1. Introduction:

   This document provides a step-by-step guide to setting up a CI/CD pipeline using Git, Ansible, Jenkins, and separate virtual machines for the master node (Ansible and Jenkins) and the web/DB server. The pipeline involves pulling code from a GitHub repository, triggering Ansible deployment through Jenkins, and deploying the application to the web server.

2. Prerequisites:
   - Three virtual machines: Master node (Ansible and Jenkins), Web Server, and DB Server.
   - Network connectivity between the machines.
   - Administrator access to configure the virtual machines.

3. Setting Up the Master Node (Ansible and Jenkins):

3.1 Master Node Configuration:

- Install the required operating system on the master node (e.g., Centos server).
- Assign a static IP address to the master node for easy access.

3.2 Install Jenkins:

Now we will install Jenkins on the master VM.

1. Jenkins is a java program so firstly we need to install java on the master VM.

```
# yum install java-11-openjdk.x86_64 -y
```

2. To install the latest stable version of Jenkins , we have to add the official Jenkins repository to the system. Execute the below commands to add the key and repo.

```
# yum install wget -y
# 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
```

3. Now install, start and enable jenkins by below commands

```
# yum install jenkins -y
# systemctl start jenkins
# systemctl enable Jenkins
```

3.3 Install Ansible:

Now we will install Ansible on the master VM.

1. Firstly we need to install the epel-release package so that we can install ansible from that package. For this we need to run the below command:

```
# sudo yum install epel-release
```

2. Now we will install ansible by below command:

```
# yum install ansible
```

3.4 Ansible hosts:

We will add ansible clients which is our Web Server VM, Database Server VM ip address in ansible configuration file.

```
# sudo vim /etc/ansible/hosts
[webservers]
192.168.20.15
[dbservers]
192.168.20.14
```

3.5 SSH key copy master to client:

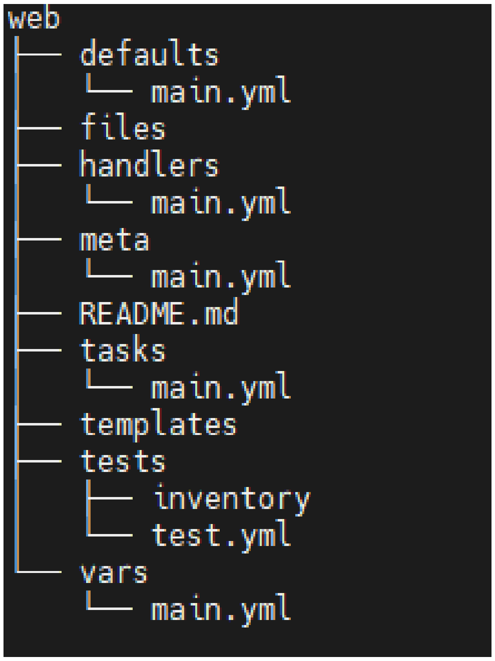
We need to copy the ssh key of the master server in ansible client machines. So that the Ansible server to its nodes communication will be password less.

```
# ssh-keygen
# ssh-copy-id root@192.168.20.225 [Web Server IP]
# ssh-copy-id root@192.168.20.144 [Database Server IP]
```

4. Setting Up the Web Server and DB Server:

4.1 Web Server Configuration:

Now we will create an ansible-playbook to install, start mandatory services in node. Firstly we need to create files (webserver and db server) and a directory (roles) in the same directory. Under the "roles" directory we created another directory which is called "web" and "db". Under this "web" and "db" directory we created four directory (vars, tasks, templates, files). We created "main.yml" files in the vars and tasks directory. In the files of "db" directory we created two files "db-load-script.sql" and "my.cnf".



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

- Now in the "webserver.yml" file we have added the below lines:

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

- **I have mentioned the below lines in "roles/web/tasks/main.yml".**

```
- 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/172.20.1.101/192.168.20.144/g'
/var/www/html/index.php
  tags: replace IP
```

- **I added the below lines in “roles/web/vars/main.yml” file**

```
---
httpd_port: 80
mysql_port: 3306
repository: https://github.com/mdnahidulislam1/Crispy_Kitchen.git
```

4.2 DB Server Configuration:

- Install the required operating system on the DB server (e.g., Centos 7).
- Assign a static IP address to the DB server for easy access.
- Install and configure the desired database server software (e.g., MySQL).
  - I have mentioned the below lines in “roles/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 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: restarted 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='192.168.20.144' 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
```

- **I have mentioned the below lines in “roles/db/vars/main.yml”**.

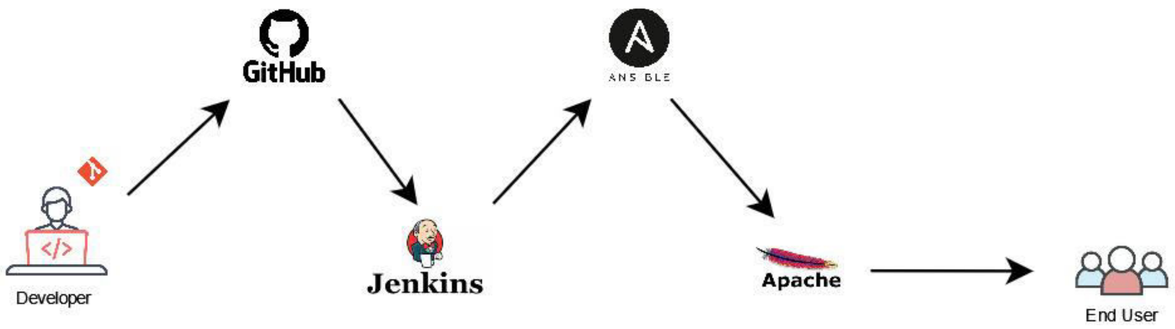
```
---
mysql_port: 3306
dbname: ecomdb
dbuser: ecomuser
dbpassword: ecompassword
```

- **I have mentioned the below lines in “roles/db/files/db-load-script.sql”**

```
GRANT ALL PRIVILEGES ON *.* TO 'ecomuser'@'192.168.20.225' IDENTIFIED BY
'ecompassword' WITH GRANT OPTION;
FLUSH PRIVILEGES;
USE ecomdb;
CREATE TABLE products (id mediumint(8) unsigned NOT NULL
auto_increment,Meal_time_name varchar(255) default NULL,Name varchar(255)
default NULL,Price varchar(255) default NULL, ImageUrl varchar(255) default
NULL,PRIMARY KEY (id)) AUTO_INCREMENT=1;

INSERT INTO products (Meal_time_name,Name,Price,ImageUrl) VALUES
("Morning","Morning Fresh","12","c-1.png"),("Breakfast","Tooplate Soup","20","c-
2.png"),("Night","Premium Steak","30","c-3.png"),("Morning","Seafood
Set","25","c-5.png"),("Dinner","Burger Set","90","c-6.png"),("Breakfast","Burger
Set","20","c-7.png");
```

- **I have mentioned the below lines in “roles/db/files/my.cnf”**

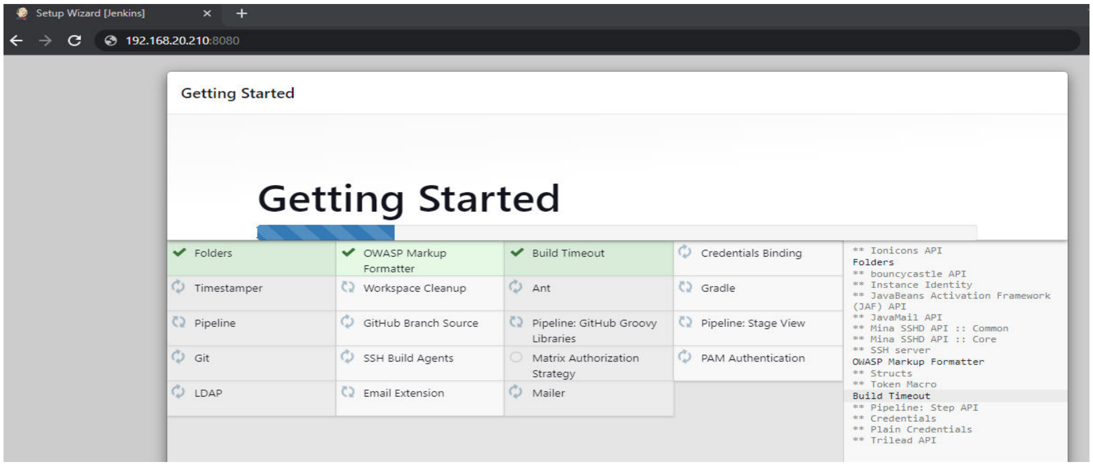
```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
symbolic-links=0
[mysqld_safe]
log-error=/var/log/mariadb/mariadb.log
pid-file=/var/run/mariadb/mariadb.pid
!includedir /etc/my.cnf.d
```



5. **Pipeline Workflow**:

5.1 Configure Jenkins Pipeline Job:

- Create a Jenkins pipeline job to define the CI/CD workflow.
- Specify the pipeline script using the Pipeline syntax.
- Set up stages and steps to execute the necessary tasks.

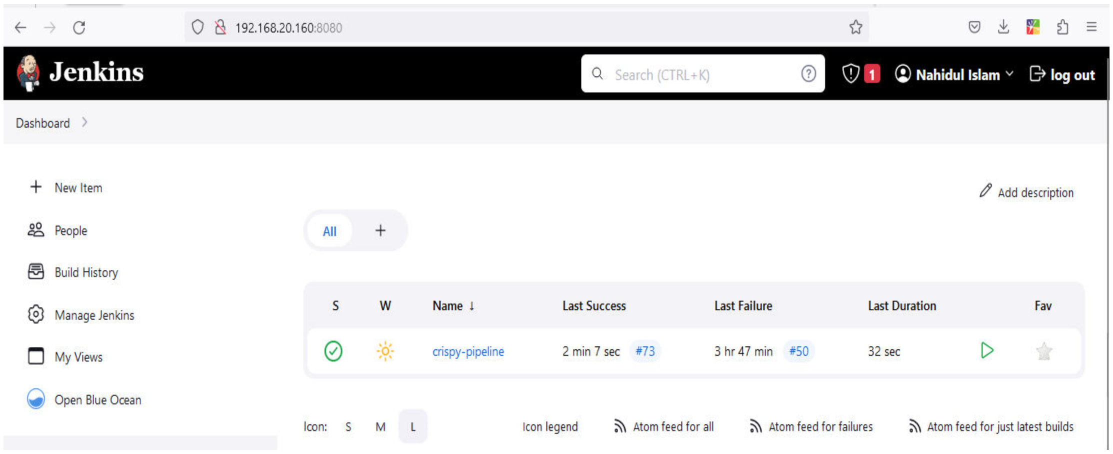


Getting Started

Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	Credentials Binding
Timestamper	Workspace Cleanup	Ant	Gradle
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline: Stage View
Git	SSH Build Agents	Matrix Authorization Strategy	PAM Authentication
LDAP	Email Extension	Mailer	

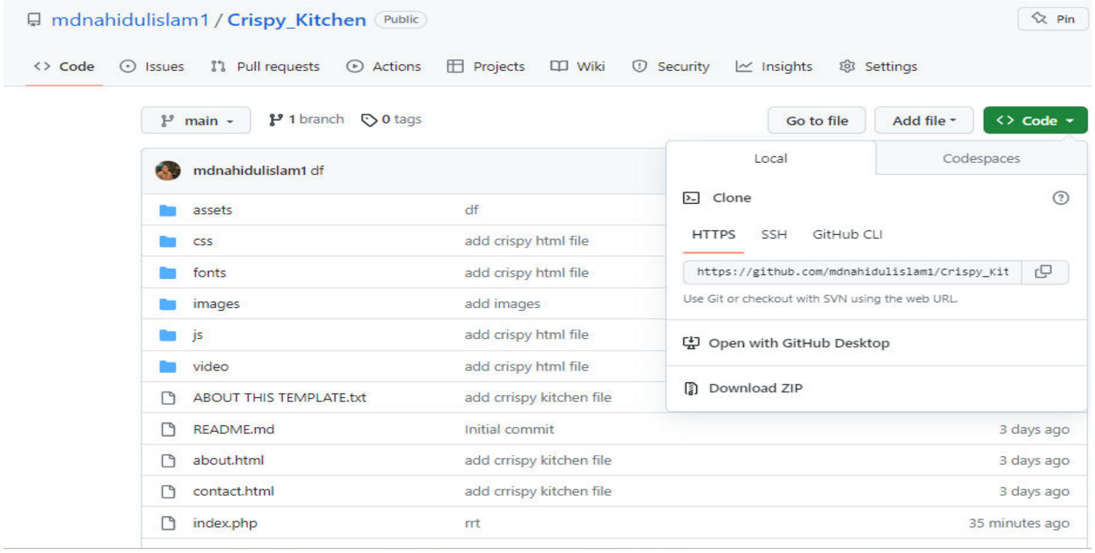
```
** Ionicons API
Folders
** bouncycastle API
** Instance Identity
** JavaBeans Activation Framework
(JAF) API
** JavaMail API
** Mina SSHD API :: Common
** Mina SSHD API :: Core
** SSH server
OWASP Markup Formatter
** Structs
** Token Macro
Build Timeout
** Pipeline: Step API
** Credentials
** Plain Credentials
** Trilead API
```



S	W	Name ↓	Last Success	Last Failure	Last Duration	Fav
✓	☀	crispy-pipeline	2 min 7 sec #73	3 hr 47 min #50	32 sec	☆

5.2 Integration with Git:

- Configure the Jenkins pipeline job to fetch the latest code changes from the GitHub repository.
- Utilize **Poll SCM** to automate the pipeline execution whenever there are new commits.



https://github.com/mdnahidulislam1/Crispy_Kit

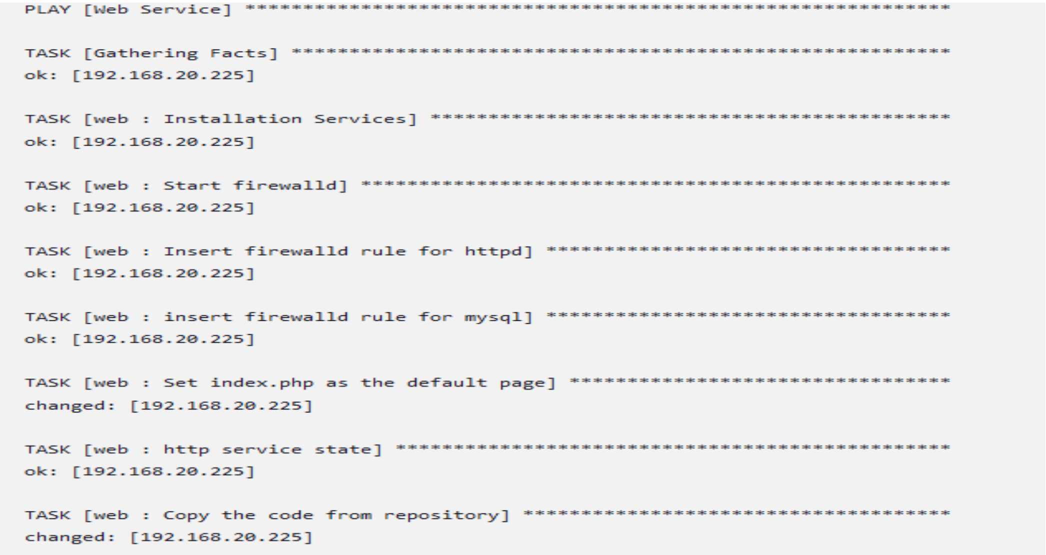
Use Git or checkout with SVN using the web URL.

mdnahidulislam1 df

assets	df	
css	add crispy html file	
fonts	add crispy html file	
images	add images	
js	add crispy html file	
video	add crispy html file	
ABOUT THIS TEMPLATE.txt	add crrispy kitchen file	
README.md	Initial commit	3 days ago
about.html	add crrispy kitchen file	3 days ago
contact.html	add crrispy kitchen file	3 days ago
index.php	rrt	35 minutes ago

5.3 Deployment using Ansible:

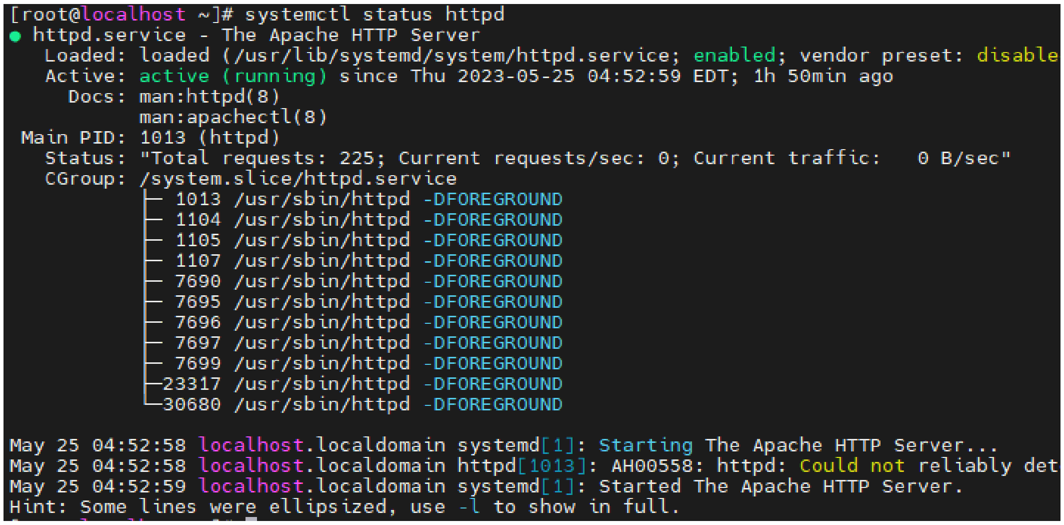
- Utilize the Ansible plugin in Jenkins to trigger Ansible playbooks for deployment.
- Write Ansible playbooks to configure the web server and deploy the application code from the GitHub repository.
- Ensure the necessary Ansible playbooks are accessible from the Jenkins pipeline.



```
PLAY [Web Service] *************************************************************

TASK [Gathering Facts] *********************************************************
ok: [192.168.20.225]

TASK [web : Installation Services] *********************************************
ok: [192.168.20.225]

TASK [web : Start firewalld] ***************************************************
ok: [192.168.20.225]

TASK [web : Insert firewalld rule for httpd] ***********************************
ok: [192.168.20.225]

TASK [web : insert firewalld rule for mysql] ***********************************
ok: [192.168.20.225]

TASK [web : Set index.php as the default page] *********************************
changed: [192.168.20.225]

TASK [web : http service state] ************************************************
ok: [192.168.20.225]

TASK [web : Copy the code from repository] *************************************
changed: [192.168.20.225]
```

5.4 Deployment to the Web Server:

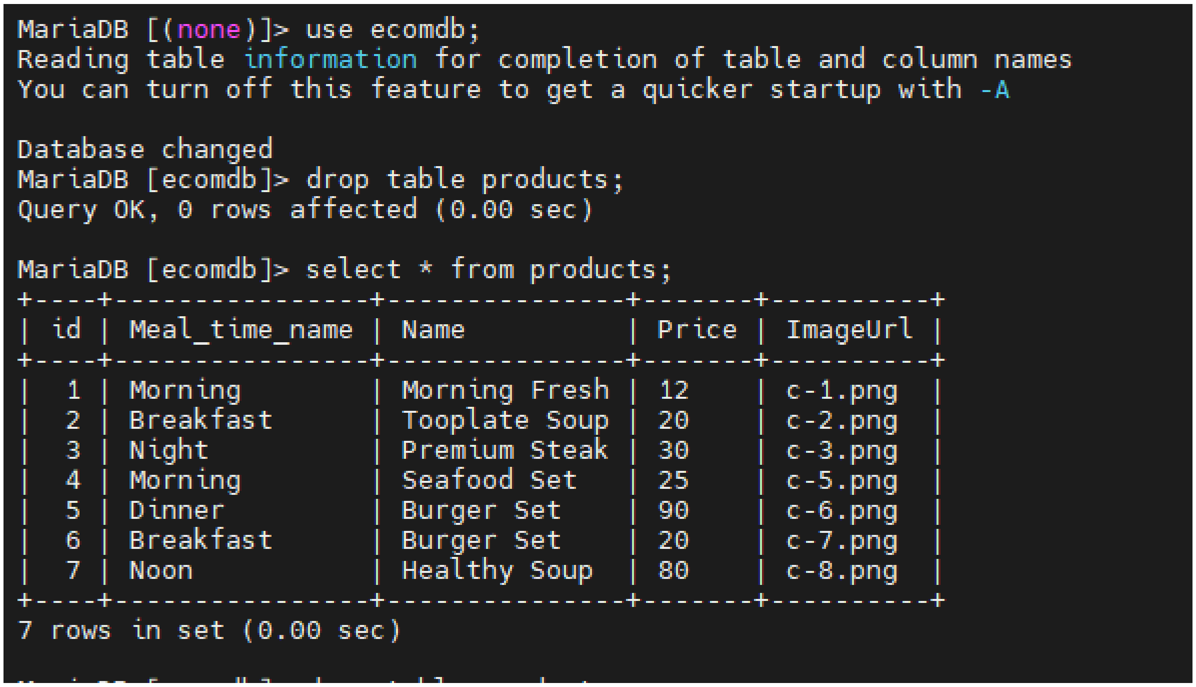
- Configure the pipeline to execute Ansible playbooks that deploy the application code to the web server.
- Ensure the Ansible inventory file specifies the target web server.



```
[root@localhost ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disable
   Active: active (running) since Thu 2023-05-25 04:52:59 EDT; 1h 50min ago
     Docs: man:httpd(8)
           man:apachectl(8)
 Main PID: 1013 (httpd)
   Status: "Total requests: 225; Current requests/sec: 0; Current traffic:   0 B/sec"
   CGroup: /system.slice/httpd.service
           ├─ 1013 /usr/sbin/httpd -DFOREGROUND
           ├─ 1104 /usr/sbin/httpd -DFOREGROUND
           ├─ 1105 /usr/sbin/httpd -DFOREGROUND
           ├─ 1107 /usr/sbin/httpd -DFOREGROUND
           ├─ 7690 /usr/sbin/httpd -DFOREGROUND
           ├─ 7695 /usr/sbin/httpd -DFOREGROUND
           ├─ 7696 /usr/sbin/httpd -DFOREGROUND
           ├─ 7697 /usr/sbin/httpd -DFOREGROUND
           ├─ 7699 /usr/sbin/httpd -DFOREGROUND
           ├─23317 /usr/sbin/httpd -DFOREGROUND
           └─30680 /usr/sbin/httpd -DFOREGROUND

May 25 04:52:58 localhost.localdomain systemd[1]: Starting The Apache HTTP Server...
May 25 04:52:58 localhost.localdomain httpd[1013]: AH00558: httpd: Could not reliably det
May 25 04:52:59 localhost.localdomain systemd[1]: Started The Apache HTTP Server.
Hint: Some lines were ellipsized, use -l to show in full.
```

5.5 Show DB server Database:

- Ansible install mysql in db server and load the sql file.

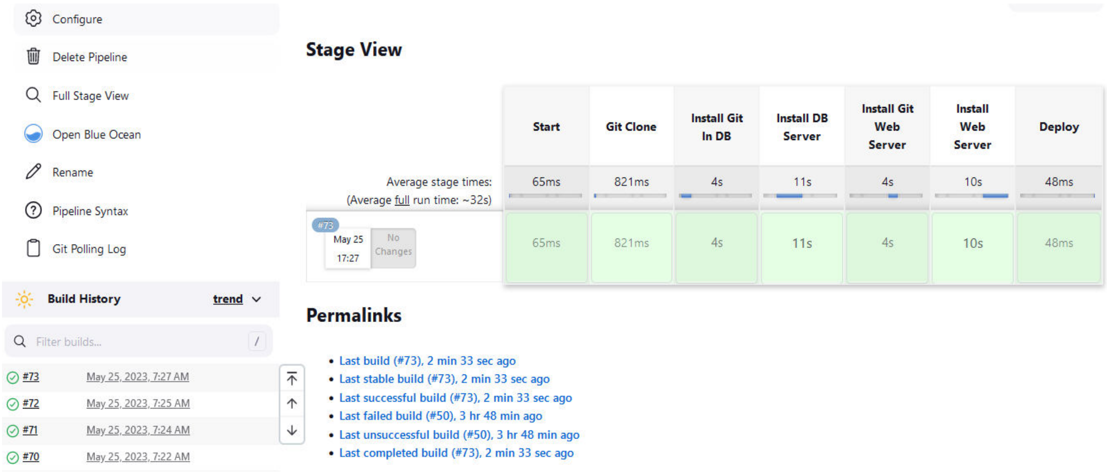


```
MariaDB [(none)]> use ecomdb;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [ecomdb]> drop table products;
Query OK, 0 rows affected (0.00 sec)

MariaDB [ecomdb]> select * from products;
+----+----------------+---------------+-------+----------+
| id | Meal_time_name | Name          | Price | ImageUrl |
+----+----------------+---------------+-------+----------+
|  1 | Morning        | Morning Fresh | 12    | c-1.png  |
|  2 | Breakfast      | Tooplate Soup | 20    | c-2.png  |
|  3 | Night          | Premium Steak | 30    | c-3.png  |
|  4 | Morning        | Seafood Set   | 25    | c-5.png  |
|  5 | Dinner         | Burger Set    | 90    | c-6.png  |
|  6 | Breakfast      | Burger Set    | 20    | c-7.png  |
|  7 | Noon           | Healthy Soup  | 80    | c-8.png  |
+----+----------------+---------------+-------+----------+
7 rows in set (0.00 sec)
```

5.6 Pipeline Execution and Monitoring:

- Start the Jenkins pipeline job to trigger the CI/CD pipeline.

- Monitor the pipeline execution in the Jenkins web interface, ensuring all stages complete successfully.
- Check the web server to verify the application has been deployed successfully.



Configure
Delete Pipeline
Full Stage View
Open Blue Ocean
Rename
Pipeline Syntax
Git Polling Log

Build History trend

Filter builds...

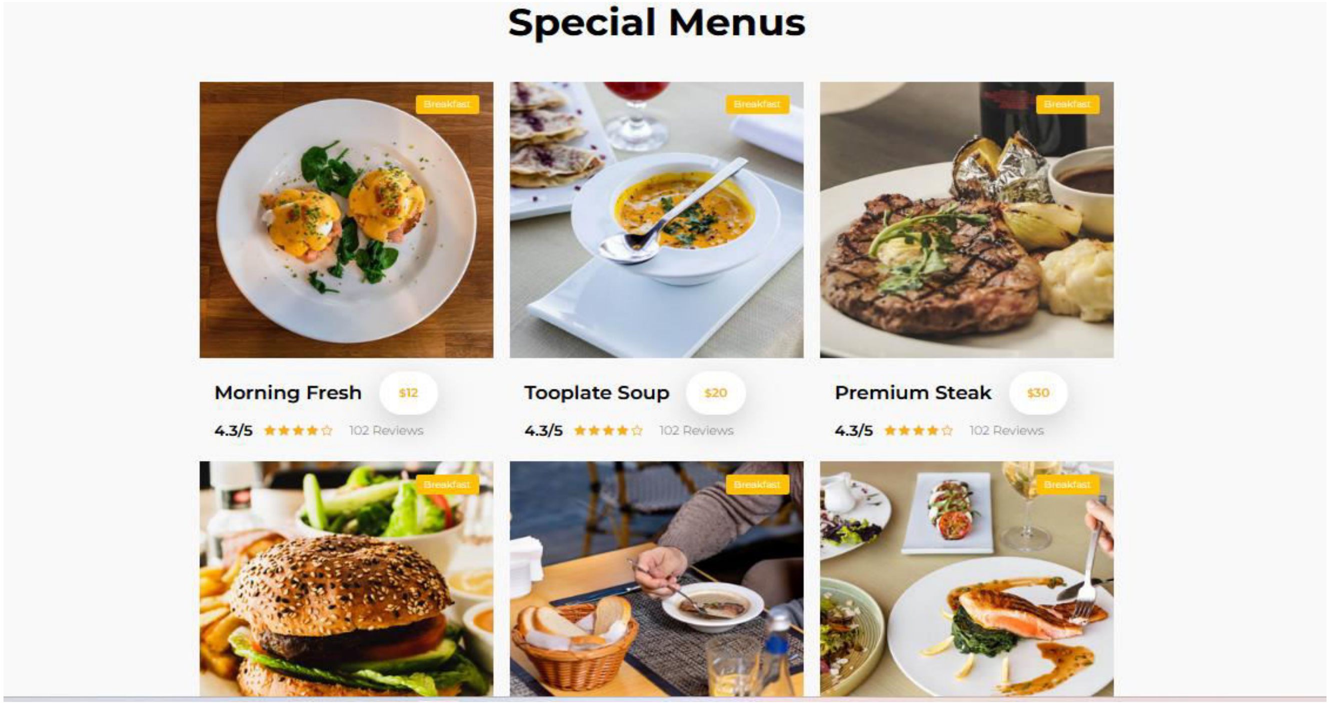
#73	May 25, 2023, 7:27 AM
#72	May 25, 2023, 7:25 AM
#71	May 25, 2023, 7:24 AM
#70	May 25, 2023, 7:22 AM

Stage View

	Start	Git Clone	Install Git In DB	Install DB Server	Install Git Web Server	Install Web Server	Deploy
Average stage times: (Average full run time: ~32s)	65ms	821ms	4s	11s	4s	10s	48ms
#73 May 25 17:27 No Changes	65ms	821ms	4s	11s	4s	10s	48ms

Permalinks

- Last build (#73), 2 min 33 sec ago
- Last stable build (#73), 2 min 33 sec ago
- Last successful build (#73), 2 min 33 sec ago
- Last failed build (#50), 3 hr 48 min ago
- Last unsuccessful build (#50), 3 hr 48 min ago
- Last completed build (#73), 2 min 33 sec ago



Special Menus

Morning Fresh $12
4.3/5 ★★★★☆ 102 Reviews

Tooplate Soup $20
4.3/5 ★★★★☆ 102 Reviews

Premium Steak $30
4.3/5 ★★★★☆ 102 Reviews

5.6 Conclusion: This document has provided a comprehensive guide to setting up a CI/CD pipeline using Ansible, Jenkins, and separate virtual machines for the master node, web server, and DB server. Following the steps outlined will allow me to automate the deployment of your application from GitHub to the web server through the Ansible and Jenkins integration.