


git

# Git: A Deep Dive Under the Surface

Matthew Heironimus

# Life Before Git

# Life Before Git

- SourceSafe
- ClearCase
- Subversion
- Team Foundation Server (Azure DevOps)

Git Bankruptcy

# Why a Deep Dive into Git?

# Brief History of Git

- Linux Kernel Development - BitKeeper
- Started by Linus Torvalds
- 12/21/2005 - 1.0 Released by Junio Hamano

# Git Goals

- Speed
- Simple design
- Strong support for non-linear development (thousands of parallel branches)
- Fully distributed
- Able to handle large projects like the Linux kernel efficiently (speed and data size)

# How is Git Different?

# How is Git Different?

- Local
- Snapshots, Not Differences
- Checksums (SHA-1)

# How is Git Different?

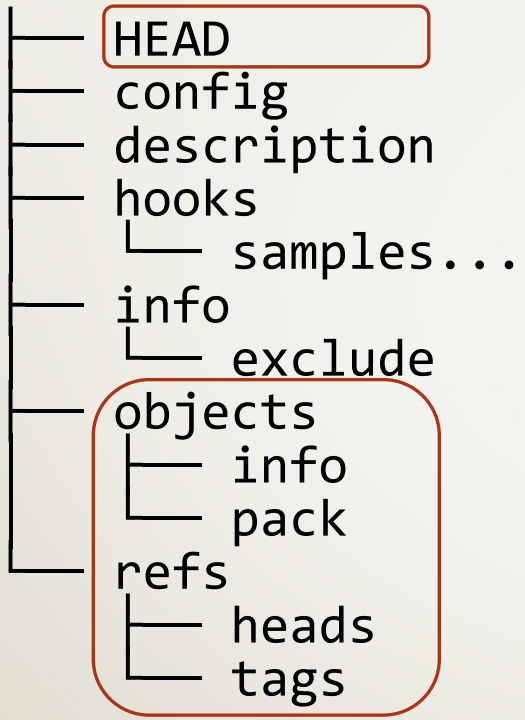
- File States in Git
  - Modified: file changed in “Working Directory”
  - Staged (`git add`): file selected for next commit
  - Committed (`git commit`): snapshot committed
  - Untracked: file not currently tracked by Git

# What is Git?

# `git init`

Creating a new Git repository

# .git Folder


HEAD
config
description
hooks
samples...
info
exclude
objects
info
pack
refs
heads
tags

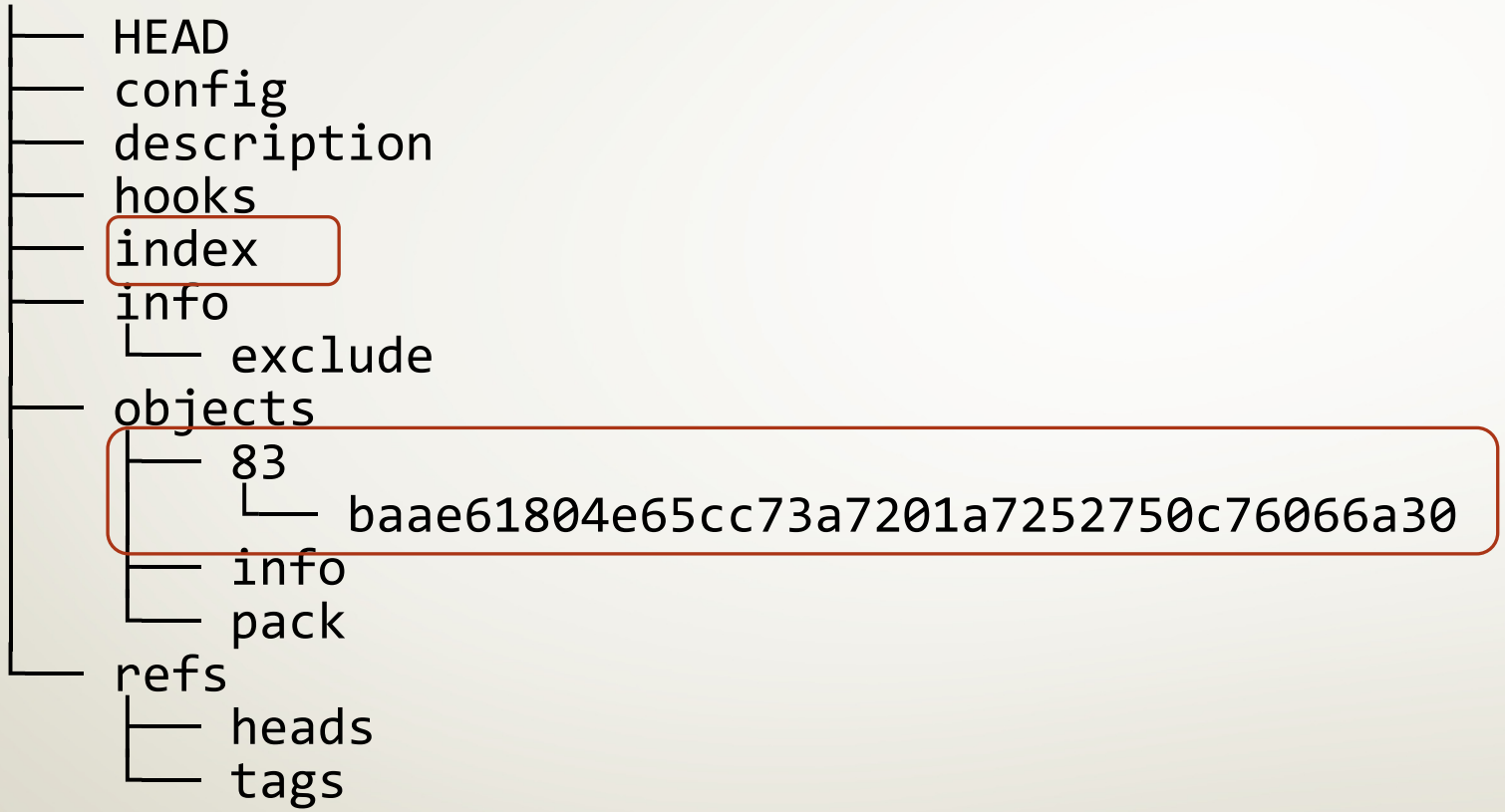
# git config

Customizing the git configuration

# `git add`

Adding a file to the Staging Area (index)

# .git Folder After "git add"



# `git ls-files`

List files in the Staging Area (index) and the Working Directory

# `git ls-files`

- -s, --stage : Include file modes and SHA-1 hash
- -o, --others : Show untracked files

# `git cat-file`

Provides content or type and size information for repository objects

# `git cat-file`

- -p : Pretty-print the contents
- -t : Show object type

# `git hash-object`

Compute object ID and optionally create a blob from a file

# What is Git?

- Content-addressable Filesystem
  - Unique Key: SHA-1 (or SHA-256)
  - Zlib Compression

# Disclaimers / Clarifications

- File contents are prepended with an object type identifier, followed by a null character

# `git hash-object`

- `-w` : Write object into the object database
- `-t` : Specify the object type (default: "blob")

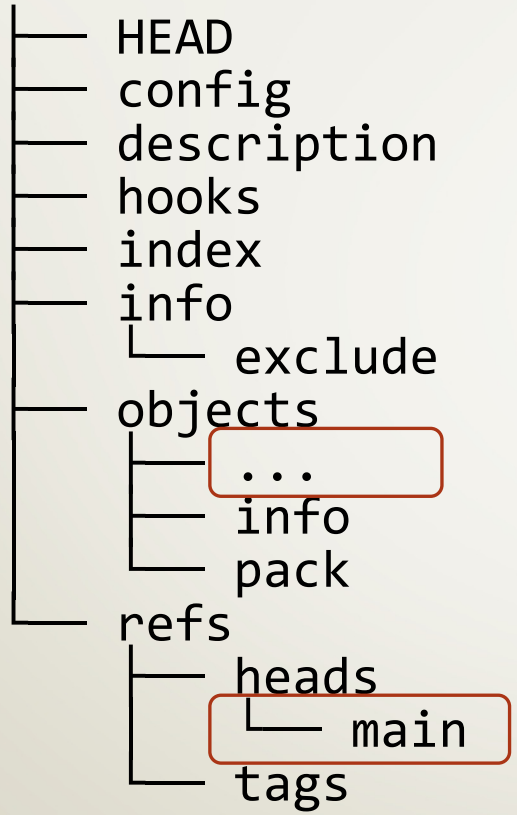
# How is Git Different?

- File States in Git
  - Modified: file changed in “Working Directory”
  - Staged (`git add`): file selected for next commit
  - Committed (`git commit`): snapshot committed
  - Untracked: file not currently tracked by Git

# `git commit`

Record changes to the repository

# .git Folder After "git commit"



# Anatomy of a Commit Object

```
tree 00b3fcb35c85f7a97a96a552ce00259c13128166
author Matthew Heironimus <me@email.com> 1688420897 -0500
committer Matthew Heironimus <me@email.com> 1688420897 -0500

My first commit
```

# Anatomy of a Tree Object

```
100644 blob 83baae61804e65cc73a7201a7252750c76066a30   another.txt
100644 blob 1f7a7a472abf3dd9643fd615f6da379c4acb3e3a   demo.txt
040000 tree 419603a4e13b5db3b350ed9bdf89aa161a0efff7   subfolder
```

Mode:

100644 – non-executable file
100755 – executable file
040000 - directory
120000 – symbolic link
…

# Anatomy of a Tree Object

```
100644 blob 83baae61804e65cc73a7201a7252750c76066a30   another.txt
100644 blob 1f7a7a472abf3dd9643fd615f6da379c4acb3e3a   demo.txt
040000 tree 419603a4e13b5db3b350ed9bdf89aa161a0efff7   subfolder
```

Object Type

# Anatomy of a Tree Object

```
100644 blob 83baae61804e65cc73a7201a7252750c76066a30   another.txt
100644 blob 1f7a7a472abf3dd9643fd615f6da379c4acb3e3a   demo.txt
040000 tree 419603a4e13b5db3b350ed9bdf89aa161a0efff7   subfolder
```

SHA-1 Hash

# Anatomy of a Tree Object

```
100644 blob 83baae61804e65cc73a7201a7252750c76066a30   another.txt
100644 blob 1f7a7a472abf3dd9643fd615f6da379c4acb3e3a   demo.txt
040000 tree 419603a4e13b5db3b350ed9bdf89aa161a0efff7   subfolder
```

Name

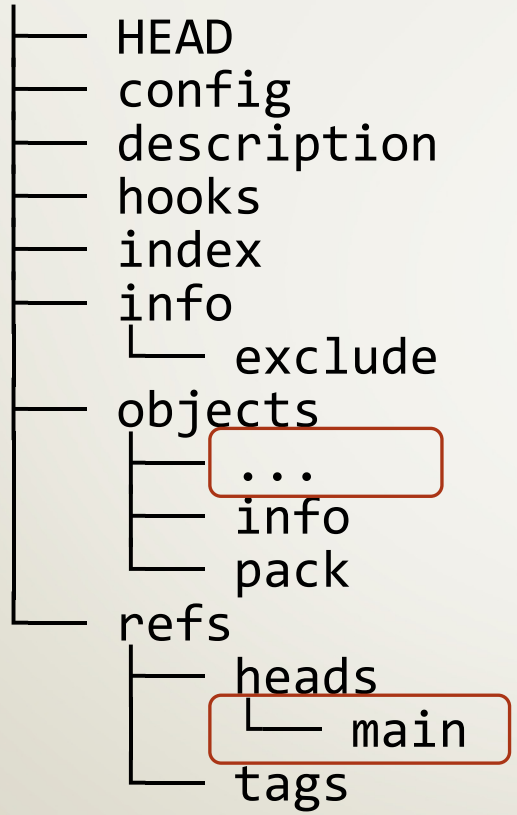
# `git ls-tree`

List the contents of a tree object

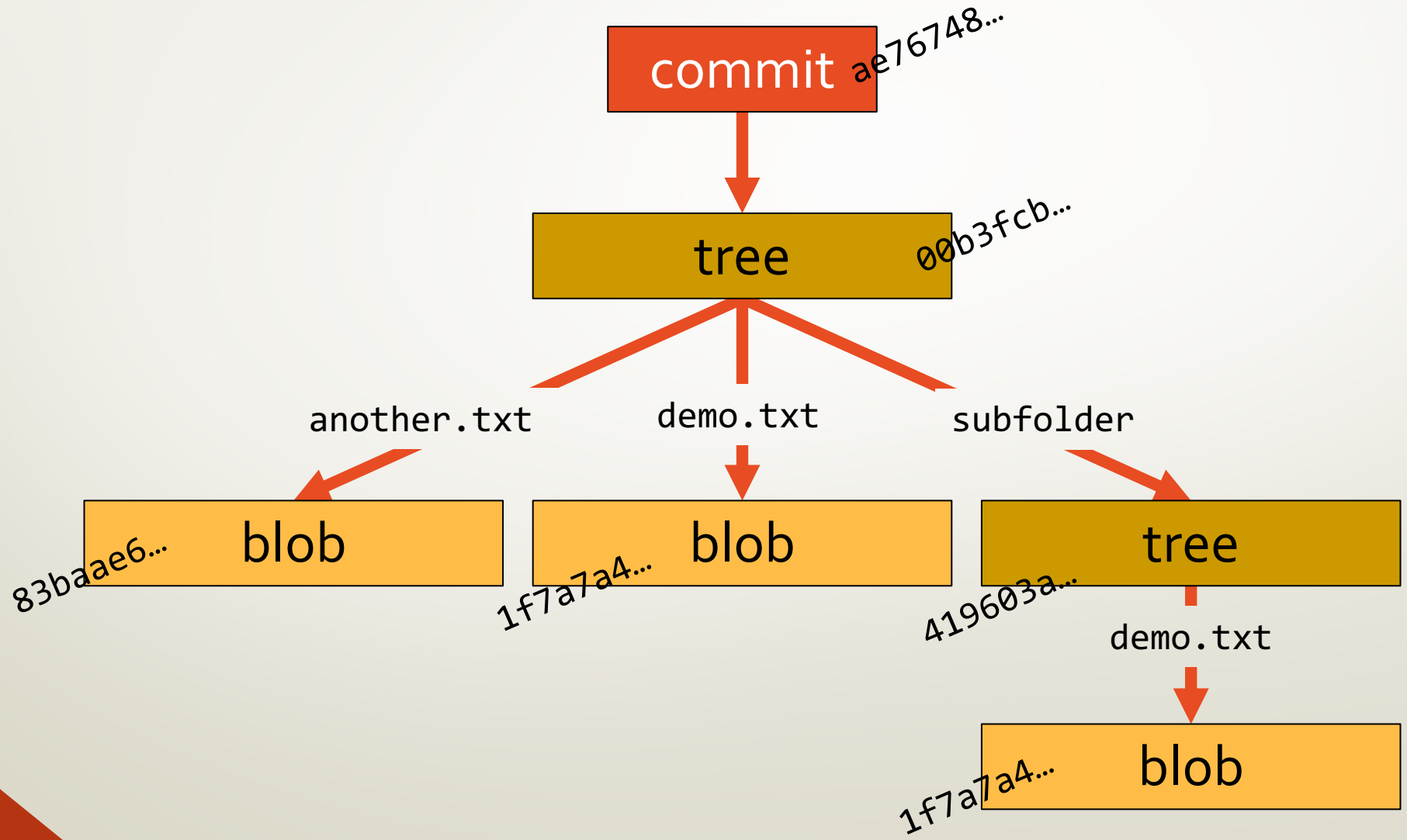
# `git ls-tree`

- `-r` : Recurse into sub-trees
- `-t` : Show tree entries when recursing sub-trees

# .git Folder After "git commit"



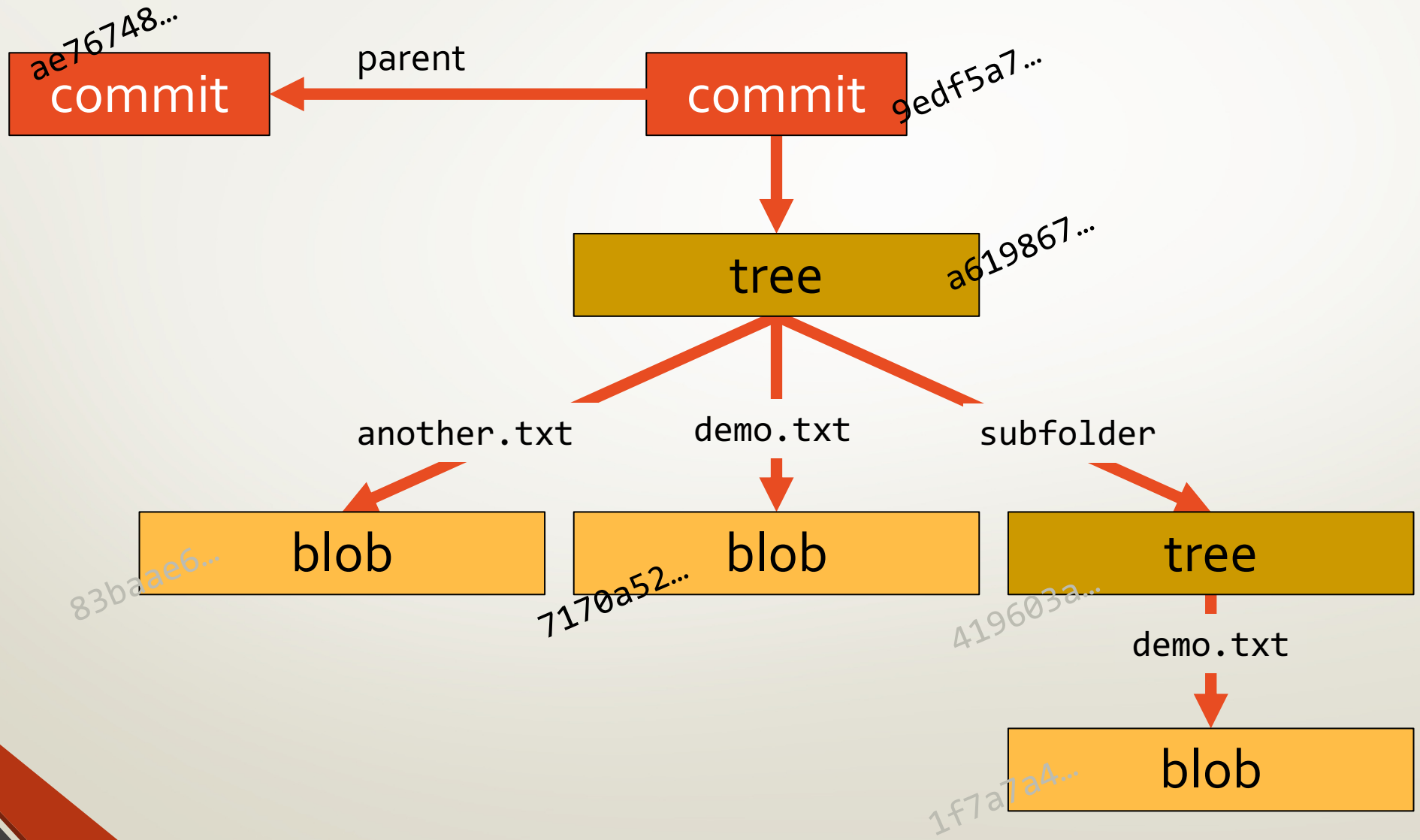
# First Commit



# Anatomy of a Commit Object

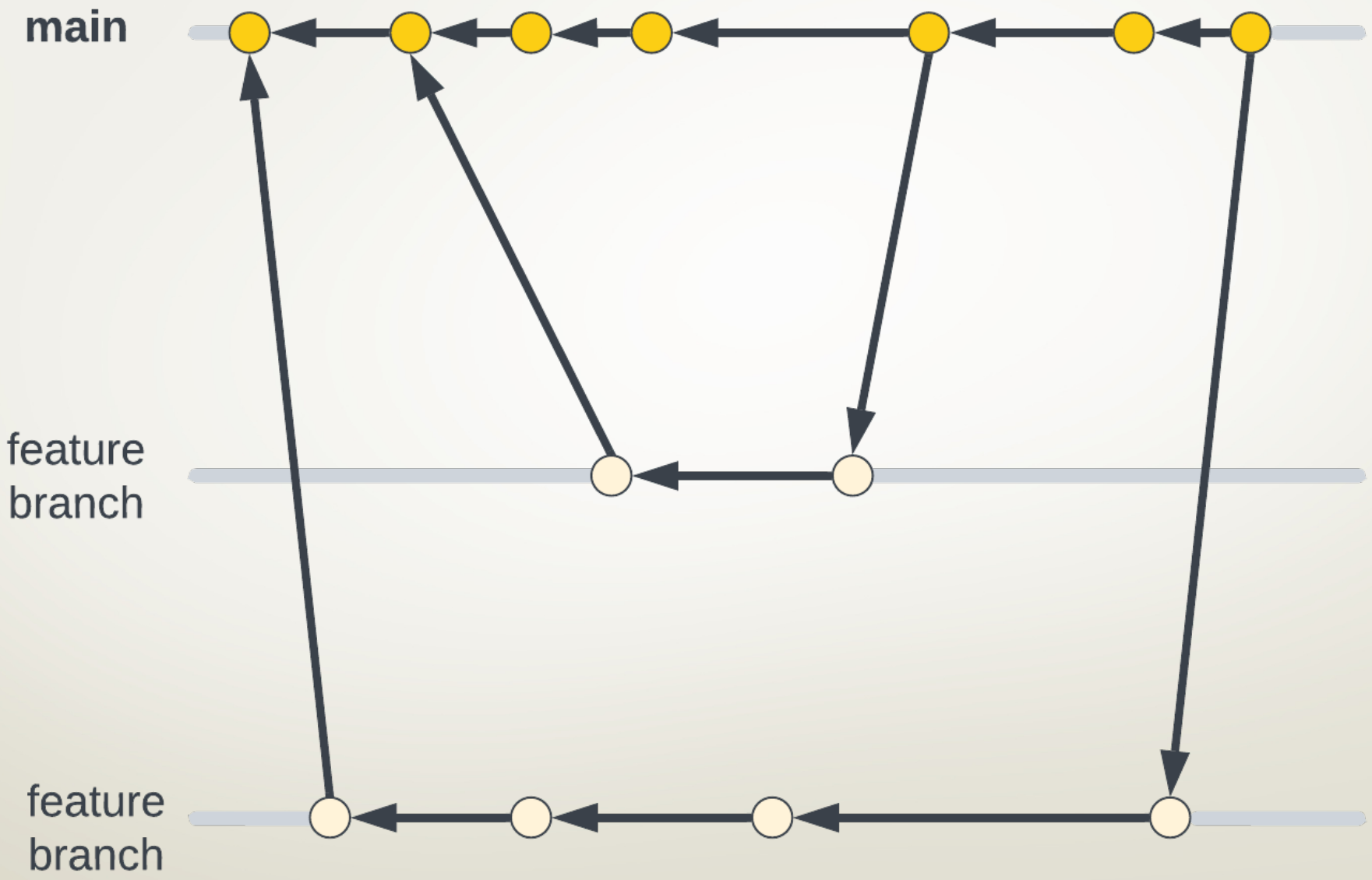
```
tree a6198679e0261b0410ecfb0cc0cb073a1c7888d8
parent ae76748d65f2edaf34c640a671969c40cda7027c
author Matthew Heironimus <me@email.com> 1688762726 -0500
committer Matthew Heironimus <me@email.com> 1688762726 -0500

My second commit
```

# Second Commit



# What is Git?

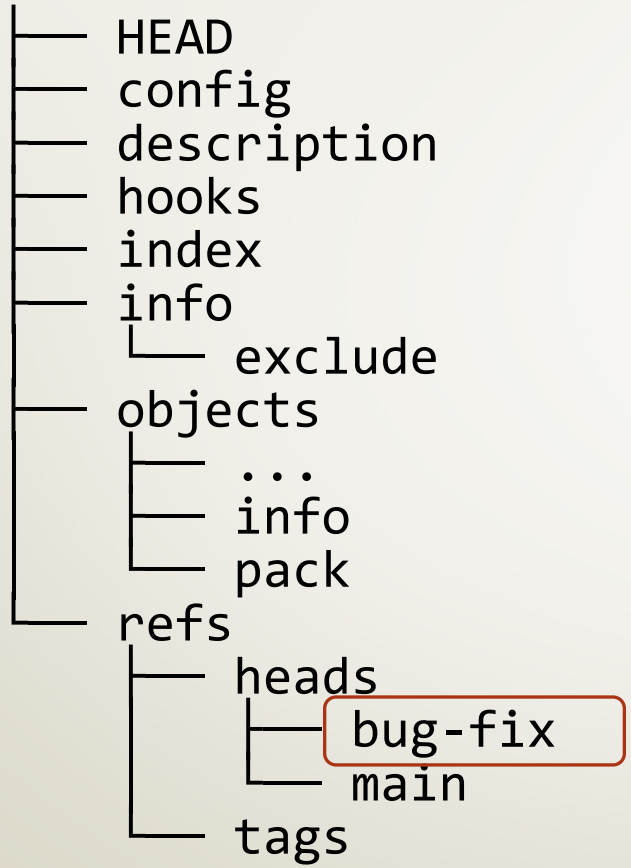
- Content-addressable Filesystem
  - Unique Key: SHA-1
  - Zlib Compression
- Series of Snapshots
  - Each commit is a snapshot
- Directed Acyclic Graph (DAG)


main
feature branch
feature branch

# `git branch`

List, create, or delete branches

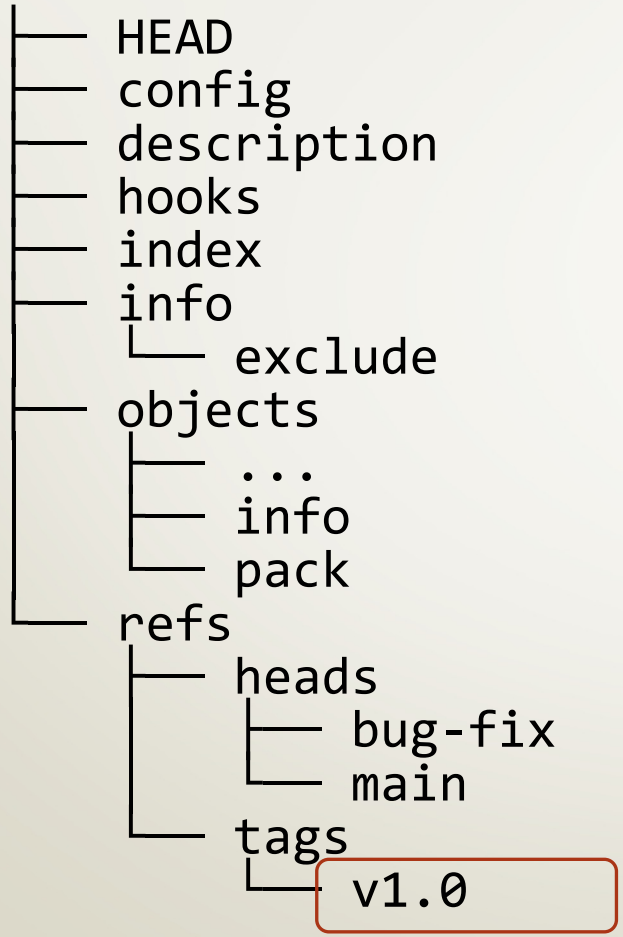
# .git Folder After "git branch"



# `git tag`

Create, list, or delete a tag object

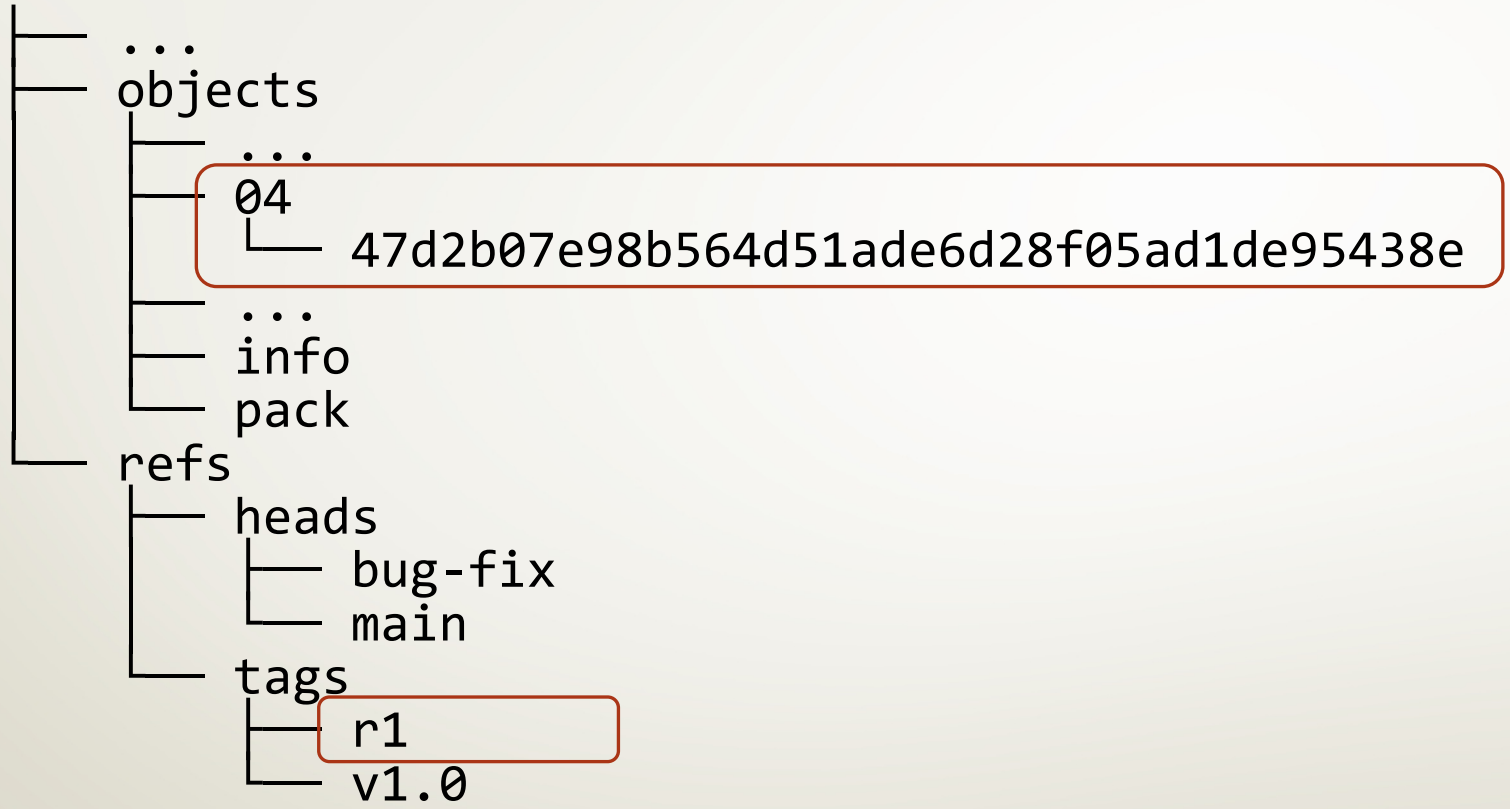
# .git Folder After “git tag”



# `git tag`

- Lightweight Tags
- Annotated Tags

# .git Folder After "git tag -m"



# Anatomy of an Annotated Tag Object

```
object d834426d708532320626b14b54401f108cd1a007
type commit
tag r1
tagger Matthew Heironimus <me@email.com> 1691517144 -0500

Release 1 - Initial Release
```

# Git Object Types

- blob – file contents
- commit - point-in-time snapshot of the repository
- tree – names of blobs in a directory
- tag – annotated tag

# HEAD

Last commit snapshot, next parent

# `git log`

Show commit log

# `git log`

- `-<num>` : Limits the number of commits shown
- `--all` : Includes all refs and HEAD
- `--oneline` : Only one line per commit
- `--graph` : Text-based commit graph
- `--show-signature` : Show commit signatures

# `git switch`

Switch branches

# `git reset`

Reset current HEAD to the specified state

# `git reset` - Modes

- `--soft` : Resets the working tree but not the index
- `--mixed` : Resets the index but not the working tree
- `--hard` : Resets the index and working tree

# `git reflog [show]`

Shows reference log (reflog) information

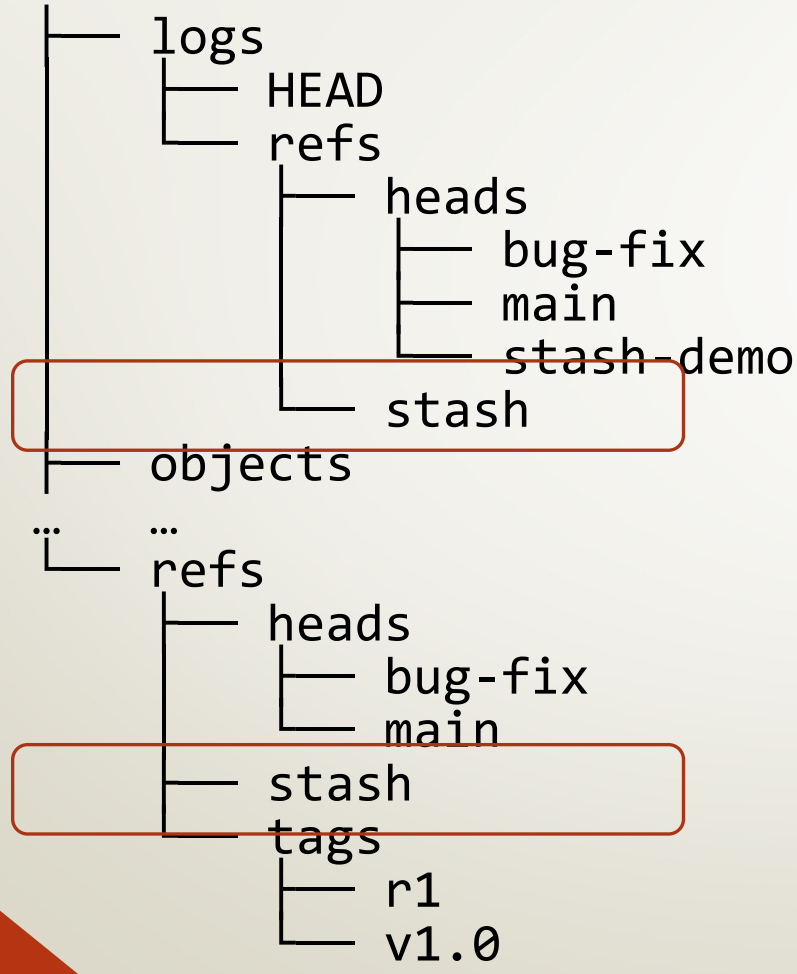
# `git rebase`

Reapply commits on top of another base tip

# `git stash`

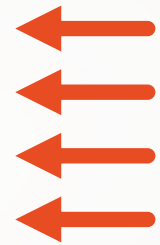
Stash the changes in a dirty working directory away

# .git Folder After "git stash push"



# Anatomy of a "Stash" Commit Object

```
tree acaf3d5962b04cda7eb95f5b1b7cd1213e885edf
parent 09292bc823190126c541a8c8568fc3c3cbf30c4e
parent 68d18f5dbb4768b5db7723c111e0eda0832cc132
parent a7e826e140674bd91a65734f304edb3ed4fe5253
author Matthew Heironimus <heironimus@live.com> 1723410442 -0500
committer Matthew Heironimus <heironimus@live.com> 1723410442 -0500

On bug-fix: My First Stash
```



Working Tree
Parent Commit / Branch HEAD
index Snapshot
Untracked Files Snapshot

Branch

Stash Name / Comment

# `git stash commands`

- `push` : Saves changes to a new stash entry and rolls the working  directory back to HEAD
- `list` : List stash entries
- `pop` : Removes a stash entry from the list and applies it to the current working directory
- `pop --index` : Reinstates both the working directory and the index

# `git stash commands`

- `apply` : Same as pop, but without removing the stash entry from the list
- `clear` : Remove all the stash entries.

# `git merge`

Join two or more development histories together

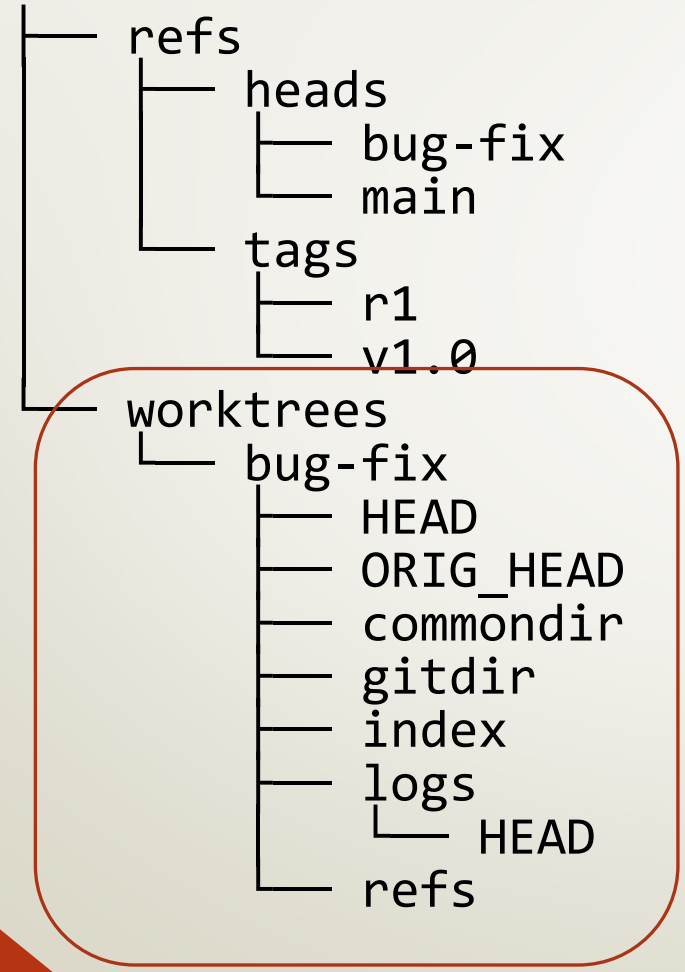
# `git merge` Types

- `--ff` : Fast-forward – No merge commit created
- `--commit` : Merge and commit the results
- `--squash` : Performs merge without committing

# `git worktree`

Manage multiple working trees attached to the same repository

# .git Folder After "git worktree add"



# git commit Tricks

Fun git commit tricks

## Different Author

- `--author=<author>`

## Different Date

- `--date=<date>`

## Signed-off-by

- `-s`

## Trailers

- `--trailer="<token>:<value>"`
- See interpret-trailers for more details

# Signed Commits

- Commits can be signed with a GPG or SSH key
- Only someone with the private key can generate the signature
- `git commit -S`
- `git log --show-signature`

# Signed Tags

# `git tag`

- Lightweight Tags
- Annotated Tags (`-m`)
- Signed Tags (`-s`)

# Experimental Features

# SHA-256

```
git init --object-format=sha256
```

# Recap

# What is Git?

- Content-addressable Filesystem
  - Unique Key: SHA-1
  - Zlib Compression
- Series of Snapshots
  - Each commit is a snapshot
- Directed Acyclic Graph (DAG)

# References


- Git - https://git-scm.com/
- Git Source Code Mirror - https://github.com/git/git
- SHA-1 - https://en.wikipedia.org/wiki/SHA-1
- Pro Git - https://git-scm.com/book/en/v2
- Head First Git - https://www.amazon.com/Head-First-Git-Learners-Understanding/dp/1492092517

# Slides

- https://github.com/MHeironimus/devup-2024
- https://bit.ly/devup2024

# Questions?