9-Dec-2024

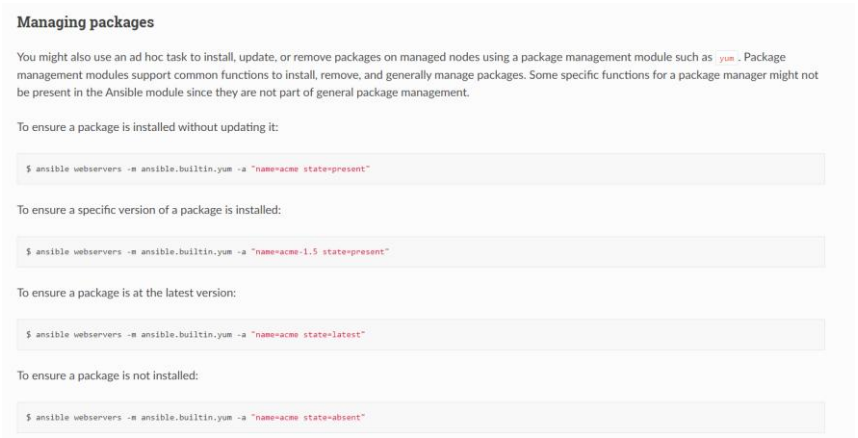
Internship Day - 96 Report:

Exercise 5: (Adhoc Commands in Ansible)

https://docs.ansible.com/ansible/latest/command_guide/intro_adhoc.html

change the hostname of your all machines if u want

Manage Packages Commands:



Managing packages

You might also use an ad hoc task to install, update, or remove packages on managed nodes using a package management module such as `yum`. Package management modules support common functions to install, remove, and generally manage packages. Some specific functions for a package manager might not be present in the Ansible module since they are not part of general package management.

To ensure a package is installed without updating it:

```
$ ansible webservers -m ansible.builtin.yum -a "name=acme state=present"
```

To ensure a specific version of a package is installed:

```
$ ansible webservers -m ansible.builtin.yum -a "name=acme-1.5 state=present"
```

To ensure a package is at the latest version:

```
$ ansible webservers -m ansible.builtin.yum -a "name=acme state=latest"
```

To ensure a package is not installed:

```
$ ansible webservers -m ansible.builtin.yum -a "name=acme state=absent"
```

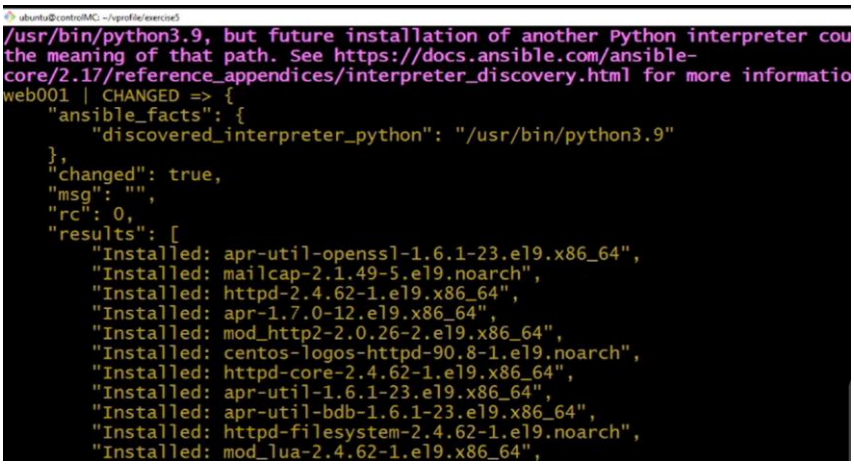
For install:

ansible <machine-name> -m ansible.builtin.yum -a "name=<webserver-name> state=present"

appu nu become lgana penda ehnu run karn leyi (become means sudo/root user)

ansible web001 -m ansible.builtin.yum -a "name=httpd state=present" –become

output ist time:



2nd time run karn ch khduga eh already done aa

For remove/ delete:

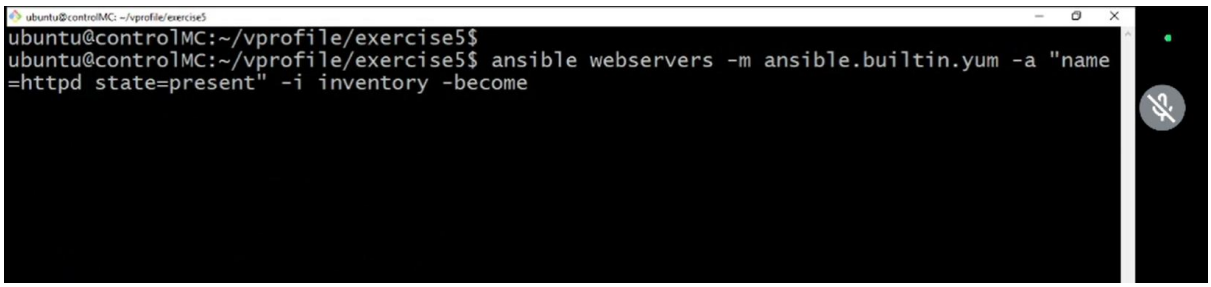
ansible web001 -m ansible.builtin.yum -a "name=httpd state=absent" –become

Server on / off command:

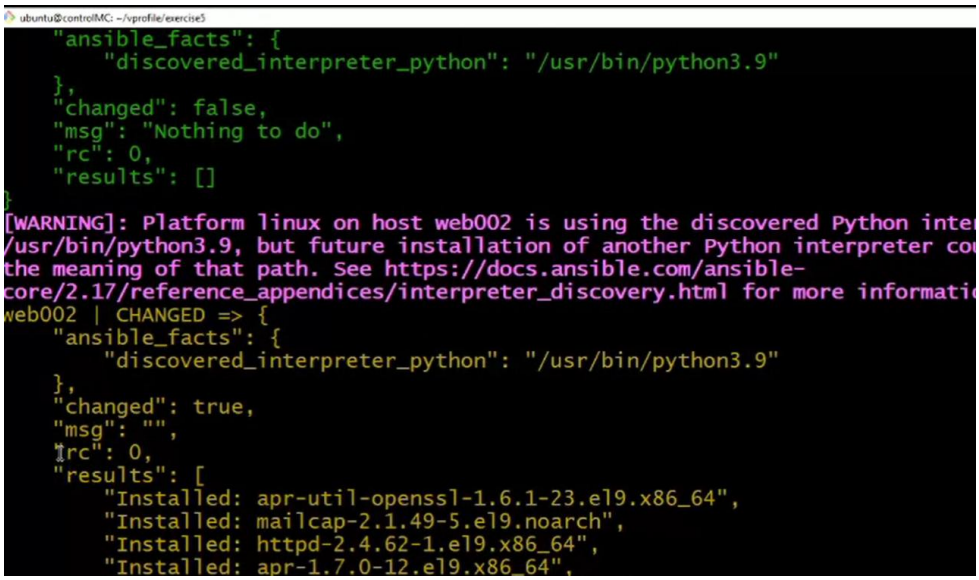
ansible web001 -m ansible.builtin.service -a "name=httpd state=started" –i inventory –b

Output:

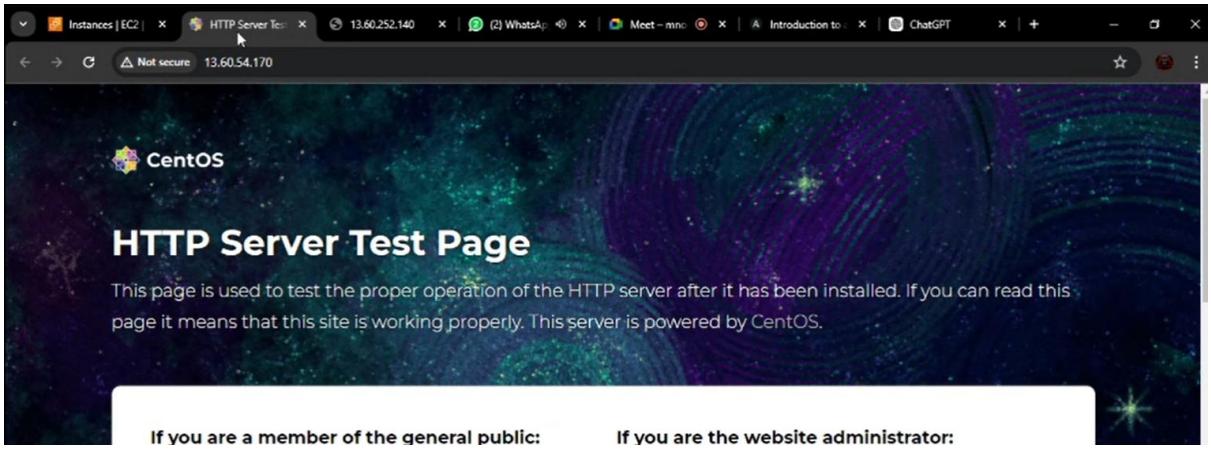
Now hun je appa ek group ch install krni howe services --- and yeh kisi ch install hegi aa tah thik aa yeh nnhi aa tah install krdu service --- agr sbb ch install aa tah kise ch kuch nhi kru



Output:



Browser o/p:



For remove service in group:

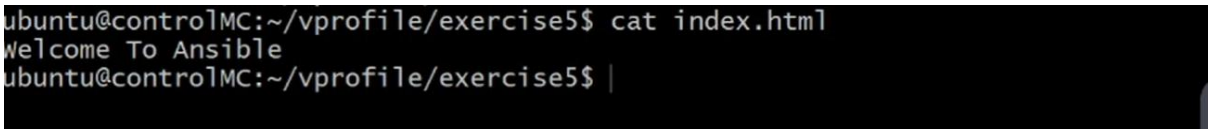
10-Dec-2024

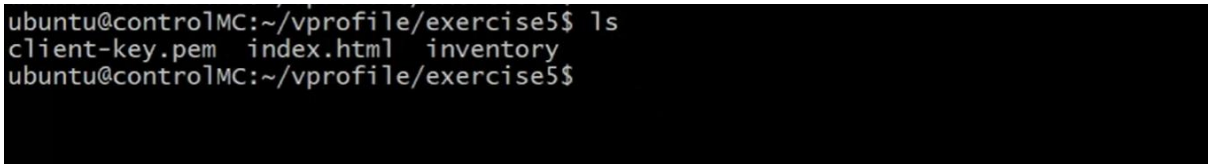
Internship Day - 97 Report:

Exercise 5:

File and project deploy using Ansible :

Machines nu http protocols allow krni security ch



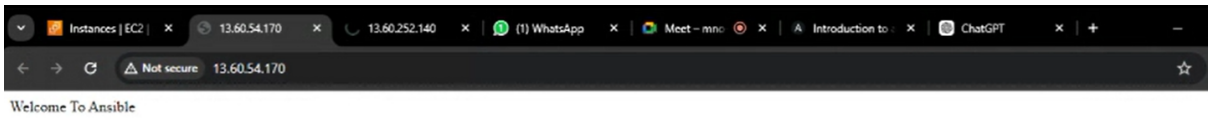


Copy krna project /var/www/html location ch

Output:

```
ubuntu@controlMC:~/vprofile/exercise5$
ubuntu@controlMC:~/vprofile/exercise5$ ansible webservers -m copy -a "src=index.html dest
=/var/www/html" -i inventory -become
[WARNING]: Platform linux on host web002 is using the discovered Python interpreter at
/usr/bin/python3.9, but future installation of another Python interpreter could change
the meaning of that path. See https://docs.ansible.com/ansible-
core/2.17/reference_appendices/interpreter_discovery.html for more information.
web002 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.9"
    },
    "changed": true,
    "checksum": "82e5c398f96bc3288f1549632641759a4b68ee32",
    "dest": "/var/www/html/index.html",
```

Done uploading:



Welcome To Ansible

11-Dec-2024

Internship Day - 98 Report:

Playbook file

https://docs.ansible.com/ansible/latest/playbook_guide/playbooks_intro.html

What are playbooks?

Ansible is an orchestration tool. It needs a list of tasks/instructions to perform on the machines listed in the inventory file. You can execute a task with Ansible more than once using a playbook. **Playbooks** offer repeatable, reusable, simple configuration management. It can be used for configuration management, orchestrating steps of any manual process on multiple machines in synchronous or asynchronous order.

A playbook is written in YAML format. It is composed of one or more 'plays' in an ordered list. And one play contains a set of tasks that runs on a group of machines.

Different YAML Tags

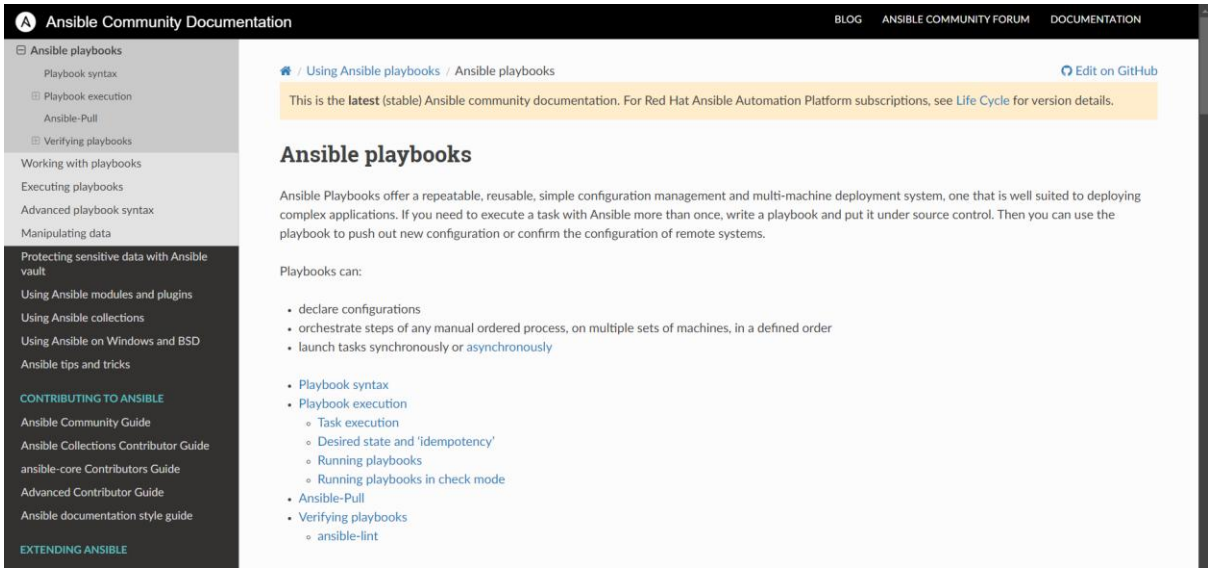
Let us now go through the different tags in a playbook-

- **name:** Logical name of the task which specifies what this playbook will be doing
- **hosts:** This specifies the lists of hosts against which we want to run the task
- **vars:** This allows you to define and use variables in your playbook
- **tasks:** Tasks are a list of actions the playbook will perform

In an ad-hoc command setup, we need to run commands repeatedly to achieve the same configuration on different machines. This means we have to manually execute commands every time we want to make changes or ensure consistency.

However, with a **playbook file**, we can write the commands in a file and use it to execute the desired tasks on multiple machines in one go. This approach saves effort and ensures consistency across all machines, as the same playbook file can be reused whenever needed.

Essentially, by creating and keeping a **playbook file**, we can streamline the process and use it whenever required instead of running ad-hoc commands repeatedly.



Exercise 6:

1. **Make a playbook yaml file**

```
ubuntu@ip-172-31-32-53:~/vprofile/exercise6$ ls
client-key.pem  index.html  inventory
ubuntu@ip-172-31-32-53:~/vprofile/exercise6$ vim server_play.yaml
ubuntu@ip-172-31-32-53:~/vprofile/exercise6$
```

```
---
- name: Webserver setup
  hosts: webservers
  become: yes
  tasks:
    - name: Install httpd
      ansible.builtin.yum:
        name: httpd
        state: present


    - name: Start service
      ansible.builtin.service:
        name: httpd
        state: started
        enable: yes
```

2. **Run the command:**

```
ubuntu@ip-172-31-32-53:~/vprofile/exercise6$ ansible-playbook -i inven
tory server_play.yaml
```

Ansible-playbook –I inventory <playbook-file-name>

3. **Output:**

```
[WARNING]: Platform linux on host web001 is using the discovered Pytho
n
interpreter at /usr/bin/python3.9, but future installation of another
Python
interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-
core/2.17/reference_appendices/interpreter_discovery.html for more inf
ormation.
ok: [web001]
[WARNING]: Platform linux on host web002 is using the discovered Pytho
n
interpreter at /usr/bin/python3.9, but future installation of another
Python
interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-
core/2.17/reference_appendices/interpreter_discovery.html for
ormation.
ok: [web002]
```

4. **Readable Form:**

   To make the output more readable and detailed, you can use the **-v** option (verbose mode) with the ansible-playbook command.

   Command:

   ansible-playbook -i inventory <playbook-file-name> -v

12-Dec-2024

Internship Day - 99 Report:

Exercise 7:

In playbook file add lines For Dbserver

```
- name: DBserver setup
  hosts: dbservers
  become: yes
  tasks:
    - name: Installing Mariab-Db in Lab3
      ansible.builtin.yum:
        name: mariadb-server
        state: present

    - name: Start Mariab DB service
      ansible.builtin.service:
        name: mariadb
        state: started
        enabled: yes
```

Run the command:

```
ubuntu@ip-172-31-32-53:~/vprofile/exercise7$ ansible-playbook -i inven
tory server_play.yaml
```

Output:

```
PLAY RECAP *********************************************************
**********
db001                      : ok=3    changed=2    unreachable
led=0    skipped=0    rescued=0    ignored=0
web001                     : ok=3    changed=1    unreachable
led=0    skipped=0    rescued=0    ignored=0
web002                     : ok=3    changed=1    unreachable
led=0    skipped=0    rescued=0    ignored=0
```

Exercise 8: (super global variable bnane sikhange)

```
ubuntu@ip-172-31-32-53:~/vprofile$ cp -r exercise7 exercise8
ubuntu@ip-172-31-32-53:~/vprofile$
ubuntu@ip-172-31-32-53:~/vprofile$ cd exercise8
ubuntu@ip-172-31-32-53:~/vprofile/exercise8$
ubuntu@ip-172-31-32-53:~/vprofile/exercise8$
ubuntu@ip-172-31-32-53:~/vprofile/exercise8$
ubuntu@ip-172-31-32-53:~/vprofile/exercise8$ ls
client-key.pem  index.html  inventory  server_play.yaml
ubuntu@ip-172-31-32-53:~/vprofile/exercise8$
```

Make a group_vars **directory.. it is predefined !**

```
mkdir group_vars
```

Ls

```
ubuntu@ip-172-31-32-53:~/vprofile/exercise8$ ls
client-key.pem  group_vars  index.html  inventory  server_play.yaml
```

Cd group_vars

Touch all (all file is complousary bnania kyuki playbok file nu access krn leyi)

```
ubuntu@ip-172-31-32-53:~/vprofile/exercise8$ cd group_vars
ubuntu@ip-172-31-32-53:~/vprofile/exercise8/group_vars$ touch all
ubuntu@ip-172-31-32-53:~/vprofile/exercise8/group_vars$ ls
all
ubuntu@ip-172-31-32-53:~/vprofile/exercise8/group_vars$
```

Vi all

```
ubuntu@ip-172-31-32-53: ~/vprofile/exercise8/group_vars
name: pankaj sharma
phonenumber: 9728082087
city: chandigarh
~
~
~
~
~
```

Cat

```
ubuntu@ip-172-31-32-53:~/vprofile/exercise8/group_vars$ vim all
ubuntu@ip-172-31-32-53:~/vprofile/exercise8/group_vars$
ubuntu@ip-172-31-32-53:~/vprofile/exercise8/group_vars$ cat all
name: pankaj sharma
phonenumber: 9728082087
city: chandigarh
ubuntu@ip-172-31-32-53:~/vprofile/exercise8/group_vars$
```

Now hun aapa playbook file edit krange

Debug predefined variable use krange and ek time ch ek hi msg print ho skda

```
---
- name: Webserver setup
  hosts: webservers
  become: yes
  tasks:
    - debug:
        msg: "my name is {{name}}"

    - debug:
        msg: "MObile no is {{phonenumber}}"

    - debug:
        msg: "City is {{city}}"
```

Now run the file:

```
ubuntu@ip-172-31-32-53:~/vprofile/exercise8$ ansible-playbook -i inven
tory server_play.yaml

PLAY [Webserver setup] ***********************************
**********
```

Output:


ok: [web001] => {
"msg": "my name is pankaj sharma"
}
ok: [web002] => {
"msg": "my name is pankaj sharma"
}
TASK [debug] **
****
ok: [web001] => {
"msg": "MObile no is 9728082087"
}
ok: [web002] => {
"msg": "MObile no is 9728082087"
}
TASK [debug] **
****
ok: [web001] => {
"msg": "City is chandigarh"
}
deep kumar is presenting

13-Dec-2024

Internship Day - 100 Report:

Main imp:

If ur variables is also present in playbook file (grup_vars – all de nal nal)

Then ansible phle tuhade playbook aale variables chku ga then all file aale super global variables chkkuga