GitHub Actions Runners

GitHub Actions provides the flexibility to run your workflows on different types of runners. Runners are the compute environments where your workflow jobs execute. There are two main categories of runners: GitHub-hosted runners and self-hosted runners.

GitHub Hosted Runners

What are GitHub Hosted Runners?

GitHub-hosted runners are virtual machines provided and managed by GitHub. They are available for your use without the need for any additional setup. These runners come in various environments, including Ubuntu, Windows, and macOS, allowing you to build, test, and deploy your code in a familiar environment.

Key Aspects of GitHub Hosted Runners

- **Availability**: GitHub-hosted runners are readily available, making them a convenient choice for many projects.
- **Maintenance**: GitHub takes care of the maintenance, updates, and scaling of these runners, ensuring they are always up to date and available for your workflows.
- **Usage Limitations**: There are usage limitations for GitHub-hosted runners, including maximum execution time and the number of concurrent jobs per repository.

Self Hosted Runners

What are Self Hosted Runners?

Self-hosted runners are machines that you provision and manage yourself, either on your own infrastructure or in a cloud environment. These runners give you full control over the runner environment, allowing you to customize it to match your project's specific requirements.

Key Aspects of Self Hosted Runners

- **Customization**: Self-hosted runners can be customized with the software, tools, and configurations needed for your project. This makes them suitable for projects with specific dependencies or hardware

requirements.

- **Security**: You have full control over the security and access controls of self-hosted runners. You can choose where they run and who has access to them.
- **Maintenance**: Self-hosted runners require ongoing maintenance, including updates and monitoring, which is your responsibility.

Configure Self hosted runner

Enable WSL Integration with Ubuntu on Docker Desktop:

- Open Docker Desktop.
- Go to Settings.
- Navigate to Resources -> WSL Integration.
- Enable integration with additional distros and select Ubuntu.

Download and Extract the GitHub Actions Runner:

- Open Windows Subsystem for Linux 2 (WSL2)
- Open a terminal or WSL shell.
- Go to the Settings tab of your repository on GitHub.
- Navigate to the "Actions" section.
- Select "Add runner" to initiate the process of adding a self-hosted runner.
- Choose the appropriate options:
    - Operating System: Select Linux.
    - Architecture: Select x64 (assuming your system architecture is 64-bit).
- After selecting Linux, follow the steps to set up the self-hosted runner.
- After configuration, the self-hosted runner should be operational and ready to execute workflows for your repository. You can use it in your workflow YAML files by specifying `runs-on: self-hosted`.

Choosing Between GitHub Hosted and Self Hosted Runners

The choice between GitHub-hosted and self-hosted runners depends on your project's needs:

- Use **GitHub-Hosted Runners** if your project is hosted on GitHub, and you need a convenient and managed environment for common use cases. These runners are a great choice for most projects.
- Use **Self-Hosted Runners** if you require custom configurations, have specific hardware requirements, or need full control over the runner environment. Self-hosted runners are suitable for complex or specialized projects.

Configuring Runners in Your Workflow

1. Create a Workflow File:

- Create a new file named `05-01-Workflow-Runners.yml` in the `.github/workflows` directory of your repository.

```yaml
name: 05-01-Workflow Runners

# This workflow is triggered manually via the GitHub UI (workflow_dispatch)
on: workflow_dispatch

jobs:
  # Job running on Ubuntu
  ubuntu-echo:
    runs-on: ubuntu-latest
    steps:
      - name: Show OS
        run: |
          echo "This job is running on an Ubuntu runner."
          echo "Runner OS: $RUNNER_OS" # Displaying the runner OS
environment variable

  # Job running on Windows
  windows-echo:
    runs-on: windows-latest
    steps:
      - name: Show OS
        shell: bash # Using bash shell on Windows
        run: |
          echo "This job is running on a Windows runner."
          echo "Runner OS: $RUNNER_OS" # Displaying the runner OS
environment variable

  # Job running on MacOS
  mac-echo:
    runs-on: macos-latest
    steps:
      - name: Show OS
        run: |
          echo "This job is running on a MacOS runner."
          echo "Runner OS: $RUNNER_OS" # Displaying the runner OS
environment variable

  # Job running on Self-Hosted runner
  self-echo:
    runs-on: self-hosted
    steps:
      - name: Show OS
        run: |
          echo "This job is running on a Self-Hosted runner."
          echo "Runner OS: $RUNNER_OS" # Displaying the runner OS
environment variable
```

- Copy and paste the provided YAML configuration into this file.

2. Understanding the Workflow:

- This workflow is triggered manually via the GitHub UI (`workflow_dispatch`).
- It defines four jobs, each running on a different operating system environment: `Ubuntu`, `Windows`, `MacOS`, and `Self-Hosted`.
- Each job consists of a single step that echoes the operating system of the runner environment.

3. Testing the Workflow:

- Commit and push the workflow file (`05-01-Workflow-Runners.yml`) to your repository.
- Go to the "Actions" tab in your GitHub repository to view the workflow runs.
- Manually trigger the workflow by clicking the "Run workflow" button and selecting the desired workflow (`05-01-Workflow-Runners`) from the dropdown list.
- Monitor the workflow runs and check the output of each job to ensure it correctly displays the operating system of the runner environment.

4. Reviewing the Output:

- For the Ubuntu, Windows, and MacOS jobs, verify that the output displays the respective operating system (Ubuntu, Windows, or MacOS).
- For the Self-Hosted job, if you have set up a self-hosted runner for your repository, ensure that the output displays "Self-Hosted" as the runner OS.
- If any job fails to execute or the output is unexpected, review the workflow configuration and troubleshoot any issues.