Kestra Overview

Introduction

Kestra is a robust data orchestrator designed to streamline IT workflow management. It offers comprehensive capabilities to automate, schedule, and orchestrate a variety of IT tasks, including running scripts, executing queries, calling APIs, and deploying infrastructure. With seamless integration across multiple tools and platforms, such as Terraform, Ansible, Python, Node.js, and PowerShell, Kestra extends its functionality to cloud providers and AI technologies, making it a versatile solution for modern IT environments.

Key Features

1. Declarative Workflow Creation:

   - Simplified Workflow Management: Users define the desired steps and actions within their IT stack, and Kestra handles the execution. This declarative approach ensures consistency and scalability in workflow management.
   - Scalable Multi-Tenant Architecture: The platform supports a multi-tenant environment, simplifying deployments and accelerating development tasks.

2. Comprehensive Dashboard:

   - User-Friendly UI: The all-inclusive dashboard allows users to create workflows directly from the interface, featuring an embedded editor for convenience.
   - In-Depth Monitoring: Users gain access to detailed metrics on execution, logs, and other critical data points.

3. Business Stakeholder Accessibility:

   - Code-Free Adjustments: Stakeholders can modify queries and input variables without needing to deploy new code.
   - Embedded Documentation and Tools: The platform includes documentation, examples, task properties, and a topology view alongside the code editor to aid efficient workflow creation.

4. Real-Time Execution Insights:

   - Live Updating Views: During execution, Kestra provides real-time updates, dynamically displaying task durations and dependencies.
   - Full-Text Log Search: Troubleshooting and inspecting workflow behaviors are simplified with searchable logs.

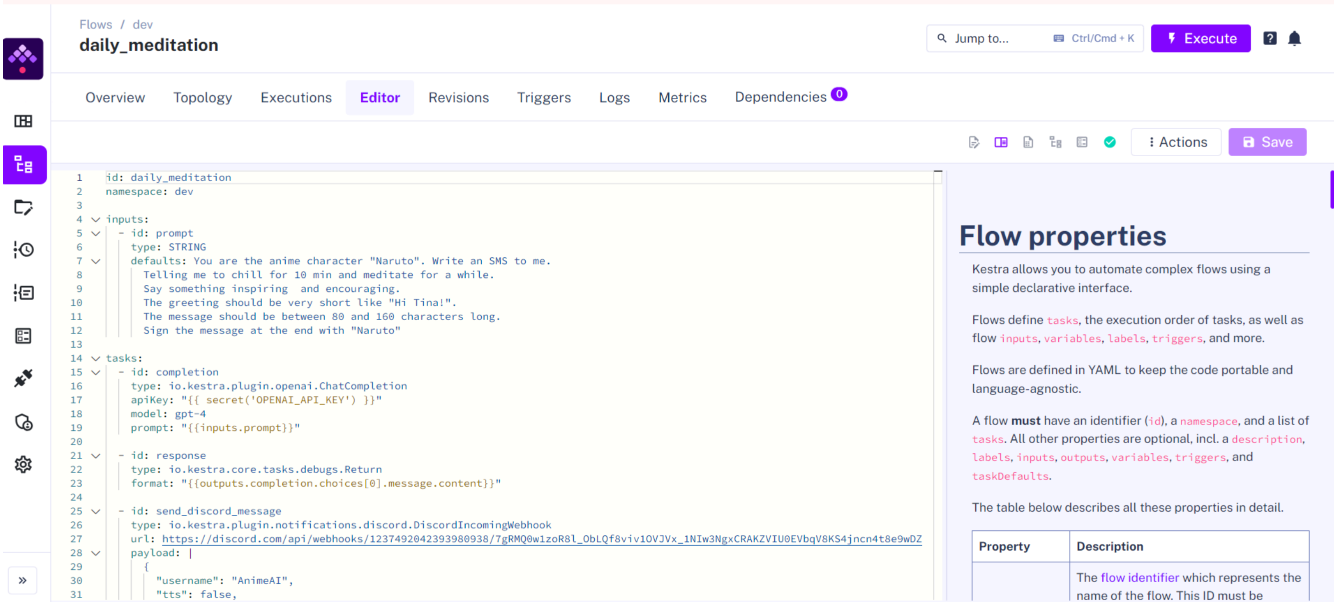
5. Output and Metrics Tracking:

    - Detailed Results: The platform tracks additional results, artifacts, and metadata, such as the number of rows returned by a query.

6. Flexible Execution Options:

    - Manual and Event-Driven Triggers: Workflows can be executed manually or in response to various events, such as time-based schedules, webhooks, new files in cloud storage, and new messages in a queue.
    - Scheduling and Backfilling: Using simple cron expressions, users can schedule workflows and backfill data for past intervals, specifying start dates and pausing flows for modifications.

Kestra provides a powerful solution for managing and orchestrating IT workflows, combining ease of use with extensive capabilities. Its support for various tools and platforms, along with features like real-time execution insights and flexible scheduling, make it an invaluable asset for IT teams looking to optimize their operations and improve efficiency.



```yaml
id: daily_meditation
namespace: dev

inputs:
  - id: prompt
    type: STRING
    defaults: You are the anime character "Naruto". Write an SMS to me.
      Telling me to chill for 10 min and meditate for a while.
      Say something inspiring  and encouraging.
      The greeting should be very short like "Hi Tina!".
      The message should be between 80 and 160 characters long.
      Sign the message at the end with "Naruto"

tasks:
  - id: completion
    type: io.kestra.plugin.openai.ChatCompletion
    apiKey: "{{ secret('OPENAI_API_KEY') }}"
    model: gpt-4
    prompt: "{{inputs.prompt}}"

  - id: response
    type: io.kestra.core.tasks.debugs.Return
    format: "{{outputs.completion.choices[0].message.content}}"

  - id: send_discord_message
    type: io.kestra.plugin.notifications.discord.DiscordIncomingWebhook
    url: https://discord.com/api/webhooks/1237492042393980938/7gRMQ0w1zoR8l_ObLQf8viv1OVJVx_1NIw3NgxCRAKZVIU0EVbqV8KS4jncn4t8e9wDZ
    payload: |
      {
        "username": "AnimeAI",
        "tts": false,
```

Flow properties

Kestra allows you to automate complex flows using a simple declarative interface.

Flows define `tasks`, the execution order of tasks, as well as flow `inputs`, `variables`, `labels`, `triggers`, and more.

Flows are defined in YAML to keep the code portable and language-agnostic.

A flow **must** have an identifier (`id`), a `namespace`, and a list of `tasks`. All other properties are optional, incl. a `description`, `labels`, `inputs`, `outputs`, `variables`, `triggers`, and `taskDefaults`.

The table below describes all these properties in detail.

Property	Description
	The flow identifier which represents the name of the flow. This ID must be

Figure 1: Kestra Flow UI

Project Description

Life can be overwhelming with endless notifications from various apps, leaving you feeling like you're juggling a hundred things at once.

Imagine having a virtual sidekick that's as awesome as your favorite fictional character, guiding you through your daily tasks and helping you stay organized effortlessly. This project is designed to simplify your life by managing your schedule, health routines, and personal goals seamlessly. You will never miss a beat. It sends you gentle reminders to meditate every morning, keeps track of your meals' macros to help you maintain a balanced diet, and even provides daily reports on your running metrics like speed and distance, empowering you to stay healthy and active. But that's not all! It is your personal assistant for everything you need. It reminds you of important meetings scheduled via Google Calendar, ensuring you're always punctual and prepared. It keeps your calendar in check, helping you manage your time effectively. Plus, it's always there to remind you of anything else you ask for, whether it's picking up groceries on your way home or calling your mom on her birthday.

It integrates seamlessly into your daily routine, making life simpler, more organized, and enjoyable.

Kestra Flows

Flow: Meditation Reminder

- Description: The first flow involves sending you daily reminders to meditate every morning at 9 AM. You will receive these messages from your anime buddy every day. Kestra enables us to schedule this flow using the Schedule Trigger, which generates new executions on a regular cadence based on a cron expression. Then, we use the OpenAI plugin. Plugins are the building blocks of Kestra's tasks and triggers. They encompass components that interact with external systems and perform the actual work in your flows. Here, we use the OpenAI plugin and the Chat Completion task, providing a prompt to get a response from the LLM. This generates the inspiring messages that we will receive from the bot. We send the data to a Discord chat bot by using the Discord plugin and the DiscordIncomingWebhook task.

- Topology:

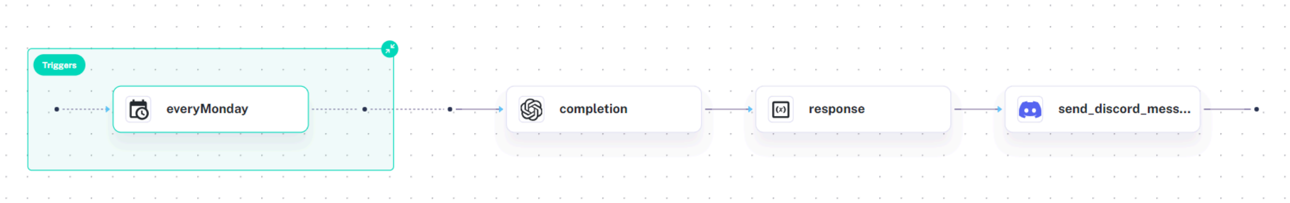


Figure 2: Kestra Flow Topology

Flow: Event Reminder

- Description: The next flow that constitutes the Discord bot project is an event reminder from your Google Calendar. We imported the meetings/events saved in Google Calendar to Google Sheets using Sheet2GCal. We read the file row by row using the Google Workspace plugin and the Google Sheet task. The output was saved in Kestra's internal storage. Then, we filtered only the columns 'Title' and 'Start Date' from the table and saved the output. We used the Script task within the Python plugin to run the Python code. Kestra also allows you to check the output directly from the UI. We then ran other Python queries to extract only the rows where the date matches the task date. We parsed the result using the Parse task and used the If task to check if the result is 30 minutes away from the current task time. If so, we sent a message to Discord using the DiscordIncomingWebhook task. The flow is scheduled every hour daily.

Topology:

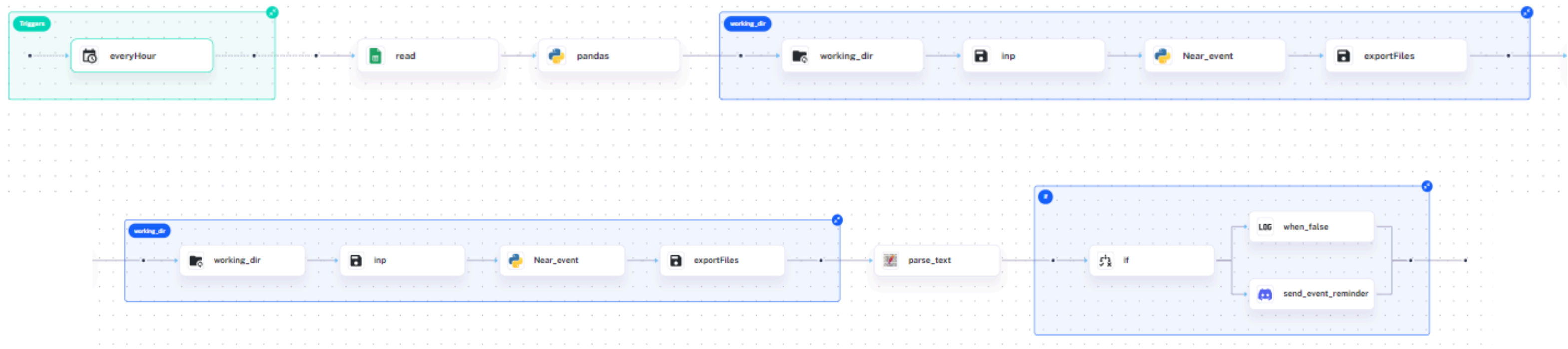


Figure 3: Kestra Flow Topology

Flow: Custom Reminders

- Description: This flow is triggered by a request from Discord. The text format should begin with 'remind please to X at a specific time.' The flow then loops over the table rows to check if the time for 'X' is approaching. It stores the event in a BigQuery table. The second flow is executed every two minutes using the BigQueryTrigger, where it loops over the table to check for events coming up in two minutes.

Topology:

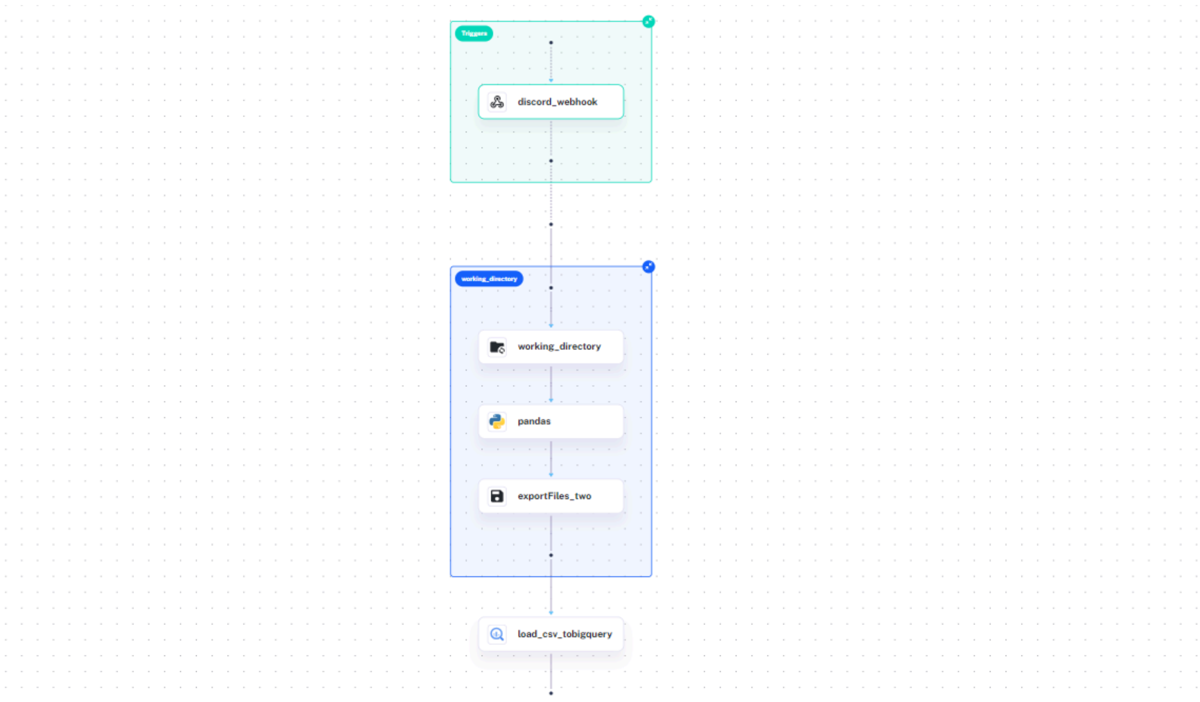


Figure 4: Kestra Flow Topology

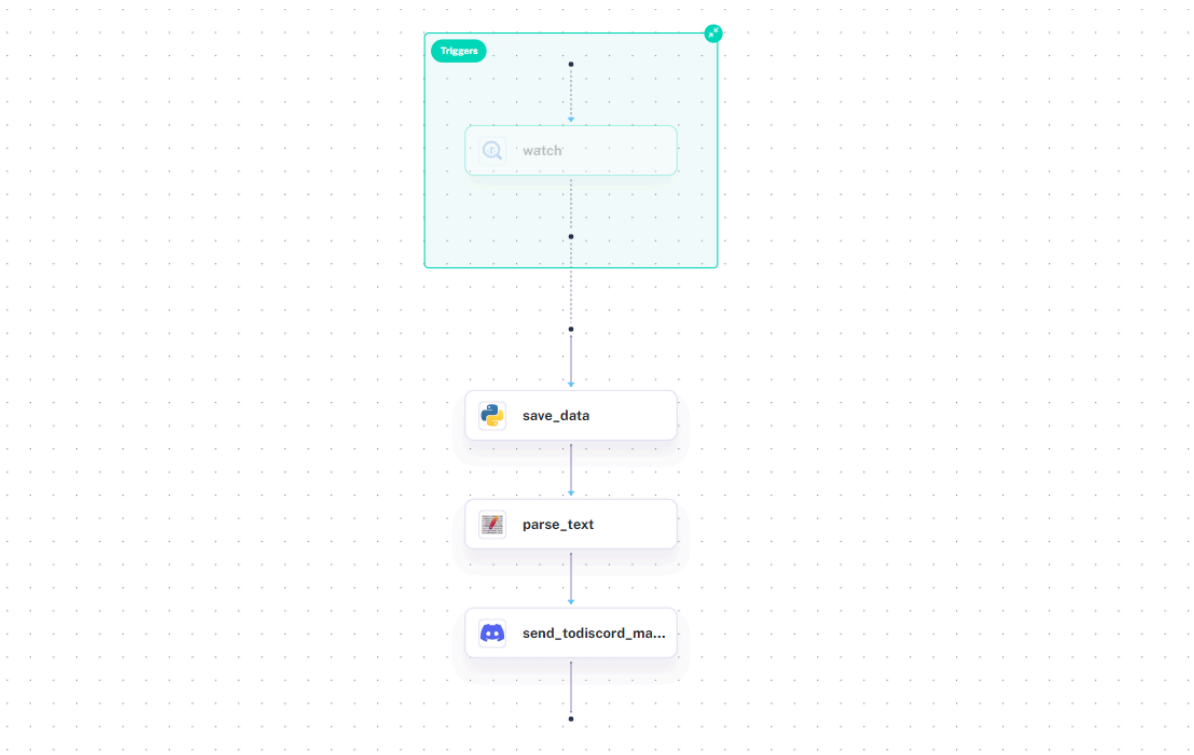
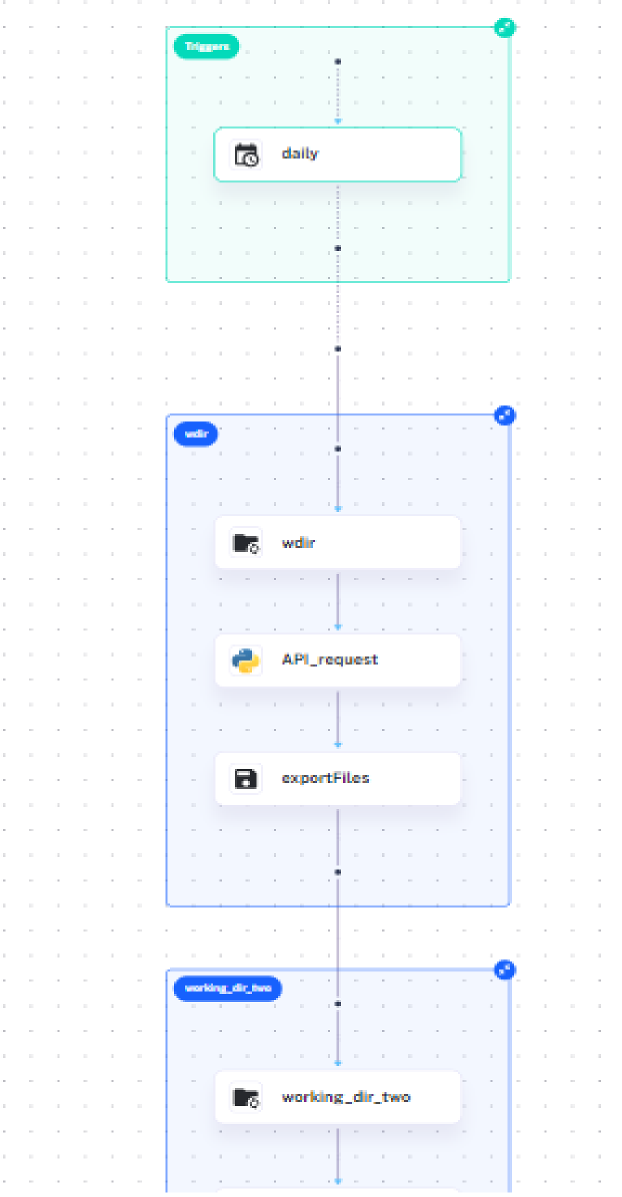


Figure 5: Kestra Flow Topology

Flow: Daily Workout

- Description:This flow involves your daily workout. If you ever want to check your daily or even weekly workouts, depending on specific metrics or physical activities that interest you or fit your own situation, Kestra can allow you to send daily reports about your workout from Strava or any other fitness application. In our case, we retrieved data from our own Strava account using a Get HTTP Request in a Python script, and then we saved the JSON file into a CSV file. We ran a Python script to get only the rows about date, type of sport (which is run), speed, and distance. Then we sent the file to BigQuery. The flow is scheduled every day at 8 PM.

- Topology:



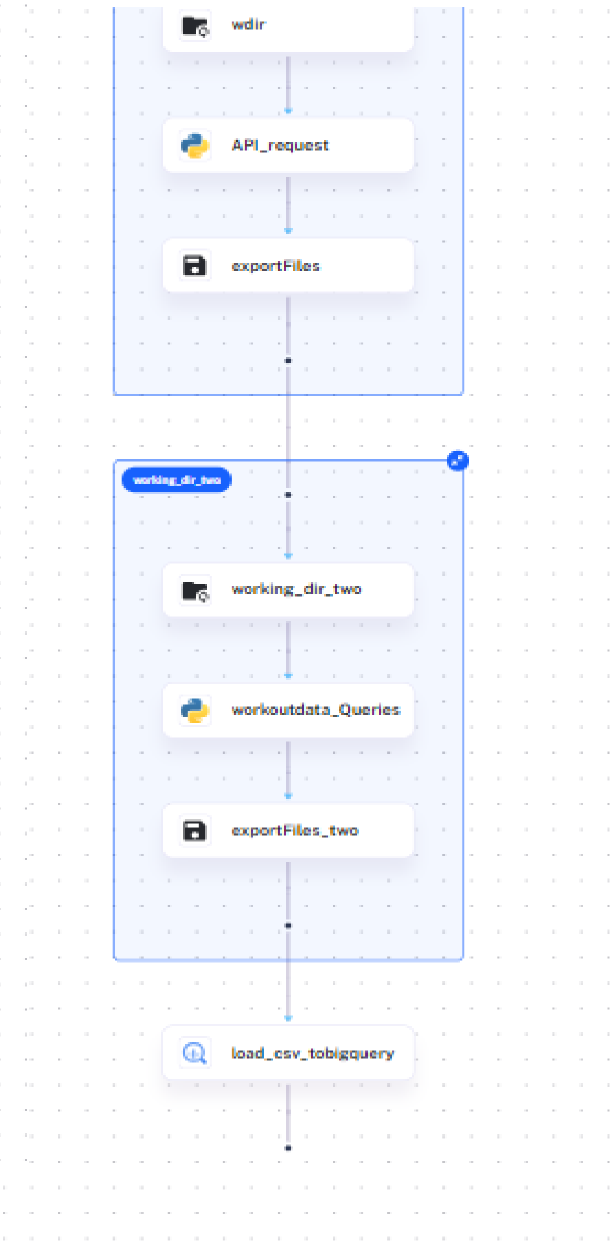


Figure 6: Kestra Flow Topology

Flow: Daily Steps

- Description: Description: In this flow, we use the BigQuery trigger to check the rows of your daily steps table. If the date column matches the task execution date, it will send you a message via Discord webhook.

Topology:

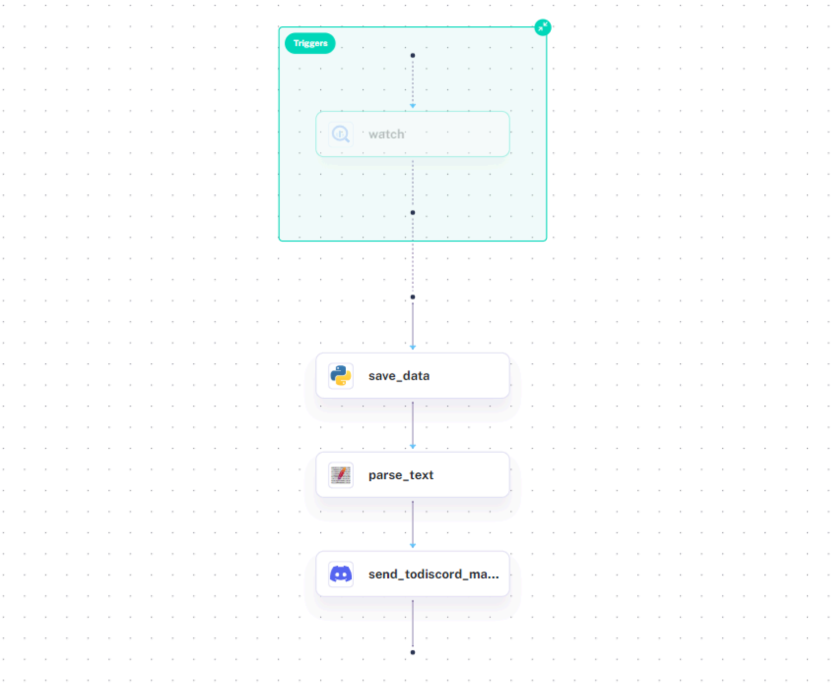


Figure 7: Kestra Flow Topology

Flow: Daily Nutrition

- Description: This flow involves daily nutrition. We used a Webhook trigger, so whenever you send a picture of your meal via the Discord bot, the flow will be executed. We used the Chat Completion task to get the estimated macros of your meal, then we parsed the result and saved it in a CSV file with columns for Date, Carbs, Protein, and Fats, where each indicates the percentage of the meal. The file will then be loaded to a BigQuery table. This will be very useful if we want to create a dashboard using Looker Studio. The parsed text about your macros will be sent to you via Discord. You will also receive a picture of your favorite anime character or buddy eating the same meal. This is possible with the CreateImage task using the OpenAI plugin.

Topology:

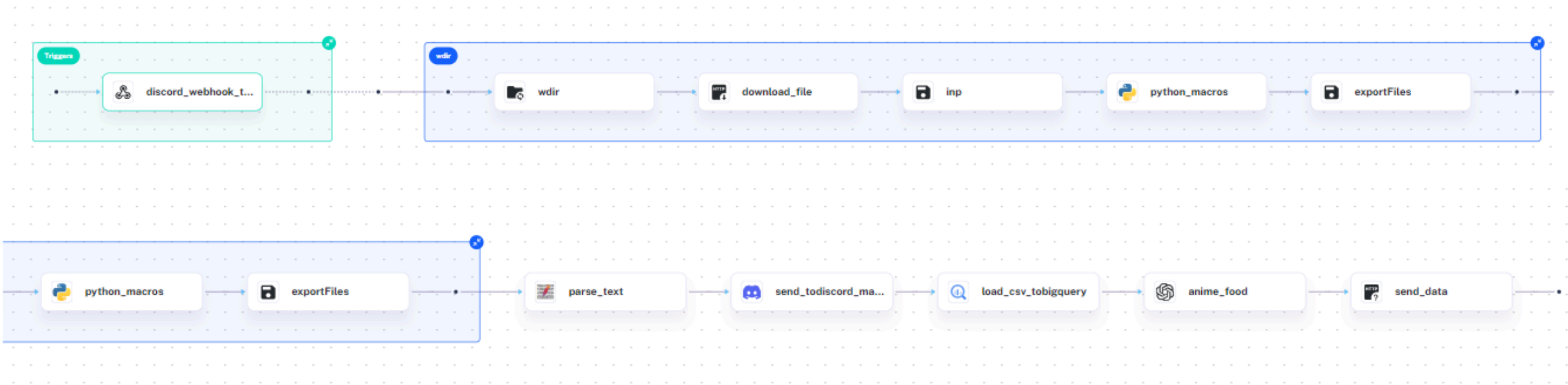


Figure 8: Kestra Flow Topology

Flow: Macro Tracking

- Description: This flow is about your BigQuery table that stores your daily macros. We used the BigQuery trigger to wait for an SQL query to return results and iterate through the rows. The query checks whether the date of the meal matches the execution date. If so, we get the row's data and then create a pie chart to show the average percentage of each macro for the day. The chart is then sent to your Discord.

Topology:

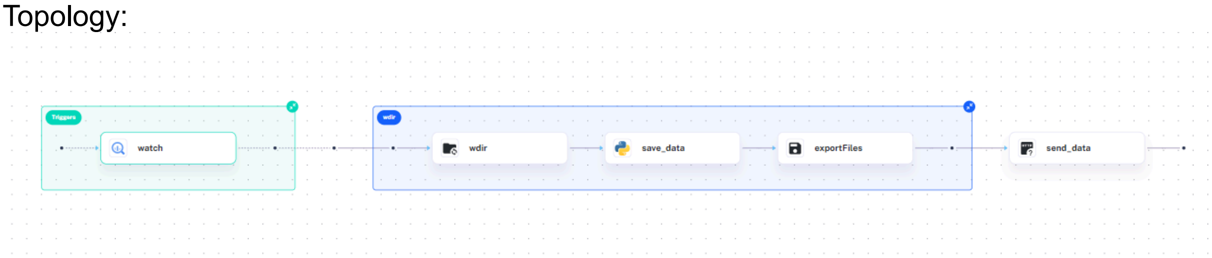


Figure 9: Kestra Flow Topology

Flow: Workout Data Check

- Description: This flow is about your BigQuery table that stores your daily workout. We used the BigQuery trigger to wait for an SQL query to return results and iterate through the rows. The query checks whether the date of the activity matches the execution date. If so, we get the row's data, save it as a CSV file, and send it via Discord.

Topology:

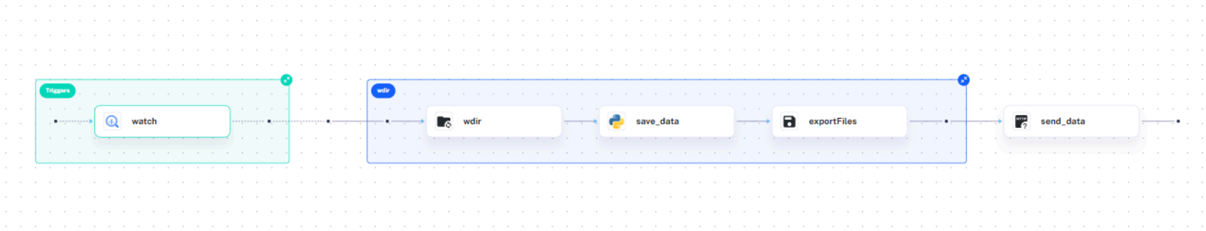


Figure 10: Kestra Flow Topology