

Setup Github Action to Build, Push Docker Image to GCR and Deploy to Google Compute Engine

Berdasarkan dokumentasi dari Github,

GitHub Actions help you automate your software development workflows in the same place you store code and collaborate on pull requests and issues. You can write individual tasks, called actions, and combine them to create a custom workflow. Workflows are custom automated processes that you can set up in your repository to build, test, package, release, or deploy any code project on GitHub.

(<https://help.github.com/en/actions/getting-started-with-github-actions/about-github-actions>)

Singkatnya, fitur ini membantu kita untuk melakukan automasi aksi lanjutan setelah melakukan push pada repository project kita di github. Bisa merupakan aksi build, push, test, bahkan deploy keserver serta lainnya.

Kali ini saya akan mencoba untuk mengimplementasikan Github action untuk melakukan proses build, push docker image ke google cloud registry, serta mendeploy docker image tersebut ke Instance VM yang ada di Google Cloud. Aplikasi yang saya gunakan berjalan diatas Node JS & Docker, serta NGINX sebagai webservernya.

Prerequisite:

- Memiliki akun github
- Memiliki akun GCP (billing enabled)

Step by Step:

- Create sebuah VM Instance di GCP, store public key ke VM sehingga bisa diakses nantinya, install docker engine & docker-compose (<https://docs.docker.com/engine/install/>)
- Enable Google Container Registry API
- Download key service-account,
Klik IAM & Admin -> Service Account -> pilih service account -> create key (json)
- Encode Service account, setelah kita mendapatkan file SA dalam format json kita harus melakukan encode base64 sebelum dimasukkan ke secret Github
Bisa dengan cara : `cat 'namafilename.json' | base64 >> 'newfile.json'`
- Setup secret di github project.
Klik Settings -> Secrets -> New Secret

Options
Manage access
Security & analysis
Branches
Webhooks
Notifications
Integrations
Deploy keys
Secrets
Actions

Name

GCR_KEY

Value

uFrp1yO4upSh8zc5NwwA+cLcGT5k86X4GG/3+JEpItuaHAVZ79zmK/3w1f5KsB
NcX20WnFXk2OiwvF1fJzJfYp5bJ6EcVbHClA/mpwb52xedpQvV/OCCjz9Q3t
2cN7AAeCgYEA612j92uGuSVnUmwDv+A5pdfdl+W9lW543J2bzS5pseewPFQgCB3G
e2hrFsr4Wwnieir1WygJupUrh3i3GEoyhwRpFeaTSuxTlCi1xmdqW2E4YPvRN
xuj9g8a+NvNkCDwqZWNSIna21B92UFZ/ONrZgtO7n9RTPNKC3TAGz/cCgYEA3Zxd
d9Q87dSAiVLdnUm5ZpbzvfALZR1/NghIjq9HJWfc6S4/oAV7mfdt62InKn/BHUM
VP28R9D8wXfZBRInH31qdO4uEPlJAZ2vOZl6lKcPMgOY6sqpccK+ZzhReWp
CtHLqadCnxx1Z5x9Nl9ngfw57mTnxZ9A0Q2iUCuqYB5QdpSlIC7KCsSLwFQxww
pa9Qq1fuiS32aXNOfm746aTO1CltGmz5aEVkuay77qBQX24r5YHDrOCF9E0xlv9

Add secret

- Tambahkan juga new secret private key, yang nantinya bertujuan untuk melakukan remote ssh ke VM.

> Code

Pull requests 0

Actions

Projects 0

Security 0

Insights

Settings

Options

Manage access

Security & analysis

Branches

Webhooks

Notifications

Integrations

Deploy keys

Secrets

Actions

Secrets

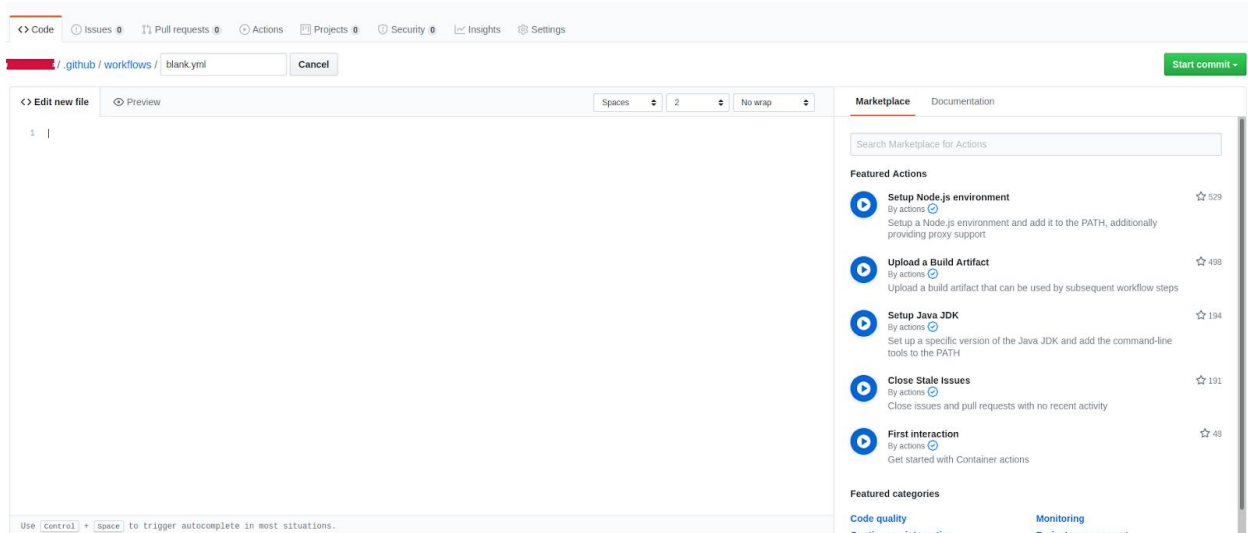
New secret

Secrets are environment variables that are **encrypted** and only exposed to selected actions. Anyone with **collaborator** access to this repository can use these secrets in a workflow.

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

GCR_KEY	Updated 16 hours ago	Update	Remove
RSA_KEY	Updated 15 hours ago	Update	Remove

- Setelah itu kita akan menambahkan actions pada github project
Klik Actions -> Set up this workflow



- Isikan blank.yml seperti code dibawah ini

```
name: CICD with Github Actions

on:
  push:
    branches: [ master ]

env:
  GITHUB_SHA: ${ github.sha }
  GITHUB_REF: ${ github.ref }
  IMAGE: name-of-image
  REGISTRY_HOSTNAME: gcr.io/your-gcr-name

jobs:
  setup-build-publish-deploy:
    name: Setup, Build, and Publish
    runs-on: ubuntu-latest
    steps:

      - name: Checkout
        uses: actions/checkout@v2

      # Setup gcloud CLI
      - uses: GoogleCloudPlatform/github-actions/setup-gcloud@master
        with:
          version: '270.0.0'
          service_account_key: ${ secrets.GCR_KEY }
```

```

# Configure docker to use the gcloud command-line tool as a
credential helper
- run: |
    # Set up docker to authenticate
    # via gcloud command-line tool.
    gcloud auth configure-docker

# Build the Docker image
- name: Build Docker image
  run: |
    export TAG=`echo $GITHUB_REF | awk -F/ '{print $NF}'`
    echo $TAG
    cd app
    docker build -t "$REGISTRY_HOSTNAME"/"$IMAGE": "$TAG" \
      --build-arg GITHUB_SHA="$GITHUB_SHA" \
      --build-arg GITHUB_REF="$GITHUB_REF" .

# Push the Docker image to Google Container Registry
- name: Push to GCR
  run: |
    export TAG=`echo $GITHUB_REF | awk -F/ '{print $NF}'`
    echo $TAG
    docker push "$REGISTRY_HOSTNAME"/"$IMAGE": "$TAG"
    docker tag "$REGISTRY_HOSTNAME"/"$IMAGE": "$TAG"
"$REGISTRY_HOSTNAME"/"$IMAGE":latest
    docker push "$REGISTRY_HOSTNAME"/"$IMAGE":latest

# Deploy Application to Server
- name: Deploy to Server
  uses: fifsky/ssh-action@master
  with:
    command: |
      cd projectFolder
      docker-compose pull
      docker-compose up --force-recreate -d
    host: 'host / IP addr'
    user: root
    key: ${ secrets.RSA_KEY }

```

Kita akan bahas sedikit code diatas. Jadi actions diatas akan dibagi menjadi 3 bagian besar yaitu **on:** , **env:** , **jobs:**

on: adalah fungsi check kondisi trigger mana yang akan menjadikan actions itu berjalan atau tidak. Kita dapat melakukan kustomisasi dibagian ini, sebagai contohnya. Jika kita melakukan push ke master, maka actions berjalan. Atau kita ingin actions berjalan jika ada tag v, kita bisa buat seperti ini

```
on:
  push:
    tags:
      - v*
```

env: untuk menambahkan environment

jobs: pada bagian ini jobs / workflow akan ditambahkan. Kita juga dapat menggunakan actions/plugins yang telah dikembangkan oleh orang lain dengan menambahkannya dengan **uses:** , hal ini dapat membantu kita mempercepat dan mempermudah jika ada beberapa case yang butuh custom jobs.

- Jika sudah, Commit action tersebut.
- Untuk melakukan monitoring action anda dapat melihatnya di tab Actions kemudian pilih commit yang ingin dilihat lebih detail

The screenshot shows the GitHub Actions interface. The top navigation bar includes 'Code', 'Pull requests', 'Actions', 'Projects', 'Security', 'Insights', and 'Settings'. The 'Actions' tab is active, showing a workflow named 'Setup, Build, Publish and Deploy' with a status of 'succeeded 15 hours ago in 2m 24s'. The workflow steps are listed on the right, including 'Set up job', 'Build ftky/sh-action@master', 'Checkout', 'Run GoogleCloudPlatform/github-actions/setup-gcloud@master', 'Run # Set up docker to authenticate', 'Build Docker image', 'Push to GCR', 'Deploy to Server', 'Post Checkout', and 'Complete job'. The 'Build Docker image' step is highlighted, showing a duration of 1m 19s.

- Actions berjalan dengan baik. Anda dapat melihat detail execute dari step-step yang ada pada jobs. Pastikan juga docker image telah tersimpan di registry. Kemudian akses aplikasi melalui IP addr / domain.

The screenshot shows the Docker Container Registry interface. The 'Images' tab is active, displaying a list of images. The table has columns for 'Name', 'Tags', 'Created', and 'Uploaded'. The images listed are:

Name	Tags	Created	Uploaded
364ec3d70c4e	latest, master	43 minutes ago	43 minutes ago
f207e8952be	—	2 hours ago	2 hours ago
5d9dcf4d5d4f	—	15 hours ago	15 hours ago
712014d4d4f5	—	15 hours ago	15 hours ago
98d4ef122f18	—	16 hours ago	16 hours ago

Referensi :

- <http://acaird.github.io/computers/2020/02/11/github-google-container-cloud-run>
- <https://help.github.com/en/actions/getting-started-with-github-actions/about-github-actions>
- <https://github.com/fifsky/ssh-action>