

CI/CD for GCP with Github



dukerspace. 22.02.2021

CI/CD

Continuous Integration(CI)

กระบวนการรวม source code ของคนในทีมพัฒนาเข้าด้วยกัน และมีการ test ด้วย test script

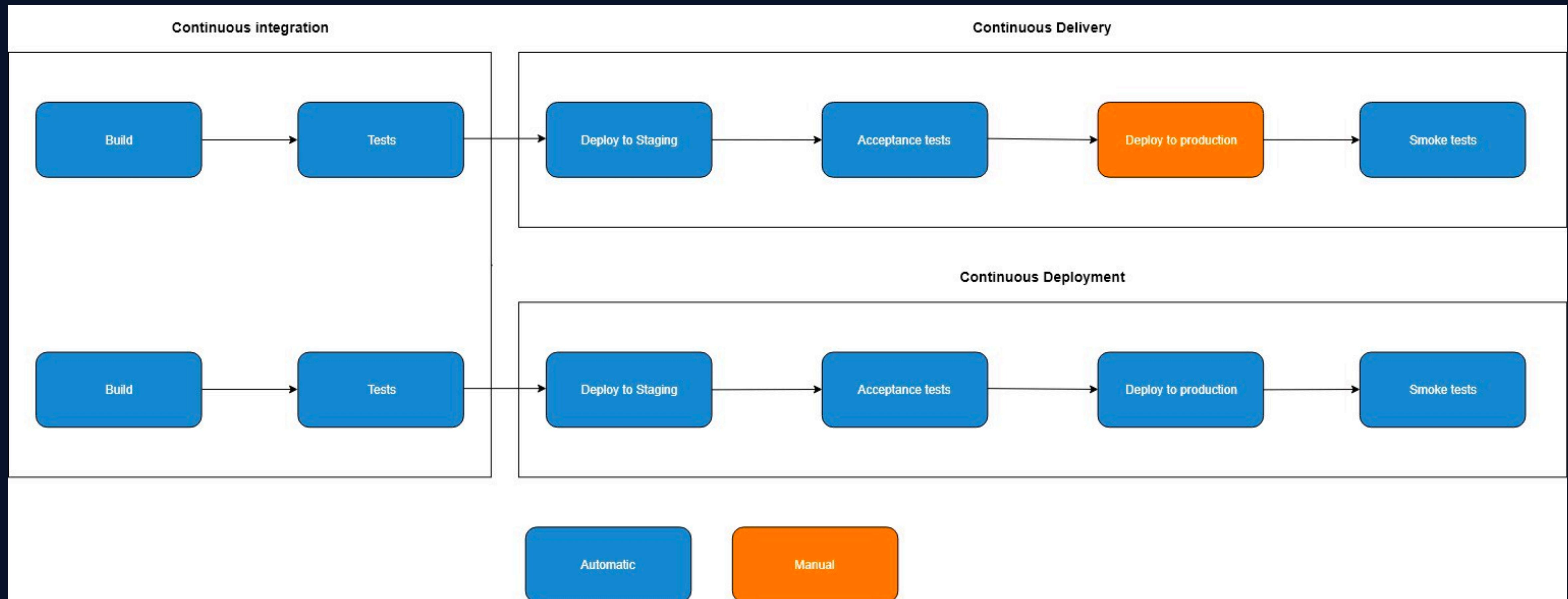
Continuous Delivery และ Continuous Deployment (CD)

Continuous Deployment จะทำทุกขั้นตอน ตั้งแต่ compile build ไปจนถึง deploy ขึ้น production แบบอัตโนมัติทั้งหมด

Continuous Delivery จะทำทุกขั้นตอน ตั้งแต่ compile build ไปจนถึง deploy ขึ้น production แบบอัตโนมัติทั้งหมด

Ref : [link](#)

CI/CD



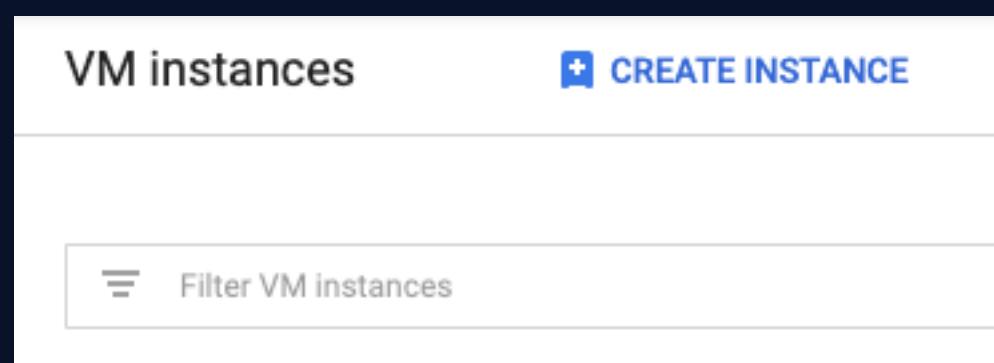
Ref : [link](#)

Set Up

- GitHub Action
- GitHub Workflows
- GitHub Secrets
- GCP Compute Engine
- GitHub SQL
- GitHub Container Registry
- GitHub Cloud Build
- GitHub Cloud Run
- VPC network
- IAM & Admin (Service Accounts)

GCP Compute Engine

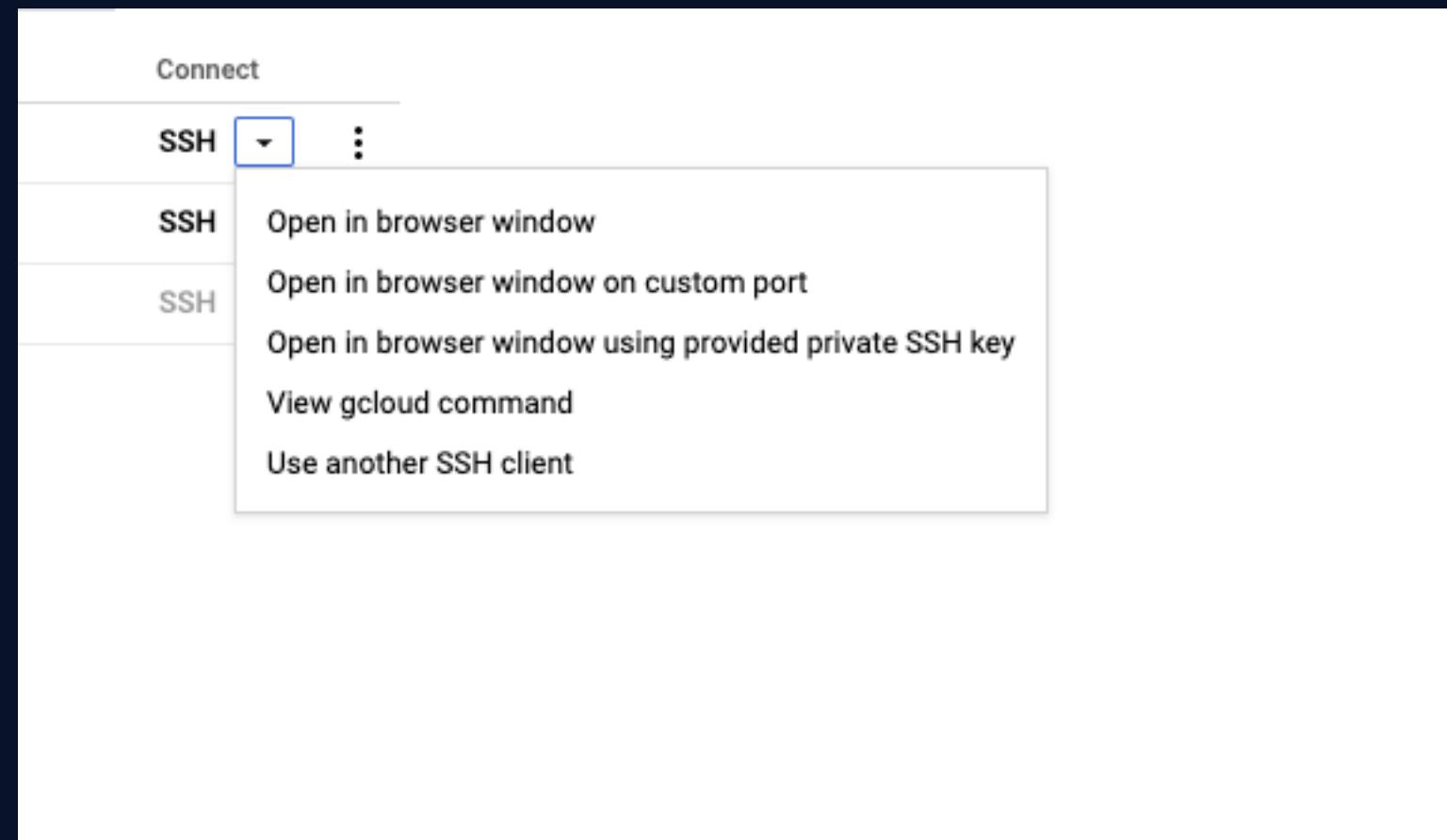
Create VM instances



Select specs

GCP Compute Engine

Login VM instance



The screenshot shows a terminal session on a GCP Compute Engine instance. The title bar indicates the session is connected to 'ssh.cloud.google.com/projects...' and the user is 'montol@github-action-artisan: ~'. The terminal displays system information, including documentation links, management, support, and system load. It also shows memory usage, swap usage, and network information. A note about MicroK8s is present, along with a link to its documentation. The terminal then lists 55 updates available for installation, with 0 being security updates. It ends with a message about a system restart required and the last login details.

```
montol@github-action-artisan:~$ ssh.cloud.google.com/projects...
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

System information as of Thu Feb 18 07:37:51 UTC 2021

System load: 0.0          Processes: 120
Usage of /: 31.7% of 19.21GB  Users logged in: 0
Memory usage: 19%          IPv4 address for ens4: [REDACTED]
Swap usage: 0%

* Introducing self-healing high availability clusters in MicroK8s.
  Simple, hardened, Kubernetes for production, from RaspberryPi to DC.

https://microk8s.io/high-availability

55 updates can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable

*** System restart required ***
Last login: Mon Jan 25 09:10:58 2021 from [REDACTED]
montol@github-action-artisan:~$
```

GitHub Action

Actions / Add self-hosted runner

Adding a self-hosted runner requires that you download, configure, and execute the GitHub Actions Runner. By downloading and configuring the GitHub Actions Runner, you agree to the [GitHub Terms of Service](#) or [GitHub Corporate Terms of Service](#), as applicable.

Operating System: Linux ▾ Architecture: X64 ▾

Download

```
# Create a folder
$ mkdir actions-runner && cd actions-runner
# Download the latest runner package
$ curl -O -L https://github.com/actions/runner/releases/download/v2.277.1/actions-runner-linux-x64-2.277.1.tar.gz
# Extract the installer
$ tar xzf ./actions-runner-linux-x64-2.277.1.tar.gz
```

Configure

```
# Create the runner and start the configuration experience
$ ./config.sh --url https://github.com/artisan-digital-asia --token
AQ[REDACTED]
# Last step, run it!
$ ./run.sh
```

Using your self-hosted runner

```
# Use this YAML in your workflow file for each job
runs-on: self-hosted
```

For additional details about configuring, running, or shutting down the runner, please check out our [product docs](#).

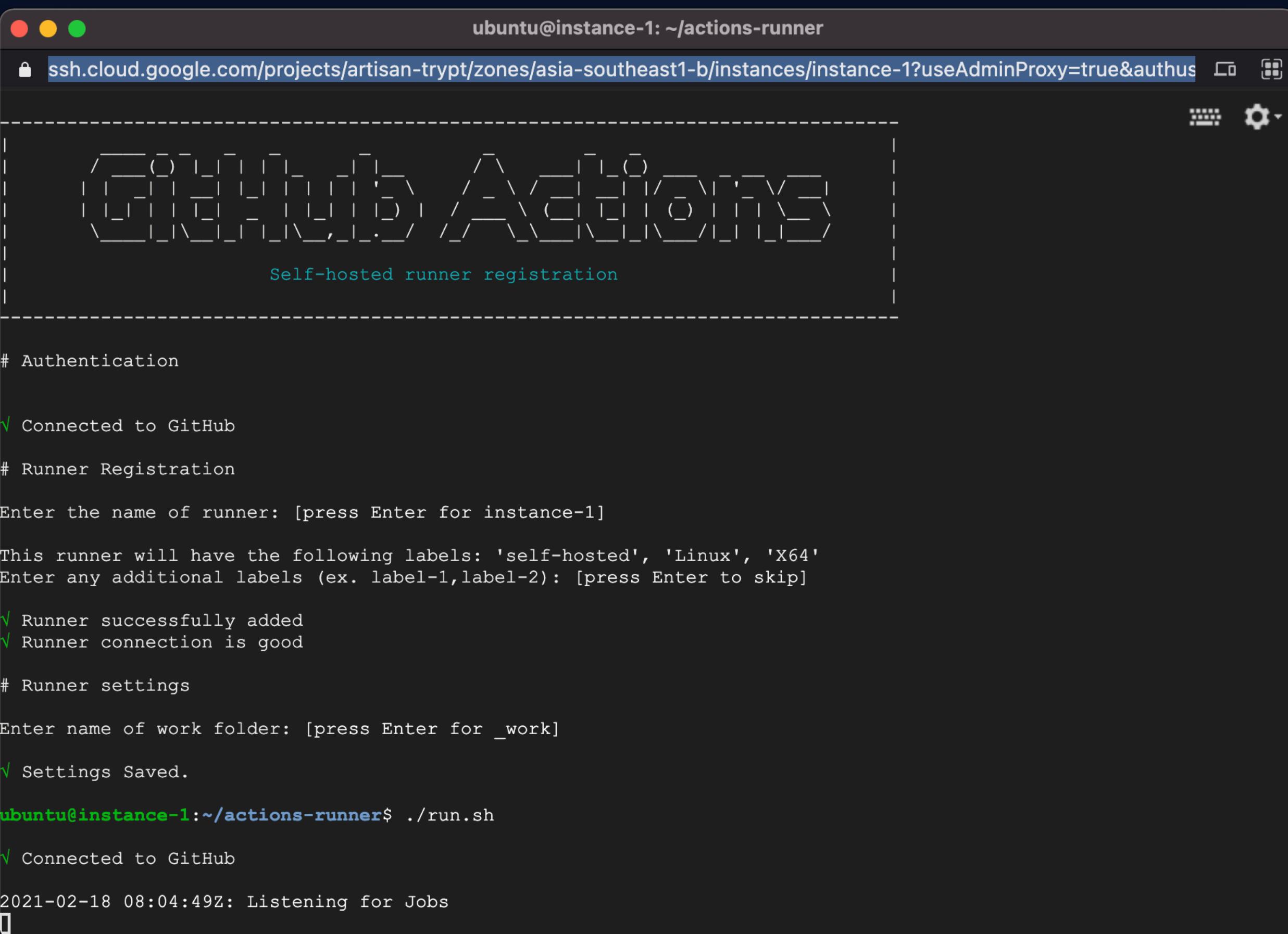
[Back to self-hosted runners listing](#)

Login ubuntu

sudo -u ubuntu -s

Copy command add self-hosted runner เพื่อ install action runners

GitHub Action



A screenshot of a terminal window titled "ubuntu@instance-1: ~/actions-runner". The window shows the process of registering a self-hosted GitHub Action runner on a Google Cloud instance. The terminal output includes:

- A decorative ASCII art banner for "Self-hosted runner registration".
- The command "# Authentication" followed by "Connected to GitHub".
- The command "# Runner Registration" followed by prompts for the runner name ("Enter the name of runner: [press Enter for instance-1]"), labels ("This runner will have the following labels: 'self-hosted', 'Linux', 'X64' Enter any additional labels (ex. label-1,label-2): [press Enter to skip]"), and success messages ("Runner successfully added", "Runner connection is good").
- The command "# Runner settings" followed by a work folder selection prompt ("Enter name of work folder: [press Enter for _work]"), a settings save confirmation ("Settings Saved."), and the command "ubuntu@instance-1:~/actions-runner\$./run.sh".
- The final message "Connected to GitHub" and the timestamp "2021-02-18 08:04:49Z: Listening for Jobs".

GitHub Action

Runner groups

- Default** ⓘ
All repositories, excluding public repositories 3 runners ...
- [REDACTED]
self-hosted Linux X64 [REDACTED] Idle ...
- github-action-[REDACTED]**
self-hosted Linux X64 github-action-[REDACTED] Active ...
- instance-1**
self-hosted Linux X64 [REDACTED] Idle ...

GitHub Action

Installing the service

```
sudo ./svc.sh install
```

Starting the service

```
sudo ./svc.sh start
```

Checking the status of the service

```
sudo ./svc.sh status
```

Ref : [Link](#)

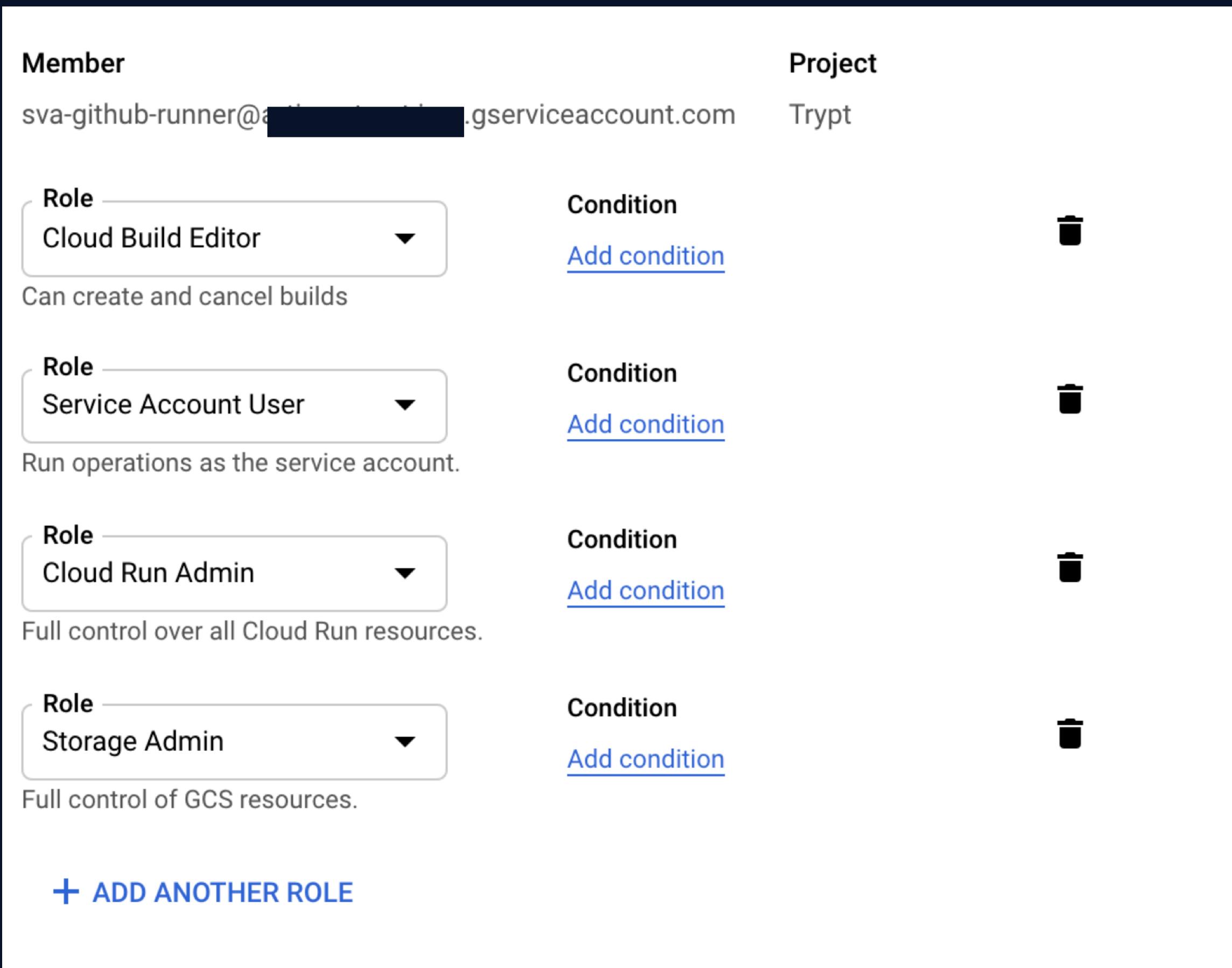
GCP Compute Engine

Member
sva-github-runner@[REDACTED].gserviceaccount.com

Project
Trypt

Role	Condition	Condition	Condition	Condition
Cloud Build Editor	Add condition			
Service Account User	Add condition			
Cloud Run Admin	Add condition			
Storage Admin	Add condition			

+ ADD ANOTHER ROLE



Add the the following Cloud IAM roles to your service account:

- Cloud Run Admin - allows for the creation of new Cloud Run services
- Service Account User - required to deploy to Cloud Run as service account
- Storage Admin - allow push to Google Container Registry
- Cloud Build Editor - allow create build to Google Cloud Build

GCP Compute Engine

Download a JSON service account key for the service account.

Create private key for "sva-github-runner"

Downloads a file that contains the private key. Store the file securely because this key can't be recovered if lost.

Key type

JSON
Recommended

P12
For backward compatibility with code using the P12 format

CANCEL **CREATE**

GitHub Action

Create actions secrets

Actions secrets

New repository secret

Secrets are environment variables that are **encrypted**. Anyone with **collaborator** access to this repository can use these secrets for Actions.

Secrets are not passed to workflows that are triggered by a pull request from a fork. [Learn more](#).

Environment secrets

There are no secrets for this repository's environments.

Encrypted environment secrets allow you to store sensitive information, such as access tokens, in your repository environments.

[Manage your environments and add environment secrets](#)

Repository secrets

There are no secrets for this repository.

Encrypted secrets allow you to store sensitive information, such as access tokens, in your repository.

Download & copy secret key from services account .json

Actions secrets / New secret

Name

SA_GCP_GITHUB_RUNNER

Value

```
{  
  "type": "service_account",  
  "path": "/path/to/service-account-file.json"  
}
```

Add secret

Start Project

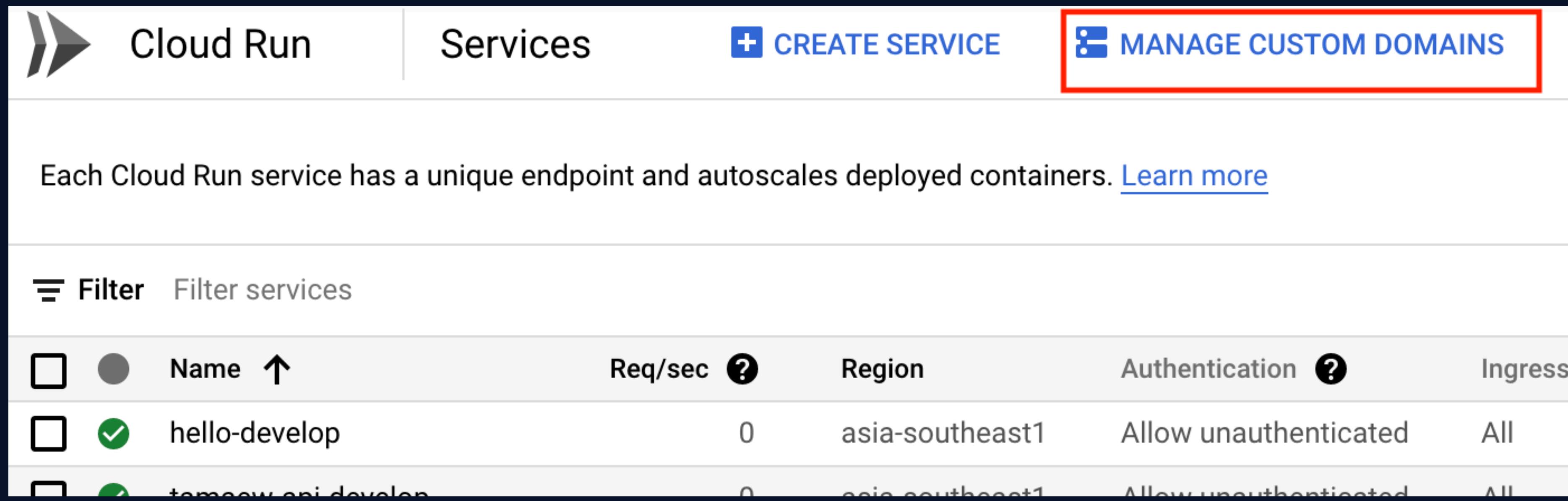
```
git clone https://github.com/dukerspace/hello-ci-cd
```

GitHub Workflows

```
env:  
  PROJECT_ID: ci-cd-artisan  
  BRANCH_NAME: develop  
  RUN_REGION: asia-southeast1  
  SERVICE_NAME: hello-duker  
  
jobs:  
  build:  
    runs-on: self-hosted  
    steps:  
      - uses: actions/checkout@v2  
      - uses: google-github-actions/setup-gcloud@master  
        with:  
          project_id: ${{ env.PROJECT_ID }}  
          service_account_key: ${{ secrets.RUNNER }}  
  
      - name: Build  
        run:  
          gcloud builds submit \  
          --tag gcr.io/$PROJECT_ID/$SERVICE_NAME:$BRANCH_NAME  
  
      - name: Deploy  
        run:  
          gcloud run deploy $SERVICE_NAME-$BRANCH_NAME \  
          --quiet \  
          --region $RUN_REGION \  
          --image gcr.io/$PROJECT_ID/$SERVICE_NAME:$BRANCH_NAME \  
          --platform managed \  
          --allow-unauthenticated
```

GCP Cloud Run

Create custom domain



The screenshot shows the GCP Cloud Run Services interface. At the top, there's a navigation bar with the Cloud Run logo, the word "Services", a "CREATE SERVICE" button, and a "MANAGE CUSTOM DOMAINS" button, which is highlighted with a red box. Below the navigation bar, a message states: "Each Cloud Run service has a unique endpoint and autoscales deployed containers. [Learn more](#)". There's also a "Filter services" section with a dropdown menu set to "Name ↑". The main table lists two services:

	Name	Req/sec	Region	Authentication	Ingress
<input type="checkbox"/>	<input checked="" type="radio"/> hello-develop	0	asia-southeast1	Allow unauthenticated	All
<input type="checkbox"/>	<input checked="" type="radio"/> temow/api-develop	0	asia-southeast1	Allow unauthenticated	All

GCP Cloud Run

Cloud Run

Domain mappings

+ ADD MAPPING

Filter domains

Domain Type Mapped to ↑

GCP Cloud Run

Add mapping BETA

You can map domains and subdomains to the selected Cloud Run service. [Learn more](#)

1 Select or enter domain — Verify — 3 Update DNS records

Select a service to map to *
hello-develop (asia-southeast1)

Select a verified domain
artisandigital.tech

Specify subdomain
https:// hello

Leave blank to map the base domain

CANCEL [CONTINUE](#)

<https://hello.artisandigital.tech>

GCP Cloud Run

Add mapping BETA

You can map domains and subdomains to the selected Cloud Run service. [Learn more](#)

✓ Select or enter domain — ✓ Verify — 3 Update DNS records

Update the DNS records on your domain host with the records below. You can view these again using the "DNS records" button in the domain mappings table. [Learn more](#)

DNS records for hello.artisandigital.tech

Name	Type	Data
hello	CNAME	ghs.googlehosted.com. 

DONE

Copy cname:
[ghs.googlehosted.com](#)

Cloudflare

The screenshot shows the Cloudflare dashboard with a dark theme. At the top, there's a navigation bar with various icons and links: Overview, Analytics, DNS, SSL/TLS, Firewall, Access, Speed, Caching, Workers, Page Rules, Network, Traffic, Stream, Custom P..., Apps, and Scrape S... Below this is a message box with a list of setup steps:

A few more steps are required to complete your setup.

- ✓ Add an A, AAAA, or CNAME record for **www** so that [www.artisandigital.tech](#) will resolve.
- ✓ Add an MX record for your **root domain** so that mail can reach [@artisandigital.tech](#) addresses.

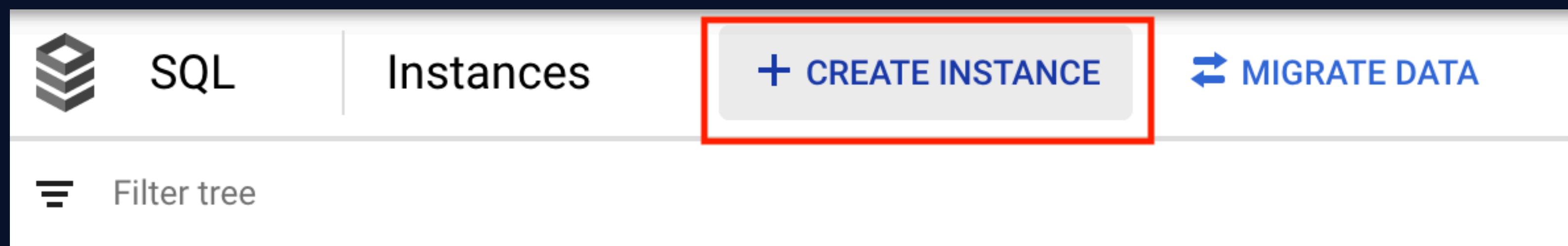
Below the message box is a section titled "DNS management for **artisandigital.tech**". It includes a search bar with "+ Add record" and "Advanced" buttons. A note states: "hello.artisandigital.tech is an alias of [ghs.googlehosted.com](#)". The main form for adding a CNAME record has the following fields:

Type	Name	Target	TTL	Proxy status
CNAME	hello	ghs.googlehosted.com	Auto	DNS only

At the bottom right are "Cancel" and "Save" buttons.

Config cloud flare
map domain to gcp

GCP SQL



GCP SQL

SQL [!\[\]\(7750f6b14c60f968a2bc65812f3f92db_img.jpg\) Create an instance](#)

Choose your database engine

 MySQL
Versions: 5.6, 5.7, 8.0
[→ Choose MySQL](#)

 PostgreSQL
Versions: 9.6, 10, 11, 12, 13
[→ Choose PostgreSQL](#)

 SQL Server
Versions: 2017
[→ Choose SQL Server](#)

Want more context on the Cloud SQL database engines? [Learn more](#)

GCP SQL

 SQL | [Create a PostgreSQL instance](#)

Instance info

Instance ID
Choice is permanent. Use lowercase letters, numbers, and hyphens. Start with a letter.

Default user password
Set a password for the 'postgres' user. A password is required for the user to log in.
[Learn more](#)
 [Generate](#)

Location ⓘ
For better performance, keep your data close to the services that need it.

Region Choice is permanent	Zone Can be changed at any time
<input type="text" value="asia-southeast1 (Singapore)"/>	<input type="text" value="asia-southeast1-b"/>

Database version

[Show configuration options](#)

[Create](#) [Cancel](#)

GCP Cloud SQL Proxy

<https://cloud.google.com/sql/docs/postgres/sql-proxy>

<https://cloud.google.com/sql/docs/mysql/connect-admin-proxy>

chmod +x cloud_sql_proxy (Mac OS)

```
./cloud_sql_proxy -instances=ci-cd-artisan:asia-southeast1:instance-db-duker=tcp:5432
```

GCP VPC network

Serverless VPC access

+ CREATE CONNECTOR DELETE

Serverless VPC Access allows Cloud Functions, Cloud Run (fully managed) services and App Engine standard environment apps to access resources in a VPC network using those resources' private IPs. [Learn more](#)

Filter Filter table

<input type="checkbox"/>	Name ↑	Network	Subnet	Region
<input type="checkbox"/>	connector-tan	default		asia-southeast1
<input type="checkbox"/>	duker-netowork	default		asia-southeast1
<input type="checkbox"/>	tiw-network	default		asia-southeast1

GCP VPC network

[←](#) Create connector

Name *

Region * asia-southeast2

A region is a specific geographical location where you can run your resources.

Network *

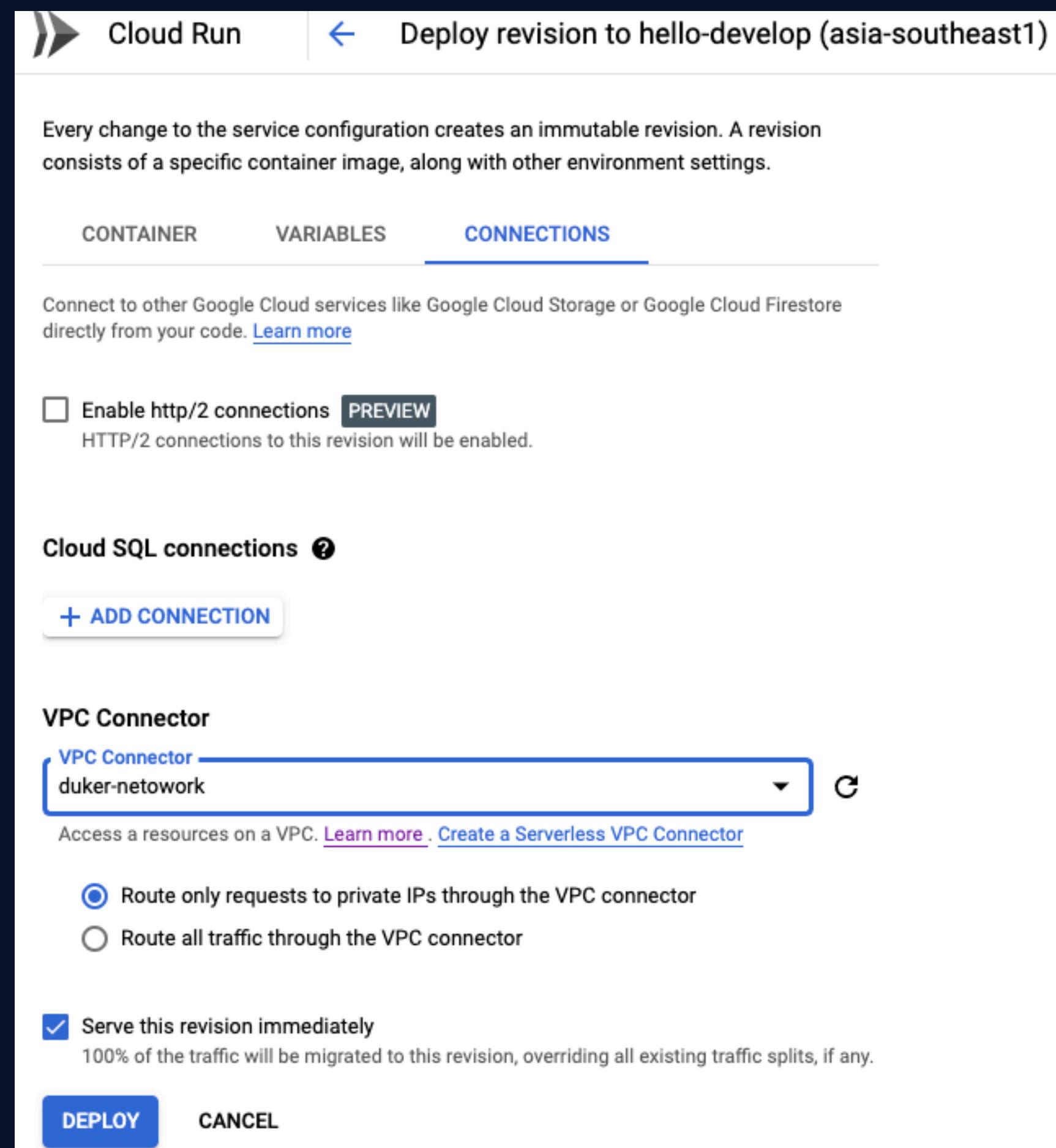
IP range * /28

IP range must be an unused /28 CIDR range in your VPC network, such as 10.8.0.0/28. The VPC Connector will create connector instances on IP addresses in this range. Ensure the range does not overlap with an existing subnet. [Learn more](#)

[▼ SHOW SCALING SETTINGS](#)

[CREATE](#) [CANCEL](#)

GCP VPC network



Release

Releases Tags

v1.0.0 @ Target: main ▾

Excellent! This tag will be created from the target when you publish this release.

v1.0.0

Write Preview

Features

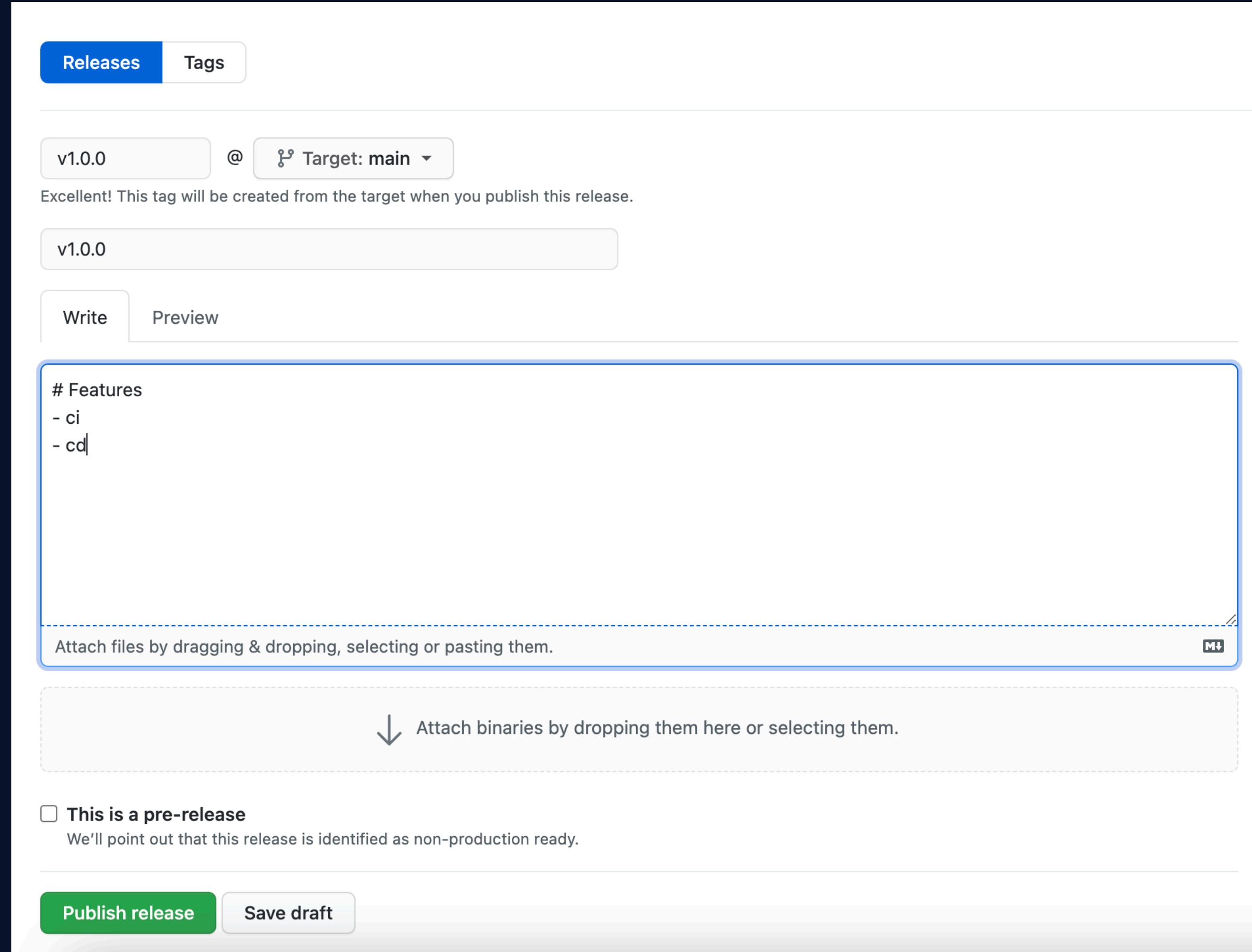
- ci
- cd|

Attach files by dragging & dropping, selecting or pasting them. M+

↓ Attach binaries by dropping them here or selecting them.

This is a pre-release
We'll point out that this release is identified as non-production ready.

Publish release Save draft



End - Jeremy Zucker

