Ansible

Host Inverntory :   
Inventory defines a group of hosts which are alike in any way.

Default path : /etc/ansible/hosts

Graphical user interface

Description automatically generated

Ansible modules:Graphical user interface, text, application, email

Description automatically generated

Modules are those executable plugins that get the real Job done. Usually modules can take the key value arguments and run in a customized way depending on the arguments themselves. So a module can be invoked by a command line or can be included in the ansible playbook.

Commands related to ansible modules:

-If you want to use the modules from command line you need to type in the command.

ansible all -m ping

-If you want to use the ping module on all the hosts defined in the inventory ,

ansible webservers -m command -a “ls”

This will basically list all the modules present and then it will ping.

-If you want to flush Ip table rules on all the hosts in the inventory then type command

ansible -i inventory all -m command -a “iptables -F” –become --ask-become-pass

this will tell ansible to execute the command with sudo previleges.

-If you want ansible to gather facts about the hosts then

ansible all -m setup

-Also if you want to extract particular facts in the documentation of the setup module

ansible -m doc setup

these are few commands that you can use with ansible modules.

Let’s checkout about YAML files.

Ansible uses YAML syntac for expressing Ansible playbooks.

Better to read,write and understand when compared with xml or json.

Every yaml file optionally starts with “---” and ends with “…”

Text

Description automatically generated

Graphical user interface, application

Description automatically generated

You can also use abbreviations in YAML to represent dictionaries.

It basically follows the pattern of a key value pair.

Here the Key value is “james” and it has various values like name, rollno, div, sex. So the key “james” has many values.

Graphical user interface, application

Description automatically generated

We can also represent the list in YAML file. So, every element of the list should be written in a new file with the same indentation starting with a “-” and a space.

As example you can see after writing countries, we have written AMrica with a – and a space and the next line we wrote china with “-” and a space and so on.

If you want to represent the list in an abbreviation form it can be represented in the form of

Countries: [‘America’, ’China’, ’Canada’, ’Iceland’]

Further talking about YAML files you can also represent the list of dictionaries. So, YAML uses the pipeline to include new lines while showing multiple lines and uses the > symbol to suppress new lines while showing multiple lines. So, due to this we can read and large lines and in both the cases indentations are ignored.

Graphical user interface, text

Description automatically generated

So that’s about Yaml files and talking about playbooks, let me tell you how to write a playbook. But before that let’s what exactly ansible playbook is?

As told before playbooks are the files where ansible code is written and these are written in the YAML format. YAML stands for Yet another Markup Language and playbooks are one of the code features of ansible and tell ansible what to execute. So, they like ‘to-do’ list for ansible that contains the list of tasks and playbooks contains the step which the user wants to execute on a particular machine. So, playbooks run sequentially, and playbooks are the building blocks for all the use cases of ansible. So, each playbook is an aggregation of one or more plays in it and playbooks are structured using the plays. So, there can be more than a play inside a playbook and the function of the plays is to map a set of instructions that defined against a particular host.

Since YAML is a strict type of language, we need to take extra care while writing the YAML files.

Now let me tell you how we can create a playbook. So, As I said before YAML starts with “---” and then we have different YAML tags. Let’s go to each YAML tag, so the different tags that are included are name, hosts, vars and tasks.

Graphical user interface, text, application, email

Description automatically generated

So the name tags specifies the name of the ansible work. Its basically as in what this playbook will be doing or any logical name that can be given to this playbook.

Coming to the hosts tag, it specifies the list of hosts or the host group against which we want to run the tasks. The host field tag is mandatory, and it tells ansible on which hosts to run the list of tasks. The tasks can run on the same machine or on a remote machine or one can run the tasks on multiple machines and hence the hosts tasks can have a group of hosts entry as well.

Coming to VARS, the vars tag let you define the variables that you can use in your playbook and usage is very similar to the variables in any programming language.

Final tag – TASKS, All the playbooks should contain a task or the list of tasks to be executed. So, the tasks are the list of actions one needs to perform. So, tasks field contains the name of task and this works as the help text for the user. Well it’s not mandatory, that proves useful in debugging the playbook. So, each task internally links to a code called module.

Hands-on part where we create a playbook and working of Ansible.  
  
  
First we have to create SSH connection. To create it with the node type:

ssh-keygen

once key has been generated we have to generate a host inventory file. So, we creating a host inventory file so that we mention the specifications of the Node, Ip address that we wanna connect with.

sudo nano /etc/ansible/hosts

[test-servers]

Knode

Once we have added hosts in inventory file, we have to mention the ssh generated key in the knode. That is the node we have just configured.

ssh-copy-id -i knode

once the key is generated, you must generate a playbook. Before I start creating a playbook let’s just test if out test servers are running or not.

ansible -m ping test-servers

we can see the o/p as successful. Let’s start creating a playbook, for it we need to write a YAML file.

sudo nano ansibledemo.yml

Text

Description automatically generated

Once you have saved your playbook, you have to run your playbook.

ansible-playbook <name of yml file>

ansible-playbook ansibledemo.yml

this is how playbook runs.

Once the playbook runs, it automatically installs nginx container i.e basically the task that we mentioned. Now to check whether it is running on the other node or not. You have to switch to your other node

ps waux | grep nginx