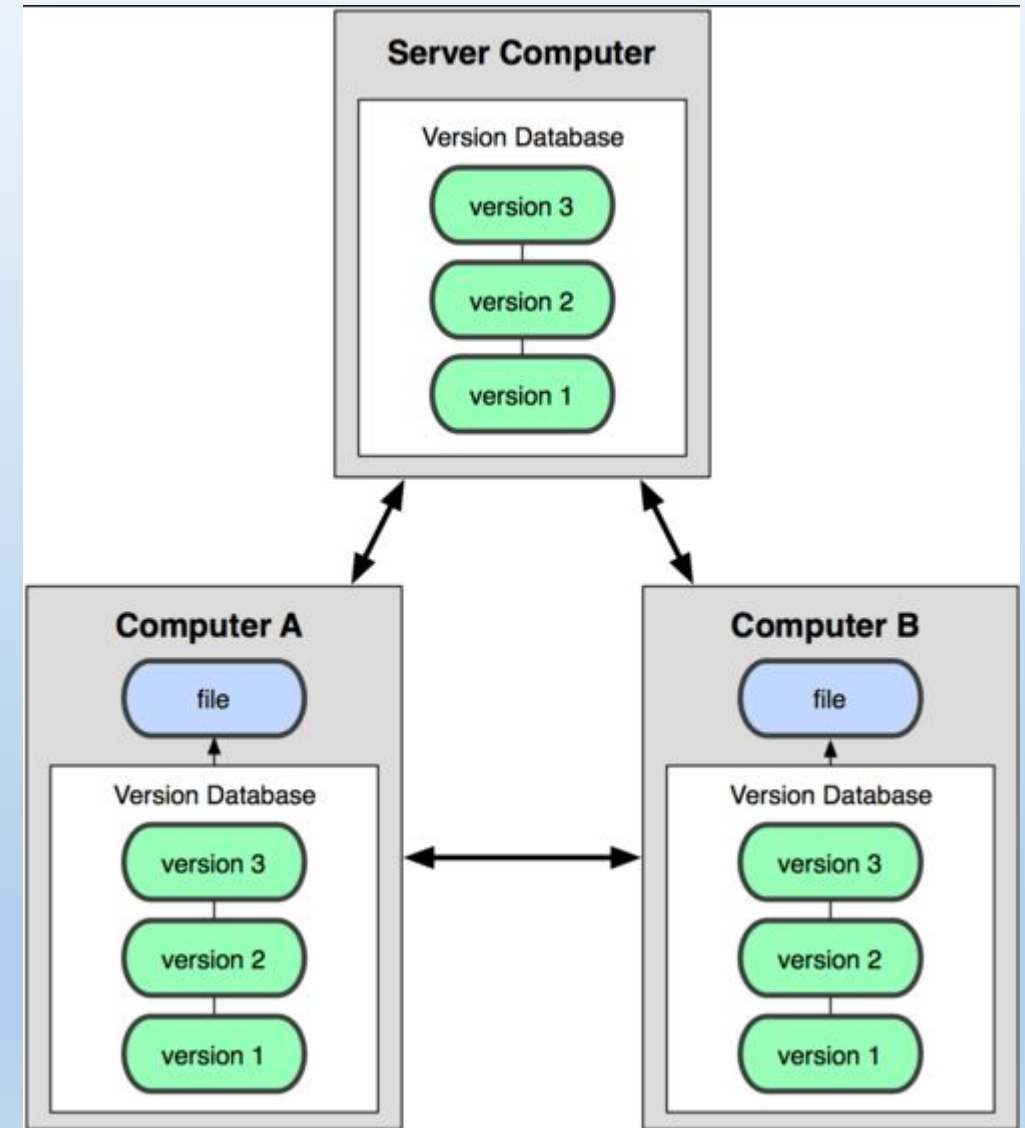


Система контроля версий Git:

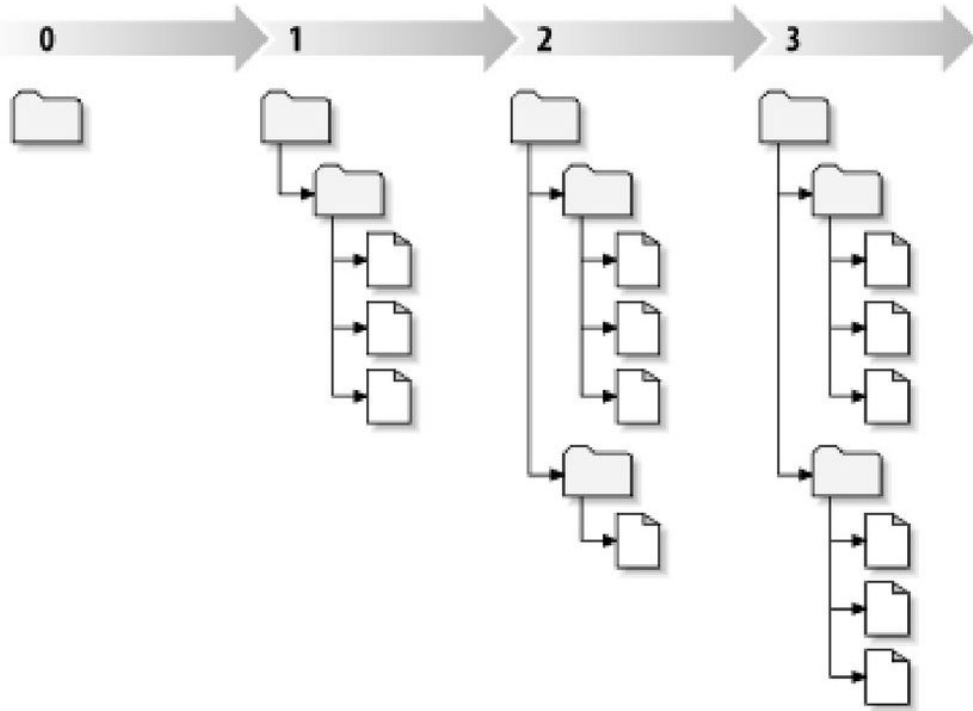
1. Основы
2. Создание репозитория
3. Размещение исходного кода

О контроле версий

Система контроля версий (СКВ) — программное обеспечение для облегчения работы с изменяющейся информацией. Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое.



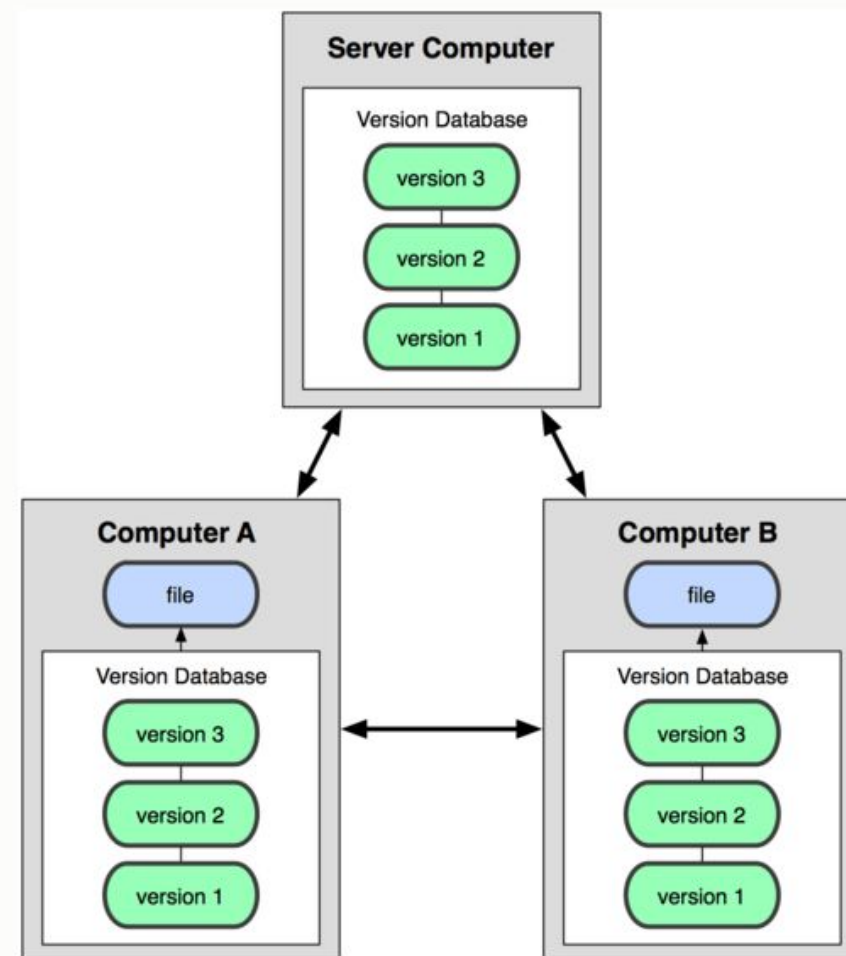
Предпосылки возникновения



Ситуация, в которой электронный документ за время своего существования претерпевает ряд изменений, достаточно типична. При этом часто бывает важно иметь не только последнюю версию, но и несколько предыдущих. В простейшем случае можно просто хранить несколько вариантов документа, нумеруя их соответствующим образом. Такой способ неэффективен (приходится хранить несколько практически идентичных копий), требует повышенного внимания и дисциплины и часто ведёт к ошибкам, поэтому были разработаны средства для автоматизации этой работы.

Распределенные системы контроля версий

Лучший выход из сложившейся ситуации – перенести данные на удаленный сервер, доступ к которому будут иметь все участники процесса. К примеру, Git, Mercurial, Bazaar или Darcs. Рассмотрим работу Git подробнее



Основы Git

Существует несколько открытых репозитириев, позволяющих хранить данные. Зарегистрируемся и создадим свой первый репозиторий по следующему адресу:

<https://github.com/>

Также для дальнейшей работы понадобится Git-приложение, скачать можно с официального сайта:

<https://git-scm.com/downloads>

После регистрации откроется следующее окно. Создадим новый репозиторий (кнопка “New repository”):

The screenshot shows the GitHub homepage. The top navigation bar includes the GitHub logo, a search bar, and links for Pull requests, Issues, Codespaces, Marketplace, and Explore. On the left sidebar, under 'Create your first project', the 'Create repository' button is highlighted with a red circle. The main content area features a 'Following' tab and a 'For you' tab (marked as Beta). A large card in the center encourages discovering interesting projects and people to populate a personal news feed, with an 'Explore GitHub' button. On the right, there are two promotional cards: 'Galaxy 2023' with a 'Register now' button and 'Start coding instantly with GitHub Codespaces' with a 'Get started' button. At the bottom, there is a 'ProTip!' section, a 'Subscribe to your news feed' link, and a footer with copyright information and various links (Blog, About, Shop, Contact GitHub, Pricing, API, Training, Status, Security, Terms, Privacy, Docs). The 'Latest changes' section at the bottom right lists recent updates, such as 'Projects on GitHub Mobile is now generally available' and 'Organization Audit Logs include IP addresses – Public Beta'.

Create your first project

Ready to start building? Create a repository for a new idea or bring over an existing repository to keep contributing to it.

[Create repository](#) [Import repository](#)

Recent activity

When you take actions across GitHub, we'll provide links to that activity here.

Following For you Beta

Discover interesting projects and people to populate your personal news feed.

Your news feed helps you keep up with recent activity on repositories you [watch](#) or [star](#) and people you [follow](#).

[Explore GitHub](#)

ProTip! The feed shows you events from people you [follow](#) and repositories you [watch](#) or [star](#).

[Subscribe to your news feed](#)

© 2023 GitHub, Inc.

[Blog](#) [API](#) [Terms](#)
[About](#) [Training](#) [Privacy](#)
[Shop](#) [Status](#) [Docs](#)
[Contact GitHub](#) [Security](#)
[Pricing](#)

Galaxy 2023 ×

Empower developers.
Supercharge productivity.
Register now for our virtual enterprise event on March 28–31.

[Register now](#)

Start coding instantly with GitHub Codespaces ×

Spin up fully configured dev environments on powerful VMs that start in seconds. Get up to 60 hours a month of free time.

[Get started](#)

Latest changes

- 2 hours ago
Projects on GitHub Mobile is now generally available
- 2 hours ago
Organization Audit Logs include IP addresses – Public Beta
- 3 hours ago
Migrate Bitbucket Server repos to GitHub.com


Далее вводим имя и выбираем тип:

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 *


 KibisavaAnastasiya ▾

 /


tests ✓

Great repository names are short and memorable. Need inspiration? How about [fluffy-guide](#)?

Description (optional)

☒  Public

Anyone on the internet can see this repository. You choose who can commit.

☐  Private

You choose who can see and commit to this repository.

Initialize this repository with:

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

☐ Add a README file

This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore


Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: None ▾

Choose a license

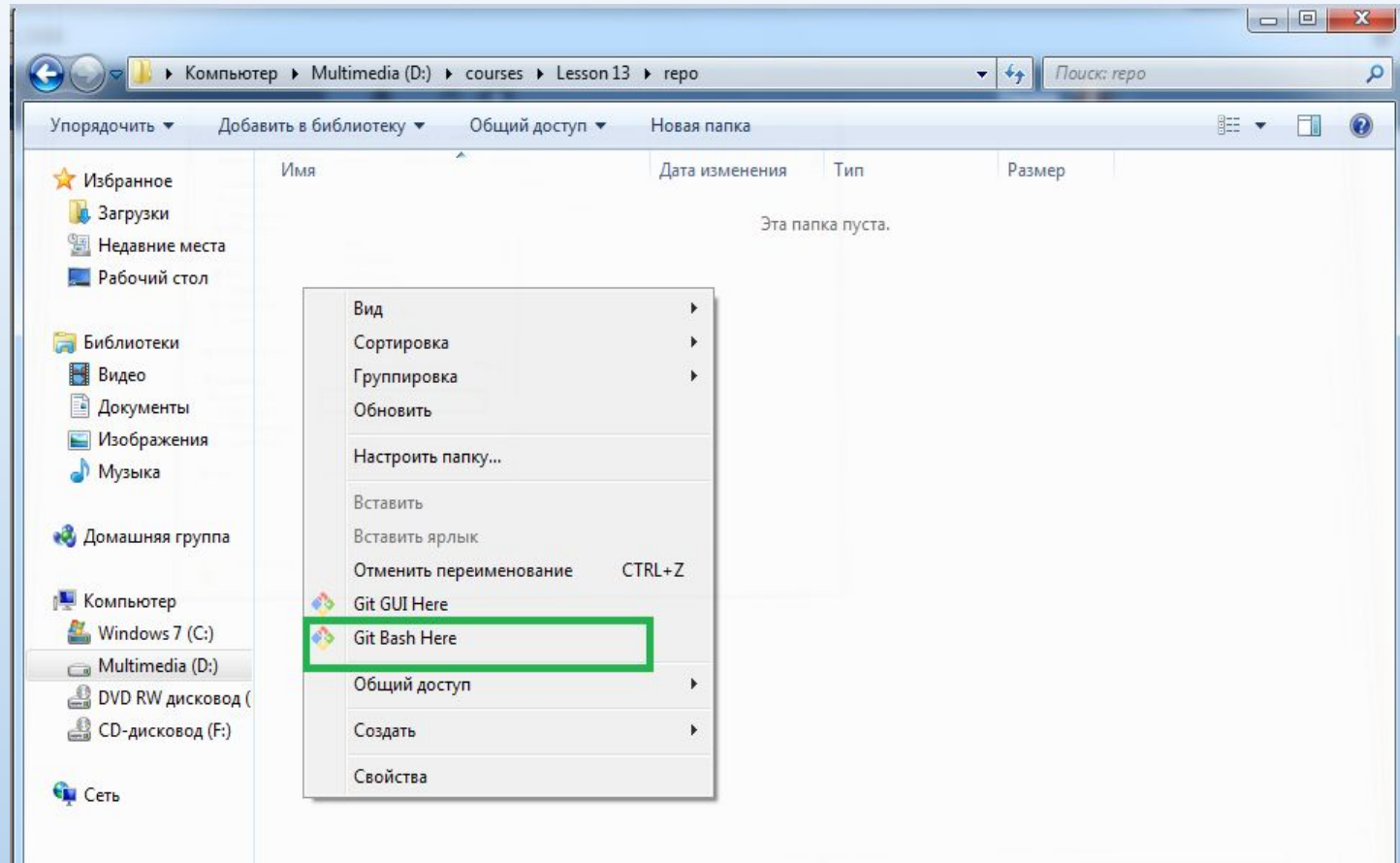
A license tells others what they can and can't do with your code. [Learn more.](#)

License: None ▾

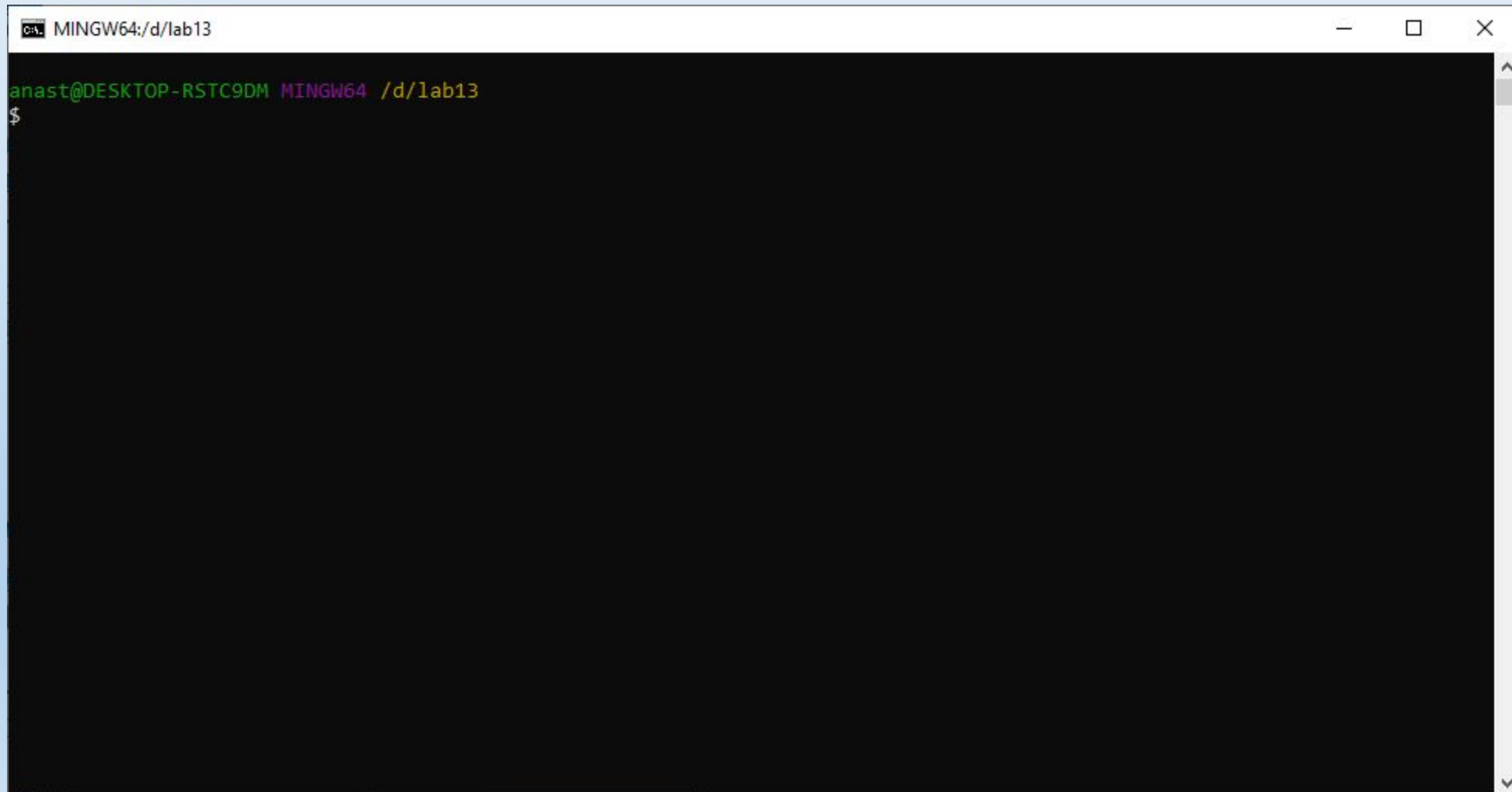
 You are creating a public repository in your personal account.

Create repository

На данном этапе создание репозитория завершено. Пока он пустой, попробуем добавить в него некоторые данные. Для это установим Git-приложение, создадим пустую папку на любом диске и нажмем правой кнопкой мыши:



Откроется окно, внешне очень похожее на командную строку. Дальше работать будем здесь:

A screenshot of a terminal window titled "MINGW64:/d/lab13". The window has a black background and a white title bar. The prompt "anast@DESKTOP-RSTC9DM MINGW64 /d/lab13" is displayed in green and yellow text, followed by a white dollar sign "\$" on the next line. The window includes standard Windows window controls (minimize, maximize, close) in the top right corner and a vertical scrollbar on the right side.

```
MINGW64:/d/lab13
anast@DESKTOP-RSTC9DM MINGW64 /d/lab13
$
```

Привяжем имя и почту к своему git-аккаунту и зададим эти данные в консоли. Для этого нужно использовать следующие команды:

```
$ git config --global user.name «John Doe»
```

```
$ git config --global user.email john@ex.com
```

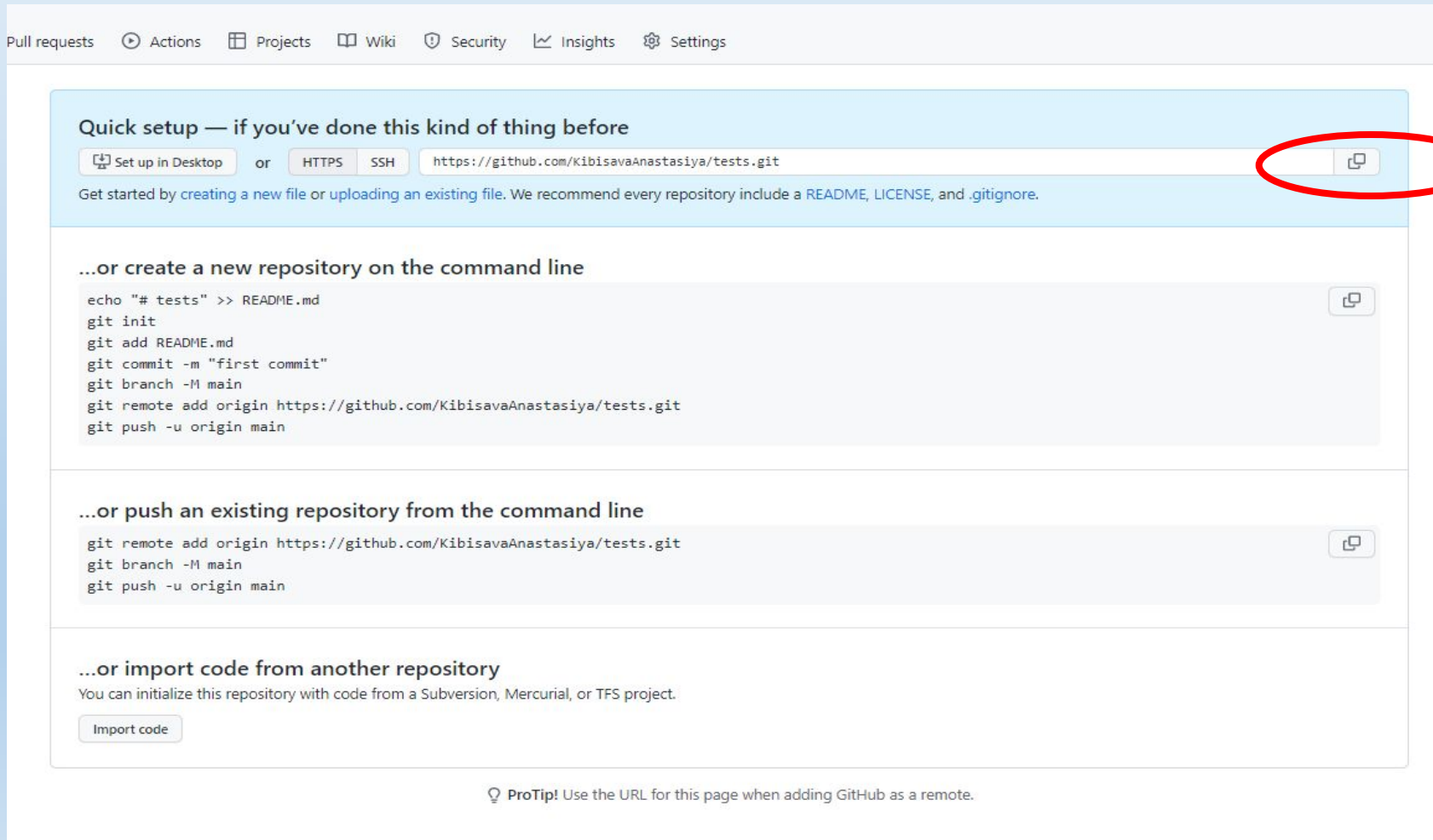
После каждой из команд нажимаем Enter

Инициализируем свой первый локальный репозиторий, используя команду:

```
$ git init
```


На экране увидим сообщение о том, что инициализация прошла успешно.

Свяжем свой локальный и удаленный репозитории, чтобы синхронизировать работу. Для этого перейдем в браузер, откроем наш репозиторий и скопируем его адрес:



Pull requests Actions Projects Wiki Security Insights Settings


Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH `https://github.com/KibisavaAnastasiya/tests.git` 

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).


...or create a new repository on the command line

```
echo "# tests" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/KibisavaAnastasiya/tests.git
git push -u origin main
```



...or push an existing repository from the command line


```
git remote add origin https://github.com/KibisavaAnastasiya/tests.git
git branch -M main
git push -u origin main
```



...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

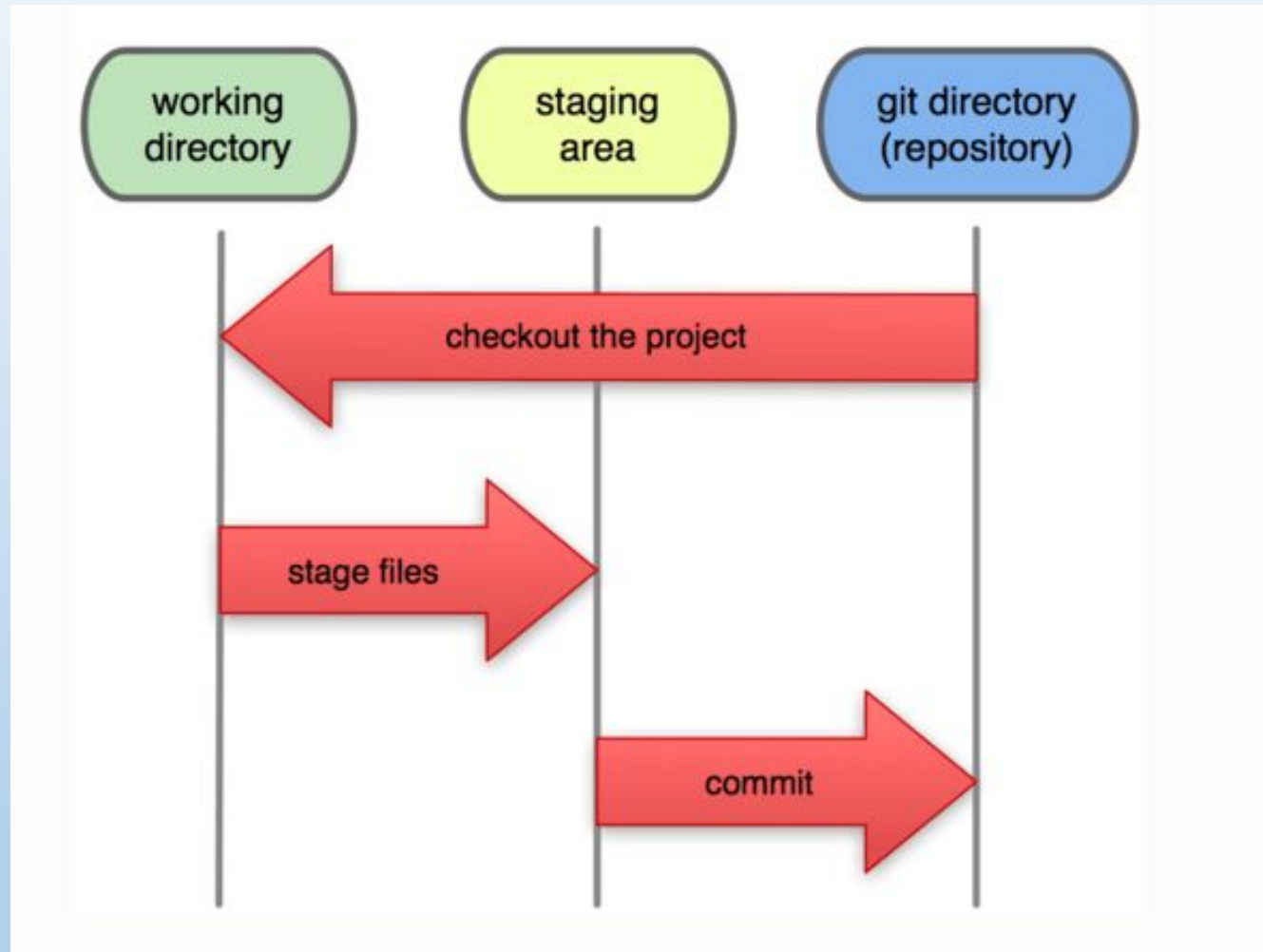
[Import code](#)

 **ProTip!** Use the URL for this page when adding GitHub as a remote.

Три состояния

В Git'e файлы могут находиться в одном из трёх состояний: зафиксированном, изменённом и подготовленном.

"Зафиксированный" значит, что файл уже сохранён в вашей локальной базе. К изменённым относятся файлы, которые поменялись, но ещё не были зафиксированы. Подготовленные файлы — это изменённые файлы, отмеченные для включения в следующий коммит.



Шаги:

- Вы вносите изменения в файлы в своём рабочем каталоге.
- Подготавливаете файлы, добавляя их слепки в область подготовленных файлов.
- Делаете коммит, который берёт подготовленные файлы из индекса и помещает их в каталог Git'a на постоянное хранение

```
$ git add 1.txt
```

```
$ git commit -m 'initial project version'
```

```
$ git push origin master
```

Если предыдущие шаги выполнены успешно, то при открытии вкладки Network в удаленном репозитории мы увидим, что наши коммиты отобразились и там:

