

# Automatizace otravné práce vývojářů pomocí Jenkinsu

Adam Saleh

2010-03-30 Tue

# Outline

Úvod

Problémy

Ukážka

- Denno-denná práca

- Vydanie novej verzie

Ako sme to dali dokopy

- Kontrolovanie PR

- Jenkinsfile a Groovy

- Zdielanie kódu a Jenkinsfile

- Konfigurácia Jenkinsu

- Samostatná pipeline a Job Builder

- Jenkins, konfigurácia a Docker

- OpenShift

Záverom

# Kde pracujem?

- ▶ RedHat Mobile
- ▶ QA v Brně
- ▶ Platforma pre vývoj mobilných aplikácií - **closed-source**
- ▶ Knížnice a integrovanie mobilných aplikácií s OpenShiftom - **open-source**

# Ako to vyzerá?

The screenshot displays the QED1-MBusS2-Live application interface. The top navigation bar includes links for Home, Forms Builder, Forms Themes, Forms Projects, Forms Subscriptions, Forms Lifecycle, and Data Sources. The left sidebar contains a 'user-activities' dropdown and a list of icons for Dashboard, Edit Form, Deploy, Field Rules, Page Rules, Data Sources, Notifications, Subscriptions, Related Projects, and Forms Lifecycle.

The main content area is titled 'uart-testforms-from-figures-2' and features a 'FORM FRAGMENTS' section on the left with a 'Drag to in-order' instruction and a 'Customer Information' fragment. The central form area includes fields for 'Ad Start Date' and 'Ad End Date' (both with YYYY-MM-DD format), an 'Advertisement Size (pixels)' dropdown, and a section for specifying image size. Below this is an 'Upload image you would like to use' section with a note that there are multiple images and a 'Choose' button. A 'Please select the sites you wish to advertise on' section includes checkboxes for Website1, Website2, Website3, Website4, and Other. An 'Include all other sites you wish to advertise on' section has a text input field containing 'http://'. At the bottom, there is a 'Customer Information' section.

On the right side, there is a 'FIELD' section with a dropdown menu set to 'QED1-MBusS2-Live'. Below this is a table of field types:

Field	Field	Form	Preview
Text	Paragraph		
1/3 Number	3/1 Final		
Website	Dropdown		
Radio Buttons	Checkboxes		
Location	Map		
File	Photo Capture		
Signature Capture	Datstamp		
Barcode	1/3 Slider (Horizontal)		
Section Break	Page Break		
Read Only			

At the bottom right, the version information '3.10.3 (C:\w\BMP\QED1-MBusS2-Live) - Copyright © 2017 Prolink' is displayed.

# Poznámky k open-source

- ▶ <https://aerogear.org>
- ▶ <http://feedhenry.org>
- ▶ Pracujeme na ďalších veciach ...

# Na čom naozaj pracujem?

- ▶ automatizovanie
- ▶ Dev-Ops/Jenkins/CI-CD
- ▶ udržiavanie infraštruktúry (OpenShift)

# Konkrétne

- ▶ tím 4 ľudí (`{gryan,mnair,pamccart}@redhat`)
- ▶ pre cca 40-70 našich kolegov
- ▶ takmer rok práce

# Upozornenie

- ▶ opakovať po nás pravdepodobne nieje dobrý nápad
- ▶ infraštruktúra je drahá, konfigurácia je komplikovaná
- ▶ poskladajte si vlastné riešenie
- ▶ ani my nepoužívame všetko naraz




# Čo myslím vlastným riešením?

Technológia	Github PR	Internal Pipelines	Testi
Jenkins 2.0			
Pipeline scripts			
Jenkinsfiles in GitHub			
Jenkins-Job-Builder			
Kubernetes-Slave provider			
OpenShift deployment			
Instrumented Configuration			




# Čo sme potrebovali vyriešiť?

- ▶ Denno-denná práca
  - ▶ kontrola zmien pri PR
  - ▶ byrokracia pri testovaní zmien
- ▶ 1/Mesiac vydanie novej verzie systému
  - ▶ skompilovať kód z 70 rôznych repozitárov
  - ▶ aktualizovať konfiguráciu pre nasadenie na server
- ▶ zároveň aktualizovať zastaralú infraštruktúru

# Jeden z repozitárov:

 This repository

[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)

[feedhenry / fh-mbaas](#)

[Unwatch](#) 26 [Star](#) 1 [Fork](#) 38


[Code](#) [Issues](#) [Pull requests](#) [Projects](#) [Wiki](#) [Insights](#) [Settings](#)

FeedHenry MBaaS Management Service [Edit](#)

[Add topics](#)

116 commits 16 branches 70 releases 19 contributors

Branch: master New pull request [Create new file](#) [Upload files](#) [Find file](#) [Clone or download](#)

 **rachael-oregan** Merge pull request #75 from rachael-oregan/update\_fh-forms Latest commit 4efb3c3 20 hours ago

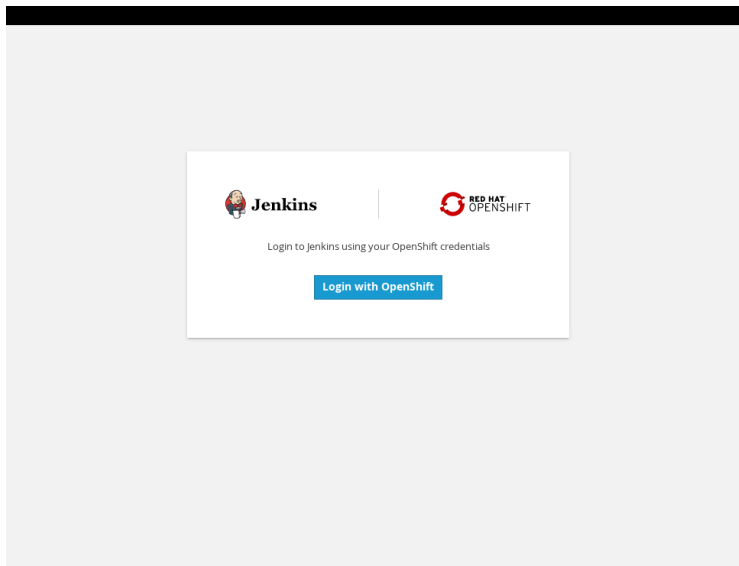
config	update shrinkwrap file	9 months ago
docker	update dockerfiles	2 months ago
docs	fh-mbaas moved to FeedHenry org	a year ago
lib	RHMAP-17599 dont try to stop undeployed app	2 months ago
scripts	fh-mbaas moved to FeedHenry org	a year ago
test	RHMAP-15813 add route to check migration jobs	4 months ago
.dockerignore	added dev dockerfile & updated readme	4 months ago
.gitignore	Use CentOS7 RPM	5 months ago
CONTRIBUTING.md	Add CONTRIBUTING.md	11 months ago
Dockerfile.dev	update dockerfiles	2 months ago
Gruntfile.js	fh-mbaas moved to FeedHenry org	a year ago
Jenkinsfile	RHMAP-17757 - Add Platform Update stage to Jenkinsfile	23 days ago



# Pull-Request

# Pull-Request

# Testy



# Testy

✓ FeedHenry / fh-mbaas 2

Pipeline

Changes

Tests

Artifacts

Pull Request: PR-79

8m 26s

Changes by wilmicko

Commit: 8a212ae

4 hours ago

Pull request #79 updated

Trust

Checkout

Install Dependencies

Lint

Parallel

h-mbaas-pr-79-2-mongodb

Unit Tests

Acceptance Tests

Build

Platform Update

Build Image

Steps Build Image

## Pull-Request

feedhenry / fh-mbaas

Unwatch26★ Star1🔗 Fork38

< CodeIssues0Pull requests4Projects0WikiInsightsSettings

RHMAP-1157 forms version bump #79Edit

Openwitmicko wants to merge 2 commits into feedhenry:master from witmicko:RHMAP-1157\_Repeating\_Sections\_fh-forms\_bump

Conversation0Commits2Files changed2+163-354

witmicko commented 4 hours agoMember+👤✎

#JIRA <https://issues.jboss.org/browse/RHMAP-1157>  
forms bump

forms version bumpcseffa

witmicko changed the title from forms version bump to RHMAP-1157 forms version bump 4 hours ago

shrinkwr

All checks have passed  
3 successful checks

- continuous-integration/jenkins/pr-head — This co...Details
- platform-update/fh-openshift-templates — SuccessDetails
- security/snyk — No new IssuesDetails

Reviewers

Suggestions

- rachael-oregan
- pb82
- psture

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Notifications

# Aktualizácia verzie v inštalátore

✓ FH OpenShift Templates Component Update #52 fh-mbaas (5.9.3-8a212ae)

Pipeline

Changes

Tests

Artifacts

↺

⚙

📄

✕

Branch: — 2m 12s No changes

Commit: — 4 hours ago Started by upstream project "feedhenry/fh-mbaas/PR-79" build number 2

Description changeUrl: <https://github.com/feedhenry/fh-mbaas/pull/79> sourceBranch: RHMAP-1157\_Repeating-Sections\_fh-forms\_bump targetBranch: master

Setup

Checkout

Ensure Target Branch

Ensure Source Branch

Component Version Update

Push Source Branch

Create PR

Update Upstream PR Status

Steps Update Upstream PR Status

No steps This stage has no steps



# Jenkins























**Jenkins**

PipelinesAdministration

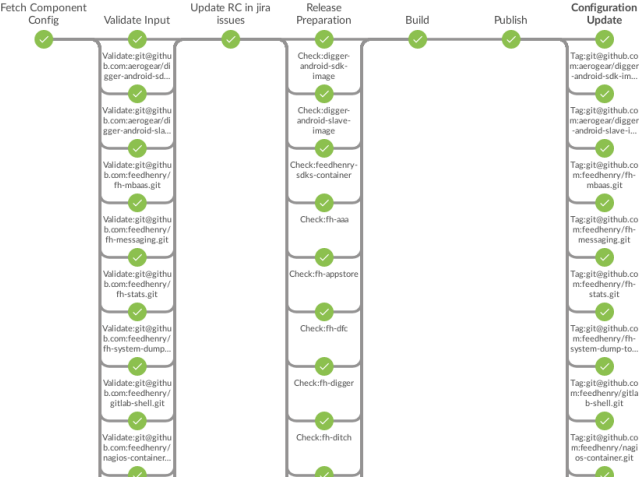
 Logout

Pipelines 

New Pipeline

Name	Health	Branches	PR	
Build Jenkinsfile		-	-	
Fhcap Test Kitchen		-	-	
Jenkins Config Update		-	-	
RHMAP Clear Components Tag		-	-	
RHMAP Cluster Create		-	-	
RHMAP Cluster Deploy		-	-	
RHMAP Cluster Destroy		-	-	
RHMAP Release Candidate		-	-	
RHMAP Release Cleanup		-	-	
RHMAP Release Finish		-	-	
RHMAP Release Start		-	-	

# RC-pipeline




# Osnova


- ▶ Kontrolovanie PR
- ▶ Jenkinsfile a Groovy
- ▶ Zdielanie kódu a Jenkinsfile
- ▶ Konfigurácia Jenkinsu
- ▶ Samostatná pipeline a Job Builder
- ▶ Jenkins, konfigurácia a Docker
- ▶ OpenShift

# GitHub Organization Plugin


**Enter an item name**  
  
= Required field




**Freestyle project**  
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.




**Maven project**  
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.




**Pipeline**  
Orchestrates long-running activities that can span multiple build slaves. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.




**Multi-configuration project**  
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.



**Folder**  
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.




**GitHub Organization**  
Scans a GitHub organization (or user account) for all repositories matching some defined markers.



**Multibranch Pipeline**  
Creates a set of Pipeline projects according to detected branches in one SCM repository.

If you want to create a new item from other existing, you can use this option:



Copy from

OK

# Takto vyzerá nastavenie organizácie

The screenshot shows the Jenkins web interface for configuring a GitHub Organization. The top navigation bar includes the Jenkins logo, a search bar, and the user 'AdamSalch' with a 'log out' link. The left sidebar contains various navigation links, with 'Configure' highlighted. The main configuration area is titled 'Jenkins > Feedhenry >' and includes fields for 'Name' (feedhenry), 'Display Name', and 'Description'. Below these is a 'Projects' section with a 'GitHub Organization' entry. This entry has a 'Credentials' dropdown set to 'github-eng@feedhenry.com/\*\*\*\*\* (GitHub Automaton Credentials)', an 'Owner' field set to 'feedhenry', and a 'Behaviors' section. The 'Behaviors' section includes a 'Filter by name (with regular expression)' field with a regular expression 'fn-mbaas|fn-messaging|fn-metrics|fn-openshift-templates|fn-status|fn-template-apps|fn', a 'Discover branches' section with a 'Strategy' dropdown set to 'Exclude branches that are also filed as PRs', and a 'Discover pull requests from origin' section with a 'Strategy' dropdown set to 'The current pull request revision'. At the bottom of the configuration area are 'Save' and 'Apply' buttons. The left sidebar also shows a 'Build Queue' section with 'No builds in the queue.' and a 'Build Executor Status' section with a list of executors: 'master' (1 idle, 2 idle, 3 idle, 4 idle, 5 idle) and 'jenkins-slave-ruby-fncap-4-84q1-67825391'.

Jenkins

Search

AdamSalch | log out

Jenkins > Feedhenry >

Up

Status

Configure

Scan Organization Now

Scan Organization Log

Organization Events

Delete Organization

People

Build History

Project Relationship

Check File Fingerprint

Open Blue Ocean

Job Config History

GitHub

Config Files

Pipeline Syntax

Credentials

Build Queue

No builds in the queue.

Build Executor Status

master

1 idle

2 idle

3 idle

4 idle

5 idle

jenkins-slave-ruby-fncap-4-84q1-67825391

Name

feedhenry

Display Name

Description

[Plain text] Preview

Projects

GitHub Organization

Credentials

github-eng@feedhenry.com/\*\*\*\*\* (GitHub Automaton Credentials)

Add

Owner

feedhenry

Behaviors

Repositories

Filter by name (with regular expression)

Regular expression

fn-mbaas|fn-messaging|fn-metrics|fn-openshift-templates|fn-status|fn-template-apps|fn

Delete

Within repository

Discover branches

Strategy

Exclude branches that are also filed as PRs

Delete

Discover pull requests from origin

Strategy

The current pull request revision

Delete

Save

Apply

# Odporúčam upresniť repozitáre

Behaviors

Repositories

Filter by name (with regular expression)

Regular expression

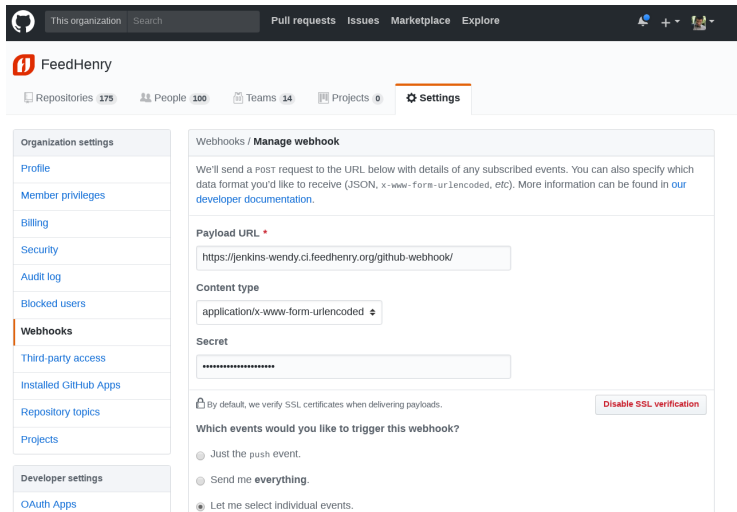
fh-mbaas|fh-messaging|fh-metrics|fh-openshift-templates|fh-stats|fh-template-apps|fh-pipeline-library|fh-dataman|fh-aaa-client|nagios-container|



Delete

Within repository

# Na strane GitHubu treba nastaviti hook




The screenshot shows the GitHub interface for the 'FeedHenry' organization. The top navigation bar includes the GitHub logo, a search bar, and links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below the navigation bar, the organization name 'FeedHenry' is displayed, followed by tabs for 'Repositories' (175), 'People' (100), 'Teams' (14), 'Projects' (0), and 'Settings' (which is the active tab).


On the left side, there is a sidebar with 'Organization settings' and 'Developer settings'. Under 'Organization settings', the 'Webhooks' option is highlighted. The main content area is titled 'Webhooks / Manage webhook' and contains the following information:


- A description: 'We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in [our developer documentation](#).'
- A 'Payload URL' field with the value: `https://jenkins-wendy.ci.feedhenry.org/github-webhook/`
- A 'Content type' dropdown menu set to 'application/x-www-form-urlencoded'.
- A 'Secret' field with a masked value (represented by dots).
- A note: 'By default, we verify SSL certificates when delivering payloads.' with a 'Disable SSL verification' button.
- A section titled 'Which events would you like to trigger this webhook?' with three radio button options:
  - ☐ Just the push event.
  - ☐ Send me **everything**.
  - ☒ Let me select individual events.


# toto nám dá automatickou kontrolu


 **Jenkins** search


Jenkins > FeedHenry >


 Up


 **Status**


 Configure


 Scan Organization Now


 Scan Organization Log


 Organization Events


 Delete Organization


 People


 Build History


 Project Relationship


 Check File Fingerprint


 Open Blue Ocean


 Job Config History

 GitHub


 Config Files

 Pipeline Syntax

 Credentials

























**Build Queue** 

No builds in the queue.

 **FeedHenry**

Folder name: feedhenry

**Repositories (35)**

S	W	Name ↓	Description
		<a href="#">fh-aaa-client</a>	FeedHenry Authentication, Authorization, and Accounting (AAA) Client
		<a href="#">fh-amqp-js</a>	AMQP client for node.js
		<a href="#">fh-cachekey-client</a>	
		<a href="#">fh-cluster</a>	Wraps node cluster module to allow cleaner usage
		<a href="#">fh-component-metrics</a>	Node module for gathering metrics within feedhenry components
		<a href="#">fh-config</a>	Utility module to parse FH configuration files
		<a href="#">fh-dataman</a>	Feedhenry mbaas data access api used for making mongodb queries
		<a href="#">fh-db</a>	
		<a href="#">fh-fhc</a>	FeedHenry CLI, the Command Line Interface to FeedHenry
		<a href="#">fh-forms</a>	Cloud API for form submissions.
		<a href="#">fh-gridfs</a>	A wrapper module for grid-fs to provide version control for files
		<a href="#">fh-internal-stats</a>	





## Takto vyzerá jeden z našich Jenkinsfile

```
@Library('fh-pipeline-library') _
node('nodejs4') {
    step([$class: 'WsCleanup'])
    stage('Checkout') {
        checkout scm
    }
    stage('Install Dependencies') {
        npmInstall {}
    }
    stage('Unit tests') {
        sh 'grunt fh-unit'
    }
    stage('Build') {
        gruntBuild { name = 'fh-mbaas-api' }
    }
}
```


...takto bez zdielanej knižnice?

```
node('nodejs4') {  
  step([$class: 'WsCleanup'])  
  stage ('Checkout') {  
    checkout scm  
  }  
  stage('Install Dependencies') {  
    sh "npm install"  
  }  
  stage('Unit tests') {  
    sh 'grunt unit'  
  }  
  stage('Build') {  
    sh "grunt dist"  
  }  
}
```

# syntax?

- ▶ <https://jenkins.io/doc/pipeline/steps/>
- ▶ <https://jenkins.io/doc/pipeline/examples/>
- ▶ <https://jenkins.io/doc/pipeline/steps/workflow-basic-steps/>


# experimenty a vývoj s tlačítkom Replay


 **Jenkins**


search


AdamSaleh | log out


Jenkins > FeedHenry > fh-aaa-client > Pull Requests (1) > PR-6 > #1 > Replay


 Back to Project


 Status


 Changes


 Console Output


 Edit Build Information


 Delete Build


 Git Build Data


 No Tags


 Open Blue Ocean


 Open Blue Ocean

 Open Blue Ocean

 Git Build Data

 Rebuild

 **Replay**

 Pipeline Steps

## Replay #1

Allows you to replay a Pipeline build with a modified script. If any Load steps were run, you can also modify the scripts they loaded.


Main Script

```
1 #!groovy
2
3 // https://github.com/feedhenry/fh-pipeline-library
4 @Library('fh-pipeline-library') _
5
6 - stage('Trust') {
7     enforceTrustedApproval()
8 }
9
10 - fhBuildNode {
11 - stage('Install Dependencies') {
12     npmInstall {}
13 }
14
15 - stage('Lint') {
16     sh 'grunt eslint'
17 }
18
```

[Pipeline Syntax](#)

Run

# formulár na generovanie kódu

 **Jenkins**

search

AdamSaleh | log out

Jenkins » FeedHenry » fh-aaa-client » Pull Requests (1) » PR-6 » Pipeline Syntax

[Back](#)

[Snippet Generator](#)

[Step Reference](#)

[Global Variables Reference](#)

[Online Documentation](#)

[IntelliJ IDEA GDSDL](#)

**Overview**

This **Snippet Generator** will help you learn the Pipeline Script code which can be used to define various steps. Pick a step you are interested in from the list, configure it, click **Generate Pipeline Script**, and you will see a Pipeline Script statement that would call the step with that configuration. You may copy and paste the whole statement into your script, or pick up just the options you care about. (Most parameters are optional and can be omitted in your script, leaving them at default values.)

**Steps**

Sample Step 

git: Git

Repository URL

git@github.com:adamsaleh/fh-aaa.git

Branch

master

Credentials

jenkins (Jenkins GitHub SSH Credentials)

Add

☒ Include in polling?

☒ Include in changelog?

Generate Pipeline Script


git credentialsId: 'jenkinsgithub', url: 'git@github.com:adamsaleh/fh-aaa.git'

**Global Variables**

There are many features of the Pipeline that are not steps. These are often exposed via global variables, which are not supported by the snippet generator. See the [Global Variables Reference](#) for details.




# zoznam sprístupnených premenných


 **Jenkins**


search


AdamSaleh | log out


Jenkins > FeedHenry > fh-aaa-client > Pull Requests (1) > PR-6 > Pipeline Syntax


 Back

 Snippet Generator

 Step Reference

 **Global Variables Reference**

 Online Documentation

 IntelliJ IDEA GDSDL

## Overview

Global variables are available in Pipeline directly, not as steps. They expose methods and variables to be accessed within your Pipeline script.

## Global Variable Reference

### Variables

[openshift](#)

#### Global Variable: openshift

The openshift variable offers convenient access to Openshift-related functions from a Pipeline script. Each method will be documented using the following conventions:

```
Method with closure: receiver.methodName(requiredParameter:type[, optionalParameter:type]):returnType {...closure body...}
```

```
Method with return value: receiver.methodName(requiredParameter:type[, optionalParameter:type]):returnType
```

Return types will may be standard types (String, List<String>, bool, int, etc.) or complex types, specific to this plugin, which have behaviors of their own (e.g. [Result](#), [Selector](#), [RolloutManager](#)). These return types are detailed after the methods have been described.

## nie úplne groovy

- ▶ všetok kód musí byť serializovateľný
- ▶ pred májom (květen) nefungovalo

```
list.each { param ->  
    println "Param: ${param}"  
}
```

- ▶ už funguje, ale treba mať jenkins >2.7
- ▶ pri problémoch sa dá použiť anotácia @NonCPS

# práca s @NonCps

```
@NonCPS
def printParams(list) {
  list.each { param ->
    println "Param: ${param}"
  }
}
```



# deklaratívna syntax?

- ▶ Existuje aj `https://jenkins.io/doc/book/pipeline/syntax/#declarative-pipeline`
- ▶ Integrované s UI
- ▶ Zatiaľ sme nepoužili, príliš veľa zjednodušení

# UI pre deklarativnu syntax

**Jenkins**PipelinesAdministrationLogout

Cancel

```
graph LR; Start((Start)) --> Install((Install)); Install --> TestUnit((Test unit)); Install --> TestIntegration((Test Integration)); TestUnit --> End1((+)); TestIntegration --> End2((+));
```

The diagram shows a Jenkins pipeline starting with a 'Start' node, followed by an 'Install' step (highlighted in blue). From 'Install', the pipeline branches into two parallel paths: 'Test unit' and 'Test Integration'. Both paths end with a terminal node marked with a plus sign (+).

← Install ...

**Shell Script**  
npm install

**Shell Script**  
npm install --dev

+ Add step

## ...v skutočnosti bez zdieľanej knižnice

```
node('nodejs4') {  
  step([$class: 'WsCleanup'])  
  stage ('Checkout') {  
    checkout scm  
  }  
  stage('Install Dependencies') {  
    sh '''  
      npm install --production  
      npm ls  
      npm install -g https://github.com/bucharest-go/  
      license-reporter --ignore-version-range --all -  
      npm install  
    '''  
  }  
}
```

# máme nakonfigurovanú knižicu

## Global Pipeline Libraries

Sharable libraries available to any Pipeline jobs running on this system. These libraries will be trusted, meaning they run without "sandbox" restrictions and may use @Grab.

Library

Name

fh-pipeline-library

Default version

master

Currently maps to revision:  
61e16e86b00c4dff5cf46a8992a5b44c079edef5

Load implicitly

☐

Allow default version to be overridden

☒

Retrieval method

☒ Modern SCM

Source Code Management

☒ Git

Project Repository

git@github.com:feedhenry/fh-pipeline-library.git

Credentials

jenkins (Jenkins GitHub SSH Credentials)

Add

Behaviors

Within Repository

Discover branches

Delete

## ktorá ukazuje na repozitár

- ▶ `https://github.com/feedhenry/fh-pipeline-library`

- ▶ `./vars/npmInstall.groovy`

```
def call() {  
    sh '''  
        npm install --production  
        npm ls  
        npm install -g https://github.com/bucharest  
        license-reporter --ignore-version-range --a  
        npm install  
        npm install grunt-cli -g  
    '''  
}
```

# Takto môžeme knižnicu použiť

```
@Library('fh-pipeline-library') _  
node('nodejs4') {  
    step([$class: 'WsCleanup'])  
    stage ('Checkout') {  
        checkout scm  
    }  
    stage('Install Dependencies') {  
        npmInstall()  
    }  
}
```

# Čo ak potrebujem testovať s verziou vo vývoji?

```
@Library('fh-pipeline-library@PR-128') _  
node('nodejs4') {  
    step([$class: 'WsCleanup'])  
    stage ('Checkout') {  
        checkout scm  
    }  
    stage('Install Dependencies') {  
        npmInstall()  
    }  
}
```

# Čo ak potrebujem zmeniť funkciu?

- ▶ toto robíme opatrne
- ▶ čo ak sa niektorý z 70 repozitárov bude spoliehať na starú verziu?

- ▶ fh-pipeline-library/vars/dockerBinaryBuild.groovyy

```
def call(String name, String version, String docker
    echo "Deprecated: this should no longer be need
    params = [name: name, org= dockerHubOrg ...]
    buildWithDockerStrategy params
}
```

- ▶ nová funkcia je v  
fh-pipeline-library/vars/buildWithDockerStrategy.groovy



## fh-pipeline-library vyzerá dôležito, ako ju testujete?

- ▶ existujú unit-testy, ale zatiaľ sme ich nepoužili
- ▶ <https://github.com/macg33zr/pipelineUnit>
- ▶ máme vo fh-pipeline-library Jenkinsfile
- ▶ Jenkinsfile vie zistiť či má testovať zmenenú knižnicu
- ▶ <https://github.com/feedhenry/fh-pipeline-library/blob/master/Jenkinsfile>

## samo-testovanie v fh-pipeline-library

```
env.BRANCH_NAME = env.BRANCH_NAME ?: 'master'
String gitref = env.CHANGE_ID ? "pr/${env.CHANGE_ID}" : "master"
def fhPipelineLibrary = library("fh-pipeline-library@${gitref}")
def utils = fhPipelineLibrary.org.feedhenry.Uutils.new()

testStage('getReleaseBranch') {
    print utils.getReleaseBranch('1.2.3')
}
```

# Otázky?

- ▶ podľa mňa som práve skončil s tým pekným a zaujímavým :)
- ▶ náleďuje ifraštruktúra ktorá drží po kope izolepou :)

# Všetko čo som tu doteraz ukazoval nastavujeme cez skripty

- ▶ Organizácie na GitHubu
- ▶ fh-pipeline-library
- ▶ účet na prihlasovanie do
  - ▶ GitHubu (kde máme zdrojáky)
  - ▶ AWS (kde nám bežia servery)
  - ▶ Jira (kde máme issue-tracking)
- ▶ a veľa ďalších vecí

# Ukážka

- ▶ interné repo
- ▶ config.ini

```
[jenkins]
user=updateme
password=updateme
url=http://updateme.com:9999
query_plugins_info=False
```

# konfiguračný skript

- ▶ scripts/configurejenkins.sh

```
SCRIPTS_DIR="$( cd "$( dirname "${BASH_SOURCE[0]}" )" &
CONFIG="$(cd "$(dirname "$1")"; pwd)/$(basename "$1")"
do_groovy_script_command() {
    $SCRIPTS_DIR/run_groovy_script.sh $CONFIG $REMOTE $SC
}
#Git (Git plugin)
do_groovy_script_command 'set_git_global_config.groovy'

#Credentials
do_groovy_script_command 'add_ssh_credentials.groovy'
```

# Groovy skript na nastavenie gitu

```
import jenkins.model.*
def inst = Jenkins.getInstance()
def desc = inst.getDescriptor("hudson.plugins.git.GitSCM")
desc.setGlobalConfigName("Jenkins")
desc.setGlobalConfigEmail("jenkins@wendy.feedhenry.net")
desc.save()
```

## Groovy na nastavenie ssh kľúča

```
import jenkins.model.*
import com.cloudbees.plugins.credentials.*
import com.cloudbees.plugins.credentials.common.*
import com.cloudbees.plugins.credentials.domains.*
import com.cloudbees.plugins.credentials.impl.*
import com.cloudbees.jenkins.plugins.sshcredentials.impl.*
import hudson.plugins.sshslaves.*;
domain = Domain.global()
store = Jenkins.instance.getExtensionList('com.cloudbees
credId = 'jenkinsgithub'
username = 'jenkins'
privateKeyStr = "-----BEGIN RSA PRIVATE KEY-----\nMIIJ"
privateKeySource = new BasicSSHUserPrivateKey.DirectEnt
description = "Jenkins GitHub SSH Credentials"

privateKey = new BasicSSHUserPrivateKey(CredentialsScop
store.addCredentials(domain, privateKey)
```



## ... trochu viac o izolepe

- ▶ tieto skripty spúšťame cez jenkins-cli
- ▶ ide o +/- ekvivalent spúšťania skriptov v Jenkins konzole
- ▶ \$JENKINS/script



### Script Console

Type in an arbitrary [Groovy script](#) and execute it on the server. Useful for trouble-shooting and diagnostics. Use the 'println' command to see the output (if you use System.out, it will go to the server's stdout, which is harder to see.) Example:

```
println(Jenkins.instance.pluginManager.plugins)
```

All the classes from all the plugins are visible. jenkins.\*, jenkins.model.\*, hudson.\*, and hudson.model.\* are pre-imported.

1

Run

# Jenkins Job Builder

- ▶ na konfiguráciu samostatnej pipeline používame
  - ▶ <https://docs.openstack.org/infra/jenkins-job-builder/baca7d15504a93c9d92f15e034470cbdc86f2615>
- ▶ s pluginom na podporu pipeline
  - ▶ <https://github.com/rusty-dev/jenkins-job-builder-pipeline>

```
jenkins-jobs --conf config.ini update $DIR_WITH_YAML
```

# Yaml konfigurácia pre Pipeline

```
- job:
  project-type: pipeline
  concurrent: true
  display-name: RHMAP Cluster Destroy
  name: rhmap_cluster_destroy
  parameters:
    - string:
      default: ''
      description: Cluster Name
      name: clusterName
  pipeline:
    script: |
      #!groovy
```

## .. v skutočnosti komplikovanejšie

- ▶ Používame Jenkinsfile ktorý je v rovnakom adresári

pipeline:

```
script-path: jobs/wendy/cluster/cluster-destroy/Jenkinsfile
scm:
  - git:
      branches:
        - 'master'
      url: 'git@github.com:fheng/jenkins-bob-build'
      credentials-id: jenkinsgithub
      skip-tag: true
      wipe-workspace: false
```

- ▶ väčšinou odkazujeme na GH
  - ▶ netreba updatovať
- ▶ máme skript na konverziu, pre testovacie účely

## Kubernetes plugin

Kubernetes	Name	<input type="text" value="openshift"/>
	Kubernetes URL	<input type="text" value="https://172.30.0.1:443"/>
	Kubernetes server certificate key	<pre>-----BEGIN CERTIFICATE----- MIICGyCCAdKgAwIBAgIATANBgqhkiG9w0BAQsFADAMSQwlgYDVQQDDbtvcGVu c2hpZnRlc2NlbnMvYVQEOOTQMzA3MzkwhNmTcwNTExMTUzODU5WhcNMjIw MTUzOTAwWjAmMSQwlgYDVQQDDbtvcGVuc2hpZnRlc2NlbnMvYVQEOOTQMzA3Mzkw gtEIMA0CSGsgSib3QEBAQUAA4BDBwkwggEKAAoIBAQCnxZ7s54mgYr7Zdsxx019H7 CQuuwewylJMFY+NblAFydwmSOEEepOLGt9ZN8UX/Ch1I03nBl2Oy+IR4yDgKPI WLEUJfaDYArTdXkubxGRfQAIdKnUGGI8PMODOMLONGO+EPNPNEPH3UdDwjKW FAEQqAHmInrbkIZe8+RXvcAdem4MKRLUuJMy6jxVPB4sk+qZoRDKm6TSWS sYfbv9VL7VE5Yifl4xna6ELIXSBHEtmzgRlsX7+bic97e1Z3ldmyUB3C3uGNWJp VEFDkgZclHJFMSSzBmh86fg7Wky1bcyGKUJEEdndVOUDJPbmPbHTkvzmKLnlb AgEMBAAGcjzAhMA4CA1UdhImEWQcJawCPDApBgNVHRMBAf8EBTADAQHMAOGCSgs Sib3QEBCwIAA4IBAQBwxYqVmsMuLcAt1J4pcEethXcu7ZkuXU+12Av2iwkud4P kFDRPsBMilHIU890W3MbM4OpzJYPfdjtzh859s84BVtkpXu/799MCTjhOXxtf NW4aRhUAazZlhpeTO2M/GNyqAKtebtgrZAenPgUAY98we1sau8S6SK2K6Kew4 VdbYIAIVOpmlkNm8bsYkceJQmLJA2AuAS+yxmZcuUAWedZAEVnzis6SWWC2c e3Zwq38HYRnG+g50M1jfeOSG8LuGttwSVSHovZYQHdzCDRFYm6S2tlDUVFg3 HaGS8CriPEca240IBPEZ/BldXRvYdsHfdrJLnstc -----END CERTIFICATE-----</pre>
	Disable https certificate check	<input type="checkbox"/>
	Kubernetes Namespace	<input type="text" value="wendy"/>
	Credentials	<input type="text" value="1a12dfa4-7fc5-47a7-aal17-cc56572a41c7"/> <span>Add ▼</span>
	Jenkins URL	<input type="text" value="http://172.30.168.75:80"/>
	Jenkins tunnel	<input type="text" value="172.30.164.50:50000"/>
	Connection Timeout	<input type="text" value="0"/>
	Read Timeout	<input type="text" value="0"/>
	Container Cap	<input type="text" value="10"/>

Test Connection

# Príklad - jenkins-slave-ansible

Jenkins » configuration

### Kubernetes Pod Template

Name

Namespace

Labels

Usage

The name of the pod template to inherit from

### Containers

#### Container Template

Name

Docker image

Always pull image ☒

Working directory

Command to run slave agent

Arguments to pass to the command

Allocate pseudo-TTY ☐

EnvVars

List of environment variables to set in slave pod

List of container in the slave pod

EnvVars

List of environment variables to set in all container of the pod

Volumes

# Príklad - jenkins-slave-ansible

- ▶ <https://hub.docker.com/r/fhwendy/jenkins-slave-ansible/>
- ▶ zdroják
  - ▶ [https://github.com/feedhenry/wendy-jenkins-s2i-configuration](https://github.com/feedhenry/wendy-jenkins-s2i-configuration/blob/master/slave-ansible/Dockerfile)
  - ▶ [/blob/master/slave-ansible/Dockerfile](https://github.com/feedhenry/wendy-jenkins-s2i-configuration/blob/master/slave-ansible/Dockerfile)

# jenkins-slave-ansible Dockerfile

- ▶ používame vlastný base-image
- ▶ pôvodne sme používali  
openshift/jenkins-slave-base-centos7

```
FROM docker.io/fhwendy/jenkins-slave-base-centos7:latest
```

```
MAINTAINER Oleg Matskiv <omatskiv@redhat.com>
```

```
USER root
```

```
RUN yum -y install \
    epel-release \
    wget &&
```



# Jenkins Master

- ▶ openshift/jenkins-2-centos7
  - ▶ s ďalšou konfiguráciou cez s2i
  - ▶ <https://github.com/feedhenry/wendy-jenkins-s2i-configuration>
  - ▶ zoznam pluginov s verziami
  - ▶ vyčistenie prednastavených jobov
- ▶ s2i zoberie openshift/jenkins-2-centos7 a aplikuje konfiguráciu z repa

# Prečo vlastne používame openshift?

- ▶ Pretože sme ho už používali na iné projekty
- ▶ Plánovali sme postaviť nový produkt na kombinácii Openshift+Jenkins
- ▶ Konfigurácia == Kód

```
oc login
```

```
oc new-project jenkins-wendy
```

# Slave konfigurácia v openshifte

## ► slave-image-template.yaml

---

apiVersion: v1

kind: Template

labels:

template: slave-image-wendy-template

metadata:

annotations:

tags: jenkins

name: slave-image-wendy

objects:

- apiVersion: v1

kind: ImageStream

metadata:

name: \${IMAGE\_NAME}

annotations:

template: slave-image-wendy-template

# Parametre a nastavenie

```
parameters:
```

- description: The name of the repository branch to ref  
  displayName: Branch  
  name: IMAGE\_NAME
- description: The label  
  name: SLAVE\_LABEL

```
oc new-app -p SLAVE_LABEL="ansible" -p IMAGE_NAME=slave
```

# Jenkins Master integrovaný s openshiftom

- ▶ autorizácia/autentizácia cez openshift
- ▶ automatické limity na spustený slave
- ▶ jednoduchšia integrácia s i.e. nexus3

# Pipeline integrovaná s openshiftom

- ▶ prístup k openshiftu na ktorom beží
- ▶ používame openshift na integrovanie databázy
  - ▶ `https://github.com/feedhenry/fh-pipeline-library/blob/master/vars/withOpenshiftServices.groovy`

```
withOpenshiftServices(['mongodb']) {  
    stage('Acceptance Tests') {  
        sh 'grunt fh-accept'  
    }  
}
```

# Openshift ako build-server

- ▶ používame openshift na update docker-hubu
  - ▶ <https://github.com/feedhenry/fh-pipeline-library/blob/master/vars/dockerBuildNodeComponent.groovy>

```
stage('Build Image') {  
    dockerBuildNodeComponent(COMPONENT, DOCKER_HUB_ORG)  
}
```

# Zhrnutie

- ▶ odporúčam používať groovy, aj s knižnicami
- ▶ konfigurácia v kóde je super aj ak lepená izolepou
- ▶ ak už niekde máte openshift, použite ho



Ďakujem za pozornosť!

# Otázky?