Build a Serverless Github Bot in GCP

Franklin Diaz

DE:AD:10:C5

Tuesday January 10, 2023

INTRODUCTION

Resources

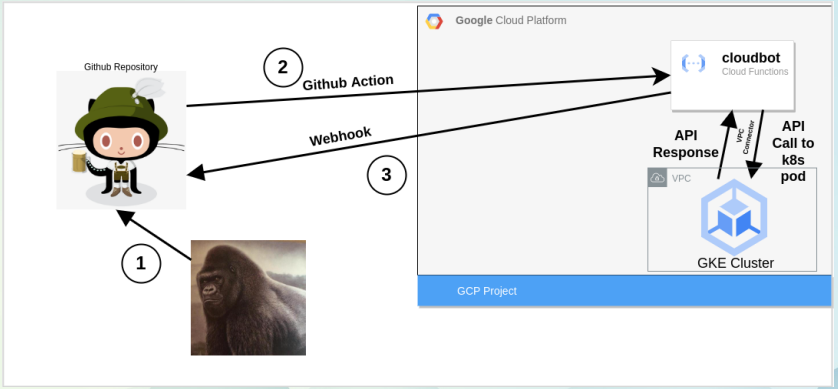
- Click here for Session Details
- Project source files are available: `https://github.com/devsecfranklin/workshop-codemash-2023`
- Prework available at this link.

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

Google Cloud: Update Project Name and Login

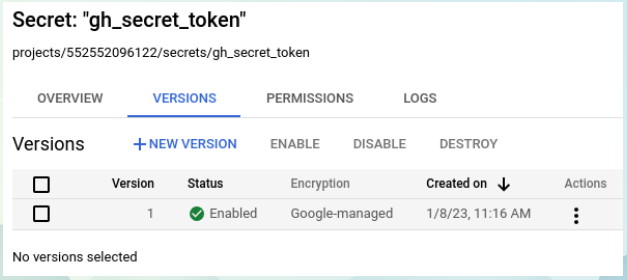
- Update your project name in the file "/workspaces/workshop-codemash-2023/.envrc"
- Update your project name in the file "/workspaces/workshop-codemash-2023/src/config.ini"
- Type the command "direnv allow ." to reload the ENV variables.
- In the dev container, run the command "gcloud auth login" and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

Google Cloud: Create Service User

- We select our root project, we click the IAM & Admin menu, Service Accounts option, and finally, on the + Create Service Account button.

Google Cloud: Create Secret in Secrets Mgr

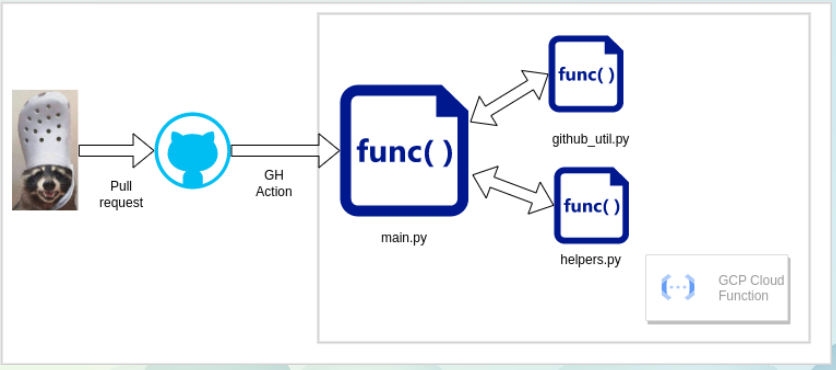
- The Cloud Function is expecting us to create a secret named “gh_secret_token”.
- Enable the Secret Manager service.
- Add the secret.



PYTHON

Overview: Python Functions

The big picture for the Python code files.



The Python Application

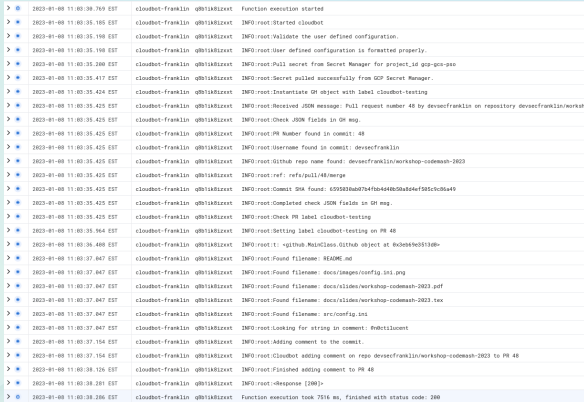
- The main function is essentially a Flask app that waits for an incoming JSON messages.
- Let's take a closer look

```
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.

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



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

```
1  [required_options]
2
3  # GCP Project ID
4  project_id = b0t-stuff
5  # Create this secret in GCP Secret Mgr, holds GH token.
6  secret_name = gh_secret_token
7  # this is the label that gets set when a PR is opened.
8  label_name = cloudbot-testing
9
```

TERRAFORM

Overview: Terraform aka Deployment

The big picture for deployment.

You making big moves

The Terraform Installer

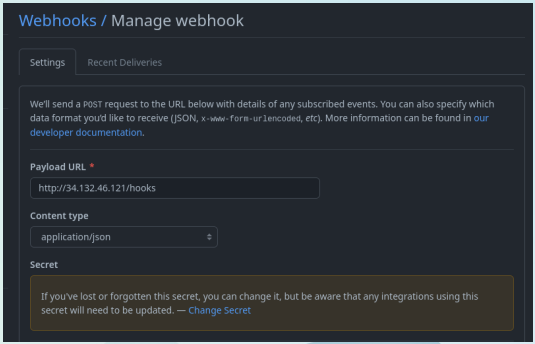
We use Terraform to automate the Cloud Function installation.

Deploying with Terraform

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

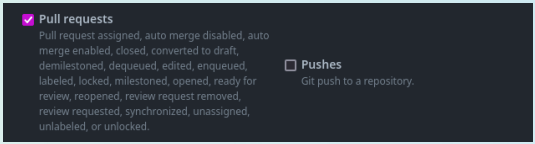
Github: Configure Webhook

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



Github: Configure Webhook (cont.)

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



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.