Docker /Maven / Cucumber

pump up your productivity !!

Illia Izotov
Nicolas Bobo
Who are we?

Illia Izotov
i.izotov@criteo.com

Nicolas Bobo
n.bobo@criteo.com
Who are you?
We will see
   technical stuff
   orchestrate deployment
   recipes

We will not see
   maven / docker training
   docker on production
   how to do a backlog
Origin of this project
Technical stack
Frameworks

Apache Storm™
Distributed • Resilient • Real-time

Symfony

Hadoop

Redis

cassandra

Jetty:

NGINX

Microsoft .NET

MariaDB
Languages

Java

Scala

C++

C#

SQL

php

HTML

Javascript

CSS
@scale
Problematic
✓ My code base is growing fast (many developpers)
✓ Manual integration tests – Less and less maintainable
✓ Manual db setup – More and more side effects
✓ Flaky tests
✓ Difficulties to release
✓ Miss of confidence in our releases
✓ Big infra
✓ Difficulty to test and debug locally
KEEP CALM AND STOP THE LINE
We need a strong test harness

- Unit tests
- Integration tests
- Component tests
- End to end tests
Integration tests vs End to end tests vs Component tests

https://martinfowler.com/articles/microservice-testing/

By Toby Clemson
**Unit testing** alone doesn't provide guarantees about the behaviour of the system

https://martinfowler.com/articles/microservice-testing/
An integration test verifies the communication paths and interactions between components to detect interface defects.

https://martinfowler.com/articles/microservice-testing/
A component test limits the scope of the exercised software to a portion of the system under test, manipulating the system through internal code interfaces and using test doubles to isolate the code under test from other components.

https://martinfowler.com/articles/microservice-testing/
An end-to-end test verifies that a system meets external requirements and achieves its goals, testing the entire system, from end to end.
YOU TAKE THE BLUE PILL, THE STORY ENDS.
YOU WAKE UP IN YOUR BED AND BELIEVE WHATEVER YOU WANT TO BELIEVE.

YOU TAKE THE RED PILL, YOU STAY IN WONDERLAND AND I SHOW YOU HOW DEEP THE RABBIT-HOLE GOES.
4 categories of use cases identified

✓ Must
✓ Should
✓ Nice to have
✓ Must not
Must

✓ Give ability to developer to run integration tests locally
✓ Give ability to developer to debug locally
✓ Give back right feedback
✓ Allow to easily run a single test and not a whole test suite run with Not flaky
Should

✓ Report the errors as close as possible to their root cause
✓ Easy to use
✓ Run test quickly, make them parallelizable
✓ Give ability to developer to run e2e tests locally
✓ Allow any kind of test framework (junit, testNG, cucumber, scalatest...)
✓ Send alert in case of failure
Should

- Run integration test from CI manually
- Run only nominal integration test
- Be able to deploy remote cluster
- Easy to deploy locally
- Do not duplicate docker file
Nice to have

✓ Generate fancy reports (HTML...)
✓ Coverage
MUST NOT DO

✓ Be unmaintainable
✓ Be not used by developper
✓ Dump a big dataset
✓ Be flaky by design
✓ Use more ram than what we have on our laptop
Solution
We are craftsmen

E2e, ComponentTests and IT should be written by developpers
We need tools

Docker / Maven / fabric8 / Cucumber match the Usecases identified
In summary
✓ Stop the line
✓ Developper agreement
✓ Define use case
✓ Choose the right technologies
A bit of theory
<table>
<thead>
<tr>
<th>PHASE</th>
<th>A Build Lifecycle is Made Up of Phases</th>
</tr>
</thead>
<tbody>
<tr>
<td>GOAL</td>
<td>A Build Phase is Made Up of Plugin Goals</td>
</tr>
<tr>
<td>PROFILE</td>
<td>Override pom configuration adapt pom at runtime for one</td>
</tr>
<tr>
<td></td>
<td>environment</td>
</tr>
<tr>
<td>PLUGINS</td>
<td>plugins are where much of the real action is performed</td>
</tr>
<tr>
<td></td>
<td>create jar files</td>
</tr>
<tr>
<td></td>
<td>compile code</td>
</tr>
<tr>
<td></td>
<td>unit test code</td>
</tr>
<tr>
<td></td>
<td>create project documentation</td>
</tr>
</tbody>
</table>
We can invoke phase or plugin

`mvn pluginId:goalId`

*ie: mvn surefire:test*

goal test of the plugin surefire
it is attached to the « test » phase
**Docker** is a software technology providing containers, promoted by the company Docker, Inc.
Docker – what’s inside?

- cgroups
- LXC
- Containerd
- Docker
- Docker-compose / Kubernetes / etc.
Basic terms

• Dockerfile
• Image
• DockerHub
• Container
• Volume
• Network
• Docker-compose
Container vs VM

**Container**
- Containers are an abstraction at the app layer that packages code and dependencies together.
- Lightweight

**VM**
- Virtual machines (VMs) are an abstraction of physical hardware turning one server into many servers.
- Full OS emulation
Fabric8 maven plugin

fabric8io/docker-maven-plugin
This is a Maven plugin for managing Docker images and containers. It focuses on two major aspects for a Docker build integration:
• Building images
• Running containers
It provides seamless integration between low-level docker API and maven build phases.
Fabric8 docker maven plugin

Maven plugin invocation

mvn docker:build

mvn docker:start

mvn docker:stop
“Behaviour” is a more useful word than “test”

Dan North
A single source of truth

Cucumber merges specification and test documentation into one cohesive whole.

Focus on the customer

Business and IT don't always understand each other. Cucumber's executable specifications encourage closer collaboration, helping teams keep the business goal in mind at all times.

Living documentation

Because they're automatically tested by Cucumber, your specifications are always bang up-to-date.

Less rework

When automated testing is this much fun, teams can easily protect themselves from costly regressions.
support building, deploying and automating any project.
Let’s practice
Mini criteo

ADVERTISER TRACKER → Send → Persist → STATISTIC DB
cassandra

PUBLISHER TRACKER

GET PRODUCT → Read → PRODUCT DB

Product
Dell XPS 13 9360 - 13.3" - Core i7 7500U - 8 Go RAM - 256 Go SSD - français
UseCase 1: I want to run and debug my project on my dev machine
Let’s focus on advertiser

AdvertiserTracker

/api/api/advertiser-tracker/view?userid=5&productid=8080
UseCase 2 : I want automatic Integration tests
Integration Test - AdvertiserTracker

FRONTEND:3000

AdvertiserTracker

/api/api/advertiser-tracker/view?userid=5&productid=:8080

ENGINE

STATISTIC DB

Write

Read

Write

ADVERTISER TRACKER IT
Integration Test - Engine

FRONTEND :3000

AdvertiserTracker

/api/api/advertiser-tracker/view?userid=5&productid=:8080

ENGINE

WRITE ENGINE

ENGINE IT

Read

Write

STATISTIC DB
UseCase 3 : I want automatic e2e tests
E2E Test – AdvertiserTracker - Engine
UseCase 4: component version is changing

https://store.docker.com/

Tags (27)

- 1.0.0 126 MB  
  Last updated 21 days ago

- latest 126 MB  
  Last updated 21 days ago

- 0.11.0.1 119 MB  
  Last updated 2 months ago

- 0.11.0.0 120 MB  
  Last updated 4 months ago
Take away

Give ability to developer to debug locally

mvn docker:build

mvn docker:start

mvn docker:stop
Take away

Give ability to developer to run integration tests locally

mvn clean install

mvn verify
Take away

Allow to easily run a single test and not a whole test suite

mvn -Dit.test=ITIntegrationTest#testmyFirstIntegrationIT verify

Thank you !!