Ansible

Ansible is an open-source IT engine that automates application deployment, cloud provisioning, intra service orchestration, and other IT tools.

Ansible is easy to deploy because it does not use any **agents** or **custom security** infrastructure on the client-side, and by pushing modules to the clients. These modules are executed locally on the client-side, and the output is pushed back to the Ansible server.

It can easily connect to clients using **SSH-Keys**, simplifying though the whole process. Client details, such as **hostnames** or **IP addresses** and **SSH ports**, are stored in the files, which are called inventory files. If you created an inventory file and populated it, then Ansible can use it.

Ansible uses the playbook to describe automation jobs, and playbook, which uses simple language, i.e., **YAML**. YAML is a human-readable data serialization language & commonly used for configuration files, but it can be used in many applications where data is being stored.

A significant advantage is that even the IT infrastructure support guys can read and understand the playbook and debug if needed.

Ansible is designed for multi-tier deployment. Ansible does not manage one system at a time, and it models IT infrastructure by describing all of your systems are interrelated. Ansible is entirely agentless, which means Ansible works by connecting your nodes through **SSH** (by default). Ansible gives the option to you if you want another method for the connection like **Kerberos**.

Ansible pushes small programs after connecting to your nodes which are known as "**Ansible Modules**". Ansible runs that module on your nodes and removes them when finished. Ansible manages the inventory in simple text files (These are the host's files). Ansible uses the host file where one can group the hosts and can control the actions on a specific group in the playbooks.

Why Use Ansible

Here are some important reasons for using Ansible, such as:

* Ansible is free to use by everyone.
* Ansible is very consistent and lightweight, and no constraints regarding the operating system or underlying hardware are present.
* It is very secure due to its agentless capabilities and open **SSH** security features.
* Ansible does not need any special system administrator skills to install and use it.
* Ansible has a smooth learning curve determined by the comprehensive documentation and easy to learn structure and configuration.
* Its modularity regarding **plugins, inventories, modules,** and **playbooks** make Ansible perfect companion orchestrate large environments.

Ansible History

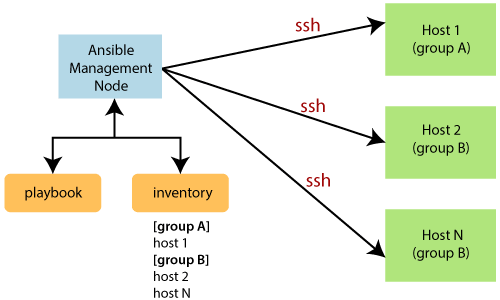
Here are some essential points from the history of Ansible, such as:

* **Michael DeHaan** developed Ansible, and the Ansible project began in **February 2012**.
* The creator of **Cobbler** and **Func** is also the controller of the **Fedora Unified** network.
* **RedHat** acquired the Ansible tool in 2015.
* Ansible is included as part of the **Fedora distribution** of the Linux.
* Ansible is also available for **RedHat Enterprise Linux, Debian, CentOS, Oracle Linux,** and **Scientific Linux** via Extra Packages for Enterprise Linux **(EPEL)** and **Ubuntu** as well as for other operating systems.

Ansible Workflow

Backward Skip 10sPlay VideoForward Skip 10s

Ansible works by connecting to your nodes and pushing out a small program called **Ansible modules** to them. Then Ansible executed these modules and removed them after finished. The library of modules can reside on any machine, and there are no daemons, **servers,** or **databases** required.



In the above image, the **Management Node** is the controlling node that controls the entire execution of the playbook. The **inventory** file provides the list of hosts where the Ansible modules need to be run. The **Management Node** makes an **SSH** connection and executes the small modules on the host's machine and install the software.

Ansible removes the modules once those are installed so expertly. It connects to the host machine executes the instructions, and if it is successfully installed, then remove that code in which one was copied on the host machine.

Terms used in Ansible

Here are some important terms which are used in Ansible, such as:

|  |  |
| --- | --- |
| **Terms** | **Explanation** |
| Ansible Server | It is a machine where Ansible is installed and from which all tasks and playbooks will be executed. |
| Modules | The module is a command or set of similar commands which is executed on the client-side. |
| Task | A task is a section which consists of a single procedure to be completed. |
| Role | It is a way of organizing tasks and related files to be later called in a playbook. |
| Fact | The information fetched from the client system from the global variables with the gather facts operation. |
| Inventory | A file containing the data regarding the Ansible client-server. |
| Play | It is the execution of the playbook. |
| Handler | The task is called only if a notifier is present. |
| Notifier | The section attributed to a task which calls a handler if the output is changed. |
| Tag | It is a name set to a task that can be used later on to issue just that specific task or group of jobs. |