

Research Paper on Redhat Ansible Automation Platform

Akshay Kushwaha

Department of Information Technology, KeraleeyaSamajam's Model College, Dombivli West, Mumbai, Maharashtra

Abstract: In today's IT infrastructure there are numerous tasks which can overwhelm the number one resources of any organization i.e. People. Let it be a developer, tester, analyst, etc they always have some repetitive tasks which they wish they didn't have to do every day. That's where automation comes in which is a technology by which a process or a procedure could be done with minimal human assistance. This research paper focuses on the whole ansible framework and is it better than the ones available in the market.

I. Introduction

IT automation is the process of creating software and systems to replace repeatable processes and reduce manual intervention. It accelerates the delivery of IT infrastructure and applications by automating manual processes that previously required a human touch. With IT automation, software is used to set up and repeat instructions, processes, or policies that save time and free up IT staff for more strategic work. With the rise of virtualized networks and cloud services that require rapid, complex provisioning, automation is an indispensable strategy for helping IT teams deliver services with improved speed, consistency, and security. IT automation is a powerful tool that can scale a business, provide significant cost savings, and allow IT staff to focus on strategic rather than administrative work. A wide range of data centre and cloud operations can be automated, resulting in faster operations. Thanks to automation, IT environments can scale more quickly with fewer errors and are more responsive to business needs. A fully automated environment can reduce the time to delivery for production-ready resources from weeks to less than a day. Redhat ansible is one of the most popular automation platforms among the IT organizations which is an open-source IT Configuration Management, Deployment & Orchestration tool. It aims to provide large productivity gains to a wide variety of automation challenges. This tool is very simple to use yet powerful enough to automate complex multi-tier IT application environments. In this research paper we are going to have an in depth review of this technology and find out what is this tool capable of and is it actually good enough to replace any existing tool.

II. What is Redhat Ansible?

Ansible is an IT automation tool. It can configure systems, deploy software, and orchestrate more advanced IT tasks such as continuous deployments or zero downtime rolling updates. Ansible's main goals are simplicity and ease-of-use. It also has a strong focus on security and reliability, featuring a minimum of moving parts, usage of OpenSSH for transport (with other transports and pull modes as alternatives), and a language that is designed around auditability by humans—even those

not familiar with the program. Ansible manages machines in an agent-less manner. Ansible is decentralized—it relies on your existing OS credentials to control access to remote machines. If needed, Ansible can easily connect with Kerberos, LDAP, and other centralized authentication management systems. Ansible releases a new major release of Ansible approximately three to four times per year. The core application evolves somewhat conservatively, valuing simplicity in language design and setup. Contributors develop and change modules and plugins, hosted in collections since version 2.10, much more quickly.

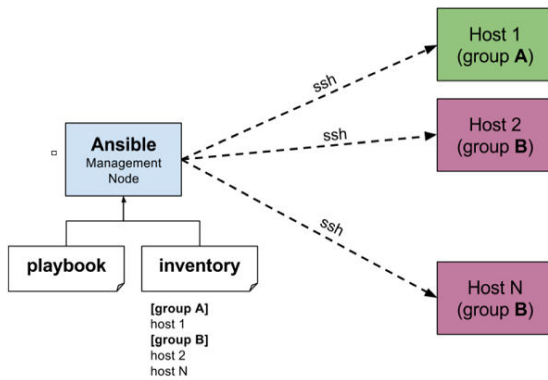
III. Features and Benefits

Red Hat Ansible Automation Platform integrates Red Hat's automation suite consisting of Red Hat Ansible Tower, Red Hat Ansible Engine, and use-case specific capabilities for Microsoft Windows, network, security, and more, along with Software-as-a-Service (SaaS)-based capabilities and features for organization-wide effectiveness. Whether an organization is just beginning its automation journey or is working to expand automation across more use cases and domains, Red Hat Ansible Automation Platform provides the tools to implement automation quickly. Organizations can use certified Ansible content without having to write their own playbooks and combine them with the correct modules and roles. Multiple roles within the business can use Red Hat Ansible Automation Platform, bringing the power of enterprise automation to more than just traditional DevOps workflows. Ansible Automation Platform provides security for content management, powerful analytics around automation deployments, and building blocks that can shorten the time to value when getting started with automation. Its SaaS-based services, available with other Red Hat cloud services at cloud.redhat.com, help deliver a more consistent automation user experience and fuel better collaboration to solve more IT challenges at once.

IV. How RedhatAnsible works?

The picture given below shows the working of Ansible.

Ansible works by connecting to your nodes and pushing out small programs, called "Ansible modules" to them. Ansible then executes these modules (over SSH by default), and removes them when finished. Your library of modules can reside on any machine, and there are no servers, daemons, or databases required.



The management node in the above picture is the controlling node (managing node) which controls the entire execution of the playbook. It's the node from which you are running the installation. The inventory file provides the list of hosts where the Ansible modules need to be run and the management node does a SSH connection and executes the small modules on the host machine and installs the product/software.

Beauty of Ansible is that it removes the modules once those are installed so effectively it connects to host machine, executes the instructions and if it's successfully installed removes the code which was copied on the host machine which was executed.

V. Comparison with Other Automation Platforms

Choosing the right configuration manager doesn't mean an organization needs to seek out the best tool on the market. What is best for one company may be wrong for a different company. Every configuration management tool, including Ansible, Chef, Puppet and SaltStack, caters to specific organizational goals and preferences. Ansible and Puppet, for instance, are agentless; Chef is not. With SaltStack, users have choices about the use of agents. If an organization has rigid OS requirements, that will dictate which tools to give a close look and which to cross off the list. Some of the selection process is about comfort, not architectural fit. Consider programming languages. A team that's not skilled in Python, for example, might not be as comfortable with Ansible as one that knows that language well. If PowerShell is of high importance to your team, you will probably want to give a close look at Puppet Enterprise. In the end, the strengths and competencies of a team will determine which configuration product will be

TABLE I: A comparison of popular Infrastructure-as-Code platforms [12].

	Chef	Puppet	RedhatAnsible	SaltStack	CloudFormation	Terraform
Code	Open	Open	Open	Open	Closed	Open
Cloud	All	All	All	All	AWS only	All
Type	Config Mgmt	Config Mgmt	Config Mgmt	Config Mgmt	Orchestration	Orchestration
Infrastructure	Mutable	Mutable	Mutable	Mutable	Immutable	Immutable
Language	Procedural	Declarative	Procedural	Declarative	Declarative	Declarative
Architecture	Client/Server	Client/Server	Client-only	Client/Server	Client-only	Client-only

VII. Structure of RedhatAnsible Scripts

most suitable. Additionally, Ansible can run as a remote configuration manager on a single machine. As the control node, this machine can control many other machines within the environment. Ansible avoids traditional, error-prone scripting approaches by working toward desired states. If the computers within a network aren't updated to reflect an OS version (or include key applications), the software automatically remedies this issue. Other configuration managers might show the path necessary to achieve the same thing, but that puts the onus on IT teams, adding a task instead of removing one. A central goal for configuration is the use of modules, which support certain actions and contextual automations. IT professionals can choose from over 1,300 modules in Ansible's main distribution.

VI. LAB Installation

Mainly, there are two types of machines when we talk about deployment –

- Control machine – Machine from where we can manage other machines.
- Remote machine – Machines which are handled/controlled by control machine.

There can be multiple remote machines which are handled by one control machine. So, for managing remote machines we have to install Ansible on control machine.

Control Machine Requirements:

Ansible can be run from any machine with Python 2 (versions 2.6 or 2.7) or Python 3 (versions 3.5 and higher) installed.

Note – Windows does not support control machine.

By default, Ansible uses ssh to manage remote machine. Ansible does not add any database. It does not require any daemons to start or keep it running. While managing remote machines, Ansible does not leave any software installed or running on them. Hence, there is no question of how to upgrade it when moving to a new version. Ansible can be installed on control machine which have above mentioned requirements in different ways. You can install the latest release through Apt, yum, pkg, pip, OpenCSW, pacman, etc

Once the communication chain is set, scripts for remote management could be created. The rudiments of these scripts are modules, playbooks and roles as depicted in Fig. 2.

- Playbook is the file containing the order of the commands, composed of aforementioned modules and is represented by a YAML file.
- Roles are complex structures that include tasks, modules, handlers and files. Role is composed of a directory that has sub-directories which contain a main.yml file, which specifies the sequence of operations to be performed.
- Modules are simple scripts focused on simple system actions. They can be run as a single command or as a part of more complex scripts, called playbooks.

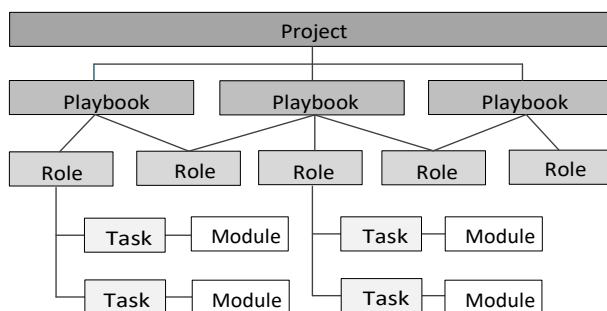


Fig. 2: Structure of RedhatAnsible scripts

Playbook is a file containing the commands to be executed on that host. It also defines the host guest group in which the tasks are run. Tasks utilize modules for unattended installation of applications on target operation systems. The following command executes the Playbook.

```
ansible-playbook run-example-playbook.yml
```

The Playbook can have a tremendous amount of information about the host station. If the playbook has to be as simple as possible, you can use a role. Ansible roles help to divide repeatable scripts into separate units. This makes it easy to read and allow the use of other necessary files, such as .sh, .ps, .conf, .msi and other scripts. Roles are ways of automatically loading certain files, tasks, and handlers based on a known file structure. Certain role is composed of a directory that has a sub-directory which contain a main.yml file. It, same as a playbook, specifies the sequence of operations to be performed. Before the installation itself, it is necessary to determine which type of the operating system will be installed. Another requirement is the availability of the repository application at the manufacturer's site. The last

condition is to select available modules to meet the set of defined requirements. Modules allow to control machine system resources, such as services, packages, files, permissions, etc. An example of a module for getting system information from a guest host may be as follows

```
ansible -i hosts windows -m setup
```

VIII. Redhat Ansible Ad-hoc commands

Ad hoc commands are commands which can be run individually to perform quick functions. These commands need not be performed later. For example, you have to reboot all your company servers. For this, you will run the Adhoc commands from '/usr/bin/ansible'. These ad-hoc commands are not used for configuration management and deployment, because these commands are of one-time usage. ansible-playbook is used for configuration management and deployment.

IX. Conclusion

Today, the infrastructure design is the software life-cycle phase that defines and configures the software infrastructure needs for that software as well as the number and type of physical hosts or virtual machines required. Redhat ansible helps streamline this process and provide the automation required to the help the users with day-to-day tasks and minimize repetitive actions in a infrastructure.

ACKNOWLEDGEMENT

I would like to thank KeraleeyaSamajam's Model College for providing me with an opportunity to present this research paper. And also, I would also like to thank Divya Ma'am for assistance and comments that greatly improved the manuscript.

REFERENCES

- [1] M. Mohaan and R. Raithatha, *Learning Ansible*. Packt Publishing Ltd, 2014.
- [2] M. Balliauw and X. Decoster, "Automated Delivery," in *Pro NuGet*, pp. 179–214, Springer, 2013.
- [3] L. Hochstein and R. Moser, *Ansible: Up and Running: Automating Configuration Management and Deployment the Easy Way*. "O'Reilly Media, Inc.", 2017.
- [4] K. Morris, *Infrastructure as code: managing servers in the cloud*. "O'Reilly Media, Inc.", 2016.