

Лабораторная работа 2

Понкратова Х. А. НПМбд-02-20

Цель работы:

Изучить идеологию и применение средств контроля версий.

Ход работы:

- Установить Git
- Создать аккаунт на Github.com
- При помощи SSH ключа подключиться к Github.com
- Создать репозиторий
- Создать и загрузить на репозиторий README.md
- Скачать и загрузить лицензионное соглашение
- Создать шаблоны игнорируемых при добавлении в репозиторий типов файлов
- Скачать git-flow
- Создать ветку выпуска
- Сделать релиз

Выполнение лабораторной работы

1. Устанавливаем Git при помощи команд `sudo apt install git`
2. Регистрируемся на Github.com
3. Создаем каталог на Git
4. При помощи SSH-keygen -C "Khristina Ponkratova cristinka02@gmail.com" мы получаем SSH-ключ

```
khristina@kaponkratova: ~  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
khristina@kaponkratova:~$ ls  
Видео      Загрузки    Музыка      'Рабочий стол'  
Документы  Изображения  Общедоступные  Шаблоны  
khristina@kaponkratova:~$ mkdir kaponkratova  
khristina@kaponkratova:~$ ls  
kaponkratova  Документы  Изображения  Общедоступные  Шаблоны  
Видео      Загрузки    Музыка      'Рабочий стол'  
khristina@kaponkratova:~$ ssh-keygen -C "Khristina Ponkratova cristinka02@gmail.  
com"
```

```
khristina@kaponkratova: ~/.ssh  
+---[RSA 3072]---+  
| ..  
|. E ..  
|.O. .+ o + .  
|+.000 = + o  
| O.o o S o  
|* = . ++. + .  
|. + o o o+=B  
| . o.==+o  
| ..oo+  
+---[SHA256]-----+  
khristina@kaponkratova:~$ cd .ssh  
khristina@kaponkratova:~/.ssh$ ls  
id_rsa id_rsa.pub  
khristina@kaponkratova:~/.ssh$ cat id_rsa.pub  
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGCstYDc8ICRp7c054nrMeFqYzKX6DSoIh5e1fjSEJ65  
iytzsdnRbVY1v/W0bHg9E+kmEZ1E8/QBDY/U2WbvKmZUiMi2pqCTWQzC1n85CIyX6NPtZg6/3Uf2Dnt4  
2jClnmyF5WssXo4d323ih3qhstPb64MUNFeaSHFS3SMVEugDzy4UT5h2mC0afDdT3Srz/VE08FgrJ7jH  
Bvznz8H12QZrF+wEGEk0qxnDWDaWeQkjtVUPPKHUgcP6NCh6z3v75d6s40y98G80vMygXst2s7sf260K  
nkfNeMicIuE9yVZiKF32wLTdoipX4bZ0kwDJkxKF1xR259aRYjgC/kbSxo/Cs8hDACrOyqmg6zgjaGxE  
/AnFVLiOmWB/nnG/m4oSHinnkUuq++7JY9HtUhuoUyXG2PAkjCXsrwp+CFFyRlDP7iWJ18Baw+sdX/Nt  
yUmIXasiPunRgeTs0JYxLELz8EDFqWrdKPBfa+KKtaBN2rjRVjP/lDxe5WSnuJbMcVagl0s= Khristi  
na Ponkratova cristinka02@gmail.com  
khristina@kaponkratova:~/.ssh$
```

5. Открываем настройки Github

6. Вводим новый ключ

SSH keys

New SSH key

There are no SSH keys associated with your account.

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

7. В терминале создаем папку Laboratory и создаем файл README.md. Инициализируем Git с помощью команды Git init

```
khristina@kaponkratova:~/.ssh$ cd
khristina@kaponkratova:~$ cd kaponkratova/
khristina@kaponkratova:~/kaponkratova$ mkdir laboratory
khristina@kaponkratova:~/kaponkratova$ ls
laboratory
khristina@kaponkratova:~/kaponkratova$ cd laboratory
khristina@kaponkratova:~/kaponkratova/laboratory$ echo "#laboratory" >> README.md
khristina@kaponkratova:~/kaponkratova/laboratory$ git init
```

8. Выбираем файл README.md как изменение командой git add README.md
9. Комментируем изменения и отправляем в репозиторий
10. Добавляем лицензию, создаем и настраиваем ignore list

```
khristina@kaponkratova:~/kaponkratova/laboratory$ wget https://creativecommons.org/licenses/by/4.0/legalcode.txt
--2021-10-21 21:29:03-- https://creativecommons.org/licenses/by/4.0/legalcode.txt
Распознаётся creativecommons.org (creativecommons.org)... 104.20.150.16, 172.67.34.140, 104.20.151.16, ...
Подключение к creativecommons.org (creativecommons.org)|104.20.150.16|:443... соединение установлено.
HTTP-запрос отправлен. Ожидание ответа... 200 OK
```

11. Инициализируем git flow, настраиваем версию релиза

```
khristina@kaponkratova:~/kaponkratova/laboratory$ git flow init

Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main] v
Fatal: Local branch 'v' does not exist.
khristina@kaponkratova:~/kaponkratova/laboratory$ git flow init

Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [/home/khristina/kaponkratova/laboratory/.git/hooks]
khristina@kaponkratova:~/kaponkratova/laboratory$ git branch
* develop
  main
khristina@kaponkratova:~/kaponkratova/laboratory$ git flow release start 1.0.0
```

12. Заходим на Github и делаем релиз

Вывод:

Мы изучили идеологию и применение средств контроля версий.