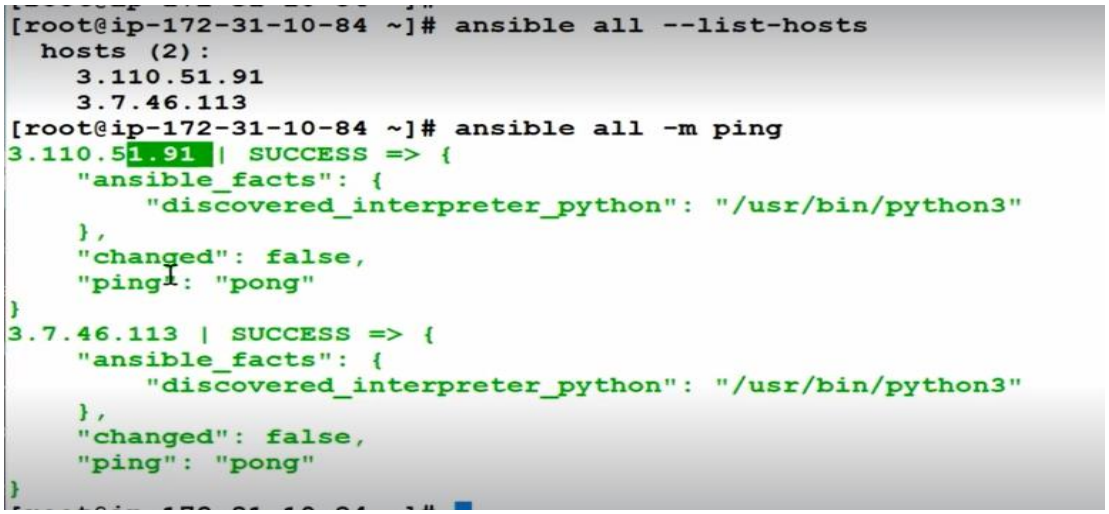
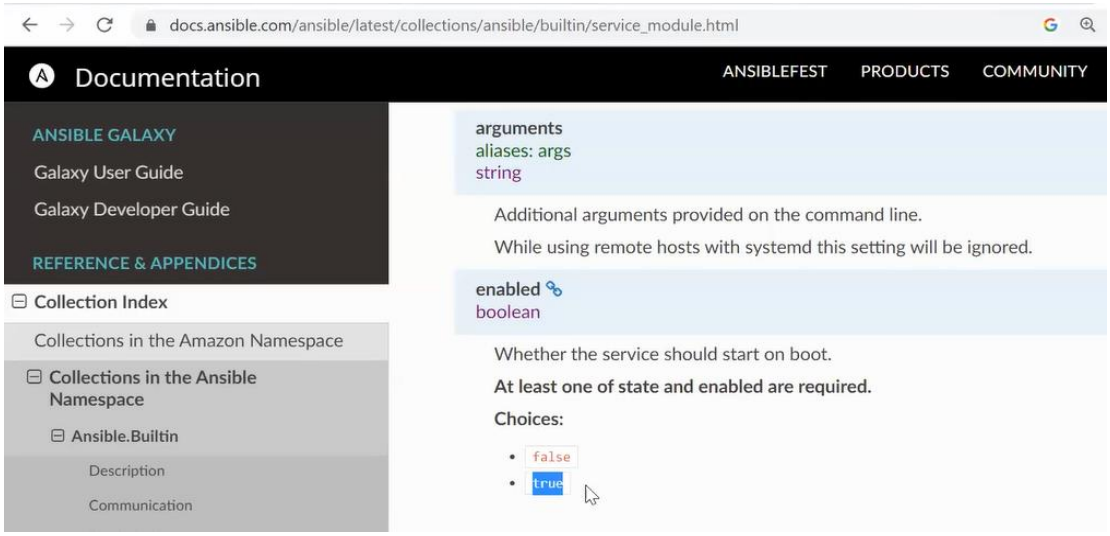
Ansible

Session 3 – 17th Dec 2022 Summary

- The "**ping**" module checks the connectivity between the controller node and target node, here we see both the target nodes are successfully connected.



- To make the service permanent at boot time we can use "**enabled**" keyword with "**true**" value in the "**service**" module



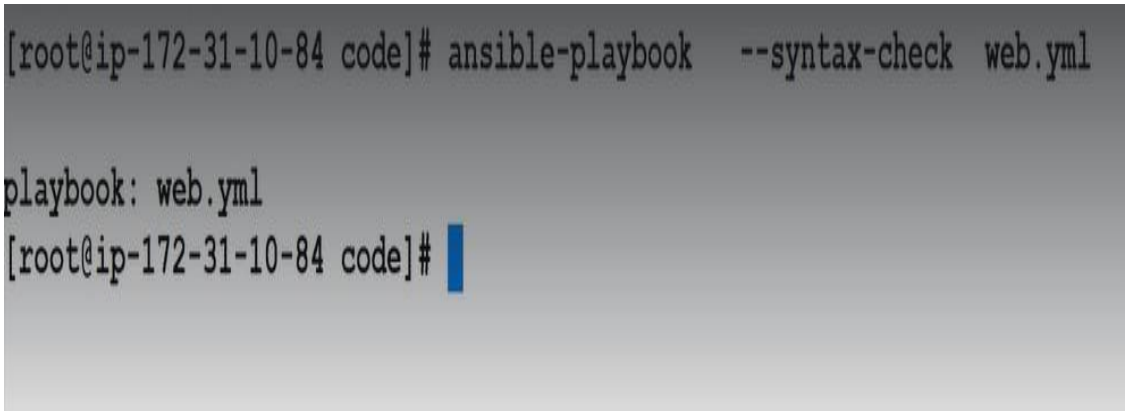
- Here we see "**name**" is the keyword and "**httpd**" is the value that we specify. The string that we specify should be put in **double quotes.** The value "**started**" and "**true**" are pre-defined value and need not be put in **double quotes.**

```
- hosts: all
  tasks:
    - package: "name=httpd state=present"
    - copy:  "src=index.html  dest=/var/www/html/"

    - service:
        name: "httpd"
        state: started
        enabled: true
~
~
```

- Similarly the same can be done for package and copy modules

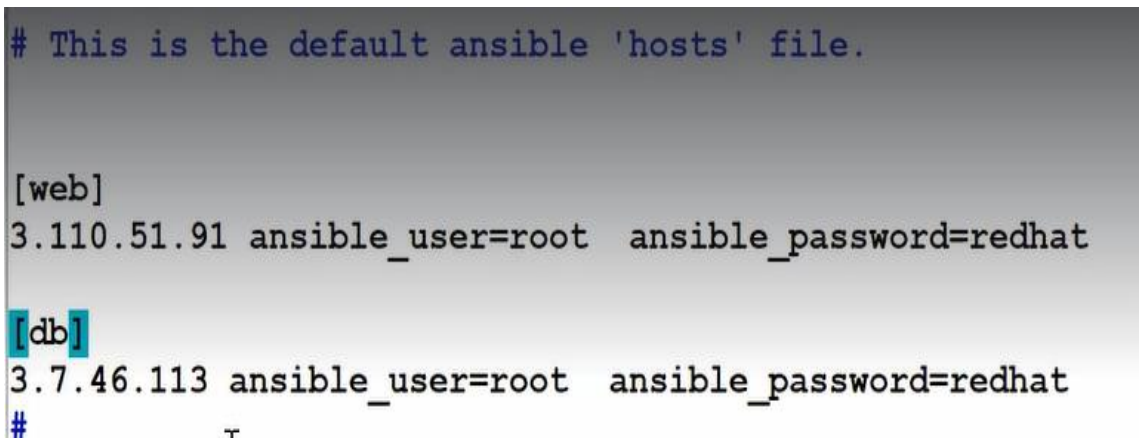
```
- hosts: all
  tasks:
    - package:
        name: "httpd"
        state: present

    - copy:
        src: "index.html"
        dest: "/var/www/html/"

    - service:
        name: "httpd"
        state: started
        enabled: true
~
```

- It's a good practice to check syntax before running the playbook, the command used is "ansible-playbook - - syntax – check playbook_name". Here the playbook name is "**web.yml**"



- The command to run the playbook "**ansible – playbook playbook_name**"

- We can create a host group and give a name in the inventory file so that management becomes easier and simple. In future if we want any node to be configured as webserver we need not go to playbook and make changes instead we can add the node in the group in the inventory file, by this scaling becomes easier



- The “all” is a special group that contains all the hosts

```
[root@ip-172-31-10-84 code]# ansible all --list-hosts
  hosts (6):
    1.2.3.4
    2.3.4.5
    3.3.3.3
    4.4.4.4
    3.110.51.91
    3.7.46.113
```

- The “**debug**” **module** is used to print a message during the execution of the code. The “**msg**” is the **keyword** and the value “**hi i m vimal**” is the **value** that we give.

```
- hosts: localhost
  tasks:
    - debug:
        msg: "hi i m vimal"
~
```

```
[root@ip-172-31-10-84 code]# ansible-playbook  var.yml

PLAY [localhost] ***************************************************************

TASK [Gathering Facts] *********************************************************
ok: [localhost]

TASK [debug] *******************************************************************
ok: [localhost] => {
    "msg": "hi i m vimal"
}

PLAY RECAP *********************************************************************
localhost                  : ok=2    changed=0    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
```

- In ansible we can create variables using “**vars**” keyword. The variables specified before the task is called **Global Variables**, that can be used anywhere in the code. Here “x” is a variable using the value “5”

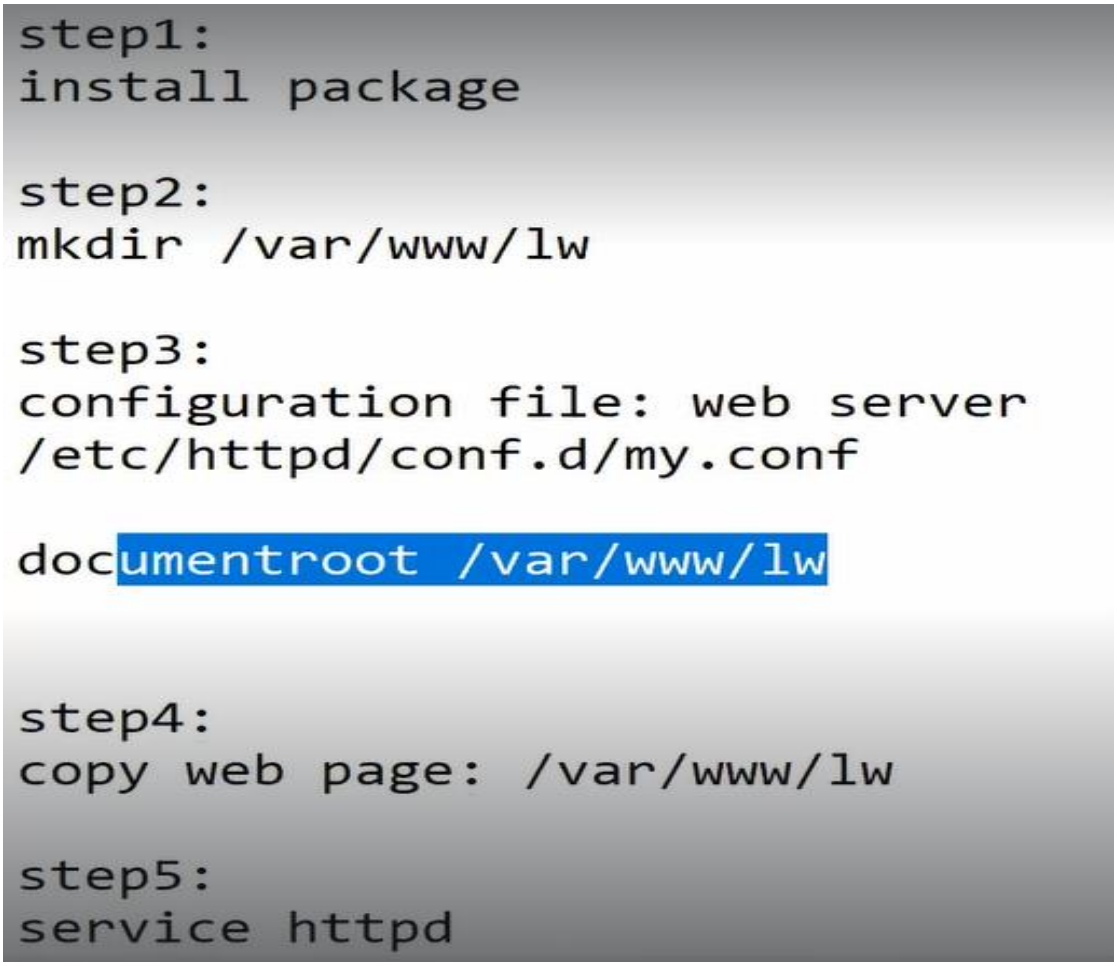
```
- hosts: localhost
  vars:
    - x: 5
    - y: 10

  tasks:
    - debug:
        msg: "hi i m vimal {{ x }} tc bye"
~
```

```
[root@ip-172-31-10-84 code]# ansible-playbook  var.yml

PLAY [localhost] ***********************************************************

TASK [Gathering Facts] *****************************************************
ok: [localhost]

TASK [debug] ***************************************************************
ok: [localhost] => {
    "msg": "hi i m vimal 5 tc bye"
}

PLAY RECAP *****************************************************************
localhost                  : ok=2    changed=0    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
```

- While configuring Apache HTTPD Webserver, we can declare "**variables**", before the tasks. Here the "**packageName**" variable holds the value of the package to be installed. The variable "**webPage**" hold the webpage hosted on the controller node and to be copied to target node.

```
- hosts: web
  vars:
    - webPage: "index.html"
    - packageName: "httpd"

  tasks:
    - package:
        name: "{{ packageName }}"
        state: present

    - copy:
        src: "{{ webPage }}"
        dest: "/var/www/html/"

    - service:
        name: "httpd"
        state: started
        enabled: true
~
```

- In the Apache Webserver the **default document root** is "**/var/www/html",** where the webpages are copied. This default document root can be change in the internal (secondary configuration) file **"/etc/httpd/conf.d**" of the Apache Webserver.



- For this we have to create directory, the **"file"** module can be used, the "**state**" is the **keyword** and "**directory**" is "**predefined value**". The "**path**" keyword specifies the path where the directory has to be created. For this we can create a variable "**documentDir**" to specify the path **"/var/www/lw**". The "**copy**" module is used to copy the content "**documentroot /var/www/lw**" to the "**destination**" **"/etc/httpd/conf.d**". After making changes in the configuration file, we have to reload the service. For this we have "**service**" module, the "**state**" keyword is used with the value "**reloaded**"

```
root@ip-172-31-10-84:/code
- hosts: web
  vars:
    - webPage: "index.html"
    - packageName: "httpd"
    - documentDir: "/var/www/lw"

  tasks:
    - package:
        name: "{{ packageName }}"
        state: present

    - file:
        state: directory
        path: "{{ documentDir }}"

    - copy:
        dest: "/etc/httpd/conf.d/my.conf"
        content: "documentroot {{ documentDir }}\n"

    - copy:
        src: "{{ webPage }}"
        dest: "{{ documentDir }}"

    - service:
        name: "httpd"
        state: reloaded
        enabled: true
```

- It's a good practice to give every task a name, when we run the playbook it shows what is the specific task doing

```
tasks:
  - name: installed httpd package
    package:
      name: "{{ packageName }}"
      state: present

  - name: create doc root
    file:
      state: directory
      path: "{{ documentDir }}"

  - name: setting conf file
    copy:
      dest: "/etc/httpd/conf.d/my.conf"
      content: "documentroot {{ documentDir }}\n"

  - name: deploy web page
    copy:
      src: "{{ webPage }}"
```

```
- name: reload service
  service:
    name: "httpd"
    state: reloaded
    enabled: true
```

- The command to check the module from the command line is “ansible-doc module_name”
- The command to list all the module from the command line is “ansible-doc –l ”

Important Links –

Hash13 link for Sessions and extra sessions recordings – https://learning.hash13.com/

How to build Inventory:- https://docs.ansible.com/ansible/latest/inventory_guide/intro_inventory.html

Ansible Documentation- Copy Module - https://docs.ansible.com/ansible/latest/collections/ansible/builtin/copy_module.html

Ansible Documentation- Package Module - https://docs.ansible.com/ansible/latest/collections/ansible/builtin/package_module.html

Ansible Documentation- Service Module – https://docs.ansible.com/ansible/latest/collections/ansible/builtin/service_module.html

Ansible Documentation – Debug Module – https://docs.ansible.com/ansible/latest/collections/ansible/builtin/debug_module.html