

## Darbas su versionavimu

### Žingsniai atliekami pirmą kart prisijungus:

1. Prisijungiame prie VU VPN tinklo. VU VPN tinklo konfigūravimas.
2. Einame į `git.mif.vu.lt`. Prisijungiame su mif vartotojo vardu. Sukuriame naują projektą.

Project name

projektas

Project URL Project slug

https://git.mif.vu.lt/lipe8679/ projektas

Want to house several dependent projects under the same namespace? [Create a group](#).

Project description (optional)

Description format

Visibility Level ⓘ

Private  
Project access must be granted explicitly to each user.

Internal  
The project can be accessed by any logged in user.

Public  
The project can be accessed without any authentication.

Initialize repository with a README  
Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.

Create project

3. Pirmą kartą prisijungus prie aplinkos reikia pridėti savo įrenginio viešą prisijungimo raktą, kad galėtumėte lengvai sukelti/pasiimti savo programą iš gitlab aplinkos. Tam reikia susikurti savo viešą raktą. Norėdami tai atlikti sekite instrukcijas: **žr. šią nuorodą**. Kurdami naudokite savo studento e-paštą. (Windows aplinkoje terminalą galite išsikviesti **anaconda navigator** → **JupyterLab** → **terminal** )

4. Susikūrus raktą (sukuriami privatus jo viešinti negalima, bei viešas su galūne .pub) jį reikia nukopijuoti ir pridėti į puslapį **žr. puslapį**. Nueikite į katalogą `C:\Users\\.ssh` bei nukopijuokite raktą iš failo `id_ed25519.pub` į puslapį.

5. Pirmą kartą susikurkite ir failą `config` kataloge `C:\Users\\.ssh` su informacija:

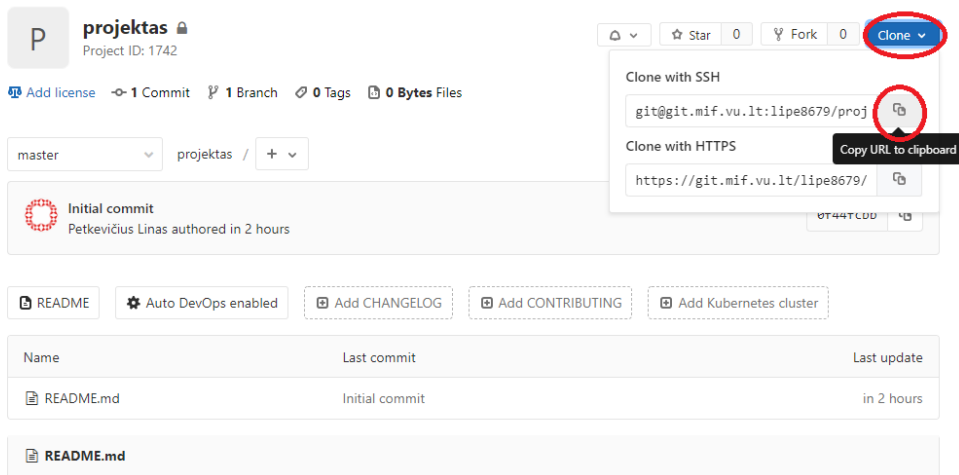
```
Host git.mif.vu.lt
  PubkeyAuthentication yes
  IdentityFile ~/.ssh/id_ed25519
```

6. Tai atlikus ir terminale įvykdžius komandą `ssh -T git@git.mif.vu.lt` pamatysite pasisveikinimo pranešimą.

7. Paprastesniam darbui naudosisime programėlę Desktop Github darbui su versionavimu. Ją galite atsisiųsti iš čia.

8. Paleiskite Desktop Github programą ir importuokime sukurtą projektą:

- Spaudžiame **Clone repository from the internet ...**
- Pasirenkame URL



- Projekte nukopijuojame jo adresą, ir nurodome jį, spaudžiame Clone

Kataloge C:\Users\\Documents\GitHub\projektas atsirado README.md failas, taigi galime pradėti dirbti. Visi failai esantys tame kataloge bus versionuojami. Programas patartina laikyti ir pildyti iš šio katalogo.

## Darbas su Github Desktop:

Pagrindinių mygtukų reikšmės

- Changes - Peržiūrėti, kurie failai yra versijuojami (ir yra pakeisti), bei kurie nėra versijuojami.
- Commit to master - Įvykio fiksavimas į žurnalą (lokalų). **Visada prie description parašykite ką naujo programoje padarėte.**
- Push Origin - Perkelti pakeitimus į git.mif.vu.lt serverį, tam kad vėliau būtų galima pasiekti iš bet kur.
- Fetch Origin - atsisiųsti atnaujinimus iš gitlab serverio į lokalų kompiuterį (jie neatnaujina programos).
- Pull Origin - pakeitimus sujungti su esamu programos kodu (esama programa gali būti pakeista).

## Darbas su versionavimu terminale:

- git clone <repozitorija> - Atsisiųsti projektą lokaliai.
- git status - Peržiūrėti, kurie failai yra versijuojami (ir yra pakeisti), bei kurie nėra versijuojami.
- git add <failas> - Pridėti failą į lokalią versionavimo duomenų bazę. pvz: git add failas1.ipynb.
- git commit -a -m "<zinute kas padaryta>" - Įvykio fiksavimas į žurnalą (lokalų).
- git push - Perkelti pakeitimus į git.mif.vu.lt, tam kad vėliau būtų galima pasiekti iš betkur.
- git fetch - atsisiųsti atnaujinimus iš bendros duomenų bazės.
- git pull - pakeitimus sujungti su esamu programos kodu.
- git checkout <saka/versija> - pasiimti specifinę kodo versiją iš atšakos pvz: git checkout master, arba specifinę kodo versiją git checkout <hash>.