­­Inventory file can be in INI or YML format.

It will contain System IP, System FQDN, System SSH Port, System SSH Pass and System Username.

Default inventory file of Ansible is in /etc/ansible/host. We can create our own inventory file, but we have provides this file name while running the command with “-I <file name>”.

In /etc/ansible/ansible.cfg file, make sure “host\_key\_checking=False” should be uncommented.

Ansible playbook is a yml file.

YAML stands for Yet Another Markup Language.

Ansible playbooks are a book of play. Play is a task.

To run a playbook use below command:

ansible-playbook <playbook name> -I <inventory file name>

sshpass

https://www.golinuxcloud.com/getting-started-with-ansible/#Overview\_on\_Ansible

Ansible is automation tool.

Ansible use SSH for Linux and winro for Windows machine.

**Ansible Inventory:**

Admin store all server IP/dns, username, passwords and port, we called that file as Inventory file. Inventory file can be in .ini or .yml format.

Default inventory file for ansible exist in /etc/ansible/hosts.

Ansible use default username as **root**, connection as **ssh**.

Make sure in **/etc/ansible/ansible.cfg** flie, **host\_key\_checking** is set to **false** and not commented.

Make sure, we have install SSHPASS in our linux controller system.

**We can create our own inventory file with below details:**

<Alias Name> ansible\_host=<host ip> ansible\_ssh\_pass=<password> ansible\_connection=<ssh/winro> ansible\_port=<port number> ansible\_user=<username>

We can have N number of server details separated by new line

**If we want to run custom inventory file, use below command:**

ansible <All/Any Alias Name> -m <module name> -i <custom inventory file name>

**WildCard Alias name in execute ansible script:**

ansible < Alias\_Name\_Begining\_Name\*> -m <module name> -i <custom inventory file name>

**Create Group in inventory file as below, we can have N no of Alias name in a group:**

[group\_name]

<Alias name1>

<Alias name2>

<Alias name3>

ansible < group\_name> -m <module name> -i <custom inventory file name>

**WildCard Alias Name in Inventory file:**

[group\_name]

<Alias\_name\_begining>[1:5]

pankaj[1:5]

**Group of Group in Inventory File, we can have N no of group\_names:**

[group\_of\_group\_name:children]

<group\_name>

<group\_name>

ansible < group\_of\_group\_name > -m <module name> -i <custom inventory file name>

**Ansible Playbook:**

Ansible Playbook is Ansible Orchestrator Language.

Playbook is YML file. Playbook is collection of Play. Play is set of task. There may be one or more Play in a Playbook. And there may be one or more task in a Play. We define instruction/actions (when need to be completed by ansible on target machine) in playbook.

**Execute Playbook**

ansible-playbook <playbook name> -i <custom\_inventory\_file\_name>

**Playbook with one play and one task:-**

-

name: <Play Name>/<Test Play>

hosts: <Alias/Group/Group\_of\_Group Name>

tasks:

-

name : “<task name>/<create a file>”

command: touch /tmp/ansible\_temp.txt

**Playbook with one play and 2 tasks:-**

-

name: <Play Name>/<Test Play>

hosts: <Alias/Group/Group\_of\_Group Name>

tasks:

-

name : “<task name>/<create a file>”

command: touch /tmp/ansible\_temp.txt

-

name : “<task name>/<create a Directory>”

command: mkdir /tmp/ansible\_directory

**Playbook with multiple play and multiple tasks:-**

-

name: <Play Name1>/<Test Play1>

hosts: <Alias/Group/Group\_of\_Group Name>/webservers1

tasks:

-

name : “<task name>/<create a file on webserver1>”

command: touch /tmp/ansible\_temp.txt

-

name : “<task name>/<create a Directory on webserver1>”

command: mkdir /tmp/ansible\_directory

-

name: <Play Name2>/<Test Play2>

hosts: <Alias/Group/Group\_of\_Group Name>/webservers2

tasks:

-

name : “<task name>/<create a file on webserver2>”

command: touch /tmp/temp.txt

-

name : “<task name>/<Copy File from one folder to another folder>”

command: cp <source\_folder> <dest\_folder> (/tmp/temp.txt /etc/my\_folder/)

**Ansible Module:-**

**Copy Module PlayBook with specific owner, group and owner.**

-

name: This is copy module playbook example

hosts: <group/server name>

tasks:

-

name: “copy a file from src to dest with owner, group and custom permission

copy: src=<file name> dest=<destination path of target machine> owner=<file owner user name> group=<owner user group name> mode=<permission of that file>

copy: src=pankaj.cnf dest=/tmp/ owner=gupta group=gupta mode=0644

copy:

src: pankaj.cnf

dest: /tmp/

owner: gupta

group: gupta

mode: 0644

In above script, owner, group, mode are optional.

We have 2 approaches to write a module as explained in above example. We can write everything in one single line or in next line also.

**LineinFile Module:**

This module inserts line in file if that line is not present in that file. This module is idempotent.

-

name: Line in File Module Example Playbook

hosts: webserver1

tasks:

-

name: “insert line if not present in file”

lineinfile: path=/tmp/text.txt line=”pankaj”

**Script Module:**

This module helps us to run script file on target machine, which file exists on Controller machine.

tasks:

-

name : “Run .sh file on target machine”

script: <file name with full path> <argument 1>… <argument N>

**Ansible variables:**

Variables in ansible are used same as Jinja Template format.

-

name: Declare variable and store value

hosts: webserver1

vars:

servicename: apache2

tasks:

-

name: “start service which is mention in variable”

service: name={{ apache2 }} state=started

**Ansible Conditions:**

-

name: Ansible Condition example playbook. We can use (==, >, >, >=, <=, and, or) operators.

hosts: webserver1

vars:

age: 18

tasks:

-

name: “execute below command, only when condition is right”

command: touch /tmp/18.txt

when: age == 18

**Ansible Loops:**

**Install multiple package on target machine using ansible loop**

tasks:

-

name: “Install multiple package using loop”

apt: name=”{{ item }}” state=present

with\_items:

* vftpd
* tree
* httpd

**Same above example but with variables**

hosts: webserver1

vars:

pkg:

* vftpd
* tree
* httpd

tasks:

-

name: “Install all packages inside pkg list”

apt: name=”{{ item }}” state=present

with\_items: “{{ pkg }}”

**Ansible Includes:**

**Import variable files into main playbook**

hosts: webserver1

vars\_files:

* <variable file 1.yml>
* <variable file 2.yml>
* <variable file N.yml>

tasks:

-

name: “task 1”

command: touch /tmp/{{ var1 }}.txt

**import tasks files into main playbook**

hosts: webserver1

vars\_files:

* <variable file1.yml>

tasks:

* Include: <task file1.yml>
* Include: <task fileN.yml>

**Ansible Roles:-**

In Ansible, the role is the primary mechanism for breaking a playbook into multiple files.

Each role is basically limited to a particular functionality or desired output, with all the necessary steps to provide that result either within that role itself or in other roles listed as dependencies.

Roles are not playbooks. Roles are small functionality which can be independently used but have to be used within playbooks. There is no way to directly execute a role. Roles have no explicit setting for which host the role will apply to.

By default, all roles are created inside /etc/ansible/roles folder.

**Create role**

ansible-galaxy init <role name>

**Import role in main playbook**

name: Import role in main playbook example

hosts: webserver1

roles:

* webserver # This is role name, which was created using above command.

In above playbook, webserver role with its variable and tasks will be imported automatically. We can include any number of roles.