SUSE

Getting started with SUSE Private Registry

Getting started with SUSE Private Registry

Publication Date: 2025-10-15

https://documentation.suse.com

Contents

Release notes

SUSE Private Registry is an on-premises container registry. It is designed for SUSE customers who need a container registry that works well with other SUSE services and products.

This document provides a high-level overview of the features, capabilities and limitations of SUSE Private Registry, and highlights important product updates.

1 Release 1.0.1

Security updates:

- CVE-2025-55198: Helm may panic due to incorrect YAML content.
- CVE-2025-55199: Helm charts with specific JSON schema values can cause memory exhaustion.
- CVE-2025-54410: Moby versions before 25.0.13, when firewall reloads, Docker fails to re-create iptables rules isolating bridge networks. This allows any container to access all ports on any other container across different bridge networks on the same host and breaks network segmentation in multi-tenant environments (only --internal networks remain protected).
- CVE-2025-29923: go-redis allows potential out of order responses when `CLIENT SETINFO` times out during connection establishment.
- CVE-2025-54388: Moby versions 28.2.0–28.3.2 fails to re-create iptables rules after a firewall reloads. This exposes containers with localhost-published ports (e.g., 127.0.0.1:8080) to remote access via the Docker bridge, while unpublished ports remain protected; fixed in version 28.3.3.
- GHSA-2464-8j7c-4cjm: go-viper's map structure may leak sensitive information in logs when processing malformed data.
- CVE-2025-8959: HashiCorp go-getter vulnerable to arbitrary read through symlink attack.
- CVE-2025-58058: github.com/ulikunitz/xz leaks memory when decoding a corrupted multiple LZMA archives.
- CVE-2025-53547: Helm chart dependency updating with malicious Chart.yaml content and symlink can lead to code execution.

Bugs fixed:

- Trivy: the correct version is shown when calling `trivy version`.

Container image updates:

- Valkey updated from 8.0.2 → 8.0.6.

Upgrade notes:

- No breaking changes in this release.

2 Release 1.0

Key features:

- SUSE Private Registry is based on Harbor 2.13.2
  - Integration with Model Spec (https://github.com/goharbor/community/blob/main/proposals/new/AI-model-processor.md) for first-class handling of AI models
  - Enhanced audit logging
- Predictable release cycle (https://www.suse.com/support/kb/doc/?id=000021405) aligned with SUSE Rancher Prime. SUSE Private Registry will be updated every 4 months
- Each release is supported by SUSE for 18 months from the date of release
  - 6 months of security and bug fix maintenance, followed by
  - 12 months of security-only maintenance
- Can be used to mirror SUSE Application Collection (https://docs.apps.rancher.io/how-to-guides/mirror-with-harbor/)
- Supports SUSE Security (https://documentation.suse.com/cloudnative/security/5.4/en/harbor.html) as an external scanner

SUSE Private Registry includes all the features of Harbor:

- On-premises private container image and OCI artifact registry
- Web interface for administration

- Role-based Access Control
- Fine-grained project configuration for image and artifact storage
- Mirroring and pull-through caching of upstream registries' artifacts
- Image retention and garbage collection controls
- Scanning images for security vulnerabilities with the Trivy scanner
- Generate SBOMs for stored images
- Content trust with Cosign (Notary is not included)

Copyright

1 Introduction

1.1 What is SUSE Private Registry?

SUSE Private Registry (Private Registry) is an on-premises container registry. Private Registry is designed for SUSE customers who need a container registry that works well with other SUSE services and products.

1.2 What are SUSE Private Registry benefits?

Private Registry is based on the Harbor project and includes all its core features as well as added benefits. For example:

- **On-premises container registry.** Private Registry is a locally hosted container registry with access to online SUSE registry services.
- **Security.** Private Registry offers security considerations for containerized environments. It includes authentication, authorization and vulnerability scanning.
- **Deployment flexibility.** You can install Private Registry on a Kubernetes environment such as SUSE Rancher Prime: RKE2. You can also deploy Private Registry with High Availability setup.
- **User management.** Private Registry provides authentication and authorization mechanism with role-based access control (RBAC).
- **User interface.** Besides a command-line interface, you can administer Private Registry via Web user interface.

1.3 How does SUSE Private Registry work?

Private Registry is delivered as *Open Container Initiative* (OCI) containers and is expected to be deployed on a Kubernetes cluster. Private Registry consists of the following containers:

- **harbor-core:** the main component of the Harbor registry, responsible for handling core functionalities such as managing projects, repositories and user interactions.
- **harbor-db:** the database container that stores all metadata related to images, users and configurations for the Harbor registry.
- **harbor-jobservice:** a service that manages background jobs, such as image replication and scheduled tasks, ensuring efficient processing of operations within the registry.
- **harbor-nginx:** the reverse proxy and load balancer that routes incoming requests to the appropriate Harbor services, providing a single entry point for users.
- **harbor-portal:** the Web-based user interface that allows users to interact with the Harbor registry, manage images, and configure settings through a graphical interface.
- **harbor-registry:** the container that serves as the actual image storage back-end, handling the storage and retrieval of container images.
- **harbor-registryctl:** a command-line tool for managing the Harbor registry, allowing users to perform administrative tasks and configurations directly from the terminal.
- **harbor-trivy-adapter:** a container that integrates the Trivy vulnerability scanner with Harbor, enabling automated security scanning of container images for vulnerabilities.
- **harbor-exporter:** the container that exports Harbor metrics in a format that can be collected by Prometheus for monitoring and observability.
- **harbor-valkey:** an in-memory key-value store.

After deployment, you can log in via Web user interface. After successful authentication and authorization, you can configure multiple aspects of the product, for example:

- Configure **global settings**, such as setting the registry to read-only mode or restricting who can create projects.
- Select an **authentication** method.
- Add users when in database authentication mode and assign the system **administrator role** to other users.

- Apply resource **quotas** to projects.
- Set up the **replication** of images between Private Registry instances.

1.4 For more information

Refer to the following sources to obtain more details:

- The Harbor project homepage is at https://goharbor.io/.
- Harbor usage is detailed in https://goharbor.io/docs.

2 Deployment

The following procedures describe how to deploy SUSE Private Registry (Private Registry) on a Kubernetes cluster.

2.1 Prerequisites

- A Kubernetes cluster version 1.20 or higher
- Helm version 3.2.0 or higher
- Persistent Volume (PV) provisioner support in your infrastructure
- An active subscription for SUSE Private Registry

2.2 Obtaining Kubernetes secrets from the SUSE Customer Center

To download and install the Private Registry images from SUSE Registry, you need a Kubernetes secret with SUSE Customer Center (SCC) mirroring credentials. To obtain the credentials from SCC, follow these steps:

1. Visit SUSE Customer Center at https://scc.suse.com and log in.
2. Select the organization with an active Private Registry subscription from the left sidebar.
3. Select `Proxies` in the top menu. The credentials are displayed in the top right corner.
4. To see the password, click the 'eye' icon.
5. Create a `password.txt` file containing the obtained password.

   ```
   $ head -1 ./password.txt | helm registry login registry.suse.com \
   --username <PRIVATE_REGISTRY_USERNAME> --password-stdin
   ```

6. Create a namespace for SUSE Registry.

   ```
   $ kubectl create namespace <PRIVATE_REGISTRY_NAMESPACE>
   ```

7. Store the mirroring credentials retrieved from SCC as Kubernetes secrets by running the following command:

```
$ kubectl create secret docker-registry suse-registry \
--namespace <PRIVATE_REGISTRY_NAMESPACE> \
--docker-server=registry.suse.com \
--docker-username=<PRIVATE_REGISTRY_USERNAME> \
--docker-password=$(head -1 ./password.txt)
```

8. Optionally, to use TLS encrypted communication, create a TLS secret from your private key and certificate files.

```
$ kubectl create secret tls suse-registry-tls \
--namespace <PRIVATE_REGISTRY_NAMESPACE> \
--cert=<CERTIFICATE>.pem \
--key=<PRIVATE_KEY>.pem
```

2.3 Installing and running Private Registry using Helm

The following procedure describes how to install Private Registry using Helm. Replace `<RELEASE_NAME>` with your custom release name for the Helm chart deployment.

1. Log in to SUSE Registry using the obtained SCC mirroring credentials.

```
$ head -1 ./password.txt | helm registry login registry.suse.com \
--username <SUSE_REGISTRY_USERNAME> --password-stdin
```

2. Install the latest version of the Private Registry Helm chart.

```
$ helm install <RELEASE_NAME> \
oci://registry.suse.com/private-registry/private-registry-helm \
--namespace <PRIVATE_REGISTRY_NAMESPACE>
```

To override the default installation with custom values from the `suse_registry_override.yaml` file, refer to *Appendix A, Overriding the SUSE Private Registry Helm chart*.

The command starts deploying several related containers and may take several minutes to complete. It also prints a message with the URL to the Private Registry Web portal and commands to obtain the administrator credentials.

2.4 Upgrading Private Registry

To upgrade the release of the Helm chart to a specific newer version, run the following command:

```
$ helm upgrade <RELEASE_NAME> \
oci://registry.suse.com/private-registry/private-registry-helm --version
 <NEW_VERSION_OF_HELM_CHART>
--namespace <PRIVATE_REGISTRY_NAMESPACE>
```

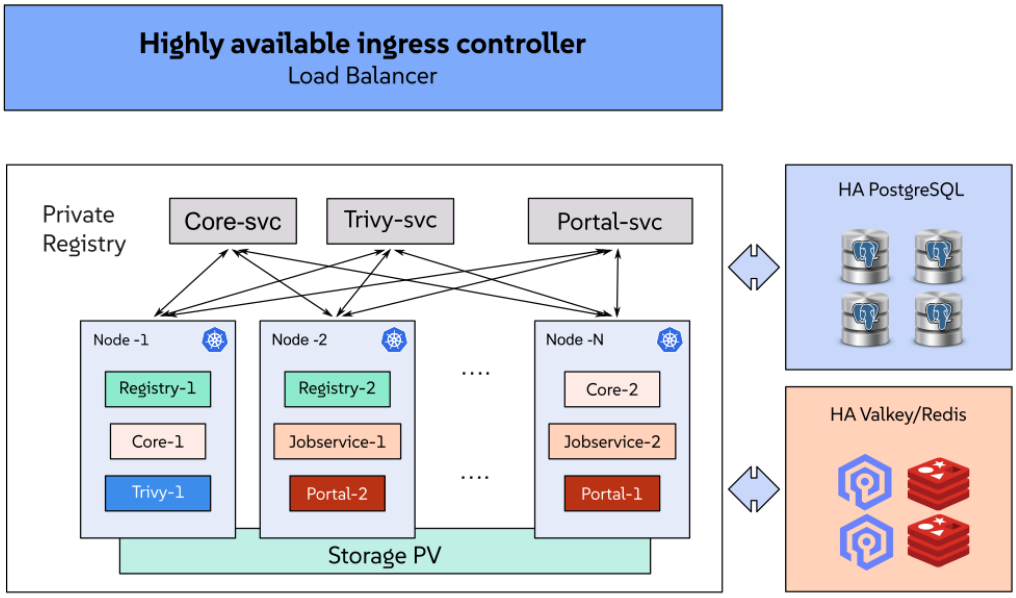
3 Deployment with High Availability

You can use Helm to deploy highly available (HA) Private Registry on a Kubernetes cluster. The HA setup ensures that users do not experience interruptions of service if one of the nodes on which Private Registry is running becomes unavailable.

3.1 Architecture of the HA setup

Most of the Private Registry components are now stateless. Therefore, we can scale them by increasing pod replicas, ensuring they run on multiple worker nodes. Kubernetes services ensure connectivity between pods.

For storage, users should provide an HA PostgreSQL and Valkey or Redis cluster for application data, along with PVCs or object storage for storing images and charts.



Private Registry HA setup. Kubernetes cluster with Ingress in HA setup using HA PostgreSQL and HA Valkey.

3.2 Prerequisites

- A Kubernetes cluster version 1.20 or higher
- Helm version 3.2.0 or higher
- HA Ingress controller (Private Registry does not manage the external endpoint)
- HA PostgreSQL 9.6+ (Private Registry does not handle the deployment of HA database)
- HA Valkey or Redis (Private Registry does not handle the deployment of HA Valkey or Redis)
- Persistent Volume Claim (PVC) that can be shared across nodes or external object storage
- An active subscription for SUSE Private Registry

3.3 Deploying Private Registry with HA

1. Download the Private Registry Helm chart.

   ```
   $ helm pull oci://registry.suse.com/private-registry/private-registry-helm --untar
   ```

2. Update the deployment parameters to match your requirements. Refer to *Appendix B, Example of a Private Registry HA setup Helm chart* for an example Helm chart for Private Registry HA setup. Refer to *Appendix A, Overriding the SUSE Private Registry Helm chart* for a complete list of values to specify or override.

3. Install the Private Registry Helm chart. Replace `<RELEASE_NAME>` with your custom release name for the Helm chart deployment.

   ```
   $ helm install <RELEASE_NAME> private-registry-helm/
   ```

A Overriding the SUSE Private Registry Helm chart

The SUSE Private Registry (Private Registry) Helm chart is delivered with default values. You can adjust the Helm chart installation in one of the following ways:

- Append specific parameters to the `--set` flags on the `helm install` command line, for example:

```
$ helm install <RELEASE_NAME> \
oci://registry.suse.com/private-registry/private-registry-helm \
--namespace <PRIVATE_REGISTRY_NAMESPACE> \
--set harborAdminPassword=<MY_PASSWORD> \
--set externalURL=https://<PRIVATE_REGISTRY_FQDN> \
--set expose.ingress.hosts.core=<PRIVATE_REGISTRY_FQDN>
```

- Create a SUSE custom `suse_registry_override.yaml` file and pass it to the `--f` flag, for example:

```
  $ helm install <RELEASE_NAME> \
  oci://registry.suse.com/private-registry/private-registry-helm \
  --namespace <PRIVATE_REGISTRY_NAMESPACE>
  -f suse_registry_override.yaml
```

A1 Examples of SUSE Registry Helm override files

EXAMPLE A1: MINIMAL DEPLOYMENT WITH INGRESS

```
expose:
  type: ingress ❶
  ingress:
    hosts:
      core: <PRIVATE_REGISTRY_FQDN> ❷

externalURL: https://<PRIVATE_REGISTRY_FQDN> ❸

harborAdminPassword: "<MY_PASSWORD>" ❹

database:
  internal:
    password: "<MY_PASSWORD_POSTGRESQL>"

redis:
  internal:
```

```
   password: "<MY_PASSWORD_REDIS>"
```

1. How SUSE Registry is exposed. Can be `ingress`, `loadBalancer`, `nodePort` or `clusterIPhis`. Default is `ingress`.
2. Host name for the Kubernetes internal networking configuration.
3. URL where the SUSE Registry application runs. It is used to generate links in the user interface, redirects and also for API responses.
4. The administrator password to the application.

EXAMPLE A2: TYPICAL DEPLOYMENT WITH `loadBalancer`

```
expose:
  type: loadBalancer (1)
  tls:
    enabled: true
    certSource: secret (2)
    secret:
      secretName: <SECRET_NAME>

    auto:
      commonName: <PRIVATE_REGISTRY_FQDN> (3)

externalURL: https://<PRIVATE_REGISTRY_FQDN> (4)

harborAdminPassword: "<MY_PASSWORD>" (5)

database:
  internal:
    password: "<MY_PASSWORD_POSTGRESQL>"

redis:
  internal:
   password: "<MY_PASSWORD_REDIS>"
```

1. How SUSE Registry is exposed. Can be `ingress`, `loadBalancer`, `nodePort` or `clusterIP`. Default is `ingress`.
2. Can be `auto`, `secret` or `none`. Depending on the option, you may have to include additional values.
3. When using TLS encryption, this field must match the `externalURL` value.
4. URL where the SUSE Registry application runs. It is used to generate links in the user interface, redirects and also for API responses.
5. The administrator password to the application.

A2 Overriding Helm chart parameters and values

The following tables list all parameters with descriptions that you can use to override the default installation values.

GLOBAL PARAMETERS

`global.imageRegistry`
: Sets a global override for the container image registry used for all images.

`global.imagePullSecrets`
: Sets global pull secrets for accessing the container image registry.

COMMON PARAMETERS

`harborAdminPassword`
: Sets the initial password for Harbor administrator. Change it from portal after deployment. Default is `Harbor12345`.

`externalURL`
: Specifies the external URL for `harbor-core` service. Default is `https://core.harbor.domain`.

`existingSecretAdminPasswordKey`
: Sets the key name in the secret containing Harbor administrator password. Default is `HARBOR_ADMIN_PASSWORD`.

`imagePullSecrets`
: Sets the `imagePullSecrets` names for all deployments.

`updateStrategy.type`
: Sets the update strategy for deployments with persistent volumes. Accepts `RollingUpdate` or `Recreate`. Use `Recreate` when RWM for volumes is not supported. Default is `RollingUpdate`.

`logLevel`
: Sets the log level for Harbor services. Accepts `fatal`, `error`, `warn`, `info`, `debug` or `trace`. Default is `debug`.

`enableMigratehelmHook`
: Runs database migration job via Helm hook. When `true`, separates migration job from `harbor-core`. Default is `false`.

`caSecretName`

Specifies the secret name containing the `ca.crt` key.

PROXY PARAMETERS

`proxy.httpProxy`

Specifies the HTTP proxy server URL. Default is `""`.

`proxy.httpsProxy`

Specifies the HTTPS proxy server URL. Default is `""`.

`proxy.noProxy`

Sets URLs that bypass the proxy configuration. Default is `127.0.0.1,localhost,.local,.internal`.

`proxy.components`

Sets components that use the proxy configuration. Default is `["core","jobservice","trivy"]`.

EXPOSE PARAMETERS

`expose.type`

Specifies service exposure type: `ingress`, `clusterIP`, `nodePort` or `loadBalancer`. Default is `ingress`.

`expose.tls.enabled`

Enables TLS. Default is `true`.

`expose.tls.certSource`

Sets TLS certificate source as `auto`, `secret` or `none`. Default is `auto`.

`expose.tls.auto.commonName`

Sets certificate common name when type is not `ingress`.

`expose.tls.secret.secretName`

Specifies name of secret containing `tls.crt` (certificate) and `tls.key` (private key).

`expose.ingress.hosts.core`

Sets Harbor core service host in Ingress rule. Default is `core.harbor.domain`.

`expose.ingress.controller`

Sets Ingress controller type. Supports `default`, `gce`, `alb`, `f5-bigip` and `ncp`. Default is `default`.

`expose.ingress.kubeVersionOverride`
: Overrides Kubernetes version for Ingress templating.

`expose.ingress.annotations`
: Sets Ingress annotations.

`expose.ingress.labels`
: Sets Ingress-specific labels. Default is `{}`.

`expose.clusterIP.name`
: Sets ClusterIP service name. Default is `harbor`.

`expose.clusterIP.annotations`
: Sets ClusterIP service annotations. Default is `{}`.

`expose.clusterIP.ports.httpPort`
: Sets HTTP service port. Default is `80`.

`expose.clusterIP.ports.httpsPort`
: Sets HTTPS service port. Default is `443`.

`expose.clusterIP.labels`
: Sets ClusterIP-specific labels. Default is `{}`.

`expose.nodePort.name`
: Sets NodePort service name. Default is `harbor`.

`expose.nodePort.ports.http.port`
: Sets HTTP service port. Default is `80`.

`expose.nodePort.ports.http.nodePort`
: Sets HTTP node port. Default is `30002`.

`expose.nodePort.ports.https.port`
: Sets HTTPS service port. Default is `443`.

`expose.nodePort.ports.https.nodePort`
: Sets HTTPS node port. Default is `30003`.

`expose.nodePort.annotations`
: Sets NodePort annotations.

`expose.nodePort.labels`
: Sets NodePort-specific labels. Default is `{}`.

`expose.loadBalancer.name`

Sets service name. Default is `harbor`.

`expose.loadBalancer.IP`

Sets loadBalancer IP when IP assignment is supported. Default is `""`.

`expose.loadBalancer.ports.httpPort`

Sets HTTP service port. Default is `80`.

`expose.loadBalancer.ports.httpsPort`

Sets HTTPS service port. Default is `30002`.

`expose.loadBalancer.annotations`

Sets loadBalancer service annotations. Default is `{}`.

`expose.loadBalancer.labels`

Sets loadBalancer-specific labels. Default is `{}`.

`expose.loadBalancer.sourceRanges`

Specifies IP address ranges for loadBalancerSourceRanges. Default is `[]`.

PERSISTENCE PARAMETERS

`persistence.enabled`

Enables or disables data persistence. Default is `true`.

`persistence.resourcePolicy`

`keep` prevents removal of PVCs during a Helm delete operation. Empty value deletes PVCs after chart deletion. Default is `keep`.

`persistence.persistentVolumeClaim.registry.existingClaim`

The existing PVC that must be created manually before binding. Requires a subPath specification if the PVC is shared with other components.

`persistence.persistentVolumeClaim.registry.storageClass`

The `storageClass` that provisions the volume.

`persistence.persistentVolumeClaim.registry.subPath`

The subpath in the volume.

`persistence.persistentVolumeClaim.registry.accessMode`

The access mode of the volume. Default is `ReadWriteOnce`.

`persistence.persistentVolumeClaim.registry.size`
: The size of the volume. Default is `5Gi`.

`persistence.persistentVolumeClaim.registry.annotations`
: The annotations of the volume.

`persistence.persistentVolumeClaim.jobservice.jobLog.existingClaim`
: The existing PVC that must be created manually before binding. Requires a subPath specification if the PVC is shared with other components.

`persistence.persistentVolumeClaim.jobservice.jobLog.storageClass`
: The `storageClass` that provisions the volume.

`persistence.persistentVolumeClaim.jobservice.jobLog.subPath`
: The subpath in the volume.

`persistence.persistentVolumeClaim.jobservice.jobLog.accessMode`
: The access mode of the volume. Default is `ReadWriteOnce`.

`persistence.persistentVolumeClaim.jobservice.jobLog.size`
: The size of the volume. Default is `1Gi`.

`persistence.persistentVolumeClaim.jobservice.jobLog.annotations`
: The annotations of the volume.

`persistence.persistentVolumeClaim.database.existingClaim`
: The existing PVC that must be created manually before binding. Requires a subPath specification if the PVC is shared with other components.

`persistence.persistentVolumeClaim.database.storageClass`
: The `storageClass` that provisions the volume.

`persistence.persistentVolumeClaim.database.subPath`
: The subpath in the volume. Ignored when an external database is used.

`persistence.persistentVolumeClaim.database.accessMode`
: The access mode of the volume. Ignored when an external database is used. Default is `ReadWriteOnce`.

`persistence.persistentVolumeClaim.database.size`
: The size of the volume. Ignored when an external database is used. Default is `1Gi`.

`persistence.persistentVolumeClaim.database.annotations`
: The annotations of the volume.

`persistence.persistentVolumeClaim.redis.existingClaim`
: The existing PVC that must be created manually before binding. Requires a subPath specification if the PVC is shared with other components.

`persistence.persistentVolumeClaim.redis.storageClass`
: The `storageClass` that provisions the volume. Uses default StorageClass if not specified.

`persistence.persistentVolumeClaim.redis.subPath`
: The subpath in the volume. Ignored when an external Valkey is used.

`persistence.persistentVolumeClaim.redis.accessMode`
: The access mode of the volume. Ignored when an external Valkey is used. Default is `ReadWriteOnce`.

`persistence.persistentVolumeClaim.redis.size`
: The size of the volume. Ignored when an external Valkey is used. Default is `1Gi`.

`persistence.persistentVolumeClaim.redis.annotations`
: The annotations of the volume.

`persistence.persistentVolumeClaim.trivy.existingClaim`
: The existing PVC that must be created manually before binding. Requires a subPath specification if the PVC is shared with other components.

`persistence.persistentVolumeClaim.trivy.storageClass`
: The `storageClass` that provisions the volume. Uses default StorageClass if not specified.

`persistence.persistentVolumeClaim.trivy.subPath`
: The subpath in the volume.

`persistence.persistentVolumeClaim.trivy.accessMode`
: The access mode of the volume. Default is `ReadWriteOnce`.

`persistence.persistentVolumeClaim.trivy.size`
: The size of the volume. Default is `1Gi`.

`persistence.persistentVolumeClaim.trivy.annotations`
: The annotations of the volume.

`persistence.imageChartStorage.disableredirect`
: Controls redirect management from content back-ends. Set to true to disable redirects for unsupported back-ends. Default is `false`.

`persistence.imageChartStorage.caBundleSecretName`

The name of secret containing CA bundle for self-signed storage service certificates.

`persistence.imageChartStorage.type`

The storage type for images and charts: `filesystem`, `azure`, `gcs`, `s3`, `swift`, or `oss`. Default is `filesystem`.

`persistence.imageChartStorage.gcs.existingSecret`

The name of existing secret containing the GCS service account JSON key. The key must be `gcs-key.json`. Default is `""`.

`persistence.imageChartStorage.gcs.useWorkloadIdentity`

Enables workload identity usage in a GKE cluster. Default is `false`.

NGINX PARAMETERS

`nginx.image.repository`

The image repository for nginx. Default is `private-registry/harbor-nginx`.

`nginx.image.tag`

The image tag for nginx.

`nginx.replicas`

The number of replicas to run. Default is `1`.

`nginx.revisionHistoryLimit`

The maximum number of old `ReplicaSet` revisions to retain. Default is `10`.

`nginx.resources`

The compute resources allocated for the container. Default is `undefined`.

`nginx.automountServiceAccountToken`

Controls automatic mounting of the service account token. Default is `false`.

`nginx.nodeSelector`

The node labels used for pod assignment. Default is `{}`.

`nginx.tolerations`

The pod assignment tolerations. Default is `[]`.

`nginx.affinity`

The node or pod affinity rules. Default is `{}`.

`nginx.topologySpreadConstraints`

The rules for spreading pods across failure-domains such as regions or availability zones. Default is `[]`.

`nginx.podAnnotations`

The annotations added to the nginx pod. Default is `{}`.

PORTAL PARAMETERS

`portal.image.repository`

Repository location for the portal image. Default is `private-registry/harbor-portal`.

`portal.image.tag`

Tag for the portal image. Default is `3.11`.

`portal.replicas`

Number of replicas to create. Default is `1`.

`portal.revisionHistoryLimit`

Maximum number of old `ReplicaSet` revisions to retain. Default is `10`.

`portal.resources`

Resources allocated to the container. Default is `undefined`.

`portal.automountServiceAccountToken`

Controls automatic mounting of the service account token. Default is `false`.

`portal.nodeSelector`

Node labels used for pod assignment. Default is `{}`.

`portal.tolerations`

Tolerations used for pod assignment. Default is `[]`.

`portal.affinity`

Node and pod affinity settings. Default is `{}`.

`portal.topologySpreadConstraints`

Defines pod distribution across failure-domains such as regions or availability zones. Default is `[]`.

`portal.podAnnotations`

Annotations added to the portal pod. Default is `{}`.

`portal.serviceAnnotations`

Annotations added to the portal service. Default is `{}`.

`portal.priorityClassName`

Priority class name for pod execution.

`portal.initContainers`

Init containers to be run before the controller container starts. Default is `[]`.

CORE PARAMETERS

`core.image.repository`

The repository for the Harbor core image. Default is `private-registry/harbor-core`.

`core.image.tag`

The tag for the Harbor core image. Default is `2.11`.

`core.replicas`

The number of replicas. Default is `1`.

`core.revisionHistoryLimit`

The revision history limit. Default is `10`.

`core.startupProbe.initialDelaySeconds`

The initial delay in seconds for the startup probe. Default is `10`.

`core.resources`

The resources to allocate for the container. Default is `undefined`.

`core.automountServiceAccountToken`

Mounts the service account token. Default is `false`.

`core.nodeSelector`

The node labels for pod assignment. Default is `{}`.

`core.tolerations`

The tolerations for pod assignment. Default is `[]`.

`core.affinity`

The node or pod affinities. Default is `{}`.

`core.topologySpreadConstraints`

The constraints that define how pods are spread across failure-domains like regions or availability zones. Default is `[]`.

`core.podAnnotations`
: The annotations to add to the core pod. Default is `{}`.

`core.serviceAnnotations`
: The annotations to add to the core service. Default is `{}`.

`core.configureUserSettings`
: A JSON string in the environment variable CONFIG_OVERWRITE_JSON to configure user settings.

`core.quotaUpdateProvider`
: The provider for updating project quota usage, options are `redis` or `db`. Default is `db`.

`core.secret`
: Used when core server communicates with other components.

`core.secretName`
: The name of a Kubernetes secret to use your own TLS certificate and private key for token encryption or decryption.

`core.tokenKey`
: The PEM-formatted RSA private key used to sign service tokens.

`core.tokenCert`
: The PEM-formatted certificate signed by `core.tokenKey` used to validate service tokens.

`core.xsrfKey`
: The XSRF key, automatically generated if not specified.

`core.priorityClassName`
: The priority class to run the pod as.

`core.artifactPullAsyncFlushDuration`
: The time duration for asynchronously updating artifact pull time and repository pull count.

`core.gdpr.deleteUser`
: Enables GDPR compliant user deletion. Default is `false`.

`core.gdpr.auditLogsCompliant`
: Enables GDPR compliance for audit logs by changing username to its CRC32 value if that user was deleted from the system. Default is `false`.

`core.initContainers`
: The init containers to run before the controller's container starts. Default is `[]`.

JOBSERVICE PARAMETERS

`jobservice.image.repository`
: The repository for the jobservice image. Default is `private-registry/harbor-jobservice`.

`jobservice.image.tag`
: The tag for the jobservice image. Default is `2.11`.

`jobservice.replicas`
: The number of replicas. Default is `1`.

`jobservice.revisionHistoryLimit`
: The revision history limit. Default is `10`.

`jobservice.maxJobWorkers`
: The maximum number of job workers. Default is `10`.

`jobservice.jobLoggers`
: The loggers for jobs: `file`, `database` or `stdout`. Default is `[file]`.

`jobservice.loggerSweeperDuration`
: The duration in days to keep job logs (ignored if `jobLoggers` is set to `stdout`). Default is `14`.

`jobservice.notification.webhook_job_max_retry`
: The maximum number of retries for webhook notification sending. Default is `3`.

`jobservice.notification.webhook_job_http_client_timeout`
: The HTTP client timeout in seconds for webhook notification sending. Default is `3`.

`jobservice.reaper.max_update_hours`
: The maximum time in hours to wait for a task to finish. If the task is not finished after the specified hours, it is marked as an error but continues to run. Default is `24`.

`jobservice.reaper.max_dangling_hours`
: The maximum time in hours for execution in running state without a new task created. Default is `168`.

`jobservice.resources`
: The [resources] to allocate for container. Default is `undefined`.

`jobservice.automountServiceAccountToken`
: Mounts the service account token. Default is `false`.

`jobservice.nodeSelector`

The node labels for pod assignment. Default is `{}`.

`jobservice.tolerations`

The tolerations for pod assignment. Default is `[]`.

`jobservice.affinity`

The node or pod affinities. Default is `{}`.

`jobservice.topologySpreadConstraints`

The constraints that define how pods are spread across failure-domains like regions or availability zones. Default is `[]`.

`jobservice.podAnnotations`

The annotations to add to the jobservice pod. Default is `{}`.

`jobservice.priorityClassName`

The priority class to run the pod as.

`jobservice.secret`

The secret used when job service communicates with other components. If a secret key is not specified, Helm generates it. Must be a string of 16 characters.

`jobservice.initContainers`

The init containers to run before the controller's container starts. Default is `[]`.

REGISTRY PARAMETERS

`registry.registry.image.repository`

The repository location for the registry image. Default is `private-registry/harbor-registry`.

`registry.registry.image.tag`

The tag for the registry image. Default is `2.11`.

`registry.registry.resources`

The [resources] to allocate for container. Default is `undefined`.

`registry.controller.image.repository`

The repository location for the registry controller image. Default is `private-registry/harbor-registryctl`.

`registry.controller.image.tag`

The tag for the registry controller image. Default is `2.11`.

`registry.controller.resources`
: The [resources] to allocate for container. Default is `undefined`.

`registry.replicas`
: The number of replica instances. Default is `1`.

`registry.revisionHistoryLimit`
: The maximum number of revisions to maintain in history. Default is `10`.

`registry.nodeSelector`
: The node labels for pod assignment. Default is `{}`.

`registry.automountServiceAccountToken`
: Controls whether to mount the service account token. Default is `false`.

`registry.tolerations`
: The tolerations for pod assignment. Default is `[]`.

`registry.affinity`
: The node or pod affinities. Default is `{}`.

`registry.topologySpreadConstraints`
: The constraints that define pod distribution across failure-domains such as regions or availability zones. Default is `[]`.

`registry.middleware`
: Middleware support for a CDN between back-end storage and Docker pull recipient.

`registry.podAnnotations`
: The annotations to add to the registry pod. Default is `{}`.

`registry.priorityClassName`
: The priority class for pod execution.

`registry.secret`
: The secret that secures the upload state between client and registry storage back-end.

`registry.credentials.username`
: The username for Harbor core's internal registry access. Default is `harbor_registry_user`.

`registry.credentials.password`
: The password for Harbor core's internal registry access. Default is `harbor_registry_password`.

`registry.credentials.existingSecret`

An existing secret containing the password for registry instance access in htpasswd auth mode. Default is `""`.

`registry.credentials.htpasswdString`

The login and password in htpasswd string format. Excludes `registry.credentials.username` and `registry.credentials.password`. Default is `undefined`.

`registry.relativeurls`

Returns relative URLs in Location headers when true. Required if Harbor is behind a reverse proxy. Default is `false`.

`registry.upload_purging.enabled`

Enables purging of upload directories. Default is `true`.

`registry.upload_purging.age`

The time period after which files in upload directories are removed, default is one week. Default is `168h`.

`registry.upload_purging.interval`

The time interval between purge operations. Default is `24h`.

`registry.upload_purging.dryrun`

Enables dryrun mode for upload purging. Default is `false`.

`registry.initContainers`

The init containers that run before the controller's container starts. Default is `[]`.

TRIVY PARAMETERS

`trivy.enabled`

Enables or disables the Trivy scanner. Default is `true`.

`trivy.image.repository`

The repository for the Trivy adapter image. Default is `private-registry/harbor-trivy-adapter`.

`trivy.image.tag`

The tag for the Trivy adapter image. Default is `2.11`.

`trivy.resources`

The resources to allocate for the Trivy adapter container. Default is `undefined`.

`trivy.automountServiceAccountToken`
: Whether to mount the service account token. Default is `false`.

`trivy.replicas`
: The number of Pod replicas. Default is `1`.

`trivy.debugMode`
: Enables Trivy debug mode for troubleshooting. Default is `false`.

`trivy.vulnType`
: Comma-separated list of vulnerability types (`os` and `library`). Default is `os,library`.

`trivy.severity`
: Comma-separated list of vulnerability severities to check. Default is `UNKNOWN,LOW,MEDIUM,HIGH,CRITICAL`.

`trivy.ignoreUnfixed`
: Displays only fixed vulnerabilities. Default is `false`.

`trivy.insecure`
: Skips registry certificate verification. Default is `false`.

`trivy.skipUpdate`
: Disables Trivy database downloads from GitHub. Default is `false`.

`trivy.skipJavaDBUpdate`
: Requires manual download of the `trivy-java.db` file when enabled. Default is `false`.

`trivy.offlineScan`
: Prevents Trivy from sending API requests to identify dependencies. Default is `false`.

`trivy.securityCheck`
: Comma-separated list of security issues to detect. Default is `vuln`.

`trivy.timeout`
: The duration to wait for scan completion. Default is `5m0s`.

`trivy.gitHubToken`
: The GitHub access token required for database downloads. Default is `undefined`.

`trivy.priorityClassName`
: The priority class for running the pod. Default is `undefined`.

`trivy.topologySpreadConstraints`
: Defines pod distribution constraints across failure domains. Default is `undefined`.

`trivy.initContainers`

List of init containers to run before the main container starts. Default is `[]`.

DATABASE PARAMETERS

`database.type`

The database type. Set to `external` when using an external database. Default is `internal`.

`database.internal.image.repository`

The repository for the database image. Default is `private-registry/harbor-db`.

`database.internal.image.tag`

The tag for the database image. Default is `2.11`.

`database.internal.password`

The password for the internal database. Default is `changeit`.

`database.internal.shmSizeLimit`

The shared memory size limit for PostgreSQL (typically 50% of the container memory limit). Default is `512Mi`.

`database.internal.resources`

The resources allocated for the database container. Default is `undefined`.

`database.internal.automountServiceAccountToken`

Controls whether the service account token is mounted. Default is `false`.

`database.internal.initContainer.migrator.resources`

The resources allocated for the database migrator init container. Default is `undefined`.

`database.internal.initContainer.permissions.resources`

The resources allocated for the database permissions init container. Default is `undefined`.

`database.internal.nodeSelector`

The node labels for pod assignment. Default is `{}`.

`database.internal.tolerations`

The tolerations for pod assignment. Default is `[]`.

`database.internal.affinity`

The node or pod affinity settings. Default is `{}`.

`database.internal.priorityClassName`

The priority class for running the pod. Default is `undefined`.

`database.internal.livenessProbe.timeoutSeconds`
: The timeout in seconds for the liveness probe (range: 1-5s). Default is `1`.

`database.internal.readinessProbe.timeoutSeconds`
: The timeout in seconds for the readiness probe (range: 1-5s). Default is `1`.

`database.internal.extrInitContainers`
: Additional init containers that run before the database container starts. Default is `[]`.

`database.external.host`
: The host name of the external database. Default is `192.168.0.1`.

`database.external.port`
: The port number of the external database. Default is `5432`.

`database.external.username`
: The username for the external database. Default is `user`.

`database.external.password`
: The password for the external database. Default is `password`.

`database.external.coreDatabase`
: The database name used by the core service. Default is `registry`.

`database.external.existingSecret`
: The existing secret containing the database password. The key must be `password`. Default is `""`.

`database.external.sslmode`
: The connection method for the external database. Options: `require`, `verify-full`, `verify-ca`, `disable`. Default is `disable`.

`database.maxIdleConns`
: The maximum number of idle connections in the pool (0 or less means no idle connections are retained). Default is `50`.

`database.maxOpenConns`
: The maximum number of open connections to the database (0 or less means unlimited). Default is `100`.

`database.podAnnotations`
: The annotations to add to the database pod. Default is `{}`.

`redis.type`
: The Redis deployment type. Set to `external` for external Redis. Default is `internal`.

`redis.internal.image.repository`
: The repository for the Redis image. Default is `private-registry/harbor-redis`.

`redis.internal.image.tag`
: The tag for the Redis image. Default is `7.2`.

`redis.internal.resources`
: The resources allocated for the Redis container. Default is `undefined`.

`redis.internal.automountServiceAccountToken`
: Controls whether the service account token is mounted. Default is `false`.

`redis.internal.nodeSelector`
: The node labels for pod assignment. Default is `{}`.

`redis.internal.tolerations`
: The tolerations for pod assignment. Default is `[]`.

`redis.internal.affinity`
: The node or pod affinity settings. Default is `{}`.

`redis.internal.priorityClassName`
: The priority class for running the Redis pod. Default is `undefined`.

`redis.internal.jobserviceDatabaseIndex`
: The database index for jobservice. Default is `1`.

`redis.internal.registryDatabaseIndex`
: The database index for registry. Default is `2`.

`redis.internal.trivyAdapterIndex`
: The database index for Trivy adapter. Default is `5`.

`redis.internal.harborDatabaseIndex`
: The database index for miscellaneous Harbor business logic. Default is `0`.

`redis.internal.cacheLayerDatabaseIndex`
: The database index for Harbor's cache layer. Default is `0`.

`redis.internal.initContainers`

The init containers that run before the Redis container starts. Default is `[]`.

`redis.external.addr`

The address of the external Redis instance. Default is `192.168.0.2:6379`.

`redis.external.sentinelMasterSet`

The name of the Redis Sentinel master set (if applicable). Default is `undefined`.

`redis.external.coreDatabaseIndex`

The database index for core. Default is `0`.

`redis.external.jobserviceDatabaseIndex`

The database index for jobservice. Default is `1`.

`redis.external.registryDatabaseIndex`

The database index for registry. Default is `2`.

`redis.external.trivyAdapterIndex`

The database index for Trivy adapter. Default is `5`.

`redis.external.harborDatabaseIndex`

The database index for miscellaneous Harbor business logic. Default is `0`.

`redis.external.cacheLayerDatabaseIndex`

The database index for Harbor's cache layer. Default is `0`.

`redis.external.username`

The username for external Redis authentication. Default is `undefined`.

`redis.external.password`

The password for external Redis authentication. Default is `undefined`.

`redis.external.existingSecret`

The existing secret containing the Redis password. The key must be `REDIS_PASSWORD`. Default is `""`.

`redis.podAnnotations`

The annotations to add to the Redis pod. Default is `{}`.

EXPORTER PARAMETERS

`exporter.replicas`

The number of replicas to run. Default is `1`.

`exporter.revisionHistoryLimit`

The revision history limit. Default is `10`.

`exporter.podAnnotations`

Annotations to add to the exporter pod. Default is `{}`.

`exporter.image.repository`

The repository for the exporter image. Default is `private-registry/harbor-exporter`.

`exporter.image.tag`

The tag for the exporter image. Default is `2.11`.

`exporter.nodeSelector`

Node labels for pod assignment. Default is `{}`.

`exporter.tolerations`

Tolerations for pod assignment. Default is `[]`.

`exporter.affinity`

Node or Pod affinities. Default is `{}`.

`exporter.topologySpreadConstraints`

Constraints that define how Pods spread across failure-domains like regions or availability zones. Default is `[]`.

`exporter.automountServiceAccountToken`

Controls whether to mount the serviceAccountToken. Default is `false`.

`exporter.cacheDuration`

The cache duration for information collected by the exporter. Default is `30`.

`exporter.cacheCleanInterval`

The cache clean interval for information collected by the exporter. Default is `14400`.

`exporter.priorityClassName`

The priority class to run the pod as. Default is `undefined`.

METRICS PARAMETERS

`metrics.enabled`

Enables Harbor metrics. Default is `false`.

`metrics.core.path`

The URL path for core metrics. Default is `/metrics`.

`metrics.core.port`

The port for core metrics. Default is `8001`.

`metrics.registry.path`

The URL path for registry metrics. Default is `/metrics`.

`metrics.registry.port`

The port for registry metrics. Default is `8001`.

`metrics.exporter.path`

The URL path for exporter metrics. Default is `/metrics`.

`metrics.exporter.port`

The port for exporter metrics. Default is `8001`.

`metrics.serviceMonitor.enabled`

Enables creation of a Prometheus ServiceMonitor (requirePrometheusus CRDs). Default is `false`.

`metrics.serviceMonitor.additionalLabels`

Additional labels to apply to the ServiceMonitor manifest. Default is `""`.

`metrics.serviceMonitor.interval`

The scrape interval for Harbor metrics. Default is `""`.

`metrics.serviceMonitor.metricRelabelings`

The relabeling rules for metrics before ingestion. Default is `[]`.

`metrics.serviceMonitor.relabelings`

The relabeling rules for metrics before scraping. Default is `[]`.

TRACE PARAMETERS

`trace.enabled`

Enables tracing functionality. Default is `false`.

`trace.provider`

The tracing provider (`jaeger` or `otel`). Jaeger version should be 1.26+. Default is `jaeger`.

`trace.sample_rate`

The sampling rate for trace data. `1` samples 100%, `0.5` samples 50%. Default is `1`.

`trace.namespace`

The namespace to differentiate different Harbor services.

`trace.attributes`

A key-value dictionary for user-defined attributes in trace provider initialization.

`trace.jaeger.endpoint`

The endpoint for Jaeger tracing. Default is `http://hostname:14268/api/traces`.

`trace.jaeger.username`

The username for Jaeger authentication.

`trace.jaeger.password`

The password for Jaeger authentication.

`trace.jaeger.agent_host`

The agent host for Jaeger.

`trace.jaeger.agent_port`

The agent port for Jaeger. Default is `6831`.

`trace.otel.endpoint`

The endpoint for OpenTelemetry tracing. Default is `hostname:4318`.

`trace.otel.url_path`

The URL path for OpenTelemetry. Default is `/v1/traces`.

`trace.otel.compression`

Enables compression for OpenTelemetry. Default is `false`.

`trace.otel.insecure`

Establishes an insecure connection for OpenTelemetry. Default is `true`.

`trace.otel.timeout`

The timeout in seconds for OpenTelemetry. Default is `10`.

CACHE PARAMETERS

`cache.enabled`

Enables the cache layer. Default is `false`.

`cache.expireHours`

The expiration time in hours for the cache layer. Default is `24`.

B Example of a Private Registry HA setup Helm chart

The following example values file illustrates parameters that are required for the minimal Private Registry HA setup.

```
expose:
  ingress:
    hosts:
      core: core.harbor.domain ❶

externalURL: https://core.harbor.domain ❷

portal:
  replicas: 2 ❸

core:
  replicas: 2 ❹

jobservice:
  replicas: 2 ❺

registry:
  replicas: 2 ❻

database:
  type: external
  external: ❼
    host: "192.168.0.1"
    port: "5432"
    username: "user"
    password: "password"
    coreDatabase: "registry"
    existingSecret: "" ❽
    sslmode: "disable" ❾

redis:
  type: external
  external: ❿
    addr: "192.168.0.2:6379" ⓫
    sentinelMasterSet: "" ⓬
    coreDatabaseIndex: "0" ⓭
    jobserviceDatabaseIndex: "1"
    registryDatabaseIndex: "2"
    trivyAdapterIndex: "5"
    harborDatabaseIndex: "6" ⓮
    cacheLayerDatabaseIndex: "7" ⓯
```

```
    username: "" 16
    password: ""
    existingSecret: "" 17

persistence:
  enabled: true 18
```

1. Core service host name in Ingress rule.
2. The external URL for the harbor-core service.
3. 4 5 6 Number of replicas to create. Specify two or more.
7. Fill the database connection details in the `external` section.
8. If using an existing secret, the value must be `password`.
9. Accepts one of the following values:

   disable
   : Do not use SSL.

   require
   : Always use SSL and skip verification.

   verify-ca
   : Always use SSL. Verify that the certificate presented by the server was signed by a trusted CA.

   verify-full
   : Always use SSL. Verify that the certificate presented by the server was signed by a trusted CA and the server host name matches the one in the certificate.
10. Fill the connection information in the `external` section.
11. Supports redis + sentinel.
    Address for redis is `<redis_host>:<redis_port>`.
    Address for redis + sentinel is `<sentinel1_host>:<sentinel1_port>,<sentinel2_host>:<sentinel2_port>`...
12. The name of the set of Valkey instances to monitor. It must be set to support redis + sentinel.
13. Must be `0` as the library that Harbor uses does not support configurations.
14. Optional. Defaults to `0` but can be configured to `6`.
15. Optional. Defaults to `0` but can be configured to `7`.
16. If empty, it is authenticated against the default user.

17 If used, the key must be <REDIS_PASSWORD>.

18 To store all the images, metadata and scans, ensure that the persistence-related settings (*Persistence parameters*) are properly configured.

C GNU Free Documentation License

C1 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or non-commercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

C2 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

C3 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or non-commercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

C4 3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

C5 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

1. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
2. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
3. State on the Title page the name of the publisher of the Modified Version, as the publisher.
4. Preserve all the copyright notices of the Document.

5. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

6. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

7. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

8. Include an unaltered copy of this License.

9. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

10. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

11. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

12. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

13. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

14. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

15. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

C6 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

C7 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

C8 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

C9 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all

Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

C10 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

C11 1. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See https://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

C12 ADDENDUM: How to use this License for your documents

```
Copyright (c) YEAR YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled "GNU
Free Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with… Texts."" line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.