RCS-IT Technical Committee: Registry

Ricardo Rocha, IT-PW

Registry Discussion

Last rounds after a request to IT (Sep 22nd 2023) and ATLAS S&C workshop (Feb)

CERN offers an officially supported central registry for OCI artifacts, including container images, helm charts, machine learning models and many others. The service is backed by Project Harbor, a CNCF Graduated Project. Service and documentation are available at:

https://registry.cern.ch https://kubernetes.docs.cern.ch/docs/registry/

Core Features

The CERN registry supports all the standard functionality expected by a docker image registry but adds a significant number of relevant features.

The most relevant include:

- per project quota control and enforcement, and policy support for tag retention and immutability.
- support for multi-arch images sharing the same tag.
- support for singularity images.
- proxy (pull-through) caches allowing a local and efficient cache for upstream registries such as dockerhub, quay.io, gcr.io, ghcr.io, among others.
- automated replication with other OCI registries.
- signing and traceability of artifacts, with upcoming Software Bill of Materials (SBOMs).
- automated vulnerability scanning and CVE allowlists.

Integrations

The service offers a number of integrations with other CERN services.

- Integration with the GitLab CI is available for all GitLab projects via an official CI job. Support exists for building and deploying both container images and helm charts, with advanced features including builds for multiple architectures and accelerated images for lazy pulling.
- Unpacked CVMFS for selected projects.
- Custom integrations can be implemented via Harbor webhooks, with event notifications for artifacts, vulnerability scans, project quotas and replication.

Technical Network

An additional (experimental) instance of the CERN registry exists inside the technical network. Its main purpose is to serve workloads from the CERN Accelerator Sector. While fully isolated from the GPN instance, automated artifact replication among instances is possible for individual projects - on request.

Availability

The service is available to all CERN internal and external users, with no restrictions or limitations. Support is available via the Kubernetes Service service element.


General / Harbor
Last 7 days
1m
project
All
Harbor Health
Info
Harbor Health
100%
Projects
292
Repos
2954
Public Projects
190
Public Repos
2610
Private Projects
102
Private Repos
344
Quota Overcommit
1.64
S3 Quota Usage (rough estimate)
8.504 TiB
Quota Usage
5.46 TiB
3.64 TiB
1.82 TiB
0 B
04/20 00:00
04/21 00:00
04/22 00:00
04/23 00:00
04/24 00:00
04/25 00:00
Last *
Max
Min
ci4fpga 932 GiB 1.54 TiB 886 GiB
docker.io 1.44 TiB 1.46 TiB 1.43 TiB
acc 823 GiB 823 GiB 763 GiB
Artifact Pulled
10
7.50
5
2.50
0
04/20
04/21
04/22
04/23
04/24
04/25
04/26
min
max
avg
acc 0 0.119 0.00338
acctesting 0 0.0111 0.000220
atlas 0 0.800 0.00262
Total Projects / Repositories
293
292.5
292
291.5
291
2965
2960
2955
2950
2945
04/20
04/22
04/24
04/26
Total projects
Total repositories
Total Artifacts
43500
43000
42500
04/20
04/22
04/24
04/26
Total artifacts
Project Quota Usage
100%
80%
60%
40%
20%
0%
04/20
04/22
04/24
Core Metrics (3 panels)
JobService Metrics (8 panels)
Registry Metric (6 panels)

Project Overview

Harbor is a CNCF Graduated project

Backed by Distribution (the Docker Registry) with many added things

Project Quotas

OCI Artifact Support

Vulnerability Scanning (trivy, …), Artifact Signing (sigstore, …), SBOMs

Proxy Caches and Automated Replication

Non Blocking Garbage Collection

Tag Immutability and Retention Policies

Graduation Stage

To graduate from sandbox or incubating status, or for a new project to join as a graduated project, a project must meet the incubating stage criteria plus:

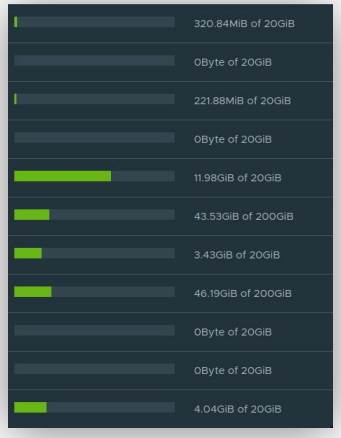
- Have committers from at least two organizations.
- Have achieved and maintained a Core Infrastructure Initiative Best Practices Badge.
- Have completed an independent and third party security audit with results published of similar scope and quality as the following example (including critical vulnerabilities addressed): https://github.com/envoyproxy/envoy#security-audit and all critical vulnerabilities need to be addressed before graduation.
- Explicitly define a project governance and committer process. The committer process should cover the full committer lifecycle including onboarding and offboarding or emeritus criteria. This preferably is laid out in a GOVERNANCE.md file and references an OWNERS.md file showing the current and emeritus committers.
- Explicitly define the criteria, process and offboarding or emeritus conditions for project maintainers; or those who may interact with the CNCF on behalf of the project. The list of maintainers should be preferably be stored in a MAINTAINERS.md file and audited at a minimum of an annual cadence.
- Have a public list of project adopters for at least the primary repo (e.g., ADOPTERS.md or logos on the project website). For a specification, have a list of adopters for the implementation(s) of the spec.
- Receive a supermajority vote from the TOC to move to graduation stage. Projects can attempt to move directly from sandbox to graduation, if they can demonstrate sufficient maturity. Projects can remain in an incubating state indefinitely, but they are normally expected to graduate within two years.

Quotas and Tag Handling

Strict quota handling for storage control with automated garbage collection

Advanced tag policies: retention and immutability

https://kubernetes.docs.cern.ch/docs/registry/best-practices/



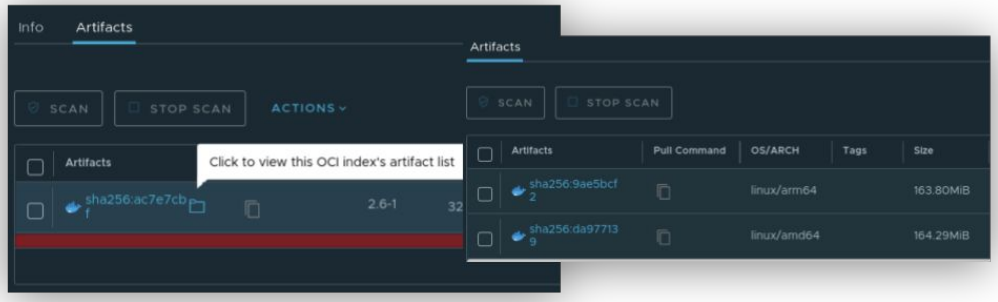
Multiple Architectures

https://kubernetes.docs.cern.ch/docs/registry/quickstart/#multi-arch

Image metadata manifest pointing to multiple artifacts

Similar mechanism for signatures and SBOMs

Ongoing work to support further optimization (CPU capabilities, …)



Continuous Integration with GitLab

https://kubernetes.docs.cern.ch/docs/registry/gitlab/

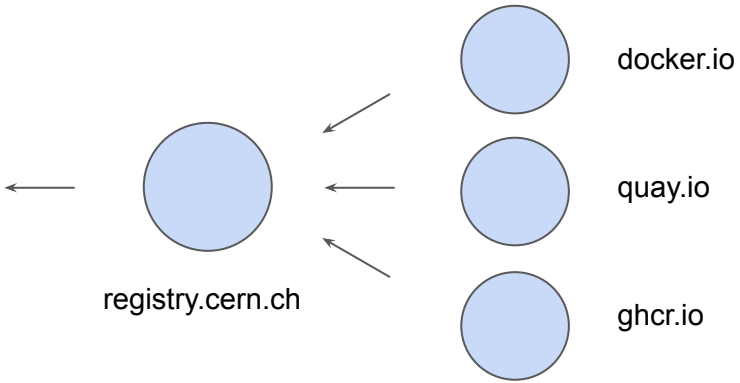
REGISTRY_USER	robot account, usually at group level
REGISTRY_PASSWORD	robot account, usually at group level
REGISTRY_IMAGE_PATH	registry.cern.ch/myrepo/myimage:mytag
PUSH_IMAGE	true/false
ACCELERATED_IMAGE	true/false
PLATFORMS	linux/amd64,linux/arm64
COSIGN_PRIVATE_KEY	base64 encoded, usually at group level

Proxy Caches

Pull-through cache for other registries

https://kubernetes.docs.cern.ch/docs/registry/quickstart/#pull-through-caches

Useful to go around rate limiting, enforced CVE/Vulnerability checks



Automated Replication

Push to target or *pull from source*

Triggered manually, scheduled or event based

In use for integration with TN, public clouds, sync with upstream registries

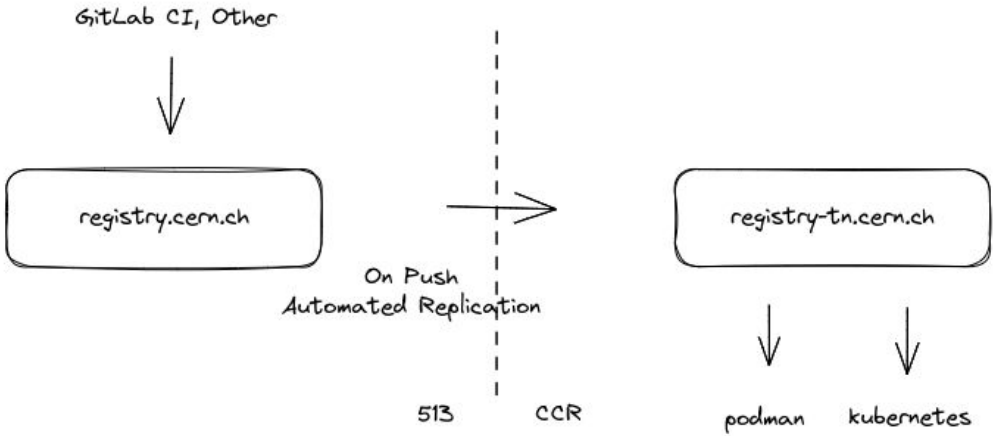
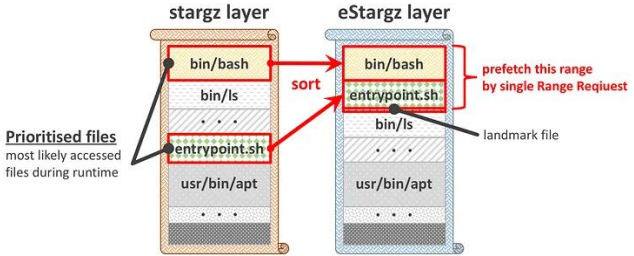


Image Lazy Pulling

https://kubernetes.docs.cern.ch/docs/runtime/lazypulling/

File based (not layer) pull at runtime, on request



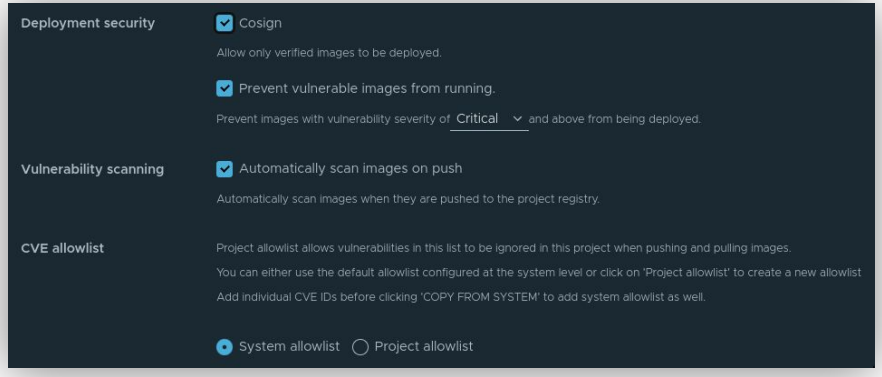
atlas/athena (~17GB)

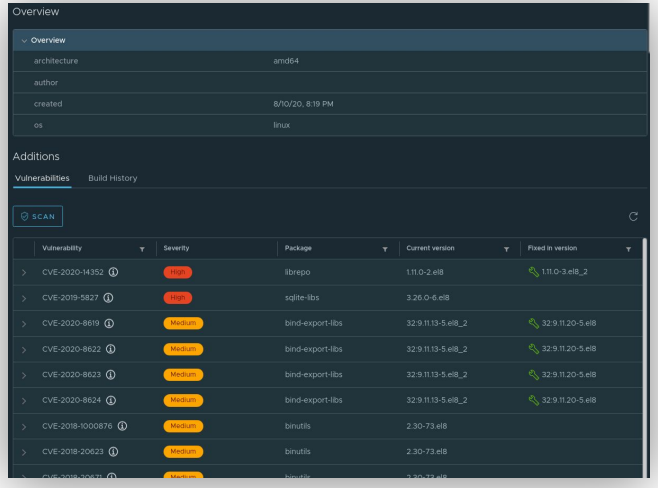
mode	**pulling time**	RAM	ingress on node	execution time
native	**3m37s**	257MB	5.84GB	7m15s
esgz	**16s**	1360MB	0.84GB	8m14s

Security Enhancements

Automated vulnerability scans, image signing, SBOM generation (preview)

Policy enforcement for image access, CVE severity, allowlists, …





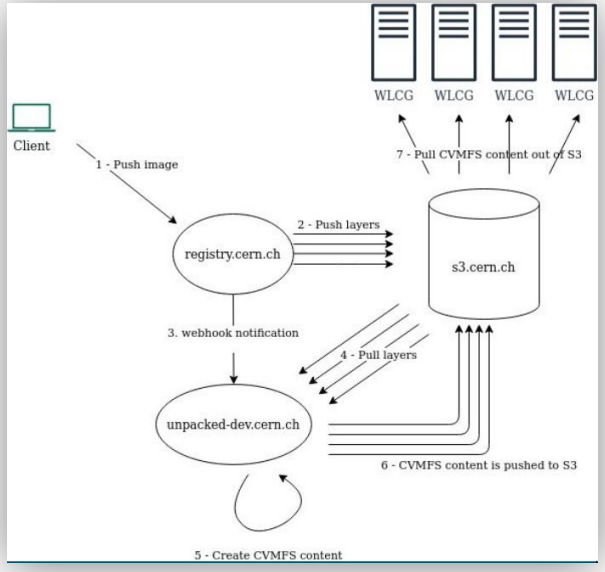
Integration Points

Available webhooks for multiple events

Artifact PUSH, PULL, DELETE,

Image scan COMPLETED, FAILED

Project Quota EXCEED



Example usage: unpacked CVMFS, ATS workflows

Ongoing Work

Business Continuity / Disaster Recovery

Instance in the PDC, S3 bucket backup in 2nd Network Hub, Public Cloud

Improved SBOM integrations

Fully supported but recent work in the upstream project for visualization

Provenance and runtime checks

Zero vulnerabilities at pull time, but days later? And who's using it?

Harbor improvements: job dashboard, easier robot accounts, CloudEvents support, …

Other Work

Ongoing thread with OSG for integration with Soteria

In a similar way to the GPN-TN automated replication described before

Questions?