DevOps

Assignment 3

Submitted by: Irum Imtiaz

Reg no: FA22-BCT-011

Table of Contents

Introduction

This report documents the implementation of a **CI/CD pipeline using Jenkins** on **AWS EC2** for a **Flask web application**. The pipeline automates building, testing, and deployment using Docker. The web application stores messages in a **PostgreSQL database**, and Selenium tests automate functional verification.

Objectives

- Install and configure Jenkins on an AWS EC2 instance.
- Integrate GitHub repository with Jenkins using webhooks.
- Write **2 Selenium test cases** for functional testing.
- Build Docker images for web app and Selenium tests.
- Implement a **Jenkins Declarative Pipeline** with the following stages:
    1. Code Linting
    2. Docker Build
    3. Unit Testing
    4. Containerized Deployment
    5. Selenium Testing

Configure and install Jenkins over the cloud.

Install google chrome

```
[7761:7761:1128/164715.252778:ERROR:ui/aura/env.cc:257] The platform failed to initialize.  Exiting.
ubuntu@ip-172-31-21-183:~/selenium_lab3$ sudo apt install ./google-chrome-stable_current_amd64.deb
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'google-chrome-stable' instead of './google-chrome-stable_current_amd64.deb'
google-chrome-stable is already the newest version (142.0.7444.175-1).
0 upgraded, 0 newly installed, 0 to remove and 16 not upgraded.
ubuntu@ip-172-31-21-183:~/selenium_lab3$ sudo apt-get install -f
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 16 not upgraded.
ubuntu@ip-172-31-21-183:~/selenium_lab3$ google-chrome -version
[7855:7855:1128/164752.220281:ERROR:ui/ozone/platform/x11/ozone_platform_x11.cc:249] Missing X server or $DISPLAY
[7855:7855:1128/164752.220312:ERROR:ui/aura/env.cc:257] The platform failed to initialize.  Exiting.
ubuntu@ip-172-31-21-183:~/selenium_lab3$ google-chrome --headless --version
Google Chrome 142.0.7444.175
ubuntu@ip-172-31-21-183:~/selenium_lab3$ google-chrome -version
```

Installed Google Chrome Driver to handle selenium

```
ubuntu@ip-172-31-21-183:~/selenium_lab3$ sudo unzip chromedriver-linux64.zip
Archive:  chromedriver-linux64.zip
  inflating: chromedriver-linux64/LICENSE.chromedriver
  inflating: chromedriver-linux64/THIRD_PARTY_NOTICES.chromedriver
  inflating: chromedriver-linux64/chromedriver
ubuntu@ip-172-31-21-183:~/selenium_lab3$ sudo mv chromedriver-linux64 /usr/bin/chromedriver
ubuntu@ip-172-31-21-183:~/selenium_lab3$ sudo chown root:root /usr/bin/chromedriver
ubuntu@ip-172-31-21-183:~/selenium_lab3$ sudo chmod +x /usr/bin/chromedriver
ubuntu@ip-172-31-21-183:~/selenium_lab3$
```

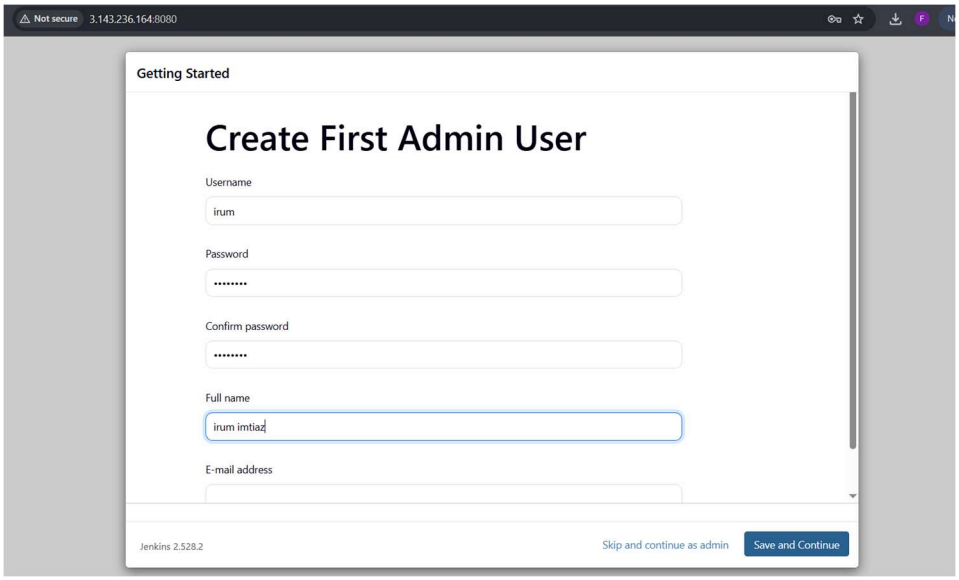
Installed docker,git,python and jenkins

```
oot@ip-172-31-21-183:/home/ubuntu# sudo apt update
it:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble InRelease
it:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
it:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
it:4 http://security.ubuntu.com/ubuntu noble-security InRelease
eading package lists... Done
uilding dependency tree... Done
eading state information... Done
6 packages can be upgraded. Run 'apt list --upgradable' to see them.
oot@ip-172-31-21-183:/home/ubuntu# sudo wget -O /etc/apt/keyrings/jenkins-keyring.asc \
 https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
-2025-11-28 08:22:35--  https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
esolving pkg.jenkins.io (pkg.jenkins.io)... 146.75.78.133, 2a04:4e42:83::645
onnecting to pkg.jenkins.io (pkg.jenkins.io)|146.75.78.133|:443... connected.
TTP request sent, awaiting response... 200 OK
ength: 3175 (3.1K) [application/pgp-keys]
aving to: '/etc/apt/keyrings/jenkins-keyring.asc'

etc/apt/keyrings/jenkins-keyring.asc     100%[=========================================================================>]   3.10K  --.-KB/s    i

025-11-28 08:22:35 (53.1 MB/s) - '/etc/apt/keyrings/jenkins-keyring.asc' saved [3175/3175]

oot@ip-172-31-21-183:/home/ubuntu# echo "deb [signed-by=/etc/apt/keyrings/jenkins-keyring.asc]" \
 https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
 /etc/apt/sources.list.d/jenkins.list > /dev/null
```

```
containers need to be restarted.

user sessions are running outdated binaries.

VM guests are running outdated hypervisor (qemu) binaries on this host.
t@ip-172-31-21-183:/home/ubuntu# cat /var/lib/jenkins/secrets/initialAdminPassword
a05c597b940cea2cac40985d987b7
t@ip-172-31-21-183:/home/ubuntu# ^C
t@ip-172-31-21-183:/home/ubuntu#
```

Installing selenium webdriver

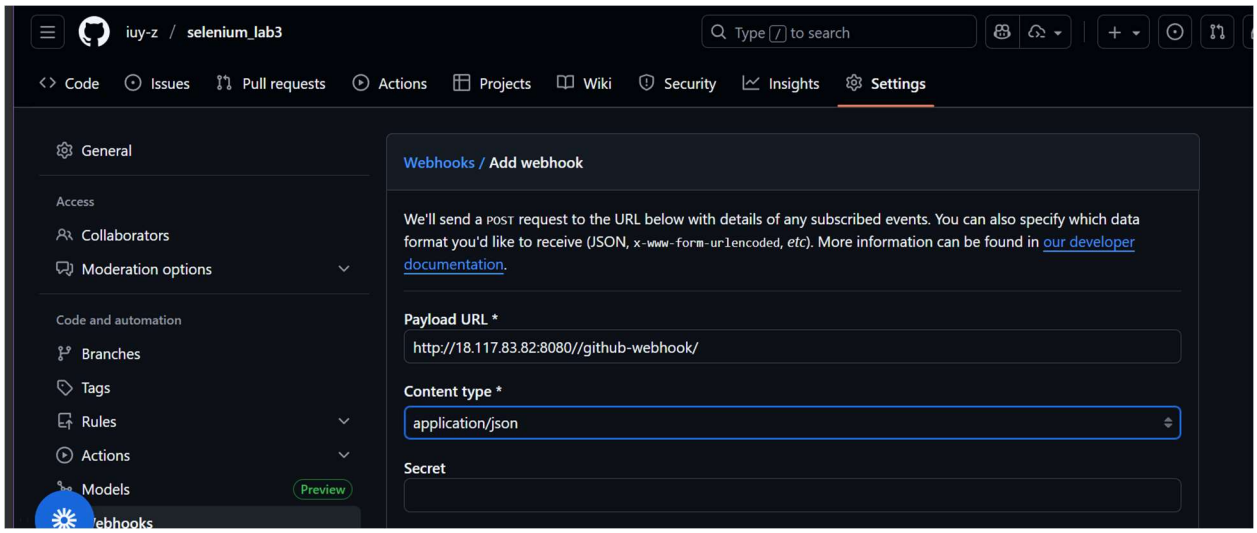
```
ubuntu@ip-172-31-21-183:~/selenium_lab3$ sudo pip install selenium --break-system-packages
Collecting selenium
  Downloading selenium-4.38.0-py3-none-any.whl.metadata (7.5 kB)
Collecting urllib3<3.0,>=2.5.0 (from urllib3[socks]<3.0,>=2.5.0->selenium)
  Downloading urllib3-2.5.0-py3-none-any.whl.metadata (6.5 kB)
Collecting trio<1.0,>=0.31.0 (from selenium)
  Downloading trio-0.32.0-py3-none-any.whl.metadata (8.5 kB)
Collecting trio-websocket<1.0,>=0.12.2 (from selenium)
  Downloading trio_websocket-0.12.2-py3-none-any.whl.metadata (5.1 kB)
Collecting certifi>=2025.10.5 (from selenium)
  Downloading certifi-2025.11.12-py3-none-any.whl.metadata (2.5 kB)
Collecting typing_extensions<5.0,>=4.15.0 (from selenium)
  Downloading typing_extensions-4.15.0-py3-none-any.whl.metadata (3.3 kB)
Collecting websocket-client<2.0,>=1.8.0 (from selenium)
  Downloading websocket_client-1.9.0-py3-none-any.whl.metadata (8.3 kB)
```

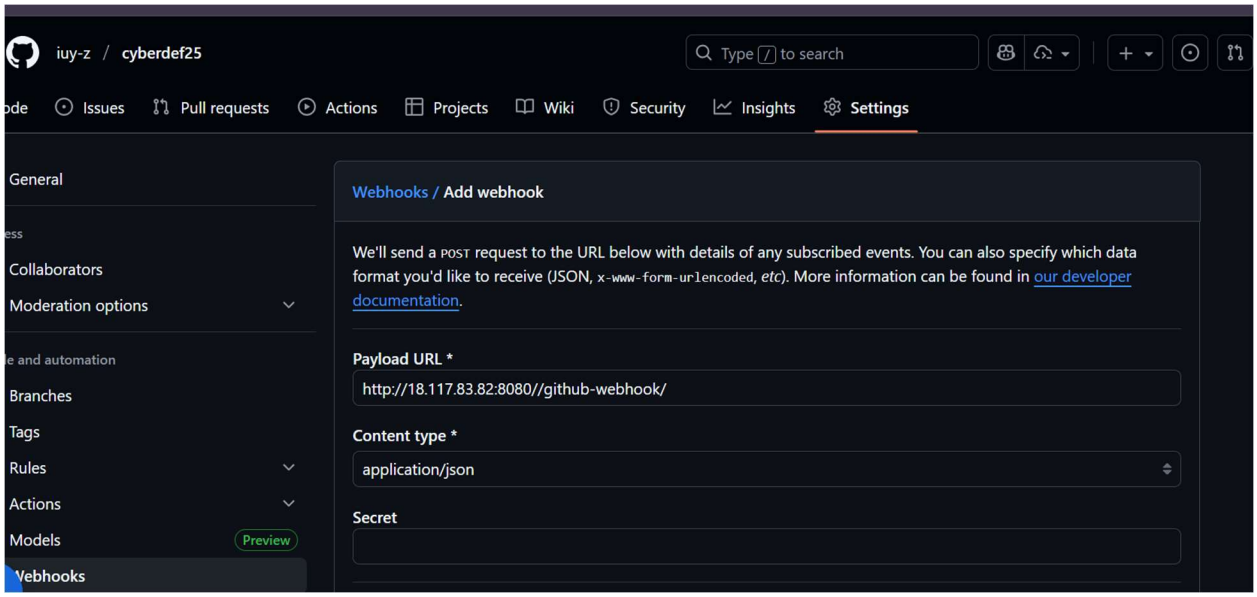


Give permission to jenkins to run docker

```
t@ip-172-31-21-183:/home/ubuntu# ^C
t@ip-172-31-21-183:/home/ubuntu# sudo usermod -aG docker jenkins
t@ip-172-31-21-183:/home/ubuntu# sudo systemctl restart jenkins
t@ip-172-31-21-183:/home/ubuntu#
```

Integrate Git with the Jenkins using GitHub Webhook approach





Created app.py

```
  GNU nano 7.2                                                app.py *
from flask import Flask, request, jsonify

app = Flask(__name__)

# Dummy database (dictionary)
db = {}

@app.route('/')
def home():
    return "Welcome to the Web App!"

@app.route('/add', methods=['POST'])
def add_item():
    data = request.json
    key = data.get("key")

^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location    M-U Undo       M-A S
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To Line  M-E Redo       M-6 C
```

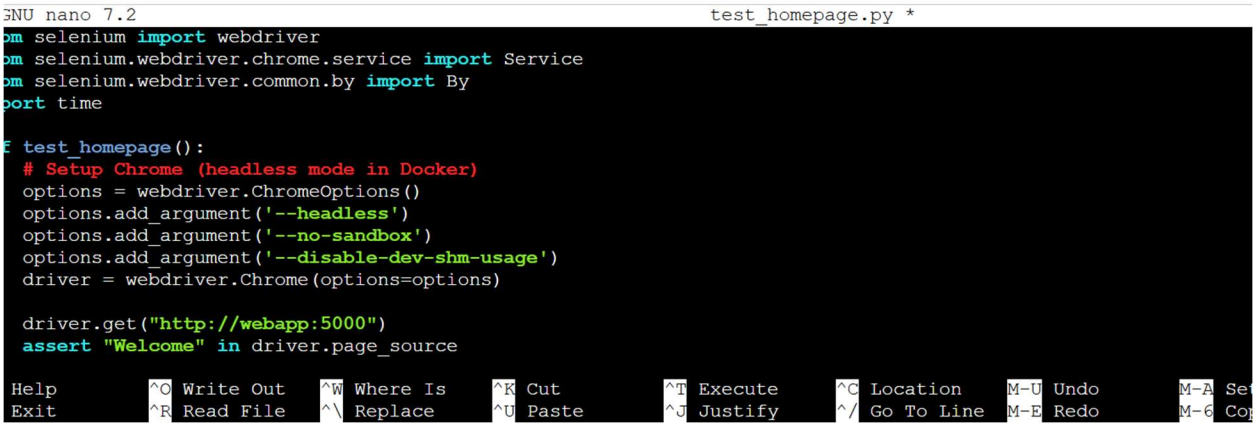
Created req.txt

```
  GNU nano 7.2                                          requirements.txt *
Flask==2.3.2
```

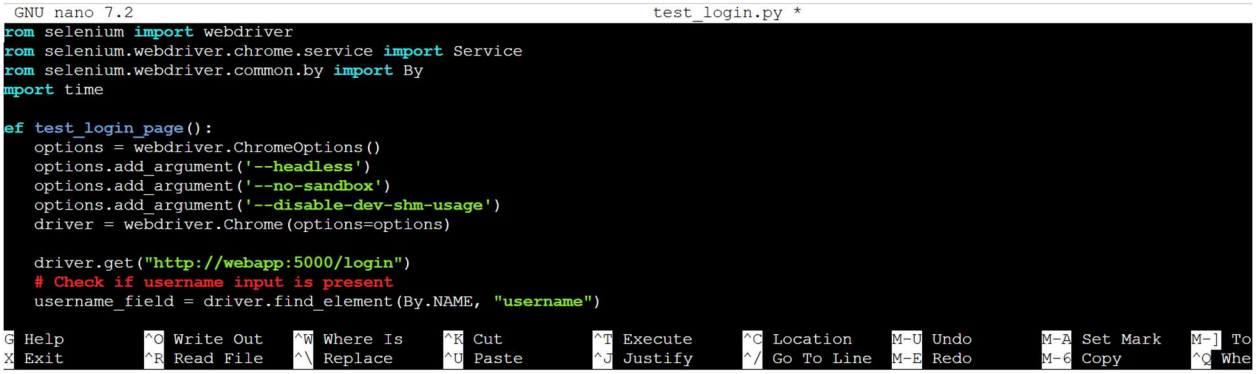
Write automated test cases using Selenium (at least two test cases).

Created a dir named selenium tests

```
ot@ip-172-31-21-183:/home/selenium_lab3/selenium_tests# cd ..
ot@ip-172-31-21-183:/home/selenium_lab3# ls -la
tal 16
wxr-xr-x 4 root root 4096 Nov 28 17:04 .
wxr-xr-x 4 root root 4096 Nov 28 17:01 ..
wxr-xr-x 7 root root 4096 Nov 28 17:01 .git
wxr-xr-x 2 root root 4096 Nov 28 17:05 selenium_tests
ot@ip-172-31-21-183:/home/selenium_lab3#
```




```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By
import time

def test_homepage():
    # Setup Chrome (headless mode in Docker)
    options = webdriver.ChromeOptions()
    options.add_argument('--headless')
    options.add_argument('--no-sandbox')
    options.add_argument('--disable-dev-shm-usage')
    driver = webdriver.Chrome(options=options)

    driver.get("http://webapp:5000")
    assert "Welcome" in driver.page_source

    driver.quit()
```




```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By
import time


def test_login_page():
    options = webdriver.ChromeOptions()
    options.add_argument('--headless')
    options.add_argument('--no-sandbox')
    options.add_argument('--disable-dev-shm-usage')
    driver = webdriver.Chrome(options=options)

    driver.get("http://webapp:5000/login")
    # Check if username input is present
    username_field = driver.find_element(By.NAME, "username")
    assert username_field is not None


    driver.quit()
```

Create Docker image of the Selenium test cases along with all dependencies.

Created dockerfile for web app

FROM python:3.10-slim

WORKDIR /app

COPY requirements.txt .

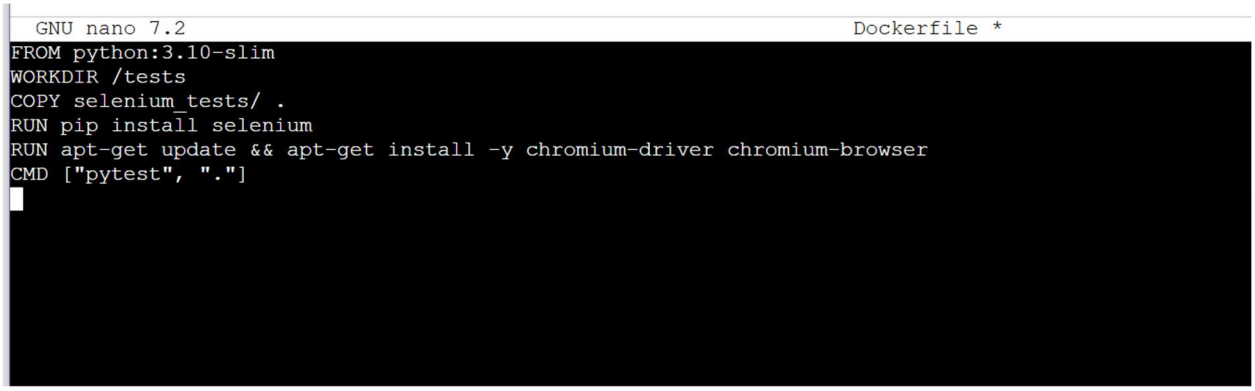
RUN pip install --no-cache-dir -r requirements.txt

COPY . .

EXPOSE 5000

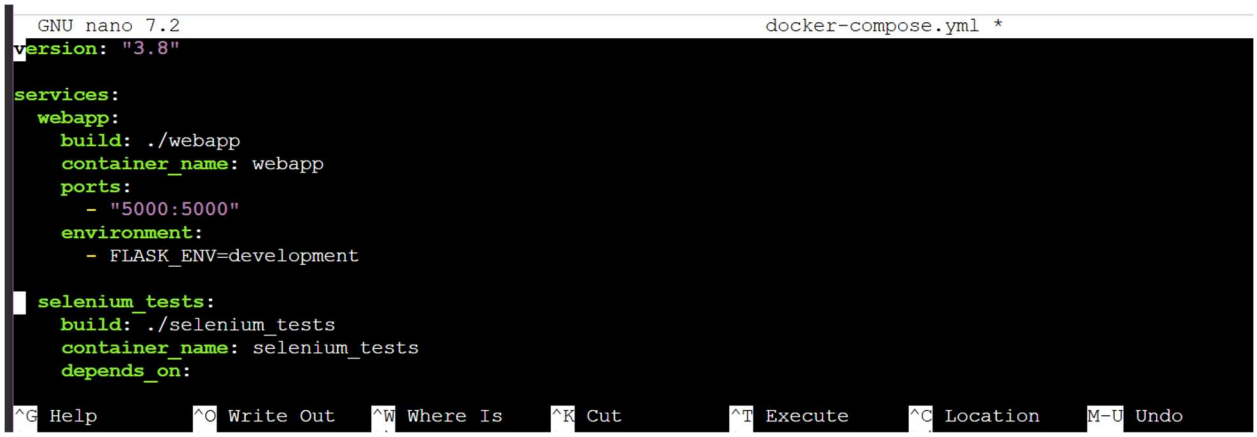
CMD ["python", "app.py"]

Created dockerfile for selenium tests



```
FROM python:3.10-slim

# Set working directory
WORKDIR /tests

# Copy test scripts
COPY selenium_tests/ .

# Install Selenium and Chromium for headless testing
RUN apt-get update && apt-get install -y \
   chromium \
   chromium-driver \
   && rm -rf /var/lib/apt/lists/*

# Install Python dependencies
RUN pip install --no-cache-dir selenium

# Set environment variable so Selenium uses Chromium
ENV CHROME_BIN=/usr/bin/chromium
ENV CHROMEDRIVER_PATH=/usr/bin/chromedriver

# Run default Selenium test
CMD ["python", "test_homepage.py"]
```

Created docker compose file



version: "3.8"

services:

webapp:

build: ./webapp

container_name: webapp

ports:

- "5000:5000"

environment:

- FLASK_ENV=development

selenium_tests:

build: ./selenium_tests

container_name: selenium_tests

depends_on:

- webapp

Use network mode so Selenium tests can reach the web app

network_mode: "service:webapp"

build n run

```
Successfully tagged selenium_lab3_selenium_tests:latest
root@ip-172-31-21-183:/home/selenium_lab3# docker-compose up
Creating network "selenium_lab3_default" with the default driver
Creating selenium_lab3_webapp_1 ... done
Creating selenium_lab3_selenium_tests_1 ... done
Attaching to selenium_lab3_webapp_1, selenium_lab3_selenium_tests_1
webapp_1           |  * Serving Flask app 'app'
webapp_1           |  * Debug mode: off
webapp_1           | WARNING: This is a development server. Do not use it in a production deployment. Use a pro
.
webapp_1           |  * Running on all addresses (0.0.0.0)
webapp_1           |  * Running on http://127.0.0.1:5000
webapp_1           |  * Running on http://172.20.0.2:5000
webapp_1           | Press CTRL+C to quit
selenium_lab3_selenium_tests_1 exited with code 0
webapp_1           | 182.184.138.171 - - [28/Nov/2025 17:47:14] "GET / HTTP/1.1" 200 -
webapp_1           | 182.184.138.171 - - [28/Nov/2025 17:47:15] "GET /favicon.ico HTTP/1.1" 404 -
```

Write a Jenkins pipeline script that includes the following stages:

- Code Linting stage
- Code Build stage
- Unit Testing stage
- Containarized Deployment stage
- Selenium Testing stage

```
 GNU nano 7.2                                                    Jenkinsfile *
pipeline {
    agent any
    environment {
        DOCKER_HUB_USER = 'irum90'
        DOCKER_HUB_PASS = credentials('docker-hub-password') // add in Jenkins
    }
    stages {
        stage('Code Linting') {
            steps {
                echo 'Linting code...'
                sh 'flake8 .'
            }
        }
        stage('Code Build') {
            steps {
Save modified buffer?
 Y Yes
```

pipeline {

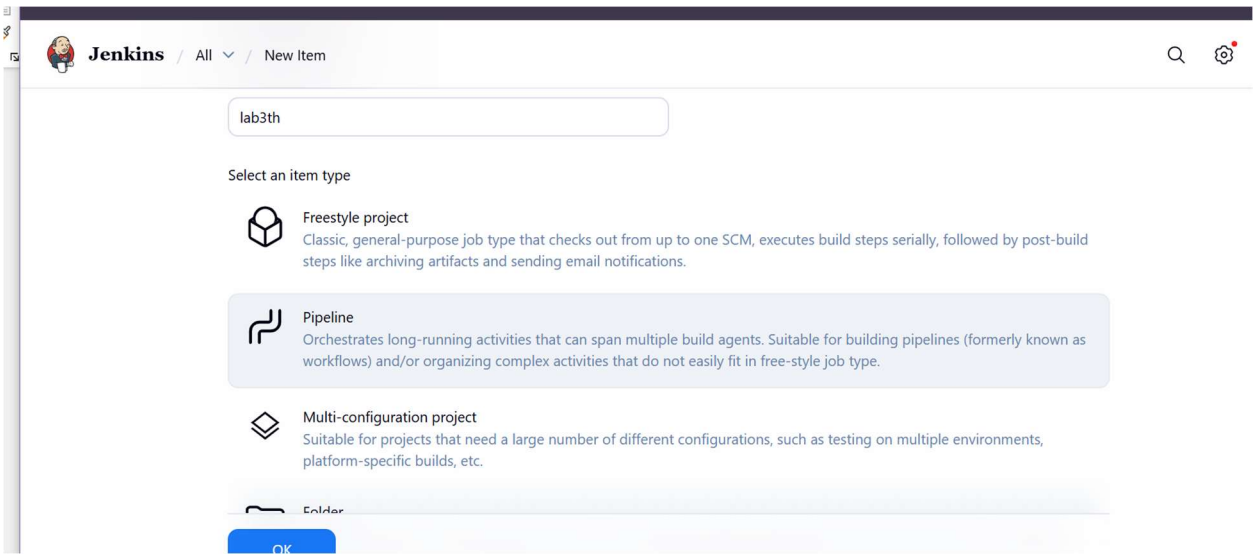
agent any

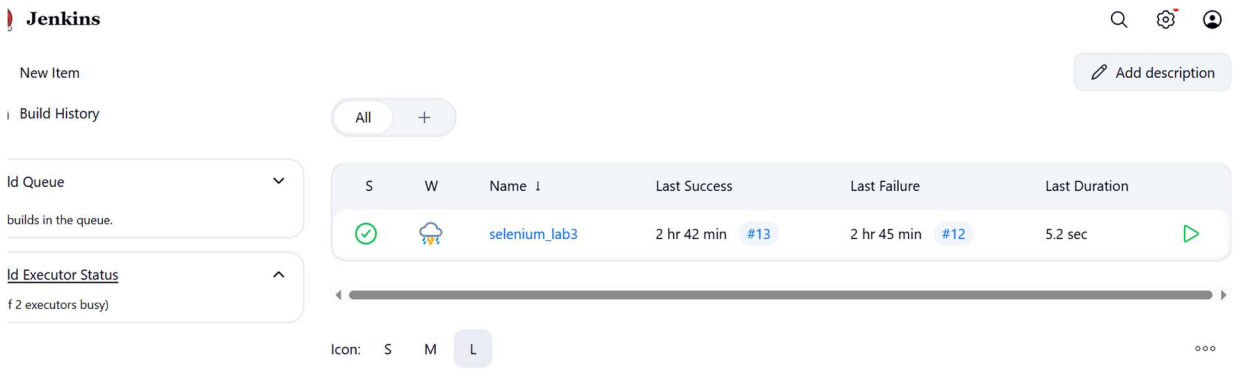
environment {

```
    DOCKER_HUB_USER = 'irum90'
    DOCKER_HUB_PASS = credentials('docker-hub-password') // add in Jenkins
  }
  stages {
    stage('Code Linting') {
      steps {
        echo 'Linting code...'
        sh 'flake8 .'
      }
    }
    stage('Code Build') {
      steps {
        echo 'Building Docker image for web app...'
        sh 'docker build -t irum90/webapp:latest .'
      }
    }
    stage('Unit Testing') {
      steps {
        echo 'Running unit tests...'
        sh 'pytest tests/'
      }
    }
    stage('Containerized Deployment') {
      steps {
        echo 'Running container using docker-compose...'
        sh 'docker-compose down --remove-orphans'
```

```
            sh 'docker-compose up -d --build'
        }
    }
    stage('Selenium Testing') {
        steps {
            echo 'Running Selenium tests...'
            sh 'docker build -t selenium-tests -f Dockerfile.selenium .'
            sh 'docker run --network="host" selenium-tests'
        }
    }
  }
}
```

Created pipeline in Jenkins




Jenkins
New Item
Build History
Add description
All
ld Queue
builds in the queue.
ld Executor Status
f 2 executors busy)
S
W
Name ↓
Last Success
Last Failure
Last Duration
selenium_lab3
2 hr 42 min #13
2 hr 45 min #12
5.2 sec
Icon: S M L