Host a HTML website using Apache server, push it on Github and integrate it through Jenkins CI/CD Pipeline

# Install Apache/Httpd server on Linux (using Amazon linux 2)

1. sudo su
2. sudo yum update
3. sudo yum install httpd -y
4. sudo systemctl start httpd
5. sudo systemctl status httpd

# Create a index.html file at /var/www/html/

1. cd /var/www/html
2. nano index.html
3. Paste the following html script in index file:

<!DOCTYPE html>

<html>

<head>

<title>My Website</title>

</head>

<body>

<h1>Welcome to my first website</h1>

</body>

</html>

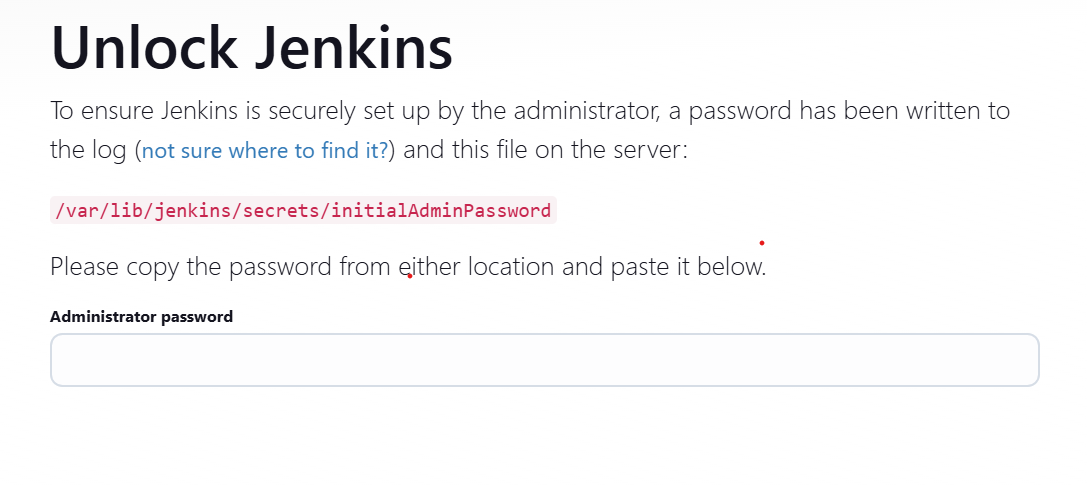
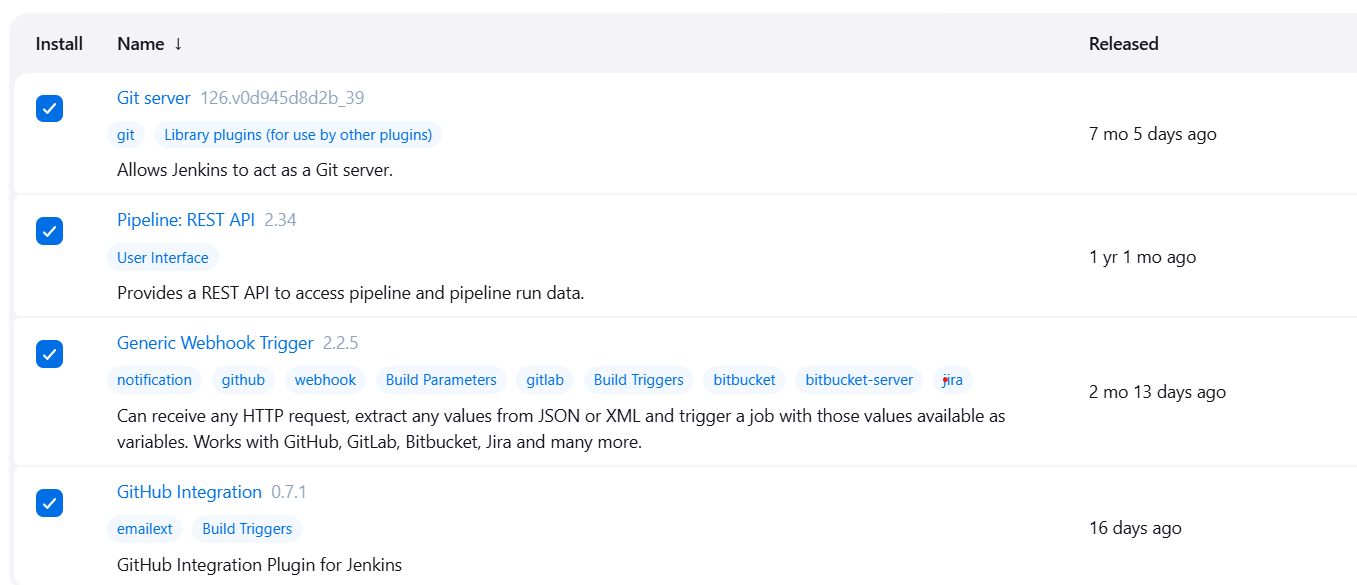
# Install Git and Push index file

1. yum install git -y
2. git config --global user.name "your name"
3. git config --global user.email "your email@gmail.com"
4. git init
5. git add .
6. git commit -m "initial commit" .
7. git remote add origin <your repository>
8. git push origin master

# Install Jenkins

1. sudo wget -O /etc/yum.repos.d/jenkins.repo <https://pkg.jenkins.io/redhat-stable/jenkins.repo>
2. sudo rpm --import <https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key>
3. sudo amazon-linux-extras enable java-openjdk11
4. sudo yum install java-17-amazon-corretto-devel -y
5. sudo yum upgrade -y
6. sudo yum install jenkins -y
7. export JAVA\_HOME=/usr/lib/jvm/java-17-amazon-corretto.x86\_64
8. source ~/.bash\_profile
9. java -version
10. sudo systemctl start jenkins
11. sudo systemctl status Jenkins

# Setup Jenkins

1. Enter this in browser:  
   http://<your-ec2-instance-public-ip>:8080
2. cat <given location>  
   and enter the result as administrator password  
   
3. Choose **Install suggested plugins** on the next window.
4. Enter required details and move to next window
5. Go to Manage Jenkins > Plugins and Install the following Plugins:  
   
6. In your Github Repo, add webhook:  
   Settings > Webhook  
   Payload URL: **http://<your-ec2-instance-public-ip>:8080/github-webhook/**Content Type: **application/json**and click on **Add Webhook**Build Pipeline
7. In Jenkins Dashboard, Create a new Pipeline job
8. Choose **GitHub hook trigger for GITScm polling** option in **Build Triggers.**
9. Add the following in Pipeline Script:

pipeline {

agent any

environment {

GIT\_REPO = ‘<your github repo url>' // Replace with your repo URL

BRANCH = 'master'

HTTPD\_DIR = '/var/www/html' // Default HTTPD web root

BACKUP\_DIR = '/var/www/backup'

DEPLOY\_USER = 'jenkins'

}

stages {

stage('Cleanup Workspace') {

steps {

cleanWs()

}

}

stage('Clone Repository') {

steps {

git branch: "${BRANCH}",

url: "${GIT\_REPO}"

}

}

stage('Backup Current Deployment') {

steps {

sh '''

TIMESTAMP=$(date +%Y%m%d\_%H%M%S)

sudo mkdir -p ${BACKUP\_DIR}

if [ -d "${HTTPD\_DIR}/\*" ]; then

sudo cp -r ${HTTPD\_DIR} ${BACKUP\_DIR}/backup\_${TIMESTAMP}

fi

'''

}

}

stage('Deploy to Httpd') {

steps {

sh '''

# Ensure httpd directory exists

sudo mkdir -p ${HTTPD\_DIR}

# Remove old files (optional)

sudo rm -rf ${HTTPD\_DIR}/\*

# Copy new files

sudo cp -r ./\* ${HTTPD\_DIR}/

# Set proper permissions

sudo chown -R apache:apache ${HTTPD\_DIR}

sudo chmod -R 755 ${HTTPD\_DIR}

# Reload Apache to apply changes

sudo systemctl reload httpd

'''

}

}

stage('Verify Deployment') {

steps {

sh '''

# Check if httpd is running

if sudo systemctl is-active --quiet httpd; then

echo "Httpd is running"

else

echo "httpd is not running, starting it..."

sudo systemctl start httpd

fi

# Test httpd configuration

sudo httpd -t

# Reload httpd to apply changes

sudo systemctl reload httpd

'''

}

}

}

post {

success {

echo 'Deployment successful!'

}

failure {

sh '''

# Restore from latest backup if deployment fails

if [ -d "${BACKUP\_DIR}" ]; then

LATEST\_BACKUP=$(ls -t ${BACKUP\_DIR} | head -n1)

if [ ! -z "${LATEST\_BACKUP}" ]; then

sudo cp -r ${BACKUP\_DIR}/${LATEST\_BACKUP}/\* ${HTTPD\_DIR}/

fi

fi

'''

echo 'Deployment failed! Restored from backup.'

}

}

}

1. Apply and Save
2. In ec2 instance,   
   sudo visudo

39. Add following at the end of file:  
jenkins ALL=(ALL) NOPASSWD: /usr/bin/mkdir -p /var/www/html  
jenkins ALL=(ALL) NOPASSWD: /usr/bin/rm -rf /var/www/html/\*  
jenkins ALL=(ALL) NOPASSWD: /usr/bin/cp -r \* /var/www/html/  
jenkins ALL=(ALL) NOPASSWD: /bin/chown -R apache apache /var/www/html  
jenkins ALL=(ALL) NOPASSWD: /bin/chmod -R 755 /var/www/html  
jenkins ALL=(ALL) NOPASSWD: /usr/sbin/nginx  
jenkins ALL=(ALL) NOPASSWD: /bin/systemctl start httpd  
jenkins ALL=(ALL) NOPASSWD: /bin/systemctl stop httpd  
jenkins ALL=(ALL) NOPASSWD: /bin/systemctl reload httpd  
jenkins ALL=(ALL) NOPASSWD: /bin/systemctl status httpd  
jenkins ALL=(ALL) NOPASSWD: /bin/mkdir  
jenkins ALL=(ALL) NOPASSWD: /bin/cp -r  
jenkins ALL=(ALL) NOPASSWD: /bin/rm -rf  
jenkins ALL=(ALL) NOPASSWD: /bin/chown -R  
jenkins ALL=(ALL) NOPASSWD: /bin/chmod -R  
jenkins ALL=(ALL) NOPASSWD: ALL

40. Click on build now in jenkins.