

JUNE 06, 2020, NÜRNBERG, GERMANY

# A quest for test automation

Testing operating system builds, multiple times a day,  
in different architectures

# openQA



<https://open.qa>

## 01 Web UI

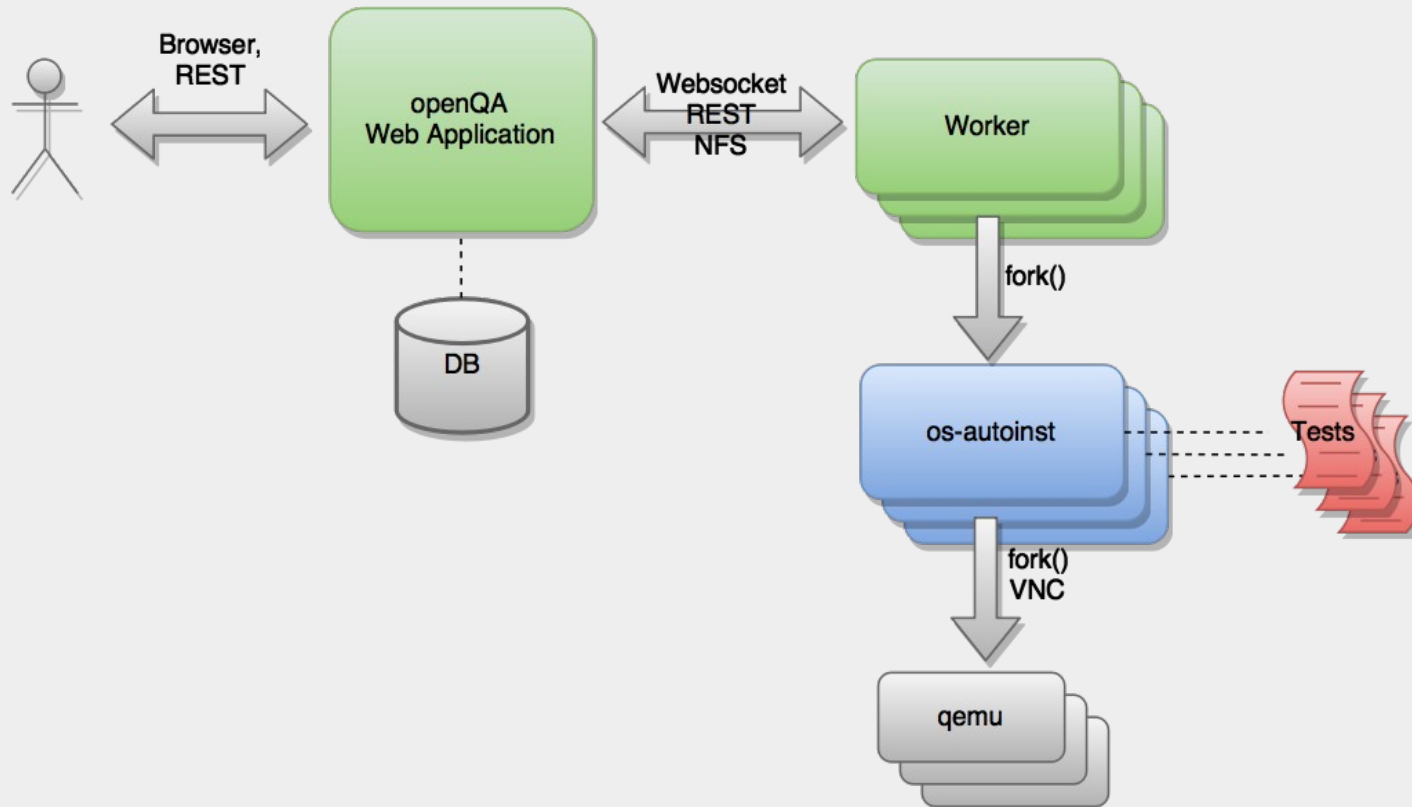
- Definitions
- Scheduling
- Reviews
- Asset Provisioning

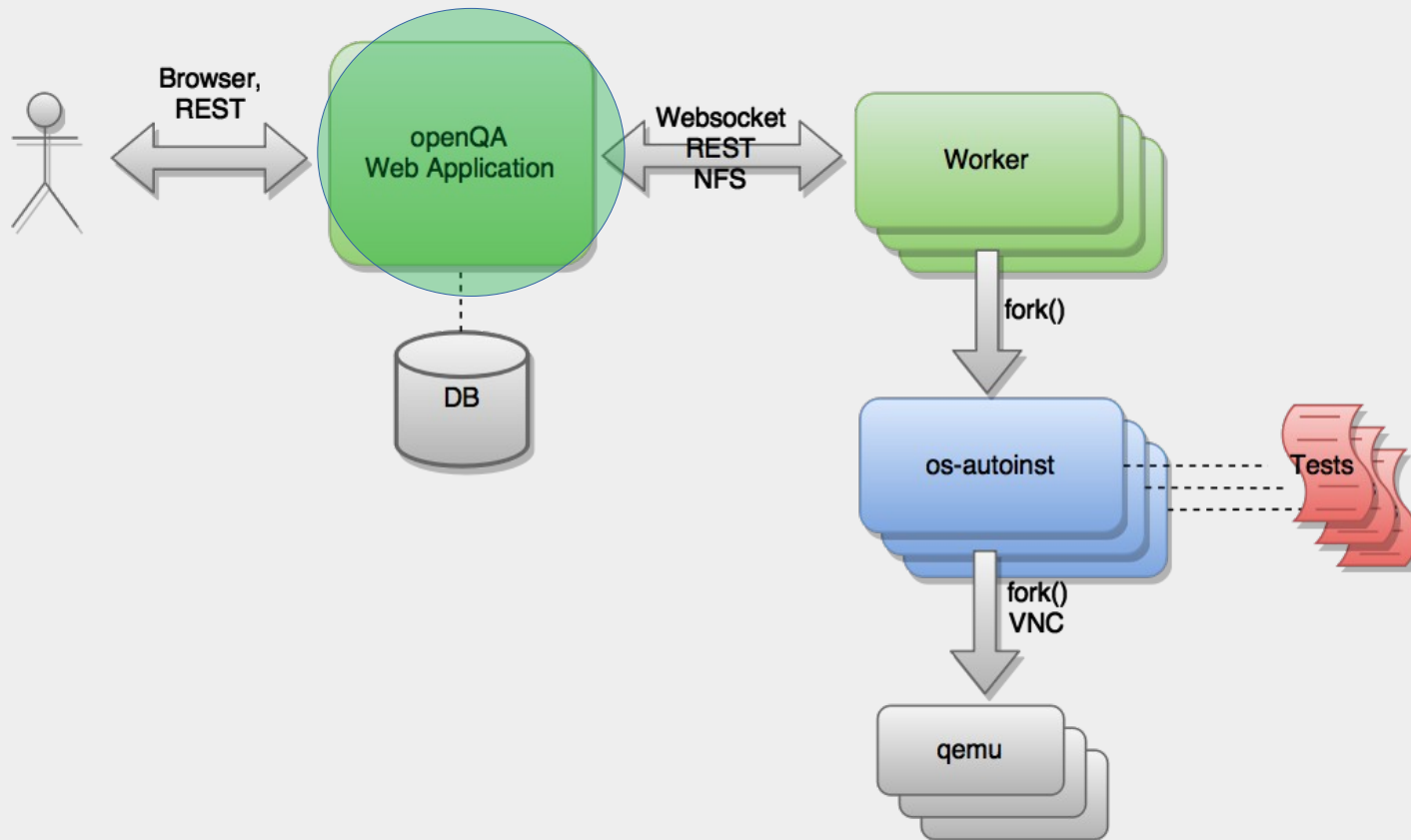
## 02 Worker

- Setup
- Execution
- Reporting

## 03 Test distribution

- DSL
- Interaction with System under Test
- Logging





- Written in Perl with Mojolicious
- PostgreSQL as database
- Exposes a REST api
  - Users
  - Workers
  - Thirdparty apps/scripts

75 jobs are running

Show 10 entries Search:

Medium	Test	Progress	Started
Build20200625 of opensuse-Tumbleweed-XFCE-Live.x86_64	xfce-live@64bit	2 %	2 minutes ago
Build20200625 of opensuse-Tumbleweed-DVD.x86_64	desktopapps-remote-desktop-xrdp-server1@64bit_virtio	running	2 minutes ago
Build20200625 of opensuse-Tumbleweed-DVD.x86_64	desktopapps-remote-desktop-xrdp-client1@64bit_virtio	0 %	2 minutes ago
Build20200625 of opensuse-Tumbleweed-KDE-Live.x86_64	kde-live_installation@uefi-2G	9 %	3 minutes ago
Build20200625 of opensuse-Tumbleweed-KDE-Live.x86_64	kde-live_installation@64bit	9 %	3 minutes ago
Build20200625 of opensuse-Tumbleweed-KDE-Live.x86_64	kde-live-wayland@64bit_virtio-3G	5 %	3 minutes ago
Build314.3 of opensuse-15.2-DVD.ppc64le	textmode@ppc64le	20 %	8 minutes ago
Build20200625 of opensuse-Tumbleweed-JeOS-for-kvm-and-xen.x86_64	jeos-extra@64bit_virtio-2G	11 %	9 minutes ago
Build20200625 of opensuse-Tumbleweed-JeOS-for-kvm-and-xen.x86_64	jeos-containers@64bit_virtio	56 %	10 minutes ago
Build20200625 of opensuse-Tumbleweed-DVD.x86_64	gnome-x11-ibus@64bit	67 %	10 minutes ago

Showing 1 to 10 of 75 entries

Previous 1 2 3 4 5 ... 8 Next

- Test result overview
  - State of the build
  - Scenarios
  - Failures

### Test result overview

Overall Summary of opensuse Tumbleweed build 20200620

Passed: 390 Incomplete: 7 Soft-Failed: 978 Failed: 157 None: 17

























































































































































































































































































































Filter ⓘ

no filter present, click to toggle filter form

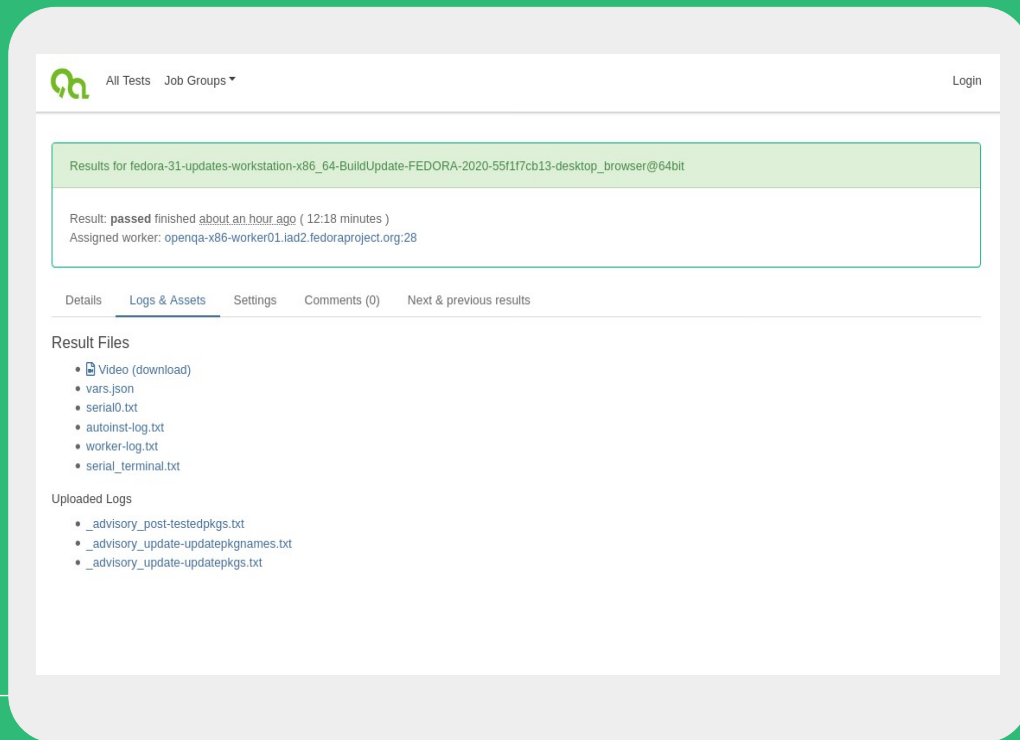
Flavor: DVD

Test	aarch64	i586	ppc64	ppc64le	s390x	x86_64
RAID0	-	-	☞ ●	☞ ●	-	-
RAID0_gpt	-	-	-	-	☞ ●	☞ ●
RAID0_gpt_uefi	☞ ●	-	-	-	-	-
RAID1	-	-	☞ ●	☞ ●	-	-
RAID10	-	-	-	☞ ●	-	-
RAID10_gpt	-	-	-	-	-	☞ ●
RAID10_gpt_uefi	☞ ●	-	-	-	-	-
RAID1_gpt	-	-	-	-	-	☞ ●
RAID1_gpt_uefi	☞ ●	-	-	-	-	-
RAID5	-	-	☞ ● first_boot	☞ ●	-	-

- Displays test results
- Provisions assets
- Help with investigation

installation_overview 26s ⚡	passed	    
disable_grub_timeout 59s ⚡	passed	            
start_install 25s ⚡	passed	    
await_install 12m 21s ⚡	passed	            
logs_from_installation_system 1m 52s	passed	                                      
reboot_after_installation 3s ⚡	passed	 
grub_test 10s ⚡	passed	
first_boot ⚡↓		                                                                                                                                                                                                                                         

- Displays test results
  - Video
  - Test Data
  - Test driver logs
  - Test logs
- Help with investigation





- Displays test results
- Provisions assets
- Help with investigation
  - Control SUT
  - Connect via VNC
  - Stop at screen match
  - Stop at a test module
  - Stop on next command

The screenshot displays the openSUSE TAF web interface. At the top, it shows the logo, 'All Tests', 'Job Groups', and a user login 'Logged in as szarate'. The main section is titled 'Results for opensuse-Tumbleweed-DVD-aarch64-Build20200625-create\_hdd\_gnome-x11@aarch64'. It indicates the test is 'running', started 'about an hour ago', and provides details about the scheduled product and assigned worker. Below this, a navigation bar includes 'Details', 'Live View' (which is active), 'Settings', 'Dependencies', 'Comments (0)', and 'Next & previous results'.

The 'Developer mode' section is expanded, showing 'regular test execution' and the 'current module: shutdown-shutdown at type\_string'. It includes instructions to change test behavior and a confirmation button 'Confirm to control this test'. Below the controls, a terminal log shows the following output:

```
Jun 26 12:15:58.592986 susetest gdm-password][4433]: pam_unix(gdm-password:auth): conversation failed
Jun 26 12:15:58.592994 susetest gdm-password][4433]: pam_unix(gdm-password:auth): auth could not identify password for [bernhard]
Jun 26 12:17:28.528747 susetest gdm-password][4446]: gkr-pam: couldn't get the password from user: Conversation error
```

- Displays test results
- Provisions assets
- Help with investigation
  - Test driver live log
  - SUT Serial console log

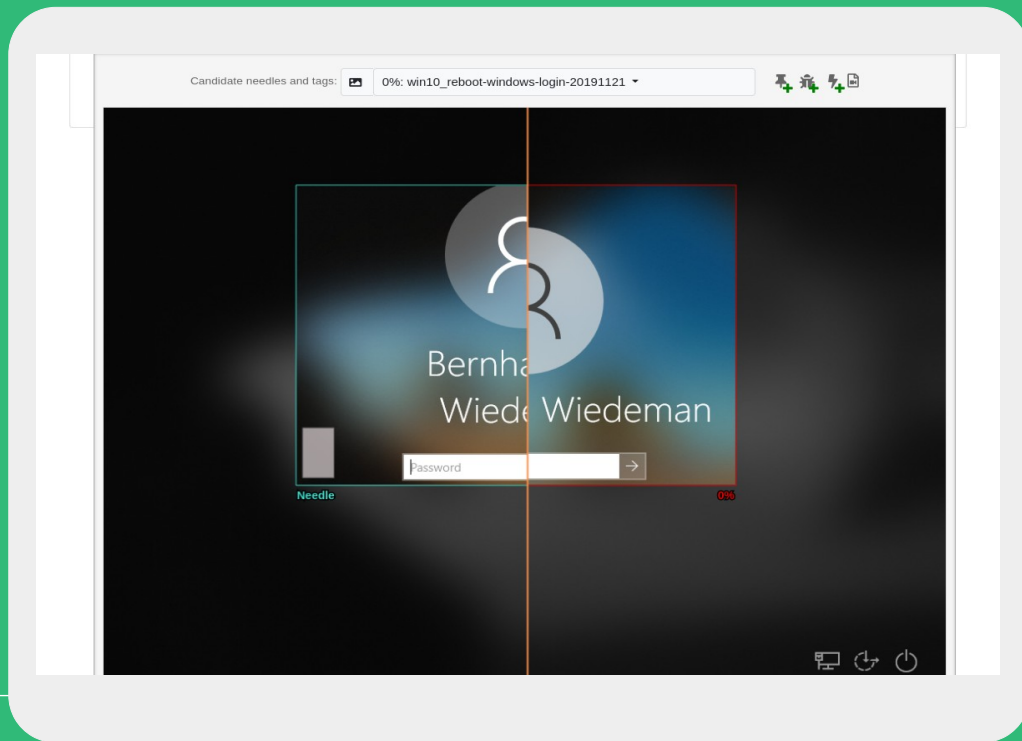
```

573.009621] BTRFS info (device wdaz2): balance: start -msusage=5
573.069551] BTRFS info (device wdaz2): balance: ended with status: 0
573.082397] BTRFS info (device wdaz2): balance: start -msusage=10
573.082397] BTRFS info (device wdaz2): balance: ended with status: 0
573.181656] BTRFS info (device wdaz2): balance: start -msusage=0 -msusage=0
573.185345] BTRFS info (device wdaz2): balance: ended with status: 0
573.137171] BTRFS info (device wdaz2): balance: start -msusage=3 -msusage=3
573.137171] BTRFS info (device wdaz2): relocating block group 23828096 flags systemdup
573.182157] BTRFS info (device wdaz2): balance: ended with status: 0
573.967668] BTRFS info (device wdaz2): scrub: started on devid 1
583.864326] BTRFS info (device wdaz2): group scan completed (inconsistency flag cleared)
593.042659] BTRFS info (device wdaz2): scrub: finished on devid 1 with status: 0
918.749965] systemd-journal[388]: Received client request to relinquish /var/log/journal/4d93b610144085a8ecd53abc3867 ac
cess.
918.769331] systemd[1]: Stopping Journal Service...
918.778535] systemd-journal[388]: Received SIGTERM from PID 1 (systemd).
918.779364] systemd[1]: systemd-journal.service: Succeeded.
918.779458] systemd[1]: Stopped Journal Service.
918.779793] systemd[1]: Starting Journal Service...
911.074968] systemd[1]: Started Journal Service.
911.122440] systemd-journal[4363]: Received client request to flush runtime journal.
913.266325] rftkll: input handler disabled
1373.357678] sysrq: Show Blocked State
1373.358240] task PC stack pid father
1376.958080] rftkll: input handler enabled
systemctl* curl -form upload-0/tmp/dmesg.log --form upname=shutdown-dmesg.log http://10.0.2.2:28094/ads52rCdJfzkc/uploadlo
g/dmesg.log; echo st01

```

```
[2020-06-26T18:28:55.977 CEST] [debug] tests/shutdown/shutdown.pm:37 called opensebasetest::post_fail_hook > lib/opensebasetest.pm:1226  
called opensebasetest::export_logs > lib/opensebasetest.pm:436 called opensebasetest::export_logs_basic >  
lib/opensebasetest.pm:418 called opensebasetest::save_and_upload_log > lib/opensebasetest.pm:81 called testapi::script_run  
[2020-06-26T18:28:55.977 CEST] [debug] << testapi::wait_serial(timeout=30, no_regex=0, expect_not_found=0, buffer_size=undef,  
record_output=undef, quiet=undef, regex=/^C408-Vd-/) >  
[32m[2020-06-26T18:28:58.186 CEST] [debug] >>> testapi::wait_serial: (7?<C408-Vd->) : ok  
[0m[2020-06-26T18:28:58.186 CEST] [debug] tests/shutdown/shutdown.pm:37 called opensebasetest::post_fail_hook > lib/opensebasetest.pm:1226  
called opensebasetest::export_logs > lib/opensebasetest.pm:436 called opensebasetest::export_logs_basic >  
lib/opensebasetest.pm:418 called opensebasetest::save_and_upload_log > lib/opensebasetest.pm:82 called testapi::upload_logs  
[2020-06-26T18:28:58.186 CEST] [debug] << testapi::upload_logs(file="/tmp/dmsg.log", failOk=0, timeout=90)  
[2020-06-26T18:28:58.187 CEST] [debug] tests/shutdown/shutdown.pm:37 called opensebasetest::post_fail_hook > lib/opensebasetest.pm:1226  
called opensebasetest::export_logs > lib/opensebasetest.pm:436 called opensebasetest::export_logs_basic >  
lib/opensebasetest.pm:418 called opensebasetest::save_and_upload_log > lib/opensebasetest.pm:82 called testapi::upload_logs  
[2020-06-26T18:28:58.187 CEST] [debug] << testapi::assert_script_output(cmd="curl -f -o /tmp/uplog/<tap/dmsg.log -f -o upname=shutdown-  
dmesg.log http://118.8.2.2:9043/adfs5x2cdVdW1zkc/unloadlog/dmesg.log" quiet=undef, fail message="" timeout=90)
```

- Displays test results
- Provisions assets
- Help with investigation
  - Why a screen didn't match





- Products
  - What do we test
- Scenarios
  - How the test should look like
- Machines
  - Where do we test
    - Architectures
    - Worker Classes

openQA All Tests Job Groups Logged in as szarate

### Job templates for group openQA

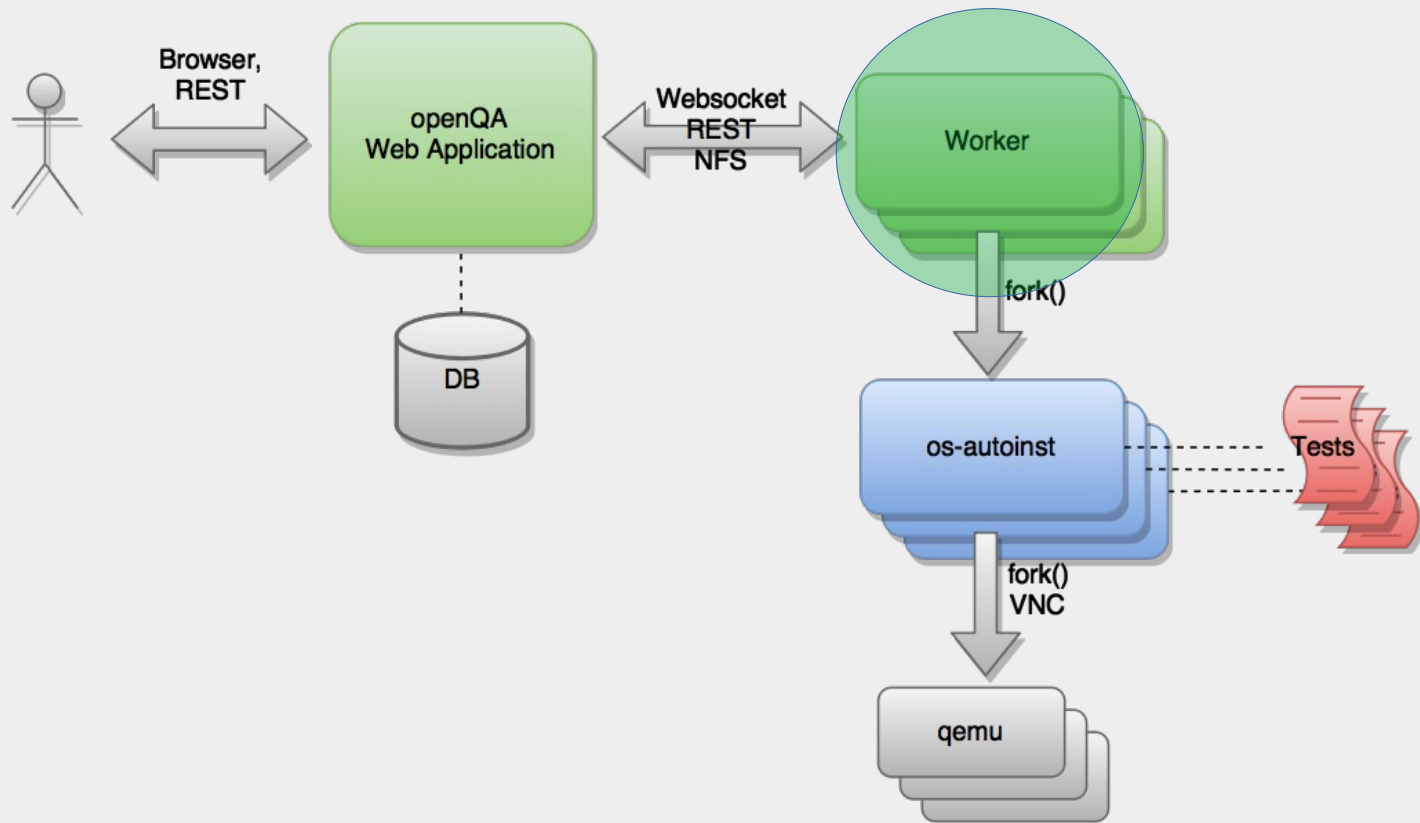
[Edit job group properties](#) [Manage all job groups](#)

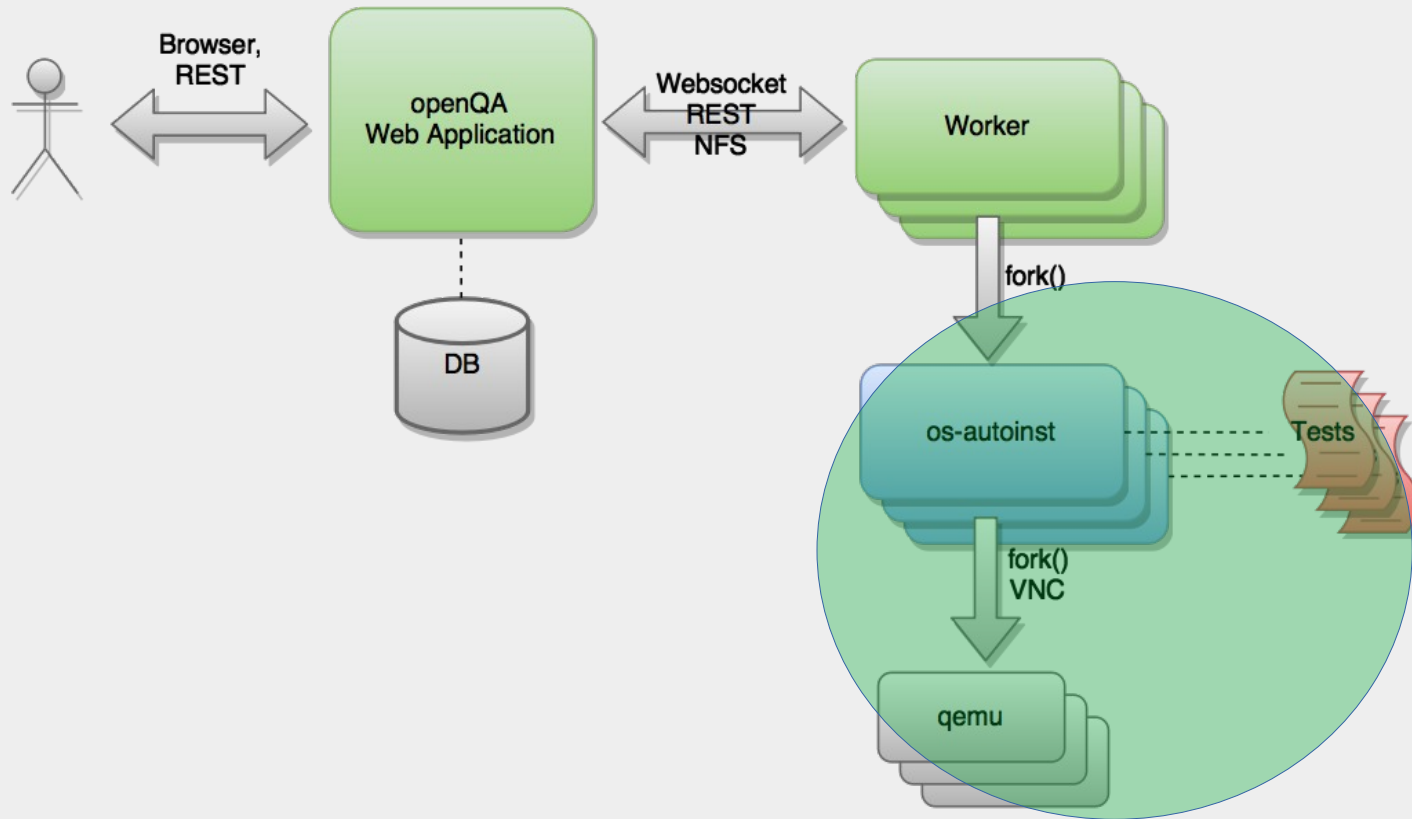
A job template is a combination of machine, product and test suite settings. These combination are used to produce actual jobs within a certain job group. Checkout the documentation for more details.

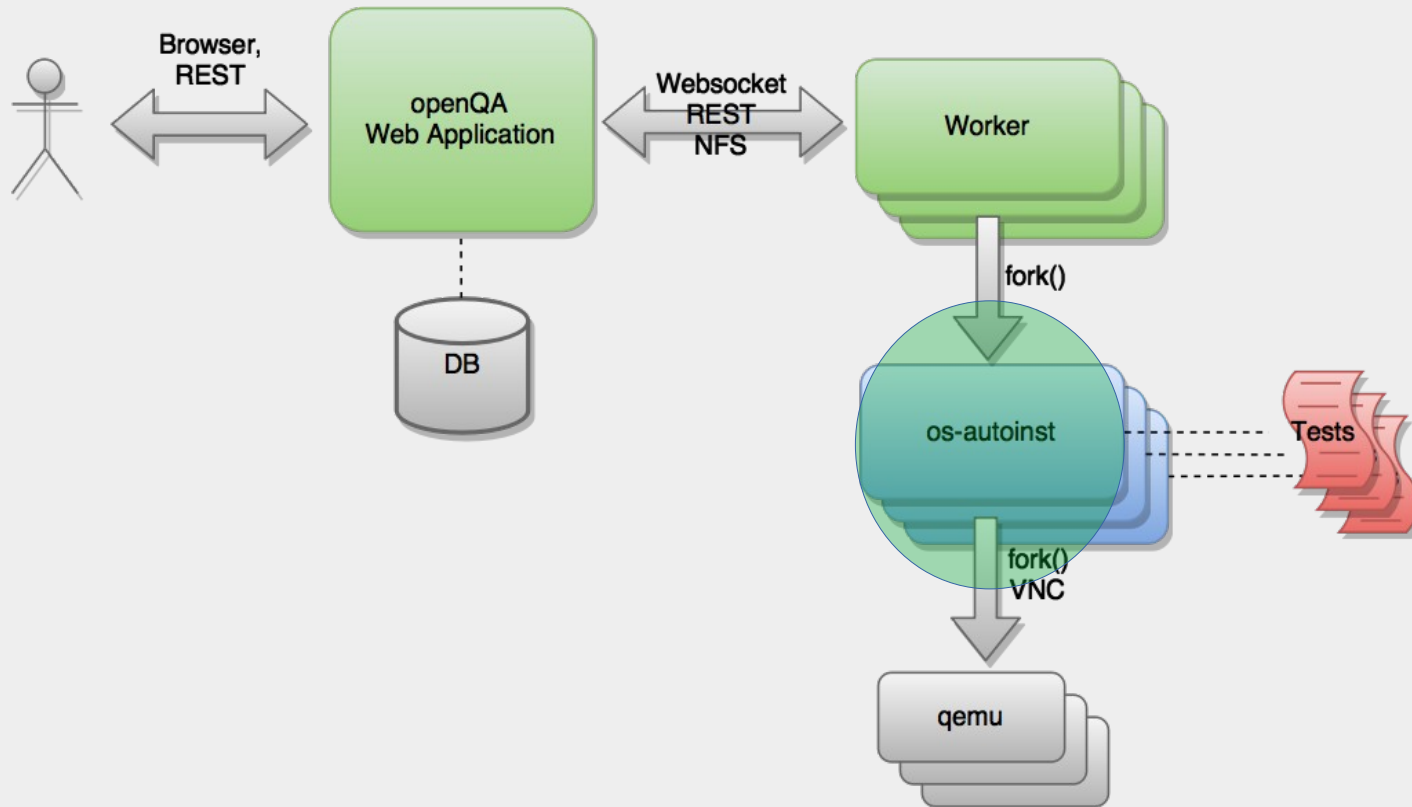
```
1 defaults:
2   x86_64:
3     machine: 64bit-2G
4     priority: 50
5     settings:
6       INSTALL: '1'
7       INSTALL_ONLY: '1'
8       QEMU_CPU: host
9       UPDATE: '0'
10
11 products:
12   openqa-*--dev-x86_64:
13     distri: openqa
14     flavor: dev
15     version: '*'
16 scenarios:
17   x86_64:
18     openqa-*--dev-x86_64:
19       - openqa_install+publish:
20         test suite: null
21         settings:
22           PUBLISH_HDD_1: 'opensuse-Tumbleweed-%ARCH%-%MACHINE%-%BUILD%.qcow2'
23           PUBLISH_PFLASH_VARS: 'opensuse-Tumbleweed-%ARCH%-%MACHINE%-%BUILD%-uefi-vars.qcow2'
24         description: >-
25           Maintainer: okurz@suse.de Test for installation of openQA itself.
26           To be used with "openqa" distri. Publishes an qcow2 image including the openQA installation
27           ready to run as an appliance.
28       - openqa_from_git:
29         test suite: null
```

We're loving YAML!

[Check YAML guide](#) [Check extended syntax](#) [Check changes](#)









# os-autoinst

(aka isotovideo)

## Is the test Driver

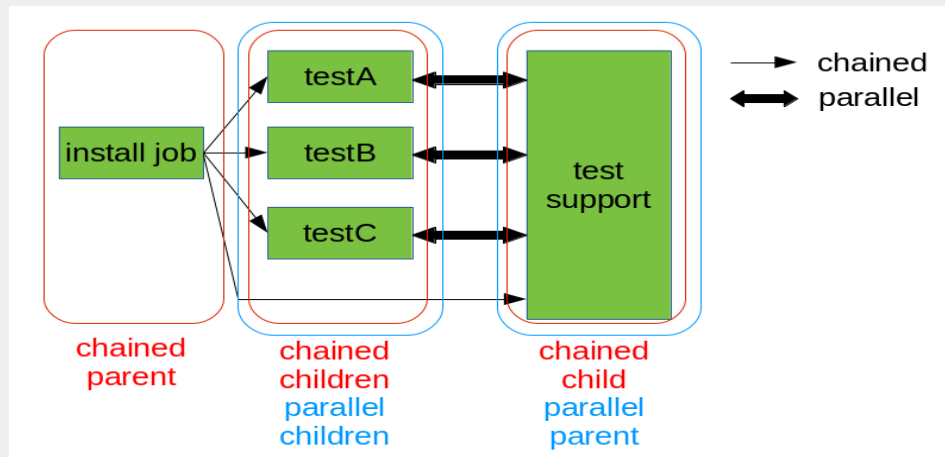
Sets up the environment  
Reports results back to the Web UI

## Written in Perl (naturally)!

A lot of Mojolicious love  
But also a lot of old code :(

## Provides Locking API

Parallel  
Chained  
N+1 nodes



# os-autoinst

(aka isotovideo)

## Is the test Driver

Sets up the environment

Reports results back to the Web UI

## Written in Perl (naturally)!

A lot of Mojolicious love

But also a lot of old code :(

## Provides Locking API

Parallel

Chained

N+1 nodes

Specify machine explicitly  
Implicitly inherit machines from parent  
Conflicting machines prevent inheritance from parent  
Implicitly creating a dependency on same machine  
Notes regarding directly chained dependencies  
Worker requirements  
Examples  
Test synchronization and locking API  
Support Server based tests  
Preparing the supportserver  
Using the supportserver  
Using text consoles and the serial terminal  
Using a serial terminal  
Sending new lines and continuation characters  
Sending signals - ctrl-c and ctrl-d  
The virtio serial terminal implementation  
Test Development tricks  
Trigger new tests by modifying settings from existing test runs  
Backend variables for

```
assert_screen 'login', 300;

# unlock by creating the lock
mutex_create 'ftp_service_ready';

# wait until all children finish
wait_for_children;
}

# On child we wait for ftp server to be ready
sub run {
    # wait until ftp service is ready
    # performs mutex lock & unlock internally
    mutex_wait 'ftp_service_ready';

    # connect to ftp and start downloading
    script_run 'ftp parent.job.ip';
    script_run 'get random_file';
}

# Mutexes can be used also for granting exclusive access to resource
# Example on child when only one job should access ftp at time
sub run {
    # wait until ftp service is ready
    mutex_lock 'ftp_service_ready';

    # Perform operation with exclusive access
    script_run 'ftp parent.job.ip';
    script_run 'put only_i_am_here';
    script_run 'bye';

    # Allow other jobs to connect afterwards
    mutex_unlock 'ftp_service_ready';
}
```

Sometimes it is useful to wait for a certain action from the child or sibling job rather than the parent. In this

# os-autoinst (aka isotovideo)

## Provides DSL via testapi.pm

Provides *type\_string*

Provides *assert\_screen*

Provides *wait\_serial*

Provides *select\_console*

... and much more

```
send_key
hold_key
release_key
send_key_until_needlematch
type_string
type_password
mouse support
mouse_set
mouse_click
mouse_dclick
mouse_tclick
mouse_hide
multi console support
add_console
select_console
console
reset_consoles
current_console
audio support
start_audiocapture
assert_recorded_sound
check_recorded_sound
miscellaneous
power
check_shutdown
assert_shutdown
eject_cd
save_memory_dump
save_storage_drives
freeze_vm
resume_vm
parse_unit_log
parse_extra_log
```

### keyboard support

#### send\_key

```
send_key($key [, wait_screen_change => $wait_screen_change]);
```

Send one *\$key* to SUT keyboard input. Waits for the screen to change when *\$wait\_screen\_change* is true.

Special characters naming:

```
'esc', 'down', 'right', 'up', 'left', 'equal', 'spc', 'minus', 'shift', 'ctrl'
'caps', 'meta', 'alt', 'ret', 'tab', 'backspace', 'end', 'delete', 'home', 'insert'
'pgup', 'pgdn', 'sysrq', 'super'
```

#### hold\_key

```
hold_key($key);
```

Hold one *\$key* until release it

#### release\_key

```
release_key($key);
```

Release one *\$key* which is kept holding

#### send\_key\_until\_needlematch

```
send_key_until_needlematch($tag, $key [, $counter, $timeout]);
```

Send specific key until needle with *\$tag* is not matched or *\$counter* is 0. *\$tag* can be string or ARRAYREF ( ['tag1', 'tag2'] ) Default counter is 20 steps, default timeout is 1s

# os-autoinst (aka isotovideo)

## Provides DSL via testapi.pm

Provides *type\_string*

Provides *assert\_screen*

Provides *wait\_serial*

Provides *select\_console*

... and much more

### Table of Contents

- Introduction
- Internal
  - init
  - set\_distribution
- video output handling
  - save\_screenshot
  - record\_info
  - record\_soft\_failure
  - assert\_screen
  - check\_screen
  - match\_has\_tag
  - assert\_and\_click
  - click\_lastname
  - assert\_and\_dclick
  - wait\_screen\_change
  - assert\_screen\_change
  - wait\_still\_screen
  - assert\_still\_screen
- test variable access
  - get\_var
  - get\_required\_var
  - set\_var
  - check\_var
  - get\_var\_array
  - check\_var\_array
- script execution helpers
  - is\_serial\_terminal
  - wait\_serial
  - x11\_start\_program
  - assert\_script\_run

### record\_info

```
record_info($title, $output [, result => $result] [, resultname => $resultname]);
```

#### Example:

```
record_info('workaround', "we know what we are doing");
```

Record a generic step result on the current test modules. This is meant for informational purposes to be interpreted by a displaying system. For example openQA can show a info box as part of the job results details. Use this instead of `record_soft_failure` for example when you do not want to mark the job as a softfail. The optional value `$result` can be 'ok' (default), 'fail', 'softfail'. `$resultname` can be specified for the additional name tag on the result file.

### force\_soft\_failure

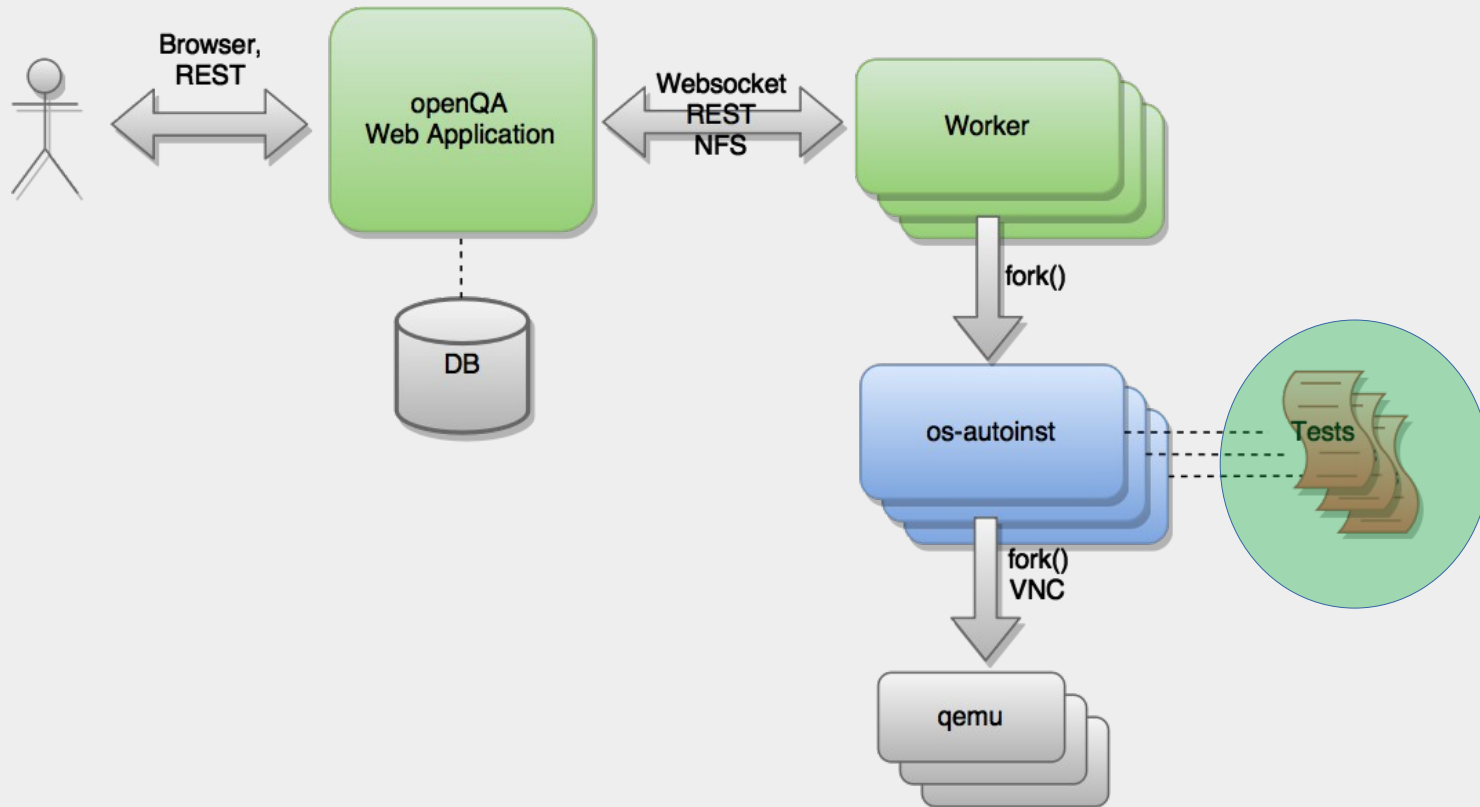
```
force_soft_failure([$reason]);
```

Similar to `record_soft_failure` but can be used to override the test module status to softfail from a `post_fail_hook` if the module would be set to fail otherwise. This can be used for easier tracking of known issues without needing to handle failed tests a lot.

### assert\_screen

```
assert_screen($mustmatch [, [$timeout] | [timeout => $timeout]] [, no_wait => $no_wait]);
```

Wait for needle with tag `$mustmatch` to appear on SUT screen. `$mustmatch` can be string or ARRAYREF of string ( [ 'tag1', 'tag2' ] ). The maximum waiting time is defined by `$timeout`. It is recommended to use a value lower than the default timeout only when explicitly needed. `assert_screen` is not very suitable for checking performance expectations. Under the normal circumstance of the screen being shown this does not



- X86
- ARM64
- PowerPC
  - PowerKVM
  - PowerVM
- Libvirt
- Mainframe (s390)
- Also Baremetal :)

## Test result overview

Overall Summary of opensuse Tumbleweed build 20200620

Passed: 390 Incomplete: 7 Soft-Failed: 97 Failed: 157 None: 17

Filter ⓘ

no filter present, click to toggle filter form

Flavor: DVD

Test	aarch64	i586	ppc64	ppc64le	s390x	x86_64
RAID0	-	-	☞ ●	☞ ●	-	-
RAID0_gpt	-	-	-	-	☞ ●	☞ ●
RAID0_gpt_uefi	☞ ●	-	-	-	-	-
RAID1	-	-	☞ ●	☞ ●	-	-
RAID10	-	-	-	☞ ●	-	-
RAID10_gpt	-	-	-	-	-	☞ ●
RAID10_gpt_uefi	☞ ●	-	-	-	-	-
RAID1_gpt	-	-	-	-	-	☞ ●
RAID1_gpt_uefi	☞ ●	-	-	-	-	-
RAID5	-	-	☞ ● first_boot	☞ ●	-	-



**THAT'S PRETTY METAL**

# Talking about Mainframe

```
136 my $sequence = <<"EO_fricken_boot_parms";
137 ${parmfile_with_Newline_s}
138 ENTER
139 ENTER
140 EO_fricken_boot_parms
141
142 # arbitrary number of retries
143 my $max_retries = 7;
144 for (1 .. $max_retries) {
145     eval {
146         # ensure that we are in cms mode before executing qaboot
147         $$3270->sequence_3270("String(\"#cp i cms\")", "ENTER", "ENTER", "ENTER", "ENTER",);
148         $r = $$3270->expect_3270(output_delim => qr/CMS/, timeout => 20);
149         $$3270->sequence_3270("String(\"qaboot $repo_host $dir_with_suse_ins\")", "ENTER", "Wait(Inpu
150         # wait for qaboot dumping us into xedit. If this fails, probably the
151         # download of kernel or initrd timed out and we retry
152         $r = $$3270->expect_3270(buffer_ready => qr/X E D I T/, timeout => 60);
153     };
154     last unless ($@);
155     diag "s3270 sequence failed: $@";
156     diag "Maybe the network is busy. Retry: $_ of $max_retries";
157 }
158 die "Download of Kernel or Initrd took too long (with retries)" unless $r;
159
160 $$3270->sequence_3270(qw( String(INPUT) ENTER ));
161
162 $r = $$3270->expect_3270(buffer_ready => qr/Input-mode/);
163
164 # can't use qw() because of space in commands...
165 $$3270->sequence_3270(split /\n/, $sequence);
```



# Say you want interact with Azure

```
261
262 sub stop_instance
263 {
264     my ($self, $instance) = @_;
265     # We assume that the instance_id on azure is actually the name
266     # which is equal to the resource group
267     # TODO maybe we need to change the azure.tf file to retrieve the id instead of the name
268     my $id = $instance->instance_id();
269     my $attempts = 60;
270
271     die('Outdated instance object') if ($self->get_ip_from_instance($instance) ne $instance->public_ip);
272
273     assert_script_run("az vm stop --ids '$id'", quiet => 1);
274     while ($self->get_state_from_instance($instance) ne 'stopped' && $attempts-- > 0) {
275         sleep 5;
276     }
277     die("Failed to stop instance $id") unless ($attempts > 0);
278 }
279
280 sub start_instance
281 {
282     my ($self, $instance, %args) = @_;
283     my $id = $instance->instance_id();
284
285     die("Try to start a running instance") if ($self->get_state_from_instance($instance) ne 'stopped');
286
287     assert_script_run("az vm start --ids '$id'", quiet => 1);
288     $instance->public_ip($self->get_ip_from_instance($instance));
289 }
290
```

# Say you want interact with Amazon Cloud

```
195
196 sub stop_instance
197 {
198     my ($self, $instance) = @_;
199     my $instance_id = $instance->instance_id();
200     my $attempts    = 60;
201
202     die("Outdated instance object") if ($instance->public_ip ne $self->get_ip_from_instance($instance));
203
204     assert_script_run('aws ec2 stop-instances --instance-ids ' . $instance_id, quiet => 1);
205
206     while ($self->get_state_from_instance($instance) ne 'stopped' && $attempts-- > 0) {
207         sleep 5;
208     }
209     die("Failed to stop instance $instance_id") unless ($attempts > 0);
210 }
211
212 sub start_instance
213 {
214     my ($self, $instance, %args) = @_;
215     my $attempts    = 60;
216     my $instance_id = $instance->instance_id();
217
218     my $i_desc = $self->describe_instance($instance);
219     die("Try to start a running instance") if ($i_desc->{State}->{Name} ne 'stopped');
220
221     assert_script_run("aws ec2 start-instances --instance-ids $instance_id", quiet => 1);
222     sleep 1; # give some time to update public ip
223     my $public_ip;
224     while (!defined($public_ip) && $attempts-- > 0) {
```

# Say you have your own test suite

```
1 use base "basetest";
2 use strict;
3 use TestAPI;
4
5 sub run() {
6     my $branch = "master";
7     my $tests_packages = "chromedriver xorg-x11-fonts";
8     if (get_var('VERSION') != "Unstable") {
9         $branch = get_var('VERSION');
10        $tests_packages = "phantomjs" unless get_var('VERSION') == "2.10";
11    }
12
13    my $ruby_version = "2.5";
14
15    assert_script_run("zypper -vv -n --gpg-auto-import-keys in --force-resolution --no-recommends $tests_packages");
16    assert_script_run("git clone --single-branch --branch $branch --depth 1 https://github.com/opensUSE/openSUSE-TestAPI");
17    assert_script_run("cd /tmp/open-build-service/dist/t/");
18    assert_script_run("set -o pipefail; prove -v *-check_required_services.ts | tee /tmp/check_required_services.txt");
19    assert_script_run("bundle.ruby$ruby_version install", 600);
20    assert_script_run("set -o pipefail; bundle.ruby$ruby_version exec rspec --format documentation | tee /tmp/rspec_tests.txt");
21    save_screenshot;
22    upload_logs("/tmp/check_required_services_tests.txt");
23    upload_logs("/tmp/rspec_tests.txt");
24 }
25
26 sub test_flags() {
27     return {important => 1};
28 }
29
30 sub post_fail_hook {
```

I want to run  
a script and  
capture the  
output!

```
501 # Retrying workaround
502 my $ret;
503 my $search_conflicts = 'awk \'BEGIN {print "Processing conflicts - ",NR; group=0}
504 /Solvrrun finished with an ERROR/,/statistics/{ print group"|",
505 $0; if ($0 ~ /statistics/ ){ print "EOL"; group++ }; }\'\'
506 /var/log/zypper.log
507 \';
508
509 for (1 .. 5) {
510     $ret = script_run("zypper -n $command $printer; ( exit \${PIPESTATUS[0]} )", $timeout);
511     die "zypper did not finish in $timeout seconds" unless defined($ret);
512     if ($ret == 4) {
513         if (script_run('grep "Error code.*502" /var/log/zypper.log') == 0) {
514             die 'According to bsc#1070851 zypper should automatically retry internally. Bugfix missing
515         } elseif (my $conflicts = script_output($search_conflicts)) {
516             record_info("Conflict", $conflicts, result => 'fail');
517             diag "Package conflicts found, not retrying anymore" if $conflicts;
518             last;
519         }
520     }
521     next unless get_var('FLAVOR', '') =~ /(Updates|Incidents)$/;
522 }
523 if (get_var('FLAVOR', '') =~ /(Updates|Incidents)/ && ($ret == 4 || $ret == 8 || $ret == 105 ||
524     if (script_run('grep "Exiting on SIGPIPE" /var/log/zypper.log') == 0) {
525         record_soft_failure 'Zypper exiting on SIGPIPE received during package download bsc#11455
526     }
527     else {
528         record_soft_failure 'Retry due to network problems p00#52319';
529     }
530 }
531 next;
532 }
```

- <https://open.qa>
- <https://openqa.opensuse.org>
- <https://openqa.fedoraproject.org>
- <http://openqa.debian.net> \*
- <https://github.com/os-autoinst>

● Check it out!

Thank you.

