Computational
Health
Informatics

Leibniz
Universität
Hannover

Retrospective & outlook for technical developments

Gerald Wiese

GNU Health Con 2024 / Palermo

Table of contents

1. Migration to Codeberg

2. New Installation

3. News from Ansible automation

4. HIS 5.0
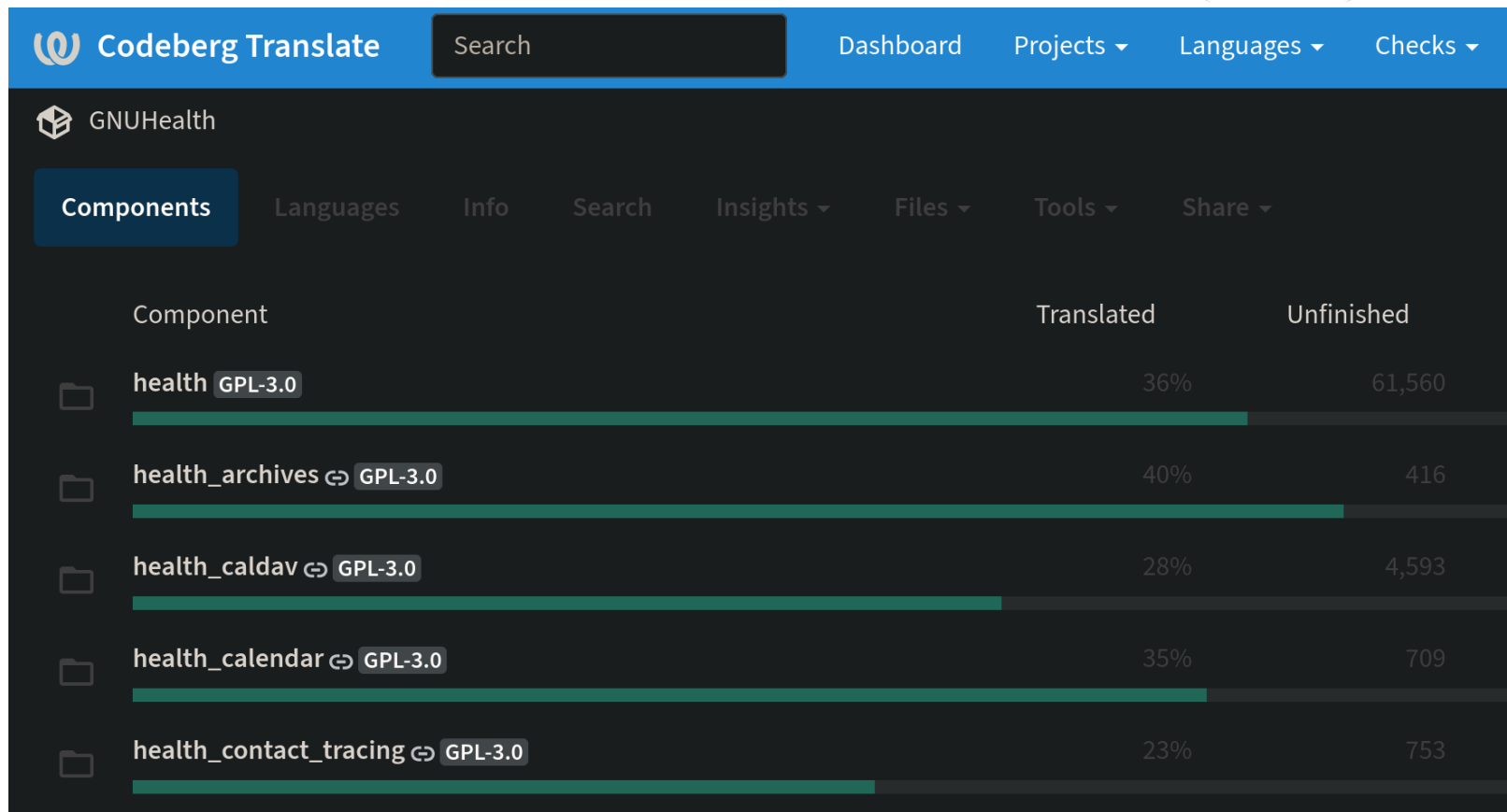
1 – Migration to Codeberg

Migration to Codeberg

- Git repositories

- Forgejo as base

- Community-driven non-profit

- For Free Software

- Association in Berlin/Germany

GNU Health
@gnuhealth@mastodon.social

Very happy to announce that we are migrating the #GNUHealth development resources to @Codeberg ❤️😎

🚧 During the process, some resources (ike the translation service @weblate) will experience some downtime, so please stay tuned 🤗

Mar 18, 2024, 11:52 AM · 🌐 · Web

26 boosts · **58** favorites

Codeberg.org
@Codeberg@social.anoxinon.de
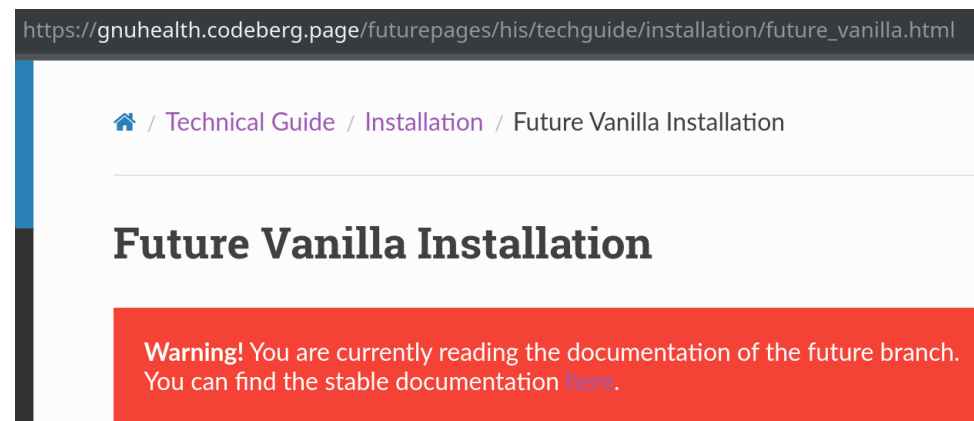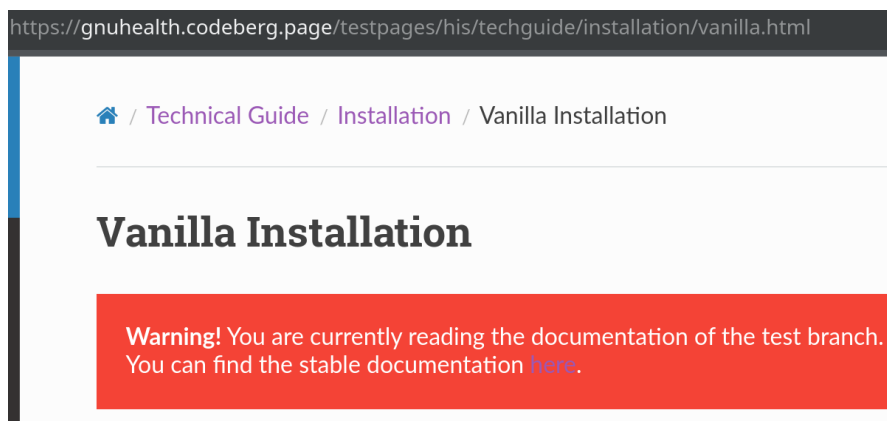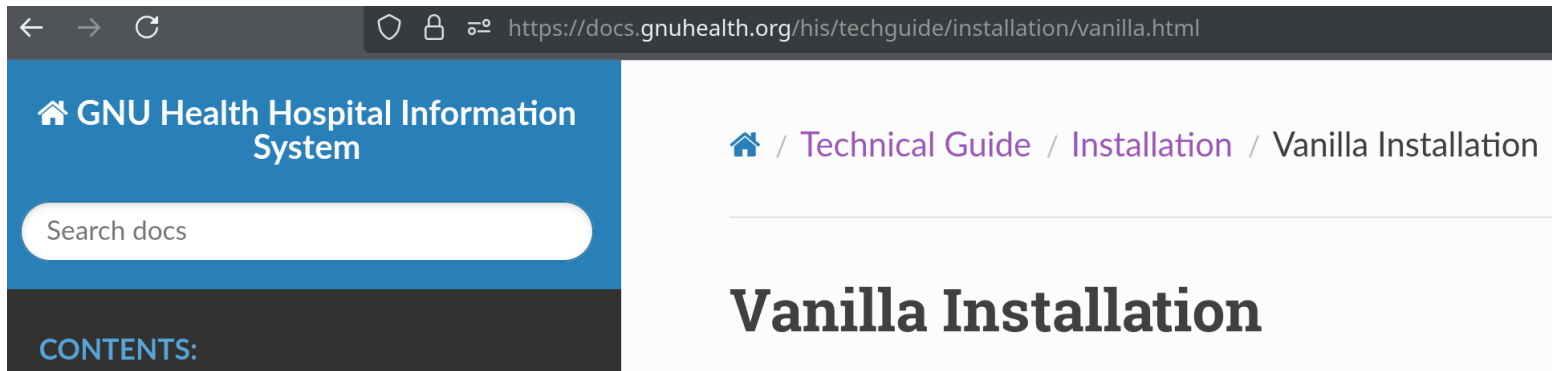🌐 Mar 18

@gnuhealth Welcome!

Codeberg Translate: Weblate instance

- Same software – new hosting provider

Codeberg Pages – GNU Health Documentation

- 3 versions of docs: Main, test & future
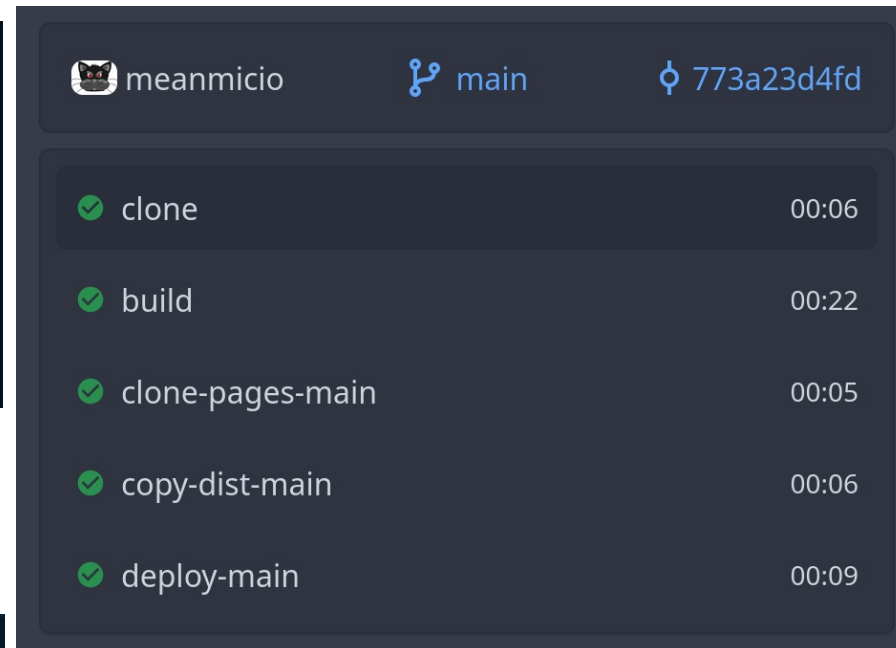
Woodpecker CI

- Until here: Linting, testing and deploying the documentation

```
steps:
  test:
    image: gnuhealth/linter:1.0.1
    commands:
        - bash tests/lint/lint_ansible.sh
        - bash tests/config_stock/check_config_stock.sh
        - bash tests/var_check/var_check.sh
        - reuse lint
```

Code Snippet from Ansible repository

meanmicio	⌥ main	◊ 773a23d4fd
✓ clone		00:06
✓ build		00:22
✓ clone-pages-main		00:05
✓ copy-dist-main		00:06
✓ deploy-main		00:09
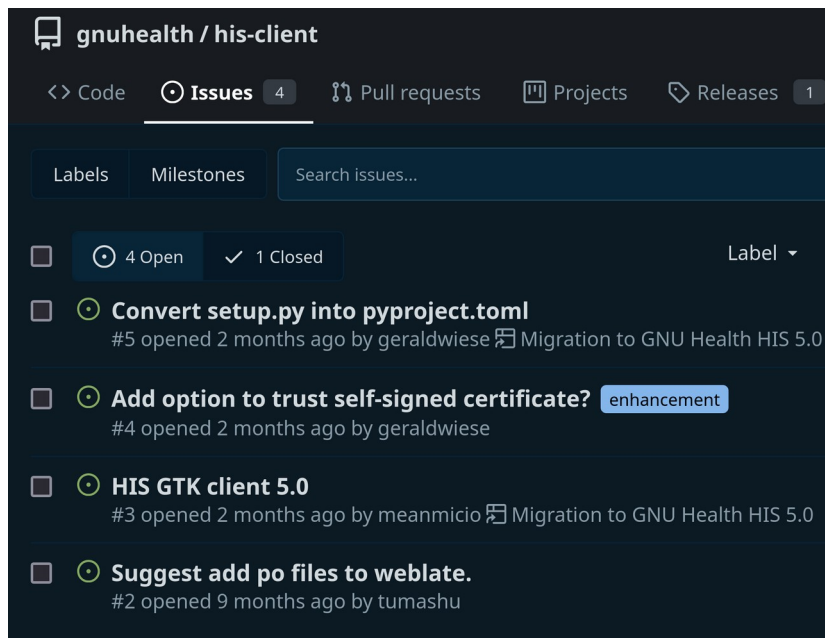
Sample output from Documentation repository

gnu `56e097bb29` ✓ health orthanc

Check mark from HIS repository

Issues & Projects

- Issues replace GNU Savannah bugs & tasks
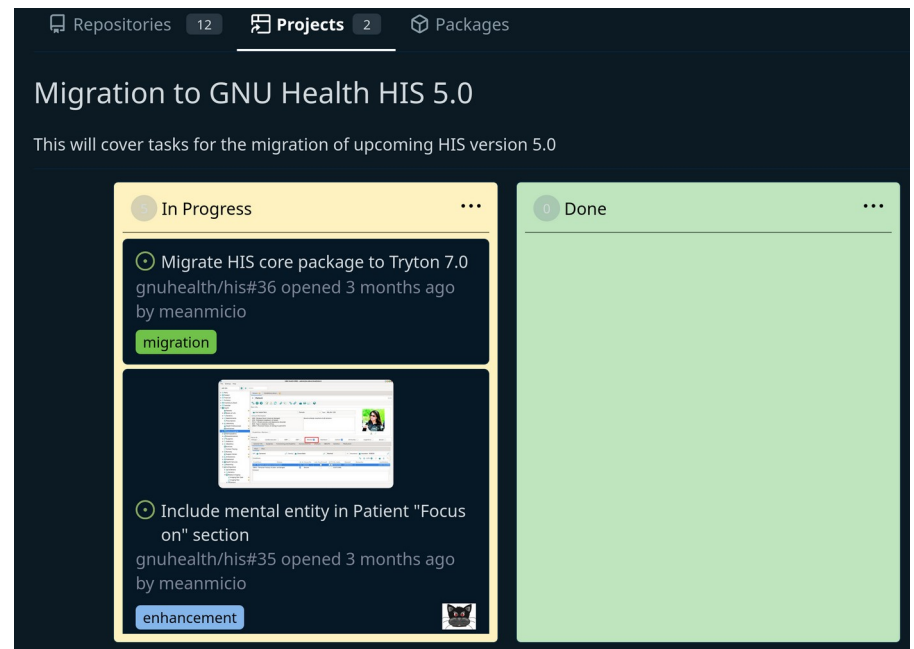
- Projects allow grouping and visualizing

2 – New Installation

Installation – where?

- Starting with 5.0: One global PyPI package **gnuhealth-his** containing all modules and **gnuhealth-utils** for config snippets and scripts

- Use Pythons builtin virtual environment instead of user space
- Use pyproject.toml

- Testing? Pip install directly from git to avoid TestPyPI
  Avoid waiting for uploads and security risks

How to handle different OS?

- Reduced number tested properly

Operating System	Version
Debian	12 (Bookworm)
openSUSE	Leap 15.6
FreeBSD	14.1
Ubuntu	24.04 LTS

- Fool-proof copy/paste

- Unified directory structure

```
root@debian12:/opt/gnuhealth# tree -L 2
.
├── his
│   ├── data
│   ├── etc
│   ├── local
│   ├── log
│   ├── run
│   ├── tls
│   └── venv
├── nginx
│   ├── log
│   ├── sites-available
│   └── sites-enabled
└── postgresql
    ├── 15
    ├── conf.d
    ├── gh_hba.conf
    └── log
```

Commands for: FreeBSD, openSUSE

```
$ sudo mkdir -p -m 0700 /opt/gnuhealth/postgresql/{conf.d,log}
$ sudo chown -R postgres:postgres /opt/gnuhealth/postgresql
$ MY_PG_MAJOR_VERSION=$(psql -V | awk '{print $NF}' | cut -d '.' -f 1)
$ sudo su postgres -c "initdb -D /opt/gnuhealth/postgresql/${MY_PG_MAJOR_VERSION}/main"
```

Further instructions for all OS

- Always run GNU Health / Tryton using a production grade application server uWSGI

- Configuring Nginx should always imply encrypted communication / HTTPS

flaticon.com

- Service for automatic start after reboot (FreeBSD as well)

```
● gnuhealth.service - GNU Health HIS Server
     Loaded: loaded (/etc/systemd/system/gnuhealth.service; enabled; preset: enabled)
    Drop-In: /run/systemd/system/service.d
             └─zzz-lxc-service.conf
     Active: active (running) since Tue 2024-12-03 10:33:39 UTC; 5h 29min ago
   Main PID: 149 (uwsgi)
     Status: "uWSGI is ready"
      Tasks: 2 (limit: 38000)
     Memory: 183.8M
        CPU: 4.426s
     CGroup: /system.slice/gnuhealth.service
             ├─149 /opt/gnuhealth/his/venv/bin/uwsgi --ini /opt/gnuhealth/his/etc/uwsgi.ini
```

Parse Shell script from docs

- Create BASH script from docs (remote or local path)

```
root@debian12:~# python3 ./make_script_from_docs.py --help
usage: make_script_from_docs.py [-h] [-o {Debian,Ubuntu,openSUSE,FreeBSD}] [-w WRITE_TO] [-r READ_FROM] [-t] [-c] [--client_only]

This script reads the GNU Health HIS installation docs from a remote URL starting with https:// or a local path and creates a BASH script from it.
The operating system has to be set and defaults to Debian.

options:
  -h, --help            show this help message and exit
  -o {Debian,Ubuntu,openSUSE,FreeBSD}, --operating_system {Debian,Ubuntu,openSUSE,FreeBSD}
                        Operating system used for installation
  -w WRITE_TO, --write_to WRITE_TO
                        Write BASH script into this path
  -r READ_FROM, --read_from READ_FROM
                        Read documentation from this path (remote URL starting with https:// or local path)
  -t, --test            Runs in testing mode. Warning: Sets the Tryton password to gnusolidario!
  -c, --client          Install the client as well
  --client_only         Install only the client
```

3 – News from Ansible automation

Synchronization & Testing

Molecule tests:

- **create** VMs
- **converge**: Run installation playbook
- **idempotence**: Run converge again, expect no changes
- **verify**: Test connection
- **destroy** VM

Synchronization test:

- Replace regular **converge** by **side-effect** which runs previous BASH script
- Make **idempotence** run **converge** and expect no changes

→ Detect inconsistencies

Test server

- Daily & weekly tests

- Tests based on last slide + examples from documentation

- Email reports
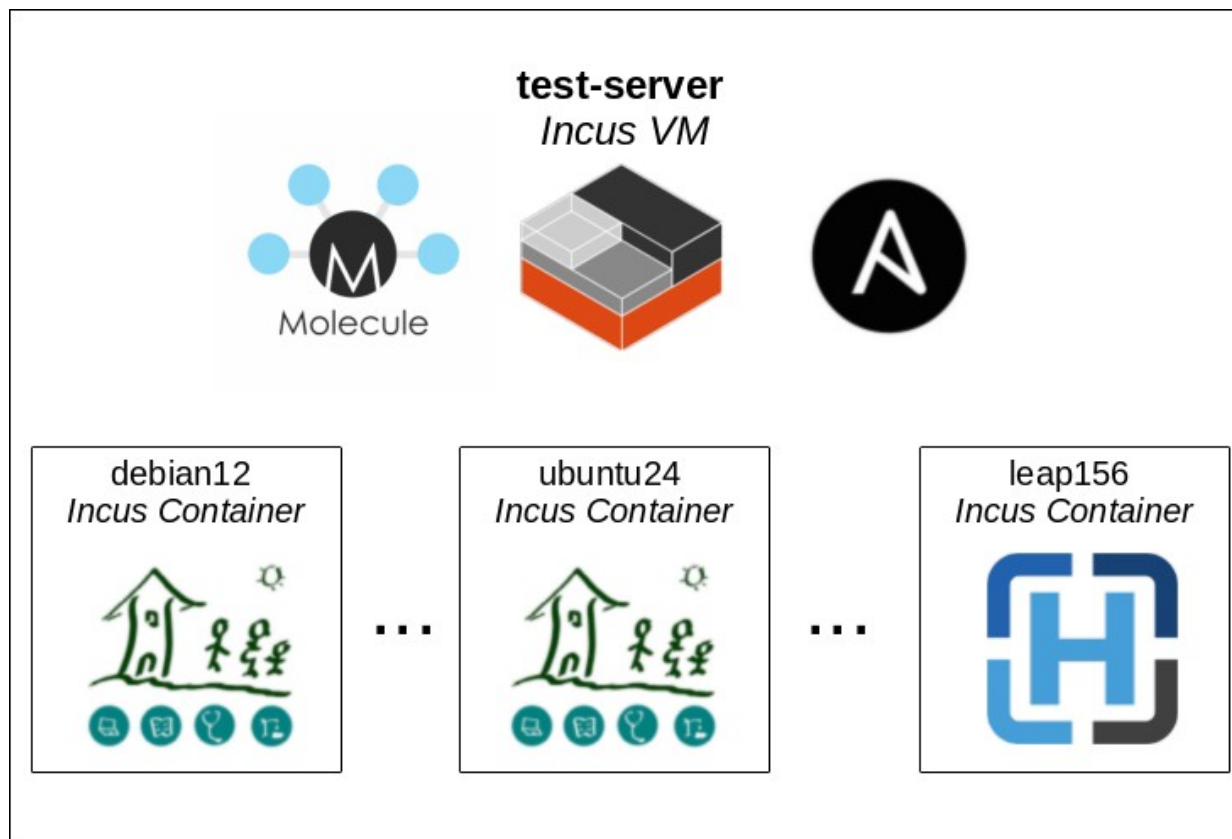
Moving to Incus / Licensing Controversies

- Incus: Manage containers or Virtual Machines (VMs)

- LXD: Canonical moved it from Linux Containers to in-house, changed license to AGPLv3 and put a CLA [Graber1, Graber2]

    → Linux Containers forked LXD as Incus

- Vagrant: Non-interactively create and configure e.g. VMs in VirtualBox or Libvirt/QEMU/KVM
- Hashicorp: Changed Vagrant to Business Source License v1.1 [Hashicorp]

    → Now we avoid Vagrant and only use Incus which is smoothly scriptable itself (and still QEMU/KVM for VMs)
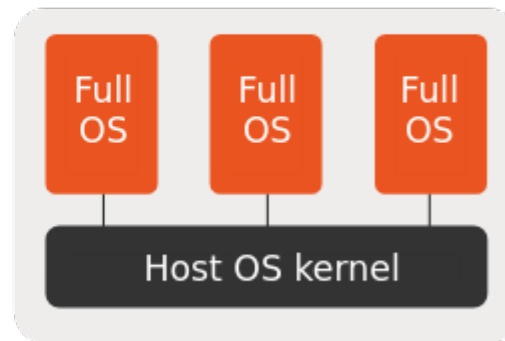
Containers vs. VMs
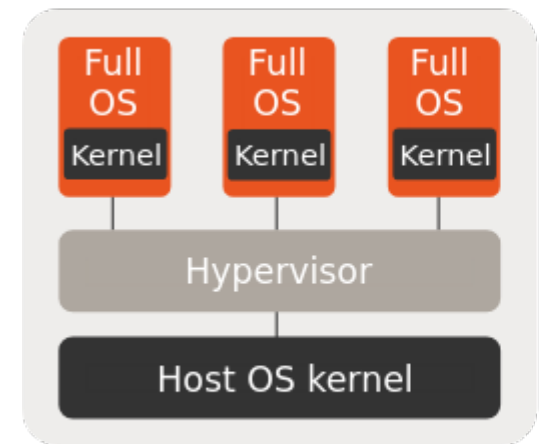
Docker   Incus container   Incus VM
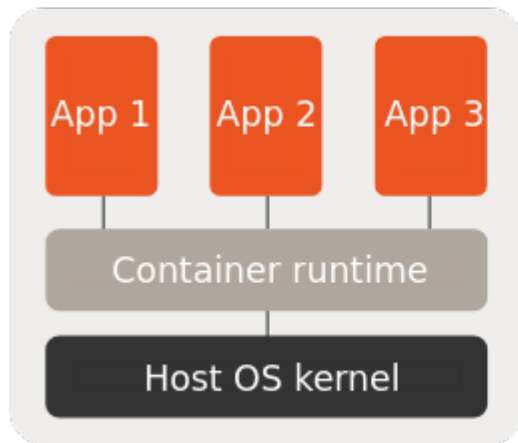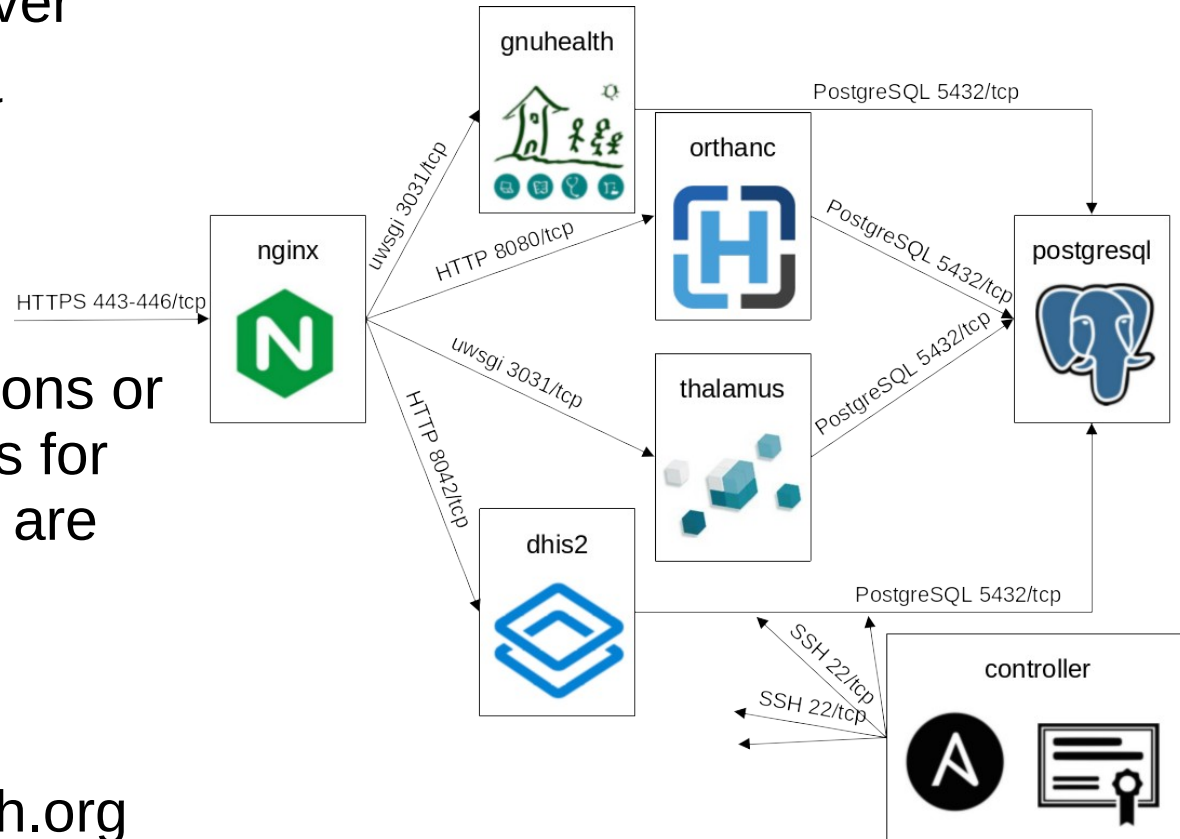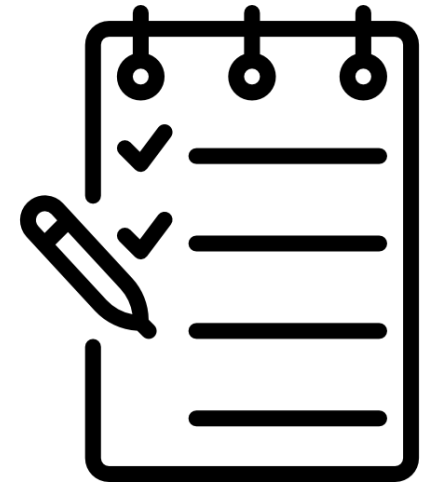

[LXC]

Community Server 2.0

- Provide demo server also for Orthanc & DHIS2

- Provide beta versions or release candidates for HIS 5.0 once they are ready

- Coming soon at sandbox.gnuhealth.org

To dos

- Put monitoring and last productive example

- Add FreeBSD to automated tests

- Finish Vanilla synchronization

flaticon.com

4 – HIS 5.0

What's new in HIS 5.0?

- Migration to Tryton 7.0

- Refactored Orthanc & Radiology modules

- Updated PyPI packaging
- New Vanilla Installation

- DHIS2 module?
  See health-dev list and HIS branch wip-dhis2

- Expected in Q1 2025

Coding conventions

- Code documentation & feature documentation

- Linting
  - Python: pycodestyle & pyflakes
  - REUSE licensing
- Security
  - pip-audit
  - bandit
- HIS server: Tryton module tests
  → See .woodpecker.yaml and scripts/ in HIS repo

- More specific unit and integration tests?

To dos – 5.0 or 5.2

- Tryton Migration of other modules

- Investigate bandit complaints about GTK client

- Finish packaging & installation

- Fulfill coding conventions everywhere

- Counter-check and continue Orthanc & DHIS2 modules

- Continue neglected sub projects: WebDAV & FHIR

About my involvement

- Leaving Leibniz University 15th April 2025

- Planning to stay in community to:
    - Maintain Ansible & Vanilla installation (stable releases)
    - Troubleshooting Woodpecker CI, docs framework, PyPI & helper scripts – at least on demand
    - Run test server and community server?
    - Provide VirtualBox image and Docker?
  But with limited resources

- Realistically speaking: No actual coding

References

- Graber1: https://stgraber.org/2023/07/10/time-to-move-on/
- Graber2: https://stgraber.org/2023/12/12/lxd-now-re-licensed-and-under-a-cla/
- Hashicorp: https://discuss.hashicorp.com/t/hashicorp-projects-changing-license-to-business-source-license-v1-1/57106
- LXC: https://linuxcontainers.org/incus/docs/main/explanation/containers_and_vms/