Software Development standard practices (CI/CD, BDD, Automated Regression Tests Suite, Containerisation e.g Docker

Karim Lawani

Agenda

1. What is BDD? Tools to achieve BDD
2. What is regression test? Tools to achieve this
3. What is performance test? Tools to achieve this
4. What is CI/CD? Build Pipelines. Tools to achieve this
5. Containerisation using Docker
6. Demo

LAWANI CONSULTING

What is BDD?

BDD = Behaviour-Driven Development

- narrow communication gaps between team members
- foster better understanding of the customer
- promote continuous communication with the real world examples

Therefore, reduce some common wasteful activities in Software Development:

- Rework caused by misunderstood or vague requirements
- Technical debts caused by reluctance to refactor code
- slow feedback cycles caused by silos and hand-overs

Tools

Gherkin => uses natural language to describe tests that can be understood by non-programmers

Cucumber => tool to execute Gherkin codes

What is regression test?

- a testing technique to confirm no breaking changes to a software systems
- can be run automatically after each changes or at a scheduled time

Tools

- Free Open Source: Selenium
- Commercials: IBM Rational Functional Tester

What is performance test?

- it’s the testing of an application/system using virtual users interactions and measuring how it responds
- 2 Types of perf testing: Load and Stress testings
- Load testing => test system/app under heavy loads using realistic/happy scenarios
- Stress testing => test system/app until it breaks; find breaking point
- Key perf metrics => number of users; response time

Tools

- Free Open Sources: JMeter, Gatling etc…
- Commercial: Gatling Enterprise etc…

What is CI/CD?

- CI = Continuous Integration => refers to integrating, building and testing code within the dev environment. Can use Mock, Fake data or sandbox for 3rd party system integration
- CD = Continuous Delivery => a discipline where you build software in a way that software can be released to production at any time; it's built on CI
- Main Benefits of CD: Reduction of Deployment Risk; Believable/Confident progress; Quick user feedback

What is CI/CD? (cont.)

- To achieve CD you need: a close, collaborative working relationship between everyone involved in delivery (= DevOpsCulture)
- extensive automation of all possible parts of the delivery process using Build/Deployment Pipeline.

Tools

- Free Open Source: Jenkins, TeamCity (up to 4 servers), Bamboo (Bitbucket)
- Commercial:

Containerisation using Docker

- Containerisation => a way to put your app into a container in an isolated fashion
- Main advantages: speedup CI/CD; easy scalability; easy versioning of app; easy app composition

Tools

- Docker => allows developers, sys-admins etc. to easily deploy their applications in a sandbox (called *containers*) to run on the host operating system i.e. Linux

Containerisation using Docker (cont.)

- Key benefit => allows users to package app with all dependencies into standardised unit

Reference: https://www.docker.com/resources/what-container

- Key docker component and architecture

https://docs.docker.com/engine/docker-overview/

LAWANI CONSULTING

Demo: Simple User Management System

- simple App written in Scala

https://bitbucket.org/klawani2/imsp-demo/src/master/

CI setup consists of:

- running unit tests
- building docker image that we use to build our service
- running the build container and compiling our service
- building the Docker image that we run and deploy
- pushing the final image to a Docker registry