Using the Dremio Connector for Tableau

October 20th, 2020

GREG PALMER

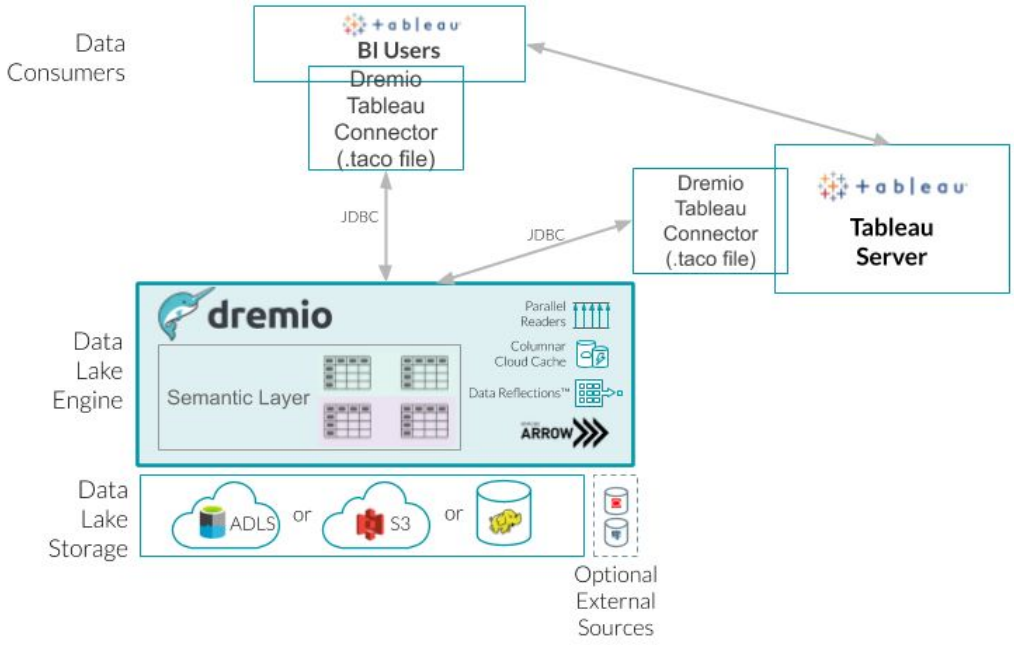
In the past, Tableau users utilized the Dremio ODBC driver to connect to the Dremio Data Lake Engine from their Tableau workbooks using the "Other Database" option. Now Dremio offers a customized connector for Tableau that is available from the Tableau Extension Gallery. This new connector was created using the Tableau Connector SDK and offers better performance and more features over the basic driver, including:

- Better live query support. Allows a customized SQL dialect to generate queries that are optimized for Dremio.
- Simpler connection experience. Users don't need to enter obscure JDBC URL strings or create a DSN or configure odbc.ini files. The connector provides a simple Dremio connection dialog.
- Runs in Tableau Desktop and Tableau Server. No configuration is required once you install the connector.
- Enables the use of Tableau's "Impersonate using embedded password" feature combined with Dremio's "Inbound Impersonation" feature.

To use the Dremio Tableau Connector, it must be deployed on each users' Tableau Desktop computer and on the Tableau Server computer. When a user creates a new Tableau workbook using Tableau Desktop, they can specify the Dremio Connector instead of using the "Other Database" driver. As a user publishes a workbook to the Tableau Server, that same Dremio Connector will be used to connect to Dremio when the workbook is opened.

Dremio Connector for Tableau - Architecture

The Dremio Connector for Tableau architecture is illustrated in the following diagram.

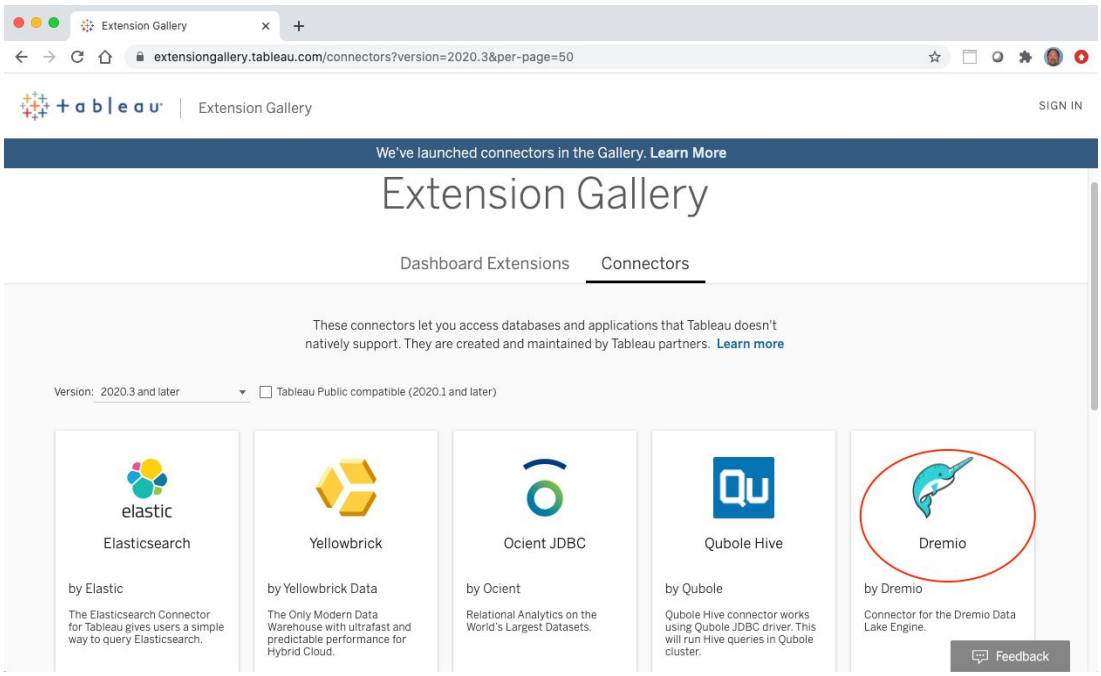


The steps required to install and use the new Dremio Connector for Tableau include the following:

1. Install the Dremio Connector on Tableau Desktop
2. Install the Dremio Connector on Tableau Server
3. Configure the Dremio “Inbound Impersonation” capability and .tds file generation
4. Create a Tableau data source using the Dremio Connector
5. Create and publish a Tableau workbook using Tableau’s “Impersonate using embedded password” configuration
6. Run a Tableau workbook in Tableau Server as a different user using Dremio’s “Inbound Impersonation” configuration

1. Install the Connector on Tableau Desktop

Download the Dremio Tableau Connector from the Tableau Extension Gallery at https://extensiongallery.tableau.com/connectors. Click on the Dremio connector and then click on the blue Download button.



Sign in with your Tableau username and password (or create an account) and download the `dremio.taco` file to your Web browser download directory.

Move the `dremio.taco` file to your Tableau Desktop connectors directory. If the directory does not exist, create it.

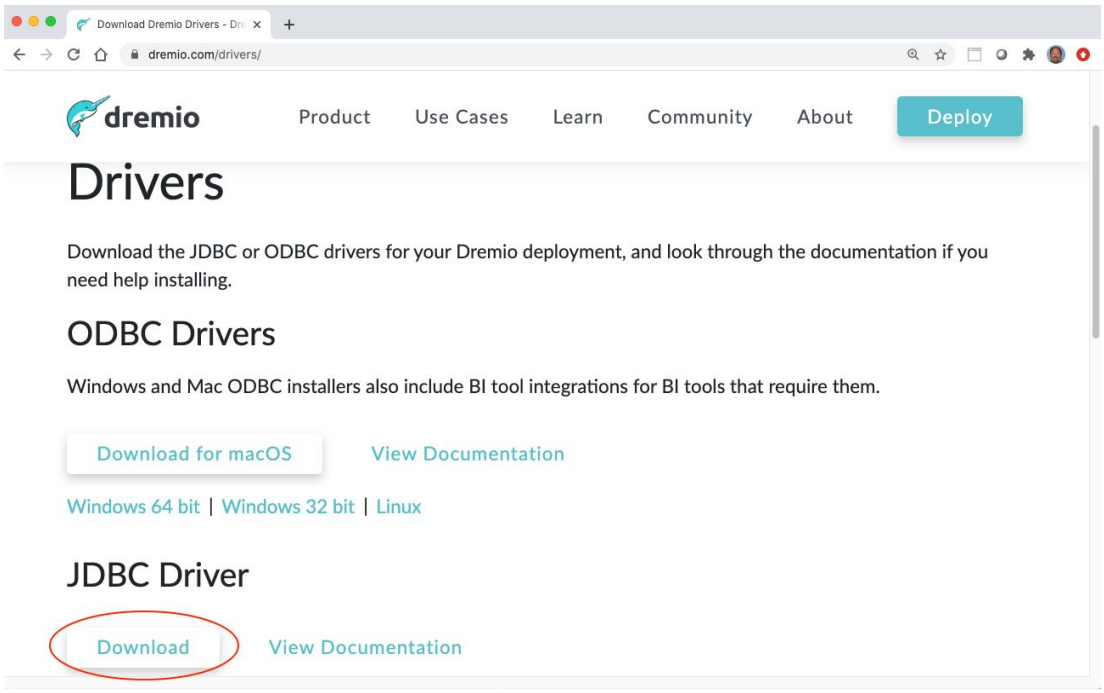
On Windows, move the file to:

```
\Users\[Windows User]\Documents\My Tableau Repository\Connectors
```

On macOS, move the file to:

```
/Users/[user]/Documents/My Tableau Repository/Connectors
```

Now download the Dremio JDBC driver from the Dremio website at https://www.dremio.com/drivers.



Click on the Download link and download the `dremio-jdbc-driver-{version}.jar` file to your Web browser's download directory.

Move the `.jar` file to your Tableau Desktop drivers directory. If the directory does not exist, create it.

On Windows, move the file to:

```
C:\Program Files\Tableau\Drivers
```

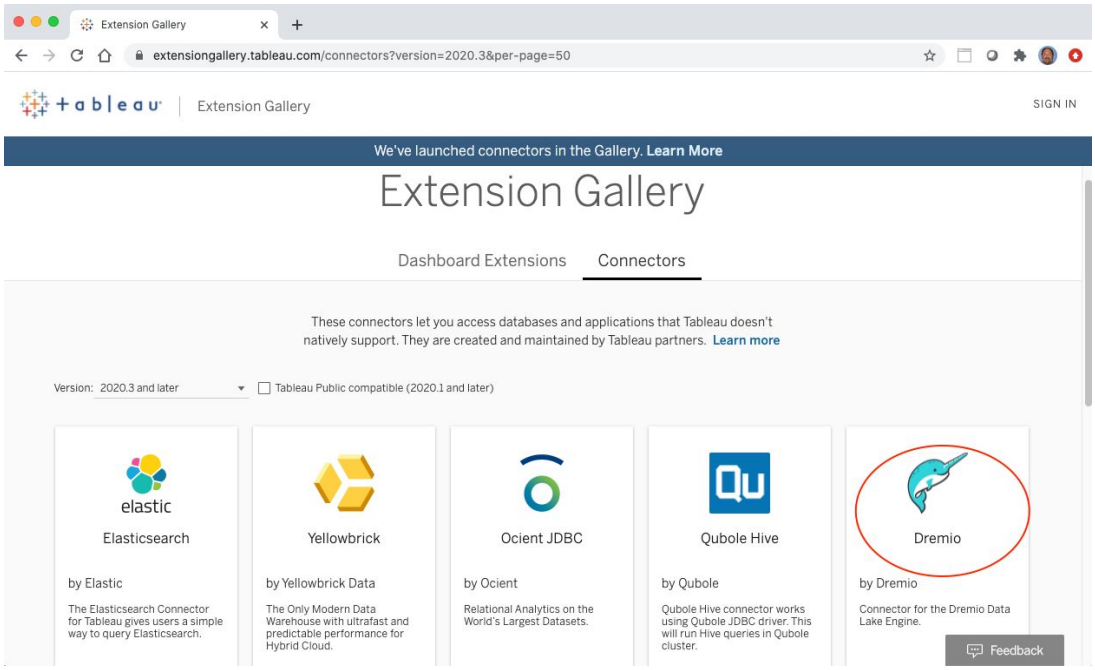
On macOS, move the file to:

```
/Users/[user]/Library/Tableau/Drivers
```

Finally, restart your Tableau Desktop application.

2. Install the Connector on Tableau Server

Download the Dremio Tableau Connector from the Tableau Extension Gallery at https://extensiongallery.tableau.com/connectors. Click on the Dremio connector and then click on the blue **Download** button.



Sign in with your Tableau username and password (or create an account) and download the `dremio.taco` file to your Web browser download directory.

Move the `dremio.taco` file to your Tableau Server connectors directory. If the directory does not exist, create it.

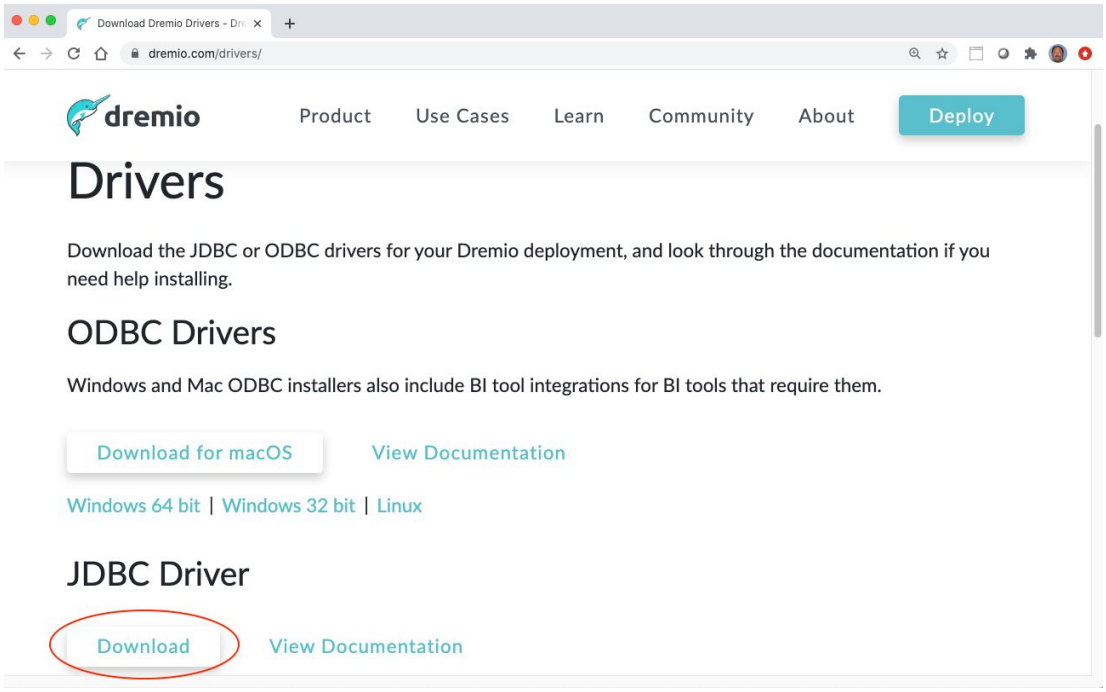
Move the `.taco` file to:

```
[Tableau_Server_Install_Dir]\data\tabsvc\vizqlserver\Connectors
```

In most cases, the Tableau Server is installed in the Program Data directory, so you would copy the .taco file to:

```
C:\ProgramData\Tableau\Tableau Server\
data\tabsvc\vizqlserver\Connectors
```

Now download the Dremio JDBC driver from the Dremio website at https://www.dremio.com/drivers.



Click on the Download link and download the `dremio-jdbc-driver-{version}.jar` file to your Web browser's download directory.

Move the `.jar` file to your Tableau Server drivers directory. If the directory does not exist, create it. Move the file to:

```
C:\Program Files\Tableau\Drivers
```

Note: If you are running Tableau Server 2030.3+ that uses the Java 11 runtime for JDBC drivers, you must add an environment variable to enable Dremio to function properly. Follow these steps:

- Log in as the user that runs the Tableau Server
- From Windows Server desktop, go to `Control Panel > System Security > System > Advanced` system settings
- Click the **Environment Variables** button in the window that pops up
- Add a new variable under `User variables for <user that runs Tableau>`
- Create the new variable name:
  `JAVA_TOOL_OPTIONS`
- Set the new variable value to:
  `-Dcdjd.io.netty.tryReflectionSetAccessible=true`

Finally, restart your Tableau Server Windows system service.

3. Configure the Dremio "Inbound Impersonation" capability and .tds file generation

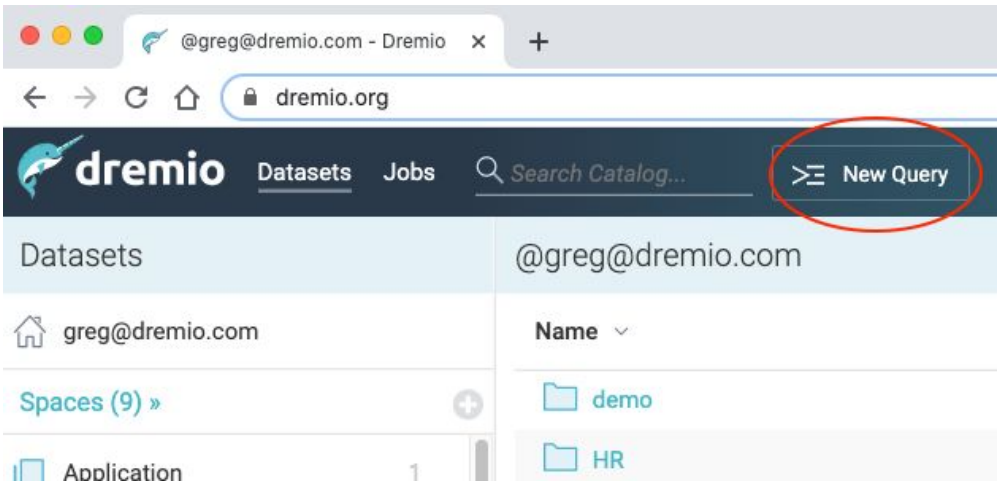
Inbound Impersonation

Dremio supports the concept of inbound impersonation, where the end-user connects to Dremio using a proxy user. A proxy user is a user that is allowed to

"connect on behalf of another user". In this case, Dremio runs a query as the real user that submitted the request (and verifies any access permissions needed).

To configure Dremio to enable proxy users that submit queries as another user, a Dremio Administrator must create an inbound policy via a Dremio support key named `exec.impersonation.inbound_policies`. Follow these steps:

- Log into the Dremio UI as an Administrator user
- Launch a New Query session by clicking on the **New Query** button:



- In the query editor window, type in an `ALTER SYSTEM` command with the following format:

```
ALTER SYSTEM SET "exec.impersonation.inbound_policies"='[
{
    proxy_principals:{
        users:["<proxy-user-1>"]
    },
    target_principals: {
        users:["<target-user-1>",
               "<target-user-2>"]
    }
},
{
    proxy_principals:{
        users:["<proxy-user-2>"]
    },
    target_principals: {
        users:["<target-user-2>,
               "<target-user-3>",
               "<target-user-4>"]
    }
}
]'
```

Where the proxy user `proxy-user-1` can submit Dremio queries for the real users `target-user-1` and `target-user-2`. And where the proxy user `proxy-user-2` can submit Dremio queries for the real users `target-user-2, target-user-3,` and `target-user-4`.

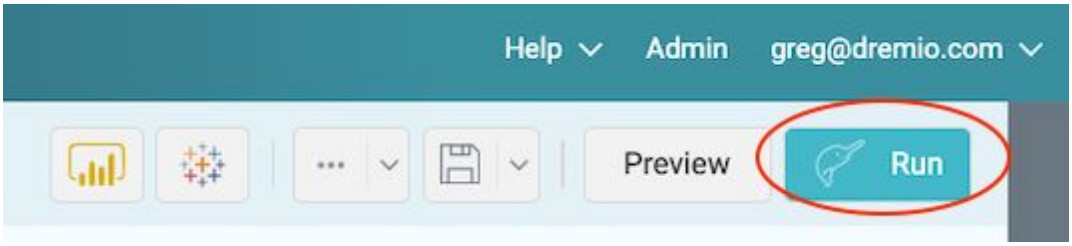
You can also specify group names in the `inbound_policies` configuration like this:

```
ALTER SYSTEM SET "exec.impersonation.inbound_policies"='[
{
    proxy_principals:{
        groups:["<proxy_group-1>"]
    },
    target_principals: {
        groups:["<target-group-1>",
               "<target-group-2>"]
    }
},
{
    proxy_principals:{
        groups:["<proxy-group-2>"]
    },
    target_principals: {
        groups:["<proxy-group-2>,
               "<proxy-group-3>",
               "<proxy-group-4>"]
    }
}
]'
```

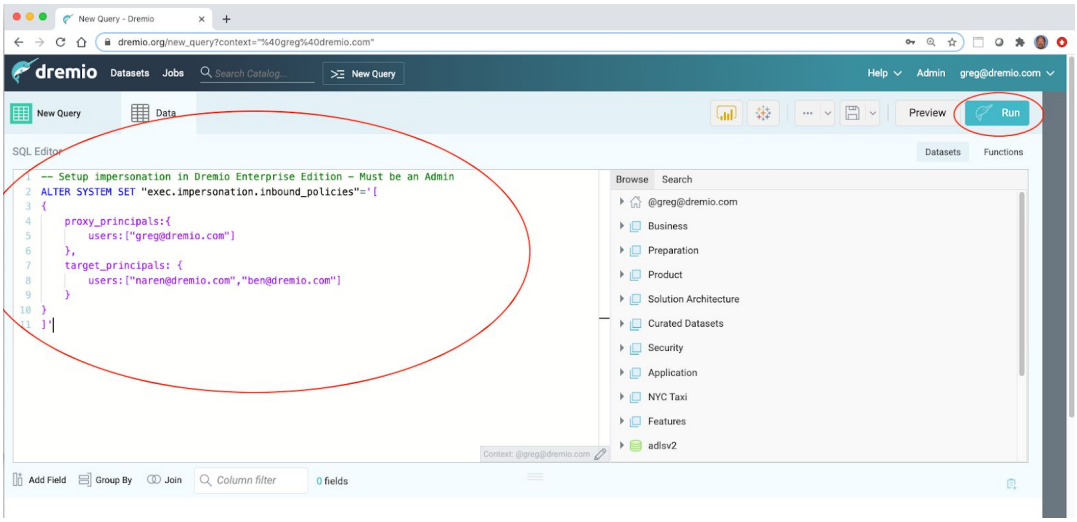
Where the proxy group `proxy-group-1` can submit Dremio queries for the users in groups `target-group-1` and `target-group-2`. And where the proxy group `proxy-group-2` can submit Dremio queries for the users in groups `target-group-2, target-group-3,` and `target-group-4`.

You can also combine the `users` and `groups` specifications in the same `inbound_policies` configuration.

- Click on the **Run** button to run the `ALTER SYSTEM` command:



A real example might look like this:



- After running the `ALTER SYSTEM` command, you can verify that it was effective by running the command:

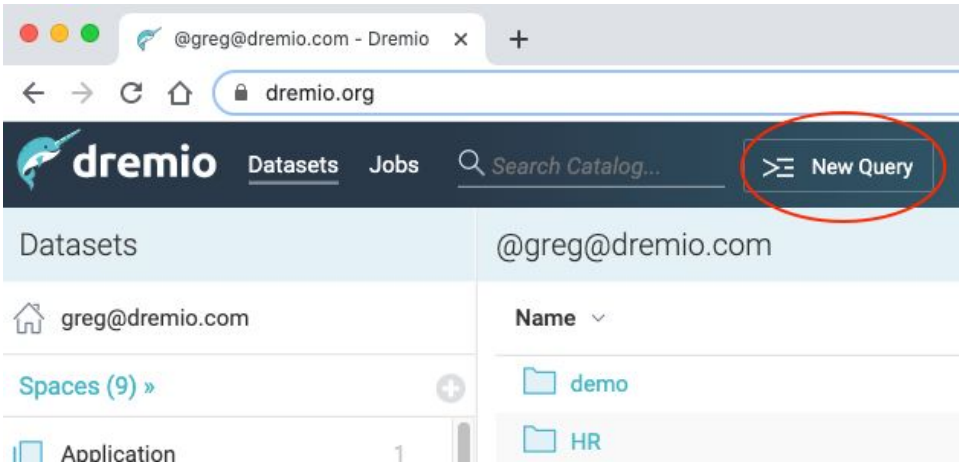
```
-- Verify the ALTER SYSTEM command
SELECT name, string_val
FROM   sys.options
WHERE  name = 'exec.impersonation.inbound_policies'
```

Tableau .tds File Generation

The Dremio UI allows users to quickly launch a Tableau Desktop session by generating a `.tds` file that contains the connection information to the Dremio server. This .tds file gets downloaded to the user's web browser download directory and then can be "launched" and create a new Tableau workbook.

By default, Dremio configures the generated `.tds` file with a generic ODBC connection specification (`connection class="genericodbc"`). To enable the Dremio Connector for Tableau (`connection class="dremio"`), a Dremio Administrator must configure a "Support Key" named `export.tableau.export-type`. Follow these steps:

- Log into the Dremio UI as an Administrator user
- Launch a New Query session by clicking on the **New Query** button:

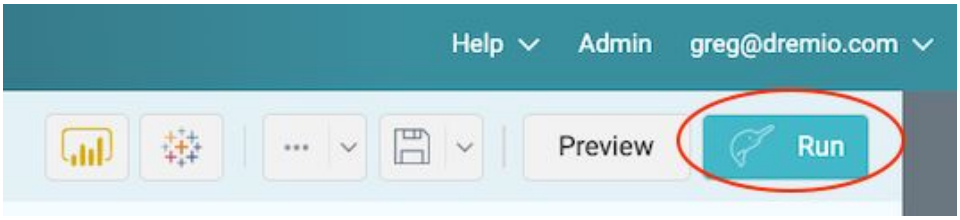


- In the query editor window, type in an ALTER SYSTEM command with the following format:

```
-- Configure Dremio to use the Dremio Connector
-- instead of generic ODBC in .tds files

ALTER SYSTEM
   SET "export.tableau.export-type" = 'NATIVE'
```

- Click on the **Run** button to run the `ALTER SYSTEM` command:



- After running the `ALTER SYSTEM` command, you can verify that it was effective by running the command:

```
-- Verify the ALTER SYSTEM command
SELECT name, string_val
FROM   sys.options
WHERE  name = 'export.tableau.export-type'
```

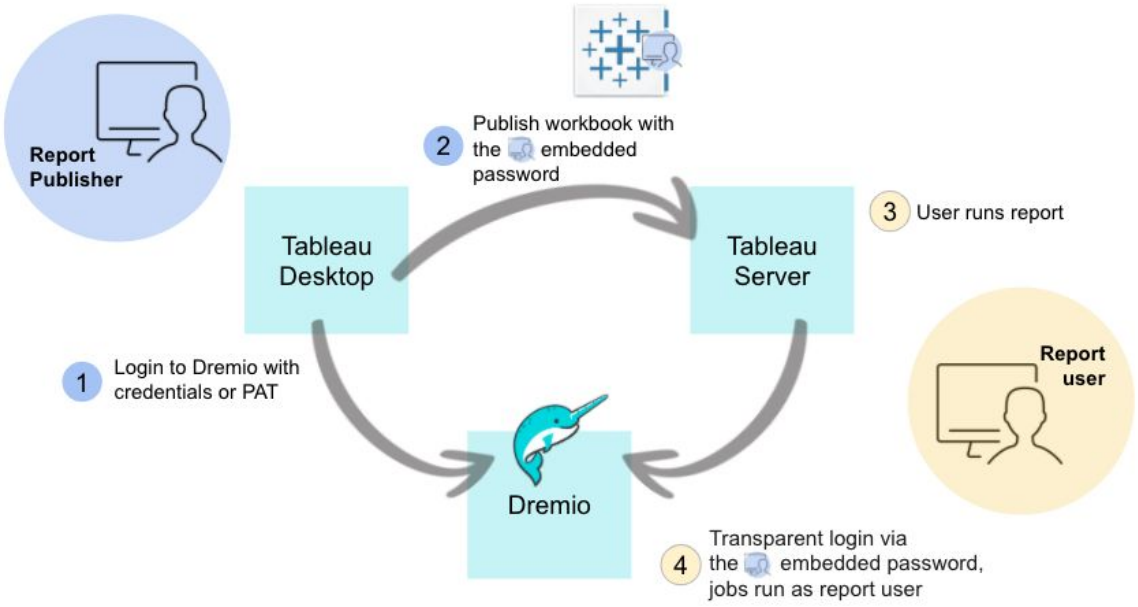
Finally, restart your Dremio server by launching an SSH session into the Dremio Coordinator Node server and running the command:

```
sudo systemctl restart dremio
```

4. Create a Tableau data source using the Dremio Connector

Now that the Dremio Connector for Tableau has been installed on the Tableau Desktop computer and the Tableau Server computer, a Tableau data source can be created that uses the new connector.

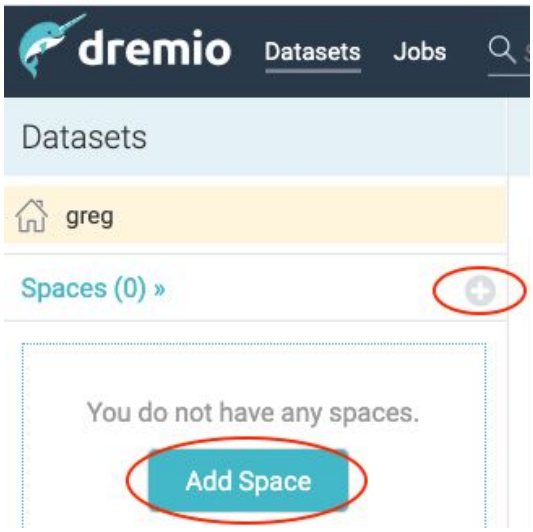
Usage Overview



- In Step 1, a report publisher user launches Tableau Desktop and connects to Dremio using their username and password (or Personal Access Token). The user configures a Tableau data source that points to a Dremio server and Virtual Datasets (VDSs).
- In Step 2., the user creates a Tableau workbook and publishes the workbook to Tableau Server using the "Impersonate via embedded password" data source option.
- In Step 3, a report user signs into the Tableau Server.
- In Step 4, the report user opens the previously published Tableau workbook and Tableau connects to the Dremio server (without prompting the report user for a username and password) and runs the Dremio queries as the report user, not as the publisher user.

Let's start by launching Tableau from the Dremio UI.

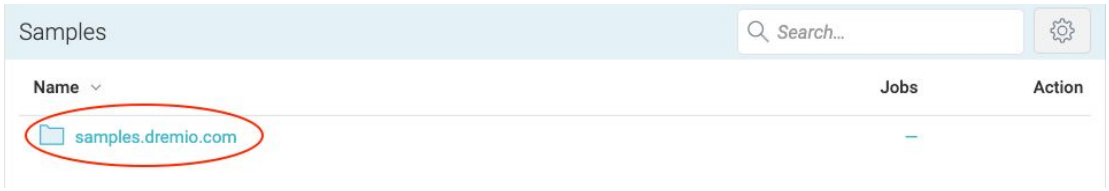
Log into the Dremio Web UI as either a non-admin user or an admin user. If you don't have a Dremio space named `Preparation`, create one by clicking on the Add Space button or the plus sign icon to the right of the Spaces.



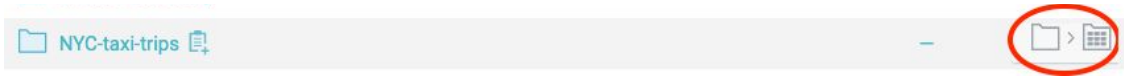
In the **Add Space** screen, enter the name `Preparation` and click the **Save** button.

Now create a new Physical Dataset from the Dremio Samples dataset. Click on the **Add Sample Source** button.

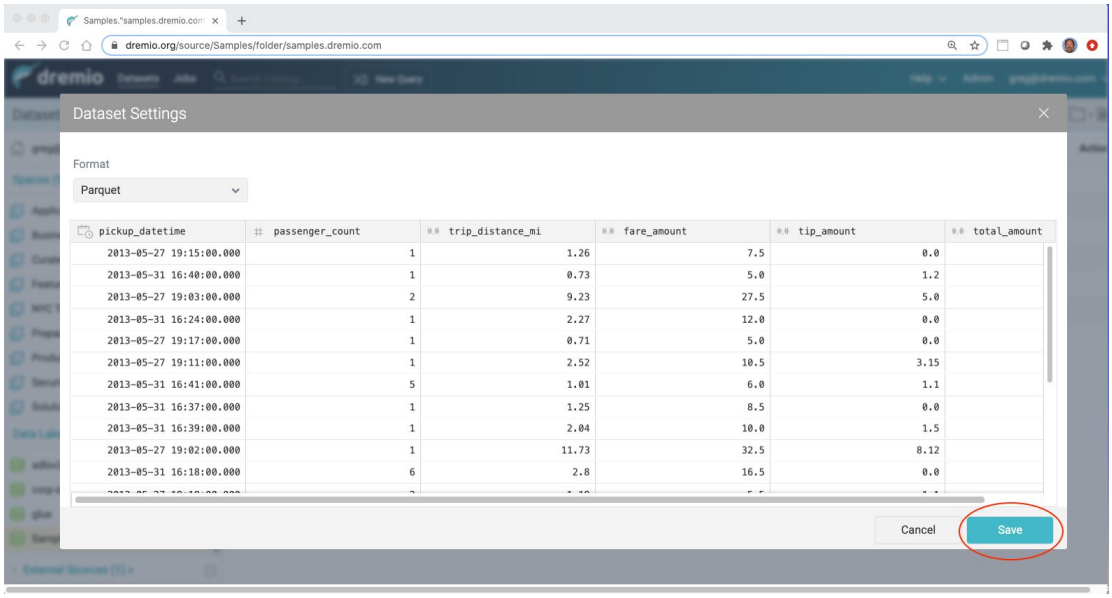
Then click on the sample folder that is generated.



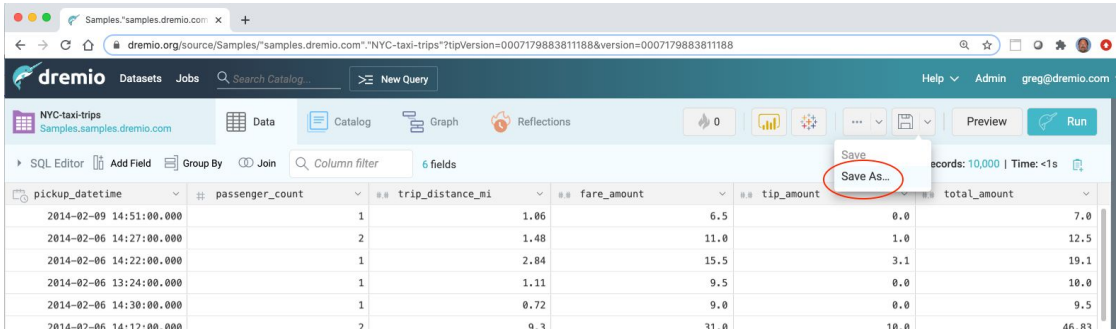
Inside the folder, click on the **Format Folder** icon in the **Actions** for the `NYC-Taxi-Trips` dataset.



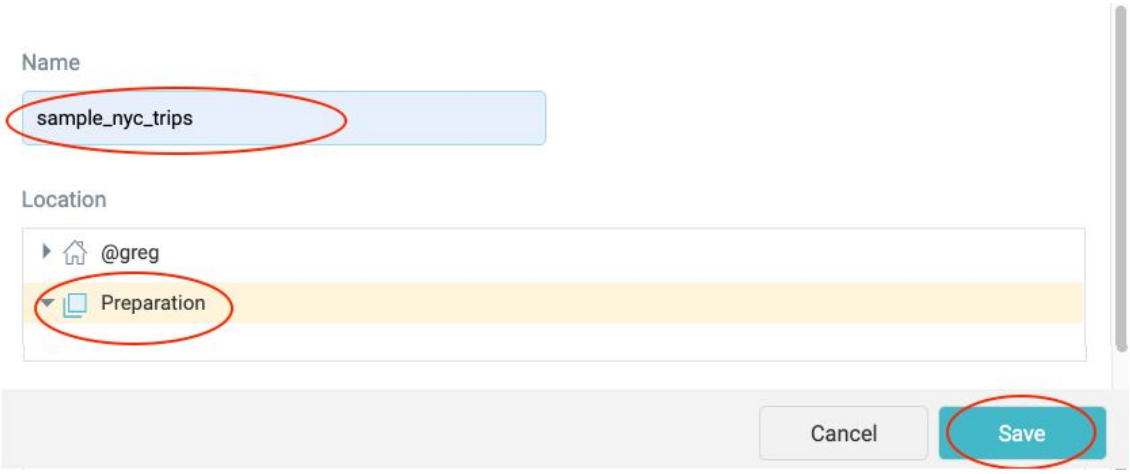
The sample data is stored using the Parquet file format. In the **Dataset Settings** screen click the **Save** button to save this default formatting.



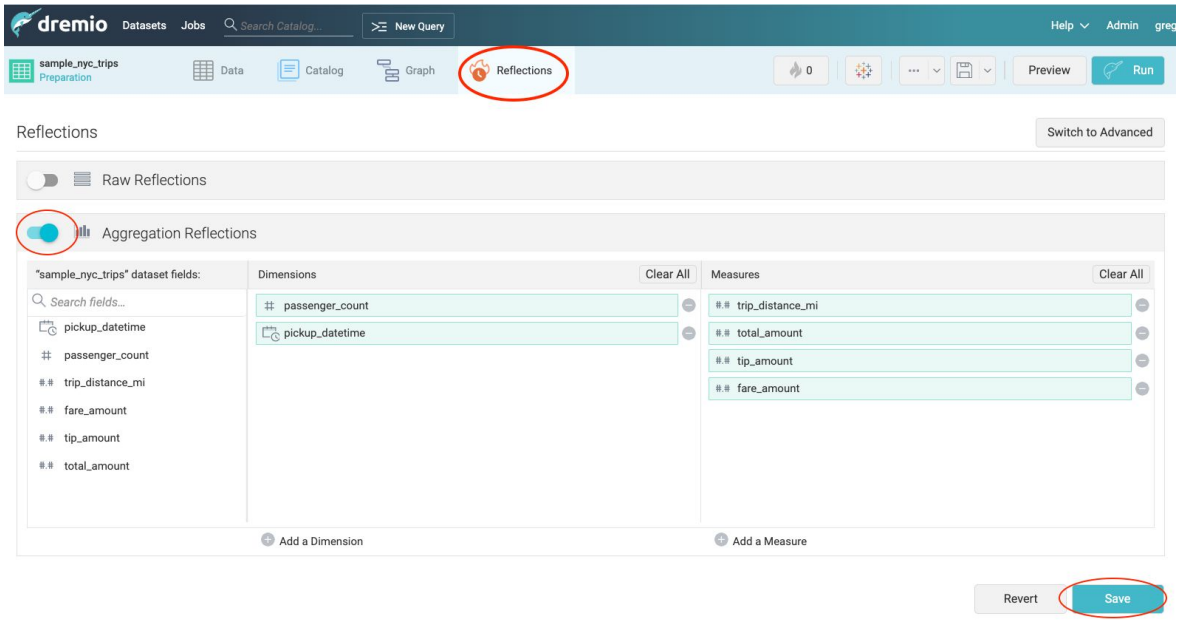
Then click on the **Save As** button to save this as a new VDS.



In the **Save Dataset As** screen, select the `Preparation` space and enter the name `sample_nyc_trips` and click on the **Save** button.

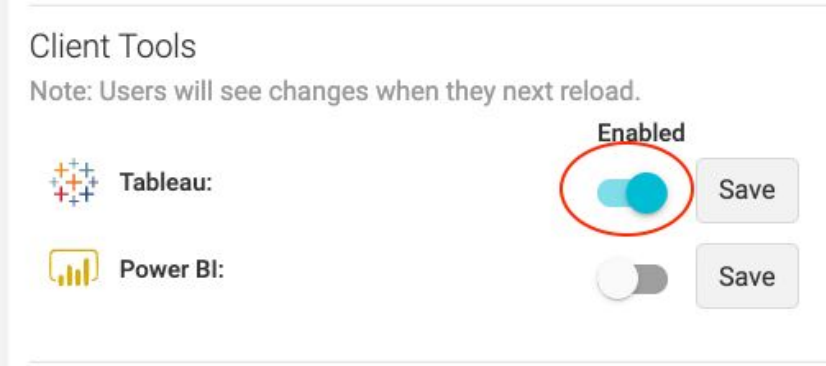


Speed up query performance against this dataset by creating a new Dremio Aggregation Reflection. Click on the **Reflections** button and enable the default Aggregation Reflection. Then click the **Save** button to begin generating this new reflection.

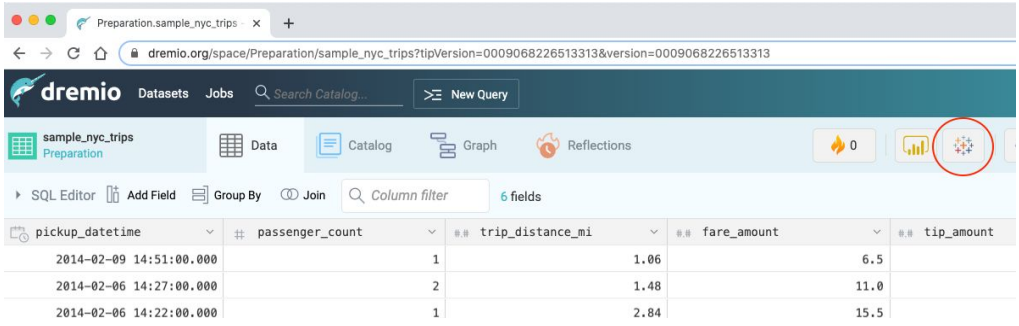


Now launch Tableau Desktop using the .tds file generated by Dremio. In the `sample_nyc_trips` dataset preview screen, click on the **Tableau** button.

The next step requires the use of the **Tableau** launch button. If you do not see a **Tableau** button, you can enable it as a Dremio Administrator user by clicking on the **Admin** link at the top right of the page and then click on the **Support** link on the left side of the page. In the **Client Tools** section, click on the **Tableau** toggle button and then click the **Save** button.



Now launch Tableau Desktop using the .tds file generated by Dremio. In the `sample_nyc_trips` dataset preview screen, click on the **Tableau** button.



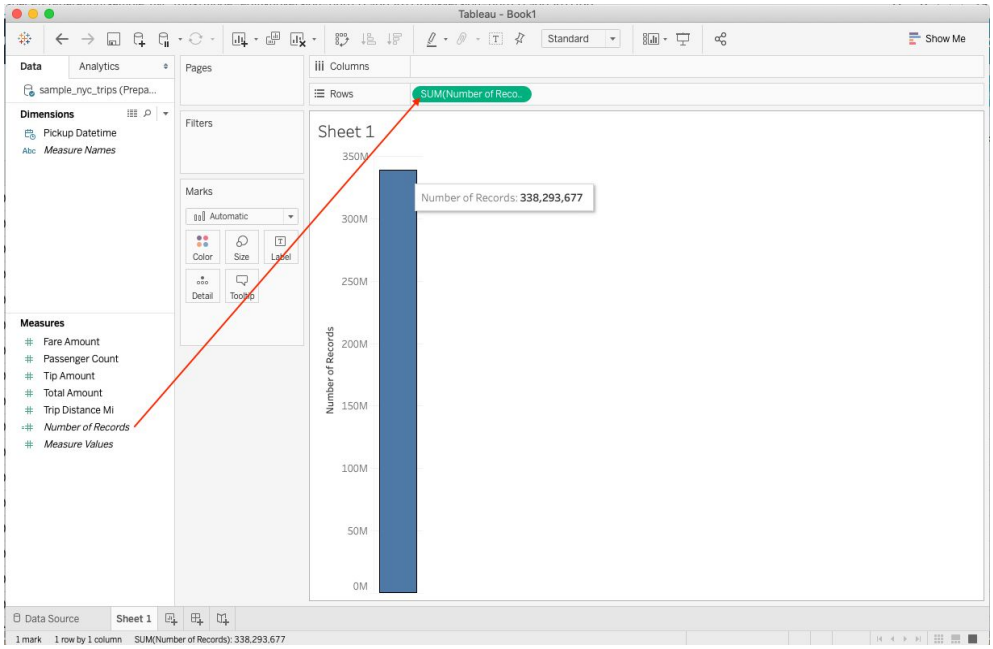
This will download a .tds file in your Web browser's download directory. At the bottom of your Web browser page, click on the downloaded file and **Open** it.

5. Create and publish a Tableau workbook using Tableau's "Impersonate using embedded password" configuration

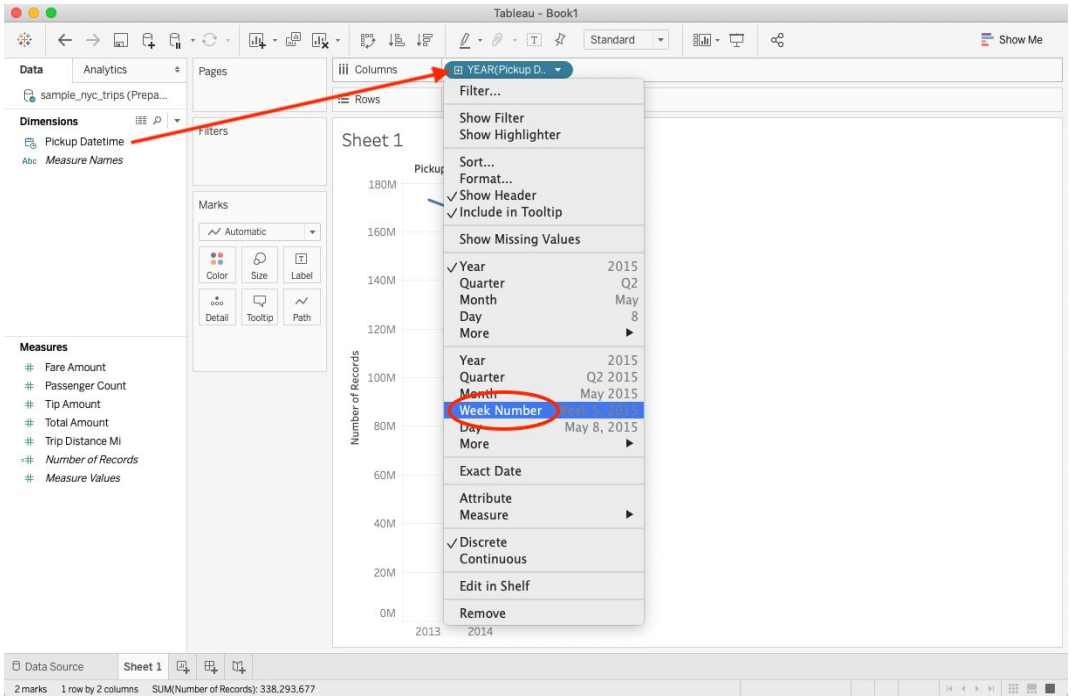
When the Tableau workbook is opened, it will prompt you for your Dremio login credentials. Enter your username and password. The Tableau workbook will

connect to Dremio and access the `Preparation.sample_nyc_trips` dataset and display the Dimensions and Measures associated with the dataset.

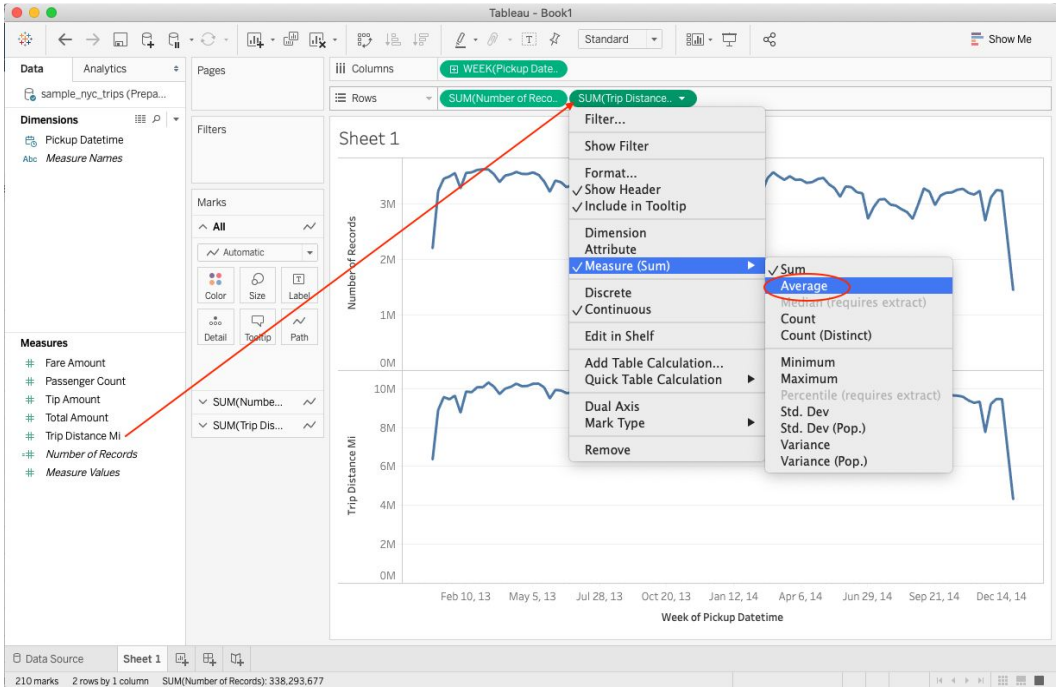
Add the number of taxi rides to the worksheet by dragging the `Number of Records` measure to the **Rows** section of the worksheet and you can see that there are over 1 billion records in this Dremio dataset.



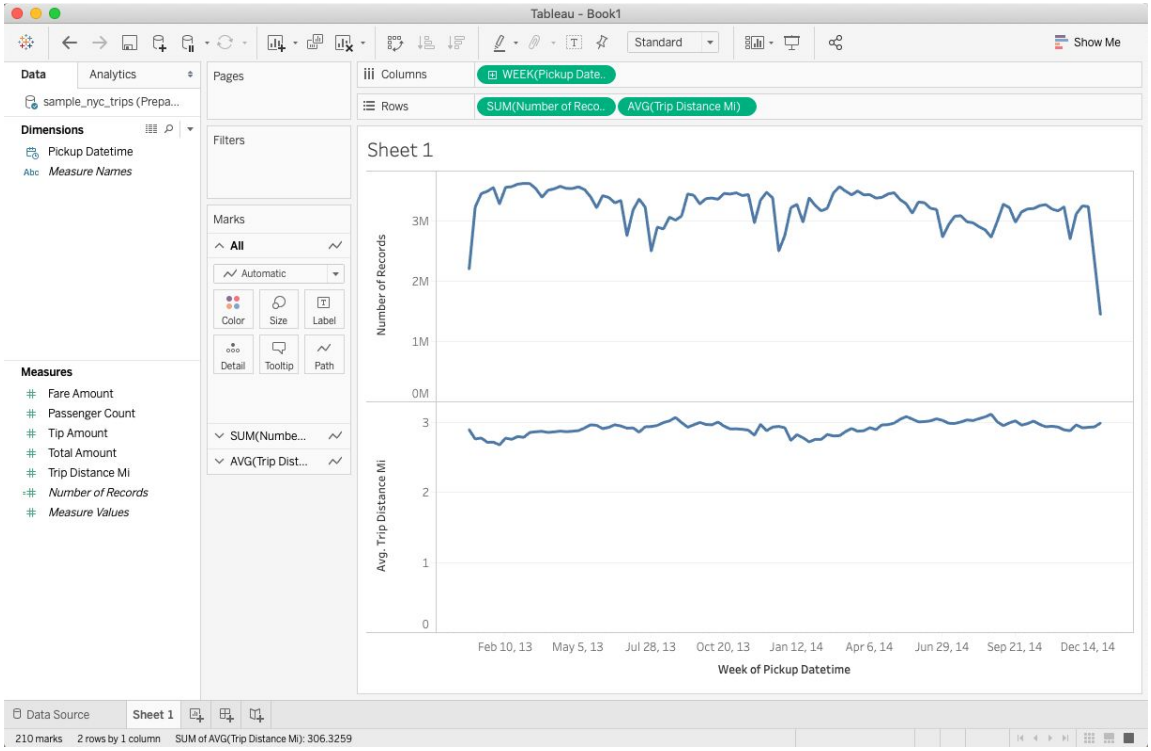
Next, add a graph of the number of taxi rides per week distributed across the years in the dataset. Drag the `Pickup Datetime` dimension to the **Columns** section of the worksheet, then select the **Week Number** group specification.



Now add a graph of the average trip distances by dragging the `Trip distance Mi` measure to the **Rows** section of the worksheet.



The final result will be a Tableau worksheet that plots the weekly taxi ride counts and the average trip distance of those taxi rides, from February 2013 through December 2014.

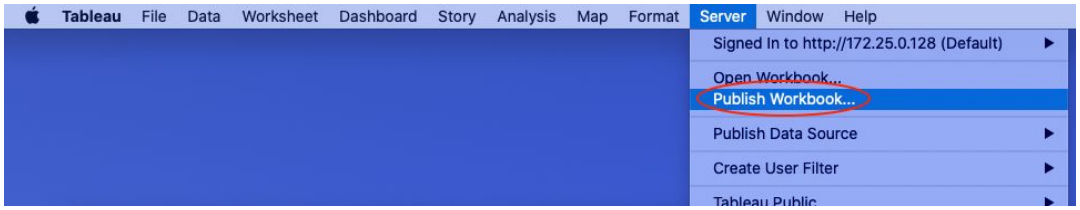


Now you are ready to publish the Tableau workbook into Tableau Server using the Tableau "Impersonate with embedded password" option.

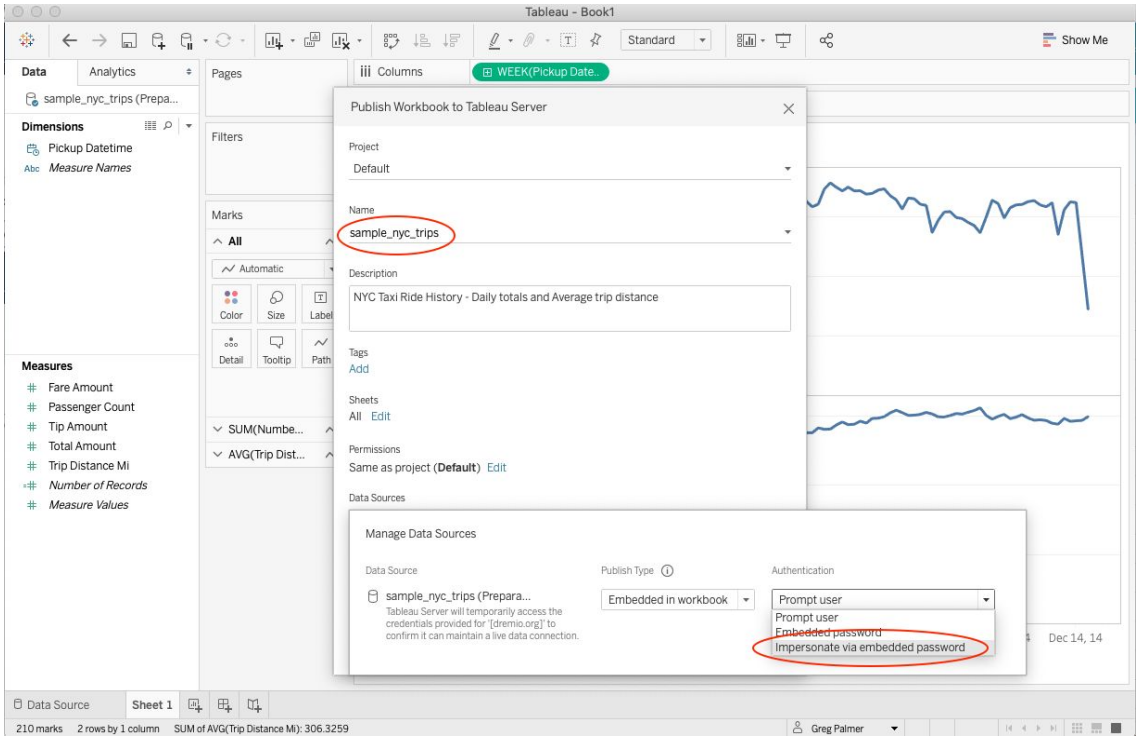
First, sign in to the Tableau Server. In the Tableau Desktop menu, select **Server > Sign in** and enter your Tableau Server address, then click on the **Connect** button.

Enter your Tableau Server username and password and click on the **Sign In** button.

Now, publish your workbook by choosing the **Server > Publish Workbook** menu option from the Tableau Desktop menu.

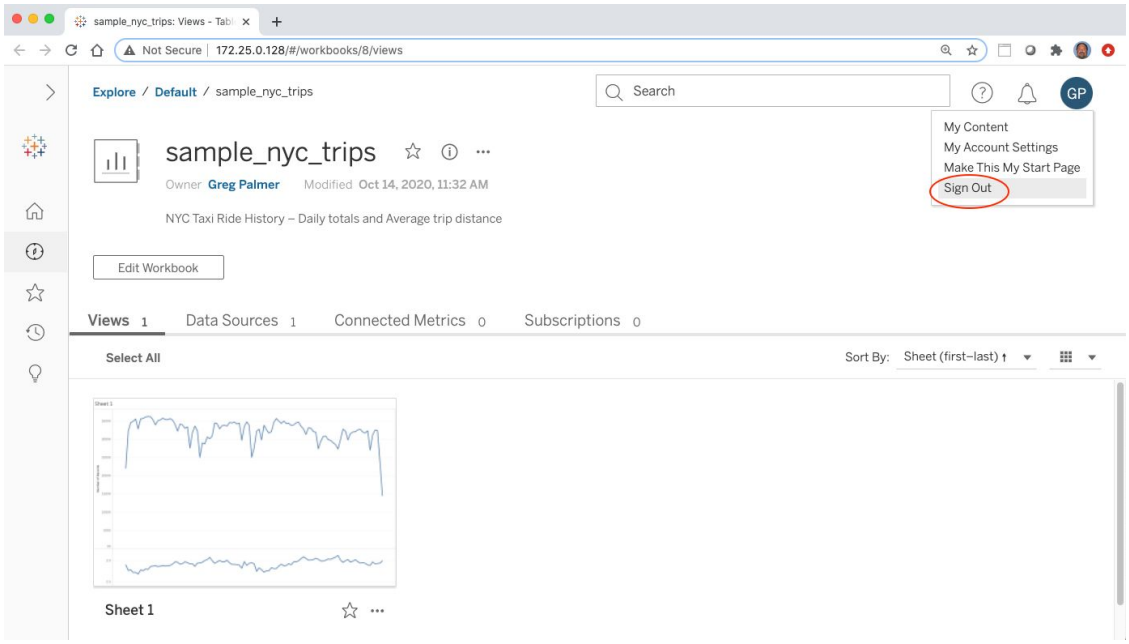


In the **Publish Workbook to Tableau Server** screen, enter a name for the workbook and click on the **Edit** link next to the data sources.



In the **Manage Data Sources** section, select the Authentication method **Impersonate via embedded password**. Then click on the **Publish** button. Tableau Desktop will open a Web browser window and display the published workbook.

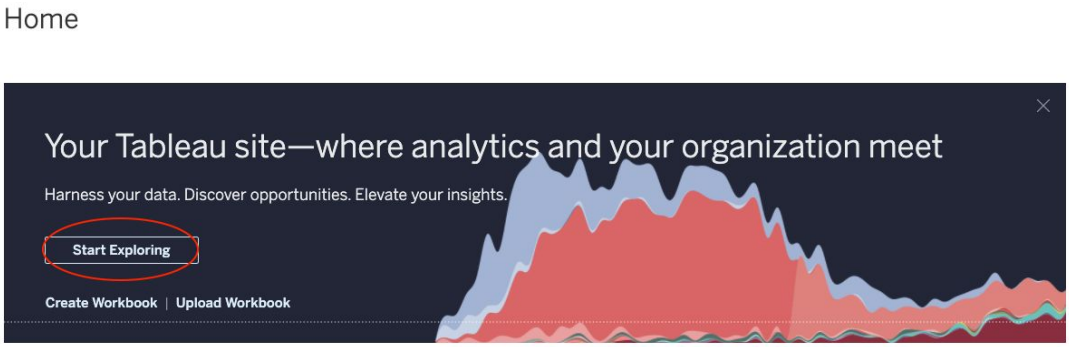
Click the **Done** button and then sign out from the Tableau Server.



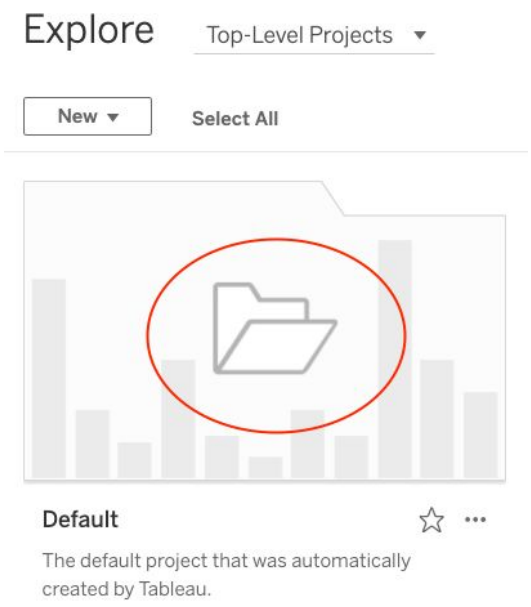
6. Run a Tableau workbook in Tableau Server as a different user using Dremio's "Inbound Impersonation" configuration

Now you will sign in to Tableau Server as a different user from the one in which you published the workbook. Note that this user must also have access to Dremio via the Dremio inbound impersonation capability (see Step 3. above).

Once you have signed in to Tableau Server, click on the **Start Exploring** button on the Home page.

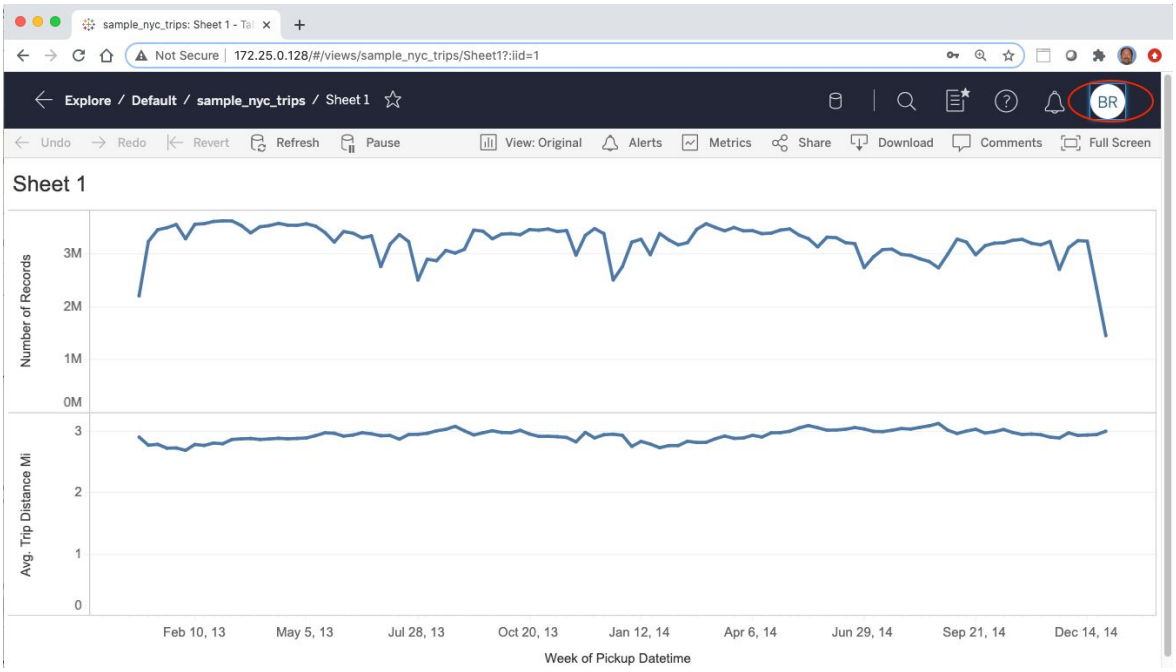


Then open the project that you published your workbook in.

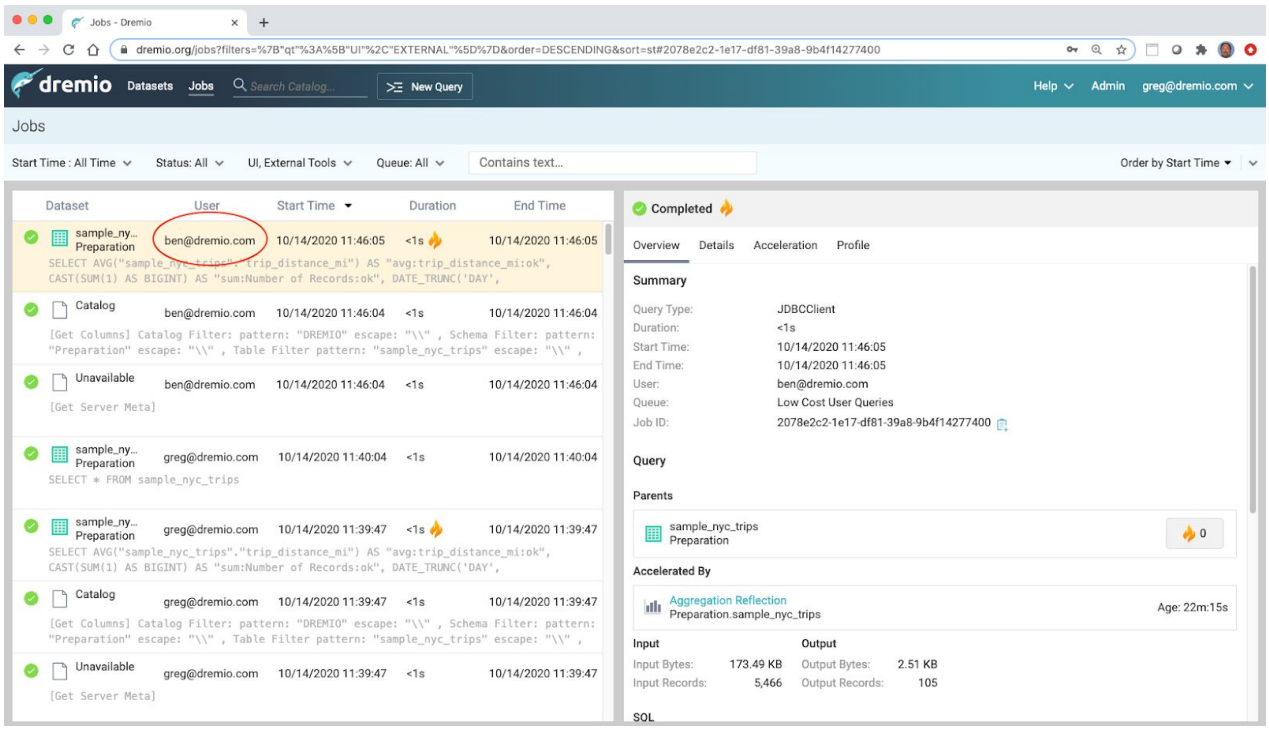


In the project folder, you will see the recently published workbook. To open the workbook, click on the `sample_nyc_trips` workbook and then double-click on the view for Sheet 1.

The workbook will be opened and you will see the same graphs that you saw when using Tableau Desktop, the number of taxi rides by week, and the average trip distance.



Now, view the Dremio Jobs screen and notice that the query sent to Dremio by the Tableau Server based workbook was submitted as your current user and not the proxy user that published the workbook. And you were not prompted for a username or password to connect to Dremio. This is because Tableau already knows who you are and that the workbook data source was set up to authenticate with the “Impersonate via embedded password” option. Additionally, Dremio was configured to allow the proxy user (greg@dremio.com) to submit a query as the real user (ben@dremio.com).



Summary

This document discussed why you would use the Dremio Connector for Tableau and how to set it up on the desktop as well as on the server. It also walked you through the steps needed to create a Tableau workbook that utilizes the connector.

For further reading and for reference, here are some useful links:

http://docs.dremio.com/client-applications/tableau-sdk.html

https://extensiongallery.tableau.com/connectors

https://tableau.github.io/connector-plugin-sdk/#