

Continuous Deployment

Time To Market increase with Continuous Deployment.

By Bart de Best

Context:

This blog is derived from my experiences as a DevOps trainer, coach, and auditor with Continuous Deployment. Each application of Continuous Deployment has provided more insights into this powerful concept. This blog describes both the success stories and the limitations.

Challenge:

The challenge of applying Continuous Deployment is that a lot of knowledge and skills of employees must be translated into the logic of the Continuous Integration / Continuous Deployment Secure Pipeline, or the CI/CE sec Pipeline. In addition, there are many checks that must be included in the pipeline. Finally, not every application is suitable for a CI/CE sec Pipeline.

Solution:

The solution to this challenge has been found in the concept of Continuous Deployment in which the CI/CD sec Pipeline is anchored. This blog discusses the concept of Continuous Deployment through the following steps:

1. Definition of the CI/CD sec Pipeline
2. Definition of Continuous Deployment value stream
3. The method
4. The experiences

1. Definition of the CI/CD sec Pipeline

The CI/CD Sec Pipeline is the core of Continuous Deployment, the value stream that defines deployment and release management within DevOps. The CI/CD sec Pipeline is an approach aimed at standardising and automating the promotion of software and infrastructure products from the Development Environment to the Test Environment and then to the Acceptance environment and finally to the Production environment, also known as the DTAP street. named. This is based on the following principles:

- One pipeline
- One rhythm
- Deployment often releases little
- Deployment for Delivery
- Security in the pipeline
- Uniform environments
- Toll gate driven deployment
- Automate everything
- Containers unless

One pipeline

The basis for the CI/CD sec Pipeline is to have as few different ways as possible to promote objects to production. This is logical because every other method requires

knowledge and skills and must be developed and managed. As there are more different pipelines, the degree of control will also decrease.

It often happens that there are different development environments, but they come together in the same test environment of the pipeline. Furthermore, the pipeline should be seen as a logical concept and various physical pipelines can be used. This is to control the workload and to separate developments.

It is also possible to link environments together, allowing different DevOps team members to open their environments to each other to collaborate. This is called a connected environment.

One rhythm

Developing DevOps teams together on one information system often requires that the DevOps teams have the same deployment rhythm. This often also means that the sprint length must be coordinated in the case of Agile Scrum. There must also be coordination about the realisation of the product backlog.

Deployment often releases little

As a rule, a deployment often takes place without the functionality being released to the user through a release. This provides the opportunity to check parts of the functionality for correct operation with production data. Deploying in the production environment without a release is also called a dark launch.

Deployment over Delivery

Going through the DTAP street without manual intervention is called Continuous Deployment. If there is still a manual step in the pipeline, it is called Continuous Delivery.

Security in the pipeline

The pipeline must strengthen information security by enforcing the management of information security risks in the form of countermeasures (called controls). The pipeline must check whether the controls are effective.

Uniform environments

Being able to promote software and infrastructure in the pipeline requires that the environments be identical in stack, except for the changes implemented by the CI/CD sec Pipeline.

Toll gate driven deployment

The promotions of objects through the CI/CD sec Pipeline should be based on functionality and quality checks by automated execution of test cases. The results of these test cases must be automatically verified for the expected results. Defects must be recorded. The decision whether or not to continue the promotion must also be automated. For example, a zero tolerance can be applied whereby no defects are

permitted. In the event of a defect, the promotion will be stopped. This requires a shift left organisation (See Blog Productivity Increase through Continuous Testing).

Automate everything

Every manual action directly delays the time-to-market. Automation is therefore an important aspect of the CI/CD sec Pipeline.

Containers unless

The advantage of a CI/CD sec Pipeline such as Docker is that the installation and configuration of the application takes place in the development environment and no longer afterwards. The container has become the object of deployment and release.

2. Definition of Continuous Deployment value stream

An example of a Continuous Deployment value stream is shown in [Figure 1](#). The value stream is kept limited in terms of steps. It is important to recognise the relationship between Continuous Testing and Continuous Integration with Continuous Deployment. Step 1 shows that the trigger of the Continuous Deployment value stream is located in a change request. This is necessary for the authorisation of the change to the production environment. On this basis, the impact (2) and the risks (3) must be determined so that the test cases (4) can be created to fully control the deployments of A-T, T-A and T-P (5) and finally the new or release modified functionality to users (6).

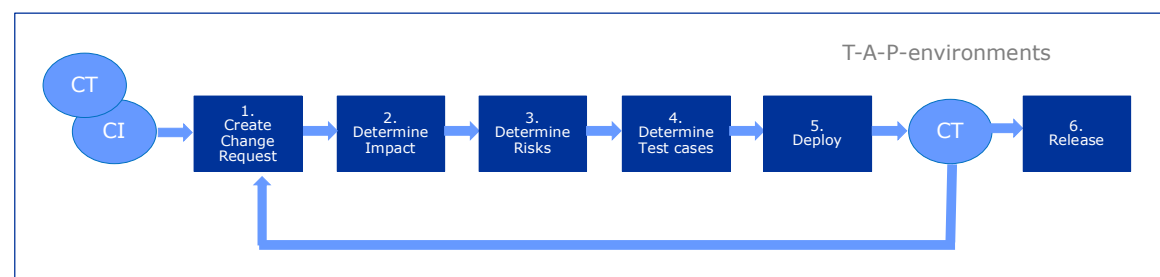


Figure 1. Continuous Deployment value stream.

3. The Way of Working

[Figure 2](#) provides an overview of the possible steps to implement the CI/CD sec Pipeline. The first step is to stabilise the production environment by removing access to the production environment. To this end, deployments must be able to take place automatically, otherwise access is still required for manual promotion.

Then an important step is to automate the checks in the DTAP street. Finally, infrastructure changes can also be automated with, for example, infrastructure as code.

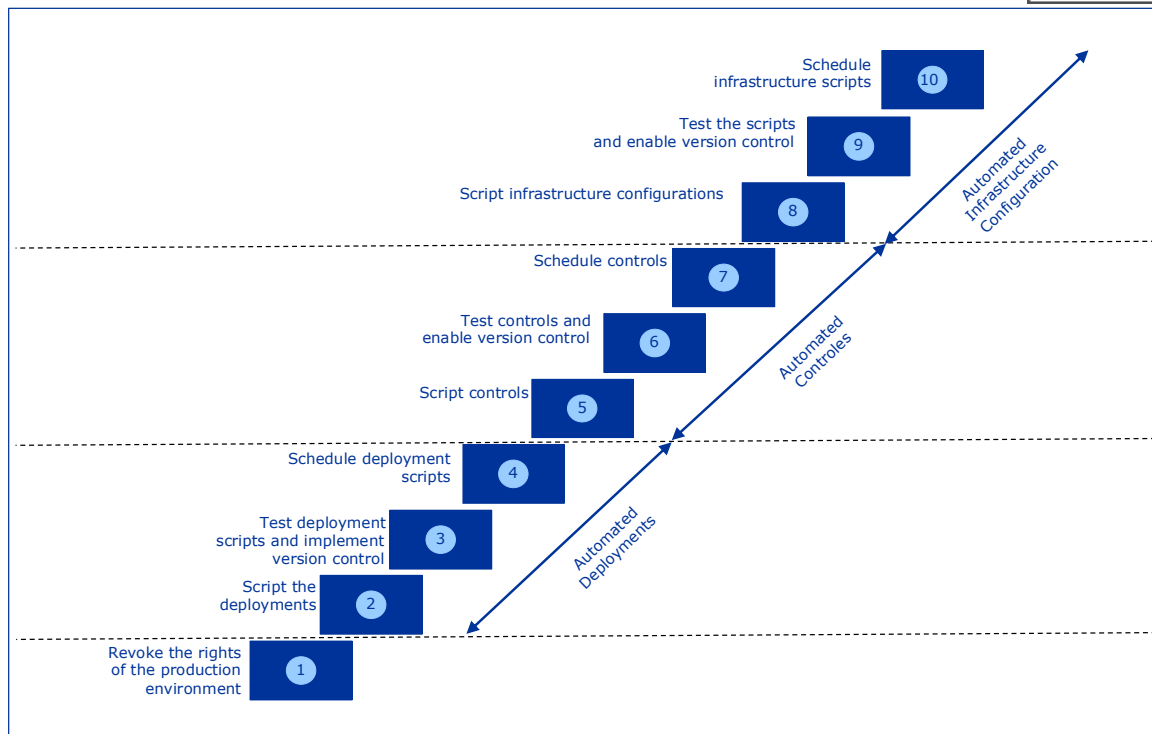


Figure 2, Steps to configure the CI/CD sec Pipeline.

4. The experiences

Over the years I have had various experiences with the CI/CD sec Pipeline in the context of Continuous Deployment, which I would like to share with you.

Own experiences

Shutting down the production environment is easier said than done. Often many more rights are given to employees than people are aware of. For example, many functional managers have access to the production environment to resolve information incidents. Reports are also often carried out manually in production. But employees are also given rights for the installation and configuration of applications and infrastructure facilities. The deprivation of rights can be seen as limiting powers and therefore a demotion. It is therefore important to clearly indicate the reason for the introduction of Continuous Deployment and what the benefits are. Every disadvantage must be taken seriously and provided with a solution where possible.

Furthermore, it is important to develop and use the CI/CD sec pipeline as early as possible. This saves time to correct errors.

Training experiences

Discussions often take place in training as to whether it is necessary to always use Continuous Deployment. The answer to this question is that there is a division. For example, the System of Engagement (SoE) applications (e-commerce applications such as online stores) usually have a simple architecture with loosely coupled systems as back-office applications. In that case, Continuous Deployment is quite possible. This is also the vast majority of DevOps applications. When it comes to System of Records (SoR) applications (transaction processing applications),

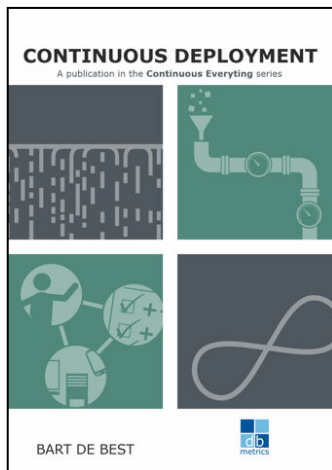
Continuous Delivery is a good option because manual checks on finances are often required that are often difficult to automate.

Audit experiences

When auditing organisations, it appears that not all possible patterns of deployment and releases are used. For example, the blue/green pattern is not used much, and A/B testing is not often used either. Read more about these patterns in the book Continuous Deployment.

With this application of the CI/CD sec Pipeline it is possible to ensure the continuity of the production environment and improve the time to market. That is why this method of implementing the CI/CD sec Pipeline is a good example of the application of Continuous Deployment.

By Bart de Best
DutchNordic.Group



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