



UbuCon Korea 2023

# 내 Ubuntu 개발환경을 GitHub Actions에 연결하기

박동하, C++ Korea Community

# 커뮤니티 소개



## C++ Korea

C++ Korea Community is where you can ask questions about C++ programming language, in Korean

👤 147 followers 📍 Seoul, Korea 🔗 <https://fb.com/groups/cppkorea/> 🐦 @CppKorea 🔗 <https://discord.gg/87SNegGZue> ✉ [contact@cppkorea.org](mailto:contact@cppkorea.org)

- C++ 프로그래밍 언어 “한국어로” 질문하는 곳 [fb.com/groups/cppkorea](https://fb.com/groups/cppkorea/)
  - 주의: 답변 없을수도 있음
- 판데믹 이전에는 학생/직장인 스터디를 중심으로 활동 [github.com/cppkorea](https://github.com/cppkorea)
- 최근에는 ...
  - 격월 MeetUp 진행 [festa.io/hosts/40](https://festa.io/hosts/40)
  - Discord를 시도하는 중 [discord.gg/87SNegGZue](https://discord.gg/87SNegGZue)



## C++ Korea Community 운영위원

- Windows, WSL 환경에서 주로 개발

## 주요 관심사

- 오픈소스 프로젝트들의 공급망(Supply Chain) 관리
- 개발조직에서 소프트웨어 자본(S/W Capital) 구축

## 요즘은...

- 좀 더 완결성있게 일하는 방법 (육각형 개발자?)



**Park DongHa**

luncliff

C++, C#, Go Developer | Member of  
@CppKorea | Working @line



GitHub Actions라는 서비스에 내 **Ubuntu**를 연결해서 사용하는 방법

1. 몇번의 짧은 Demo

Self-hosted Runner를 사용할 때 고민거리

1. 오픈소스 프로젝트들이 사용하는 환경과 개인 환경의 비교
2. 다수의 개발자를 위한 환경을 구축할 때
3. 다수의 환경을 사용하는 작업을 설계할 때



## GitHub Actions

- <https://docs.github.com/en/actions>
  - [/hosting-your-own-runners](https://docs.github.com/en/actions/hosting-your-own-runners)
  - [/hosting-your-own-runners/managing-self-hosted-runners/about-self-hosted-runners](https://docs.github.com/en/actions/hosting-your-own-runners/managing-self-hosted-runners/about-self-hosted-runners)
- <https://github.blog/?s=Self-Hosted>
  - <https://github.blog/changelog/label/self-hosted/>
- <https://github.com/actions/runner-images>
- <https://youtube.com/@GitHub>
- [GitHub Actions로 개발 주기 자동화 - Microsoft Developer Korea](#)



## Continuous Integration/Delivery/Deployment

- <https://www.redhat.com/ko/topics/devops?cicd=32h281b>
- <https://www.redhat.com/ko/topics/devops/what-is-ci-cd>

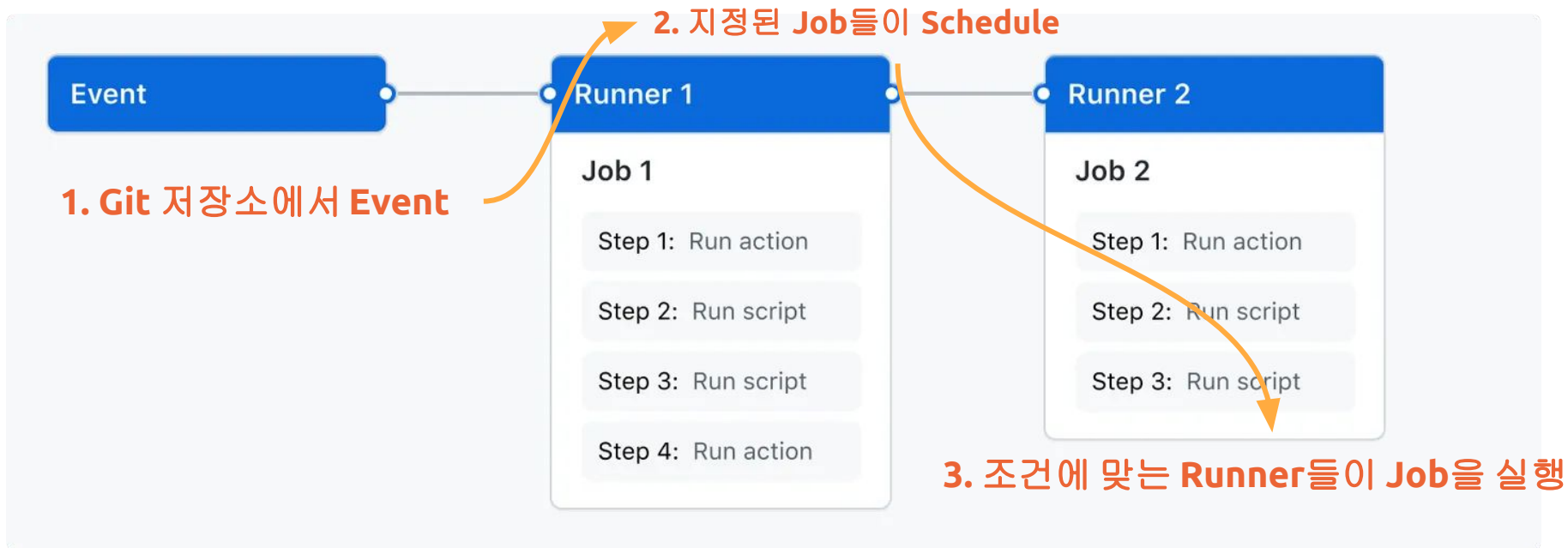


# 1. 매뉴얼 읽어보기

# Manual: GitHub Actions



CI/CD 플랫폼. 작업 흐름 Workflow의 자동화를 지원





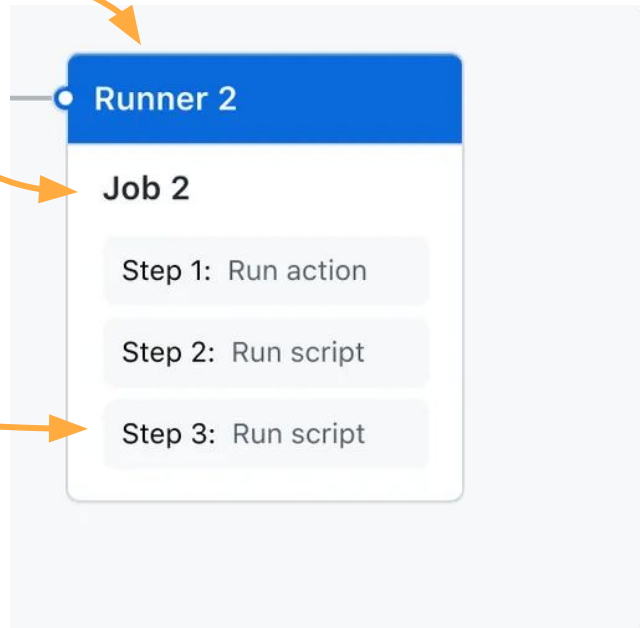
# Manual: GitHub Actions



**Runner:** Job을 실행하는 Machine

**Job:** 여러 Step들의 조합과 Runner 제약사항

**Step:** Action 또는 Script



# Manual: GitHub Actions



## Step: Action 또는 Script

- Action: 재사용 가능한 App. YAML, JavaScript, Dockerfile로 작성
- Script: Bash, PowerShell, Cmdlet 등의 CLI 입력

```
- uses: ConorMacBride/install-package@v1  
  with:  
    apt: build-essential procs curl file git gcc
```

Action

```
- name: "Install Homebrew"  
  run: |  
    /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"  
    eval "$(/home/linuxbrew/.linuxbrew/bin/brew shellenv)"  
    brew --version
```

Script

# 각 플랫폼마다 서로 대응하는 이름이 있음



굳이 외우기보다는, 개념들만 알아두어도 좋습니다

- Azure Pipelines

<https://learn.microsoft.com/en-us/azure/devops/pipelines/get-started/what-is-azure-pipelines>

- GitLab CI/CD

<https://docs.gitlab.com/ee/ci/index.html>

그 외에도 여러 CI 서비스가 상품을 판매하고, 기술지원을 제공

- <https://github.com/marketplace?category=continuous-integration>



## Quickstart

> Learn GitHub Actions

> Examples

> Using workflows

> Using jobs

> Manage workflow runs

> Build and test

> Deployment

> Containerized services

> Publish packages

> Manage issues and pull requests

> Migrate to GitHub Actions

> Monitor & troubleshoot

> GitHub-hosted runners

> Self-hosted runners

> Security guides

> Creating actions

Guides for GitHub Actions

GitHub Actions 전반에 대한 내용

Workflow, Job YAML 명세, 표현식 등

각 **Phase**에서의 사용예시들(프로그래밍 언어, 시나리오 등)

모니터링 (Web UI) 설명

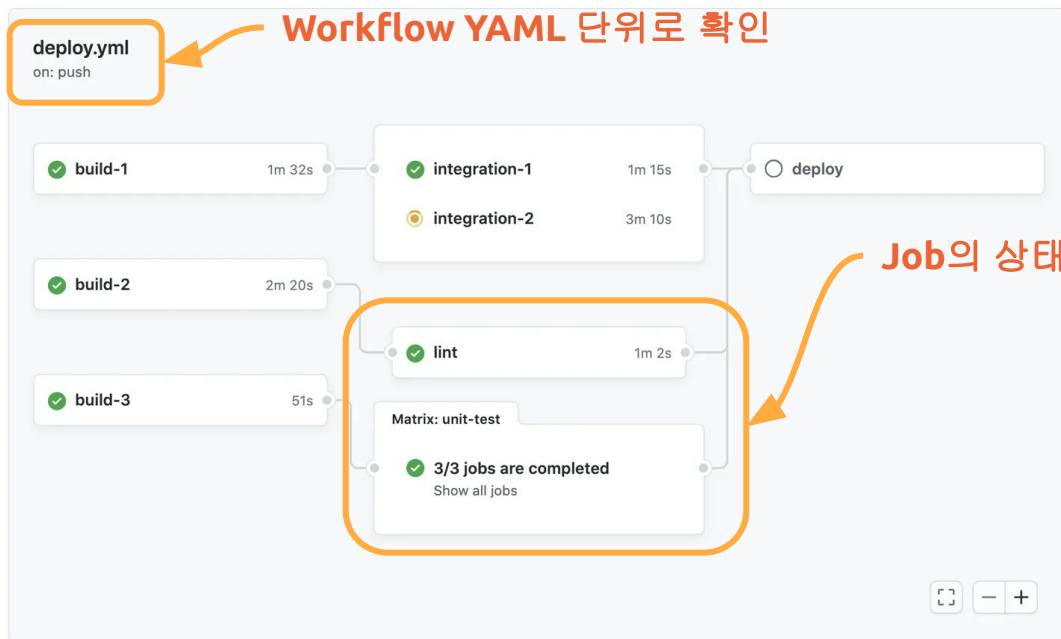
Runner Hosting에 대하여

보안 가이드라인

# Manual: Monitoring



Workflow의 Job들은 Web UI로 확인 가능



# Manual: Troubleshooting



문제 발생시 원인파악은 Workflow에서 생성한 Log를 사용

```
> ✓ Run ConorMacBride/install-package@v1 0s
> ✓ Show Folder/Files 0s
v ✗ Run stanleynguyen/thanos@snap 2s
  1 ▶ Run stanleynguyen/thanos@snap
  4 HttpError: Resource not accessible by integration
  5   at /home/luncliff/actions-runner/_work/_actions/stanleynguyen/thanos@snap/node/index.js:66:1
  6   at processTicksAndRejections (node:internal/process/task_queues:95:5)
  7   at run (/home/luncliff/actions-runner/_work/_actions/stanleynguyen/thanos@snap/node/index.js:66:1)
  Show Folder/Files
> ✓ Post Run actions/checkout@v3
> ✓ Complete job
```

Re-run single job ✕

check 후 실행

A new attempt of this workflow will be started, including balance and dependents:

✗ balance

Enable debug logging

Cancel Re-run jobs

# Manual: GitHub-hosted Runner



Job을 실행할 때 Virtual Machine을 생성하여 실행. 종료 후 VM 소멸

OS와 H/W 자원 제한

- Ubuntu 20.04, 22.04
- x64, 2 core, 7GB RAM, 14GB SSD

GitHub Plan에 따라서 사양/Job 실행시간 허용치가 달라짐

- 저장공간<sup>storage</sup>과 분<sup>minute</sup>단위로 계산

etc/hosts 설정되어있음(IP 주소 제한)

[참고: About GitHub-hosted runners](#)

[참고: About billing for GitHub Actions](#)

# Manual: Self-hosted Runner



GitHub, GitHub Enterprise에 Physical Machine, VM, Container등을 연결

자유로운 H/W, S/W 결정

- **Ubuntu 16.04+**
- x64, ARM64, ARM32 가능

Job 실행 후 clean 하지 않음

보안관리가 필요함





# Demo: Attaching Runner to GitHub Server

# Demo: Attaching Runner To GitHub Server



C++ Korea

Organization account [Switch to another](#)

## GitHub Organization의 Settings

General

Code, planning, and automation

Repository

Codespaces

Copilot

Actions

General

Runners

Runner groups

Caches

## 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.](#)

Search runners

New runner

Runners

Status

Standard GitHub-hosted runners

Ready-to-use runners managed by GitHub. [Learn more.](#)

0 active jobs

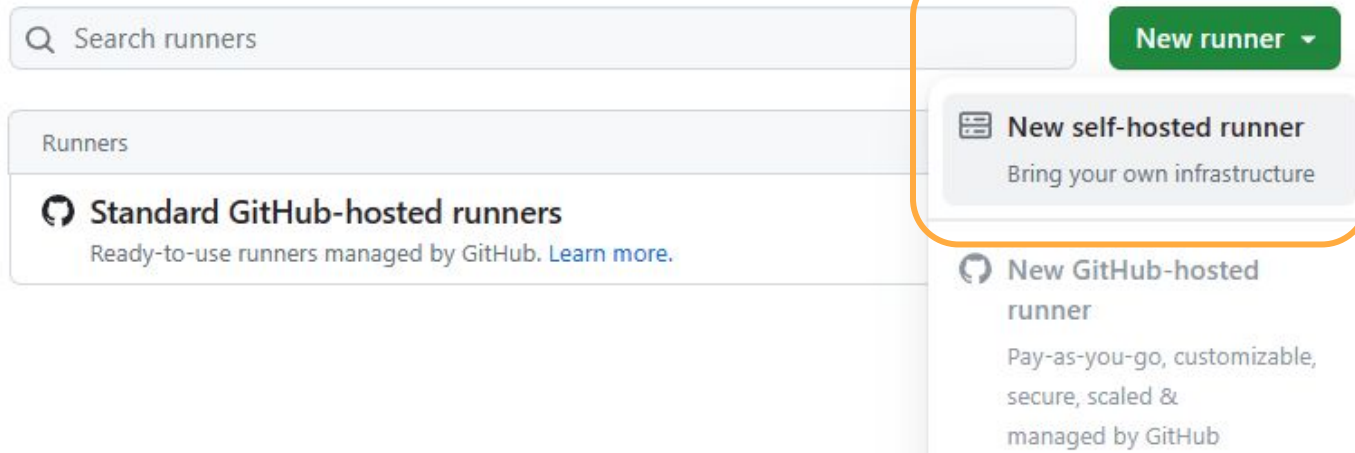
# Demo: Attaching Runner To GitHub Server



## 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.](#)

A screenshot of the GitHub Runners page. At the top, there is a search bar with the text "Search runners" and a magnifying glass icon. To the right of the search bar is a green button labeled "New runner" with a dropdown arrow. Below the search bar, there is a list of runner options. The first option is "New self-hosted runner" with a server rack icon and the subtitle "Bring your own infrastructure". This option is highlighted with an orange rounded rectangle and an orange arrow points from it to the text "GoTo: Runner 연결 방법". The second option is "New GitHub-hosted runner" with a GitHub logo icon and the subtitle "Pay-as-you-go, customizable, secure, scaled & managed by GitHub".

Search runners

New runner ▾

Runners

-  **New self-hosted runner**  
Bring your own infrastructure
-  **New GitHub-hosted runner**  
Pay-as-you-go, customizable, secure, scaled & managed by GitHub

**GoTo: Runner 연결 방법**



## Runners / Create self-hosted runner

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

```
# Download the latest runner package
$ curl -o actions-runner-linux-x64-2.308.0.tar.gz -L
https://github.com/actions/runner/releases/download/v2.308.0/actions-runner-linux-x64-
2.308.0.tar.gz
```

운영체제, 아키텍처 선택

GitHub Enterprise의 경우  
Version에 따라서 다를 수 있음



## Download

```
# Create a folder
```

```
$ mkdir actions-runner && cd actions-runner
```

```
# Download the latest runner package
```

```
$ curl -o actions-runner-linux-x64-2.308.0.tar.gz -L  
https://github.com/actions/runner/releases/download/v2.308.0/actions-runner-linux-x64-2.308.0.tar.gz
```

```
# Extract the installer
```

```
$ tar xzf ./actions-runner-linux-x64-2.308.0.tar.gz
```

## GitHub Server + Organization

## Configure

```
# Create the runner and start the configuration experience
```

```
$ ./config.sh --url https://github.com/CppKorea --token ADLSYE6LZ3A76N5RLXLGKN3E6W5AW
```

주기적으로 만료되는 Token



```
luncliff@Capital:~/actions-runner$ ./config.sh --url https://github.com/CppKorea --token ADLSYE5JXHUIDLA2C3CSQCDE6W5XI
```

```
-----  
GitHub Actions  
-----  
Self-hosted runner registration  
-----
```

```
# Authentication
```

만료된 Token을 사용하면 실패

```
✓ Connected to GitHub
```

```
# Runner Registration
```

```
Enter the name of the runner group to add this runner to: [press Enter for Default]
```

```
Enter the name of runner: [press Enter for Capital] luncliff-ubuntu-wsl
```

Runner의 name, label 지정

```
This runner will have the following labels: 'self-hosted', 'Linux', 'x86'
```

```
Enter any additional labels (ex. label-1,label-2): [press Enter to skip] WSL,Ubuntu
```

```
✓ Runner successfully added
```

```
✓ Runner connection is good
```

```
# Runner settings
```

```
Enter name of work folder: [press Enter for _work]
```

Runner가 Job 실행때 사용할 folder

```
✓ Settings Saved.
```



## # Runner Registration

Enter the name of the runner group to add this runner to: [press Enter for Default]



Enter the name of runner: [press Enter for Capital] luncliff-ubuntu-wsl

This runner will have the following labels: 'self-hosted', 'Linux', 'X64'

Enter any additional labels (ex. label-1,label-2): [press Enter to skip] WSL,Ubuntu

✓ Runner successfully added

✓ Runner connection is good

Runners	Status
 <b>Standard GitHub-hosted runners</b> Ready-to-use runners managed by GitHub. <a href="#">Learn more.</a>	● 0 active jobs
 <b>luncliff-ubuntu-wsl</b> <span>self-hosted</span> <span>Linux</span> <span>X64</span> <span>Ubuntu</span> <span>WSL</span> Runner group: Default	● Offline ...

아직 Self-hosted Runner 등록만 끝난 상태



```
luncliff@Capital:~/actions-runner$ ./run.sh
```

```
✓ Connected to GitHub
```

```
Current runner version: '2.308.0'
```

```
2023-09-04 10:25:01Z: Listening for Jobs
```

luncliff-ubuntu-wsl self-hosted Linux X64 Ubuntu WSL

Runner group: Default

● Idle ...

Job 실행 가능



## Runner groups / Default



### Group name

### Repository access

1 selected repository  **Allow public repositories**

Runners can be used by public repositories. Allowing self-hosted runners on public repositories and allowing workflows on public forks introduces a significant security risk. [Learn more about self-hosted runners.](#)

**Workflow 실행을 위해 접근 가능한 저장소를 설정**

### Workflow access

Control how these runners are used by restricting them to specific workflows. [Learn more about managing runner groups.](#)

Runners

Status

luncliff-ubuntu-wsl

self-hosted

Linux

X64

Ubuntu

WSL

● Idle

...



```
luncliff@Capital:~/actions-runner$ ./config.sh remove --token ADLSYE6WIDTCOMW6443TURDE4IO4E
```

```
# Runner removal
```

```
✓ Runner removed successfully  
✓ Removed .credentials  
✓ Removed .runner
```

Runner 제거 후 확인

🔍 Search runners

New runner ▾

There are no runners configured

[Learn more about using runners](#) to run actions on your own servers.



## 2. Self-Hosted Runner 탐구

# Self-Hosted Runner 필요성



## 프로젝트 특성에 맞는 사양Specification

- 오픈소스 소프트웨어<sup>OSS</sup> 생태계에 내가 원하는 사양이 없을수도 있다..
  - ex) 고사양 GPU, 대용량 Storage와 연결된 VM

## 조직Organization에서 사용하는 환경과의 유사성

- 개발/테스트/배포 환경에 필요한 인프라Infrastructure 자원
  - 개발장비 수준의 사양? 테스트용 사양?
- 단계마다 다른 사양 수준이 요구될수도 있음
  - 실험/연구 단계에서는 보다 고사양이 필요할 수 있음 (Disk Space? RAM? GPU?)
  - 개발이 활발한 상황이라면 실행해야 하는 작업량이 더 높음

# Self-Hosted Runner 사용범위



개발 단계<sup>phase</sup>마다 실행하는 작업 흐름<sup>Workflow</sup>이 다르다!

- Setup → Build → Test → Merge → Deploy → Operation

## Continuous XXX???

- 작업<sup>Job</sup>들을 분할해서
- Phase 별로 분배하고
- 실행을 자동화한다

# Phase별 관심사항?



- Setup → Build → Test → Merge → Deploy → Operation

## 관심사항

- 툴체인 설치(Language Runtime, Tools, etc)
- 성능/사양 제약(CPU, RAM, GPU, etc)
- 저장공간 제약(Disk Storage)
- 작업의 실행시간 최적화

# Phase별 관심사항?



- Setup → Build → Test → Merge → Deploy → Operation

## 관심사항

- 테스트를 제한하는 입력/출력 제약
- 반복적인 테스트 자원 준비
- 통합 전/후 산출물 비교(벤치마크)

# 관리자: 다수의 개발자들을 위한 Runner 환경?



개발환경은 이미 파편화 되어있다!

- 개발자들의 환경은 개인의 선호, 직무에 최적화 되어있음
- 개인환경과 범용환경에는 괴리가 발생할 수 밖에 없다

개발조직 수준의 합의된 개발/통합환경?

- 개발자들의 환경 - **오픈소스 소프트웨어<sup>OSS</sup>들의 범용환경** 사이의 어딘가
- 타협점을 결정하기 위한 고민/기준이 있어야 한다  
(주의: 고민만으로 충분하고, 기준을 결정해야하는 부분이 아닐수도 있음)



# 주의! 오픈소스 생태계는 계속 변화한다



Ubuntu images update on August 14, 2023  Inbox x



**AppVeyor** <team@appveyor.com> [Unsubscribe](#)  
to me ▾

## What's new in Ubuntu images

- Go 1.19.12, 1.20.7
- Python 3.7.16, 3.8.17, 3.9.17, 3.10.12, 3.11.4
- Flutter 3.10.6
- Git 2.41.0
- Git LFS 3.4.0
- Ruby 3.1.4, 3.2.2
- CMake 3.27.1
- NVM 0.39.4
- Node 20.5.0, 19.9.0, 18.17.0, 16.20.1
- MongoDB 6.0.8 (Bionic and Focal)

CI/CD 서비스들을 사용하고, 따라하는 것  
만으로도 **신선한** 개발환경을 유지할 수 있다

== 우리(조직/제품)는 금방 부패<sup>Rot</sup>한다

# 오픈소스들이 사용하는 Ubuntu 환경 살펴보기



## GitHub Actions

- <https://github.com/actions/runner-images Ubuntu2204-Readme.md>

## AppVeyor

- <https://www.appveyor.com/docs/linux-images-software/>
- <https://github.com/appveyor/build-images>

## CircleCI

- <https://circleci.com/developer/images/image/cimg/base>



# Demo: Workflows on Ubuntu Runner

# Workflow 예시: 실행 Log 읽어보기



GitHub 저장소: <https://github.com/CppKorea/UbuConKR-ubuntu-self-hosted-runner>

생각해볼 부분

- 툴체인 설치가 쉬운가?
- 환경이 사라지지 않기 때문에 발생하는 GitHub-hosted Runner와의 차이점



### 3. 다수의 Runner를 사용하기 위한 고민

# Self-hosted Runner 관리



안정적인 운영을 위해서는 직무정립이 필요(인프라 관리)

- 직책에 맞는 권한 관리
  - Role-Based Access Control
- 환경 복잡도
  - Runner Group의 종류와 전체 Runner 수
- 자동화에서 발생하는 부하
  - 사용인원 수
  - 작업 발생량
- 안정성 수준
  - 장애, 서비스 변화 모니터링
  - 사고 상황별 대응방법

# Self-hosted Runner 관리



개발조직이 함께 사용하는 환경에 대해 고민

유지보수

- 환경의 업데이트 범위/주기
- 장애 예상/대비/조치
- 지식/권한 관리
- 효율화 연구

확장성

- 복제가능성
- 이식성

# Self-hosted Runner 구축 (with Ubuntu OS)



Ubuntu OS를 사용하는 조직에는 여러 이점들이 존재

- 낮은 학습곡선
  - 많은 CI 서비스에서 First-class Environment로 지원
  - Dockerfile처럼 Copy-Paste 하기 쉬운 예시들
  - <https://ubuntu.com/server/docs>
- Package 설치가 쉬움
  - APT, DPKG 외에도 Snap, Homebrew, SDKMan 등이 Ubuntu를 지원함
- GitHub Action 들의 기준 환경
  - GitHub Action들이 test 환경으로 (GitHub-hosted) Ubuntu를 사용



# 다수의 Runner를 사용해보니...



모든 작업<sup>Workflow</sup>들이 동질의 환경을 필요로 하지는 않는다

- 각 작업<sup>Job</sup>가 필요로하는 자원 규모<sup>Resource Class</sup>에 맞춰 분배할 필요가 있음
- Runner들의 파편화는 관리 효율과 편의를 저해한다

# 다수의 Runner를 사용할때 이로운 부분?



다수의 Runner를 사용하는 것이 더 견고한 Pipeline을 만든다

- 여러 Disk Storage로 RAID를 구성하는 것과 같은 원리
  - 이 경우는 순차적인 OS/보안 업데이트가 가능
- 적은 수의 Runner 장애가 Pipeline을 붕괴시키는 상황을 방지
  - 일부가 Disconnected 될 확률은 높아도 전체가 Disconnected될 확률은 더 낮다
  - 운영서버보다 안정성 수준이 낮아도 큰 문제가 없음

# Runner Group의 전략과 Workflow 변화



개발자들이 **Workflow**를 설계할 때 영향을 준다.

## Scale-Up (더 높은 사양을 지원)

- **Workflow**는 더 높은 사용률<sup>Utilization</sup>을 지향
- 단일 작업단위<sup>Job</sup>에서 더 많은 **CPU, RAM**을 사용

## Scale-Out (더 많은 Runner를 사용)

- **Workflow**는 더 높은 동시성<sup>Concurrency</sup>을 지향
- 더 작은 작업단위<sup>Job</sup>로 분할

# 보안관리: GitHub Actions



- **GitHub Token**
  - 최소한의 권한만, 주기적으로 갱신 **모든 Token이 Expire 하도록 관리**
- **저장소 간 cross-repository 접근**
  - GitHub Token이 여러 저장소에 대한 권한을 가진 경우
  - 개발자의 개인 Access Token 혹은 SSH key를 사용하는 경우
- **GitHub Action**
  - Step에 Script보다, Action 들을 조합하는 것을 권장
  - 사용할때는 commit SHA를 최대길이 full-length로 참조하거나, Tag 사용
  - Token을 기대한 방법으로 사용하고 있는지 검수/감독 **Workflow 외부도 계속 관찰하고, 교체가 가능하도록 대안을 준비**
- **Dependabot**
  - 조치가 필요할 수 있는 경우, 사용자들에게 알림
- **GitHub Action을 사용해 PR 생성, 승인은 금지** Prevent
  - 적절한 감독 없이 변경사항이 발생하지 않도록

# 보안관리: GitHub-hosted runner



- Software Bill Of Materials(SBOM) 관리
  - <https://github.com/actions/runner-images/releases> 에서 설치된 S/W 확인 가능
  - 공급망에 개입하는 S/W들의 확인 Identification에 사용하는 것을 권장
- 허용되지 않은 Host에 접근 거부
  - Workflow를 위해 생성한 Isolated VM 내에서 작업을 시작-마무리

## Self-hosted Runner의 보안과 다른 점?

- Self-hosted는 Persistent한 환경을 사용하기 때문에, Isolated(Clean) 환경을 보장하지 않는다

다른 저장소들의 파일들이  
같은 파일시스템의 “예상 가능한 위치”에 존재

# 보안: Self-hosted runner



- Public 저장소에는 사용하지 말 것
  - Internal, Private 저장소에서도 주의

하나의 **Action**이 잘못되면  
**Workflow** 전체로 위험이 확산

- 필요하다면 저장소마다 **Runner**를 따로 설정
  - Machine에 **SSH Key, Token**이 저장되어있지는 않은가?
  - 중요한 서비스와 통신한다면 방화벽 설정

저장소가 많아질수록  
잠재적 위험 영역이 증가한다

- **Runner**들의 관리전략을 갖출 것

- Centralized? - Enterprise, Organization에서 관리. 직무 체계화
- Decentralized? - Organization, Repo Owner들이 관리

개인장비를 사용한다면 더욱 필수

규모에 맞는 관리형태, 책임을 분석하고  
고민하는 과정이 필요



# Demo: Multi-Runner Workflow



# 되짚어보기 + 결론



# 되짚어보기



GitHub Actions 매뉴얼 읽어보기

Self-hosted Runner 연결/연결 후 조치/제거

Self-hosted Runner 에 구성할 공용 개발환경에 대한 고민

오픈소스 생태계를 지탱하는 Ubuntu 환경들

Ubuntu Runner들에서 Workflow 실행, 디버깅



Self-hosted runner 사용법은 쉽지만 관리는 어렵다(고민거리가 많다)

Ubuntu runner들은 오픈소스 환경을 참조하고 모방하기 쉽기 때문에 복제/학습/실험이 쉽다는 장점



질문하는 시간!



나중에라도 발표 내용에 대해서 궁금한 부분이 생긴다면...

프로필: [github.com/luncliff](https://github.com/luncliff) [linkedin.com/in/dong-ha-park-171481229/](https://linkedin.com/in/dong-ha-park-171481229/)

문의 메일: [luncliff@gmail.com](mailto:luncliff@gmail.com) [luncliff@cppkorea.org](mailto:luncliff@cppkorea.org)

C++ Korea Discord: [discord.gg/87SNegGZue](https://discord.gg/87SNegGZue)

C++ Korea Facebook Group: [fb.com/groups/cppkorea](https://fb.com/groups/cppkorea)



감사합니다!