* SSH - konfiguracja
* Inicjowanie lokalnego repozytorium oraz zdalnego
* Tworzenie / poruszanie się po branchy
* Modyfikacja pliku na masterze, stworzenie nowej branchy, wysłanie do zdalnego repozytorium commitów
* GIT RESTORE PRZYKŁAD
* Komendy
* REBASE

# SSH - konfiguracja

Doniak@Laptok MINGW64 ~ (master)

$ cd ~

Doniak@Laptok MINGW64 ~ (master)

$ mkdir .ssh

Doniak@Laptok MINGW64 ~ (master)

$ cd .ssh

Doniak@Laptok MINGW64 ~/.ssh (master)

$ ssh-keygen

Generating public/private rsa key pair.

Enter file in which to save the key (/c/Users/Doniak/.ssh/id\_rsa):

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /c/Users/Doniak/.ssh/id\_rsa

Your public key has been saved in /c/Users/Doniak/.ssh/id\_rsa.pub

The key fingerprint is:

SHA256:NgmjgsZZNTj66odO+VC40va9Om+gPsUPLbvdh+Sw2YI Doniak@Laptok

The key's randomart image is:

+---[RSA 3072]----+

| .o |

| o. . |

| ... o |

|.o+ . o . |

|.=oo.. S |

|..+o\* o... |

|.==o B B . |

|o+=oEo\*.+ . |

|o+oo+\*=o.. |

+----[SHA256]-----+

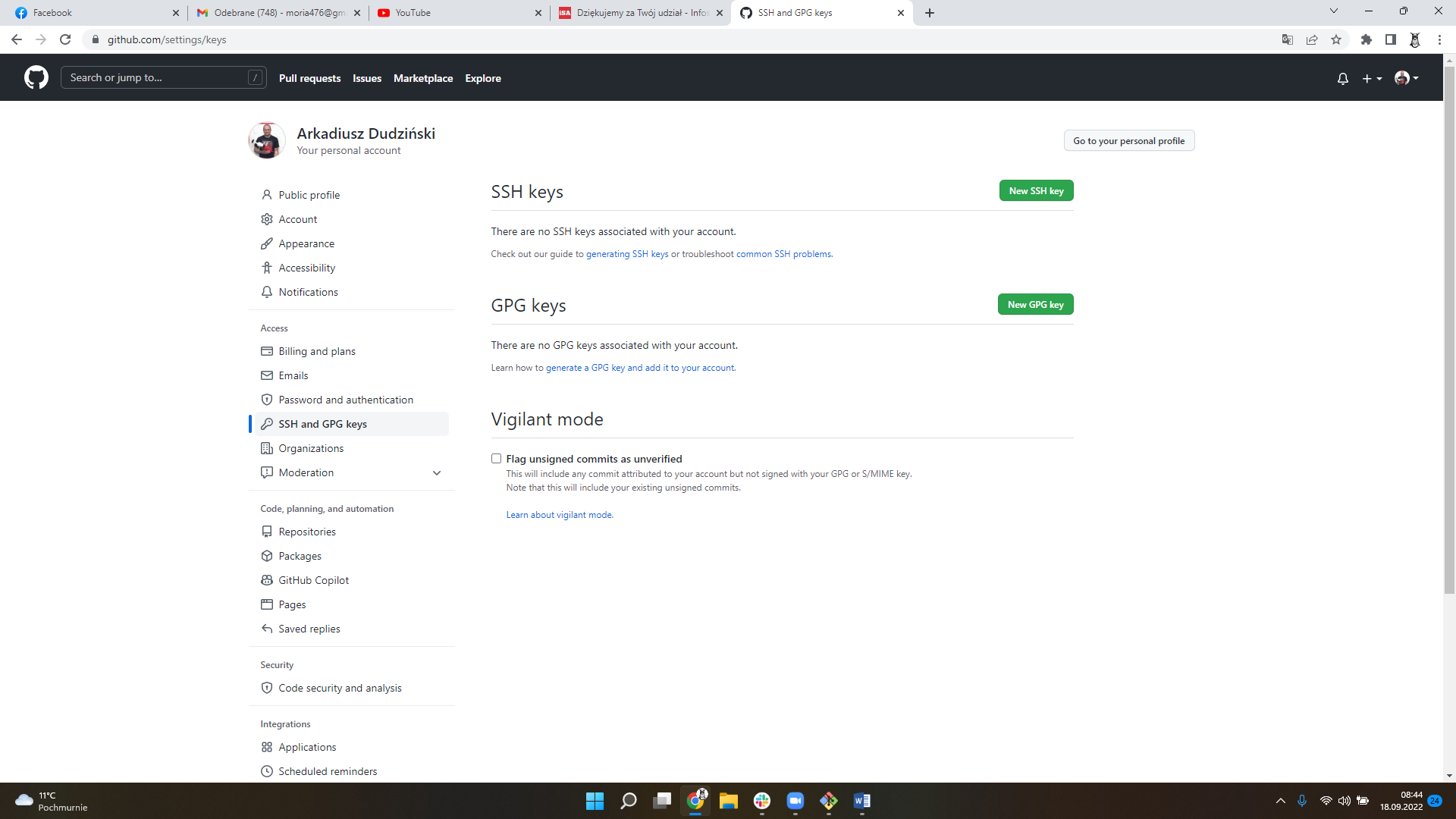
$ ls

id\_rsa id\_rsa.pub

Doniak@Laptok MINGW64 ~/.ssh (master)

$ cat id\_rsa.pub

ssh-rsa  Doniak@Laptok



Doniak@Laptok MINGW64 ~/.ssh (master)

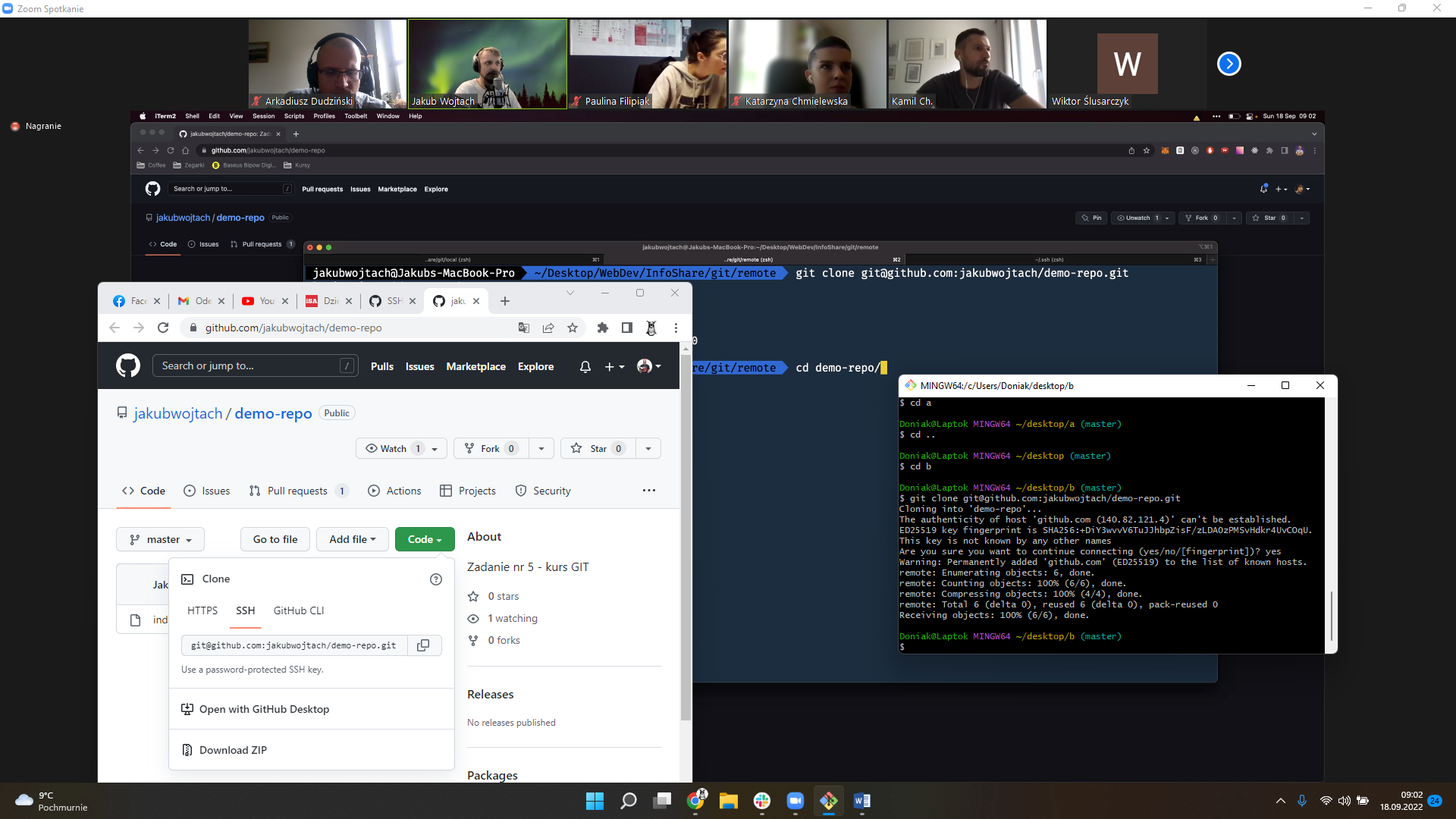
$ eval $(ssh-agent -s)

Agent pid 2007

Doniak@Laptok MINGW64 ~/.ssh (master)

$ ssh-add id\_rsa

Identity added: id\_rsa (Doniak@Laptok)



# Inicjowanie lokalnego repozytorium oraz zdalnego

Git init

Doniak@Laptok MINGW64 ~/desktop/d (master)

$ git remote add origin git@github.com:Moria476/zad2.git

Doniak@Laptok MINGW64 ~/desktop/d (master)

$ git remote -v

origin git@github.com:Moria476/zad2.git (fetch)

origin git@github.com:Moria476/zad2.git (push)

W rozróżnieniu od clona to tutaj nie pobiera/nadpisuje plikow lokalnie

# Tworzenie / poruszanie się po branchy

Git pull pobiera i merguje

Git fetch pobiera ale nie merguje

Doniak@Laptok MINGW64 ~/desktop/b/demo-repo (master)

$ git fetch

From github.com:jakubwojtach/demo-repo

\* [new branch] demo-branch -> origin/demo-branch

Doniak@Laptok MINGW64 ~/desktop/b/demo-repo (master)

$ git status

On branch master

Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

Doniak@Laptok MINGW64 ~/desktop/b/demo-repo (master)

$ git checkout demo-branch

Switched to a new branch 'demo-branch'

branch 'demo-branch' set up to track 'origin/demo-branch'.

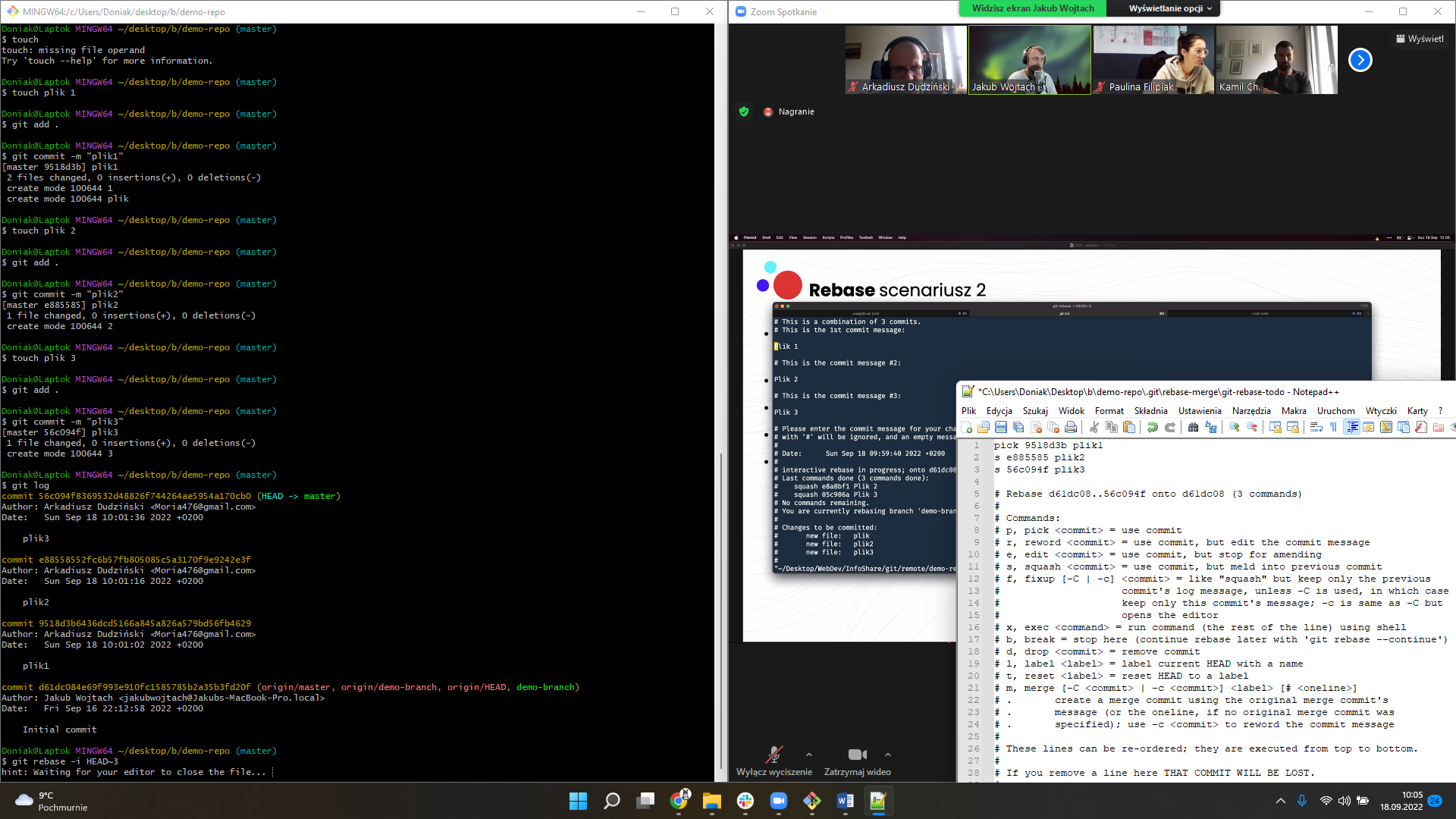
Doniak@Laptok MINGW64 ~/desktop/b/demo-repo (demo-branch)

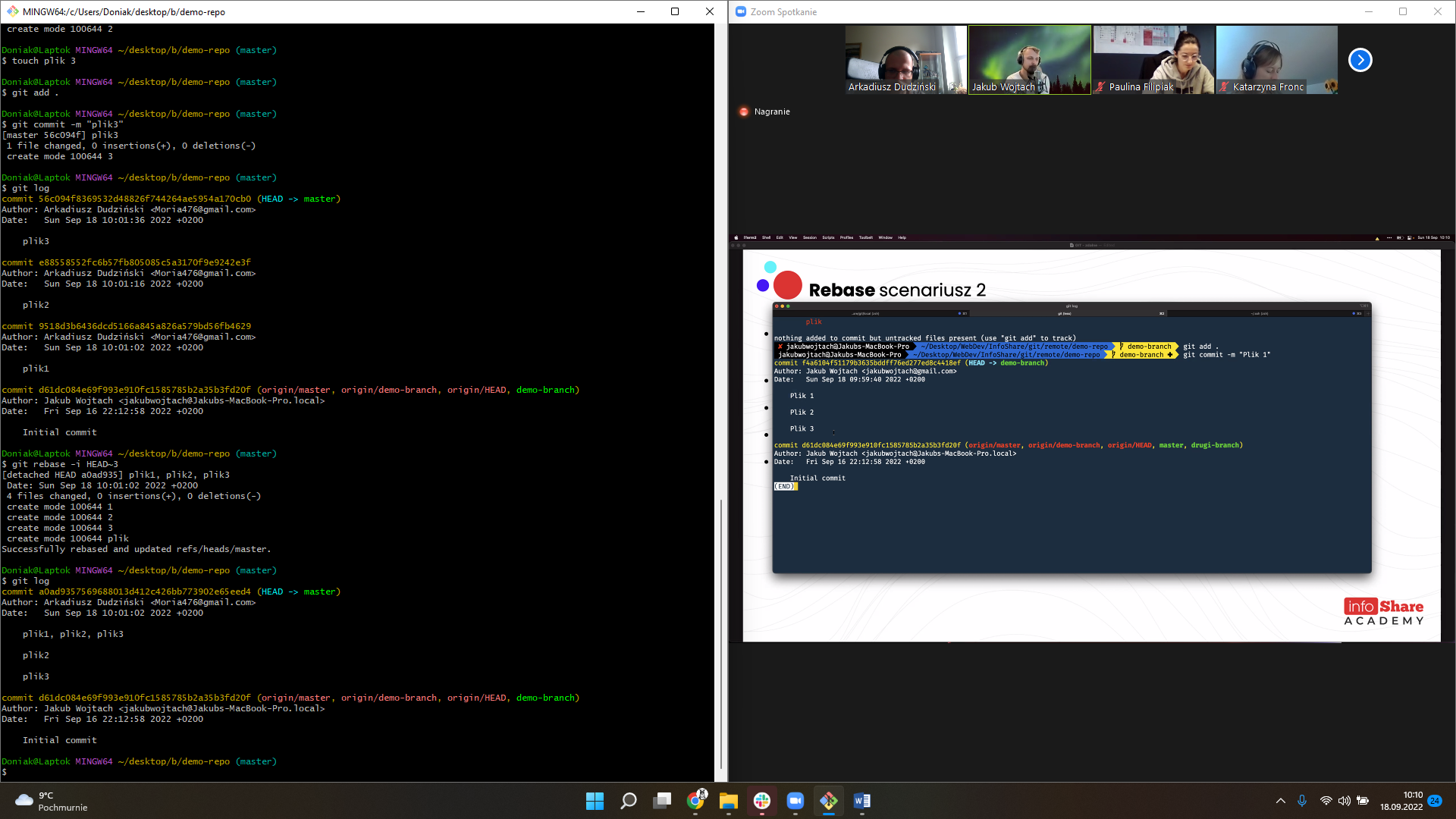
$ git checkout master

Switched to branch 'master'

Your branch is up to date with 'origin/master'.

Doniak@Laptok MINGW64 ~/desktop/b/demo-repo (master)





# Modyfikacja pliku na masterze, stworzenie nowej branchy, wysłanie do zdalnego repozytorium commitów

Doniak@Laptok MINGW64 ~/desktop/b/demo-repo (master)

$ code .

Doniak@Laptok MINGW64 ~/desktop/b/demo-repo (master)

$ git checkout -b student/arkadiusz-dudzinski

Switched to a new branch 'student/arkadiusz-dudzinski'

Doniak@Laptok MINGW64 ~/desktop/b/demo-repo (student/arkadiusz-dudzinski)

$ git add .

Doniak@Laptok MINGW64 ~/desktop/b/demo-repo (student/arkadiusz-dudzinski)

$ git status

On branch student/arkadiusz-dudzinski

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: ArkadiuszDudzinski.txt

modified: index.html

new file: plik.txt

Doniak@Laptok MINGW64 ~/desktop/b/demo-repo (student/arkadiusz-dudzinski)

$ git commit -m "Arkadiusz Dudziński robi brancha"

[student/arkadiusz-dudzinski 9206053] Arkadiusz Dudziński robi brancha

3 files changed, 4 insertions(+), 1 deletion(-)

create mode 100644 ArkadiuszDudzinski.txt

create mode 100644 plik.txt

Doniak@Laptok MINGW64 ~/desktop/b/demo-repo (student/arkadiusz-dudzinski)

$ git log

commit 9206053f6765699fcdf2cc0443b78b73f717337e (HEAD -> student/arkadiusz-dudzinski)

Author: Arkadiusz Dudziński <Moria476@gmail.com>

Date: Sun Sep 18 10:28:35 2022 +0200

Arkadiusz Dudziński robi brancha

commit a0ad9357569688013d412c426bb773902e65eed4 (master)

Author: Arkadiusz Dudziński <Moria476@gmail.com>

Date: Sun Sep 18 10:01:02 2022 +0200

plik1, plik2, plik3

plik2

plik3

commit d61dc084e69f993e910fc1585785b2a35b3fd20f (origin/master, origin/demo-branch, origin/HEAD, demo-branch)

Author: Jakub Wojtach <jakubwojtach@Jakubs-MacBook-Pro.local>

Date: Fri Sep 16 22:12:58 2022 +0200

Initial commit

Doniak@Laptok MINGW64 ~/desktop/b/demo-repo (student/arkadiusz-dudzinski)

$ git push origin student/arkadiusz-dudzinski

Enumerating objects: 10, done.

Counting objects: 100% (10/10), done.

Delta compression using up to 12 threads

Compressing objects: 100% (5/5), done.

Writing objects: 100% (8/8), 768 bytes | 768.00 KiB/s, done.

Total 8 (delta 2), reused 1 (delta 0), pack-reused 0

remote: Resolving deltas: 100% (2/2), completed with 1 local object.

remote:

remote: Create a pull request for 'student/arkadiusz-dudzinski' on GitHub by visiting:

remote: https://github.com/jakubwojtach/demo-repo/pull/new/student/arkadiusz-dudzinski

remote:

To github.com:jakubwojtach/demo-repo.git

\* [new branch] student/arkadiusz-dudzinski -> student/arkadiusz-dudzinski

Doniak@Laptok MINGW64 ~/desktop/b/demo-repo (student/arkadiusz-dudzinski)

# GIT RESTORE PRZYKŁAD

EXAMPLES

The following sequence switches to the master branch, reverts the Makefile to two revisions back, deletes hello.c by mistake, and gets it back from the index.

$ git switch master

$ git restore --source master~2 Makefile (1)

$ rm -f hello.c

$ git restore hello.c (2)

take a file out of another commit

restore hello.c from the index

# Komendy

|  |  |
| --- | --- |
| Nawigacja | |
| ls | Wyświetla pliki w katalogu. –a ukryte – l dodatkowe info |
| mkdir | Tworzy folder |
| touch | Tworzy plik |
| echo "to jest moj tekst" >> tekst.txt | tworzy tekst w pliku lub jesli tego pliku nie ma to go tworzy |
| cd | Podać ścieżkę lub .. lub cd../../.. |
| rm –rf [nazwa folderu] | Usuwa folder |
| rm [nazwa pliku] | Usuwa plik |
| cat[nazwa pliku] | Podgląda co jest w pliku |
| mv[nazwa pliku] [ścieżka] | przesun plik |
| find ./ -name '\*.html' | wyszukuje wszystkie pliki z rozszerzeniem .html |
| code [nazwa pliku] lub . | Odpali visual studio |
| clear lub ctrl + k | Czyści ekran |
| Q lub ctrl + c | exit |
| Git | |
| git init | Tworzy lokalne repozytorium |
| git add | Dodaje pliki do staged |
| git reset | cofa pliki ze staged || HEAD~3 cofa o trzy |
| git commit -m "" | Commituje pliki ze staged. –amend zmieni komentarz |
| git log | wyświetla historie na danym banchu --graph |
| git status | sprawdza co się dzieje lokalnie |
| git branch | Tworzy branche || -d usunię branche po push. –D przed push |
| git checkout | Przechodzi między branchami || -b tworzy od razu branche |
| git merge (branch) | Zcala branch || --no ff zcala branch ale bez zakańczania jej historii |
| git push origin master | (wypchniecie commita do githuba) ewentulanie z -f |
| git clone (adres) | Jak uzyskać lokalną kopię istniejącego repozytorium do pracy. Zwykle jest on używany tylko raz dla danego repozytorium |
| git pull | (lub git fetch + git merge) w ten sposób zaktualizujesz tę lokalną kopię nowymi zatwierdzeniami ze zdalnego repozytorium. |
| git remote add origin [adres] | Dodaje repozytorium z githuba |
| git remote -v | Pokazuje czy mamy remota |
| git restore | cofniecie zmian (przykład wyżej) |
| git revert | tworzy commit by cofnac commit ktory puscilismy |

# REBASE

Doniak@Laptok MINGW64 ~

$ cd desktop

Doniak@Laptok MINGW64 ~/desktop

$ mkdir git-test

Doniak@Laptok MINGW64 ~/desktop (master)

$ cd git-test

Doniak@Laptok MINGW64 ~/desktop/git-test

$ git init

Initialized empty Git repository in C:/Users/Doniak/Desktop/git-test/.git/

Doniak@Laptok MINGW64 ~/desktop/git-test (master)

$ git remote add origin git@github.com:Moria476/git-test.git

Doniak@Laptok MINGW64 ~/desktop/git-test (master)

$ mkdir test

Doniak@Laptok MINGW64 ~/desktop/git-test (master)

$ cd test

Doniak@Laptok MINGW64 ~/desktop/git-test/test (master)

$ echo temp.txt >>temp.txt

Doniak@Laptok MINGW64 ~/desktop/git-test/test (master)

$ git add .

Doniak@Laptok MINGW64 ~/desktop/git-test/test (master)

$ git commit -m "temp.txt is created"

[master (root-commit) 9972b51] temp.txt is created

1 file changed, 1 insertion(+)

create mode 100644 test/temp.txt

Doniak@Laptok MINGW64 ~/desktop/git-test/test (master)

$ git push origin master

Enumerating objects: 4, done.

Counting objects: 100% (4/4), done.

Writing objects: 100% (4/4), 278 bytes | 278.00 KiB/s, done.

Total 4 (delta 0), reused 0 (delta 0), pack-reused 0

To github.com:Moria476/git-test.git

\* [new branch] master -> master

Doniak@Laptok MINGW64 ~/desktop/git-test/test (master)

$ echo local >> temp.txt

Doniak@Laptok MINGW64 ~/desktop/git-test/test (master)

$ git commit -m "local"

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: temp.txt

no changes added to commit (use "git add" and/or "git commit -a")

Doniak@Laptok MINGW64 ~/desktop/git-test/test (master)

$ git add .

warning: in the working copy of 'test/temp.txt', LF will be replaced by CRLF the next time Git touches it

Doniak@Laptok MINGW64 ~/desktop/git-test/test (master)

$ git commit -m "local"

[master 3101374] local

1 file changed, 1 insertion(+)

Doniak@Laptok MINGW64 ~/desktop/git-test/test (master)

$ git push origin master

To github.com:Moria476/git-test.git

! [rejected] master -> master (fetch first)

error: failed to push some refs to 'github.com:Moria476/git-test.git'

hint: Updates were rejected because the remote contains work that you do

hint: not have locally. This is usually caused by another repository pushing

hint: to the same ref. You may want to first integrate the remote changes

hint: (e.g., 'git pull ...') before pushing again.

hint: See the 'Note about fast-forwards' in 'git push --help' for details.

Doniak@Laptok MINGW64 ~/desktop/git-test/test (master)

$ git fetch origin

remote: Enumerating objects: 7, done.

remote: Counting objects: 100% (7/7), done.

remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0

Unpacking objects: 100% (4/4), 678 bytes | 37.00 KiB/s, done.

From github.com:Moria476/git-test

9972b51..0a19b59 master -> origin/master

Doniak@Laptok MINGW64 ~/desktop/git-test/test (master)

$ git rebase origin/master

Auto-merging test/temp.txt

CONFLICT (content): Merge conflict in test/temp.txt

error: could not apply 3101374... local

hint: Resolve all conflicts manually, mark them as resolved with

hint: "git add/rm <conflicted\_files>", then run "git rebase --continue".

hint: You can instead skip this commit: run "git rebase --skip".

hint: To abort and get back to the state before "git rebase", run "git rebase --abort".

Could not apply 3101374... local

Doniak@Laptok MINGW64 ~/desktop/git-test/test (master|REBASE 1/1)

$ alias conflict='git diff --name-only --diff-filter=U'

Doniak@Laptok MINGW64 ~/desktop/git-test/test (master|REBASE 1/1)

$ conflict

test/temp.txt

Doniak@Laptok MINGW64 ~/desktop/git-test/test (master|REBASE 1/1)

$ nano temp.txt

Doniak@Laptok MINGW64 ~/desktop/git-test/test (master|REBASE 1/1)

$ git add temp.txt

Doniak@Laptok MINGW64 ~/desktop/git-test/test (master|REBASE 1/1)

$ git rebase --continue

[detached HEAD 03a94ac] rebasing

1 file changed, 1 insertion(+)

Successfully rebased and updated refs/heads/master.

Doniak@Laptok MINGW64 ~/desktop/git-test/test (master)

$ git push origin master

Enumerating objects: 7, done.

Counting objects: 100% (7/7), done.

Writing objects: 100% (4/4), 329 bytes | 329.00 KiB/s, done.

Total 4 (delta 0), reused 0 (delta 0), pack-reused 0

To github.com:Moria476/git-test.git

0a19b59..03a94ac master -> master