Configuration testing for better DevOps

Anatoly Vasilevskiy
Stockholm, KTH, Chaos Engineering & DevOps Meetup
December 6, 2017
Agenda

• Configuration testing
• Configuration amplification
• Automation of configuration testing
• Demo
Configuration testing in Nutshell

Idea 1: Generate different configurations => Amplification of configurations
Idea 2: Integrate into continuous delivery pipeline => Automation of configuration testing
Challenges: Configuration testing

• Lacking formalization of configurations
  • Docker

• Lacking formalization of variation points
  • Features

• Lacking formalization of valid configuration
  • Constraints

• Lacking tooling to generate various configuration
  • Configuration amplification

• Lacking tooling to integrate into CI pipeline
  • Configuration Testing Framework
Use-case: XWiki

Diverse environments

Diverse topology

Test case

XWiki

MySQL

Postgres

AppServer (tomcat...)

Java (OpenJDK8...)

OS (Unbuntu...)

?
What is configuration amplification

• Environment amplification
  • Monolithic architecture
    • vary libraries, framework versions etc.

• Topology amplification
  • Distributed architectures (microservice, multitier)
    • vary databases, application servers etc.

• Runtime amplification
  • Distributed architectures (microservice, multitier)
    • scaling in/out, restarting services
How to amplify configurations

- Reusable pieces
- Sample config file
- Generated config files

Features and constraints
Configuration model
Constraint solver
Generated models
Environment and topology amplification: XWiki

• Features
  • Linux (Alpine, Ubuntu...), Java (OpenJDK9, OpenJDK8..., IBMJDK9...), AppServer (Tomcat8, Tomcat9...), Database (MySQL5, MySQL9..., Postgres9, Postgres10...)

• Reusable pieces
  • Docker files: Tomcat8 (requires Java, introduces Tomcat8)...
  • Standard Images: OpenJDK8, MySQL9

• Put them together in a meaningful way to cover different features
  • Constraint solving and optimization

• Output:
  • Docker images + Docker compose files – ready to run test with one command
Runtime amplification

- Scale service n up/down
- Delay/lose network packets
- Kill container n
- Load endpoints n with n req/s

Assess resilience/robustness of the system and help defining auto-scaling, self-healing strategies

Globally
Per Service
Per Container

Very initial proof-of-concept: https://github.com/brice-morin/dracarys
Configuration Testing Framework (CTF)

- Pluggable build/test systems
  - maven
  - ant
  - tox
  - etc

- Pluggable reporting systems
  - mvn site
  - ant
  - etc

- Pluggable infrastructure management
  - sidecar
Prototype implementation of CTF
Links & Contacts

• STAMP Project at [https://www.stamp-project.eu](https://www.stamp-project.eu)
• Configuration Testing Framework at [https://github.com/SINTEF-9012/config-testing](https://github.com/SINTEF-9012/config-testing)
• Environment and topology amplification at [https://github.com/SINTEF-9012/ozepy](https://github.com/SINTEF-9012/ozepy)
• Runtime amplification at [https://github.com/brice-morin/dracarys](https://github.com/brice-morin/dracarys)
• Contacts
  • Anatoly Vasilevskiy ([anatoly.vasilevskiy@sintef.no](mailto:anatoly.vasilevskiy@sintef.no))
  • Brice Morin ([brice.morin@sintef.no](mailto:brice.morin@sintef.no))
  • Hui Song ([hui.song@sintef.no](mailto:hui.song@sintef.no))
Q&A