Release Strategie Insidion

Mitchell Herrijgers, INF2B

21 januari 2014

Contents

1 Inleiding

Dit document bevat de Release Strategie van de Insidion homepage. Insidion is een bedrijf dat recent opgericht is door studenten. Deze studenten willen een website ontwikkelen voor het bedrijf. Hiermee hopen ze meer en grotere klanten binnen te halen. Een goede release strategie is dan ook nodig. De website willen ze regelmatig uitbreiden en verbeteren nadat de initiële versie al uitgebracht is. Hierbij kan downtime van de website mislopen van klanten beteken, ook indirect doordat downtime onbetrouwbaarheid uitstraalt. Ook de gegevens van de website moeten regelmatig veilig gesteld worden, om dataverlies door crashes en dergelijke te voorkomen.

2 Belanghebbenden

Dit gedeelte van het document bevat informatie over de belanghebbenden bij de release strategie van dit project.

2.1 Bezoekers van de website

Één van de belanghebbenden bij dit project zijn de bezoekers van de website. Deze bezoekers willen snel informatie over het bedrijf kunnen vinden, en ook contact opnemen met het bedrijf als ze geïnteresseerd zijn.

2.2 Medewerkers van Insidion

De medewerkers van Insidion hebben baat bij deze strategie omdat er minder dingen mis kunnen gaan. Als het toch misgaat zijn er vaak kant-en-klare oplossingen, wat leidt tot een betere werkervaring met minder stress.

3 Deployment technologiën

Voor deployment van het project gebruiken wij meerdere, al beschikbare, technologiën om dit zo soepel mogelijk te laten verlopen. Dit zijn:

- Apache Maven 3.1.1
- Ubuntu 12.04 LTS
- Sonarqube 4.0
- Apache Tomcat 7
- JUnit 4.11
- Selenium
- Jenkins

Waarom en hoe wij deze verschillende technologiën gebruiken zullen wij in dit hoofdstuk specificeren.

3.1 Apache Maven 3.1.1

Maven gebruiken wij voor het onderhouden van verschillende dependencies van het project. Bij het ontwikkelen van het project zullen wij third-party plugins gebruiken, wat kan leiden tot een wildgroei aan versies van deze plugins. Door Maven lossen wij dit op. Een ander probleem wat ook door Maven opgelost wordt, is de configuratie van verschillende omgevingen. Meer hierover is te vinden in het volgende hoofdstuk van dit document.

3.2 Ubuntu 12.04 LTS

Ubuntu 12.04 is onze keus als besturingssysteem voor de servers. Deze versie van Ubuntu heeft Long Term Support, en hier is dus nog vele jaren ondersteuning voor. Dit maakt onderhoud erg makkelijk.
Voor dit project gebruiken wij meerdere omgevingen. Aangezien de aanschafkosten van meerdere servers is, maken wij gebruik van een cloud service. De keuze die wij gemaakt hebben hiervoor is Digital Ocean (www.digitalocean.com) als platform. Dit vanwege zijn relatief lage prijzen en gebruiksvriendelijkheid.

3.3 Sonarqube 4.0

Een omgeving waarin progammeerfouten zichtbaar worden, dat is Sonarqube. Door dit te gebruiken zijn veel fouten in de code snel zichtbaar. Bij iedere bouw van het project wordt deze omgeving van nieuwe resultaten voorzien.

3.4 Apache Tomcat 7

Tomcat 7 is de keuze voor het draaien van de applicatie. Het is erg snel en simpel in gebruik. Ook heeft het goede Maven plugins om te gebruiken voor deployment.

3.5 JUnit 4.11

Er zijn verschillende manieren om het project te testen. Unit testing is hier een van, en hier zullen wij gebruik van maken met JUnit. JUnit is een zeer wijd ondersteunde en bekende standaar voor het uitvoeren van unit tests.

3.6 Selenium

Een ander onderdeel van testen zijn integration tests. Unit tests zijn nogal nietszeggende als verschillende onderdelen van de applicatie niet met elkaar kunnen samenwerken. Selenium lost dit voor ons op, en wederom heeft het een goede Maven ondersteuning.

3.7 Jenkins

Al deze losse onderdelen op zichzelf vereisen een hoop werk. Jenkins automatiseert dit. Wanneer er de code op de git server geüpdate wordt, zorgt jenkins automatisch voor het volgen van de deployment pipeline en het uitvoeren van tests. Jenkins is zeer uitbreidbaar door zijn plugins en heeft een zeer goede ondersteuning.

4 Configuratie Management Strategie

In dit hoofdstuk bespreken wij de configuratie van het project en de servers.

4.1 Maven

Het merendeel van alle configuratie dat dit project bevat zal te vinden zijn in de POM.xml. Dit is een bestand wat Maven vertelt wat en hoe het een taak moet doen. Er wordt gebruik gemaakt van verschillende profielen voor verschillende fasen in de deployment pipeline.

4.2 Near Zero Downtime

Een belofte van 100% uptime is uiteraard heel moeilijk om te maken. Toch is dit bijna mogelijk met de strategie die wij voor deployment gebruiken. Door gebruik te maken van Maven profielen en de deployment pipeline is de kans op fouten erg klein. Mocht er toch een fout ontstaan, kan dit snel opgelost worden door het gebruik van twee servers in Production modus. Wij zorgen ervoor altijd de mogelijkheid te hebben deze servers terug te downgraden naar de vorige, goede, versie door het gebruik van goede back-ups.

5 Deployment Pipeline

Om het project zo gunstig mogelijk te laten verlopen maken wij gebruik van een Deployment Pipeline, zoals in dit hoofdstuk omschreven. Deze pipeline bevat de volgende fases:

-