**Step 1: Login to GitHub**

Access your target repository where you want to configure the automation.

**Step 2: Create Workflow File**

Inside the repo, create a .yml file under:  
.github/workflows/trigger\_dataflow.yml

A screenshot of a computer

AI-generated content may be incorrect.

To create a workflow, go to the **Actions** tab in your GitHub repository. Inside the workflow, create a YAML file that specifies when to run the Python script—such as on every commit.  
 Confidential information like login credentials is stored in a config file within the same directory. The Python script uses these details to trigger the DataGaps dataflow using the dataflow ID.

**Step 3: Add Python Script**

Add a script (e.g., trigger\_script.py) that uses the requests module to call the DataGaps API.  
In this script, include the Python code that triggers the dataflow by calling the API using the appropriate dataflow ID.

**Step 4: Store Secrets values like login details in the config file**

* DATAOPS\_CLIENT\_ID
* DATAOPS\_CLIENT\_SECRET
* DATAOPS\_USERNAME
* DATAOPS\_PASSWORD

A screenshot of a computer

AI-generated content may be incorrect.

**Further implementation**: If the user wants to trigger multiple dataflows or use cases, the dataflow IDs can be stored in a list. The script can then iterate over this list to trigger multiple dataflows sequentially in a single run.