Ansible:

Ansible is it automation tool , written in python., it is agent less software and its playbooks written in simple plain English called yaml

Installation:

#yum install ansible

Python is manadatory.

SSH service will have the Controlepersist ( it is performance feature).

Inventory file:

* Ansible mainly works with inventory file, it is available in /etc/ansible/hosts

We can also specify other inventory file path by giving the commond

-I <path>

* We can also create multiple inventory files.
* We can create dynamic inventory file

#ansible -m ping -i /root/stage/hosts test3

Configuring clients:

Create ssh key and add that keys

#ssh-agent bash

#ssh-add

To ping the listed hosts in the inventory file like below

We can use –I option to use another hosts file

#ansible serverip -I hosts –m ping all

#ansible –m ping all

🡺To copying time when you knownhosts error you need to add below entry in the ansible server config file /etc/ansible/ansible.cfg

host\_key\_checking = False

or

we can set system variable like below

export ANSIBLE\_HOST\_KEY\_CHECKING= False

=🡺 configuring inventory file:

Inventory file is /etc/ansible/hosts

Hosts and groups:-

Siva.example.com

[webservers] siva1.example.com siva2.example.com siva3.example.com

[dbservers] sivadb1.example.com sivadb2.example.com sivadb3.example.com

Above webservers and dbservers are groups.

* If clients are running on different ssh port we have to mention that in the after name of server example below

Siva.example.com:567

* Adding similar pattern servers in to the inventory like below

[webservers]

www[01:50].example.com

[webservers:vars]  
ansible\_user=username   
ansible\_port=3980   
ntp\_server=ntp.example.com  
proxy=proxy.example.com

[dbservers]

Db-[a:f].example.com

Host variables:

Host1 http\_port=80

🡺 List of behavioral inventory parameters:

Hostconnection:

Ansible\_connection:

Here we can describe

Ansible\_user

Ansible\_port

Ansible\_host

**Non –ssh servers with ansible**:

Local : this connector will deploy in the local server.

Docker: it will deploy in to the docker container

**Ansible tower:**

Ansible tower provides the database to store the inventory results

To connect AWS:

To successfully call a api to aws , we need to configure BOTO (python interface to AWS)

Export AWS\_ACCESS\_KEY\_ID=’AK123’

Export AWS\_SECRET\_ACCESS\_KEY=’abc123’

**PATTERN:**

#Ansible –m <module> –a <argument>

#ansible <pattern> -m <module> -a <argument>

Ex: ansible webservers –m service –a “name=httpd state=restarted”

In the inventory file groups are separated by colon/or

**Ad-hoc commands:**

File transfer :

To test connectivity with particular user

#ansible –m ping all –u suer

Same above command we can run with sudo

#ansible –m ping all –u user –b

We are executing a command

#ansible –m command –a “df –h” all and also ansible -m command -a "free -m" all

Shell module:

#ansible –m shell –a “command” all

Copy module:

#ansible group -m copy –a “src=/opt/siva dest=/opt/kesav”

* The file module is responsible changing file permission and ownership.

File module:

#ansible webservers –m file –a “dest=/opt/kesav mode=755”

Managing packages module:

We can use yum and apt modules.

#ansible –m yum –a “name=package state=present

Stae:

Present= it will install

Absent= it will remove

Latest = it will update

**Modules:**

**File modules:**

Acl module:

Ex: - acl:

Path=/path

Entity=username

Etype=user/group

Permissions:rw

State=present

**Ansible playbook:**

It is heart of the ansible.

A play book module divided in to multiple parts those

Hosts: where you want deploy your configuration

Remote\_user:

Tasks: execution modules with specific variables.

Handlers:

Example of a playbook:

-Hosts: webservers

Vars:

http\_port=80

remote\_user:root

tasks:

-name:ensure apache is at latest version

Yum: pkg=httpd state=latest

-name: write apache config file

template: src=/srv/httpd.j2 dest=/etc/httpd.conf

   notify:

   - restart apache

- name: ensure apache is running

   service: name=httpd state=started

handlers:

   - name: restart apache

     service: name=httpd state=restarted

we need create a directory under home directory of a user

playbooks

here we need to write a file

file must end with .yml

* To test syntax of yml file #ansible-playbook myfile.yml –check
* To run playbook

#ansible-playbook myfile.yml

Using variables in playbook:

We can use internal and external variables

Target ?

**Roles:**

Mkdir redhatserver

Cd redhatserver

Mkdir files

Mkdir templates

**Siva written adhoc commands:**

Gathering facts:

#ansible –m

Example playbooks:

Example playbooks:

1. Installing tomcat by using playbook

* Hosts : all

Tasks :

* Name: this is installing tomcat

Yum: pkg=tomcat state=installed update\_cache=true

Handler:

Outline playbook:

:playbook for dry run:

Means –check

:Asynchronous polling:

:interactive playbook: 2.24 mins

Siva written play books:

[root@fiserver3 playbooks]# cat createfile.yml

- hosts: all

tasks:

- name: this is file creation

copy: path=/home/jogi state=touch mode=755

2)

[root@fiserver3 playbooks]# cat copydir.yml

- hosts: all

tasks:

- name: this is copying a direcory

copy: src=/root/centos dest=/root

[root@fiserver3 playbooks]#

1. Installing package

[root@fiserver3 playbooks]# cat installpkg.yml

- hosts: all

tasks:

- name: this is installing logwatch

yum: pkg=logwatch state=installed

[root@fiserver3 playbooks]#

1. Installing multiple packages

[root@fiserver3 playbooks]# cat test1.yml

- hosts: all

tasks:

- name: this is simple test

yum: name={{item}} state=installed

with\_items:

- http\*

- tree\*

5) installing package and starting service.

[root@fiserver3 playbooks]# cat startservice.yml

- hosts: all

tasks:

- name: installing service

yum: name=http\* state=installed

- name: this is service starting

service: name=httpd state=started

[root@fiserver3 playbooks]#

**Modules:**

**-acl: module**

-acl module:modules:

-acl :

acl: path=/root/copydir.yml entity=kesav etype=user permissions=rw state=present

to remove absent

-**copy modules:**

Copying the file if dest file has differ compare to source then take backup.

copy: src=/root/playbooks/copydir.yml dest=/root/ owner=root group=root mode=755 backup=yes

-

-**command module:**

We can run the scripts by using command module.

**- docker** module also we have

-**ec2 module** we can create the instance start and stop we can do it

We should have ec2key and vpc id

-**firewalld:**

We can manage firewall by using firewalld command

Example

- firewalld: Service=http permanent=true state=enabled

**Get\_url:**

By using above module we can download the files

Ex:

Get\_url: url=www.siva.com path=downloadfilepath mode

**Group module:**

Group: name=somegroup state=present

**-jboss:**

We can deploy war file in to the deploy folder

-jboss: src=/path deployment=x.war state=present

-**jenkins\_job:** we can create Jenkins job

**-lvg** module we can create a volume group

-lvg: vg=vgname pvs=pvname(/dev/sda5) pezize=4M

Delete=

-lvg: vg=vgname state=absent

**-lvol**: by using this module we can create logical volume

-lvol: vg=vgname lv=lvname size=size

**Mount**: we can mount the files system

Mount: path=where to mount src=from where ur getting fstype=ext4

Net\_interface: it will make interface active or down

**-Pam\_limits:**

By using this module we can set the limits to a user

Ex: pam\_limits:

**-pause:**

We can pause the playbook execution for certain time

Ex: pause: minutes=5

And also you can promt

Pasue: promt=sys certain thing

-**rpm\_key**:

We can import and remove the rpm key

-rpm\_key: state=present key=key,,,.

**-selinux:**

Selinux: policy=targeted state=enforcing

-service: we can start and stop

-service name=http state=start enabled=yes

**Shell:**

We can use shell module to execute shell commands.

**Systemd**: this module is similar to service module.

Timezone:

-timezone: name=Tokyo

**Yum:**

We can use install packages

Yum: name=http state=latest

**Yum\_repository:**

We can add and remove the repository

yum\_repository: name= epel description=EPEL YUM repo baseurl= <https://download.fedoraproject.org/pub/epel/$releasever/$basearch/>

**wait\_for** : we can use for waiting certain time for this service

-wait\_for : port=80 time=1

Until port 80 listen

Bydefault it will wait 300 sec

**Uri:**

By using uri module we can test the url and also we can create jira

Ansible roles:

It will contains file and tasks

ansible-galaxy init control

ansible-galaxy init nginx

converting ansible playbooks in to roles:

**Include:**

By using include module we can call another playbook inside a playbook.

Ex:

- include: backup.yml

- include: /root/installpkg.yml