

# GitHub and Git

# GitHub

Git is a version control system

GitHub is the place where people store their projects

# Tutorial

- Create a GitHub Account
- Create and use a repository
- Start and manage a new branch
- Make changes to a file and push them to GitHub as commits
- Open and merge a pull request

# Creating New Repository


**Repositories**

New repository

You don't have any repositories yet!

Owner

Repository name


 DevOpsVerity ▾

 / 


hello-world ✓

Great repository names are short and memorable. Need inspiration? I

Description (optional)

☒  **Public**

Anyone can see this repository. You choose who can commit.

☐  **Private**


You choose who can see and commit to this repository.

☒ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer. Skip thi repository.

Add .gitignore: **None** ▾

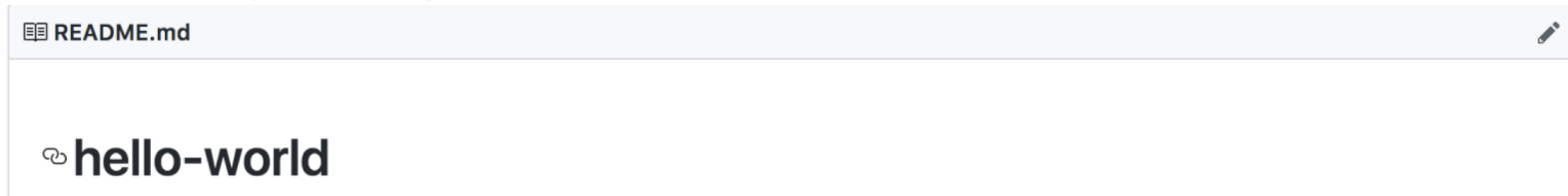
 | 

Add a license: **None** ▾ 

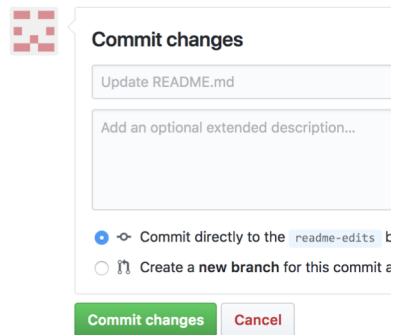
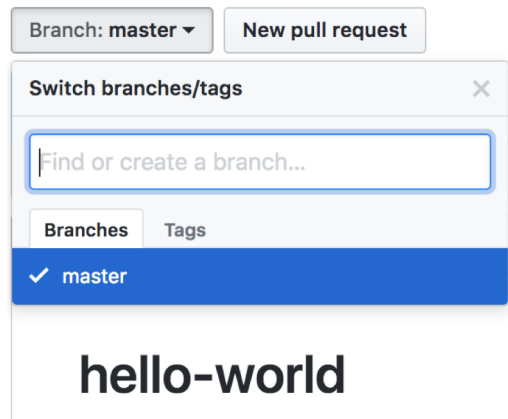
Create repository

# Branching and Commit

- Create a branch names  
readme-edit
- Edit the readme in the  
branch by clicking edit icon



- Write something and  
commit



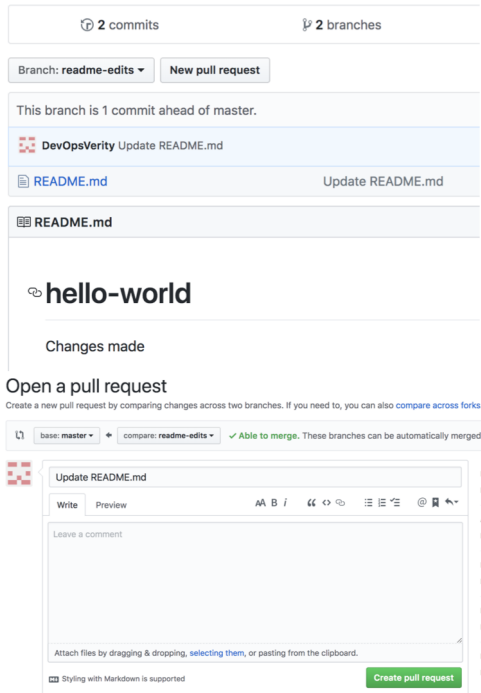
# Pull Request

To do comparisons and merge changes

Look at the changes

Create a pull request

When you open a *pull request*, you're proposing your changes and requesting that someone review and pull in your contribution and merge them into their branch. Pull requests show *diffs*, or differences, of the content from both branches. The changes, additions, and subtractions are shown in green and red.



# Merging

- Click the Merge pull request button to merge the changes into master
- Click Confirm merge
- Delete the branch, since its changes have been incorporated, with the Delete branch button



**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](#).



**Pull request successfully merged and closed**

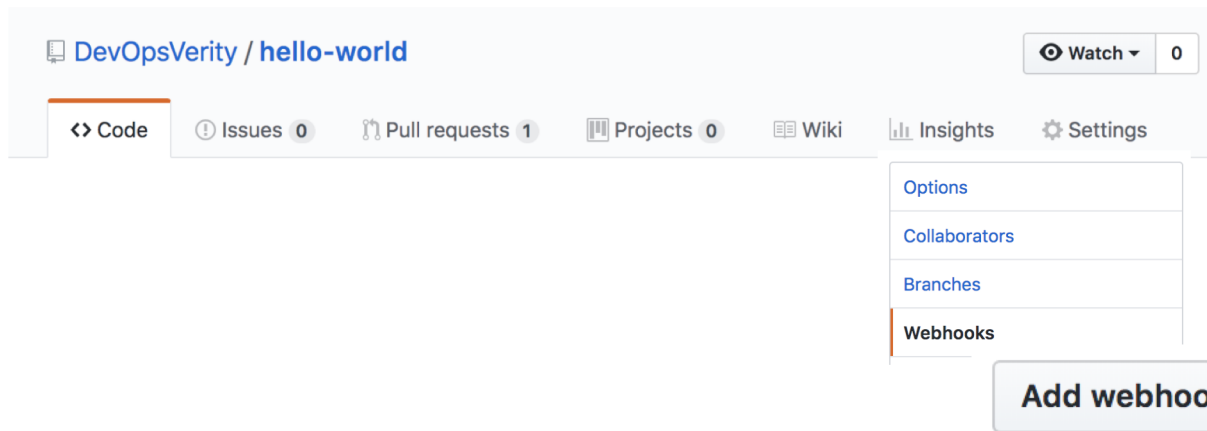
You're all set—the [readme-edits](#) branch can be safely deleted.



**Delete branch**

# Setting up Webhooks in GitHub

- Webhook will notify Jenkins when new commit is done
- GitHub web > Go to "Settings" (of repository) > Click "Webhooks" > Click "Add webhook"
- Enter "Payload URL" `http://localhost:8080/github-webhook/`
- Select Application type "application/json"
- Click "Add webhook"



## Webhooks / Add webhook

We'll send a POST request to the URL below with the data format you'd like to receive (JSON, x-www-form-urlencoded, or text). See [developer documentation](#).

### Payload URL \*

`https://example.com/postreceive`

### Content type

`application/x-www-form-urlencoded` ⌵



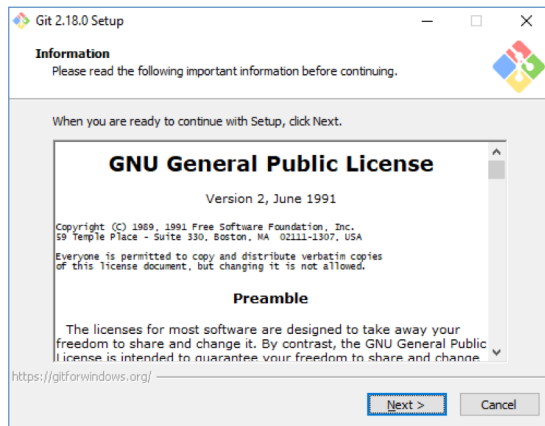
# Download & Install GitHub Desktop

- Open the browser and go to <https://desktop.github.com/>
- Click “Download for Windows (64bit)”
- Run the installer
- Follow the instructions from the wizard until the installation is completed



# Git

- Download Git from <https://git-scm.com/downloads> if not already installed
- Run the installer
- Follow the instructions from the wizard until the installation is completed
- Create a folder(eg. git) in any drive in your machine(e.g. C:\git)
- Launch GitBash from C:\Program Files\Git\git-bash.exe
- Launch following commands in GitBash:
  - `cd /C/git`
  - `git config --global user.name "<your github username>"`
  - `git config --global user.email "<your email id>"`



# Some Git Commands

- git clone <https://github.com/umangsaltuniv/testing.git>
- cd testing
- Copy HelloWorld.java file from Git folder and paste it in to C:\git\testing
- git add HelloWorld.java
- git status
- git commit -m "myfirstcommit" HelloWorld.java
- git push -u origin master
- Go to GitHub web and click repository link name(e.g. testing)