Let’s GIT Started
Collaborating with others with Version Control

James Harr:
Sr NetDevOps Engineer, Internet2
Abstract

Join us for a talk exploring Git, the most popular version control system. Review fundamental concepts in Git including commits, branches, merging, and synchronizing. Learn effective workflow patterns on platforms like GitLab, GitHub, or Gitea. Gain insights into how team workflows mature. Finally, we’ll explore tips and techniques for becoming a more effective Git power-user. Whether you’re a novice, seasoned developer, individual contributor, or team lead, there’s something new to learn about revision tracking with Git.
Why GIT?

- Track Changes
- Switch between versions
- Undo changes
- Collaborate with a Team
- Keep a full copy of history on your computer
What is GitHub / GitLab / Gitea / BitBucket

GitHub ≠ GIT

GitHub/GitLab/etc:
  • A remote repository to share code
  • Create Issues / Tickets
  • Create Pull Requests & Review Code
  • UI for viewing code & code history
Getting Started with the GIT CLI

% git clone git@github.com:project/repo1.git
Cloning into 'repo1'...
remote: Enumerating objects: 11730, done.
remote: Counting objects: 100% (66/66), done.
remote: Compressing objects: 100% (66/66), done.
remote: Total 11730 (delta 29), reused 0 (delta 0), pack-reused 11664
Receiving objects: 100% (11730/11730), 2.67 MiB | 2.16 MiB/s, done.
Resolving deltas: 100% (8018/8018), done.

% cd repo1
Saving & Sharing Changes

% echo "New Note in README" >> README.md

% git add README.md

% git commit -m "Update README"

% git push
Saving & Sharing Changes

- **add** to index
- create a **commit**
- **push** to a remote

https://twitter.com/ChrisStaud/status/1554091413272494080
Pulling Changes

% git pull

From github.com:project/repo1
* branch        main   -> FETCH_HEAD
Updating 6c109b9..064fadf
Fast-forward
  README.md     | 24 ++++++++++++++++++++---
  Other.md      |  4 +++--
2 files changed, 6 insertions(+), 10 deletions(-)
Pulling changes

% git pull
From github.com:project/repo1
  * branch        main   -> FETCH_HEAD
hint: You have divergent branches and need to specify how to reconcile them.
hint: You can do so by running one of the following commands sometime before
hint: your next pull:
hint:   git config pull.rebase false  # merge
hint:   git config pull.rebase true   # rebase
hint:   git config pull.ff only      # fast-forward only
hint: You can replace "git config" with "git config --global" to set a default
hint: preference for all repositories. You can also pass --rebase, --no-rebase,
hint: or --ff-only on the command line to override the configured default per
hint: invocation.
fatal: Need to specify how to reconcile divergent branches.
Branches

Feature A

commit 1

Feature A commit

Feature A merge

Feature A merge

Feature B

commit 1

Feature B commit

Feature B merge
Branches - Viewing

% git branch
* main

% git branch -av
* main 064fada Added feat 1
remotes/origin/main 064fada Added feat 1
Branches - Creating a new branch

% git checkout -b foo
Switched to a new branch 'foo'

% git branch foo
% git checkout foo

% git branch -m new-name  # rename branch
Branches - Creating a new branch

% git branch -av

main       064fadf   Added feat 1
* foo       064fadf   Added feat 1
remotes/origin/main 064fadf   Added feat 1
Branches - Creating a new branch

% echo "Change" >> README.md
% git add .  # add all changes
% git commit -m "Demo change"

% git branch -av
main 064fadf Added feat 1
* foo cfadf3d Demo Change
remotes/origin/main 064fadf Added feat 1
Branches - Creating a new branch

% git push -u        # or --set-upstream

Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote: This repository moved. Please use the new location:
remote:   git@github.com:project/repo1.git
remote:
remote: Create a pull request for 'foo' on GitHub by visiting:
remote:   https://github.com/project/repo1/pull/new/foo
remote:
To github.com:project/repo1.git
 * [new branch]  foo -> foo
branch 'foo' set up to track 'origin/foo'.
Branches - Creating a new branch

% git branch -av
main                   064fadf   Added feat 1
* foo                  cfadfd3d   Demo Change
remotes/origin/main    064fadf   Added feat 1
remotes/origin/foo     cfadfd3d   Demo Change
Branch Naming - Recommendations

- `feature/new-widget` # adding something new
- `fix/invalid-vlan-id` # fixing something
- `doc/howto-add-device` # adding docs

- `182-howto-add-device` # tracking an issue ID

- `doc/182-howto-add-device` # hybrid

Easily differentiate between `main` and `topic` branches
Merging / Pull Requests

Most teams: Just use the GitHub/GitLab/etc WebUI to merge

- Creates a better audit-trail
- Allows for code-review/comments prior to merging
- Allows for CI/CD to run prior to merging
- Many teams will lock down `main` so only the WebUI can merge
Tips for using Pull Requests

• Keep branches topic-specific
• Share early, share often
• Branches can be updated after PR is created
  PR will update automatically
  CI/CD can re-run for branch to catch issues
• Utilize Code Review
  Tech & learn from others
  Spot issues early
  Free spell-check if you work with Karl
• Step gradually into requiring review & using CODEOWNERS
Viewing Commit Graph

% git log --oneline --graph

* 1c0d9c39 - (HEAD -> example1) feat(pdp): Add cloud-connect attributes to PDP (2 minutes ago) <James H
* 8315d94c - feat(vn): Add BGP neighbor description if none is set (2 minutes ago) <James H
* f5725929 - Adding global SRLG distribution back into the ISIS config (2 minutes ago) <Chad
* 7cbbafa6 - Fix/rpl pas blended add asn (11 days ago) <Chad Sorrell
* f947072d - Merge branch 'feature/bgp-peer-telemetry' into 'main' (12 days ago) <Karl Newell
* 6047226c - update interval to 240s - matches TSDS expected interval (12 days ago) <Karl
* 66742e75 - Add BGPPeerData sensor-group (12 days ago) <Karl Newell
* 1d6b375b - Merge branch 'fix/netflow-outbundlemember' into 'main' (12 days ago) <Karl Newell
* 78cda5d - fix(nodconfig): Remove outbundlemember from flow monitor maps (12 days ago) <Karl Newell
* 4d17bc15 - Merge branch 'fix/rpl-pas-blended-update2' into 'main' (12 days ago) <Karl Newell
* 12ad17f1 - Added PerfSonar v6 /48 to EBGP-RE-ORIGINATED prefix-set (12 days ago) <Chad
* 2ac3b438 - Adding PerfSonar originated prefix to EBGP-RE-ORIGINATED (12 days ago) <Chad
Viewing Commit Graph

% git config --global alias.lol "log --graph
 --pretty=format:'%Cred%h%Creset
-%C(yellow)%d%Creset %s %Cgreen(%cr) %C(bold blue)<%an>%Creset' --abbrev-commit"

% git lol      # --all

... or just google "git lol"

https://www.edwardthomson.com/blog/advent_day_9_git_lol.html
GitUp

- Visualize GIT
- Follows your changes
- Can make some changes
- Free

https://gitup.co/
Cleaning up a Branch for Commit

Enter **git rebase**
- Rewinds & re-plays commits on a new “base”
- Rebasing on-top of `origin/main` probably all you need
- Why?
  - Keep history linear
  - Rewrite history and fix mistakes

... Unless you like histories that look like Guitar Hero
GIT Rebase

main

my-branch

origin/my-branch

Time
GIT Rebase
Rebasing like a Pro

% git rebase origin/main

% git rebase -i origin/main
Rebasing like a Pro

% git rebase -i origin/main

pick 7cbb0faf Fix/rpl pas blended add asn
pick 12dee55e Adding global SRLG distribution back into the ISIS config
pick e5c9a2dd feat(vn): Add BGP neighbor description if none is set
pick 3c2183ae feat(pdp): Add cloud-connect attributes to PDP

# Rebase f947072d..9b5dd048 onto f947072d (4 commands)
#
# Commands:
# p, pick <commit> = use commit
# r, reword <commit> = use commit, but edit the commit message
# e, edit <commit> = use commit, but stop for amending
# s, squash <commit> = use commit, but meld into previous commit
# f, fixup [-C | -c] <commit> = like "squash" but keep only the previous
Conventional Commits

% git commit -m 'feat: Add option to specify VLAN'
% git commit -m 'fix: Incorrect interface description'
% git commit -m 'test: Add automated tests'
% git commit -m 'refactor: Rewrite config generator'
% git commit -m 'revert: Accidental change of hostname'
% git commit -m 'feat(switch): Enable IPv6 RA Guard'

https://www.conventionalcommits.org/en/v1.0.0/
Alternative to rebase

% git rebase origin/main # Just prepare
% git reset origin/main  # Leaves changes in WD

% git add ...
% git add --patch ...  # My favorite
% git commit -m ...

% git push -uf        # Force push, set upstream
Other favorite tools

% tig  https://jonas.github.io/tig/
       https://www.youtube.com/watch?v=z0mb0NteYjg

% gitui  https://github.com/extrawurst/gitui
Table of Contents (if needed)

• Topic 1
• Topic 2
• Topic 3
SECTION HEAD

Additional text (if needed)