

# Controlul versiunilor GIT

Marin Bînzari  
Pentalog



# Cuprins

1. Ce este Controlul versiunilor ?
2. Instalare
3. GIT
4. Crearea / clonarea unui repozitoriu git
5. Getting hands dirty
6. Pull request
7. Useful tips

Ce este controlul  
versiunilor ?





# Controlul versiunilor

Controlul versiunilor este un sistem care se ocupă cu gestionarea mai multor versiuni ale unor fișiere. Git este un exemplu de control al versiunilor.

- Un sistem ce deține istoricul tuturor modificărilor făcute
- Oferă posibilitatea dezvoltării colaborative
- Permite să afli cine, cînd și ce modificări a făcut
- Permite să anulezi modificările și să te întorci la o stare anterioară

Dacă ai dat-o în bară sau ai șters fișiere  
îți permite să le recuperezi cu ușurință !

# GIT

## Cazuri de utilizare

- Dezvoltare individuală
- Dezvoltare colaborativă
- Utilizare offline

## De ce GIT ?

- Totul este în local (tot istoricul e disponibil)
- Este rapid
- Stochează snapshoturi ( nu doar diferențele )
- E distribuit, nu centralizat
- Este cel mai bun ( discutabil )



Instalare





# Instalare git

<https://git-scm.com/downloads>

Instalare : `sudo apt-get install git`

Verificarea versiunii : `git --version`

Configurarea identității

`git config --global user.name "Marin Binzari"`

`git config --global user.email "mbinzari@pentalog.com"`

Activare culori

`git config --global color.ui auto`



# Create cont Github

<https://github.com>

Alternative :

- <https://bitbucket.org>
- <https://gitlab.com>

Username

Pick a username

Email

you@example.com

Password

Create a password

Use at least one letter, one numeral, and seven characters.

**Sign up for GitHub**

By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy policy](#). We'll occasionally send you account related emails.





# Generare cheie SSH

Generare :

→ `ssh-keygen`

Generating public/private rsa key pair.

Enter file in which to save the key (/home/mbinzari/.ssh/id\_rsa):

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /home/mbinzari/.ssh/id\_rsa.

Your public key has been saved in /home/mbinzari/.ssh/id\_rsa.pub.

# Adaugare cheie SSH pe Github



Search GitHub

[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)



Personal settings

[Profile](#)

[Account](#)

[Emails](#)

[Notifications](#)

[Billing](#)

**SSH and GPG keys**

## SSH keys

[New SSH key](#)

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.



SSH

**mbinzari@P5MD-PC075**

**Fingerprint:** 42:e8:12:4e:ff:a3:6f:8a:8f:22:07:ee:11:ef:01:22

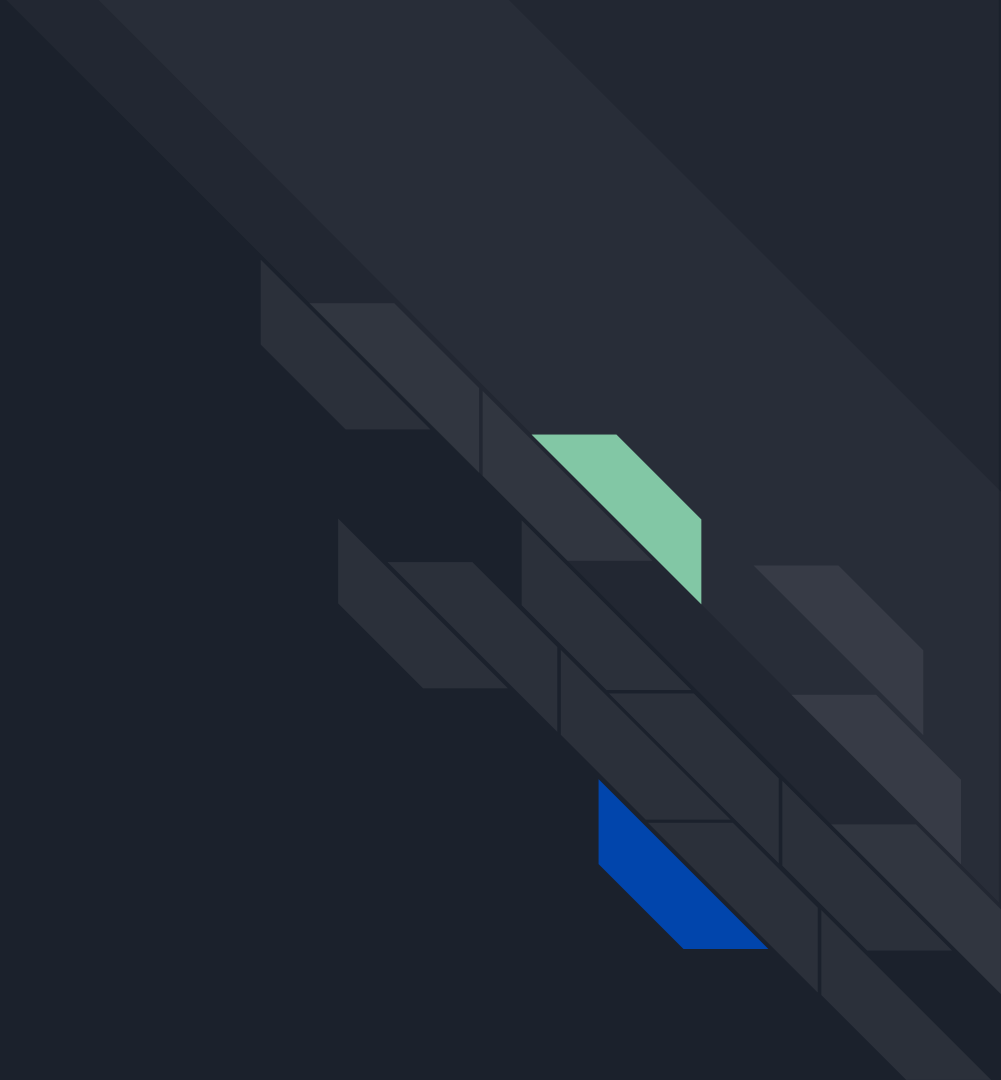
Added on Mar 23, 2015

Last used within the last 3 weeks — Read/write

[Delete](#)

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH Problems](#).

GIT





# Git workflow : stare fișiere

Stare fișiere : tracked, untracked, ignored

Stare fișiere tracked : unmodified, modified, staged or committed

.gitignore : <https://github.com/github/gitignore>

Crearea / clonarea unui  
repozitoriu git





# Crearea unui repository git local

Creați un dosar nou, deschideți-l și executați git init pentru a inițializa un repository nou.

→ `git init`

Initialized empty Git repository in /home/mbinzari/git-training/.git/

→ `git remote add origin <remote-repository-url>`



# Clonarea unui repository

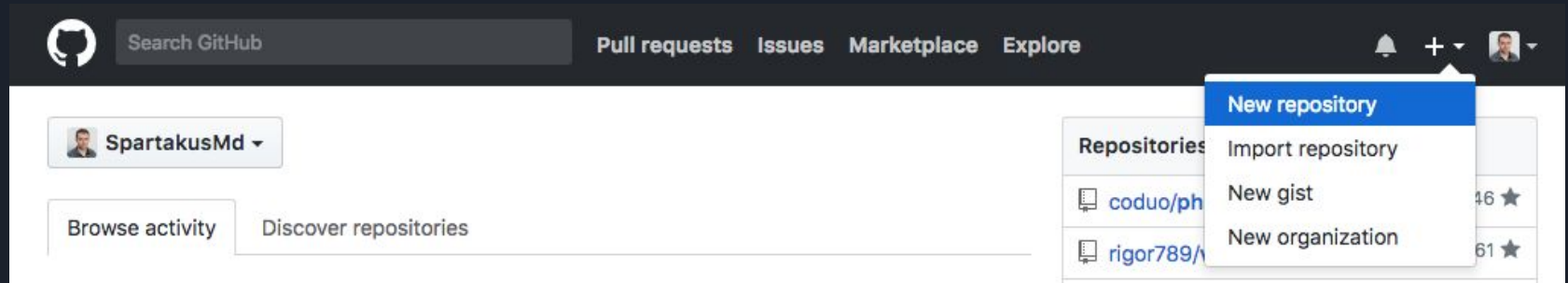
În dosarul în care doriți să creați o copie de lucru locală a unui repository executați comanda :

```
→ git clone git@github.com:<username>/<repository>.git
```

sau

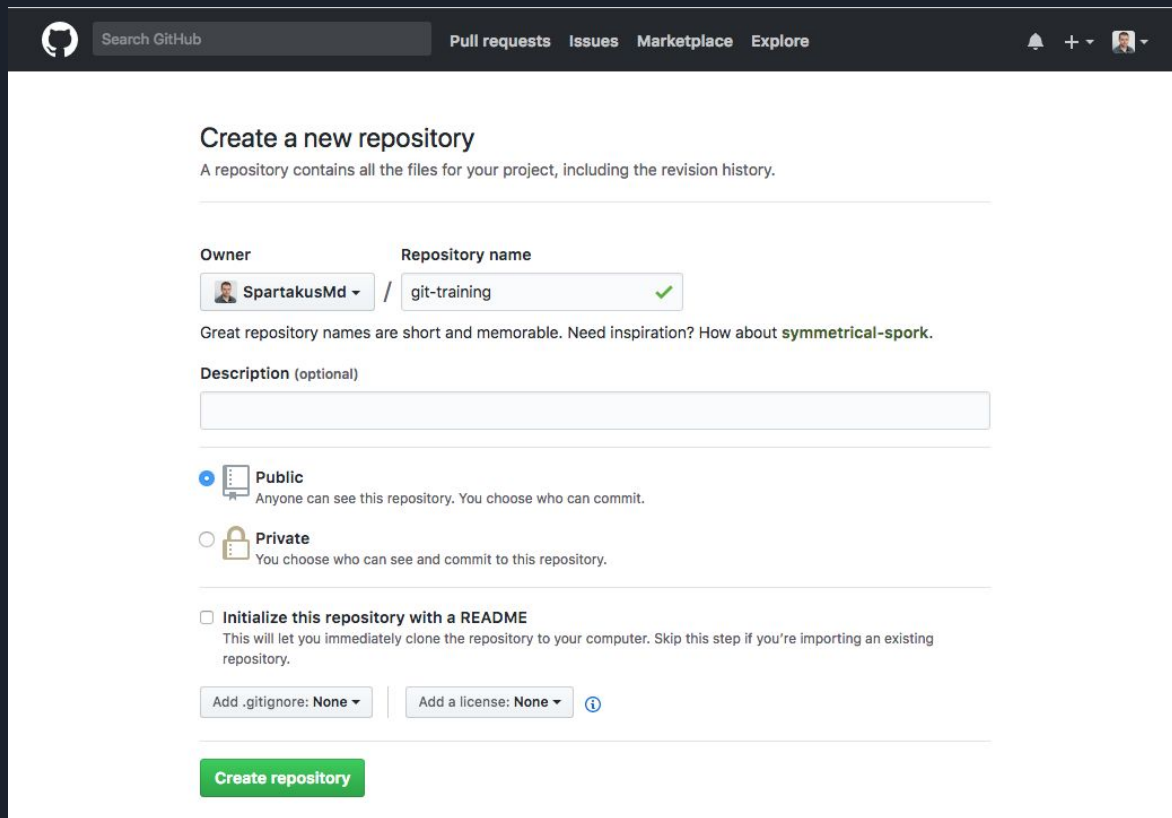
```
→ git clone https://github.com/<username>/<repository>.git
```

# Crearea unui repository pe github.com





# Crearea unui repository pe github.com



The screenshot shows the GitHub interface for creating a new repository. At the top, there's a navigation bar with the GitHub logo, a search bar, and links for Pull requests, Issues, Marketplace, and Explore. Below this, the main heading is 'Create a new repository' with a subtext explaining that a repository contains all files for a project, including revision history. The form is divided into sections: 'Owner' (SpartakusMd) and 'Repository name' (git-training), which is validated with a green checkmark. A note suggests great repository names are short and memorable, with an example 'symmetrical-spork'. The 'Description (optional)' field is empty. Under 'Visibility', 'Public' is selected, indicating anyone can see the repository. 'Private' is also an option. There's a checkbox for 'Initialize this repository with a README', which is currently unchecked. At the bottom, there are dropdowns for adding a .gitignore file (set to 'None') and a license (set to 'None'). A green 'Create repository' button is at the very bottom.

Search GitHub

Pull requests Issues Marketplace Explore

## Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: SpartakusMd / Repository name: git-training ✓

Great repository names are short and memorable. Need inspiration? How about [symmetrical-spork](#).

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 this step if you're importing an existing repository.

Add .gitignore: None | Add a license: None ⓘ

**Create repository**

# Crearea unui repository pe github.com

## Quick setup — if you've done this kind of thing before

 Set up in Desktop

or

HTTPS

SSH

git@github.com:SpartakusMd/git-training.git



We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

## ...or create a new repository on the command line

```
echo "# git-training" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin git@github.com:SpartakusMd/git-training.git
git push -u origin master
```



## ...or push an existing repository from the command line

```
git remote add origin git@github.com:SpartakusMd/git-training.git
git push -u origin master
```



## ...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#)

Getting hands dirty





# Comenzi GIT

- `git status` # starea curentă a fișierelor
- `git diff` # modificările în fișiere non-staged
- `git diff --cached` # modificările în fișiere staged
- `git add <fișier-directoriu>` # marchează ca staged
- `git commit -m "Mesaj"` # comitează modificările cu mesaj specific
- `git commit --amend -m "Mesaj nou"` # Modifică mesajul ultimului commit
- `git push origin master` # trimite modificările pe remote, pe master
- `git push -u origin master` # push + setează master remote ca branch default pentru branch-ul local curent
- `git pull origin master` # descarcă modificările de pe remote



# Comenzi GIT : reset

- `git reset # anulează git add`
- `git reset 56e05fced # resetează indexul la acel commit`
- `git reset --soft HEAD~1 # mergi înapoi cu un commit fără a pierde modificările`
- `git reset --hard HEAD~1 # mergi înapoi cu un commit fără a păstra modificările`
- `git reset origin/master # anulează comiturile locale`
- `git fetch origin ; git reset --hard origin/master # resetează varianta locală la cea remote`



# Comenzi GIT

Afișarea istoricului de commituri :

- `git log` # lista ultimelor commituri
- `git log --oneline` # o linie per commit
- `git log --stat` # afiseaza si fisierele modificate

Afișarea modificărilor unui commit :

- `git show <commitID>`

Compararea commiturilor :

- `git diff <commitID1> <commitID2>`

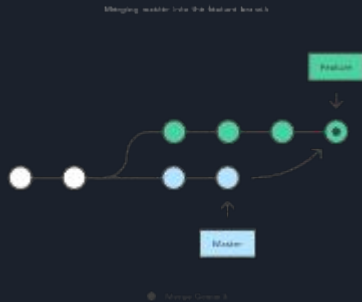


# Comenzi GIT : branch

- `git branch` # lista branch-urilor locală
- `git branch` # lista completă de branchuri
- `git branch PROJ-132` # creare branch PROJ-132
- `git checkout PROJ-132` # switch branch ; schimba versiunea de lucru la PROJ-132
- `git checkout -b PROJ-132` # creaza si schimba versiunea de lucru
- `git branch -d PROJ-132` # sterge branch

# Comenzi GIT : merge vs rebase

- git checkout feature
- git merge master



- git checkout feature
- git rebase master







# Comenzi GIT : rebase interactiv

- `git checkout feature`
- `git rebase -i master`

pick 33d5b7a Message for commit #1

fixup 9480b3d Message for commit #2 # acest commit va fi unit cu #1

pick 5c67e61 Message for commit #3

Conflicts ?

`git add <some-file>`  
`git rebase --continue`

`git rebase --abort`



# Comenzi GIT : stash

- `git stash` # Salveaza temporar
- `git stash pop` # Aplica fisierele temporare
- `git stash apply` # Aplica si pastreaza stash
- `git stash -u` # Salveaza inclusiv fisierele care nu sunt tracked
- `git stash save "Mesaj"` # Salveaza cu mesaj
- `git stash list` # Afiseaza lista
- `git stash pop stash@{2}` # Aplica dupa index
- `git stash clear` # Sterge toate stash-urile

Pull request  
aka merge request





This repository

Search

Pull requests

Issues

Marketplace

Explore



SpartakusMd / php-matcher

forked from coduo/php-matcher

Unwatch

1


★ Star

0

Fork

47

<> Code

 Pull requests 0

 Projects 0

 Insights

 Settings

Filters

is:pr is:open

Labels

Milestones

New pull request

[This repository](#)[Pull requests](#)[Issues](#)[Marketplace](#)[Explore](#)[coduo / php-matcher](#)[Watch](#)

16

[★ Unstar](#)

248

[Fork](#)

47

[Code](#)[Issues](#) 13[Pull requests](#) 1[Projects](#) 0[Insights](#)

## Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

base fork: [coduo/php-matcher](#)base: [master](#)head fork: [SpartakusMd/php-matcher](#)compare: [master](#)

✗ **Can't automatically merge.** Don't worry, you can still create the pull request.

[Create pull request](#)

Discuss and review the changes in this comparison with others.

[4 commits](#)[19 files changed](#)[0 commit comments](#)[2 contributors](#)

Commits on Aug 21, 2017

[SpartakusMd](#)

OptionalExpander for : ...



4fe85e7



Commits on Sep 10, 2017

[SpartakusMd](#)

OptionalExpander : create tests



97a5851

[SpartakusMd](#)

Don't pass unused variable



a941229



Commits on Sep 19, 2017

[SpartakusMd](#)

Test LowerThan and GreaterThan expanders on @number@



f684fcb

Showing **19 changed files** with 148 additions and 64 deletions.[Unified](#)[Split](#)

6



src/Factory/SimpleFactory.php

[View](#)



# Conflicte

<https://help.github.com/articles/resolving-a-merge-conflict-using-the-command-line/>

If you have questions, please

<<<<<< HEAD

open an issue

=====

ask your question in Slack.

>>>>>> branch-a



↑ ↓ Highlight words ▾      

5 changes. 1 conflict

Local Changes (Read-only)

Result

Changes from Server (revision 9d4b6795...

 using System.Reflecti	7	7	using System.Reflection;	7	using System.Reflecti 
using Microsoft.Exter	8	8	using Microsoft.Extension	8	using Microsoft.Exter
	×	» 9		9 << ×	
/// <summary>	10	10	/// <summary>	10	/// <summary>
/// Default implement	11	11	/// Default implementati	11	/// Default implement
/// retrieving <see c	12	12	/// retrieving <see cref	12	/// retrieving <see c
/// </summary>	13	13	/// </summary>	13	/// </summary>
public class Depend	14	14	public class DependencyC	14	public class Depend
{	15	15	{	15	{
private static re	16	16	private static readon	16	private static re
private readonly	17	17	private readonly Dep	17	private readonly
	18	18		18	
/// <summary>	19	19	/// <summary>	19	/// <summary>
/// Initializes a	×	» 20	/// Initializes a new	20 << ×	/// Initializes a
/// using <see ci	21	21	/// </summary>	21	/// using <see ci
/// </summary>	22	22	public DependencyCon	22	/// </summary>
public Dependency	23	23	{	23	public Dependency
: this(Assemb	×	» 24	var entryAssemb	24 << ×	: this(Assemb
{	25	25		25	{
	×	» 26	this.dependencyC	26 << ×	
	27	27	}	27	}
/// <summary>	28	28		28	/// <summary>
/// Initializes a	29	29	/// <summary>	29	/// Initializes a
/// using <param	30	30	/// Gets all <see cr	30	/// using <param
/// </summary>	31	31	/// </summary>	31	/// </summary>
public Dependency	32	32	/// <returns>An <see	32	public Dependency
{	33	33	public IReadOnlyColl	33	{
dependencyCor	34	34	{	34	this.depender
}	35	35	var results = new	35	}
	36	36		36	

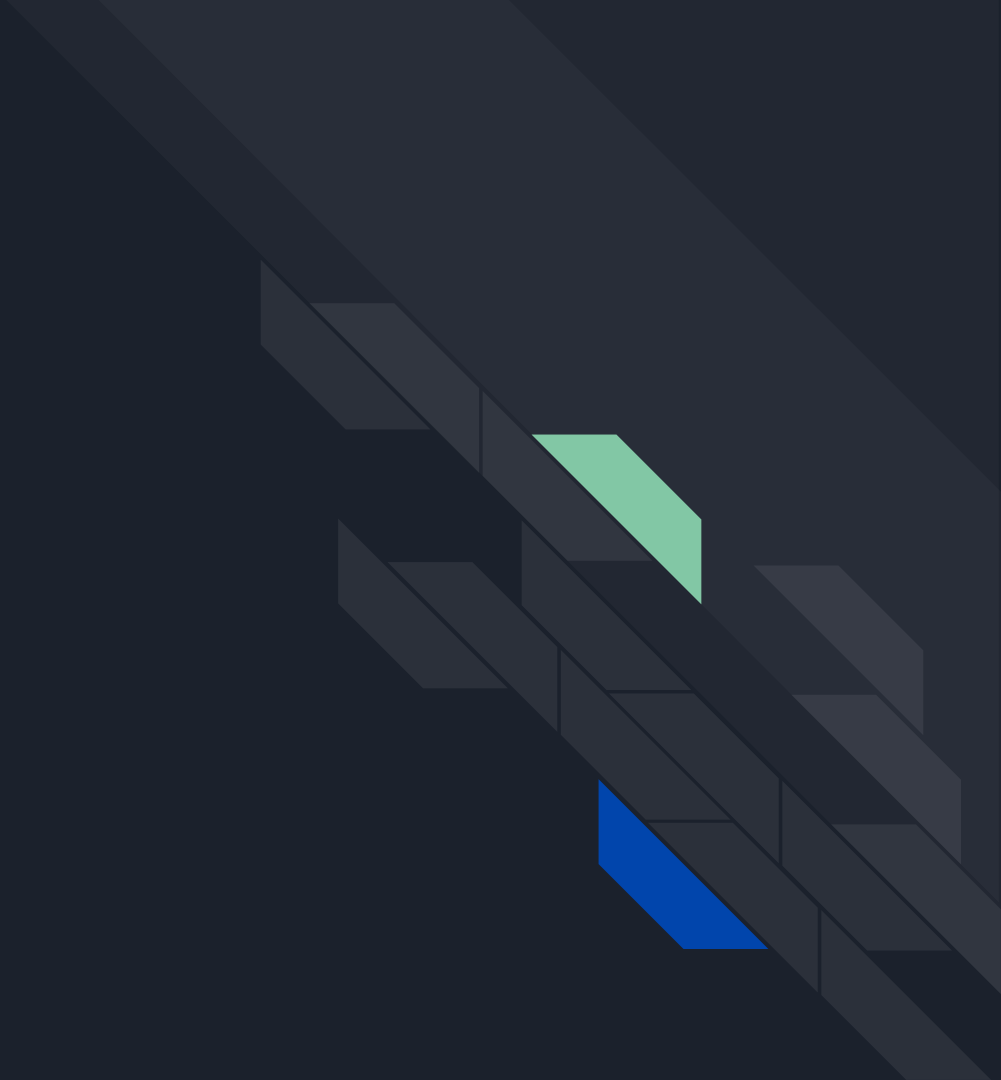
Accept Left

Accept Right

Apply

Abort

Gitflow







# Gitflow

<https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>

<https://www.atlassian.com/git/tutorials/comparing-workflows>

- Branch-ul develop e creat de pe master
- Branch-ul release e creat de pe develop
- Branch-uri per ticket / feature sunt create de pe develop
- Feature branch va fi merged pe develop
- Release branch va fi merged pe master
- Bug in productie, branch hotfix de pe master
- Hotfixurile sunt merged pe develop si master

Branch model

<http://nvie.com/img/git-model@2x.png>

Useful tips





~/.gitconfig

<https://git-scm.com/book/en/v2/Git-Basics-Git-Aliases>

git config --global alias.co checkout

git config --global alias.st status



## ~/.ssh/config

```
Host github-personal
    Hostname github.com
    IdentityFile ~/.ssh/github-personal
```

```
git clone git@github-personal:<username>/<repository>.git
```



# Helpful links

- <https://try.github.io>
- <https://www.atlassian.com/git/tutorials>
- <https://services.github.com/on-demand/downloads/github-git-cheat-sheet.pdf>