# Container Roadmap

This page outlines the strategy for managing and evolving containerized packaging for COmanage. It includes the guiding principles for the work's direction and a description of the resources that support this work.

***This roadmap is continually updated. Please see above for the date of the last update.***

**On this Page**

## Rationale

1. Operating system-level (OS-level) virtualization facilitates efficient software deployment and IT operations (DevOps). The COmanage Community (Community) has requested that the COmanage Project (Project) deliver containers (images) for OS-level virtualization.
2. Open Container Initiative (OCI) compliant images that follow the OCI Image Format Specification deliver the highest level of interoperability.
3. Container best practices continuously evolve along with deployment and orchestration technologies.
4. Community members require container images that may both be directly deployed and serve as a base for building more customized containers.
5. Community deployments rely on multiple authentication strategies and implementations including (but not limited to) HTTP Basic Authentication, SAML and specifically the Shibboleth Service Provider (SP), OIDC and specifically mod_auth_openidc.
6. Community deployments integrate into and federate with many different scales and types of infrastructure ranging from isolated services using single authentication sources to services simultaneously federated with multiple higher education and resource identity federations.
7. The initial containerization work, funded by SWAMID/SUNET, was exploratory and the release cadence for container images was decoupled from the application release cadence. As such the initial decision was made to keep the container tooling in a separate version control repository. Having two distinct version control repositories, however, creates confusion for deployers.

   Now that the container work is stabilized and the release workflow integrated with the application release workflow, integrating the container tooling into the application version control repository will be straightforward and result in a single place for deployers to go looking for "source".
8. Version control of container documentation is easier using the Markdown markup language and a traditional version control repository rather than a Confluence wiki. As such the initial decision was made to document the containers using Markdown and to keep the Markdown source in the container version control repository.

   Having the container documentation separate from the application documentation, however, creates substantial confusion for deployers and diminishes the benefits of using Markdown for documentation.

## Plan

1. The COmanage Project (Project) will produce container images and related artifacts at the request of the Community. Other virtualization technologies beyond containers will be considered at the request of the Community.
2. The Project will produce OCI-compliant container images suitable for both direct deployment and for using as a base to build customized images.
3. The Project will monitor evolving container best practices (e.g., here, here, and here) and adapt as necessary to produce container images optimized for efficient DevOps with current deployment targets and for security.
4. For each Project application (Registry and Match), the Project will develop and maintain tooling to allow deployers to build the following container images:
   a. Images with HTTP Basic Authentication, including simple defaults to allow for quick deployments for application evaluation and exploration by people new to the applications.
   b. Images with HTTP Basic Authentication, the CakePHP DebugKit, and enabled debugging for developers.
   c. Images with the Shibboleth SP for authentication.
   d. Images with SimpleSAMLphp for authentication.
   e. Images with mod_auth_openidc for authentication.
   f. OpenLDAP slapd with additional schema that facilitates integration with Registry.
   g. Images for other services and tools that facilitate deployment of Registry as determined by the Project and solicitation by the Community.
5. Additionally the Project will begin the design of a container service stack and documentation that provides for tight integration between Registry and Match.
6. The Project will base its Registry and Match images on the official PHP Docker Image maintained by the Docker Community at https://hub.docker.com/_/php. The Project will continue to monitor the community commitment to keeping the PHP images up to date and for other image repositories that may provide a better base image.

7. The container images supporting federated authentication will not include specific configuration details for any identity federation but will support injecting such configuration both by building further images using the Project images as a base and dynamically at runtime.
8. The Project will migrate the existing Registry Docker repository served from the Internet2 Enterprise GitHub at https://github.internet2.edu/docker/comanage-registry-docker into the Registry GitHub repository. The migration is targeted for the first version of COmanage Registry based on the CakePHP version 4.x framework (currently code named Pupal Eclosion or PE).
9. The Project will migrate the existing Match Docker repository served from the Internet2 Enterprise GitHub at https://github.internet2.edu/COmanage/docker with the Match Internet2 Enterprise GitHub repository https://github.internet2.edu/COmanage/match.
10. The Project will migrate the existing container documentation (currently delivered using Markdown markup language) to and integrate with the application documentation served from the Internet2 Confluence Spaces wiki.
11. The COmanage Project will publish images to the following public repositories:
    a. Docker Hub in the COmanage Project organization: https://hub.docker.com/orgs/comanageproject
    b. Amazon AWS ECR public registry https://gallery.ecr.aws/cilogon/ managed by the CILogon project.
    c. GitHub Container registry https://ghcr.io/cilogon/ managed by the CILogon Project.
    d. Other public repositories as determined by the project.
12. The COmanage Project will begin an initiative to evolve images/containers so that they do not run as root (A first step will be containers that may start as root but where the processes drop privileges after initialization).
13. The COmanage Project will publish images to support the following architectures:
    a. amd64
    b. arm64v8
14. The COmanage Project will use a continuous delivery service to automate periodic (e.g., daily, weekly) production of images.

# Resources

1. The CILogon Project will provide an unrestricted contribution to the project to serve as Project Packaging Lead. As lead, they will ensure that all project contributions related to Packaging are in alignment with this roadmap, and will coordinate all project contributions related to Project Packaging.
2. Scott Koranda (CILogon) will serve as primary Lead. Terry Fleury (CILogon) also will contribute.
3. Other restricted and directed contributions may be made by these and/or other parties and will be coordinated with this Roadmap by the Packaging Lead (above).