**Lab 8: Error Reporting with Selenium and AWS SNS**

Objective:*Send notifications on Selenium test failures using AWS SNS.*

Tasks:

1. Modify Selenium scripts to detect test failures.

2. Integrate AWS SNS to send notifications upon failures.

Documentation:

- Error detection in Selenium.

- Basics of AWS SNS.

- Integrating Selenium with SNS for notifications.

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:

**Error Detection in Selenium**

Detecting errors in Selenium involves checking for elements, expected outcomes, or conditions that indicate a test failure. This could be done through conditional statements, assertions, or exception handling.

**Step 1: Modifying Selenium Scripts for Error Detection**

Here is the Python script in pycharm that we will use for error detection and in next step modify it for integrating with AWS SNS.

| # Import necessary modules from selenium import webdriver # Import the Selenium WebDriver module from selenium.webdriver.chrome.service import Service as ChromeService # Import the ChromeService from webdriver\_manager.chrome import ChromeDriverManager # Import the ChromeDriverManager from selenium.webdriver.common.by import By # Import the 'By' class for locating elements import time # Import the 'time' module for adding delays  # Initialize a Chrome WebDriver instance driver = webdriver.Chrome() # This will open a Chrome browser window  # Navigate to the specified URL driver.get("https://alnafi.com/auth/sign-in")  try:  # Try to locate an element with the name "email1"  element = driver.find\_element(By.NAME, "email1")   # If the element is found, print a success message  print("Element Found")  except:  # If the element is not found, print an error message and mark the test as failed  print("Element not found. Test Failed.")  # Close the browser window driver.quit() |
| --- |

**Basics of AWS SNS**

Amazon Simple Notification Service (SNS) is a fully managed messaging service for both application-to-application (A2A) and application-to-person (A2P) communication. It enables you to send notifications, alerts, or messages to distributed systems and endpoints via a variety of protocols (e.g., email, SMS, HTTP).

**Step 2: Setting Up AWS SNS**

* Log in to your AWS account.
* Go to the SNS dashboard.
* Create a new topic and set its permissions.
* Add subscriptions (e.g., email addresses) to receive notifications.
* Open email and confirm subscription.
* Make sure to remember topic name and AWS account ID as well as region.

**Step 3: Modifying Selenium Scripts for SNS Integration**

| from selenium import webdriver from selenium.webdriver.chrome.service import Service as ChromeService from webdriver\_manager.chrome import ChromeDriverManager from selenium.webdriver.common.by import By import time import boto3  driver = webdriver.Chrome() driver.get("https://alnafi.com/auth/sign-in") try:  element = driver.find\_element(By.NAME,"email1")  print("Element Found") except:  print("Element not found. Test Failed.")    # Initialize the SNS client  sns = boto3.client('sns',  aws\_access\_key\_id='YOUR\_ACCESS\_KEY\_ID',  aws\_secret\_access\_key='YOUR\_SECRET\_ACCESS\_KEY',  region\_name='YOUR\_REGION')   # Publish a message to the SNS topic  sns.publish(  TopicArn='arn:aws:sns:YOUR\_REGION:YOUR\_ACCOUNT\_ID:YOUR\_TOPIC\_NAME',  Message='Selenium test failed. Check logs for details.'  )  # Close the browser driver.quit() |
| --- |

**Explanation:**

* We use Selenium to perform actions on a webpage and check for conditions indicating a test failure.
* If a failure occurs, an exception is raised, and we catch it to handle the failure.
* If a failure is detected, we use boto3 to interact with AWS SNS and send a notification.

**Conclusion:**

By following these steps, you have successfully set up error reporting with Selenium and AWS SNS. This lab demonstrates how to integrate notification services into your Selenium testing framework for timely alerts on test failures.