/* ACADEMY SOFTWARE FOUNDATION

Introducing the CI WG

Presentation to the USD WG
Jean-François Panisset
April 2025

/* ACADEMY SOFTWARE FOUNDATION
#ASWF

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).

/* ACADEMY SOFTWARE FOUNDATION

#ASWF

Our Meetings and Slack

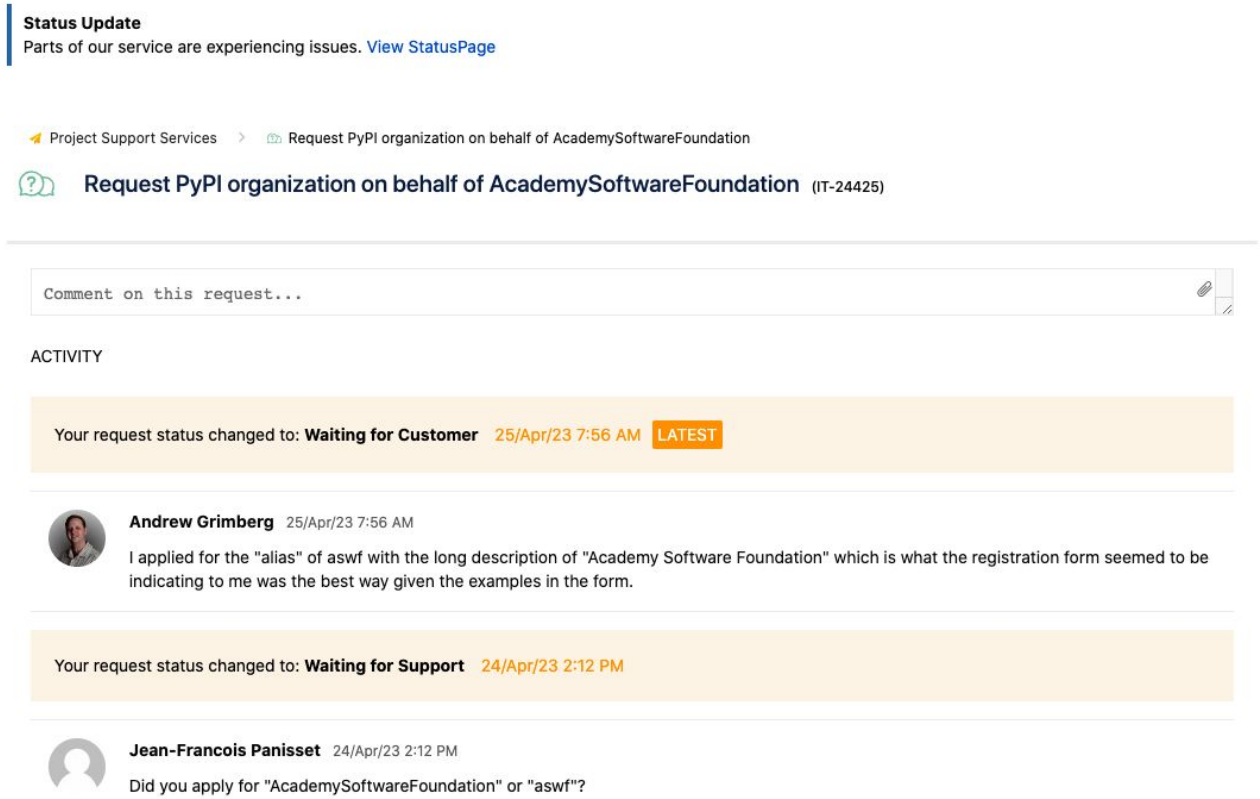
- One meeting a month, off cadence from TAC (next is June 19th at 13:00 Pacific Time)
  - Overlaps with USD WG Meeting...
- A good place to interact with LF Release Engineering
- Meeting minutes in wg-ci repo: https://github.com/AcademySoftwareFoundation/wg-ci/tree/main/meetings
- Our Slack Channel: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

/* ACADEMY SOFTWARE FOUNDATION

#ASWF

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

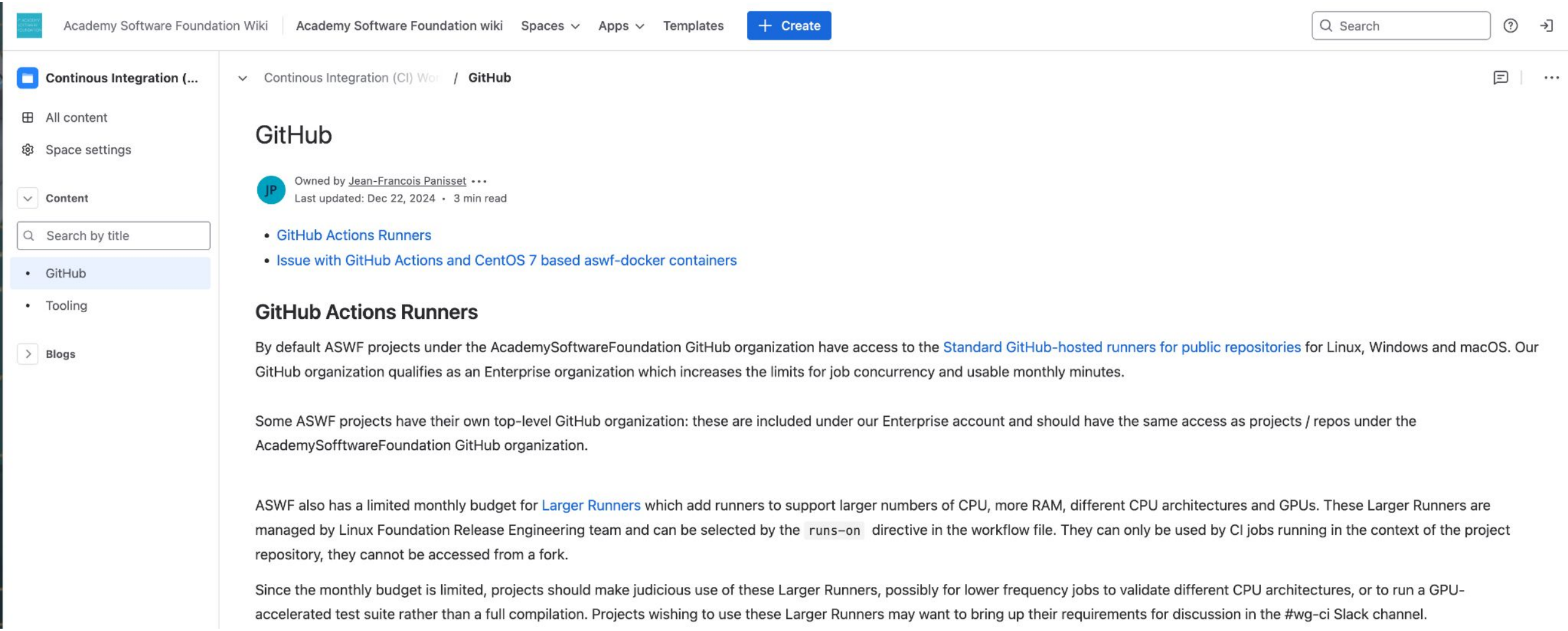


/* ACADEMY SOFTWARE FOUNDATION

#ASWF

CI WG Wiki

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



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.

/* ACADEMY SOFTWARE FOUNDATION
#ASWF

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]

/* ACADEMY SOFTWARE FOUNDATION
#ASWF

ASWF Enterprise GitHub Organization

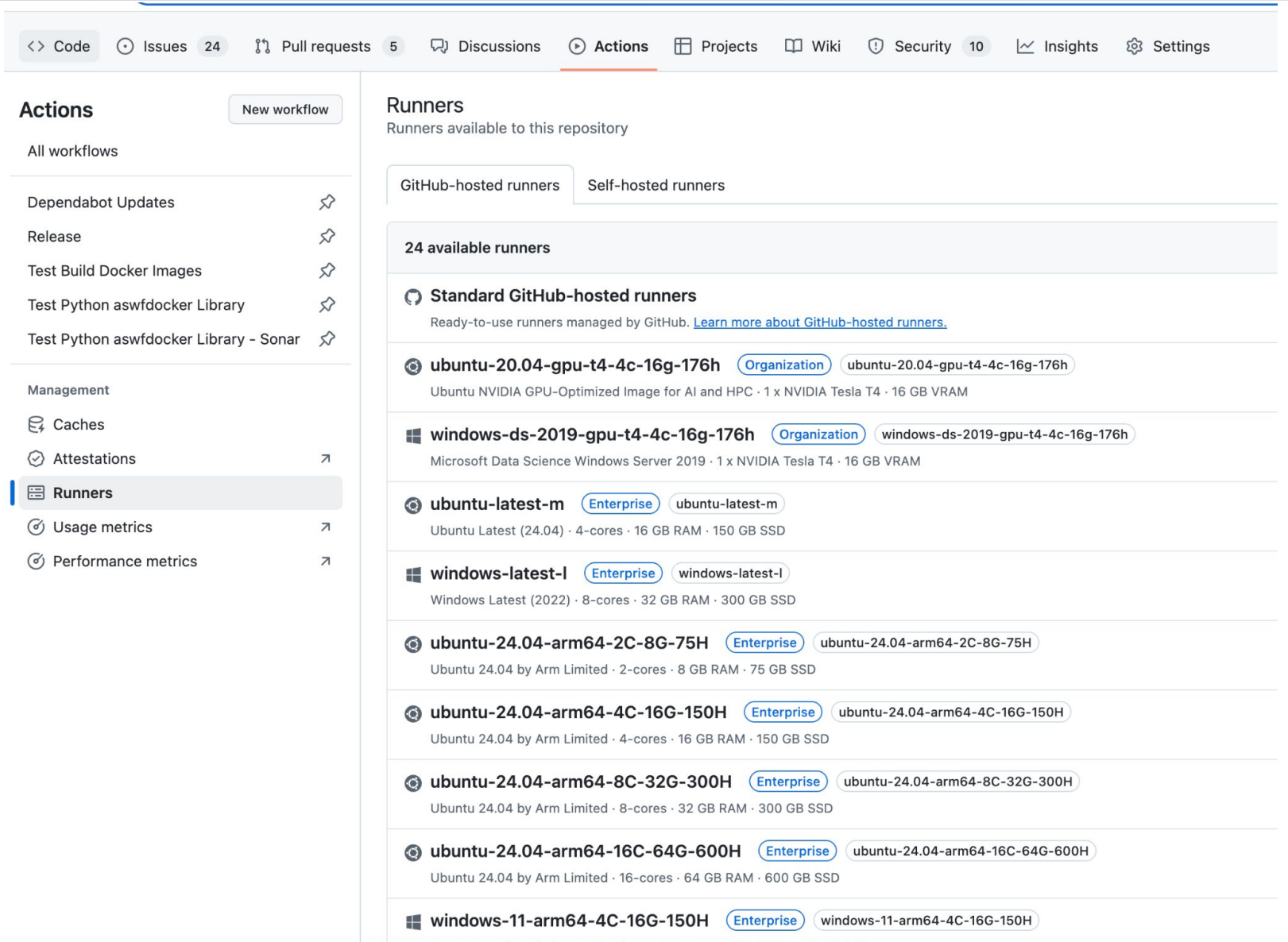
- Higher limits on free GHA minutes, concurrent jobs
- Access to larger, for pay runners:
  - $1,500/month pre-authorized
  - Used by OpenVDB, aswf-docker for builds that wouldn't complete otherwise
  - Informally managed
  - Available to all projects, but to be used judiciously
- Enterprise org benefits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

/* ACADEMY SOFTWARE FOUNDATION

#ASWF

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)



/* ACADEMY SOFTWARE FOUNDATION
#ASWF

Larger Runners: OpenVDB

github.com/AcademySoftwareFoundation/openvdb/blob/master/.github/workflows/weekly.yml

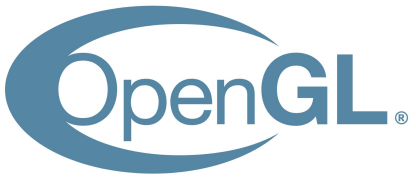
master | openvdb / .github / workflows / weekly.yml

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:
```

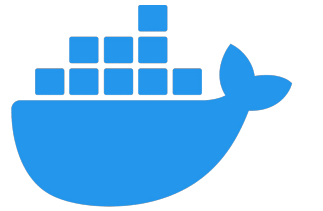
Native GPU GitHub Actions Runners

- 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

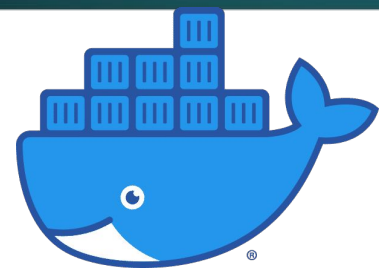
- Python and PyPI
  - Wheels packaging expertise in several projects, ask in #wg-ci
  - aswf organization now registered with PyPI
- Paid Docker Hub account for Docker containers
  - No throttling on downloads of aswf-docker containers
  - Available to all projects (GitHub org level secrets)
- JFrog 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



docker

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

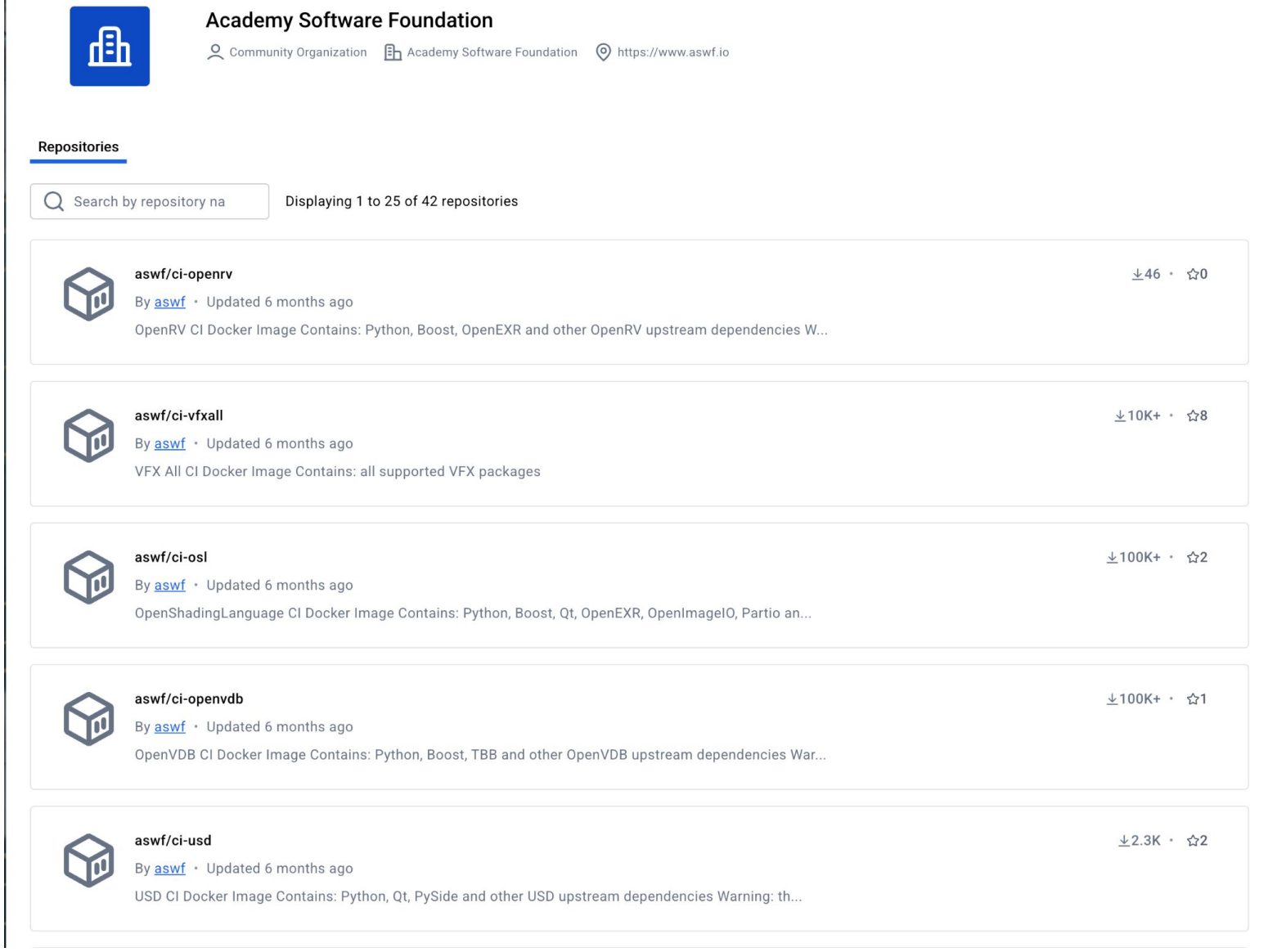
The aswf-docker Project

- 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

/* ACADEMY
SOFTWARE
FOUNDATION

#ASWF

The aswf-docker Project



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 provider 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