

Github actions runner

This page is documentation of the github actions runner we plan to use

- [Possible setup #1 - GitHub enterprise](#)
- [Possible setup #2 - Self hosted runner](#)
- [Possible setup #2.1 - Self hosted runner in a container](#)
- [Possible setup #3 - Self hosted runner with jobs in containers](#)
- [Possible setup #4 - Self hosted runner autoscaling \(kubernetes\)](#)

Setting up organization wide runner

- [Workflow Access](#)

Possible setup #1 - GitHub enterprise

Possible setup #2 - Self hosted runner

I made a self-hosted runner that runs on s3df on a repository named BuildSystem [slaclab/BuildSystem: BuildSystem for EED software development and configuration management \(github.com\)](#). The actions themselves doesn't do much, since this is a proof of concept.

But also communicates with github, and posts build results there under the 'Action' tab.

1. Here is setup

1. And then run, and manually trigger the workflow using Github CLI (I installed github CLI on my goenv)

Result on github repo 'Actions'

To think about: Should we use github CLI to be the base tool for our CLI? It seems to have features we need like triggering a workflow manually. But it also allows you to create your own commands. Maybe we can make a simplified wrapper around the CLI - And make only certain options visible to the end user?

Resources:

[Adding self-hosted runners - GitHub Docs](#)

If go this route:

1. Add the self-hosted runner at the organization level (So every repo just needs to enable their self-hosted runner to the 'buildsystem' so they're able to communicate to the buildsystem).
2. (optional) configure the self-hosted runner as a service

Possible setup #2.1 - Self hosted runner in a container

1. This is the same as #2 except it runs in an apptainer container

```
(goenv) [pnispero@sdfiana008 BuildSystem]$ ls
actions-runner      build_system_runner.sif      gh-test-ext
build_system_runner.def  build_system_runner_test.def  README.md
(goenv) [pnispero@sdfiana008 BuildSystem]$ apptainer instance start build_system
runner.sif build1
INFO: Instance stats will not be available - requires cgroups v2 with systemd
as manager.
INFO: instance started successfully
(goenv) [pnispero@sdfiana008 BuildSystem]$ apptainer instance list
INSTANCE NAME    PID      IP      IMAGE
build1           1800387  /sdf/home/p/pnispero/BuildSystem/build_system_
runner.sif
(goenv) [pnispero@sdfiana008 BuildSystem]$ apptainer run instance://build1
```

```
ion.GetConnectionData
[RUNNER 2024-04-29 23:44:54Z INFO GitHubActionsService] Finished operation Locat
ion.GetConnectionData
[RUNNER 2024-04-29 23:44:54Z INFO GitHubActionsService] Finished operation Locat
ion.GetConnectionData
[RUNNER 2024-04-29 23:44:54Z INFO MessageListener] VssConnection created
[RUNNER 2024-04-29 23:44:54Z INFO Terminal] WRITE LINE:

✓ Connected to GitHub
[RUNNER 2024-04-29 23:44:54Z INFO Terminal] WRITE LINE:

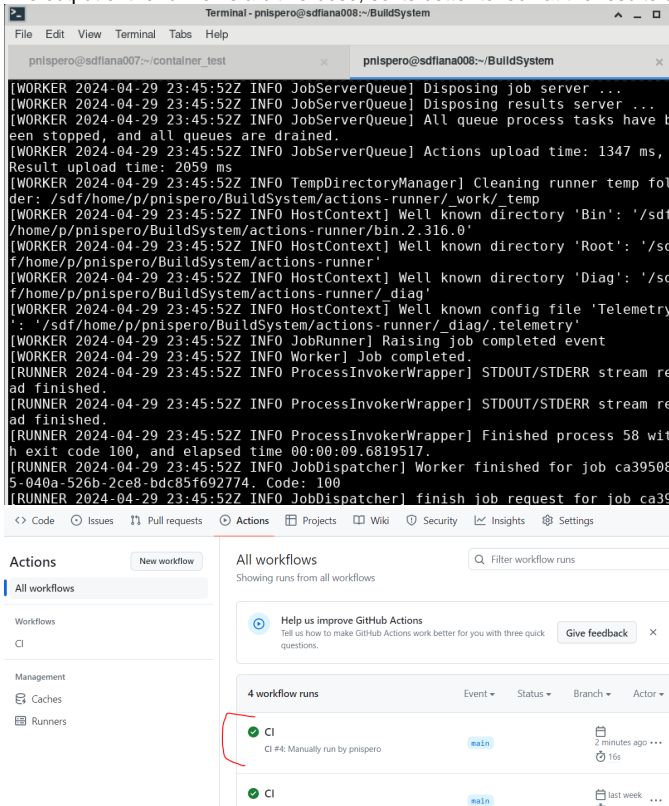
[RUNNER 2024-04-29 23:44:54Z INFO RSAFileKeyManager] Loading RSA key parameters
from file /sdf/home/p/pnispero/BuildSystem/actions-runner/.credentials_rsaparams
[RUNNER 2024-04-29 23:44:54Z INFO RSAFileKeyManager] Loading RSA key parameters
from file /sdf/home/p/pnispero/BuildSystem/actions-runner/.credentials_rsaparams
[RUNNER 2024-04-29 23:44:54Z INFO GitHubActionsService] AAD Correlation ID for t
his token request: Unknown
[RUNNER 2024-04-29 23:44:55Z INFO MessageListener] Session created.
[RUNNER 2024-04-29 23:44:55Z INFO Terminal] WRITE LINE: Current runner version:
'2.316.0'
Current runner version: '2.316.0'
[RUNNER 2024-04-29 23:44:55Z INFO Terminal] WRITE LINE: 2024-04-29 23:44:55Z: Li
stening for Jobs
2024-04-29 23:44:55Z: Listening for Jobs
[RUNNER 2024-04-29 23:44:55Z INFO JobDispatcher] Set runner/worker IPC timeout t
o 30 seconds.
```

2. Run the workflow. (The first x means not completed, then it completes)

```
(goenv) [pnispero@sdfiana008 BuildSystem]$ gh workflow run
? Select a workflow CI (basic.yml)
✓ Created workflow_dispatch event for basic.yml at main

To see runs for this workflow, try: gh run list --workflow=basic.yml
(goenv) [pnispero@sdfiana008 BuildSystem]$ gh run list --workflow=basic.yml
STATUS  TITLE      WORKFLOW  BRANCH  EVENT      ID          ELAPSED  AGE
* [x] CI      CI         main    workfl...  888693...  2s       less t...
✓ CI      CI         main    workfl...  879177...  17s      about ...
✓ CI      CI         main    workfl...  879069...  16s      about ...
✓ Create ba... CI         main    push     878936...  4m4s     about ...
(goenv) [pnispero@sdfiana008 BuildSystem]$ gh run list --workflow=basic.yml
STATUS  TITLE      WORKFLOW  BRANCH  EVENT      ID          ELAPSED  AGE
✓ [x] CI      CI         main    workfl...  888693...  16s      less t...
✓ CI      CI         main    workfl...  879177...  17s      about ...
✓ CI      CI         main    workfl...  879069...  16s      about ...
✓ Create ba... CI         main    push     878936...  4m4s     about ...
(goenv) [pnispero@sdfiana008 BuildSystem]$
```

3. The output of the runner is a bit verbose, so its better to look at the results on github actions



This works too but i am not sure if is scalable, because in the definition file (build_system_runner.def), i cd into the actions-runner/ directory which is already preconfigured for 1 runner. If multiple runners are made, then may have to configure all of them.

Resources:

[Provide runner as a Docker Image · Issue #367 · actions/runner \(github.com\)](#)

[Package actions-runner \(github.com\)](#)

Possible setup #3 - Self hosted runner with jobs in containers

May be the optimal route if we host the build system on our own hardware (or virtual).

This solves 2 problems:

1. No bottleneck on the self hosted runner since the runner will just spin up containers to do the jobs, not the runner itself.
2. Not having the runner rely on github resources that can potentially cost money if overused. Also not having a self-hosted runner for each project, we can host the runner organization wide (like propose #2) and have every project enable their actions runner to 'build-system'.

[Running jobs in a container - GitHub Docs](#)

[Creating a Docker container action - GitHub Docs](#)

Possible setup #4 - Self hosted runner autoscaling (kubernetes)

May also be optimal route, but need to look into it. Surface level: Runners are created/deleted based off usage. This route involves kubernetes infrastructure.

[Autoscaling with self-hosted runners - GitHub Docs](#)

[About Actions Runner Controller - GitHub Docs](#)

Setting up organization wide runner

1. Made personal organization for sandbox 'ad-build-test'
2. Added a runner called runner1 in runner group ad-build-test-runner

- then i forked the build system repo to ad-build-test
- and the actions runner was automatically set to the organization runner
- So i clones the repo on s3df, did a gh repo set-default to set the repo to ad-build-test/BuildSystem
- then did a gh workflow run to see if it worked and it did.
- So i then created another runner called runner2, but ran the same commands, the only difference was i changed the actions-runner folder to actions-runner2

```

Self-hosted runner registration

# Authentication
✓ Connected to GitHub

# Runner Registration
Enter the name of the runner group to add this runner to: [press Enter for ad-build-test-runner]
Enter the name of runner: [press Enter for sdflana007] runner2
This runner will have the following labels: 'self-hosted', 'Linux', 'X64'
Enter any additional labels (ex. label-1,label-2): [press Enter to skip]
✓ Runner successfully added
✓ Runner connection is good

# Runner settings
Enter name of work folder: [press Enter for _work]
✓ Settings Saved.

[gnispero@sdflana007 actions-runner2]$

gnispero@sdflana007:~$ container test
# Authentication
✓ Connected to GitHub

# Runner Registration
Enter the name of the runner group to add this runner to: [press Enter for ad-build-test-runner]
Enter the name of runner: [press Enter for sdfllogin003] runner1
This runner will have the following labels: 'self-hosted', 'Linux', 'X64'
Enter any additional labels (ex. label-1,label-2): [press Enter to skip]
✓ Runner successfully added
✓ Runner connection is good

# Runner settings
Enter name of work folder: [press Enter for _work]
✓ Settings Saved.

[gnispero@sdfllogin003 actions-runner]$ ls
actions-runner-linux-x64-2.316.0.tar.gz  env.sh  run.sh

```

```

[gnispero@sdflana007 BuildSystem]$ gh workflow run
? Select a workflow CI-2 (basic.yml)
✓ Created workflow_dispatch event for basic2.yml at main

To see runs for this workflow, try: gh run list --workflow=basic2.yml
[gnispero@sdflana007 BuildSystem]$ gh run list --workflow=basic2.yml
STATUS  TITLE      WORKFLOW  BRANCH  EVENT  ID      ELAPSED  AGE
-----
CI-2    CI-2       main      workfl...  push   891335...  19s     less t...
Create ba... CI-2       main      push     891306...  1m11s    about ...
[gnispero@sdflana007 BuildSystem]$

[gnispero@sdflana008 BuildSystem]$ gh run list --workflow=basic.yml
STATUS  TITLE      WORKFLOW  BRANCH  EVENT  ID      ELAPSED  AGE
-----
CI       CI         main      workfl...  push   891325...  16s     less t...
Create ba... CI         main      push     891306...  1m7s     about ...
[gnispero@sdflana008 BuildSystem]$ gh workflow run
? Select a workflow CI (basic.yml)
✓ Created workflow_dispatch event for basic.yml at main

To see runs for this workflow, try: gh run list --workflow=basic.yml
[gnispero@sdflana008 BuildSystem]$

Terminal - gnispero@sdflana007:~$ ad-build-test_runneractions-runner2
✓ Connected to GitHub

# Runner Registration
Enter the name of the runner group to add this runner to: [press Enter for ad-build-test-runner]
Enter the name of runner: [press Enter for sdflana007] runner2
This runner will have the following labels: 'self-hosted', 'Linux', 'X64'
Enter any additional labels (ex. label-1,label-2): [press Enter to skip]
✓ Runner successfully added
✓ Runner connection is good

# Runner settings
Enter name of work folder: [press Enter for _work]
✓ Settings Saved.

[gnispero@sdflana007 actions-runner2]$ ./run.sh
✓ Connected to GitHub
Current runner version: '2.316.0'
2024-05-01 18:27:48Z: Listening for Jobs
2024-05-01 18:30:51Z: Running job: build
2024-05-01 18:31:06Z: Job build completed with result: Succeeded

Terminal - gnispero@sdfllogin003:~$ ad-build-test_runneractions-runner
Enter any additional labels (ex. label-1,label-2): [press Enter to skip]
✓ Runner successfully added
✓ Runner connection is good

# Runner settings
Enter name of work folder: [press Enter for _work]
✓ Settings Saved.

[gnispero@sdfllogin003 actions-runner]$ ls
actions-runner-linux-x64-2.316.0.tar.gz  env.sh  run.sh
bin                                     externals                                safe_sleep.sh
config.sh                               run-helper.cmd.template                 svc.sh
diag                                   run-helper.sh.template
[gnispero@sdfllogin003 actions-runner]$ ./run.sh
✓ Connected to GitHub
Current runner version: '2.316.0'
2024-05-01 18:16:40Z: Listening for Jobs
2024-05-01 18:21:52Z: Running job: build
2024-05-01 18:22:02Z: Job build completed with result: Succeeded
2024-05-01 18:30:51Z: Running job: build
2024-05-01 18:31:01Z: Job build completed with result: Succeeded

```

- So I ran 2 different workflows (content doesn't matter) And since workflow 'CI' occupied runner1, running workflow 'CI-2' didn't queue on runner1, the job moved to next available runner, since runner2.

This proves that we can have multiple runners to satisfy users requests

Drawback:

- Since repos on slaclab are public, we would have to enable the 'Allow public repositories' on the runner groups. (Which may have a security risk if non-slaclab users fork the repo and trigger the runners)
 - (Update, we can allow public repos, just as long as the runners are only allowed to run within the organization, which is allowed for enterprise - [Managing access to self-hosted runners using groups - GitHub Enterprise Cloud Docs](#)) To solve this, in the organization's 'General actions permissions', set the policy to 'Allow <organization_name> actions', and select non-ad-build-test, actions and reusable workflows'. So only people within the organization can access the runners. And can include other actions like default github ones

Workflow Access

- In the github organization settings, we can specify 'runner groups'. Where each group can take jobs from any workflow, or certain workflows chosen by administrator.
 - This could be useful if we want 'nightly builds' to be their own group, so there will always be runners available in other runner groups.
 - You could also add labels to the runners, and then on the workflow job yaml you can specify the label to determine which runner to run on.
 - [Choosing the runner for a job - GitHub Docs](#)