# openQA overview

Ondrej Holecek (oholecek@suse.com)

# Tests and jobs

# openQA tests

- main.pm

```
use autotest;
if (get_var('TEST')) {
  loadtest 'test.pm';
}
```

- tests/test.pm

```
use base 'basetest';
use testapi;
if (get_var('TEST_DESKTOP')) {
  assert_screen('generic_desktop', 200);
}
else {
  wait_serial('login:');
}
```

# openQA tests (cont'd)

- Test code
  - test loader (`main.pm`)
  - test code (under `/tests`)

- Needles
  - "screenshots with benefits"

- Test API
  - `testapi` imports
  - `basetest`, `opensusebasetest`, ... base objects
  - other APIs (`mmapi`, `lockapi`, ...)

# Test vs Job

- Test
    - test code and needles
    - stateless
    - static

# Test vs Job

- Test
  - test code and needles
  - stateless
  - static
- Job
  - job is to test as process is to code
  - influenced by test variables
  - stateful (scheduled|running|done)
  - `job = test_code($test_variables)` (figuratively)

# Running openQA tests

- 'iso centric' approach
  - create and schedule all jobs from matched media by single command
  - openQA default work flow

- 'job centric' approach
  - create jobs by hand using API
  - useful when developing one test and iso would create tens of them
  - more initial work and long command line
  - does not handle job dependencies!

# Scheduling the test suite

- get new ISO and put it inside asset directory:

  ```
  /var/lib/openqa/factory/iso
  ```

- create test jobs

  ```
  openqa-client isos post                        \
         DISTRI=opensuse VERSION=tumbleweed  \
         FLAVOR=DVD ARCH=x86_64              \
         ISO=os.iso BUILD=20150427
  ```

  - DISTRI, VERSION, ARCH, FLAVOR are mandatory to match the ISO in media table
  - ISO is (*only!*) the filename of iso inside asset directory
  - output should be like

    ```
    {count => 29, ids => [411 .. 439] }
    ```

    - if count is zero, there were problems

# Scheduling individual tests

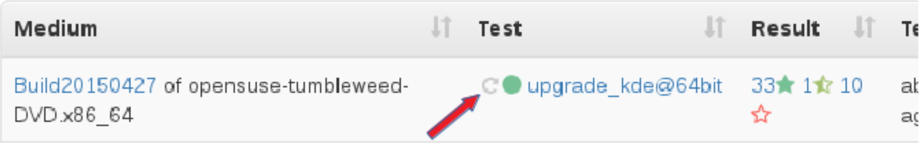
```
openqa-client jobs post                         \
        TEST=whatever BACKEND=qemu ARCH=x86_64  \
        BUILD=20150524 VERSION=Tumbleweed       \
        HDD_1=tumbleweed_kde.qcow2 YAST_HEAD=1  \
        ISO=openSUSE-TW-DVD-x86_64-20150524.iso \
        YAST_RUN_ONLY="console/yast2_i;x11/yast2_users"
```

- need to supply all required variables by hand
- create exactly one or none job
- output should be like

```
{ id => 7 }
```

# Restarting (cloning) jobs

- individual jobs can be restarted
  - technically jobs are duplicated to preserve old results
- using WebUI



| Medium | Test | Result | Te |
|---|---|---|---|
| Build20150427 of opensuse-tumbleweed-DVD.x86_64 | ● upgrade_kde@64bit | 33★ 1★ 10<br>☆ | ab<br>ag |

- using openQA API (e.g. via client script)

```
/usr/share/openqa/script/client jobs/$jobid/restart post
```

- using clone_job helper

```
/usr/share/openqa/script/clone_job.pl $jobid [$optional_variables]
```

# Writing and updating tests

# openQA test

- test code + needles + variables

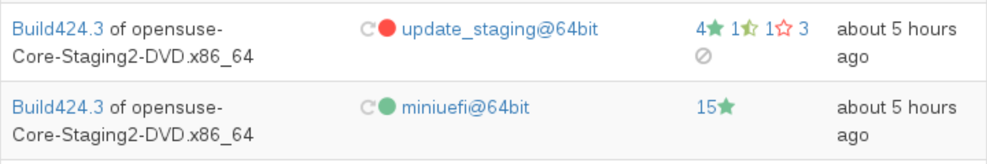
# openQA test

- test code + needles + variables
- test code
  - simulate usual user input (keyboard strokes, mouse actions)
  - and usual user output (visual comparison of results)
  - can be divided in modules
  - influenced by variables passed from scheduler

# openQA test

- test code + needles + variables
- test code
  - simulate usual user input (keyboard strokes, mouse actions)
  - and usual user output (visual comparison of results)
  - can be divided in modules
  - influenced by variables passed from scheduler
- needle
  - expected visual output of SUT
  - screenshot
  - metadata
    - regions to match
    - tags

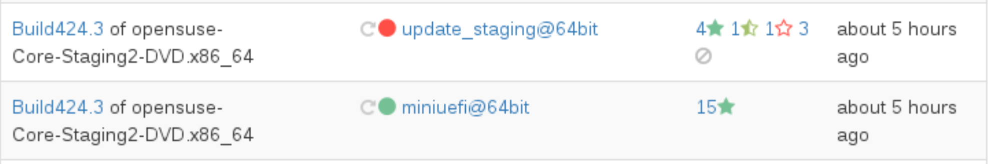
# Interpreting results



| | | | |
|---|---|---|---|
| Build424.3 of opensuse-Core-Staging2-DVD.x86_64 | ↻ 🔴 update_staging@64bit | 4★ 1☆ 1☆ 3 ⊘ | about 5 hours ago |
| Build424.3 of opensuse-Core-Staging2-DVD.x86_64 | ↻ 🟢 miniuefi@64bit | 15★ | about 5 hours ago |

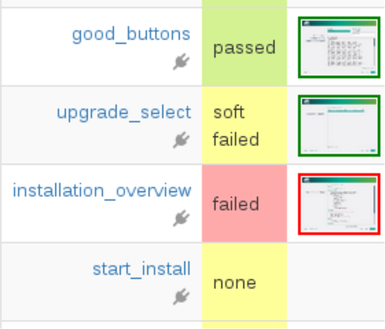
- Job results
  - ok, soft-failed or fail

# Interpreting results



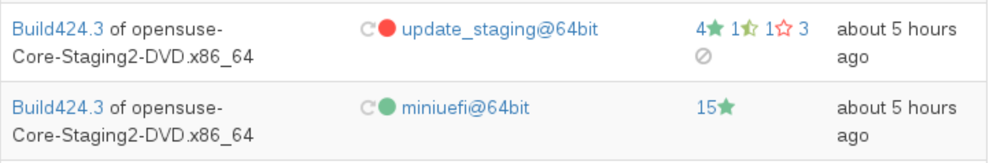
| | | | |
|---|---|---|---|
| Build424.3 of opensuse-Core-Staging2-DVD.x86_64 | ● update_staging@64bit | 4★ 1★ 1☆ 3 ⊘ | about 5 hours ago |
| Build424.3 of opensuse-Core-Staging2-DVD.x86_64 | ● miniuefi@64bit | 15★ | about 5 hours ago |

- Job results
  - ok, soft-failed or fail
- Job module results
  - passed, failed, soft failed, none, running, …



| | | |
|---|---|---|
| good_buttons | passed | |
| upgrade_select | soft failed | |
| installation_overview | failed | |
| start_install | none | |

# Interpreting results



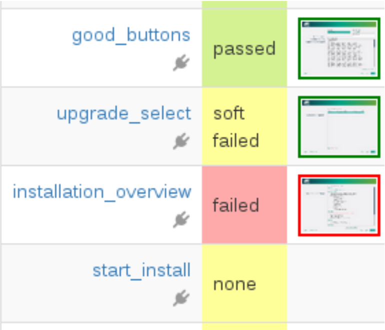
| | | | |
|---|---|---|---|
| Build424.3 of opensuse-Core-Staging2-DVD.x86_64 | C ● update_staging@64bit | 4★ 1★ 1☆ 3 ⊘ | about 5 hours ago |
| Build424.3 of opensuse-Core-Staging2-DVD.x86_64 | C ● miniuefi@64bit | 15★ | about 5 hours ago |

- Job results
  - ok, soft-failed or fail
- Job module results
  - passed, failed, soft failed, none, running, …
  - job module flags influence overall result

! - result is fail if module failed

🔌 - result is fail and job is stopped if module failed

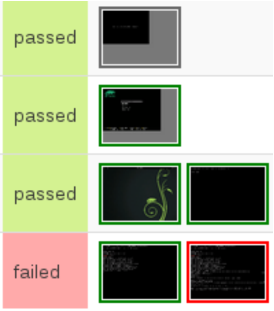
⚓ - save VM snapshot if module passed



| | | |
|---|---|---|
| good_buttons 🔌 | passed | ▢ |
| upgrade_select 🔌 | soft failed | ▢ |
| installation_overview 🔌 | failed | ▢ |
| start_install 🔌 | none | |

# Needle matching

- matching result
    - match indicated by green border
    - mismatch indicated by red border
    - no matching done indicated by grey border



passed

passed

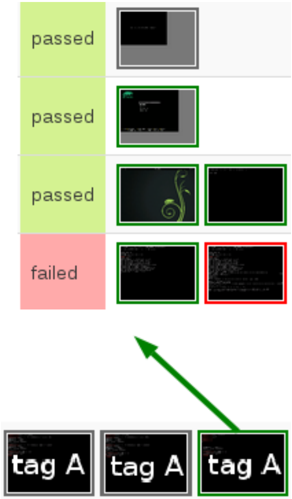
passed

failed

# Needle matching

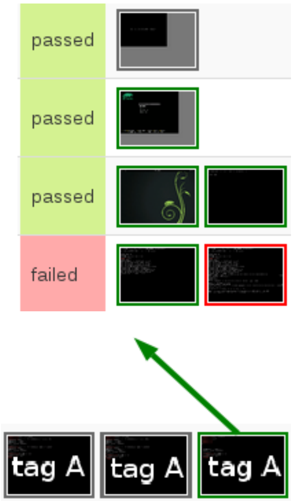
- matching result
  - match indicated by green border
  - mismatch indicated by red border
  - no matching done indicated by grey border

- tag
  - one needle can have multiple tags
  - multiple needles can have the same tag
  - allow changing needle without changing test code



# Needle matching

- matching result
  - match indicated by green border
  - mismatch indicated by red border
  - no matching done indicated by grey border
- tag
  - one needle can have multiple tags
  - multiple needles can have the same tag
  - allow changing needle without changing test code
- matching area
  - area on the original screenshot
  - matched against current SUT screen
    - initially at the original location
    - if not found, increase search margin until full screen



# Test structure

```
# /var/lib/openqa/tests/opensuse

data/
lib/
tests/
needles/
main.pm
```

- use data/ for test data
- everything in lib/ is imported by test engine thus available to all tests
- tests/ is where our test modules are
- entry point `main.pm`

  ```
  use autotest;
  autotest::loadtest($testfilename);
  ```

  - $testfilename relative to $testdir/tests
  - imported by test engine before(!) SUT is running!

# Test loader - main.pm

```
use strict;
use autotest;

loadtest "bootloader.pm";
loadtest "installation/license.pm";
loadtest "installation/partitioning.pm";
if ( defined( get_var("RAIDLEVEL") ) ) {
    loadtest "installation/partitioning_raid.pm";
}
if ( get_var("TOGGLEHOME") ) {
    loadtest "installation/partitioning_togglehome.pm";
}
loadtest "installation/partitioning_finish.pm";

...
```

- can access (get and set) jobs variables
- use `loadtest` to load test modules
- SUT is not yet running, DO NOT call test API functions

# Test module - tests/test.pm

```
use base "installbasetest";
use strict;
use testapi;

sub run() {
    # wait booted
    assert_screen 'generic-desktop', 200;

    x11_start_program('xterm');
    become_root;

    type_string "PS1=\$\n";    # set constant shell promt
    sleep 1;

    # Disable console screensaver
    script_run("setterm -blank 0");
}

1;
```

# Test modules

- inherits from one of base tests

  ```
  use base "opensusebasetest";
  ```

  - y2logstep, installbasetest, ...
- mandatory subroutine

  ```
  sub run() {
    my $self = shift;
    ...
  }
  ```

- optional test_flags

  ```
  sub test_flags {
    return { 'fatal' => 1, 'important' => 1, 'milestone' => 0 };
  }
  ```

# Test API

- part of os-autoinst package
  - https://github.com/os-autoinst/os-autoinst/blob/master/testapi.pm
  - `/usr/lib/os-autoinst/testapi.pm`

```
use testapi;
```

- variable management
  - get_var, set_var, check_var
- keyboard and mouse
  - send_key, type_string, mouse_click, mouse_set, mouse_hide
- script control
  - script_run, script_sudo, validate_script_output
- serial output
  - wait_serial
- display control
  - assert_screen, check_screen

# Reporting results

- two outcomes
  - ok
  - fail

  ```
  sub run() {
    my $self = shift;
    $self->result('ok');
  }
  ```

- some API calls record that automatically
  - assert_screen, wait_serial, validate_script_output, ...
- severity of outcome influenced by test_flags
- for warnings use record_soft_failure

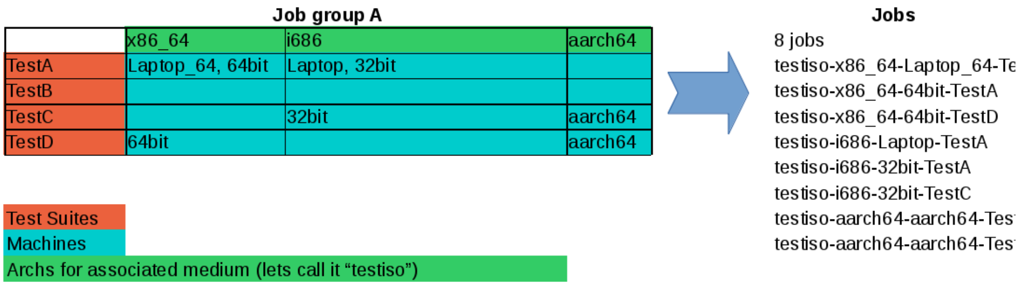
  ```
  use testapi;
  sub run() {
    my $self = shift;
    record_soft_failure;
  }
  ```

# openQA variables

- simple key=value (string=string)
- divided in logical categories
  - Machines
  - Test suites
  - Medium types
- and their relation matrix
  - Job groups

# openQA variables

- simple key=value (string=string)
- divided in logical categories
  - Machines
  - Test suites
  - Medium types
- and their relation matrix
  - Job groups



**Job group A**

| | x86_64 | i686 | aarch64 |
|---|---|---|---|
| TestA | Laptop_64, 64bit | Laptop, 32bit | |
| TestB | | | |
| TestC | | 32bit | aarch64 |
| TestD | 64bit | | aarch64 |

Test Suites

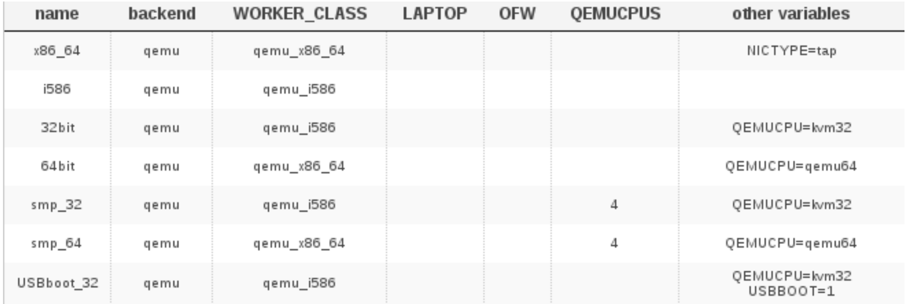
Machines

Archs for associated medium (lets call it “testiso”)

**Jobs**

8 jobs
testiso-x86_64-Laptop_64-Te
testiso-x86_64-64bit-TestA
testiso-x86_64-64bit-TestD
testiso-i686-Laptop-TestA
testiso-i686-32bit-TestA
testiso-i686-32bit-TestC
testiso-aarch64-aarch64-Tes
testiso-aarch64-aarch64-Tes

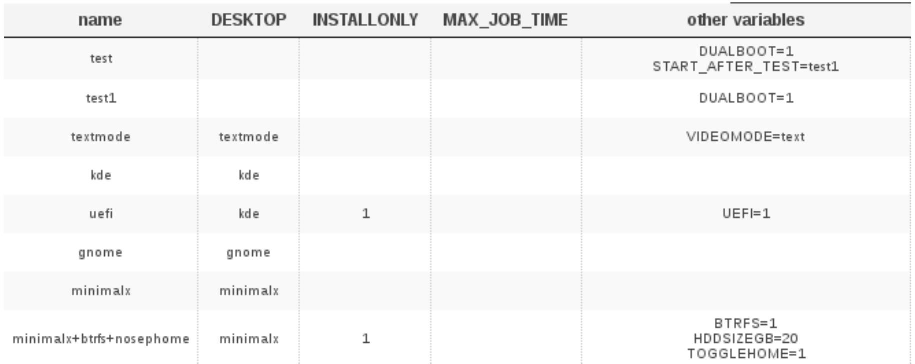
# Machines



| name | backend | WORKER_CLASS | LAPTOP | OFW | QEMUCPUS | other variables |
|---|---|---|---|---|---|---|
| x86_64 | qemu | qemu_x86_64 | | | | NICTYPE=tap |
| i586 | qemu | qemu_i586 | | | | |
| 32bit | qemu | qemu_i586 | | | | QEMUCPU=kvm32 |
| 64bit | qemu | qemu_x86_64 | | | | QEMUCPU=qemu64 |
| smp_32 | qemu | qemu_i586 | | | 4 | QEMUCPU=kvm32 |
| smp_64 | qemu | qemu_x86_64 | | | 4 | QEMUCPU=qemu64 |
| USBboot_32 | qemu | qemu_i586 | | | | QEMUCPU=kvm32<br>USBBOOT=1 |

- hypothetical SUT (System Under Test - the testing machine)
- backend specific variables
  - CPU model
  - RAM size
  - RAID
  - ...
  - https://github.com/os-autoinst/os-autoinst/blob/master/doc/backend_vars.asciidoc

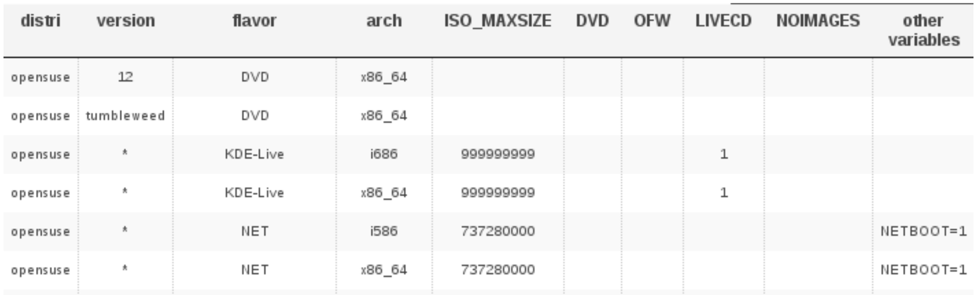
# Test suites



| name | DESKTOP | INSTALLONLY | MAX_JOB_TIME | other variables |
|---|---|---|---|---|
| test | | | | DUALBOOT=1<br>START_AFTER_TEST=test1 |
| test1 | | | | DUALBOOT=1 |
| textmode | textmode | | | VIDEOMODE=text |
| kde | kde | | | |
| uefi | kde | 1 | | UEFI=1 |
| gnome | gnome | | | |
| minimalx | minimalx | | | |
| minimalx+btrfs+nosephome | minimalx | 1 | | BTRFS=1<br>HDDSIZEGB=20<br>TOGGLEHOME=1 |

- this is what one would call a test
- test flow altering variables
  - desktop selection
  - install/upgrade path
  - ...

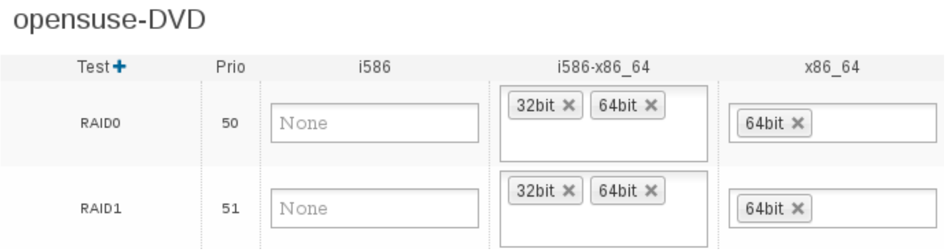
# Assets and Media



| distri | version | flavor | arch | ISO_MAXSIZE | DVD | OFW | LIVECD | NOIMAGES | other variables |
|---|---|---|---|---|---|---|---|---|---|
| opensuse | 12 | DVD | x86_64 | | | | | | |
| opensuse | tumbleweed | DVD | x86_64 | | | | | | |
| opensuse | * | KDE-Live | i686 | 999999999 | | | 1 | | |
| opensuse | * | KDE-Live | x86_64 | 999999999 | | | 1 | | |
| opensuse | * | NET | i586 | 737280000 | | | | | NETBOOT=1 |
| opensuse | * | NET | x86_64 | 737280000 | | | | | NETBOOT=1 |

- Assets are
  - ISOs, HDDs, AutoYast profiles, ...
  - everything test needs as a source
- Media are
  - assets matching templates
  - variables related to installation source

# Job Groups



opensuse-DVD

| Test ✚ | Prio | i586 | i586-x86_64 | x86_64 |
|---|---|---|---|---|
| RAID0 | 50 | None | 32bit ✕ 64bit ✕ | 64bit ✕ |
| RAID1 | 51 | None | 32bit ✕ 64bit ✕ | 64bit ✕ |

- organizing jobs to different categories
  - staging tests, tumbleweed tests, AArch64 tests, ...
- each group associated with some media

# Obtaining openSUSE tests

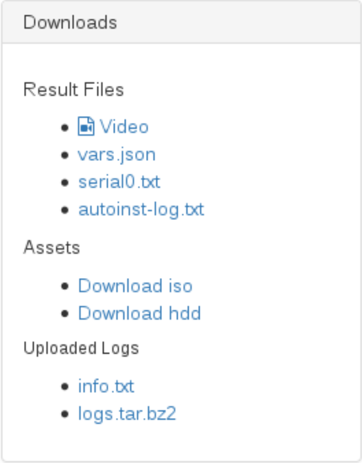
- 2 test directories - but use only one else you are asking for trouble!

```
> ll /var/lib/os-autoinst/
tests -> /var/lib/openqa/share/tests
```

- get the test suite
  - `/var/lib/openqa/tests/opensuse`
  - openSUSE and SLES
    - https://github.com/os-autoinst/os-autoinst-distri-opensuse
  - Fedora
    - https://bitbucket.org/rajcze/openqa_fedora/
- get needles
  - `/var/lib/openqa/tests/opensuse/needles`
  - openSUSE
    - https://github.com/os-autoinst/os-autoinst-needles-opensuse
  - SLES
    - https://gitlab.suse.de/openqa/os-autoinst-needles-sles

# Debugging

- debugging test
  - autoinst-log.txt
  - vars.json
  - serial0.txt
  - bmwqemu functions
    - diag, fctres, fctinfo, fctwarn



Downloads

Result Files

- Video
- vars.json
- serial0.txt
- autoinst-log.txt

Assets

- Download iso
- Download hdd

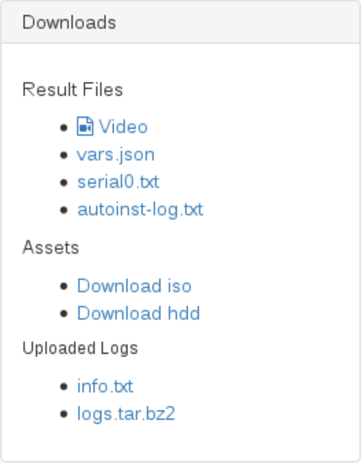
Uploaded Logs

- info.txt
- logs.tar.bz2

# Debugging

- debugging test
  - autoinst-log.txt
  - vars.json
  - serial0.txt
  - bmwqemu functions
    - diag, fctres, fctinfo, fctwarn
- debugging SUT
  - KEEPHDDS=1
  - MAKESNAPSHOTS=1
  - SKIPTO=$testmodule
  - run worker with `--no-cleanup` option

```
/usr/share/openqa/script/clone_job.pl \
                $jobid SKIPTO=$testmodule
```



Downloads

Result Files

- Video
- vars.json
- serial0.txt
- autoinst-log.txt

Assets

- Download iso
- Download hdd

Uploaded Logs

- info.txt
- logs.tar.bz2

# Test relations

- CHAINED
  - START_AFTER_TEST=testname
  - eliminating redundant jobs
  - reusing job assets
- PARALLEL
  - PARALLEL_WITH=testname
  - e.g. for HA testing
- can be used together

# Multi-Machine tests

- standard part of os-autoinst package

```
use mmapi;
use lockapi;
```

- locks for synchronization
    - mutex_create, mutex_lock, mutex_unlock
- related jobs status querying
    - get_children_by_state
- bare bones only

# Job assets

- use HDD image created by another job
- creating - `STORE_HDD_1` or `PUBLISH_HDD_1`
  - value of variable is asset name
  - STORE_HDD_1 for private HDD assets
  - PUBLISH_HDD_1 for public HDD assets
- usage - HDD_1
- use CHAINED relation to assure right order

```
testA:
  STORE_HDD_1='A.qcow2'
testB:
  HDD_1='A.qcow2'
  START_AFTER_TEST='testA'
```

# Resources

- openQA (https://github.com/os-autoinst/openQA)
- os-autoinst (https://github.com/os-autoinst/os-autoinst)
    - check backend docs (https://github.com/os-autoinst/os-autoinst/blob/master/doc/backend_vars.asciidoc)
- openSUSE tests (https://github.com/os-autoinst/os-autoinst-distri-opensuse)
- internal needles (https://gitlab.suse.de/openqa/os-autoinst-needles-sles)

# Installation and administration

# Installation

- OBS (https://build.opensuse.org/project/show/devel:openQA)
  - devel devel:openQA
  - stable devel:openQA:stable

# Installation

- OBS (https://build.opensuse.org/project/show/devel:openQA)
  - devel devel:openQA
  - stable devel:openQA:stable
- openQA WebUI
  - openqa-common, openqa (, apache2)

# Installation

- OBS (https://build.opensuse.org/project/show/devel:openQA)
  - devel devel:openQA
  - stable devel:openQA:stable
- openQA WebUI
  - openqa-common, openqa (, apache2)
- Database
  - sqlite3; postgresql

# Installation

- OBS (https://build.opensuse.org/project/show/devel:openQA)
  - devel devel:openQA
  - stable devel:openQA:stable
- openQA WebUI
  - openqa-common, openqa (, apache2)
- Database
  - sqlite3; postgresql
- openQA Worker
  - openqa-common, openqa-worker, os-autoinst

# Installation

- OBS (https://build.opensuse.org/project/show/devel:openQA)
  - devel devel:openQA
  - stable devel:openQA:stable
- openQA WebUI
  - openqa-common, openqa (, apache2)
- Database
  - sqlite3; postgresql
- openQA Worker
  - openqa-common, openqa-worker, os-autoinst
- Backends
  - qemu, kvm

# Installation - configuration

- check `/etc/openqa/database.ini`
  - [producion] section
  - default is SQLite

# Installation - configuration

- check `/etc/openqa/database.ini`
  - [producion] section
  - default is SQLite
- check `/etc/openqa/openqa.ini`
  - authentication method
  - logging

# Installation - configuration

- check `/etc/openqa/database.ini`
  - [producion] section
  - default is SQLite
- check `/etc/openqa/openqa.ini`
  - authentication method
  - logging
- almost ready to start that thing!

  ```
  systemctl start openqa-webui
  ```

  ```
  /usr/share/openqa/script/openqa daemon
  ```

- one thing left!

# User management - before running

- `/etc/openqa/openqa.ini` section [auth]
- OpenID
  - default authentication method
  - default provider - opensuse.org/openid

# User management - before running

- `/etc/openqa/openqa.ini` section [auth]
- OpenID
  - default authentication method
  - default provider - opensuse.org/openid
- iChain
  - present, but not tested in production

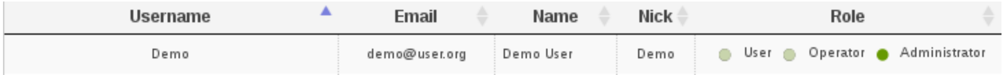
# User management - before running

- `/etc/openqa/openqa.ini` section [auth]
- OpenID
  - default authentication method
  - default provider - opensuse.org/openid
- iChain
  - present, but not tested in production
- Fake
  - !DANGER!
  - for development only

# User management - before running

- `/etc/openqa/openqa.ini` section [auth]
- OpenID
  - default authentication method
  - default provider - opensuse.org/openid
- iChain
  - present, but not tested in production
- Fake
  - !DANGER!
  - for development only
- now start that thing!

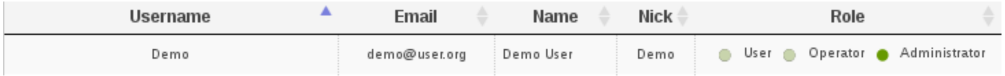
# User management - WebUI



| Username | Email | Name | Nick | Role |
|---|---|---|---|---|
| Demo | demo@user.org | Demo User | Demo | ○ User ○ Operator ● Administrator |

- first one to log in becomes administrator

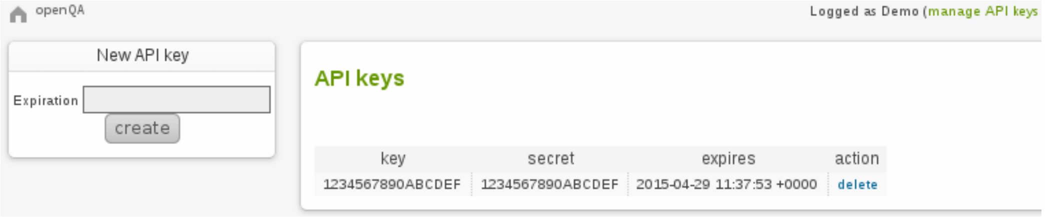
# User management - WebUI



| Username | Email | Name | Nick | Role |
|---|---|---|---|---|
| Demo | demo@user.org | Demo User | Demo | ○ User ○ Operator ● Administrator |

- first one to log in becomes administrator
- set authorizations to individual users
- operator and administrator levels
  - operator can restart (clone) tests and create needles
  - administrator has access to administration

# API keys



openQA

Logged as Demo (manage API keys

New API key

Expiration

create

API keys

| key | secret | expires | action |
|---|---|---|---|
| 1234567890ABCDEF | 1234567890ABCDEF | 2015-04-29 11:37:53 +0000 | delete |

- authentication for API access
- authorization level tied with user
- HMAC checking
  - API key
    - transferred in each authentication header
  - API secret
    - used as salt for message checksum

# Worker configuration

- /etc/openqa/client.conf
  - API key and secret goes here
- /etc/openqa/worker.ini
  - backend and openQA hostname
  - worker class override for special HW or configuration
- shared storage for remote worker

  ```
  /var/lib/openqa/share
  ```

  - up to admin, nfs usually
  - read only mount is sufficient
- start the worker

  ```
  systemctl start openqa-worker@1
  ```

  ```
  /usr/share/openqa/script/worker --instance 1 --verbose
  ```

EOF