Jenkins as a code

Łukasz Szczęsny & Marcin Zajączkowski

FOSDEM, 30-31th January 2016

About Łukasz

- Software engineer @ Uber
- FOSS and Open Hardware lover
- Co-organizer of the Warsaw Linux User Group

About Marcin

- Areas of expertise
  - Automatic Testing / TDD
  - Software Craftsmanship / Code Quality
  - Java8 / Groovy
  - Concurrency / Parallel Computing / Reactive Systems
  - **Deployment Automation / Continuous Delivery**
- FOSS projects author and contributor, blogger
- Leads a small software house - Codearte
  - targeted at clients who care about the quality

Agenda

Agenda

- Manual Jenkins maintenance

Agenda

- Manual Jenkins maintenance
- Job configuration as code
  - Jenkins Job DSL

Agenda

- Manual Jenkins maintenance
- Job configuration as code
  - Jenkins Job DSL
- Infrastructure as code

Agenda

- Manual Jenkins maintenance
- Job configuration as code
  - Jenkins Job DSL
- Infrastructure as code
- Case study - Continuous Delivery in Jenkins

Manual Jenkins maintenance

Manual Jenkins maintenance

Manual Jenkins maintenance

- configuration via GUI does not scale
  - slow, error prone, and boring

Manual Jenkins maintenance

- configuration via GUI does not scale
  - slow, error prone, and boring
- problematic with dozens of jobs and plugins

Manual Jenkins maintenance

- configuration via GUI does not scale
  - slow, error prone, and boring
- problematic with dozens of jobs and plugins
- mission impossible with
  - several microservices
  - deployed in several countries
  - for multiple products
  - using deployment pipeline with several steps each

Automation to the rescue!

Jenkins Job DSL

Job configuration in code

Łukasz Szczęsny & Marcin Zajączkowski

@wybczu & @SolidSoftBlog

(Jenkins) Job DSL - 2 parts

(Jenkins) Job DSL - 2 parts

- Domain Specific Language
  - to specify job configuration

(Jenkins) Job DSL - 2 parts

- Domain Specific Language
  - to specify job configuration
- Jenkins plugin
  - to transform configuration DSL into real jobs in Jenkins

Job DSL - part 1 - configuration

Job DSL - part 1 - configuration

- Groovy based DSL (Domain Specific Language)
- job/view/dashboard configuration
- developed as "normal" code in IDE with
  - auto-completion
  - type check
  - Groovy magic if needed
- outside Jenkins instance

Job DSL - simple example

Job DSL - simple example

```
job('FOSDEM-website-publish') {
    scm {
        github('FOSDEM/website')
    }
    triggers {
        scm('*/15 * * * *')
    }
    steps {
        rake('publish')
    }
}
```

Job DSL - dynamic example

```
String repo = 'FOSDEM/mobile-app'

URL branchUrl = "https://api.github.com/repos/$repo/branches".toURL()
List branches = new JsonSlurper().parseText(branchUrl.text)

branches.each { branch ->

    String safeBranchName = branch.name.replaceAll('/', '-')

    job("$repo-$safeBranchName-build") {
        scm {
            github repo, branch.name
        }
        triggers {
            scm 'H/10 * * * *'
        }
        steps {
            gradle 'check'
        }
    }
}
```

Job DSL - features

Job DSL - features

- comprehensive support for Jenkins Core stuff
- extensive support for additional plugins
  - 177 plugins in version 1.42
  - active community - continuous flow of new pull requests

Job DSL - features

- comprehensive support for Jenkins Core stuff
- extensive support for additional plugins
  - 177 plugins in version 1.42
  - active community - continuous flow of new pull requests
- powerful `configuration` block for
  - not yet supported features
  - custom stuff
- virtually everything possible in XML should be achievable

Job DSL - part 2 - Jenkins plugin

Job DSL - part 2 - Jenkins plugin

- installed in Jenkins instance
- used in *seed jobs* on Jenkins
- leverages DSL configuration
- updates jobs & views in Jenkins
  - to bring them to desired state
  - by XML configuration files modification

Job DSL - benefits

Job DSL - benefits

- source code instead of XML or GUI
- single source of truth
- manageable jobs and views
  - backed by SCM
  - reviewable - possibly with pull requests

Job DSL - benefits

- source code instead of XML or GUI
- single source of truth
- manageable jobs and views
  - backed by SCM
  - reviewable - possibly with pull requests
- testable
  - automatic "unit" testing
  - pre-production environment

Job DSL - benefits

- source code instead of XML or GUI
- single source of truth
- manageable jobs and views
  - backed by SCM
  - reviewable - possibly with pull requests
- testable
  - automatic "unit" testing
  - pre-production environment
- scalable
  - hundreds of jobs created/modified in seconds*

Job DSL - drawbacks/limitations

Job DSL - drawbacks/limitations

- quite steep learning curve
- can become hard to understand for complex configurations

Job DSL - drawbacks/limitations

- quite steep learning curve
- can become hard to understand for complex configurations
- small error in DSL can remove all jobs
  - can be easily recreated, but without execution history

Job DSL - drawbacks/limitations

- quite steep learning curve
- can become hard to understand for complex configurations
- small error in DSL can remove all jobs
  - can be easily recreated, but without execution history
- occasional backward compatibility issues with new versions

Job DSL - drawbacks/limitations

- quite steep learning curve
- can become hard to understand for complex configurations
- small error in DSL can remove all jobs
  - can be easily recreated, but without execution history
- occasional backward compatibility issues with new versions
- very old Groovy 1.8.9 - version bundled in Jenkins 1.x

Job DSL - drawbacks/limitations

- quite steep learning curve
- can become hard to understand for complex configurations
- small error in DSL can remove all jobs
  - can be easily recreated, but without execution history
- occasional backward compatibility issues with new versions
- very old Groovy 1.8.9 - version bundled in Jenkins 1.x
- not suitable for global Jenkins configuration management
  - credentials, machine provisioning, Jenkins and plugin update, ...

Infrastructure

Infrastructure challenges

Infrastructure challenges

- install and configure Jenkins master
- install and configure all required dependencies
- install and configure plugins
- create and connect slaves
- add JDK installation
- configure authentication
- create credentials
- ...

Infrastructure toolbelt

- configuration management tools
  - Ansible
  - Puppet
  - Chef
  - Salt
  - etc.
- Groovy console
- Jenkins CLI

Infrastructure toolbelt

- Slave management
  - Swarm plugin
  - Docker plugin
  - SSH slaves

Continuous Delivery in Jenkins

Case study

Łukasz Szczęsny & Marcin Zajączkowski

@wybczu & @SolidSoftBlog

Continuous Delivery

Continuous Delivery

- Clearly defined way how to transform source code into project deployed to production
  - a set of steps arranged into pipeline
  - unified way for various projects/variants/realms

Continuous Delivery in Jenkins

Continuous Delivery in Jenkins

- not a first class citizen in Jenkins 1.x
- bunch of jobs triggering each other
- can be emulated with various plugins
  - Delivery Pipeline Plugin, Build Flow Plugin, Pipeline Plugin, ...

Continuous Delivery in Jenkins

- not a first class citizen in Jenkins 1.x
- bunch of jobs triggering each other
- can be emulated with various plugins
  - Delivery Pipeline Plugin, Build Flow Plugin, Pipeline Plugin, ...
- no easy (and unified) way to setup
- usually even harder to maintain

Continuous Delivery - case study

Continuous Delivery - case study

- custom Continuous Delivery framework
  - on top of Jenkins Job DSL
- one standardized way for Continuous Delivery
- reused in all projects in the company

Continuous Delivery - case study

- custom Continuous Delivery framework
  - on top of Jenkins Job DSL
- one standardized way for Continuous Delivery
- reused in all projects in the company
- Ansible for infrastructure management
- Rundeck for deployment

Continuous Delivery - case study

- custom Continuous Delivery framework
  - on top of Jenkins Job DSL
- one standardized way for Continuous Delivery
- reused in all projects in the company
- Ansible for infrastructure management
- Rundeck for deployment
- open sourced to make live easier to others
  - jenkins-pipeline-dsl - core library
  - sample-jenkins-microservice-pipeline - sample pipeline

Live demo

The whole delivery pipeline with one click!

Summary

Summary

- automation is good for you!
- Jenkins DSL is great but there are other tools
  - jenkins-job-builder
  - Pipeline (formerly Workflow) plugin
  - ...
- looking forward to Jenkins 2.0 release!

Questions?

Thank you

Marcin Zajączkowski

@SolidSoftBlog
http://blog.solidsoft.info
m.zajaczkowski@gmail.com
IRC: szpak@freenode

Łukasz Szczęsny

@wybczu
https://wybcz.pl
luk@{uber.com,wybcz.pl}
IRC: wybczu@freenode