

**RESILIENCE
REALIZED**



KubeCon



CloudNativeCon

North America 2021



KubeCon



CloudNativeCon

North America 2021

RESILIENCE

REALIZED

Harbor : enterprise cloud native artifact registry

Steven Zou, VMware

Alex Xu, VMware

Yan Wang, VMware

Ziming Zhang, VMware

Harbor

Harbor is an open source **trusted cloud native registry** that stores, signs, and scans content. The mission is to provide cloud native environments the ability to confidently manage and serve container images and related artifacts.



<https://github.com/goharbor/harbor>



<https://goharbor.io>



Key Features

- **Access control**
 - RBAC
 - Project isolation
- **Management policy**
 - Immutability
 - Quotas
 - Retention
- **Artifact distribution**
 - Replication
 - Proxy cache
 - P2P preheat
- **Security & Compliance**
 - Signing
 - Scanning
 - CVE allowlist
- **Extensibility - Compatible with existing investments in infra & services**
 - Integration with LDAP, OIDC provider for authentication
 - Robot Accounts
 - Webhook

Main features/enhancements:

- **Declarative Config:** Enhanced the current configuration by adding environment variables to overwrite the Harbor configuration including auth, backing store, system permissions and more. Addresses issue [#8076](#).
- **IP V6:** Support running Harbor in Kubernetes with harbor-helm on IPv6-only infrastructure
- **Photon Upgrade:**
 - Postgresql upgrade to v13.3
 - Redis upgrade to v6.0.13.

Additional updates:

- Performance enhancement for concurrent requests.
- Observability Metrics Improvement: Support Jobservice metrics.
- Swagger API Improvement: The APIs of project scanners & webhooks support both project id and name in their path.
- Replication enhancement to support destination namespace flattening.
- Move the legacy APIs to new programming model.
- Golang v1.15.12. Harbor is now built using Golang v1.15.12 as of this release.
- Bump up Trivy 0.17 which adds support for Java JAR/WAR/EAR archives and Go binaries scanning.

V2.4 – Ongoing release

Key features/enhancements:

- 💡 **Tracing:** Add tracing capabilities to Harbor for enhanced troubleshooting, identifying performance bottlenecks etc.
- **OIDC:** Delete OIDC auth based user
- **Scanning:** Allow to stop scanning jobs
- **Replication:** Allow exclusions in replication rule; introduce bandwidth limitation in replication rule

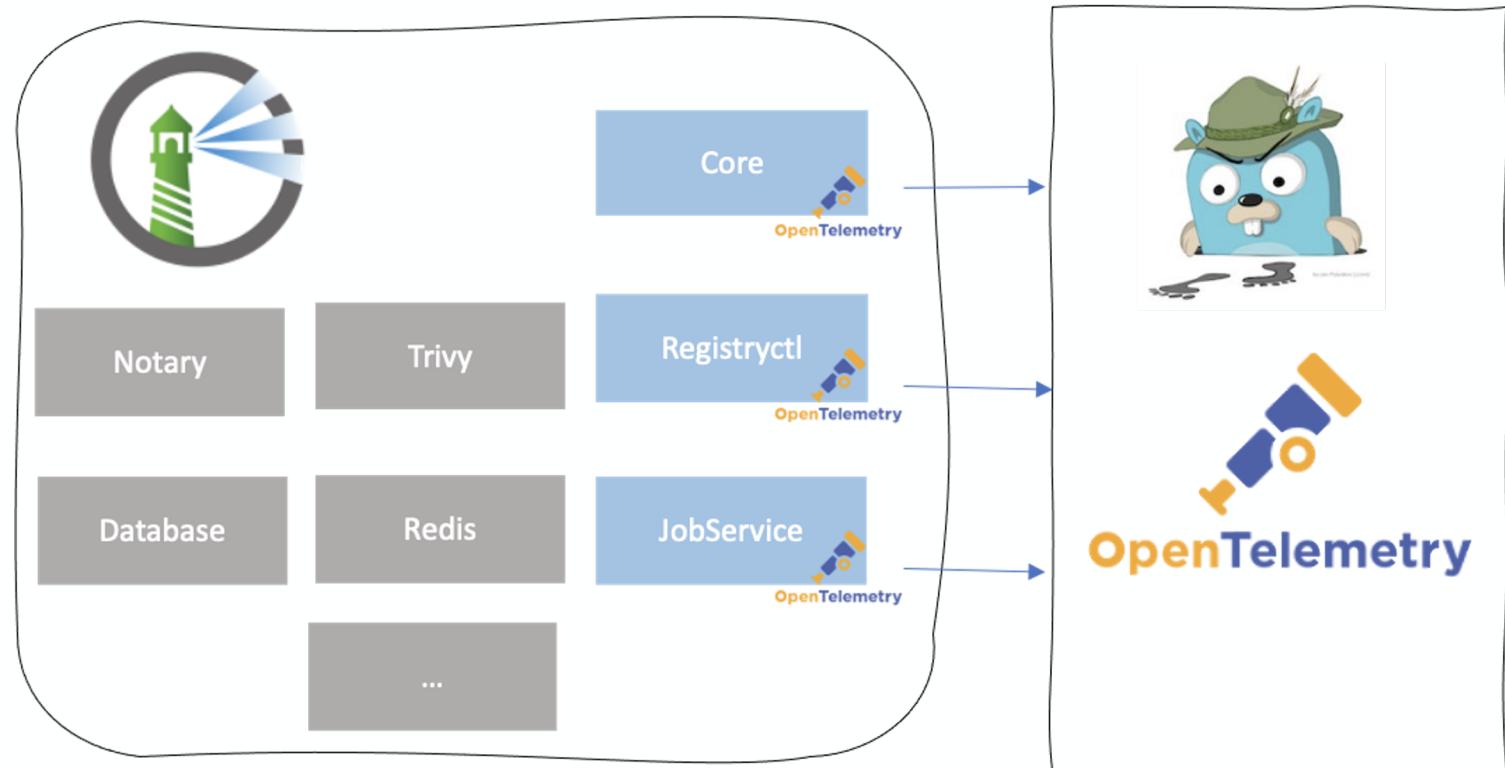
V2.4 Highlight - Tracing

- Will tracing the data of following three components

- Harbor-core
 - Http request
 - Transaction
 - Link log ID to span
- Jobservice
 - Http reuquest
 - Task status and errors
- Registryctl
 - HTTP request
 - GC details

- Supported Tracing Backend

- Jaeger
 - Export data to jaeger directly
- OTLP
 - Export to OpenTelemetry-Collector
 - OpenTelemetry-Collector support almost all the tracing service



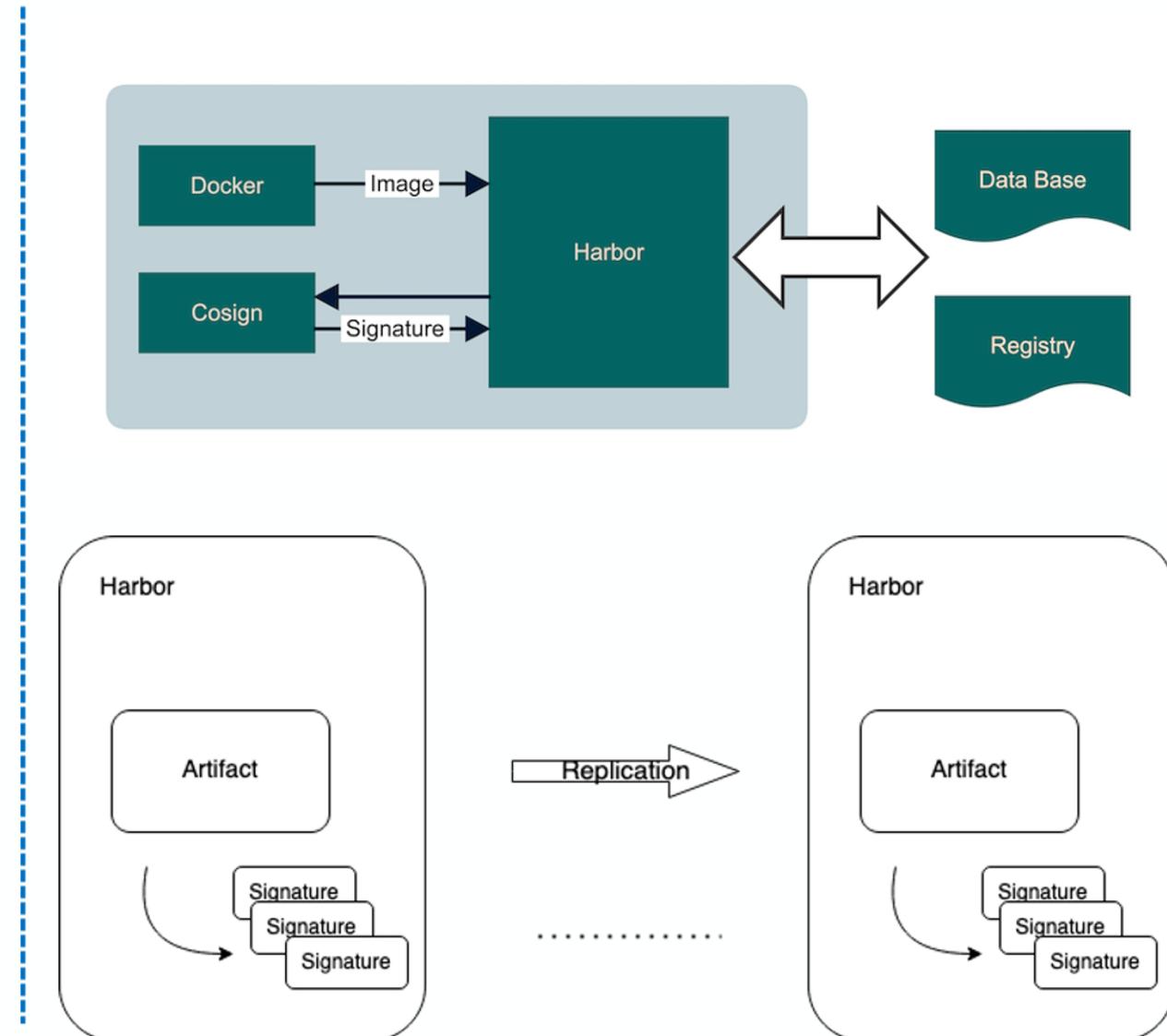
Roadmap

Theme	Description	Timeline
Harbor Lite for Edge	Create a lightweight Harbor with reduced feature set for minimal footprint	2021 H2
Deployment	Improve Kubernetes Operator for Harbor, enabling improved Day1 and Day2 operations including enterprise grade HA, faster deployments and upgrades, automate backups and restores, and sensible defaults	2021 H2
Image Signing 	Leverage Notary v2 as well as Sigstore Cosign to deliver persisting image signatures across image replications	2021 H2
Regex Support	Add full Regex support to all modules within Harbor consistently including configuration of replication policies, retention policies, immutability policies and more	2021 H2
Backup & Restore	Leverage Project Velero to offer application-aware Harbor backup, including databases, Kubernetes objects and Persistent Volumes	2022 H1

Roadmap highlight - Image Signing

Cosign Integration:

- Support Cosign to sign a OCI artifact, and store signatures in Harbor.
- Support replicate artifact signatures to another Harbor instance.
- Signatures are considered not subject to garbage collection if they are associated with an existing artifact.
- Policy enforcement to disallow pulling an unsigned image in Harbor.



Harbor Community - Thriving



KubeCon



CloudNativeCon

North America 2021

GitHub Stars

15.7K+

Committers

200+

Contributing Companies

50+

Contributors

3000+

Forks

4000+**12 Maintainers across 5 companies**

Commits

12K+

GitHub Views/Visitors(2w)

50K/12K+

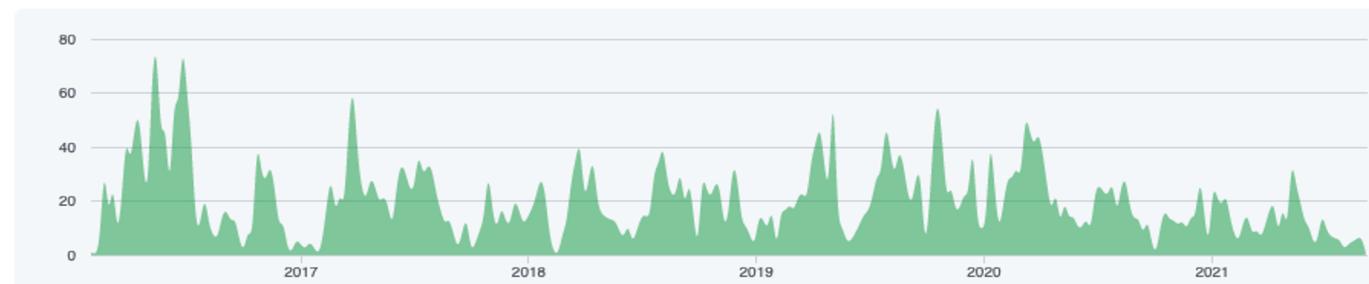
Downloads

10K+

Jan 31, 2016 – Sep 8, 2021

Contributions: Commits ▾

Contributions to master, excluding merge commits and bot accounts

**Harbor Community**

Workgroups

Workgroup is a virtual team focusing on a specified topic and delivers features related to the topic by aggregating efforts across all the interested parties.

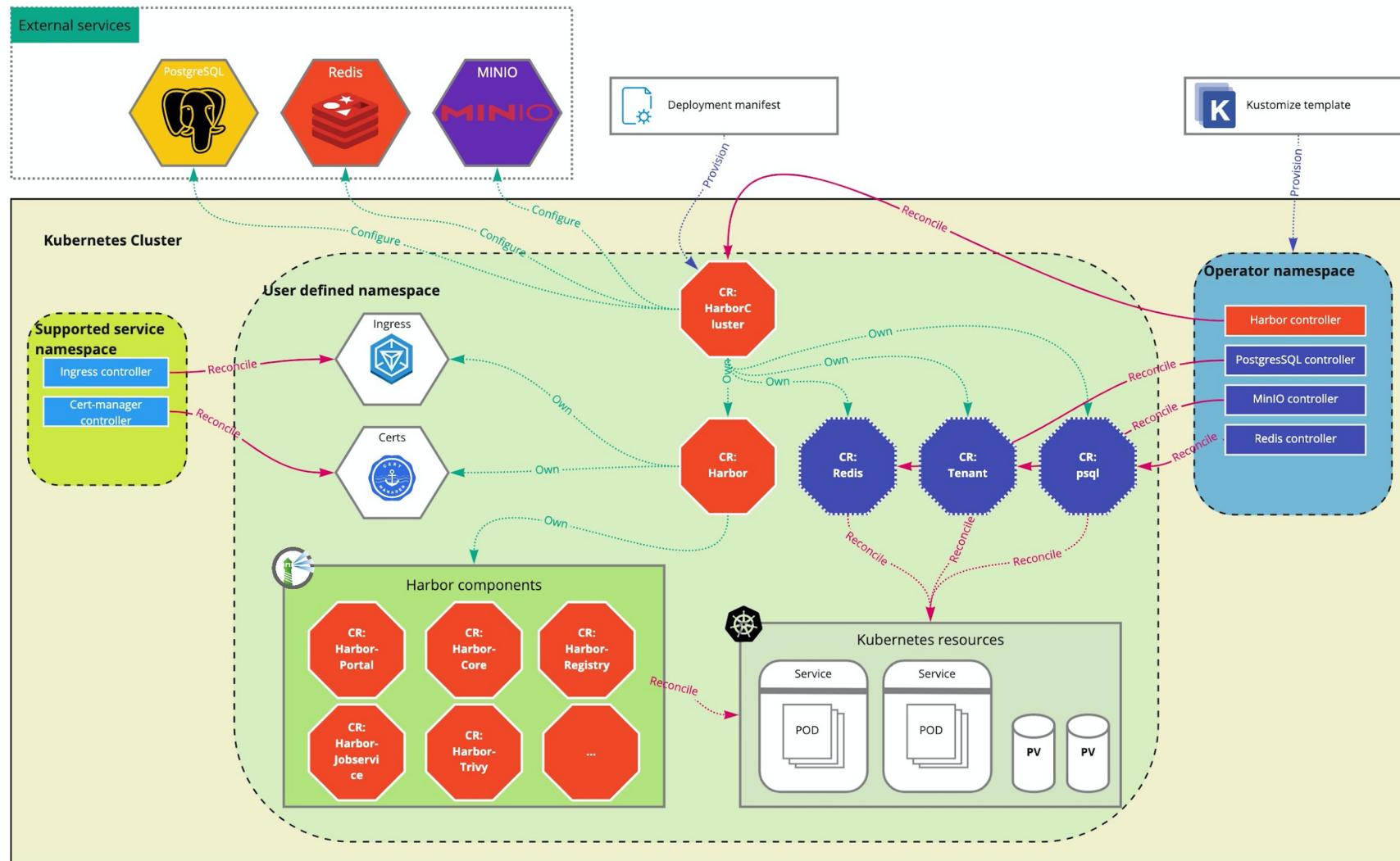
- **WG: Replication**
 - Design and maintain Harbor replication feature
 - Verifications of Supported adapters: <https://github.com/goharbor/replication-verification#report>
- **WG: P2P Distribution**
 - Design and maintain the P2P distribution framework
 - Dragonfly & Kraken are supported
- **WG: Scanning**
 - Design and maintain the pluggable scanner framework and specification
 - Spec: <https://github.com/goharbor/pluggable-scanner-spec>
 - Slack: #harbor-scanning-wg

Workgroups - New

- WG: Multi Architecture
 - Enable Harbor to support multiple architectures like X86 and arm etc.
 - arm arch: <https://github.com/goharbor/harbor-arm>
 - loongson arch: <https://github.com/goharbor/harbor-loongson>
 - Slack: #harbor-multi-arch-workgroup
- WG: Performance
 - Improve the performance and scalability of Harbor in the use case of large-scale data
 - Created perf testing tools (a set of scripts) to evaluate Harbor perf : <https://github.com/goharbor/perf>
 - Improved perf of APIs (list artifacts and v2 get catalog)
 - Run perf testing for some APIs of V2.3
 - Slack: #harbor-perf-workgroup
- WG: Image acceleration
 - Introduce and integrate accelerated image format support into Harbor
 - General image acceleration service: <https://github.com/goharbor/acceleration-service>
 - Slack: #harbor-image-accelerator
- WG: Operator
 - Design and develop harbor operator for achieving better experiences of running Harbor on Kubernetes
 - Repo: <https://github.com/goharbor/harbor-operator>
 - Slack: #harbor-operator-dev

WG Highlight: Harbor Operator

Overall architecture: how does it work



Key capabilities:

- Manage both Harbor registry and its dependent services
- Support both configuring existing and deploy own dependent services
- Deploy Harbor with the stack matching your requirements
- In Scalable and high-available way
- Cover Day2 operations
- Good extensibilities to support different vendors of the dependent services

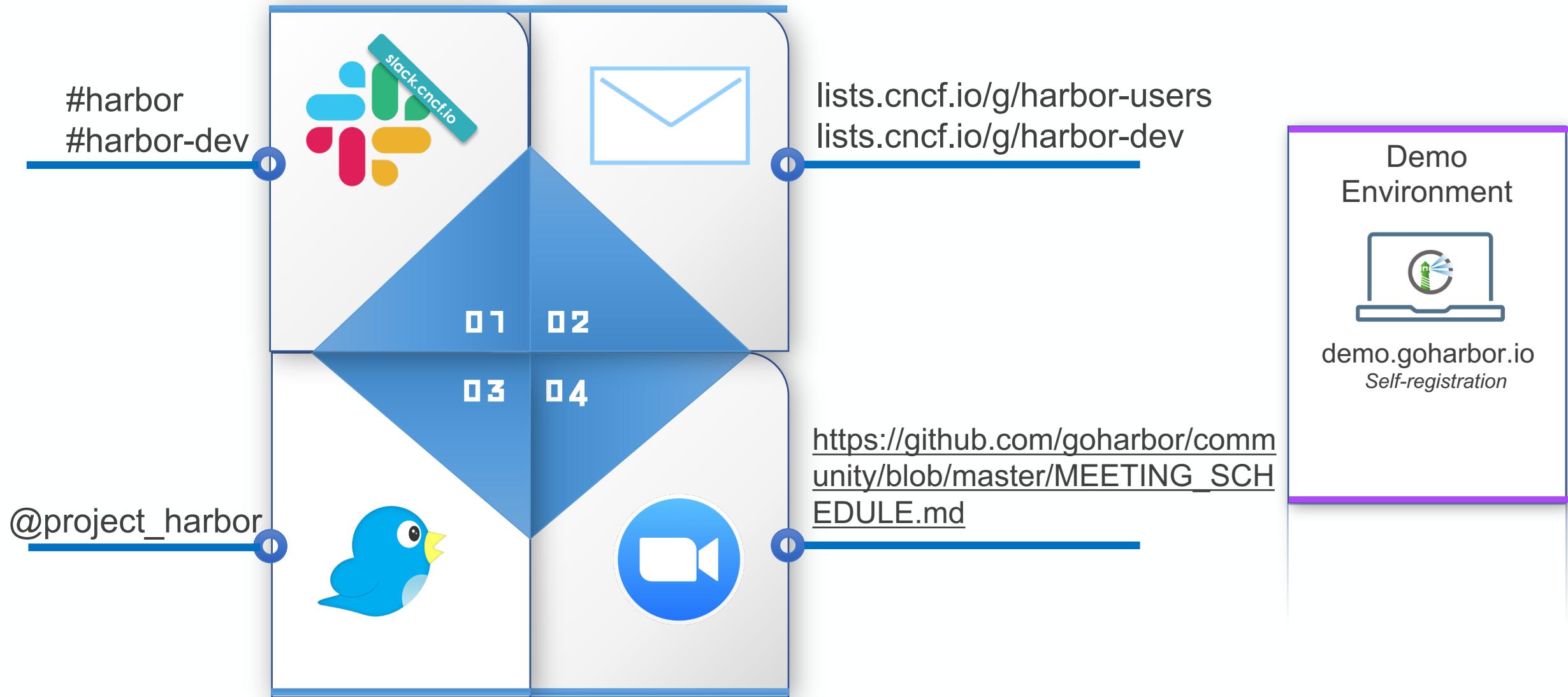
Harbor operator: Roadmap

V1.1 (GA: 08/22)	V1.2 (Developing)	Long-term
Support deploying Harbor v2.3	Support deploying Harbor v2.4	Data backup and restore
Support Kubernetes version 1.21	Support Kubernetes version 1.22	Manage Harbor resources with the declarative way (CRD based)
Upgrade ingress version to v1	Supply more day2 operations	Day2 operation: harbor CA cert population
Refine CRD definitions to provide consistent database, storage, and cache configuration spec (introduce new CRD version v1beta1)	Extend storage drive supporting list (azure, OSS and GCS)	Support other dependent service vendors
Support applying day2 configuration with CRD-based declaration way	Make Harbor portal component optional	Support `scale` subresources of Harbor components
Expose additional configuration options for S3 storage	Let registry and registrtctl live in one pod	Expose harbor service in LB/Nodeport way
		Easy deployment of operator

Demo time: Harbor Operator



Collaboration



RESILIENCE
REALIZED



KubeCon



CloudNativeCon

North America 2021

Thank you!