**Lab 6: Continuous Integration with Jenkins and Selenium on AWS**

Objective:*Set up a CI pipeline using Jenkins to automatically trigger Selenium tests.*

Tasks:

1. Install Jenkins on EC2.

2. Create a Jenkins job to run Selenium tests.

3. Store and review test results.

Documentation:

- Basics of Continuous Integration.

- Introduction to Jenkins.

- Integrating Jenkins with Selenium.

Prerequisites:

1- An AWS account with administrative access.

2- Python Automation Course

3- Python Selenium Course

4- Bash Script Deep Dive Course

5- Jenkins Course

6- Git Course

7- Previous Lab completed

Implementation Documentation:

**Basics of Continuous Integration (CI)**

Continuous Integration is a development practice where developers integrate code into a shared repository multiple times a day. Each integration is verified by automated build and test scripts to detect integration errors as quickly as possible.

**Introduction to Jenkins**

Jenkins is an open-source automation server that helps automate parts of the software development process. It facilitates Continuous Integration and Continuous Deployment (CI/CD) by automating building, testing, and deploying code.

**Step 1: Installing Jenkins on EC2**

Launch a new EC2 instance (Choose an OS based on your preference).

Connect to the instance via Remote Desktop Connection.

Install Jenkins using the official instructions for your Linux distribution.e.g., for Ubunut.

| 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-get update sudo apt-get install jenkins  sudo apt update sudo apt install openjdk-17-jre java -version  sudo apt install openjdk-8-jdk java --version  sudo systemctl enable jenkins sudo systemctl start jenkins sudo systemctl status jenkins |
| --- |

Make sure to add the inbound rule for Security Group attached to your EC2 for port 8080.

**Python Script**

Open Pycharm

create new python file selenium\_tests.py

| from selenium import webdriver import time driver = webdriver.Chrome() driver.get("https://aws.amazon.com/") time.sleep(5) driver.quit() |
| --- |

The simple selenium script above opens AWS and keep browser open then closes the browser.

**Connecting with Github**

We need to commit this script to GitHub. Open PyCharm terminal.

| git config --global user.name usamaaslam2017 git config --global user.email usamaaslam2017@gmail.com |
| --- |

#for push you need ssh key

| goto ssh key generate ssh key ssh-keygen -t ed25519 -C "usamaaslam2017@gmail.com" eval "$(ssh-agent -s)" ssh-add ~/.ssh/id\_ed25519 cat ~/.ssh/id\_ed25519.pub |
| --- |

#Now open github account in the browser

copy the key from terminal

Goto setting and then new ssh key

Give key name e.g., usama

paste key here

#Go back to PyCharm Terminal and type following commands

| git init git add . or git add -A git commit -m "first commit" git status |
| --- |

Open browser and create public git repository

#use push commands from there to push main/master repo to the repository

**Integrating Jenkins with Selenium**

open Jenkins in browser

*<Public IP of EC2>:8080*

Goto PyCharm terminal and type

| cat /var/lib/jenkins/secrets/initialAdminPassword |
| --- |

Copy the key and paste in browser to login jenkins

Create Admin user and password at first login

#Install necessary plugins

In the Jenkins dashboard, go to "Manage Jenkins" > "Manage Plugins" and install the necessary plugins, including the Selenium plugin.

#Create a New Jenkins Job

Click on "New Item" in the Jenkins dashboard.

Enter a name for the job (e.g., SeleniumTestJob) and select "Freestyle project". Click "OK".

**Configure the Jenkins Job:**

Under "Source Code Management", choose your version control system (e.g., Git).

In the "Build" section, add a build step to execute your Selenium tests.

| # Navigate to your project directory cd /home/ubuntu/PycharmProjects/pythonProject # Activate your virtual environment source /home/ubuntu/PycharmProjects/pythonProject/bin/activate # Run your Selenium tests python selenium\_tests.py |
| --- |

Save the Configuration.

**Trigger the Job:**

In the Jenkins dashboard, find your job (SeleniumTestJob) and click "Build Now" to trigger a build.

**Conclusion:**

By following these steps, you have successfully set up a Continuous Integration pipeline using Jenkins on an AWS EC2 instance to automatically trigger Selenium tests. This lab demonstrates the power of automating testing as part of the development process for better code quality.