

# **ANSIBLE USE CASES AND CASE STUDY**

**TASK:-**

* **Create a Article, blog or Video on how industries are solving challenges using Ansible.**

**What is Ansible?**

[Ansible](https://www.ansible.com/) is a software tool that provides simple but powerful automation for cross-platform computer support. It is primarily intended for IT professionals, who use it for application deployment, updates on workstations and servers, cloud provisioning, configuration management, intra-service orchestration, and nearly anything a systems administrator does on a weekly or daily basis. Ansible doesn’t depend on agent software and has no additional security infrastructure, so it’s easy to deploy.

Because Ansible is all about automation, it requires instructions to accomplish each job. With everything written down in simple script form, it’s easy to do version control. The practical result of this is a major contribution to the “infrastructure as code” movement in IT: the idea that the maintenance of server and client infrastructure can and should be treated the same as software development, with repositories of self-documenting, proven, and executable solutions capable of running an organization regardless of staff changes.

# How Ansible works?

In Ansible, there are two categories of computers: the *control node* and *managed nodes*. The control node is a computer that runs Ansible. There must be at least one control node, although a backup control node may also exist. A managed node is any device being managed by the control node.

Ansible works by connecting to nodes (clients, servers, or whatever you’re configuring) on a network, and then sending a small program called an Ansible *module* to that node. Ansible executes these modules over SSH and removes them when finished. The only requirement for this interaction is that your Ansible control node has login access to the managed nodes. [SSH keys](https://opensource.com/article/19/4/ssh-keys-seahorse) are the most common way to provide access, but other forms of authentication are also supported.

# 2

# Configuration Management Tools

There are multiple configuration management systems. The most popular are for Unix-based systems and include Ansible, Chef, Puppet, Saltstack, CFEngine.

All of them have pros and cons, but there is also a common trait they share — this is a new approach in the field of system administration. This new paradigm is called infrastructure as code. Before its creation, the process was as follows. Servers were set up, brought online, and then configured using various scripts to ease and automate many tasks. In most cases, this was required for every individual server. With this new approach, a configuration is created once and shared across many servers.

# What Can Ansible Do?

The following tasks can be solved using Ansible:

* OS (de)installation
* OS configuration
* creation and removal of users
* management of users’ passwords and keys
* creation and removal of virtual environments and containers
* deployment of code
* running scripts and tests, etc.

This is a short list of tasks that can be accomplished using Ansible, but the full range of options is much more comprehensive and only limited by the imagination.

# Key Features of Ansible

* Agentless — SSH protocol is used to connect to servers and work on them instead of external agents;
* Idempotent — same operations provide the same result as many times as they are carried out;
* Easy and expandable — Ansible is written in Python and uses YAML to write the commands, and both languages are used to teach the beginners as they are considered to be easy to learn.

We are going to take a simple path while studying Ansible, which will include examples and their explanations. However, it’s preferable to know what a programming language, variable, cycle, and vocabulary used will be. Also, a basic knowledge of YAML structures and simple Unix commands would be useful. For all interactions, we are going to be using the client installed on Ubuntu 18.04, and on a server with Ubuntu 18.04 installed on it.

# Ansible Playbooks

We have seen how to run single tasks or one time tasks using Modules, but what if you need to execute multiple tasks? Playbooks help to run them in a scripted way.

Playbooks define variables, configurations, deployment steps, assign roles, perform multiple tasks. For ****E.g.**** COPY / DELETE Files and Folders, install packages, start services. So primarily playbooks are defined to orchestrate the steps to multiple machines or servers and get them all to a certain desired state.

Playbook is written in[YAML format](https://www.softwaretestinghelp.com/yaml-tutorial/) with a .yml file extension. One needs to be very careful with the format and alignment which makes it very sensitive.

****It contains the following sections:-****

1. Every playbook starts with 3 hyphens ‘ — ‘
2. ****Host section**** — Defines the target machines on which the playbook should run. This is based on the Ansible inventory file.
3. ****Variable section**** — This is optional and can declare all the variables needed in the playbook. We will look at some examples as well.
4. ****Tasks section**** — This section lists out all the tasks that should be executed on the target machine. It specifies the use of Modules. Every task has a name which is a small description of what the task will do and will be listed while the playbook is run.

# Ansible Modules

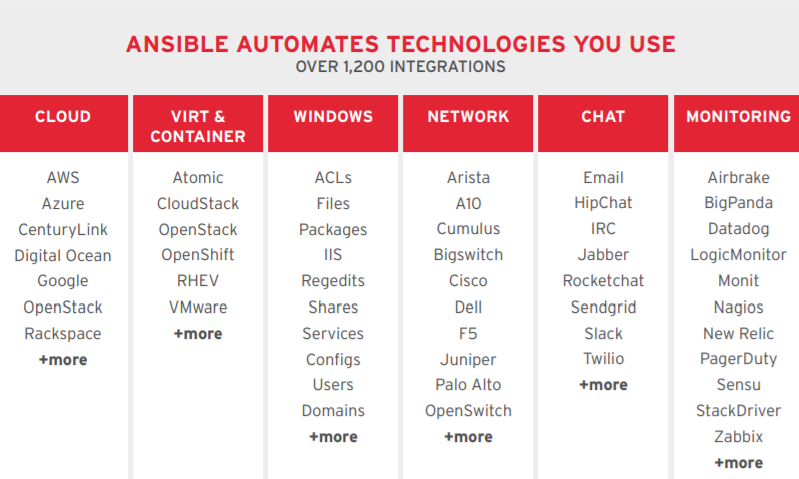
Let’s review three modules that allow us run commands on the remote hosts.

* ****raw****— allows us to run commands on the remote host even if Python is not installed on it, also the command is sent as it is and not checked by Ansible whether the syntax is correct.
* ****command —****a command is run directly past the shell installed for the user.
* ****shell —****the command is executed using shell mentioned in the */etc/passwd* file for the user, which is used by Ansible for the session (usually /bin/sh or /bin/bash).

# Ansible Case Study

Linux systems administrator Corban Raun was looking for a solution that was high-quality but also intuitive. While Raun was weighing various options, he had a friend advise him to look at a less widely known application, Ansible.Raun proceeded to use Ansible up through the present, a span of two years. He used the system for a variety of platforms and stacks such as web applications, user management, [secure virtual private server hosting](http://atlantic.net/vps-hosting/) configurations, and tracking.

He also used Ansible to access cloud services and for any projects needing repetition on a stable foundation.



For applications, you can automate building, testing, and code deployment to any environment in your infrastructure with auditing and reporting for the entire process. This allows you to eliminate manual image building and script execution and focus on innovation. In addition, Ansible easily integrates with the majority of existing version control systems and many other tools right out of the box.

# Use Cases for Modernizing and Automating Workflows



Managing an organization’s many tools and business processes is becoming increasingly complicated as technology expands. Whether your teams are performing their weekly system reboot, or looking to configure instances to a desired state, it’s no secret that automation is critical to increase speed, efficiency, productivity, and accuracy. Listed below are several instances1 where automation can help across your enterprise.

****Weekly system reboot:****There’s nothing worse than doing the same thing for 8 hours a day .Eliminate repetitive, manual processes with automation.

****Enforce security guidelines:****Rules are rules. It’s best to automate in an effort to achieve strict security standards.

****Monitor configuration drift:****Use check mode with Ansible tasks to enforce desired settings and see if your configuration has drifted.

****Disaster recovery:****Disaster recovery can involve a wide range of components. Act across different variables of the technology stack to identify problems and eliminate cross team dependencies.

****Command blaster:****Remarkably easy to write, you can run commands across your environment for any number of servers.

****Database binary patching:****Several databases use outdated binary sets. Patch the binaries in accordance with the release of the latest patch.

****Instance provisioning:****Use modules for several cloud providers to create new instances and tailor their configuration.

****Service license agreements:****Mistakes cost time and money. Eliminate errors that can crop up in detailed software contracts.

# Ansible’s Features and Capabilities

# 1. Configuration Management

Ansible is designed to be very simple, reliable, and consistent for configuration management. If you’re already in IT, you can get up and running with it very quickly. Ansible configurations are simple data descriptions of infrastructure and are both readable by humans and parsable by machines. All you need to start managing systems is a password or an SSH (Secure Socket Shell, a network protocol) key. An example of how easy Ansible makes configuration management: If you want to install an updated version of a specific type of software on all the machines in your enterprise, all you have to do is write out all the IP addresses of the nodes (also called remote hosts) and write an Ansible playbook to install it on all the nodes, then run the playbook from your control machine.

# 2. Application Deployment

Ansible lets you quickly and easily deploy multitier apps. You won’t need to write custom code to automate your systems; you list the tasks required to be done by writing a playbook, and Ansible will figure out how to get your systems to the state you want them to be in. In other words, you won’t have to configure the applications on every machine manually. When you run a playbook from your control machine, Ansible uses SSH to communicate with the remote hosts and run all the commands (tasks).

# 3. Orchestration

As the name suggests, orchestration involves bringing different elements into a beautifully run whole operation — similar to the way a musical conductor brings the notes produced by all the different instruments into a cohesive artistic work. For example, with application deployment, you need to manage not just the front-end and backend services but the databases, networks, storage, and so on. You also need to make sure that all the tasks are handled in the proper order. Ansible uses automated workflows, provisioning, and more to make orchestrating tasks easy. And once you’ve defined your infrastructure using the Ansible playbooks, you can use that same orchestration wherever you need to, thanks to the portability of Ansible playbooks.

# 4. Security and Compliance

As with application deployment, sitewide security policies (such as firewall rules or locking down users) can be implemented along with other automated processes. If you configure the security details on the control machine and run the associated playbook, all the remote hosts will automatically be updated with those details. That means you won’t need to monitor each machine for security compliance continually manually. And for extra security, an admin’s user ID and password aren’t retrievable in plain text on Ansible.

# 5. Cloud Provisioning

The first step in automating your applications’ life cycle is automating the provisioning of your infrastructure. With Ansible, you can provision cloud platforms, virtualized hosts, network devices, and bare-metal servers.

Next in our learning of what is Ansible, let us learn the Ansible architecture.

# Advantages of Ansible

Free: Ansible is an open-source tool.

**Very simple to set up and use**: No special coding skills are necessary to use Ansible’s playbooks (more on playbooks later).

**Powerful**: Ansible lets you model even highly complex IT workflows.

**Flexible**: You can orchestrate the entire application environment no matter where it’s deployed. You can also customize it based on your needs.

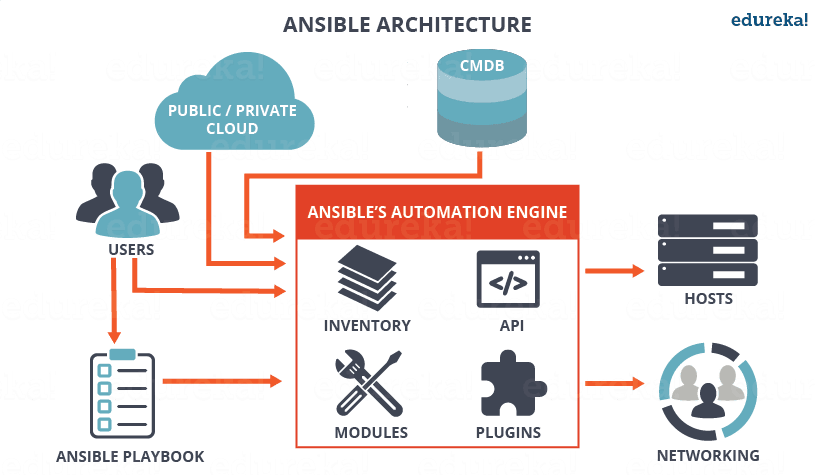
**Agentless**: You don’t need to install any other software or firewall ports on the client systems you want to automate. You also don’t have to set up a separate management structure.

**Efficient**: Because you don’t need to install any extra software, there’s more room for application resources on your server.

Next, in our path to understanding what ansible is, let us find out the features and capabilities of Ansible.

# Ansible Architecture

Now let’s talk a bit about the pieces that make up the Ansible environment.



# **1. Modules**

Modules are like small programs that Ansible pushes out from a control machine to all the nodes or remote hosts. The modules are executed using playbooks (see below), and they control things such as services, packages, and files. Ansible executes all the modules for installing updates or whatever the required task is, and then removes them when finished. Ansible provides more than 450 modules for everyday tasks.

# 2. Plugins

As you probably already know from many other tools and platforms, plugins are extra pieces of code that augment functionality. Ansible comes with a number of its plugins, but you can write your own as well. Action, cache, and callback plugins are three examples.

# 3. Inventories

All the machines you’re using with Ansible (the control machine plus nodes) are listed in a single simple file, along with their IP addresses, databases, servers, and so on. Once you register the inventory, you can assign variables to any of the hosts using a simple text file. You can also pull inventory from sources like EC2 (Amazon Elastic Compute Cloud).

# 4. Playbooks

Ansible playbooks are like instruction manuals for tasks. They are simple files written in YAML, which stands for YAML Ain’t Markup Language, a human-readable data serialization language. Playbooks are really at the heart of what makes Ansible so popular is because they describe the tasks to be done quickly and without the need for the user to know or remember any particular syntax. Not only can they declare configurations, but they can orchestrate the steps of any manually ordered task, and can execute tasks at the same time or at different times.

Each playbook is composed of one or multiple plays, and the goal of a play is to map a group of hosts to well-defined roles, represented by tasks.

# 5. API’s

Various API’s (application programming interfaces) are available so you can extend Ansible’s connection types (meaning more than just SSH for transport), callbacks, and more.

Now that we’ve come this far to understand what Ansible is, let us next look into the Ansible tower.

**Cisco Company**

Cisco is the worldwide technology leader that has been making the Internet work since 1984. Its people, products, and partners help society securely connect and seize tomorrow’s digital opportunity today.Cisco Systems is an IT and **networking** brand that specializes in switches, routers, cyber security, and IoT and whose logo seems to be on every office telephone or conference hardware.

Now, Let see how “Cisco Company” are Solving Challenges Using Ansible.



Cisco is a global leader in compute, network, and cloud solutions for enterprises, data centers, and any infrastructure in between. But as businesses expand and needs evolve, companies are searching for new approaches to meet increased end-user demand while continuing to deliver value to the business — all with limited resources. Automation will become an ever-increasing strategy for enterprises, because as they are finding out, automation is no longer about getting ahead — it is more about keeping up.

Red Hat can help. As the global, trusted leader in open source solutions for today’s enterprises, RedHat has been helping organizations of all shapes and sizes adapt to continually changing IT requirements. With Red Hat Ansible Automation as part of the Red Hat management portfolio, organizations are now able to gain greater efficiency, deliver value faster, and solve IT and business workflow challenges across the entire IT organization.

## **How cisco manage workload by using Automation**

Enterprises are under extreme pressure to manage the ever-increasing demand of applications, workloads, and larger, more complex networks. Despite increasingly tight budgets, global enterprises will collectively spend more than US$1.3 trillion in 2018 on digital infrastructure.2 They face an array of increasingly complex and evolving challenges, as their business’s digital footprint expands.

There is more infrastructure to deploy and more endpoints to manage but with fewer resources available and with less time to react and respond. Many organizations still manage their networks by logging into each network device manually, making any required changes — often during an emergency — with little to no tracking or validation. Within the Cisco universe alone, there are dozens of environments and platforms available to manage everything from network switching, routing, and firewall configurations, that make efficiently managing and orchestrating automation in an entire environment nearly impossible.

Research suggests that the vast majority of organizations are still managing through a command line on individual devices.3 Given the heavy workload and sprawling environment, many IT teams lack visibility into their infrastructure and lose track of network devices and their configurations — and the impact they may have on security and compliance.

Automating these processes — planning, testing, deployment, and maintenance of various infrastructure elements — is a transformational approach to simplifying IT operations. Yet adoption of automation solutions has been much slower among network teams than expected because:

* Automation tools are often specific to a specific function or platform.
* Adopting new methodologies that affect day-to-day processes and procedures can be daunting to traditional operators utilizing manual methods.
* Management’s strategy and operator’s activities require 100% alignment.

Successfully doing more with less requires all aspects of the IT organization to first buy-in and then to create and execute an action plan that incorporates automation as part of the culture.

**Ansible Automation Platform provides automation support for a wide range of Cisco products and platforms, including:**

* Networking — routing and switching
* Networking — Security
* Cisco software-defined networking (SDN) and controllers
* Wireless
* Servers

