Collaborative Programming

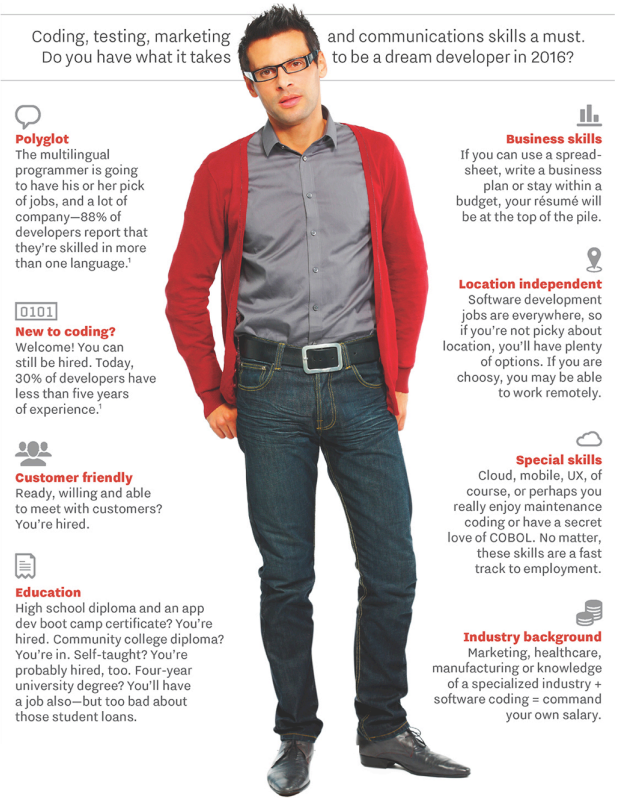
Working as a Solo Developer

Working in a Small Team (2-5)

Working on a Open Source Project

What does Industry wants ?

Dream

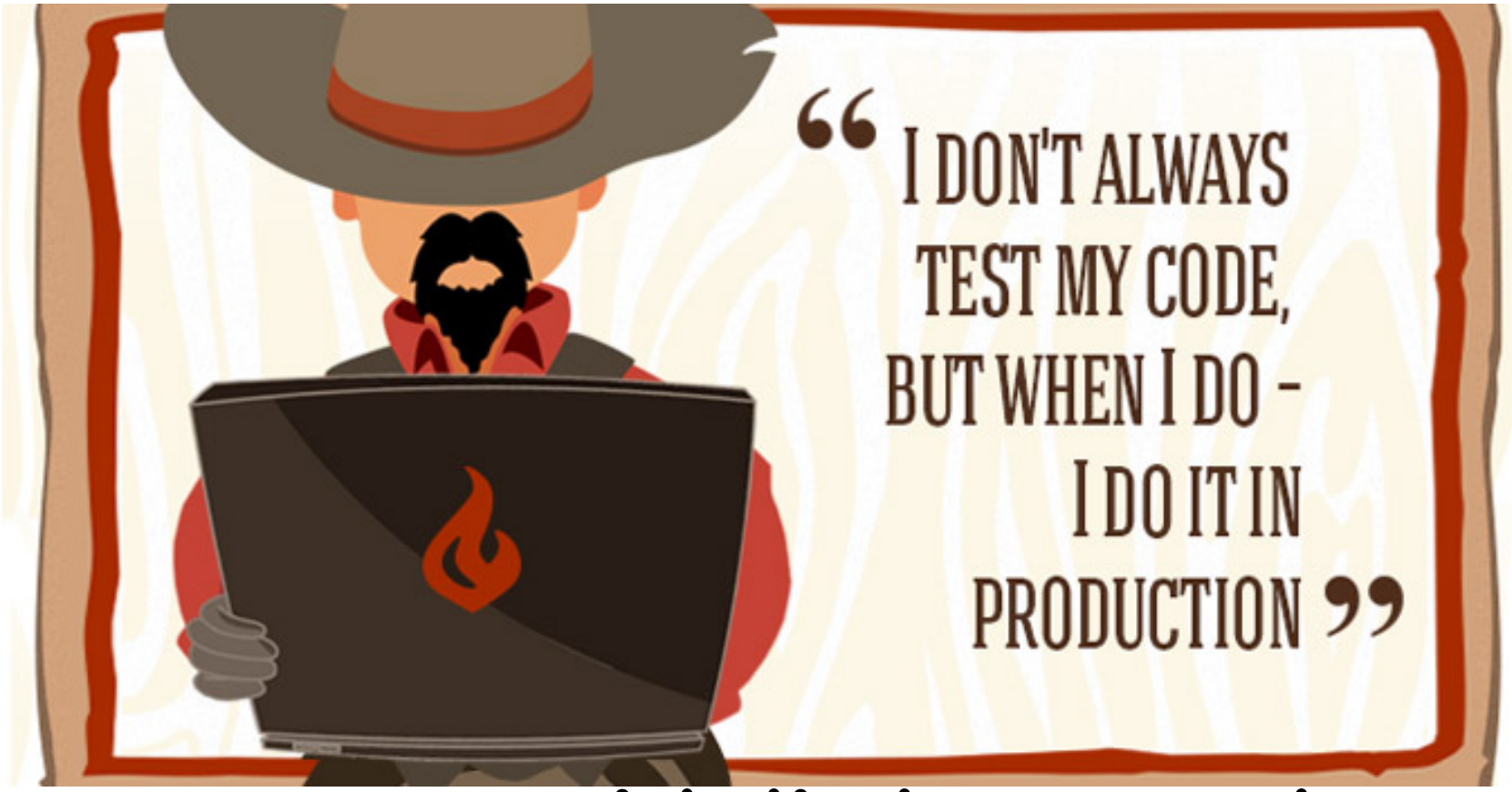


Developer

Cowboy

Developer

Who is a Cowboy Coder ?



A programmer with little regards to best practices

Martin Flower

Good Programmer write code that humans understand

A Typical Scenario

A client has asked you (Developer) to make a Website

He has requested a new feature few days back but on intermediate delivery to remove it

But after few days he has requested again for the same feature in the final delivery of the website

A Typical Scenario

Challenge 1

How will the developer find the code for the last implemented feature

Solution 1

Developer has to search all his emails with the client conversation

Challenge 2

Project has multiple file changes to achieve the feature

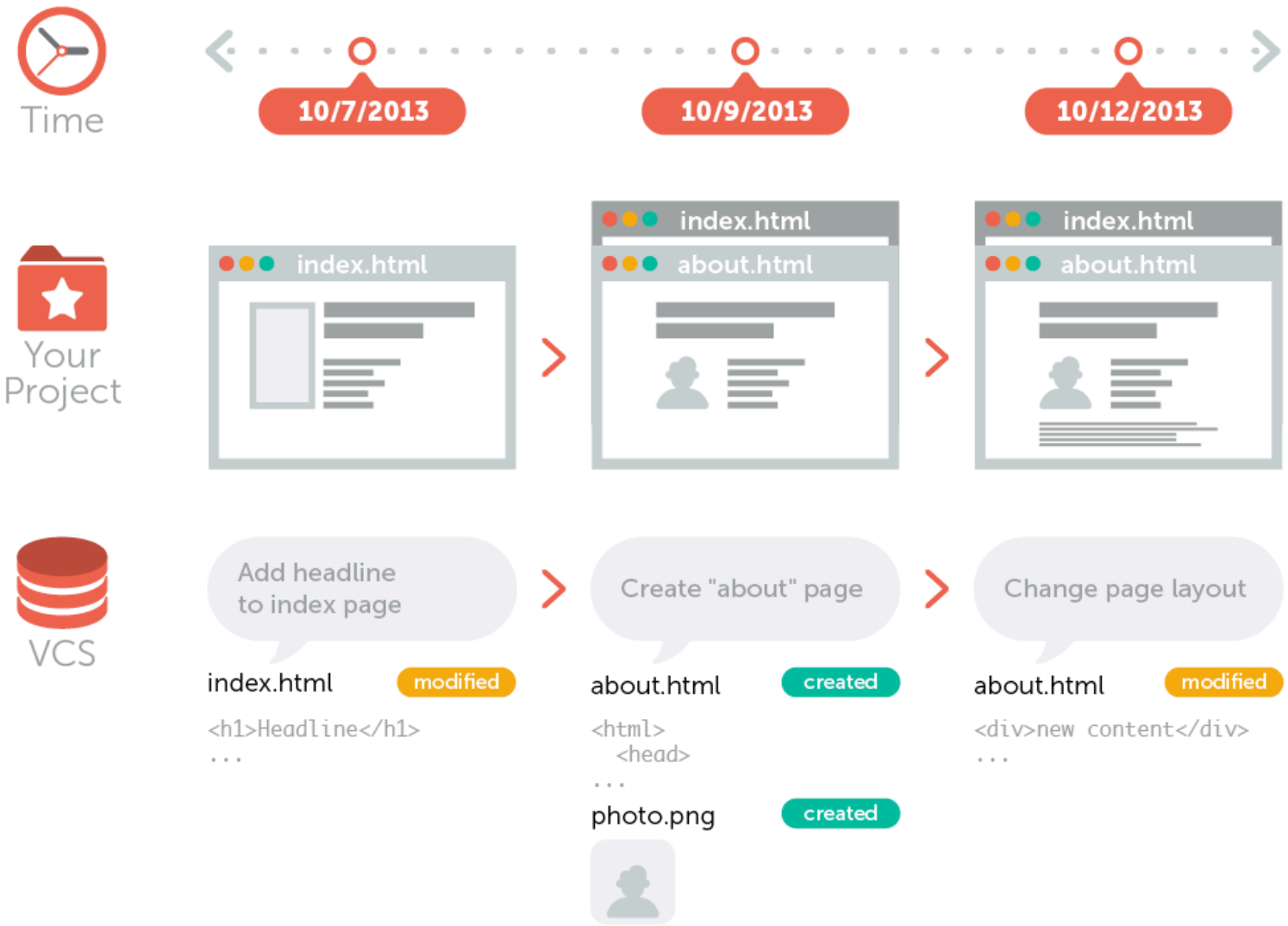
Solution 2

Keep the backup of full project by zipping it with time stamps

Best solution is a System to ...

- keep track of changes to files
- let you undo mistakes
- work on different versions of same project

What is Version Control System (VCS) ?



Why use VCS or SCM ?

Enforce Discipline

Archive Versions

Maintain Historical Information

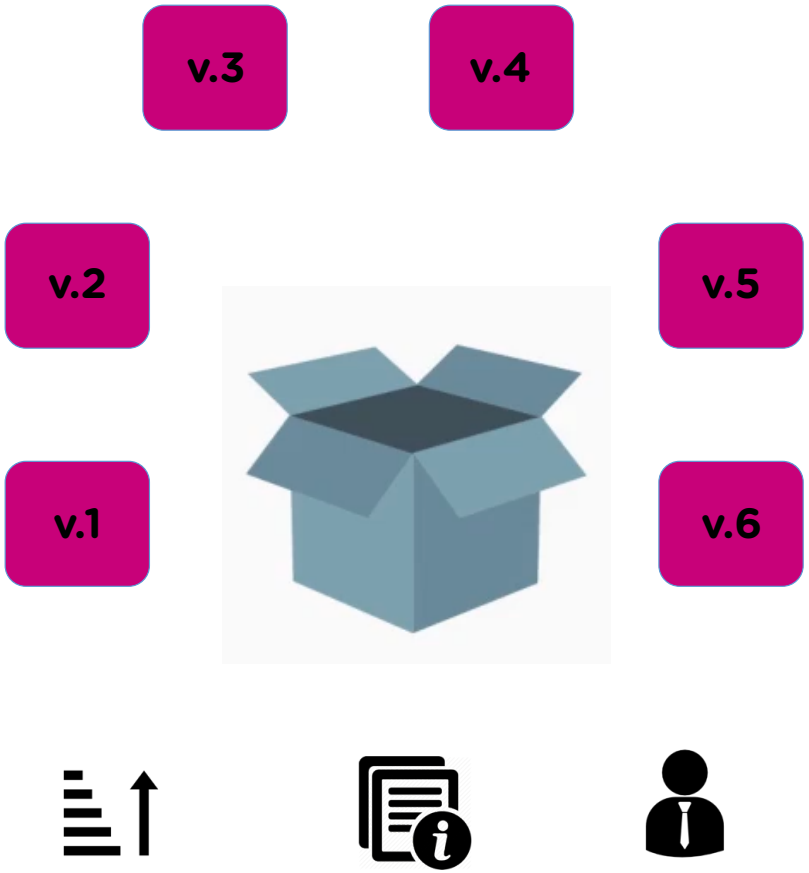
Enable Collaboration

Recover from accidental deletions or edits

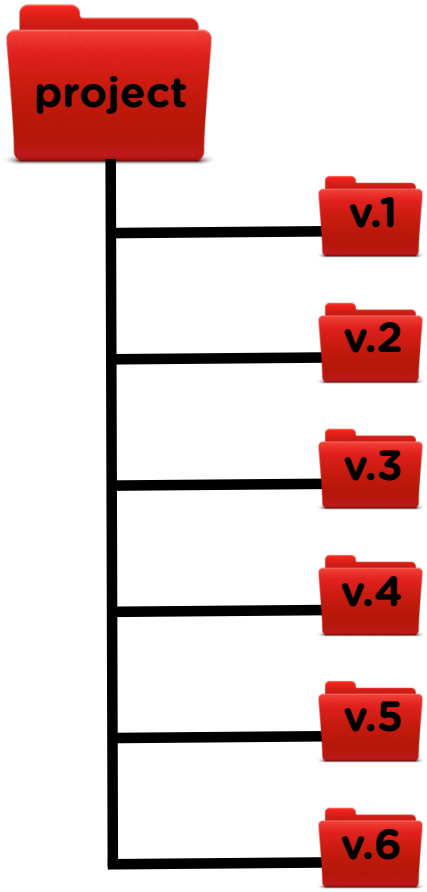
Conserve Disk space

*SCM = Source Code Management

How VCS looks like?

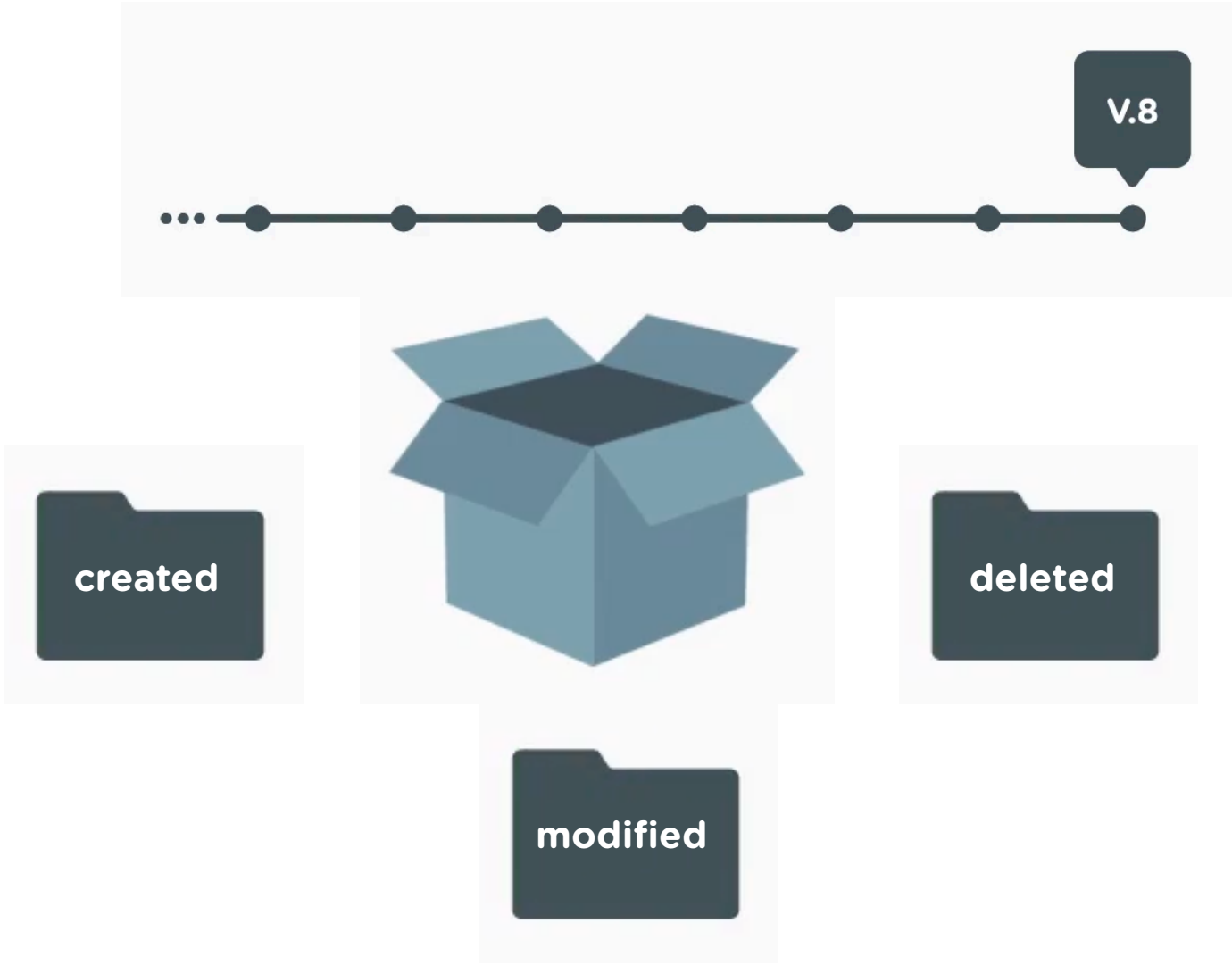


Digital Repository

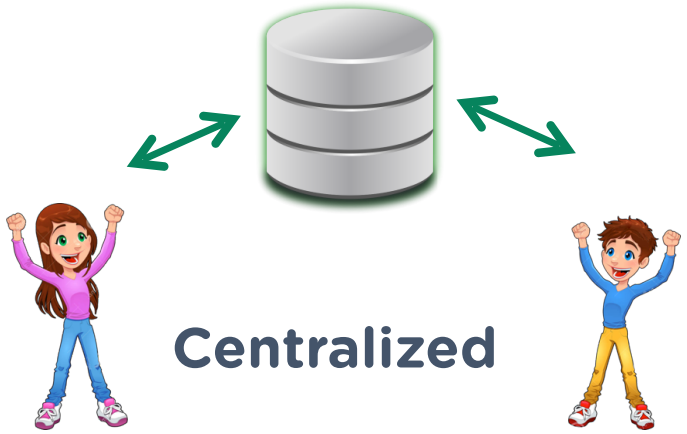


Manual versioning

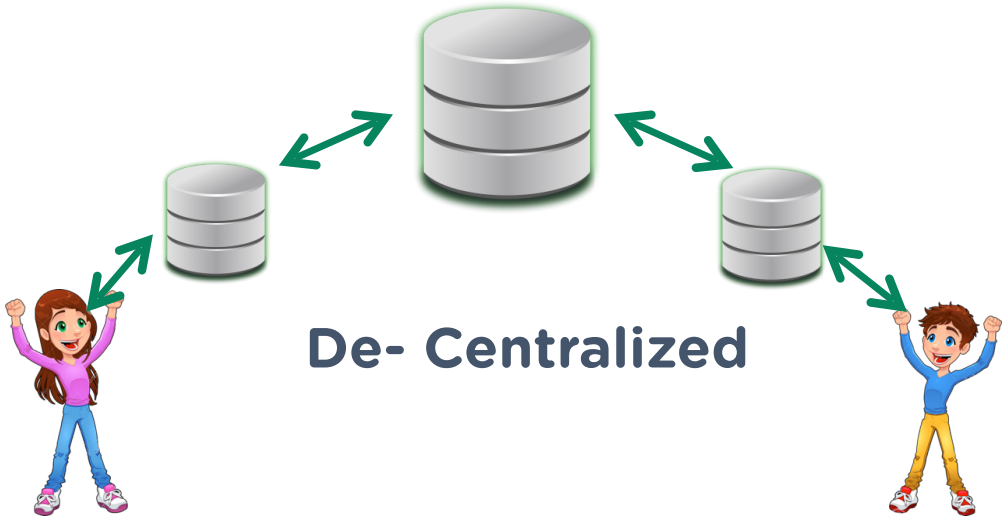
How VCS works ?



Which VCS to use ?



Centralized



De- Centralized

SVN

Mercurial

Perforce

GIT

Quiz

Which of the following does git do ?

- keep track of changes to files.
- Notice conflicts between changes made by different people.
- Synchronise files between different computers.
- All of the above.

Answer

Which of the following does git do ?

- keep track of changes to files.
- Notice conflicts between changes made by different people.
- Synchronise files between different computers.
- ✓ All of the above.

How to verify that git is installed ?

mac (on terminal)

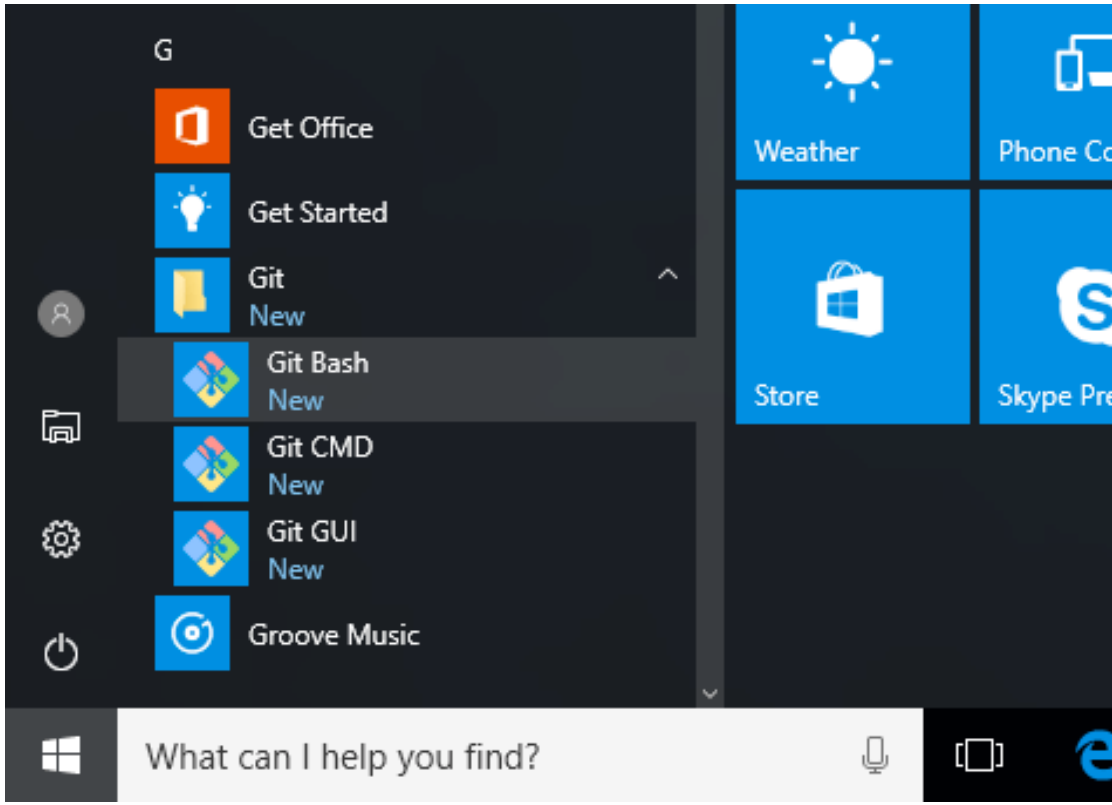
$ git --version
git version 2.6.4

linux (on terminal)

$ git --version
git version 2.6.4

windows (on git bash)

$ git --version
git version 2.6.4



Setting up git on your computer ?

mac

It's installed with Xcode

or:

http://git-scm.com/download/mac

http://code.google.com/p/git-osx-installer

linux

$ sudo yum install git

or:

$ sudo apt-get install git

windows

http://msysgit.github.io/

or:

http://git-scm.com/download/win

How to re-verify that git is installed ?

mac (on terminal)

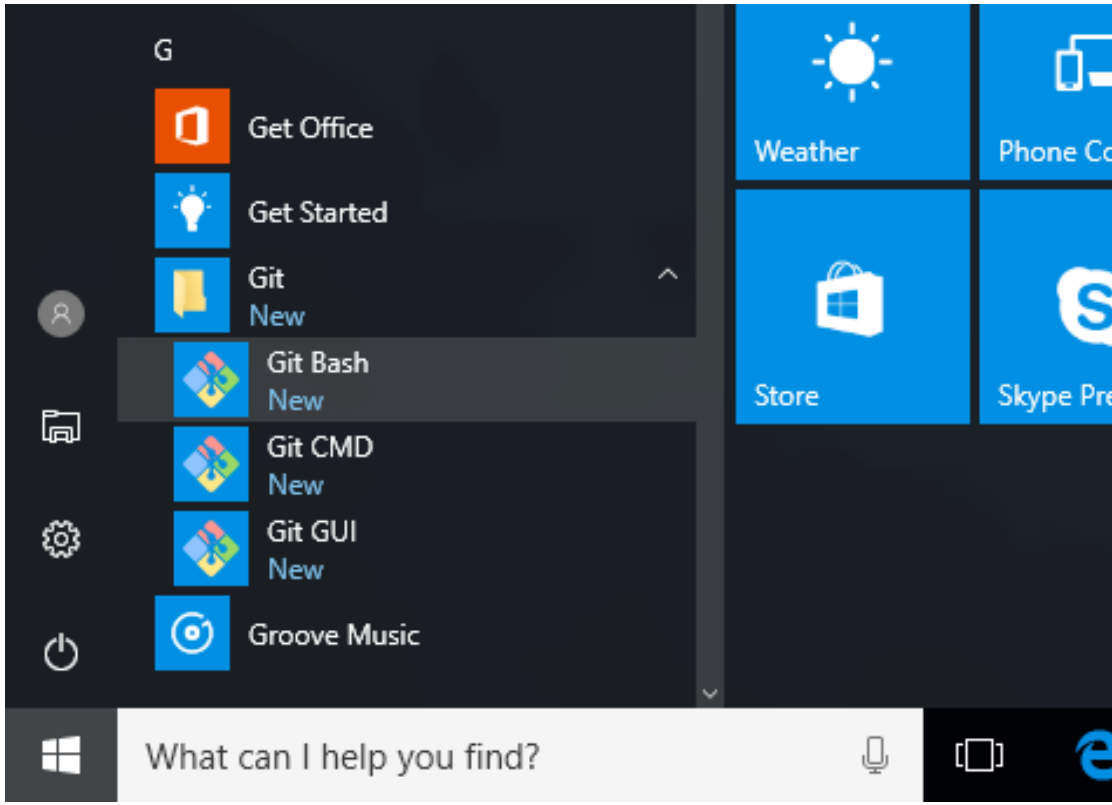
$ git --version
git version 2.6.4

linux (on terminal)

$ git --version
git version 2.6.4

windows (on git bash)

$ git --version
git version 2.6.4



How can I see git is configured ?

```
$ git config --list
```

```
$ git config --global user.name
```

```
$ git config --global user.email
```

```
$ git config --global color.ui
```

How can I change my git configuration ?

```
$ git config --global user.name “Dr. Sylvester Fernandes”

$ git config --global user.email “sylvester@forsk.in”

$ git config --global color.ui true
```

What is a Working Directory?

A folder containing the files you want to track

The folder of your project

What is a Local Repository or repo ?

Think of a repo as a kind of database where your VCS stores all the versions and metadata

repo is just a simple hidden folder named ".git" in the root of the working directory

Create a Empty Repository of working dir

$ cd ~/Documents

$ git init my_repo

{ Initialized empty Git repository in /Users/sylvester/Documents/my_repo/.git/}

Or

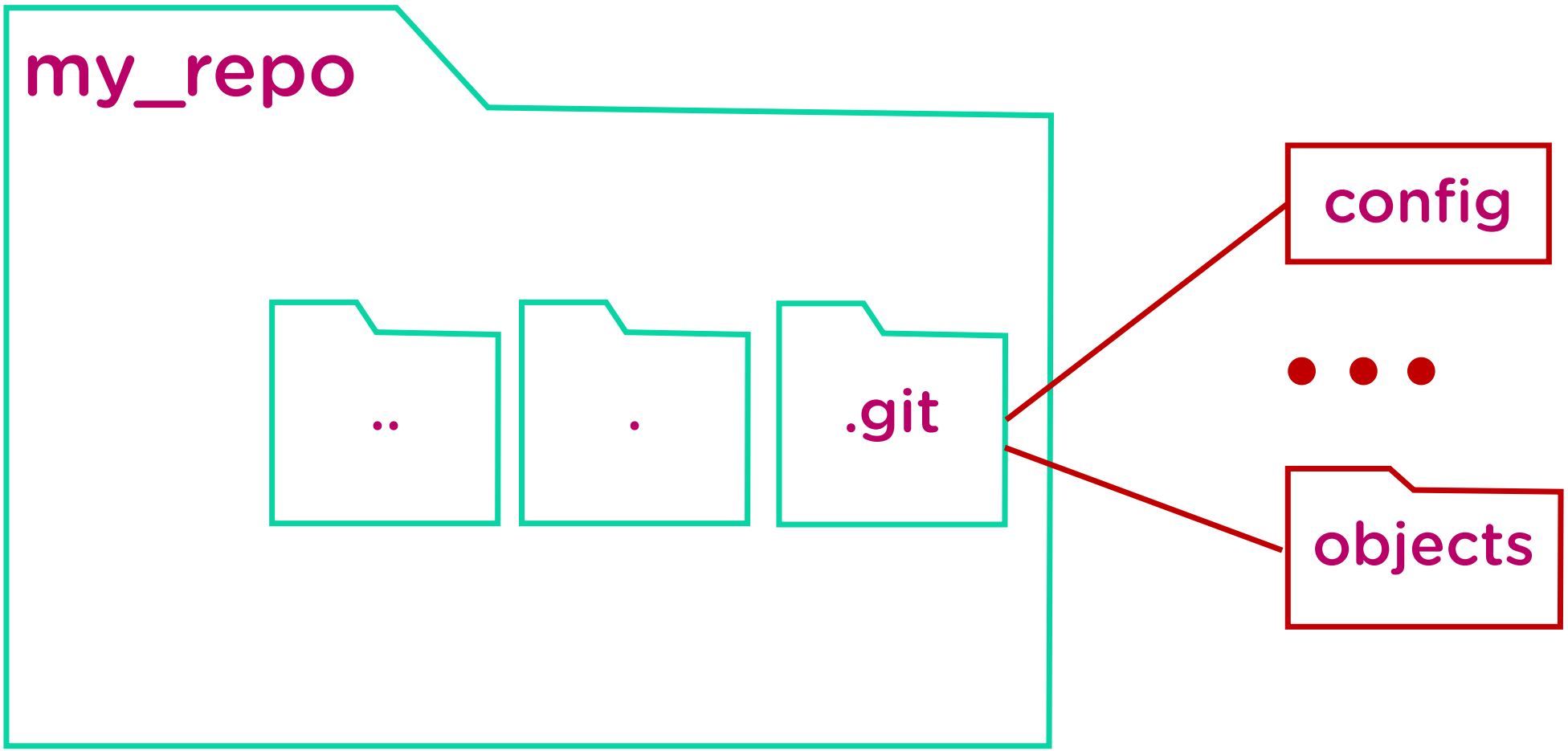
$ cd ~/Documents

$ cd old_proj

$ git init

{ Initialized empty Git repository in /Users/sylvester/Documents/old_proj/.git/}

Empty Repository created



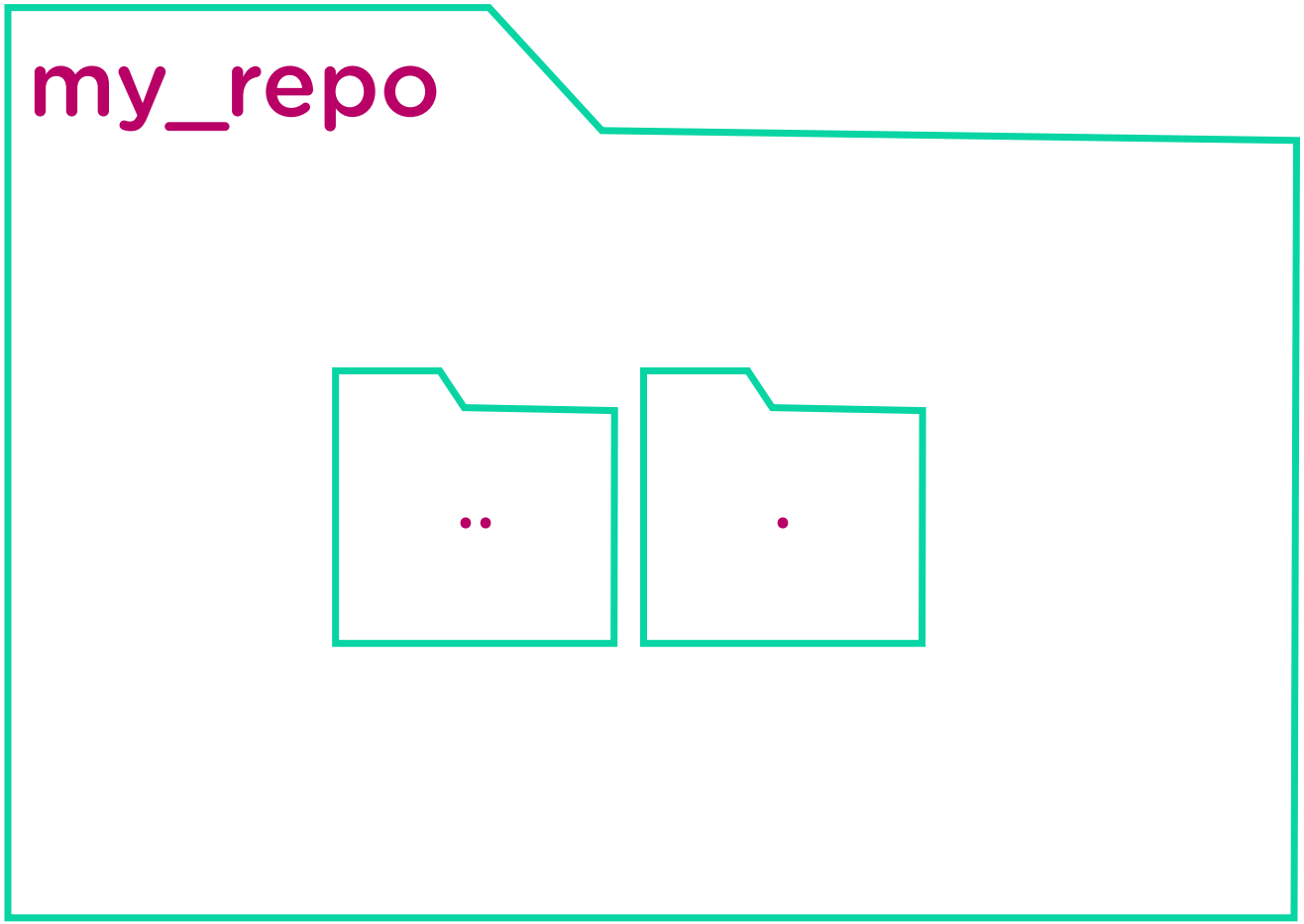
Delete a Repository of a working dir

```
$ rm -r my_repo/.git
```

All git information is stored in a .git folder in the project's root.

Just delete the .git folder, the history is gone

Repository deleted



Quiz

Suppose your home directory /home/repl contains a repo dental, which has a sub directory data. Where is the information about the history of the files in /home/repl/dental/data stored ?

- /home/repl/.git
- /home/repl/dental/.git
- /home/repl/dental/data/.git
- None of the above.

Answer

Suppose your home directory /home/repl contains a repo dental, which has a sub directory data. Where is the information about the history of the files in /home/repl/dental/data stored ?

- [ ] /home/repl/.git
- [x] /home/repl/dental/.git
- [ ] /home/repl/dental/data/.git
- [ ] None of the above.

commit (snapshot / taking a picture)

commit is a snapshot of your project at a certain point of time

keep track of your changes by committing changed files to the repo

use the **"$git status"** command to get a list of all the changes you performed since the last commit

Golden rule of VCS !!

No. 1: Git doesn’t tracks files by default

No. 2: Commit early, commit often

No. 3: Commit Only Related Changes

No. 4: Write Good Commit Messages

Status of file in repo

- untracked – new file that's never been added to vcs

- tracked – new file that's been added for tracking
  - working – Tracked files with new changes
  - staged – File added to staging area
  - committed – File is added to local repo

Getting an Overview of Your Changes?

$ git status

untracked

staged

new file

modified

committed

How can I view repo history ?

```
$ git log
```

Whats the first steps in saving the change

$ touch README.txt

($ git status)

$ git add README.txt

($ git status)

How do I commit changes ?

$ git commit –m “My first commit”

"$git log" command lists all the commits that were saved in chrono order

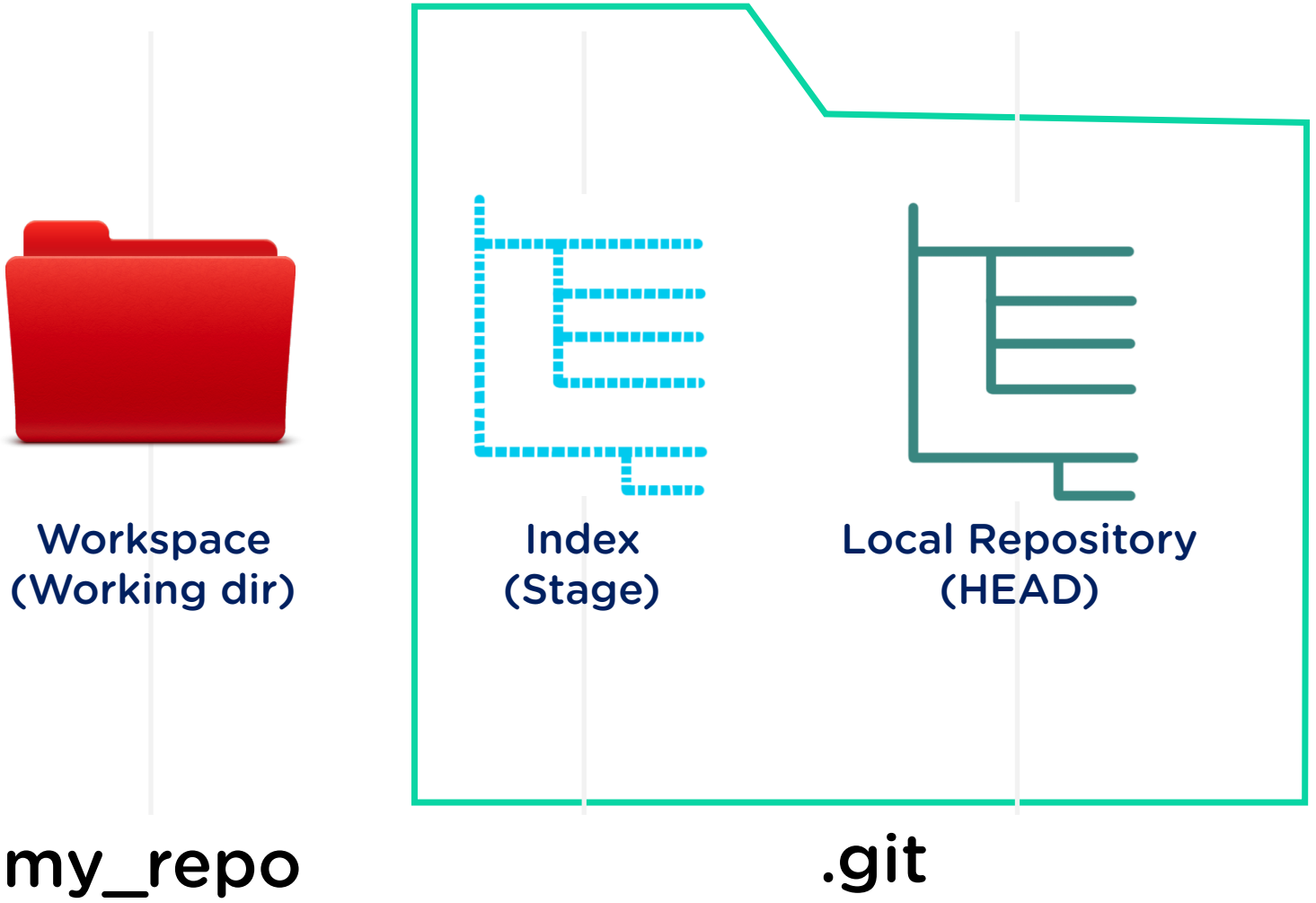
"$git status" command to get list of all changes to local repo

$ mkdir my_repo && cd my_repo

(Empty dir)

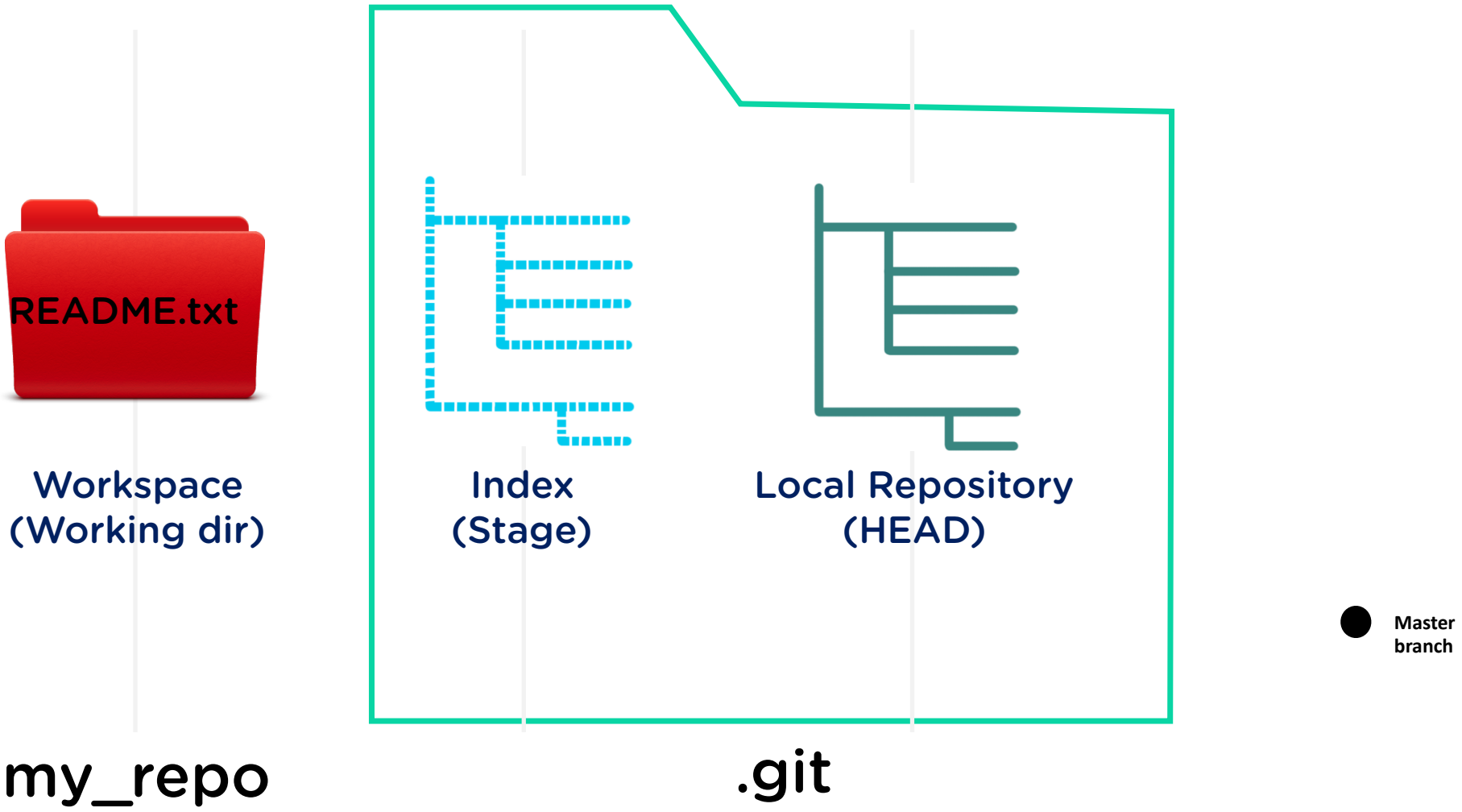
my_repo

$ git init

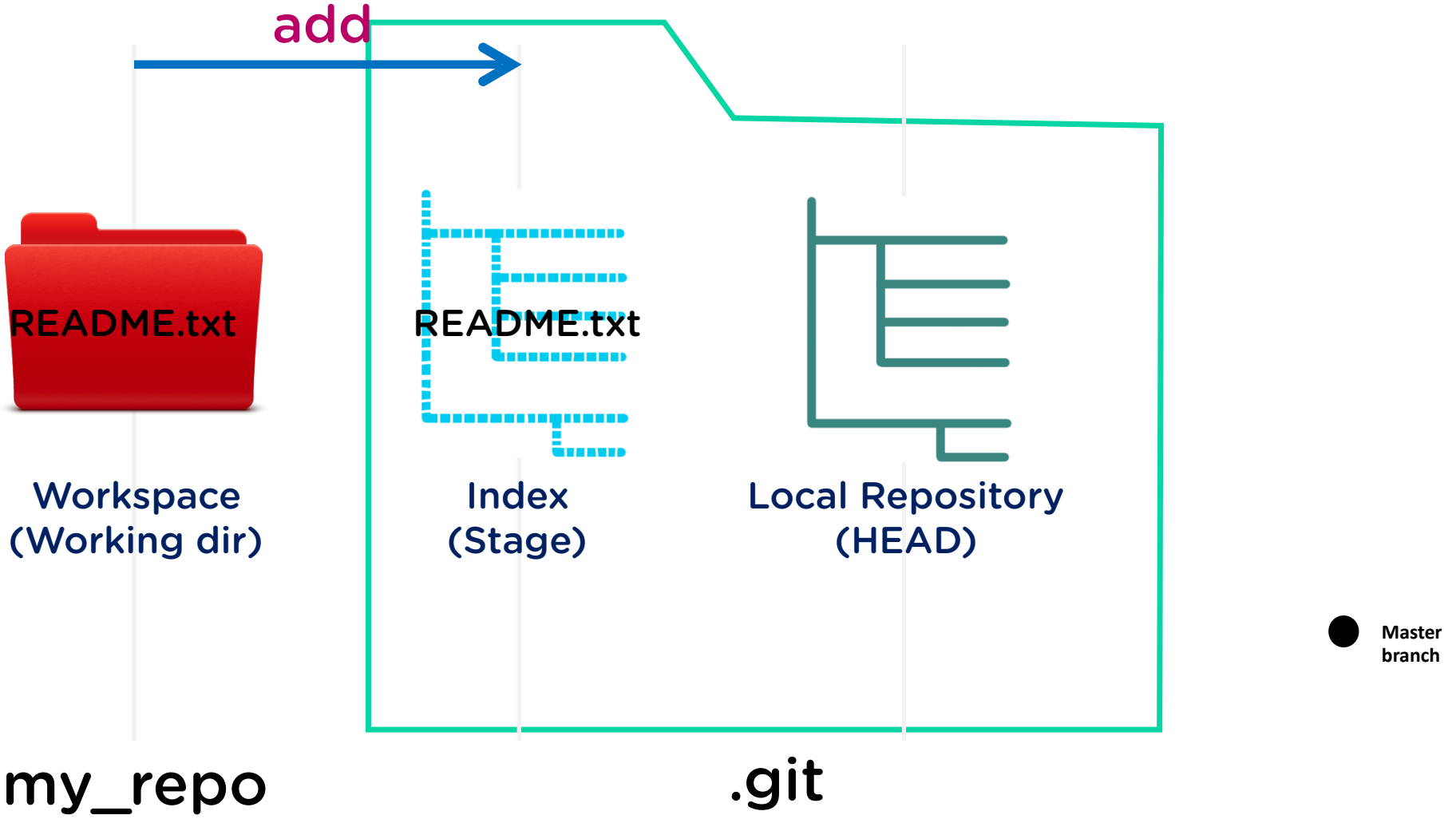


Master branch

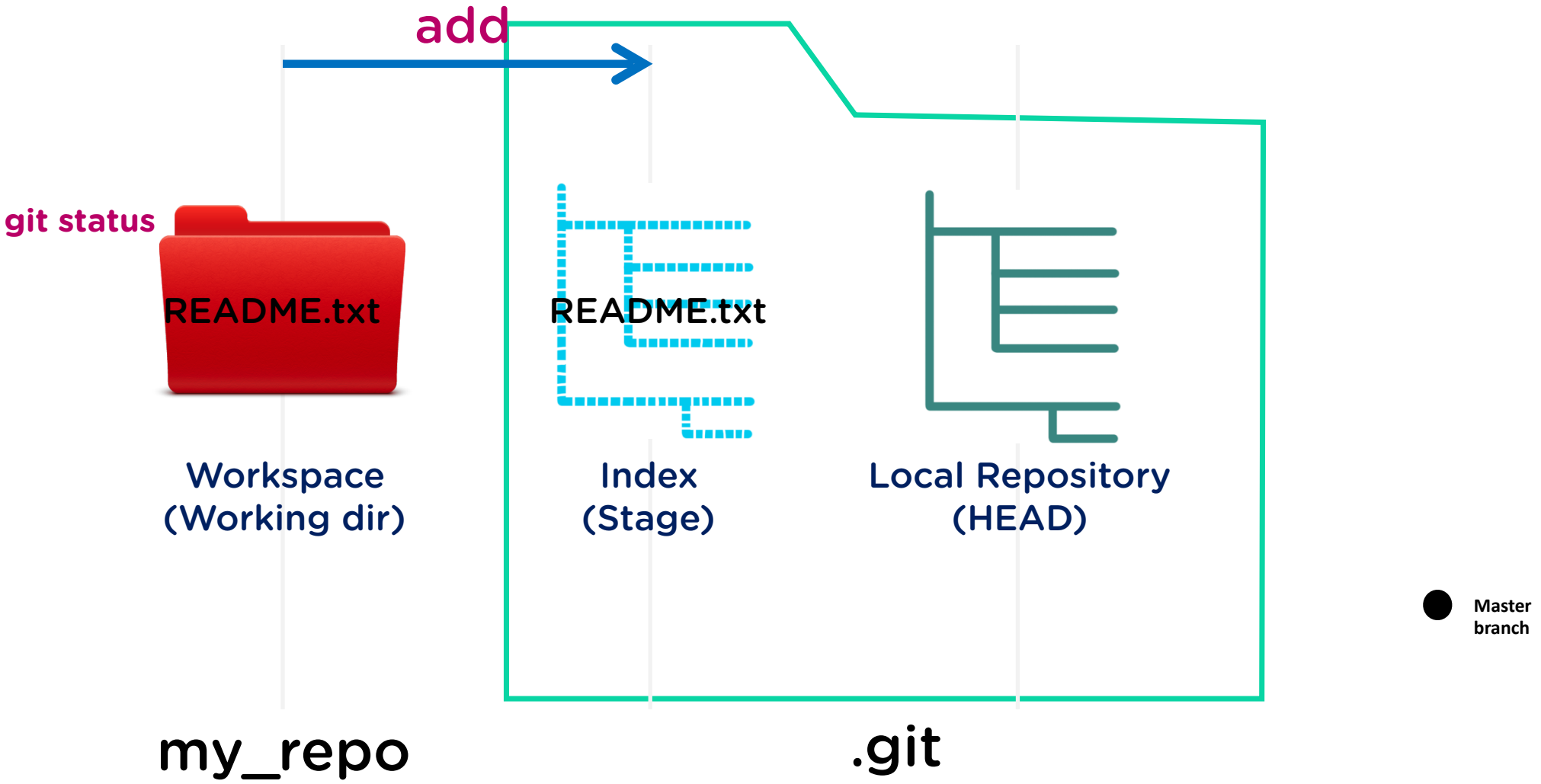
$ touch README.txt



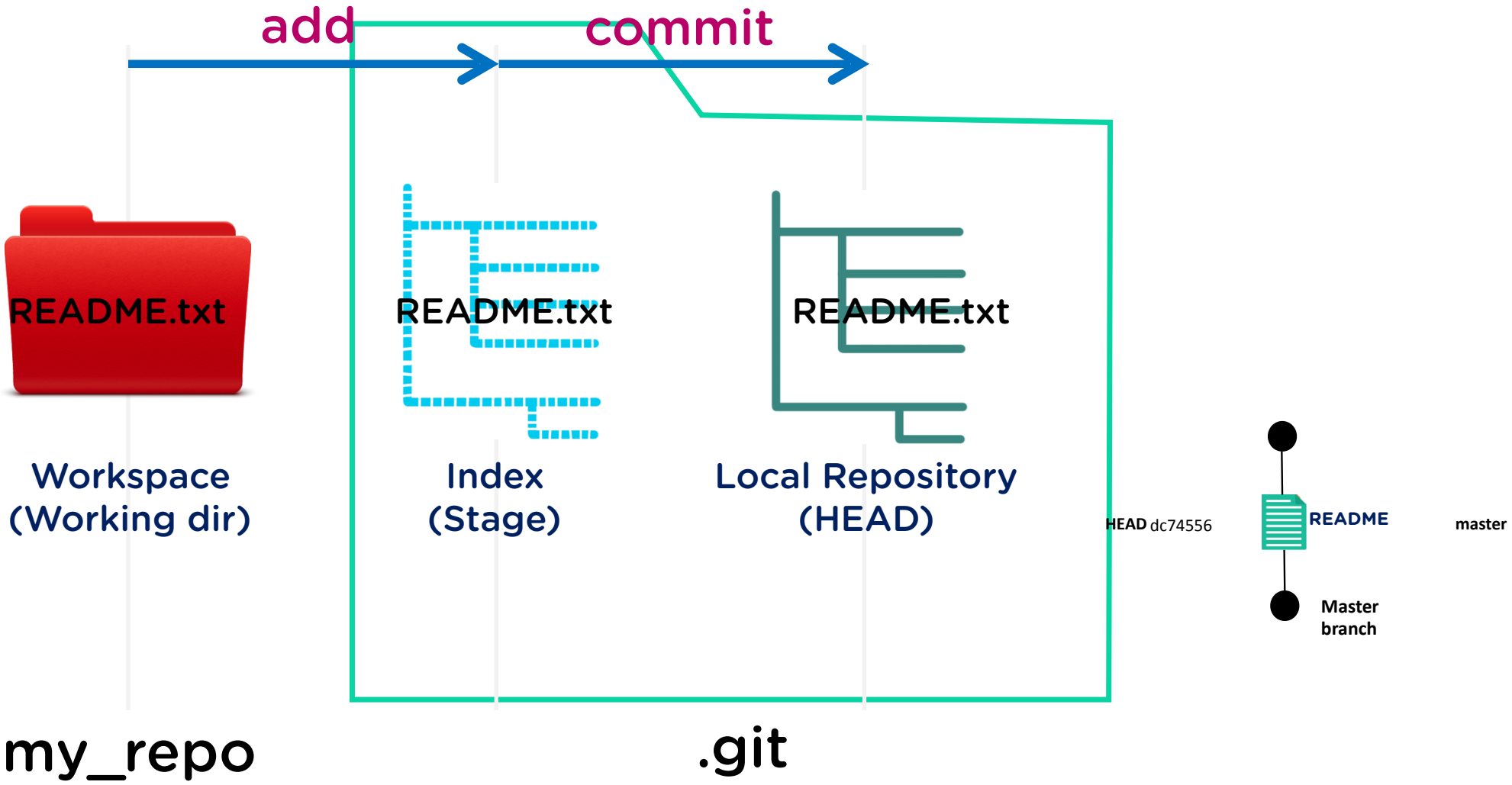
$ git add README.txt



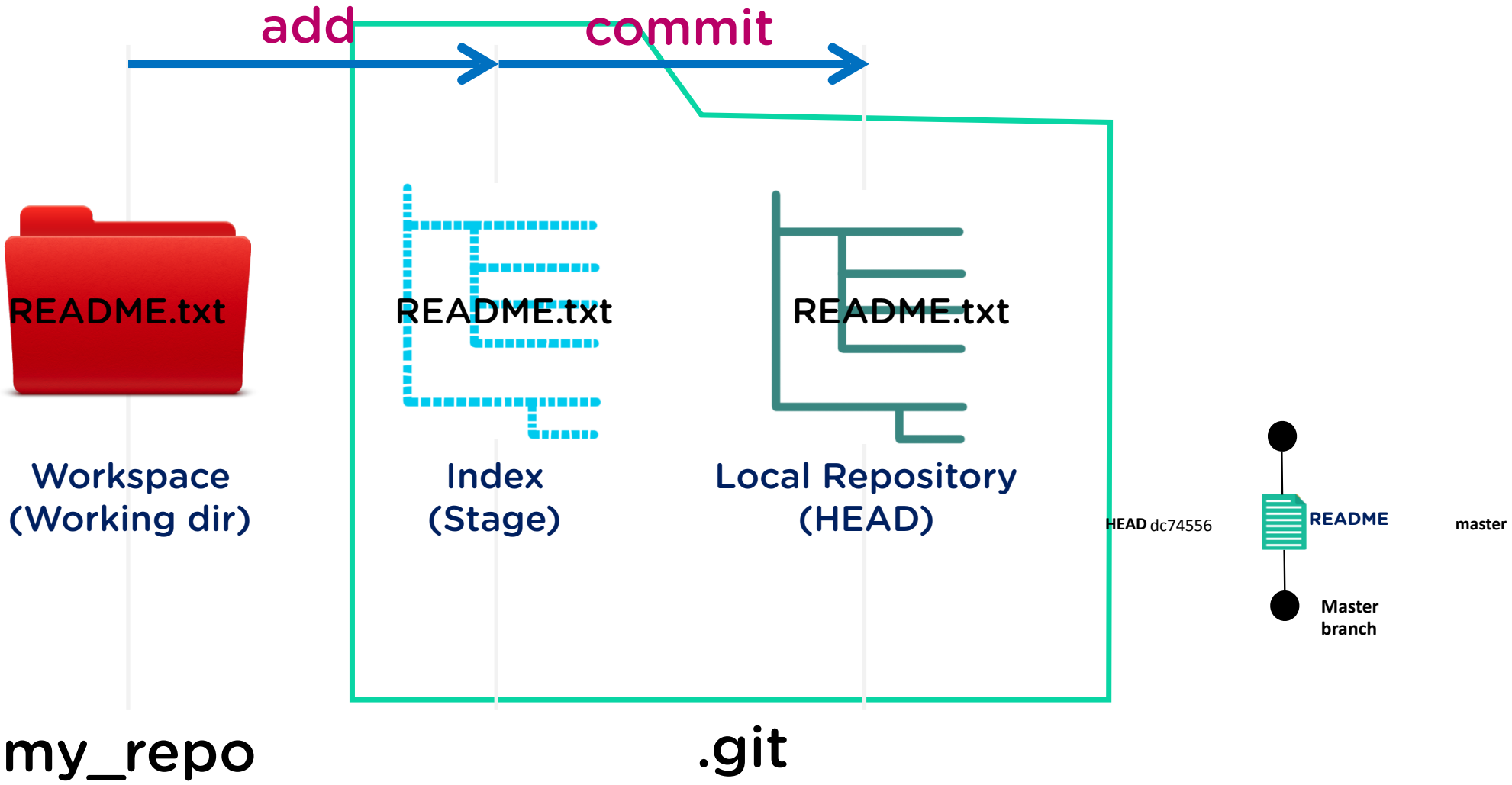
How can I check the state of the repo?



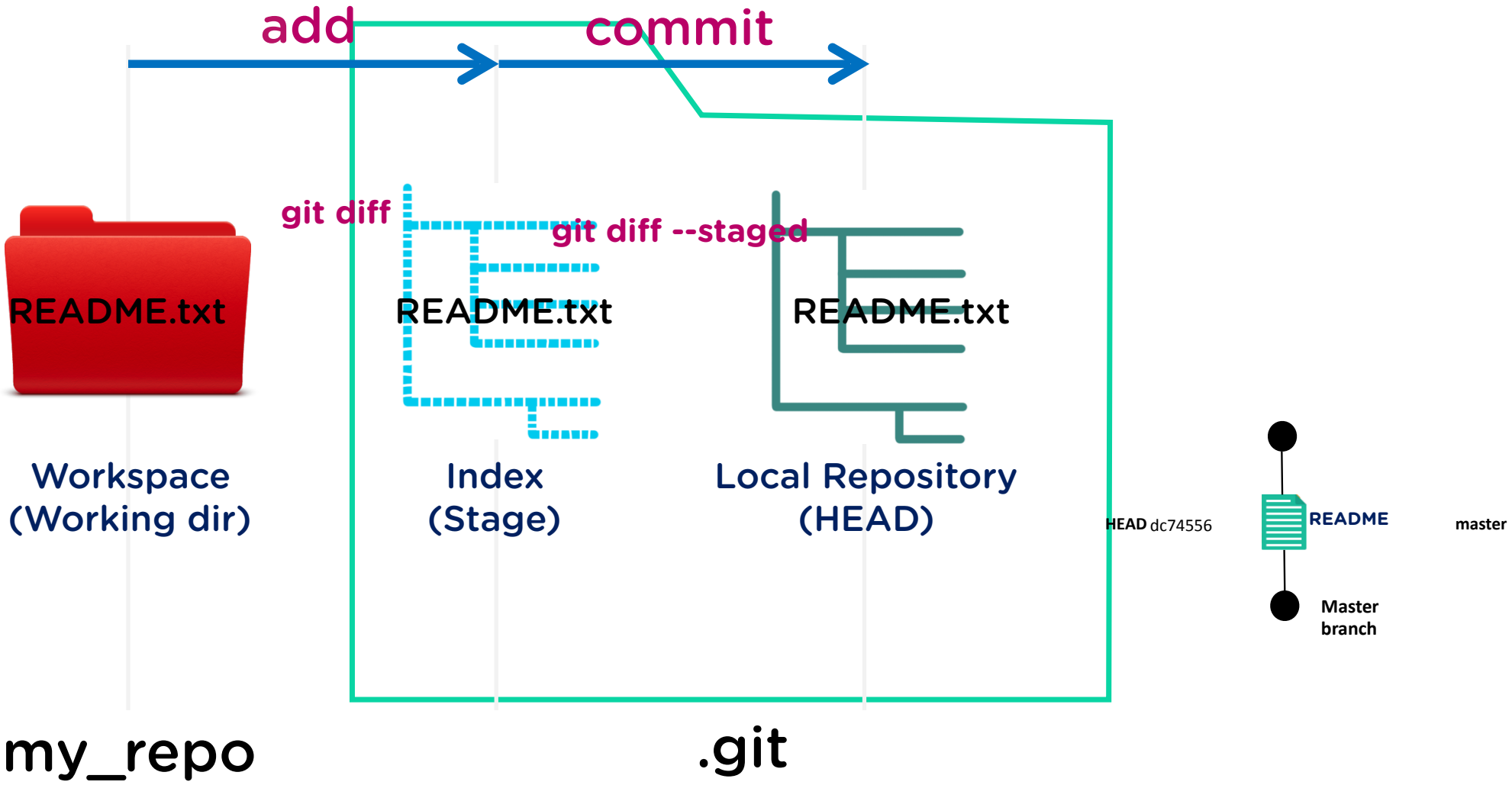
$ git commit –m “First commit”



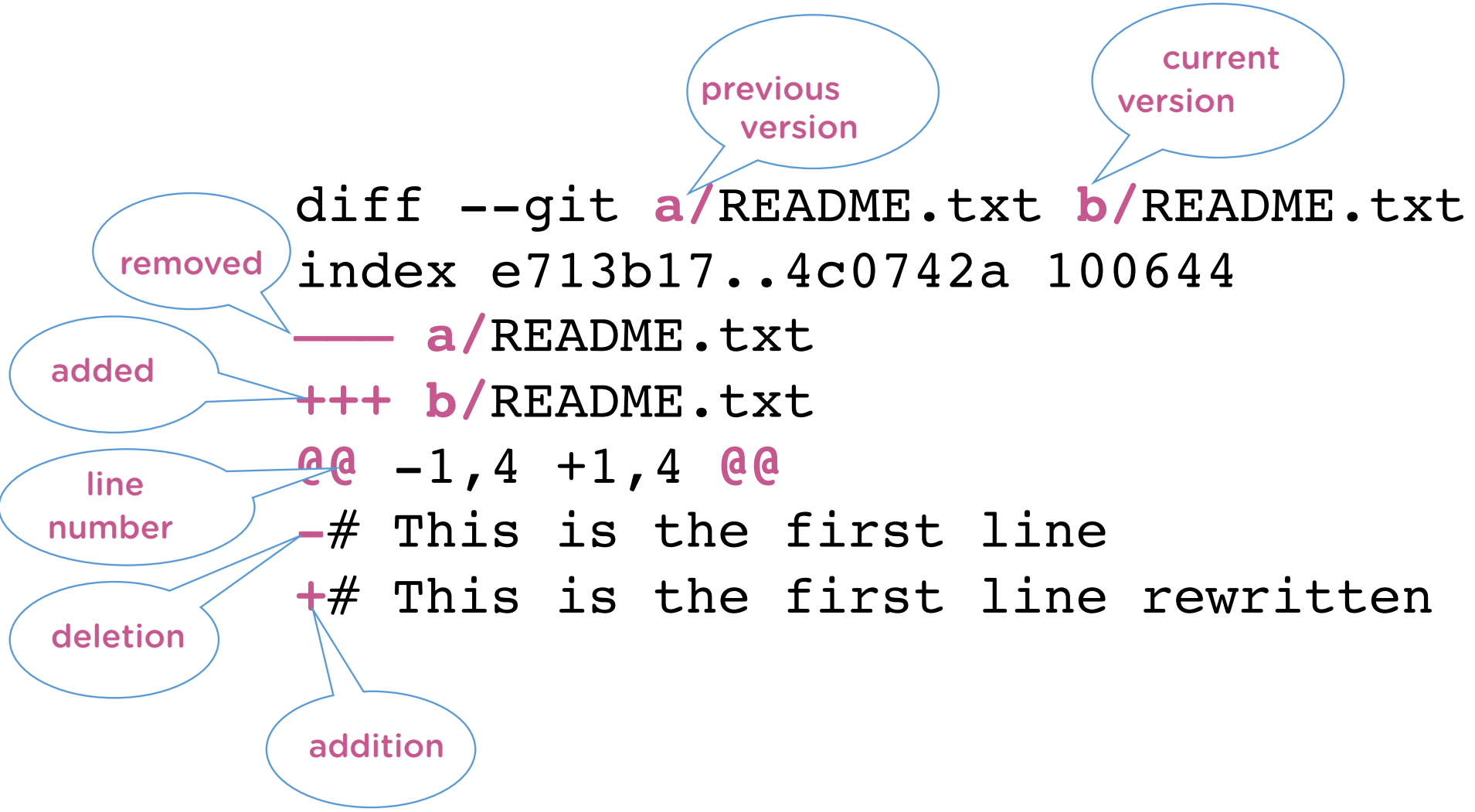
How can I tell what I have changed ?



$ git diff AND git diff --staged



What is in diff command ?



How can I view a specific file history ?

```
$ git log path/to/file
```

How can I view a specific commit ?

```
$ git show few/char/of/hash
```

How can I see what changed between 2 commits ?

```
$ git diff ID1..ID2
```

hands-on for first commit

Your second commit

$ notepad README.txt

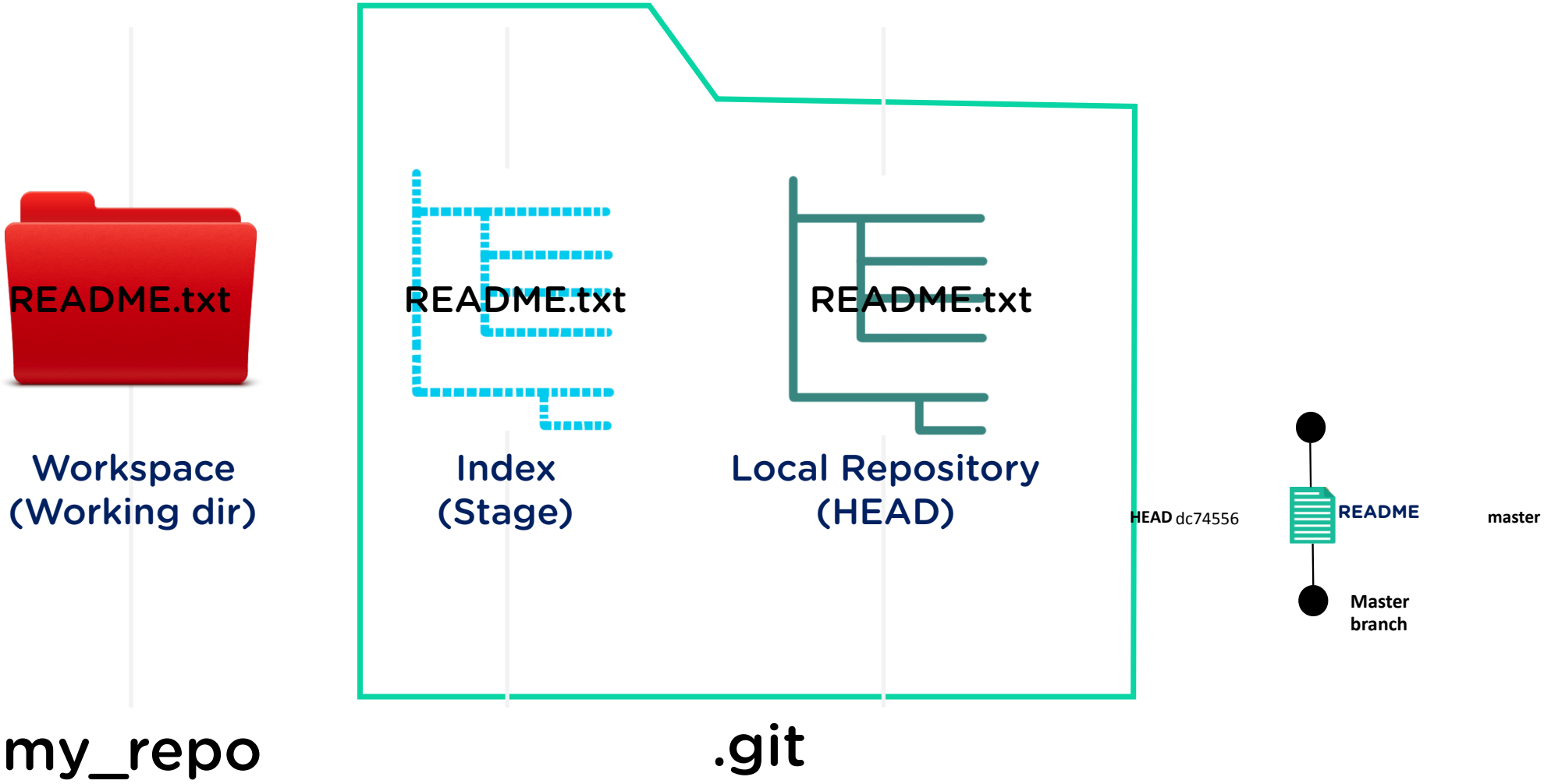
$ git add .

$ git commit –m “My sec commit”

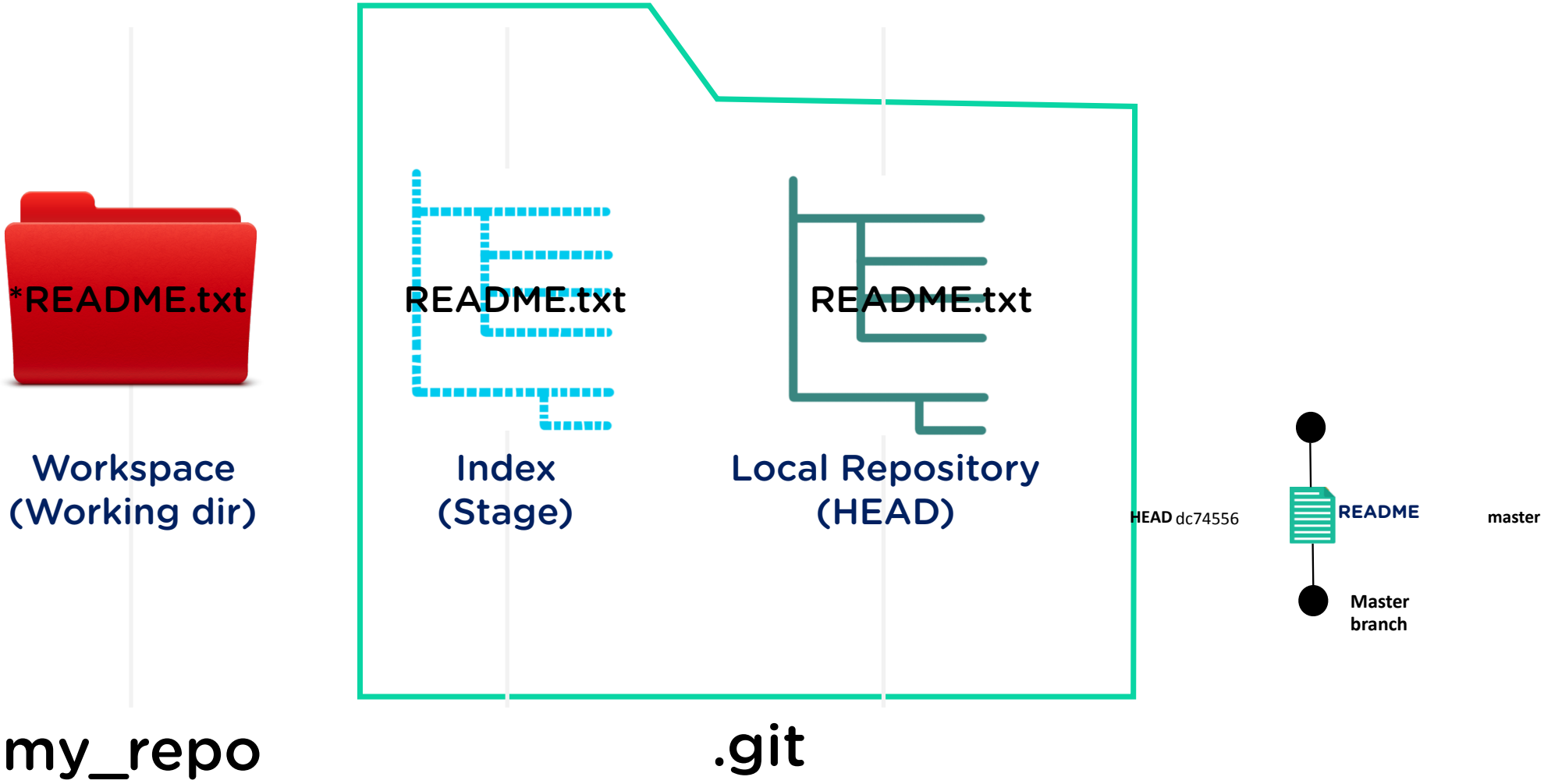
Or

$ git commit –a –m “My sec commit”

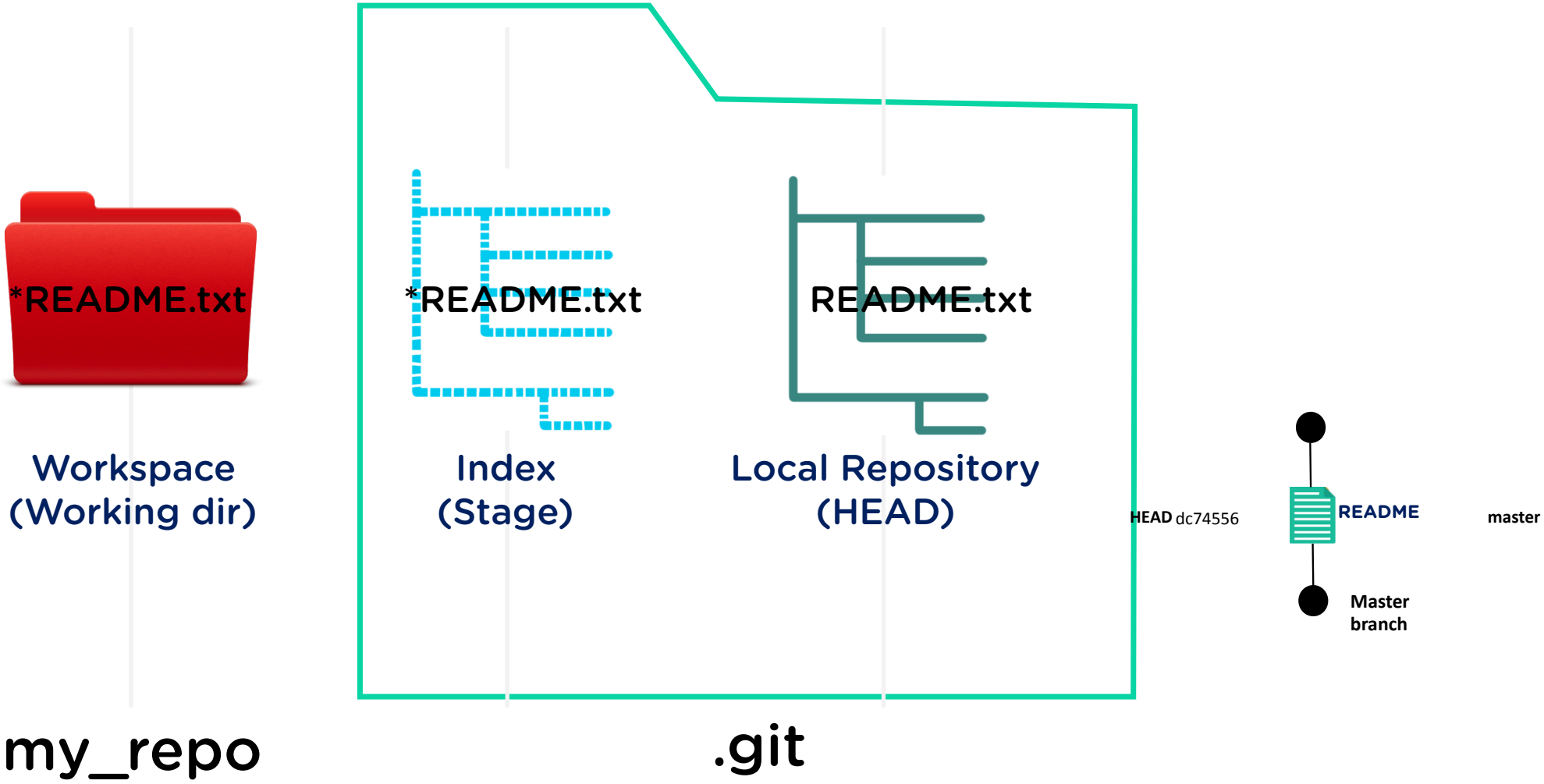
What's happening in Directory ?



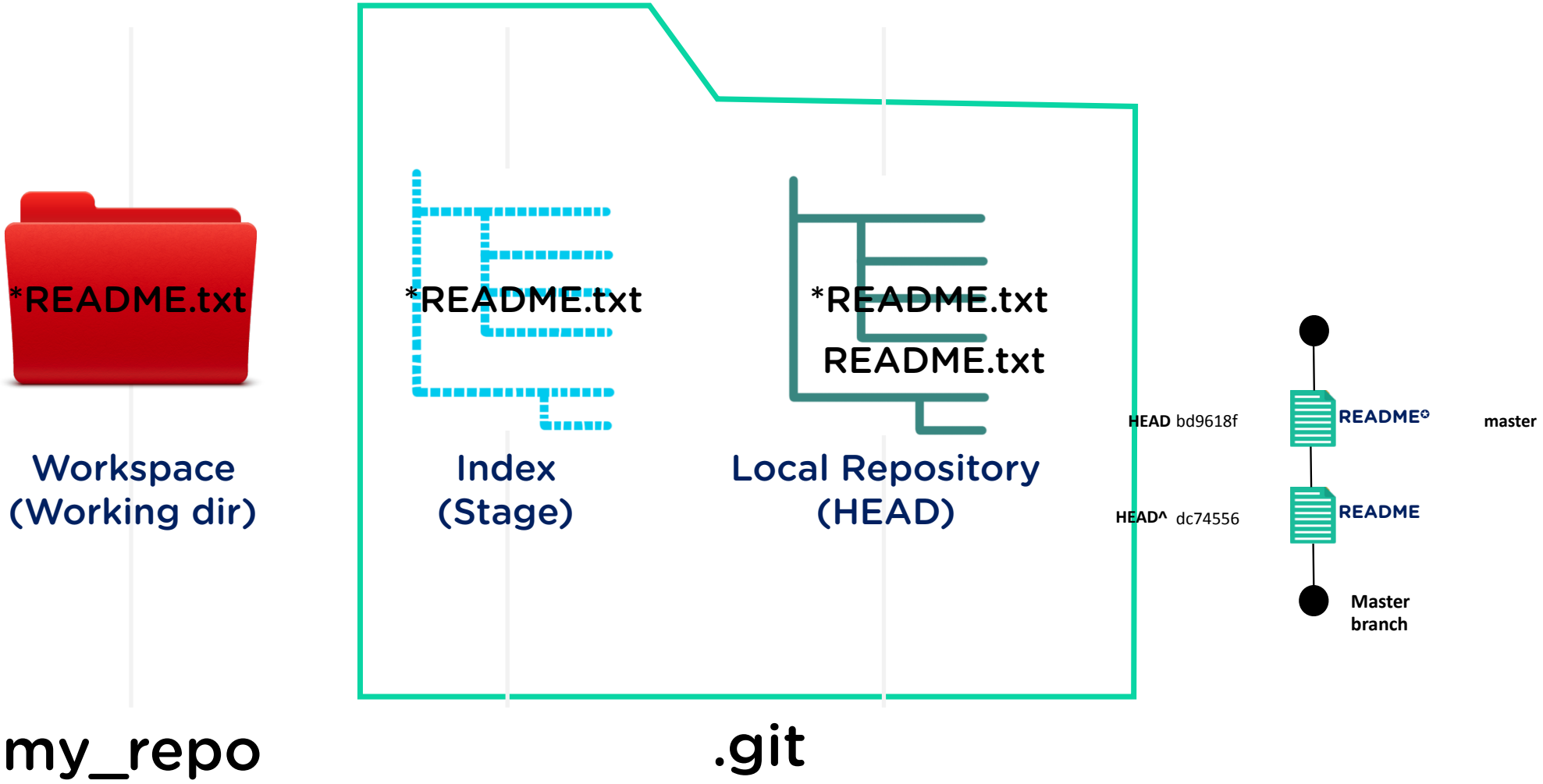
$ notepad README.txt



$ git add .

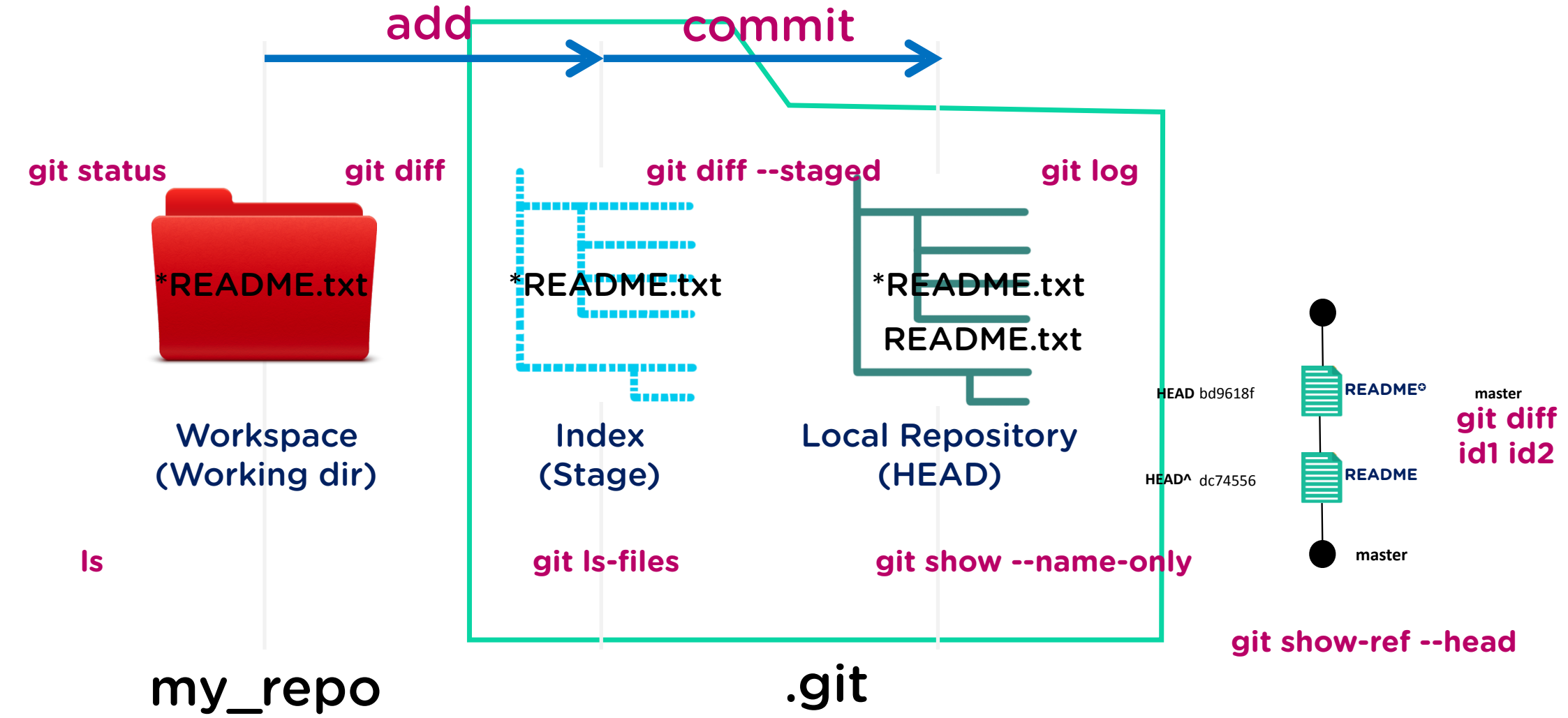

*README.txt
*README.txt
README.txt
Workspace
(Working dir)
Index
(Stage)
Local Repository
(HEAD)
HEAD dc74556
README
master
Master
branch
my_repo
.git

$ git commit –m “Second commit”



hands-on for second commit

Summary of commands



When things go wrong

Messed working but NOT staged

$ git checkout HEAD path/to/file

Revert to the last committed version of the file
(If file is NOT committed and NOT staged)
This brings from the repo to the staging area and working directory

Messed working (but staged)

$ git reset HEAD path/to/file

Brings the file from repo to the staging area
and also Un-stage the staged file, it resets the file to the state you last staged.

and then

$ git checkout HEAD path/to/file

This brings the file from staging area to the working directory
this is undoing the changes

Messed working (staged and commit)

$ git checkout HEAD^ path/to/file

Revert to version of file from prior to latest commit
(HEAD^ represents the prior to latest commit)

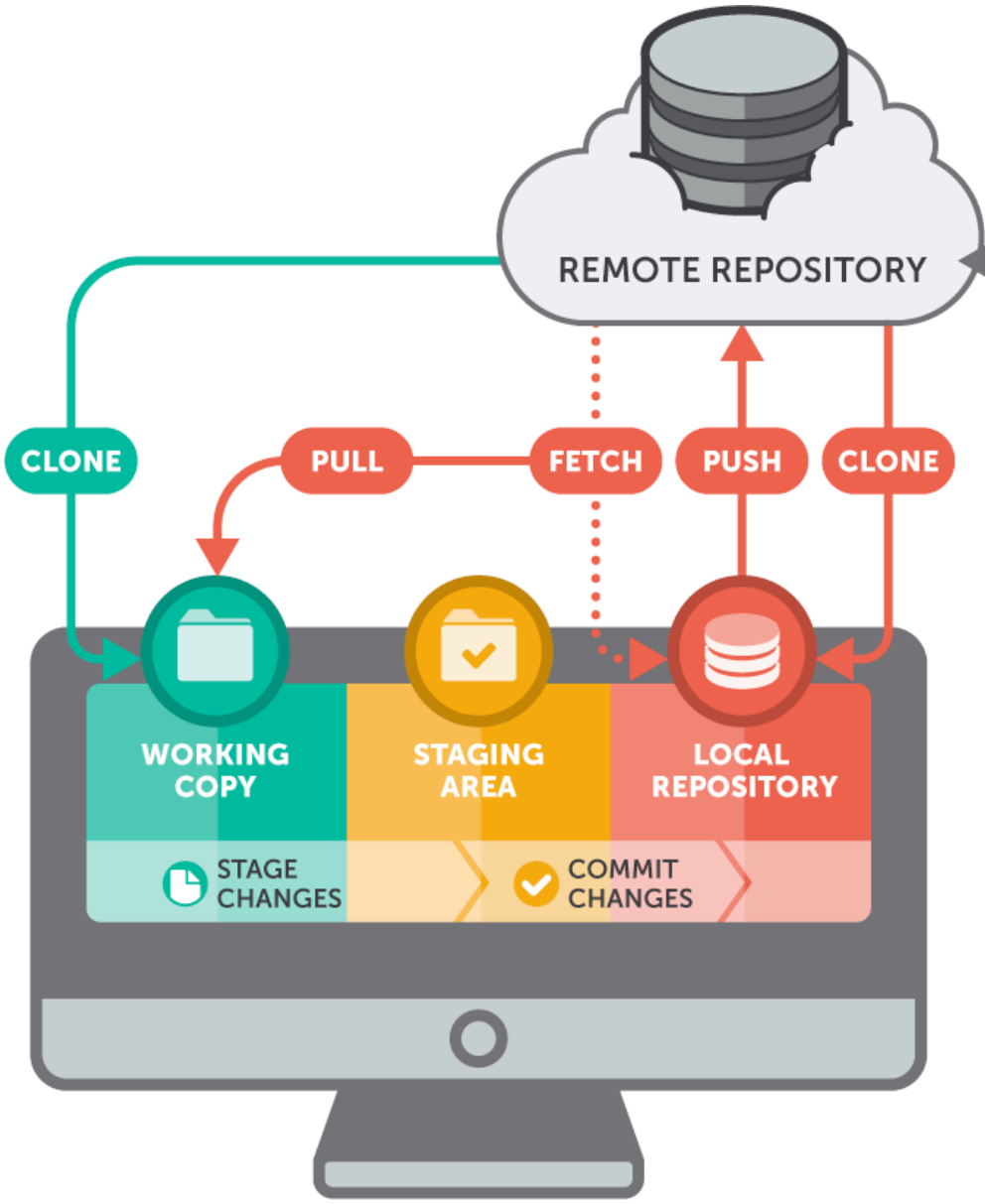
Careful: this overwrites changes to files in your working area and staging area
HEAD~1 can also be used

the act of restoring the file is saved as another commit, because you might later want to undo your undoing.

My Laptop got crashed/stolen

Solution !!

Working with
Remote Repo
on cloud to backup and
share with other
developers


REMOTE REPOSITORY
CLONE
PULL
FETCH
PUSH
CLONE
WORKING COPY
STAGING AREA
LOCAL REPOSITORY
STAGE CHANGES
COMMIT CHANGES

Github/Bitbucket

- Create an account on Github
- Update profile
- Explore Github
- Create a new public empty repo
- Push your local repo to remote repo
- Open the link in browser to view files

Adding remote to local repo

$ git remote -v

To show your remotes for repo

$ git remote show origin

to show your remote named origin

$ git remote add name/of/remote https/path/to/repo

you can connect to many remotes as you like

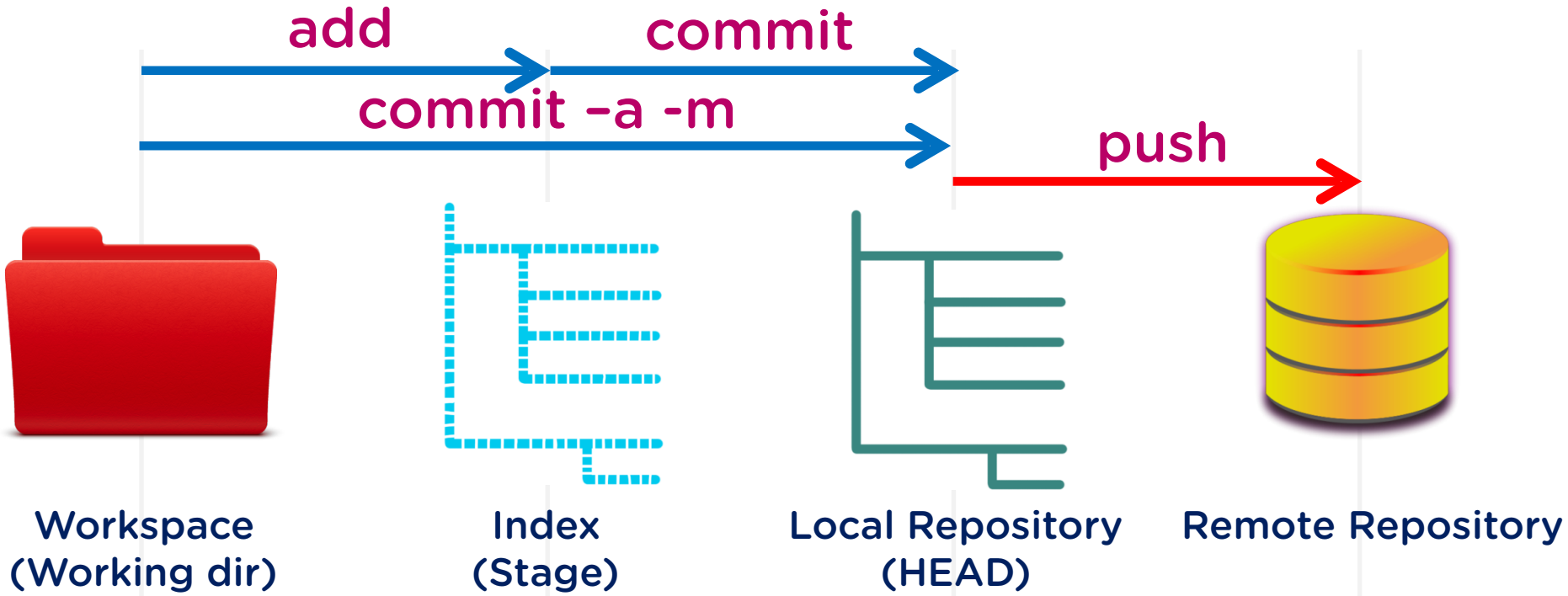
$ git remote add origin https://github.com/neuerung/my_repo.git

Pushing local repo to remote

$ git remote add origin https://github.com/neuerung/my_repo.git

$ git push –u origin master

Git workflow summary



When things go wrong

Case 1: My laptop crashed

$ cd ~/Documents

on your new laptop

$ git clone /path/to/repo name/of/new_repo

on your new laptop

Case 2: Forgot to bring my laptop

$ cd ~/Documents

on your friends laptop

$ git clone /path/to/repo name/of/new_repo

clone the remote repo to a local repo

$ git push origin master

once you have completed, push the changes back to remote

$ git pull origin master

once you are back to your own laptop, pull back the changes to local

Git workflow summary