SUSE Manager at Scale
A Case Study

Cleber Paiva de Souza <cleber@ssys.com.br>
Rodolpho Pivetta <rodolpho.pivetta@ssys.com.br>
SSYS Sistemas
About SSYS

• Founded in 2014.
• Experts in Linux Enterprise in Brazil.
• Experienced professionals certified in all SUSE portfolio (SEA, SCE, SCA e SCI).
• Projects with SUSE Manager, SUSE OpenStack Cloud, SUSE Storage, SUSE HA with SAP Application and HANA etc.
• A software development company.
About Via Varejo

• Merge Casas Bahia and Ponto Frio.
• More than 1000 stores over 20 states in Brazil.
• Almost 20,000 sales terminals.
• Biggest retail of electronics and furniture in Brazil with around 50,000 employees.
Challenges

- Around 20,000 devices spread over 1,000 stores
- Moving from an old legacy PXE-boot model
- Changing tires while driving 😊
- Total replacement should happen in at most 5 months
- Support for further changes without new installations
- Version control of infrastructure
- Offline operation and maintenance tasks
- Support web interface for 1st, 2nd and 3rd levels

Source: https://lifepalette.com/changing-a-tire-while-still-driving/
Proposed architecture
Kiwi

- SUSE opensource imaging tool
- SLED 12 SP3 based image for terminals
- Bootstrapping from USB stick
- Salt code in image for offline operation and maintenance
- OS tuning and hardening
- Auto-deployment of customer software
- Network and security rules built-in
- Simplicity in mind

Kiwi process

- OS settings
- Hardening
- Salt Automation with SUMA bootstrapping
- Internal softwares

config.xml
config.sh
images.sh

OEM

BUILD
SUSE Proxy

- Salt-broker to forward connections back and forth to salt-master
- Local caching (Squid) for repositories
- Around 20,000 minion connected to SUSE Proxy Server via F5 load balancer
- Easily scalable by adding more SUSE Manager Proxies
- Tunings in OS to accommodate high number of connections
- Limited to a maximum of 15 SUSE Manager Proxies due to restrictions in Subject Alternative Name (SAN) field in the X.509 certificate
Salt

Components:
• **Salt Master**: Server side
• **Salt Minion**: Client side

Resources:
• **State**: On Salt you define the desired state of resources
• **Grains**: Derive public information about the underlying system (OS version, domain, IP Addresses, kernel, CPU, RAM, custom etc)
• **Pillars**: Derive private information. Useful for sensitive data specific to a particular minion.
• **Mine**: Collect arbitrary information from minion and store it in Salt Master.
• **Scheduler**: Allow scheduling of action in Salt.

**SUSE Manager Server and proxies' tunings**

- **First:** SUSE recommendations available in SUSE Manager docs
- **Tomcat tuning**
- **Custom sysctl parameters**
- **/var/cache/salt (job cache) out of BTRFS**

```bash
# Reuse connections
net.ipv4.tcp_tw_recycle = 1
net.ipv4.tcp_tw_reuse = 1

# Controls of network
net.ipv4.tcp_syncookies = 1
net.ipv4.tcp_max_syn_backlog = 819200
net.ipv4.tcp_synack_retries = 3
net.ipv4.tcp_fin_timeout = 10
net.ipv4.tcp_keepalive_time = 60
net.ipv4.tcp_max_tw_buckets = 65536
fs.file-max = 6553600
net.core.rmem_default = 16777216
net.core.wmem_default = 16777216
net.core.rmem_max = 134217728
net.core.wmem_max = 134217728
net.ipv4.tcp_rmem = 4096 87380 16777216
net.ipv4.tcp_wmem = 4096 65536 16777216
net.core.somaxconn = 40000
net.core.netdev_max_backlog = 300000
net.ipv4.tcp_mtu_probing = 0
net.core.optmem_max = 327680

# Timeout
net.netfilter.nf_conntrack_tcp_timeout_close = 5
net.netfilter.nf_conntrack_tcp_timeout_close_wait = 15
net.netfilter.nf_conntrack_tcp_timeout_fin_wait = 15
net.netfilter.nf_conntrack_tcp_timeout_last_ack = 10
net.netfilter.nf_conntrack_tcp_timeout_max_retrans = 60
net.netfilter.nf_conntrack_tcp_timeout_syn_recv = 15
net.netfilter.nf_conntrack_tcp_timeout_syn_sent = 15
net.netfilter.nf_conntrack_tcp_timeout_time_wait = 15
net.netfilter.nf_conntrack_tcp_timeout_unacknowledged = 60
net.netfilter.nf_conntrack_udp_timeout = 15
net.netfilter.nf_conntrack_udp_timeout_stream = 60

# double amount of allowed conntrack
net.netfilter.nf_conntrack_max = 2621440
net.netfilter.nf_conntrack_tcp_timeout_established = 1800

# Disable netfilter on bridges.
net.bridge.bridge-nf-call-ip6tables = 0
net.bridge.bridge-nf-call-iptables = 0
net.bridge.bridge-nf-call-arptables = 0
```
**SUSE Manager customizations (1/3)**

- Avoid direct changes in susemanager.conf or trace all changes for returning them back after updates/upgrades.
  - **Suggestion**: creation of `/etc/salt/master.d/99-file.conf` file structure to organize merge of configurations. We have used `/etc/salt/master.d/zcustoms.conf`.
- Some configurations in `/etc/salt/master.d/*.conf` are merged but some use latest occurrence. Prefixing files with numbers should help.
- Salt-api binds to localhost by default. We changed that for our application.
  - Tuning `thread_pool` and `socket_queue_size` in `rest_cherrypie` to support more Salt API connections.
  - Increase limits (TasksMax=infinity and LimitNOFILE=61440 in systemd for salt-api.service)
- Updates/Upgades return configuration to defaults.
SUSE Manager customizations (2/3)

- file_roots and pillar_roots definitions are merged among files but reactor definition don’t.
- Supporting changes during rollout without disruption of services
  - States saved in bitbucket and synchronized in salt master.
  - Changes in top.sls using id and network class.
  - Merging multiples top.sls (top_file_merging_strategy: merge_all)
- Supporting new kind of devices in future, like POS, Kiosks etc
  - Roles in salt using pillars.
- Rollback support in case of failure in updates or changes (BTRFS snapshots)
  - Salt state to create initial snapshot (never deleted) and 2 rotating weekly snapshots.
SUSE Manager customizations (3/3)

- Increase sock_pool_size
- Enabling:
  - con_cache
  - presence_events
  - ping_on_rotate
- Tuning ZeroMQ
  - pub_hwm
  - zmq_backlog
- key_cache: 'sched'
  - Increase Salt Master targeting response time

Tuning reactor:
- reactor_refresh_interval, reactor_worker_threads and reactor_worker_hwm
SUSE Manager issues

- **Update from salt 2016.11.x to 2018.3.0**
  - Some grains changed their format from string to integer: osmajorrelease.
  - **Some state options do not work**: attrs in file.managed. We had to change to cmd.run with chattr.
  - archive.extracted do not work well with archives with lots of files during permissions and owner corrections.
  - **salt-master[?????]: TypeError: unicode not allowed, use setsockopt_string**
    - `/usr/lib/python2.7/site-packages/salt/utils/master.py`
    - From: cupd_in.setsockopt(zmq.SUBSCRIBE, ")
    - To: cupd_in.setsockopt(zmq.SUBSCRIBE, b")
  - [https://github.com/saltstack/salt/pull/46878/files](https://github.com/saltstack/salt/pull/46878/files)
  - top.sls does not work with many items in matching list (>200). Used jinja instead.
  - Minimum 16 dedicated cores or prefer physical server. 32 GB RAM is enough for this size.
Minion customizations

• Standardize minion id (minion_id_lowercase, domain name added etc)
• Increase timeout (acceptance_wait_time, master_alive_interval, recon_default etc)
• Add randomization all around (random_startup_delay, random_reauth_delay, recon_randomize)
• Ran state could get confused, so set master_tops_first to True so SUSE Manager states come first. Important to solve repositories dependences before installing new packages.
• Local/offline mode and detection of infrastructure to change to online mode.
Custom States for business

• More than 4400 line of states
  • Gnome, services, network, printer, applications, security, users and corrections/remediations.

• Custom grains in python
  • geolocalization, store information, state, region, state version etc
  • used for queries, supporting tools and desktop wallpaper automation

• Pillars for screen resolution and roles

• Salt Highstate applied at regular intervals
  • Guarantee everything is working properly (softwares, configurations, security etc)
Supporting tools

- Supporting team needs a simple tool to provide support
- SUSE Manager is too powerful for them
- Abstract salt states in buttons and action
- We have developed Wise to help customer on daily task
# Supporting tools

![Wise Interface](image)

## Projects

### Project Overview

<table>
<thead>
<tr>
<th>key</th>
<th>saltstack</th>
<th>susemanager</th>
<th>ipv4/ipv6</th>
<th>SO</th>
<th>Arch</th>
<th>Hardware</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.192.7.49</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td>SUSE 12.3 x86_64</td>
<td>physical</td>
<td>3.93 GB</td>
<td></td>
</tr>
<tr>
<td>10.192.7.49</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td>SUSE 12.3 x86_64</td>
<td>physical</td>
<td>3.93 GB</td>
<td></td>
</tr>
<tr>
<td>10.192.7.49</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td>SLE 12.3</td>
<td>--</td>
<td>7.88 GB</td>
<td></td>
</tr>
<tr>
<td>10.192.7.49</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td>SLE 12.3</td>
<td>--</td>
<td>7.88 GB</td>
<td></td>
</tr>
<tr>
<td>10.192.7.49</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td>SLE 12.3</td>
<td>--</td>
<td>7.88 GB</td>
<td></td>
</tr>
<tr>
<td>10.192.7.49</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td>SUSE 12.3 x86_64</td>
<td>physical</td>
<td>7.88 GB</td>
<td></td>
</tr>
<tr>
<td>10.192.7.49</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td>SLE 12.3</td>
<td>--</td>
<td>3.93 GB</td>
<td></td>
</tr>
<tr>
<td>10.192.7.49</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td>SLE 12.3</td>
<td>--</td>
<td>3.93 GB</td>
<td></td>
</tr>
<tr>
<td>10.192.7.49</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td>SUSE 12.3 x86_64</td>
<td>physical</td>
<td>7.87 GB</td>
<td></td>
</tr>
<tr>
<td>10.192.7.49</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td>SUSE 12.3 x86_64</td>
<td>physical</td>
<td>7.69 GB</td>
<td></td>
</tr>
<tr>
<td>10.192.7.49</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td>SLE 12.3</td>
<td>--</td>
<td>7.88 GB</td>
<td></td>
</tr>
</tbody>
</table>

### Sync Options
- Sync all
- Reboot
- Shutdown
- Halt

---

*Note: This function synchronizes custom modules, states, beacons, grains, returners, output modules, renderers, and saltenv.*

---

*Image credit: Wise Interface [Source](image)
Applications on top of SUSE Manager
Thank you.
General Disclaimer
This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. SUSE makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. The development, release, and timing of features or functionality described for SUSE products remains at the sole discretion of SUSE. Further, SUSE reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All SUSE marks referenced in this presentation are trademarks or registered trademarks of SUSE LLC. in the United States and other countries. All third-party trademarks are the property of their respective owners.