OpenShift Release Types Guide

Table of Contents

Terminology

- Z-stream - A series of OpenShift versions for a particular MAJOR.MINOR. For example, '4.6.1, 4.6.2, 4.6.3, …" are all published as part of the 4.6 Z-stream.
- Release Payload - An immutable version of an OpenShift release. A release payload is a container image storing the exact details of all of the content necessary to install or update OpenShift to its specific version. Each OpenShift release has a physical representation on quay.io in the form of a release payload (e.g. quay.io/openshift-release-dev/ocp-release:4.11.18-x86_64 is the 4.11.18 release payload for the x86_64/amd64 CPU architecture).
- Public Release - A release payload which is available to users with a pull secret from https://console.redhat.com .
- GA Release - Generally Available Release. This means a release is publicly available and fully supported by Red Hat.
- OSUS - OpenShift Update Service (a.k.a Cincinnati). Provides upgrade recommendations to clusters to inform potential upgrade paths.
- EOL - End of Life. Applies to minor release versions(4.y). At EOL no new patch versions(4.y.z) of this minor release will be provided.
- Branch Cut - A moment in the OpenShift development lifecycle where feature development in the main/master branches in GitHub stop tracking 4.y.0 content and start tracking a future minor release 4.(y+1).0. Development continues on 4.y.0 in a dedicated branch, but the effort is focused on bug fixes and final productization.
- Release Controller - An information hub for OpenShift releases / CI testing data. Each CPU architecture has a release controller https://<cpu_arch>.ocp.releases.ci.openshift.org/ where cpu_arch can be one of (amd64, arm64, s390x, ppc64le) OR "multi" for heterogeneous architecture release payloads.

Background

The OpenShift software development lifecycle has several phases preceding the final release of software for customer use in their production environments. Certain phases are reflected in the availability of "Release Types". This guide explains each type and how they are expected to be used by both internal and external parties.

OpenShift Release Types can be recognized by their version, which is visible on OpenShift release controllers OR via "oc version" against an installed cluster. OpenShift release versions are formatted as follows: MAJOR.MINOR.PATCH[-addition-release-details].

For the remainder of this document x, y, and z will stand in respectively for the MAJOR, MINOR, and PATCH fields of the version. So for 4.12.1, x=4, y=12, and z=1.

Releases Types

Release Type	Naming Convention	Public	Release Controller Stream	Appropriate Use	Signed Release Payload	Testing	Available as an Upgrade from	OSUS Channels	Update Cadence
Standard / GA (The only release with an Advisory)	4.y.z	Yes	4-stable	Highest quality release and intended for customer use once available in the OSUS "fast" channel.	Red Hat Release Key	Full QE Testing Passed before availability in the "fast" channel.	Recent Standards, Recent y-1 Standards, and any recent Release Candidates.	All	Approximately weekly, for the two most recent GA Z-streams. Older Z-streams are updated at least every 6 weeks. General Availability until EOL
Release Candidate	4.y.0-rc.#	Yes	4-stable	Early customer access for staging environments and integration testing. Red Hat internal use for inter-op testing and business critical infrastructure.	Red Hat Release Key	Passed Release Gating Tests & Partial QE.	Recent Release Candidates, 4.y Engineering Candidates, and Recent y-1 Standards.	Candidate	Weekly Branch Cut until GA
Engineering Candidate	4.y.0-ec.#	Yes	4-dev-preview	Very early customer access for feature assessment and dev environments. Red Hat internal use for inter-op testing and non-business critical infrastructure.	Red Hat Beta2 Key	Passed Release Gating Tests, Approved by the Technical Release Team, & Partial QE.	Recent Engineering Candidates + Recent y-1 Standards	Candidate	Most three week sprints and based on the current release in feature development. Once we hit feature complete/branch-cut for a release, we start producing Release Candidates for that release, and Engineering Candidates become based on the next release.
Nightly (Internal)	4.y.0-0.nightly-[datestamp]	No	4.y.0-0.nightly **Periodically Garbage Collected**	Red Hat internal use for inter-op and continuous QE testing.	Release Controller Private Key	If "Accepted" by the Release Controller, it has passed Release Gating Tests.	N/A - Not registered with OSUS.	N/A	Frequent (on code change) New Release→EOL
CI (Internal)	4.y.0-0.ci-[datestamp]	No	4.y.0-0.ci **Periodically Garbage Collected**	CI subsystems and product engineering testing.	Release Controller Private Key	If "Accepted" by the Release Controller, it has passed Release Gating Tests.	N/A - Not registered with OSUS.	N/A	Frequent (on code change) New Release→EOL **Produced by non-production build systems and suitable only for CI.**
Custom Release Payload (a.k.a Hotfix)	4.y.z-assembly.art####	Yes	N/A - Never presented in the Release Controller.	Bespoke releases for customers to deliver a fix outside of the normal standard release cadence. Requires a support exception approved by OCP Director/BU.	Red Hat Beta2 Key	Testing negotiated as part of support exception and based on hotfix changes.	N/A - Not registered with OSUS.	N/A	On request.