DAY1

launch 2 instances using ubuntu ami
connect to mobax
manager:

```
sudo apt update
sudo apt install software-properties-common
sudo add-apt-repository --yes --update ppa:ansible/ansible
sudo apt install ansible
ansible --version
```

vi ansible path
copy slave private to ansible path hosts

```
ssh-keygen
cd .ssh/
cat id_pub
```

and launch node instance

```
ssh-keygen
cd.ssh/
vi authorizedkeys
```

copy pubkey in master and paste in authorizedkeys
connect to node ssh privateip

```
vim /etc/ansible/hosts
vim /etc/ansible/ansible.cfg
```

remove yash sysmbol in ineventory and sudo user

```
ansible all -a "touch file1.txt"
ansible all -a "ls"
ansible all -a "useradd nagesh"
ansible all -a "cat /etc/passwd"
ansible all -ba "apt install tree -y"
ansible all -ba "tree --version"
ansible all -ba "apt remove tree -y"
ansible all -m "shell" -a "touch devopsclass"
```

DAY2

```
ansible all -m file -a "dest=/root/aws state=directory"
ansible all -m file -a "dest=/root/aws/aws.txt state=touch"
ansible all -m file -a "dest=/root/aws state=absent"
ansible all -m copy -a "src=/root/hello.txt dest=/root/hello.txt"
ansible all -m shell -a "ifconfig | grep -i 'inet '"
ansible all -m shell -a "mkdir /root/aws"
ansible all -m apt -a "name=wget state=absent"
ansible all -m user -a "name=Nagesh"
```

```
vi task.yaml
---
- hosts: all
  become: yes


  tasks:
  - name: creating data folder on remote machine
    file: dest=/root/devops state=directory

  - name: creating data folder on remote machine
    file: dest=/root/devops/devops.txt state=touch

  - name: creating data folder on remote machine
    copy: src=/root/test_file.txt dest=/root/devops/

---
- hosts: all
  become: yes


  tasks:
  - name: installing nginx pkg on remote machine
    apt: pkg=nginx state=latest


---
- hosts: all
  become: yes


  tasks:
  - name: installing nginx pkg on remote machine
    apt: pkg={{ item }} state=latest
    with items:
    - nginx
    - httpd

---
- hosts: all
  become: yes
  tasks:
  - name: installing packages
    apt: pkg={{ item }} state=latest
    with_items:
    - nginx
```

DAY3

```
vi lampstack.yaml
```

```
---
- name: Installing LAMP stack
  hosts: all
  become: yes

  tasks:
  - name: Run apt-get update
    apt:
      update_cache: yes

  - name: installing apache
    apt:
      name: apache2
      state: latest

  - name: enable apache service
    service:
      name: apache2
      enabled: yes
      state: started

  - name: delete default welcome page
    file:
      path: /var/www/html/index.html
      state: absent
    notify:
      - restart apache2 service

  - name: install mysql db community edition
    apt:
      name: mariadb-server
      state: latest
  notify:
   name: restart mysql service

 - name: enable mysql service
  service:
   name: mariadb
   enabled: yes
   state: started

 - name: install php
  apt:
   name:
   - php
   - php-mysql
   state: latest

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

```
vi index.php
```

```
<html>
  <head>
    <title> Ansible Application </title>
  </head>

<body>
  <?php
  echo "welcome to Agasthya solutions"
  ?>
</body>
</html>
```

```
vi handlers.yaml
```

```
---
- name: Installing LAMP stack
 hosts: all
 become: yes

 tasks:
```

```
- name: Run apt-get update
  apt:
    update_cache: yes

- name: installing apache
  apt:
    name: apache2
    state: latest

- name: enable apache service
  service:
    name: apache2
    enabled: yes
    state: started

- name: delete default welcome page
  file:
    path: /var/www/html/index.html
    state: absent
  notify:
    - restart apache2 service

- name: install mysql db community edition
  apt:
    name: mariadb-server
    state: latest
  notify:
    name: restart mysql service

- name: enable mysql service
  service:
    name: mariadb
    enabled: yes
    state: started

- name: install php
  apt:
    name:
    - php
    - php-mysql
    state: latest

- name: copy php file
```

```
    copy:
      src: /root/index.php
      dest: /var/www/html/index.php
      force: yes
    notify:
      - restart apache2 service
      - restart mysql service

  handlers:
  - name: restart apache2 service
    service:
      name: apache2
      state: restarted

  - name: restart mysql service
    service:
      name: mariadb-server
      state: restarted
```

variable concept

```
---
- name: variable usecase
  hosts: all
  become: yes
  vars:
    CourseName: devops
    Technology: terraform

  tasks:
  - name: print vars data into text file
    shell: echo "I am learning {{ CourseName }} course and i liked {{ Technology }} most" > /root/vars1.txt
```

example2

```
vi variables.txt
CourseName: devops
Technology: terraform
vi vars_ex2.yaml
---
- name: variable usecase
  hosts: all
  become: yes
  vars_files:
  - variables.txt

  tasks:
  - name: print vars data into text file
    shell: echo "I am learning {{ CourseName }} course and i liked {{ Technology }} most" > /root/vars1.txt
```

example3

```
---
- name: variable usecase
  hosts: all
  become: yes
  tasks:
  - include_vars: variables.txt
  - name: print vars data into text file
    shell: echo "I am learning {{ CourseName }} course and i liked {{ Technology }} most" > /root/vars1.txt
```

example4

```
---
- name: variable usecase
  hosts: all
  become: yes
  vars:
    colours:
       - White
       - Black
       - Blue
  tasks:
  - name: variable example
    shell: echo "My favorite color is {{ Colors[0] }}, i dont like {{ Colors[1] }}, and my car color is {{ Colors[2] }}" > /root/vars6.txt
```

day4

```
---
- hosts: all
  become: yes
  tasks:
     - name: installing httpd in RedHat
       command: yum install httpd -y
       when: ansible_os_family == "RedHat"
```

```
   - name: installing apache2 on ubuntu
     command: apt install apache2 -y
     when: ansible_os_family == "Debian"


- hosts: all
  become: yes
  tasks:
   - name: adding users
     action: user name='{{item}}' state=present
     with_items:
           - nagesh
           - sai
           - subbu
           - mahesh
           - kranthi
```

LOOPS

```
---
- hosts: all
  become: yes
  tasks:
   - name: installing packages
     action: apt name='{{item}}' state=present
     with_items:
           - apache2
           - nginx
           - tree
```

```
---
- hosts: all
  become: yes
  tasks:
   - name: installing httpd in RedHat
     command: yum install httpd -y
     when: ansible_os_family == "RedHat"

   - name: installing httpd on ubuntu
     command: apt install apache2 -y
```

```
      when: ansible_os_family == "Debian"
```

VAULT

```
vi creds
user = Nagesh
password = Nagesh
cat creds
ansible-vault create creds
cat creds
ansible-vault edit creds
ansible-vault decrypt creds
ansible-vault encrypt creds
ansible-vault rekey creds
```

TAGS

```
vi Nagesh.yml

---
- hosts: all
  become: yes
  tasks:
    - name: installing git
      command: apt install git -y
      tags: a

    - name: installing java
      command: apt install default-jre
      tags: b

    - name: installing maven
      command : apt install maven -y
      tags: c

    - name: installing docker
      command: apt install docker.io
      tags: d

ansible-playbook Nagesh.yml --tags a,b
ansible-playbook Nagesh.yml --tags c
ansible-playbook Nagesh.yml --skip-tags "c"
```

ROLES

```
cd /etc/ansible/
ls
roles
cd roles/
ls
ansible-galaxy init install_apps
ls
cd install_apps/
ls
apt install tree -y
tree install_apps

mkdir roles
vim without role.yaml
---
- name: Without roles, a regular playbook
  hosts: all
  become: yes
  tasks:
    - name: Install apache2
      apt:
        name: apache2
        state: absent

    - name: Install mariadb-server
      apt:
        name: mariadb-server
        state: present

    - name: Install nginx
      apt:
        name: nginx
        state: present

ansible-playbook without role.yaml

mkdir roles
cd roles/
mkdir httpd-role
cd httpd-role/
```

```
mkdir tasks
cd tasks/
main.yaml
---
    - name: Install apache2
      apt:
        name: apache2
        state: present
cd ..
apt install tree -y
cd roles/
mkdir git-role
cd git-role/
mkdir tasks
cd tasks/
main.yaml
    - name: Install git
      apt:
        name: git
        state: installed
cd roles/
mkdir git-role
cd git-role/
mkdir tasks
cd tasks/
main.yaml
- name: Install docker
      apt:
        name: docker.io
        state: installed
cd
vi master.yaml
- name: master playbook
  hosts: all
  become: yes
  roles:
     - apache2-role
     - mariadb-role
     - nginx-role
ansible-playbook master.yaml
roles demo2
files
index.html
```

```
<h1> THis is a ansible roles demo page </h1>
cd handlers
  - name: installing apache
    apt:
      name: apache2
      state: latest
cd tasks

---
- name: Installing httpd as phase2
  apt:
    name: apache2
    state: latest
  notify: start apache2 service

- name: copy index.html to /var/www/html/ path
  copy:
    src: "index.html"
    dest: "{{ index_html_dest }}"
cd vars
---
index_html_dest: "/var/www/html"

index.html
```