Running Ansible within Salt
How to smoothly migrate away from Ansible to Salt

Pablo Suárez Hernández
SUSE Senior Software Engineer
psuarezhernandez@suse.com

Don Vosburg
SUSE Technology Specialist
dvosburg@suse.com
People are using Ansible out there!
People are running Ansible out there!

- Customer’s IT infrastructure already defined with Ansible
- Efforts already invested. Security audit.
- Ansible modules and playbooks all over the places.
... and they always face the same problems!

- Real-time monitoring
- Event-driven Orchestration
- Scalability
Salt enables you! Let’s put Salt in your IT.

• Real-time monitoring → beacons!

• Event-driven Orchestration → reactors!

• Scalability → zeromq, syndics!
But wait! We have a problem!
But wait! We have a problem!

- Customer IT infrastructure already defined with Ansible
- Efforts already invested. Security audit.
- Ansible modules and playbooks all over the place.

No way of a smooth, step-by-step, planned migration. Only the hard way. UNTIL NOW!
Don’t panic!
What would happen if ... 

we could just simply consider the whole Ansible as a subset of the functionality that Salt provides?
Ansible Gate: Cooking Ansible with a bit of Salt

- New module in Salt 2019.2.0 “Fluorine” release
- An innovation coming from SUSE
- Execute Ansible modules from Salt
- Run your playbooks!
Ansible Gate: Cooking Ansible with a bit of Salt

The Salt “Fluorine” release package is available in:

- SLE15 and SLE15SP1. Base system package
- OpenSUSE Leap 15/15.1, 42.3 and Tumbleweed
- “systemsmanagement:saltstack:products:next/salt” at OBS [1]

How does the Ansible Gate work?
How does the Ansible Gate work?

- Manage your Ansible from Salt
- Execute Salt commands on your Ansible managed systems (salt-master & salt-ssh)
How does the Ansible Gate work?

# Some examples:
# Ansible needs to be installed on the running system.

$ salt-call --local ansible.list

$ salt-call --local ansible.help system.ping

$ salt-call --local ansible.system.ping data="Hello from Salt"
How does the Ansible Gate work?

# Some examples:
# Ansible needs to be installed on “ansiblenode” minion.
# Playbooks are already stored on the “ansiblenode” minion.

$ salt “ansiblenode” ansible.system.ping

$ salt “ansiblenode” ansible.packaging.os.zypper name=“iotop”
  state=“installed”

$ salt “ansiblenode” ansible.playbook
  playbook=/srv/playbooks/install_nginx.yml
- Manage your Ansible from Salt
- Enables real-time monitoring and event-driven orchestration.
- Add the whole minion functionality to the Ansible managed systems.
# Example of running Salt commands
# on Ansible managed systems.

```bash
$ salt-ssh --roster=ansible --roster-file=/etc/ansible/hosts -N all test.ping
```

```bash
$ salt-ssh --roster=ansible --roster-file=/etc/ansible/hosts -N webservers service.stop apache2
```
How to *smoothly* migrate from Ansible to Salt
The common Ansible scenario

- Ansible modules
- Playbooks
Step 0: Adding just few grains of Salt
Adding some grains of Salt ...

- Ansible modules
- Playbooks
- Execute Salt commands (salt-ssh) on Ansible managed systems
Step 1: Salt makes it tasty!
Salt makes it tasty!

- Running playbooks as Salt jobs.
- Execute Ansible modules on the salt-minion system.
- Running playbooks in SLS files!

- Ansible modules
- Playbooks
- Execute Salt commands (salt-ssh) on Ansible managed systems
- New growing Salt managed IT
- Event-driven orchestration on a hybrid environment
- Ansible modules
- Playbooks
- Execute Salt commands (salt-ssh) on Ansible managed systems
Step 2: Spreading the Salt all over the place!
● Manage your legacy Ansible as Salt jobs.
● Real-time monitoring and event-driven orchestration in all your IT (salt-minion)
● The whole minion functionality on the old Ansible managed systems.
Demo time!
Questions?
Thank You!
Unpublished Work of SUSE LLC. All Rights Reserved.
This work is an unpublished work and contains confidential, proprietary and trade secret information of SUSE LLC. Access to this work is restricted to SUSE employees who have a need to know to perform tasks within the scope of their assignments. No part of this work may be practiced, performed, copied, distributed, revised, modified, translated, abridged, condensed, expanded, collected, or adapted without the prior written consent of SUSE. Any use or exploitation of this work without authorization could subject the perpetrator to criminal and civil liability.

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 Novell, Inc. in the United States and other countries. All third-party trademarks are the property of their respective owners.