SINE NOMINE
ASSOCIATES

Github Action Runners

Neale Ferguson

2024-06-20

Agenda

- What are Github Actions
- What are Action Runners
- Why would you care
- Types:
  - Self-hosted
  - Marketplace
- Using Action Runners
- References
- Building and using your own Runner

Github Actions

- A built-in CI/CD platform offered by GitHub
- Allows you to automate various tasks within your software development workflow directly from your GitHub repositories

Github Actions

- Workflow Automation:
  - Define automated workflows using YAML files within your repository.
  - These workflows can be triggered by various events, such as pushes to specific branches, pull requests, or scheduled intervals.

Github Actions

- Jobs and Steps:
  - Break down your workflows into smaller, manageable units called jobs.
  - Each job can consist of multiple steps, which define specific actions to be executed.

Github Actions

- Actions:
  - The building blocks of your workflows.
  - They represent specific tasks you want to perform, for example:
    - Building your code
    - Running tests
    - Deploying your application
    - Sending notifications.

Github Actions

- Marketplace
  - A rich marketplace offers a vast collection of pre-built actions for common tasks.
  - You can also create your own custom actions or reuse actions from other developers.

Github Actions - Benefits

- Streamlined Development
  - Automates repetitive tasks, freeing up developer time for more creative work.
- Improved Quality
  - Integrates testing and linting into your workflow, ensuring code quality.

Github Actions - Benefits

- Faster Releases
  - Automates deployments, enabling faster and more frequent releases.
- Collaboration
  - Integrates with other tools and services used in your development process.

Example Workflow

- This workflow gets triggered when there's a push to the main branch and performs the following actions:
  - Checks out the code from the repository.
  - Uses a pre-built action to build the code.
  - Runs another action to execute unit tests.
  - If all tests pass, deploys the application to a staging server.

```
name: CI
on:
  push:
    branches: [ main ]
jobs:
  build-and-test:
    runs-on: ubuntu-latest  # Workflow will run on Ubuntu Linux
    steps:
      - uses: actions/checkout@v3  # Checks out the code from the repo
      - name: Build the code
        run: ./build.sh  # Replace this with your specific build command
      - name: Run unit tests
        run: ./test.sh  # Replace this with your unit test command
      - name: Deploy to staging (if tests pass)
        if: success()  # Runthis step if all previous steps succeed
        run: ./deploy-to-staging.sh  # Replace with your deployment cmd
```

Github Action Example Explained

- name: Name of the workflow, which is displayed in the GitHub Actions UI for better organization
- on: Defines the event that triggers the workflow. In this case, the workflow runs when there's a push to the main branch

Github Action Example Explained

- jobs: Defines the jobs within the workflow. Here, we have one job named build-and-test
- runs-on: Specifies the runner operating system where the job will execute. Here, we're using an Ubuntu runner.

Github Action Example Explained

- steps: This defines the individual steps within the job. Each step executes a specific command:
  - uses the actions/checkout@v3 action to check out the code from the repository
  - builds the code using a custom command (./build.sh)
  - runs unit tests using a custom command (./test.sh)

Github Action Example Explained

- steps: This defines the individual steps within the job
  - – conditionally deploys the application to a staging server using a custom command (./deploy-to-staging.sh)
    - The if: success() condition ensures this step only runs if all previous steps succeed.

Action Runners

- GitHub Actions runners are the machines that execute the jobs defined in your GitHub Actions workflows:
  - GitHub-hosted runners
    - VMs provided by GitHub itself. They come with pre-installed tools and environments commonly used in development workflows.
    - VMs provided by marketplace: coming soon to IBM Cloud for z and Power

Action Runners

- Self-hosted runners:
  - Machines you set up and manage yourself. This gives you more control over the hardware, software, and security of your workflow execution.

Github-Hosted Action Runners

SINE NOMINE
ASSOCIATES

- Advantages:
  - Easy to use: No setup or maintenance required: you just define your workflow.
  - Scalability: GitHub automatically scales the runners based on demand.
  - Free for basic use: Limited free minutes are included in your GitHub account.

Github-Hosted Action Runners

- Considerations:
  - Limited control: You don't have control over the hardware or software configuration of the runners.
  - Cost: Free minutes have limitations, exceeding them incurs charges.
  - Security: Public runners might not be suitable for workflows handling sensitive data.

Self-Hosted Motivation

- Flexibility & Control
  - Hardware – you control the environment to match specific requirements
  - Software – you install the packages required by the workflow
  - Security: behind your own firewalls and access controls

Self-Hosted Motivation

- Flexibility & Control
  - Customization
    - Specialized runners to fit specific needs
  - Cost
    - Heavy workloads that aren’t limited by hosted usage settings

Self-Hosted Motivation

- Considerations:
  - Maintenance: Requires setting up, maintaining, and securing the runner machines.
  - Scalability: Scaling up or down runners requires manual intervention.

Choosing Runner Type

- Dependent on Requirements:
  - GitHub-hosted runners are a great option for getting started or for workflows that don't require specific configurations.
  - Self-hosted runners are ideal for scenarios where you need more control, have specialized software requirements, or handle sensitive data.

Github Action Runners Come to z

- Why now?
  - The Github action runners are written for .NET
  - z and Power now have .NET
- Rest of Presentation addresses self-hosted Action Runners

Self-Hosted Runner Types

- One-to-One
  - An instance per github repo

```
CONTAINER ID   IMAGE       COMMAND                  STATUS          NAMES
c6219e38da6c   runner:test /bin/sh -c /opt/r... Up 17 minutes   gallant_boyd


√ Connected to GitHub

Current runner version: '2.317.0'
2024-06-18 18:20:18Z: Listening for Jobs
```

Self-Hosted Runner Types

- One-to-Many
  - An instance triggers a runner for any repo
  - Docker + LXD Containers

```
CONTAINER ID  IMAGE           COMMAND                    NAMES
8a16f2b7adb3  rabbitmq:3.12   rabbitmq-server (healthy)  actions-runner_rabbitmq_1
8f7fdf67f5e2  couchdb:3.3.2   /opt/couchdb/bin/...       actions-runner_couchdb_1
5040e7390bcc  gh-app:latest   /gh-app                    actions-runner_gh-app_1
6bcc0de16409  listener:latest /listener                  actions-runner_listener_1
47b02d7ba170  lxd:latest      /lxd                       actions-runner_lxd_1

+-------------------+-----------------------------------------+-----------+
|       ALIAS       | DESCRIPTION                             | SIZE      |
+-------------------+-----------------------------------------+-----------+
| ubuntu-22.04-s390x | GitHub Actions ubuntu 22.04 Runner for s390x | 2494.32MiB |
+-------------------+-----------------------------------------+-----------+
```

Demo Time

- Live chicken has been obtained
- Incense has been burned
- Let’s go…

Self-Hosted Runner Configuration

- If we have time…

https://youtu.be/SASoUr9X0QA

References

- GitHub Actions - Self-hosted runners - Installation & Calling - https://youtu.be/SASoUr9X0QA
- GitHub Actions: Write your first workflow with GitHub APIs - https://youtu.be/-hVG9z0fCac

Building a Self-Hosted Runner

- Download patch

https://download.sinenomine.net/vmworkshop/runner-sdk-8.patch

- Add following slides into Dockerfile
- Change the RUNNERPATCH argument or use the `--build-arg` option to specify the location of where you placed patch
- Build the container image

```
docker build --tag runner:8 .
```

Building a Self-Hosted Runner

```
FROM    almalinux:9

ARG     RUNNERREPO="https://github.com/actions/runner" \
        RUNNERPATCH=runner-sdk-8.patch \
        SDK=8 ARCH=s390x

RUN     dnf update -y -q && \
        dnf install -y -q wget git which langpacks-en glibc-all-langpacks sudo

RUN     dnf install -y -q dotnet-sdk-${SDK}.0 && \
        echo "Using SDK - `dotnet --version`"

COPY    ${RUNNERPATCH} /tmp/runner.patch

RUN     cd /tmp && \
        git clone -q ${RUNNERREPO} && \
        cd runner && \
        git checkout $(git describe --tags $(git rev-list --tags --max-count=1)) -b build && \
        git apply /tmp/runner.patch

RUN     cd /tmp/runner/src && \
        ./dev.sh layout && \
        ./dev.sh package && \
        ./dev.sh test && \
        rm -rf /root/.dotnet /root/.nuget
```

Building a Self-Hosted Runner

```
RUN     useradd -c "Action Runner" -m almalinux && \
        usermod -L almalinux && \
        echo "almalinux  ALL=(ALL)       NOPASSWD: ALL" >/etc/sudoers.d/almalinux

RUN     mkdir -p /opt/runner && \
        tar -xf /tmp/runner/_package/*.tar.gz -C /opt/runner && \
        chown -R almalinux:almalinux /opt/runner && \
        su -c "/opt/runner/config.sh --version" almalinux

RUN     dnf install -y -q cmake make automake autoconf m4 gcc gcc-c++ libtool epel-release

RUN     rm -rf /tmp/runner /var/cache/dnf/* /tmp/runner.patch && \
        dnf clean all

USER    almalinux

EXPOSE  443

CMD     /bin/bash
```

Running a Self-Hosted Runner

- Create an action runner instance from this Dockerfile

```
FROM    localhost/runner:8
ARG     REPO TOKEN
RUN     /opt/runner/config.sh --url ${REPO} --token ${TOKEN}
CMD     /opt/runner/run.sh
```

- Use the TOKEN obtained from configuring the target REPO on github

```
docker build --tag <instance-name>:runner --squash \
           --build-arg REPO=<repo> TOKEN=<token> .
```

- Run the container

```
docker run <instance-name>:runner
```

Running a Self-Hosted Runner

- Run the container and check its logs

```
> podman run –d <instancename>:runner
> podman logs <name of container>

√ Connected to GitHub

Current runner version: '2.317.0'
2024-06-20 16:54:19Z: Listening for Jobs
```