

Grafana QuickStart Guide

By: Jackson Ballew | Spring '24

Docker and Getting Grafana Running

There are at least two ways to run Grafana:

- 1) Grafana Cloud is a web based option that I think allows multiple users to work together and share dashboards easily.
- 2) Grafana OpenSource/Enterprise is a local Grafana server that runs in a docker container.

We will be implementing the second option.

Steps:

- 1) **Install Docker.** I recommend going ahead and getting Docker Desktop as it will make it easier to manage the container.
- 2) Once we do this we need to get the distro of grafana we want. This is rather straightforward since GrafanaLabs maintains a Docker image of what we want. Go ahead and **make a folder to house all of your files.**
- 3) We will actually be making the Docker container and installing the grafana image at the same time (Docker is magical like that). The command to do this looks like:

```
docker run -d -p 3000:3000 --name=rps-grafanaOSS -v  
C:\Users\balle\Desktop\grafana:/etc/grafana grafana/grafana-oss:latest
```

- a) -d tells docker to run the container in the background so it doesn't put you into the container's terminal
- b) -p 3000:3000 maps the port on the host machine to that port on the docker container
- c) -name=rps-grafanaOSS sets the name of our Docker container
- d) The -v indicates we are talking about the container volumes (storage). What follows tells Docker how to relate our system's file path to the container's file path. For the example command, I linked my working directory "grafana" with "etc/grafana". You will leave the "etc/grafana" part the same.
 - i) **Note: specifying this volume is what tells Docker where to save our data. Without it, everytime we launch the container we would start from a fresh instance of Grafana. We obviously do not want that so it is very important this is included.**
 - e) The "grafana/grafana-oss:latest" part tells Docker what image we are using. If you don't have that image saved, Docker will download the image.

- 4) **Rewrite the given command and either run it or save it in a batch file and run it.** This only has to happen once. This command, despite it being a “run” command, is what creates the Docker container. If you have Docker Desktop, then to run the server in the future, open the Docker app and click run on the container.
 - a) One thing to figure out is what command needs to run to both launch Docker and to start the grafana instance. Then both of these will need to be added on startup.
- 5) Now you should have a grafana instance up and running. If you **go to “localhost:3000”** you should reach the login page for the server. The default signin is “admin”/“admin”. You will be prompted to create a new admin password on signin. You can create other user accounts on the server to avoid handing out admin but I’m not sure how much that matters for what we are doing.

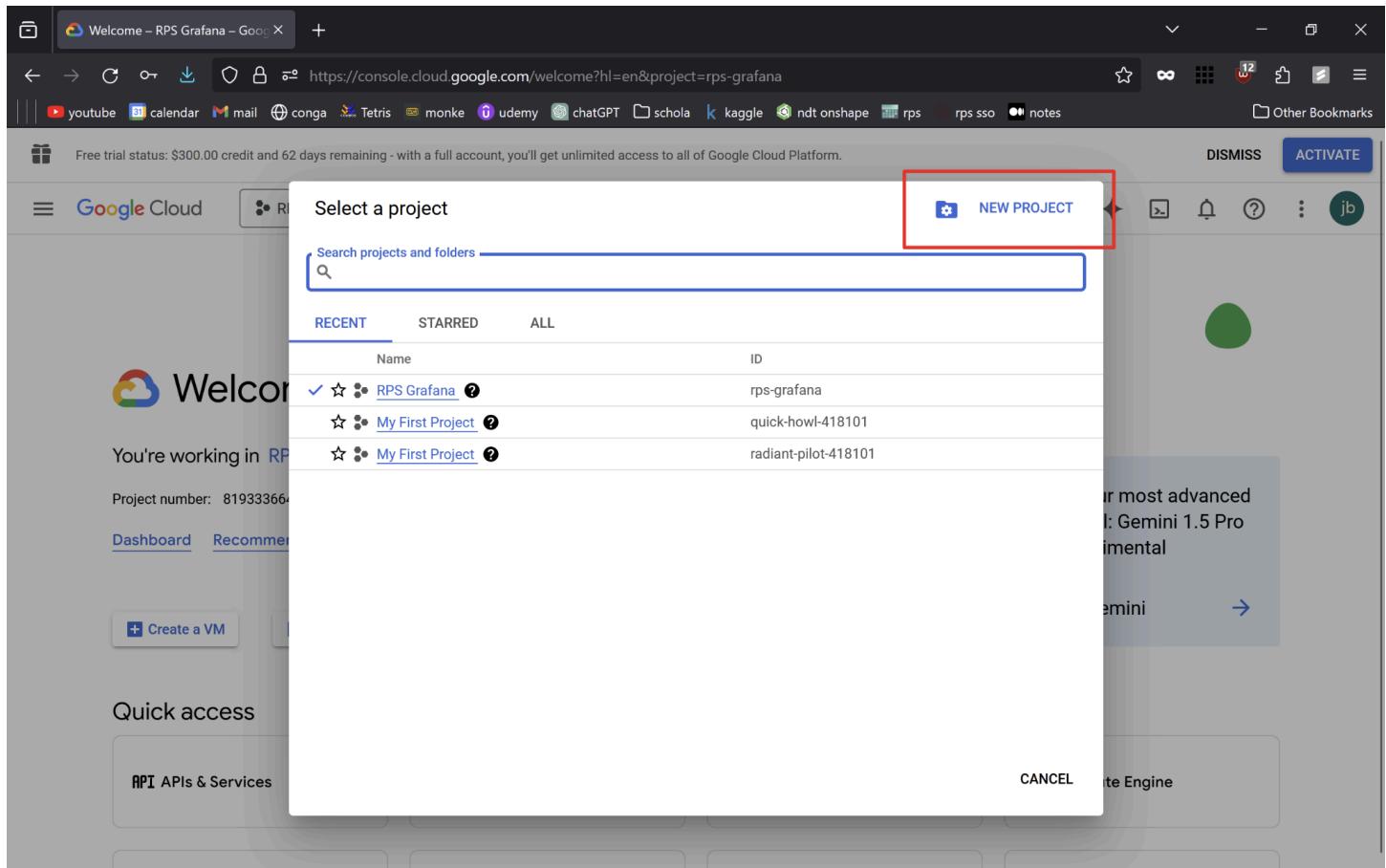
Getting the Google Service Account Set Up

The next step is to get a google service account set with access to the spreadsheet. I did all of my testing with a copy of the real orders spreadsheet. IDK if Alex will have you use a copy again or go ahead and set it up with the real thing. I would encourage you to discuss with Alex what we need to do for this step since Alex should probably have access to this service account if we are setting it up for the real spreadsheet.

Steps:

- 1) Goto [Google Cloud Console](#)
- 2) Set up your account. This will involve adding payment information. None of what we are implementing will require payments.
- 3) Once your account is set up, make a new project:

The screenshot shows the Google Cloud Console interface. At the top, there's a navigation bar with a 'Welcome' tab, a search bar, and various bookmarked links. A red box highlights the 'RPS Grafana' dropdown menu in the top-left corner of the main content area. The main content area displays a 'Welcome' message for the 'RPS Grafana' project, showing its project number (8193366478) and ID (rps-grafana). It features a 'Try Gemini' button and a 'Quick access' section with links to API & Services, IAM & Admin, Billing, and Compute Engine.



There is an organization option once you select a new project. I don't know how we need to go about setting it up with TAMU.

- 4) Once you have the project set up we will navigate to API's and Services:

The screenshot shows the Google Cloud Platform (GCP) console interface. At the top, there's a navigation bar with links like youtube, calendar, mail, conga, Tetris, monke, udemy, chatGPT, schola, kaggle, ndt onshape, rps, rps sso, and notes. Below the navigation bar, a banner displays "Free trial status: \$300.00 credit and 62 days remaining - with a full account, you'll get unlimited access to all of Google Cloud Platform." There are "DISMISS" and "ACTIVATE" buttons.

The main content area has a sidebar on the left with sections for "Cloud overview" and "Products & solutions". Under "PINNED PRODUCTS", there's a list of services: APIs & Services, Billing, IAM & Admin, Marketplace, Compute Engine, Kubernetes Engine, Cloud Storage, BigQuery, VPC network, and Cloud Run. The "APIs & Services" item is expanded, showing sub-options: Library, Credentials, OAuth consent screen, and Page usage agreements. A red box highlights the "Enabled APIs & services" link under "Library".

In the center of the screen, there's a large, abstract geometric graphic consisting of triangles and circles. To the right of the graphic, a callout box says "Try our most advanced model: Gemini 1.5 Pro Experimental" and "Try Gemini".

A dark overlay at the bottom of the screen displays the message "Now viewing project 'RPS Grafana' in organization 'No organization'" with a close button (X).

The screenshot shows the Google Cloud APIs & Services dashboard. On the left, a sidebar lists 'Enabled APIs & services', 'Library', 'Credentials', 'OAuth consent screen', and 'Page usage agreements'. The main area displays three metrics: 'Traffic' (0.03/s), 'Errors' (1), and 'Median latency' (0.25 milliseconds). A modal window is open, stating 'Now viewing project "RPS Grafana" in organization "No organization"'. At the top right, there is a red box around the '+ ENABLE APIs AND SERVICES' button.

Our goal here is to enable the sheets API and the google drive API.

API APIs & Services – RPS Grafana

https://console.cloud.google.com/apis/library/browse?hl=en&organizationId=0&project=rps-grafana&q=Sheets

Free trial status: \$300.00 credit and 62 days remaining - with a full account, you'll get unlimited access to all of Google Cloud Platform.

DISMISS ACTIVATE

Google Cloud RPS Grafana

API API Library Sheets

API Library > "Sheets"

Filter Type to filter 1 result

Visibility Public (1)

Category Google Enterprise APIs (1) Google Workspace (1)

Google Sheets API Google Enterprise API With the Google Sheets API, you can create and modify spreadsheets data in Google Sheets.

Now viewing project "RPS Grafana" in organization "No organization"

API APIs & Services – RPS Grafana

https://console.cloud.google.com/apis/library/browse?hl=en&organizationId=0&project=rps-grafana&q=Drive

Free trial status: \$300.00 credit and 62 days remaining - with a full account, you'll get unlimited access to all of Google Cloud Platform.

DISMISS ACTIVATE

Google Cloud RPS Grafana

API API Library Drive

API Library > "Drive"

Filter Type to filter 11 results

Visibility Public (11)

Category Analytics (2) Big data (3) Databases (2) Developer tools (2) Healthcare (1) Google Enterprise APIs (6)

Drive Labels API Google Use the Drive Labels API to define custom metadata taxonomies to organize, find, enforce policy, and support business processes on Google Drive files and folders.

Google Drive API Google Enterprise API With the Google Drive API, you can access resources from Google Drive to create files, manage file sharing, search for files and folders, and more.

Google Drive Activity API Google Enterprise API With the Google Drive Activity API, you can retrieve information about a user's Google Drive activity. This provides additional functionality on top of the Google Drive API for your app to display activity on a user's files, track changes to specific files or folders, or alert a user to new comments or changes to files.

Make sure both of these are enabled!

- 5) Now we need to **make a service account**. We'll navigate to “Credentials” to do this:

The screenshot shows the Google Cloud Platform console with the URL <https://console.cloud.google.com/welcome?hl=en&organizationId=0&project=rps-grafana>. The left sidebar has sections like 'Cloud overview' and 'Products & solutions'. Under 'PINNED PRODUCTS', there's a list of services including 'APIs & Services', 'Billing', 'IAM & Admin', 'Marketplace', 'Compute Engine', 'Kubernetes Engine', 'Cloud Storage', 'BigQuery', and 'VPC network'. A dropdown menu for 'APIs & Services' is open, showing options: 'Enabled APIs & services', 'Library', 'Credentials' (which is highlighted with a red box), 'OAuth consent screen', and 'Page usage agreements'. Below the dropdown, there are buttons for 'Create a GKE cluster' and 'Create a storage bucket'. To the right, there are cards for 'IAM & Admin', 'Billing', and 'Compute Engine'. A sidebar on the right says 'Try our most advanced model: Gemini 1.5 Pro Experimental' and 'Try Gemini'.

The screenshot shows the 'APIs & Services' page with the URL <https://console.cloud.google.com/apis/credentials?hl=en&organizationId=0&project=rps-grafana>. The left sidebar has sections: 'Enabled APIs & services', 'Library', 'Credentials' (which is highlighted with a red box), 'OAuth consent screen', and 'Page usage agreements'. The main area has a heading 'Create credentials to access your enabled APIs.' with a 'Learn more' link. A warning message says 'Remember to configure the OAuth consent screen with information about your application.' with a 'CONFIGURE CONSENT SCREEN' button. Below this, there are sections for 'API Keys', 'OAuth 2.0 Client IDs', and 'Service Accounts'. The 'Service Accounts' section shows a table with one row: 'Email' (rps-grafana@rps-grafana.iam.gserviceaccount.com), 'Name' (RPS-Grafana), and 'Actions' (edit and delete icons). There are also 'RESTORE DELETED CREDENTIALS' and 'Manage service accounts' buttons.

The screenshot shows the Google Cloud Platform API Credentials page. The left sidebar has 'Enabled APIs & services' expanded, with 'Credentials' selected. A modal window is open over the main content, titled 'Create credentials to access your enabled APIs'. It lists three options: 'API key', 'OAuth client ID', and 'Service account'. The 'Service account' option is highlighted with a red box. Below it, there's a 'Help me choose' section with a few questions. The main content area shows sections for 'OAuth 2.0 Client IDs' and 'Service Accounts'. Under 'Service Accounts', there is one entry: 'rps-grafana@rps-grafana.iam.gserviceaccount.com' with 'Actions' buttons.

Fill out the name and description for the service account and hit “Done”. You should see your new service account under “Service Accounts”. Select it and navigate to “Keys”.

The screenshot shows the Google Cloud Platform API Credentials page again. The 'Service Accounts' section now includes the newly created service account: 'rps-grafana@rps-grafana.iam.gserviceaccount.com' and 'rps-grafana-tutorial@rps-grafana.iam.gserviceaccount.com'. Both entries have 'Actions' buttons. The 'rps-grafana-tutorial' entry is highlighted with a red box.

Free trial status: \$300.00 credit and 62 days remaining - with a full account, you'll get unlimited access to all of Google Cloud Platform.

DISMISS ACTIVATE

Google Cloud RPS Grafana Search (/) for resources, docs, products, and more

IAM & Admin rps-grafana-Tutorial

DETAILS PERMISSIONS KEYS METRICS LOGS

Service account details

Name: rps-grafana-Tutorial SAVE

Description: making this for the tutorial SAVE

Email: rps-grafana-tutorial@rps-grafana.iam.gserviceaccount.com

Unique ID: 114250278486702804034

Service account status

Disabling your account allows you to preserve your policies without having to delete it.

Enabled

DISABLE SERVICE ACCOUNT

Advanced settings

Free trial status: \$300.00 credit and 62 days remaining - with a full account, you'll get unlimited access to all of Google Cloud Platform.

DISMISS ACTIVATE

Google Cloud RPS Grafana Search (/) for resources, docs, products, and more

IAM & Admin rps-grafana-Tutorial

DETAILS PERMISSIONS KEYS METRICS LOGS

Keys

⚠ Service account keys could pose a security risk if compromised. We recommend you avoid downloading service account keys and instead use the [Workload Identity Federation](#). You can learn more about the best way to authenticate service accounts on Google Cloud [here](#).

Add a new key pair or upload a public key certificate from an existing key pair.

Block service account key creation using [organization policies](#). [Learn more about setting organization policies for service accounts](#)

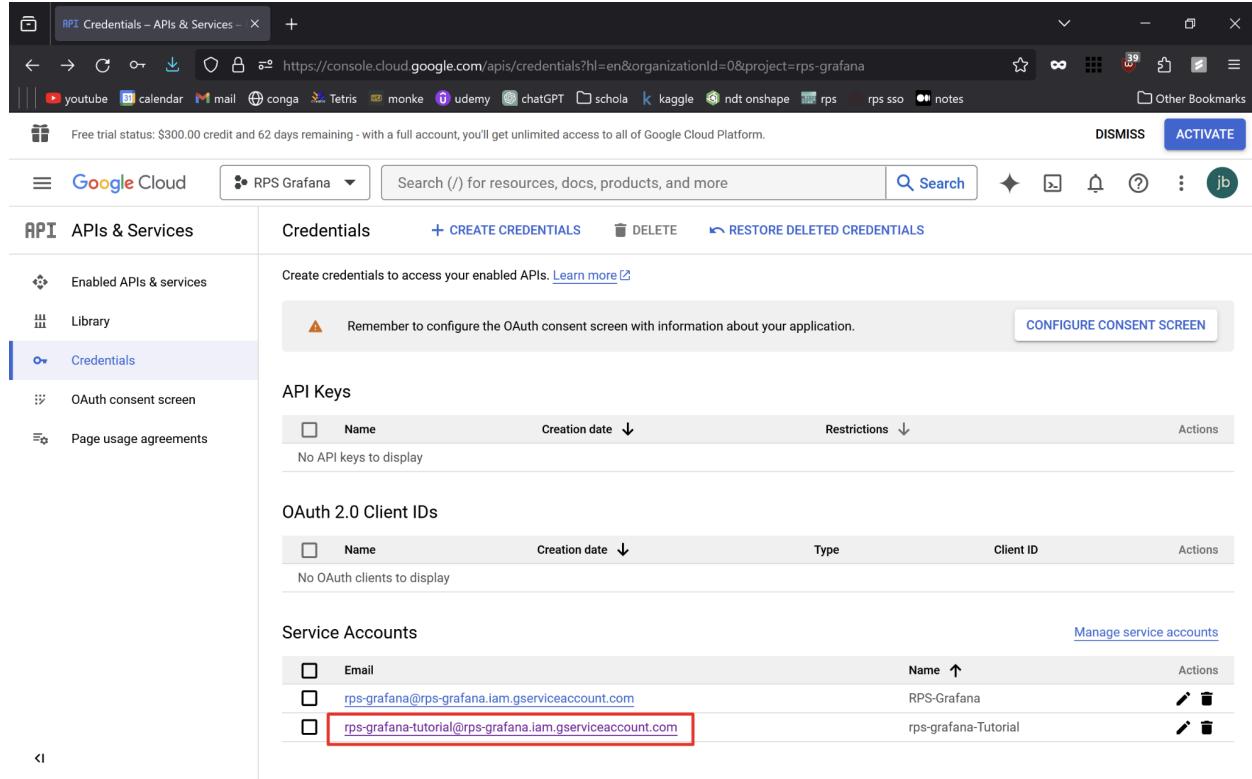
ADD KEY

Type	Status	Key	Key creation date	Key expiration date
No rows to display				

Create a new JSON Key. This JSON key will be downloaded. This is the only copy of this file that validates your access. If you lose it you have to create a new one and if it is

compromised you should create a new one. We will need this later to validate our access. For now, just move it into your project folder.

- 6) Now you need to share the google sheet you are using with the service account.
Copy the email address here:



The screenshot shows the Google Cloud API Credentials page. The left sidebar has 'APIs & Services' selected, with 'Enabled APIs & services', 'Library', and 'Credentials' (which is selected) listed. The main area shows 'Credentials' with a 'CREATE CREDENTIALS' button, a 'DELETE' button, and a 'RESTORE DELETED CREDENTIALS' link. A note says 'Remember to configure the OAuth consent screen with information about your application.' with a 'CONFIGURE CONSENT SCREEN' button. Below this is the 'API Keys' section, which is currently empty. The 'OAuth 2.0 Client IDs' section is also empty. The 'Service Accounts' section shows two entries: 'rps-grafana@rps-grafana.iam.gserviceaccount.com' (Email, Name: RPS-Grafana, Actions: edit, delete) and 'rps-grafana-tutorial@rps-grafana.iam.gserviceaccount.com' (Email, Name: rps-grafana-Tutorial, Actions: edit, delete). The second entry is highlighted with a red box.

And share edit access with that email on the google sheet. Then the service account should have access.

Getting the Python Data Scripts Set Up

The python files to pull the data, do the necessary analysis, and save the CSV files with the filtered data are located in [this github repo](#). I can add you as a collaborator if you give me your github username or email address. You will need to create a key.py file with the necessary file path (to your security token) and sheet id. The sheet id is part of the url when you visit the google sheet in your browser. If you have any questions about this, let me know.

***Note: once you are a collaborator on this repo and adding your own changes you need to make sure you correctly have your .gitignore setup. While I don't think it would expose a vulnerability if the path to your token file or the spreadsheet ID were public, we really should not publish these things.**

Once you have the python files saved and your keys.py file setup, run the files and voila! You (hopefully) have several CSV files saved in your folder and are ready to load them into Grafana.

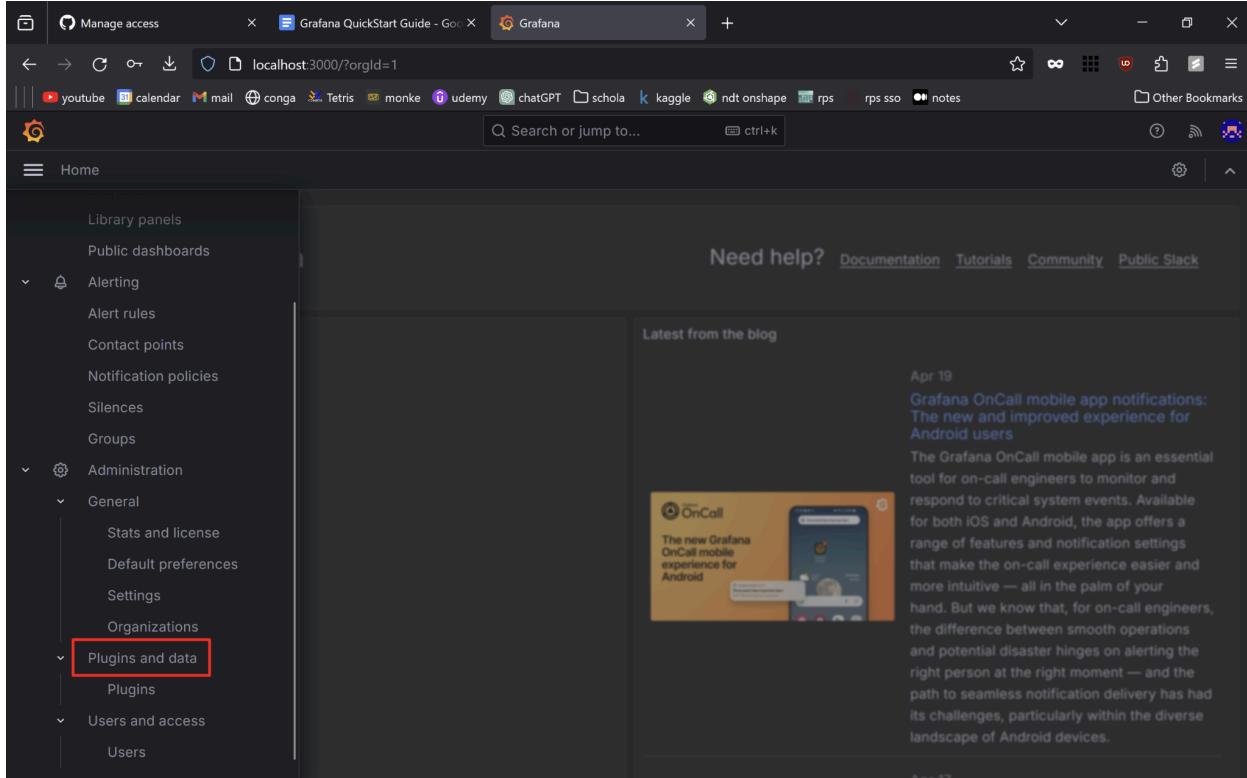
Another Note: One of the files in the git repo is “grafana.ini”. This file contains one line that allows the CSV extension in the next part to run. So make sure it is in your grafana directory too.

Loading the Data into Grafana

Grafana was built to interact with true databases like SQL. We are kind of abusing it to get it to do what we want. In this section we need to add the necessary plugins and our data sources as connections.

Steps:

- 1) Navigate to your Grafana server and sign in. I'm pretty sure you need to be signed in as Admin for some of the following steps.
- 2) Now we want to get to "Plug-Ins"



The screenshot shows the Grafana web interface. The left sidebar contains a navigation menu with the following items:

- Library panels
- Public dashboards
- Alerting
 - Alert rules
 - Contact points
 - Notification policies
 - Silences
 - Groups
- Administration
 - General
 - Stats and license
 - Default preferences
 - Settings
 - Organizations
 - Plugins and data
 - Users and access
 - Users

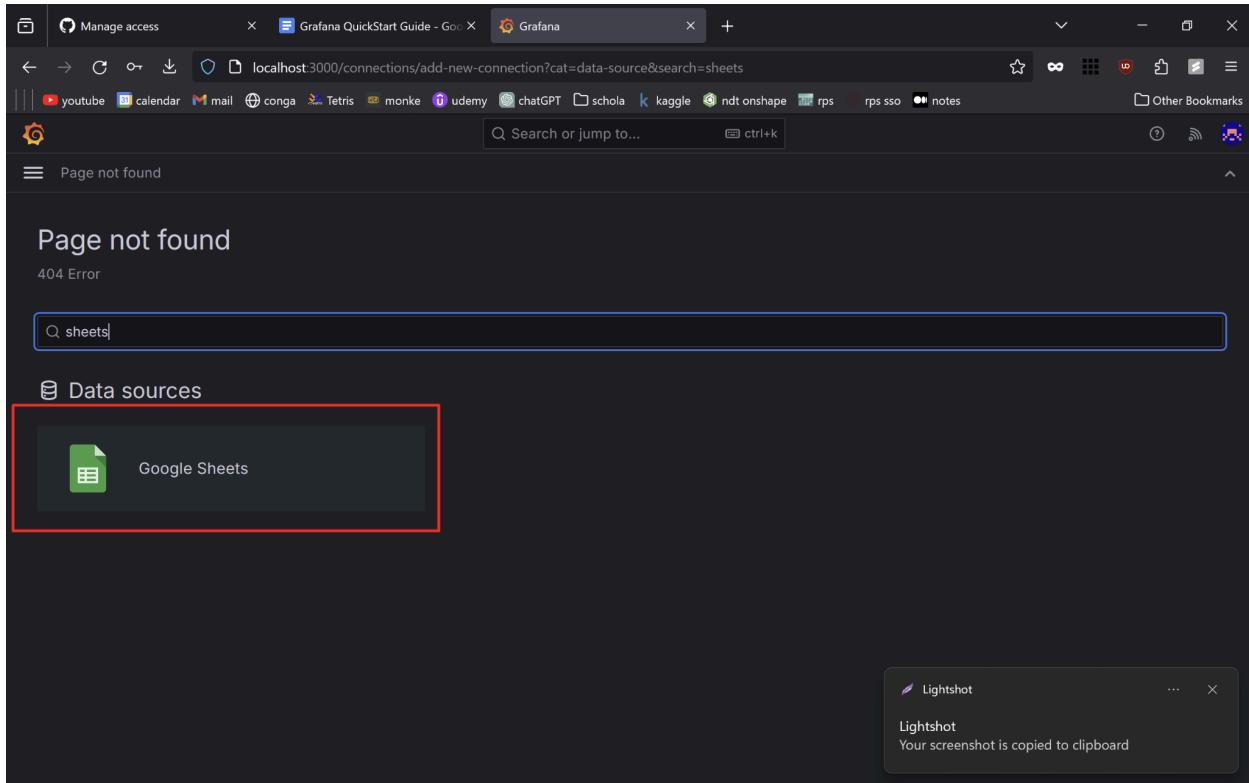
The 'Plugins and data' item is highlighted with a red box. The main content area displays a 'Latest from the blog' section with a post about the new Grafana OnCall mobile app experience for Android.

- 3) And from here we go to "Connections"

A screenshot of a web browser window showing the Grafana Plugins page. The URL in the address bar is `localhost:3000/plugins?filterBy=installed&filterByType=all&q=csv`. The page title is "Plugins". A search bar at the top contains the text "csv". Below the search bar are filter options: "Type" set to "All", "State" set to "Installed", and "Sort" set to "By name (A-Z)". A red box highlights the word "Connections" in the text "Extend the Grafana experience with panel plugins and apps. To find more data sources go to [Connections](#)".

- 4) We want to install two plug-ins although for now we will only use one. They are “CSV” and “Google Sheets”. Install both of them.

A screenshot of a web browser window showing the Grafana Connections page. The URL in the address bar is `localhost:3000/connections/add-new-connection?cat=data-source&search=csv`. The page title is "Grafana". A search bar at the top contains the text "csv". Below the search bar is a section titled "Data sources" containing two items: "CSV" and "Infinity". The "CSV" item is highlighted with a red box. The "CSV" icon is a blue circle with a white semi-circle and a yellow exclamation mark. The "Infinity" icon is a blue circle with a white infinity symbol.



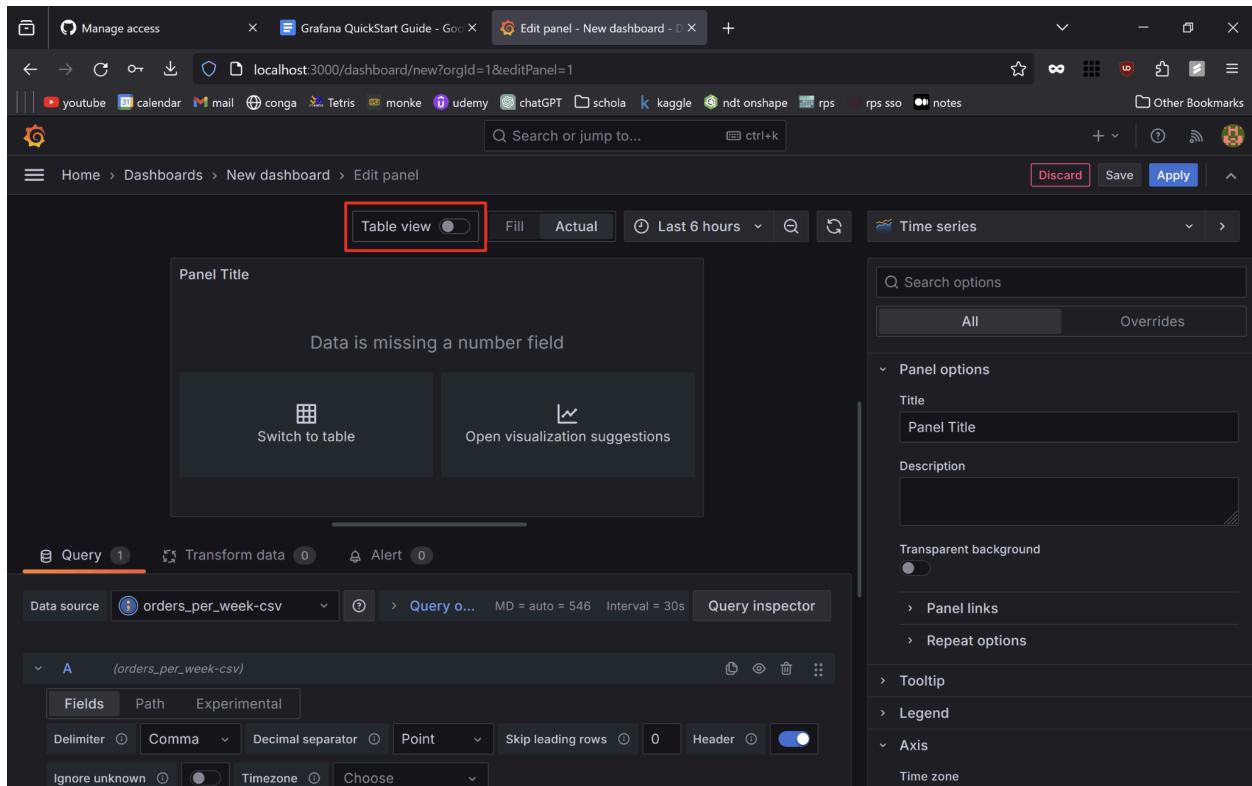
- 5) Great! Now we are ready to add our data sources. Each CSV we want to add will be a separate data source. Navigate to “Data Sources” in the left hand bar. There should be a very large blue button that says: “Add data source”. Click this button and then search for CSV and select it. It will bring you to a menu for the CSV plugin.
 - a) In the name field, specify the name of the CSV you want to load in. This does not have to match the file name.
 - b) For storage location select local.
 - c) For path to file you will specify: “/etc/grafana/filename”. Basically whatever is in the folder you told Docker to use as the volume is located at “/etc/grafana/” so all of the data needs to be in that folder. If it is in a subdirectory add that to the file path. So for me I’m putting: “/etc/grafana/orders_per_week.csv” in this field.
- 6) Now you should see the data source you added under “Data Sources”. Feel free to add the rest of your CSV files now or to go ahead and set up a dashboard.

Getting a Dashboard Set Up

Great now you get to do the fun part, visualizing your data. To do this, we'll create a dashboard and add a visualization to it.

Steps:

- 1) Navigate to Dashboards and create a new dashboard. Note: dashboards don't auto save so make sure to save your changes after you make them.
- 2) Now add a visualization and select the data source you want to use.
- 3) This will bring you to one mess of a menu.
 - a) First things first let's actually look at our data. In order to do this we'll switch the window in the top left to Table View:



- b) Now you'll actually see the data in what is basically a spreadsheet layout. Next we want to change the type of visualization we are trying to make. Navigate to this menu in the top right and change the type to "Trend".

The screenshot shows the Grafana interface with a dashboard being edited. The top navigation bar includes links for 'Manage access', 'Grafana QuickStart Guide', and 'Edit panel - New dashboard'. The browser address bar shows 'localhost:3000/dashboard/new?orgId=1&editPanel=1'. The main content area displays a table with data for two periods: '23_Fall' and '24_Spring'. The table has columns for 'Week' and values for weeks 1 through 5. Below the table is a 'Query' section for a data source named 'orders_per_week-csv'. The 'Fields' tab is selected in the Query inspector. On the right, a sidebar titled 'Time series' contains various panel options such as 'Title', 'Description', and 'Axis' settings.

- c) Now on the right you will see all the options for this panel type and on the bottom left you will see the “Query” section. If we were using an actual database and making queries this would make more sense. But we already have all our data and it is already filtered. So we just need to select our data and type.
- 4) In order to use our data we need to specify the data and type.
 - a) In the bottom left panel find the Field line

The screenshot shows the Grafana dashboard editor with a table panel. The table has three columns: 'Week', '23_Fall', and '24_Spring'. The 'Week' column contains numerical values (1, 2, 3, 4) but is currently classified as a string type. To the right of the table is a 'Trend' panel configuration sidebar. The 'Fields' tab in the sidebar shows the 'Type' dropdown set to 'String'. A red box highlights this dropdown, indicating it needs to be changed to 'Number'.

- b) Enter “Week” into the field section and change the type from string to number. The view of this column in the table will change accordingly.
- c) Click the plus sign to add another field. Classify each column of your spreadsheet (that you want to plot) as a number.
- 5) Once you have classified all of the data columns, move your attention to the right panel with the trend plot options. Specify the “Week”’s field as the X axis. If you have added a lot of fields you sometimes have to remove and re-add this field for it to register.
- 6) Now you can exit table view and start playing with the aesthetic options for the panel.
- 7) Once you are happy with the style of your panel, click Apply in the top right and you will get brought back to your dashboard. You can receive and move your panel here as well as changing the color of specific trends.
- 8) Don’t forget to save your Dashboard!**

If you have any questions don’t hesitate to reach out!