1. How to install any tools/services using ansible playbook??

state =>present/absent/latest

install\_pgk1.yaml file.

---

- hosts: samplenode1

become: yes

tasks:

- name: Install git tool

yum:

name: git

state: present

[devopsadmin@ip-172-31-81-101 ansible]$ vi install\_pgk1.yaml

[devopsadmin@ip-172-31-81-101 ansible]$ ansible-playbook install\_pgk1.yaml

PLAY [samplenode1] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

[WARNING]: Platform linux on host samplenode1 is using the discovered Python

interpreter at /usr/bin/python, but future installation of another Python

interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference\_appendices/interpreter\_discovery.html for more information.

ok: [samplenode1]

TASK [Install git tool] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [samplenode1]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

samplenode1 : ok=2 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

[devopsadmin@ip-172-31-81-101 ansible]$

1. How to pass variables in runtime using ansible playbook ??

Vi Install\_pkg3.yaml

ansible-playbook Install\_pkg3.yaml -e "host\_name=samplenode1 tool\_name=git tool\_state=present"

---

- hosts: "{{ host\_name }}"

become: yes

tasks:

- name: Install "{{tool\_name}}"

yum:

name: "{{ tool\_name }}"

state: "{{ tool\_state }}"

[devopsadmin@ip-172-31-81-101 ansible]$ clear

[devopsadmin@ip-172-31-81-101 ansible]$ vi Install\_pkg3.yaml

[devopsadmin@ip-172-31-81-101 ansible]$ cat Install\_pkg3.yaml

---

- hosts: "{{ host\_name }}"

become: yes

tasks:

- name: Install "{{tool\_name}}"

yum:

name: "{{ tool\_name }}"

state: "{{ tool\_state }}"

[devopsadmin@ip-172-31-81-101 ansible]$ ansible-playbook Install\_pkg3.yaml -e "host\_name=samplenode1 tool\_name=git tool\_state=present"

PLAY [samplenode1] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

[WARNING]: Platform linux on host samplenode1 is using the discovered Python

interpreter at /usr/bin/python, but future installation of another Python

interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen

ce\_appendices/interpreter\_discovery.html for more information.

ok: [samplenode1]

TASK [Install "git"] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [samplenode1]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

samplenode1 : ok=2 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

[devopsadmin@ip-172-31-81-101 ansible]$

---

- hosts: samplenode1

**gather\_facts: false**

while installing git no need to gather all information from the target system.

become: yes

tasks:

- name: Install git tool

yum:

name: git

state: present

1. Run the shell commands in target system using ansible playbook??

Vi regis1.yaml

---

- hosts: samplenode1

gather\_facts: false

tasks:

- shell : "bash --version"

register: bash\_ver

- debug: var=bash\_ver

[devopsadmin@ip-172-31-81-101 ansible]$ vi regis1.yaml

[devopsadmin@ip-172-31-81-101 ansible]$ cat regis1.yaml

---

- hosts: samplenode1

gather\_facts: false

tasks:

- shell : "bash --version"

register: bash\_ver

- debug: var=bash\_ver

[devopsadmin@ip-172-31-81-101 ansible]$ ansible-playbook regis1.yaml

PLAY [samplenode1] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [shell] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

[WARNING]: Platform linux on host samplenode1 is using the discovered Python

interpreter at /usr/bin/python, but future installation of another Python

interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen

ce\_appendices/interpreter\_discovery.html for more information.

changed: [samplenode1]

TASK [debug] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [samplenode1] => {

"bash\_ver": {

"ansible\_facts": {

"discovered\_interpreter\_python": "/usr/bin/python"

},

"changed": true,

"cmd": "bash --version",

"delta": "0:00:00.039437",

"end": "2023-04-27 18:57:17.195778",

"failed": false,

"rc": 0,

"start": "2023-04-27 18:57:17.156341",

"stderr": "",

"stderr\_lines": [],

"stdout": "GNU bash, version 4.2.46(2)-release (x86\_64-koji-linux-gnu)\nCopyright (C) 2011 Free Software Foundation, Inc.\nLicense GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>\n\nThis is free software; you are free to change and redistribute it.\nThere is NO WARRANTY, to the extent permitted by law.",

"stdout\_lines": [

"GNU bash, version 4.2.46(2)-release (x86\_64-koji-linux-gnu)",

"Copyright (C) 2011 Free Software Foundation, Inc.",

"License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>",

"",

"This is free software; you are free to change and redistribute it.",

"There is NO WARRANTY, to the extent permitted by law."

],

"warnings": [

"Platform linux on host samplenode1 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See https://docs.ansible.com/ansible/2.9/reference\_appendices/interpreter\_discovery.html for more information."

]

}

}

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

samplenode1 : ok=2 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

[devopsadmin@ip-172-31-81-101 ansible]$

1. If I want to get particular data from that artifacts then how can we do that ?

Vi register\_set\_fact.yaml

---

- hosts: samplenode1

gather\_facts: false

tasks:

- shell : "bash --version"

register: bash\_ver

- set\_fact:

bash\_version: "{{bash\_ver.stdout.split('\n')[0].split()[3]}}"

my\_value: "bash\_version"

- debug: var=bash\_version

1. Handlers in ansible playbook yaml script ?

---

- hosts: samplenode1

#gather\_facts: false

become: yes

tasks:

- name : install httpd

yum:

name: httpd

state: present

- name : start httpd

service:

name: httpd

state: started

Service Module ===> started/stopped/restarted

yum/apt Module ===> state: present/absent/latest

Handlers is special type of task.

Start service is one task.

Handlers also work outside of playbook. When is used as an if condition . when we have multiple tasks inside one playbook we should use loops inside yaml.

LOOPS in ANSIBLE PLAYBOOK ?

---

- hosts: samplenode1

gather\_facts: false

become: yes

tasks:

- yum:

name: "{{item}}"

state: absent

loop:

- git

- httpd

- vim

ANSIBLE ROLES ?

Ansible Roles are used to organize the Ansible Components.

To reuse & Share.

Ansible Repository ::

Understand the structure of Ansible Components ::::

playbooks, handlers, loop