  
  
  
  
  
  
  
  
  
**Ansible Project**  
  
  
  
  
  
  
  
Nafiur Rashid

ID- 00-30080

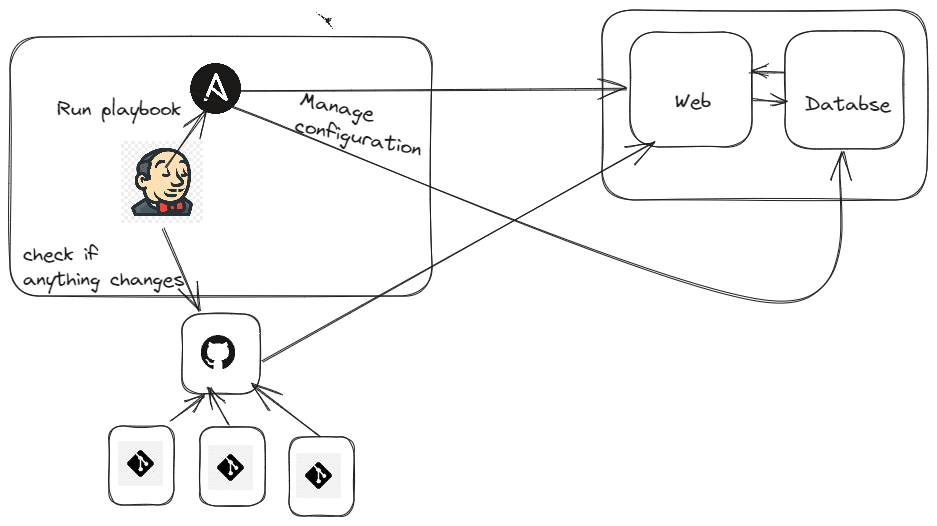
# **Project Objective:** The objective is to deploy CI/CD to a **two tier** website application(**LAMP stack**). But our approach is to make the development journey smooth by using following DevOps Tools

## **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.

#!/bin/sh

# Author : Nafiur Rashid

echo "Update yum>>>>>>>>>>>>>>>>"

yum update -y

echo "installing vim, curl, wget, open-vm-tools>>>>>>>>>>>>>>>>"

yum install vim curl wget open-vm-tools -y

echo "installing java >>>>>>>>>>>>>>>>"

yum install java-11-openjdk.x86\_64 -y

echo "disabling SELINUX>>>>>>>>>>>>>>>>"

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

# Author : Nafiur Rashid

echo "Update yum>>>>>>>>>>>>>>>>"

yum update -y

echo "installing vim, curl, wget, open-vm-tools>>>>>>>>>>>>>>>>"

yum install vim curl wget open-vm-tools -y

echo "disabling SELINUX>>>>>>>>>>>>>>>>"

sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/selinux/config

echo "disabling Firewall>>>>>>>>>>>>>>>>"

systemctl disable firewalld

systemctl stop firewalld

echo "here is your IP address>>>>>>>>>>>>>>>>"

hostname -I | awk '{print $2}'

**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\_serve\_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  


Let's create two yml file here



---

- name: DB Service

hosts: Your\_db\_IP\_address

roles:

- db



---

- name: Web Service

hosts: 192.168.56.113

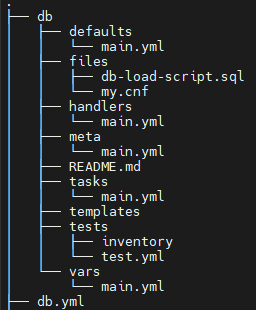
roles:

- web

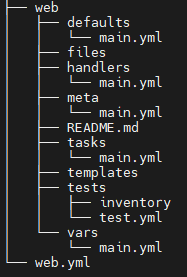
**ii)** Hit the following command for ansible roles



Let's check the folder structure


Again, we will get the same folder structure



**iii)** Let’s work on db first  


—--

mysql\_port: Your\_expected\_port\_number

dbname: Your\_Database\_name

dbuser: Your\_Database\_user\_name

dbpassword: Your\_Database\_password



[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

  
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;



---

# 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 firewalld rule for mysql

firewalld: port={{ mysql\_port }}/tcp permanent=true state=enabled immediate=yes

tags: enable mysql port

- name: Restart firewalld

service: name=firewalld state=reloaded enabled=yes

tags: restart firewalld

- 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  


output:  


**iv)** Playbook for web server:  


---

**# 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



# 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

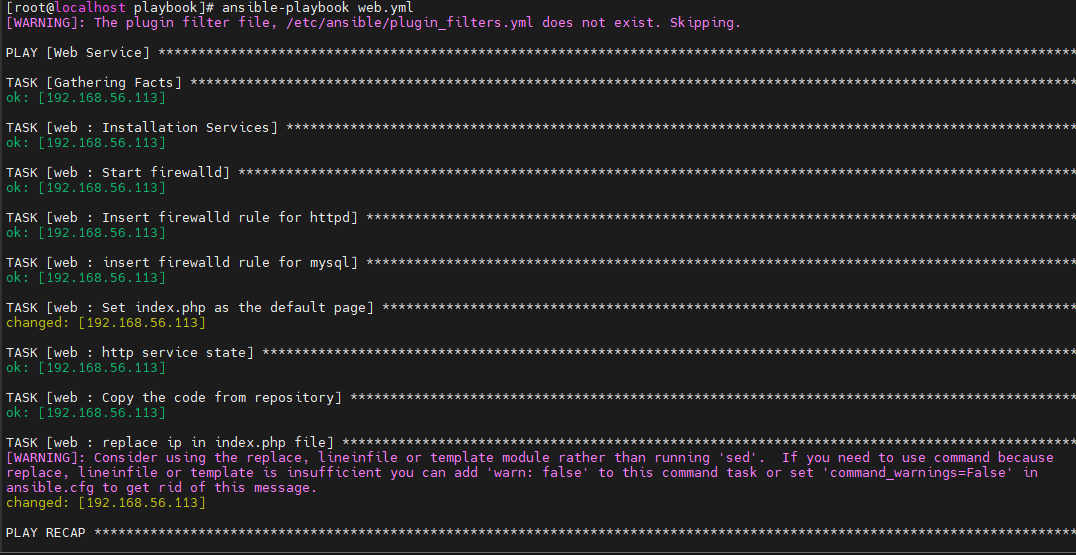
- name: replace ip in index.php file

command: sed -i 's/172.20.1.101/Your\_database\_machine\_IP/g' /var/www/html/config.php

tags: replace IP

Lets Run it.





**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'

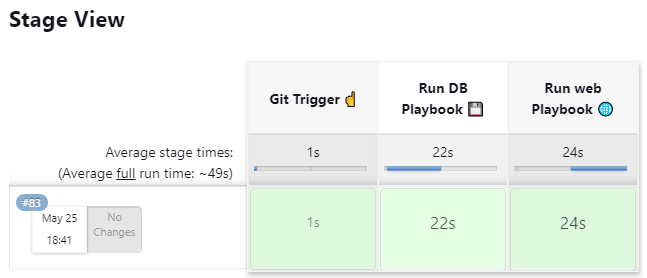
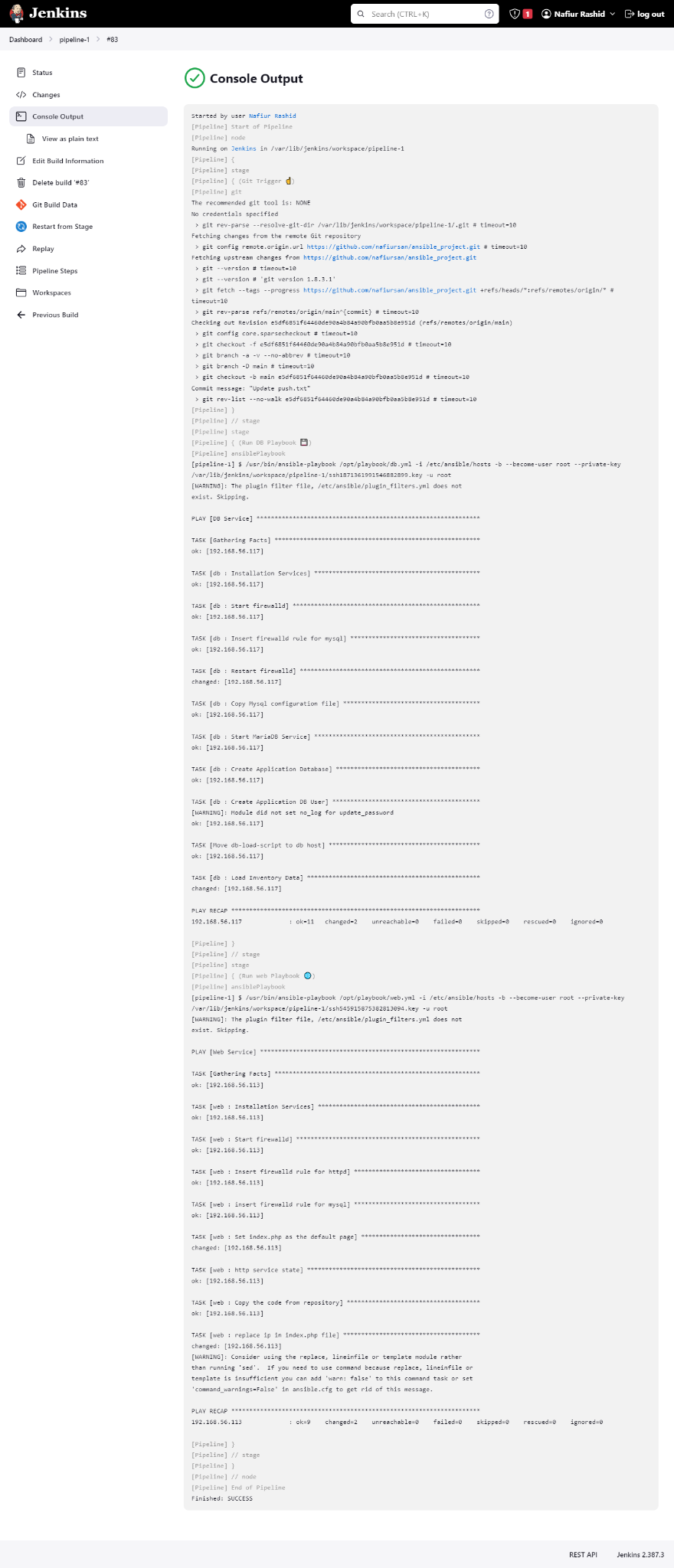
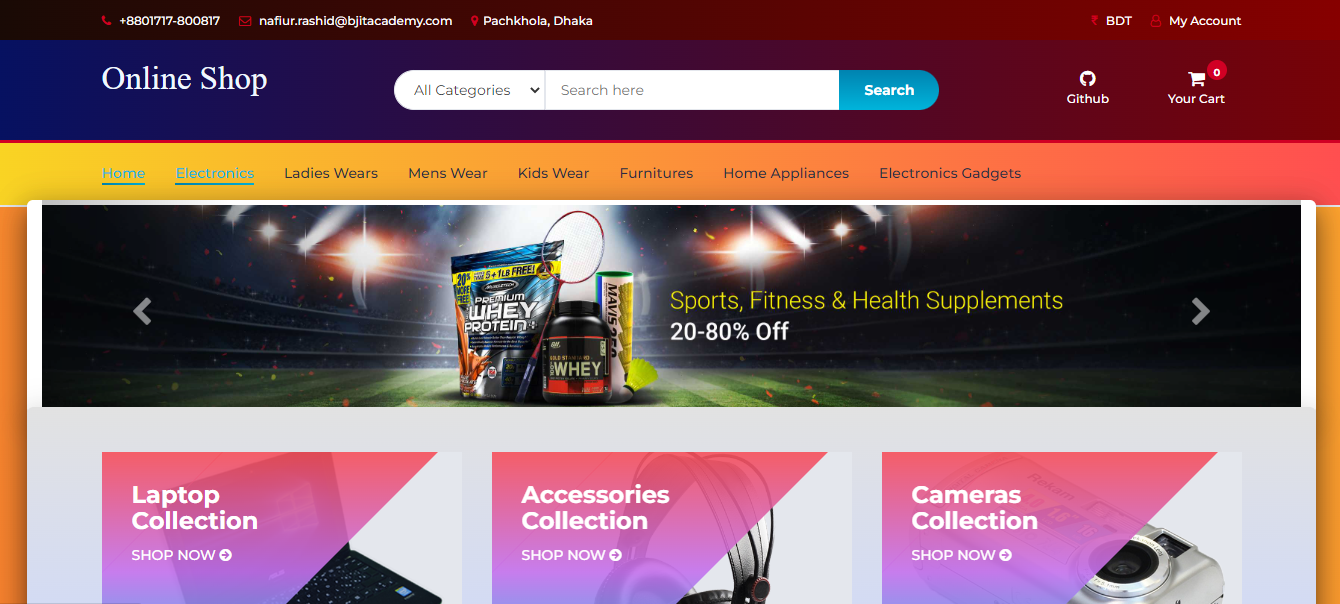
}

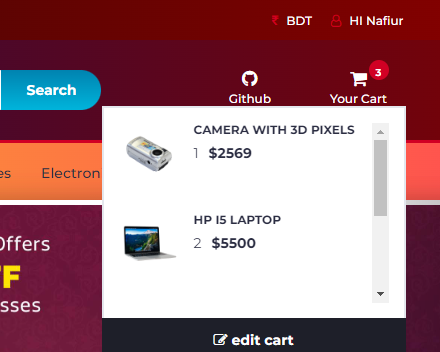
}

}

}

Build the jenkins pipeline.

  
  
Here is an example of console output  
  
  
**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>