GIT

credentials and config

~/.netrc

```
machine git.csgalileo.org
login blabla
password blabla
```

~/.gitconfig

```
[user]
    name = Stefano Scipioni
    email = blabla
[core]
    autocrlf = input
    #safecrlf = true
    editor = vim
[http]
    sslverify = false
[alias]
    co = checkout
    ci = commit
    st = status
    br = branch
    hist = log --pretty=format:\"%h %ad | %s%d [%an]\" --graph --
date=short
    type = cat-file -t
    dump = cat-file -p
[push]
    default = simple
[color]
    ui = auto
```

New repo

```
NAME=project.git
mkdir $NAME
cd $NAME
git init --bare
mv hooks/post-update* hooks/post-update
chmod +x hooks/post-update
```

```
git update-server-info
```

change origin

```
git remote -v
```

```
git remote set-url origin http://git.csgalileo.org/livenet-server.git
```

check origin

```
git remote show origin
```

if prune needed (on server restart apache if dav is used)

```
git remote prune origin
```

```
git push --set-upstream origin master
```

create branch with current changes

```
git checkout -b newBranch
```

tag

```
git tag -a v1.1 -m "versione 1.1"
git push origin v1.1
```

new remote branch

reference

create locally

```
git branch [name_of_your_new_branch]
git checkout [name_of_your_new_branch]
```

create or update remotly

```
git push --set-upstream origin [name_of_branch]
```

view remote branches

```
git fetch -v
```

fetch remote branches

```
git checkout -b [branch] origin/[branch]
```

delete remote branch

```
git push origin --delete [branch]

# and locally
git branch -d [branch]
```

rename branch (local and remote)

```
git branch -m new-name

# Delete the old-name remote branch and push the new-name local branch.
git push origin :old-name new-name

# Reset the upstream branch for the new-name local branch.
git push origin -u new-name
```

Preserve specific files while merging

start by defining a merge driver that would always favor our current version of the file

```
git config --global merge.ours.driver true
```

into repo mark examplefile.json to use that driver instead of the standard one:

```
echo 'examplefile.json merge=ours' >> .gitattributes
```

integrate changes from one branch into another branch

merging

from feature branch to master

```
git checkout feature
git merge master
```

or rebasing

```
git checkout feature
git rebase -i master

# if remote feature is present force push on origin
git push --force origin feature
```

rebase remote branch (or original fork) with local branch

add forked repo if none (git remote -v)

```
git remote add upstream https://github.com/pjreddie/darknet.git
```

rebase remote on master

```
git fetch upstream
git checkout master
git merge upstream/master
git push
```

merge specific commit from branch to another

```
git co <branch-target>
git cherry-pick <commit>
```

Home Assistant

```
git clone https://github.com/scipioni/home-assistant.git
cd home-assistant
git remote add upstream https://github.com/home-assistant/home-assistant.git
git pull upstream dev
```

apache git

```
apt install apache2
a2enmod dav_lock dav dav_fs macro
systemctl restart apache2
```

submodule

```
cd <submodule>
```

```
git checkout master
git branch -u origin/master master
cd <parent>
git config -f .gitmodules submodule.<submodule>.branch master
```

From:
https://wiki.csgalileo.org/ - **Galileo Labs**

Permanent link:
https://wiki.csgalileo.org/tips/git

Last update: **2019/03/13 09:30**