Ansible:

Install a httpd in centos machin through ansible adhoc commands

Inorder to install httpd in centos through ansible adhoc command first we need to understand hoe to install manually.

Step1: sudo yum update

The system will run the software repositories and refresh to the latest versions.

Step 2: Install Apache HTTP

Install Apache on your web server by using this command

sudo yum install httpd

Step 3: Activate Apache

To activate Apache; we have to start its service first by using command:

sudo systemctl start httpd

now the Apache service we installed is activated.

Step 4: Enable Apache HTTP

Enable Apache server by using this command:

sudo systemctl enable httpd

We have now; enabled the Apache service to start when the system boots.

Step 5: Verify Apache Service

Display information about Apache, to verify its currently running status:

sudo systemctl status httpd

The below output verifies that the Apache service is running and active.

**ansible -i inventory\_file\_name -m module\_name -a "arguments"**

**ansible all -b -m yum -a "name=httpd update\_cache=yes state=present"**

In the above example we used yum module to install the httpd server .this yum module has several arguments like name, state and update\_cache.

It just installed the httpd but not running the services, to make sure that the services are running we should use another module like **service.**

**ansible all -b -m yum -a "name=httpd update\_cache=yes state=present" -m service -a "name=httpd state=started enabled=true"**

**ansible all -b -m yum -a "name=httpd update\_cache=yes state=removed" -m service -a "name=httpd state=stopped enabled=true"**

Now it's time to write a ansible play book for above adhoc command

**---**

name: installing apache httpd and enable the service

hosts : all

become : true

tasks:

-name: install httpd package

yum:

name: httpd

update\_cache=yes

state=present

-name: make sure service can activate

service:

name: httpd

state: started

enabled: yes

and save above file as httpd\_playbook.yaml

If we want to run above playbook then **ansible-playbook all httpd\_playbook.yml**

I used copy module to copy a file from controller to node system as follwos

copy:

src:source path

dist: Destination path.

Consider an example, We need to replcae the existing index.html page with the custome index.html page for that we need to create a index.html page and replcae that page into the following place

copy:

src: index.html

dist: /var/www/html/index.html

Vars:

Vars can be used to pass the variables as a dynamicallyv to the anisible playbook

vars:

value: key1

value: key2

---

hosts : all

become : true

vars:

courseName: Ansible

batchNumber: 10

tasks:

-name: install httpd package

yum:

name: httpd

update\_cache=yes

state=present

-name: make sure service can activate

service:

name: httpd

state: started

enabled: yes

-name: copy index.html

copy:

src: index.html

dist: /var/www/html/index.html

index.html

<h1> Welcome to {{courseName}} batch {{batchNumber}}

If we run the above play book the output will be **like "welcome {{courseName}} batch {{batchNumber}}".**

Because, Copy module won't update the variables. so we used template

We can pass the variables as arguments during playbook execution time as well by using

--extra-vars key=value

-e key=value

Notifiers and handlers:

Handlers are special type of tasks in ansible it can trigger at the end of the play which can be notified by other task.

Vars\_files: this can be used to call the variables.yml files in the playbook.

With\_Items