

Ansible

PAGE:

DATE:

1 / 1

What is ansible?

→ Ansible is an open source IT configuration management, deployment & orchestration tool. It aims to provide large productivity gains to a wide variety of automation challenges. This tool is very simple to use yet powerful enough to automate complex multi-tier IT application environments.

① How to install ansible in linux VM [RHEL]

→ Sudo yum install ansible -y

② How to install ansible in ubuntu M/C?

→ Sudo apt-get install ansible -y

[if its not working run sudo apt-get update]

③ Ansible playbooks are written in Yaml language

④ How to check ansible version?

→ ansible --version.

④ What are ansible roles?

Ansible role is a set of tasks to configure a host to serve a certain purpose like configuring a service.

Roles are defined using YAML files. With a predefined directory structure. A role directory structure contains. directories : defaults, vars, tasks, files, templates, meta handlers.

For ex: The directory will look like this:

 | ansible.cfg

 |- hosts

 |- roles

 |- mongodb

 |- defaults

 |- → main.yml

 |- files

 |- handlers

 |- → main.yml

 |- meta

 |- → main.yml

 |- tasks

 |- → main.yml

 |- templates

 |- tests

 |- → main.yml

 |- vars

 |- → main

Scanned By Scanner Go

→ It is need to comment the ansible line.

You can create a role using
→ ansible-galaxy init command inside /etc /
ansible / roles.

These directories are tasks, handlers,
defaults, vars, files, templates, and meta
and a README.md file.

Tasks - contains the main list of tasks that
are to be executed by the role. It contains
the main.yml file for that particular role.

Handlers - contains handlers which may be
used by this role or even anywhere outside
this role.

~~Vars~~:

Defaults: contains the default variables
that are going to be used by this role.

Vars: This directory consists of other variables
that are going to be used by the role.

These variables can be defined in your
playbook, but it's a good habit to define
them in this section.

Files - contains files that can be deployed
by this role. It contains files that need to

be sent to the hosts while configuring the role.

meta - defines metadata for this role.

Basically, it contains files that establish role dependencies.

Every task directory must contain of main.yml file where the actual code for that particular role is written.

Sudo ansible-galaxy init <role-name>

Ex - Sudo ansible-galaxy init mongodb.

" " nodejs

" " java

" " apache-ant

⑤ How to run ansible play book?

→ ansible-playbook -vvv playbook.name.yml.

Ex: ansible-playbook -vvv assignment1.yml.

vvv - to check error clearly.

⑥ What are ansible play books?

→ An ansible play book is a blueprint of automation tasks - which are complex IT actions executed with limited ⑦ no human involvement. Ansible play books are executed on a set, group, ⑧ classification of hosts, which together make up an ansible inventory.

for Ex:

- hosts: localhost
tasks:

- name: set the given azure subscription.

Shell :

⑦ what is ansible inventory?

The ansible inventory file defines the hosts and groups of hosts upon which commands, modules, and tasks, in a playbook operate. The file can be in one of many formats, depending on your ansible environment and plugins. The inventory file can list individual hosts or user-defined groups of hosts.

List of ansible modules.

Linux:

find, copy, shell, lineinfile, win-lineinfile, Service, state, delegate-to, register, when, retries, delay, with-items, fail, debug,

windows: win-find, win-copy, win-shell.

① copy module:

used to copy files and folders from the local machine to the remote servers.

Ex①

- hosts: localhost
tasks:

- name: copy the task
copy :

src: /home/ubuntu/Sample2/test4.txt
dest: /home/ubuntu/Sample3/

Ex②

- hosts: localhost
tasks:

- name: copy the task
copy :

src: /home/ubuntu/Sample5/test1.txt
dest: /home/ubuntu/Sample6/

② shell module:

Shell module takes the command name followed by the list of space-delimited arguments.

part 1 → Copy using shell module

a)

- hosts : local host

tasks:

- name: Shell module example.

Shell : "cp /home/ubuntu/example1.txt
/home/ "

become: yes

become-method: sudo

become-user: root.

b)

- hosts : local host

tasks:

- name: Shell module example.

Shell : "cp /home/ubuntu/example2.txt
/home/ "

become: yes

become-method: sudo

become-user: root.

② part-2 : child directory:

② - - -

- hosts: local host

tasks:

- name: Shell module example.

Shell: "cp example2.txt /home/ubuntu/
hanumant/example2.txt"

args:

chdir: "/home/ubuntu/"

⑥. - - -

- hosts: local host

tasks:

- name: Shell module example.

Shell: "cp example3.txt /home/ubuntu/
hanumant/example3.txt"

args:

chdir: "/home/ubuntu/"

part-3 - move using shell module.

④ ---

- hosts: localhost

tasks:

- name: move using shell module

shell: "mv /home/ubuntu/test6.txt
/home/ubuntu/hanumant/"

⑤

- hosts: local host

tasks:

- name: move using shell module

shell: "mv /home/ubuntu/test7.txt
/home/ubuntu/hanumant/"

⑥

Part-4 : remove using shell module.

⑦

- hosts: local host

tasks:

- name: remove the file/dir using shell
module.

shell: "rm -rf /home/ubuntu/example6"

⑥ - - -

- hosts: localhost

tasks:

- name: remove the file/dir using shell module

Shell: "rm -rf /home/ubuntu/example7"

⑦ - - -

- hosts: localhost

tasks:

- name: remove the file using shell module

Shell: "rm /home/ubuntu/examples.txt"

part-6 - change permission using shell module.

⑧ - - -

- hosts: local host

tasks:

- name: change permission of a file

Shell: "chmod 997 /home/ubuntu/hanuman/test7.txt"

⑨ - - -

- hosts: local host

tasks:

- name: change permission of a file.

shell: "chmod 455 /home/ubuntu/hancuman/test7.txt"

part-6 - Display the content of a file using Shell module.

⑥ ---

- hosts: localhost
tasks:

- name: display the content of a file
shell: "cat /home/ubuntu/samples.txt"

⑥ ---

- hosts: localhost
tasks:

- name: display the content of a file.
shell: "cat /home/ubuntu/Sample1.txt"

part-7

③ Register module :-

A ~~possible~~ register is a way to capture the output from tasks execution and store it in a variable.

@ ---

- hosts: local host
tasks:

- name: register module example
stat:

path: /home/ubuntu/examples

register: examples5-folder

- name: register -2

shell: "cat /home/ubuntu/examples/test 4.txt"

when: examples5-folder.stat.exists == true

- name: register -3

file :

path: "/home/ubuntu/examples"

stat: directory

when: examples5-folder.stat.exists == false.

When = true \rightarrow register -2 run.

\rightarrow register -3 is skipped.

When = false \rightarrow register -2 is skipped.

\rightarrow register -3 run.

(b) - --

- hosts: localhost
tasks:

- name: register module example.
stat:

paths: /home/ubuntu/register
register: register_folder.

- name: register-2
shell: "rm -rf /home/ubuntu/register"
when: register_folder.stat.exists == true

- name: register-3
file:

paths: "/home/ubuntu/register"
state: directory
when: register_folder.stat.exists == false

When = true → register-2 run.

→ register-3 is skipped.

When = false → register-2 is skipped.

→ register-3 run.

④ When condition module -

The simplest conditional statement applies to a single task. Create the task, then add a when statement that applies a test. When you run the task @ playbook. Ansible evaluates test for all hosts, on any host the test passes - the ansible runs task.

⑤ Ignore error module -

In ansible if anyone of the task fails then it will stop the entire execution of playbook @ role.

So to avoid this problem we use ignore-errors = true.

If you mention ignore-errors = true at the end of the task, if the task fails also still the playbook @ role will run.

⑥ copy /home/ubuntu/Sample4 /tests5.txt /home/ubuntu. don't create tests5.txt file. add ignore-errors : true module. Then ignored this task.

⑥ next task : rm -rf /home/ubuntu/Sample4 [remove sample4 directory].

→ --

- hosts: localhost
tasks:

- Name: copy the task
Shell: "cp /home/ubuntu/Sample4/
tests5.txt /home/ubuntu/

ignore_errors: true

- name: remove task

Shell: "rm -rf /home/ubuntu/Sample1"

17. ---

- hosts: localhost

tasks:

- name: Copy the task

Shell: "cp /home/ubuntu/Samples1/
tests.txt /home/ubuntu/1"

ignore_errors: true

- name: remove task

Shell: "rm -rf /home/ubuntu/Sample1"

⑥. find module

used when you need to retrieve a list of files in the remote server. which matches some conditions like name, size, etc.

Ex- it Find the zip file:

- hosts: localhost

tasks:

- name: check if zip file available.

find :

paths: "/home/ubuntu/exampleg/"

patterns: "* .zip"

register: zip_file

- debug : var=zip_file

- name: register1

shell : "cp /home/ubuntu/exampleg/*.zip /home/ubuntu/hanuman"

when : zip_file.matched|int != 1

- name: register2

file :

path: "/home/ubuntu/example11/"

state: directory

when: zip_file.matched|int == 1

⑦. file module -

used to deal with the files, directories and symlinks.

Create one directory using file module.

Ex: ① ---

- hosts: localhost

tasks:

- name: file module example,

file:

path: "/home/ubuntu/sample6/test4.txt"

state: directory

become: yes

- name: with -item example

file :

dest: "/home/ubuntu/Sample6"

mode: 0777

state: directory.

assignment11.yml

② change ownership & permission (444) to
to directories using file module.

Ex-2

- hosts: localhost

tasks:

- name: file module example

file :

path: "/home/ubuntu/Scanned By Scanner Go"

state: directory

- name: example1

file :

dest: "/home/ubuntu/sample10"

mode: 0444

state: directory

- name: with-items example2

file :

dest: "{{ item }}"

owner: root

group: root

become: yes

with-items:

- "/home/ubuntu/sample10"

- "/home/ubuntu/sample11"

assignment12.yaml

Shell module examples

- Q. print all lines start with d using grep command → create one file /home/ubuntu/sample7/test5.txt.

- hosts: localhost
 tasks:

- name: using shell module
 shell : "grep '^d' /home/ubuntu/sample7/test5.txt"
 become: yes
 become_method: sudo
 become_user: root.

- assignment13a.yml

- Q. Search all files which ends with 'f'

- hosts: localhost
 tasks:

- name: using shell module
 shell : "grep 'fs' /home/ubuntu/sample7/test5.txt"

- assignment13b.yml

- Q. print last 10 lines [using tail command]

- assignment13c.yml

- hosts: localhost
 tasks:

- name: shell module example

shell : "tail -10 /home/ubuntu/sample7/test5.txt"

④ print 5,6,7 line using head & tail command.

hosts: localhost

tasks:

- name: Shell module example

shell: "head -7 /home/ubuntu/sample7/test5.txt | tail -3"
- assignment13d.yml.

⑤. print 6th line using sed command.

- hosts: local host

tasks:

- name: Shell module example.

shell: "sed -n '6p' /home/ubuntu/
sample7/test5.txt"
- assignment13e.yml.

⑥ delete 10 line using sed command.

- hosts: local host

tasks:

- name: Shell module example

shell: "sed '10d' /home/ubuntu/
sample7/test5.txt"

- assignment13f.yml

⑦ Replace from to (string1/string2) .

from 7^{th} to end of file using sed cmd.

- hosts: localhost

tasks:

- name: shell module example

shell: "sed '7,\$\$'!JavaLang/hamant/
g' /home/ubuntu/sample7/test5.txt"

-assignment13g.yml

⑧ find the file which modified 90 days back

- hosts: localhost

tasks:

- name: ~~find~~ shell module example.

shell: "find . /home/ubuntu/ -type
f -mtime +90"

-assignment13h.yml

⑨ find the file which modified within 45 days

- hosts: localhost

tasks:

- name: shell module example.

shell: "find . /home/ubuntu/ -type
f -mtime -45"

-assignment13i.yml

⑯. find the file which modified within 5 min.

- hosts: localhost

tasks:

- name: shell module example.

shell: "find . /home/ubuntu -type f
-mmin -5"

→ assignment¹³j.yml

⑰ find the file which modified 20 min back.

- hosts: localhost

tasks:

- name: shell module example.

shell: "find . /home/ubuntu -type f
-mmin +20"

→ assignment13k.yml

⑲. find empty directories in /home/ubuntu dir

- hosts: localhost

tasks:

- name: shell module example

shell: "find . /home/ubuntu -type d
-empty"

→ assignment13l.yml

(13). print 2nd | 3rd column using awk cmd.
[create one separate file].

- hosts: localhost

tasks:

- name: ~~awk command~~ Shell module example
shell: "awk '{print \$2,\$3}' /home/ubuntu/
/Sample9/test5.txt"

- assignment13m.yml.

(14) Search multiple pattern [String1/String2] using grep

- hosts: localhost

tasks:

- name: Shell module example.
shell: "grep -e 'Hello' -e 'today' /home/
ubuntu/Sample9/test4.txt"

- assignment13n.yml.

(15). count no. of lines, words, char. using a shell.

- hosts: localhost

tasks:

- name: Shell module example.

shell: "wc -l /home/ubuntu/Sample9/
test5.txt"

Shell: "wc -w /home/ubuntu/sample9/test5.txt"
- count no. of words.

Shell: "wc -c /home/ubuntu/sample9/test5.txt"
- count no. of characteristics.

8) State module:

state as 'present' and 'installed' are used interchangeably. They both do the same thing, i.e. it will ensure that a desired package in your case 'yum' is installed.

where state as 'Latest' means in addition to installation, it will go ahead and update if it is not of the latest available.

State module examples ---> state [present, absent, started, stopped, restarted, touch, directory]

state: present → check the file.

state: directory → create dir

state: absent → remove, delete,

state: touch → create empty file.

State module works on - both file & dir

⑤ Line in file module -

used to insert a line, modify, remove, and replace an existing line.

①

- hosts: local host
tasks:

- name: lineinfile module

lineinfile:

dest: "/home/ubuntu/Samples/tests.txt"

regexp: '^Karthik ='

line: 'Karthik=localhost'

- name: example2

lineinfile:

dest: "/home/ubuntu/Samples/tests.txt"

regexp: '^Akshatha='

line: 'akshatha=1000'

- name: example3

lineinfile:

dest: "/home/ubuntu/Samples/tests.txt"

regexp: '^hanuman'

line: 'hanuman=2000'

- assignment 17.xml

⑩ Service module

Service module controls services on remote hosts and is useful for these common tasks: Start, Stop & restart

- - -

- hosts: localhost
tasks:

- name: ensure rghcl Service is started
Service:

name: rghcl

state: started

become: true

- name: ensure rghcl Service is stopped
Service:

name: rghcl

state: stopped

become: true

- name: ensure rghcl Service is restarted.
Service:

name: rghcl

state: restarted

become: true

-assignment15.yml

⑪ retries module: [retries, delay, until]

- Retry task 10 times with interval 1sec until return code of the command will not be 0. Ignore if even all tries will fail.

Ex①

- hosts : localhost

tasks :

- name : start wildfly

Service:

name : wildfly

state : restarted

become : true

register : task_result

until : task_result is succeeded

retries : 5

delay : 120

⑫ delegate_to module:

delegate_to is a directive, not an individual module, it integrates with other modules and it controls the task execution by deciding which host should run the task at runtime.

Ex②

- hosts : localhost

tasks :

- name : state absent

file :

Scanned By Scanner Go
path : "/home/fabwutu/Sample10"

~~path~~:

state: directory
delegate_to: "165.10.28.38"

- assignment16.yml.

D) Fail module:

- Fail module fails the progress with a custom message.
- Intentionally fails the build.

Ex:-

- hosts: localhost

tasks:

- name: fail module example.

fail :

msg: fail the build

when: example_folder.stat.exists == false

Include-tasks: within role run another tasks

import-tasks:

set-fact: set value in the yml file.

+HumanReadable=1

include-~~task~~ role: include role only

1) What are ansible vaults & why are they used?

— Ansible vault is a feature that allows you to keep all your secrets safe. It can encrypt entire files, entire YAML playbooks or even a few variables. It provides a facility where you can not only encrypt sensitive data but also integrate them into your play books.

Vault is implemented with file-level granularity where the files are either entirely encrypted or entirely unencrypted.

It uses the same password for encrypting as well as for decrypting files which makes using ansible vault very user-friendly.

② What is ansible tower?

— Ansible tower is ansible at a more enterprise level. It is a web-based solution for managing your organization with a very easy user interface that provides a dashboard with all of the state summaries of all the hosts, allows quick deployments, and monitors all configurations.

The tower allows you to share the SSH credentials without exposing them,

logs all the jobs, manage inventories graphically and syncs them with a wide variety of cloud providers.

③ What is ansible Galaxy?

- Galaxy is a website that lets ansible users share their roles and modules.

The galaxy command line tool comes packed with ansible, and it can be used to install roles from galaxy from a source control management sdm such as git. It can also be used to build new roles, remove existing ones and perform tasks on the galaxy website.

You can use the below command to download roles from the Galaxy website:
Ansible-galaxy install username.role-name

2. with-items -

It is used create multiple file by using single task.

Windows module:

- (1) Ansible execution in windows. Please also execute the below command to install winrm before ansible execution.

- * To install EPEL repo on a CentOS and RHEL 7.x:

Sudo yum install epel-release

- * To install python pip on RHEL 7:

Sudo yum install python27-pygments-pip

- * To install pywinrm without RHEL repository:

curl -O

Sudo yum install

python2-winrm-0.3.0-1.el7.noarch.rpm

For windows:

hosts.ini

[windows]

1dkfx8cmi6vpcw1

[windows: vars]

ansible_host = 10.2.230.232

ansible_user = mpadmin

ansible_password = "Vmp4ssword!123"

ansible_port = 5985

ansible_connection = winrm

ansible_winrm_server_cert_validation = ignore

[mpadmin@1dkfx8cmi6vpcw1: testing]\$

ansible windows -i

hosts.ini -m win-ping

1dkfx8cmi6vpcw1 | success => {"changed": false, "ping": "pong"}

Run the ansible playbook in windows m/c
ansible-playbook -vvv test.xml -i hosts.ini.

Windows modules:

① Win_copy:

its used to copy the file/dir from local m/c
to the remote server.

Ex ① ---

- hosts: windows
strategy: linear
tasks:

- name: Copy the file.

win-copy:

src: "F:\\Temenos\\jboss\\standalone\\
\\deployments\\axis2.war"

dest: "F:\\Temenos\\t24home\\
axis2.war"

remote_src: true.

Ex ② ---

- hosts: windows
strategy: linear
tasks:

- name: Copy the file.

win-copy:

src: "F:\\test\\test5.txt"

dest: "F:\test\sample6\test5.txt"
remote_src: true
assignment19.yml

③ Win-unzip module

Unzips compressed files and archives on the windows node.

Ex: ①

- - -

- hosts: windows
strategy: linear
tasks:

- name: unzip the file

win-unzip:

src: "C:\sample\test6.zip"

dest: "C:\sample6\temp"

remote_src: true

assignment20.yml.

Ex ②

- - -

- hosts: windows
strategy: linear
tasks:

- name: unzip the file.

win-unzip:

src: "F:\Temenos\temp\homework\

authenticator.zip"

dest: "F:\temenos\temp"

⑤ win_service module:

It is used to manage and query windows services.

E

Ex① ---

- hosts: windows
- Strategy : linear
- tasks :

- name: Restart distributed transaction co-ordinator Service.

win-service:

Name: MS DTC

State: restarted.

Ex② ---

- hosts: windows
- strategy: linear
- tasks :

- name: Service restart

win-service:

Name: RHEL

State: restarted.

assignment21.yml.

- hosts: windows
- Strategy: linear
- tasks :

- name: Service started.

win-service:

name: UBUNTU

state: started.

assignment21a.yaml

④ ~~win~~ win-shell:

It is used to execute shell commands on target hosts.

Ex ①

- hosts: windows

strategy: linear

tasks:

- name: Delete some folder

win-shell: "Remove-Item f:\temp*\n
\\t24home\test.zip"

ignore_errors: true

② Copy the file from F:\example\test8.txt to
F:\example\sample4\test8.txt using
shell command.

- hosts: windows

strategy: linear

tasks:

- name: Copy the file using shell

win-shell: "Copy-Item f:\example*\n
\\Sample4\test8.txt f:\example\sample4\test8.txt"

⑤ assignment22a.yml

- ⑥ move the files from F:\\example\\test9.txt
to F:\\example\\sample9\\test9.txt

- hosts: windows

strategy: linear

tasks:

- name: move using shell cmd

win-shell: "Move-Item F:\\example\\test9.txt F:\\example\\sample9\\"

assignment22b.yml

- ⑥ Remove the file from F:\\example\\test9.txt.

- hosts: windows

strategy: linear

tasks:

- name: remove the file

win-shell: "Remove-Item F:\\example\\test9.txt"

ignore-errors: true

assignment22c.yml

⑤ Win-file:

It is used to creates, touches & removes files & directories.

EX①

- hosts: windows

strategy: linear

tasks:

- name: Create one dir/file.

win-file:

path : "F:\Tiempos\test\test5.txt"

state: touch

*③. @ Create one dir F:\Example7\test4.

- hosts: windows

strategy: linear

tasks :

- name: Create dir

win-file:

path : "F:\example7\test4"

state: directory.

assignment 23.yml

④ Create one file: F:\example7\tests\test1.txt"

- hosts: windows

strategy: linear

tasks:

- name: Create file,

win-file:

path : "F:\example7\tests\test1.txt"

state: touch

⑥. win-lineinfile:

Ensure a particular line is in a file. ⑦

replace an existing line using a back-referenced regular expression.

- used regular expression.

Ex ① change one line Karthik=2002 .hassan=800.

- hosts: windows

strategy: linear

tasks :

- name: lineinfile module

~~lineinfile~~

win-lineinfile :

dest : "F:\example\sample2\test1.txt"

reg exp: 'Karthik='

line : "Karthik = 2002"

- name: example1

win-lineinfile :

dest : "F:\example\sample2\test1.txt"

reg exp: 'hassan='

line : "hassan=800"

assignment 24.7m).

Ex ② Create ③ add new line.

- hosts: windows

strategy: linear

tasks :

- name: lineinfile module.
win_lineinfile:
dest: "F:\\example\\sample2\\test1.txt"
regexp: 'bangalore ='
line: "bangalore =9987"

assignment 24a.vim1.

⑦ win_start:-

It is used to get information about windows files

Ex(1) ---

- hosts: windows
strategy: linear
tasks:

- name: ~~check~~ check file exists @ not
win_start:

path: "F:\\temenos\\t24homel\\
authenticator.waz"

register: waz-file

- debug: var=waz-file

Ex(2) check the file exist @ not [path: F:\\test\\
test6.txt].

- hosts: windows
strategy: linear
tasks:

- name: check file exists or not.

win-state:

path: "F:\\test\\ test6.txt"

register: war-file

- debug: var=war-file

assignment25.yml

⑧ win-find :-

it is used to find the files.

Return a list of files based on specific criteria

Ex①

- hosts: windows

strategy: linear

tasks:

- name: find files in the folder

win-find:

path: "F:\\temenos\\ t24home"

patterns: "*.zip"

register: zip-file

- debug: var=zip-file.

Ex②. find the multiple zip files. [*.zip].

- hosts: windows

strategy: linear

tasks:

- name: multiple files

Scanned By Scanner Go

win-find:

path: "F:\test\default"

patterns: "*.*.zip"

register: zip-file.

→

-debug: var2 zip-file