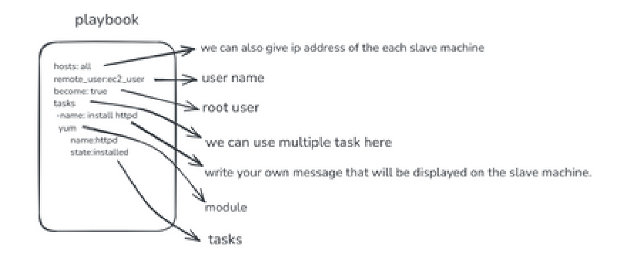
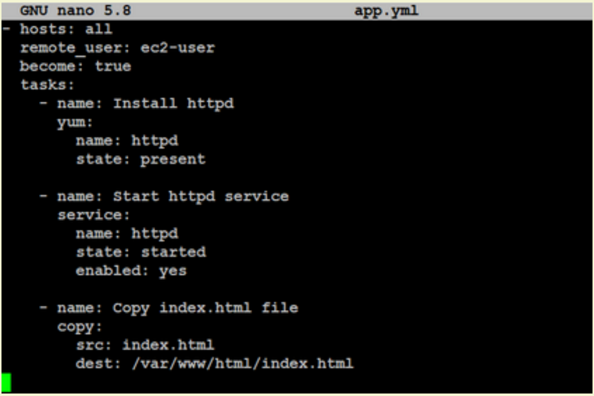
Ansible playbook

- An Ansible playbook is a YAML script used to automate system configurations, deployments, and management tasks.
- it is also used to execute multiple tasks.


playbook
hosts: all
remote_user:ec2_user
become: true
tasks
-name: install httpd
yum
name:httpd
state:installed
we can also give ip address of the each slave machine
user name
root user
we can use multiple task here
write your own message that will be displayed on the slave machine.
module
tasks

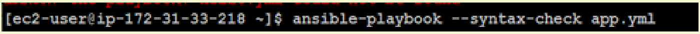
playbook:



```
- hosts: all
  remote_user: ec2-user
  become: true
  tasks:
    - name: Install httpd
      yum:
        name: httpd
        state: present

    - name: Start httpd service
      service:
        name: httpd
        state: started
        enabled: yes

    - name: Copy index.html file
      copy:
        src: index.html
        dest: /var/www/html/index.html
```

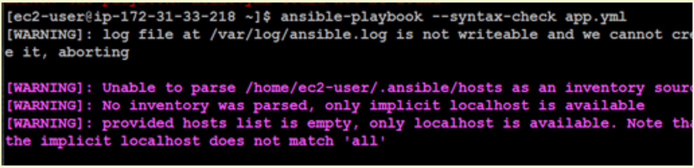
Verify and correct the syntax of the Ansible playbook

ansible-playbook --syntax-check app.yml- The ansible-playbook --syntax-check app.yml command validates YAML syntax without executing the playbook.



```
[ec2-user@ip-172-31-33-218 ~]$ ansible-playbook --syntax-check app.yml
```

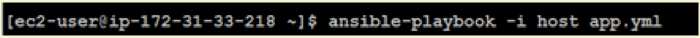
output:



```
[ec2-user@ip-172-31-33-218 ~]$ ansible-playbook --syntax-check app.yml
[WARNING]: log file at /var/log/ansible.log is not writeable and we cannot cre
e it, aborting

[WARNING]: Unable to parse /home/ec2-user/.ansible/hosts as an inventory sour
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note tha
the implicit localhost does not match 'all'
```

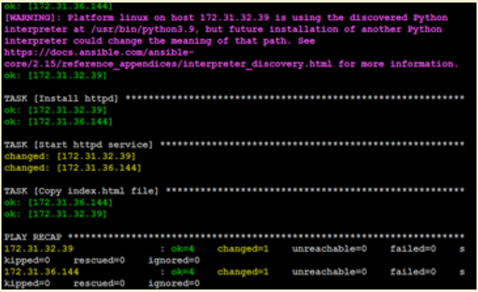
To execute the Ansible playbook:

ansible-playbook -i host app.yml- The command runs the app.yml playbook using the specified host inventory for target machines.



```
[ec2-user@ip-172-31-33-218 ~]$ ansible-playbook -i host app.yml
```

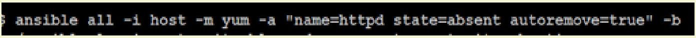
output:



```
ok: [172.31.36.144]
[WARNING]: Platform linux on host 172.31.32.39 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/2.15/reference_appendices/interpreter_discovery.html for more information.
ok: [172.31.32.39]

TASK [Install httpd] ***********************************************************
ok: [172.31.32.39]
ok: [172.31.36.144]

TASK [Start httpd service] *****************************************************
changed: [172.31.32.39]
changed: [172.31.36.144]

TASK [Copy index.html file] ****************************************************
ok: [172.31.36.144]
ok: [172.31.32.39]

PLAY RECAP *********************************************************************
172.31.32.39               : ok=4    changed=1    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
172.31.36.144              : ok=4    changed=1    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
```

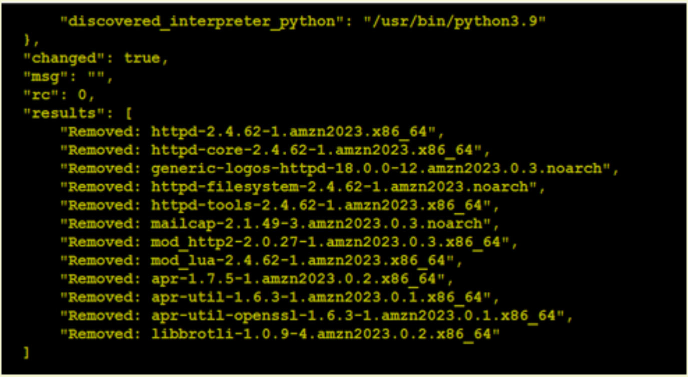
To uninstall software using an Ansible playbook:

ansible all -i host -m yum -a “name=httpd state=absent autoremove=true” -b - It removes httpd package from all hosts using yum with autoremove, running with become privileges.



```
$ ansible all -i host -m yum -a "name=httpd state=absent autoremove=true" -b
```

output:



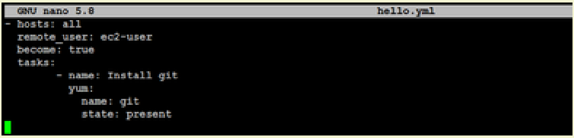
```
        "discovered_interpreter_python": "/usr/bin/python3.9"
    },
    "changed": true,
    "msg": "",
    "rc": 0,
    "results": [
        "Removed: httpd-2.4.62-1.amzn2023.x86_64",
        "Removed: httpd-core-2.4.62-1.amzn2023.x86_64",
        "Removed: generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch",
        "Removed: httpd-filesystem-2.4.62-1.amzn2023.noarch",
        "Removed: httpd-tools-2.4.62-1.amzn2023.x86_64",
        "Removed: mailcap-2.1.49-3.amzn2023.0.3.noarch",
        "Removed: mod_http2-2.0.27-1.amzn2023.0.3.x86_64",
        "Removed: mod_lua-2.4.62-1.amzn2023.x86_64",
        "Removed: apr-1.7.5-1.amzn2023.0.2.x86_64",
        "Removed: apr-util-1.6.3-1.amzn2023.0.1.x86_64",
        "Removed: apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64",
        "Removed: libbrotli-1.0.9-4.amzn2023.0.2.x86_64"
    ]
```

• project 1

To install git software in playbook:

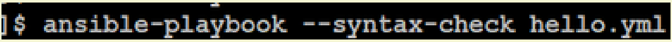
steps:

1.Create an Ansible playbook named hello.yml



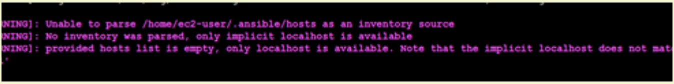
```
GNU nano 5.8                                          hello.yml
- hosts: all
  remote_user: ec2-user
  become: true
  tasks:
        - name: Install git
          yum:
            name: git
            state: present
```

2.To check the syntax of ansible playbook



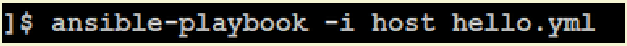
```
]$ ansible-playbook --syntax-check hello.yml
```

output:



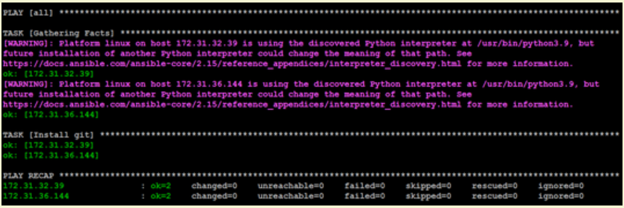
```
NING]: Unable to parse /home/ec2-user/.ansible/hosts as an inventory source
NING]: No inventory was parsed, only implicit localhost is available
NING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not mat
.'
```

3.To execute the playbook:



```
]$ ansible-playbook -i host hello.yml
```

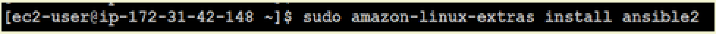
output:



```
PLAY [all] *************************************************************************************************

TASK [Gathering Facts] *************************************************************************************
[WARNING]: Platform linux on host 172.31.32.39 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/2.15/reference_appendices/interpreter_discovery.html for more information.
ok: [172.31.32.39]
[WARNING]: Platform linux on host 172.31.36.144 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/2.15/reference_appendices/interpreter_discovery.html for more information.
ok: [172.31.36.144]

TASK [Install git] *****************************************************************************************
ok: [172.31.32.39]
ok: [172.31.36.144]

PLAY RECAP *************************************************************************************************
172.31.32.39               : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.31.36.144              : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

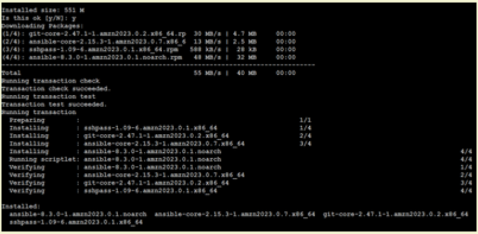
How to install ansible without pip:

1.sudo amazon-linux-extras install ansible2 - This command installs Ansible 2 on Amazon Linux using the amazon-linux-extras repository with sudo.

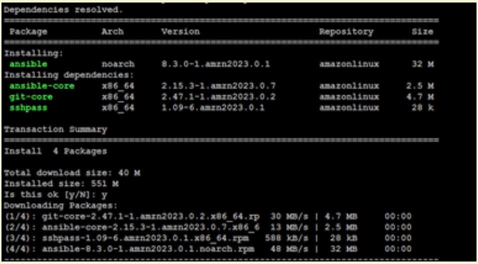


```
[ec2-user@ip-172-31-42-148 ~]$ sudo amazon-linux-extras install ansible2
```

output:



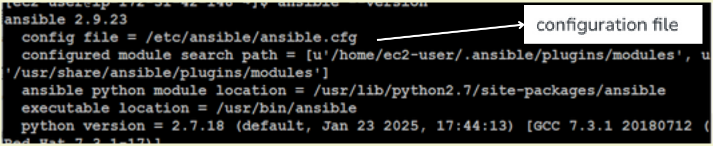
```
Installed size: 551 M
Is this ok [y/N]: y
Downloading Packages:
(1/4): git-core-2.47.1-1.amzn2023.0.2.x86_64.rp  30 MB/s | 4.7 MB     00:00
(2/4): ansible-core-2.15.3-1.amzn2023.0.7.x86_6  13 MB/s | 2.5 MB     00:00
(3/4): sshpass-1.09-6.amzn2023.0.1.x86_64.rpm   588 kB/s |  28 kB     00:00
(4/4): ansible-8.3.0-1.amzn2023.0.1.noarch.rpm   48 MB/s |  32 MB     00:00
--------------------------------------------------------------------------------
Total                                            55 MB/s |  40 MB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Installing       : sshpass-1.09-6.amzn2023.0.1.x86_64                     1/4
  Installing       : git-core-2.47.1-1.amzn2023.0.2.x86_64                  2/4
  Installing       : ansible-core-2.15.3-1.amzn2023.0.7.x86_64              3/4
  Installing       : ansible-8.3.0-1.amzn2023.0.1.noarch                                                    4/4
  Running scriptlet: ansible-8.3.0-1.amzn2023.0.1.noarch                                                    4/4
  Verifying        : ansible-8.3.0-1.amzn2023.0.1.noarch                                                    1/4
  Verifying        : ansible-core-2.15.3-1.amzn2023.0.7.x86_64                                              2/4
  Verifying        : git-core-2.47.1-1.amzn2023.0.2.x86_64                                                  3/4
  Verifying        : sshpass-1.09-6.amzn2023.0.1.x86_64                                                     4/4

Installed:
  ansible-8.3.0-1.amzn2023.0.1.noarch  ansible-core-2.15.3-1.amzn2023.0.7.x86_64  git-core-2.47.1-1.amzn2023.0.2.x86_64
  sshpass-1.09-6.amzn2023.0.1.x86_64
```



```
Dependencies resolved.
================================================================================
 Package          Arch        Version                     Repository       Size
================================================================================
Installing:
 ansible          noarch      8.3.0-1.amzn2023.0.1        amazonlinux      32 M
Installing dependencies:
 ansible-core     x86_64      2.15.3-1.amzn2023.0.7       amazonlinux     2.5 M
 git-core         x86_64      2.47.1-1.amzn2023.0.2       amazonlinux     4.7 M
 sshpass          x86_64      1.09-6.amzn2023.0.1         amazonlinux      28 k

Transaction Summary
================================================================================
Install  4 Packages

Total download size: 40 M
Installed size: 551 M
Is this ok [y/N]: y
Downloading Packages:
(1/4): git-core-2.47.1-1.amzn2023.0.2.x86_64.rp  30 MB/s | 4.7 MB     00:00
(2/4): ansible-core-2.15.3-1.amzn2023.0.7.x86_6  13 MB/s | 2.5 MB     00:00
(3/4): sshpass-1.09-6.amzn2023.0.1.x86_64.rpm   588 kB/s |  28 kB     00:00
(4/4): ansible-8.3.0-1.amzn2023.0.1.noarch.rpm   48 MB/s |  32 MB     00:00
--------------------------------------------------------------------------------
```

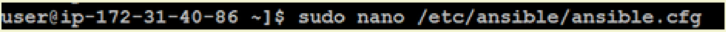
2.ansible --version - We can check the Ansible version and the file that already contains the configuration.

configuration file:



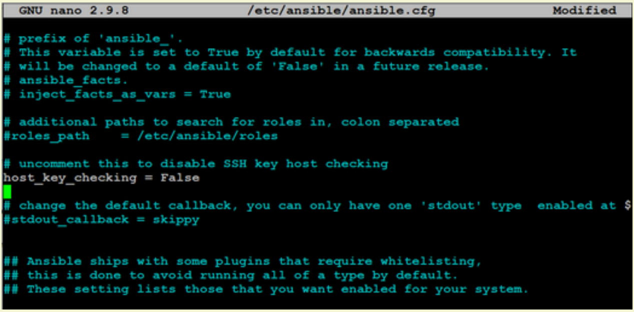
```
[ec2-user@ip-172-31-42-146 ~]$ ansible --version
ansible 2.9.23
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/home/ec2-user/.ansible/plugins/modules', u
'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python2.7/site-packages/ansible
  executable location = /usr/bin/ansible
  python version = 2.7.18 (default, Jan 23 2025, 17:44:13) [GCC 7.3.1 20180712 (
Red Hat 7.3.1-17)]
```

3.sudo nano /etc/ansible/ansible.cfg - This command opens Ansible's configuration file in the Nano editor with superuser privileges for editing.



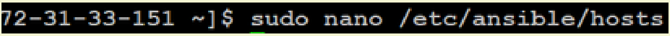
```
user@ip-172-31-40-86 ~]$ sudo nano /etc/ansible/ansible.cfg
```

"Change the setting in the configuration file to host_key_checking = False so that we don’t have to specify the inventory file each time."

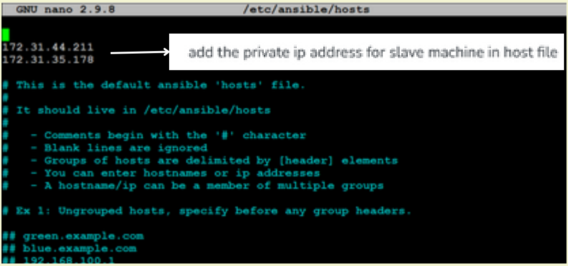


```
  GNU nano 2.9.8                /etc/ansible/ansible.cfg                  Modified

# prefix of 'ansible_'.
# This variable is set to True by default for backwards compatibility. It
# will be changed to a default of 'False' in a future release.
# ansible_facts.
# inject_facts_as_vars = True

# additional paths to search for roles in, colon separated
#roles_path    = /etc/ansible/roles

# uncomment this to disable SSH key host checking
host_key_checking = False

# change the default callback, you can only have one 'stdout' type  enabled at $
#stdout_callback = skippy


## Ansible ships with some plugins that require whitelisting,
## this is done to avoid running all of a type by default.
## These setting lists those that you want enabled for your system.
```

4.sudo nano /etc/ansible/hosts - This command opens the Ansible inventory file (/etc/ansible/hosts) in the Nano editor with superuser privileges,



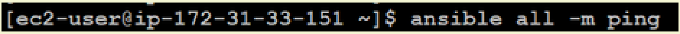
```
72-31-33-151 ~]$ sudo nano /etc/ansible/hosts
```



```
  GNU nano 2.9.8                    /etc/ansible/hosts

172.31.44.211
172.31.35.178

# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
#   - Comments begin with the '#' character
#   - Blank lines are ignored
#   - Groups of hosts are delimited by [header] elements
#   - You can enter hostnames or ip addresses
#   - A hostname/ip can be a member of multiple groups

# Ex 1: Ungrouped hosts, specify before any group headers.

## green.example.com
## blue.example.com
## 192.168.100.1
```

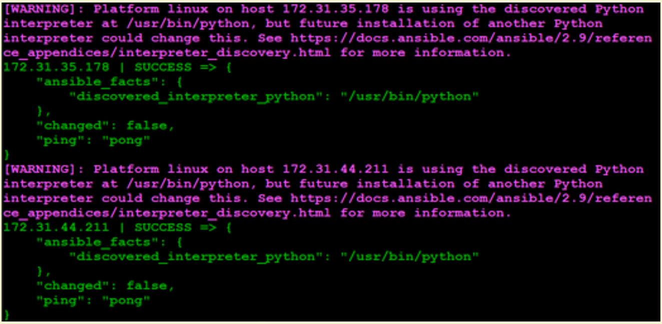
add the private ip address for slave machine in host file

5.ansible all -i host -m ping - used to check connectivity to all hosts listed in the inventory file.



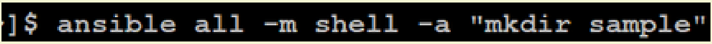
```
[ec2-user@ip-172-31-33-151 ~]$ ansible all -m ping
```

output:



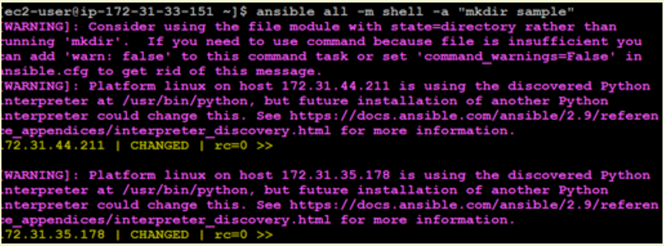
```
[WARNING]: Platform linux on host 172.31.35.178 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
172.31.35.178 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Platform linux on host 172.31.44.211 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
172.31.44.211 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
```

6.ansible all -i host -m shell -a “mkdir sample” - This Ansible command runs the mkdir sample shell command on all hosts in the host inventory, creating a directory named "sample".



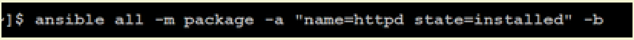
```
]$ ansible all -m shell -a "mkdir sample"
```

output:



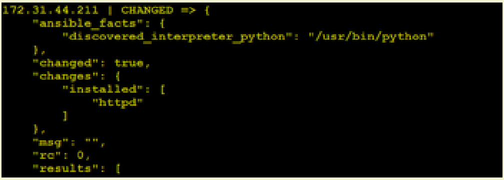
```
[ec2-user@ip-172-31-33-151 ~]$ ansible all -m shell -a "mkdir sample"
[WARNING]: Consider using the file module with state=directory rather than
running 'mkdir'.  If you need to use command because file is insufficient you
can add 'warn: false' to this command task or set 'command_warnings=False' in
ansible.cfg to get rid of this message.
[WARNING]: Platform linux on host 172.31.44.211 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
172.31.44.211 | CHANGED | rc=0 >>

[WARNING]: Platform linux on host 172.31.35.178 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
172.31.35.178 | CHANGED | rc=0 >>
```

7.ansible all -m package -a name=httpd state=installed “-b -install software in slave machine using package module

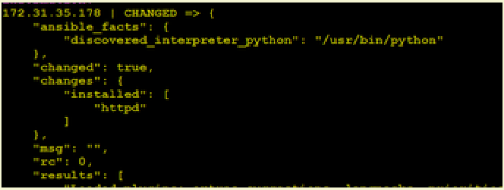


```
]$ ansible all -m package -a "name=httpd state=installed" -b
```

output:



```
172.31.44.211 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": true,
    "changes": {
        "installed": [
            "httpd"
        ]
    },
    "msg": "",
    "rc": 0,
    "results": [
```



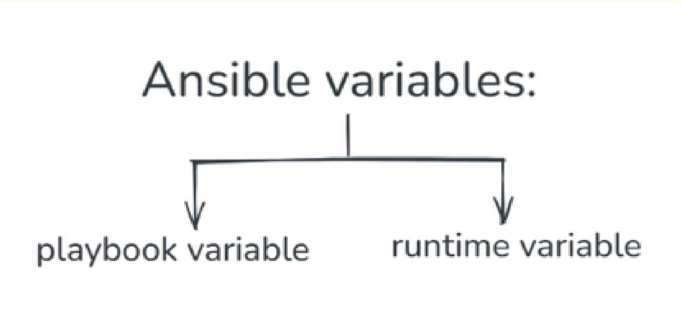
```
172.31.35.178 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": true,
    "changes": {
        "installed": [
            "httpd"
        ]
    },
    "msg": "",
    "rc": 0,
    "results": [
```

Ansible variables

- Ansible variables control automation, including playbook, inventory, role, extra-vars, facts, environment, and registered values.

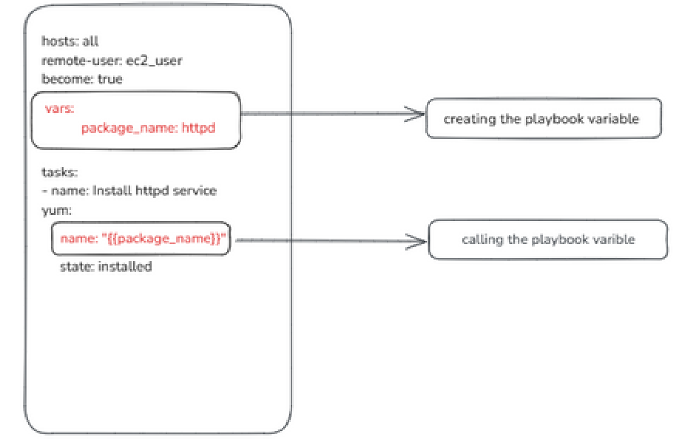
- In Ansible, variables are used to store values that can be referenced throughout playbooks, tasks, and roles.

- They allow for dynamic configurations and make playbooks reusable and flexible.



Playbook variables:

- Playbook variables in Ansible are variables defined directly within a playbook to store dynamic values.
- They help customize tasks and make playbooks reusable.
- In Ansible, variables allow you to store and reuse values dynamically, making playbooks more flexible and scalable.
- Variables can store information such as file paths, usernames, IP addresses, and configuration settings.

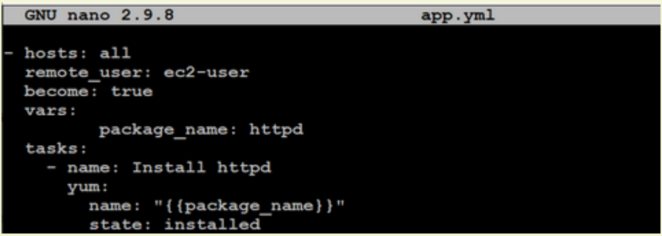


```
hosts: all
remote-user: ec2_user
become: true
vars:
        package_name: httpd

tasks:
- name: Install httpd service
yum:
    name: "{{package_name}}"
    state: installed
```

creating the playbook variable

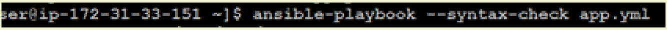
calling the playbook varible

1.create a playbook:



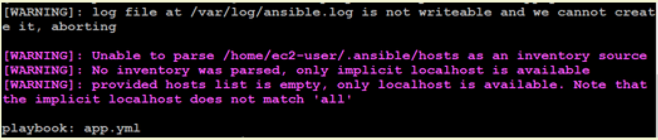
```
  GNU nano 2.9.8                                    app.yml

- hosts: all
  remote_user: ec2-user
  become: true
  vars:
           package_name: httpd
  tasks:
     - name: Install httpd
       yum:
          name: "{{package_name}}"
          state: installed
```

2.To check the syntax of ansible playbook



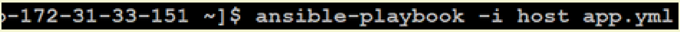
```
ser@ip-172-31-33-151 ~]$ ansible-playbook --syntax-check app.yml
```

output:



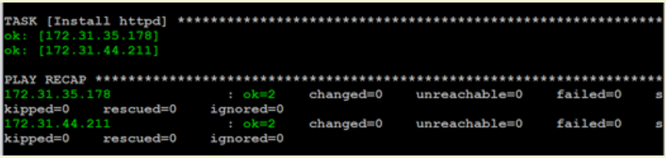
```
[WARNING]: log file at /var/log/ansible.log is not writeable and we cannot creat
e it, aborting

[WARNING]: Unable to parse /home/ec2-user/.ansible/hosts as an inventory source
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that
the implicit localhost does not match 'all'

playbook: app.yml
```

3.To execute the playbook:



```
-172-31-33-151 ~]$ ansible-playbook -i host app.yml
```

output:

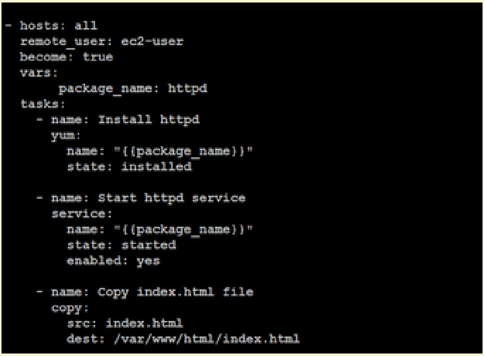


```
TASK [Install httpd] ***********************************************************
ok: [172.31.35.178]
ok: [172.31.44.211]

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

project 2

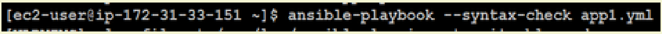
To deploy the index.html file on the slave machine

1.create a playbook:

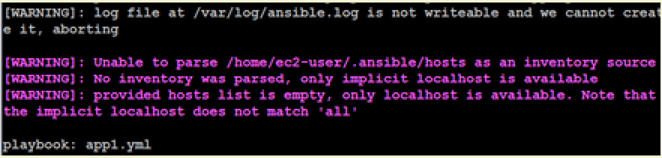


```
- hosts: all
  remote_user: ec2-user
  become: true
  vars:
       package_name: httpd
  tasks:
    - name: Install httpd
      yum:
        name: "{{package_name}}"
        state: installed

    - name: Start httpd service
      service:
        name: "{{package_name}}"
        state: started
        enabled: yes

    - name: Copy index.html file
      copy:
        src: index.html
        dest: /var/www/html/index.html
```

2.To check the syntax of ansible playbook

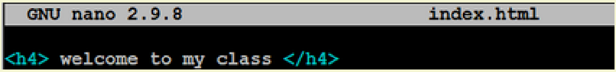


```
[ec2-user@ip-172-31-33-151 ~]$ ansible-playbook --syntax-check app1.yml
```



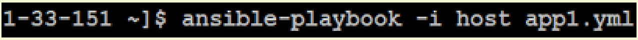
```
[WARNING]: log file at /var/log/ansible.log is not writeable and we cannot creat
e it, aborting

[WARNING]: Unable to parse /home/ec2-user/.ansible/hosts as an inventory source
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that
the implicit localhost does not match 'all'

playbook: app1.yml
```

3.Create the index.html:



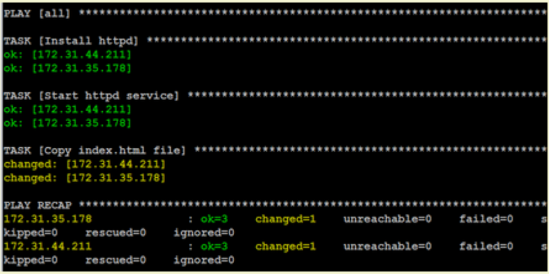
```
GNU nano 2.9.8                    index.html

<h4> welcome to my class </h4>
```

4.To execute the playbook:

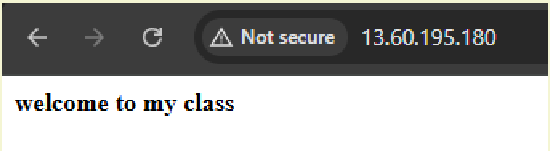


```
1-33-151 ~]$ ansible-playbook -i host appl.yml
```

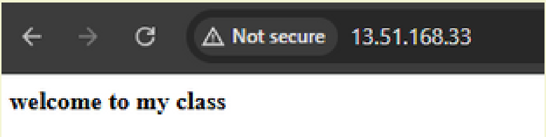
output:



```
PLAY [all] *************************************************************

TASK [Install httpd] ***************************************************
ok: [172.31.44.211]
ok: [172.31.35.178]

TASK [Start httpd service] *********************************************
ok: [172.31.44.211]
ok: [172.31.35.178]

TASK [Copy index.html file] ********************************************
changed: [172.31.44.211]
changed: [172.31.35.178]

PLAY RECAP *************************************************************
172.31.35.178              : ok=3    changed=1    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
172.31.44.211              : ok=3    changed=1    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
```



welcome to my class

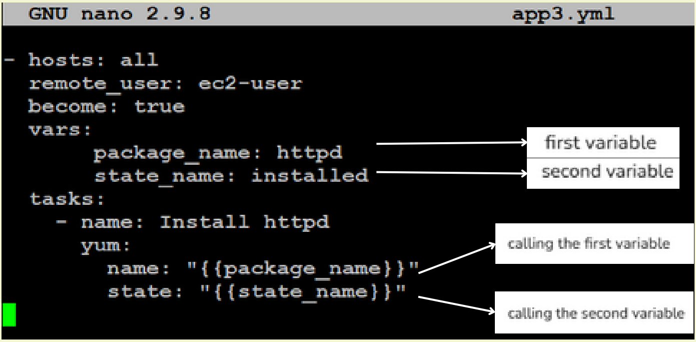


welcome to my class

how to use more than one variable in playbook

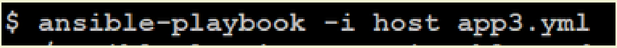
variables

1.Create the playbook :



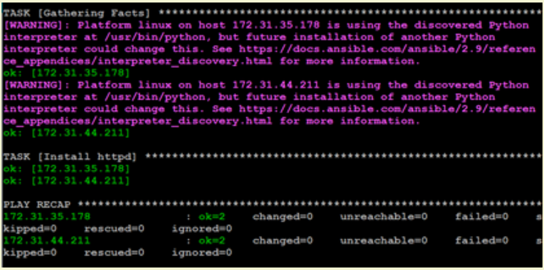
```
  GNU nano 2.9.8                           app3.yml

- hosts: all
  remote_user: ec2-user
  become: true
  vars:
       package_name: httpd
       state_name: installed
  tasks:
    - name: Install httpd
      yum:
        name: "{{package_name}}"
        state: "{{state_name}}"
```

2.To execute the playbook:



```
$ ansible-playbook -i host app3.yml
```

output:

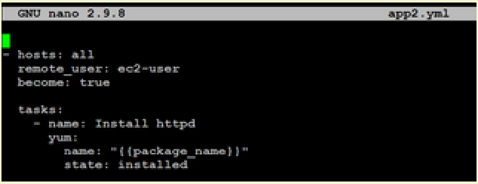


```
TASK [Gathering Facts] **********************************************************
[WARNING]: Platform linux on host 172.31.35.178 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
ok: [172.31.35.178]
[WARNING]: Platform linux on host 172.31.44.211 is using the discovered Python
interpreter at /usr/bin/python, but future installation of another Python
interpreter could change this. See https://docs.ansible.com/ansible/2.9/referen
ce_appendices/interpreter_discovery.html for more information.
ok: [172.31.44.211]

TASK [Install httpd] ************************************************************
ok: [172.31.35.178]
ok: [172.31.44.211]

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

Runtime Variable

- When a playbook is running, you can declare variables
- Runtime variables in Ansible are variables that are defined or provided when executing the playbook, rather than being pre-defined in the playbook or inventory files.

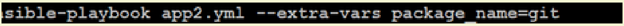
1.Create the playbook :



```
  GNU nano 2.9.8                                          app2.yml

- hosts: all
  remote_user: ec2-user
  become: true

  tasks:
    - name: Install httpd
      yum:
        name: "{{package_name}}"
        state: installed
```

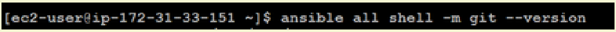
2.To execute the playbook:

Executing the Git software at runtime.



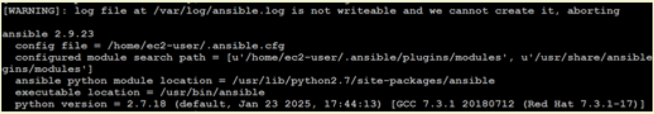
```
sible-playbook app2.yml --extra-vars package_name=git
```

3.To check whether Ansible is installed on the slave machine or not.



```
[ec2-user@ip-172-31-33-151 ~]$ ansible all shell -m git --version
```

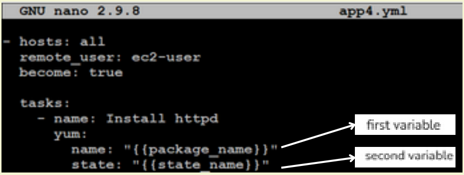
output:



```
[WARNING]: log file at /var/log/ansible.log is not writeable and we cannot create it, aborting

ansible 2.9.23
  config file = /home/ec2-user/.ansible.cfg
  configured module search path = [u'/home/ec2-user/.ansible/plugins/modules', u'/usr/share/ansible
gins/modules']
  ansible python module location = /usr/lib/python2.7/site-packages/ansible
  executable location = /usr/bin/ansible
  python version = 2.7.18 (default, Jan 23 2025, 17:44:13) [GCC 7.3.1 20180712 (Red Hat 7.3.1-17)]
```

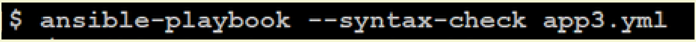
how to use more than one variable in playbook

1.Create the playbook :



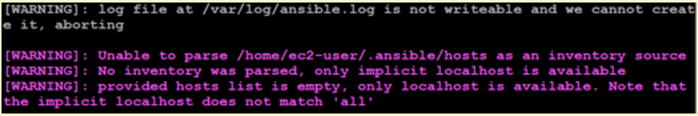
```
  GNU nano 2.9.8                          app4.yml

- hosts: all
  remote_user: ec2-user
  become: true

  tasks:
    - name: Install httpd
      yum:
        name: "{{package_name}}"
        state: "{{state_name}}"
```

2.To check the syntax of ansible playbook



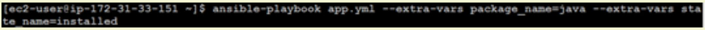
```
$ ansible-playbook --syntax-check app3.yml
```

output:



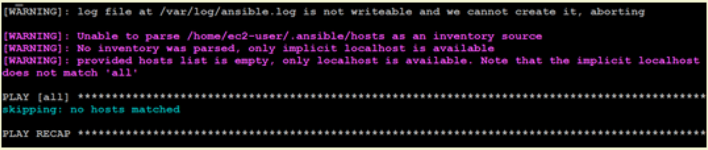
```
[WARNING]: log file at /var/log/ansible.log is not writeable and we cannot creat
e it, aborting

[WARNING]: Unable to parse /home/ec2-user/.ansible/hosts as an inventory source
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that
the implicit localhost does not match 'all'
```

3.To execute the playbook:



```
[ec2-user@ip-172-31-33-151 ~]$ ansible-playbook app.yml --extra-vars package_name=java --extra-vars sta
te_name=installed
```

output:



```
[WARNING]: log file at /var/log/ansible.log is not writeable and we cannot create it, aborting

[WARNING]: Unable to parse /home/ec2-user/.ansible/hosts as an inventory source
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost
does not match 'all'

PLAY [all] ***********************************************************************************************
skipping: no hosts matched

PLAY RECAP ***********************************************************************************************
```