ĐẠI HỌC QUỐC GIA TP.HCM

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

MÔN HỌC:QUẢN TRỊ MẠNG VÀ HỆ THỐNG

BÁO CÁO ĐỒ ÁN CUỐI KÌ

Chủ đề : System Automation
Giảng viên hướng dẫn: Trần Thị Dung

NT132.N22.ATCL

Sinh viên thực hiện :

20521543 - Trần Đặng Hồng Loan

20521977 - Phạm Văn Thời

20521560 - Phạm Ngọc Lợi

Mục lục

I Introduction

1. Overview :

1.1 Overview about System automation

- System automation is the process of automating the provisioning, patching, scheduling, security, compliance.
- Today, network operations teams typically do everything manually, and they need automation tools to support.

1.2 Ansible

- Ansible is an open source automation and orchestration tool for software provisioning, configuration management, and software deployment.

Ansible can easily run and configure Unix-like systems as well as Windows systems to provide infrastructure as code. It contains its own declarative programming language for system configuration and management.

- Ansible is popular for its simplicity of installation, ease of use in what concerns the connectivity to clients, its lack of agent for Ansible clients and the multitude of skills. It functions by connecting via SSH to the clients, so it doesn't need a special agent on the client-side, and by pushing modules to the clients, the modules are then executed locally on the client-side and the output is pushed back to the Ansible server.
- Since it uses SSH, it can very easily connect to clients using SSH-Keys, simplifying though the whole process. Client details, like hostnames or IP addresses and SSH ports, are stored in files called inventory files. Once you have created an inventory file and populated it, ansible can use it.

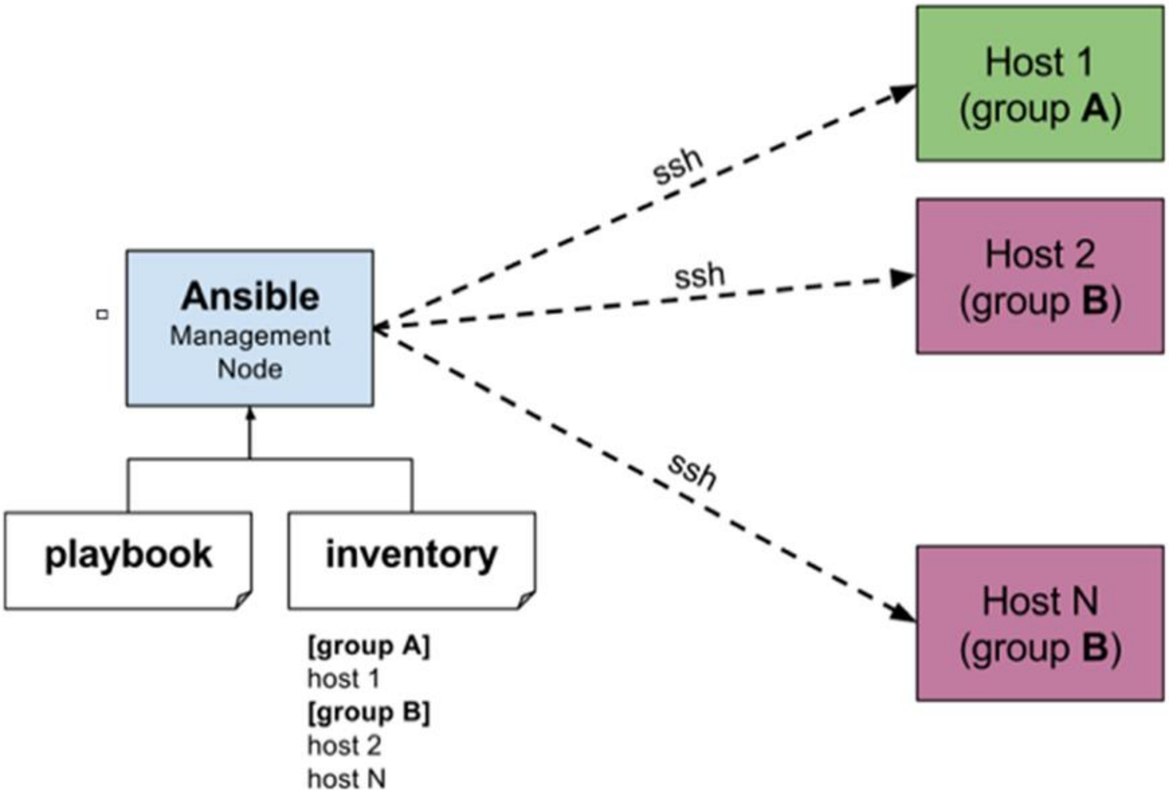
2. **Components**

- **Ansible server** :
  - o **Ansible** :
    - **Inventory** : File containing data about the ansible client servers. Defined in later examples as hosts file
    - **Playbook file** (using YAML language) : Ansible Playbooks are lists of tasks that automatically execute against hosts. Groups of hosts form your Ansible inventory
    - **Task** : The tasks that ansible intends to implement for the remote host (in Playbook)
    - **Module** : Each module within an Ansible Playbook performs a specific task and contains metadata that determines when and where a task is executed, as well as which user executes it. (in playbook)
    - **IP address**
    - **Roles** : A way of organizing tasks and related files to be later called in a playbook (inplaybok)
  - o **Website components** :
    - Source code
    - Database
    - Others
  - o **Browser** : Firefox Ubuntu
- **SSH protocol**:
  - o **SSH keys** : are the safest way to ensure that your connection remains secure, and also, you won't be facing issues while connecting the Ansible hosts
- **Remote Host** :
  - o **Browser** : Firefox Ubuntu
  - o **IP address**

3. Operation :

3.1 Ansible overview works : The operation part is divided into two steps

- **Step 1 : Authenticate and connect**
  - The remote hosts are listed in the inventory
  - Ansible server will first connect to remote hosts using ssh protocol
  - Ansible uses an SSH channel for communication to control the remote machines/servers. Hence, any system that has an open an SSH port can be configured by an Ansible machine.
- **Step 2 : Implementation**
  - Once authenticated and connected to the remote hosts, ansible pushes the modules to the client, the modules are then executed locally on the client side and the output pushed back to the Ansible server
  - Tasks defined in the playbook will be deployed to remote hosts and even to ansible if desired automatically.
  - The remote hosts deployed by the ansible server execute the tasks defined sequentially in the playbook file



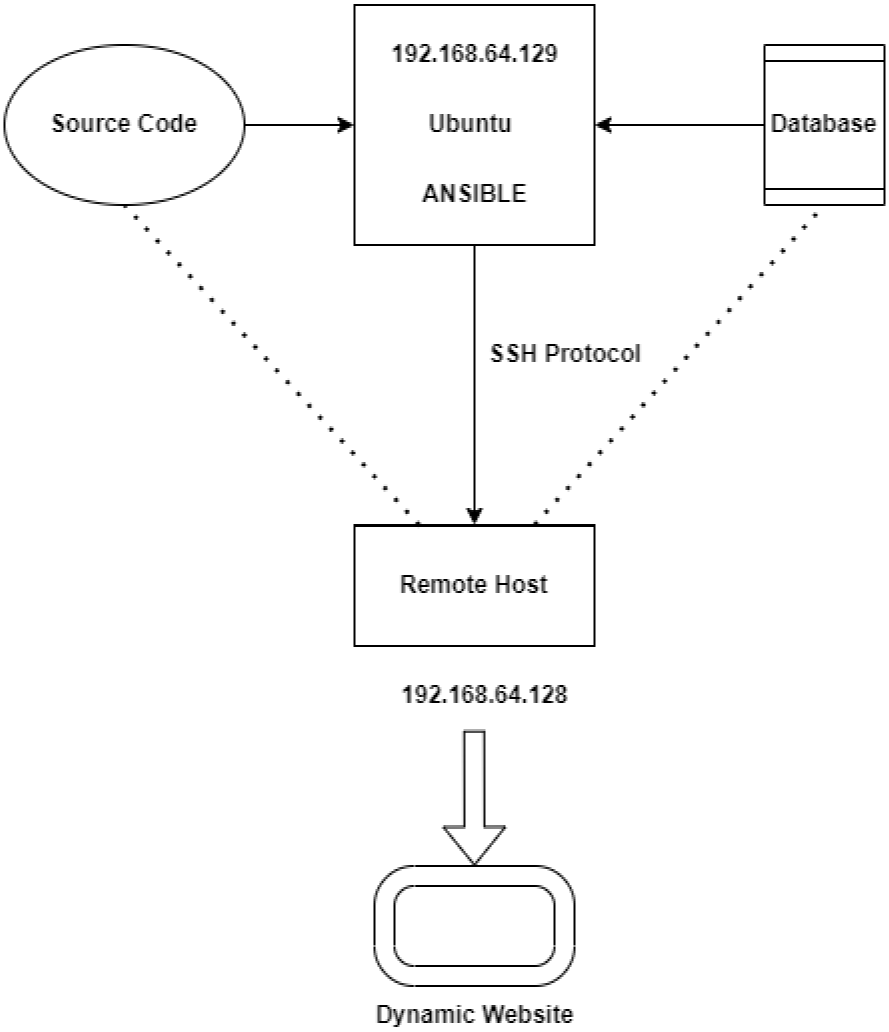
3.2 Project work : (MAIN IDEA)

- **Step 1 Authenticate and connect : same as above**

- **Step 2 Implementation**
  - First, if you want to deploy the website to remote hosts, we will install the webserver (Nginx) on the remote hosts' browser.
  - Next, we will deploy the source code, database and other components of the website that we intend to deploy into the installed webserver (nginx) so that we can complete web services.
  - The above tasks will be defined in the playbook file and deployed to the remote hosts listed in the inventory file
  - Detailed presentation will be implemented in implementation.

II. Implementation

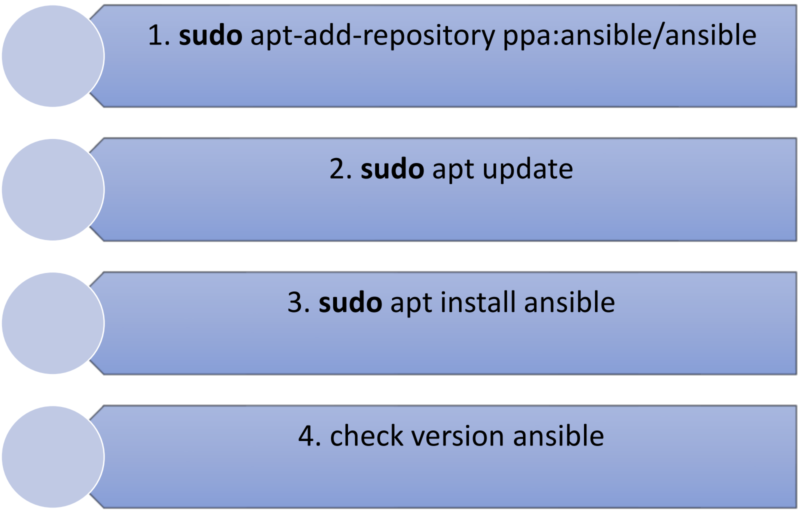
1. Topology:



Name	IP	TASK
Ansible server	192.168.64.129	Automatic desploy source code, database .
Remote host	192.168.64.128	Provice WebService

2. Installation

- **Deployment resources** : Two virtual machines running ubuntu
  - Ubuntu 1 IP : 192.168.64.129 : The machine where Ansible is installed and from which all tasks and playbooks will be ran
  - Ubuntu 2 IP : 192.168.64.128 : Remote host is deployed to dynamic website.
- **Ansible installation steps** :



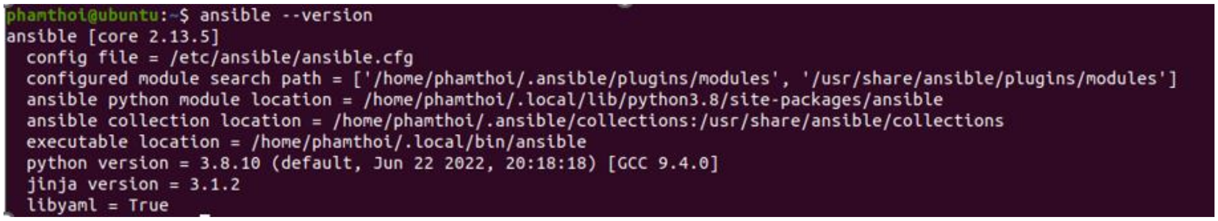
1. **sudo** apt-add-repository ppa:ansible/ansible

2. **sudo** apt update

3. **sudo** apt install ansible

4. check version ansible

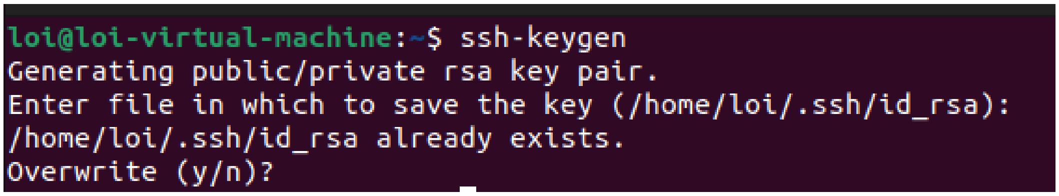
- Version **:**



```
phamthoi@ubuntu:~$ ansible --version
ansible [core 2.13.5]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/phamthoi/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /home/phamthoi/.local/lib/python3.8/site-packages/ansible
  ansible collection location = /home/phamthoi/.ansible/collections:/usr/share/ansible/collections
  executable location = /home/phamthoi/.local/bin/ansible
  python version = 3.8.10 (default, Jun 22 2022, 20:18:18) [GCC 9.4.0]
  jinja version = 3.1.2
  libyaml = True
```

3. Configuration:

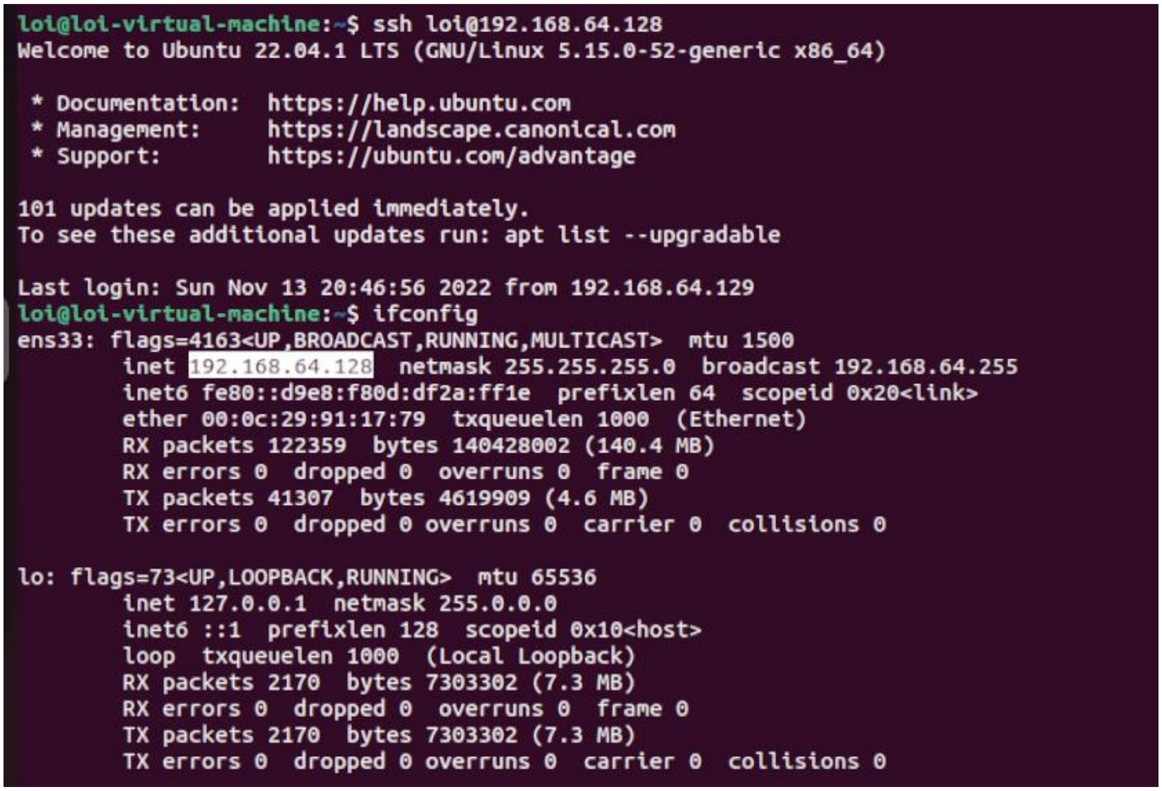
- **Establish connections to remote hosts**.
  - Ansible uses an SSH channel for communication to control the remote machines/servers



```
loi@loi-virtual-machine:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/loi/.ssh/id_rsa):
/home/loi/.ssh/id_rsa already exists.
Overwrite (y/n)?
```

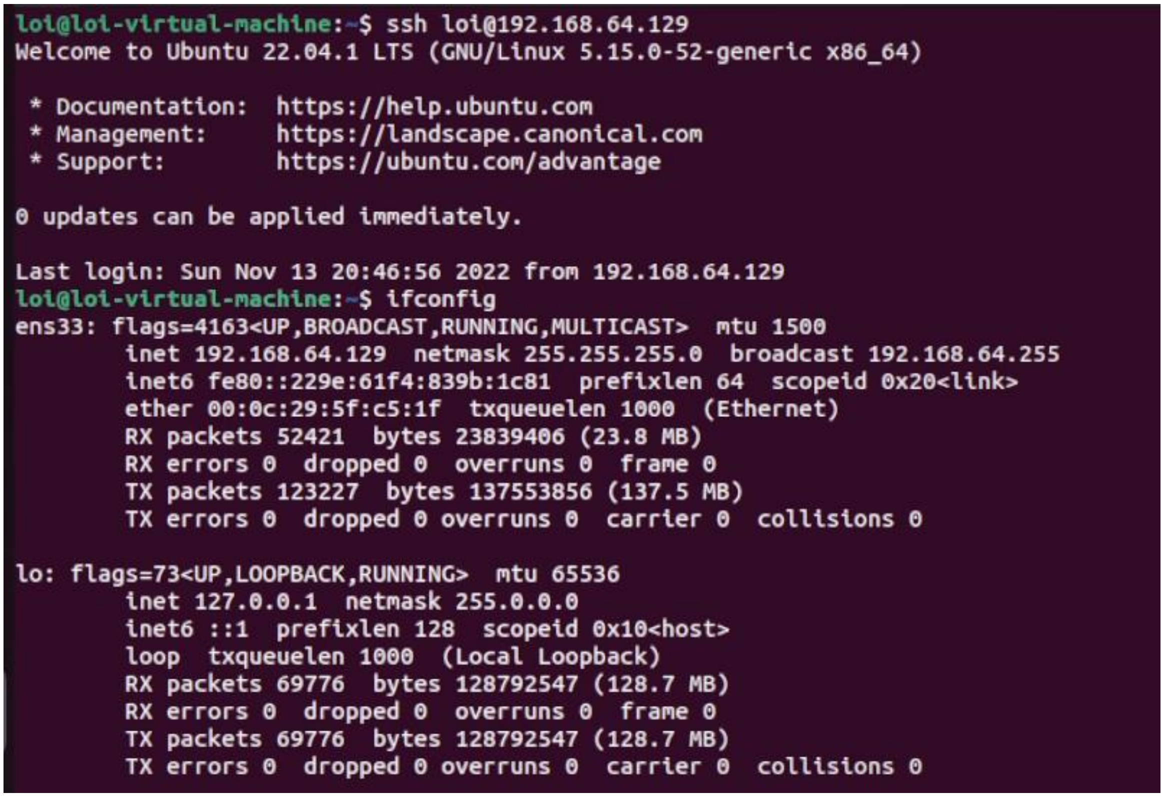
(ssh authentication with key and distribution)

  - Check connection to remote hosts



```
loi@loi-virtual-machine:~$ ssh loi@192.168.64.128
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-52-generic x86_64)

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

101 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Last login: Sun Nov 13 20:46:56 2022 from 192.168.64.129
loi@loi-virtual-machine:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.64.128  netmask 255.255.255.0  broadcast 192.168.64.255
        inet6 fe80::d9e8:f80d:df2a:ff1e  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:91:17:79  txqueuelen 1000  (Ethernet)
        RX packets 122359  bytes 140428002 (140.4 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 41307  bytes 4619909 (4.6 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 2170  bytes 7303302 (7.3 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2170  bytes 7303302 (7.3 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

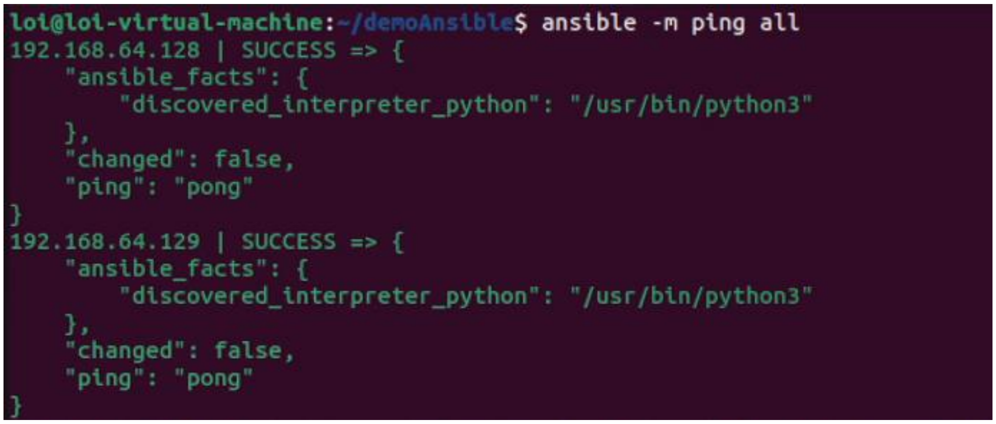
- Check connection to Ansible server



```
loi@loi-virtual-machine:~$ ssh loi@192.168.64.129
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-52-generic x86_64)

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

0 updates can be applied immediately.

Last login: Sun Nov 13 20:46:56 2022 from 192.168.64.129
loi@loi-virtual-machine:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.64.129  netmask 255.255.255.0  broadcast 192.168.64.255
        inet6 fe80::229e:61f4:839b:1c81  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:5f:c5:1f  txqueuelen 1000  (Ethernet)
        RX packets 52421  bytes 23839406 (23.8 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 123227  bytes 137553856 (137.5 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 69776  bytes 128792547 (128.7 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 69776  bytes 128792547 (128.7 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

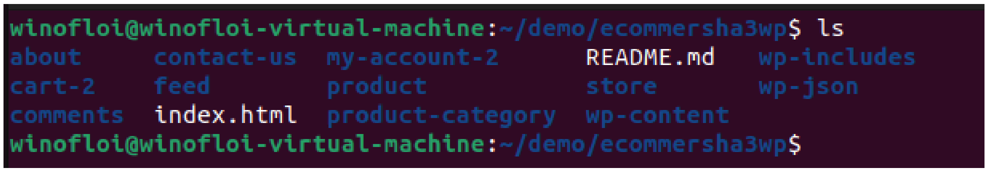
- The last step is to copy the ssh key to the Ansible server and to the remote hosts .

- Now it's time to test whether our Ansible control node can connect with remote hosts.



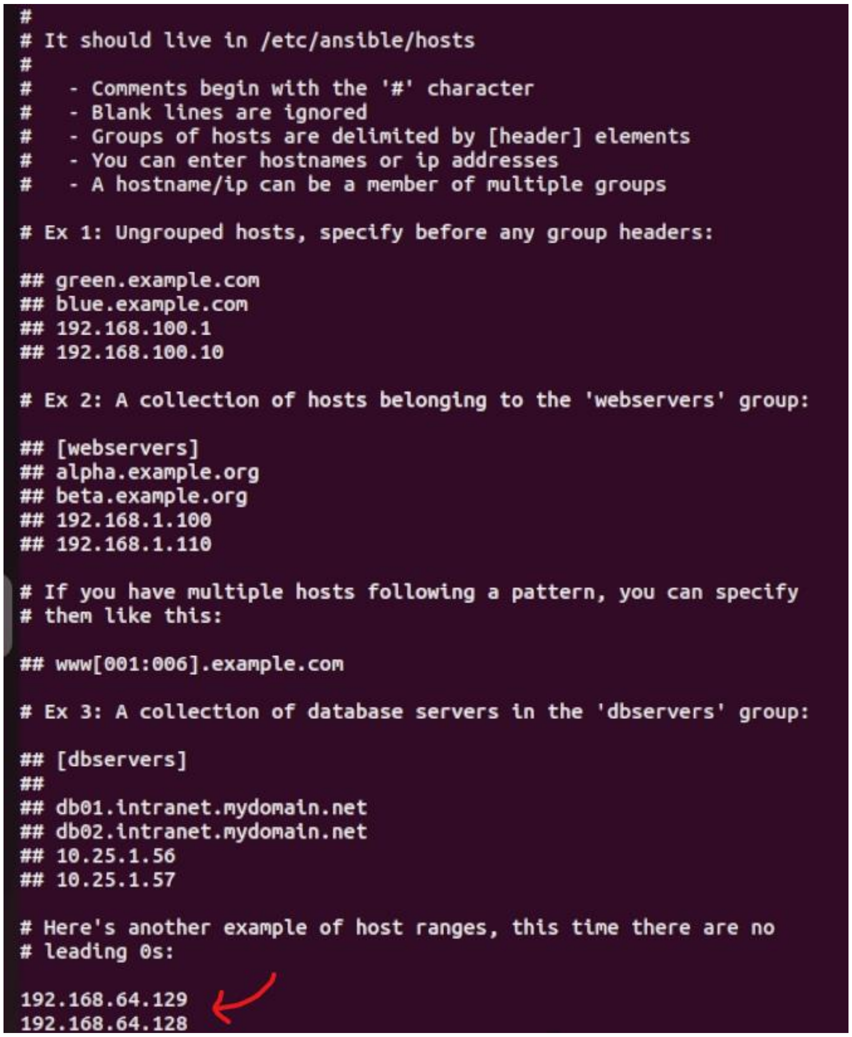
```
loi@loi-virtual-machine:~/demoAnsible$ ansible -m ping all
192.168.64.128 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
192.168.64.129 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```

- **Installing source code, database of website**
  - Dynamic website we refer to on github with link : https://github.com/sahedalomsumit/ecommersha3wp.git
  - Overview of the components that make up the website



```
winofloi@winofloi-virtual-machine:~/demo/ecommersha3wp$ ls
about      contact-us  my-account-2      README.md  wp-includes
cart-2     feed        product           store      wp-json
comments   index.html  product-category  wp-content
winofloi@winofloi-virtual-machine:~/demo/ecommersha3wp$
```

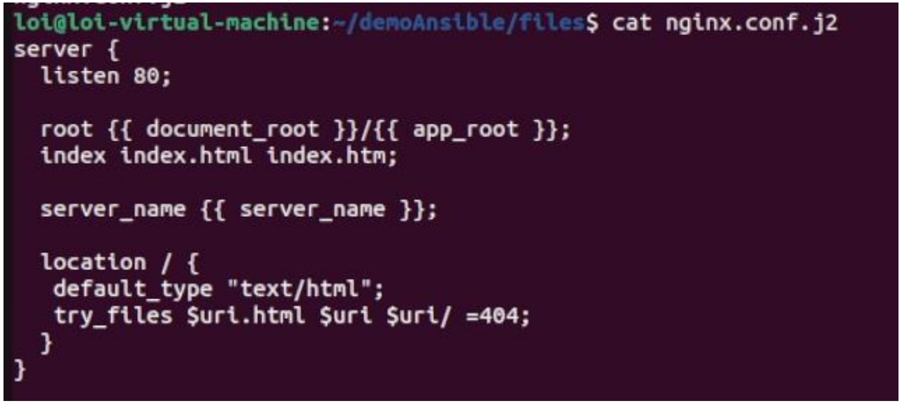
- **Inventory configuration**
  - List machines that will be deployed to dynamic website



```
#
# 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
## 192.168.100.10

# Ex 2: A collection of hosts belonging to the 'webservers' group:

## [webservers]
## alpha.example.org
## beta.example.org
## 192.168.1.100
## 192.168.1.110

# If you have multiple hosts following a pattern, you can specify
# them like this:

## www[001:006].example.com

# Ex 3: A collection of database servers in the 'dbservers' group:

## [dbservers]
##
## db01.intranet.mydomain.net
## db02.intranet.mydomain.net
## 10.25.1.56
## 10.25.1.57

# Here's another example of host ranges, this time there are no
# leading 0s:

192.168.64.129
192.168.64.128
```

- Remote host IP : **192.168.64.128**
- Ansible server IP : **192.168.64.129**

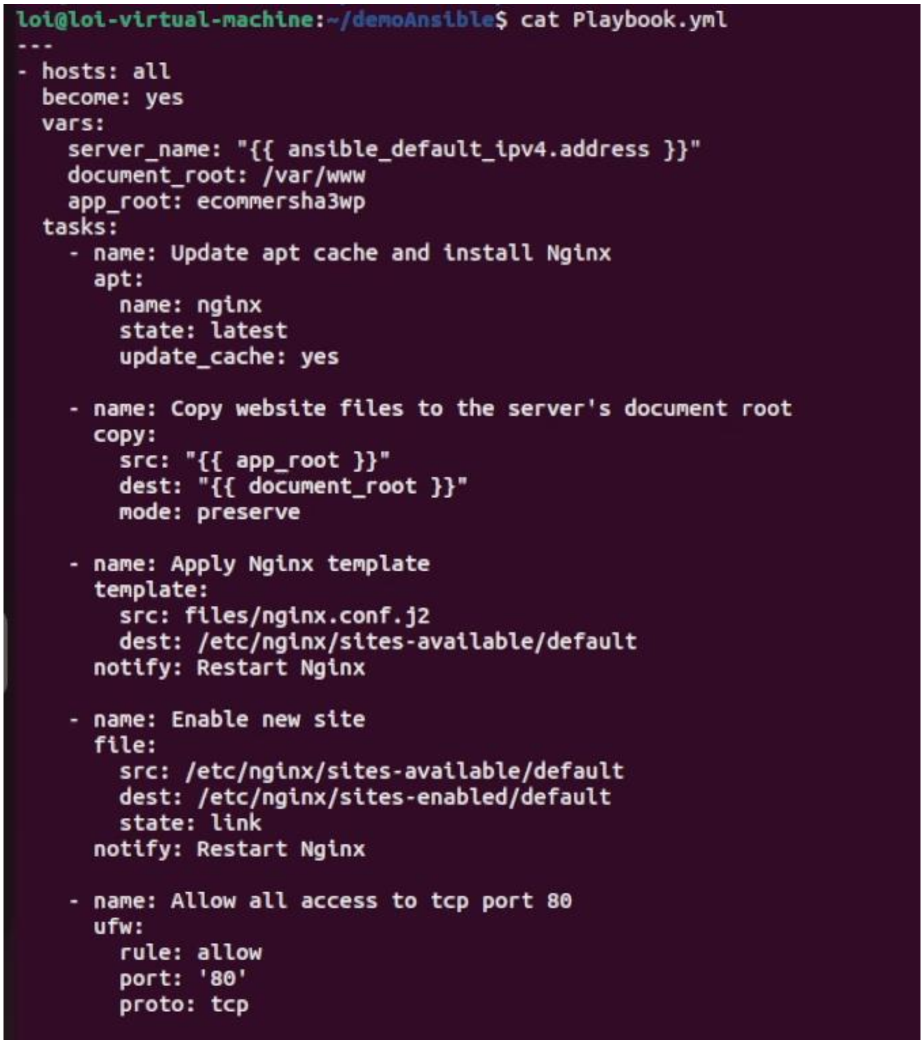
- **Nginx web server configuration :**



```
loi@loi-virtual-machine:~/demoAnsible/files$ cat nginx.conf.j2
server {
  listen 80;

  root {{ document_root }}/{{ app_root }};
  index index.html index.htm;

  server_name {{ server_name }};

  location / {
   default_type "text/html";
   try_files $uri.html $uri $uri/ =404;
  }
}
```

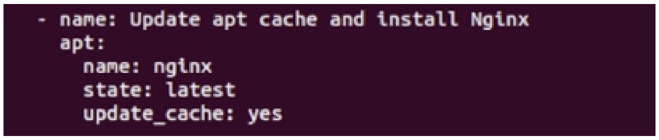
  - This template file contains an Nginx server block configuration for Dynamic website. It uses three variables**: document_root, app_root, and server_name.**

- **Setup playbook file to deploy tasks on remote hosts**



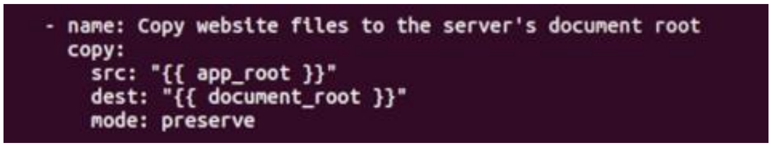
```
loi@loi-virtual-machine:~/demoAnsible$ cat Playbook.yml
---
- hosts: all
  become: yes
  vars:
    server_name: "{{ ansible_default_ipv4.address }}"
    document_root: /var/www
    app_root: ecommersha3wp
  tasks:
    - name: Update apt cache and install Nginx
      apt:
        name: nginx
        state: latest
        update_cache: yes

    - name: Copy website files to the server's document root
      copy:
        src: "{{ app_root }}"
        dest: "{{ document_root }}"
        mode: preserve

    - name: Apply Nginx template
      template:
        src: files/nginx.conf.j2
        dest: /etc/nginx/sites-available/default
      notify: Restart Nginx

    - name: Enable new site
      file:
        src: /etc/nginx/sites-available/default
        dest: /etc/nginx/sites-enabled/default
        state: link
      notify: Restart Nginx

    - name: Allow all access to tcp port 80
      ufw:
        rule: allow
        port: '80'
        proto: tcp
```

- This playbook starts with the **hosts** definition set to **all** and a become directive that tells Ansible to run all tasks as the **root user** by default (the same as manually running commands with sudo). Within this playbook's var section, we'll create three variables: **server_name, document_root, and app_root**. These variables are used in the Nginx configuration template to set up the domain name or IP address that this web server will respond to, and the full path to where the website files are located on the server
- **Ansible_default_ipv4.address** : contains the remote server's public IP address



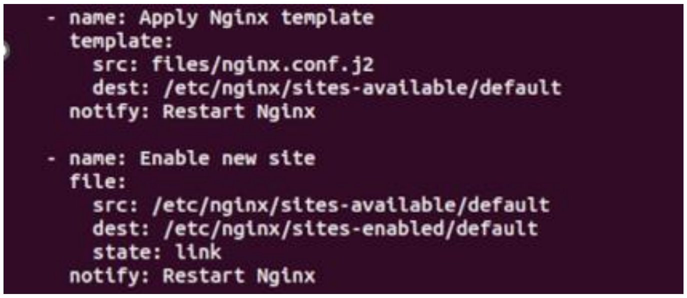
```
    - name: Update apt cache and install Nginx
      apt:
        name: nginx
        state: latest
        update_cache: yes
```

- **Installing Required Packages** : task will update the apt cache and then install the nginx package on remote nodes



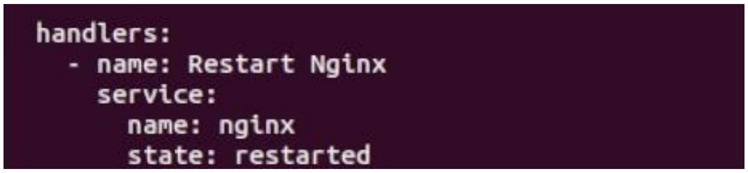
```
- name: Copy website files to the server's document root
  copy:
    src: "{{ app_root }}"
    dest: "{{ document_root }}"
    mode: preserve
```

- **Uploading Website Files to Remote Nodes** : The next task will use the copy built-in module to upload the website files to the remote document root
- Document_root : set the destination on the server where the application folder should be created



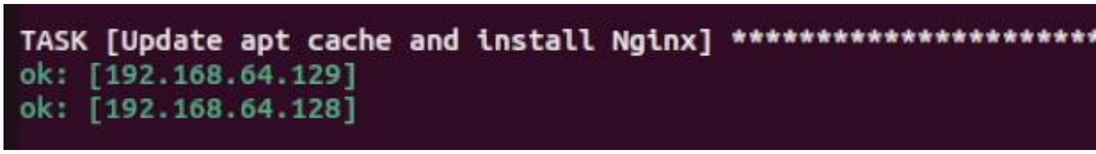
```
- name: Apply Nginx template
  template:
    src: files/nginx.conf.j2
    dest: /etc/nginx/sites-available/default
  notify: Restart Nginx

- name: Enable new site
  file:
    src: /etc/nginx/sites-available/default
    dest: /etc/nginx/sites-enabled/default
    state: link
  notify: Restart Nginx
```

- **Applying and Enabling the Custom Nginx Configuration** : apply the Nginx template that will configure the web server to host my dynamic file and notify the Nginx service for a posterior restart.
- **Creating a Handler for the Nginx Service**.

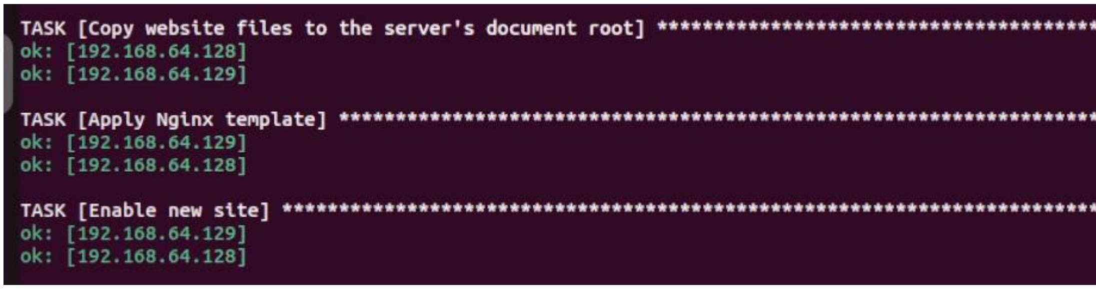


```
handlers:
  - name: Restart Nginx
    service:
      name: nginx
      state: restarted
```

- **Execute the playbook file with the command below to deploy the dynamic website to remote hosts**

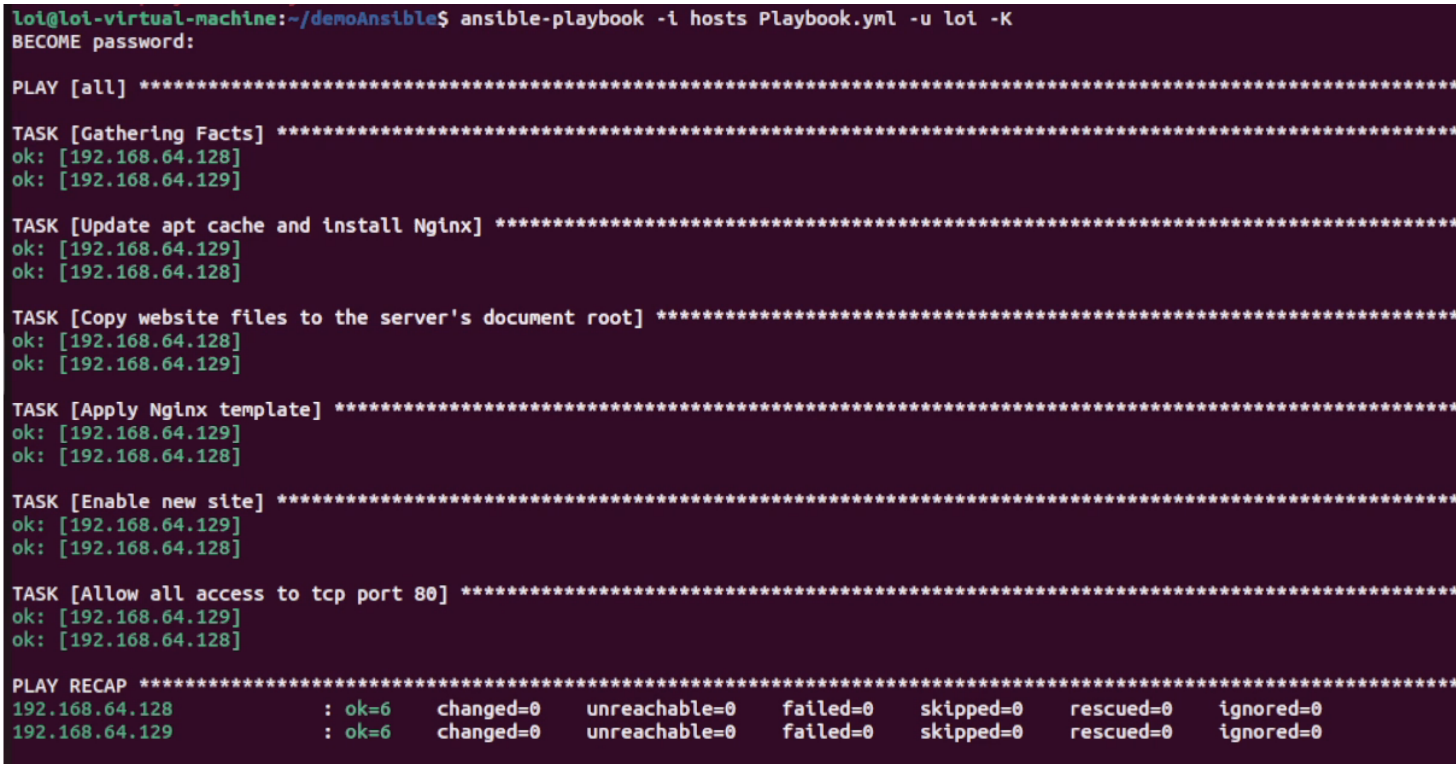


```
TASK [Update apt cache and install Nginx] ***********************
ok: [192.168.64.129]
ok: [192.168.64.128]
```



```
TASK [Copy website files to the server's document root] ****************************************
ok: [192.168.64.128]
ok: [192.168.64.129]

TASK [Apply Nginx template] *******************************************************************
ok: [192.168.64.129]
ok: [192.168.64.128]

TASK [Enable new site] ************************************************************************
ok: [192.168.64.129]
ok: [192.168.64.128]
```

- The process takes quite a while.

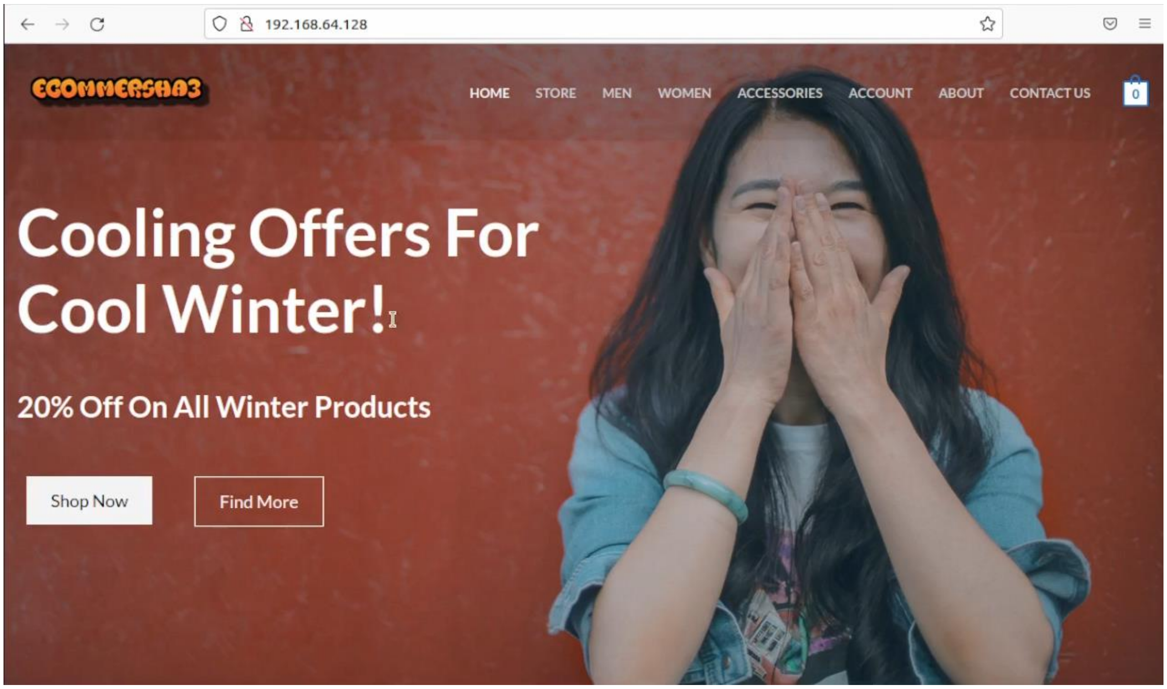


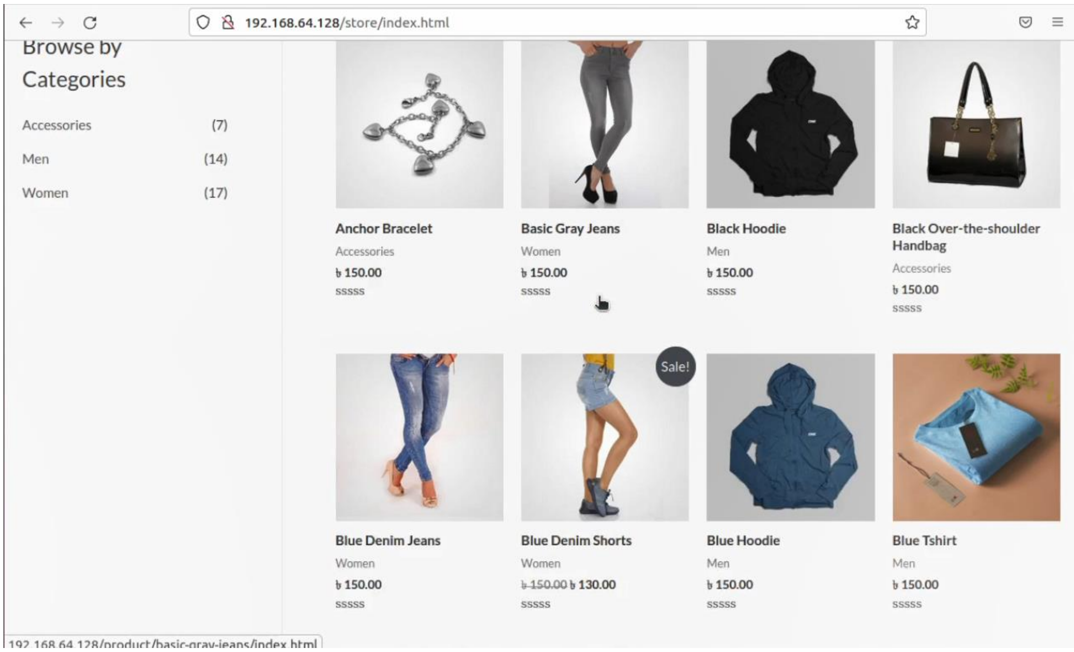
```
loi@loi-virtual-machine:~/demoAnsible$ ansible-playbook -i hosts Playbook.yml -u loi -K
BECOME password:

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

TASK [Gathering Facts] ***************************************************************************
ok: [192.168.64.128]
ok: [192.168.64.129]

TASK [Update apt cache and install Nginx] ********************************************************
ok: [192.168.64.129]
ok: [192.168.64.128]

TASK [Copy website files to the server's document root] *****************************************
ok: [192.168.64.128]
ok: [192.168.64.129]

TASK [Apply Nginx template] **********************************************************************
ok: [192.168.64.129]
ok: [192.168.64.128]

TASK [Enable new site] ***************************************************************************
ok: [192.168.64.129]
ok: [192.168.64.128]

TASK [Allow all access to tcp port 80] ***********************************************************
ok: [192.168.64.129]
ok: [192.168.64.128]

PLAY RECAP ***************************************************************************************
192.168.64.128             : ok=6    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.64.129             : ok=6    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

- Successful implementation

III. Result & Conclusion

- **Remote Host (192.168.64.128)**





192.168.64.128/store/index.html

Browse by Categories

Accessories	(7)
Men	(14)
Women	(17)

Anchor Bracelet
Accessories
৳ 150.00
sssss

Basic Gray Jeans
Women
৳ 150.00
sssss

Black Hoodie
Men
৳ 150.00
sssss

Black Over-the-shoulder Handbag
Accessories
৳ 150.00
sssss

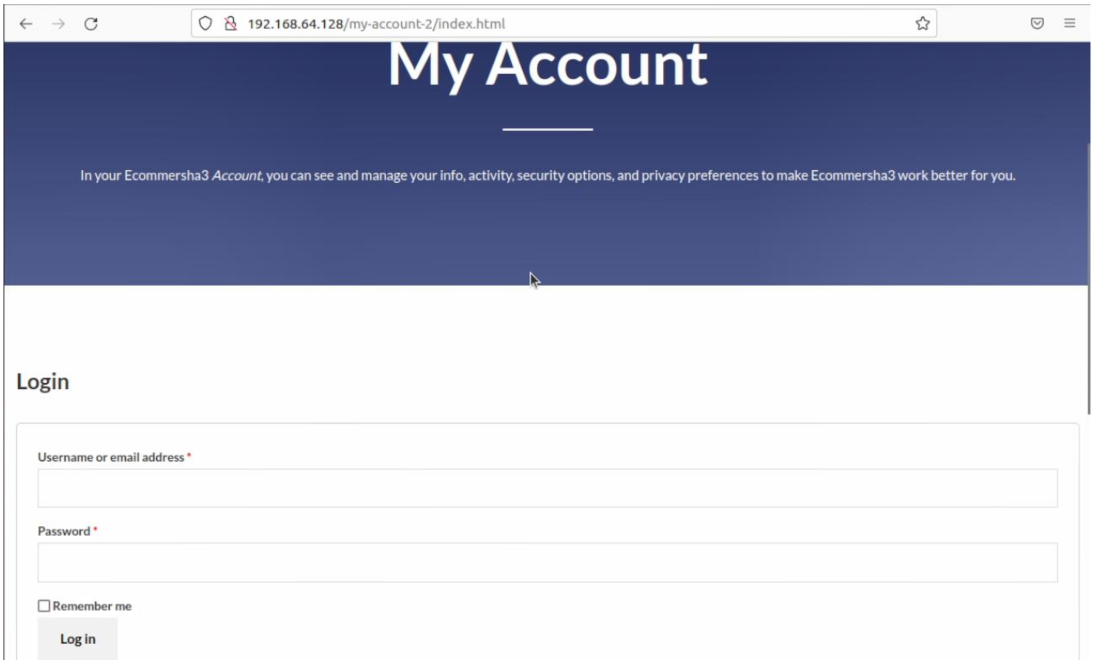
Blue Denim Jeans
Women
৳ 150.00
sssss

Blue Denim Shorts
Women
~~৳ 150.00~~ ৳ 130.00
sssss

Blue Hoodie
Men
৳ 150.00
sssss

Blue Tshirt
Men
৳ 150.00
sssss

192.168.64.128/product/basic-gray-jeans/index.html



192.168.64.128/my-account-2/index.html

My Account

In your Ecommersha3 *Account*, you can see and manage your info, activity, security options, and privacy preferences to make Ecommersha3 work better for you.

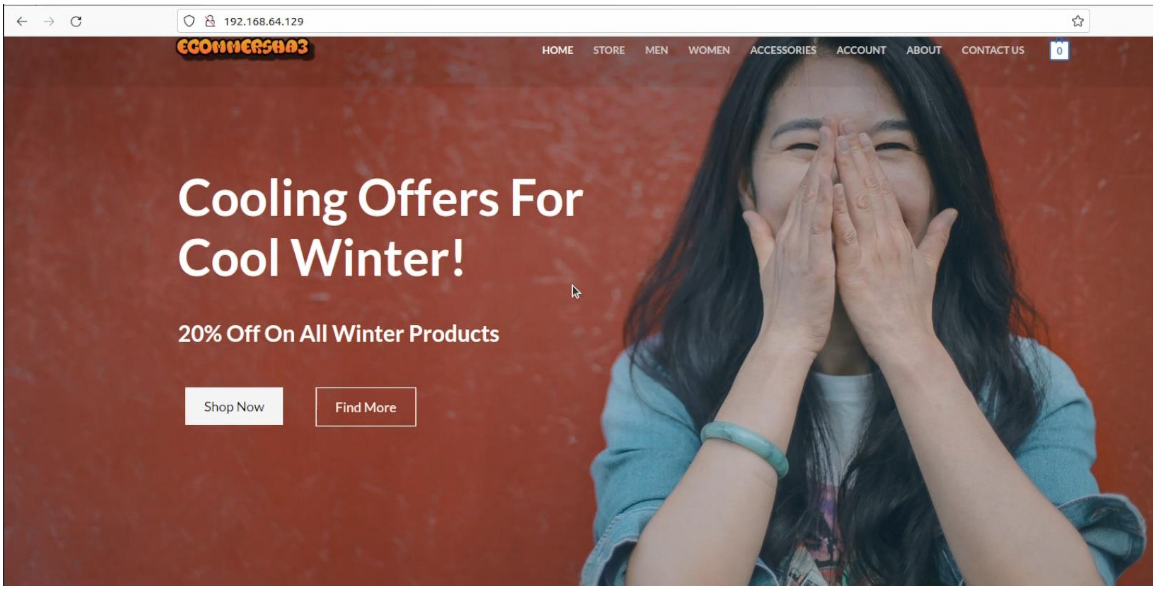
Login

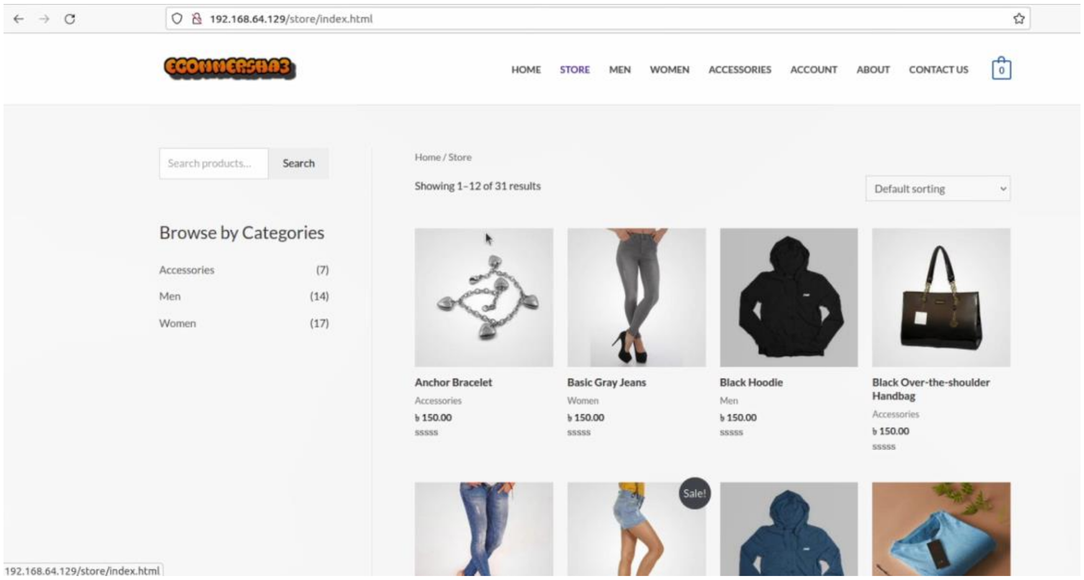
Username or email address *

Password *

□ Remember me

- **Ansible Server (192.168.64.129)**





- **About website**
  - Sales website
  - There are login functions
  - There are many categories of user interaction
- **Conclusion**
  - Ansible easy to start and use from day one, without the need for any special coding skills.
  - Simple deployment workflow without any extra agents.
  - Here, we learned what Ansible is and how to install the Ansible management tool
  - Ansible makes your life easier by automating administration tasks and helps you become better DevOps professional.

B. ASSIGNMENT TABLE.

1. Task :

Task	Name	Completion schedule
Project implementation ,record video , Presentation of the Demo, write and edit the filnal report, make slide	Phạm Ngọc Lợi	100%
Presentation of the introduction, write and edit the final report, make slide	Trần Đặng Hồng Loan	100%
Presentation of installation, edit video demo, make slide	Phạm Văn Thời	100%
Learn theory, collect materials.	Tất cả thành viên	100%

2. Self assessments:

Point Skill	1	2	3	4	
Report(1 point)			X		0.75
Present(1 point)				X	1
Theory(2 point)				X	1.75
Demo(5 point)				X	4.5
Total/9					8

HẾT