

White Paper
July 2021

SUSE Manager and Ansible: Making Automation Easier and More Powerful

2021

SUSE Manager is a single, powerful tool that makes it possible for your IT operations team to manage the complete lifecycle of RPM- and DEB-based Linux systems. With a feature-rich, web-based interface, you can use SUSE Manager to administer, deploy, configure, and audit all of your Linux systems, no matter if they are running on bare metal or within a virtual environment.

SUSE Manager includes several deployment, orchestration, and automation tools that not only make the job of the administrator easier but also give them even more power to work faster. With SUSE Manager in place, you can minimize staffing costs and maximize efficiency.

One way SUSE Manager accomplishes this is by integrating with Ansible. And although these features are already available using the built-in Salt components with SUSE Manager, the addition of Ansible will offer important options for those already familiar with the technology.

What Is Ansible?

Before we dive into how SUSE Manager makes Ansible more powerful and accessible, we should first define this tool. Most IT pros are already familiar with Ansible, as it is used by over 77 percent of businesses who require the use of centralized configuration management. For those that aren't familiar

with Ansible, it is an open source tool used for:

- Provisioning
- Configuration management
- Application deployment

Ansible runs on the Linux operating system and can be used to configure Unix-like and Windows systems (both desktop and server).

The best way to describe Ansible is to imagine you have hundreds (or thousands) of computers on a network and you need to prepare software to be deployed, deploy software to some (or all) of those computers, or make a configuration change to any number of those systems. Doing this manually would take days or weeks. However, deploying from a centralized manager only takes minutes or hours.

For any business looking to save money and keep systems running smoothly and seamlessly, this is a must-have functionality and one that Ansible delivers. Because Ansible makes it possible to automate these tasks, they can even be built into your DevOps pipeline, increasing efficiency and reliability.

Ansible works by way of a controller node that connects to managed nodes on a network. The Ansible controller node sends small programs (called Ansible modules)

that execute various actions (such as installing applications or configuring the OS or installed software) to the managed nodes via SSH. A module is nothing more than a description of what should be true on any given managed node at any given time. For instance, an administrator might create a module that defines a state that all Linux desktops should include the latest version of certain pieces of software. Say, for instance, you want to make sure all Linux desktops run the most recent iteration of the Firefox browser. Ansible modules make this possible. To successfully achieve this, login access (such as with SSH keys) to the managed nodes must be configured on the controller node.

One of the most important aspects of Ansible is the playbook, which is a repeatable, reusable configuration management and deployment script that you write and the Ansible controller node deploys to the managed nodes. A single Ansible playbook can target multiple system types. For example, you have managed nodes that function as web servers and some that function as database servers. You could create a single playbook that would update both the Apache and PostgreSQL server that might look like:

```
- name: Update web servers
hosts: webservers
remote_user: root

tasks:
- name: Update Apache to the latest version
  ansible.builtin.yum:
    name: httpd
    state: latest
- name: Write the apache config file
  ansible.builtin.template:
    src: /srv/httpd.j2
    dest: /etc/httpd.conf

- name: Update PostgreSQL
hosts: databases
remote_user: root

tasks:
- name: Update postgresql to the latest version
  ansible.builtin.yum:
    name: postgresql
    state: latest
- name: Ensure that postgresql is started
  ansible.builtin.service:
    name: postgresql
    state: started
```

Once you've created the playbook, you can then have Ansible automatically deploy it to the necessary systems, be they in the tens, hundreds, or thousands.

It's also important to understand what the Ansible inventory file does. This particular configuration file defines the host and

groups of hosts that will receive commands, modules, and tasks from the controller node. In other words, it's a list of your managed nodes, organized into tasks (such as web hosts and database hosts), locations (such as Warehouse A and HQ), or individual systems. An inventory file might look something like this:

```
all:
  hosts:
    mail.example.com:
  children:
    webservers:
      hosts:
        wb1.example.com:
        wb2.example.com:
    dbservers:
      hosts:
        db1.example.com:
        db2.example.com:
        db3.example.com:
```

Now that you have a fundamental idea of how Ansible works, let's see how it integrates with SUSE Manager.

How Does Ansible Integrate with SUSE Manager?

Currently, SUSE Manager works with Ansible 2.9 (LTS), which is shipped in the SUSE Linux Enterprise 15 Client Tools channel and includes the following integrations:

- Introspection of inventory files: After defining an inventory path, you can then use SUSE Manager to introspect its contents. This is taken care of in System |

Ansible | Inventories.

- Discovery of playbooks: After defining a directory of Ansible playbooks, SUSE Manager can discover them from within System | Ansible | Playbooks.
- Execution of playbooks: Playbooks can be scheduled for execution from within System | Ansible | Playbooks. After selecting a playbook for execution, you can then select the inventory to deploy the playbook from the drop-down element. If no inventory is selected, the default inventory configured in your control node will be used. You can then choose the time for playbook execution or select an action chain.

As of SUSE Linux Enterprise Server (SLES) 15, Ansible will be provided as packages that can be easily deployed to other SLES machines. For non-SUSE operating systems (such as RHEL, Ubuntu, and Debian), you will have to download those packages from your operating system repository or another provider. Initially, all controllers will be enrolled and inventory and playbooks are shown. All discovered playbooks can be run from within SUSE Manager (which will in turn trigger Ansible on the Ansible controller).

Why Would You Choose SUSE Manager's Ansible Features over the Standalone Ansible Platform?

Ansible and Salt are different but complementary. If you are already familiar with SUSE Manager, you will realize that some of

the use cases can also be fulfilled by using Salt.

When it comes to a one-time operation, also known as “one-shot,” Ansible, as we have seen, is a great tool, but it misses some features that SUSE Manager will add, such as:

- **Auditing:** By executing your playbooks from SUSE Manager, you will keep track of these executions, including the result, the output, and any problems.
- **Reusability:** You can connect to your existing Ansible environment from SUSE Manager and just use the same playbooks and inventory.
- **Scheduling:** You can decide when to run your playbooks

However, as we said, Ansible and Salt are different. But what is the main difference? While **Ansible is imperative** (it tells the system what is the desired configuration, but it doesn't track any changes, nor revert to the desired state), **Salt is declarative** (you specify the desired status, and if something or someone changes the state, Salt will track it and reapply the desired status).

Currently, there are a few reasons why you'd want to work with Ansible integration in SUSE Manager:

- If your company is currently using Red Hat Satellite to manage RHEL, or Canonical Landscape to manage Ubuntu,

etc., and you have Ansible automation and expertise, rewriting all of that in Salt before your company can jump to SUSE Manager would be a huge barrier to entry. By providing Ansible integration, you will be able to reuse your existing automation and migrate to SUSE Manager immediately. This would then make it possible for you to start writing your new automation in Salt or rewrite Ansible playbooks in Salt (if necessary).

- You want to deploy some application that provides Ansible automation (e.g., to deploy, secure, etc.), but the vendor of that particular application does not provide Salt automation for a particular use case or feature. When that is the case, SUSE and SUSE Manager will be there to help make it possible.
- You're looking to add real-time monitoring along with the power of Ansible.
- You're looking for more event-driven orchestration and scalability.
- You want a GUI tool for managing Ansible playbooks.
- Ansible users will have a clean migration path to SUSE Manager, without losing the effort they have invested in their Ansible playbooks. So not only will you not lose the power and flexibility found in Ansible, you'll gain the added orchestration, provisioning, and automation found in SUSE Manager. In other words, you won't have to rebuild what you've already done with Ansible, during or after the migration to SUSE Manager.

With the help of SUSE Manager and Salt, you can achieve real-time monitoring with beacons, event-driven orchestration with reactors, scalability with ZeroMQ/hub, and a first-class, web-based interface.

Possible use cases for Ansible include:

- Provisioning
- Configuration management
- Application deployment
- Continuous delivery
- Network automation
- Security automation
- Orchestration

Some of these use cases overlap with SUSE Manager, which means you can be sure you will gain even more power and flexibility by combining the two. In addition, you can address situations where you need to execute an action once as well as situations where you need to maintain the configuration over time.

How Easy Will It Be for Ansible Managers to Migrate to SUSE Manager?

Simple: Just register your existing Ansible controller with SUSE Manager and start driving those Ansible clients with SUSE Manager. That's it. It'll take you five minutes to make this happen. And, once you register the Ansible clients as Salt clients (Salt Minion agent or agentless with Salt SSH), then you will also benefit from a host of other features

offered in SUSE Manager (that you won't find in standalone Ansible).

SUSE Manager 4.2 will also be able to register and operate Ansible control nodes through the UI, in Systems | Overview | Add Systems. After selecting the Ansible control node from Add-on Systems Types, the controller will be added and displayed as an Ansible page in the system profile.

If you already have a collection of playbooks on SUSE Enterprise Linux Server, you can easily add the directory housing those playbooks in the Control Node sub-tab of the control node's Ansible tab. Type the directory of your Ansible playbooks in the Add a Playbook Directory text field, and click Save. You can add as many directories as you need. You can then set your inventory files by typing the name of a specific inventory in the Add an Inventory File section and then click Save.

Once you've done this, your playbooks and inventories will be available to use in SUSE Manager. You'll then find those playbooks and inventories in the Playbooks and Inventories sub-tabs under the controller node's Ansible tab.

Caveats to the SUSE Manager/Ansible Integration

There are, of course, a few things to note about this integration. One thing to remember is

that the first release of SUSE Manager that includes Ansible support should be considered a “tech preview.” Once SUSE Manager v4.2 is released, Ansible will be much more deeply integrated.

For that tech preview, some of the caveats to the Ansible integration include:

- The Ansible control nodes must be registered as Salt clients.
- SUSE Manager only operates your Ansible control node, it does not configure it.
- Ansible inventories are not yet created by SUSE Manager but use the existing inventories in the Ansible control node.
- Currently, the only version of Ansible that is supported is 2.9.
- You can use SLES 15 SP3 as your Ansible control node (so there’s no need to install and manage the control node from a separate machine).
- Although SUSE Manager supports Ansible, it assumes an Ansible host runs somewhere on your network and has playbooks, inventories, etc. SUSE Manager then plugs into that existing host.
- Variables within playbooks (i.e., that pass the parameter of what exactly is to be done instead of creating a playbook for every single situation) are not supported via SUSE Manager at this time. This feature will be supported and provided as a Maintenance Update for 4.2.

- Although Ansible is composed of several products and projects, SUSE Manager only leverages Ansible technology, which is the Ansible Engine. SUSE Manager will not integrate with Ansible Tower, Content Collections, Automation Hub, Automation Analytics, and the Automation Services Catalog.

Conclusion

Ansible is an incredibly powerful provisioning, configuration, and deployment tool, one that many IT admins and businesses would struggle without. But when you combine the power of Ansible with the automation and lifecycle management abilities of SUSE Manager, there’s almost nothing you cannot do for your IT landscape. If your developers and administrators are already familiar with Ansible, you will be doing your business a major service by upping their game with the added benefits of SUSE Manager. And with the Ansible integration, you can harness the declarative nature of Salt and the imperative nature of Ansible. With both of these tools on your side, the sky’s the limit for systems management via SUSE Manager.

SUSE
Maxfeldstrasse 5
90409 Nuremberg
www.suse.com

For more information, contact SUSE at:
+1 800 796 3700 (U.S./Canada)
+49 (0)911-740 53-0 (Worldwide)

Thank You