OpenQA testing platform

Train your own testing pet

Lukáš Růžička
(lruzicka@redhat.com)

Fedora QE

April 2020

What is openQA?

openQA is an automated test tool, developed by SuSE, that allows to test various features of operating systems using the *hands-on* approach:

- it runs the operating system
- it provides an interface (console, GUI)
- it takes input, passes it to the machine and triggers actions
- it evaluates the reactions

It behaves similarly to a *human* tester.

Some technologies used

qemu	runs the machines
VNC	provides the screens
perl scripts	define actions and evaluations

openQA architecture

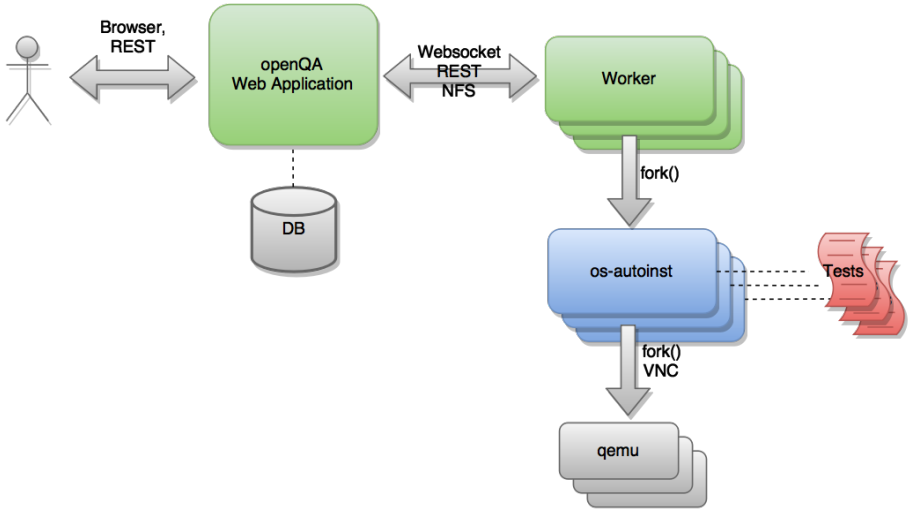
Controller

- Web UI (control and visualization)
- job handling (scheduling, starting, stopping)
- live viewing and interaction (monitoring and development)
- results
- database (postgresql)

Worker

- assets (iso or qcow2)
- tests and needles
- syncing and caching
- video recording

openQA architecture diagram



Basic concepts

- machine
- tests grouping
- image
- job
- test suite
- test
- needle

Machine

A **machine** is a qemu based virtual machine created with various settings. The following example shows the predefined **UEFI x86_64** machine.

```
ARCH_BASE_MACHINE=64bit
PART_TABLE_TYPE=gpt
QEMUCPU=Nehalem
QEMUCPUS=2
QEMURAM=2048
QEMUVGA=virtio
QEMU_VIRTIO_RNG=1
UEFI=1
UEFI_PFLASH_CODE=/usr/share/edk2/ovmf/OVMF_CODE.fd
UEFI_PFLASH_VARS=/usr/share/edk2/ovmf/OVMF_VARS.fd
WORKER_CLASS=qemu_x86_64
```

Tests grouping

Single tests can be organized into groups according to several criteria and different test groups can be run for different images.

distri
: a collection of definitions, perl functions, methods, tests, and needles, that covers the entire testing programme, such as `os-autoinst-distri-fedora`

product
: the main *system under test* (SUT), such as **fedora** that is defined by a *version*, *flavor*, *arch*, and has *settings* assigned.

version
: one version of a product, such as **Rawhide** or **32**.

flavor
: a specific variant of a product to distinguish differing variants, e.g. **Workstation-live-iso**.

arch
: an architecture variant of a product, e.g. **x86_64**.

settings
: variables to provide various settings, e.g. `DESKTOP=gnome`.

Image

Images are image files with the operating system to be tested. They are used to create and provide the virtual machines on which the actual tests are performed.

- iso $\longrightarrow$ qcow2
- qcow2

Job

is one *physical* run of a *test case* or a *test suite*, defined as:

fedora-Rawhide-KDE-live-iso-x86_64-BuildFedora-Rawhide-20200329.n.0-base_service_manipulation@64bit

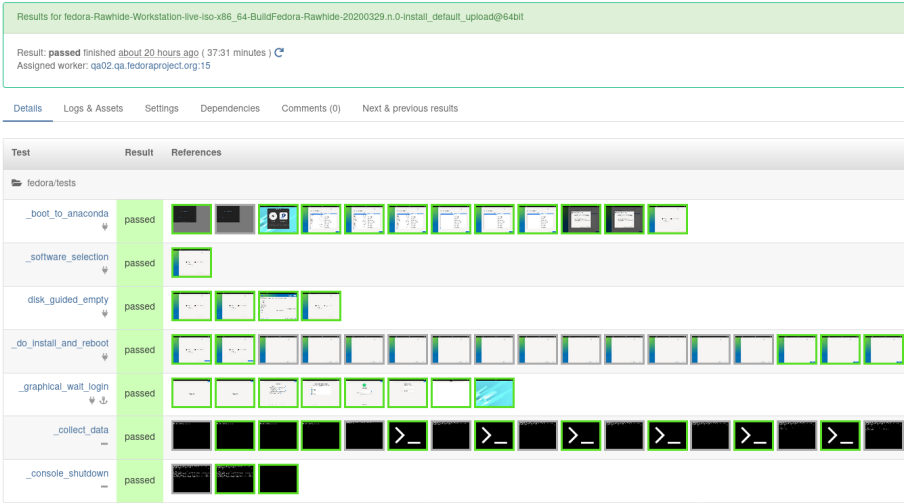
- product ⟶ fedora
- version ⟶ Rawhide
- flavor ⟶ KDE-live-iso
- arch ⟶ x86_64
- build ⟶ BuildFedora-Rawhide-20200329.n.0
- test suite ⟶ base_service_manipulation
- machine ⟶ 64bit

Test Suite

a collection of several test cases that follow one another and together make sense, for example the **install_default_upload** test suite:

1. _boot_to_anaconda
2. _software_selection
3. disk_guided_empty
4. _do_install_and_reboot
5. _graphical_wait_login
6. _collect_data
7. _console_shutdown

A WebUI look on a test suite (job)



Test

A **test** is a Perl script that defines what to do with the running virtual machine content and what to expect.
It mainly:

- defines mouse actions
- defines keyboard actions
- checks and evaluates needles
- evaluates script outcomes

The test (job) can have various *statuses*, such as **passed**, **failed**, **softfailed**, **cancelled**, **running**, or **none**, etc.

Test example

```
# Open the text editor and print the file.
wait_screen_change { send_key "alt-f2"; };
wait_still_screen(stilltime=>5, similarity_level=>45);
type_very_safely "$editor /home/test/testfile.txt\n";
wait_still_screen(stilltime=>5, similarity_level=>44);

# Print the file using the Cups-PDF printer
send_key "ctrl-p";
wait_still_screen(stilltime=>5, similarity_level=>45);
if ($desktop eq 'gnome') {
    assert_and_click "printing_select_pdfprinter";
}
else {
    assert_screen "printing_pdfprinter_ready";
}
```