SETTING UP OPENQA TESTING FOR GNOME

Sam Thursfield
FOSDEM 2023

ABOUT ME

ABOUT GNOME

GNOME is a graphical desktop environment with an open development model.

GNOME IS OLDER THAN ...

(2000)

ubuntu®

(2004)

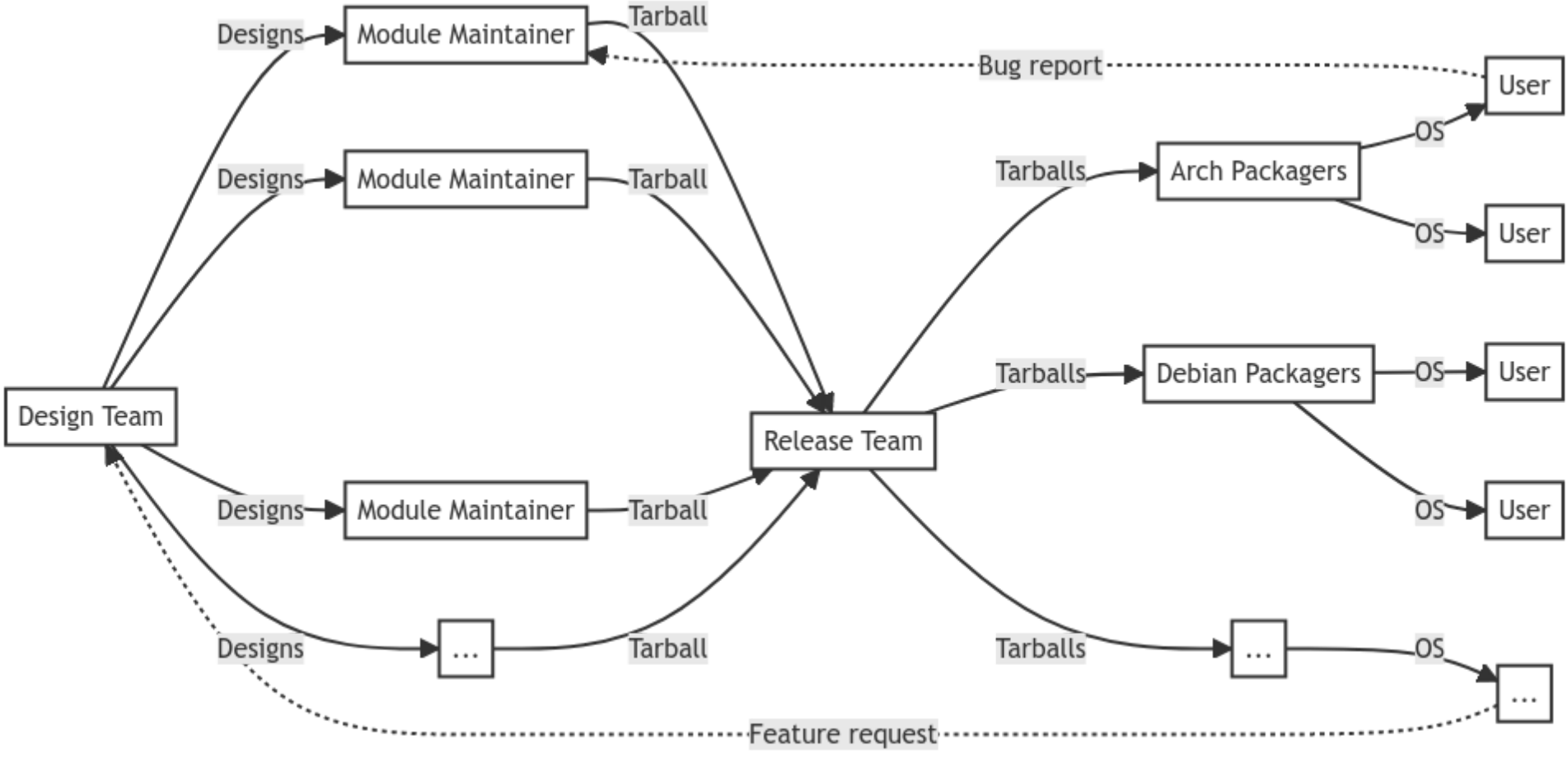
(2005)

(2005)

GNOME IS HARD TO TEST

- Designed as a whole
- Released as a kit of parts (200+ modules)

Who is responsible for integration testing?



Release process, 1999

👷 **Maintainer**: *"It works on my machine!"*

👩 **Release team**: *"It builds... ship it!"*

🧔 **Distributions**: *"A new upstream release - ship it!"*

👼 **Users**: Time to test if anything works...

Many "bugs" appear at integration time.

TIME PASSES

- New **build tools** (*jhbuild*, *Meson*, *BuildStream*, ...)
- New **collaboration tools** (*Git*, *Gitlab*, ...)
- **CI** becomes practical

Release process, 2023

👷 **Maintainer**: Review merge requests, check tests

👵 **Release team**: Update and check integration repo (gnome-build-meta)

👴 **Distributions**: Downstream regression testing

👩👱 **Users**: Friendly messages thanking volunteers for their hard work

There's still a large gap between 'main' branch and distro releases.

What if GNOME had its own distro built from 'main' branches?

GNOME OS

Part of the "Testable" initiative (*OSTree*, *GNOME Continuous*, *gnome-build-meta*, ...)

Goals:

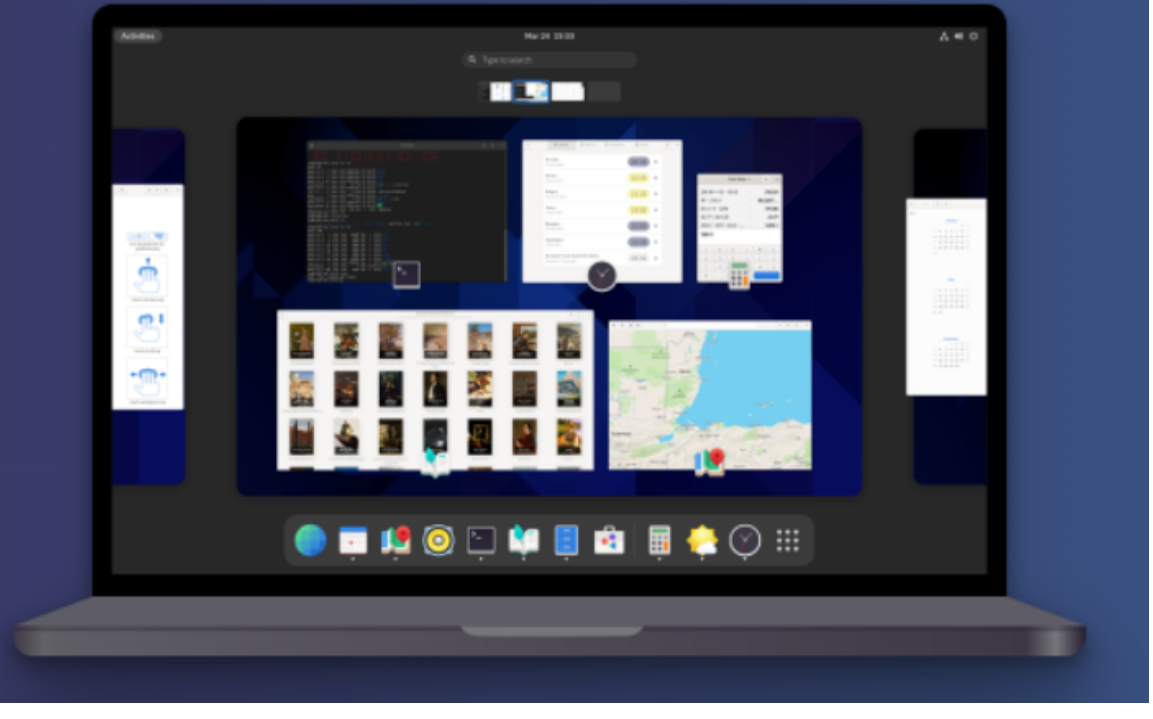
- Provide a "known good" full system integration
- Allow designers and developers to test in-progress changes
- Automated regression testing

Non-goals:

- Reliability
- Security updates, hardware enablement, user support

GNOME OS Nightly

Try the latest and greatest GNOME software
in a VM or on real hardware



https://os.gnome.org

GNOME OS

Only a few people test GNOME OS today.

Building an OS image takes **several hours**.

How can we catching complex regressions *before* releasing tarballs?

OpenQA

Initial commit in 2010.

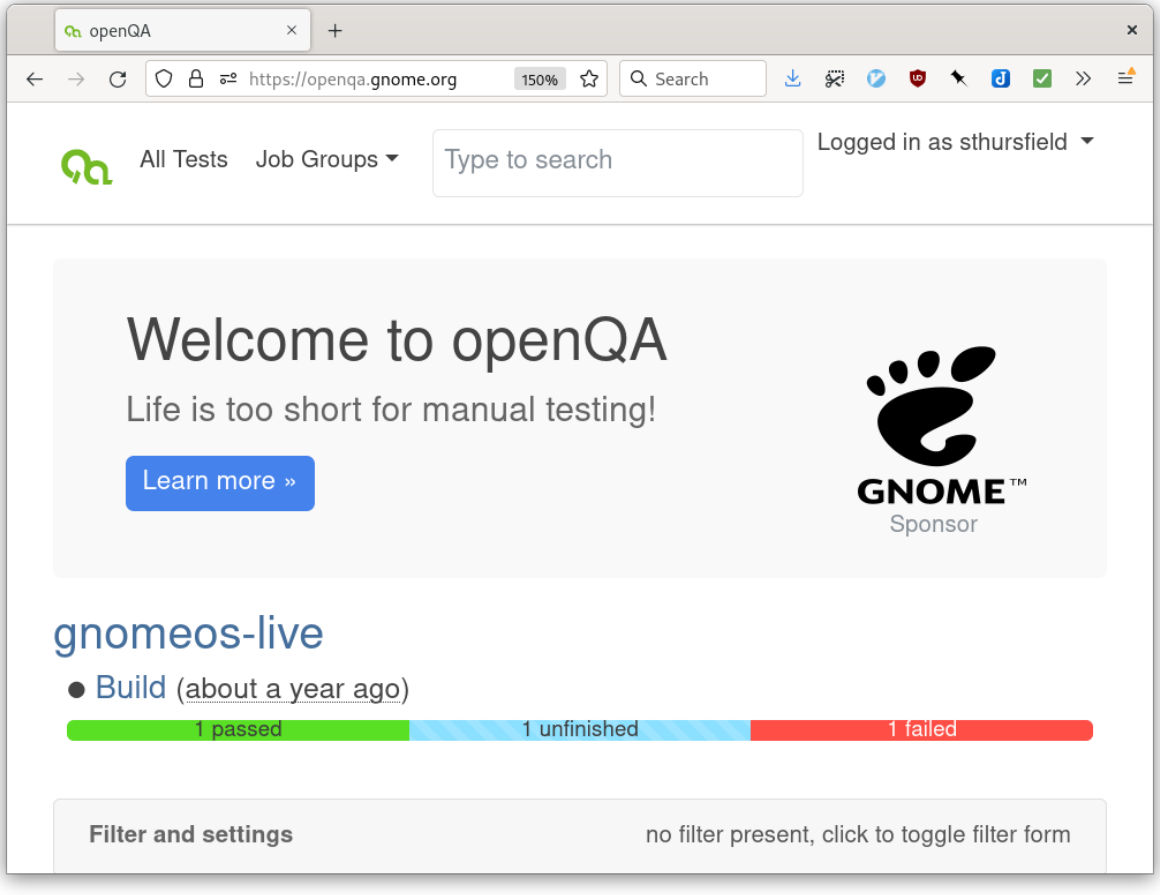
Components:

- **Web interface** (OpenQA)
- **Test driver** (os-autoinst)
- **Test library** (os-autoinst-distri-opensuse)

WHO USES OPENQA?

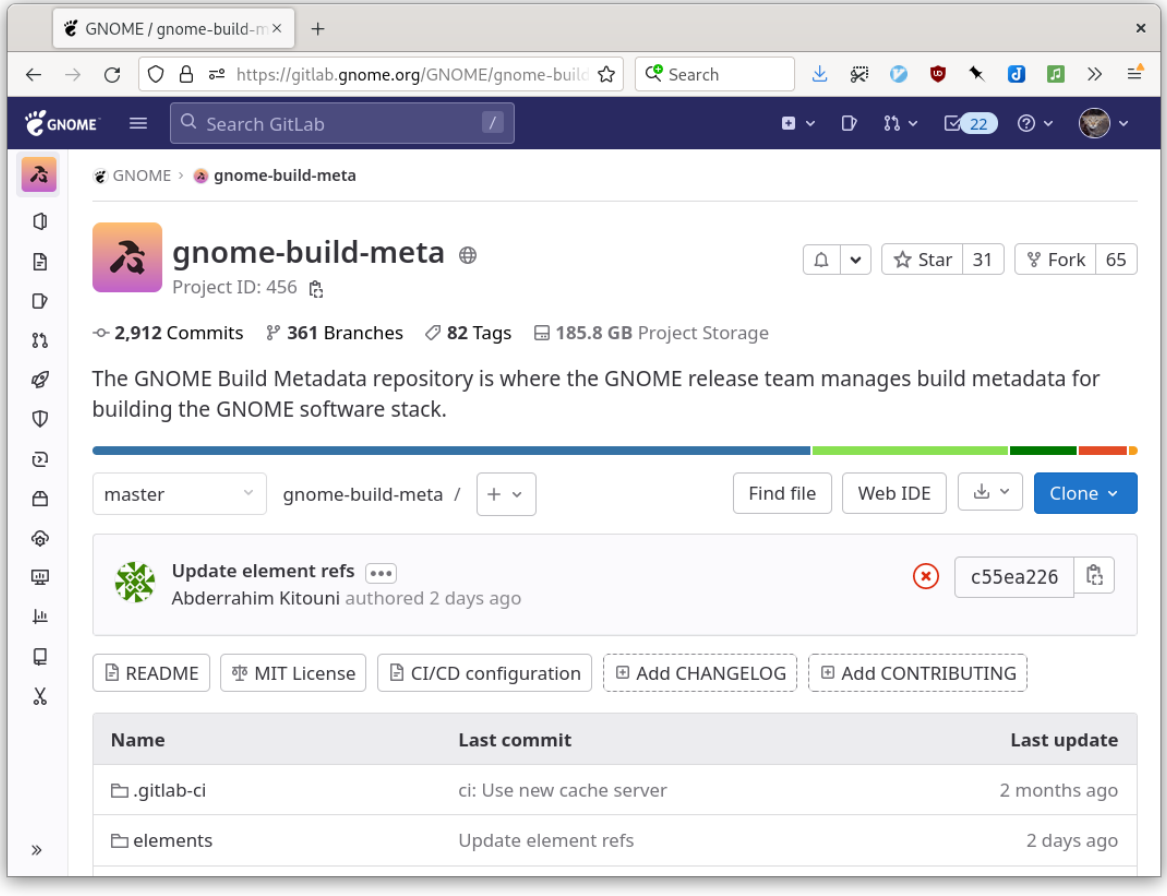
- OpenSUSE (https://openqa.opensuse.org/)
- Fedora (https://openqa.fedoraproject.org/)
- EuroLinux (CentOS derivative)
- Codethink (https://openqa.qa.codethink.co.uk/)
- Various Codethink clients in automotive industry
- ...you?

OpenQA: main page

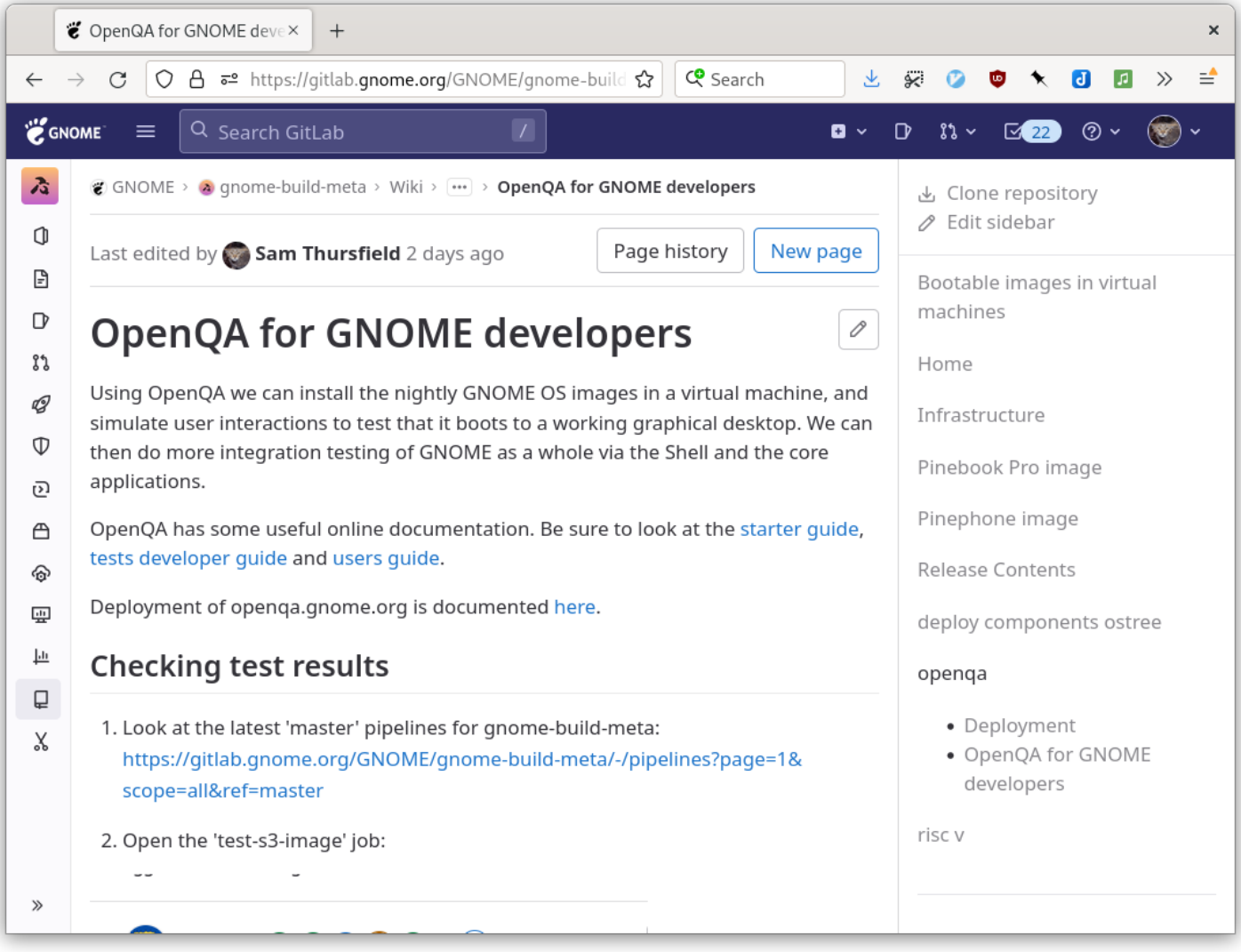


https://openqa.gnome.org

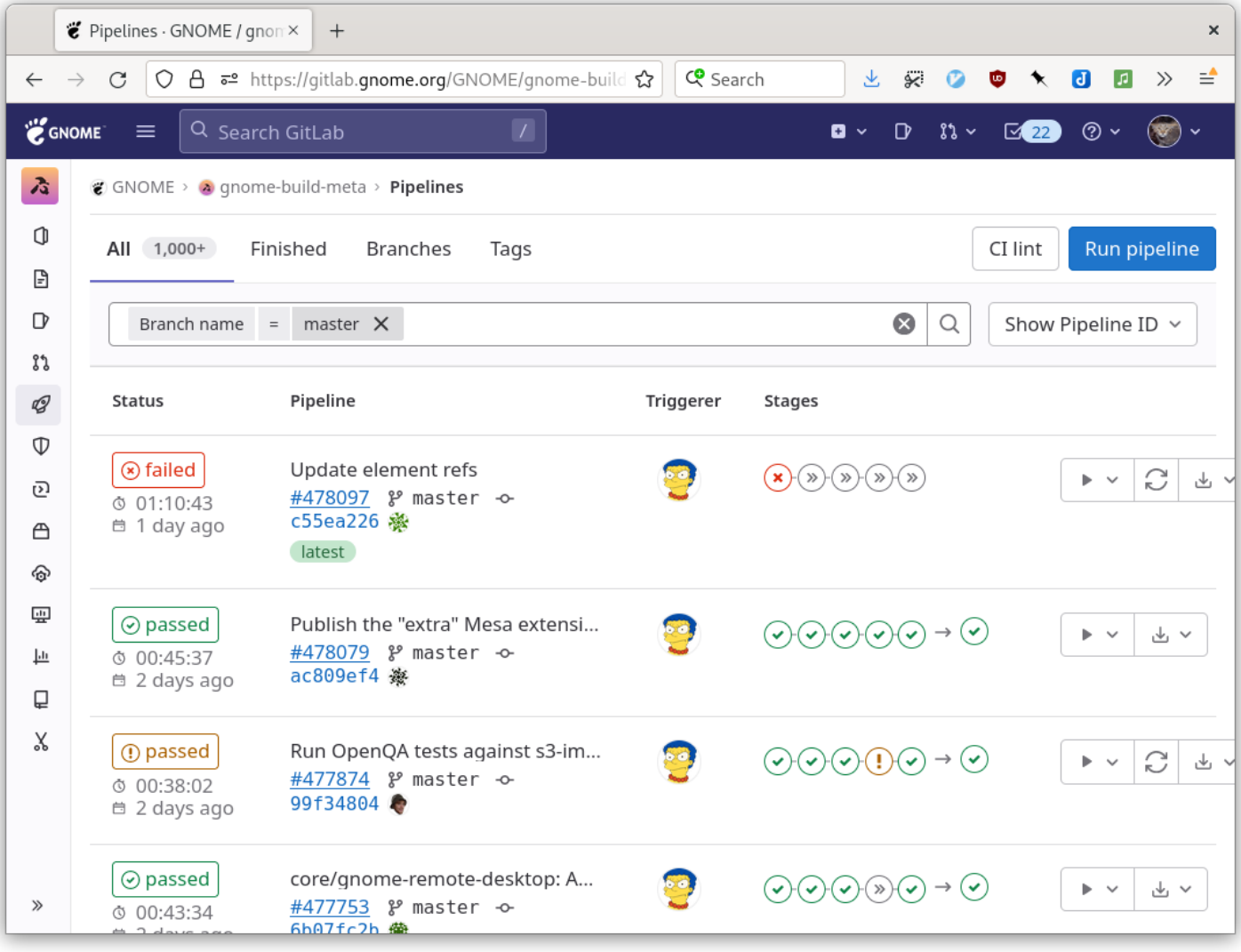
Gitlab: gnome-build-meta repo



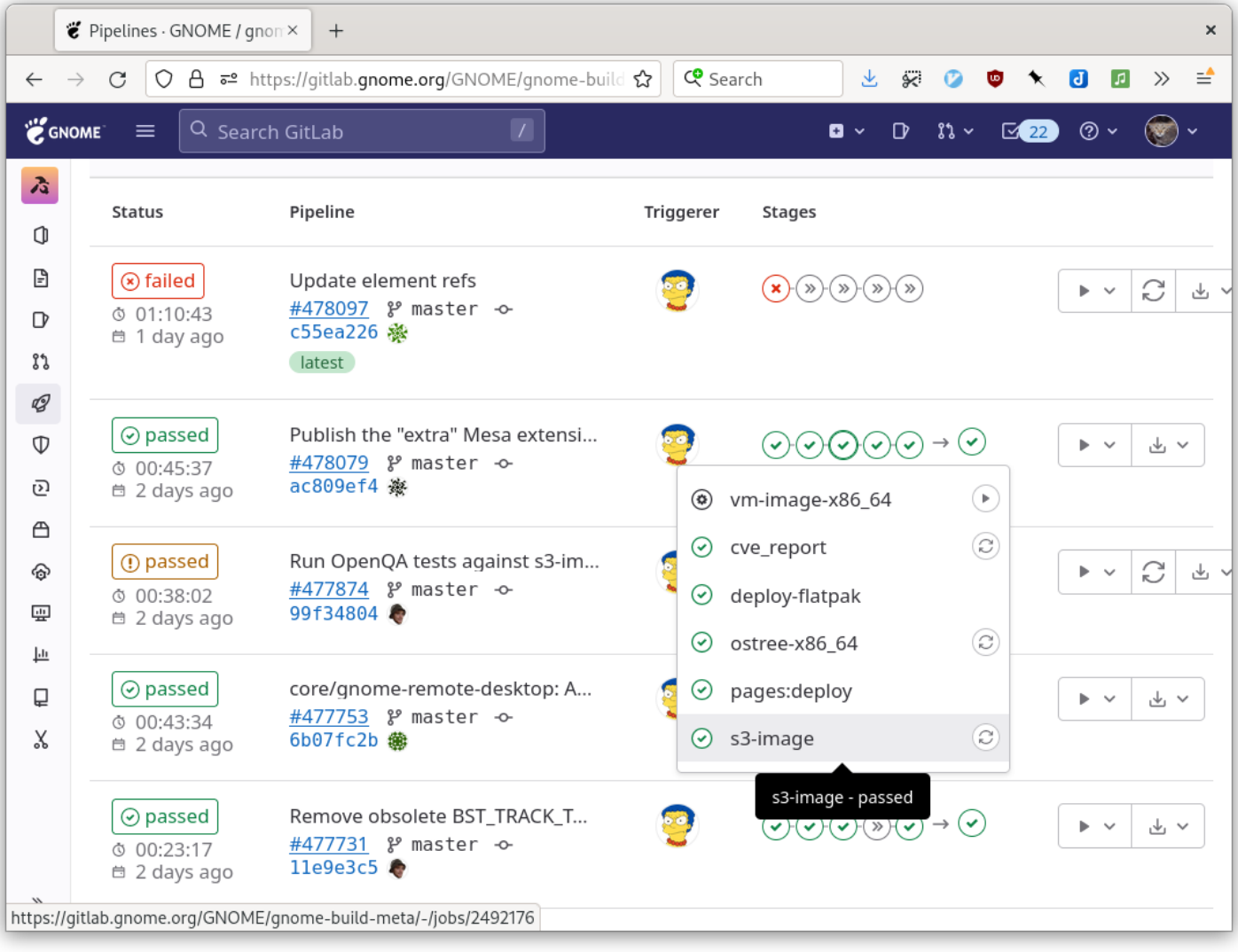
Gitlab: gnome-build-meta wiki



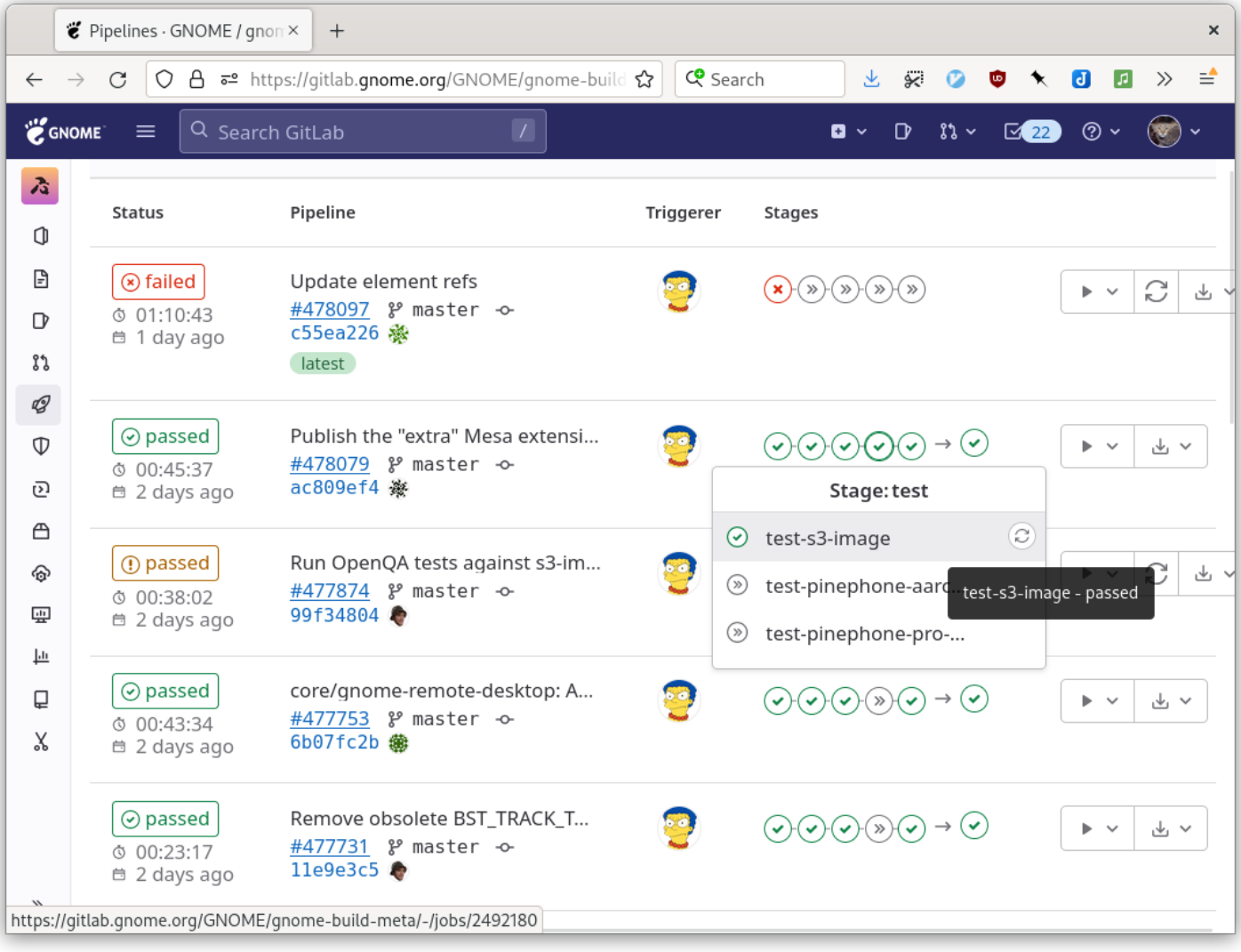
Gitlab: CI pipelines



Gitlab: s3-image



Gitlab: test-s3-image

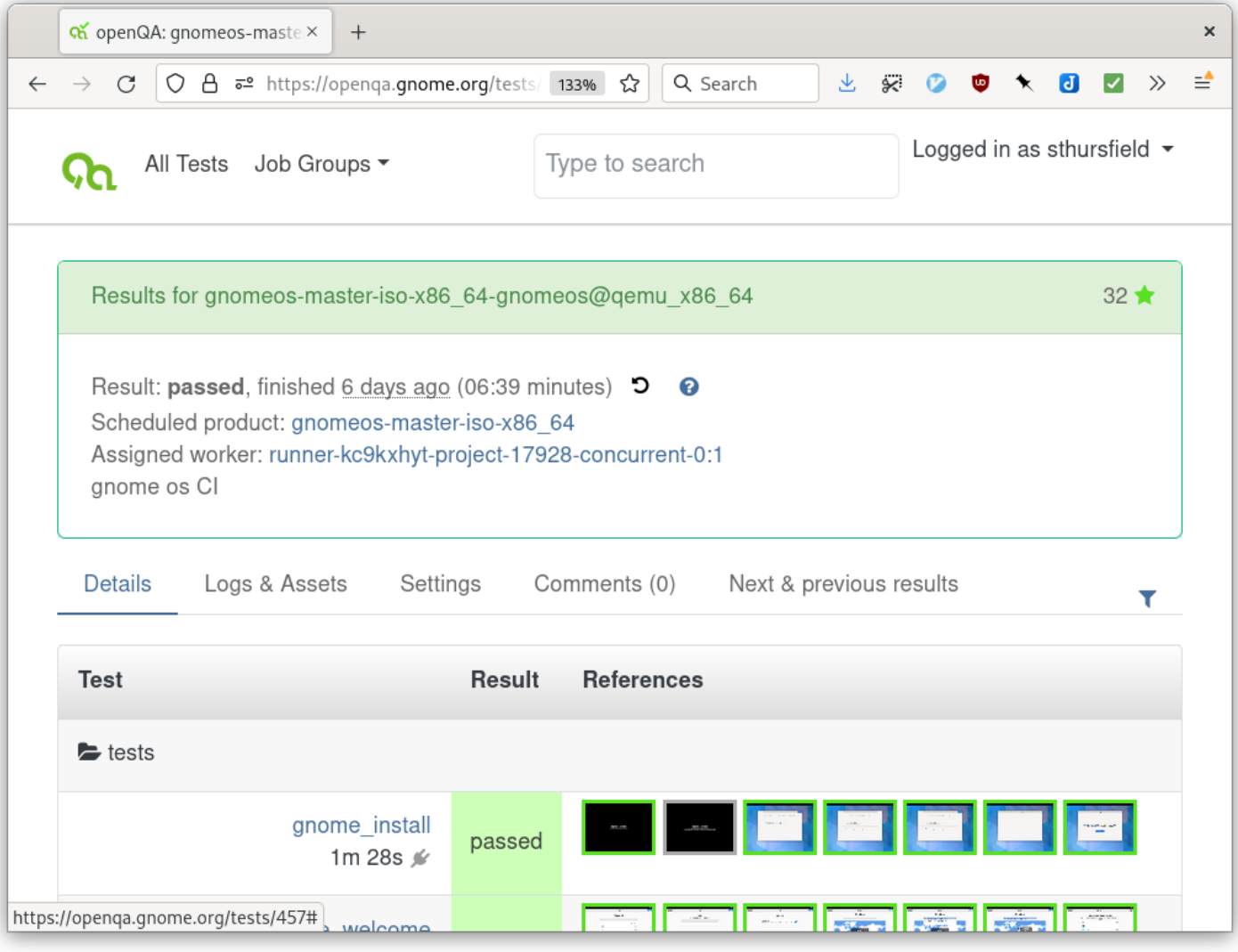


Gitlab: test-s3-image

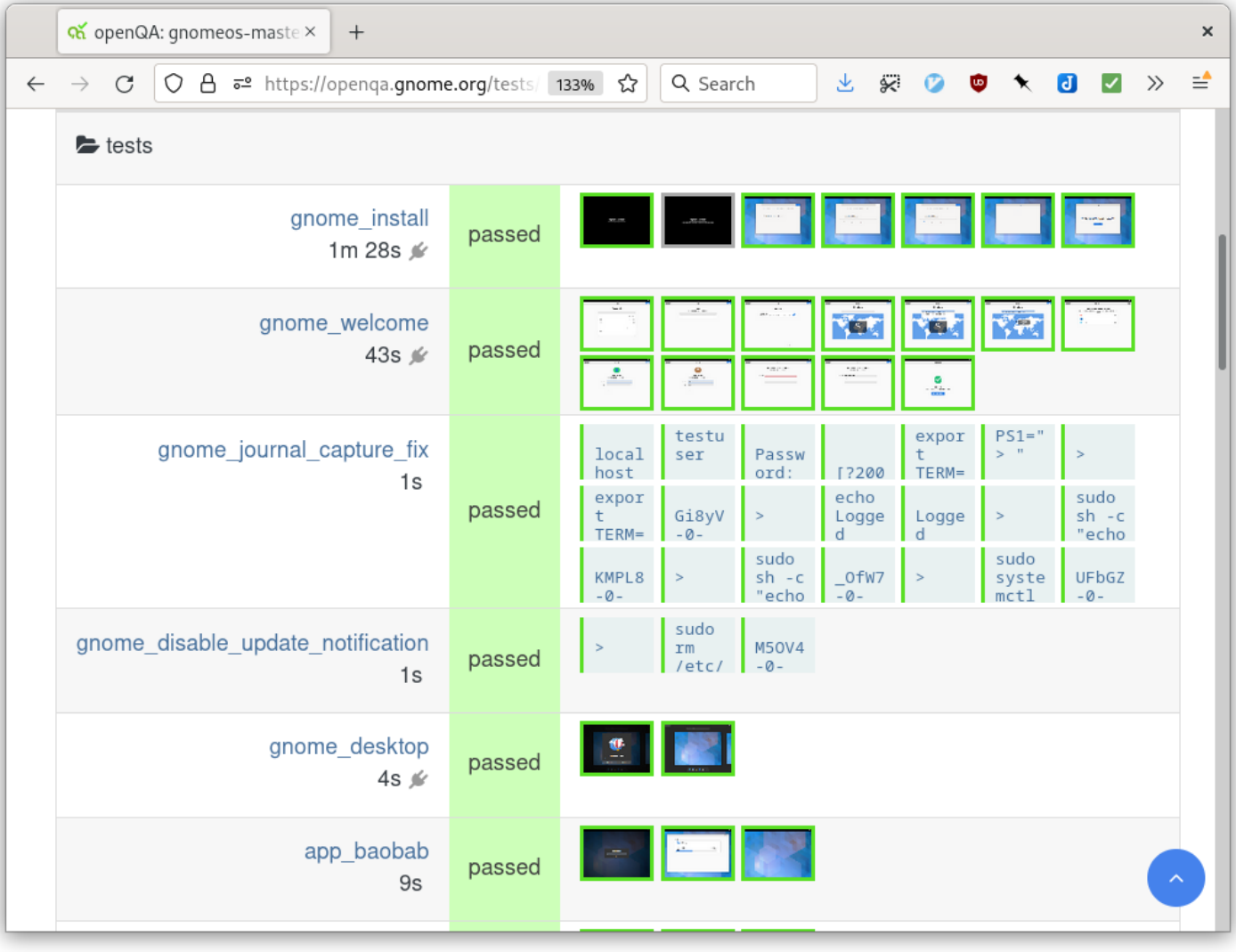
This job:

- runs using the upstream 'openqa-worker' Docker image
- downloads the ISO from S3
- creates a temporary, unique 'machine' and connects it to OpenQA
- submits a test job to OpenQA, tagging it with the unique machine ID
- starts the 'run_worker' script in bg, and waits for the job
- polls job status report until its passed/failed
- job runs, against the locally downloaded ISO, and passes or fails
- removes the machine again and pipeline exists

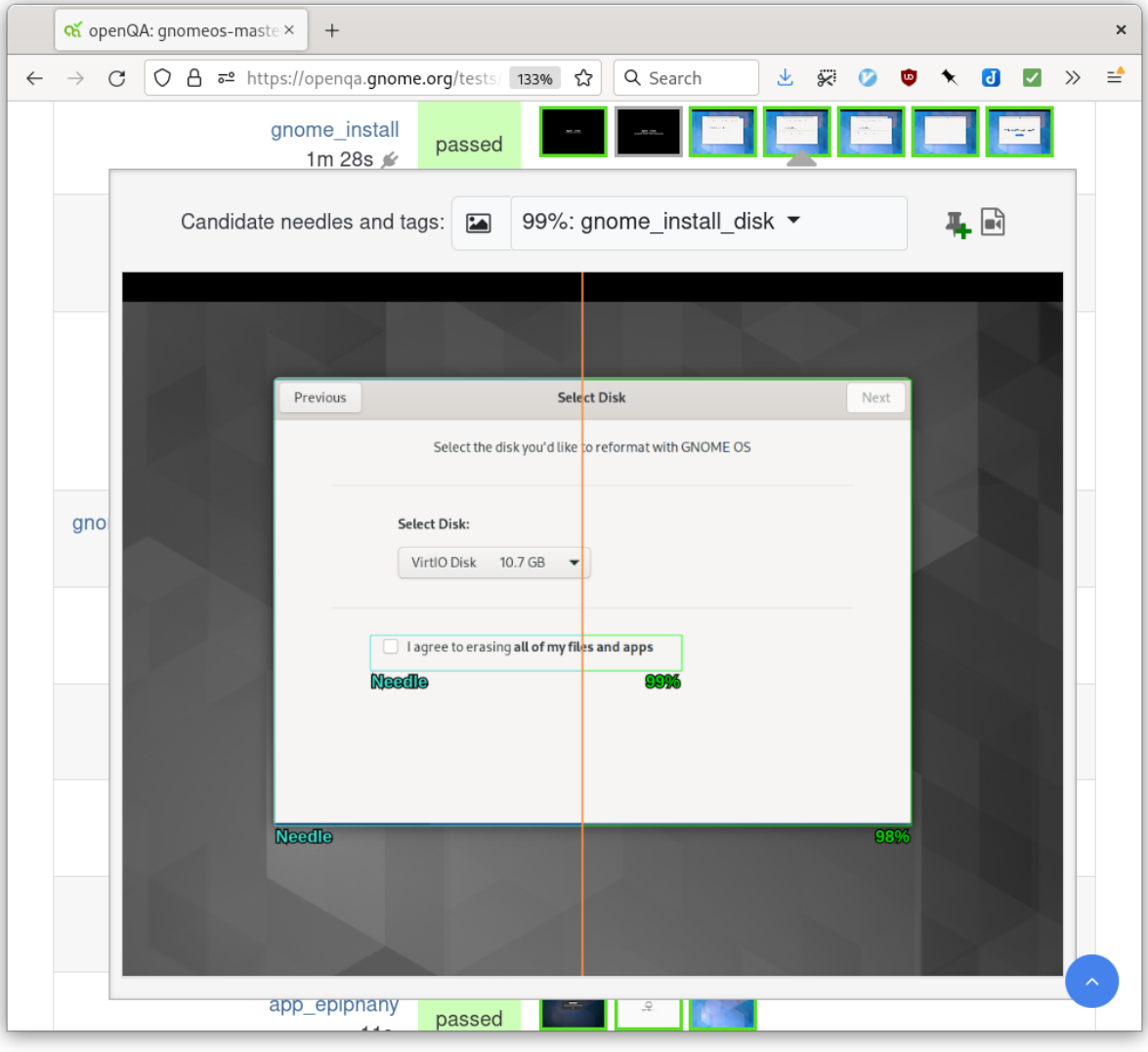
OpenQA: Test results



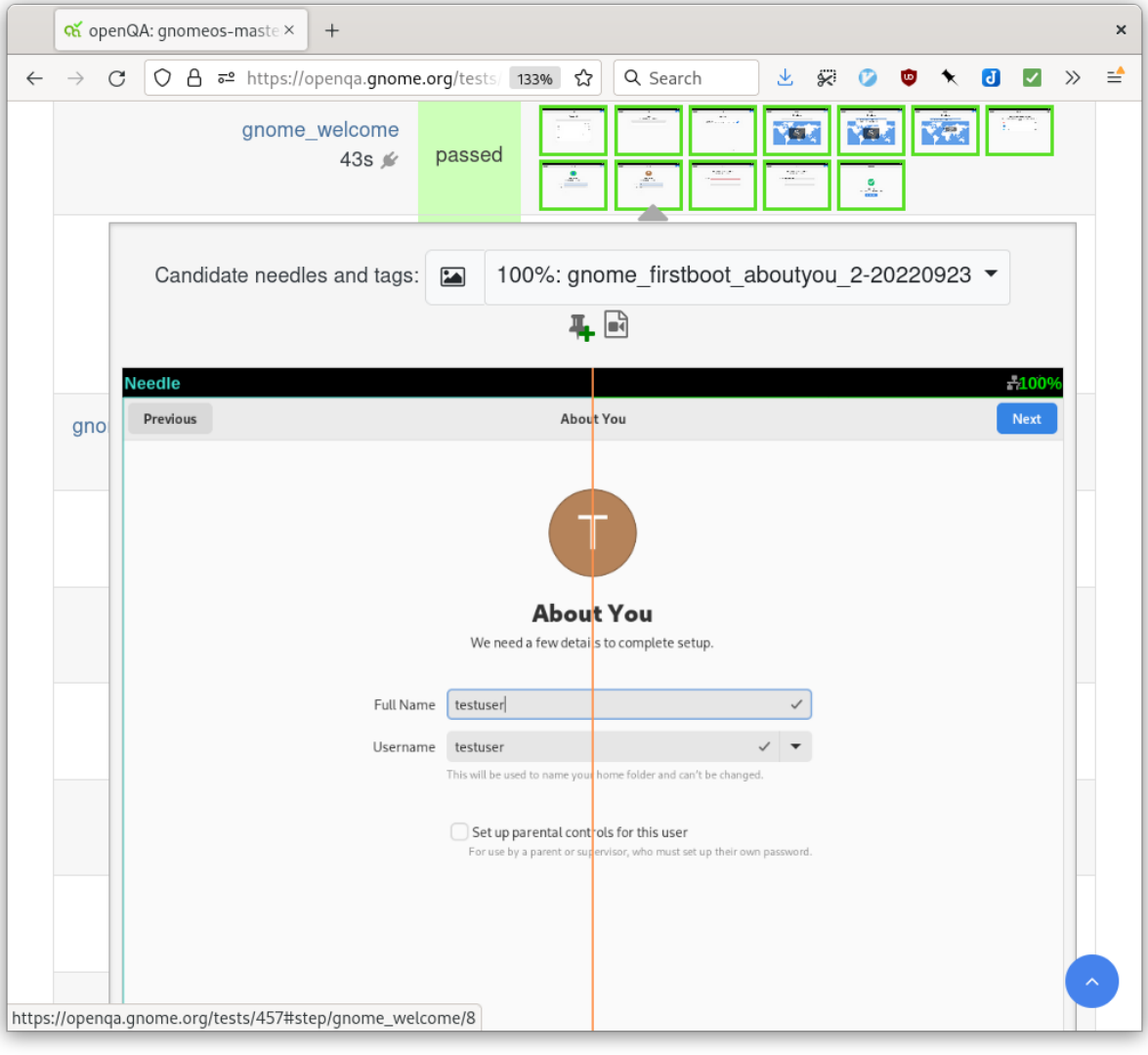
OpenQA: Test results



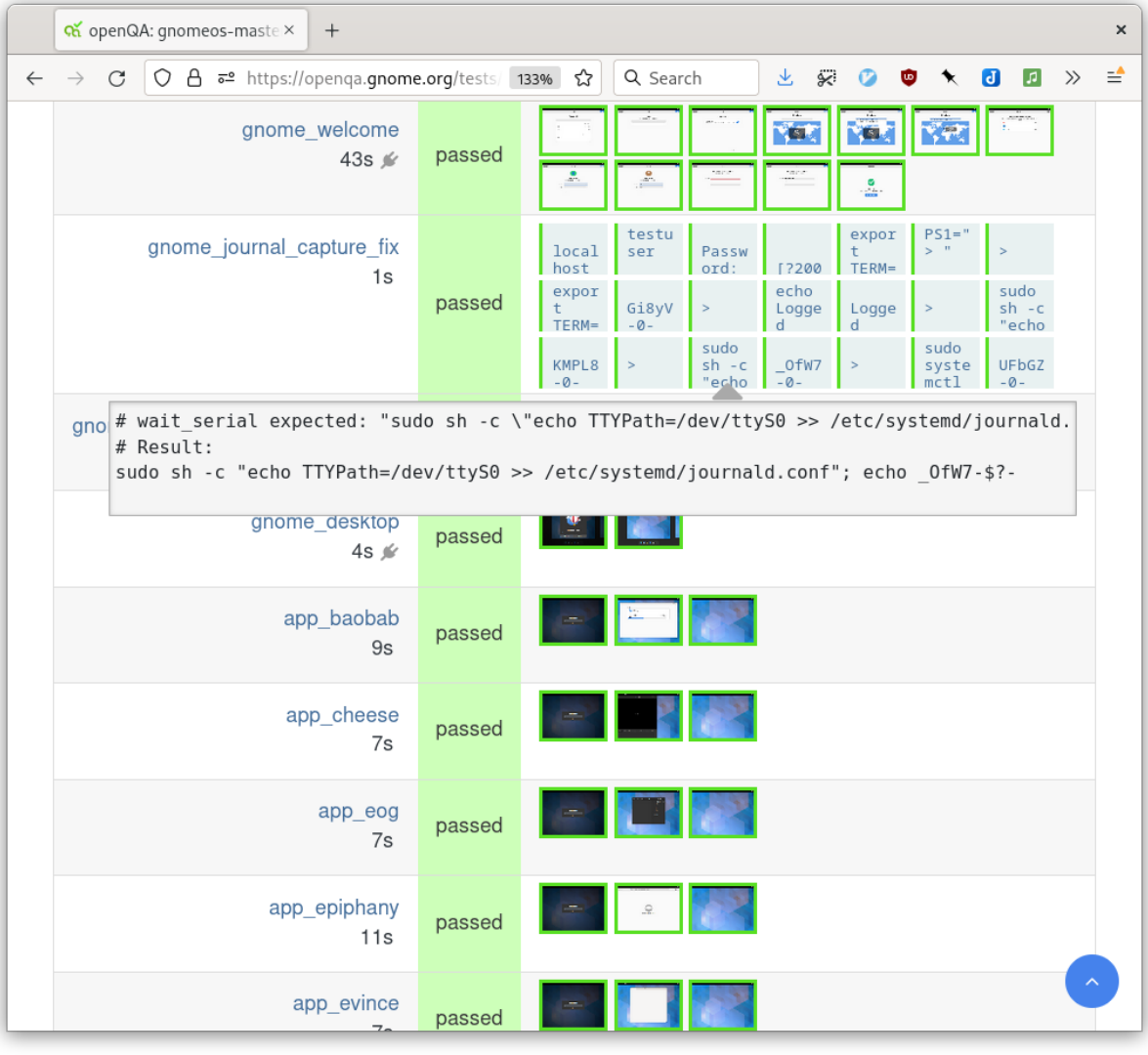
OpenQA: `gnome_install` test



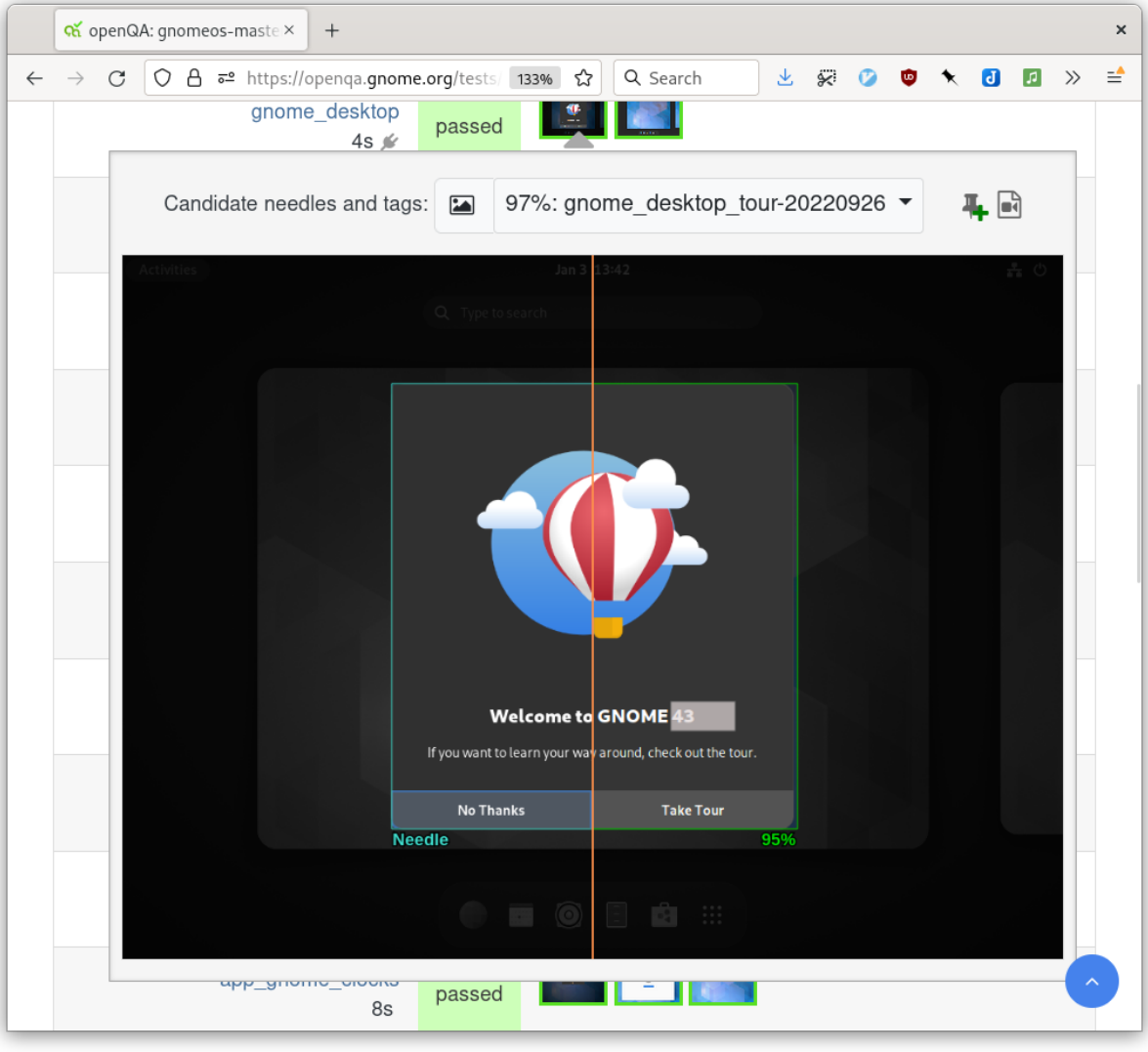
OpenQA: gnome_welcome test



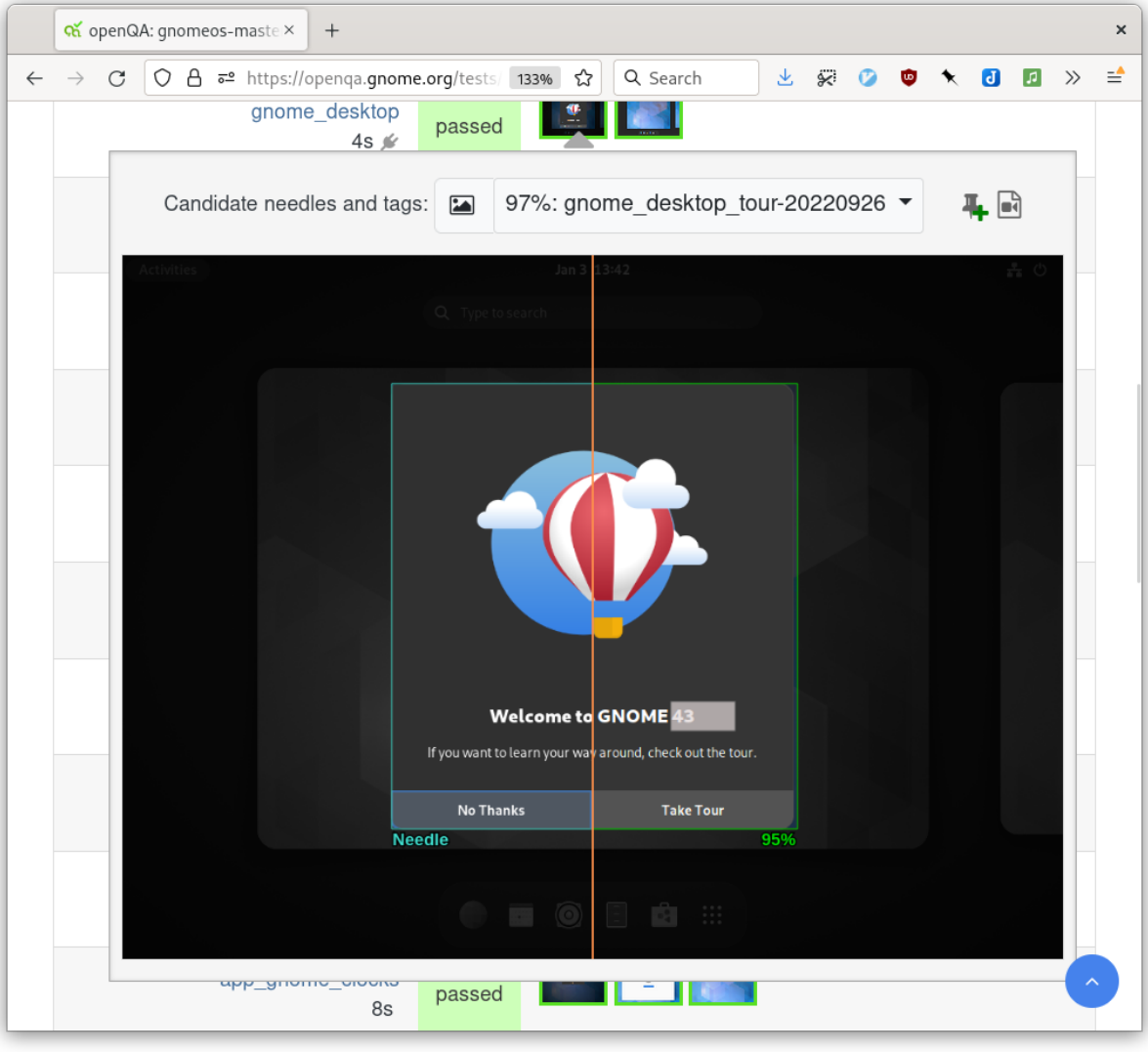
OpenQA: gnome_journal_capture_fix test



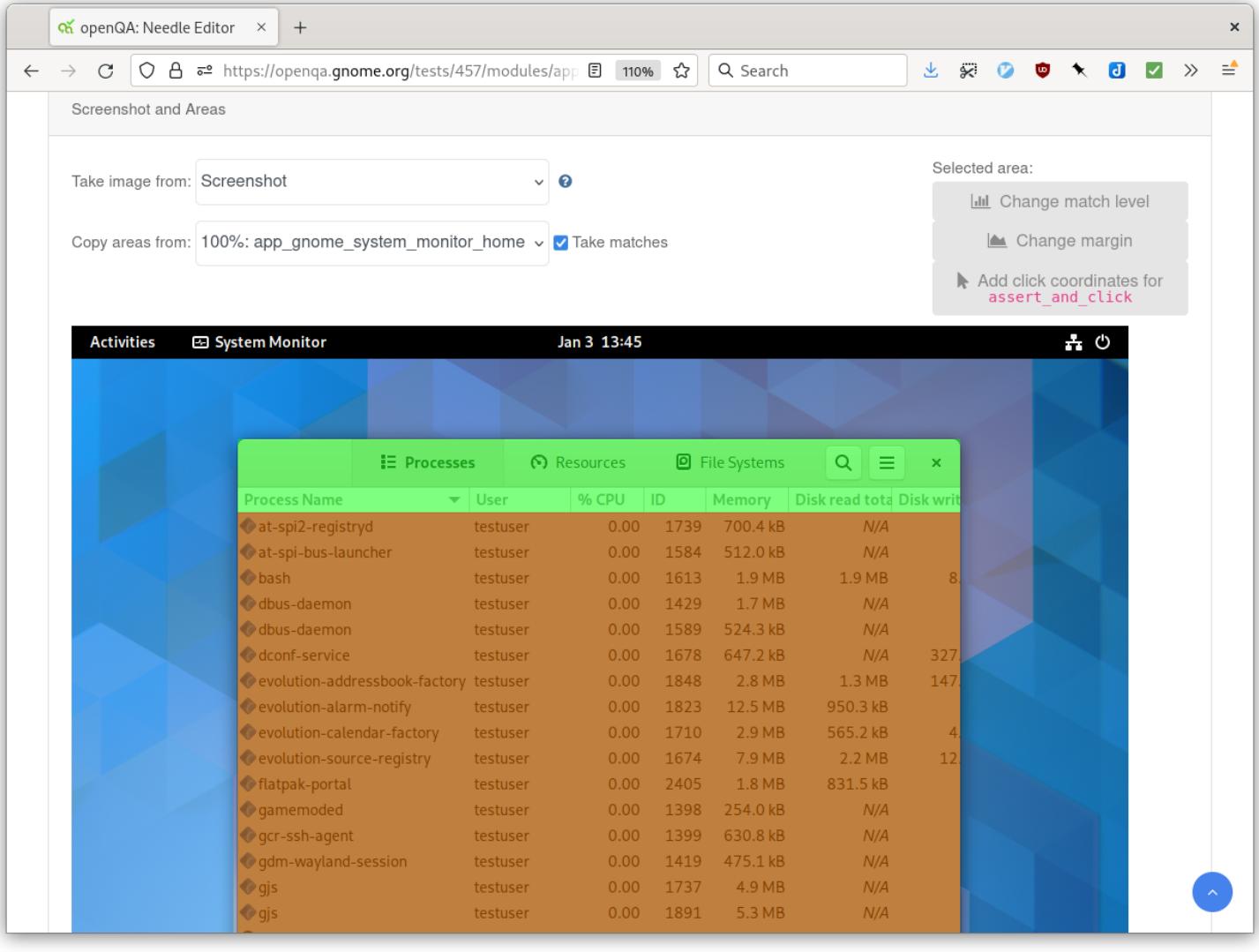
OpenQA: gnome_desktop test



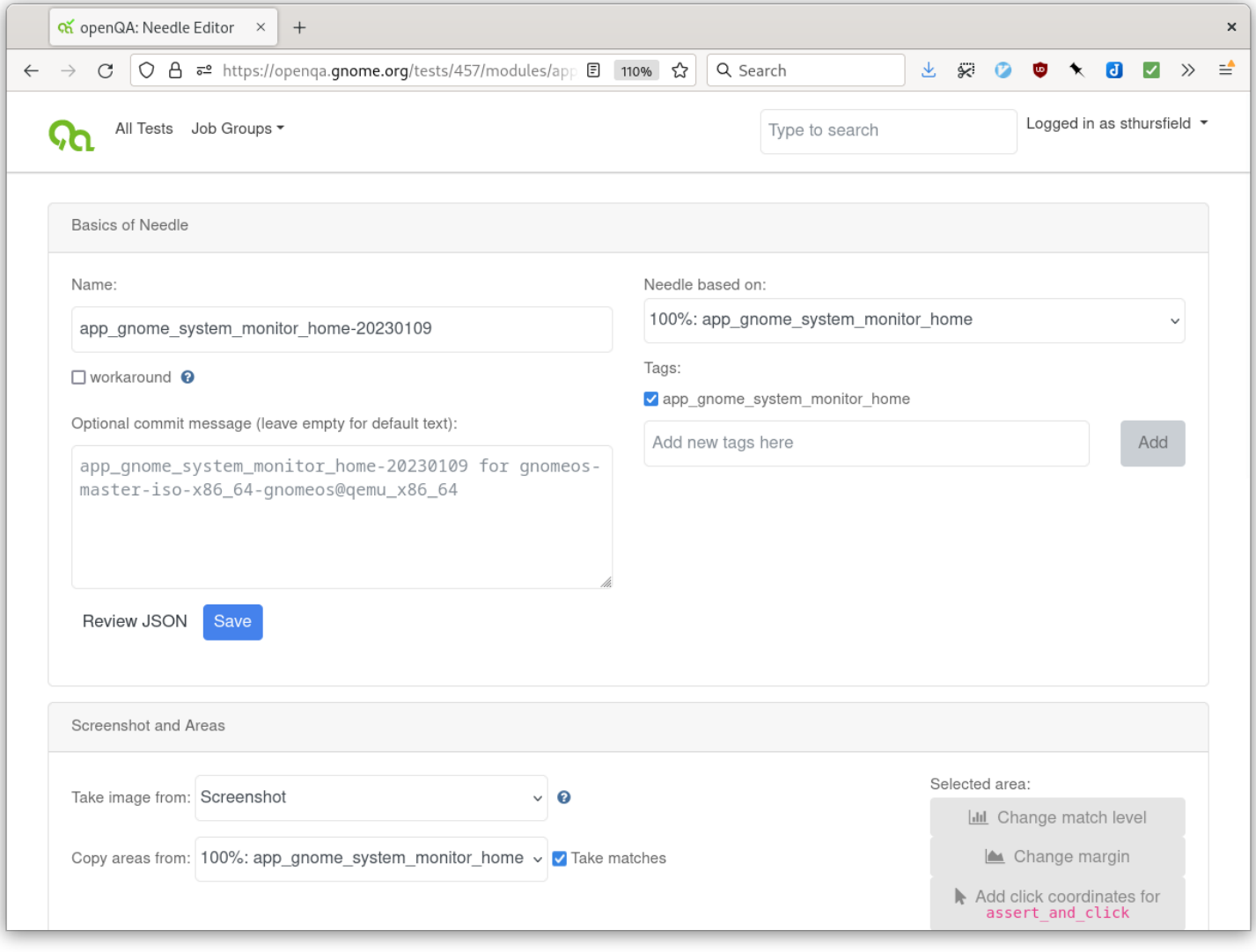
OpenQA: `gnome_system_monitor` test



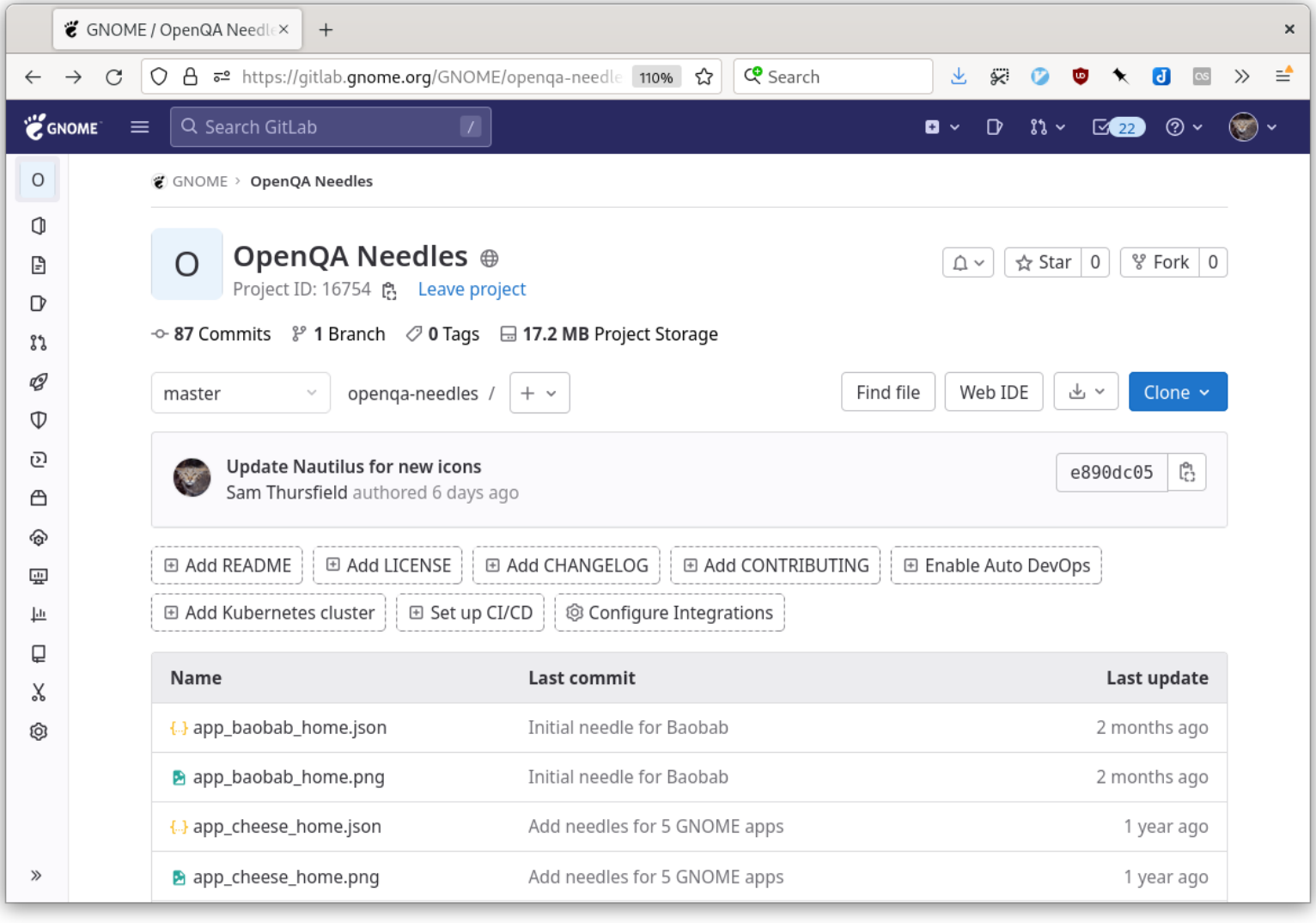
OpenQA: needle editor



OpenQA: needle editor 2

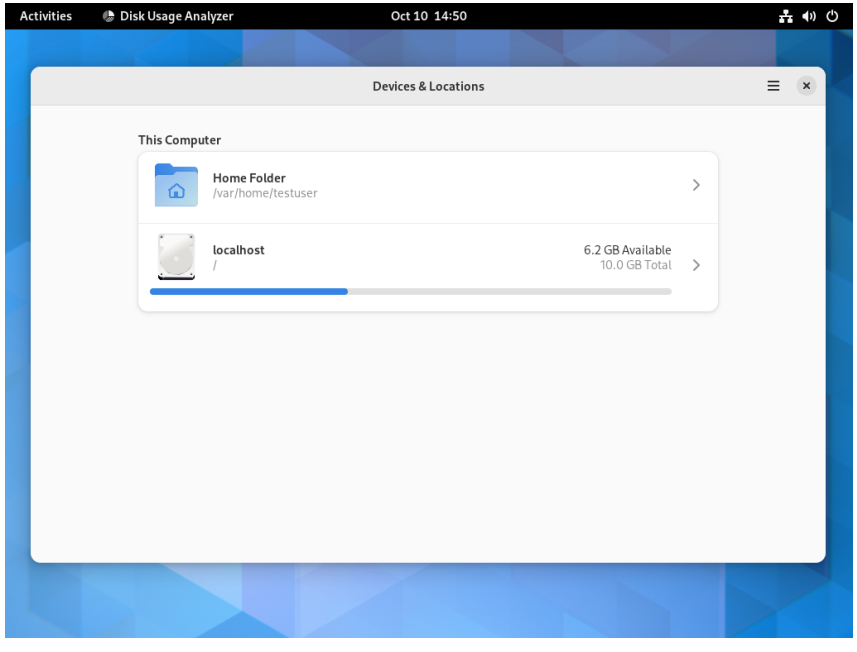


OpenQA: openqa-needles Git repo

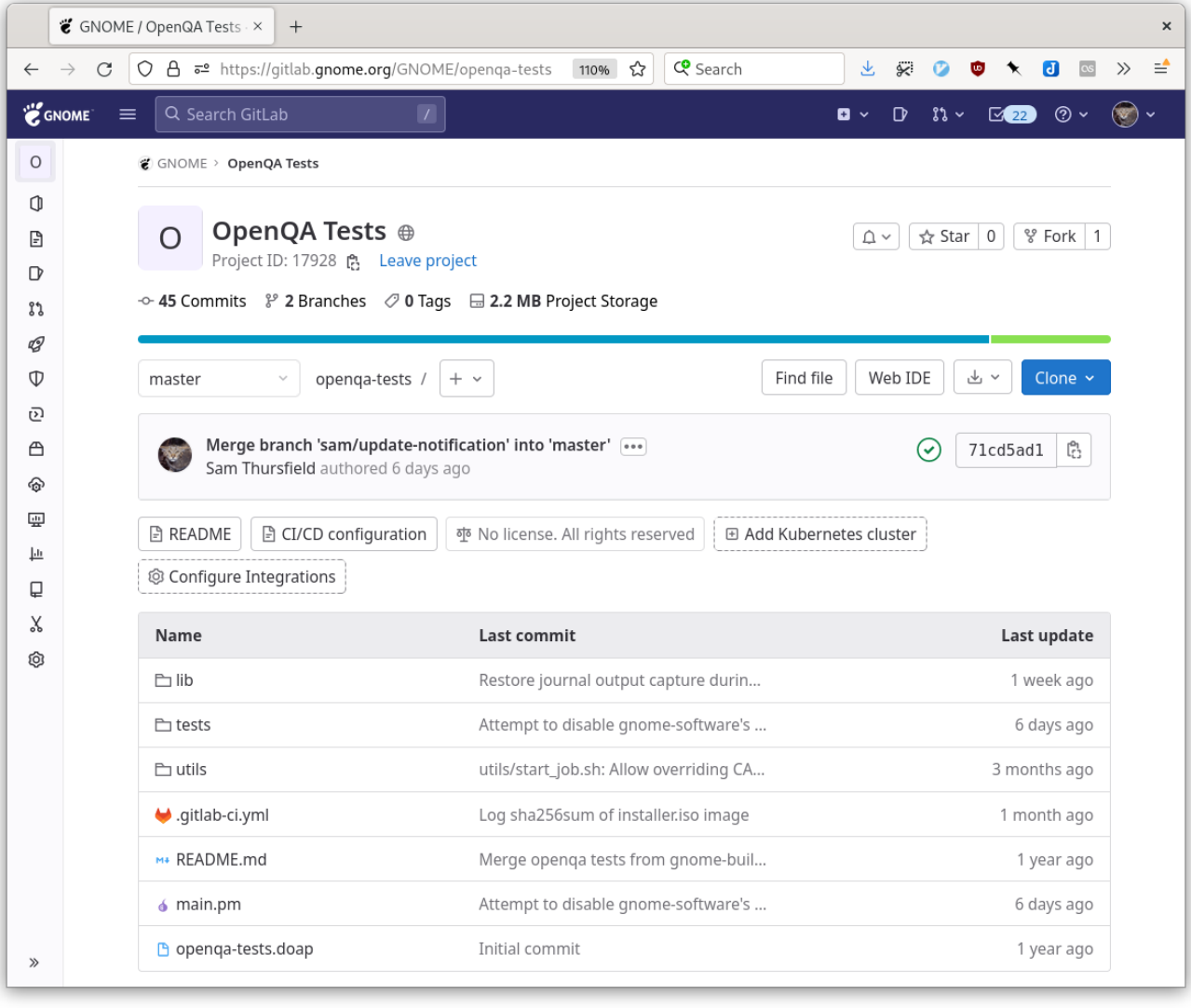


OpenQA: a needle


```
{
  "area": [
    {
      "xpos": 31,
      "ypos": 78,
      "width": 959,
      "height": 599,
      "type": "match"
    }
  ],
  "properties": [],
  "tags": [
    "app_baobab_home"
  ]
}
```

openqa-tests.git



openqa-tests/main.pm

```
my $distri = testapi::get_required_var('CASEDIR') . '/lib/gnomeos
require $distri;
testapi::set_distribution(gnomeosdistribution->new);

$testapi::username = 'testuser';
$testapi::password = 'testingtesting123';

autotest::loadtest("tests/gnome_install.pm");
autotest::loadtest("tests/gnome_welcome.pm");
autotest::loadtest("tests/gnome_journal_capture_fix.pm");
autotest::loadtest("tests/gnome_disable_update_notification.pm");
...
```

openqa-tests/tests/gnome-install.pm

```
use base 'basetest';
use strict;
use testapi;
use bootloader;

sub run {
    my $self = shift;

    bootloader_add_kernel_args(' console=ttyS0 systemd.journald.f

    assert_and_click('gnome_install_1', timeout => 120, button =>
    assert_and_click('gnome_install_disk', timeout => 10, button
    assert_and_click('gnome_install_disk2', timeout => 10, button
    assert_screen('gnome_install_reformatting1', timeout => 120);
    assert_screen('gnome_install_complete', timeout => 180);
    eject_cd;
```

openqa-tests/tests/gnome-welcome.pm

```
sub run {
    my $self = shift;

    assert_and_click('gnome_firstboot_welcome', timeout => 600, b
    assert_and_click('gnome_firstboot_language', timeout => 10, b
    assert_and_click('gnome_firstboot_privacy', timeout => 10, bu
    assert_screen('gnome_firstboot_timezone_1', 30);
    send_key('tab');
    type_string('London, East', wait_screen_change => 6, max_inte
    ...
```

TIPS AND TRICKS

1. OpenQA is great! ❤️ Use it!
2. Explore the test library code:
    - testapi docs
    - os-autoinst-distri-opensuse
3. Keep tests simple.
4. Always check the os-autoinst logs.
    - Example: If needle bounds are invalid, you get a log message and a "0% match" in web UI
5. Learn how to run the testsuite locally.
6. Take care with upstream containers - pin versions using container hash.

NEXT STEPS FOR GNOME OPENQA

1. Build a small team to maintain tests and infra.
2. Reach "production ready" state.
3. GNOME module teams maintaining & extending their own tests.
4. Add example user content (text documents, multimedia, etc)

CREDITS

Top names from gnome-continuous and gnome-build-meta repos:

- Abderrahim Kitouni
- Carlos Garcia Campos
- Colin Walters
- Debarshi Ray
- Dor Askayo
- Emmmanuele Bassi
- Giovanni Campagna
- Iñigo Martinez
- Jasper St. Pierre
- Javier Jardón
- Jeremy Bicha
- Jordan Petridis
- Michael Catanzaro
- Owen Taylor
- Philip Chimento
- Tristan Van Berkom
- Vadim Rutkovsky
- Valentin David

Special mentions:

- Allan Day (blogs, documentation)
- Andrea Veri (OpenID help)
- James Thomas (openQA tests & QEMU help)
- Will Thompson (Endless installer, OpenQA advice, ...)

CODETHINK PLANS FOR 2023

- Continue Linux mainline testing (OpenQA + LAVA)
- **QAD**: open source tool to control hardware from OpenQA tests
- Hardware **USB switcher** device

Follow us for details:

- @codethink@social.codethink.co.uk (Fediverse)
- @codethink (Twitter)
- https://www.codethink.co.uk/

HOW TO GET INVOLVED

I will provide training on infra maintenance and writing tests - just ask!

- Chat: Matrix #gnome-os:gnome.org (Libera.chat #gnome-os)
- Email: *sam@afuera.me.uk*
- Forum: https://discourse.gnome.org/

Also: documentation, issue tracker

Sam Thursfield
FOSDEM 2023