JenkinsPipelineUnit

Test your Continuous Delivery Pipeline

Ozan Gunalp - Emmanuel Quincerot

Jenkins World

A global DevOps event

2017

Who we are

Ozan Günalp

Developer at LesFurets

PhD in Computer Science

@ozangunalp

Emmanuel Quincerot

Developer at LesFurets

@EQuincerot

LesFurets.com

Jenkins World
A global DevOps event
2017

First independant insurance comparison website in France

Operating since 2012

50 insurers on Vehicle, Home, Health and Loan products

3M quotes per year

Delivery @ LesFurets

20+ developers

30~ developments in parallel

Daily deployment to production since 2014

~220 releases per year

Continuous delivery @ LesFurets

Feature Branching + Continuous Integration

One branch per development
Continuous merge for staging
Daily release of ready branches

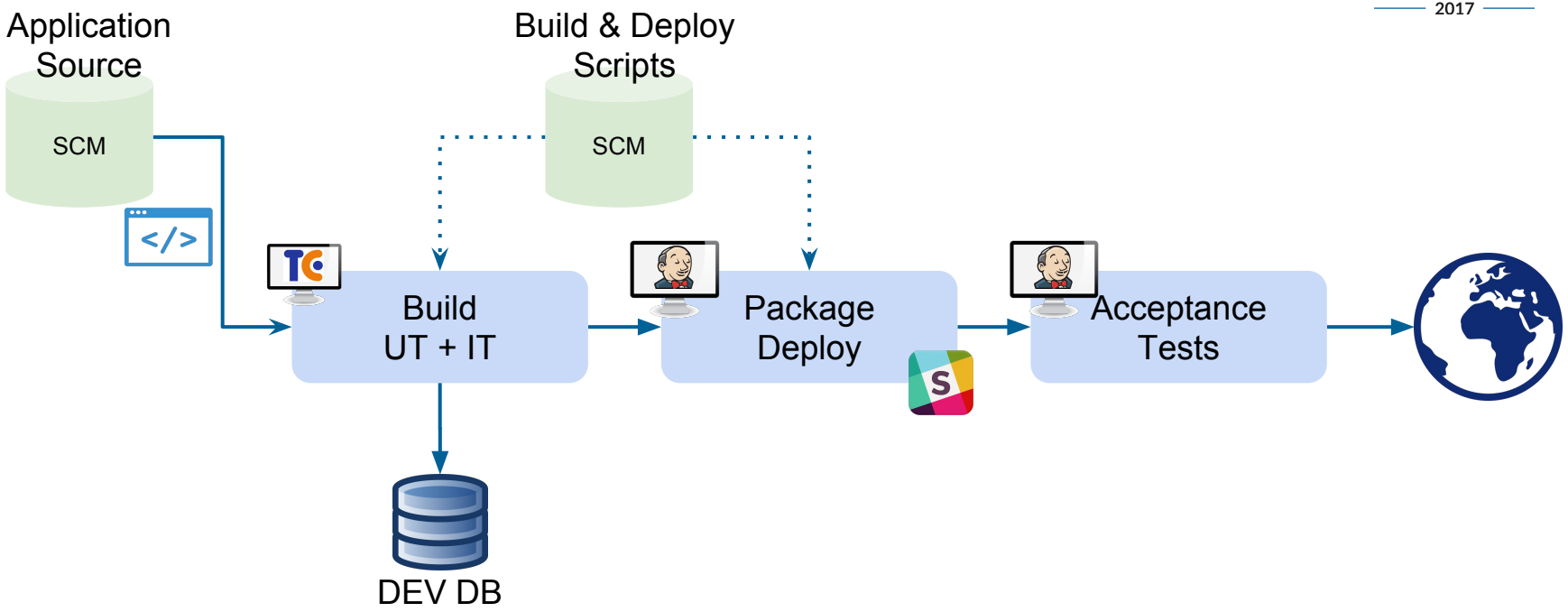
Continuous delivery @ LesFurets

Feature Branching + Continuous Integration

One branch per development
Continuous merge for staging
Daily release of ready branches

Isolated developments
Early feedback

Before: Legacy Continuous delivery

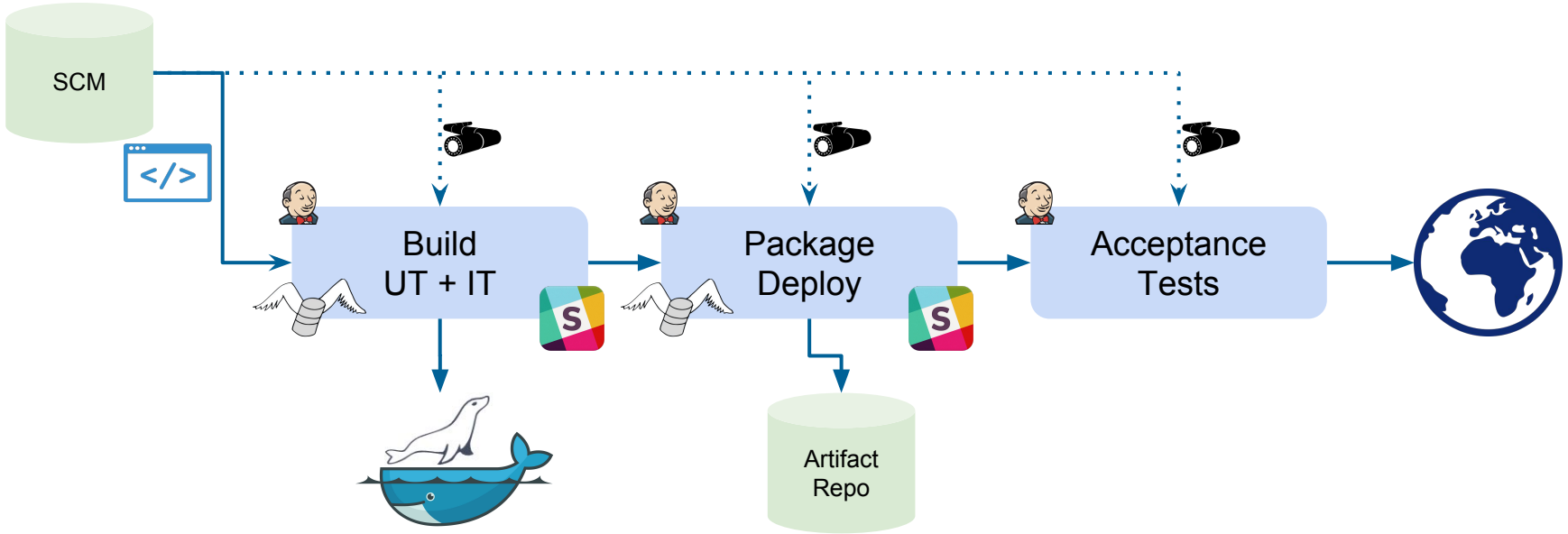


Before: Legacy Continuous delivery

- Teamcity + Jenkins (Configuration in UI)
- Multi-repo (code + deployment scripts)
- Shared development database
- Manual database migration
- No artifact repository
- Hard-coded notifications on Slack

Jobs are broken often by evolutions

After: Jenkins *pipeline-as-code*



After: Jenkins *pipeline-as-code*

- Jenkins pipelines everywhere
- Same repository for code & pipeline
- Automated tests with database on docker
- Fully automated database migration
- Lazy packaging with artifact repository
- Job result feedback with Slack notifications

Every evolution is tested separately and released

pipeline-as-code

Coordinate tasks

Handle errors and retry

Parallelize

Externalize configuration

Separate configuration from workflow

No conf in shell scripts

...and they lived happily after

Jenkins World
A global DevOps event
2017

Migration to Jenkins pipelines

Pay the price of early adoption : discovering Jenkins pipelines with incomplete documentation

Workspaces and multi-branch

One branch validated on many agents

One workspace per branch on each agent

100 branches x 1GB

No automatic cleanup

Workspaces and multi-branch

One branch validated on many agents
One workspace per branch on each agent

100 branches x 1GB

No automatic cleanup

No space left on disk

Workspaces - first try

Clean the workspace

Disk usage problem solved

Clone from scratch, **slow and intensive network usage**

Workspaces - second try

Share the workspace among branches, with:

```
ws('validation') {
    checkout(scm)
    sh 'git clean -xdf'
    ...
}
```

Quick checkout and workspace reused

DRY

- jenkins
  - job
    - deploy.jenkins
    - integration.jenkins
  - lib
    - commons.jenkins
    - jiraApi.jenkins
    - scriptLoader.jenkins
  - step
    - deploy
      - checkPackaging.jenkins
      - flyway.jenkins
      - postDeploy.jenkins
      - seleniums.jenkins
      - tomcat.jenkins
    - integration
      - ci.jenkins
      - octopus.jenkins
      - slack.jenkins

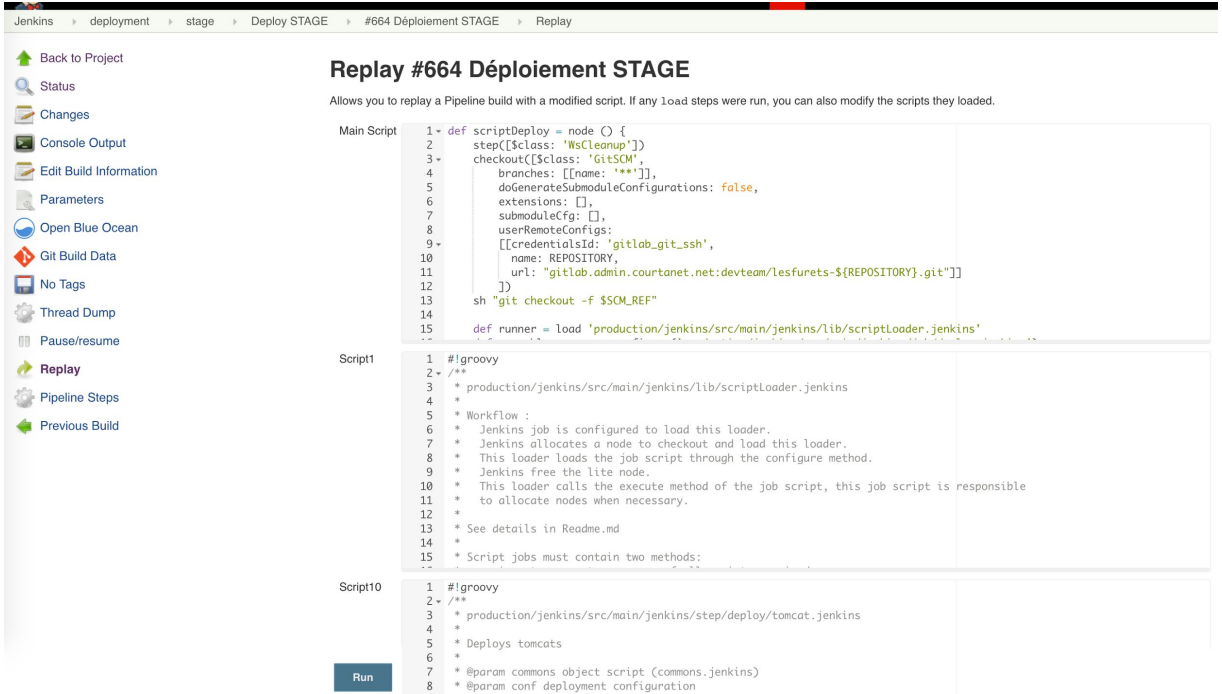
Splitting pipeline scripts
Script loader

```
// jenkins/step/deploy/flyway.jenkins

def execute() {
   node() {
       commons.dlFromRepository(scmHash, 'flyway-migrations.zip')
       sh 'unzip -o flyway-migrations.zip'
       dir('flyway-migrations') {
           sh "./flyway-migrate.sh"
       }
   }
}

return this
```

Development by Trial and Error



Replay is your friend...

Development by Trial and Error


```
java.lang.UnsupportedOperationException: no known implementation of interface
jenkins.tasks.SimpleBuildStep is named FooBar
...
      at org.kohsuke.groovy.sandbox.impl.Checker$1.call(Checker.java:153)
      at org.kohsuke.groovy.sandbox.impl.Checker.checkedCall(Checker.java:157)
      at org.kohsuke.groovy.sandbox.impl.Checker.checkedCall(Checker.java:127)
      at com.cloudbees.groovy.cps.sandbox.SandboxInvoker.methodCall(SandboxInvoker.java:17)
      at Script17.execute(Script17.groovy:14)
      at ___cps.transform___(Native Method)
      at com.cloudbees.groovy.cps.impl.ContinuationGroup.methodCall(ContinuationGroup.java:57)
      at com.cloudbees.groovy.cps.impl.ContinuationGroup.methodCall(ContinuationGroup.java:57)
...
```

...until it is not.

Risky business

One change can cause regression

“Oops! The load test is triggered after deploying the production!”

We need to test and track the impact of our changes!

JenkinsPipelineUnit

Continuous delivery pipelines are described with code

Pipeline code is versioned in our code base

JenkinsPipelineUnit

Unit tests to check what will be executed

https://github.com/lesfurets/JenkinsPipelineUnit

DEMO

Setup

Add dependency from Maven Central

```
com.lesfurets:jenkins-pipeline-unit:1.0
```

Extend the base class

```
class MyTest extends BasePipelineTest
```

Mock variables

```
binding.setVariable('DEPLOY_TO', 'DEV')
```

Mock Jenkins steps

```
helper.registerAllowedMethod('sh', [Map],
    { println "calling sh with $it" })
```

Features

Unit test

Debug in IDE

Callstack print

Non-regression test

```
Jenkinsfile.run()
 Jenkinsfile.node(groovy.lang.Closure)
    Jenkinsfile.stage(Checkout, groovy.lang.Closure)
       Jenkinsfile.checkout(...)
       Jenkinsfile.sh(git clean -xdf)
    Jenkinsfile.stage(Build and test, groovy.lang.Closure)
       Jenkinsfile.sh(./gradlew build)
       Jenkinsfile.junit(build/test-results/test/*.xml)
    Jenkinsfile.stage(Create a Manifest, groovy.lang.Closure)
       Jenkinsfile.sh({returnStdout=true, script=git rev-parse HEAD})
       Jenkinsfile.writeFile({file=MANIFEST.txt, text=2345aef})
       Jenkinsfile.archive(MANIFEST.txt)
```

How does it work?

How does it work?

Transforms Groovy code to Continuation Passing Style with `groovy-cps`, as pipelines in Jenkins

Runs scripts with `GroovyScriptEngine` by injecting variables

Uses Groovy `metaClass` to intercept method calls for mocking steps and generating call stack

vs. JenkinsRule

JUnit @Rule

Spawns Jenkins instance, with plugins

Mandatory for UI tests and plugins

More representative but slower

https://wiki.jenkins.io/display/JENKINS/Unit+Test#UnitTest-Mocking

When to use what?

Developing a plugin ?
JenkinsRule

Continuous delivery pipeline of your organization ?
JenkinsPipelineUnit

Global Shared Library ?
JenkinsPipelineUnit

Declarative pipelines ?
JenkinsPipelineUnit - ?

Tested *pipeline-as-code*

Rapid development lifecycle for pipeline code

Testing in local before pushing to Jenkins

Breaking boundaries between Dev and Ops

Tested *pipeline-as-code*

2/3 of developer team is contributing to pipelines

Optional stages

Lazy packaging

Shared workspace

Deployment scripts optimization

Tests with dockerized database

Cache warmup

Slack notifications

Automated commits

Community

lesfurets/JenkinsPipelineUnit

Framework for unit testing Jenkins pipelines . Contribute to JenkinsPipelineUnit development by creating an account on GitHub.

github.com

Unwatch 42 | Star 192 | Fork 20

Steffen Gebert @StGebert · May 19

Jenkins Pipeline Unit testing looks so diligently maintained by @ozangunalp so that I want to try it immediately!

> **Jenkins** @jenkinsci
> Pipeline Development Tools goo.gl/fb/ckfS9T

1 3 3

Bill Dennis @macg33zr · Jul 27

My automation project has 6k lines of Jenkins pipeline code & 12k lines of Spock unit tests using **JenkinsPipelineUnit** #Jenkins #GroovyLang

1 1 1

CloudBees and 1 other follow

Patxi Gortázar @fgortazar · May 19

A **Jenkins pipeline testing** framework. Test your pipelines before pushing them to your repo

miyata @miyajan · Jul 7

Jenkins Pipeline Unit testing framework 知らなかったけど、lintとは完全に違う方向っぽいので被ることはなさそう。焦ったーw

Translate from Japanese

2

JenkinsPipelineUnit

Unit test

Callstack print

Non-regression test

Debug in IDE

Improved declarative pipelines support

Linting

Open Source @ LesFurets

github.com/lesfurets
JenkinsPipelineUnit
git-octopus

github.com/lesfurets/lesfurets-conferences

@ozangunalp
@equincerot

Jenkins World

A global DevOps event

2017