Ansible (PYTHON) for automation to the provision of the target environment and to then deploy the application.
It helps you with configuration management, application deployment, task automation, and also IT orchestration.

It can run tasks in a sequence and create a chain of events happening on different servers or devices.

Ansible works by connecting to nodes and sending small programs called modules to be excuted remotely on the nodes.

You can write codes that describe the installation and deployment of you applications/ services -- write once for installation and deploy multiple times.
-Ensures automation, reduce work
-Ensures consistency of the systems in the

infrastructure
-Automatic deployment/ scaling up

Main actors in Ansible : (-* we have on our machines)
1. Controller Machine: Machine where Ansible is installed
Inventory: Information regarding servers to be managed (-*)
Playbook: Automation is defined using tasks defined in YAML format
4. Task: Procedure to be executed
5. Module: Predefined commands executed directly on remote hosts
6. Play: Execution of a playbook
7. Role: a Pre-defined way for organizing playbooks (*)
8. Handlers: Tasks with unique names that will only be executed if notified by another task(*)

9. *(^) *Meta: neither a plugin or module. Influence Ansible internal execution or state
clear facts
clear host_errors
end host
end play
flush_handlers
noop
refresh_inventory
reset_connection

**Within a play, ALL hosts are going to get the SAME TASK DIRECTIVES. **

///////sUMMARY/////////
Playbooks contain plays which contain tasks. Tasks call modules & run sequentially & trigger handlers (runs once at

the end of plays)

In playbook Yml:
●write all the tasks you want the client nodes to do
●keep the value of the hosts the same as what you have defined in the inventory file
●indentation is very important. the first dash refers to the highest stage of the playbook which is the name of
the playbook etc sample book
●run ansible-playbook sample.yml --syntax-check
●when u run ansible-playbook, ok means its already there, changed means its not there but you have deployed to make
the change

YAML in ansible : https: //docs .

ansible.com/ansible/latest/
reference_appendices/YAMLSyntax. html

Tasks List:
Each play contains a list of tasks.
Tasks are executed in order, one at a time, against all machines matched by the host pattern, before moving on to the next task.

When running the playbook, which runs top to bottom, hosts with failed tasks are taken out of the rotation for the entire playbook. If things fail, simply correct the playbook file and rerun.

The goal of each task is to execute a module, with very specific arguments.

Ansible task options (tags) & conditional statements (when:)

******Ansible by default transfer files to the remote machine via the use of stfp by default*****

Refer to this if you want to disable it as it may not be required for all program: https://serverfault. com/questions/582171/is-the-ssh-sftp-subsystem-required- on-the-managed-nodes-for-ansible-to -work

Creating reusable playbook:
1. Ansible Role:
- An independent component of playbook which has to be used within playbook
- A Pre-defined way for organizing playbooks
- Always executed first before any other play tasks

- Allows the reuse of common configuration steps

(^)- meta/mains.yml -> consist of all the roles & parameters & all the dependencies (ROLE DEPENDENCIES EXECUTED BEFORE THE ROLE THAT INCLUDES THEM)

To execute the same role multiple times, define each of the instance with different parameters or add allow_duplicates: true to the roles/name of the repeated role/meta/main.yml file for the role.

Roles are defined using YAML files with a Predefined directory structure -> Use ansible-galaxy init <ROLE NAME> to generate a new role in your working directory, with the main directories and

yaml files created within the new rolle
->use tree <ROLE NAME> to list out the directories.

Essentially a GROUP of variables *, tasks*, templates *, files and handlers* stored in a standardized
file structuree

Setting your roles: https://
linuxacademy.com/blog/linux academy/
ansible -roles-explained/

>>You can create your pre-defined variables in the defaults and variables main.yml> their values will be used in the tasks" main.yml

2. Import (static, ansible preprocess these during Playbook parsing time)
PARENT TASK options will be copied to all

child tasks contained within the import
>cannot use variables from inventory sources

3. Include (dynamic, processed during runtime, at the point in which that task is encountered)
Apply to the dynamic task as it is evaluated, NOT COPIED to child tasks. It is good when you use it with looping, buttttt
Tags which only exist inside a dynamic include will not show up in --list-tags output.
Tasks which only exist inside a dynamic include will not show up in --list-tasks output.
You cannot use notify to trigger a handler name which comes from inside a dynamic include
You cannot use - -start-at-task to begin

execution at a task inside a dynamic include.