Automating the Installation & Administration of GNU Health using Ansible

Gerald Wiese

GNU Health Con 2023
Agenda

- 0. About Us
- 1. Ansible
- 2. PyPI package
- 3. Installation
- 4. Administration
- 5. Testing
- 6. Repository
- 7. Future Work
0. About Us

- Gerald Wiese, M. Sc. in computer science
- Computational Health Informatics (CHI) & GNU Health:
  - Prof. Dr. Gabriele von Voigt: Head of the group
  - Dr. Hans Georg Krojanski: Feedback on Ansible project
  - Hilke Brodersen: Support logistics of conference

- Julian Menon, Jonathan Feilmeier, Lucas Kiesel: Update GNU Health documentation, testing of Ansible

- Patryk Rosik: Master thesis about Orthanc integration
- Julian Menon: Bachelor thesis about Monitoring
- Florian Liermann: Master thesis about DHIS 2 module
1. Ansible - Advantages

- Very big and active community
- Only Python & SSH needed on target systems
- YAML format is human-readable
- Easy to learn but extremely scalable
- Reproducability & Idempotence
1. Ansible - Playbooks & Roles

- Playbooks: „Scripts“, list of tasks for manipulating systems

- Roles: Modular reusable structure with its own tasks, variables, templates, etc.

- The playbook gnuhealth_minimal.yml contains the roles postgresql, gnuhealth, uwsgi, nginx & gnuhealth-client
  All playbooks for servers in this project use the same roles postgresql & nginx
1. Ansible - Goals of this project

- **Easy** installation
- Minimize & hide OS differences
- **Modularity**: Many scenarios & configuration options prepared
- Guidance for non experienced system administrators
2. PyPI package

- Independance from package maintainers:
  - Available for every OS – package only on openSUSE
  - Patchsets available without delay
  - Dependencies more uniform
- Config snippets shipped for GNU Health itself, uWSGI, Nginx, Systemd – for use without Ansible
- How-to for uploading, testing, GPG signatures, SHA256 sums

→ Add to Mercurial repository?
→ Co-maintainers wanted

https://pypi.org/project/gnuhealth-all-modules/
https://gitlab.com/geraldwiese/gnu-health-all-modules-pypi/
3. Installation - GNU Health HMIS

- **GNU Health HMIS**: Hospital Management Information System
  - Server: PostgreSQL, GNU Health PyPI package, uWSGI, Nginx
  - Desktop: GNU Health Client

- **PostgreSQL**: Database backend
- **uWSGI**: Application server, executing GNU Health
- **Nginx**: Web server / reverse proxy, providing encryption and performance optimization

- Tested OS: Debian 12, all Ubuntu LTS, openSUSE Leap 15.5, FreeBSD 13.2
3. Installation - What is different?

- Add Nginx & uWSGI in front
- Use virtual python environment and PyPI package
- Restrictive access permissions
- Introduced new directory structure for all OS
- Systemd services for Tryton Cron & WebDAV
- Consider separate systems
  ...

...
3. Installation – .gnuhealthrc & gnuhealth-control

- User gnuhealth: 
  - `.bashrc` loads aliases from `.gnuhealthrc`:
    - Traditional commands like `cdlogs` & `editconf`
    - `cdexe` replaced by `activate`

- Updated `gnuhealth-control` command:
  - Traditional: Save database dump, home folder and combination
  - Updated: Save database dump, attachments folder, config folder & pip freeze output
3. Installation - Thalamus & Orthanc

- **Thalamus** shares all roles with HMIS (except *gnuhealth* & *gnuhealth-client*)
- Thalamus role itself is very similar to *gnuhealth*

- **Orthanc** is installed as system package
- Does not need the *uWSGI* role because it has a HTTP server
- Configuration options for e.g. plugins and demo users
4. Administration - Overview
4. Administration - Backup/Restore

- Remote / separate system:
  - **Barman** for PostgreSQL: Handle WAL streaming, base backups, recovery, etc. Allows RPO=0 if synchronous
  - **Restic** for folders on application servers: Encryption, snapshots, deduplication, etc. RPO>0

- Local:
  - Adapted traditional `gnuhealth-control` command
  - Either to attached device or as base for remote
4. Administration - Barman

- Written in Python
- License: GPLv3
- Developed and maintained by EnterpriseDB

https://docs.pgbarman.org/release/3.8.0/
4. Administration - Certificates

- **Certificate Authority (CA):** Manage certificates, sign with private key, distribute CA certificate for trust

- Distribute and use **server certificates:**
  - Issued by custom CA
  - Issued by Let’s Encrypt
  - Existing, only set paths

- Clients: Trust CA as base for **verification**
4. Administration - Other Roles

- SSH: Create SSH keys, trust SSH keys & host keys, etc.
- Nftables: Firewall rules
- Fail2Ban: Ban malicious IPs, protect against BF & DDoS
- sSMTP: Realize scriptable email access
- systemd-alert: Send email if important service stops
- Time zone
- Unattended Upgrades: Automatic updates for Debian
- GPG: Create GPG key for crypto plugin of GH Client
- Check: Update package manager cache, check OS distr.
5. Testing

- Tests based on simple **BASH** scripts:
  - Ansible Lint
  - Config stock
  - Variable consistency
  - Reuse compliance (licensing)

- Integration tests based on **Molecule, Vagrant & VirtualBox**:
  - Installation: Test GH HMIS, Thalamus & Orthanc for Debian/Ubuntu & openSUSE Leap (FreeBSD soon)
  - Administration: Test GH & Orthanc for Debian/Ubuntu
6. Repository – Mercurial vs. GitLab

- Published in official GNU Health repository & documentation:
  - https://hg.savannah.gnu.org/hgweb/health-ansible/
  - https://docs.gnuhealth.org/ansible/

- Development still on GitLab, benefit from CI tests:
  - https://gitlab.com/geraldwiese/gnuhealth-automatic-deployment

- Different examples in documentation for quickstart
- Testing, contributing & maintainers welcome
7. Future Work

- Last planned set of functionalities:
  - Monitoring: Logwatch & Zabbix
  - Postfix mailgate as alternative to sssmtp
  - Test server for Molecule tests

- Continue Polishing, Test cases & Documentation

- Overtake as default installation strategy with GH HMIS 4.4
- Provide ready-to-go VMs
GNU Health Documentation Portal

**GH::HMIS**
Hospital Management Information System
[https://docs.gnuhealth.org/hmis](https://docs.gnuhealth.org/hmis)

**MyGNUHealth**
Personal Health Record (PHR)
[https://docs.gnuhealth.org/mygnuhealth](https://docs.gnuhealth.org/mygnuhealth)

**THALAMUS**
GNU Health Federation Message and Authentication Server
[https://docs.gnuhealth.org/thalamus](https://docs.gnuhealth.org/thalamus)

**GH::HMIS**
GNU Health components deployment using Ansible
[https://docs.gnuhealth.org/ansible](https://docs.gnuhealth.org/ansible)

[https://docs.gnuhealth.org/](https://docs.gnuhealth.org/)