

# Instrukcja do ćwiczenia

## git-annex

Michał Liszcz      Jakub Sawicki

31 maja 2016

wersja dla prowadzącego

- 1 Skonfiguruj dwie maszyny wirtualne **centos6** z zainstalowanymi następującymi paczkami (git-annex dostępny jest w repozytorium EPEL <sup>1</sup>, pozostałe są w standardowych repozytoriach):

- `git >= 1.7.1`
- `git-annex >= 3.20120522`

Dodaj w pliku `/etc/hosts` wpisy, by maszyny były dostępne pod nazwami `hostA` i `hostB`.

```
yum upgrade ca-certificates
```

```
wget http://download.fedoraproject.org/pub/epel/6/x86_64/ \
    epel-release-6-8.noarch.rpm
rpm -ivh epel-release-6-8.noarch.rpm
```

```
hostA:~$ ssh-keygen
hostA:~$ cat 'root123' | ssh-copy-id root@hostB
```

```
hostB:~$ ssh-keygen
hostB:~$ cat 'root123' | ssh-copy-id root@hostA
```

- 2 Na obu maszynach skonfiguruj serwer SSH, utwórz użytkownika `git` oraz utwórz dla niego parę kluczy (**ssh-keygen**) Wymień klucze publiczne między maszynami (**ssh-copy-id**) tak, by było możliwe logowanie przy ich pomocy.

Sprawdź czy z każdej maszyny możesz zalogować się na drugą z nich:

```
hostA:~$ ssh hostB
hostB:~$ ssh hostA
```

- 3 Utwórz na obu maszynach repozytoria `hostA-main` i `hostB-main` (gdzie prefiks oznacza maszynę, na której dane repozytorium powinno się znajdować). w każdym z repozytoriów zainicjalizuj `git-annex` i dodaj drugie repozytorium jako `remote`.

---

<sup>1</sup>CentOS6.2 dostępny w laboratorium nie posiada go w repozytorium. Konieczne jest ściągnięcie i zainstalowanie paczki `epel-release-6-8.noarch`. Pomocne może być także uaktualnienie certyfikatów: `yum upgrade ca-certificates`.

```

hostA:~$ git init hostA-main.git
hostA:~$ cd hostA-main.git
hostA:hostA-main.git$ git annex init 'hostA - main'
...
hostA:hostA-main.git$ git remote add hostB-main git@hostB:hostB-main.git

hostA:~$ git init hostA-main.git
hostA:~$ cd hostA-main.git
hostA:hostA-main.git$ git annex init 'hostA - main'

hostB:~$ git init hostB-main.git
hostB:~$ cd hostB-main.git
hostB:hostB-main.git$ git annex init 'hostB - main'
hostB:hostB-main.git$ git remote add hostA-main root@hostA:hostA-main.git
hostB:hostB-main.git$ git pull hostA-main master # nie zadziala dopoki nie ma commi

hostA:hostA-main.git$ git remote add hostB-main root@hostB:hostB-main.git

```

- 4 W repozytorium `hostA-main` utwórz plik `important_file.txt` i dodaj go do indeksu git-annex. Uzyskaj dostęp do zawartości pliku na maszynie `hostB`. Zweryfikuj gdzie znajduje się aktualnie plik z `hostA` i `hostB`.

```

hostA:hostA-main.git$ echo "important big file" > important_file.txt
hostA:hostA-main.git$ git annex add important_file.txt
hostA:hostA-main.git$ git commit -m "important_file"

hostB:hostB-main.git$ git annex sync
hostB:hostB-main.git$ ll
total 4
lrwxrwxrwx 1 git git 178 May 19 10:18 important_file.txt -> .git/annex/objects/3J/Z
hostB:hostB-main.git$ cat important_file.txt
cat: important_file.txt: No such file or directory
hostB:hostB-main.git$ git annex get important_file.txt
hostB:hostB-main.git$ cat important_file.txt
important big file
hostB:hostB-main.git$ git annex whereis important_file.txt
whereis important_file.txt (2 copies)
    bad498fb-e968-467a-98a8-c59fd7800932 — hostA-main (hostA - main)
    f0c9f4ba-7550-40ce-9eb0-a325b991d306 — here (hostB - main)
ok

hostA:hostA-main.git$ git annex whereis important_file.txt
whereis important_file.txt (1 copy)
    bad498fb-e968-467a-98a8-c59fd7800932 — here (hostA - main)
ok
hostA:hostA-main.git$ git annex sync
hostA:hostA-main.git$ git annex whereis important_file.txt
whereis important_file.txt (2 copies)
    bad498fb-e968-467a-98a8-c59fd7800932 — here (hostA - main)
    f0c9f4ba-7550-40ce-9eb0-a325b991d306 — hostB-main (hostB - main)
ok

```

- 5 Zmodyfikuj zawartość pliku w `hostB-main` i zweryfikuj jak rozprzestrzeniają się zmiany.

```

hostB:hostB-main.git$ echo " modified" >> important_file.txt
bash: important_file.txt: Permission denied
hostB:hostB-main.git$ git annex unlock important_file.txt
hostB:hostB-main.git$ ll
total 4
-rw-r--r-- 1 git git 20 May 19 10:55 important_file.txt
hostB:hostB-main.git$ echo " modified" >> important_file.txt
hostB:hostB-main.git$ git commit important_file.txt -m "Modified important_file.txt"
hostB:hostB-main.git$ ll
total 4
lrwxrwxrwx 1 git git 178 May 19 10:55 important_file.txt -> .git/annex/objects/wG/2
hostB:hostB-main.git$ git annex whereis important_file.txt
whereis important_file.txt (1 copy)
    f0c9f4ba-7550-40ce-9eb0-a325b991d306 — here (hostB - main)
ok

hostA:hostA-main.git$ git annex whereis important_file.txt
whereis important_file.txt (2 copies)
    bad498fb-e968-467a-98a8-c59fd7800932 — here (hostA - main)
    f0c9f4ba-7550-40ce-9eb0-a325b991d306 — hostB-main (hostB - main)
ok
hostA:hostA-main.git$ git annex sync
hostA:hostA-main.git$ git annex whereis important_file.txt
whereis important_file.txt (1 copy)
    f0c9f4ba-7550-40ce-9eb0-a325b991d306 — hostB-main (hostB - main)
ok

```

- 6 Utwórz kolejne repozytorium **hostB-backup**. Powinno to być repozytorium, które przechowywane będzie na dysku backupowym **/dev/sdb** (utwórz odpowiednią partycję i zamontuj w **/mnt/backup**).

```

hostB$ fdisk /dev/sdb -> utworzenie nowej partycji
hostB$ mkfs.ext4 /dev/sdb1
hostB$ mkdir /mnt/backup
hostB$ mount /dev/sdb1 /mnt/backup

```

```

hostB:/mnt/backup$ git init hostB-backup.git
hostB:/mnt/backup$ cd hostB-backup.git
hostB:hostB-backup.git$ git annex init 'hostB - backup'

```

```

hostB:hostB-main.git$ git remote add hostB-backup /mnt/backup/hostB-backup.git
hostB:hostB-main.git$ git annex sync

```

- 7 Utwórz plik **plik.bin** o wielkości 500MB w **hostB-main**. Zbadaj szybkość transferu przy kopiowaniu pliku do **hostA-main** (przez SSH) i do **hostB-media** (kopia lokalna).

	prędkość transferu [MB/s]
przez SSH	
kopia lokalna	

```

hostB:hostB-main.git$ dd if=/dev/zero of=large.bin bs=1024 count=500000
hostB:hostB-main.git$ git annex add large.bin
hostB:hostB-main.git$ git commit -m "Added large.bin"

```

Przykładowe wyniki:

```
[root@lab429-08 hostB-main.git]# git annex copy --to hostA-main plik.bin
copy plik.bin (checking hostA-main...) (to hostA-main...) SHA256-s512000000--4c98cf638799a07eb85872e7f5f5d8
512000000 100\% 45.98MB/s 0:00:10 (xfer#1, to-check=0/1)
```

```
sent 512062644 bytes received 31 bytes 48767873.81 bytes/sec
total size is 512000000 speedup is 1.00
ok
(Recording state in git...)
```

```
[root@lab429-08 hostB-main.git]# git annex copy --to hostB-backup plik.bin
copy plik.bin (to hostB-backup...) SHA256-s512000000--4c98cf638799a07eb85872e7f5f5d8
512000000 100\% 115.10MB/s 0:00:04 (xfer#1, to-check=0/1)
```

```
sent 512062644 bytes received 31 bytes 113791705.56 bytes/sec
total size is 512000000 speedup is 1.00
ok
(Recording state in git...)
```

- 8 Zbadaj działanie flagi `numcopies`. Zwróć uwagę na to, że wersja `git-annex` dostępna w repozytorium `centos6` nie implementuje jeszcze polecenia `numcopies`.

- Ustaw flagę `numcopies` na 2.
- Następnie utwórz w wybranym repozytorium bardzo ważny plik (`my_data.txt`) i skopiuj go do jednego z pozostałych repozytoriów, tak by w systemie były dwie kopie.
- Spróbuj porzucić plik lokalnie i obserwuj wyniki.

```
hostB:hostB-main.git$ git config annex.numcopies 2 # config sie nie propaguje przy
hostB:hostB-main.git$ git annex copy important_file.txt --to hostA-main
hostB:hostB-main.git$ git annex drop important_file.txt
drop important_file.txt (checking hostA-main...) (unsafe)
Could only verify the existence of 1 out of 2 necessary copies
```

```
No other repository is known to contain the file.
```

```
(Use --force to override this check, or adjust annex.numcopies.)
failed
git-annex: drop: 1 failed
```

- 9 Umieść plik z poprzedniego punktu w dwóch zdalnych repozytoriach i porzuć lokalną kopię. Spróbuj pobrać ją z flagą `get -auto`. Zasymuluj awarię jednego ze zdalnych repozytoriów a następnie usuń je z indeksu `git-annex`. Sprawdź ile kopii pliku istnieje w systemie. Czy teraz możesz pobrać go do lokalnego repozytorium z użyciem flagi `-auto`?

```
[root@lab429-08 hostB-main.git]# git annex get --auto plik.bin
get plik.bin (from hostB-backup...) SHA256-s512000000--4c98cf638799a07eb85872e7f5f5d8
512000000 100\% 56.26MB/s 0:00:08 (xfer#1, to-check=0/1)
```

```
sent 512062644 bytes received 31 bytes 53901334.21 bytes/sec
total size is 512000000 speedup is 1.00
ok
(Recording state in git...)
```

- 10 Zasymuluj zmianę którejś z kopii pliku `plik.bin`. Upewnij się, że posiadasz co najmniej dwie kopie i zmodyfikuj zawartość pliku bezpośrednio w systemie plików. Wykorzystaj `git-annex` do wykrycia niepoprawnej kopii, jej usunięcia z systemu i odtworzenia poprawnej kopii.

```
hostB-backup.git$ echo "asdfasd" >> .git/annex/objects/aa/SHA256-s512000000--4c98cf638799a07eb85872e7f5f5d8
```

```
hostB-main.git$ git annex fsck --from hostB-backup plik.bin
fsck plik.bin Bad file size (7 B larger); dropped from hostB-backup
```

```
Only 1 of 2 trustworthy copies exist of plik.bin
```

```
Back it up with git-annex copy.
```

```
failed
```

```
(Recording state in git...)
```

```
git-annex: fsck: 1 failed
```

```
hostB-main.git$ git annex get --auto plik.bin
```

```
get plik.bin (from hostA-main...) SHA256-s512000000--4c98cf638799a07eb85872e7f5f5d8
512000000 100% 45.54MB/s 0:00:10 (xfer#1, to-check=0/1)
```

```
sent 30 bytes received 512062649 bytes 44527189.48 bytes/sec
```

```
total size is 512000000 speedup is 1.00
```

```
ok
```

```
(Recording state in git...)
```

- 11 **Zadanie dodatkowe.** Załóż konto w serwisie <https://gitlab.com><sup>2</sup>. Gitlab wspiera tylko autentykację przy użyciu klucza publicznego. Dodaj w panelu ustawień konta klucz wygenerowany na początku ćwiczenia. Utwórz repozytorium i dodaj je jako remote do istniejącego repozytorium. Zsynchronizuj status git-annex. Zbadaj szybkość transferu 50MB pliku do Gitlab.com, do drugiej maszyny w sieci lokalnej i do repozytorium na tej samej maszynie.

	prędkość transferu [MB/s]
przez SSH	
kopia lokalna	
Gitlab.com	

<sup>2</sup>Gitlab.com oferuje 10GB darmowego miejsca na przechowywanie plików z użyciem systemu git-annex