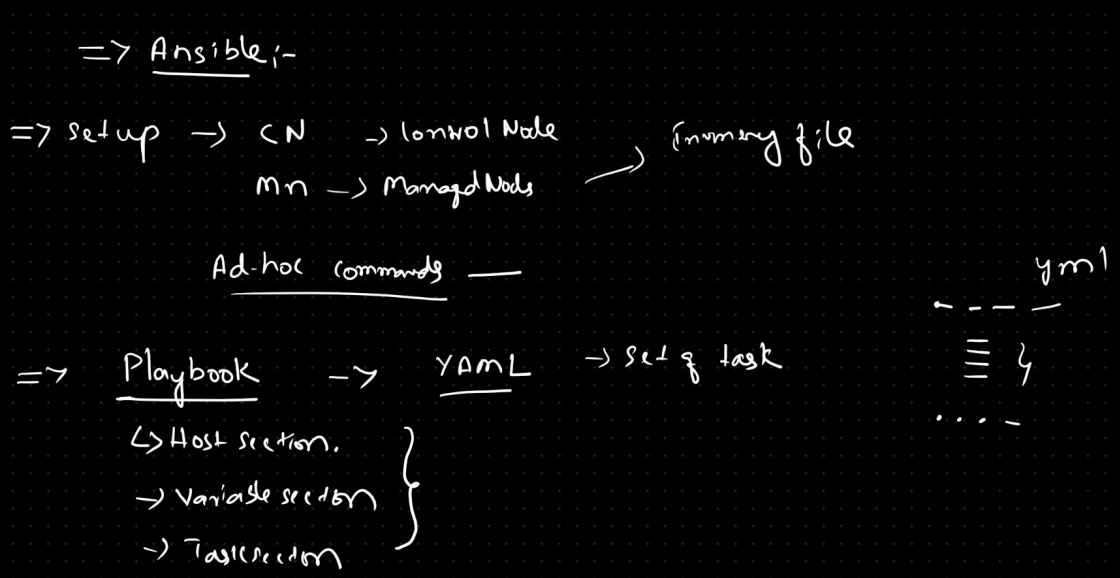


start all managed nodes and control node

get connected to control node(Ansible is installed here)

ansible user is already created
ansible directory created

switched to ansible user $ sudo su ansible

Get into ansible directory : cd ~ or cd ../ansible
$pwd
/home/ansible/

$ vi 01-pi

```
---
- hosts: all
  tasks:
    - name: Ping all managed nodes
      ping:
      remote_user: ansible
...
```

$ ansible-playbook 01-ping.yml

To get more info about the playbook execution or internal info about the playbook --> verbosity

$ ansible-playbook 01-ping.yml -v basic verbosity
$ ansible-playbook 01-ping.yml -vv more detailed output
$ ansible-playbook 01-ping.yml -vvv very very verbose extensive detail
$ ansible-playbook 01-ping.yml -vvvv extremely verbose

To check syntax of playbook

```
$ ansible-playbook 01-ping.yml --syntax-check
```

To check host info which host will be affected by a playbook

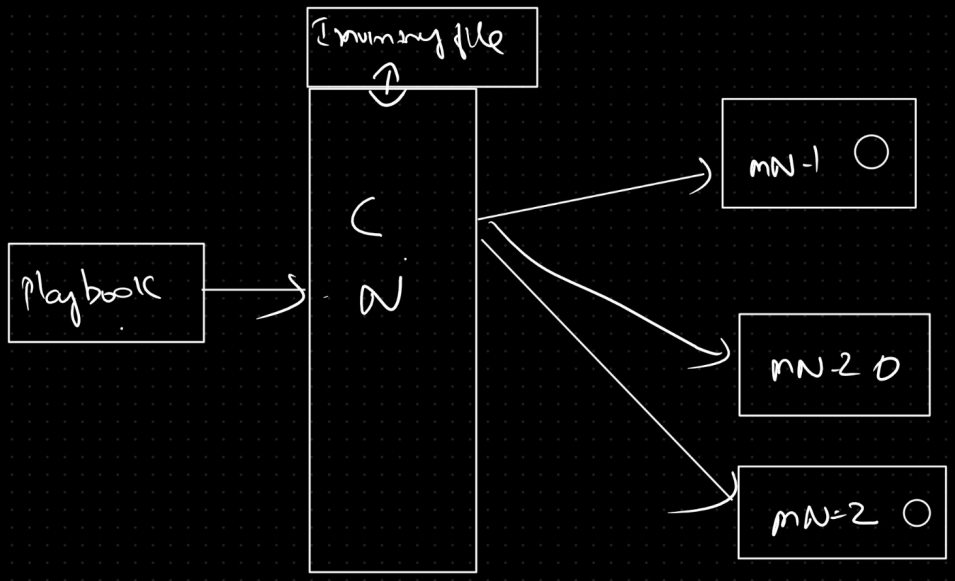
```
$ ansible-playbook 01-ping.yml --list-hosts
```

```
Execute one step at a time , conirm each task before running with yes or no or conitnue
```

```
$ ansible-playbook 01-ping.yml --step
```

Check what will happen if we execute our playbook or dry run

```
$ ansible-playbook 01-ping.yml --check
```



Create files in managed nodes using ansible
==========
create a new yml playbook file in control node
$ vi 02-create-file.yml

```
---
- hosts: all
  tasks:
  - name: Create file in all managed node
    file:
      path: /home/ansible/alien.txt
      state: touch
...
```

ansible-playbook 02-create-file.yml

Later get connected to managed nodes and check if files are created

Install git in client(managed nodes) --> Assignment to install maven
==========

Install git in client (Managed Nodes)
Ans :
vi 03-install-git.yml

```
---
- hosts: all
  become: true
 tasks:
  - name: Installing Git in all Managed Nodes
    yum:
      name: git
      state: latest
...
```

ansible-playbook 03-install-git.yml --syntax-check
ansible-playbook 03-install-git.yml

Hosting static website only in webservers group i.e host is only webserver group which we created but not in all mn
==========

$ vi 04-website-hosting.yml

```
---
- hosts: webservers
  become: true
  tasks:
  - name: install httpd package
    yum:
      name: httpd
      state: latest
  - name: copy index.html file
    copy:
      src: index.html
      dest: /var/www/html/index.html
  - name: start httpd server
    service:
      name: httpd
      state: started
...
```

$ ansible-playbook 04-website-hosting.yml --list-hosts

$ ansible-playbook 04-website-hosting.yml --syntax-check

$ ansible-playbook 04-website-hosting.yml

Use public ip of webservers group machine and check if website is hosted

Variables :
Used to store data /info
id: 1
name: rohan

We can use concept of variables in 4 ways

1) Runtime Variable

2) Playbook variable

3) Group Variable

4) Host Variable

1) Runtime Variable --> We can pass variable value at runtime

```
---
- hosts: webservers
  become: true
  tasks:
  - name: "{{package_name}}"
    yum:
      name: httpd
      state: latest
  - name: copy index.html file
    copy:
      src: index.html
      dest: /var/www/html/index.html
  - name: "{{package_name}}"
    service:
      name: httpd
      state: started
...
```

$ ansible-playbook <yml file name> --extra-vars package_name=httpd

2) Playbook variable --> Declare and use the variables within playbook

```
---
- hosts: webservers
  become: true
  vars:
   package_name: httpd webserver
  tasks:
  - name: "{{package_name}}"
    yum:
      name: httpd
      state: latest
  - name: copy index.html file
    copy:
      src: index.html
      dest: /var/www/html/index.html
  - name: "{{package_name}}"
    service:
      name: httpd
      state: started
...
```

$ ansible-playbook <yml file name>

assignment: Write ansible playbook to install Java in webserver group and MySQL in dbserver group

Group Variable : Group variable file should be created at host inventory location

Host Inventory : /etc/ansible/hosts

$ mkdir /etc/ansible/group_vars

$ sudi vi /etc/ansible/group_vars/webservers.yml
 package_name: java

$ sudo vi /etc/ansible/group_vars/dbservers.yml
 package_name: mysql

Host vars: Sever/machine specific variables --> for ever host if we want specific variable we can go with host variables

$ mkdir /etc/ansible/host_vars
--> Create a file with host name or ip

$ sudo vi /etc/ansible/host_vars/hostname.yml --> sudo vi /etc/ansible/host_vars/172.130.0.1.yml

Handlers and Tags
==========

==> In Playbook all the tasks by default will be executed in sequential order

Using Handlers we can execute tasks based on other task status
(If 2nd task status is changed only then the 3rd task is executed)
It will notify the tasks to execute and we use 'notify' keyword to inform handler to execute

Using Tag we can map task to a tag name and we can execute particular task and we can skip particular task also

i 05.handlers.yml

```
---
- hosts: webservers
  become: true
  vars:
    package-name: httpd

  tasks:
  - name: installing httpd
      yum
       name: "{{package-name}}"
       state: latest
     tags:
       -install
  - name: copy index.html
    copy:
      src: index.html
      dest: /var/www/html/index.html
   tags:
     - copy
   notify:
     starting httpd
  - name: starting httpd
    service:
      name: "{{package-name}}"
      state: started
...
```

```
ansible-playbook 05-handlers.yml --syntax-check
ansible-playbook 05-handlers.yml --list-tags
ansible-playbook 05-handlers.yml --tags "copy"  // execute the copy tags
ansible-playbook 05-handlers.yml --skip-tags "install,copy"  // skip the particular tags and executes the remaining portion of playbook.
```

Ansible Vault
========

To secure our playbooks we go with the concept of Ansible Vault --> We encrypt and Decrypt playbooks

$ ansible-vault encrypt <playbook.yml> --> To encrypt our yml

$ ansible-vault decrypt <playbook.yml> --> To decrypt our yml

$ ansible-vault edit <playbook.yml> --> To edit our yml

$ ansible-vault view encrypt <playbook.yml> --> To see original data from our playbook yml