**Gambia**

**Important Links**
- Gantt Chart (updated Aug 15, 2018)
- Projects Intending to Participate in the Gambia Release
- Summary of Release Plans for Gambia
- Gambia Scenario Status
- Milestone Compliance for Gambia (Traditional release process)
- Documentation Compliance for Gambia
- Milestone Exception Requests for Gambia
- Tagging Instructions for Gambia

**Introduction**

This is the Release Plan for the “Gambia” release of OPNFV.

**Definitions**

**Scenario**

OPNFV uses **scenarios** to deploy and test features that are developed by OPNFV projects, or to validate installers, test frameworks, or CI infrastructure.

A scenario is a combination of:

- A collection of upstream components (e.g., OpenStack, OpenDaylight, etc.)
- An installer (i.e., Apex, Compass, Daisy, Fuel, Joid)
- A configuration

Progress in the release is partially measured by completion of scenario documentation (MS1), feature implementation (MS5), and test case implementation (MS6).

Note that not all OPNFV projects use scenarios. For example, test frameworks often do not use specific scenarios.

Scenario owners must list their scenarios in the Gambia Scenario Status page.

**Scenario Owner**

A scenario owner is the individual that takes responsibility for ensuring that a scenario meets all milestone requirements and is successfully released, per the release schedule. Every scenario must have a scenario owner. Note that the scenario owner is often the same person as the PTL for the feature that is deployed by the scenario.

The scenario owner works with installer owners, test framework owners, and feature project teams to setup Jenkins jobs, fix build issues, enable testing, and resolve bugs and document the scenario. Note that it is NOT the responsibility of installer teams or test framework teams to integrate scenarios, fix build issues, or fix bugs. These teams may provide support for these activities when requested by the scenario owner, however, it is ultimately up to the scenario owner to ensure that the scenario is ready for release.

**Requirements for Participation**

**Planning**

- Projects must submit a proposal and be approved by the TSC before they may join a release.
- Projects must **express their intent to participate** in the release between MS0 and MS1. This is true, regardless of whether a project has participated in a past release.
  - Note that infrastructure projects, including RELENG, OPNFVDOCS and Pharos are presumed to be participating and are not required to submit intent-to-participate.
- Projects must develop and post a release plan by MS1. The content and format of the release plan is the decision of the PTL and project team. However, fell free to use this **suggested template**, if you’d like.

**OpenStack Version**
Prior to each release cycle the TSC approves a milestone schedule and the version of OpenStack that will be used by projects and scenarios that have OpenStack dependence (most). For the Gambia release, the TSC approved the selection of OpenStack Queens. Therefore, all OPNFV project participating in the release will use this version of OpenStack.

Milestone Compliance

- Projects must adhere to the milestone requirements, according to the schedule approved by the TSC, unless they have an approved milestone exception.
- Milestone compliance will be evaluated by the Release Manager and results will be published on the Milestone Compliance page.

Continuous Integration

- Projects delivering software for an OPNFV release must be integrated with OPNFV CI, including builds (MS5) and automated testing (MS6).
- Build and test results must be accessible to any user with access to the OPNFV Jenkins dashboard.

Issue Tracking

- OPNFV projects must track issues, using their own project, within the OPNFV JIRA project.
  - Projects should use epic, new feature, story, task, or improvement issue types to represent their plans for the release.
  - Projects should use the bug issue type for design problems, build failures, software failures, test failures, performance problems, or programming errors.
- Issues must be assigned to a release using the "fix version" field.
  - DO NOT leave the "fix version" field at "None". This creates confusion about the workload for a particular release, since it's unclear which release the project intends to work on a particular release. If you are unclear about when an issue will be resolved, then assign the issue to a valid future release, or to the release named "Future Release".
  - DO NOT make up your own version names. Valid version names have already been added to the projects. If your JIRA project does not contain these version names, you may add them yourself, or contact the release manager.
  - Valid release names for the Gambia release, are as follows:
    - 7.0.0, 7.1.0, 7.2.0
    - Future Release
  - Note: use this string if you don't anticipate resolving the issue in the current release. However, this means that you should periodically review issues assigned to "Future Release" and reassign them to a specific release.
- In order to support the use of JIRA for program management and for data collection (e.g., Bitergia), projects must do housekeeping on their JIRA project.
  - Keep the status updated.
  - Close issues that have been verified resolved, or that are no longer relevant.
  - Update the "fix version" field for issues that you plan to defer to a future release. We should not have unresolved issues that are assigned to past releases.
  - Update the "Assignee" field as people leave and join the project.

Documentation

Documentation compliance is checked at MS6 (preliminary) and MS9 (final). Projects indicate their compliance by completing the documentation compliance tables for MS6 and MS9.

Content

Projects are free to determine what documentation is appropriate for their project. However, the following documents are suggested for various project types:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Test</td>
<td>![ ]</td>
<td>![ ]</td>
<td>![ ]</td>
<td>![ ]</td>
<td>![ ]</td>
<td>![ ]</td>
</tr>
</tbody>
</table>

Format and Structure
Release Requirements

Scenarios

1. Meets the requirements of all applicable release milestones, especially MS2, MS5, MS6, and MS9, unless there is an approved milestone exception.
2. CI deploy job has run at least once during the last week of the release and at least three times over the preceding two weeks.
3. Most recent deploy job has passed.
4. Majority of deploy jobs over the two weeks preceding the release have passed.
5. Functest and/or Yardstick data is available for all builds over the past two weeks where deploy was successful.
6. Scenario owner asserts intent-to-release on scenario status page.
7. Known issues documented in JIRA and published in documentation for any failing tests.

Installer Projects

1. Meets the requirements of all applicable milestones, especially MS3 and MS9.

Tool and Test Framework Projects

1. Meets the requirements of all applicable milestones, especially MS4 and MS9.
2. Passes self-validation tests.

Schedule

Milestone Gantt Chart

The Gantt chart view (see "Gantt Chart" in Important Links) is useful for seeing an overview of the schedule, including related events (e.g. upstream component releases, community events, plugfests, etc.).

Milestone Schedule and Events

- The deadline time for all milestones is 5 p.m. Pacific Time, unless otherwise stated.
- Color Key:
  - Red - feature projects only
  - Yellow - requirement for installer projects only
  - Blue - requirement for test framework projects only
  - Green - scenario owners only

<table>
<thead>
<tr>
<th>Milestone</th>
<th>Date</th>
<th>Events</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS0</td>
<td>4/1</td>
<td>1. First day that intent-to-participate notifications will be accepted. a. Note that infrastructure projects, including RELENG, OPNFVDOCS and Pharos are presumed to be participating and are not required to submit intent-to-participate.</td>
</tr>
<tr>
<td></td>
<td>3/18</td>
<td></td>
</tr>
<tr>
<td>MS1</td>
<td>5/1</td>
<td>1. Last day that intent-to-participate notifications will be accepted. 2. Complete release plan and fill out release plan summary. 3. Add scenarios to scenario status page, if applicable.</td>
</tr>
<tr>
<td></td>
<td>1/18</td>
<td></td>
</tr>
</tbody>
</table>
1. Add test cases to the test case database.

1. Successfully deploy scenario "os-nosdn-nofeature-noha" in OPNFV CI using the version of OpenStack designated for the release.
2. Run Health Check (not required to pass, yet).

1. Functest supports Health Check.

1. Pass Health Check for scenario "os-nosdn-nofeature-ha".
2. Successfully deploy and pass health check for an sdn/nofeature scenario (e.g. "os-odl-nofeature-noha").
3. OpenDaylight version locked down for all installers

1. Applies to Functest, Yardstick, VSPerf, StorPerf, and Bottlenecks.
2. Test frameworks support OpenStack version planned for the release, if applicable.
3. Tests frameworks have completed major changes planned for the release
4. Test frameworks have unit tests that run in CI and that are passing
5. Feature projects can successfully develop and execute test cases in OPNFV CI, using the test frameworks

1. Scenario integration completed
   a. All scenarios have deploy jobs in OPNFV CI
2. Scenario status page is locked. No new scenarios may be added. However, scenario owners may request the removal of a scenario.

1. Test case implementation completed.
2. Preliminary documentation completed.

1. Stable branch window closes. Stable branch creation process.

1. Completion of formal testing.

1. Completion of final documentation.
   a. Complete [documentation compliance](#) table for MS9.

1. JIRA Cleanup
   a. All issues assigned to the current release should be:
      i. Closed
      ii. Reassigned to the next release, or other future release, using the "fix version" field

1. Initial technical release
2. Project (PTL or other team member) tags own repository (Git Tagging Instructions for Gambia).
   a. Any projects still in the release that have not been tagged by the project, will be tagged by Aric Gardner.
3. Artifact delivery.
1. First point release
2. Project (PTL or other team member) tags own repository.
   a. Any projects still in the release that have not been tagged by the project, will be tagged by Aric Gardner.
3. Artifact delivery

1. Second point release
2. Project (PTL or other team member) tags own repository.
   a. Any projects still in the release that have not been tagged by the project, will be tagged by Aric Gardner.
3. Artifact delivery
4. CI resources reassigned to Gambia release.

Artifact Delivery Process

On the technical release day:

1. Installer teams complete release artifacts according to the following schedule on the day of the technical release (MS11, 12, 13):
   a. 5 p.m. (China Standard Time) - Compass, Daisy
   b. 3 p.m. (EST) - Apex, Joid, Fuel@x86, Fuel@aarch64
2. Installer PTLs send email to [cc], with pointers to release artifacts.
   Aric Gardner
   David McBride
3. LF move artifacts to the appropriate directories and create and verify artifact links for use on the download page.

Within 2 - 3 business days after the technical release day:

1. LF will prepare the download page.
2. LF will prepare and disseminate press release.
3. The download page will go live.

Milestone Compliance Evaluation

Project PTLs should update the specified wiki forms with their compliance evidence as described below. If a wiki page is not specified, please send compliance evidence via email to David McBride.

<table>
<thead>
<tr>
<th>Milestone</th>
<th>Compliance Evidence</th>
</tr>
</thead>
</table>
| MS1       | 1. intent-to-participate notification received.  
           | 2. Release plan and release plan summary completed.  
           | 3. Scenarios added to scenario status page, if applicable. |
| MS2       | 1. URL with test case listing from test case DB received (example). |
| MS3.0     | 1. Each installer will submit a link to a Jenkins deploy job for os-nosdn-nofeature-ha.  
           | a. The job does not have to be passing at this point, but it must be enabled /active/running in OPNFV CI. |
| MS3.1     | 1. Each installer will submit a link to the results of a Jenkins job that shows Health Check passing for os-nosdn-nofeature-ha.  
           | 2. Each installer will submit a link to the results of a Jenkins job that shows Health Check passing for os-<sdn>-nofeature-ha. |
| MS4       | 1. Each test framework team will submit a link to the most recent output of the Jenkins job that runs the frameworks unit tests. |
| MS5       | 1. Each scenario owner will submit a link to the results of a Jenkins job, that shows Health Check passing for the scenario, for each scenario for which they are the owner. |
| MS6  | 1. Each scenario owner will submit a link to the results of a Jenkins job that shows all implemented test cases running, for each scenario for which they are the owner.  
   2. Complete documentation compliance page for preliminary documentation. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MS9</td>
<td>1. Complete documentation compliance page for final documentation.</td>
</tr>
</tbody>
</table>