

Update BillingIDs in GCP

Script Explanation

Purpose

The main goal of this script is to automate the process of updating the billing account associated with multiple Google Cloud Platform (GCP) projects. It's designed to be run from the command line and assumes you have the gcloud command-line tool installed and authenticated.

Steps

1. Import the json module:

- import json brings in Python's built-in library for working with JSON (JavaScript Object Notation) data. JSON is a standard format for exchanging data, and it's used here to store and manipulate project information.

2. Command Line Execution (not in the script):

- The comment indicates that you'd run this command in your terminal/CLI:

Bash

```
gcloud projects list --format="json" > projects.json
```

This command fetches a list of all your GCP projects in JSON format and saves it to a file named "projects.json."

3. Read Project Data:

- with open("projects.json", "r") as f: opens the file in read mode ("r") and gives it the alias f.
- projects = json.load(f) reads the JSON data from the file and converts it into a Python list of dictionaries (where each dictionary represents a project).

4. Set New Billing Account ID:

- new_billing_account_id = "YOUR_NEW_BILLING_ACCOUNT_ID" Here, you'd replace "YOUR_NEW_BILLING_ACCOUNT_ID" with the actual ID of the billing account you want to assign to the projects.

5. Iterate and Update:

- for project in projects: loops through each project dictionary in the projects list.
- project_name = project["projectId"] extracts the project ID.
- ptype = project["parent"]["type"] checks if the project belongs to an organization or not (some GCP resources might not be directly under an organization).

6. Conditional Update:

- if ptype == 'organization': If the project is part of an organization:
 - command = ... constructs the gcloud command to link the project to the new billing account.
 - print(command) (currently active) displays the command on your screen so you can review it.
 - # os.system(command) (commented out) If you uncomment this line and comment the print(command) line, it will actually execute the gcloud command and update the

- billing.
- else: If the project is not part of an organization:
 - `print(f"Skipping resource: {project_name} type: {ptype}")` The script informs you that it's skipping this project.

Key Points

- **Automation:** This script saves you from having to manually update each project's billing through the GCP console.
- **Safety Net:** By default, it only prints the commands it would run. This lets you review and make sure everything is correct before enabling the actual execution.
- **Flexibility:** You can easily modify the if condition to filter which projects get updated based on other criteria (e.g., project labels).