DevOps 2TIN

Chapter 5b

CI in jenkins

DE HOGESCHOOL MET HET NETWERK

Elfde-Liniestraat 24, 3500 Hasselt, www.pxl.be

CI in jenkins

Credential manager
jenkinsfile (github)
compiling code
Global tool configuration
unit testing
artifacts
input steps

HOGESCHOOL PXL

Wat is een Pipeline?

Definitie “deployment pipeline”:
(first defined by Jez Humble and David Farley in their book *Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation*)

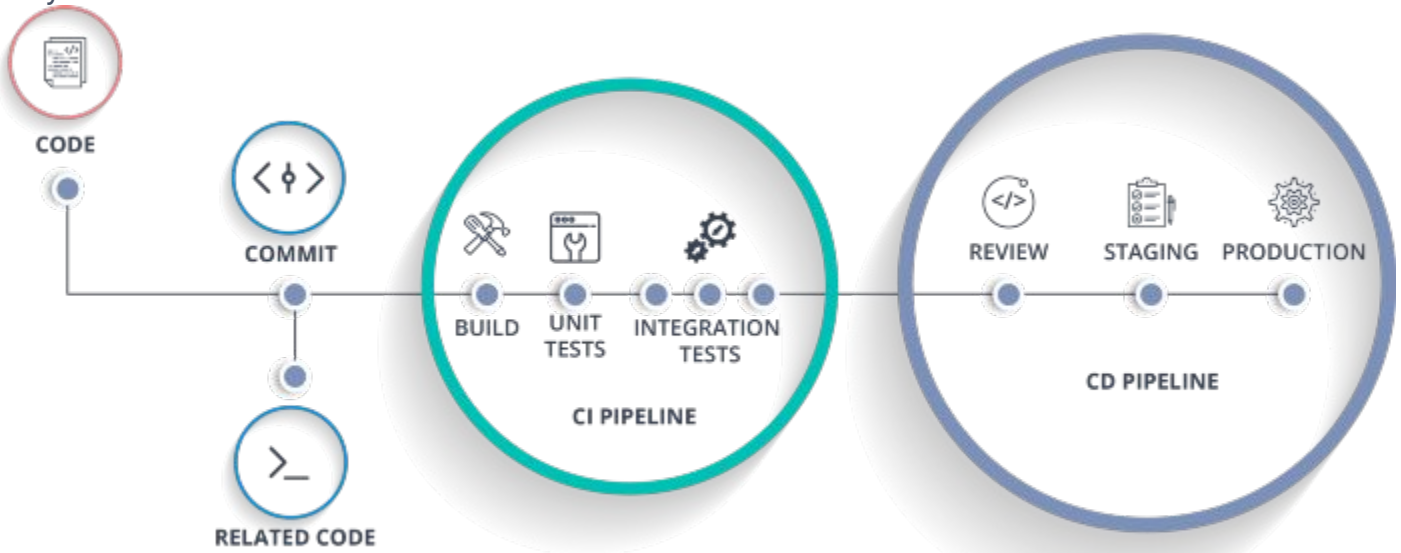
It ensures that all code checked in to version control is automatically built and tested in a production-like environment.

Sleutelwoorden:

- Alle Code
- Versiebeheer
- **Automatisch gebouwd**
- Automatisch getest
- Productie-waardige omgeving

Recap: CI/CD pipeline

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

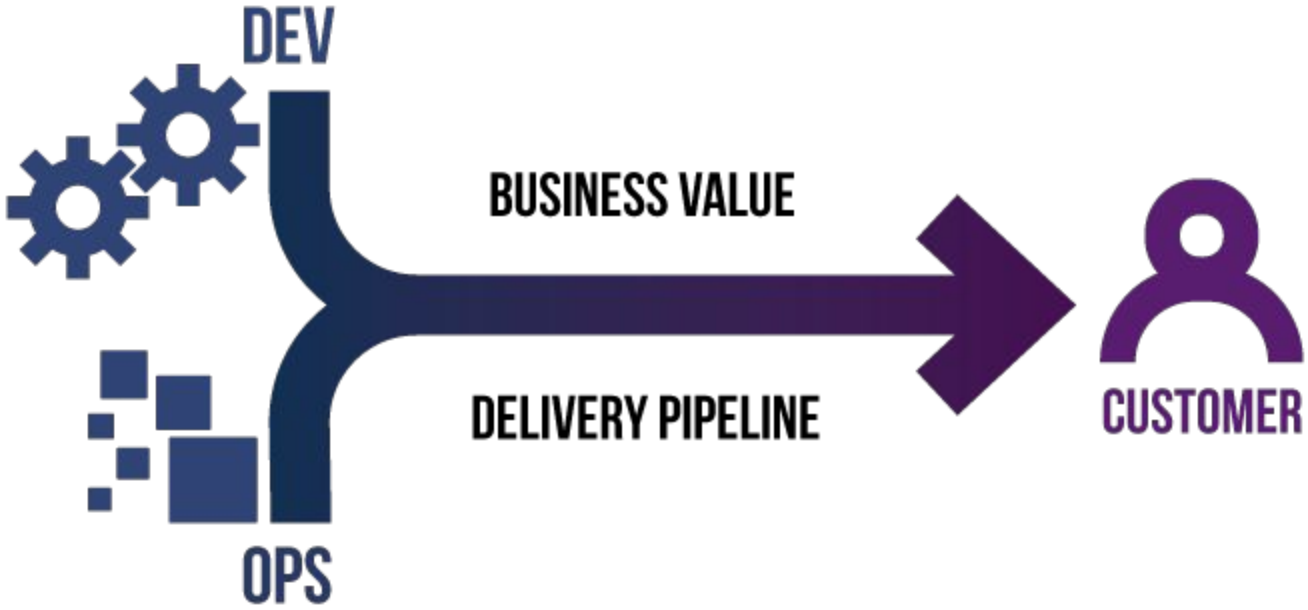


Every change to your software (committed in source control) goes through a complex process on its way to being released. This process involves building the software in a reliable and repeatable manner, as well as progressing the built software (called a "build") through multiple stages of testing and deployment.

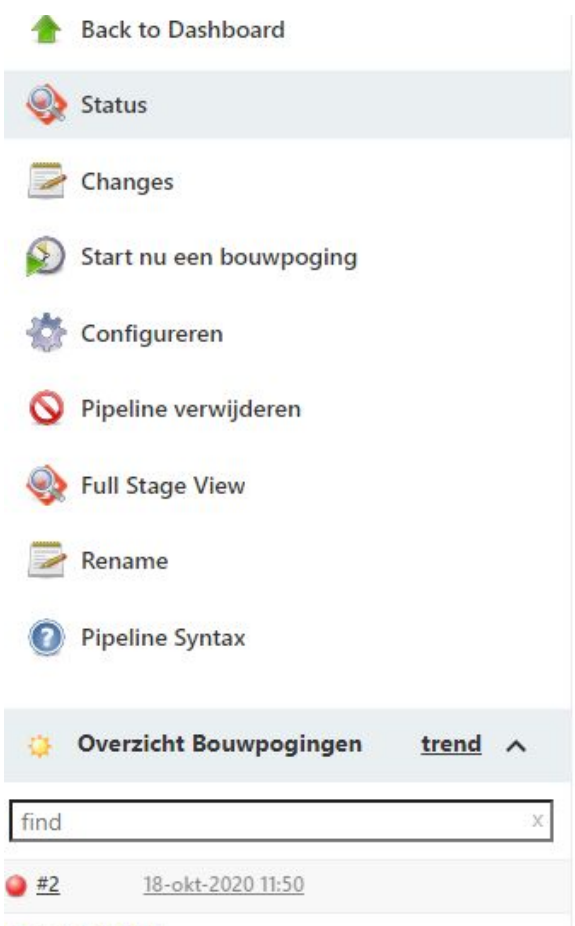
THE FIRST WAY

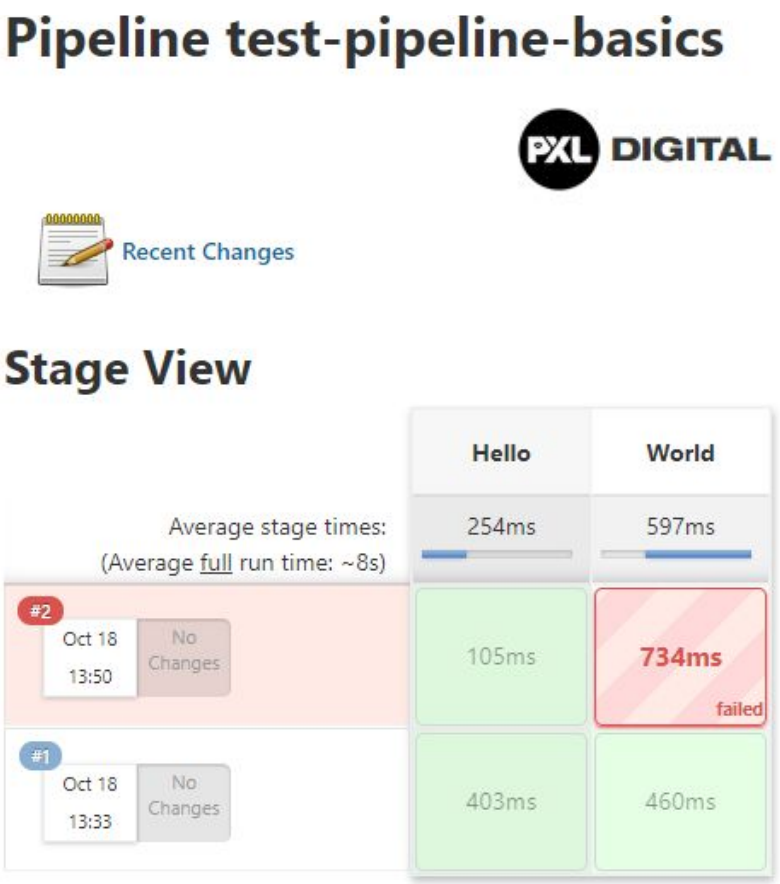
CREATE A DELIVERY SYSTEM

PXL DIGITAL

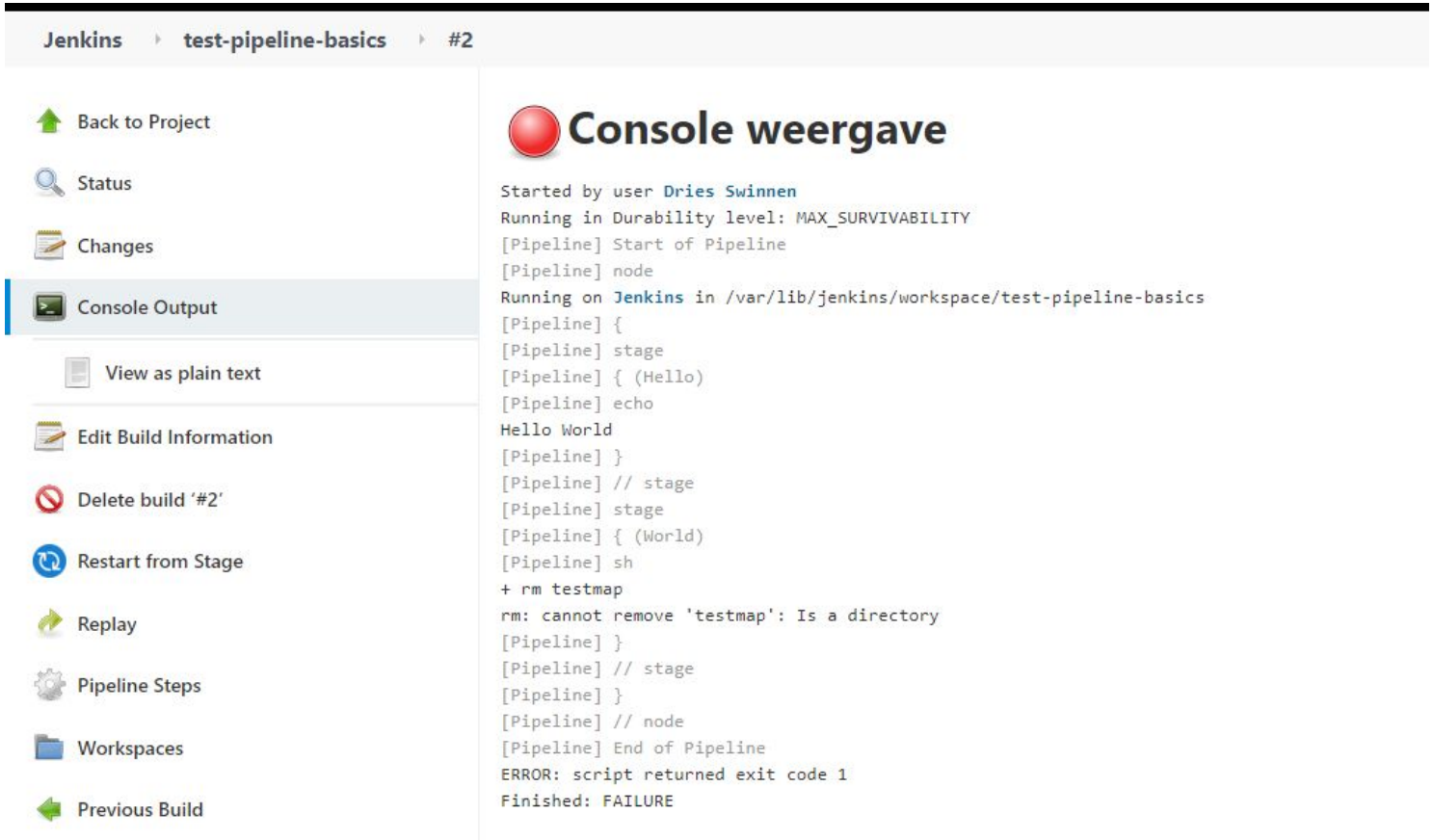


Recap


Back to Dashboard
Status
Changes
Start nu een bouwpoging
Configureren
Pipeline verwijderen
Full Stage View
Rename
Pipeline Syntax
Overzicht Bouwpogingen
trend
find
#2
18-okt-2020 11:50


Pipeline test-pipeline-basics
PXL DIGITAL
Recent Changes
Stage View
Hello
World
Average stage times:
(Average full run time: ~8s)
254ms
597ms
#2
Oct 18
13:50
No Changes
105ms
734ms
failed
#1
Oct 18
13:33
No Changes
403ms
460ms

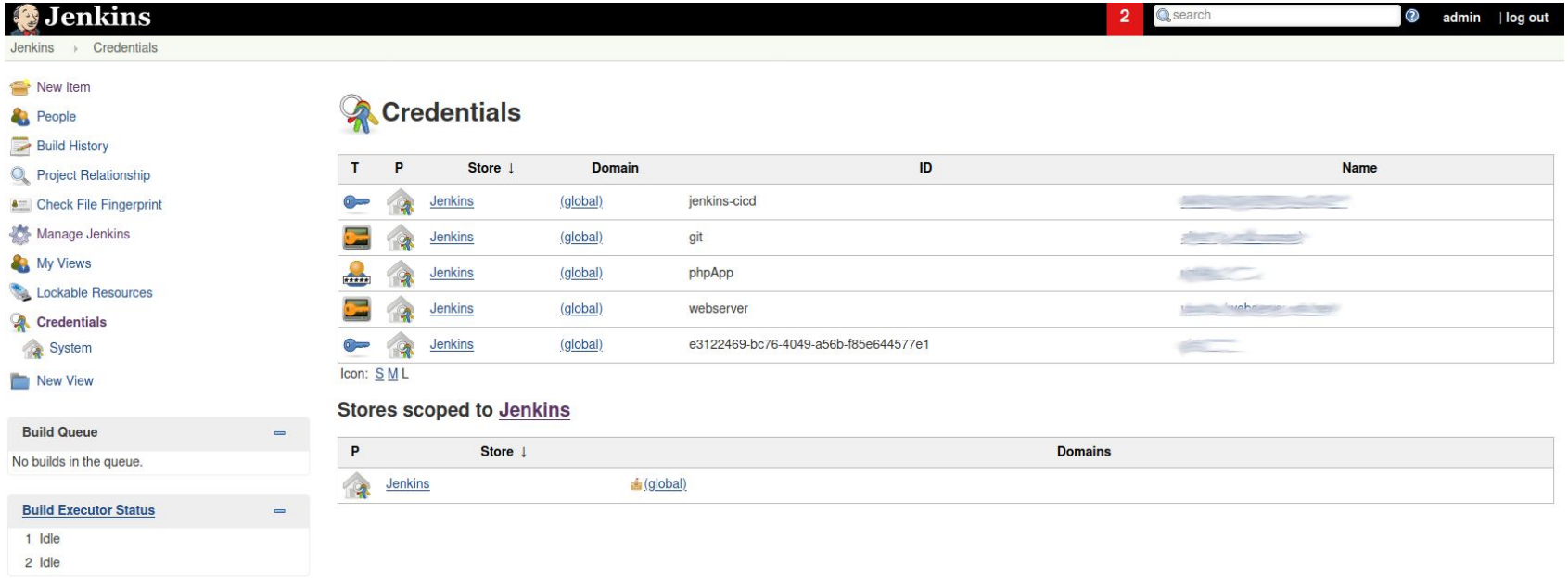
Recap


Jenkins
test-pipeline-basics
#2
Back to Project
Status
Changes
Console Output
View as plain text
Edit Build Information
Delete build '#2'
Restart from Stage
Replay
Pipeline Steps
Workspaces
Previous Build
Console weergave
Started by user Dries Swinnen
Running in Durability level: MAX_SURVIVABILITY
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/test-pipeline-basics
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Hello)
[Pipeline] echo
Hello World
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (World)
[Pipeline] sh
+ rm testmap
rm: cannot remove 'testmap': Is a directory
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
ERROR: script returned exit code 1
Finished: FAILURE

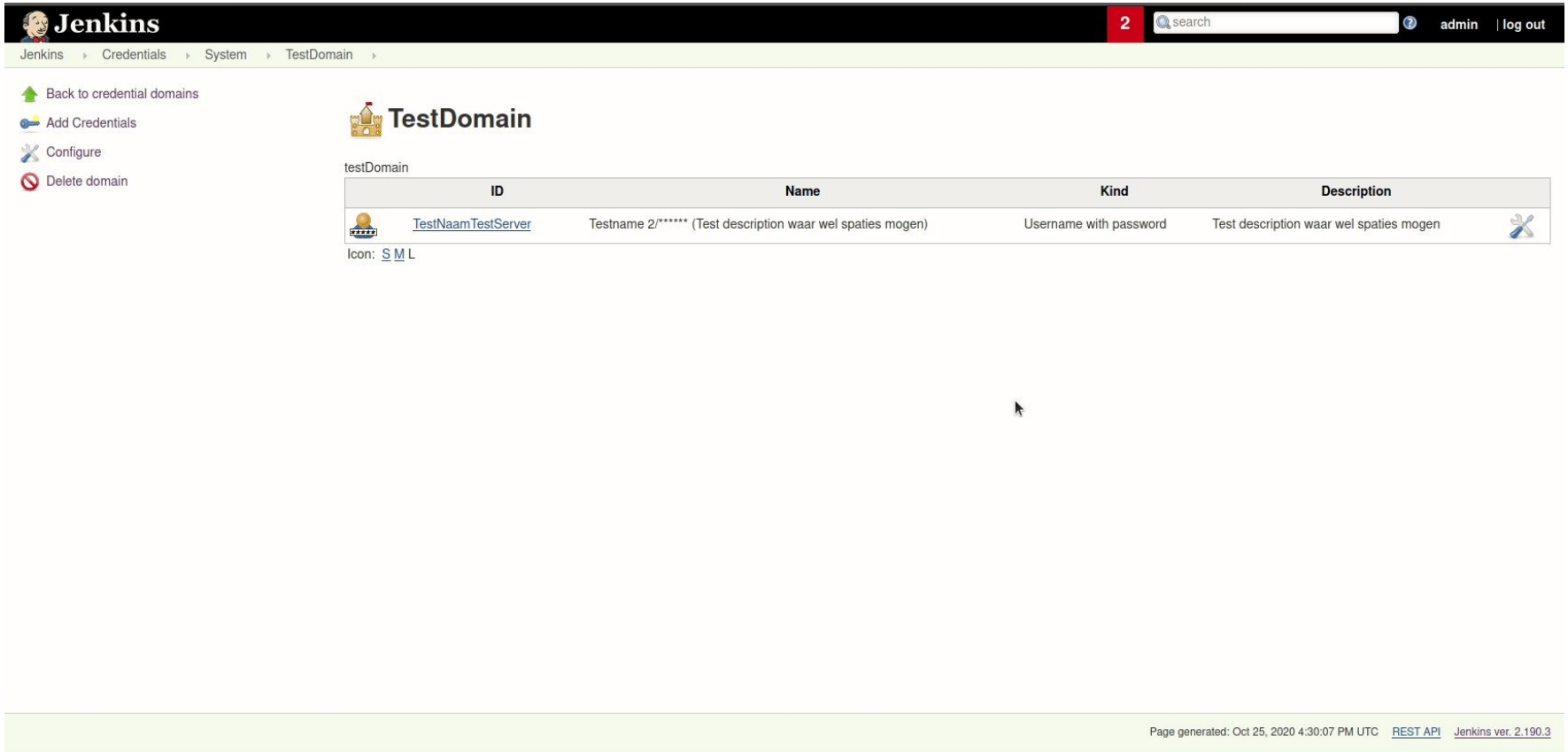
PXL DIGITAL

Jenkins - pipeline - credential manager

credentials in een veilige plaats opslaan:



Jenkins - pipeline - credential manager



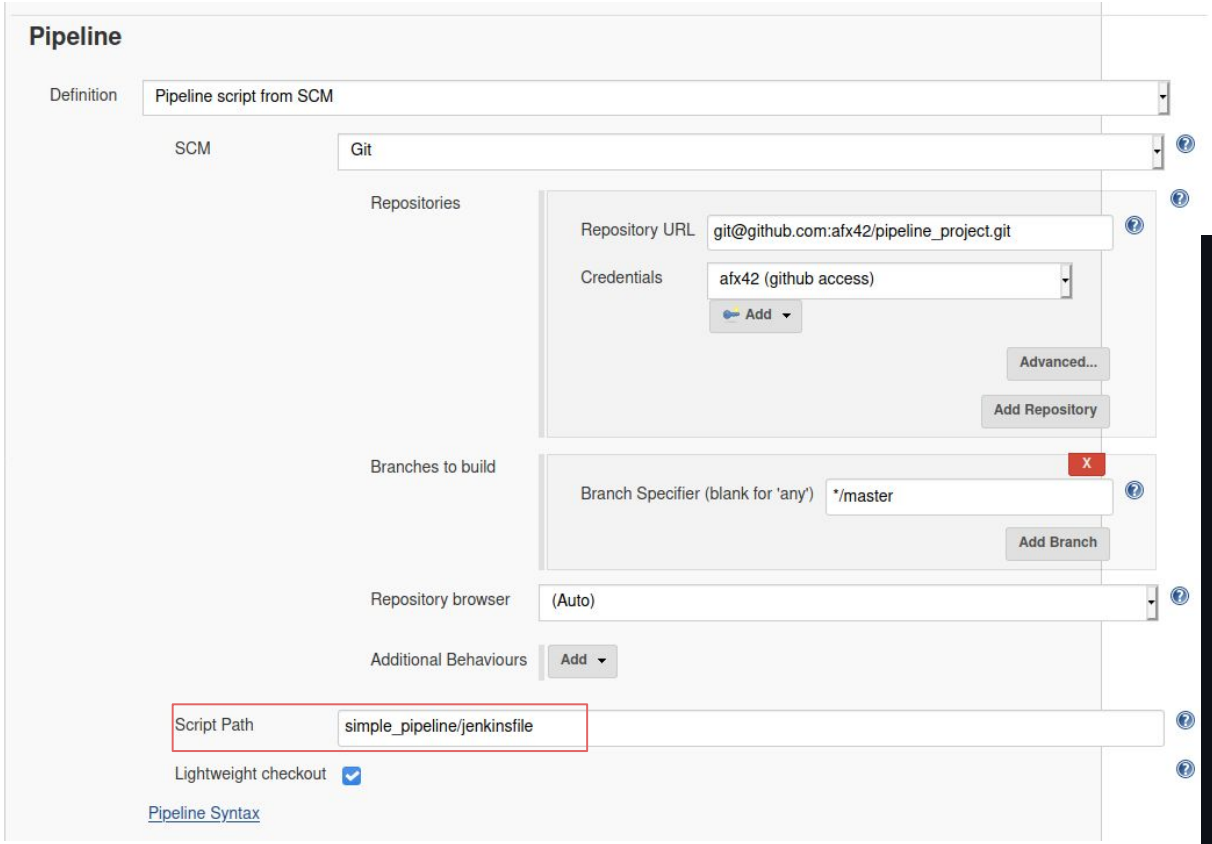
Jenkins - pipeline - credential manager

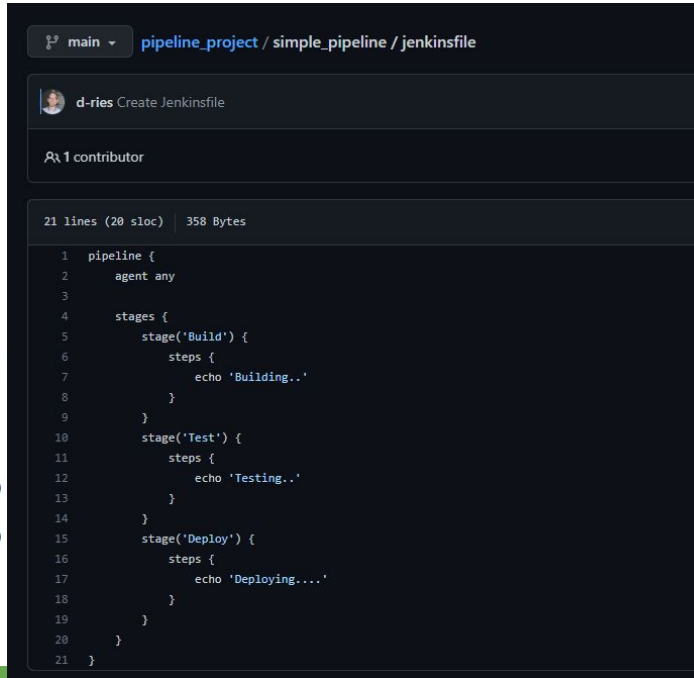
Aanroepen credentials in jenkinsfile:

```
stage('Deploy') {
  // A sleep to make sure we actually get a real difference!
  sleep (30)
sshagent(credentials : ['webserver']) {

    sh "scp -o StrictHostKeyChecking=no CurrentBuild.zip ubuntu@$instanceInfo$remotedir"
    sh "ssh -T -o StrictHostKeyChecking=no ubuntu@$instanceInfo sudo unzip CurrentBuild.zip -d /var/www/html"
      }
```

```
[Pipeline] sshagent
[ssh-agent] Using credentials ubuntu (webserver ssh key)
[ssh-agent] Looking for ssh-agent implementation...
[ssh-agent]   Exec ssh-agent (binary ssh-agent on a remote machine)
$ ssh-agent
SSH_AUTH_SOCK=/tmp/ssh-UAasre7XkOXS/agent.3227
SSH_AGENT_PID=3230
Running ssh-add (command line suppressed)
Identity added: /var/jenkins_home/workspace/DevOps_Calculator@tmp/private_key_8600551491206348113.key (/var/jenkins_home/workspace/DevOps_Calculator@tmp
/private_key_8600551491206348113.key)
[ssh-agent] Started.
[Pipeline] {
[Pipeline] readFile
[Pipeline] echo
172.31.36.68
[Pipeline] sh
+ scp -o StrictHostKeyChecking=no CurrentBuild.zip ubuntu@172.31.36.68:.
Warning: Permanently added '172.31.36.68' (ECDSA) to the list of known hosts.
```

Jenkins - pipeline - jenkinsfile





Jenkins - pipeline - Compiling code

- Java -> Maven

Script

```
1  pipeline {
2      agent any
3      stages {
4          stage('Build') {
5              steps {
6                  git 'https://github.com/jglick/simple-maven-project-with-tests.git'
7
8                  sh "mvn clean package"
9              }
```

Stage Logs (Build)

Git -- https://github.com/jglick/simple-maven-project-with-tests.git (self time 407ms)

Shell Script -- mvn -Dmaven.test.failure.ignore=true clean package (self time 461ms)

```
+ mvn -Dmaven.test.failure.ignore=true clean package
/var/lib/jenkins/workspace/magic-unicorn-bicycle@tmp/durable-c2a01eab/script.sh: 1: mvn: not found
```

Jenkins - pipeline - Compiling code

- Tools die gebruikt worden in de pipeline -> installeren op agents
  - Agent any = Jenkins kiest zelf; huidige jenkins server

```
$ cd /opt
$ wget https://apache.belnet.be/maven/maven-3/3.6.3/binaries/apache-maven-3.6.3-bin.tar.gz
$ sudo tar -xvzf ~/Downloads/apache-maven-3.6.3-bin.tar.gz
$ PATH="...:/opt/apache-maven-3.6.3/bin"
$ mvn --version
```

- Handmatig?

Jenkins - pipeline - Global tool configuration

- Java -> Maven

Pipeline

Definition: Pipeline script

Script:

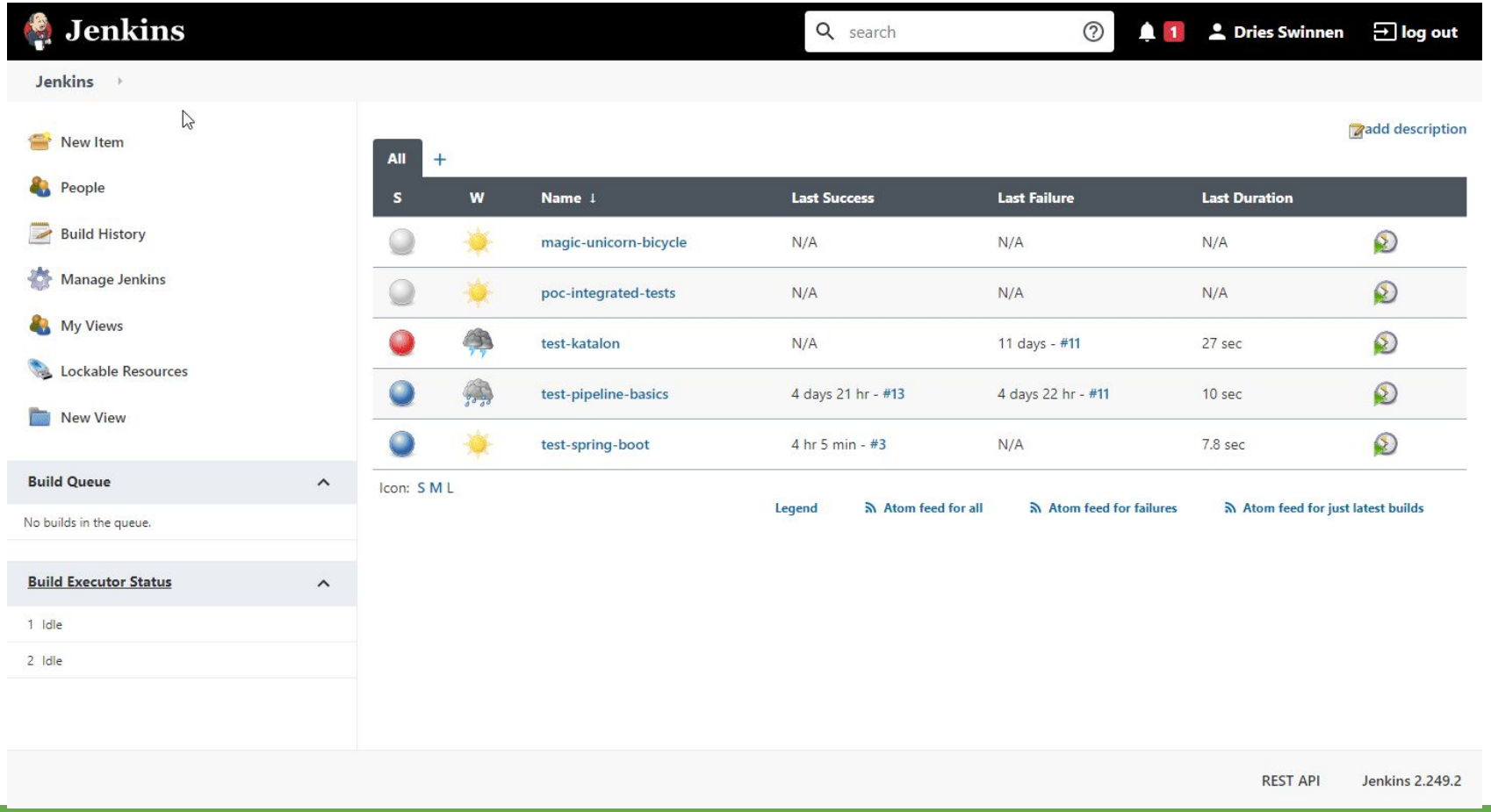
```
pipeline {
    agent any

    tools {
        // Install the Maven version configured as "M3" and add it to the path.
        maven "M3"
    }

    stages {
        stage('Build') {
            steps {
                // Get some code from a GitHub repository
                git 'https://github.com/jglick/simple-maven-project-with-tests.git'

                // Run Maven on a Unix agent.
                sh "mvn -Dmaven.test.failure.ignore=true clean package"
```

GitHub + Maven

Use Groovy Sandbox

Pipeline Syntax

Jenkins - pipeline - Global tool configuration



Jenkins - pipeline - Global tool configuration

- Tools globaal definiëren (maven / Git / JDK / NPM / …)
  - definitie in pipeline => Werkt ook altijd op de agent
- Niet beschikbaar via gobal tool configuration?
  - Plugin?
  - Handmatige installatie op server / agents

Jenkins - pipeline - Compiling code / Installing dependencies

- React(of Angular) / NodeJS/… - NPM

Pipeline

Definition: Pipeline script

Script

```
pipeline {
    agent any

    stages {
        stage('Code pull') {
            steps {
                // Get some code from a GitHub repository
                git branch: 'jenkins2-course',
            url: 'https://github.com/jenkins-docs/simple-node-js-react-npm-app.git'
            }
        }
        stage('build') {
            steps {
            nodejs('npm') {
                echo "Performing npm build..."
                sh 'npm install'
                    }
            }
        }
```

Jenkins - pipeline - Unit testing

- hangt af van testrunner applicatie
  - Java => JUnit
  - NodeJS => Jest
  - Vaak integratie vanuit andere talen naar JUnit

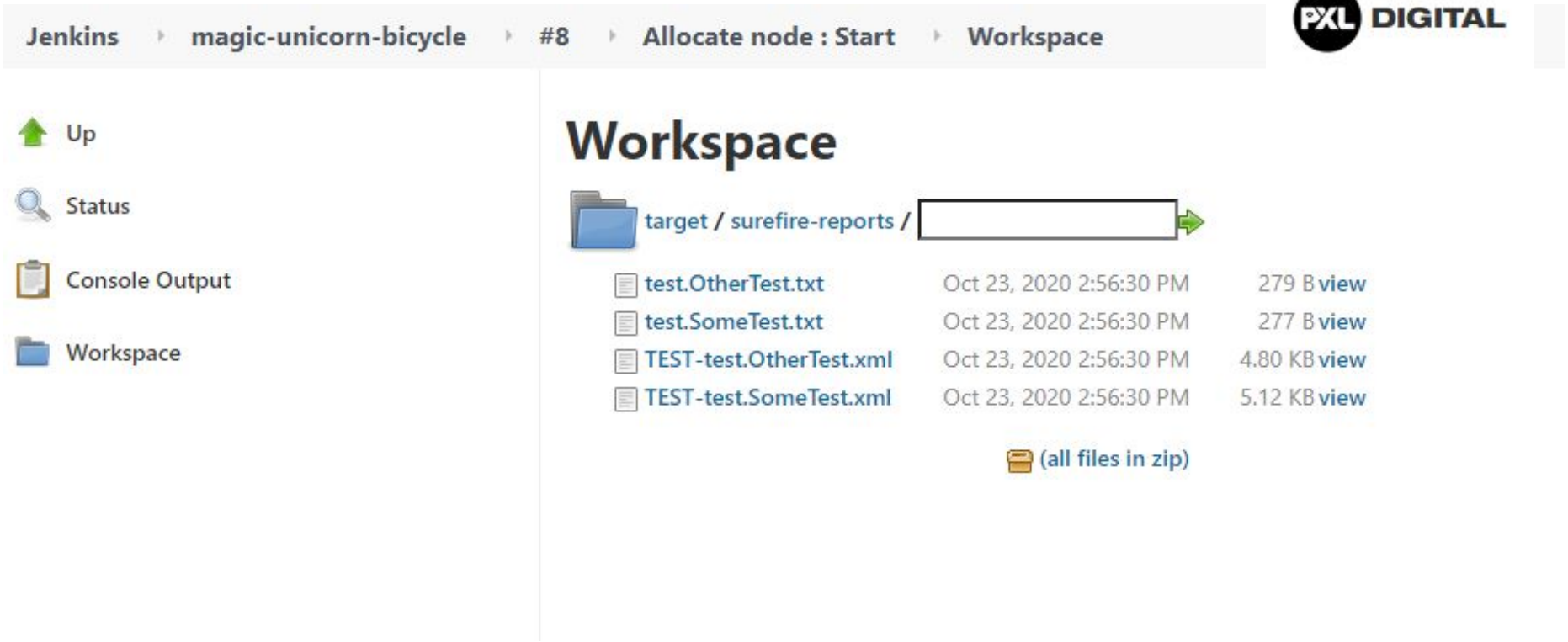
```
stage('unittest'){
    steps {
        sh "mvn test"
    }
}
```

Jenkins - pipeline - Unit testing

```
[Pipeline] sh
+ mvn test
[INFO] Scanning for projects...
[INFO]
[INFO] ----------------< test:simple-maven-project-with-tests >----------------
[INFO] Building simple-maven-project-with-tests 1.0-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ simple-maven-project-with-tests ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /var/lib/jenkins/workspace/magic-unicorn-bicycle/src/main/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ simple-maven-project-with-tests ---
[INFO] No sources to compile
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ simple-maven-project-with-tests ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /var/lib/jenkins/workspace/magic-unicorn-bicycle/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ simple-maven-project-with-tests ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.18.1:test (default-test) @ simple-maven-project-with-tests ---
[INFO] Surefire report directory: /var/lib/jenkins/workspace/magic-unicorn-bicycle/target/surefire-reports

-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running test.OtherTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.012 sec - in test.OtherTest
```

Jenkins - pipeline - Unit testing - Rapporten

PXL DIGITAL

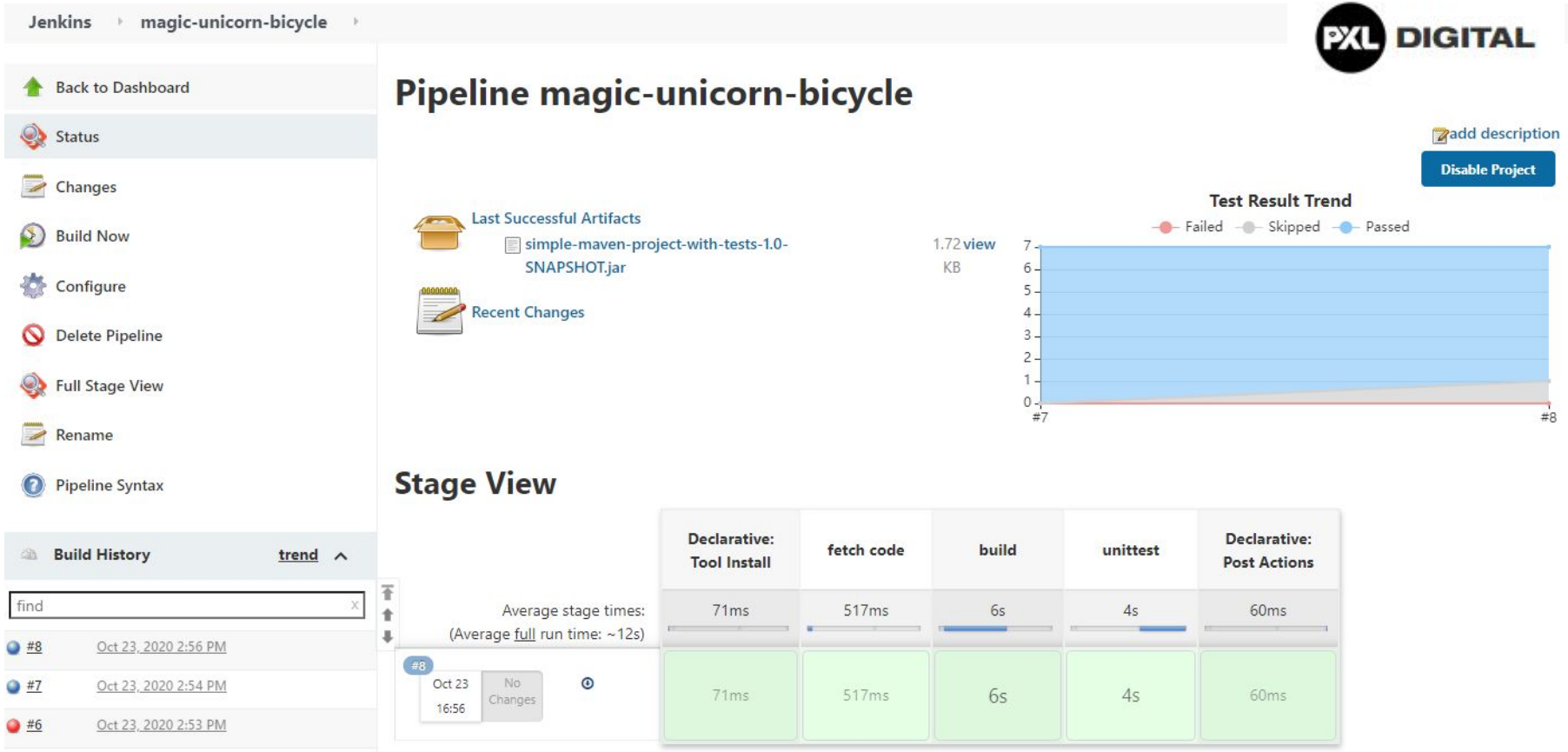


Jenkins - pipeline - Unit testing - Rapporten

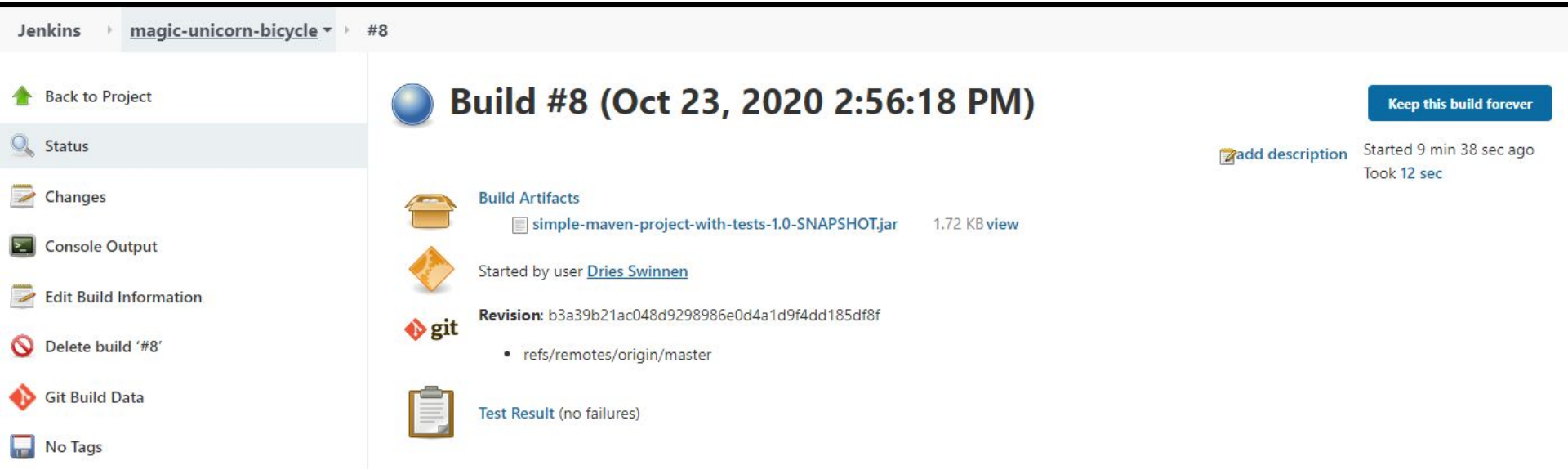
PXL DIGITAL

```
stage('unittest'){
    steps {
        sh "mvn test"
        junit '**/target/surefire-reports/TEST-*.xml'
    }
}
```

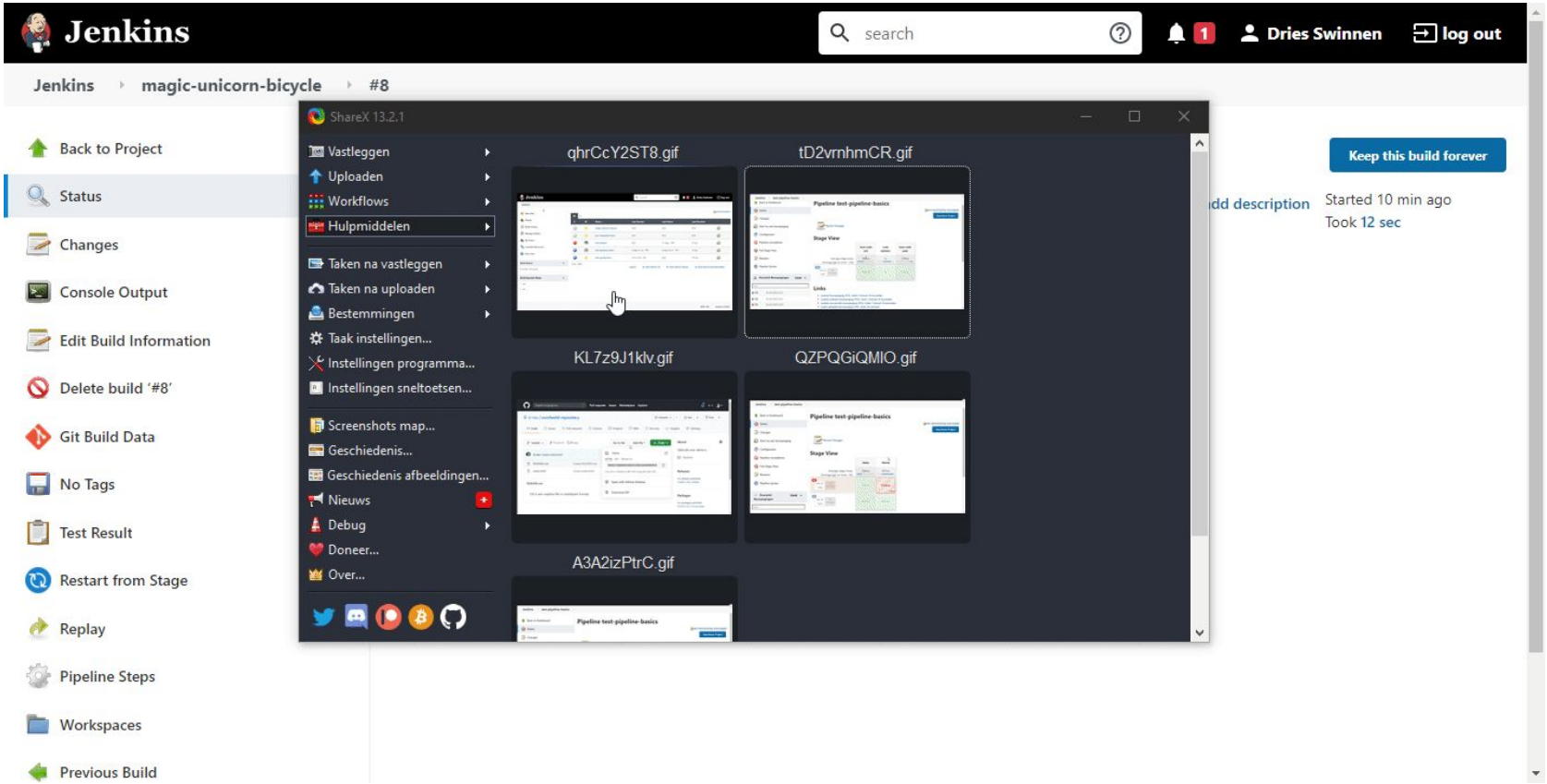
Jenkins - pipeline - Unit testing - Rapporten



Jenkins - pipeline - Unit testing - Rapporten



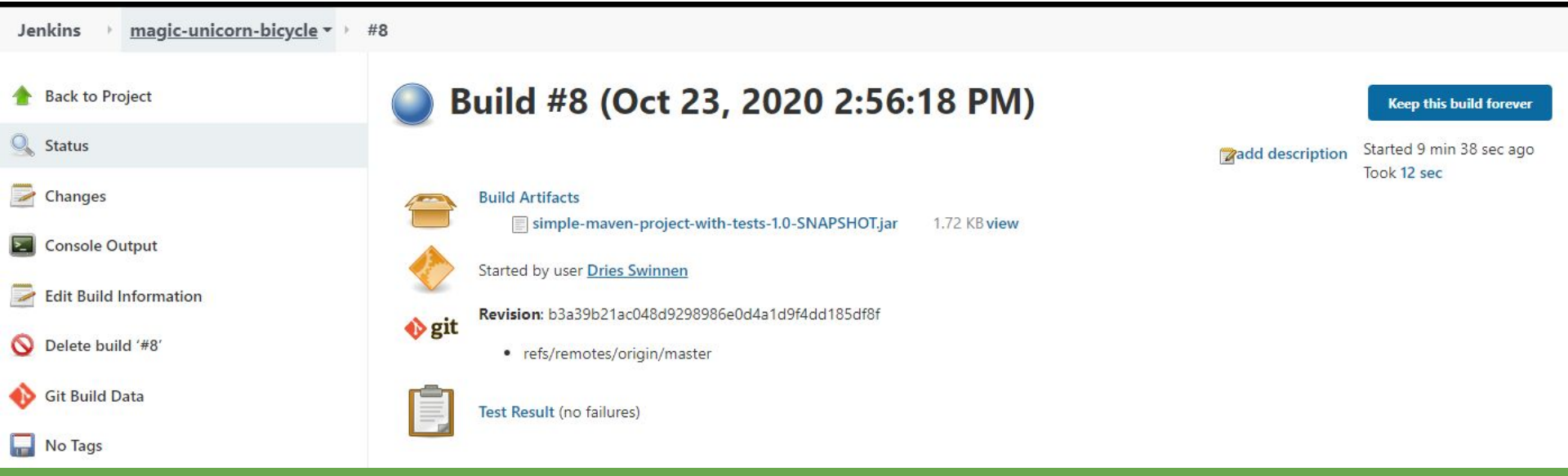
Jenkins - pipeline - Unit testing - Rapporten



Jenkins - pipeline - Artifacts

In jenkins - java applicatie

```
post {
        success {
            archiveArtifacts 'target/*.jar'
        }
 }
```

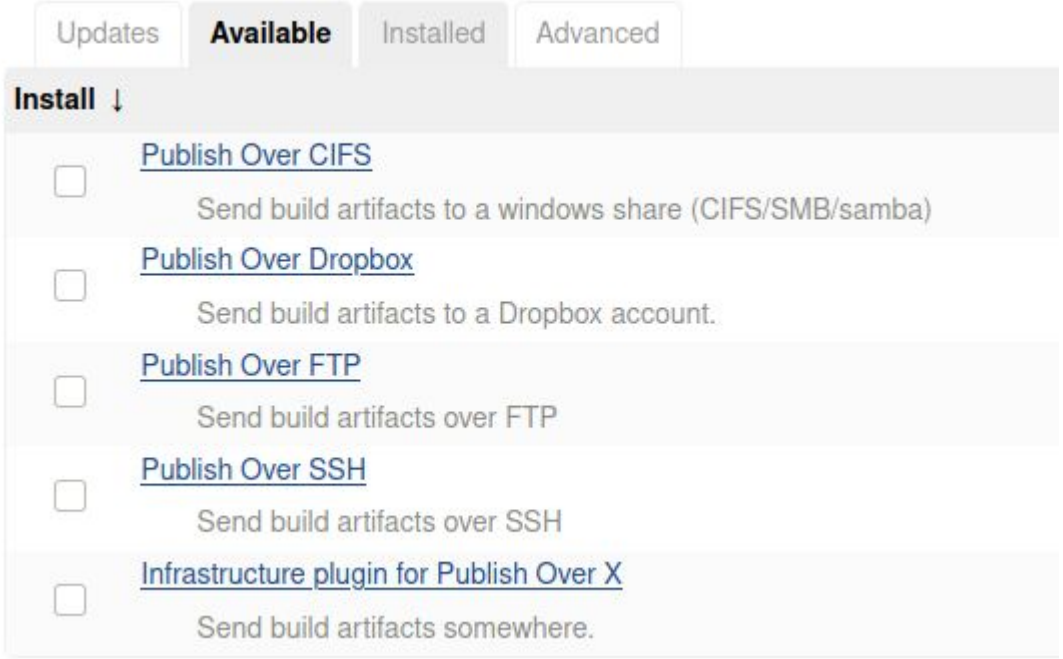


Jenkins - pipeline - Artifacts

PXL DIGITAL

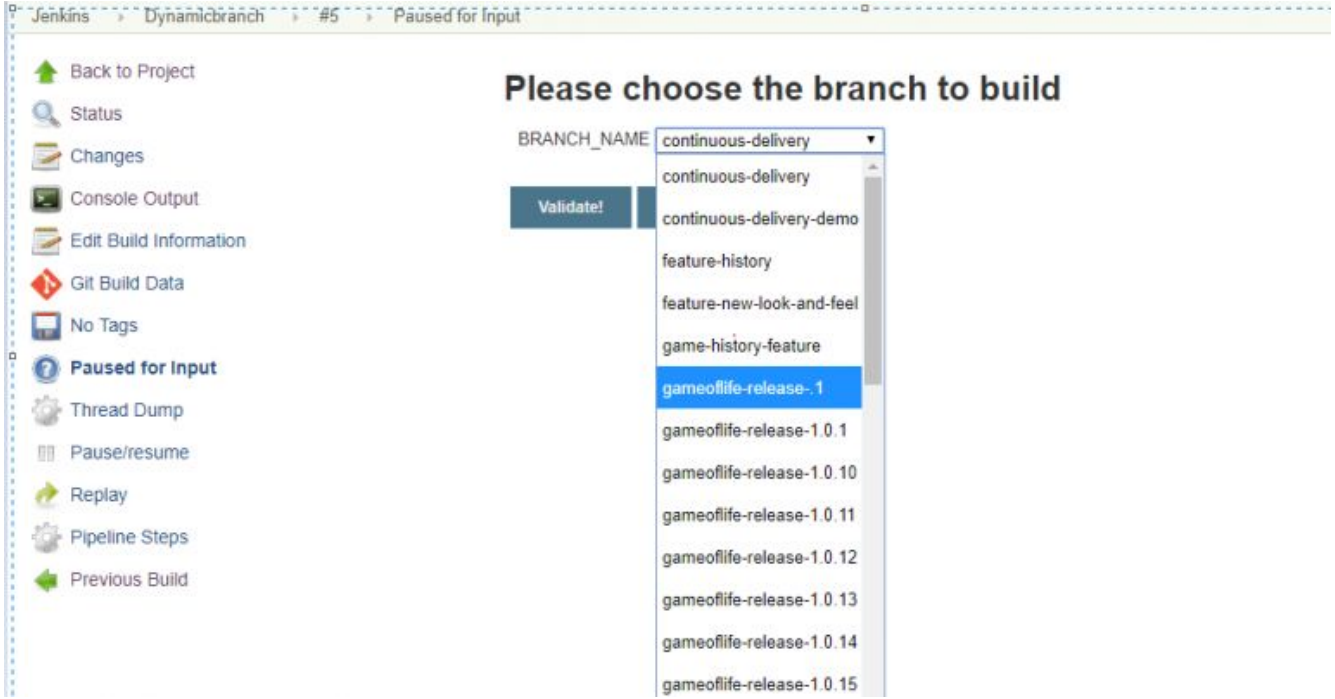
Extern

Via plugins:



Jenkins - pipeline - Input step

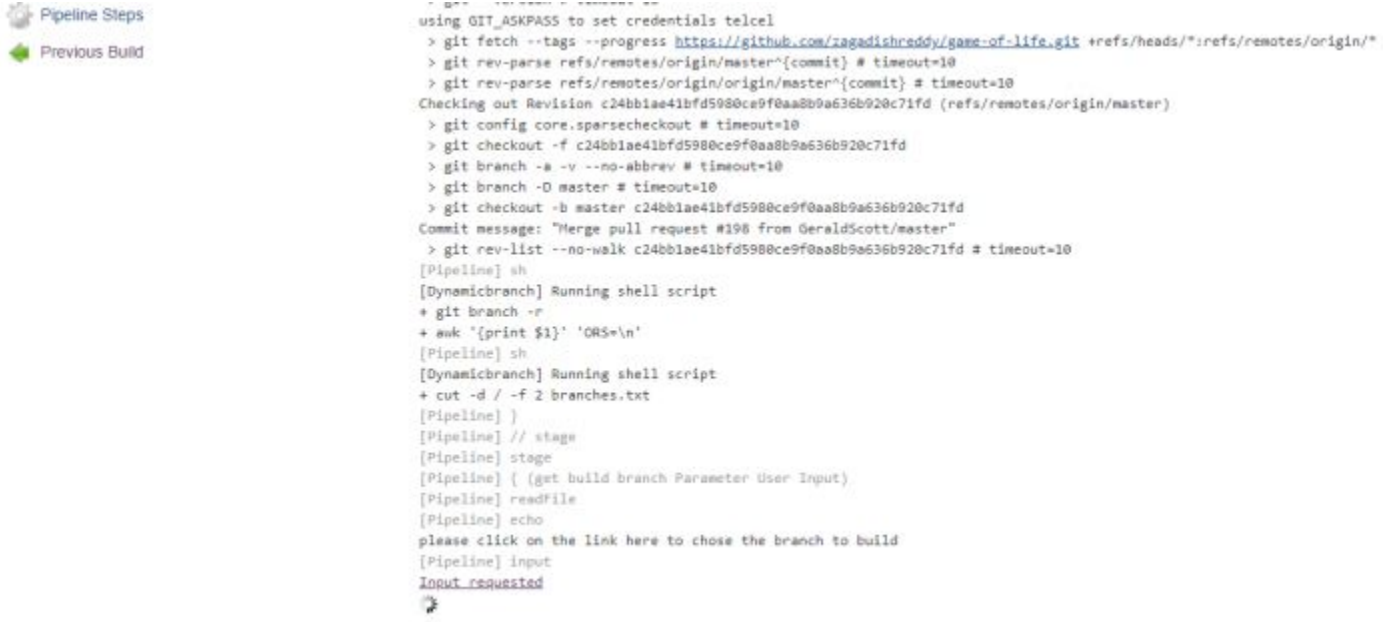
Interactieve steps: Vraag welke git branch gebuild moet worden:



https://labs.tadigital.com/index.php/2018/07/18/list-the-git-branches-dynamically-in-jenkins-pipeline-job/

Jenkins - pipeline - Input step

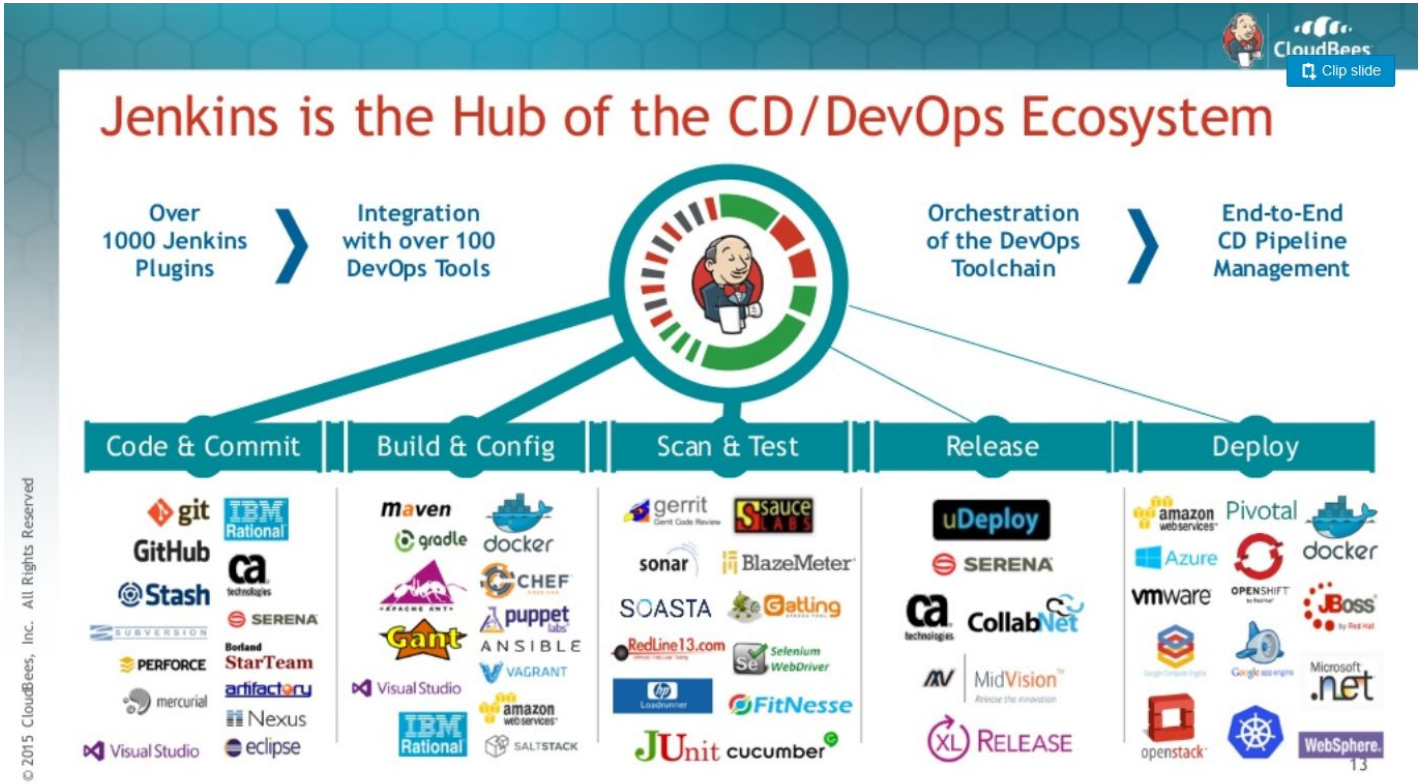
Interactieve steps: Vraag welke git branch gebuild moet worden:



https://labs.tadigital.com/index.php/2018/07/18/list-the-git-branches-dynamically-in-jenkins-pipeline-job/

Recap

PXL DIGITAL



© 2015 CloudBees, Inc. All Rights Reserved

Assignments

Jenkins - CI

HOGESCHOOL PXL