# Automating a Multi-Tier Application on VMware with Ansible

## **Objective:**

Using Ansible to fully automate the deployment, configuration, and orchestration of a multi-tier application (e.g., LAMP stack) across multiple CentOS VMs (master node, node1, and node2) in a VMware environment.

# Steps:

## 1. Provisioning VMs:

- Use Ansible to deploy and configure two nodes (node1 and node2) on VMware.
- Automate the installation of necessary dependencies like Apache on one node and MySQL on the other.

## 2. Role-based Deployment:

- Use Ansible roles to separate the tasks:
  - Web server role (for node1 to host Apache/PHP).
  - Database server role (for node2 to host MySQL/MariaDB).
- Deploy a simple PHP application on the web server node and configure it to communicate with the database on node2.

# 3. Network Configuration & Load Balancing:

 Automate firewall rules, IP tables, and networking configurations between the nodes.

## 4. Backup and Recovery Automation:

- o Automate backups of the database on node2 using Ansible scheduled jobs.
- Automate web content backup on node1.

# 5. Security:

- Implement Ansible Vault to manage secrets such as database credentials.
- Use Ansible to apply security updates and harden the systems.

#### Outcome:

- A fully automated infrastructure with multiple VMs on VMware, deploying a multi-tier application.
- Centralized monitoring and logging for system health and performance.

• Secure and efficient infrastructure with automated backups and firewall configurations.

# **Define the Inventory, Setup and Provisioning VMs**

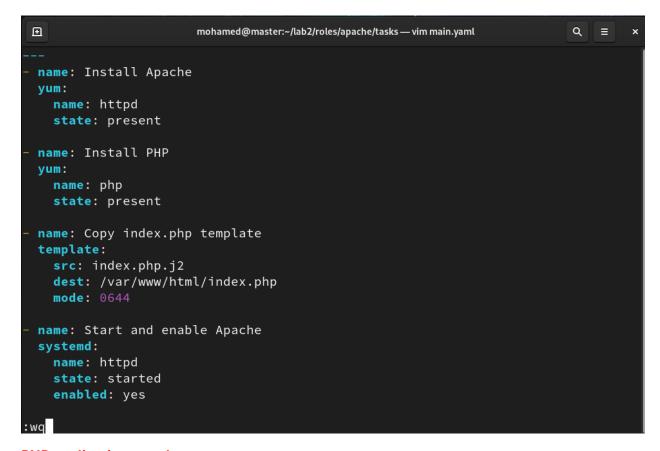
```
mohamed@master:~/lab2 — vim ansible.cfg
                                                                    Q ≡
[defaults]
inventory = /home/mohamed/lab2/inventory
remote_user = devops
[privilege_escalation]
become = True
become_method = sudo
become_user = root
become_ask_pass = False
"ansible.cfg" 10L, 176B
                                                                         All
                                                            1,1
[mohamed@master lab2]$ ansible all -m ping
192.168.250.141 | SUCCESS => {
192.168.250.128 | SUCCESS => {
```

Set Global Variables and use Ansible Vault

```
ⅎ
                               mohamed@master:~/lab2/inventory/group_vars — vim all.yaml
                                                                                                       ≡
mysql_root_password: rootpassword
                                                                                     2,0-1
                                                                                                         All
                                              mohamed@master:~/Ansible_Final_Project/group_vars — vim all.yml
$ANSIBLE_VAULT;1.1;AES256
363631363938363564636239663830313166666333338626334646332313966326664303637313465
6638383266373435363131383064386261326538656232610a386565616435626237323836633032
38383663663561376464346163393833383737326262343638313435366662396463373433343662
3439386566626437640a383564363838333730646136623965356461663966643665353134383636
333536323062626634346165653637383232356366643364646465303236666663735616630623063
```

## **Create Roles**

**Apache Role (Web Server)** 



PHP application template

**Apache Configuration** 

```
ⅎ
                        mohamed@master:~/lab2/roles/apache/files — vim main.yml
                                                                            a ≡
name: Copy custom Apache configuration file
 copy:
   src: httpd.conf
   dest: /etc/httpd/conf/httpd.conf
   owner: root
   group: root
   mode: 0644
 name: Restart Apache to apply new configuration
 systemd:
   name: httpd
   state: restarted
                                                                                  All
-- INSERT --
                                                                  4,1
```

# MySQL Role (Database Server)

```
⊕
                                                mohamed@master:~/lab2/roles/mysql/tasks — vim main.yml
name: Install Python 3 and PyMySQL
   name:
     - python3
     - python3-PyMySQL
   state: present
 name: Install MySQL
 yum:
  name: mysql-server
  state: present
 name: Start and enable MySQL
 systemd:
  name: mysqld
  state: started
  enabled: yes
 name: Set MySQL root password
mysql_user:
  name: root
  password: "{{ mysql_root_password }}"
   login_unix_socket: /var/lib/mysql/mysql.sock
   state: present
```

# **Firewall Role (For Both Nodes)**

```
∄
                       mohamed@master:~/lab2/roles/firewall/tasks — vim main.yml
                                                                            Q ≡
name: Open HTTP port for Apache
firewalld:
   port: 80/tcp
   permanent: yes
   state: enabled
name: Open MySQL port
firewalld:
   port: 3306/tcp
   permanent: yes
   state: enabled
name: Reload firewall
firewalld:
   state: reloaded
                                                                  17,0-1
                                                                                 All
```

# **Backup Role**

• For MySQL Database

• For Apache Web Content

```
mohamed@master:-/lab2/roles/backup/tasks—vim backup_web.yml Q 
---
name: Backup web content
command: tar czf /root/web_content_backup.tar.gz /var/www/html
```

# • Backup Main Role

**Create the Playbook** 



## **Use Ansible Vault**

```
[mohamed@master lab2]$ ansible-vault encrypt group_vars/all.yml
New Vault password:
Confirm New Vault password:
Encryption successful
```

**Apply The Playbook** 

[moha Vault	amed@master lab2]\$ ansible-playbook -i inventory/hosts playbook.ymlask-vault-pass t password:
PLAY	[web] ************************************
	[Gathering Facts] ************************************
	[apache : Install Apache] ************************************
TASK ok: [	[apache : Install PHP] **********************************
	[apache : Copy index.php template] ************************************
TASK ok: [	[apache : Start and enable Apache] ************************************
	[firewall : Open HTTP port for Apache] ************************************
TASK ok: [	[firewall : Open MySQL port] ************************************
TASK chang	[firewall : Reload firewalld service] ************************************
	[backup : include_tasks] ***********************************
	[backup : include_tasks] ***********************************
	[backup : Backup web content] ************************************
TASK	[backup : Backup web content] ************************************
chang	
PLAY	[db] ************************************
TASK ok: [	[Gathering Facts] ************************************
	[mysql : Install Python 3 and PyMySQL] ************************************
TASK ok: [	[mysql : Install MySQL] ************************************
	[mysql : Start and enable MySQL] ************************************
TASK ok: [	
	[mysql : Set MySQL root password] ************************************
	[mysql : Set MySQL root password] ************************************
ok: [	[mysql : Set MySQL root password] ************************************
ok: [ TASK ok: [ TASK	[mysql : Set MySQL root password] ************************************
ok: [ TASK ok: [ TASK chang	[mysql : Set MySQL root password] ************************************
ok: [ TASK ok: [ TASK chang TASK inclu	[mysql : Set MysQL root password] ************************************
ok: [ TASK ok: [ TASK chang TASK inclu TASK ok: [ TASK	[mysql : Set MysQL root password] ************************************
ok: [ TASK ok: [ TASK chang TASK inclu TASK ok: [ TASK chang	[mysql : Set MysQL root password] ************************************