Simple CI / CD with Jenkins & Docker

December 2017 @ Devstaff

Petros Koumantarakis

petros@webtrails.gr

Basic JS React SPA Application

based on create-react-app

Environment: Node

Package Manager: npm

Bundler: Webpack

Build: Node based script to bundle JS, CSS, and images for production

Unit testing: Jest

Unit testing reporter: jest-junit

Deployment artifact: Docker image

Jenkins Pipeline

"A *continuous delivery pipeline* is an automated expression of your process for getting software from version control right through to your users and customers"

- Repeatable
- Version able (Process as "code")
- Automated

Jenkins Pipeline

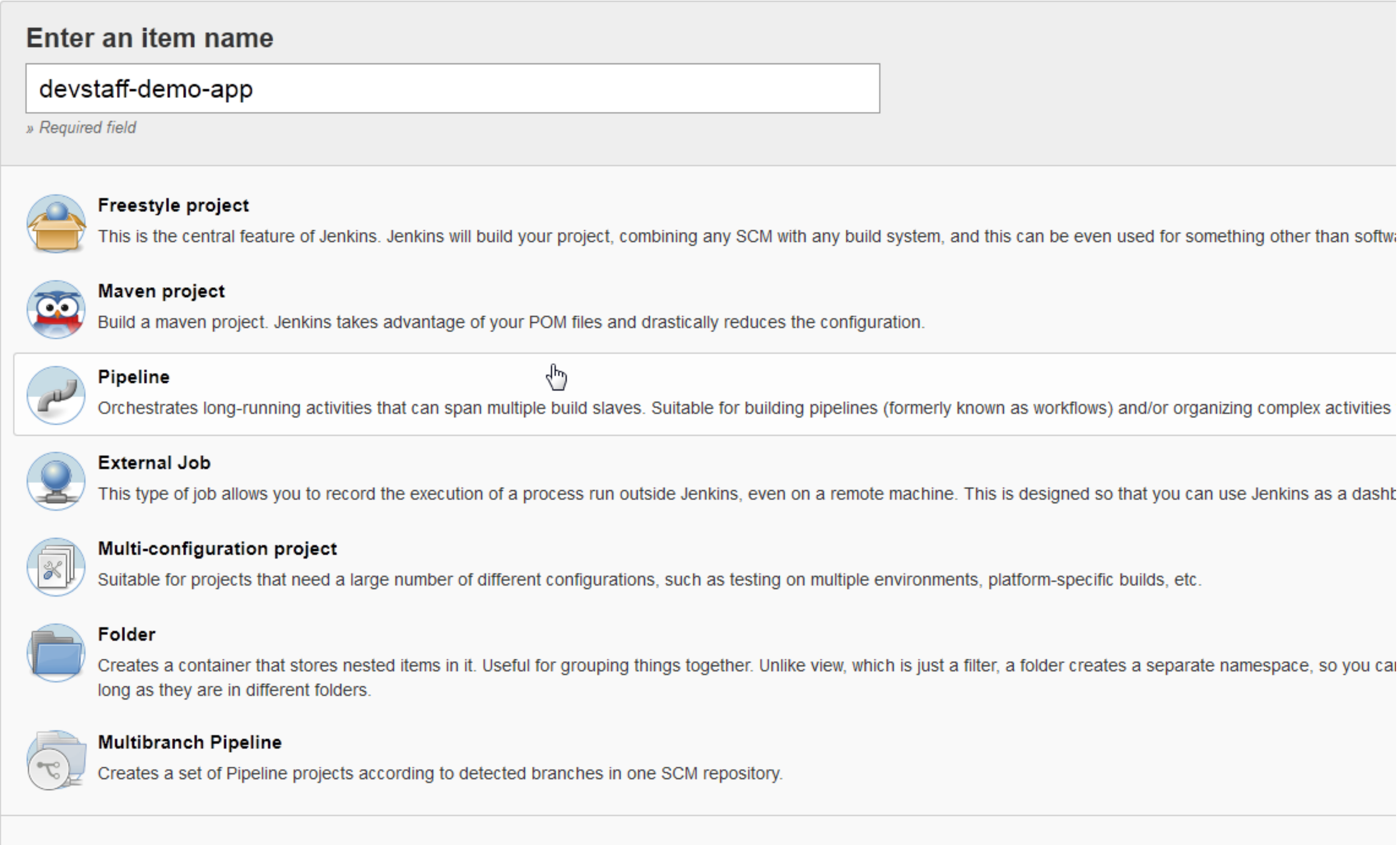
Simple or Complex

Serial and / or parallel execution

Stages & steps

```
pipeline { /* Declarative Jenkins pipeline */
    agent any
    stages {
        stage('SCM') {
            steps {
                echo 'Checkout...'
            }
        }
        stage('Build') {
            steps {
                echo 'Building...'
            }
        }
    }
}
```

Jenkins Pipeline



Enter an item name

devstaff-demo-app

» 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 softw

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

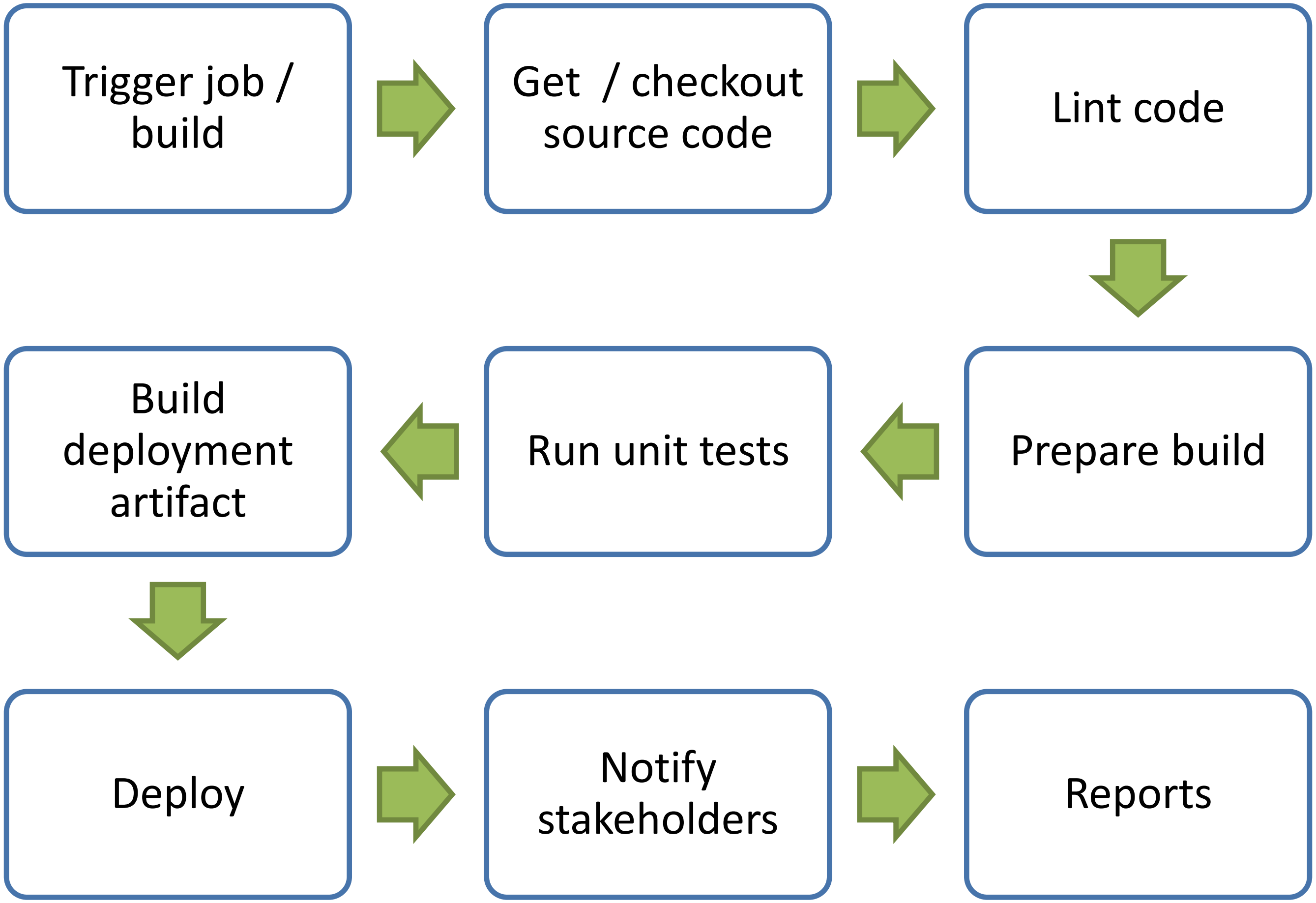
External Job
This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashb

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 ca long as they are in different folders.

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

Pipeline - Flow



Trigger job / build

Get / checkout source code

Lint code

Prepare build

Run unit tests

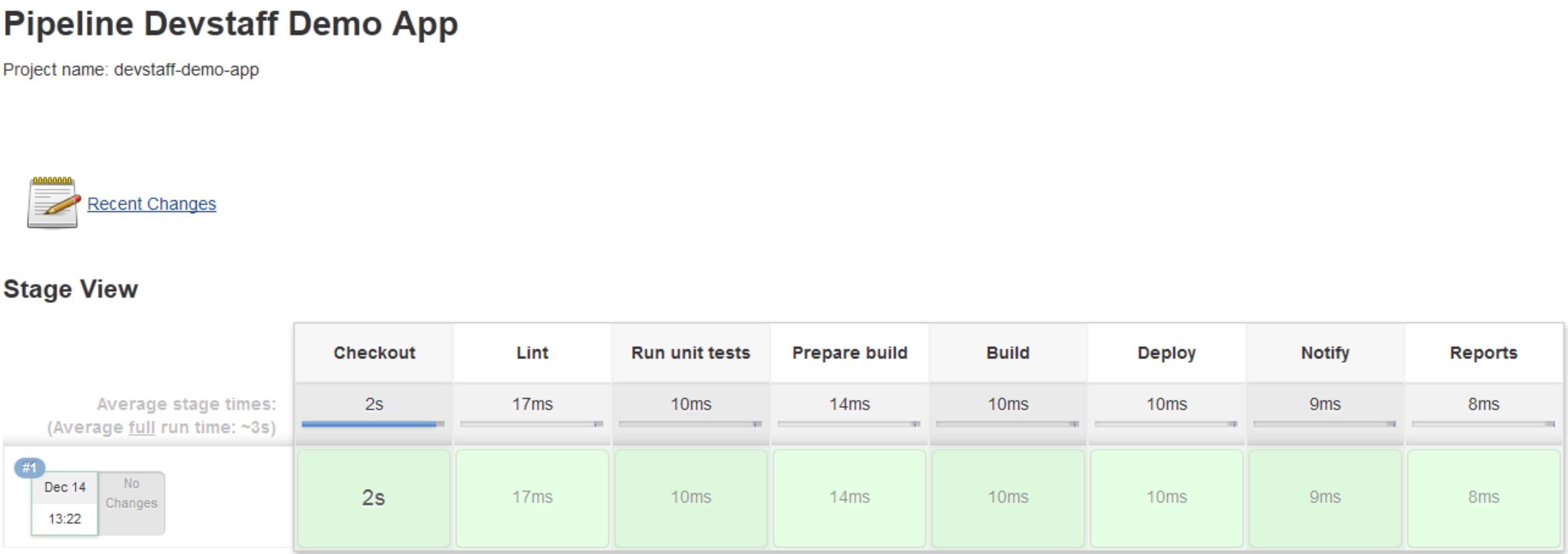
Build deployment artifact

Deploy

Notify stakeholders

Reports

Pipeline - Flow (Stage View)



Pipeline Devstaff Demo App

Project name: devstaff-demo-app

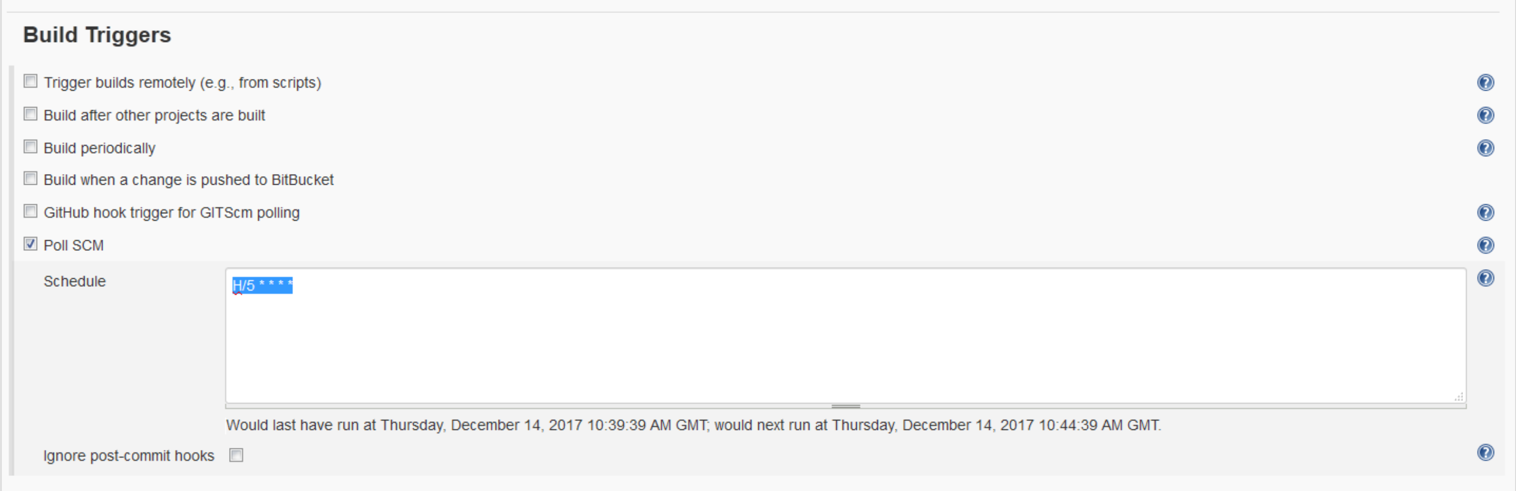
Recent Changes

Stage View

	Checkout	Lint	Run unit tests	Prepare build	Build	Deploy	Notify	Reports
Average stage times: (Average full run time: ~3s)	2s	17ms	10ms	14ms	10ms	10ms	9ms	8ms
#1 Dec 14 13:22 No Changes	2s	17ms	10ms	14ms	10ms	10ms	9ms	8ms

Flow (Trigger Execution)

Multiple ways: SCM polling, SCM hooks (github, bitbucket etc)



Build Triggers

- ☐ Trigger builds remotely (e.g., from scripts)
- ☐ Build after other projects are built
- ☐ Build periodically
- ☐ Build when a change is pushed to BitBucket
- ☐ GitHub hook trigger for GITScm polling
- ☑ Poll SCM

Schedule

```
H/5 * * * *
```

Would last have run at Thursday, December 14, 2017 10:39:39 AM GMT; would next run at Thursday, December 14, 2017 10:44:39 AM GMT.

Ignore post-commit hooks ☐

Pipeline - Flow (Checkout / Lint)

```
stages {
  stage('Checkout') {
     /* Get source code */
     steps {
       git 'https://github.com/pkouman/devstaff-demo-app'
     }
  }
  stage('Lint') {
   /* check source code for correctness, style */
    steps {
       sh 'npm run lint'
    }
  }
  ...
```

Flow (Prepare Build / Unit testing)

```
…
 stage('Prepare build') {
       steps {
           /* install build & project dependencies */
           sh 'npm install'
       }
   }
   stage('Run unit tests') {
       steps {
           /* run tests and create junit report */
           sh 'CI=true npm run test -- --testResultsProcessor=jest-junit'
       }
   }
…
```

Pipeline - Flow (Build)

```
…
stage('Build') {
      steps {
        /* transform & bundle dependencies */
        sh 'npm run build'
        /* build the actual Docker image */
        script {
            image = docker.build("pkou/devstaff-demo-app")
        }
        /* image.withRun(…) can be used to execute
tests on running container */
      }
   }
…
```

Pipeline - Flow (Deploy)

…

```
stage('Deploy') {
    steps {
        echo 'deploying'
        /* Push the image with tags: incremental build number, latest tag */
        /* alternative registries e.g. ECR supported by plugins */
        script {
            docker.withRegistry('https://registry.hub.docker.com', 'docker-hub-credentials') {
                image.push("${env.BUILD_NUMBER}")
                image.push("latest")
            }
        }
        /* pull new image & deploy via script, ansible, cloud service etc */
        /* usual “dance”: pull, stop, remove, tag, run */
        sh 'ssh deploy@staging "/opt/deploy.sh"'
    }
}
```

…

Pipeline - Flow (Notify & Report)

```
…
   stage('Notify') {
       steps {
        echo 'notify'
        /* send message to team hipchat channel */
        hipchatSend credentialId: "hipchat-token", room: "jenkins", message: "Build finished: ${env.JOB_NAME} ${env.BUILD_NUMBER}"
       }
   }
   stage('Reports') {
       steps {
           echo 'reports'
           /* pickup junit report from jest-junit reporter */
           junit './**/junit.xml'
       }
   }
…
```

Pipeline - Flow (Notify on failure)

```
pipeline {

   ...

   post { /* additional steps upon completion */
       /* notify on pipeline failure */
       failure {
           hipchatSend credentialId: "hipchat-token", room: "jenkins", message:
"Build has failed!: ${env.JOB_NAME} ${env.BUILD_NUMBER}", color: 'RED'
       }
    }

    ...
}
```

Pipelines – Endless possibilities & Integrations

- Use Docker containers as build agents
  - Build / Run tests in multiple platform versions
  - Isolated environments
  - Consistent / repeatable container image building
- SCM
  - Merge pull requests, branches
  - Tag releases, versions
  - Archive build / release artifacts
  - Provide review / merge conflict feedback
- Code Quality
  - Code coverage reports
  - Vulnerability / Security assessment reports
- Ticketing Systems
  - Tag / update issues
  - Create release notes
- Pipelines
  - Build/Start dependent pipelines / projects (upstream / downstream)

Jenkins Pipeline

https://jenkins.io/

https://jenkins.io/doc/book/pipeline/

https://jenkins.io/doc/book/pipeline/syntax/

https://jenkins.io/doc/book/pipeline/syntax/#compare

https://jenkins.io/doc/tutorials/building-a-node-js-and-react-app-with-npm/