

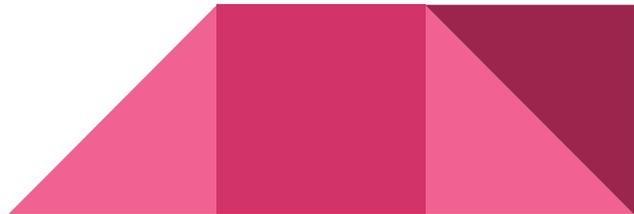
Jenkins hints for large git repos

When you have to do what you don't want to do...

Mark Waite - Jenkins World 2016
14 Sep 2016

Large git repos are usually unhealthy

- Large git repos (2 GB+) often are a sign of git misuse (large binaries, etc.)
- Yes, there are cases where a large git repo is not a sign of misuse
 - Linux kernel repo is over 1 GB
 - Linux kernel repo is one of the oldest git repos
 - Git is able to handle large repos, some operations are slower on some systems
- Git misuse doesn't matter - we need to run our Jenkins jobs anyway
 - The 18 GB+ repository at my work isn't pretty, but it is business critical...



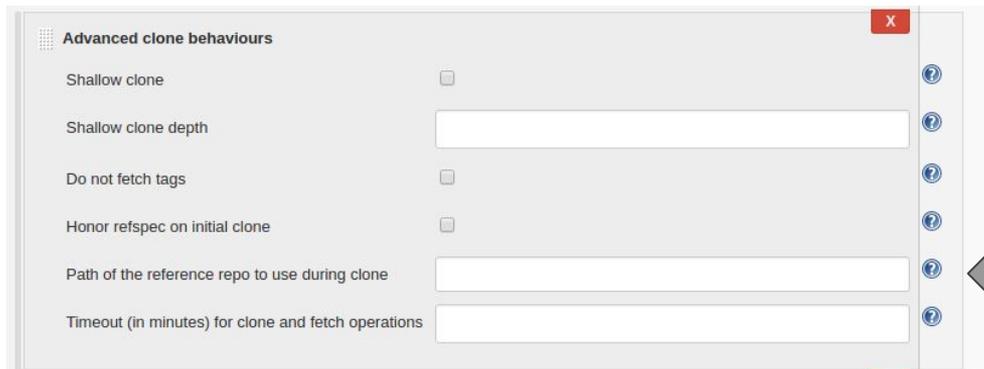
Jenkins tricks for large repos

- Use command line git rather than JGit
 - Command line git explicitly manages memory and handles large repos
 - JGit large repo support has improved, will likely always lag behind command line git
- Reduce data transfer & disc space for history
 - Reference repositories - only clone what's new
 - Shallow clone - only clone recent history
 - Don't fetch tags
 - Narrow refspecs - only clone specific branches
- Reduce disc space for working directory
 - Sparse checkout



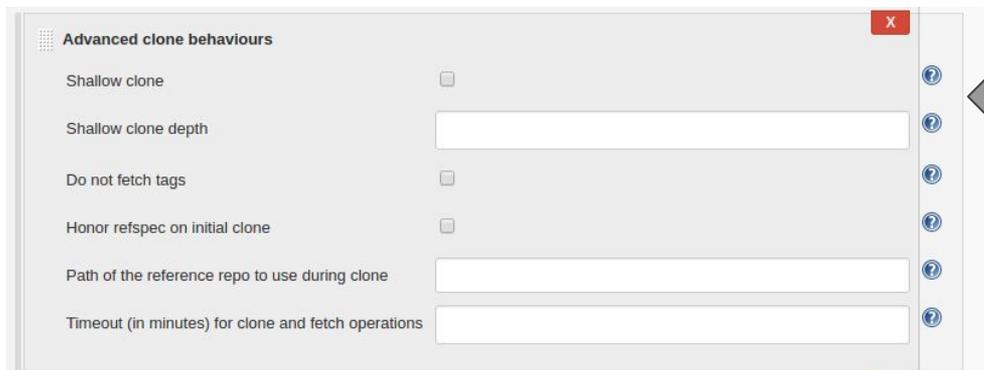
Reference repository - pointers, not copies

- A bare git repository can be used as a “reference repository” for other repositories
- Git creates pointers to the reference repo
 - Saves space by not copying data to disc
 - Saves time by not transferring over the wire



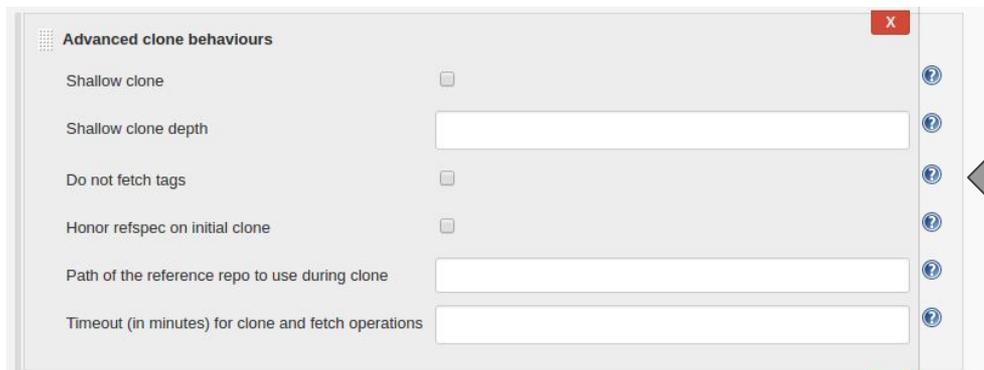
Shallow clone - only clone recent commits

- Clones only “n” most recent commits
 - Sacrifices completeness of “recent changes” to reduce data transfer and disc use
- Git 1.9.0+ can push from a shallow clone



Dont fetch tags

- Most Jenkins jobs don't need tags
- Skipping tags can reduce data transfer and local disc use



Narrow refspecs - clone only what you need

- Git refspeg describes what to copy from remote
- Reducing the refspeg can reduce the data transfer and disc use in the local copy of the git history

Advanced clone behaviours X

Shallow clone

Shallow clone depth

Do not fetch tags

Honor refspeg on initial clone

Path of the reference repo to use during clone

Timeout (in minutes) for clone and fetch operations

Repository URL

Credentials

Name

Refspec



Sparse checkout - work with only what you need

- Sparse checkout reduces working directory size
- Exclude irrelevant directories and files
 - Large binaries in a subtree you don't need
 - Deep directory structures you don't need
 - Etc.



Submodule Authentication - JENKINS-20941

- Most requested feature of the git plugin
 - 100+ watchers of JENKINS-20941
 - 1 year in beta test - special thanks to Jacob Keller for his patience
 - Changes the git client plugin authentication technique
- Submodules allow git repositories nested in other git repositories
- Git plugin 3.0 and git client plugin 2.0 support submodule authentication
- Submodule authentication requires:
 - Same protocol (https or ssh) for repository and all its submodules
 - Same credentials used for repository and all its submodules

