

Week 2:









































Voor het project wordt er gebruik gemaakt van Docker-containers om verschillende onderdelen van de applicatie te isoleren en te beheren, wat de ontwikkeling en deployment processen vereenvoudigt. Centraal in de architectuur staat een hoofdcontainer (itvb23ows-starter-code-fvd), die verantwoordelijk is voor het starten van drie specifieke subcontainers via een docker-compose.yml bestand. Deze subcontainers omvatten:

- Een container voor de Hive database. Het gebruik van de database biedt een krachtige, schaalbare opslaglaag die geoptimaliseerd is voor query's en data-analyse, cruciaal voor de applicatiebehoeften.
- De app container host de webapplicatie, het gebruikersinterface component dat interactie met gebruikers mogelijk maakt. Door dit in een afzonderlijke container te draaien, zorgen we voor flexibiliteit en schaalbaarheid in hoe gebruikers toegang krijgen.
- Een speciale container voor de AI, die de logica bevat voor het genereren van tegenstanderzetten binnen het spel. Dit zorgt ervoor dat een speler tegen een computer kan spelen.

Elk component kan onafhankelijk worden ontwikkeld, bijgewerkt en gedeployed, wat het ontwikkelproces stroomlijnt.

Bovendien draaien Jenkins en SonarQube in afzonderlijke containers. Jenkins wordt gebruikt voor automatisering van het build- en deploymentproces. SonarQube geeft inzicht in de codekwaliteit en veiligheidsissues.

De dockerfiles die behoren tot de verschillende containers kunnen gevonden worden in de repository.

<input type="checkbox"/>	 sonarqube 07d8b8848b48 	sonarqube:latest	Running	1.51%	9000:9000 	4 hours ago			
<input type="checkbox"/>	 jenkins-docker 135c53c8803e 	docker:dind	Running	0.28%	2376:2376 	10 hours ago			
<input type="checkbox"/>	 jenkins-blueocean 12ebfe749f50 	myjenkins-blueocean:2.440.2-1	Running	0.28%	50000:50000  Show all ports (2)	7 minutes ago			
<input type="checkbox"/>	 itvb23ows-starter-c		Running (3/3)	2.4%		19 seconds ago			
<input type="checkbox"/>	 hive 49eef4579640 	itvb23ows-starter-code-fvd-db	Running	1.37%	3306:3306 	20 seconds ago			
<input type="checkbox"/>	 app 925da70dfa83 	itvb23ows-starter-code-fvd-app	Running	0.01%	80:80 	19 seconds ago			
<input type="checkbox"/>	 ai e12c328a65c0 	itvb23ows-starter-code-fvd-ai	Running	1.02%	5000:5000 	20 seconds ago			

Week 3:

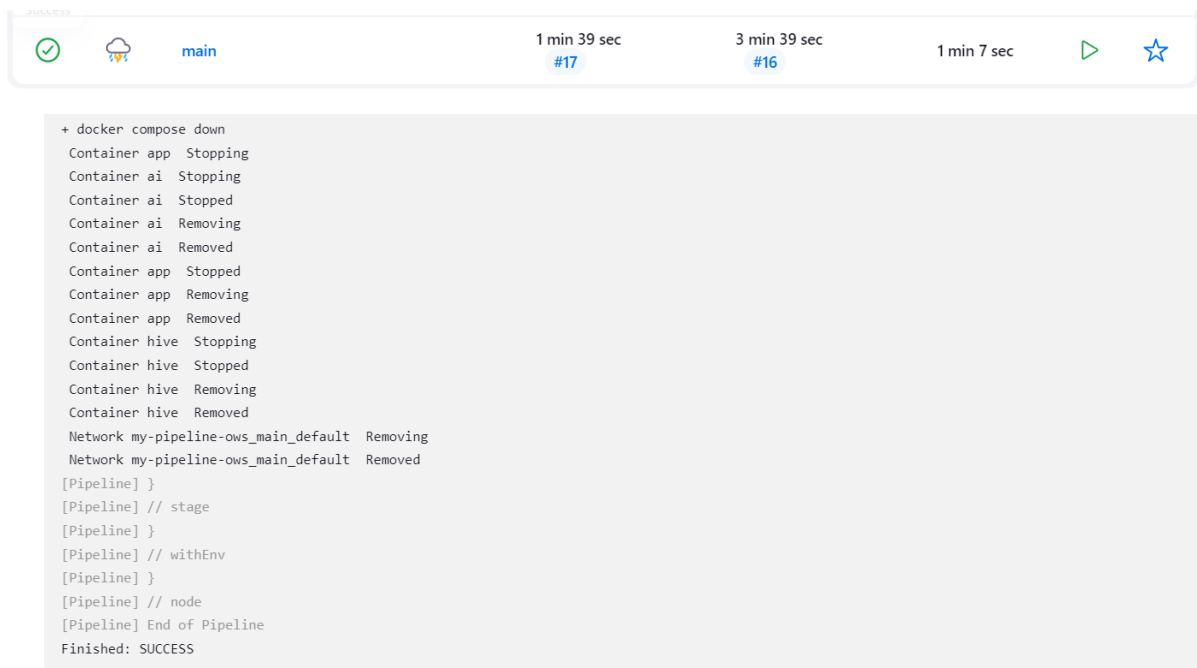
Toelichting:

Voor het project is gekozen voor een branchingstrategie die gebruikmaakt van feature branches, een keuze die veel voordelen biedt voor het ontwikkelingsproces met betrekking tot de requirements van het Ontwikkelstraten project. Deze strategie staat centraal in het faciliteren van een modulaire en gefocuste aanpak waarbij elke feature wordt geïmplementeerd in een eigen branche. Dit zorgt ervoor dat de 'main' branch ten alle tijden een werkende applicatie bevat, en 'features' los geïmplementeerd kunnen worden.

Bovendien biedt het gebruik van feature branches een vangnet voor risicomanagement. Als een feature niet aan de verwachtingen voldoet of niet op tijd gereed is, kan het eenvoudig worden uitgesteld zonder de stabiliteit van de hoofdbanche (main) in gevaar te brengen. Dit houdt de hoofdbanche schoon en release-klaar, vrij van onafgewerkte of onstabiele features.

Tot slot helpt de strategie bij het stroomlijnen van de integratie van nieuwe features. Door regelmatig de wijzigingen van de hoofdbanche in de feature branche te mergen, blijven features synchroon met de basiscode. Dit vermindert de kans op complexe merge conflicten en vergemakkelijkt een soepele en efficiënte integratie wanneer de feature klaar is om te worden samengevoegd met de hoofdbanche.

Jenkins draait de repository en verschillende containers succesvol.



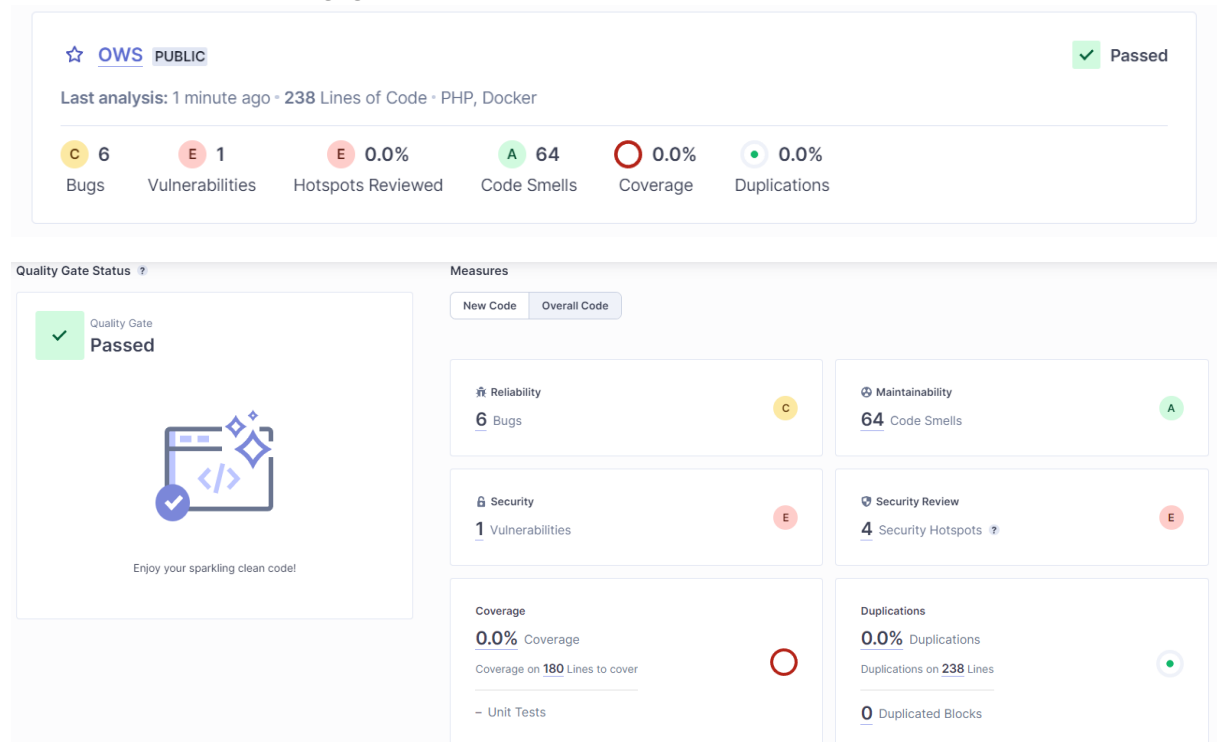
```
+ docker compose down
Container app Stopping
Container ai Stopping
Container ai Stopped
Container ai Removing
Container ai Removed
Container app Stopped
Container app Removing
Container app Removed
Container hive Stopping
Container hive Stopped
Container hive Removing
Container hive Removed
Network my-pipeline-ows_main_default Removing
Network my-pipeline-ows_main_default Removed
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Week 4:

SonarQube draait succesvol binnen Jenkins.

```
INFO: Sensor VB.NET Properties [vbnet]
INFO: Sensor VB.NET Properties [vbnet] (done) | time=0ms
INFO: Sensor IaC Docker Sensor [iac]
INFO: 3 source files to be analyzed
INFO: 3/3 source files have been analyzed
INFO: Sensor IaC Docker Sensor [iac] (done) | time=80ms
INFO: ----- Run sensors on project
INFO: Sensor Analysis Warnings import [csharp]
INFO: Sensor Analysis Warnings import [csharp] (done) | time=2ms
INFO: Sensor Zero Coverage Sensor
INFO: Sensor Zero Coverage Sensor (done) | time=23ms
INFO: CPD Executor 1 file had no CPD blocks
INFO: CPD Executor Calculating CPD for 29 files
INFO: CPD Executor CPD calculation finished (done) | time=30ms
INFO: Analysis report generated in 101ms, dir size=398.0 kB
INFO: Analysis report compressed in 80ms, zip size=167.1 kB
INFO: Analysis report uploaded in 39ms
INFO: ANALYSIS SUCCESSFUL, you can find the results at: http://sonarqube:9000/dashboard?id=ows
INFO: Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
INFO: More about the report processing at http://sonarqube:9000/api/ce/task?id=48af903a-890d-4a2b-ad91-09b9fa061c16
INFO: Analysis total time: 32.657 s
INFO: -----
INFO: EXECUTION SUCCESS
INFO: -----
INFO: Total time: 35.042s
INFO: Final Memory: 29M/104M
INFO: -----
```

Het resultaat wordt weergegeven in SonarQube



Categorie: Consistentie

Software kwaliteit: Onderhoudbaarheid

Probleem: Add a new line at the end of the file

Beschrijving: Een nieuwe regel toe te voegen aan het einde van broncodebestanden kan integratieproblemen veroorzaken met versiebeheersystemen en kan complicaties veroorzaken bij het aaneenschakelen van bestanden of uitvoeren van command-line utilities. Een nieuwe regel aan het einde zorgt voor betere compatibiliteit en voorkomt onnodige verschillen tussen bestandsversies.

Mogelijke oplossing: Voeg een lege regel toe aan het einde van elk bestand.

Categorie: Consistentie

Software kwaliteit: Onderhoudbaarheid

Probleem: Rename function to match regular expression

Beschrijving: Consistentie in naamgeving binnen een codebase is cruciaal voor de leesbaarheid en onderhoudbaarheid. Afwijken van een vastgestelde naamgevingsconventie kan verwarring veroorzaken bij teamleden. Het aanpassen van functienamen om te voldoen aan een specifieke regular expression zorgt voor uniformiteit en verbetert de samenwerking en codekwaliteit.

Mogelijke oplossing: Schrijf de functies zo zodat ze overeenkomen met het opgegeven patroon, en zorg ervoor dat ze beginnen met een kleine letter gevolgd door alfanumerieke tekens.

Categorie: Consistentie

Software kwaliteit: Onderhoudbaarheid

Probleem: Remove closing PHP tag `?>`

Beschrijving: In PHP-bestanden, vooral die enkel PHP-code bevatten, is het aanbevolen de afsluitende PHP-tag `?>` weg te laten. Dit voorkomt het risico van onbedoeld verzenden van witruimte of extra regels naar de client, wat kan resulteren in headers already sent fouten. Het verwijderen van de afsluitende tag draagt bij aan een schonere code en voorkomt mogelijke output buffering fouten.

Mogelijke oplossing: Verwijder de afsluitende `?>`-tag uit de PHP-bestanden.

Categorie: Consistentie

Software kwaliteit: Onderhoudbaarheid

Probleem: Conditional statement execution issue

Beschrijving: Incorrect gebruik van accolades in conditionele statements kan leiden tot blokken code die altijd uitgevoerd worden, ongeacht de bedoeling. Dit resulteert in onvoorspelbaar gedrag en bugs die moeilijk te traceren zijn. Het correct omringen van codeblokken met accolades verzekert de gewenste uitvoeringsflow en verhoogt de code betrouwbaarheid.

Mogelijke oplossing: Zorg voor het juiste gebruik van accolades `{ }` om alle bedoelde code binnen het blok op te nemen.

Categorie: Consistentie

Software kwaliteit: Betrouwbaarheid

Probleem: Replace "include" with "include_once"

Beschrijving: Het herhaaldelijk opnemen van hetzelfde bestand met include kan leiden tot fouten zoals variabele conflicten. include_once zorgt ervoor dat het bestand slechts één keer wordt opgenomen, waardoor de kans op dergelijke fouten wordt geminimaliseerd. Dit verbetert de stabiliteit en voorspelbaarheid van de applicatie.

Mogelijke oplossing: Gebruik include_once om ervoor te zorgen dat het bestand slechts één keer wordt opgenomen.

Categorie: Consistentie

Software kwaliteit: Onderhoudbaarheid

Probleem: Split Long Lines (greater than 120 characters)

Beschrijving: Lange regels code kunnen de leesbaarheid aanzienlijk verminderen en maken het moeilijk om de code te volgen zonder horizontaal te scrollen. Het opdelen van deze regels in kortere, overzichtelijke stukken maakt de code toegankelijker en eenvoudiger te onderhouden. Dit is vooral belangrijk in teams waarbij code reviews een essentieel onderdeel zijn van het ontwikkelproces.

Mogelijke oplossing: Breek lange regels op in meerdere kortere regels. Dit kan vaak worden gedaan door variabelen te introduceren voor tussenresultaten.

Categorie: Intentionaliteit

Software kwaliteit: Onderhoudbaarheid

Probleem: Add curly braces around the nested statement(s)

Beschrijving: Het weglaten van accolades rond geneste instructies kan leiden tot verwarring over welke statements deel uitmaken van een conditionele of lus constructie. Dit verhoogt het risico op fouten wanneer de code wordt gewijzigd of uitgebreid. Door consequent accolades te gebruiken, zelfs voor enkele instructies, wordt de code duidelijker en robuuster.

Mogelijke oplossing: Gebruik altijd accolades { } rond instructies in besturingsstructuren, zelfs als er maar één instructie is. Dit maakt de reikwijdte van de controlestructuur duidelijk en voorkomt toekomstige fouten wanneer er meer regels worden toegevoegd.

Categorie: Intentionaliteit

Software kwaliteit: Betrouwbaarheid

Probleem: Add "lang" and/or "xml:lang" attributes to the "<html>" element

Beschrijving: Het specificeren van de taal van de inhoud via het "lang" attribuut in de <html> tag is cruciaal voor toegankelijkheid. Het helpt schermlezers en andere hulpmiddelen om de inhoud correct te interpreteren en voor te lezen. Dit verhoogt de bruikbaarheid voor gebruikers met visuele beperkingen en verbetert de algemene toegankelijkheid van webinhoud.

Mogelijke oplossing: Voeg het lang-attribuut toe aan de <html>-tag. Als het document XML-inhoud bevat, voeg je ook xml:lang toe.

Categorie: Intentionaliteit

Software kwaliteit: Onderhoudbaarheid

Probleem: Remove the parentheses from "echo" call

Beschrijving: 'Echo' is een taalconstructie in PHP, geen functie, wat betekent dat het gebruik van haakjes optioneel en vaak overbodig is. Het vermijden van haakjes draagt bij aan een helderdere syntax en is in overeenstemming met PHP best practices. Dit verhoogt de consistentie en vermijdt verwarring bij minder ervaren ontwikkelaars.

Mogelijke oplossing: Verwijder haakjes bij gebruik van echo.

Categorie: Intentionaliteit

Software kwaliteit: Onderhoudbaarheid

Probleem: Function with more than three returns

Beschrijving: Functies met meerdere returnpunten kunnen moeilijk te volgen en te begrijpen zijn, wat de codeonderhoud bemoeilijkt. Het stroomlijnen van functies om een enkele uitgangspunt te hebben, verbetert de duidelijkheid en maakt de code makkelijker te debuggen en te refactoren. Dit kan vaak bereikt worden door een variabele te gebruiken om het resultaat op te slaan en terug te keren aan het einde van de functie.

Mogelijke oplossing: Herstructureer de functie om het aantal returns te verminderen. Overweeg het gebruik van een variabele om de return waarde vast te houden en een enkele return-instructie aan het einde van de functie.

Categorie: Aanpassingsvermogen

Software kwaliteit: Onderhoudbaarheid

Probleem: Replace include_once/include with namespace import mechanism

Beschrijving: include en include_once zijn manieren om externe bestanden toe te voegen in PHP. Hoewel ze nog steeds functioneel zijn, sluiten ze niet aan bij moderne, objectgeoriënteerde programmeerpraktijken. Het gebruik van include/include_once biedt niet onmiddellijk duidelijkheid over welke klassen, functies of constanten uit het opgenomen bestand worden gebruikt. Vooral bij grotere projecten kan dit tot verwarring leiden. Moderne PHP-applicaties maken vaak gebruik van Composer en de autoloading-mogelijkheden ervan, die goed werken met het importeren van namespaces. Autoloaders kunnen indien nodig automatisch bestanden toevoegen, waardoor include/include_once-instructies overbodig en minder efficiënt worden.

Mogelijke oplossing: Zorg ervoor dat PHP-klassen binnen namespaces zijn gedefinieerd. Dit is een standaard in moderne PHP voor het logisch organiseren van code. In plaats van bestanden handmatig op te nemen, gebruik de use-instructie aan het begin van uw PHP-bestanden om de benodigde klassen, interfaces of eigenschappen te importeren. Als Composer gebruikt wordt, zorg er dan voor dat een autoloader gedefinieerd wordt. Composer zal dan automatisch de benodigde bestanden toevoegen wanneer klassen worden gebruikt, waardoor handmatige opname niet meer nodig is.

Categorie: Verantwoordelijkheid

Software kwaliteit: Beveiliging

Probleem: Compromised Password

Beschrijving: Het opnemen van hard gecodeerde wachtwoorden in broncode of configuratiebestanden vormt een ernstig beveiligingsrisico, aangezien het toegang tot kritieke systemen kan verschaffen aan onbevoegden. Dit brengt de integriteit en vertrouwelijkheid van het systeem in gevaar. Het gebruik van omgevingsvariabelen of veilige configuratiebeheertools om credentials te beheren, beschermt tegen ongeoorloofde toegang en verbetert de algehele beveiligingspostuur van de applicatie.

Mogelijke oplossing: Als credentials moeten worden opgeslagen voor toegang tot databases of andere services, gebruik dan omgevingsvariabelen.

Week 7:

De Ai draait in een eigen container zo de flexibiliteit en schaalbaarheid van de AI gemakkelijk te kunnen beheren.

itvb23ows-starter-c

Running (3/3)

2.97%

27 minutes ago

:

hive

49eef4579640

itvb23ows-starter-code-fvd-db

Running

1.27%

[3306:3306](#)

27 minutes ago

:

app

925da70dfa83

itvb23ows-starter-code-fvd-app

Running

0%

[80:80](#)

27 minutes ago

:

ai

e12c328a65c0

itvb23ows-starter-code-fvd-ai

Running

1.7%

[5000:5000](#)

27 minutes ago

:

ai

itvb23ows-starter-code-fvd-ai

e12c328a65c0

[5000:5000](#)

STATUS

Running (26 minutes ago)

Logs

Inspect

Bind mounts

Exec

Files

Stats

2024-03-29 13:22:43 * Running on http://172.18.0.3:5000

2024-03-29 13:22:43 Press CTRL+C to quit

2024-03-29 13:22:43 * Restarting with stat

2024-03-29 13:22:43 * Debugger is active!

2024-03-29 13:22:43 * Debugger PIN: 463-849-261

2024-03-29 13:30:17 172.18.0.1 - - [29/Mar/2024 12:30:17] "POST / HTTP/1.1" 200 -

2024-03-29 13:30:25 172.18.0.1 - - [29/Mar/2024 12:30:25] "POST / HTTP/1.1" 200 -

2024-03-29 13:33:26 WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.

2024-03-29 13:33:26 * Running on all addresses (0.0.0.0)

2024-03-29 13:33:26 * Running on http://127.0.0.1:5000

2024-03-29 13:33:26 * Running on http://172.18.0.2:5000

2024-03-29 13:33:26 Press CTRL+C to quit

2024-03-29 13:33:26 * Restarting with stat

2024-03-29 13:33:26 * Debugger is active!

2024-03-29 13:33:26 * Debugger PIN: 526-987-832

2024-03-29 13:37:02 172.18.0.4 - - [29/Mar/2024 12:37:02] "POST / HTTP/1.1" 200 -

2024-03-29 20:20:50 WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.

2024-03-29 20:20:50 * Running on all addresses (0.0.0.0)

2024-03-29 20:20:50 * Running on http://127.0.0.1:5000

2024-03-29 20:20:50 * Running on http://172.18.0.3:5000

2024-03-29 20:20:50 Press CTRL+C to quit

2024-03-29 20:20:50 * Restarting with stat

2024-03-29 20:20:50 * Debugger is active!

2024-03-29 20:20:50 * Debugger PIN: 463-849-261

2024-03-29 13:22:43 * Serving Flask app 'app'

2024-03-29 13:22:43 * Debug mode: on

2024-03-29 13:33:26 * Serving Flask app 'app'

2024-03-29 13:33:26 * Debug mode: on

2024-03-29 20:20:50 * Serving Flask app 'app'

2024-03-29 20:20:50 * Debug mode: on

RAM 7.61 GB

CPU 0.94%

Slashed in

v4.28.0