Ansible: Introduction to this open-source automation platform

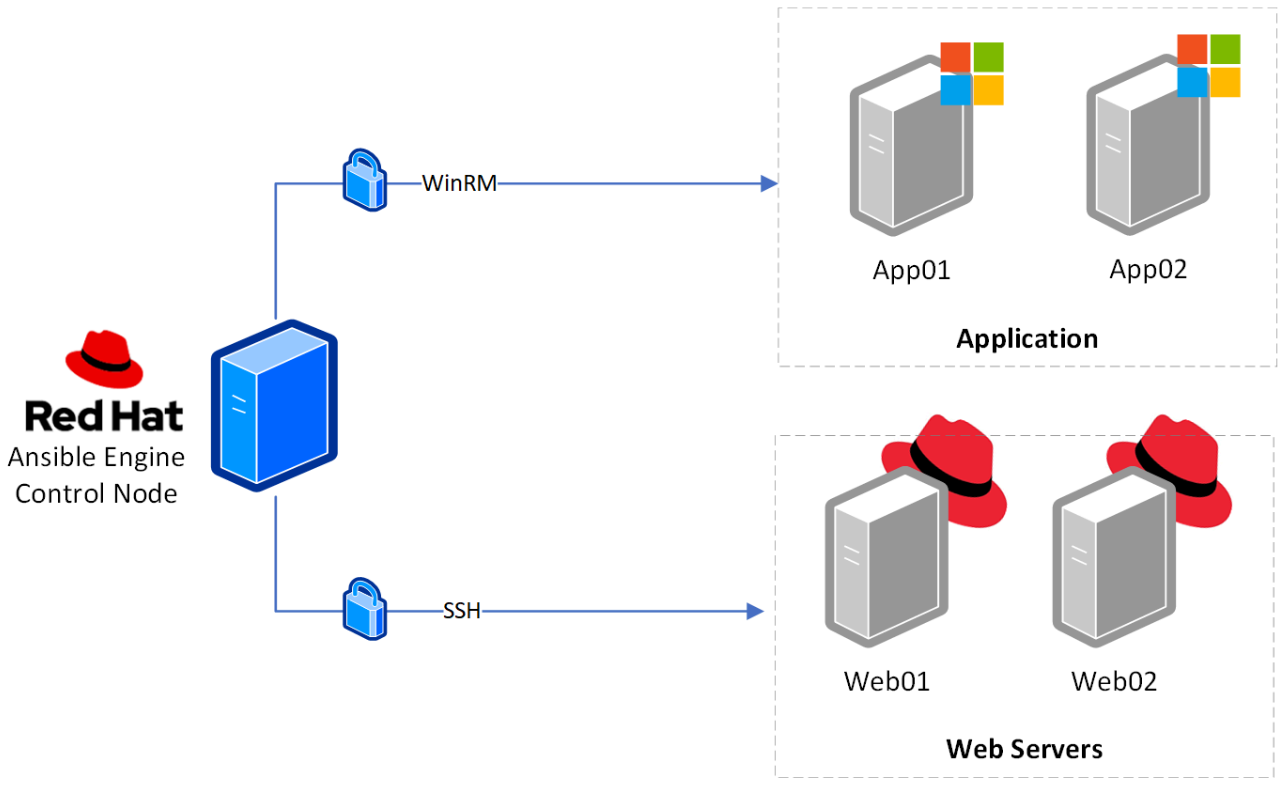
Ansible is an open-source solution managed by Red Hat, and it is part of the Red Hat Ansible Automation Platform. Ansible helps organizations to orchestrate and automate applications and environments from a central location to be consistent based on defined Ansible configuration files (playbooks. It uses YAML format, which means that it is easy to read, understand, and adapt to new requirements.

Because Ansible is an open-source project, we will use modules, which are how the platform interacts with products and features, and it has support for several technologies, including Linux, Windows, Microsoft Azure, NetApp, F5, Check Point, Aruba, Cisco, Google Cloud, and AWS. Long story short, we will be able to manage operating systems, infrastructure, applications, and networks from a single location.

A look under the hood

One of the key features of Ansible is that it does not require an agent. The *control node* will connect to the target system using the protocols available as depicted in the image below and execute the instructions to make sure that the target system is configured based on the Ansible Playbooks/Roles/Ad-hoc commands (we will delve into these terms in more detail in this series).

Another key feature is that it can be used to address simple scenarios and complex ones using the same process across the border, which helps solutions architects and operations to use the technology.



Infrastructure-as-a-code (IaaC) is a theme that is here to stay, and we have explored several flavors of it here at TechGenix. I covered a few topics around DevOps in the Azure DevOps and ARM templates in my previous articles. In this article series, we will switch gears and explore Ansible and how to integrate it with existing solutions.

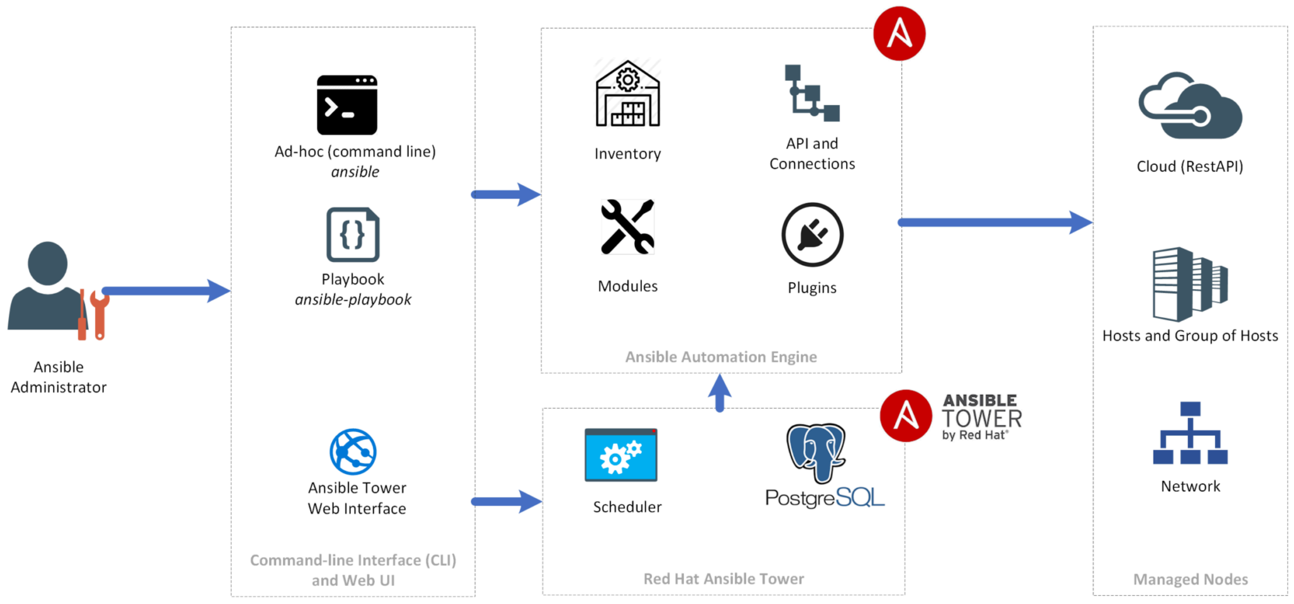
The previous diagram shows the *Red Hat Ansible Engine*, which can be installed on any Linux to use Ansible and start managing your environment. However, they have a solution that complements the Ansible Engine, and it is called Red Hat Ansible Tower, which is a web service that provides a web console and several other features to help the corporate environment to be more successful. Here are a few of the features:

- Dashboard
- Real-time updates
- Workflow editor for multiple playbooks
- Capabilities to support CI/CD (continuous integration/continuous deployment)
- Auditing
- High availability and scalability for the Ansible Tower (cluster)
- Integrated notifications
- A job that can be continuously running to keep compliance at its best
- Integration with Active Directory
- Role-based access control
- Allow a single button to trigger actions

If you are studying for your Red Hat Certified Engineer (RHCE) exam, then Ansible is must-have knowledge before taking the exam. The certification has changed, and the new exam is based in Ansible with a few topics from RHCSA (Red Hat Certified System Administrator).

High-level architecture

We will explore several areas of Ansible here at TechGenix, but before going into details, we should get a good understanding of how it works from a high-level standpoint.



If we look at the diagram above, and we start from left to right, the first thing that we will notice is that the administrator can interact with Ansible in several ways using either a CLI (command line interface) or a web interface.

The command-line interface has two different approaches: We can send commands from the command-line using **ansible** executable file, and that is called *Ad-hoc* interaction. The second approach is using a *playbook*, which is a YML file that contains human-readable instruction to perform tasks on the desired hosts or group of hosts when calling playbooks, and we will use the **ansible-playbook** executable file.

If we want to use a web interface and have a more corporate group of features to improve Ansible, then we need to add **Red Hat Ansible Tower,** which is the software component that will take Ansible to the next level. It can be one or more servers to provide high-availability and fault tolerance, and it contains a PostgreSQL database on the backend.

The critical component that makes everything works is the *Ansible Automation Engine*, and it is comprised of key areas: **Inventory, Modules, API and Connections**, and **Plugins**.

- **Inventory**: It is a predefined list of nodes that Ansible can execute its playbooks/ad-hoc commands against it. An inventory could be as simple as a static file, expression using a range of servers, customized, and even dynamic.
- **Modules:** Ansible comes with 400+ modules, and they are a piece of code that allows Ansible to interact and perform tasks on the target hosts. We define the module that we want to use, and then the actions/tasks that will be executed as part of that module.
- **API and Connections**: This is the transport communication between Ansible and the managed nodes. It can be SSH/WinRM or even API when talking to cloud platforms like Azure.
- **Plugins**: They improve the Ansible universe by adding a piece of code that enhances the current environment to manage or perform additional tasks.

One key feature that brings most of the components that we discussed briefly is the **playbook**. It is a list of tasks that will be executed in the order described in the document. For each task, we will have a module and the operations that we want to perform on the desired nodes. Since it is a YAML file, we know that it is easy on the eye, and even non-technical people can read and have a good understanding of what is going on.

Stay tuned for Ansible in-depth

In this initial article, we went through a high-level overview of Ansible and Ansible Tower. We covered their core components and how they interact with each other to manage nodes. In other articles, we will be going more in-depth in some of the Ansible components and how to use and apply them to real-world scenarios.

Featured image: Shutterstock / TechGenix photo illustration