**Ansible**

Anisble is a agent less configuration management tool. This is used when we want to work with many servers parallely .

This has only 2 agent anbsible server and client.

We need to install ansible software on server side nothing to be installed on client side.

we need to enable epel on both client and server side

yum install epel-release

Install ansible on server side

yum install ansible

Software will be installed under /etc/ansible with ansible.cfg, roles and host files

Host file is also know as inventory file which contain list of client and client groups.

Open cfg file and enable hostname and sudo user.

Create user on server and client side

Useradd<username> ; User will created under **/etc/passwd**

Password – set password

Give sudo permission for created users

Visudo

Add user and permission under

#Allow user to run all command

<username> ALL=(ALL) NOPASSWD:ALL

Communicate server and client without credential generate ssh key in server and copy public key in client

Ssh-keygen : to generate key

This will create key under current folder with folder .ssh and copy public file in client.

Ssh-copy-id <username>@<IPAddress > : this will copy file to client

Note : If we are not able to copy pub key follow below steps :  
1. PasswordAuthentication yes  
2.PermitRootLogin yes   
3.in the /etc/ssh/sshd\_config   
4.service sshd restart

Example :

Ansible <group/All/IpAddress> -a “mkdir /etc/chethan”

Ansible all -a “ls –lrt /etc/chethan”

ansible all -m ping // to ping server

ansible all -m copy –a “ src=/etc/text.txt dest=/etc/” –s // to copy the file

ansible all –m yum –s –a “name=package state=started/present/absent” // to install package

ansible all –m user -s –a “name=chethu” //to create new user

ansible all –m service –s –a “name=httpd state=started/present/absent” // to start the service of httpd

ansible –doc –l // list the modules in ansible

-a attributes -m : module –s : sudo

rpm –qa <packageName> // used to check if package installed or not

ansible –s –a “rpm –qa <packageName> ” // used to check if package installed or not

**To install package and start service**

---  
- hosts : chethu  
 become : yes  
 tasks :  
 - name : "insatll httpd"  
 yum : name= http state=started  
 - name : "start service"  
 service : name = http state=present

**To install multiple package**

---  
- hosts : batch  
 become : yes  
 tasks :  
 - name : "insatll multiple package"  
 yum : state = started name = {{ item }}  
 with\_items :  
 - git  
 - httpd

$ ansible all  -i  /<path\_of\_inventory\_file>  –list-hosts // list all the servers in inventory

$ ansible group  -i  /<path\_of\_inventory\_file>  –list-hosts // list all the servers in group

**Handlers**:

When ever changes occurred to remote system, the playbook recognize and notify will trigger.

‘notify’ actions are triggered at the end of each block of tasks in a play, and will only be triggered once even if notified by multiple different tasks.

**//once change occur below file notify to restart apache**

- name: template configuration file

template:

src: template.j2

dest: /etc/foo.conf

notify:

- restart apache

**Variables :**

Variable in playbooks are very similar to using variables in any programming language. It helps you to use and assign a value to a variable and use that anywhere in the playbook.

**// We can use tomcat\_port anywhere in playbook**

- hosts : <your hosts>

vars:

tomcat\_port : 8080

**Register variable –**

This is used to hold the output of the task.

tasks:

- name: Ansible register variable basic example

Command : cat /ect/home/file.txt

register: find\_output

**Modules :**

Modules are pre built library. Which used in playbooks using this we can build new module.

**File module :**

Used to create or delete file/directory also to change permission.

$ ansible -m file -a "path=/srv/foo/b.txt mode=600 owner=mdehaan group=mdehaan"

file:

path : /etc/delete.conf

state : present/absent/started

mode : 777

**Copy and Fetch :**

Used to copy from one path to different path in remote machine

copy:

src: /tmp/hello6

dest: /etc

**to copy multiple file**

copy:

src: ~/{{item}}

dest: /tmp

mode: 0774

with\_items:

['hello1','hello2','hello3','sub\_folder/hello4']

**Package Module :** used to install or un install packages

- name: install the latest version of ntpdate

package:

name: ntpdate

state: latest

**lineinfile module :**  used to add data in existing file.

lineinfile:

path: /Users/mdtutorials2/Documents/Ansible/Input.txt

line: 'Added Line 1'

insertbefore: BOF ; to add line before need to assign

insertafter:\*.com ; If the pattern found in file above line will get replaced

**Replace :** used to replace data in file/ we can also do this using lineinfile

replace:

path: /Users/mdtutorials/Documents/Ansible/Input.txt

regexp: 'This line should be'

replace: '#This line should be'

**archive : used to zip the file.**

archive:

path:

- /Users/mdtutorials2/Documents/Ansible/zipfile.txt

- /Users/mdtutorials2/Documents/Ansible/zipfile2.txt

dest: /Users/mdtutorials2/Documents/multi.zip

format: zip

**debug :** used to print data during execution

- name: To display data

vars:

count : “Learing Ansible“

- debug:

Mgs: {{count}}

**When** : is used to check the condition

- name: To display data

vars:

count : “Learing Ansible“

- debug:

Mgs: “Equals”

When : count == “Learing Ansible“

**Vault :** used to hide data in remote machine

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

After 1.7 we can specify password using file

ansible-playbook site.yml --vault-password-file ~/.vault\_pass.txt

ansible-playbook site.yml --vault-password-file ~/.vault\_pass.py

**Service :** used to start or restart the service

service:

name: httpd

state: started/stop/restart

**Pip Module :**

Ansible pip module is used when you need to manage Python libraries on the remote servers.

- hosts: all

tasks:

- name: Installing NumPy python library using Ansible pip module

pip:

name: NumPy