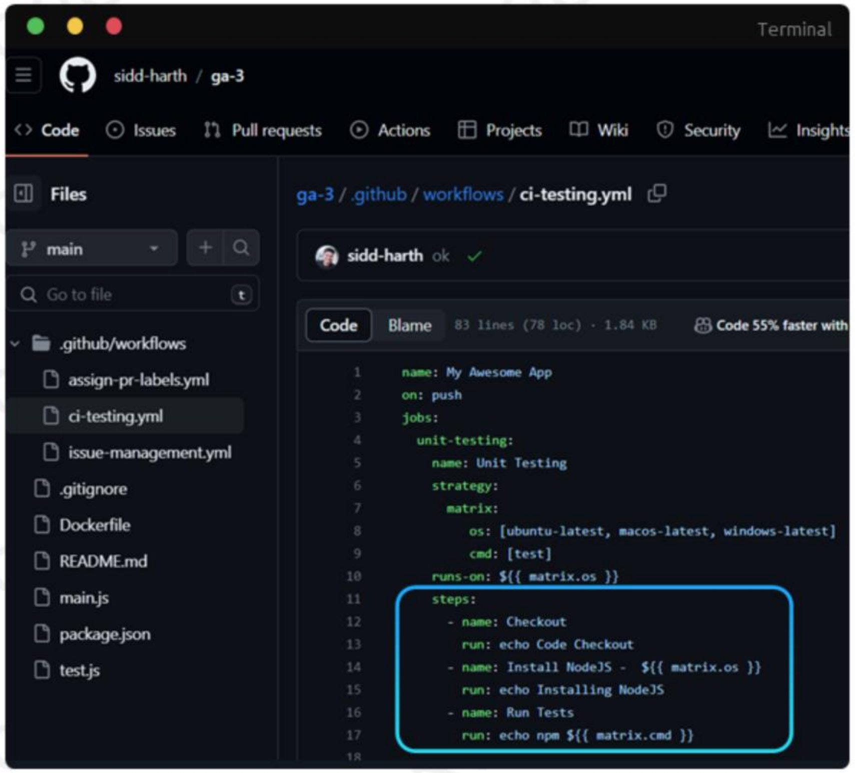
KodeKloud

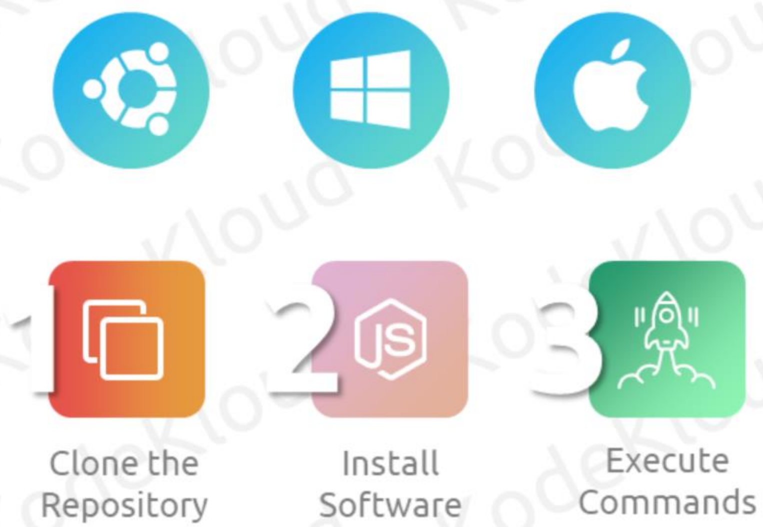
Follow us on https://kodekloud.com/ to learn more about us.

Types of Runners

GitHub Runners



```yaml
name: My Awesome App
on: push
jobs:
  unit-testing:
    name: Unit Testing
    strategy:
      matrix:
         os: [ubuntu-latest, macos-latest, windows-latest]
         cmd: [test]
    runs-on: ${{ matrix.os }}
    steps:
      - name: Checkout
        run: echo Code Checkout
      - name: Install NodeJS -  ${{ matrix.os }}
        run: echo Installing NodeJS
      - name: Run Tests
        run: echo npm ${{ matrix.cmd }}
```

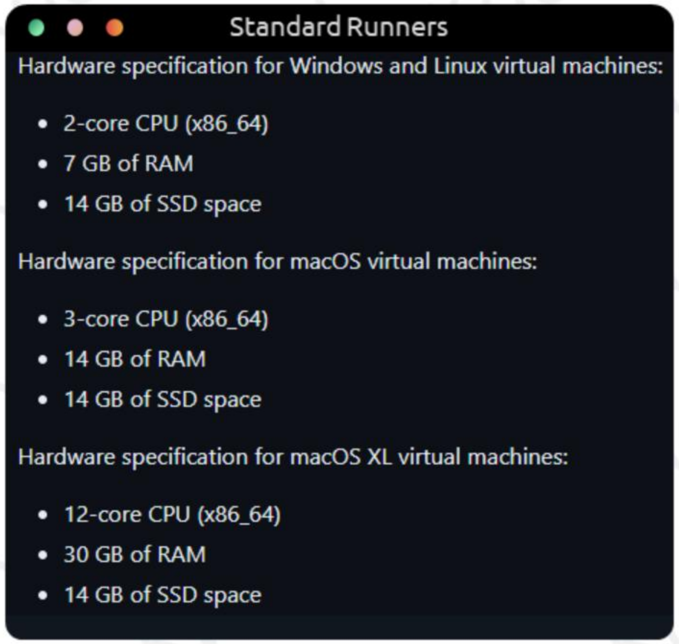


Up to this point, we've utilized github hosted runners to execute our workflows.

Runners are essentially virtual machines responsible for performing various tasks within a GitHub Actions workflow. For example, a runner can handle tasks such as cloning your repository, installing necessary software, and executing specified commands.

Each GitHub-hosted runner is essentially a fresh virtual machine that GitHub hosts. It comes pre-equipped with the runner application and other essential tools, and you have the option to choose from different operating systems, including Ubuntu Linux, Windows, or macOS.

Hardware Resources Supported by GitHub-Hosted Runners



Standard Runners

Hardware specification for Windows and Linux virtual machines:

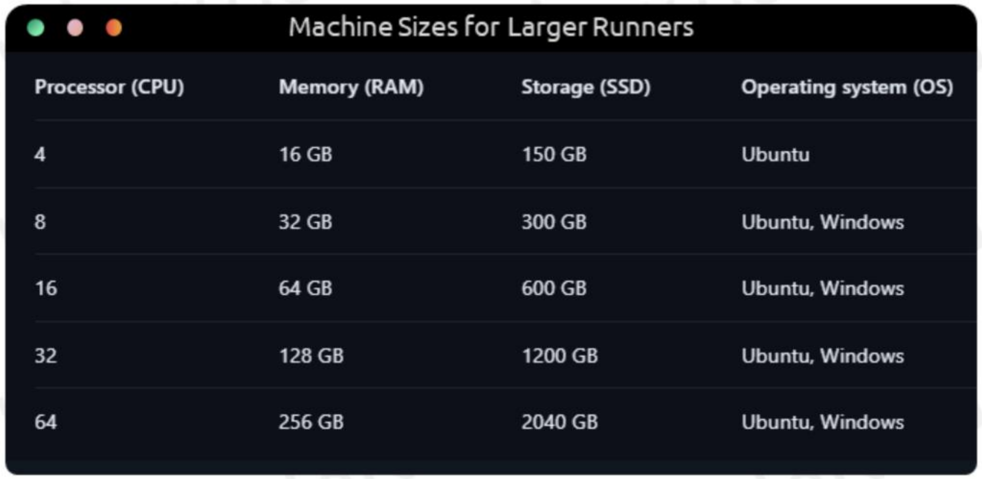
- 2-core CPU (x86_64)
- 7 GB of RAM
- 14 GB of SSD space

Hardware specification for macOS virtual machines:

- 3-core CPU (x86_64)
- 14 GB of RAM
- 14 GB of SSD space

Hardware specification for macOS XL virtual machines:

- 12-core CPU (x86_64)
- 30 GB of RAM
- 14 GB of SSD space



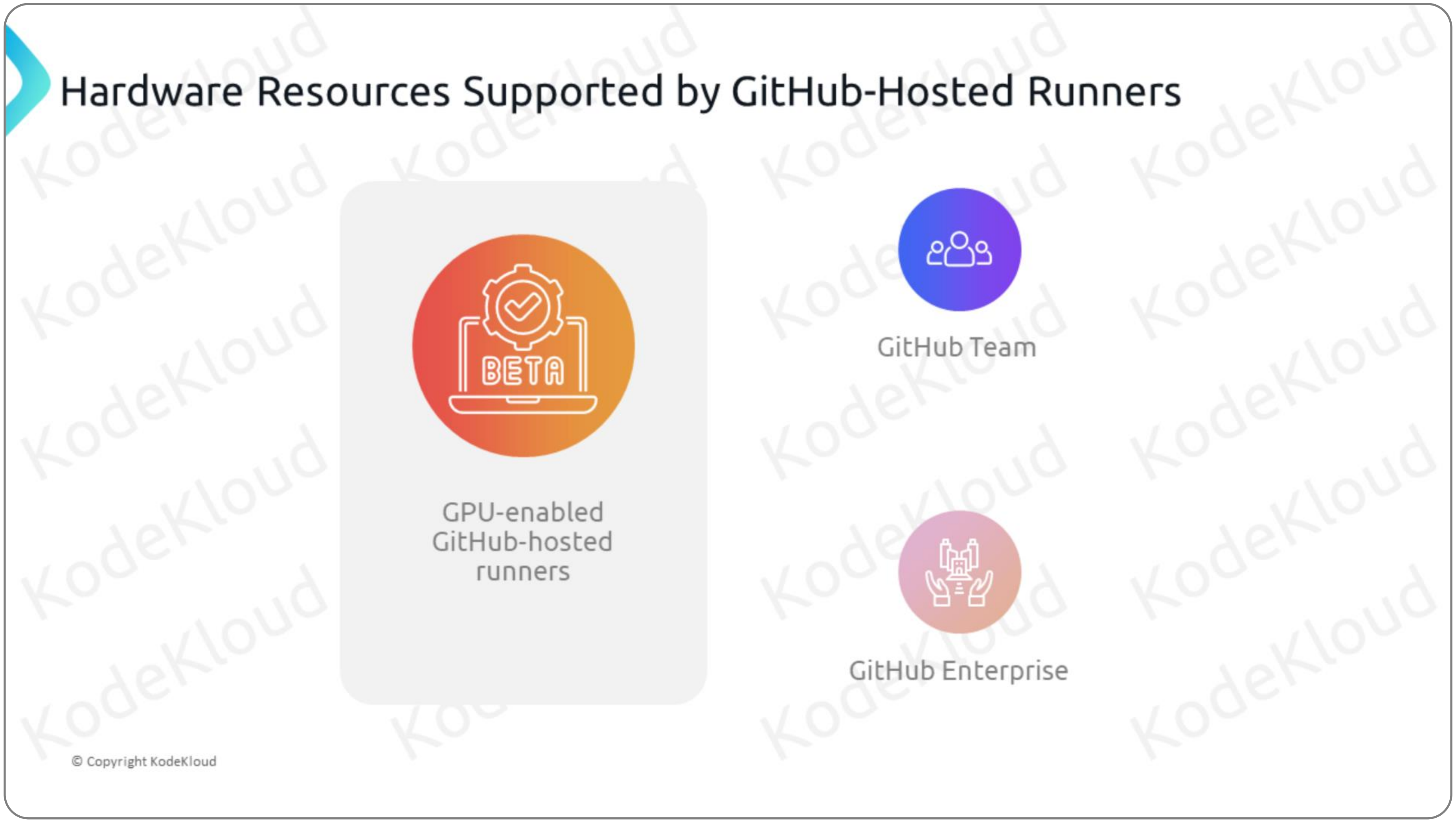
Machine Sizes for Larger Runners

Processor (CPU)	Memory (RAM)	Storage (SSD)	Operating system (OS)
4	16 GB	150 GB	Ubuntu
8	32 GB	300 GB	Ubuntu, Windows
16	64 GB	600 GB	Ubuntu, Windows
32	128 GB	1200 GB	Ubuntu, Windows
64	256 GB	2040 GB	Ubuntu, Windows

GitHub Hosted Runners come in two types. The first is the standard runner, which has limited resources in terms of RAM, CPU, and disk space. We've been using these standard runners up to this point.

In addition to the standard GitHub-hosted runners, GitHub provides a range of managed virtual machines with greater RAM, CPU, and disk space for customers on GitHub Team and GitHub Enterprise Cloud plans. These runners are also hosted by GitHub and come with the same runner application and necessary tools preinstalled.

However, as of September 2023, it's important to note that GitHub does not currently offer GitHub-hosted runners with GPU capabilities.



Hardware Resources Supported by GitHub-Hosted Runners

GPU-enabled GitHub-hosted runners

GitHub Team

GitHub Enterprise

Nevertheless, there's an ongoing development in the form of a beta program for GPU-enabled GitHub-hosted runners.

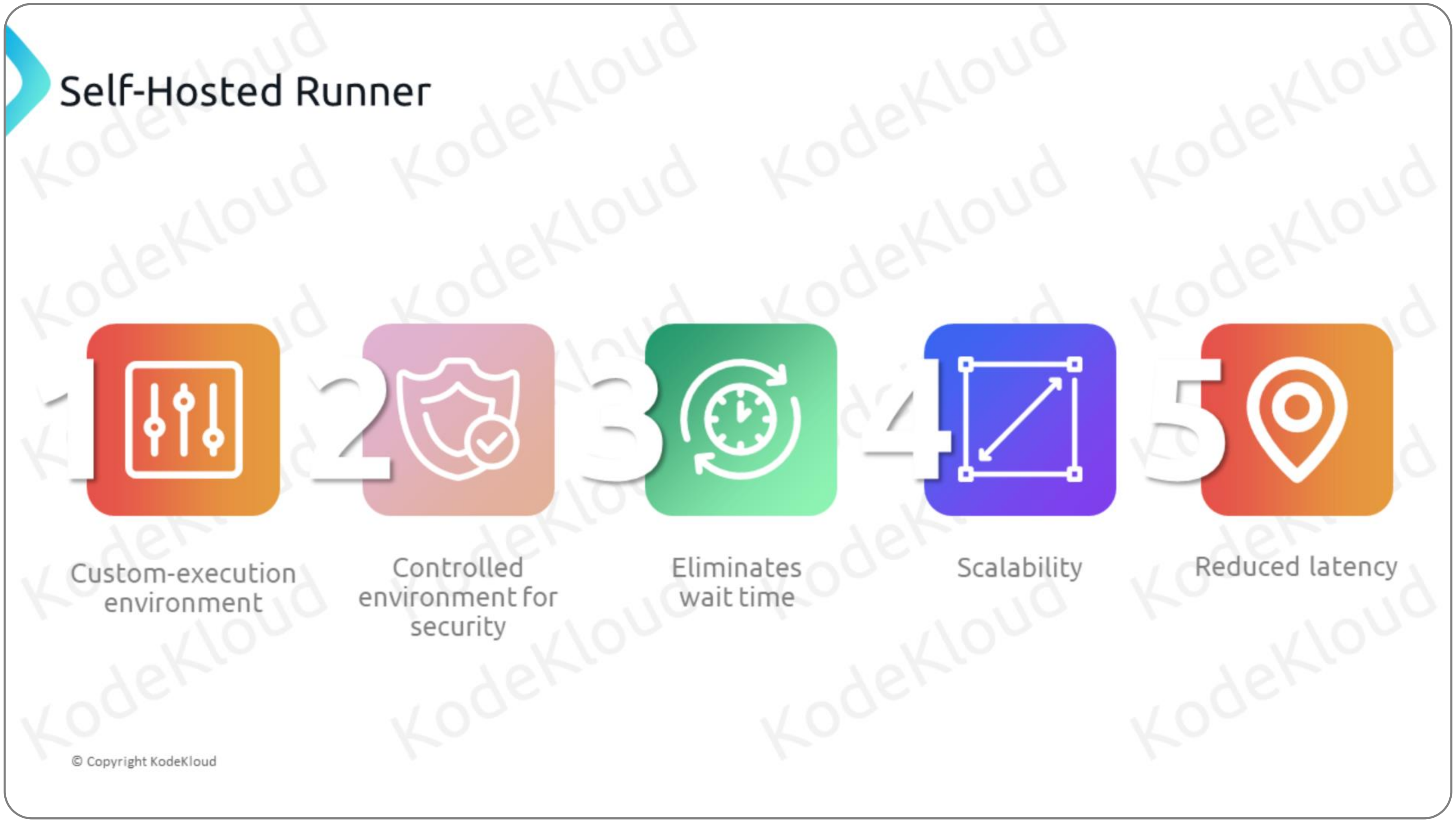
To participate in the beta program, you must have a GitHub Team or GitHub Enterprise Cloud plan. You can then request access to the beta program by filling out a form.

Once you have been accepted into the beta program, you will be able to create runner groups that include GPU machines. You can then use these runner groups to run your workflows on GPU machines.

It is important to note that the GPU-enabled GitHub-hosted runners are still in beta, so there may be some instability. However, GitHub is working to improve the stability of these runners and to make them available to more users in the future.

If you're not accepted into the beta program or have GPU requirements and need more control, you can opt for self-hosted runners.

Now, what exactly are self-hosted runners?



Self-Hosted Runner

1
Custom-execution environment
2
Controlled environment for security
3
Eliminates wait time
4
Scalability
5
Reduced latency

Self-hosted runners are machines that you deploy and manage yourself. This gives you more control over the hardware, operating system, and software tools that your workflows run on. Self-hosted runners can be run on any operating system, including custom operating systems.

Here are some reasons why you might choose to use self-hosted runners:

1. Self-hosted runners allow you to create and maintain **custom execution environments** tailored to your specific project requirements. This is especially useful if your project relies on specific software configurations, dependencies, or hardware that GitHub-hosted runners don't provide.

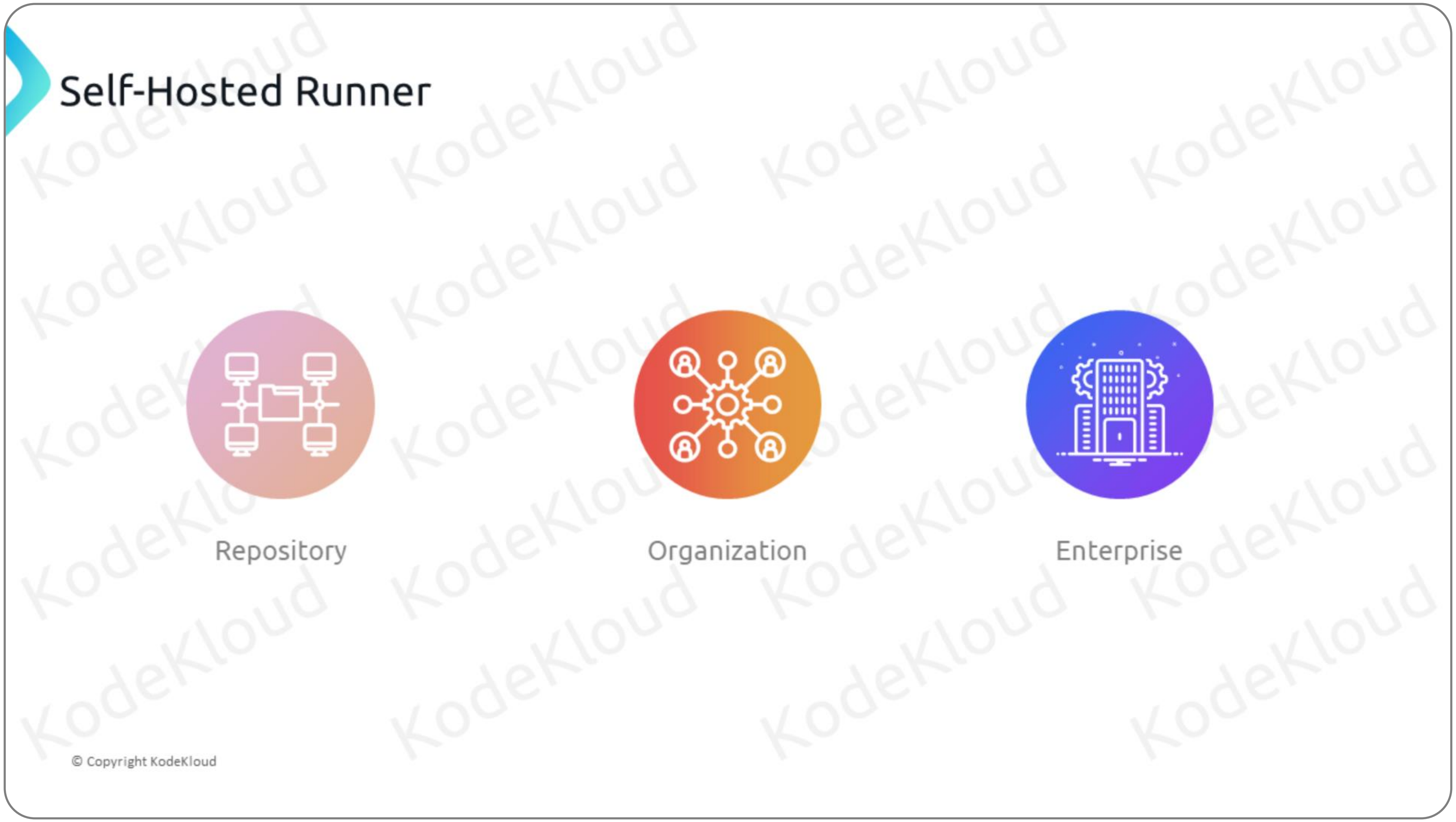
2. In some organizations, **strict security and compliance policies** may prevent the use of GitHub-hosted runners for certain workflows. Self-hosted runners can be set up in a controlled environment that meets your organization's security and compliance requirements.

3. Depending on the demand for GitHub-hosted runners in your organization, there might be wait times for available runners. Self-hosted runners eliminate this wait time, as they are dedicated to your projects.

4. You can scale your self-hosted runner pool based on your project's needs. This can be particularly useful if you have many concurrent workflows or if you need to run multiple workflows in parallel, as GitHub-hosted runners have predefined concurrency limits.

5. Self-hosted runners can be placed in specific geographic locations to reduce latency for your CI/CD workflows, especially if you have a global user base or need to comply with data residency regulations.

In summary, while GitHub-hosted runners are convenient and suitable for many use cases, self-hosted runners provide more control, flexibility, and customization options.



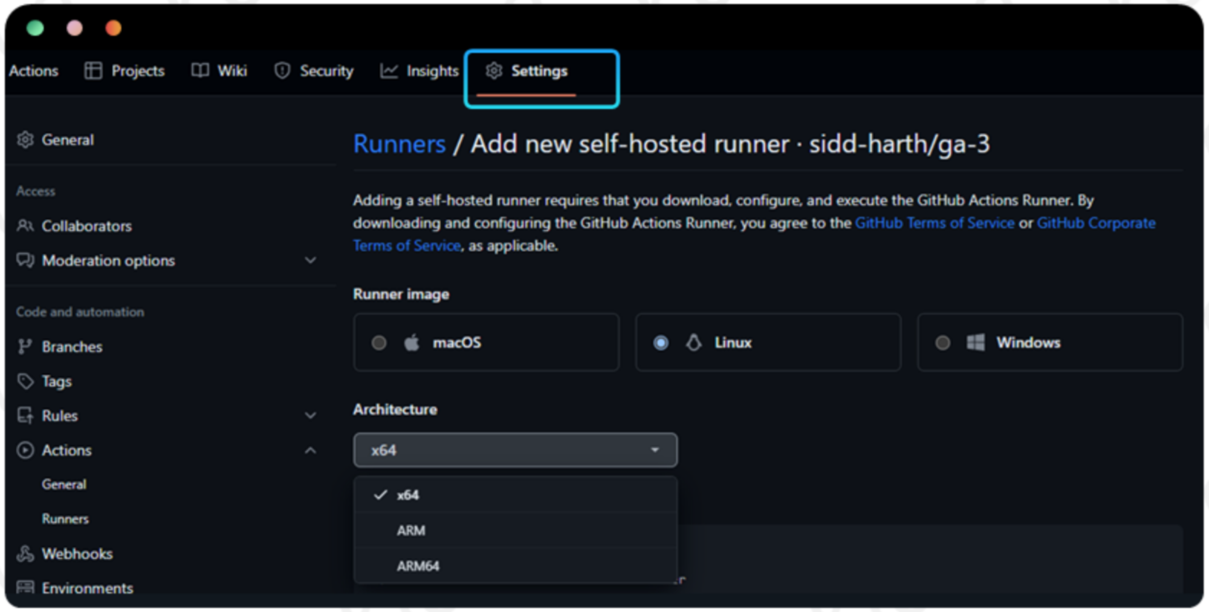
Self-Hosted Runner

In summary, while GitHub-hosted runners are convenient and suitable for many use cases, self-hosted runners provide more control, flexibility, and customization options.

The next step is to understand how to configure self-hosted runners in the context of GitHub Actions.

Self-hosted runners can be added to a github repository, an organization, or an enterprise.

Self-Hosted Runner



Runners / Add new self-hosted runner · sidd-harth/ga-3

Adding a self-hosted runner requires that you download, configure, and execute the GitHub Actions Runner. By downloading and configuring the GitHub Actions Runner, you agree to the GitHub Terms of Service or GitHub Corporate Terms of Service, as applicable.

Runner image

○ macOS [✓ Linux] ○ Windows

Architecture

x64

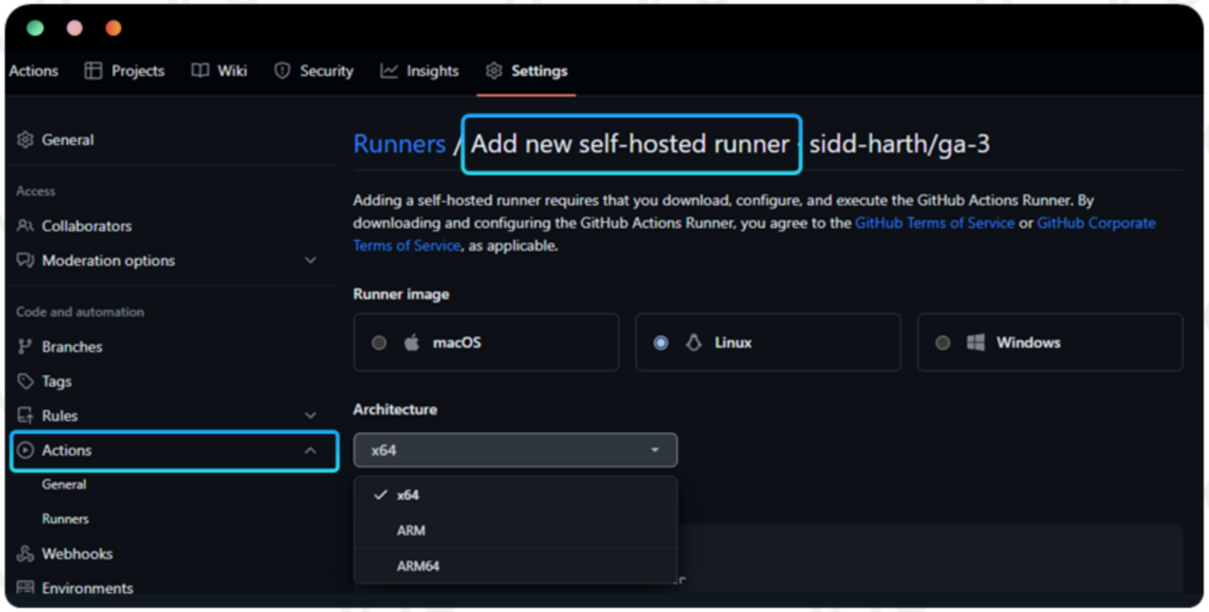
✓ x64
ARM
ARM64

To set up a self-hosted runner within a specific repository, you can navigate to the repository's settings. There, you'll find an option to create a new self-hosted runner under the "Actions" section.

During this setup, you'll have the opportunity to choose the operating system image and architecture for your self-hosted runner machine.

This provides instructions guiding you through the process of obtaining and installing the runner application on your self-hosted runner machine.

Self-Hosted Runner



Runners / Add new self-hosted runner · sidd-harth/ga-3

Adding a self-hosted runner requires that you download, configure, and execute the GitHub Actions Runner. By downloading and configuring the GitHub Actions Runner, you agree to the GitHub Terms of Service or GitHub Corporate Terms of Service, as applicable.

Runner image

macOS [✓ Linux] Windows

Architecture

x64

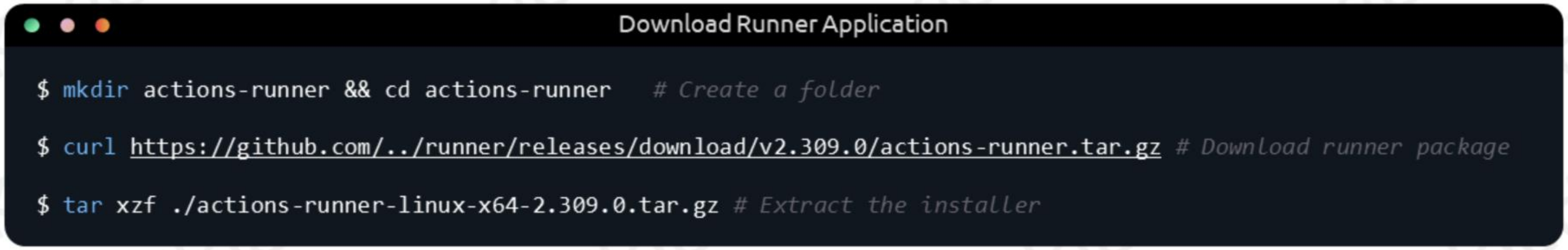
✓ x64
ARM
ARM64

To set up a self-hosted runner within a specific repository, you can navigate to the repository's settings. There, you'll find an option to create a new self-hosted runner under the "Actions" section.

During this setup, you'll have the opportunity to choose the operating system image and architecture for your self-hosted runner machine.

This provides instructions guiding you through the process of obtaining and installing the runner application on your self-hosted runner machine.

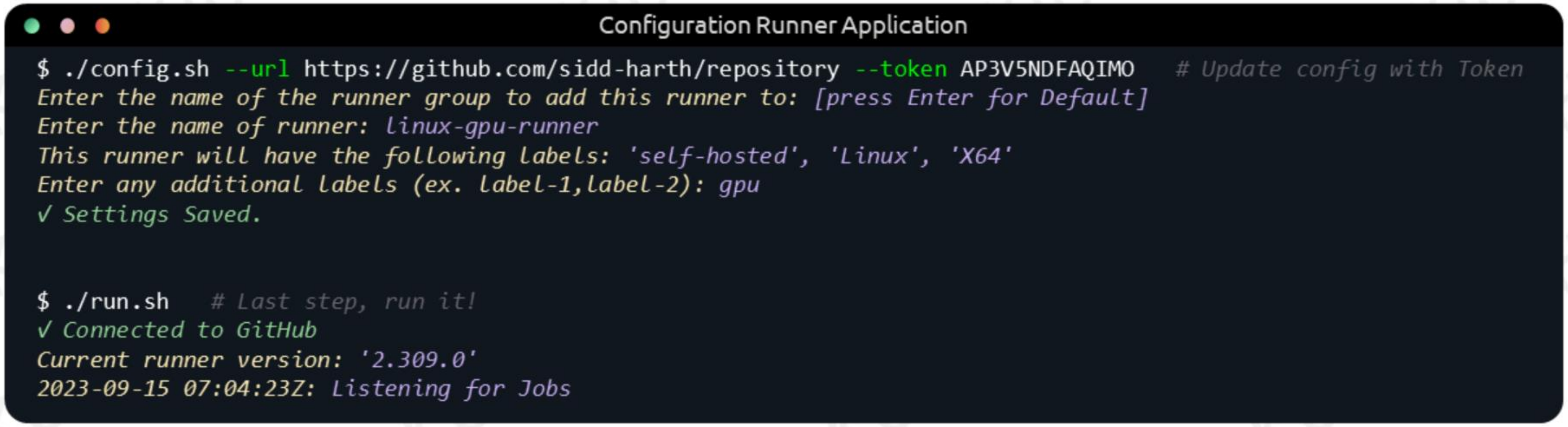
Self-Hosted Runner


Download Runner Application

```
$ mkdir actions-runner && cd actions-runner   # Create a folder

$ curl https://github.com/../runner/releases/download/v2.309.0/actions-runner.tar.gz # Download runner package

$ tar xzf ./actions-runner-linux-x64-2.309.0.tar.gz # Extract the installer
```


Configuration Runner Application

```
$ ./config.sh --url https://github.com/sidd-harth/repository --token AP3V5NDFAQIMO  # Update config with Token
Enter the name of the runner group to add this runner to: [press Enter for Default]
Enter the name of runner: linux-gpu-runner
This runner will have the following labels: 'self-hosted', 'Linux', 'X64'
Enter any additional labels (ex. label-1,label-2): gpu
√ Settings Saved.


$ ./run.sh   # Last step, run it!
√ Connected to GitHub
Current runner version: '2.309.0'
2023-09-15 07:04:23Z: Listening for Jobs
```

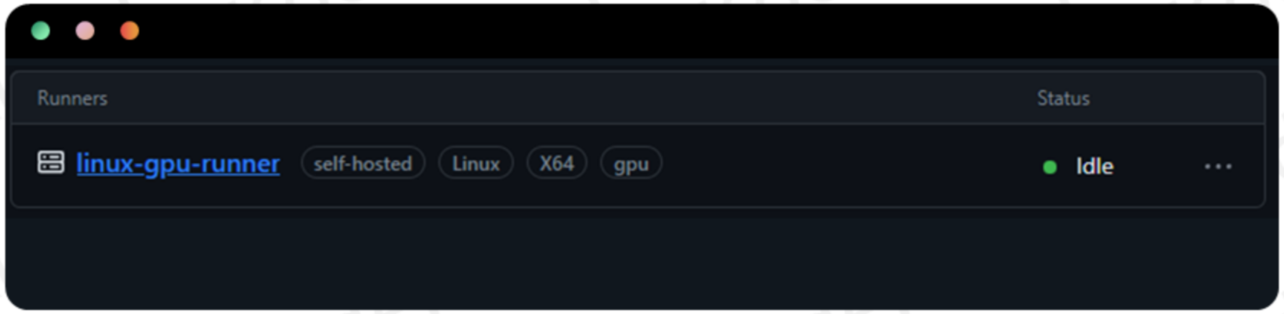
To set it up, you'll start by creating a new directory, downloading and extracting the GitHub runner package.

Next, you'll execute the "config" script to configure the self-hosted runner application and register it with GitHub Actions using an automatically-generated time-limited token to authenticate the request. This script will prompt you for specific details like the runner's name, default labels, and any additional labels needed to distinguish this runner from others.

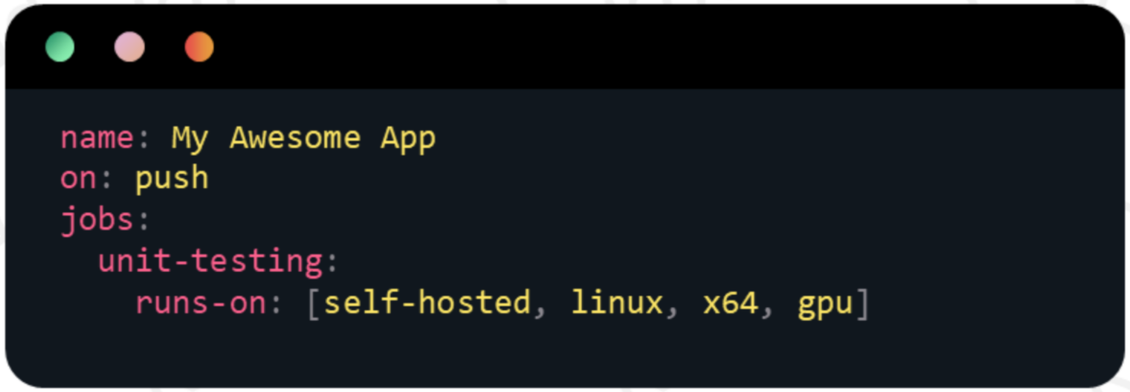
Since you're adding this self-hosted runner to a repository, the destination URL will point to the repository where it's being created.

Finally, execute the "run" script for the self-hosted runner application will establish the connection with GitHub Actions.

Self-Hosted Runner



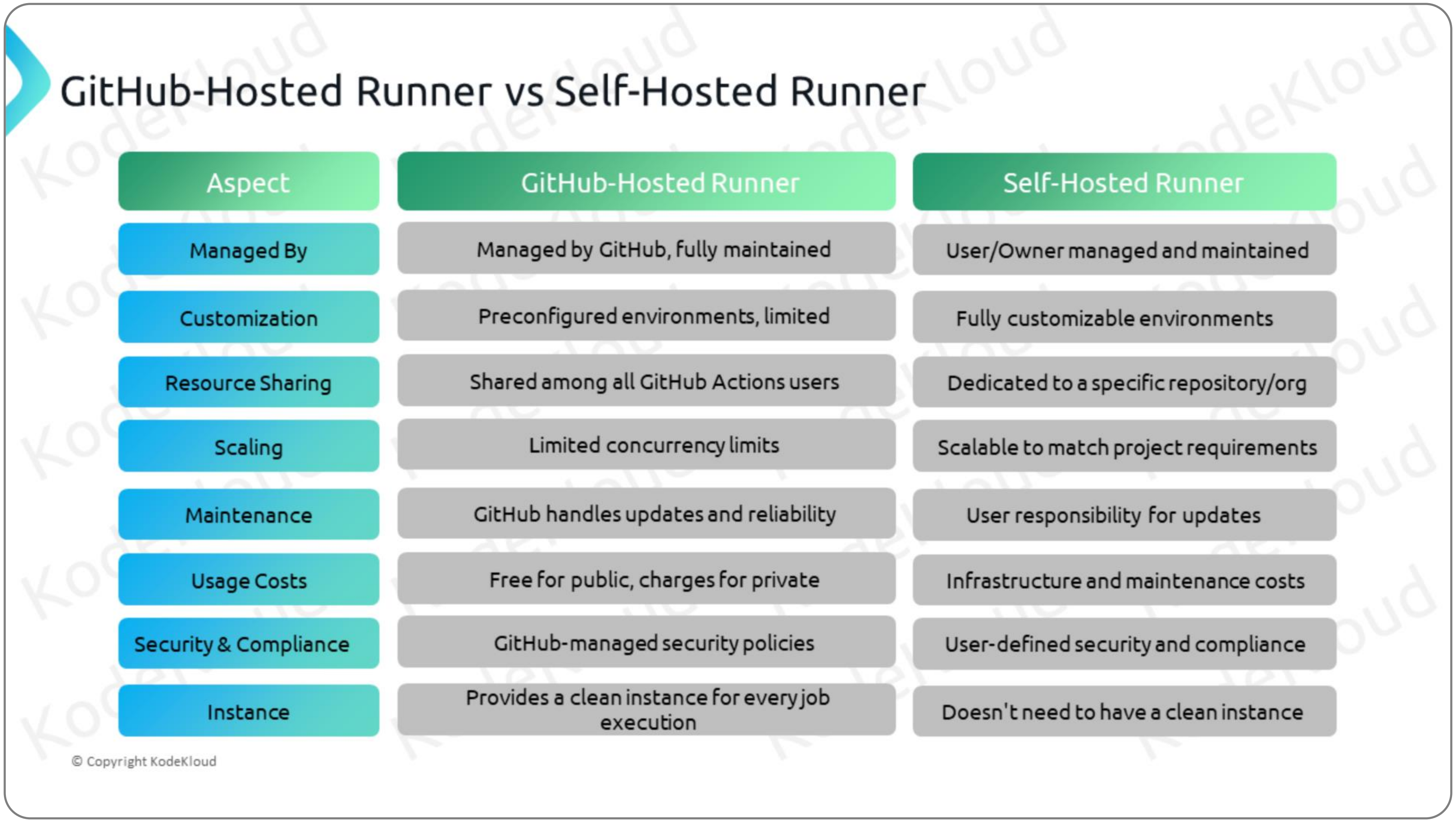
Runners	Status
linux-gpu-runner self-hosted Linux X64 gpu	● Idle



```yaml
name: My Awesome App
on: push
jobs:
  unit-testing:
    runs-on: [self-hosted, linux, x64, gpu]
```

To check the status of this setup, you can go to the GitHub repository's settings page.

To utilize self-hosted runners in a workflow, you'll specify the desired labels in the "runs-on" field for each job.



GitHub-Hosted Runner vs Self-Hosted Runner

Aspect	GitHub-Hosted Runner	Self-Hosted Runner
Managed By	Managed by GitHub, fully maintained	User/Owner managed and maintained
Customization	Preconfigured environments, limited	Fully customizable environments
Resource Sharing	Shared among all GitHub Actions users	Dedicated to a specific repository/org
Scaling	Limited concurrency limits	Scalable to match project requirements
Maintenance	GitHub handles updates and reliability	User responsibility for updates
Usage Costs	Free for public, charges for private	Infrastructure and maintenance costs
Security & Compliance	GitHub-managed security policies	User-defined security and compliance
Instance	Provides a clean instance for every job execution	Doesn't need to have a clean instance

Lets look at some key differences between GitHub-hosted runners and self-hosted runners:

Managed By: GitHub-hosted runners are provided and maintained entirely by GitHub, while self-hosted runners are managed and maintained by the user or organization that sets them up.

Customization: GitHub-hosted runners come with predefined environments and limited customization options, while self-hosted runners allow for full customization, including hardware, software, and dependencies.

Resource Sharing: GitHub-hosted runners are shared among all GitHub Actions users and have concurrency limits, whereas self-hosted runners are dedicated to a specific repository or organization, ensuring resource availability.

Scaling: GitHub-hosted runners have predefined concurrency limits, while self-hosted runners can be scaled up or down to match the project's workload and resource needs.

Maintenance: GitHub handles updates and reliability for GitHub-hosted runners, whereas users are responsible for maintaining and updating self-hosted runners.

Usage Costs: GitHub-hosted runners are typically free for public repositories and have usage charges for private repositories. Self-hosted runners incur infrastructure and maintenance costs but no usage charges.

Security & Compliance: GitHub-hosted runners adhere to GitHub's security policies, while self-hosted runners allow users to define and enforce their own security and compliance measures.

These differences highlight the trade-offs between convenience and control when choosing between GitHub-hosted and self-hosted runners in GitHub Actions. The choice depends on your specific project requirements and constraints.

KodeKloud

Follow us on https://kodekloud.com/ to learn more about us.