

# **Отчёт по лабораторной работе №3**

**Первоначальна настройка git**

Бабенко Роман Игоревич

# Содержание

|   |                                |    |
|---|--------------------------------|----|
| 1 | Цель работы                    | 5  |
| 2 | Выполнение лабораторной работы | 6  |
| 3 | Выводы                         | 11 |

## Список иллюстраций

|      |  |    |
|------|--|----|
| 2.1  | Установка git . . . . .                    | 6  |
| 2.2  | . . . . .                                  | 6  |
| 2.3  | Зададим имя и email . . . . .              | 6  |
| 2.4  | . . . . .                                  | 6  |
| 2.5  | . . . . .                                  | 7  |
| 2.6  | Создание ключей . . . . .                  | 7  |
| 2.7  | Генерируем ключ . . . . .                  | 8  |
| 2.8  | Выводим список ключей . . . . .            | 8  |
| 2.9  | Копируем ключ . . . . .                    | 8  |
| 2.10 | . . . . .                                  | 9  |
| 2.11 | . . . . .                                  | 9  |
| 2.12 | Авторизуемся на githube . . . . .          | 9  |
| 2.13 | Создаём рабочее пространство . . . . .     | 10 |
| 2.14 | Удаление лишних файлов . . . . .           | 10 |
| 2.15 | . . . . .                                  | 10 |
| 2.16 | отправка файлов на сервер github . . . . . | 10 |

## Список таблиц

# 1 Цель работы

Изучить идеологию и применение средств контроля версий и освоить мения работы с git.

## 2 Выполнение лабораторной работы

Установим Git (рис. 2.1).

```
[root@ribabenko ~]# dnf install git
Последняя проверка окончания срока действия метаданных: 0:03:39 назад
, с6 18 фев 2023 15:09:03.
Пакет git-2.39.2-1.fc37.x86_64 уже установлен.
Зависимости разрешены.
Отсутствуют действия для выполнения.
Выполнено!
```

Рис. 2.1: Установка git

Установим gh (рис. 2.2)

```
[root@ribabenko ~]# dnf install gh
```

Рис. 2.2:

Зададим имя и email владельца репозитория(рис. 2.3).

```
[root@ribabenko ~]# git config --global user.name "Roman Babenko"
[root@ribabenko ~]# git config --global user.email "babenk00r@yandex.ru"
```

Рис. 2.3: Зададим имя и email

Настраиваем utf-8 (рис. 2.4).

```
[root@ribabenko ~]# git config --global core.quotePath false
```

Рис. 2.4:

Настраиваем верификацию и подписание коммитов git и других параметров (рис. 2.5).

```
[root@ribabenko ~]# git config --global init.defaultBranch master
[root@ribabenko ~]# git config --global core.autocrlf input
[root@ribabenko ~]# git config --global core.safecrlf warn
```

Рис. 2.5:

Создаём ключи по алгоритмам (рис. 2.6).

```
Enter file in which to save the key (/root/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_ed25519
Your public key has been saved in /root/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:7XyDcnF56mjKyoXxkvxqHkf0wM4n2Lg+bZwpKdeKP7U root@ribabenko.rababenko.net
The key's randomart image is:
+--[ED25519 256]--+
|
|      .
|      +
|      B + .
|      + S = o .
|      . B. = + o
|      B* = + +
|      .+BBEo.+ .
|      .*OO+o. .
+-----[SHA256]-----+
[root@ribabenko ~]#
```

Рис. 2.6: Создание ключей

Создаём ключи pgr(рис. 2.7).

```

Какой размер ключа Вам необходим? (3072) 4096
Запрошенный размер ключа - 4096 бит
Выберите срок действия ключа.
    0 = не ограничен
    <n> = срок действия ключа - n дней
    <n>w = срок действия ключа - n недель
    <n>m = срок действия ключа - n месяцев
    <n>y = срок действия ключа - n лет
Срок действия ключа? (0) 0
Срок действия ключа не ограничен
Все верно? (y/N) y

GnuPG должен составить идентификатор пользователя для идентификации ключа.

Ваше полное имя: Roman
Адрес электронной почты: babenk00r@yandex.ru
Примечание:
Вы выбрали следующий идентификатор пользователя:
    "Roman <babenk00r@yandex.ru>"
Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход?

```

Рис. 2.7: Генерируем ключ

Выводим список ключей и копируем отпечаток приватного ключа(рис. 2.8).

```

-----
sec   rsa4096/E5AC83D9488AE4C6 2023-02-18 [SC]
      D63B9C202682367EFFF2B333E5AC83D9488AE4C6
uid   [ абсолютно ] Roman <babenk00r@yandex.ru>
ssb   rsa4096/5097AE70C1828275 2023-02-18 [E]

```

Рис. 2.8: Выводим список ключей

Скопируем сгенерированный ргр ключ в буфер обмена(рис. 2.9).

```

[root@ribabenko ~]# gpg --armor --export E5AC83D9488AE4C6 | xclip -sel clip

```

Рис. 2.9: Копируем ключ

Добавляем ключ на github (рис. 2.10).



GPG keys / Add new

Title

key\_1

Key

key\_1

-----BEGIN PGP PUBLIC KEY BLOCK-----

-----END PGP PUBLIC KEY BLOCK-----

Add GPG key

Рис. 2.10:

Используя приведённый email указываем git применять его при подписи коммитов (рис. 2.11).

```
[root@ribabenko ~]# git config --global user.signingkey E5AC83D9488AE4C6
[root@ribabenko ~]# git config --global commit.gpgsign true
[root@ribabenko ~]# git config --global gpg.program $(which gpg2)
```

Рис. 2.11:

Авторизируемся на github через командную строку Linux (рис. 2.12).

```
[root@ribabenko ~]# gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations? SSH
? Upload your SSH public key to your GitHub account? Skip
? How would you like to authenticate GitHub CLI? Paste an authentication token
Tip: you can generate a Personal Access Token here https://github.com/settings/tokens
The minimum required scopes are 'repo', 'read:org'.
? Paste your authentication token: *****
- gh config set -h github.com git_protocol ssh
✓ Configured git protocol
✓ Logged in as skrabik
```

Рис. 2.12: Авторизируемся на githubе

Создадим репозиторий курса на основе шаблона (рис. 2.13).

```

и репозитории существует.
[root@ribabenko ~]# git clone https://github.com/skrabik/study_2022-2023_os-intro.git os-intro
Клонирование в «os-intro»...
remote: Enumerating objects: 27, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (26/26), done.
remote: Total 27 (delta 1), reused 11 (delta 0), pack-reused 0
Получение объектов: 100% (27/27), 16.93 КиБ | 8.46 МБ/с, готово.
Определение изменений: 100% (1/1), готово.
[root@ribabenko ~]#

```

Рис. 2.13: Создаём рабочее пространство

Удаляем лишние файлы (рис. 2.14).

```

[root@ribabenko os-intro]# rm package.json
rm: удалить обычный файл 'package.json': у

```

Рис. 2.14: Удаление лишних файлов

Создаём необходимые каталоги (рис. 2.15).

```

[root@ribabenko os-intro]# echo os-intro > COURSE
[root@ribabenko os-intro]# make

```

Рис. 2.15:

Отправляем файлы на сервер (рис. 2.16).

```

[root@ribabenko os-intro]# git push
Перечисление объектов: 40, готово.
Подсчет объектов: 100% (40/40), готово.
Сжатие объектов: 100% (30/30), готово.
Запись объектов: 100% (38/38), 343.02 КиБ | 11.83 МБ/с, готово.
Всего 38 (изменений 4), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To https://github.com/skrabik/study_2022-2023_os-intro.git
   e947e41..aeefb2  master -> master
[root@ribabenko os-intro]#

```

Рис. 2.16: отправка файлов на сервер github

## **3 Выводы**

В ходе выполнения данной лабораторной работы я изучил идеологию и применение средств контроля версий и освоил умения работы с git.