Releases

This page is outdated and will be updated. Please see this page for our Release Principles for Apache CloudStack 4.6 and on.

Overview

All available releases can be found on the Apache CloudStack project's website, on the Downloads page. The release-specific child pages in this wiki are not intended to be the official documentation for any release, but are intended to be documents used by the Apache CloudStack community for collaboration and delivery of releases.

The remainder of this document provides an overview of the planning, scheduling and support processes that the community has adopted for the Apache CloudStack project.

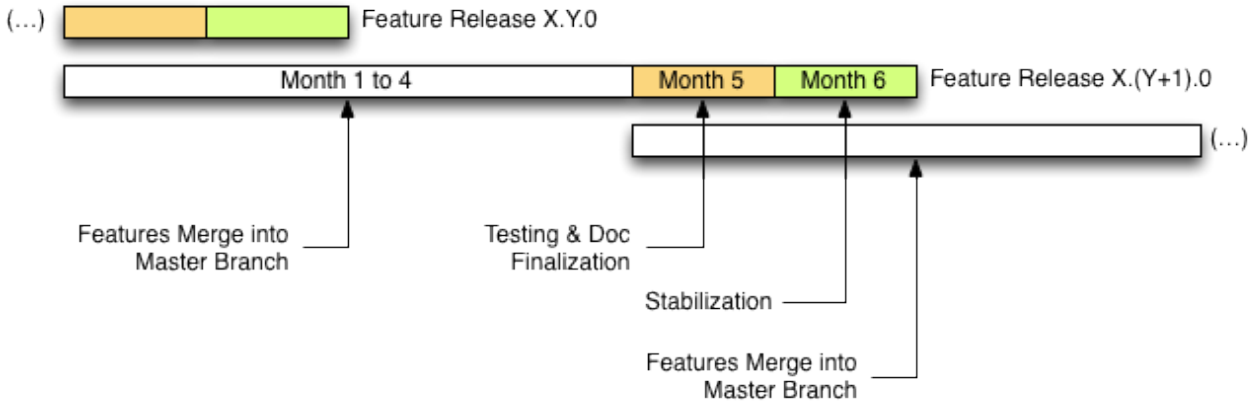
Release Planning

The Apache CloudStack community uses a time-based approach to it's feature releases, with as-needed (i.e.: less formal schedule) for incremental bug-fix releases in between these feature releases. Our goal is to have a consistent schedule for new features to be merged into the master branch, along with incremental bug-fix updates between each major feature release.

- Overview
- Release Planning
  - Feature Release Cycle
  - Bug-Fix Release Schedule
  - Release Branch Lifecycle
  - Versioning
  - Support Lifetime of a Release
- Other Resources

Feature Release Cycle

Feature releases are planned for every 4 months, with a 2 month overlapping development window, giving each feature 6 months to go through it's cycle.



Months one though four are designated for features to merge into the project's master branch. Typically, architectural change and high risk feature additions should be introduced within the first two months of that phase. Month five is dedicated to end to end testing of the merged features and changes, as well as documentation finalization. Month six is focused on wrapping up final translation updates, last minute blocker bug fixes and preparing / executing the formal release voting process.

Critical dates for each feature release cycle will be documented on that release's planning page, but a generalized schedule is as follows:

- Month 1, Day 1
  - Immediately after the previous feature release branch is created, master is open to new feature commits.
  - The release manager will issue a call for volunteers to work on unassigned New Feature and Improvement items
- Month 1 through 4
  - Feature development, documentation and testing
- Month 3, Day 1
  - Unassigned new features and improvements will be removed from the release in Jira (housekeeping)
- Month 4, Last Day
  - Feature Freeze
  - Release branch will be cut.

    - CI systems updated to include new release branch in builds.
- Month 5
    - Testing/Bug Fixes (testing against jenkins artifacts)
    - Documentation Finalization
- Month 5, Last Day
    - Docs Completion Target (except release notes and translations)
    - Release Branch moves to limited updates only (only commits allowed in would be release blockers fixes, translation updates, etc...)
- Month 6
    - Final Translation Development and Integration (should be ongoing, but focused effort)
    - Final regression testing / bug fixes / doc fixes
- Month 6, End of third week
    - Release Candidate is created, and project level VOTE is called

Voting on a release is a formal act within the ASF. For that reason, no specific release day can be scheduled in advance (only the start of the first round of voting). For more information about that process, see the Apache Release Policy.

Typically, the project will have one or more release votes, which last at least 72 hours each. The release manager has the right to cancel any release vote, if an issue is discovered that warrants a fix and restart of the voting process.

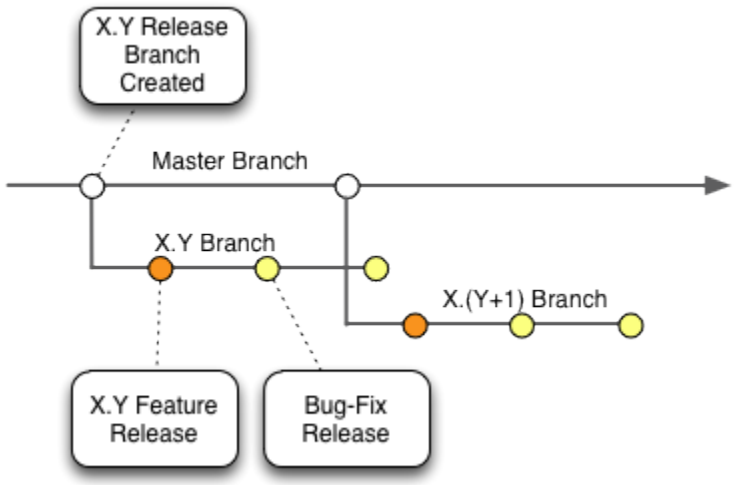
Bug-Fix Release Schedule

Bug fix (including security related bugs) are provided as incremental updates between scheduled feature releases. The project has not decided to do them on a specific schedule, but instead when there are enough appropriately important bug-fixes applied to the source release branch, a community member will proceed with creating a release. A single security vulnerability fix is typically considered enough to warrant a bug-fix release.

Release Branch Lifecycle

Similar to the schedule noted above, the project uses a branching strategy that reflects the overlapping nature of our work. The master branch is considered to be the development target for new features, while release branches are created for each feature release to support stabilization of that release. The project also considers feature release branches (which have had their feature release completed) to be a stable target from which bug-fix releases can be cut.

Fixes that are typically applied to both master and any active / relevant release branch.



Versioning

Starting with our 4.0.0-incubating release, Apache CloudStack makes use of Semantic Versioning for it's release numbering. One exception to this numbering scheme, is that the "-incubating" qualifier will be added to all release numbers until the project graduates from the Apache Incubator (this is specifically due to the rules stated in the Guide to Release Management During Incubation). Releases of CloudStack from before it's donation to the ASF did not follow Semantic Versioning.

For those that may not be familiar with Semantic Versioning, the number format is: X.Y.Z, where X is the major version, Y is the minor version, Z is the patch number. The community strives to ensure backward API compatibility within each major version (i.e.: code written against the CloudStack 4.0.0-incubating API should work with all future 4.y.z versions). The community may decide to increment the major version number in situations where underlying implementation details require a **cloud operator** to face significant challenges in upgrading from one version to the next. This should be rare situation.

In practice, feature releases will normally be an increment of the *minor* version number of the project. Feature releases that break backward compatibility will cause the *major* version number to be incremented. Bug fix releases will *never* increment anything except the patch number.

Support Lifetime of a Release

Due to the Apache CloudStack community's adoption of Semantic Versioning, the project has chosen to adopt a release support model as follows:

- The latest feature release of our active major version number will receive critical bug and security fixes until the next feature release with the same major version number.
- The latest feature release of our last major version number will receive critical bug and security fixes for a period of 12 months. This situation doesn't exist within the Apache CloudStack project yet, but will once the project releases a 5.0.0 feature release.

In practical terms, this means that someone running 4.0.0 should reasonable expect to see *at least* a 4.0.1 release to address any critical bugs found (and resolved) before 4.1.0 is released.

Releases of CloudStack from before it's donation to the ASF are not supported by the Apache CloudStack community.

Given the nature of this release support model, the community believes that providing an upgrade path from any CloudStack release to any later CloudStack release is a critical feature for our users. This includes upgrades from one major version number to the next. It should be expected that any exception to a smooth upgrade path is either a critical bug to be addressed, or will be documented very clearly (and it will cause a *major* version number increment).

Other Resources

Expand all Collapse all