

Continuous Deployment

Time To Market verhoging met Continuous Deployment.

Door Bart de Best

Context:

Deze blog is ontleend aan mijn ervaringen als DevOps trainer, coach en auditor met Continuous Deployment. Elke toepassing van Continuous Deployment heeft weer meer inzichten gegeven over dit krachtige concept. Deze blog beschrijft zowel de succes verhalen als de beperkingen.

Uitdaging:

De uitdaging van de toepassing van Continuous Deployment is dat veel kennis en kunde van medewerkers vertaald moet worden naar de logica van de Continuous Integration / Continuous Deployment Secure Pipeline of wel de CI/CE sec Pipeline. Daarnaast zijn er vele controles die in de pipeline moeten worden meegenomen. Tot slot leent niet iedere applicatie zich voor een CI/CE sec Pipeline.

Oplossing:

De oplossing voor deze uitdaging is gevonden in het concept van Continuous Deployment waarin de CI/CD sec Pipeline verankerd is. Deze blog bespreekt het concept Continuous Deployment aan de hand van de volgende stappen:

1. Definitie van de CI/CD sec Pipeline
2. Definitie van Continuous Deployment value stream
3. De werkwijze
4. De ervaringen

1. Definitie van CI/CD sec Pipeline

De CI/CD Sec Pipeline is de kern van Continuous Deployment, de value stream die binnen DevOps invulling geeft deployment en release management. De CI/CD sec Pipeline is een aanpak die gericht is op het standaardiseren en automatiseren van het promoten van software en infrastructuur producten van de Ontwikkelomgeving naar de Testomgeving en vervolgens naar de Acceptatie omgeving en ten slotte naar de Productie omgeving, ook wel de OTAP straat genoemd. Hiertoe wordt uitgegaan van de volgende principes:

- Eén pipeline
- Eén ritme
- Deployment vaak release weinig
- Deployment voor Delivery
- Security in de pipeline
- Uniforme omgevingen
- Toll gate driven deployment
- Automatiseer alles
- Containers tenzij

Eén pipeline

De basis voor het CI/CD sec Pipeline is om zo min mogelijk verschillende wijzen te hebben voor de promotie van objecten naar productie.

Dit is logisch omdat elke andere werkwijze weer kennis en kunde vereist en ontwikkeld en beheerd moet worden. Naar mate er meer verschillende pipelines zijn zal de mate van control ook afnemen.

Wel komt het vaak voor dat er verschillende ontwikkelomgeving zijn maar dat deze samenkomen in dezelfde testomgeving van de pipeline. Verder moet de pipeline gezien worden als een logisch concept en kunnen er diverse fysieke pipelines gebruikt worden. Dit om de workload te beheersen en de ontwikkelingen te kunnen scheiden.

Ook is het mogelijk om omgevingen aan elkaar te koppelen waardoor verschillende DevOps team leden hun omgeving voor elkaar openstellen om samen te kunnen werken. Dit wordt een connected omgeving genoemd.

Eén ritme

Het samen ontwikkelen van DevOps teams aan één informatiesysteem vereist vaak dat de DevOps teams hetzelfde ritme van deployen hebben. Dit betekent veelal ook dat de sprintlengte moet zijn afgestemd in geval van Agile Scrum. Ook moet er afstemming zijn over de realisatie van de product backlog.

Deployment vaak release weinig

Een deployment vindt in de regel vaak plaats zonder dat de functionaliteit aan de gebruiker wordt vrijgegeven middels een release. Dit geeft de mogelijkheid om delen van de functionaliteit al vast te controleren op een juiste werking met productie data. Het deployen in de productieomgeving zonder release wordt ook wel een dark launch genoemd.

Deployment voor Delivery

Het doorlopen van de OTAP-straat zonder manuele interventie wordt Continuous Deployment genoemd. Als er nog een manuele stap in de pipeline zit, dan wordt gesproken over Continuous Delivery.

Security in de pipeline

De pipeline moet de informatiebeveiliging versterken door het afdwingen van het beheersen van informatiebeveiliging risico's in de vorm van tegenmaatregelen (controls genaamd). De pipeline moet controleren of de controls effectief zijn.

Uniforme omgevingen

Het kunnen promoten van software en infrastructuur in de pipeline vereist dat de omgevingen identiek zijn qua stack, behoudens de wijzigingen die doorgevoerd worden door de CI/CD sec Pipeline.

Toll gate driven deployment

De promoties van objecten door de CI/CD sec Pipeline moet gebaseerd zijn op controles van functionaliteit en kwaliteit door geautomatiseerd testcases uit te voeren. De resultaten van deze testcases moet geautomatiseerd geverifieerd worden op de verwachte resultaten. Defecten moeten worden geregistreerd. De beslissing om wel of niet de promotie door te zetten moet ook geautomatiseerd worden.

Zo kan bijvoorbeeld een zero tolerance worden gehanteerd waarbij geen enkel defect toegestaan is. Bij een defect wordt de promotie dan gestaakt. Dit vereist een shift left organisatie (Zie Blog Productiviteitsverhoging door Continuous Testing).

Automatiseer alles

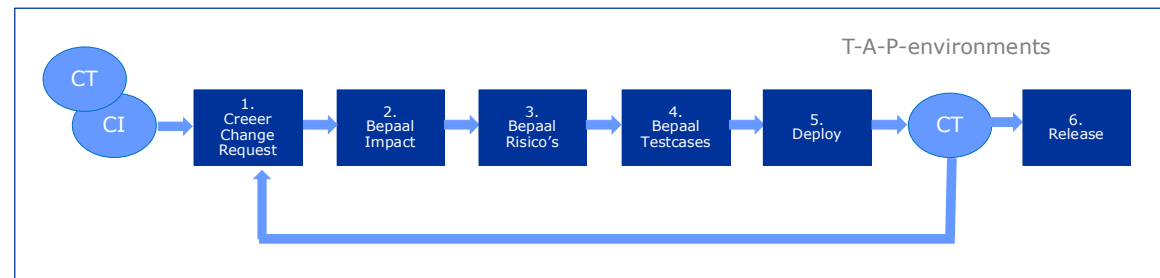
Elke manuele handeling is een directe vertraging van de time-to-market. Het automatiseren is dan ook een belangrijk aspect van de CI/CD sec Pipeline.

Containers tenzij

Het voordeel van een CI/CD sec Pipeline als Docker is dat de installatie en configuratie van de applicatie in de ontwikkelomgeving plaatsvindt en daarna niet meer. De container is het object van deployment en release geworden.

2. Definitie van Continuous Deployment value stream

Een voorbeeld van een Continuous Deployment value stream is weergegeven in [figuur 1](#). De value stream is beperkt gehouden qua stappen. Belangrijk is om de relatie te onderkennen van Continuous Testing en Continuous Integration met Continuous Deployment. In stap 1 is weergegeven dat de trigger van de Continuous Deployment value stream gelegen is in een change request. Dit is nodig voor de autorisatie van de mutatie van de productieomgeving. Op basis daarvan moet de impact (2) en de risico's (3) worden bepaald zodat de testcases (4) aangemaakt kunnen worden om volledig in control de deployments van A-T, van T-A en van T-P uit te voeren (5) en tenslotte de nieuwe of aangepast functionaliteit aan de gebruikers vrij te geven (6).

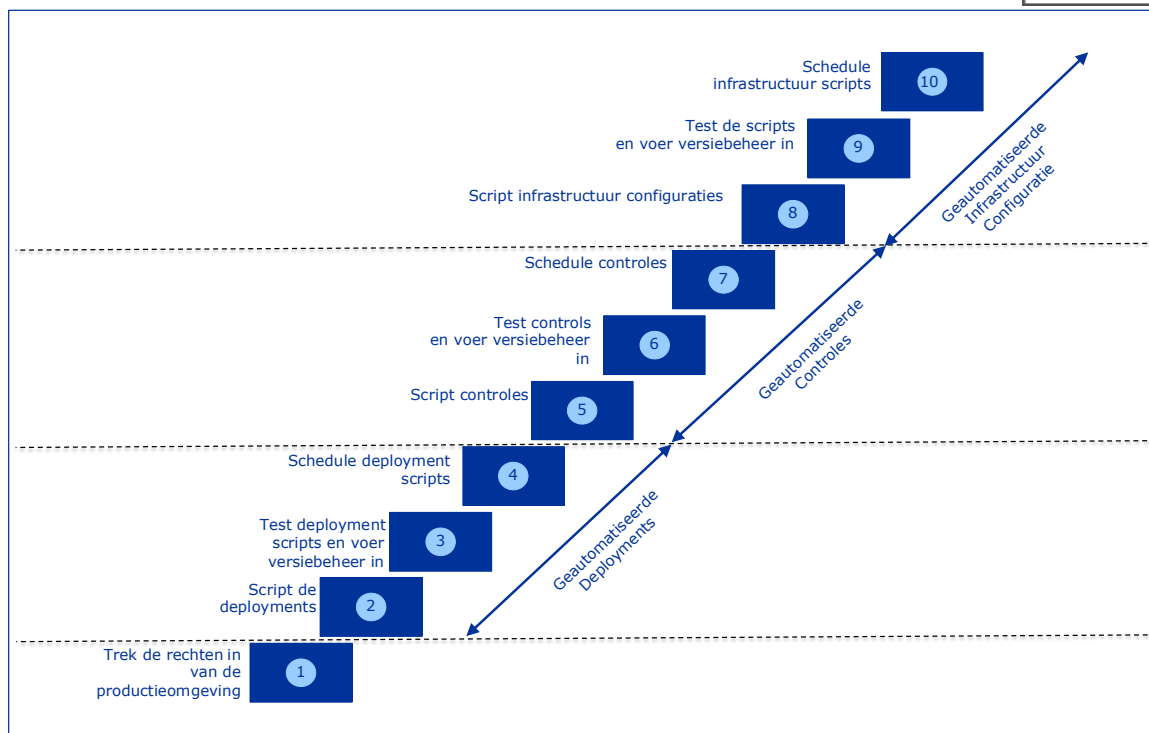


Figuur 1. Continuous Deployment value stream.

3. De werkwijze

In [figuur 2](#) is een overzicht gegeven van de mogelijke stappen om de CI/CD sec Pipeline invulling te geven. De eerste stap is het stabiliseren van de productieomgeving door de toegang tot de productieomgeving te ontnemen. Daartoe moeten deployments wel geautomatiseerd kunnen plaatsvinden, anders is er nog steeds toegang nodig voor een manuele promotie.

Daarna is een belangrijke stap om de controles in de OTAP straat te automatiseren. Tenslotte zijn ook infrastructurele wijzigingen te automatiseren met bijvoorbeeld infrastructure as code.



Figuur 2, Stappen om de CI/CD sec Pipeline in te regelen.

4. De ervaringen

In de loop van de jaren heb ik verschillende ervaringen mogen opdoen met de CI/CD sec Pipeline in het kader van Continuous Deployment die ik graag met u deel.

Eigen ervaring

Het dichtzetten van de productieomgeving is simpeler gezegd dan gedaan. Vaak zijn er veel meer rechten gegeven aan medewerkers dan men zich bewust is. Zo hebben veel functioneel beheerders toegang tot de productieomgeving om informatie incidenten op te lossen. Ook worden er in productie vaak nog rapportages met de hand uitgevoerd.

Maar ook voor de installatie en configuratie van applicaties en infrastructuurvoorzieningen worden rechten gegeven aan medewerkers. Het ontnemen van rechten kan gezien worden als het inperken van bevoegdheden en daarmee een demotie. Het is dan ook belangrijk om duidelijk aan te geven wat de reden is voor de introductie van Continuous Deployment en wat de baten zijn. Elk nadeel moet serieus worden genomen en waar mogelijk voorzien worden van een oplossing.

Verder is het belangrijk om de CI/CD sec pipeline zo vroeg als mogelijk te ontwikkelen en te gebruiken. Daardoor wordt tijd gewonnen om fouten te herstellen.

Training ervaringen

In trainingen vinden vaak discussies plaats of het wel noodzakelijk is om altijd Continuous Deployment te gebruiken. Het antwoord op deze vraag is dat er een tweedeling valt te onderkennen.

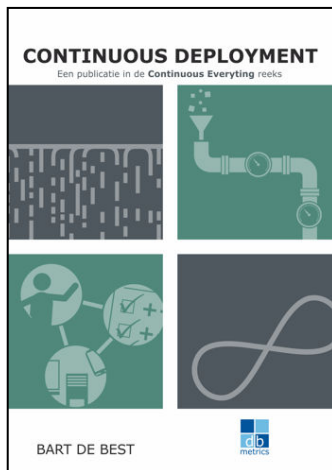
Zo hebben de System of Engagement (SoE) applicaties (e-commerce applicaties zoals webwinkels) meestal een eenvoudige architectuur met loosely coupled systemen als backoffice applicaties. In dat geval is Continuous Deployment goed mogelijk. Dit is overigens ook het overgrote deel van de DevOps toepassingen. Als het over System of Records (SoR) applicaties gaat (transactie verwerkende applicaties) dan is Continuous Delivery een goede mogelijkheid omdat er vaak manuele controles op de financiën nodig zijn die vaak moeizaam geautomatiseerd kunnen worden.

Audit ervaringen

Bij het auditen van organisaties blijkt dat lang niet alle mogelijk patterns van deployment en releases worden gebruikt. Zo wordt de blauw/groen pattern niet veel gebruikt en ook A/B testing niet vaak. Lees meer over deze patterns in het boek Continuous Deployment.

Met deze toepassing van de CI/CD sec Pipeline is het mogelijk om de continuïteit van de productieomgeving te borgen en de time to market te verbeteren. Daarom is deze wijze van invulling geven aan de CI/CD sec Pipeline een goed voorbeeld van de toepassing van Continuous Deployment.

Door Bart de Best
DutchNordic.Group



<https://www.dbmetrics.nl/ce-nl/continuous-deployment-nl/>