OCS LIFE SCIENCES
A DIVISION OF OCS CONSULTING

Getting Started with Git: An Intro to Version Control

Javier Rodriguez Soto

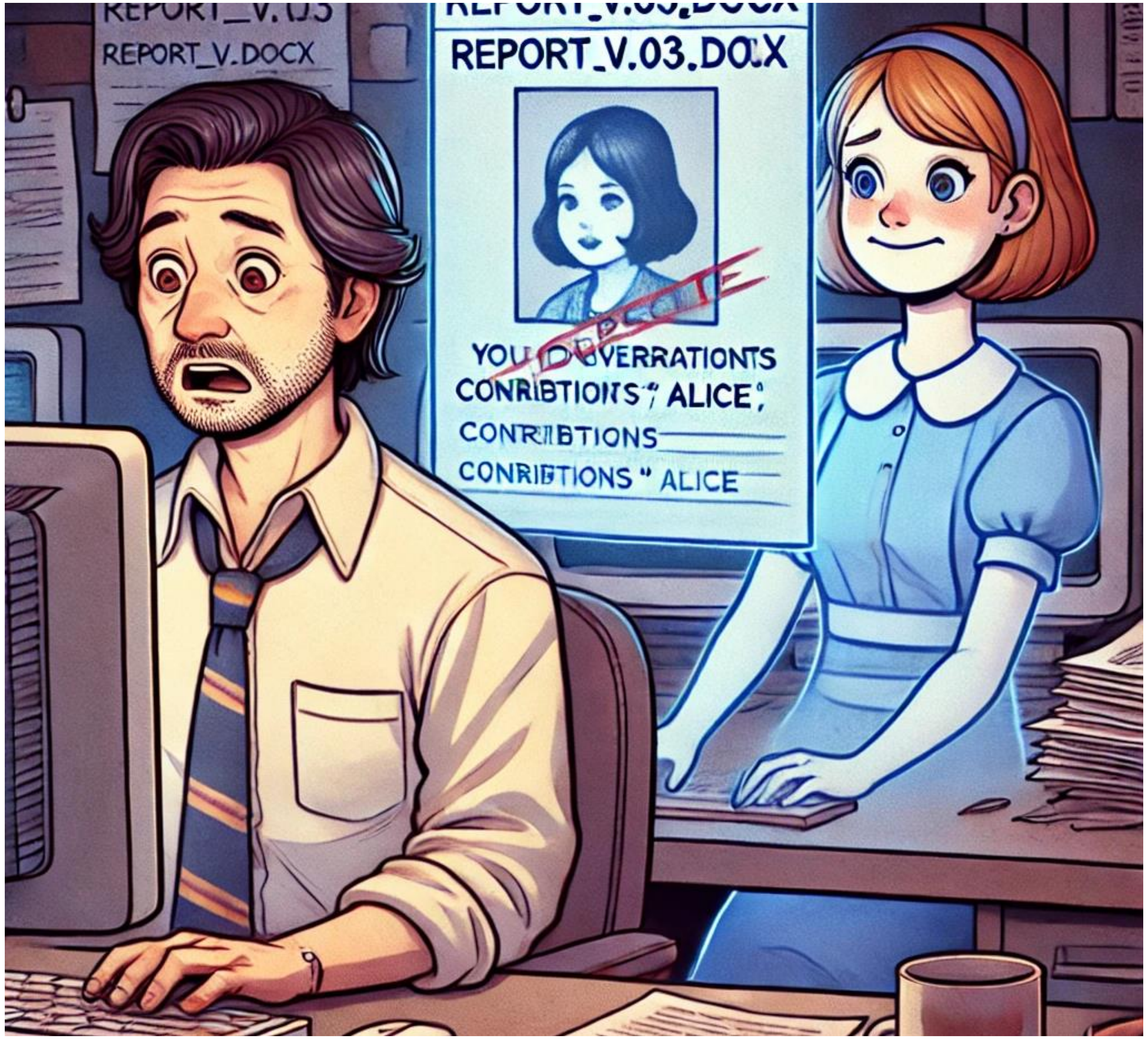
Biostatistician, OCS Life Sciences

git

Lost Edits in 'Parallel Universes'

- Alice and Bob start with the same document, but each saves a new version without realizing the other is working on it.
- They change the same sections differently
- They may change the document structure (Bob removed a section and Alice keeps editing this section)
- Manual merging leads to potential missed edits or conflicting changes.

The Accidental Overwrite Disaster



- Bob accidentally overwrites 'Report_v.03.docx' with his changes, erasing Alice’s recent edits.
- Alice’s contributions are lost, and redoing her work adds frustration and delays.

The 'Final FINAL DRAFT' Chaos

- Alice and Bob both save their final versions of the report, each with a slightly different name.
- Both think they have the 'official' final version.
- Confusion arises: Which version is truly final?

Common issues

- **Losing Track of Versions**
  - **In collaborative projects, it is easy to lose track of which version is the most up-to-date. Multiple versions of the file can lead to confusion and errors.**
- **Conflicting Changes**
  - **Working simultaneously on the same file can lead to conflicting changes, e.g., two team members editing the same line/section.**
- **Manual Renaming**
  - **Manually renaming files (e.g., file_v1, file_v2, etc.) is inefficient and difficult to manage as the number of versions grows.**

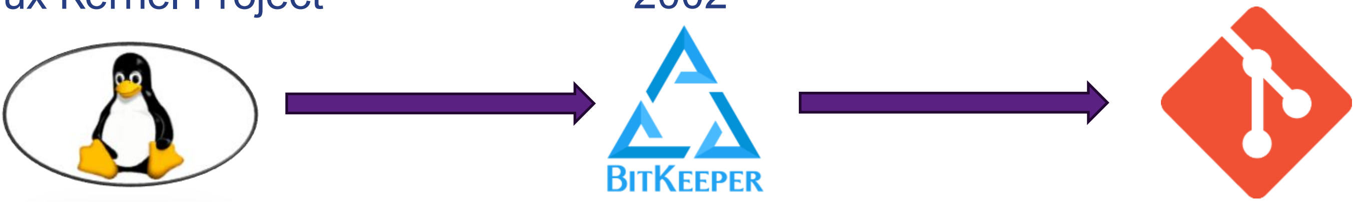
Need for Version Control

What is Version Control?

A system that tracks changes made to files over time, allowing file management and better collaboration.

No need to manually rename file names or losing previous versions.

History of Git

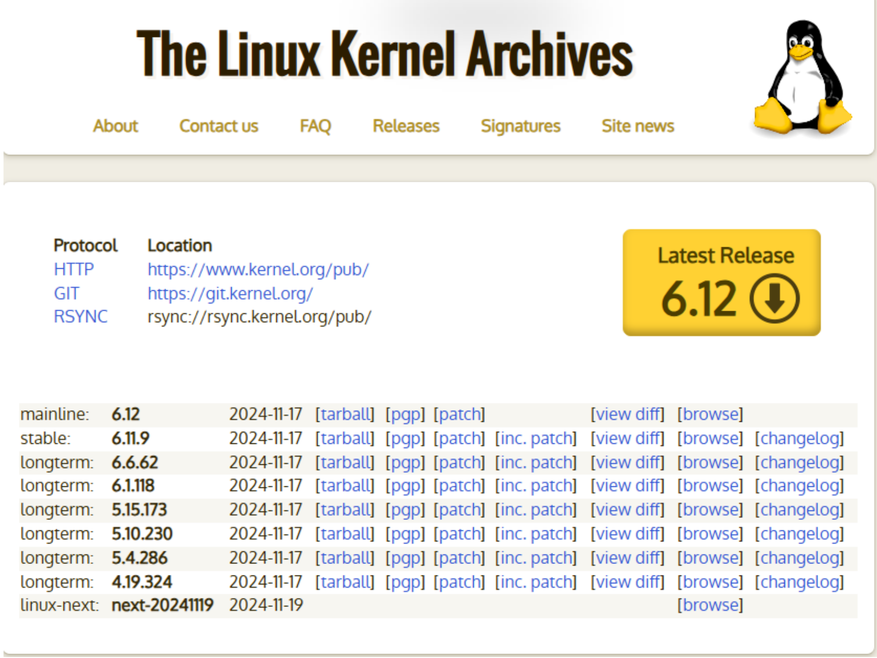

Linux Kernel Project

2002

2005

Development time: 10 days

Linus Torvalds



The Linux Kernel Archives

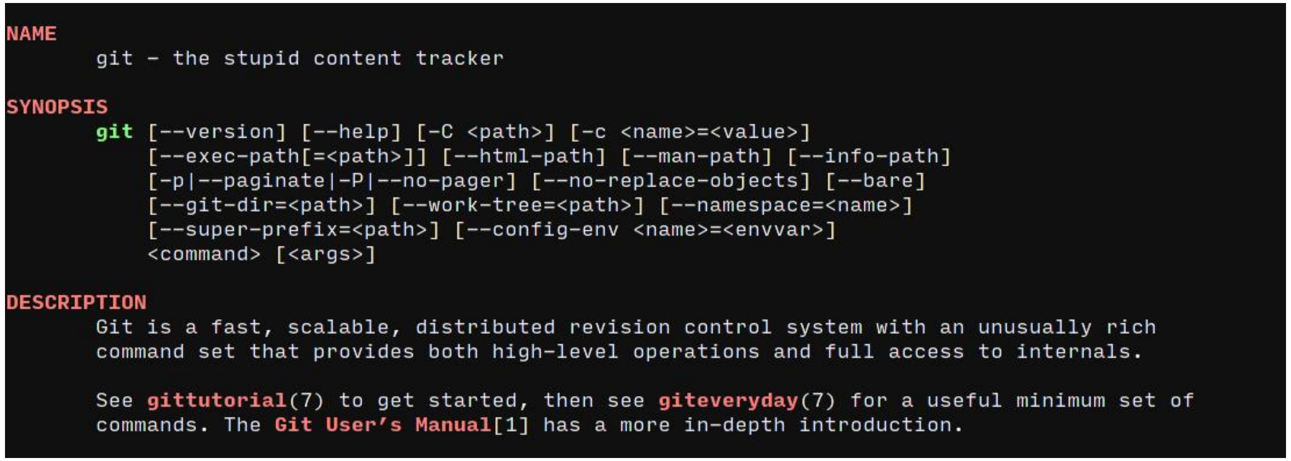
About Contact us FAQ Releases Signatures Site news

Protocol	Location
HTTP	https://www.kernel.org/pub/
GIT	https://git.kernel.org/
RSYNC	rsync://rsync.kernel.org/pub/

Latest Release

6.12

mainline:	6.12	2024-11-17	[tarball]	[pgp]	[patch]		[view diff]	[browse]	
stable:	6.11.9	2024-11-17	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	6.6.62	2024-11-17	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	6.1.118	2024-11-17	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	5.15.173	2024-11-17	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	5.10.230	2024-11-17	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	5.4.286	2024-11-17	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.19.324	2024-11-17	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
linux-next:	next-20241119	2024-11-19						[browse]	



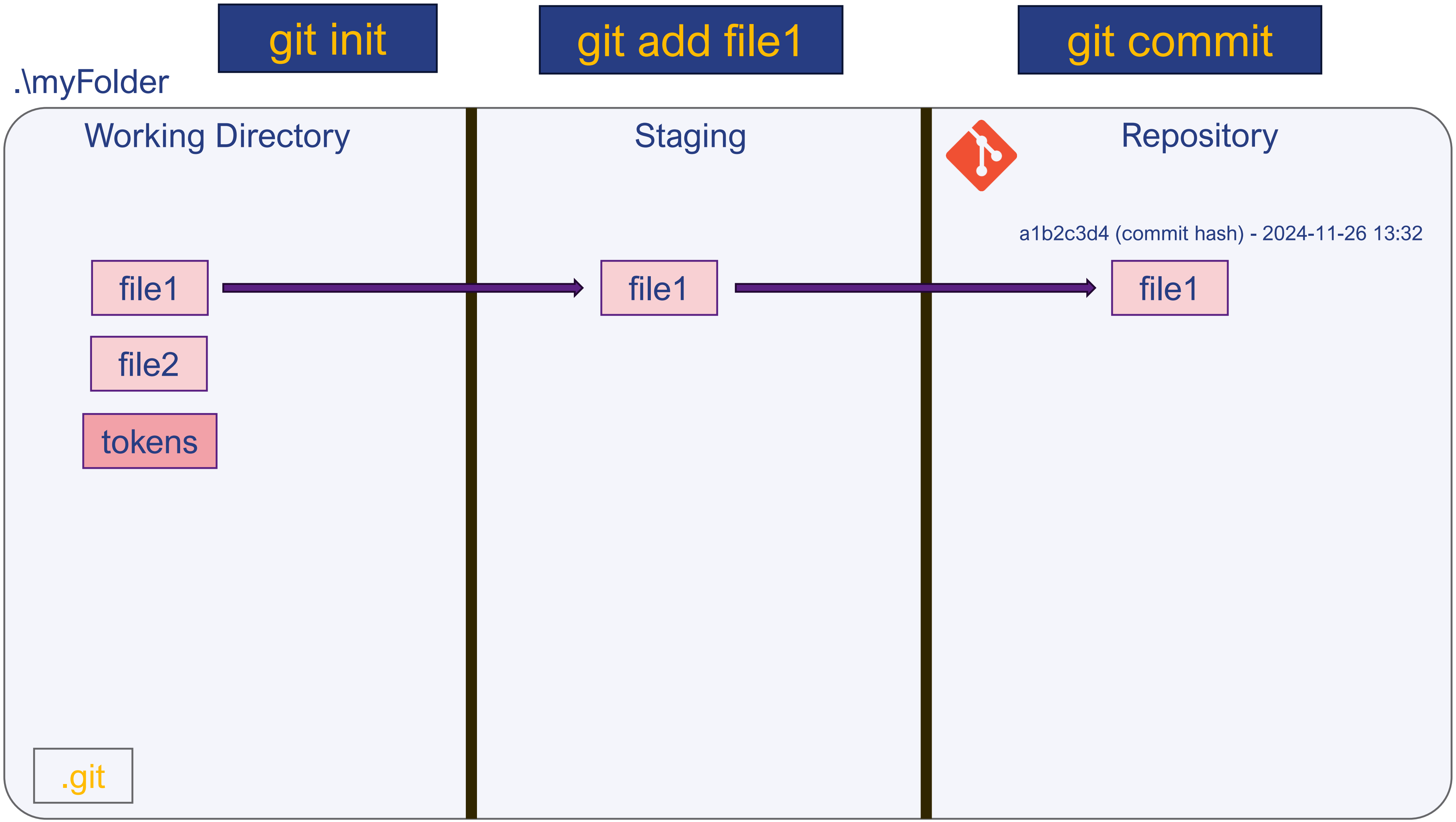
```
NAME
       git - the stupid content tracker

SYNOPSIS
       git [--version] [--help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p|--paginate|-P|--no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           [--super-prefix=<path>] [--config-env <name>=<envvar>]
           <command> [<args>]

DESCRIPTION
       Git is a fast, scalable, distributed revision control system with an unusually rich
       command set that provides both high-level operations and full access to internals.

       See gittutorial(7) to get started, then see giteveryday(7) for a useful minimum set of
       commands. The Git User's Manual[1] has a more in-depth introduction.
```

Why Git? (The Benefits of Using Git)

- Track Changes: Saves a history of changes.
  - Traceability of changes – who did what, when and why (use Description!)
- Centralized trusted location
  - It is easy to find the latest version, check what changed since the last time you worked etc.
- Collaboration: Prevents overwriting each other's work
  - Everyone is using own working copy
  - Feel free to change, play, break things – production files will be OK
- Undo Mistakes: Revert to previous versions
  - You can always return to the state when everything worked fine
- Efficiency: Saves changes instead of full copies
  - Git works fast even for big projects

But mainly, it protects your code from yourself!

Visualizing Git Workflow



git init

git add file1

git commit

.\myFolder

Working Directory

Staging

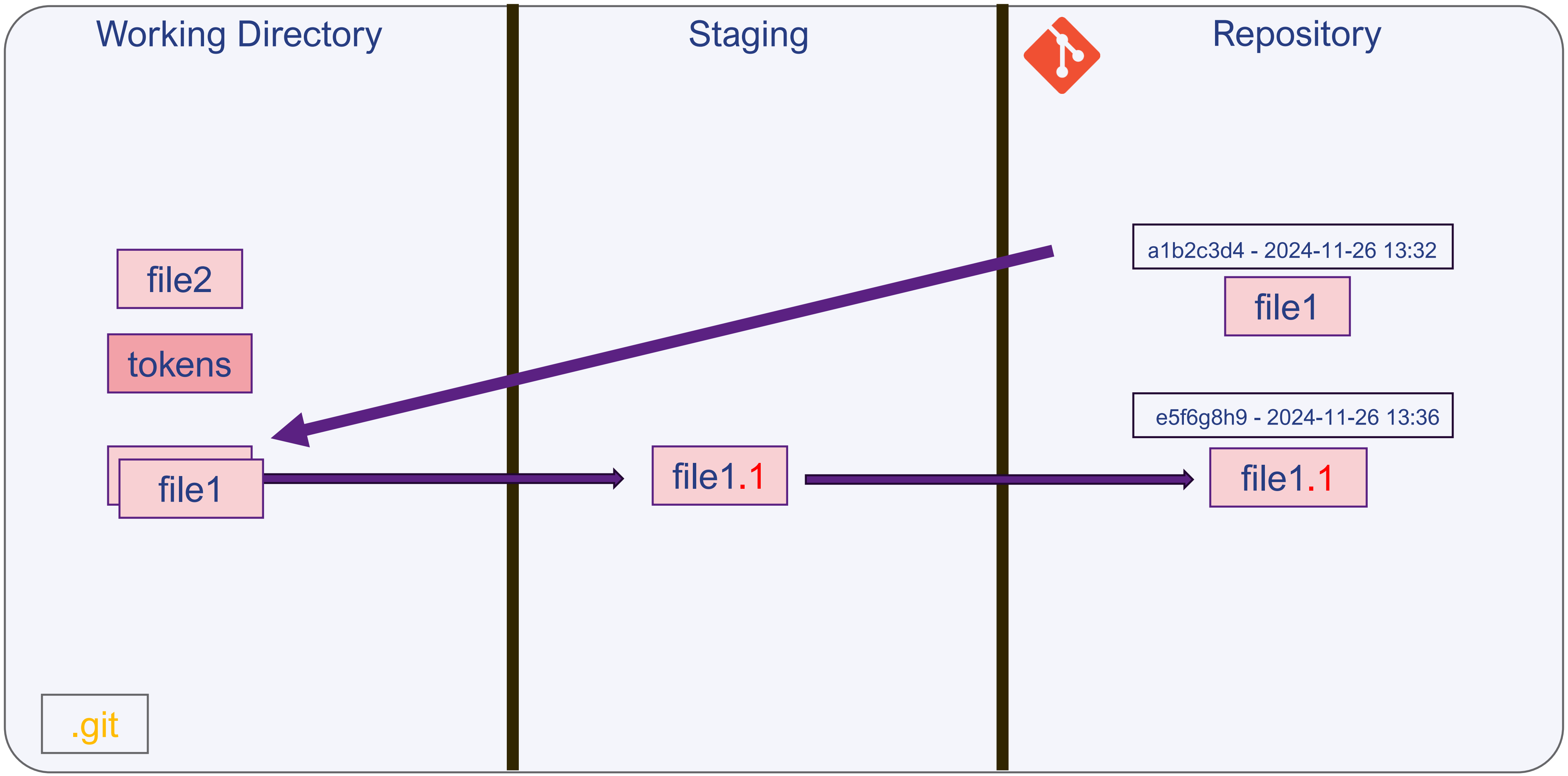
Repository

a1b2c3d4 (commit hash) - 2024-11-26 13:32

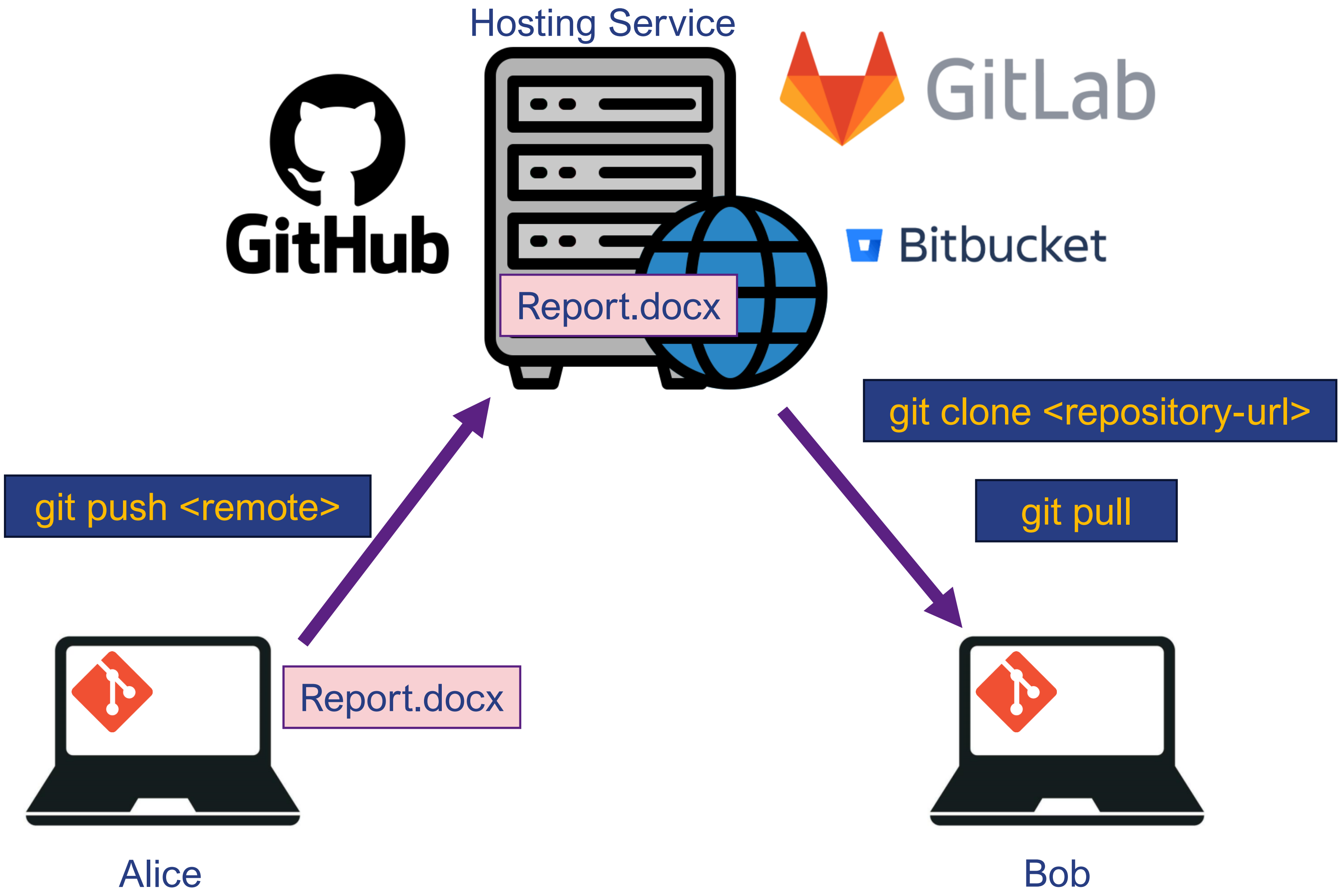
Visualizing Git Workflow (cont.)

git checkout <commit hash>

git log



Collaboration (remote repo)



Hosting Service

git push <remote>

git clone <repository-url>

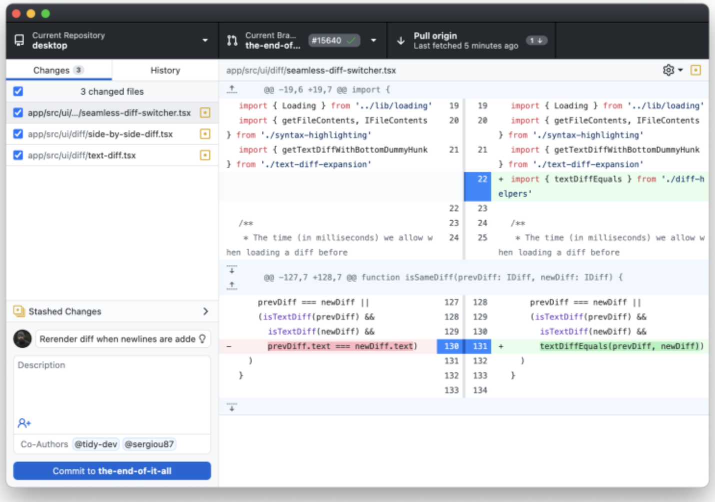
git pull

Short Summary

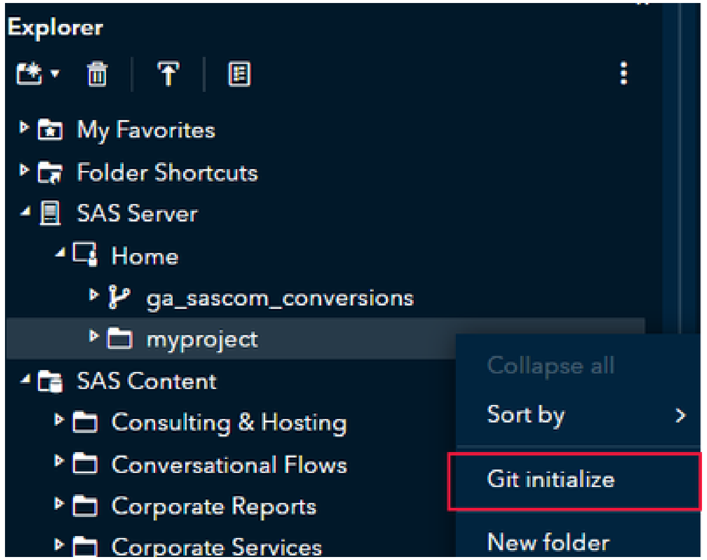
- **What is a (local) Repository?**
  - **A folder storing your project, including files and their version history.**
  - **Command:** `git init`
- **Staging area**
  - **A space to organize changes before the next commit.**
  - **Command:** `git add <file>`
- **Commit**
  - **A saved snapshot of your changes in your repository.**
  - **Command:** `git commit –m “message”`
- **Remote Repository**
  - **A repository hosted on a server (e.g., GitHub, Gitlab, etc.)**
  - **Commands:** `git clone <url>, git push <remote>, Git pull <remote>`

Graphical Git Tools and Integration

- **GitHub Desktop**
  - **Ideal for beginners and GitHub users.**
- **Source Tree**
  - **Ideal for GUI lovers and Bitbucket users.**
- **GitKraken**
  - **For developers that wants powerful experience.**
- **Git Integration with:**
  - **SAS, via SAS Studio and SAS Enterprise Guide (limited).**
  - **R, built-in RStudio.**
  - **MATLAB, built-in MATLAB IDE.**
  - **Etc.**


Current Repository
desktop
Current Bra...
the-end-of...
#15640
Pull origin
Last fetched 5 minutes ago
1
Changes 3
History
3 changed files
app/src/ui/.../seamless-diff-switcher.tsx
app/src/ui/diff/side-by-side-diff.tsx
app/src/ui/diff/text-diff.tsx
Stashed Changes
Rerender diff when newlines are adde
Description
Co-Authors @tidy-dev @sergiou87
Commit to the-end-of-it-all
app/src/ui/diff/seamless-diff-switcher.tsx

```
      @@ -19,6 +19,7 @@ import {
  import { Loading } from '../lib/loading'          import { Loading } from '../lib/loading'
  import { getFileContents, IFileContents          import { getFileContents, IFileContents
} from './syntax-highlighting'                   } from './syntax-highlighting'
  import { getTextDiffWithBottomDummyHunk          import { getTextDiffWithBottomDummyHunk
} from './text-diff-expansion'                   } from './text-diff-expansion'
                                                 + import { textDiffEquals } from './diff-h
                                                 elpers'

  /**                                              /**
   * The time (in milliseconds) we allow w          * The time (in milliseconds) we allow w
hen loading a diff before                        hen loading a diff before
      @@ -127,7 +128,7 @@ function isSameDiff(prevDiff: IDiff, newDiff: IDiff) {
      prevDiff === newDiff ||                          prevDiff === newDiff ||
      (isTextDiff(prevDiff) &&                         (isTextDiff(prevDiff) &&
        isTextDiff(newDiff) &&                           isTextDiff(newDiff) &&
-       prevDiff.text === newDiff.text)          +       textDiffEquals(prevDiff, newDiff))
    )                                                )
  }                                                }
```



What Git is NOT

- × A Backup Solution
  - × Git doesn't duplicate entire projects or keep redundant backups
  - × Version Control, not disaster recovery.
- × Folder Versioning
  - × Git tracks changes to individual files.
- × A File Hosting Service
  - × Git operates locally.
  - × GitHub, Gitlab, or Bitbucket enable collaboration and remote hosting.

Wrap-Up & Resources

- Git tracks changes, facilitates collaboration, and keeps work organized.
- Resources:
  - https://docs.gitlab.com/ee/tutorials/learn_git.html
  - https://www.youtube.com/watch?v=8JJ101D3knE “Learn Git in 1 Hour”
  - Deep dive: https://git-scm.com/book/en/v2

Questions?