

Git werkwijze en discipline

Git is een repository voor code die wijzigingen en versies bijhoudt, ontwikkeld door Linus Torvalds, die ook Linux begonnen is.

De onderdelen zijn de gitserver, dat is je remote, ook wel origin, de git software die lokaal draait op de plek waar je de bewerkingen doet. wijzigingen die je doet doe je lokaal.

de maniere waarop je wijzigingen doorvoert in de gitserver staat hieronder beshreven.

voorbereiding

Het is handig om voor de lokale repositories, op de plek waar je gaat bewerken (Laptop of steppingstone/ansible server), een directory aan te maken om het overzichtelijk te houden, er zijn namelijk nogal wat repositories.

Zelf kies ik meestal voor een directory ~/gitrepos/ met daarin weer een directory per omgeving b.v.~/gitrepos/nationaalarchief/. en vervolgens cd ~/gitrepos/nationaalarchief/.

instaleer git.

Stel je persoonlijke gegevens in:

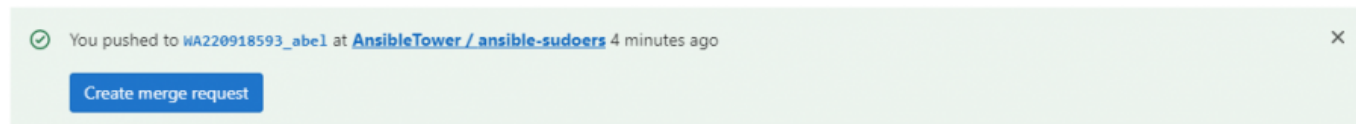
```
git config --global user.name "Your Name" git config --global user.email  
"your_email@whatever.com"''' ===== cloning ===== Branch purpose ===== De  
branch main (in oudere repositories ook wel master) wordt gebruikt voor  
productie. Hierin bevindt zich dus alleen productie waardige code. <note  
warning>edit nooit in de main/master branch</note> Voor werk aan de code maak  
je altijd een nieuwe branch aan vanuit main. ===== Branch nameing ===== De  
naamgeving van de branch waarin je werkt is vrij in te vullen, maar het is  
handig om hier een topdesk ticketnummer of een Jira userstory nummer in te  
verwerken zodat het eenvoudiger is om de reden van veranderingen terug te  
vinden. ===== Braching ===== zorg dat je branch niet te lang blijft bestaan  
(liefst alleen op de dag dat je er aan werkt), zo zitten wijzigingen van  
verschillende mensen elkaar het minst in de weg. hoe maak je in een bash  
shell (of een VSCode terminal) een nieuwe branch aan: * clone de repository  
van het gitlab project waar je aan gaat werken, b.v.:  
git clone git@git.nationaalarchief.net:tb/openshift/minio-manager/surf-s3-resources.git * zorg dat  
je in de main branch bent:  
git checkout main * zorg dat je de laatste versie lokaal hebt:  
git pull origin main * maak lokaal een nieuwe branch vanuit deze main branch (als  
voorbeeld hier een Topdesk ticket TEAMINFRA-11):  
git checkout -b TEAMINFRA-11 rclone test buckets * doe je wijzigingen. * voeg eventueel  
nieuwe en gewijzigde files toe door:  
git add. * commit de wijzigingen in alle files:  
git commit -am "commentaar over wijziging" * Doe je een push naar de GitLab server  
van jou branch:  
git push origin TEAMINFRA-11_rclone_test_buckets"
```

op dit moment is je werk opgeslagen in je eigen remote branch (dwz op de git server)

Merge Request

Als je alle wijzigingen hebt doorgevoerd en vanuit je eigen branch hebt getest is het tijd om deze te mergen naar **sios**.

Dit doe je op de GitLab server. Log in en ga naar de repository waar je aan werkt. bovenaan zul je je laatste push zien:



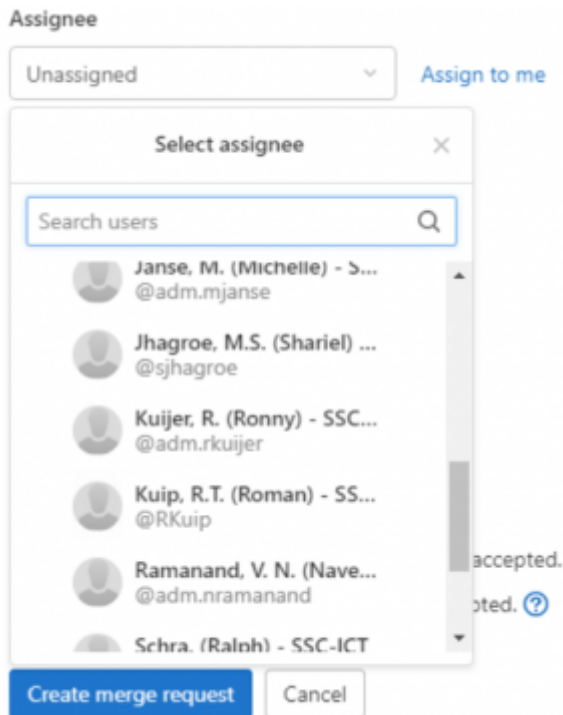
druk op **Create merge request** en druk vervolgens op **change branches** om de Target branch naar **sios** te veranderen aangezien GitLab automatisch daar de **main** branch invult.

New merge request

Source branch	Target branch
ansibletower/ansible-sudoers WA220918593_abel  New Leonardo servers Hans Vervaart authored 4 days ago edd2e3dd 	ansibletower/ansible-sudoers sios  Merge branch 'WA220918593' into 'sios' ... Kuip, R.T. (Roman) - SSC-ICT authored 1 day ago ee5bf2e2 
Compare branches and continue	

als je nog niet helemaal klaar bent maar alleen een onderdeel van je wijzigingen wilt testen in samenhang met de wijzigingen van anderen kan je eventueel het vinkje **Delete source branch** uitzetten, normaal gesproken wordt je change branch opgeruimd.

Vervolgens druk je op **Assign** om de merge request toe te kennen aan een collega die de merge uitvoert.



Merge naar develop of sios branch

Als collega krijg je een merge request op je naam van andere collegae om het vier ogen principe te hanteren.

Als er geen merge conflicten zijn check je de wijzigingen op rare dingen en mogelijke verstoringen druk hiervoor op de knop **Open in Web IDE**, hier krijg je ge-highlight de wijzigingen te zien.

(plaatje hier)

Als je geen rare dingen bent tegen gekomen druk je op **Merge** en geef je aan degene die de wijziging heeft gedaan door dat hij kan gaan testen met de **sios** branch.

Merge naar main

Wanneer de tests met develop succesvol zijn is het tijd om een merge request te doen voor develop of sios naar main.

in de rechter zijbalk druk je op **Merge requests**, in het scherm wat verschijnt druk linksboven op de blauwe knop **New merge request**

in het volgende venster selecteer **sios** in het pulldown menu **Select source branch** Target branch staat meestal al op **main** druk op **Compare branches and continue**

vervolgens assign je de merge request weer aan een andere collega wederom ter controle en om er voor te zorgen dat er als team gewerkt wordt en niet als een "Remy" allerlei wijzigingen door te voeren waar het team later door verrast wordt.

Last update:

2025/09/04

15:37

werkinstructies:git:gitlab-discipline <https://wiki.auriel.nl/doku.php?id=werkinstructies:git:gitlab-discipline&rev=1756993065>

From:

<https://wiki.auriel.nl/> -

Permanent link:

<https://wiki.auriel.nl/doku.php?id=werkinstructies:git:gitlab-discipline&rev=1756993065>



Last update: **2025/09/04 15:37**