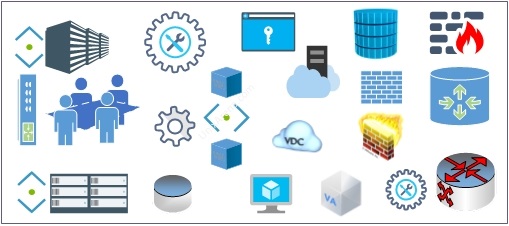
**Why do we need Configuration Management tool?**

Anyone who works as an operations engineer has witnessed a bunch of issues with manual configuration approach and more repetitive tasks which are time-consuming. How many times key resources left the company and new engineer struggle to understand the environment and start performing the tasks without escalation? Server configuration is a very broad landscape which needs to be maintained properly from the beginning. Organization Standard will be documented in KM but people will forget/miss to follow due to resource crunch, laziness and skill gap. Scripting is one of the option to automate and maintain the configuration but it’s not an easy task.



**What is Ansible?**

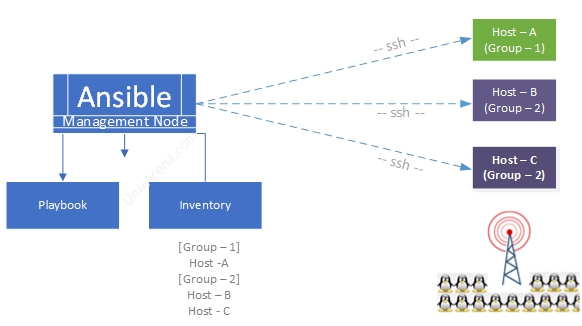
Configuration management and Orchestration tool is the solution to eliminate all the problem in the system management. Ansible is one of the most popular ones which is supported by Red Hat. Ansible is simple IT automation engine to save time and be more productive. Human resources can spend more time on innovation to make the operation more cost-effective.

**Why Ansible?**

* Ansible is free and Open Source.
* Agentless. Ansible doesn’t require any agent on client machines unlike other automation tool exists in the market (Puppet, Chef, Salt.). It uses SSH protocol to connect the servers. Ansible required Python to make the use of modules on client machines. Ansible also works with a system which doesn’t have python installed using the “raw” module.
* Ansible uses YAML language which is very easy to learn.
* Supported by Red Hat.

**How Ansible work?**

Ansible works by connecting to your server using “SSH” and pushing out small programs, called “Ansible modules” to it. Using these modules, playbooks (a small piece of YAML code), we should be to perform the specific task on all the ansible clients. The specific task could be installing the packages, restarting the services, rebooting the servers etc..There are lots of things that you could do using ansible.



**Ansible Use cases**

* Provisioning
* Configuraiton Management
* App Deployment
* Continous Delivery
* Security & Compliance
* Orchestration

**Ansible – Supported Operating Systems**

* Linux, including RHEL, CentOS, Fedora, Ubuntu, and others.
* Windows and Windows Server
* UNIX
* OS X

**Ansible – Supported Hypervisors**

* VMware
* Red Hat Enterprise Virtualization (RHEV)
* Libvirt
* Xenserver
* Vagrant

**Ansible – Supported Cloud platforms**

* [Amazon Web Services](https://www.ansible.com/integrations/cloud/amazon-web-services?hsLang=en-us)
* [Google Cloud Platform](https://www.ansible.com/integrations/cloud/google-cloud-platform?hsLang=en-us)
* [OpenStack](https://www.ansible.com/integrations/cloud/openstack?hsLang=en-us)
* [Microsoft Azure](https://www.ansible.com/integrations/cloud/microsoft-azure?hsLang=en-us)

**Ansible – Tower vs Engine:**

Here are the key difference between “**Ansible Engine**” and “**Ansible Tower**“.

|  |  |
| --- | --- |
| **Red Hat Ansible Engine** | **Red Hat Ansible Tower** |
| Support for the Ansible execution engine | A GUI dashboard |
| Support for Ansible core modules per product lifecycle | Role-based access control |
| Integration with Red Hat Enterprise Linux | Job scheduling |
|  | Graphical inventory management |
|  | Multi-playbook workflow |
|  | REST API |
|  | External logging integrations |
|  | Real-time job status updates |

**Ansible Tower without Red Hat’s subscription**

AWX is the upstream project from which the Red Hat Ansible Tower offering is ultimately derived. You could [download AWX](https://github.com/ansible/awx) and experience the Ansible Tower experience. To know more about[AWX project, check out here.](https://www.ansible.com/products/awx-project/faq)