

DevOps 2TIN

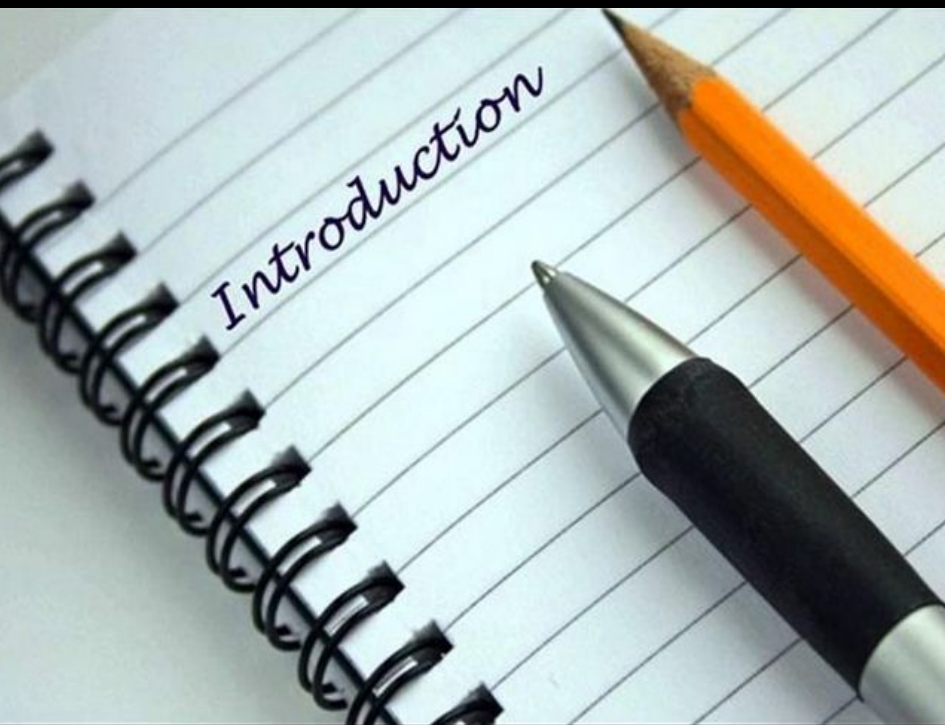
Chapter 5

Pipelines in jenkins



**DE HOGESCHOOL
MET HET NETWERK**

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



Pipelines

Recap
Jenkins workspace
Pipeline scripts
Groovy
Stages & steps
Commands
Plugins

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*)

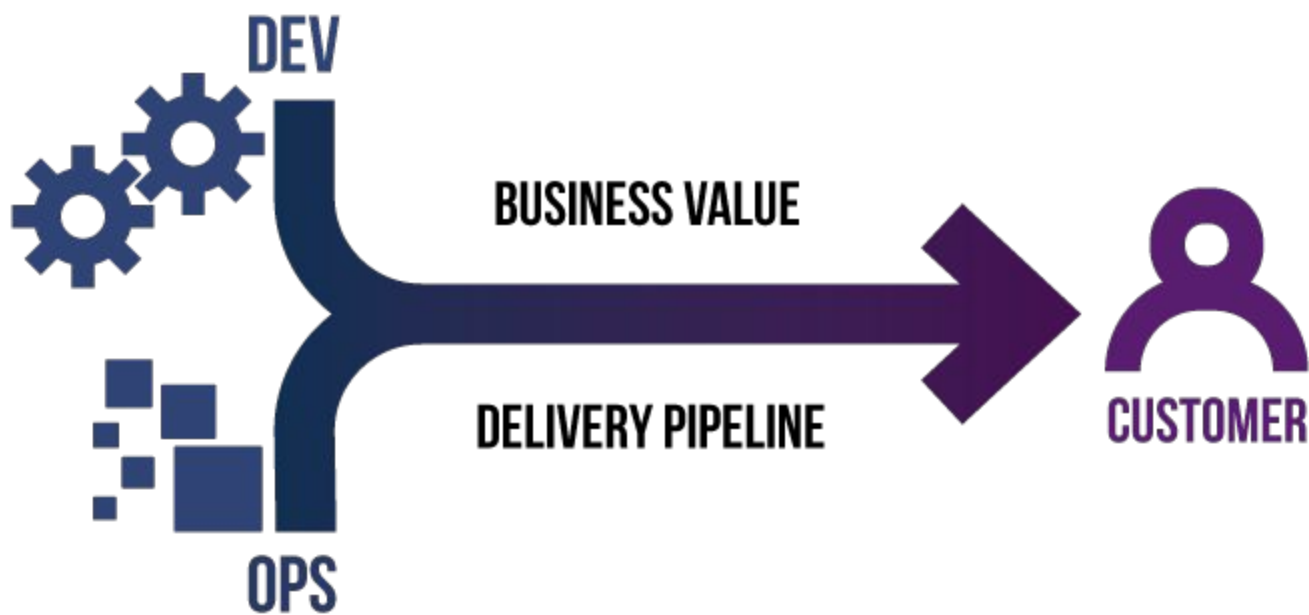
It ensures that all code checked in to version control is automatically built and tested in a production-like environment.

Sleutelwoorden:

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

THE FIRST WAY

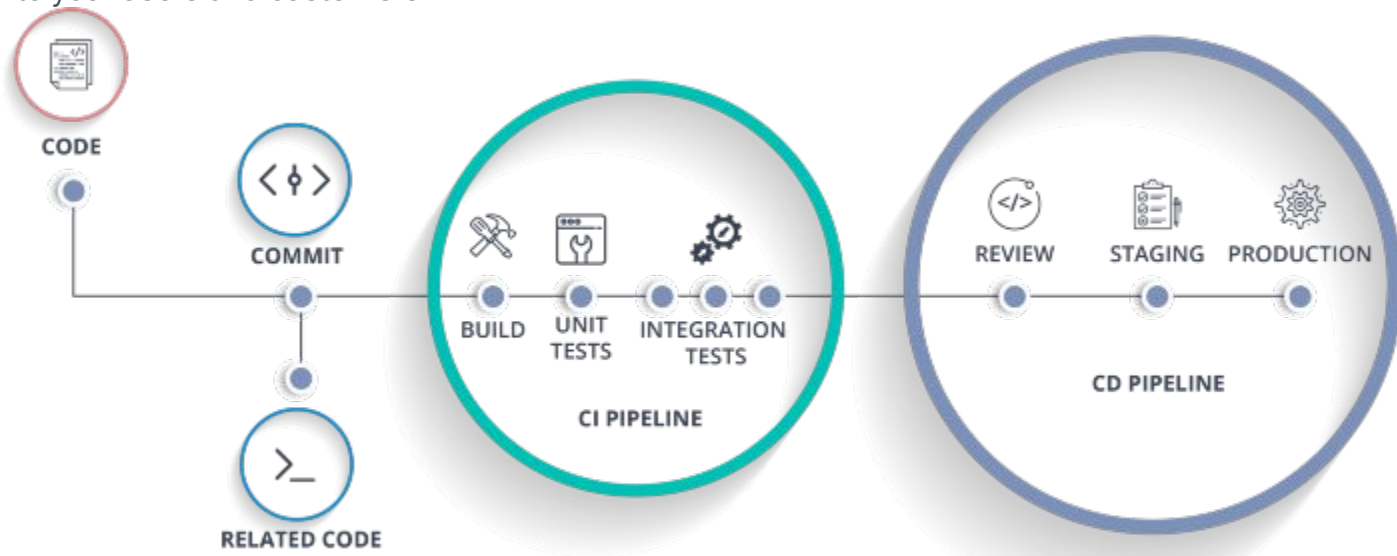
CREATE A DELIVERY SYSTEM



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.

Jenkins items



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.



Pipeline

Orchestrates long-running activities that can span multiple build agents. 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.

<https://app.pluralsight.com/course-player?clipId=c5537166-12a3-48fa-9cfd-67f2024cf366>

Jenkins - Pipeline



Freestyle project vs pipeline

Build

Conditional step (single) X ?

Run? Strings match ?

String 1

String 2

Case insensitive ☒ ?

[Advanced...](#)

Builder

Command

[See the list of available environment variables](#)

[Advanced...](#)

[Add build step](#)

job

General Build Triggers Advanced Project Options **Pipeline**

Pipeline

Definition

Script


```
1 node {
2   stage 'Build Application'
3   // do what you need to build your application
4   echo 'Compilation is done'
5   stage 'Deploy Application'
6   // do what you need to deploy your application
7   echo 'Deploy is done'
8   stage 'Execute JMeter Performance Tests'
9   build job: 'JMeter - Freestyle'
10 }
```



☒ Use Groovy Sandbox

[Pipeline Syntax](#)

[Save](#) [Apply](#)

Jenkins - Pipeline as code

 **Jenkins**


 **dries**  **log out**

Jenkins > **All** >

Enter an item name


test-pipeline-basics

» 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.




Pipeline

Orchestrates long-running activities that can span multiple build agents. 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.



Get Team/Project

Scans a Bitbucket Cloud Team (or Bitbucket Server Project) for all repositories matching some defined markers.

Jenkins - Pipeline - stage view



Jenkins > test-pipeline-basics >

Back to Dashboard

Status

Changes

Start nu een bouwpoing

Configureren

Pipeline verwijderen

Full Stage View

Rename

Pipeline Syntax

Overzicht Bouwpoingen trend ^

#5 18-okt-2020 12:12

#4 18-okt-2020 12:12

#3 18-okt-2020 12:12

#2 18-okt-2020 11:50

Pipeline test-pipeline-basics

Recent Changes

Stage View

	Hello	do something	more stuff
Average stage times: (Average <u>full</u> run time: ~2s)	87ms	358ms	625ms
#5 Oct 18 14:12 No Changes	80ms	350ms	607ms
#4 Oct 18 14:12 No Changes	75ms	367ms	623ms
#3 Oct 18 14:12 No Changes	108ms	359ms	647ms

Jenkins - Pipeline - stage view



Handmatig starten van het pipeline script

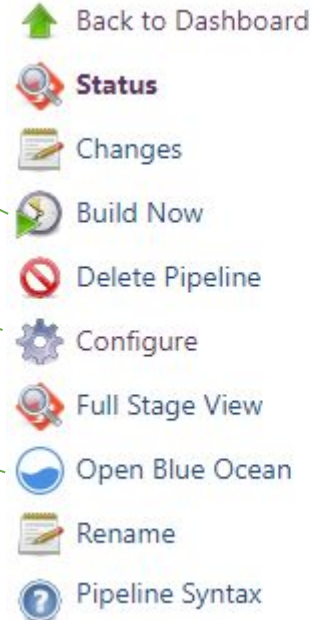
- Maakt een nieuwe build poging (#)

Configureren van het pipeline script


- Jenkins as code!


GUI schil over een pipeline script



- Pas beschikbaar na installatie plugin "blue ocean"




Jenkins - Pipeline - Hello world


 **Jenkins**


search 


 **dries**  **log out**


Jenkins >


 New Item


 People


 Build History


 Manage Jenkins


 My Views









 Open Blue Ocean

 Lockable Resources




 New View


 [add description](#)

All 


S	W	Name ↓	Last Success	Last Failure	Last Duration	Fav
		test	N/A	N/A	N/A	 
		test-pipeline-basics ▼	N/A	N/A	N/A	 

Icon: [S](#) [M](#) [L](#)

[Legend](#)  [Atom feed for all](#)  [Atom feed for failures](#)  [Atom feed for just latest builds](#)

Build Queue 

No builds in the queue.

Build Executor Status 

1 Idle

2 Idle

3.93.172.23:8080/job/test-pipeline-basics/

Page generated:
Sep 5, 2020, 10:22:11 AM UTC

REST API

Jenkins 2.235.5

Jenkins - Build history & details

Algemene status van pipeline
- op basis van laatste 4 pogingen

Build history van alle pogingen

Status specifieke build poging
- Link naar detail-view
build poging



The screenshot shows the Jenkins 'Build History' widget. At the top, there is a cloud icon, the title 'Build History', and a 'trend' link with a dropdown arrow. Below this is a search bar containing the text 'find' and a close button 'x'. The main content is a list of three builds:

Status Icon	Build Number	Timestamp
Red circle with white dot	#3	Sep 5, 2020, 10:25 AM
Red circle with white dot	#2	Sep 5, 2020, 10:24 AM
Blue circle with white dot	#1	Sep 5, 2020, 10:22 AM

At the bottom of the widget, there are two links: 'Atom feed for all' and 'Atom feed for failures', each preceded by an RSS icon.

Jenkins - Build history & details



Jenkins > test-pipeline-basics > #3

- Back to Project
- Status**
- Changes
- Console Output
- Edit Build Information
- Delete build '#3'
- Open Blue Ocean
- Restart from Stage
- Replay
- Pipeline Steps
- Workspaces
- Previous Build



Build #3 (Sep 5, 2020, 10:25:28 AM)



Started by user [dries](#)

[Keep this build forever](#)

[add description](#)

Started 3 min 53 sec ago

Took [1.5 sec](#)

Jenkins - Build history & details



Console output

- Debugging pipeline
- Geeft output van elke stap

Aanvullende informatie voorzien voor specifieke build

Detailweergave build per pipeline step

- Console output per step

Handmatig bekijken bestanden workspace specifieke pipeline

Jenkins > test-pipeline-

- Back to Project
- Status**
- Changes
- Console Output
- Edit Build Information
- Delete build '#3'
- Open Blue Ocean
- Restart from Stage
- Replay
- Pipeline Steps
- Workspaces
- Previous Build

Jenkins - pipeline - debugging a pipeline



Jenkins ▾ ▸ experiment ▸ phoenix-jenkins-test ▸ #107

Back to Project

Status

Changes

Console Output

View as plain text

Edit Build Information

Delete Build

Git Build Data

No Tags

Replay


Pipeline Steps

Previous Build

Console Output

```
Started by user Trang, Anh: GIS \(SGP\)
Loading library phoenix/phoenix-sharedpipeline@master
Setting origin to ssh://git@ibgit.barcapint.com:7999/ib_phoenix/phoenix-sharedpipeline-devops.git
Fetching origin...
Fetching changes from the remote Git repository
Checking out Revision e314bff36216ef83e575d8febcd43cdab7b28733 (master)
[Pipeline] stage
[Pipeline] { (clone)
[Pipeline] node
Running on master in /app/phoenix/jenkins2home/workspace/phoenix-jenkins-test
[Pipeline] {
[Pipeline] httpRequest
HttpMethod: GET
URL: http://ldndsr000004893:7761/eureka/apps/phoenix-access360-ui
Accept: application/json
Using authentication: f0740abf-7191-4ed8-a8ee-54aa7aed112b
Sending request to url: http://ldndsr000004893:7761/eureka/apps/phoenix-access360-ui
Response Code: HTTP/1.1 200
Success code from [0?1000]
[Pipeline] echo
PhoenixEurekaInstance{serviceName='phoenix-access360-ui', hostname='ldndsr000004893.intranet.barcapint.com', port='12000', tagName='null'}
[Pipeline] echo
PhoenixEurekaInstance{serviceName='phoenix-access360-ui', hostname='ldndsr000004893.intranet.barcapint.com', port='12001', tagName='null'}
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] // stage
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Jenkins - Pipeline errors

 [Back to Dashboard](#)



Status



Changes



Start nu een bouwpoing



Configureren



Pipeline verwijderen



Full Stage View



Rename



Pipeline Syntax



Overzicht Bouwpogingen

[trend](#) ^

x



#2

18-okt-2020 11:50

Pipeline test-pipeline-basics



[Recent Changes](#)

Stage View

Average stage times:
(Average full run time: ~8s)

#2

Oct 18
13:50

No
Changes

#1

Oct 18
13:33

No
Changes

Hello

World

254ms

597ms

105ms

734ms

failed

403ms

460ms

Jenkins - Pipeline errors



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

Jenkins - Pipeline errors

Jenkins

test-pipeline-basics

Back to Dashboard

Status

Changes

Start nu een bouwopgoging

Configureren

Pipeline verwijderen

Full Stage View

Rename

Pipeline Syntax

Overzicht

Bouwopgogingen

find

Pipeline test-pipeline-basics

Recent Changes

Stage View

Average stage times:
(Average full run time: ~8s)

	Hello	World
#2	105ms	734ms failed
#1	403ms	460ms

Deactiveer Project

Jenkins - Minimaal pipeline script

Pipeline

Definition

Pipeline script

Script

```
1 pipeline {  
2     agent any  
3  
4     stages {  
5         stage('Hello') {  
6             steps {  
7                 echo 'Hello World'  
8             }  
9         }  
10    }  
11 }  
12
```

Hello World

☒ Use Groovy Sandbox

[Pipeline Syntax](#)

Jenkins - Groovy



Flat learning curve

Concise, readable and expressive syntax, easy to learn for Java developers



Powerful features

Closures, builders, runtime & compile-time meta-programming, functional programming, type inference, and static compilation



Smooth Java integration

Seamlessly and transparently integrates and interoperates with Java and any third-party libraries



Domain-Specific Languages

Flexible & malleable syntax, advanced integration & customization mechanisms, to integrate readable business rules in your applications



Vibrant and rich ecosystem

Web development, reactive applications, concurrency / asynchronous / parallelism library, test frameworks, build tools, code analysis, GUI building




Scripting and testing glue




Great for writing concise and maintainable tests, and for all your build and automation tasks

<https://www.jenkins.io/doc/book/pipeline/syntax/>


<https://groovy-lang.org/syntax.html>


Jenkins - Helper & docs


 **Jenkins**


  **dries**  **log out**


Jenkins > test-pipeline-basics > Pipeline Syntax


 Back


 **Snippet Generator**


 Declarative Directive Generator


 Declarative Online Documentation

 Steps Reference

 Global Variables Reference

 Online Documentation

 Examples Reference

 IntelliJ IDEA GDSL

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

Files to archive

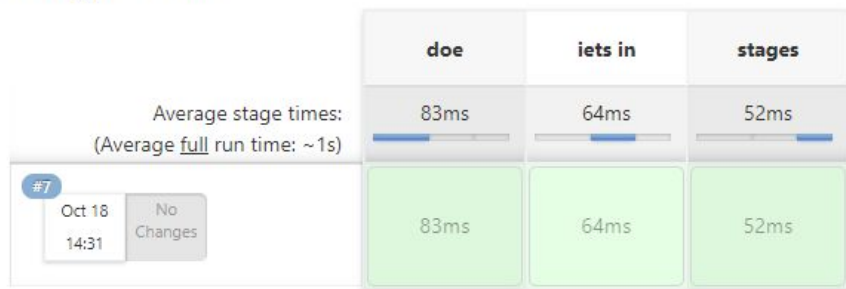
Advanced...

Generate Pipeline Script

Jenkins - pipeline - stages

- Stages geven structuur aan de pipeline
- Worden opgenomen als kolom in de stage view

Stage View



```
1 pipeline {
2     agent any
3
4     stages {
5         stage('doe') {
6             steps {
7                 echo 'Hello World'
8             }
9         }
10        stage('iets in'){
11            steps{
12                echo 'Hello World'
13            }
14        }
15        stage('stages'){
16            steps{
17                echo 'Hello World'
18            }
19        }
20    }
21 }
```

Jenkins - pipeline - steps

- Steps definiëren stappen in een stage van de pipeline
- één of meerdere commando's, triggers, plugins, ...

```
1 pipeline {
2   agent any
3
4   stages {
5     stage('voorbeeld') {
6       steps {
7         sh 'ls -lah $HOME 1>./output.txt'
8         echo "output ls (showing last 10)"
9         sh 'tail output.txt' //use bash in windows
10      }
11    }
12  }
13}
```

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] { (voorbeeld)
[Pipeline] sh
+ ls -lah /var/lib/jenkins
[Pipeline] echo
output ls (showing last 10)
[Pipeline] sh
+ tail output.txt
-rw-r--r-- 1 jenkins jenkins 130 Oct 16 15:02 queue.xml.bak
-rw-r--r-- 1 jenkins jenkins 64 Oct 10 15:56 secret.key
-rw-r--r-- 1 jenkins jenkins 0 Oct 10 15:56 secret.key.not-so-secret
drwx----- 4 jenkins jenkins 4.0K Oct 18 11:33 secrets
drwxr-xr-x 3 jenkins jenkins 4.0K Oct 10 18:52 snap
drwxr-xr-x 2 jenkins jenkins 4.0K Oct 18 11:22 updates
drwxr-xr-x 2 jenkins jenkins 4.0K Oct 10 15:56 userContent
drwxr-xr-x 3 jenkins jenkins 4.0K Oct 10 15:58 users
drwxr-xr-x 2 jenkins jenkins 4.0K Oct 10 15:57 workflow-libs
drwxr-xr-x 5 jenkins jenkins 4.0K Oct 18 11:33 workspace
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Jenkins - pipeline - agents

- Agents bepalen waar stages gedraaid worden
 - Op huidige server, docker containers, VM's, ...
- Buiten de scope van dit vak

```
1 pipeline {
2     agent any
3
4     stages {
5         stage('doe') {
6             steps {
7                 echo 'Hello World'
8             }
9         }
10        stage('iets in'){
11            steps{
12                echo 'Hello World'
13            }
14        }
15        stage('stages'){
16            steps{
17                echo 'Hello World'
18            }
19        }
20    }
21 }
```


Jenkins - pipeline - Post run steps



- Stappen uitvoeren na het runnen van de volledige pipeline
- Pipelines stoppen bij het falen van een stap
 - Stappen & stages na deze stap worden niet meer uitgevoerd
 - Post run steps worden wel nog uitgevoerd

Jenkins - pipeline - Post run steps



```
pipeline {  
  agent any  
  stages {  
    stage('Test') {  
      steps {  
        sh 'echo "Fail!"; exit 1'  
      }  
    }  
  }  
  post {  
    always {  
      echo 'This will always run'  
    }  
    success {  
      echo 'This will run only if successful'  
    }  
    failure {  
      echo 'This will run only if failed'  
    }  
    unstable {  
      echo 'This will run only if the run was marked as unstable'  
    }  
    changed {  
      echo 'This will run only if the state of the Pipeline has changed'  
      echo 'For example, if the Pipeline was previously failing but is now successful'  
    }  
  }  
}
```

Jenkins - pipeline - plugins



- Uitbreiding jenkins functionaliteit
 - Plugin manager
- Gebruik via freestyle project of integratie in jenkinsfile

The screenshot shows the Jenkins Plugin Manager interface. On the left, there are links for 'Terug naar het overzicht' and 'Beheer Jenkins'. The main area has a search bar and tabs for 'Nieuwere versies', 'Beschikbaar', 'Geïnstalleerd', and 'Uitgebreid'. Below the tabs is a table with columns for 'Installeren', 'Naam', 'Versie', 'Released', and 'Geïnstalleerd'. The table lists three plugins: 'Credentials Binding', 'JUnit', and 'Pipeline: REST API'. Each plugin entry includes a checkbox for installation, the plugin name, a description, the current version, the release date, and the installed version. At the bottom, there is a button to 'Nu downloaden en installeren tijdens herstart' and a note that update information was obtained 1 hour and 58 minutes ago, along with an 'Actualiseer nu' button.

Installeren	Naam	Versie	Released	Geïnstalleerd
<input type="checkbox"/>	Credentials Binding Build Wrappers Allows credentials to be bound to environment variables for use from miscellaneous build steps.	1.24	3 dagen 14 uren ago	1.23
<input type="checkbox"/>	JUnit Build Reports Allows JUnit-format test results to be published.	1.39	4 dagen 0 uren ago	1.38
<input type="checkbox"/>	Pipeline: REST API User Interface Provides a REST API to access pipeline and pipeline run data.	2.17	6 dagen 12 uren ago	2.16

- Geen plugin? script it!


Jenkins - pipeline - plugins



Plugins voegen extra functionaliteit toe aan je pipeline script

Jenkins > Plugin Manager >

 Terug naar het overzicht

 Beheer Jenkins

Nieuwere versies

Beschikbaar

Geïnstalleerd

Uitgebreid

Installeren ↑

Naam

Versie

☐

Amazon Web Services SDK

api-plugin aws

1.11.8

This plugin provides [AWS SDK for Java](#) for other plugins.

☐

CloudBees AWS Credentials

aws

1.28

Allows storing Amazon IAM credentials within the Jenkins Credentials API. Store Amazon IAM access keys (AWSAccessKeyId and AWSSecretKey) within the Jenkins Credentials API. Also support IAM Roles and IAM MFA Token.

☐

Amazon EC2

agent aws Cloud Providers Cluster Management and Distributed Build spotinst

1.53

This plugin integrates Jenkins with [Amazon EC2](#) or anything implementing the EC2 API's such as an Ubuntu.

Jenkins - pipeline - plugins



Steps

Sample Step

ec2: Cloud template provisioning

Amazon EC2 cloud name

Template name

Generate Pipeline Script

ec2 cloud: ", template: "

Jenkins - pipeline - plugin - git



- Git plugins zorgen voor integratie met GIT & Github

Jenkins ▸ **Plugin Manager**

Terug naar het overzicht

Beheer Jenkins

Nieuwere versies

Beschikbaar

Geïnstalleerd

Uitgebreid

Actief	Naam ↓	Versie	Vorige geïnstalleerde versie	Uninstall
<input checked="" type="checkbox"/>	Credentials Plugin This plugin allows you to store credentials in Jenkins.	2.3.13		Uninstall
<input checked="" type="checkbox"/>	Display URL API Provides the DisplayURLProvider extension point to provide alternate URLs for use in notifications	2.3.3		Uninstall
<input checked="" type="checkbox"/>	Git plugin This plugin integrates Git with Jenkins.	4.4.4		Uninstall
<input checked="" type="checkbox"/>	GitHub API Plugin This plugin provides GitHub API for other plugins.	1.116		Uninstall

Jenkins - pipeline - plugin - git



- Git plugins zorgen voor integratie met GIT & Github

Steps

Sample Step

Repository URL

Branch

Credentials

☒ Include in polling?

☒ Include in changelog?

`git 'https://github.com/d-ries/voorbeeld-repository.git'`

Jenkins - pipeline - plugin - git




- Git plugins zorgen voor integratie met GIT & github

```
1 pipeline {
2   agent any
3
4   stages {
5     stage('toon code pre') {
6       steps {
7         sh 'rm -rf ./*'
8         sh 'ls -lah .'
9       }
10    }
11    stage('code ophalen') {
12      steps {
13        git 'https://github.com/d-ries/voorbeeld-repository.git'
14      }
15    }
16    stage('toon code post') {
17      steps {
18        sh 'ls -lah .'
19      }
20    }
21  }
22 }
23 }
```


Jenkins - pipeline - plugin - git

Jenkins > test-pipeline-basics >

 Back to Dashboard

Status

Changes

Start nu een bouwpoing

Configureren

Pipeline verwijderen

Full Stage View

Rename

Pipeline Syntax

Overzicht Bouwpogingen

trend ^

find

#13

18-okt-2020 13:47

#12

18-okt-2020 13:12


#11

18-okt-2020 13:09

#10

18-okt-2020 13:08

Pipeline test-pipeline-basics

 Recent Changes

Stage View

Average stage times:
(Average full run time: ~10s)

toon code pre	code ophalen	toon code post	
985ms	1s	379ms	
#13 Oct 18 15:47 No Changes	985ms	1s	379ms

Links

- Laatste bouwpoing (#13), sinds 1 minuut 14 seconden
- Laatste stabiele bouwpoing (#13), sinds 1 minuut 14 seconden
- Laatste succesvolle bouwpoing (#13), sinds 1 minuut 14 seconden
- Laatst gefaalde bouwpoing (#11), sinds 39 minuten

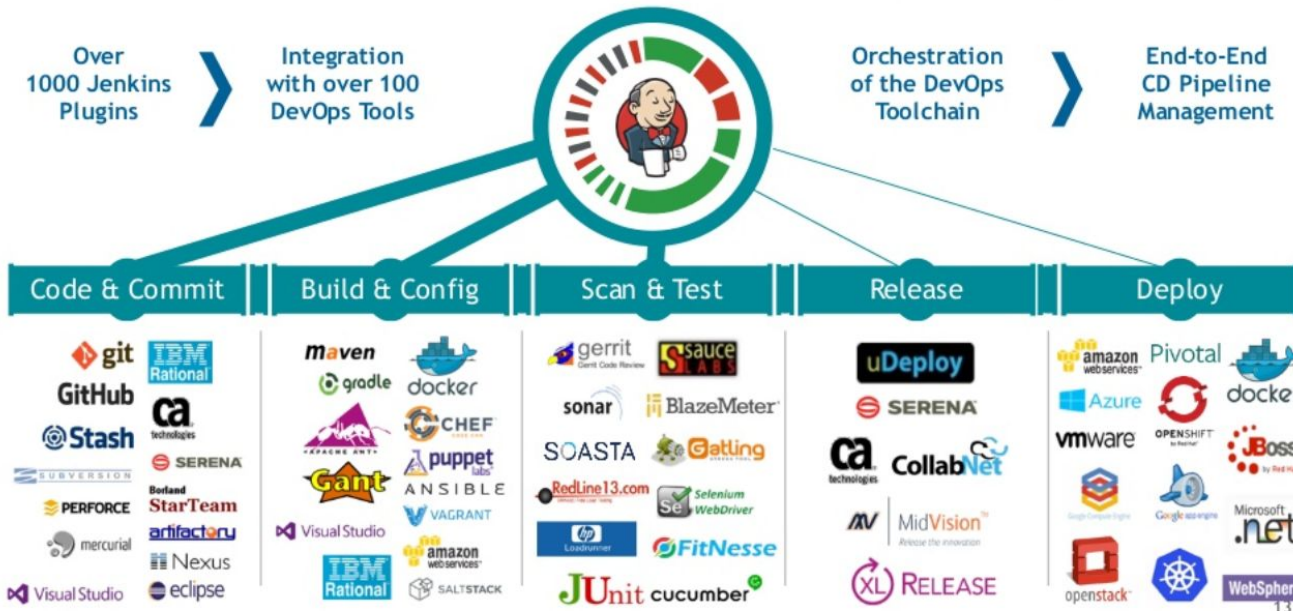
een beschrijving toevoegen

Deactiveer Project

Recap

Jenkins is the Hub of the CD/DevOps Ecosystem

Over 1000 Jenkins Plugins > Integration with over 100 DevOps Tools > Orchestration of the DevOps Toolchain > End-to-End CD Pipeline Management





Assignments

Pipelines - taak 2