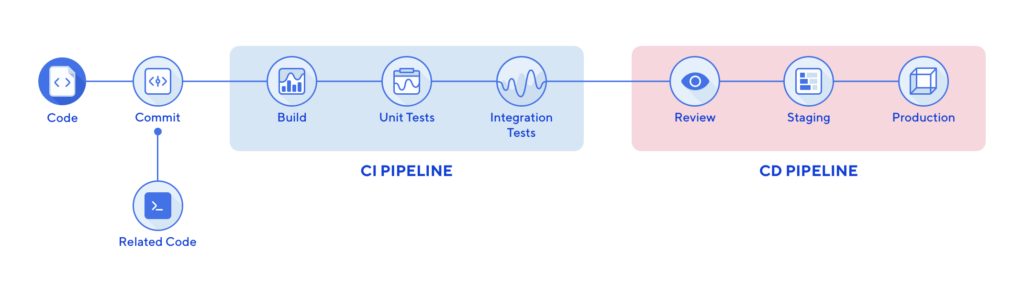
# Hoe zorg ik voor een veilige CI/CD pipeline?

Voor mijn applicatie wil ik een CI/CD pipeline maken, alleen deze moet natuurlijk wel beschermd zijn tegen gevaren van buiten. Hiermee kan ik zeker weten dat de pipeline werkt tijdens dat hij wordt uitgevoerd.

## Wat is CI/CD?

Continuous Integration en Continuous Delivery is een manier hoe software teams te werk gaan. Hierbij worden wijzigingen in de code geautomatiseerd afgehandeld. Denk hierbij aan testen en het doorzetten van de code naar de productieomgeving. Hierdoor wordt de code betrouwbaarder omdat er minder vaak fouten in de code worden toegelaten.



<https://www.elasticweb.nl/kennisbank/continuous-integration-en-continuous-delivery-verder-uitgelegd>

## Hoe kan ik mijn pipeline beschermen?

Het maken van een pipeline zelf vergt al best wat tijd, maar het is natuurlijk extra belangrijk dat deze pipeline veilig is, zodat er niet tussen de geautomatiseerde workflow schadelijke code kan komen die je applicatie doet crashen. Er zijn verschillende manieren om dit te voorkomen:

* **Het scannen/analyseren van de source code**, kan worden gedaan door een externe applicatie (denk aan SonarQube of SonarCloud). Hiermee voorkom je potentiële veiligheid risico’s.
* **Het testen van de veiligheid van je applicatie**, spreekt voor zichzelf. Hiermee schrijf je testen die je applicatie test op het gebied van veiligheid. Deze testen kunnen ook door externe applicaties zoals SonarQube worden gedaan.
* **Runtime veiligheid**, kan de veiligheid van je applicatie monitoren zelfs na de deployment van de applicatie. Dit wordt gedaan door gevaren die tijdens productie kunnen voorkomen te identificeren en erop te reageren.

<https://www.paloaltonetworks.com/cyberpedia/what-is-the-ci-cd-pipeline-and-ci-cd-security>

## Waarom is SonarQube een goede oplossing?

SonarQube is heel veelzijdig. Het is geschreven in Java maar het ondersteunt meer dan 20+ andere codeertalen. Daarnaast zijn er ook vele plugins die de nuttigheid van dit programma vergroten, zoals SonarLint.

SonarQube dekt zeven verschillende secties die te maken hebben met de kwaliteit van je code:

* *Architectuur en design*
* *Unit testen*
* *Dubbele code*
* *Potentiele bugs*
* *Complexe code*
* *Codeer standaarden*
* *Comments*

Hiermee zorgt het ervoor dat de geschreven code minder kwetsbaar is, dus minder snel wordt blootgesteld aan veiligheid risico’s. Daarnaast zorgt het er ook voor dat de code een stuk leesbaarder en netjes is, waardoor de productiviteit van het team wordt vergroot. Een groot deel van de nuttige maar tijdrovende taken worden overgenomen door de applicatie. Ook detecteert het errors en geeft notificaties zodat de ontwikkelaars weten wat er fout is in de code. Met al deze dingen zorgt SonarQube er dus voor dat het risico op slecht geschreven en kwetsbare code wordt vermindert.

Door SonarQube in mijn pipeline te verwerken kan ik er dus voor zorgen dat mijn pipeline ook de veiligheid van mijn code checkt, en niet alleen of de code het doen.

<https://www.vizteck.com/post/benefits-of-using-sonarqube-for-code-reviews>

<https://www.loginradius.com/blog/engineering/sonarqube/#:~:text=SonarQube%20is%20an%20open%2Dsource,smells%2C%20vulnerabilities%2C%20code%20duplications>.

## Conclusie

Na het onderzoeken van verschillende bronnen ben ik erachter gekomen dat SonarQube precies is wat ik zoek om mijn pipeline te verbeteren op het gebied van softwarekwaliteit, maar voornamelijk veiligheid. Ik heb aar verschillende artikelen gekeken om te valideren dat SonarQube ook echt doet wat het zegt op de website, waardoor het betrouwbaarder voor mezelf is.