INTEGRATING UYUNI IN ANSIBLE

AND EVENT-DRIVEN ANSIBLE 🚀

// WHOAMI

> Christian Stankowic

> Consultant and Trainer @ SVA

> Linux, Virtualization, IaC, Automation

> moderates the FOCUS ON: Linux podcast 🇩🇪 🎙️

> https://cstan.io/en 📖

// AGENDA

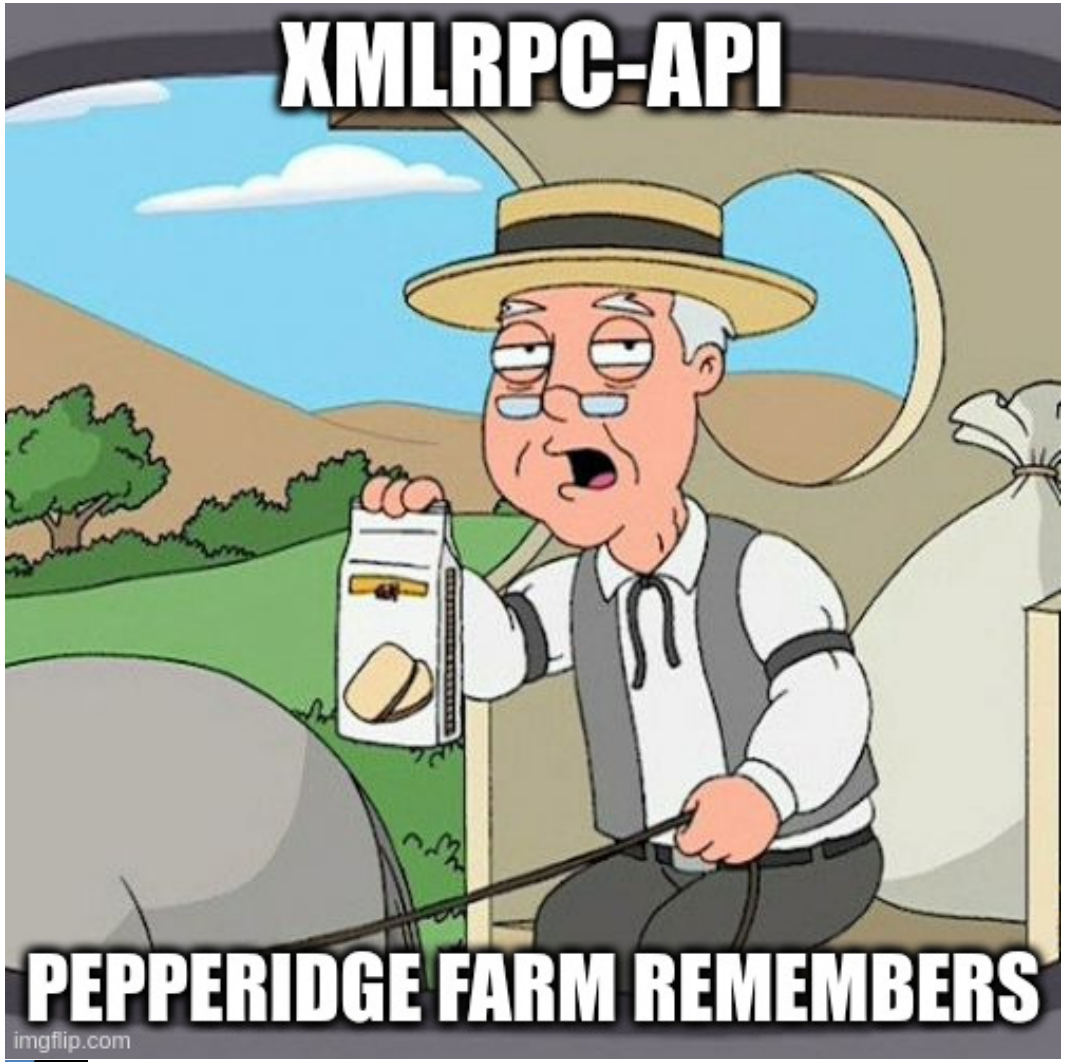
> Recap: Uyuni automation capabilities
> The Uyuni Ansible collection
> Example: AWX workflow
> Uyuni and Event-driven Ansible

// RECAP

UYUNI AUTOMATION CAPABILITIES

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

- **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

- **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
- `spacecmd`
  - Command-line interface to API
  - Many commands support wildcards, offers shell mode

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

SALT VS. ANSIBLE

SALT VS. ANSIBLE

	Ansible	SaltStack
Initial release	2012	2011
Agent	No	Yes and no
Networking	Push	Push
Language	YAML, Python	YAML, Python
Syntax level	easy	advanced
Community	Ansible Galaxy	SaltStack Formulas
	~20.000 members	50 members
	280 collections, 33.000 roles	350 formulas
High availability	Active/Active	Active/Active
Enterprise version	Ansible Automation Platform	vRealize Automation SaltStack Config

> 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*

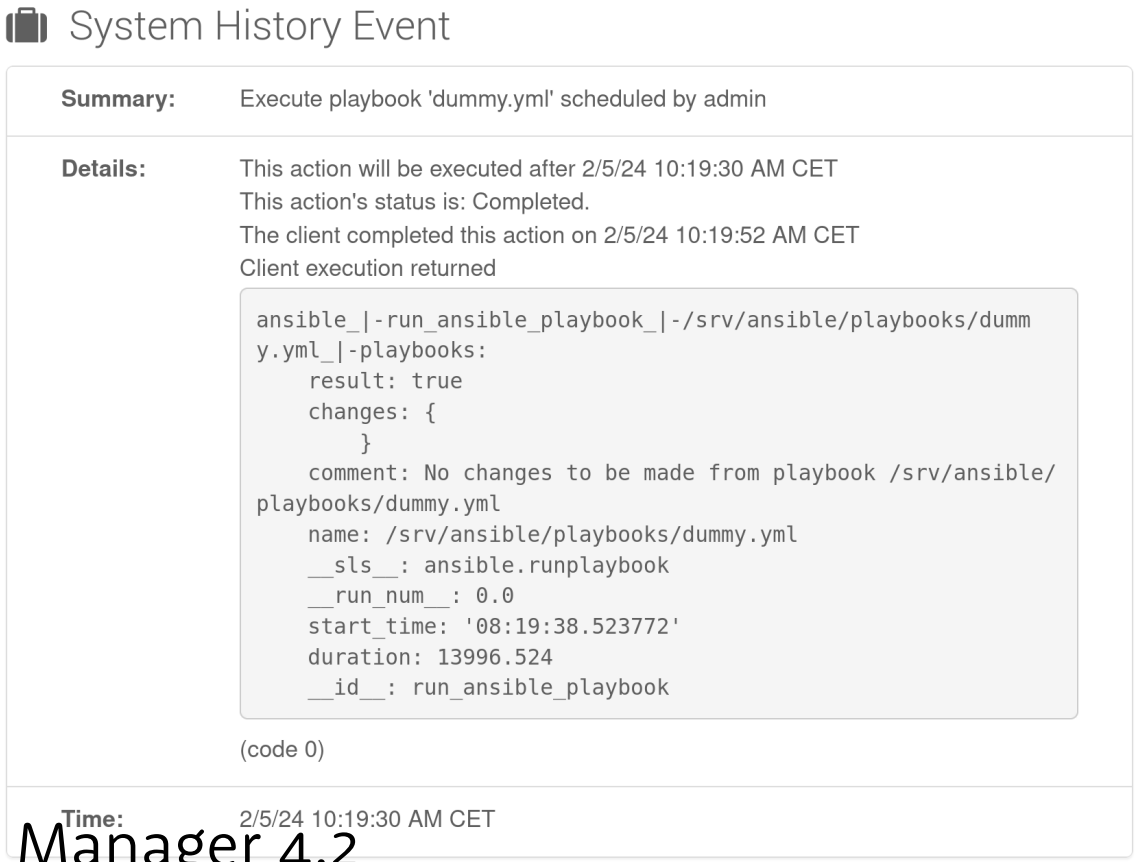
> using this, they can run Ansible playbooks - triggered by Salt

> supported with SUSE Manager 4.3

> limited usability and **outdated** (*2.9*)

> newer versions in the future

> usually Ansible infra already exist



* introduced with Uyuni 2021.06 and SUSE Manager 4.2

// THE UYUNI ANSIBLE COLLECTION

THE UYUNI ANSIBLE COLLECTION

- Collection of various **roles**
  - `storage` (*server preparation*)
  - `server` (*Uyuni/SUMA installation*)
  - `client` (*bootstrapping*)

THE UYUNI ANSIBLE COLLECTION

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

THE UYUNI ANSIBLE COLLECTION

This is a **work in progress** project.

Available on GitHub and Ansible Galaxy:

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

https://galaxy.ansible.com/ui/repo/publish

Contributions / feedback welcome!

INSTALLING SUSE MANAGER 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 Manager. Automatically registers the system with **SCC*** in order to retrieve packages.

* **S**USE **C**ustomer **C**enter

INSTALLING UYUNI AND 3RD PARTY REPOSITORIES

```
---
- name: Install Uyuni
  hosts: uyuni.giertz.loc
  become: true
  roles:
    - role: stdevel.uyuni.server
      uyuni_release: '2023.12'
      uyuni_channels:
        - name: opensuse_leap15_5
          arch: x86_64
      uyuni_enable_monitoring: true
      uyuni_install_monitoring_formulas: true
```

Installing a **specific** Uyuni version and configuration 3rd party repositories using `spacewalk-common-channels`. Also enables **monitoring** support.

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* – e.g. `bootstrap-almalinux9.sh`.

See also SUSE Manager documentation.

* below the `/pub/bootstrap` web directory

INSTALLING PATCHES

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

Parameters specify SUMA / 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
  become: true
  tasks:
    - name: Installing patches
      stdevel.uyuni.install_patches:
        uyuni_host: 192.168.1.1
        uyuni_user: admin
        uyuni_password: admin
        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: admin
  uyuni_password: admin
  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

```
---
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 needs to end with `.uyuni.yml`.

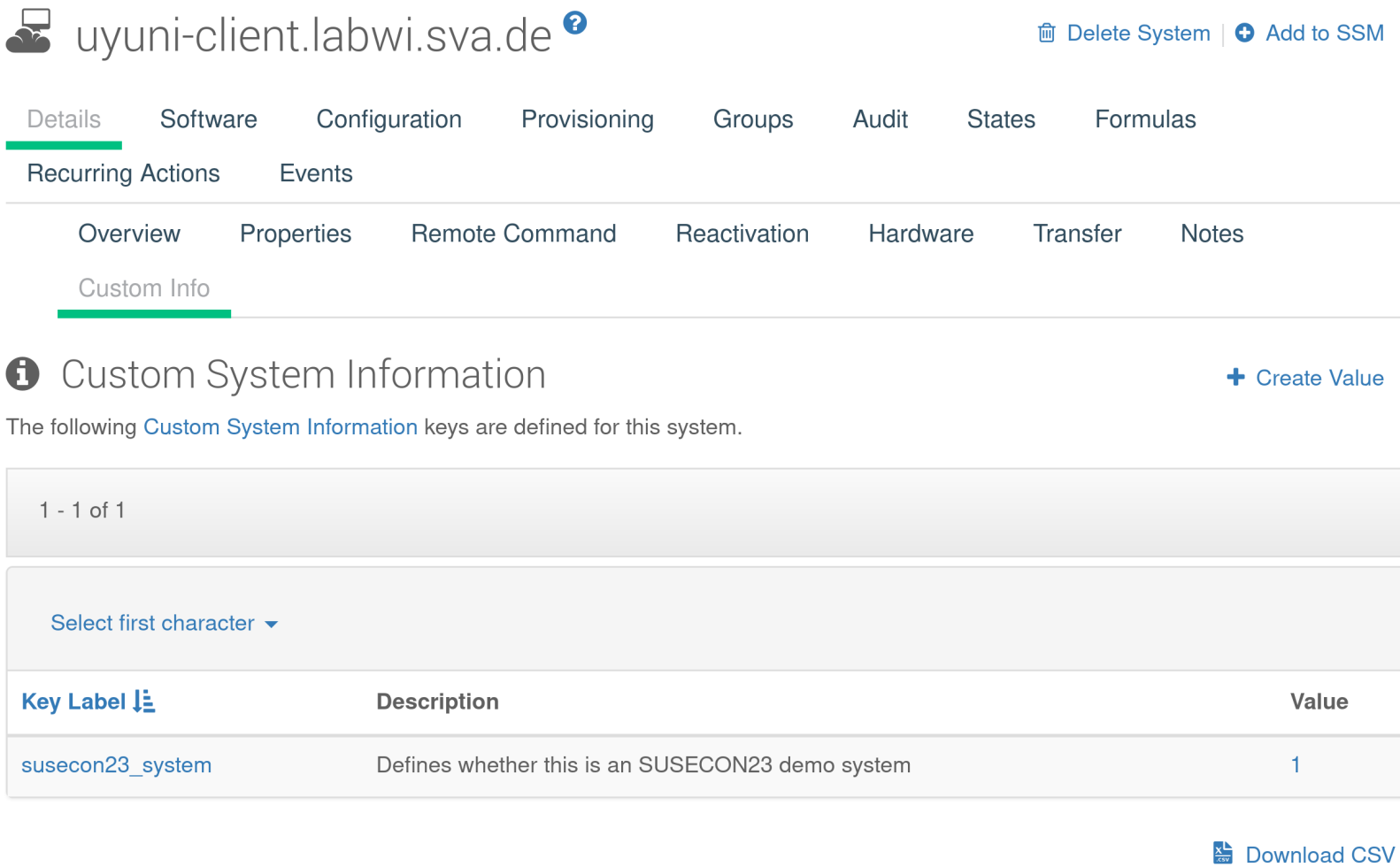
Configuration file specifies:

> 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"
            }
        }
    }
…
}
```

The Ansible inventory includes **hostgroups** and custom **system information** specified in the web interface.

Those information can be used in Ansible roles and playbooks.

// EXAMPLE: AWX WORKFLOW

VMWARE + CHECK_MK + UYUNI

OVERVIEW

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

`collections/requirements.yml`

```
---
collections:
  - name: stdevel.uyuni
    version: 0.1.5
  - name: checkmk.general
    version: 4.2.0
  - name: community.vmware
    version: 4.1.0
```

```
$ ansible-galaxy collection install -r requirements.yml
```

Always pin versions when using Ansible collections.

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 }}"
```

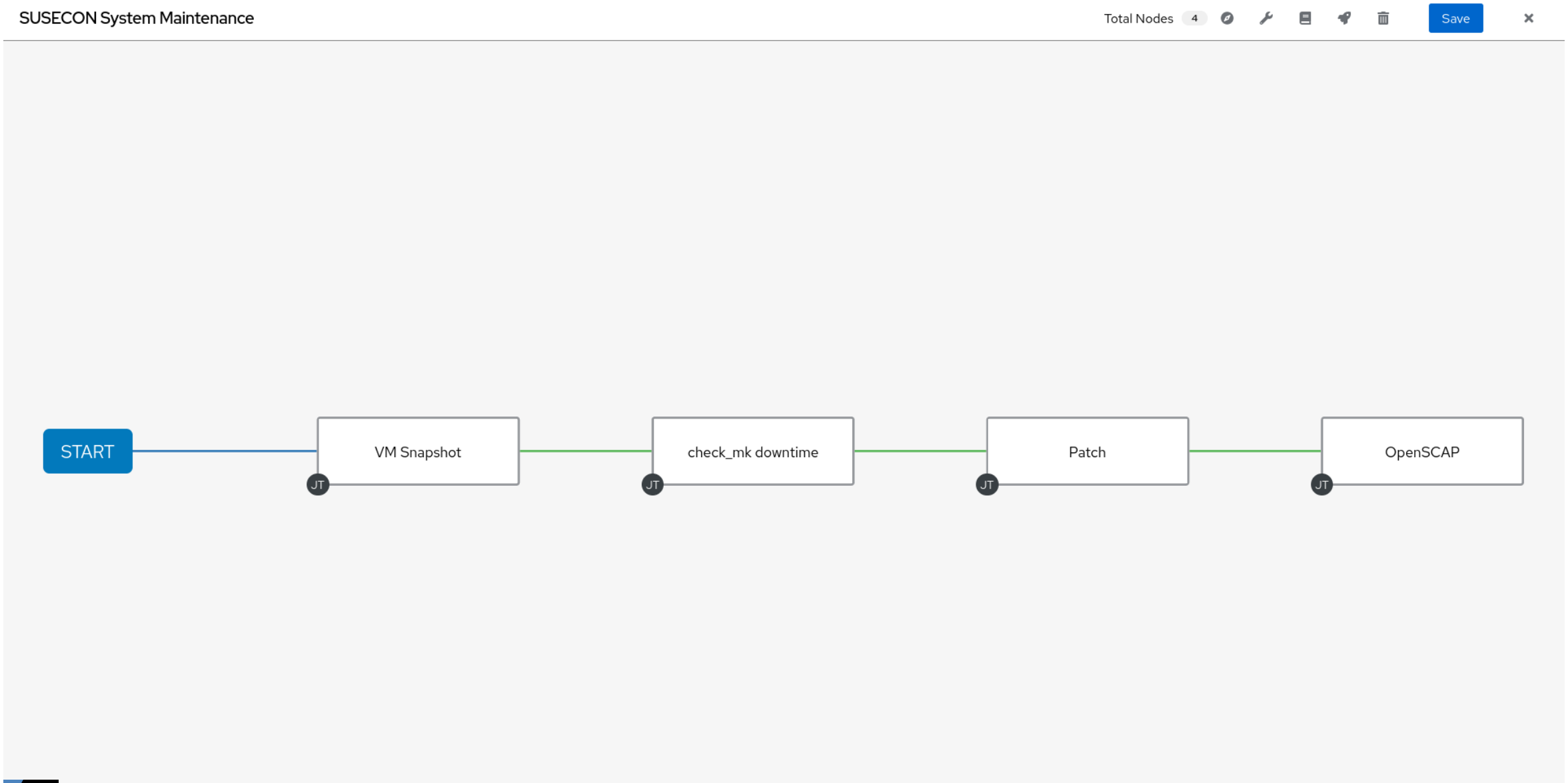
Installs available patches on all hosts, if not overwritten.

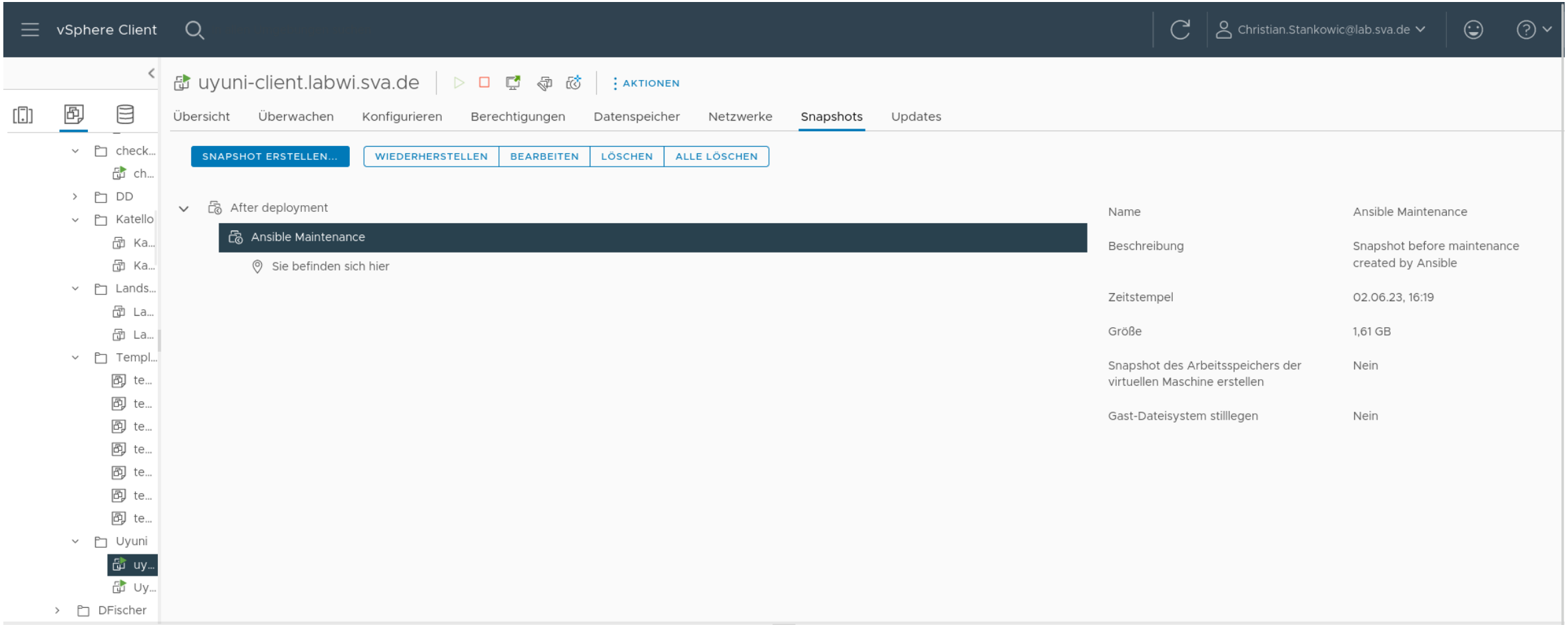
CODE

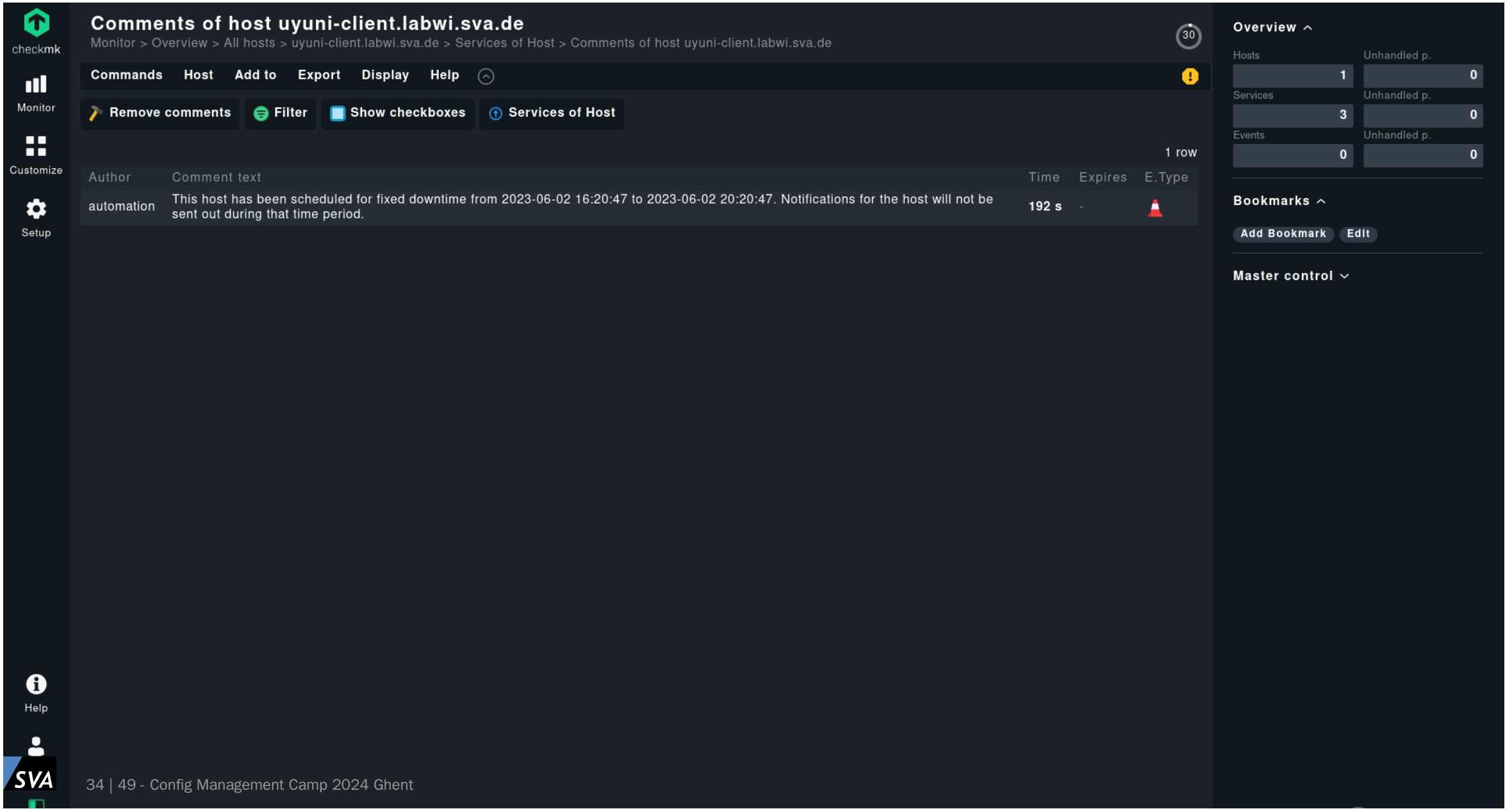
`run_openscap.yml`

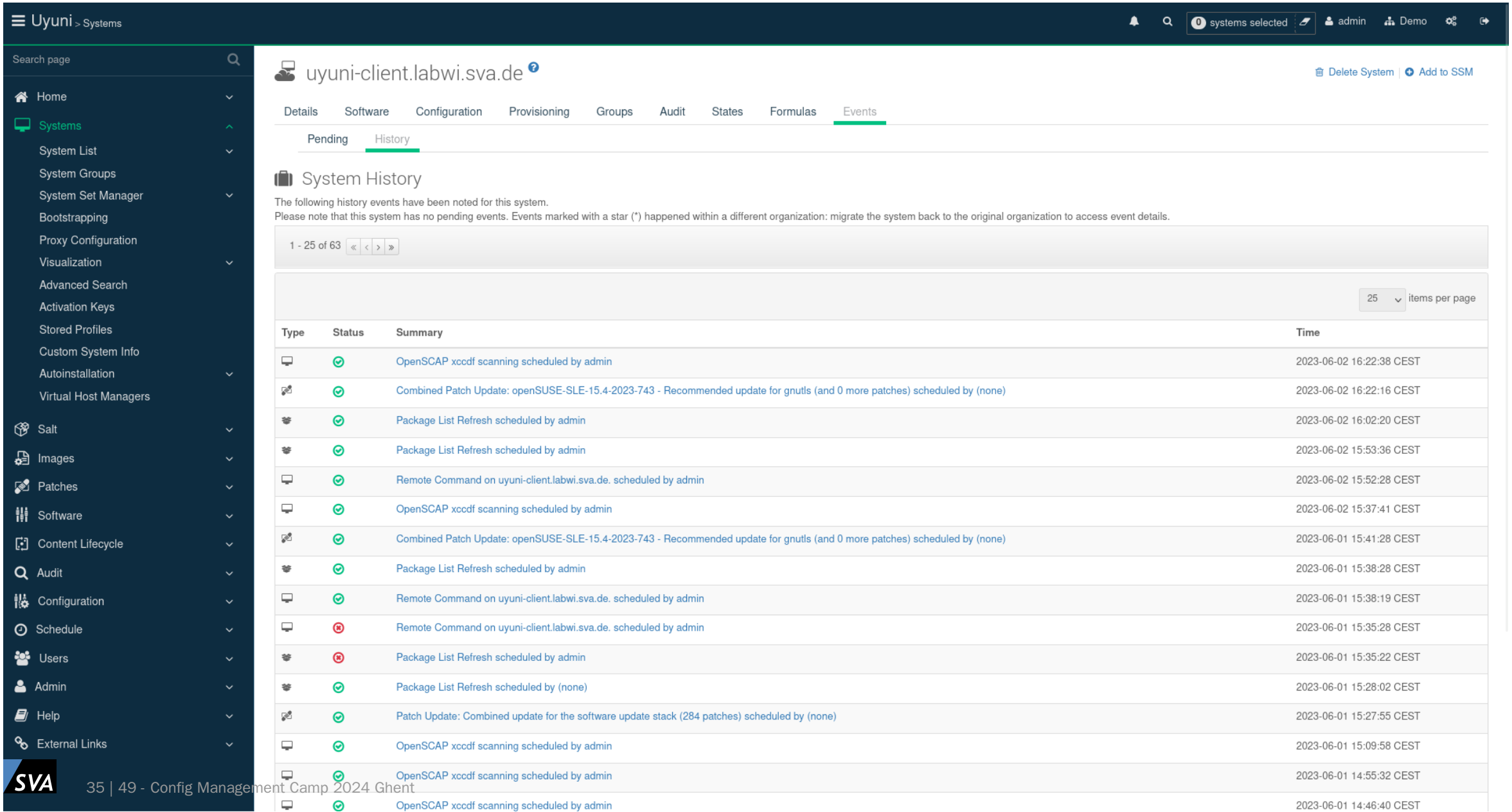
```
...
    - name: Schedule OpenSCAP run
      stdevel.uyuni.openscap_run:
        uyuni_host: "{{ uyuni_hostname }}"
        uyuni_user: "{{ uyuni_username }}"
        uyuni_password: "{{ uyuni_password }}"
        uyuni_verify_ssl: false
        name: "{{ item }}"
        document: /usr/share/openscap/scap-yast2sec-xccdf.xml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

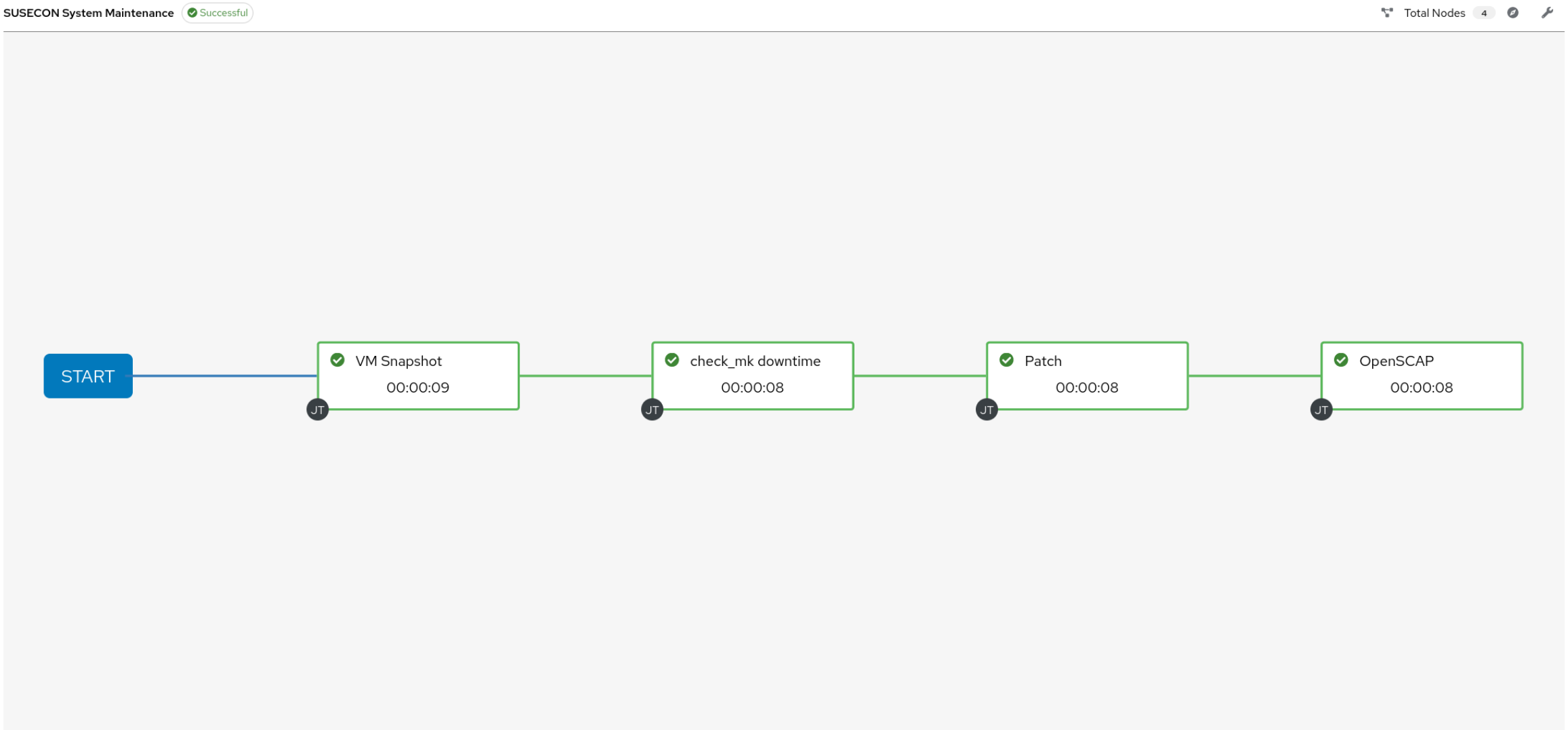
Runs a specific example catalog on all hosts.


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
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
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

// UYUNI AND

EVENT-DRIVEN ANSIBLE

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

EVENT-DRIVEN ANSIBLE

- **Rulebooks** define **events** to be **monitored***
  - can be filtered, **action** define countermeasures

EVENT-DRIVEN ANSIBLE

- **Rulebooks** define **events** to be **monitored***
  - can be filtered, **action** define countermeasures
- Event source **plugins** handle monitoring and filtering
  - currently **~22** plugins available
- Rulebooks are executed by `ansible-rulebook` command

EVENT-DRIVEN ANSIBLE

> **Rulebooks** define **events** to be **monitored***
  > can be filtered, **action** define countermeasures

> Event source **plugins** handle monitoring and filtering
  > currently **~22** plugins available

> Rulebooks are executed by `ansible-rulebook` command

> Part of Ansible Automation Platform

> executed in **Decision Environments**
  > literally a Podman container + content + `ansible-rulebook`

* yet another lovely YAML document

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

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
  gather_facts: false
  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 }}"
```

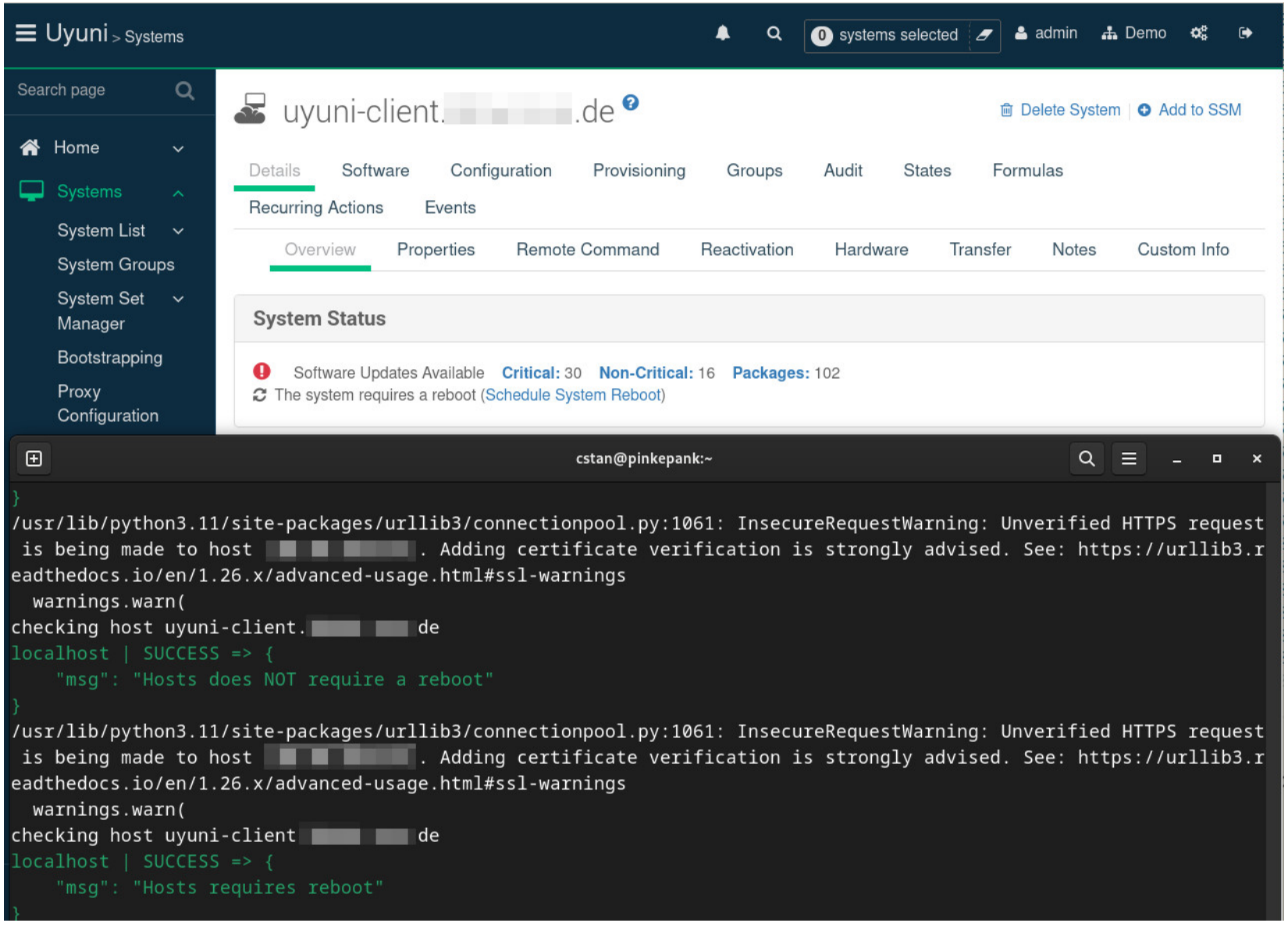
RUNNING

```
$ ansible-rulebook -i inventory.ini --rulebook trigger_reboots.yml
...
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.readthedocs.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.readthedocs.io/en/1.26.x/advanced-usage.html#ssl-warnings
warnings.warn(
checking host uyuni-client[redacted]de
localhost | SUCCESS => {
"msg": "Hosts requires reboot"
}

// ROADMAP

- **0.1.0** (*April 2023*)
  - initial release
- **0.1.1** (*April 2023*)
  - added `client` role
- **0.1.5** (*January 2024*)
  - added support for Uyuni 2023.09+
  - fixed several bugs
- **0.1.x** (*???*)
  - Event-driven Ansible support
  - activation key/bootstrap support
- **Some next version**
  - your ideas?

// LINKS

> Uyuni project page

> SUSE Manager documentation

> Salt Project website

> Uyuni Ansible Collection on GitHub

> Event-driven Ansible website

> Blog post about developing the Uyuni EDA support