Roles

Preparing windows server

Ansible galaxy

Asynchronous action

Error handling

Jinja2 templating

Lookups

Vaults

Dynamic inventory

Custom module

Plugin

Contents

[YAML Basic: 2](#_Toc106883931)

[Introduction to Ansible & lab Setup 2](#_Toc106883932)

[Documentation: https://docs.ansible.com/ansible/latest/ 2](#_Toc106883933)

[Ansible inventory files 2](#_Toc106883934)

[Ansible Playbook 5](#_Toc106883935)

[Ansible Modules & Parameters 7](#_Toc106883936)

[1. Copy Module: 7](#_Toc106883937)

[ansible-playbook copy.yml -i ../inventory\_group.txt 7](#_Toc106883938)

[2. Line in file Module 7](#_Toc106883939)

[3. Command Module 7](#_Toc106883940)

[4. Script Module 7](#_Toc106883941)

[5. Service Module 8](#_Toc106883942)

[6. User module 8](#_Toc106883943)

[Ansible Variables 9](#_Toc106883944)

[Scope of Variable 9](#_Toc106883945)

[Variable Type: 9](#_Toc106883946)

[a. List Variable : 9](#_Toc106883947)

[b. Dictionary Variable : 9](#_Toc106883948)

[c. global Variable : 10](#_Toc106883949)

[d. runtime variable: 10](#_Toc106883950)

[Ansible Conditions 10](#_Toc106883951)

[Ansible Loops 11](#_Toc106883952)

# YAML Basic:

1. All yml file start with “---“ and end with “…”
2. Item could be list of key/value pair or group of list and key/value
3. All member of list is start with same identation as “-“
4. All member of key/Value is “key: value” (the colon must be followed by a space)
5. For Comment use “ #”

Example :

---

*# Employee records*

**-** **martin:**

**name:** Martin D'vloper

**job:** Developer

**skills:**

**-** python

**-** perl

**-** pascal

**-** **tabitha:**

**name:** Tabitha Bitumen

**job:** Developer

**skills:**

**-** lisp

**-** fortran

**-** erlang

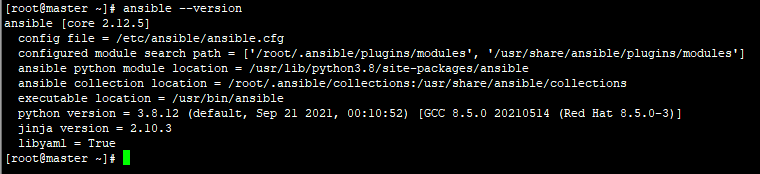
…

# Introduction to Ansible & lab Setup

## Documentation: <https://docs.ansible.com/ansible/latest/>

**Installation**: yum install ansible -y

**Validation:** ansible –version



# Ansible inventory files

1. Ansible Default Inventory File path : /etc/Ansible/hosts
2. Add sample inventory entry in hosts file :

10.121.27.49 ansible\_ssh\_pass=redhat

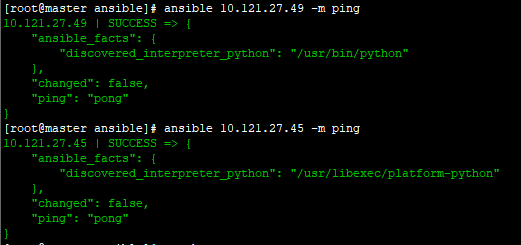
10.121.27.45 ansible\_ssh\_pass=redhat

Syntax.

IP ansible\_ssh\_pass=<password of server>

1. Ping the system using Ansible (For connecting SSHPASS Package should be installed)

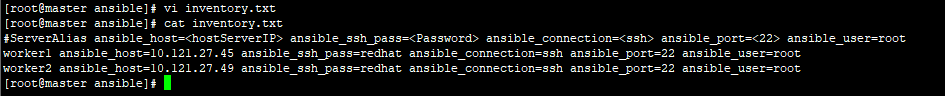
ansible 10.121.27.49 –m ping ### -m means module

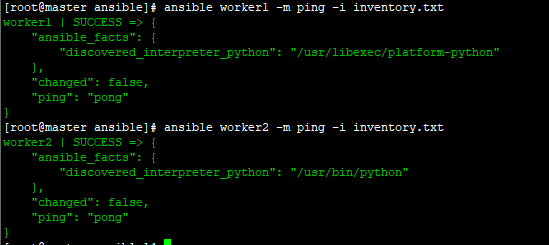


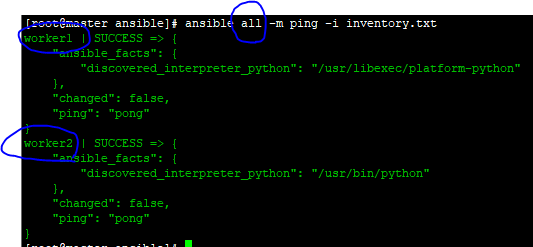
**Create custom Inventory Files Separately and get the ping response from different inventory file:**

Vi inventory

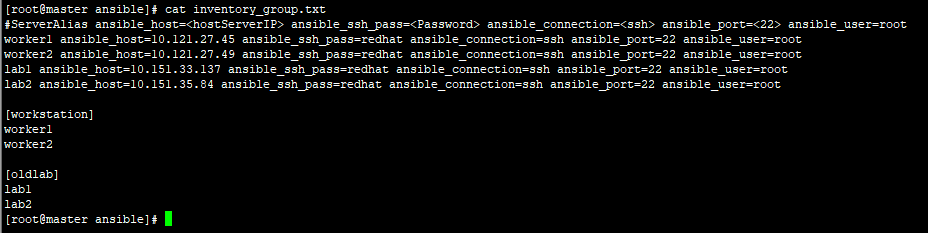
ServerAlias ansible\_host=<hostServerIP> ansible\_ssh\_pass=<Password> ansible\_connection=<ssh> ansible\_port=<22> ansible\_user=root



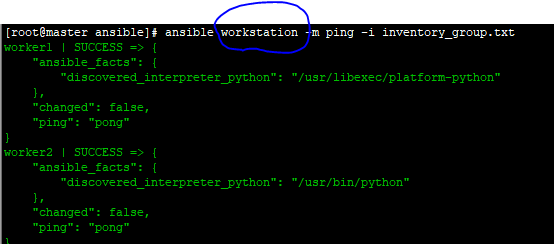




**Create Inventory Group**



**Ping using inventory group**

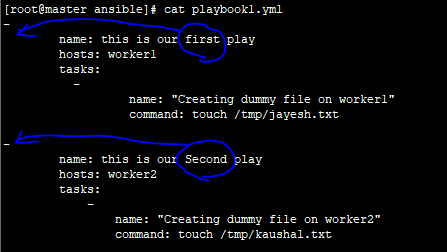
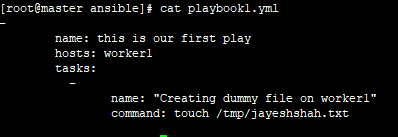


# Ansible Playbook

1. Ansible is orchestration language where we are creating playbook and in playbook we are defining the task what we are going to execute on target machine
2. Playbook is yml file
3. Set of task is called “play” which we are defining in yml or playbook

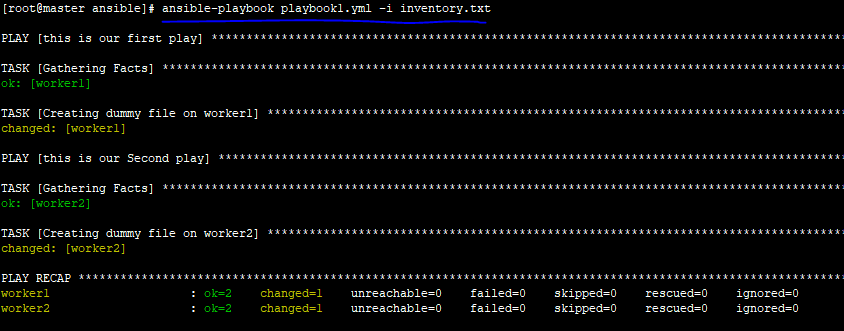
Sample playbook:

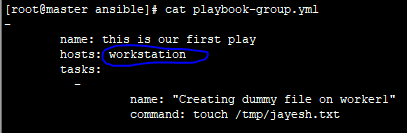
1. Single play & multiple play

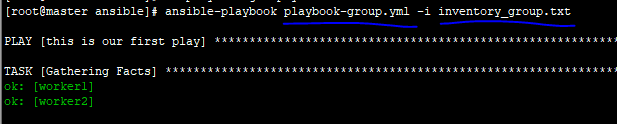


How to run play book

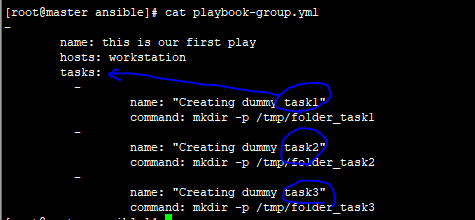
Syntax: ansible-playbook <playbookname> -i <inventory-name>







Multiple task in single play



-

name: this is our first play

hosts: workstation

tasks:

-

name: "Creating dummy task1"

command: mkdir -p /tmp/folder\_task1

-

name: "Creating dummy task2"

command: mkdir -p /tmp/folder\_task2

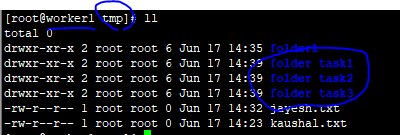
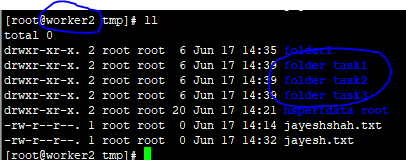
-

name: "Creating dummy task3"

command: mkdir -p /tmp/folder\_task3

Output on worker1 & Worker2 or a workstation group

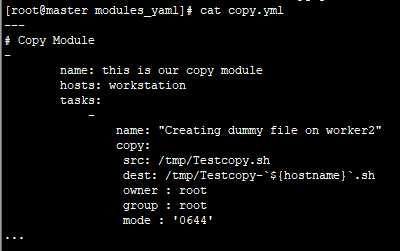
================ ===========================



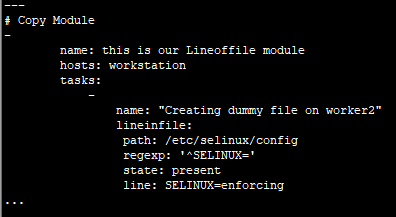
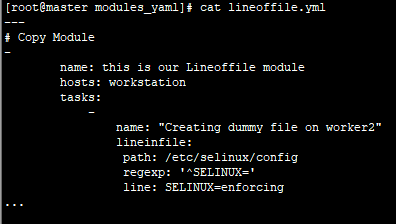
# Ansible Modules & Parameters

1. Copy Module:

ansible-playbook copy.yml -i ../inventory\_group.txt



1. Line in file Module



<https://docs.ansible.com/ansible/latest/collections/ansible/builtin/lineinfile_module.html>

Some of the Important Attribute:

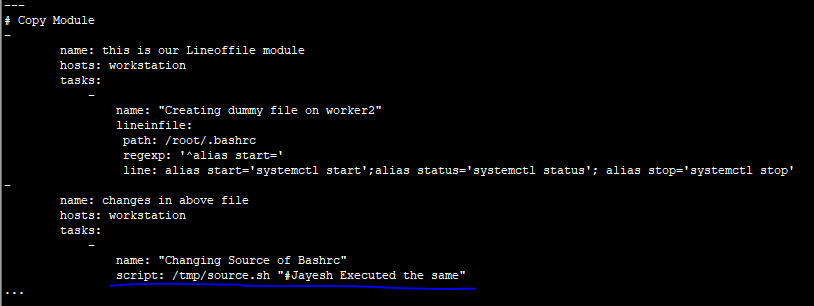
Regexp: checking given syntax line by line

Line: insert line in to a remote file

State: **present** (the pattern to replace if found) /**absent** (the pattern of the line(s) to remove)

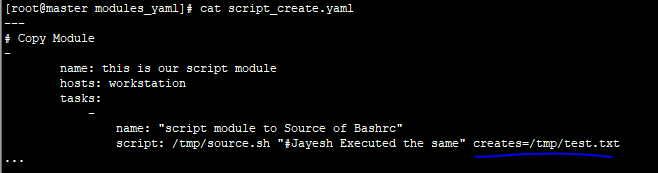
**C**reate**:** If specified, the file will be created if it does not already exist

1. Command Module
2. Script Module

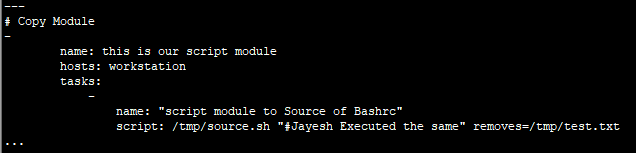


Some of the Important Attribute

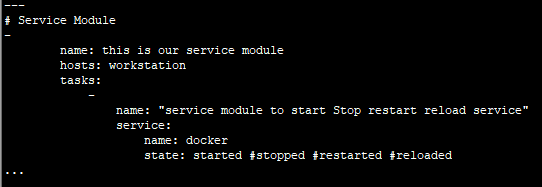
Creates: if it is present it will skip execution of the script



Removes: if the file present with removes tag than only script will execute



1. Service Module

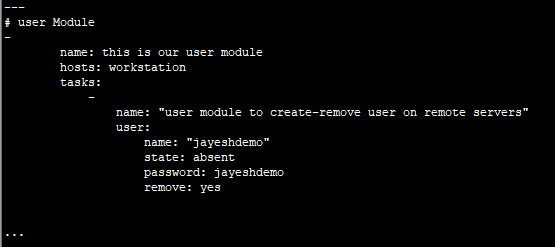
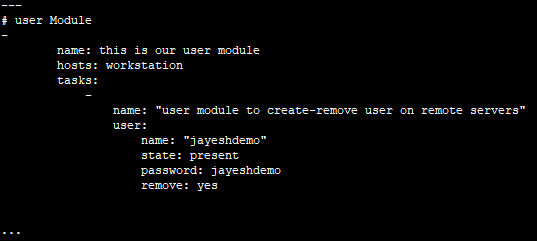


Some of the Important Attribute

State:  idempotent actions that will not run commands unless necessary.

Enabled: Whether the service should start on boot.

1. User module



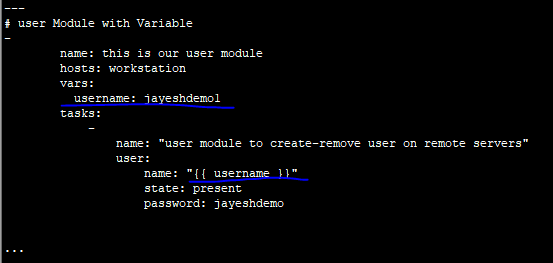
Some of the Important Attribute

State: absent/present if absent ---it will remove the user if present it will allow to create

Remove: remove tag will be used while removing user along with working directory too

Ansible Variables

You can define a simple variable using standard YAML syntax, after you define a variable, use Jinja2 syntax to reference it. Jinja2 variables use double curly braces. you must quote the whole expression to create valid YAML syntax



Scope of Variable

1. Global

* this is set by config, environment variables and the command line

1. Play

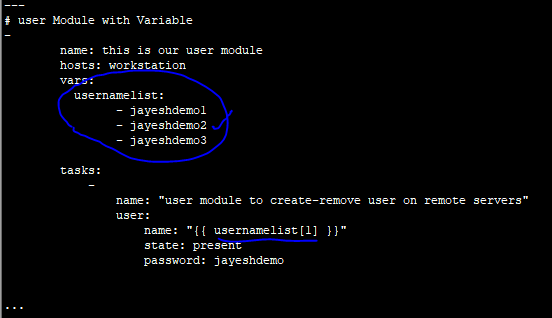
* each play and contained structures, vars entries

1. host

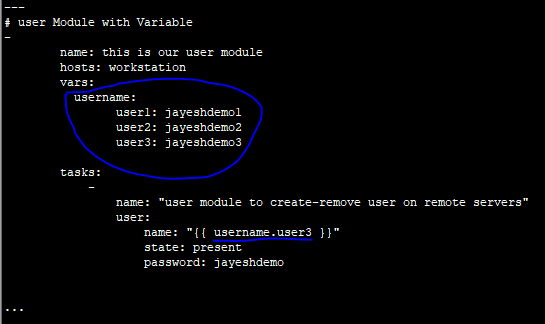
* variables directly associated to a host, like inventory, include\_vars

## Variable Type:

1. List Variable :



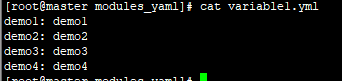
1. Dictionary Variable :



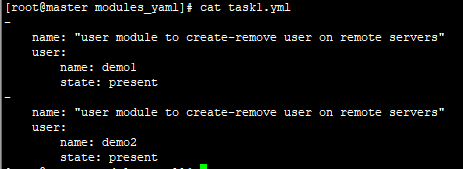
1. global Variable :

you can create separate yml file for storing variables and task and use that file in actual playbook

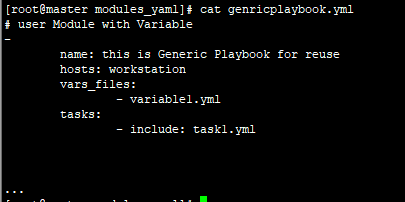
variable1.yml



task1.yml

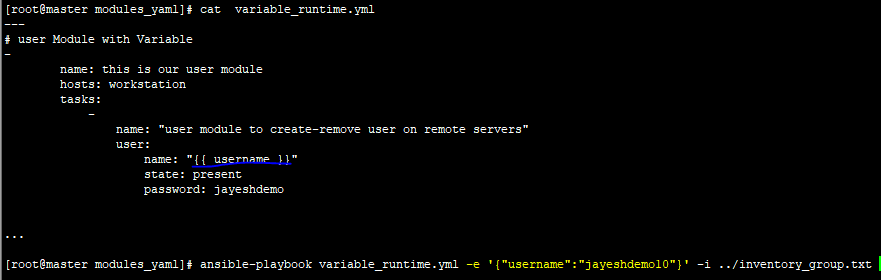


genricplaybook.yml

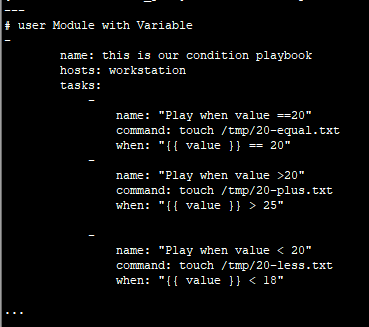
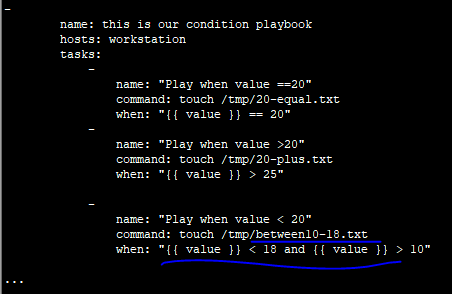


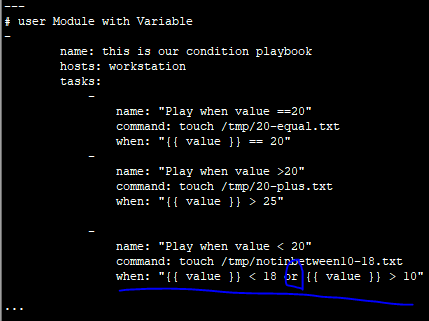
1. runtime variable:

You can include variable on runtime while playing playbook using –e or –extra-vars



# Ansible Conditions



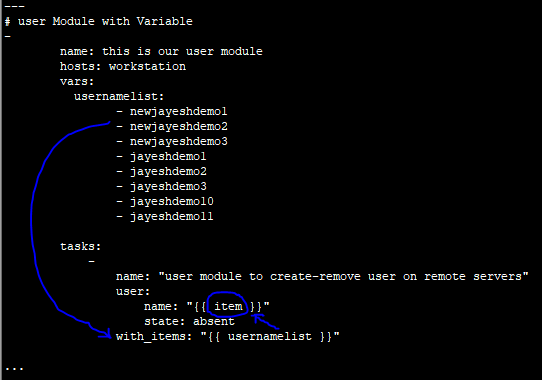
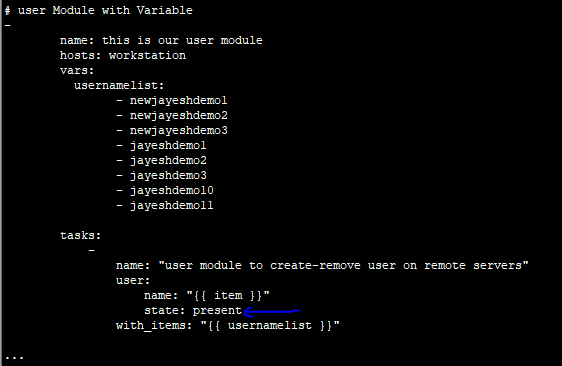
ansible-playbook condition\_or.yml -e '{"value":"8"}' -i ../inventory\_group.txt

ansible-playbook condition\_or.yml -e '{"value":"15"}' -i ../inventory\_group.txt

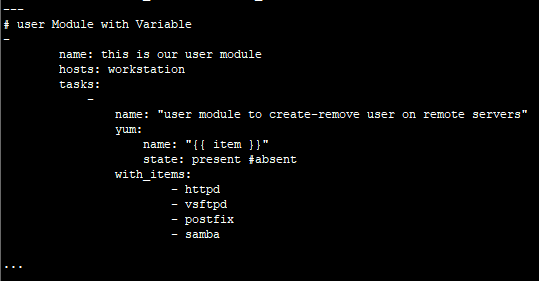
ansible-playbook condition\_or.yml -e '{"value":"25"}' -i ../inventory\_group.txt

# Ansible Loops

Loop is used to execute same task multiple time with different dynamic variables defined in list



Removed and installed multiple services using yum and Loop to remote hosts

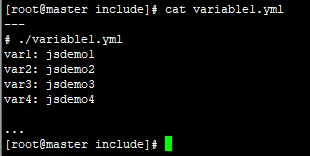


State= absent will uninstall the application and present will install the same.

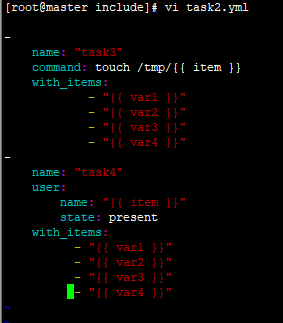
# Include Files or Modularization in Ansible

Using Include we can create separate files for variables and task separately and manage the playbook effectively.

Step 1: Create Variable File:



Step2: Create Task files :



Step3: Include Variables in play or Task:

1. The [include\_vars](http://docs.ansible.com/ansible/latest/modules/include_vars_module.html" \l "include-vars-module) module (works in any list of tasks, such as a playbook or role
2. The [vars\_files](http://docs.ansible.com/ansible/latest/reference_appendices/playbooks_keywords.html" \l "play) keyword (works on plays only)

