Spring 2020, CSE 4283 / 6283 – Software Testing and Quality Assurance
Assignment-3, 100 Points

Objective

Develop testing and deployment plans that enable continuous deployment of an existing software system that is extended for web access. Create automated acceptance tests (end-to-end testing) and integration (regression) tests.

Scenario

You have been asked to create a web interface for the application you created for Assignment-2. The VP of engineering at your firm also wants to ensure that you can continuously deploy new features, bug fixes, and changes to the application. He also want to ensure the quality of the overall system by adopting a quality assurance and test plan.

You will augment the existing application and make it accessible via a web interface and build & document a deployment pipeline for the system using various tools and cloud infrastructure.

Requirements

Web Interface - Add a web interface to your command line app and make it accessible via the Google Cloud Platform (e.g., container engine [vm], Google App Engine, etc.).

Deployment Pipeline - Setup a deployment pipeline using continuous integration and delivery tools (can make your GitHub project public).

Steps for pushing to production environment:

1. Source control
2. Continuous Integration
3. Static analysis (e.g., code / style linter, static bug checker [e.g., SpotBugs]).
4. Automated unit tests
5. Automated end-to-end tests (at least one per functionality)
6. Automated deploy to staging environment
7. Manual push to production
8. Connect 3rd Party Code Coverage tool —> https://coveralls.io/, others for coverage reporting

Assignment

Project Report (submitted to Canvas as a *PDF* File): Write a report summarizing your efforts. Your report should consist of the following content:

- **(10 pts) Report should be Professionally Organized / Presented** (content should not solely be a list of bullets that delimit the required content - write in prose / paragraph form). NO ORDER SPECIFIED – Include Name, NetID, and Github username, and link to your Github repository on first page of report.

- **(20 pts) Discuss your Deployment Pipeline.** Explain each step, tools used, and benefits provided. What were challenges you encountered (e.g., setup, implementation)?

- **(10 pts) Detailed setup and execution instructions** (put as a separate SECTION in the report).

- **(10 pts) Test Cases for Manual Testing:** Use Test Case Specification to specify 2 tests that should be performed during manual testing phase.

- **(10 pts) Discuss automated unit testing** (using unit testing tool) you performed.

- **(20 pts) Tool Description -** briefly describe tools used in the process (purpose, functionality provided, ease of use, setup, do you recommend others to use it?). How does the tool integrate with the rest of the pipeline?

- **(10 pts) Google Cloud Platform Usage:** Brief discussion of challenges/benefits using Google Cloud Platform.

- **(10 pts) Code Coverage Report:** Include output from code coverage tool. Turn off any ignored lines or disabled functions from coverage analysis (i.e., coverage should indicate actual lines covered by tests and not ignored)

Notes & Resources

- Tools you may consider
    - Continuous Integration
        - CircleCI, TravisCI, Bamboo*, Jenkins*, Team City, others
    - End-to-End Testing / Acceptance Testing / Web Interface Testing
        - [PhantomJS | Nightwatch] —> Selenium, Fitnesse, others
    - Backends/Hosting
        - Google App Engine, Google Kubernetes Engine, Google Compute Engine

- Use GitHub flow or similar to manage contributions (Main pristine branch with feature branches for each function). Can use fork and pull request flow.
- Recommended to use GitHub project boards for task / issue tracking.

Turn-in the report to Canvas. Include the link to your Github repository on the first page.

Grading

Quality of tests and plan - CD / Deployment Pipeline Implementation - Professional presentation of report, grammar/style, assignment instructions followed - Required functionality implemented.