Brahmaputra Testing Page

Brahmaputra Testing

Scope

Brahmaputra testing includes:

1. Brahmaputra scenarios testing for release verification:
   - executed on PODs dedicated for CI production;
   - master and stable branches;
   - testing projects: Functest and Yardstick.
2. Brahmaputra additional testing:
   - executed on CI-connected PODs, which are not allocated for CI production;
   - stable branch;
   - testing projects: all with deliverables to Brahmaputra and feature projects with specific test needs (e.g. extended measurements, tests which cause disturbances on the environment).

Note: This wiki is a snapshot in time. Definition of these common files is still work in progress!

Minutes of breakout sessions during Design summit


Labs

see Pharos Labs

Test Cases

Test cases advertized in release B must be declared in the database through the Test collection API.

src: Test collection API (23/11/2015)

After D milestone, we have

- 16 projects are declared for testing
- 88 testcases declared

A high level document will be created based on the testcases declared in the Database and will be part of the B-release deliverables of the testing group.

We must collect the OPNFV config need for these tests and capture them in a config file.

The testcases declared in the DB are described at a high level however an url field is available to point to a better description either on the wiki on on project specific deliverable.

Additionally, as the results must be stored in DB, it is recommended to add a virtual "status" testcase by project. This virtual testcase could be used to provide a global summary of each test project.

= Success Criteria

criteria for B-release:

- matrix (installer/test cases)
- test case documented
- test case in DB
- run functest and yardstick - 4 successful consecutive iterations in CI

Matrixes

Note discussions on the config files to reflect the tables: https://wiki.opnfv.org/genesis/configuration-files-discussion

Test Projects

Scenario indication shall be understood as follow: [X],[Y], with

- X = installer (apex|compass|fuel|joid)
- Y = scenario id as defined in the scenario section (e.g. 1 = bm + odl i.e. Arno)
Which feature do they need to achieve their testing?

<table>
<thead>
<tr>
<th>Test project</th>
<th>Config Need</th>
<th>Scenario</th>
<th>Tested with Apex</th>
<th>with Compass</th>
<th>with Fuel</th>
<th>with Joid</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>armband</td>
<td>Need an ARM-based lab (ANEAA)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>copper</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>doctor</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>functest</td>
<td>need to detect the controller to run either odl/onos /ovno suite</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ha</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ipv6x</td>
<td>Need to disable odl-l3 and enable neutron-l3-agent as part of the end-state of installation due to lack of support of IPv6 L3 routing in ODL Lithium</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>kvmfornv</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>qtp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>osnfv</td>
<td>custom OVS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ovno</td>
<td>opencontrail</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>promise</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sdnvpn</td>
<td>ODL needs VPN service karaf feature enabled</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sfc</td>
<td>Needs ODL Beryllium with SFC, GBP, and OVSDB SB configured. Also needs Tack installed via post-install scripts.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vnfngraph</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vsperf</td>
<td>Need Ixia Loader (e.g. Intel POD3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>yardstick</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bottlenecks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Test Scenarios for Brahmaputra**

In Arno we just had two different deployment scenarios (one based on Fuel, one based on Foreman/Quickstack as installer). With Brahmaputra we'll target a larger set. Given that we can think of more combinations than we have resources and time, we'll need to nail those scenarios that we care about for Brahmaputra.

**What is a scenario?** A scenario is a particular deployment, i.e. the deployed composition of a set of components and their associated configuration. A scenario would ultimately be described through a set of configuration files (which is why we're defining the common config files right now - [https://wiki.openfv.org/genesis/configuration-files-discussion#common_config_files](https://wiki.openfv.org/genesis/configuration-files-discussion#common_config_files)). Apart from the description of the configuration, a scenario would come with its own documentation and eventually with its own test cases (e.g. to prove the specific functionality enabled by the scenario).

Which scenarios do we take on for Brahmaputra? The “starting point” scenarios are naturally those that we inherited from Arno SR1 – which is a bare-metal or virtual deployment of (Openstack, OpenDaylight, OVS, KVM) on either Centos or Ubuntu. But we obviously desire a lot of additional ones – especially to also reflect the various feature developments for Brahmaputra.

**Call to action:**

- **Definition:** If you, your project or group-of-projects desire a scenario for Brahmaputra other than the ones we inherited from Arno SR1, please add your scenario to the table below. Arno SR1 scenarios are reflected as (A.1, C.1, F.1, J.1) and (A.2, C.2, F.2, J.2).
- **Ownership:** When adding a scenario, please specify the owner of the scenario. It is the owner of a scenario who is responsible for creating the scenario (config files, debugging/maintenance, documentation, documentation of test results, specific tests etc.). Owners of a scenario are likely feature projects. Note that installers are typically not an owner (Arno was an exception), because installers only install features and components, but take no responsibility for the correct functioning of the features/components and the associated documentation. Also note that scenarios without owners are unlikely to happen...

**Available Scenarios**

Available scenarios mean that the scenarios listed on below page are currently running on CI.

Click Jenkins link to access console log of latest build. [Jenkins view per PODs](#)

**Candidate Scenarios**

NB: each scenario needs a leader..
<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
<th>Apex</th>
<th>Compass</th>
<th>Fuel</th>
<th>Joid</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>bm odl_l3-ha</td>
<td>Apex</td>
<td>Owner</td>
<td>Status</td>
<td>Owner</td>
</tr>
<tr>
<td>2</td>
<td>virt odl_l3-ha</td>
<td>Apex</td>
<td>Owner</td>
<td>Status</td>
<td>Owner</td>
</tr>
<tr>
<td>3</td>
<td>bm odl_l2-ha</td>
<td>IP v6</td>
<td>see Apex comment on scenario 3</td>
<td>IP v6</td>
<td>see A.3 note; dropped for B-release</td>
</tr>
<tr>
<td>4</td>
<td>virt odl_l2-ha</td>
<td>IP v6</td>
<td>see Apex comment on scenario 3</td>
<td>IP v6</td>
<td>see A.3 note; dropped for B-release</td>
</tr>
<tr>
<td>9</td>
<td>virt ODL Beryllium with SFC, GBP, and OVSDB SouthBound. OVS with NSH.</td>
<td>Apex</td>
<td>Owner</td>
<td>Status</td>
<td>Owner</td>
</tr>
<tr>
<td>10</td>
<td>bm VSPERF</td>
<td>Apex</td>
<td>Owner</td>
<td>Status</td>
<td>Owner</td>
</tr>
<tr>
<td>11</td>
<td>bm, nosdn-dpdk_ovs * Ubuntu 14.0.4 * No ODL Backend * networking-ovs-dpdk mech driver * libvirt &gt; 1.2.10 * Qemu 2.1+ * One NIC must be provided solely for OVS-DPDK for the private network. (and I guess in case DVR is enabled two interfaces needs to be reserved – Public and private - not sure yet).</td>
<td>IP v6</td>
<td>see Apex comment on scenario 3</td>
<td>IP v6</td>
<td>see A.3 note; dropped for B-release</td>
</tr>
<tr>
<td>12</td>
<td>bm, Ubuntu 14, ODL with VPN Service enabled, BGPVPN with ODL backend</td>
<td>Apex</td>
<td>Owner</td>
<td>Status</td>
<td>Owner</td>
</tr>
<tr>
<td>13</td>
<td>virt, Ubuntu 14, ODL with VPN Service enabled, BGPVPN with ODL backend</td>
<td>Apex</td>
<td>Owner</td>
<td>Status</td>
<td>Owner</td>
</tr>
<tr>
<td>15</td>
<td>bm nosdn-ovs-ha</td>
<td>dropped for B-release</td>
<td>dropped for B-release</td>
<td>dropped for B-release</td>
<td>dropped for B-release</td>
</tr>
<tr>
<td>16</td>
<td>bm nosdn-kvm-ha</td>
<td>dropped for B-release</td>
<td>dropped for B-release</td>
<td>dropped for B-release</td>
<td>dropped for B-release</td>
</tr>
<tr>
<td>17</td>
<td>bm nosdn-ovs_kvm-ha</td>
<td>dropped for B-release</td>
<td>dropped for B-release</td>
<td>dropped for B-release</td>
<td>dropped for B-release</td>
</tr>
<tr>
<td>18</td>
<td>virt os-opencontrail-nofeature-ha</td>
<td>Apex</td>
<td>Owner</td>
<td>Status</td>
<td>Owner</td>
</tr>
<tr>
<td>19</td>
<td>bm os-opencontrail-nofeature-ha</td>
<td>Apex</td>
<td>Owner</td>
<td>Status</td>
<td>Owner</td>
</tr>
<tr>
<td>20</td>
<td>virt, Ubuntu 14, ONOS with OpenStack neutron-l3-agent being disabled</td>
<td>ONOS FW</td>
<td>Owner</td>
<td>Status</td>
<td>Owner</td>
</tr>
<tr>
<td>21</td>
<td>bm, Ubuntu 14, ONOS with OpenStack neutron-l3-agent being disabled</td>
<td>ONOS FW</td>
<td>Owner</td>
<td>Status</td>
<td>Owner</td>
</tr>
<tr>
<td>22</td>
<td>virt, CentOS 7, with ONOS neutron-l3-agent being disabled</td>
<td>ONOS FW</td>
<td>Owner</td>
<td>Status</td>
<td>Owner</td>
</tr>
</tbody>
</table>
possible scenarios listed but no leader yet:

- X.6: bm, ocl-ha
- X.9: virt, Centos/Ubuntu 14,
- X.10: bm, Centos 7/Ubuntu 14, VSPERF

Scenario and Jenkins Job Naming Scheme

It has been proposed to construct and align short scenario names in order to ease the Releng and Test Projects work to proceed with CI work and release verification.

An important aspect to point out is that even though short scenario names can give an idea regarding what a specific scenario is about, it is always important/necessary to look at and work with actual configuration files.

This naming convention is only valid for short scenario names and the details (config files, etc.) are handled within Genesis.

Both naming scenarios and Jenkins jobs have been touched upon during Daily Release Meeting on 2016-01-13 and there were no objections. See this link for minutes.

Scenario Naming

Short scenario names should follow the scheme below.

\texttt{os-[controller]-[feature]-[mode]-[option]}

Details of the fields are

- \texttt{os}: mandatory
  - possible value: os
  - please note that this field is needed in order to select parent jobs to list and do blocking relations between them.

- \texttt{[controller]}: mandatory
  - example values: nosdn, ocl, odl, onos

- \texttt{[feature]}: mandatory
  - example values: nofeature, kvm, ovs

- \texttt{[mode]}: mandatory
  - possible values: ha, noha

- \texttt{[option]}: optional
  - used for the scenarios those do not fit into naming scheme.
  - optional field in the short scenario name should not be included if there is no optional scenario.

Examples are

- os-nosdn-kvm-noha
- os-odl_l2-nofeature-ha
- os-onos-kvm_ovs-noha

Please note that the dashes are the separator between fields.
Underscores (_) can be used for separating the values/words if a field contains more than one value (ie kvm_ovs).

This scenario naming scheme is currently in use/followed by compass, fuel, and joid.

Jenkins Job Structure

Jenkins jobs use Freestyle project type.

Please take a look at the examples from this or this link to see job structuring.

In this new job structure, there is 1 main job per scenario-pod-branch, controlling the executions of deploy, functest, and yardstick jobs. Main jobs are used for setting triggers, triggering jobs manually or automatically.

There is 1 job for each of the deploy, functest, and yardstick for pod-branch and these jobs take scenario short names from parent job and do what they are supposed to do with given scenario on the POD.

This job structure is currently in use/followed by compass, fuel and joid.

Jenkins Job Naming

Names of Jenkins Jobs require alignment as well in order to ease the triggering, configuration, management, and checking logs.
The jobs on Jenkins should be named according to below naming scheme.

- **Parent/Main Jobs:** `[installer][scenario][pod][keep][branch]`
  - `[installer]`: apex, compass, fuel, joid
  - `[scenario]`: only the scenario names that fit into scenario naming scheme above.
  - `[pod]`: all the PODs that are assigned to specific installer and currently connected to Jenkins.
  - `[loop]`: daily
  - `[branch]`: master or brahmaputra

- **Deploy Jobs:** `[installer][deploy][pod][keep][branch]`
  - `[installer]`: apex, compass, fuel, joid
  - `[pod]`: all the PODs that are assigned to specific installer and currently connected to Jenkins.
  - `[loop]`: daily
  - `[branch]`: master or brahmaputra

- **Test Jobs:** `[testproject][installer][pod][loop][branch]`
  - `[testproject]`: functest, yardstick, bottlenecks, qtip
  - `[installer]`: apex, compass, fuel, joid
  - `[pod]`: all the PODs that are assigned to specific installer and currently connected to Jenkins.
  - `[loop]`: daily or suite
  - `[branch]`: master or brahmaputra

Example Parent/Main Job names are:
- joid-os-ocl-nofeature-ha-intel-pod6-daily-master
- fuel-os-odl_l2-nofeature-ha-opnfv-jump-2-daily-brahmaputra

Example Deploy Job names are:
- joid-deploy-intel-pod6-daily-master
- fuel-deploy-opnfv-jump-2-daily-brahmaputra

Example Test Job names are:
- yardstick-joid-intel-pod6-daily-master
- functest-fuel-opnfv-jump-2-daily-brahmaputra

See all the OPNFV Platform CI jobs on this link.

This Jenkins Job naming scheme is currently in use/followed by compass, fuel and joid.

**Brahmaputra Job naming scheme is currently in use/followed by compass, fuel and joid.**

**Brahmaputra Testing using non-CI production PODs**

The PODs dedicated for CI production activities for Brahmaputra release are to be used for stable and latest release verification, including testing using Functest and Yardstick. Those PODs are fully allocated in daily basis for these activities.

In order to cover verification for all the testing frameworks delivering in Brahmaputra, it is necessary to use other OPNFV PODs, connected to CI, but not assigned to CI production.

The PODs listed in Pharos are eligible for the additional testing, see: [https://wiki.opnfv.org/pharos_rls_b_labs#community_lab_resources_-_operational](https://wiki.opnfv.org/pharos_rls_b_labs#community_lab_resources_-_operational)

The table below shows the additional testing and the lab used for it.

<table>
<thead>
<tr>
<th>Test Project</th>
<th>Scope of testing</th>
<th>Lab used</th>
<th>Installer</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yardstick</td>
<td>Test cases for HA Project</td>
<td>China Mobile</td>
<td>Fuel</td>
<td>On demand testing; disturbs the POD; Responsible for testing: Yardstick + HA; responsible for documentation: Yardstick, with review from HA</td>
</tr>
<tr>
<td>Yardstick</td>
<td>Yardstick generic test cases + Test cases for virtual Traffic Classifier, part of Yardstick</td>
<td>Ericsson POD1</td>
<td>Fuel</td>
<td>Daily Jenkins Job; Responsible for testing: Yardstick; responsible for documentation: Yardstick</td>
</tr>
<tr>
<td>Yardstick</td>
<td>Yardstick generic test cases, additional measurements</td>
<td>ZTE</td>
<td>Fuel</td>
<td>Daily Jenkins Job; Responsible for testing: Yardstick; Responsible for documentation: Yardstick</td>
</tr>
<tr>
<td>Yardstick</td>
<td>Test cases for Parcer Project</td>
<td>Huawei</td>
<td>Compass</td>
<td>On demand, verify Parser Tosca2Heat; Responsible for testing: Yardstick + Parser; responsible for documentation: Yardstick, with review from Parser</td>
</tr>
<tr>
<td>QTIP</td>
<td>QTIP test suite</td>
<td>Dell POD1</td>
<td>Fuel</td>
<td>Daily Jenkins Job for QTIP test suite. Responsible for documentation</td>
</tr>
<tr>
<td>QTIP</td>
<td>QTIP test suite</td>
<td>Dell POD3</td>
<td>Fuel</td>
<td>Daily Jenkins Job for QTIP test suite.</td>
</tr>
<tr>
<td>VSPERF</td>
<td>VSPERF test suite</td>
<td>Intel POD3</td>
<td>None</td>
<td>Daily Jenkins Job for VSPERF test suite.</td>
</tr>
<tr>
<td>bottlenecks</td>
<td>rubbos and VSTF test suite</td>
<td>Huawei</td>
<td>Compass</td>
<td>Daily Jenkins Job; Responsible for testing: bottlenecks; Responsible for documentation: bottlenecks</td>
</tr>
</tbody>
</table>