Ansible Labs

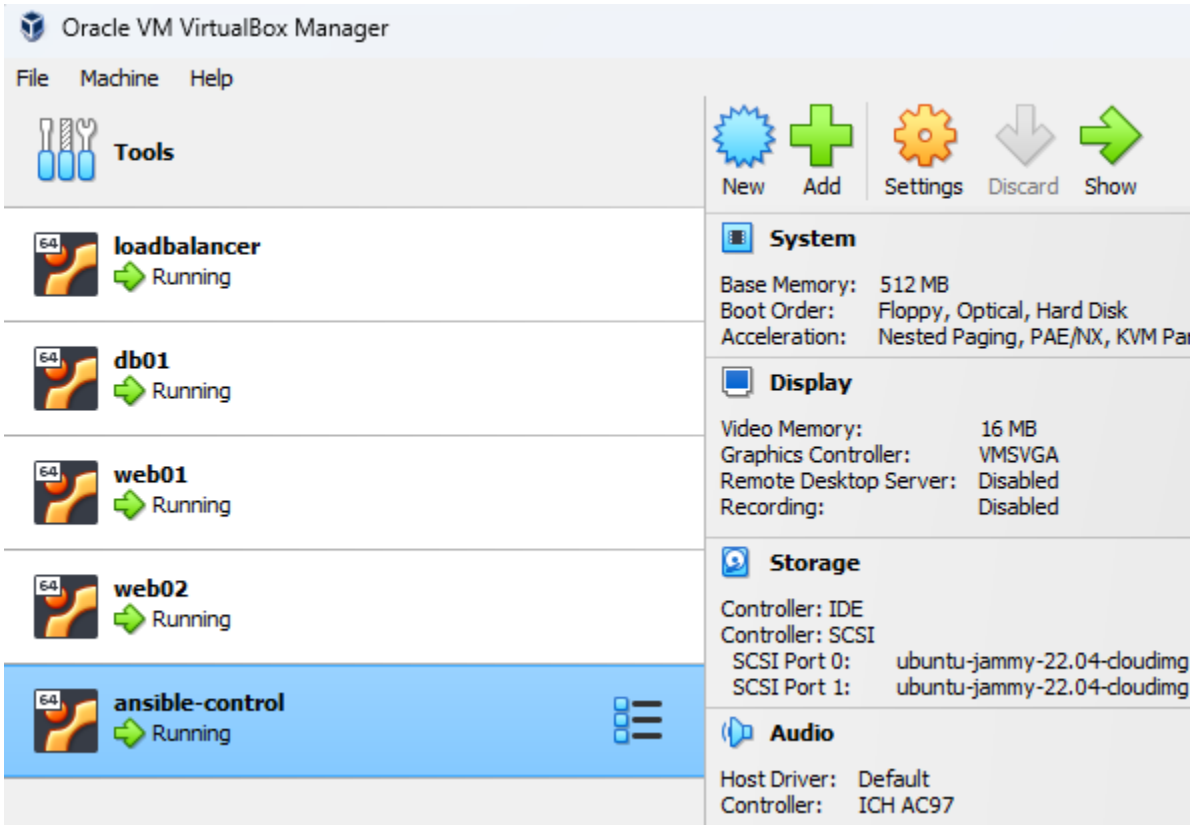
Ansible Lab 1 - Installation and Inventory file basics

1.1. Create the Vagrantfile

With the Vagrantfile we defined an array of servers with their respective hostname, IP address, and SSH port. It then loops through each machine in the array and defines a Vagrant machine with the specified hostname, box, and network settings.

```
Vagrant.configure("2") do |config|
 servers=[
  {
   :hostname => "loadbalancer",
   :box => "ubuntu/jammy64",
   :ip => "192.168.7.101",
   :ssh_port => '2341'
  },
  {
   :hostname => "db01",
   :box => "ubuntu/jammy64",
   :ip => "192.168.7.102",
   :ssh_port => '2342'
  },
  {
   :hostname => "web01",
   :box => "ubuntu/jammy64",
   :ip => "192.168.7.103",
   :ssh_port => '2343'
  },
  {
   :hostname => "web02",
   :box => "ubuntu/jammy64",
   :ip => "192.168.7.104",
   :ssh_port => '2344'
  },
  {
   :hostname => "ansible-control",
   :box => "ubuntu/jammy64",
   :ip => "192.168.7.105",
   :ssh_port => '2345'
  }

 ]

 servers.each do |machine|

  config.vm.define machine[:hostname] do |node|
   node.vm.box = machine[:box]
   node.vm.hostname = machine[:hostname]

   node.vm.network :public_network, bridge: "enp0s8: Wi-Fi (AirPort)"
   node.vm.network :private_network, ip: machine[:ip]
   node.vm.network "forwarded_port", guest: 22, host: machine[:ssh_port], id: "ssh"

   config.vm.provision "shell", inline: <<-SHELL
    sed -i 's/PasswordAuthentication no/PasswordAuthentication yes/g' /etc/ssh/sshd_config
    systemctl restart sshd.service
   SHELL

   node.vm.provider :virtualbox do |v|
    v.customize ["modifyvm", :id, "--memory", 512]
    v.customize ["modifyvm", :id, "--cpus", 1]
    v.customize ["modifyvm", :id, "--name", machine[:hostname]]
    v.customize ["modifyvm", :id, "--graphicscontroller", "vmsvga"]
    v.customize ["modifyvm", :id, "--natdnshostresolver1", "on"]
   end
  end
 end

end
```

1.2. Create VMs using vagrant and ssh to our control server




```
PS C:\Users\V&M\Desktop\DevOps\Homework and Labs\ansible-labs> vagrant ssh ansible-control
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-69-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sat Apr 22 13:03:39 UTC 2023

  System load:  0.97705078125     Users logged in:         0
  Usage of /:   3.7% of 38.70GB   IPv4 address for enp0s3: 10.0.2.15
  Memory usage: 44%               IPv4 address for enp0s8: 192.168.1.110
  Swap usage:   0%                IPv4 address for enp0s9: 192.168.7.105
  Processes:    103


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


vagrant@ansible-control:~$
```

1.2. Create the /vagrant/hosts_file

```
vagrant@ansible-control:/vagrant$ cat hosts_file
192.168.7.101 loadbalancer
192.168.7.102 db01
192.168.7.103 web01
192.168.7.104 web02
192.168.7.105 ansible-control
```

1.4. Copy /vagrant/hosts_file to /etc/hosts

```
vagrant@ansible-control:/etc$ cat hosts
192.168.7.101 loadbalancer
192.168.7.102 db01
192.168.7.103 web01
192.168.7.104 web02
192.168.7.105 ansible-controlvagrant@ansible-control:/etc$
```

1.5. Install Ansible

```
Setting up ansible (2.10.7+merged+base+2.10.8+dfsg-1) ...
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
vagrant@ansible-control:~$
```

1.6. Create an inventory file named hosts

```
vagrant@ansible-control:~$ ls
ansible-lab1  ansible-lab2  ansible-lab3  ansible-lab4  ansible-lab5  hosts
vagrant@ansible-control:~$
```

1.7. Test out a command.

```
vagrant@ansible-control:~$ ansible localhost -m command -a hostname
[WARNING]: No inventory was parsed, only implicit localhost is available
localhost | CHANGED | rc=0 >>
ansible-control
vagrant@ansible-control:~$ ansible localhost -m command -a date
[WARNING]: No inventory was parsed, only implicit localhost is available
localhost | CHANGED | rc=0 >>
Sat Apr 22 17:20:02 UTC 2023
vagrant@ansible-control:~$
```

1.8. Generate SSH Keys and copy to hosts

```
Your identification has been saved in /home/vagrant/.ssh/id_rsa
Your public key has been saved in /home/vagrant/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:9R8Su5RaP1CGKBm4fOFCUTBLcB/JjArZc7wSEa8Fevo vagrant@ansible-control
The key's randomart image is:
+---[RSA 3072]----+
|   o==*O+.       |
|  o.+=*+=+ . .   |
|  ...B+++.o o o  |
|   ooo= oo . *   |
|  . .. oS   O .  |
|   .       + * . |
|    E     . . +  |
|               . |
|                 |
+----[SHA256]-----+
```

```
Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'localhost'"
and check to make sure that only the key(s) you wanted were added.
```

```
Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'db01'"
and check to make sure that only the key(s) you wanted were added.
```

1.9. Test running ad-hoc commands to all hosts

```
vagrant@ansible-control:~$ ansible webstack -i /home/vagrant/hosts -m command -a hostname
web02 | CHANGED | rc=0 >>
web02
web01 | CHANGED | rc=0 >>
web01
loadbalancer | CHANGED | rc=0 >>
loadbalancer
db01 | CHANGED | rc=0 >>
db01
vagrant@ansible-control:~$
```

2. Ansible Lab 2 - Ad HOC tasks and Modules

```
vagrant@ansible-control:~$ rsync -WaP /vagrant/ansible-lab1/ /vagrant/ansible-lab2/
sending incremental file list
created directory /vagrant/ansible-lab2
./
hosts
            134 100%    0.00kB/s    0:00:00 (xfr#1, to-chk=0/2)
vagrant@ansible-control:~$ cd /vagrant/ansible-lab2/
vagrant@ansible-control:/vagrant/ansible-lab2$
```

2.1. Ansible ad hoc commands for checking the uptime of the hosts

```
vagrant@ansible-control:/vagrant/ansible-lab2$ ansible all -i /home/vagrant/hosts -m command -a uptime
web02 | CHANGED | rc=0 >>
 17:50:21 up 47 min,  1 user,  load average: 0.15, 0.03, 0.01
loadbalancer | CHANGED | rc=0 >>
 17:50:21 up 50 min,  1 user,  load average: 0.00, 0.00, 0.00
web01 | CHANGED | rc=0 >>
 17:50:21 up 48 min,  1 user,  load average: 0.13, 0.03, 0.01
db01 | CHANGED | rc=0 >>
 17:50:21 up 49 min,  1 user,  load average: 0.32, 0.09, 0.03
ansible-control | CHANGED | rc=0 >>
 17:50:21 up 45 min,  2 users,  load average: 1.01, 0.49, 0.26
```

```
vagrant@ansible-control:/vagrant/ansible-lab2$ ansible all -i /home/vagrant/hosts -m shell -a uptime
loadbalancer | CHANGED | rc=0 >>
 17:50:55 up 50 min,  1 user,  load average: 0.00, 0.00, 0.00
db01 | CHANGED | rc=0 >>
 17:50:55 up 49 min,  1 user,  load average: 0.18, 0.08, 0.02
web01 | CHANGED | rc=0 >>
 17:50:55 up 48 min,  1 user,  load average: 0.07, 0.03, 0.01
web02 | CHANGED | rc=0 >>
 17:50:55 up 47 min,  1 user,  load average: 0.09, 0.03, 0.01
ansible-control | CHANGED | rc=0 >>
 17:50:56 up 46 min,  2 users,  load average: 1.34, 0.60, 0.31
```

```
vagrant@ansible-control:/vagrant/ansible-lab2$ ansible all -i /home/vagrant/hosts -a uptime
web01 | CHANGED | rc=0 >>
 17:51:07 up 48 min,  1 user,  load average: 0.06, 0.03, 0.00
web02 | CHANGED | rc=0 >>
 17:51:07 up 47 min,  1 user,  load average: 0.07, 0.03, 0.01
db01 | CHANGED | rc=0 >>
 17:51:07 up 49 min,  1 user,  load average: 0.15, 0.08, 0.02
loadbalancer | CHANGED | rc=0 >>
 17:51:07 up 50 min,  1 user,  load average: 0.00, 0.00, 0.00
ansible-control | CHANGED | rc=0 >>
 17:51:07 up 46 min,  2 users,  load average: 1.14, 0.58, 0.30
```

2.2. Check the free memory or memory usage of hosts using ansible ad hoc command.

```
vagrant@ansible-control:~/ansible-lab2$ ansible all -a "free -m" -i /home/vagrant/hosts
loadbalancer | CHANGED | rc=0 >>
               total        used        free      shared  buff/cache   available
Mem:             466         168          23           0         274         286
Swap:              0           0           0
web02 | CHANGED | rc=0 >>
               total        used        free      shared  buff/cache   available
Mem:             466         172          26           0         267         286
Swap:              0           0           0
web01 | CHANGED | rc=0 >>
               total        used        free      shared  buff/cache   available
Mem:             466         170          18           0         277         286
Swap:              0           0           0
db01 | CHANGED | rc=0 >>
               total        used        free      shared  buff/cache   available
Mem:             466         167          19           0         278         286
Swap:              0           0           0
ansible-control | CHANGED | rc=0 >>
               total        used        free      shared  buff/cache   available
Mem:             466         229          78           1         158         214
Swap:              0           0           0
```

2.3. Update and upgrade all machines

```
ansible all -i /home/vagrant/hosts -m command -a 'sudo apt update'
ansible all -i /home/vagrant/hosts -m command -a 'sudo apt upgrade -y'
```

```
Check GRUB_DISABLE_OS_PROBER documentation entry.
done
WARNING: apt does not have a stable CLI interface. Use with caution in scripts.
vagrant@ansible-control:~/ansible-lab2$
```

2.4. Use APT module to install services

```
vagrant@ansible-control:/vagrant/ansible-lab2$ ansible all -i /home/vagrant/hosts --become -m apt -a "update_cache=yes"
web01 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
```

```
vagrant@ansible-control:/vagrant/ansible-lab2$ ansible all -i /home/vagrant/hosts --become -m apt -a "name=swapspace state=present"
web01 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
```

```
vagrant@ansible-control:/vagrant/ansible-lab2$ ansible all -i /home/vagrant/hosts --become -m apt -a "name=net-tools state=present"
web02 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
```

```
vagrant@ansible-control:/vagrant/ansible-lab2$ ansible webservers -i /home/vagrant/hosts --become -m apt -a "name=apache2 state=present"
web01 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
```

```
vagrant@ansible-control:/vagrant/ansible-lab2$ ansible database -i /home/vagrant/hosts --become -m apt -a "name=mysql-server state=present"
db01 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
```

2.5. Use service module to manage services

```
vagrant@ansible-control:~/ansible-lab2$ ansible database -i /home/vagrant/hosts -m service -a "name=mysql state=started"
db01 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "name": "mysql",
    "state": "started",
    "status": {
        "ActiveEnterTimestamp": "Sat 2023-04-22 18:27:22 UTC",
```

```
vagrant@ansible-control:~/ansible-lab2$ ansible database --become -i /home/vagrant/hosts -m service -a "name=mysql
state=restarted"
db01 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": true,
    "name": "mysql",
    "state": "started",
    "status": {
        "ActiveEnterTimestamp": "Sat 2023-04-22 18:27:22 UTC",
```

2.6. Use ansible to reboot webstack

```
vagrant@ansible-control:~/ansible-lab2$ ansible webstack -i /home/vagrant/hosts --become -a "reboot --reboot"
loadbalancer | UNREACHABLE! => {
    "changed": false,
    "msg": "Failed to connect to the host via ssh: Shared connection to loadbalancer closed.",
    "unreachable": true
}
web01 | UNREACHABLE! => {
    "changed": false,
    "msg": "Failed to connect to the host via ssh: Shared connection to web01 closed.",
    "unreachable": true
}
web02 | UNREACHABLE! => {
    "changed": false,
    "msg": "Failed to connect to the host via ssh: Shared connection to web02 closed.",
    "unreachable": true
}
db01 | UNREACHABLE! => {
    "changed": false,
    "msg": "Failed to connect to the host via ssh: Shared connection to db01 closed.",
    "unreachable": true
}
```

3. Ansible Lab 3 - Playbooks, Templates and Handlers

```
vagrant@ansible-control:~$ rsync -WaP /vagrant/ansible-lab2/ /vagrant/ansible-lab3/
sending incremental file list
created directory /vagrant/ansible-lab3
./
hosts
            134 100%    0.00kB/s    0:00:00 (xfr#1, to-chk=0/2)
vagrant@ansible-control:~$ cd /vagrant/ansible-lab3/
vagrant@ansible-control:/vagrant/ansible-lab3$
```

3.1. Exam the playbook and look over the details of the YAML file

```yaml
- hosts: webservers
  become: yes
  vars:
    http_port: 8000
    https_port: 4443
    html_welcome_msg: "Hello Scalefocus Academy!"
  tasks:
  - name: ensure apache is at the latest version
    apt:
      name: apache2
      state: latest
  - name: write the apache2 ports.conf config file
    template:
      src: templates/ports.conf.j2
      dest: /etc/apache2/ports.conf
    notify:
    - restart apache
  - name: write a basic index.html file
    template:
      src: templates/index.html.j2
      dest: /var/www/html/index.html
    notify:
    - restart apache
  - name: ensure apache is running
    service:
      name: apache2
      state: started
  handlers:
    - name: restart apache
      service:
        name: apache2
        state: restarted
      listen: "restart apache"
```

3.2. Check the Templates

```
vagrant@ansible-control:/vagrant/ansible-lab3/templates$ ls
index.html.j2  ports.conf.j2
vagrant@ansible-control:/vagrant/ansible-lab3/templates$
```

3.3. Run the playbook

```
vagrant@ansible-control:/vagrant/ansible-lab3$ ansible-playbook -i /home/vagrant/hosts playbook1.yml

PLAY [webservers] ********************************************************

TASK [Gathering Facts] ***************************************************
ok: [web01]
ok: [web02]

TASK [ensure apache is at the latest version] ****************************
ok: [web01]
ok: [web02]

TASK [write the apache2 ports.conf config file] **************************
changed: [web01]
changed: [web02]

TASK [write a basic index.html file] *************************************
changed: [web01]
changed: [web02]

TASK [ensure apache is running] ******************************************
ok: [web01]
ok: [web02]

RUNNING HANDLER [restart apache] *****************************************
changed: [web01]
changed: [web02]

PLAY RECAP ***************************************************************
web01                      : ok=6    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
web02                      : ok=6    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

3.4. Test connectivity to servers

```
vagrant@ansible-control:/vagrant/ansible-lab3$ curl web01:8000
<html>
<h1>Hello Scalefocus Academy! You have reached the web01
server.</h1>
</html>
```

```
vagrant@ansible-control:/vagrant/ansible-lab3$ curl web02:8000
<html>
<h1>Hello Scalefocus Academy! You have reached the web02
server.</h1>
</html>
```

4. Ansible Lab 4 - Re-usable playbooks, import_tasks, Roles and Ansible Galaxy

```
vagrant@ansible-control:/vagrant/ansible-lab3$ rsync -WaP /vagrant/ansible-lab3/ /vagrant/ansible-lab4/
sending incremental file list
created directory /vagrant/ansible-lab4
./
hosts
            134 100%    0.00kB/s    0:00:00 (xfr#1, to-chk=4/6)
playbook1.yml
          2,836 100%    2.70MB/s    0:00:00 (xfr#2, to-chk=3/6)
templates/
templates/index.html.j2
            106 100%    6.09kB/s    0:00:00 (xfr#3, to-chk=1/6)
templates/ports.conf.j2
            353 100%   18.14kB/s    0:00:00 (xfr#4, to-chk=0/6)
vagrant@ansible-control:/vagrant/ansible-lab3$ cd /vagrant/ansible-lab4/
vagrant@ansible-control:/vagrant/ansible-lab4$
```

4.1. Re-usable playbooks, import_tasks

4.1.1 Create new directory tasks and new yml file apache2_install. Then edit the playbook1.yml and move the tasks code to the newly created file.

```
vagrant@ansible-control:/vagrant/ansible-lab4/tasks$ cat apache2_install.yml
  - name: ensure apache is at the latest version
    apt:
      name: apache2
      state: latest
  - name: write the apache2 ports.conf config file
    template:
      src: templates/ports.conf.j2
      dest: /etc/apache2/ports.conf
    notify:
    - restart apache
  - name: write a basic index.html file
    template:
      src: templates/index.html.j2
      dest: /var/www/html/index.html
    notify:
    - restart apache
  - name: ensure apache is running
    service:
      name: apache2
      state: started
```

4.1.2 Create new directory handlers and new yml file main.yml. Than edit the playbook1.yml and move the handlers code to the newly created file.

```
vagrant@ansible-control:/vagrant/ansible-lab4/handlers$ cat main.yml
- name: restart apache
  service:
    name: apache2
    state: restarted
```

After we move around our code, we will use the import_tasks with the path of our new files

```
vagrant@ansible-control:/vagrant/ansible-lab4$ cat playbook1.yml
- hosts: webservers
  become: yes
  vars:
    http_port: 8000
    https_port: 4443
    html_welcome_msg: "Hello Scalefocus Academy!"
  tasks:
  - import_tasks: tasks/apache2_install.yml
  handlers:
  - import_tasks: handlers/main.yml
```

4.1.2 Run the playbook

```
vagrant@ansible-control:/vagrant/ansible-lab4$ ansible-playbook -i hosts -K playbook1.yml
BECOME password:

PLAY [webservers] ***************************************************************************************

TASK [Gathering Facts] **********************************************************************************
ok: [web01]
ok: [web02]

TASK [ensure apache is at the latest version] ***********************************************************
ok: [web02]
ok: [web01]

TASK [write the apache2 ports.conf config file] *********************************************************
ok: [web01]
ok: [web02]

TASK [write a basic index.html file] ********************************************************************
ok: [web02]
ok: [web01]

TASK [ensure apache is running] *************************************************************************
ok: [web02]
ok: [web01]

PLAY RECAP **********************************************************************************************
web01                      : ok=5    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    i
gnored=0
web02                      : ok=5    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    i
gnored=0
```

4.2. Ansible Roles and Ansible Galaxy

4.2.1 Use ansible-galaxy to create Apache2 webserver role scaffolding
ansible-galaxy init roles/apache2

4.2.2 Move the tasks to the roles\webserver folder and edit the main.yml in apache2/tasks.
mv tasks/apache2_install.yml roles/apache2/tasks/
mv handlers/main.yml roles/apache2/handlers/main.yml
mv templates/ roles/apache2/
rmdir tasks/ handlers/

```
vagrant@ansible-control:/vagrant/ansible-lab4$ tree
.
├── hosts
├── playbook1.yml
└── roles
    └── apache2
        ├── README.md
        ├── defaults
        │   └── main.yml
        ├── handlers
        │   └── main.yml
        ├── meta
        │   └── main.yml
        ├── tasks
        │   ├── apache2_install.yml
        │   └── main.yml
        ├── templates
        │   ├── index.html.j2
        │   └── ports.conf.j2
        ├── tests
        │   ├── inventory
        │   └── test.yml
        └── vars
            └── main.yml

9 directories, 13 files
```

```
vagrant@ansible-control:/vagrant/ansible-lab4$ cat roles/apache2/tasks/main.yml
---
# tasks file for roles/apache2
- include: apache2_install.yml
```

4.2.3 Run the playbook

```
vagrant@ansible-control:/vagrant/ansible-lab4$ ansible-playbook -i hosts -K playbook1.yml
BECOME password:

PLAY [webservers] ***************************************************************************************************

TASK [Gathering Facts] **********************************************************************************************
ok: [web01]
ok: [web02]

TASK [apache2 : ensure apache is at the latest version] *************************************************************
ok: [web01]
ok: [web02]

TASK [apache2 : write the apache2 ports.conf config file] ***********************************************************
ok: [web02]
ok: [web01]

TASK [apache2 : write a basic index.html file] **********************************************************************
ok: [web01]
ok: [web02]

TASK [apache2 : ensure apache is running] ***************************************************************************
ok: [web02]
ok: [web01]

PLAY RECAP **********************************************************************************************************
web01                      : ok=5    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
web02                      : ok=5    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

4.2.4 Use ansible-galaxy to create 'common' and nginx role scaffolding

```
vagrant@ansible-control:/vagrant/ansible-lab4$ ansible-galaxy init roles/common
- Role roles/common was created successfully
vagrant@ansible-control:/vagrant/ansible-lab4$ ansible-galaxy init roles/nginx
- Role roles/nginx was created successfully
```

4.2.5 Setup 'common' role tasks/main.yml and tasks/install_tools.yml

```
vagrant@ansible-control:/vagrant/ansible-lab4$ cat roles/common/tasks/install_tools.yml
- name: "Install Common packages"
  apt: name={{ item }} state=latest
  with_items:
    - net-tools
    - tree
    - python3-pip

- name: Install pymysql python package
  pip:
    name: pymysql
```

```
vagrant@ansible-control:/vagrant/ansible-lab4$ cat roles/common/tasks/main.yml
---
# tasks file for roles/common
- include: install_tools.yml
```

4.2.6 Setup the nginx role

```
vagrant@ansible-control:/vagrant/ansible-lab4$ cat roles/nginx/tasks/install_packages.yml
- name: "Install Nginx packages"
  apt:
    name: nginx
    state: present
```

```
vagrant@ansible-control:/vagrant/ansible-lab4$ cat roles/nginx/tasks/configure_nginx.yml
- name: Deploy Nginx sites configuration
  template:
    src: mysite.j2
    dest: /etc/nginx/sites-enabled/mysite
  notify: restart nginx
- name: Remove defaults
  file:
    path: /etc/nginx/sites-enabled/default
    state: absent
```

```
vagrant@ansible-control:/vagrant/ansible-lab4$ cat roles/nginx/tasks/main.yml
---
# tasks file for roles/nginx
- include: install_packages.yml
- include: configure_nginx.yml
```

```
vagrant@ansible-control:/vagrant/ansible-lab4$ cat roles/nginx/templates/mysite.j2
upstream webservers {
    server 192.168.7.103:8000;
    server 192.168.7.104:8000;
}

server {
    listen 80;
    location / {
        proxy_pass http://webservers;
    }
}
```

```
vagrant@ansible-control:/vagrant/ansible-lab4$ cat roles/nginx/handlers/main.yml
---
# handlers file for roles/nginx
- name: restart nginx
  service: name=nginx state=restarted
```

```
vagrant@ansible-control:/vagrant/ansible-lab4$ cat playbook1.yml
# This playbook consists of two plays.
# The first play targets the hosts with the tag "webservers".
- hosts: webservers
  become: yes
  vars:
    http_port: 8000
    https_port: 4443
    html_welcome_msg: "Hello Scalefocus Academy!"
  roles:
    - common
    - apache2

# The second play targets the hosts with the tag "proxy".
- hosts: proxy
  become: yes
  roles:
    - common
    - nginx
```

4.2.7 Run the playbook and test.

```
PLAY RECAP *********************************************************************************************************
loadbalancer               : ok=7    changed=6    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
web01                      : ok=7    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
web02                      : ok=7    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

```
vagrant@ansible-control:/vagrant/ansible-lab4$ for i in {1..10}; do curl loadbalancer; done
<html>
<h1>Hello Scalefocus Academy! You have reached the web01
server.</h1>
</html>
<html>
<h1>Hello Scalefocus Academy! You have reached the web02
server.</h1>
</html>
<html>
<h1>Hello Scalefocus Academy! You have reached the web01
server.</h1>
</html>
<html>
<h1>Hello Scalefocus Academy! You have reached the web02
server.</h1>
</html>
<html>
<h1>Hello Scalefocus Academy! You have reached the web01
server.</h1>
```

Ansible Lab 5 - Variables, Ansible Vault

5.1 Create mysql role using ansible-galaxy.

```
vagrant@ansible-control:/vagrant/ansible-lab5$ ansible-galaxy init roles/mysql
- Role roles/mysql was created successfully
```

5.2 Create tasks, handlers and templates for new mysql role.

```
vagrant@ansible-control:/vagrant/ansible-lab5$ cat roles/mysql/tasks/install_mysql.yml
- name: Install MySQL server
  apt:
    name: mysql-server
    update_cache: yes

- name: Installing python module MySQL-python
  pip:
    name: PyMySQL

- name: Ensure mysql-server is running
  service:
    name: mysql
    state: started
```

```
vagrant@ansible-control:/vagrant/ansible-lab5$ cat roles/mysql/tasks/setup_mysql.yml
- name: Create my.cnf configuration file
  template:
    src: templates/my.cnf.j2
    dest: /etc/mysql/conf.d/mysql.cnf
  notify: restart mysql

- name: Configure MySQL server to listen on all interfaces
  lineinfile:
```

```
vagrant@ansible-control:/vagrant/ansible-lab5$ cat roles/mysql/tasks/main.yml
---
# tasks file for roles/mysql
- include: install_mysql.yml
- include: setup_mysql.yml
```

```
vagrant@ansible-control:/vagrant/ansible-lab5$ cat roles/mysql/templates/my.cnf.j2
[mysql]
bind-address = 0.0.0.0
```

```
vagrant@ansible-control:/vagrant/ansible-lab5$ cat roles/mysql/handlers/main.yml
---
# handlers file for roles/mysql
- name: restart mysql
  service:
    name: mysql
    state: restarted
  listen: "restart mysql"
```

5.3 Set our vars and encrypt the important data

```
vagrant@ansible-control:/vagrant/ansible-lab5$ cat vars/main.yml
---
http_port: 8000
https_port: 4443
html_welcome_msg: "Hello Scalefocus Academy!"
mysql_user: simple_user
mysql_password: "{{ vaultMySqlPassword }}"
mysql_root_password: "{{ vaultMySqlRootPassword }}"
```

```
vagrant@ansible-control:/vagrant/ansible-lab5$ cat vars/vault.yml
$ANSIBLE_VAULT;1.1;AES256
353036346164386633336163353231323461336463383665636463373866656564626535636463653
333732353133383066383666373964626130306336343963 0a6431333436336531343236666330393
376466333137393962376636316362646165373739363365666635623835613833303038646537333
616234633466346239 0a34386636623336643933653562316166356431333962363332383966353331
313631376461333962363631633131363162373663373766393366316232306135653731633435303
303435363332336138656664616164636362653137653039613034306331613139316330386666663
333934393333316333343538323137653762373133333963303361363533343665346335666236336
30613835613061333937
```

5.4 Modify the playbook1.yml and add our new play

```
vagrant@ansible-control:/vagrant/ansible-lab5$ cat playbook1.yml
- hosts: webservers
  become: yes
  vars_files:
    - vars/main.yml
  roles:
    - common
    - apache2

- hosts: proxy
  become: yes
  roles:
    - common
    - nginx

- hosts: database
  become: yes
  vars_files:
    - vars/main.yml
    - vars/vault.yml
  vars_prompt:
    - name: mysql_database
      prompt: Please enter the database name.
      private: no
  roles:
    - common
    - mysql
```

5.5 Run playbook

```
PLAY RECAP *********************************************************************************************************
db01                       : ok=13   changed=9    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
loadbalancer               : ok=6    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
web01                      : ok=7    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
web02                      : ok=7    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

5.6 Test a mysql connection to database

```
vagrant@ansible-control:/vagrant/ansible-lab5$ mysql -h 192.168.7.102 -u simple_user -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.32-0ubuntu0.22.04.2 (Ubuntu)
```

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mydb               |
| performance_schema |
+--------------------+
3 rows in set (0.02 sec)
```

Project idea: Deploying a web application with Ansible

In progress…