

Autoscaling for HPC Runners

Tri Nguyen

Indiana University Bloomington

Mentors: Haiying Xu, Brian Vanderwende

July 30th, 2024



Objectives

- 1. Motivations**
- 2. CI/CD concepts**
- 3. Methods for Scaling HPC Runners**
 - a. Container in Container**
 - b. Autoscaling Runner with Webhooks**
- 4. Results**
- 5. Challenges and Future Work**



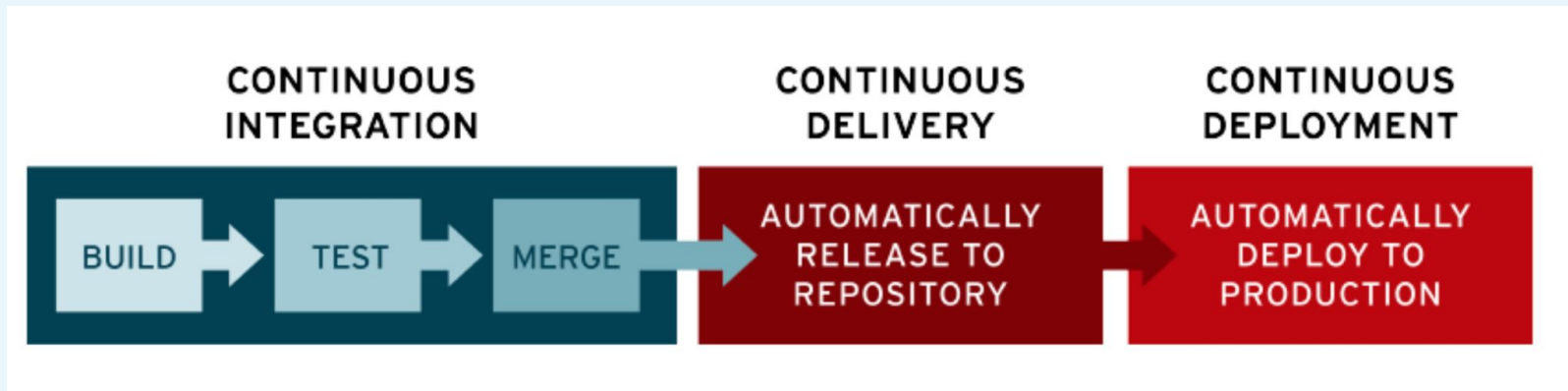
Motivations

- Repository-specific self-hosted runners were developed in previous CI/CD projects
- Inconvenient to set up self-hosted runners for every project.
- Developing centralized CI/CD server for scalability of organization-level runners



What is CI/CD?

- Automated workflow integrating code changes into source code
- Facilitates rapid and reliable delivery of software by automating the build, test, and deployment phases



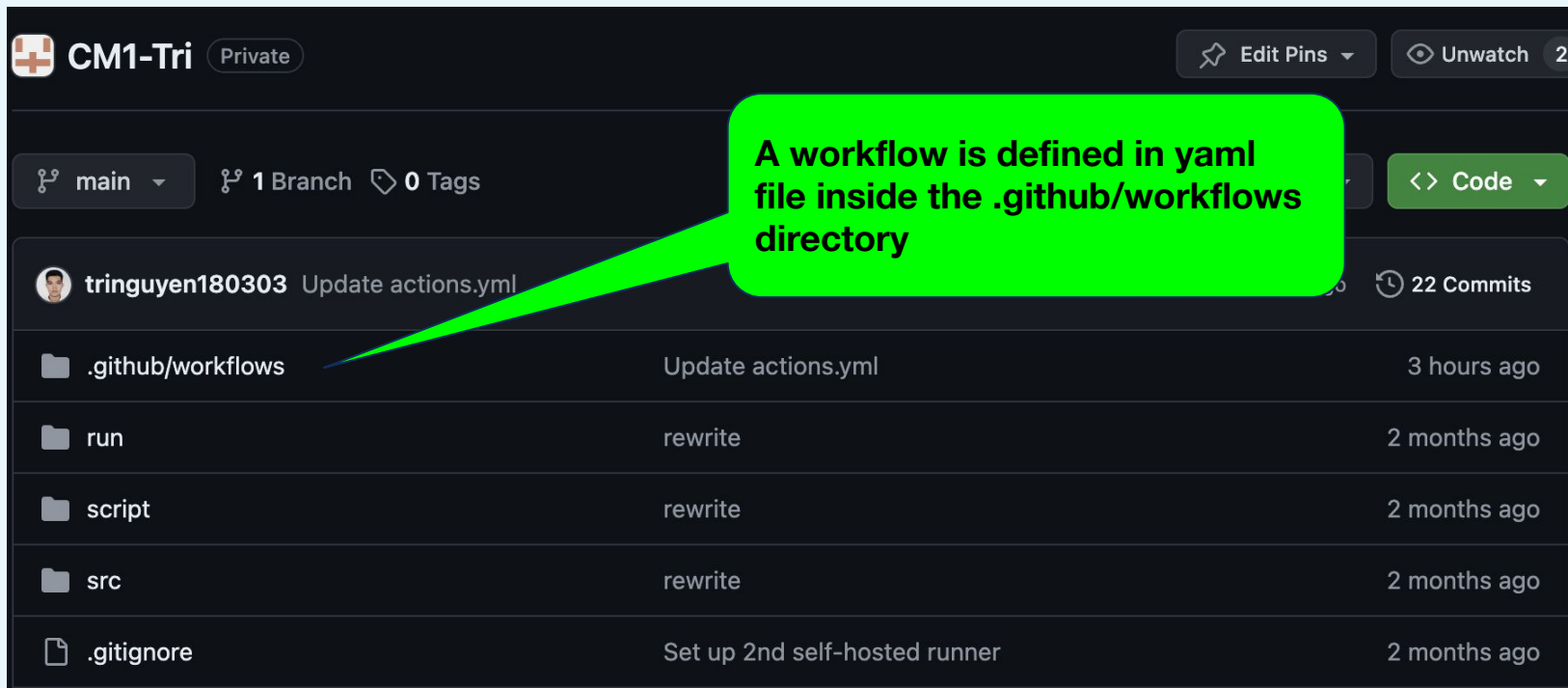
<https://www.redhat.com/en/topics/devops/what-is-ci-cd>

Benefits of CI Server

- Automated testing to ensure the stability in the codebase and cut down time of manual testing and debugging
- Enabling secure and scalable organization-level runners on the HPC system supporting Github Actions
- Avoiding costs of high-job counts, external servers (Circle CI)



Github Actions



CM1-Tri Private

main 1 Branch 0 Tags

Update actions.yml 22 Commits

.github/workflows	Update actions.yml	3 hours ago
run	rewrite	2 months ago
script	rewrite	2 months ago
src	rewrite	2 months ago
.gitignore	Set up 2nd self-hosted runner	2 months ago

A workflow is defined in yaml file inside the .github/workflows directory

Github Actions

Events

Workflow
jobs

Job name

Runner
Label

Each job could have a
set of steps to run a
command or a script

```
1  name: Submit PBS Jobs
2  run-name: ${github.actor} is building project on ${github.server_url}
3  on:
4    push:
5      branches:
6        - main
7
8  jobs:
9    build-and-submit1:
10     runs-on: self-hosted
11
12     steps:
13       - name: Checkout repository
14         uses: actions/checkout@v4
15       - name: Submit job1
16         run: |
17           job1_id=$(qsub first_job.sh)
18           echo "Job ID: $job1_id"
19           echo "job1_id=$job1_id" >> $GITHUB_ENV
20       - name: Monitor job1
21         run: |
22           echo "Job ID in ENV ${env.job1_id}"
23           ./monitor_job.sh ${env.job1_id} create 8 jobs like this
24
```

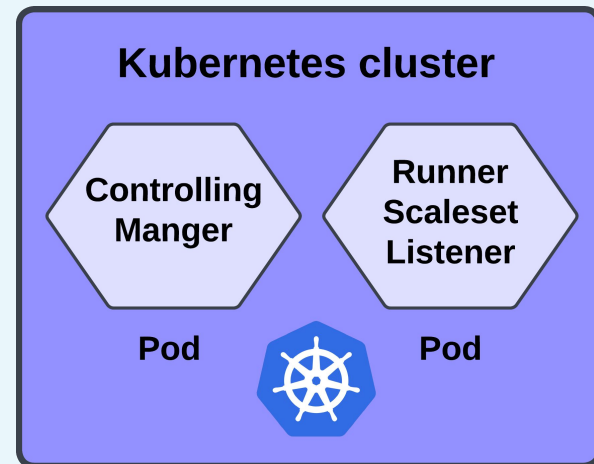
What is a runner?

Runner is program on computer that execute jobs from the workflow



Autoscaling the runners from Virtual Machine

- **Kubernetes:** container orchestration platform used for scale, manage and deploy containerized applications
- **Helm:** package manager for Kubernetes providing the way to define, install and upgrade the applications inside the Kubernetes cluster.
- **Actions Runner Controller:** Kubernetes operator that orchestrate and scale the runners for Github Actions



How do we autoscale runners in HPC?

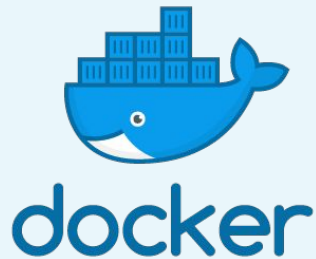


1. Container in Container



What is Container?

- Virtualized operating system
- Lightweight, standalone and executable package of software
- Include code, runtime, system tools , system libraries and settings



What is Container in Container?

- Running container technologies inside a container

Why?

- Kubernetes cluster require driver Docker or Podman
- Isolating environment for building and testing without affecting the host system
- Portable server deployment

Container

Installation:

- Kubernetes
- Helm
- Minikube/Kind/K3s
- Docker/Podman



Scaling Runners inside the Container

Organization name

Github Secret

Minimum Runners

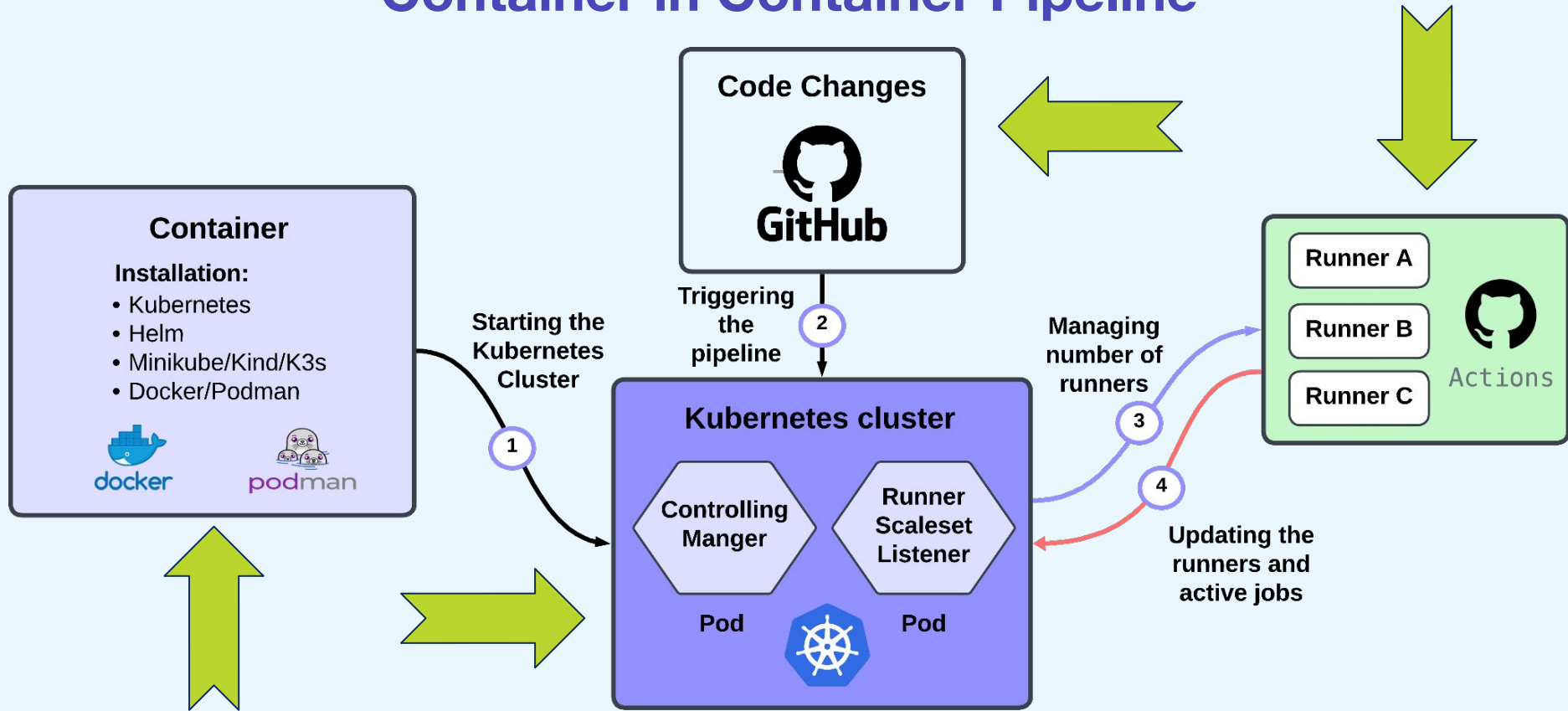
Maximum Runners

Runner Name

Starting the runner
for incoming jobs

```
1 # values.yaml
2 githubConfigUrl: "https://github.com/SIPARCS-CICD"
3 githubConfigSecret:
4   github_token: "$GITHUB_TOKEN"
5
6 minRunners: 10
7 maxRunners: 15
8
9 runnerGroup: "Default"
10 runnerScaleSetName: "my-awesome-scale-set-vm"
11 template:
12   spec:
13     containers:
14     - name: runner
15       image: ghcr.io/actions/actions-runner:latest
16       command: ["/home/runner/run.sh"]
```

Container in Container Pipeline



Roadblock

- Kubernetes cluster need Control group v2 for rootless environment
- Control group allows system to allocate resources such as CPU, memory, disk I/O and network bandwidth
- Derecho and most HPC system currently run cgroup v1 as default!



2. Autoscaling Runners with Webhooks

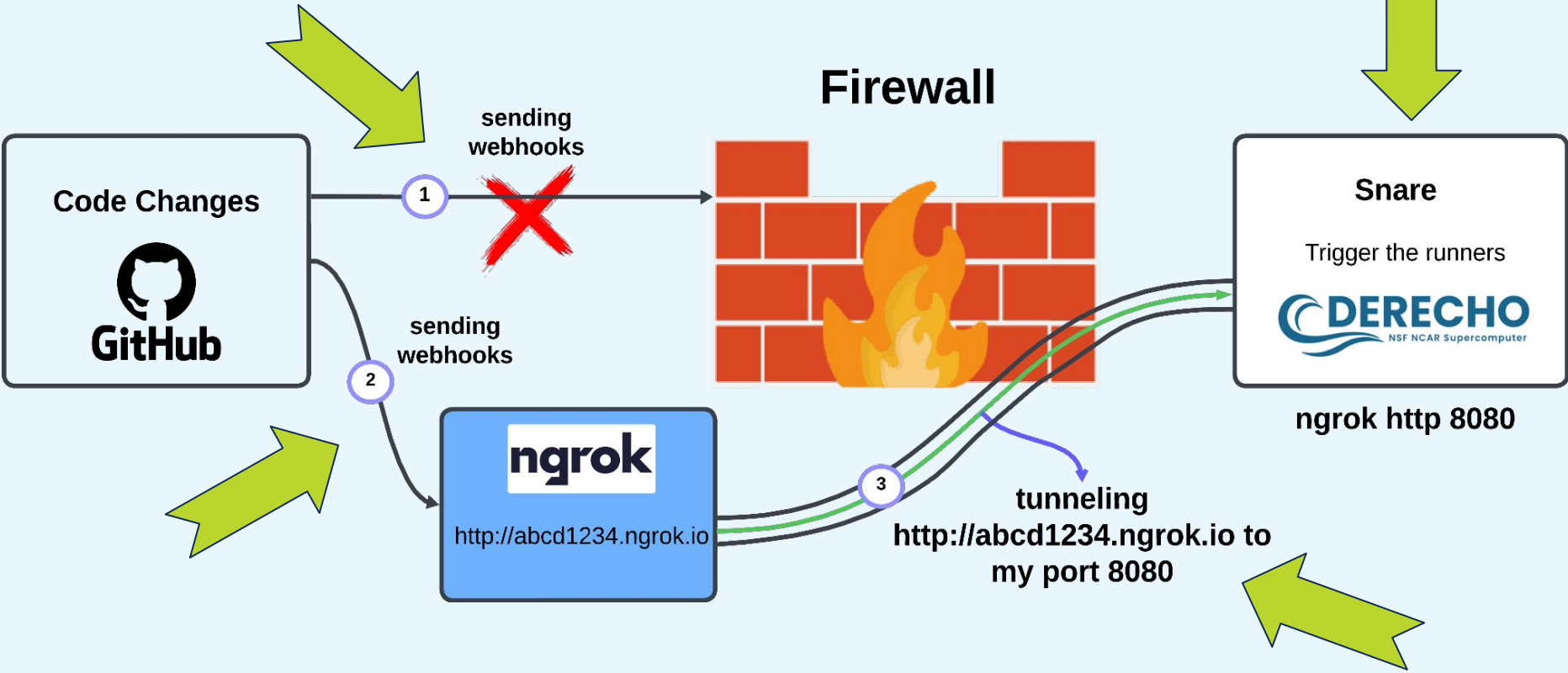


Webhooks

Whenever something happens to Github repository (Pull Request), Github will send an HTTP request to a specified URL



Autoscaling Runners with Webhooks



Snare

A simple program that listens to Github webhook events and runs the Unix command



Snare

Listening to
localhost port
8080

Running Unix
command

Sharing the
secret with
webhooks

```
listen = "127.0.0.1:8080";  
  
github {  
  match "*" {  
    cmd = "cd /glade/derecho/scratch/tringuyen/Parallel-Computing/snare-dev; ./autoscale-runners-parallel.sh"  
    secret = "hello-ncar";  
  }  
}
```

Sending Webhooks with Github

Webhooks / Manage webhook

We'll send a POST request to the URL below with details of any subscribed events. You can specify the content type you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in the GitHub documentation.

Provided ngrok tunneling

Payload URL *
http://abcd1234.ngrok.io

Content type *
application/x-www-form-urlencoded

Secret
hello-ncar

Shared secret with snare

Cancel

SSL verification
By default, we verify SSL certificates when delivering payloads.

Enable SSL verification Disable (not recommended)

Which events would you like to trigger this webhook?

Just the push event.

Results

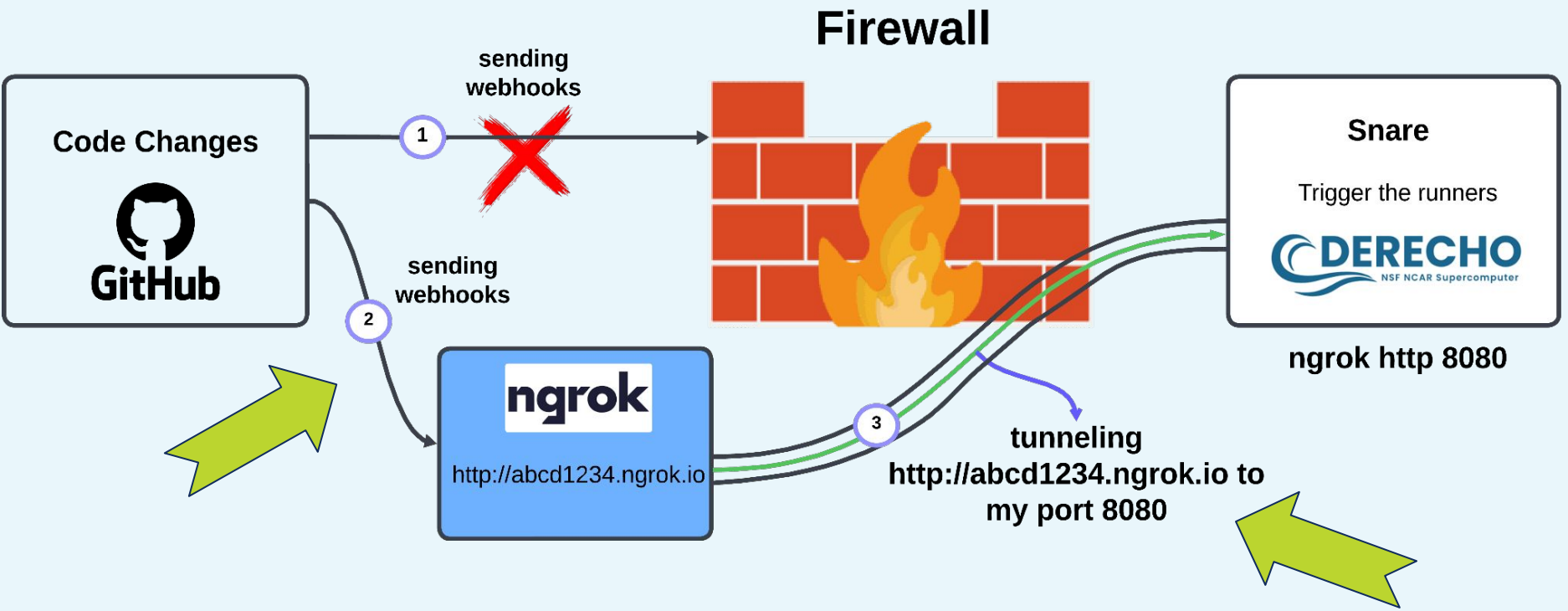
The screenshot shows the GitHub interface for the organization SIPARCS-CICD. The 'Repositories' tab is active, displaying a list of four private repositories. The repository 'Runner_from_2user' is highlighted with a red callout box. The repository 'CM1-Tri' is also highlighted with a red callout box. The repository 'Testing-Runners' is highlighted with a red callout box. The repository 'Parallel-Computing' is highlighted with a red callout box. The repository 'Runner_from_2user' is highlighted with a red callout box.

Repository Name	Visibility	Owner	License	Forks	Stars	Issues	Pull Requests	Last Updated
CM1-Tri	Private	Roff	MIT License	0	0	0	0	Updated now
Testing-Runners	Private			0	0	0	0	Updated 3 minutes ago
Parallel-Computing	Private	Shell		0	0	0	0	Updated 4 minutes ago
Runner_from_2user	Private			1	0	0	1	Updated 2 weeks ago

Code Changed

2nd user repo

Autoscaling Runners with Webhooks



Results

```
Checking for repositories with active workflows...
Repository with active workflows: Parallel-Computing
Workflow Name: Submit PBS Jobs
Jobs:
Job Name: build-and-submit (1)
Job Name: build-and-submit (2)
Job Name: build-and-submit (3)
Job Name: build-and-submit (4)
Repository with active workflows: CM1-Tri
Workflow Name: Project Builds
Jobs:
Job Name: test_without_container_derecho
Workflow Name: tringuyen180303 is building project on https://github.com
Jobs:
Job Name: test_without_container_derecho
Repository with active workflows: Testing-Drivers
Workflow Name: Scale test
Jobs:
Job Name: scale-test1
Job Name: scale-test3
Job Name: scale-test4
Job Name: scale-test2
Job Name: scale-test5
Job Name: scale-test6
```

The diagram illustrates the execution of workflows and their associated jobs. A central yellow box labeled "Jobs running inside a workflow" has three blue arrows pointing to the workflow names: "Parallel-Computing", "CM1-Tri", and "Testing-Drivers". Additionally, three yellow arrows point from this central box to the job lists of each workflow: "build-and-submit (1-4)", "test_without_container_derecho", and "scale-test1-6".

Update scale-test-webhook-2.yaml #25

Summary

Jobs

- scale-test1
- scale-test2
- scale-test3
- scale-test4
- scale-test5
- scale-test6

Run details

- Usage
- Workflow file

Triggered via push 4 minutes ago

Status: **In progress**

tringuyen180303 pushed -> 8484125 main

scale-test-webhook-2.yaml
on: push

scale-test1	2m 33s
scale-test2	1m 45s
scale-test3	2m 7s
scale-test4	1m 21s
scale-test5	59s
scale-test6	28s

Runners

Includes all runners across self-hosted and GitHub-hosted runners.

Host your own runners and customize the environment used to run jobs in your GitHub Actions workflows. Runners added to this organization can be used to process jobs in multiple repositories in your organization. [Learn more about self-hosted runners.](#)

Q Search runners New runner ▾

Runners	Status
Standard GitHub-hosted runners Ready-to-use runners managed by GitHub. Learn more.	● 0 active jobs
New-runner-20240724094956-6 self-hosted X64 macOS no-gpu Runner group: Default	● Active ...
New-runner-20240724094934-5 self-hosted X64 macOS no-gpu Runner group: Default	● Active ...
New-runner-20240724094911-4 self-hosted X64 macOS no-gpu Runner group: Default	● Active ...
New-runner-20240724094849-3 self-hosted X64 macOS no-gpu Runner group: Default	● Active ...
New-runner-20240724094823-2 self-hosted X64 macOS no-gpu Runner group: Default	● Active ...
New-runner-20240724094757-1 self-hosted X64 macOS no-gpu Runner group: Default	● Active ...
New-runner-20240724094714-1 self-hosted X64 macOS no-gpu Runner group: Default	● Active ...

Challenges

- Implementing Kubernetes cluster inside rootless container
- Github API response time is highly variable (10 seconds ~ 10 minutes)
- Assigned runners sometimes lingering forever



Future Work

- Develop robust authentication for user mapping in PBS schedulers
- Explore Github Teams and Enterprise for infrastructure
- Replacing ngrok with production public access method



Thank you

- **Mentors: Haiying Xu**
Brian Vanderwende
- **Technical Support: Nick Cote**
- **SIParCS 2024**



- Billing and plans
- Repository roles
- Member privileges
- Import/Export
- Moderation

Code, planning, and automation

- Repository
- Codespaces
- Planning
- Copilot
- Actions
 - General
 - Runners
 - Runner groups
 - Caches
- Webhooks
- Discussions
- Packages
- Pages

Security

- Authentication security
- Code security
- Compliance

Host your own runners and customize the environment used to run jobs in your GitHub Actions workflows. Runners added to this organization can be used to process jobs in multiple repositories in your organization. [Learn more about self-hosted runners.](#)

Search runners

New runner

Runners	Status
Standard GitHub-hosted runners Ready-to-use runners managed by GitHub. Learn more	0 active jobs
my-awesome-scale-set-vm Runner group: Default	Online
my-awesome-scale-set-vm -dt2bp-runner-4tth5 Runner group: Default	Idle
my-awesome-scale-set-vm -dt2bp-runner-4kr4h Runner group: Default	Idle
my-awesome-scale-set-vm -dt2bp-runner-5blr5 Runner group: Default	Idle
my-awesome-scale-set-vm -dt2bp-runner-c9jrc Runner group: Default	Idle
my-awesome-scale-set-vm -dt2bp-runner-sdb9f Runner group: Default	Idle
my-awesome-scale-set-vm -dt2bp-runner-prl28 Runner group: Default	Idle
my-awesome-scale-set-vm -dt2bp-runner-nczzt Runner group: Default	Idle
my-awesome-scale-set-vm -dt2bp-runner-lfv7k Runner group: Default	Idle
my-awesome-scale-set-vm -dt2bp-runner-lhst8 Runner group: Default	Idle
my-awesome-scale-set-vm -dt2bp-runner-5fgm9 Runner group: Default	Idle

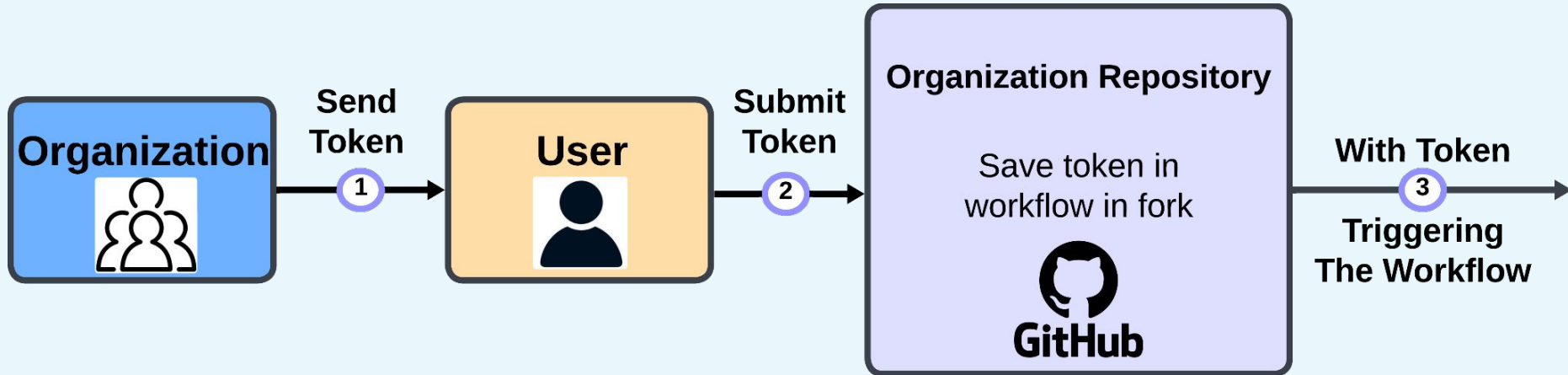
Runner name

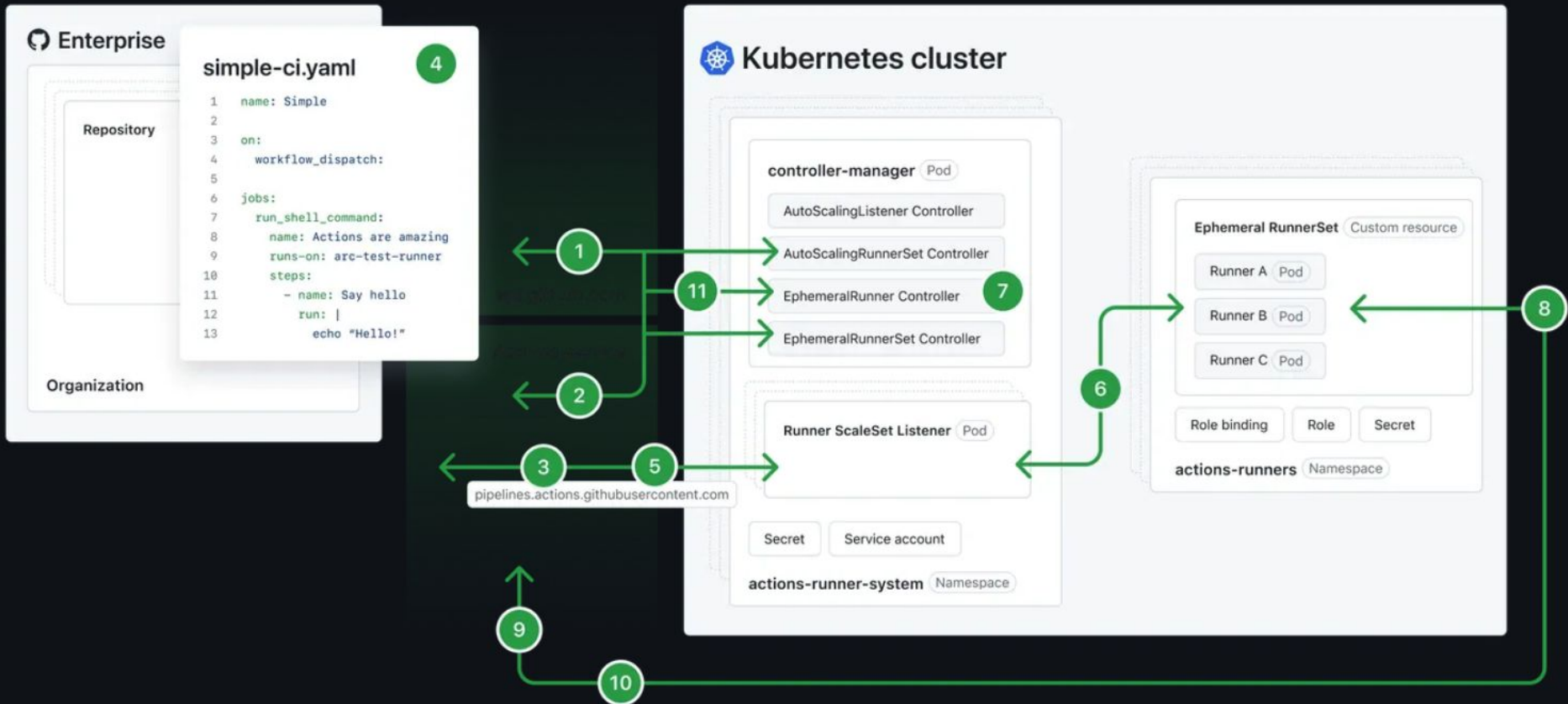


Autoscaling for HPC Runners



How to Autoscale the Organization Runners?





Autoscaling for HPC Runners

