

# ANSIBLE

Ansible is an open-source tool used for configuration management, application deployment, along with infrastructure orchestration.

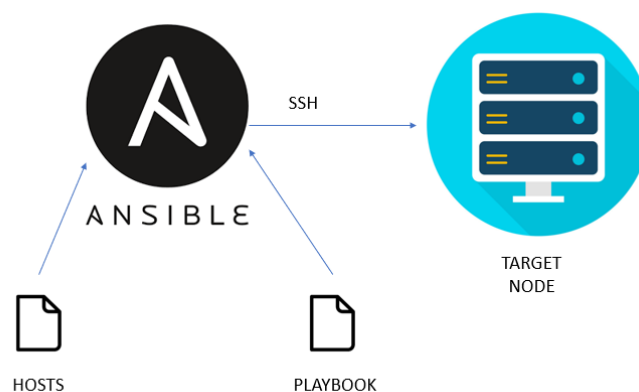
Ansible is a push-based configuration management tool

1. Ansible does not require any dedicated agent running on the target host machines.
2. The requirement for host machines is python installed on them and a proper ssh connection between the controller and the host machines has to be established.

## **Advantages:**

- free/open source
- Very simple to setup
- Agent less configuration tool

## **Architecture:**



**Modules:** Ansible works by connecting to your nodes and pushing out scripts called “Ansible modules” to them, Ansible then executes these modules (over SSH by default), and removes them when finished.

**Plugins:** While modules execute on the target system, plugins execute on the control node. Plugins offer options and extensions for the core features of Ansible

**Controller Machine:** Controller is a Machine where Ansible is installed

## **Installation:**

```
sudo apt-add-repository ppa:ansible/ansible
sudo apt update
sudo apt install ansible
```

## Ansible INVENTORY:

**Inventory:** Inventory is the place where we put information about all our managed machine i.e. host machines

### **Static Inventory**

- It is a file which contains the ip and configuration of target host machines on which we want execute our playbooks
- Default location of host file is /etc/ansible/hosts

## AD-HOC Commands

ansible <hosts> -m <module> -a <arguments of the module>

- ansible all -m ping
- ansible frontend -m ping
- ansible backend -m ping
- ansible all -m shell "touch f1"
- ansible all -m shell -a ls
- ansible all -m shell -a uptime

## Ansible Playbook

**Playbook:** Playbooks are list of tasks written in yaml format that gets executed on host machines

- Can contain n number of plays
- Each play is designated to run n number tasks
- Each task is designated to execute a single module (i.e., only one module per task)

### **To Execute Playbook:**

```
Ansible-playbook <file_name>.yaml
```

**become: true** will execute the task as root i.e sudo

### **state:**

- present will simply ensure that a desired package is installed.
- latest will update the specified package if it's not of the latest available version.
- absent will remove the specified package.
- build-dep will also install any dependency packages along with the desired package

**update\_cache: yes** tells Ansible's apt module to refresh the caches before applying whatever change is necessary (i.e., apt-get update)

## **Ansible facts:**

Ansible facts are data gathered about host machines and returned back to the controller. Ansible facts are stored in JSON format and setup module is used to gather all the facts

```
ansible all -m setup
```

## **Ansible Register and Debugging:**

Ansible register is a way to capture the output from task execution and store it in a variable.

### **Ansible run a task only if previous task is successfully changed using register module**

- Register module will save the output of a task to a variable in JSON format only if the task executes successfully
- We can register the output of one task and on another task, we can check whether variable has information or empty

## **Ansible Run specific tasks / plays**

using tags, we can run specific task or play in a playbook

### **To run only task with the tag**

```
ansible-playbook <playbook_name>.yaml --tags="unzip "  
ansible-playbook <playbook_name>.yaml --tags="git,unzip"
```

### **To skip a tagged play / task**

```
ansible-playbook <playbook_name>.yaml --skip-tags="git"  
ansible-playbook <playbook_name>.yaml --skip-tags="git,unzip"
```

## **Ansible Roles:**

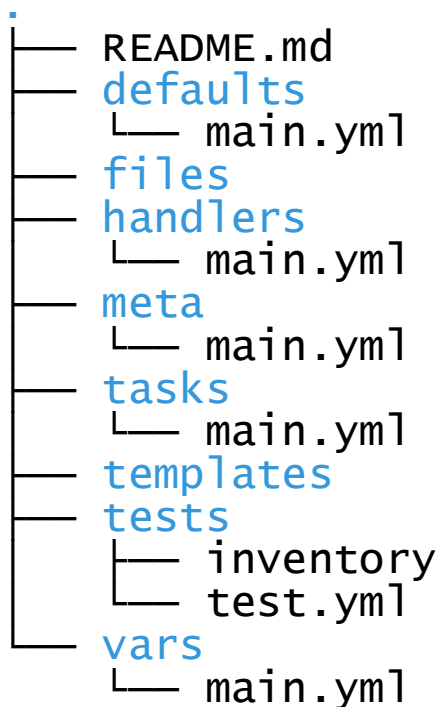
Ansible Roles are a way of breaking down a playbook into multiple files. Therefore, simplifying writing complex playbooks and making them easier to reuse.

Default location of roles is /etc/ansible/roles (Create the directory if it is not present)

Ansible provides a feature called Ansible Galaxy that helps you create roles.

```
ansible-galaxy init <role_name> → To create a Role
```

## The Directory Structure of Ansible Roles:



**Tasks** — Contains the main list of tasks that are to be executed by the role. It contains the main.yml file on which we define our tasks

**Handlers** — The handlers directory is used for storing Ansible handlers. Handlers are the tasks which are executed only if they are invoked by another task

**Defaults** — Contains the default variables that are going to be used by this role. It has the least precedence in Ansible Roles

**Vars** — This directory consists of other variables that are going to be used by the role.

**Files** — Contains static files that can be deployed to the host machines by this role.

**Meta** — Defines metadata for this role. Basically, it contains files that establish role dependencies.

**templates** — Similar to files templates are used to copy files to host machines but with dynamic values. This uses jinja2 format templating.

Tomcat Installation Role: [adhig93/tomcat-role \(github.com\)](https://github.com/adhig93/tomcat-role)

Docker Installation Role: [adhig93/docker-install-ubuntu-role \(github.com\)](https://github.com/adhig93/docker-install-ubuntu-role)

## **Dynamic Inventory:**

With scalable cloud infrastructure it is difficult to maintain a static inventory file as we might miss those instances that are newly created or created by auto-scaling. Ansible provides a way to capture these new host machines with Dynamic Inventory concept using plugins

Ansible dynamic inventory plugin is supported from Ansible 2.7 version.

### **Pre-requisites:**

1. Install boto3 and botocore

```
sudo apt-get install -y python3-botocore
sudo apt-get install -y python3-boto3
```

2. Install AWS CLI and configure with access\_Key and secret\_Key

### **aws\_ec2 Plugin:**

```
aws_ec2.yaml

plugin: aws_ec2

regions:
  - ap-south-1

keyed_groups:
  - key: tags
    prefix: tag
  - key: instance_type
    prefix: instance_type
  - key: placement.region
    prefix: aws_region
```

### **Configuration:**

```
In ansible.cfg file

enable_plugins = aws_ec2
inventory = /etc/ansible/inventory/aws_ec2.yaml
```

### **Commands:**

```
ansible-inventory -graph
ansible <tag> -m ping
```