

COMP 7940 Cloud Computing

2023/24 S2 Lab6: Git Action

ROLE	NAME	EMAIL
Instructor	Dr. Qichen WANG	qcwang@hkbu.edu.hk
Teaching Assistant	Mr. Yu XU	csyuxu@comp.hkbu.edu.hk
Teaching Assistant	Mr. Hongduan TIAN	cshdtian@comp.hkbu.edu.hk

Intended Learning Outcomes

Throughout this lab session, you are expected to:

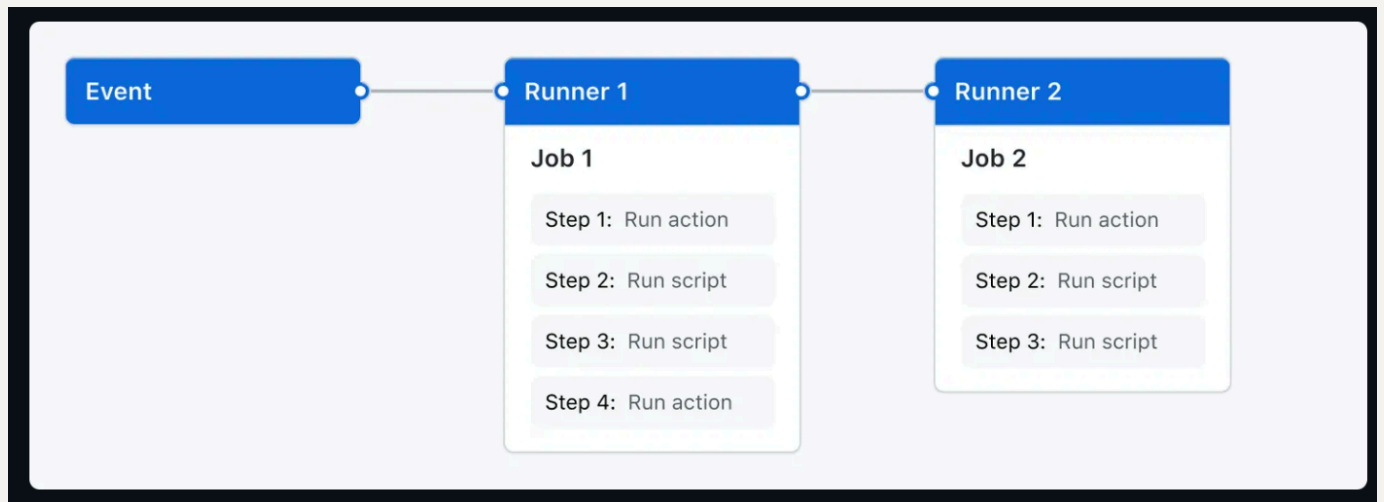
- Learn about git action and workflow;
 - Try to apply git action and workflow to your chatbot.
-

1. Introduction of Git Action

GitHub Actions is a continuous integration and continuous delivery (CI/CD) platform that allows you to automate your build, test, and deployment pipeline. You can create workflows that build and test every pull request to your repository, or deploy merged pull requests to production.

2. Components of GitHub Action

You can configure a GitHub Actions *workflow* to be triggered when an *event* occurs in your repository, such as a pull request being opened or an issue being created. Your workflow contains one or more *jobs* which can run in sequential order or in parallel. Each job will run inside its own virtual machine *runner*, or inside a container, and has one or more *steps* that either run a script that you define or run an *action*, which is a reusable extension that can simplify your workflow.



Workflows

A workflow is an automated procedure that you add to your repository. Workflows are made up of one or more jobs and can be scheduled or triggered by an event. The workflow can be used to build, test, package, release, or deploy a project on GitHub.

Event

An event is a specific activity that triggers a workflow. For example, in our lab session, the activity can originate from GitHub when you push a commit, where more functions are added to your chatbot, to a repository. You can see list of events that can be used to trigger workflows [here](#).

Jobs

A job is a set of steps that execute on the same runner. By default, a workflow with multiple jobs will run those jobs in parallel. You can also configure a workflow to run jobs sequentially.

Steps

A step is an individual task that can run commands in a job. A step can be either an *action* or a shell command. Each step in a job executes on the same runner, allowing the actions in that job to share data with each other.

Actions

Actions are standalone commands that are combined into *steps* to create a *job*. Actions are the smallest portable building block of a workflow. You can create your own actions, or use actions created by the GitHub community. To use an action in a workflow, you must include it as a step.

Runners

A runner is a server that has the GitHub Actions runner application installed. You can use a runner hosted by GitHub, or you can host your own. A runner listens for available jobs, runs one job at a time, and reports the progress, logs, and results back to GitHub.

Create an Example Workflow

GitHub Actions uses YAML syntax to define the events, jobs, and steps. These YAML files are stored in your code repository, in a directory called **.github/workflows**.

In the following, we are going to create an example workflow based on the chatbot we have completed before.

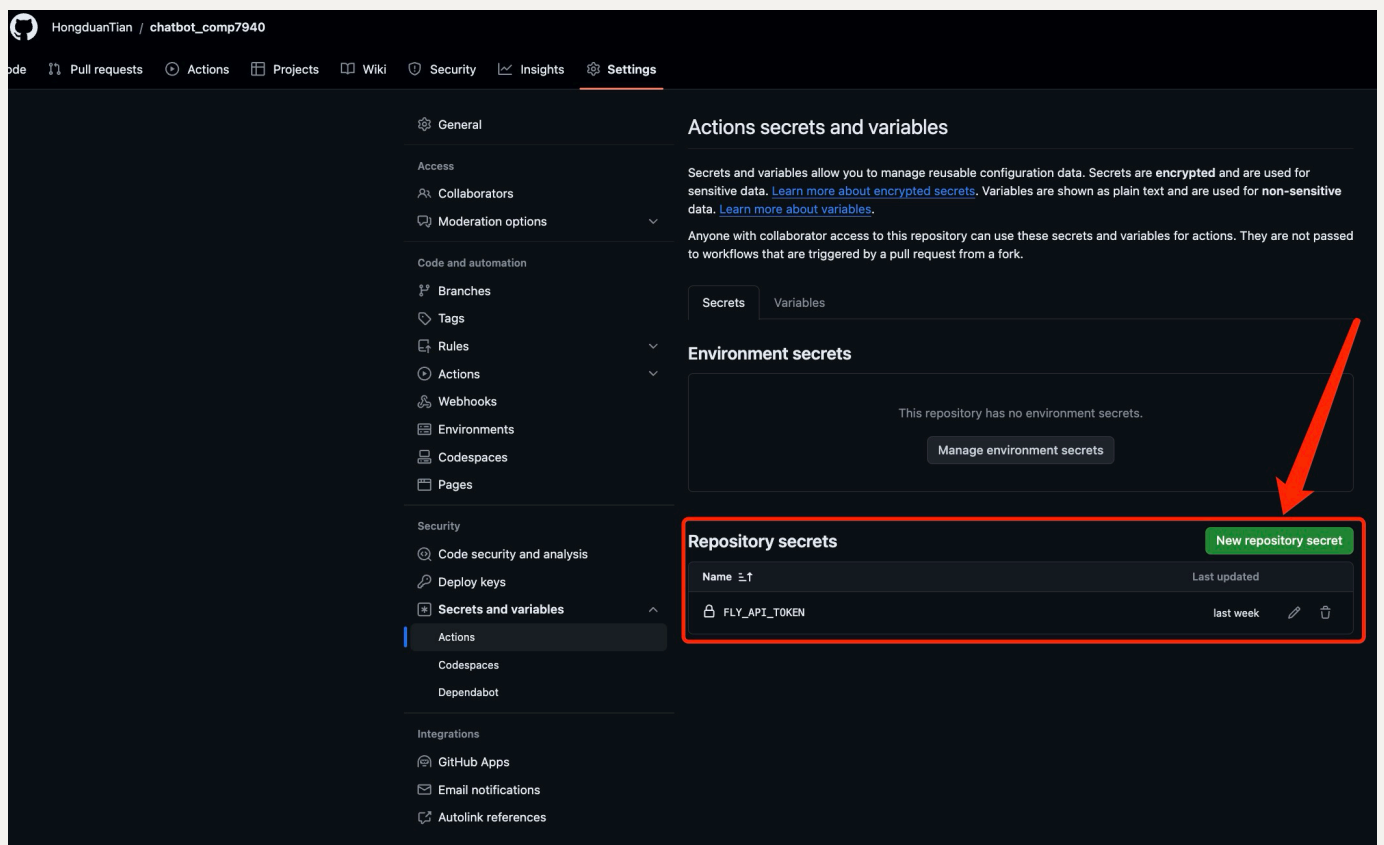
Preliminary

According to our application, we hope that the chatbot app can be deployed every time we push a new commit to the repository. Thus, the configuration mainly takes place before you push the commit.

Fly API

Before we configure YAML file, we need to first generate a Fly API in the project source directory with the command: `fly tokens create deploy -x 999999h` and copy the output to the repository you have created via `settings -> Secrets and Variables -> Actions`. There, you need to create a new repository secret call `FLY_API_TOKEN` with the tokens you generated.

```
(chatbot) tianhongduan@192 chatbot_comp7940 % fly tokens create deploy -x 999999h
FlyV1 fm2_1JPECAAAAAAAAAAiiA4RCziVAAeQT3wp1Hw1bnCr4wrVedHPwczovL2FwaS5mbHkuaW8vYWFhL3YxxDxu+44vQkUF4XE2QR+3WRzWob755...
dHRwczovL2FwaS5mbHkuaW8vYWFhL3YxxDxu+44vQkUF4XE2QR+3WRzWob755...
C0qog/lsDKZ+kR02rETjrDiync8E0ixqNNbhYl...
fsG8tIv2hfmBnpwQsnJs, vgmngZ5LAUKgc4AKz37HwWRond...
aRNTzz9poW0w=-.fm2_1JPETjrDiync8E0ixqNNbhYl...
v/rENfsG8tIv2hfmBnpwQsnJs, vgmngZ5LAUKgc4AKz37HwWRond...
wwFzAAAAE8ZqITCpHOAAeYzgZEEKqzCJamHFfFUKoigzC5R8PEIEVMgc+2koKwEUA1Ak100ajQKcB3rT8Ehq750YC80Nh
```



Configure .yaml File

Then, in your project source directory, you can create the `.github/workflows/fly.yaml` file.

1. Create the `.github/workflows` directory:

```
mkdir -p ../github/workflows
```

2. Create `fly.yml` in the workflows directory via the following contents:

```
name: Fly Deploy
on:
  push:
    branches:
      - master    # change to main if needed
jobs:
  deploy:
    name: Deploy app
    runs-on: ubuntu-latest
    concurrency: deploy-group    # optional: ensure only one action
    runs at a time
    steps:
      - uses: actions/checkout@v3
      - uses: superfly/flyctl-actions/setup-flyctl@master
      - run: flyctl deploy --remote-only
    env:
      FLY_API_TOKEN: ${ secrets.FLY_API_TOKEN }
```

NOTE: The default branch is `master`, you can check this with `git status` in the repository you have cloned. If it is `main` in your local machine, change it in the corresponding place of `fly.yml` file.

Push your repository and deploy your app

After pushing the files into the repository, the fly.io will automatically build and deploy your app. You can monitor the process details by clicking the Action menu in your repository.

HongduanTian / chatbot

<> Code

🔍 Pull requests

🔍 Actions

📁 Projects

📖 Wiki

🔒 Security

📊 Insights

⚙️ Settings

Q Type / to search

Actions

New workflow

All workflows

Workflows

Fly Deploy

Management

Caches

Runners

All workflows

Showing runs from all workflows

2 workflow runs

Event - Status - Branch - Actor -

🟡 second commit

Fly Deploy #2: Commit 940ee54 pushed by HongduanTian

master

📅 now

🔄 in progress

...

🔴 first commit

Fly Deploy #1: Commit 419e12c pushed by HongduanTian

master

📅 4 minutes ago

🕒 56s

...




Deploy app

Started 1m 3s ago

- > ☒ Set up job
- > ☒ Run actions/checkout@v3
- > ☒ Run superfly/flyctl-actions/setup-flyctl@master
- > ☒ Run flyctl deploy --remote-only
- ☐ Post Run actions/checkout@v3

Deploy app

Started 1m 34s ago

- >  Set up job
- >  Run actions/checkout@v3
- >  Run superfly/flyctl-actions/setup-flyctl@master

▼ Run flyctl deploy --remote-only

```
1  ▶ Run flyctl deploy --remote-only
6  ==> Verifying app config
7  --> Verified app config
8  Validating /home/runner/work/chatbot/chatbot/fly.toml
9  ✓ Configuration is valid
10 ==> Building image
11 Waiting for remote builder fly-builder-floral-morning-8104...
12 Remote builder fly-builder-floral-morning-8104 ready
13 Waiting for remote builder fly-builder-floral-morning-8104...
14 Remote builder fly-builder-floral-morning-8104 ready
15 ==> Building image with Buildpacks
16 --> docker host: 20.10.12 linux x86_64
17 base: Pulling from paketobuildpacks/builder
18 Digest: sha256:17ea21162ba8c7717d3ead3ee3836a368aced7f02f2e59658e52029bd6d149e7
19 Status: Image is up to date for paketobuildpacks/builder:base
20 base-cnb: Pulling from paketobuildpacks/run
21 Digest: sha256:1af9935d8987fd52b2266d288200c9482d1dd5529860bbf5bc2d248de1cb1a38
22 Status: Image is up to date for paketobuildpacks/run:base-cnb
23 ==> ANALYZING
24 Restoring data for SBOM from previous image
25 ==> DETECTING
26 6 of 9 buildpacks participating
27 paketo-buildpacks/ca-certificates 3.6.3
28 paketo-buildpacks/cpython         1.8.11
29 paketo-buildpacks/pip              0.17.4
30 paketo-buildpacks/pip-install      0.5.16
31 paketo-buildpacks/python-start     0.14.11
32 paketo-buildpacks/procfile         5.6.4
33 ==> RESTORING
34 Restoring metadata for "paketo-buildpacks/ca-certificates:helper" from app image
35 Restoring data for SBOM from cache
36 ==> BUILDING
37
38 Paketo Buildpack for CA Certificates 3.6.3
39 https://github.com/paketo-buildpacks/ca-certificates
40 Launch Helper: Reusing cached layer
41 Paketo Buildpack for CPython 1.8.11
42 Resolving CPython version
43   Candidate version sources (in priority order):
44     -> ""
45     <unknown> -> ""
46   Selected CPython version (using ): 3.10.12
47 Executing build process
48 Installing CPython 3.10.12
```

```
Deploy app
succeeded now in 2m 41s

Beta Give feedback Q Search logs

Run flyctl deploy --remote-only

248 Image: registry.fly.io/chatbot-hongdun:deployment-01HPXF0N1R2342Z1WAEQ567YXP
241 Image size: 290 MB
242 Watch your deployment at https://fly.io/apps/chatbot-hongdun/monitoring
245 Provisioning ips for chatbot-hongdun
246 Dedicated ipv6: 2a09:8280:1::2b:25e4:0
247 Shared ipv4: 66.241.124.252
248 Add a dedicated ipv4 with: fly ips allocate-v4
258 This deployment will:
251 * create 2 "app" machines
252 > Launching new machine
253 No machines in group app, launching a new machine
255 > Machine 3287339ce95185 [app] was created
256 > Checking that 3287339ce95185 [app] is up and running
257 > Waiting for 3287339ce95185 [app] to have state: started
258 > Machine 3287339ce95185 [app] has state: started
259 ✓ Machine 3287339ce95185 [app] update finished: success
261 WARNING The app is not listening on the expected address and will not be reachable by fly-proxy.
262 You can fix this by configuring your app to listen on the following addresses:
263 - 0.0.0.0:8080
264 Found these processes inside the machine with open listening sockets:
265 PROCESS | ADDRESSES
266 -----
267 /.fly/hallpass | [fdaa:6:a21f:a7b:b4f1:bcc2:66a6:2]:22
268 Creating a second machine to increase service availability
278 ✓ Machine 148ed667c51e38 [app] was created
271 ✓ Checking that 148ed667c51e38 [app] is up and running
272 ✓ Waiting for 148ed667c51e38 [app] to have state: started
273 ✓ Machine 148ed667c51e38 [app] has state: started
274 ✓ Machine 148ed667c51e38 [app] update finished: success
275 Finished launching new machines
276 NOTE: The machines for [app] have services with 'auto_stop_machines = true' that will be stopped when idling
277 Checking DNS configuration for chatbot-hongdun.fly.dev
278 Visit your newly deployed app at https://chatbot-hongdun.fly.dev/

Post Run actions/checkout@v3

1 Post job cleanup.
2 /usr/bin/git version
3 git version 2.43.0
4 Temporarily overriding HOME="/home/runner/work/_temp/fc1afd31-6c11-4825-b449-750140876aa5" before making global git config changes
5 Adding repository directory to the temporary git global config as a safe directory
6 /usr/bin/git config --global --add safe.directory /home/runner/work/chatbot/chatbot
7 /usr/bin/git config --local --name-only --get-regexp core.sshCommand
8 /usr/bin/git submodule foreach --recursive sh -c "git config --local --name-only --get-regexp 'core.sshCommand' && git config --local --unset-all 'core.sshCommand' || :"
9 /usr/bin/git config --local --name-only --get-regexp http.https://github.com/\.\.extraheader
10 http.https://github.com/.extraheader
11 /usr/bin/git config --local --unset-all http.https://github.com/.extraheader
12 /usr/bin/git submodule foreach --recursive sh -c "git config --local --name-only --get-regexp 'http.https://github.com/\.\.extraheader' && git config --local --unset-all 'http.https://github.com/.extraheader' || :"
```

Write up:

Answer the following questions and submit them to moodle:

1. Identify which ONE command does the job to upload/deploy your chatbot to fly.io.
2. How does your GitHub repo related/connected to your fly.io app and your local repo? Please draw them into a single diagram and explain it briefly.
3. Explain why it is a bad idea to use config.ini to store our password and how the current setting different than before?
4. Prove that you have finished lab 6. Make some screen captures of your Git Action and explain them.