

Ik heb ervoor gekozen om voor dit project gebruik te maken van de GitHub Flow branching strategie. De redenen dat ik voor GitHub Flow gekozen heb zijn als volgt:

Eenvoudig: GitHub Flow is eenvoudig te begrijpen en te implementeren. Het bestaat uit slechts een paar basisregels, zoals het maken van branches voor nieuwe features en het doen van pull requests om ze samen te voegen.

Continue levering: GitHub Flow is gericht op continue levering, wat betekent dat wijzigingen regelmatig en snel naar productie kunnen worden gebracht. Dit past goed bij het OWSproject omdat we CD/CI moeten leren en met deze aanpak is het iets makkelijker.

Flexibiliteit: GitHub Flow is flexibel. Het moedigt aan tot het maken van kleine, zelfstandige features en het regelmatig samenbrengen van code.

Snelheid: GitHub Flow zorgt voor een snelle ontwikkeling door de nadruk te leggen op kleine en behapbare taken. Het vermijdt long-lived branches en complexe merge-strategieën.

Nu dat ik de voordelen van GitHub Flow heb besproken, is het belangrijk om kort uit te leggen hoe deze strategie werkt:

- **Maak een branch:** Begin met het maken van een nieuwe branch voor elke feature of bugfix. De naam van de branch moet beschrijvend zijn voor wat er ontwikkeld wordt.
- **Voer werk uit op de branch:** Schrijf en commit alle codewijzigingen op deze branch. Dit zorgt ervoor dat de main branch schoon blijft.
- **Open een pull request:** Wanneer de feature of bugfix klaar is, open je een pull request (PR) om de wijzigingen te bespreken en te beoordelen. Dit biedt een kans voor teamleden om feedback te geven en eventuele problemen te identificeren.
- **Code review en testen:** De code wordt beoordeeld en getest. Eventuele opmerkingen worden verwerkt door aanvullende commits toe te voegen aan de branch.
- **Merging:** Zodra de PR is goedgekeurd en alle tests slagen, wordt de branch samengevoegd met de main branch. Dit kan automatisch gebeuren via de PR-interface van GitHub.
- **Deployment:** Na het mergen wordt de code automatisch of handmatig naar productie gepusht, afhankelijk van de ingestelde continuous deployment (CD) pipeline.

Door deze stappen te volgen, blijft de codebase schoon en up-to-date, en kunnen nieuwe features snel en efficiënt worden geïmplementeerd.