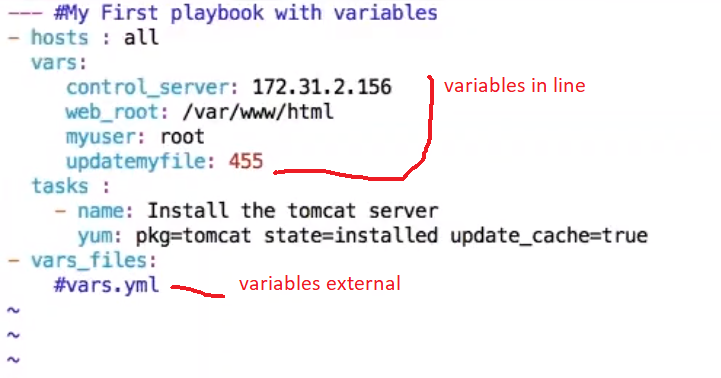
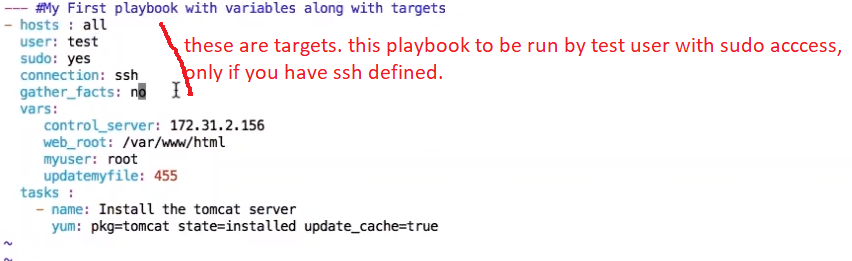
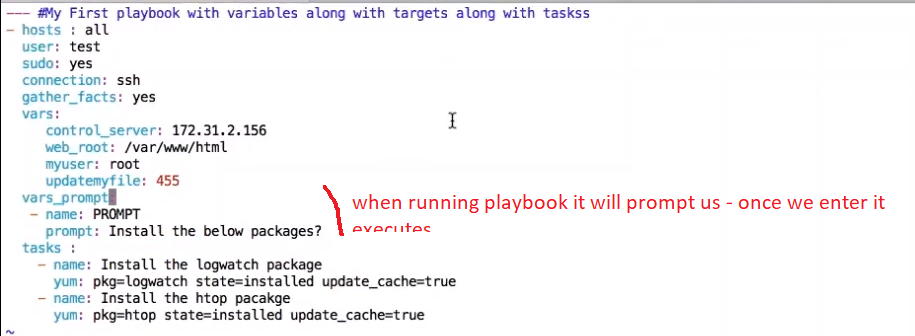
Ansible Extra Points



Targets



Prompts

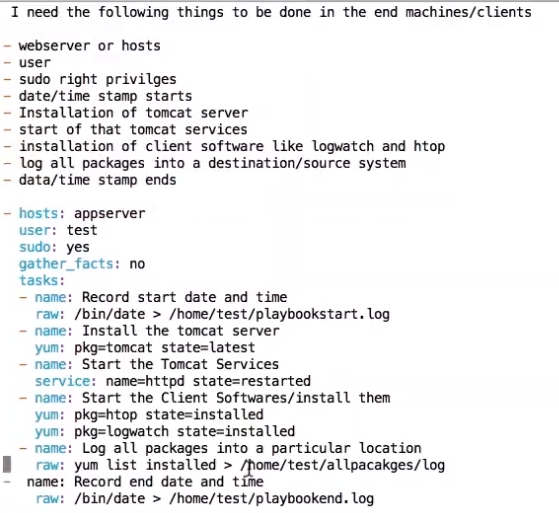


To uninstall a package change the state to absent instead of installed.

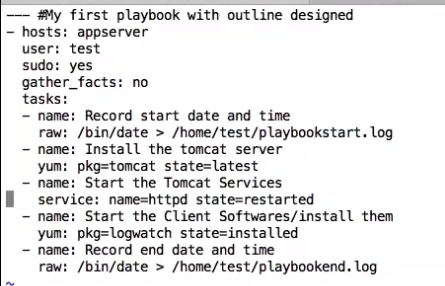
>ps -ef |grep -I httpd //to know httpd processes running

Outline of playbook

Ex:



Playbook content



Creating a new folder – using ansible playbook

---

- hosts: webservers

become: true

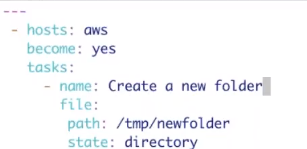
become\_user: root

tasks:

- name: create a new folder

file: path=/tmp/newfolder state=directory

or



**Creating a new file** in ansible using playbook

---

- hosts: webservers

become: true

become\_user: root

tasks:

- name: create a new file

file: path=/tmp/newfile2 state=touch

**using git module** in ansible in playbook

---

- hosts: webservers

become: true

become\_user: root

tasks:

- name: install git

yum: name=git state=present

- name: git clone

git: repo=https://github.com/venkata-coder/test5.git dest=/home/venkat

**debug module** to print data

---

- hosts: webservers

become: true

become\_user: root

vars:

y: hello

x: GM

tasks:

- name: print a message

debug: msg="the value of {{y}}" msg="the value of {{x}}"

note: variable precedence. Its prints x here and not y. prints the latest one which is x. in a single module – we cant have two. In a separate debug module we can.

**debug module** to print data – using a loop – it prints all messages

- hosts: webservers

tasks:

- name: run a loop

debug: msg="{{item}}"

with\_items:

- new york

- atlanta

**Register module** – output of previous command save it for use later

- hosts: webservers

tasks:

- shell: ls /tmp

register: out

- debug: msg="{{out}}"

**Output**

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

ok: [172.31.55.62] => {

"msg": {

"changed": true,

"cmd": "ls /tmp",

"delta": "0:00:00.003535",

"end": "2020-06-22 20:10:25.335950",

"failed": false,

"rc": 0,

"start": "2020-06-22 20:10:25.332415",

"stderr": "",

"stderr\_lines": [],

"stdout": "ansible\_command\_payload\_BIDXKt\nfolder\nnewfile2\nnewfolder\nnewfolder2\nsystemd-private-c74b9a143a374747be94a8d187d336cc-chronyd.service-X8lfNh",

"**stdout\_lines**": [

"ansible\_command\_payload\_BIDXKt",

"folder",

"newfile2",

"newfolder",

"newfolder2",

"systemd-private-c74b9a143a374747be94a8d187d336cc-chronyd.service-X8lfNh"

]

}

}

You need only output of stdout\_lines then lets change to as below

- hosts: webservers

tasks:

- shell: ls /tmp

register: out

- debug: msg="{{out.stdout\_lines}}"

Output:

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

ok: [172.31.55.62] => {

"msg": [

"ansible\_command\_payload\_WlLRdX",

"folder",

"newfile2",

"newfolder",

"newfolder2",

"systemd-private-c74b9a143a374747be94a8d187d336cc-chronyd.service-X8lfNh"

]

**Condition when** in ansible

---

- hosts: webservers

become: yes

tasks:

- name: install a webserve

yum: name=httpd state=present

when: ansible\_distribution == 'RedHat'

- name: start service

service: name=httpd state=started

it will install httpd if it’s a RedHat os.

**Stat module** for checking status

- hosts: webservers

become: yes

tasks:

- name: check if file exists

stat: path=/tmp/file.txt

register: stat\_result

- name: create file if not there

file: path=/tmp/file.txt state=touch

when: stat\_result.stat.exists == false

**Script module for running script – installing docker using a script**

- hosts: webservers

become: yes

tasks:

- name: run a script

script: /root/venkat/installdocker.sh

**adhoc command :** ansible webservers -a “docker version” --limit 34.34.342.45

we csn see docker installed on that host

**installing docker** using playbook

|  |
| --- |
| --- |
|  |

|  |
| --- |
| - hosts: all |
|  |

|  |
| --- |
| become: yes |
|  |

|  |
| --- |
| vars: |
|  |

|  |
| --- |
| dockerceversion: 5:18.09.9~3-0~ubuntu-bionic |
|  |

|  |
| --- |
| dockercecliversion: 5:18.09.9~3-0~ubuntu-bionic |
|  |

|  |
| --- |
| containerdioversion: 1.2.6-3 |
|  |

|  |
| --- |
| tasks: |
|  |

|  |
| --- |
| - name: Add Docker GPG key |
|  |

|  |
| --- |
| apt\_key: |
|  |

|  |
| --- |
| url: https://download.docker.com/linux/ubuntu/gpg |
|  |

|  |
| --- |
| - name: Install basic list of packages |
|  |

|  |
| --- |
| apt: |
|  |

|  |
| --- |
| name: ['apt-transport-https','ca-certificates','curl','gnupg2','software-properties-common'] |
|  |

|  |
| --- |
| state: present |
|  |

|  |
| --- |
| update\_cache: yes |
|  |

|  |
| --- |
| - name: Add Docker APT repository |
|  |

|  |
| --- |
| apt\_repository: |
|  |

|  |
| --- |
| repo: deb [arch=amd64] https://download.docker.com/linux/{{ansible\_distribution|lower}} {{ansible\_distribution\_release}} stable |
|  |

|  |
| --- |
| - name: Install Docker packages |
|  |

|  |
| --- |
| apt: |
|  |

|  |
| --- |
| # name: "{{ item }}" |
|  |

|  |
| --- |
| name: ['docker-ce={{ dockerceversion }}','docker-ce-cli={{ dockercecliversion }}', 'containerd.io={{ containerdioversion }}'] |
|  |

|  |
| --- |
| state: present |
|  |

|  |
| --- |
| # with\_items: |
|  |

|  |
| --- |
| # - docker-ce={{ dockerceversion }} |
|  |

|  |
| --- |
| # - docker-ce-cli={{ dockercecliversion }} |
|  |

|  |
| --- |
| # - containerd.io={{ containerdioversion }} |
|  |

|  |
| --- |
| # - docker-ce |
|  |

|  |
| --- |
| # - docker-ce-cli |
|  |

# - containerd.io

**Create a EC2 instance using ansible playbook**

1. Setup aws command line
   1. sudo yum update -y
   2. sudo yum install python3-pip
   3. sudo yum install python-pip
   4. sudo yum install awscli
   5. sudo yum install boto
   6. aws --version
2. Authentication and authorization
   1. Crate user in aws with amazonec2fullaccess policy
   2. Aws configure in acm – give acces an secret access key
3. Playbook ec2.yml

- name: create ec2 instance

hosts: localhost

tasks:

- ec2:

key\_name: 23

instance\_type: t2.micro

image: ami-09d95fab7fff3776c

wait: yes

group: ans\_sg\_2

count: 1

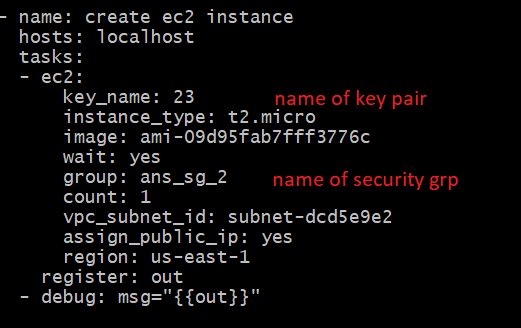
vpc\_subnet\_id: subnet-dcd5e9e2

assign\_public\_ip: yes

region: us-east-1

register: out

- debug: msg="{{out}}"



We can further extend this – adding hosts to a host file. Then installing apache



Look at some more playbooks

<https://github.com/lerndevops/labs/tree/master/aws>

<https://github.com/lerndevops/labs/blob/master/aws/t2-medium-u18.yml>

<https://github.com/lerndevops/labs/blob/master/aws/t2-micro-u18.yml>

create an ec2 instance with tags and terminate it

- name: create ec2 instance

hosts: localhost

tasks:

- ec2:

key\_name: 23

instance\_type: t2.micro

image: ami-09d95fab7fff3776c

wait: yes

group: ans\_sg\_2

count: 1

instance\_tags: nam=myapp

vpc\_subnet\_id: subnet-dcd5e9e2

assign\_public\_ip: yes

region: us-east-1

register: out

- debug: msg="{{out}}"

- name: terminate instance

hosts: localhost

tasks:

- ec2: state=absent instance\_ids="{{out.instance\_ids}}" region=us-east-1