**Lab 7: Monitoring Web Application Performance with Selenium and CloudWatch**

Objective:*Monitor web application performance metrics using Selenium and send data to AWS CloudWatch.*

Tasks:

1. Develop Selenium scripts to measure web page load times.

2. Push these metrics to AWS CloudWatch.

Documentation:

- Basics of web performance monitoring.

- Introduction to AWS CloudWatch.

- Integrating Selenium with CloudWatch.

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:

**Basics of Web Performance Monitoring**

Web performance monitoring involves measuring and analyzing various metrics related to the speed and responsiveness of a website. Key metrics include page load times, resource loading times, and user interaction responsiveness.

**Introduction to AWS CloudWatch**

Amazon CloudWatch is a monitoring service for AWS resources and the applications you run on the cloud platform. It provides data and actionable insights to monitor your applications, understand and respond to system-wide performance changes, optimize resource utilization, and get a unified view of operational health.

**Step 1: Setting Up AWS CloudWatch**

* Log in to your AWS account.
* Go to the CloudWatch dashboard.
* Create a new custom metric.

**Step 2: Create AWS Access Keys**

Open IAM

Select Your User

Goto Security Credentials tab

Scroll down and click create access keys

Select the Application option

Click create

Download the csv file and keep it safe

**Step 3: Developing Selenium Scripts for Performance Monitoring in PyCharm and Integrating Selenium with CloudWatch**

| from selenium import webdriver import time import boto3  # Initialize the WebDriver (e.g., Chrome) driver = webdriver.Chrome()  # Open the webpage you want to monitor driver.get("https://example.com")  # Measure page load time start\_time = time.time() driver.get("https://example.com") end\_time = time.time() load\_time = end\_time - start\_time  # Initialize the CloudWatch client cloudwatch = boto3.client('cloudwatch',  aws\_access\_key\_id='YOUR\_ACCESS\_KEY\_ID',  aws\_secret\_access\_key='YOUR\_SECRET\_ACCESS\_KEY',  region\_name='YOUR\_REGION')  # Push the metric to CloudWatch cloudwatch.put\_metric\_data(  Namespace='WebPerformance',  MetricData=[  {  'MetricName': 'PageLoadTime',  'Value': load\_time,  'Unit': 'Seconds',  'Dimensions': [  {  'Name': 'WebPage',  'Value': 'example.com'  },  ]  },  ] )  # Close the browser driver.quit() |
| --- |

**Explanation:**

* We use Selenium to open a webpage and measure the time it takes to load.
* We use the time module to record the start and end times.
* We use the boto3 library to interact with AWS services.
* We push the measured load time as a custom metric to CloudWatch.

**Conclusion:**

By following these steps, you have successfully set up web performance monitoring using Selenium and pushed the metrics to AWS CloudWatch. This lab demonstrates how to integrate web application performance monitoring into your AWS environment for real-time insights.