SUSE | SUSECON 26

SUSE Multi-Linux Manager best practices from the field

TUTORIAL-1070

whoami

Christian Stankowic

Technical Leader Linux

SVA System Vertrieb Alexander GmbH

christian.stankowic@sva.de

Automation and Linux enthusiast

Spacewalk user since 2014

Agenda

1. Wrap-Up: Benefits of using SUSE Multi-Linux Manager
2. Installation and migration best practices
3. Orchestration and automation
4. Content and configuration

Wrap-Up

Benefits of using SUSE Multi-Linux Manager

Multi-Linux Manager in a nutshell

- Linux distribution-agnostic lifecycle management tool
  - Multi-tenancy
  - Patch management
  - Software distribution and caching
  - Infrastructure as Code using Salt
  - Provisioning new systems via PXE
  - Security audits using OpenSCAP
- 14 supported distributions
  - SLE, SLES4SAP, SL(E) Micro, openSUSE Leap, Liberty Linux, AlmaLinux, Amazon Linux, Debian, Open Enterprise Server, Oracle Linux, RHEL, Rocky Linux, Ubuntu
  - Additional distributions always in research and development
- Scales from small (single server) over big (servers and proxies) to huge environments (Hub and multiple servers and proxies)

SUSE | SUSECON 26

Installation and migration best practices

Installation best practices

- Double-check requirements
  - At least 4 CPU cores
  - 32 GB memory for production
  - 50 GB operating system disk (40 GB OS + 8-12 GB for swap)
- Use dedicated disk for database and packages
  - 50 GB for database
  - At least 150 GB for downloaded packages
- SUSE Linux Micro or SUSE Linux Enterprise Server?
  - Both SL Micro 6.1 and SLES 15 SP7 are supported for MLM 5.1
  - SL Micro subscription is part of the Lifecycle Management+ subscription
    - **Limited package set**, only ~5.000 out of ~30.000 packages (SLE)
  - Some customers aren't willing to go fully transactional yet

Installation best practices

- Add your SSL certificates before installing MLM
  - Copy root and intermediate CA certificates to `/etc/pki/trust/anchors`
  - They will be copied to the containers during installation
- Create certificates before the installation
  - Don't forget **SANs** `db` and `reportdb` if you want you use one certificate for both Uyuni and the database container
- Set a customized organization name
  - Unless your company name is really Default Organization 👀
- Use YAML answer file when installing multiple MLM servers
  - Reduces errors
  - Reproducible installations

Using an answer file for installation

```
# Organization settings
organization: MyCorp
emailFrom: mlm@mycorp.lan

# SCC credentials
scc:
  user: sccUsername
  password: sccPassword
```

```
# Administrator account
admin:
  password: trustno1
  login: fmulder
  firstName: Fox
  lastName: Mulder
  email: fmulder@pinkepa.nk


# mgradm -c mlm.yml install
podman $(hostname -f)
```

Migration best practices

- Update to latest supported version of SUSE Manager first (4.3.16.2 or 5.0.6)
- **Clean-up** before migration
- Remove unneeded and unsupported channels (e.g. SLE `11`, SLE `12` before SP5)
  - `spacewalk-remove-channel -a <name>`
  - `spacecmd remove_orphans`
  - `spacewalk-data-fsck -vrR`
- Archive old tasks (reduces database size)
  - `spacecmd schedule_archivecompleted`
  - `spacecmd schedule_deletearchived`
- Remove layover files from previous PostgreSQL database versions
  - E.g. `/var/lib/pgsql/data/data-pg14`
  - Can save a lot of space

Migration best practices

- Run migration **preparation** first
  - `mgradm migrate podman --prepare old.mycorp.lan`
  - Does an initial sync but won't stop source system (can take a long time)
- Double-check connectivity to old system
  - In some cases `pg_hba.conf` might be misconfigured
  - `podman run -ti registry.suse.com/suse/manager/5.0/x86_64/server-migration-14-16 /bin/bash`
  - `psql -U pythia_susemanager -d reportdb --host old.mycorp.lan`
  - `psql -u uyuni -d uyuni --host old.mycorp.lan`
- Create a **VM snapshot** of both old and new system
  - Easier to fallback to old status

/var/lib/pgsql/data/pg_hba.conf for migration

```
local reportdb postgres trust
local reportdb all trust
host reportdb all ::/0 trust
host reportdb all 0.0.0.0/0 trust
host susemanager all 0.0.0.0/0 trust
host reportdb all ::1/128 trust
host reportdb all 127.0.0.1/32 trust
local susemanager postgres trust
local susemanager spacewalk trust
host susemanager spacewalk 127.0.0.1/32 trust
host susemanager spacewalk ::1/128 trust
local replication all trust
host replication all ::1/128 trust
host replication all 127.0.0.1/32 trust
local all all trust
host all all ::1/128 trust
host all all 127.0.0.1/32 trust
local replication postgres trust
host susemanager susemanager all scram-sha-256
local replication postgres peer
local reportdb pythia_susemanager trust
host reportdb pythia_susemanager 127.0.0.1/32 scram-sha-256
```

SUSE | SUSECON 26

Orchestration and automation

Automation possibilities

- MLM offers multiple utilities for automation
  - XMLRPC API
  - JSON over HTTP API
  - `spacecmd` utility
- Beside Salt also **Ansible Integration** is available for configuring managed hosts
  - Ansible is executed on Salt-managed Ansible control-node
  - Playbooks and inventories managed in the web interface
- Unofficial community tools
  - Contributed Uyuni scripts (uyuni-project/contrib)
  - Ansible collection (stdevel.uyuni)
  - Terraform / OpenTofu provider (svalabs.uyuni)

spacecmd

- Command-line interface for nearly all API calls
  - Content Lifecycle Projects, Action Chains, Organization credentials and VHM missing
- Can be used for single commands but also offers an **interactive shell**
  - Command prefixes are easy to remember (e.g. `system_` or `errata_`)
- Pre-installed in MLM container
  - Also part of the Client Tools, can be installed on other hosts as well
- Configuration file (`~/.spacecmd/config`) can store connection details

```
[spacecmd]
server=mlm01.stankowic.loc
username=admin
nossl=1
```

spacecmd

```
# spacecmd -- system_listerrata st-devel01.stankowic.loc
Security Errata
---------------
openSUSE-SLE-15.6-2026-829  moderate: Security update for gnutls
stable     2026-03-05
…

# spacecmd -y -- system_applyerrata st-devel01.stankowic.loc
openSUSE-SLE-15.6-2026-829 -s now
Errata              Systems
--------------      -------
openSUSE-SLE-15.6-2026-829          1

Start Time: 20260307T11:14:15
INFO: Scheduled 1 system(s) for openSUSE-SLE-15.6-2026-829
```

spacecmd

```
# spacecmd
Spacewalk Username: admin
Spacewalk Password:

spacecmd {SSM:0}> system_list
st-client01.stankowic.loc : 1000010000
st-client02.stankowic.loc : 1000010001
st-devel01.stankowic.loc : 1000010002

spacecmd {SSM:0}> system_applyerrata st-devel01.stankowic.loc
openSUSE-SLE-15.6-2026-228
Start Time [now]:
Errata              Systems
--------------      -------
openSUSE-SLE-15.6-2026-228          1
Start Time: 20260306T23:49:20
Apply these patches [y/N]: y
```

Uyuni Tools

- Collection of various useful scripts that are **not yet officially** part of Uyuni
- Available on GitHub (uyuni-project/contrib)
- Configuration file (`configsm.yaml`) defines connection, mail and patch management information
- Some of the useful scripts
  - `create_software_project.py` - Create and update Content Lifecycle Projects
  - `group_system_update.py` - Update and reboot all systems of a host group
  - `migrate_to_new_client_tools.py` - Update client tools after MLM upgrade
  - `system_update.py` - Fully update systems (patches, configuration, reboot)

Fully update, configure and reboot a system

```
$ ./system_update.py --server st-devel01.stankowic.loc --applyconfig
07-03-2026 15:20:28 | st-devel01.stankowic.loc | INFO | Hostname : st-devel01.stankowic.loc
07-03-2026 15:20:28 | st-devel01.stankowic.loc | INFO | Systemid : 1000010002
07-03-2026 15:20:28 | st-devel01.stankowic.loc | INFO | Performing highstate
07-03-2026 15:21:29 | st-devel01.stankowic.loc | INFO | Server st-devel01.stankowic.loc will be
upgraded with latest available patches
07-03-2026 15:21:30 | st-devel01.stankowic.loc | INFO | Performing zypper update
07-03-2026 15:22:31 | st-devel01.stankowic.loc | INFO | zypper update completed successful.
07-03-2026 15:22:31 | st-devel01.stankowic.loc | INFO | Running Package refresh
07-03-2026 15:23:01 | st-devel01.stankowic.loc | INFO | Package refresh completed successful.
07-03-2026 15:23:01 | st-devel01.stankowic.loc | INFO | Performing Errata update
07-03-2026 15:25:32 | st-devel01.stankowic.loc | INFO | Errata update completed successful.
07-03-2026 15:25:32 | st-devel01.stankowic.loc | INFO | Running Package refresh
07-03-2026 15:26:03 | st-devel01.stankowic.loc | INFO | Package refresh completed successful.
07-03-2026 15:26:03 | st-devel01.stankowic.loc | INFO | Rebooting server
07-03-2026 15:29:34 | st-devel01.stankowic.loc | INFO | Reboot completed successful.
07-03-2026 15:30:05 | st-devel01.stankowic.loc | INFO | Performing highstate
07-03-2026 15:31:36 | st-devel01.stankowic.loc | INFO | Apply highstate completed successful.
07-03-2026 15:31:40 | st-devel01.stankowic.loc | INFO | Finished
```

Ansible collection

- Unofficial collection that allows MLM being **controlled by** Ansible
  - Often customers already have Ansible infrastructure for automation
  - Collection currently uses the XMLRPC API
- Can be used to trigger system maintenance
  - Modules for installing patches, running OpenSCAP, applying Highstate, rebooting systems
- Supports using MLM as **Dynamic Inventory**
  - No need to manually assign hosts to your inventory
- Installs MLM servers and proxys
  - Reproducible deployments, e.g. for on-demand deployments
  - Can be part of CI/CD pipeline and GitOps setups
- Allows bootstrapping new clients
- Available on Ansible Galaxy and Codeberg

Install and configure MLM using Ansible

```
- name: Install SUMA 5.0 on SLE 15.6 and add AlmaLinux 9 content
  hosts: suma-servers
  roles:
    - role: stdevel.uyuni.server
      server_suma_release: 5.0
      server_scc_reg_code_os: DERP1337LULZ
      server_scc_reg_code_mlm: RFL0815CPTR
      server_scc_mail: dlynch@twnp.ks
      server_channels:
        - name: almalinux9
          arch: x86_64
        - name: almalinux9-appstream
          arch: x86_64
        - name: almalinux9-uyuni-client
          arch: x86_64
```

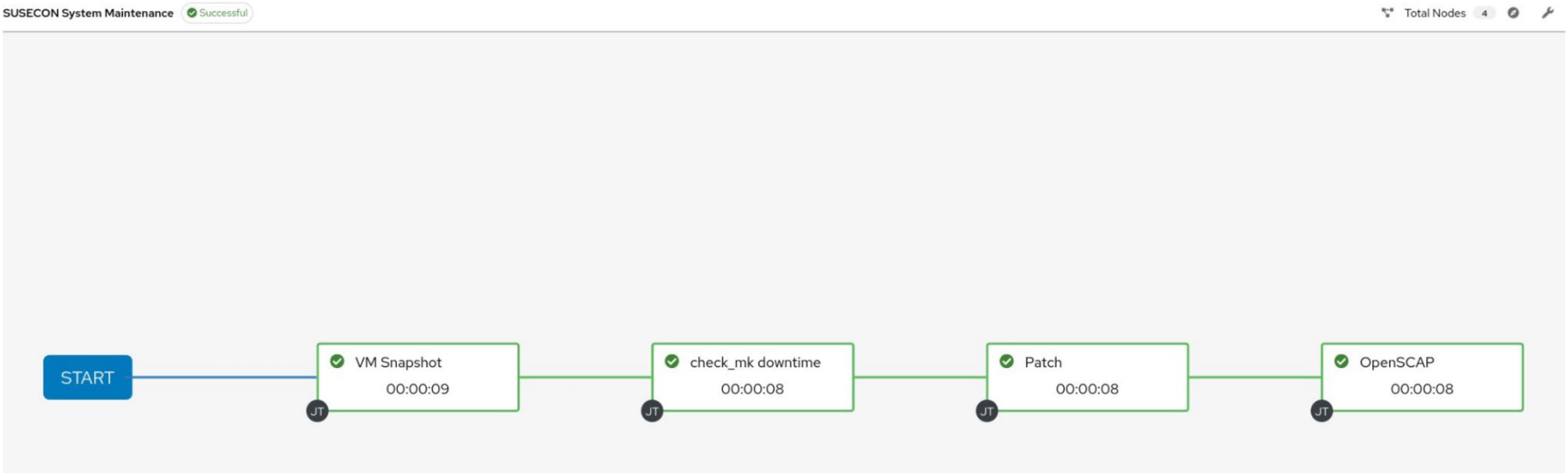
Ansible collection

Install patches and schedule OpenSCAP run

```
- name: Installing patches
  stdevel.uyuni.install_patches:
    uyuni_host: 192.168.1.1
    uyuni_user: admin
    uyuni_password: trustno1
    uyuni_verify_ssl: false
    name: web.evilcorp.lan
    exclude_patches:
      - openSUSE-SLE-15.6-1337
```

```
- name: Check compliance
  stdevel.uyuni.openscap_run:
    uyuni_host: 192.168.1.1
    uyuni_user: admin
    uyuni_password: trustno1
    uyuni_verify_ssl: false
    name: web.evilcorp.lan
    document: /opt/my-xccdf.xml
    arguments: --profile Default
```

AWX-based workflow for full system maintenance



SUSE | SUSECON 26

Content and configuration

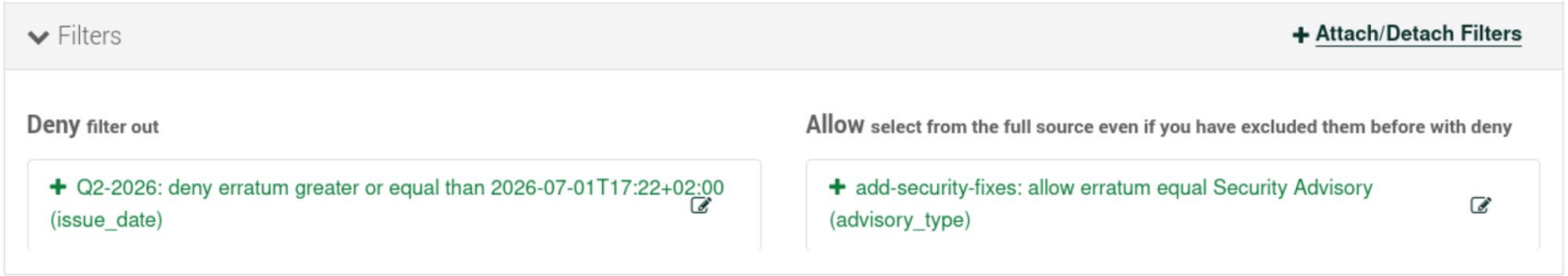
Third-party repositories and client tools

- Use `spacewalk-common-channels` after installation to mirror required software repositories
- Use official Multi-Linux Manager Client Tools **from SCC**
  - **Don't mix** MLM and Uyuni client tools

```
# spacewalk-common-channels -a x86_64 -c opensuse_leap15_6
# spacewalk-common-channels -a x86_64 -c
opensuse_leap15_6-sle-updates
# spacewalk-common-channels -a x86_64 -c
opensuse_leap15_6-backports-updates
```

Content Lifecycle Projects

- Create dedicated projects per OS release
  - E.g. SLE 15 SP7 and SLE 16
  - Improves clarity
  - Don’t assign systems to upstream channels
- Create additional projects for applications if patch release working differs
- Use filters sparingly
  - E.g. apply patches on a quarterly basis but always include security fixes



Hostgroups

- Hosts can be ordered in **multiple host groups** in order to improve clarity
- System Set Manager (SSM) can benefit from using multiple groups per
  - Lifecycle (e.g. development, production)
  - Application/Function (e.g. all web servers)
  - Network (e.g. all DMZ hosts)
  - OS release (e.g. SLES 15 SP7 hosts)
- Assists patching process in combination with **Union/Intersection** SSM modes
  - Patch all non-productive web servers
  - Patch all productive database servers
  - ...

Activation keys and bootstrap scripts

- Create activation keys per
  - OS release (e.g. `ak-sles-15-sp7`)
  - Environment (e.g. `ak-sles-15-sp7-prod`)
  - Application (if needed, e.g. `ak-sles-15-sp7-prod-sap`)
- Rotate bootstrap scripts after Multi-Linux Manager upgrade
  - Might fix bugs and add new features, see version string in the script

```
# mgrctl term
# cd /srv/www/htdocs/pub/bootstrap
# mgr-bootstrap --activation-key=1-sle-15-sp7 --script=bs-sle-15-sp7.sh
…
```

Salt states

- Avoid conventional config states
  - Old functionality, might cease to exist at some time
  - Most setups use Salt anyway
- Do not assign states to individual machines
  - Easy to lose track
- Assign global states on organization level
  - E.g. SSH and LDAP configuration
- Assign application-specific states on host group level
  - E.g. firewall rules and application configuration
- KISS principle, try not to over-engineer things

SUSE | SUSECON 26

Thank you