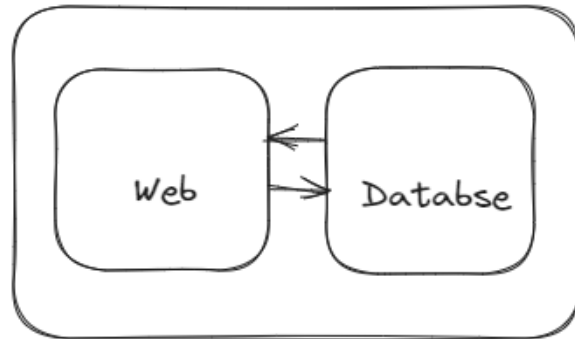




Ansible Project

Nafiur Rashid
ID- 00-30080

Project Objective:

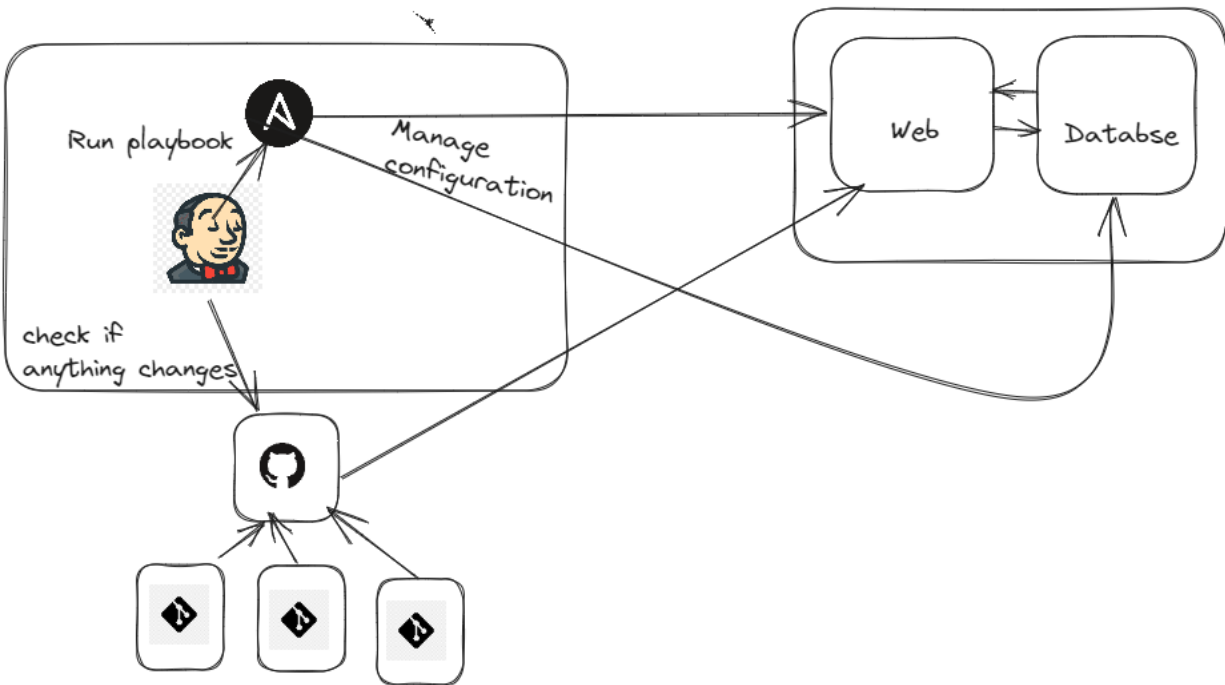


The objective is to deploy CI/CD to a **two tier** website application(**LAMP stack**). But our approach is to make the development journey smooth by using following DevOps Tools

- ☐ **Shell scripting**- To make the task easier
- ☐ **Ansible**- To configure management from a master server
- ☐ **Jenkins**- To facilitate CI/CD
- ☐ **Github & Git**- As Source code management and CI

Expected Blueprint:

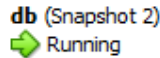
We will check from jenkins if any new “commit” happened in github. Yes we will run the ansible playbook. Ansible playbook will configure , clone and run the servers.



Step 1:Virtual machine setup

First set up 3 CentOS machine with following configuration (atleast)

- Memory: 10 GB
- Ram: 2 GB
- CPU: 2 Core



Step 2: Master machine set up

Following task has to be done while setting the master node

- ☐ Update yum
- ☐ Install vim, curl, wget, open-vm tools
- ☐ Install Java (Open jdk)
- ☐ Disable SELINUX
- ☐ Disable Firewall
- ☐ Install jenkins
- ☐ Enable & start jenkins
- ☐ Install Git
- ☐ Install Ansible
- ☐ Find IP of the master machine and jenkins password

We do all these tasks using a single shell script. Here it is.

[illegible]

```
sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/selinux/config  
echo "disabling Firewall>>>>>>>>>>>>>>>"  
systemctl disable firewalld  
systemctl stop firewalld  
echo "installing jenkins>>>>>>>>>>>>>>>"  
sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo  
sudo rpm --import https://pkg.jenkins.io/redhat/jenkins.io-2023.key  
yum install jenkins -y  
echo "enabling jenkins>>>>>>>>>>>>>>>"  
systemctl enable jenkins  
systemctl start jenkins  
echo "installing git>>>>>>>>>>>>>>>"  
yum install git -y  
echo "installing ansible>>>>>>>>>>>>>>>"  
sudo yum install epel-release  
yum install ansible -y  
echo "here is your IP address>>>>>>>>>>>>>>>"  
hostname -I | awk '{print $2}'  
echo "finding admin password of jenkins>>>>>>>>>>>>>>>"  
echo "please copy following password>>>>>>>>>>>>>>>"  
cat /var/lib/jenkins/secrets/initialAdminPassword
```

Step 3: Two Client machine setup

The following task has to be done to set up client machines

- ☐ Update yum
- ☐ Install vim, curl, wget, open-vm tools
- ☐ Disable SELINUX
- ☐ Disable Firewall
- ☐ Stop Firewall
- ☐ Know the Ip address

Let's use shell script to do it.

```
#!/bin/sh
```

[illegible]

Step 4: Connect two client machines with master through ssh

As we note down the IP of all the three machine now we use it to configure ssh

- ☐ Mention the hosts for ansible
- ☐ Configure the ssh for master and two clients

```
#!/bin/sh
# Author : Nafiur Rashidvim /etc/ansible/hosts
echo "[web]
Your_web_server_ip
[db]
Your_db_server_ip
">> /etc/ansible/hosts
ssh-keygen
ssh-copy-id root@Your_web_server-ip #[Web Server IP]
ssh-copy-id root@Your_web_server-ip #[Database Server IP]
```

Step 5: Github setup

- i) Generate ssh key using the following command on host machine:

```
ssh-keygen -t rsa -C "github-access" -f id_rsa
```

Add the containing public key in id_rsa.pub file to your github account

- ii) Open a repository in the github

Step 6: Ansible playbook setup

i) In the master machine we will have to write required play books. In our case we will write in `/opt/playbook/` directory

```
[root@localhost ~]# cd /opt/  
[root@localhost opt]# mkdir playbook
```

Let's create two yml file here

```
[root@localhost playbook]# vim db.yml
```

```
---  
- name: DB Service  
  hosts: Your_db_IP_address  
  roles:  
    - db
```

```
is directories, 10 files  
[root@localhost playbook]# vim web.yml
```

```
---  
- name: Web Service  
  hosts: 192.168.56.113  
  roles:  
    - web
```

ii) Hit the following command for ansible roles

```
[root@localhost playbook]# ansible-galaxy init db --offline
```

Let's check the folder structure

```

├── db
│   ├── defaults
│   │   └── main.yml
│   ├── files
│   │   ├── db-load-script.sql
│   │   └── my.cnf
│   ├── handlers
│   │   └── main.yml
│   ├── meta
│   │   └── main.yml
│   ├── README.md
│   ├── tasks
│   │   └── main.yml
│   ├── templates
│   ├── tests
│   │   ├── inventory
│   │   └── test.yml
│   └── vars
│       └── main.yml
└── db.yml

```

```
[root@localhost playbook]# ansible-galaxy init web --offline
```

Again, we will get the same folder structure

```

├── web
│   ├── defaults
│   │   └── main.yml
│   ├── files
│   ├── handlers
│   │   └── main.yml
│   ├── meta
│   │   └── main.yml
│   ├── README.md
│   ├── tasks
│   │   └── main.yml
│   ├── templates
│   ├── tests
│   │   ├── inventory
│   │   └── test.yml
│   └── vars
│       └── main.yml
└── web.yml

```

iii) Let's work on db first

```
vim db/vars/main.yml
```

```

---
mysql_port: Your_expected_port_number
dbname: Your_Database_name
dbuser: Your_Database_user_name

```



```
dbpassword: Your_Database_password
```

```
[root@localhost files]# vim /opt/playbook/db/files/my.cnf
```

```
[mysqld]
bind-address=0.0.0.0
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
symbolic-links=0
[mysqld_safe]
log-error=/var/log/mariadb/mariadb.log
pid-file=/var/run/mariadb/mariadb.pid
!includedir /etc/my.cnf.d
```

```
[root@localhost files]# vim /opt/playbook/db/files/db-load-script.sql
```

White the sql script here. But before the script add these lines

```
GRANT ALL PRIVILEGES ON *.* TO
'Your_Database_user_name'@'Your_Database_user_name' IDENTIFIED BY
'Your_Database_password' WITH GRANT OPTION;
FLUSH PRIVILEGES;
USE Your_Database_name;
```

```
[root@localhost files]# vim /opt/playbook/db/tasks/main.yml
```

```
---
# tasks file for db
- name: Installation Services
  yum:
    name:
      - libselinux-python
      - libsemanage-python
      - mariadb-server
      - MySQL-python
      - php-mysql
    state: installed
  tags: install
- name: Start firewalld
  service: name=firewalld state=started enabled=yes
  tags: start firewalld
```

```

- name: Insert firewall rule for mysql
  firewallld: port={{ mysql_port }}/tcp permanent=true state=enabled
  immediate=yes
  tags: enable mysql port
- name: Restart firewallld
  service: name=firewalld state=reloaded enabled=yes
  tags: restart firewallld
- name: Copy Mysql configuration file
  copy: src=files/my.cnf dest=/etc/my.cnf
  tags: mysql conf copy
- name: Start MariaDB Service
  service: name=mariadb state=started enabled=yes
  tags: start mariadb
- name: Create Application Database
  mysql_db: name={{ dbname }} state=present
  tags: create database
- name: Create Application DB User
  mysql_user: name={{ dbuser }} password={{ dbpassword }} priv=*.*:ALL
  host='192.168.56.117' state=present
  tags: create user
- name: Move db-load-script to db host
  copy:
    src: files/db-load-script.sql
    dest: /tmp/db-load-script.sql
  tags: copy sql
- name: Load Inventory Data
  shell: mysql -f < /tmp/db-load-script.sql
  tags: run sql

```

Now run and check if it works

```
[root@localhost playbook]# ansible-playbook db.yml
```

output:

```
PLAY [DB Service] *****
TASK [Gathering Facts] *****
ok: [192.168.56.117]

TASK [db : Installation Services] *****
ok: [192.168.56.117]

TASK [db : Start firewall] *****
ok: [192.168.56.117]

TASK [db : Insert firewall rule for mysql] *****
ok: [192.168.56.117]

TASK [db : Restart firewall] *****
changed: [192.168.56.117]

TASK [db : Copy Mysql configuration file] *****
ok: [192.168.56.117]

TASK [db : Start MariaDB Service] *****
ok: [192.168.56.117]

TASK [db : Create Application Database] *****
ok: [192.168.56.117]

TASK [db : Create Application DB User] *****
[WARNING]: Module did not set no_log for update_password
ok: [192.168.56.117]

TASK [Move db-load-script to db host] *****
ok: [192.168.56.117]

TASK [db : Load Inventory Data] *****
changed: [192.168.56.117]

PLAY RECAP *****
192.168.56.117      : ok=11  changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

iv) Playbook for web server:

```
[root@localhost vars]# vim /opt/playbook/web/vars/main.yml
```

```
---
# vars file for web
httpd_port: Your_port_number_for_apache
mysql_port: Your_port_number_for_mysql
repository: https://github.com/github_username/github_repo_name.git
```

```
[root@localhost vars]# vim /opt/playbook/web/tasks/main.yml
```

```
# tasks file for web
---
- name: Installation Services
  yum:
    name:
      - libselinux-python
      - libsemanage-python
      - httpd
      - git
      - php
```

```

- php-mysql
  state: installed
  tags: install

- name: Start firewalld
  service: name=firewalld state=started enabled=yes
  tags: start firewalld

- name: Insert firewalld rule for httpd
  firewalld: port={{ httpd_port }}/tcp permanent=true state=enabled
immediate=yes
  tags: enable httpd port

- name: insert firewalld rule for mysql
  firewalld: port={{ mysql_port }}/tcp permanent=true state=enabled
immediate=yes
  tags: enable mysql port

- name: Set index.php as the default page
  replace:
    path: /etc/httpd/conf/httpd.conf
    regexp: 'DirectoryIndex index.html'
    replace: '#DirectoryIndex index.html \nDirectoryIndex index.php'
  tags: rename html file

- name: http service state
  service: name=httpd state=started enabled=yes
  tags: httpd start

- name: Copy the code from repository
  git: repo={{ repository }} dest=/var/www/html/ force=yes
  tags: clone

- name: replace ip in index.php file
  command: sed -i 's/172.20.1.101/Your_database_machine_IP/g'
/var/www/html/config.php
  tags: replace IP

```

Lets Run it.

```
[root@localhost playbook]# ansible-playbook web.yml
```

```
[root@localhost playbook]# ansible-playbook web.yml
[WARNING]: The plugin filter file, /etc/ansible/plugin_filters.yml does not exist. Skipping.

PLAY [Web Service] *****
TASK [Gathering Facts] *****
ok: [192.168.56.113]

TASK [web : Installation Services] *****
ok: [192.168.56.113]

TASK [web : Start firewalld] *****
ok: [192.168.56.113]

TASK [web : Insert firewalld rule for httpd] *****
ok: [192.168.56.113]

TASK [web : insert firewalld rule for mysql] *****
ok: [192.168.56.113]

TASK [web : Set index.php as the default page] *****
changed: [192.168.56.113]

TASK [web : http service state] *****
ok: [192.168.56.113]

TASK [web : Copy the code from repository] *****
ok: [192.168.56.113]

TASK [web : replace ip in index.php file] *****
[WARNING]: Consider using the replace, lineinfile or template module rather than running 'sed'. If you need to use command because
replace, lineinfile or template is insufficient you can add 'warn: false' to this command task or set 'command_warnings=False' in
ansible.cfg to get rid of this message.
changed: [192.168.56.113]

PLAY RECAP *****
```

Step 7:Jenkins setup

Here ansible-Version_mentioned_in_jenkins is ansible-2.9.27

```
pipeline {
    agent any

    stages {
        stage('Git Trigger 🙌') {
            steps {
                git branch: 'main', url:
'https://github.com/github_username/github_repo_name.git'
            }
        }

        stage ('Run DB Playbook 💾'){
            steps{
                ansiblePlaybook become: true, credentialsId:
'ansible-master-controller',
                disableHostKeyChecking: true, installation:
'ansible-Version_mentioned_in_jenkins', inventory: '/etc/ansible/hosts',
                playbook: '/opt/playbook/db.yml'
            }
        }
    }
}
```

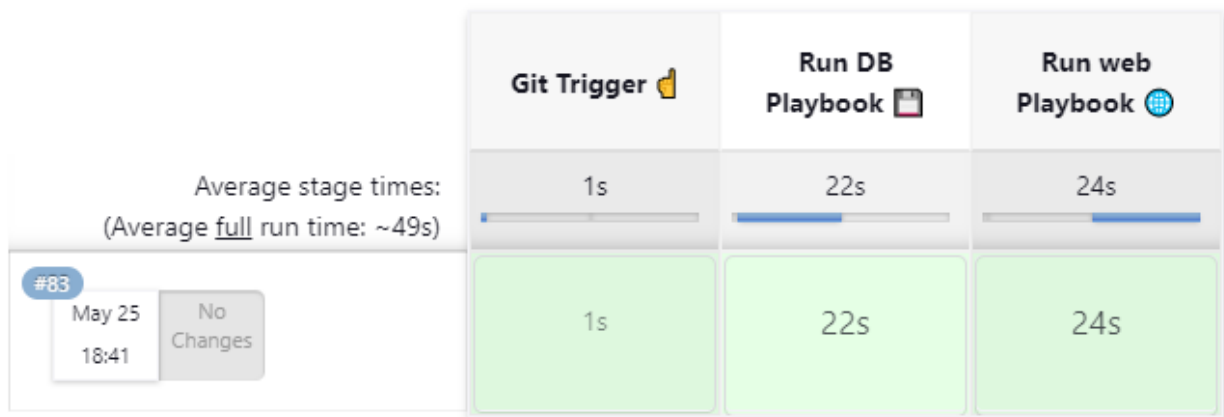
```

stage ('Run web Playbook 🌐'){
  steps{
    ansiblePlaybook become: true, credentialsId:
'ansible-master-controller',
    disableHostKeyChecking: true, installation:
'ansible-2.9.27', inventory: '/etc/ansible/hosts',
    playbook: '/opt/playbook/web.yml'
  }
}
}

```

Build the jenkins pipeline.

Stage View



Here is an example of console output



Status

</> Changes

Console Output

View as plain text

Edit Build Information

Delete build '#83'

Git Build Data

Restart from Stage

Replay

Pipeline Steps

Workspaces

Previous Build

Console Output

```
Started by user Nafiur Rashid
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/pipeline-1
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Git Trigger)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/pipeline-1/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/nafiursan/ansible_project.git # timeout=10
Fetching upstream changes from https://github.com/nafiursan/ansible_project.git
> git --version # timeout=10
> git --version # 'git version 1.8.3.1'
> git fetch --tags --progress https://github.com/nafiursan/ansible_project.git +refs/heads/*:refs/remotes/origin/* #
timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision e5df6851f64460de90a4b84a90fb0aa5b8e951d (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f e5df6851f64460de90a4b84a90fb0aa5b8e951d # timeout=10
> git branch -a -v --no-abbrev # timeout=10
> git branch -D main # timeout=10
> git checkout -b main e5df6851f64460de90a4b84a90fb0aa5b8e951d # timeout=10
Commit message: "Update push.txt"
> git rev-list --no-walk e5df6851f64460de90a4b84a90fb0aa5b8e951d # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Run DB Playbook)
[Pipeline] ansiblePlaybook
[pipeline-1] $ /usr/bin/ansible-playbook /opt/playbook/db.yml -i /etc/ansible/hosts -b --become-user root --private-key
/var/lib/jenkins/workspace/pipeline-1/ssh1871361991546882899.key -u root
[WARNING]: The plugin filter file, /etc/ansible/plugin_filters.yml does not
exist. Skipping.

PLAY [DB Service] *****

TASK [Gathering Facts] *****
ok: [192.168.56.117]

TASK [db : Installation Services] *****
ok: [192.168.56.117]

TASK [db : Start firewall] *****
ok: [192.168.56.117]

TASK [db : Insert firewall rule for mysql] *****
ok: [192.168.56.117]

TASK [db : Restart firewall] *****
changed: [192.168.56.117]

TASK [db : Copy Mysql configuration file] *****
ok: [192.168.56.117]

TASK [db : Start MariaDB Service] *****
ok: [192.168.56.117]

TASK [db : Create Application Database] *****
ok: [192.168.56.117]

TASK [db : Create Application DB User] *****
[WARNING]: Module did not set no_log for update_password
ok: [192.168.56.117]

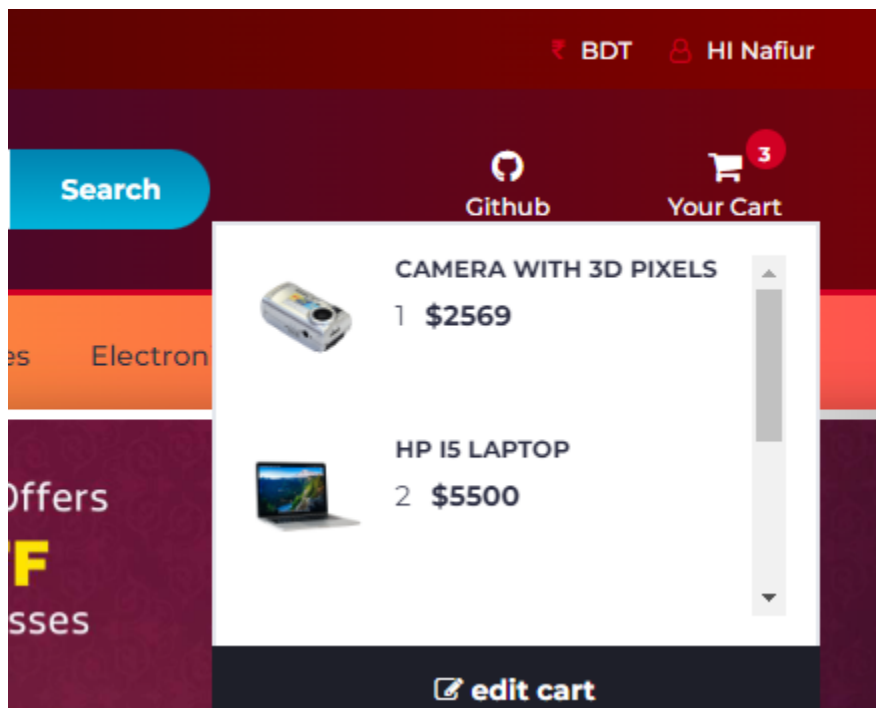
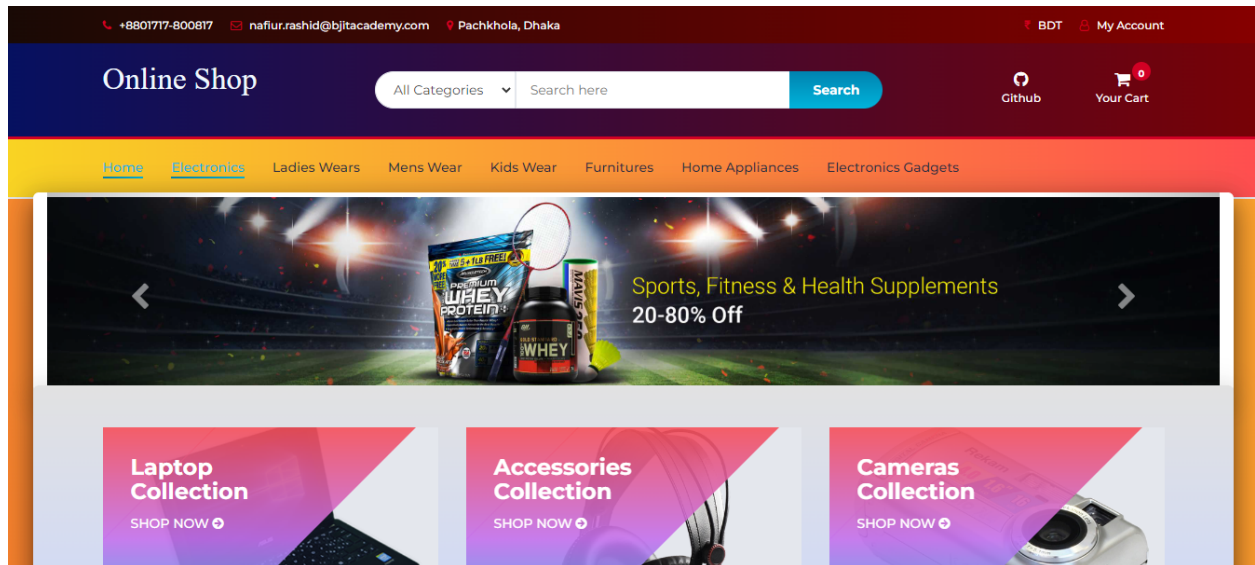
TASK [Move db-load-script to db host] *****
ok: [192.168.56.117]

TASK [db : Load Inventory Data] *****
changed: [192.168.56.117]

PLAY RECAP *****
192.168.56.117 : ok=11 changed=2 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Run web Playbook)
[Pipeline] ansiblePlaybook
[pipeline-1] $ /usr/bin/ansible-playbook /opt/playbook/web.yml -i /etc/ansible/hosts -b --become-user root --private-key
/var/lib/jenkins/workspace/pipeline-1/ssh545915875382813894.key -u root
[WARNING]: The plugin filter file, /etc/ansible/plugin_filters.yml does not
exist. Skipping.
```

Step 8: Hit the web url and enjoy seamless CI/CD experience



References:

1. <https://docs.github.com/en/authentication/connecting-to-github-with-ssh>
2. <https://medium.com/@kelom.x/ansible-mysql-installation-2513d0f70faf>
3. <https://www.jenkins.io/doc/book/installing/>
4. <https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>
5. <https://www.digitalocean.com/community/tutorials/how-to-use-ansible-to-install-and-set-up-lamp-on-ubuntu-18-04>