**Lab 9: Running Headless Selenium Tests on EC2**

Objective:*Configure and run headless Selenium tests on an EC2 instance.*

Tasks:

1. Set up necessary dependencies on EC2.

2. Write and execute a headless Selenium test.

Documentation:

- Introduction to headless browsing.

- Configuring EC2 for headless tests.

- Running headless Selenium tests.

Prerequisites:

1- An AWS account with administrative access.

2- Python Automation Course

3- Python Selenium Course

4- Bash Script Deep Dive Course

5- Previous Lab completed

Implementation Documentation:

**Introduction to Headless Browsing**

Headless browsing allows web browsing without a graphical user interface (GUI). It is useful for automated tasks or environments without a display server.

**Configuring EC2 for Headless Tests**We have already configured it in lab - 1 of Selenium, so it's not necessary. We will also use PyCharm for this lab.

Open Pycharm and create a new python file in the virtual environment you created.

| from selenium import webdriver  # Configure Chrome options for headless mode chrome\_options = webdriver.ChromeOptions() chrome\_options.add\_argument('--headless') chrome\_options.add\_argument('--disable-gpu') chrome\_options.add\_argument('--no-sandbox')  # Initialize the WebDriver with Chrome options driver = webdriver.Chrome()  # Open a webpage and perform actions (e.g., capturing screenshots) driver.get("https://alnafi.com") driver.save\_screenshot("headless\_screenshot.png")  # Close the browser driver.quit() |
| --- |

**Explanation:**

* We configure Chrome options to enable headless mode and set certain flags to ensure smooth operation.
* We initialize the WebDriver with these options.
* We open a webpage, perform actions (in this case, capturing a screenshot), and then close the browser.

**Additional Notes:**

* Headless mode is especially useful for running Selenium tests on servers or in CI/CD pipelines where a GUI isn't available.
* Remember to use the appropriate web drivers and browser versions for your specific environment.

**Conclusion:**

By following these steps, you have successfully set up and executed headless Selenium tests on an AWS EC2 instance. This lab demonstrates how to run automated tests in a headless environment, which is often crucial for continuous integration and server-based testing.