

Brahmaputra

Brahmaputra release page

Need to add pages around release planning, projects involved, infrastructure, reporting & share relevant information here.

- [Brahmaputra release plan](#)
 - [Brahmaputra 2.0 stable release plan](#)
 - [Brahmaputra 3.0 stable release plan](#)
- [Brahmaputra daily status updates](#)
- [Daily release scrum meeting for Brahmaputra - agendas and minutes](#)
- [Test-scenarios for Brahmaputra](#)
- [Schedules of upstream projects relevant for Brahmaputra](#)
- [Discussion on release artifacts for Brahmaputra](#)
- [FuncTest R2 Brahmaputra wiki](#)
- [Latest results of Brahmaputra builds](#)
- [Brahmaputra Retrospective](#)

Overview

The Brahmaputra release will consist of a common deployment which may be extended by feature and development projects.

The common deployment of OPNFV components and functionality will be agreed in the genesis and test projects. All deployment tools participating in Brahmaputra are expected to provide a repeatable deployment of the common deployment that passes the associated test cases. The common deployment will be deployable on all Pharos labs, provide virtualised deployment capability and be fully validated by the testing projects. The genesis common deployment must be deployable by an install script which may derive from a single image or by the compilation of stable upstream components. Projects that are working on feature development for the Brahmaputra project are able to extend the common deployment with additional components and use cases. A feature project will need to ensure the components required are deployable by at least one deploy tool as part of the OPNFV CI pipeline. The feature project will be required to ensure the use cases proposed to be part of the Brahmaputra release are able to be verified by the testing projects.

All deployment options should be repeatable and able to be cleaned when the deployment is to be removed. All test cases and test suites must be repeatable.

Any needed exemptions to the above guidelines may be raised to the TSC for discussion. It is envisioned that any exemptions raised after code freeze will be extremely difficult to accommodate.

FAQ

How are different features/functions enabled for a Brahmaputra installation?

From the Brahmaputra release onwards, a user can choose the features to be installed for a particular installation. Feature selection and the associated configuration of the features is done via a set of configuration files which are common for all installers ("install.yaml" – for general configuration and feature selection, "feature_x.yaml" – for configuration specific to a "feature_x", like for example a SDN-controller like OpenDaylight). Note that not all features /functions are install-time features. Installers only take care of install-time features. Note that installers only install a particular feature or component, i.e. they don't take any responsibility for the appropriate operation of a feature, component, or function.

Several projects require specific components installed. How are their needs being met?

Projects which are interested in a particular combination of features and functionality are expected to supply the configuration files (i.e. the specific versions of "install.yaml" and "features.yaml") they require to the CI/CD (Octopus) and Release engineering (Releng) projects. In addition, the projects are to supply a set of system tests (typically as part of one of the testing projects like FuncTest, Yardstick, etc.) to verify their functionality, leveraging the particular combination installed – along with a description of what "successful testing" of the feature means. Note that as part of a test, additional features and/or configuration to the installer-supplied setup may need to be added ("post-install"). Using OpenDaylight as an example: OpenDaylight might only come configured with a set of base features by the installation process. Additional features that specific project testing requires would be installed by the test itself, i.e. through feature:install in karaf. In summary, a deployment can be phased into 3 steps - typically followed by testing, which would be the 4th step:

- *Hardware configuration* (so that an installer can execute successfully): Pharos system description and associated hardware configuration scripts supply for this.
- *Base system installation*: One of the deployment tools ("installers") supplies for this. If a project has a specific feature/component that needs to be installed as part of base system setup, the project needs to engage with the installer projects to get its feature installed/configured. Features and functions of the OPNFV platform fall into this category. See also: [Is my feature an install-time or post-install-time feature?](#)
- *Post base-install feature installation/configuration*: Per the above, several projects might have the need to perform project specific installation and /or configuration (often required by test projects).
- *Test*: Tests (leveraging one of the test frameworks in OPNFV) are carried out post any installation (base, or post-base install).

What is the role of testing and test-documentation for the Brahmaputra release?

Test projects (i.e. Functest, Yardstick, etc.) supply a set of tests for Brahmaputra. Brahmaputra testing adds to the existing test suites available in the OPNFV Arno release. Tests which are to be run on an installation are defined by a set of test-configuration scripts. Test configuration scripts define which test framework is used (e.g. tempest, robot, yardstick, ...), which tests are executed, how these tests are configured, and eventually even where the tests are to be executed (e.g. in case a certain hardware dependency is given). Tests are automatically executed as part of the CI/CD pipeline. Test results along with the test-system definition and test descriptions are archived in a test-database which documents all test runs. Projects can add project-specific tests to the existing set of tests supplied by the testing projects. Those tests could be run exclusively for a particular deployment/installation (e.g. specific L3VPN tests would only be run on an installation which has L3VPN features installed and configured). The configuration of these tests adds to the above described test-configuration.

What constitutes the Brahmaputra release?

A release is constituted through a set of deployments (defined by the configuration files), the associated tests (defined by the test configuration) and the corresponding test results. Note that the set of combinations tested follows interest: For a release not all possible feature/component and deployment configurations will be tested. Testing will follow the interest (and associated invested effort) of the projects participating, rather than try to test any possible combination.

What does it mean for a feature to be delivered by the Genesis project?

A feature or functionality delivered by the Genesis project will be supported by all installers participating in Genesis, which are currently Apex (RDO-Manager based), Compass4NFV (Compass based), Fuel@OPNFV (Fuel based), JOID (Juju based). This means that a common deployment of OPNFV can be achieved using any installer according to the common features or functionality delivered by Genesis.

What if a feature request did not get delivered by the Genesis project? Is there any impact on that feature being available as part of Brahmaputra?

In case Genesis does not define a feature as a common requirement for all installers participating in Genesis, then this simply means that not all installers are going to support this particular feature or functionality. It could still mean that a subset of the installers could support the feature or functionality – and as such be part of the Brahmaputra release.

How will the Brahmaputra release be maintained? Will there be specific service releases?

Brahmaputra will be maintained. Incremental fixes will be supplied to the release as patches. There won't be any specific service releases. OPNFV releases are defined by labeling in git, i.e. the Brahmaputra release will be referred to by a label in git. Brahmaputra is maintained and bug-fixes are applied to the Brahmaputra release. There will always be two active branches in OPNFV:

- "latest" – which are the set of artifacts which will become the next release (once Brahmaputra is released, this will be the content of the "C-river" release)
- "stable" – which is the Brahmaputra release plus fixes on top of Brahmaputra (i.e. on top of the labeled artifacts of Brahmaputra)

The "stable" branch will be available and stable at all times after the release until the next release. This requires higher validation settings on "stable" for all changes.

Note that once Brahmaputra is released, no further maintenance for Arno is supplied.

How will Brahmaputra be installed?

Brahmaputra installation will be facilitated by an install script which is used to install the installer on a "jumphost". Once operational, the installer will automatically install the overall OPNFV system on the hosts chosen for the installation.