

Introducing the CI WG

Presentation to the USD WG
Jean-François Panisset
April 2025

/* ACADEMY
SOFTWARE
FOUNDATION

Continuous Integration WG Mission Statement

The ASWF Continuous Integration (CI) Working Group is tasked to investigate and deliver tools, procedures and services used by ASWF projects in their development process, including:

- Revision control
- Build toolchains
- CI build environments
- Testing
- Packaging and distribution
- Provide a forum for the exchange of ideas between individuals interested in all aspects of the tooling and processes involved in software development, building and distribution
- Nothing we do is prescriptive, projects are free to adopt or not, and are encouraged to share infrastructure with other ASWF projects.

We are poorly named, we should be called something like the "Tooling and Integration" WG. We may end up becoming an actual ASWF project (since we have some code deliverables).

Our Meetings and Slack

- One meeting a month, off cadence from TAC (next is June 19th at 13:00 Pacific Time)
 - Overlaps with USD WG Meeting...
- A good place to interact with LF Release Engineering
- Meeting minutes in wg-ci repo:
<https://github.com/AcademySoftwareFoundation/wg-ci/tree/main/meetings>
- Our Slack Channel: #wg-ci
 - Currently 374 members
 - Reasonable traffic volume, enough to help fill meeting agenda, a good place to share an interest in tooling

Where to Get Help with Infrastructure

- #wg-ci for general discussion
- Linux Foundation Release Engineering Helpdesk:
<https://jira.linuxfoundation.org/plugins/servlet/desk>
 - GitHub Permissions
 - Secrets Management
 - External Integrations
- Our monthly call

Status Update

Parts of our service are experiencing issues. [View StatusPage](#)

Project Support Services > Request PyPI organization on behalf of AcademySoftwareFoundation

 Request PyPI organization on behalf of AcademySoftwareFoundation (IT-24425)

Comment on this request...

ACTIVITY

Your request status changed to: **Waiting for Customer** 25/Apr/23 7:56 AM **LATEST**



Andrew Grimberg 25/Apr/23 7:56 AM

I applied for the "alias" of aswf with the long description of "Academy Software Foundation" which is what the registration form seemed to be indicating to me was the best way given the examples in the form.

Your request status changed to: **Waiting for Support** 24/Apr/23 2:12 PM



Jean-Francois Panisset 24/Apr/23 2:12 PM

Did you apply for "AcademySoftwareFoundation" or "aswf"?

CI WG Wiki

<https://wiki.aswf.io/display/CIWG>



Academy Software Foundation Wiki

Academy Software Foundation wiki

Spaces ▾

Apps ▾

Templates

+ Create

🔍 Search



Continuous Integration (...)

▾ Continuous Integration (CI) Workflows / GitHub



All content



Space settings



Content



Search by title



GitHub



Tooling



Blogs

GitHub



Owned by [Jean-Francois Panisset](#) ...

Last updated: Dec 22, 2024 • 3 min read

- [GitHub Actions Runners](#)
- [Issue with GitHub Actions and CentOS 7 based aswf-docker containers](#)

GitHub Actions Runners

By default ASWF projects under the AcademySoftwareFoundation GitHub organization have access to the [Standard GitHub-hosted runners for public repositories](#) for Linux, Windows and macOS. Our GitHub organization qualifies as an Enterprise organization which increases the limits for job concurrency and usable monthly minutes.

Some ASWF projects have their own top-level GitHub organization: these are included under our Enterprise account and should have the same access as projects / repos under the AcademySoftwareFoundation GitHub organization.

ASWF also has a limited monthly budget for [Larger Runners](#) which add runners to support larger numbers of CPU, more RAM, different CPU architectures and GPUs. These Larger Runners are managed by Linux Foundation Release Engineering team and can be selected by the `runs-on` directive in the workflow file. They can only be used by CI jobs running in the context of the project repository, they cannot be accessed from a fork.

Since the monthly budget is limited, projects should make judicious use of these Larger Runners, possibly for lower frequency jobs to validate different CPU architectures, or to run a GPU-accelerated test suite rather than a full compilation. Projects wishing to use these Larger Runners may want to bring up their requirements for discussion in the [#wg-ci](#) Slack channel.

AWS Infrastructure

Digital Production Example Library hosts and serves out large production relevant sample assets on Amazon AWS infrastructure.

Linux Foundation Release Engineering manages AWS infrastructure for ASWF projects that require it.



GitHub Actions Free Runners

- Free runners adequate for many jobs
 - Recently upgraded: 4 core, 16GB RAM (was 2 cores, 7GB)
 - But still only 14GB disk, can be challenging
- Apple Silicon runners now available (Jan 2024)
- Linux on ARM, but no Windows on ARM yet

Virtual Machine	Processor (CPU)	Memory (RAM)	Storage (SSD)	Architecture	Workflow label
Linux	4	16 GB	14 GB	x64	ubuntu-latest , ubuntu-24.04 , ubuntu-22.04 , ubuntu-20.04
Windows	4	16 GB	14 GB	x64	windows-latest , windows-2025 [Public preview], windows-2022 , windows-2019
Linux [Public preview]	4	16 GB	14 GB	arm64	ubuntu-24.04-arm , ubuntu-22.04-arm
macOS	4	14 GB	14 GB	Intel	macos-13
macOS	3 (M1)	7 GB	14 GB	arm64	macos-latest , macos-14 , macos-15 [Public preview]

ASWF Enterprise GitHub Organization

- Higher limits on free GHA minutes, concurrent jobs
- Access to larger, for pay runners:
 - \$1,500/month pre-authorized
 - Used by OpenVDB, aswf-docker for builds that wouldn't complete otherwise
 - Informally managed
 - Available to all projects, but to be used judiciously
- Enterprise org benefits can be extended to projects not under ASWF GitHub org (OTIO uses its own org)

Access to Larger Runners

- More cores, memory, disk space
- Windows on ARM
- T4 GPUs (better coming soon)
- Windows 2x cost of Linux
- Pre-built custom images
 - Bake in large dependencies
(alternative to container builds)

The screenshot shows the GitHub Actions interface. The top navigation bar includes links for Code, Issues (24), Pull requests (5), Discussions, Actions (selected), Projects, Wiki, Security (10), Insights, and Settings. The left sidebar has a 'Runners' section highlighted under 'Management', with sub-items for Caches, Attestations, Usage metrics, and Performance metrics. The main content area is titled 'Runners' and shows '24 available runners'. It lists various runner configurations, including Standard GitHub-hosted runners and Self-hosted runners. The runners are categorized by OS and hardware specifications.

Runner Name	OS	Cores	RAM	Disk
ubuntu-20.04-gpu-t4-4c-16g-176h	Ubuntu	4	16 GB	176 GB
windows-ds-2019-gpu-t4-4c-16g-176h	Windows	4	16 GB	176 GB
ubuntu-latest-m	Ubuntu	4	16 GB	150 GB
windows-latest-l	Windows	8	32 GB	300 GB
ubuntu-24.04-arm64-2C-8G-75H	Ubuntu	2	8 GB	75 GB
ubuntu-24.04-arm64-4C-16G-150H	Ubuntu	4	16 GB	150 GB
ubuntu-24.04-arm64-8C-32G-300H	Ubuntu	8	32 GB	300 GB
ubuntu-24.04-arm64-16C-64G-600H	Ubuntu	16	64 GB	600 GB
windows-11-arm64-4C-16G-150H	Windows	4	16 GB	150 GB

Larger Runners: OpenVDB

← → ↻ 🔍 github.com/AcademySoftwareFoundation/opencvdb/blob/master/.github/workflows/weekly.yml

📁 master ↕ [opencvdb](#) / [.github](#) / [workflows](#) / [weekly.yml](#)

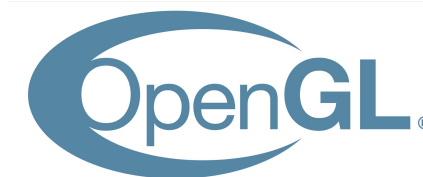
Code Blame 486 lines (467 loc) · 21.4 KB · ⓘ

```
137 #####
138 ##### Core Library Extras #####
139 #####
140
141 # Extra configuration tests for the OpenVDB Core library. These test a
142 # variety of options with newer compilers.
143 linux-extra:
144   if: |
145     github.event_name != 'workflow_dispatch' ||
146     github.event.inputs.type == 'all' ||
147     github.event.inputs.type == 'extra'
148   runs-on: ${ (github.repository_owner == 'AcademySoftwareFoundation' && 'ubuntu-20.04-8c-32g-300h') || 'ubuntu-latest' }}
149   name: linux-extra:${ matrix.config.name }
150   container:
151     # @note we specifically use clang15.0 (not clang15) here as the newest
152     # versions of the clang15.X containers have some issues with the GLFW
153     # installation
154     image: aswf/ci-openvdb:2023-clang15.0
155   env:
```



Native GPU GitHub Actions Runners

- Windows / Linux T4 GPU runners
 - 4 core / 16 GB RAM / 176GB disk
 - 16GB VRAM
 - runs-on: ubuntu-20.04-gpu-t4-4c-16g-176h
 - runs-on: windows-ds-2019-gpu-t4-4c-16g-176h
 - for pay, so typically to run GPU test suite
- Runner provides NVIDIA driver 535.54.03
- aswf-docker container provides OpenGL dev and runtime environment
 - but no X server: see OpenColorIO for how to use EGL to get OpenGL context
- Currently netter GPUs via AWS CodeBuild
 - NanoVDB needs INT8 / TF32 data format support
 - GitHub Hosted Runners to gain newer GPUs soon



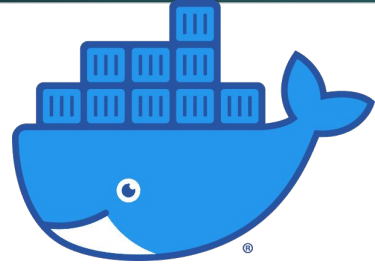
Packaging and Distribution

- Python and PyPI
 - Wheels packaging expertise in several projects, ask in #wg-ci
 - aswf organization now registered with PyPI
- Paid Docker Hub account for Docker containers
 - No throttling on downloads of aswf-docker containers
 - Available to all projects (GitHub org level secrets)
- JFrog Artifactory
 - <https://linuxfoundation.jfrog.io/artifactory/aswf-conan/>
 - Conan packages from aswf-docker, supports many other formats
 - Working towards making Conan packages usable outside of aswf-docker for project dependencies



The aswf-docker Project

<https://github.com/AcademySoftwareFoundation/aswf-docker>



- Yearly implementation of the VFX Reference Platform
- Hierarchy of layered containers:
 - aswf/ci-base aswf/ci-baseqt have tools and most prerequisites
 - aswf/ci-usd has all dependencies to build USD
 - aswf/ci-vfxall has everything pre-built
- Used by most ASWF projects to build in a controlled environment with all required dependencies
- VFX2025 based on Rocky Linux 8, leverages RedHat Developer Toolset 11 for newer dev tools (gcc 11.2.1)
- Clang also included (18 and 19 in 2025)

Using aswf-docker containers

```
name: VFX Platform 2025 CI
on:
  push:
    branches: [ main ]
jobs:
  container-test-job:
    runs-on: ubuntu-latest
    container:
      image: aswf/ci-usd:2025
```

Use a matrix strategy to build for multiple VFX Platform years

The aswf-docker Project

- Includes CUDA, OpenGL
 - Can be used to run GPU accelerated test suites
 - Can be used to run entire applications
- Can be used to build "difficult to build" open source projects on CentOS 7 by providing updated dependencies and tools
- Pushed to paid, unthrottled Docker Hub account:
 - <https://hub.docker.com/u/aswf>
 - but 4+ GB vfx-all container can still take a while to pull...
 - pre-built images for paid runners could be an option

The aswf-docker Project



Academy Software Foundation

Community Organization Academy Software Foundation <https://www.aswf.io>

Repositories

Search by repository na

Displaying 1 to 25 of 42 repositories



aswf/ci-openrv

↓46 · ☆0

By [aswf](#) · Updated 6 months ago

OpenRV CI Docker Image Contains: Python, Boost, OpenEXR and other OpenRV upstream dependencies W...



aswf/ci-vfxall

↓10K+ · ☆8

By [aswf](#) · Updated 6 months ago

VFX All CI Docker Image Contains: all supported VFX packages



aswf/ci-osl

↓100K+ · ☆2

By [aswf](#) · Updated 6 months ago

OpenShadingLanguage CI Docker Image Contains: Python, Boost, Qt, OpenEXR, OpenImageIO, Partio an...



aswf/ci-openvdb

↓100K+ · ☆1

By [aswf](#) · Updated 6 months ago

OpenVDB CI Docker Image Contains: Python, Boost, TBB and other OpenVDB upstream dependencies War...



aswf/ci-usd

↓2.3K · ☆2

By [aswf](#) · Updated 6 months ago

USD CI Docker Image Contains: Python, Qt, PySide and other USD upstream dependencies Warning: th...

aswf-docker: Work in Progress

- Convert more dependencies and packages to Conan
 - Almost done, only a few projects remain
 - Create Conan packages from build results
 - Push to JFrog Artifactory repository
 - More granularity for clients consuming packages
 - Better applicable to Windows and macOS
 - `aswf-docker/packages/conan/recipes/`
- VFX Platform 2025 support
 - Qt6 was a big step for 2024
 - Minimize differences with Conan Center Index recipes
 - Starting to look a bit like a "VFX Linux" distribution (almost 100 packages)

aswf-docker: What's Next?

- Build containers for remaining ASWF projects: OpenFX, xStudio
- Complete transition to Conan: OpenVDB, OSL, USD
- Support use of Conan packages without having to build inside container (lighter weight dependencies)
- Windows support
 - Leverage CMake / Conan integration to provide simpler Windows builds?
 - Public Conan recipes already multi-platform
 - No "containerized macOS" unfortunately, but could use Conan packages on native macOS

aswf-docker and USD

- ci-usd: a container with all the pre-built dependencies in /usr/local to build USD
- ci-vfxall: a container with everything pre-built, including USD
- USD version not prescribed by VFX Platform, so try to pick the most recent for the new VFX Platform year
- Tricky interaction with MaterialX versions, aswf-docker wants recent MaterialX
 - End up having to pull additional patches
 - Using "25.02a.eae7e67" for 2025 containers to work with MaterialX 1.39.3

Open Discussion