**Ansible**

-- Configuration management tool.

-- Agent less configuration management tool.

-- You need not to install any agent on remote servers

-- Totally relies on ssh connections. so just make sure that ssh is available and running.

-- Ansible uses - python (puppet and chef uses ruby)

-- Host file call as inventory file

**Install Ansible in Linux(centos,redhat):**

yum install ansible --enablerepo=epel

**Ansible Installation in Ubuntu:**

1 sudo apt-get update

2 sudo apt-get install software-properties-common

3 sudo apt-add-repository ppa:ansible/ansible

4 sudo apt-get install ansible

5 ansible --version

6 cd /etc/ansible/

7 ls

8 vi hosts

Host file location:

/etc/ansible/hosts

**Add IP details in host file:**

18.220.99.42 ansible\_connection=ssh ansible\_ssh\_user=ec2-user ansible\_ssh\_private\_key\_file=/docker-key.pem

**List of ansible commands:**

ansible ipaddress -m ping -u root -k

ansible ipaddress -m setup -u root -k

ansible ipaddress -m file -a 'path=/etc/fstab'

ansible ipaddress -m file -a 'path=/tmp/hello state=directory mode=0700 owner=root'

ansible ipaddress -m copy -a 'src=/root/hello.txt dest=/tmp'

ansible ipaddress -m file -a 'path=/tmp/install/log state=absent' -vvvv

if host not reachable

chmod 400 \*.pem

add ssh-agent // to start ssh agent

**Password less authentication:**

ssh-keygen

ssh-copy-id -i target m/h

cd .ssh/

ll

cat authorized\_keys

ll /etc/fstab

**Playbook:**

- This executes a series of commands in order, according to a playbook.

**write a playbook to deploy application in apache server**

**Yum install httpd**

---

-

become: true

hosts: all

tasks:

-

name: "install the latest version of Apache"

yum:

name: httpd

state: latest

-

service:

name: httpd

state: started

enabled: yes

-

name: "copy application code to document root"

template: "src=index.html dest=/var/www/html/index.html"

Create a folder

Ansible-playbooks

Vi install-apache.yum

if host not reachable

chmod 400 \*.pem

add ssh-agent // to start ssh agent

**Ansible Playbook: Deploy Application in Apache server**

---

-

hosts: all

tasks:

-

name: install apache2

action: apt pkg=apache2 state=latest

notify: start apache2

-

name: download via maven

local\_action: maven\_artifact group\_id=javahome extension=war artifact\_id=Realtime dest=/tmp/Realtime-0.0.2.war repository\_url=http://172.20.64.99:8085/nexus/content/repositories/releases/0.0.2/

-

unarchive:

src: /tmp/Realtime-0.0.2.war

dest: /var/www/Realtime

-

name: Copy index.html file to destination

copy: "src=/tmp/Realtime-0.0.2.war dest=/var/www/Realtime"

-

name: update index.html to be owned by www-data

action: file path=/var/www/ owner=www-data group=www-data

-

name: start apache2

action: service name=apache2 state=started

**Using Loops:**

- name: Install Packages  
 apt: name={{ item }} state=latest  
 with\_items:  
 - vim  
 - git  
 - curl

**Using Array Variable**

---  
- hosts: all  
 sudo: true  
 vars:  
 packages: [ 'vim', 'git', 'curl' ]  
 tasks:  
 - name: Install Package  
 apt: name={{ item }} state=latest  
 with\_items: packages

**Ansible Roles:**

roles - predefined structure should be used to configure the application

Create a Role:

**ansible-galaxy init apache --offline**

**tree apache**

apache/

├── defaults

│ └── main.yml

├── files

├── handlers

│ └── main.yml

├── meta

│ └── main.yml

├── README.md

├── tasks

│ └── main.yml

├── templates

├── tests

│ ├── inventory

│ └── test.yml

└── vars

└── main.yml

**Components of a Role: (Apache):**

1.defaults = data about the role / application. Default variables stored here

Eg: i want to run webserver on port 8080

2.files = put the static files here.Files which i might be use to push into remote machine.

3.handlers = Tasks which are based on some actions.like triggers.

Eg:In case my httpd.conf config changes,it should trigger service restart.

4.meta =Information about the role. which platform supported,who is author of the role.dependency,if any.

5: tasks = core logic or code. Installing package,coping files etc.

6. Templates = dynamic files similar to files that templates support modifications.

Jinja2- template language.

7. Vars = Both vars and defaults stores variable. Variables stored under “vars” has got higher priority and difficult to override.

**Ansible vault:**

Securing playbooks

To encrypt the playbooks

command:

ansible-vault create topsecret.yml

cat topsecret.yml

vi topsecret.yml

content will be encrypted.

ansible-vault edit topsecret.yml

ansible-vault decrypt topsecret.yml

ansible-vault encrypt topsecret.yml

ansible-playbook topsecret.yml --ask-vault-pass

**Testing and Tags**

-----------------------

ansible-playbook test.yml

**===========**

**--syntax-check**

**===========**

- when we execute a playbook,ansible first check the syntax of the playbook

- If some error is encountered, ansible will point the error

and stop there.. will not move further.

- syntax checking is done only when you use "ansible-playbook" command.

#ansible-playbook nameofplaybook.yml --syntax-check

**============**

**-- check**

**============**

-- Will not apply anything, or any changes,just show what changes will be introduced if you apply the playbook

#ansible-playbook nameofplaybook.yml --check

**Ansible Help:**

-----------------

ansible-doc -l

ansible-doc user

## Allow root to run any commands anywhere

root ALL=(ALL) ALL