

## Inhoudsopgave

1	Gitkraken .....	3
2	Enkele kernwoorden die belangrijk zijn voor configuratie .....	3
2.1	Repository .....	3
2.2	Remote .....	3
2.3	Branches .....	3
2.4	Merging .....	3
2.5	Commits.....	4
2.6	Pushing .....	4
2.7	Fetching/pulling.....	4
3	Github en IntelliJ.....	5
3.1	Registreren van een bestaande account.....	5
3.2	Project toevoegen aan GitHub .....	7
3.3	GitHub .....	9
3.4	Repositorie verplaatsen op GitHub .....	10
4	Gitkraken .....	12
4.1	Integratie GitKraken .....	12
4.2	Connectie GitHub met GitKraken .....	13
4.3	Nieuwe repo aan GitHub toevoegen via GitKraken .....	17
4.4	Bestaande repo van GitHub toevoegen aan GitKraken .....	19
4.4.1	Toolbar.....	21
4.4.2	Left Panel.....	22
4.4.3	Commit Panel .....	22
5	Project in IntelliJ aanpassen.....	23
6	Aanmaken nieuwe branch GitKraken.....	26
7	Push en Pull GitKraken .....	29
7.1	Push .....	30
7.2	Pull .....	31
7.2.1	Voorbeeld fetch all .....	32
7.2.2	Voorbeeld Fast Forward if possible .....	33
8	Mergen van branches GitKraken.....	34
8.1	Merge .....	34
8.2	Pull request.....	36
9	TimeLines.....	<b>Fout! Bladwijzer niet gedefinieerd.</b>
10	JGlo .....	<b>Fout! Bladwijzer niet gedefinieerd.</b>

10.1	Installatie .....	<b>Fout! Bladwijzer niet gedefinieerd.</b>
10.2	Authenticatie.....	<b>Fout! Bladwijzer niet gedefinieerd.</b>

# 1 Gitkraken

Download de Gitkraken UI van onderstaande website.

[https://www.gitkraken.com/student-resources?utm\\_source=hs\\_automation&utm\\_medium=email&utm\\_content=81350528&\\_hsenc=p2ANqtz-9esg9oxfjk7LrEnpGsY24c5AdIOY\\_Dr9FiS5z\\_x7-xRyTChM3msF\\_9UH3eQrOcOgBCcaEsoMn6guKbLfnL\\_ye2sYn9-odAAazXLRbxGt-eggjicw&\\_hsmi=81350528](https://www.gitkraken.com/student-resources?utm_source=hs_automation&utm_medium=email&utm_content=81350528&_hsenc=p2ANqtz-9esg9oxfjk7LrEnpGsY24c5AdIOY_Dr9FiS5z_x7-xRyTChM3msF_9UH3eQrOcOgBCcaEsoMn6guKbLfnL_ye2sYn9-odAAazXLRbxGt-eggjicw&_hsmi=81350528)

Je wordt dan doorgestuurd naar een andere website die vraagt of je een student bent. In de tekst staat er een link met “[verify that you are a student.](#)” Volg de instructies die daar staan. Je zal je doorgestuurd worden naar de website van GitHub. Registreer je met je student.vives.be email account.

## 2 Enkele kernwoorden die belangrijk zijn voor configuratie

### 2.1 Repository

Een repository bevat alle projectbestanden (inclusief de documentatie) en slaat de revisiegeschiedenis op van elk bestand. Repositories kunnen meerdere bijdragers hebben en kunnen openbaar of privé zijn.

### 2.2 Remote

Een remote in Git is een gemeenschappelijke opslagplaats die alle teamleden gebruiken om hun wijzigingen uit te wisselen. Deze wordt in de meeste gevallen opgeslagen in de hostservice van GitHub of een interne server. Laat deze op ‘origin’ staan. Later kan je dit eventueel nog aanpassen.

### 2.3 Branches

Wanneer je aan een nieuwe toevoeging van je project zal werken (vb inloggen) of een fout uit het project halen, maak je best een nieuwe branch.

Een branch is een pointer naar een specifieke commit in the repository. In principe wordt er een kopie gemaakt van het volledige project.

### 2.4 Merging

Merging is het samenbrengen van twee verschillende branches en combineert deze. Met een merge worden bestanden automatisch samengebracht, tenzij er conflicten zijn tussen bestanden. (vb commits van verschillende branches die dezelfde lijn code aanpassen op een verschillende manier)

## 2.5 Commits

Je lokale repo bestaat uit 3 verschillende trees, die beheerd worden door git.

De eerste is de working directory. Deze working directory bevat al je bestanden.

De tweede is de Index. De index gedraagt zich als tussen stadium.

De derde en laatste is de head. De head verwijst naar de laatst 'commit' die je hebt gedaan.

Een commit doen, betekent definitief toevoegen aan de head. Soms zijn deze veranderingen al toegevoegd aan de repo, maar soms ook niet (als de repo op een andere locatie staat).

## 2.6 Pushing

Een Push zal alle aanpassingen die lokaal gebeurd zijn, beschikbaar maken op de remote (GitHub).

Een push zal iedere nieuwe commit proberen toe te voegen aan de remote branch. Erna zal er een fast-forward gebeuren waarbij de remote up to date zal geplaatst worden met de lokale repo.

## 2.7 Fetching/pulling

Fetching en Pulling gaan updates van de remote repo halen en deze lokaal opslaan.

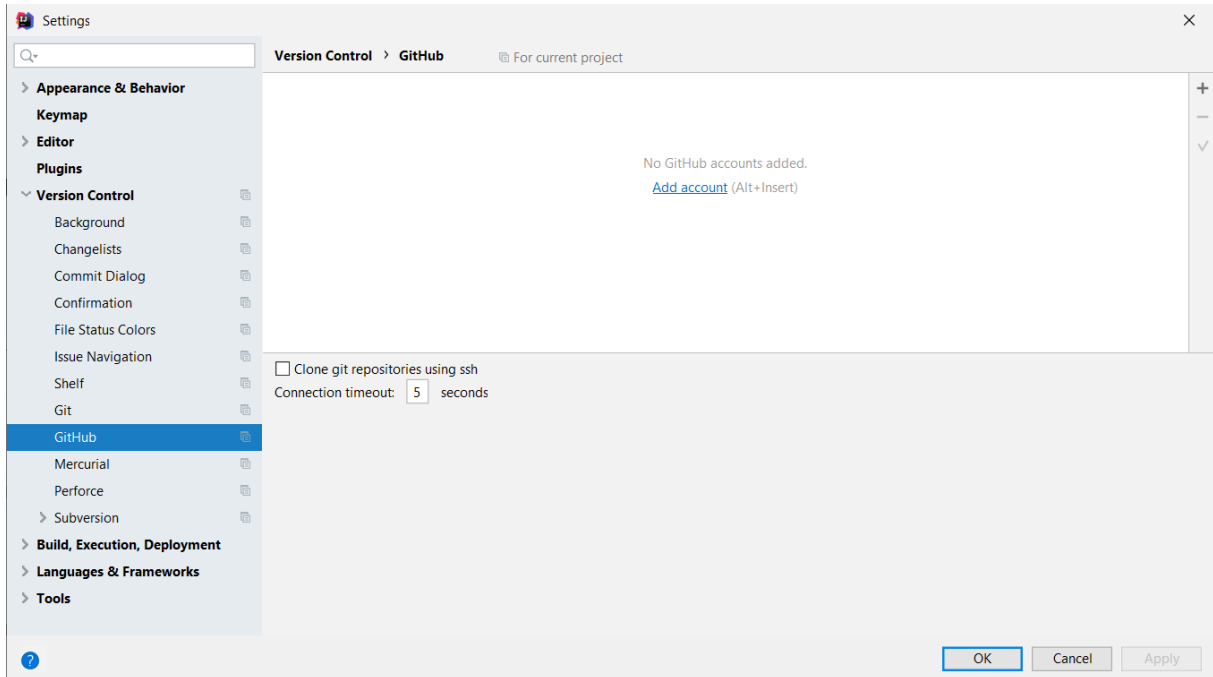
Pulling voert eerst een ophaalactie (fetching) uit en neemt vervolgens eventuele commits in de externe repository op in de lokale kopie.

### 3 Github en IntelliJ

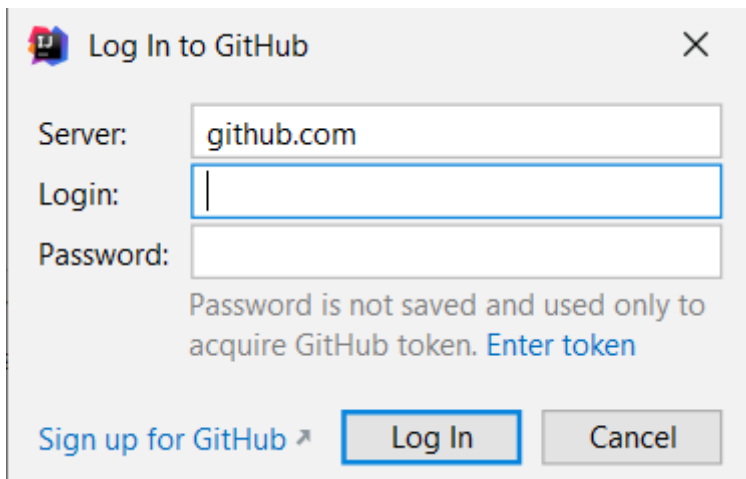
Om bestanden/projecten te ontvangen vanop een repository die op Github staat, of om projecten te delen, moet je je GitHub Account registreren in IntelliJ IDEA.

#### 3.1 Registreren van een bestaande account

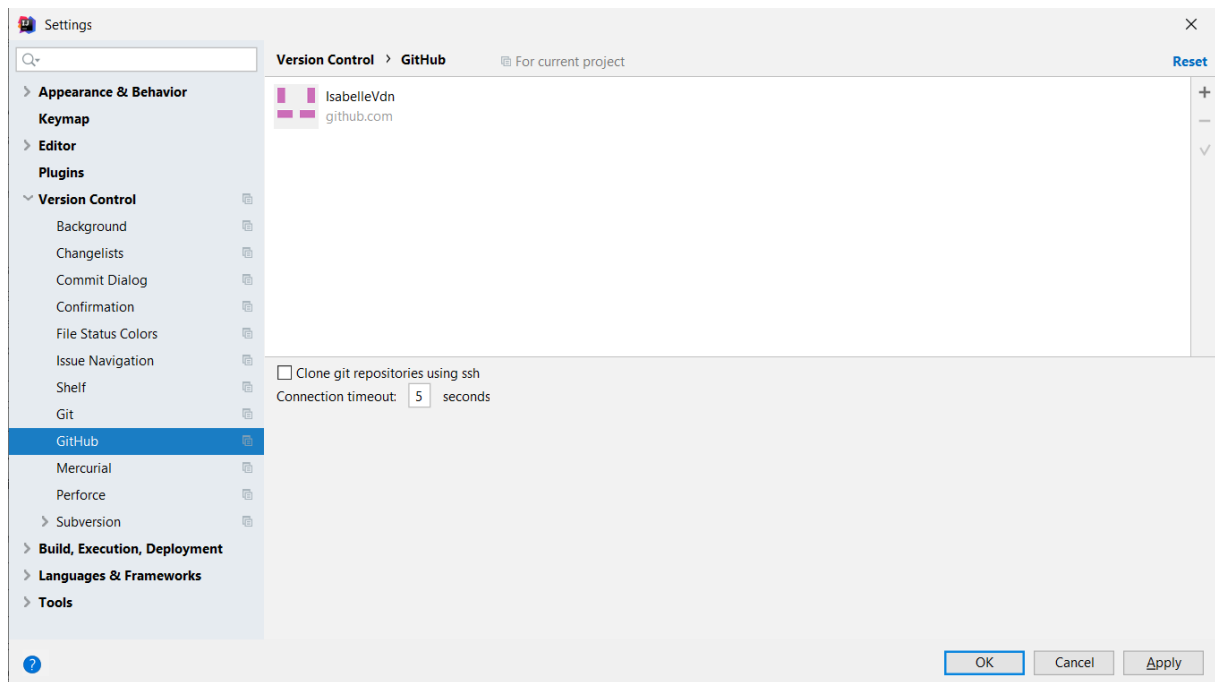
File → Settings/Preferences → Version Control → Github



Klik op Add account



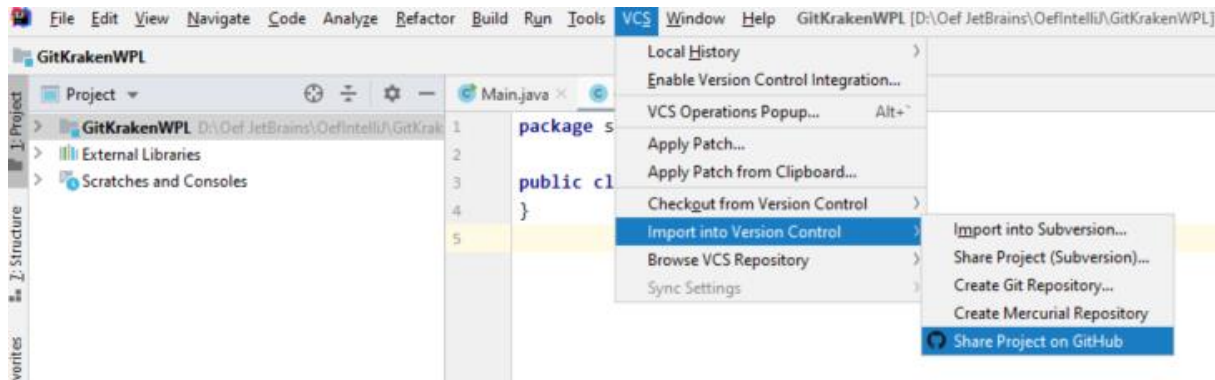
En vul je gegevens in en je account van Github wordt toegevoegd aan IntelliJ



### 3.2 Project toevoegen aan GitHub

Open in IntelliJ het project dat je wilt op GitHub plaatsen

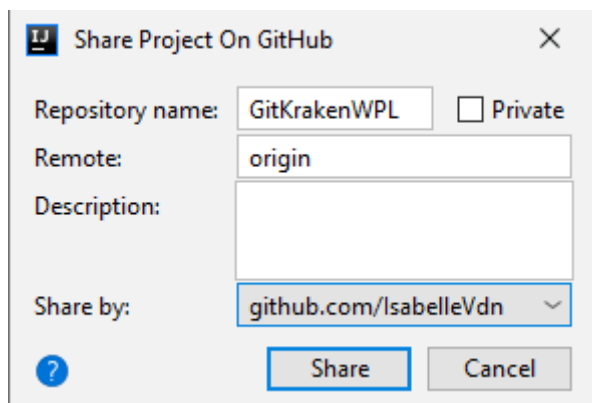
In het menu bovenaan selecteer VCS → Import into Version Control → Share project on Github



Als je je github account al hebt geregistreerd in IntelliJ, dan zal een connectie opgezet worden.

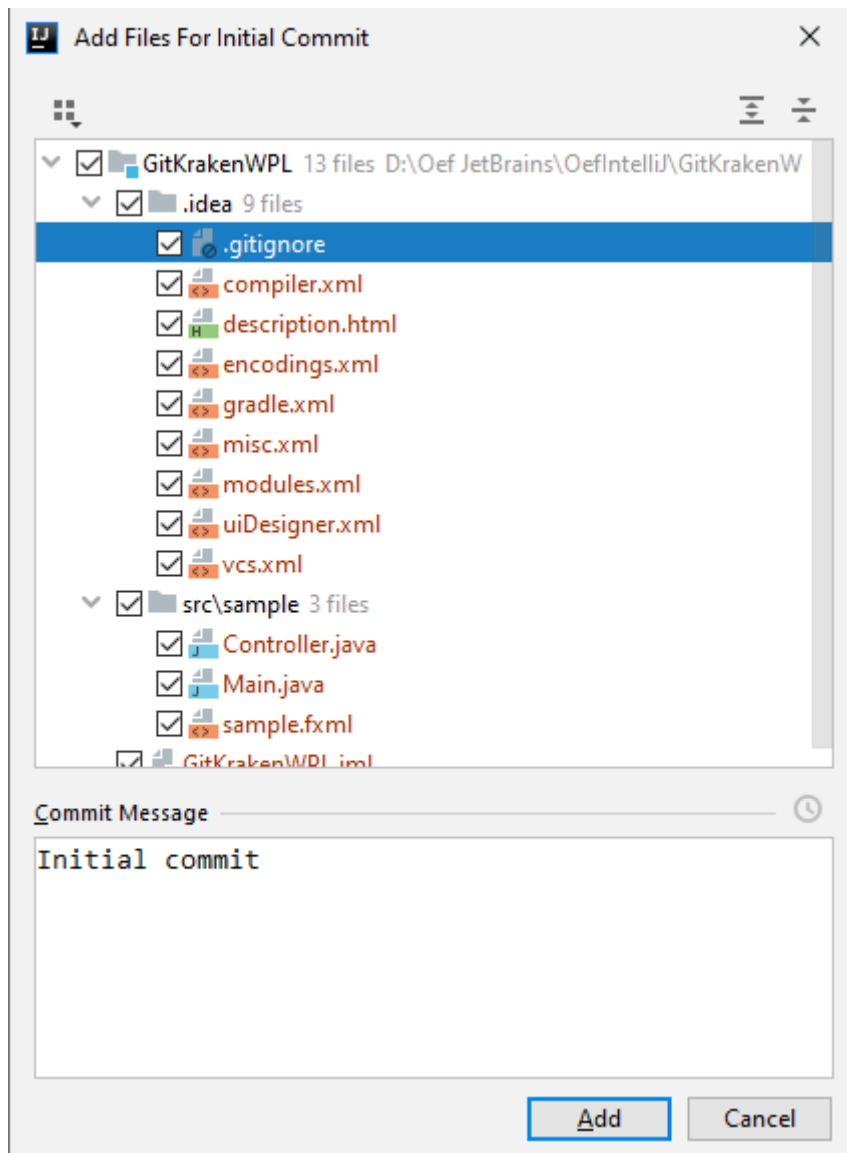
Indien je nog niet ingelogd bent met je GitHub account in IntelliJ, zal de login to Github dialoogvenster geopend worden.

Wanneer de connectie met GitHub opgezet is, zal de “Share Project on Github” venster openen.

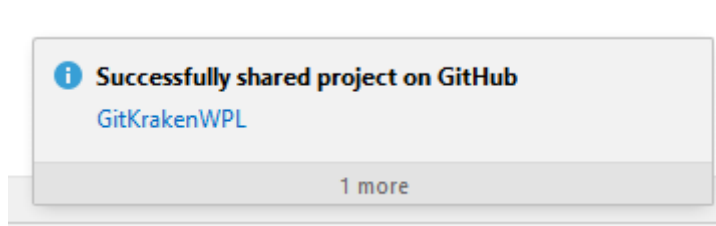


Geef een Repository name op (deze is in principe vrij te kiezen), de naam van de remote (laat de remote staan op origin) en eventueel een beschrijving van het project en klik op Share.

Er zal een extra venster geopend worden met als titel “Add Files For Initial Commit”. Klik hier op “Add” zonder iets aan te passen.



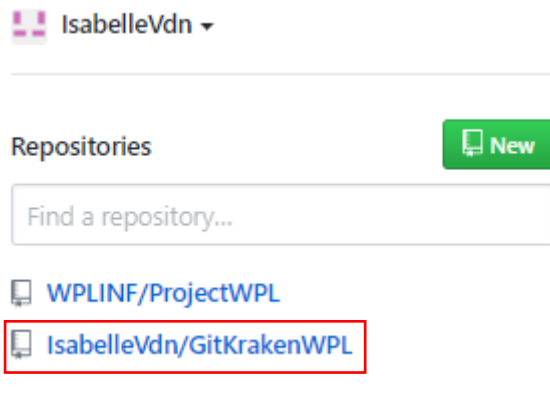
Als alles goed verlopen is, zal je onderaan volgende scherm te zien krijgen



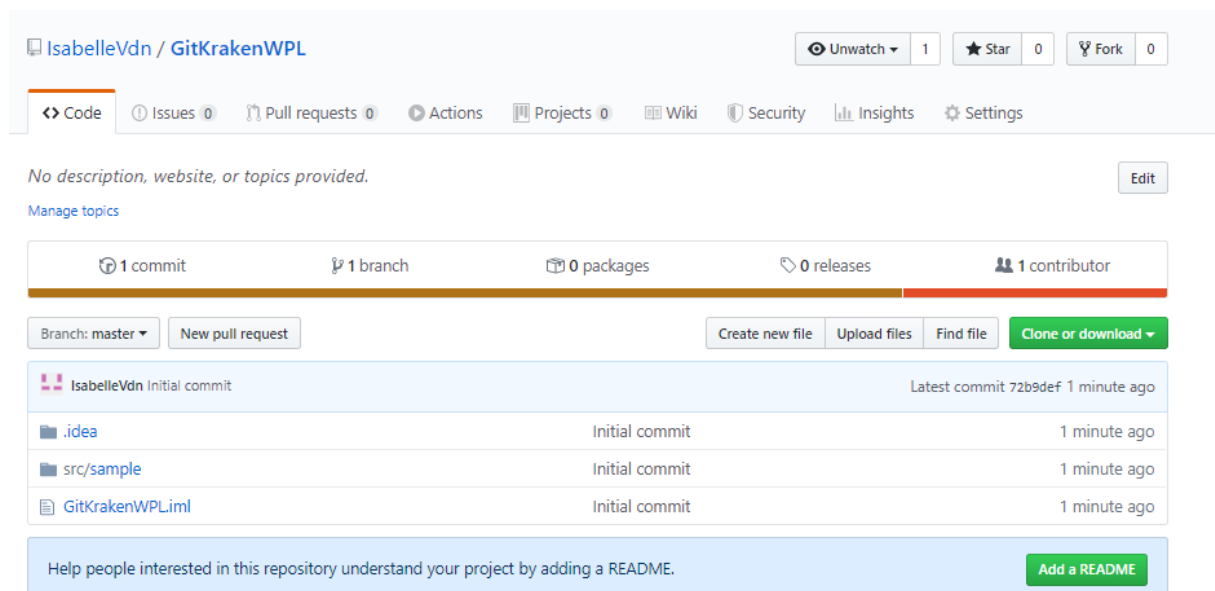


### 3.3 GitHub

Als je zou kijken op je GitHub zal je zien dat de repository is aangemaakt



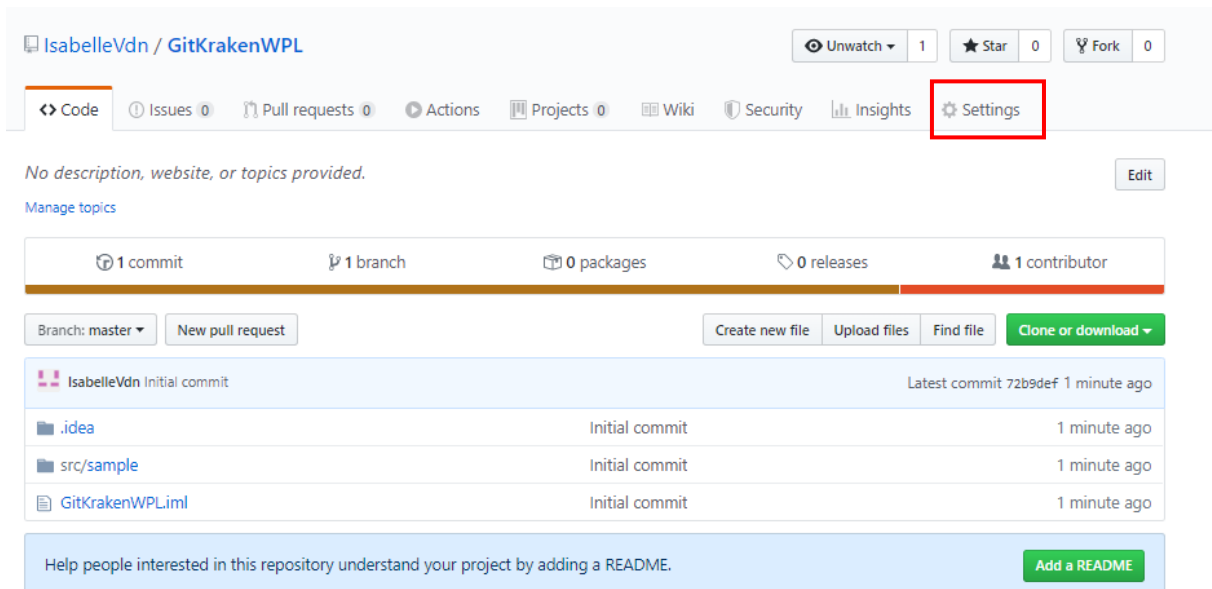
En kan je in detail alle bestanden die aan deze repository verbonden zijn gaan bekijken. Dit doe je door op de naam van de repository te klikken.



### 3.4 Repositorie verplaatsen op GitHub

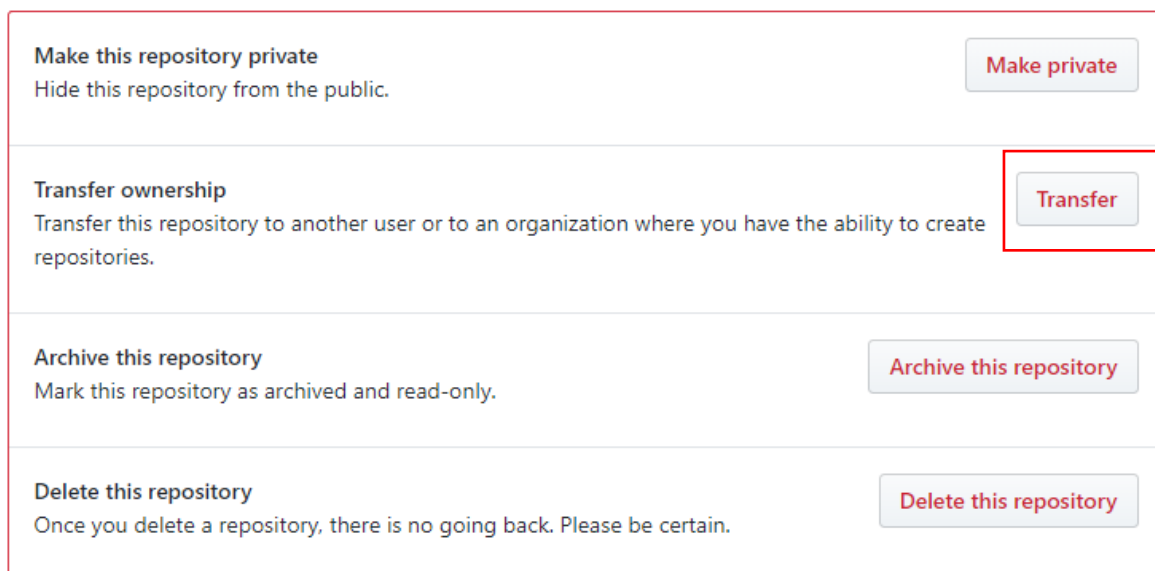
De repositorie dat je hebt aangemaakt, staat onder je eigen naam en niet onder de groepsnaam. Dus moet je de repositorie nog verplaatsen naar de groep.

Ga hiervoor naar onderstaand scherm en klik op 'settings'.



Scroll naar beneden tot je volgend scherm te zien krijgt.

## Danger Zone



Klik op Transfer.

Geef de naam van de huidige repository in en geef de naam in van de organisatie (team) dat je in zit. En klik op “I understand, transfer this repository”.

Transfer repository

To understand admin access, teams, issue assignments, and redirects after a repository is transferred, see ["Transferring a repository"](#) in GitHub Help.

Transferring may be delayed until the new owner approves the transfer.

Type IsabelleVdn/GitKrakenWPL to confirm

IsabelleVdn/GitKrakenWPL

New owner's GitHub username or organization name

Werkpleklerin1920

I understand, transfer this repository.

Bovenaan krijg je onderstaande melding

Moving repository to Werkpleklerin1920/GitKrakenWPL. This may take a few minutes.

Als je na enige tijd naar je groep zou gaan kijken, is het project daar beschikbaar

 Werkpleklerin1920 ▾

[View organization](#)

Repositories



New

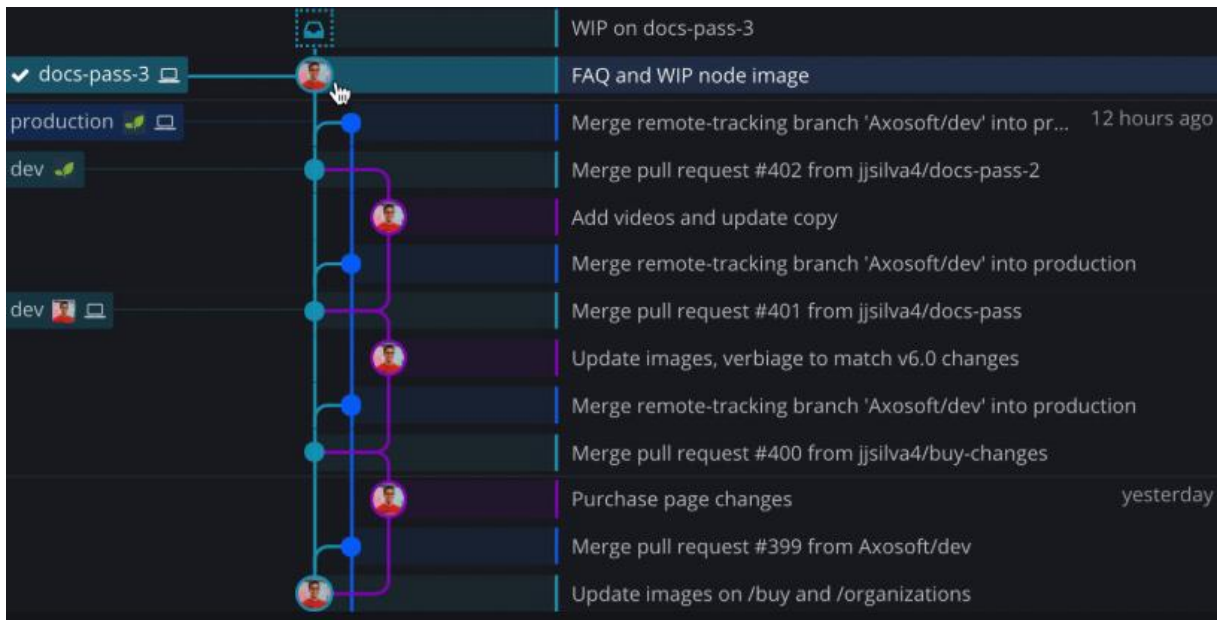
Find a repository...



GitKrakenWPL

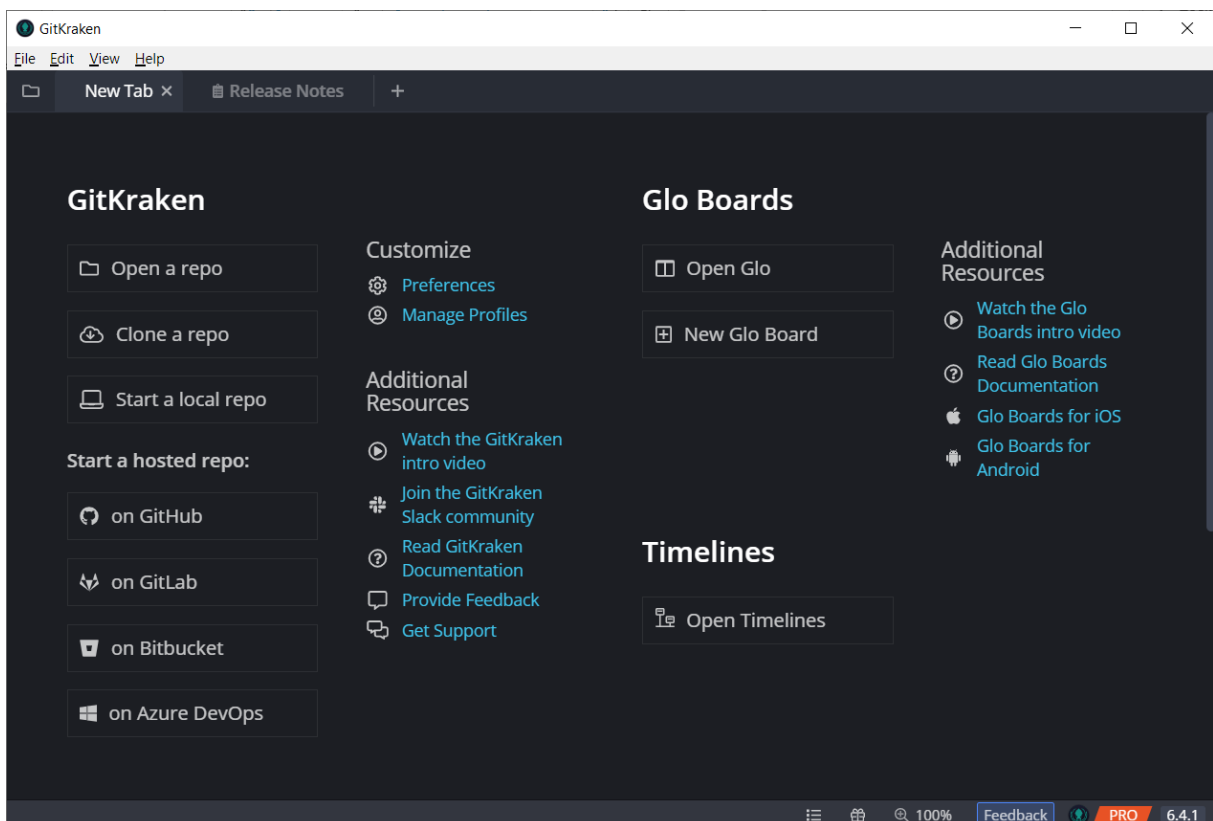
## 4 Gitkraken

Gitkraken is een GUI die visueel voorstelt wat er allemaal op Github gebeurt. Heeft er iemand iets aangepast aan het project, welke bestanden zijn er veranderd, wat is er toegevoegd? Wat is er verwijderd?, ...

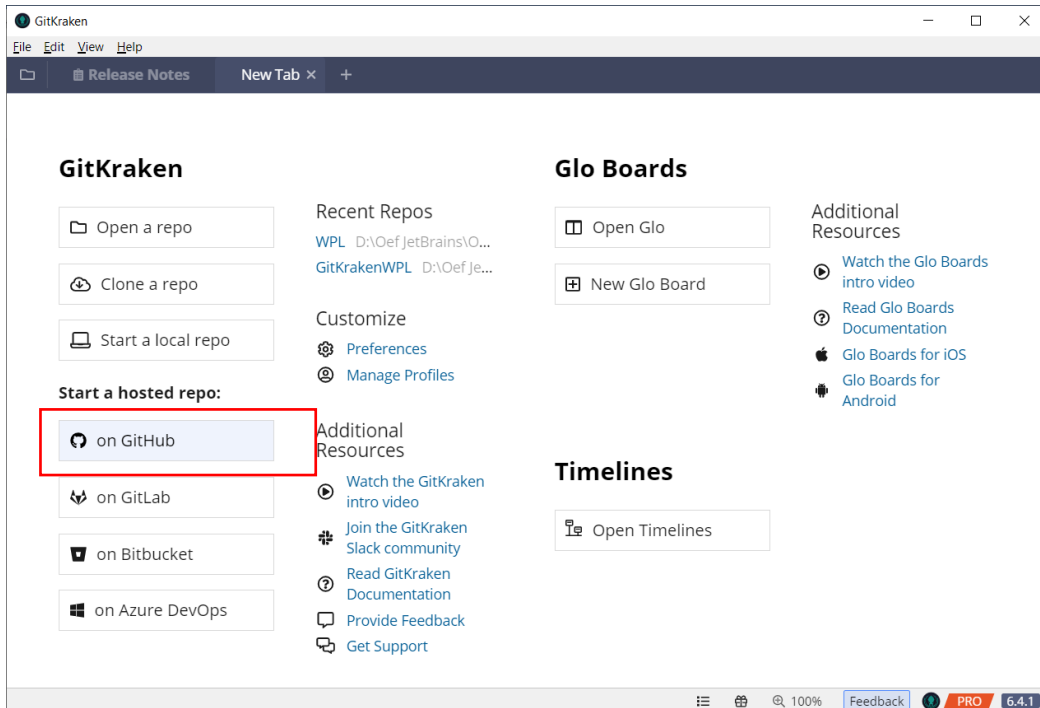


### 4.1 Integratie GitKraken

Start Gitkaken en je krijgt onderstaand scherm te zien. Zorg hierbij dat je Gitkraken met GitHub hebt geconnecteerd.

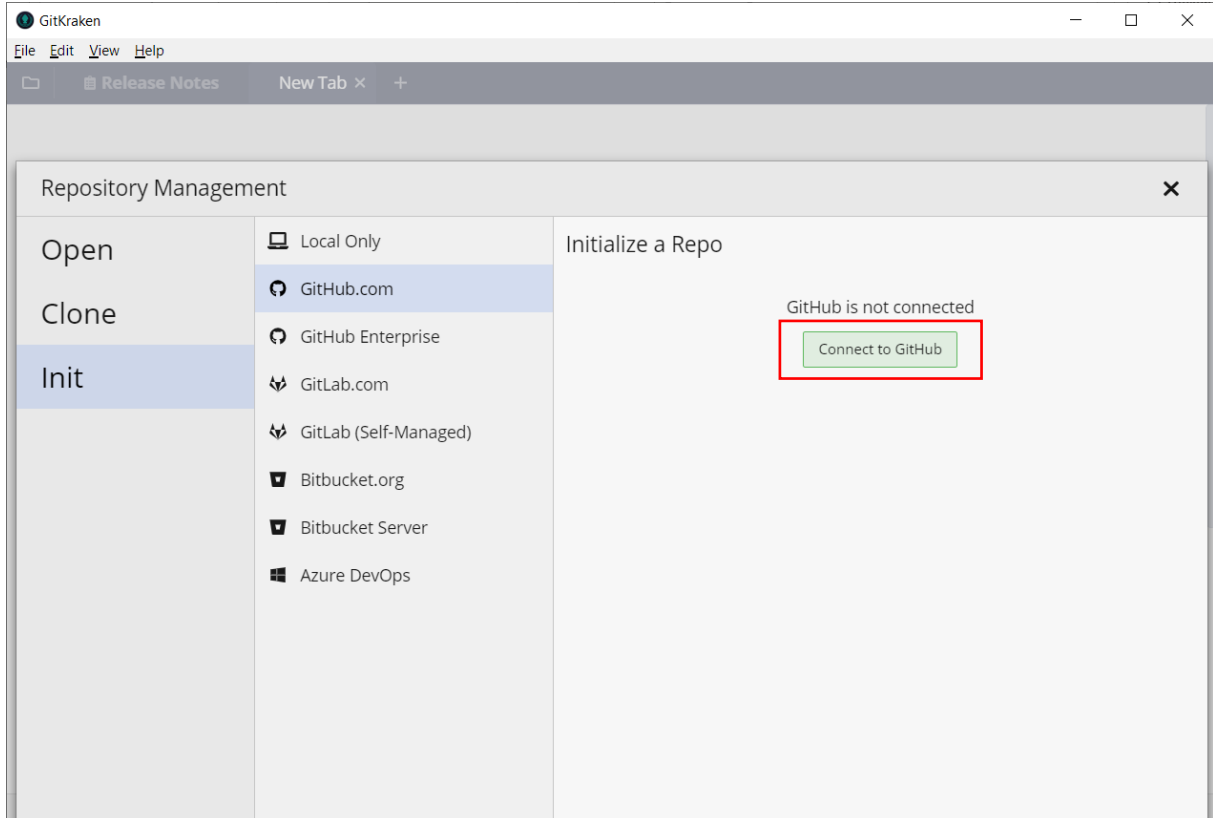


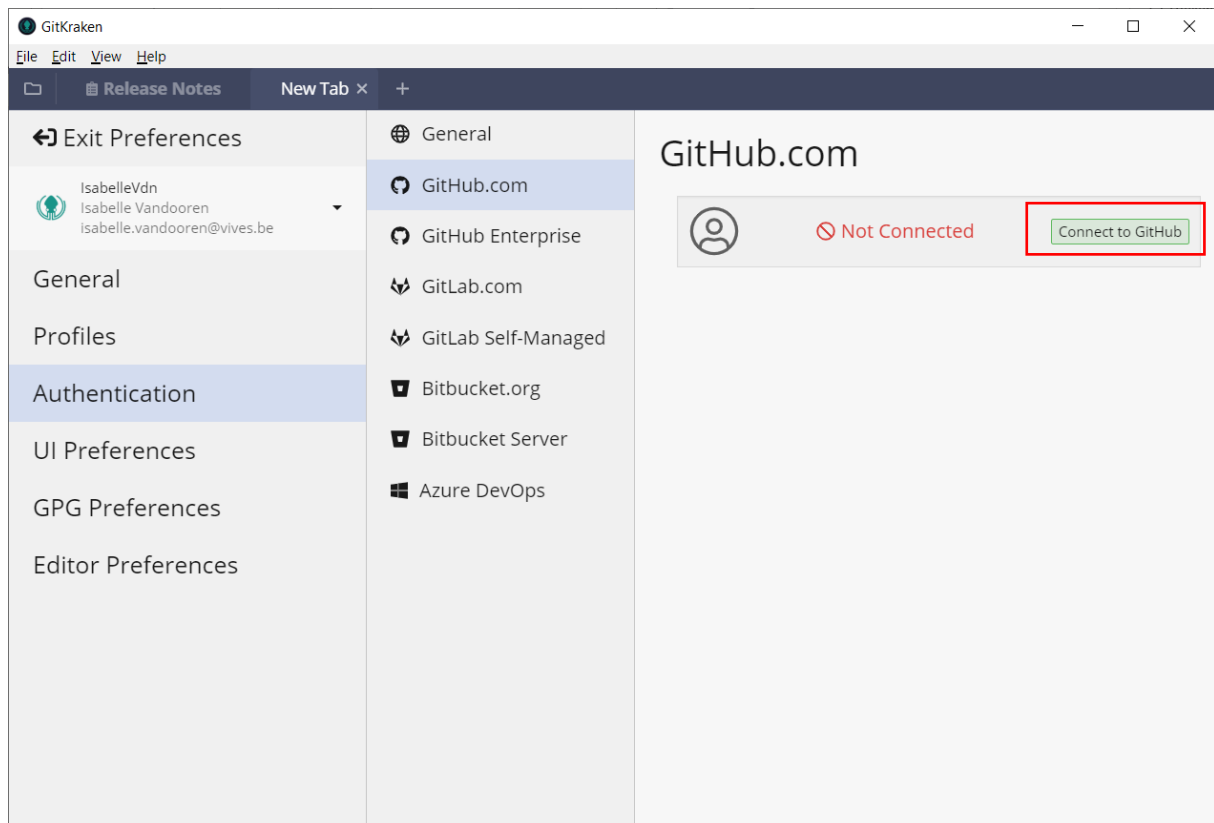
Aangezien wij al onze repo hebben gepost op GitHub kunnen we hier van start gaan met “start a hosted repo” en selecteer ‘on GitHub’.



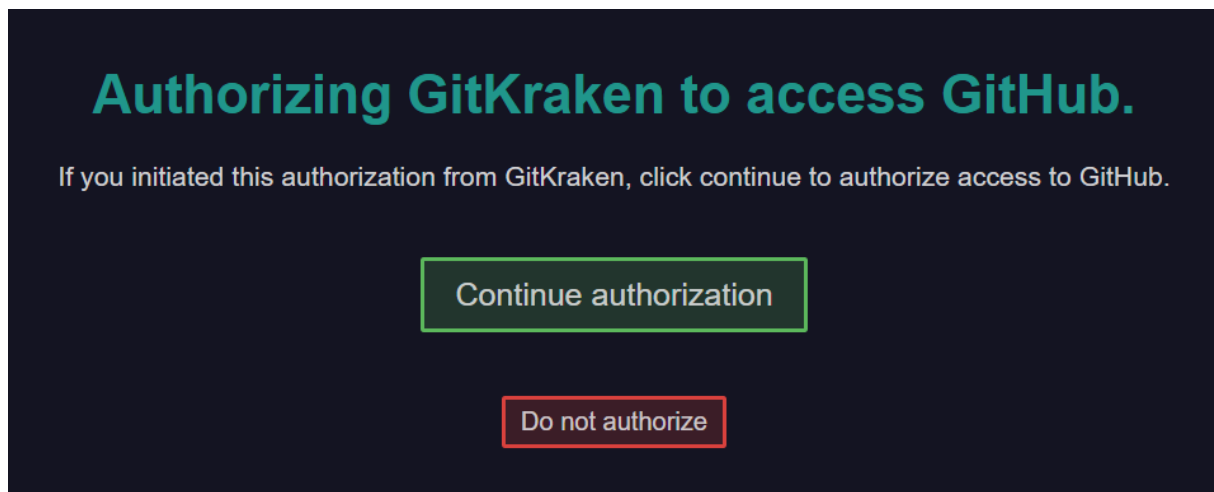
## 4.2 Connectie GitHub met GitKraken

Erna moet je connecteren met GitHub.



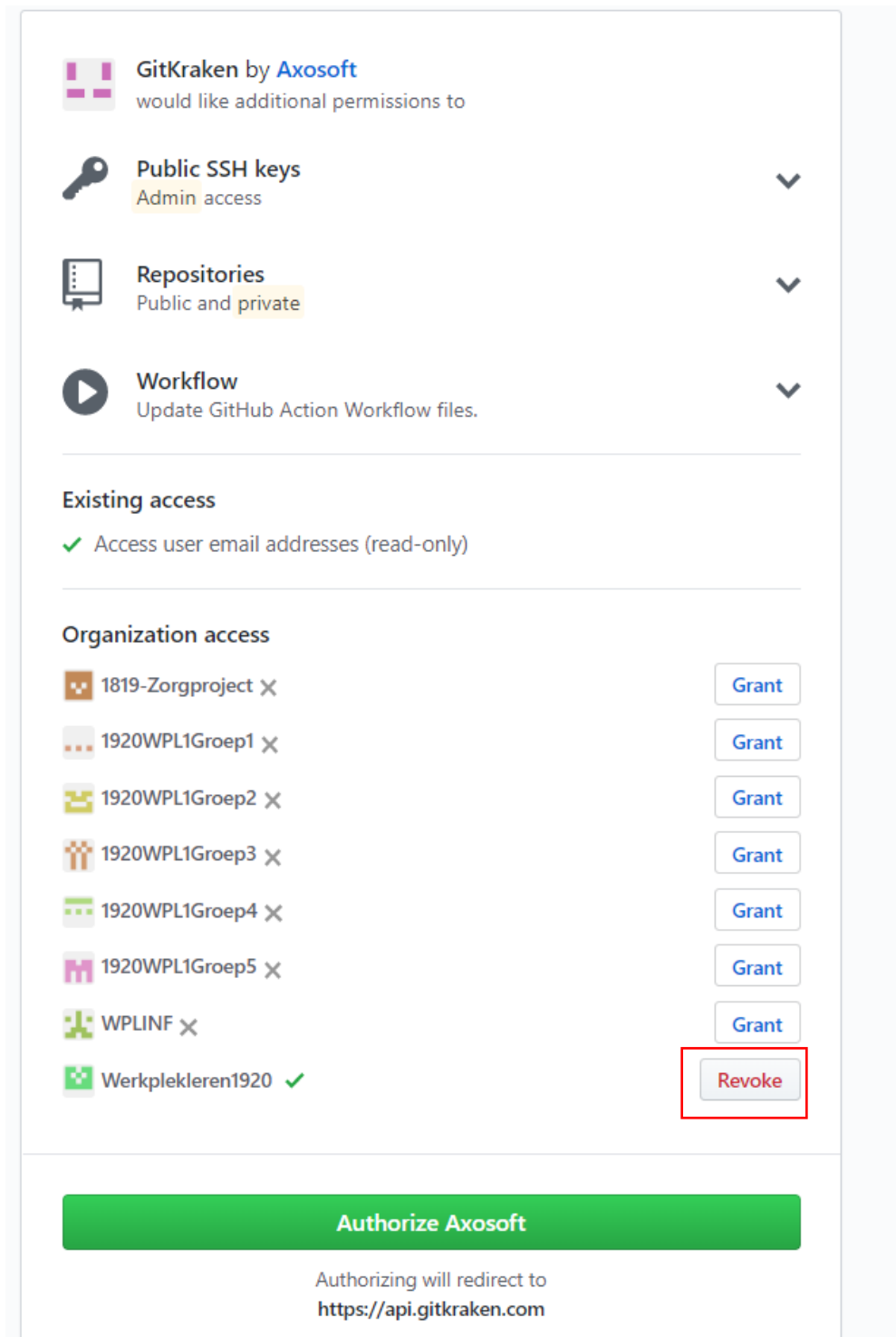


Hierna zal je doorverwezen worden naar een webpagina die graag of GitKraken toegang mag krijgen tot GitHub. Klik hier op 'Continue authorization'.



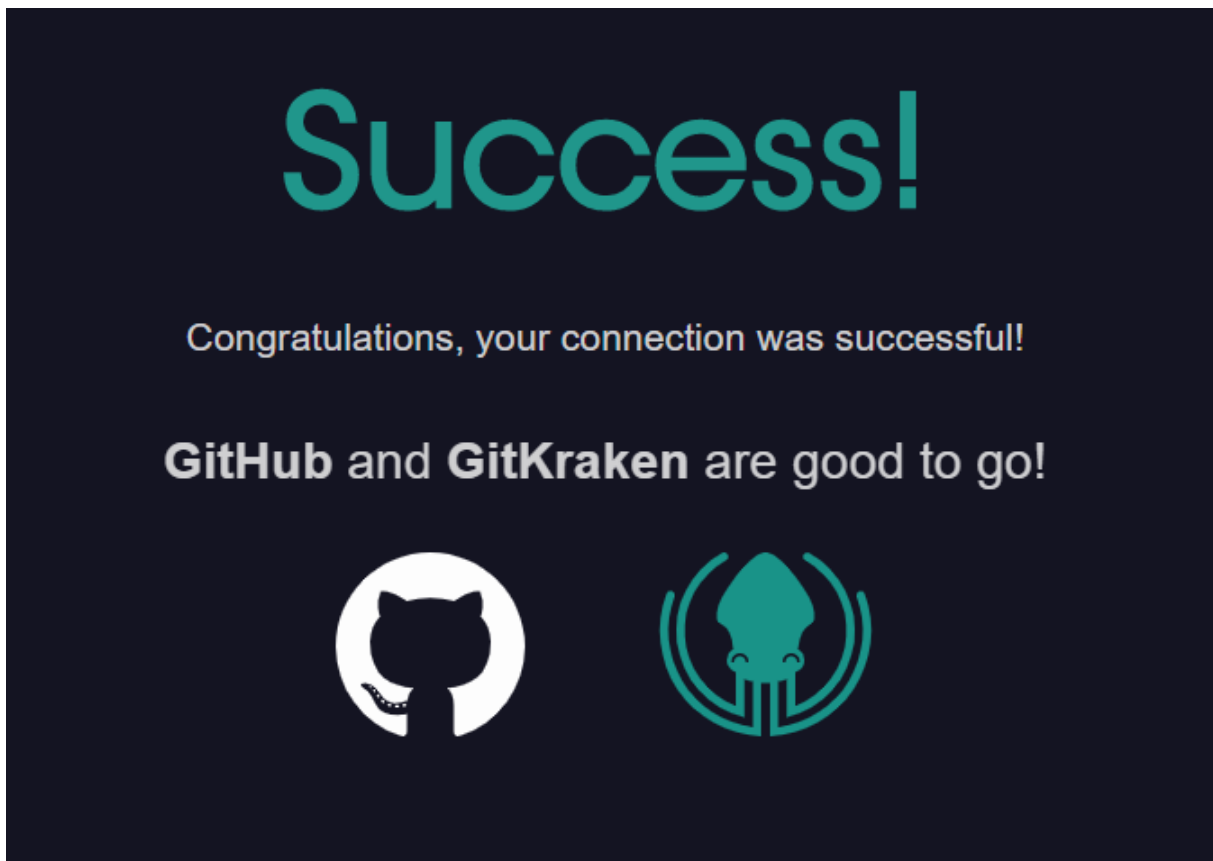
Je zal eventueel doorgestuurd worden naar GitHub en daar kan je zeggen tot welke projecten GitKraken allemaal toegang mag hebben. Dit kan je doen door op Grant te klikken.

Ik heb op onderstaande figuur enkel maar de repo Werkpleklerin1920 toegang gegeven.

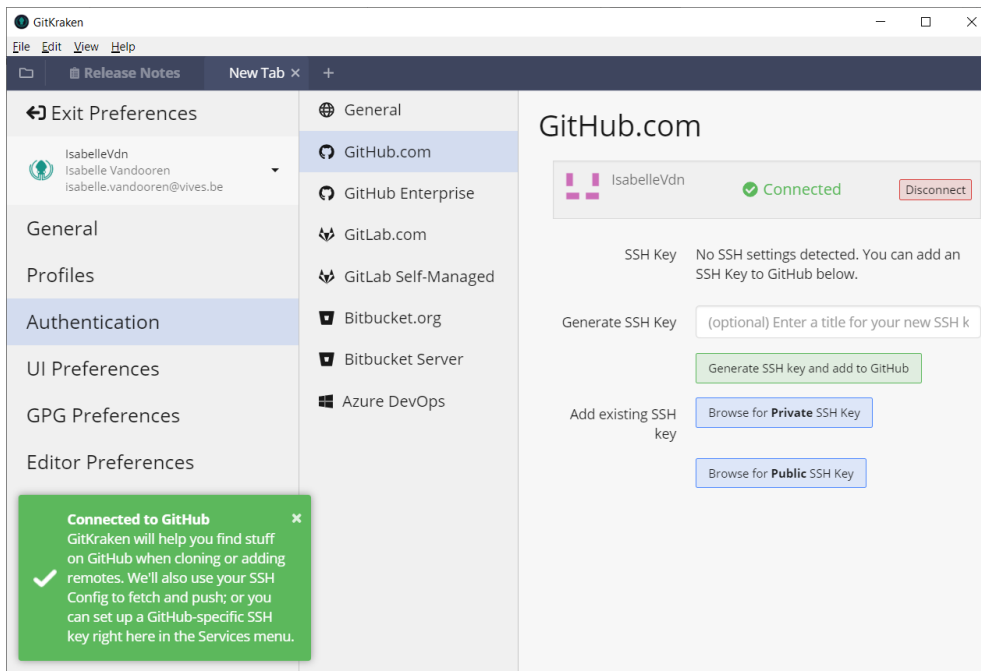


Je kan dit later nog aanpassen op de account van GitHub.

Hierna krijg je de melding dat de connectie tussen GitHub en GitKraken ok is.



Ook in GitKraken zie je dat de connectie ok is en dat je GitHub account gekoppeld is aan GitKraken.

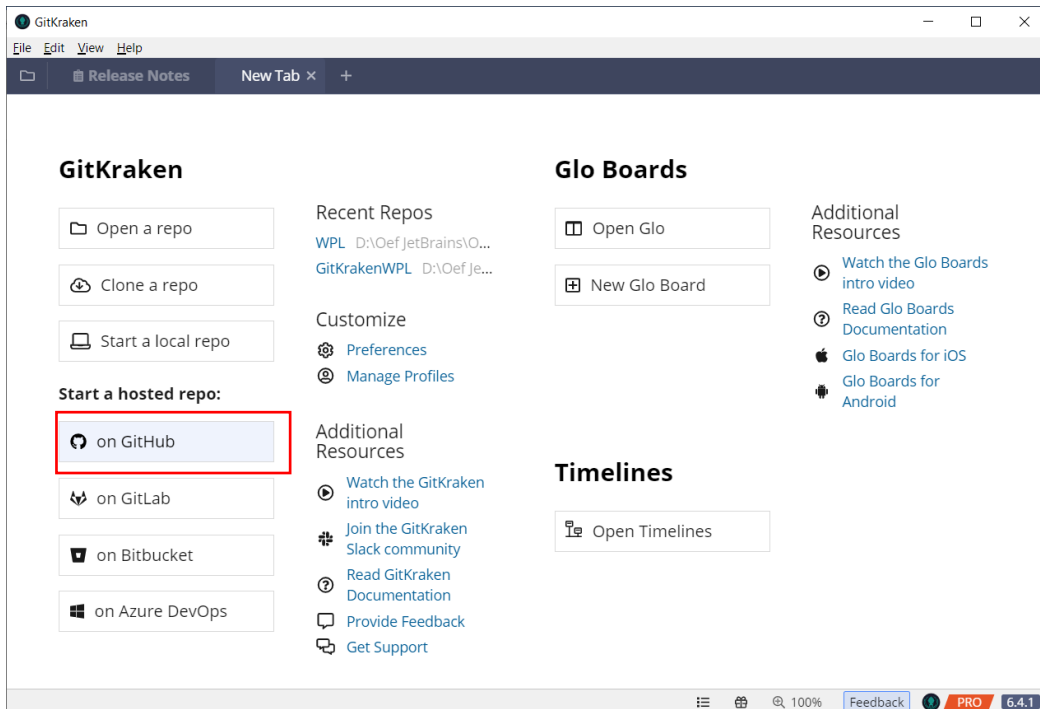


Klik bovenaan op "Exit Preferences" zodat je terug op het startscherm terecht komt.

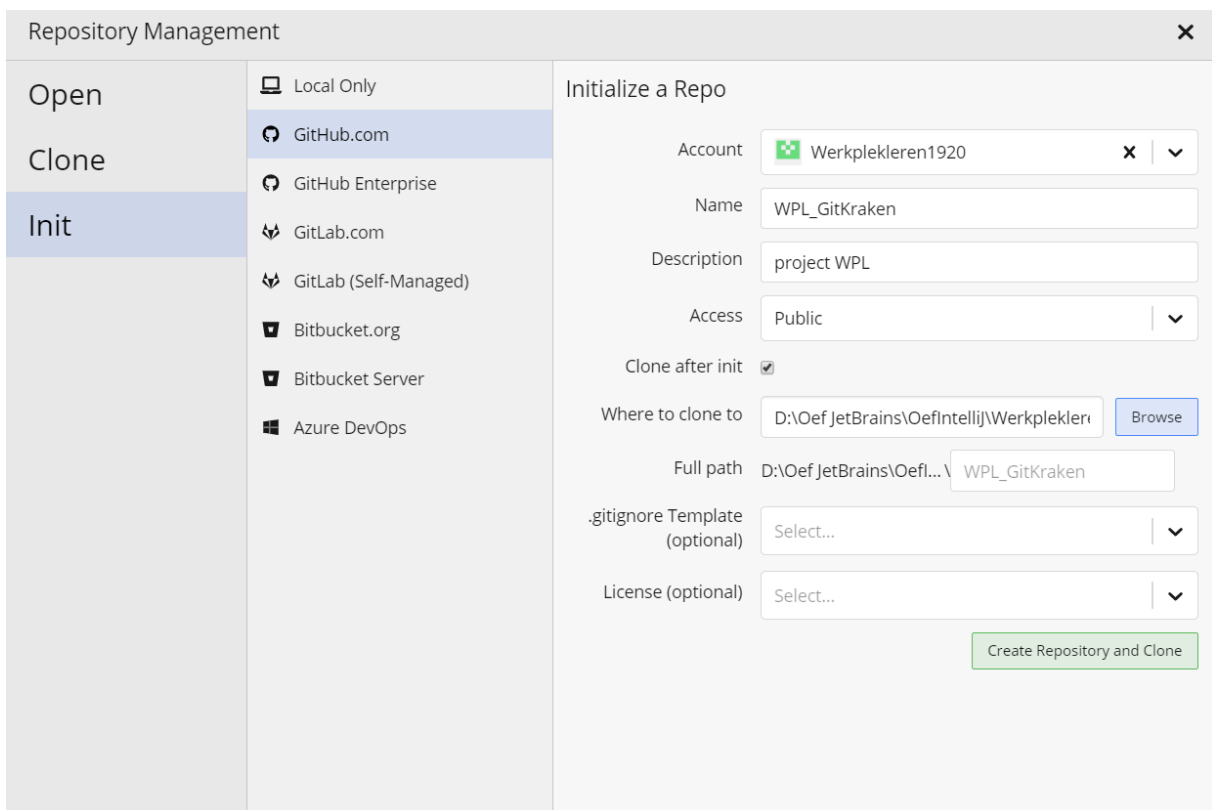


### 4.3 Nieuwe repo aan GitHub toevoegen via GitKraken

Selecteer in het startvenster “Start a hosted repo: on GitHub”



Erna krijg je een scherm met gegevens over de repo dat je wenst aan te maken.

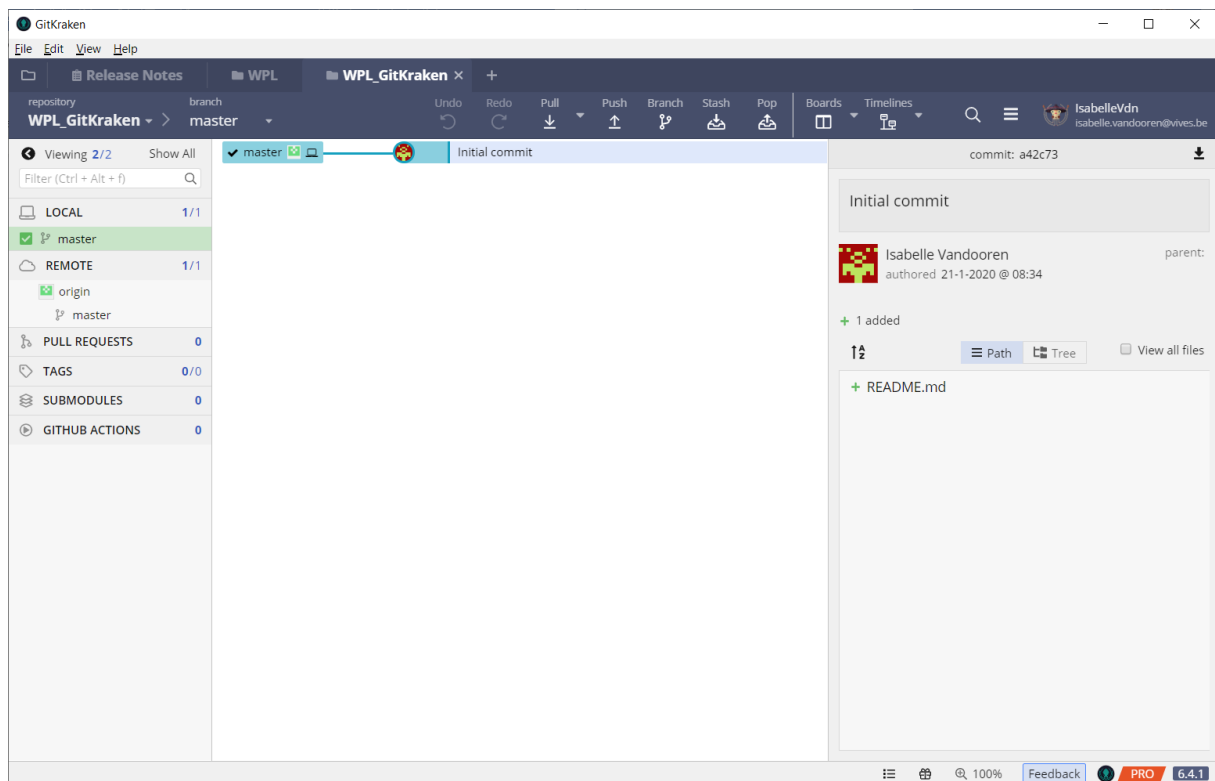


- Als eerste selecteer je de account op welke de repo moet staan.
  - Hier selecteer je de groep account
- Als tweede geef je een naam op

- Deze mag je zelf kiezen
- Description
  - Is extra info, maar mag je ook open laten
- Acces
  - Toegang, plaatst je best op public
- Clone after init
  - Vink je aan. Er wordt direct na het initialiseren een kopie gemaakt van de huidige toestand van het project dat zich nu op GitHub bevindt.
- Where to clone to
  - Waar moet de kopie van het project geplaatst worden (hoef je niet in te vullen, maar handig dat je hier toch een locatie opgeeft)

Klik op “Create repository and Clone”.

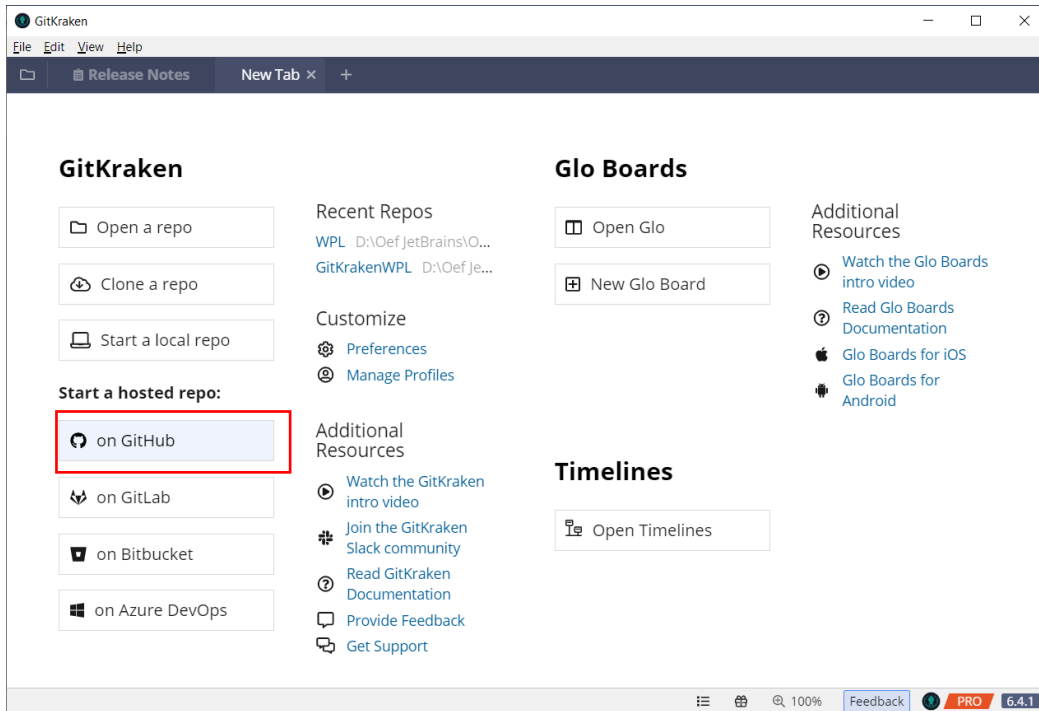
Daarna zie je dat er een nieuwe repo is aangemaakt en enkel één bestand bevat nm README.md



## 4.4 Bestaande repo van GitHub toevoegen aan GitKraken

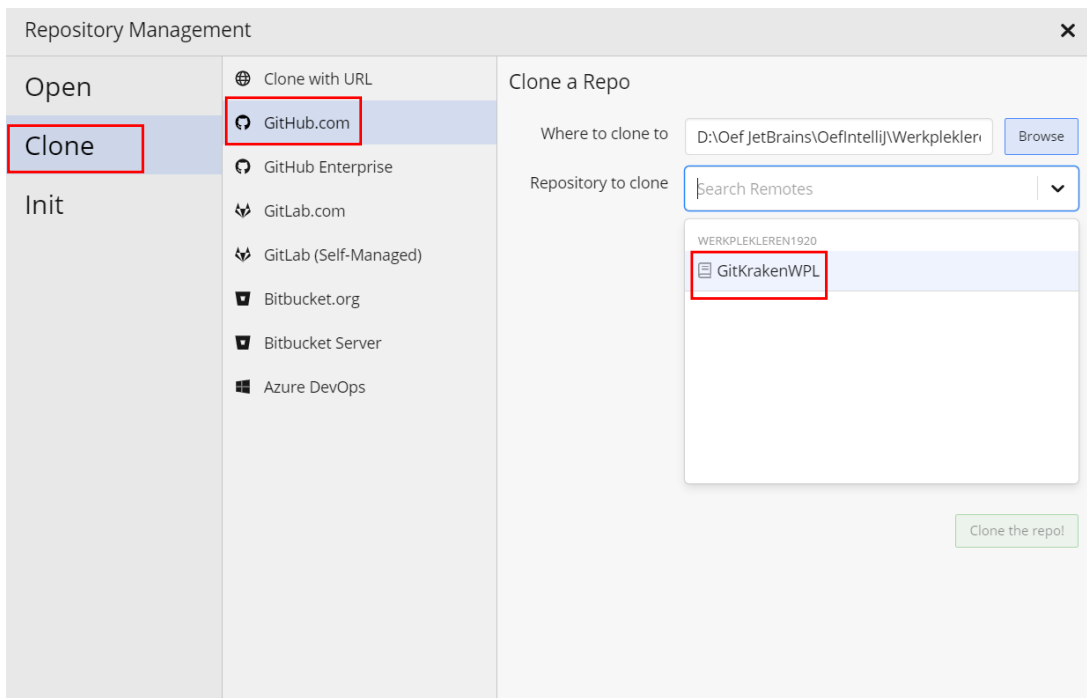
Aangezien we al een bestaande repo hebben, moeten we deze repo toevoegen aan GitKraken. Dit kan je doen via onderstaande

Selecteer in het startvenster “Start a hosted repo: on GitHub”

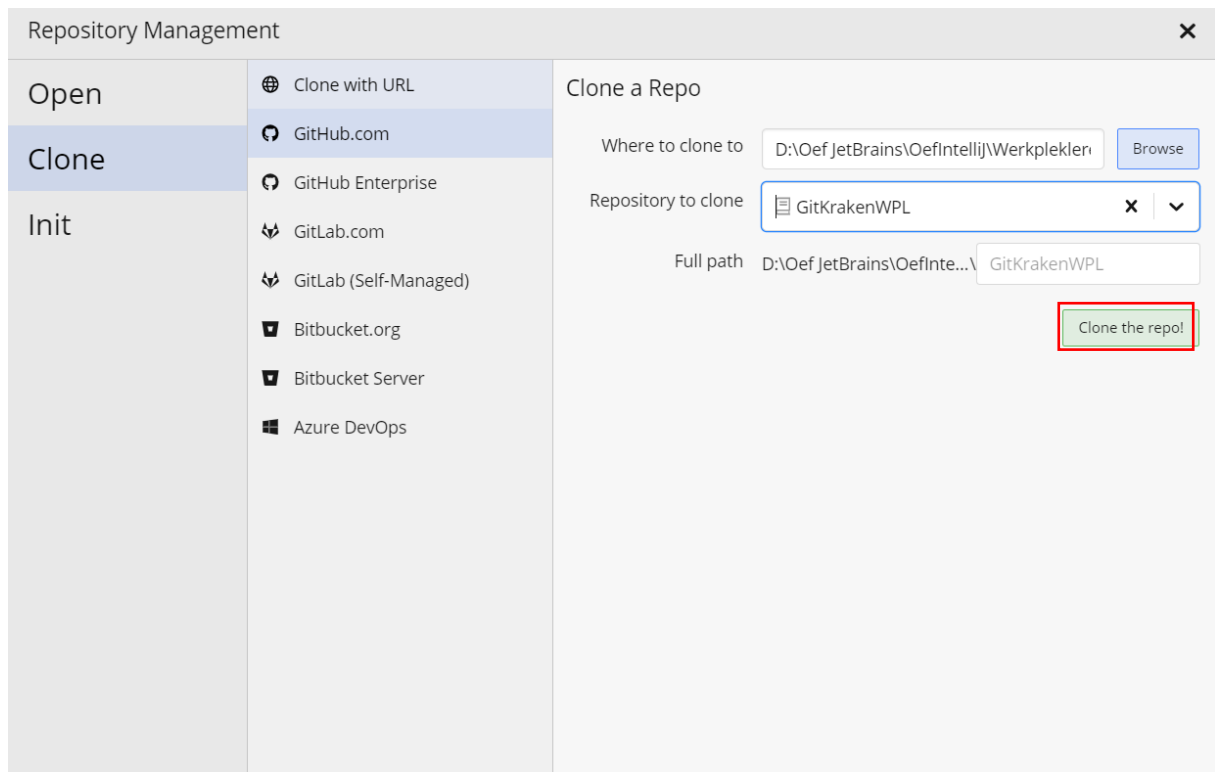


Aangezien we al een repo hebben op GitHub, moeten we daarvan een clone aanmaken in GitKraken. Selecteer Clone → GitHub.com.

Kies waar je de repo wenst lokaal op te slaan, en selecteer de repo.



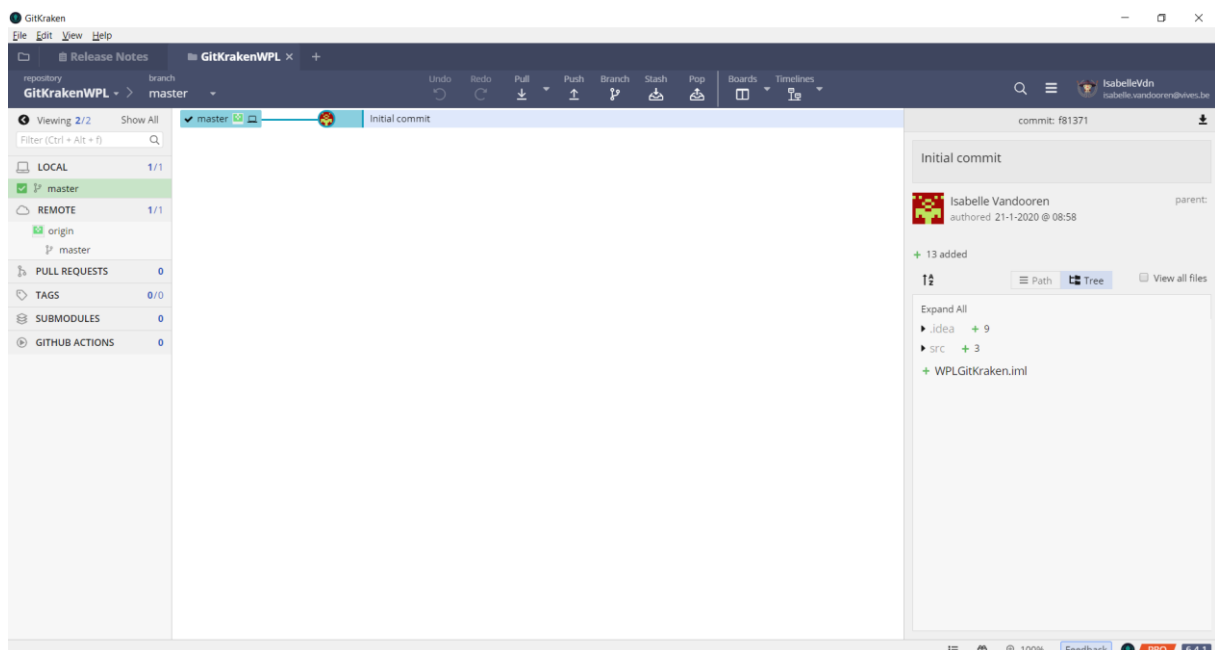
En klik op “Clone the repo”



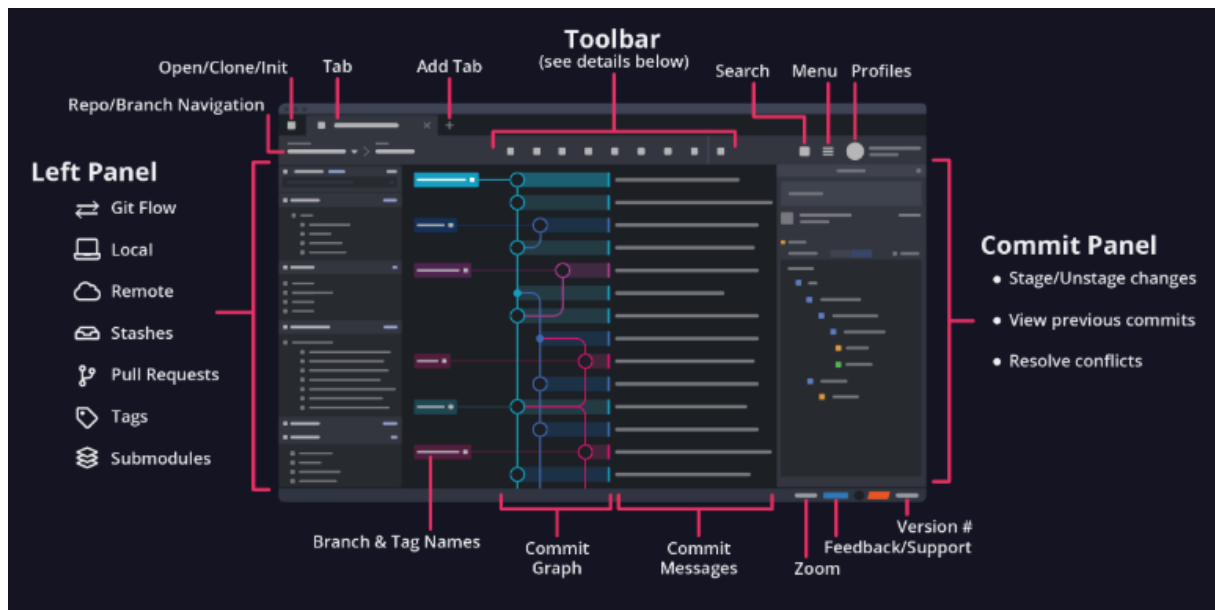
Je zal de melding krijgen dat de repo gecloned is en open deze.



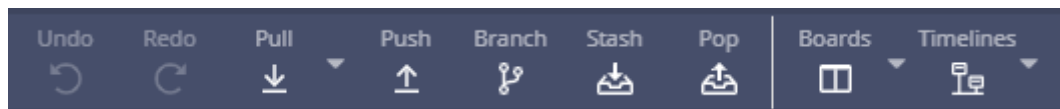
Als je deze opent, zal je zien dat alle bestanden van GitHub zichtbaar zijn in GitKraken.



Hier zie je al verschillende zaken binnen GitKraken.



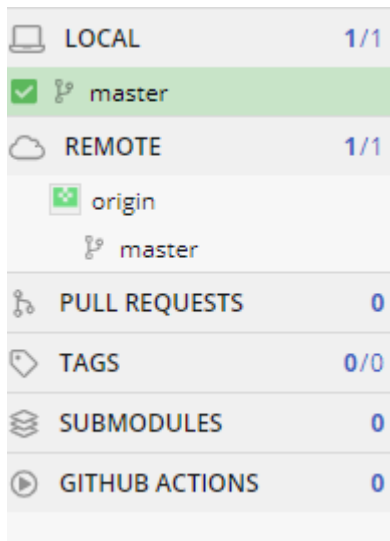
#### 4.4.1 Toolbar



- Undo
  - Ongedaan maken van actie
  - Indien niet mogelijk, ook niet mogelijk om te selecteren
- Redo
  - Ongedaan actie ongedaan maken
  - Indien niet mogelijk, ook niet mogelijk om te selecteren
- Pull
  - Pull veranderen vanop je remote repo
    - Fetch all
    - Pull (fast-forward if possible)
    - Pull (fast-forward only)
    - Pull (rebase)
- Push
  - Push veranderingen naar de remote repo
- Branch
  - Branch maken op huidige lokale repo
- Stash
  - Geef overzicht van werk in progress
- Pop Stash
  - Herstellen van work in progress
- LFS
  - Indien je grote bestanden hebt in je repo, zal deze knop zichtbaar worden
- Boards
  - Traceren van de uit te voeren taken en problemen met Glo Boards
- Timelines

- Visualisatie van de milestones

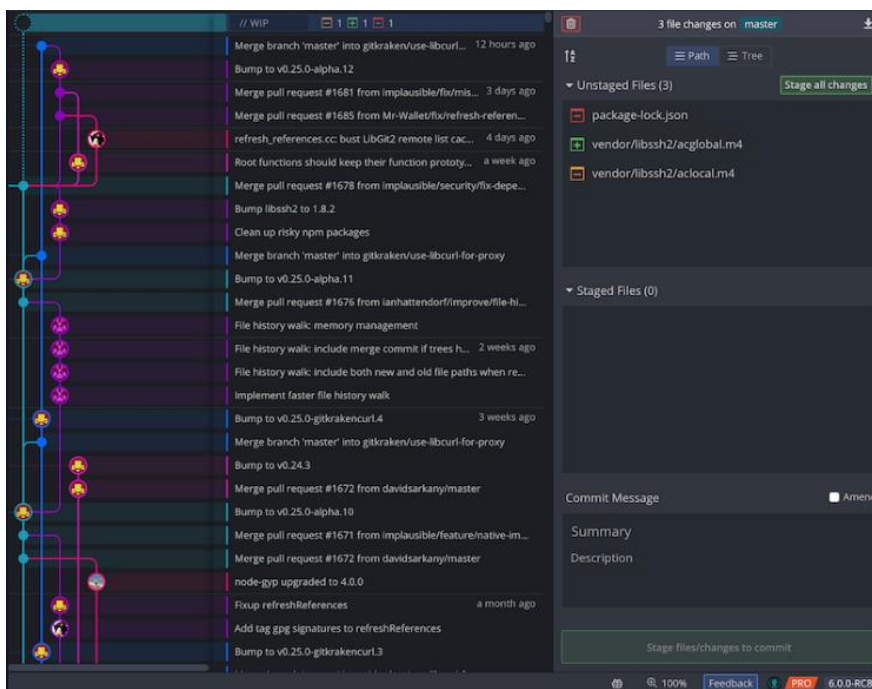
#### 4.4.2 Left Panel



- Local
  - Lokale branches
- Remote
  - Refereert naar de remote branches (branches op GitHub)
- Pull requests
  - Toont de actieve aanvragen om een branch te mergen met een andere branch
- Tags
  - Actieve aanwijzingen voor commit, maar verandert nooit van plaats
- Submodules
  - Een github repo is een subdirectory van de huidige repo
- Github actions
  - Acties die op GitHub moeten uitgevoerd worden

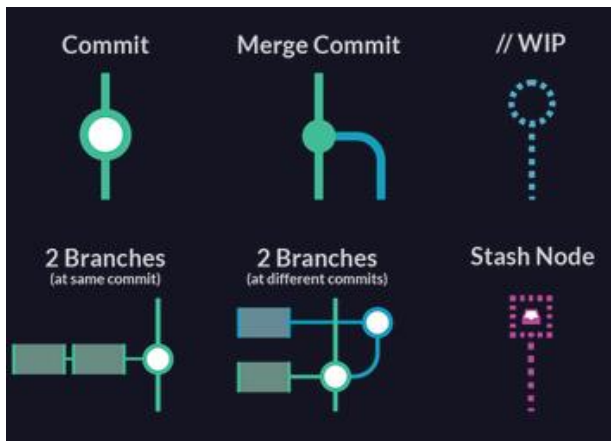
#### 4.4.3 Commit Panel

In het commit panel zie je al bestanden en aanpassingen die gebeurd zijn op je working directory



Links zie je de momenten van de aanpassingen en door wie deze uitgevoerd zijn.

Branches en tag labels zijn pointers naar specifieke commits.



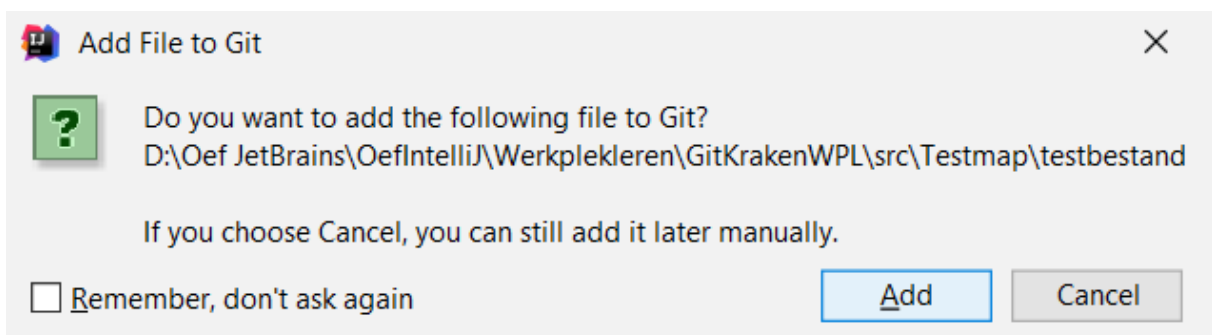
In het midden zie je de commentaar die toegevoegd, dus wat de aanpassingen zijn

Rechts zie je het staging panel

- Unstaged Files
  - Overzicht van alle bestanden die aangepast zijn sinds de vorige commit
    - Aangepast
    - Toegevoegd
    - Verwijderd
    - Aangepast van naam
- Staged Files
  - Files die manueel toegevoegd zijn aan de index en die klaar zijn voor commit
- Commit message
  - Opslaan van aanpassingen aan de repo

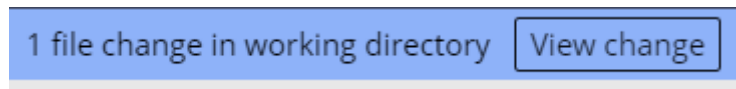
## 5 Project in IntelliJ aanpassen

Indien we nu een map aanmaken in IntelliJ met daarin een bestand zal je de vraag krijgen of je deze aan Git wilt toevoegen.

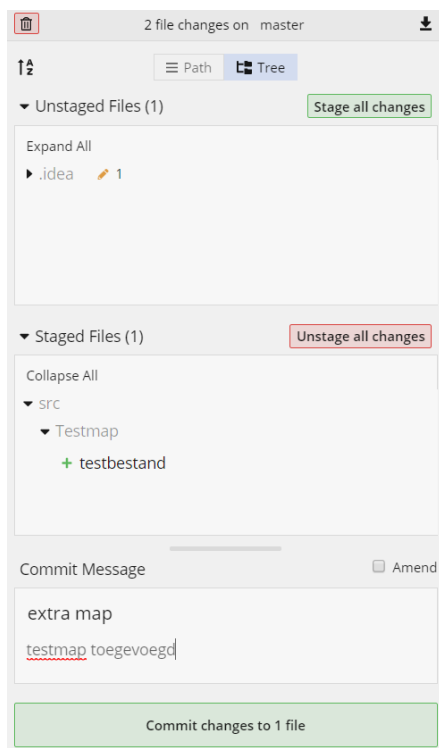


Het best is hier effectief op "Add" te klikken. Daardoor komt het bestand geselecteerd om met een commit mee te gaan.

Doordat GitKraken en IntelliJ op hetzelfde project/repo zitten, zal je zien dat er in GitKraken een melding zal komen.



Klik op 'View Changes' en er wordt een nieuwe scherm zichtbaar met wat er bijgekomen/aangepast/verwijderd is van bestanden



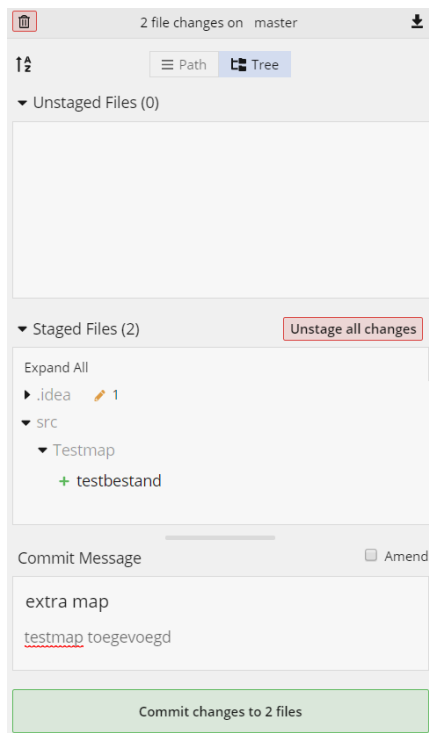
Unstaged files: je ziet dat er een bestand aangepast is sinds de vorige commit. Het is belangrijk dat dit bestand ook meegaat met de commit, dus klik je hier op 'stage all changes'

Staged files: daarnaast zie je dat er een map en een bestand manueel zijn toegevoegd aan het project. Doordat we in IntelliJ op Add hadden geklikt, komt dit hier zichtbaar.

Commit message: hier komt de beschrijving van wat we hebben gedaan. Belangrijk is om hier duidelijk te zijn, zodat indien je iets moet terugzetten naar een vorige positie, je dit aan de omschrijving kan achterhalen.

Nadat je het bestand van de unstaged Files hebt toegevoegd, krijg je onderstaand scherm

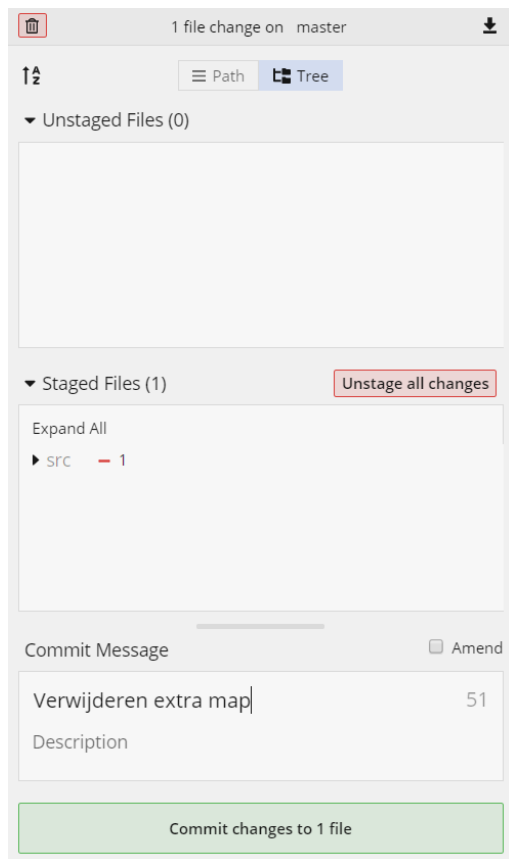




Ook zie je visueel dat er een commit is gebeurd, met als beschrijving, de tekst dat je hebt opgegeven in het commit message scherm.



Als we de tesmap terug verwijderen in IntelliJ, zien we dit ook terug in GitKraken en kunnen we een commit doen.



Eens de commit doorgevoerd is, is dit terug ook zichtbaar.



Je merkt misschien op dat er hier nu 2 masters staan. De ene master in de remote master (deze met het groene icoontje), de andere is de lokale master (deze met het computer icoontje). Uiteindelijk werk je lokaal op je computer en pas als iets volledig werkt, ga je dit gaan mergen met de master (zie later).

## 6 Aanmaken nieuwe branch GitKraken

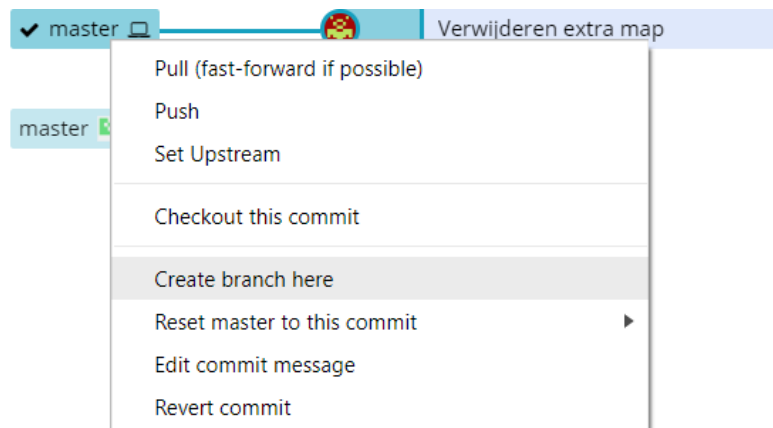
In het voorgaande voorbeeld hebben we de aanpassingen doorgevoerd op de master.

Het is niet de bedoeling dat we op de master werken, want de master is de laatste complete werkende versie van ons project.

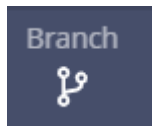
Als je iets aan het bij programmeren bent, maakt je altijd een nieuwe branch aan om te testen en eens alles werkt, zal je mergen met de master.

Het aanmaken van een nieuwe branch kan op 2 verschillende manieren:

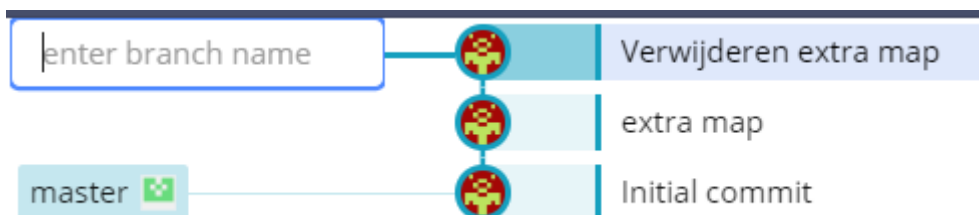
- Je selecteert de lokale master branch (hier de lokale want we hebben al aanpassingen doorgevoerd) en je klikt met je rechtermuis en selecteer 'create branch here'



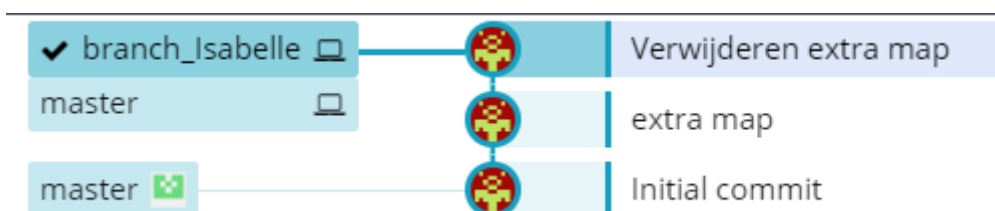
- Ofwel selecteer je je terug een tag of branch naam en klik je bovenaan op branch














Je zal zien dat je nu de naam van de branch kan ingeven.





Na ingave van de branch naam, zal je zien dat er nu 2 branches gekoppeld zijn aan de commit 'verwijderen extra map'



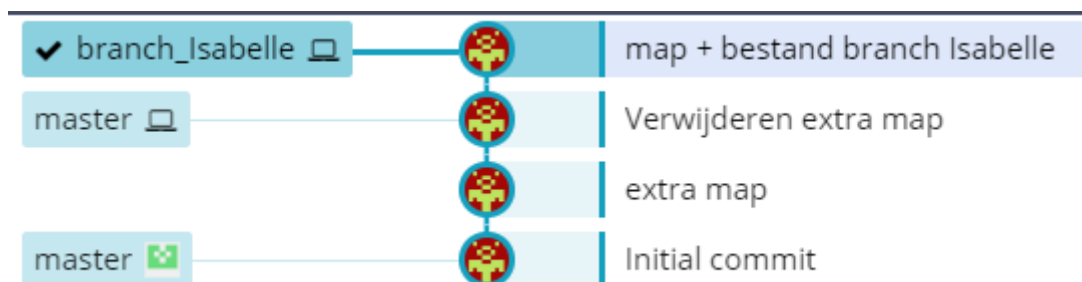
Ook zie je in het left panel dat de extra branch erbij gekomen is en dat er nu op deze branch gewerkt wordt vanuit GitKraken.

 LOCAL	2/2
  branch_Isabelle	
 master	2 ↑
 REMOTE	1/1
 origin	
 master	
 PULL REQUESTS	0
 TAGS	0/0
 SUBMODULES	0
 GITHUB ACTIONS	0

Ook zie je in IntelliJ dat de branch aangepast is (onderaan rechts)

 Event Log
 Git: branch_Isabelle

Als we nu zaken aanpassen in IntelliJ en deze aanpassingen doorvoeren in GitKraken, wordt alles zichtbaar en zie je dat de aanpassing enkel op branch\_Isabelle gebeurd is.



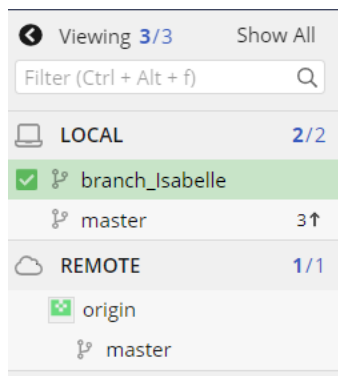
## 7 Push en Pull GitKraken

Ons project bestaat uit een lokale repository en een remote repository. Wat je lokaal doet (op je eigen computer), kan enkel jij zien.

Aangezien je in een team aan éénzelfde project werkt, is het natuurlijk ook belangrijk dat je teamleden je aanpassingen kunnen zien en daarop verder werken. Daarvoor heb je de remote repository.

Als we voor het moment kijken naar ons project in GitKraken, zien we dat we 2 lokale branches hebben (master en branch\_Isabelle) en 1 remote branch (master).

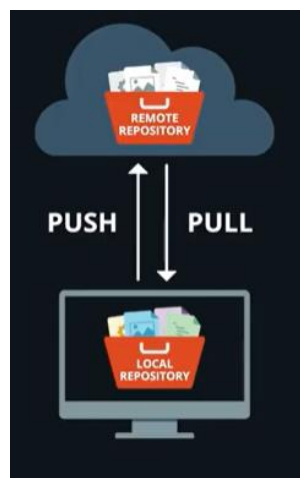
Ook zien we een **3↑** staan bij local-Master. Dit wilt zeggen dat de lokale master 3 commits voor is op de remote master. Als je kijkt in het overview is dit ook zo (map-bestand aanmaken, map-bestand verwijderen, map-bestand aanmaken)



Om onze lokale master en remote master gelijk te krijgen (zelfde toestand) gaan we hiervoor moeten met push en pull werken.

Onderstaande figuur geeft weer wanneer je wat gebruikt. Wil je je lokale repository om de remote repository plaatsen, moet je een push doen.

Wil je de remote repository naar je lokale repository brengen, moet je een pull doen.

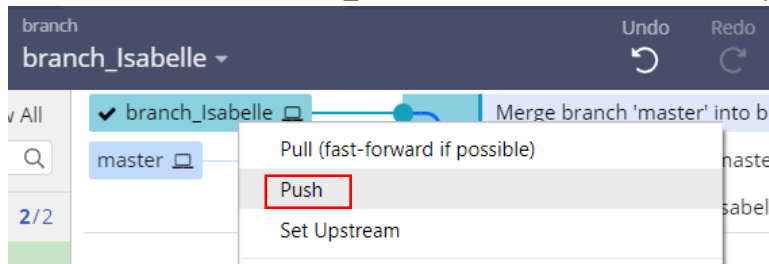



## 7.1 Push

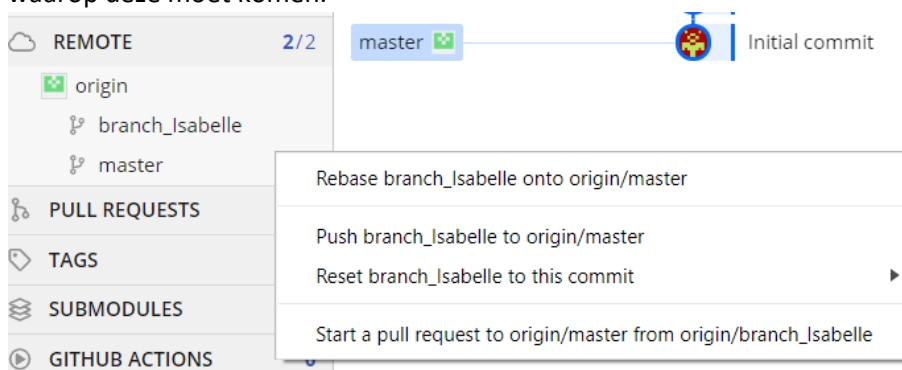
In ons overzicht zagen we dat we een lokale branch `branch_Isabelle` hadden, maar geen remote branch `branch_Isabelle`.

Om de `branch_Isabelle` ook remote beschikbaar te maken, kunnen we dit eenvoudig doen door terug 3 verschillende manieren:

- Selecteer de branch (`branch_Isabelle`), rechtermuisklik en klik op 'push'



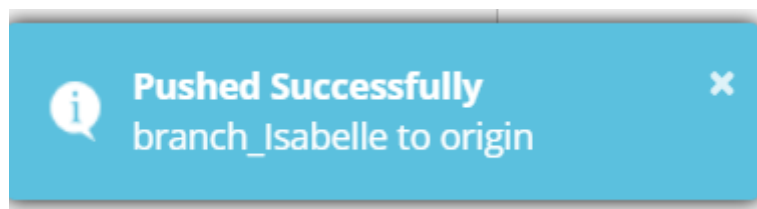
- Ofwel selecteer je de branch en klik je bovenaan op push 
- Ofwel drag en drop. Selecteer je branch in overview, en sleep je naar de remote branch waarop deze moet komen.



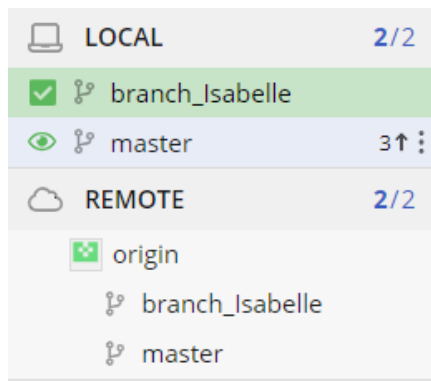
Je zal onderstaande melding krijgen (bij de eerste 2 mogelijkheden), dit omdat het de eerste keer is dat je de push uitvoert. Laat alles hier standaard staan en klik op 'submit'



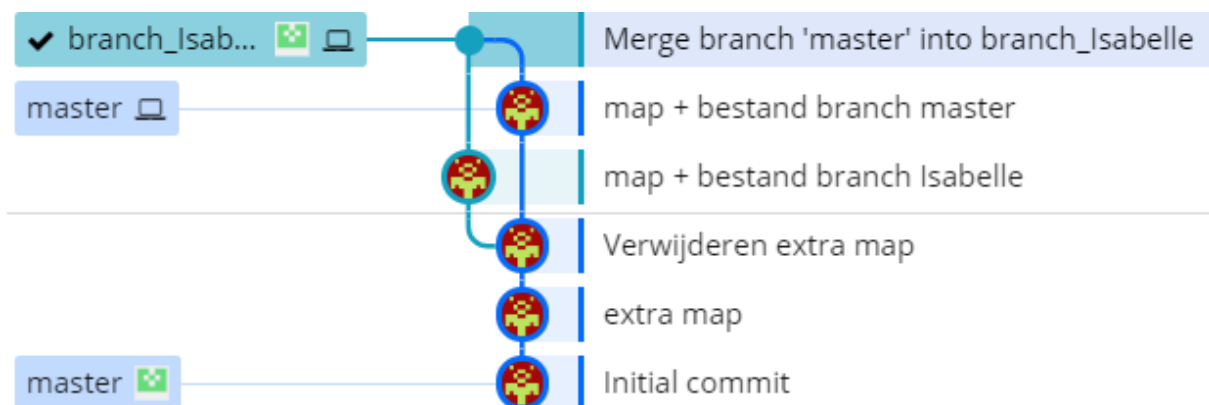
Je zal een melding krijgen dat de Push gelukt is.



En ook zal je zien in GitKraken dat de `branch_Isabelle` nu ook beschikbaar is remote.



In de Timeline zie je ook dat branch\_Isabelle zowel remote als lokaal zich op de laatste versie bevinden van het project. En dat de master lokaal voorloopt op de master remote.



## 7.2 Pull

Pull en fetch nemen de update van de remote branch en wordt in de lokale branch geplaatst.

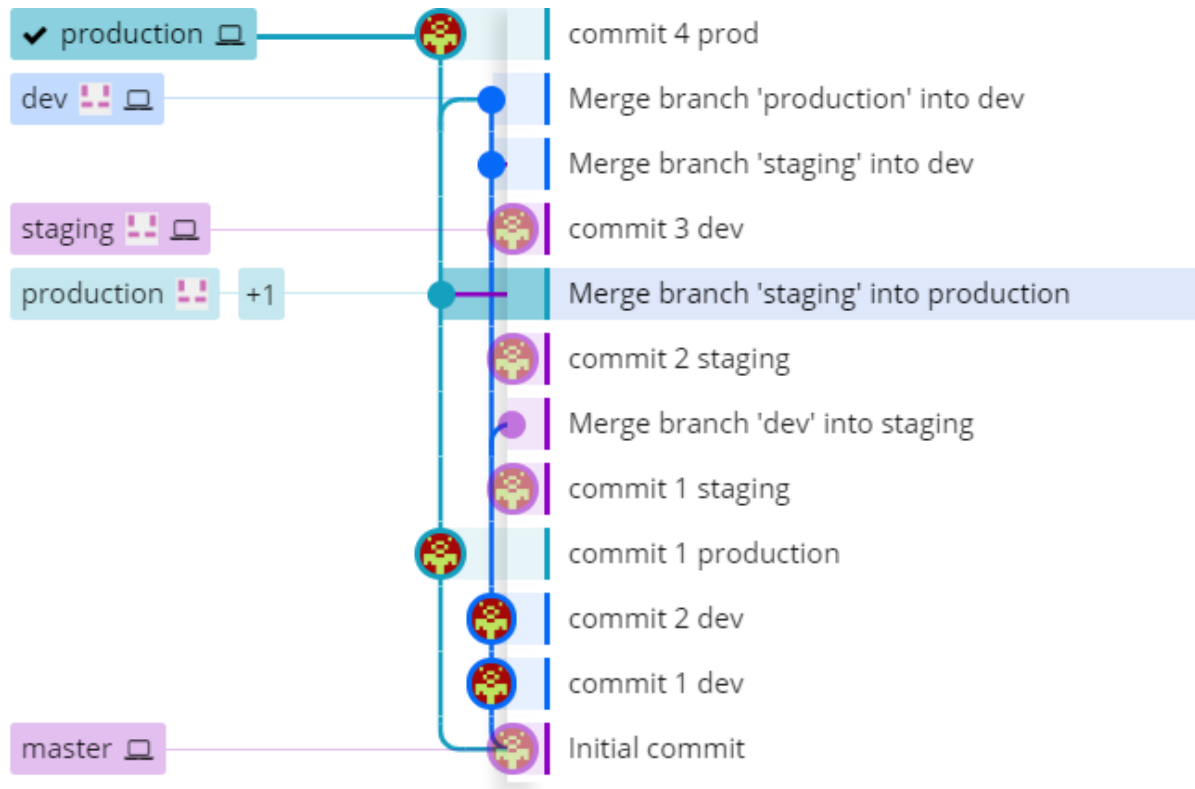
Een Pull kunnen er 4 opties geselecteerd worden:

- Fetch all
  - Haalt updates op van de remote branch, maar er worden geen bestanden in de working directory aangepast.
  - Updates gaan verschijnen in de overview en ook zal er in de left pane een overzicht komen van hoeveel commits je voor of achter loopt
  - Wanneer je behind bent, wilt dit zeggen dat je commits achter komt op de remote branch. Dus zal je een fast-forward if possible moeten doorvoeren om de verandering op je lokale branch te krijgen
  - Indien je ahead bent op de remote branch, betekent dit dat je lokale branch nog commits heeft gedaan die nog niet gepushed zijn naar de remote
  - Indien je ahead en behind bent tegelijkertijd, zal je geen pull fast-forward if possible kunnen uitvoeren
- Pull (fast-forward if possible)
  - Zal de momenteel uitgekeekte commit verplaatsen naar een commit die later werd toegevoegd. Alle tussenliggende commits worden mee verplaatst in de volgorde dat de commits toegevoegd geweest zijn.
- Pull (fast-forward only)
  - Haal de updates op en probeert een fast-forward.
  - Indien dit niet lukt zal GitKraken geen wijzigingen aanbrengen aan de lokale repo

- Pull (rebase)
  - Verzamelt alle commits op de branch, haalt de nieuwe commits op van de remote en voegt de commits toe in volgorde dat ze zijn toegevoegd.
  - Alle commits worden op één regel bijgehouden

### 7.2.1 Voorbeeld fetch all

We vertrekken van onderstaande view



Je krijgt in de left pane een overzicht van hoeveel commits voor of achter

LOCAL	4/4
dev	
master	7↓
✓ production	1↑
staging	

De lokale production zit 1 commit voor op de remote production.

De lokale master zit 7 commits achter op de remote master



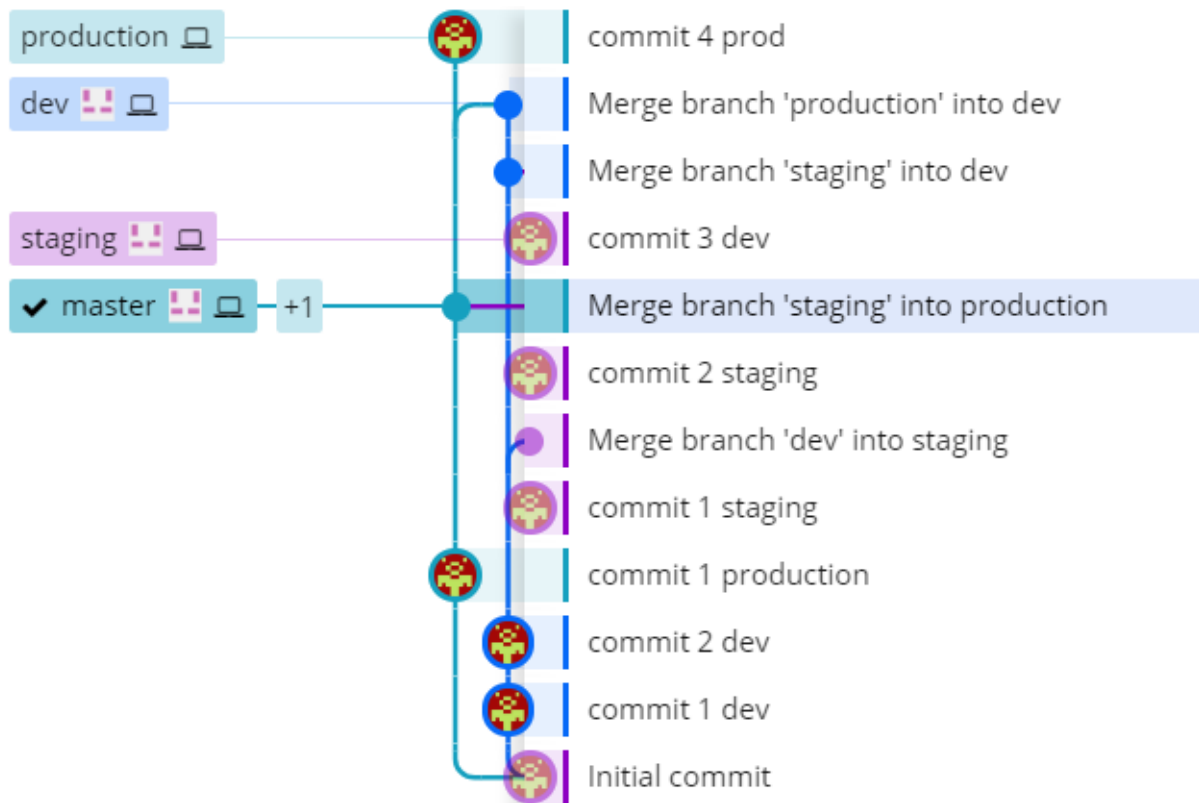
### 7.2.2 Voorbeeld Fast Forward if possible

Lokale master loopt 7 commits achter op remote master, we gaan hier een fast-forward if possible doen.

We krijgen volgende melding



En we zien in de overview dat de lokale master, gelijk geplaatst wordt met de remote master.



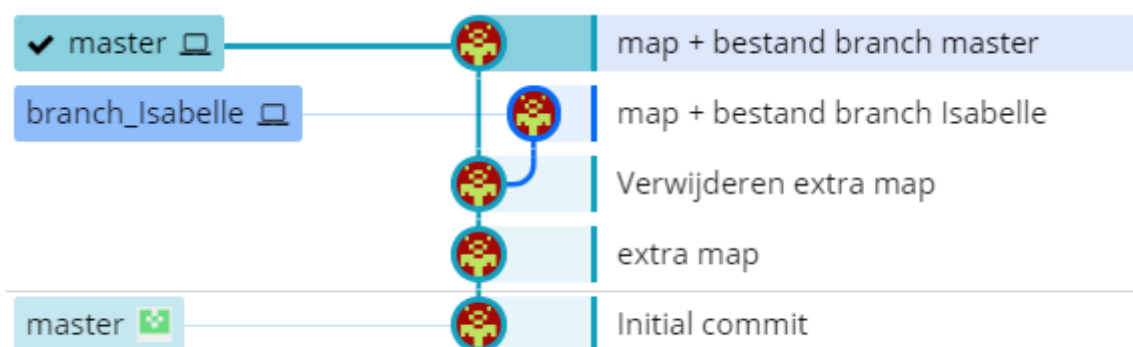
## 8 Mergen van branches GitKraken

### 8.1 Merge

Merging is het samenbrengen van twee verschillende branches en combineer deze. Met een merge worden bestanden automatisch samengebracht, tenzij er conflicten zijn tussen bestanden. (vb commits van verschillende branches die dezelfde lijn code aanpassen op een verschillende manier).

Ik zit met onderstaande situatie. 2 verschillende branches branch\_isabelle en master. Op de 2 branches heb ik 2 verschillende commites gedaan. Op branch\_isabelle (map en bestand toegevoegd) op branch master (map en bestand toegevoegd). Dit zie je ook in onderstaand schema. Na de tag van 'verwijderen extra map', zie je dat branch\_isabelle een commit heeft gehad en dat master ook een commit heeft gehad. Nu willen we deze 2 commits toevoegen in 1 branch.

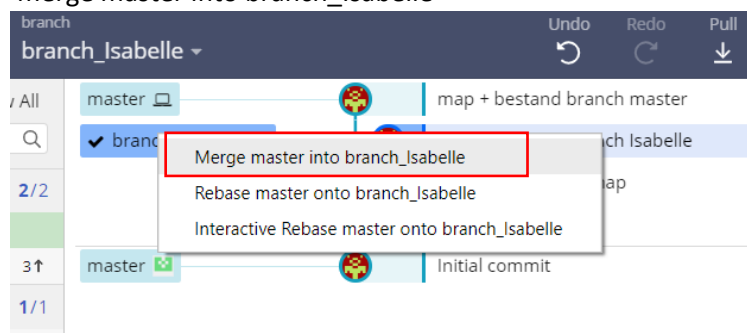
Aangezien we normaal niet op de master werken, gaan we de master mergen met de branch\_isabelle;



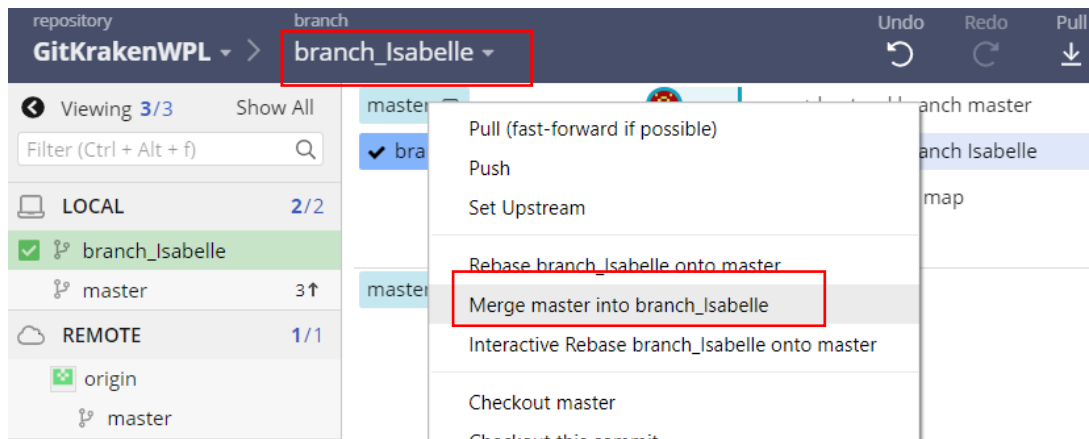
Mergen van branches kan je terug op 2 manieren doen (vb hier met master naar branch\_isabelle):

Zorg dat je op de juiste branch zit (branch\_isabelle: dit zie je bovenaan, ofwel zie je ook dat de huidige geselecteerde branch donkerder blauw is)

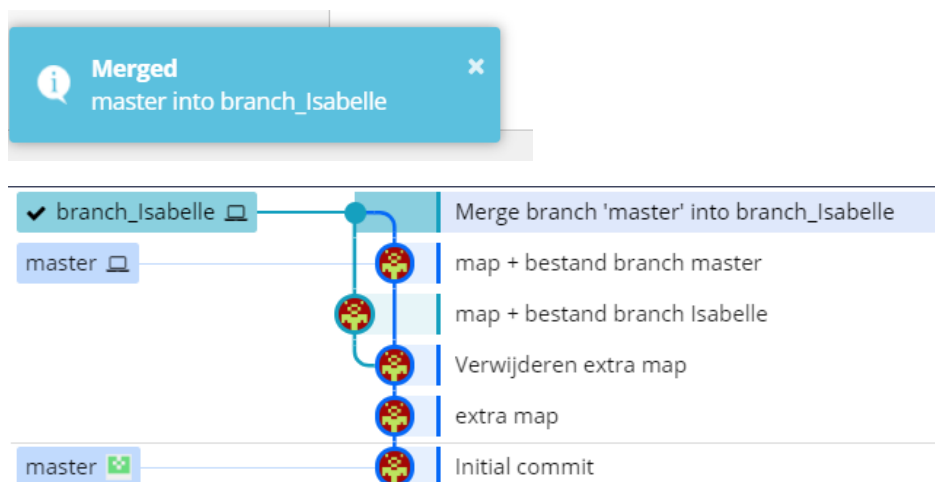
- de ene branch op de andere branch te slepen (master naar branch\_isabelle slepen) en 'merge master into branch\_isabelle'



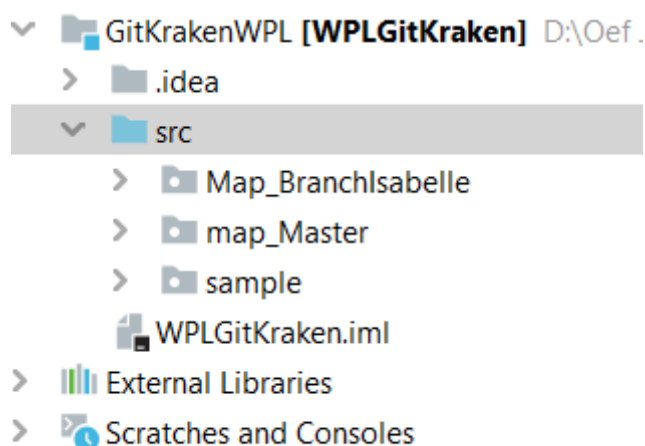
- Selecteer de branch dat je wilt mergen (master) en rechtermuisklik en selecteer 'merge master into branch\_isabelle'



Je zal de melding krijgen dat de merge heeft plaatsgevonden en dit is ook visueel zichtbaar.



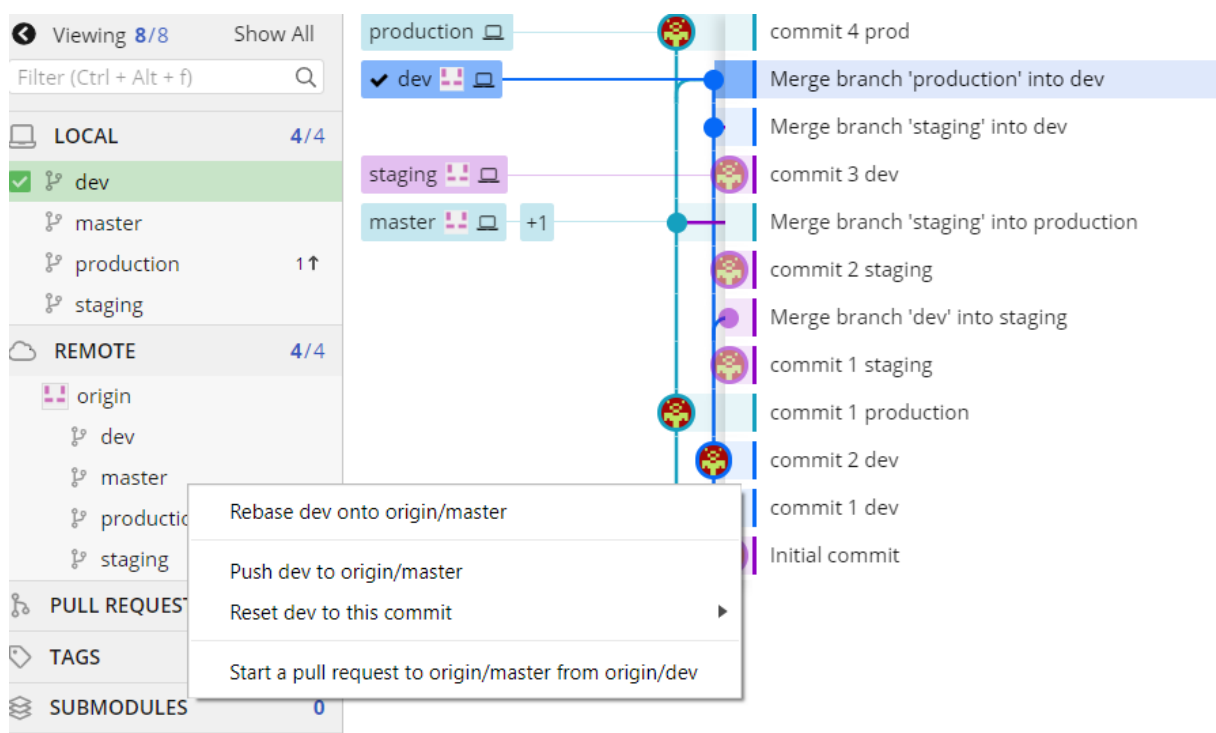
Ook in IntelliJ is de map vanop de master erbij toegekomen



## 8.2 Pull request


Pull request worden ook soms merge requests genoemd. Pull request komen erop neer dat je aan iemand vraagt om de aanpassingen op een branch te bekijken/checken voordat deze gemerged worden met een andere branch.


Pull request kunnen op volgende manier gestart worden. Indien je op een remote branch werkt, kan je een pull request aanvragen door de remote branch naar een andere remote branch te slepen en 'start a pull request to origin/master from origin/dev'. Dit is correct, want de branch dev loopt voor op branch master en we willen alles van de branch dev ook op de branch master krijgen.




Er wordt een extra venster geopend waar je terug een overzicht krijg van welke branch naar welke branch

Create Pull Request

 GitHub.com

 GitLab.com

 Bitbucket.org

From Repo

IsabelleVdn/test\_wpl

▼

→

To Repo

IsabelleVdn/test\_wpl

▼

Branch

dev

▼

Branch

master

▼

Title

merge dev into master

Description


mergen van branch dev naar branch master

Reviewers

Add reviewers...

▼

Assignees

 IsabelleVdn

×

▼

Indien het de eerste keer is dat je een pull request aanmaakt, bekijk dan zeker onderstaande link:  
<https://help.github.com/en/github/building-a-strong-community/creating-a-pull-request-template-for-your-repository>

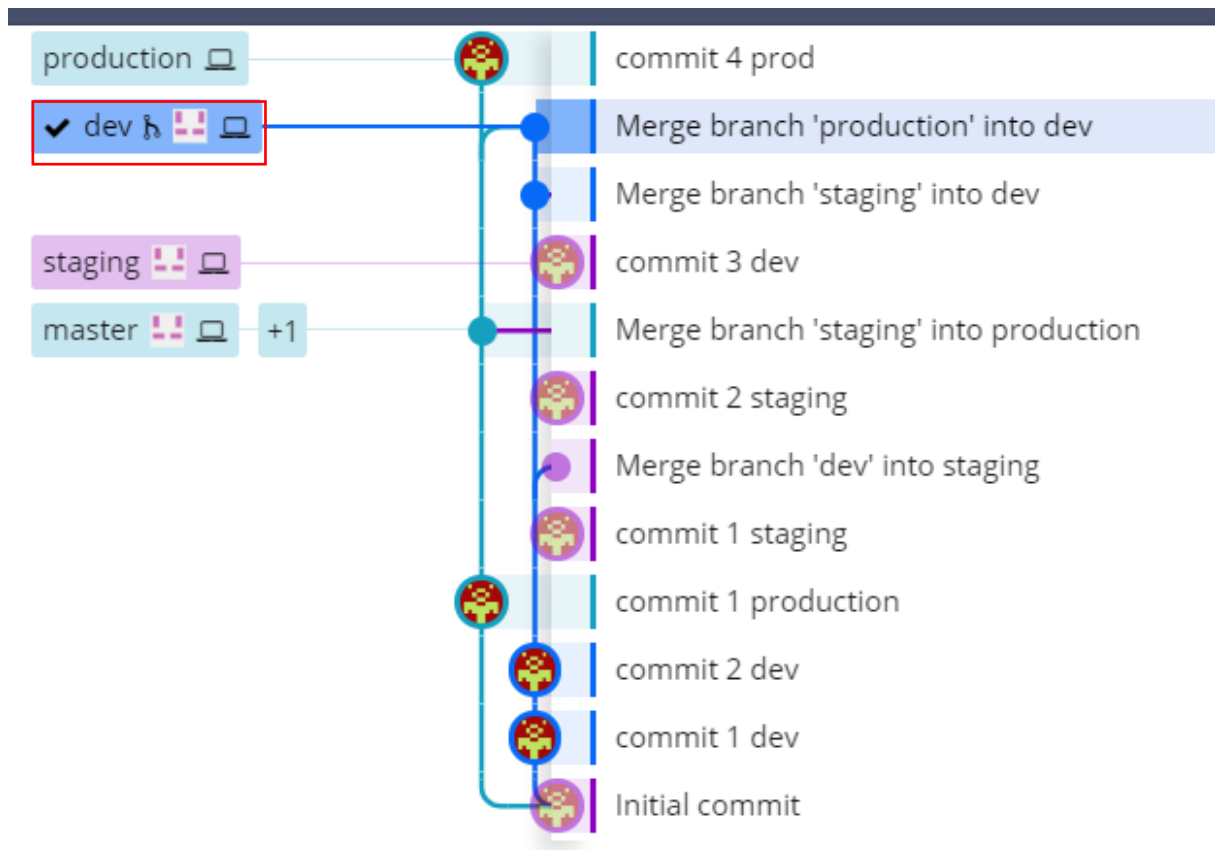
Zo is het ook mogelijk om op te geven wie dit moet bekijken en wie hiervoor verantwoordelijk is. Ook geef je een titel op.

Klik onderaan op “create pull request”

Edit on GitHub

Create Pull Request

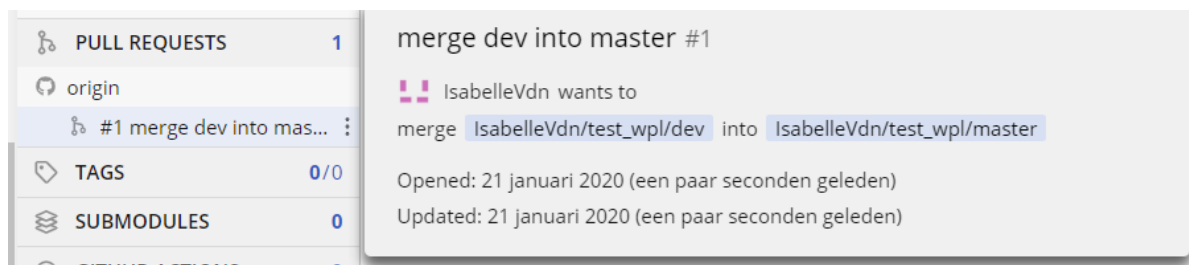
In de overview zal je zien dat er een pull request is op de branch dev



In de left pane zal je nu een pull request zien staan



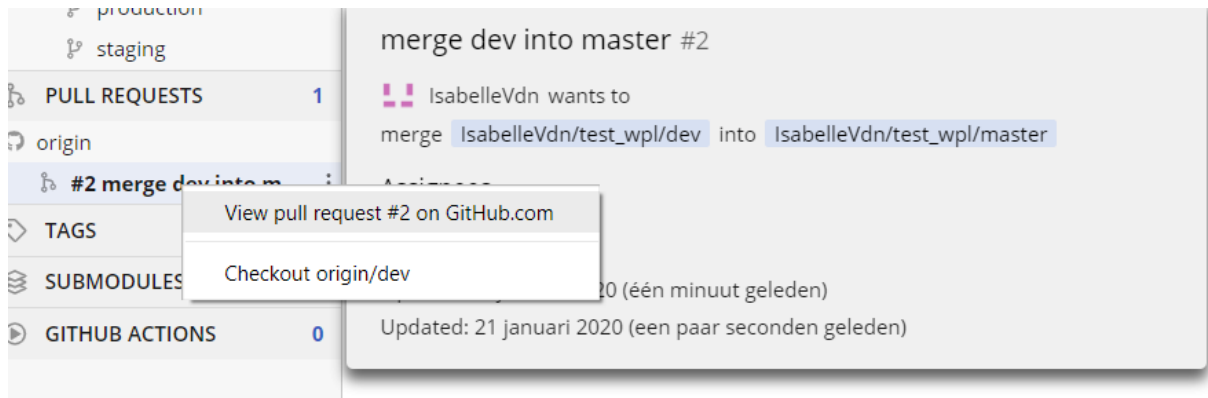
Als je over de pull request hoovert, krijg je extra info te zien wanneer de pull request geopend was en voor welke branches.



Indien na controle alles ok blijkt voor jou en je bent een reviewer, mag je de branch mergen.

Indien je vragen hebt, kan je ook een bericht achterlaten voor extra info.

Dit kan je doen op GitHub voor op 'View pull request #2 on GitHub.com'



Op GitHub krijg je onderstaande te zien

## merge dev into master #2 Edit

[Open](#) IsabelleVdn wants to merge 3 commits into `master` from `dev`

Conversation 0 Commits 3 Checks 0 Files changed 1 +0 -0

**IsabelleVdn** commented 2 minutes ago Owner + 👤 ...

mergen van branch dev naar branch master

**IsabelleVdn** added 3 commits 37 minutes ago

- commit 3 dev 35f8a1c
- Merge branch 'staging' into dev 35fc14b
- Merge branch 'production' into dev 195a531

**IsabelleVdn** self-assigned this 2 minutes ago

Add more commits by pushing to the `dev` branch on IsabelleVdn/test\_wpl.

**Continuous integration has not been set up**  
[GitHub Actions](#) and [several other apps](#) can be used to automatically catch bugs and enforce style.

**This branch has no conflicts with the base branch**  
 Merging can be performed automatically.

[Merge pull request](#) You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

**Reviewers**  
No reviews

**Assignees**  
**IsabelleVdn**

**Labels**  
None yet

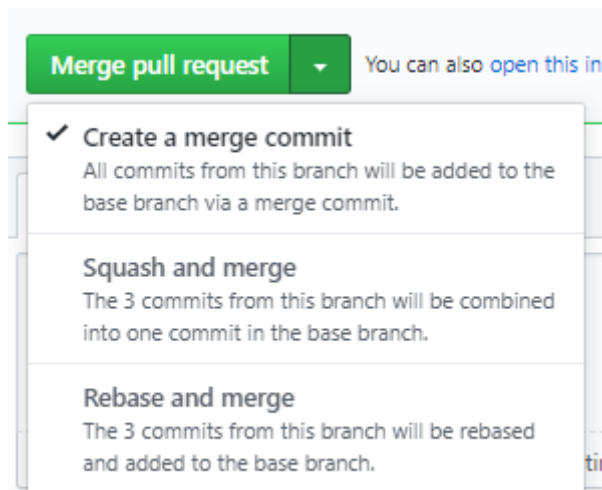
**Projects**  
None yet

**Milestone**  
No milestone

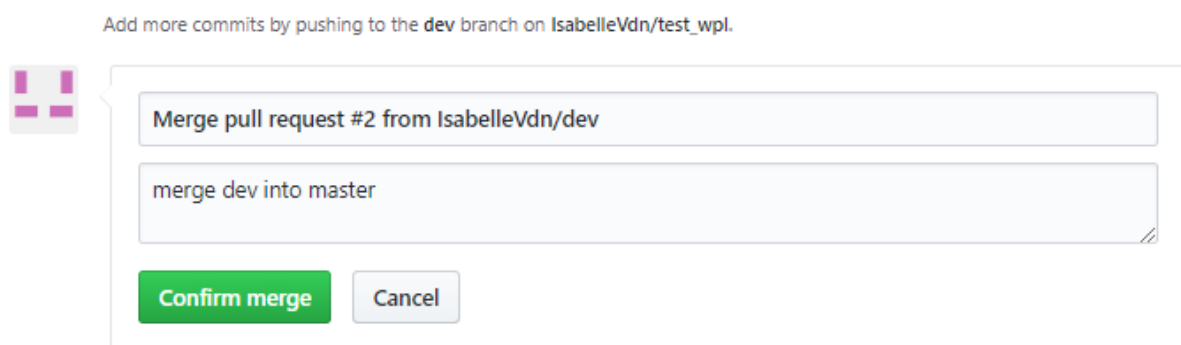
**Notifications** [Customize](#)  
[Unsubscribe](#)  
 You're receiving notifications because you're watching this repository.

**1 participant**  
**IsabelleVdn**

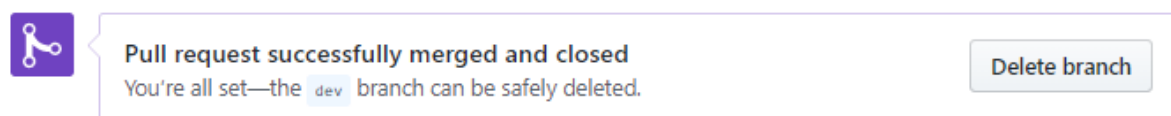
En kan je de merge selecteren want je wilt



Als je op Merge pull request klikt, zal je nog eens extra moeten bevestigen











Als alles goed verlopen is, zal je dit ook te zien krijgen en krijg je de mogelijkheid om de branch te verwijderen (want in principe heb je die branch niet meer nodig)



Verwijderen origin/dev



LOCAL 4/4	
✓  dev	
 master	4 ↓
 production	1 ↑
 staging	
REMOTE 3/3	
 origin	
 master	
 production	
 staging	

In het overzicht zie je ook dat de pull request doorgevoerd is.

