

# 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>
- Pework available at this link.



## Contact

Github: [devsecfranklin .xXx.](#) E-mail: [devsecfranklin@duck.com](mailto:devsecfranklin@duck.com)

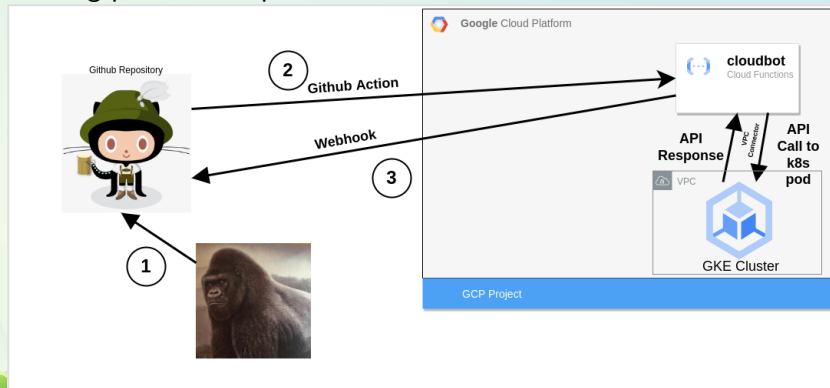


Twitter:  
[@thedeilsvoice](#)

Mastodon:  
[@devsecfranklin@defcon.social](#)

## Overview: Usage

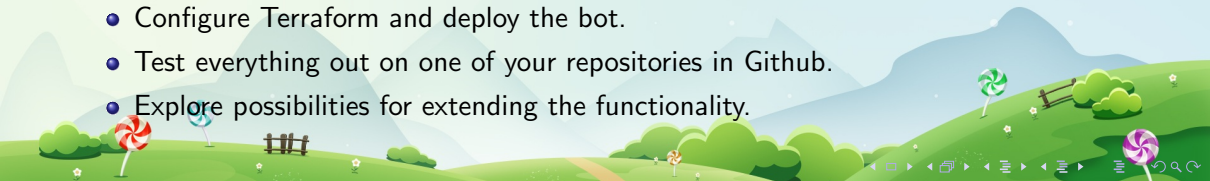
The big picture for operation.



## 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.
- Test everything out on one of your repositories in Github.
- Explore possibilities for extending the functionality.



# PRE-WORK

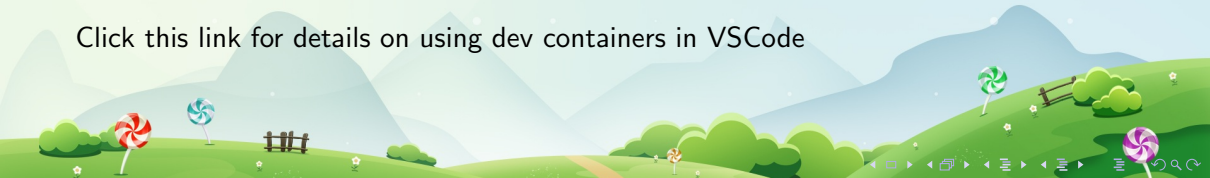


## Setup: VSCode

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 information 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 create a service user in GCP with limited scope of permissions.



## 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

OVERVIEW **VERSIONS** PERMISSIONS LOGS

Versions [+ NEW VERSION](#) ENABLE DISABLE DESTROY

<input type="checkbox"/>	Version	Status	Encryption	Created on ↓	Actions
<input type="checkbox"/>	1	✓ Enabled	Google-managed	1/8/23, 11:16 AM	⋮

No versions selected

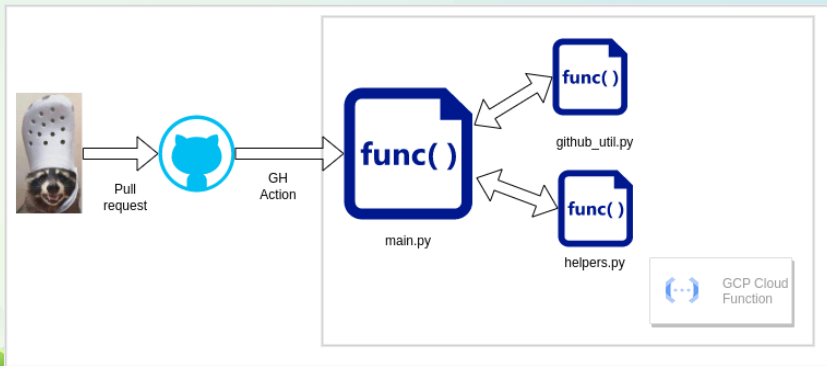


# PYTHON



# 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.

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

```
> 2023-01-08 11:03:38.769 EST cloudbot-franklin qbt1a8l2xxt Function execution started
> 2023-01-08 11:03:38.785 EST cloudbot-franklin qbt1a8l2xxt INFO:root:Started cloudbot
> 2023-01-08 11:03:39.198 EST cloudbot-franklin qbt1a8l2xxt INFO:root:Validate the user defined configuration.
> 2023-01-08 11:03:39.168 EST cloudbot-franklin qbt1a8l2xxt INFO:root:User defined configuration is formatted properly.
> 2023-01-08 11:03:39.280 EST cloudbot-franklin qbt1a8l2xxt INFO:root:Pull secret from Secret Manager for project_id gcp-gcp-gcp
> 2023-01-08 11:03:39.417 EST cloudbot-franklin qbt1a8l2xxt INFO:root:Secret pulled successfully from GCP Secret Manager.
> 2023-01-08 11:03:39.424 EST cloudbot-franklin qbt1a8l2xxt INFO:root:Instantiate GH object with label cloudbot-testing
> 2023-01-08 11:03:39.425 EST cloudbot-franklin qbt1a8l2xxt INFO:root:Received JSON message: Pull request number 48 by devsecfranklin on repository devsecfranklin/workshop
> 2023-01-08 11:03:39.425 EST cloudbot-franklin qbt1a8l2xxt INFO:root:Check JSON fields in GH msg.
> 2023-01-08 11:03:39.425 EST cloudbot-franklin qbt1a8l2xxt INFO:root:PR Number found in commit: 48
> 2023-01-08 11:03:39.425 EST cloudbot-franklin qbt1a8l2xxt INFO:root:Username found in commit: devsecfranklin
> 2023-01-08 11:03:39.425 EST cloudbot-franklin qbt1a8l2xxt INFO:root:Github repo name found: devsecfranklin/workshop-codemash-2023
> 2023-01-08 11:03:39.425 EST cloudbot-franklin qbt1a8l2xxt INFO:root:ref: refs/pull/48/merge
> 2023-01-08 11:03:39.425 EST cloudbot-franklin qbt1a8l2xxt INFO:root:Commit SHA found: 659583ba0704f0b440b58a5d4ef585c0d5a49
> 2023-01-08 11:03:39.425 EST cloudbot-franklin qbt1a8l2xxt INFO:root:Completed check JSON fields in GH msg.
> 2023-01-08 11:03:39.425 EST cloudbot-franklin qbt1a8l2xxt INFO:root:Check PR label cloudbot-testing
> 2023-01-08 11:03:39.964 EST cloudbot-franklin qbt1a8l2xxt INFO:root:Setting label cloudbot-testing on PR 48
> 2023-01-08 11:03:39.488 EST cloudbot-franklin qbt1a8l2xxt INFO:root:1: <github.MainClass.Github object at 0x3eb85e5513d0>
> 2023-01-08 11:03:37.847 EST cloudbot-franklin qbt1a8l2xxt INFO:root:Found filename: README.md
> 2023-01-08 11:03:37.847 EST cloudbot-franklin qbt1a8l2xxt INFO:root:Found filename: docs/images/config.png
> 2023-01-08 11:03:37.847 EST cloudbot-franklin qbt1a8l2xxt INFO:root:Found filename: docs/slides/workshop-codemash-2023.pdf
> 2023-01-08 11:03:37.847 EST cloudbot-franklin qbt1a8l2xxt INFO:root:Found filename: docs/slides/workshop-codemash-2023.tex
> 2023-01-08 11:03:37.847 EST cloudbot-franklin qbt1a8l2xxt INFO:root:Found filename: src/config.ini
> 2023-01-08 11:03:37.847 EST cloudbot-franklin qbt1a8l2xxt INFO:root:Looking for string in comment: @bottilicunt
> 2023-01-08 11:03:37.154 EST cloudbot-franklin qbt1a8l2xxt INFO:root:Adding comment to the commit.
> 2023-01-08 11:03:37.154 EST cloudbot-franklin qbt1a8l2xxt INFO:root:Cloudbot adding comment on repo devsecfranklin/workshop-codemash-2023 to PR 48
> 2023-01-08 11:03:38.126 EST cloudbot-franklin qbt1a8l2xxt INFO:root:Finished adding comment to PR 48
> 2023-01-08 11:03:38.281 EST cloudbot-franklin qbt1a8l2xxt INFO:root:Response [200]
> 2023-01-08 11:03:38.286 EST cloudbot-franklin qbt1a8l2xxt 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
```

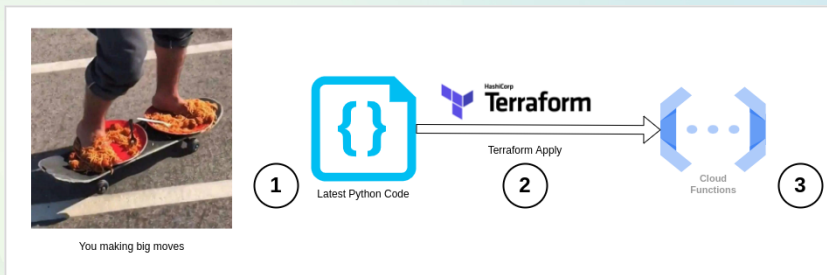
```
1  [required_options]
2
3  # GCP Project ID
4  project_id = bot-stuff
5  # Create this secret in GCP Secret Mgr, holds GH token.
6  secret_name = gh_secret_token
7  # this is the label that gets set when a PR is opened.
8  label_name = cloudbot-testing
9
```

# TERRAFORM



# Overview: Terraform aka Deployment

The big picture for deployment.



# The Terraform Installer

We use Terraform to automate the Cloud Function installation.





# Deploying with Terraform

Let's do a Terraform deployment.



# EXTRA



## Extra: Dockerfile and docker-compose.yml

Check out the docker container and framework, see how all that works.



## Extra: Connect it to your GKE cluster

I can demo this or we can try it if we have time.



## Extra: GNU Autotools

Wow we must be super bored let's play with GNU Autotools.



## 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.

