Change & release management

Fontys Hogeschool | Tilburg

Auteur: Job Verwiel, Siebren Kraak, Marco Ketelaars, Myron Antonissen, Angelo Bruggemans
Locatie: Tilburg
Versie: 0.1
Datum: 22 mei 2021

1 Versiebeheer

Versie	Wijziging	Datum
0.1	Opzet document	22 mei 2021
1.0	Verwerken release management	25 mei 2021

Inhoud

2 Inleiding

Dit is het document omtrent de change management van het ACI Project. In dit document wordt beschreven: de definition of done, release management en branch management.

3 Definition of done

Om duidelijk te houden wat de status van het project is moet worden vastgesteld wanneer een User Story op “done” mag worden gezet.

- User Story is geïmplementeerd volgende de Acceptance Criteria die staan op DevOps
- Unit Tests zijn toegevoegd en succesvol
- Documentatie is bijgewerkt
- SonarQube geeft geen fouten aan

4 Branch management

Binnen het ACI project zijn er twee bijzondere branches. Dit zijn de main en de dev branch. Naar beide branches kan niet direct gepushed worden. Om hier code naartoe te krijgen moet dus een pull request worden aangemaakt door een van de developers.

Zodra een user story wordt opgepakt door een developer, maakt deze een branch aan gebaseerd op de dev branch. Dit zorgt ervoor dat er zoveel mogelijk functionaliteiten op de nieuwe branch staan en dat de nieuwe branch zo up-to-date mogelijk is. Zodra een user story af is kan er een pull request worden aangemaakt naar de dev branch.

Om succesvol een pull request erdoorheen te krijgen moet de pull request door een aantal stappen heen. Eerst moest de pull request door de pipeline heen. De pipeline kijkt of alle testen goed runnen, of het project überhaupt opstart en pushed het project naar sonarqube om te checken op code smells en bugs. Zodra dit allemaal goed is gegaan en klopt wordt door een andere developer gekeken naar de code. De developer test hier de functionaliteit en kijkt onder andere naar naamgeving van variabelen en comments. Als dit allemaal correct is kan de pull request geaccepteerd worden.

Zodra een aantal functionaliteiten op de dev branch staan is het tijd om een nieuwe release te openbaren. Om dit te doen wordt er vanuit de dev een pull request gemaakt naar de main branch. Zodra deze pull request aangemaakt wordt, wordt er wederom een pipeline afgetrapt om zeker te weten dat alles nog correct werkt. Zodra de pull request wordt geaccepteerd wordt er een nieuwe pipeline afgetrapt. Deze pipeline pushed de microservices naar de dockerhubs waardoor de Kubernetes een nieuwe versie kan pullen.

Als een user story klaar is wordt de branch niet verwijderd. Hierdoor kan er later eventueel nog terug gekeken worden naar commits.

5 Release management

Een release is het live zetten van een stabiele versie van de front- en backend. Het maken van een release moet dus ook zorgvuldig gedaan worden en moet volledig getest zijn.

Aan het maken van een release zitten een aantal eisen. Meeste eisen zijn al gedefinieerd in de definition of done, zoals bijvoorbeeld test coverage, leesbaarheid en de applicatie is gereviewed en getest door een tweede persoon. Echter dient bij een release eerst een beta-release gemaakt te worden op de development of dev-branch. Deze branch moet dan volledig getest worden en kan eventueel op een tijdelijke server (bijvoorbeeld een beta of staging server) live gezet worden om te kijken of alles werkt zoals verwacht.

Wanneer alles op de dev server werkt zoals verwacht kan de dev-branch naar de main-branch gepusht worden. Wanneer dit gedaan wordt, word er automatisch een nieuwe Docker container gegenereerd. Deze Docker container wordt daarna gepushed naar Docker Hub.

Een nieuwe release gebeurt gebruikelijk wanneer een nieuwe sprint start of wanneer er een grote, belangrijke feature klaar is om gereleased te worden. Eventuele andere redenen om te releasen (bijvoorbeeld grote bugs die verholpen moeten worden) kunnen goedgekeurd worden indien het hele team + de PO het hier mee eens is. Uiteraard moeten ook kleine patches compleet getest worden en voldoen aan de definition of done.

Wanneer er images naar Docker Hub gepushed zijn kan alles gedeployed worden op de server(s). De scripts die gebruikt kunnen worden om de images in de lucht te krijgen zijn beschikbaar in op de servers zelf of op de GitHub van de services. Het script kan uitgevoerd worden met het commando ‘sudo microk8s kubectl apply -f <yaml_van_service>’ voor een specifieke service of ‘sudo microk8s kubectl apply -f .’ voor alle services.

Releases (push naar master) dienen in de GitHub omgeving getagged te worden met een versie nummer en een beschrijving van de veranderingen. Het nummeren van de versies gaat als volgt:

Versienummer	**Type**	**Verandering**
x.0.0	Major	Grote veranderingen in de service. Denk hierbij aan grote (mogelijk breaking) changes aan API calls of nieuwe features.
0.x.0	Minor	Kleinere veranderingen aan een service die nooit breaking changes kunnen zijn voor andere services.
0.0.x	Patch	Bug fixes of andere kleine patches die geen veranderingen aan de functionaliteit toepassen.