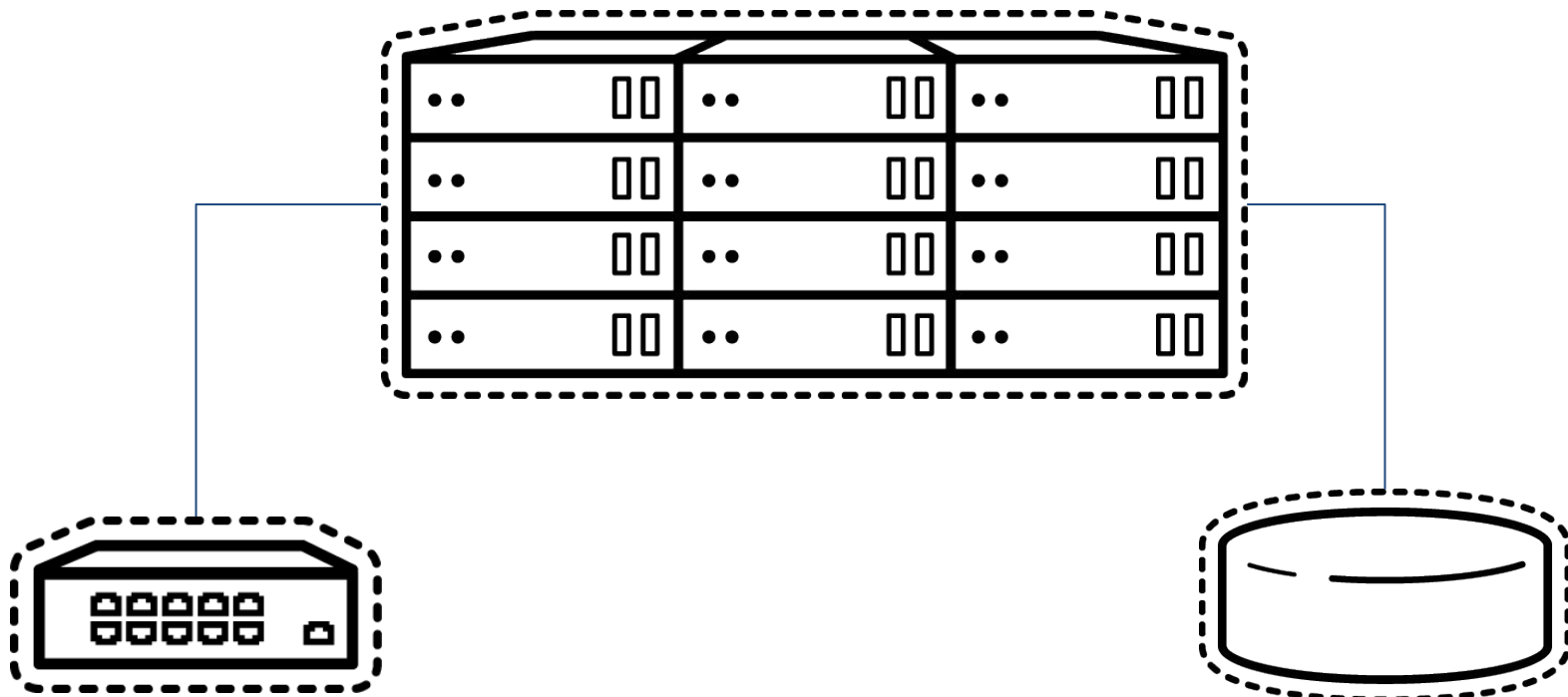


Virtualization development improved with Lago

Rafael Martins
Software Engineer
Red Hat
FOSDEM 2017, February 4 2017

Virtualization-related products are hard to test manually and automatically



oVirt 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.

oVirt

oVirt What is oVirt?

The screenshot displays the oVirt dashboard interface. At the top, there is a search bar and navigation tabs for Dashboard, Data Centers, Clusters, Hosts, Networks, Storage, Disks, Virtual Machines, and Pools. The left sidebar contains a 'System' section with a tree view of resources including Data Centers (lab3, lab4), local storage, networks, templates, and clusters, as well as External Providers (ovirt-image-repository). The main content area shows a 'Last Updated' timestamp and a grid of summary cards for Data Centers, Clusters, Hosts, Storage Domains, Virtual Machines, and Events. Below this is a 'Global Utilization' section with three donut charts for CPU (16% used), Memory (408.0 GiB used), and Storage (1.8 TiB used). The bottom status bar shows the last message, alerts, events, and tasks.

Dashboard: [x] [☆] [Q]

Dashboard | Data Centers | Clusters | Hosts | Networks | Storage | Disks | Virtual Machines | Pools

System
Expand All Collapse All

- System
 - Data Centers
 - lab3
 - Storage
 - Networks
 - Templates
 - Clusters
 - lab4
 - Storage
 - Networks
 - Templates
 - Clusters
 - local
 - Storage
 - Networks
 - Templates
 - Clusters
 - External Providers
 - ovirt-image-repository

System Overview
Last Updated: 1/24/2017 5:14:04 PM

| | | |
|----------------------------|-------------------------------------|--------------------|
| 3 Data Centers 1 2 | 4 Clusters N/A | 29 Hosts 1 28 |
| 3 Storage Domains 1 2 | 190 Virtual Machines 2 96 92 | 498 Events 498 |

Global Utilization

| Metric | Used | Available | Over Commit |
|---------|-----------|-------------------|----------------------|
| CPU | 16% | 84% of 100% | 97% (allocated 224%) |
| Memory | 408.0 GiB | 85.0 of 492.9 GiB | 68% (allocated 294%) |
| Storage | 1.8 TiB | 1.7 of 3.5 TiB | 41% (allocated 222%) |

Last Message: ✓ Jan 24, 2017 5:14:23 PM VM nested-rgj-7.2-20160302.0-ge_worker-6 was restarted on Host cinteg21.ci.lab.tlv.re... Alerts (0) Events Tasks (2)

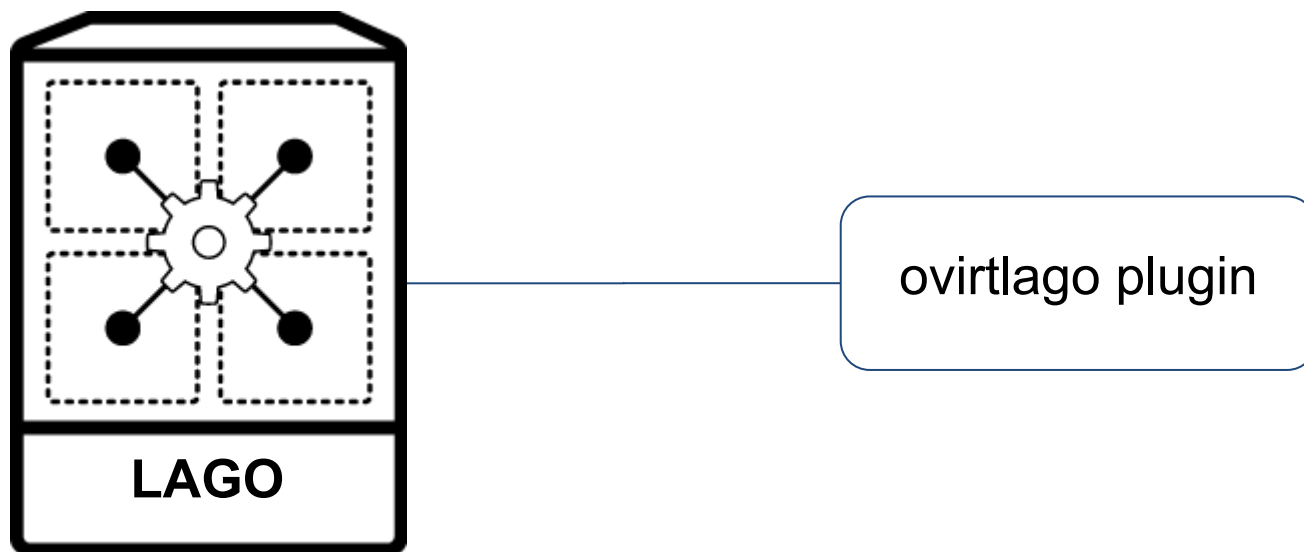
Lago Project

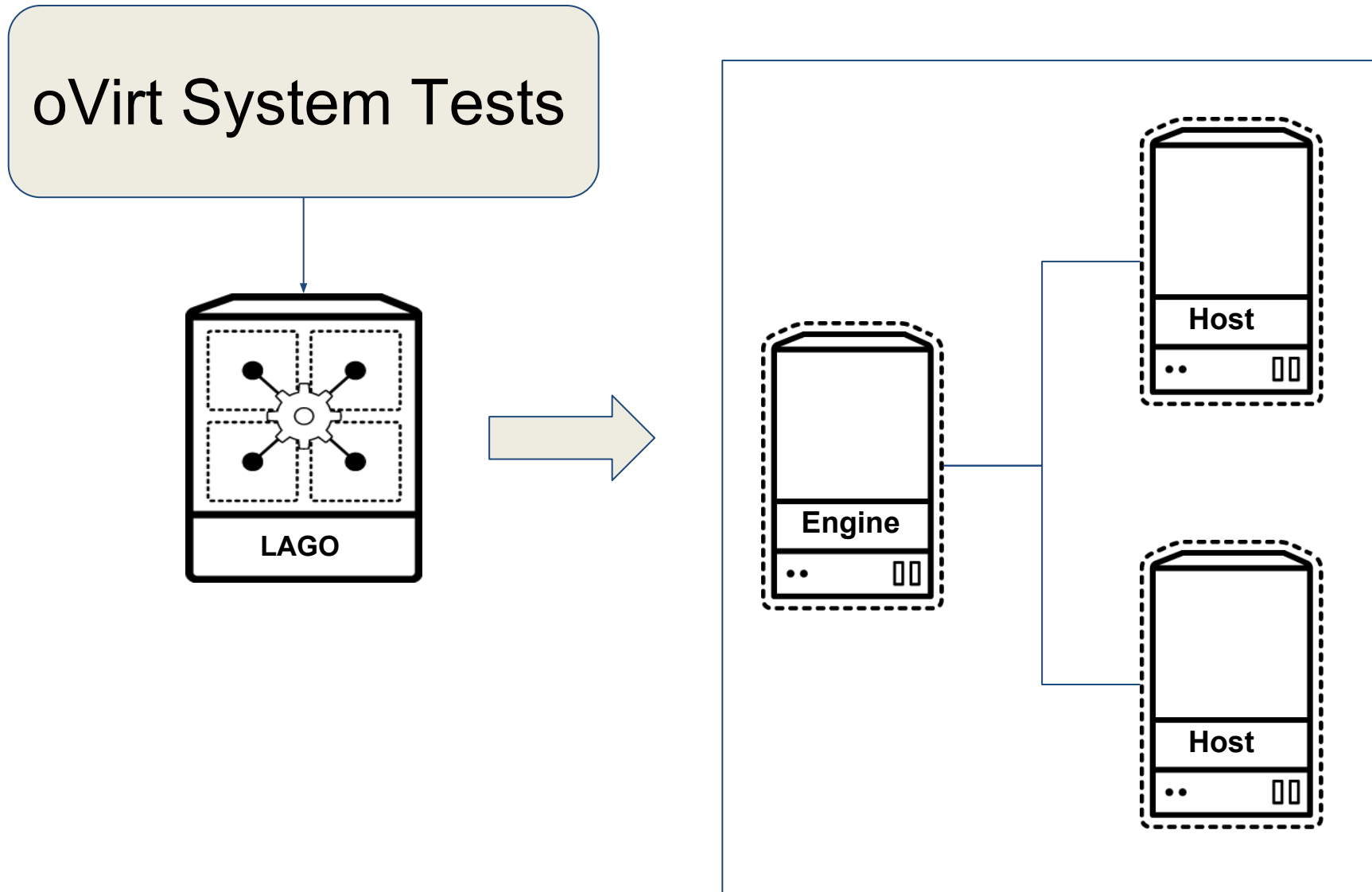
- Ad-hoc virtual framework which helps you build virtualized environments on your server or laptop for various use cases.
- Based on libvirt and KVM
- Easily extensible in Python, with plugin infrastructure.



oVirt How oVirt uses Lago

oVirt support is implemented as a Lago plugin (ovirtlago), that knows how to deal with oVirt engine and hosts.































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.

Welcome to oVirt Jenkins continuous integration server

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

| S | W | Name ↓ | Last Success |
|---|---|---|------------------------------------|
|  |  | ovirt_3.6_he-system-tests | 1 day 6 hr - #808 |
|  |  | ovirt_3.6_image-ng-system-tests | 21 days - #285 |
|  |  | ovirt_3.6_system-tests | 1 day 6 hr - #855 |
|  |  | ovirt_3.6_system-tests_manual | N/A |
|  |  | ovirt_4.0_he-system-tests | 3 days 6 hr - #626 |
|  |  | ovirt_4.0_image-ng-system-tests | 7 hr 29 min - #156 |
|  |  | ovirt_4.0_system-tests | 6 hr 32 min - #773 |
|  |  | ovirt_4.0_system-tests_manual | 4 days 1 hr - #8 |
|  |  | ovirt_4.1_system-tests | 9 days 7 hr - #27 |
|  |  | ovirt_4.1_system-tests_manual | 9 days 18 hr - #2 |
|  |  | ovirt_master_system-tests | 1 day 6 hr - #906 |
|  |  | ovirt_master_system-tests_manual | 13 hr - #20 |
|  |  | ovirt_master_system-tests_on_demand | 1 day 22 hr - #34 |

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.

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.

- \$ lago opv deploy \
 --vm engine,name=engine,memory=8192 \
 --vm host,name=host1 \
 --vm host,name=host2 \
 --custom-source [http://jenkins.ovirt.org/job/...](http://jenkins.ovirt.org/job/)
- \$ lago opv engine-setup
- \$ lago shell engine
 # engine-setup

```
martins@dev-19:~$ ./ovirt-patch-verifier.py
* Starting VM engine:
* Starting VM engine: Success (in 0:00:00)
* Starting VM host2:
* Starting VM host2: Success (in 0:00:00)
* Starting VM host1:
* 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:
* [Thread-6] Deploy VM host2:
* [Thread-7] Deploy VM host1:
* [Thread-6] Deploy VM host2: Success (in 0:00:23)
* [Thread-7] Deploy VM host1: Success (in 0:00:23)
* [Thread-5] Deploy VM engine: 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.

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

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/>