AAA: Nitrogen: Release Plan

Contents

Introduction

Release Deliverables

Name	Description
n/a	n/a

Release Milestones

- **Offset: 0**

Milestone	Offset 2 Date [2]	Events
M0/M1	6/21 /2017	**Nitrogen Simultaneous Release Open** 1. Contact Freeze • Projects must have declared intent to participate in Simultaneous Release • Projects must have elected their Project Leads and specify a Test Contact • Participating Projects must have published a candidate Release Plan for public comment (Release Plan Template) 2. *Note: the date for M0 will be at least one day after the TSC approves the Nitrogen release plan.*
last call for project proposals	6/28 /2017	1. This is the latest date a project proposal can be sent to the project-proposals list and still have the required two week public comment period before its project creation review at the last TSC meeting before the M2/M3/M4 milestone. Project proposals submitted after this date will not be able to become formal projects by M2/M3/M4 and thus will not be able to participate in the Nitrogen release.[3]

M2/M3/M4	7/14 /2017	**API Freeze: See more information in the definition above.** *Note that the release plan includes details about negotiating inter-project dependencies, expectations, and incompatibilities.* 1. Plan Freeze • Participating Projects must have declared their final Release Plan with all sections fully completed. • Projects that need extra configuration or resources other than those availble in the OpenDaylight CI infrastructure must have opened helpdesk tickets to add them. • Project Checklist completed (for *all* projects, not just new ones). • Projects may apply for a system test waiver if they think they have top-level features not requiring system test or covered by other top-level features test. • Projects must specify whether they plan to use OpenDaylight CI infrastructure for system test. It is recommended to use the OpenDaylight CI infrastructure unless there is some HW or SW resource that cannot be installed there. Projects running system test in external Labs are required to report system test results in a timely fashion after release creations, e.g., weekly, RC, and formal releases. • Project must get acknowledged from all projects that it depends on. 2. Feature/Functionality Freeze • Final list of externally consumable APIs defined and documented ◦ Projects must state for each TENTATIVE API they have (if any) whether they are formally planning to deliver it. ▪ If so, it should be noted that it will be delivered. ▪ If not projects requesting the API must be informed so that they can take corrective actions. ◦ Externally consumable APIs are available at beta-quality • All inter-project dependencies are resolved (all project functionality is declared as either "In" or "Out" of this release) • Karaf Features defined ◦ Instructions can be found in the Karaf:Step by Step Guide ▪ Each feature should be tested in every appropriate jenkins job (at least -verify, -merge, and -integration) using the "SingleFeatureTest" as defined in the Karaf step-by-step guide ◦ Any feature repositories containing features intended for release must be added to the main features.xml file in the integration git repository as explained in the Karaf step-by-step guide ▪ Projects must have a distribution job to verify changes in code do not impact the integration distribution (this will be automatically setup by the releng/builder group). ◦ Features that are intended to be "top-level", "user-facing" and/or "stable" must be called out in the milestone readout. These features will have additional requirements: ▪ Each "top-level" feature must have a developer guide section (See the documentation requirements section above) and a system test (See the integration and test requirements section above). ▪ Each "user-facing" feature must have a user guide section (See the documentation requirements section above) ▪ Each "stable" feature must meet the requirements explained in the definitions section above. ◦ Changing the name of a Karaf feature or removing a Karaf feature should be handled via an API freeze waiver after this point • Documentation Started ◦ Identified the kinds of documentation to be provided, created AsciiDoc files for them with outlines, and committed those files in an appropriate location. (See the documentation requirements section above for more details.) • Feature Test Started ◦ Instructions can be found in the System Test Step by Step Guide. ◦ Projects must have filled out a basic system test plan template for each top-level feature (karaf and not karaf). Stable features have additional requirements for functionality, cluster, scalability, performance, longevity /stability. 3. API Freeze: See more information in the definition above. • Documentation: ◦ Project readouts MUST include a word count of each relevant .adoc file with a goal of draft documentation done. ◦ Projects must have a maven site with automatically generated Javadoc per instructions from the RelEng /builder project • Projects are encouraged to meet the requirements to be included in maven central ◦ Project readout MUST include whether or not this was accomplished • Feature Test Continues ◦ Participating projects Projects must have all extra SW configuration and resources required for system test installed in the OpenDaylight CI[4]. More information in How To Install SW in CI.
M5	8/14 /2017	**Code Freeze (bug fixes only from here as defined above)** 1. Stability branch, i.e., stable/nitrogen, must be cut and local project versions bumped on master to avoid overwriting Nitrogen SNAPSHOTS • Follow steps 1–4 from the instructions on cutting stability branches • *Note:* Branch cutting will occur sometime between offset 0 M5 and offset 2 M5 and may be either staggered by offsets or done all at once. See TSC meeting minutes from 7/9/2015 item 5.ac. 2. String Freeze (all externally visible strings frozen to allow for translation & documentation) 3. Documentation Complete: Only editing and and enhancing should take place after this point. 4. Feature Test Complete • Stable features should fulfill quality requirements listed in definitions section • Projects must run at least one basic automated system test job for each top-level feature and several automated system test jobs including functionality, cluster, scalability, performance, longevity/stability for each stable feature[4].

RC0	N/A	1. The build for RC1 will start at 23:59:59 UTC • At the start of the build for RC0, all stable/nitrogen branches will be locked and only release engineering staff will be able to merge patches and will only do so for patches that fix blocking bugs. 2. During the RC process, regular, e.g., daily, IRC meetings will take place to identify and address issues 3. During the RC process, blocking bugs will be tracked in bugzilla and a common spreadsheet
RC1	N/A	1. The build for RC1 will start at 23:59:59 UTC 2. All stable/nitrogen branches will remain locked and only release engineering staff will be able to merge patches and will only do so for patches that fix blocking bugs. 3. During the RC process, regular, e.g., daily, IRC meetings will take place to identify and address issues 4. During the RC process, blocking bugs will be tracked in bugzilla and a common spreadsheet
RC2	N/A	1. The build for RC2 will start at 23:59:59 UTC 2. All stable/nitrogen branches will remain locked and only release engineering staff will be able to merge patches and will only do so for patches that fix blocking bugs. 3. During the RC process, regular, e.g., daily, IRC meetings will take place to identify and address issues 4. During the RC process, blocking bugs will be tracked in bugzilla and a common spreadsheet
RC3	N/A	1. Participating Projects must hold their Release Reviews, including User Facing Documentation. • The release review should be based on the Sample Release Review and should point to release notes based on Sample Release Notes. 2. The version suffix for RC3 will be -Nitrogen instead of -Nitrogen-RCX or -Nitrogen-RCX-<timestamp> so that the build can be released if it is found to be free of blocking bugs. 3. The build for RC3 will start at 23:59:59 UTC 4. All stable/nitrogen branches will remain locked and only release engineering staff will be able to merge patches and will only do so for patches that fix blocking bugs. 5. During the RC process, regular, e.g., daily, IRC meetings will take place to identify and address issues 6. During the RC process, blocking bugs will be tracked in bugzilla and a common spreadsheet

Formal Nitrogen Release	N/A	1. Formal Nitrogen Release
		  • *NOTE: The build to produce the formal release artifacts is likely to occur before 5/25/2016.*
		2. After the release, except for projects that have opted-out, the release engineering staff will apply the release patch to the stable/nitrogen branch and bump versions.
		  • *Note:* Any patches merged to stable/nitrogen after the auto-release build that produces the formal release artifacts, but before the release patch and version bumps are applied will have to be reverted and re-applied after the release and version bump patches. This shouldn't happen in Nitrogen as the stable/nitrogen branches will have been locked since RC0.

SR1 (Service Release 1 aka Nitrogen.1)	N/A	1. First Service Release for Nitrogen. NOTE: This date is provisional, but will not move earlier. Please note, event based Updates (security/critical bugs) are distinct and may occur at any point.
		  • To allow time for testing, a release candidate will be built before the service release and projects are expected to not merge patches except for blocking bugs between that time and the actual service release.
		  • Blocking bugs will be tracked via bugzilla and a spreadsheet.
		2. After the release, projects MUST apply the release patch to the stable/nitrogen branch and bump versions. Unless a project opts out, this will be done automatically by the release team after the release.
		  • *Note:* Any patches merged to stable/nitrogen after the auto-release build that produces the formal release artifacts, but before the release patch and version bumps are applied will have to be reverted and re-applied after the release and version bump patches.

SR2 (Service Release 2 aka Nitrogen.2)	N/A	1. Second Service Release for Nitrogen. NOTE: This date is provisional, but will not move earlier. Please note, event based Updates (security/critical bugs) are distinct and may occur at any point.
		  • To allow time for testing, a release candidate will be built before the service release and projects are expected to not merge patches except for blocking bugs between that time and the actual service release.
		  • Blocking bugs will be tracked via bugzilla and a spreadsheet.
		2. After the release, projects MUST apply the release patch to the stable/nitrogen branch and bump versions. Unless a project opts out, this will be done automatically by the release team after the release.
		  • *Note:* Any patches merged to stable/nitrogen after the auto-release build that produces the formal release artifacts, but before the release patch and version bumps are applied will have to be reverted and re-applied after the release and version bump patches.

SR3 (Service Release 3 aka Nitrogen.3)	N/A	1. Third Service Release for Nitrogen. NOTE: This date is provisional, but will not move earlier. Please note, event based Updates (security/critical bugs) are distinct and may occur at any point. • To allow time for testing, a release candidate will be built before the service release and projects are expected to not merge patches except for blocking bugs between that time and the actual service release. • Blocking bugs will be tracked via bugzilla and a spreadsheet. 2. After the release, projects MUST apply the release patch to the stable/nitrogen branch and bump versions. Unless a project opts out, this will be done automatically by the release team after the release. • *Note:* Any patches merged to stable/nitrogen after the auto-release build that produces the formal release artifacts, but before the release patch and version bumps are applied will have to be reverted and re-applied after the release and version bump patches.
SR4 (Service Release 4 aka Nitrogen.4)	N/A	1. Fourth Service Release for Nitrogen. NOTE: This date is provisional, but will not move earlier. Please note, event based Updates (security/critical bugs) are distinct and may occur at any point. • To allow time for testing, a release candidate will be built before the service release and projects are expected to not merge patches except for blocking bugs between that time and the actual service release. • Blocking bugs will be tracked via bugzilla and a spreadsheet. 2. After the release, projects MUST apply the release patch to the stable/nitrogen branch and bump versions. Unless a project opts out, this will be done automatically by the release team after the release. • *Note:* Any patches merged to stable/nitrogen after the auto-release build that produces the formal release artifacts, but before the release patch and version bumps are applied will have to be reverted and re-applied after the release and version bump patches.

Externally Consumable APIs

- AAAService.java
- EncryptionService.java
- aaa.yang

Expected Dependencies on Other Projects

- odlparent, controller, yangtools

Expected Incompatibilities with Other Projects

None

Compatibility with Previous Releases

None

Themes and Priorities

- Karaf4
- Blueprint only Activation
- Fewer Bundles/Greater code organization
- MDSAL based realm

Requests from Other Projects

None

Test Tools Requirements

None

Other

N/A