

Roadmap

Hadoop Roadmap

This page describes Hadoop's release policies.

Release planning is primarily coordinated through [Hadoop's JIRA database](#). Use the "Roadmap" tab to see specific plans for upcoming releases.

For ideas about *what* you might contribute, please see the [ProjectSuggestions](#) and [HowToContribute](#) pages.

Release Numbering

Hadoop release numbers are of the form *major.minor.point*.

Major Releases

Major releases signify incompatible API changes. Upgrading between major releases will generally require changes to user code. We try to facilitate API upgrades by introducing new APIs in the prior major version, deprecating APIs that will be removed.

A major release primarily just removes APIs and features that were deprecated in the prior major release. Thus, to upgrade to a new major release, one should update ones code so that it compiles without deprecation warnings in the final minor release of the prior major cycle.

Across all releases with the same *major* version, user-applications work without the need for 'any' change. Thus every release with the same *major* version is both API and 'on the wire' compatible.

Major releases are made as needed, perhaps annually or even further apart.

Minor Releases

Minor releases are made regularly, every few months. Their APIs are back-compatible with prior minor releases, but might include new features, improvements and bug fixes. Also, across minor releases

Across all releases with the same *major* version, user-applications work without the need for 'any' change. Thus every release with the same *major* version is both API and 'on the wire' compatible.

Point Releases

Point releases are made to fix **blocker** bugs from an operational perspective. They do not introduce new features or make other improvements other than fixing bugs. They are made on-demand.

Across all releases with the same *major* version, user-applications work without the need for 'any' change. Thus every release with the same *major* version is both API and 'on the wire' compatible.

Note: The above rules do not apply until the 1.0 release. Prior to 1.0, minor releases follow the rules for major releases, except they are still made every few months.

Release Process

For major and minor releases, a branch date is selected for the release. The proposed branch date is set as the release date in Jira. The release date in Jira is then updated to the expected date that the release will be available.

On the branch date, a branch is created for the minor release cycle. After this, only patches for issues rated "blocker" are applied to the branch. No new features or other improvements are added to branches. All new features must be added to trunk prior to the branch date.

For more details on how releases are created, see [HowToRelease](#).

Hadoop 3.x Releases

Planned for hadoop-3.0.0

- HADOOP
 - Classpath isolation on by default [HADOOP-11656](#)
- HDFS
- YARN
- MAPREDUCE

hadoop-3.0.0-alpha1

- HADOOP

- Move to JDK8+
 - Shell script rewrite [HADOOP-9902](#)
 - Move default ports out of ephemeral range [HDFS-9427](#)
- HDFS
 - Removal of http in favor of webhdfs [HDFS-5570](#)
 - Support for more than two standby [NameNodes](#) [HDFS-6440](#)
 - Support for Erasure Codes in HDFS [HDFS-7285](#)
 - Intra-datanode balancer [HDFS-1312](#)
- YARN
 - YARN Timeline Service v.2 [YARN-2928](#)
- MAPREDUCE
 - Derive heap size or mapreduce.*.memory.mb automatically [MAPREDUCE-5785](#)

Hadoop 2.x Releases

hadoop-2.x

- HADOOP
 - Compatibility tools to catch backwards, forwards compatibility issues at patch submission, release times. Some of it is captured at YARN-3292. This also involves resurrecting jdifi (HADOOP-11776/YARN-3426/MAPREDUCE-6310) and/or investing in new tools.
 - Classpath isolation for downstream clients [HADOOP-11656](#)
- HDFS
- YARN
 - Early work for disk and network isolation in YARN: YARN-2139, YARN-2140
 - YARN Timeline Service Next generation: [YARN-2928](#).
- MAPREDUCE

hadoop-2.9

- HADOOP
- HDFS
- YARN
- MAPREDUCE

hadoop-2.8

- HADOOP
 - [HADOOP-11090](#) Support *both* JDK7 and JDK8 runtimes
- HDFS
 - [HDFS-6200](#) Create a separate jar for hdfs-client
 - [HDFS-8008](#) Support client-side back off when the datanodes are congested
 - [HDFS-8009](#) Signal congestion on the [DataNode](#)
- YARN
 - [YARN-4233](#) YARN Timeline Service v1.5
 - [YARN-3214](#) Supporting non-exclusive node-labels: Alpha feature
 - [YARN-1963](#) Support priorities across applications within the same queue: Alpha feature
 - [YARN-3611](#) Support Docker Containers In [LinuxContainerExecutor](#): Alpha feature
 - [YARN-1197](#) Support changing resources of an allocated container: Alpha feature

hadoop-2.7

- HADOOP
 - Move to JDK7+ [HADOOP-10530](#)
 - Support JDK8 in Hadoop [HADOOP-11090](#)
- HDFS
 - Removal of old Web UI [HDFS-6252](#)
- YARN
 - Support disk as a resource in YARN for scheduling and isolation [YARN-2139](#)
 - Dynamic Resource Configuration [YARN-291](#)
 - Support [NodeGroup](#) layer topology on YARN [YARN-18](#)
- MAPREDUCE

hadoop-2.6

- HADOOP
 - Credential provider [HADOOP-10607](#)
 - Key management server [HADOOP-10433](#)
- HDFS
 - Heterogeneous storage (Phase 2)
 - Support APIs for using storage tiers by the applications [HDFS-5682](#)
 - Memory as storage tier [HDFS-5851](#)
 - SSD storage tier [HDFS-7228](#)
 - Transparent encryption [HDFS-6134](#)
 - Archival storage [HDFS-6584](#)

- Hot swap storage volumes in Datanodes [HDFS-6740](#)
 - AES support for faster wire-encryption [HDFS-6606](#)
- YARN
 - Support for rolling upgrades in YARN [YARN-666](#)
 - [ResourceManager](#) HA Phase 2 [YARN-556](#)
 - [NodeManager](#) Restart [YARN-1336](#)
 - Support for long-running services on YARN [YARN-896](#)
 - YARN service registry [YARN-913](#)
 - Support for admin-specified labels in YARN [YARN-796](#)
 - YARN reservation-subsystem [YARN-1051](#)
 - Support for automatic, shared cache for YARN application artifacts [YARN-1492](#)
 - Support for Docker containers in YARN [YARN-1964](#)
- MAPREDUCE

Released: 2014-11-18

hadoop-2.5

- HDFS
 - Extended attributes [HDFS-2006](#)
- YARN
 - Security for the timeline server [YARN-1935](#)
 - RM writable Webservices [YARN-1695](#)
- MAPREDUCE

Released: 2014-08-11

hadoop-2.4

- HDFS
 - Use protobuf for fsimage [HDFS-5698](#)
 - ACLs for HDFS [HDFS-4685](#)
 - Support for HDFS Rolling upgrades [HDFS-5535](#)
 - Bug fixes
- YARN
 - Application History Server [YARN-321](#)
 - Resource Manager Automatic Failover [YARN-149](#)
 - Application Timeline Server [YARN-1530](#)
- MAPREDUCE
 - Bug fixes

Released: 2014-04-07

hadoop-2.3

- HDFS
 - Enable support for heterogeneous storages in HDFS (phase 1) [HDFS-2832](#)
 - HDFS in-memory caching [HDFS-4949](#)
- YARN
- MAPREDUCE
 - Deploy MR through HDFS & Distributed Cache [MAPREDUCE-4421](#)
- Overall
 - Important bug fixes and improvements since hadoop-2.2.

Released: 2014-02-20

hadoop-2.2

Released: 2013-10-13

- HDFS
 - High Availability for HDFS
 - HDFS Federation
 - HDFS Snapshots
 - NFSv3 access to data in HDFS
- YARN
 - Stable
 - API Stability
- Map-Reduce
 - Binary Compatibility for [MapReduce](#) applications between Hadoop v1 and Hadoop v2 to ease migration
- Overall
 - Stable
 - Performance
 - Support for running Hadoop on Microsoft Windows
 - Integration testing for the entire Apache Hadoop ecosystem at the ASF.

Hadoop 1.x - Sustaining Releases

The current sustaining branch is branch-1. It was branched from branch-0.20 with many features including security integrated in.

Because 0.21 had already been released, we added a new level to the release numbering. Thus version numbers on this branch look like 0.20.X.Y where X is > 200 and denotes the minor release from the branch. The Y denotes the point release that is intended for critical bug fixes to a previous sustaining release. The 0.205.0.1 release became the Hadoop 1.0.0 release, and we are now back to a three part (major.minor.point) version scheme. Changes for the next point releases of the sustaining branch need to be merged from branch-1 to branch-1.0 (or branch-1.1 etc). See the [Hadoop Releases Page](#) to see the full list of sustaining releases.

The Release Manager for a sustaining release should announce the code freeze date as far in advance as possible, on the general list. Prior to that date, anyone interested in contributing to the release submits a patch to branch-1, and adds the sustaining release number to "Target Version/s" in the Jira. Only functionality already committed to trunk should be submitted to a sustaining release. (The exception is if the functionality is not applicable to trunk.) The Release Manager is fully responsible for release content and timelines.

After the code freeze date, the Release Manager will generate the release candidate and call a release vote in the usual way. After that point, only patches for issues rated "blocker" may be added to the release.