Git Installation

Git is an open source, distributed version control system designed for speed and efficiency

Git Characteristics

- Strong support for non-linear development
    - Rapid branching and merging, and includes specific tools for visualizing and navigating a non-linear development history.
    - Branches in git are very lightweight: A branch in git is only a reference to a single commit. With its parental commits, the full branch structure can be constructed.

Git Characteristics

- Distributed development
  - Like Darcs, BitKeeper, Mercurial, SVK, Bazaar and Monotone, Git gives each developer a local copy of the entire development history, and changes are copied from one such repository to another. These changes are imported as additional development branches, and can be merged in the same way as a locally developed branch.

- Compatibility with existing systems/protocols
  - Repositories can be published via HTTP, FTP, rsync, or a Git protocol over either a plain socket, or ssh. Git also has a CVS server emulation, which enables the use of existing CVS clients and IDE plugins to access Git repositories. Subversion and svk repositories can be used directly with git-svn.

Git Characteristics

- Efficient handling of large projects
- Cryptographic authentication of history
- Toolkit-based design
- Pluggable merge strategies
- Garbage accumulates unless collected
- Periodic explicit object packing

Git Server

- As git is a distributed version control system, it can be used as server out of the box. Dedicated git server software helps, amongst other features, to add access control, display the contents of a git repository via web, and help managing multiple repositories.
- Remote file store and shell access
- Gitdaemon, instaweb
- Gitolite
- Gerrit

Git Server

- Gitblit
- Gitiles
- Bonobo GitServer
- Commercial solutions

Git : Benefits

- Distributed : local & remote repositories
- No Network Needed for
  - Performing Diff
  - Viewing file history
  - Committing changes
  - Merging branches
  - Switching branches
- Immutable
- Snapshots over patches

Git Basics

Commit

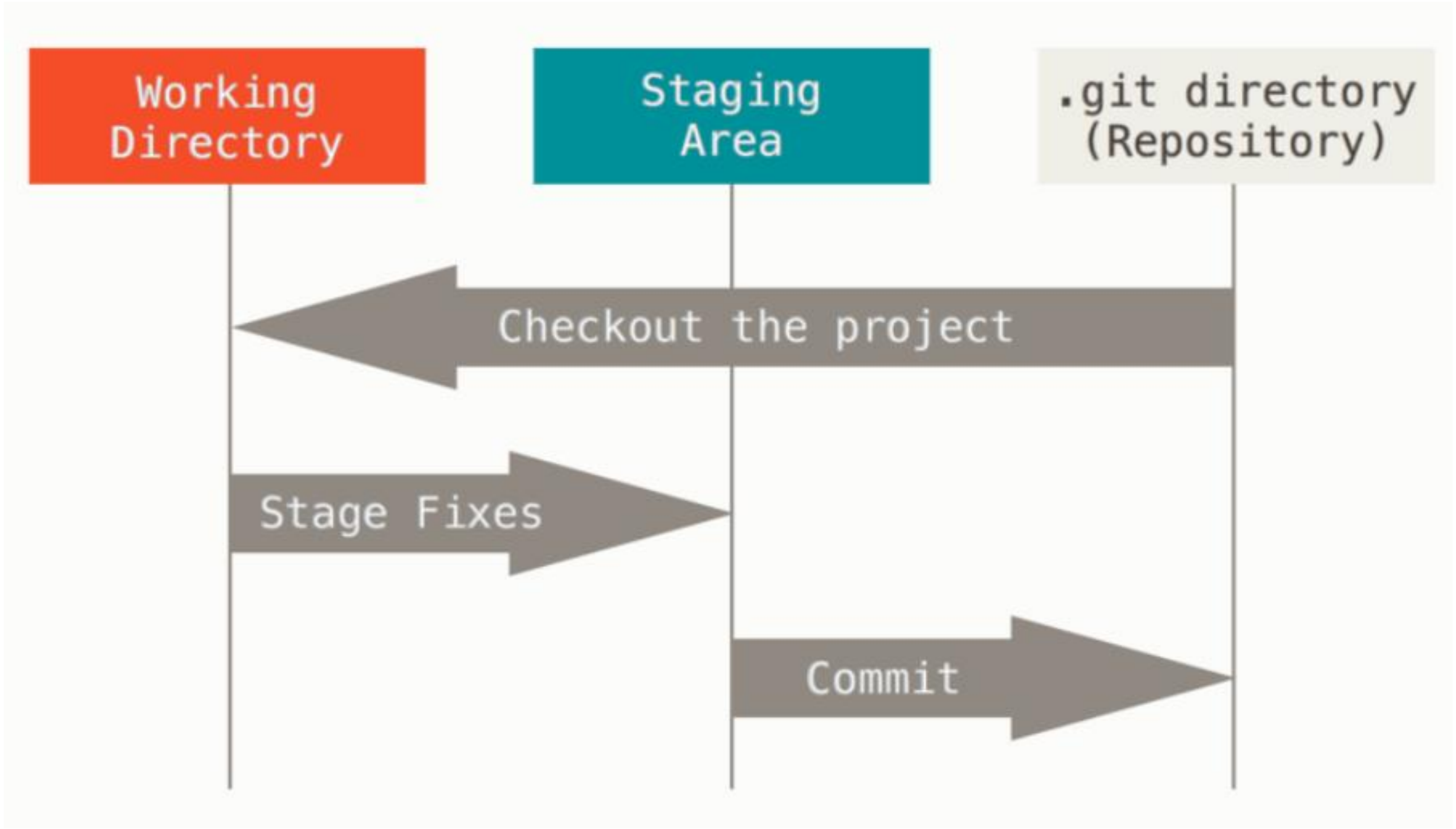
Branch

Tags

Local Repository

Remote Repository

Git : Workspace



Git : Installation

- Available on Linux
  - sudo apt-get install git
- Available on MacOS
- Available on Windows
  - Git for Windows (Installer)
  - Portable Git
  - Both Versions available at https://git-scm.com/download/win

Git : One time setup

- Being a distributed version control system, need to identify current user through config:

```
$ git config --global user.name “user”
$ git config --global user.email “email”
```

Repositories

- Repositories:
- It is a collection of refs together with an object database containing all objects which are reachable from the refs, possibly accompanied by meta data from one or more porcelains.
- A repository can share an object database with other repositories via alternates mechanism.
- What to store in repositories?
    - Anything, however any sort of editable files are preferred.
- How to get GIT repository:
    - $ git clone git://git.kernel.org/pub/scm/git/git.git
- It does approx. 225 MB download.

Git Working with Local Repository

- Create a new empty repository

```
$ git init
```

- Add files

Create files using favourite editor and save in current directory

- Stage / Index files

```
$ git add .
$ git staus
```

- Commit changes

```
$ git commit -m "Initial change!"
```

Understanding History - Commits

- Every change in the history of a project is represented by a commit. The git-show(1) command shows the most recent commit on the current branch:

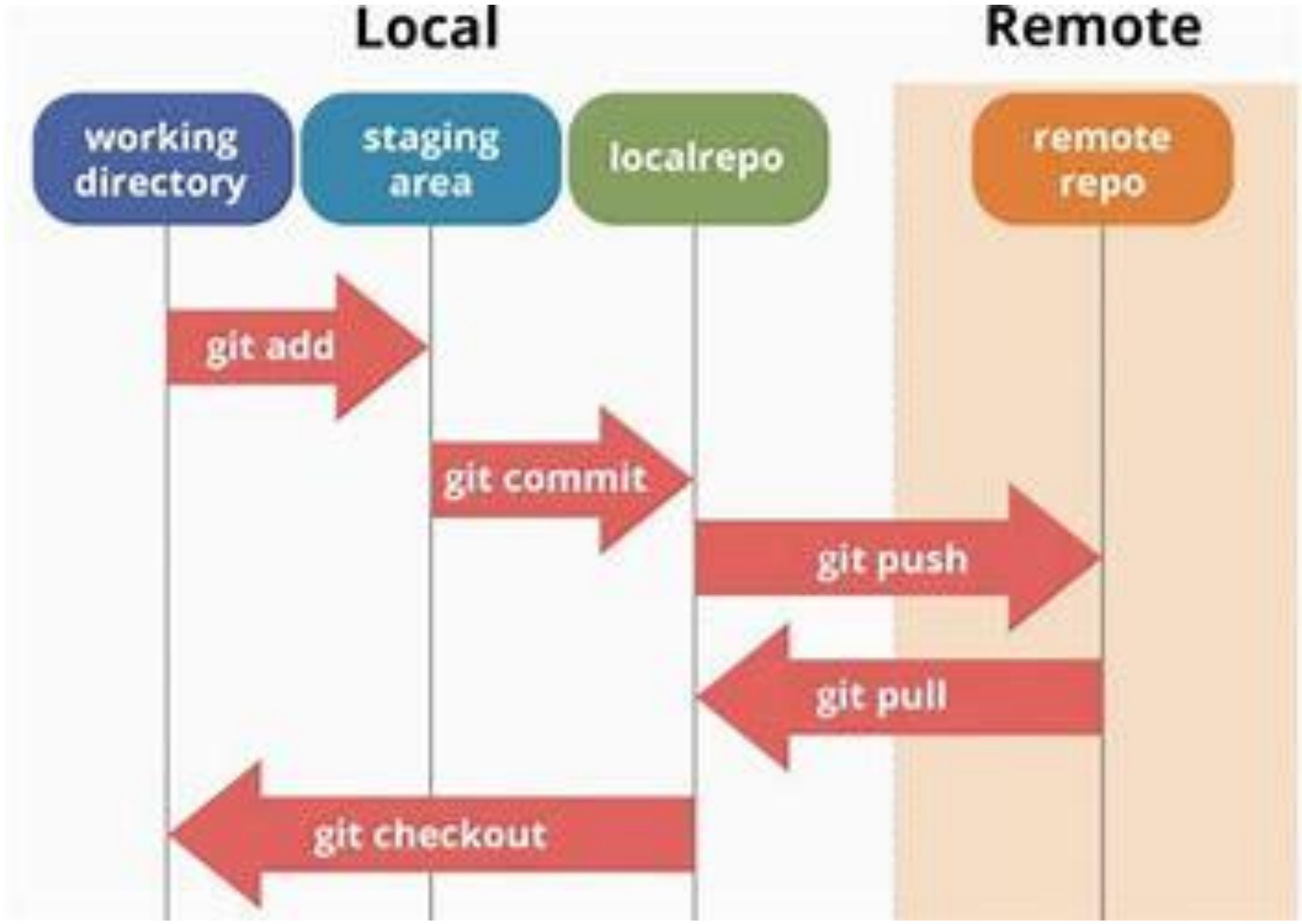
```
$ git show
```

- Every commit (except the very first commit in a project) also has a parent commit which shows what happened before this commit. Following the chain of parents will eventually take you back to the beginning of the project.

Tracking Changes

- git diff
- git commit –a –m “commit multiple files”

Git File Workflow



Thank you