

# Project-1

**Project:** Complete DevOps Lifecycle Implementation using CI/CD

**Company:** Abode Software

**Role:** Sr. DevOps Engineer

**Description:** Designed and implemented an end-to-end CI/CD pipeline with automated build, test, and production deployment using Jenkins, Docker, GitHub, and AWS.

## **Application Repo:**

<https://github.com/hshar/website.git>

---

## **Problem Statement:**

Abode Software, a product-based organization, required a complete transformation of their software delivery process by implementing an end-to-end DevOps lifecycle.

The objective was to automate the entire workflow from source code management to production deployment in order to improve release speed, ensure application reliability, and eliminate manual deployment errors.

The company needed a system that would:

- Enforce a proper Git workflow for development and production environments
- Automatically build and test code upon every commit
- Containerize the application using Docker for consistent deployments
- Implement a robust CI/CD pipeline using Jenkins
- Enable branch-based deployment logic (develop for testing, master for production)
- Achieve a fully automated production release process with zero manual intervention

The challenge was to design and implement a production-grade DevOps solution that integrates these requirements into a single automated pipeline while ensuring system stability, security, and scalability.

This project delivers a Complete DevOps Lifecycle Implementation that meets all these requirements and operates successfully in a live production environment.

---

## FINAL ARCHITECTURE:



---

## Infrastructure Details:

Server Role	Public IP	Private IP
EC2-1 Jenkins / CI Server	52.207.232.20	10.0.4.224
EC2-2 Production Server	54.198.106.173	10.0.5.125

---

## Tools & Technologies

AWS EC2, GitHub, Jenkins, Docker, Ansible, Linux, Shell Scripting

---

## 1. Base System Setup (Both Servers)

```
sudo apt update -y  
sudo apt upgrade -y  
sudo apt install -y git curl unzip
```

```
ubuntu@ip-10-0-4-224:~$ sudo apt update -y
sudo apt upgrade -y
sudo apt install -y git curl unzip
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1684 kB]
Get:14 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1391 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [311 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [175 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [15.8 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1506 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [306 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [377 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [31.4 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [2413 kB]
Get:23 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [225 kB]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [550 kB]
```

### i-074372a5e59debedd (ec2-1)

```
ubuntu@ip-10-0-5-125:~$ sudo apt update -y
sudo apt upgrade -y
sudo apt install -y git curl unzip
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1684 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [311 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [175 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [15.8 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1506 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [306 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [377 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [31.4 kB]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [2413 kB]
Get:22 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1391 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [550 kB]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
```

### i-00a4c23c15518a0e5 (ec2-2)

## 2. Jenkins Installation (EC2-1)

```
sudo apt install -y openjdk-17-jre
```

```
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee
```

```
/usr/share/keyrings/jenkins-keyring.asc > /dev/null
```

```
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] https://pkg.jenkins.io/debian-stable
binary/ | sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null
```

```
sudo apt update -y
```

```
sudo apt install -y jenkins
```

```
sudo systemctl start jenkins
```

```
sudo systemctl enable jenkins
```

## Access Jenkins:

```
http://52.207.232.20:8080
```

## Unlock Jenkins:

```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

```
ubuntu@ip-10-0-4-224:~$ sudo apt install -y openjdk-17-jre
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee /usr/share/keyrings/jenkins-keyring.asc > /dev/null
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] https://pkg.jenkins.io/debian-stable binary/ | sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
adwaita-icon-theme alsa-topology-conf alsu-ucm-conf at-spi2-common at-spi2-core ca-certificates-java dconf-gsettings-backend dconf-service fontconfig
fontconfig-config fonts-dejavu-core fonts-dejavu-extra fonts-dejavu-mono gsettings-desktop-schemas gtk-update-icon-cache hicolor-icon-theme humanity-icon-theme
java-common libasound2-data libasound2t64 libatk-wrapper-java libatk-wrapper-java-jni libatk1.0-0t64 libatspi2.0-0t64 libavahi-client3
libavahi-common-data libavahi-common3 libcairo-gobject2 libcairo2 libcups2t64 libdatriel libdconf1 libdeflate0 libdrm-amdgpu libdrm-intel libfontconfig
libgail-common libgail18t64 libgbm1 libgdk-pixbuf2.0-0 libgdk-pixbuf2.0-bin libglif7 libgl1 libgl1-mesa-dri libglvnd0 libglx-mesa0
libgraphite2-3 libgtk2.0-0t64 libgtk2.0-bin libgtk2.0-common libharfbuzz0b libice6 libjbig0 libjpeg-turbo8 liblcms2-2 liblerc4 liblilmv20 libpango-1.0-0
libpangocairo-1.0-0 libpangotft-1.0-0 libpccsclient libpixman-1-0 librsvg2-2 librsvg2-common libsharpuyu0 libsm6 libthai-data libthai0 libtiff6
libvulkan1 libwayland-client0 libwayland-server0 libwebp7 libx11-xcb1 libxaw7 libxcb-dri3-0 libxcb-glx0 libxcb-present0 libxcb-render0 libxcb-render0 libxcb-shape0
libxcb-shm0 libxcb-sync1 libxcb-xfixes0 libcomposite1 libcursor0 libxdamage1 libxfixes3 libxft2 libxi6 libxinerama1 libxkbfile1 libxmu6 libxpm4 libxrandr2
libxrender1 libxshmfence1 libxt6t64 libxtst6 libxvi libxxf86dg1 libxxf86vm1 mesa-lib�allium mesa-vulkan-drivers openjdk-17-jre-headless session-migration
ubuntu-mono x11-common x11-utils
Suggested packages:
default-jre alsu-utils libasound2-plugins cups-common gvfs liblcms2-utils pcscd librsvg2-bin libnss-mdns fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei
| fonts-wqy-zenhei fonts-indic mesa-utils
Recommended packages:
luit
The following NEW packages will be installed:
adwaita-icon-theme alsa-topology-conf alsu-ucm-conf at-spi2-common at-spi2-core ca-certificates-java dconf-gsettings-backend dconf-service fontconfig
fontconfig-config fonts-dejavu-core fonts-dejavu-extra fonts-dejavu-mono gsettings-desktop-schemas gtk-update-icon-cache hicolor-icon-theme humanity-icon-theme
java-common libasound2-data libasound2t64 libatk-wrapper-java libatk-wrapper-java-jni libatk1.0-0t64 libatspi2.0-0t64 libavahi-client3
ubuntu@ip-10-0-4-224:~$ i-074372a5e59debedd (ec2-1)
```

```
ubuntu@ip-10-0-4-224:~$ sudo apt update -y
sudo apt install -y jenkins
sudo systemctl start jenkins
sudo systemctl enable jenkins
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Ign:5 https://pkg.jenkins.io/debian-stable binary/ InRelease
Get:6 https://pkg.jenkins.io/debian-stable binary/ Release [2044 B]
Get:7 https://pkg.jenkins.io/debian-stable binary/ Release.gpg [833 B]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [21.5 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [71.5 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [208 B]
Get:11 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Get:12 https://pkg.jenkins.io/debian-stable binary/ Packages [30.3 kB]
Fetched 253 kB in 1s (432 kB/s)
Reading package lists... 4%
```

i-074372a5e59debedd (ec2-1)

```
ubuntu@ip-10-0-4-224:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
81b63f1c822a4e208710a1feb937ad24
ubuntu@ip-10-0-4-224:~$
```

## Getting Started

# Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

`/var/lib/jenkins/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

**Administrator password**

.....|

[Continue](#)

### 3. Docker Installation (Both Servers)

`curl -fsSL https://get.docker.com | sh`

`sudo usermod -aG docker jenkins`

`sudo usermod -aG docker ubuntu`

`sudo systemctl restart jenkins`

`newgrp docker`

```
ubuntu@ip-10-0-4-224:~$ curl -fsSL https://get.docker.com | sh
sudo usermod -aG docker jenkins
sudo usermod -aG docker ubuntu
sudo systemctl restart jenkins
newgrp docker
# Executing docker install script, commit: 8b33a64d28ec86a1121623f1d349801b48f2837b
+ sudo -E sh -c apt-get -qq update >/dev/null
```

**i-074372a5e59debedd (ec2-1)**

DNS-IP-10-0-4-224-202-272-20 - Private-IP-10-0-4-224

```
ubuntu@ip-10-0-5-125:~$ curl -fsSL https://get.docker.com | sh
sudo usermod -aG docker jenkins
sudo usermod -aG docker ubuntu
# Executing docker install script, commit: 8b33a64d28ec86a1121623f1d349801b48f2837b
+ sudo -E sh -c apt-get -qq update >/dev/null
+ sudo -E sh -c DEBIAN_FRONTEND=noninteractive apt-get -y -qq install ca-certificates curl >/dev/null
+ sudo -E sh -c install -m 0755 -d /etc/apt/keyrings
+ sudo -E sh -c curl -fsSL "https://download.docker.com/linux/ubuntu/gpg" -o /etc/apt/keyrings/docker.asc
+ sudo -E sh -c chmod a+r /etc/apt/keyrings/docker.asc
+ sudo -E sh -c echo "deb [arch=amd64 signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu noble stable" > /etc/apt/sources.list.d/docker.list
+ sudo -E sh -c apt-get -qq update >/dev/null
```

i-00a4c23c15518a0e5 (ec2-2)

X

## 4. Secure SSH Trust Setup (EC2-1 → EC2-2)

### Step-1: Generate SSH Key on Jenkins Server (EC2-1)

[ssh-keygen](#)

### Step-2: Connect to Production Server via AWS Console

AWS Console → EC2 → Instances →

Select EC2-2 (54.237.228.253) → Connect → EC2 Instance Connect

You will be logged in as:

ubuntu@ip-10-0-11-139

### Step-3: Prepare SSH Directory on EC2-2

[mkdir -p ~/.ssh](#)

[chmod 700 ~/.ssh](#)

[nano ~/.ssh/authorized\\_keys](#)

### Step-4: Install Jenkins Public Key

On EC2-1:

[cat ~/.ssh/id\\_ed25519.pub](#)

Copy the entire key.

On EC2-2, paste it as a new line in:

[~/.ssh/authorized\\_keys](#)

Save and exit, then run:

[chmod 600 ~/.ssh/authorized\\_keys](#)

### Step-5: Verify Passwordless SSH

Back on EC2-1:

[ssh ubuntu@54.237.228.253](#)

Successful login without key prompt confirms secure trust is established.

```
ubuntu@ip-10-0-5-125:~$ mkdir -p ~/.ssh  
chmod 700 ~/.ssh  
nano ~/.ssh/authorized_keys
```

### i-00a4c23c15518a0e5 (ec2-2)

```
ubuntu@ip-10-0-4-224:~$ cat ~/.ssh/id_ed25519.pub  
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAICrW+1JDXbzBeb5EZ3YiESTiROaqWzFKmkLnIkYiccpx ubuntu@ip-10-0-4-224  
ubuntu@ip-10-0-4-224:~$
```

### i-074372a5e59debedd (ec2-1)

```
GNU nano 7.2                                         /home/ubuntu/.ssh/authorized_keys *  
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQcQq4Or16PY2yLAZMxCkg8e4Fe4ARCw6zmpSCAhHE/2qYsrtt4AyEC3gylnJVsSmrIpc6CFmkLnIkYiccpx ubuntu@ip-10-0-4-224  
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAICrW+1JDXbzBeb5EZ3YiESTiROaqWzFKmkLnIkYiccpx ubuntu@ip-10-0-4-224
```

^G Help ^Q Write Out ^X Where Is ^X Cut ^X Execute ^X Location M-! Undo M-A Set Mark M-] To Bracket M-` Previous  
^X Exit ^R Read File ^R Replace ^P Paste ^J Justify ^L Go To Line M-B Redo M-C Copy ^Q Where Was M-W Next  
i-00a4c23c15518a0e5 (ec2-2) X

```
ubuntu@ip-10-0-5-125:~$ chmod 600 ~/.ssh/authorized_keys
ubuntu@ip-10-0-5-125:~$ █
```

## i-00a4c23c15518a0e5 (ec2-2)

```
ubuntu@ip-10-0-4-224:~$ ssh ubuntu@10.0.5.125
The authenticity of host '10.0.5.125 (10.0.5.125)' can't be established.
ED25519 key fingerprint is SHA256:xcf/zzyUdwcsXJ3apE7ZtmBUiJyptCOZM9bGahYaps.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.5.125' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1015-aws x86_64)

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

System information as of Tue Dec 30 12:33:58 UTC 2025

  System load:  0.02              Temperature:          -273.1 °C
  Usage of /:   39.7% of 6.71GB  Processes:            120
  Memory usage: 20%              Users logged in:      1
  Swap usage:   0%               IPv4 address for ens5: 10.0.5.125

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
```

## i-074372a5e59debedd (ec2-1)

## 5. Ansible Setup (EC2-1)

```
sudo apt install -y ansible
```

```
sudo mkdir -p /etc/ansible
```

```
sudo nano /etc/ansible/hosts
```

paste this:

[prod]

10.0.5.125 ansible user=ubuntu

## Test:

```
ansible prod -m ping
```

```
ubuntu@ip-10-0-4-224:~$ sudo apt install -y ansible
sudo mkdir -p /etc/ansible
sudo nano /etc/ansible/hosts
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ansible-core python3-argcomplete python3-dnspython python3-kerberos py
    python3-resolvelib python3-selinux python3-simplejson python3-winrm py
Suggested packages:
  cowsay sshpass python3-trio python3-aioquic python3-h2 python3-htpx p
The following NEW packages will be installed:
  ansible ansible-core python3-argcomplete python3-dnspython python3-kei
    python3-resolvelib python3-selinux python3-simplejson python3-winrm py
0 upgraded, 15 newly installed, 0 to remove and 0 not upgraded.
Need to get 19.5 MB of archives.
After this operation, 315 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 py
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/unive
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 py
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64
```

i-074372a5e59debedd (ec2-1)

GNU nano 7.2

[prod]

```
10.0.5.125 ansible user=ubuntu
```

```
ubuntu@ip-10-0-4-224:~$ ansible prod -m ping
10.0.5.125 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-10-0-4-224:~$ █
```

---

## 6. Git Workflow Implementation

Repository forked:

<https://github.com/Vikky9387/website.git>

git clone https://github.com/Vikky9387/website.git

cd website

### Create develop branch

git checkout -b develop

git push -u origin develop

```
ubuntu@ip-10-0-4-224:~$ git clone https://github.com/Vikky9387/website.git
cd website
Cloning into 'website'...
remote: Enumerating objects: 46, done.
remote: Counting objects: 100% (24/24), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 46 (delta 11), reused 13 (delta 6), pack-reused 22 (from 2)
Receiving objects: 100% (46/46), 89.73 KiB | 14.96 MiB/s, done.
Resolving deltas: 100% (15/15), done.
ubuntu@ip-10-0-4-224:~/website$ █
```

```
ubuntu@ip-10-0-4-224:~/website$ git checkout -b develop
git push -u origin develop
Switched to a new branch 'develop'
Username for 'https://github.com': vikky9387
Password for 'https://vikky9387@github.com':
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:     https://github.com/Vikky9387/website/pull/new/develop
remote:
To https://github.com/Vikky9387/website.git
 * [new branch]      develop -> develop
branch 'develop' set up to track 'origin/develop'.
ubuntu@ip-10-0-4-224:~/website$ █
```

---

## 7. Dockerfile Creation

[nano Dockerfile](#)

paste this:

```
FROM hshar/webapp
COPY . /var/www/html
```

Run:

```
git add .
git commit -m "Added Dockerfile"
git push origin develop
```

```
ubuntu@ip-10-0-4-224:~/website$ nano Dockerfile
```

```
GNU nano 7.2
FROM hshar/webapp
COPY . /var/www/html
```

```
ubuntu@ip-10-0-4-224:~/website$ git add .
git commit -m "Added Dockerfile"
git push origin develop
[develop c72a89b] Added Dockerfile
Committer: Ubuntu <ubuntu@ip-10-0-4-224.ec2.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 2 insertions(+)
create mode 100644 Dockerfile
Username for 'https://github.com': vikky9387
Password for 'https://vikky9387@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 360 bytes | 360.00 KiB/s, done.
```

**i-074372a5e59debedd (ec2-1)**

---

## 8. Jenkins Pipeline

[nano Jenkinsfile](#)

paste this:

```
pipeline {
    agent any

    stages {
        stage('Build') {
            steps {
                sh 'docker build -t abodeapp .'
            }
        }
    }
}
```

```
stage('Test') {
```

```
    steps {
```

```
        sh ""
```

```

    docker rm -f testapp || true
    docker run -d --name testapp -p 8081:80 abodeapp
    sleep 5
    curl http://localhost:8081
    ""
}

}

stage('Prod') {
    when {
        expression {
            env.GIT_BRANCH == 'origin/master' || env.BRANCH_NAME == 'master'
        }
    }
    steps {
        sh """
        docker save abodeapp:latest > abodeapp.tar
        scp abodeapp.tar ubuntu@10.0.5.125:/home/ubuntu/
        ssh ubuntu@10.0.5.125 "docker load < /home/ubuntu/abodeapp.tar"
        ssh ubuntu@10.0.5.125 "docker rm -f prodapp || true"
        ssh ubuntu@10.0.5.125 "docker run -d --name prodapp -p 80:80 abodeapp"
        ""
    }
}

}

```

Now run these commands:

```

git add Jenkinsfile
git commit -m "Fix Jenkinsfile syntax and correct prod stage condition"
git push origin develop

```

```
ubuntu@ip-10-0-4-224:~/website$ nano Jenkinsfile
```

```
GNU nano 7.2
pipeline {
    agent any
    stages {
        stage('Build') {
            steps { sh 'docker build -t abodeapp .' }
        }

        stage('Test') {
            steps {
                sh '''
                    docker rm -f testapp || true
                    docker run -d --name testapp -p 8081:80 abodeapp
                    sleep 5
                    curl http://localhost:8081
                '''
            }
        }

        stage('Prod') {
            when { branch 'master' }
            steps {
                sh '''

```

```
^G Help          ^O Write Out      ^W Where Is      ^K Cut           ^T Execute
^X Exit          ^R Read File       ^\ Replace       ^U Paste         ^J Justify
```

**i-074372a5e59debedd (ec2-1)**

PublicIPs: 52.207.232.20 PrivateIPs: 10.0.4.224

## 9. Jenkins Job Setup

Plugins installed:

- Git
- Pipeline
- Docker Pipeline

Pipeline job: **Abode-DevOps-Pipeline**

**SCM:**

<https://github.com/Vikky9387/website.git>

## Branch specifier:

\*/develop

\*/master

## Build Trigger:

GitHub hook trigger for GITScm polling

Manage Jenkins / Plugins

The screenshot shows the Jenkins 'Plugins' page. At the top, there's a search bar with 'Search available plugins'. Below it, a table lists two plugins: 'Docker Pipeline' and 'Pipeline Graph Analysis'. The 'Docker Pipeline' row has a blue checkmark icon next to it, indicating it is installed. The 'Name' column shows 'Docker Pipeline' and 'Pipeline Graph Analysis'. The 'Released' column shows '2 mo 17 days ago' for Docker Pipeline and '3 mo 21 days ago' for Pipeline Graph Analysis. The 'Health' column shows a green circle with the number '76' for Docker Pipeline and a green circle with the number '97' for Pipeline Graph Analysis. A note at the bottom of the Docker Pipeline row says: 'This plugin is up for adoption! We are looking for new maintainers. Visit our [Adopt a Plugin](#) initiative for more information.'

## Download progress

### Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

Authentication Tokens API

Docker Commons

Docker Pipeline

Loading plugin extensions

## New Item

Enter an item name

Abode-DevOps-Pipeline

Select an item type



Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.



Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.



Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.



Folder

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

OK

## Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

- Build after other projects are built [?](#)
- Build periodically [?](#)
- GitHub hook trigger for GITScm polling [?](#)
- Poll SCM [?](#)
- Trigger builds remotely (e.g., from scripts) [?](#)

## Definition

Pipeline script from SCM

SCM [?](#)

Git

Repositories [?](#)

Repository URL [?](#)

`https://github.com/Vikky9387/website.git`

## Branches to build [?](#)

Branch Specifier (blank for 'any') [?](#)

`*/develop`

Branch Specifier (blank for 'any') [?](#)

`*/master`

## 10. GitHub Webhook

<http://52.207.232.20:8080/github-webhook/>

## Webhooks / Add webhook

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in [our developer documentation](#).

### Payload URL \*

http://52.207.232.20:8080/github-webhook/

### Content type \*

application/json

### Secret

### SSL verification

By default, we verify SSL certificates when delivering payloads.

Enable SSL verification    Disable (not recommended)

### Which events would you like to trigger this webhook?

Just the push event.

---

## 11. CI/CD Execution Flow

### Branch Pipeline

develop Build → Test

master Build → Test → Prod

## 12. Final Verification & Validation

### Test Develop Branch (CI Only)

```
echo "Test change" >> index.html
```

```
git add .
```

```
git commit -m "Test develop"
```

```
git push origin develop
```

### Result:

- ✓ Jenkins Pipeline triggered
- ✓ **Build + Test stages executed**
- ✓ No production deployment (as expected)

```
ubuntu@ip-10-0-4-224:~/website$ echo "Test change" >> index.html
git add .
git commit -m "Test develop"
git push origin develop
[master 8b3e762] Test develop
Committer: Ubuntu <ubuntu@ip-10-0-4-224.ec2.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:
```

```
git config --global --edit
```

After doing this, you may fix the identity used for this commit with:

```
git commit --amend --reset-author
```

```
2 files changed, 15 insertions(+), 11 deletions(-)
Username for 'https://github.com': vikky9387
Password for 'https://vikky9387@github.com':
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Vikky9387/website.git
 8185a54..4ef170b  develop -> develop
ubuntu@ip-10-0-4-224:~/website$ █
```

## i-074372a5e59debedd (ec2-1)

 Jenkins / Abode-DevOps-Pipeline

Status      **Abode-DevOps-Pipeline**

</> Changes      Permalinks

▷ Build Now

⌚ Configure

Delete Pipeline

⠇ Stages

✍ Rename

ⓘ Pipeline Syntax

📋 GitHub Hook Log

Builds

Pending

🕒 #1      ✖

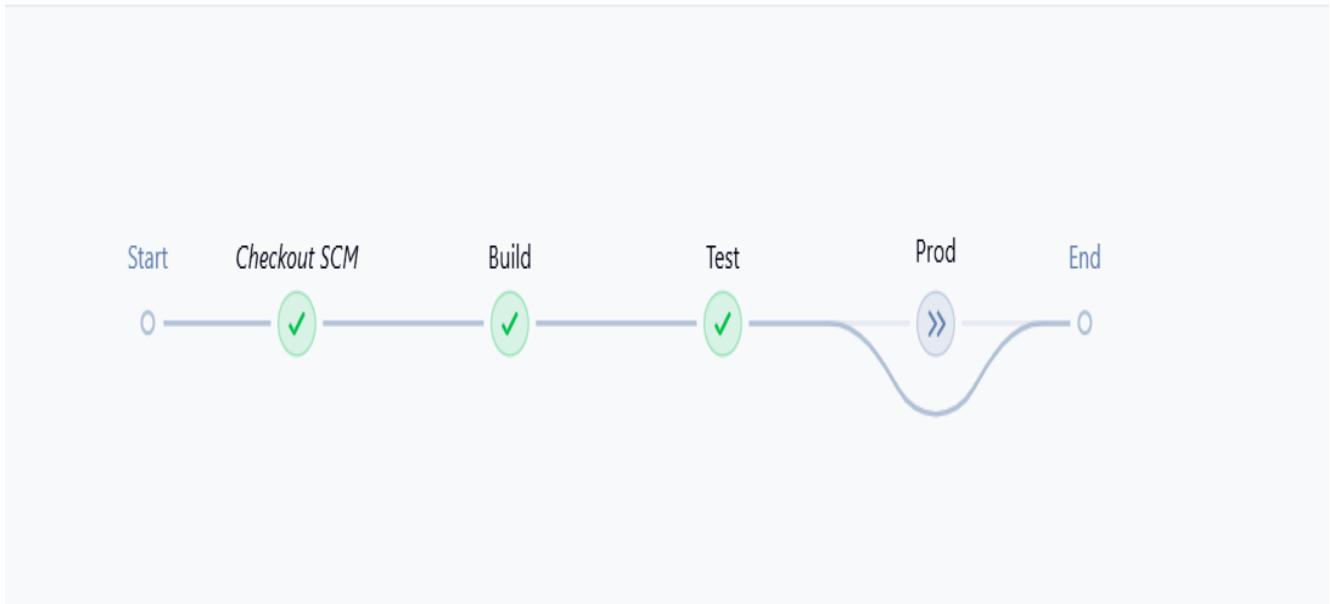
In the quiet period. Expires in 2.4 sec

Abode-DevOps-Pipeline / #1 / Console Output

SYNOPSIS

```
CI/CD Test
Final Prod Trigger
Auto trigger working now
Final clean run
Production release
LIVE PRODUCTION
CI/CD Test
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Prod)
Stage "Prod" skipped due to when conditional
[Pipeline] getContext
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Took 24 sec



## Deploy to Production

`git checkout master`

`git merge develop`

`git push origin master`

## Result:

✓ Jenkins Pipeline triggered

✓ **Build → Test → Prod stages executed**

✓ Docker image deployed automatically on Production Server

```
ubuntu@ip-10-0-4-224:~/website$ git checkout master
git merge develop
git push origin master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
Updating 33914b2..4ef170b
Fast-forward
 Dockerfile |  2 ++
 Jenkinsfile | 34 ++++++=====
 index.html |  2 ++
 3 files changed, 38 insertions(+)
 create mode 100644 Dockerfile
 create mode 100644 Jenkinsfile
Username for 'https://github.com': vikky9387
Password for 'https://vikky9387@github.com':
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 307 bytes | 307.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/Vikky9387/website.git
 33914b2..4ef170b master -> master
ubuntu@ip-10-0-4-224:~/website$
```

## i-074372a5e59debedd (ec2-1)

Status

Changes

Build Now

Configure

Delete Pipeline

Stages

Rename

Pipeline Syntax

GitHub Hook Log

Abode-DevOps-Pipeline

Permalinks

- Last build (#1), 1 min 29 sec ago
- Last stable build (#1), 1 min 29 sec ago
- Last successful build (#1), 1 min 29 sec ago
- Last completed build (#1), 1 min 29 sec ago

Builds

Filter

Pending

#2

In the quiet period. Expires in 2:1 sec

```

CI/CD Test
Final Prod Trigger
Auto trigger working now
Final clean run
Production release
LIVE PRODUCTION
CI/CD Test
LIVE PRODUCTION
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Prod)
Stage "Prod" skipped due to when conditional
[Pipeline] getContext
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```



Search  Filter

Prod 29s Started 49s ago Jenkins ...

<span>✓</span> Checkout SCM 0.27s	<span>✓</span> docker save abodeapp:latest > abodeapp.tar scp abodeapp.tar ubuntu@10.0.5.125:/home/ubuntu/ ssh ubuntu@10.0.5.125 "docker ... v
<span>✓</span> Build 18s	0 + docker save abodeapp:latest
<span>✓</span> Test 6.9s	1 + scp abodeapp.tar ubuntu@10.0.5.125:/home/ubuntu/
<span>✓</span> Prod 29s	2 + ssh ubuntu@10.0.5.125 docker load < /home/ubuntu/abodeapp.tar
	3 Loaded image: abodeapp:latest
	4 + ssh ubuntu@10.0.5.125 docker rm -f prodapp    true

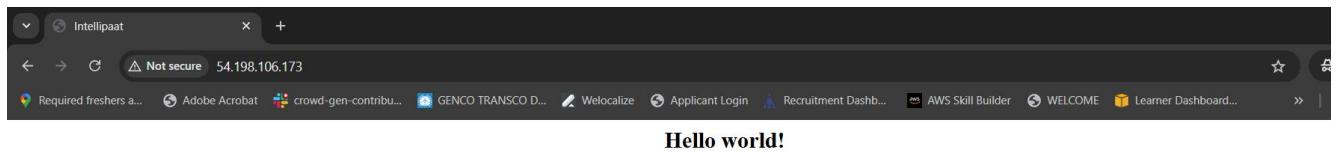
2.207.232.20:8080/job/Abode-DevOps-Pipeline/6/pipeline-overview/?selected-node=18

---

## 13. Final Live Check

Open in browser:

<http://54.198.106.173>



# GitHub

---

---

## 14. Production Image Distribution Fix:

If the image does NOT exist, reload it:

On EC2-1 (Jenkins server):

```
docker save abodeapp:latest > abodeapp.tar
```

```
scp abodeapp.tar ubuntu@3.81.164.105:/home/ubuntu/
```

On EC2-2:

```
docker load < abodeapp.tar
```

```
docker run -d --name prodapp -p 80:80 abodeapp
```

Verify:

```
docker ps
```

```
curl http://localhost
```

```
ubuntu@ip-10-0-12-71:~/website$ docker save abodeapp:latest > abodeapp.tar
scp abodeapp.tar ubuntu@3.81.164.105:/home/ubuntu/
abodeapp.tar
abodeapp.tar
abodeapp.tar
```

```
ubuntu@ip-10-0-12-71:~/website$
```

i-0fa7c9278c7605841 (ec2-1)

```
ubuntu@ip-10-0-6-109:~$ docker load < abodeapp.tar
docker run -d --name prodapp -p 80:80 abodeapp
Loaded image: abodeapp:latest
83a7c4c85161f023fa5252fa6a6e6acf4ed85a479da688b199527437e69a1c55
ubuntu@ip-10-0-6-109:~$ █
```

### i-0a4303ba6d475e8b0 (ec2-2)

```
Docker ID: 7-01-164-105 IP Address: 10.0.6.100
```

```
ubuntu@ip-10-0-6-109:~$ docker ps
curl http://localhost
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
83a7c4c85161 abodeapp "/bin/sh -c 'apachec..." 37 seconds ago Up 36 seconds 0.0.0.0:80->80/tcp, [::]:80->80/tcp prodapp
<html>
<head>
<title> Intellipaat </title>
</head>
<body style = "background-image:url('images/github3.jpg'); background-size: 100%">
<h2 ALIGN= CENTER>Hello world!</h2>
</body>
</html>
CI/CD Test
Final Prod Trigger
Auto trigger working now
Final clean run
Production release
LIVE PRODUCTION
CI/CD Test
LIVE PRODUCTION
Test change
Another test
Another test
Production deploy trigger
ubuntu@ip-10-0-6-109:~$ █
```

### i-0a4303ba6d475e8b0 (ec2-2)

---

## 15. Final Result & Conclusion

A complete production-grade DevOps lifecycle was implemented with:

- Automated CI/CD
- Branch-based deployments
- Dockerized application
- Zero-manual production release
- Live production system

**Project Implemented By:**

Vikram Gangadhara.