Git Ready

Armaja LaRue-Hill (Your favorite of the 3 interns...sometimes)

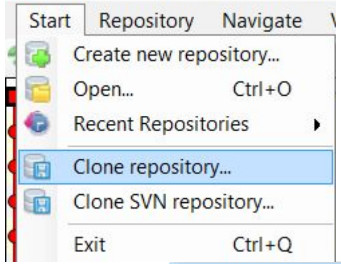
01/24/18

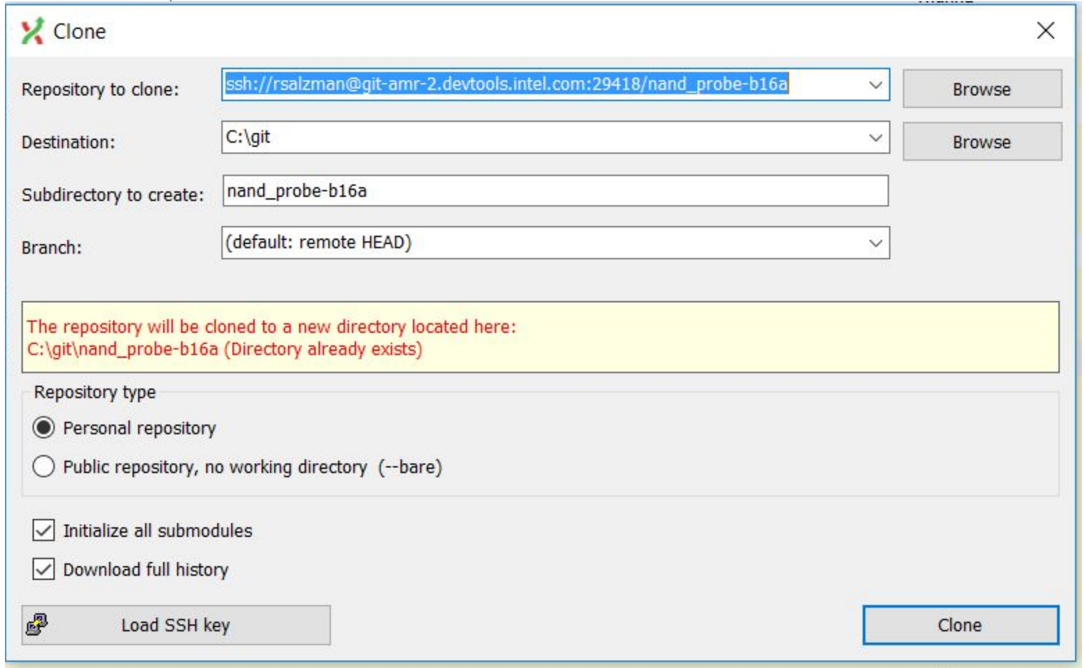
Agenda

1. Beginning Concepts of Git
    - I. Cloning & Committing
    - II. Pushing, Fetching & Pulling
    - III. Branching
    - IV. Stashing
    - V. Local vs Remote Branches
    - VI. Staging and Tracking
2. Advanced Concepts of Git
    - I. Rebasing vs Merging
    - II. Centralized Workflow
    - III. Merge Conflicts

Cloning and Committing

git clone(svn checkout): copies a repository into a newly created directory





Repository to clone: ssh://rsalzman@git-amr-2.devtools.intel.com:29418/nand_probe-b16a

Destination: C:\git

Subdirectory to create: nand_probe-b16a

Branch: (default: remote HEAD)

The repository will be cloned to a new directory located here:
C:\git\nand_probe-b16a (Directory already exists)

Repository type

● Personal repository

○ Public repository, no working directory (--bare)

☑ Initialize all submodules

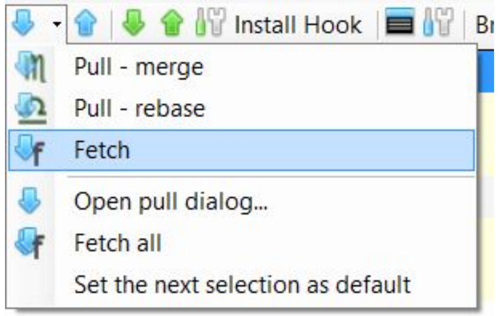
☑ Download full history

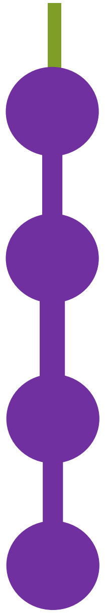
git commit: Record the changes to the repository.

- Every time you commit something, it has a reference number, or commit ID
- Anytime you modify something, you must add then you can commit
- If you have a commit with the same name, you can run into merge conflicts
- Need a commit message (In GUI)

Remote

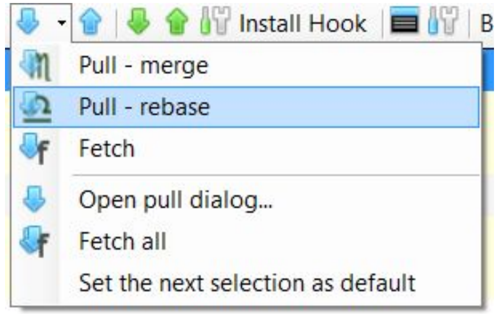
git fetch





Commits

git pull –rebase master



Master

Pushing , Fetching, and Pulling

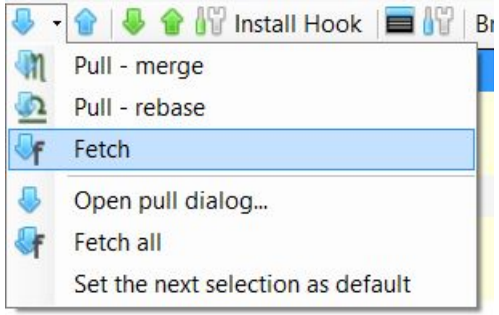
git push (svn commit): 'push' commits on your local branch to a remote repository

You can push tags (the commit ids)

git pull rebase (svn update): git fetch and rebase in one command. Grabs remote's copy of current branch merges into local copy

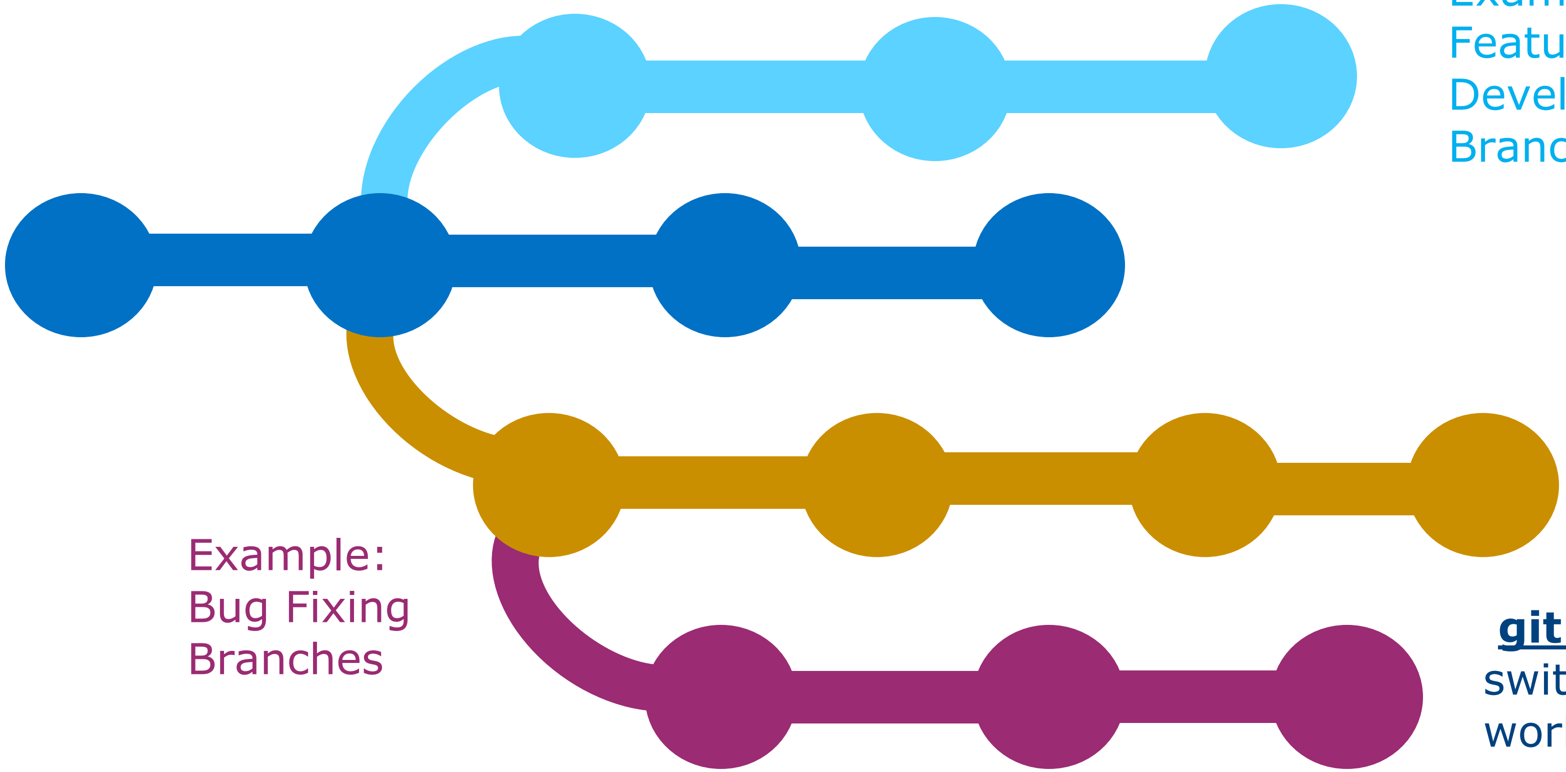
Get your new changes and puts on top of everyone's old stuff

git fetch: gets commits from remote repository and places in your local. Helps with accepting or declining before integrating



Branching: to allow for isolation of work and various versions of project

Master Branch: Location of latest releases or main project

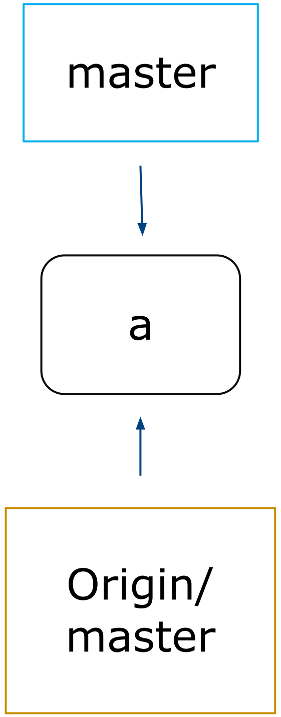


Example: Feature Development Branches

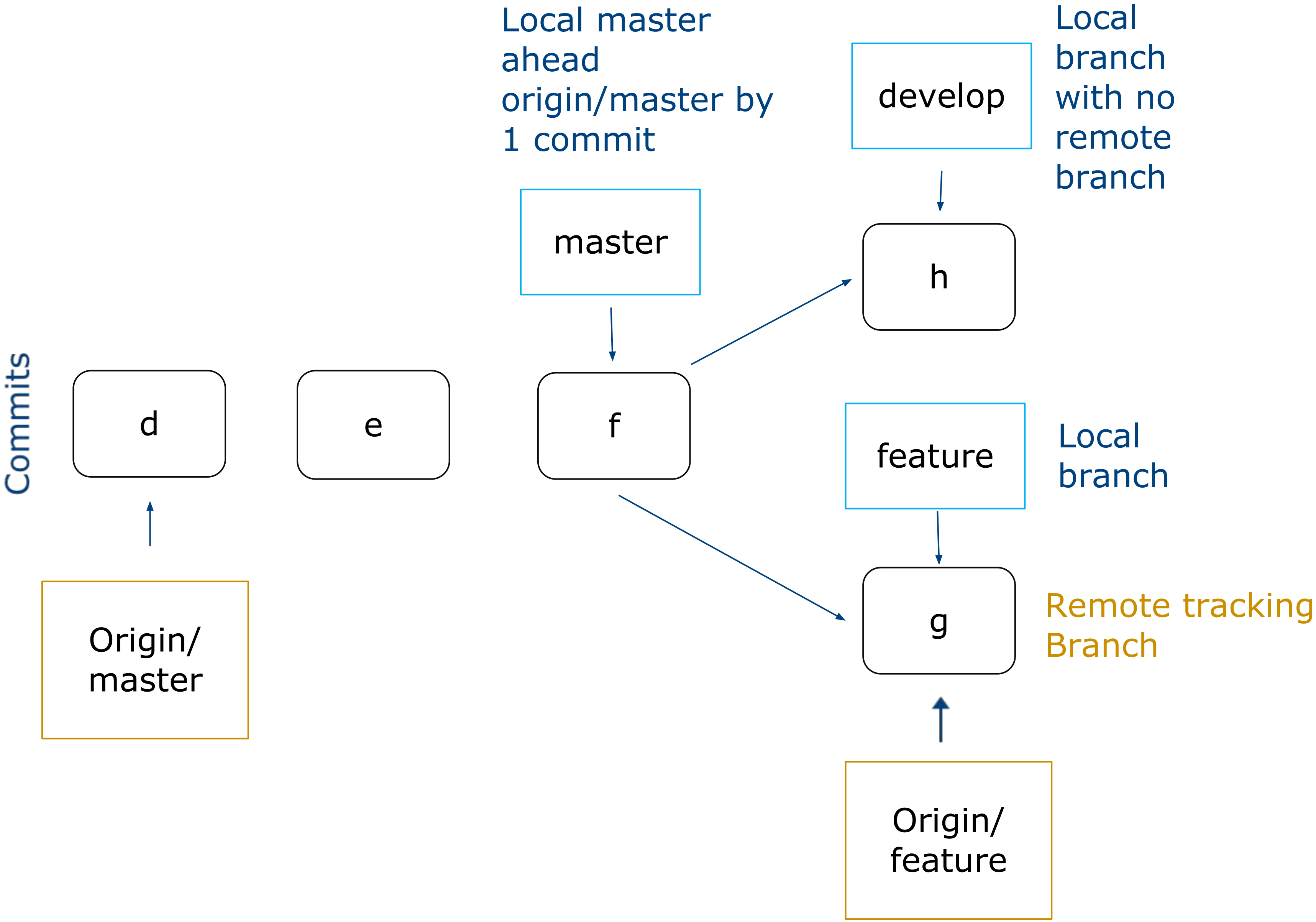
Example: Bug Fixing Branches

git checkout(svn switch): switches branches or restore working tree files top of everyone’s old stuff

Local: where your own system is located. Master is your local branch. Only you can see your local branches



Remote: branches that everyone can access. Some remote branches have a local counterpart that it is tracking. Origin/master is everyone’s master



Local master ahead origin/master by 1 commit

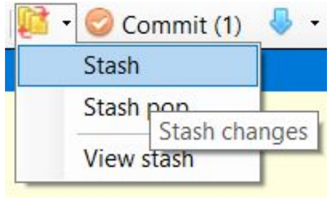
Local branch with no remote branch

Local branch

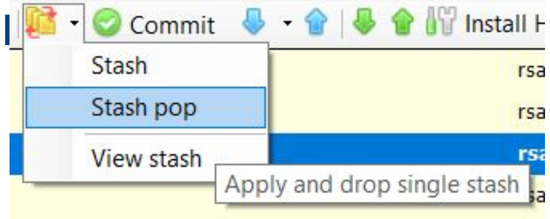
Remote tracking Branch

Stashing

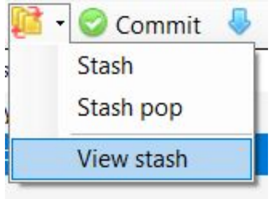
git stash(copy and pasting outside of SVN): When you want to checkout to another branch or commit but you don’t want to commit your modified work(Closing without saving changes)

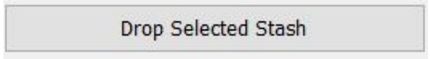


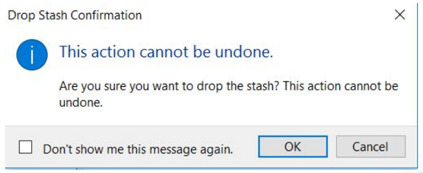
git stash pop(copy and pasting back into SVN): When you want to revisit the changes that are stored in your stash (Recover



git stash clear: After stashing your changes you decide everything you did was garbage and want to clear everything (Clearing the AutoRecover list) **CAN NOT BE UNDONE!**





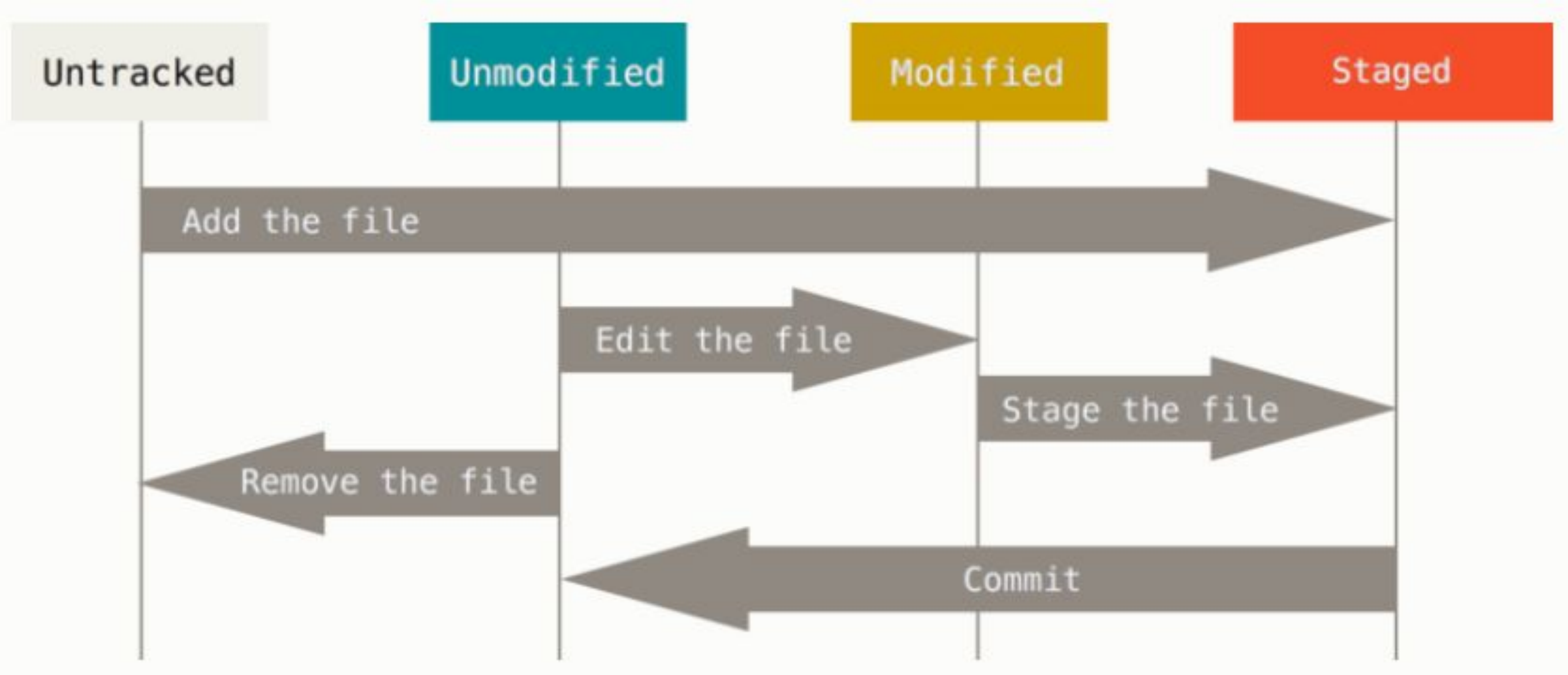

Drop Stash Confirmation

This action cannot be undone.

Are you sure you want to drop the stash? This action cannot be undone.

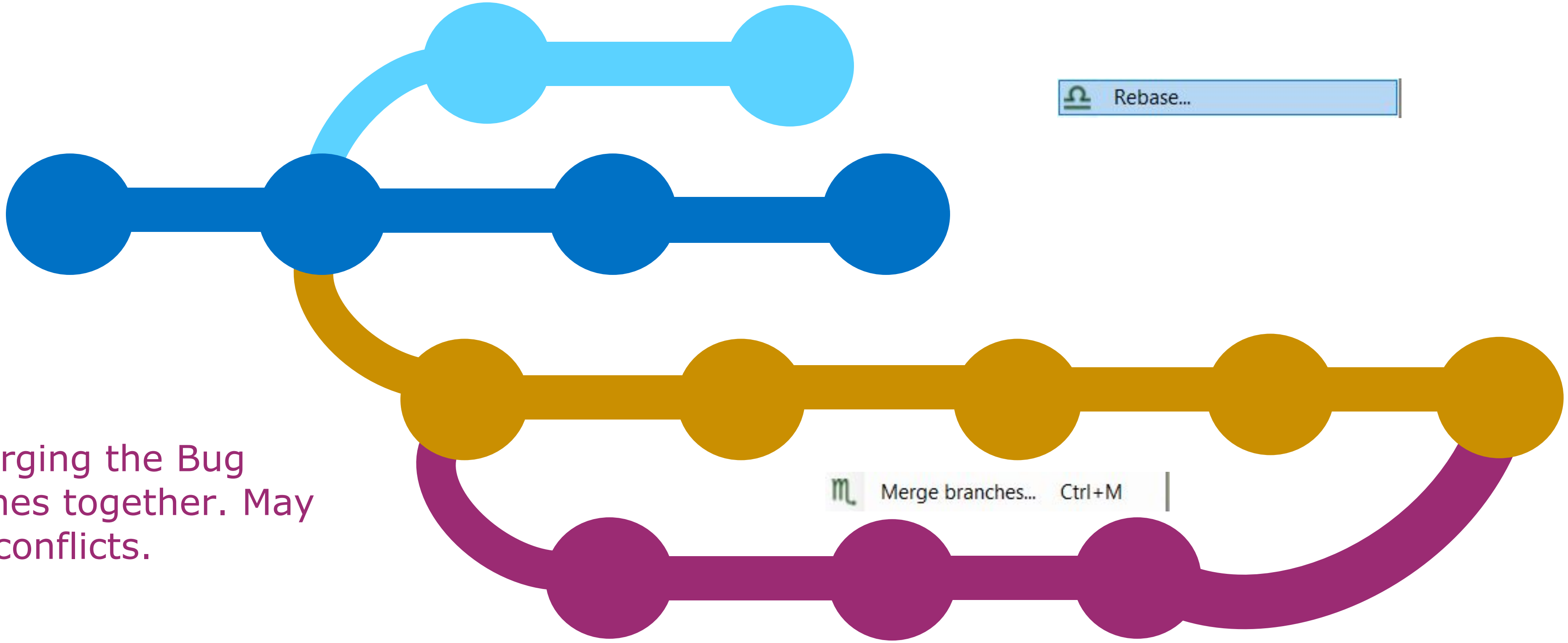
□ Don't show me this message again.

Staging and Tracking



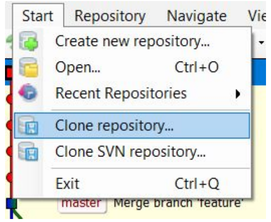
Merging and Rebasing

Example: Rebasing Feature branch into Master. Allowing singular line of commit history



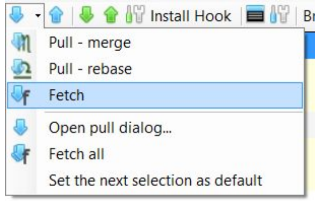
Example: Merging the Bug Fixing Branches together. May have merge conflicts.

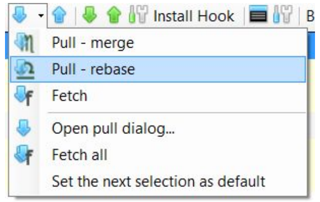
Centralized Workflow



Clone from remote

Make Commits

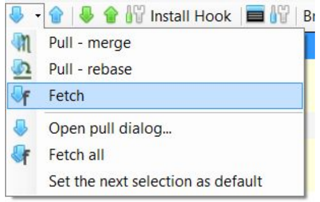


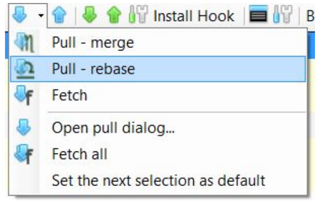


Fetch Pull --Rebase

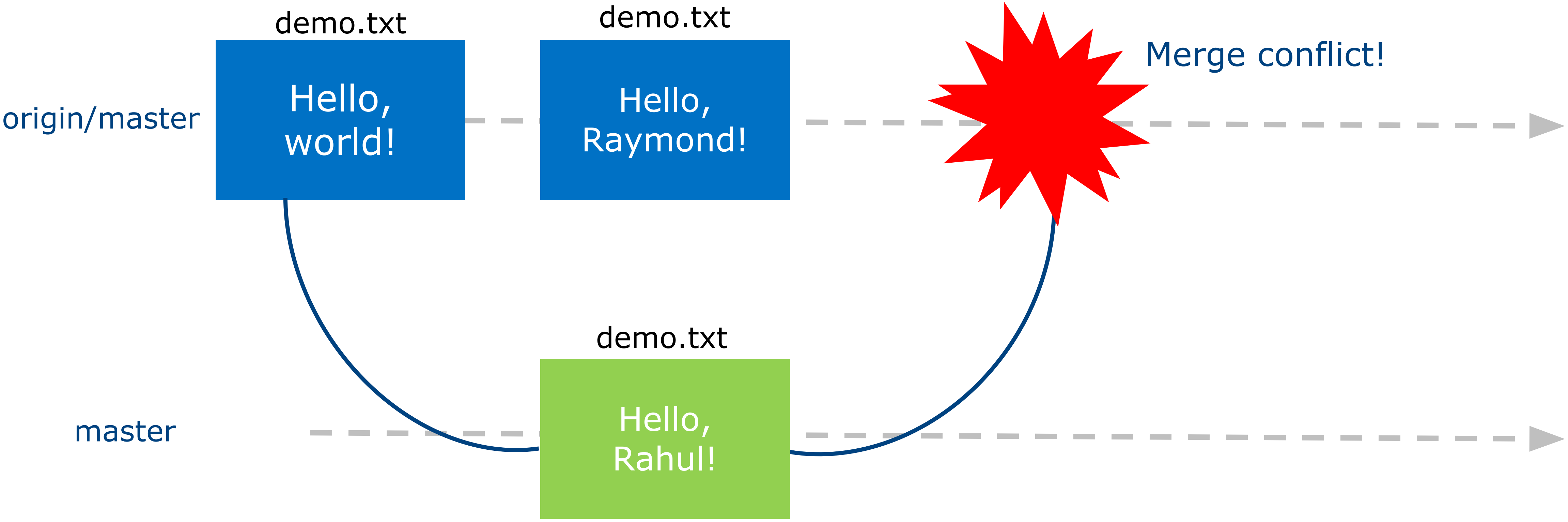
Push all changes to remote

Fetch Pull --Rebase





Merge Conflicts



DONE

Disclaimer: Practice on your own