

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:

```
ubuntu@controlMC: ~/vprofile/exercise5
/usr/bin/python3.9, but future installation of another Python interpreter could
the meaning of that path. See https://docs.ansible.com/ansible-
core/2.17/reference_appendices/interpreter_discovery.html for more information.
web001 | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.9"
  },
  "changed": true,
  "msg": "",
  "rc": 0,
  "results": [
    "Installed: apr-util-openssl-1.6.1-23.el9.x86_64",
    "Installed: mailcap-2.1.49-5.el9.noarch",
    "Installed: httpd-2.4.62-1.el9.x86_64",
    "Installed: apr-1.7.0-12.el9.x86_64",
    "Installed: mod_http2-2.0.26-2.el9.x86_64",
    "Installed: centos-logos-httpd-90.8-1.el9.noarch",
    "Installed: httpd-core-2.4.62-1.el9.x86_64",
    "Installed: apr-util-1.6.1-23.el9.x86_64",
    "Installed: apr-util-bdb-1.6.1-23.el9.x86_64",
    "Installed: httpd-filesystem-2.4.62-1.el9.noarch",
    "Installed: mod_lua-2.4.62-1.el9.x86_64",
  ]
}
```

2nd time run karn ch khduga eh already done aa

```
ubuntu@controlMC:~/vprofile/exercise5$ ansible web001 -m ansible.builtin.yum -a "name=httpd state=present" -i inventory -become
[WARNING]: Platform linux on host web001 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.
web001 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.9"
  },
  "changed": false,
  "msg": "Nothing to do",
  "rc": 0,
  "results": []
}
```

For remove/ delete:

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

```
ubuntu@controlMC:~/vprofile/exercise5$ ansible web001 -m ansible.builtin.yum -a "name=httpd state=absent" -i inventory -become
[WARNING]: Platform linux on host web001 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.
web001 | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.9"
  },
  "changed": true,
  "msg": "",
  "rc": 0,
  "results": [
    "Removed: mod_http2-2.0.26-2.el9.x86_64",
    "Removed: httpd-2.4.62-1.el9.x86_64"
  ]
}
```

deep kumar is presenting

Server on / off command:

Managing services

Ensure a service is started on all webserver:

```
$ ansible webservers -m ansible.builtin.service -a "name=httpd state=started"
```

Alternatively, restart a service on all webserver:

```
$ ansible webservers -m ansible.builtin.service -a "name=httpd state=restarted"
```

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

Output:

```
"Transient": "no",  
"Type": "notification",  
"UID": "[not set]",  
"UMask": "0022",  
"UnitFilePreset": "disabled",  
"UnitFileState": "disabled",  
"UtmpMode": "init",  
"Wants": "httpd-init.service",  
"WatchdogSignal": "6",  
"WatchdogTimestampMonotonic": "0",  
"WatchdogUseC": "infinity"  
}  
}
```

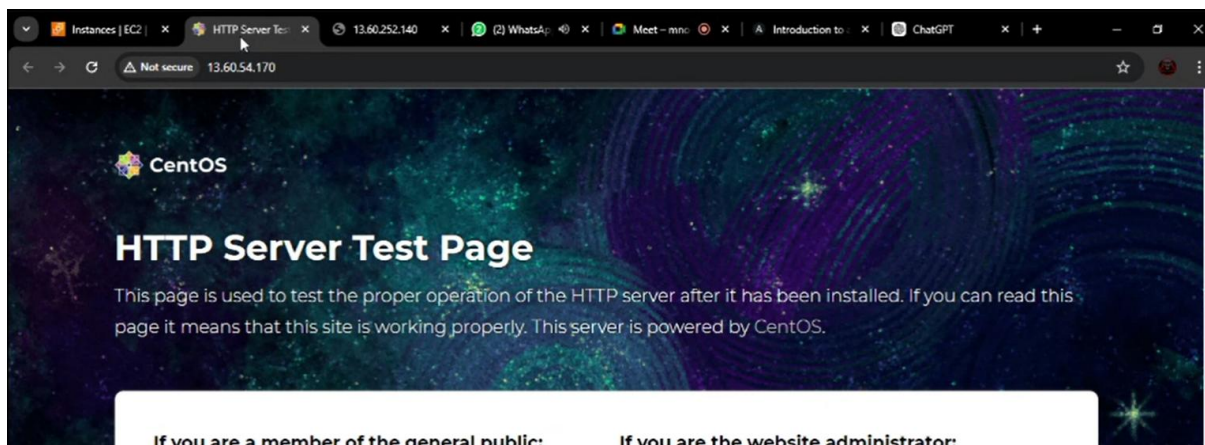
**Now hun je appa ek group ch install krni howe services --- and yeh kisi ch install hegi
aa tah thik aa yeh nnhi aa tah install krdu service --- agr sbb ch install aa tah kise ch
kuch nhi kru**

```
ubuntu@controlMC: ~/vprofile/exercise5
ubuntu@controlMC:~/vprofile/exercise5$ ansible webserver -m ansible.builtin.yum -a "name=httpd state=present" -i inventory -become
```

Output:

```
ubuntu@controlMC: ~/vprofile/exercise5$  
  "ansible_facts": {  
    "discovered_interpreter_python": "/usr/bin/python3.9"  
  },  
  "changed": false,  
  "msg": "Nothing to do",  
  "rc": 0,  
  "results": []  
}  
[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,  
  "msg": "",  
  "rc": 0,  
  "results": [  
    "Installed: apr-util-openssl-1.6.1-23.el9.x86_64",  
    "Installed: mailcap-2.1.49-5.el9.noarch",  
    "Installed: httpd-2.4.62-1.el9.x86_64",  
    "Installed: apr-1.7.0-12.el9.x86_64",  
  ]  
}
```

Browser o/p:



For remove service in group:

```
ubuntu@controlMC:~/vprofile/exercise5$ ansible webserver -m ansible.builtin.yum -a "name=httpd state=absent" -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"  
  },  
}
```

10-Dec-2024

Internship Day - 97 Report:

Exercise 5:

File and project deploy using Ansible :

Machines nu http protocols allow krni security ch

```
ubuntu@controlMC:~/vprofile/exercise5$ vim index.html
ubuntu@controlMC:~/vprofile/exercise5$
```

```
ubuntu@controlMC: ~/vprofile/exercise5
Welcome To Ansible
~
~
~
```

```
ubuntu@controlMC:~/vprofile/exercise5$ cat index.html
Welcome To Ansible
ubuntu@controlMC:~/vprofile/exercise5$ |
```

```
ubuntu@controlMC:~/vprofile/exercise5$ ls
client-key.pem  index.html  inventory
ubuntu@controlMC:~/vprofile/exercise5$
```

Copy krna project /var/www/html location ch

Check mode

In check mode, Ansible does not make any changes to remote systems. Ansible prints the commands only. It does not run the commands.

```
$ ansible all -m copy -a "content=foo dest=/root/bar.txt" -C
```

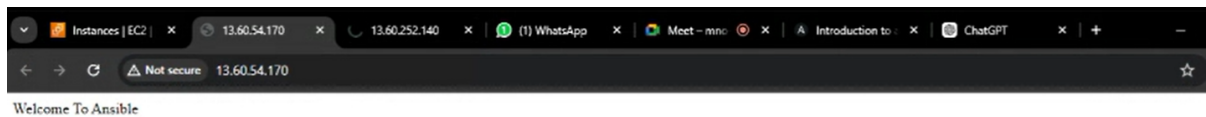
Enabling check mode (`-C` or `--check`) in the above command means Ansible does not actually create or update the `/root/bar.txt` file on any remote systems.

```
ubuntu@controlMC:~/vprofile/exercise5$ ansible webserver -m copy -a "src=index.html dest=/var/www/html" -i inventory -become
```

Output:

```
ubuntu@controlMC:~/vprofile/exercise5$  
ubuntu@controlMC:~/vprofile/exercise5$ ansible webserver -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",  
  "src": "index.html"
```

Done uploading:



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

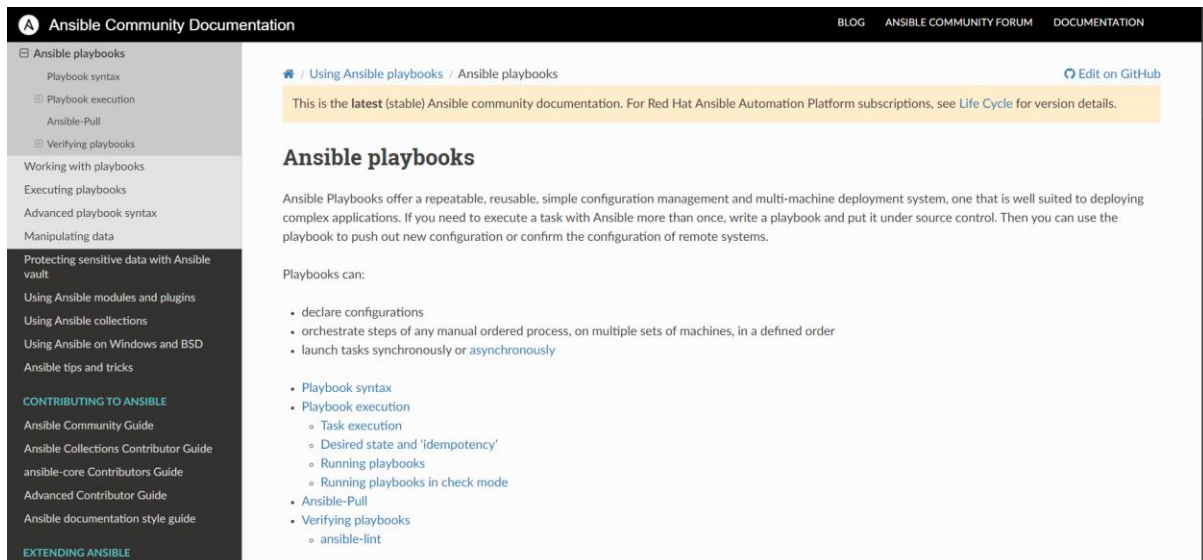
Let us now go through the different tags in a playbook-

- **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.



The screenshot shows the 'Ansible Community Documentation' website. The header includes the site name and navigation links for 'BLOG', 'ANSIBLE COMMUNITY FORUM', and 'DOCUMENTATION'. A left sidebar contains a tree view of documentation topics, with 'Using Ansible playbooks' selected. The main content area has a breadcrumb trail 'Using Ansible playbooks / Ansible playbooks' and an 'Edit on GitHub' link. A yellow banner states: 'This is the latest (stable) Ansible community documentation. For Red Hat Ansible Automation Platform subscriptions, see Life Cycle for version details.' The title 'Ansible playbooks' is followed by a paragraph explaining that Ansible Playbooks offer a repeatable, reusable, simple configuration management and multi-machine deployment system. Below this, a section 'Playbooks can:' lists capabilities: declare configurations, orchestrate steps, and launch tasks synchronously or asynchronously. A detailed bulleted list follows, covering 'Playbook syntax', 'Playbook execution' (with sub-points for task execution, idempotency, running playbooks, and check mode), 'Ansible-Pull', and 'Verifying playbooks' (with a sub-point for ansible-lint).

Ansible Community Documentation

BLOG ANSIBLE COMMUNITY FORUM DOCUMENTATION

Using Ansible playbooks / Ansible playbooks [Edit on GitHub](#)

This is the **latest** (stable) Ansible community documentation. For Red Hat Ansible Automation Platform subscriptions, see [Life Cycle](#) for version details.

Ansible playbooks

Ansible Playbooks offer a repeatable, reusable, simple configuration management and multi-machine deployment system, one that is well suited to deploying complex applications. If you need to execute a task with Ansible more than once, write a playbook and put it under source control. Then you can use the playbook to push out new configuration or confirm the configuration of remote systems.

Playbooks can:

- declare configurations
- orchestrate steps of any manual ordered process, on multiple sets of machines, in a defined order
- launch tasks synchronously or [asynchronously](#)

- Playbook syntax
- Playbook execution
 - Task execution
 - Desired state and 'idempotency'
 - Running playbooks
 - Running playbooks in check mode
- Ansible-Pull
- Verifying playbooks
 - [ansible-lint](#)

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: webserver
  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 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 inf
ormation.
ok: [web001]
[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 inf
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=0    failed=0
skipped=0  rescued=0    ignored=0
web001     : ok=3    changed=1    unreachable=0    failed=0
skipped=0  rescued=0    ignored=0
web002     : ok=3    changed=1    unreachable=0    failed=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 krangle

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

```
---
- name: Webserver setup
  hosts: webserver
  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

```
ubuntu@ip-172-31-32-53: ~/vprofile/exercise8
--
- name: Webserver setup
  hosts: webserver
  become: yes
  vars:
    name: rohan
    phonenuber: 723222134
    city: Ambala
  tasks:
    - debug:
        msg: "my name is {{name}}"

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

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

~
```