Build a Serverless Github Bot in GCP

Franklin Diaz

DE:AD:10:C5

Tuesday January 10, 2023

INTRODUCTION

Resources

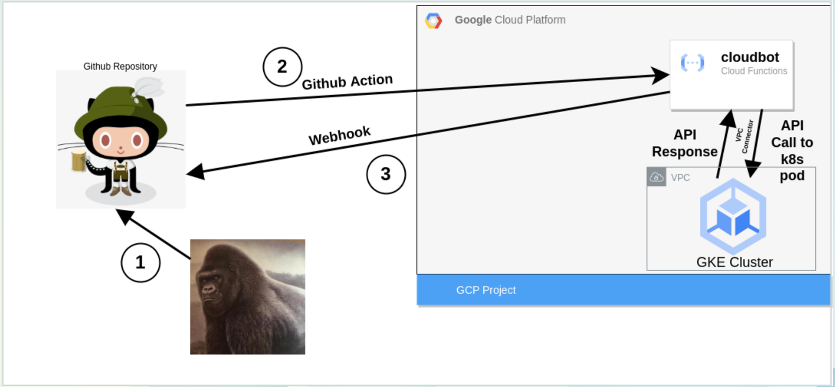
- Click here for Session Details
- Project source files are available: `https://github.com/devsecfranklin/workshop-codemash-2023`
- Prework available at this link.

Contact

Github: devsecfranklin .xXx. E-mail: **devsecfranklin@duck.com**

Overview: Usage

The big picture for operation.



Outline: What will we cover?

A high level overview of the learning path is as follows:

- Prework
  - Github project repository setup.
  - Set up the development environment.
  - Set up Google Cloud account.
- In Class setup slides.
- Review the Python source for the bot.
- Configure Terraform and deploy the bot.
- (Optional) Deploy to one of your repositories in Github.
- (Time permitting) Explore possibilities for extending the functionality.

PRE-WORK

Setup: VSCode

VSCode (https://code.visualstudio.com)

- Windows 64 bit User Installer: VSCodeUserSetup-x64-1.73.1.exe
- Mac Universal: VSCode-darwin-universal.zip
- Linux (Debian, Ubuntu): code_1.73.1-1667967334_amd64.deb
- Linux (Red Hat, Fedora, SUSE): code-1.73.1-1667967421.el7.x86_64.rpm

Click this link for details on using dev containers in VSCode

Setup: git

GIT (https://git-scm.com/downloads)

- Windows 32 Bit: Git-2.38.1-64-bit.exe
- Windows 64 Bit: Git-2.38.1-32-bit.exe
- Mac: git-2.15.0-intel-universal-mavericks.dmg

Setup: Docker Desktop

Docker Desktop (https://www.docker.com/)

- Windows: Docker Desktop Installer.exe
- MacOS (Intel Chip): Docker.dmg
- MacOS (M1 Chip): Docker.dmg
- Linux instructions can be found: here

Click here to see Docker setup steps from Microsoft

Setup: Clone and Open the Project Repository

- Time to clone the repository.
- Click this link for the Github repository
- In VSCode, press F1 and enter the command “Dev Containers: Open Folder in Container”
  - You can also choose “Dev Containers: Open Workspace in Container”
  - Here is the Microsoft VSCode dev containers tutorial
- From the top menu select “Terminal – New Terminal”
- Now “cd /workspaces/workshop-codemash-2023/bin” and type “setup-dev-env.sh”

Google Cloud: Account Setup

- Sign up for a free tier GCP account.
- Navigate to https://cloud.google.com/ and make sure you have a usable project to work in.
- Here is some infomration about creating projects in GCP

IN CLASS SETUP

Overview: direnv

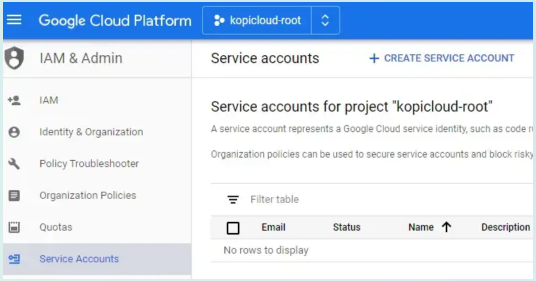
What the heck is direnv?

Google Cloud: Update Project Name and Login

- Update your project name in the file "/workspaces/workshop-codemash-2023/.envrc"
- Update your project name in the file "/workspaces/workshop-codemash-2023/src/config.ini"
- Type the command "direnv allow ." to reload the ENV variables.
- In the dev container, run the command "gcloud auth login" and follow the directions there.
- Verify you are connected to GCP with the command "gcloud auth list"

Google Cloud: Create Service User (1/7)

- We select our root project, we click the IAM & Admin menu, Service Accounts option, and finally, on the + Create Service Account button.
- The information on this slide was shamelessly ripped off from a Medium page written by Guillermo Musumeci.



Service accounts for project "kopicloud-root"

A service account represents a Google Cloud service identity, such as code r

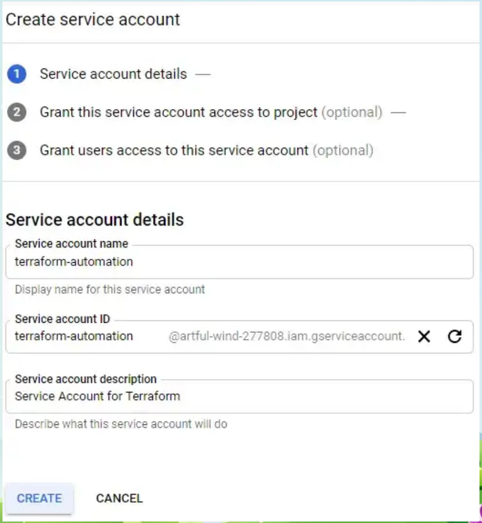
Organization policies can be used to secure service accounts and block risky

Filter table

	Email	Status	Name ↑	Description
No rows to display				

Google Cloud: Create Service User (cont. 2/7)

- We enter a name and description for the Service Account and click the CREATE button.


Create service account

1 Service account details —

2 Grant this service account access to project (optional) —

3 Grant users access to this service account (optional)

Service account details

Service account name: terraform-automation

Display name for this service account

Service account ID: terraform-automation @artful-wind-277808.iam.gserviceaccount.

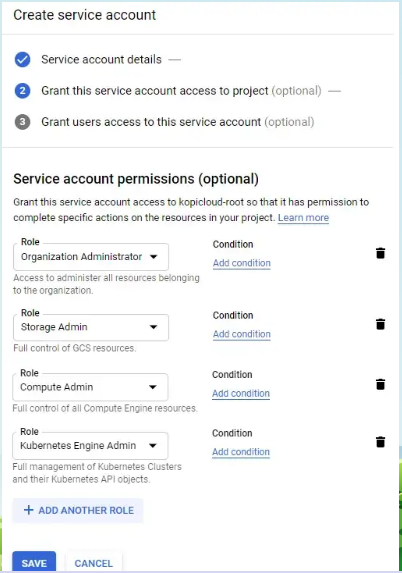
Service account description: Service Account for Terraform

Describe what this service account will do

Google Cloud: Create Service User (cont. 3/7)

In this step, we grant the Service Account access to the project. We will need to add the following Roles and click the CONTINUE button.

- Organization Administrator
- Storage Admin → Full access to Google Cloud Storage
- Compute Admin → Full control of Compute Engine resources (Virtual Machines)



Create service account

Service account details —
2 Grant this service account access to project (optional) —
3 Grant users access to this service account (optional)

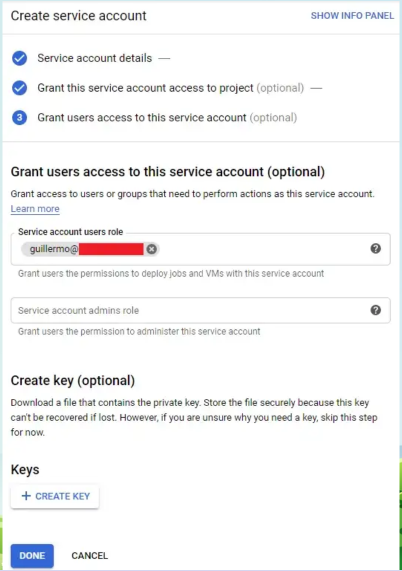
Service account permissions (optional)

Grant this service account access to kopicloud-root so that it has permission to complete specific actions on the resources in your project. Learn more

Role	Condition
Organization Administrator — Access to administer all resources belonging to the organization.	Add condition
Storage Admin — Full control of GCS resources.	Add condition
Compute Admin — Full control of all Compute Engine resources.	Add condition
Kubernetes Engine Admin — Full management of Kubernetes Clusters and their Kubernetes API objects.	Add condition

Google Cloud: Create Service User (cont. 4/7)

In the last step, we grant users access to the Service Account.



Create service account

Service account details —

Grant this service account access to project (optional) —

3 Grant users access to this service account (optional)

Grant users access to this service account (optional)

Grant access to users or groups that need to perform actions as this service account. Learn more

Service account users role: guillermo@[redacted]

Grant users the permissions to deploy jobs and VMs with this service account

Service account admins role

Grant users the permission to administer this service account

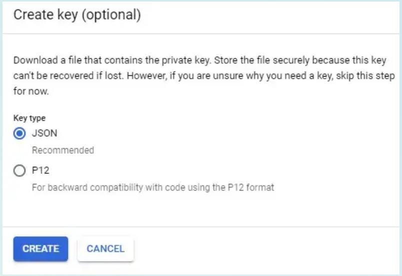
Create key (optional)

Download a file that contains the private key. Store the file securely because this key can't be recovered if lost. However, if you are unsure why you need a key, skip this step for now.

Keys

Google Cloud: Create Service User (cont. 5/7)

We click on the + CREATE KEY button to generate our authentication key file. This key in JSON format will be used by Terraform to authenticate to GCP.



Create key (optional)

Download a file that contains the private key. Store the file securely because this key can't be recovered if lost. However, if you are unsure why you need a key, skip this step for now.

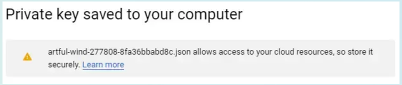
Key type

[✓ JSON] Recommended

P12 For backward compatibility with code using the P12 format

Google Cloud: Create Service User (cont. 6/7)

We download the JSON file and store it in a secure folder or vault.

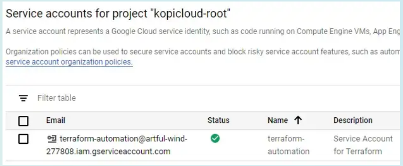


Private key saved to your computer

artful-wind-277808-8fa36bbabd8c.json allows access to your cloud resources, so store it securely. Learn more

Google Cloud: Create Service User (cont. 7/7)

- We click on the Done button to create the Service Account, and here is our new Service Account.
- This concludes my shameless ripped off from a Medium page written by Guillermo Musumeci.



Service accounts for project "kopicloud-root"

A service account represents a Google Cloud service identity, such as code running on Compute Engine VMs, App Eng

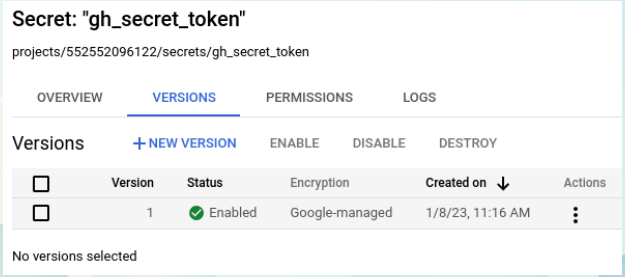
Organization policies can be used to secure service accounts and block risky service account features, such as autom
service account organization policies.

Filter table

	Email	Status	Name ↑	Description
☐	terraform-automation@artful-wind-277808.iam.gserviceaccount.com	✓	terraform-automation	Service Account for Terraform

Google Cloud: Create Secret in Secrets Mgr

- The Cloud Function is expecting us to create a secret named "gh_secret_token".
- Enable the Secret Manager service.
- Add the secret.



Secret: "gh_secret_token"

projects/552552096122/secrets/gh_secret_token

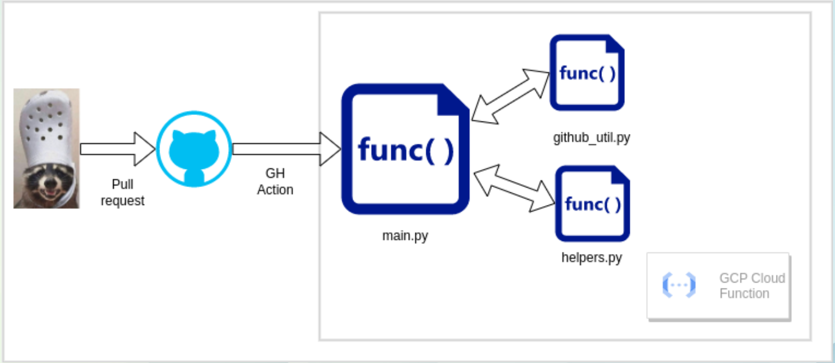
☐	Version	Status	Encryption	Created on ↓	Actions
☐	1	✓ Enabled	Google-managed	1/8/23, 11:16 AM	⋮

No versions selected

PYTHON

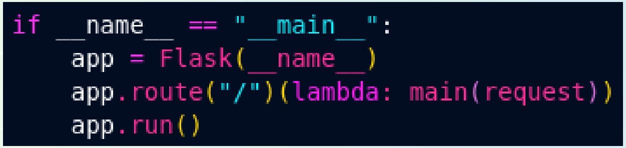
Overview: Python Functions

The big picture for the Python code files.



The Python Application

- The main function is essentially a Flask app that waits for an incoming JSON messages.
- Let's take a closer look

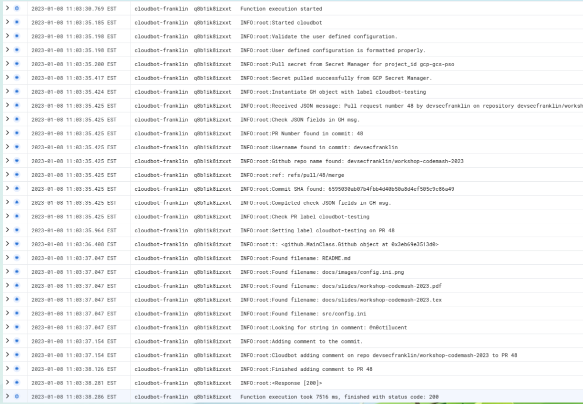


```
if __name__ == "__main__":
    app = Flask(__name__)
    app.route("/")(lambda: main(request))
    app.run()
```

Python: Logging

- Logging is set to the “INFO” level.
- The log files show up in GCP under the cloud function.

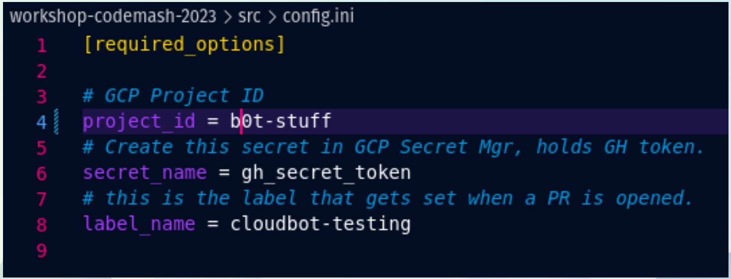
```python
logging.basicConfig(level=logging.INFO)
logger = logging.getLogger()
logger.setLevel(logging.INFO)
```



```
2023-01-08 11:03:30.769 EST  cloudbot-franklin  q8b1ik8izxxt  Function execution started
2023-01-08 11:03:35.185 EST  cloudbot-franklin  q8b1ik8izxxt  INFO:root:Started cloudbot
2023-01-08 11:03:35.198 EST  cloudbot-franklin  q8b1ik8izxxt  INFO:root:Validate the user defined configuration.
2023-01-08 11:03:35.198 EST  cloudbot-franklin  q8b1ik8izxxt  INFO:root:User defined configuration is formatted properly.
2023-01-08 11:03:35.200 EST  cloudbot-franklin  q8b1ik8izxxt  INFO:root:Pull secret from Secret Manager for project_id gcp-gcs-pso
2023-01-08 11:03:35.417 EST  cloudbot-franklin  q8b1ik8izxxt  INFO:root:Secret pulled successfully from GCP Secret Manager.
2023-01-08 11:03:35.424 EST  cloudbot-franklin  q8b1ik8izxxt  INFO:root:Instantiate GH object with label cloudbot-testing
2023-01-08 11:03:35.425 EST  cloudbot-franklin  q8b1ik8izxxt  INFO:root:Received JSON message: Pull request number 48 by devsecfranklin on repository devsecfranklin/worksh
2023-01-08 11:03:35.425 EST  cloudbot-franklin  q8b1ik8izxxt  INFO:root:Check JSON fields in GH msg.
2023-01-08 11:03:35.425 EST  cloudbot-franklin  q8b1ik8izxxt  INFO:root:PR Number found in commit: 48
2023-01-08 11:03:35.425 EST  cloudbot-franklin  q8b1ik8izxxt  INFO:root:Username found in commit: devsecfranklin
2023-01-08 11:03:35.425 EST  cloudbot-franklin  q8b1ik8izxxt  INFO:root:Github repo name found: devsecfranklin/workshop-codemash-2023
2023-01-08 11:03:35.425 EST  cloudbot-franklin  q8b1ik8izxxt  INFO:root:ref: refs/pull/48/merge
2023-01-08 11:03:35.425 EST  cloudbot-franklin  q8b1ik8izxxt  INFO:root:Commit SHA found: 6595030ab07b4fbb4d40b50a8d4ef505c9c86a49
2023-01-08 11:03:35.425 EST  cloudbot-franklin  q8b1ik8izxxt  INFO:root:Completed check JSON fields in GH msg.
2023-01-08 11:03:35.425 EST  cloudbot-franklin  q8b1ik8izxxt  INFO:root:Check PR label cloudbot-testing
2023-01-08 11:03:35.964 EST  cloudbot-franklin  q8b1ik8izxxt  INFO:root:Setting label cloudbot-testing on PR 48
2023-01-08 11:03:36.408 EST  cloudbot-franklin  q8b1ik8izxxt  INFO:root:t: <github.MainClass.Github object at 0x3eb69e3513d0>
2023-01-08 11:03:37.047 EST  cloudbot-franklin  q8b1ik8izxxt  INFO:root:Found filename: README.md
2023-01-08 11:03:37.047 EST  cloudbot-franklin  q8b1ik8izxxt  INFO:root:Found filename: docs/images/config.ini.png
2023-01-08 11:03:37.047 EST  cloudbot-franklin  q8b1ik8izxxt  INFO:root:Found filename: docs/slides/workshop-codemash-2023.pdf
2023-01-08 11:03:37.047 EST  cloudbot-franklin  q8b1ik8izxxt  INFO:root:Found filename: docs/slides/workshop-codemash-2023.tex
2023-01-08 11:03:37.047 EST  cloudbot-franklin  q8b1ik8izxxt  INFO:root:Found filename: src/config.ini
2023-01-08 11:03:37.047 EST  cloudbot-franklin  q8b1ik8izxxt  INFO:root:Looking for string in comment: @n0ctilucent
2023-01-08 11:03:37.154 EST  cloudbot-franklin  q8b1ik8izxxt  INFO:root:Adding comment to the commit.
2023-01-08 11:03:37.154 EST  cloudbot-franklin  q8b1ik8izxxt  INFO:root:Cloudbot adding comment on repo devsecfranklin/workshop-codemash-2023 to PR 48
2023-01-08 11:03:38.126 EST  cloudbot-franklin  q8b1ik8izxxt  INFO:root:Finished adding comment to PR 48
2023-01-08 11:03:38.281 EST  cloudbot-franklin  q8b1ik8izxxt  INFO:root:<Response [200]>
2023-01-08 11:03:38.286 EST  cloudbot-franklin  q8b1ik8izxxt  Function execution took 7516 ms, finished with status code: 200
```

Python: config.ini

- The configparser module is used to make customization easier.
- The Cloud Function is expecting us to create a secret named "gh_secret_token".



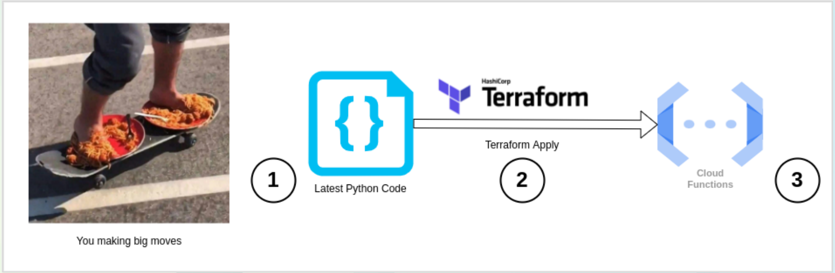
workshop-codemash-2023 > src > config.ini

```ini
[required_options]

# GCP Project ID
project_id = b0t-stuff
# Create this secret in GCP Secret Mgr, holds GH token.
secret_name = gh_secret_token
# this is the label that gets set when a PR is opened.
label_name = cloudbot-testing

```

TERRAFORM

Overview: Terraform aka Deployment

The big picture for deployment.



The Terraform Installer

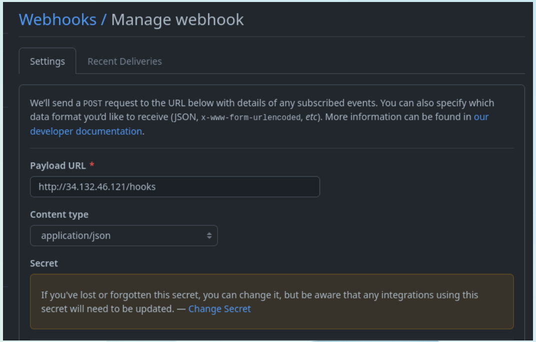
We use Terraform to automate the Cloud Function installation.

Deploying with Terraform

Let's do a Terraform deployment of the Python code to GCP Cloud Function.

Github: Configure Webhook

Configure the webhook in the settings of each repo we want to add our bot to.



Webhooks / Manage webhook

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, `x-www-form-urlencoded`, *etc*). More information can be found in our developer documentation.

Payload URL *

http://34.132.46.121/hooks

Content type

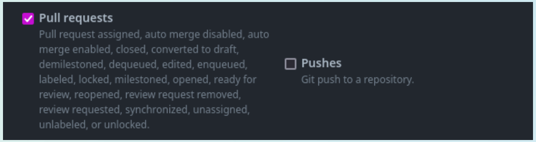
application/json

Secret

If you've lost or forgotten this secret, you can change it, but be aware that any integrations using this secret will need to be updated. — Change Secret

Github: Configure Webhook (cont.)

We will do a custom response, only to this single event.



[✓] **Pull requests**
Pull request assigned, auto merge disabled, auto merge enabled, closed, converted to draft, demilestoned, dequeued, edited, enqueued, labeled, locked, milestoned, opened, ready for review, reopened, review request removed, review requested, synchronized, unassigned, unlabeled, or unlocked.

[ ] **Pushes**
Git push to a repository.

EXTRA

Extra: Connect it to your GKE cluster

- Assuming you already have a GKE cluster, add a VPC connector so the cloud function can talk to the VPC the cluster is in.
- There is YAML in “yaml/cloudbot” that can be used to add a service to an existing GKE cluster.

Extra: GNU Autotools

- Execute the “bootstrap.sh” script from the top level of the repository.
- That should generate the “configure” script and the Makefiles listed in “configure.ac”
- Type “make python” at the top level to build all the python deps. Now you can do “. _build/bin/activate” to get into Python venv.
- You can type “make docs” to build the PDF files from LaTeX.
- The docker directory has separate Makefile, type “make build” and “make push” from that directory.

Extra: Dockerfile and docker-compose.yml

We use Docker to build the container we are working in.

Extra: Github CI Pipeline

Information about what actions we are using and why.

Future: Scan the PR comments for commands

The Cloud Function could monitor the PR for certain strings, using these to trigger actions.