Quality Assurance In Microservice Architectures

Krishnan Chandran Irina Barykina

Department of Informatics,
Intelligent Adaptive Systems, UHH

2016

Structure

Goal: explore QA techniques and approaches, that could cope with challenges specific for microservice architectures.
Structure:

- Give an insight of why QA is important and what challenges it meets in microservice architectures.
- Give and overview of QA techniques that are used in all stages of development: from coding phase to production. Explain benefits that we gain, using these techniques.
- Show on the example of metasearch engine how these techniques could be applied and how they could be adapted to different scenarios.
- Provide some examples of software tools and show how they could be used for an implementation of explored QA techniques.

Theoretical part

- Challenges of testing in microservice architectures.
- Types of tests (Cohn Test Pyramid), their purposes, scopes and quantities. Ice Cream Cone antipattern.
- Non-functional testing: performance, security.
- Rapid application deployment as a prerequisite for microservices. Deployment pipeline. Continuous integration, deployment and delivery.
- Releasing strategies: blue/green deployment, canary releasing, smoke tests.
- After release quality assurance: monitoring, DevOpsCulture.

Example

We are going to use an example of metasearch engine (for traveling-related information) throughout our presentation to show how explored techniques could be applied. We also want to show that testing strategies should be adapted to concrete scenarios on the example of:

- Scenario 1: Testing microservices within application.
- Scenario 2: Testing microservices that use third-party service
- Scenario 3: Testing microservices that will be or is already exposed to public domain

Tools

- xUnit framework
- stubbing and mocking (on the example of Mockito)
- smart stubbing with Mountebank
- testing of data passing between services (on the example of SOAP UI)
- consumer driven testing (on the example of Pact)
- End-to-End Testing (BDD Tools, JBehave, Cucumber)

Deployment

RAD and Deployment Pipeline

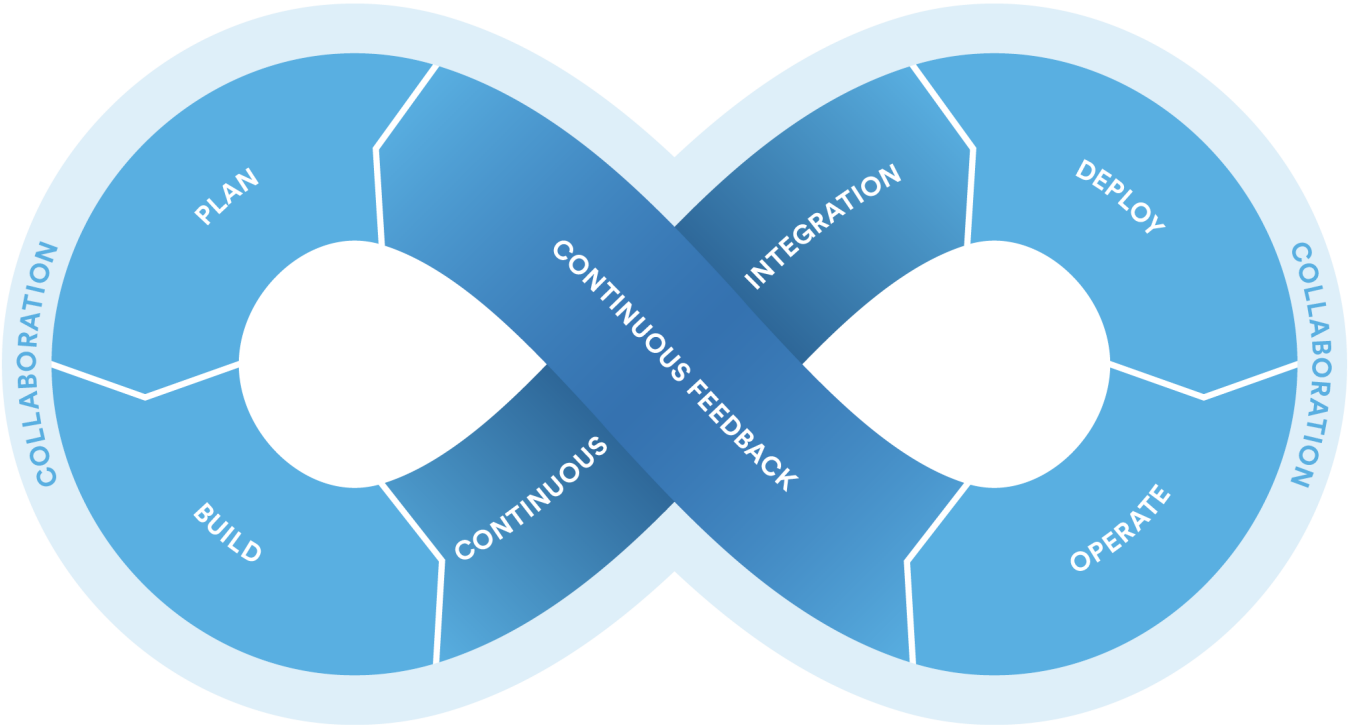
Deployment

Continuous Deployment and Delivery

Deployment
DevOps Culture

DevOps Culture:

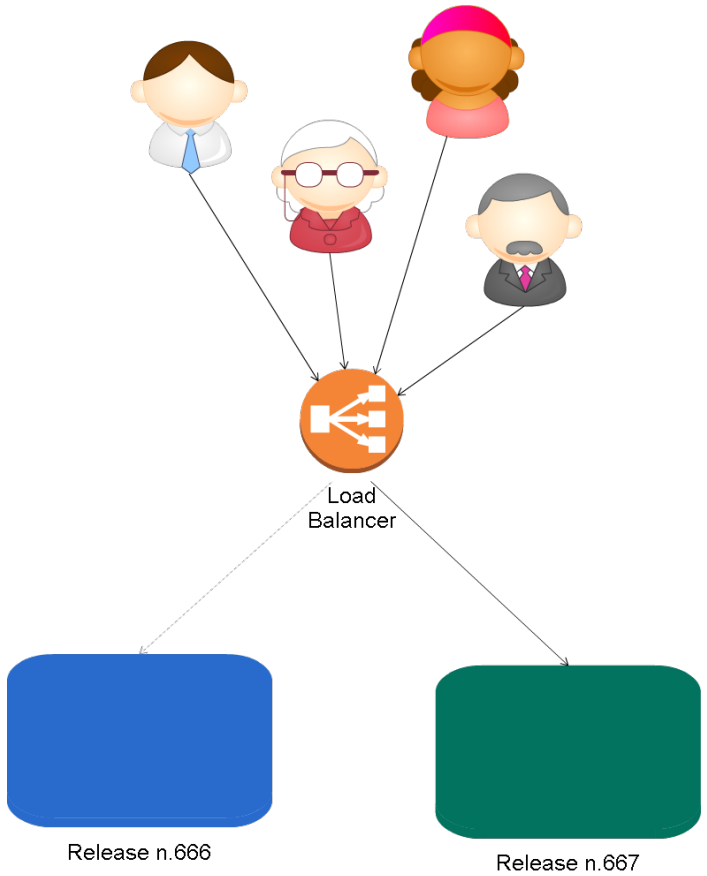
- Aim: break silos between development and later stages
- Requirements: shared responsibility and autonomy of teams



After Deployment

Smart releasing strategies

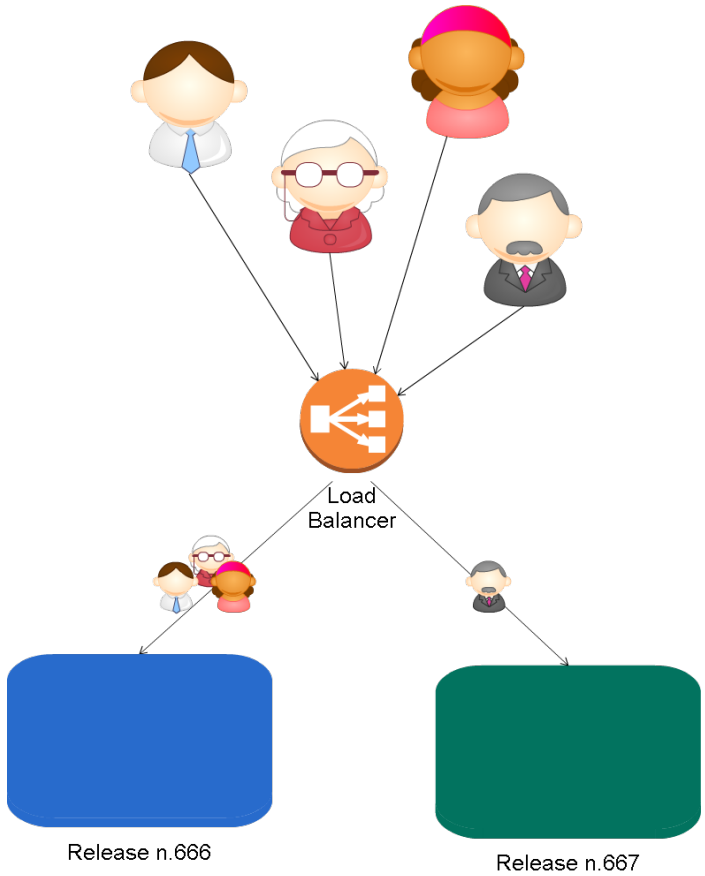
- Smoke Test Suites
- Blue/Green Deployment
- Canary releasing



After Deployment

Smart releasing strategies

- Smoke Test Suites
- Blue/Green Deployment
- Canary releasing



After Deployment

Monitoring

References

Sam Newman. *Building Microservices*. O'Reilly and Associates, 2015.

Mike Cohn. *Succeeding with Agile: Software Development Using Scrum*. Addison Wesley, 2009.

Arvind Sundar. An insight into microservices testing strategies, 2016. URL https://www.infosys.com/it-services/validation-solutions/white-papers/documents/microservices-testing-strategies.pdf.

Toby Clemson. Testing strategies in a microservice architecture, 2014. URL http://martinfowler.com/articles/microservice-testing.

Martin Fowler. Continuousdelivery, 2014. URL http://martinfowler.com/bliki/ContinuousDelivery.html.

Vishal Naik. Architecting for continuous delivery, 2016. URL https://www.thoughtworks.com/insights/blog/architecting-continuous-delivery.