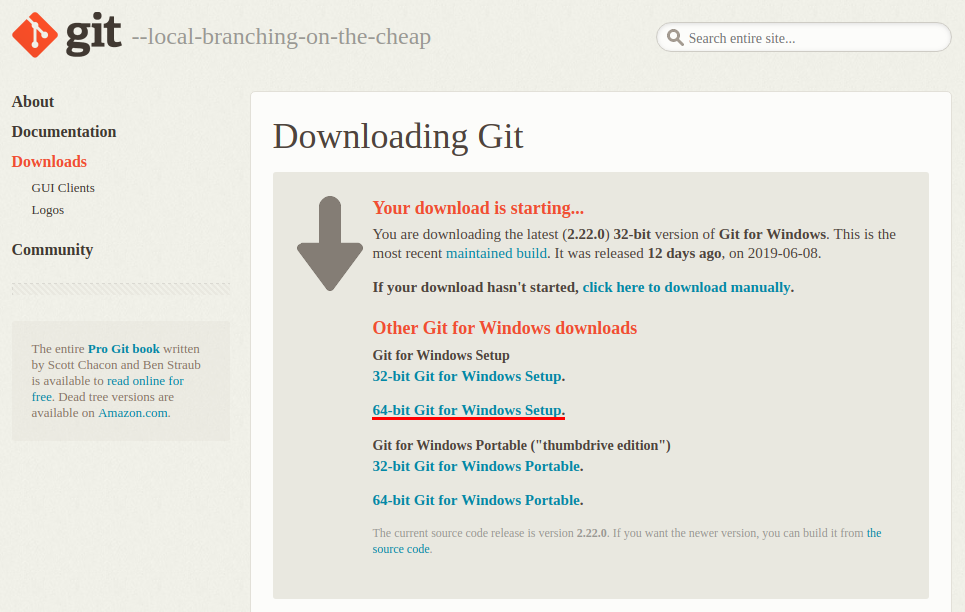
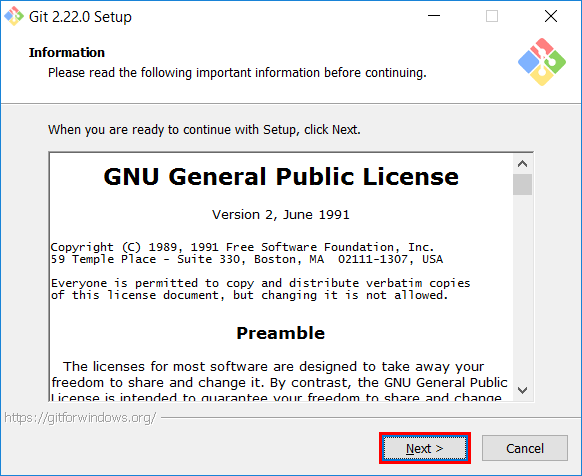
Открываем официальный сайт [проекта](https://git-scm.com/download/win), кликаем по 64-битной версии для Windows.

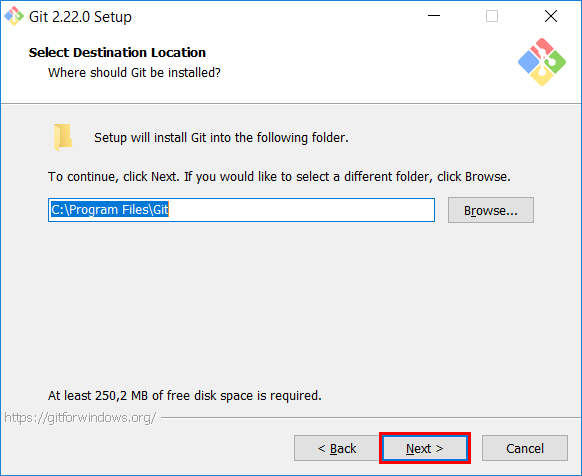
.

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

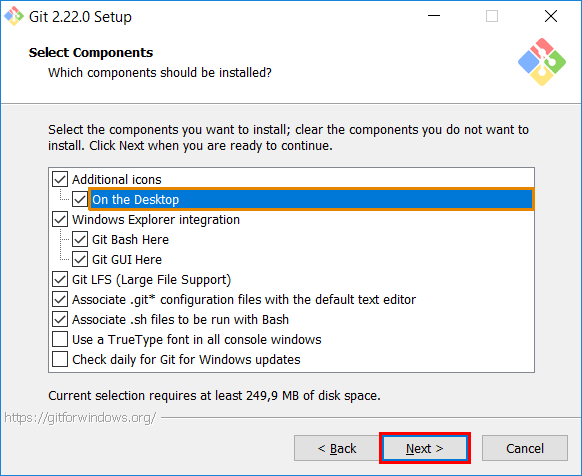
Программа установки приветствует нас лицензионным соглашением. Кликаем по кнопке “Next”.



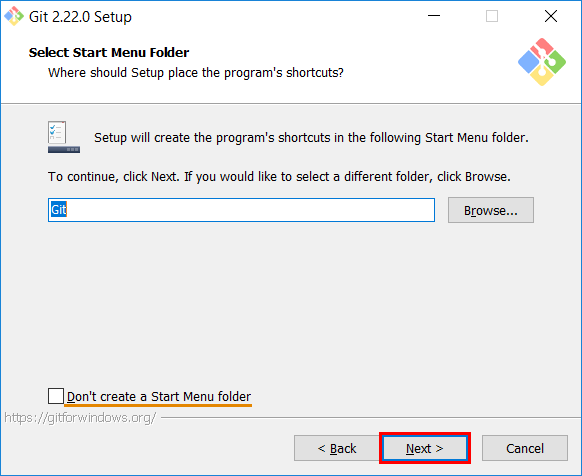
На следующем шаге указываем путь установки. Нас устраивает указанный по умолчанию. Кликаем “Next”.



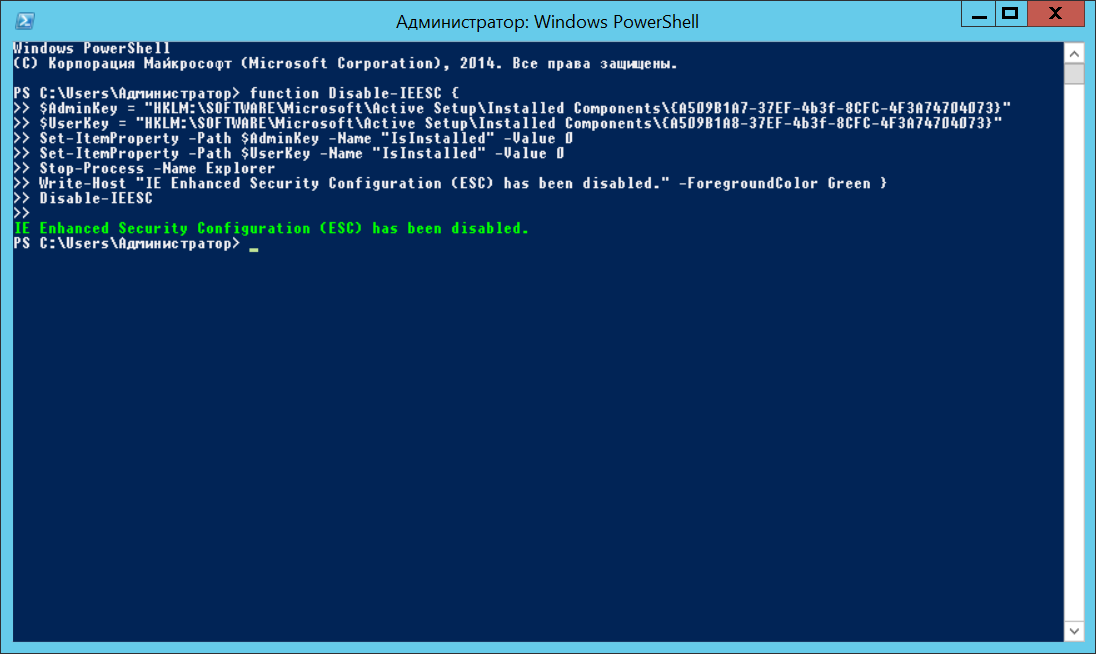
На этапе выбора компонентов, отметим галочкой создание ярлыка Git на рабочем столе и переходим к следующему этапу.



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



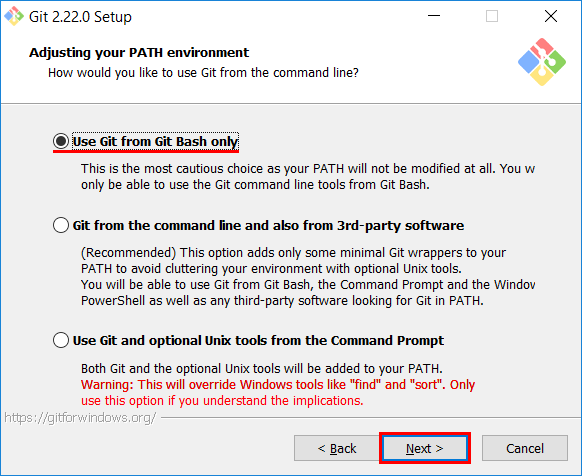
На следующем этапе, установщик предложит выбрать редактор для git по умолчанию. Для новичков лучше использовать nano, для тех, кто любит посложнее — Vim. Хотя допускается использование Notepad++ и прочего. Переходим к следующему шагу.



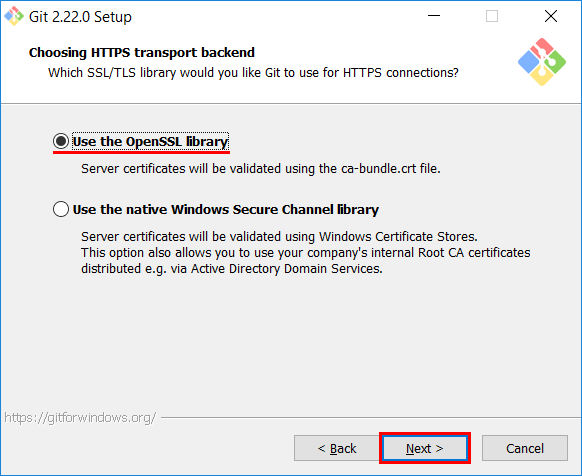
Теперь программа установщик спрашивает как Вы собираетесь использовать командную строку:

* 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.

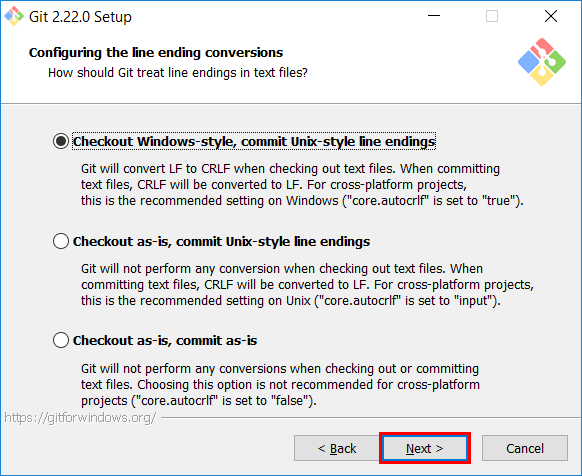


Теперь установщик спрашивает о том, какую библиотеку следует использовать для HTTPS-соединений. Оставляем openssl и переходим к следующему шагу.



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.

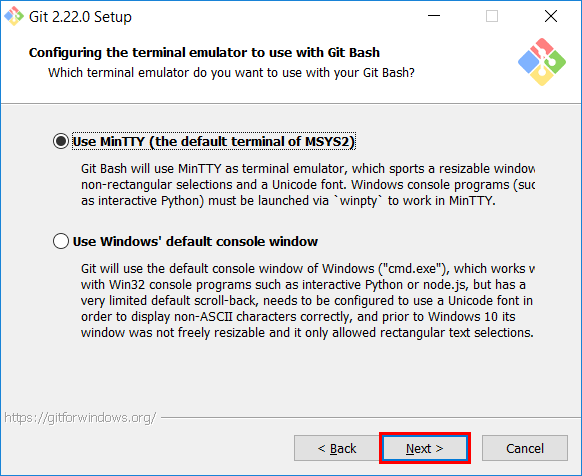
Мы рекомендуем выбрать параметр предложенный системой — отладка файла в формате Windows, а загрузка файла — в Unix-формате. Кликаем по кнопке “Далее”.



На данном этапе, программа запросит каким терминалом пользоваться далее:

* MinTTY — терминал Unix;
* Windows — стандартный терминал (командная строка) Windows.

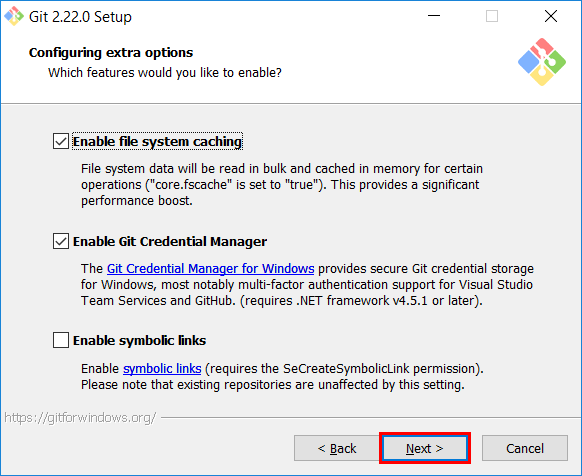
В нашем случае оставляем “как есть” и двигаемся дальше.



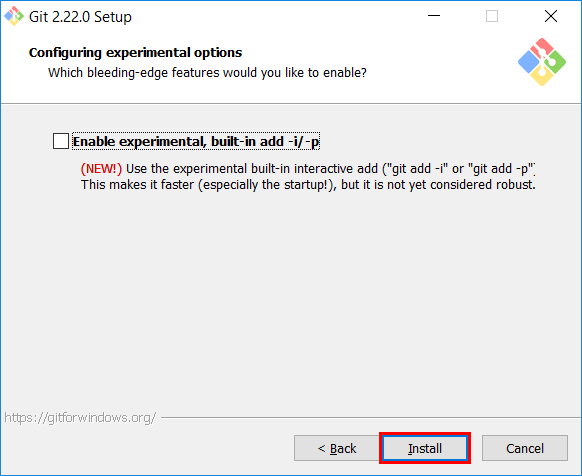
На следующем шаге, установщик попросит отметить необходимые функции:

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

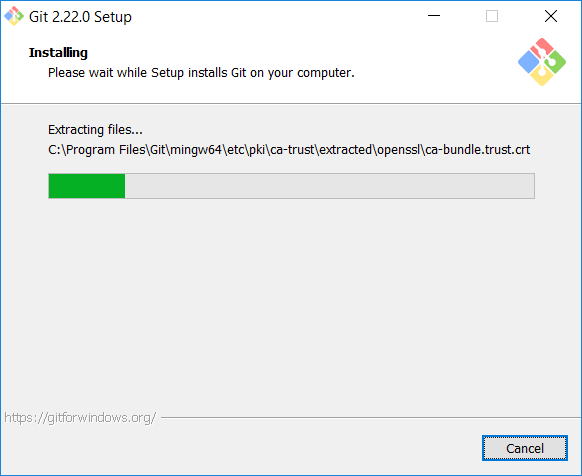
Выбираем то, что необходимо и переходим к следующему шагу.



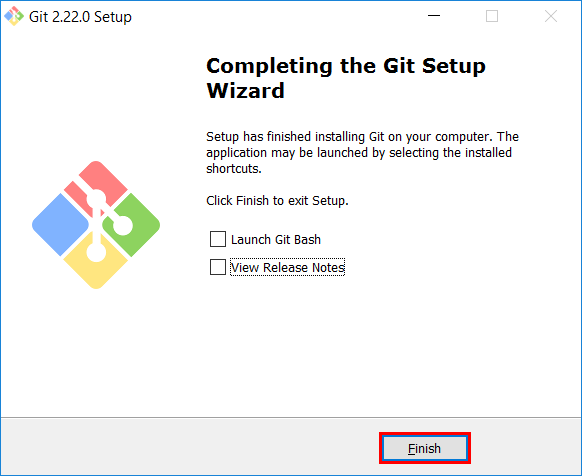
На данном шаге, программа установки предлагает использовать экспериментальную опцию. Использовать ее или нет — решать Вам. Кликаем по кнопке “Install”.



Программа приступает к копированию файлов. Дожидаемся окончания процесса.

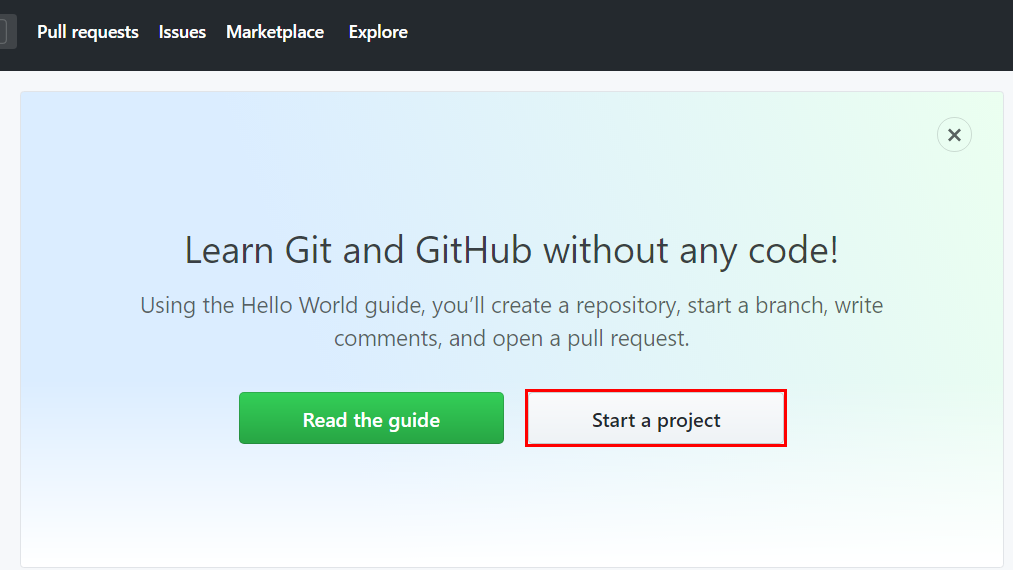


По окончании процесса установки нажимаем “Finish”.

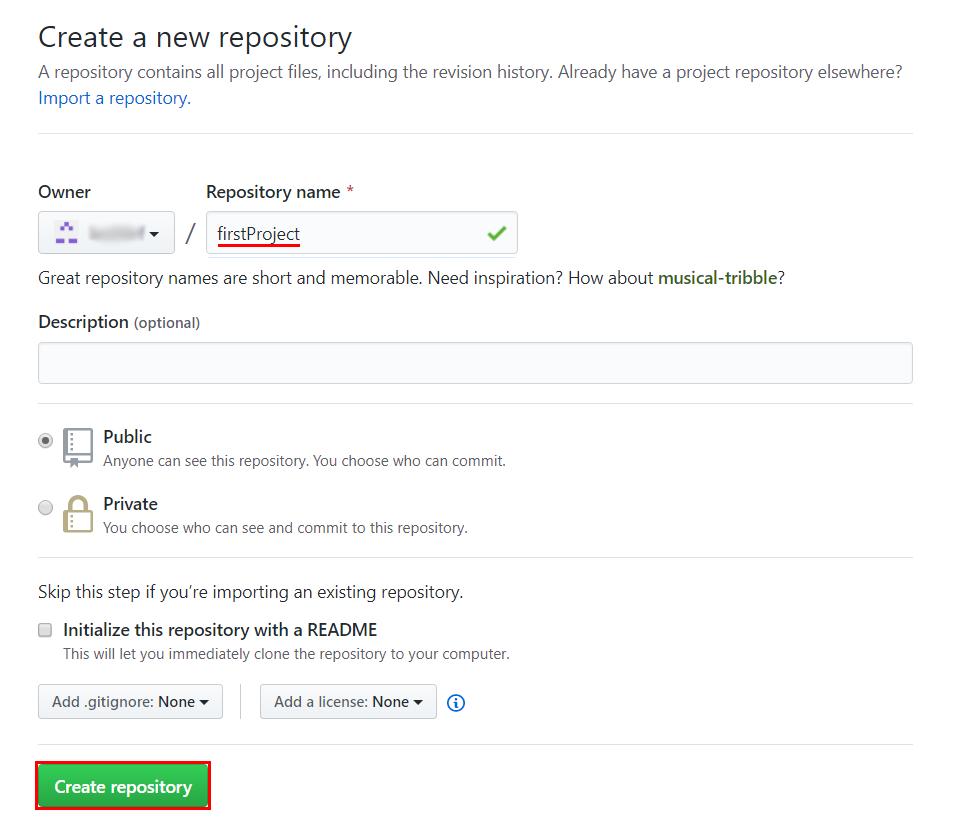


Подключаемся к репозиторию.

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](https://github.com/). Затем авторизуемся на нем и создаем репозиторий. Для этого кликаем по кнопке “Start a project”.



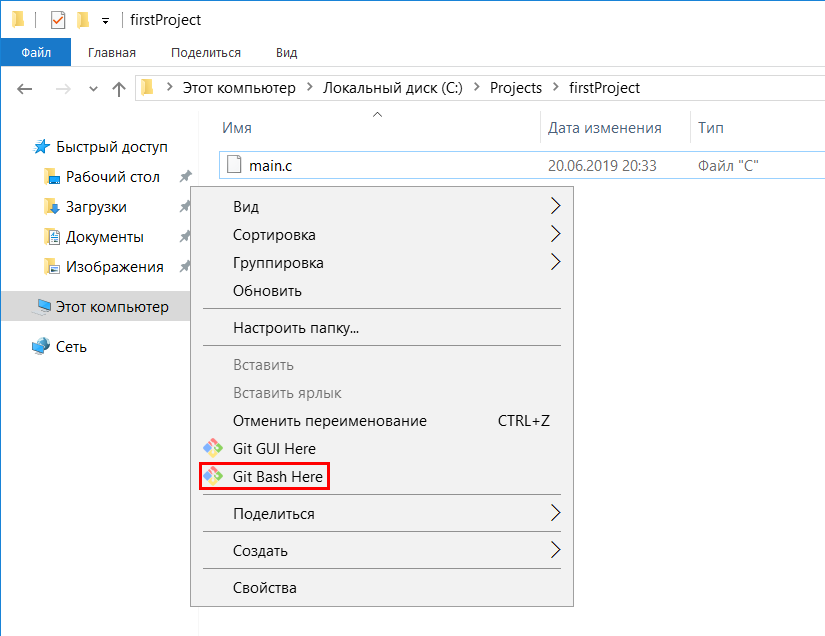
Указываем имя репозитория и кликаем по кнопке “Create repository”.



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”.

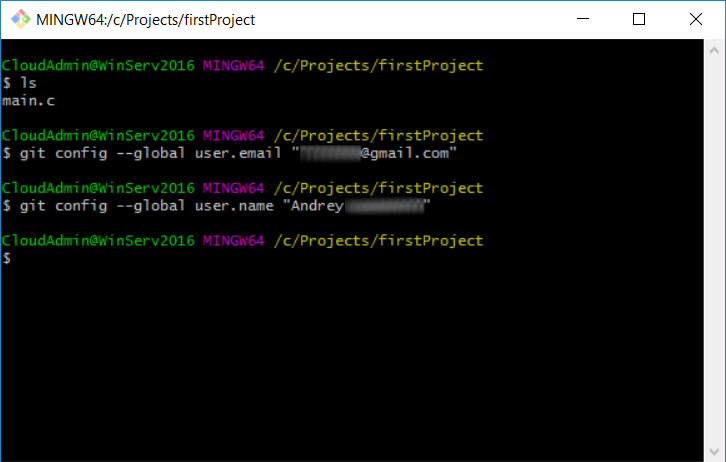


Откроется командная строка. Последовательно выполняем следующие команды:

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

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

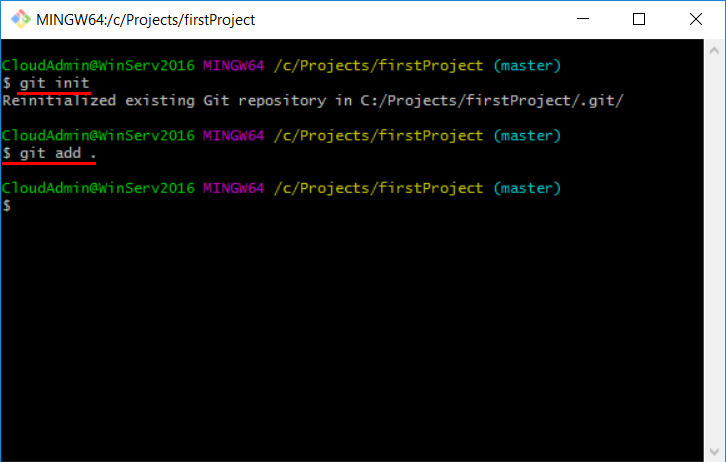
Таким образом мы указали нашу почту и имя.



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

git init

git add .



Выполняем свой первый коммит и добавляем изменения на сервер:

git commit -m "Init"

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

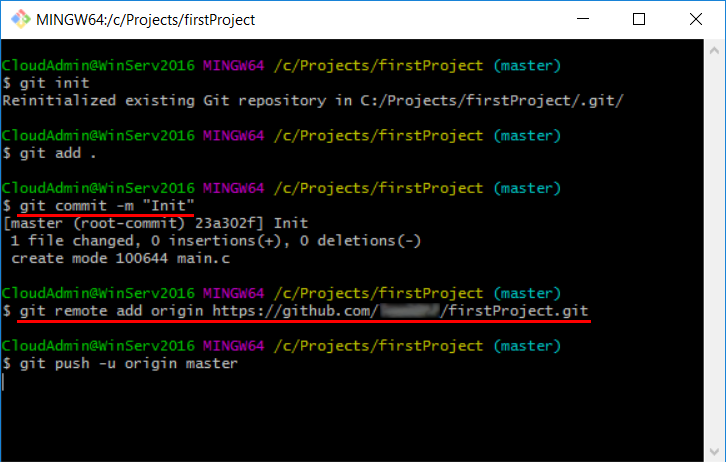
git remote add origin https://github.com/пользователь/название\_репозитория.git

git push -u origin master

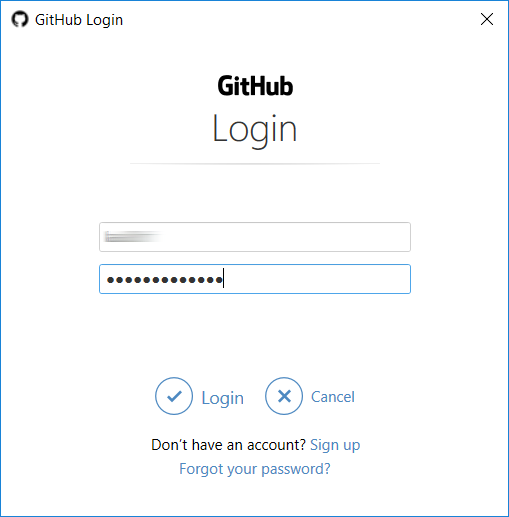
Для gitlab:

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

git push -u gitlab master



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



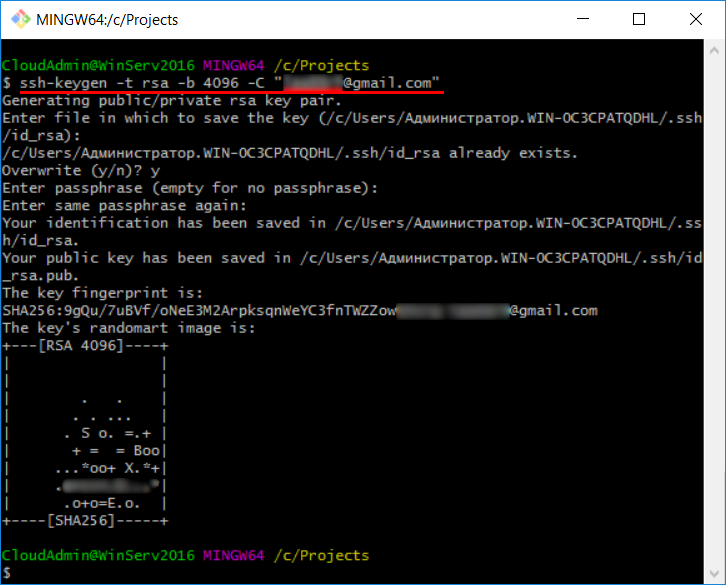
Настройка аутентификации по SSH-ключам.

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

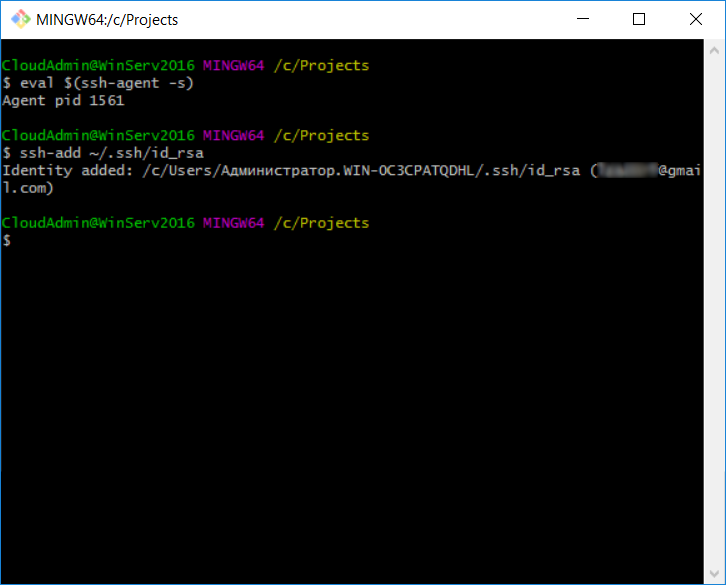


Запускаем SSH-агента:

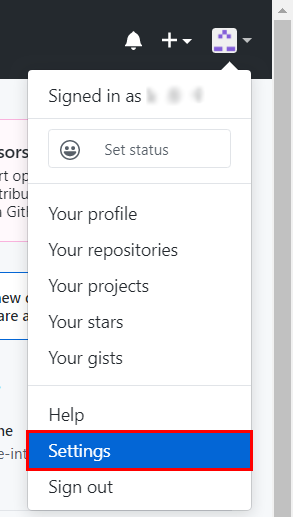
eval $(ssh-agent -s)

Добавляем ключи к ssh-агенту:

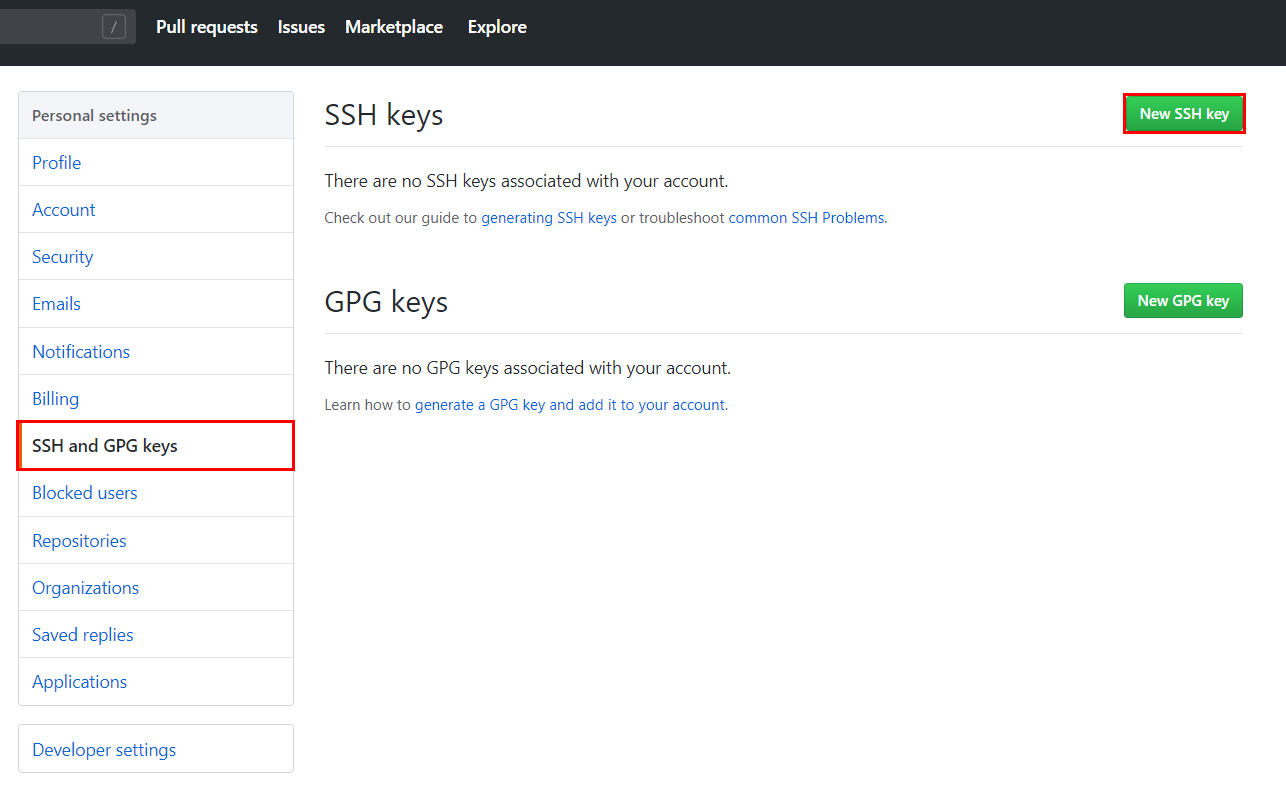
ssh-add ~/.ssh/id\_rsa



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

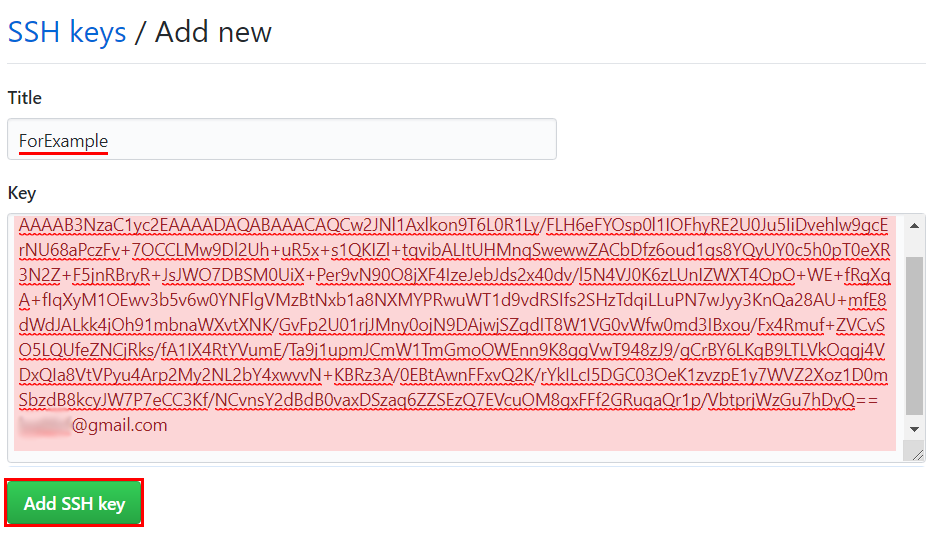


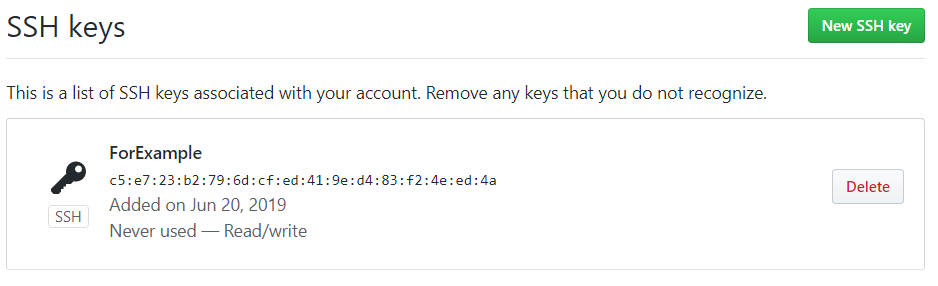
На обновленной странице, в левом меню, выбираем “SSH and PGP keys”, а в основной части окна, напротив SSH keys, кликаем по кнопке “New SSH key”.



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

Кликаем по кнопке “Add SSH key”.



Если все выполнено правильно, появится примерно такая запись:

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.

