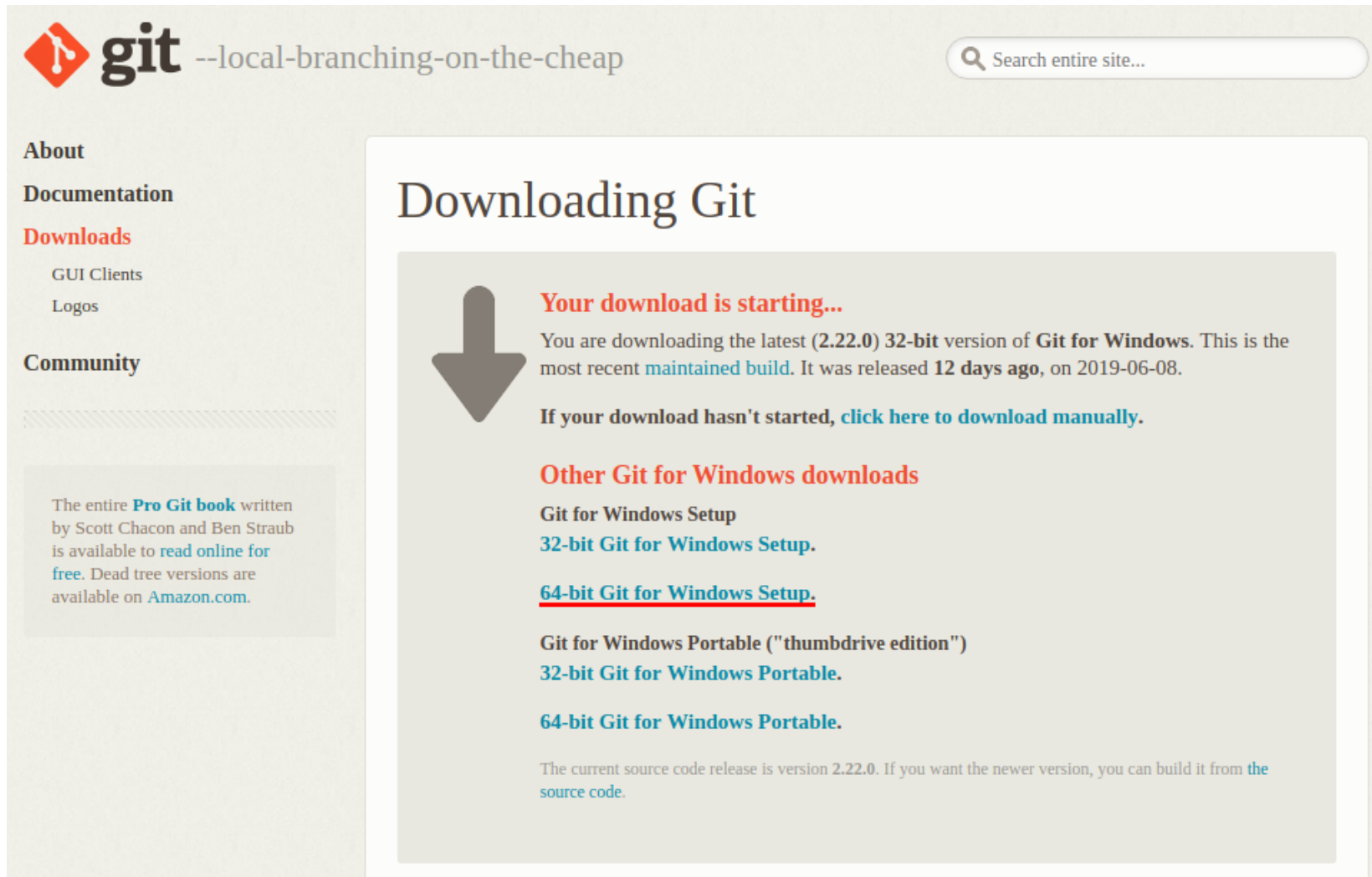


Открываем официальный сайт [проекта](#), кликаем по 64-битной версии для Windows.

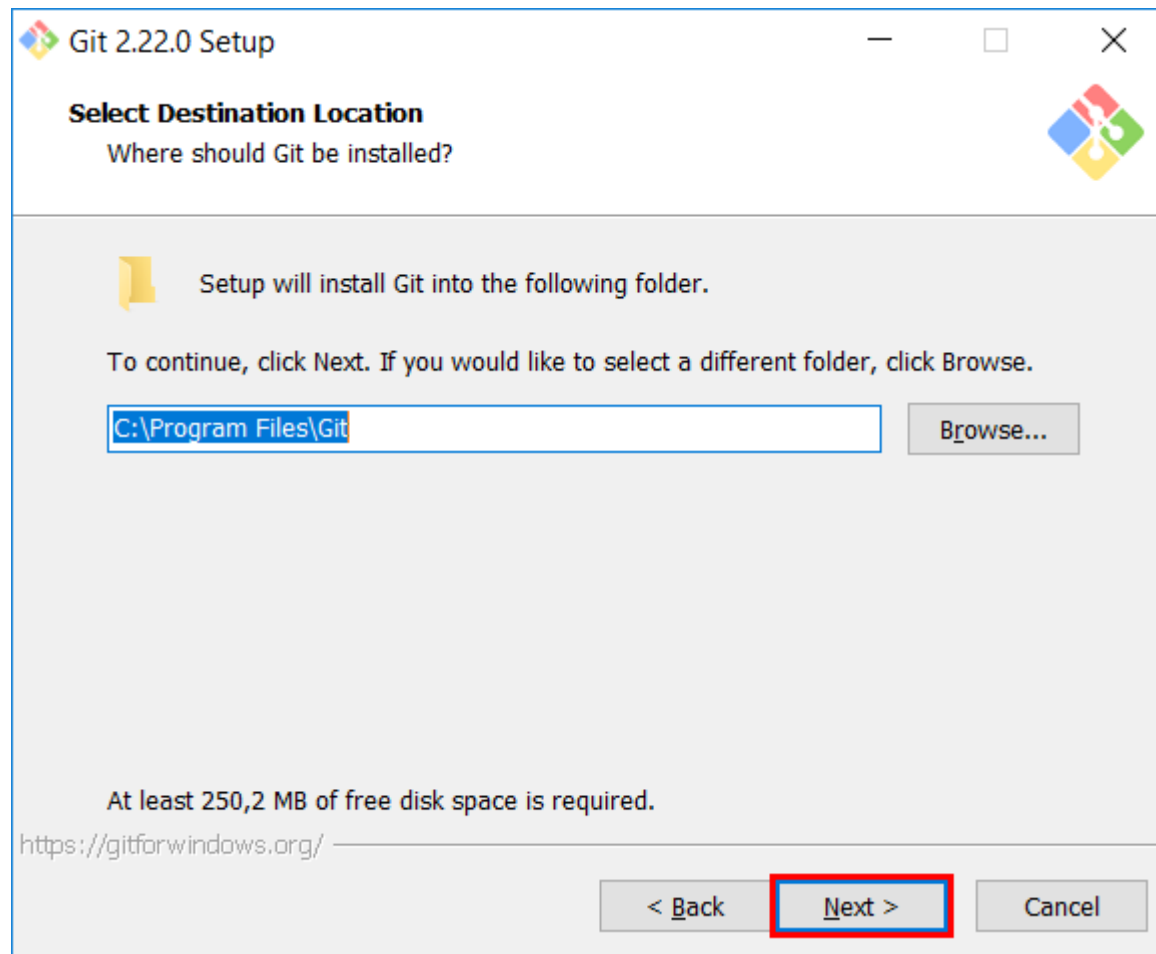


Загрузка начнется автоматически, ждем ее окончания. Запускаем полученный файл.

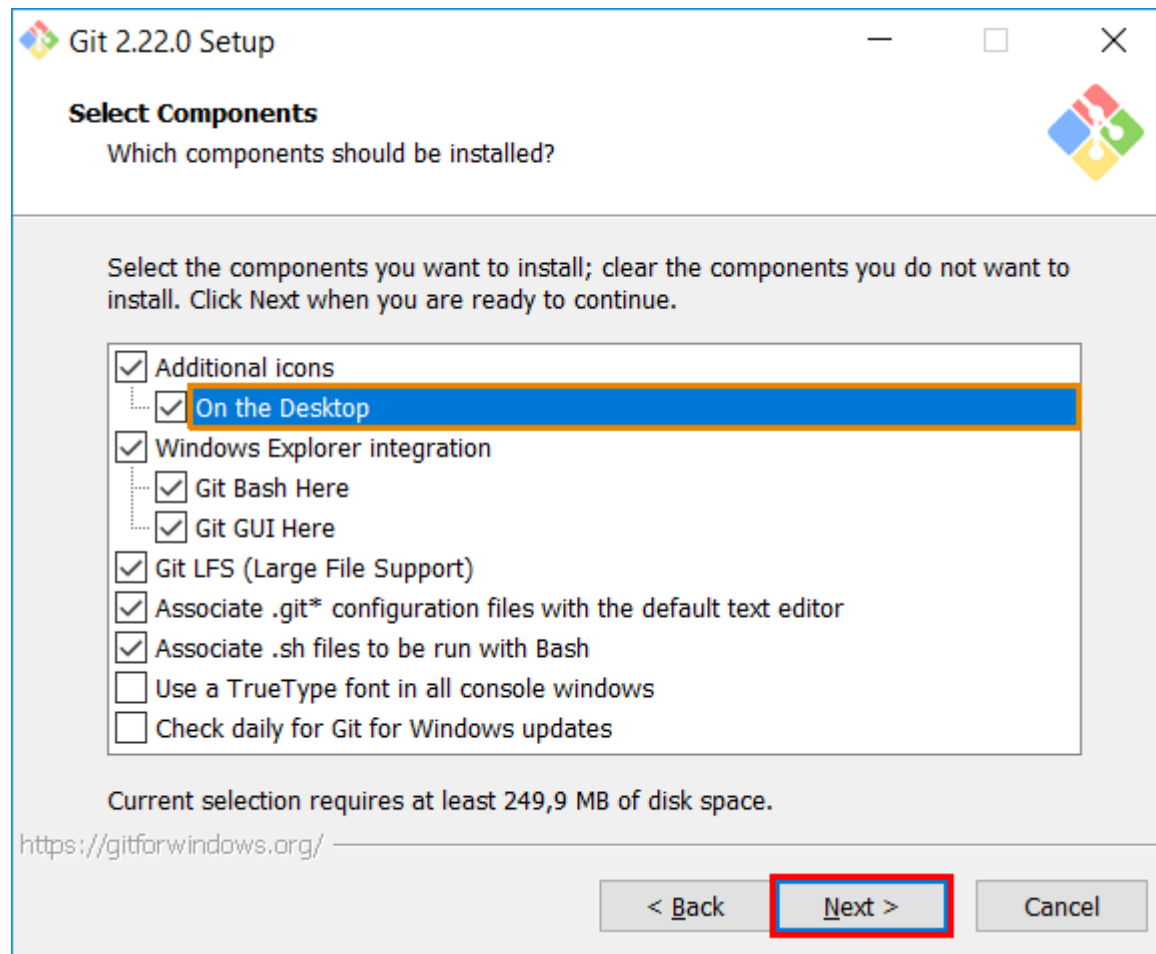
Программа установки приветствует нас лицензионным соглашением. Кликаем по кнопке “Next”.



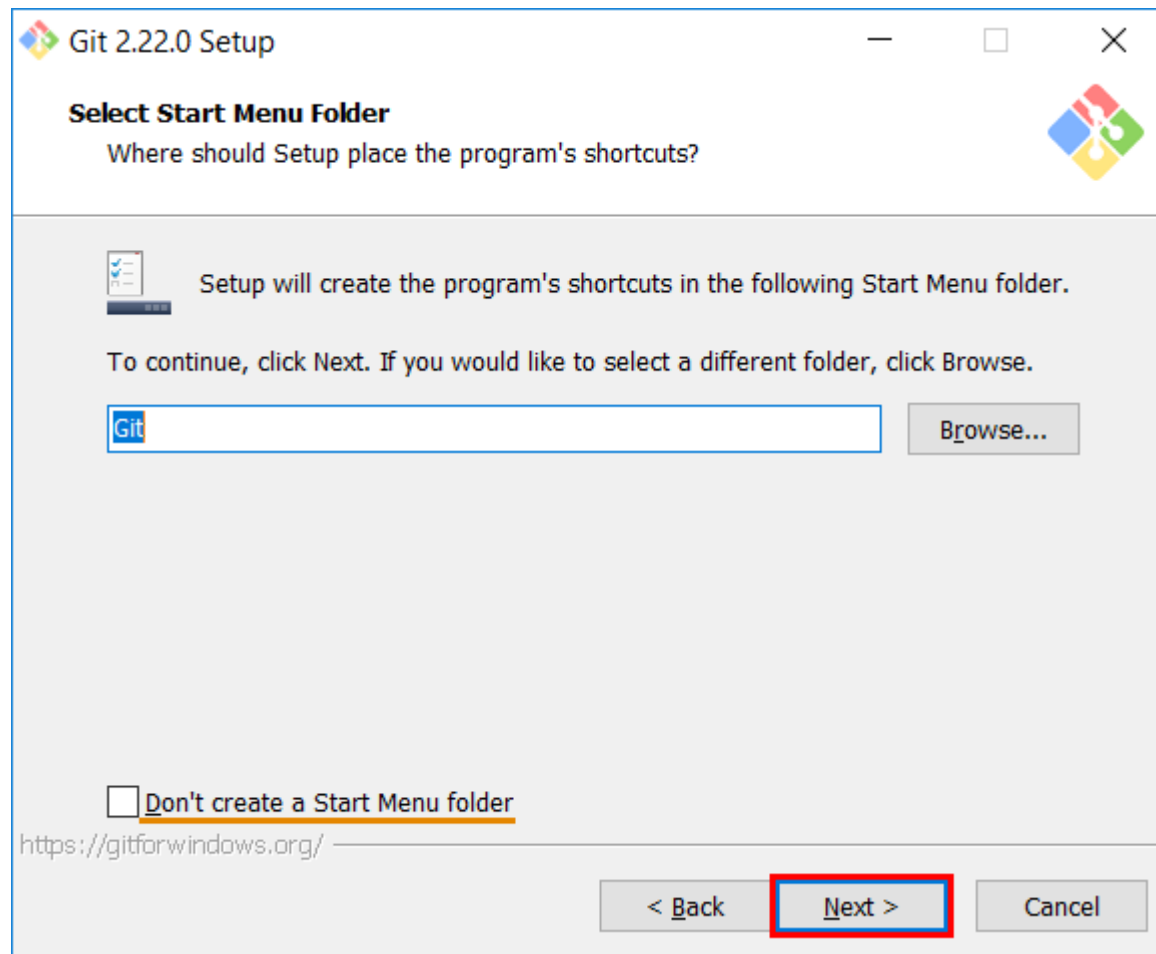
На следующем шаге указываем путь установки. Нас устраивает указанный по умолчанию. Кликаем “Next”.



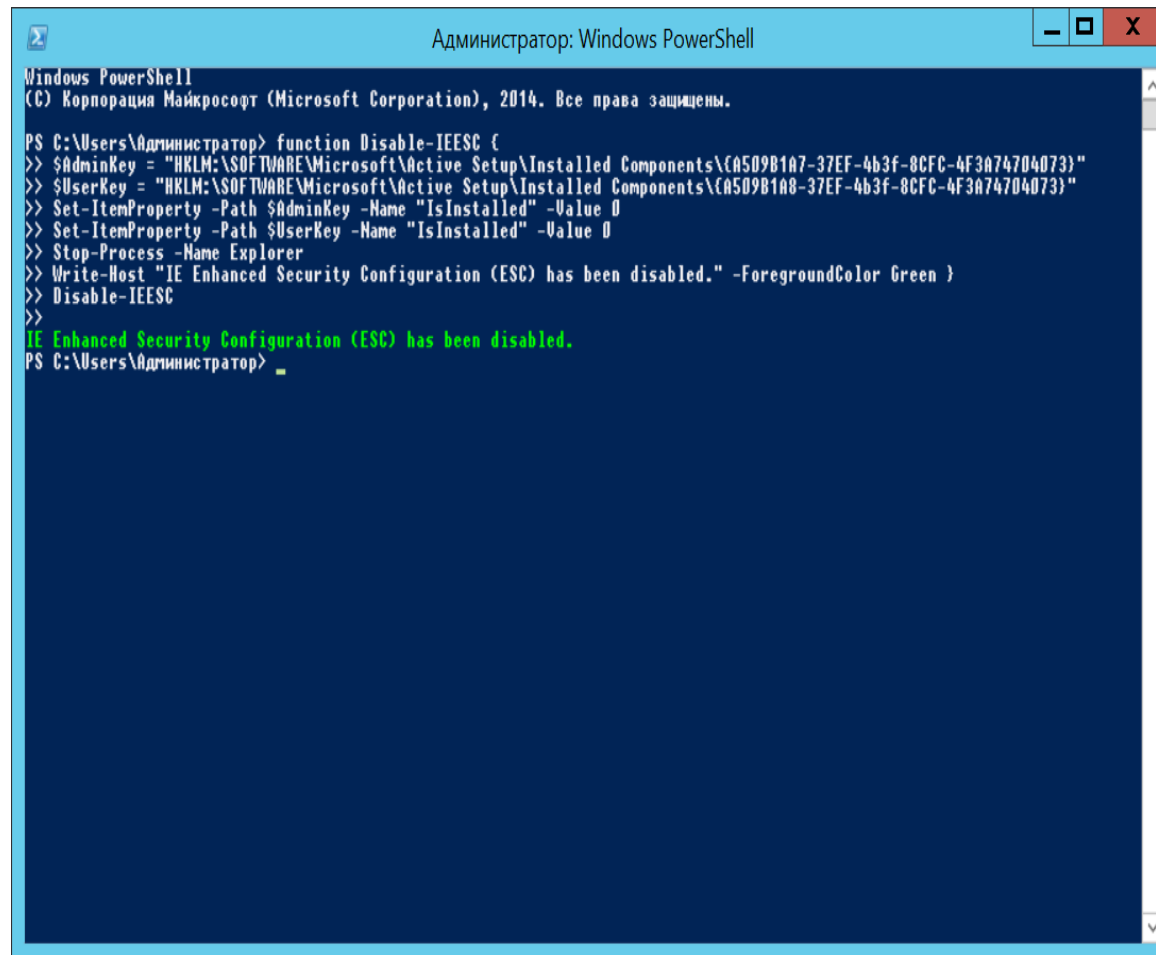
На этапе выбора компонентов, отметим галочкой создание ярлыка Git на рабочем столе и переходим к следующему этапу.



Теперь программа установки предлагает создать ярлык в меню Пуск. Если в этом не видите необходимости, установите соответствующую галку. Кликаем по кнопке “Next”.



На следующем этапе, установщик предложит выбрать редактор для git по умолчанию. Для новичков лучше использовать nano, для тех, кто любит посложнее — Vim. Хотя допускается использование Notepad++ и прочего. Переходим к следующему шагу.



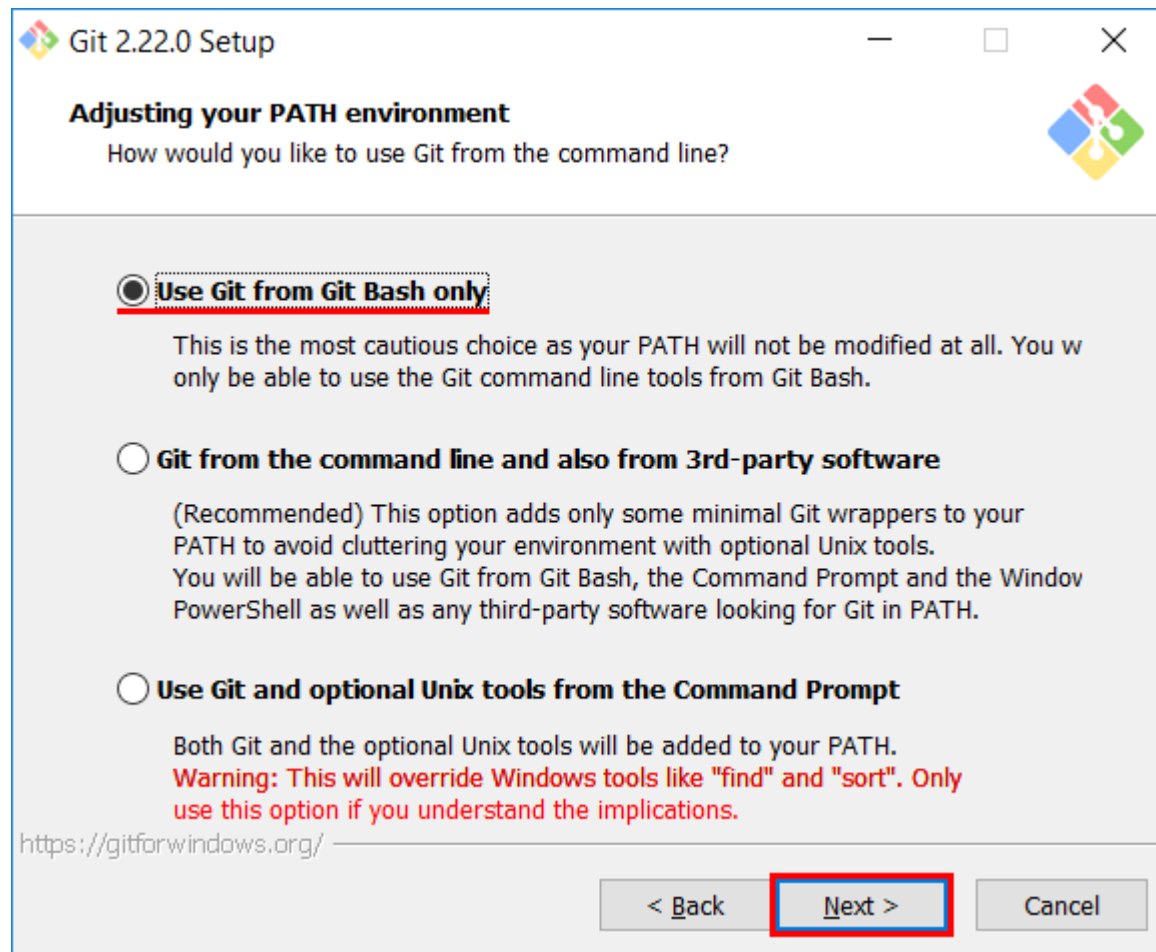
```
Администратор: Windows PowerShell
Windows PowerShell
(С) Корпорация Майкрософт (Microsoft Corporation), 2014. Все права защищены.

PS C:\Users\Администратор> function Disable-IEESC {
>> $AdminKey = "HKLM:\SOFTWARE\Microsoft\Active Setup\Installed Components\{A509B1A7-37EF-4b3f-8CFC-4F3A74704073}"
>> $UserKey = "HKLM:\SOFTWARE\Microsoft\Active Setup\Installed Components\{A509B1A8-37EF-4b3f-8CFC-4F3A74704073}"
>> Set-ItemProperty -Path $AdminKey -Name "IsInstalled" -Value 0
>> Set-ItemProperty -Path $UserKey -Name "IsInstalled" -Value 0
>> Stop-Process -Name Explorer
>> Write-Host "IE Enhanced Security Configuration (ESC) has been disabled." -ForegroundColor Green }
>> Disable-IEESC
>>
IE Enhanced Security Configuration (ESC) has been disabled.
PS C:\Users\Администратор> _
```

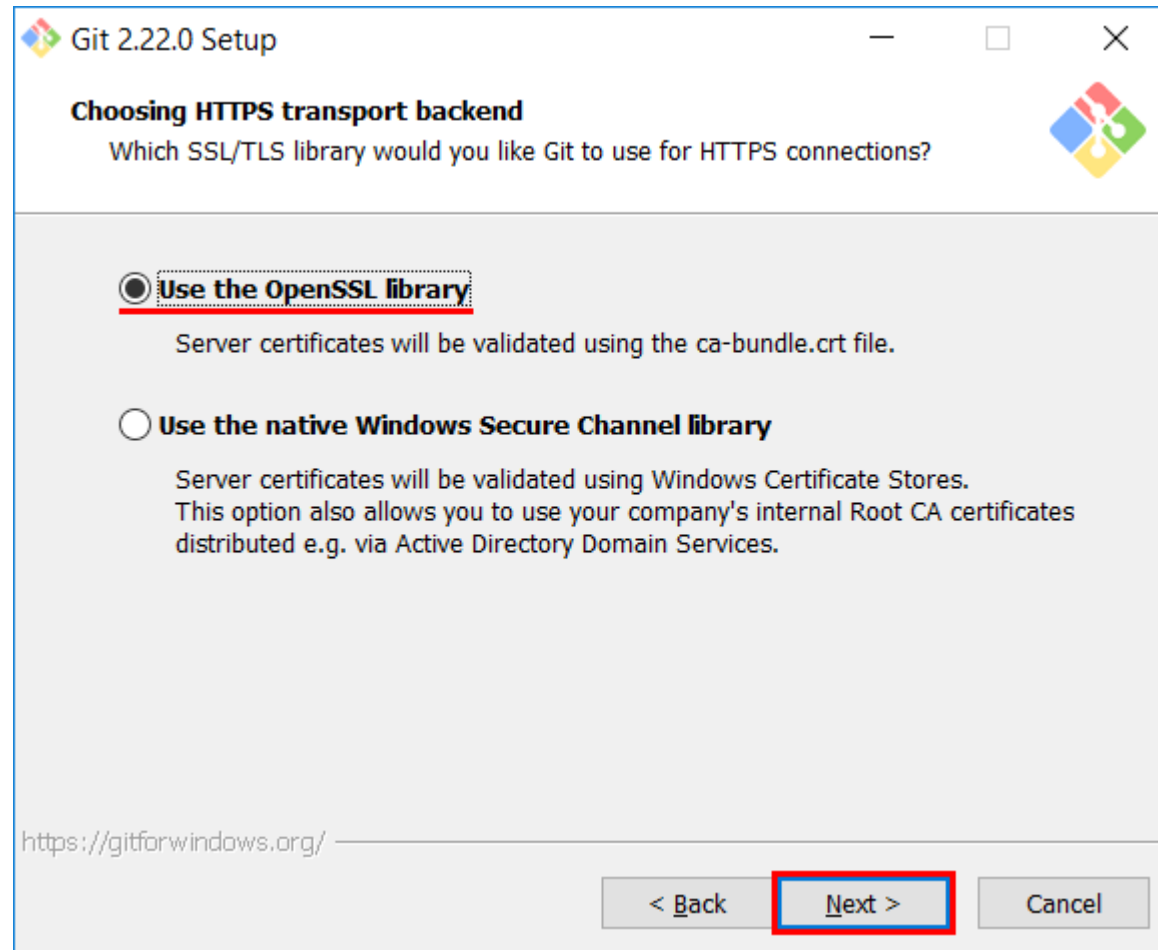
Теперь программа установщик спрашивает как Вы собираетесь использовать командную строку:

- Use Git from Git Bash only — использование только из командной строки Bash.

- Use Git from the Windows Command Prompt — использование командной строки Bash, а также минимальный набор команд Git из командной строки Windows.
- Use Git and optional Unix tools from the Windows Command Prompt — использование Git и утилит Unix из командной строки Windows. При выборе этой опции будут перезаписаны некоторые утилиты ОС Windows, например find и sort. Оставляем первый вариант и двигаемся дальше.

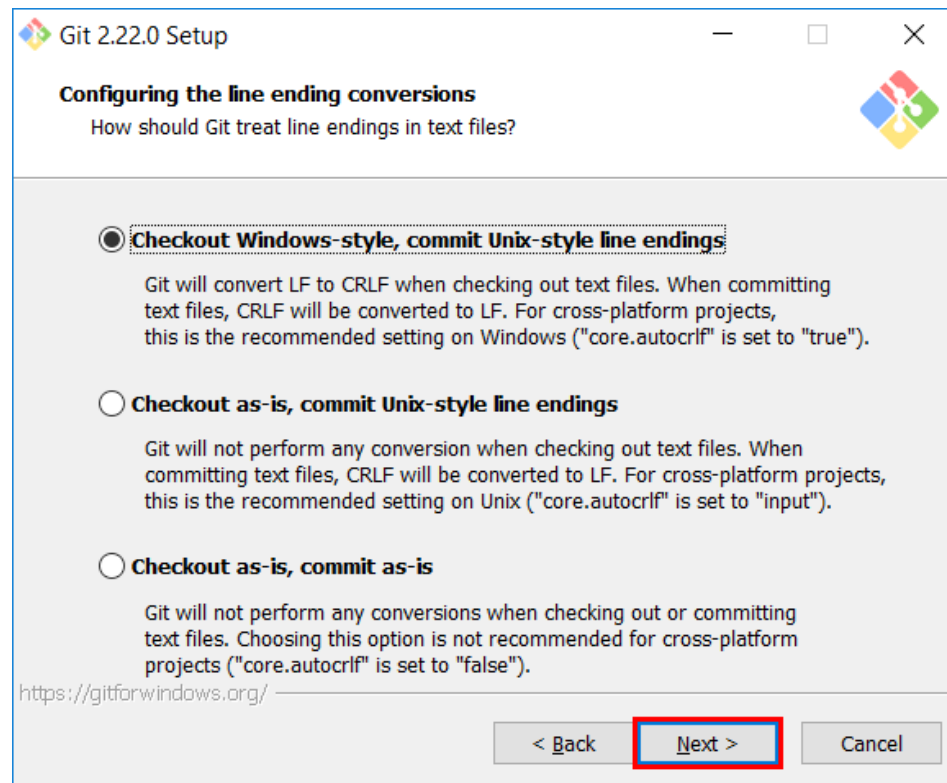


Теперь установщик спрашивает о том, какую библиотеку следует использовать для HTTPS-соединений. Оставляем openssl и переходим к следующему шагу.



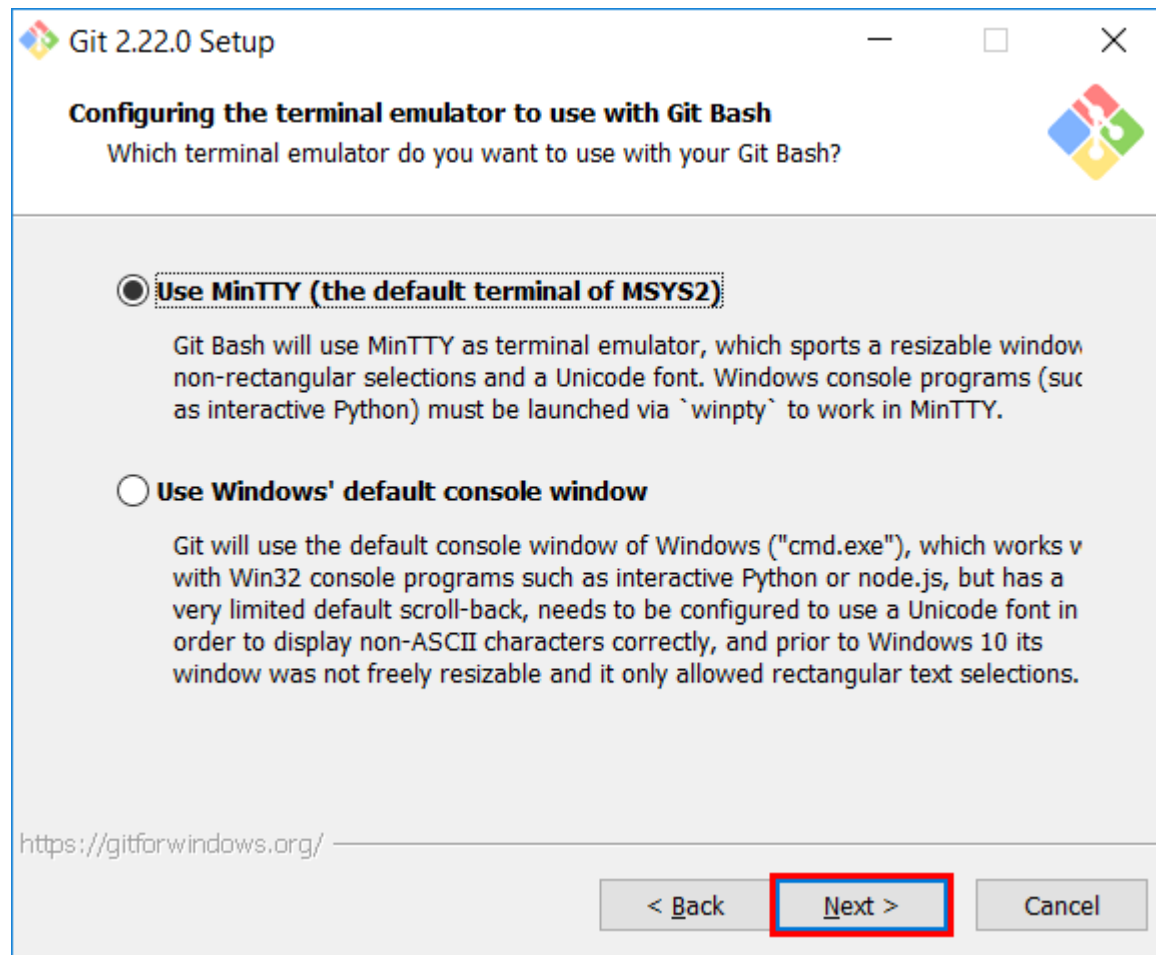
Следующий шаг можно назвать самым значимым. Вопрос программы установки в том, какой формат завершения строки использовать. Одной из отличительных особенностей Windows от Unix-подобных систем — символ перехода на новую строку в файлах, а именно, LF и CRLF — управляющие символ для переноса строки в Unix и Windows соответственно.

Мы рекомендуем выбрать параметр предложенный системой — отладка файла в формате Windows, а загрузка файла — в Unix-формате. Кликаем по кнопке “Далее”.



На данном этапе, программа запросит каким терминалом пользоваться далее:

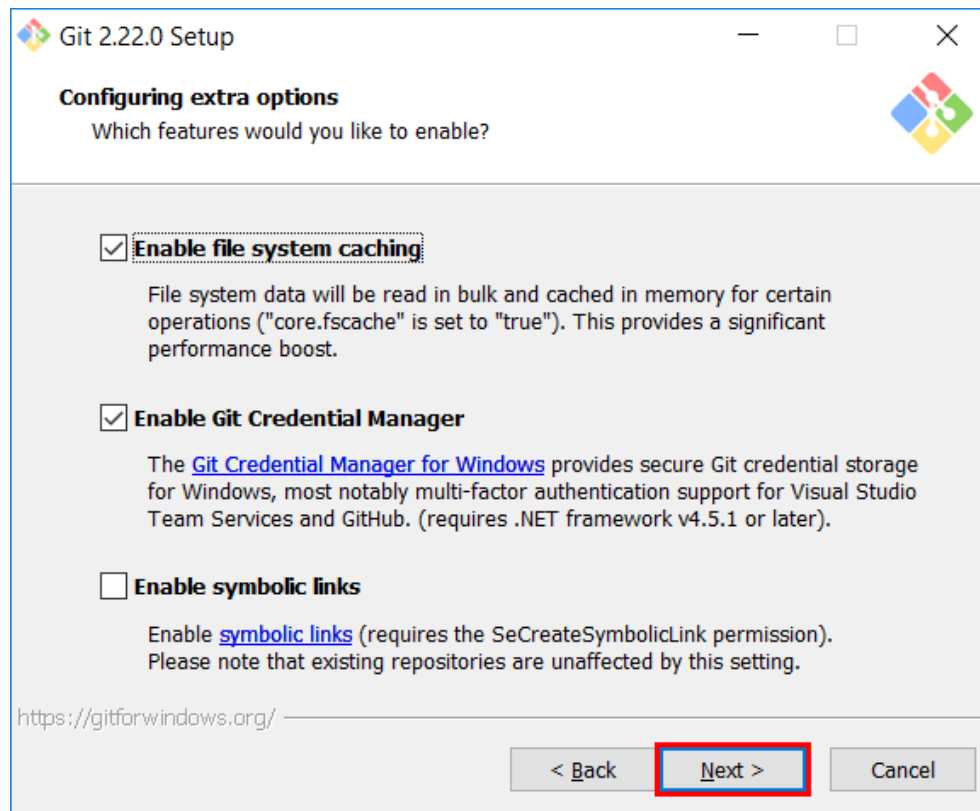
- MinTTY — терминал Unix;
 - Windows — стандартный терминал (командная строка) Windows.
- В нашем случае оставляем “как есть” и двигаемся дальше.



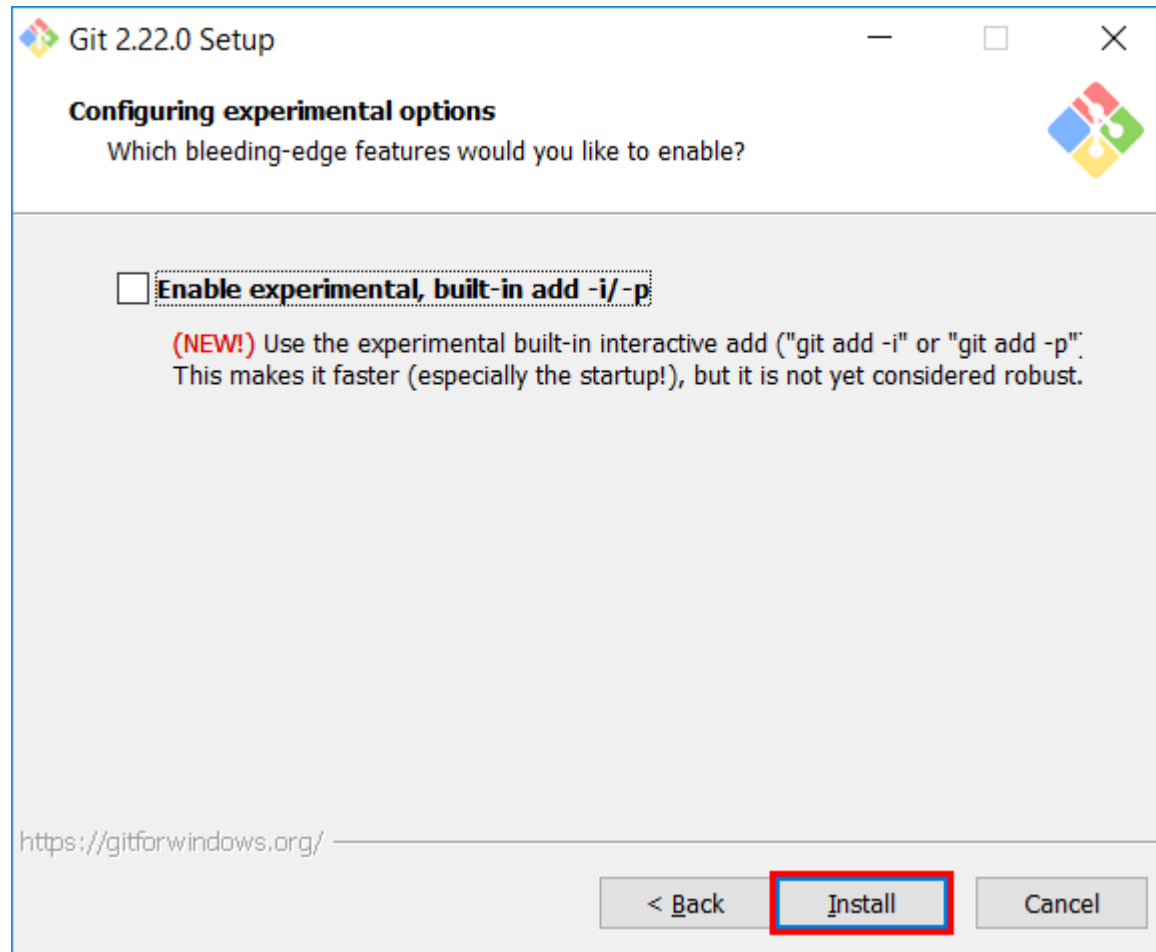
На следующем шаге, установщик попросит отметить необходимые функции:

- File system caching — кэширование файловой системой.
- Git Credential Manager — включить менеджер учетных данных.
- Symbolic links — разрешить использование символьных ссылок.

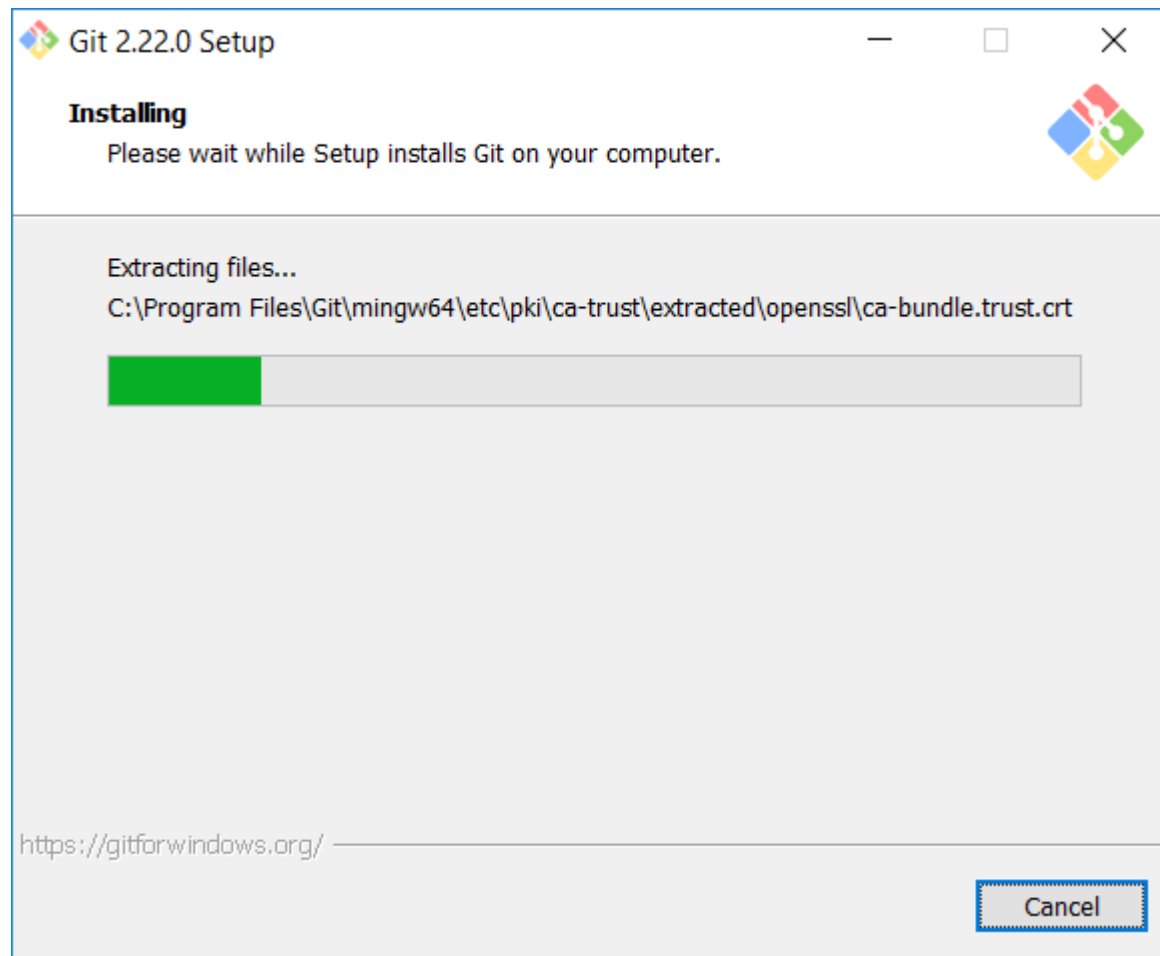
Выбираем то, что необходимо и переходим к следующему шагу.



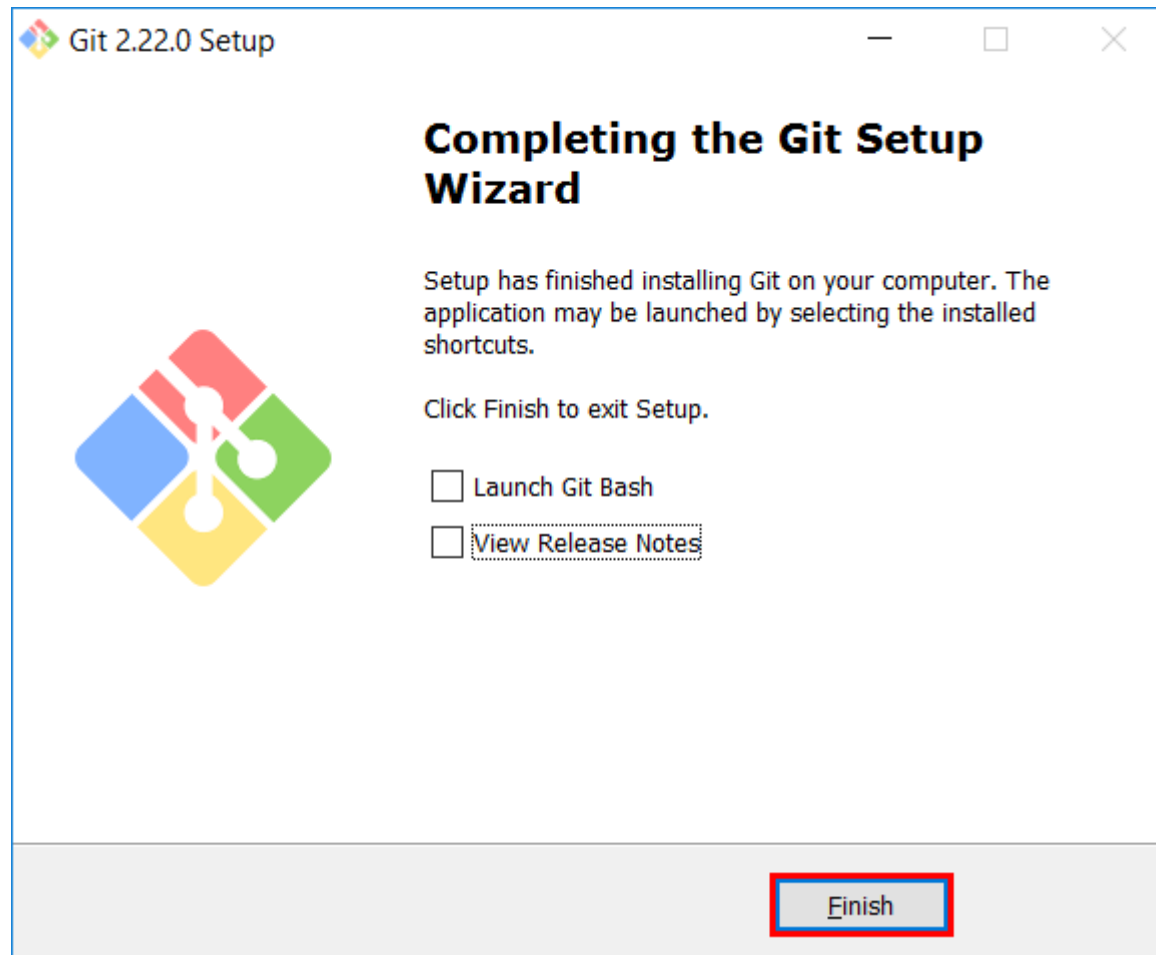
На данном шаге, программа установки предлагает использовать экспериментальную опцию. Использовать ее или нет — решать Вам. Кликаем по кнопке “Install”.



Программа приступает к копированию файлов. Дожидаемся окончания процесса.



По окончании процесса установки нажимаем "Finish".



Подключаемся к репозиторию.

Предполагается, что у вас уже есть своя учетная запись на github/gitlab. Если нет, то, в случае с github, проходим несложный процесс регистрации на сайте. Затем авторизуемся на нем и создаем репозиторий. Для этого кликаем по кнопке "Start a project".



Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

[Read the guide](#)

[Start a project](#)

Указываем имя репозитория и кликаем по кнопке "Create repository".

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner





Repository name *

/ firstProject ✓

Great repository names are short and memorable. Need inspiration? How about **musical-tribble**?

Description (optional)

- ☒  **Public**
Anyone can see this repository. You choose who can commit.
- ☐  **Private**
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

- ☐ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer.

Add .gitignore: None ▾

Add a license: None ▾

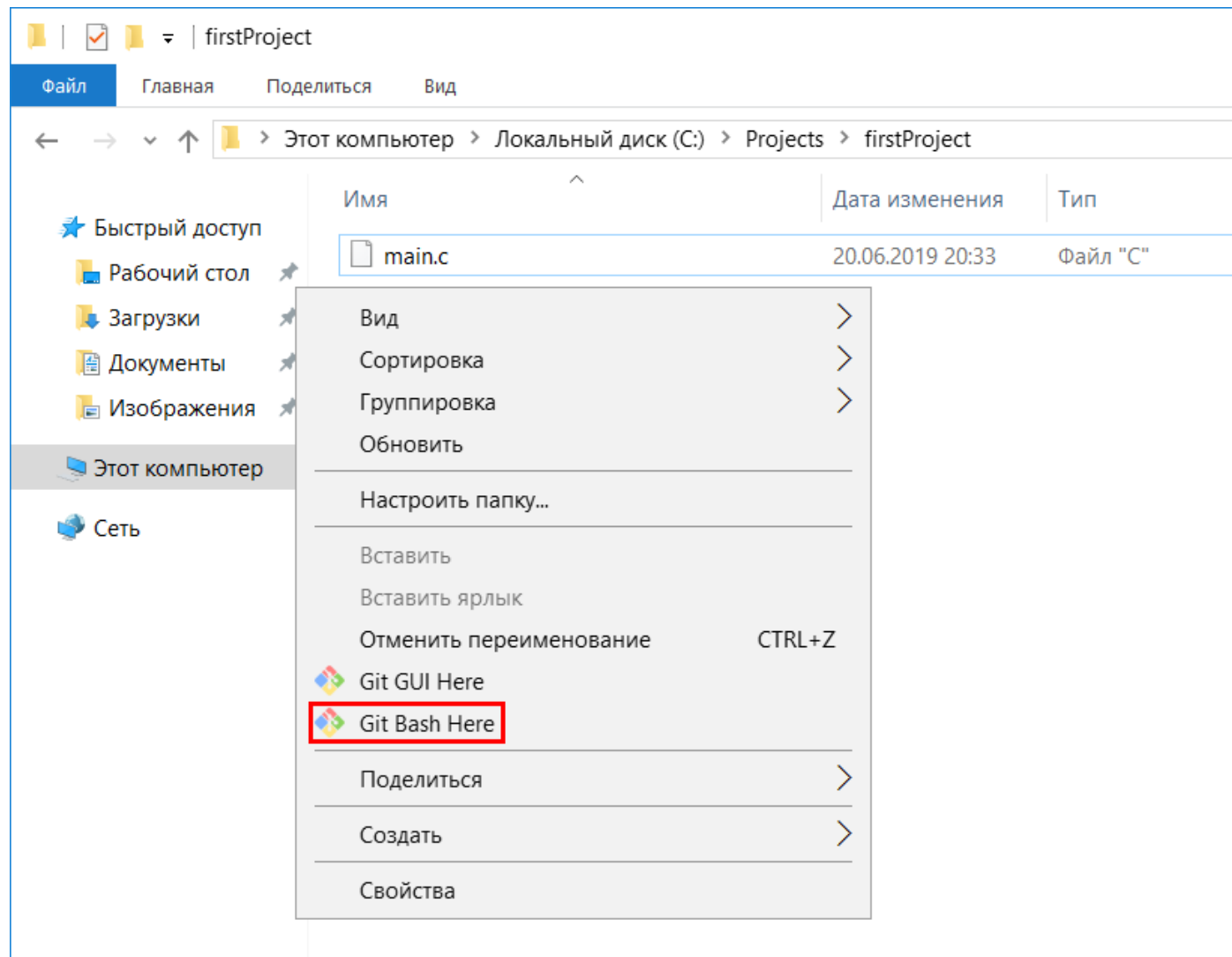


Create repository

Репозиторий создан.

Добавляем файлы.

Предположим, что у нас есть файл, который необходимо добавить в репозиторий git или gitlab. Открываем папку, где располагается файл, кликаем правой кнопкой мыши по свободному месту и из открывшегося контекстного меню выбираем “Git Bash Here”.

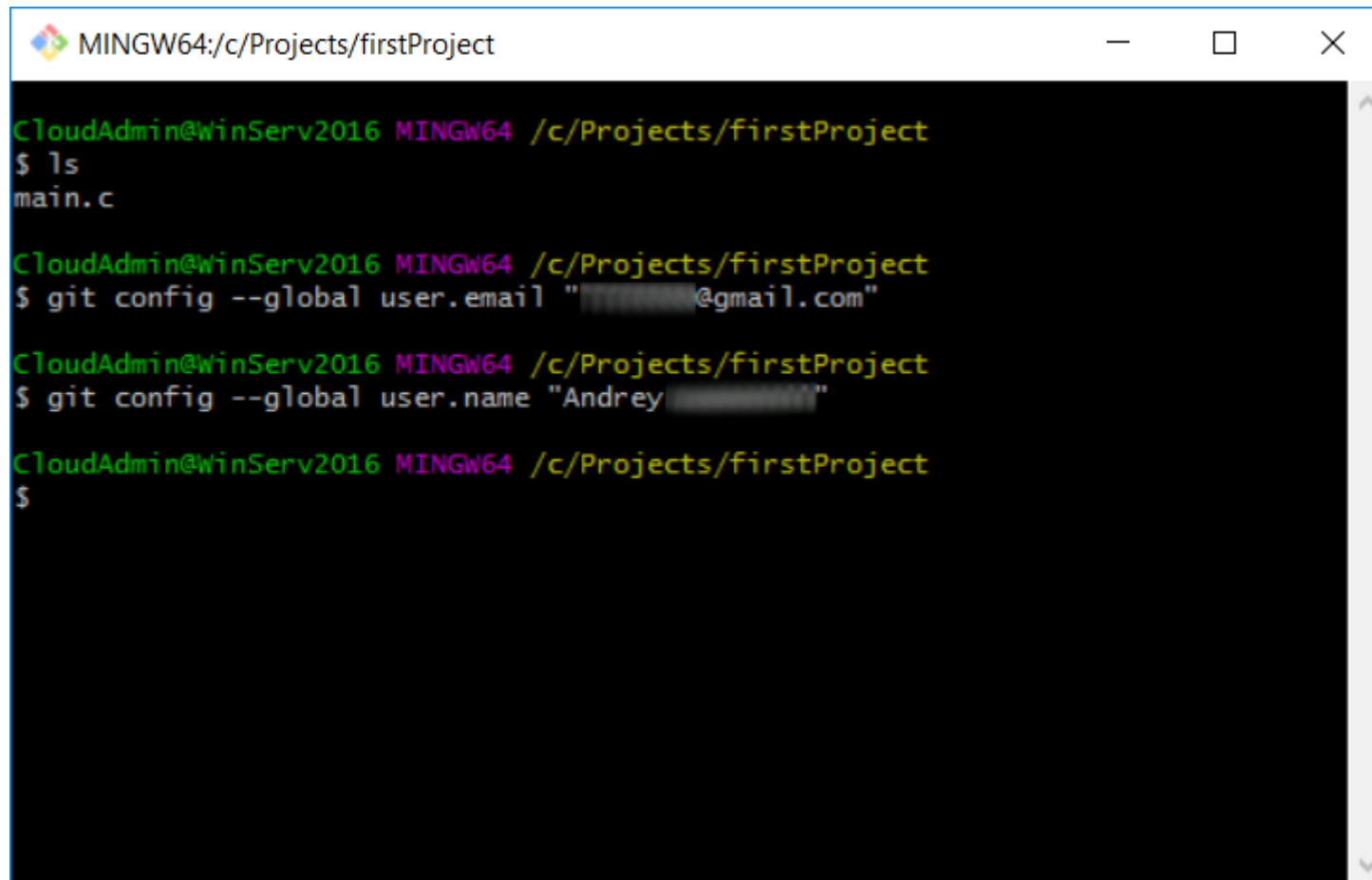


Откроется командная строка. Последовательно выполняем следующие команды:

```
git config --global user.email "youmail@example.com"
```

```
git config --global user.name "Ваше имя"
```

Таким образом мы указали нашу почту и имя.

A screenshot of a Windows command prompt window titled "MINGW64:/c/Projects/firstProject". The window has standard Windows window controls (minimize, maximize, close) in the top right corner. The command prompt shows a series of commands and their outputs. The prompt is "CloudAdmin@WinServ2016 MINGW64 /c/Projects/firstProject". The first command is "\$ ls", which outputs "main.c". The second command is "\$ git config --global user.email \"[redacted]@gmail.com\"". The third command is "\$ git config --global user.name \"Andrey [redacted]\"". The fourth command is "\$", which shows the prompt without any output. The text is color-coded: the prompt is green, the command is yellow, and the output is white. There are redaction marks (black boxes) over the email address and the name in the configuration commands.

```
MINGW64:/c/Projects/firstProject

CloudAdmin@WinServ2016 MINGW64 /c/Projects/firstProject
$ ls
main.c

CloudAdmin@WinServ2016 MINGW64 /c/Projects/firstProject
$ git config --global user.email "[redacted]@gmail.com"

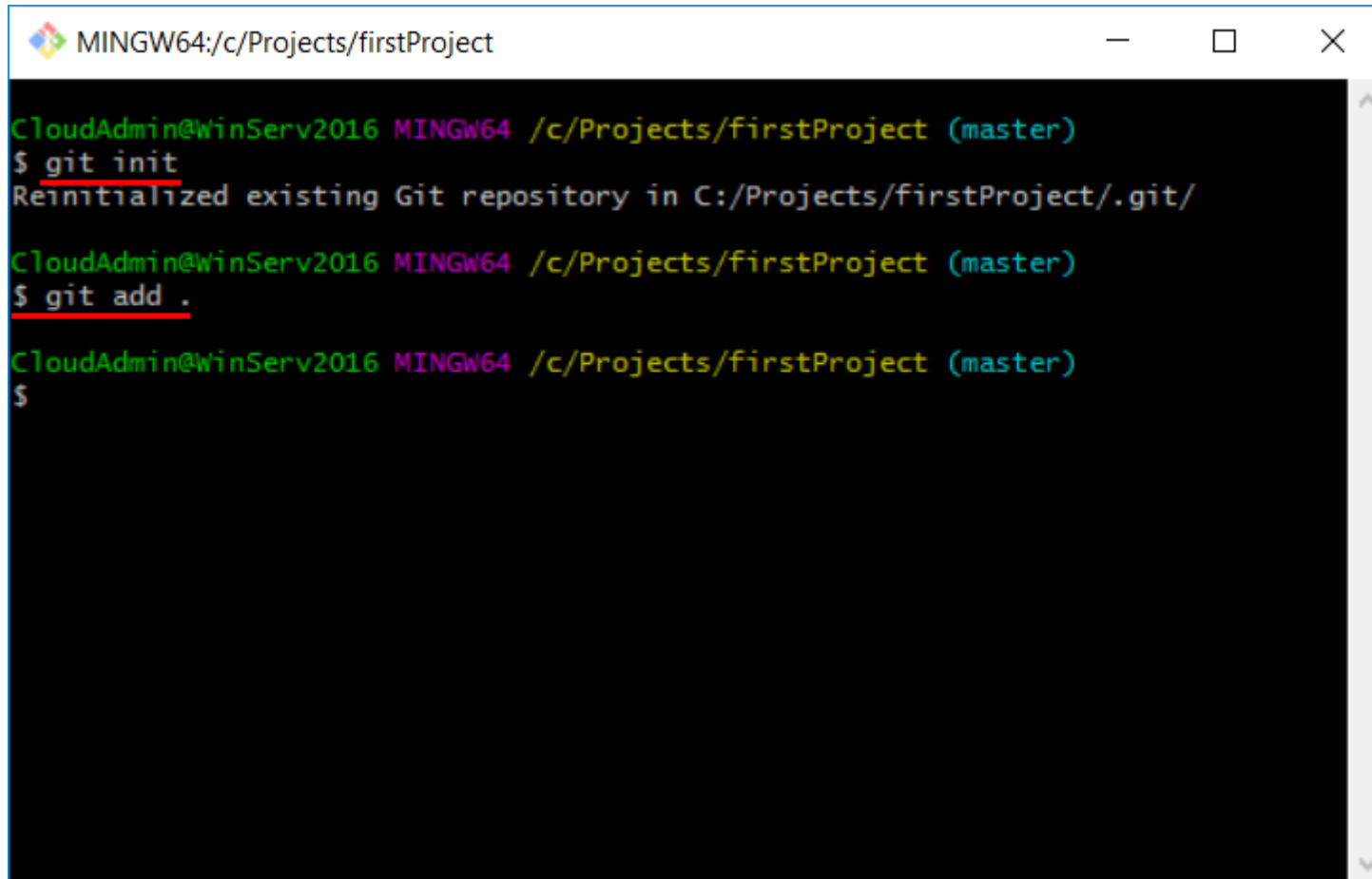
CloudAdmin@WinServ2016 MINGW64 /c/Projects/firstProject
$ git config --global user.name "Andrey [redacted]"

CloudAdmin@WinServ2016 MINGW64 /c/Projects/firstProject
$
```

Для того, чтобы начать отслеживать изменения в текущей папке, выполняем следующие команды:

```
git init
```

```
git add .
```

A screenshot of a Windows command prompt window titled "MINGW64:/c/Projects/firstProject". The window has standard Windows window controls (minimize, maximize, close) in the top right corner. The command prompt shows the following sequence of commands and output:
1. Prompt: CloudAdmin@WinServ2016 MINGW64 /c/Projects/firstProject (master)
2. Command: \$ git init
3. Output: Reinitialized existing Git repository in C:/Projects/firstProject/.git/
4. Prompt: CloudAdmin@WinServ2016 MINGW64 /c/Projects/firstProject (master)
5. Command: \$ git add .
6. Prompt: CloudAdmin@WinServ2016 MINGW64 /c/Projects/firstProject (master)
7. Prompt: \$
The commands "git init" and "git add ." are underlined in red in the original image. The prompt text is color-coded: "CloudAdmin" is green, "WinServ2016" is green, "MINGW64" is pink, and "/c/Projects/firstProject (master)" is yellow.

Выполняем свой первый коммит и добавляем изменения на сервер:

```
git commit -m "Init"
```

Важно. Чтобы внести изменения на сервер, следует использовать для github:

```
git remote add origin https://github.com/пользователь/название_репозитория.git
```

```
git push -u origin master
```

Для gitlab:

```
git remote add gitlab https://server/namespace/project.git
```

```
git push -u gitlab master
```

```
MINGW64:/c/Projects/firstProject

CloudAdmin@WinServ2016 MINGW64 /c/Projects/firstProject (master)
$ git init
Reinitialized existing Git repository in C:/Projects/firstProject/.git/


CloudAdmin@WinServ2016 MINGW64 /c/Projects/firstProject (master)
$ git add .

CloudAdmin@WinServ2016 MINGW64 /c/Projects/firstProject (master)
$ git commit -m "Init"
[master (root-commit) 23a302f] Init
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 main.c


CloudAdmin@WinServ2016 MINGW64 /c/Projects/firstProject (master)
$ git remote add origin https://github.com/[REDACTED]/firstProject.git


CloudAdmin@WinServ2016 MINGW64 /c/Projects/firstProject (master)
$ git push -u origin master
|
```

После запуска команды, откроется окно авторизации. Вводим данные, которые использовали при регистрации.

 GitHub Login ×

GitHub Login

 Login

 Cancel

Don't have an account? [Sign up](#)

[Forgot your password?](#)

Настройка аутентификации по SSH-ключам.

Для начала создадим ключ:

```
ssh-keygen -t rsa -b 4096 -C "<a href='mailto:yourmail@gmail.com'>yourmail@gmail.com</a>"
```

В ходе генерации ключа, программа запросит ввод парольной фразы, которая необходима для защиты самого ключа. Можно ответить на вопрос двойным нажатием на Enter.

Закрытый ключ будет находиться по пути C:Users<Пользователь>.sshid_rsa. Открытый ключ — C:Users<Пользователь>.sshid_rsa.pub


```
MINGW64:/c/Projects

CloudAdmin@WinServ2016 MINGW64 /c/Projects
$ ssh-keygen -t rsa -b 4096 -C "[REDACTED]@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Администратор.WIN-0C3CPATQDHL/.ssh/id_rsa):
/c/Users/Администратор.WIN-0C3CPATQDHL/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Администратор.WIN-0C3CPATQDHL/.ssh/id_rsa.
Your public key has been saved in /c/Users/Администратор.WIN-0C3CPATQDHL/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:9gQu/7uBVf/oNeE3M2ArpksqnWeYC3fnTwZZow[REDACTED]@gmail.com
The key's randomart image is:
+---[RSA 4096]---+
|
|      .  .
|     . . . .
|    . S o. =.+
|   + = = Boo|
|  ...*oo+ X.*+
|  [REDACTED]
|   .o+o=E.o.
|
+---[SHA256]-----+

CloudAdmin@WinServ2016 MINGW64 /c/Projects
$
```

Запускаем SSH-агента:

```
eval $(ssh-agent -s)
```

Добавляем ключи к ssh-агенту:

```
ssh-add ~/.ssh/id_rsa
```

MINGW64:/c/Projects

CloudAdmin@WinServ2016 MINGW64 /c/Projects

\$ eval \$(ssh-agent -s)

Agent pid 1561

CloudAdmin@WinServ2016 MINGW64 /c/Projects

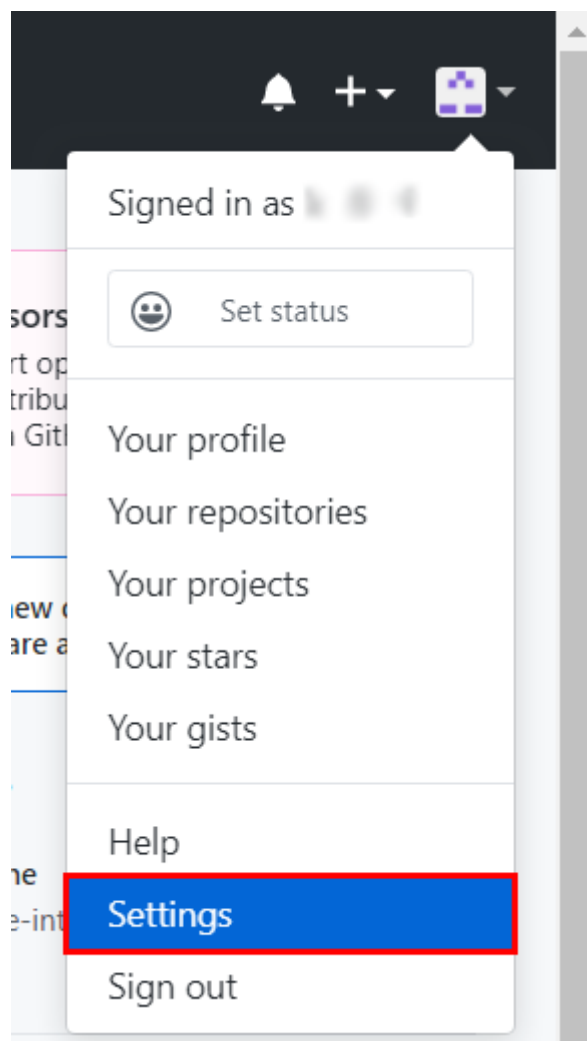
\$ ssh-add ~/.ssh/id_rsa

Identity added: /c/Users/Администратор.WIN-0C3CPATQDHL/.ssh/id_rsa (██████████@gmail.com)

CloudAdmin@WinServ2016 MINGW64 /c/Projects

\$

Добавляем созданный ключ на сервер Git. На сайте github, в верхнем правом углу, кликаем по иконке учетной записи, выбираем "Setting".



На обновленной странице, в левом меню, выбираем “SSH and PGP keys”, а в основной части окна, напротив SSH keys, кликаем по кнопке “New SSH key”.

- Personal settings
- [Profile](#)
- [Account](#)
- [Security](#)
- [Emails](#)
- [Notifications](#)
- [Billing](#)
- SSH and GPG keys**
- [Blocked users](#)
- [Repositories](#)
- [Organizations](#)
- [Saved replies](#)
- [Applications](#)
- [Developer settings](#)

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](#).

GPG keys

New GPG key

There are no GPG keys associated with your account.

Learn how to [generate a GPG key and add it to your account](#).

Заполняем поле Title. В поле Key вставляем содержимое файла C:\Users<Пользователь>.sshid_rsa.pub

Кликаем по кнопке “Add SSH key”.

SSH keys / Add new

Title

ForExample

Key

```
AAAAB3NzaC1yc2EAAAADAQABAAQCAQCw2JNl1Axlkon9T6L0R1Ly/FLH6eFY0sp0l1IOFhyRE2U0Ju5liDvehlw9gcE  
rNU68aPczFv+7OCCLMw9DI2Uh+uR5x+s1QKIZl+ tqvibALItUHMnqSwewwZACbDfz6oud1qs8YQyUY0c5h0pT0eXR  
3N2Z+F5jnRBryR+JsJWO7DBSM0UiX+Per9vN90O8jXF4lzeJebJds2x40dv/l5N4VJ0K6zLUnlZWXT4OpO+ WE+fRqXq  
A+flqXyM1OEww3b5v6w0YNFlgVMzBtNxb1a8NXMYPRwuWT1d9vdRSIfs2SHzTdqiLLuPN7wJyy3KnQa28AU+mfE8  
dWdJALkk4jOh91mbnaWXvtXNK/GvFp2U01rjJMny0ojN9DAjwjSZqdIT8W1VG0vWfw0md3lBxou/Fx4Rmuf+ZVCvS  
O5LQUfeZNCjRks/fA1IX4RtYVumE/Ta9j1upmJCmW1TmGmoOWEnn9K8ggVwT948zJ9/gCrBY6LKqB9LTLVkoqgj4V  
DxQla8VtVPyu4Arp2My2NL2bY4xwvvN+KBRz3A/0EBtAwnFFxvQ2K/rYklLcl5DGC03OeK1zvzpE1y7WVZ2Xoz1D0m  
SbzdB8kcyJW7P7eCC3Kf/NCvnsY2dBdB0vaxDSzaq6ZZSEzQ7EVcuOM8gxFFf2GRuqaQr1p/VbtprijWzGu7hDyQ= =  
@gmail.com
```



Add SSH key

Если все выполнено правильно, появится примерно такая запись:

SSH keys

[New SSH key](#)

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.



ForExample

c5:e7:23:b2:79:6d:cf:ed:41:9e:d4:83:f2:4e:ed:4a

Added on Jun 20, 2019

Never used — Read/write

Delete

Для проверки настроек, выполняем команду:

```
ssh -T git@<домен или ip-адрес git-сервера>
```

Например:

```
ssh -T <a href="mailto:git@github.com">git@github.com</a>
```

В ходе выполнения команды, программа запросит: “Уверены ли вы в продолжении соединения”. Это стандартный вопрос безопасности. Отвечаем — yes.

Если все выполнено правильно, программа выведет приветственную речь.

MINGW64:/c/Projects

CloudAdmin@WinServ2016 MINGW64 /c/Projects

\$ ssh -T git@github.com

The authenticity of host 'github.com (140.82.118.3)' can't be established.

RSA key fingerprint is SHA256:nThbg6kXUpJWG17E1IGOCspRomTxdCARLviKw6E5SY8.

Are you sure you want to continue connecting (yes/no/[fingerprint])? yes

Warning: Permanently added 'github.com,140.82.118.3' (RSA) to the list of known hosts.

Hi [REDACTED]! You've successfully authenticated, but GitHub does not provide shell access.

CloudAdmin@WinServ2016 MINGW64 /c/Projects

\$ |