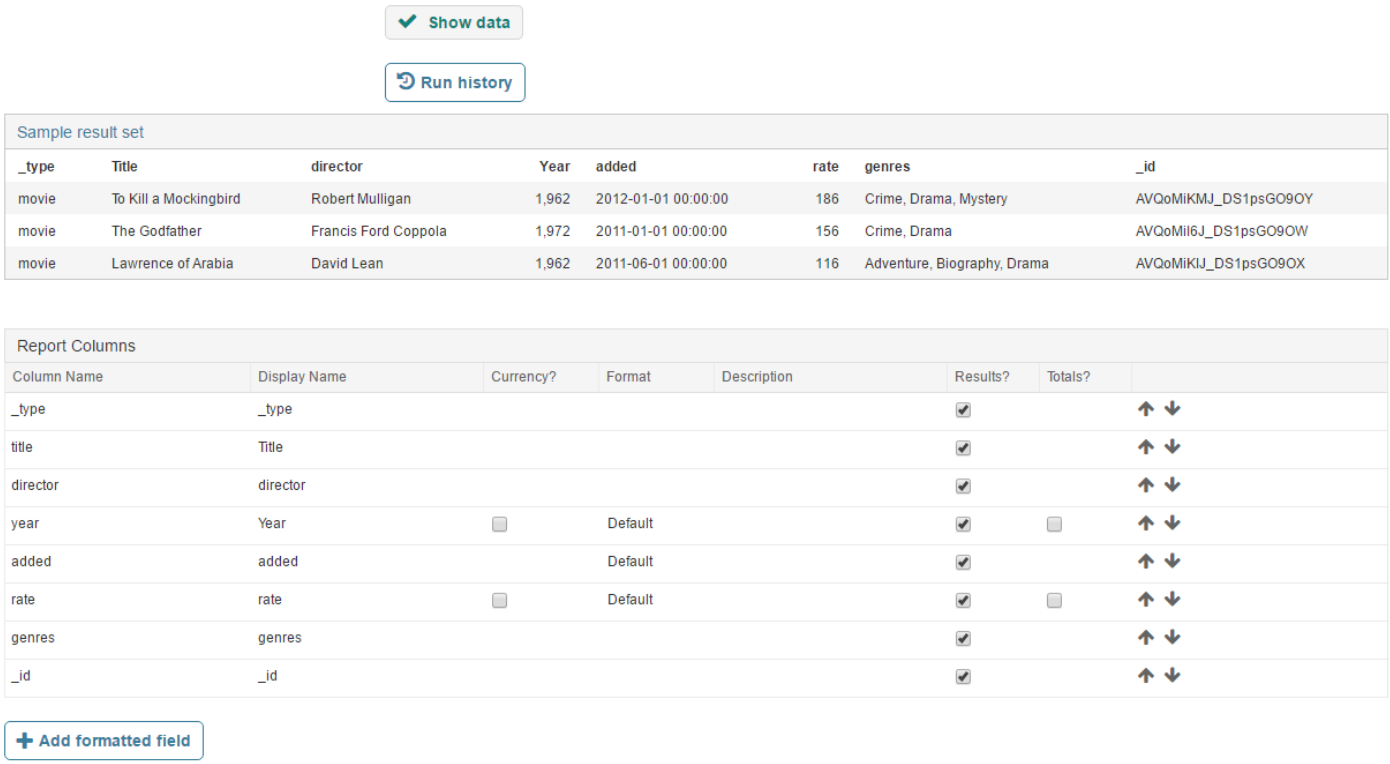
1. **Name the Report**: Define a unique descriptive name of your element
2. **Reported**: choose the measurement interval from the drop-down list
3. **Category**: define a category this element belongs to
4. To move on to defining data collection details, click **Next: Define Report**

2. Full Editor displays the Data Collection tab



1. **Data Source:** select the account you have created for **Elasticsearch**
2. **Data Collection Schedule:** Specify the trigger that will be used to collect the data for your report
3. Input **Plug-in Command** listing all the data you would like to fetch from **Elasticsearch.** For more details refer to Elasticsearch String Query Guide
4. Once you are ready with you command, click **Show Data**.

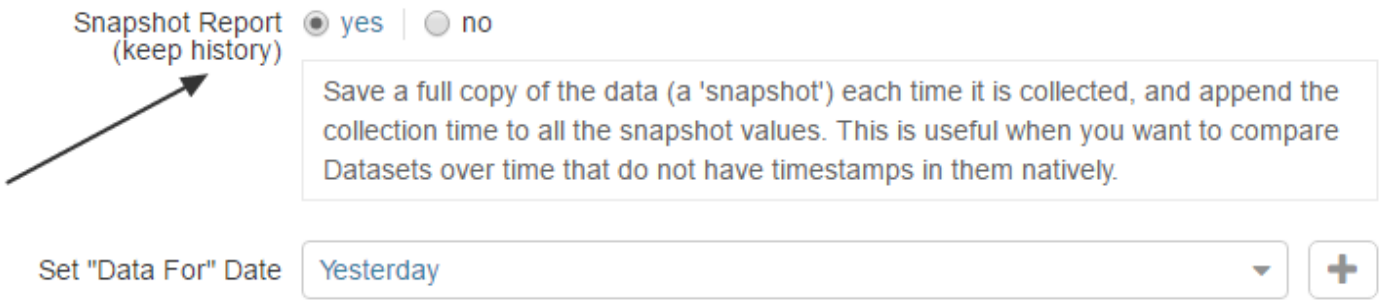
3. Plug-in command will be validated and Data Collected on Save from Visual Editor



If the command is validated successfully, the **Sample Results set** and **Report columns** are going to be shown below.

At the upper right corner of the screen click **Update live Report**

4. [Optional] Snapshot Report



You can choose to keep the history of data changes by turning this Report into a Snapshot Report. For more details refer to: [How to create a Snapshot Report]

Result

Reports / Uncategorized / ElasticSearch Report All Data

New... Content Admin Julia

ElasticSearch Report All Data Data for Saturday 03/04/2017

ElasticSearch Dimensioned Report Report

ElasticSearch Report All Data

_type	Title	director	Y...	added	r	genres	_id
movie	To Kill a Mockingbird	Robert Mulligan	1...	2012-01-01 00:00:00	1	Crime, Drama, Mystery	AVQoMiKMJ_DS1psGO9OY
movie	The Godfather	Francis Ford Coppola	1...	2011-01-01 00:00:00	1	Crime, Drama	AVQoMil6J_DS1psGO9OW
movie	Lawrence of Arabia	David Lean	1...	2011-06-01 00:00:00	1	Adventure, Biography, Drama	AVQoMiKIJ_DS1psGO9OX
movie	Apocalypse Now	Francis Ford Coppola	1...	2013-01-01 00:00:00	9	Drama, War	AVQoMiKQJ_DS1psGO9OZ
movie	The Assassination of Jesse James by the Coward...	Andrew Dominik	2...	2014-06-01 00:00:00	1	Biography, Crime, Drama	AVQoMiKaJ_DS1psGO9Ob
movie	Kill Bill => Vol. 1	Quentin Tarantino	2...	2014-01-01 00:00:00	2	Action, Crime, Thriller	AVQoMiKUJ_DS1psGO9Oa

Add Expert Analysis

16. Sourcing Data using File Data Plugin

16.1 Establish connectivity to File Data

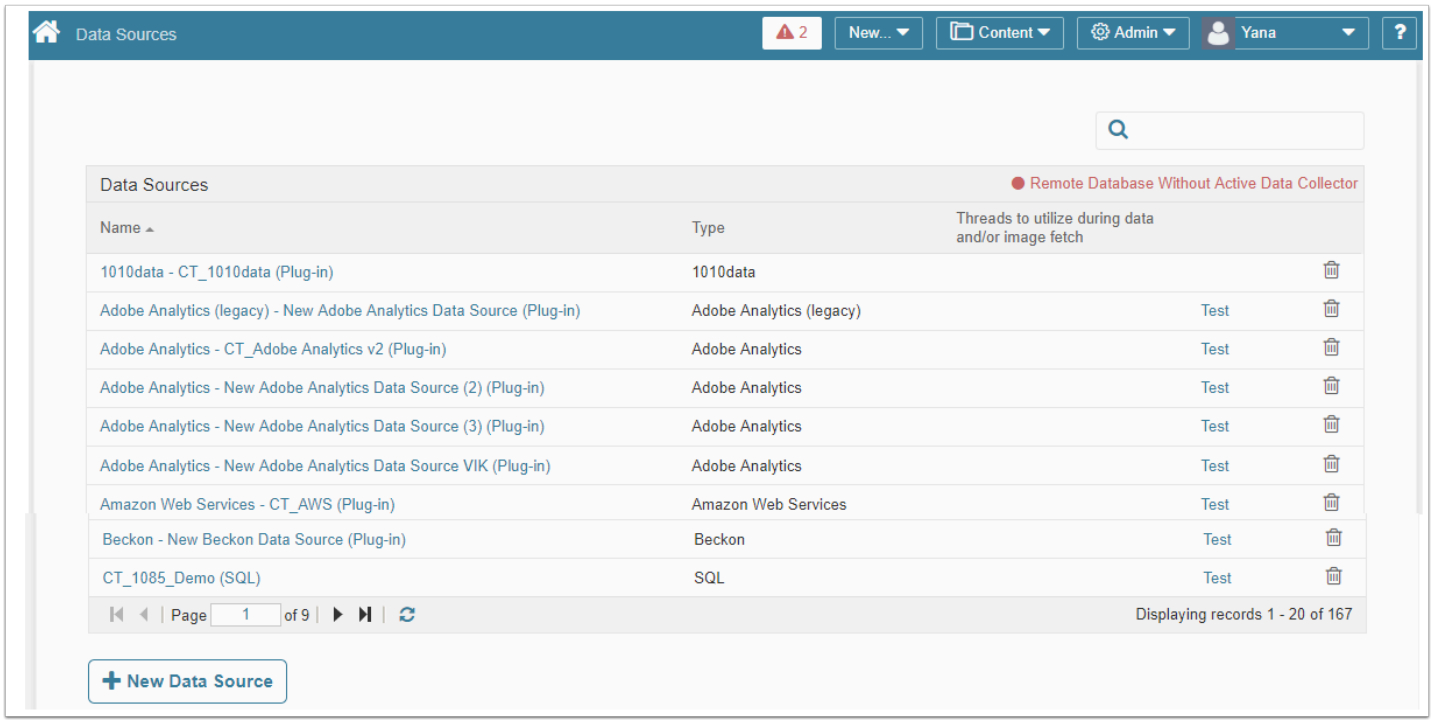
This article describes how to use a File Data Plugin to create a connection profile with Metric Insights.

The file that serves as a source of data can be located on:

- MI server or a server where the Remote Data Collector is installed (local file system)
- ftp/sftp server (over SSH)
- S3 (Amazon file server)

[Release 6.x]: 'Remote Data Collector' renamed to 'Remote Data Processor'

1. Access Admin > Data Sources

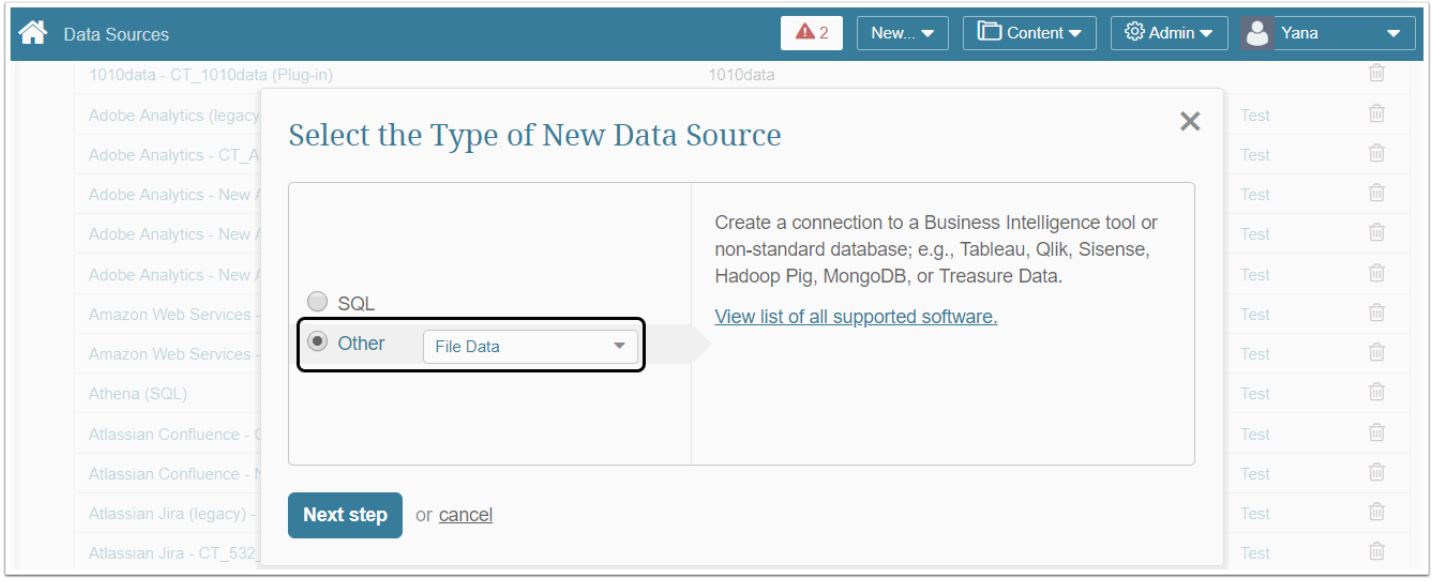


At the bottom of the screen click **[+ New Data Source]**

The *Select the Type of New Data Source* pop-up opens.

2. Select "Other" Data Source Type and choose "File Data"

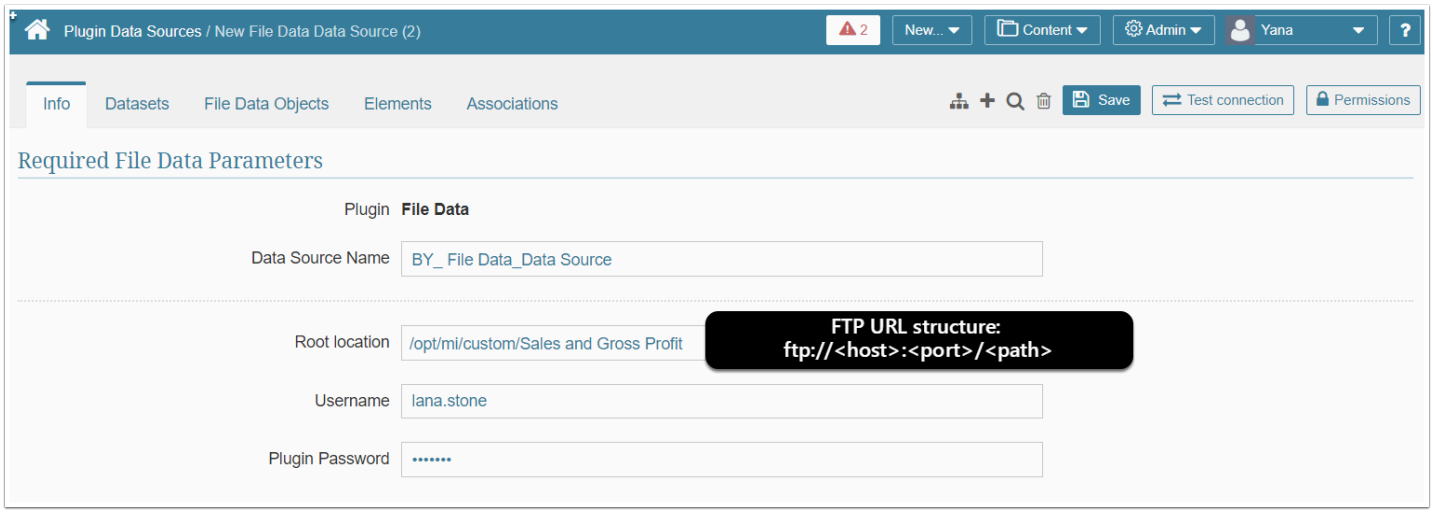
from the drop-down list



Proceed with creating a Data Source by moving to the **Next step**.

3. Provide File Data Parameters

3.1. Required File Data Parameters

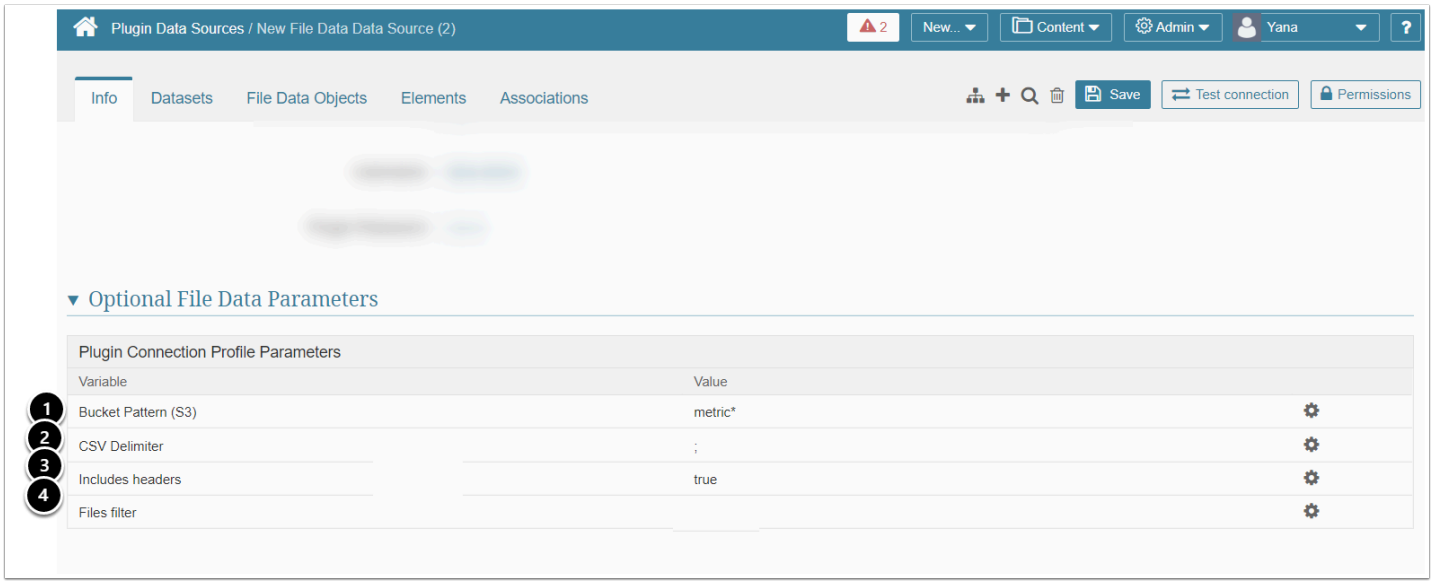


Specify the Required Parameters:

1. **Data Source Name:** will default but you may modify it
2. **Root location:** URL of the directory containing files from which data will be extracted
3. **Username/Plugin Password**:
    - Are not required when the local file system is used

- If access to the **ftp/sftp server** is password-protected, enter the above credentials in the corresponding fields
- **S3 server** requires *accessKey* and *secretKey* that have to be provided in the Username and Plugin Password fields. respectively

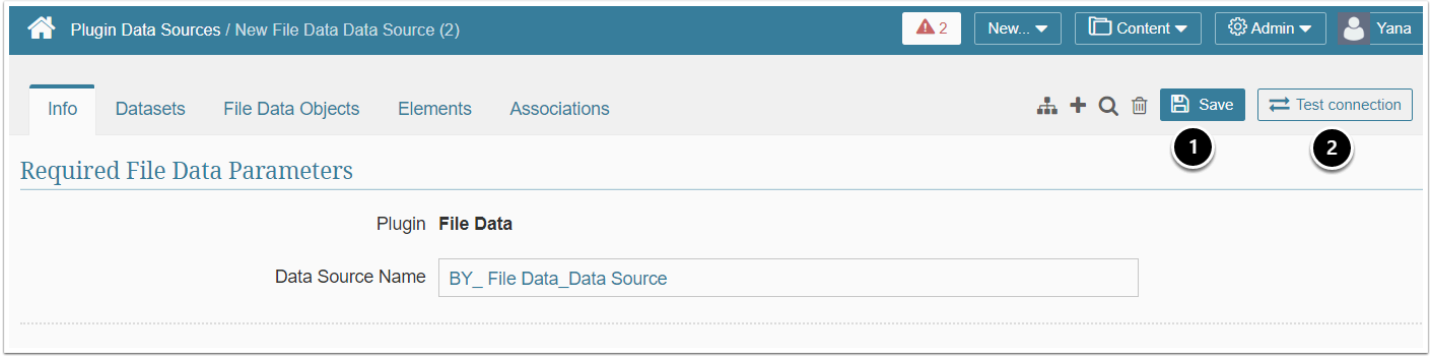
3.2. Optional File Data Parameters (Variables)



You may include the following Optional File Data Parameters:

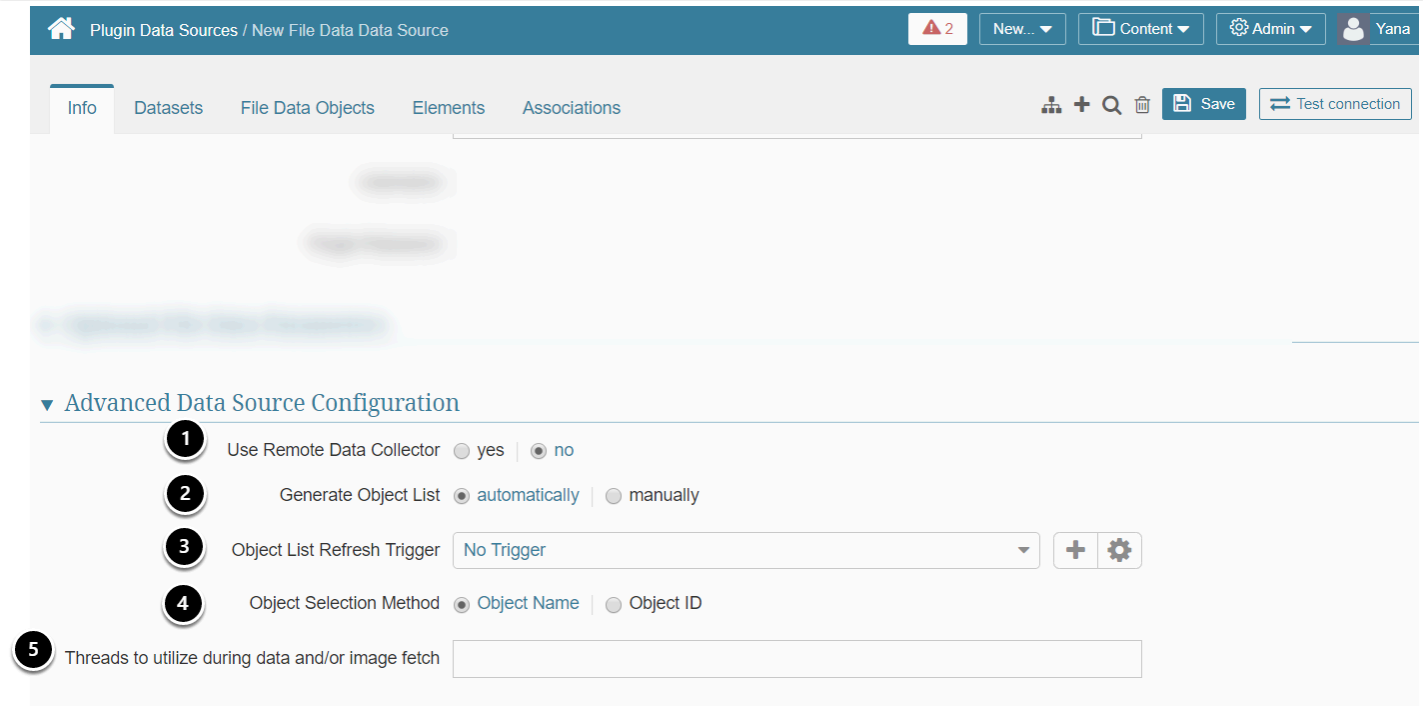
1. **Bucket Pattern (S3):** is used to filter high-level directories (buckets) on the Amazon S3 server
   1. Based on this filter, Bucket Objects will be collected by the File Data Plugin
2. **CSV Delimiter:** if values in your files are separated by delimiters other than the default comma (","), you can change the CSV Delimiter Variable to the required value
3. **Includes headers:** allowed values are "true"/"false", "yes"/"no", "1"/"0"
   1. If your file contains headers and you would like to include them in your File Data Report, set Variable values to "true", "yes", "1"
4. **Files filter:** can be employed to collect files according to the required value into the list of File Data Reports that are used to build other elements in Metric Insights

4. Save your entries and Test Connection



If your connection is successful, you may move on to **Advanced settings**.

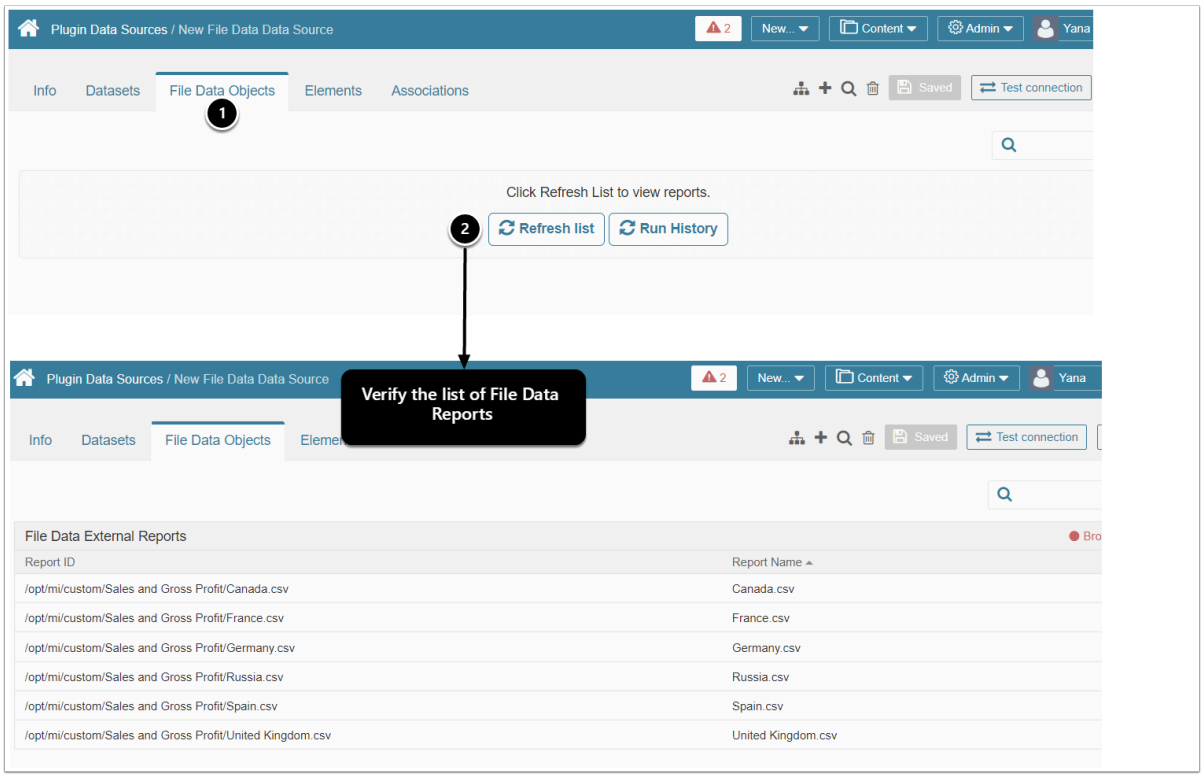
5. Advanced Configuration



1. **Use Remote Data Collector:** is set to "no" by default
   1. If required, switch to "yes" and add a Remote Data Collector by clicking **[+New Remote Collector]** under **Remote Collectors** settings
2. **Generate Object List**
   - *automatically*: all Reports are going to be fetched by the system
   - *manually*: Reports may be added one-by-one or via CSV file
3. **Object List Refresh Trigger:** from the drop-down, select the Trigger that will be used to fetch data via the File Data plugin

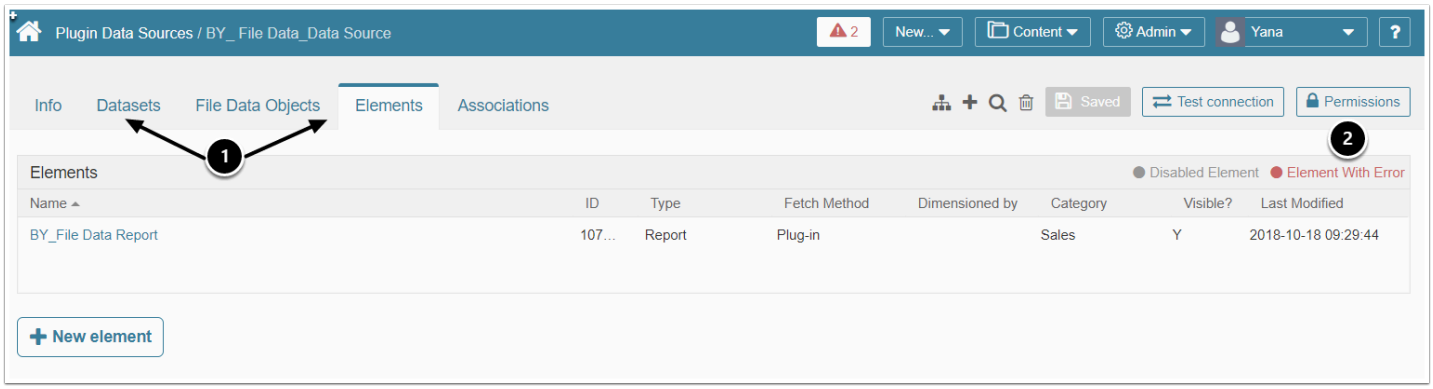
4. **Object Selection Method:** specify how File Data Reports will be fetched
5. Optionally, state the maximum number of concurrent **Threads to utilize during data and/or image fetch** to be used in background processing when the system updates Reports for this Data Source
    1. If you do not specify any value for this setting, batch data collection processing will be single-threaded

6. Obtain a list of External Reports



1. Go to **File Data Objects** tab
2. To obtain a list of File Data External Reports, click the **[Refresh list]** button

7. Other Settings



1. You can create Datasets or Elements directly from the respective tabs
2. Click **Permissions** to assign them to Groups or Power Users

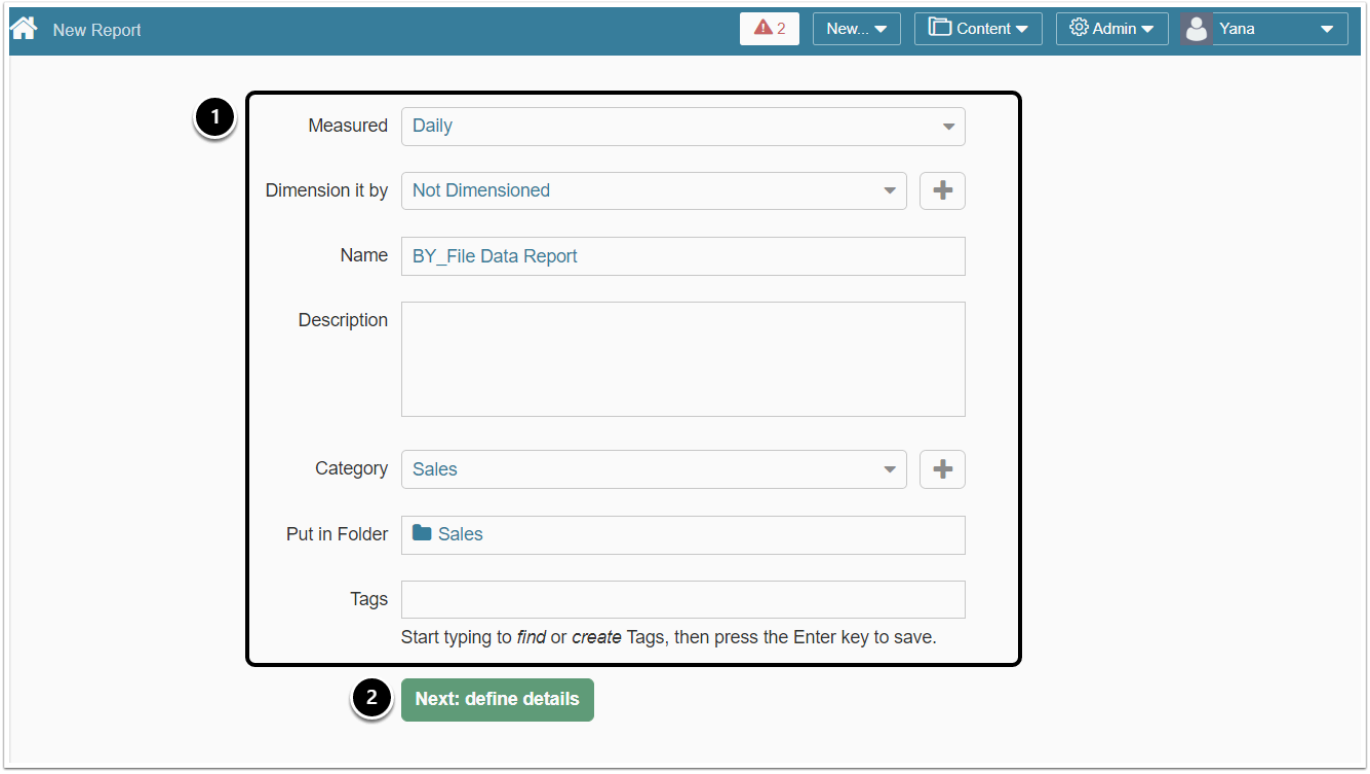
What's next?

How to collect data from Beckon

16.2 How to collect data with File Data

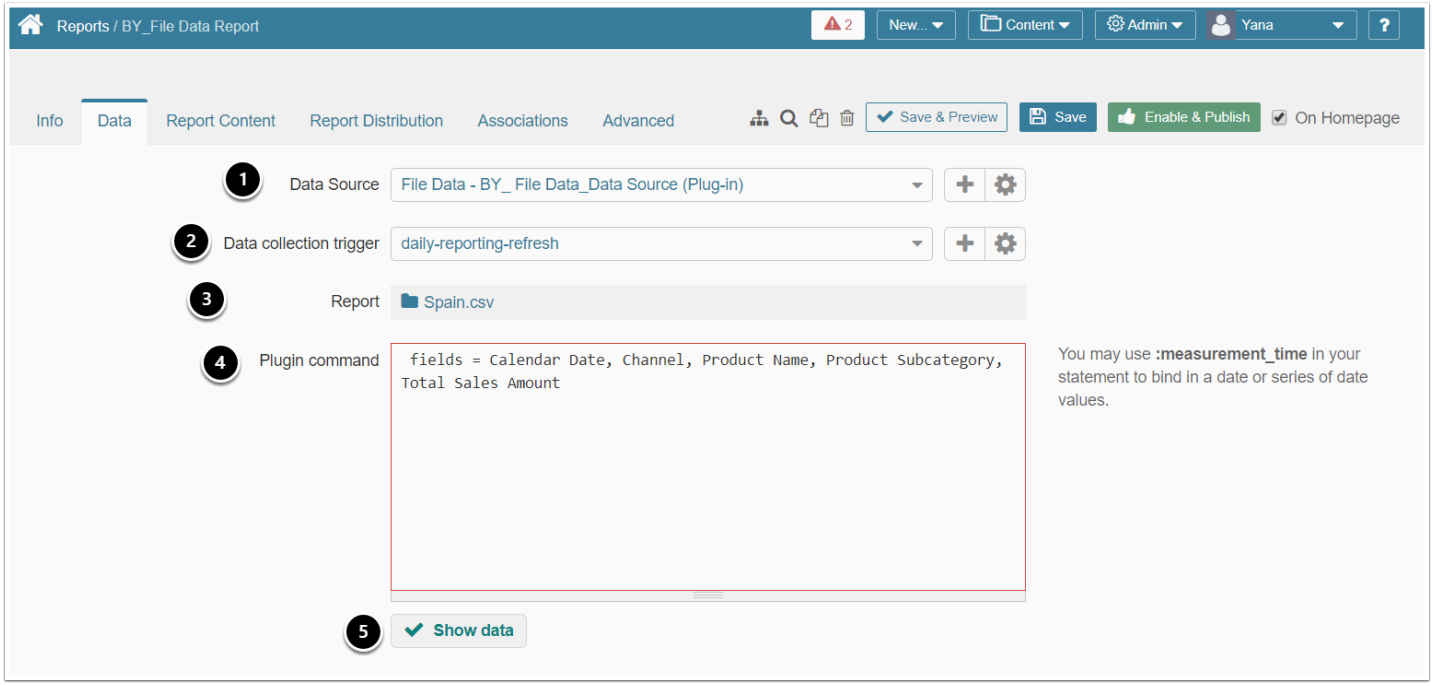
This article will show you how to create a Metric or Report using a File Data Object as a Data Source. It assumes that you have already established connectivity to the File Data Data Source.

1. Access New > Report



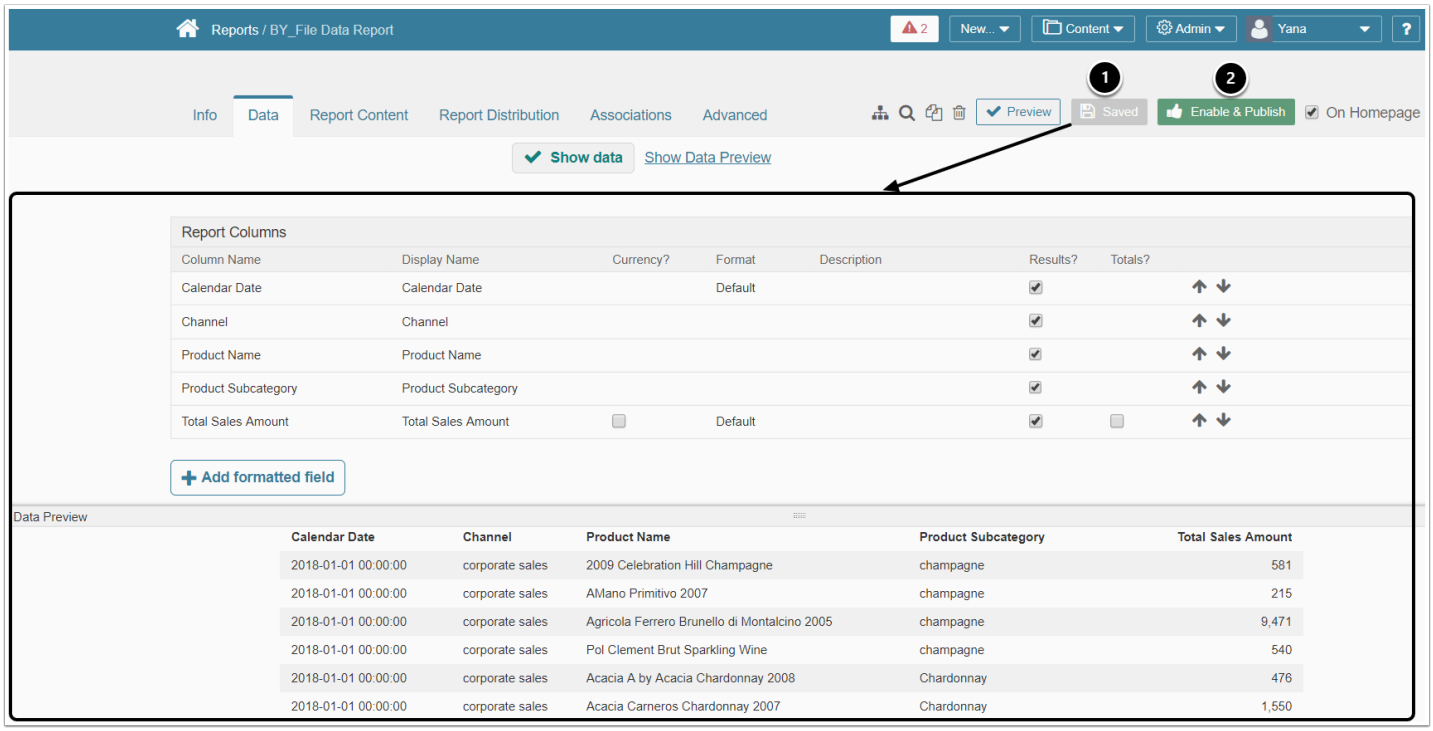
1. Define the Basics for your Report
2. To continue defining data collection details, click **Next: Define Details**

2. Full Editor displays the Data Collection tab



1. **Data Source**: select the connection profile you have created for the File Data plugin
2. **Data collection trigger**: Specify the Trigger that will be used to collect data for your Report
3. **Report:** select an External Report that should serve as a basis of a new internal Report
4. Input **Plugin Command:** enter the command in MIQL (**M**etric**I**nsights **Q**uery **L**anguage) listing all the data you would like to fetch
5. Once you are ready with you command, click **Show Data**

3. Plug-in command will be validated and Data Collected on Save



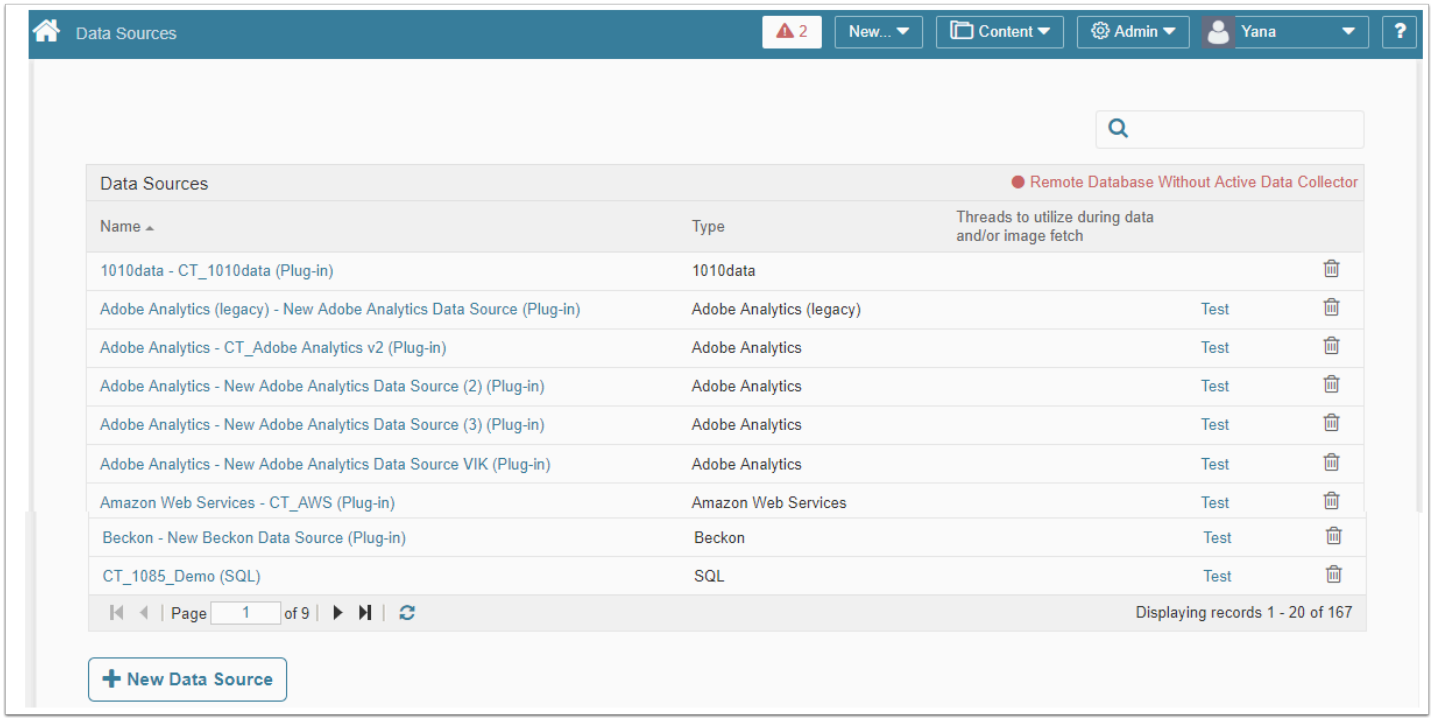
1. If the command is validated successfully, the **Report columns** and **Data Preview** are shown in the Data Preview section
2. At the upper right corner of the screen, click **Enable & Publish**

17. Sourcing Data using File Metadata Plugin

17.1 Establish connectivity to File Metadata

This article describes the process of creating a plugin Data Source that will allow to collect metadata (information on who created the files or folders and when, the time they were modified, ownership, versions and revisions) to be used in building elements using Metric Insights tools.

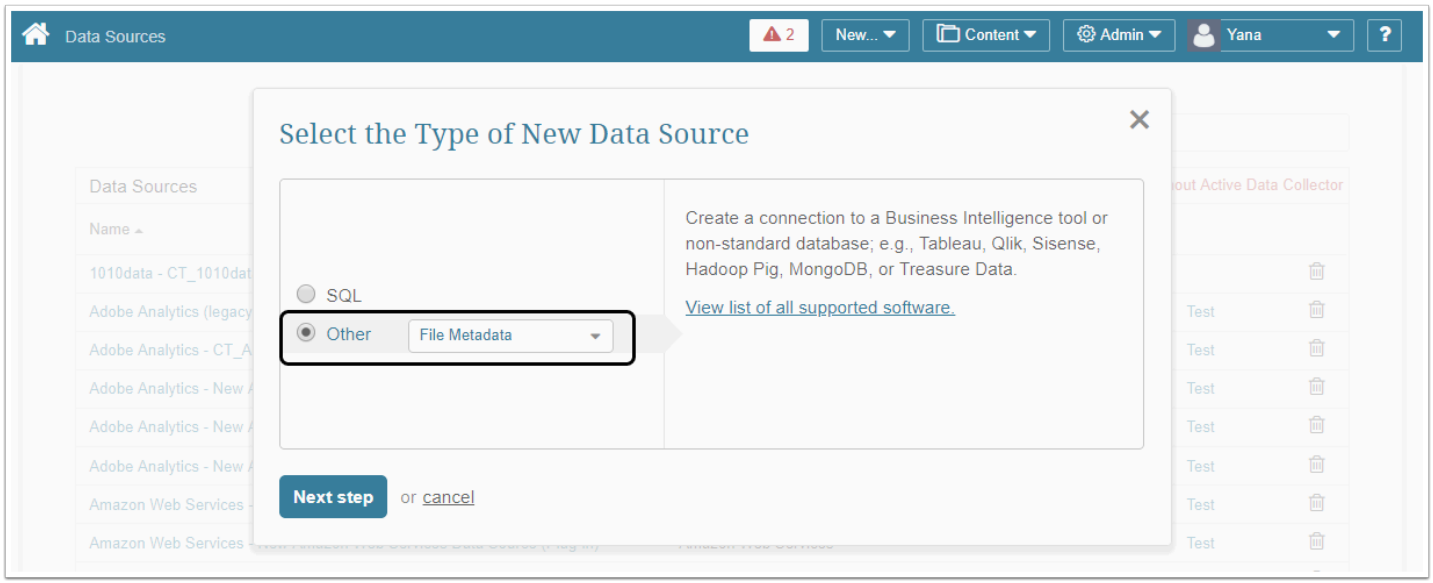
1. Access Admin > Data Sources



At the bottom of the screen click **[+ New Data Source].**

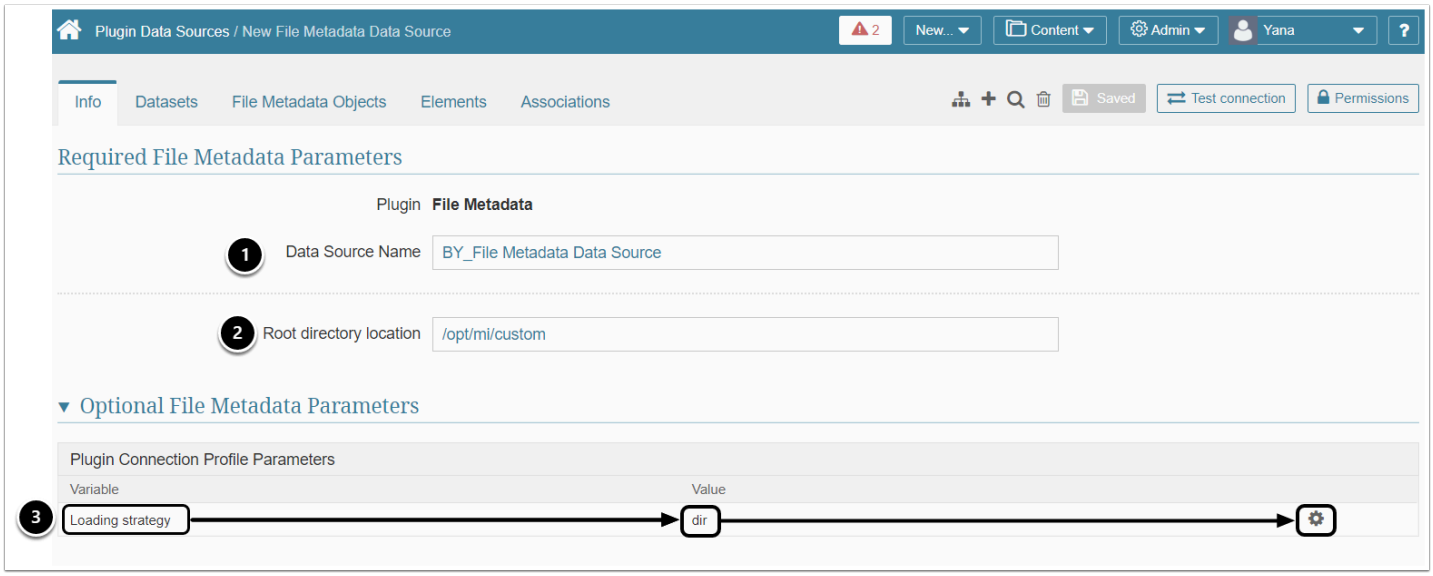
The *Select the Type of New Data Source* pop-up opens.

2. Select "Other" Data Source Type and choose "File Metadata" from the drop-down list



Proceed with creating a Data Source by moving to the **Next step**.

3. Provide File Metadata Parameters



Specify how to connect to File Metadata. The **parameters** you need to include are:

1. **Data Source Name:** will default but you may modify it
2. **Root directory location:** URL of the directory containing files/folders whose metadata will be collected

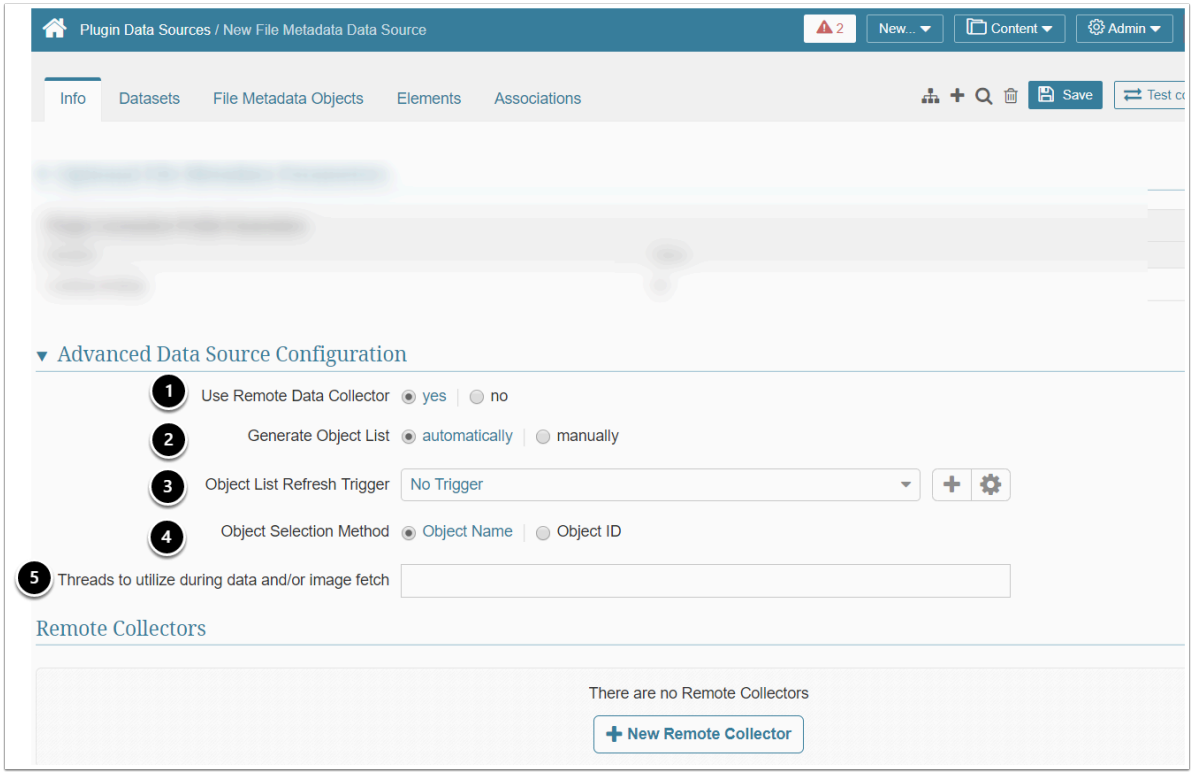
   1. This directory may be located on the MI server or a server where the Remote Data Collector is installed

3. **Loading strategy variable** (optional parameter)**:** determines which entity (a single file or a folder/directory) will be used as a Report in File Metadata. You may change the default Value if required. Available values are:
   1. "dir" (default, denotes a directory)
   2. "single_file" or "single" (denotes a file)

4. **Save** your entries and **Test Connection**

If your connection is successful, you may move on to **Advanced settings**.

4. Advanced Configuration



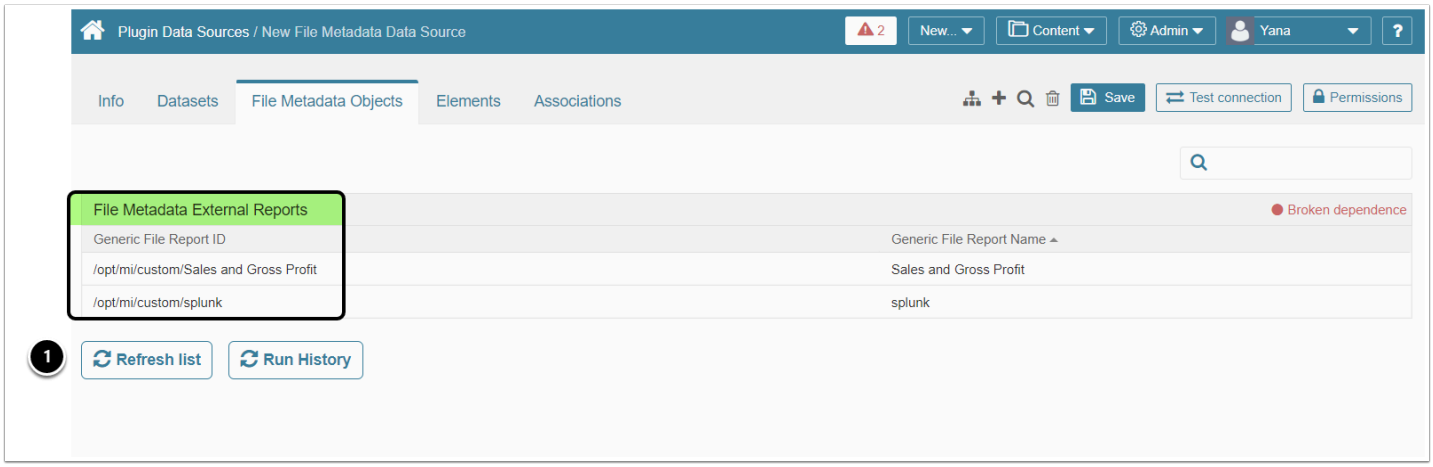
1. **Use Remote Data Collector:** is set to "no" by default
   1. If required, switch to "yes" and add a Remote Data Collector by clicking **[+New Remote Collector]** under **Remote Collectors** settings

2. **Generate Object List**
   - *automatically*: all Reports are going to be fetched by the system
   - *manually*: Reports may be added one-by-one or via CSV file

3. **Object List Refresh Trigger:** from the dropdown, select the Trigger that will be used to fetch data via the File Metadata plugin
4. **Object Selection Method:** specify how File Metadata Reports will be fetched

5. Optionally, state the maximum number of concurrent **Threads to utilize during data and/or image fetch** to be used in background processing when the system updates Reports for this Data Source
    1. If you do not specify any value for this setting, batch data collection processing will be single-threaded

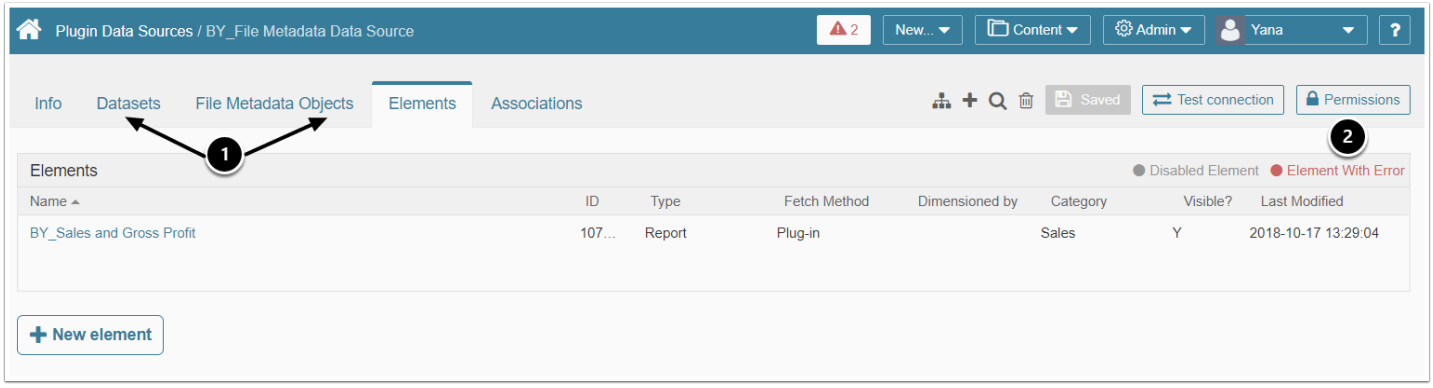
5. Get a list of External Reports



Go to **File Metadata Objects** tab:

1. To obtain a list of File Metadata External Reports, click the **[Refresh list]** button

6. Other Settings



1. You can create Datasets or Elements directly from the respective tabs
2. Click **Permissions** to assign them to Groups or Power Users

What's next?

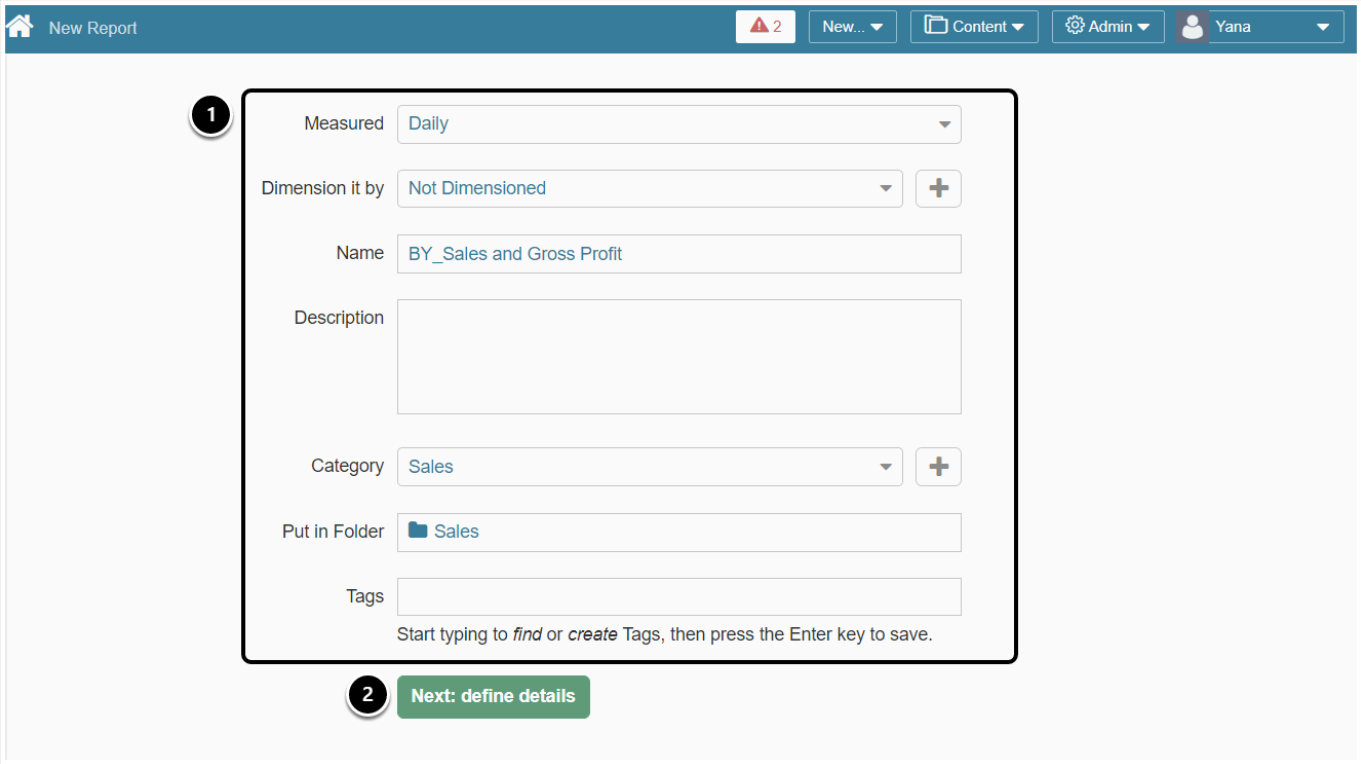
[How to collect data from Beckon]

17.2 How to collect data with File Metadata

This article explains how to create a Metric or Report using a File Metadata Object as a Data Source. It assumes that you have already established connectivity to the File Metadata Data Source.

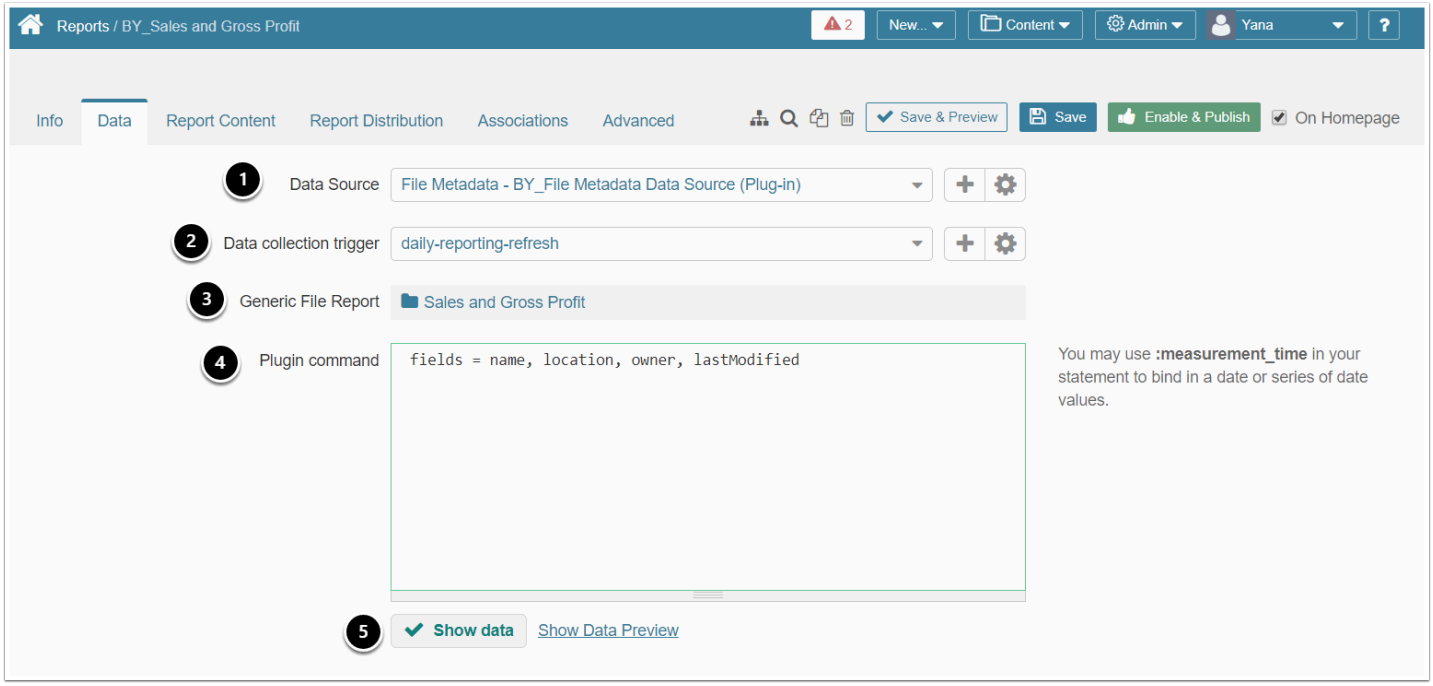
[Release 6.x]: 'Remote Data Collector' renamed to 'Remote Data Processor'

1. Access New > Report



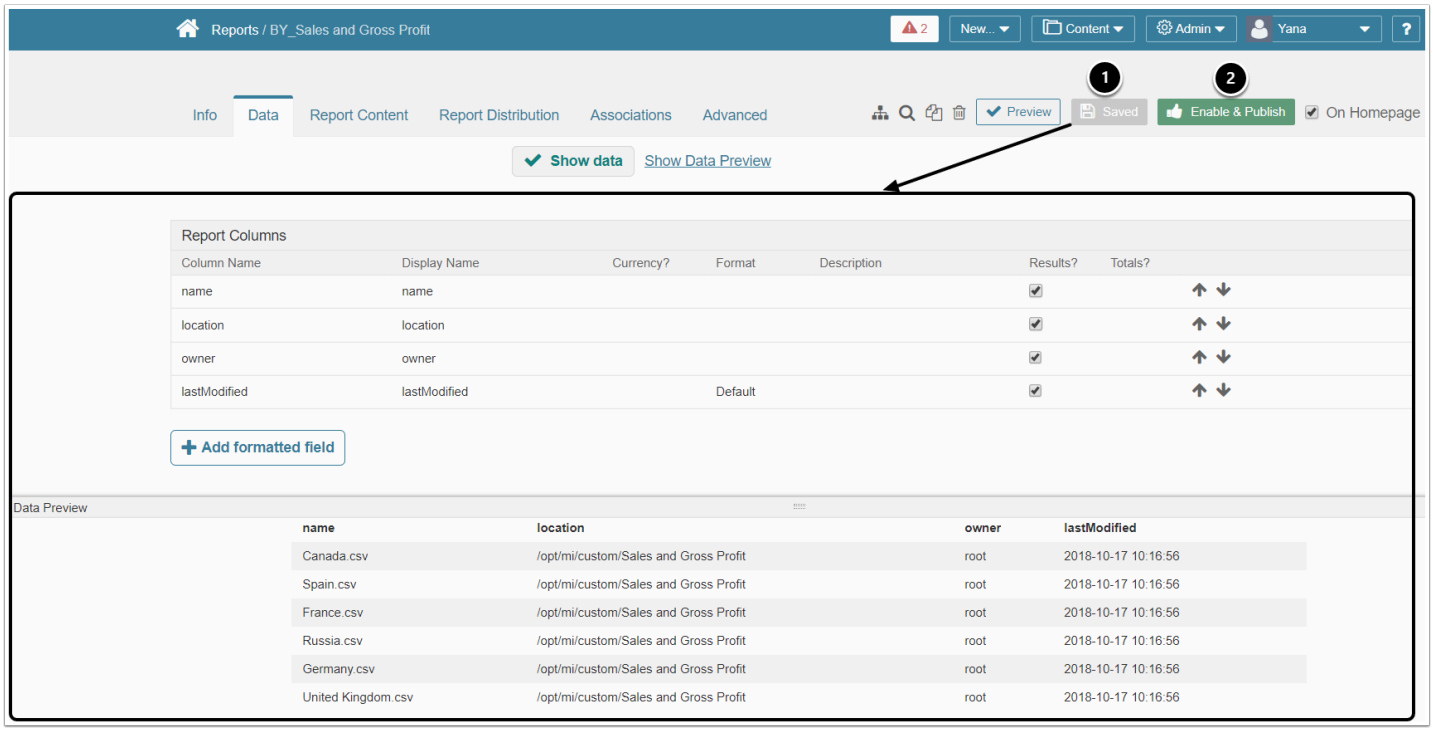
1. Define the Basics for your Report
2. To continue defining data collection details, click **Next: Define Details**

2. Full Editor displays the Data Collection tab



1. **Data Source**: select the connection profile you have created for the File Metadata plugin
2. **Data collection trigger**: Specify the Trigger to be used to collect data for your Report
3. **Generic File Report:** select an External Report to serve as a basis of a new internal Report
4. Input **Plugin Command:** enter the command in MIQL (**M**etric**I**nsights **Q**uery **L**anguage) listing all the data you would like to fetch
5. Once you are ready with you command, click **Show Data**

3. Plug-in command will be validated and Data Collected on Save



1. If the command is validated successfully, the **Report columns** and **Data Preview** are shown shown below in the Data Preview
2. At the upper right corner of the screen click **Enable & Publish**.

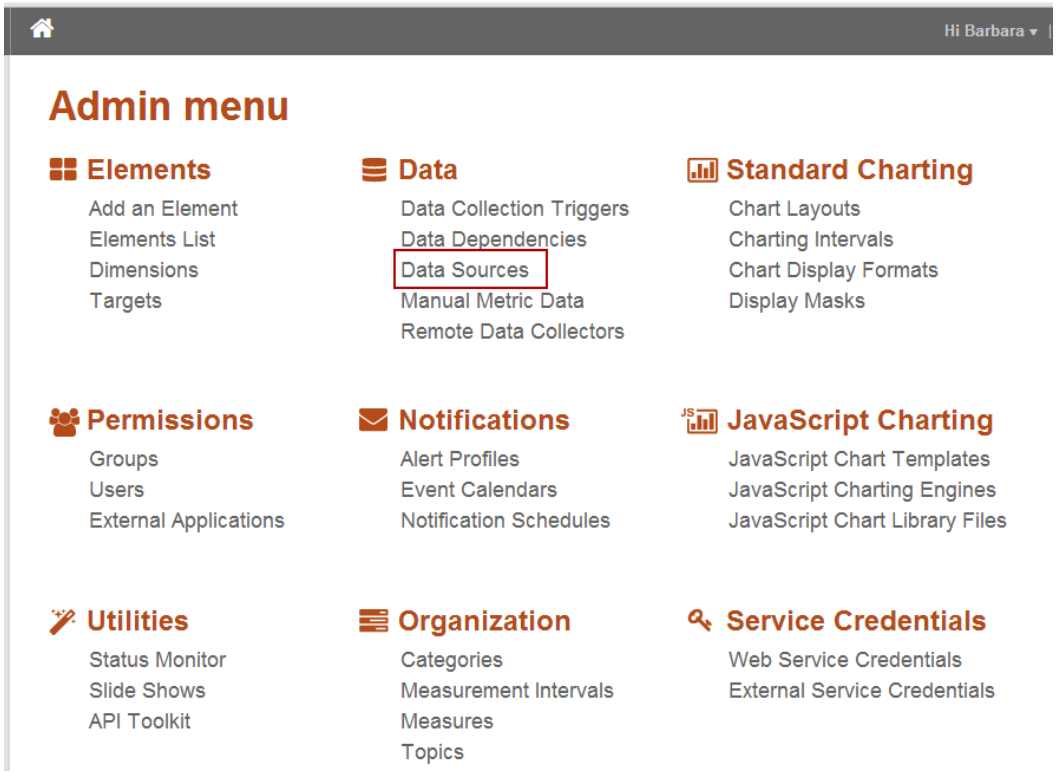
18. Sourcing Data from Fogbugz

18.1 Establish connectivity to FogBugz

This article describes the process of creating a new **Plug-in Data Source** for accessing FogBugz so you can query the **Data Source** for a new or existing element.

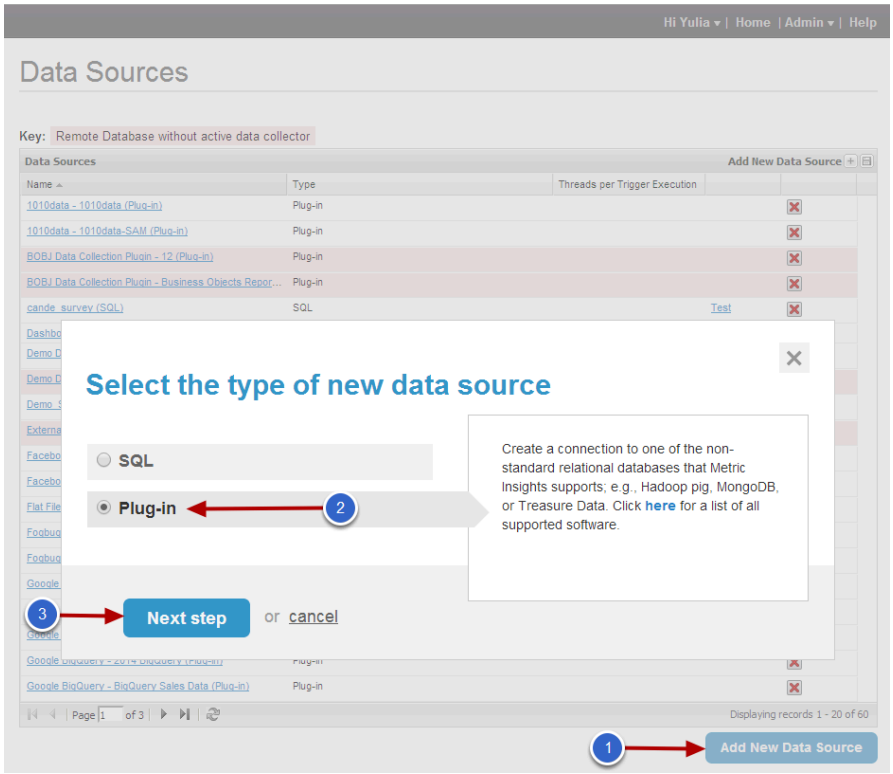
FogBugz is an integrated web-based project management system with bug/issue tracking, discussion forums, customer relationship management, and evidence based scheduling.

1. Open Admin Menu



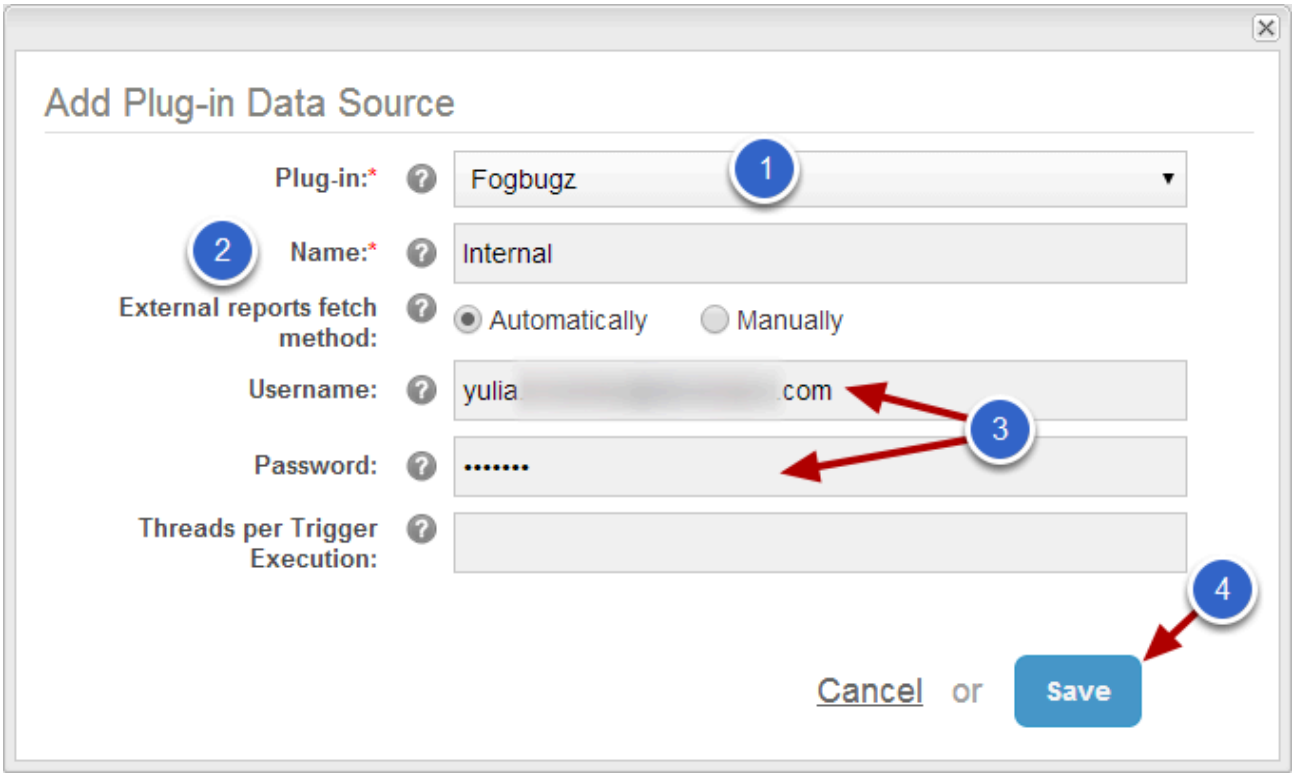
Click the **Data Sources** link to access *Data Sources* list

2. Add new Data Source



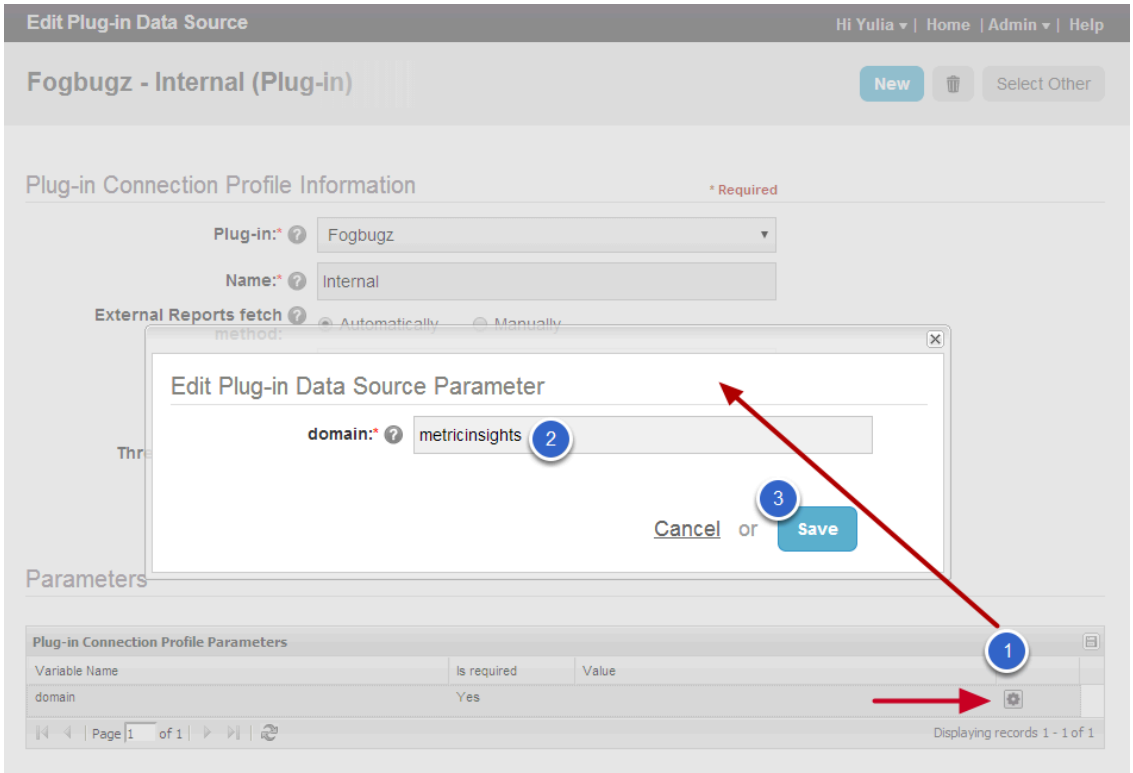
1. Click the **Add New Data Source** button
2. Select **Plug-in** as type
3. Click **Next step** button

3. Define main settings



1. Select 'FogBugz' plug-in in the drop-down list
2. Enter **Name**
3. Complete credential fields
4. Click **Save**
5. *Edit Plug-in Data Sourc*e page opens

4. Set required parameter

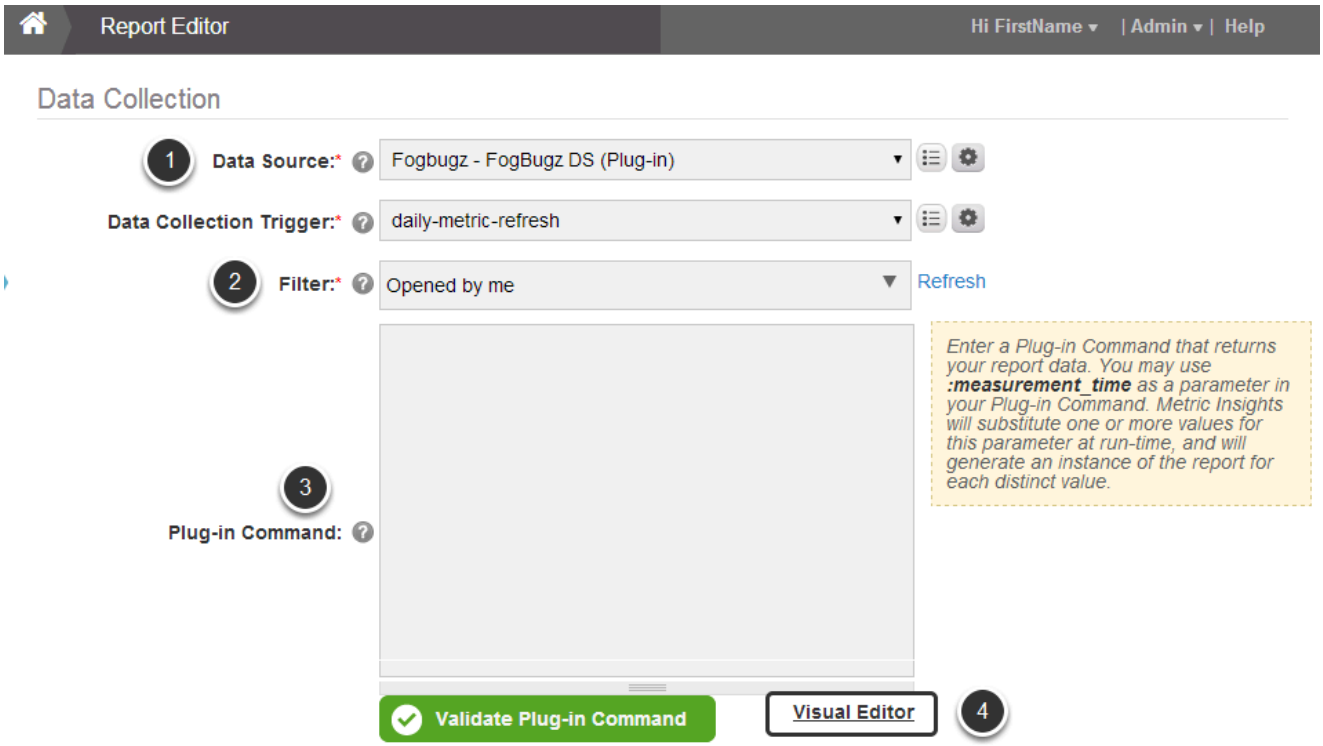


1. Click the gear icon to open the *Edit Plug-in Data Source Parameter* pop-up
2. **domain**: Enter the domain name of your FogBugz account
3. Click **Save**

18.2 How to collect data from Fogbugz

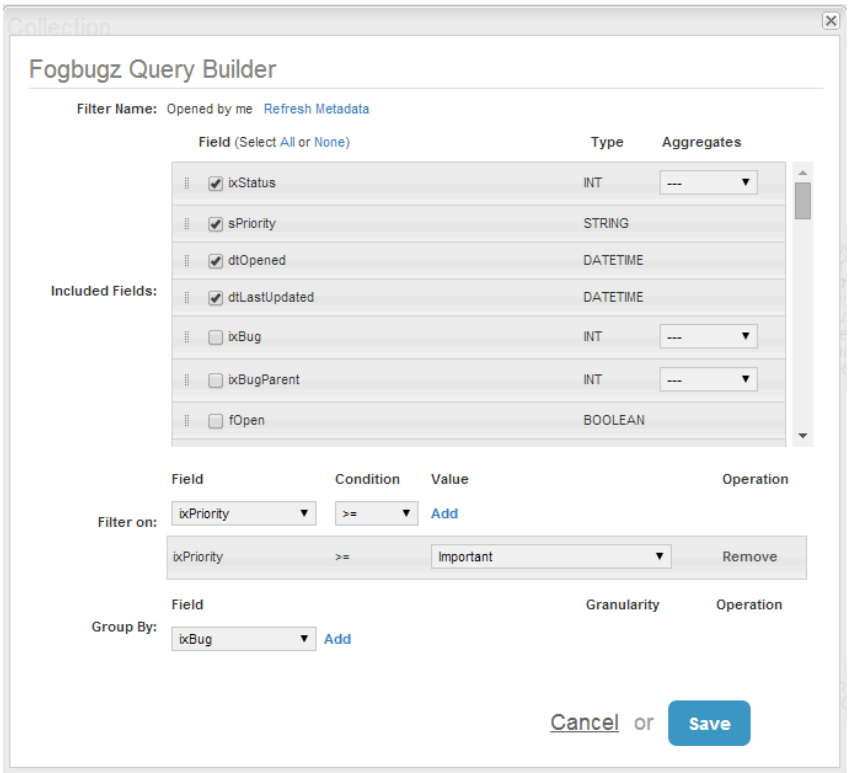
This article will show you how to create an Element using a Fogbugz plug-in as a data source. It assumes that you have already established connectivity to your Fogbugz account.

1. Select either 'Report' or 'Metric' from New menu - (report example)



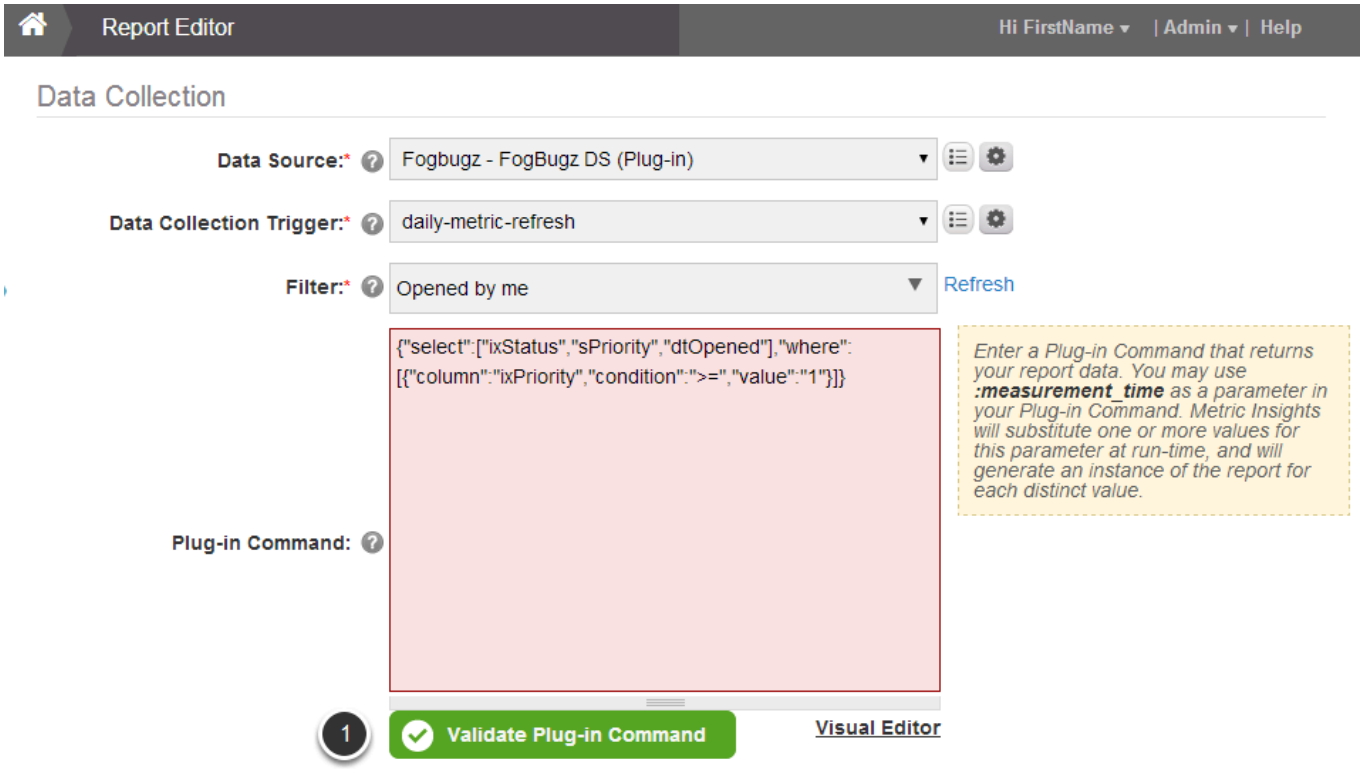
1. Choose Fogbugz as your **Data Source**.
2. Select one of the **Filter** from the drop-down.
3. Input your **Plug-in Command** manually, OR
4. Use **Visual Editor.**

2. Fogbugz Query Builder is called by Visual Editor link



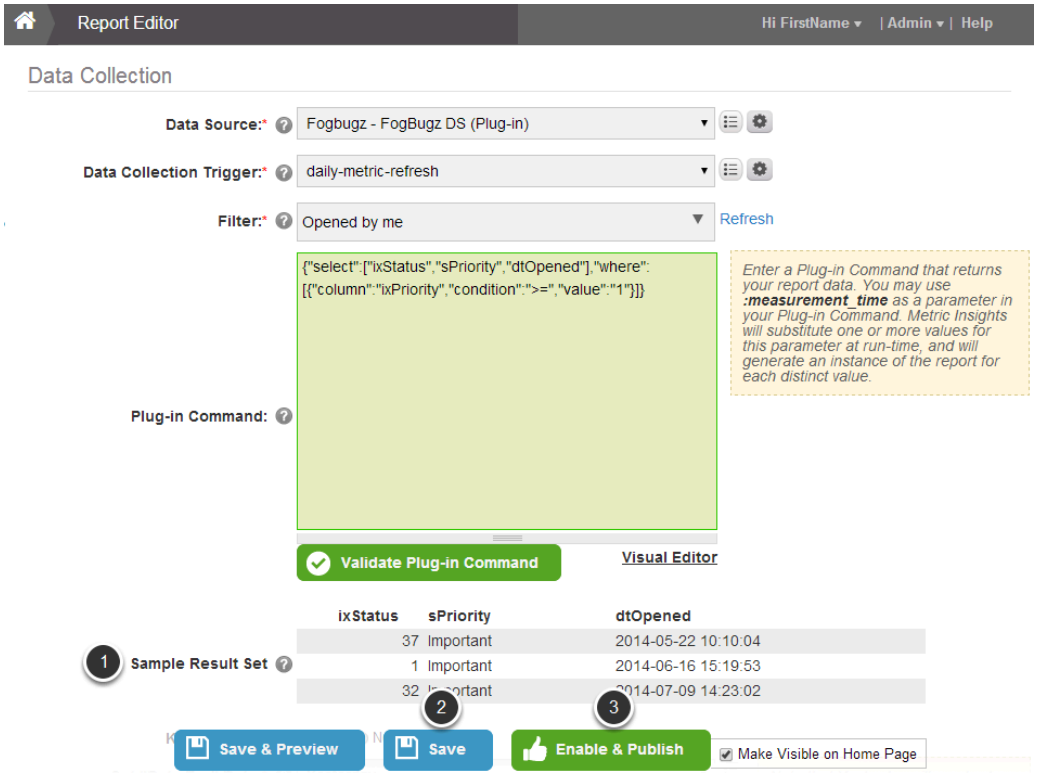
You can select fields in **Included Fields** section, filter selection or group fields based on **Filter on** and **Group By** options.

3. Validate your Plug-in Command



1. **Validate** your Plug-in Command

4. Enable and Publish Report



1. Upon successful validation, you will see **Sample Result Set**.
2. **Save** the Report and continue to configure it OR
3. If configuring of the Report completed, **Enable and Publish.**

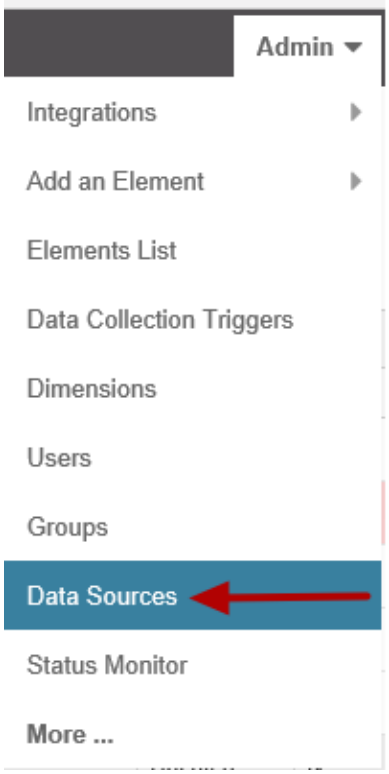
19. Sourcing Data from Graphite

19.1 Establish Connectivity to Graphite

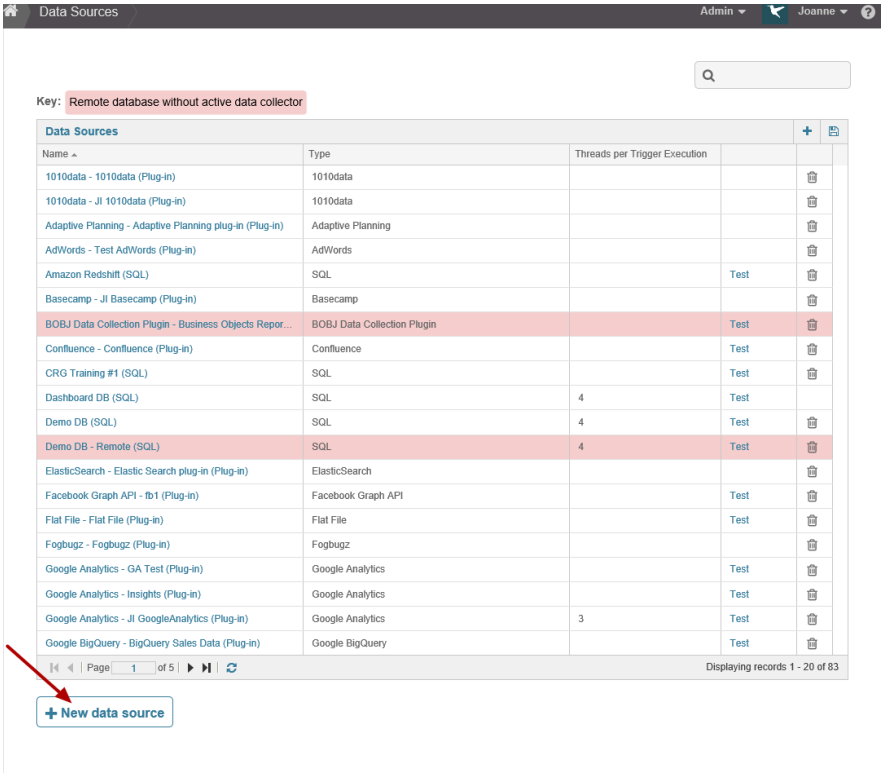
This article describes how to connect to **Graphite** in order to use their reports as Data Sources in Metric Insights.

General instructions on setting up data sources based on plug-ins can be found here.

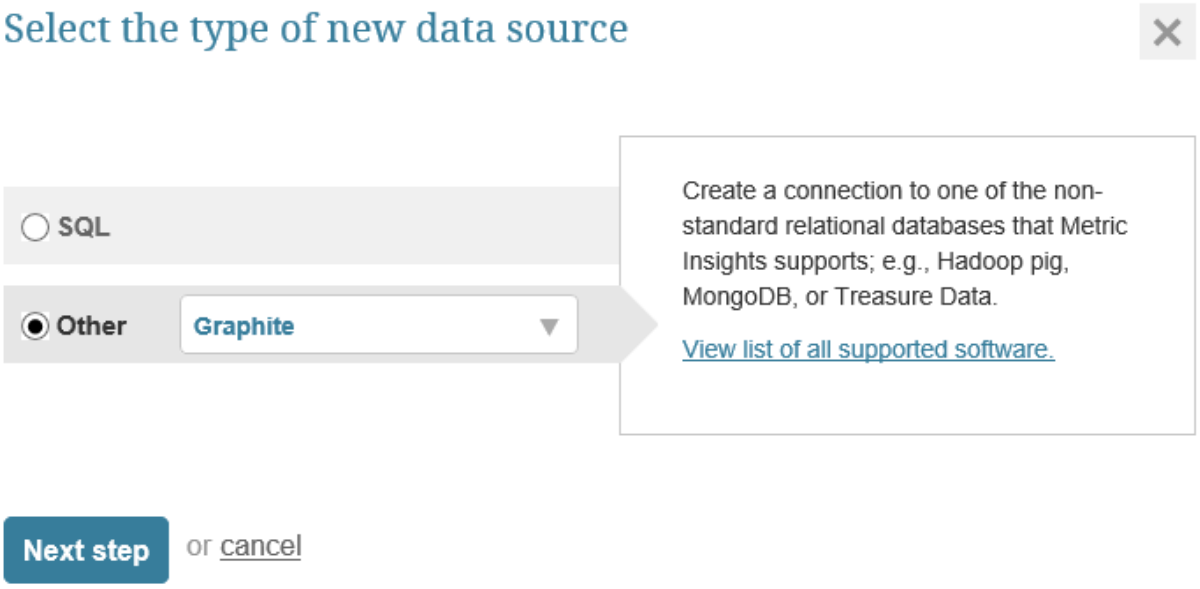
1. Select Data Sources from Admin drop-down



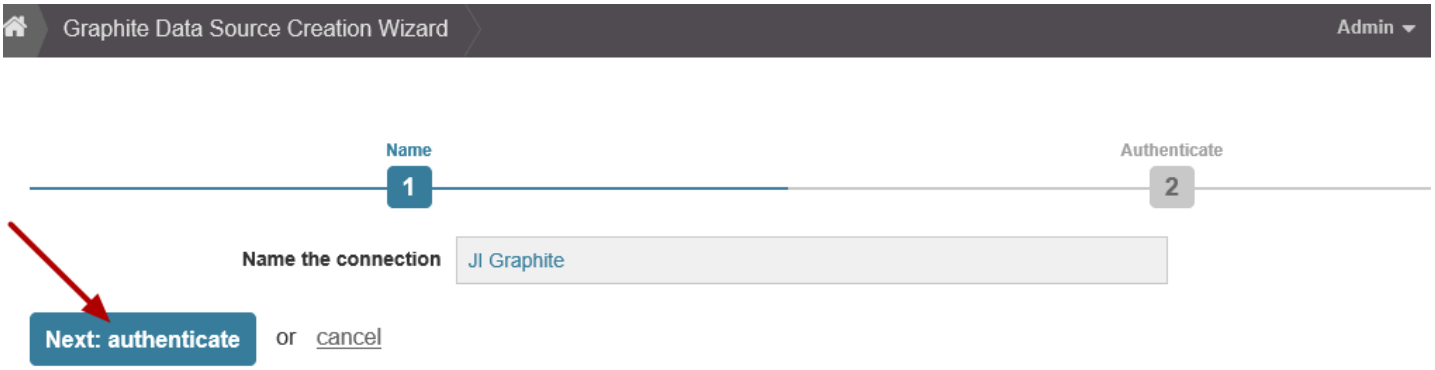
2. Add New Data Source



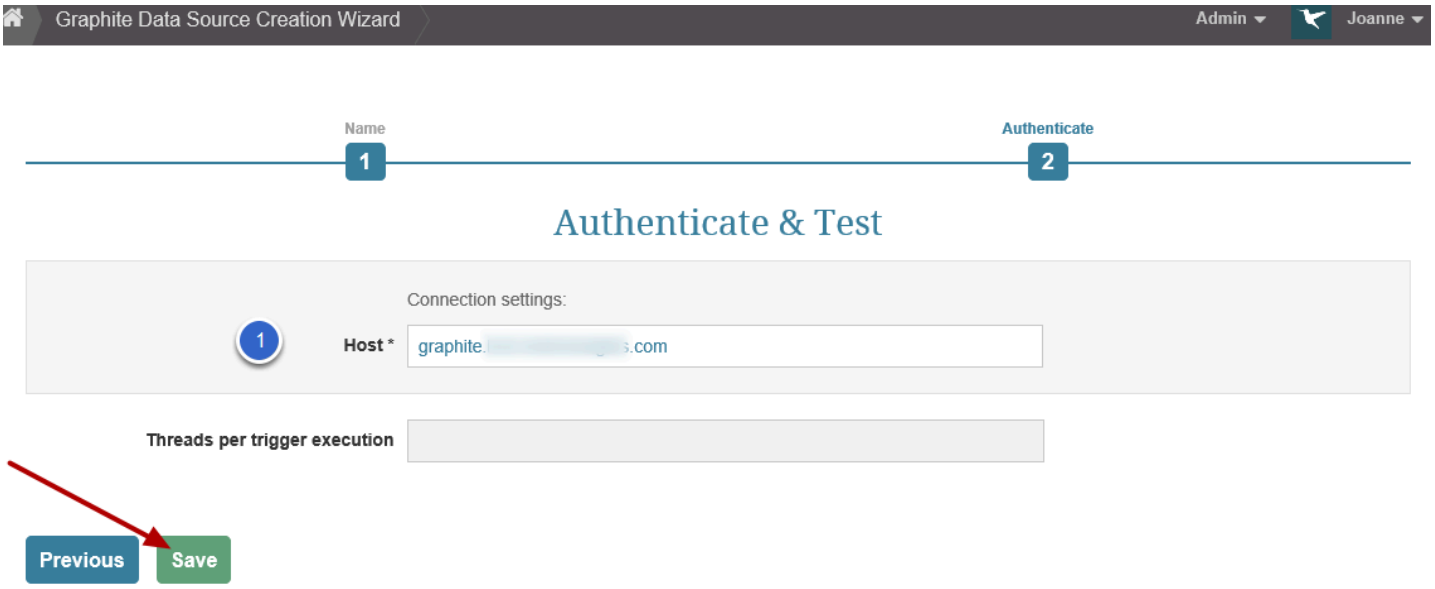
3. Select "Other" Data Source Type and choose "Graphite" from the drop-down



4. Input a unique Name and next step



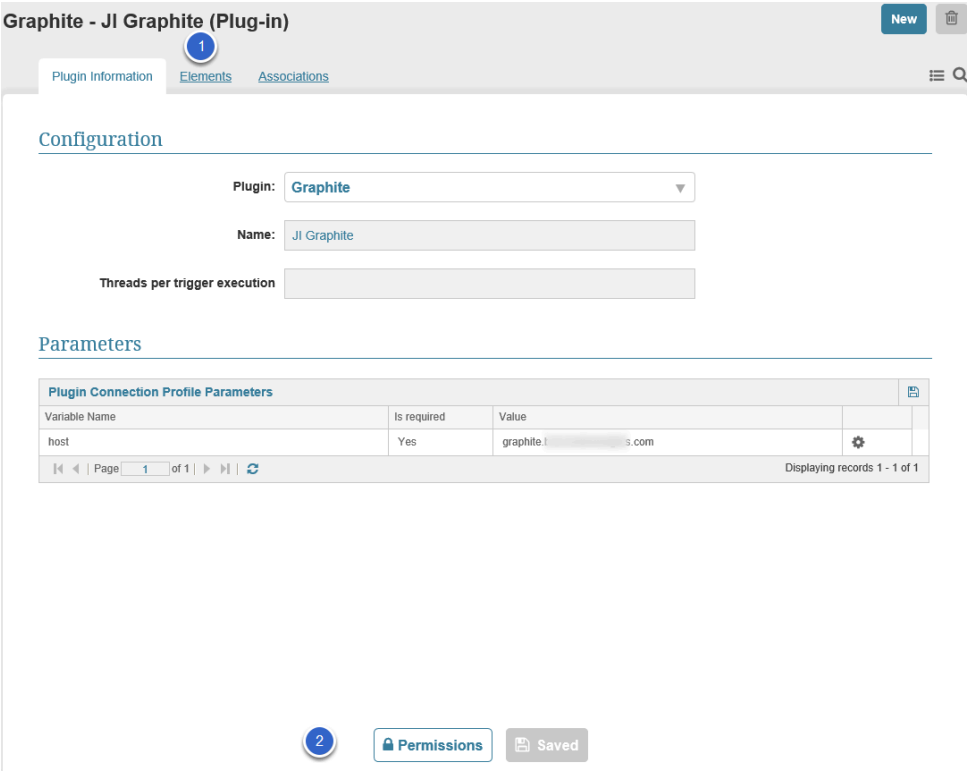
5. Input Host name



1. Enter **Host** parameter

Save

6. Full Data Source Editor displays



1. You can create **elements** directly from the Elements tab
2. You can assign **permissions** to Groups or Power Users here also

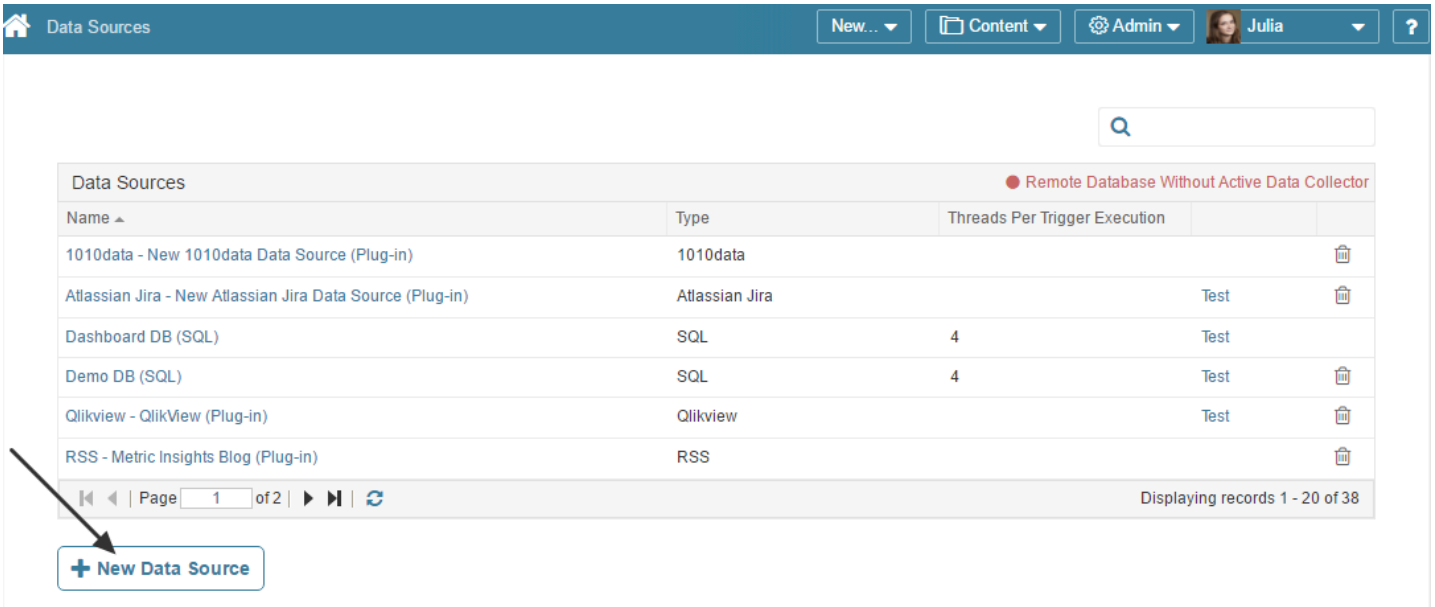
20. Sourcing Data from IBM CoreMetrics

20.1 Establish connectivity to IBM Coremetrics

This article describes how to connect to IBM Coremetrics in order to use their reports as Data Sources in Metric Insights.

General instructions on setting up data sources based on plugins can be found here.

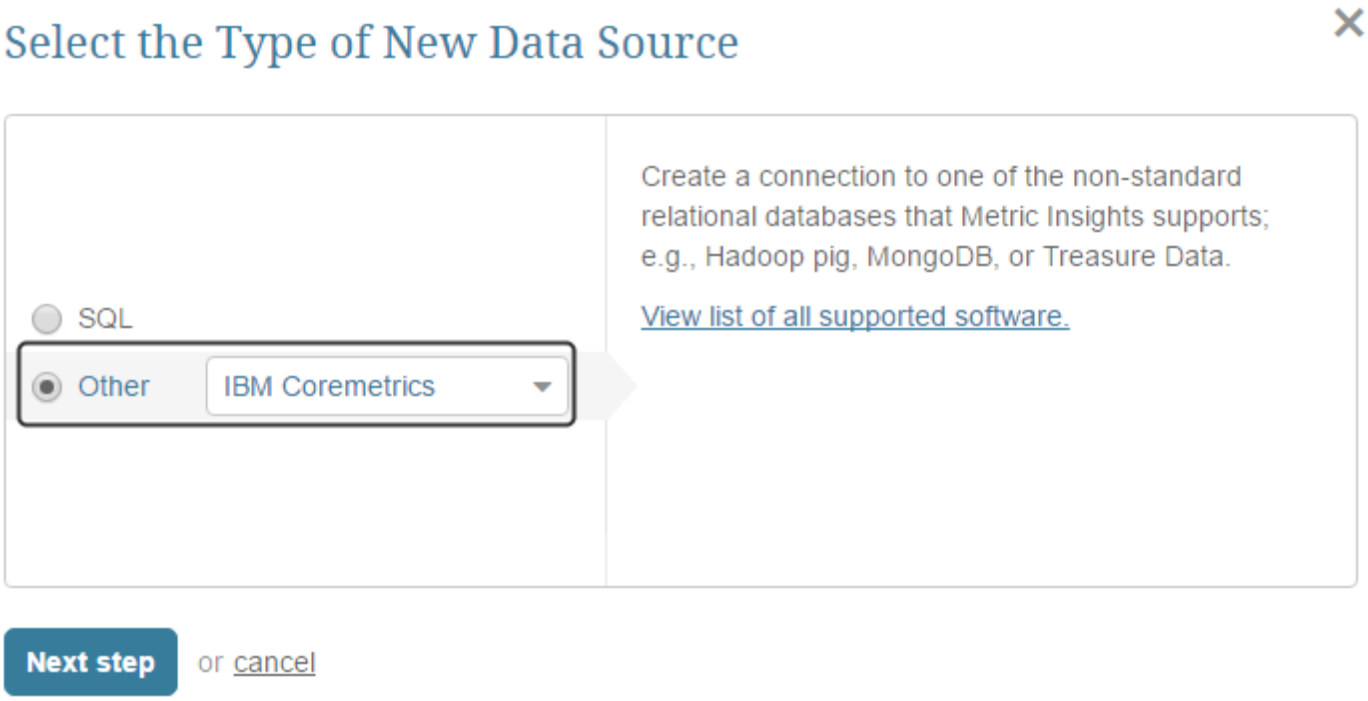
1. Add New Data Source



At the bottom of the screen click **[+ New Data Source].**

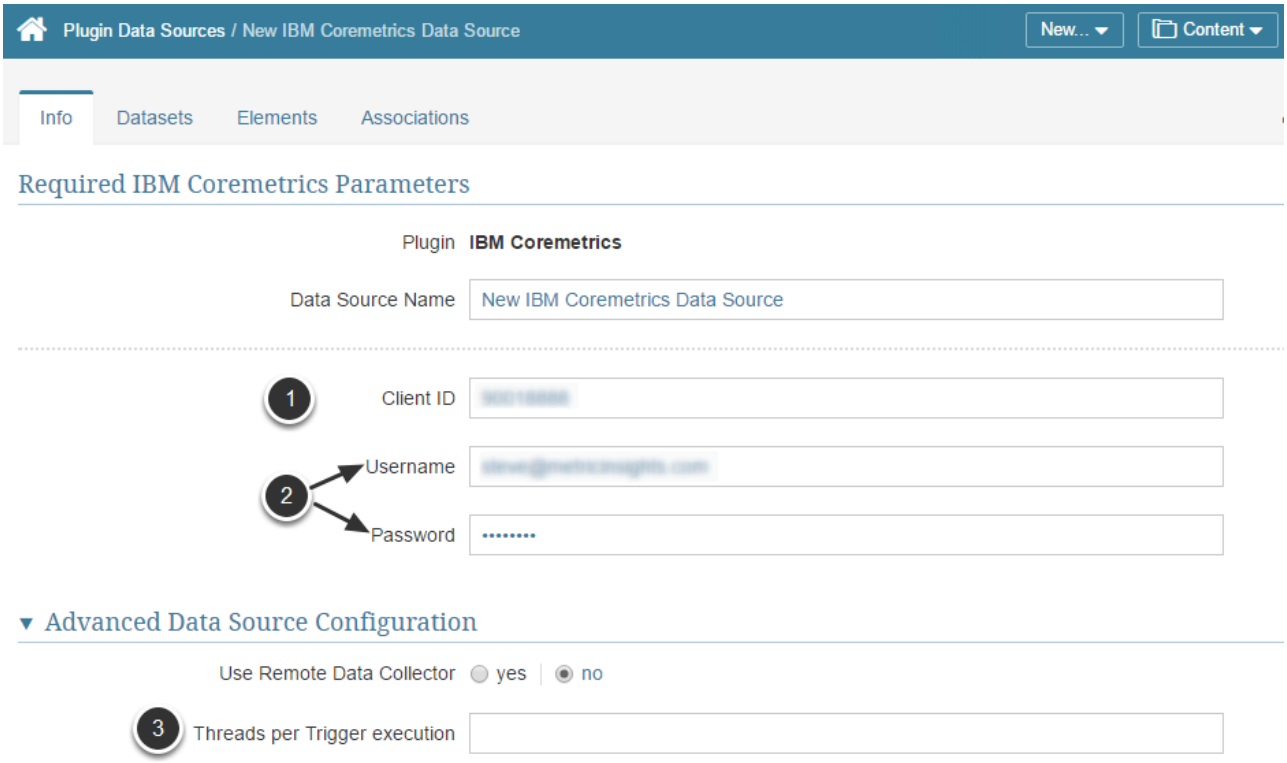
The *Select the Type of New Data Source* pop-up opens.

2. Select "Other" Data Source Type and choose "IBM Coremetrics" from the drop-down list



Proceed with creating a Data Source by moving to the **Next step**.

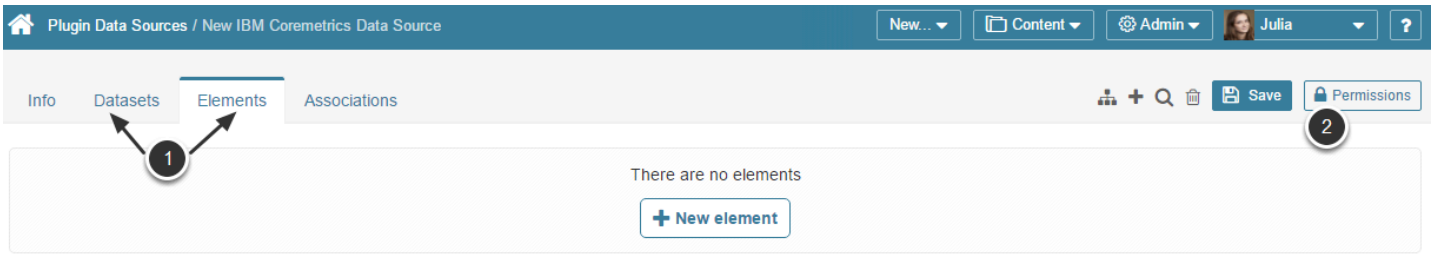
3. Authenticate and Test



1. **Client ID:** Unique 8-digit IBM Digital Analytics-assigned account code associated with a single analytics data warehouse and reporting instance.
2. **Username/Password:** Note that your **Username** must be in the same format that your IBM Coremetrics server uses for authentication.
3. Optionally, specify the maximum number of concurrent **Threads per Trigger execution** to be used in background processing when the system updates Metrics and Reports for this Data Source. If you do not specify any value for this setting, batch data collection processing will be single-threaded.

Save your entries.

4. Full Data Source Editor displays



1. You can create elements or Datasets directly from the respective tabs
2. Click **Permissions** to assign them to Groups or Power Users

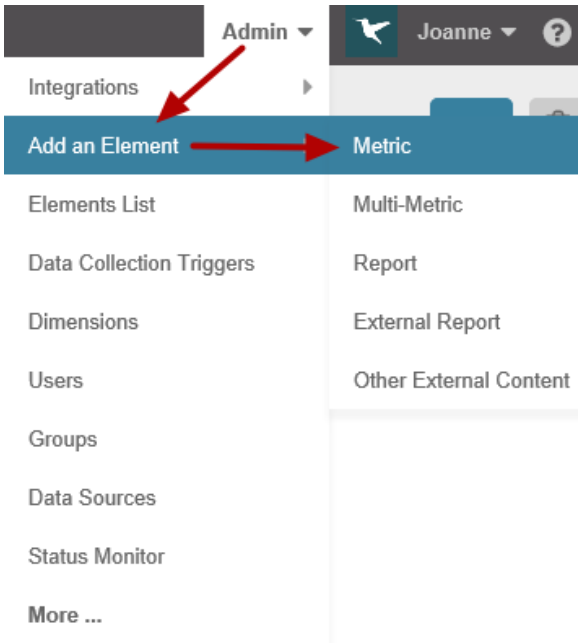
5. What's next?

How to collect data from IBM Coremetrics

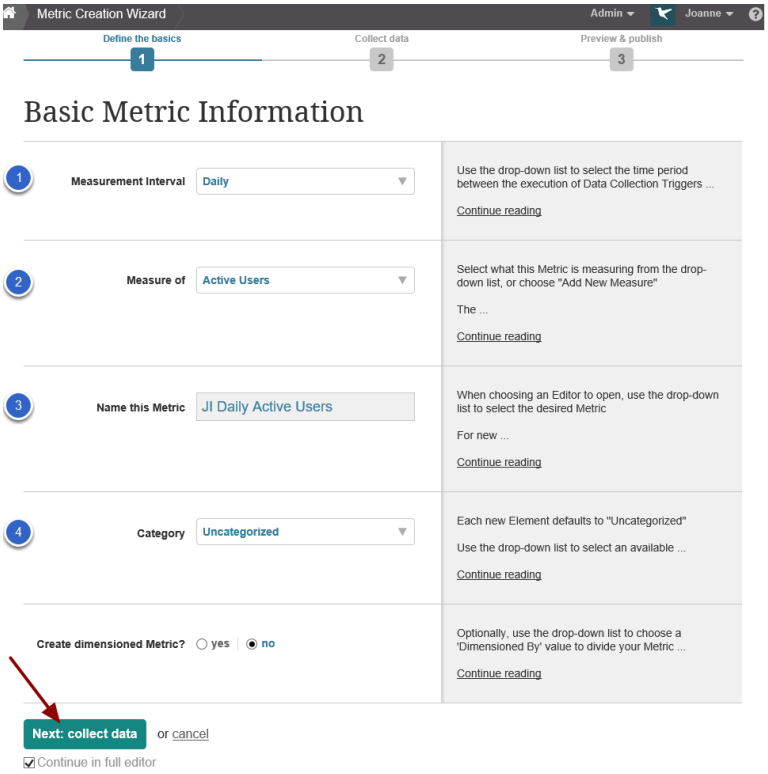
20.2 How to collect data from IBM Coremetrics

This article will show you how to create a Metric or Report using a **IBM Coremetrics** report as a data source. It assumes that you have already established connectivity to your **IBM Coremetrics** server.

1. Add a new element based on your IBM Coremetrics plug-in data source



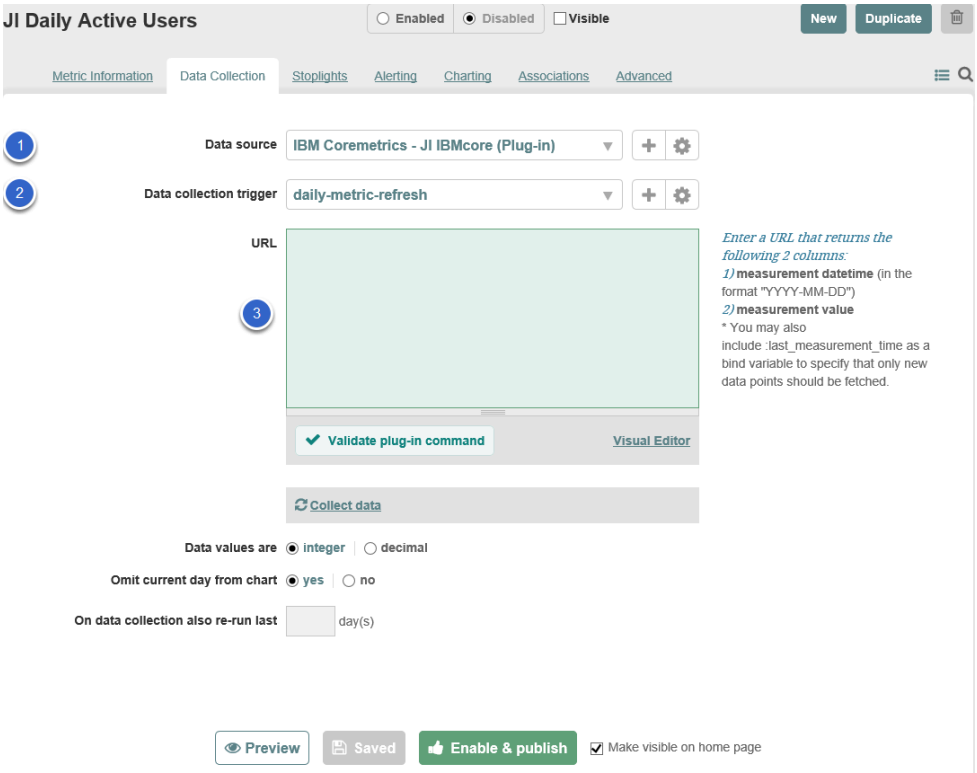
2. Provide basic information on Wizard (or Editor) - metric example



1. Select the **Measurement Interval** that applies to your element
2. Specify what this metric is **measuring**. If you do not see the measure that you want to use, you can create one from this drop-down
3. Give the element a unique **name**
4. Optionally, assign a **Category**

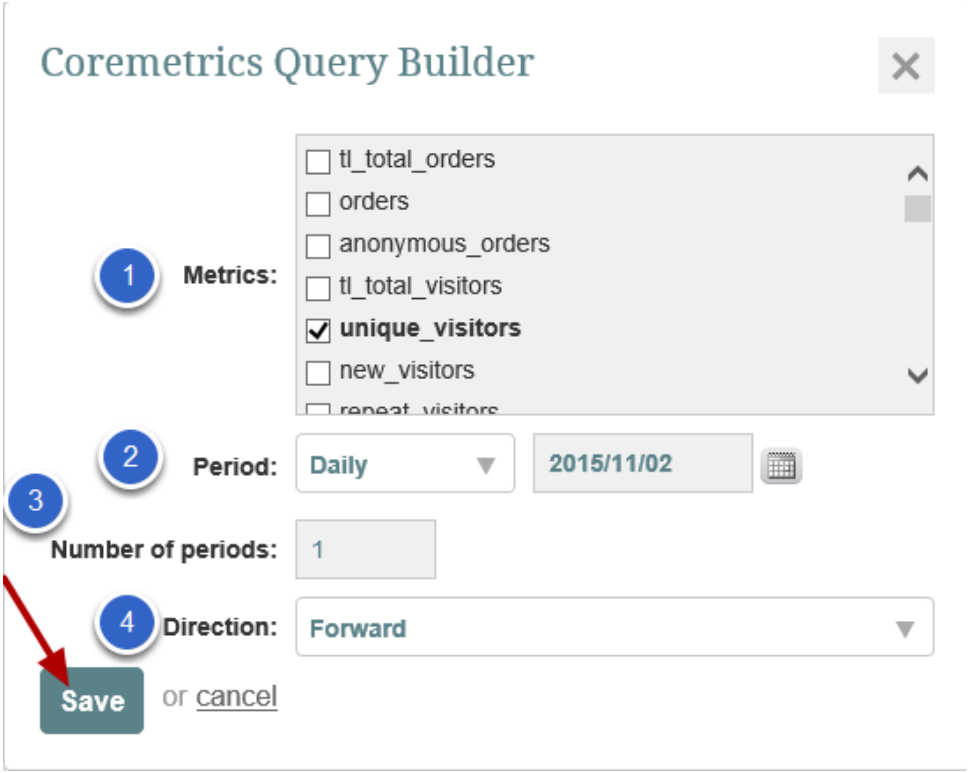
Next: (example is using full editor, but same steps apply when using Wizard)

3. Full Editor displays the Data Collection tab



1. Select **IBM Coremetrics** plug-in from **Data Source** drop-down
2. Set **Trigger**
3. Input Plug-in Command manually or using Visual Editor

3.1. Example using the Visual Editor



1. Select fields
2. Choose measurement **Period**
3. Input **Number of Periods** to measure
4. Select **Direction**

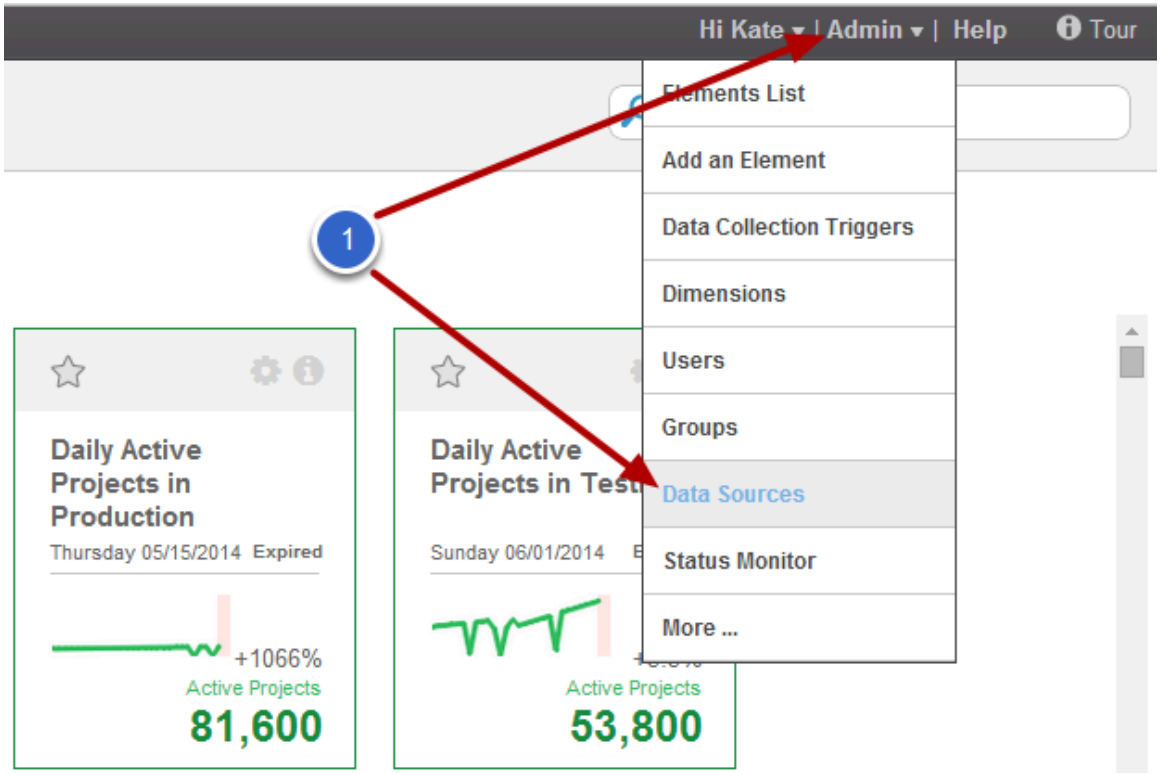
Save

21. Sourcing Data from Marketo

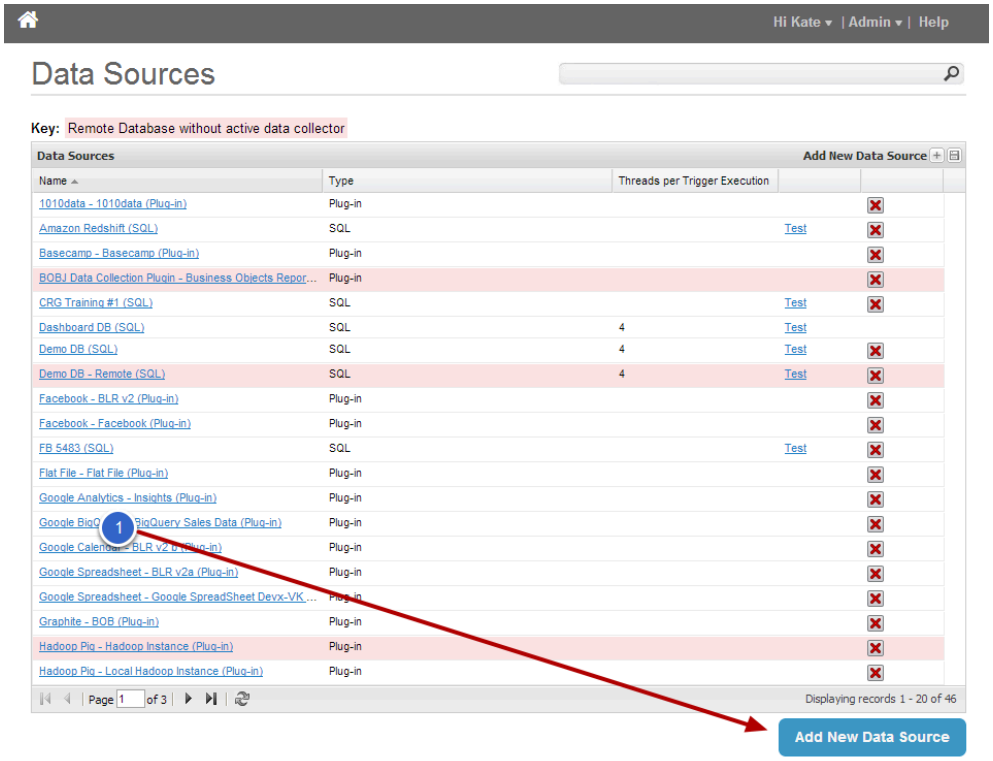
21.1 Establish Connectivity to Marketo

An Administrator can use the process described in this article to create a new Plug-in Data Source that is required to allow Elements to fetch data from Marketo to create a visualization in Metric Insights.

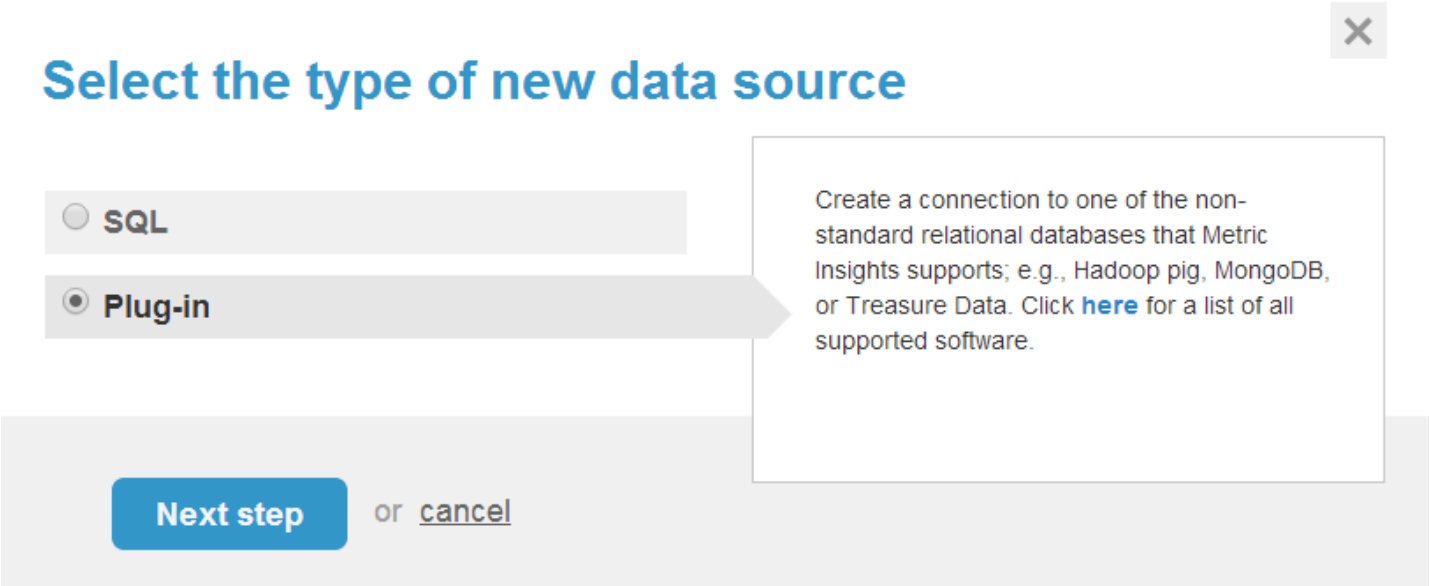
1. Go to Data Source list



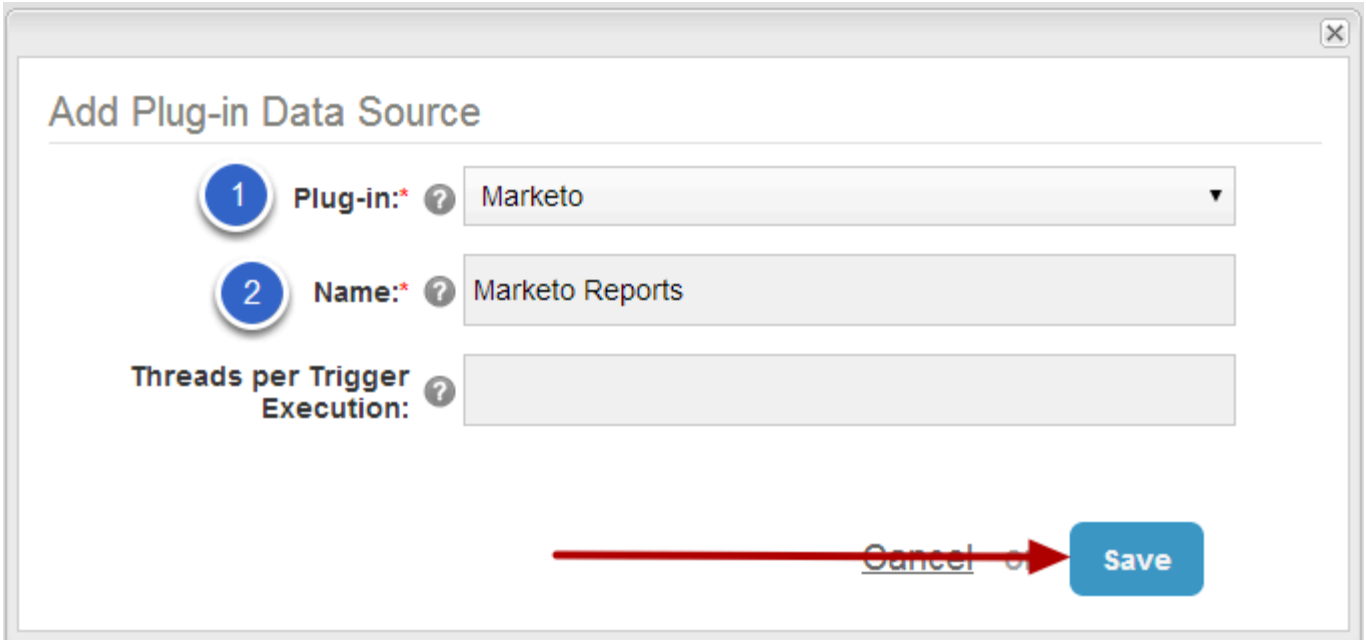
2. Add New Data Source



3. Select "Plug-in" as Data Source Type

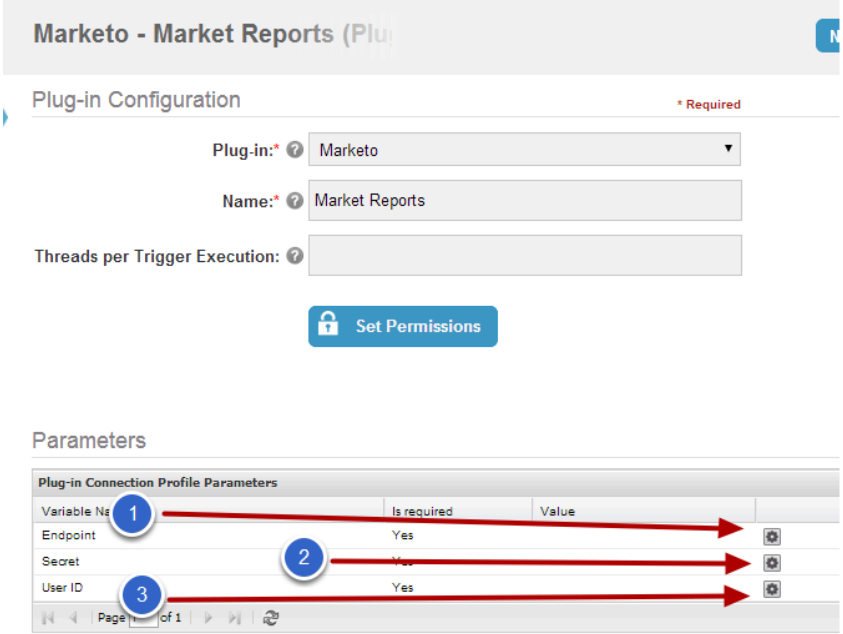


4. Select 'Marketo' from Plug-in picklist and input meaningful Name



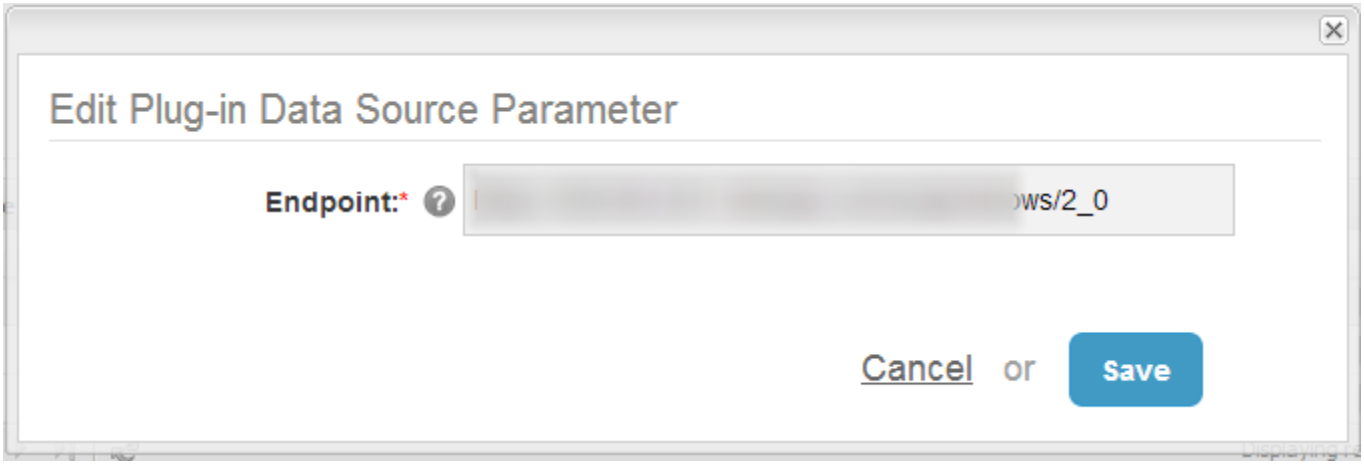
Save

5. Plug-in Data Source Editor will open

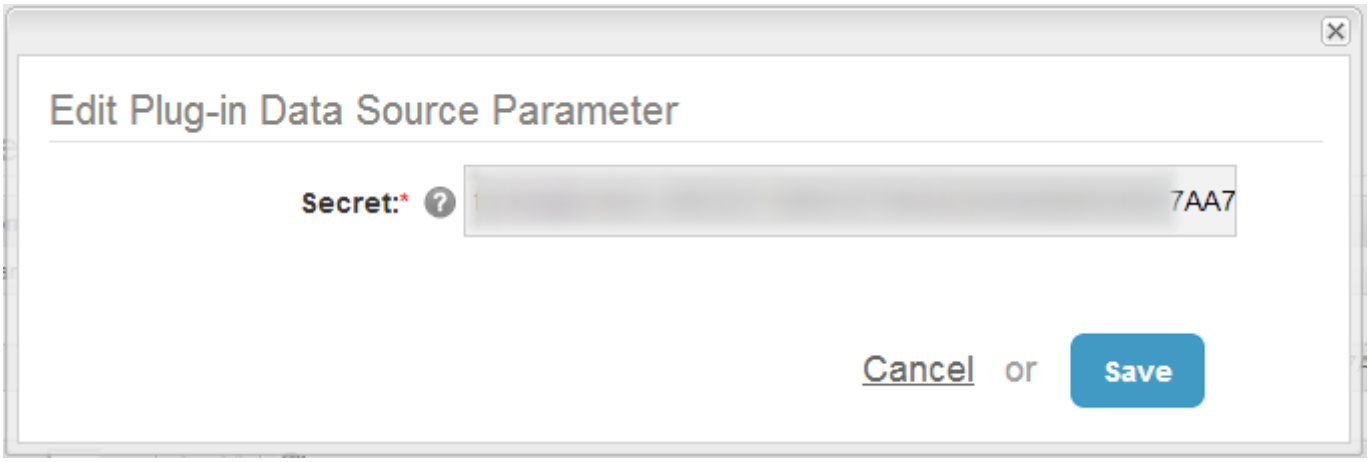


1. Click on gear icon to set **Endpoint** parameter
2. Click on gear icon to set **Secret** parameter
3. Click on gear icon to set **User ID** parameter.

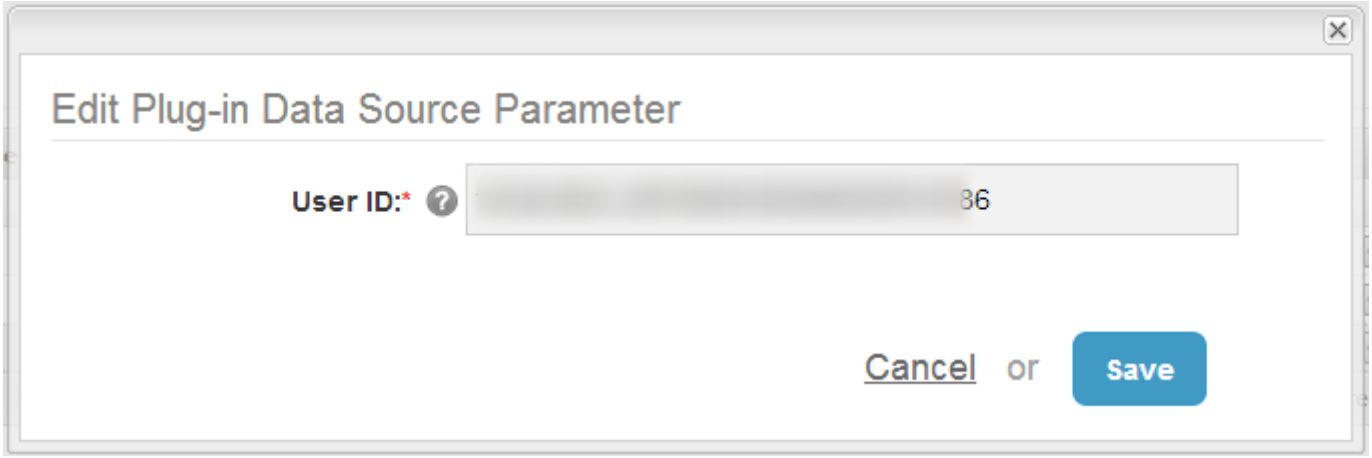
6. Add Endpoint parameter to profile



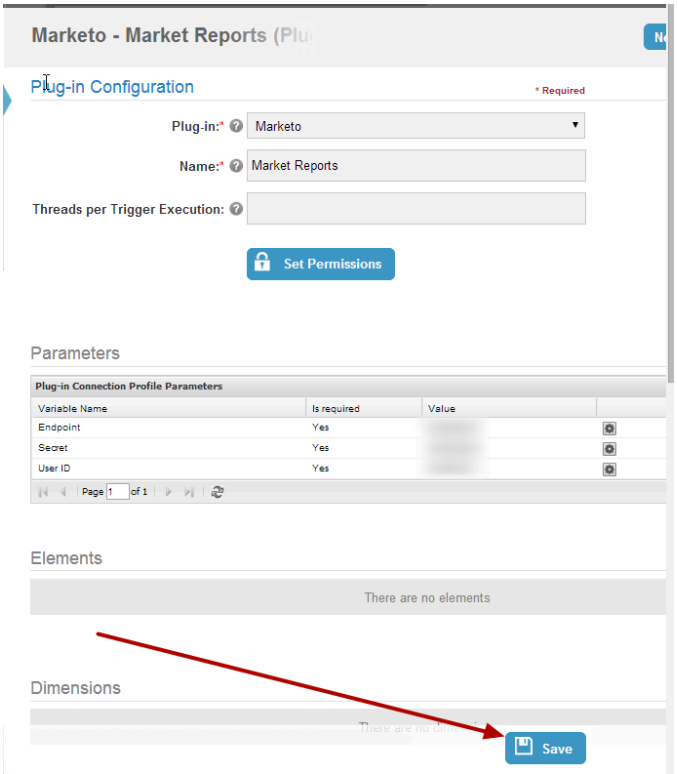
7. Add Secret parameter to profile



8. Add Secret parameter to profile



9. Save Marketo Plug-in



22. Sourcing Data from MicroStrategy

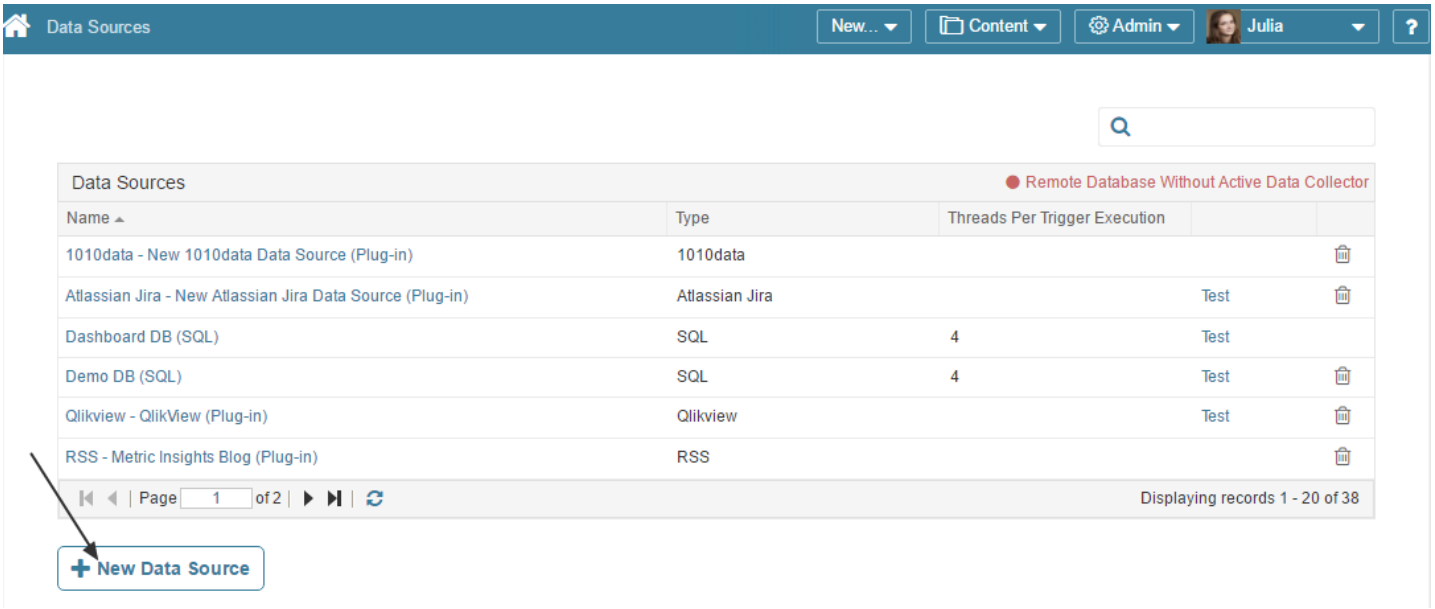
22.1 Establish Connectivity to MicroStrategy

This article describes the process of creating plugin Data Source to connect to MicroStrategy. This Data Source will allow data from existing MicroStrategy objects to be used in building elements using Metric Insights tools.

PREREQUISITES:

- **You must have network connectivity access from Metric Insights to your MicroStrategy Intelligence server (default port 34952).** If there is no access, then you can use a lighter version of this plugin connector by setting a plugin parameter to bypass ('Use only Task API'). See below.

1. Access Admin > Data Sources



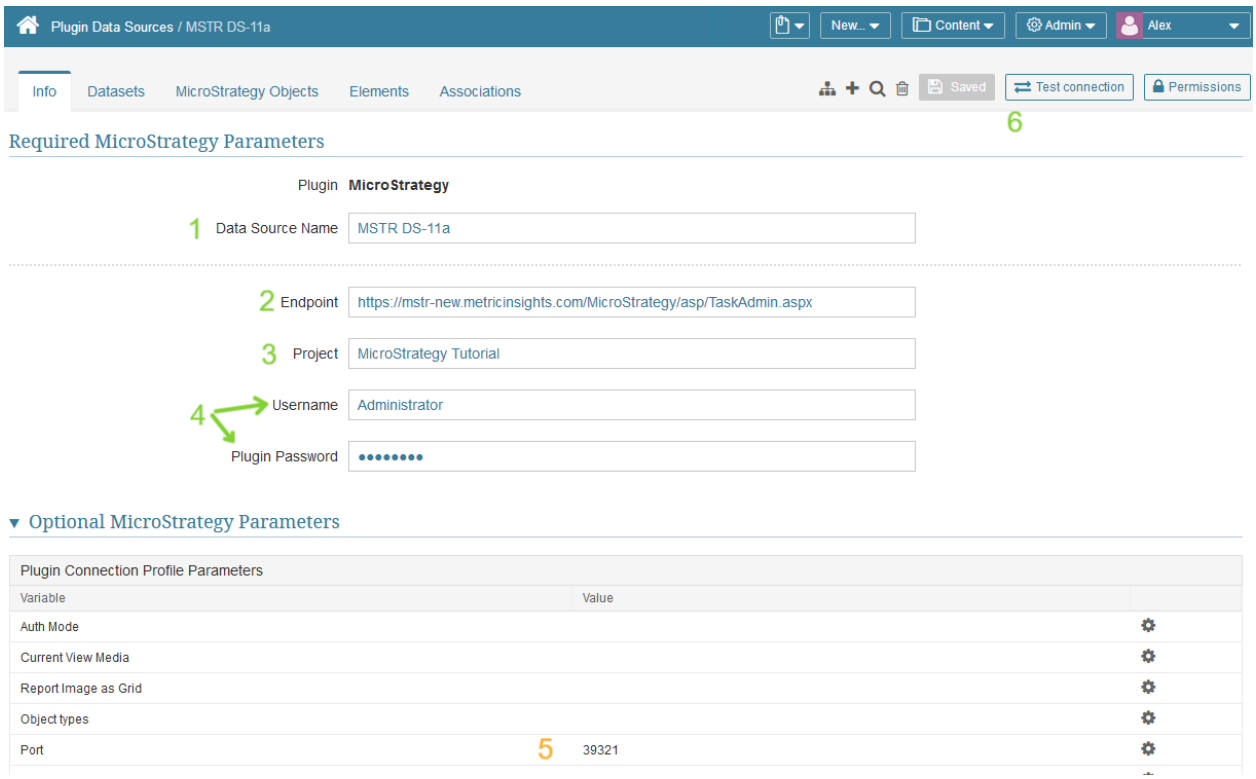
At the bottom of the screen click **[+ New Data Source].**

The *Select the Type of New Data Source* pop-up opens.

2. Select "Other" Data Source Type and choose "Microstrategy" from the drop-down list

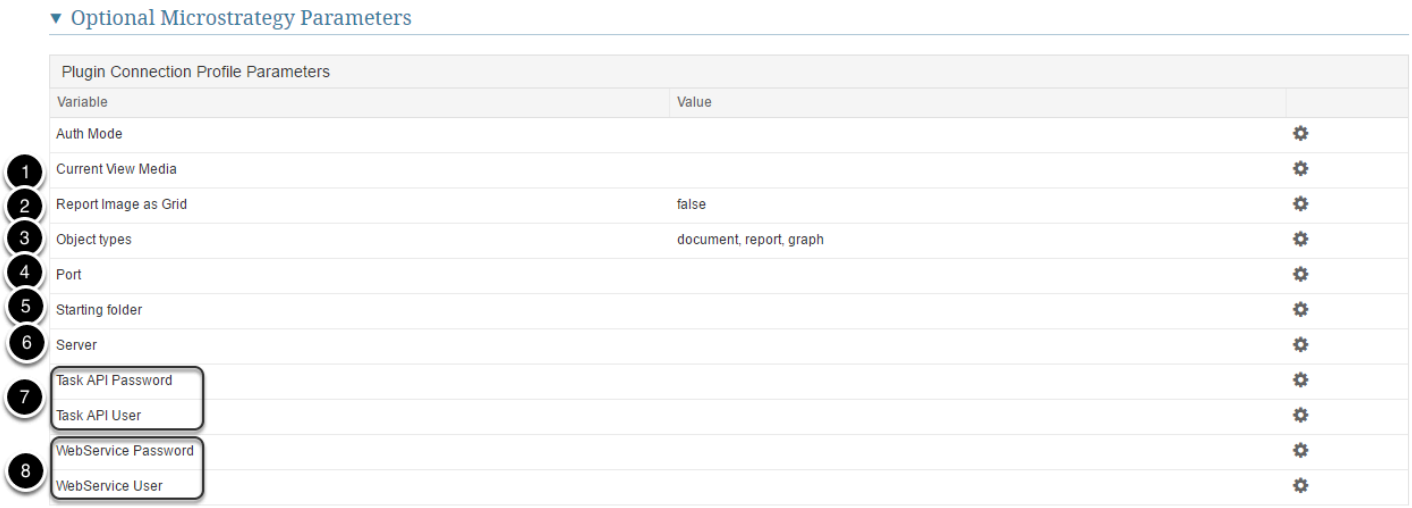
Move to the **Next step**.

3. Provide the Required Microstrategy Parameters



1. **Data Source Name:** defaults to a unique name, but may be modified to a descriptive name of your own.
2. **Endpoint:** The URL to MicroStrategy Task API. This is the API Metric Insights uses for accessing MicroStrategy. For .NET the url is typically **https://<mstr host>/MicroStrategy/asp/TaskAdmin.aspx**. For Java, this is **https://<mstr host>/servlet/taskAdmin**. As confirmation, you can access this URL via your web browser.
3. **Project:** the MicroStrategy project
4. **Username / Password:** This is the username and password you use to access the **Task API Endpoint** (Java Web API).
5. **Optionally,** to enable secure connection to WEB API using a certificate:
   - specify proper SSL port value of MicroStrategy Intelligence server in a **port field** (default configuration is **39321**).
   - add a configuration parameter to file **/opt/mi/datacollector/plugins/mstr.conf** : `pathToSSLCer=/path_to_your_cert`
6. **Test Connection** (this will also **Save** your data)

4. Define the values for Optional Parameters (if needed)

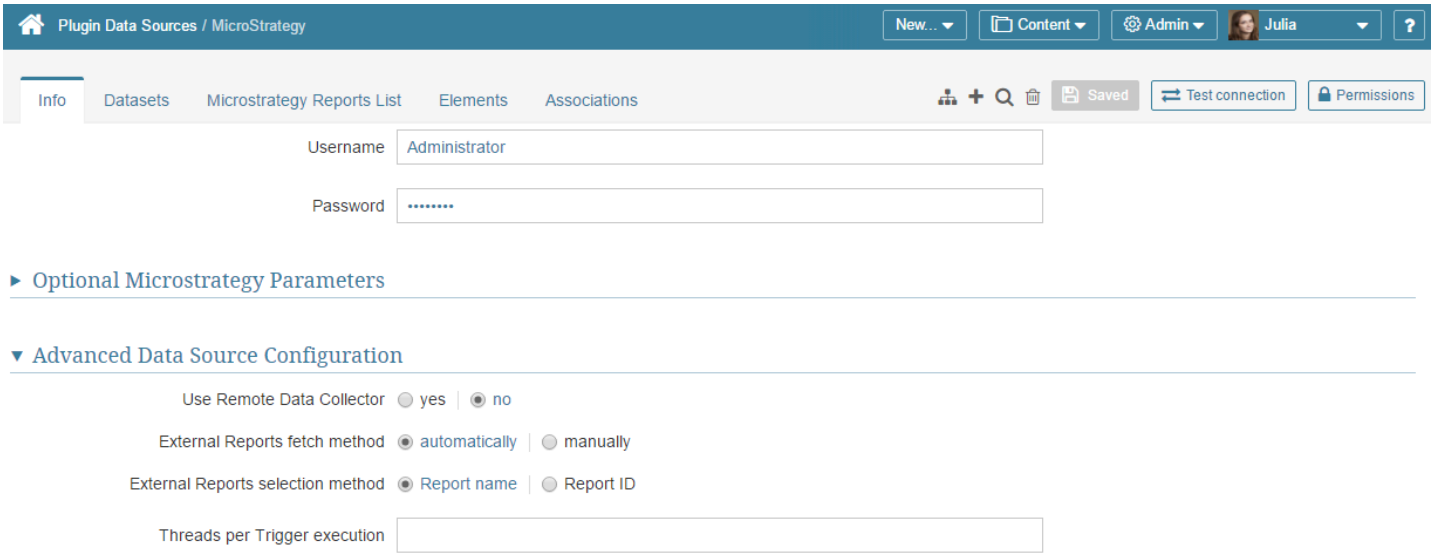


The plugin uses two API methods for pulling data and images from MicroStrategy. The MicroStrategy **Task API**, and the MicroStrategy **Java Web API**. The reason for this is each API only returns a subset of what is needed by Metric Insights.

Number on a screen	Field Name	Description
1	**Current View Media**	Values: 1 = Standard (default), 16 = LDAP Authentication, 4 = Database Authentication, 8 = Guest, 2 = Windows Authentication, 64 = Trusted Authentication, 128 = Integrated Authentication.

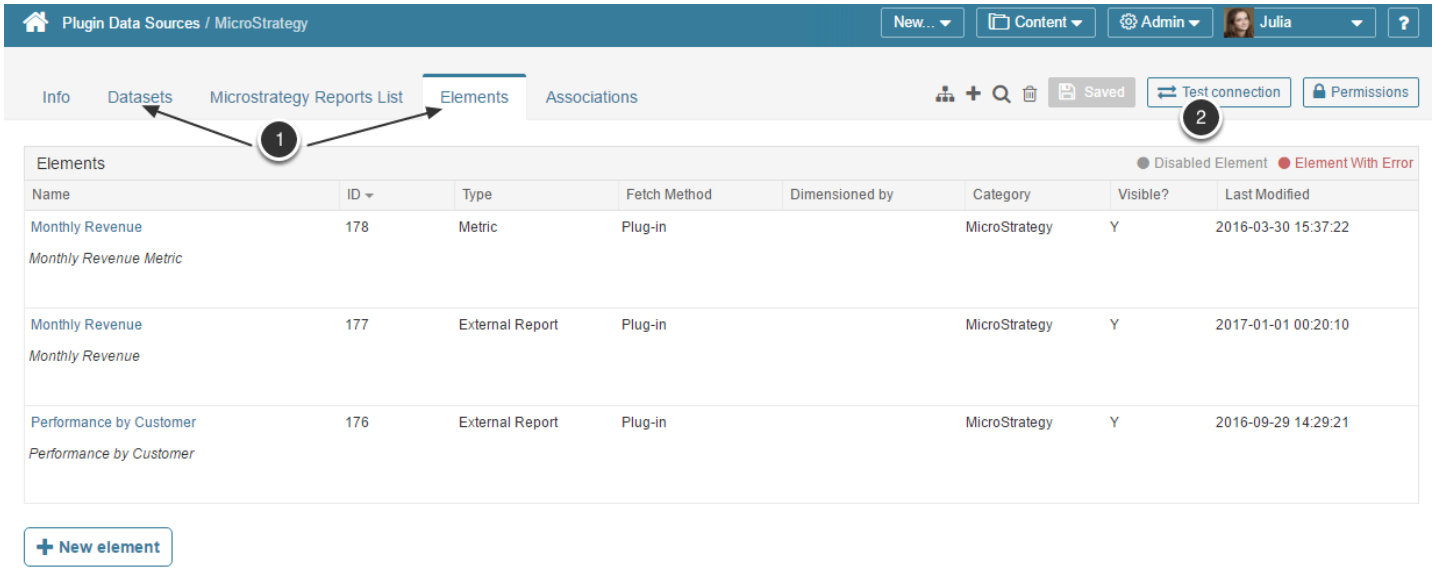
Number on a screen	Field Name	Description
2	**Report Image as Grid**	MicroStrategy renders Reports as Grid or Chart. Default is 'Grid'.
3	**Object types**	The list of object types to use
4	**Port**	Port used for accessing the MicroStrategy Intelligence Server via the MicroStrategy Java Web API. Default port is 34952. If using a different port then specify here. If not specified here, then the MicroStrategy API uses the default port value (34952).
5	**Starting folder**	The starting (root) folder for your Project. All MicroStrategy documents and reports under this folder will be available.
6	**Server**	For Task API if the server name is different than the host name in the Task API **Endpoint**.
7	**Task API Password / Task API Username**	Specify the Task API username/password here if the credentials for accessing the Task API differ from accessing the Java Web API (saved in **Username** and **Password** above).
8	**WebService Password / WebService User**	If there is a separate Web Service User to access the Task Admin login page, then supply the credentials here.

5. Advanced Configurations



1. **External Reports fetch method**: This setting influences options available in the *Microstrategy Reports List* tab:
   - **automatically**: just click **Refresh list** and all Reports are going to be fetched by the system
   - **manually**: Reports may be added one-by-one or via CSV file
2. Optionally, specify the maximum number of concurrent **Threads per Trigger execution** to be used in background processing when the system updates Metrics and Reports for this Data Source. If you do not specify any value for this setting, batch data collection processing will be single-threaded.

6. Other Settings



1. You can create elements or Datasets directly from the respective tabs
2. Click **Permissions** to assign them to Groups or Power Users

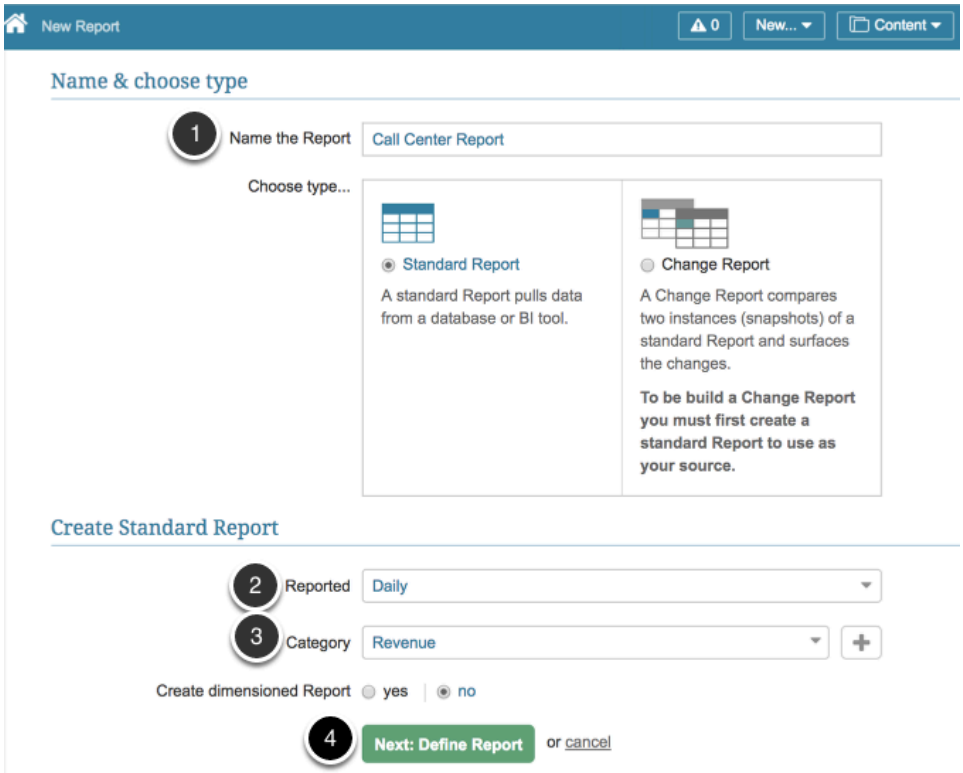
What's Next?

[How to collect data using Microstrategy plugin]

22.2 How to collect data using Microstrategy plugin

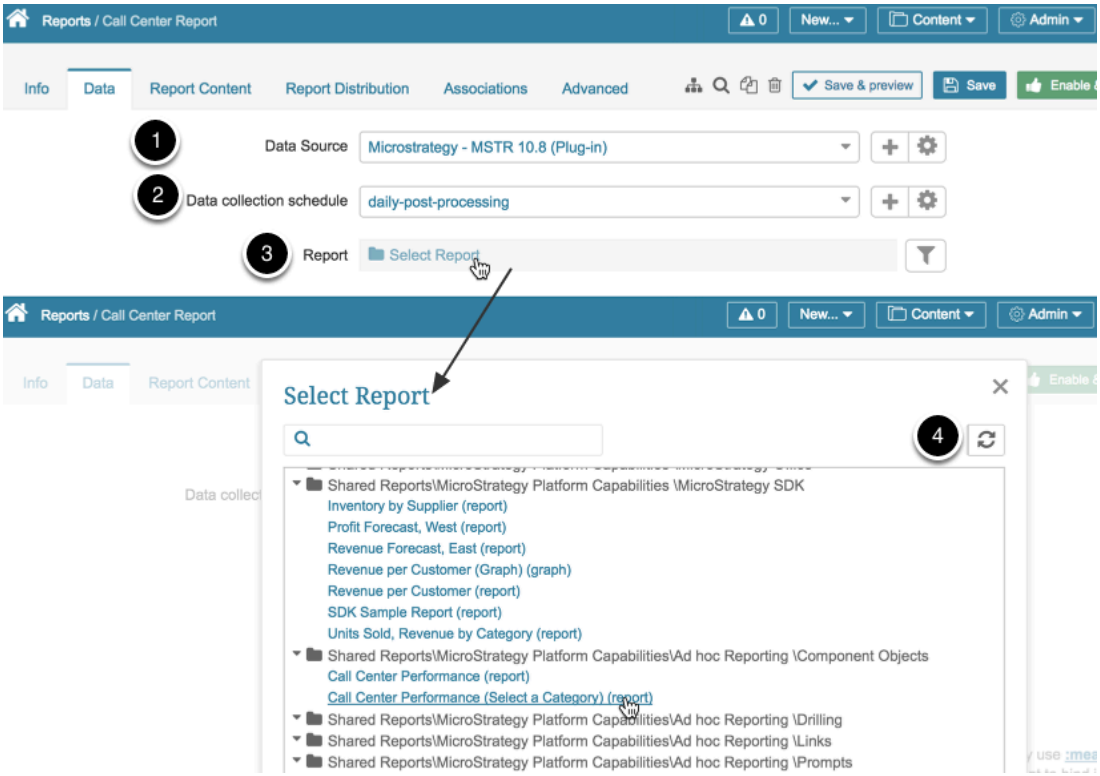
This article demonstrates how to create a Report using MicroStrategy as a data source. It assumes that you have already established connectivity to Microstrategy via the plugin.

1. Access New > Report



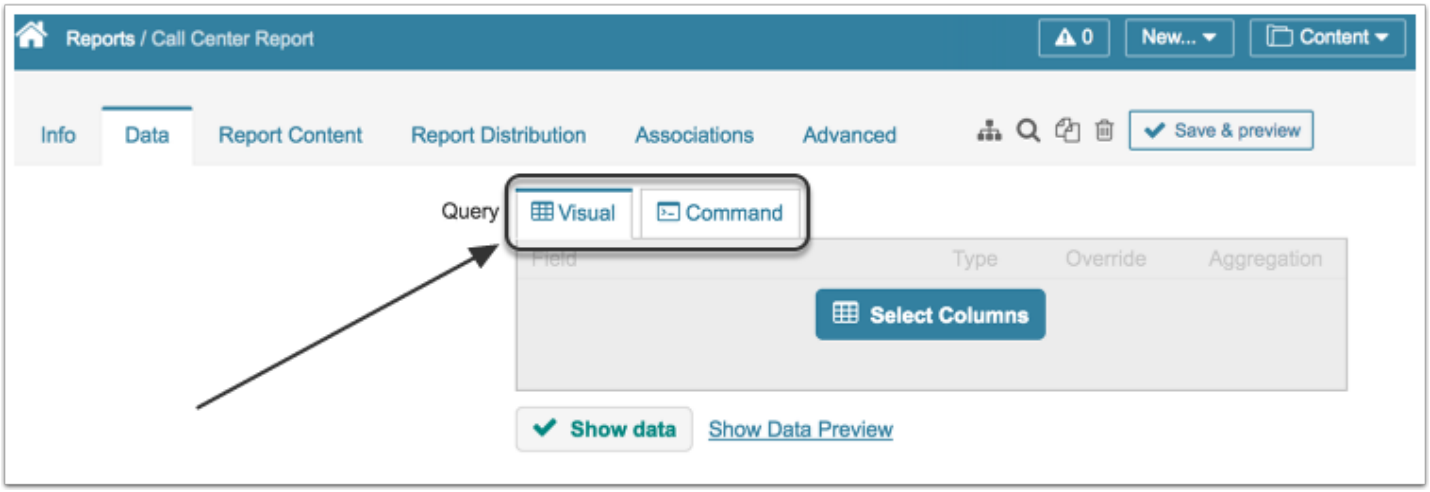
1. **Name the Report**: Define a unique descriptive name of your element
2. **Reported**: choose the measurement interval from the drop-down list
3. **Category**: define a category this element belongs to
4. To move on to defining data collection details, click **Next: Define Report**

2. Define settings for Data Collection

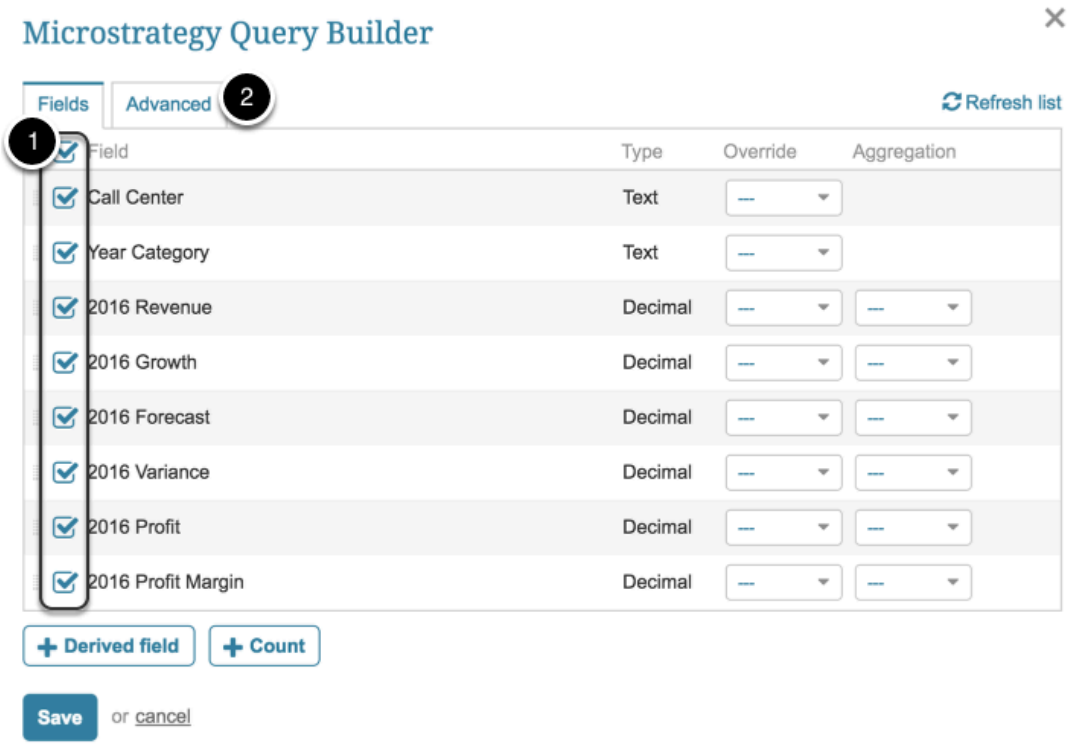


1. **Data Source:** select the connection profile you have created for **Treasure Data**
2. **Data Collection Schedule:** Specify the trigger that will be used to collect the data for your report
3. **Report:** Click **Select Report** to open the pop-up with the list of objects available from the Microstrategy site.
4. If you do not see the required item, try refreshing the list by clicking the **Refresh** icon at the upper right corner of the pop-up.

3. Create a Query (via the Visual Editor or a Command)



3.1. Example using the Visual Editor

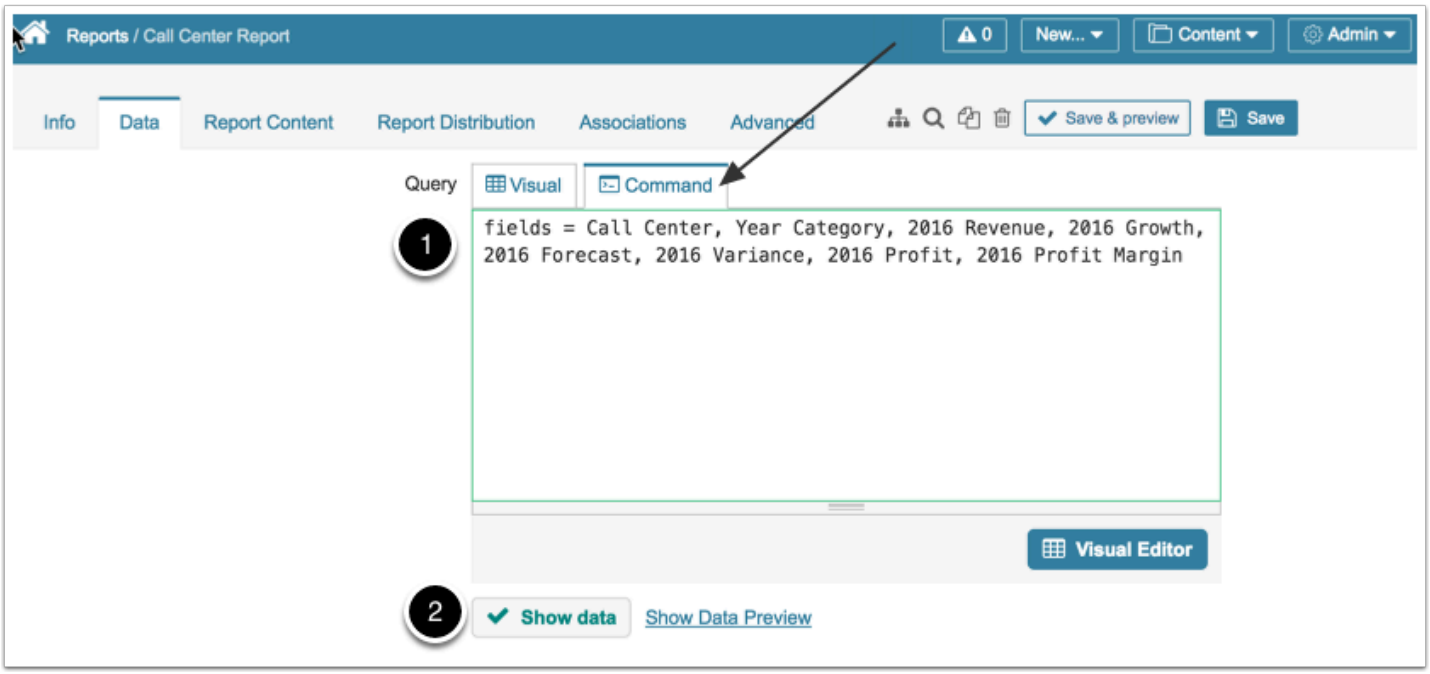


To open the *Microstrategy Query Builder,* click **Select / Modify Columns.**

1. Select **Fields** and set **Expressions**
2. You can pre-filter the information before fetching it into Metric Insights or add 'ascending' and 'descending' **Sorting** to the field values, open the *Advanced* tab.

Save your settings. If the Query is validated successfully, Report data is shown in the drawer below.

3.2. Example using a Command

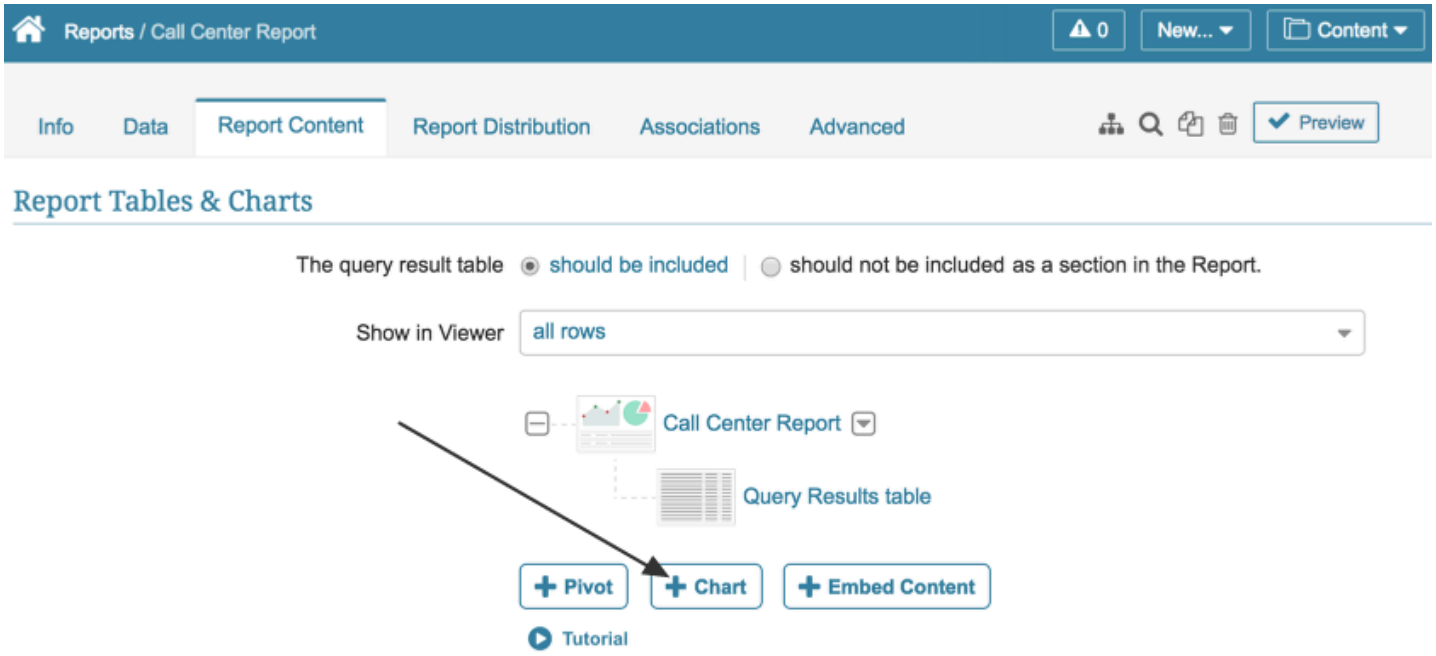


1. **Query:** Construct a command listing all the columns with data that should be included into the Report (manually or via the visual editor)
2. Once you are ready with the query, click **Show Data**.

If your plugin command is valid, the command box is **green** and the Preview of the Report data is shown in the drawer below; if there are any errors, the box is colored in **red** and errors are explained below the statement box.

4. Enable and Publish

5. [Otional] Creating a Chart from the Report's result set

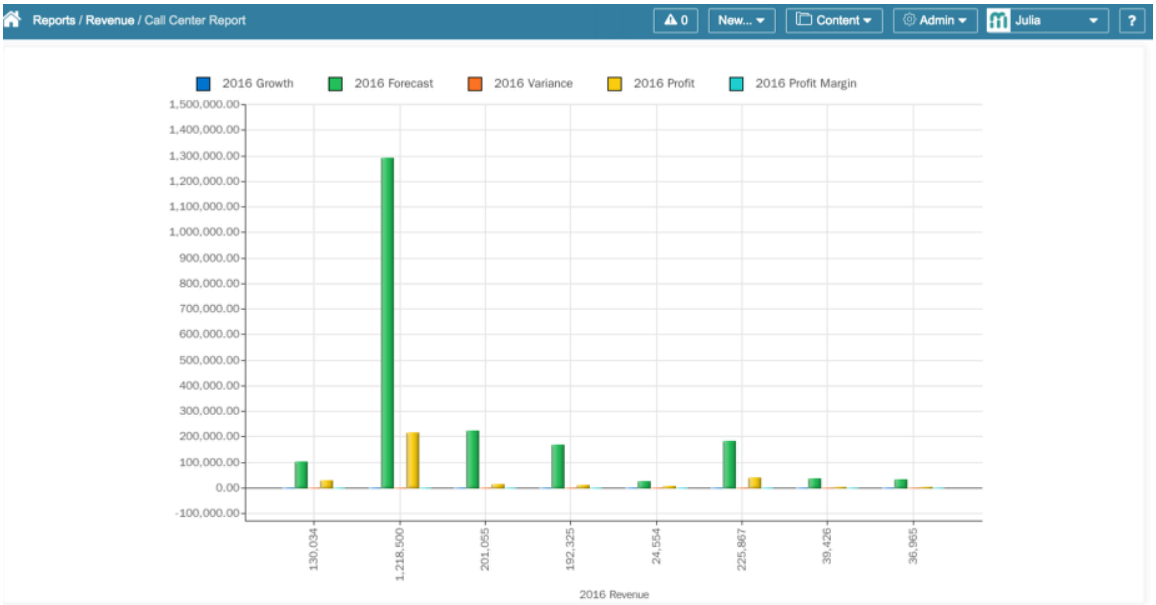


You may add a Chart based on the Report result set.

To do that, go to *Report Editor* > *Report Content tab* > *[+ Chart]*.

For more details, refer to: [Develop Report Chart based on the Result Set (without a Pivot)]

6. Result



Call Center Report

Call Center	Year Category	2016 Revenue	2016 Growth	2016 Forecast	2016 Variance	2016 Profit	2016 Profit Margin
Milwaukee	Books	130,034	0.27	101,427	0.28	28,107	0.22
Milwaukee	Electronics	1,218,500	0.28	1,291,610	-0.06	214,706	0.18
Milwaukee	Movies	201,055	0.21	223,171	-0.10	12,700	0.06
Milwaukee	Music	192,325	0.23	169,246	0.14	9,004	0.05

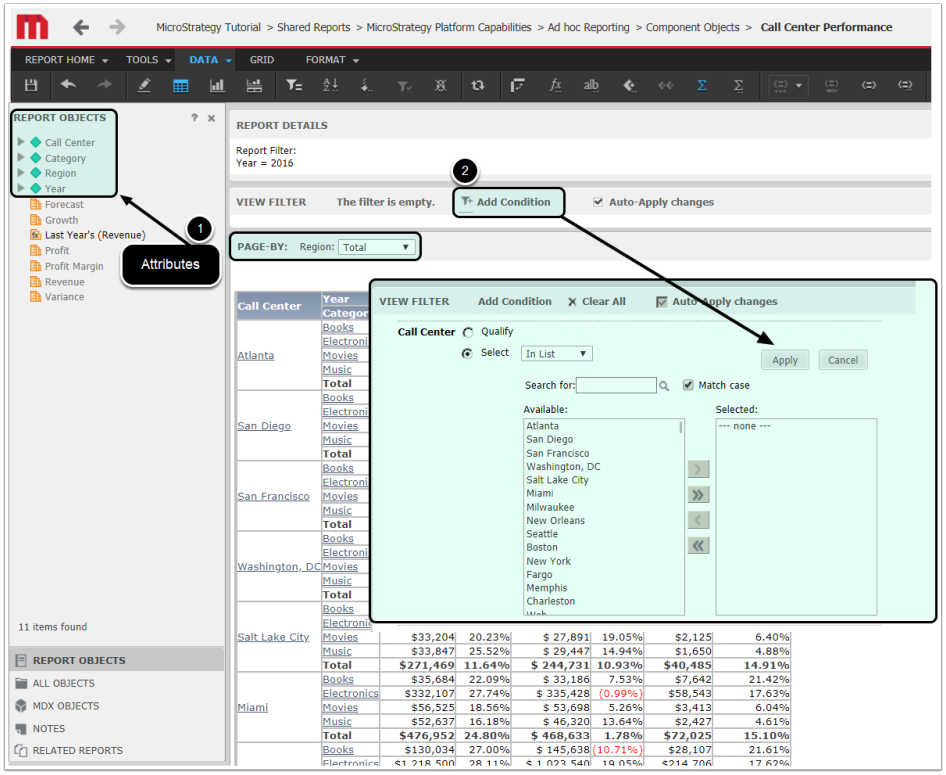
22.3 Pre-filtering MicroStrategy data (Dataset example)

When sourcing data from MicroStrategy for Metrics, Reports, External Reports, Dimensions and Datasets, you can pre-filter your data before fetching it. This function allows to focus on the slice of data that you really need and exclude those values that are currently irrelevant for you and your research.

PREREQUISITES:

Establish Connectivity to MicroStrategy

How to find Filter names in MicroStrategy?



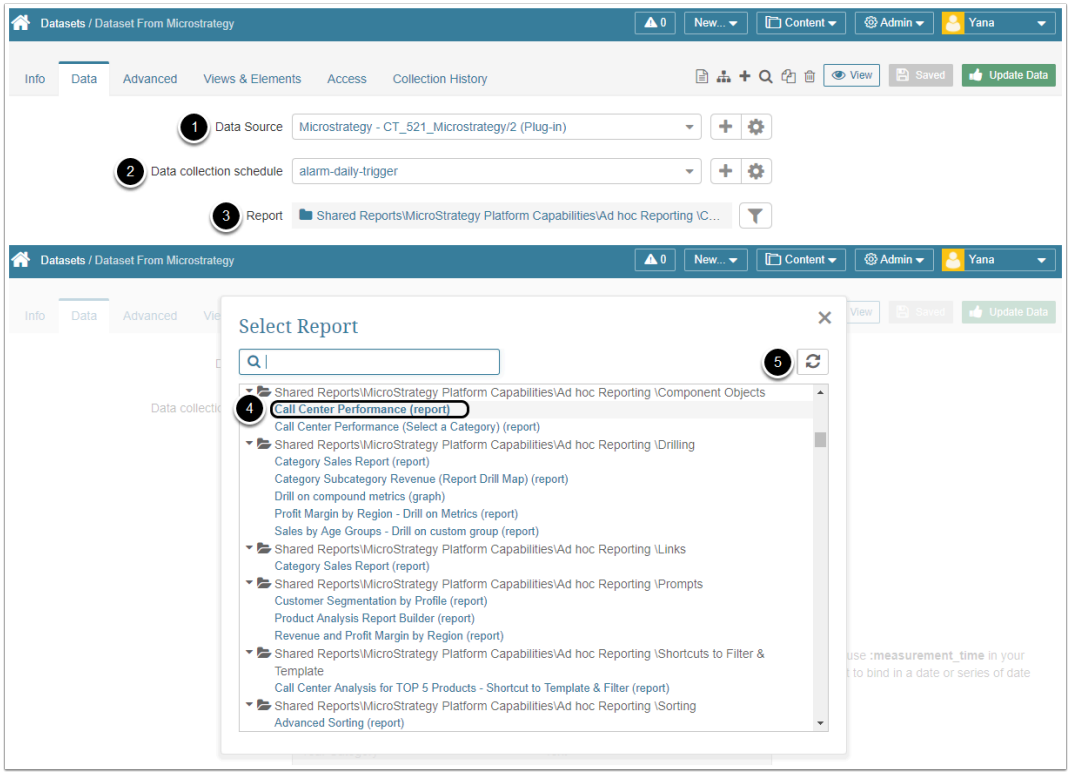
Data displayed in MicroStrategy Viewers can be filtered by different criteria. Filters in MicroStrategy are built from **Attributes**. On the example above, they are displayed as **Report Objects** on the left of the Report Viewer.

To find **Filter names** in MicroStrategy you can:

1. Check available **Attributes**
2. Use **View Filter grid > [+Add Condition]** to open a list of available Filter values.

To view all Filters and Filter values at once, set **PAGE-BY option** to display **Total** (ALL values).

1. Define a Source Object for Data Collection



Start by creating an element. Once you get to the process of Data Collection, define the following:

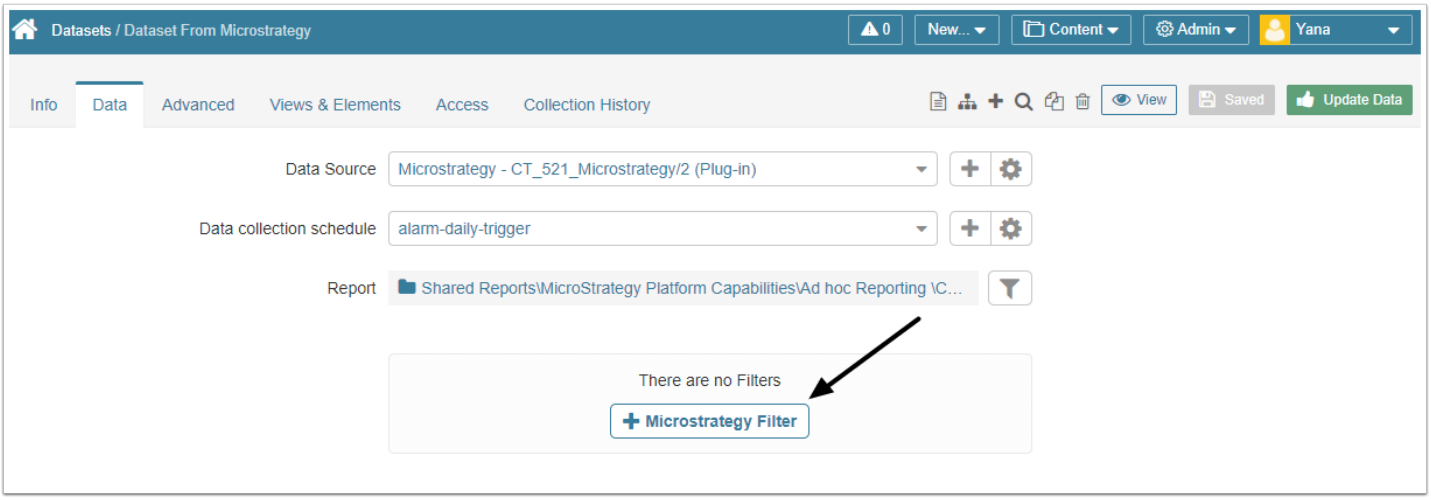
1. **Data Source:** This is an entity that connects MicroStrategy and Metric Insights. For more information, see: Establish Connectivity to MicroStrategy
2. **Data Collection Trigger:** select the Trigger that is going to initiate updating information in this Metric.
3. **Report:** Click **Select Report** to open the pop-up with the list of available MicroStrategy objects that can be a source of data.
4. Each item in the list is represented as the path (hierarchy) to the respective Report in MicroStrategy. Find the desired object in the list.
5. If you do not see the required item, try refreshing the list by clicking the **Refresh** icon at the upper right corner of the pop-up.

2. Adding MicroStrategy Filters to Metric Insights

Once filters are added to a Metric / Report or External Report for the first time, they are going to be automatically added to all new respective elements with the same Data Source / Report.

NOTE:

- External filters are tied to MicroStrategy Reports, not Metric Insights' elements. This allows Filters to be reused for multiple elements (there is no need to create new Filters every time an element is created in Metric Insights).
- If there are more External Filters or Filter Values that you would like to use for the current element, you can always set the redundant ones to "ignore".

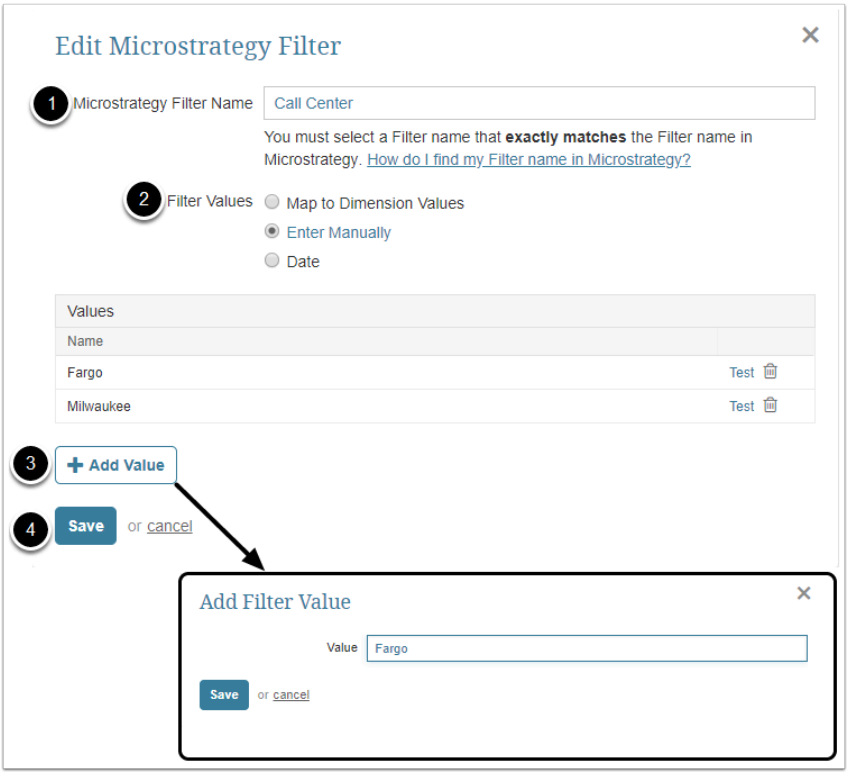


When creating a Metric / Report / External Report fetched from MicroStrategy, after you define the **Report** that should serve as a Data Source, you may pre-filter information that is going to be fetched.

To pre-filter the data, click **[+Microstrategy Filter].** The following pop-up will give you 3 options to add Filters.

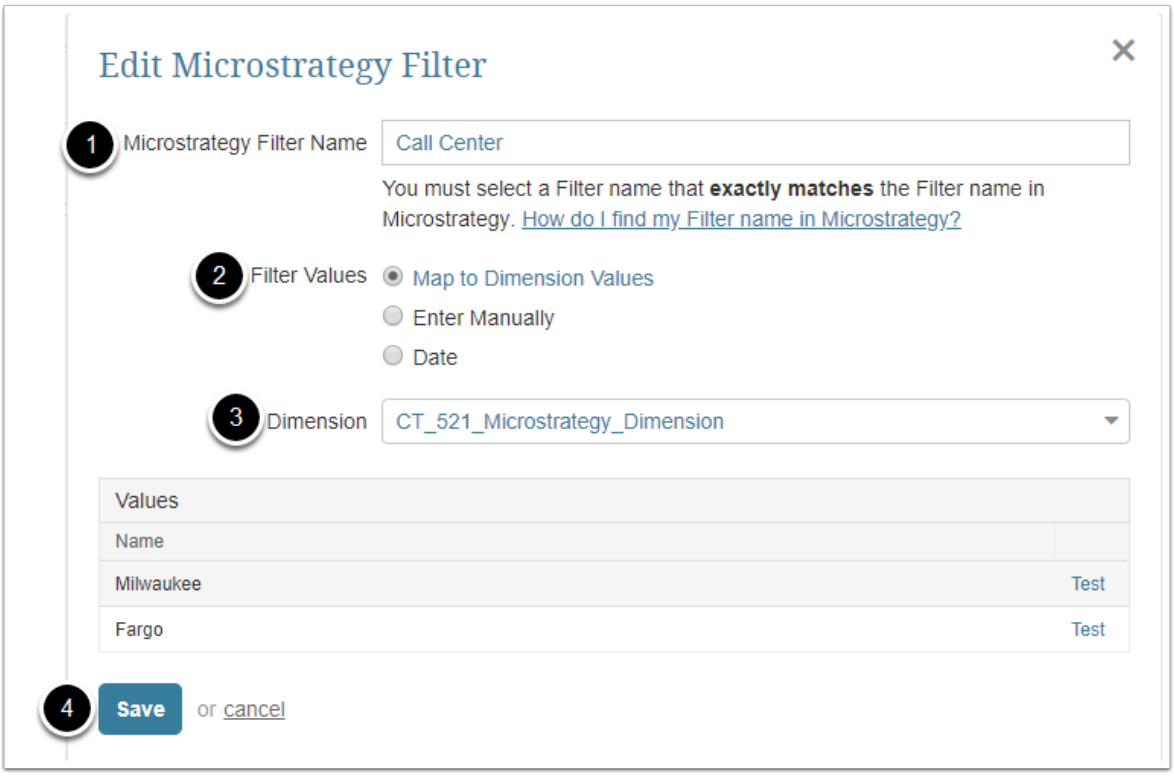
NOTE: Examples given below are taken from the MicroStrategy Report shown at the top of the page.

2.1 Enter manually



1. **MicroStrategy Filter Name**: Define the name of the filter from MicroStrategy (The name of the Filter must exactly match the column names of the MicroStrategy Report. Filter names are case sensitive. Unless the match is exact, the Filter will not work).
2. **Filter Values:** Choose 'Enter Manually' and click **Save** at the bottom of the pop-up.
3. Click **[+ Add Value]** and in the opened pop-up manually type in the name of the filter value. **Save** your entry. All added values appear in the *Values* list.
4. **Save** your entries.

2.2 Using dimension values



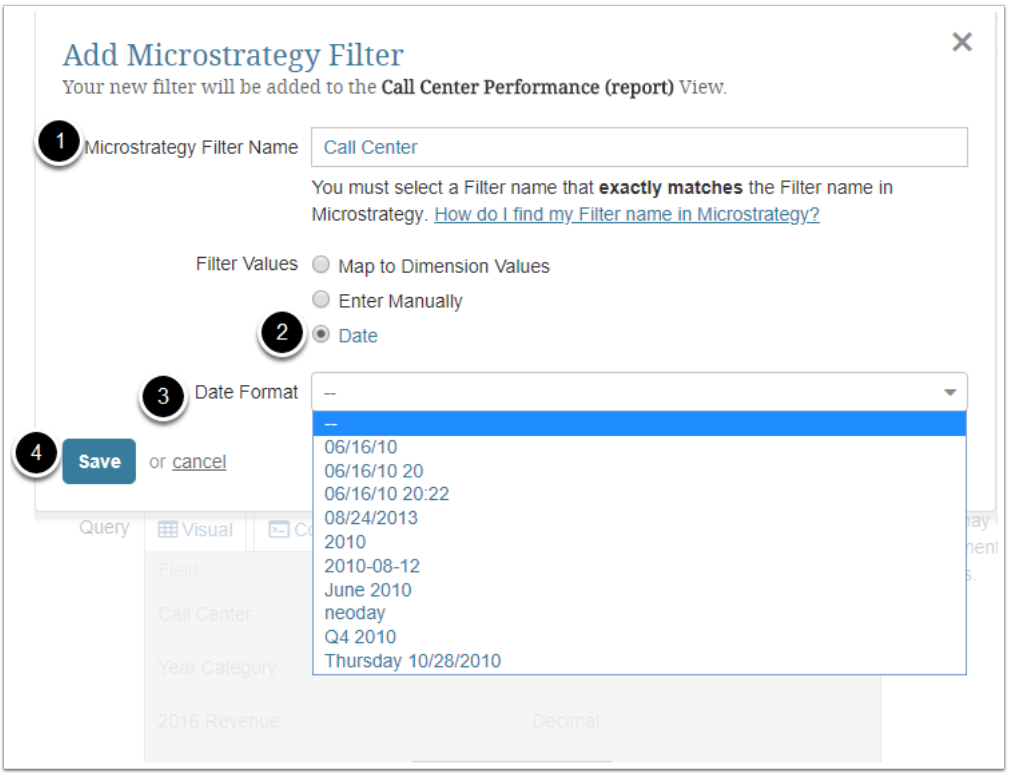
If you have already used MicroStrategy filters to create Dimensions in Metric Insights, you can quickly choose which Dimension Values you want to use for pre-filtering:

1. **MicroStrategy Filter Name**: Define the name of the filter from MicroStrategy.
2. **Filter Values:** choose 'Map to Dimension Values'.
3. **Dimension:** select a corresponding Dimension from the drop-down list and all its Values are going to be loaded to the Values list automatically.
4. **Save** your entry.

NOTE: Dimensions used here must have Values that exactly match the Filter Values in MicroStrategy (if the Values do not match, the Filter will not work)

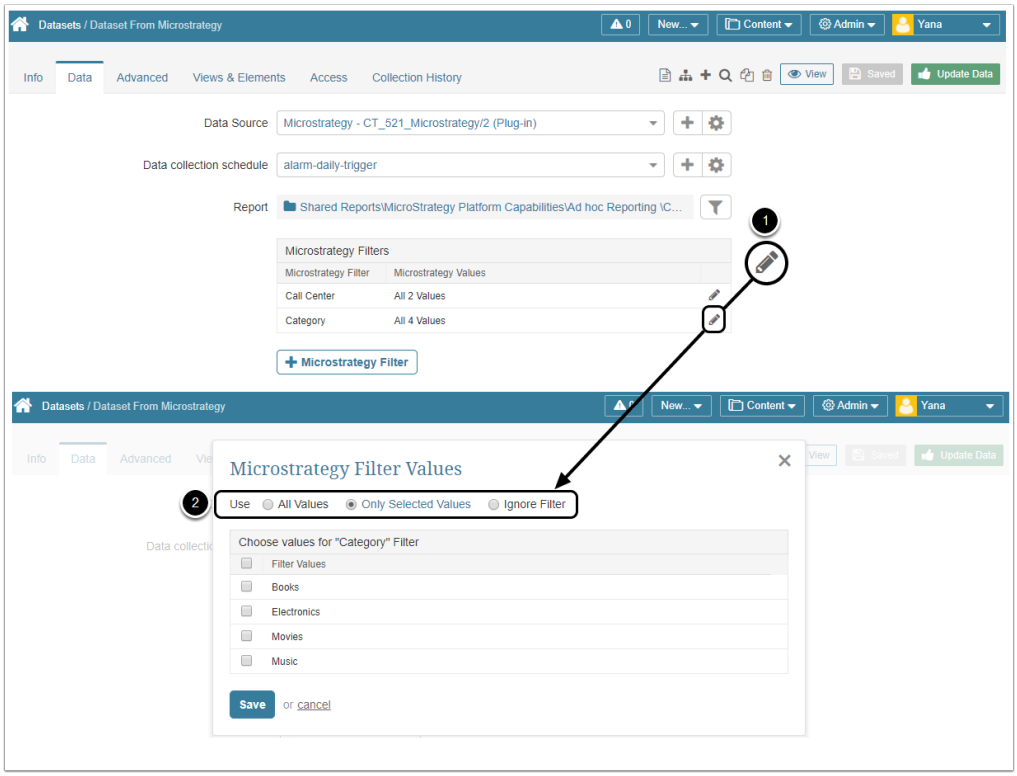
To source Dimension Values from MicroStrategy, see Create a Dimension with values automatically fetched from MicroStrategy

2.3 Using Date



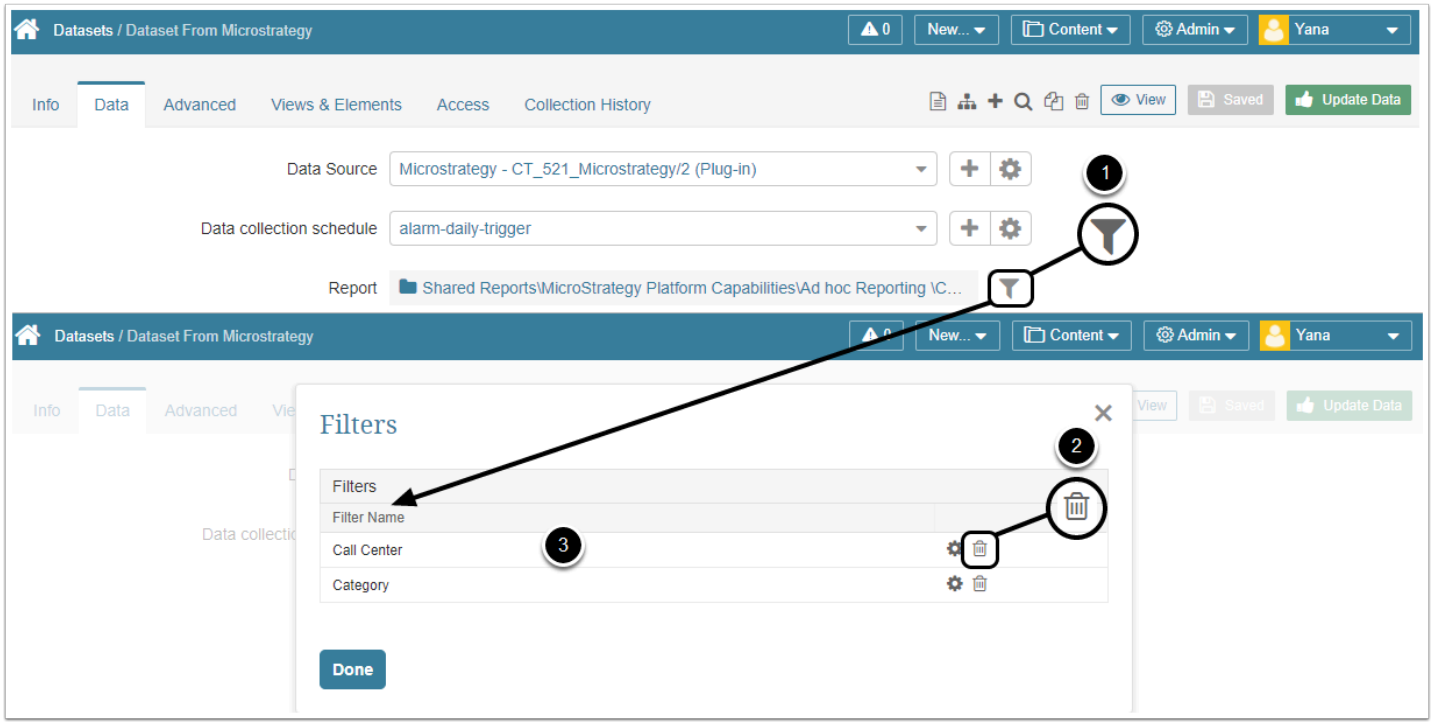
1. **MicroStrategy Filter Name**: Define the name of the filter from MicroStrategy (The name of the Filter must exactly match the column names of the MicroStrategy Report. Filter names are case sensitive. Unless the match is exact, the Filter will not work).
2. **Filter Values:** Choose 'Date'
3. Select the **Date Format** used in your MicroStrategy Report
4. **Save** your entries

3. How do I add filters to a results set from MicroStrategy?



1. Click the **Pencil** icon in the filter row to add it.
2. When the filter is added, you can use it for "All Values", "Only Selected Values" or ignore it.

4. Deleting Filters



To delete some of the added filters: (1) click the **Filter** icon in the **Report** field and (2) choose the unnecessary filters. Click the **Trashcan** icon in the corresponding row.

22.4 Create a Dimension with values automatically fetched from MicroStrategy

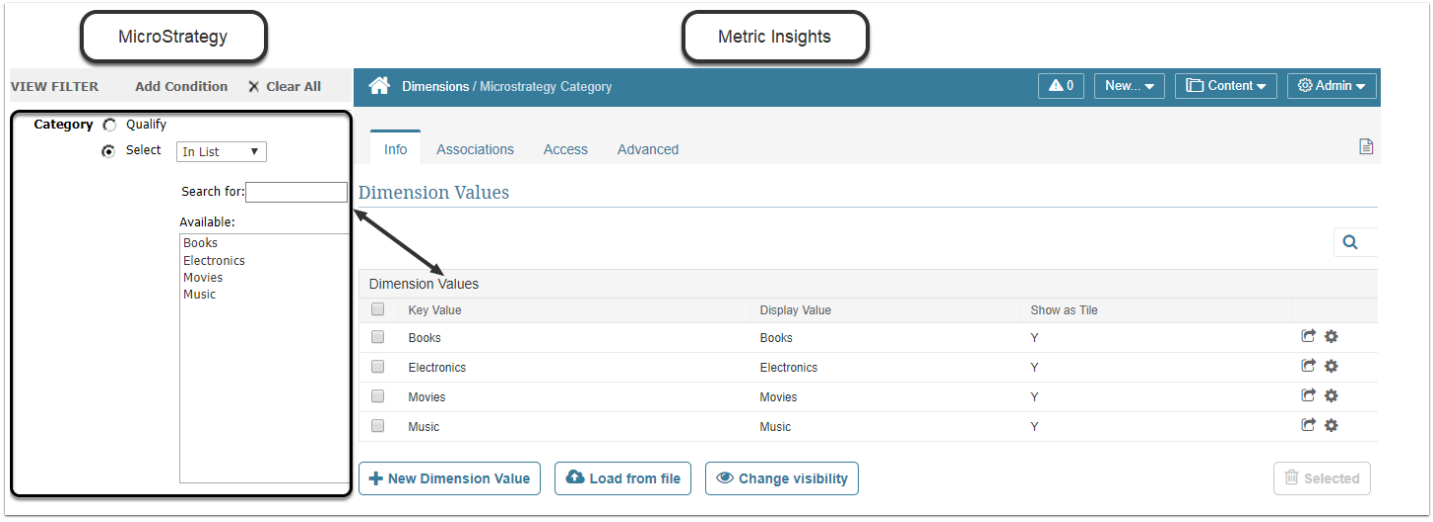
If your MicroStrategy report contains information for several Dimensions, you do not need to create a separate element for each Dimension Value in Metric Insights. All you need to do is just copy MicroStrategy "filter values" into MI. This article covers step-by-step instructions on fetching this data from MicroStrategy.

NOTE: If a new filter value is added to this MicroStrategy report later, it will be automatically copied to MI upon triggered data collection.

PREREQUISITES:

[Establish Connectivity to MicroStrategy]

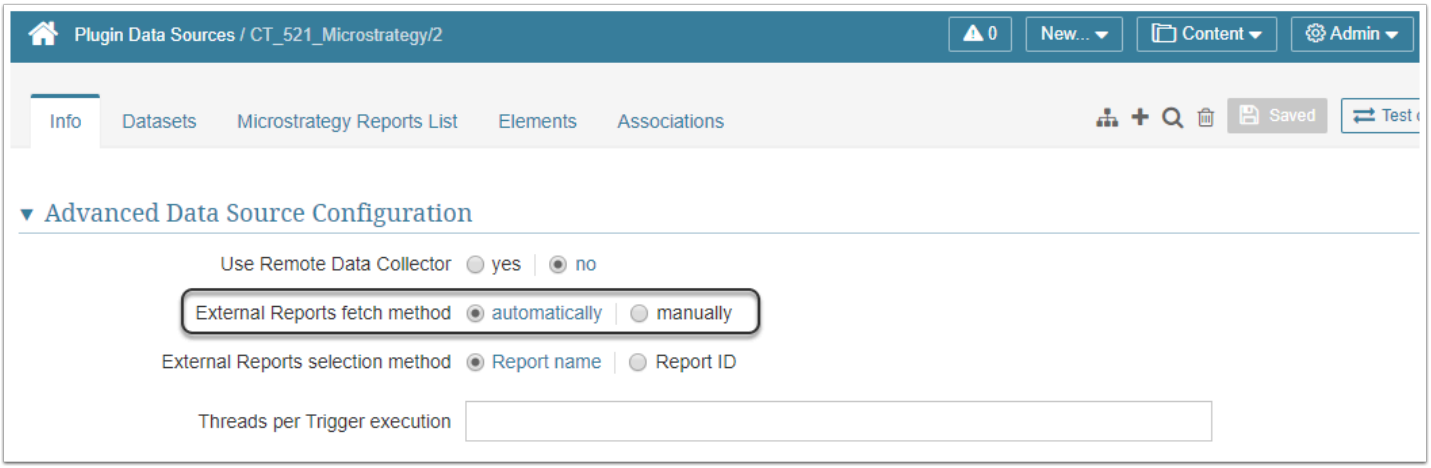
Use Case



1. Open the MicroStrategy Plugin to be used as a Data Source for the new Dimension

1. Go to *Admin* > *Data Sources.* The list with all data sources created in the system opens.
2. Select the plugin you plan to use as a Data Source for this Dimension. Alternatively, crate a new one: Establish Connectivity to MicroStrategy
3. Reports can be added or refreshed in the Plugin Editor.

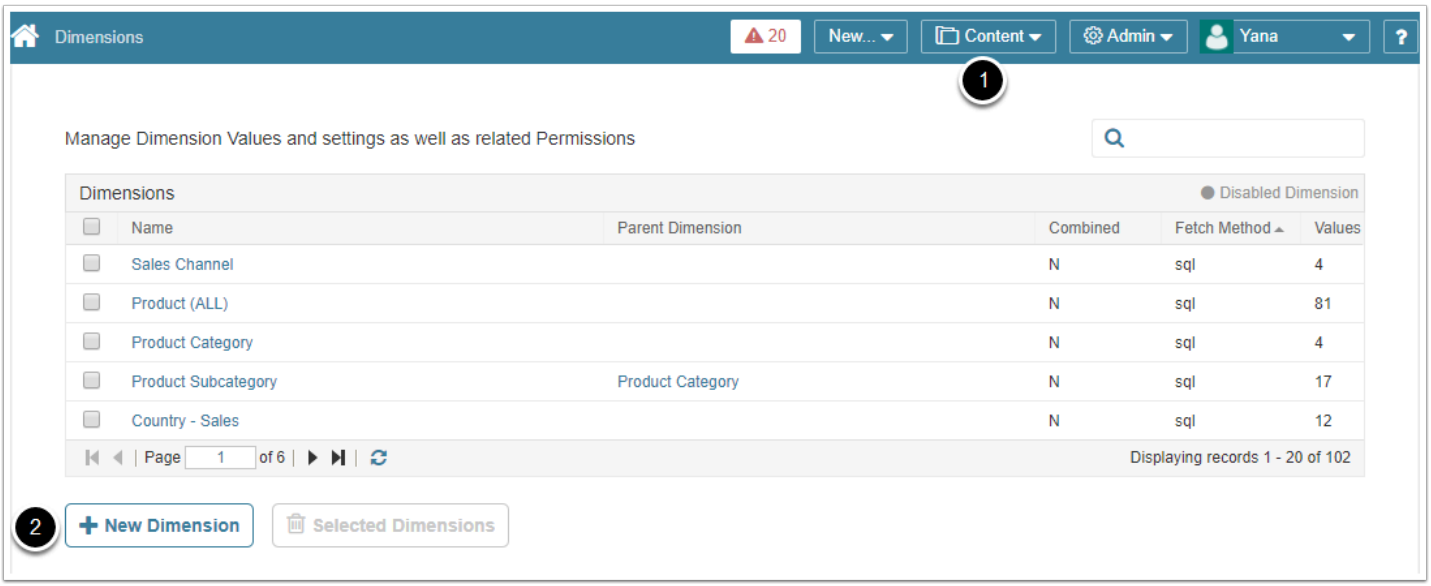
1.1. Update the list of MicroStrategy Reports to ensure you have most current report data



There are two options to add reports to the plugin:

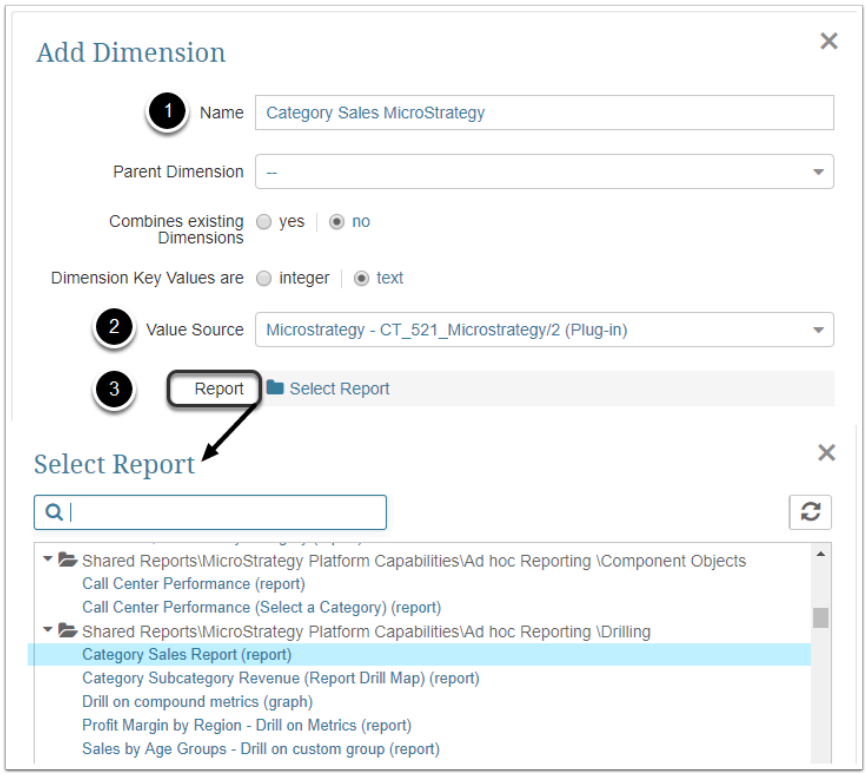
- **Automatically**: If the External Reports fetch method field is set to 'automatically', go to the MicroStrategy Reports List tab and simply click **Refresh** to collect all MicroStrategy reports currently available at the server
- **Manually**: You can also update the MicroStrategy Reports List by adding report IDs and Names one-by-one or via a CSV file

2. Add a new Dimension



1. Go to *Content* > *Dimensions*. The list page containing with all existing Dimensions in the system opens.
2. Click **[+ New Dimension]**

2.1. Define the Basics

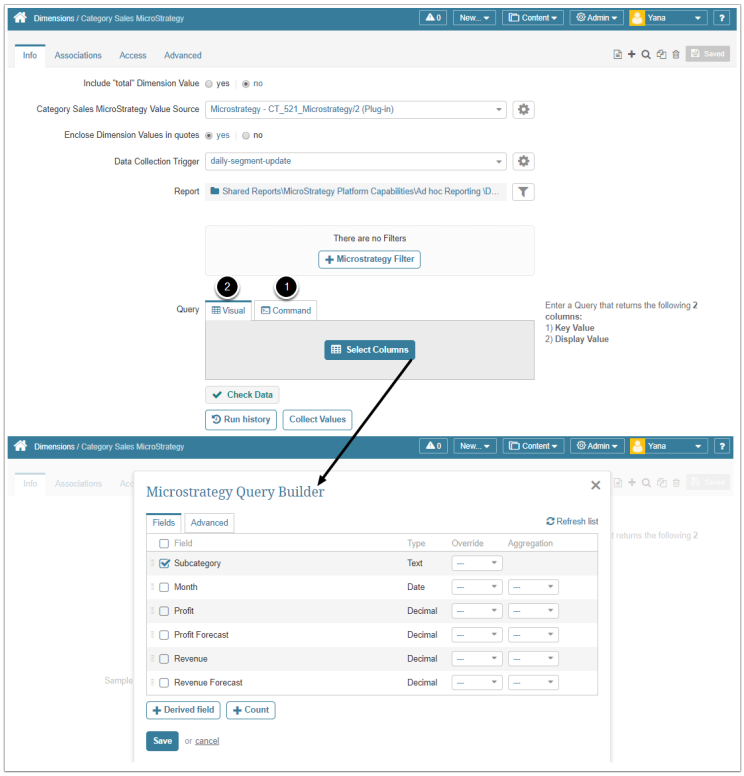


Provide the basic Dimension definition information, including

1. A unique **Name** for your Dimension
2. **Value Source:** Specify how Dimension Values will be collected for the new Dimension. In this example, we are selecting "MicroStrategy'"plugin from previous steps which is going to serve as a data source
3. **Report:** Define the MicroStrategy report that contains the required values in the **Select Report** pop up. In our example, we need to fetch Division Values, so we have selected a "Category Sales Report" item since it contains the required information.

Save your entries. The *Dimension Editor* opens.

2.2. Enter the command for fetching data

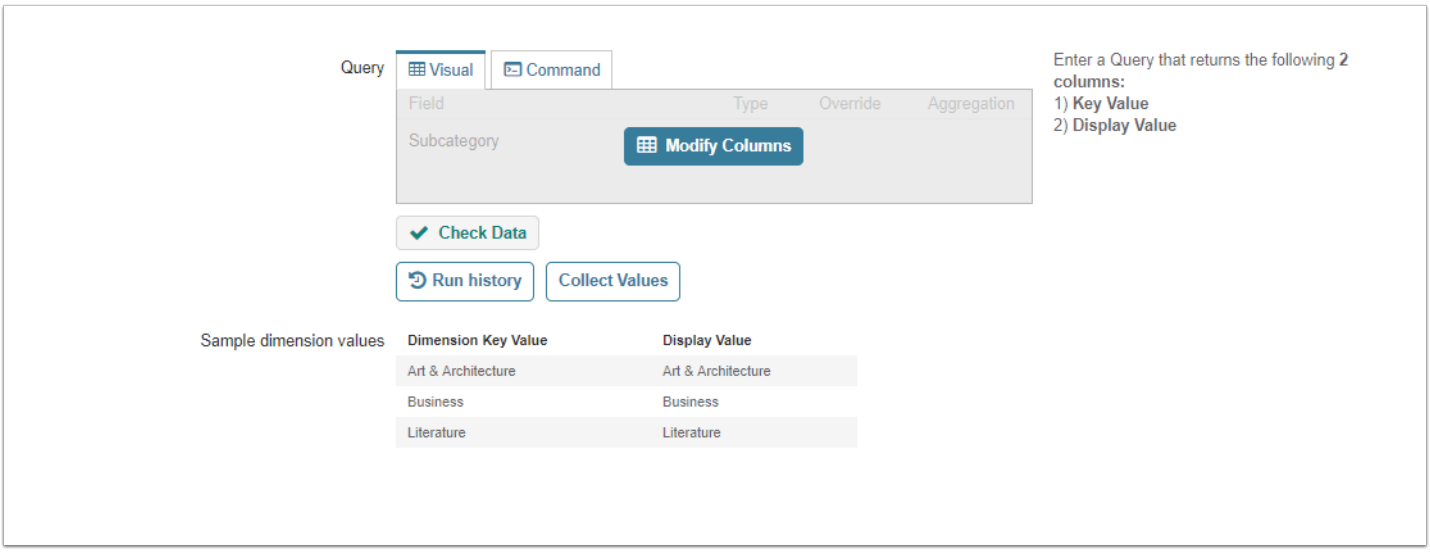


1. Enter the command manually

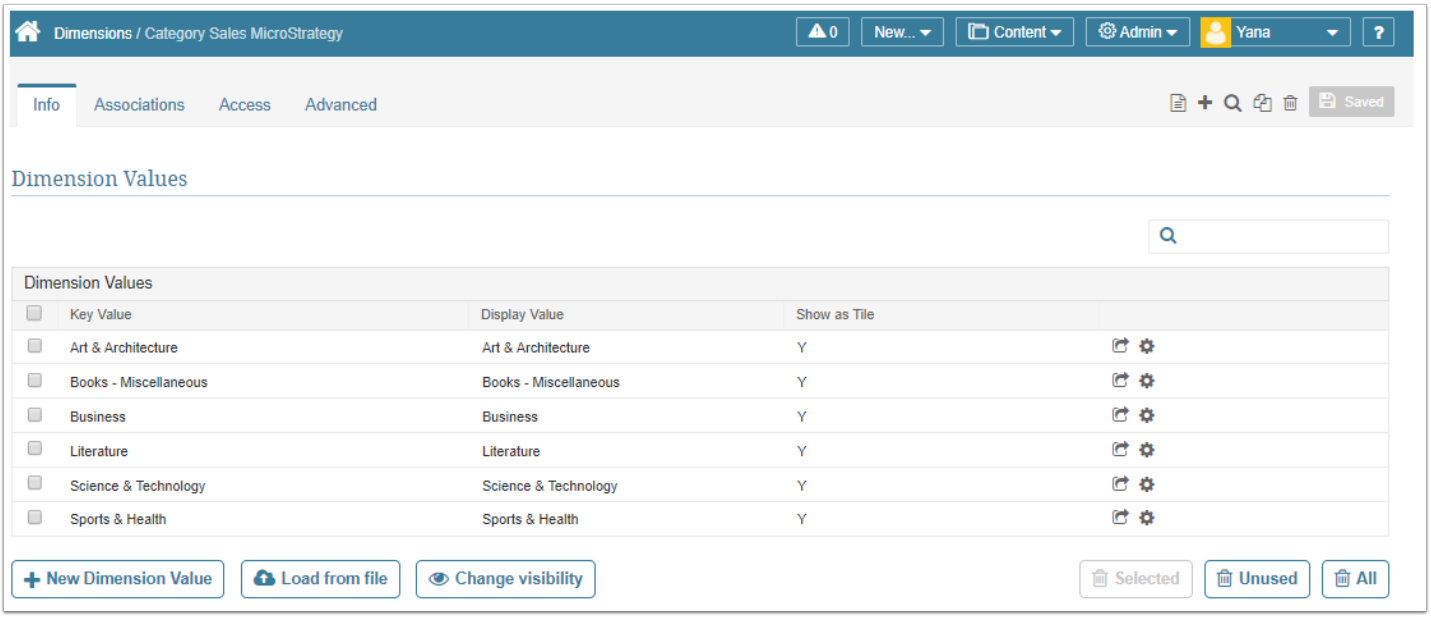
OR

2. Use the Visual Editor and choose the required fields

2.3. Check Data and Collect Values



2.4. Result



What would you like to do next?

You can now use this Dimension to create dimensioned elements from MicroStrategy.

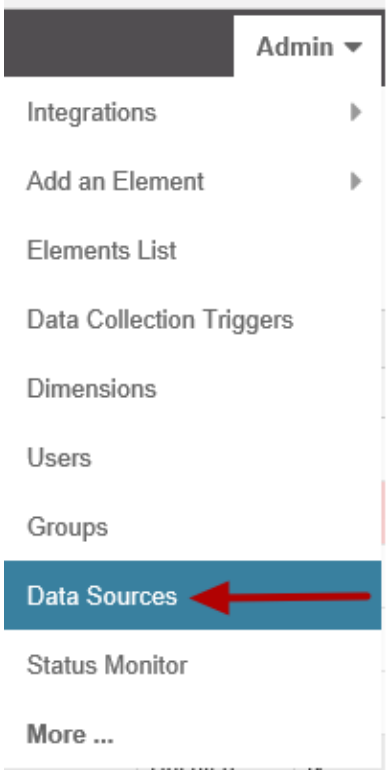
- Create a Dimensioned Report
- Create a Dimensioned Metric

23. Sourcing Data from Mixpanel

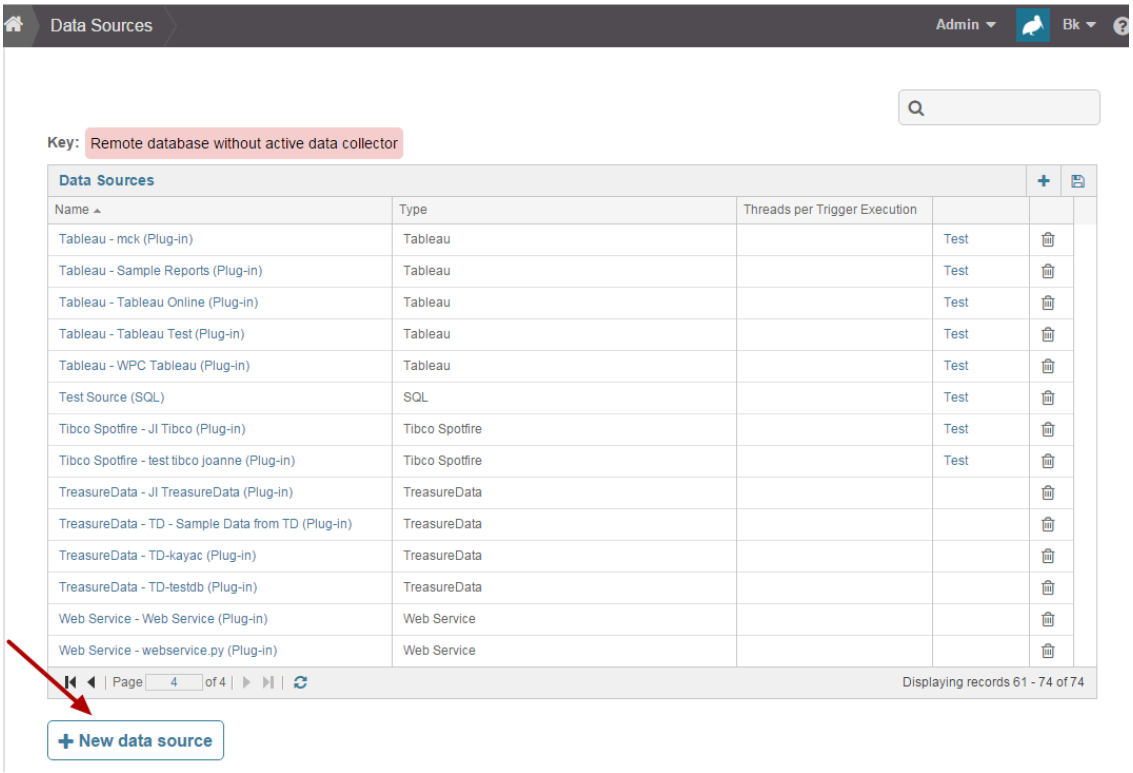
23.1 Establish Connectivity to Mixpanel

This article describes how to create a Data Source that can be used to retrieve event data from **Mixpanel** to be used in creating Metrics and Reports in Metric Insights.

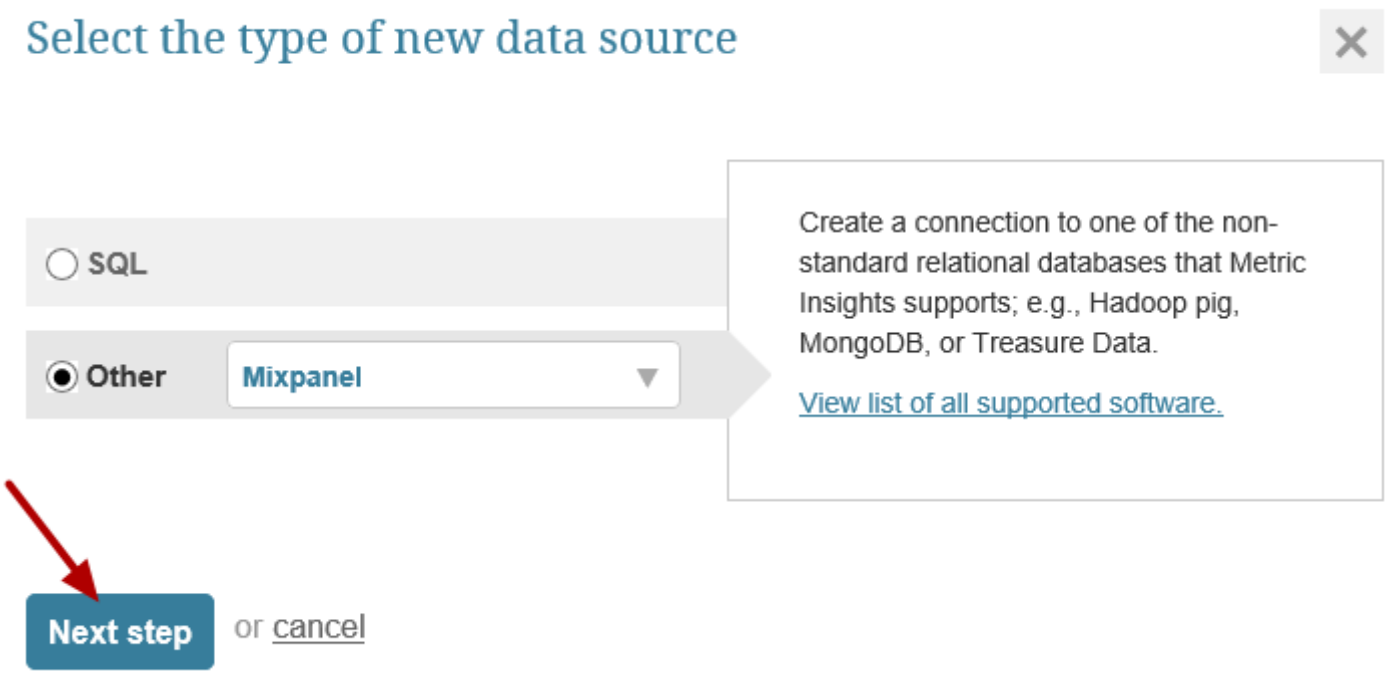
1. Select Data Sources from Admin drop-down



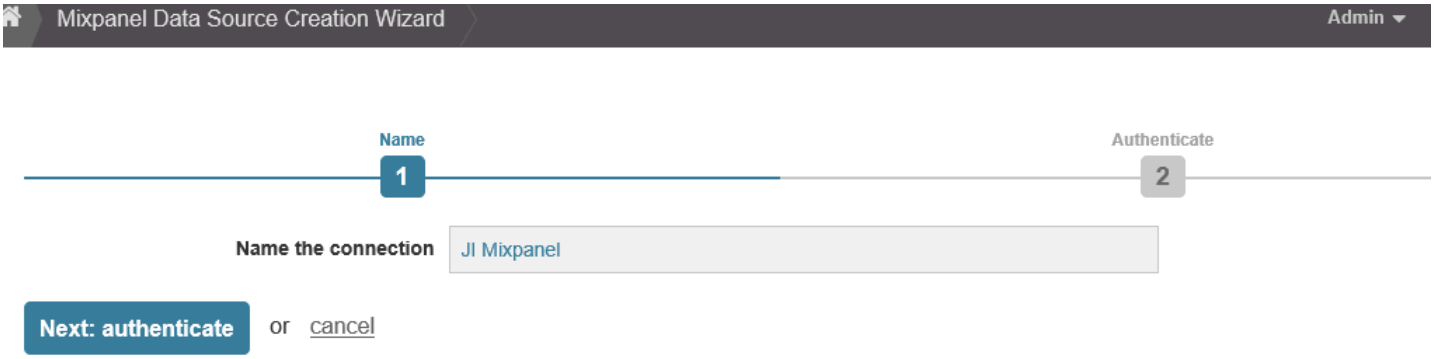
2. Add New Data Source



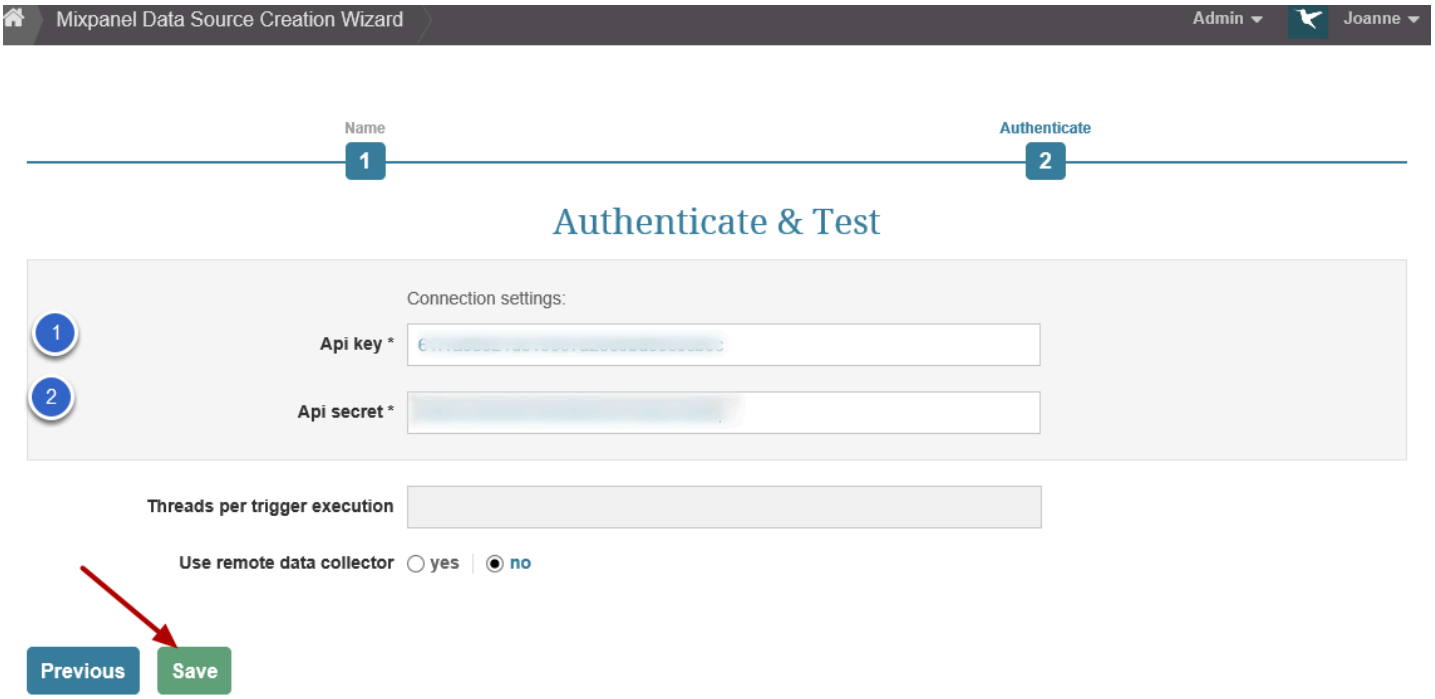
3. Select "Other" Data Source Type and choose "Mixpanel" from the drop-down



4. Input a meaningful Name and next step



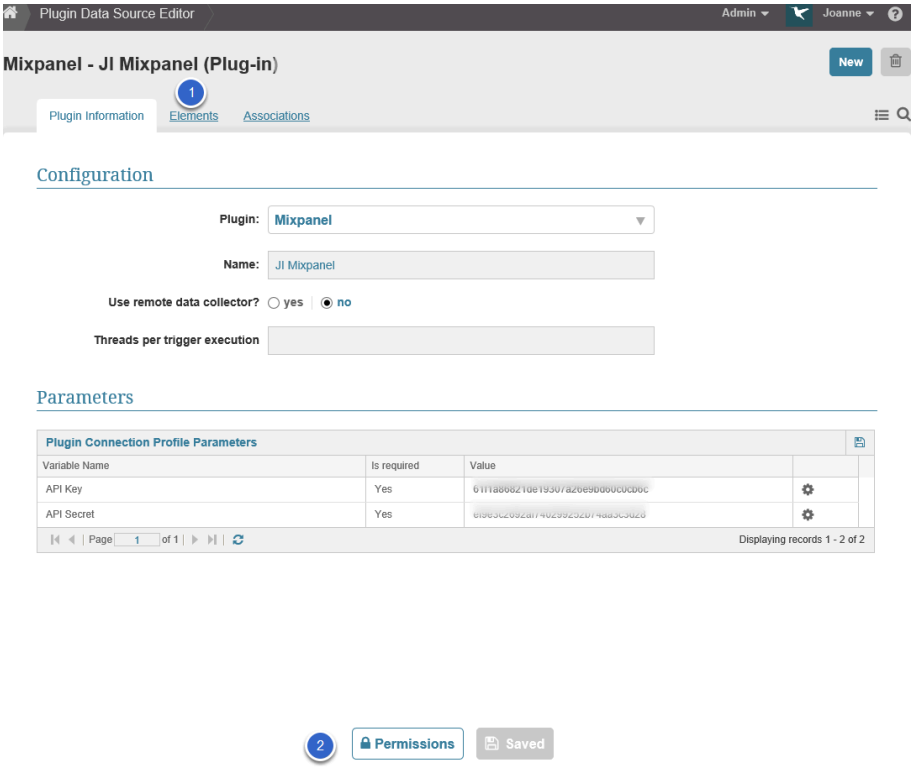
5. Authenticate and Test



1. Enter your **Api key**
2. Input your **Api secret**

Save

6. Full Data Source Editor displays

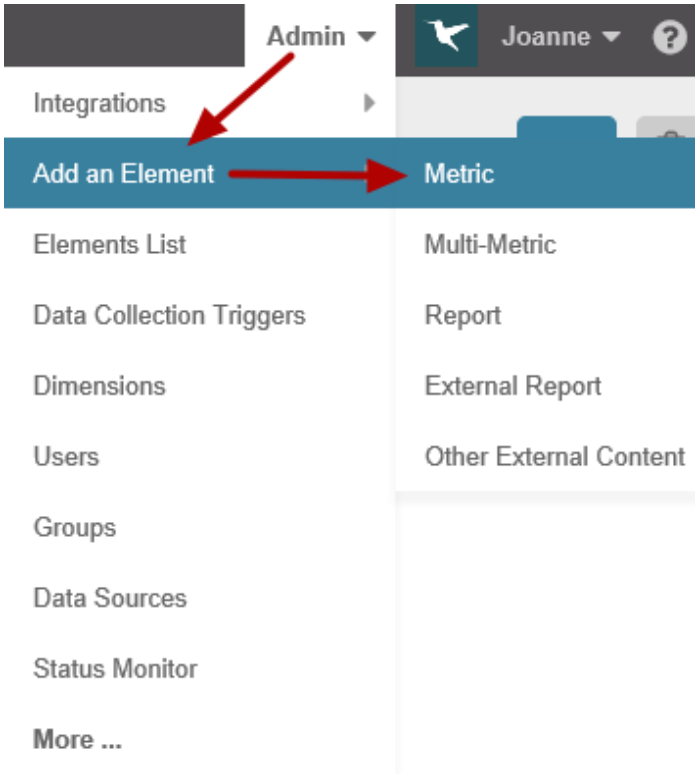


1. You can create **elements** directly from the Elements tab
2. You can assign **permissions** to Groups or Power Users here also

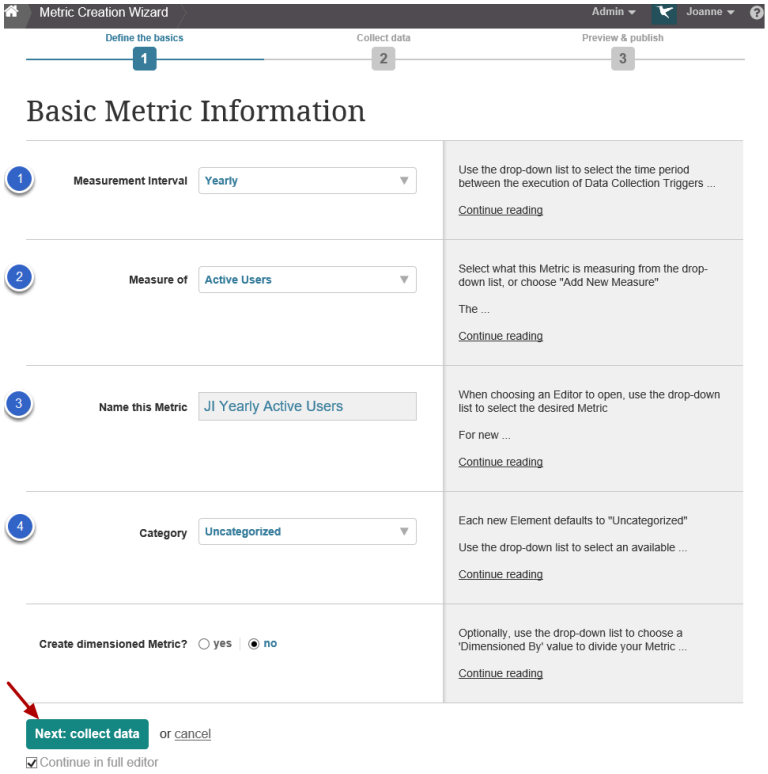
23.2 How to Collect Data from Mixpanel

The Mixpanel plug-in allows you to retrieve information from its tool that tracks "events' on websites and segments them by such dimensions as: source, campaign, medium, keyword. Using the Metric Insights plug-in allows you to use this data to create visualizations easily using our robust Metrics and Reports.

1. Add a new element based on your Mixpanel plug-in data source



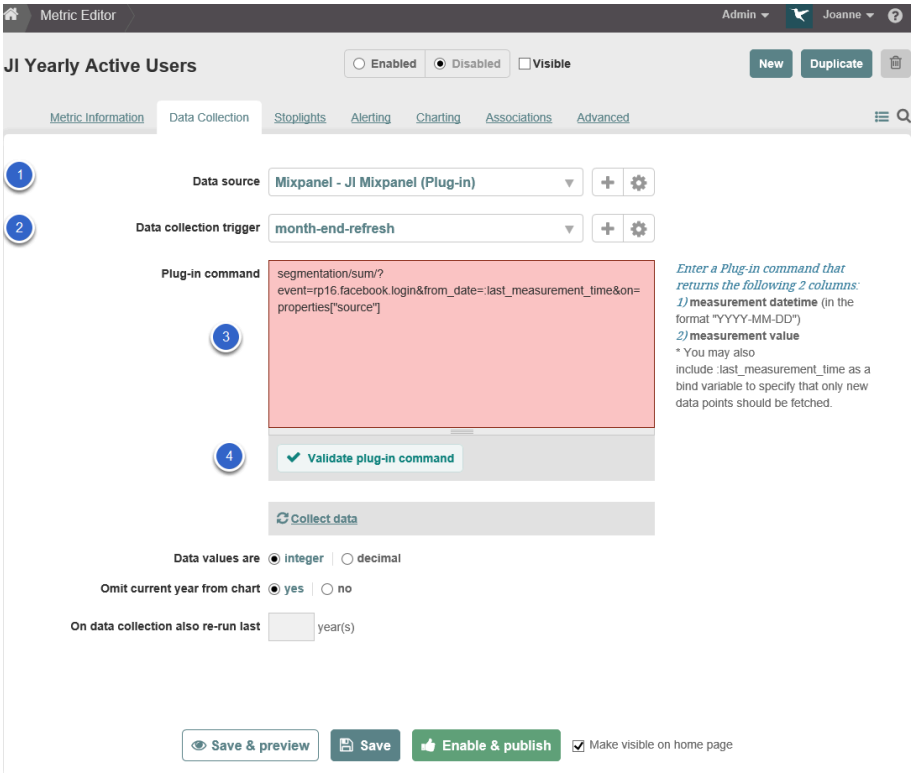
2. Provide basic information on Wizard (or Editor) - metric example



1. Select the **Measurement Interval** that applies to your element
2. Specify what this metric is **measuring**. If you do not see the measure that you want to use, you can create one from this drop-down
3. Give the element a unique **name**
4. Optionally, assign a **Category**

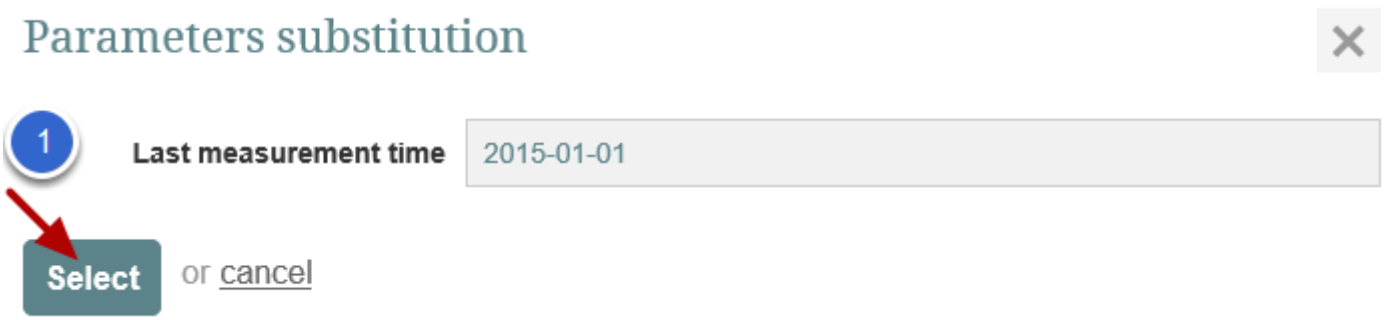
Next: (example is using full editor, but same steps apply when using Wizard)

3. Full Editor displays the Data Collection tab

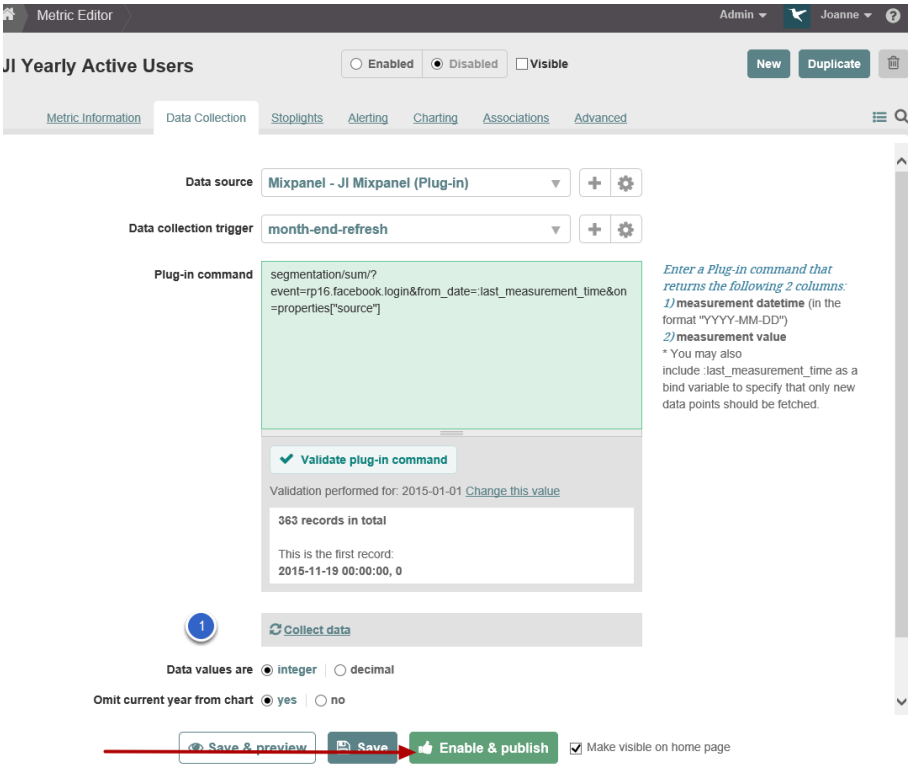


1. Select **Mixpanel** plug-in as **Data Source**
2. Set **Trigger**
3. Input **Plug-in Command**
4. **Validate Plug-in command**

3.1. Input Last measurement time



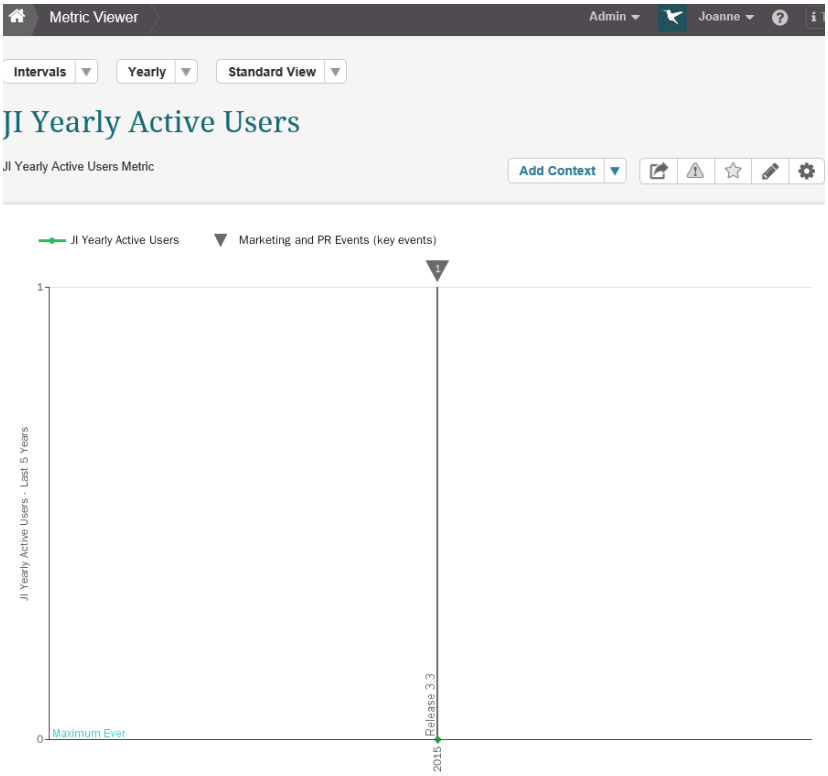
4. Collect data and publish



1. **Collect data**

Enable & Publish

5. Metric will be displayed in viewer

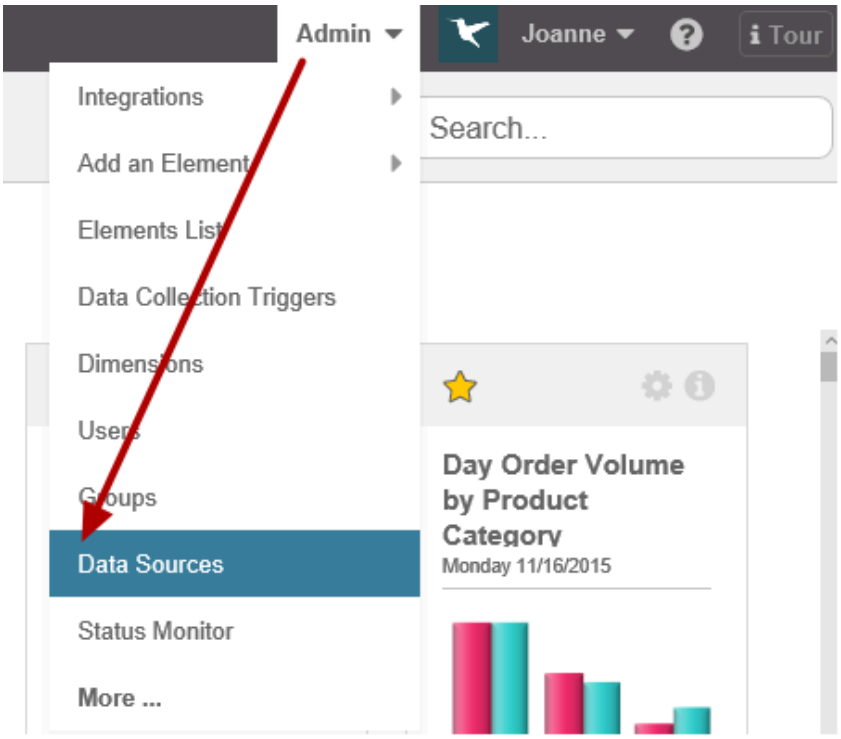


24. Sourcing Data from MongoDB

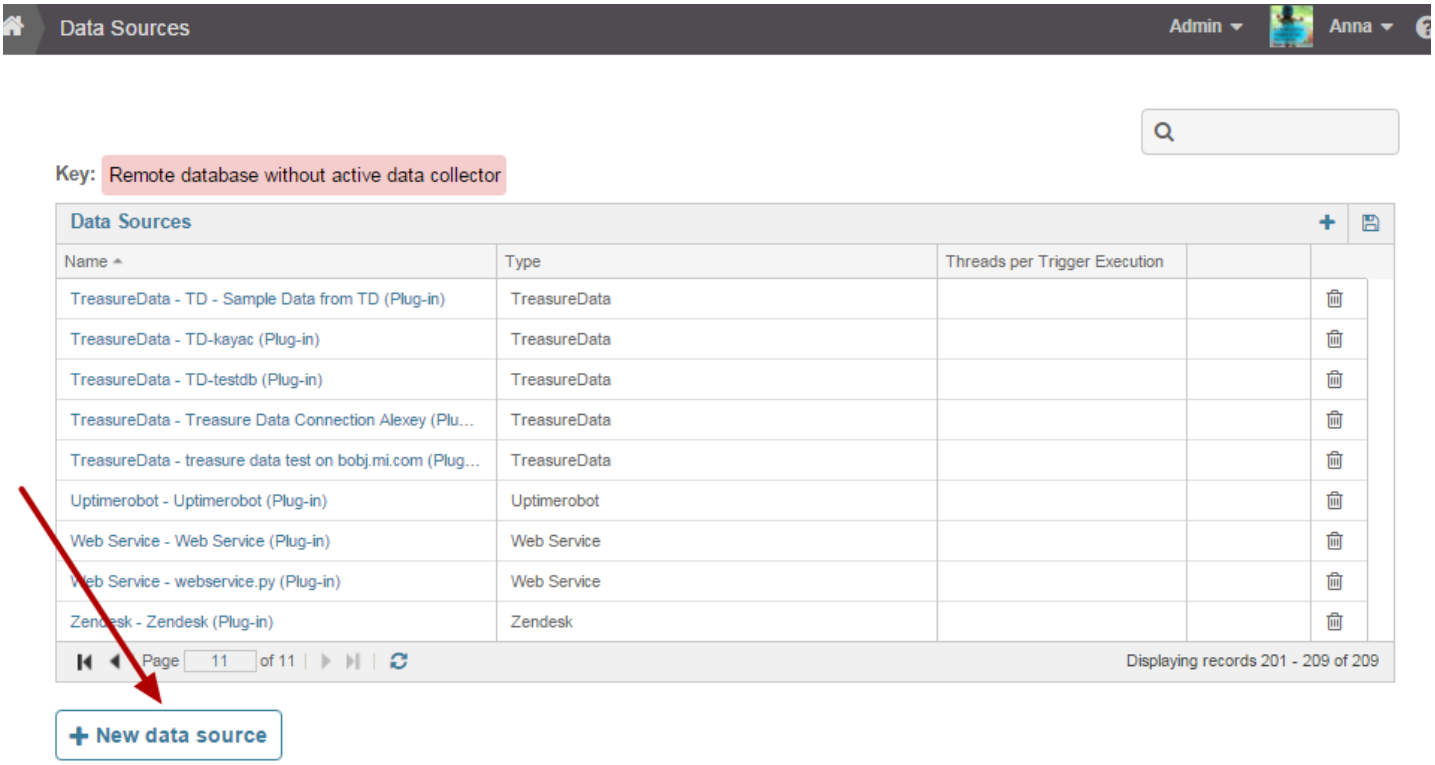
24.1 Establish Connectivity to MongoDB

An Administrator can use the process described in this article to create a new **Plug-in Data Source** that is required to allow Elements to fetch data from MongoDB to create a visualization in Metric Insights.

1. Go To Data Source Editor



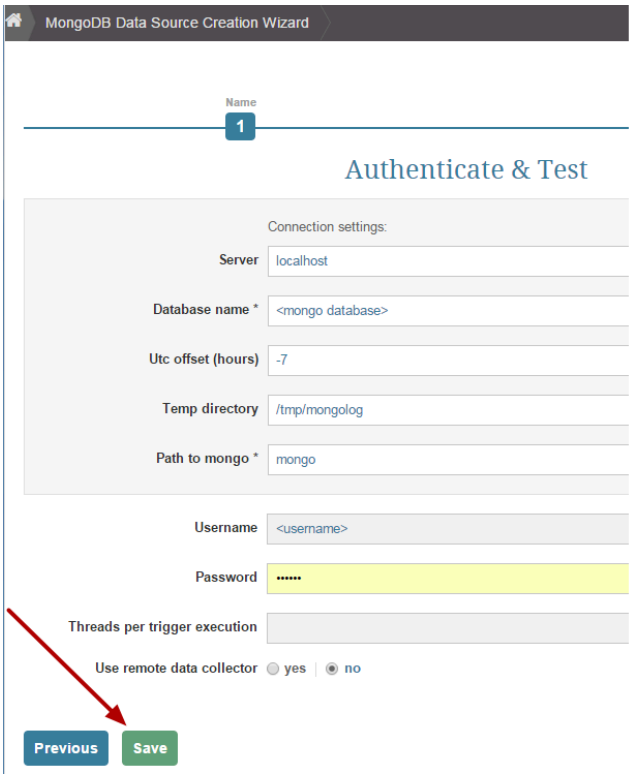
2. Add New Data Source



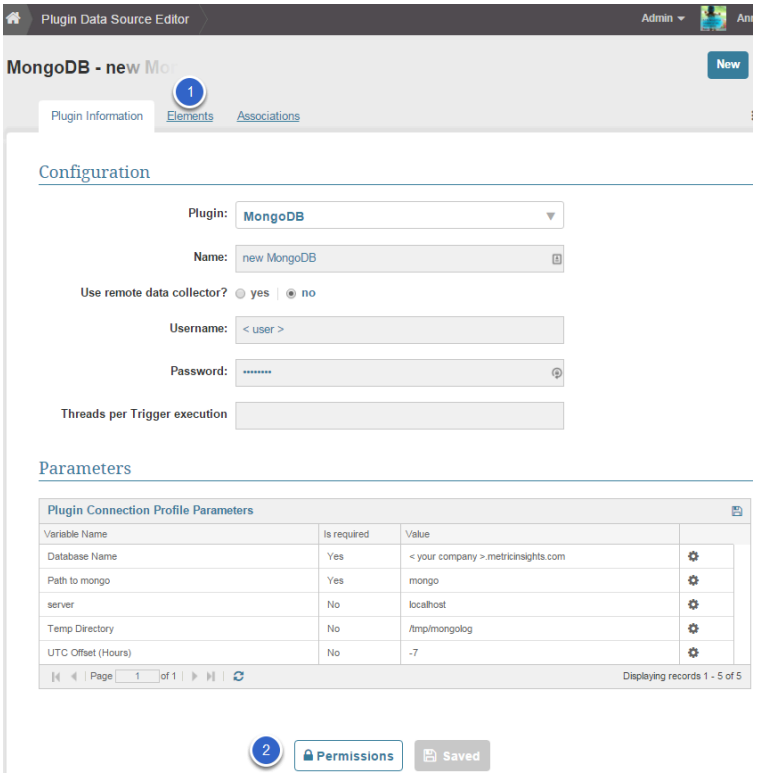
3. Select "Other" as Data Source Type and select MongoDB from the list.

4. Input a meaningful Name and authenticate

5. Enter Connection Settings



6. Full Data Source Editor displays

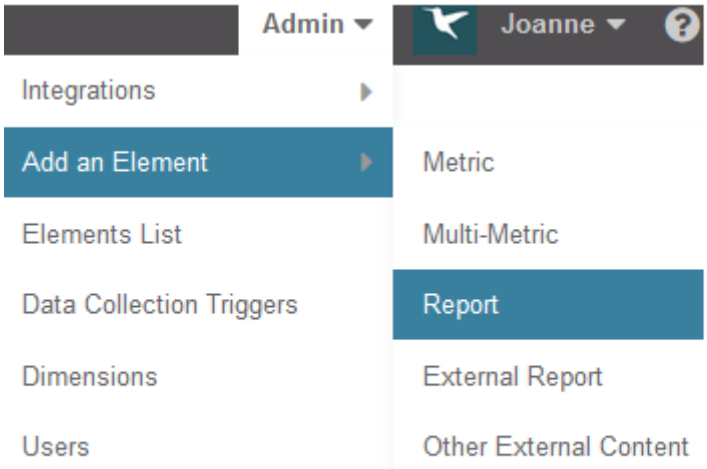


1. You can create **elements** directly from Elements tab
2. You can assign **permissions** to Groups or Power User here also

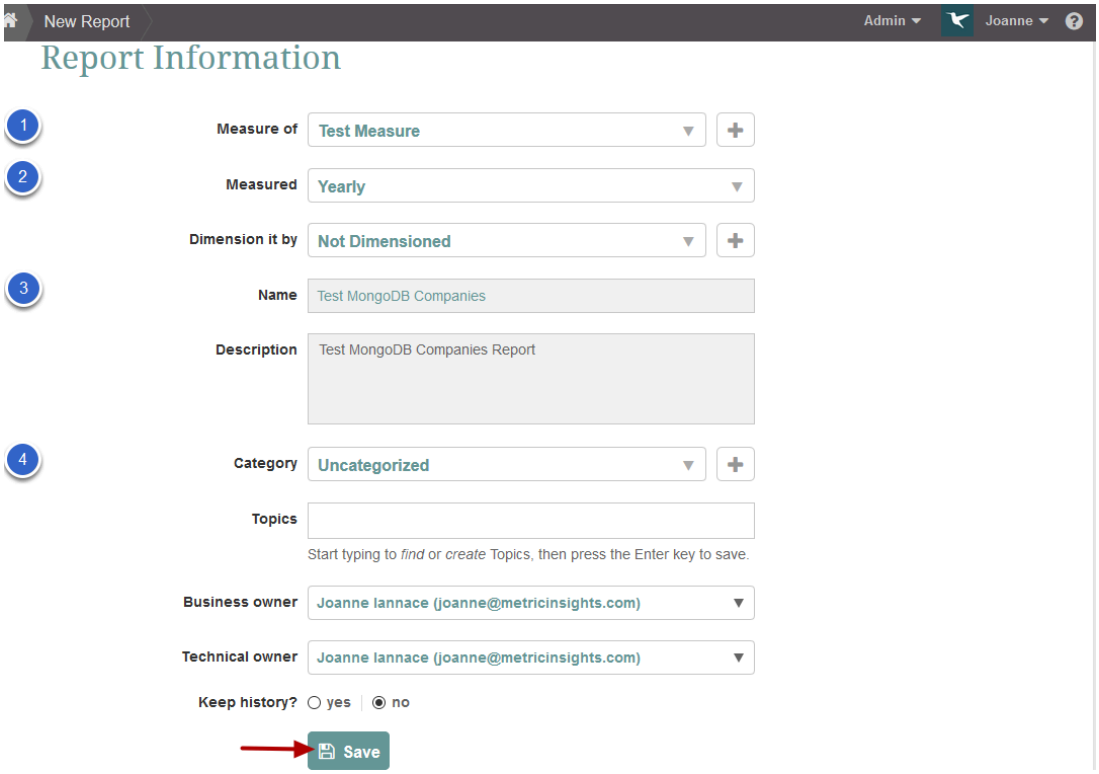
24.2 How to Collect Data using MongoDB Plug-in

This article will show you how to create an Element using the MongoDB plug-in as a data source. It assumes that you have already established connectivity to MongoDB.

1. Add a new element



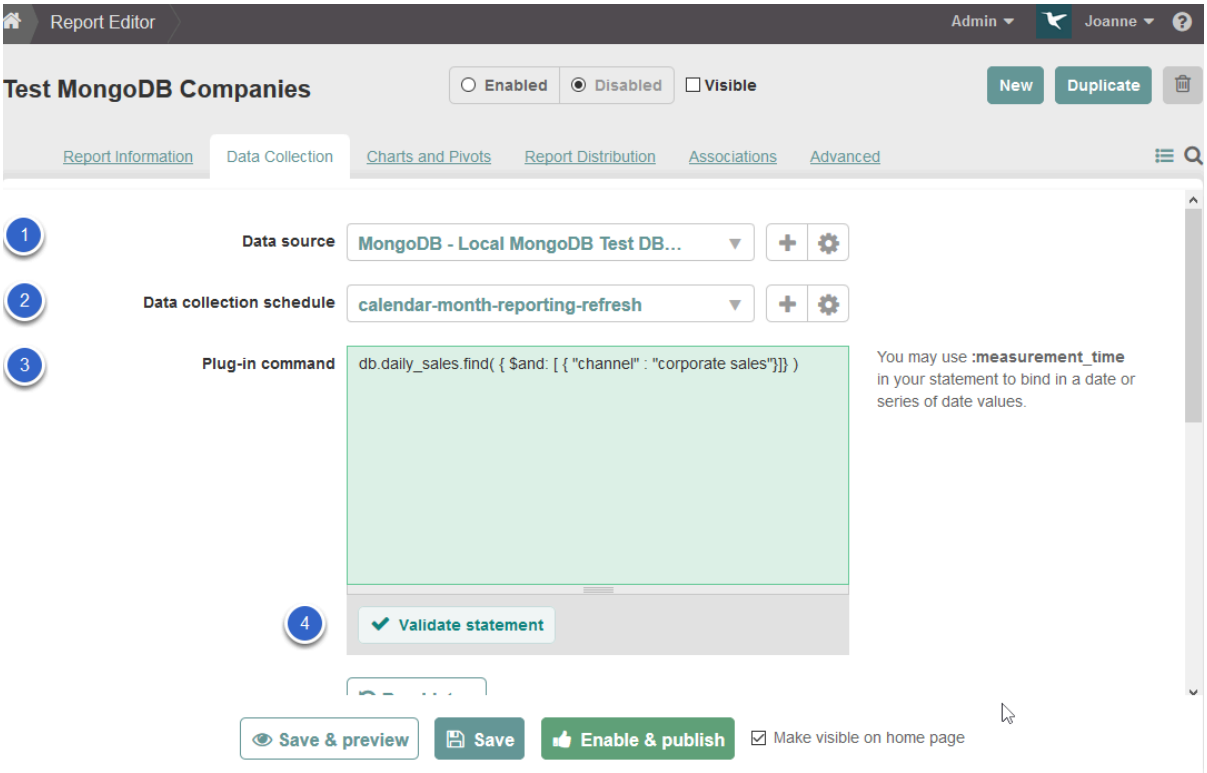
2. Provide basic information on Wizard (or Editor) - report example



1. Specify what this report is **measuring**. If you do not see the measure that you want to use, you can create one from this drop-down
2. Select the **Measurement Interval** that applies to your element
3. Give the element a unique **name**
4. Optionally, assign a **Category**

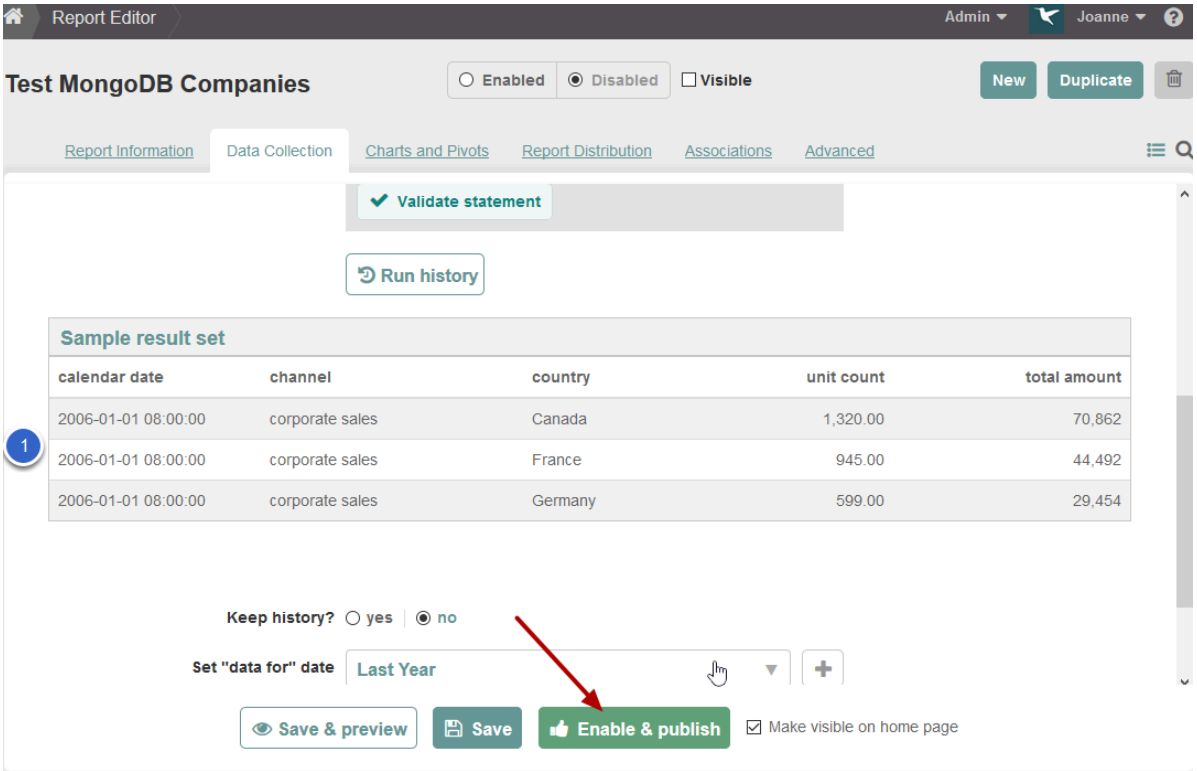
Save

3. Full Editor displays the Data Collection tab



1. Select BOBJ plug-in in **Data Source** drop-down
2. Set **Data collection schedule**
3. Input **Plug-in command**
4. **Validate statement**

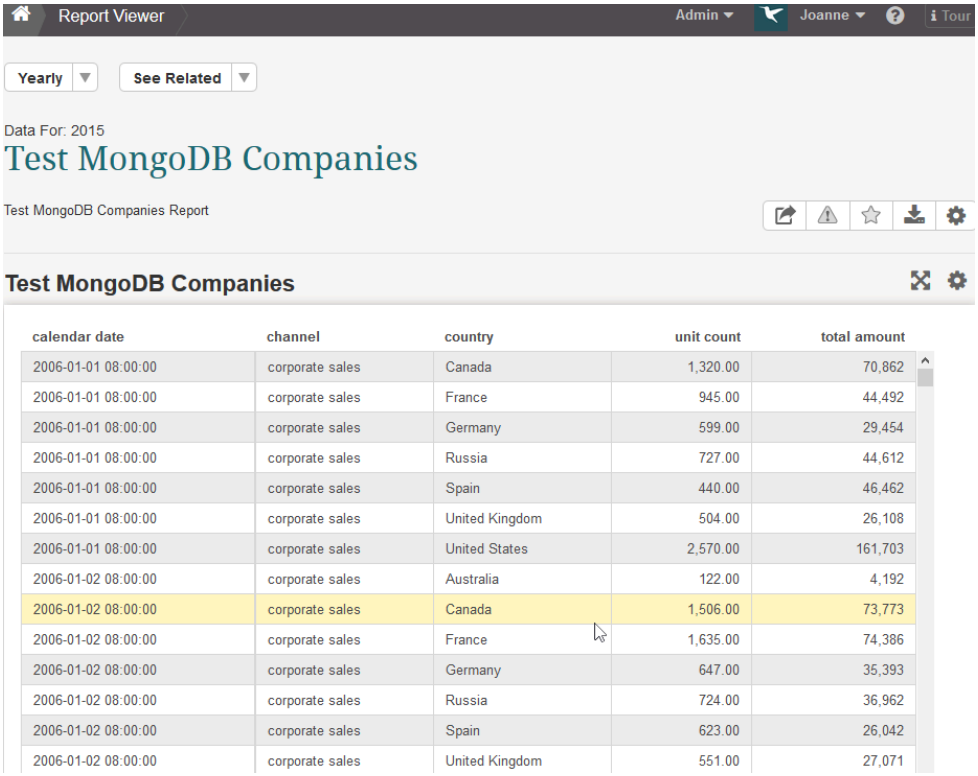
4. Enable & Publish



1. **Sample result set** is Displayed

Enable & publish

5. Report will be displayed in viewer



calendar date	channel	country	unit count	total amount
2006-01-01 08:00:00	corporate sales	Canada	1,320.00	70,862
2006-01-01 08:00:00	corporate sales	France	945.00	44,492
2006-01-01 08:00:00	corporate sales	Germany	599.00	29,454
2006-01-01 08:00:00	corporate sales	Russia	727.00	44,612
2006-01-01 08:00:00	corporate sales	Spain	440.00	46,462
2006-01-01 08:00:00	corporate sales	United Kingdom	504.00	26,108
2006-01-01 08:00:00	corporate sales	United States	2,570.00	161,703
2006-01-02 08:00:00	corporate sales	Australia	122.00	4,192
2006-01-02 08:00:00	corporate sales	Canada	1,506.00	73,773
2006-01-02 08:00:00	corporate sales	France	1,635.00	74,386
2006-01-02 08:00:00	corporate sales	Germany	647.00	35,393
2006-01-02 08:00:00	corporate sales	Russia	724.00	36,962
2006-01-02 08:00:00	corporate sales	Spain	623.00	26,042
2006-01-02 08:00:00	corporate sales	United Kingdom	551.00	27,071

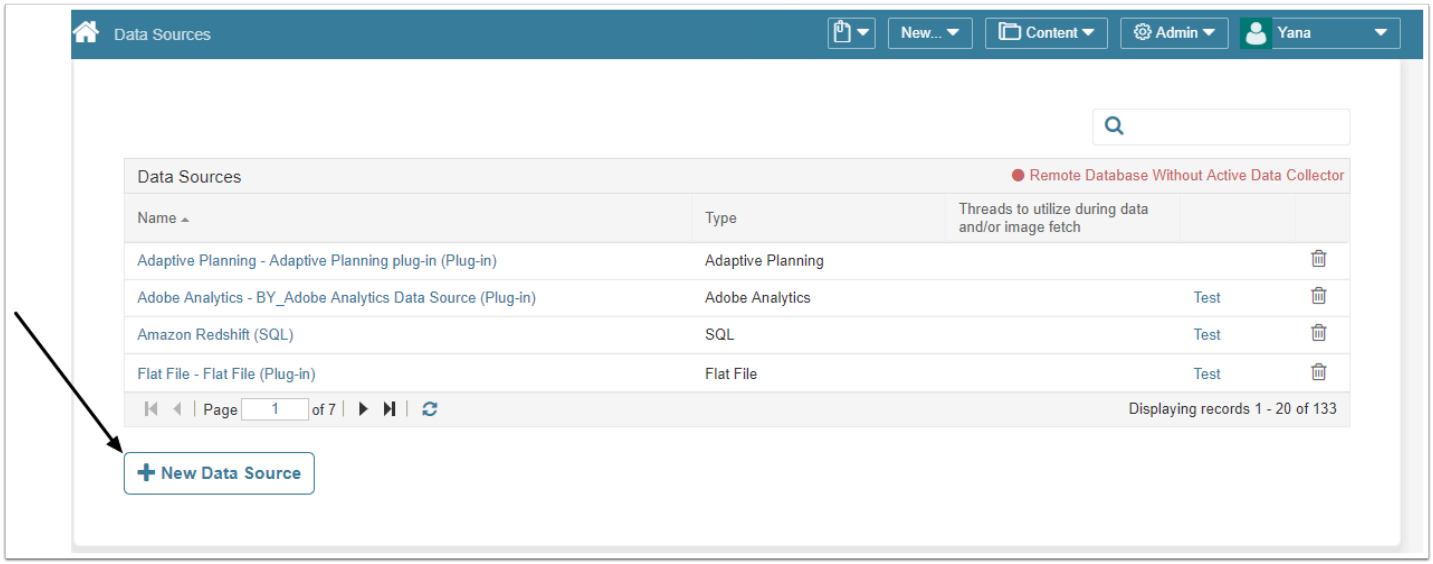
25. Sourcing Data from OBIEE (Oracle Business Intelligence)

25.1 Establish Connectivity to Oracle Business Intelligence

This article describes how to connect to **Oracle Business Intelligence** in order to load data into Datasets and Reports in Metric Insights.

[Release 6.x]: 'Remote Data Collector' renamed to 'Remote Data Processor'

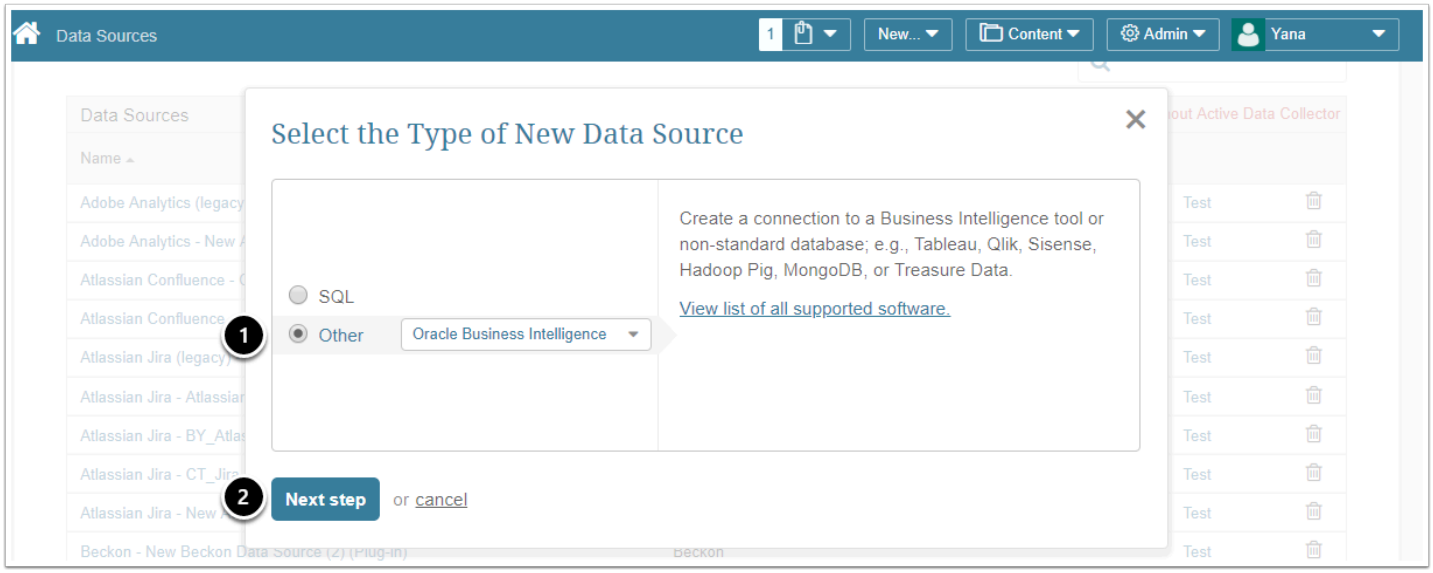
1. Access Admin > Data Sources



At the bottom of the screen click **[+ New Data Source].**

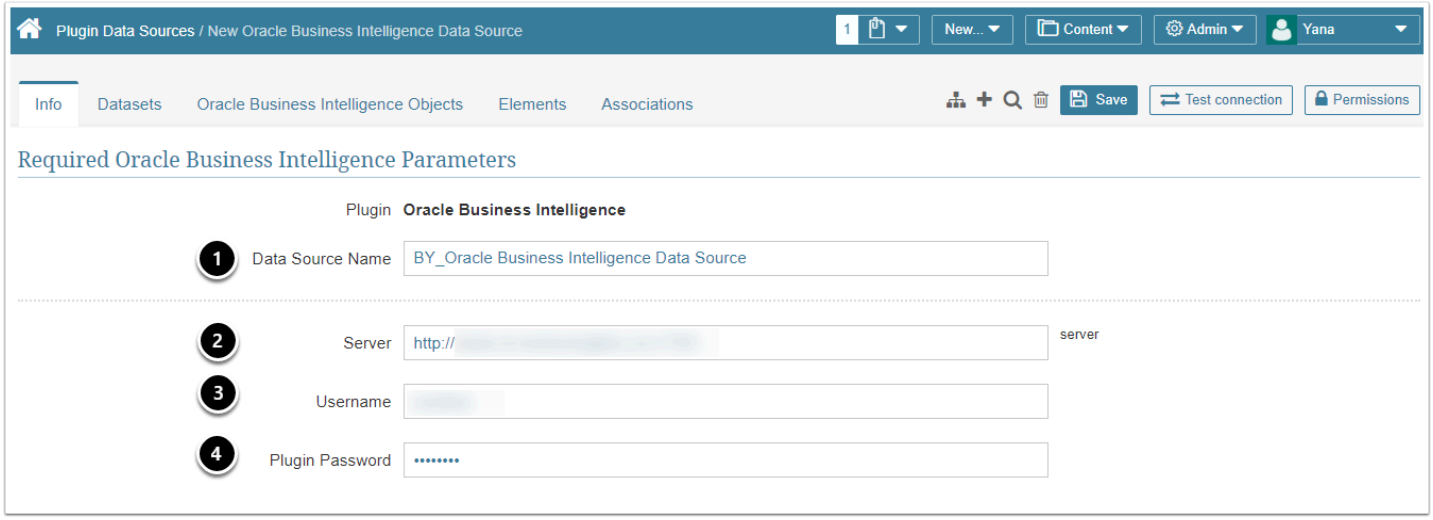
The *Select the Type of New Data Source* pop-up opens.

2. Select the Type of the New Data Source



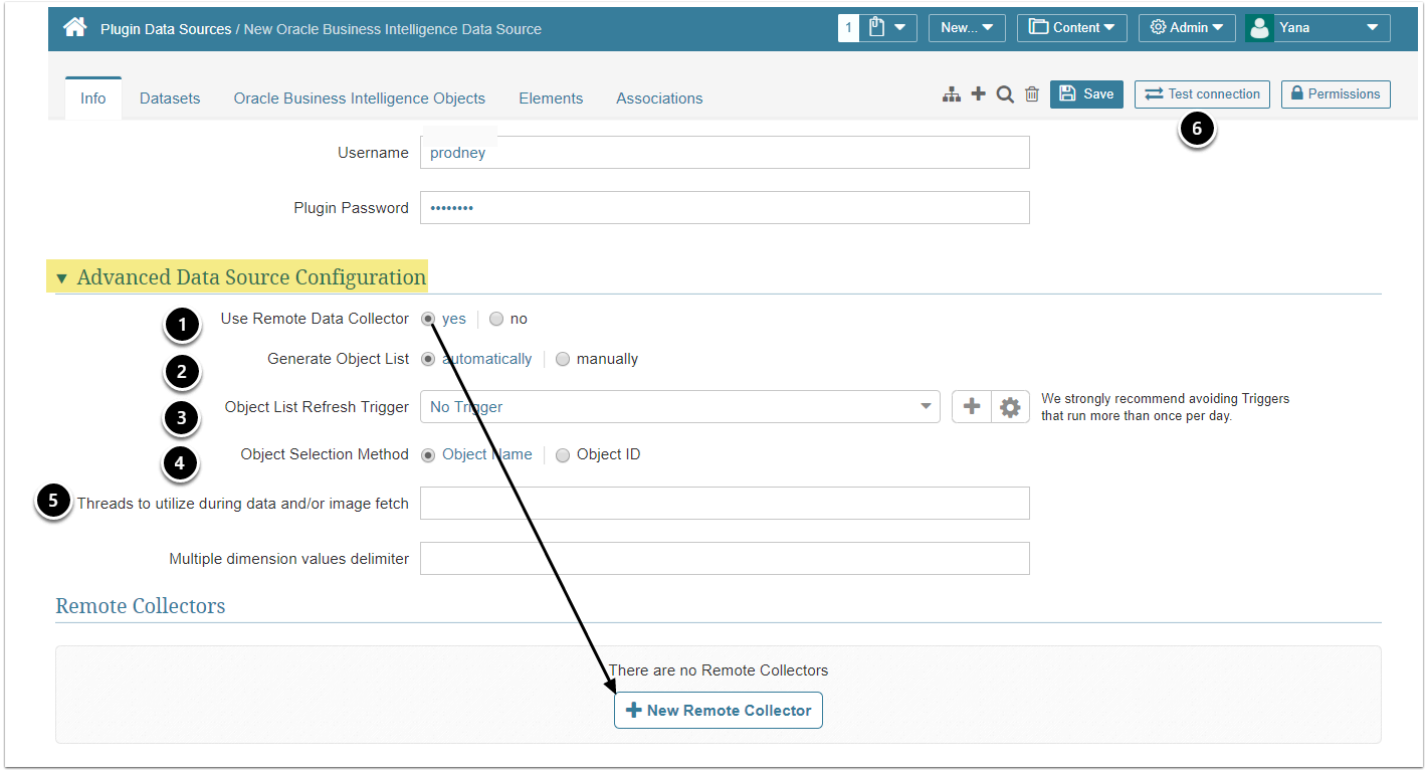
1. Select "Other" and choose **"Oracle Business Intelligence"** from the drop-down list
2. Move to the **Next step**

3. Provide the Required Parameters



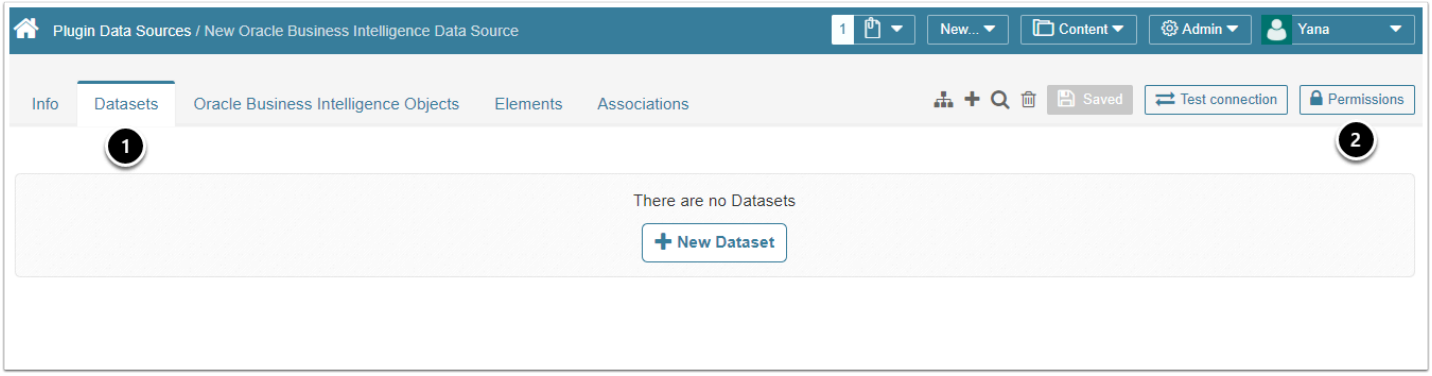
1. **Data Source Name:** set a name for the data source
2. **Server:** define the server protocol (**http** or **https**) and a hostname
3. **Username:** note that your **Username** must be in the same format that your *Oracle Business Intelligence* server uses for authentication
4. **Password:** provide your password credential

4. Advanced Configuration



1. **Use Remote Data Collector:** is set to "no" by default
   - If required, switch to "yes" and add a Remote Data Collector by clicking **[+New Remote Collector]**
2. **Generate Object List**
   - *automatically*: all Reports are going to be fetched by the system
   - *manually*: Reports may be added one-by-one or via CSV file
3. **Object List Refresh Trigger:** from the dropdown, select the Trigger that will be used to fetch data via the Oracle Business Intelligence plugin
4. **Object Selection Method:** specify how Oracle Business Intelligence Reports will be fetched
5. Optionally, state the maximum number of concurrent **Threads to utilize during data and/or image fetch** to be used in background processing when the system updates Reports for this Data Source
   - If you do not specify any value for this setting, batch data collection processing will be single-threaded
6. **Test Connection** (this will also **Save** your data)

5. Other settings



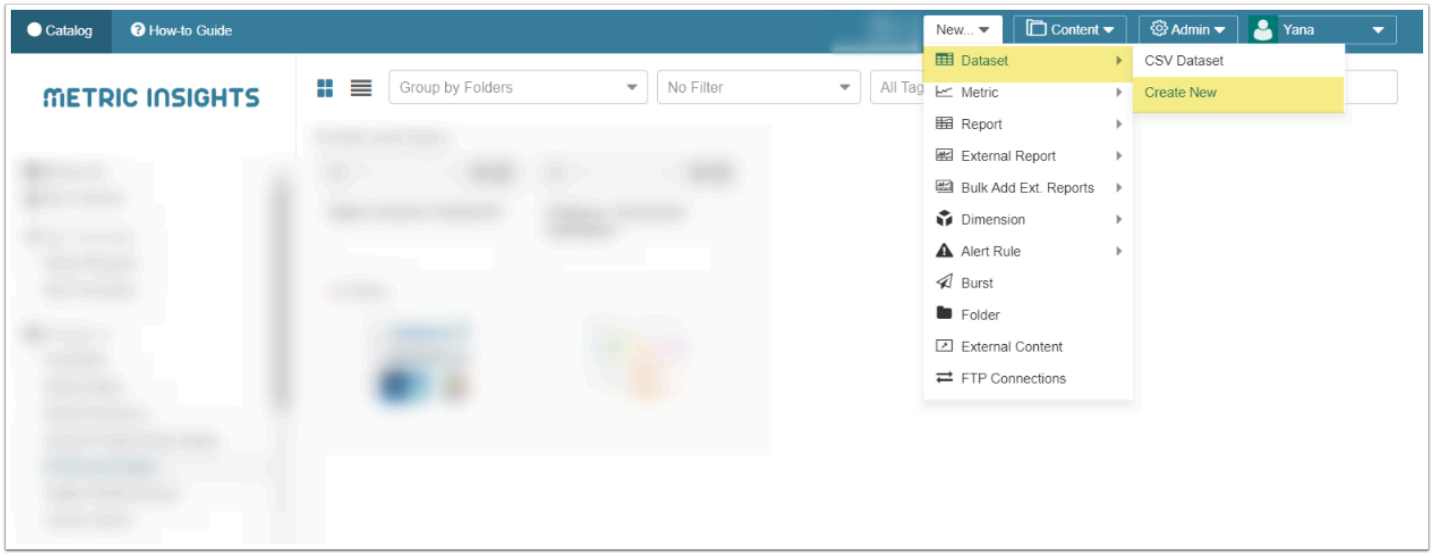
1. You can create **Datasets** directly from the respective tab
2. Click **Permissions** to assign permissions to Groups or Power Users

25.2 Collect data from Oracle Business Intelligence

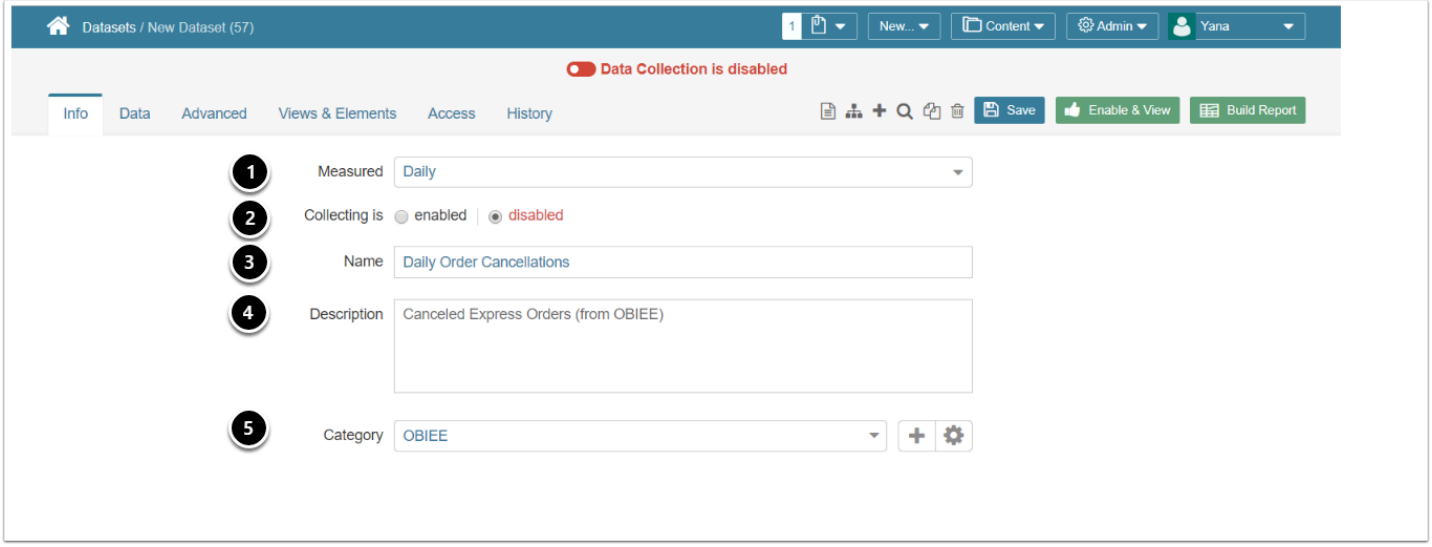
This article details how to create a Dataset populated with data sourced from **Oracle Business Intelligence**.

It assumes that you have already established connectivity with your Oracle Business Intelligence server via the respective plugin connection profile.

1. Access New > Dataset > Create New



2. Dataset Editor > Info tab



Define the basics:

1. **Measured:** select the measurement interval that applies to the level of aggregation that you want in your result set
2. **Collecting**: new Datasets are always disabled by default to make sure that you can take time to configure them properly before enabling. This setting is duplicated at the top of the screen
3. **Name:** provide a unique name for your Dataset. Preferably, the Dataset name should explain what kind of data it contains
4. **Description:** optionally, provide any additional information about your Dataset
5. **Category:** specify the Category where you Dataset will be placed

Move to the ***Data tab*** to define the source of data and how often it should be updated.

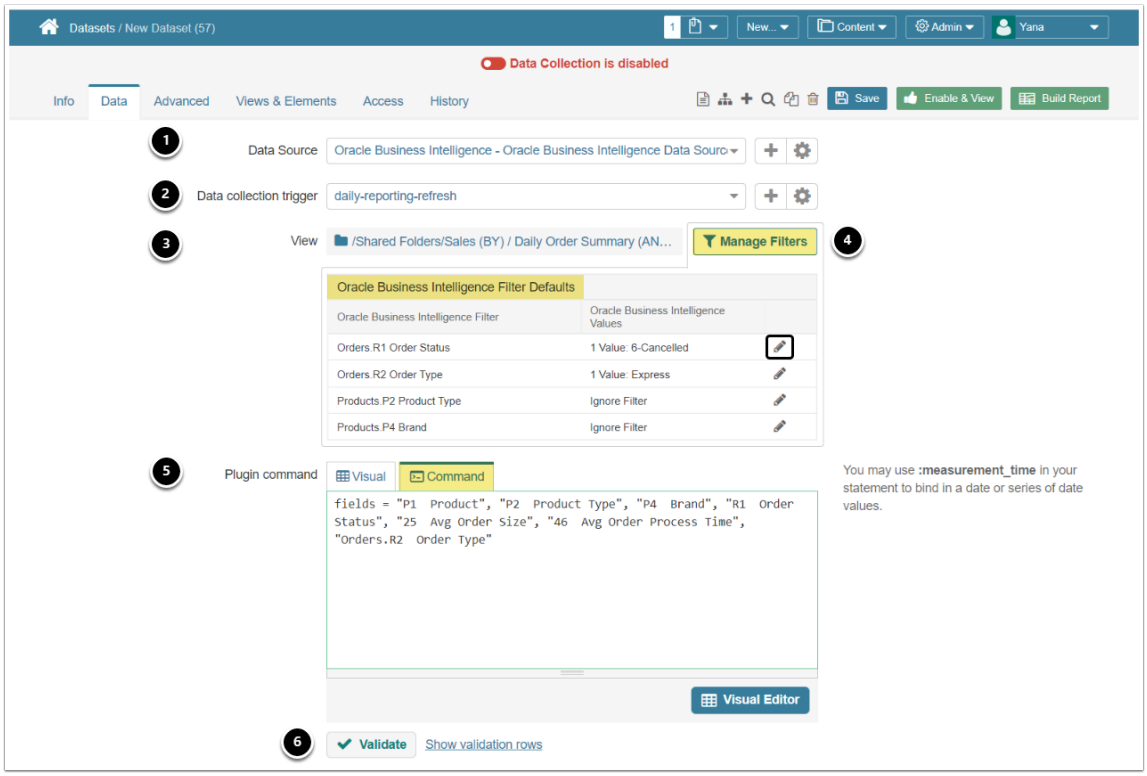
3. Define the Settings for Data Collection

Data fetching can be configured using the following options:

- Plugin query
- Visual Editor

See details below.

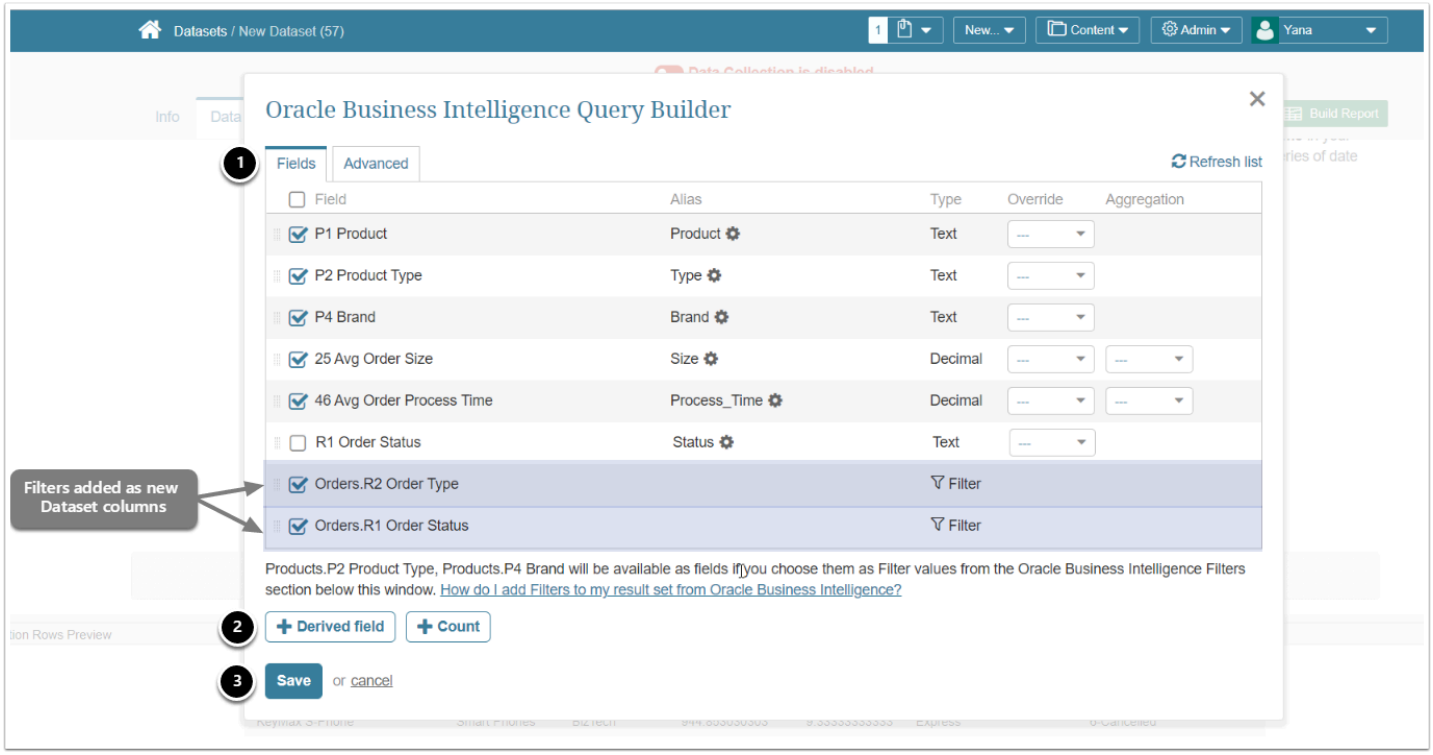
3.1. Using the Plugin Command



1. **Data Source**: select the connection profile you have created for *Oracle Business Intelligence*
2. **Data collection trigger**: specify the Trigger that will be used to collect data for your Dataset
3. **Element:** select an *Oracle Business Intelligence Object* that should serve as a basis of your Dataset
4. Click **[Manage Filters]** to add Filters to your data
   - All added Filters will be displayed under **Oracle Business Intelligence Filter Defaults** below
   - Click the **Edit (Pencil)** icon to specify how many Values will be used for data filtering
5. In the **Command tab**, input a **Plugin Command** *(as exemplified above)* listing all the data that needs to be fetched from *Oracle Business Intelligence*
6. **Validate** your query

> **Note!**
>
> Field and Filter Names must be enclosed in **double** quotes.

3.1.1. Using the Visual Editor



The **Oracle Business Intelligence Query Builder** allows for data fetching without the need to learn the plugin syntax and helps avoiding typos/mistakes.

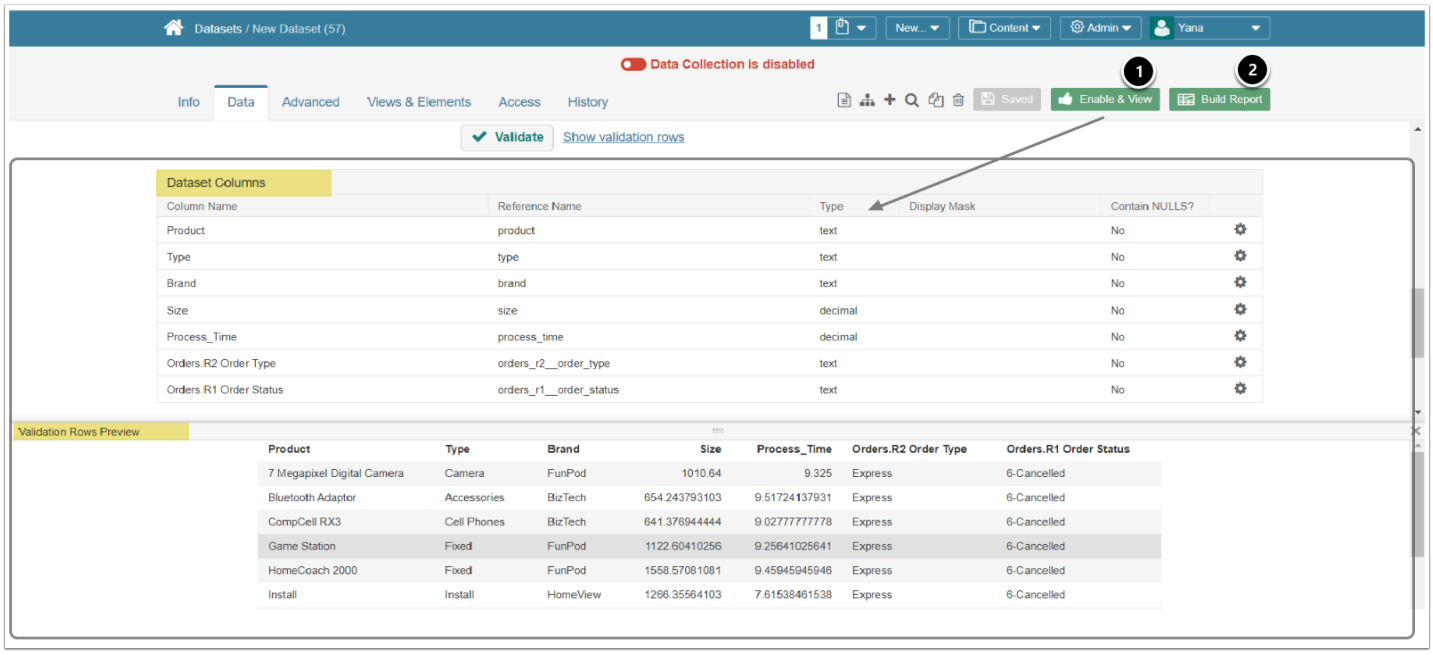
1. Select the **fields** for your Dataset
2. *Optionally,* add **Derived fields** and/or **Count** of duplicate rows

Save your settings. Plugin command validation will start automatically.

Note!

1. *Derived fields* can later be modified using the **Edit (Gear) icon**
2. *Filters added as a field to the results set* are distinguishable by a **Filter (Funnel) icon**
3. Filters in Datasets build from OBIEE data are added as new columns. To avoid Column duplication, do not select the corresponding fields to be shown in the results set.

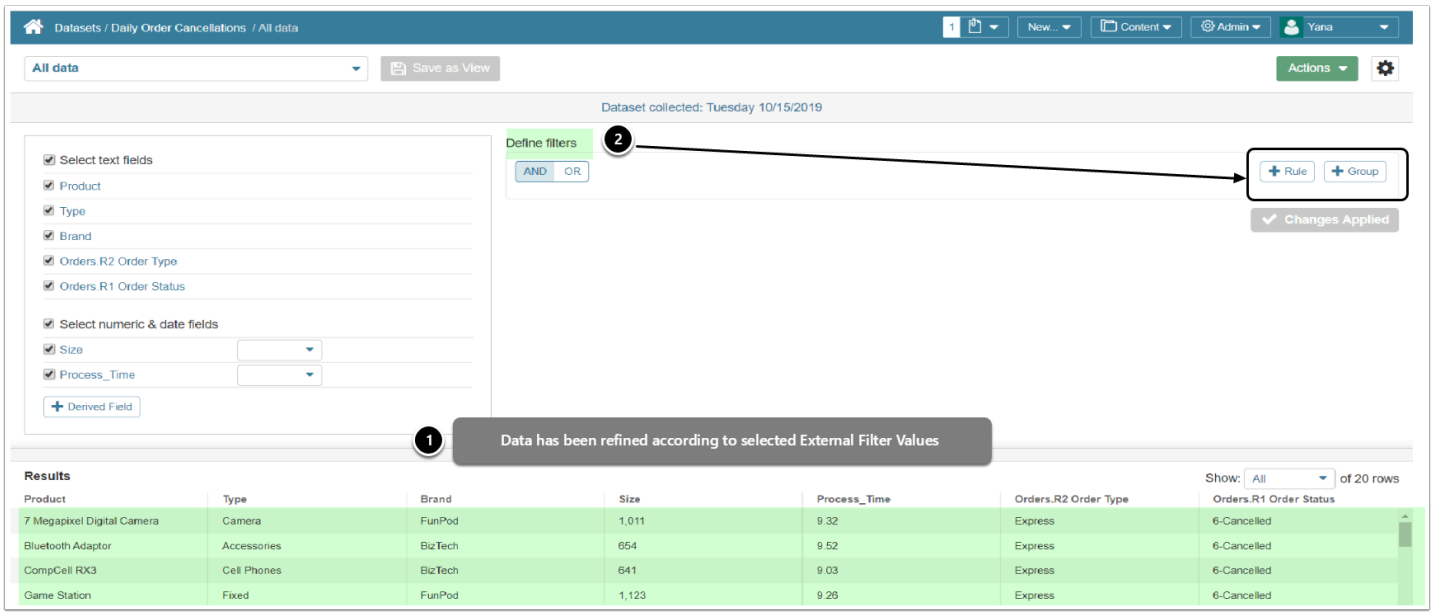
4. Plugin command will be validated and data collected on Save



1. If the command is validated successfully, the **Dataset columns** and **Data Preview** are going to be shown below.
2. At the upper right corner of the screen, click **Enable & View**.

5. Dataset will be displayed in Viewer

If any Filter has been applied, pre-filtered data will be displayed in Viewer.



In the **Dataset Viewer**:

1. In the Results Section, you will see data with already applied **External Filters**
2. You can further refine your data with Dataset's *internal filtering options* by **applying Rules** and **Grouping Data**:
   - For more information on using Internal Filters, refer to Create a Dataset View
   - For general instruction on building Datasets, see Create a Dataset from any Data Source

25.3 Pre-filtering Oracle Business Intelligence data

As of Release 5.6.+, pre-filtering functionality has been redesigned to include auto-retrieval of Filters, while retaining the option to manually add Filter Values to Datasets/Elements sourced from Oracle Business Intelligence (OBIEE).

OBIEE Filters can be added by:

1. Retrieving Filter Names from OBIEE (auto-retrieval of all Filters/Filter Values from Oracle Business Intelligence)
2. Loading Filter Values from OBIEE (auto-retrieval of a single Filter and its Values from Oracle Business Intelligence)
3. Loading Filter Values from a Dataset (auto-loading of Filter Values from a Dataset in Metric Insights)
4. Mapping to Dimension Values in Metric Insights (auto-mapping of Filter Values in Oracle Business Intelligence to Dimension Values in Metric Insights)
5. Entering Filter Values Manually

PREREQUISITES:

- Establish Connectivity to OBIEE

Capabilities Matrix for OBIEE Filters

OBIEE Objects	Prefiltering	OBIEE Filters passed to Metric Insights	Notes
Analyses	can be prefiltered	Column Prompts	Analyses must have *Column Prompts*. Only this type of Prompt Filter can be passed to Metric Insights.
Reports	can be prefiltered	Parameters	Reports can be prefiltered in Metric Insights if they are sourced from *Data Models* with added *Parameters*.
KPIs	not supported	not applicable	not applicable

⚠ Once Filters are added to a Dataset/Element for the first time, they will automatically be added to all new respective Datasets/Elements with the same Data Source.

NOTE:

- External Filters are tied to OBIEE Objects (Analyses and Reports), not Metric Insights' Objects/Elements. This allows Filters to be reused multiple times.
- Redundant Filters or Filter Values can be set to "ignore".

Filter names automatically fetched from Oracle Business Intelligence (OPTION: Retrieving Filter Names from OBIEE) are passed to *Metric Insights* in the following format: `table.column`

In other cases, Users will have to enter OBIEE Filter names manually in the corresponding fields.

1. Loading Filter Values from OBIEE
2. Loading Filter Values from a Dataset
3. Mapping to Dimension Values in Metric Insights
4. Entering Filter Values Manually

To find correct OBIEE filter names, refer to:

- How to get Column Prompt names for Analyses?
- How to get Parameter names for Reports?

How to get Column Prompt names for Analyses?

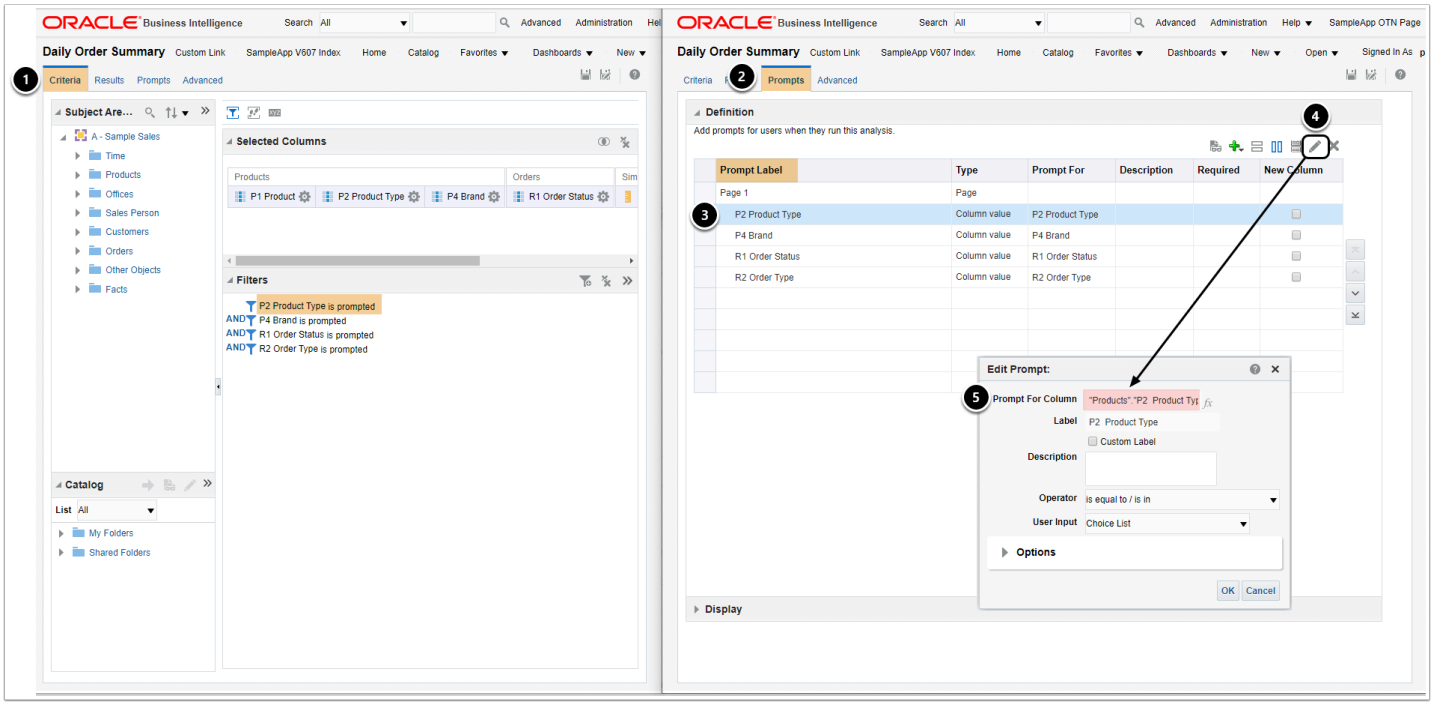
When mapping OBIEE Column Prompts to Filters in Metric Insights, make sure to use exact Prompt Names. Otherwise, prefiltering will not work.

For more information on Column Prompts, refer to:

1. **OBIEE – Prompts**
2. **Prompting in Dashboards and Analyses**

PREREQUISITES: Set up your Column Prompts

- To be able to fetch Prompt Filters to Metric Insights, you must first configure Column Prompts both on the **Criteria tab** and the **Prompts tab**.



In the Analysis *edit mode*:

1. On the ***Criteria tab***, make sure the required Column Filter **is prompted**
2. On the ***Prompts tab***, the prompted Column Filter must be added to the **Prompt Label** section
3. Select a **Column Prompt** whose name you need to copy
4. Click the **Edit icon** to open the **Edit Prompt pop-up**
5. Copy the **Column Prompt name** from the corresponding field
   - **NOTE:** *In Metric Insights UI, the Prompt name should be used without the double quotation marks*

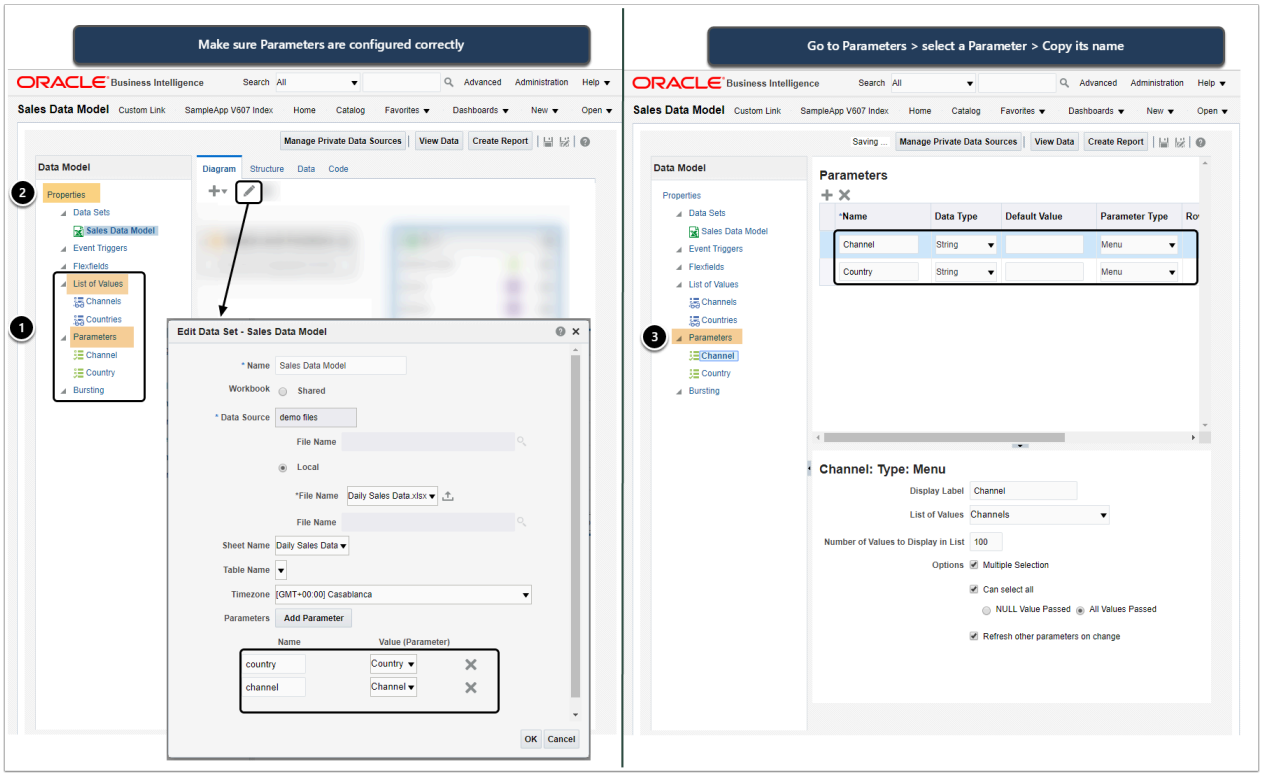
How to get Parameter names for Reports?

When mapping OBIEE Data Model Parameters to Filters in Metric Insights, make sure to use exact Parameter Names. Otherwise, prefiltering will not work.

- For more information on OBIEE Parameters, refer to Adding Parameters and Lists of Values

- For general instructions on how to build Reports in Oracle Business Intelligence, see [Creating and Editing Reports]

PREREQUISITES: Configure Parameters and Lists of Values in the Data Model *(Data Model is an OBIEE Report Component used at Report generation. Data Models contain sets of instructions for structured data retrieval)*.

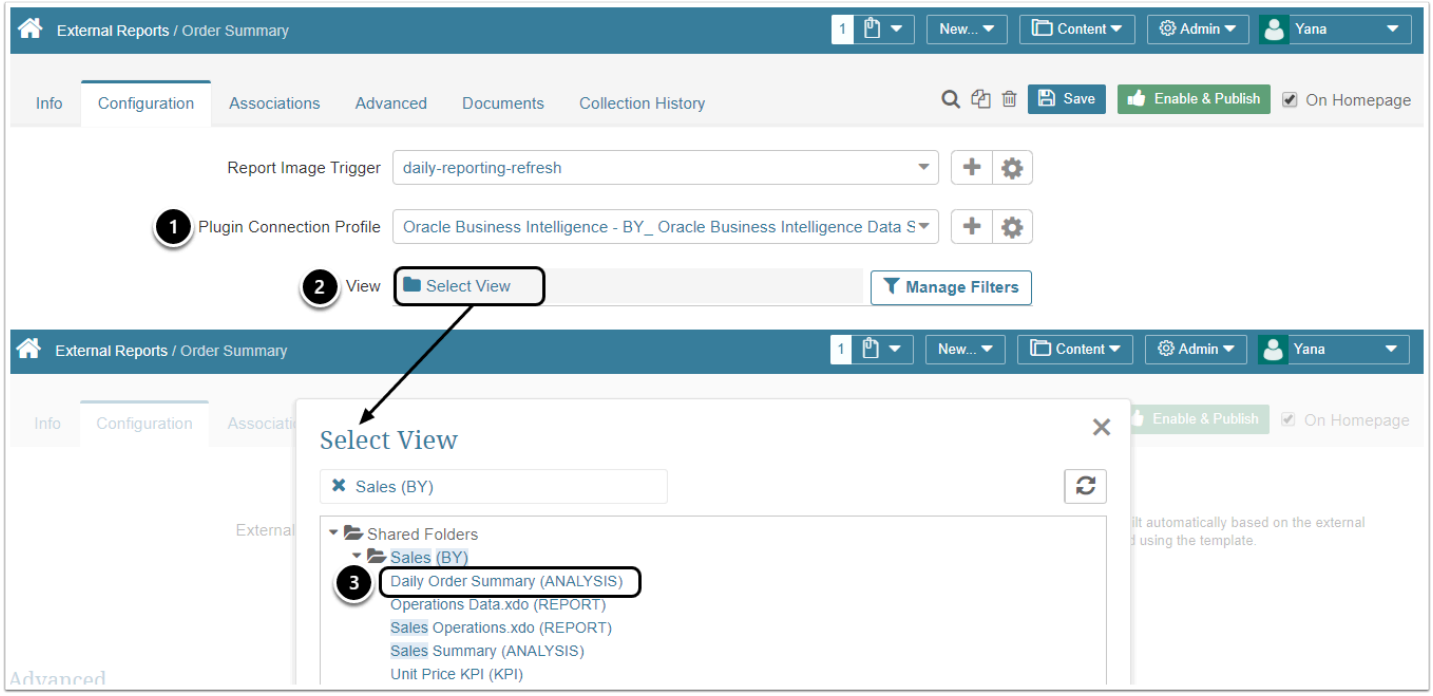


To verify that Data Model Parameters are configured properly, in the *Data Model Editor*:

1. Make sure that **Parameters** and the associated **Lists of Values** have been added
2. Go to Data Model **Properties** > **Data Sets** > select **Data Set** > click the **Edit icon:**
   - Check that **Column Names** are mapped to **Parameter Values** in OBIEE
3. If all the Parameter settings in the Data Model are correct, go to the **Parameters section,** select the required Parameter and copy its Name

1. Define a Source Object for an External Report

By specifying an OBIEE source element, Users will be able to fetch Filters applied to that element.

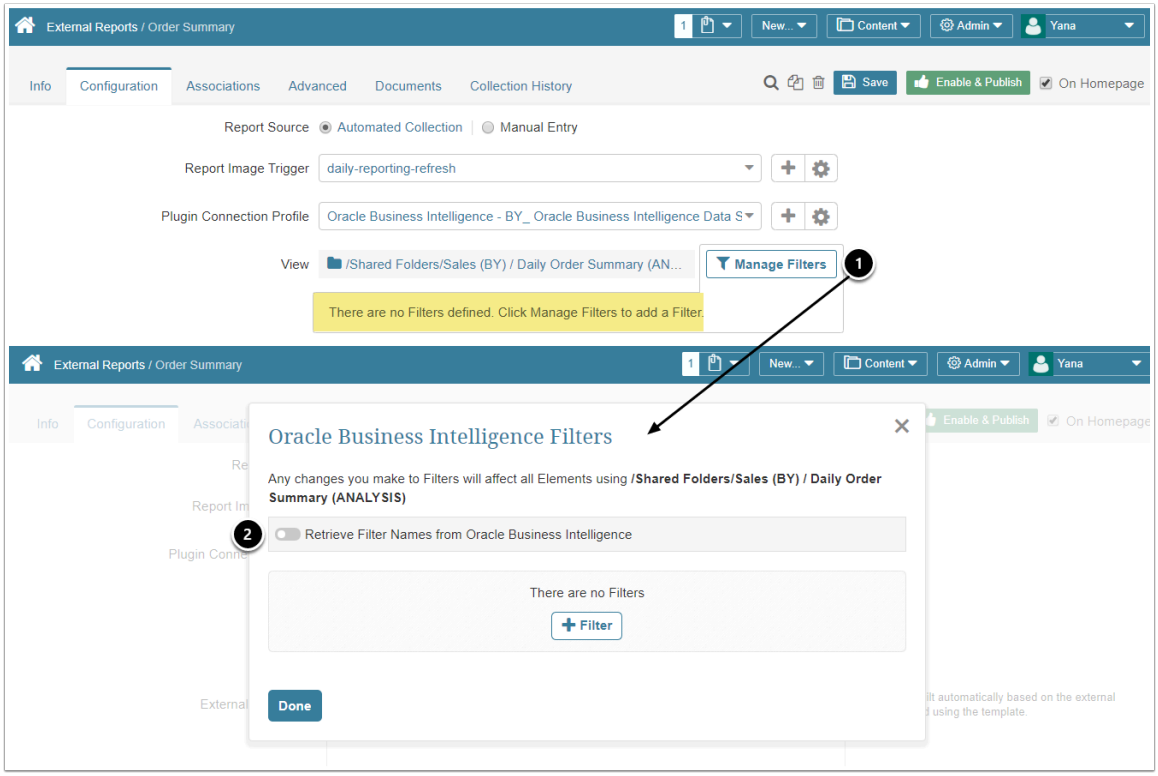


In the ***External Report Editor > Configuration tab***:

1. Select a **Plugin Connection Profile** that will be used for data collection
   - *For details on how to create a Plugin Connection Profile, refer to [Establish Connectivity to Oracle Business Intelligence]*
2. Click **[Select View]** to access the list of available OBIEE Objects
3. Click **[View Name]** for the OBIEE View to be selected as a data source in Metric Insights
4. If you do not see the required item, use **Refresh**

2. Add OBIEE Filters to Metric Insights

The **Filter Management** option allows Users to add Filters and access the related functionality.



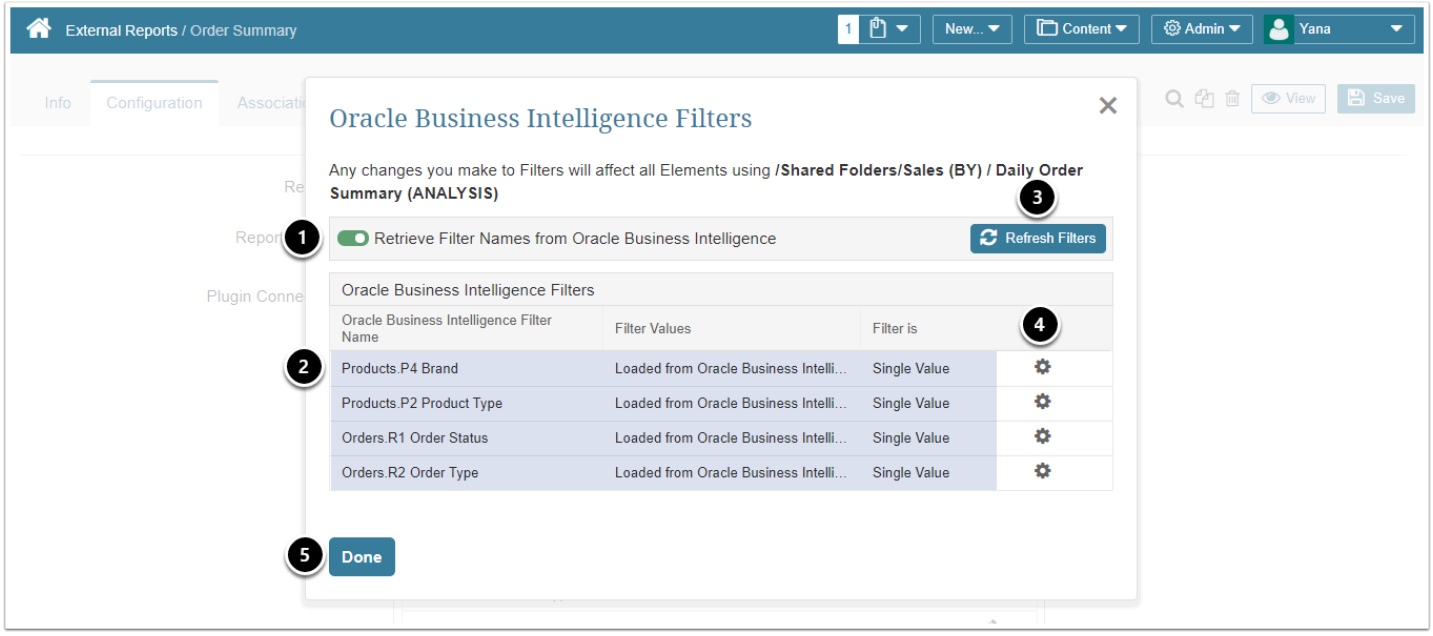
To be able to add Filters:

1. Click **[Manage Filters]**
2. For **auto-retrieval and complete Filter syncing**, activate the toggle (see details in Step 2.1)
3. For **other Filter-adding methods**, click **[+Filter]** (see details in Step 2.2, Step 2.3, Step 2.4 and Step 2.5)

2.1. Retrieve Filter Names from OBIEE

Selecting this option means that all Filters and Filter Values will automatically be fetched from Oracle Business Intelligence.

⚠ Filters added automatically cannot be deleted if the **"Retrieve Filter Names from Oracle Business Intelligence"** option is activated.

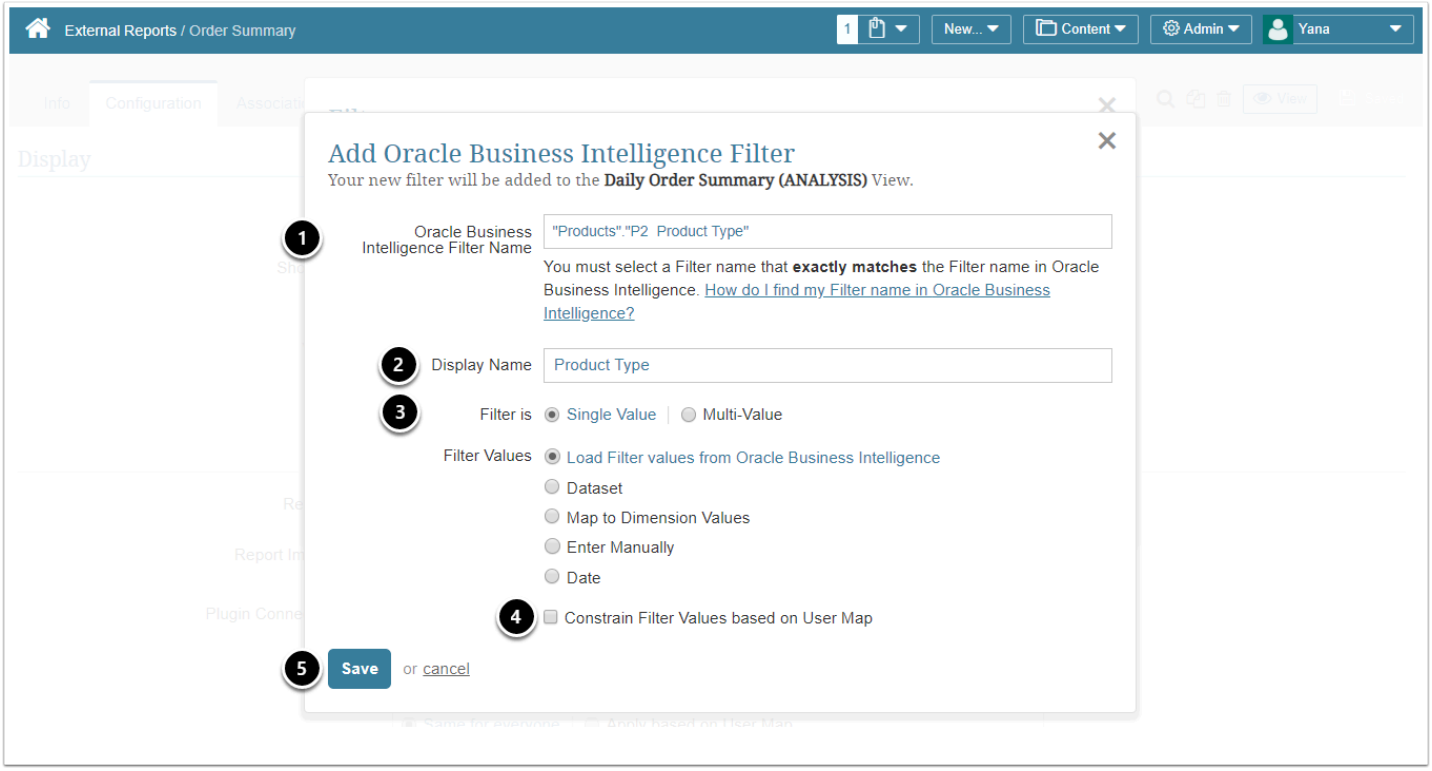


To enable auto-retrieval:

1. Activate the **Auto-Retrieval** option
2. The loaded **OBIEE Filters** will appear in a list below
3. **Refresh** the Filters' list as needed
4. To edit a Filter, use the **Edit (Gear) icon**
    - *For details refer to [Edit Filter Properties]*
5. Click **[Done]** to proceed

2.2. Load Filter Values from OBIEE

This option gives more control over which Filter Values to load to Metric Insights. Filters are added one by one, enabling Users to determine how many Filters will be fetched.

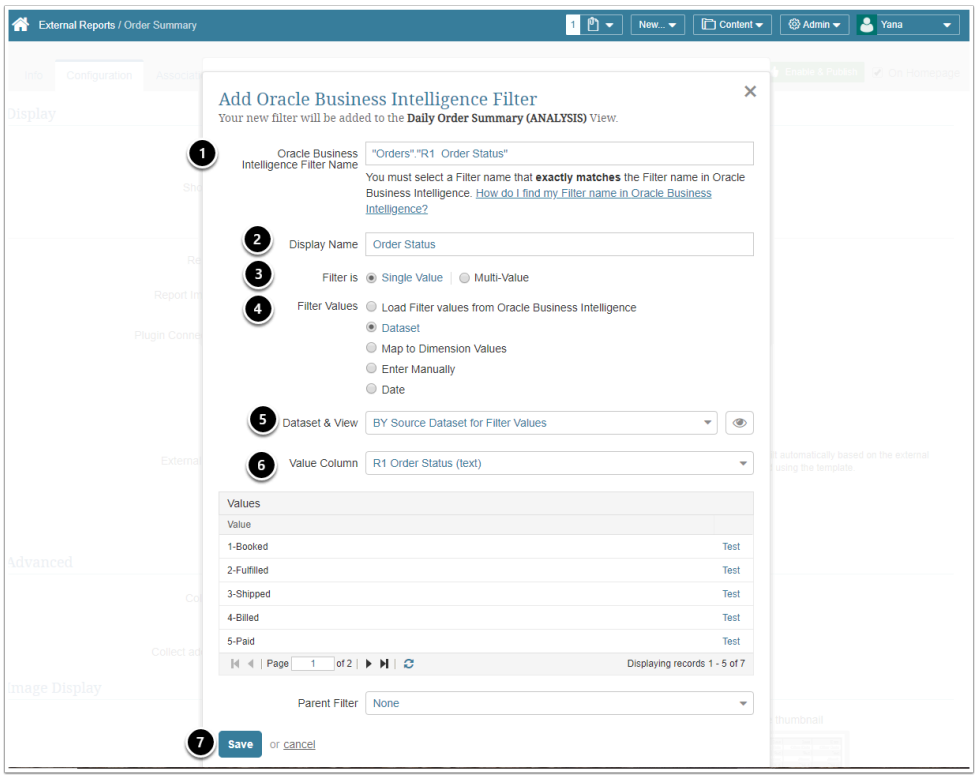


To automatically load a Filter:

1. Specify the **name of the OBIEE Filter** *(Prompt or Parameter)* that needs to be loaded
2. Optionally, specify a **Display Name** to override the original Name of an OBIEE Filter
3. Select the Type of Filter:
   - **Single Value** allows choosing one Filter Value in the Viewer
   - **Multi-Value** allows choosing several Filter Values simultaneously and showing visualizations for the selected Values
4. Optionally, restrict certain Filter Values to specific Users with the **"Constrain via User Map"** setting
5. **Save** your entries

2.3. Load Filter Values from a Dataset

Loading Filter Values from a Dataset involves using a selected Dataset column as a source of Filter Values and mapping it to a specified OBIEE Filter Name.



To use a Dataset as a source of Filter Values:

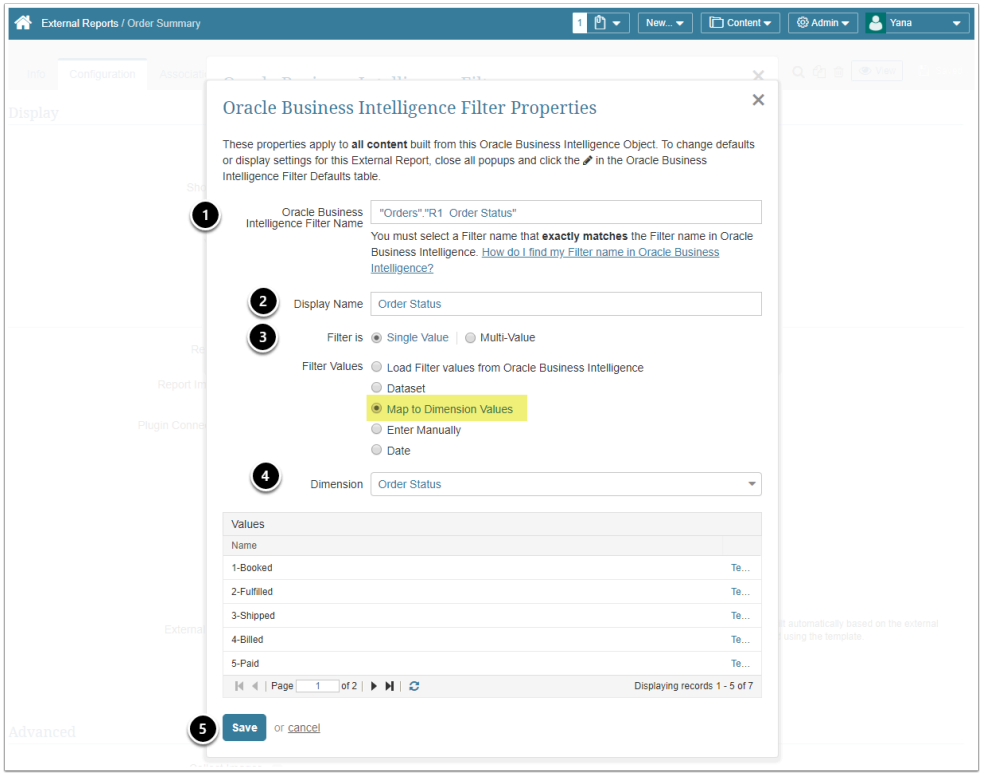
1. Input the **name of the OBIEE Filter** *(Prompt or Parameter)*
2. Optionally, specify a **Display Name** to override the original Name of an OBIEE Filter
3. Select the Type of Filter:
   1. **Single Value** allows choosing one Filter Value in Viewer
   2. **Multi-Value** allows choosing several Filter Values simultaneously and showing visualizations for the selected Values
4. Choose **Dataset** as a source of Filter Values
5. Specify **Dataset & View** from which Values will be loaded
6. Select a Dataset **Column** that will be mapped to the specified Qlik Sense Filter
7. **Save** your entries

2.4. Map to Dimension Values in Metric Insights

OBIEE Filters can also be mapped to Dimensions in Metric Insights.

PREREQUISITES:

- A Dimension must be configured in Metric Insights

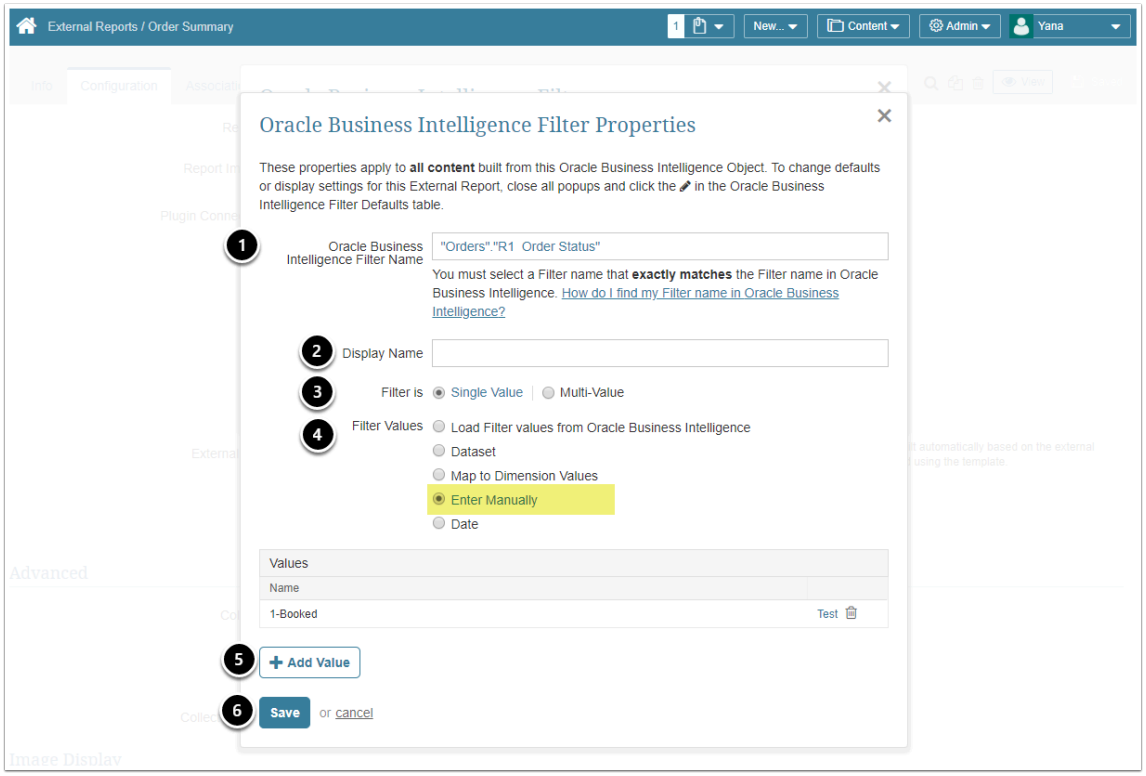


To map an OBIEE Filter to a preconfigured Metric Insights' Dimension:

1. Input the **name of the OBIEE Filter** *(Prompt or Parameter)*
2. Optionally, specify a **Display Name** to override the original Name of an OBIEE Filter
3. Select the Type of Filter:
   - **Single Value** allows choosing one Filter Value in Viewer
   - **Multi-Value** allows choosing several Filter Values simultaneously and showing visualizations for the selected Values
4. **Filter Values:** choose "Map to Dimension Values"
5. Select a **Dimension** whose Values will automatically be loaded to the Values list
6. **Save** your entries

2.5. Enter Filter Values Manually

Using the manual setting, Users have full control over which Filters and Filter Values are added to an Object/Element in Metric Insights.

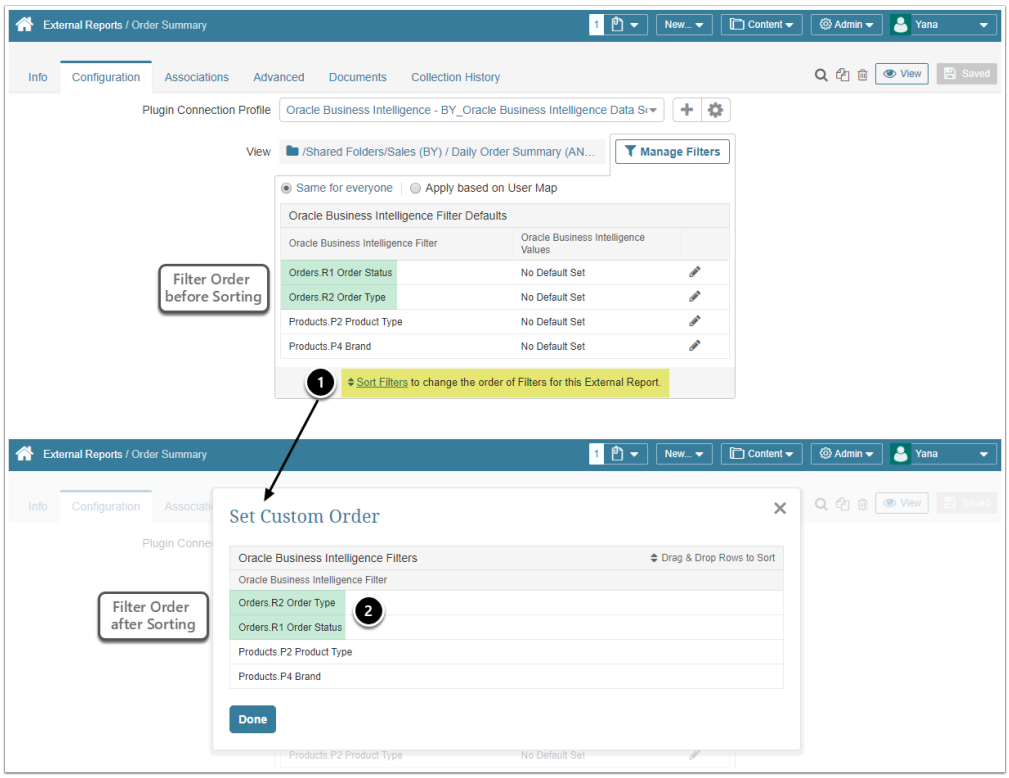


To enter Filter Values manually:

1. Input the **name of the OBIEE Filter** *(Prompt or Parameter)*
2. **Display Name** will allow you to override the original Filter name; this Name will be used in Metric Insights
3. Select the Type of Filter:
   - **Single Value** setting will allow you to choose only one Filter Value in Viewer
   - **Multi-Value** setting enables the display of data in the External Report Viewer for several Filter Values at once
4. **Filter Values:** choose "Enter Manually"
5. Click **[+Add Value]** to add Values by hand
6. **Save** your entries

3. Customize the Filter Order with Sorting

You can specify the order in which the Filters will be displayed in the External Report Viewer.

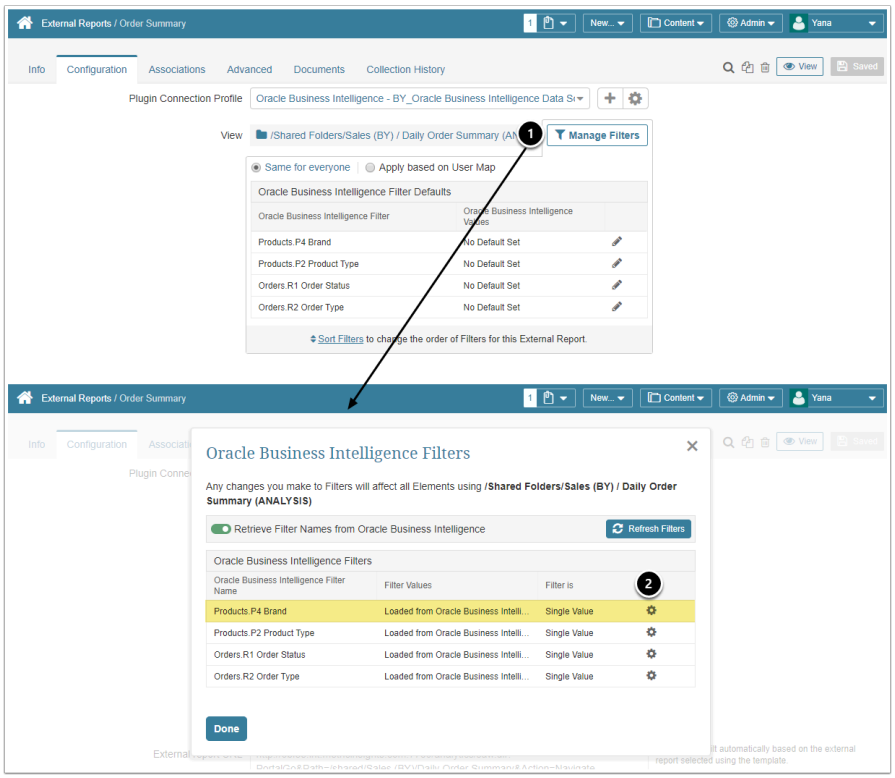


To set custom order in which the Filters will be displayed in Viewer:

1. Click **[Sort Filters]**
2. **Drag & Drop** rows to sort

4. Edit Filter Properties

Having added the Filters, Users can make custom changes to their settings.



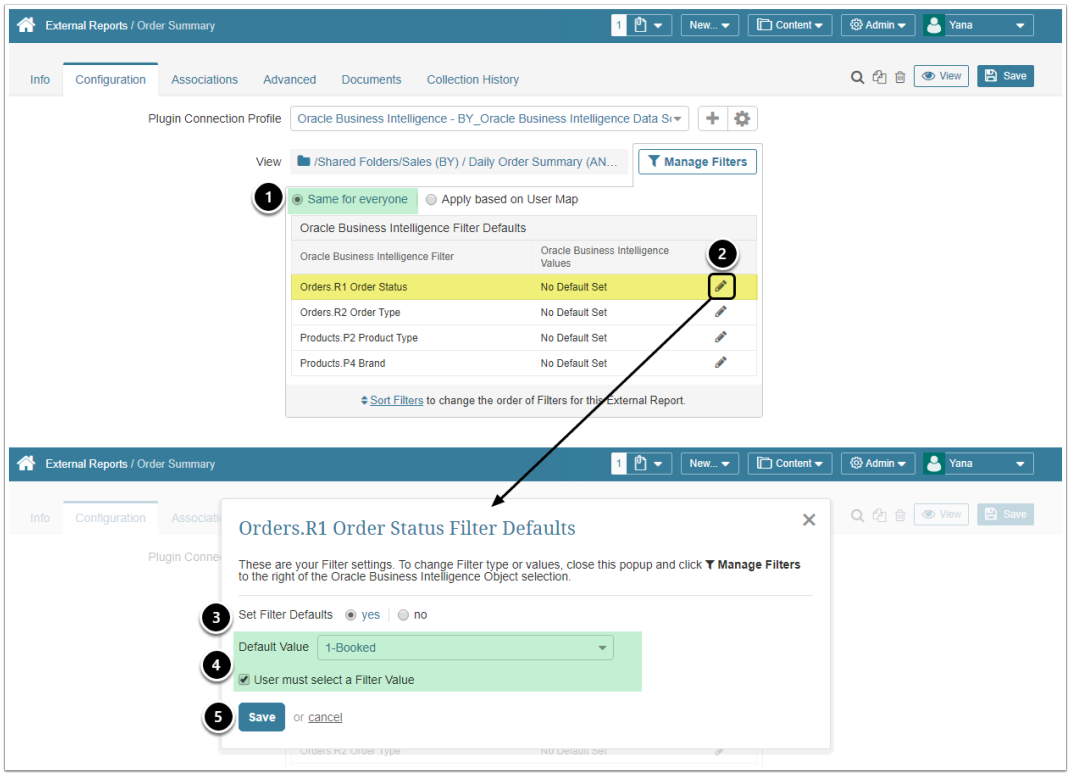
To edit a Filter:

1. Click **[Manage Filters]**
2. Choose the Filter that needs changing and click the **Edit (Gear)** icon

5. Set Filter Defaults

While configuring Filters, it is possible to apply default settings that are the same of everyone, or customize them with a User Map.

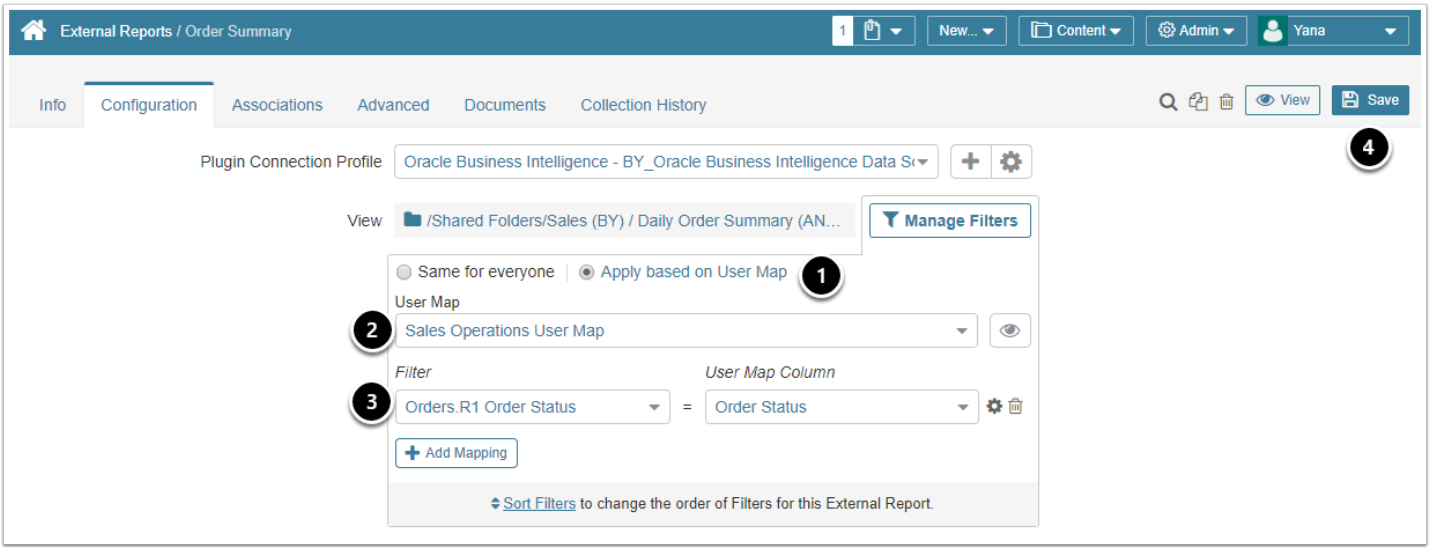
5.1. Configuring shared Defaults



To set shared Defaults:

1. Select **"Same for everyone"**
2. Click the Filter **Edit** (Pencil) icon
3. In the pop-up, **Set Filter Defaults** to "yes"
4. Specify the required **defaults**
5. **Save** your entries

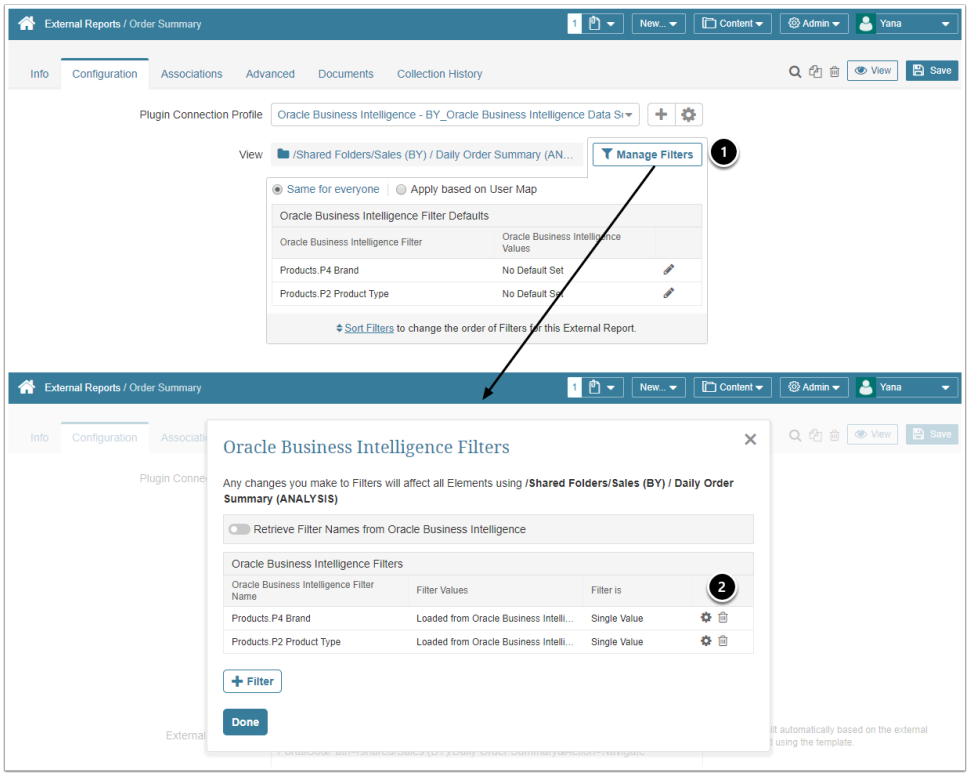
5.2. Personalizing Defaults



To set custom Defaults:

1. Select **"Apply based on User Map"**
2. Select a **preconfigured User Map**
3. Map **Filters** to **User Map Columns**
4. **Save**

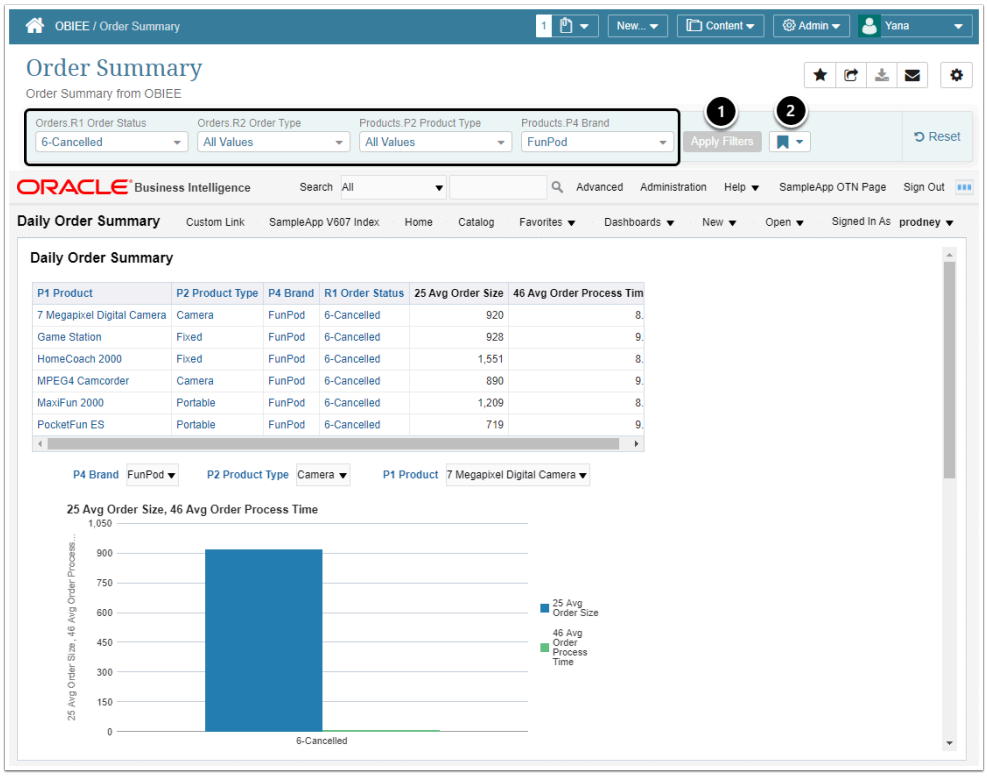
6. Delete Filters



To delete some of the added Filters:

1. Click **[Manage Filters]** next to the name of an OBIEE Object (View)
2. In the pop-up window, use the **Trashbin** icon in the respective row

7. Verify the display in the Report Viewer



1. Select the required Values and click **[Apply Filters]** to see the refined data
2. Optionally, save your Filter Settings as **Bookmarks**
   - For more information, refer to Setting Personal Bookmarks (External Reports)

What's next?

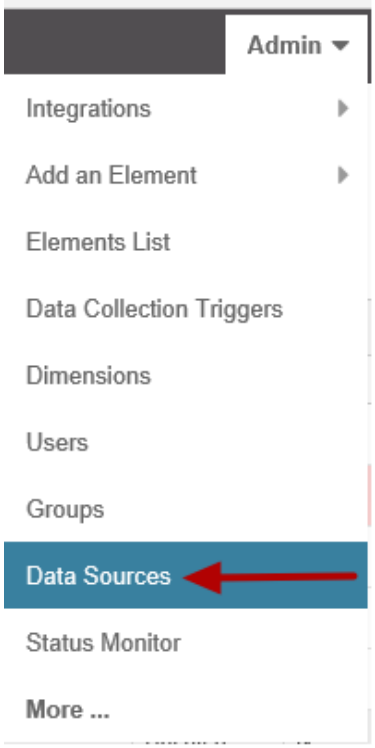
Setting Personal Bookmarks

26. Sourcing Data using OLAP

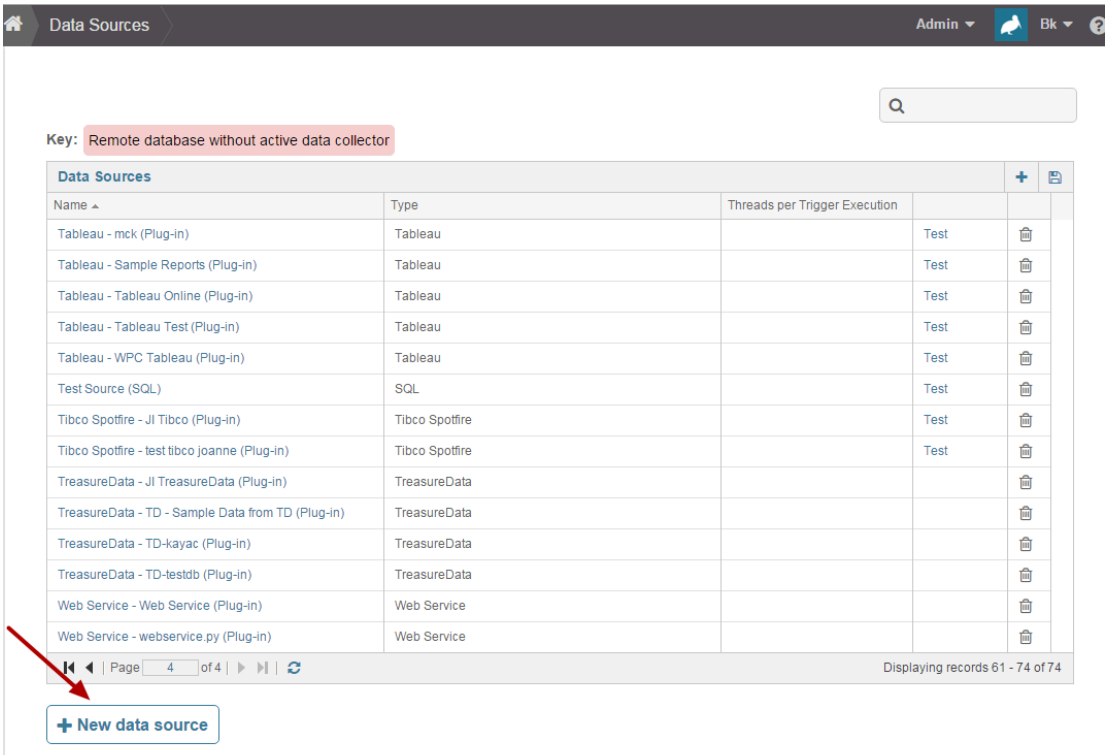
26.1 Establish Connectivity to OLAP

An Administrator can use the process described in this article to create a new **Plug-in Data Source.** It is required to allow Elements to fetch data using **OLAP**.

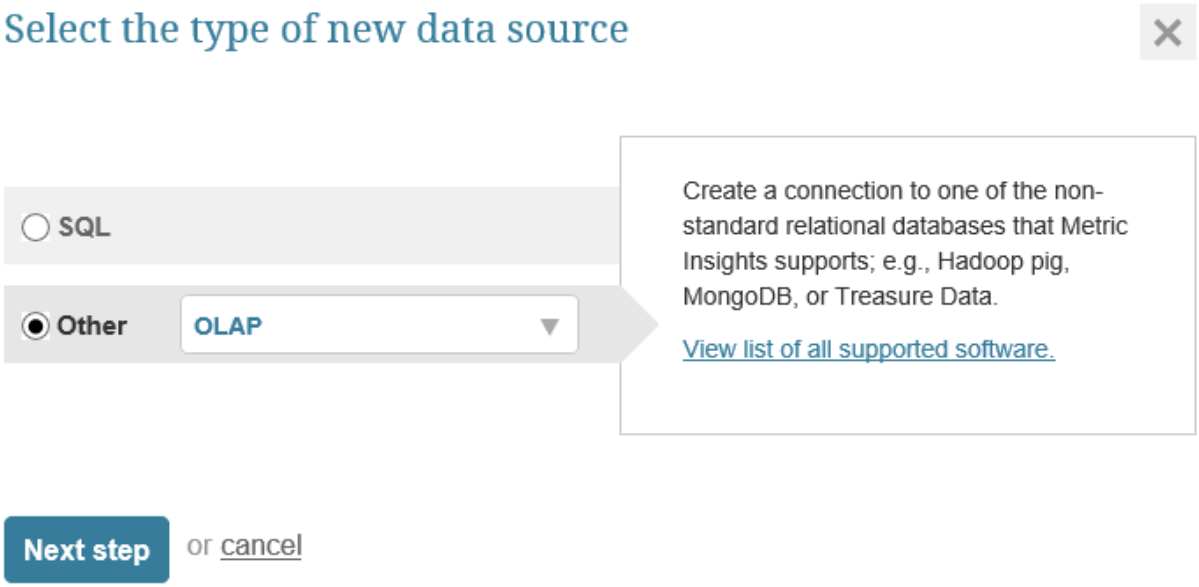
1. Select Data Sources from Admin drop-down



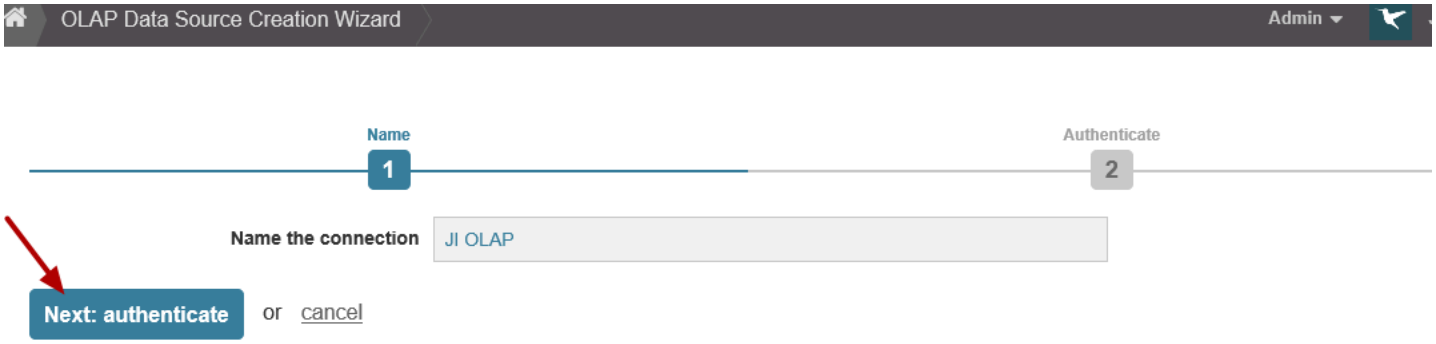
2. Add New Data Source



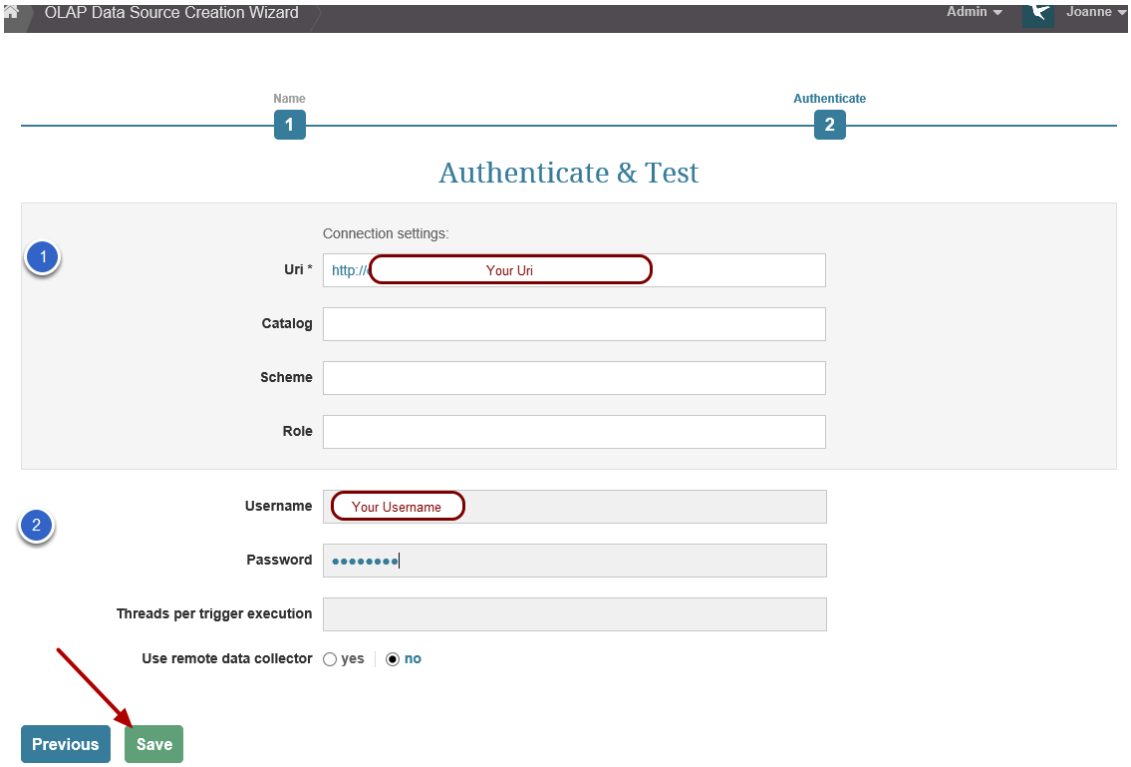
3. Select "Other" Data Source Type and choose "OLAP" from the drop-down



4. Input a meaningful Name and next step



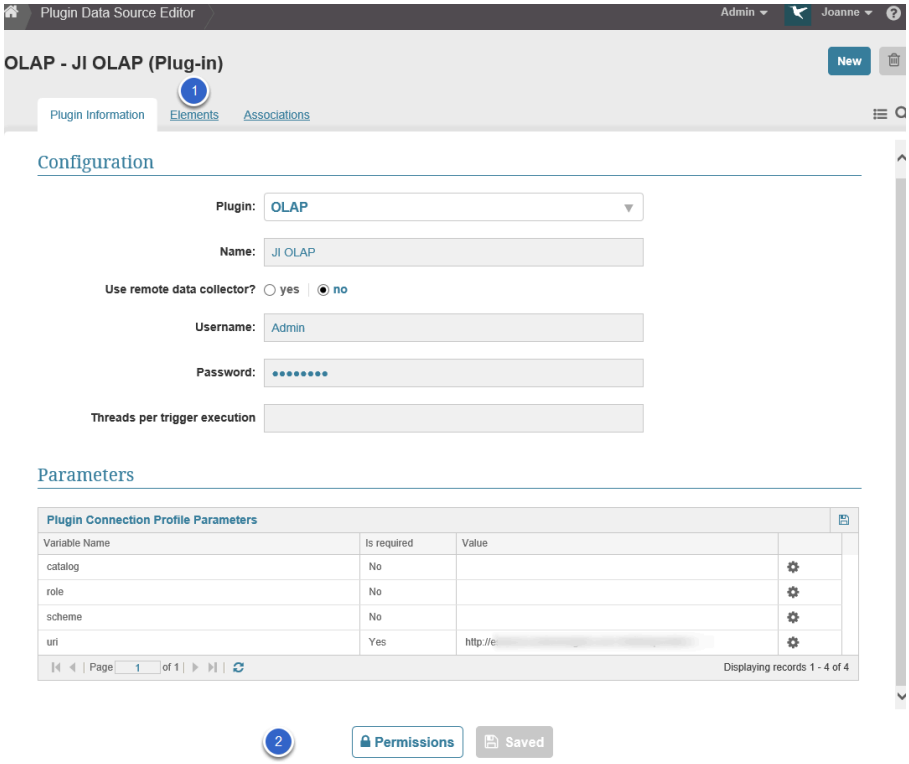
5. Authenticate and Test



1. Input **Uri**
2. Enter your **Username** and **Password**

Save

6. Full Data Source Editor displays

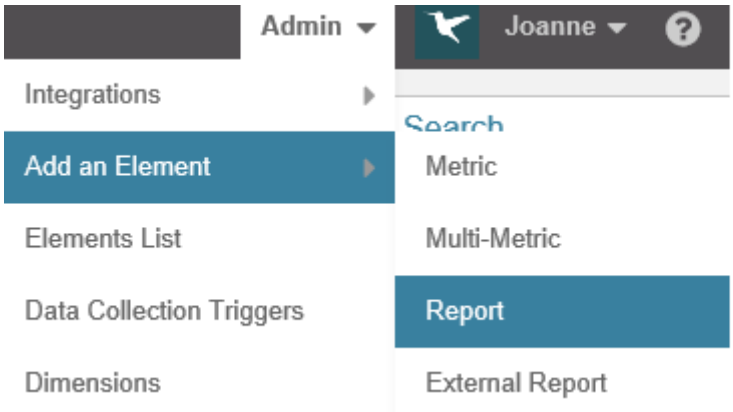


1. You can create **elements** directly from the Elements tab
2. You can assign **permissions** to Groups or Power Users here also

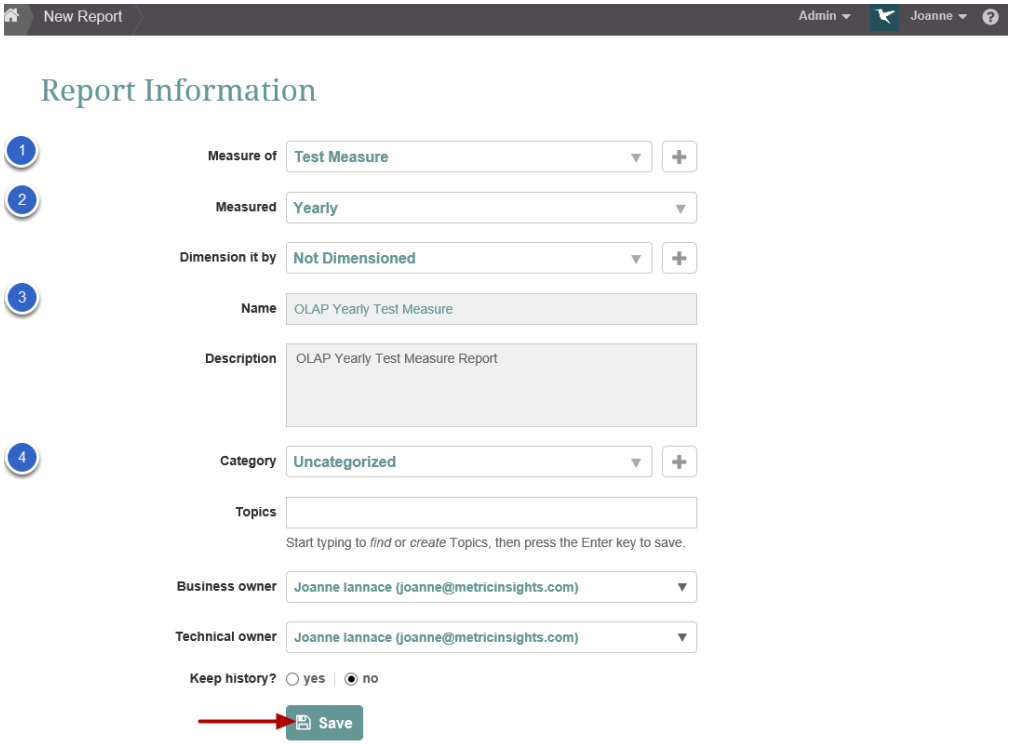
26.2 How to Collect Data from OLAP

This article will show you how to create an Element (Report in this case) using an OLAP plug-in as a Data Source. It assumes that you have already established connectivity to OLAP.

1. Add a new element based on your Salesforce plug-in data source



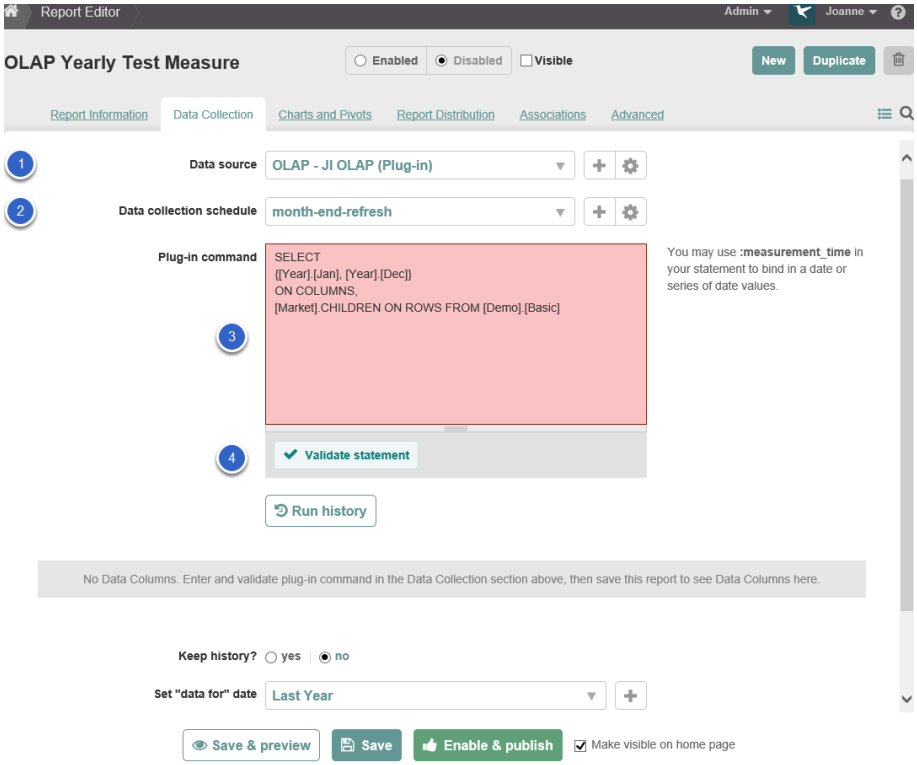
2. Provide basic information on Wizard (or Editor) - report example



1. Specify what this report is **measuring**. If you do not see the measure that you want to use, you can create one from this drop-down
2. Select the **Measurement Interval** that applies to your element
3. Give the element a unique **name**
4. Optionally, assign a **Category**

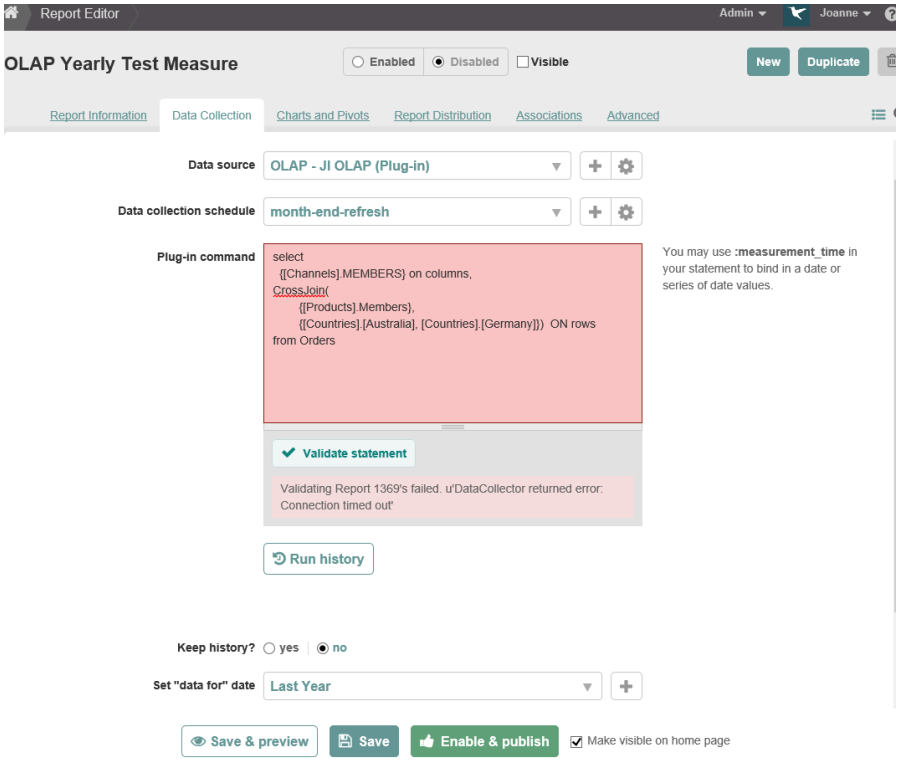
Next: (example is using full editor, but same steps apply when using Wizard)

3. Full Editor displays the Data Collection tab



1. Select **OLAP** plug-in in **Data Source** drop-down
2. Set **Data collection schedule**Select a **Report** from drop-down list
3. Input Plug-in Command
4. **Validate statement**

Connection times out



27. Sourcing Data from QlikView

27.1 QlikView Overview

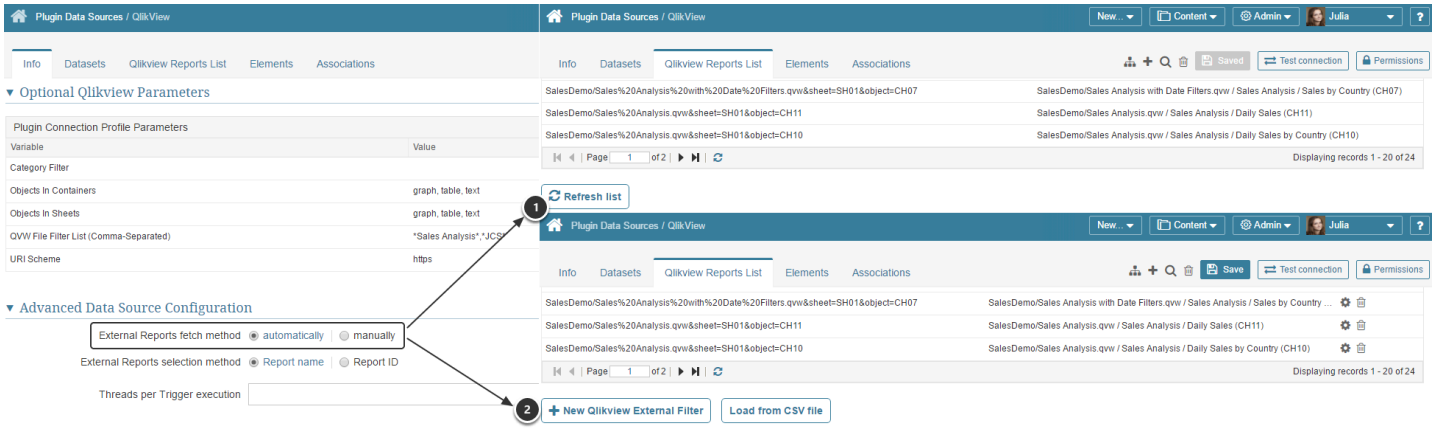
QlikView can be easily integrated with Metric Insights to join forces and provide better solutions for analyzing data and supporting business decisions.

[Release 6.x]: 'Remote Data Collector' renamed to 'Remote Data Processor'

Capabilities

Here is a list of QlikView plugin capabilities in Metric Insights:

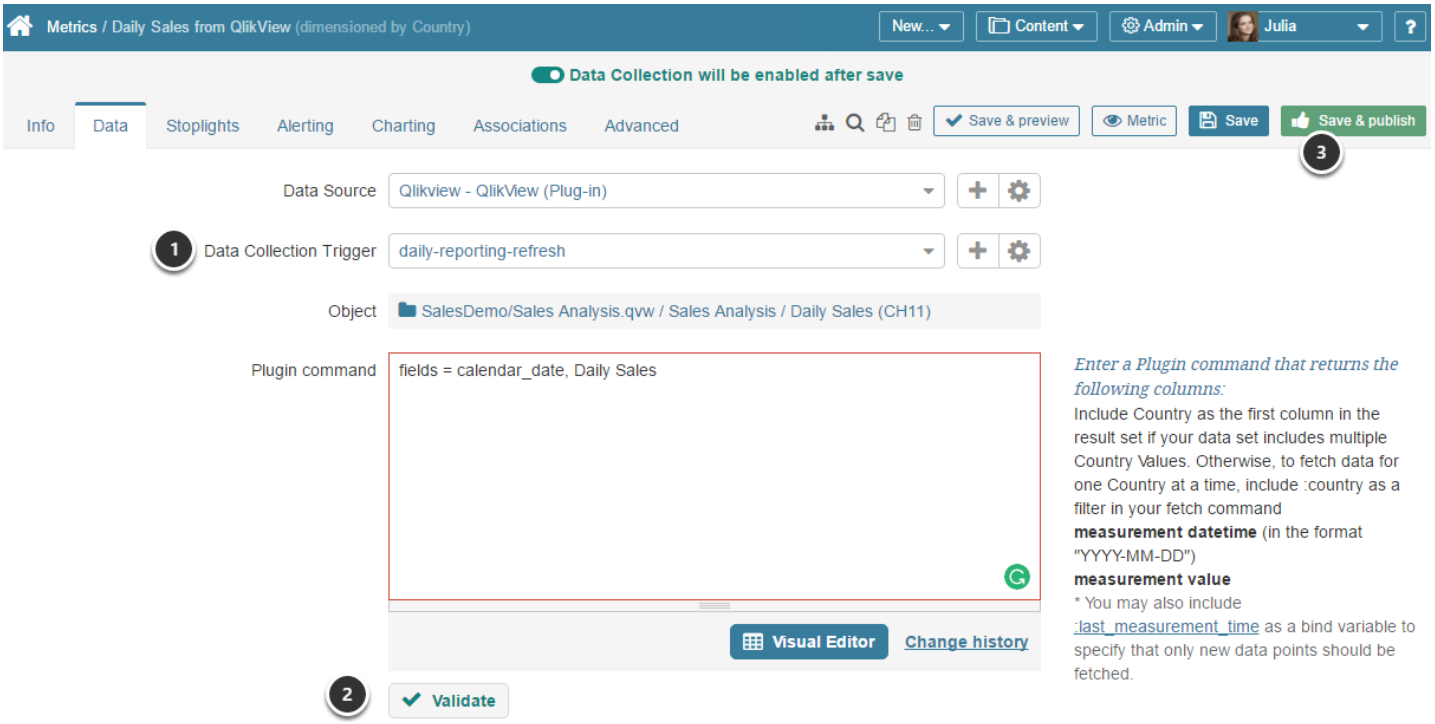
Fetching QlikView Reports (manually and automatically)



Metric Insights extracts data from the QlikView server in the form of reports. Information obtained from these external reports is further used as data source for MI elements: Metrics and Reports. QlikView reports can be added in the *Plugin Editor* (**NOTE:** To get to the *Plugin Editor* go to **Admin** drop down menu > **Data Sources** > select a plugin to open its Editor). There are two options to add reports to the plugin:

1. **Automatically**: If the **External Reports fetch method** field is set to 'automatically', go to the QlikView *Reports List* tab and simply click **Refresh list** to collect all QlikView reports currently available at the server.
2. **Manually**: You can also update the QlikView reports list by adding report IDs and Names one-by-one or specified in the CSV file.

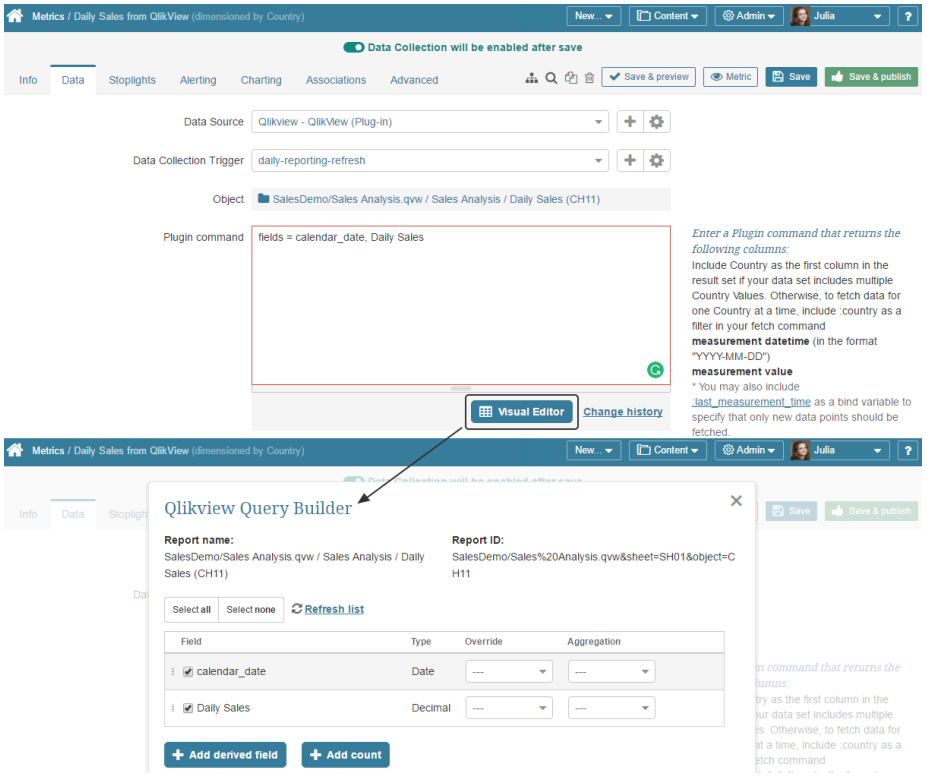
Automated Data Collection



All the data fetched from QlikView is always extracted automatically and this can be performed via:

1. **Data Collection Trigger**: You can configure the periodic trigger that will independently start data collection
2. **Validate** option: You can start automatic data collection by validating your plugin command
3. **Save & Publish** OR **Update Live Chart** options: An attempt to update a chart always starts automatic data collection

Visual Editor for constructing Plugin command



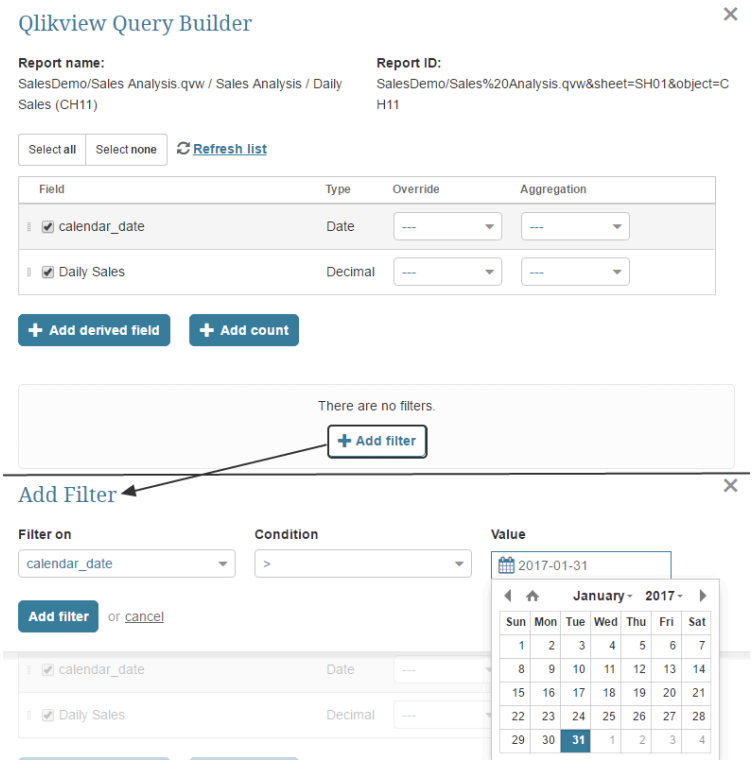
Data for Metrics and Reports is extracted via a plugin command, consisting of parameters specifying what data should be fetched. See the list of available parameters.

The check-list with fields available for constructing a plugin command is automatically captured from the reports and displayed at the QlikView **Visual Editor** located at *Metric Editor* > *Data Collection* tab > below the **Plugin command** text box.

Visual Editor is a handy tool. It saves time of those who are unaware of specific syntax of certain plugins and eliminates the likelihood of mistakes often made when typing the command manually.

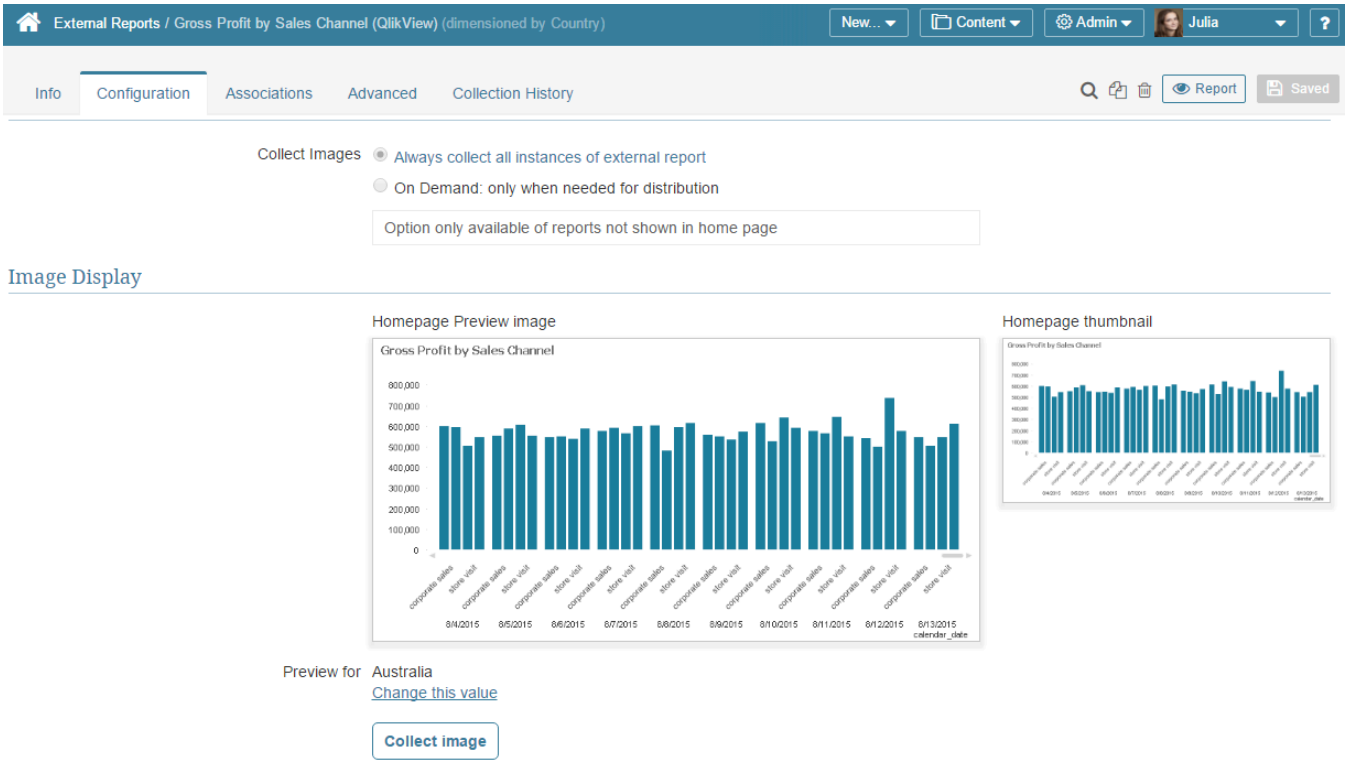
The command is automatically constructed from the selected fields and values defined from the drop-down lists consisting of all available options.

Pre-filtering of fetched data is supported in the Visual Editor



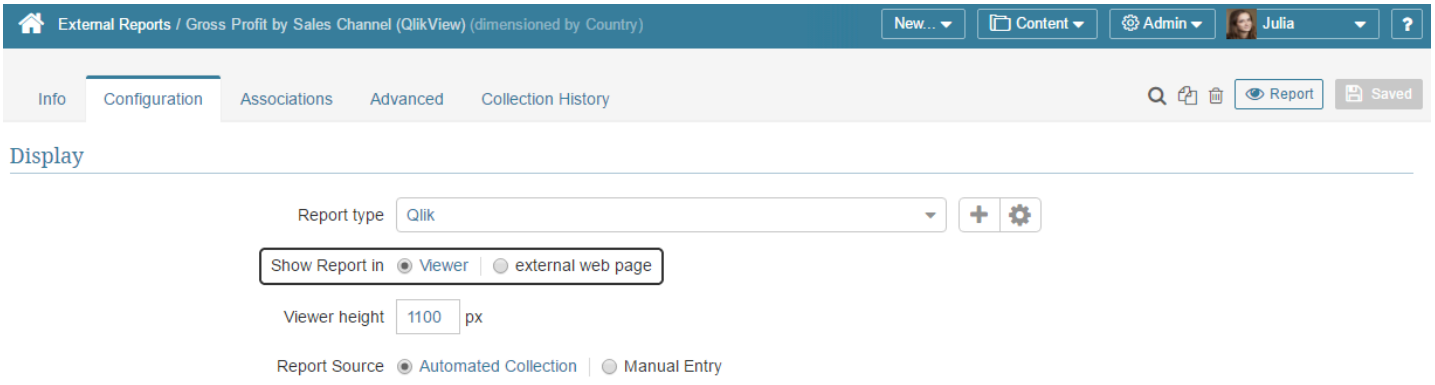
Data pre-filtering can be configured from the **QlikView Query Builder** pop-up screen (**Visual Editor**).

Image fetching



External reports can be created by the means of fetching an image directly from QlikView.

Live visualizations



QlikView reports can be either embedded directly into the Viewer as Live Visualizations or can serve as links to a specified external web pages.

Requirements

- Remote data collector

- Other requirements for establishing connection with QlikView server are listed in this article

Available Help Docs

- Download Instructions for PoC
- Establish Connectivity to QlikView
- How to Collect Data using QlikView Plug-in
- How to create an External Report from QlikView

27.2 Download Instructions for PoC

This article describes how you should prepare for Meric Insights PoC if you also use Qlikview.

What to Expect in our Proof of Concept (PoC)

1. Prepare the Qlikview QVW file

1. Make sure your QVW is loaded with data and **Save**
2. Copy the .qvw file. This file is usually located at *C:\ProgramData\QlikTech\Documents*

We need an unsecured version of the QVW, so if 'Section Access' is included in your QVW:

1. Create a copy of your QVW first
2. Open the copy in **Qlikview Desktop**
3. Go to *File* > *Edit Script*
4. Locate the **Section Access** block in your load script and either remove or comment out

   NOTE: Section Access may be implemented in a hidden script

- File - Edit Script
- File - Edit Hidden Script (in the Script Editor)
- Username & Password will be required

5. Reload the QVW at *File* > *Reload*

2. Upload the QVW to Metric Insights

1. Go to: ftp bob.metricinsights.com

2. Provide the username and password you've been supplied with

3. Open the "upload" directory. This is the only directory you're allowed to upload to on the ftp site:

```
cd upload
```

4. Change the working directory:

```
lcd /local/directory
```

5. Store a file by executing this command:

```
put local_file
```

On the receiving end, files go here: /var/ftp/demo/upload/

[OPTION 1] Command line (Mac Terminal, Windows PuTTY)

1. Go to: ftp bob.metricinsights.com
2. Provide the username and password you've been supplied with
3. Open the "upload" directory. This is the only directory you're allowed to upload to on the ftp site:

```
cd upload
```

4. Change the working directory:

```
lcd /local/directory
```

5. Store a file by executing this command:

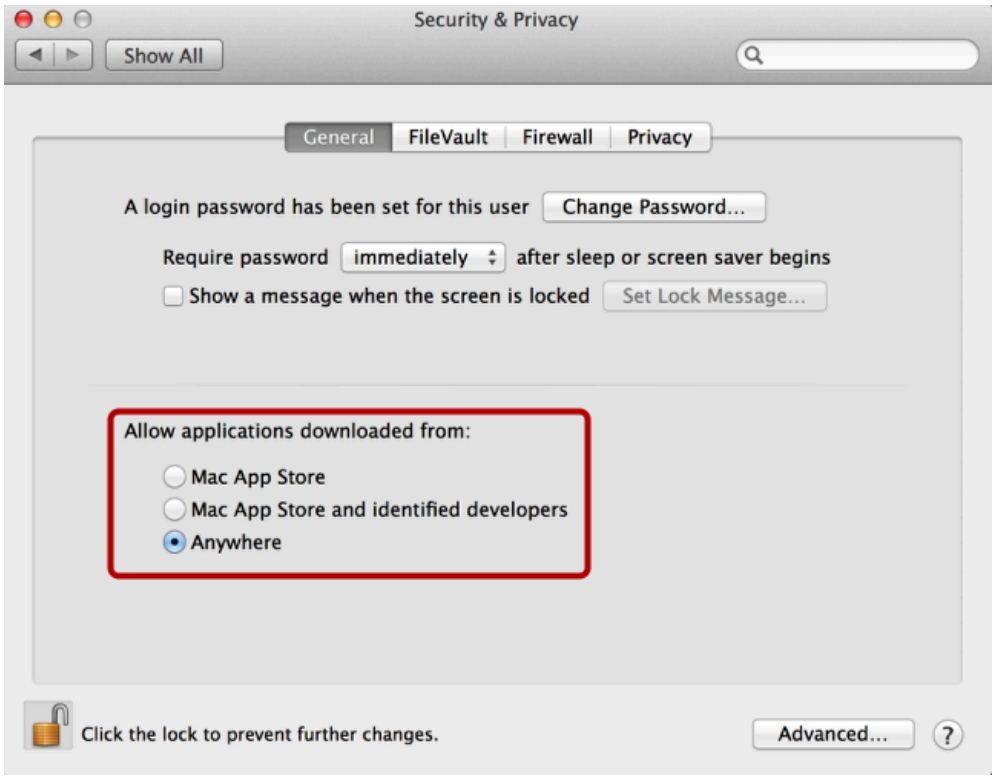
```
put local_file
```

On the receiving end, files go here: /var/ftp/demo/upload/

[OPTION 2] Filezilla

1. Download Filezilla from https://filezilla-project.org/
2. Unzip the download file

MacOS NOTE: You might need to change your security settings at **Apple menu > System Preferences > Security & Privacy > General tab** in order to "Allow applications downloaded from: anywhere" (see screen below)



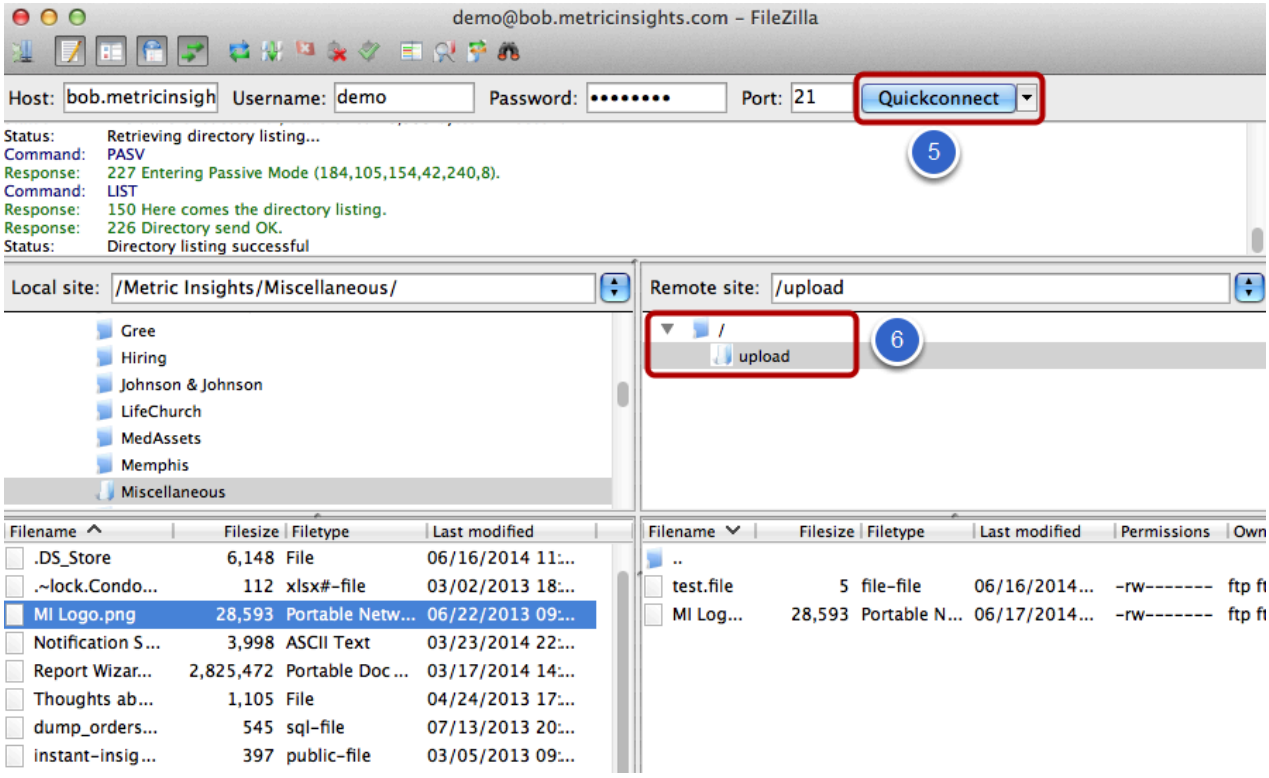
3. Start the Filezilla application.

4. Enter the following values at the top of the screen:

- **Host**: bob.metricinsights.com
- Provide the **Username** and **Password** you've been supplied with
- **Port**: 21 (or leave blank)

5. Click **Quickconnect** at the top right corner of the screen (see screen below)

6. Once the connection is established, you should see a directory tree on the right-hand side that includes the **upload** directory. Double-click it to open it. Then just drag your file from the left-hand side to the upload directory



[OPTION 3] Dropbox

We can also share a Dropbox folder for you to upload the QVW file.

27.3 Prerequisites to connecting to QlikView server

For QlikView you will need access to the QlikView server. Metric Insights requires you to install a small agent on the QlikView server. This agent communicates with QlikView via its API and returns data to Metric Insights.

A port (8443) will need to be opened for communication back from the agent on the QlikView server to Metric Insights.

The agent is a Java / .NET program that is a few MB in size. It is referred to as a Remote Data Collector and we have detailed instructions on how to install it. You will need to install JRE (Java Runtime Engine) and .NET on the machine that runs the Remote Data Collector.

Alternate Approach

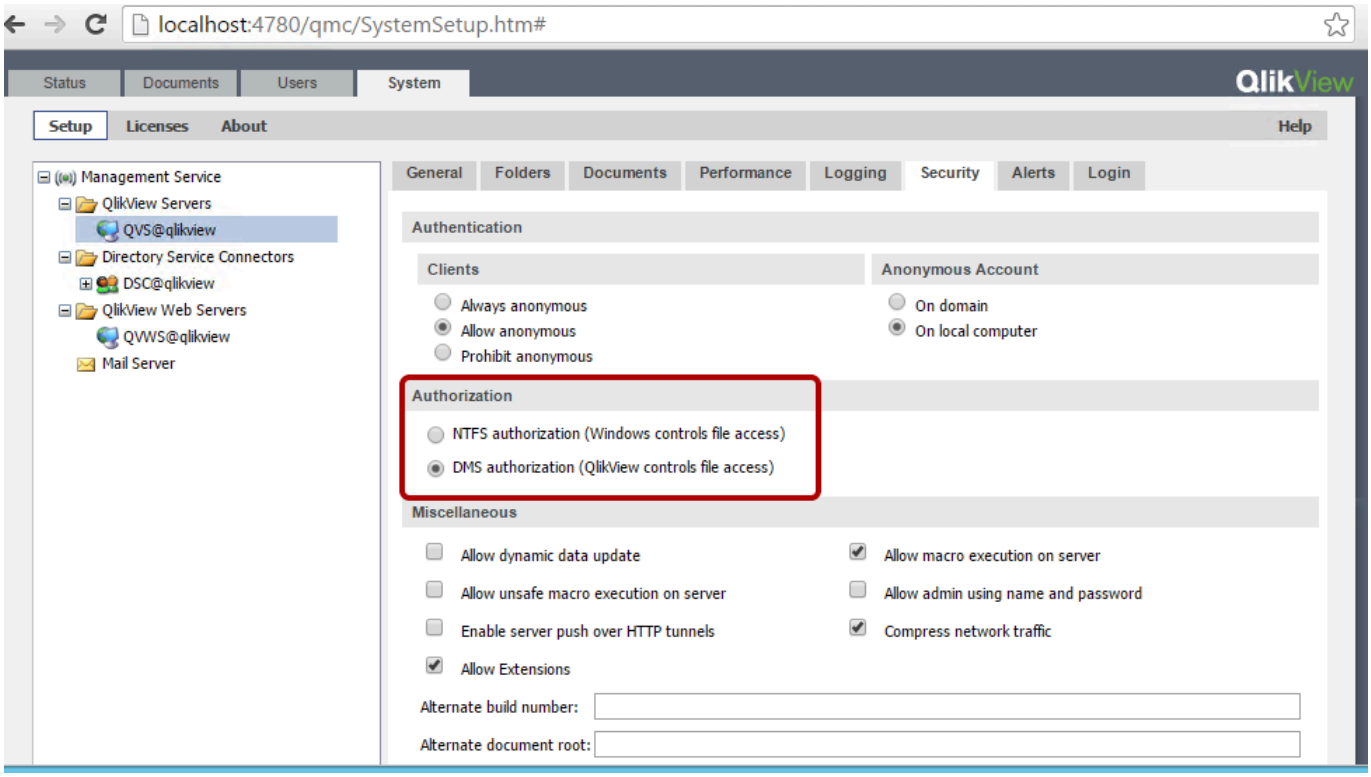
An alternate approach is to host the Metric Insights agent on another Windows server in the same network as QlikView. In this configuration the agent communicates with QlikView server over ports 80 and 443 (http/https), and also port 4747 (using "qvp" - the QlikView protocol) and port 4799.

A port (8443) will need to be opened for communication back from the agent on the Windows server to Metric Insights.

1. QlikView Authorization options

When installing QlikView Server, either QlikView Web Server (QVWS) or IIS Authorization can be used. Prerequisites per both options are described below.

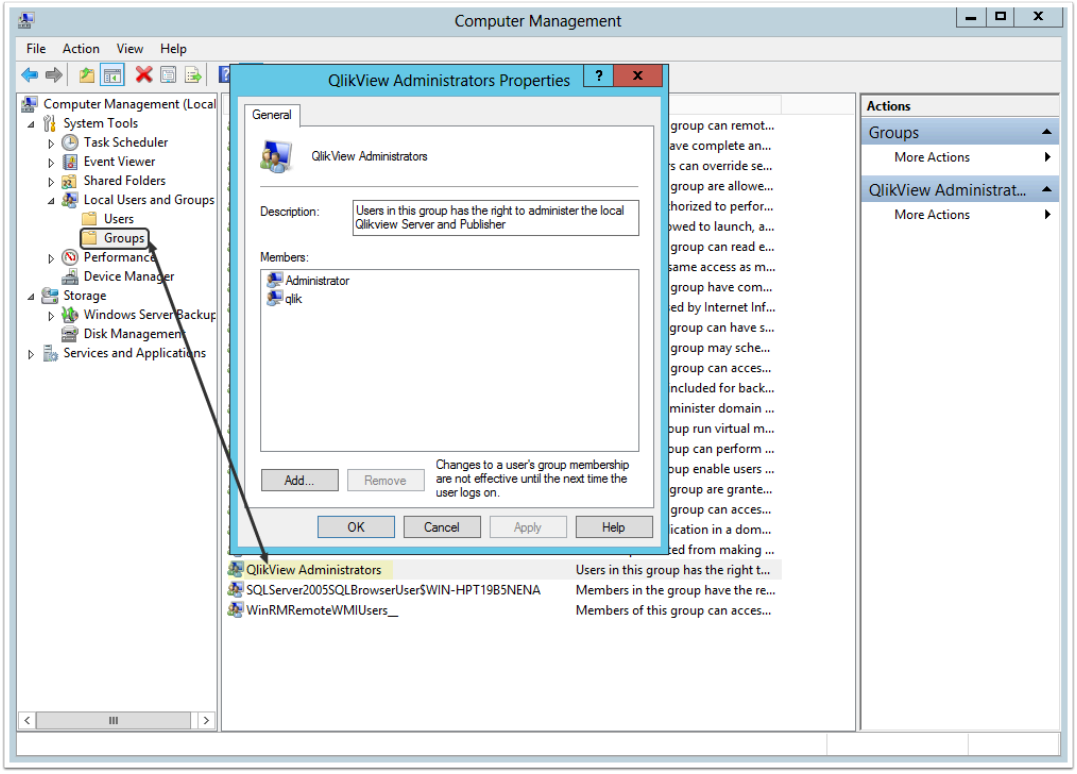
1.1. QVWS Authorization



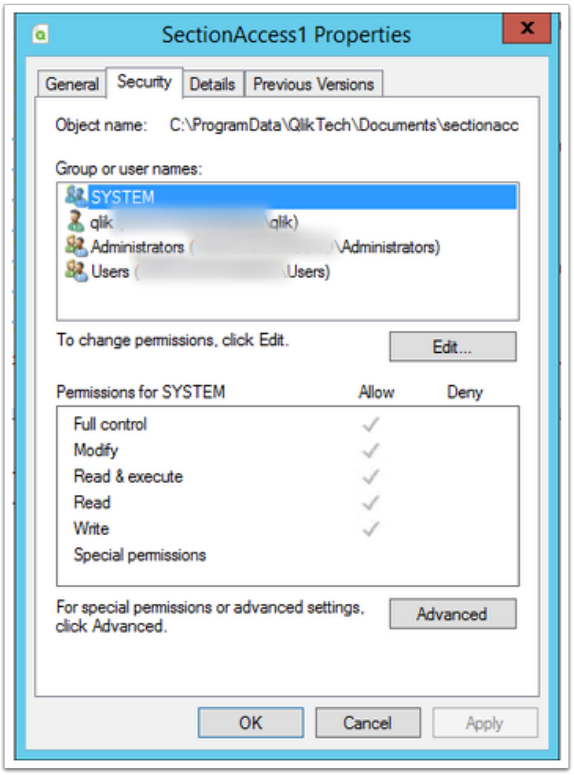
QlikView Web Server supports 2 types of authorization:

- **NTFS** (with Windows controlling file access)
- **DMS** (with QlikView controlling file access)

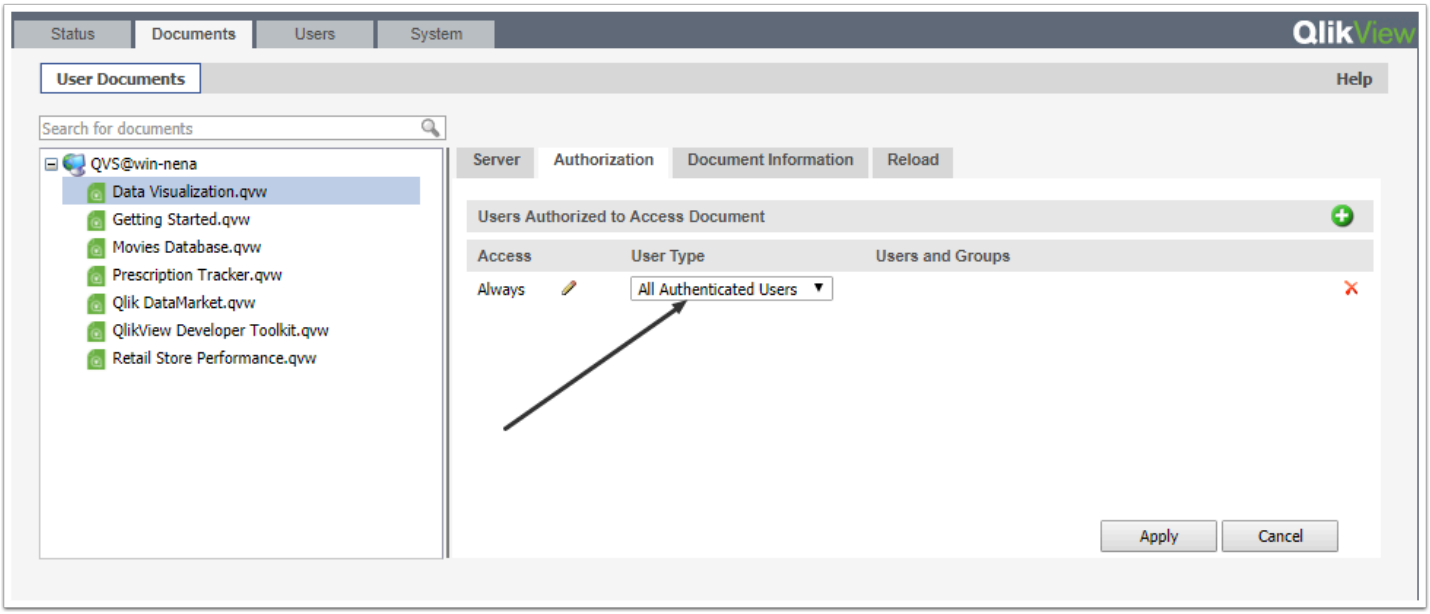
NOTE: Make sure that the user whose credentials are provided when establishing connection with QlikView server has access to required **Documents** and is a Member of a **QlikView Administrators Group** (see the picture below).



Make sure that Windows user that is used in the QlikView connection profiles has permissions to access .qvw document in Windows (Right click *.qvw doc* > *Security* > *Edit* > *Add qlik user* > *Apply Full control permissions* and Save). After that we can get QlikView object into Metric Insights. This needs to be done because with NTFS auth Windows controls docs permissions and if user is not in the list of users that have full access to a doc, QlikView plugin can't open this object and get its Metadata.



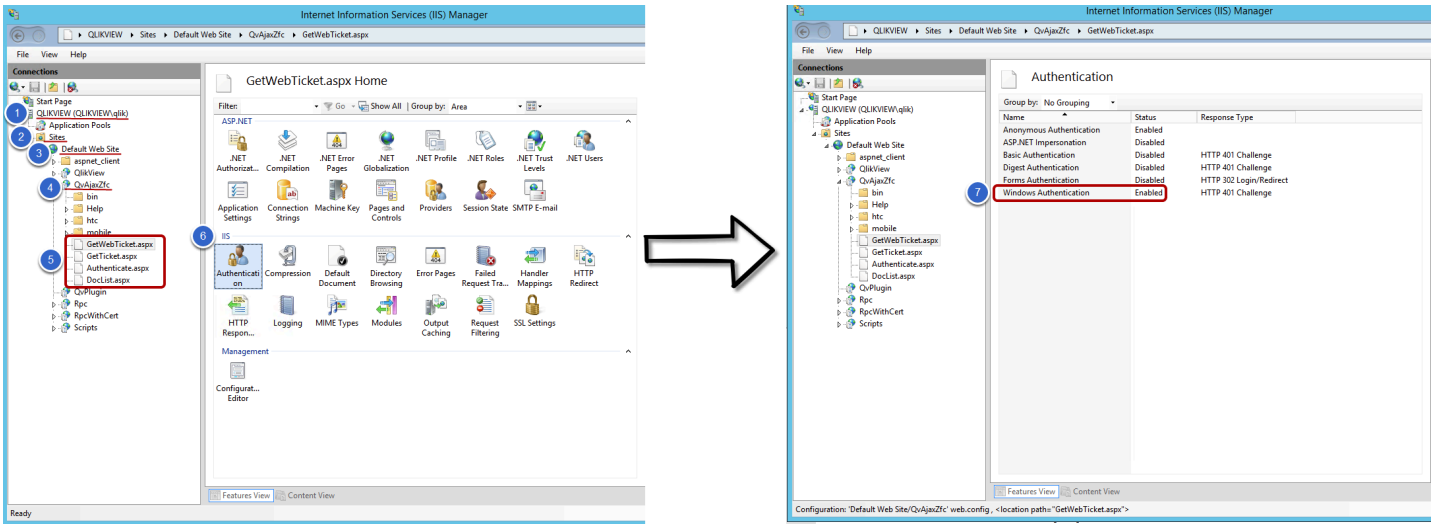
All Authorised users should get access to the **Documents** (see the picture below).



1.2. IIS Authorization

Ensure the following is installed:

- ASP.NET 4.5
- ISAPI Extensions
- .NET Extensibility
- Dynamic Content Compression
- Windows Authentication



To setup an Authentication via IIS, navigate to:

QLIKVIEW > *Sites* > *Default Web Site* > *QvAjaxZfc* > select a file and click **Authentication** at the folder option screen > Make sure that Windows Authentication is set to 'Enabled'.

Next: Collecting data. Finally, once setup, pulling data from QlikView is easy. You configure a data connection in Metric Insights. Then begin collecting data.

27.4 Establish Connectivity to QlikView

This article describes the process of creating plug-in Data Source to connect to QlikView. This Data Source will allow data from existing QlikView objects to be used in building elements using Metric Insights tools.

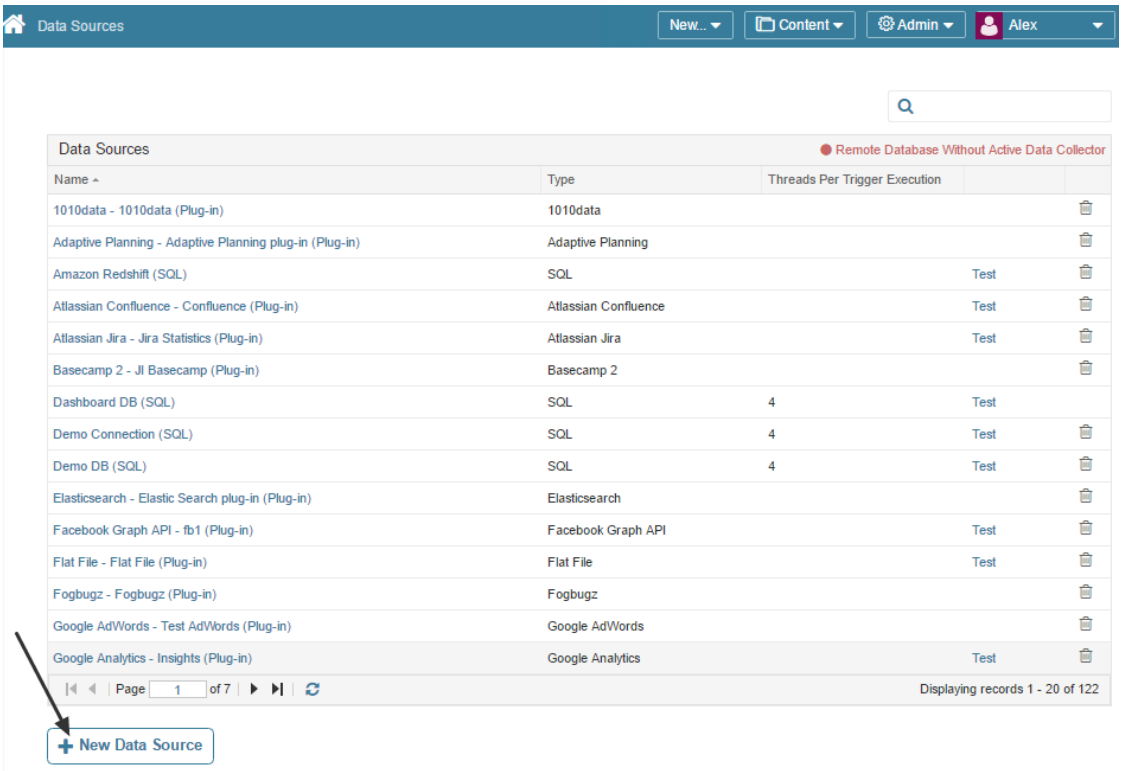
PREREQUISITES:

Your Metric Insights instance must be configured to support QlikView. This requires you to Configure a Remote Data Collector on your QlikView server or another Windows machine that has network access to your QlikView server. You will also need to install JRE (Java Runtime Engine) and .NET on the machine that runs the Remote Data Collector.

This article covers:

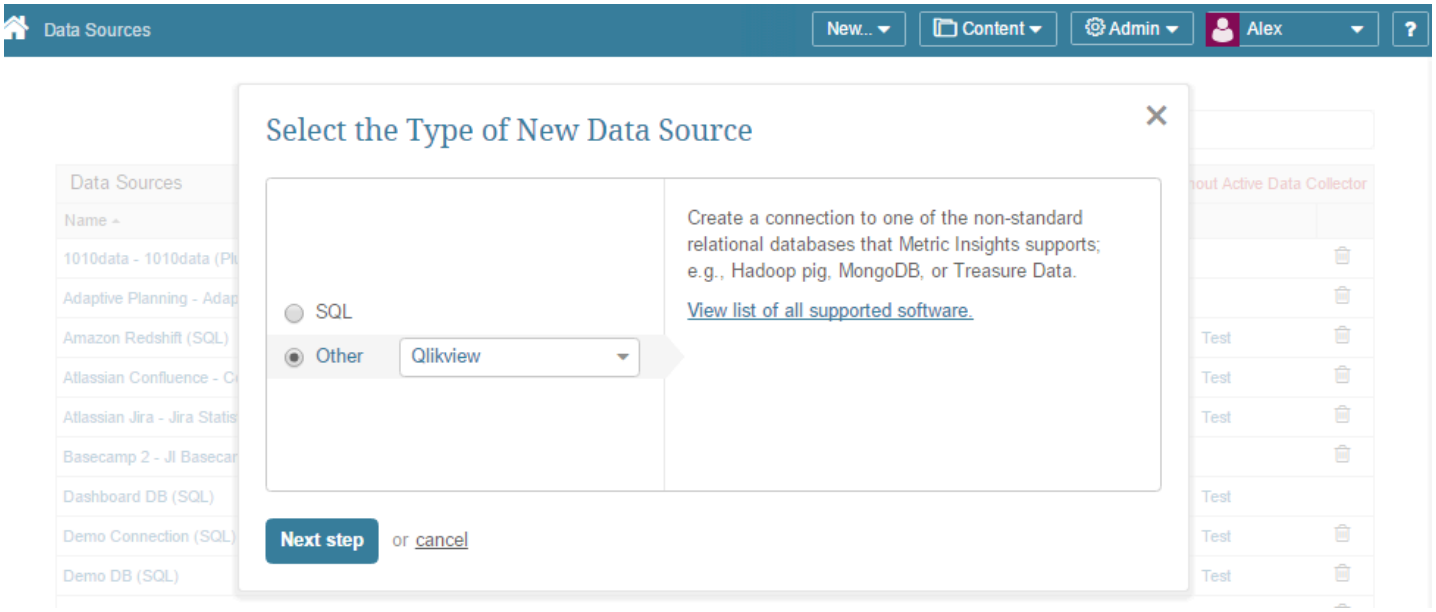
- Required Qlikview Parameters
- Optional Qlikview Parameters
- Advanced Configurations
- Troubleshooting

1. Access Admin > Data Sources



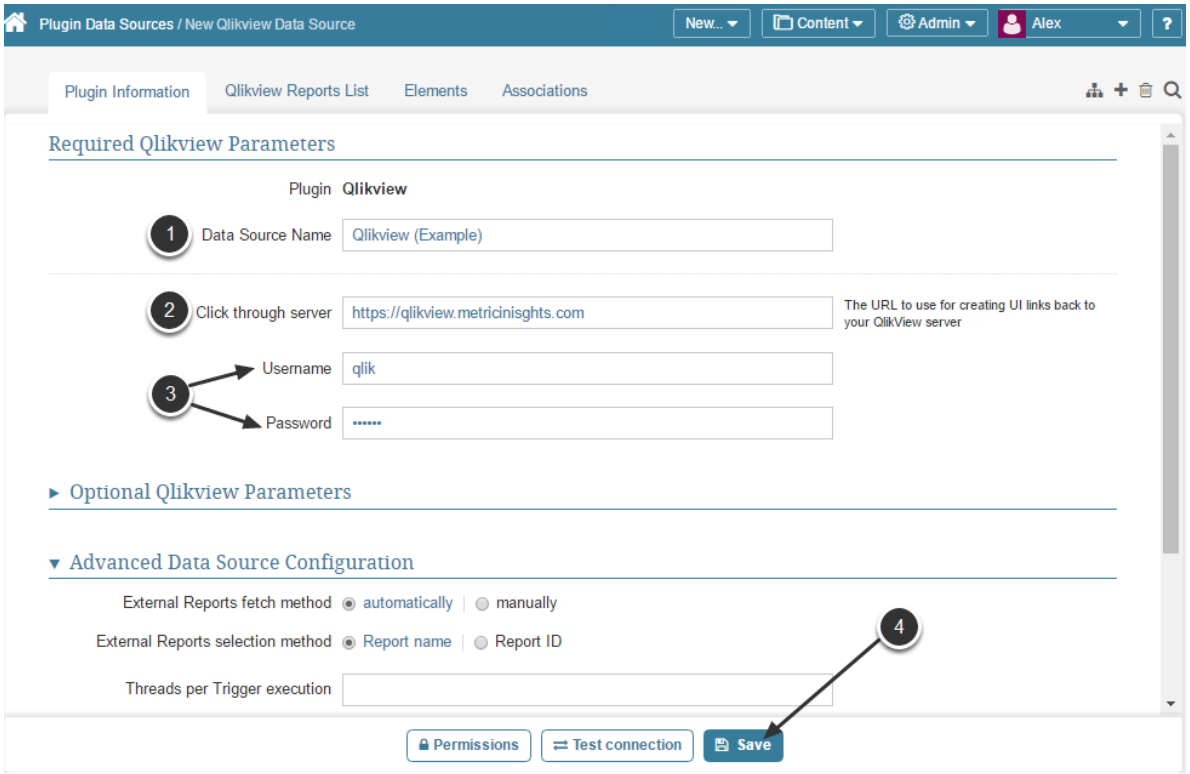
At the bottom of the screen click **[+New Data Source].**

2. Select "Other" Data Source Type and choose "Qlikview" from the drop-down



Move to the **Next step.**

3. Provide Required Qlikview Parameters

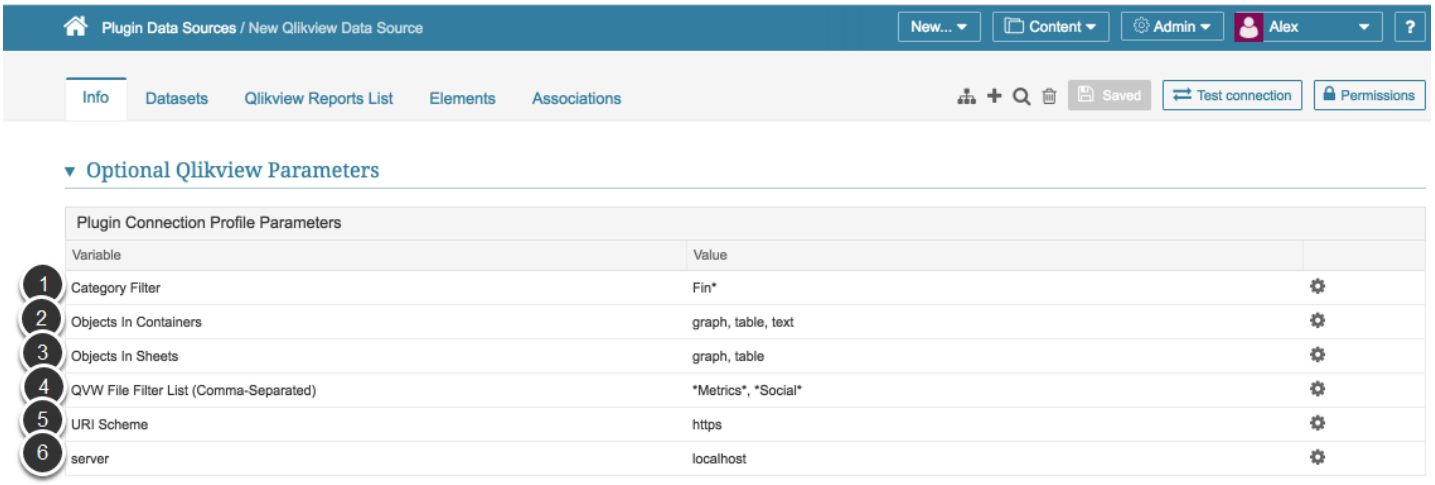


Specify how to connect to QlikView. The parameters include:

1. **Data Source Name** Will default but you may modify it.
2. **Click through server:** The URL to use for creating UI links back to your QlikView server
3. **Username / Password:** Note that your **Username** must be in the same format that your QlikView server uses for Authentication. For example, if Active Directory then you typically include the Domain like: <Domain>\<Username> (e.g., corp\edun)
4. **Save** your entries.

4. Optional Qlikview Parameters

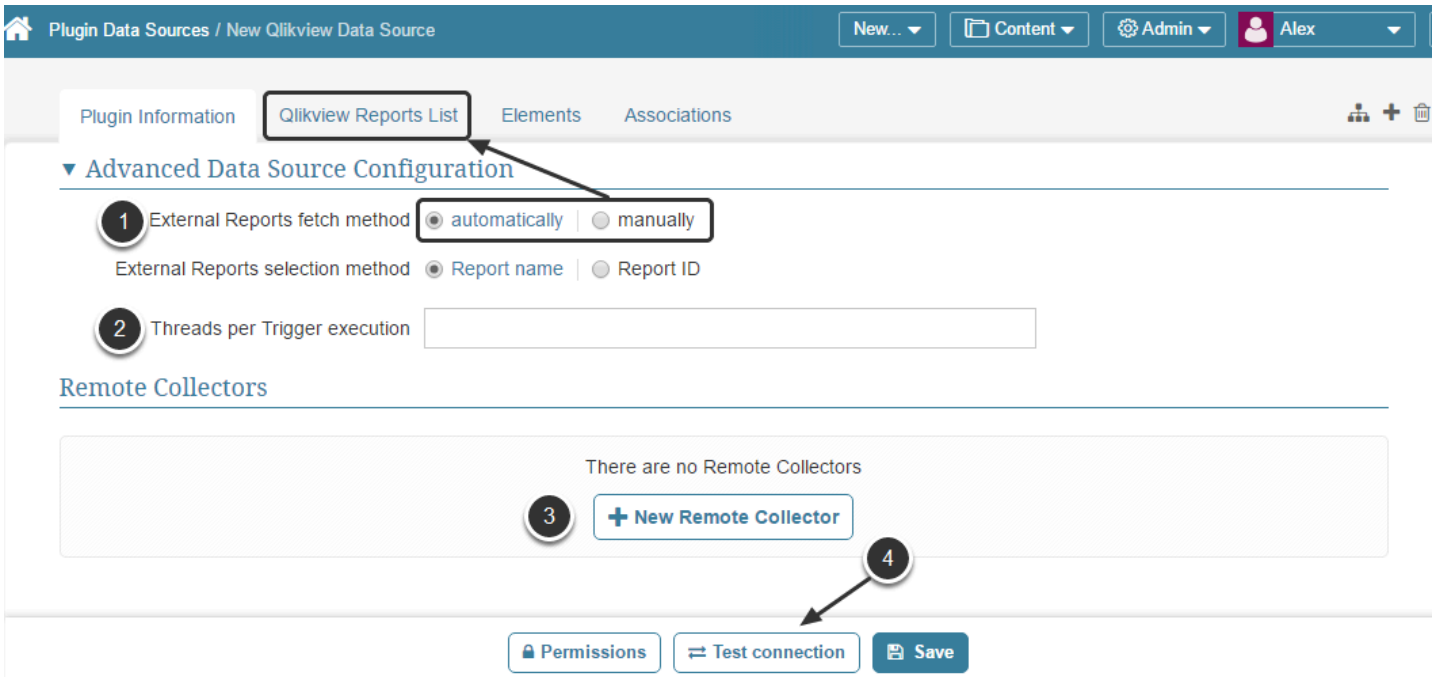
You can edit any values in the **Plugin Connection Profile Parameters** grid by clicking the gear icon in the corresponding row:



	Qlikview Parameter	Variable Name	Description
1	Category Filter	categoryFilter	Specify the names of QlikView categories (comma-separated) to pull data from these categories only (e.g., Financial, Social). Wild card (*) is allowed. (e.g., Fin*, S*).
2	Objects in Containers	objectsInContainers	Specify the types of QlikView objects to pull data from Containers (e.g., graph, table, text)
3	Objects in Sheets	objectsInSheets	Specify the types of QlikView objects to pull data from Sheets (e.g., graph, table, text)
4	QVW File Filter List (Comma-Separated)	qvwFilter	Used to narrow the list of QlikView documents (comma-separated) before loading of QVW files. Wild card (*) is allowed. (e.g., Social Media*, demo*)
5	URI	scheme	Enter URI Scheme to access the QlikView Server (either http or https)

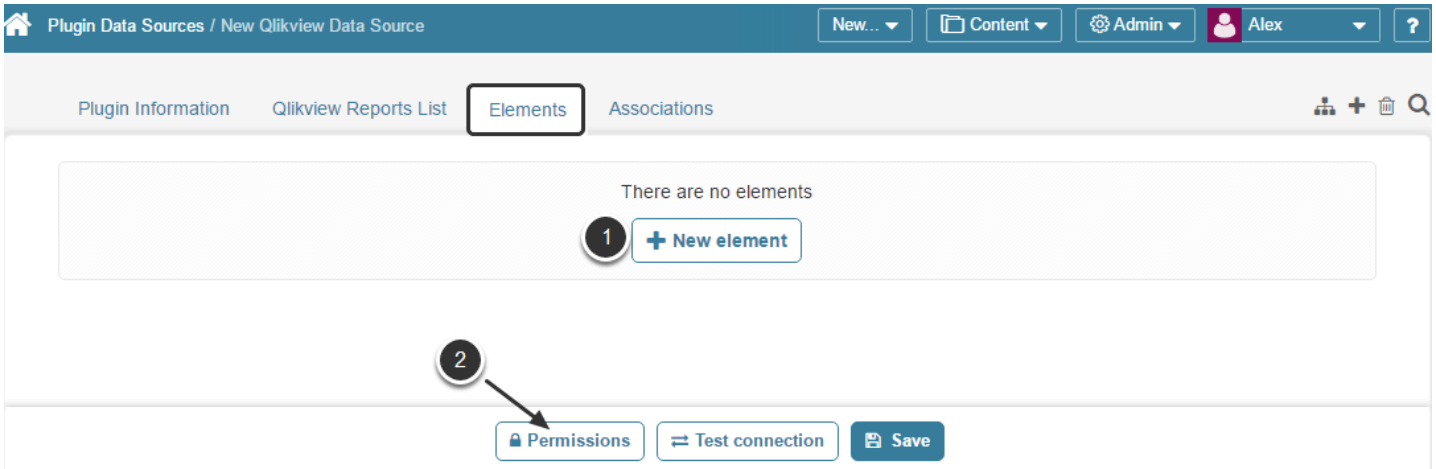
	Scheme		
6	server	server	Enter server address. If Remote Data Collector runs on the QlikView server then enter 'localhost'; otherwise if it runs on a different machine, enter the QlikView server name (e.g. server=qlikview.metricinsights.com). Leave blank if pulling .qvw documents via the file system

5. Advanced Configuration



1. **External Reports fetch method**: This setting influences options available in the *Qlikview Report List* tab:
   - **automatically**: just click **Refresh list** and all Reports are going to be fetched by the system
   - **manually**: Reports may be added one-by-one or via CSV file
2. Optionally, specify the maximum number of concurrent Threads per Trigger execution to be used in background processing when the system updates Metrics and Reports for this Data Source. If you do not specify any value for this setting, batch data collection processing will be single-threaded.
3. Select a **New Remote Collector.** See more on [Configuring Remote Data Collector]
4. **Test connection**

Other settings



1. You can create elements directly from the *Elements* tab
2. Click **Permissions** to assign them to Groups or Power Users

Troubleshooting

When setting up connectivity to QlikView you can use the **Test Connection** button to verify that your Plug-in parameters are correct.

The following is a list of errors with possible actions

1. **Error:** There is currently NO data collector configured to service this remote data fetch request.

   Action: Add a remote data collector in the Remote Collector section.

2. **Error:** Fail to get ticket from 'server'

   Action: Add the user to the QlikView Administrators group on the QlikView server machine. This is required by QlikView in order to use the QlikView server Ticket API. More information about the QlikView server **Ticket API** can be found here. Note: you can work around this by configuring Metric Insights to access the .QVW files via the file system instead of via the QlikView server Ticket API. See the section on setting the **server** parameter. Also note that one of the requirements for QlikView is that QlikView authorization uses DMS and not NTFS: "*QlikView needs to be running in DMS mode for security*"

3. **Error:** The underlying connection was closed: Could not establish trust relationship for the SSL/TLS secure channel.

   Action: For https (ssl) connection to your QlikView server, make sure that the 'server' parameter contains the correct host name that matches the server certificate of QlikView. One way to test is to point your web browser to the 'server' parameter and verify that you do not

see an error that indicates that the Server's certificate does not match the URL. This error could result from using IP address instead of host name, or vice versa.

4. **Error:** The remote server returned an error: (401) Unauthorized.

   Action: Verify username and password are correct.

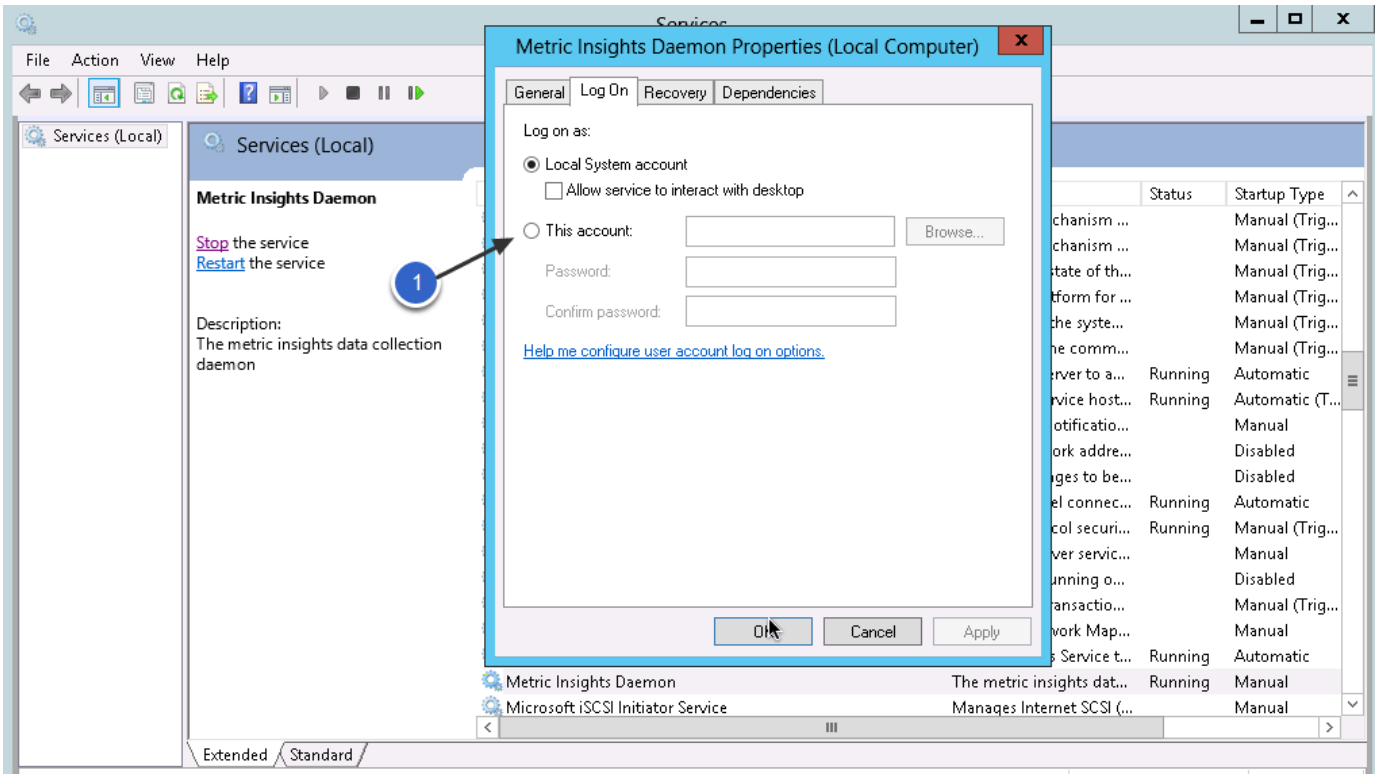
In addition, if SSL connection to QlikView server, then make sure plugin parameter 'URI Scheme' is 'https'.

Also, if you installed the remote data collector on the QlikView server, then make sure that the 'server' parameter is 'localhost' and not the name of the server (such as 'qlik.metricinsights.com)

5. **Error:** The remote name could not be resolved: 'server'

   Action: Make sure you have network connectivity from the machine that runs the Metric Insights data collector and your QlikView server.

Set user credentials when pulling .QVW file data via file system



When pulling QVW data from the file system instead of going through QlikView server, you will need to pay attention to what User is configured to run the Windows Service for the QlikView plugin.

This is not common.

Note: if you specified the 'server' parameter with the QlikView server name, then you can skip this step.

If you left the 'server' parameter blank, so that you will pull .QVW document data via the file system, then you will want to complete this step.

User credentials are specified in the the Remote Data Collector. This user is utilized for accessing the .QVW documents on the file system.

On the machine where the Remote Data Collector is installed, open the 'Metric Insights Daemon' Windows Service

1. Set the User in Log On menu. The default user is the local system account. You can enter a different user here. You will then need to restart the Windows Service for it to take effect.

What's Next?

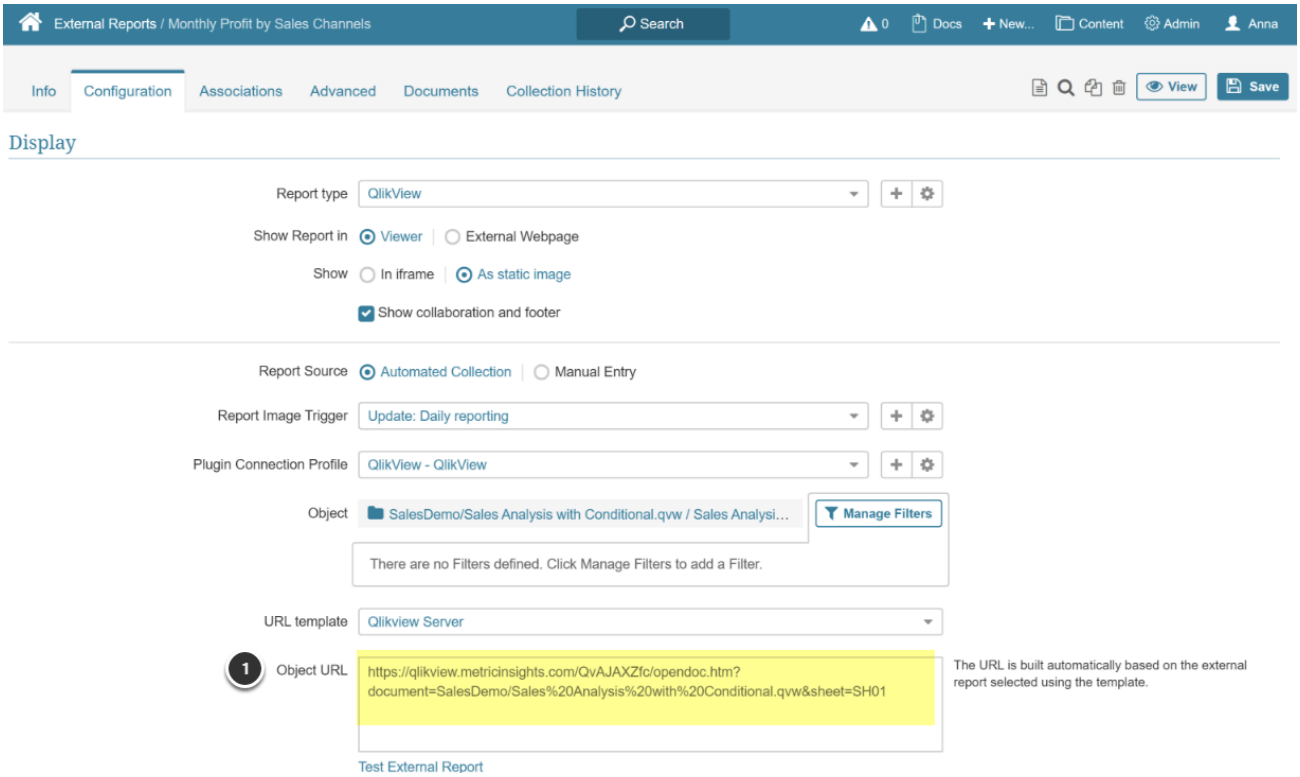
- Collect Data from QlikView
- Create External Report from QlikView
- Pre-Filter for QlikView

27.5 Find Field Name and Alias in QlikView

This article describes how to find the field name and alias (ID) in a QlikView dashboard.

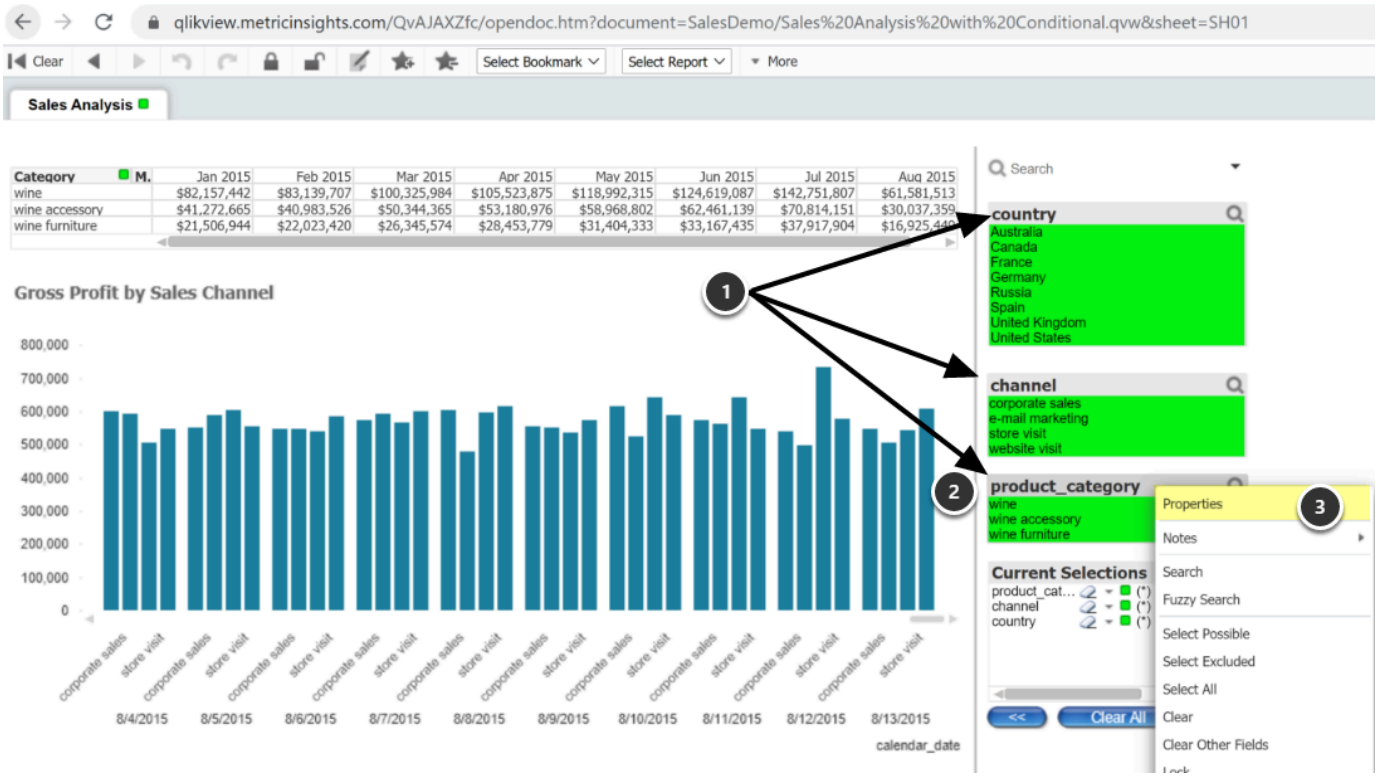
Prerequisite: Establish Connectivity for QlikView

Access the QlikView Dashboard from your External Report



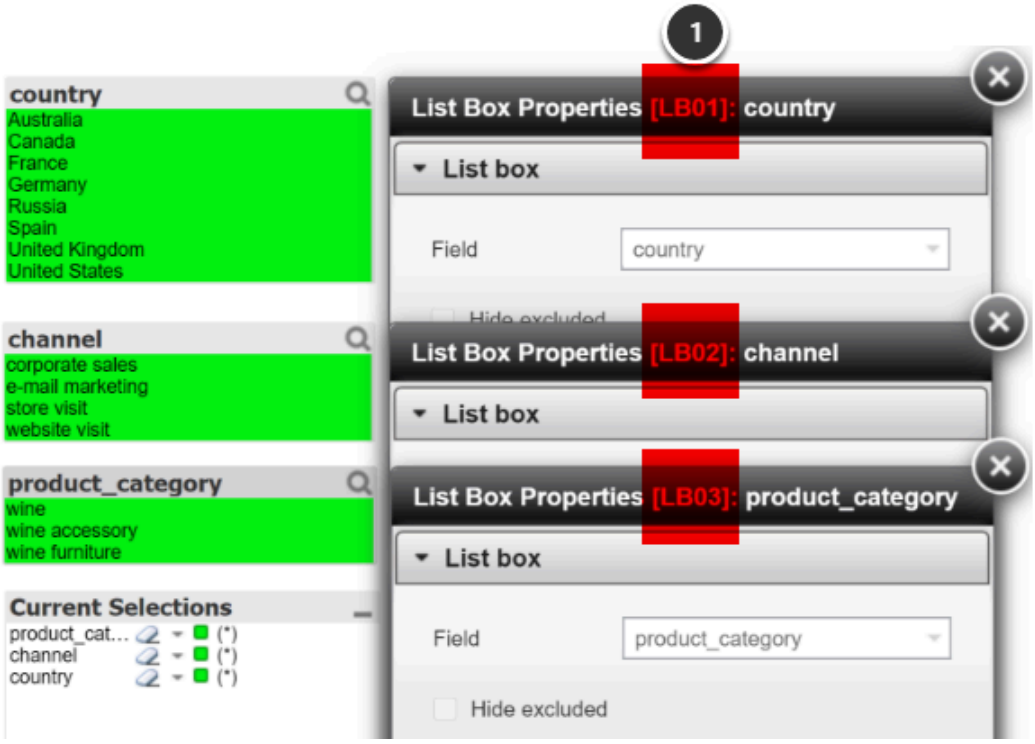
1. Copy the URL and put it in the Command line in a new tab to access QlikView object

Locate filter name(s)



1. Find the filters on right-side of panel - the filters are case-sensitive
2. Right click on a filter name to expose dropdown
3. Select "Properties" from dropdown

Locate filter alias in List Box Properties



1. The alias field is located in top line within [ ]. This example shows the aliases to be "LB01, LB02, and LB03". These fields are case-sensitive.

27.6 Create External Report from QlikView

External Reports using a QlikView Data Source are defined using the Editor. Using filters on the report focuses on including the slice of data needed and excluding those values that are irrelevant to the research.

PREREQUISITES:

Establish Connectivity to QlikView

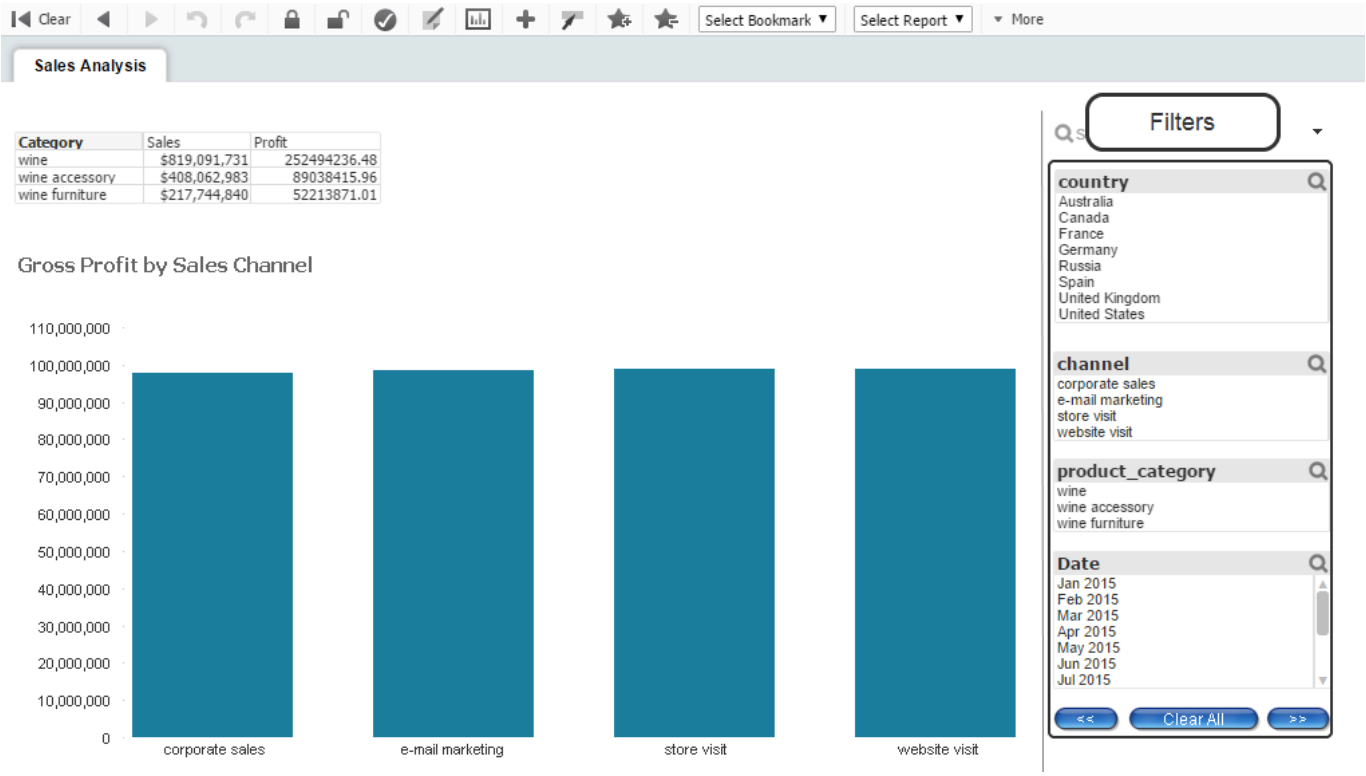
Identifying QlikView Filters

Sample of a QlikView Dashboard

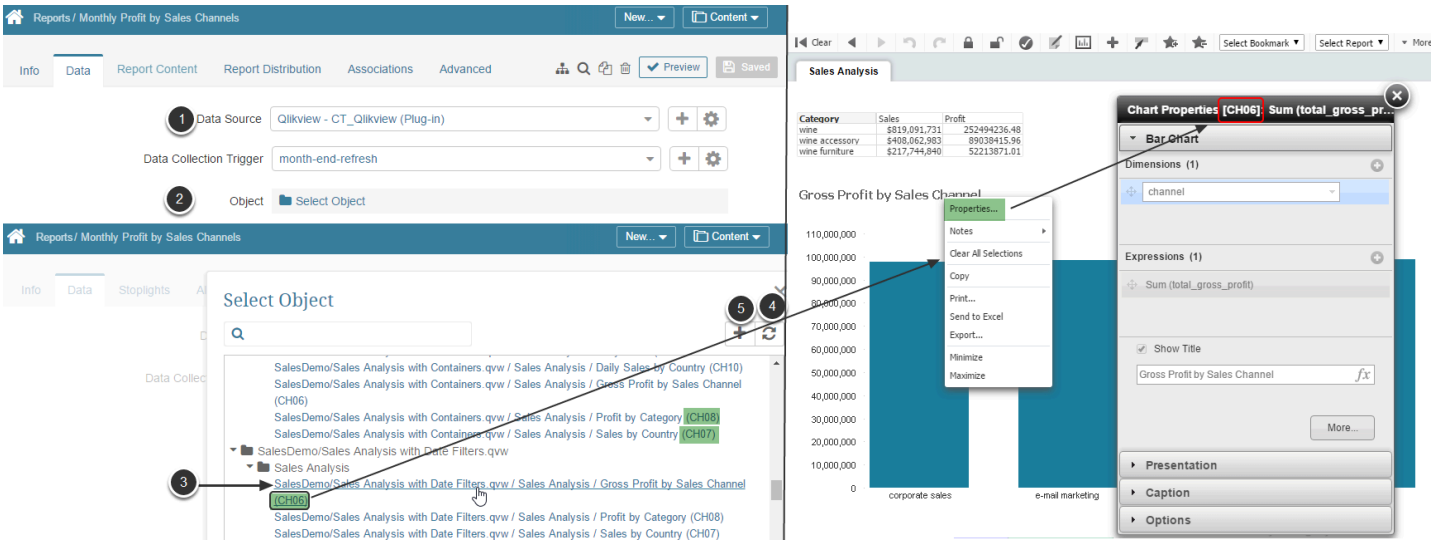
Most of the data displayed on the QlikView Objects can be filtered by the offered criteria.

On the example below, all available filters are located to the right of the graph. The given example allows modifying the View by including or excluding values from the following filters:

- Country
- Channel
- Product Category
- Date



1. Define a Source element (Object) for Data Collection



Start off by creating an element. Once you get to the process of Data Collection, define the following:

1. **Data Source:** This is an entity that connects QlikView and Metric Insights. For more information, see: [Establish Connectivity to QlikView]
2. **Object**: Click **Select Object** to open the pop-up with the list of available Reports.
3. Each item in the list is represented as the path (hierarchy) to the respective object in QlikView. Find the object in the list.
   - **ID next to the Report Reference:** Note that each item in the list has its code next to its path. Each object in QlikView has such ID. To see the ID of a particular object, go to the Qlik server, right click it and choose **Properties** from the opened context menu. The pop up listing object details opens and the ID can be found in its header.
4. If you do not see the required item, try refreshing the list by clicking the **Refresh** icon at the upper right corner of the pop-up.
5. Alternatively, rather than choosing an Object from the list in the pop-up, you can manually add Object reference as follows:

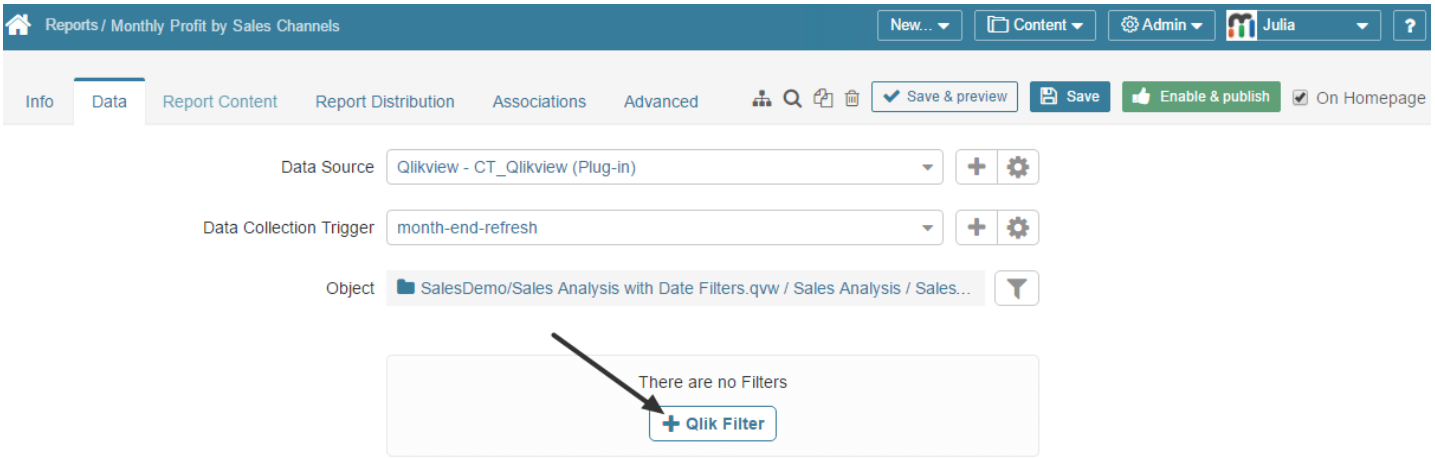
```
[Application name]&object=[Sheet name]&object=[Object name]
```

2. Adding QlikView Filters to Metric Insights

⚠ Once filters are added to a Report or External Report for the first time, they are going to be automatically added to all new respective elements with the same Data Source / Sheet.

NOTE:

- External filters are tied to QlikView Sheets, not Metric Insights' elements. This allows Filters to be reused for multiple elements (there is no need to create new Filters every time an element is created in Metric Insights).
- If there are more External Filters or Filter Values that you would like to use for the current element, you can always set the redundant ones to "ignore".

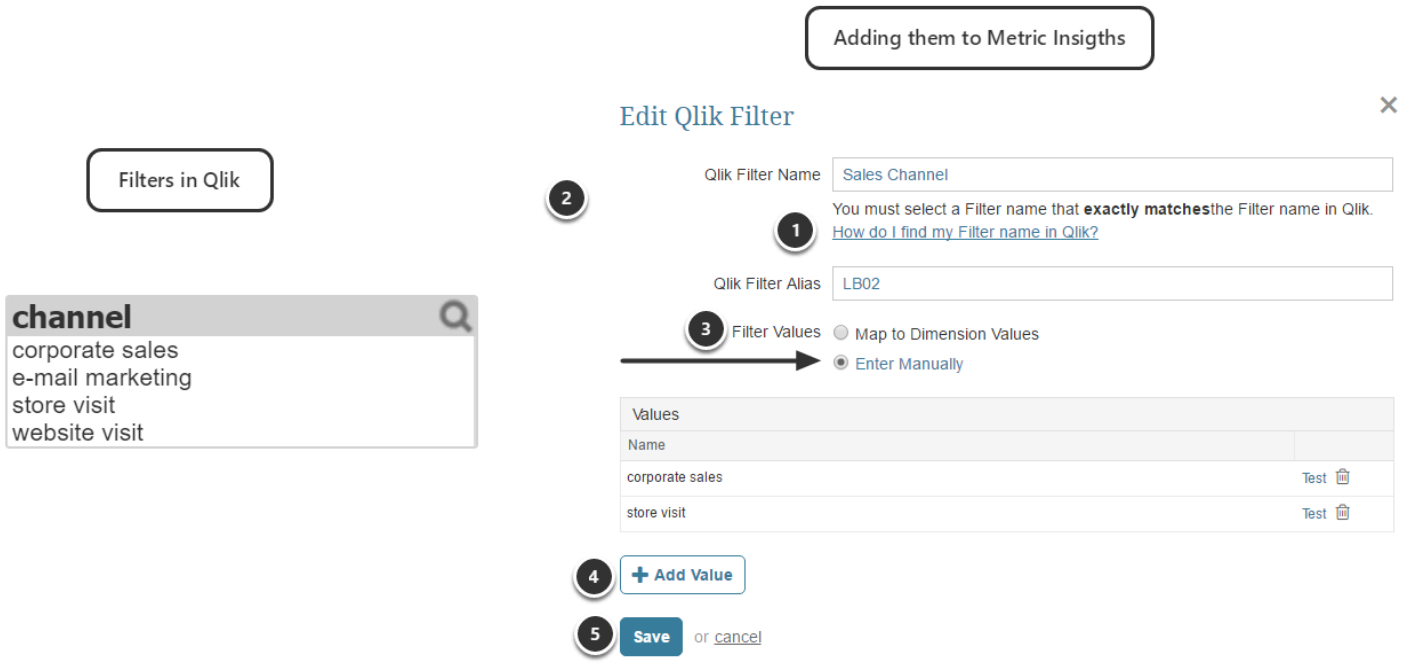


When creating a Metric / Report / External Report fetched from QlikView, after you define the **Object** that should serve as a Data Source, you may pre-filter information that is going to be fetched.

To do that, click **[+ Qlik Filter]**. Next, you can choose whether you are going to define filters manually or via the existing Dimension Values.

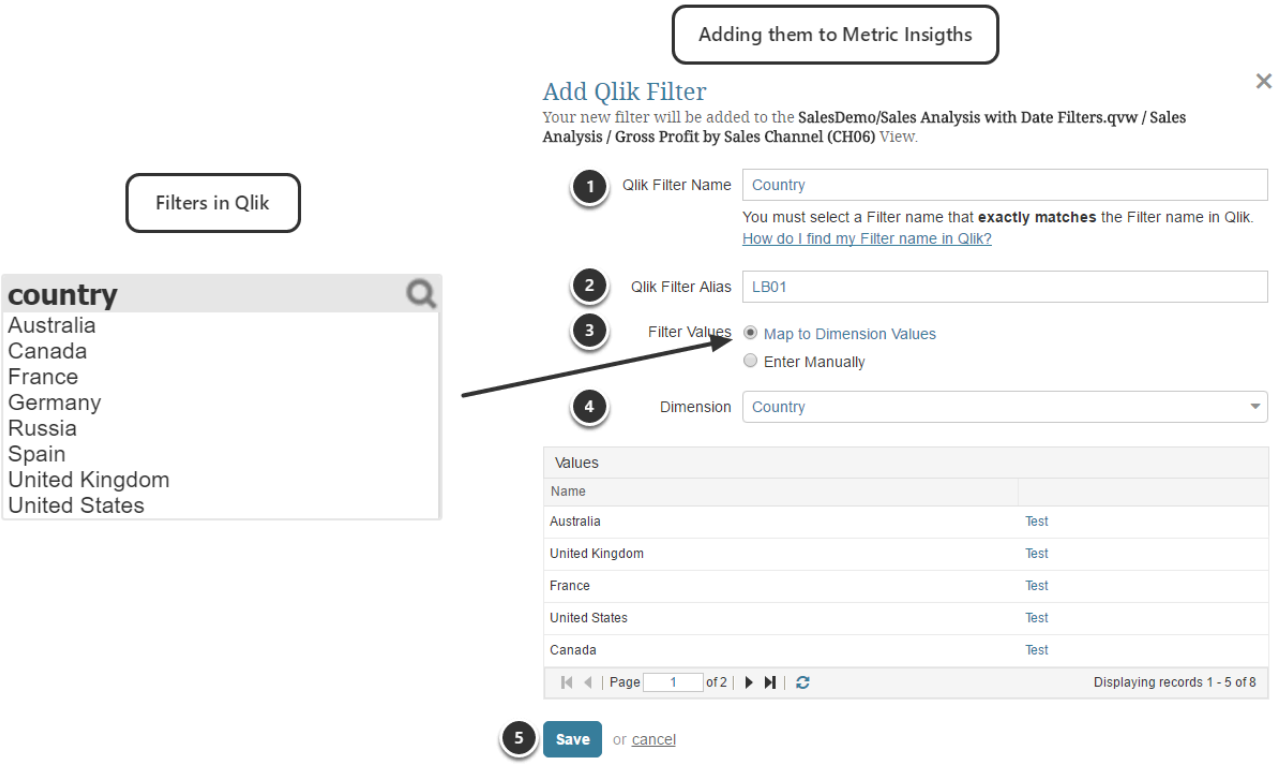
NOTE: Examples given below are taken from the QlikView Objects shown at the top of the page.

2.1. Enter Manually



1. **Qlik Filter Name & Qlik Filter Alias:** Click on "How do I find my Filter name in Qlik" if you need help determining these fields
2. **Filter Values:** choose 'Enter Manually'
3. Click **[+ Add Value]** and in the opened pop-up manually type in the name of the filter, for example, "corporate sales". **Save** your entry. All added values should appear in the *Values* list.
4. **Save** your entries.

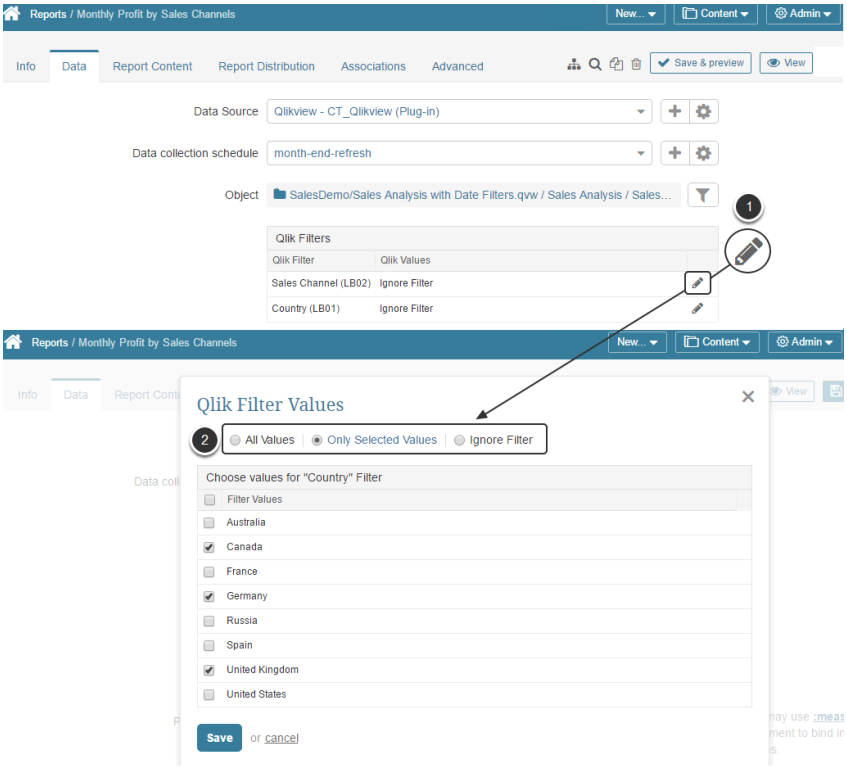
2.2. Using Dimension Values



If you have already used Qlik filters to create Dimensions in Metric Insights, you can quickly choose which Dimension Values you want to use for pre-filtering:

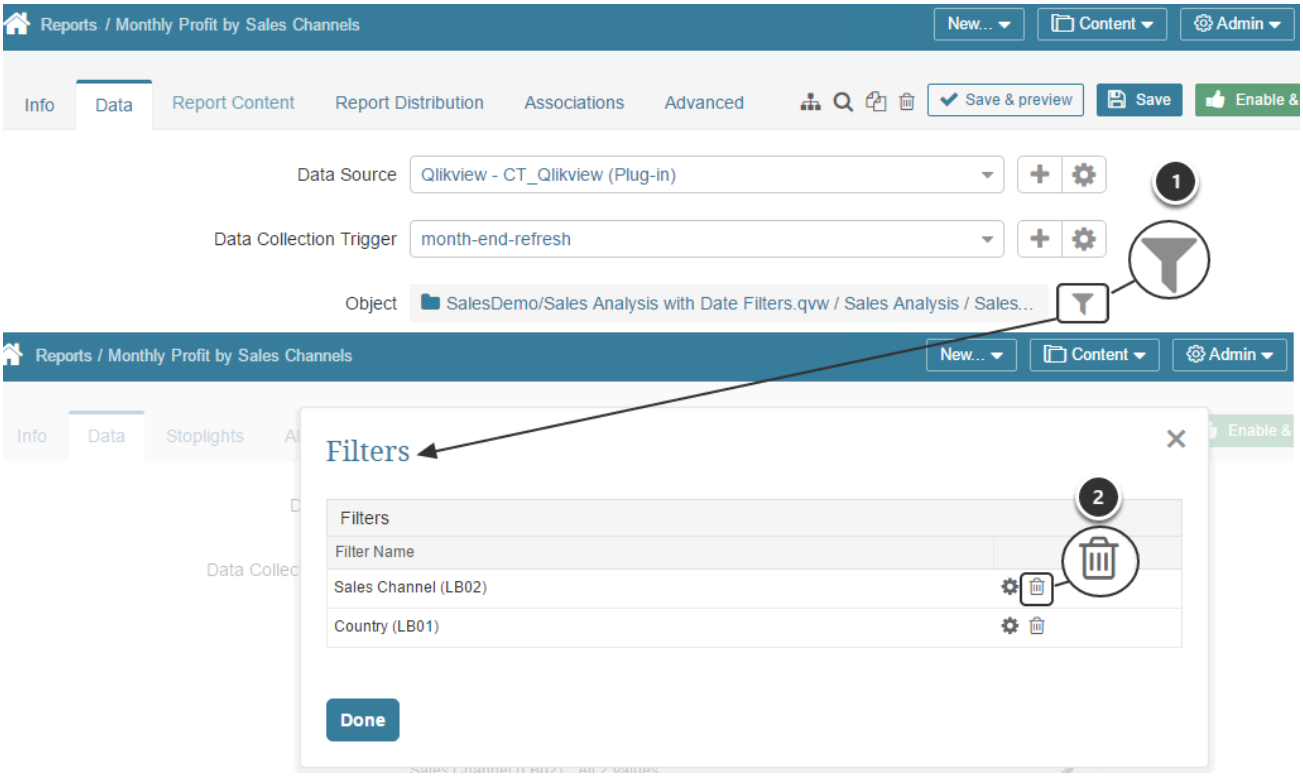
1. **Qlik Filter Name & Qlik Filter Alias:** Click on "How do I find my Filter name in Qlik" if you need help determining these fields
2. **Filter Values:** choose 'Use Dimension Values'.
3. **Dimension:** select a corresponding Dimension from the drop-down list and all its Dimension Values are going to be loaded to the Values list automatically. For more details refer to: [Create a Dimension with values fetched from QlikView]
4. **Save** your entry.

3. How do I add filters to a results set from Qlik?



1. Click the **Pencil** icon in the filter row to set it up.
2. When the filter is added, you can use it for "All Values", "Only Selected Values" or ignore it.

4. Deleting Filters



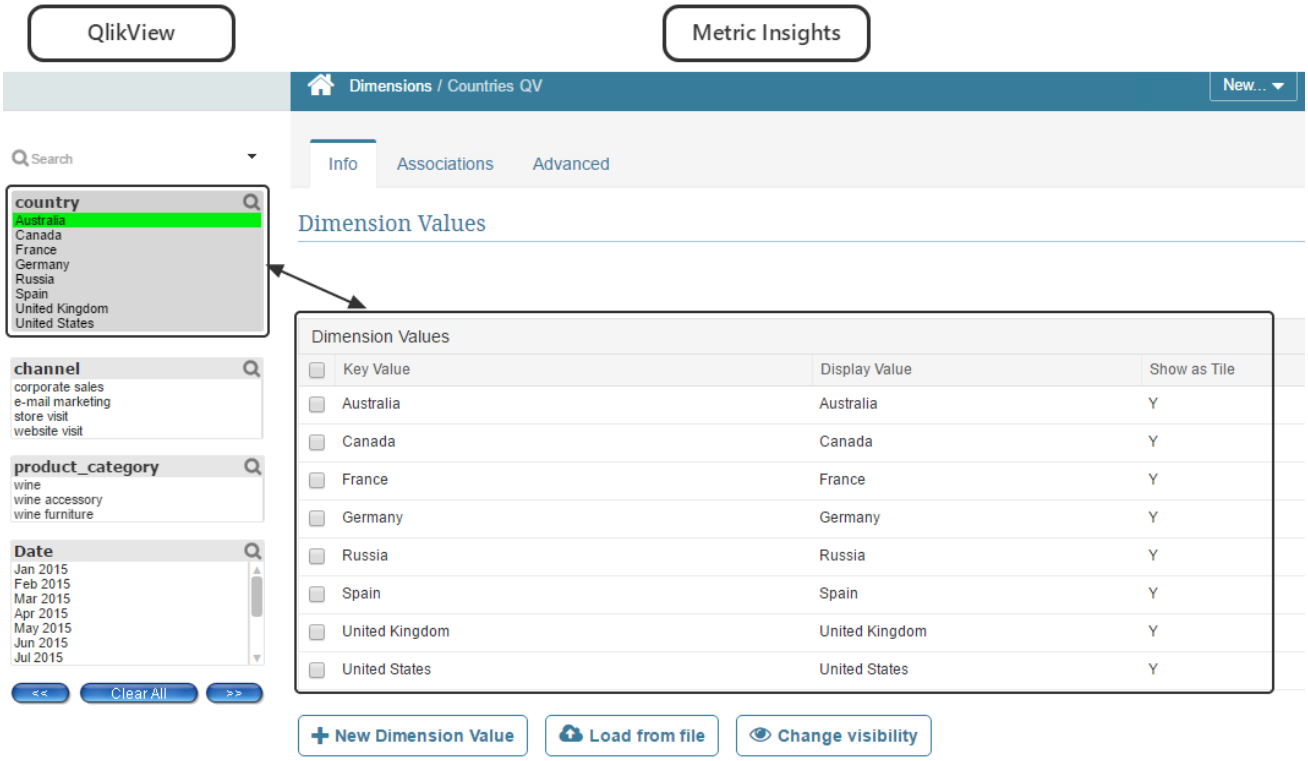
1. Click the **Filter** icon in the **Object** field open Filter popup
2. Click the **Trashbin** icon in the respective row to delete the filter

27.7 Create a Dimension with values fetched from QlikView

PREREQUISITES:

[Establish connectivity with QlikView]

Use Case



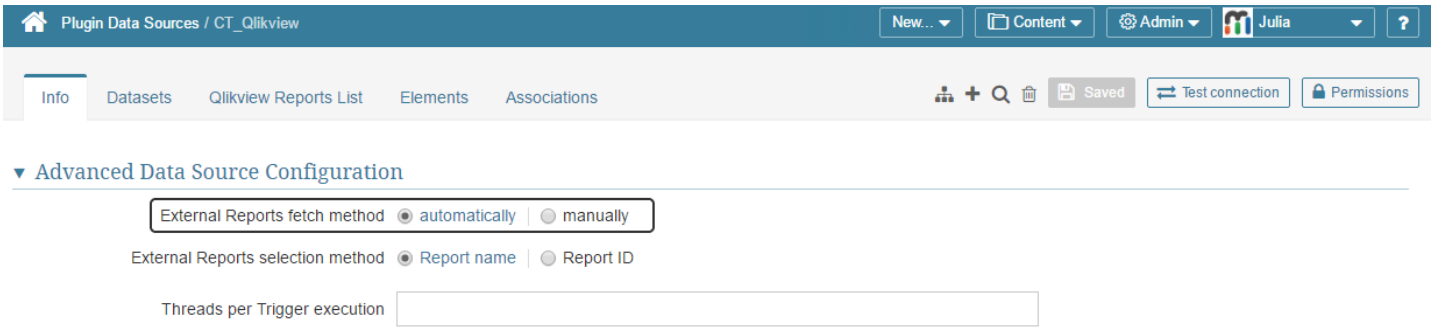
If your QlikView report contains information for several Dimensions, you do not need to create a separate element for each Dimension Value in Metric Insights. All you need to do is just copy QlikView "filter values" into MI. This article covers step-by-step instructions on fetching this data from QlikView.

NOTE: If a new filter value is added to this QlikView report later, it will be automatically copied to MI upon triggered data collection.

1. Open the QlikView Plugin you are going to use as a Data Source for the future Dimension

1. Go to *Admin* > *Data Sources.* The list with all data sources created in the system is going to be opened.
2. Select the plugin you plan to use as a Data Source for the future Dimension. Alternatively, crate a new one: Establish Connectivity to QlikView

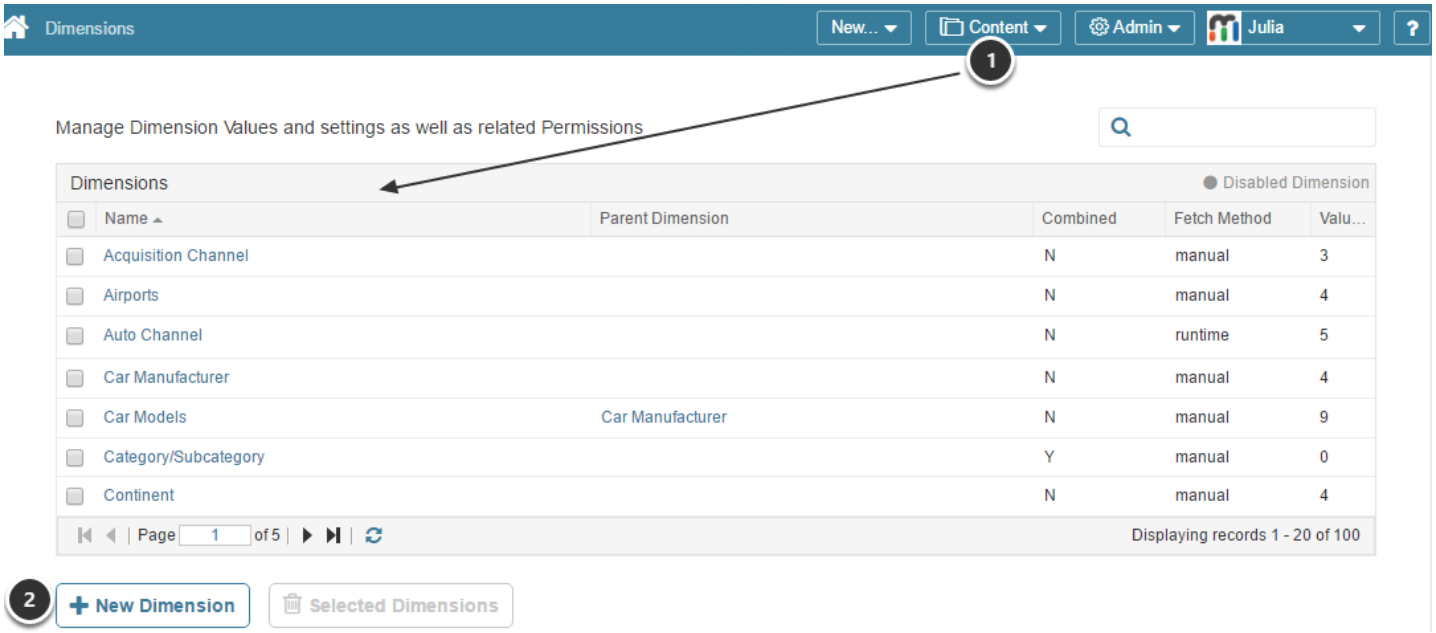
1.1. Update the list of QlikView Reports



Metric Insights extracts data from the QlikView server in the form of reports. Information obtained from these external reports is further used as data source for MI elements: Metrics, Reports, Datasets, etc. QlikView Reports can be added in the Plugin Editor. There are two options to add reports to the plugin:

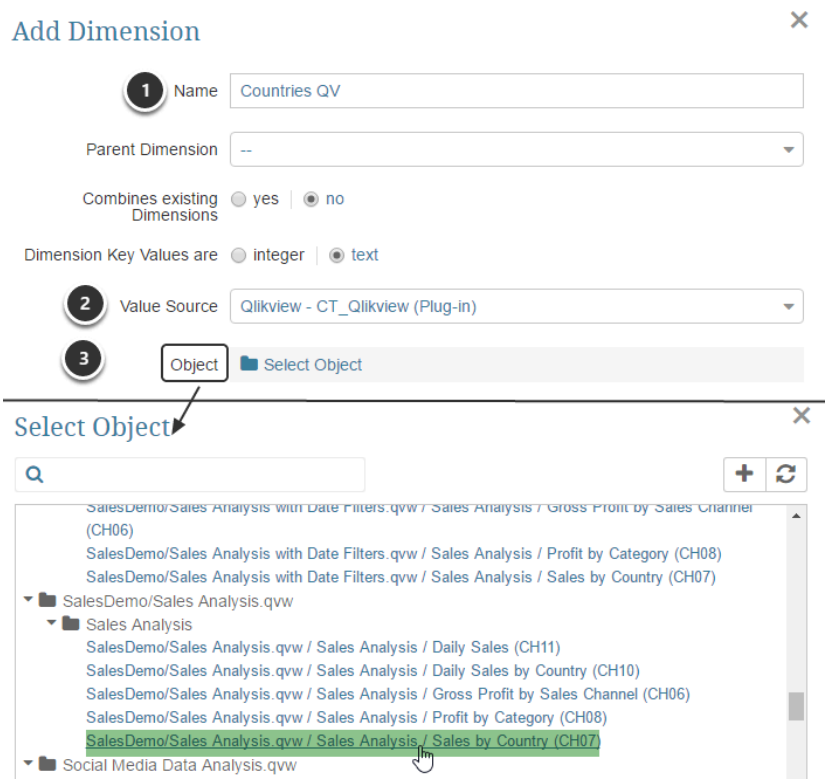
- **Automatically**: If the External Reports fetch method field is set to 'automatically', go to the QlikView Reports List tab and simply click **Refresh** list to collect all QlikView reports currently available at the server
- **Manually**: You can also update the QlikView Reports List by adding report IDs and Names one-by-one or via a CSV file

2. Add a new Dimension



1. Go to Content > Dimensions. The list with all dimensions created in the system opens.
2. Click **[+ New Dimension]**

2.1. Define the Basics

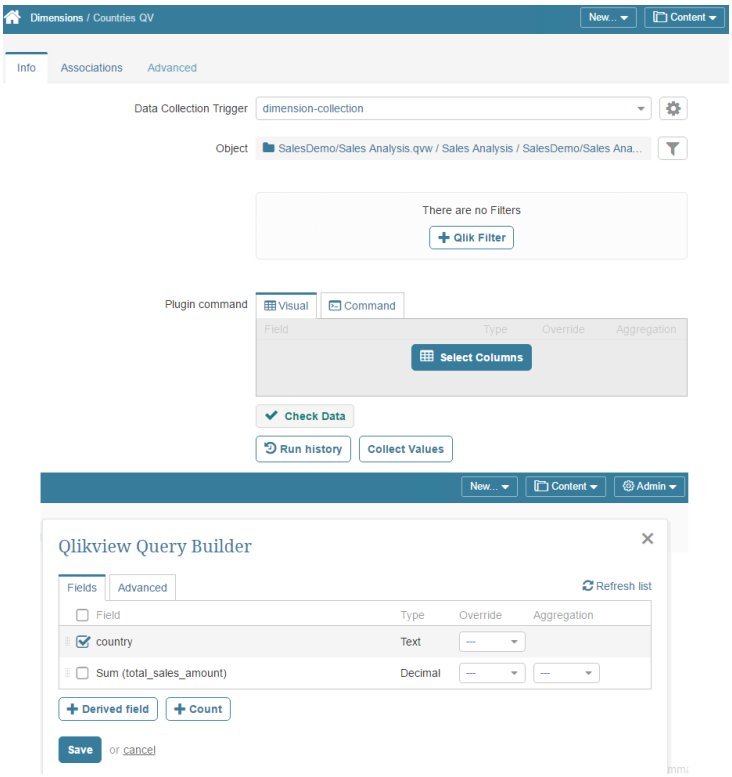


Provide the basic Dimension definition information, paying attention to:

1. A unique **Name** for your Dimension
2. **Value Source:** Specify how Dimension Values will be collected for the new Dimension. In this example, we are selecting 'QlikView' plugin from previous steps. It is going to serve as data source
3. **Object:** Define the QlikView item report that contains the required values in the **Select Object** pop up. In our example we need to fetch Country Values, so we have selected a 'Sales by Country' item since it contains the info we are looking for.

Save your entries. The *Dimension Editor* opens.

2.2. Enter the command for fetching data

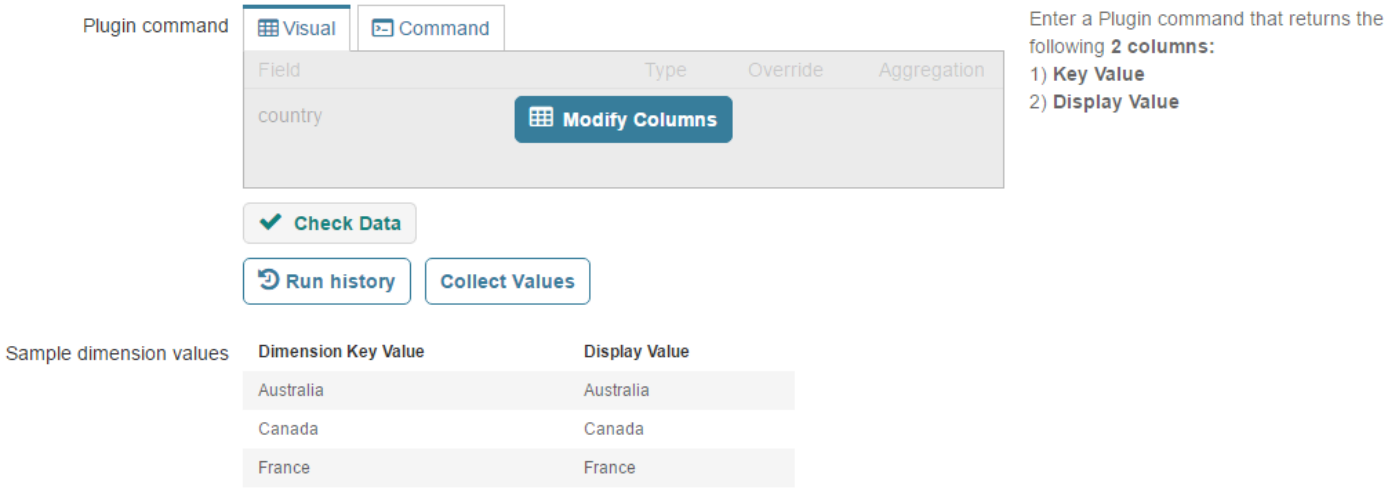


1. You can enter the command manually

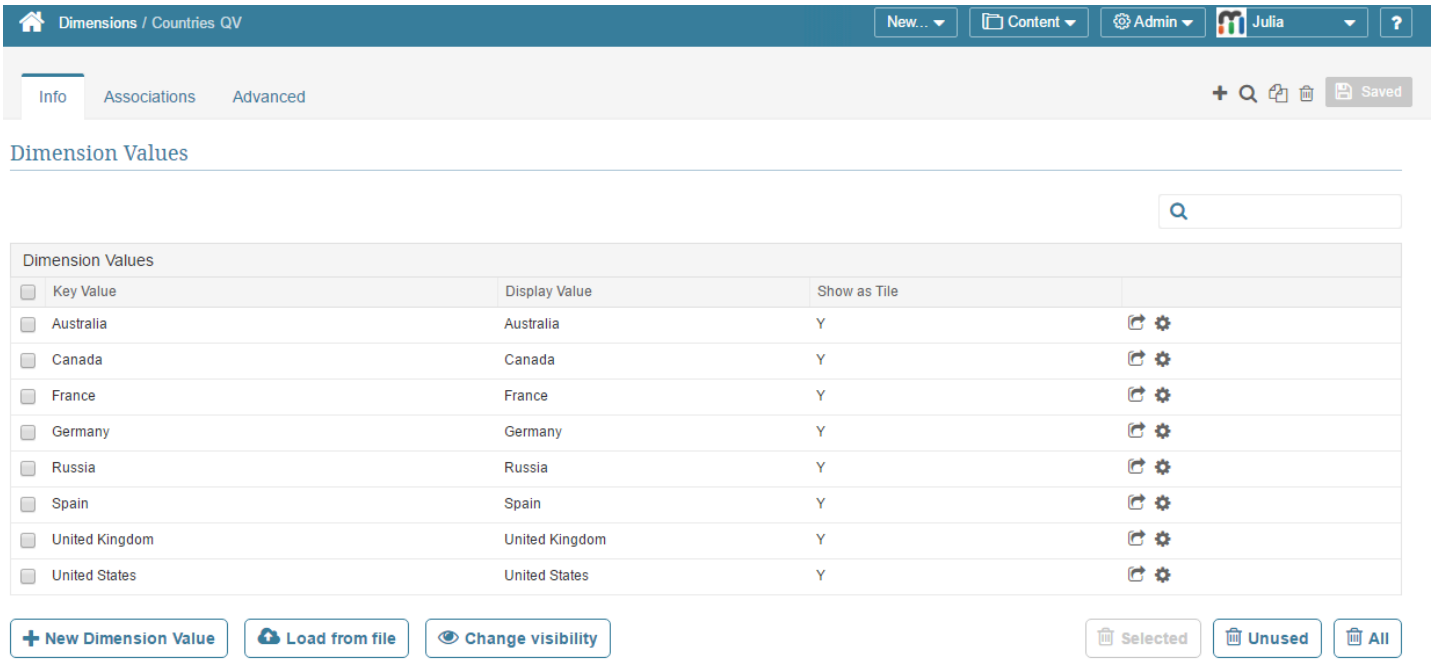
OR

2. Use a Visual Editor and choose the required fields

2.3. Check Data and Collect Values



2.4. Result



What's next?

You can now use this Dimension to create dimensioned Metrics, Reports and External Reports from QlikView.

28. Sourcing Data from Re:dash

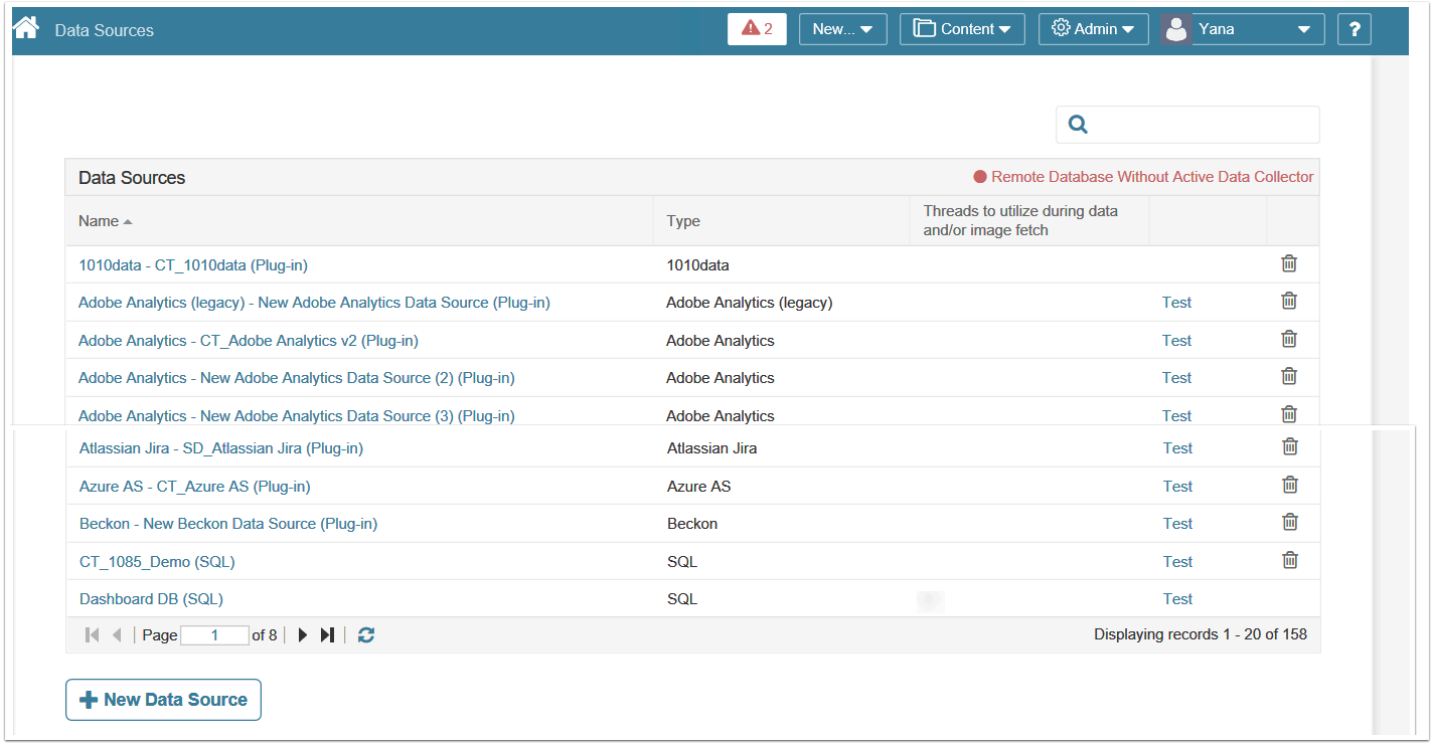
28.1 Establish Connectivity to Re:dash

This article describes how to connect to **Re:dash** server in order to use its reports as Data Sources in Metric Insights.

General instructions on setting up data sources based on plug-ins can be found here.

[Release 6.x]: 'Remote Data Collector' renamed to 'Remote Data Processor'

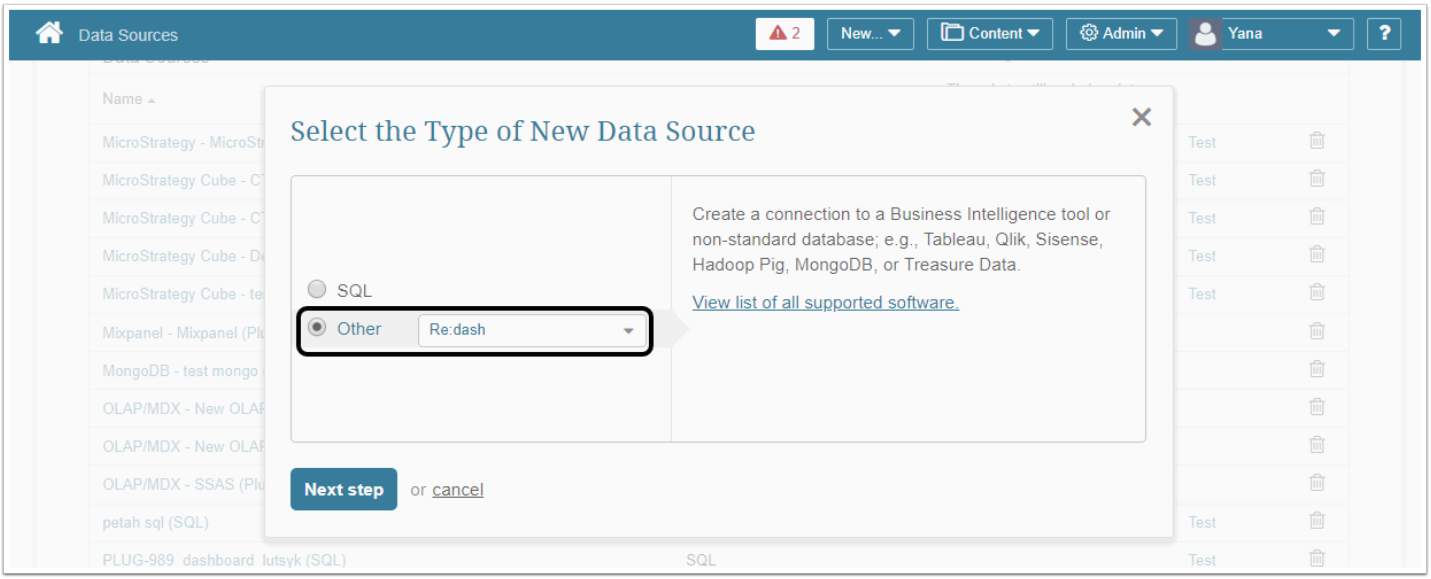
1. Access Admin > Data Sources



At the bottom of the screen click **[+ New Data Source].**

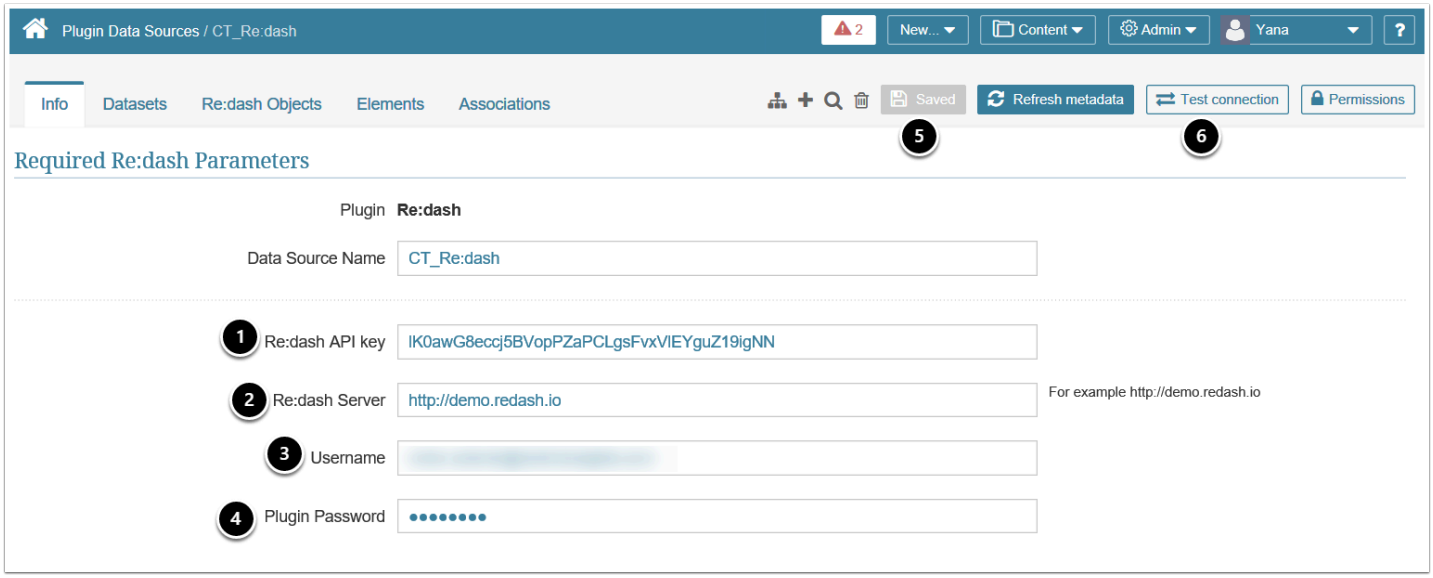
The *Select the Type of New Data Source* pop-up opens.

2. Select "Other" Data Source Type and choose "Re:dash" from the drop-down list



Move to the **Next step**

3. Provide the Required Re:dash Parameters

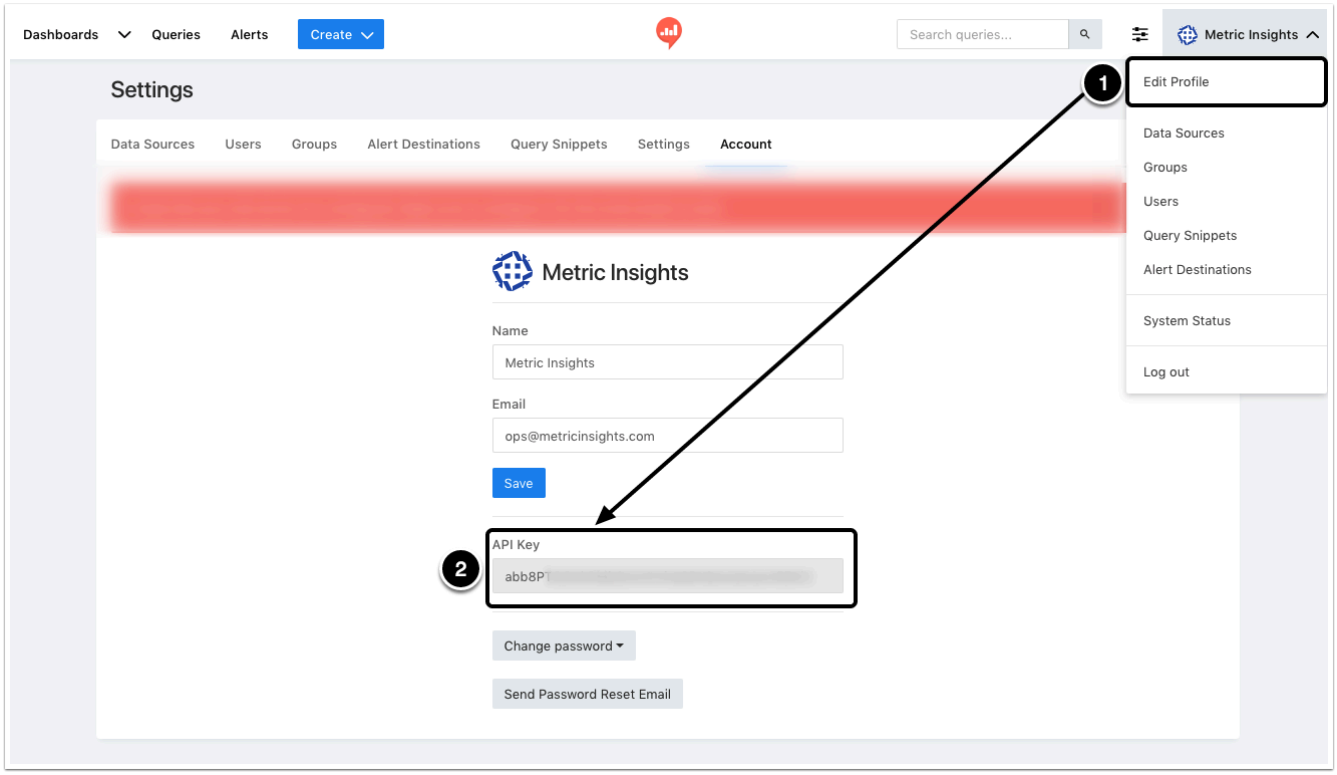


1. **Re:dash API key:** enter the API key from your **Re:dash Profile** settings. It can be used either alongside or without the Username/Password credentials.
2. **Re:dash Server:** Define the server protocol (**http** or **https**) and a hostname.
3. **Username:** Note that your **Username** must be in the same format that your Re:dash server uses for authentication.
4. **Plugin Password:** enter your password credential.

5. **Save** your entries.
6. **Test Connection.**

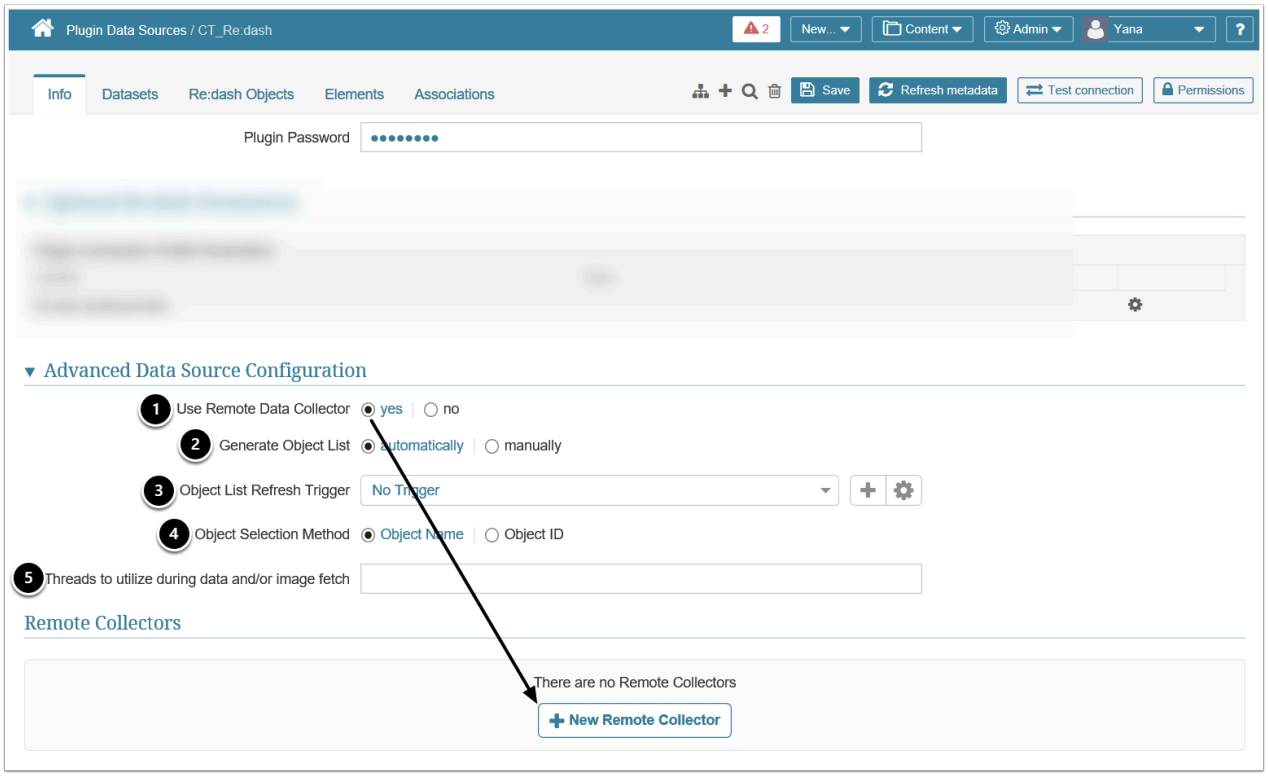
If your connection is successful, you may move on to **Advanced settings**.

3.1. Where to find the API key in Re:dash



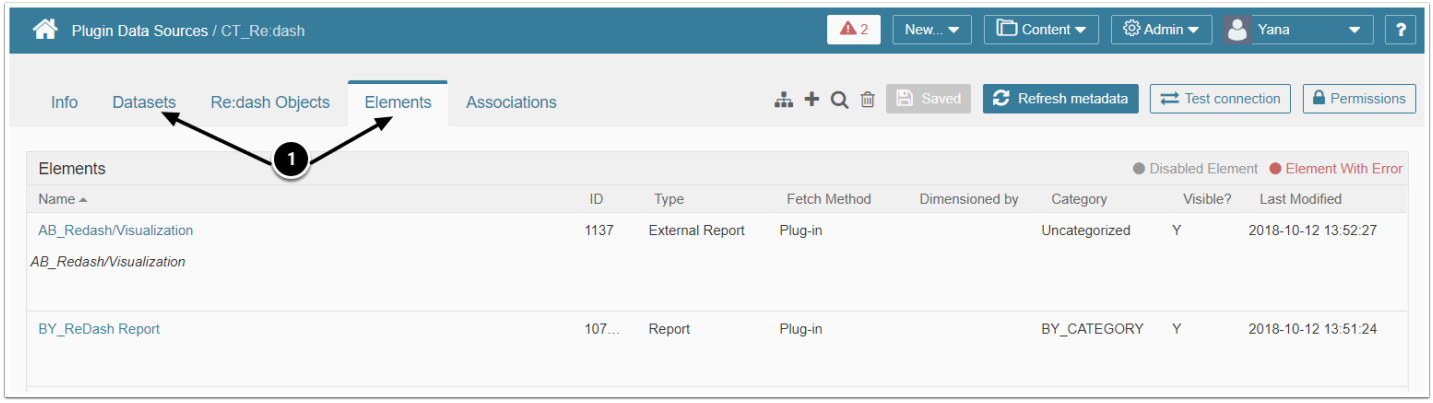
1. Access your profile Settings
2. Find the **API Key** field

4. Advanced Settings



1. **Use Remote Data Collector:** is set to "no" by default. If required, switch to "yes" and add a Remote Data Collector by clicking **[+New Remote Collector]**.
2. **Generate Object List:**
   - **automatically**: just click **Refresh metadata** or use the Trigger function displayed below
   - **manually**: Reports may be added one-by-one or via CSV file
3. **Object List Refresh Trigger:** from the dropdown, select the Trigger that will be used to fetch data from Re:dash.
4. **Object Selection Method:** specify how Re:dash objects will be fetched.
5. Optionally, state the maximum number of concurrent **Threads to utilize during data and/or image fetch** to be used in background processing when the system updates Objects for this Data Source. If you do not specify any value for this setting, batch data collection processing will be single-threaded.

5. Other Settings



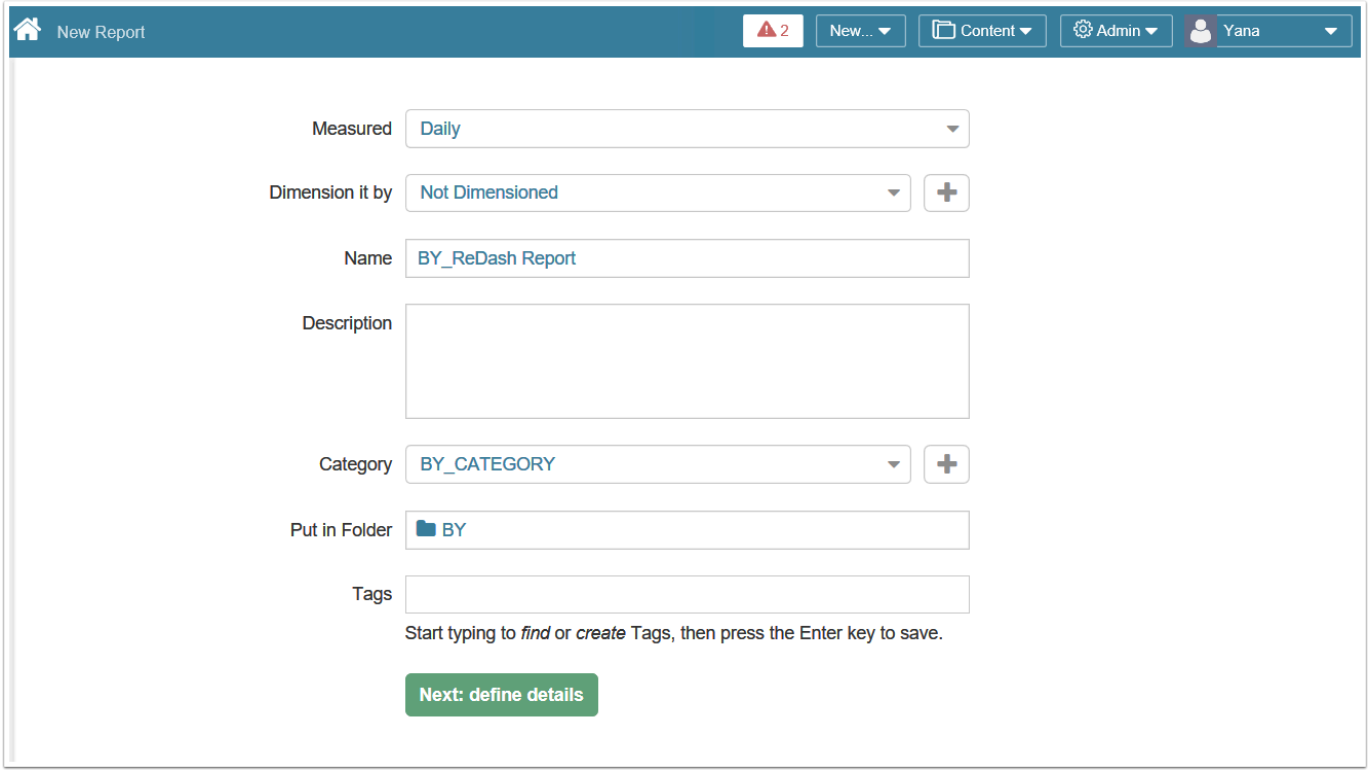
1. You can create Datasets or Elements directly from the respective tabs
2. Click **Permissions** to assign them to Groups or Power Users

What's next?

28.2 How to collect data from Re:dash

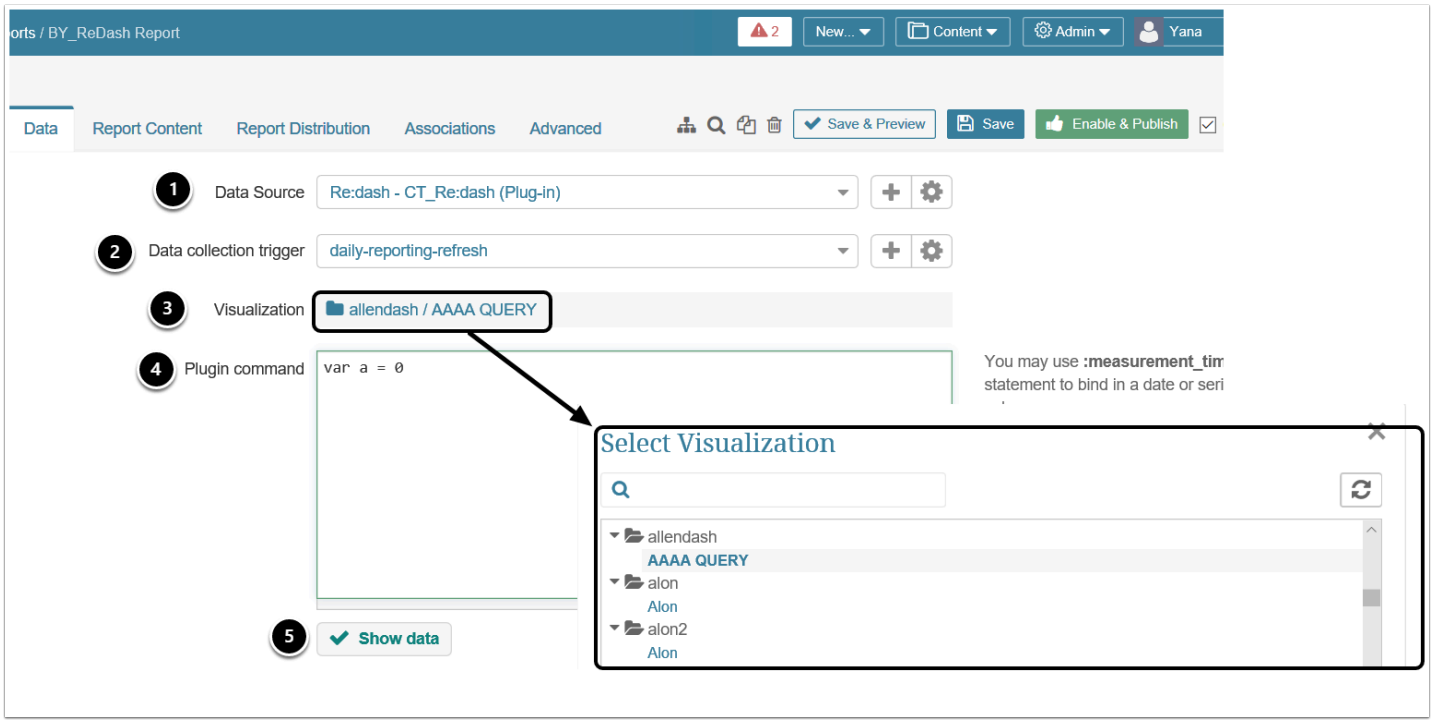
This article will show you how to create a Metric or Report using a Redash report as a data source. It assumes that you have already established connectivity to your Re:dash server.

1. Access New > Report



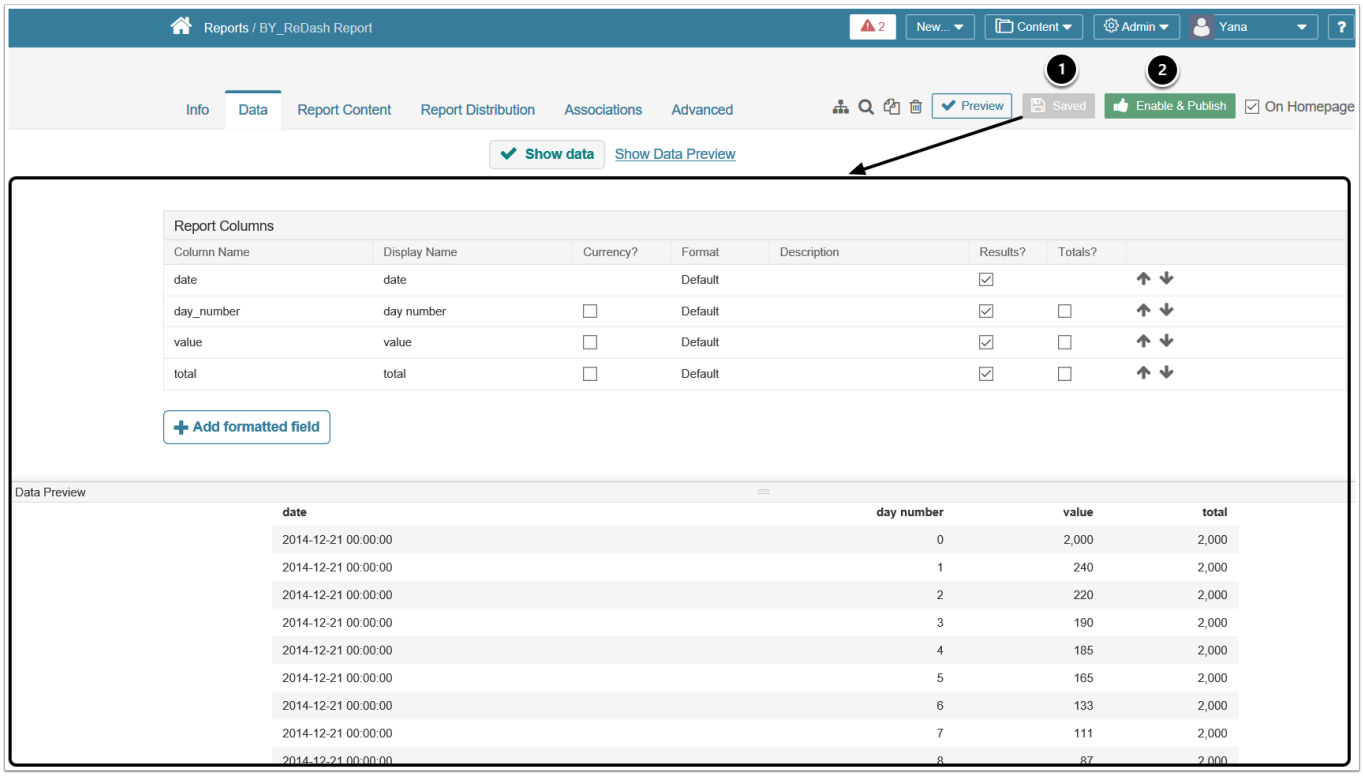
1. Define the Basics for your Report
2. To move on to defining data collection details, click **Next: Define Details**

2. Full Editor displays the Data Collection tab



1. **Data Source**: select the account you have created for Re:dash
2. **Data collection trigger**: Specify the trigger that will be used to collect the data for your Report
3. **Visualization:** select a Visualization that should serve as a basis of a new internal Report
4. Input **Plug-in Command** listing all the data you would like to fetch from Re:dash
5. Once you are ready with you command, click **Show Data**

3. Plug-in command will be validated and Data Collected on Save



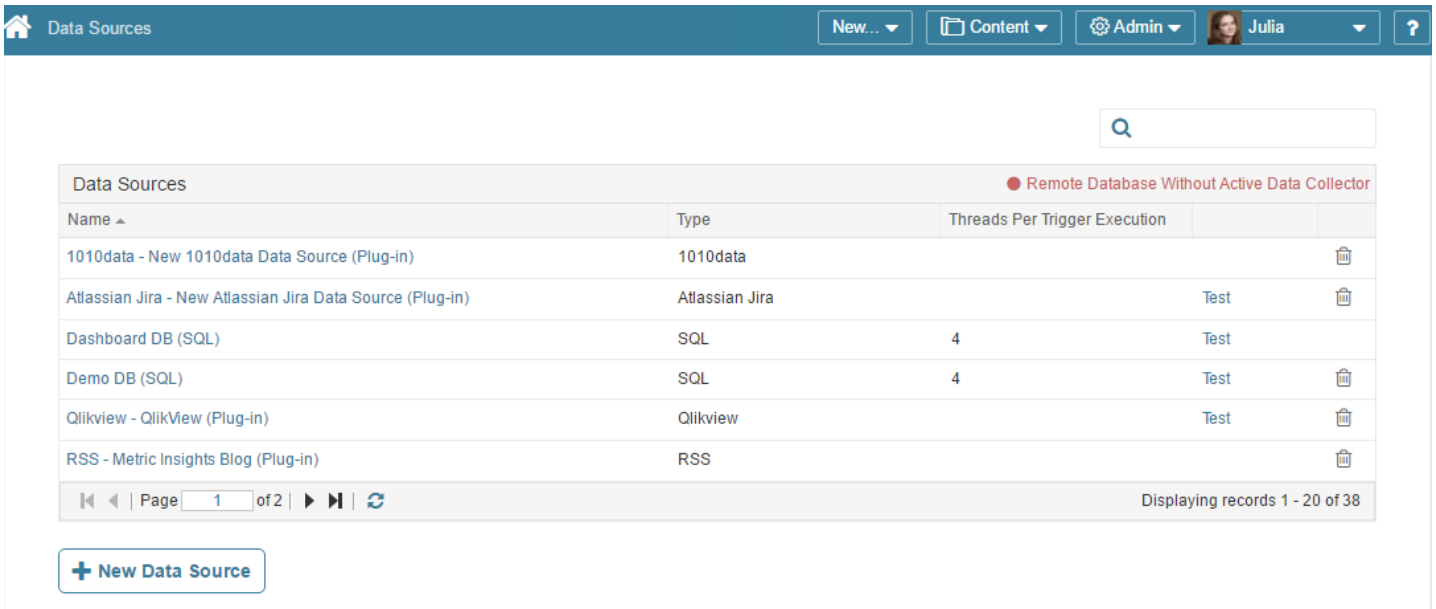
1. If the command is validated successfully, the **Report columns** and **Data Preview** are going to be shown below.
2. At the upper right corner of the screen click **Enable & publish**.

29. Sourcing Data using RSS

29.1 Establish connectivity to RSS

An Administrator can use the process described in this article to create a new **Plug-in Data Source** that is required to allow Elements to fetch data from **RSS** to create a visualization in Metric Insights.

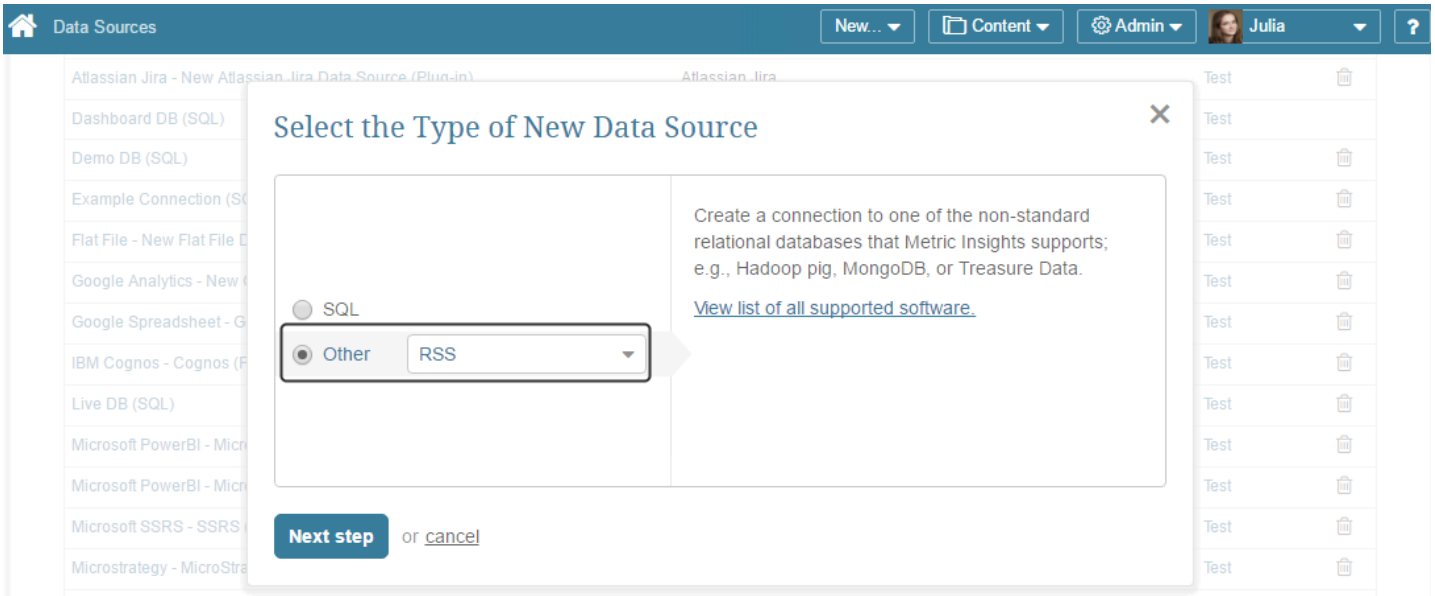
1. Access Admin > Data Sources



At the bottom of the screen click **[+ New Data Source].**

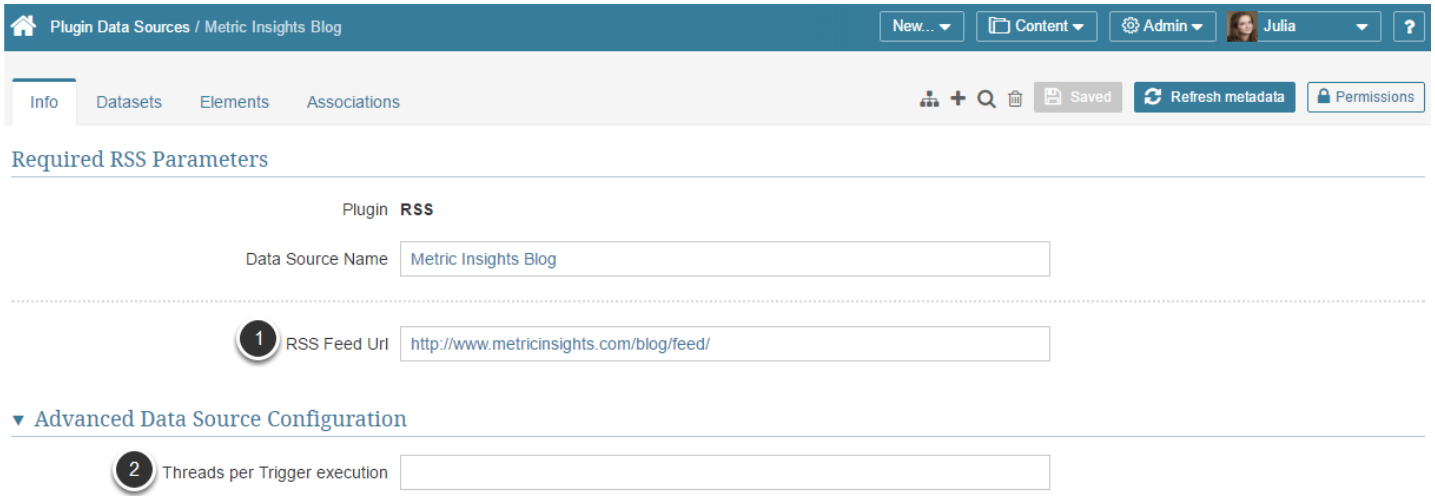
The *Select the Type of New Data Source* pop-up opens.

2. Select "Other" Data Source Type and choose "RSS" from the drop-down list



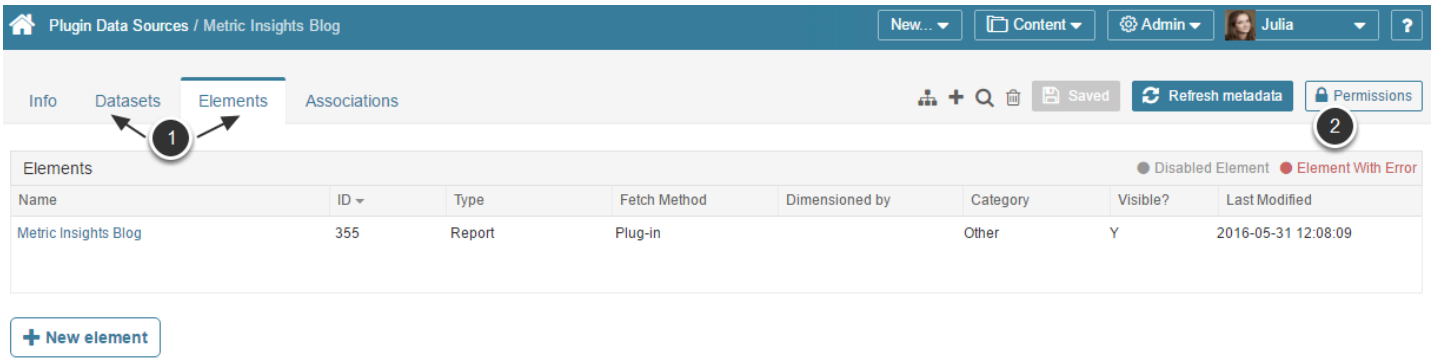
Move to the **Next step**.

3. Provide the RSS Parameters



1. Enter **RSS feed URL**
2. Optionally, enter the number of data collection threads that can be run concurrently for this Data Source

4. Other Settings



1. You can create elements or Datasets directly from the respective tabs
2. Click **Permissions** to assign them to Groups or Power Users

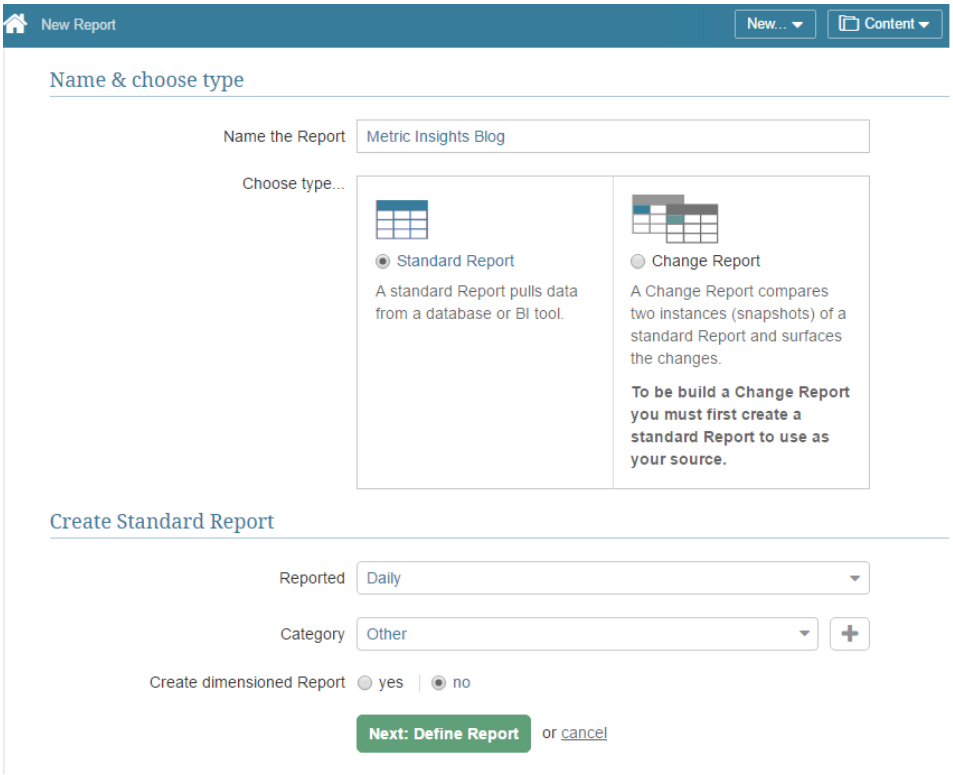
What's next?

[Collect data from RSS]

29.2 How to collect data using RSS plugin

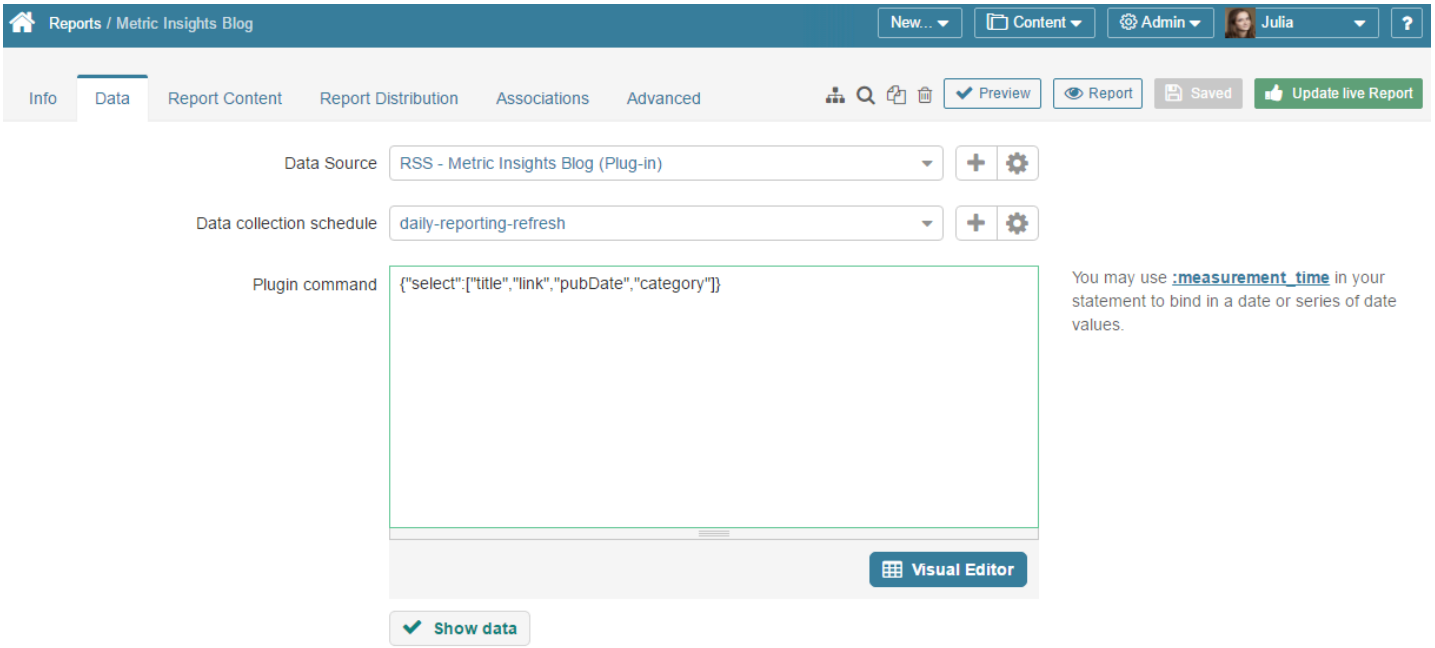
This article will show you how to create a Metric or Report using a RSS as a data source. It assumes that you have already established connectivity to RSS

1. Access New > Report



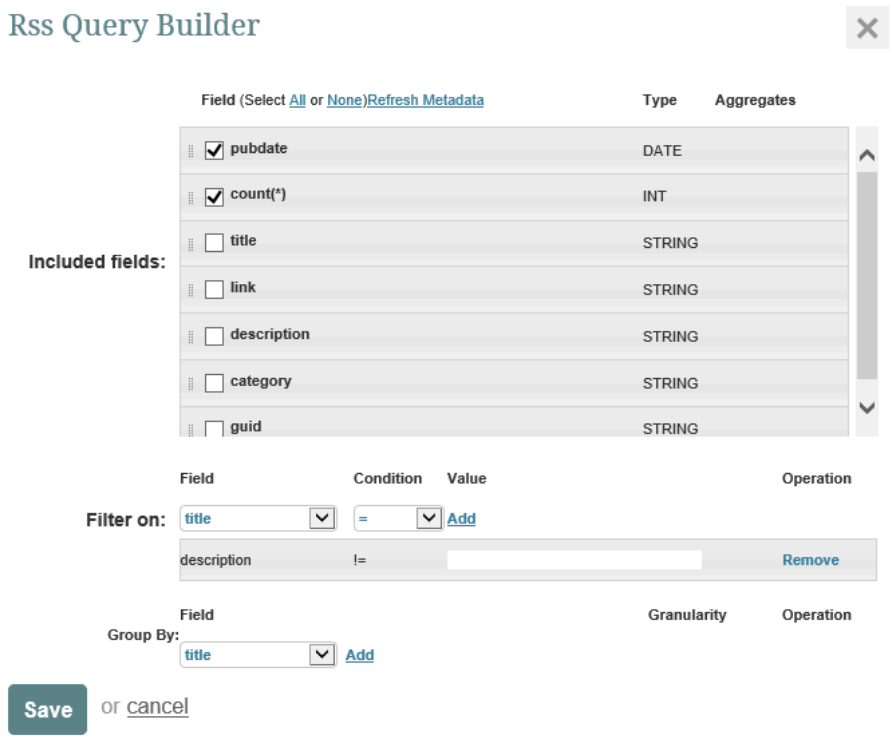
1. **Name the Report**: Define a unique descriptive name of your element
2. **Reported**: choose the measurement interval from the drop-down list
3. **Category**: define a category this element belongs to
4. To move on to defining data collection details, click **Next: Define Report**

2. Full Editor displays the Data Collection tab



1. **Data Source**: select the account you have created for **IBM Cognos**
2. **Data Collection Schedule**: Specify the trigger that will be used to collect the data for your report
3. **Plug-in Command:** Enter a plugin command directly or utilize the visual editor to construct a query
4. Once you are ready with you command, click **Show Data**

2.1. Using the RSS Query Builder.



Select desired fields, and any filters and grouping parameters

Save

2.2. Query Syntax Examples

- Return ALL fields:

```
{
"select":["*"]
}
```

- Select the number of articles that contain a string of text (“value”:”your search term(s) here”) along with the publication date (pubDate):

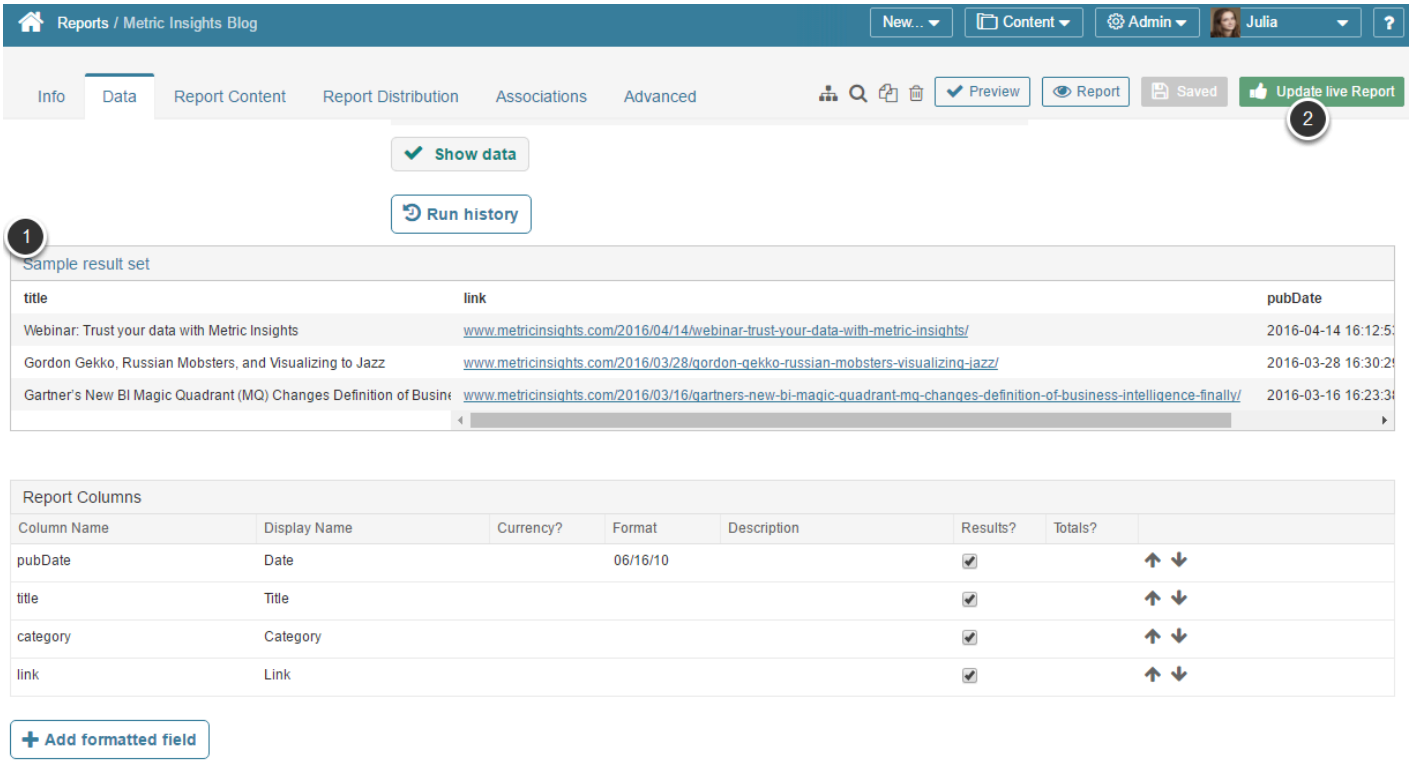
http://rss.nytimes.com/services/xml/rss/nyt/US.xml

The values are not case sensitive, e.g., “Amazon” and “amazon” will return the same result. The "contains","value":" statement will work on multiple terms as an exact phrase match. For the title, “Amazon increases customers worldwide” you can include

```
{"column":"title","condition":"contains","value":"Amazon increases"} but inexact
phrases will not work like:
```

```
{"column":"title","condition":"contains","value":"Amazon customers"}. To do this you
need to create separate conditions:
{
"select":["pubDate","COUNT(*)"],
"where":[{"column":"title","condition":"contains","value":"amazon"},
{"column":"title","condition":"contains","value":"customers"},
{"column":"description","condition":"contains","value":"netflix"}],
"group":["pubDate GRANULARITY DAY"]
}
```

3. Plugin command will be validated and Data Collected



1. If the command is validated successfully, the **Sample Results set** and **Report columns** are going to br shown below.
2. At the upper right corner of the screen click **Update live Report**

30. Sourcing data using Salesforce Reports

30.1 Establish connectivity to Salesforce Reports

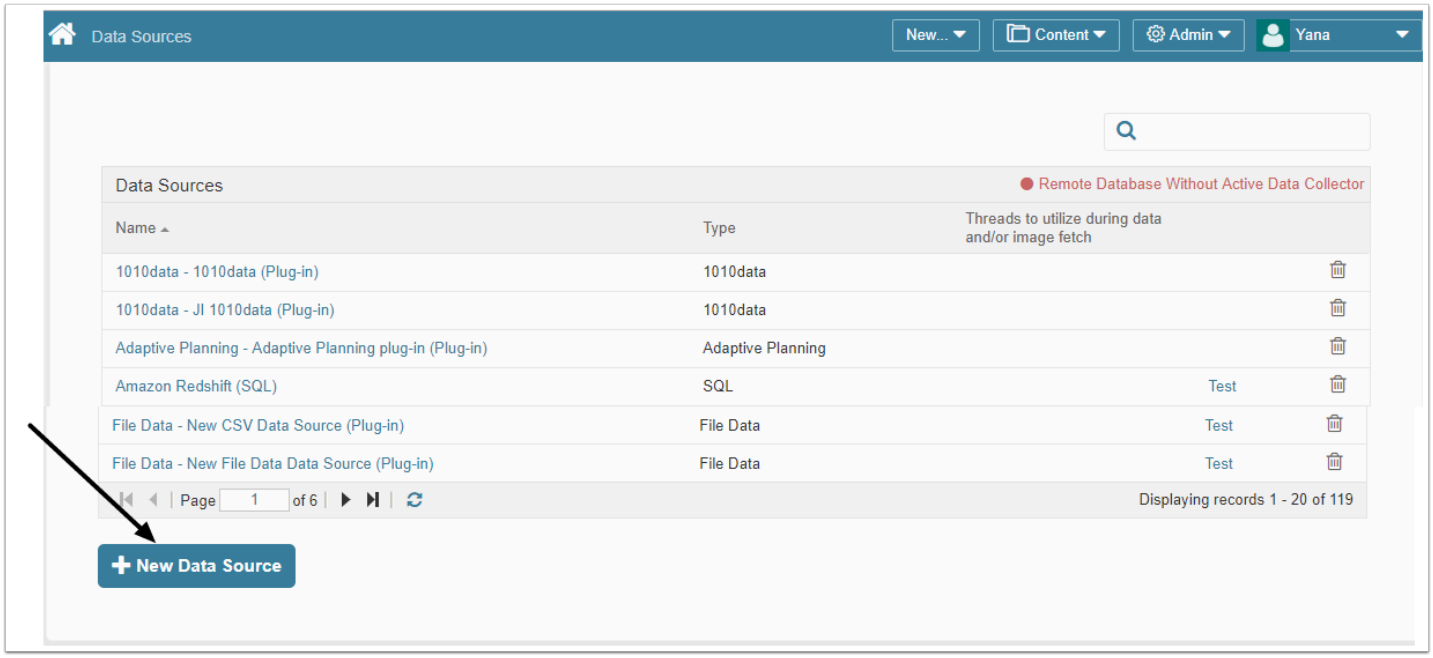
This article describes how to connect to **Salesforce** using Salesforce Reports plugin in order to load data into Datasets and Reports in Metric Insights.

NOTE:

Our plugin currently supports functioning with **Salesforce Classic UI**, so make sure that the User account has Salesforce Classic UI enabled on the Salesforce site.

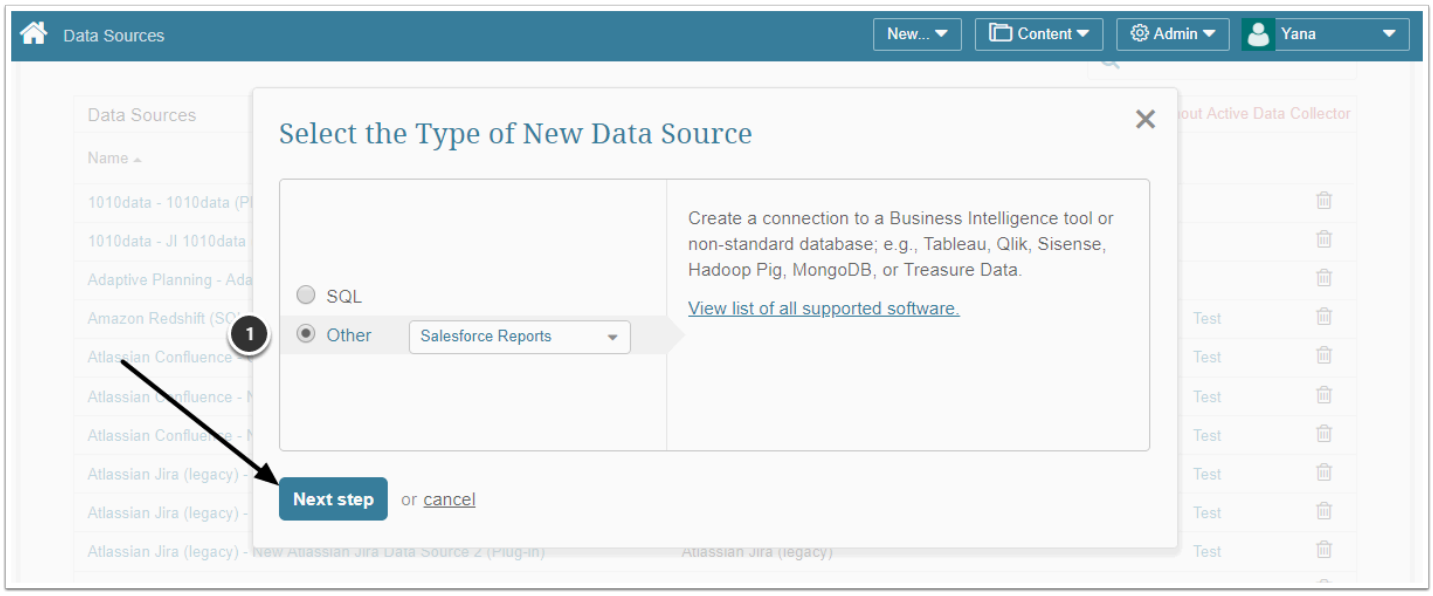
General instructions on setting up data sources based on plugins can be found in Create a New Plugin Data Source.

1. Access Admin > Data Sources



[+ New Data Source].

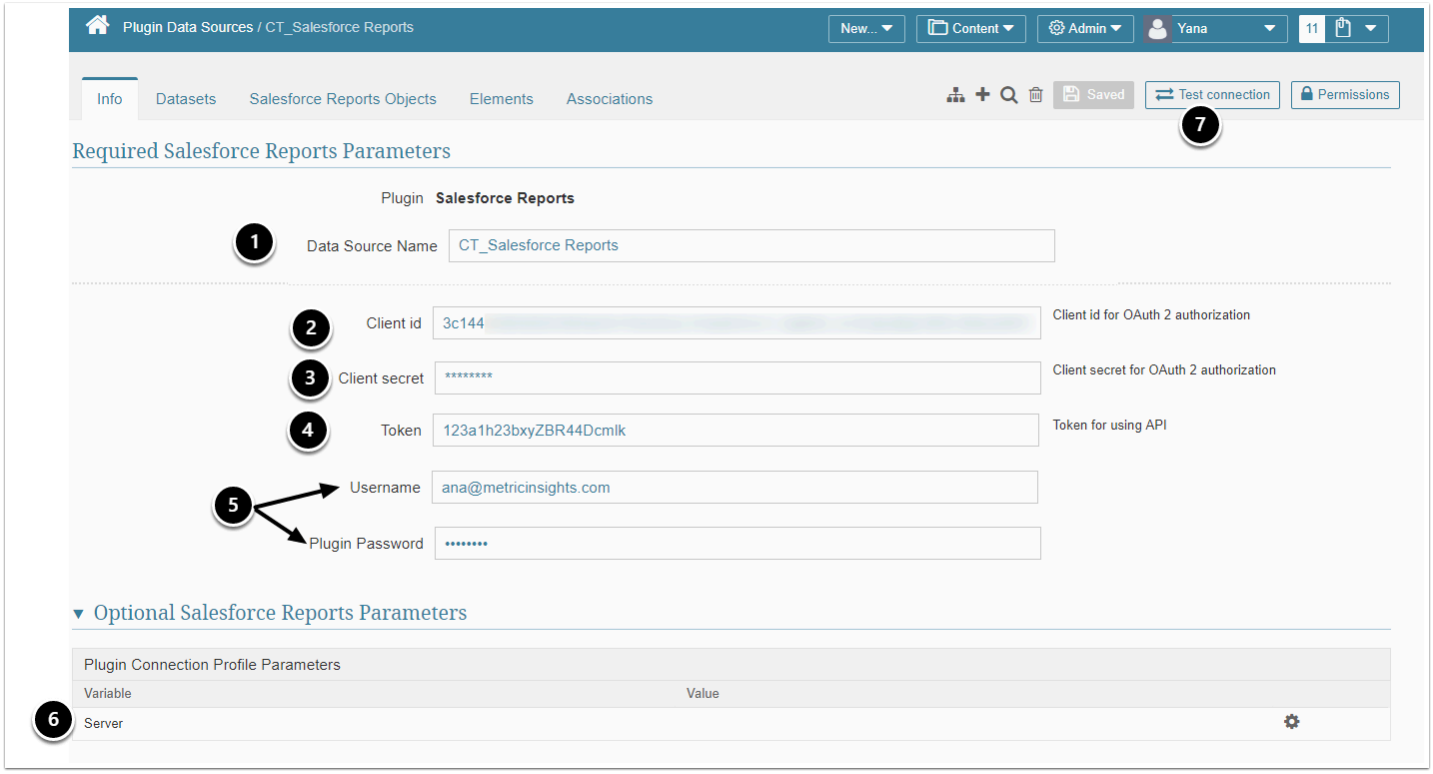
2. "Select the Type of New Data Source" pop-up opens



1. Select "Other" and choose "Salesforce Reports" from the drop-down list

Next step.

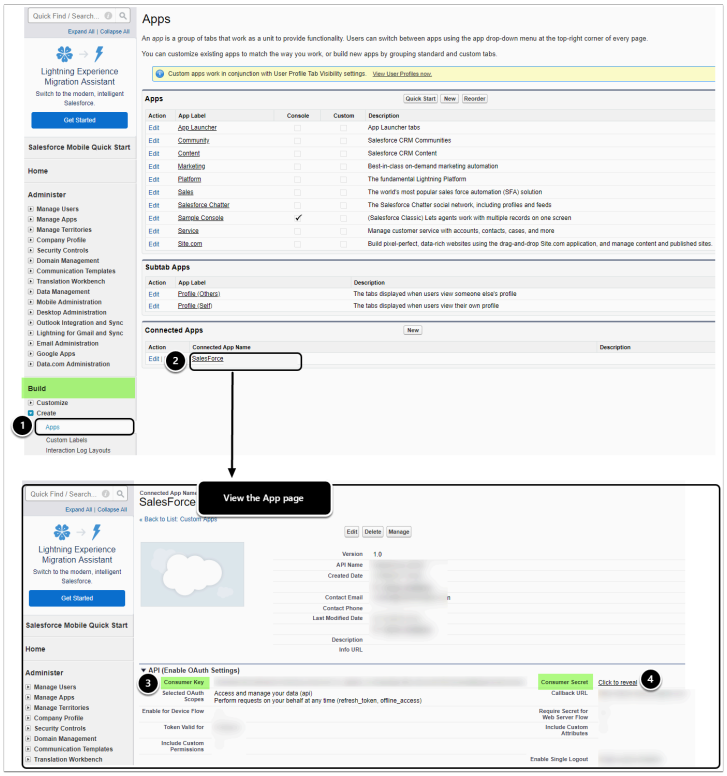
3. Provide Required and Optional Parameters



1. **Data Source Name:** will default but you may modify it

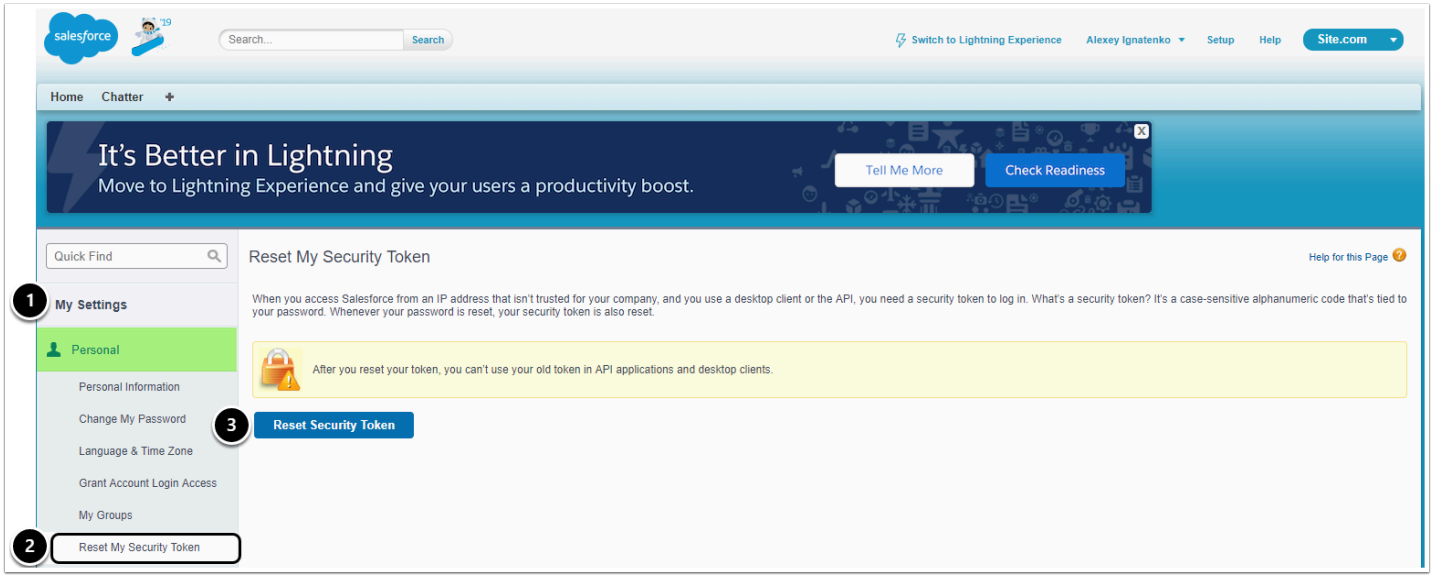
2. **Client id:** enter the API key value ("Consumer Key" in Salesforce) that is used for OAuth 2 authorization of connected Apps
3. **Client secret:** enter the API key value ("Consumer Secret" in Salesforce) that is used for OAuth 2 authorization of connected Apps
4. **Token:** enter the security token provided by Salesforce
5. **Username / Password:** note that your **Username** must be in the same format that your Salesforce server uses for authentication
6. Optionally, specify the name of your custom Salesforce **Server** that you want to use instead of a default
7. **Test connection** (this will also save your entries)

3.1. Where to find Consumer Key and Consumer Secret in Salesforce (Classic UI)



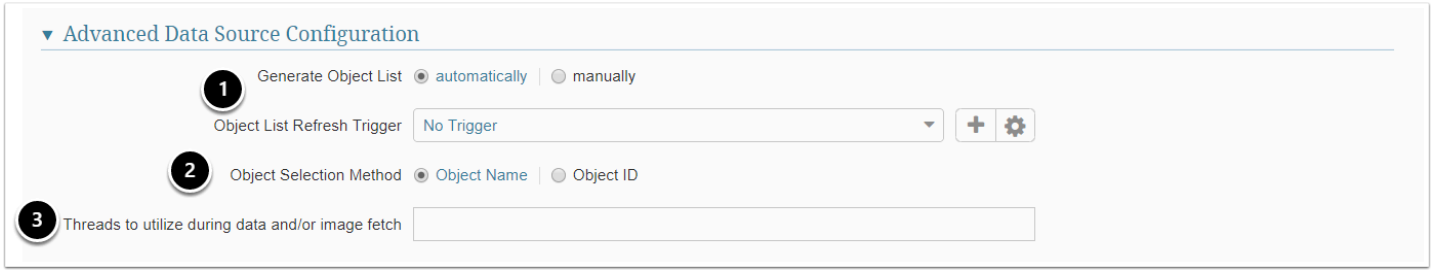
1. Go to **Build > Create > Apps**
2. Connected **Apps > Open** the App link
3. In the API section, copy **Consumer Key**
4. Click the link to reveal and copy **Consumer Secret**

3.2. How to obtain a security Token in Salesforce



1. Go to **My Settings**
2. Under Personal, select **Reset My Security Token**
3. Click **[Reset Security Token]**
    - Clicking the button invalidates your existing token. After resetting your token, it will sent to your email

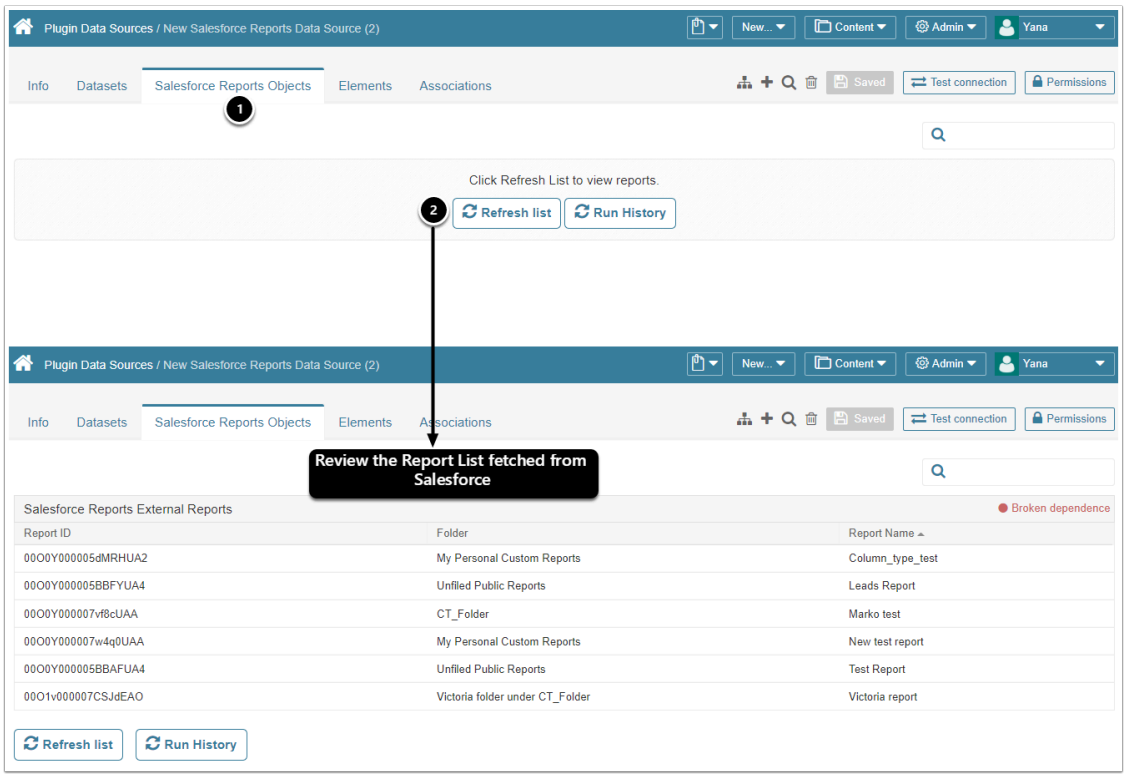
4. Advanced Configuration



1. **Generate Object List**: This setting influences options available in the ***Salesforce Reports Objects*** tab:
    - **automatically**:
        - In the ***Adobe Analytics Objects*** *tab* click **Refresh list** to refresh/add all Objects currently in the BI system
        - [New in 5.3.2] **Object List Refresh Trigger** will appear allowing you to schedule the Refresh function to run automatically (Optional)
    - **manually**:
        - Reports must be added one-by-one or via CSV file in the ***Salesforce Reports Objects*** *tab*

2. **Object Selection Method:** specify how Salesforce Reports will be fetched
3. Optionally, state the maximum number of concurrent **Threads to utilize during data and/or image fetch** to be used in background processing when the system updates Reports for this Data Source
    1. If you do not specify any value for this setting, batch data collection processing will be single-threaded

5. Obtain a list of External Reports



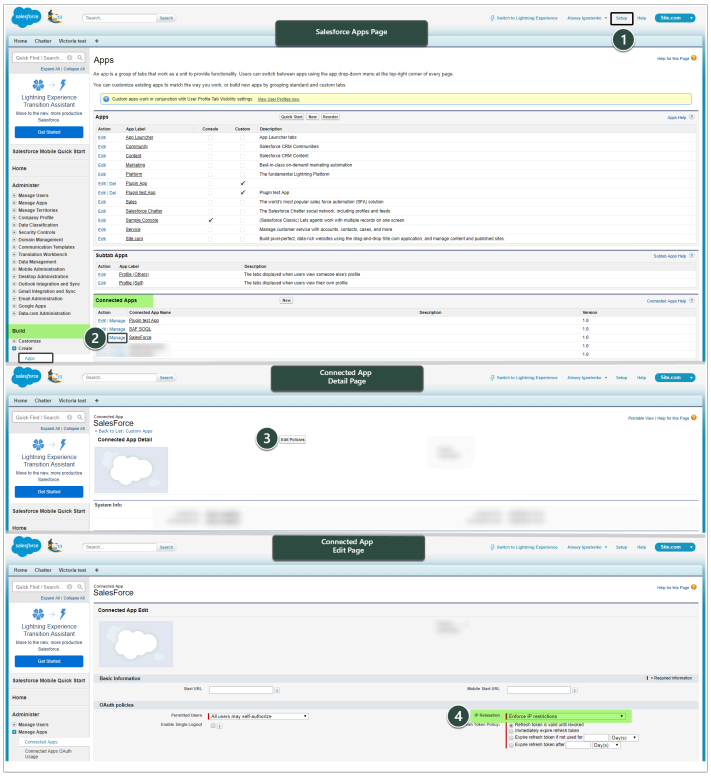
1. Go to **Salesforce Reports Objects** tab
2. To obtain a list of External Reports, click the **[Refresh list]** button

5.1. Troubleshoot 4xx errors when getting External Reports List

A comprehensive list of Status Codes and Error Responses is provided by Salesforce REST API Developer Guide.

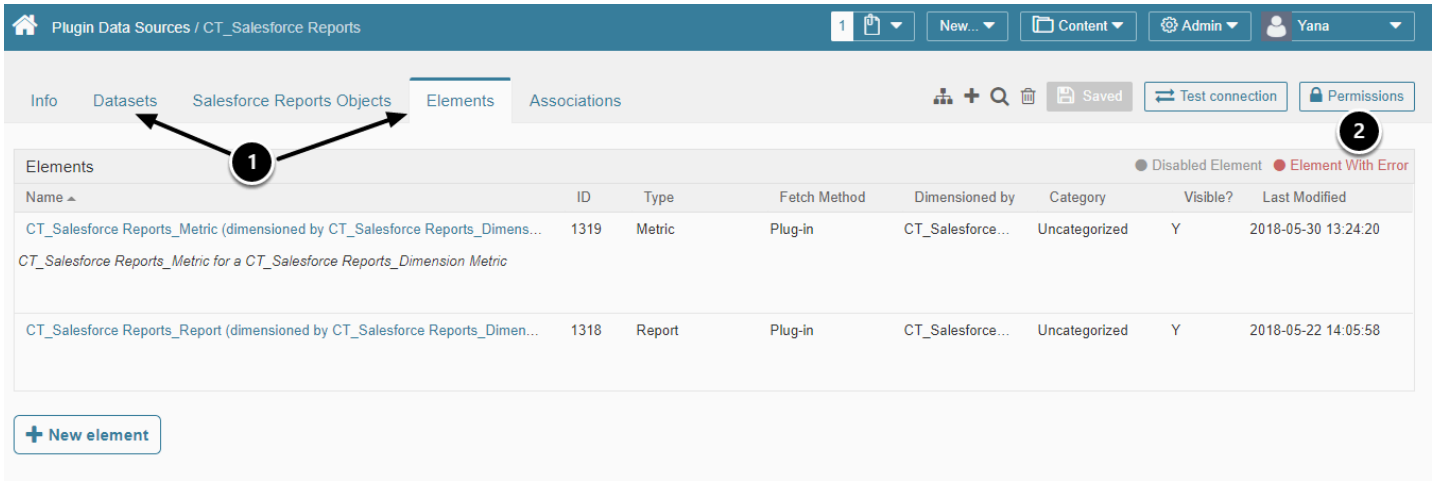
- When trying to establish connection with Salesforce via the corresponding plugin connection profile, you might get 4xx client errors caused by security constraints.
- In order to alleviate the problem of authorization, you may need to relax IP restrictions in Salesforce.

For reference, see instructions below.



1. To access ***Connected Apps***:
   - From the Homepage, go to **Setup**
   - In the Left Side menu, find the **Build** section
   - Expand **Create** to reveal the list of **Apps**
   - Move down to **Connected Apps**
2. Next to the *Connected App name,* click **[Manage]**
3. On the *Detail page*, click **[Edit Policies]**
4. On the *Edit page*, for IP Relaxation select **Relax IP restrictions** under "OAuth policies"

6. Other Settings



1. You can create Datasets or Elements directly from the respective tabs
2. Click **Permissions** to assign use of this Plugin to Groups or Power Users

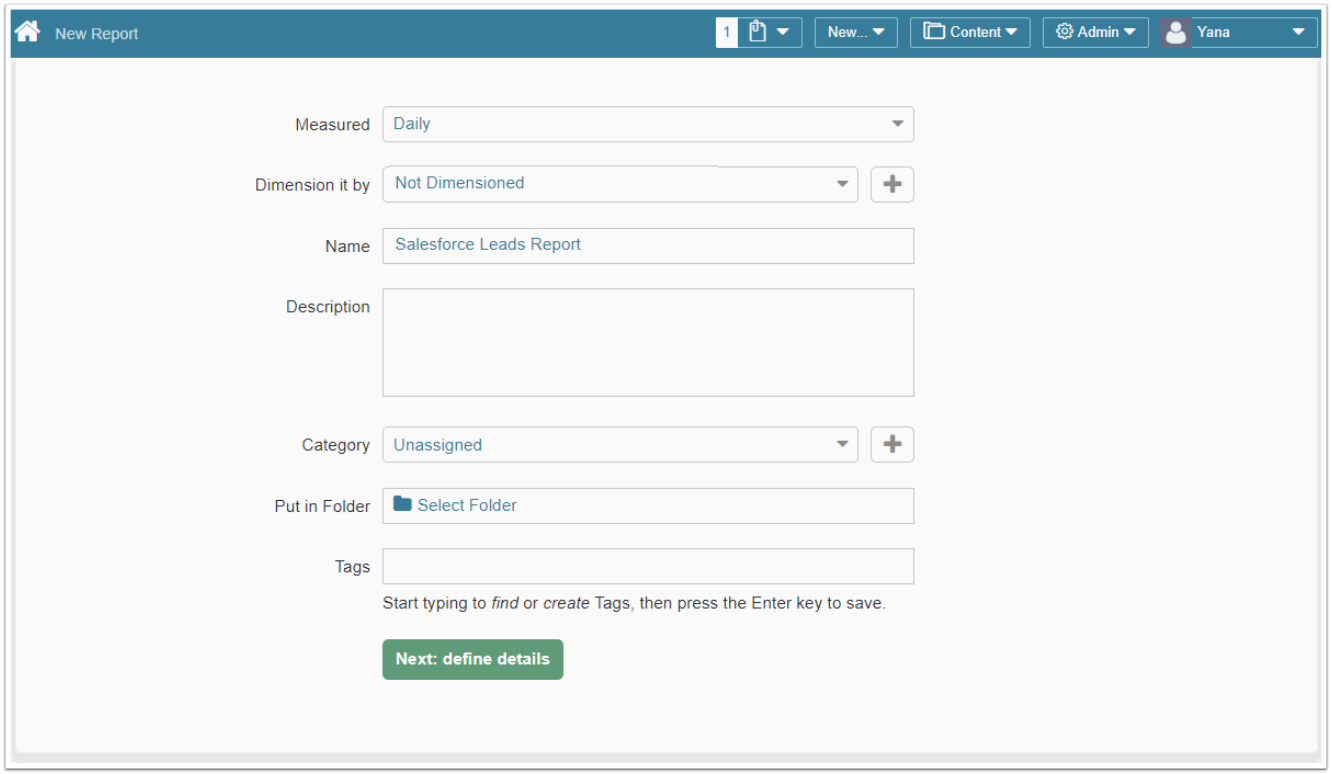
What's next?

[How to collect data from Salesforce]

30.2 How to collect data using Salesforce Reports

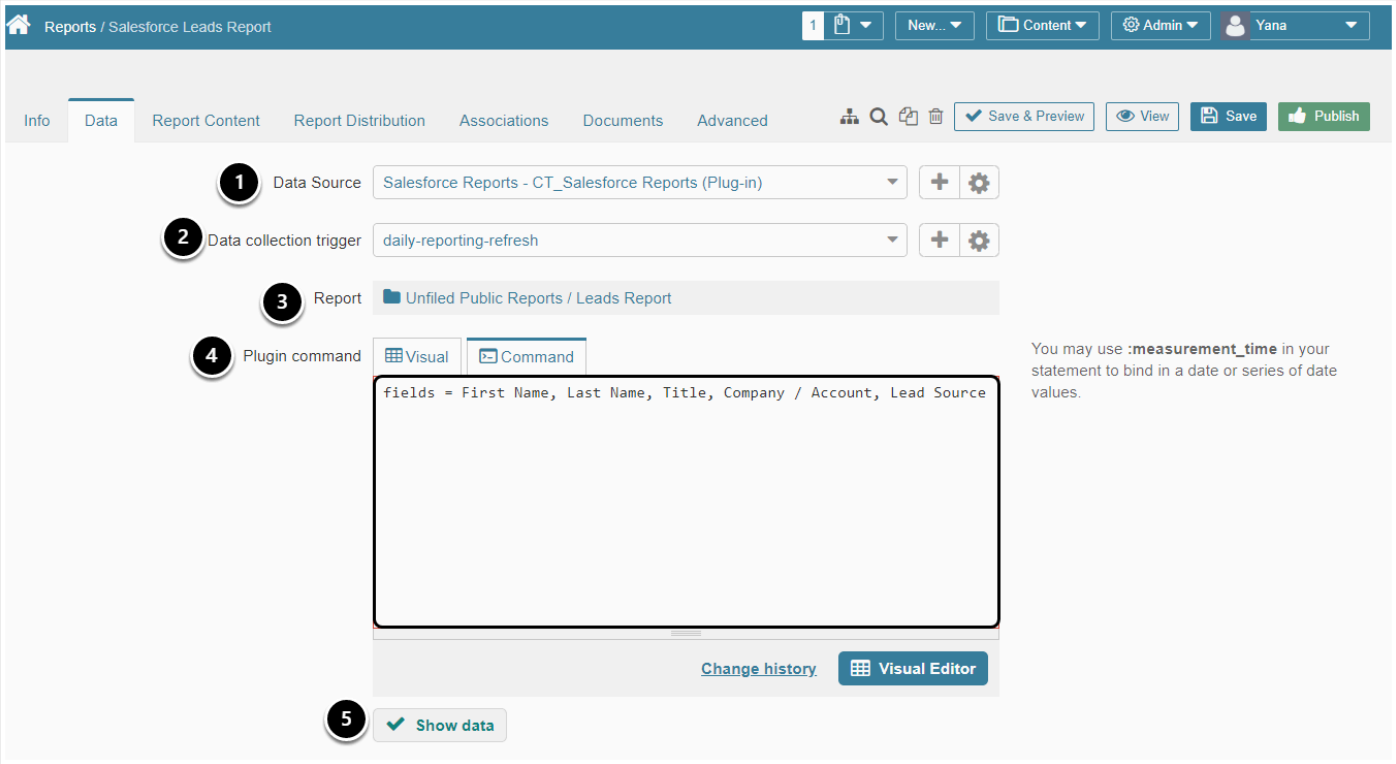
This article will show you how to create a Metric or Report using a **Salesforce** report as a data source. It assumes that you have already established connectivity to your **Salesforce** server.

1. Access New > Report



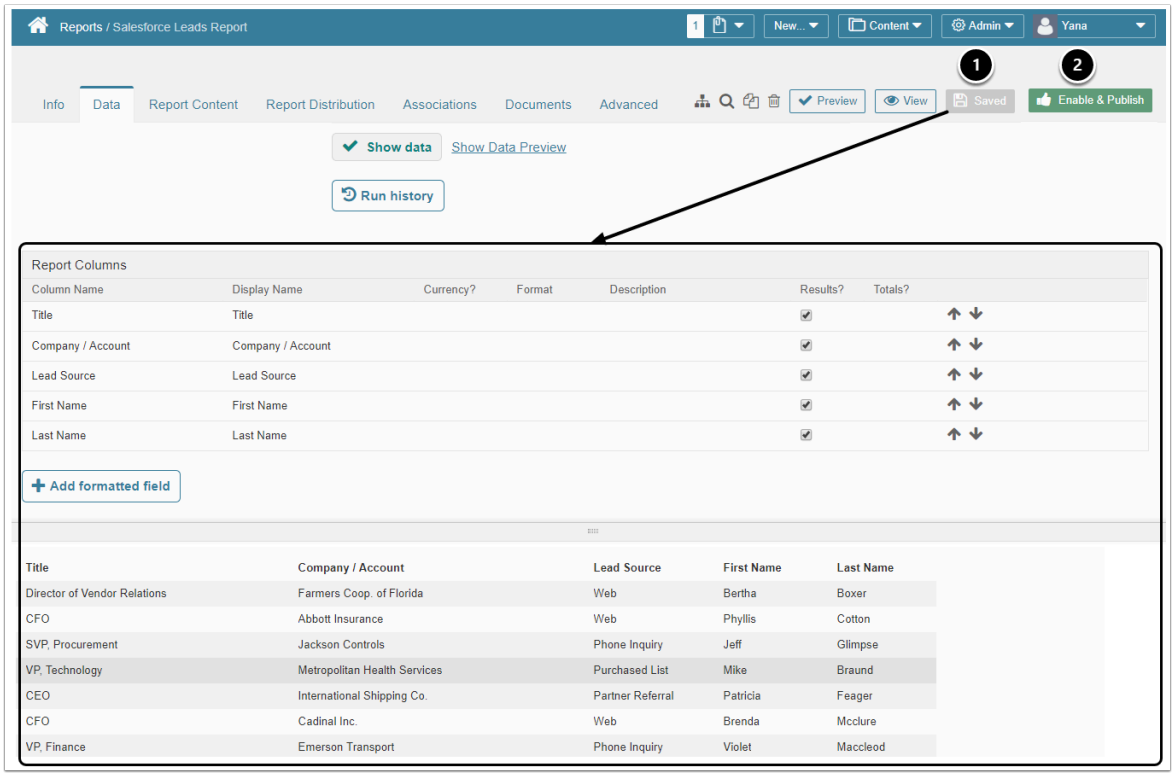
1. Define the Basics for your Report
2. To move on to defining data collection details, click **Next: Define Details**

2. Full Editor displays the Data Collection tab



1. **Data Source**: select the connection profile you have created for the Salesforce Reports plugin
2. **Data collection trigger**: specify the Trigger that will be used to collect data for your Report
3. **Salesforce Report**: select a Salesforce Report that should serve as a basis of a new internal Report
4. Input an MIQL **Plugin Command** listing all the data you would like to fetch from Salesforce
   - Alternatively, use the **Visual Editor**
5. Once you are ready with you command, click **Show Data**

3. Plugin command will be validated and Data Collected on Save



1. If the command is validated successfully, the **Report columns** and **Data Preview** are going to be shown below
2. At the upper right corner of the screen click **Enable & publish**

31. Sourcing data using Salesforce SOQL

31.1 Salesforce Reports vs Salesforce SOQL: key distinctions

There are two ways to get data from Salesforce:

1. using Salesforce Reports plugin
2. using Salesforce SOQL plugin

The main differences are detailed below.

1. Salesforce Reports

Salesforce Reports plugin allows you to create elements using the Visual editor.

The main distinctions of using Salesforce Reports are:

- You don't need to know SOQL (Salesforce Object Query Language) to create plugin commands
- You can use Visual editor
- BUT, you need to have existing reports in Salesforce as a source

See more information here

2. Salesforce SOQL

Salesforce SOQL plugin allows you to create elements by adding plugin commands manually.

The main distinctions of using Salesforce SOQL are:

- You can easily create new element by typing in your plugin command
- You don't need additional setup before creating a new element
- BUT, you have to know SOQL (Salesforce Object Query Language) to create plugin commands

See more information here

31.2 Setting up Salesforce OAuth

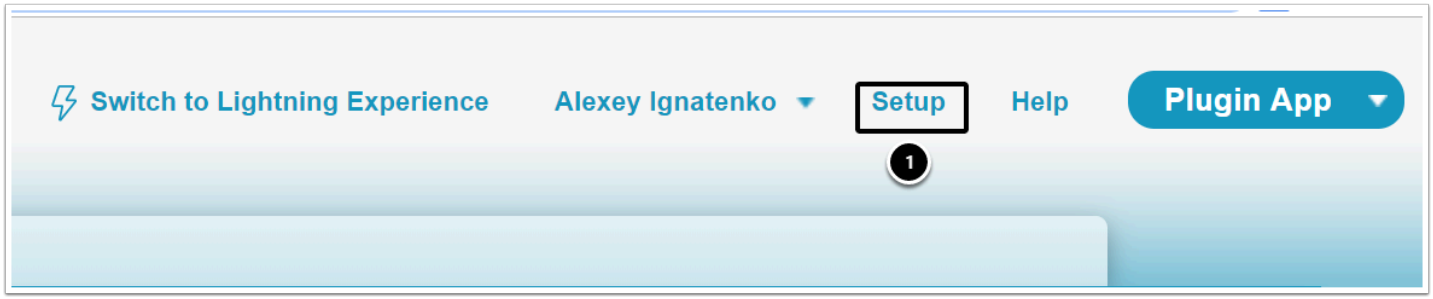
In order to use the **Salesforce SOQL** plugin, you will need to log into the Salesforce browser interface and setup an **OAuth** connection. This article summarizes the key steps.

- Detailed instructions can be found in the Salesforce support area. Refer to Creating a Connected App for more information.

NOTE

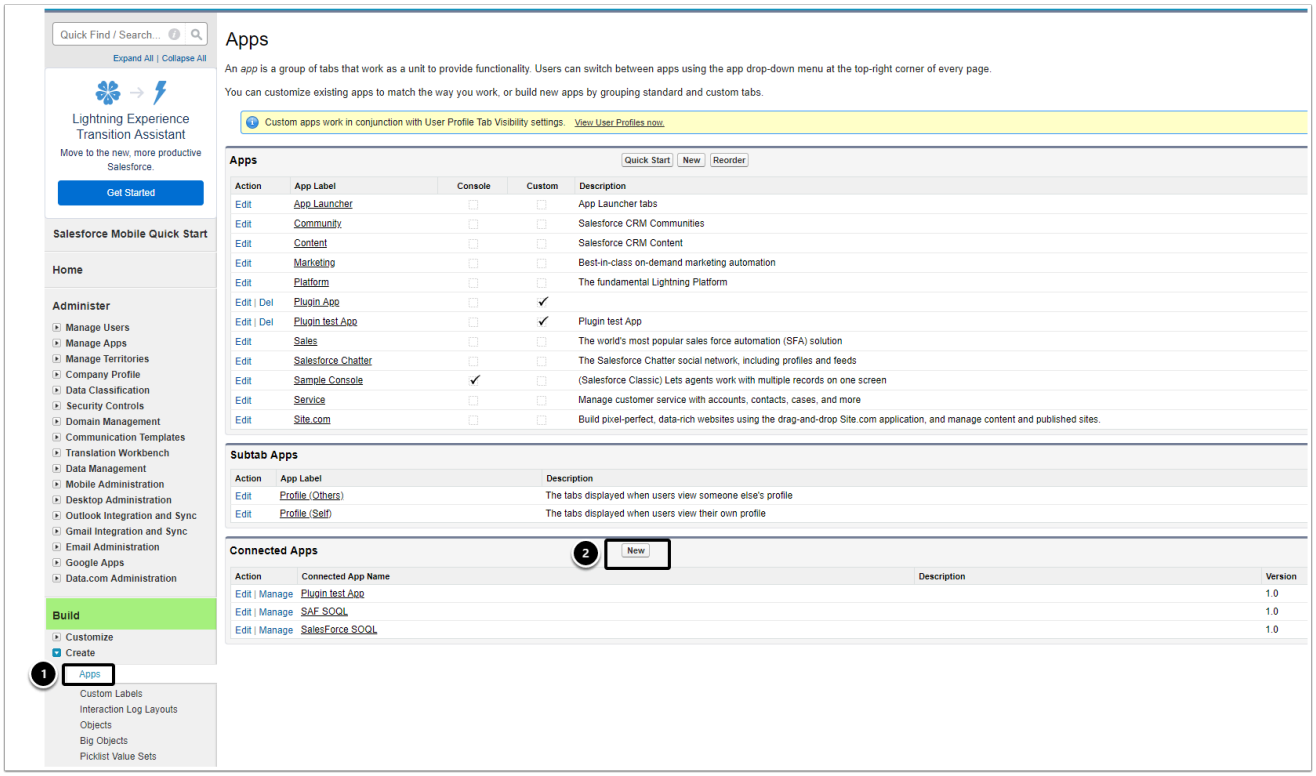
- The "regular" Salesforce plugin does not require an *OAuth connection*.
- You can set up a connection using that plugin simply by inputting your Salesforce username and password, plus your *"security token"* (which is essentially an extra password, obtainable from the Salesforce browser interface).

1. Log into Salesforce



1. Go to the "Setup" menu

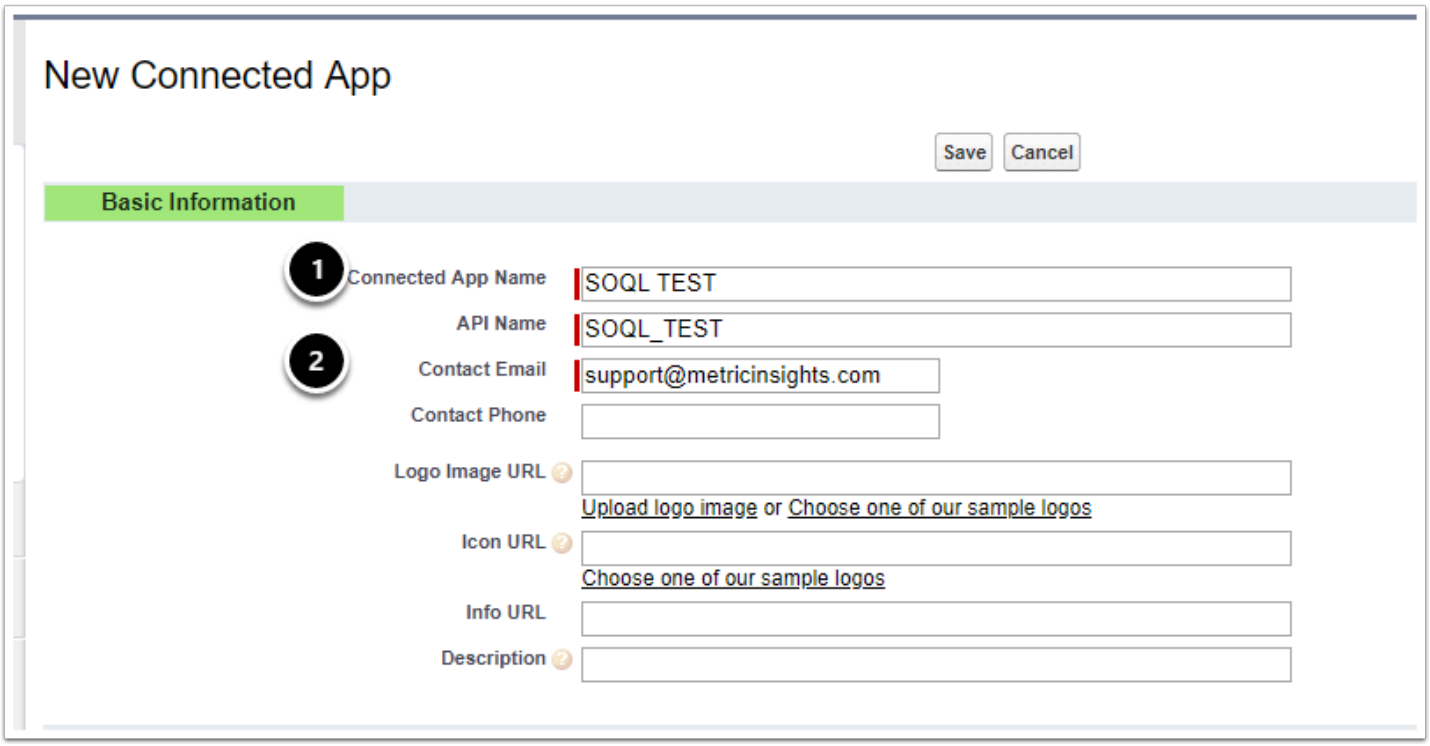
2. Go to Build > Apps > New



Use the ***Build Section*** in the Left Sidebar:

1. Choose **"Apps"** in the *Create* sub-section
2. Click the **"New"** button in the *Connected Apps section*

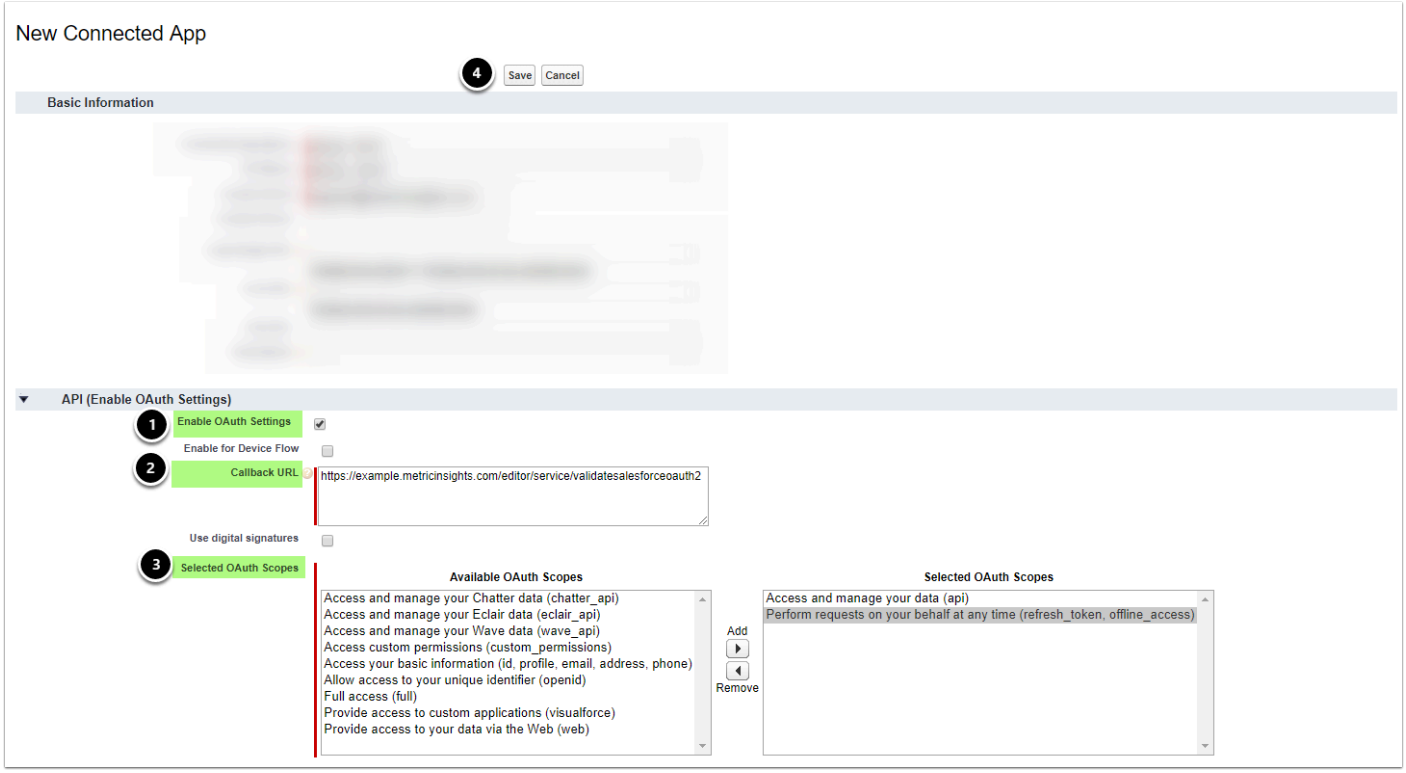
3. Define the Basics



1. Specify **Connected App Name**
   - Enter the API name used when referring to your app from a program
2. Enter the **Contact Email** for Salesforce to use when contacting you or your support team. This address isn't given to Salesforce admins who install the app

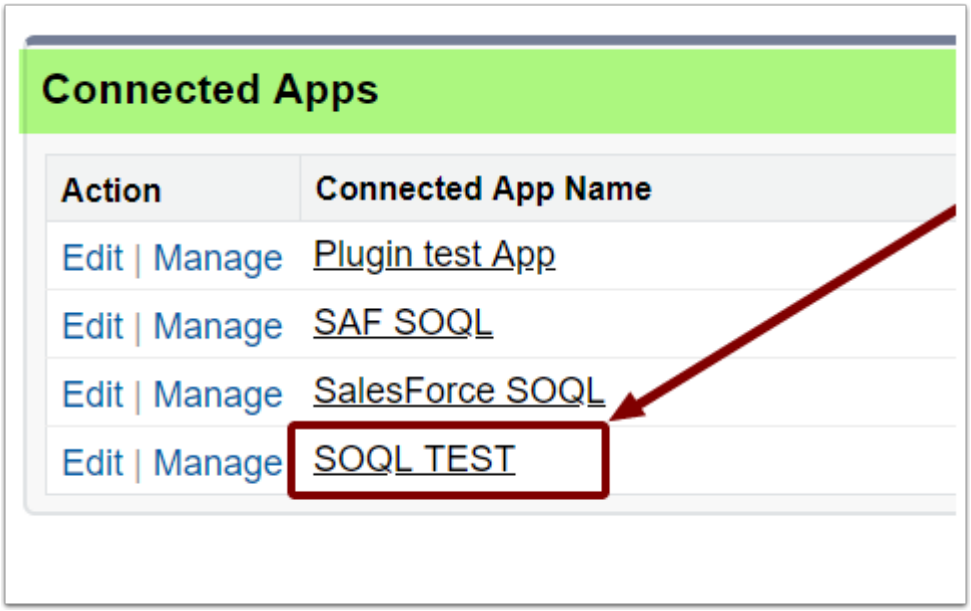
For more information, refer to [Create a Connected App]

4. Enable OAuth Settings



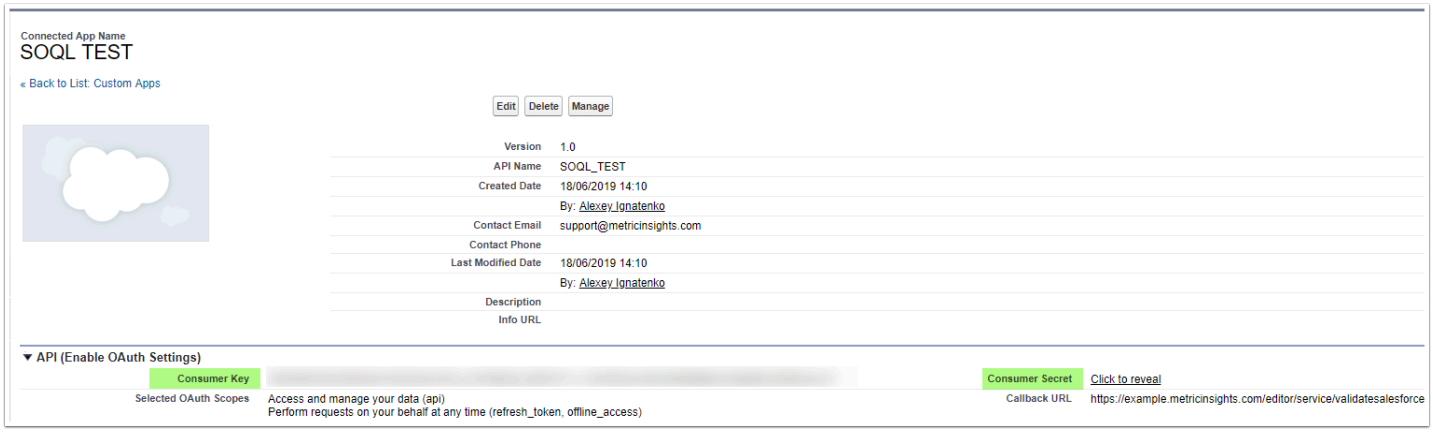
1. Choose "**Enable OAuth Settings"**
2. Provide a **Callback URL** (endpoint) in this form where `example.metricinsights.com` is the hostname for your Metric Insights server
3. **Selected OAuth Scopes** should include:
   - Access and manage your data (API)
   - Perform requests on your behalf at any time (refresh_token, offline_access): *this will enable Metric Insights to refresh Your Token*
4. Click **[Save]**

5. Connected Apps > your new App



Click the **Connected App Name** to open its configuration page containing *Consumer Key (ID)* and *Consumer Secret*

6. Client ID and Client Secret



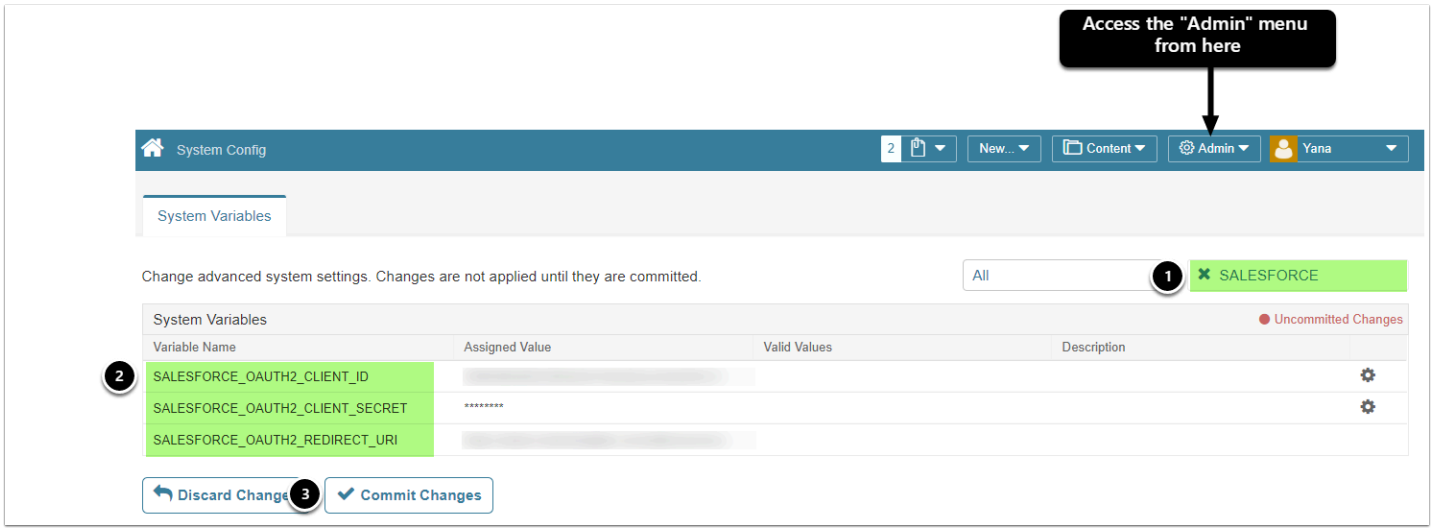
You should see your *Consumer Key (ID)* and your *Consumer Secret* in the middle of the page.

```
 ('SALESFORCE_OAUTH2_CLIENT_ID', '3MVxxxxxC92_J.LmfhKJ0Z_...
5JydLTMG5bh6eosdfWExxxbc_2FWeBclda5gD6');
 ('SALESFORCE_OAUTH2_CLIENT_SECRET', 'nnnnnn..');
```

Note: These are client specific.

> ⚠ You need to add your *Salesforce Secret* and *Client ID* to the **Config variables**. For details, see Step 7 below.

7. Configure Salesorce Variables in Metric Insights



Access Config Variables via the **Admin menu:** ***Admin*** **>** ***Utilities*** **>** ***System config*** (For details, see Setting the Configuration Variables):

1. Filter config settings for Salesforce
2. Modify the variables to include "Salesforce Secret" and "Client ID"
3. Commit Changes

31.3 Establish connectivity to Salesforce SOQL

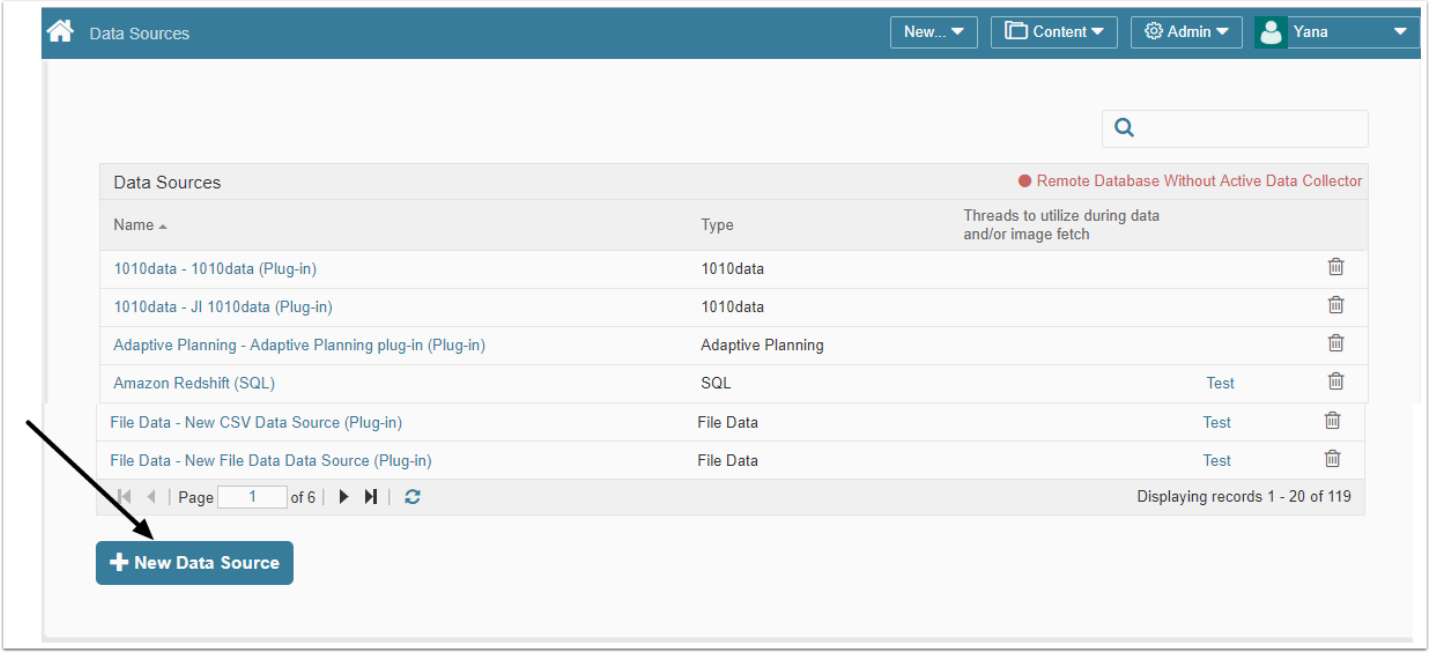
This article describes how to connect to **Salesforce** using Salesforce SOQL plugin in order to load data into Datasets and Reports in Metric Insights.

PREREQUISITES:

- **Setting up Salesforce OAuth**

General instructions on setting up data sources based on plugins can be found in Create a New Plugin Data Source.

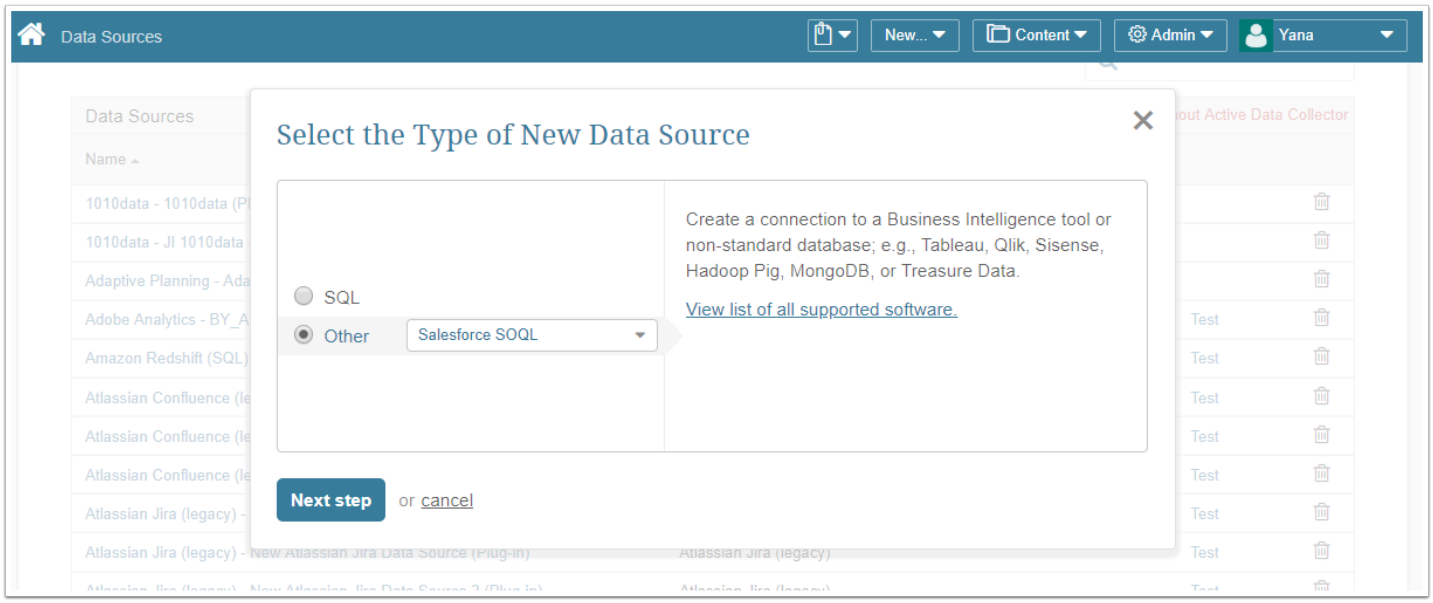
1. Access Admin > Data Sources



[+ New Data Source].

The *Select the Type of New Data Source* pop-up opens.

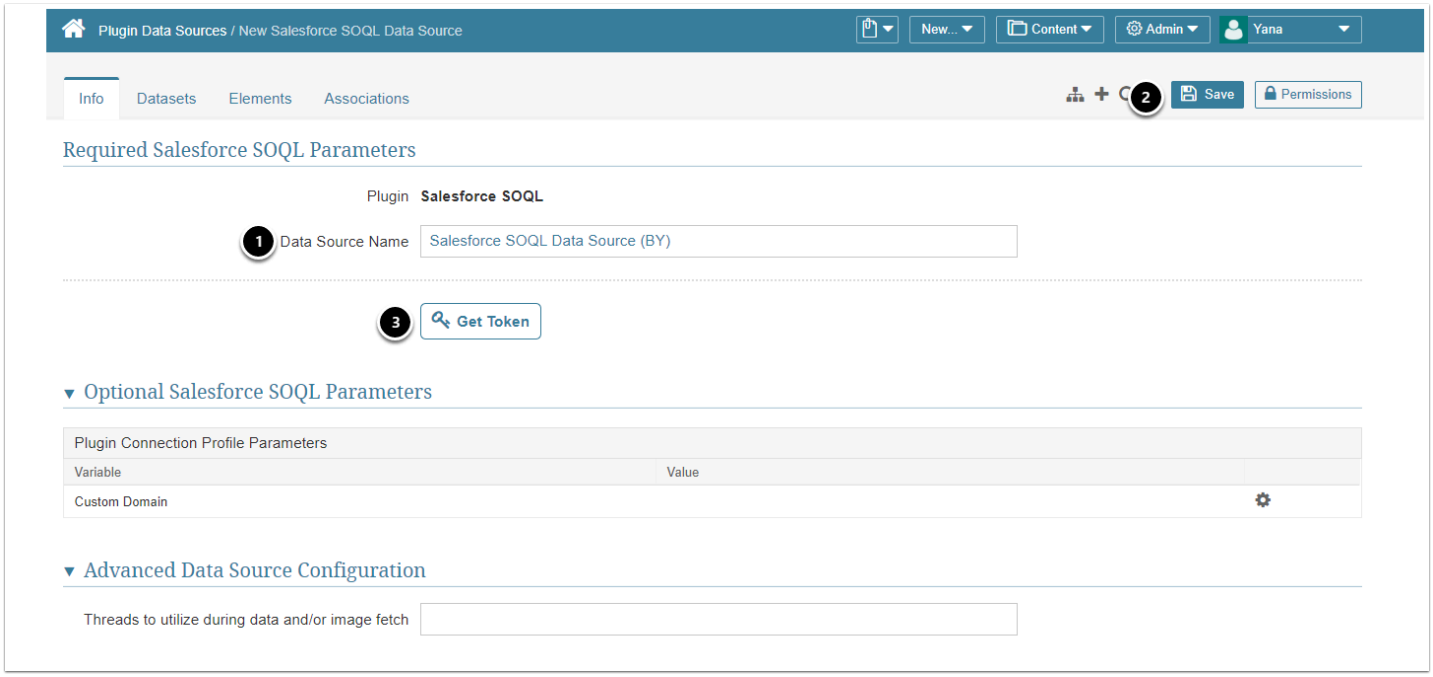
2. Select the Type of New Data Source



1. Select "Other" and choose "Salesforce SOQL" from the drop-down list

Next step

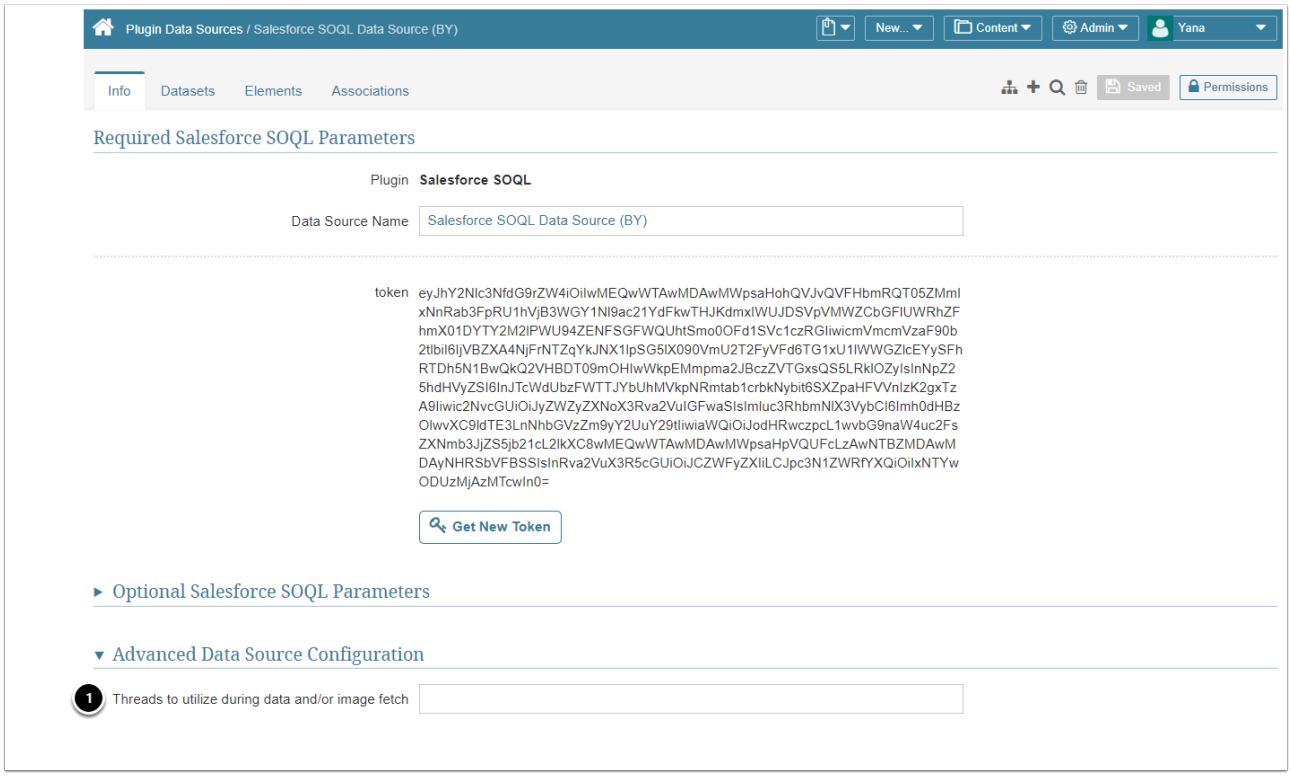
3. Configure the Plugin Parameters



1. **Data Source Name** is defaulted but you may modify it
2. **[Save]** your settings before getting the Token

3. **[Get Token]:** you will be prompted to authenticate yourself by logging into your Salesforce account

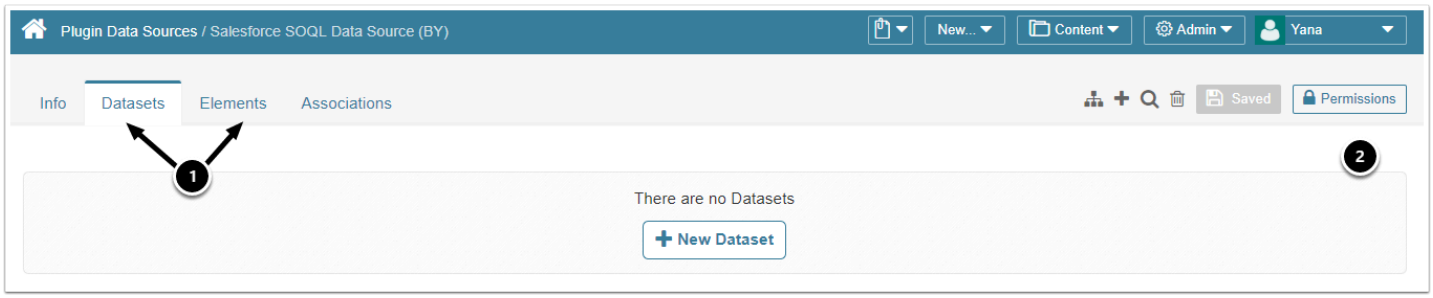
4. Advanced Configuration (optional)



Having obtained the **Token**, you can configure **Advanced Settings:**

1. Optionally, state the maximum number of concurrent **Threads to utilize during data and/or image fetch** to be used in background processing when the system updates Reports for this Data Source
   - If you do not specify any value for this setting, batch data collection processing will be single-threaded

5. Other Settings

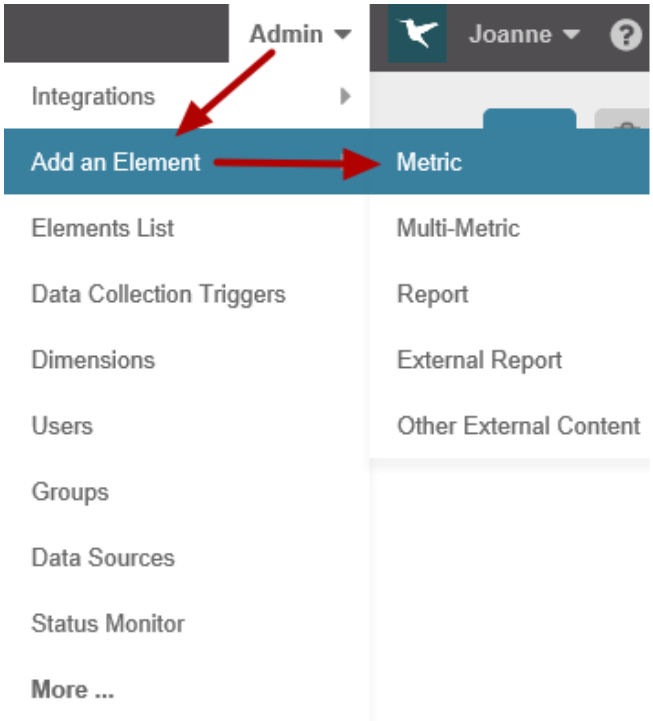


1. You can create Datasets or Elements directly from the respective tabs
2. Click **Permissions** to assign permissions to the Data Source to Groups or Power Users

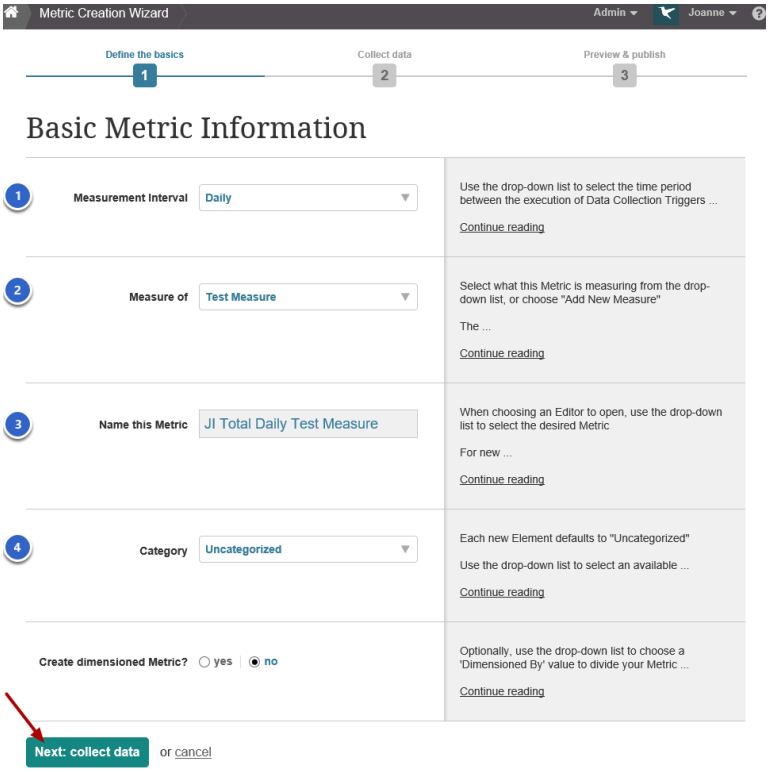
31.4 How to collect data using Salesforce SOQL

This article will show you how to create an Element using a SalesForce SOQL plug-in as a data source. It assumes that you have already established connectivity to SalesForce SOQL.

1. Add a new element based on your Salesforce SOQL plug-in data source



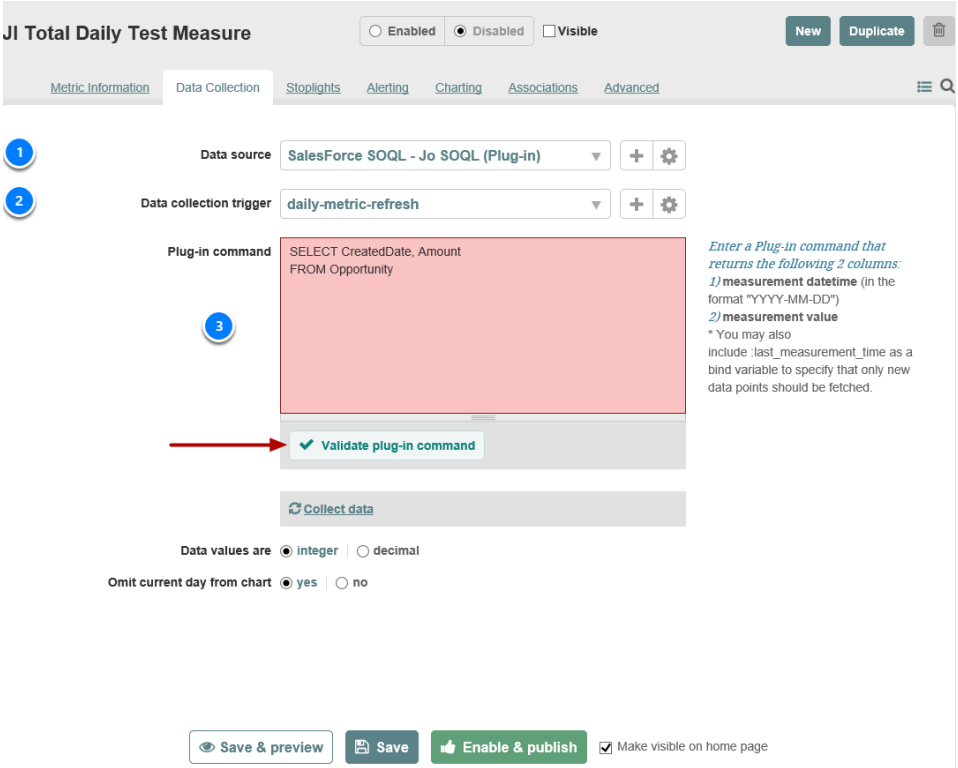
2. Provide basic information on Wizard (or Editor) - metric example



1. Select the **Measurement Interval** that applies to your element
2. Specify what this metric is **measuring**. If you do not see the measure that you want to use, you can create one from this drop-down
3. Give the element a unique **name**
4. Optionally, assign a **Category**

Next: (example is using full editor, but same steps apply when using Wizard)

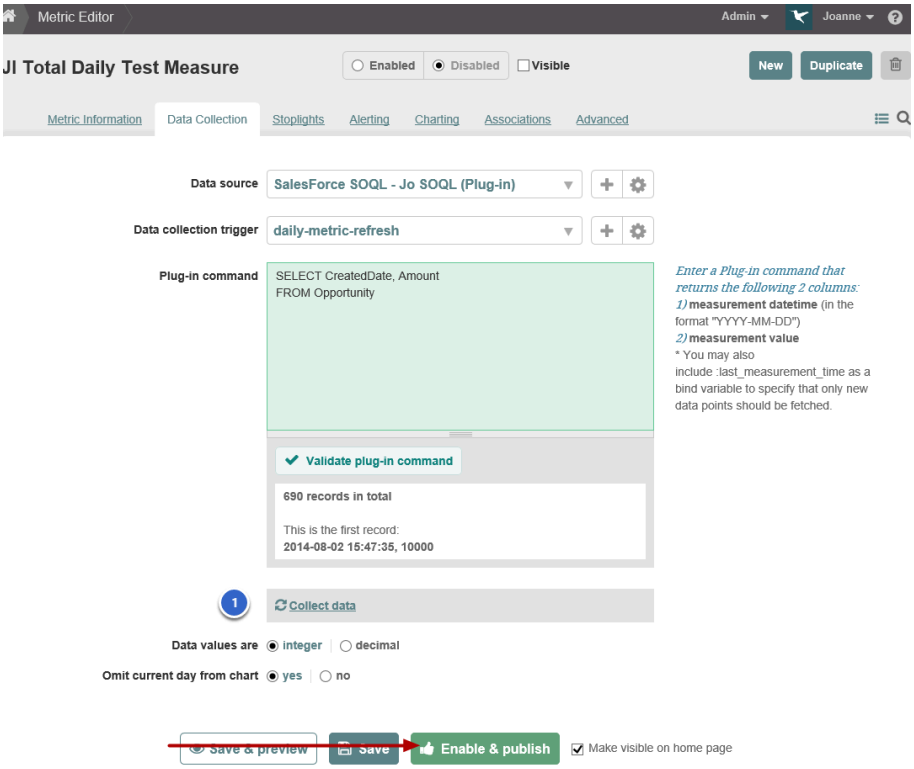
3. Full Editor displays the Data Collection tab



1. Select **Salesforce** plug-in in **Data Source** drop-down
2. Set **Trigger**
3. Input Plug-in Command

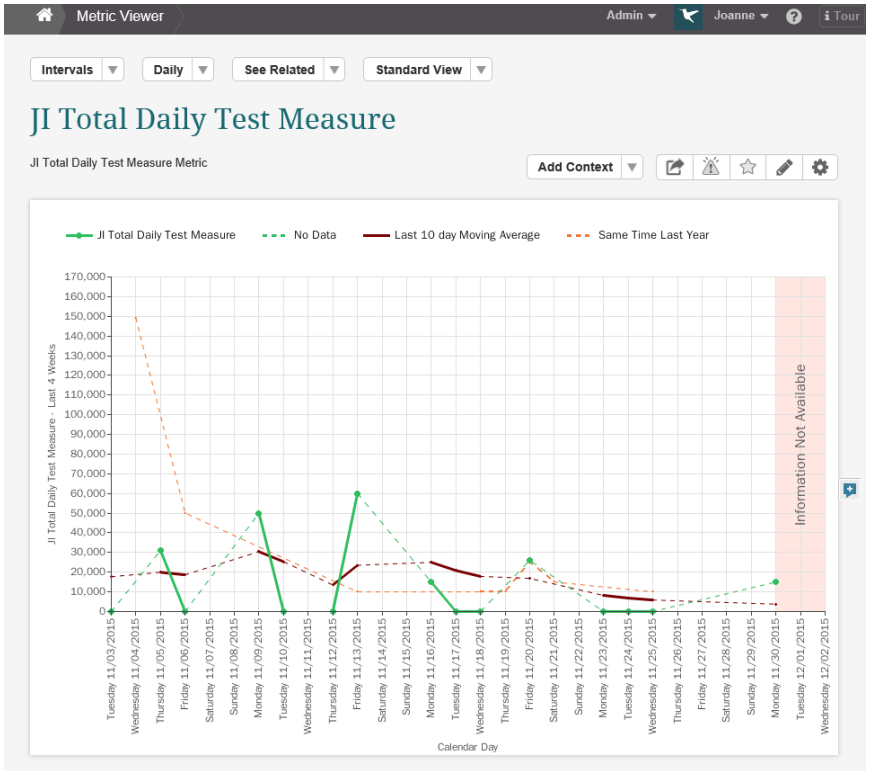
Validate plug-in command

4. Collect Data and Publish



Upon successful validation, **Collect data** then **Enable and Publish**

5. Metric will be displayed in viewer

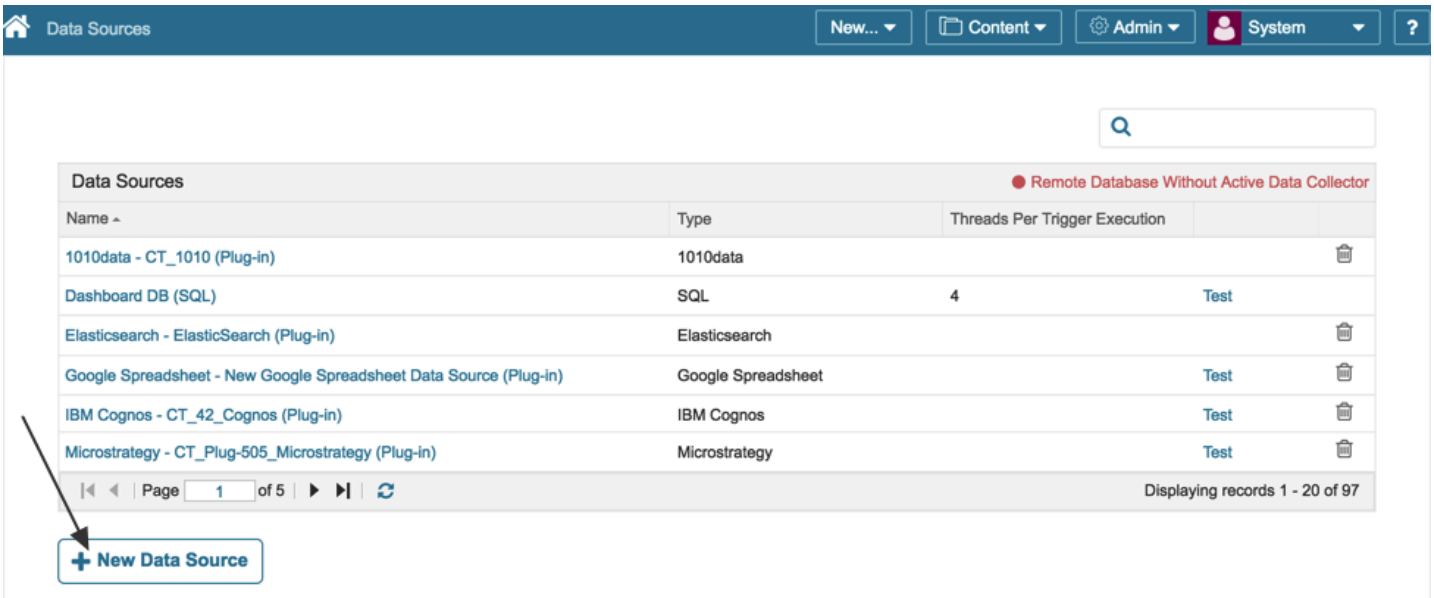


32. Sourcing Data from Sisense

32.1 Establish Connectivity to Sisense

This article describes the process of creating plug-in Data Source to connect to the Sisense server. This Data Source will allow data from existing Sisense objects to be used in building elements using Metric Insights tools.

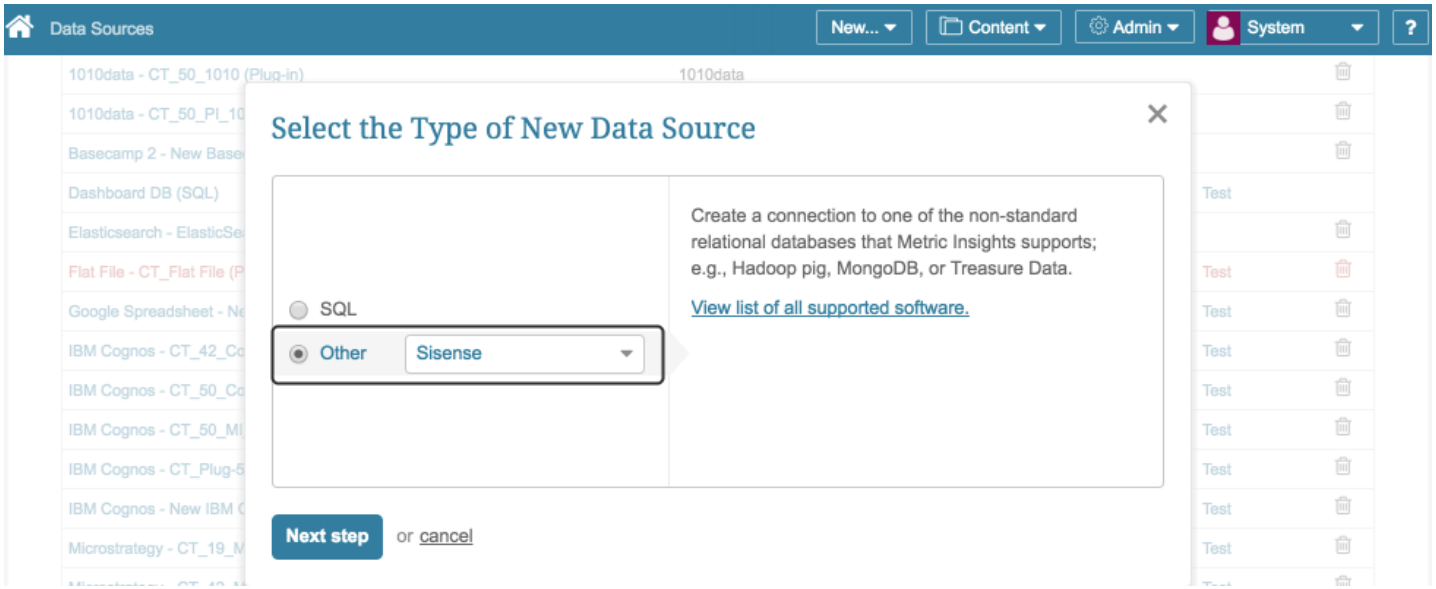
1. Access Admin > Data Sources



At the bottom of the screen click **[+ New Data Source].**

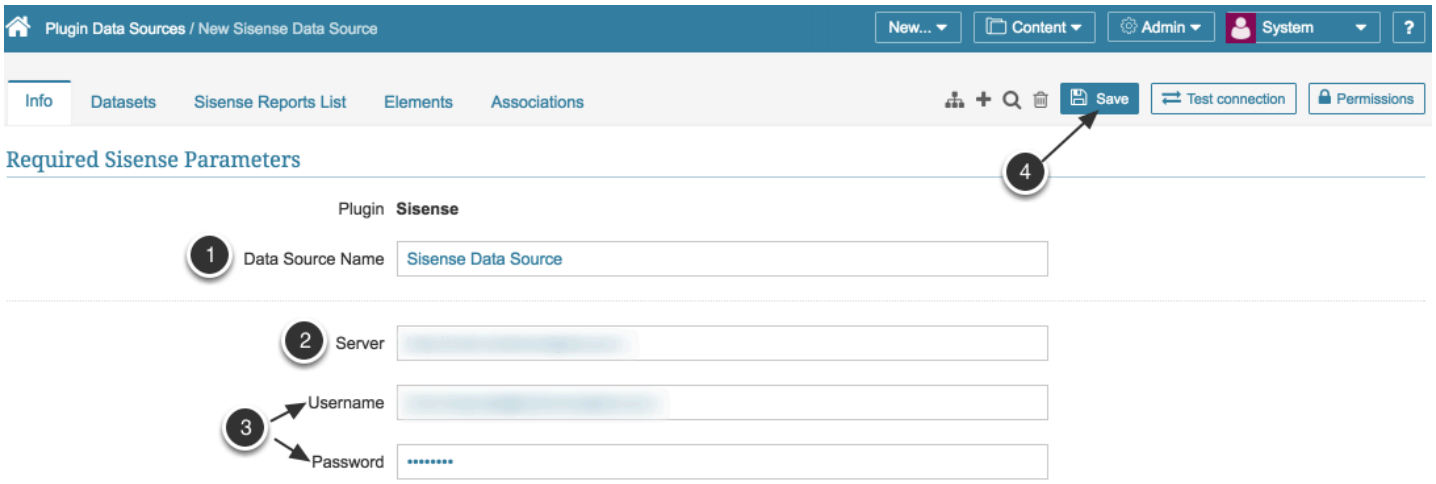
The *Select the Type of New Data Source* pop-up opens.

2. Select "Other" Data Source Type and choose "Sisense" from the drop-down



Move to the **Next step**.

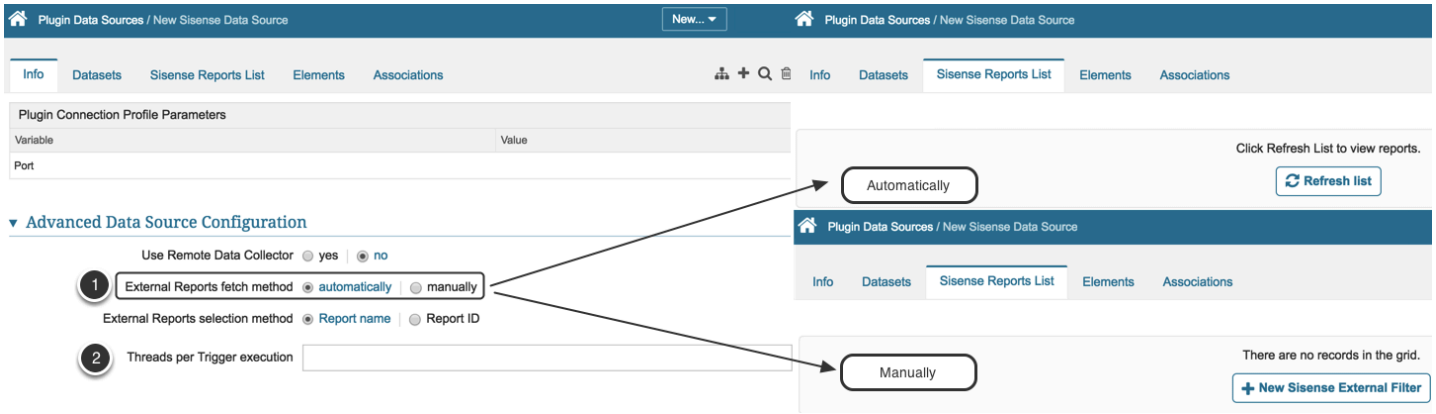
3. Provide Required Sisense Parameters



Specify how to connect to Sisense. The parameters include:

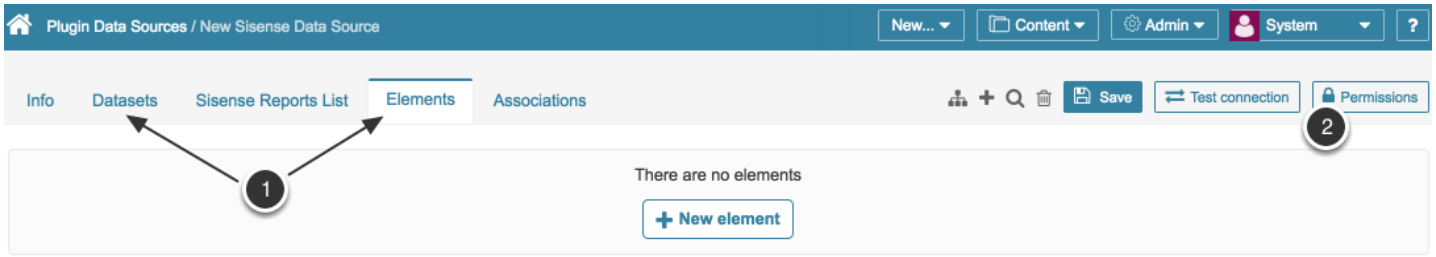
1. **Data Source Name:** Will default but you may modify it.
2. **Server:** The URL to use for creating UI links back to your Sisense server
3. **Username / Password:** Note that your **Username** must be in the same format that your Sisense server uses for Authentication (it may be an email address)
4. **Save** your entries and **Test Connection**

4. Advanced Configuration



1. **External Reports fetch method**: This setting influences options available in the *Sisense Report List* tab:
   - **automatically**: just click **Refresh list** and all Reports are going to be fetched by the system
   - **manually**: Reports may be added one-by-one or via CSV file
2. Optionally, specify the maximum number of concurrent **Threads per Trigger execution** to be used in background processing when the system updates Metrics and Reports for this Data Source. If you do not specify any value for this setting, batch data collection processing will be single-threaded.

5. Other Settings



1. You can create elements or Datasets directly from the respective tabs
2. Click **Permissions** to assign them to Groups or Power Users

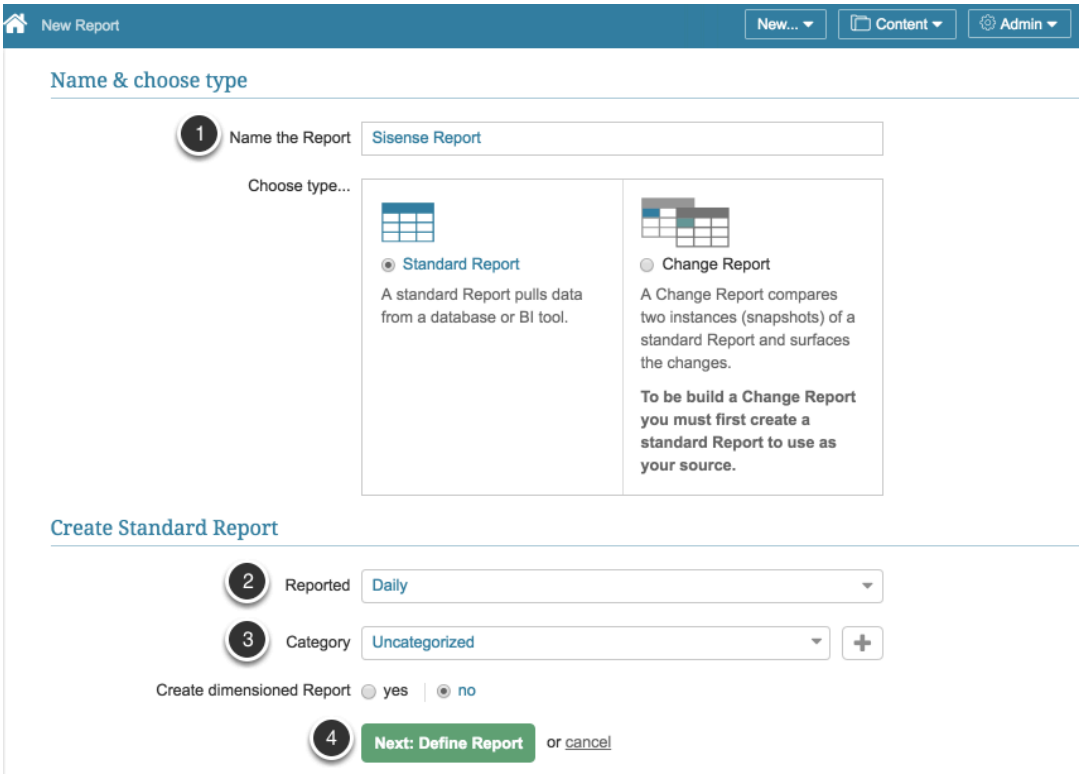
What's next?

[How to collect data from Sisense?]

32.2 How to collect data from Sisense

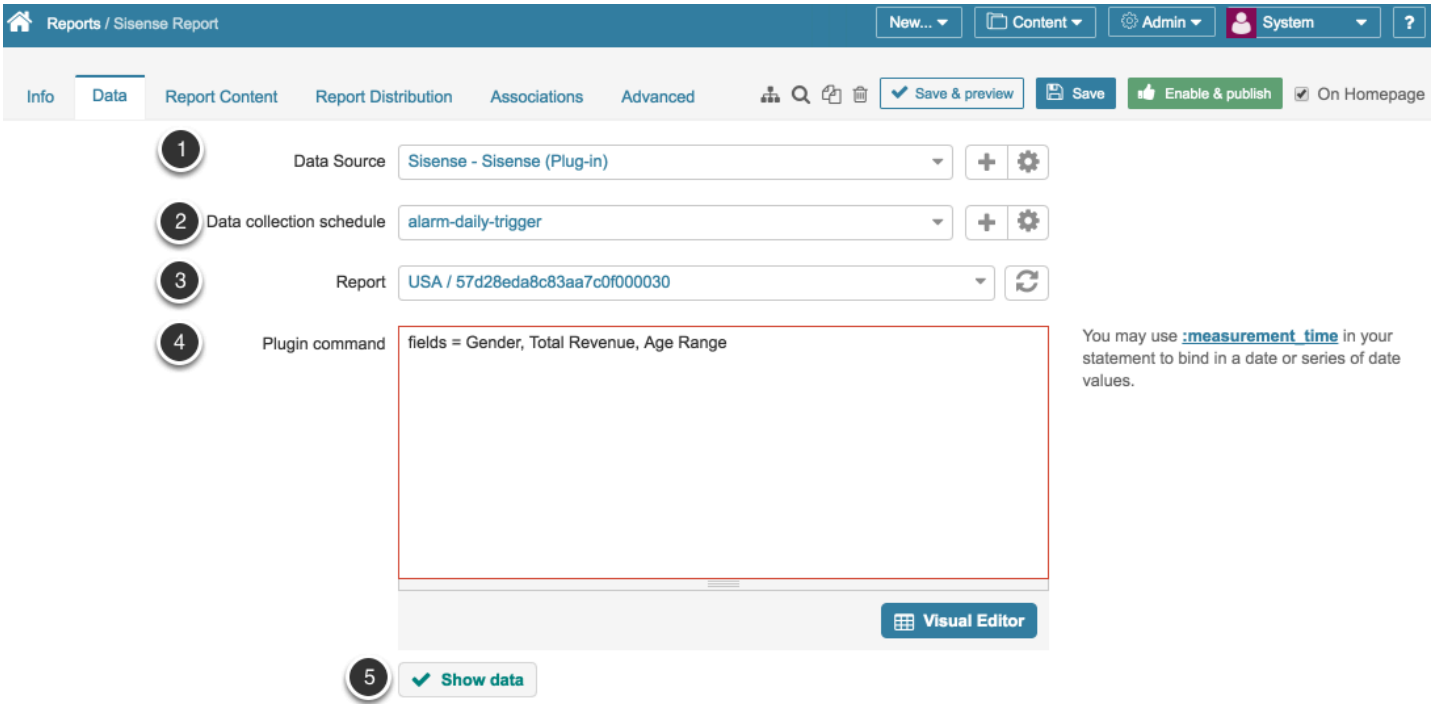
This article demonstrates how to create a Metric or Report using a Sisense as a data source. It assumes that you have already established connectivity to Sisense.

1. Access New > Report



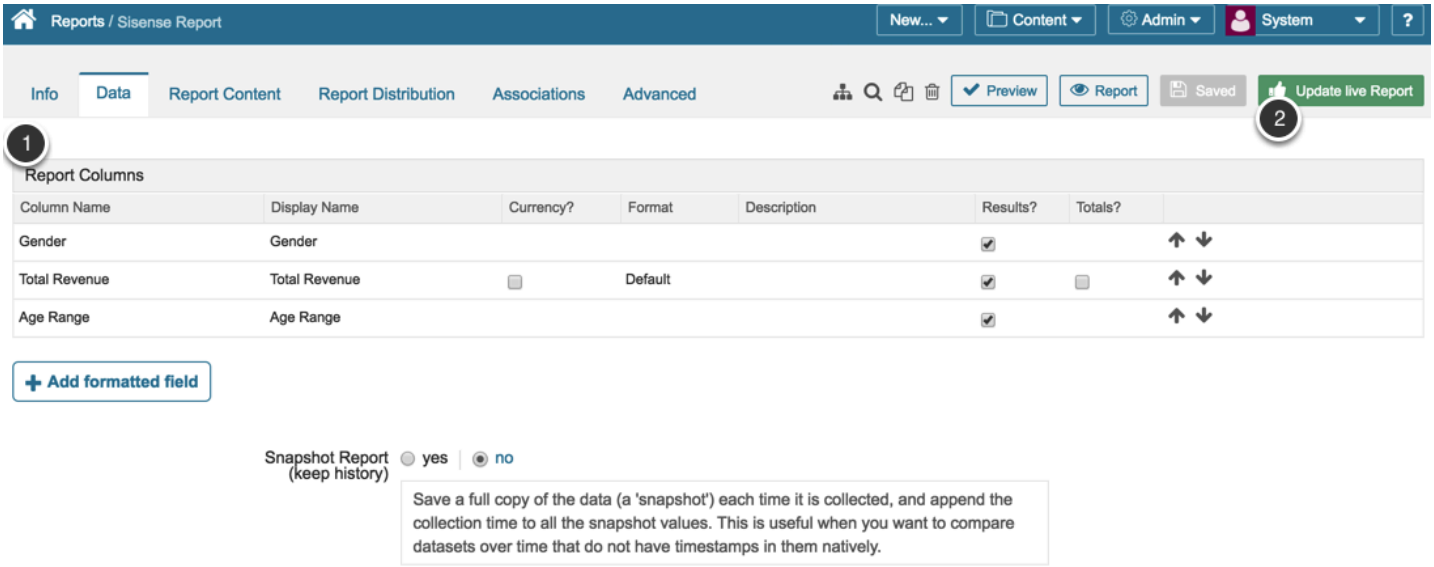
1. **Name the Report**: Define a unique descriptive name of your element
2. **Reported**: choose the measurement interval from the drop-down list
3. **Category**: define a category this element belongs to
4. To move on to defining data collection details, click **Next: Define Report**

2. Define settings for Data Collection



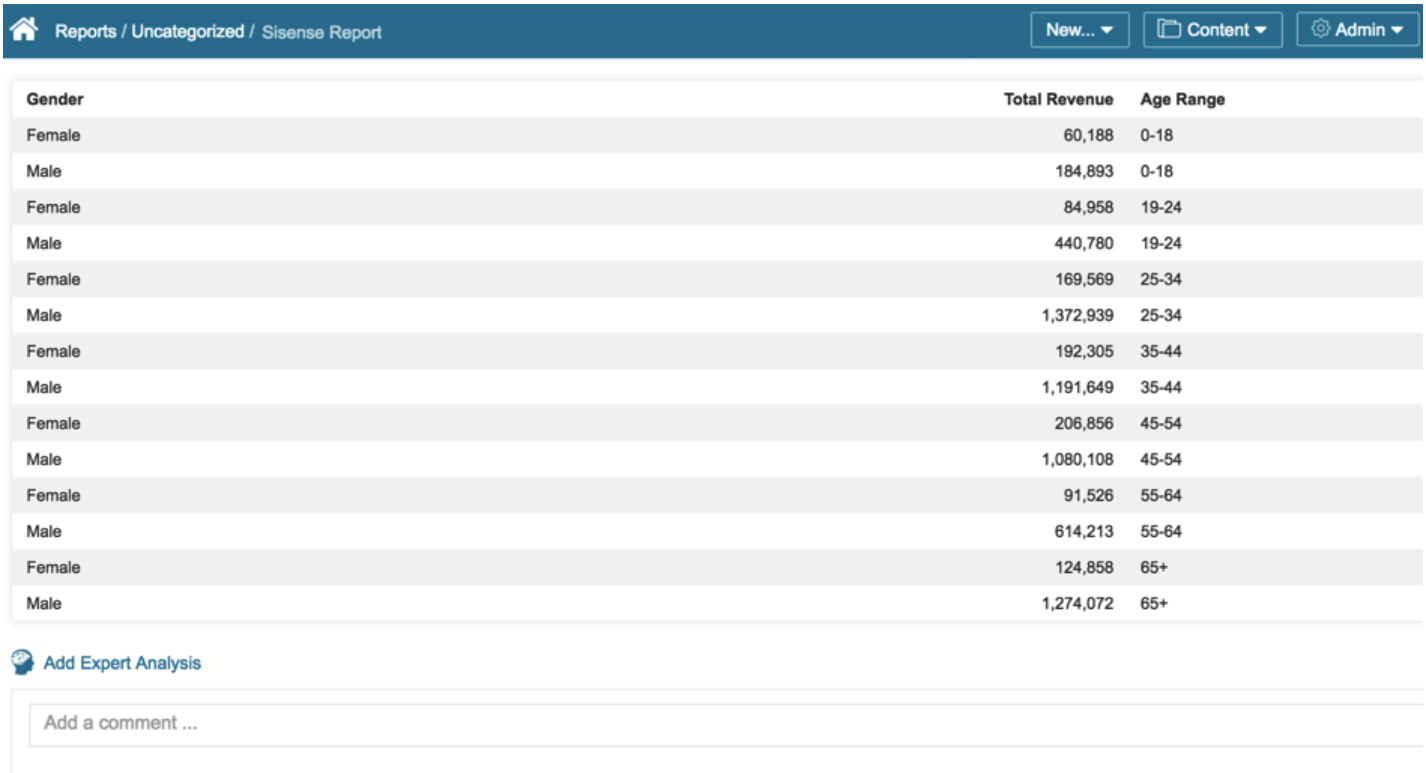
1. **Data Source**: select the account you have created for Sisense
2. **Data Collection Schedule**: Specify the trigger that will be used to collect the data for your report
3. **Report**: Choose the report that should serve as a basis from the drop-down list
4. Input **Plug-in Command** listing all the data you would like to fetch from Sisense (manually or using the **Visual Editor**)
5. Once you are ready with you command, click **Show Data**.

3. If the command is validated successfully



1. If your plugin command is valid, the command box is green and the **Report Columns** are shown in the table below; if there are any errors, the box is colored in red and errors are explained below the statement box.
2. Click Update Live Report to save the changes and move to the *Report Viewer*.

Result



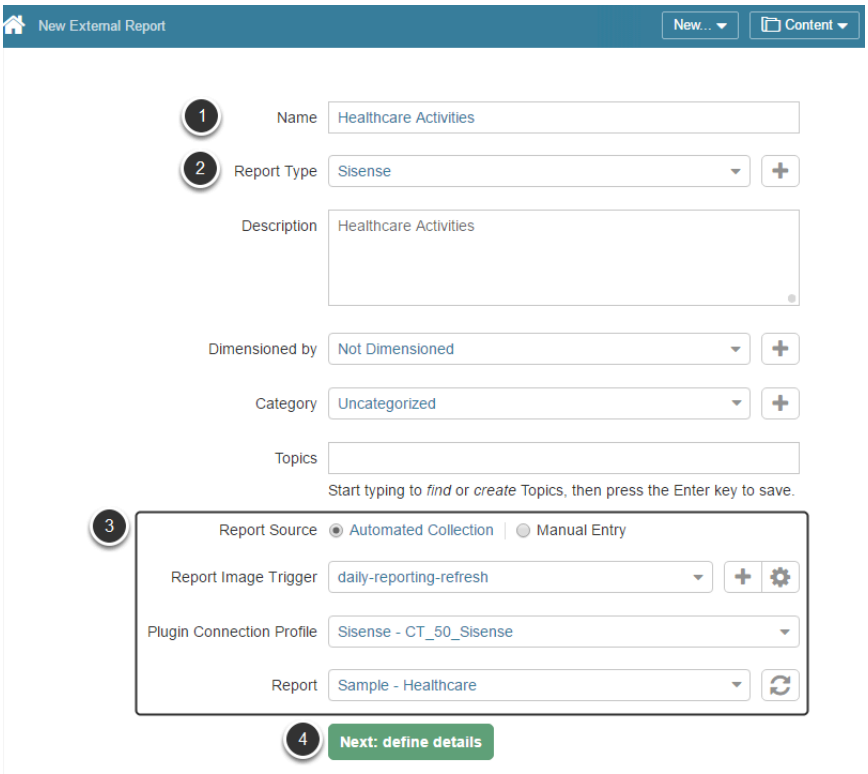
Gender	Total Revenue	Age Range
Female	60,188	0-18
Male	184,893	0-18
Female	84,958	19-24
Male	440,780	19-24
Female	169,569	25-34
Male	1,372,939	25-34
Female	192,305	35-44
Male	1,191,649	35-44
Female	206,856	45-54
Male	1,080,108	45-54
Female	91,526	55-64
Male	614,213	55-64
Female	124,858	65+
Male	1,274,072	65+

Here is a Report with the data fetched using Sisense plug-in.

32.3 How to create an External Report from Sisense

This article will show you how to create an External Report that is linked to a report from your Sisense server and is based on the assumption that you have already established a connection with Sisense server.

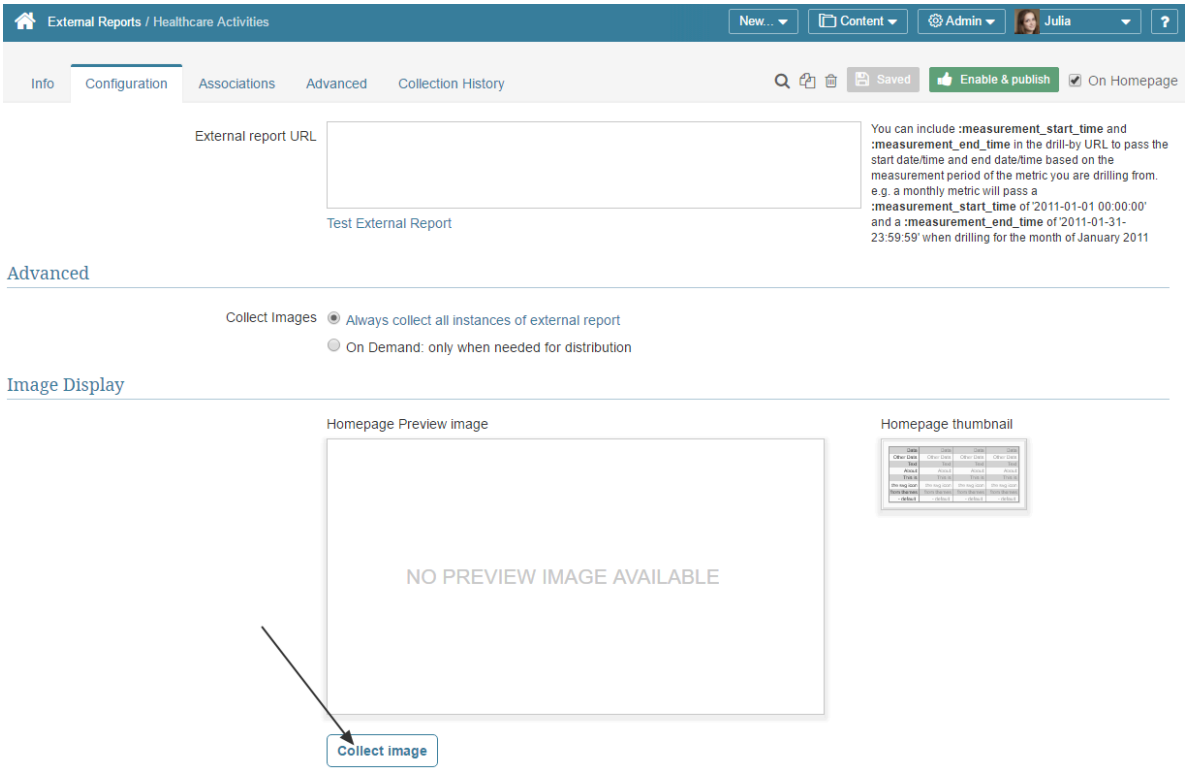
1. Access New > External Report > Sisense



The *New External Report* screen opens. Provide the following information:

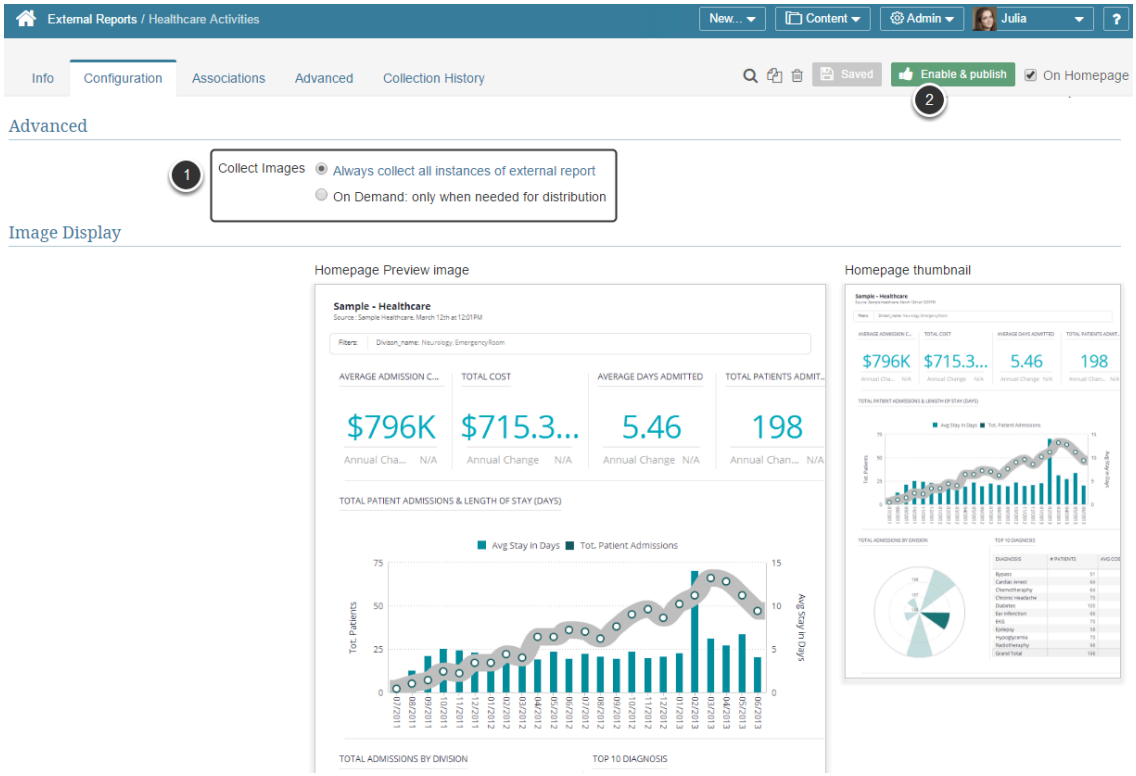
1. Give your new External Report a **Name**
2. **Report Type:** If there is no required Report Type in the list, click the Plus (+) button and create a new one.
3. Define whether you want report content to be updated manually or automatically. In case you choose **Automatic Collection**, define the following settings:
   - Define the **Report Image Trigger** from drop-down list
   - Select the **Plugin Connection Profile** you have created for Sisense. For more details refer to: Establish Connectivity to Sisense
   - **Report:** Select a required workbook available from the selected connection profile
4. Click **Next: define details** to proceed with Report creation.

2. Collect Image



Scroll down to the bottom of the page and click **Collect Image**.

3. Advanced Settings



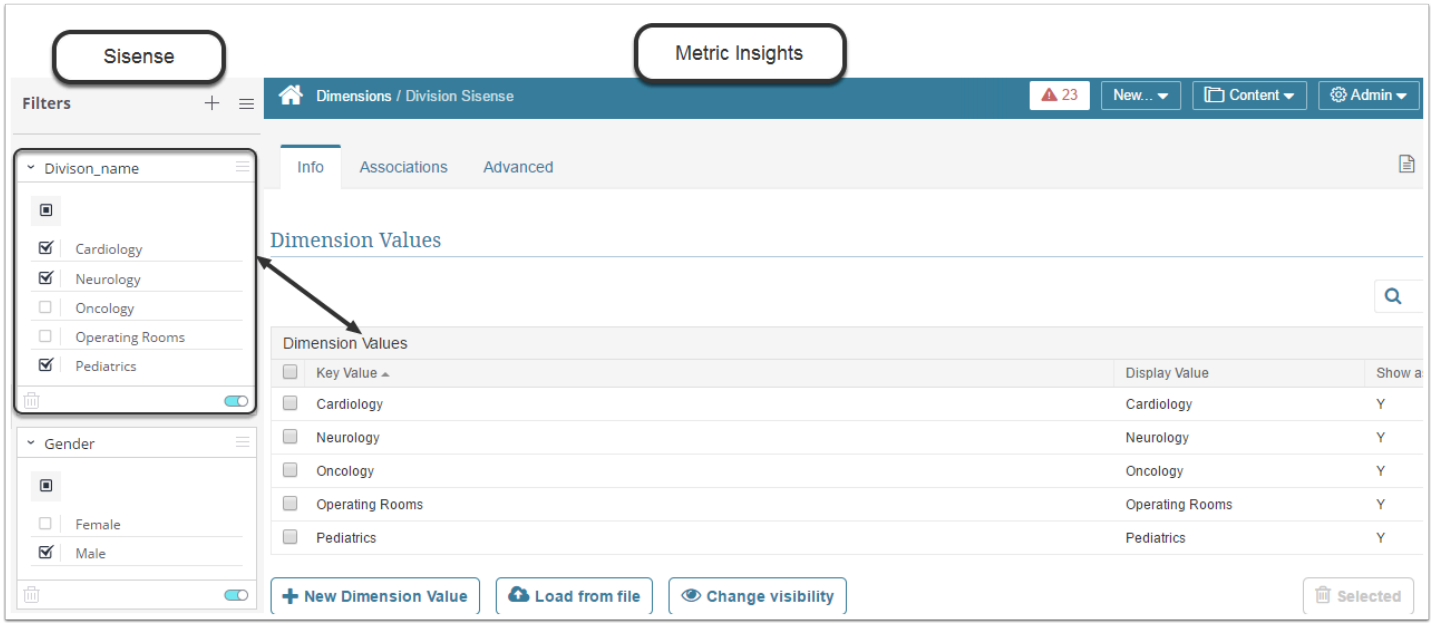
1. **Collect Image**:
   - **Always collect all instances of external report**: Collect all images and cache them on a schedule.
   - **On Demand: only when needed for distribution**: Individual images are only collected when they need to be included in an email.
2. **Enable and Publish**

32.4 Create a Dimension with values automatically fetched from Sisense

PREREQUISITES:

[Establish Connectivity to Sisense]

Use Case



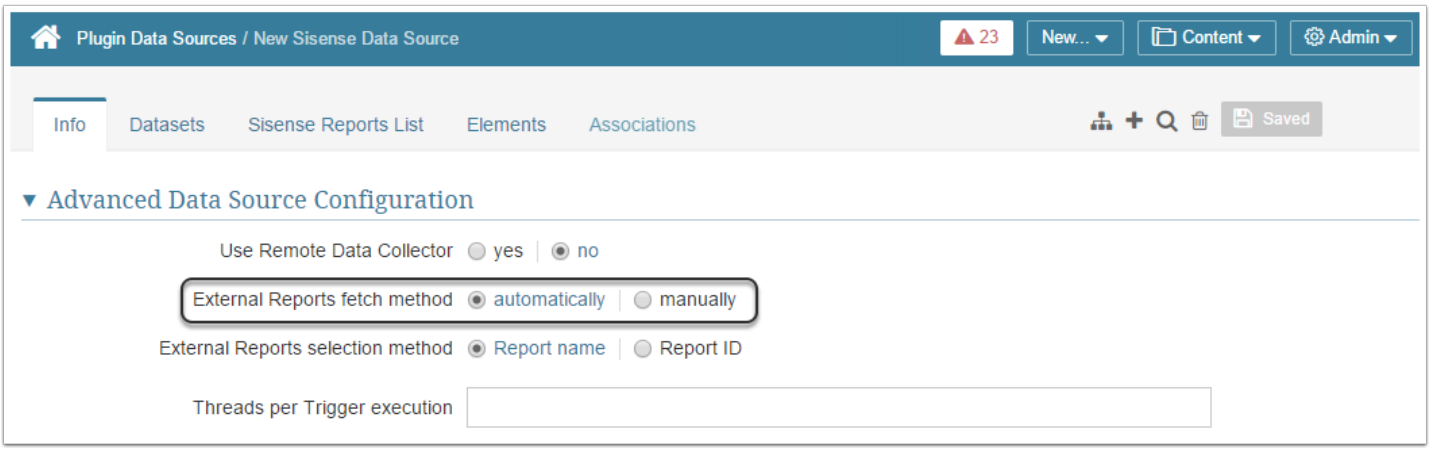
1. Open the Sisense Plugin you are going to use as a Data Source for the future Dimension

1. Go to *Admin* > *Data Sources.* The list with all data sources created in the system is going to be opened.
2. Select the plugin you plan to use as a Data Source for the future Dimension. Alternatively, crate a new one: [Establish Connectivity to Sisense]

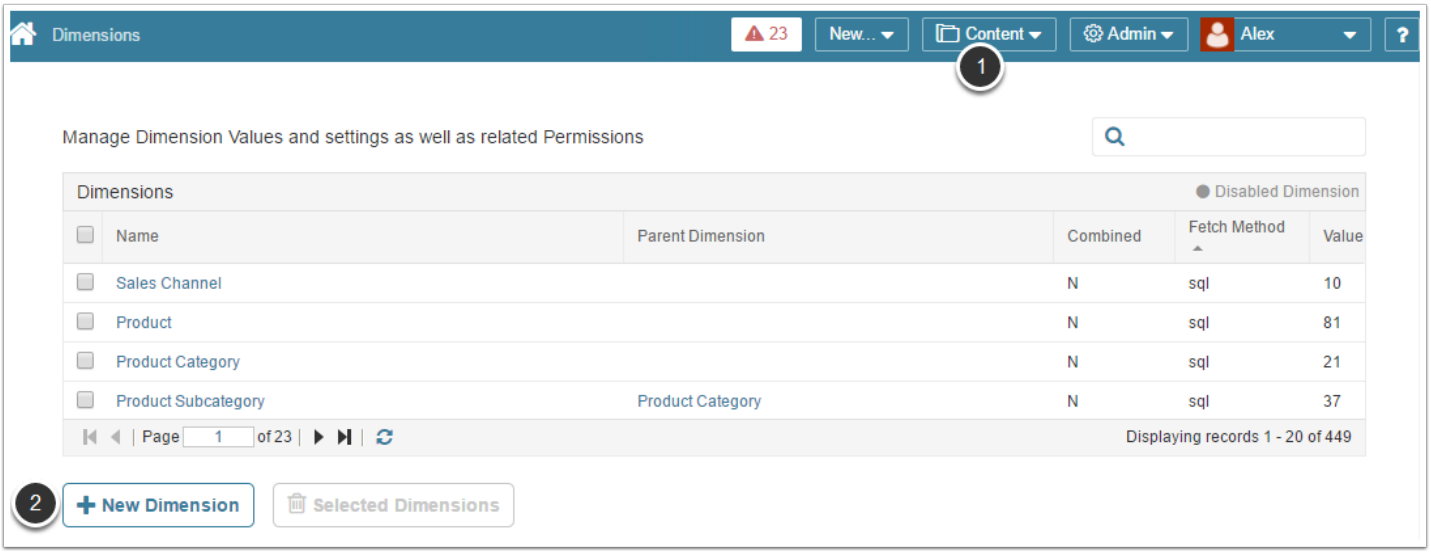
1.1. Update the list of Sisense Reports

Metric Insights extracts data from the QlikView server in the form of reports. Information obtained from these external reports is further used as data source for MI elements: Metrics, Reports, Datasets, etc. QlikView Reports can be added in the Plugin Editor. There are two options to add reports to the plugin:

- **Automatically**: If the External Reports fetch method field is set to 'automatically', go to the QlikView Reports List tab and simply click **Refresh** list to collect all QlikView reports currently available at the server
- **Manually**: You can also update the QlikView Reports List by adding report IDs and Names one-by-one or via a CSV file

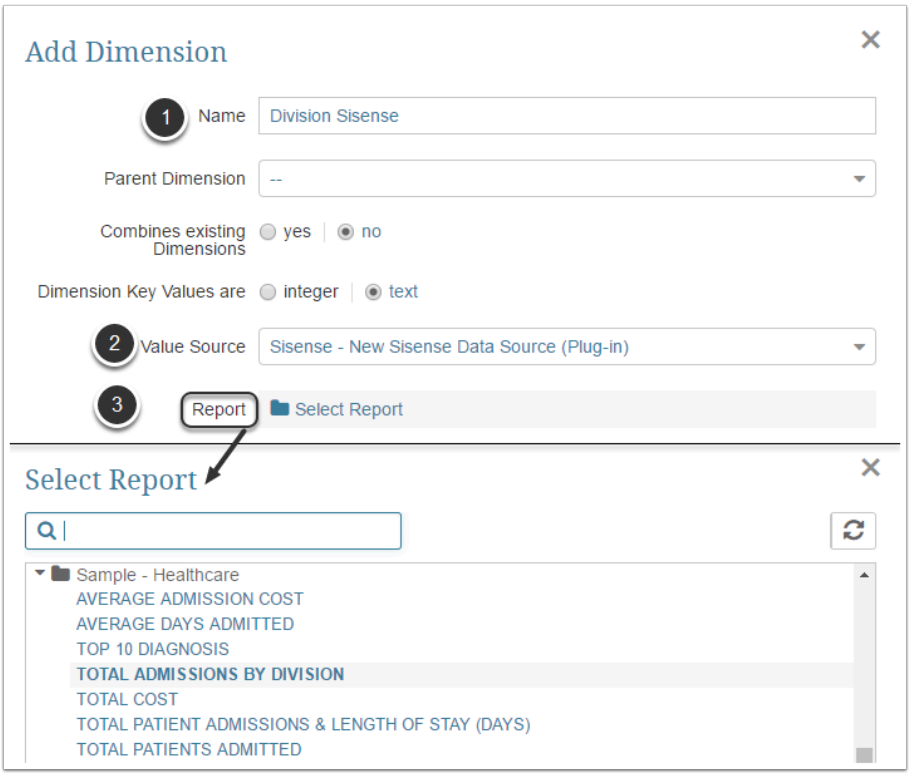


2. Add a new Dimension



1. Go to *Content* > *Dimensions*. The list with all dimensions created in the system opens.
2. Click **[+ New Dimension]**

2.1. Define the Basics

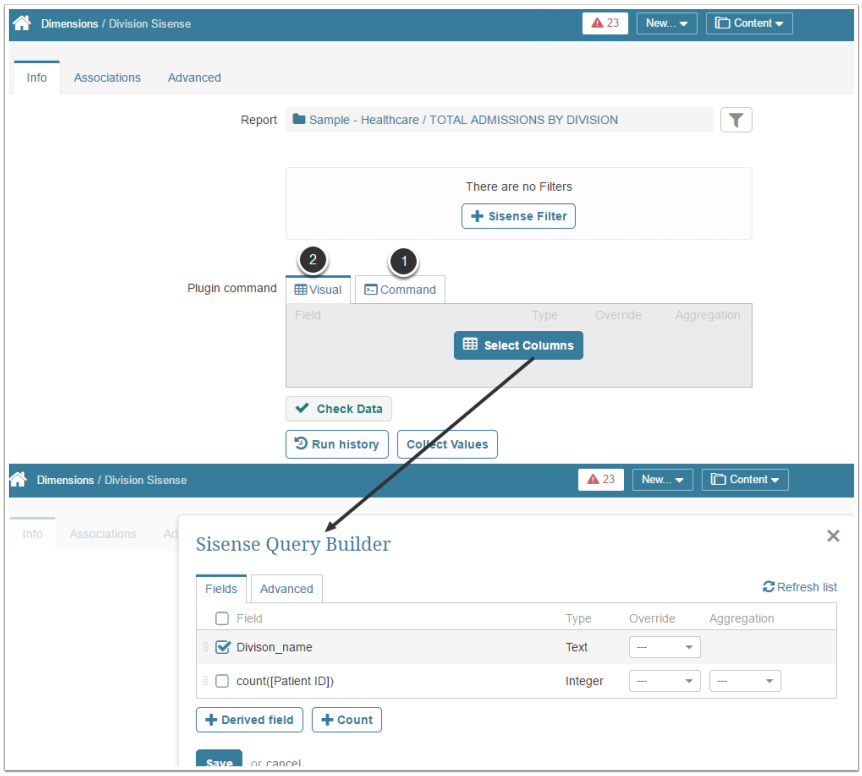


Provide the basic Dimension definition information, paying attention to:

1. A unique **Name** for your Dimension
2. **Value Source:** Specify how Dimension Values will be collected for the new Dimension. In this example, we are selecting 'Sisense' plugin from previous steps which is going to serve as a data source
3. **Report:** Define the Sisense item report that contains the required values in the **Select Report** pop up. In our example we need to fetch Division Values, so we have selected a 'Sales by Country' item since it contains the info we are looking for.

Save your entries. The *Dimension Editor* opens.

2.2. Enter the command for fetching data

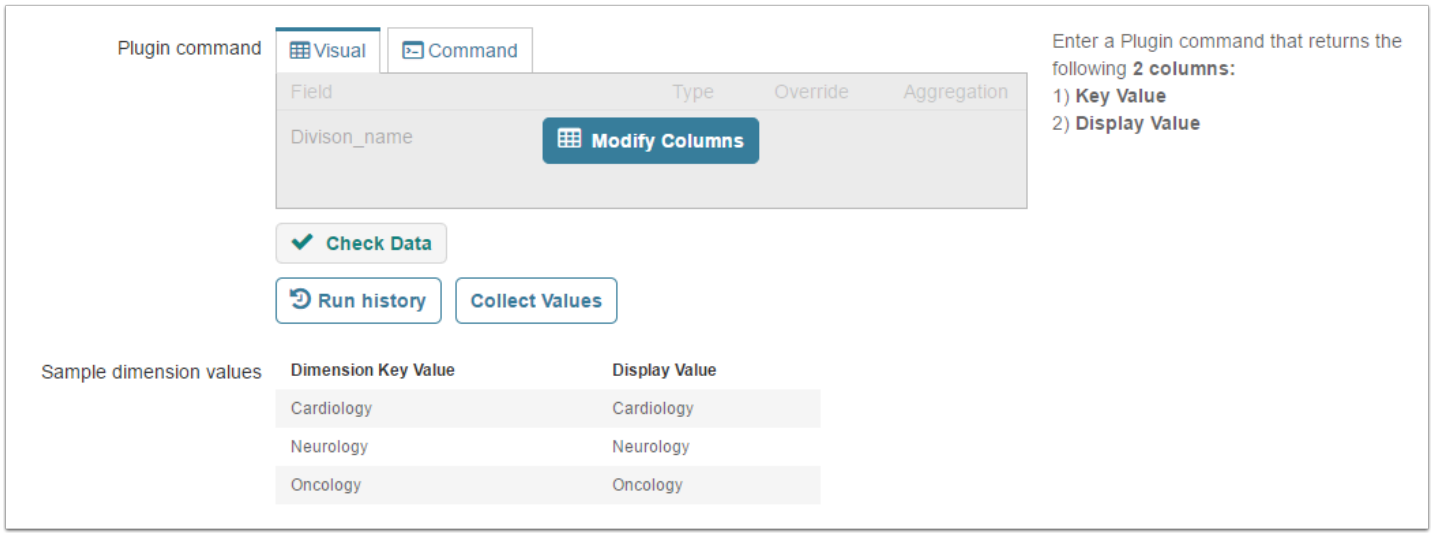


1. You can enter the command manually

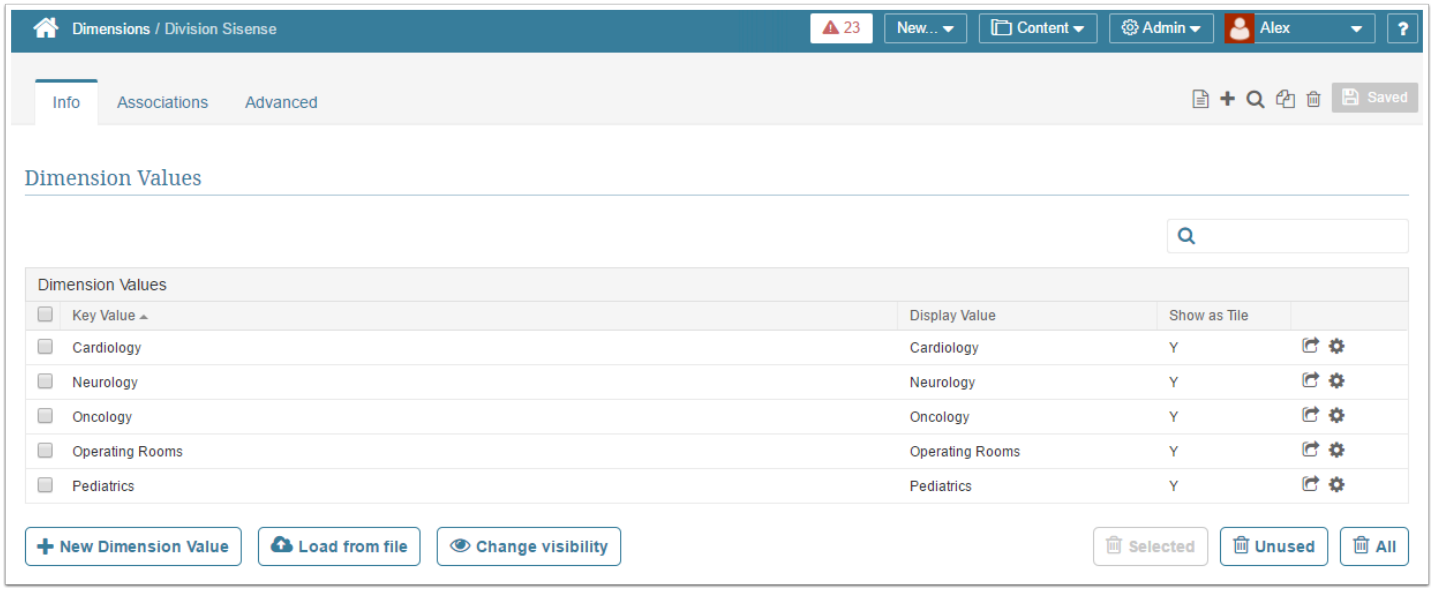
OR

2. Use a Visual Editor and choose the required fields

2.3. Check Data and Collect Values



Result



What's next?

You can now use this Dimension to create dimensioned Metrics, Reports and External Reports from Sisense.

32.5 Pre-filtering Sisense data (Version 5.1)

When sourcing data for Metrics, Reports, External Reports, Dimensions and Datasets from Sisense dashboard, you can pre-filter data before fetching it. This function allows focusing on the slice of data that you really need and exclude those values that are currently irrelevant for you and your research.

PREREQUISITES:

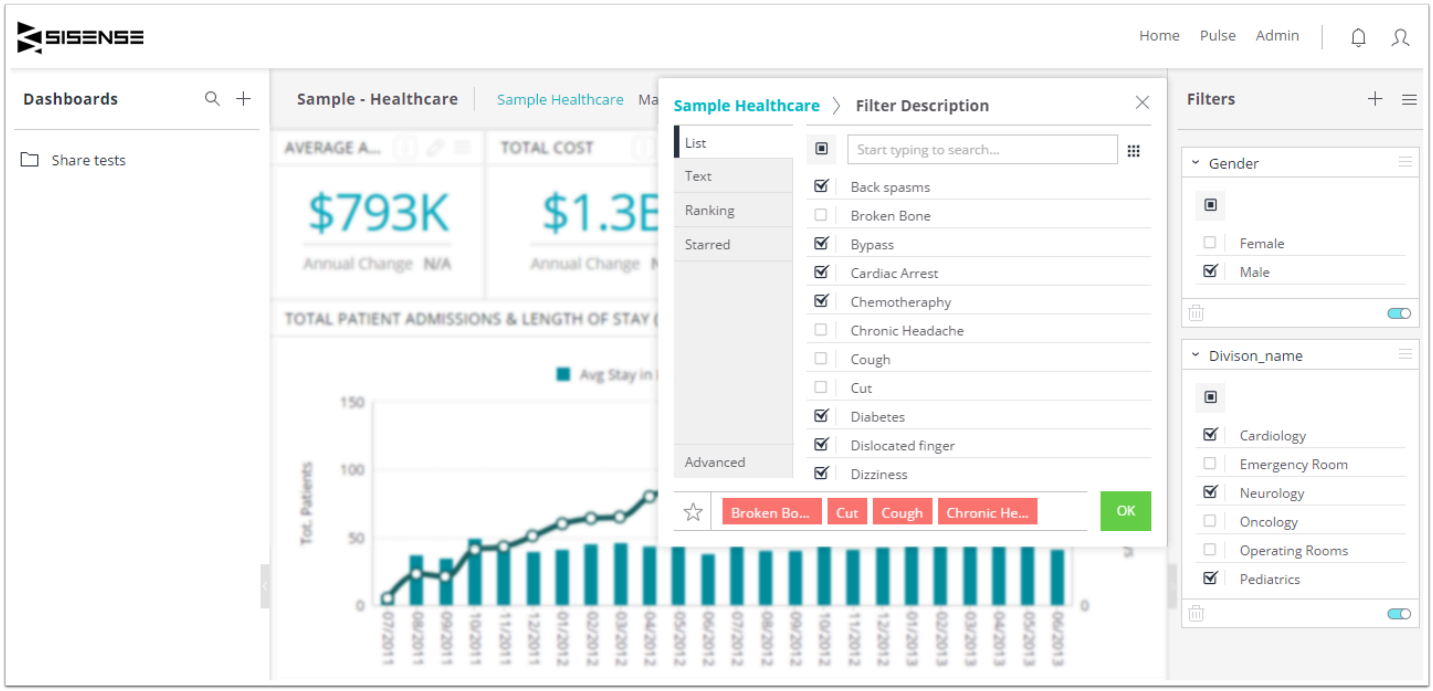
Establish Connectivity to Sisense

Sample of a Sisense Dashboard

Most of the data displayed on the Sisense Dashboards can be refined using the flexibility of filter settings.

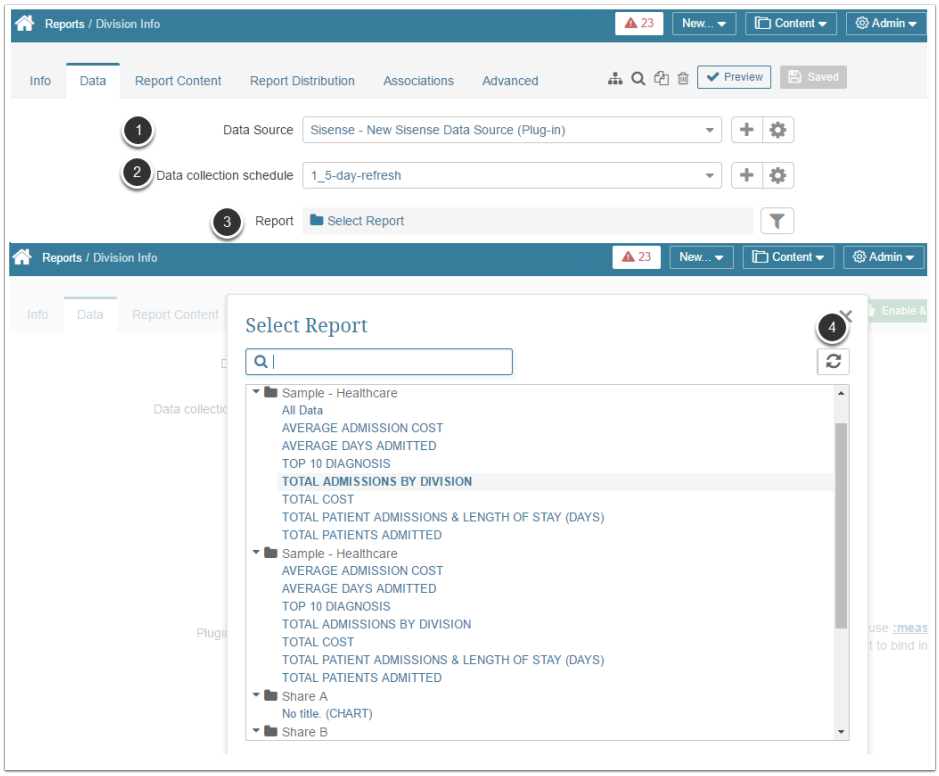
On the example below, the filters that have beehn applied to the dashboard Reports are shown to the right of the graph:

- Gender
- Division Name



1. Define a Source Report for Data Collection in the Element

Editor



Start off by creating an element. Once you get to the process of Data Collection, define the following:

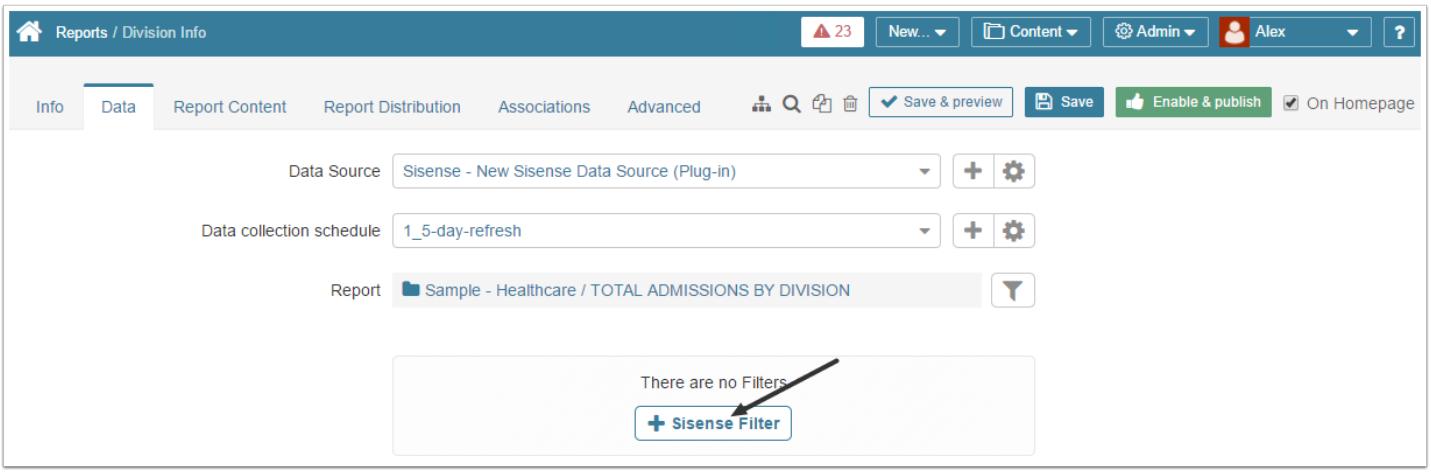
1. **Data Source:** This is an entity that connects QlikView and Metric Insights. For more information, see: [Establish Connectivity to Sisense]
2. **Data Collection Trigger:** select the trigger which is going to initiate updating information in this Metric
3. **Report:** Click **Select Report** to open the pop-up with the list of available Sisense objects that can work as a source of data.
4. Each item in the list is represented as the path (hierarchy) to the respective report in Sisense. Find the object in the list.
5. If you do not see the required item, try refreshing the list by clicking the **Refresh** icon at the upper right corner of the pop-up.

2. Add Sisense Filters to Metric Insights

> ⚠ Once filters are added to a Metric / Report or External Report for the first time, they are going to be automatically added to all new respective elements with the same Data Source / Report.

NOTE:

- External filters are tied to Sisense Reports, not Metric Insights' elements. This allows Filters to be reused for multiple elements (there is no need to create new Filters every time an element is created in Metric Insights).
- If there are more External Filters or Filter Values that you would like to use for the current element, you can always set the redundant ones to "ignore".

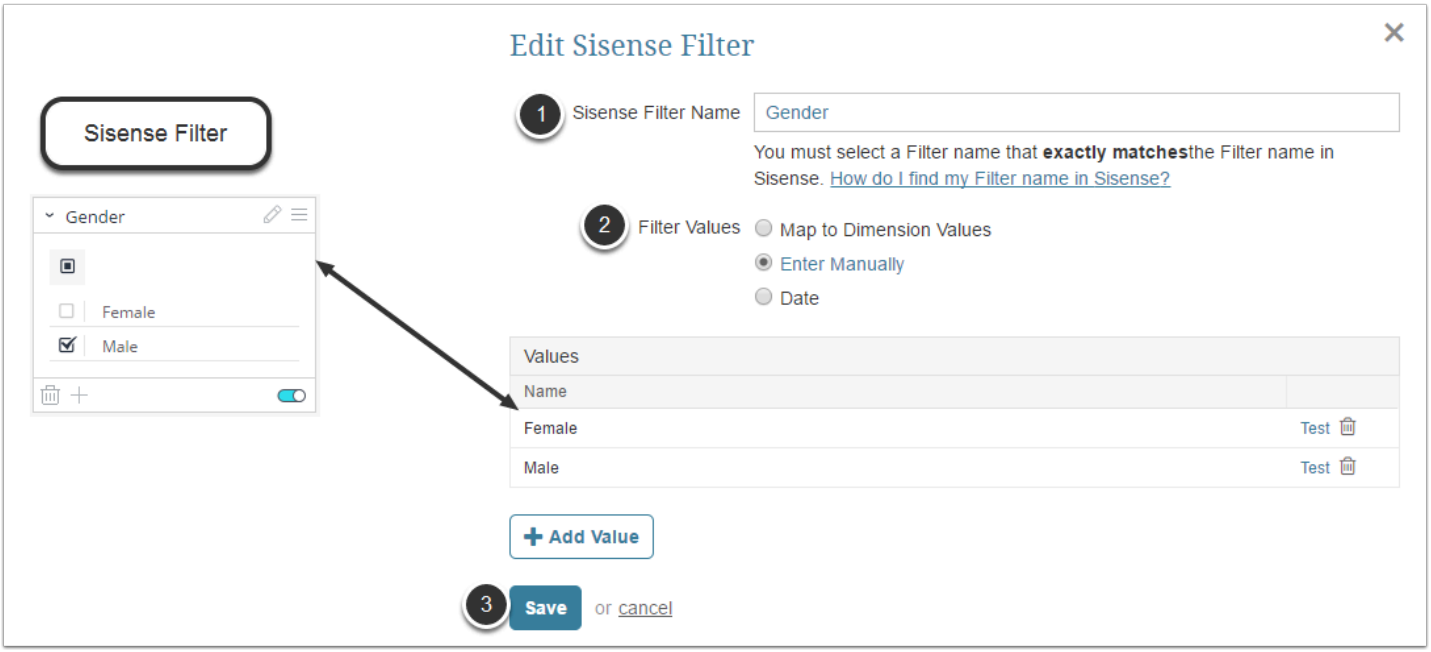


When creating a Metric / Report / External Report generated from the Sisense dashboard, after you define the Report that should serve as a source of data, you may pre-filter information that is going to be fetched.

To do that, click **[+ Sisense Filter]**. Next, you can choose whether you are going to define filters manually or via the existing Dimension Values. For instructions on creating a Dimension sourced from Sisense see:

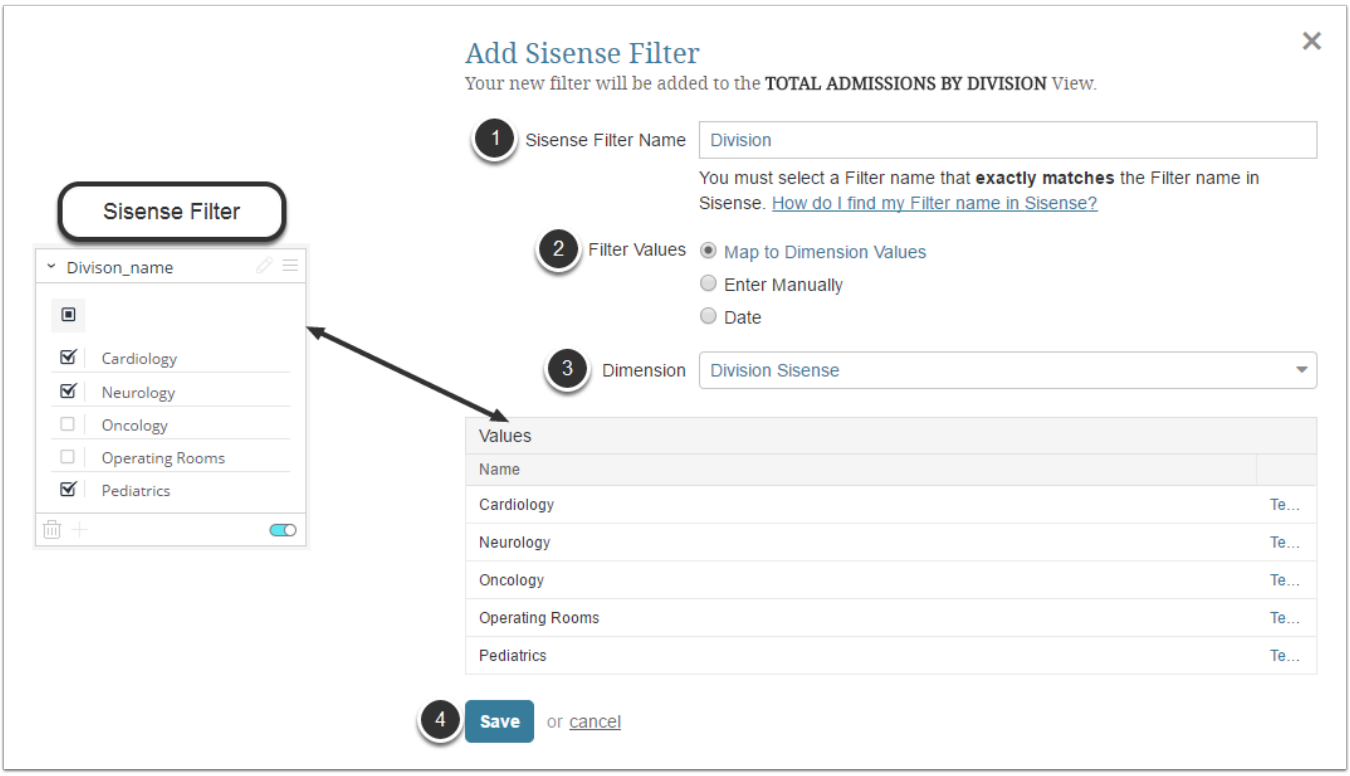
NOTE: Examples given below are taken from the Sisense Report shown at the top of the page.

Enter Manually



1. **Sisense Filter Name**: Define the name of the filter from Sisense
2. **Filter Values:** choose 'Enter Manually' and click **Save** at the bottom of the pop-up.
3. Click **[+ Add Value]** and in the opened pop-up manually type in the name of the filter value. **Save** your entry. All added values should appear in the *Values* list.
4. **Save** your entries.

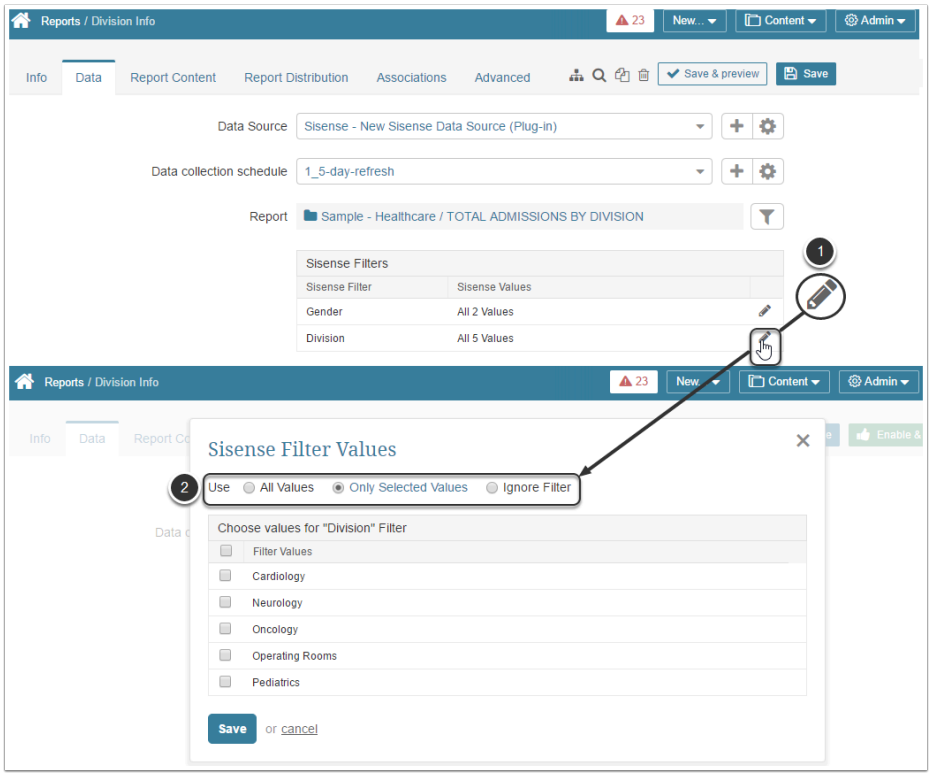
Using Dimension Values



If you have already used Qlik filters to create Dimensions in Metric Insights, you can quickly choose which Dimension Values you want to use for pre-filtering:

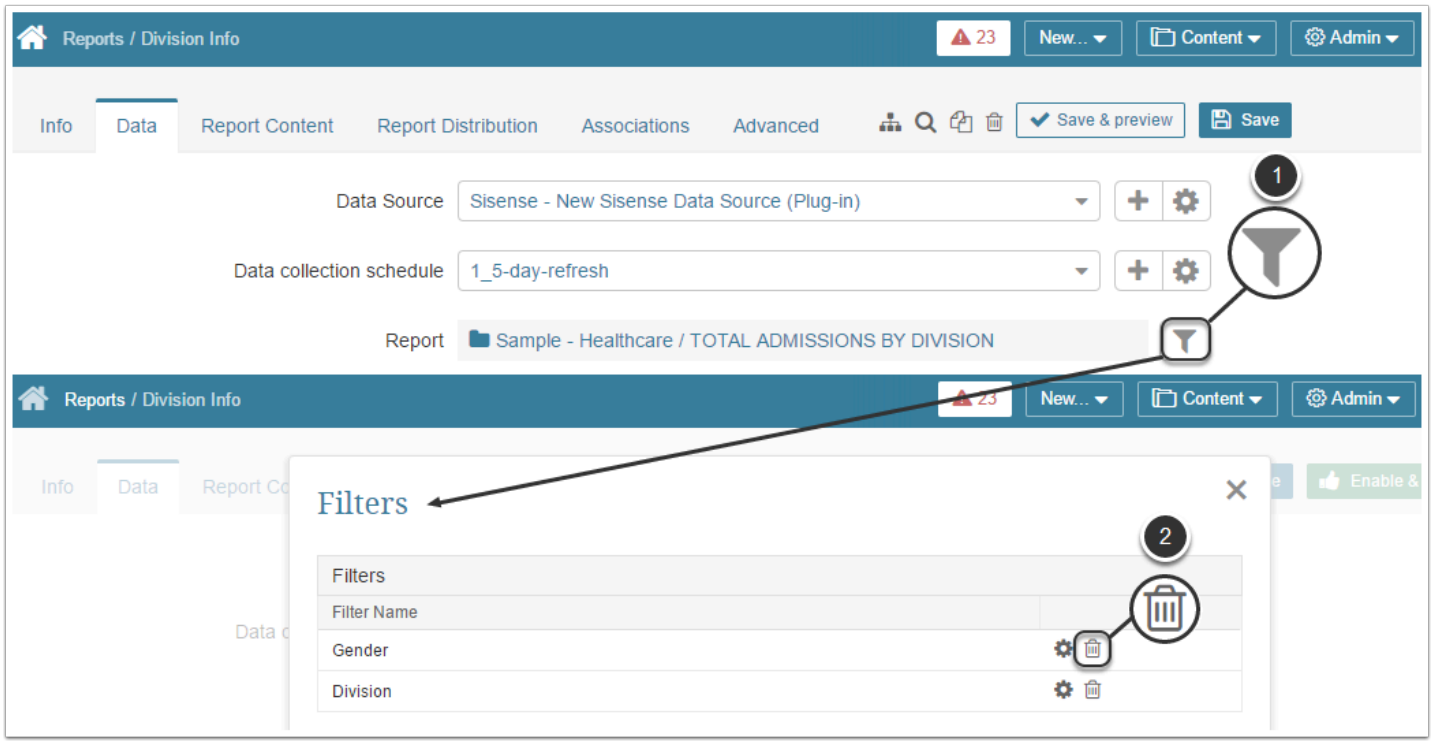
1. **Sisense Filter Name**: Define the name of the filter from QlikView.
2. **Filter Values:** choose 'Map to Dimension Values'.
3. **Dimension:** select a corresponding Dimension from the drop-down list and all its Dimension Values are going to be loaded to the Values list automatically. For more details refer to: Create a Dimension with values fetched from Sisense
4. **Save** your entry.

How do I add filters to a results set from Sisense?



1. Click the **Pencil** icon in the filter row to set it up.
2. When the filter is added, you can use it for "All Values", "Only Selected Values" or ignore it.

Deleting Filters



To delete some of the added filters: (1) click the **Filter** icon in the **Report** field and (2) choose the unnecessary filters. Click the **Trashbin** icon in the respective row.

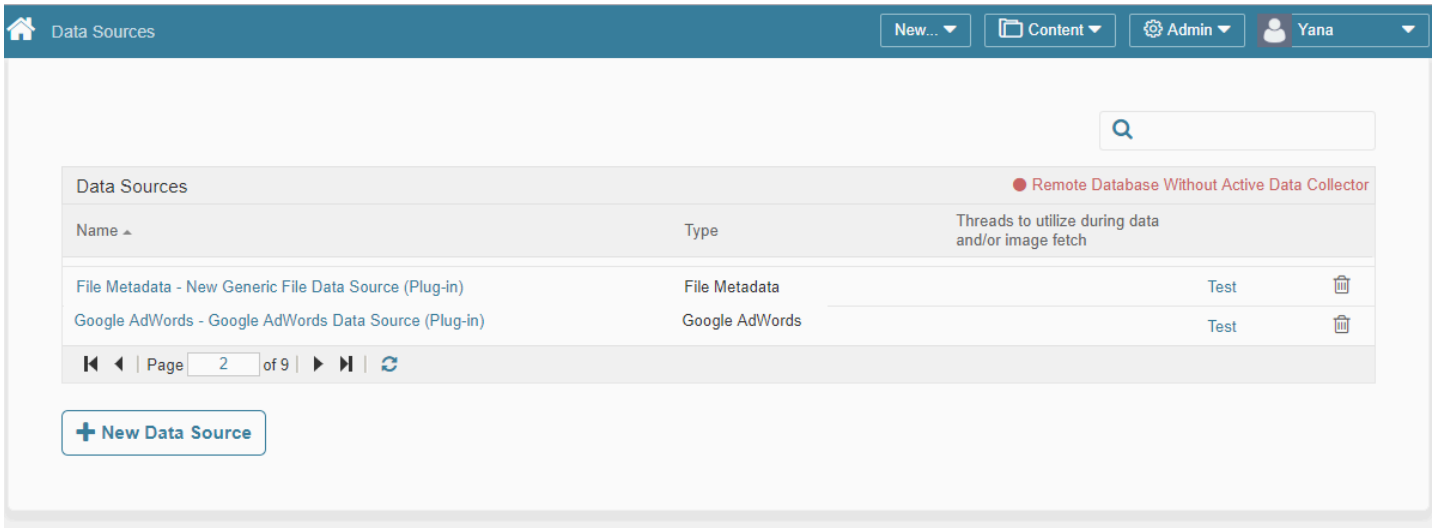
> Once filters are added to Metric / Report or External Report for the first time, they are going to be automatically added to all new respective elements of the same Data Source.

33. Sourcing Data from Splunk

33.1 Establish Connectivity to Splunk

This article describes the process of creating a plug-in Data Source to connect to Splunk. This Data Source will allow to build elements using Metric Insights tools.

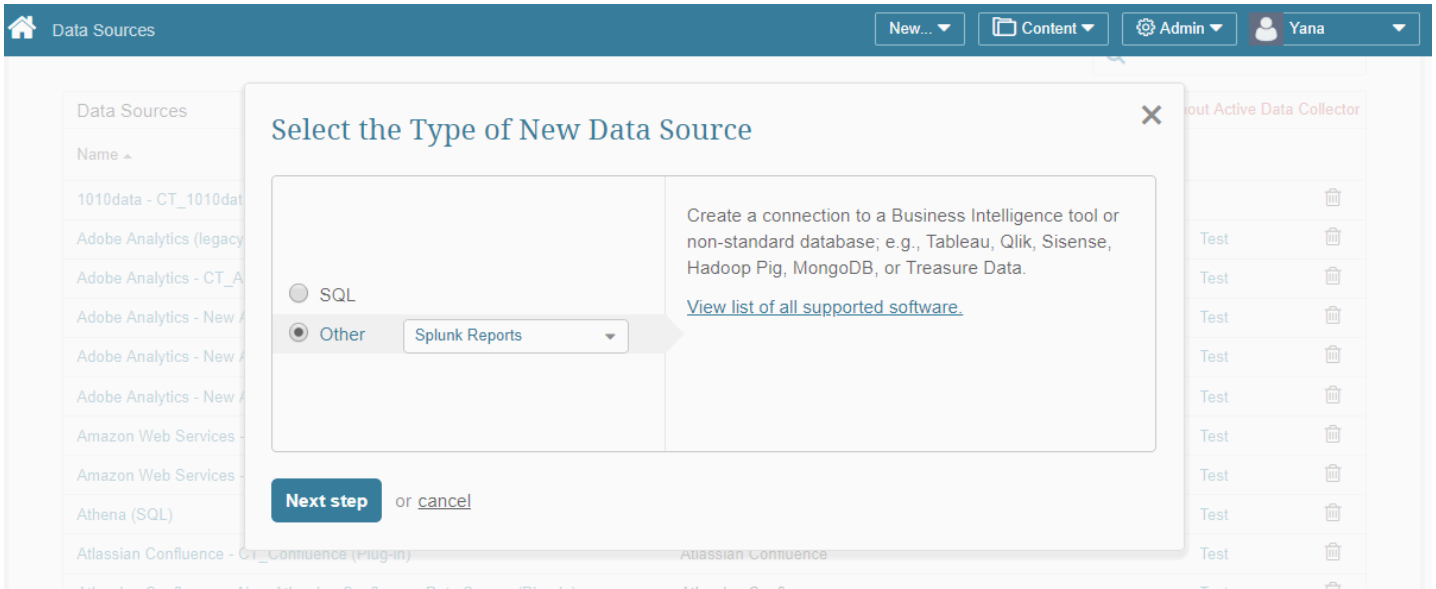
1. Add new Data Source



At the bottom of the screen click **[+ New Data Source].**

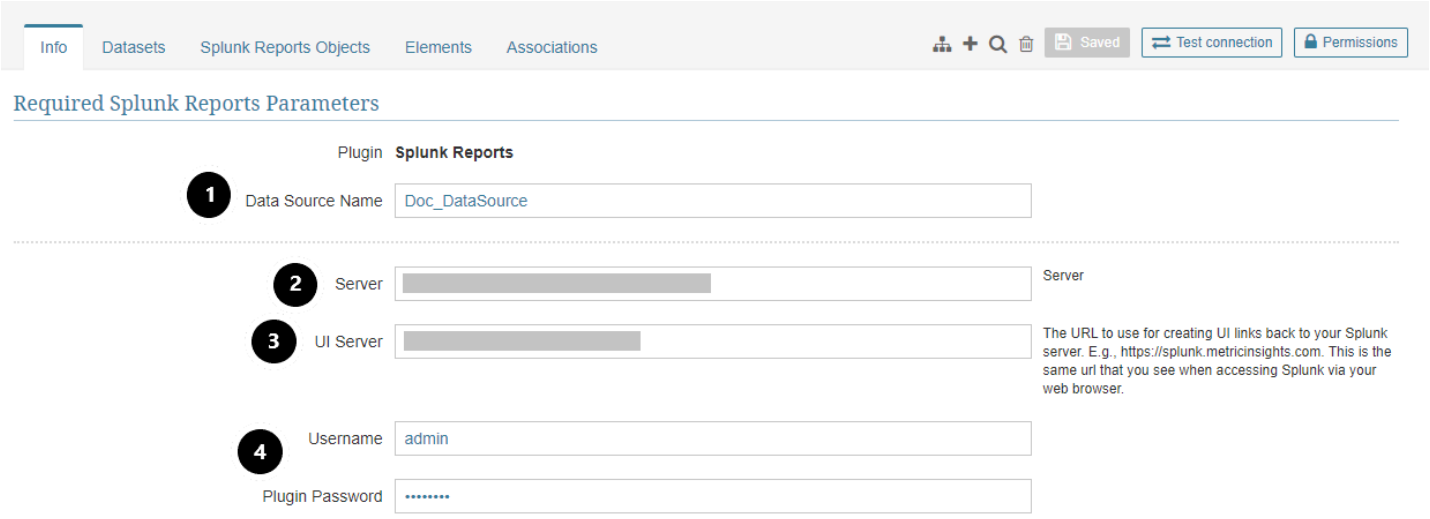
The *Select the Type of New Data Source* pop-up opens.

2. Select "Other" Data Source Type and choose "Splunk Reports" from the drop-down list



Move to the **Next step**.

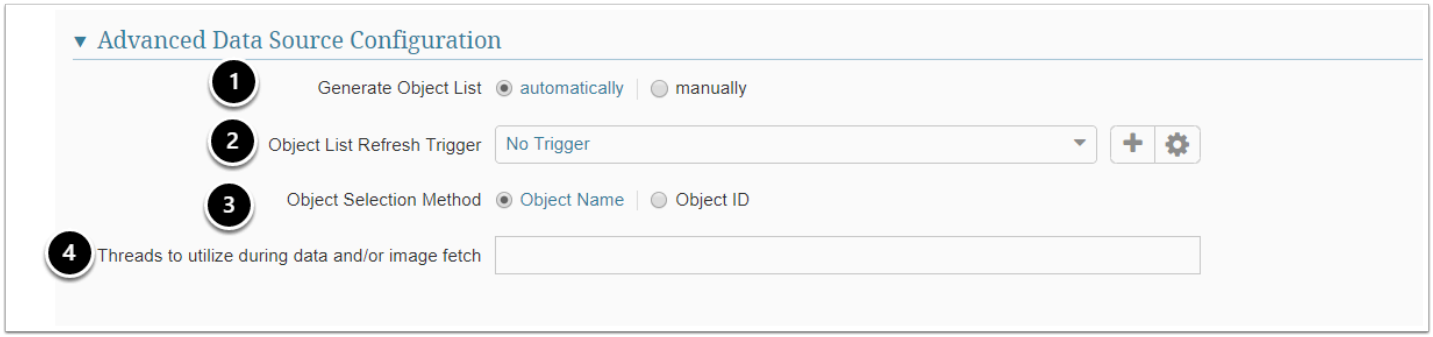
3. Provide the Required Parameters



1. **Data Source Name:** defaults to a unique name, but may be modified to a descriptive name of your own
2. **Server:** Define the server protocol (**http** or **https**) and a hostname to connect to the Splunk server
3. **UI Server**: Define the server protocol and name to build URLs for the External Reports to display the content in MI Iframes.

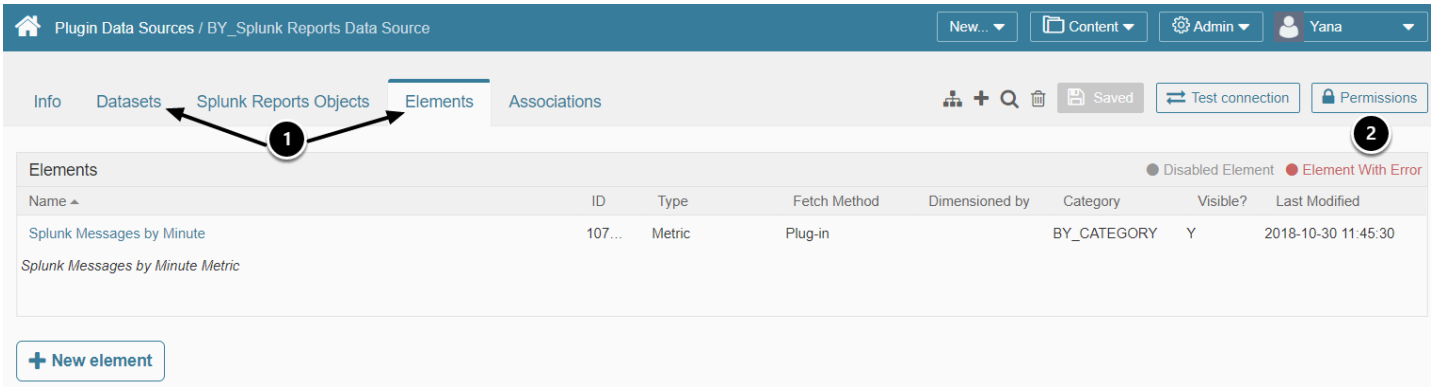
4. **Username / Password:** Note that your **Username** must be in the same format that your Splunk server uses for authentication
5. **Save** your entries and **Test Connection**

4. Advanced Configuration



1. **Generate Object List**
   - *automatically*: all Reports are going to be fetched by the system
   - *manually*: Reports may be added one-by-one or via CSV file
2. **Object List Refresh Trigger:** from the dropdown, select the Trigger that will be used to fetch data using the Splunk plugin
3. **Object Selection Method:** specify how Splunk Reports will be fetched
4. Optionally, state the maximum number of concurrent **Threads to utilize during data and/or image fetch** to be used in background processing when the system updates Reports for this Data Source. If you do not specify any value for this setting, batch data collection processing will be single-threaded

5. Full Data Source Editor displayed



1. You can create elements or Datasets directly from the respective tabs
2. Click **Permissions** to assign them to Groups or Power Users

What's next?

How to Collect Data from Splunk

33.2 How to Collect Data from Splunk

⚠ The option to build Legacy Reports with data sourced from Splunk will be available until you migrate to Metric Insights Version 6.X.

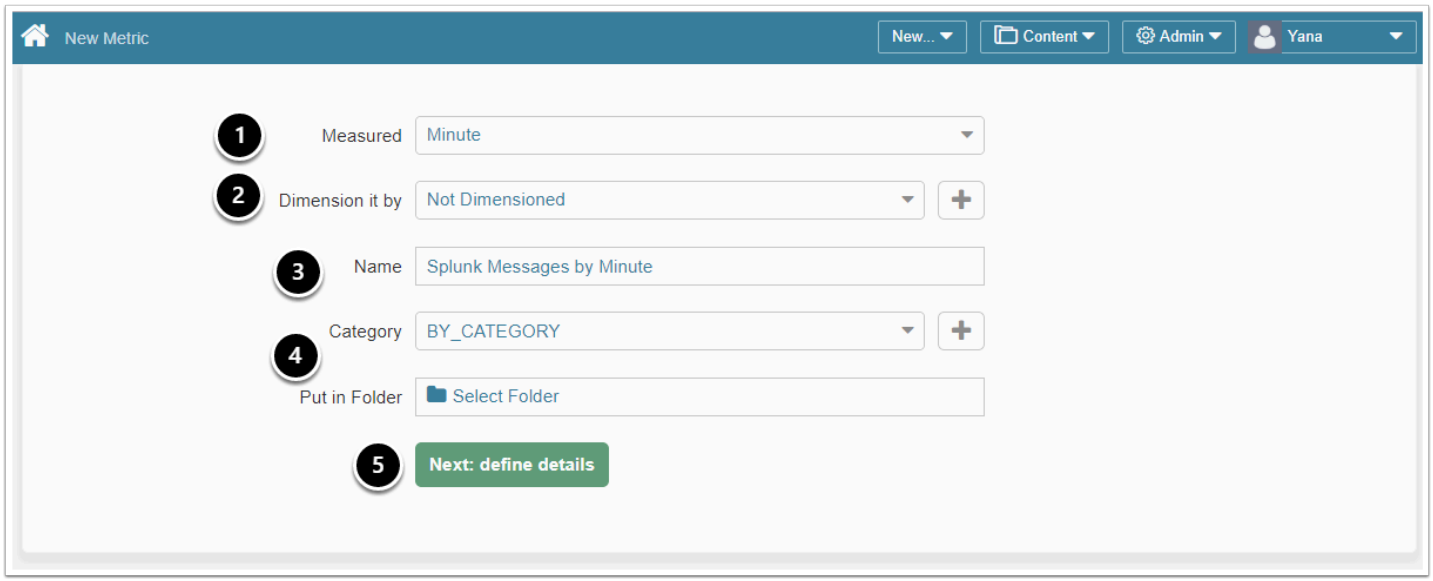
This article will show you how to create a Metric with data sourced from Splunk. The process of building other elements/Datasets is essentially the same.

1. Any **Saved Search** can be queried by the Splunk plugin and used as a data source in Metric Insights.
2. **Saved Searches** are accessible in Metric Insights via a dropdown in Element/Dataset Editors and displayed as a hierarchical list of Splunk Objects.
3. On Splunk Objects refresh the Splunk plugin fetches all the **Saved Searches** that are accessible for the User set in the Splunk Data Source Profile.

PREREQUISITES:

- You must have already established connectivity to the Splunk data source.
- Your Splunk version must be 6.x or 7.x

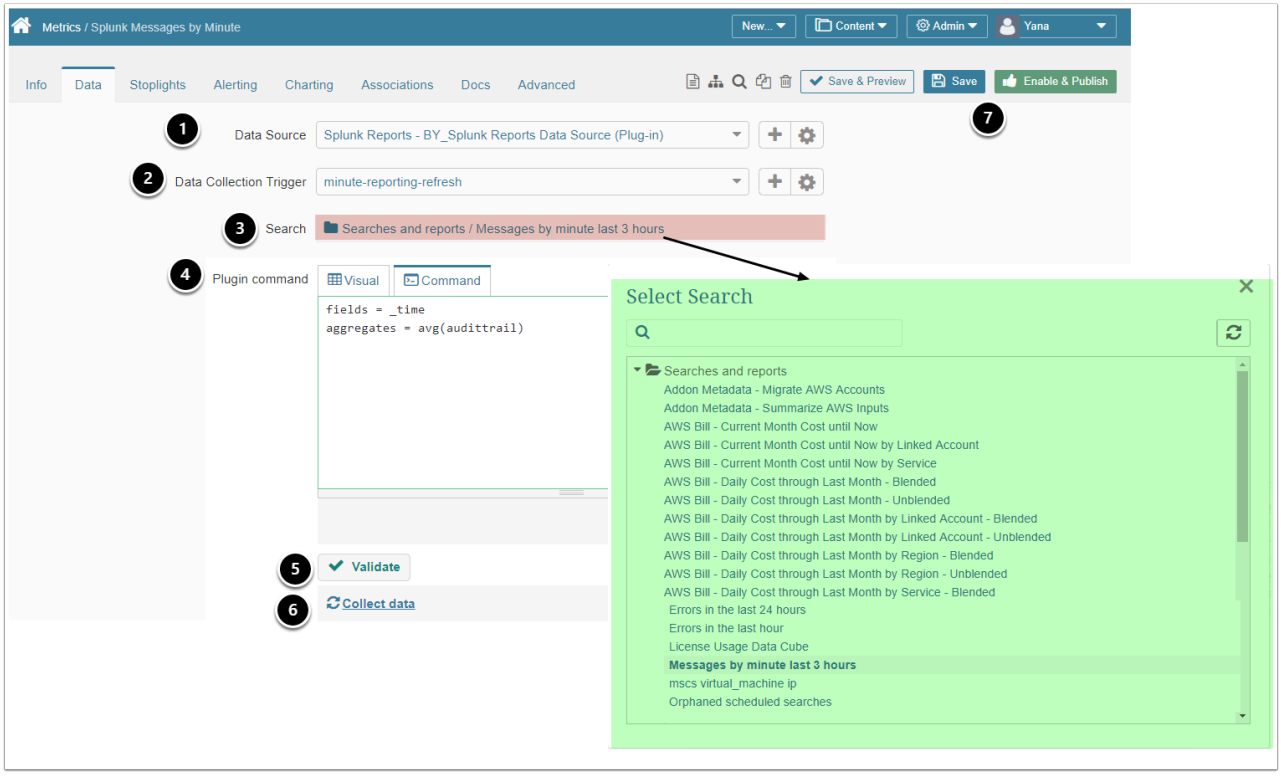
1. Access New > Metric



Provide the basic information required for creating a new metric:

1. Select the **Measurement Interval** that applies to your element
2. *Optionally*, select **Dimension**
3. Give the element a unique **Name**
4. *Optionally*, assign a **Category, Put in Folder**
5. Click **Next: define details** to proceed with data collection

2. Full Editor displays the Data Collection tab



1. Select the **Splunk** connection profile serving as a **Data Source** for this Metric
2. Set the **Data Collection Trigger** that is going to initiate updating information in this Metric
3. Specify a Splunk **Search** from the dropdown
4. Construct a **Plugin command** that should list the data you would like to include into the Metric (manually of via the **Visual Editor**)
   - Please note that Metrics represent time series data, so one of the columns should contain dates

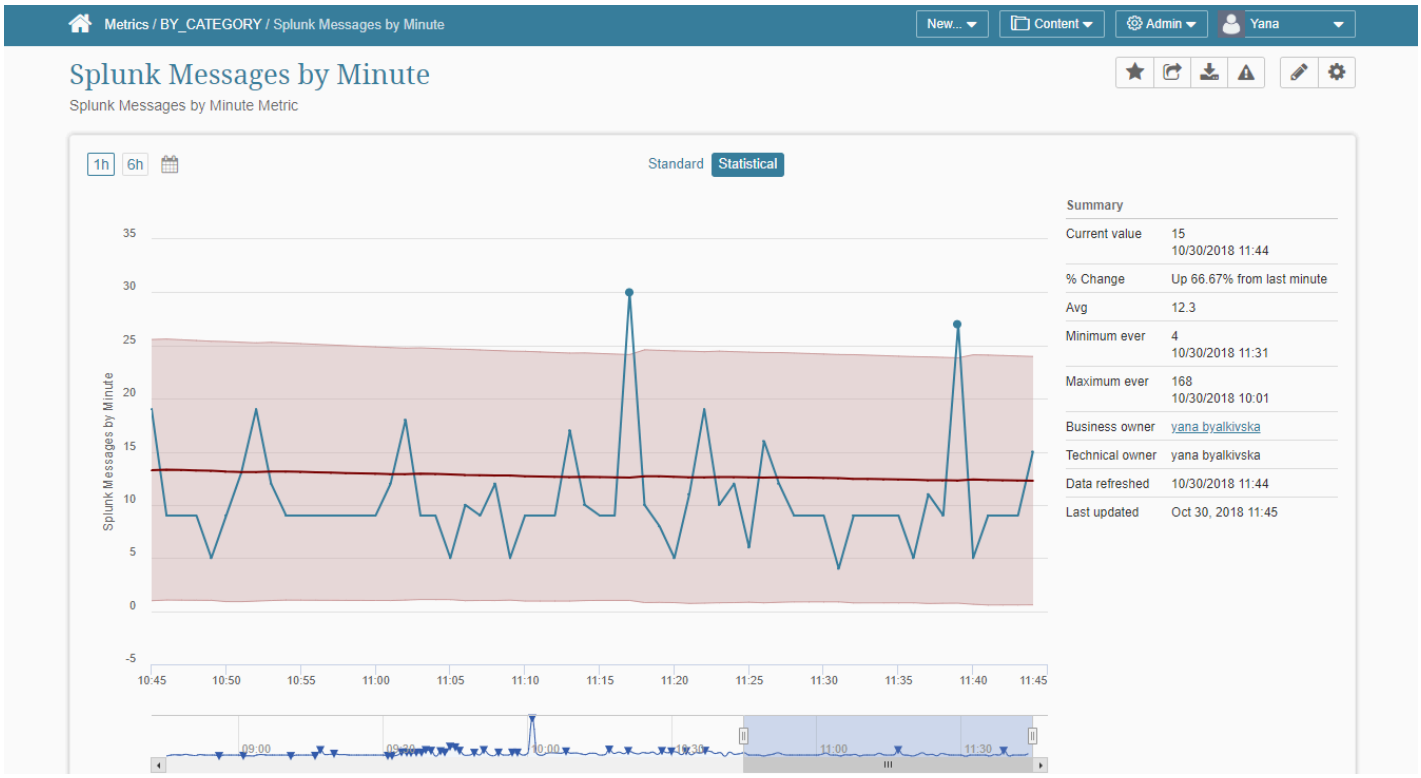
5. **Validate** your Plugin Command:
   - If your statement is valid, the statement box is **green**
   - If there are any errors, the box is colored in **red** and errors will be explained in the field below

6. **Collect Data**
7. **Save, Enable and Publish** your Metric

Plugin commands

To get more information on Splunk plugin syntax and full list of commands, refer to the Splunk command cheat sheet and search for commands by category.

3. Metric will be displayed in viewer

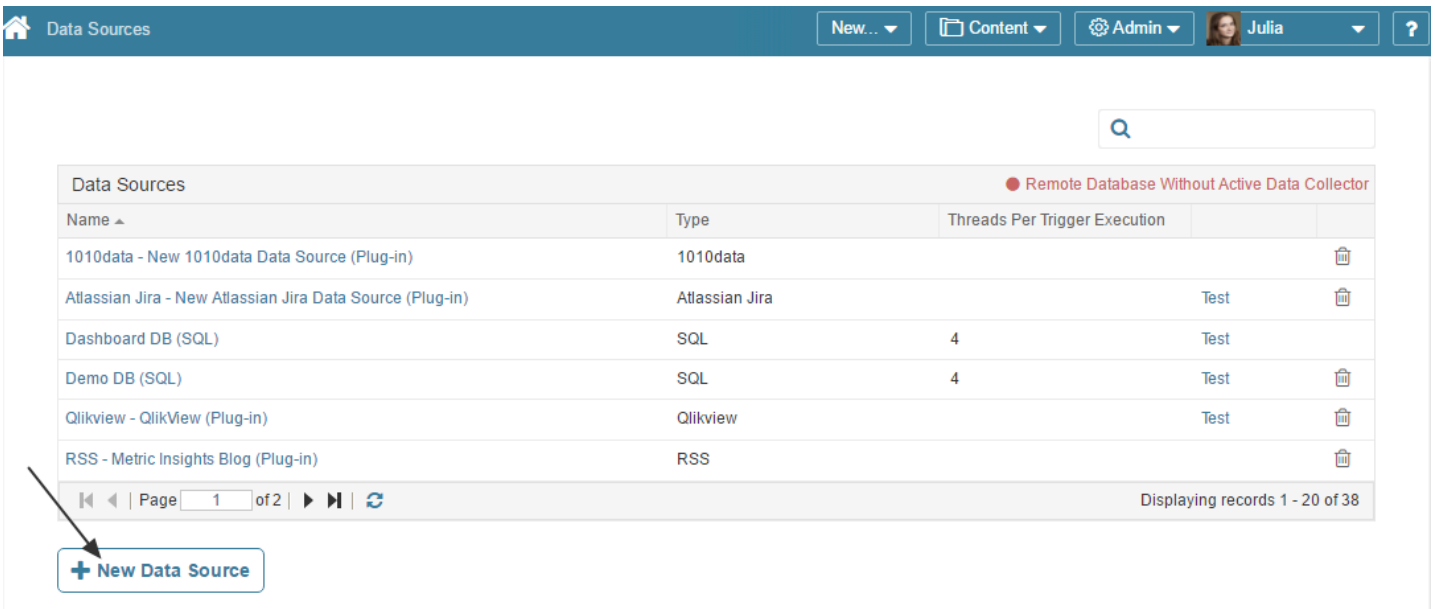


34. Sourcing Data from Microsoft SSRS

34.1 Establish Connectivity to Microsoft SSRS

This article describes the process of creating plug-in Data Source to connect to SSRS. This Data Source will allow data from existing SSRS objects to be used in building elements using Metric Insights.

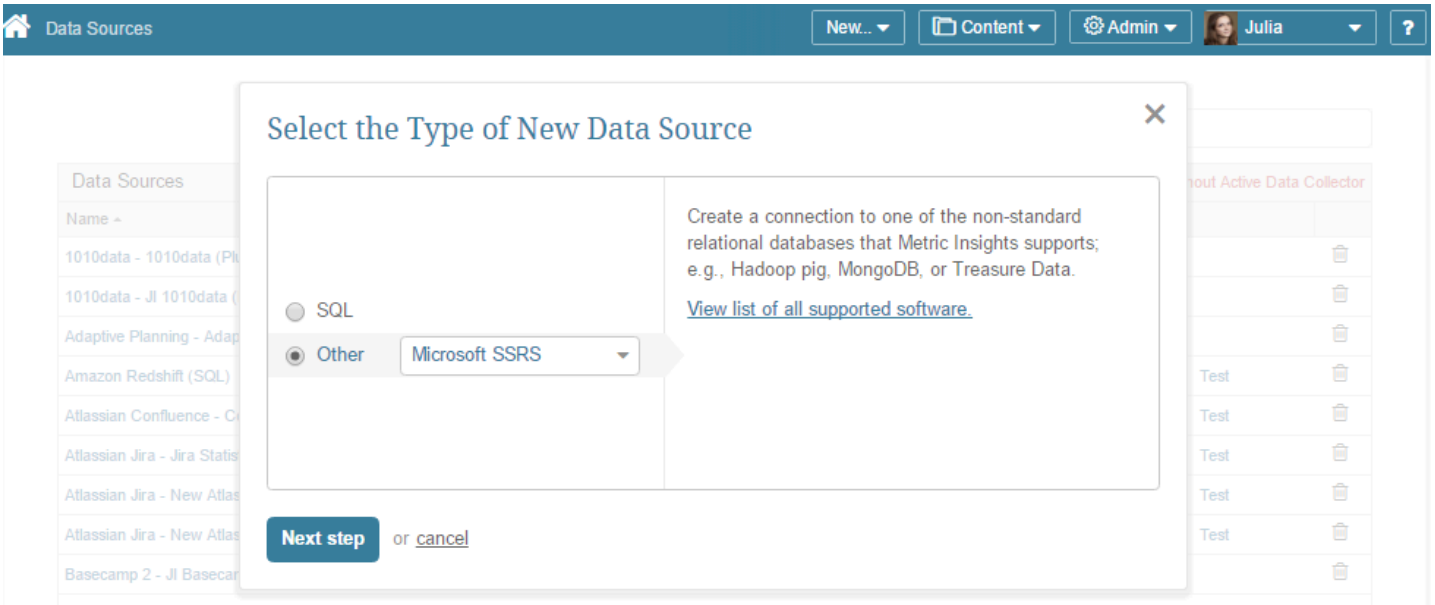
1. Access Admin > Data Sources



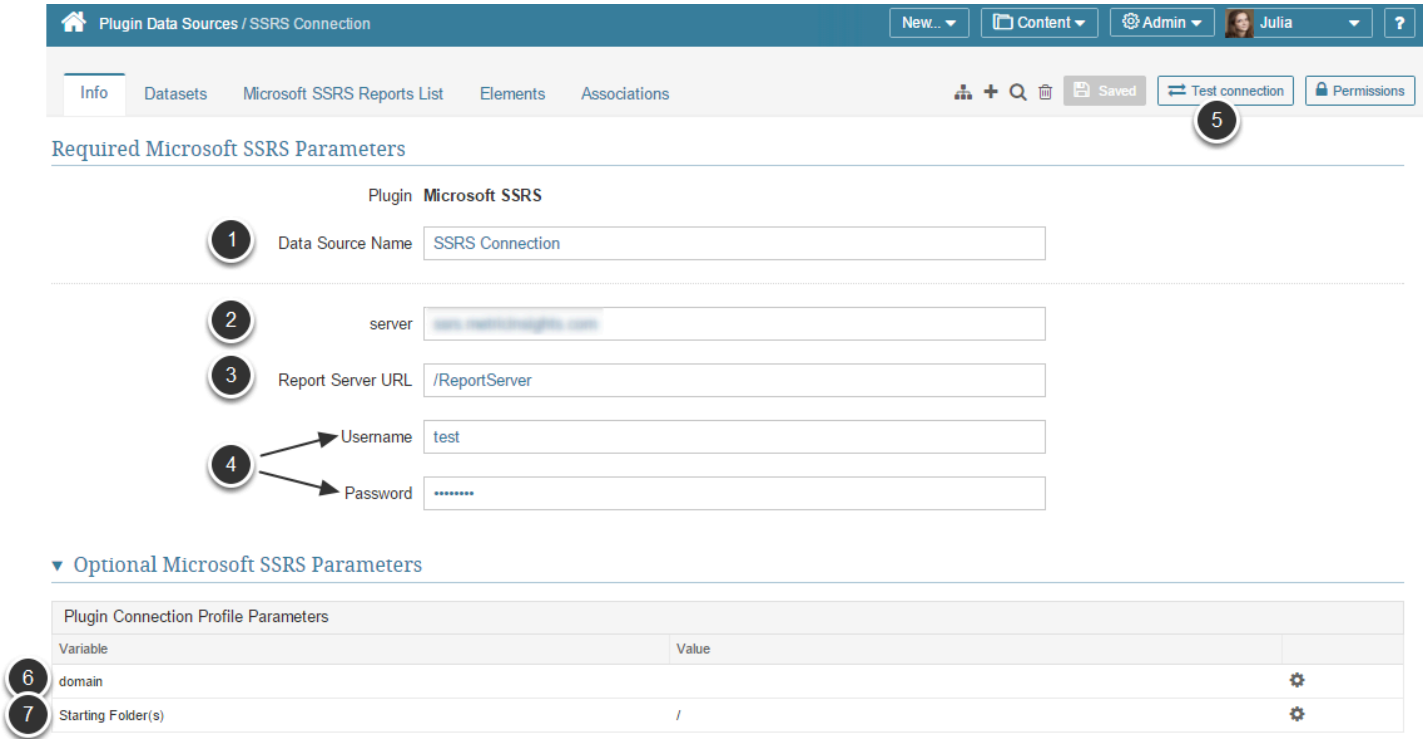
At the bottom of the screen click **[+ New Data Source].**

The *Select the Type of New Data Source* pop-up opens.

2. Select "Other" Data Source Type and choose "Microsoft SSRS" from the drop-down list



3. Provide the Required Microsoft SSRS Parameters



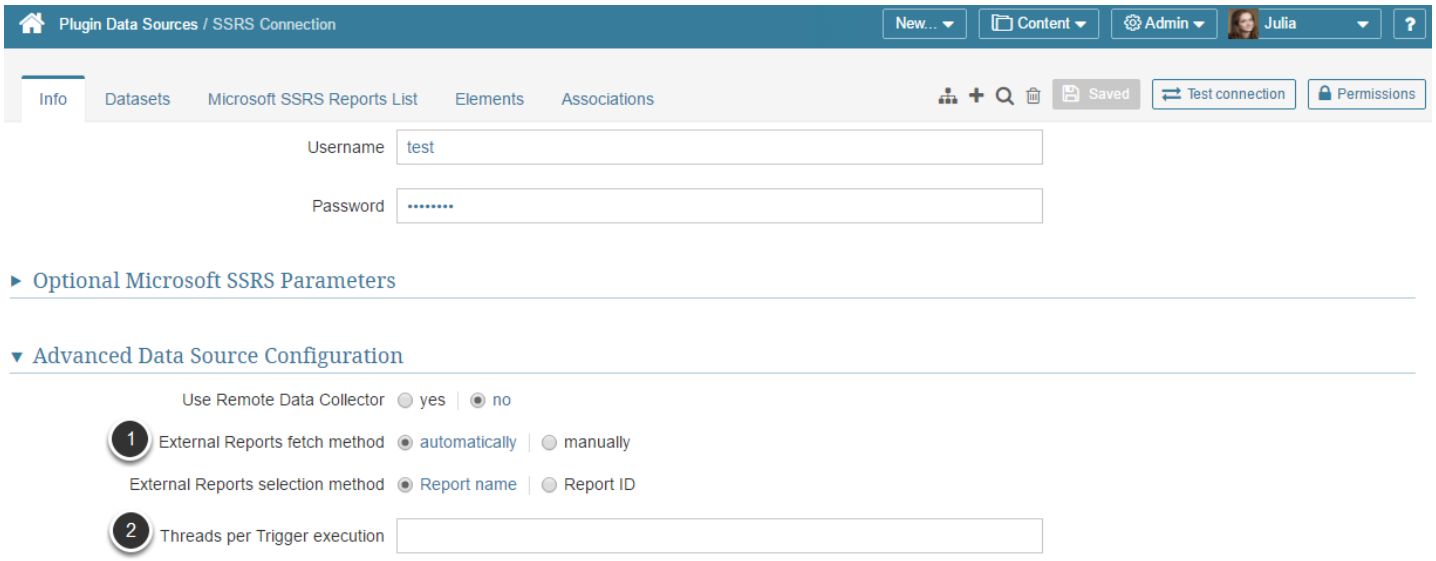
1. **Data Source Name:** defaults to a unique name, but may be modified to a descriptive name of your own.
2. **server:** the host name of your SSRS server
3. **Report Server URL:** the relative url for accessing your SSRS reports

4. **Username / Password:** This is the username and password you use to access **Microsoft SSRS**.
5. **Test Connection** (this will also Save)

Optional Microsoft SSRS Parameters:

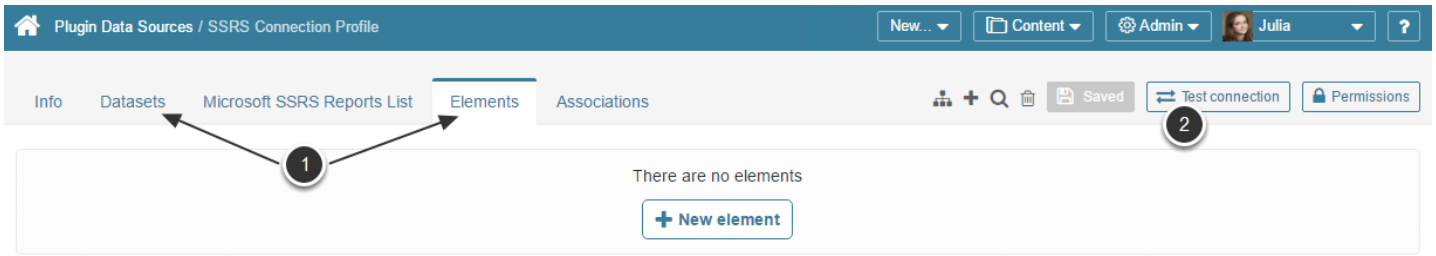
6. **domain:** Enter domain if SSRS login requires it
7. **Starting folder(s)**: Use this parameter if you want to specify specific SSRS folders only

4. Advanced Configuration



1. **External Reports fetch method**: This setting influences options available in the *Microsoft SSRS Reports List* tab:
   - **automatically**: just click **Refresh list** and all Reports are going to be fetched by the system
   - **manually**: Reports may be added one-by-one or via CSV file
2. Optionally, specify the maximum number of concurrent **Threads per Trigger execution** to be used in background processing when the system updates Metrics and Reports for this Data Source. If you do not specify any value for this setting, batch data collection processing will be single-threaded.

5. Other Settings



1. You can create elements or Datasets directly from the respective tabs
2. Click **Permissions** to assign them to Groups or Power Users

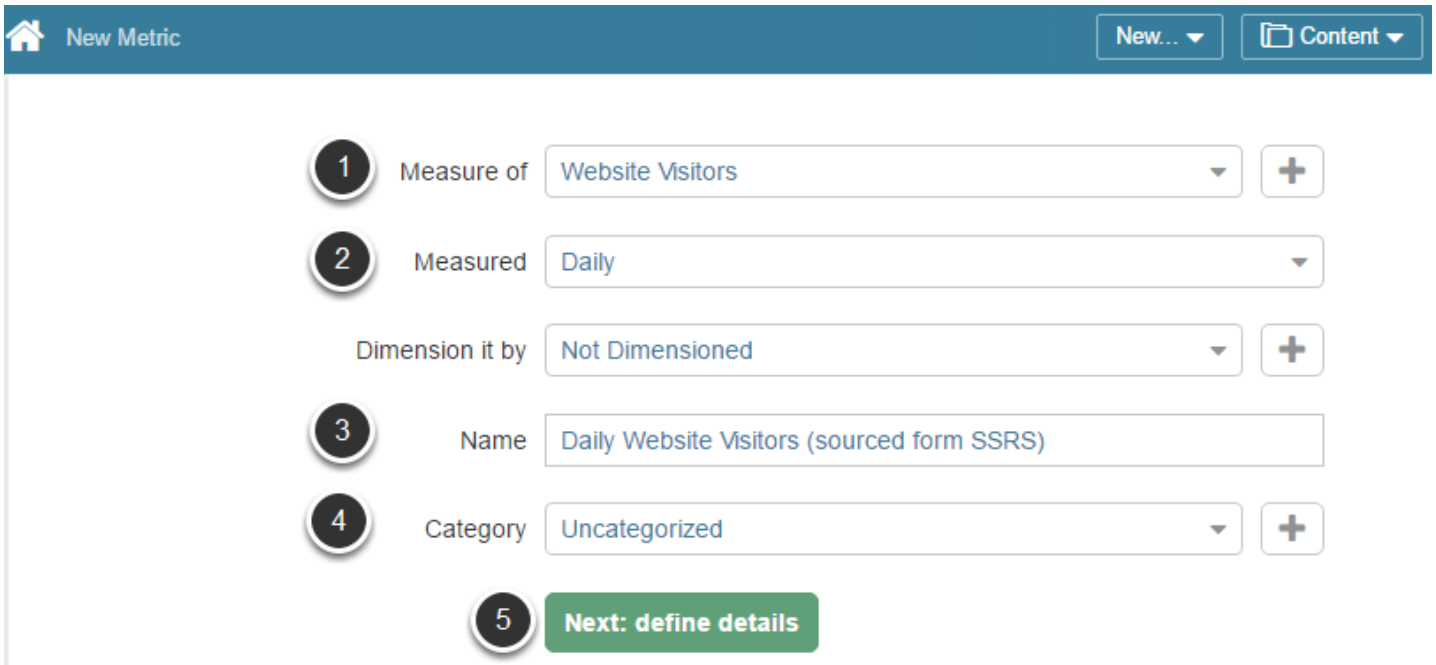
What's next?

How to Collect Data from SSRS

34.2 How to Collect Data from SSRS

This article will show you how to create a Metric or Internal Report using a SSRS as a data source. It assumes that you have already established connectivity to the SSRS data source desired.

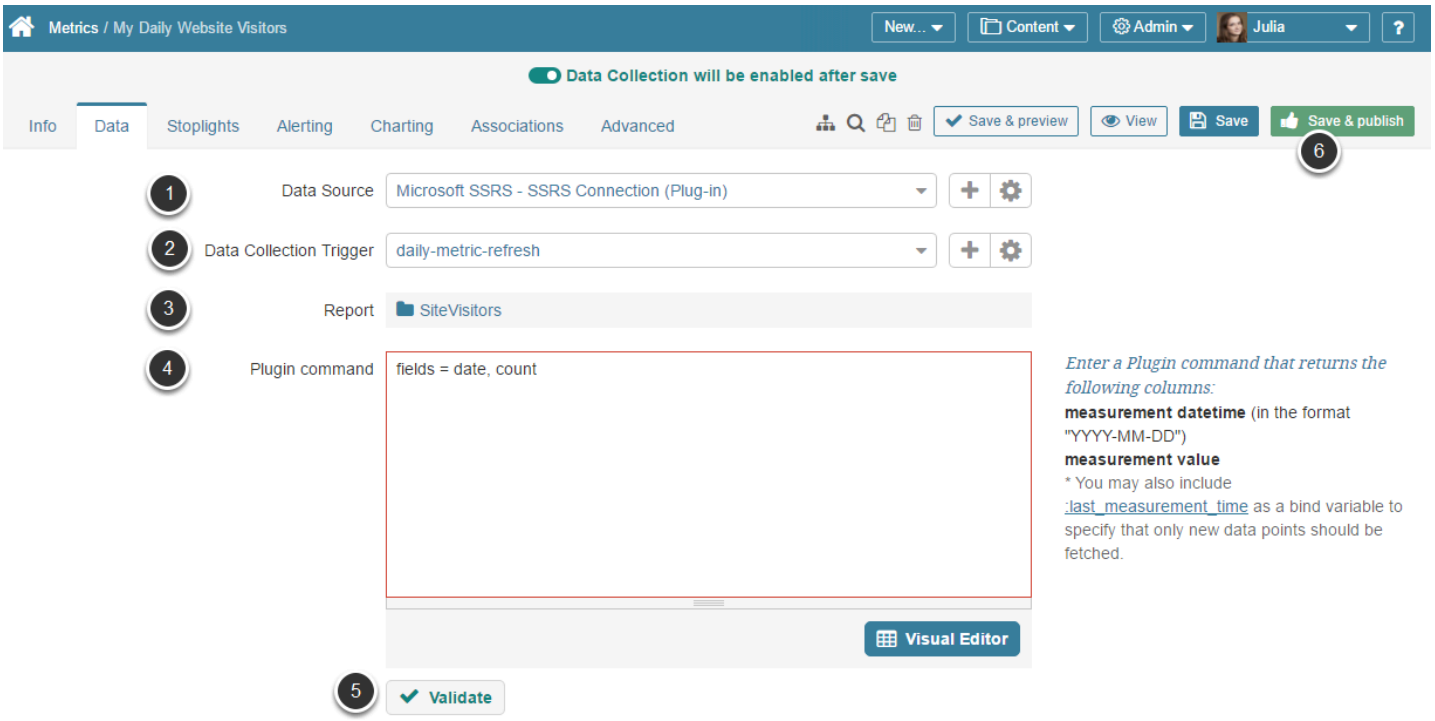
1. Access New > Metric



Provide the basic information required for creating a new metric:

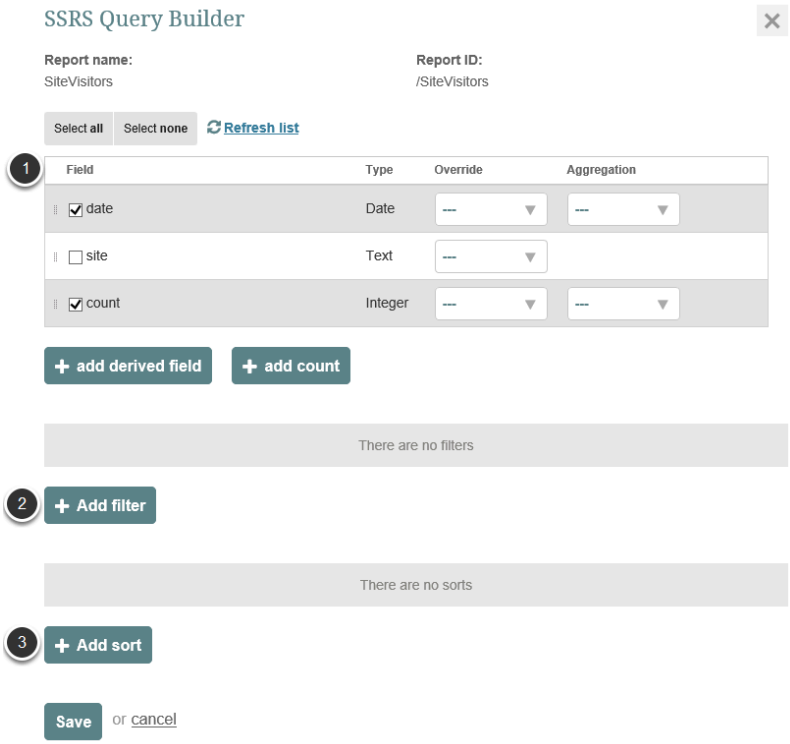
1. Specify this Metric's **Measure**. If you do not see the measure that you want to use, scroll down to the bottom of the drop-down list and create a new one by clicking **Add New Measure**
2. Select the **Measurement Interval** that applies to your element
3. Give the element a unique **Name**
4. *Optionally*, assign a **Category**
5. Click **Next: define details** to proceed with data collection

2. Full Editor displays the Data Collection tab



1. Select the **Microsoft SSRS** connection profile serving as a **Data Source** for this Metric
2. Set the **Data Collection Trigger** which is going to initiate updating information in a Metric
3. **Report:** Choose a Microsoft SSRS Report that should serve as a basis for a future Metric
4. Construct a **Plugin command** that should list the data you would like to include into the Metric (manually of via the **Visual Editor**). Please note that Metrics represent time series data, so one of the columns should contain dates.
5. **Validate** your Plug-in Command. If your statement is valid, the statement box is **green**; if there are any errors, the box is colored in **red** and errors will be explained in the field below.
6. **Collect Data** and **Enable and Publish**

2.1. Example using the Visual Editor

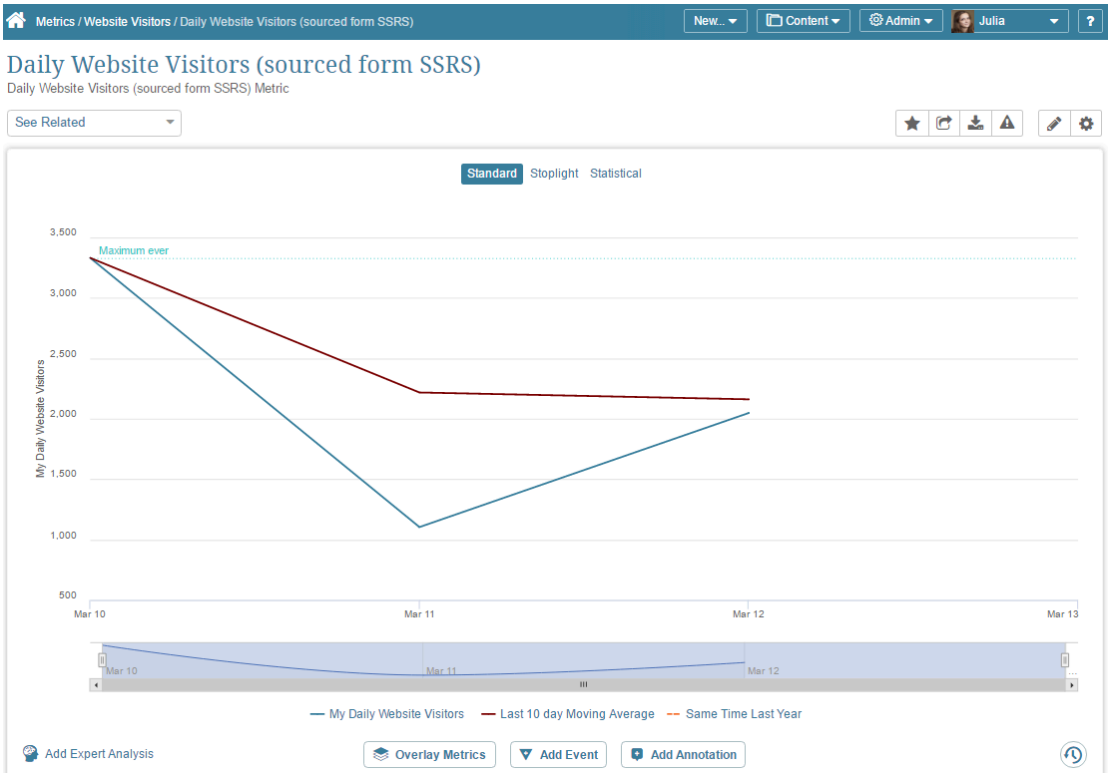


The *SSRS Query Builder* is called by Visual Editor link

1. Select **Fields** and set **Expressions**
2. You can pre-filter the information before fetching it into Metric Insights. To do so, add Filters in the *Query Builder*
3. Optionally you can add 'ascending' and 'descending' **Sorting** to the field values

Save your settings. Command validation will start automatically.

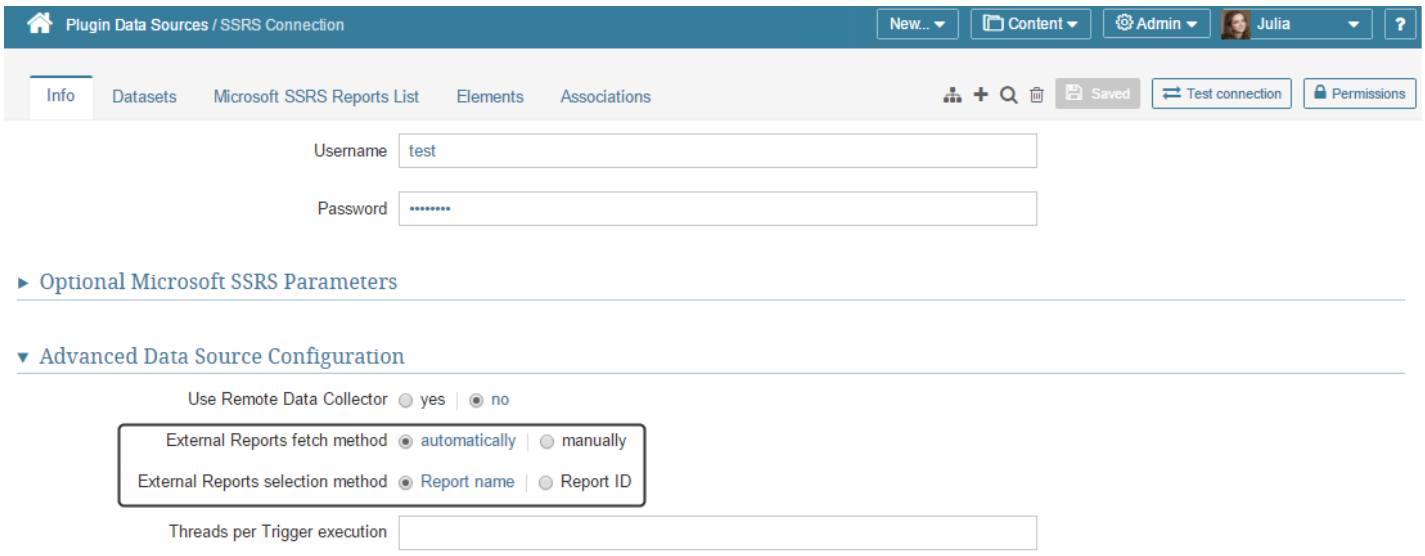
3. Metric will be displayed in viewer



34.3 How SSRS report data is obtained

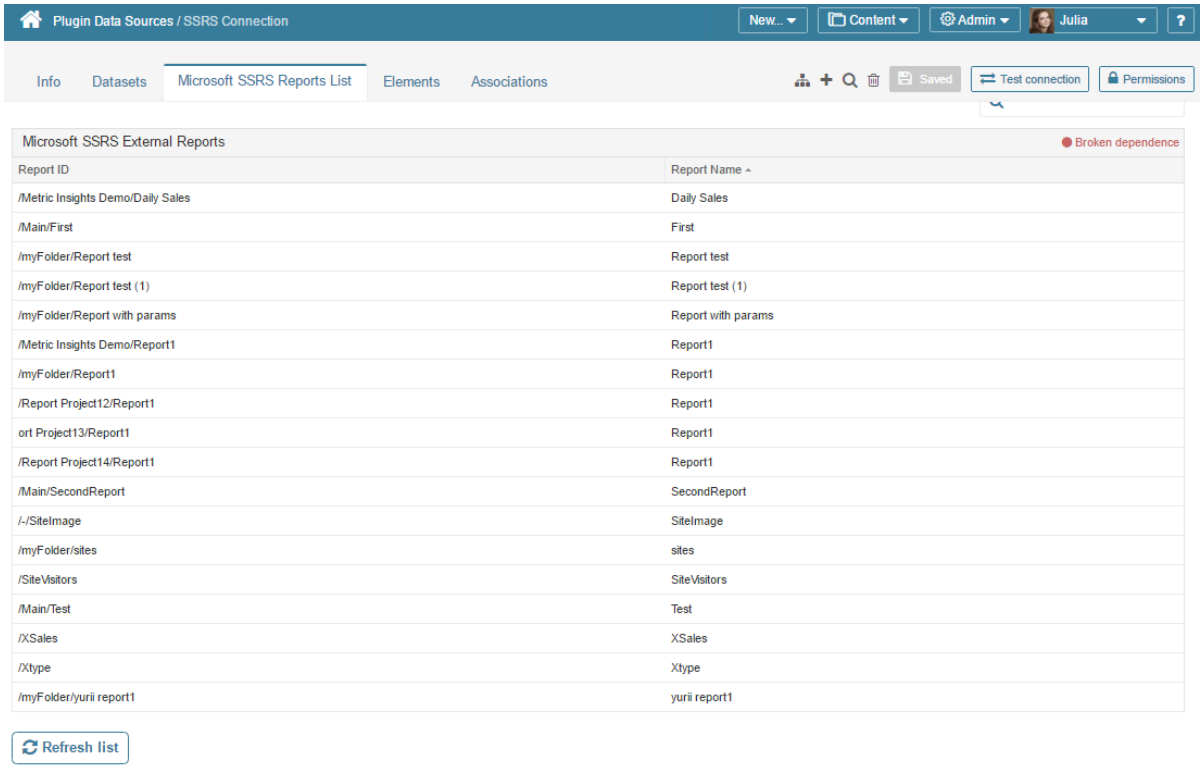
This article gives a behind the scenes look at how Metric Insights captures data from your SSRS reports. It covers how Metric Insights designates each SSRS report via an "id" and a name. It covers how Metric Insights identifies each chart or table within a SSRS report as a separate entity for pulling data. Finally, it covers how the fields are identified in each chart or table and how data is extracted.

SSRS report id and report name



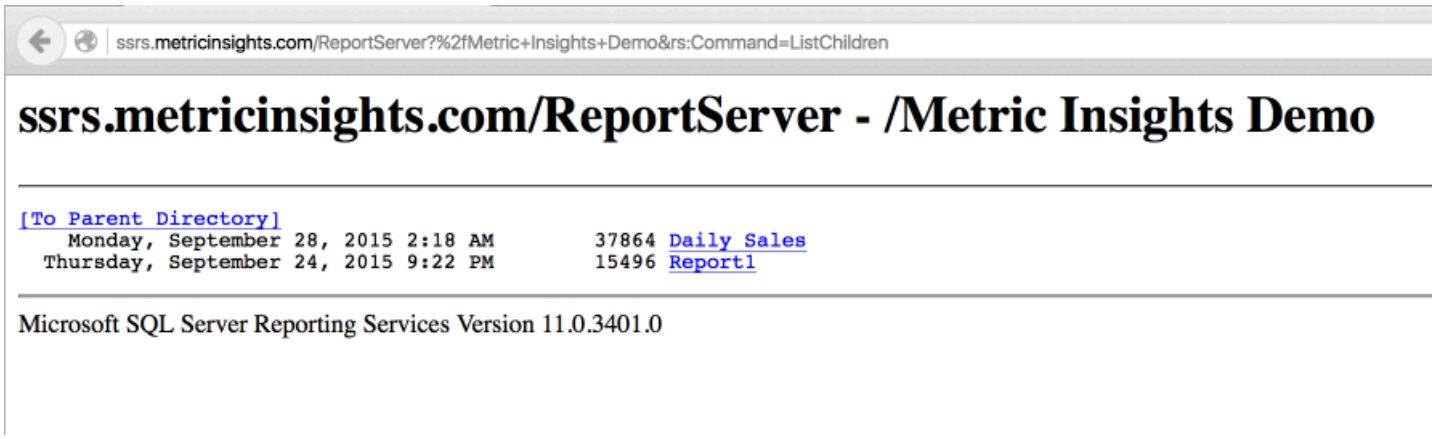
Metric Insights captures the 'report name' and the 'report id' of each report in SSRS, and gives you the option of which to show in the Report dropdown list in the Element Editor. This is the list of SSRS reports.

SSRS report id and report name



Here is an example listing of SSRS reports, designated by 'Report ID' and 'Report Name'. We will look specifically at the "/Metric Insights Demo/" reports.

How does Metric Insights get the list of SSRS reports



Metric Insights obtains the list of SSRS reports by navigating through the SSRS folders.

It pulls each 'report id' and 'report name' from relative url:

```
/ReportServer?</path/to/report>&rs:Command=ListChildren
```

e.g., for the "/Metric Insights Demo/" reports is uses:

http://ssrs.metricinsights.com/ReportServer?/Metric Insights Demo&rs:Command=ListChildren

In the resulting HTML (see screen shot) it then uses the url of each report for the 'report id', and the name of each report for the 'report name'.

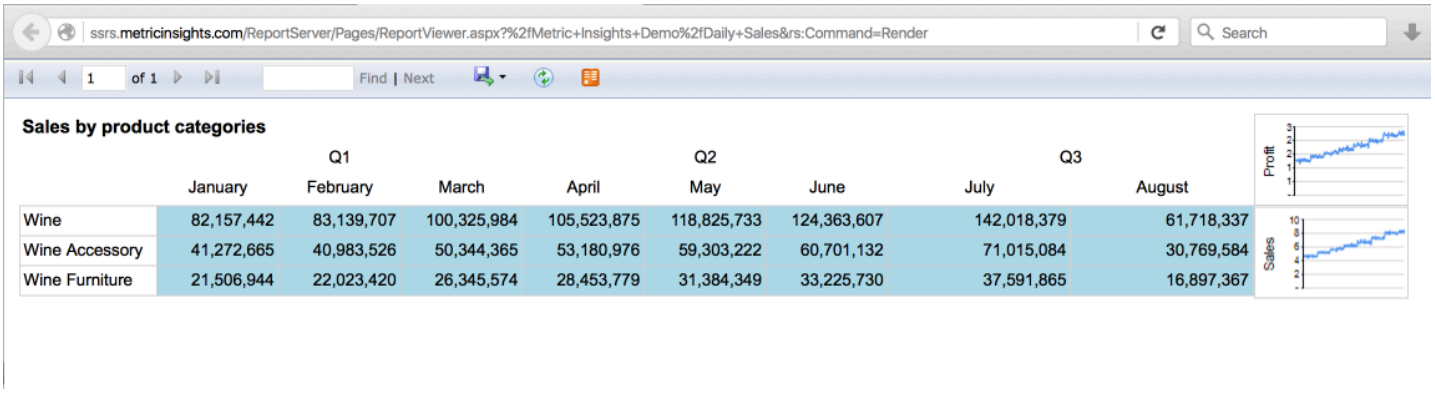
You can view the HTML results via View Source in your browser and you will see the link (<A HREF> </A>) to the report. In this example one report name is Daily Sales:

```
<A HREF="?%2fMetric+Insights+Demo%2fDaily+Sales&amp;rs:Command=Render">Daily Sales</A>
```

In this example, 'report id' is "/Metric Insights Demo/Daily Sales" (url decode of "%2fMetric+Insights+Demo%2fDaily+Sales").

And 'report name' is "Daily Sales".

More than one chart, table in SSRS report



If the SSRS report contains more than one chart or table, then Metric Insights will append the chart or table name to the 'report id' and 'report name'.

In this screen shot, the SSRS report contains one table and two charts (seen to the right).

In the next section we will describe how Metric Insights determines the internal name for the table and charts found in a SSRS report.

Get the name of each chart or table in SSRS report

File Path : ~/Downloads/Daily Sales.rdl

Daily Sales.rdl (no symbol selected)

```
44          <Field Name="sales">
45            <DataField>sales</DataField>
46            <rd:TypeName>System.Decimal</rd:TypeName>
47          </Field>
48        </Fields>
49      </DataSet>
50      <DataSet Name="DataSet3">
51        <SharedDataSet>
52          <SharedDataSetReference>DataSet3</SharedDataSetReference>
53        </SharedDataSet>
54        <Fields>
55          <Field Name="country">
56            <DataField>country</DataField>
57            <rd:TypeName>System.String</rd:TypeName>
58          </Field>
59          <Field Name="calendar_date">
60            <DataField>calendar_date</DataField>
61            <rd:TypeName>System.DateTime</rd:TypeName>
62          </Field>
63          <Field Name="sum_total_gross_profit_">
64            <DataField>sum_total_gross_profit_</DataField>
65            <rd:TypeName>System.Decimal</rd:TypeName>
66          </Field>
67          <Field Name="sum_total_sales_amount_">
68            <DataField>sum_total_sales_amount_</DataField>
69            <rd:TypeName>System.Decimal</rd:TypeName>
70          </Field>
71          <Field Name="month">
72            <Value>=monthname(month(Fields!calendar_date.Value))</Value>
73          </Field>
74        </Fields>
75      </DataSet>
76    </DataSets>
77    <ReportSections>
78      <ReportSection>
79        <Body>
80          <ReportItems>
81            <Tablix Name="Tablix1">
82              <TablixCorner>
```

Metric Insights obtains the chart or table names for each SSRS report via the Report Definition Language (RDL).

You can see the RDL for the report by appending "&SelectedTabId=PropertiesTab&Export=true" to the url. For example:

http://ssrs.metricinsights.com/Reports/Pages/Report.aspx?ItemPath=/Metric Insights Demo/Daily Sales&SelectedTabId=PropertiesTab&Export=true

1. The name of the table in this example is determined to be "Tablix1.quarter.month**"** (the full table name is constructed from more than one XML element in the RDL)

Thus, 'report id' for "/Metric Insights Demo/Daily Sales" for the table becomes "/Metric Insights Demo/Daily Sales&Tablix1.quarter.month", where it includes the table name of "Tablix1.quarter.month". And 'report name' is "Daily Sales&Tablix1.quarter.month".

Metric Insights will then be able to pull data for this table. Subsequently, it can do the same for the two charts as well.

Get the name of each chart or table in SSRS report


```
- <ReportSections>
  - <ReportSection>
    - <Body>
      - <ReportItems>
        - <Tablix Name="Tablix1">
          + <TablixCorner></TablixCorner>
          + <TablixBody></TablixBody>
          - <TablixColumnHierarchy>
            - <TablixMembers>
              - <TablixMember>
                + <Group Name="quarter"></Group>
                + <SortExpressions></SortExpressions>
                + <TablixMembers></TablixMembers>
                </TablixMember>
              </TablixMembers>
            </TablixColumnHierarchy>
          + <TablixRowHierarchy></TablixRowHierarchy>
            <DataSetName>DataSet2</DataSetName>
            <Top>0.375in</Top>
            <Left>0.125in</Left>
            <Height>0.775in</Height>
            <Width>2.16166in</Width>
          + <Style></Style>
          </Tablix>
        - <Chart Name="Chart2">
          - <ChartCategoryHierarchy>
            - <ChartMembers>
              - <ChartMember>
                + <Group Name="Chart2_CategoryGroup"></Group>
                + <SortExpressions></SortExpressions>
                  <Label>=Fields!calendar_date.Value</Label>
                </ChartMember>
              </ChartMembers>
            </ChartCategoryHierarchy>
          + <ChartSeriesHierarchy></ChartSeriesHierarchy>
```


1. In this example of the RDL for SSRS report "/Metric Insights Demo/Daily Sales", you can also see the "Chart2.Chart2_CategoryGroup" chart name (the full chart name is constructed from more than one XML element in the RDL)

Get list of fields in SSRS report


```
File Path : ~/Downloads/Daily Sales table.atom
Daily Sales table.a...  (no symbol selected)
<?xml version="1.0" encoding="utf-8" standalone="yes"?><feed
xmlns:d="http://schemas.microsoft.com/ado/2007/08/dataservices"
xmlns:m="http://schemas.microsoft.com/ado/2007/08/dataservices/metadata"
xmlns="http://www.w3.org/2005/Atom"><title
type="text"></title><id>uuid:2666683c-8f98-4225-9d0d-36f45f9f2e42;id=1</id><updated>2016-
01-20T15:10:33Z</updated><entry><id>uuid:2666683c-8f98-4225-9d0d-36f45f9f2e42;id=2</id><
title type="text"></title><updated>2016-01-20T15:10:33Z</updated><author /><content
type="application/xml"><m:properties><d:product_category>Wine</d:product_category><d:
quarter1>Q1</d:quarter1><d:month>January</d:month><d:sales
m:type="Edm.Decimal">82157442.16</d:sales></m:properties></content></entry><entry><id>uuid
:2666683c-8f98-4225-9d0d-36f45f9f2e42;id=3</id><title
type="text"></title><updated>2016-01-20T15:10:33Z</updated><author /><content
type="application/xml"><m:properties><d:product_category>Wine</d:product_category><d:
quarter1>Q1</d:quarter1><d:month>February</d:month><d:sales
m:type="Edm.Decimal">83139706.75</d:sales></m:properties></content></entry><entry><id>uuid
:2666683c-8f98-4225-9d0d-36f45f9f2e42;id=4</id><title
type="text"></title><updated>2016-01-20T15:10:33Z</updated><author /><content
type="application/xml"><m:properties><d:product_category>Wine</d:product_category><d:
quarter1>Q1</d:quarter1><d:month>March</d:month><d:sales
m:type="Edm.Decimal">100325984.41</d:sales></m:properties></content></entry><entry><id>
uuid:2666683c-8f98-4225-9d0d-36f45f9f2e42;id=5</id><title
type="text"></title><updated>2016-01-20T15:10:33Z</updated><author /><content
type="application/xml"><m:properties><d:product_category>Wine</d:product_category><d:
quarter1>Q2</d:quarter1><d:month>April</d:month><d:sales
m:type="Edm.Decimal">105523875.36</d:sales></m:properties></content></entry><entry><id>
uuid:2666683c-8f98-4225-9d0d-36f45f9f2e42;id=6</id><title
type="text"></title><updated>2016-01-20T15:10:33Z</updated><author /><content
type="application/xml"><m:properties><d:product_category>Wine</d:product_category><d:
quarter1>Q2</d:quarter1><d:month>May</d:month><d:sales
m:type="Edm.Decimal">118825732.63</d:sales></m:properties></content></entry><entry><id>
uuid:2666683c-8f98-4225-9d0d-36f45f9f2e42;id=7</id><title
type="text"></title><updated>2016-01-20T15:10:33Z</updated><author /><content
type="application/xml"><m:properties><d:product_category>Wine</d:product_category><d:
quarter1>Q2</d:quarter1><d:month>June</d:month><d:sales
m:type="Edm.Decimal">124363607.42</d:sales></m:properties></content></entry><entry><id>
uuid:2666683c-8f98-4225-9d0d-36f45f9f2e42;id=8</id><title
type="text"></title><updated>2016-01-20T15:10:33Z</updated><author /><content
type="application/xml"><m:properties><d:product_category>Wine</d:product_category><d:
quarter1>Q3</d:quarter1><d:month>July</d:month><d:sales
```

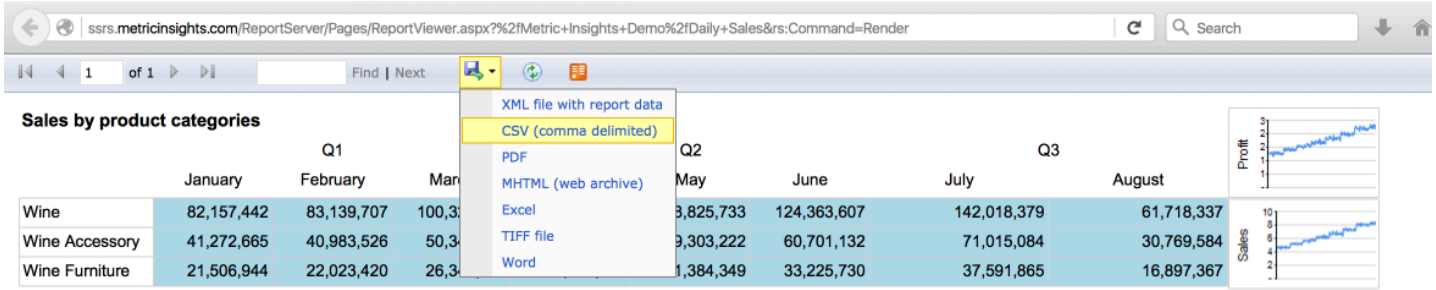

Metric Insights obtains names and data types for each field in a chart or table via the SSRS ATOM file for that chart or table.

You can see the ATOM file from the report chart or table by appending "&rs:Command=Render&rs:Format=ATOM&rc:ItemPath=<chart or table>" to the url. For example:

http://ssrs.metricinsights.com/ReportServer?/Metric Insights Demo/Daily Sales&rs:Command=Render&rs:Format=ATOM&rc:ItemPath=Tablix1.quarter.month

Metric Insights also uses the ATOM file to get the data for each chart or table in a SSRS report.

Get data from SSRS report



Metric Insights obtains the data for each chart or table via the SSRS ATOM file as in previous example.

However, you can see the similar data when you manually export the report to CSV.

Get data from SSRS report when more than one chart, table in SSRS report

Daily Sales data.csv

```
sum total gross profit ,8/10/2015 12:00:00 AM,2373563.83
sum total gross profit ,8/11/2015 12:00:00 AM,2331347.84
sum total gross profit ,8/12/2015 12:00:00 AM,2350774.03
sum total gross profit ,8/13/2015 12:00:00 AM,2204531.87

product_category,quarter1,month,sales
Wine,Q1,January,"82,157,442"
Wine,Q1,February,"83,139,707"
Wine,Q1,March,"100,325,984"
Wine,Q2,April,"105,523,875"
Wine,Q2,May,"118,825,733"
Wine,Q2,June,"124,363,607"
Wine,Q3,July,"142,018,379"
Wine,Q3,August,"61,718,337"
Wine Accessory,Q1,January,"41,272,665"
Wine Accessory,Q1,February,"40,983,526"
Wine Accessory,Q1,March,"50,344,365"
Wine Accessory,Q2,April,"53,180,976"
Wine Accessory,Q2,May,"59,303,222"
Wine Accessory,Q2,June,"60,701,132"
Wine Accessory,Q3,July,"71,015,084"
Wine Accessory,Q3,August,"30,769,584"
Wine Furniture,Q1,January,"21,506,944"
Wine Furniture,Q1,February,"22,023,420"
Wine Furniture,Q1,March,"26,345,574"
Wine Furniture,Q2,April,"28,453,779"
Wine Furniture,Q2,May,"31,384,349"
Wine Furniture,Q2,June,"33,225,730"
Wine Furniture,Q3,July,"37,591,865"
Wine Furniture,Q3,August,"16,897,367"

sum total sales amount _label,sum total sales amount _Chart2_CategoryGroup2_label,sum total sales
amount _Chart2_CategoryGroup2_Value_Y
sum total sales amount ,1/1/2015 12:00:00 AM,4686558.65
sum total sales amount ,1/2/2015 12:00:00 AM,4734442.25
sum total sales amount ,1/3/2015 12:00:00 AM,4645611.02
sum total sales amount ,1/4/2015 12:00:00 AM,4670093.86
sum total sales amount ,1/5/2015 12:00:00 AM,4563635.59
sum total sales amount ,1/6/2015 12:00:00 AM,4836325.63
sum total sales amount ,1/7/2015 12:00:00 AM,4755772.39
sum total sales amount ,1/8/2015 12:00:00 AM,4607848.05
sum total sales amount ,1/9/2015 12:00:00 AM,4677732.15
sum total sales amount ,1/10/2015 12:00:00 AM,4409527.88
sum total sales amount ,1/11/2015 12:00:00 AM,4860540.65
```

In this example, the SSRS report has one table and two charts.

If your SSRS report has more than one chart or table in it, then you will see a group of data in your exported CSV for each chart or table. In this example, the middle section of the CSV contains the data for the table.

Note: This is why Metric Insights identifies each chart or table in your SSRS report for data pull. Each chart or table contains a different set of data. And Metric Insights pulls the data via the ATOM file instead of CSV as explained earlier.

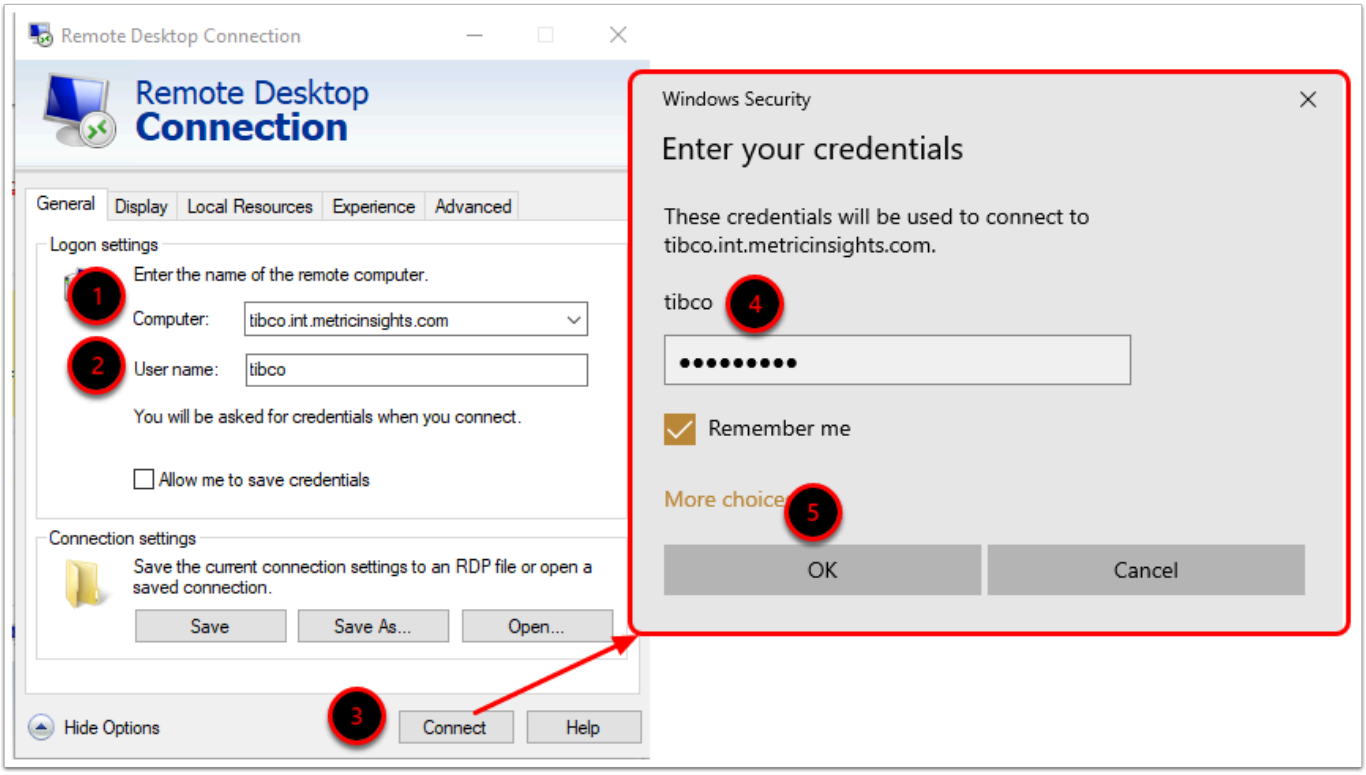
36. Sourcing data from Tibco Spotfire

36.1 Prerequisites for connecting to Tibco Spotfire

To integrate with Tibco Spotfire, particularly for getting data, you must deploy Metric Insights' **Remote Data Collector (RDC)** to your server and install a custom **MIAddIn package** (extension) on a Spotfire server.

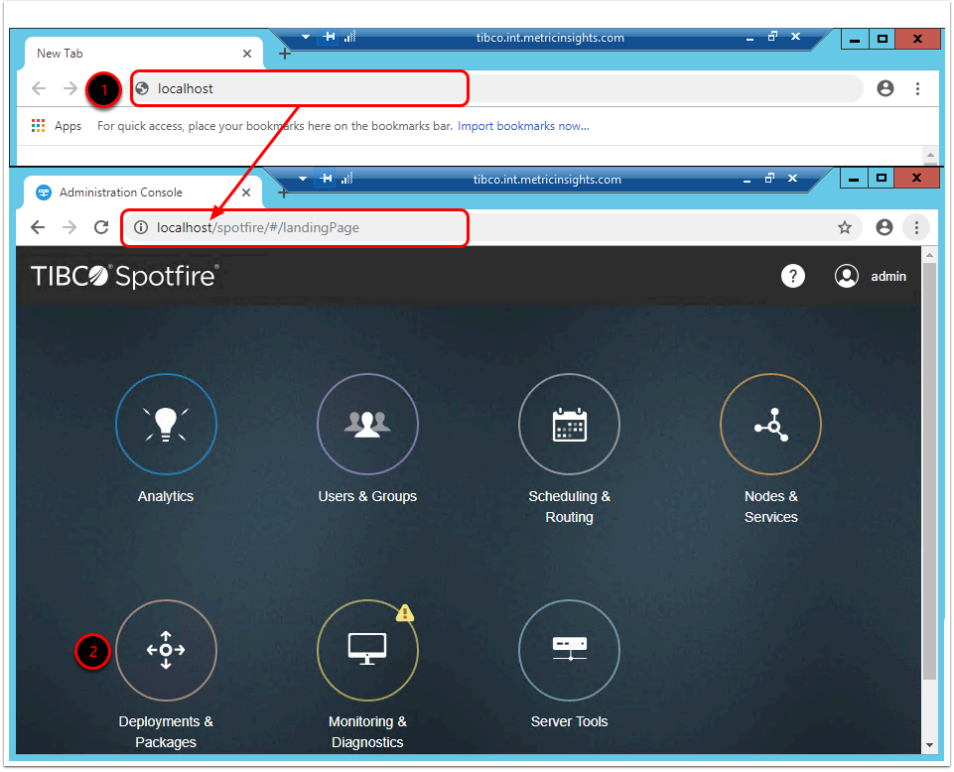
This article assumes that you have already configured a Remote Data Collector profile in the Metric Insights UI and installed Insightd on your remote Tibco Spotfire Windows Server

1. Connect to Spotfire Remote Desktop client



1. Enter the **hostname** of the computer with the installed Remote Desktop client
2. Specify the **User name** of the remote computer
3. Click **[Connect]**
4. Provide your **Password** credential
5. Click **[OK]** to proceed

2. Access Spotfire server



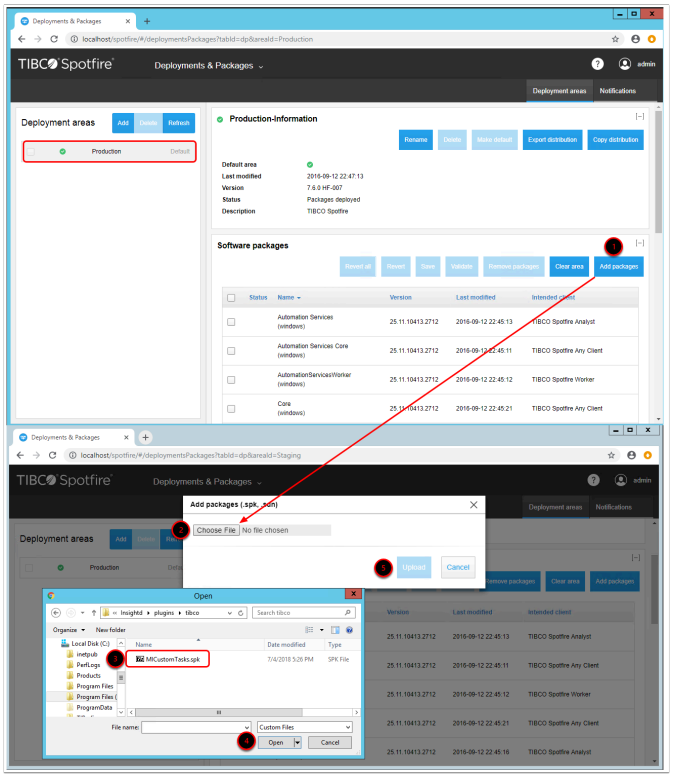
1. Open your browser and type "**localhost**" in the address bar
   - When you press enter, you will be redirected to the Spotfire Homepage
2. Click **Deployments & Packages** icon in the Homepage navigation menu

3. Add the Extension to Deployment area

Spotfire extensions are functional units included in Spotfire add-ins. When added, they enable custom capabilities.

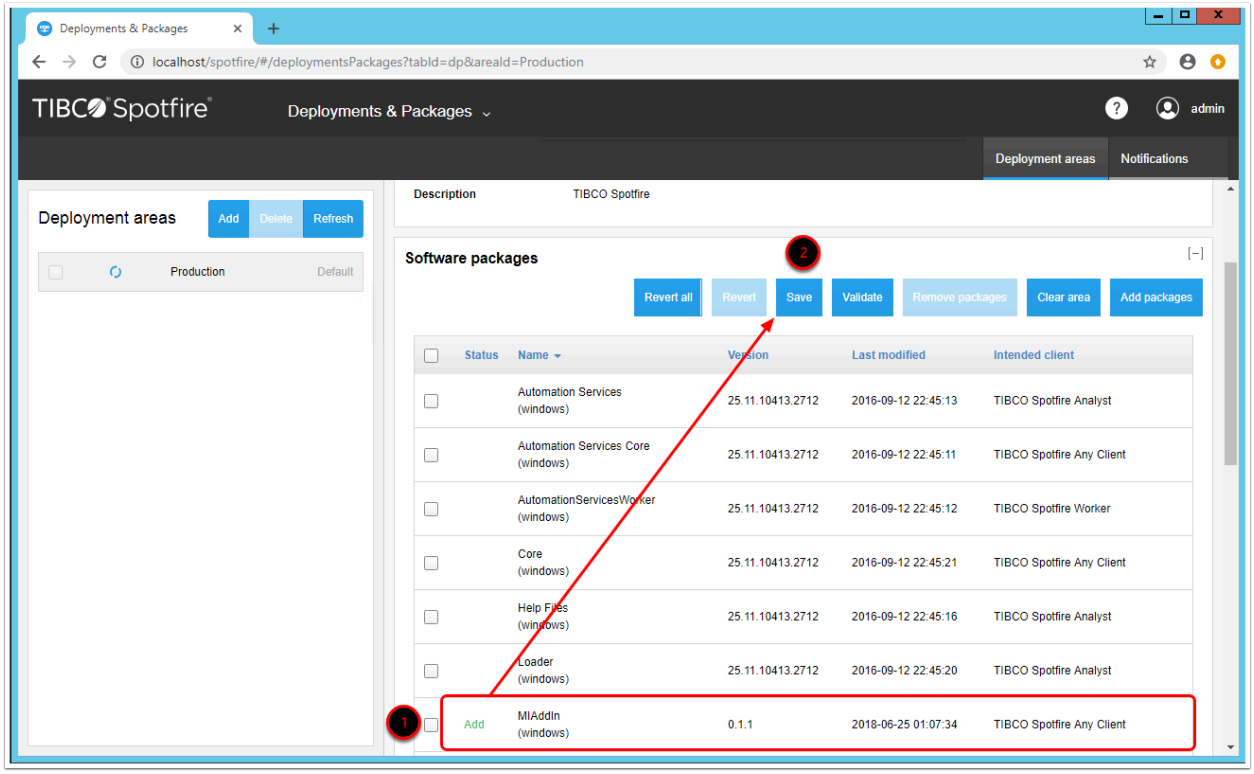
Extension artifacts are compiled and packaged into Spotfire packages, **.spk files** (containing the **.dll**'s for the extension code and other resources as well as a **module.xml file** describing the metadata of the extension).

Since the **.spk** (extension) file can be large, we recommend uploading this file from the Spotfire server directly (**localhost**). Otherwise, you may get the error "Request Entity Too Large" (413).



1. **Deployment areas** page will display available areas for deploying your extension
2. **[Add packages]:** click to add a package to a deployment area
3. **[Choose File]:** click to access the **MICustomTasks.spk** file
   - The file is located in the Tibco plugin directory `C:\Program Files (x86)\Metric Insights\Insightd\plugins\tibco\MICustomTasks.spk`
4. **Open**
5. **Upload**

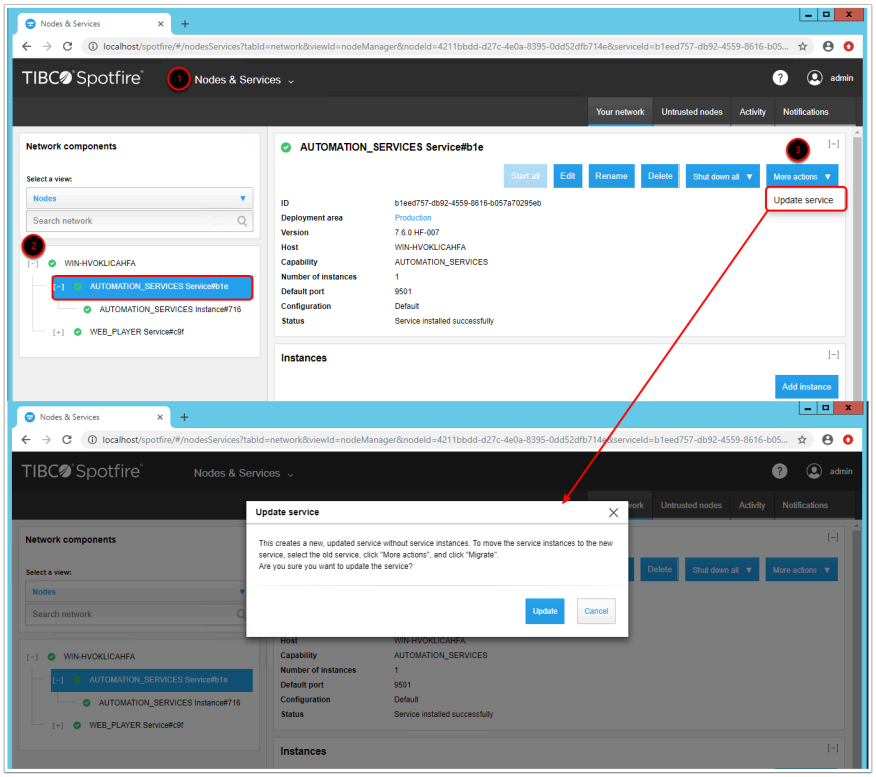
4. Save the new Distribution



1. The *.spk extension file* will be uploaded as an **MIAddIn package** with the **"Add"** status
2. **[Save]:** click to save and validate the new distribution

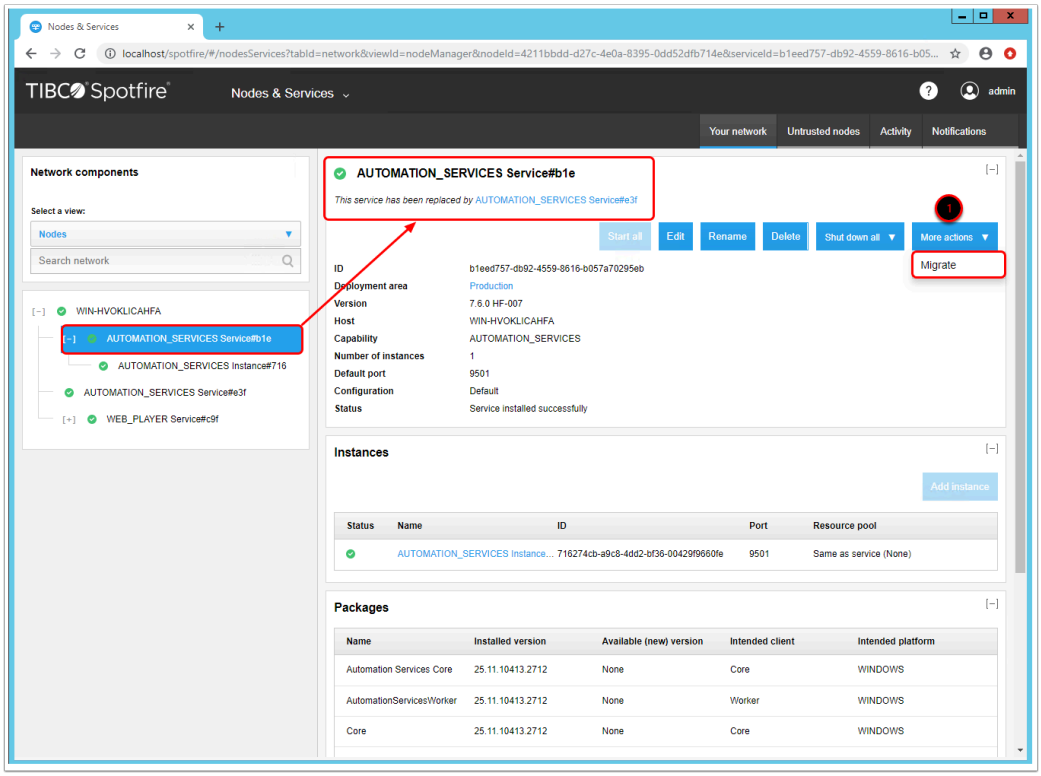
5. Update Automation Services

> ⚠ After adding the MIAddIn extension package, you need to update Automation Services to the new deployment.



1. From the navigation menu dropdown, select **Nodes & Services**
2. Click the **tree view symbol [+]** to open the hierarchical structure of Network components and select the required **Automation Service**
3. Go to **More actions > Update Service**
   - Updating the Service will get the latest modules from the deployment area and create a new Service

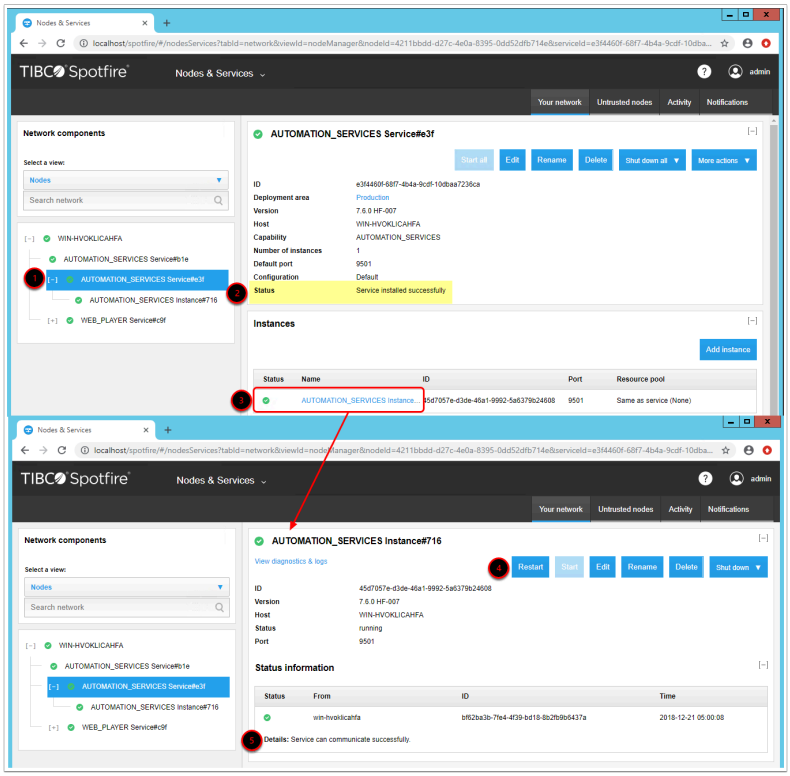
6. Migrate Instances to the new Service



After the new Service has been created, you need to migrate Instances from the old to the new Automation Service.

1. Go to **More actions** > click **Migrate**

7. Restart your Instances

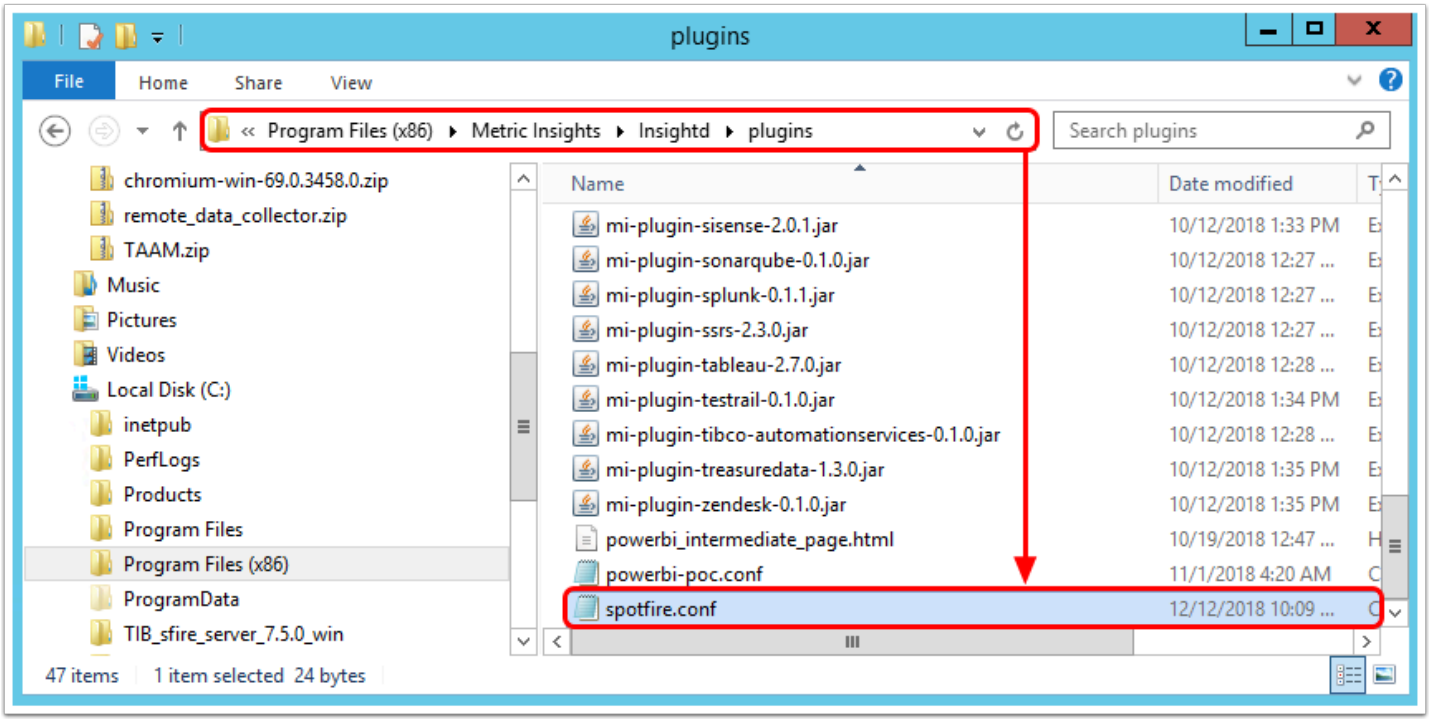


1. **Access** the newly created Automation Service
2. **Verify** that the Service has been installed successfully
3. Click the name of the Instance to **access its configuration page**
4. **Restart** the Instance
5. Make sure that **Service can communicate successfully**

If the new Automation Service runs successfully, you can optionally delete the old Service.

8. Add configuration file to Insighd directory

To connect the MI server to the Tibco Spotfire server, you need to create the **spotfire.conf** file and add it to the **Insightd package**.



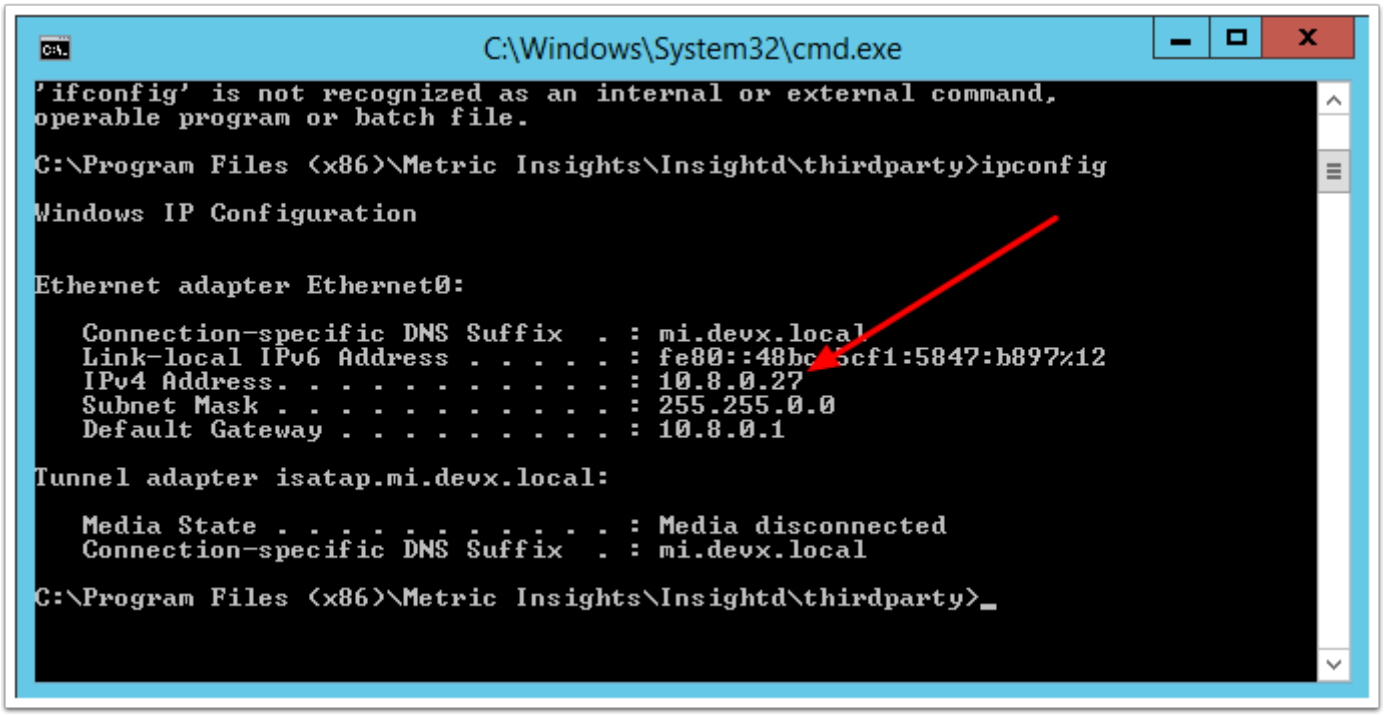
1. In the file, specify the parameter (RDC hostname) as `rdchostname=ip-address or hostname`
   1. For example, `rdchostname=10.8.0.27`
2. Optionally, you can override the port via `rdcPort=port`
   1. For example, `rdcPort=81`
   2. The default port used for TCP socket communication from Spotfire to RDC is 4444
3. Place the configuration file in the **Insightd/plugins** directory

After that, run RDC and Test Connection from the Metic Insights UI (as available in the Tibco Spotfire connection profile).

8.1. How to obtain the ip-address (for spotfire.config)

To get the ip-address of the machine with the installed RDC, enter `ipconfig` in the Windows Command Prompt.

If there are several ip-addresses listed, use the first.


C:\Windows\System32\cmd.exe
'ifconfig' is not recognized as an internal or external command,
operable program or batch file.
C:\Program Files (x86)\Metric Insights\Insightd\thirdparty>ipconfig
Windows IP Configuration
Ethernet adapter Ethernet0:
Connection-specific DNS Suffix . : mi.devx.local
Link-local IPv6 Address . . . . . : fe80::48bc:5cf1:5847:b897%12
IPv4 Address. . . . . : 10.8.0.27
Subnet Mask . . . . . : 255.255.0.0
Default Gateway . . . . . : 10.8.0.1
Tunnel adapter isatap.mi.devx.local:
Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . : mi.devx.local
C:\Program Files (x86)\Metric Insights\Insightd\thirdparty>_

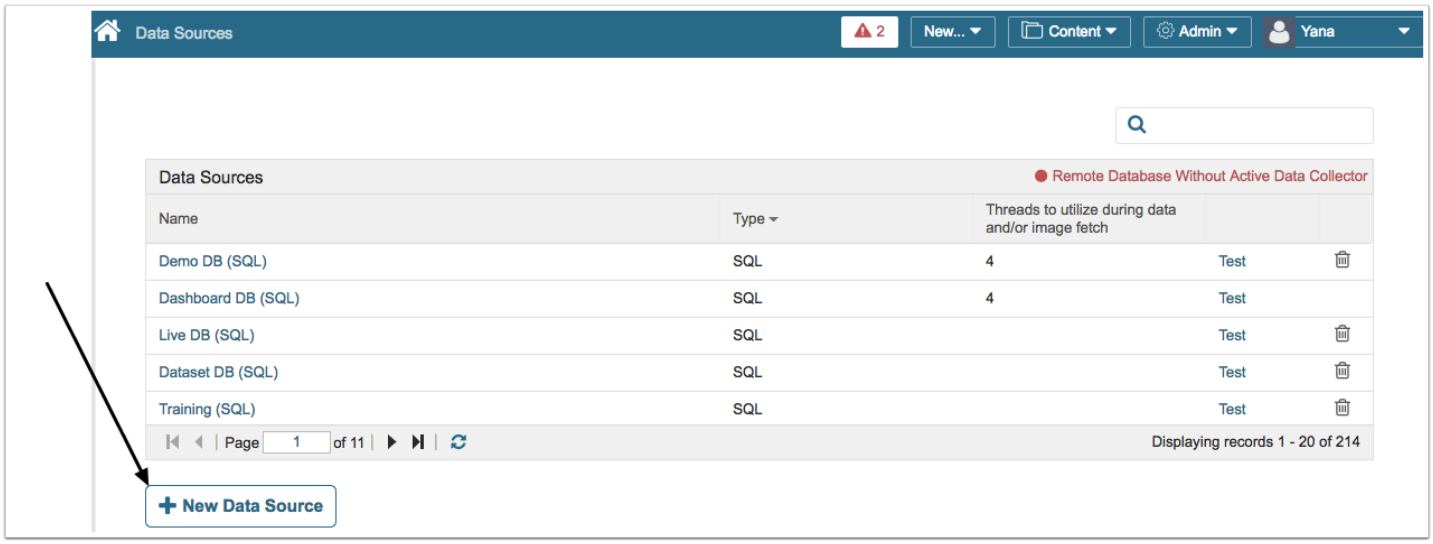
36.2 Establish Connectivity to TIBCO Spotfire Analytics

This article describes how to connect to **Tibco Spotfire** in order to load data into Datasets and Reports in Metric Insights.

PREREQUISITES

For details, refer to Prerequisites for connecting to Tibco Spotfire

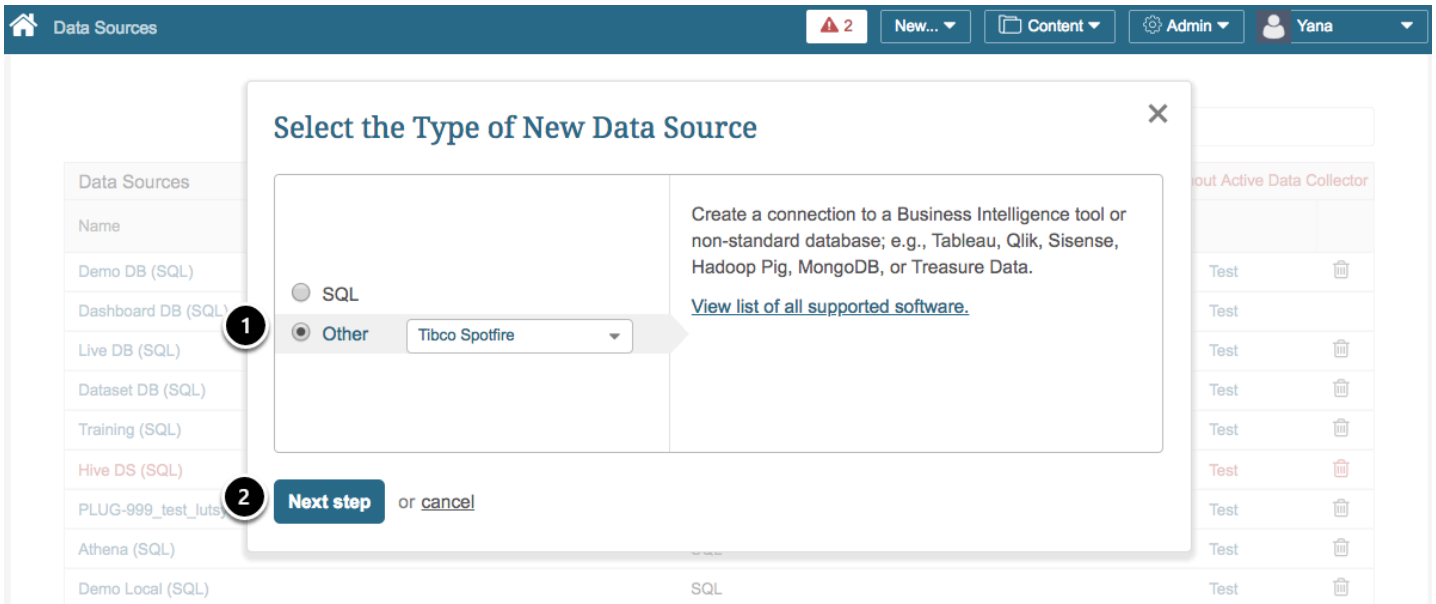
1. Access Admin > Data Sources



At the bottom of the screen click **[+ New Data Source].**

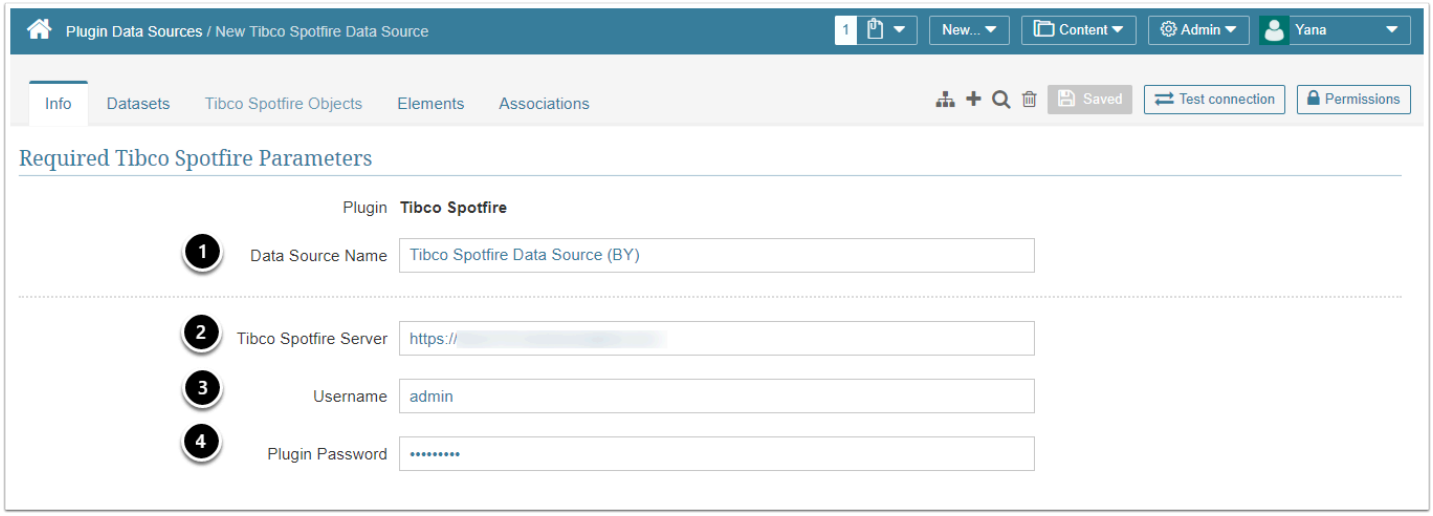
The *Select the Type of New Data Source* pop-up opens.

2. Select the Type of the New Data Source



1. Select "Other" and choose "Tibco Spotfire" from the drop-down list
2. Move to the **Next step**

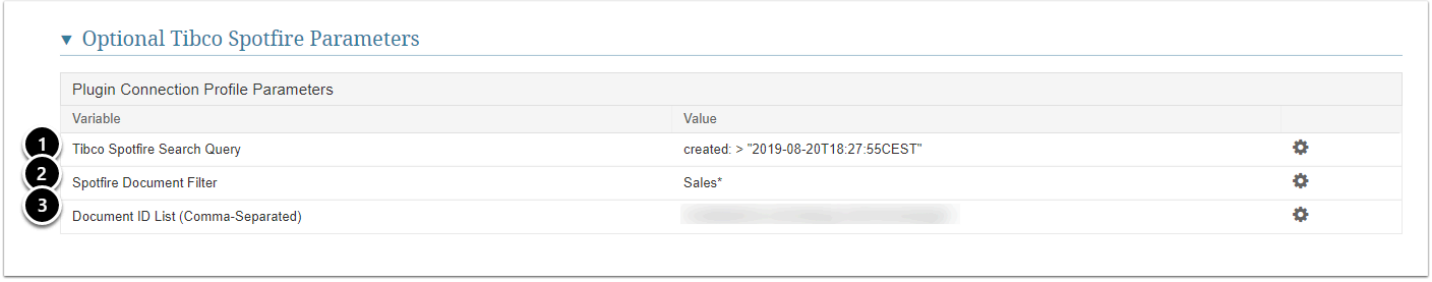
3. Required Tibco Spotfire Parameters



1. **Data Source Name:** set a name for the data source
2. **Tibco Spotfire Server:** define the server protocol (**http** or **https**) and a hostname
3. **Username:** note that your **Username** must be in the same format that your Tibco Spotfire server uses for authentication
4. **Password:** provide your password credential

4. Optional Tibco Spotfire Parameters

Optional Parameters allow Users to limit the number of Objects fetched to Metric Insights.



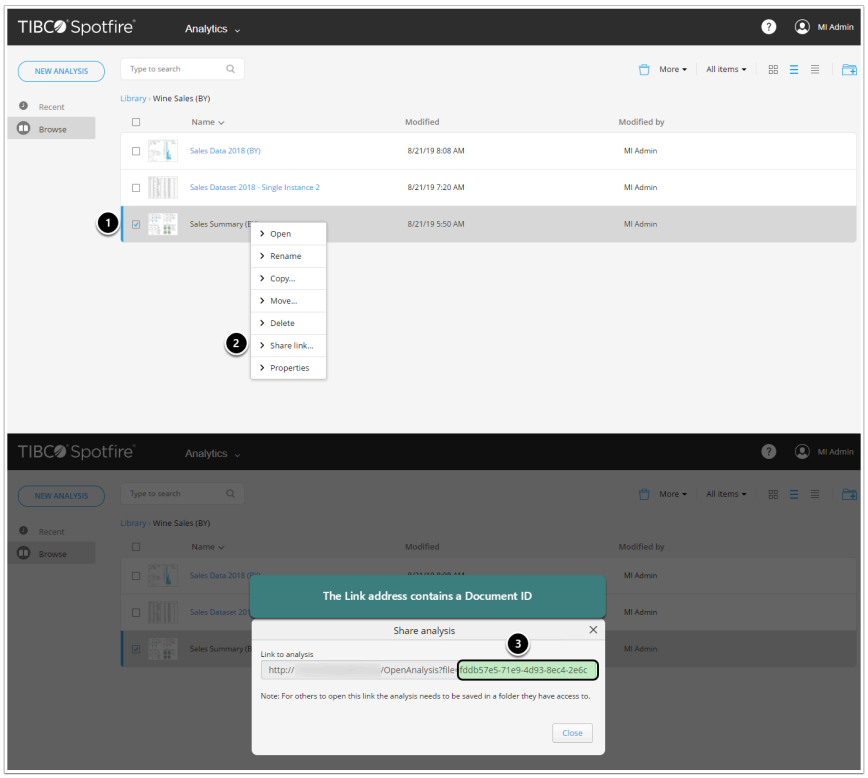
If necessary, configure any of the the following parameters:

1. **Tibco Spotfire Search Query:** use a search command to narrow down the list of Spotfire Objects brought into Metric Insights
2. **Spotfire Document Filter:** limit the number of collected Spotfire Objects by name
   - *wildcard (*) is supported*
3. **Document ID List (comma-separated):** specify IDs to fetch only selected Documents

4.1. How to obtain Document IDs in Tibco Spotfire?

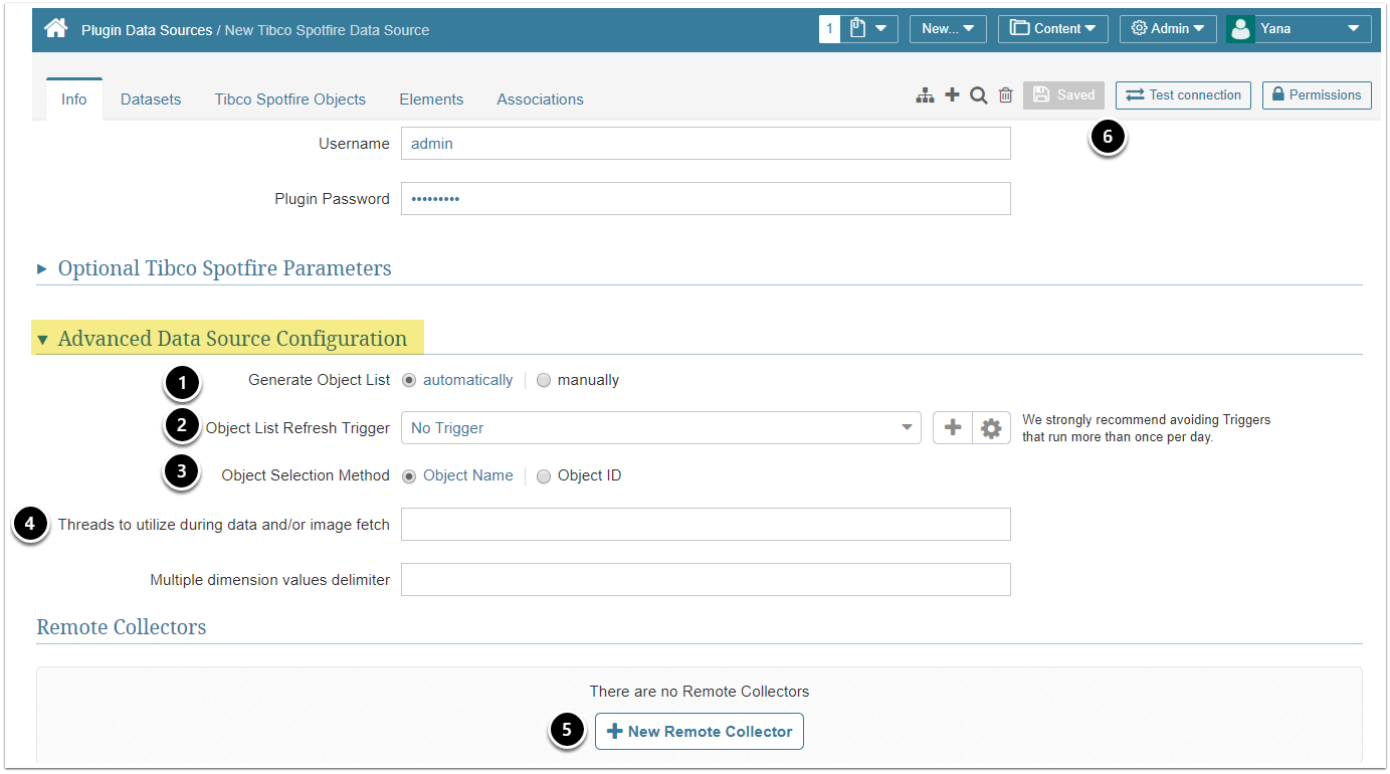
Global Unique Identifiers (GUIDs) are used to filter Tibco Spotfire objects.

- A GUID can be obtained from the UI
- See details below



1. **Right-click** a Document
2. Select **"Share Link"**
3. The value after **file=represents the Document ID**

5. Advanced Configuration

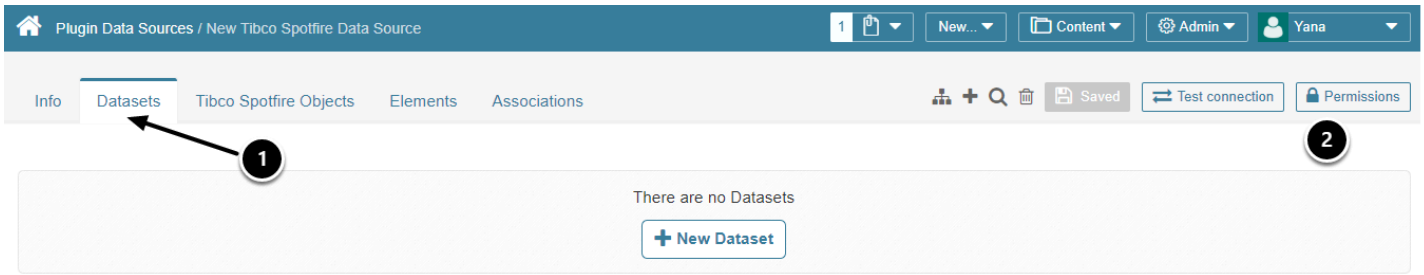


1. **Generate Object List**
   - *automatically*: all Reports are going to be fetched by the system
   - *manually*: Reports may be added one-by-one or via CSV file

2. **Object List Refresh Trigger:** from the dropdown, select the Trigger that will be used to fetch data via the Tibco Spotfire plugin
3. **Object Selection Method:** specify how Tibco Spotfire Reports will be fetched
4. Optionally, state the maximum number of concurrent **Threads to utilize during data and/or image fetch** to be used in background processing when the system updates Reports for this Data Source
   - If you do not specify any value for this setting, batch data collection processing will be single-threaded

5. **New Remote Collector:** make sure you add a Remote Collector, otherwise the plugin connection will fail
6. **Test Connection** (this will also **Save** your data)

6. Other settings



1. You can create Datasets directly from the respective tab
2. Click **Permissions** to assign permissions to Groups or Power Users

What's next?

[How to Collect Data Using the Tibco Spotfire Plugin]

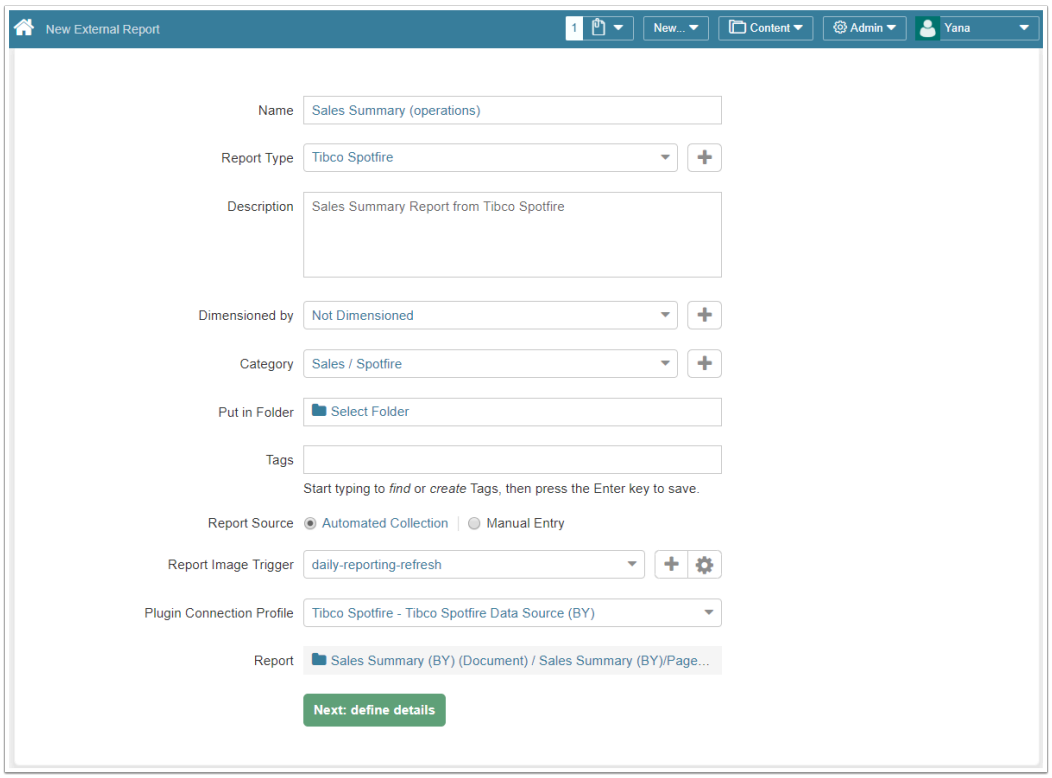
36.3 Create External Report from TIBCO Spotfire Analytics

This article details how to build an External Report in Metric Insights that is linked to a Report on your Tibco Spotfire server.

It assumes that you have already established connectivity to Tibco Spotfire via the respective plugin connection profile.

1. Access New > External Report > Tibco Spotfire

If necessary, you can create a new *Report Type, Category*, or *Report Image Trigger* on the go by clicking the **[+] icon** next to the corresponding field.



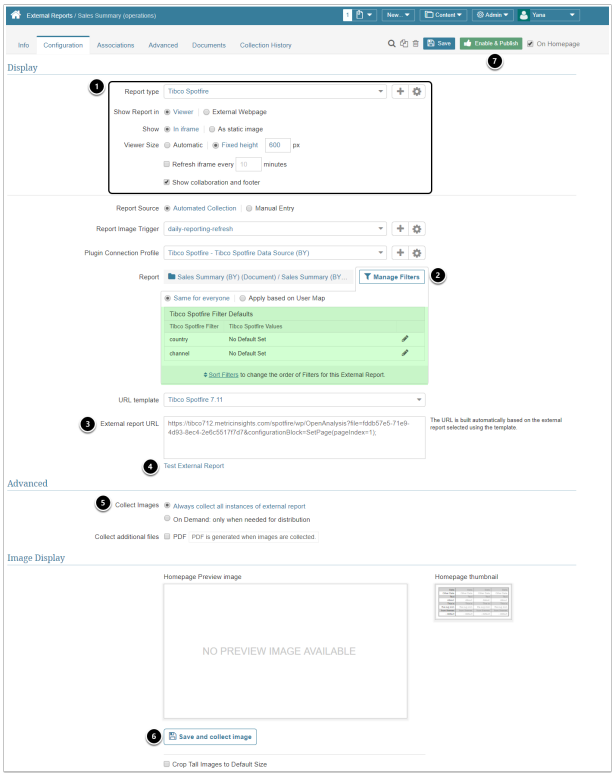
The *New External Report* screen opens. Provide the following information:

1. Give your new External Report a **Name**
2. Place your Report in a relevant **Category**

3. Define whether you want Report content to be updated manually or automatically.
   For **Automated Collection**, define the following settings:
   - Choose the **Report Image Trigger** from the drop-down list
   - Select the **Plugin Connection Profile** you have created for Tibco Spotfire
   - **Report:** Select a Tibco Spotfire Report (Object) from the corresponding connection profile

4. **Next: define details** to proceed with Report creation

2. Configuration tab > specify Report Details



1. **Show Report in:** change from default "External Webpage" if you want your Report to be displayed in Viewer
   - You can choose between the **iframe** and **static image** options
   - If you select **iframe**, you can specify the **Viewer Image Size**

2. Apply filters to your Report data by clicking **[+Manage Filters]**
   - For details, see [Pre-filtering BI tools (External Reports)]

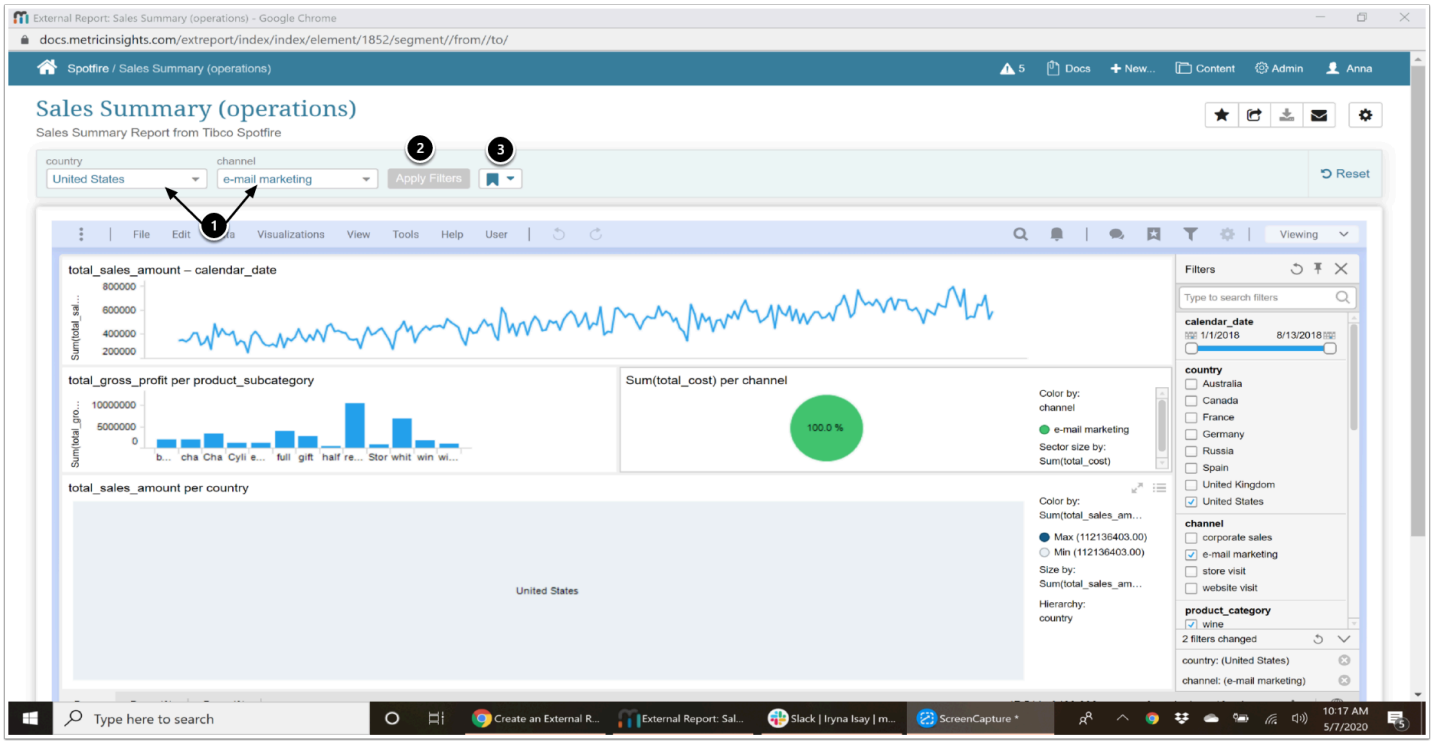
3. The **External Report URL** will be generated automatically based on your other inputs
   - You can modify the URL by appending a question mark (?) followed by any filter or parameter settings

4. **Test External Report**: you can optionally test how your Report will be displayed on External Webpage or in Viewer, depending on the display option selected in #1
5. **Advanced:**
   - *Always collect all instances of external report:* Collect all images and cache them on a schedule

- *On Demand: only when needed for distribution:* Individual images are only collected when they need to be included in an email

6. **Save and Collect Image** to generate a Preview Image for the Homepage
   - NOTE: when opting to display your Report as a **Static Image**, make sure you **Collect Image** before going to Viewer
7. **Enable and Publish** to be able to go to Viewer

3. Verify display in Viewer



To filter Report Data in the Viewer:

1. Select the required **Filter Values**
2. Click **[Apply Filters]**
3. Optionally, save your Filter selections as personalized **Bookmarks**
   - For details, go to [Setting Personal Bookmarks (External Reports)]

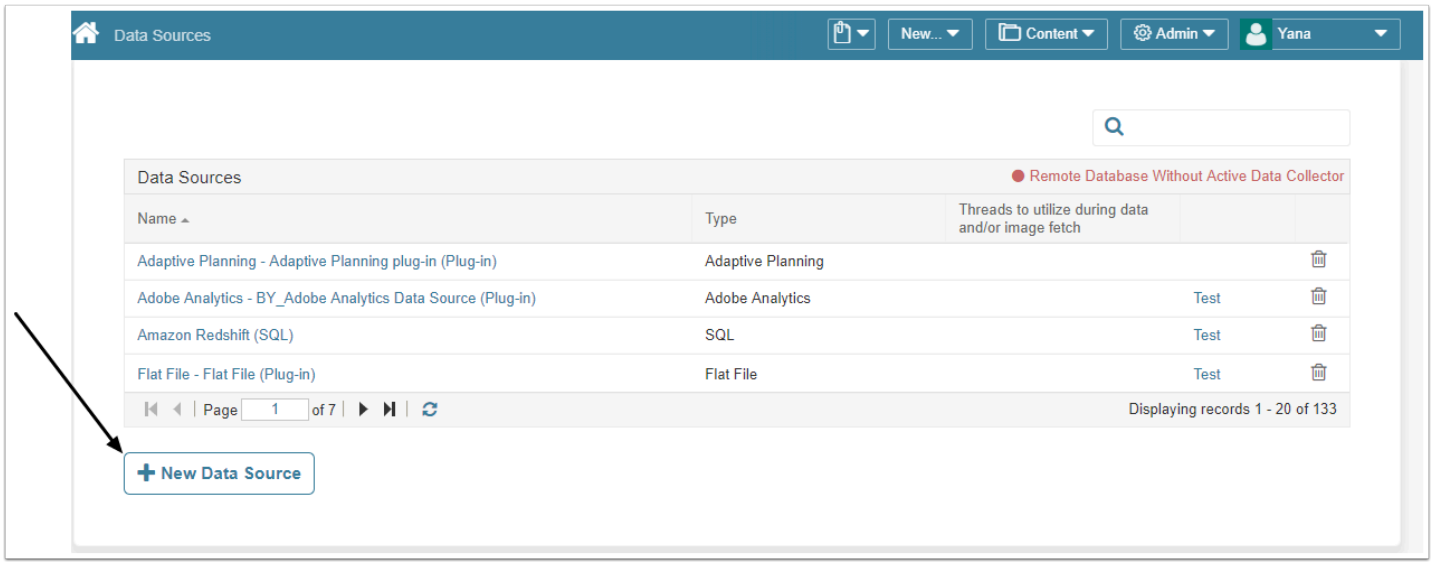
37. Sourcing Data from ThoughtSpot

37.1 Establish Connectivity to ThoughtSpot

This article describes how to connect to **ThoughtSpot** in order to load data into Datasets and Reports in Metric Insights.

[Release 6.x]: 'Remote Data Collector' renamed to 'Remote Data Processor'

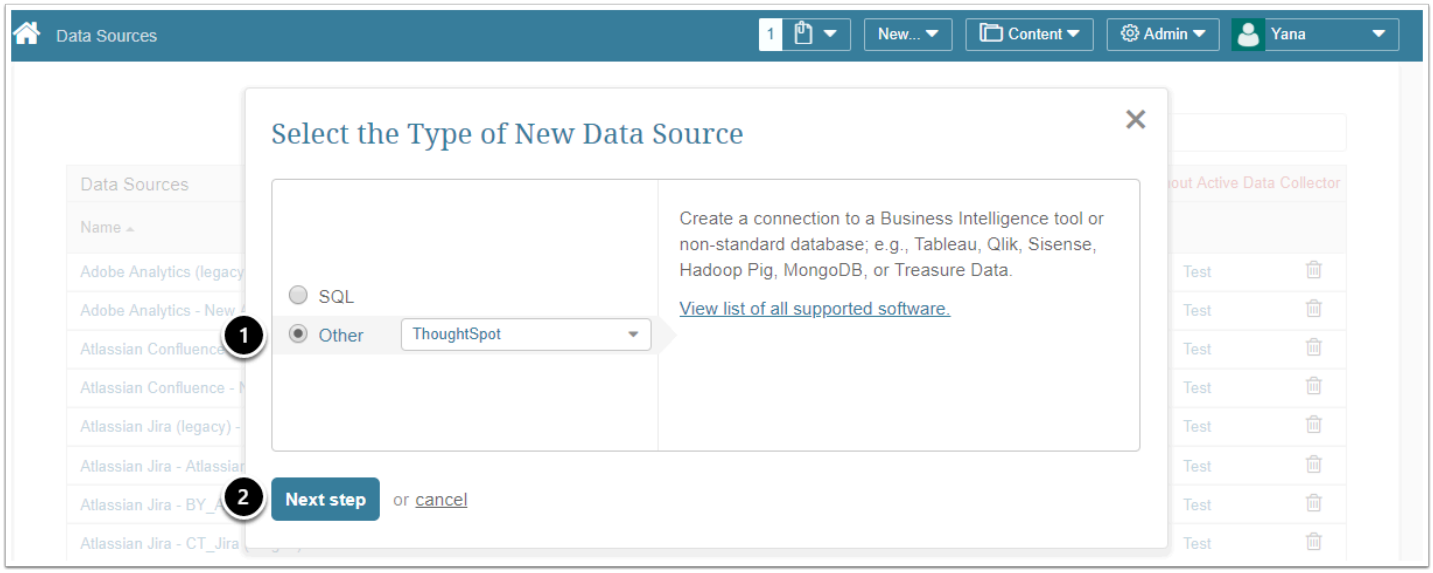
1. Access Admin > Data Sources



At the bottom of the screen click **[+ New Data Source].**

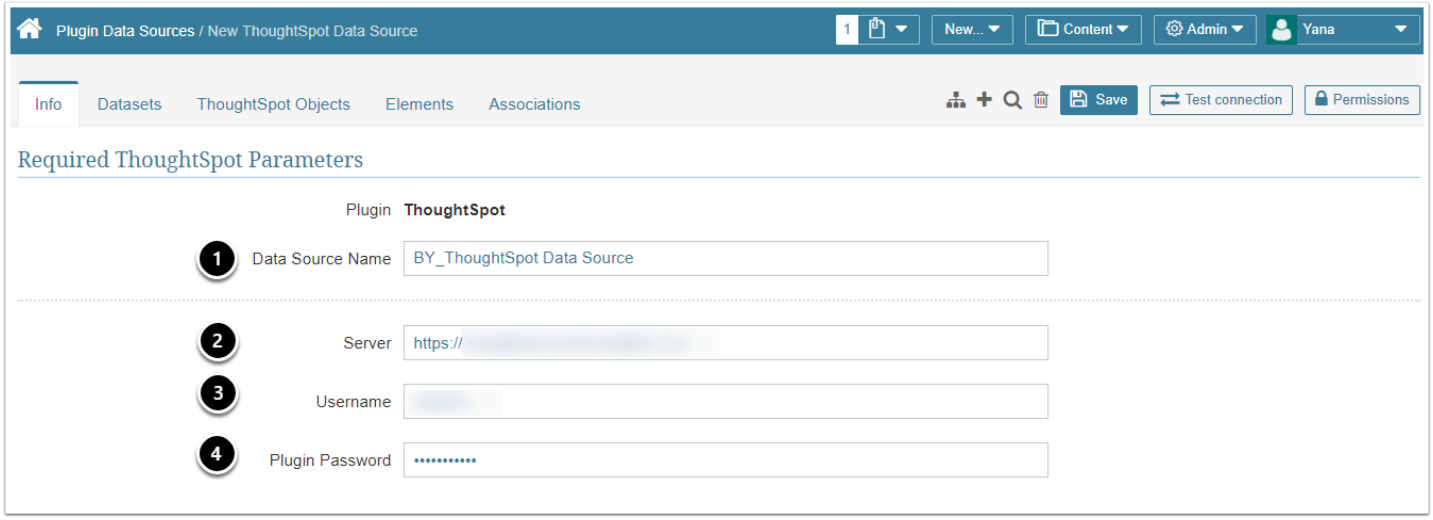
The *Select the Type of New Data Source* pop-up opens.

2. Select the Type of the New Data Source



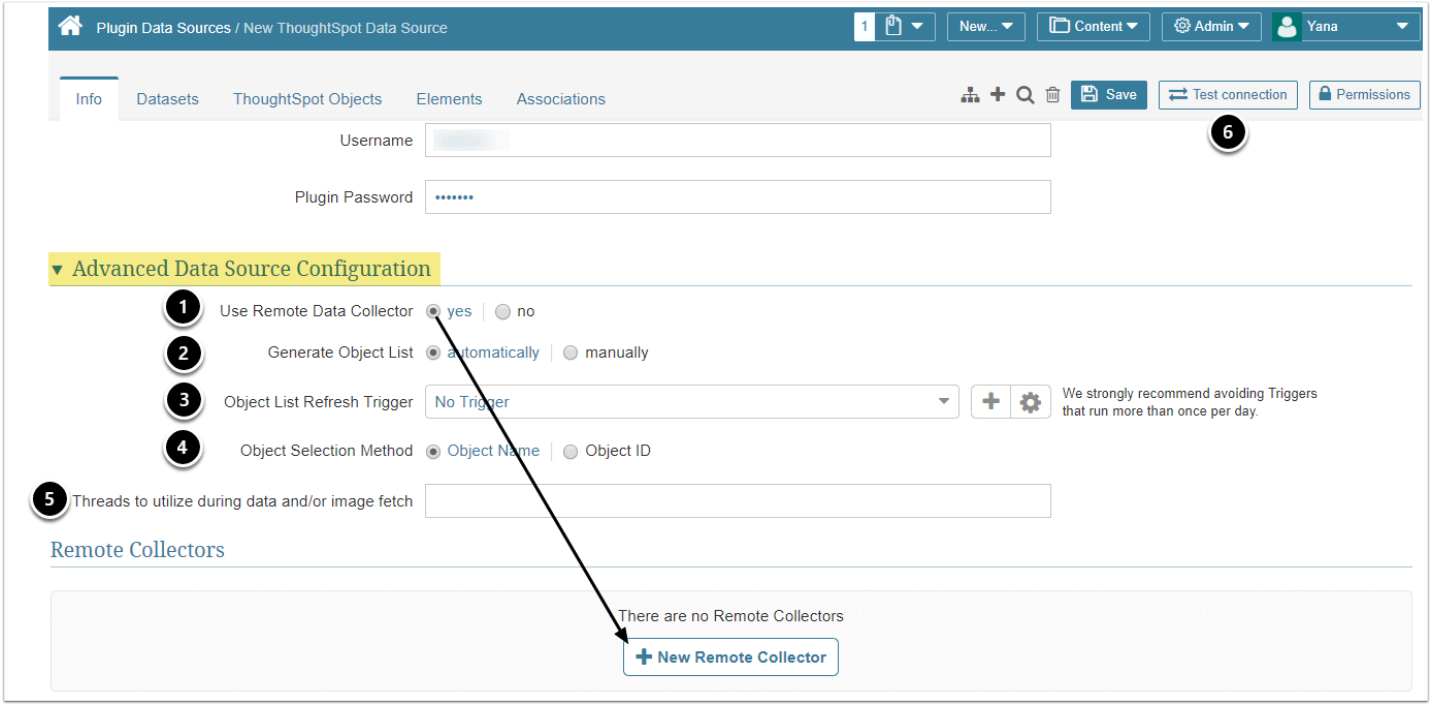
1. Select "Other" and choose "ThoughtSpot" from the drop-down list
2. Move to the **Next step**

3. Required ThoughtSpot Parameters



1. **Data Source Name:** set a name for the data source
2. **Server:** define the server protocol (**http** or **https**) and a hostname
3. **Username:** note that your **Username** must be in the same format that your *ThoughtSpot* server uses for authentication
4. **Password:** provide your password credential

4. Advanced Configuration



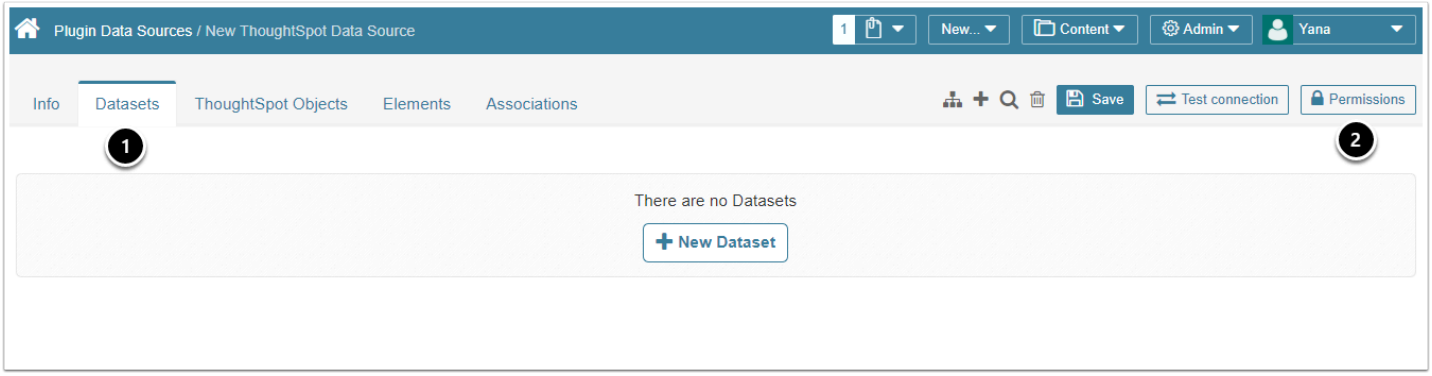
1. **Use Remote Data Collector:** is set to "no" by default
   - If required, switch to "yes" and add a Remote Data Collector by clicking **[+New Remote Collector]**

2. **Generate Object List**
   - *automatically*: all Reports are going to be fetched by the system
   - *manually*: Reports may be added one-by-one or via CSV file

3. **Object List Refresh Trigger:** from the dropdown, select the Trigger that will be used to fetch data via the ThoughtSpot plugin
4. **Object Selection Method:** specify how ThoughtSpot Reports will be fetched
5. Optionally, state the maximum number of concurrent **Threads to utilize during data and/or image fetch** to be used in background processing when the system updates Reports for this Data Source
   - *If you do not specify any value for this setting, batch data collection processing will be single-threaded*

6. **Test Connection** (this will also **Save** your data)

5. Other settings



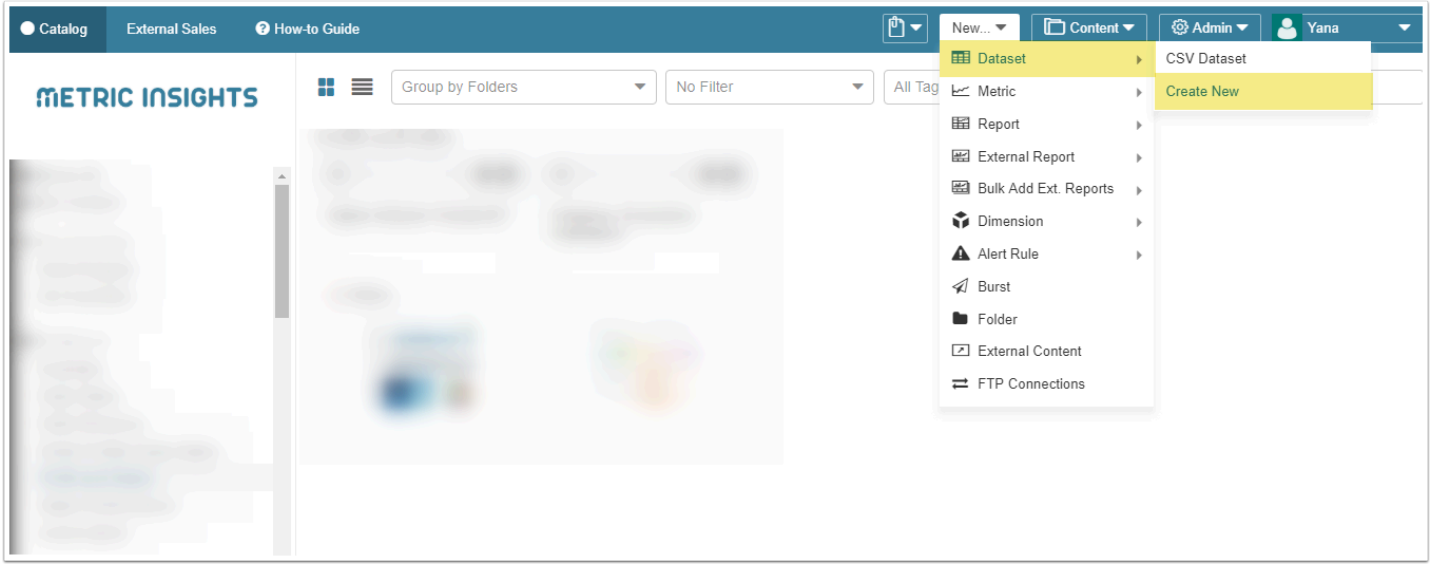
1. You can create **Datasets** directly from the respective tab
2. Click **Permissions** to assign permissions to Groups or Power Users

37.2 Create a Dataset from ThoughtSpot

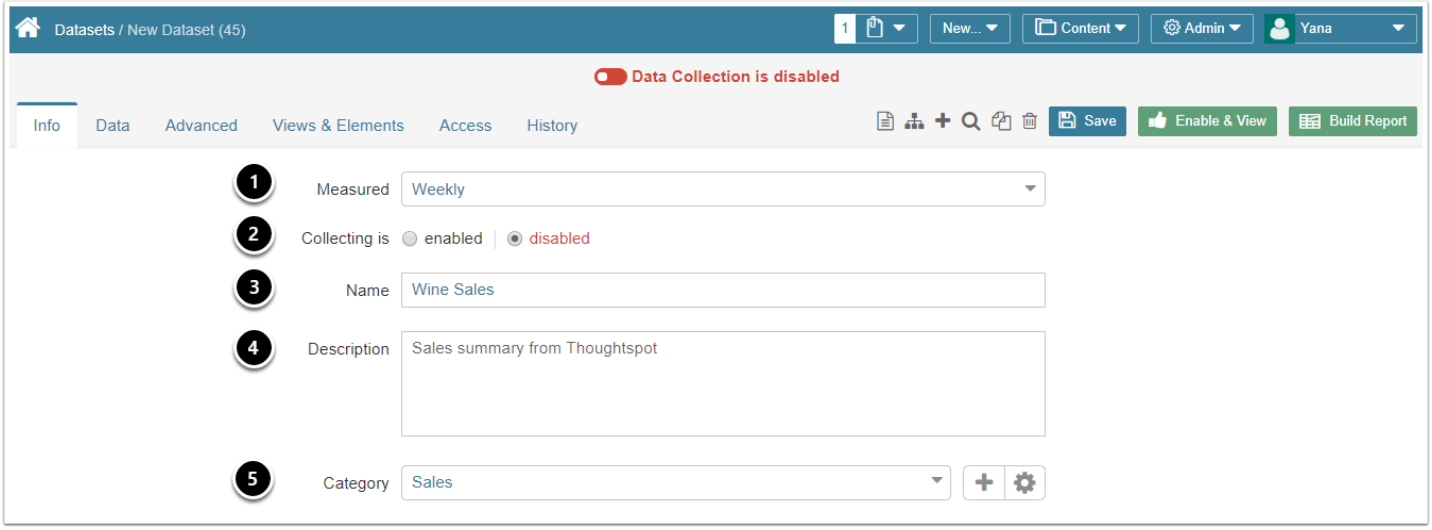
This article details how to create a Dataset populated with data sourced from ThoughtSpot.

It assumes that you have already established connectivity with your ThoughtSpot server via the respective plugin connection profile.

1. Access New > Dataset > Create New



2. Dataset Editor > Info tab

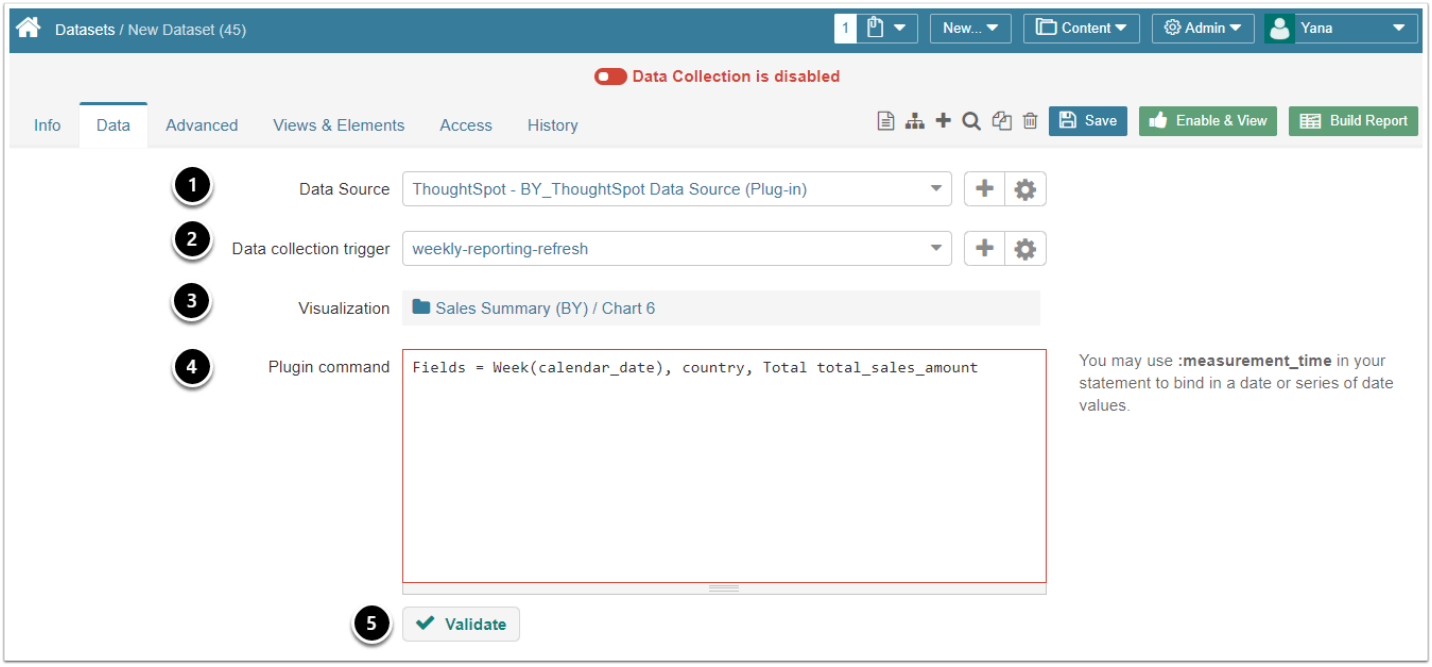


Define the basics:

1. **Measured:** select the measurement interval that applies to the level of aggregation that you want in your result set
2. **Collecting**: new Datasets are always disabled by default to make sure that you can take time to configure them properly before enabling. This setting is duplicated at the top of the screen
3. **Name:** provide a unique name for your Dataset. Preferably, the Dataset name should explain what kind of data it contains
4. **Description:** optionally, provide any additional information about your Dataset
5. **Category:** specify the Category where you Dataset will be placed

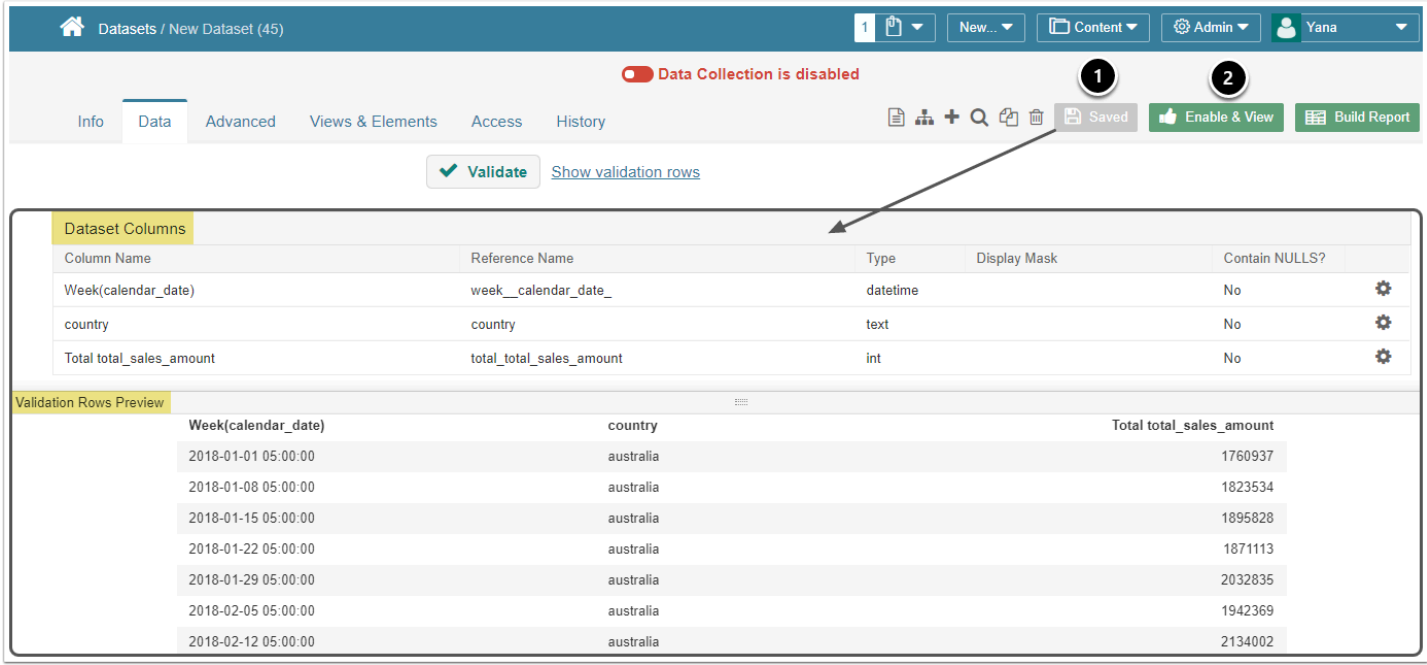
Move to the ***Data tab*** to define the source of data and how often it should be updated.

3. Define the Settings for Data Collection



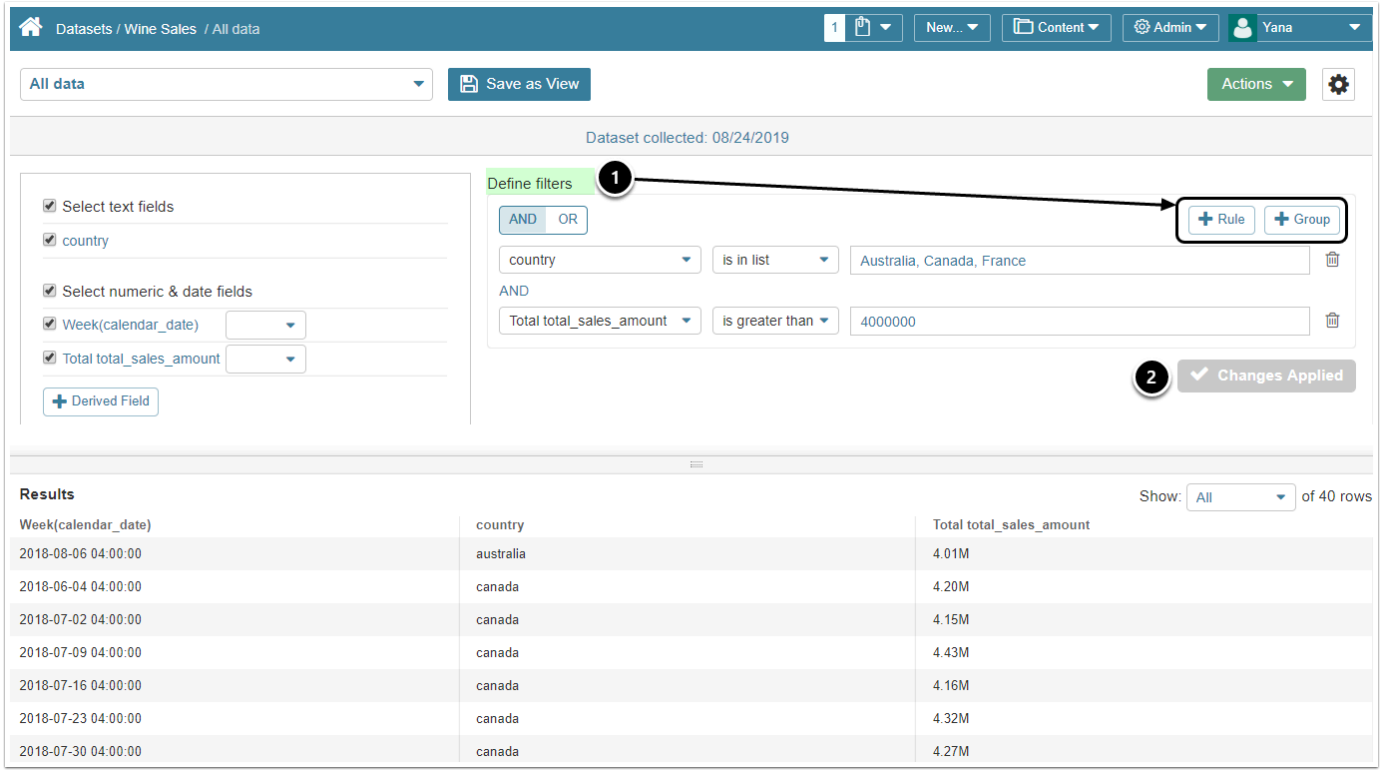
1. **Data Source**: select the connection profile you have created for *ThoughtSpot*
2. **Data collection trigger**: specify the Trigger that will be used to collect data for your Dataset
3. **Visualization:** select a *ThoughtSpot Object* that should serve as a basis of your Dataset
4. Input an MIQL **Plugin Command** listing all data that needs to be fetched from *ThoughtSpot*
5. **Validate** your query

4. Plugin command will be validated and data collected on Save



1. If the command is validated successfully, the **Dataset columns** and **Data Preview** are going to be shown below.
2. At the upper right corner of the screen, click **Enable & View**.

5. Dataset will be displayed in Viewer



In the **Dataset Viewer**:

1. You can refine your data with Dataset's *internal filtering options* by **applying Rules** and **Grouping**:
   - For more information on filtering Dataset data, refer to Create a Dataset View
   - For general instructions on building Datasets, see Create a Dataset from any Data Source
2. Use the corresponding button to **apply changes**

38. Sourcing data from Treasure Data

38.1 Establish connectivity to Treasure Data

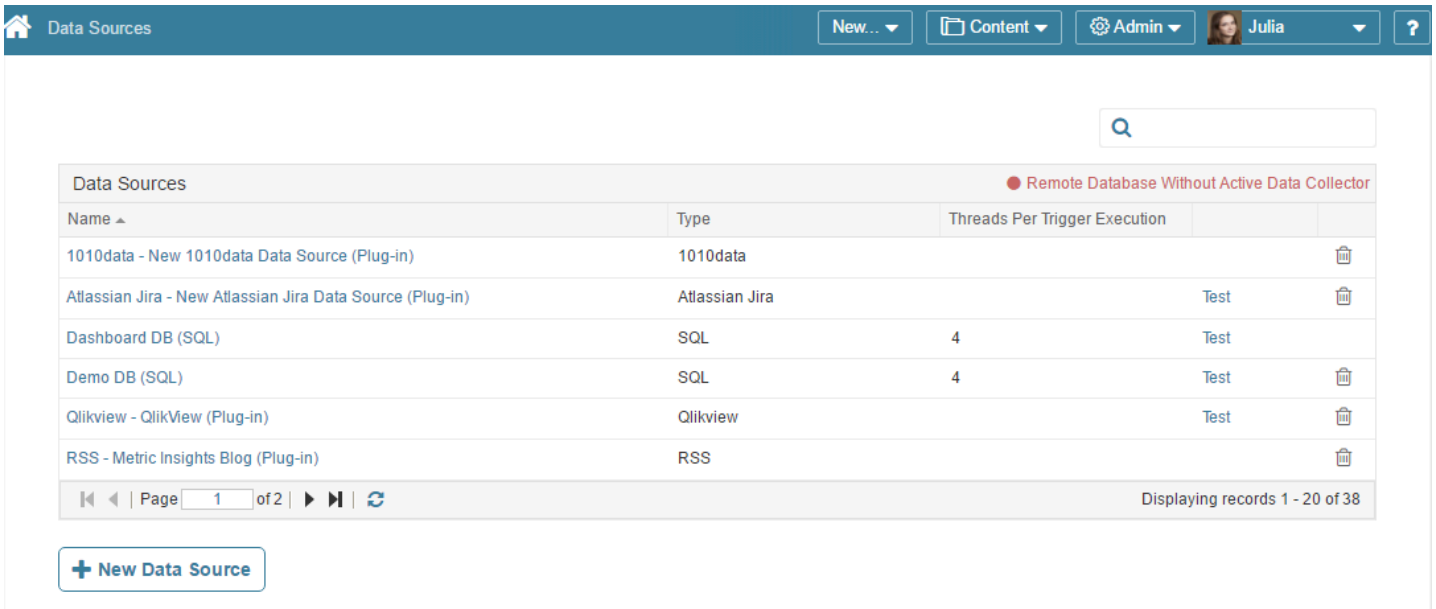
This article describes the process of connecting to a Treasure Data Server in order to use their reports as data sources in Metric Insights.

PREREQUISITES

Before you begin, be sure that you have:

- **API key:** please refer to Treasure On-line Data documentation for information on obtaining your API key.
- **Database name** for connecting to Treasure Data

1. Access Admin > Data Sources

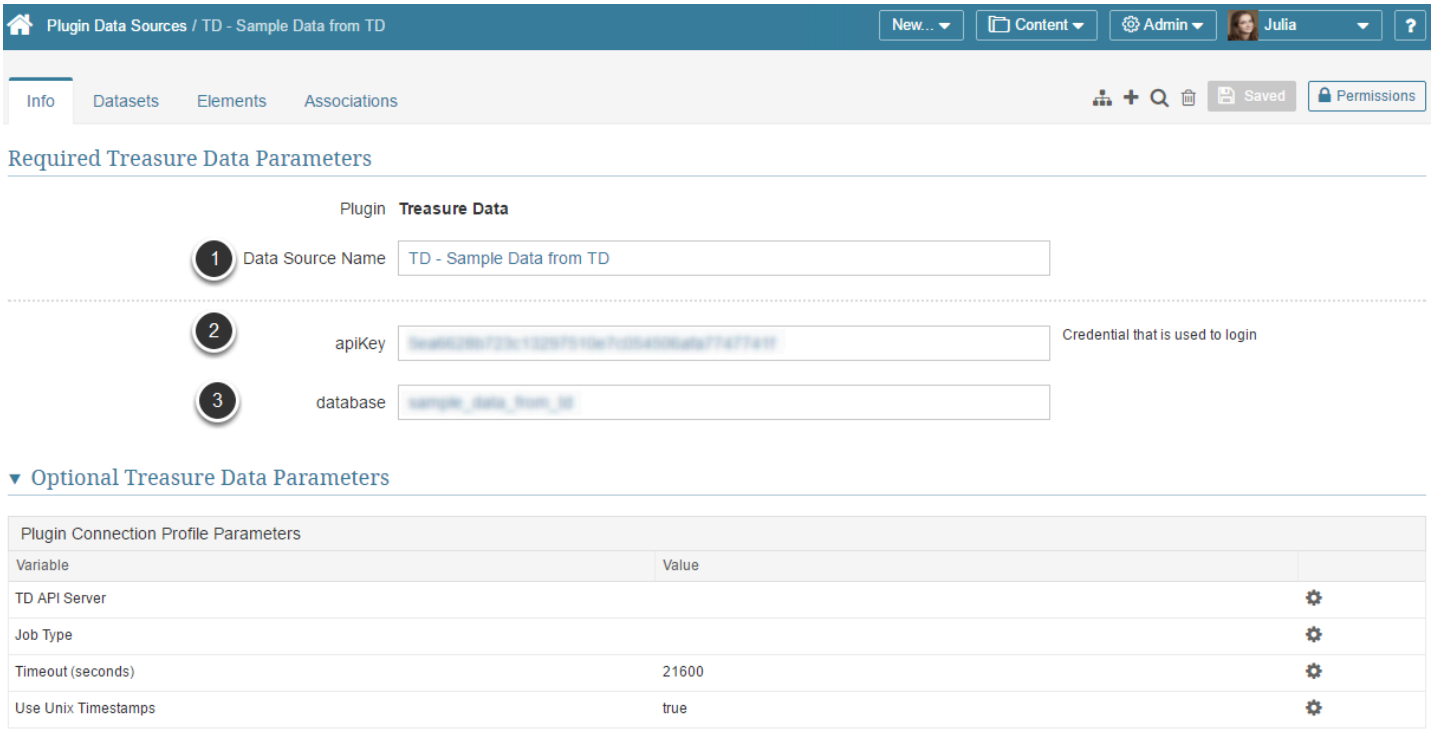


1. At the bottom of the screen click **[+ New Data Source].**
2. The *Select the Type of New Data Source* pop-up opens.

2. Select "Other" Data Source Type and choose "Treasure Data" from the drop-down list

Move to the **Next step**.

3. Provide the required Treasure Data Parameters



1. **Data Source Name:** is defaulted but may be modified
2. Enter **apiKey**

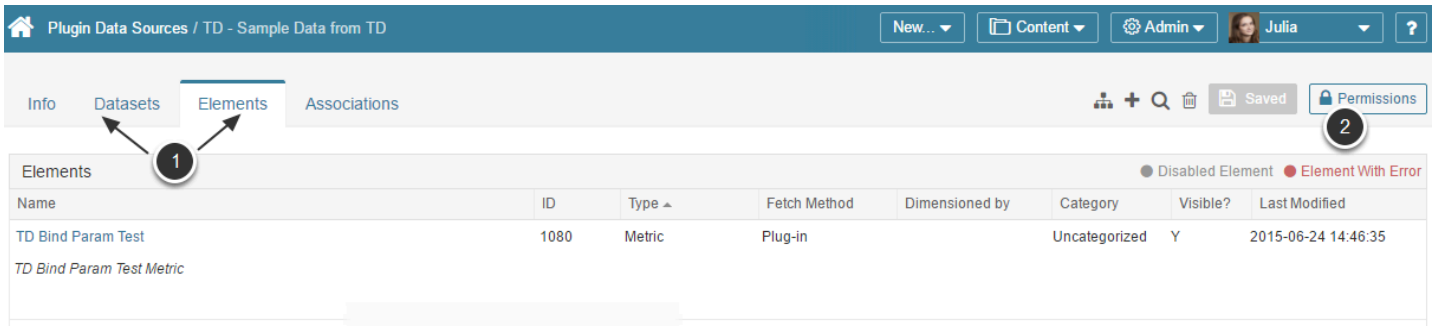
3. Enter the **Database** name

Save your entries.

Optional Parameters

Parameter	Description
API Server	Define the Treasure Data Server.
Job Type	Possible values: HIVE ("hive"), MAPRED ("mapred"), IMPALA ("impala"), PRESTO ("presto"), UNKNOWN ("none")
Timeout (seconds)	Specify the allowed time period during the response from the server shall be received.
Use Unix Timestamps	TRUE - the Unix format timestamp is to be used; FALSE - the Metric Insigtht's format timestamp is to be used.

4. Other Settings



1. You can create elements or Datasets directly from the respective tabs
2. Click **Permissions** to assign them to Groups or Power Users

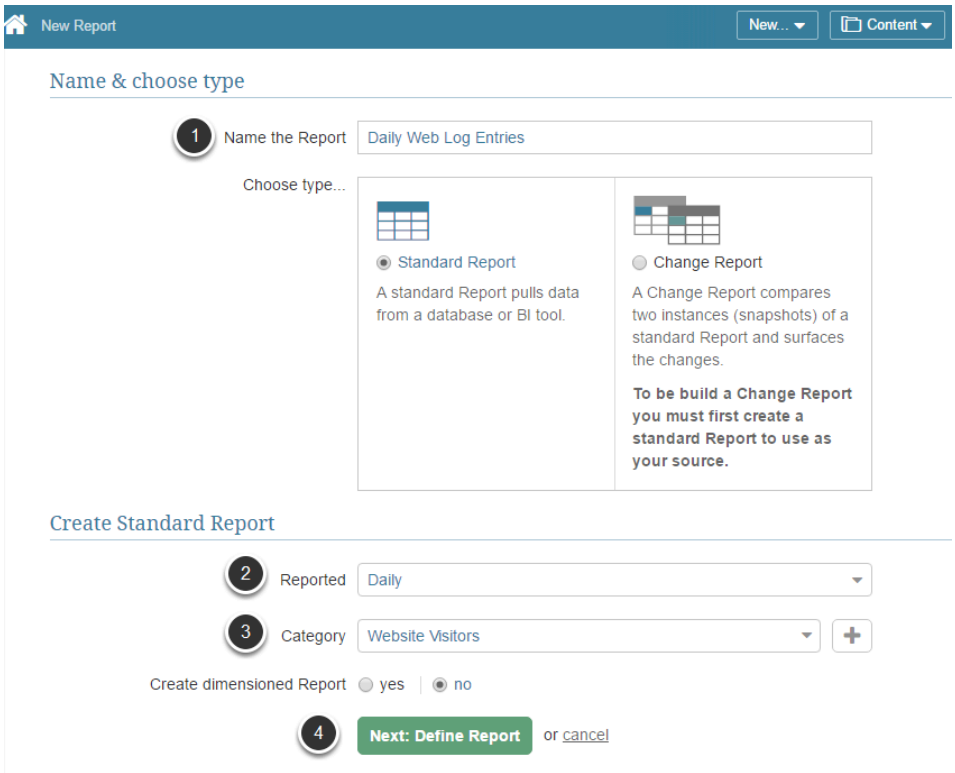
What's next?

[How to collect data from Treasure Data]

38.2 How to collect data from Treasure Data

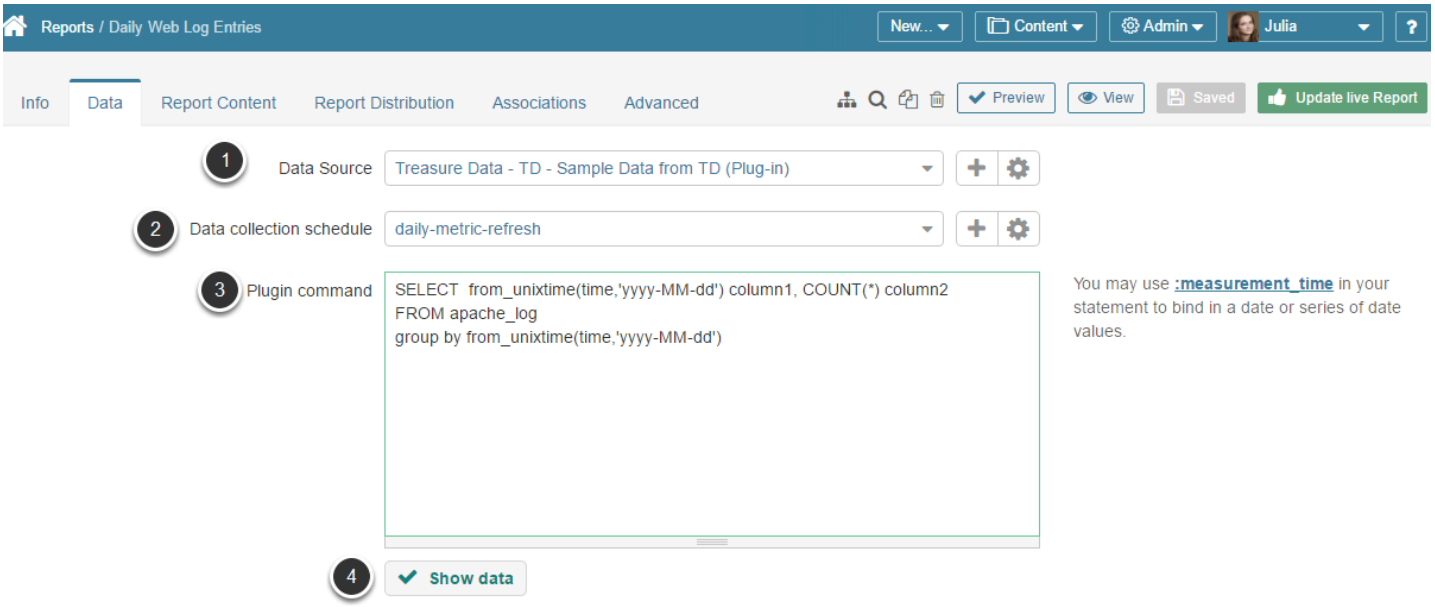
This article will show you how to create a Metric or Report using a **Treasure Data** report as a data source. It assumes that you have already established connectivity to your Treasure Data server.

1. Access New > Report



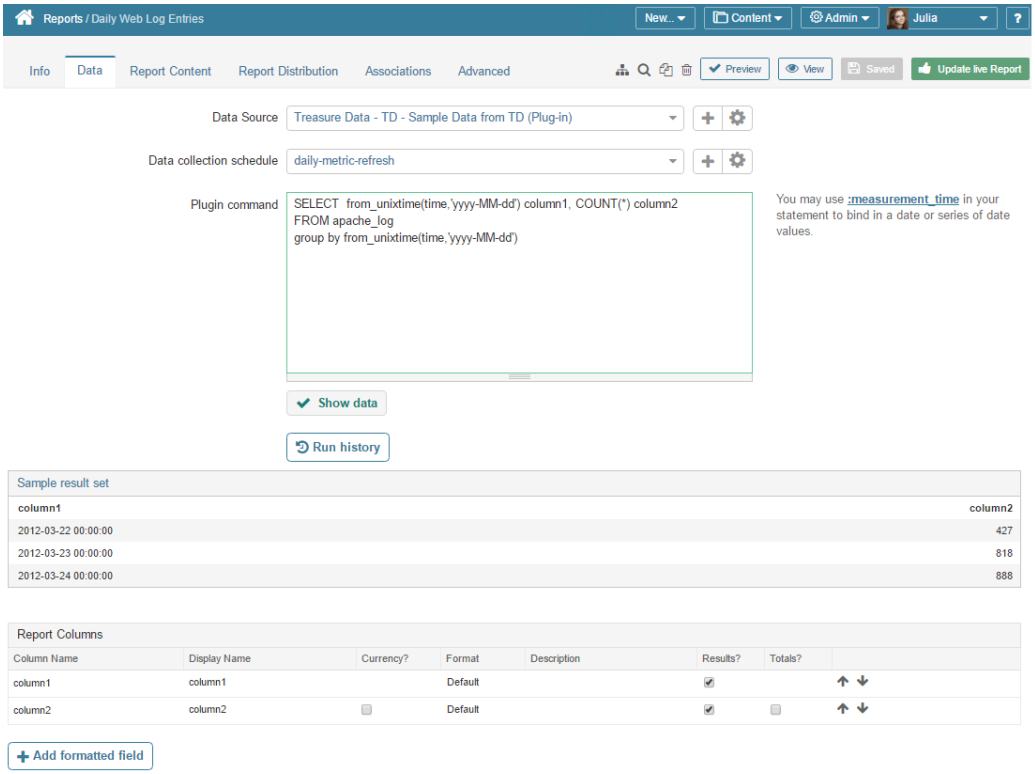
1. **Name the Report**: Define a unique descriptive name of your element
2. **Reported**: choose the measurement interval from the drop-down list
3. **Category**: define a category this element belongs to
4. To move on to defining data collection details, click **Next: Define Report**

2. Define settings for Data Collection



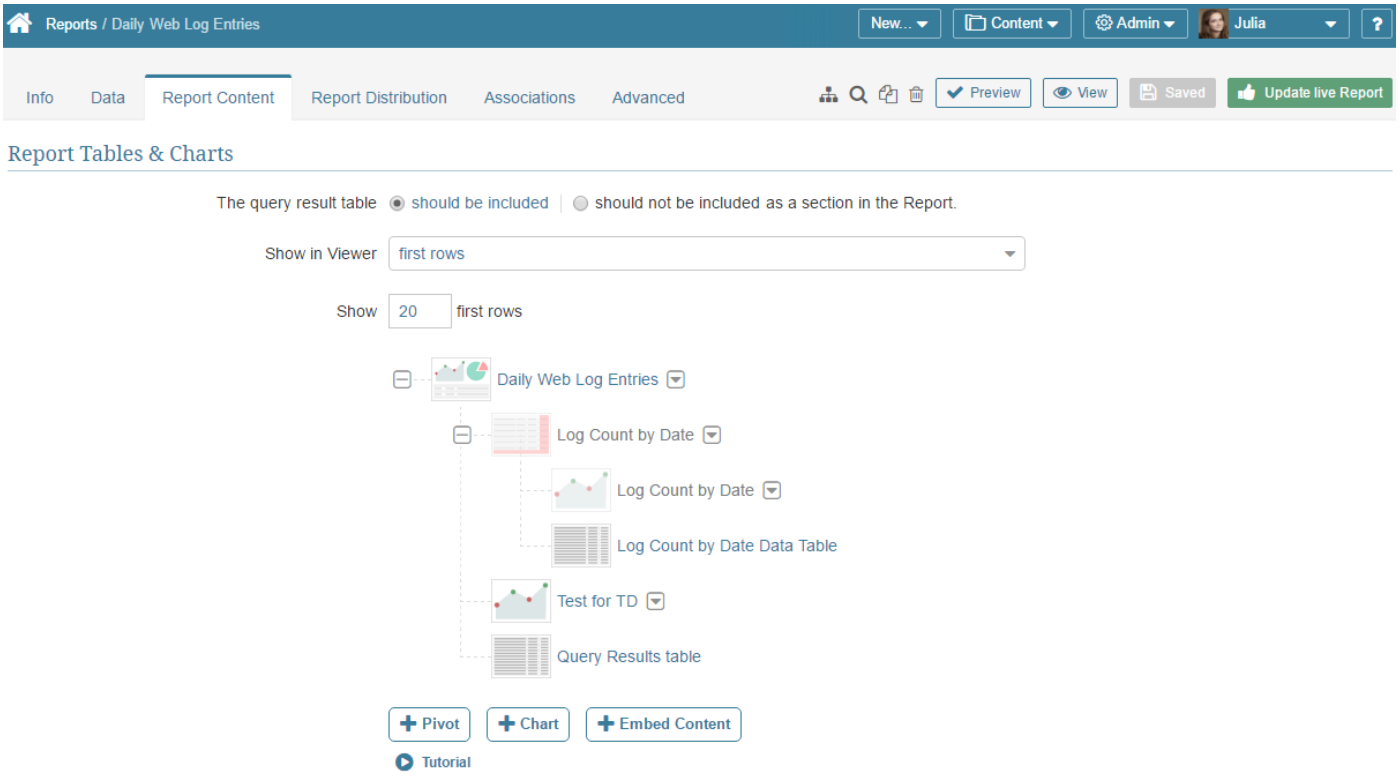
1. **Data Source:** select the connection profile you have created for **Treasure Data**
2. **Data Collection Schedule:** Specify the trigger that will be used to collect the data for your report
3. Input the HiveQL statement in the **Plugin command** text box
4. Once you are ready with you command, click **Show Data**.

3. If the command is validated successfully



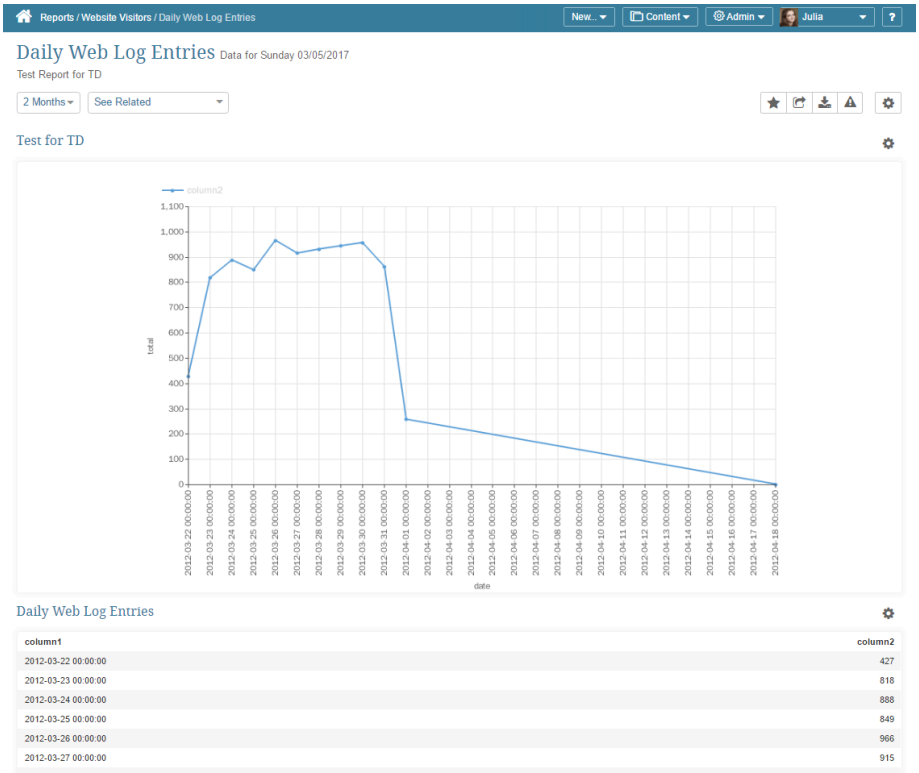
1. If your plugin command is valid, the command box is **green** and the **Report Columns** are shown in the table below; if there are any errors, the box is colored in **red** and errors are explained below the statement box.
2. Click **Update Live Report** to save the changes and move to the *Report Viewer*.

4. Creating a Chart from the Report's result set



You may add a Chart based on the Report result set. For more details, refer to: [Develop Report Chart based on the Result Set (without a Pivot)]

Result



column1	column2
2012-03-22 00:00:00	427
2012-03-23 00:00:00	818
2012-03-24 00:00:00	888
2012-03-25 00:00:00	849
2012-03-26 00:00:00	966
2012-03-27 00:00:00	915

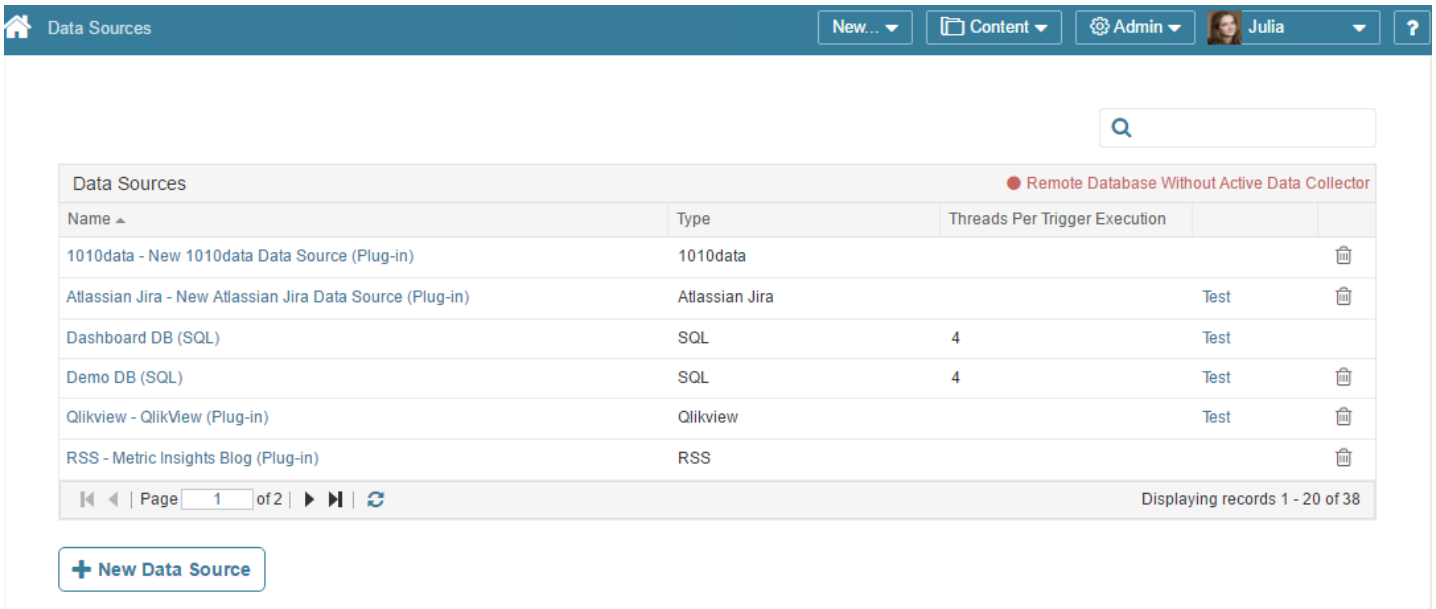
39. Sourcing Data from Uptimerobot

39.1 Establish Connectivity to Uptimerobot

This article describes how to connect to **Uptime Robot** in order to use their reports as Data Sources in Metric Insights.

General instructions on setting up data sources based on plug-ins can be found here.

1. Add New Data Source



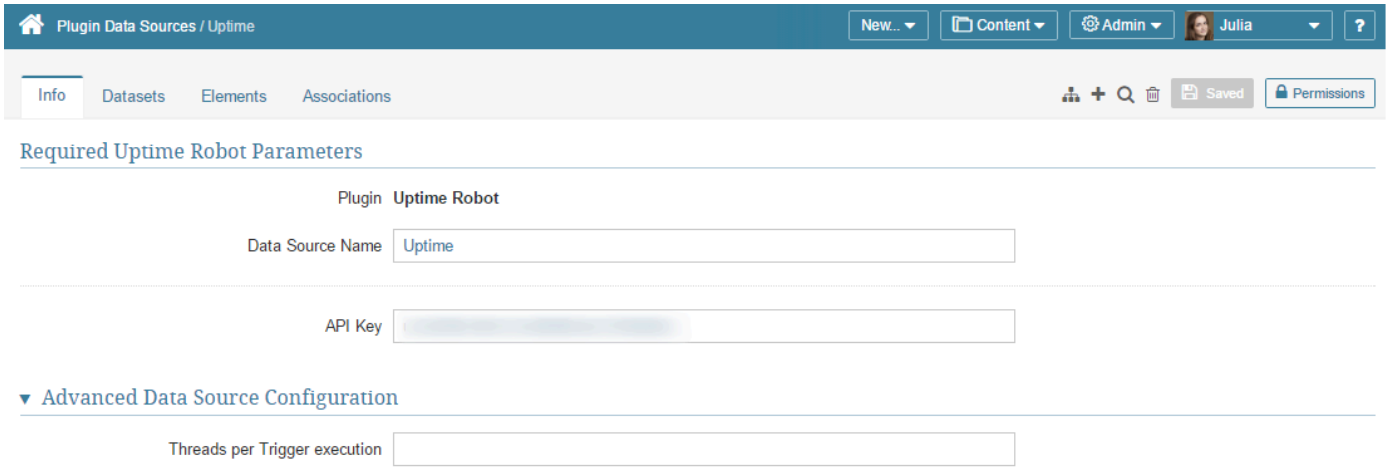
At the bottom of the screen click **[+ New Data Source].**

The *Select the Type of New Data Source* pop-up opens.

2. Select "Other" Data Source Type and choose "Uptime Robot" from the drop-down list

Move to the **Next step**.

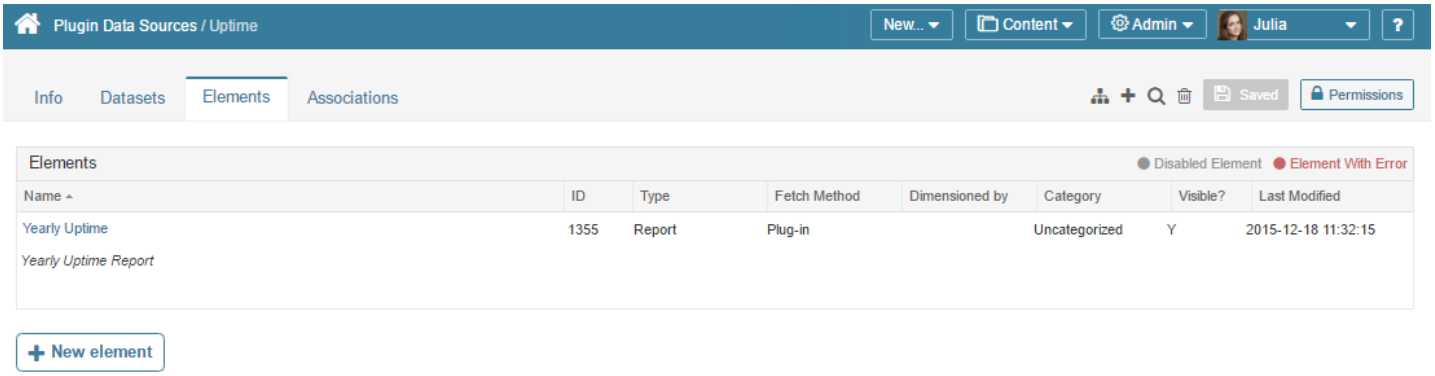
3. Authenticate and Test



1. Enter **API** token. For more details refer to Uptime Robot help doc.
2. **External Reports fetch method**: This setting influences options available in the *Microstrategy Reports List* tab:
   - **automatically**: just click **Refresh list** and all Reports are going to be fetched by the system
   - **manually**: Reports may be added one-by-one or via CSV file

Save your entries.

4. Other Settings



1. You can create elements or Datasets directly from the respective tabs
2. Click **Permissions** to assign them to Groups or Power Users

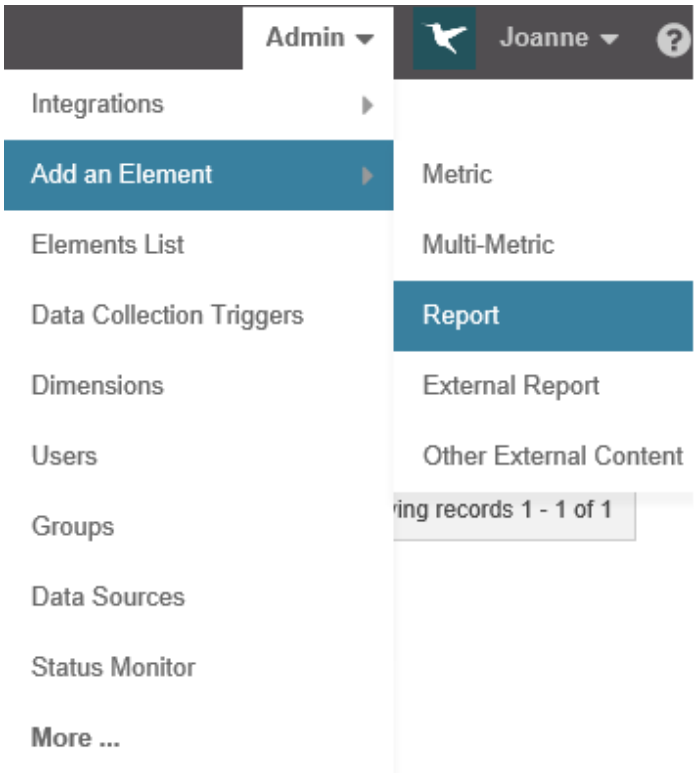
What's next?

[How to Collect Data using Uptime Robot]

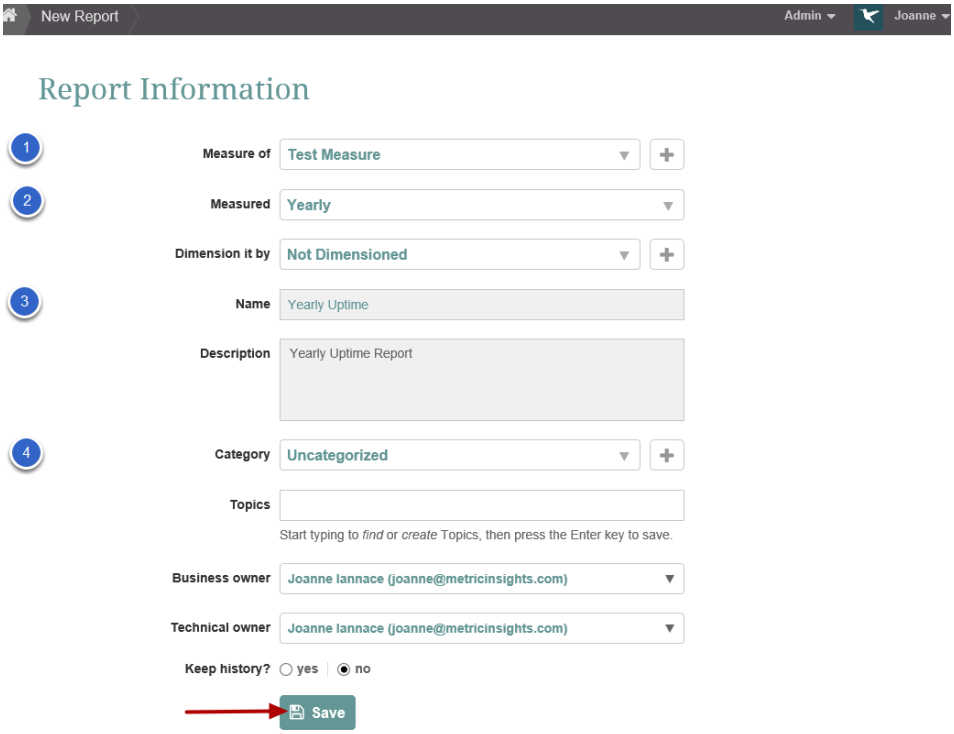
39.2 How to Collect Data using Uptimerobot

This article will show you how to create an Element using a **Uptimerobot** plug-in as a data source. It assumes that you have already established connectivity to Uptimerobot.

1. Add a new element based on your Uptimerobot plug-in data source



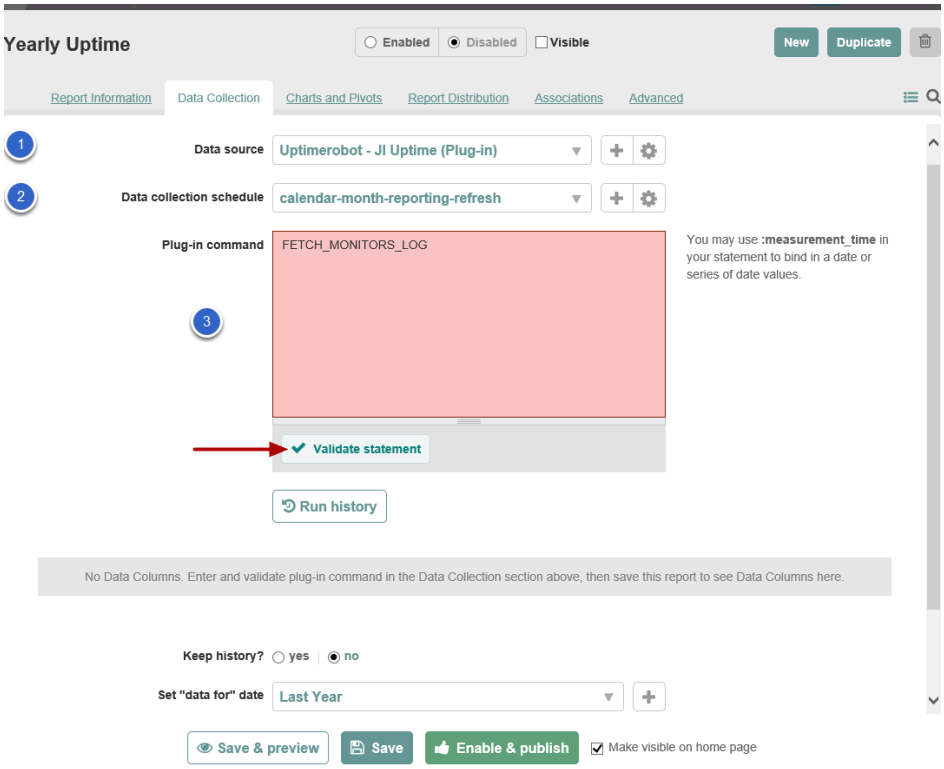
2. Provide basic information on Wizard (or Editor) - report example



1. Specify what this report is **measuring**. If you do not see the measure that you want to use, you can create one from this drop-down
2. Select the **Measurement Interval** that applies to your element
3. Give the element a unique **name**
4. Optionally, assign a **Category**

Save

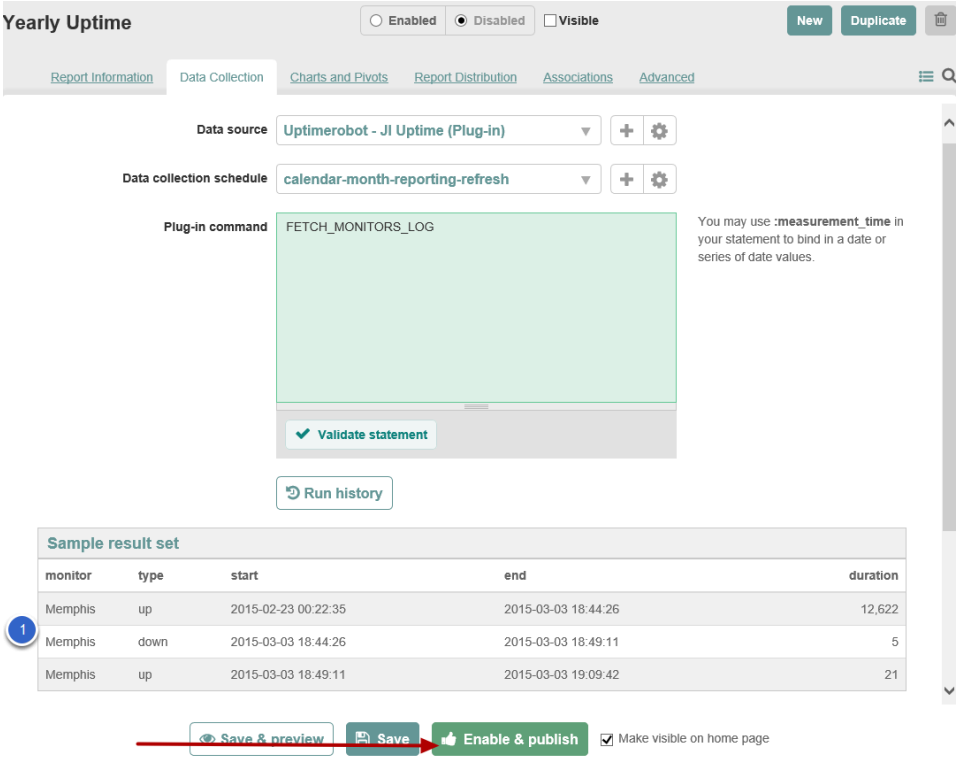
3. Full Editor displays the Data Collection tab



1. Select **Uptimerobot** plug-in in **Data Source** drop-down
2. Set **Data collection schedule**
3. Input **Plug-in Command**

Validate statement

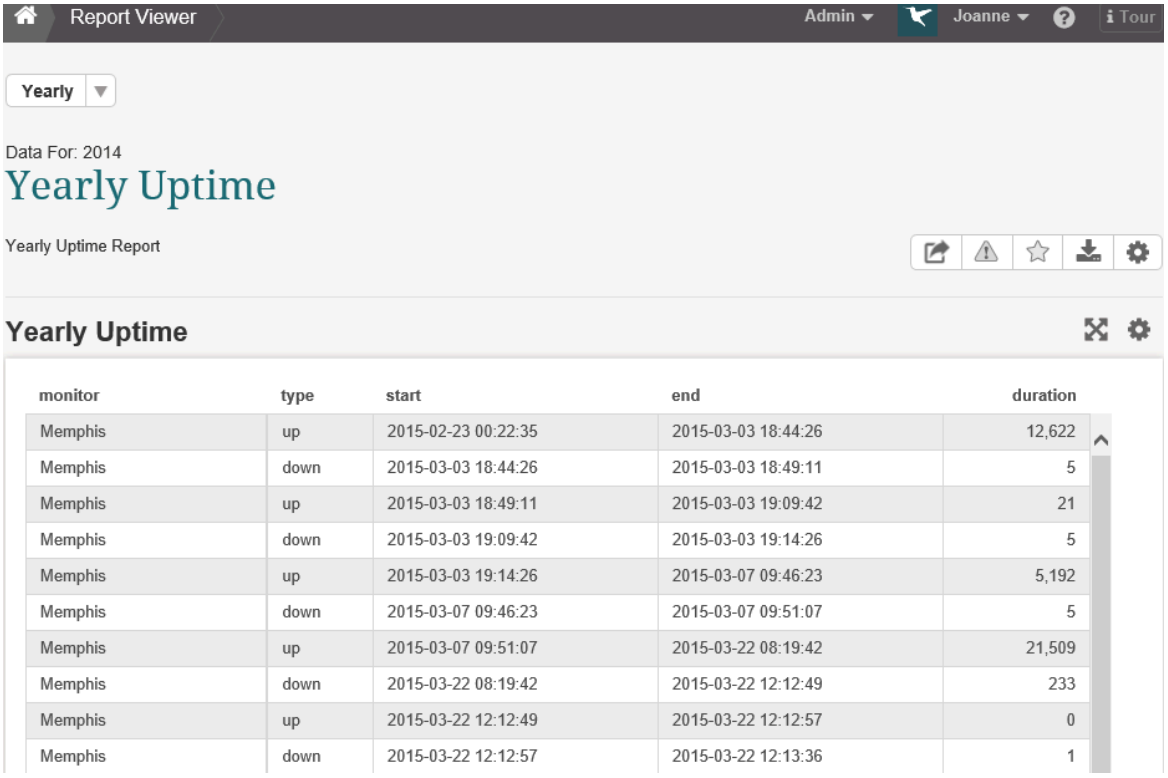
4. Enable & Publish



1. Sample result set returned

Enable & publish

5. Full Report is displayed in viewer



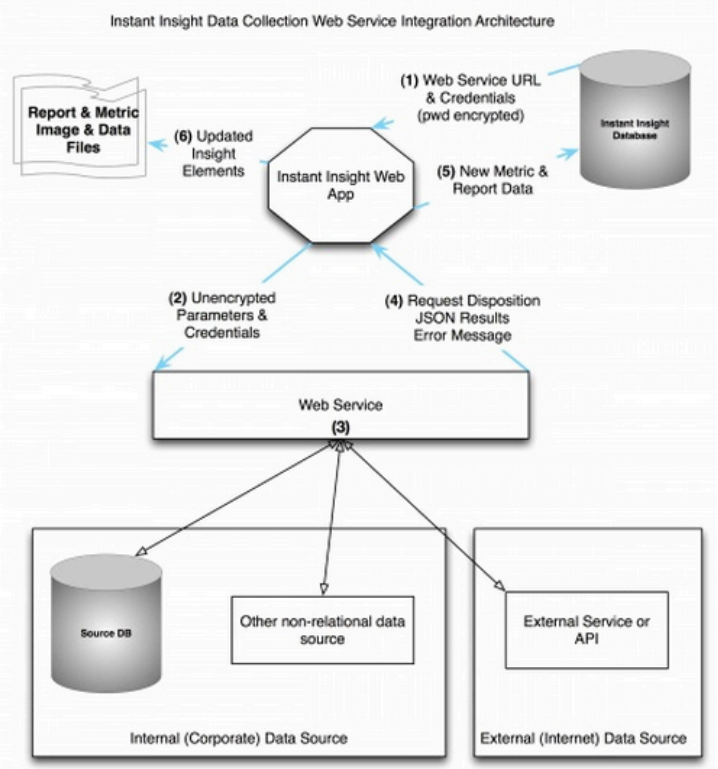
monitor	type	start	end	duration
Memphis	up	2015-02-23 00:22:35	2015-03-03 18:44:26	12,622
Memphis	down	2015-03-03 18:44:26	2015-03-03 18:49:11	5
Memphis	up	2015-03-03 18:49:11	2015-03-03 19:09:42	21
Memphis	down	2015-03-03 19:09:42	2015-03-03 19:14:26	5
Memphis	up	2015-03-03 19:14:26	2015-03-07 09:46:23	5,192
Memphis	down	2015-03-07 09:46:23	2015-03-07 09:51:07	5
Memphis	up	2015-03-07 09:51:07	2015-03-22 08:19:42	21,509
Memphis	down	2015-03-22 08:19:42	2015-03-22 12:12:49	233
Memphis	up	2015-03-22 12:12:49	2015-03-22 12:12:57	0
Memphis	down	2015-03-22 12:12:57	2015-03-22 12:13:36	1

40. Sourcing Data using a Web Service

40.1 Connecting to a Web Service

You can source metric and report data from any web service that returns information using a JSON message format that conforms to Metric Insights Query Response format.

Architecture Overview



The diagram above illustrates graphically the Metric Insights (MI) Web Services architecture and high-level process flow; the numbering on the chart corresponds to the list below:

1. The Web Service URL and parameter data its obtained from the MI database; if required for the call, Web Service credential data (username and password) is also retrieved and decrypted, if necessary
2. An HTTP POST request is made using the parameters described in the '**Web Service POST Data**' section below and uses the unencrypted credentials (that were retrieved in Step 1) to authenticate to the Web Service, using HTTPS, if requested
3. The Web Service performs a data fetch process as defined by the developer of that service
4. The Web Service returns the results from the data fetch process to Metric Insights in JSON format as described in the **'Data Returned from Web Service**' section
5. MI parses JSON data and updates its database with the returned Metric data or Report instance data

6. Data received from the Web Service is used by MI to generate new/updated tables and charts according to the parameters defined for the Metric or Report associated with the Web Service call

1. Adding a New Authentication

If you need to establish the the values for the 'Authentication' field on an Element's Editor, you use to the right of the field's text display. There you add the URL for the source of your data. The authentication process associates a Username and password with the URL as it constructs the web service call.

2. Using Web Service POST Data

The system collects and passes the following information to the Web Service when performing the data fetch command:

2.1. Web Service URL

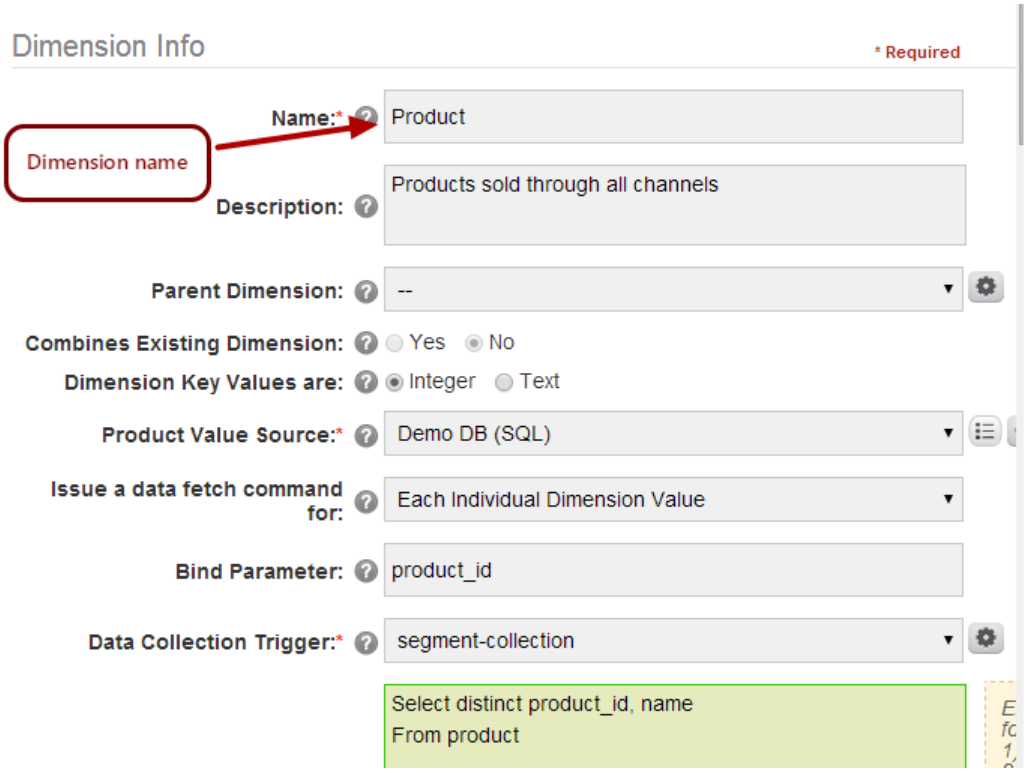
The Web Service URL is provided by the Administrator via the Report or Metric Editor and is used by the system to determine both the location of the Web Service and whether the fetch is to be performed using HTTPS or HTTP. In addition to any information specified directly in the URL, the table below contains items that are automatically appended to the list of HTTP POST parameters as determined by the type of element:

2.2. Information Added to POST

Information Added to POST	Variable Name in POST	Included in *Report* Web Service Call?	Included in *Metric* Web Service Call?
Element ID	element_id	Yes	Yes
Last Measurement Time for which a value was recorded for the metric and segment value	last_measurement_time	No	Yes
Measurement time to use for the report data fetch	measurement_time	Yes	No
Segment Name	segment_name	Yes	Yes
Segment Value (for segmented elements only)	segment_value	Yes	Yes

The table above illustrates the information that is appended to the POST that is performed to the Web Service URL

2.3. Dimension Values



As shown in the table in the preceding section, the dimension_name information is passed to the Web Service URL for Metrics and Reports that are dimensioned.

2.4. Dimension Values

A separate Web Service call is performed for every 'dimension Value' defined for the 'dimension'. Each individual 'Key Value' of that dimension is passed in the dimension_value

parameter. This value may either be an integer or text, depending upon the definition of the dimension.

2.5. Date Formats from Metrics and Reports

- Metrics:

```
last_measurement_time
```

- Reports:

```
measurement_time
```

The POSTed parameters shown above are re-formatted prior to substitution, based on the date format mask specified in the Web Service credentials associated with the Insight Element:

2.6. Date Formats if the 'MySQL to Web Service Format' is not specified

Parameters are passed using the following format:

```
YYYY-MM-DD HH24:MI:SS format
```

3. Authentication Credentials

If Username and/or password data is present for the credentials associated with the Insight Element, these parameters are passed to the web service using HTTP Basic authentication.

NOTE: It is important that an Element's developer understand that authentication credential information that the Web Service requires in order to perform the data fetch, including Usernames and passwords to other external services, must be managed by the Web Service since this information is not stored or processed by Metric Insights. Authentication and data collection processing performed by the Web Service occurs outside of this system.

4. Web Service Processing

The Web Service performs whatever processing is required to collect data for the Metric or Report. Data can be sourced by the Web Service from internal corporate information resources as well as from external systems hosted outside the corporate firewall. The Web Service is triggered by Metric Insights based on the execution of the **Data Collection Trigger** associated with the Element as well as when the Administrator performs the Web Service call through the Editor.

5. Data Returned from Web Service for Metrics

Data collected by the Web Service is returned in JSON with date/time values returned from the service conforming to the following standard format:

```
"YYYY-MM-DD HH24:MI:SS"
```

If it does not meet this requirement, it must conform to the date format string specified in the web services credentials record associated with the Insight Element:

```
web_service_credentials.web_service_to_mysql_date_format_string (Web Service to MySQL Date Format)
```

5.1. Data Returned for a Metric

```
'result' = {"header":[{"name":"col1_name","type":"<col1_type>"},{"name":"col2_name","type":"<col2_type>"}],
'data ': [
[val1,val2]
.
.
.
]
}
```

Metric Insights performs the following validation on the returned Metric data set before processing the data to ensure that:

- The Web Service always returns data sets that include one numeric (type of “integer” or “decimal”) and one date/time value (type of “date”) per row
- If no date/time format is specified in the Web Service credentials for the Metric, all date/time column values conform to the date format of

"YYYY-MM-DD HH24:MI:SS"

- If a date/time format is provided as part of the Web Service credentials for the Metric, all date/time column values conform to the specified format mask

5.2. Metric Example

1. If a Web Service uses the following SQL statement to generate a dataset:

```
Select measurement_time 'Calendar Date', sum(measurement_value) 'Total Sales'
From...
```

The Result Set contains the following 2 rows:

```
['2011-01-01',1234.10] ,
['2011-01-02',5678.00]
```

The returned JSON would be:

```
{"header":[{"name":"Calendar Date","type":"DATE"}, {"name":"Total
Sales","type":"DECIMAL"}],
"data": [
["2011-01-01 00:00:00",1234.10]
["2011-01-02 00:00:00",5678.00]
]
}
```

The example above has been arranged for readability but a Web Service JSON result set does not have to be formatted and would typically look like the following example:

```
{"header":[{"name":"Calendar Date","type":"DATE"},{"name":"Total
Sales","type":"DECIMAL"}], "data":[["2011-05-10 07:00:00",4730661.59],["2011-05-11
07:00:00",4602004.14],["2011-05-12 07:00:00",4635604.11],["2011-05-13 07:00:00",4873962.
11],["2011-05-14 07:00:00",4614745.529999999],["2011-05-15 07:00:00",4699752.8],["2011-
05-16 07:00:00",4774199.8],["2011-05-17 07:00:00",4793545.17],["2011-05-18
07:00:00",4529600.81],["2011-05-19 07:00:00",4605180.539999999]]}
```

5.3. Returned Metric Data Validation

Metric Insights performs the following validation on the returned Metric data set before processing the data to ensure that:

- The Web Service always returns data sets that include one numeric (type of "integer" or "decimal") and one date/time value (type of "date") per row
- If no date/time format is specified in the Web Service credentials for the Metric, all date/time column values conform to the date format of

```
        YYYY-MM-DD HH24:MI:SS
```

- If a date/time format is provided as part of the Web Service credentials for the Metric, all date/time column values conform to the specified format mask

6. Data Returned from Web Service for Reports

Web Services that populate Reports must return the three JSON elements listed below:

1. Header Column Names
2. Column data types (“DATE”,”DECIMAL”,”INTEGER”,”TEXT”) given that the “DATE” data type is used for date-time values as well as date-only values
3. Data Set values for each row

6.1. Expected JSON structure for a result set with four columns and three rows

```
{
"header": [
{
"type": "DATE",
"name": "Order Date"
},
{
"type": "DECIMAL",
"name": "US order Volume (US$)"
},
{
"type": "DECIMAL",
"name": "Intl order Volume (US$)"
},
{
"type": "DECIMAL",
"name": "Total order Volume (US$)"
}
],
"data": [
[
"2011-04-06 00:00:00",
415037.54999999999,
758473.73999999999,
1173511.29
],
[
"2011-04-05 00:00:00",
```

```
346160.52000000002,
738350.46999999997,
1084510.99
]
]
}
```

7. Error Reporting

Any error encountered in Web Service processing are returned to Metric Insights so that the Administrator can be notified of any problems. The following information must be included in the JSON message (for both metric and report data fetches) when an error is encountered:

```
{
"error": ''| 'error string'
}
```

7.1. Returned Report Data Validation

Any errors encountered in Web Service processing are returned to Metric Insights so that the Administrator can be advised of any problems. The information that must be included in the JSON message (for both metric and report data fetches) when an error is encountered is shown above.

Metric Insights logs (or displays in the Editor during validation) both error messages explicitly returned in the JSON message as well as standard HTTP header error codes; e.g., 404,503,505. For a data fetch to be considered successful all of the following conditions must be satisfied:

- The web service call returns a 'OK' HTTP status
- No "error" section exists in the returned JSON
- The returned JSON format adheres to the provided JSON specification
- All validation rules set for the Metric's or Report's data are satisfied

8. Remotely Invoking Web Service Calls

Feature coming soon!

9. Sample Code

The following sub-steps contain examples of Web Services for consumption of Metric and Report data written in php and python:

9.1. PHP Web Service example for collecting metrics data. Note the use of last_measurement_time.

```
Above is a Php
<?php
/**
* @see Devx_Model
*/
require_once 'Devx/Model.php';
/**
* DemoModel object
*
* @version 1.0
* @package Insight
*/
class DemoModel extends Devx_Model
{
public function getDemoData1($subst) {
$cfg = Zend_Registry::get('config');
$password = Custom_Model_External::decryptPassword($cfg->database->password);
$params = array('host'     => $cfg->database->host,
'username' => $cfg->database->username,
'password' => $password,
'dbname'   => 'demo',
'port'     => 3306 );
try {
$db = new Devx_Db_Adapter_Pdo_Mysql($params);
$db->query('SET NAMES "utf8"');
$sql = "select sum(amount), date(order_time)
from customer_order ord, customer_order_detail ord_line
where ord.order_id = ord_line.order_id
and ord.order_time > :last_measurement_time
and date(ord.order_time) < now()
group by 2";
if (!isset($subst['last_measurement_time'])) return array('error' => 'No substitution
provided for \':last_measurement_time\' pattern');
$pattern = ':last_measurement_time';
$sql = trim(str_ireplace($pattern, "'" . $subst['last_measurement_time'] . "'", $sql));
$rows = $db->query($sql)->fetchAll();
//print_r($rows);
if (is_array($rows)) {
return array(
'header' => array(
array('name' => 'sum(amount)', 'type' => 'decimal'),
array('name' => 'date(order_time)', 'type' => 'date')
),
```

```
'data' => $rows
);
} else return array('error' => 'Unexpected data structure is returned');
} catch (Exception $ex) {
return array('error' => $ex->getMessage());
}
}
}
?>
```

9.2. Python code example used to collect report data using a Web Service”

```
#!/usr/bin/env python2.5
"""
Requirements: apache, mod_python, python2.5, MySQLdb
Sample of virtual host file:
<Directory /var/www/generator/>
    Options MultiViews
    Order allow,deny
    allow from all
    AddHandler mod_python .py
    PythonHandler webservice
    PythonAuthenHandler webservice
    AuthType Basic
    AuthName "Restricted Area"
    require valid-user
    AuthBasicAuthoritative Off
    PythonDebug On
    PythonOption mod_python.legacy.importer *
</Directory>
<VirtualHost *:80>
    DocumentRoot /var/www/generator/
    ServerName generator
    ServerAlias www.generator
</VirtualHost>
Needed modules: apache, util (from mod_python)
Local modules: simplejson

"""
import os
import sys
import datetime
import MySQLdb
path = os.path.abspath(os.path.dirname(__file__))
sys.path.append(path)
```

```
import simplejson
import logging
import logging.handlers
import os, tempfile
from datetime import date
class MLogger:
    def __init__(self, name):
        self._logger = logging.getLogger(name)
        self._logger.setLevel(logging.INFO)
        log_name = 'log-%s-.txt' % date.today()
        full_log_dir = '/var/www/generator/log/'#os.path.join(os.path.split(os.path.
split(os.path.split(os.path.abspath( __file__ ))[0])[0])[0], 'log')
        full_log_name = os.path.join(full_log_dir, log_name)
        try:
            os.chmod(full_log_name, 0777)
        except OSError:
            pass
        try:
            self._ch = logging.FileHandler(full_log_name)
        except IOError:
            tmp = tempfile.mkstemp(prefix='log_', dir = full_log_dir)
            self._ch = logging.FileHandler(tmp[1])
        self._formatter = logging.Formatter("%(asctime)s - %(name)s - %(levelname)s -
%(message)s","%Y-%m-%d %H:%M:%S")
        self._ch.setFormatter(self._formatter)
        self._logger.addHandler(self._ch)
    def get_logger(self):
        return self._logger
"""
local testing data
"""
#main date format
datetime_format = '%Y-%m-%d %H:%M:%S'
date_format = '%Y-%m-%d'
def unformat_date(var):
    """
    unformat string to datetime
    """
    date = None
    if var:
        try:
            date = datetime.datetime.strptime(var, datetime_format)
        except:
            try:
                date = datetime.datetime.strptime(var, date_format)
            except:
                # cannot format it
                pass
```

```
    return date
def format_date(var):
    """
    unformat datetime to string
    """
    date = None
    if var:
        try:
            date = datetime.datetime.strftime(var, datetime_format)
        except:
            # cannot format it
            pass
    return date
"""
Login and password to access this script
"""
web_service_credentials = {'username': 'user',
                      'password': ''
                       #'password': 'U2FsdGVkX19z/09S2MlKaiqCS3YmkwcCnOPqnFkX1Yc='
                    }
reports = {27: {'data_fetch_command_sql':
                    """
                        SELECT calendar_date 'Order Date',
                            sum(if( country = 'United States', total_amount, 0)) 'US
order Volume (US$)',
                            sum(if( country = 'United States', 0, total_amount)) 'Intl
order Volume (US$)',
                            sum(total_amount) 'Total order Volume (US$)'
                        FROM daily_order_summary
                        WHERE calendar_date > date(%(measurement_time)s) - INTERVAL 60
DAY
                            AND channel = %(channel)s
                        GROUP BY 1
                        ORDER BY 1 DESC
                    """,
                'measurement_time_fetch_command_sql':
                    """
                    SELECT DISTINCT calendar_date
                        FROM demo.daily_order_summary
                    WHERE calendar_date < date(now())
                        AND calendar_date > date(%(last_measurement_time)s)
                    """
                }
            }
field_type = {
    0: 'DECIMAL',
    1: 'TINY',
    2: 'SHORT',
```

```
    3: 'LONG',
    4: 'FLOAT',
    5: 'DOUBLE',
    6: 'NULL',
    7: 'TIMESTAMP',
    8: 'LONGLONG',
    9: 'INT24',
    10: 'DATE',
    11: 'TIME',
    12: 'DATETIME',
    13: 'YEAR',
    14: 'NEWDATE',
    15: 'VARCHAR',
    16: 'BIT',
    246: 'NEWDECIMAL',
    247: 'INTERVAL',
    248: 'SET',
    249: 'TINY_BLOB',
    250: 'MEDIUM_BLOB',
    251: 'LONG_BLOB',
    252: 'BLOB',
    253: 'VAR_STRING',
    254: 'STRING',
    255: 'GEOMETRY' }
simple_field_type = {
    0: 'DECIMAL',
    1: 'INTEGER',
    2: 'INTEGER',
    3: 'INTEGER',
    4: 'DECIMAL',
    5: 'DECIMAL',
    6: 'TEXT',
    7: 'DATE',
    8: 'INTEGER',
    9: 'INTEGER',
    10: 'DATE',
    11: 'DATE',
    12: 'DATE',
    13: 'DATE',
    14: 'DATE',
    15: 'NVARCHAR',
    16: 'INTEGER',
    246: 'DECIMAL',
    247: 'TEXT',
    248: 'TEXT',
    249: 'TEXT',
    250: 'TEXT',
    251: 'TEXT',
```

```
    252: 'TEXT',
    253: 'NVARCHAR',
    254: 'NVARCHAR',
    255: 'TEXT' }
_NAME = 'channel'
class MysqlConnect(object):
    error = ''
    connection = None
    headers = []
    #headers_types = []
    result = None
    rows = []
    def __init__(self, *args, **kargs):
        self.info = {
                     'host': 'localhost',
                     'user': 'generators',
                     'passwd': 'p0rtal',
                     'db': 'demo',
                     'port': 3306,
                     'use_unicode': True,
                     'charset': 'utf8'
                     }
        if kargs.has_key('host'):
            self.info['host'] = kargs['host']
        if kargs.has_key('user'):
            self.info['user'] = kargs['user']
        if kargs.has_key('passwd'):
            self.info['passwd'] = kargs['passwd']
        if kargs.has_key('db'):
            self.info['db'] = kargs['db']
        if kargs.has_key('port'):
            self.info['port'] = int(kargs['port'])
    def connect(self):
        try:
            self.connection = MySQLdb.connect(*[], **self.info)
            return True
        except MySQLdb.Error, e:
            self.error = "%d %s" % (e.args[0], e.args[1])
        except Exception, e:
            self.error = e
        return False
    def close(self):
        if self.connection is not None:
            try:
                self.connection.close()
            except Exception, e:
                pass
    def query(self, query, params):
```

```
        try:
            cursor = self.connection.cursor(MySQLdb.cursors.Cursor)
            cursor.execute(query, params)
        except MySQLdb.Error, e:
            self.error = "%d %s" % (e.args[0], e.args[1])
            return False
        try:
            self.result = {'header': [{'name': header[0], 'type':
simple_field_type[header[1]]} for header in cursor.description],
                      'data': []}
            records = cursor.fetchall()
            for record in records:
                row = []
                for i, item in enumerate(record):
                    if self.result['header'][i]['type'] == 'DATE':
                        item = item.strftime(datetime_format)
                    else:
                        item = unicode(item)
                    row.append(item)
                self.result['data'].append(row)
            #self.json_result = simplejson.dumps(result)
            return True
        except Exception, e:
            self.error = e
        return False
def is_int(s):
    try:
        int(s)
        return True
    except ValueError:
        return False
def authenhandler(req):
    pw = req.get_basic_auth_pw()
    user = req.user
    if user == web_service_credentials['username'] and (
            (web_service_credentials['password'] and pw ==
web_service_credentials['password']) or not web_service_credentials['password']):
       return apache.OK
    else:
       return apache.HTTP_UNAUTHORIZED
def handler(req):
    """
    Binds handler routing
    """
    req.log_error('handler')
    req.content_type = 'application/json'
    #req.content_type = 'text/html'
    req.send_http_header()
```

```
    form = util.FieldStorage(req, keep_blank_values=1)
    process(form, req, ret_answer)
    return apache.OK
def ret_answer(ret, req):
    """
    Formats answer to json answer and returns to apache
    """
    #req.write(simplejson.dumps(ret, indent=4))
    req.write(simplejson.dumps(ret))
    return apache.OK
def print_answer(ret, req):
    """
    Print answer to stdout. For test purposes.
    """
    print simplejson.dumps(ret,4)
    pass

def process(form, req, ret_answer):
    """
    Main routine
    """
    # empty answer dict
    #ret = {'error': ''}
    ret = {}
    # check for last_measurement_time field
    if 'measurement_time' in form:
        form['last_measurement_time'] = None
    log = MLogger('webservice')
    logger = log.get_logger()
    logger.info('before elem id checks')
    # check for element_id field
    if 'element_id' not in form:
        ret['error'] = 'ERROR. element_id is not set'
        ret_answer(ret, req)
        return
    # check if element_id is correct
    if not is_int(form['element_id']) or int(form['element_id']) not in reports:
        ret['error'] = 'ERROR. element_id is incorrect %s ' % form['element_id']
        ret_answer(ret, req)
        return
    element_id = int(form['element_id'])
    # get mysql connection
    outer_conn = MysqlConnect()
    if not outer_conn.connect():
        ret['error'] = "ERROR. Cannot connect to db: %s" % outer_conn.error
        ret_answer(ret, req)
        return
    if 'command' in form and form['command'] == 'get_measurement_times':
```

```
        if 'last_measurement_time' in form and form['last_measurement_time']:
            last_meas_time = unformat_date(form['last_measurement_time'])
        else:
            last_meas_time = datetime.datetime(1900, 1, 1, 0, 0, 0)
        if not last_meas_time:
            last_meas_time = datetime.datetime(1900, 1, 1, 0, 0, 0)
        query = reports[element_id]['measurement_time_fetch_command_sql']
        params = {'last_measurement_time': last_meas_time}
    else:
        # check for  value substitution
        _value = ''
        if _NAME in form:
            _value = unicode(form[_NAME])
        if not _value:
            ret['error'] = "ERROR. _value is not specified"
            ret_answer(ret, req)
            return
#        # check for  names substitution
#        _name = ''
#        if '_name' in form:
#            _name = unicode(form['_name'])
#
#        if not _name:
#            ret['error'] = "ERROR. _name is not specified"
#            ret_answer(ret, req)
#            return

        # check for measurement time
        #if 'measurement_time' in form and form['measurement_time']:
        #    meas_time = unformat_date(form['measurement_time'])
        if 'last_measurement_time' in form and form['last_measurement_time']:
            meas_time = unformat_date(form['last_measurement_time'])
        else:
            meas_time = None
        if not meas_time:
            ret['error'] = "ERROR. Measurement time is required"
            ret_answer(ret, req)
            return
        query = reports[element_id]['data_fetch_command_sql']
        params = {'measurement_time': meas_time, _NAME: _value}
    if not outer_conn.query(query, params):
        ret['error'] = "ERROR. Cannot execute query: %s" % outer_conn.error
        ret_answer(ret, req)
        return
    result = outer_conn.result

    if not result:
        ret['error'] = "ERROR. Source db returned empty result"
```

```
        ret_answer(ret, req)
        return
    ret['header'] = result['header']
    ret['data'] = result['data']
    ret_answer(ret, req)
    return
if __name__ == "__main__":
    """
    for testing from bash
    """
    form = {'element_id': 27, 'username': u'user', 'meas_time': '', '_value':
u'corporate sales', '_name': u'channel', 'last_measurement_time': None, 'password': ''}
    if len(sys.argv) >= 3:
        form['command'] = 'get_measurement_times'
        form['last_measurement_time'] = sys.argv[2]
    elif len(sys.argv) >= 2:
        form['measurement_time'] = sys.argv[1]
    process(form, sys.stdout, print_answer)
else:
    from mod_python import apache, util
    directory = os.path.dirname(__file__)
```

41. Sourcing Data from Zendesk

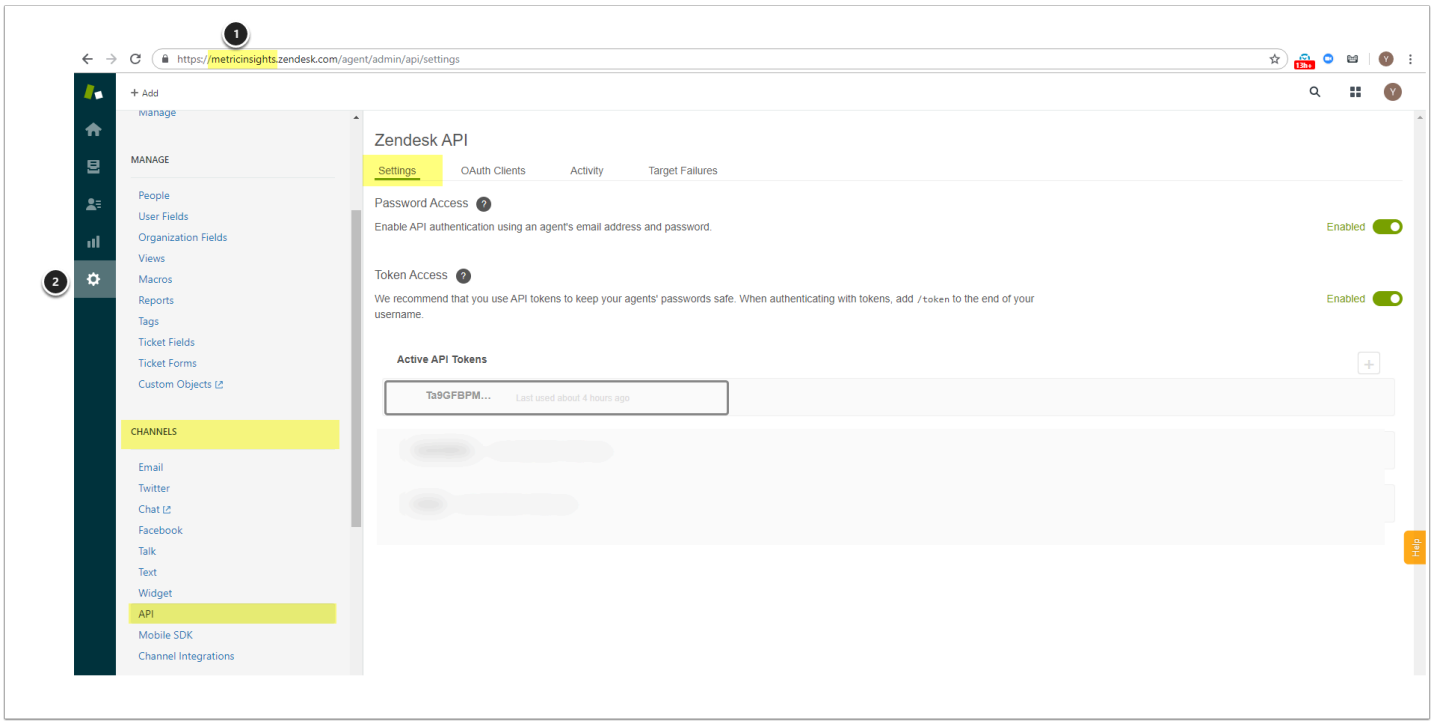
41.1 Obtaining Credentials for Zendesk plugin

To be able to collect data, your Zendesk plugin will require the following **credentials**:

1. Zendesk Domain
2. Security Token

NOTE: While Domain name is accessible to All Zendesk Users, you must be an Admin to obtain the Token

Login to your Zendesk Account as Admin



1. **Domain:** your Zendesk domain identifier can be obtained from the account's URL: https://[**yoursubdomain**].zendesk.com
2. **API Token:** API Tokens are used as part of authentication; you can use the Token that has already been generated for your account or create a new one
   - Access the Token configuration page via *Admin (gear) icon* > *Channels* > *API* > *Settings tab*
   - For more information, go to Generating a new Token

Both User Domain and the API Token are necessary to configure the *Required Parameters* for the Zendesk Plugin.

What's next?

After getting User authentication credentials, you can proceed to creating a connection profile with Zendesk via the Metric Insights UI.

41.2 Establish Connectivity to Zendesk

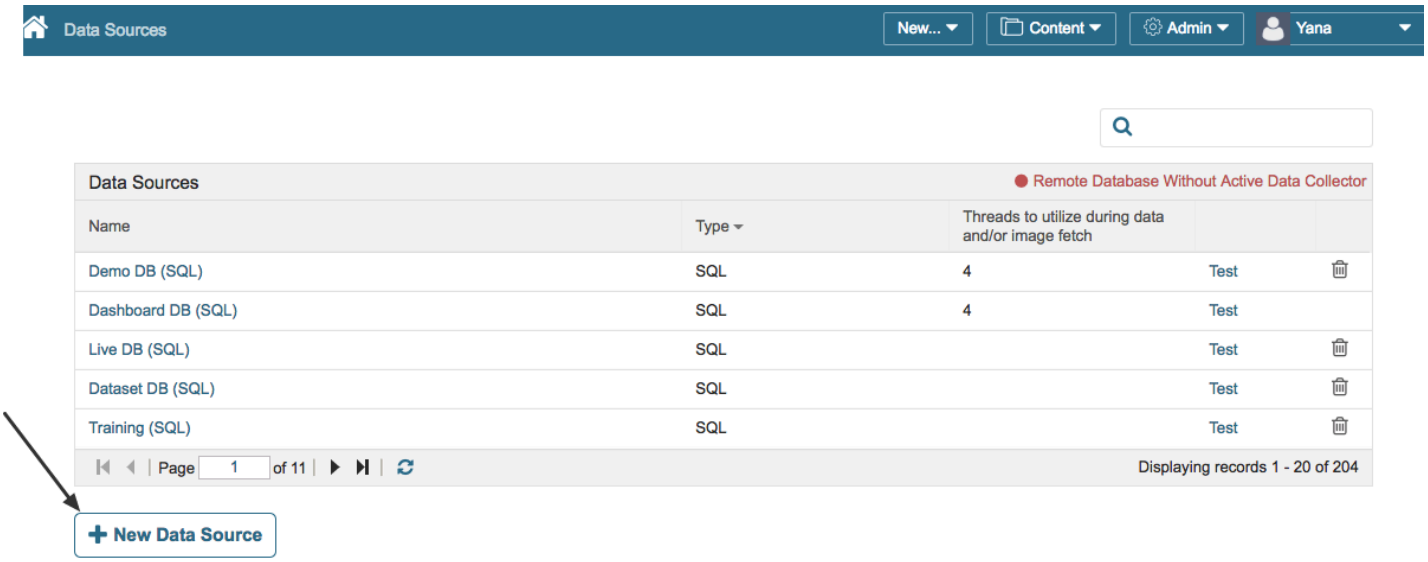
This article describes how to connect to **Zendesk** in order to load data into Datasets and Reports in Metric Insights.

PREREQUISITES:

- Obtaining Credentials for Zendesk plugin

[Release 6.x]: 'Remote Data Collector' renamed to 'Remote Data Processor'

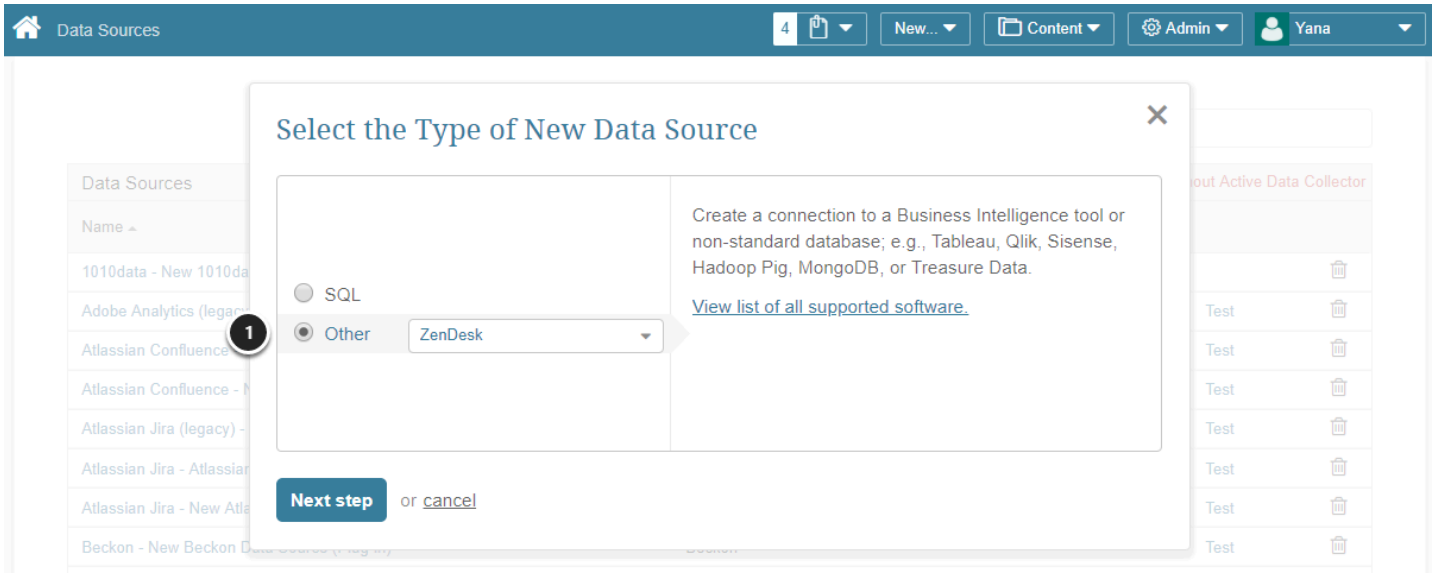
1. Access Admin > Data Sources



At the bottom of the screen click **[+ New Data Source].**

The *Select the Type of New Data Source* pop-up opens.

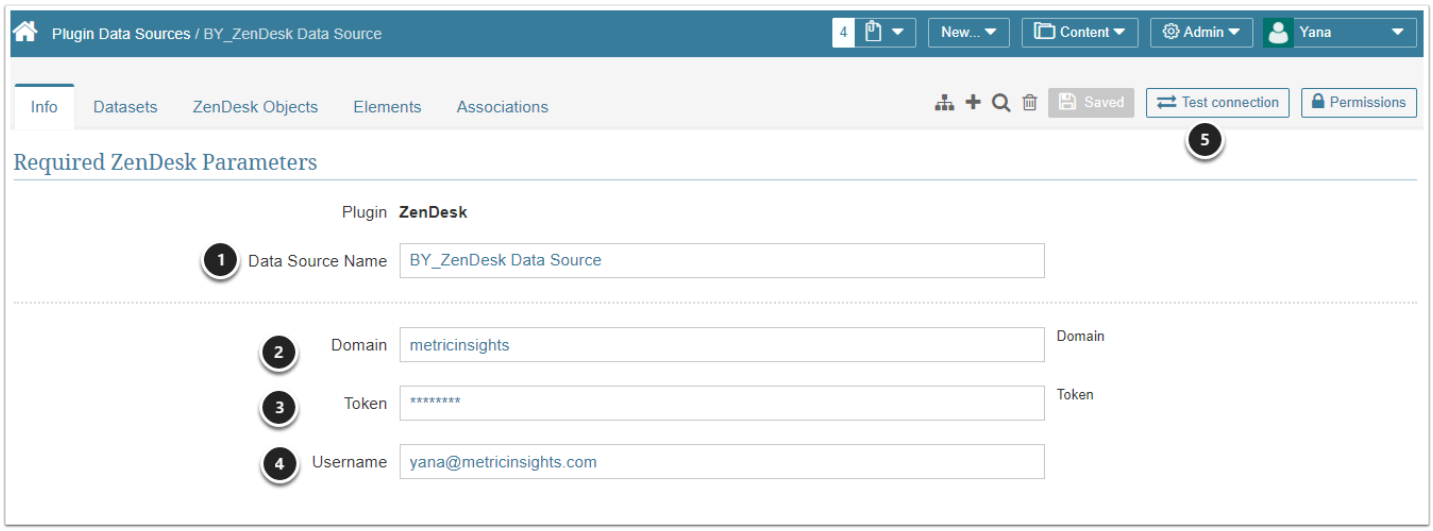
2. Select the Type of New Data Source



1. Select "Other" and choose "Zendesk" from the drop-down list

Next step

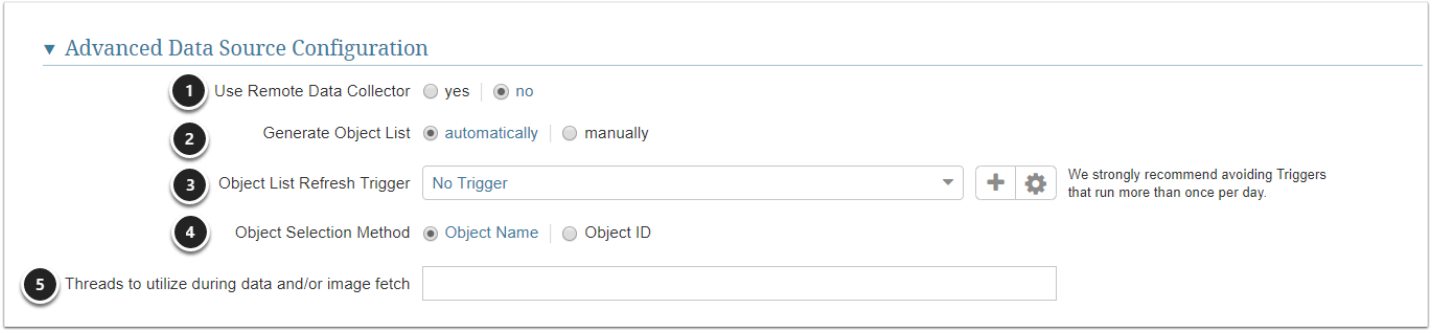
3. Provide the Required Parameters



1. **Data Source Name:** is defaulted but you may modify it
2. Enter the **Domain** identifier: can be obtained from your account's URL: https://[yourdomain].zendesk.com
3. Input your Zendesk Security **Token**
4. **Username:** provide your Username to access Zendesk
5. **Test Connection** (this will also **Save** your data)

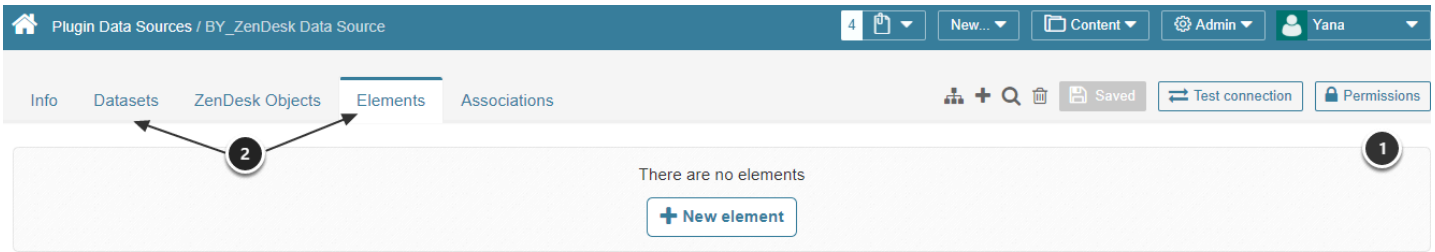
If your connection was successful, you may proceed to configuring *Advanced Parameters*

4. Advanced Configuration



1. **Use Remote Data Collector:** if required, set to "yes"
2. **Generate Object List**
   - *automatically*: all Objects are going to be fetched by the system
   - *manually*: Objects may be added one-by-one or via CSV file
3. **Object List Refresh Trigger:** from the dropdown, select the Trigger that will be used to fetch data via the Zendesk plugin
4. **Object Selection Method:** specify how Zendesk Objects will be fetched
5. Optionally, state the number of **Threads to utilize during data and/or image fetch** to be used in background processing when the system updates Objects for this Data Source
   - If you do not specify any value for this setting, batch data collection processing will be single-threaded

5. Other Settings



1. You can create Datasets or Elements directly from the respective tabs
2. Click **Permissions** to assign permissions to the Data Source to Groups or Power Users

What's next?

[How to Collect Data from Zendesk]

41.3 How to Collect Data from Zendesk (Dataset)

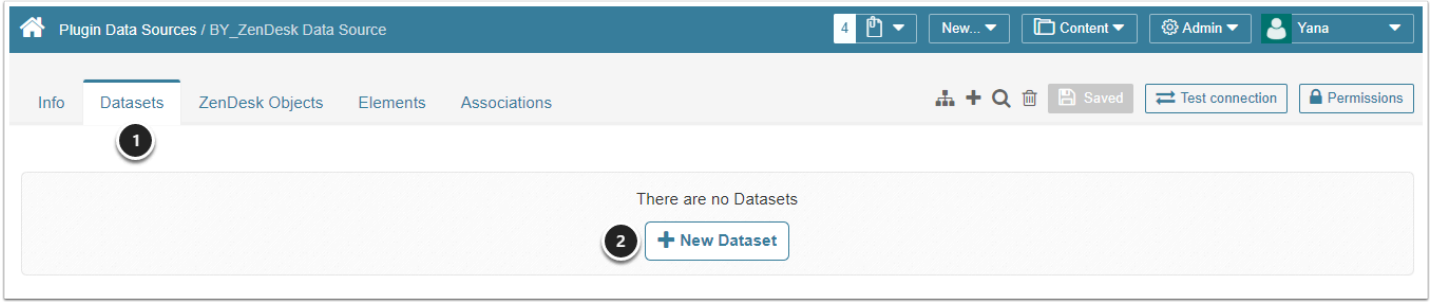
A Metric Insights' Dataset can be populated automatically based on data fetched from Zendesk.

PREREQUISITE:

You must have already established connectivity to your **Zendesk** server via the respective plugin connection profile.

This article describes how to create a Dataset from the **Plugin Data Sources Editor**.

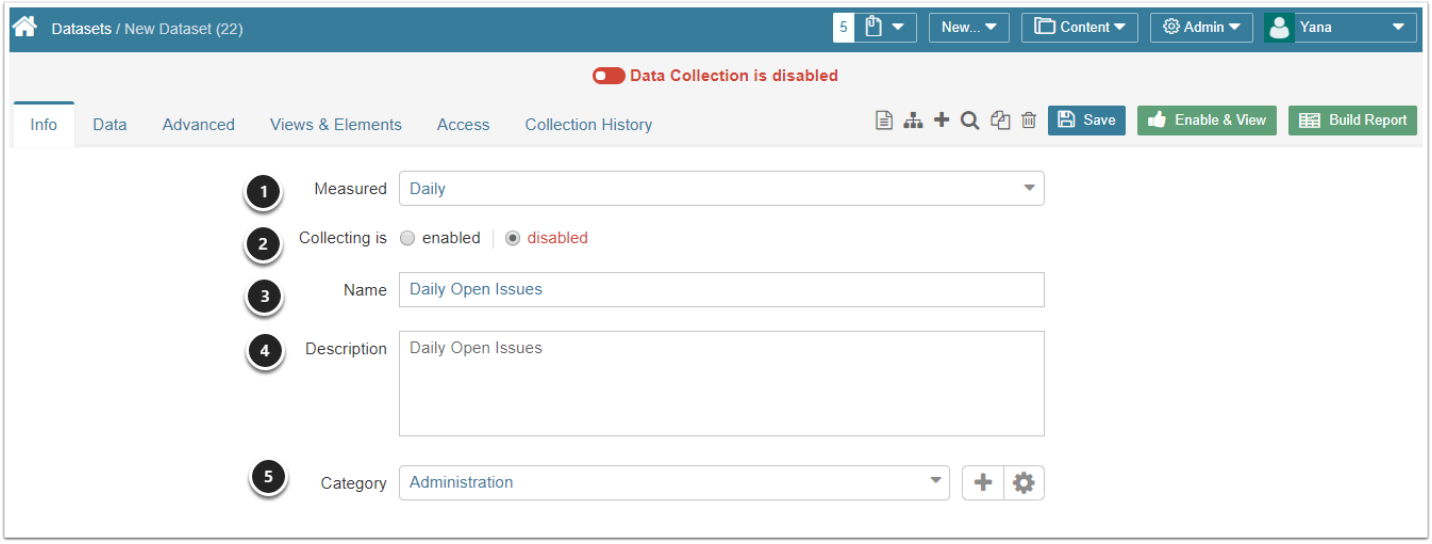
1. Plugin Data Sources Editor > Datasets tab



1. Access **Plugin Data Sources Editor > Datasets tab**
2. Click **[New Dataset]**

You will be redirected to the Dataset Editor.

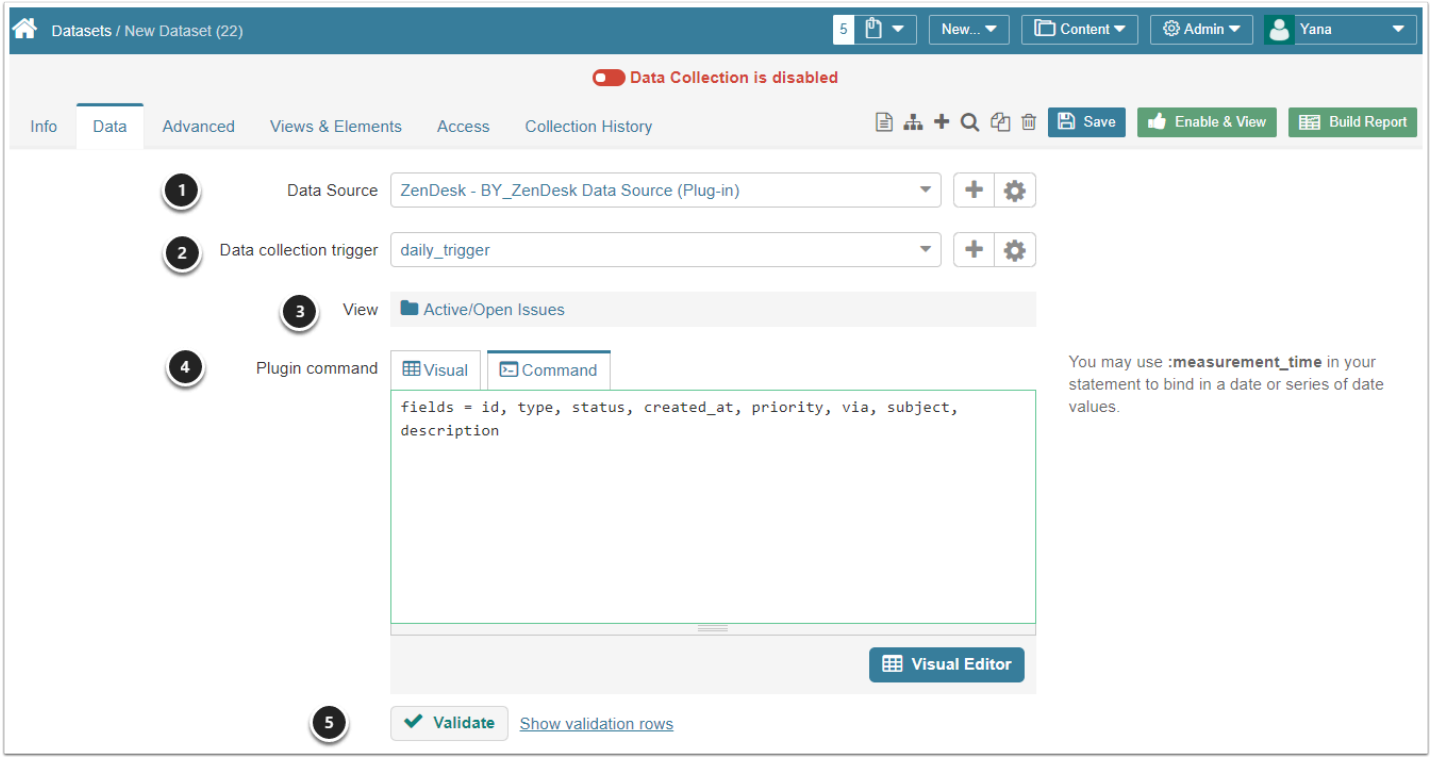
2. Dataset Editor > Info tab



1. **Measured:** select the measurement interval that applies to the level of aggregation that you want in your result set
2. **Collecting**: new Datasets are always disabled by default to make sure that you can take time to configure them properly before enabling. This setting is duplicated at the top of the screen
3. **Name:** provide a unique name for your Dataset. Preferably, the Dataset name should explain what kind of data it contains
4. **Description:** optionally, provide any additional information about your Dataset
5. **Category:** specify the Category where you Dataset will be placed

Move to the ***Data tab*** to define the source of data and how often it should be updated

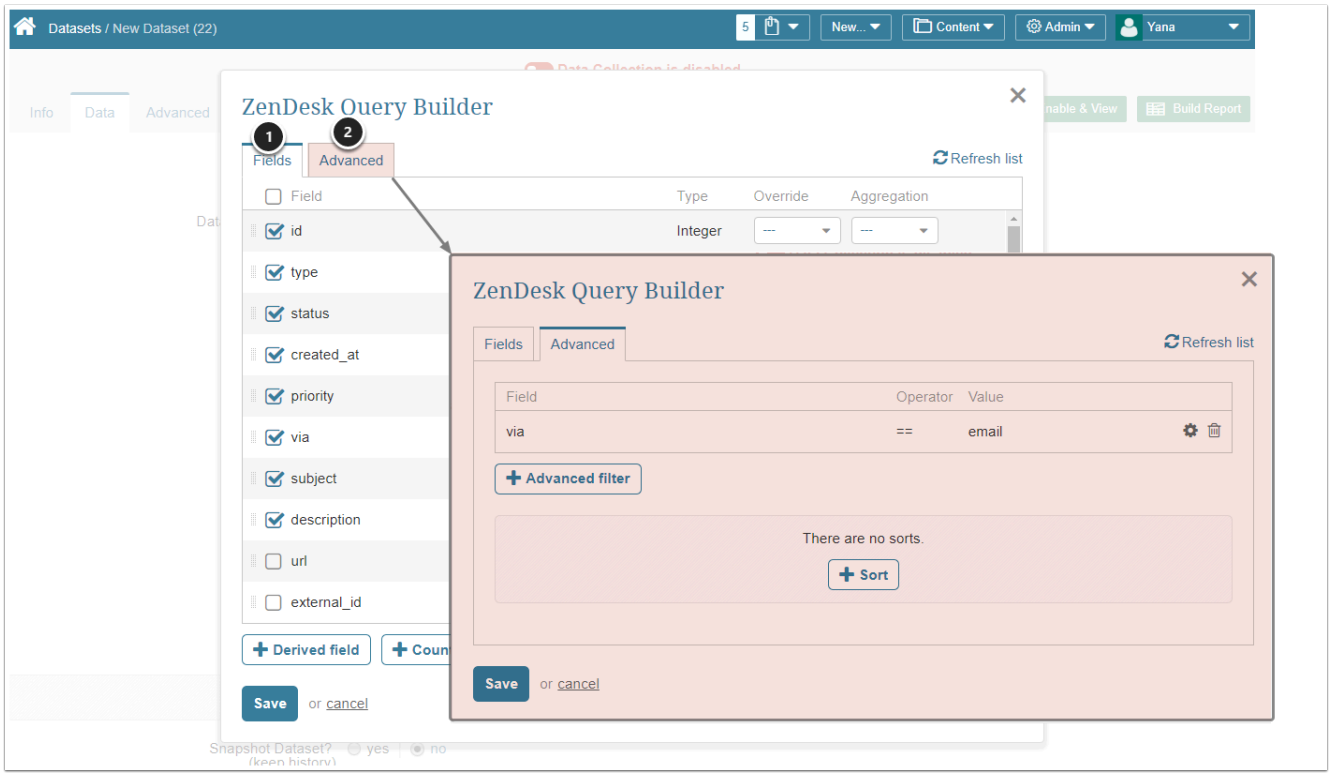
3. Define the Settings for Data Collection



1. **Data Source**: select the connection profile you have created for Zendesk
2. **Data collection trigger**: specify the Trigger that will be used to collect data for your Dataset
3. **View:** select a Zendesk View that should serve as a basis of your Dataset
4. Input a **Plugin Command** listing all the data you would like to fetch from *Zendesk*
   - Build your query in MIQL syntax
   - Alternatively, use the **Visual Editor**

5. Once you are ready with you command, click **Validate**

3.1. Example using the Visual Editor

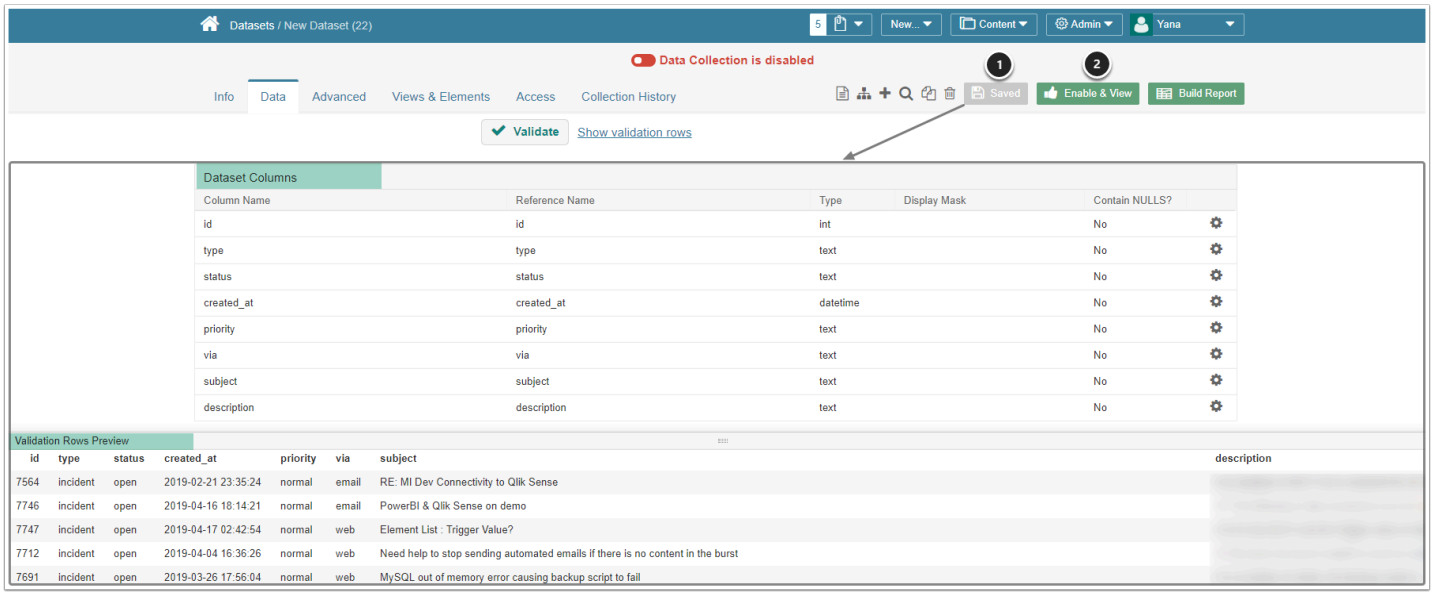


The **Zendesk Query Builder** is called by clicking the Visual Editor button

1. In the **Fields** tab, select the required *fields* for your Dataset
2. In the **Advanced** tab, add *Filters* and *Sorts* as desired

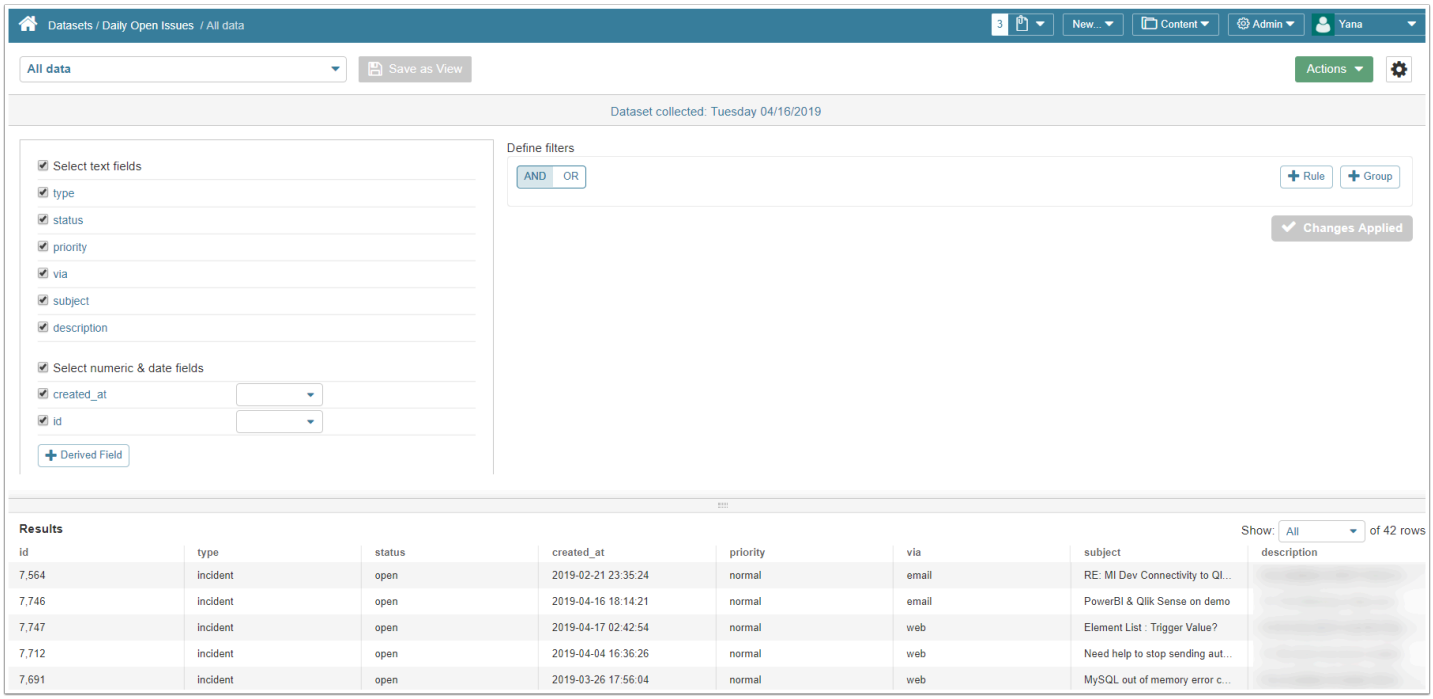
Save your entries

4. Plugin command will be validated and data collected on Save



1. If the command is validated successfully, the **Dataset columns** and **Data Preview** are going to be shown below
2. At the upper right corner of the screen click **Enable & View**

5. Dataset will be displayed in Viewer



What's next?

Create a Dataset Report