Chapter 02

DEPLOYING ANSIBLE

→ Inventory

→ An inventory defines a collection of hosts that Ansible will manage.

→ Hosts can be assigned to groups.

→ Groups can contain child groups.

→ Hosts can be manage member of multiple groups.

→ Inventory can also set variables that apply to the hosts and groups that it defines.

Host inventory

- Static host inventory
  - Text file
- Dynamic host inventory
  - Generated by script or program

Static host inventory can be written in INI-style format or YAML.

```
node1.com
node2.com          |→ ungroup manged hosts
node3.com

[group1]
node4.com          |→ grouped hosts
192.142.82.10
[group2]
node6.com
[group3:children]
group1             |→ Nested group
group2
```

host specification with ranges:

1) 192.168.4.[0: 255]
   IP from 192.168.4.0 to 192.168.4.255

2) server[01:05]
   servers from server01 to server05

3) server [a:c]
   servers from servera to serverc.

$ ansible node1 --list-hosts
→ checks for 'node1' in inventory

$ ansible group1 --list-hosts
→ checks for hosts in group 'group1' in inventory

$ ansible all --list-hosts
→ checks and shows all hosts in inventory.

If 'host' & 'Group' have same name, then ansible will ignore the group.

Defaut systems host file = /etc/ansible/hosts

To specify inventory made somewhere else, we can use;
→ -i path_to_inventory
→ --inventory path_to_inventory
Options in 'ansible' & 'ansible-playbook' command

$ ansible ungrouped --list-hosts
→ list upgrouped hosts in the inventory.

Ansible Configuration

Behaviour of ansible installation can be customized by modifying settings in the ansible configuration file.

1) /etc/ansible/ansible.cfg
→ Base configuration file.
→ Chooses this when no other config file is found.

2) ~/.ansible.cfg
→ present in users home directory
→ It is being used instead of /etc/ansible/ansib.cfg

3) ./ansible.cfg
→ Ansible.cfg exists where ansible command is run
→ it is being used instead of /etc/ansible/ansible.cfg & ~/.ansible.cfg.

Priority

./ansible.cfg
↓
~/.ansible.cfg
↓
/etc/ansible/ansible.cfg

We can also specify environmental variable ANSIBLE_CONFIG along with its location to we if from anywhere. This overrides all the above three file locations.

```
$ ansible --version
```

→ Shows ansible version & config file location that it will follow.

$ ansible servers --list-hosts

→ shows hosts & ansible configuration file that it will use

The Ansible configuration file consists of several sections. Each section contains settings defined as key-value pairs.

For basic operation two sections are used:

1) [defaults]
sets defaults for ansible operation.

2) [privilege_escalation].
performs privilege actions on managed node.

```
[defaults]                          → default permission / config.
inventory = /inventory              → inventory path
remote_user = user                  → user doing tasks
ask_pass = false                    → Prompt of SSH authentication
[privilege_escalation]              → for privilege action
 become = true                      → Weather to become or not
 become_user = root                 → what to become (default = root)
 become_method = sudo               → How to switch user
 become_ask_pass = false.           → weather to ask pass or not
```

Ansible uses 'smart' protocol by default, which determines the most efficient way to use SSH.

To change the protocol manually,

→ create host_vars subdirectory in which ansible is run.

→ create a file named localhost

→ vim "ansible_connection: smart" line in it

ex. vim group_vars/windows
ansible_connection: winrm
ansible_port: 5986

Configuring ansible connection requirement

- Location of inventory.
- Which protocol to use for connection (SSH by default).
- Which remote user
- weather to give the user sudo permission or not.
- Weather or not prompt for SSH password.

To comment ansible configuration file content

→ The hash or number sign (at start of line)

; → The semicolon (at start of line)

AD HOC commands

→ An ad hoc command is a way of executing a single ansible task quickly. They can be written without using playbook.

Syntax:

```
ansible host-pattern -m module -a arg -i inventory
```

ping module of ansible does not do ICMP ping.
It only sees if we can run python-based modules on managed Hosts

Modules are the tools that ad hoc commands use to accomplish tasks.

```
$ ansible-doc -l
```

→ lists all modules installed on a system

- Adhoc → idempotent → changed : true → changed : false

$ Running abitrary commands.

$ ansible host -m command -a "Command of
shell linux."
raw

→ 1st line of output = Status report / rc=0 ⇒ success
= Name of managed node / rc≠0 ⇒ fail
= Outcome of operation

2nd line of output = Output of the command executed

$ ansible host -m command -a "linux command" -o
→ shows both two lines output in one single line.

If no option is given after '-m' then adhoc command uses 'command' module by default

For 'shell' abitrary command, shell environmental variables are accessible & shell operations such as redirection & piping are also available for use

'Command' & 'Shell' required python installation on the managed node.
'Raw' does not require python installation & they used in configuring network routers, etc.

The directives & privilege escalations can be configured or used in Ad-hoc command by using some options:

- inventory = -i
- remote_user = -u
- become = -b, --become
- become_method = --become_method
- become_user = --become_user
- become_ask_pass = --ask-become-pass, K

wget -O file_1 link

→ specifies the output file name where the downloaded content will be saved.