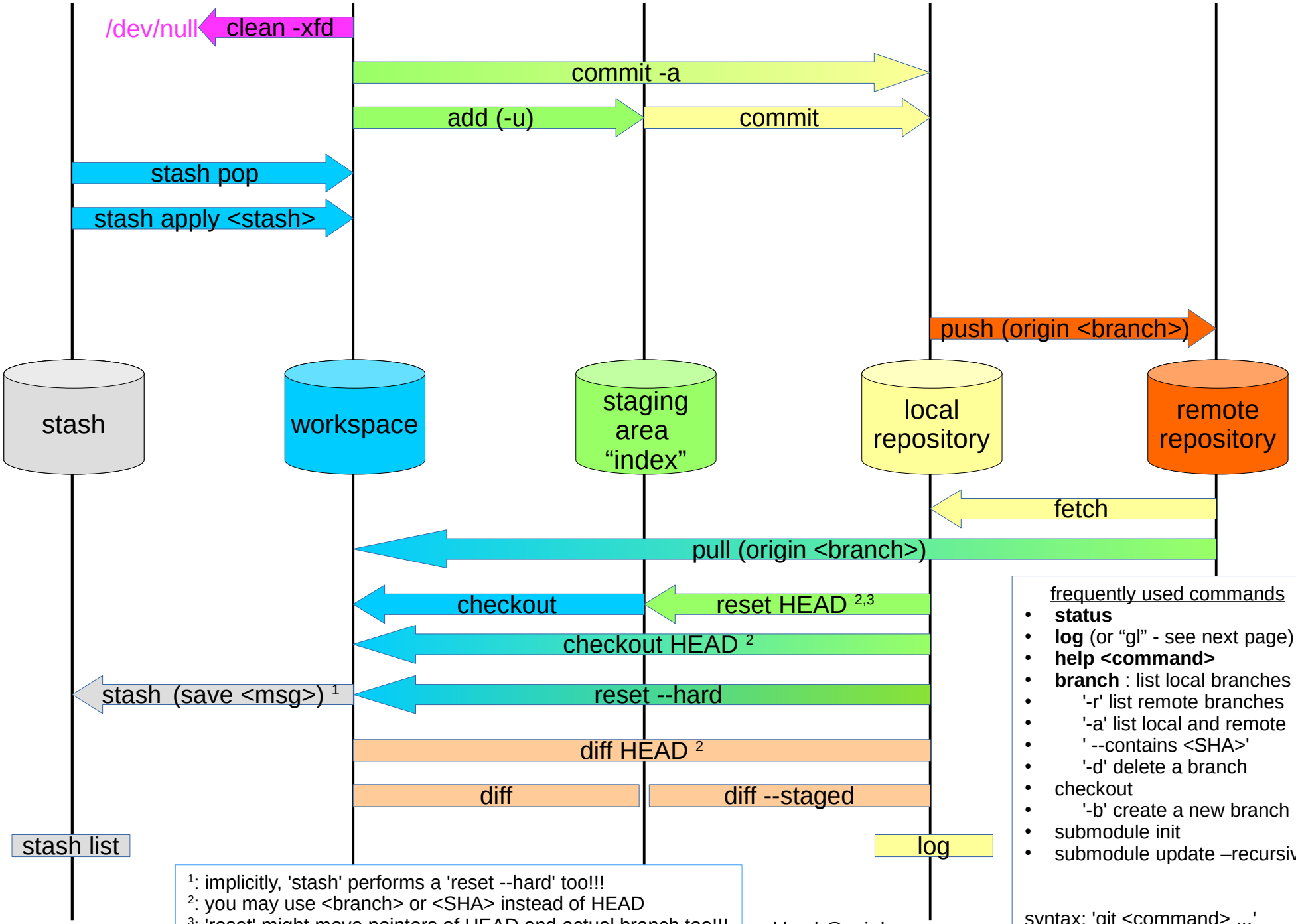

[1]: implicitly, 'stash' performs a 'reset --hard' too!!!
[2]: you may use <branch> or <SHA> instead of HEAD
[3]: 'reset' might move pointers of HEAD and actual branch too!!!

d.beck@gsi.de

frequently used commands

- **status**
- **log** (or "gl" - see next page)
- **help <command>**
- **branch** : list local branches
- '-r' list remote branches
- '-a' list local and remote
- ' --contains <SHA>'
- '-d' delete a branch
- checkout
- '-b' create a new branch
- submodule init
- submodule update –recursive

syntax: 'git <command> ...'

Fundamental

1. all operations act on where HEAD is pointing at
2. if the HEAD pointer is moved, the local copy is recalculated

- concept of Directed Acyclic Graph (DAG):
  <SHA>: each node has a unique hash
  pointer: a “node” depends on another node (pointer)
  direction: a “node” only depends on its previous node(s); it does/must not know its next node(s)
- special pointers are HEAD (see 1. and 2.) and <branch(name)>

Good to Know

- attached head state: HEAD → <branch> → <SHA> (result of “git checkout <branch>”)
- detached head stated: HEAD → <SHA> (result of “git checkout <SHA>”)
- variables/settings are defined recursively (system->user->repo); user: ~/.gitconfig
- possibility to define alias in ~/.gitconfig; example: “gl = log --graph --oneline --decorate –branches”. Usage: “**git gl**”

- option “--amend”: tell “commit” to modify last commit (message) by committing again (staged changes); bad style for commits already pushed
- option “--patch”: useful for many commands to step through multiple changes in a series of hunks
- option “-b”: tell “diff” to ignore changes with spaces
- option “--word-diff”: tell “diff” to improve visibility of changes
- option “origin <branch>”: tell push/fetch to use a specific *remote*; useful in case of multiple remote repositories (see cmd “remote”)

- cmd “merge <branch>” merges <branch> onto HEAD
- cmd “fsck” shows unreachable commits
- cmd “rebase --interactive <'stable' SHA>” useful for cleaning up (not yet published!) history
- cmd “remote” manages a set of tracked repositories; use verbose mode for useful listing

- operand “~” or “~n” addresses previous (n) predecessors; example “reset HEAD~”
- operand “^” or “^n” addresses previous (n) 'joint'; example “reset HEAD^”

Alex’ git merge course

‘don’t use merge’, consider using pull

- 1st : ‘git fetch’, get all the latest stuff
- 2nd: ‘git pull origin A’, pulls branch A into the local working branch
- alternative (DB)
- 1st: ‘git pull’, brute force, policy ‘pull.ff only’ preferred
- 2nd: ‘git pull . origin/A’, pulls branch A into the local working branch

in case of conflicts, try ‘merge’

- ‘git merge origin/A’, merges branch A into the local working branch

http://ndpsoftware.com/git-cheatsheet.html : another git cheat sheet
https://xkcd.com/1597/ : the ultimate git cheat sheet

This cheat sheet has been inspired by http://blog.osteele.com/posts/2008/05/my-git-workflow