Integrating Uyuni with Ansible and Terraform

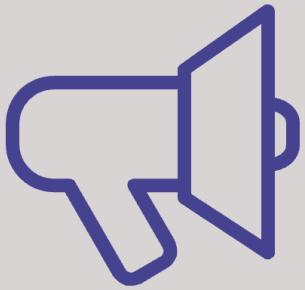
`whoami`

Christian Stankowic

- IT consultant and trainer at SVA
  - SUSE, Red Hat, Debian, etc.
  - Patch management
  - Uyuni project contributor
  - IaC, Automation, DevO(o)ps
  - Terraform, Packer, Vagrant,...
- 📔 cstan.io
- 🎙️🇩🇪 (Co-)host of various podcasts

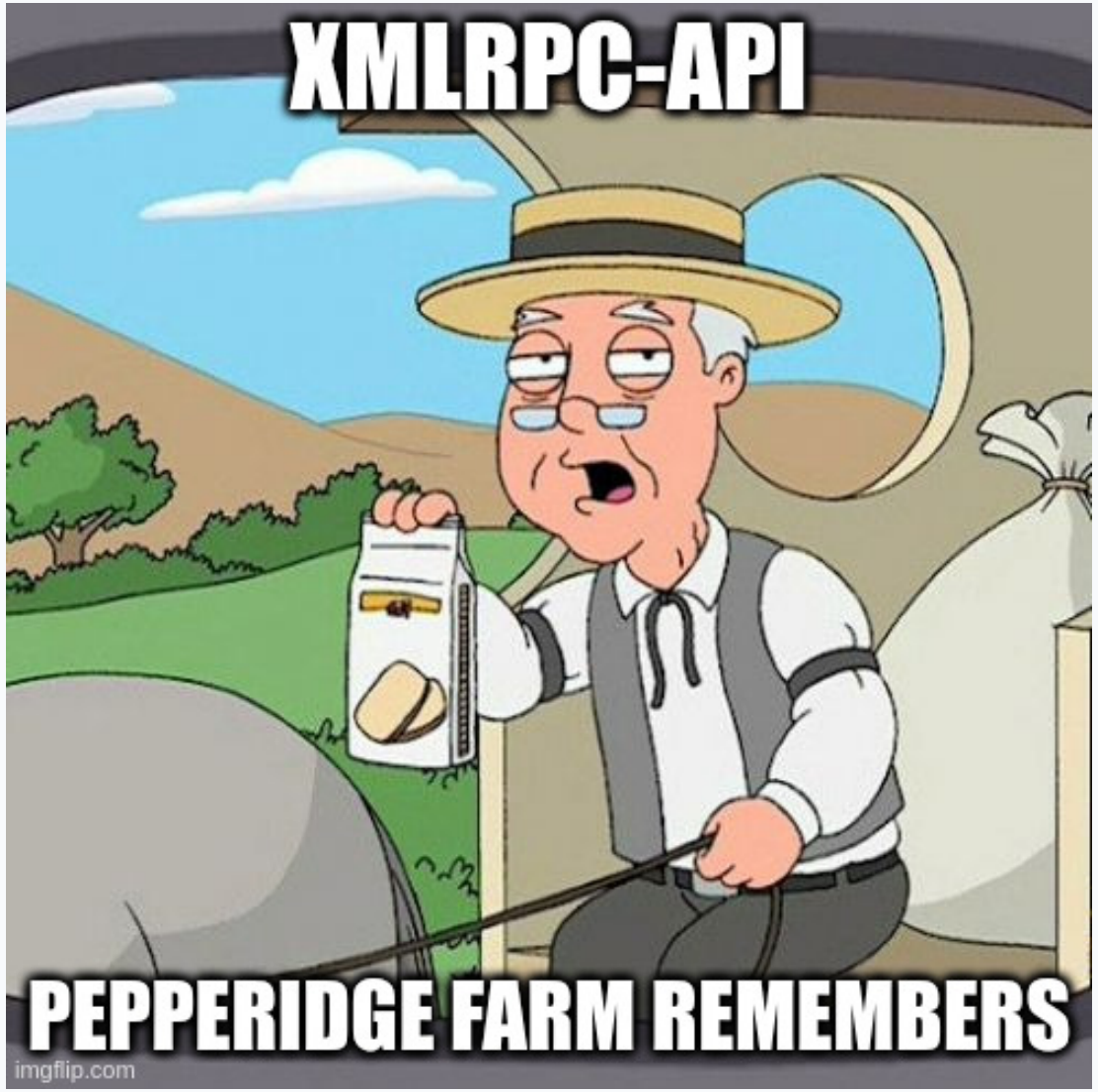
Agenda

1. Recap: Uyuni automation capabilities
2. The Uyuni Ansible collection
3. Terraform integration MVP
4. Uyuni API in the future?



Recap: Automation 🚀

Recap: Uyuni automation capabilities



Recap: Uyuni automation capabilities

- **XML-RPC-API**
  - good ol' default API
  - Introduced in **Spacewalk** (2008), hard to use, requires special client libraries
- **HTTP API**
  - New JSON over HTTP API, easier to consume
  - Introduced in Uyuni 2022.05, supported in SUSE Manager 4.3

Recap: Uyuni automation capabilities

- `spacecmd`
  - Command-line interface to API
  - Many commands support wildcards, offers shell mode
- **Salt**
  - Mighty IaC solution for **comprehensive** configuration management

Salt vs. Ansible

- Uyuni uses SaltStack as **integral** component
- optional Ansible integration beginning with Uyuni 2021.06 and SUSE Manager 4.2

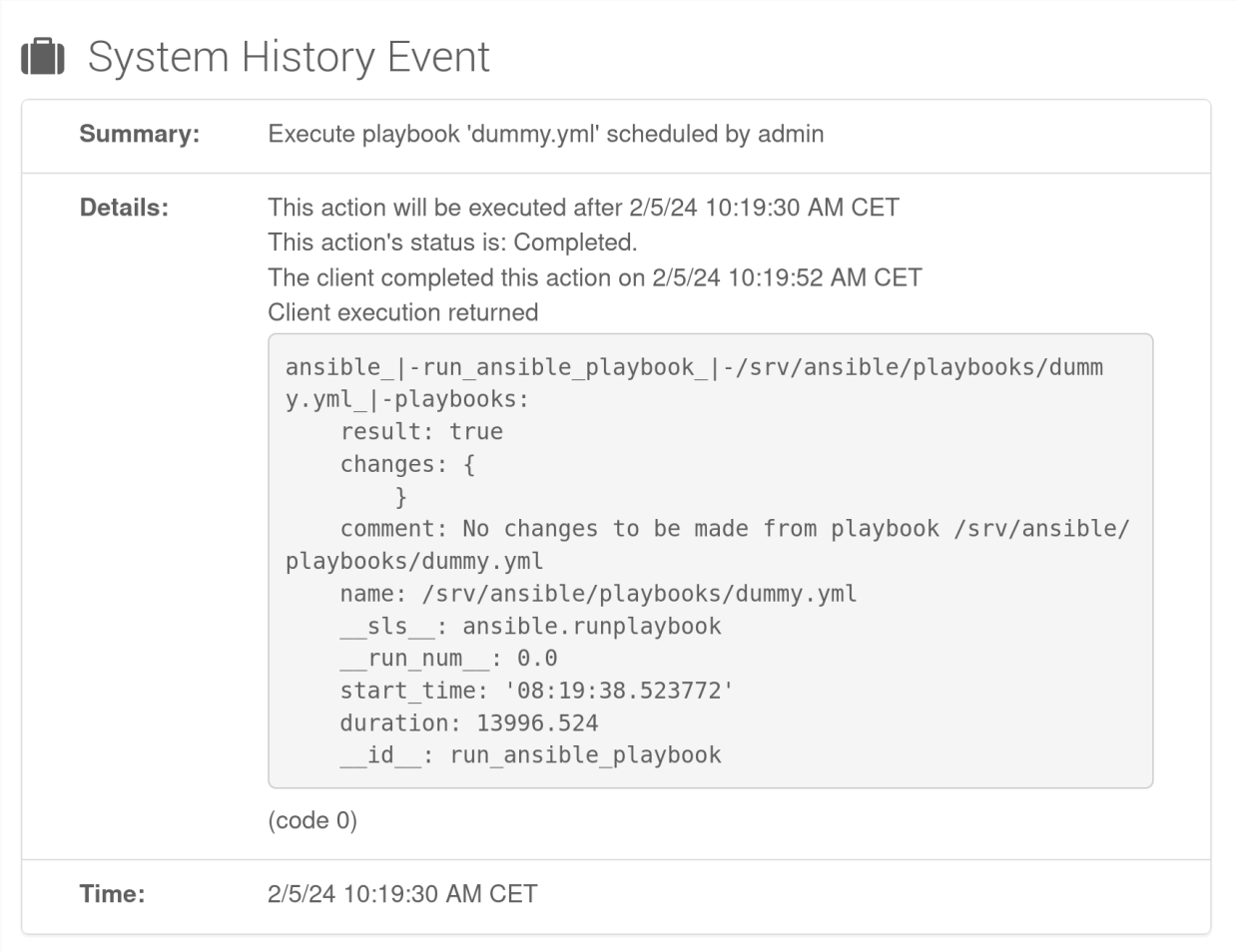
Salt vs. Ansible

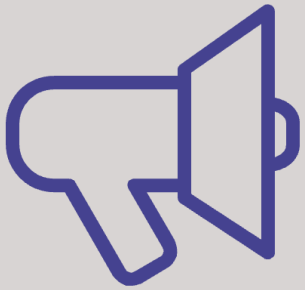
	Ansible	**SaltStack**
Initial release	2012	2011
Agent	No	Yes (SSH-only option)
Networking	Push	Push
Language	YAML, Python	YAML, Python
Syntax level	easy	advanced
Community	Ansible Galaxy	SaltStack Formulas
	~20.000 members	~50 members
	3.5k+ collections, 35k+ roles	350 formulas

- both IaC tools are **very capable** and have their **use cases**
- anyhow, the majority uses **Ansible** because of the broader range of predefined code

Uyuni / SUMA Ansible Integration

- managed hosts can have the **Ansible Control Node** add-on type
- using this, they can run Ansible playbooks - triggered by Salt
- supported with SUSE Manager 4.3
- limited usability and **outdated** (*2.9*)
  - newer versions in the future
  - confirmed for SUSE MLM 5.1
- usually Ansible infra already exist





Ansible collection

The Uyuni Ansible collection

- Collection of various **roles**
  - `server` (*Uyuni/SUMA installation*)
  - `proxy` (*Proxy server*)
  - `client` (*bootstrapping*)
- Also contains various **plugins**
  - Managing patches, upgrades
  - Running OpenSCAP checks
  - Dynamic inventory

The Uyuni Ansible collection

Available on GitHub and Ansible Galaxy:

https://github.com/stdevel/ansible-collection-uyuni

https://galaxy.ansible.com/ui/repo/published/stdevel/uyuni/

Contributions / feedback welcome!

Installing SUSE MLM in under 10 minutes

```
---
- name: Install SUSE Manager
  hosts: suma.giertz.loc
  become: true
  roles:
    - role: stdevel.uyuni.server
      uyuni_scc_reg_code: DERP1337LULZ
      uyuni_scc_mail: simone@gier.tz
```

- installing the most recent supported version of SUSE (Multi-Linux) Manager
- automatically registers the system with SCC in order to retrieve packages
- supports both SL(E) Micro and SLES 15 SP6+

Installing Uyuni and 3rd party repositories

```
---
- name: Install Uyuni
  hosts: uyuni.giertz.loc
  become: true
  roles:
    - role: stdevel.uyuni.server
      uyuni_release: '2024.12'
      uyuni_channels:
        - name: opensuse_leap15_6
          arch: x86_64
```

- installs a **specific** Uyuni version
- configures 3rd party repositories using `spacewalk-common-channels`

Registering clients

```
---
- name: Register clients
  hosts: clients
  become: true
  roles:
    - role: stdevel.uyuni.client
```

Downloads and executes a `bootstrap.sh` script.

By default, it searches for the file `bootstrap-${distro}${version}.sh` on the Uyuni host (below the `/pub/bootstrap` web directory) – e.g. `bootstrap-almalinux9.sh`.

See also SUSE Manager documentation.

Installing patches

```
---
- name: Installing patches
  stdevel.uyuni.install_patches:
    uyuni_host: 192.168.1.1
    uyuni_user: simone.giertz
    uyuni_password: shittyr0b0ts
    name: myserver.localdomain.loc
    exclude_patches:
      - openSUSE-2022-10013
      - openSUSE-SLE-15.3-2022-2118
```

Parameters specify MLM / Uyuni **connection details** and managed host.

Installs **all available** patches by default. Can also be explicitly included (`include_patches`) or excluded (`exclude_patches`).

Installing patches

```
---
- name: Patch management
  hosts: myservers
  tasks:
    - name: Installing patches
      stdevel.uyuni.install_patches:
        uyuni_host: 192.168.1.1
        uyuni_user: sebastian
        uyuni_password: linuxtechtips
        name: "{{ item }}"
      delegate_to: localhost
      loop: "{{ groups['myservers'] | list }}"
```

Can also be used with the **dynamic inventory** (`hosts`) and delegating the task to the management node (`delegate_to`).

Checking OpenSCAP compliance

```
---
- name: Check compliance
  stdevel.uyuni.openscap_run:
    uyuni_host: 192.168.1.1
    uyuni_user: notahacker
    uyuni_password: securitay
    name: myserver.localdomain.loc
    document: /opt/scap-yast2sec-xccdf.xml
    arguments: --profile Default
```

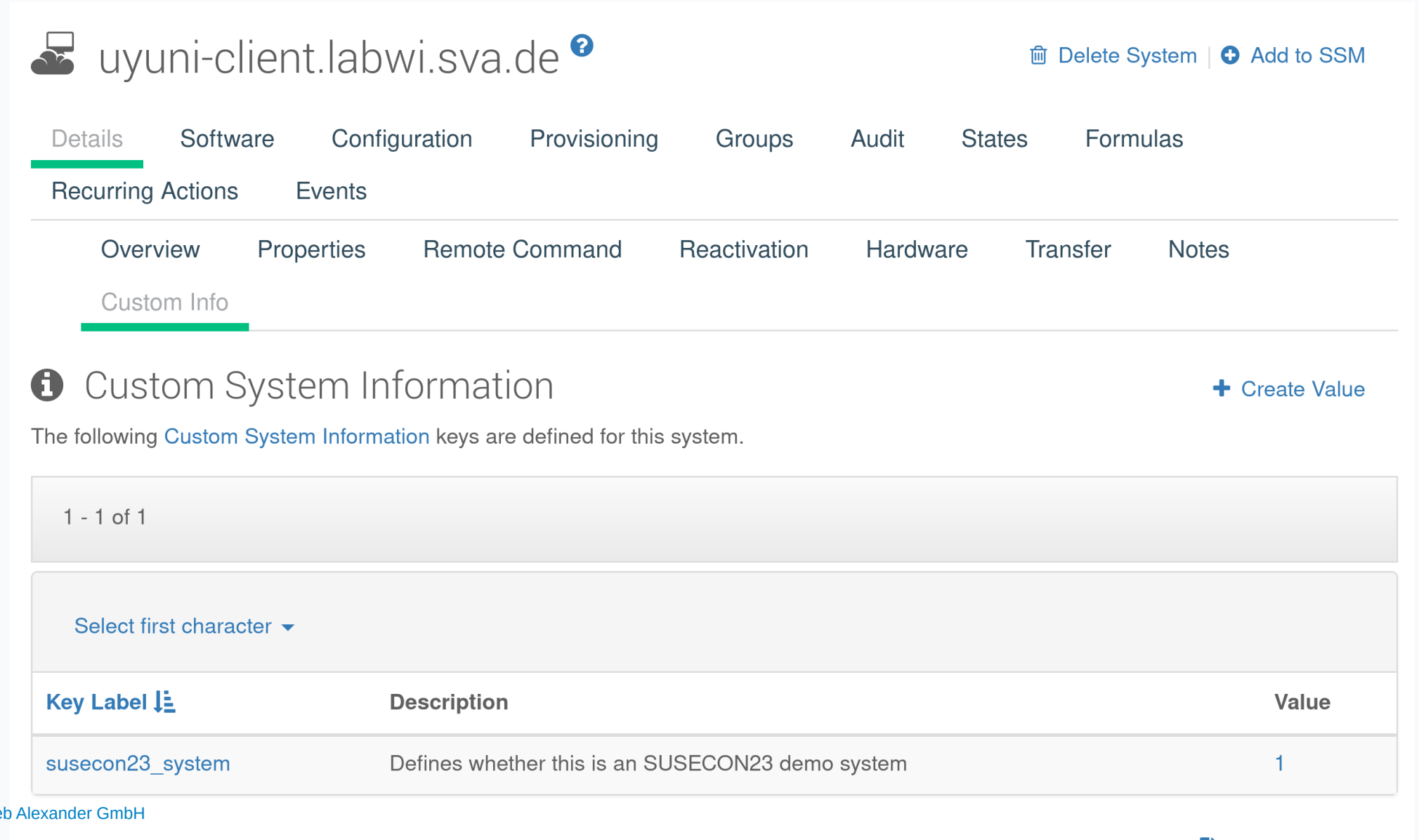
XCCDF catalog (`document`) and `oscap` parameters (`arguments`) are defined using **parameters**.

Dynamic inventory: configuration

```yaml
---
plugin: stdevel.uyuni.inventory
host: 192.168.1.1
user: admin
password: admin
show_custom_values: true
# ipv6_only: true
verify_ssl: false
# groups:
#   - Test
pending_reboot_only: true
```

- 🚨 Inventory file name **postfix**: `.uyuni.yml` 🚨
- Configuration file specifies:
  - Connection details
  - Hostgroup / custom value / IPv6 filter
  - Filter for pending reboots

Dynamic inventory: usage



Dynamic inventory: usage

```
$ ansible-inventory homelab.uyuni.yml --list
{
    "Test": {
        "hosts": [
            "uyuni-client.labwi.sva.de"
    ]
    },
    "_meta": {
        "hostvars": {
            "uyuni-client.labwi.sva.de": {
            "ansible_host": "192.168.1.2",
            "susecon23_system": "1"
            }
        }
    }
...
}
```

- inventory includes **hostgroups** and custom **system information**
- can be used as regular Ansible **variables**

Fully automated system maintenance

Automated patch cycle in the **lab**:

- **VMware** vSphere hypervisor
- **check_mk** for monitoring
- Uyuni

AWX uses various Ansible **collections**:

- `checkmk.general`
- `community.vmware`
- `stdevel.uyuni`

The AWX workflow will **prepare** maintenance by:

- Scheduling downtime
- Creating VM spashots

Afterwards it will **install** patches.

Code

`install_patches.yml`

```
- name: Schedule patch installation
  stdevel.uyuni.install_patches:
    uyuni_host: "{{ uyuni_hostname }}"
    uyuni_user: "{{ uyuni_username }}"
    uyuni_password: "{{ uyuni_password }}"
    uyuni_verify_ssl: false
    name: "{{ item }}"
    include_patches: "{{ include_patches | default(omit) }}"
  delegate_to: localhost
  loop: "{{ groups['all'] | list }}"
```

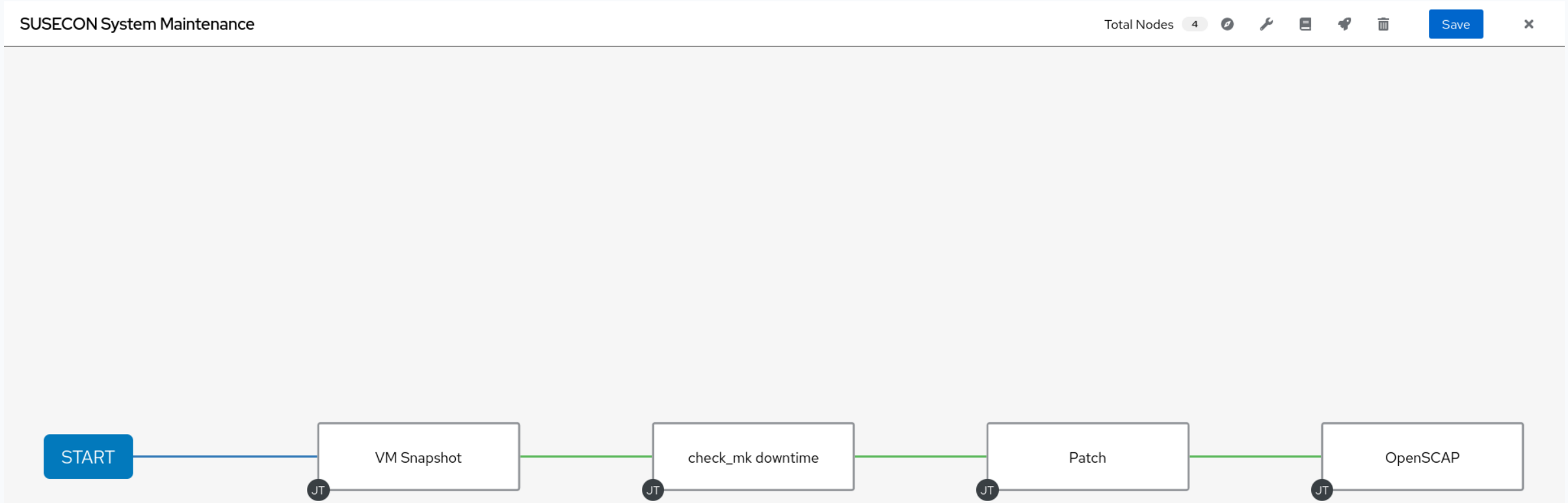
Installs available patches on all hosts, if not overwritten.

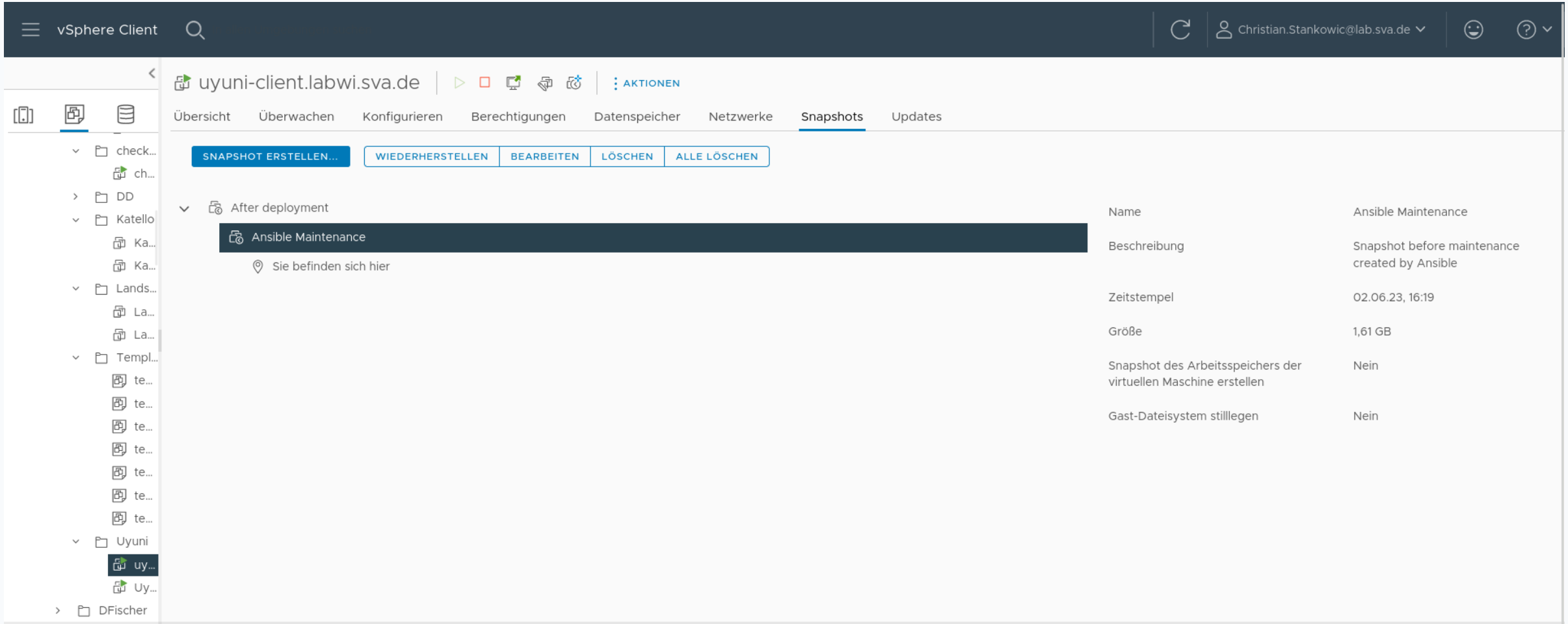
Code

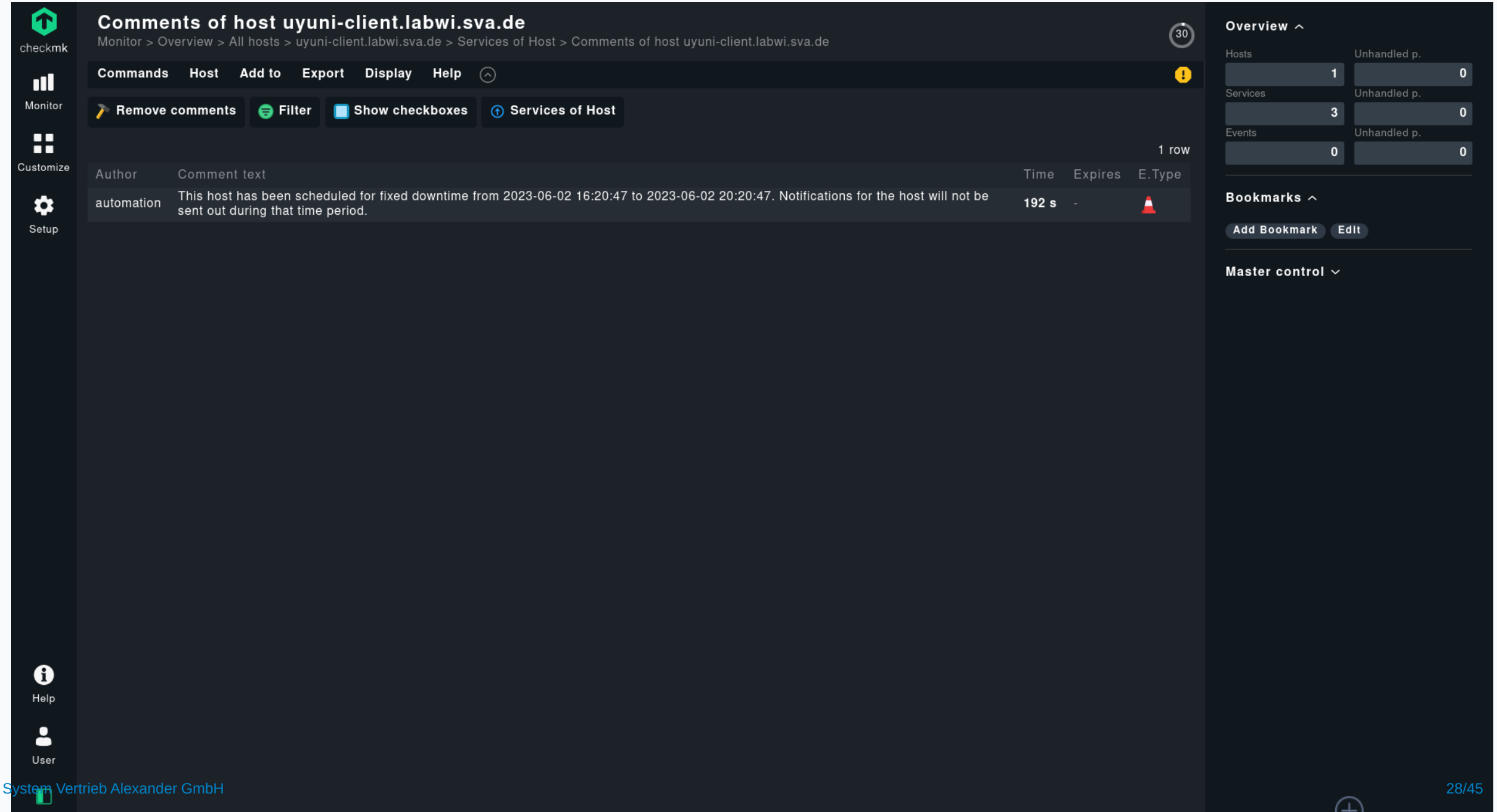
`run_openscap.yml`

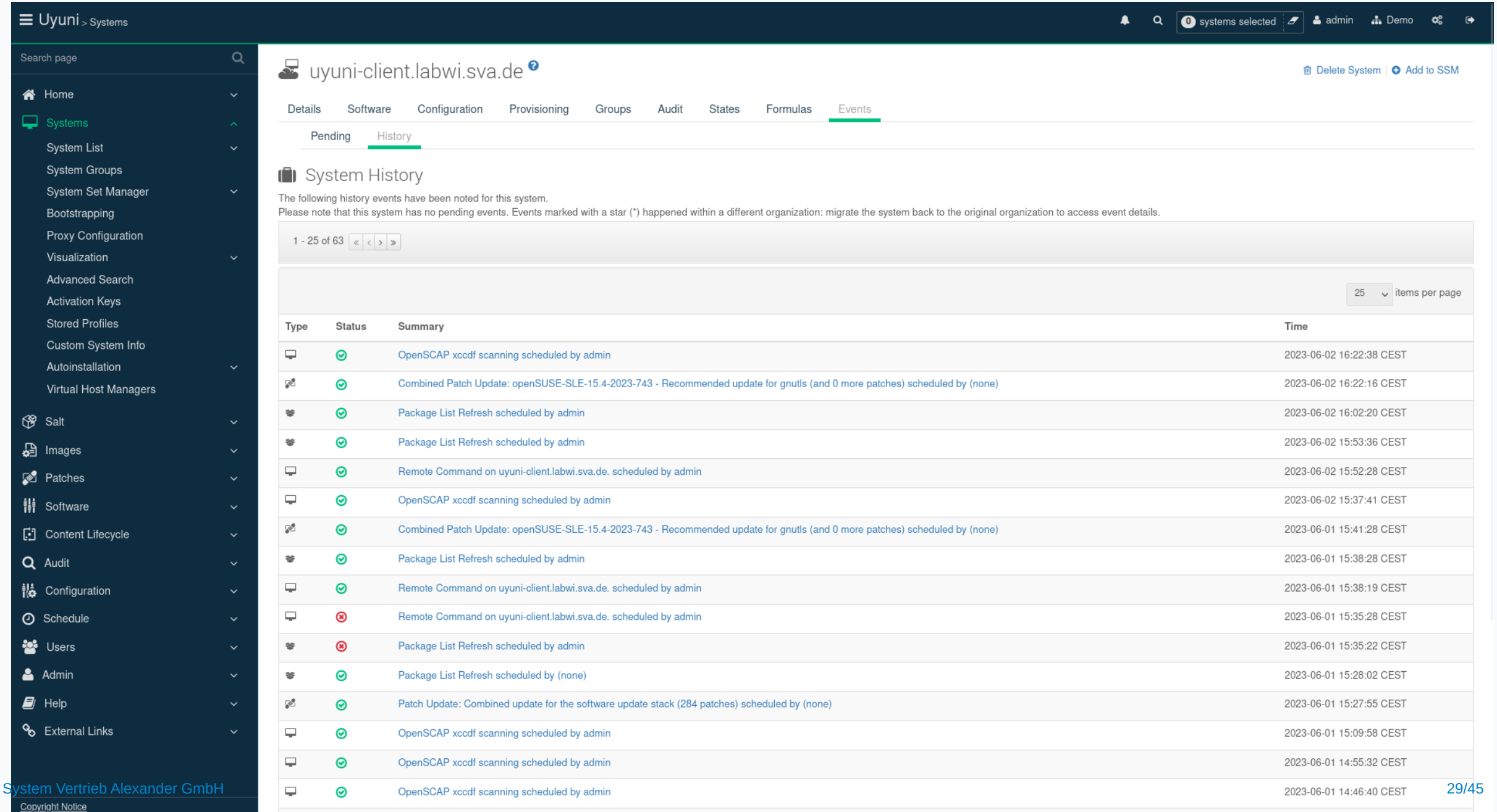
```
- name: Schedule OpenSCAP run
  stdevel.uyuni.openscap_run:
    uyuni_host: "{{ uyuni_hostname }}"
    uyuni_user: "{{ uyuni_username }}"
    uyuni_password: "{{ uyuni_password }}"
    uyuni_verify_ssl: false
    name: "{{ item }}"
    document: /usr/share/openscap/scap-yast2sec-xccdf.xml
    arguments: --profile Default
  delegate_to: localhost
  loop: "{{ groups['all'] | list }}"
```

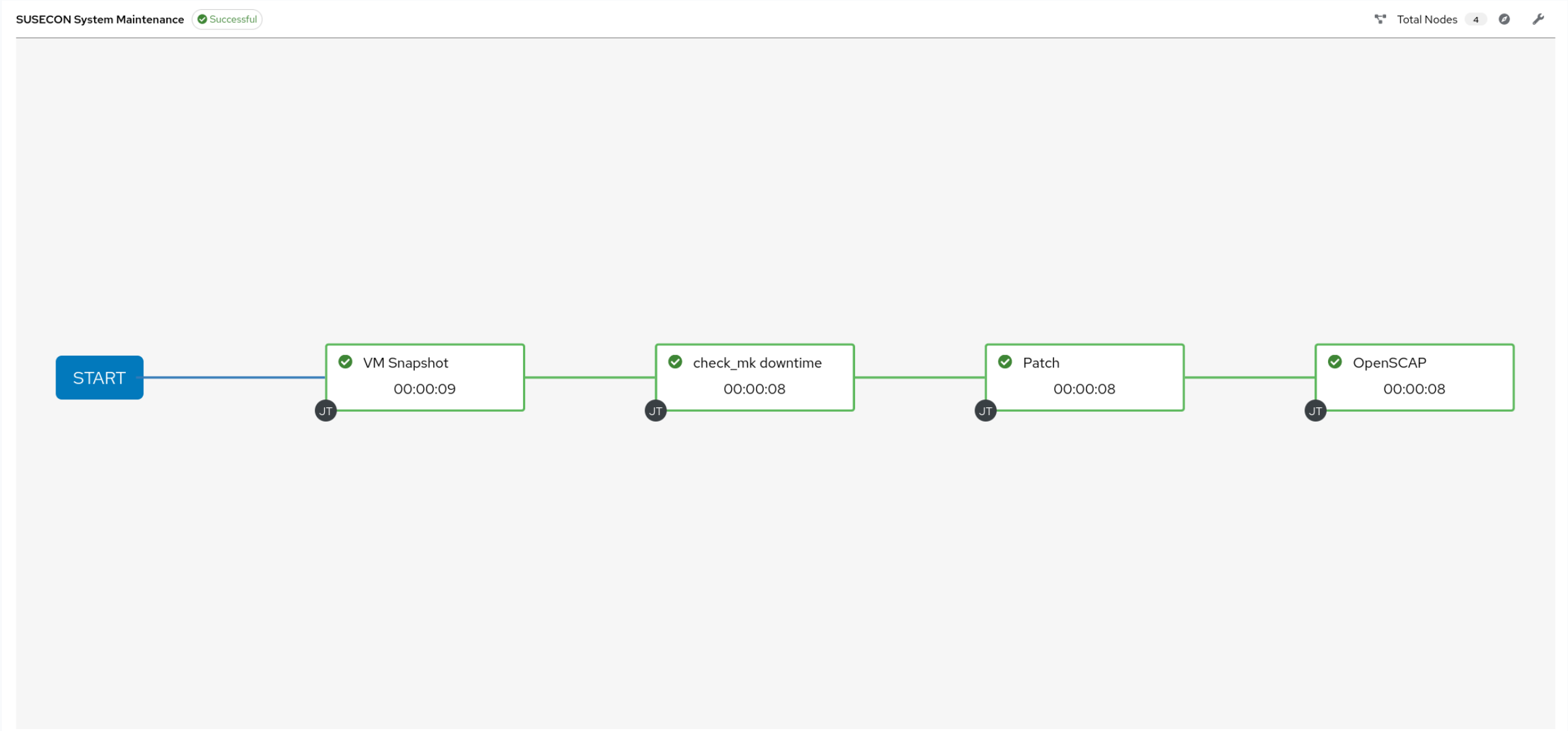
Runs a specific example catalog on all hosts.


SUSECON System Maintenance
Total Nodes
4
Save
START
VM Snapshot
JT
check_mk downtime
JT
Patch
JT
OpenSCAP
JT


vSphere Client
Christian.Stankowic@lab.sva.de
uyuni-client.labwi.sva.de
AKTIONEN
Übersicht
Überwachen
Konfigurieren
Berechtigungen
Datenspeicher
Netzwerke
Snapshots
Updates
SNAPSHOT ERSTELLEN...
WIEDERHERSTELLEN
BEARBEITEN
LÖSCHEN
ALLE LÖSCHEN
After deployment
Ansible Maintenance
Sie befinden sich hier
Name
Ansible Maintenance
Beschreibung
Snapshot before maintenance created by Ansible
Zeitstempel
02.06.23, 16:19
Größe
1,61 GB
Snapshot des Arbeitsspeichers der virtuellen Maschine erstellen
Nein
Gast-Dateisystem stilllegen
Nein
Kürzlich bearbeitete Aufgaben
Alarme
Name der Aufgabe
Ziel
Status
Details
Initiator
In Warteschlange
Startzeit
Zeitpunkt der Fertigstellung
Server
Keine Elemente gefunden


checkmk
Monitor
Customize
Setup
Help
User
Comments of host uyuni-client.labwi.sva.de
Monitor > Overview > All hosts > uyuni-client.labwi.sva.de > Services of Host > Comments of host uyuni-client.labwi.sva.de
30
Commands
Host
Add to
Export
Display
Help
Remove comments
Filter
Show checkboxes
Services of Host
1 row
Author
Comment text
Time
Expires
E.Type
automation
This host has been scheduled for fixed downtime from 2023-06-02 16:20:47 to 2023-06-02 20:20:47. Notifications for the host will not be sent out during that time period.
192 s
-
Overview
Hosts
1
Unhandled p.
0
Services
3
Unhandled p.
0
Events
0
Unhandled p.
0
Bookmarks
Add Bookmark
Edit
Master control


Uyuni > Systems
0 systems selected
admin
Demo
Search page
Home
Systems
System List
System Groups
System Set Manager
Bootstrapping
Proxy Configuration
Visualization
Advanced Search
Activation Keys
Stored Profiles
Custom System Info
Autoinstallation
Virtual Host Managers
Salt
Images
Patches
Software
Content Lifecycle
Audit
Configuration
Schedule
Users
Admin
Help
External Links
Copyright Notice
uyuni-client.labwi.sva.de
Delete System | Add to SSM
Details Software Configuration Provisioning Groups Audit States Formulas Events
Pending History
System History
The following history events have been noted for this system.
Please note that this system has no pending events. Events marked with a star (*) happened within a different organization: migrate the system back to the original organization to access event details.
1 - 25 of 63
25 items per page
Type Status Summary Time
OpenSCAP xccdf scanning scheduled by admin 2023-06-02 16:22:38 CEST
Combined Patch Update: openSUSE-SLE-15.4-2023-743 - Recommended update for gnutls (and 0 more patches) scheduled by (none) 2023-06-02 16:22:16 CEST
Package List Refresh scheduled by admin 2023-06-02 16:02:20 CEST
Package List Refresh scheduled by admin 2023-06-02 15:53:36 CEST
Remote Command on uyuni-client.labwi.sva.de. scheduled by admin 2023-06-02 15:52:28 CEST
OpenSCAP xccdf scanning scheduled by admin 2023-06-02 15:37:41 CEST
Combined Patch Update: openSUSE-SLE-15.4-2023-743 - Recommended update for gnutls (and 0 more patches) scheduled by (none) 2023-06-01 15:41:28 CEST
Package List Refresh scheduled by admin 2023-06-01 15:38:28 CEST
Remote Command on uyuni-client.labwi.sva.de. scheduled by admin 2023-06-01 15:38:19 CEST
Remote Command on uyuni-client.labwi.sva.de. scheduled by admin 2023-06-01 15:35:28 CEST
Package List Refresh scheduled by admin 2023-06-01 15:35:22 CEST
Package List Refresh scheduled by (none) 2023-06-01 15:28:02 CEST
Patch Update: Combined update for the software update stack (284 patches) scheduled by (none) 2023-06-01 15:27:55 CEST
OpenSCAP xccdf scanning scheduled by admin 2023-06-01 15:09:58 CEST
OpenSCAP xccdf scanning scheduled by admin 2023-06-01 14:55:32 CEST
OpenSCAP xccdf scanning scheduled by admin 2023-06-01 14:46:40 CEST


SUSECON System Maintenance
Successful
Total Nodes
4
START
VM Snapshot
00:00:09
JT
check_mk downtime
00:00:08
JT
Patch
00:00:08
JT
OpenSCAP
00:00:08
JT

Event-driven Ansible

- By now, Ansible has been working **reactively**
  - Whenever a problem occurs, you run a playbook to **fix** it
- One of SaltStacks **advantages** is the ability to **proactively** react on changes
  - e.g. hard-disk is extended **before** the database fills up space

- People wanted to have the same for **Ansible**
  - = **E**vent-**d**riven **A**nsible
- **Rulebooks** define **events** to be **monitored**
  - can be filtered, **actions** define countermeasures

EDA + Uyuni

- EDA support was started in a recent hackathon
- **early** stage, **feedback** wanted
- PoC: can react on systems requiring a reboot and trigger actions
  - e.g. running a playbook to reboot the host
- **additional use-cases** possible:
  - handle unresponsive systems
  - react to specific system events (*harden insecure systems*)
  - your ideas?

Code

`trigger_reboots.yml`

```
---
- name: Rebooting hosts
  hosts: localhost
  tasks:
    - name: Show system that will be rebooted
      ansible.builtin.debug:
        msg: "Host to be rebooted: {{ ansible_eda.event.host }}"

    - name: Reboot system
      stdevel.uyuni.reboot_host:
        uyuni_host: "{{ uyuni_hostname }}"
        uyuni_user: "{{ uyuni_username }}"
        uyuni_password: "{{ uyuni_password }}"
        uyuni_verify_ssl:
        name: "{{ ansible_eda.event.host }}"
```

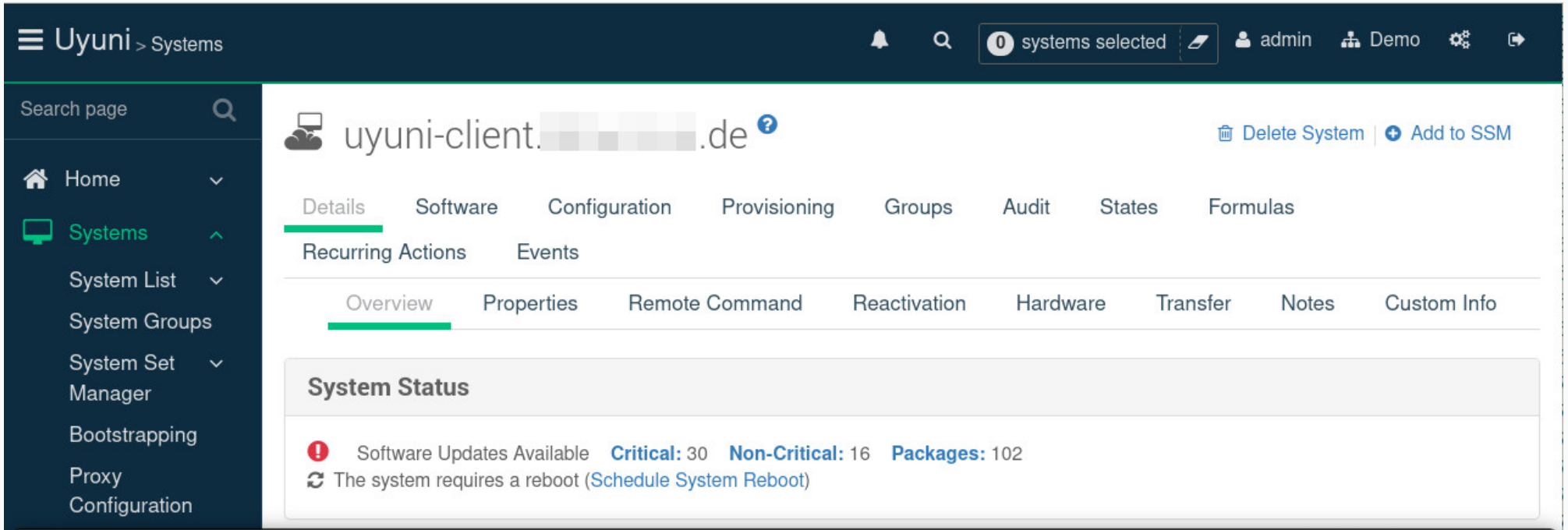
Running

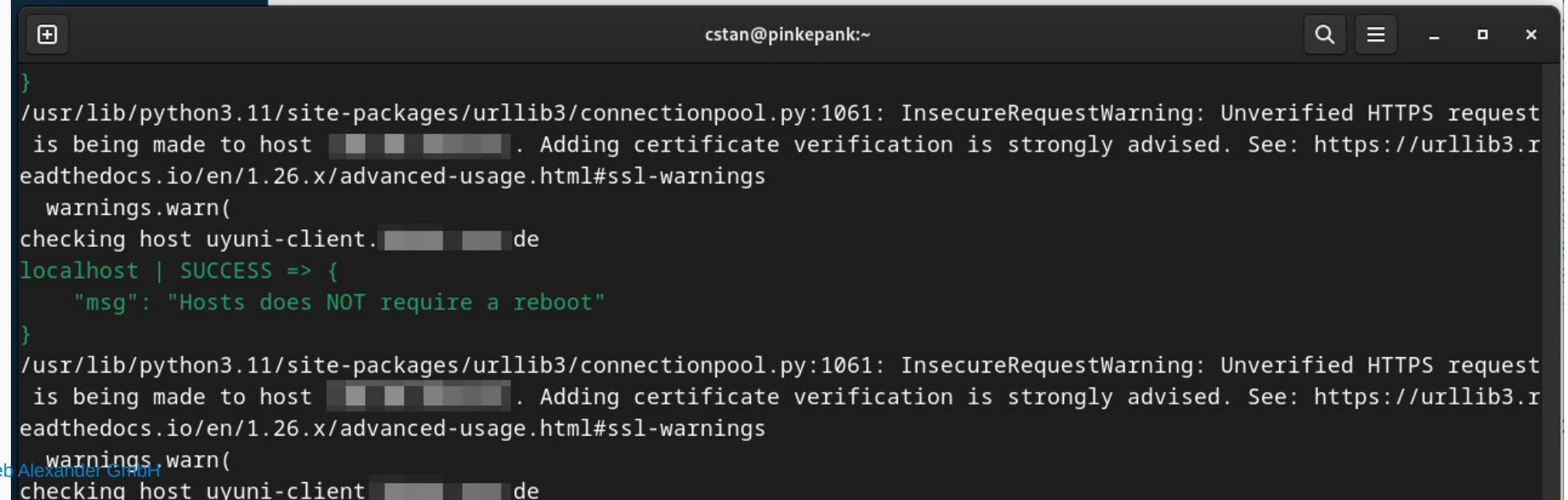
```
$ ansible-rulebook -i inventory.ini --rulebook trigger_reboo
...
checking host uyuni-client.xxx.de

PLAY [Rebooting hosts] ***************************************

TASK [Show system that will be rebooted] *********************
ok: [localhost] => {
    "msg": "Host to be rebooted: uyuni-client.xxx.de"
}

TASK [Reboot system] *****************************************
changed: [localhost]
```


Uyuni > Systems
0 systems selected
admin
Demo
Search page
Home
Systems
System List
System Groups
System Set Manager
Bootstrapping
Proxy Configuration
uyuni-client.[redacted].de
Delete System
Add to SSM
Details
Software
Configuration
Provisioning
Groups
Audit
States
Formulas
Recurring Actions
Events
Overview
Properties
Remote Command
Reactivation
Hardware
Transfer
Notes
Custom Info
System Status
Software Updates Available Critical: 30 Non-Critical: 16 Packages: 102
The system requires a reboot (Schedule System Reboot)


cstan@pinkepank:~
}
/usr/lib/python3.11/site-packages/urllib3/connectionpool.py:1061: InsecureRequestWarning: Unverified HTTPS request
is being made to host [redacted]. Adding certificate verification is strongly advised. See: https://urllib3.r
eadthedocs.io/en/1.26.x/advanced-usage.html#ssl-warnings
warnings.warn(
checking host uyuni-client.[redacted]de
localhost | SUCCESS => {
"msg": "Hosts does NOT require a reboot"
}
/usr/lib/python3.11/site-packages/urllib3/connectionpool.py:1061: InsecureRequestWarning: Unverified HTTPS request
is being made to host [redacted]. Adding certificate verification is strongly advised. See: https://urllib3.r
eadthedocs.io/en/1.26.x/advanced-usage.html#ssl-warnings
warnings.warn(
checking host uyuni-client.[redacted]de

Terraform Provider

The Uyuni Terraform Provider

- Broad range of providers for Terraform and OpenTofu
  - currently **~4.780** providers
- Good coverage of mainstream and also niche integrations

- Some things are still missing
  - Uyuni
  - Forgejo
- Another hackathon nerd-sniped us into writing providers

Example: Uyuni users

- A very **first MVP** was created in a two-day Hackathon
- Creates `user` objects
  - `login`, `firstname`, `lastname`, `email`, `password`
- Pinned [experimental Uyuni API] package as dependency ¯\(ツ)/¯
- Feel free [to contribute]

Example: Uyuni users

```
resource "uyuni_user" "sgiertz" {
  login = "sgiertz"
  firstname = "Simone"
  lastname = "Giertz"
  email = "sgiertz@foo.bar"
  password = "fakuchad"
}

output "users" {
  value = data.uyuni_users.my_users
}
```

Example: Uyuni users

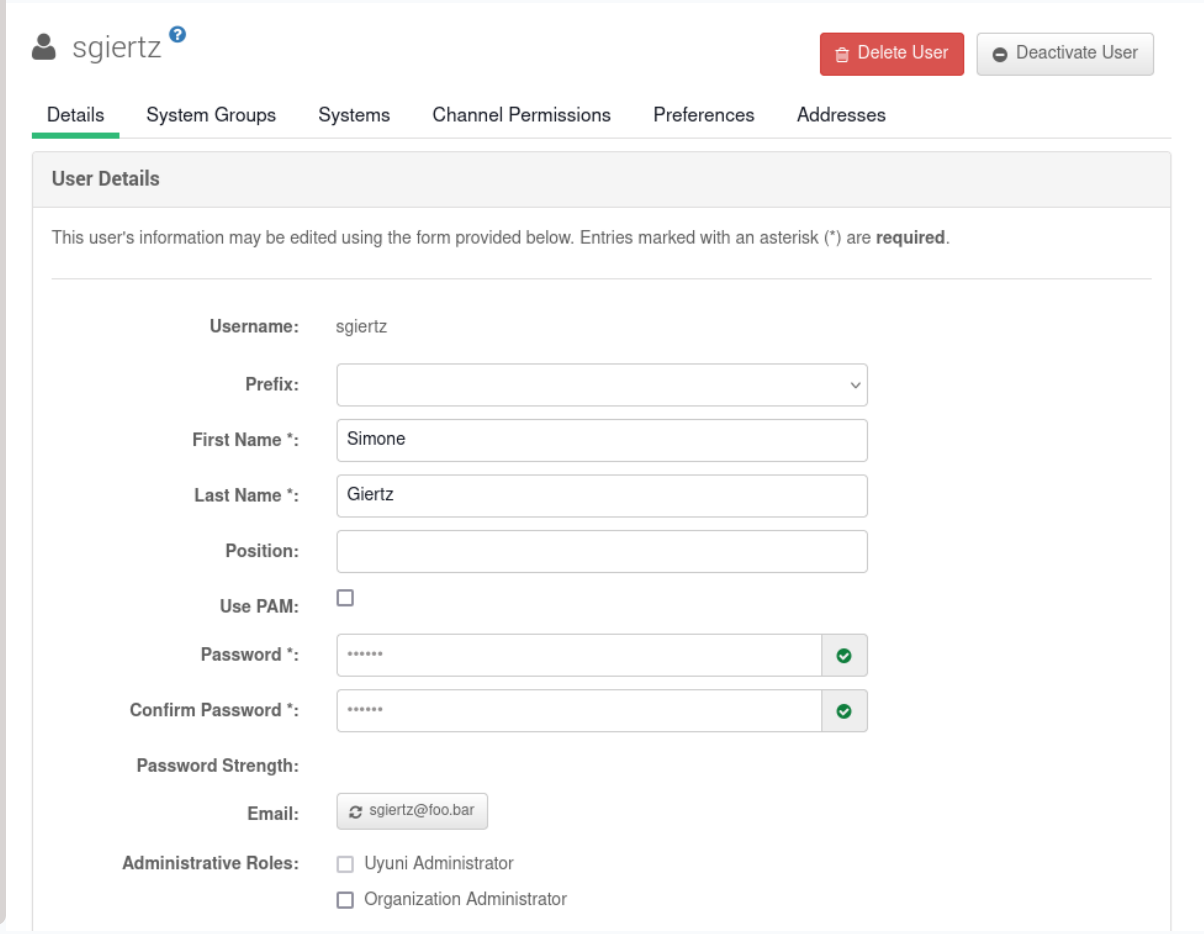
```
$ terraform apply
...
  # uyuni_user.sgiertz will be created
  + resource "uyuni_user" "sgiertz" {
      + email     = "sgiertz@foo.bar"
      + firstname = "Simone"
      + lastname  = "Giertz"
      + login     = "sgiertz"
      + password  = (sensitive value)
    }

Plan: 1 to add, 0 to change, 0 to destroy.


Do you want to perform these actions?
...

  Enter a value: yes

uyuni_user.sgiertz: Creating...
uyuni_user.sgiertz: Creation complete after 1s
```



Future of Uyuni API?

Future of Uyuni API?

- In December 2024, a RFC for leveraging OpenAPI definitions was **accepted**
  - widely recognized standard
  - robust **ecosystem** of tools, e.g.
    - **client generators**
    - testing frameworks
- Huge **game changer**
  - will describe the HTTP API in a way that can be used by Swagger
  - easier API exploration for new developers and users
  - automatic client generation

Future of Uyuni API?

- The Foreman project uses `apipie` and `apinsible` for developing Ansible collections
  - Jinja2 for templating
  - collection scraffolding using `ansible.content builder`
  - manual development is too big, automation where possible
  - see also CfgMgmtCamp 2024 talk Generating Ansible modules for REST APIs without AI by Evgeni Golov and Matthias Dellweg

- AWS has a similiar approach for generating providers from their SDK
- Can we also adopt something like this for Uyuni? 👉👈🥺
  - would **speed-up** onboarding new developers and new integrations **heavily**

Links

- Uyuni HTTP API: [click!]
- Uyuni Ansible Integration documentation: [click!]
- Uyuni Ansible Collection: [click!]
- Uyuni Terraform Provider: [click!]
- CfgMgmtCamp 2025 - Writing a Terraform/OpenTofu provider MVP for dummies: [click!]
- Uyuni OpenAPI RFC: [click!]

Thanks!