Virtualization development improved with Lago

Rafael Martins  
Software Engineer  
Red Hat  
FOSDEM 2017, February 4 2017
Virtualization-related products are hard to test manually and automatically.
What is oVirt?

oVirt manages virtual machines, storage and virtualized networks.

oVirt is a virtualization platform with an easy-to-use web interface.

oVirt is powered by the Open Source you know - KVM on Linux.
What is oVirt?
Lago Project

- Ad-hoc virtual framework which helps you build virtualized environments on your server or laptop for various use cases.
- Based on libvirt and KVM
- Easily extensible in Python, with plugin infrastructure.
oVirt support is implemented as a Lago plugin (ovirtlago), that knows how to deal with oVirt engine and hosts.
oVirt System Tests

LAGO

Engine

Host

Host
oVirt System Tests

oVirt relies on a suite of automated tests, that deploy and run several tests cases.

- Quick visibility of critical breakages.
- Well maintained set of test cases, frequently improved.
- Virtual machines are left to be used after test execution.
## oVirt System Tests

Welcome to oVirt Jenkins continuous integration server

For more information about the oVirt project visit http://www.ovirt.org

<table>
<thead>
<tr>
<th>S</th>
<th>W</th>
<th>Name</th>
<th>Last Success</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="https://example.com/red.png" alt="Red" /> <img src="https://example.com/cloud.png" alt="Cloud" /></td>
<td><img src="https://example.com/lightning.png" alt="Lightning" /></td>
<td>ovirt_3.6_he-system-tests</td>
<td>1 day 6 hr - #808</td>
</tr>
<tr>
<td><img src="https://example.com/red.png" alt="Red" /> <img src="https://example.com/cloud.png" alt="Cloud" /></td>
<td><img src="https://example.com/lightning.png" alt="Lightning" /></td>
<td>ovirt_3.6_image-ng-system-tests</td>
<td>21 days - #285</td>
</tr>
<tr>
<td><img src="https://example.com/red.png" alt="Red" /> <img src="https://example.com/cloud.png" alt="Cloud" /></td>
<td><img src="https://example.com/lightning.png" alt="Lightning" /></td>
<td>ovirt_3.6_system-tests</td>
<td>1 day 6 hr - #855</td>
</tr>
<tr>
<td><img src="https://example.com/lightning.png" alt="Lightning" /></td>
<td><img src="https://example.com/sun.png" alt="Sun" /></td>
<td>ovirt_3.6_system-tests_manual</td>
<td>N/A</td>
</tr>
<tr>
<td><img src="https://example.com/red.png" alt="Red" /> <img src="https://example.com/cloud.png" alt="Cloud" /></td>
<td><img src="https://example.com/lightning.png" alt="Lightning" /></td>
<td>ovirt_4.0_he-system-tests</td>
<td>3 days 6 hr - #626</td>
</tr>
<tr>
<td><img src="https://example.com/green.png" alt="Green" /> <img src="https://example.com/sun.png" alt="Sun" /></td>
<td><img src="https://example.com/sun.png" alt="Sun" /></td>
<td>ovirt_4.0_image-ng-system-tests</td>
<td>7 hr 29 min - #156</td>
</tr>
<tr>
<td><img src="https://example.com/green.png" alt="Green" /> <img src="https://example.com/sun.png" alt="Sun" /></td>
<td><img src="https://example.com/sun.png" alt="Sun" /></td>
<td>ovirt_4.0_system-tests</td>
<td>6 hr 32 min - #773</td>
</tr>
<tr>
<td><img src="https://example.com/green.png" alt="Green" /> <img src="https://example.com/sun.png" alt="Sun" /></td>
<td><img src="https://example.com/sun.png" alt="Sun" /></td>
<td>ovirt_4.0_system-tests_manual</td>
<td>4 days 1 hr - #8</td>
</tr>
<tr>
<td><img src="https://example.com/red.png" alt="Red" /> <img src="https://example.com/cloud.png" alt="Cloud" /></td>
<td><img src="https://example.com/lightning.png" alt="Lightning" /></td>
<td>ovirt_4.1_system-tests</td>
<td>9 days 7 hr - #27</td>
</tr>
<tr>
<td><img src="https://example.com/green.png" alt="Green" /> <img src="https://example.com/sun.png" alt="Sun" /></td>
<td><img src="https://example.com/sun.png" alt="Sun" /></td>
<td>ovirt_4.1_system-tests_manual</td>
<td>9 days 18 hr - #2</td>
</tr>
<tr>
<td><img src="https://example.com/red.png" alt="Red" /> <img src="https://example.com/cloud.png" alt="Cloud" /></td>
<td><img src="https://example.com/lightning.png" alt="Lightning" /></td>
<td>ovirt_master_system-tests</td>
<td>1 day 6 hr - #906</td>
</tr>
<tr>
<td><img src="https://example.com/green.png" alt="Green" /> <img src="https://example.com/cloud.png" alt="Cloud" /></td>
<td><img src="https://example.com/lightning.png" alt="Lightning" /></td>
<td>ovirt_master_system-tests_manual</td>
<td>13 hr - #20</td>
</tr>
<tr>
<td><img src="https://example.com/green.png" alt="Green" /> <img src="https://example.com/sun.png" alt="Sun" /></td>
<td><img src="https://example.com/sun.png" alt="Sun" /></td>
<td>ovirt_master_system-tests_on_demand</td>
<td>1 day 22 hr - #34</td>
</tr>
</tbody>
</table>
What about manual testing?

oVirt System Tests relies on the “ovirtlago” plugin, that supports installation of custom RPM packages.

- oVirt project provides Jenkins jobs to build RPM packages from Gerrit patches.
- A developer can build RPMs on his laptop!
- Simple patches can be verified by running the full test suite with a custom RPM.
What about manual testing?

There are a few downsides when using oVirt System tests to run manual tests:

- Using oVirt System Tests to do manual testing requires running the full test suite. This is not always required/wanted.

- If the patch changes the behavior expected by the test suite, it needs to be adapted too.

- Sometimes the environment deployed by oVirt System tests is not suitable to verify the patch.
oVirt Patch Verifier

Relies on Lago and the ovirtlago plugin to create test environments that are targeted to manual testing.

- Environment is defined easily on command line, when calling the plugin to deploy.
- No need to run automated test suites to get oVirt engine and hosts running.
- Can create as many oVirt hosts as wanted, and attach them to the oVirt engine automatically.
oVirt Patch Verifier

- $ lago opv deploy \
  --vm engine,name=engine,memory=8192 \
  --vm host,name=host1 \
  --vm host,name=host2 \
  --custom-source http://jenkins.ovirt.org/job/...

- $ lago opv engine-setup

- $ lago shell engine
  # engine-setup
oVirt Patch Verifier

* Starting VM engine: Success (in 0:00:00)
* Starting VM host2: Success (in 0:00:00)
* Starting VM host1: Success (in 0:00:00)
# Start vms: Success (in 0:00:01)
@ Start Prefix: Success (in 0:00:07)
@ Deploy oVirt environment:
  # Deploy environment:
  * [Thread-5] Deploy VM engine: Success (in 0:00:23)
  * [Thread-6] Deploy VM host2: Success (in 0:00:23)
  * [Thread-7] Deploy VM host1: Success (in 0:04:51)
  # Deploy environment: Success (in 0:04:51)
@ Deploy oVirt environment: Success (in 0:04:52)
[rmartins@dev-19 prefix]$
There are also some downsides on this approach:

- Fully automated, requires caching of a lot of unneeded RPMs for the internal repository, or download of dependencies for each VM.
- Can’t deploy more than one oVirt engine simultaneously.
Supporting other projects

Lago is not tied to oVirt, so it means that it is possible to write plugins to test other projects.

- Test virtualization managers
- Test appliances
- Build virtualization environment
Why should I use Lago instead of ... ?

Lago may be compared to other similar tools, like:

- Vagrant
- Avocado
- LAVA
THANK YOU!

http://www.ovirt.org
http://lago.readthedocs.io/

rmartins@redhat.com
rafaelmartins @ GitHub, Twitter, Freenode, OFTC
https://rgm.io/