**Part 1 Create a local demo folder.**

**Step 1**

Download base folder “GitHub\_Actions” from FaaSr/ base\_docker\_images.

Link: <https://github.com/renatof/FaaSr/tree/main/base_docker_images/GitHub_Actions>

**Step 2**

The structure of github action demo folder should look like:

github\_actions\_demo/

--Dockerfile

--app/

-- faasr.R

-- exec.R

-- simple\_sequence1.R

-- FaaSr.schema.json

“faasr.R” can download from <https://github.com/renatof/FaaSr/blob/main/R/faasr.R>

“simple\_sequence1.R” can download from <https://github.com/renatof/FaaSr/blob/main/examples/simple_sequence1.R>

**Part 2 Build docker image and push to Docker hub.**

After creating github\_actions\_demo folder, we build image and push image to Docker hub.

Reference link: <https://docs.docker.com/docker-hub/repos/create/>

**Part 3 Create GitHub Action workflow and setting env variables.**

**Step 1** Create Personal Access Token

To use GitHub REST API to trigger workflow and get payload, we need to create a **personal access token**. Note: please save this token, this is the only time we see the value of this token.

by referencing this link: <https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/managing-your-personal-access-tokens>

Note: if create a **Fine-grained token,** under **Repository access**, we can select “**All repositories**” or “**Only select repositories**”, under **Repository permissions**, select **Actions** (read and write) and **Contents** (read-only or read-and-write).

If create a **classic token,** Under **Select scopes,** we need to select “**repo**” and “**workflow**”.

**Step 2** Create Workflow file.

Create a github repository (for example: demo\_repo)

In this repo, under **Actions,** choose **“set up a workflow yourself**”, copy content of **github\_demo\_workflow.yml** to your workflow file, update the image name in workflow file (your\_dockerhub\_name/your\_dockerhub\_repo:tag)

github\_demo\_workflow.yml link: <https://github.com/renatof/FaaSr/blob/main/examples/github_demo_workflow.yml>

**step 3** Save the demo example\_payload\_for\_gh\_actions.json to one repo.

we can save the json payload in same repo or different repo in your account (for example, payload\_repo)

the format **example\_payload\_for\_gh\_actions.json** file can download from: <https://github.com/renatof/FaaSr/blob/main/examples/example_payload_for_gh_actions.json>

modify the relevant “key:value” pair.

For “computerServers” – “github\_a”, define UserName (github account name), RepoName (the repo that we create github action workflow.yml file), WorkflowName (e.g., workflow.yml), Ref (the workflow branch, main or other branch)

For “DataStores” part, update the S3 bucket information.

**Step 4** Set github action repo secrets and variables.

1. **Set One Variable**

We need to create one **Variables.** In your demo\_repo, Under **Settings—**select **Secrets and Variables—**select **Actions---**select **Variables.**

图形用户界面, 应用程序

描述已自动生成

Create a **New Repository Variables:** PAYLOAD\_REPO (i.e., the path of your json payload)

Format: {username}/{repo}/{path}

For example: ABC/ payload\_repo/payload\_for\_gh\_action.json

1. **Set Three Secrets**

In your demo\_repo,

Under **Settings—**select **Secrets and Variables—**select **Actions---**select **Secrets**

Create three **Secrets**:

1. DOCKERHUB\_USERNAME (your docker hub account name)
2. DOCKERHUB\_SECRET (your docker account secret)
3. SECRET\_PAYLOAD

For the **SECRET\_PAYLOAD**, we will put all secrets which will be used in container to SECRET\_PAYLOAD secret with json format.

For Example: copy following {} to your SECRET\_PAYLOAD secret.

{

    "GITHUB\_lambda\_ACCESS\_KEY":"111111111111111",

    "GITHUB\_lambda\_SECRET\_KEY":"222222222222222222222222",

    "GITHUB\_S3\_ACCESS\_KEY":"your\_aws\_access\_key\_ID",

    "GITHUB\_S3\_SECRET\_KEY":"your\_aws\_secret\_access\_key",

    "GITHUB\_gita\_TOKEN":"your\_Github\_Personal\_Access\_Token",

    "PAYLOAD\_GITHUB\_TOKEN":" your\_Github\_Personal\_Access\_Token"

}

图形用户界面, 文本, 应用程序

描述已自动生成

Figure: SECRET\_PAYLOAD example, replace with your secrets

**Note:**

(1) The key in “**SECRET\_PAYLOAD”** is the value of payload. json

For example, in “example\_payload\_for\_gh\_actions.json”

"ComputeServers":{

            "github\_a":{

                    "FaaSType": "GitHubActions",

                    "Token":" GITHUB\_gita\_TOKEN"

            }

    },

 "DataStores":{

            "S3\_A":{

                    "Endpoint":"my\_s3",

                    "AccessKey":" GITHUB\_S3\_ACCESS\_KEY",

                    "SecretKey":" GITHUB\_S3\_SECRET\_KEY"

If we set github\_a "Token":" GITHUB\_gita\_TOKEN",

then in SECRET\_PAYLOAD, the relevant pair is "GITHUB\_gita\_TOKEN":"1234567" (assume 1234567 is your access token)

(2) Add the **PAYLOAD\_GITHUB\_TOKEN** in SECRETS\_PAYLOAD secret,

We need to include the pair "PAYLOAD\_GITHUB\_TOKEN":"1234567" (assume 1234567 is your access token), this pair will be used to GET your payload.json file when container start to run.

**Part 4 Test on GitHub**

There are two methods --Curl or Github Console --to invoke workflow.

**Note**, we should first upload the "input1.csv" and "input2.csv" to the S3 bucket folder.

1. Use Github Console:

Go to your demo\_repo, under **Actions,** select the workflow, select **Run Workflow,** enter **FunctionInvoke** with **F1**

图形用户界面, 文本, 应用程序, 聊天或短信

描述已自动生成

2. Use Curl:

Open a terminal window, enter command:

curl -L \

-X POST \

-H "Accept: application/vnd.github+json" \

-H "Authorization: Bearer {your\_personal\_access\_token}"\

-H "X-GitHub-Api-Version: 2022-11-28" \

https://api.github.com/repos/{account\_name}/{repo\_name}/actions/workflows/{workflow\_file\_name}/dispatches \

-d '{"ref":"{main\_or\_other\_branch}","inputs":{"InvokeName":"F1\_process"}}'