Engineer

Git

◆ このサイトの履歴を保存するバージョン管理

どのようなものか

Gitは分散型バージョン管理システムです。ファイル群の状態をスナップショットの連鎖として記録し、過去のどの状態でも見たり、現在と比較したり、戻したりできます。プロジェクト名はGit、コマンドは `git` と書き、同じ道具を指します。

各スナップショットがコミットです。追跡する全ファイルのその時点の状態、変更者と時刻、理由のメッセージ、それより前のコミットへのポインターを保存します。前のコミットは複数の場合もあります。一度見たファイルは一回しか保存せず、変更のないファイルは、次のスナップショットからディスク上の既存のコピーを指します。毎回すべてを複製するより、長い履歴の容量は大幅に小さくなります。

中のものは、内容で名前が決まります。ファイル、ディレクトリーの内容の一覧、コミットを保存する前にバイトからチェックサムを計算し、従来の形式では40桁の十六進数で書くSHA-1ハッシュを名前にします。一バイト変えると名前も、その上に積むすべてのコミットの名前も変わります。これで破損を検知し、何年後でも同じ識別子がまったく同じコードを指します。

分散とは、各コピーが完全なことです。複製は最新ファイルだけでなく全履歴を持ち、ログ閲覧、版の比較、コミットはサーバーなしに手元でできます。共有は別の、意図的な操作です。別のコピーへプッシュするか、そこから取得してマージします。どのコピーもサーバーになれますが、多くのチームでは、合意して選んだ一つのコピーがその役を担います。ホスティングサービス上のコピーであることも多く、これは設計上の決まりではなく慣行です。

仕事は三つの場所を通ります。作業ツリーには編集するファイルがあり、ステージング領域、別名インデックスには、次のコミットへ入る内容があります。一部だけ記録し、残りを後へ回せます。プロジェクトの最上位にあるGitディレクト

リー `.git` には、オブジェクトデータベースと全参照があります。ブランチは一コミットを指す移動可能な名前にすぎず、作るのは一瞬です。

Gitは2005年、Linus TorvaldsとLinuxカーネルのコミュニティーが作りました。それまで使っていたプロプライエタリーソフトウェアのBitKeeperが無料でなくなったためです。目標は、速さ、単純な設計、何千もの並行ブランチへの強力な対応、完全な分散、カーネル規模を扱う能力でした。

概念を判断に生かす

「Gitとは？」を実際の判断に生かしましょう。その概念を、開発または購入するシステムに結び付けて考えるお手伝いをします。 変更を検証し、リリースし、元に戻せる仕組みを整え、金曜の深夜ではなく火曜の午後にお客様へ届けられるようにします。

サービスを見る お問い合わせ →

用語の背後にある前提を検証する

解説とリンク先の実績を参考に、プロジェクトについての質問を整理してください。ツールを選ぶ前に、前提を一緒に検討できます。 GitHub ActionsでGIS SaaSのデプロイを自動化 — デプロイ、データ、報告を手作業から解放

GitHub ActionsでGIS SaaSのデプロイを自動化 お問い合わせ →

なぜ重要なのか

複数人、または一人でも一週間を超えて同じファイルを触れば、何が、いつ、誰によって、なぜ変わったかという問いが生まれます。手作業なら、記憶、`final-v3` のような名前、メールのやり取りに散ります。Gitなら各変更と結び付いてコードの隣に残り、誰でも探せます。

分散は日々の仕事も変えます。どの複製にも全履歴があり、ログ閲覧やコミットで中央サーバーを待たず、サーバーが停止しても全員が作業を続けられます。ブランチは安く、機能、修正、実験ごとに作り、別々にレビューし、準備できたらマージして、主系統を配布可能に保てます。

同じ設計によって、Gitはデータを慎重に扱います。ほぼすべての操作はオブジェクトデータベースへ追加するだけで、コミット済みの仕事は失いにくくなります。誤って消したブランチ、失敗したリベース、間違ったコミットへのリセットは、通常、元へ戻せます。安全網は、各ブランチがいつどこを指したかを記録する手元のreflogです。既定では、到達可能なコミットに関する記録は90日、到達不能なコミットに関する記録は30日保存されます。

Gitは、ソフトウェアを届ける工程で使う多くの道具の共通基盤でもあります。フォージ、レビュー用ツール、ビルドシステム、デプロイスクリプト、静的サイト生成ツールが読み、ページの最終変更日、リリース元のコミット、バグを入れ

た変更など、コードを超える事実の出所になります。

実際の活用例

自社のコードにも、その三つの面が見えます。

第三者のホスティングにあったソースコードは、今は自社のハードウェアで動かすGitサーバーにあります。OSが管理する単一バイナリーで、SSH中心、管理用ウェブ画面なしを理由に選びました。複製とプッシュはSSHの23231ポートを通りますが、待ち受けはループバックだけです。ホスト外からは踏み台を経由し、匿名と鍵なしの接続はどちらも拒否し、宣言したリポジトリーはすべて非公開です。代償も明記します。自分で運用すれば、認証、アクセス制御、保存、バックアップも引き受けます。

複数マシンの完全な履歴を、バックアップと取り違えがちです。自社のバックアップと復元テストの実装は、事業者のスナップショットを超えて復旧できるよう設計しました。resticスクリプトとsystemdユニットは、設定された場合のデータベースダンプ、暗号化し重複を除いたスナップショットのSFTPによる別事業者への保存、保持方針に沿う削除、整合性検査に対応します。別の月次処理は、既知のファイルを一時領域へ復元し、バイトを比べます。復旧可能な状態の証拠には、実際の実行が必要です。2026-10-27の記録ではリポジトリー未設定でタイマーは無効、完了した実環境の復元は記録されていません。ノートPCの複製をいくら増やしても、この欠けた証拠は補えません。

履歴はサイト自体にも情報を供給します。クレジットは、W3C Feed Validatorが、フィードの多くの項目は更新日が同じと警告し、それは正しいと記します。日付はGitの履歴から取り、多くのページを少数のコミットで書いたためです。架空の日付より、警告を見せる方を選びます。履歴は、多くが同時に起きたことも含めて、そのまま示します。

一つのコマンドで、その事実を読み戻せます。次は、一ファイルを最後に変更したコミットの短い識別子、ISO 8601形式のコミット日時、件名を表示します。

```
git log -1 --format='%h %cI %s' -- README.md
```

問題が起きるところ

- **一度コミットした秘密は履歴に残ります。** 次でファイルを消しても、最新から消えるだけで、全複製がすでに持つ前の版には残ります。漏えい済みとして認証情報を交換してください。後の履歴の書き換えは広がりを狭めても、その間のコピーは古いコミットを持ち続けます。
- **強制プッシュが他人の仕事を置き換えます。** `git push --force` は、送信先のブランチが送る内容の祖先かという検査を止め、途中で同僚が送ったコミットをブランチから落とします。 `--force-with-lease` は、最後に見た位置をまだ指す場合だけ更新します。
- **複製をバックアップと見なしてしまいます。** ノートPCのコピーは遅れ、持つのは所有者が取得したブランチだけで、そのコピーを使って意図的に復元したこともありません。課題、アクセス規則、フォージ自体のデータは、そもそもリポジトリーの外です。
- **大きいバイナリーファイルが、すべての複製の容量を永久に膨らませます。** 各ファイルの各版を保ち、全複製が取得するため、誤って入れた動画を翌日消しても、その後に複製するすべてのマシンへ届きます。
- **一か所だけの仕事は、その場所とともに消えます。** リセットを救うreflogは一台にあり、期限もあります。送っていないコミットはほかになく、何か月も放置したブランチは主系統から離れ、戻すマージ自体が一つの仕事になります。

参考資料

- Git文書 — プロジェクト自身の参照マニュアル、書籍、動画です。
- Pro Git — 初回のコミットから内部のオブジェクトデータベースまで、全編をオンラインで読める書籍です。
- Gitとは — スナップショット、手元の操作、チェックサム、三つの状態を説明する章です。
- Gitの短い歴史 — 2005年にカーネルのコミュニティーが自作した理由と目標です。
- git-reflog — 誤操作前にブランチが指していた場所を探す方法と、その記録の保存期間です。
- git-push — 強制更新の各オプションと、それぞれの保護範囲です。



用語集

実際の制約に合わせて選ぶ

デプロイの流れ、運用上のリスク、緊急対応の担当者をお知らせください。検査、復旧手段、明確な責任の所在を伴う変更の範囲を定められます。

料金を見る お問い合わせ →

用語を実務に結び付ける

「Gitとは？」がお客様の組織にどう当てはまるかをご相談ください。現在のシステムの具体例が一つあれば、話を始められます。

サービスを見る

welcome@engineer.company +45 50 10 61 84 デンマーク、コペンハーゲン

このページを持ち歩く

連絡先カード

心を込めて丁寧に制作： D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

著作権 © 2025 – 現在 · Engineer ApS · すべての権利を留保します。

印刷元： https://engineer.company/ja/glossary/git/