Automation Architecture

Author: Jason Smith

Prompt
We would like for you to describe (using text and/or diagram) how you would integrate the scripts you wrote in the previous questions into a CI/CD environment. These can include integration with cloud based services or locally and describe the reasoning behind choices you've made.

I would integrate my UI automation in a CI/CD pipeline right after the developers have compiled and packaged their front-end code. In this case, it would be the developers working at SwagLabs pushing their user-facing web code for saucedemo.com. Specifically, the process I would be testing would be the full user flow of buying all available t-shirts after logging in.

After the developers compile their code, I would then implement my UI automation test to verify that the full user experience of logging in and then buying all the t-shirts works as expected. This would include integration tests, stress tests, and positive/negative tests. Once that is finished, I would compile my automation test with an attached instructions file and push it along the CI/CD pipeline. The tests would then go through checks of their own to verify that they are validating all web elements in the user flow correctly via staging builds. Once that is complete and sign-off has occurred, we would be safe to deploy the webpage to production.

In the case of the API automation, I would integrate my tests into the CI/CD pipeline after the back-end developers at openweathermap.org have had a chance to update the API with their changes. Once they have compiled and packaged their code, I would use my automation tests to verify that a user trying to access the API to find weather data for London would have no problems doing so. Since the changes would be directly affecting the API, the primary tests to be used would be stress and performance tests.

Once the backend is verified as working, I would push along my tests in the pipeline to verify that the tests themselves are working as expected when checking for any possible bugs. The most robust way to conduct these final checks would be through a staging API environment. After the team is able to verify that all development code and tests are working as expected, we would be good to go for deploying the API changes to production.

Continuous integration pipeline can be set up using services such as Jenkins, CircleCI, Gitlab, etc.