**Ansible**

Help Link:

Installation Guide - <https://docs.ansible.com/ansible/latest/installation_guide/index.html>

Tutorilas : <https://www.guru99.com/ansible-tutorial.html>   
Trainner : <https://docs.google.com/document/d/1Z4yLDjMOk6ZYDg3oN3pAfvh243SgIPUEE3bLSTFLhVc/edit>

Problam: Supposes you have 50 system/server/node/slave and you want to perform some opration like cleaup/scaing/software installtion/any others task. The we have two option

* **Do it one by one** => it is very lenghly task and it will take time/effort/cost efective
* **Automation(**like **Ansible, pupet, sas)**: you can atomate it by some tol which hrlp you to install on every system by single command in very less time by identify the group/ip of system.

**Ansible(**Configuration management tool**)** is an open source automation and orchestration tool for software provisioning, configuration management, and software deployment. Ansible can easily run and configure Unix-like systems as well as Windows systems to provide infrastructure as code.

It contains its own declarative programming language for system configuration and management. It’s worked over SSH(**Secure Shell**).

It is Agnet less Tool and no need to install the agent/additional software.

**PlayBook**: Ansible writen is yaml language which human readble and which help to atomate configuraction manage toll and deploye/install the software with parller execution(same time we can run n number of system) in less to less time.

**Why use Ansible?**

Here are some important pros/benefits of using Ansible

* One of the most significant advantages of Ansible is that it is free to use by everyone.
* It does not need any special system administrator skills to install and use Ansible, and the official documentation is very comprehensive.
* Its modularity regarding plugins, modules, inventories, and playbooks make Ansible the perfect companion to orchestrate large environments.
* Ansible is very lightweight and consistent, and no constraints regarding the operating system or underlying hardware are present.
* It is also very secure due to its agentless capabilities and due to the use of OpenSSH security features.
* Another advantage that encourages the adoption of Ansible is its smooth learning curve determined by the comprehensive documentation and easy to learn structure and configuration.



**Benefits of Ansible**

* **Free**: Ansible is an open-source tool.
* **Very simple to set up and use**: No special [coding](https://www.simplilearn.com/tutorials/programming-tutorial/coding-for-beginners) 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.

**Important terms used in Ansible**

* **Ansible server:**

The machine where Ansible is installed and from which all tasks and playbooks will be ran

* **Module:**

Basically, a module(like yum or apt) is a command or set of similar Ansible commands meant to be executed on the client-side. It help us to install software / automation task on server through yum/apt module. Ansible connected the nodes and spread out the Ansible modules programs. Ansible executes the modules and removed after finished. These modules can reside on any machine; no database or servers are required here. You can work with the chose text editor or a terminal or version control system to keep track of the changes in the content.

-apt:

name: apache2

state: present

* **Task:**

A task is a section that consists of a single procedure to be completed

* **Role:**

A way of organizing tasks and related files to be later called in a playbook

* **Fact:**

Information fetched from the client system from the global variables with the gather-facts operation

* **Inventory:**

File containing data about the ansible client servers. Defined in later examples as hosts file. Inventory is lists of nodes or hosts having their IP addresses, databases, servers, etc. which are need to be managed.

* Playbooks

Playbooks consist of your written code, and they are written in YAML format, which describes the tasks and executes through the Ansible. Also, you can launch the tasks synchronously and asynchronously with playbooks

* **Play:**

Execution of a playbook

* **Handler:**

Task which is called only if a notifier is present

* **Notifier:**

Section attributed to a task which calls a handler if the output is changed

* **Tag:**

Name set to a task which can be used later on to issue just that specific task or group of tasks.

API's

**The Ansible API's works as the transport for the public or private cloud services.**

CMDB

CMDB is a type of repository which acts as a data warehouse for the IT installations.

**Who uses Ansible Automation Platform?**

Automation teams need to quickly provide reliable automation where and when the business needs it. In many organizations these roles may not be dedicated to a person or team, some operations team members may serve in multiple roles.

* **Automation architects** elevate automation across teams to align with IT processes and streamline adoption. IT managers and architects can more easily expand automation across the enterprise, while managing automation policy and governance with automation services catalog and getting real-time reporting across the entire stack with Red Hat Insights for Ansible Automation Platform.
* **Automation developers** create Ansible playbooks, roles, and modules. Developers retain the freedom to build, without the operational overhead of maintaining many tools and frameworks. Execution environments deliver a consistent experience for building and scaling automation, with new tooling included to help build and manage them. There are over 100 certified Ansible Content Collections that offer pre-built automation content, with solutions available for nearly every use case.
* **Automation admins and operations teams** ensure the automation platform and framework are operational. Administrators and operators have powerful tools in the automation controller and automation hub to manage and share automation projects more efficiently, with a common language and broadly accessible mix of command line interfaces (CLIs), graphical user interfaces (GUIs), and text-based user interfaces (TUI) across endpoints



Ansible Installation in Linux

When you have compared and weighed your options and decided to go for Ansible. Then installed it on your system. Let's go step by step of the installation in different Linux distributions, such as:

Prerequisites

* PyYAML: a YAML parser and emitter for the python programming language.
* Httplib2: a comprehensive HTTP client library.
* parmiko: native python SSHv2 protocol library.
* Distro: RHEL/ CentOS/ Debian/ Ubuntu Linux.
* Jinja2: a modern and designer friendly templating language for python.
* sshpass: a non-interactive ssh password authentication.
* **Note: It is mandatory to insall python/SSH on linux system and WinRM/Python on windows system to porforme the any automation through ansible**

**Installation Process:** <https://docs.ansible.com/ansible/latest/installation_guide/index.html>

**Class: 1 => Installation Process(Ansible, ssh) & Password less connectivity**

* Create AWS EC2(with ubuntu OS & t2.micro and name as **AnsibleMaster** Node) Instace whre we can install Ansible Node/Ansible Master
* Run the below command on your master node
* Sudo apt update –y
* sudo apt install software-properties-common
* sudo apt update –y
* sudo add-apt-repository --yes --update ppa:ansible/ansible
* sudo apt install ansible
* Or You can aslo install it through .sh file(create one file with .sh extention and put all these command in single file to get it run)
* Create file

touch install.sh #For create the file

sudo apt update -y

sudo apt install software-properties-common -y

sudo add-apt-repository --yes --update ppa:ansible/ansible

sudo apt install ansible –y

bash install.sh #For run & install the above software

ansible –version #For get the Ansible version

sudo apt list –installed

* Create Another Ec2 ubunto instance with name of **AnsibleManagedNode1**
* Create Public & private Key through ssh keygen as normal user for Passwordless connectivity
* Generate Key on both server
* **ssh-keygen**
* After Run above command some parheps will ask(afeer that enter ---)
* Your identification has been saved in id\_rsa
* Your public key has been saved in id\_rsa.pub
* Get Private & Public Key from Bothe server
* **cat /home/ubuntu/.ssh/id\_rsa.pub** #For Pulic Key
* **cat /home/ubuntu/.ssh/id\_rsa** #For Private Key
* Copy The Public key from Master/Controller Node
* **cat /home/ubuntu/.ssh/id\_rsa.pub**
* Then Go to .ssh authorized directory of slave/manage node and add your master public key
* **vim /home/ubuntu/.ssh/ authorized\_keys** #Pest/Add you public key
* Back to Your Master/Controller Node
* cd
* cd .ssh/
* ubuntu@ip-172-31-33-243 #userName@ip-IPAddress

**Class:2 => How Ansible will connect with manage node & what permission required**

* ControllerNode(Master): Where Ansible is install
* Manages Node(Slave): Where we need to perform automation through Ansible
* Practice
* Login to your manage node and add some user
* sudo adduser tom
* sudo adduser rom
* sudo add user shyam
* Switch to diff user
* su - tom
* whoami
* try to install some software through newly created user
* apt install apache2 –y
* Make super user to other user
* 1st login with default/super user
* sudo su
* pwd
* whoami
* cd /etc/sudoers.d/
* ls
* cat 90-cloud-init-users
* visudo
* And Below the root user and add other user which you want to make as su user
* tom ALL=(ALL:ALL) NOPASSWD: ALL
* rom ALL=(ALL:ALL) ALL
* su – tom
* sudo apt install apache2
* **Inventory**: Used to manage IP address of Managed node for Ansible machine. Ansile read/get the IP address from inventory file and execute/perform the automation
* 1st change the Master/controller/ansible Node name
* **sudo hostnamectl set-hostname controller-node**
* Ansible inventory file available in ansible home directory: **cd /etc/ansible/hosts** (where we provide slave/manage/remote machine details on which we need to perform automation**)**
* cd /etc/ansible/
* sudo vim /etc/ansible/hosts
* echo "" | sudo tee /etc/ansible/hosts #To clear/Delete existing details
* sudo vim /etc/ansible/hosts
* And Add your private Or Public IP(salve, manage, child) where you want to perform the automation
* Copy, Add you privat /public iD and Press ESC and :wq
* #For get the list of available host in your inventory file
* ansible all --list-hosts
* cat hosts
* Group/Category(like we have list web server, app server, db server, mail server) in Inventory file:
* Open your host file below like like

#Single or Invidious IP

172.46.07.34

172.46.87.84

#For Grouping like webserver group IP

[ webserver ]

172.46.07.34

172.46.07.34

172.46.07.34

#For Grouping like appserver group IP

172.31.33.243

172.31.33.244

[webser]

172.31.33.243

172.31.33.242

[dbserver]

172.31.33.243

172.31.33.246

ii. Get this of host/group

cat hosts or cat /etc/ansible/hosts

iii. Get list of IP associated with IP

ansible webser --list-host

iv.

* Performed the automation on on host/remote/slave server
* ansible all --list-host
* ansible webser --list-host
* If want to make super/parent host group with the combination of child hostgroup
* 1st create your normal host group

[webser]

172.31.33.243

172.31.33.242

[dbserver]

172.31.33.243

172.31.33.246

* Make parent like below(host name will be nay for child & parrent)

[parent: children] #Create parent group/host

webser #child group/host name

dbserver #child group/host name

* ansible parent --list-host #For get the list of prent IP address

#For single or Indivisoul IP

172.31.33.243

172.31.33.244

#For grouping the IP

[webser]

172.31.33.243

172.31.33.242

#For grouping the IP

[dbserver]

172.31.33.243

172.31.33.246

#For grouping the IP like parent & child

[parent:children]

webser

dbserver