Version Control with Git

`in 10 commands`

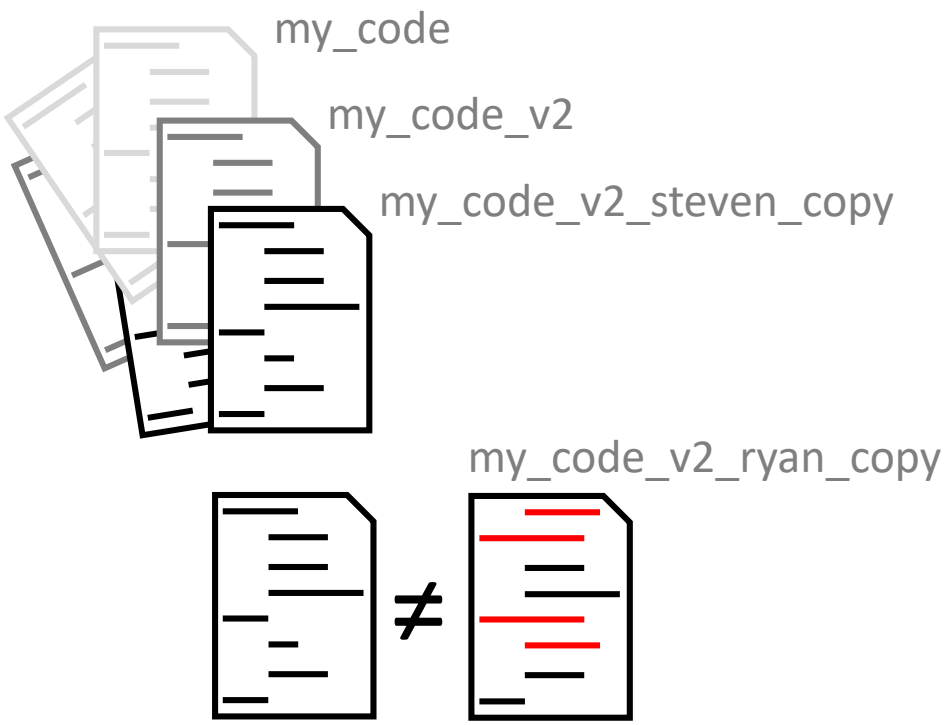
Some material adapted from Software Carpentry:
http://swcarpentry.github.io/git-novice/
http://software-carpentry.org/

spestana@uw.edu

Why use version control?

Helps keeps track of:

- **What** was changed?
- **Who** made a change?
- **When** a change was made?
- **Why** a change was made?



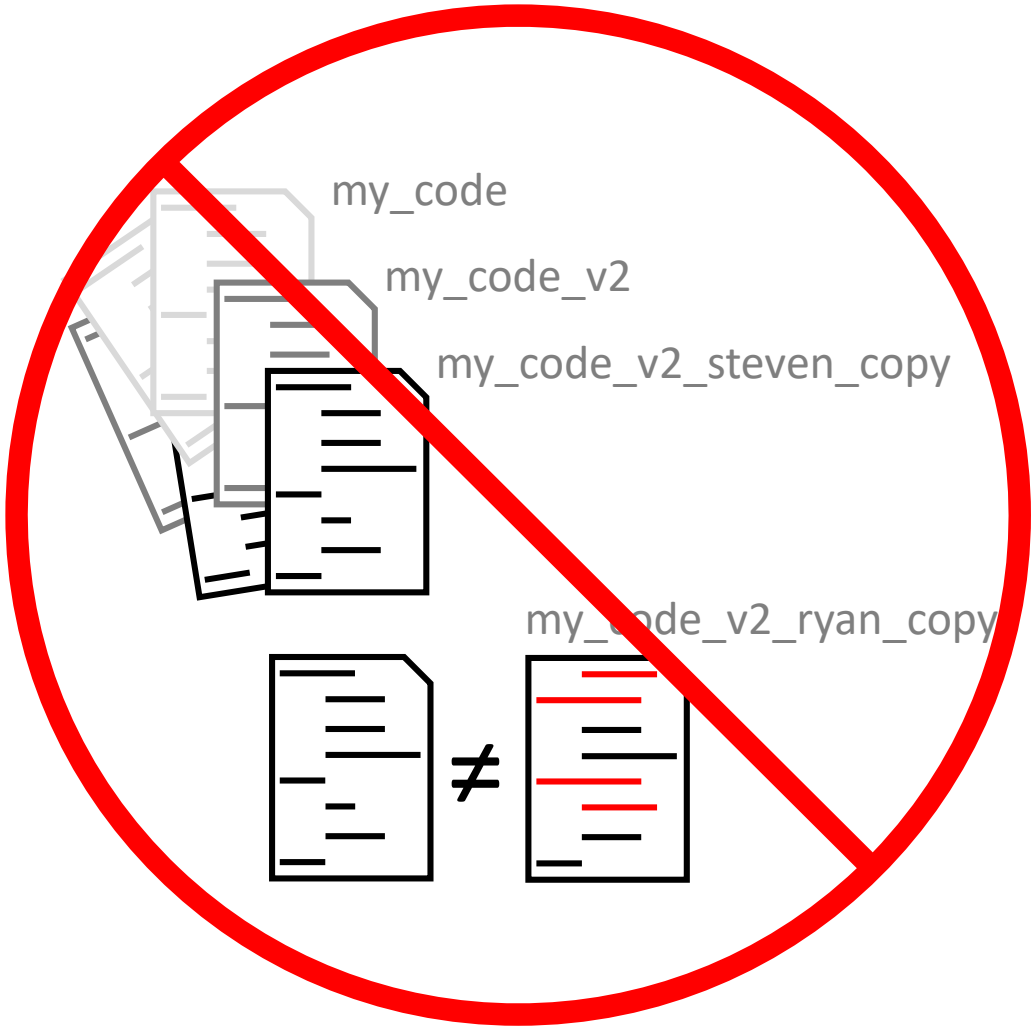
- Enables easy code sharing/collaboration
- Encourages “note taking” for changes committed
- Repositories can be on remote/cloud storage
- “*Unlimited Undo*”

Why use version control?

Helps keeps track of:

- **What** was changed?
- **Who** made a change?
- **When** a change was made?
- **Why** a change was made?

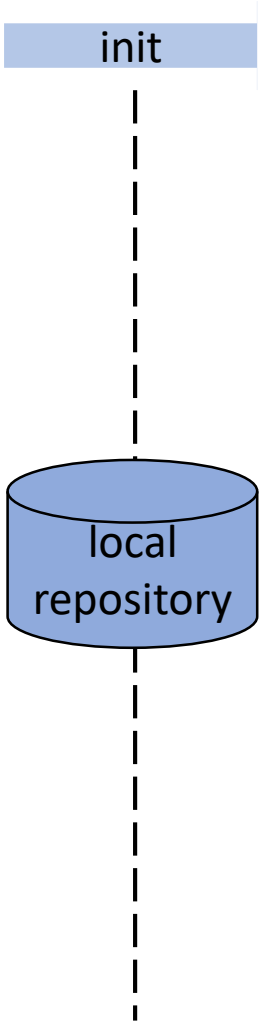
- Enables easy code sharing/collaboration
- Encourages “note taking” for changes committed
- Repositories can be on remote/cloud storage
- “*Unlimited Undo*”



Initialize a new repository

```
$ git init
```

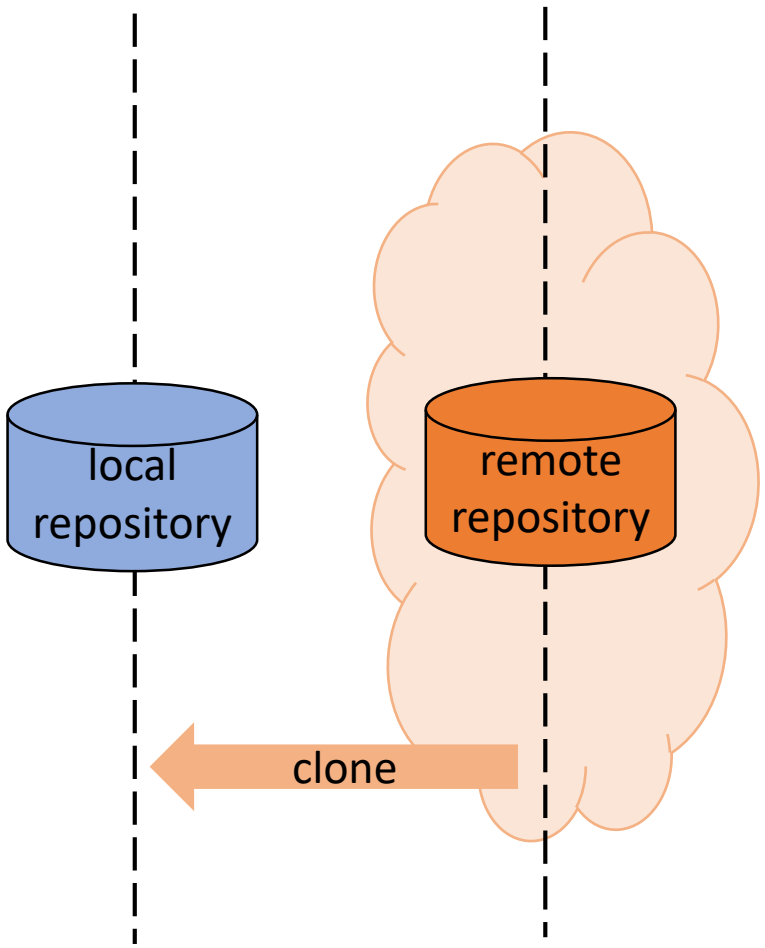
- Create a new “repo”
- Run ***init*** command within directory to be used
- ***.git*** hidden subdirectory (stores repo data, history, configuration)



Clone an existing repository

```
$ git clone
```

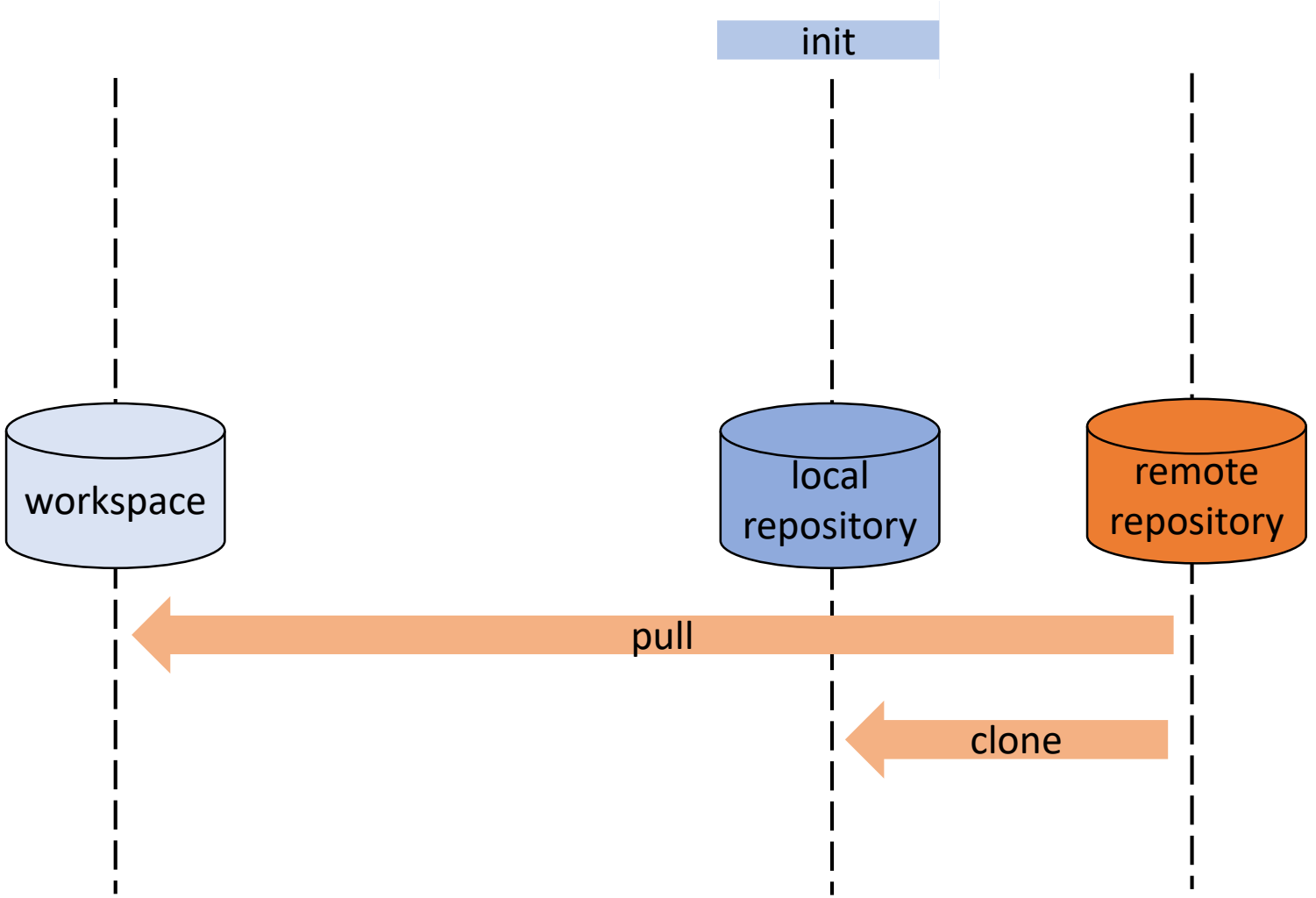
- Creates a local copy of the remote repository
- ***git clone*** over HTTPS



Pull from remote repository

```
$ git pull
```

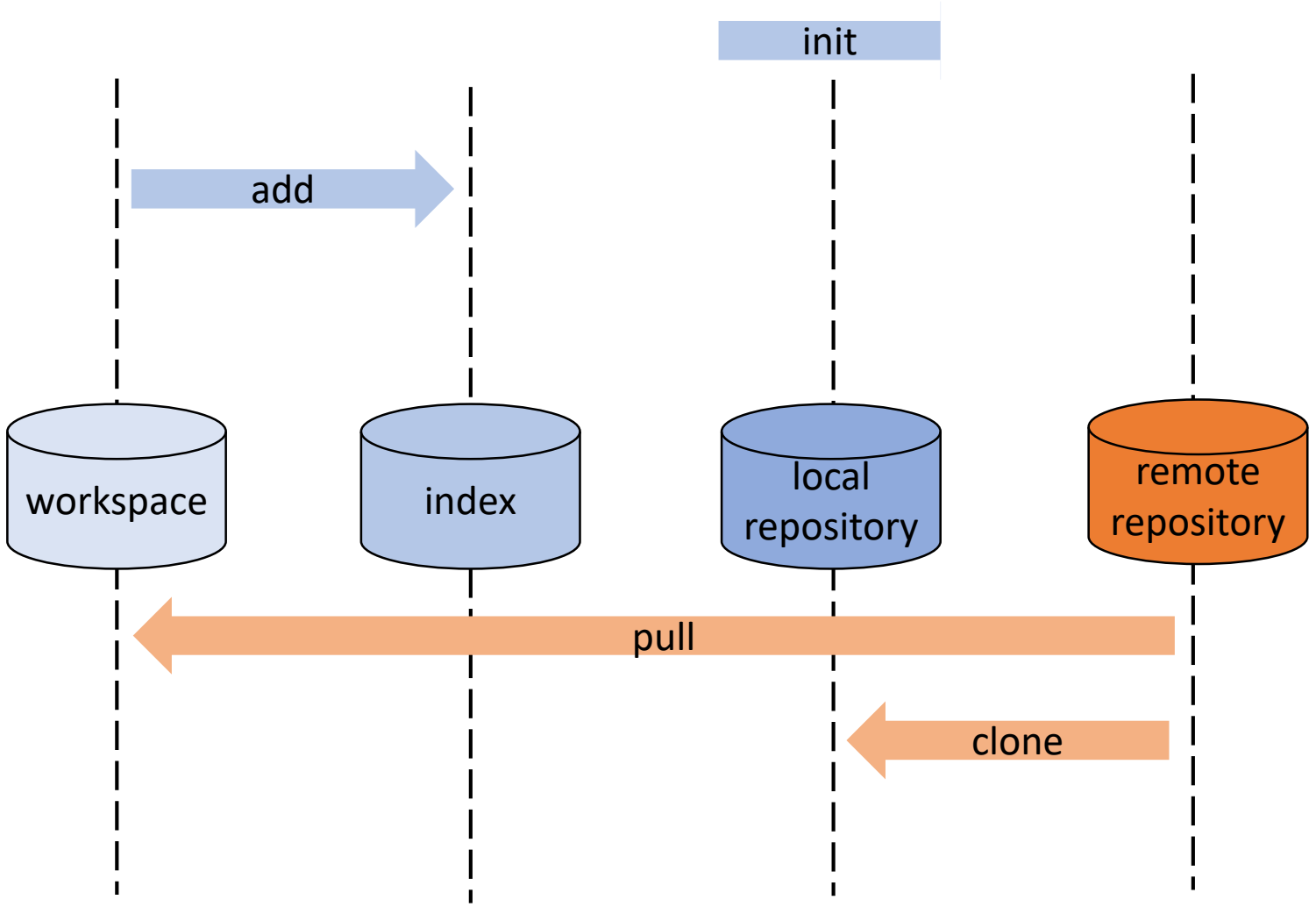
- Updates your current git workspace from the remote repository



Add changes

```
$ git add
```

- Record changes that have been made
- Tells Git to start tracking something
- Send to “staging area”




```
git add filename.txt
```

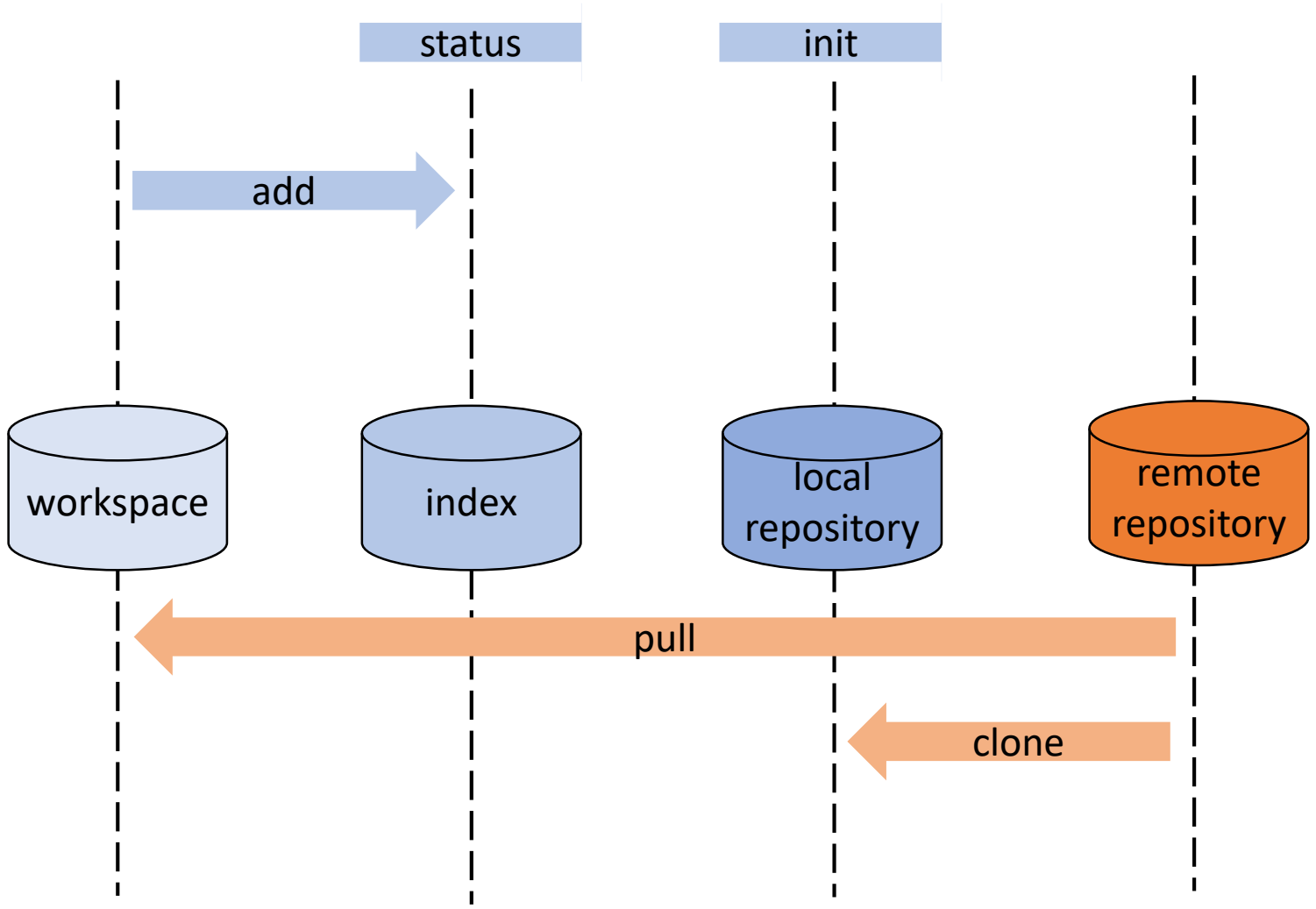
```
git add *.txt
```

```
git add *
```

Check status of changes

```
$ git status
```

- Is there a new file?
- Are there modifications to existing files?



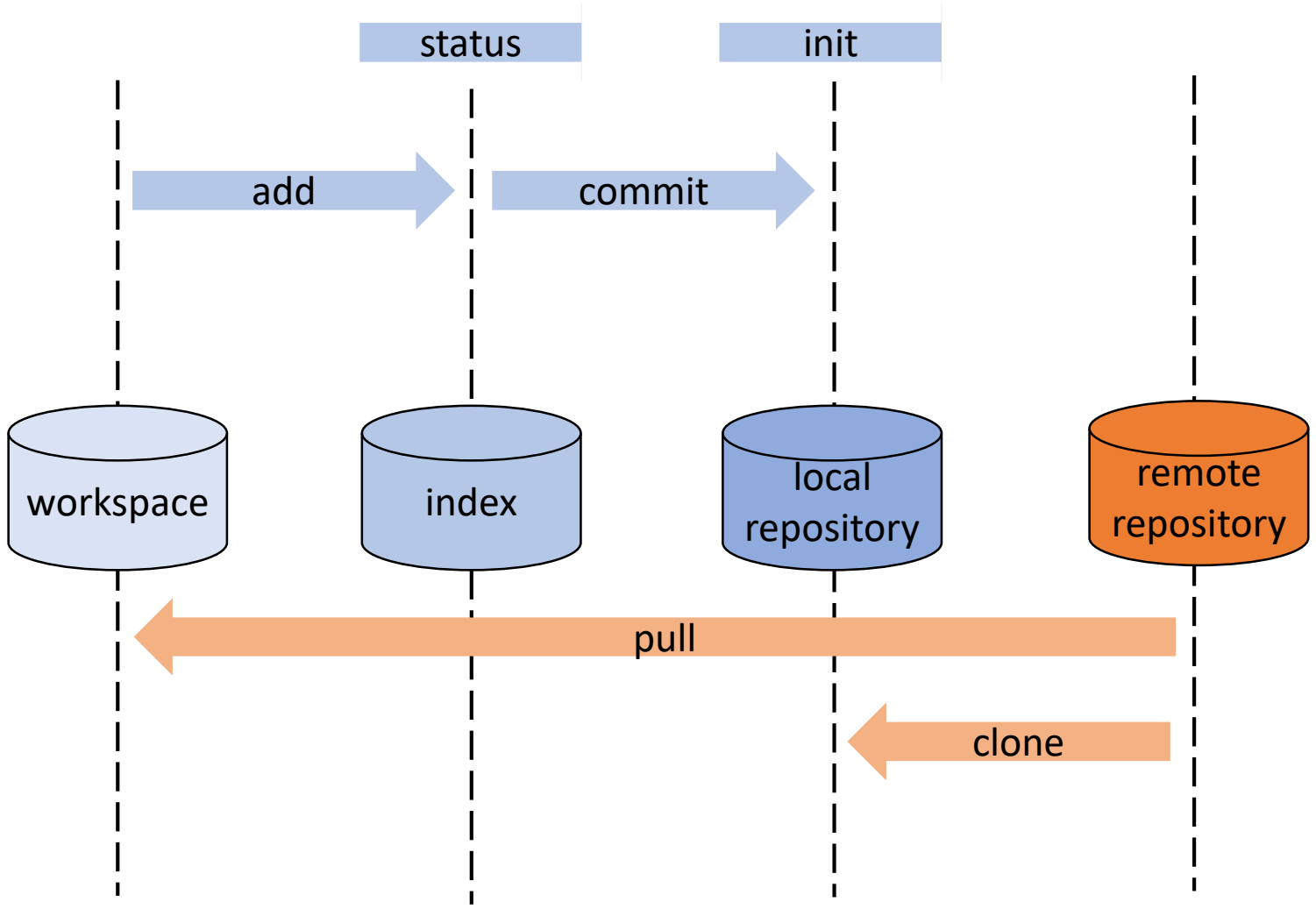
Commit changes to local repository

```
$ git commit
```

- Sends staged changes to local repository
- Record notes about changes made

```
git commit –m “Fix bug …”
```

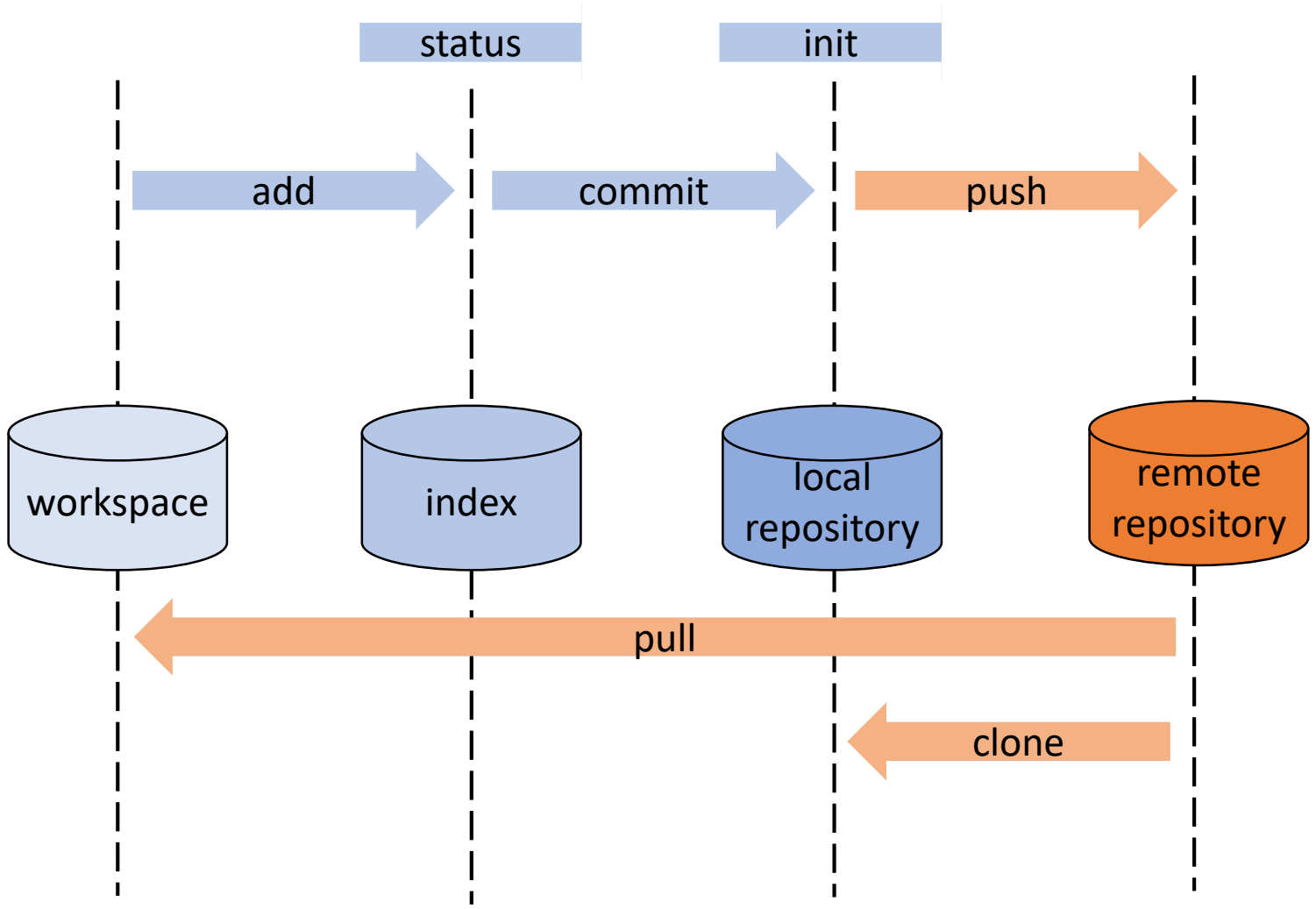
- Should be < 50 characters
- Start with a capitalized, present tense verb (Add, Fix, Remove, Refactor, Clarify, Change, Correct, …)



Push changes to remote repository

```
$ git push
```

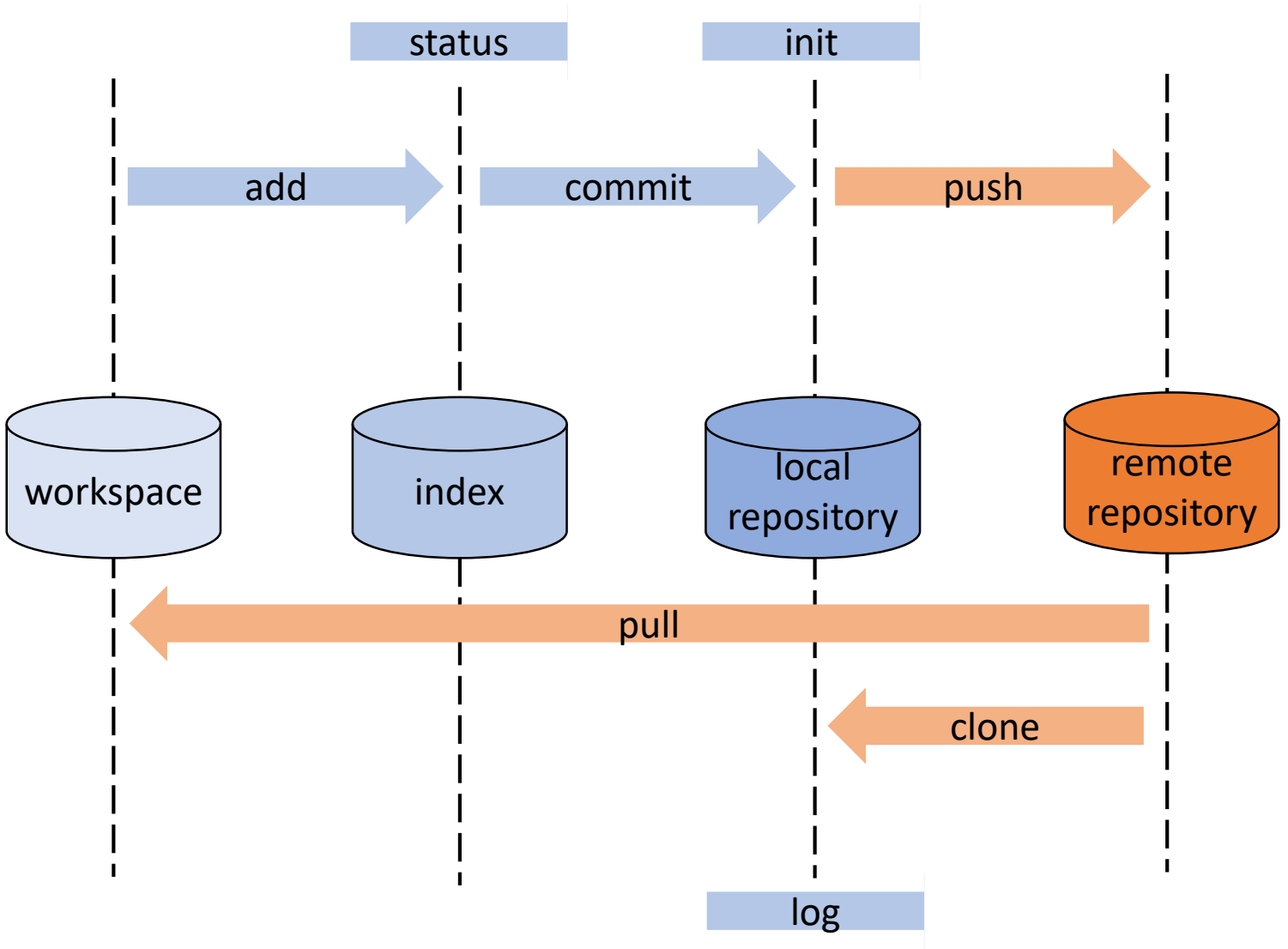
- Update remote repository with local repository (if you are the owner)



View commit log

```
$ git log
```

- shows all commits in reverse chronological order



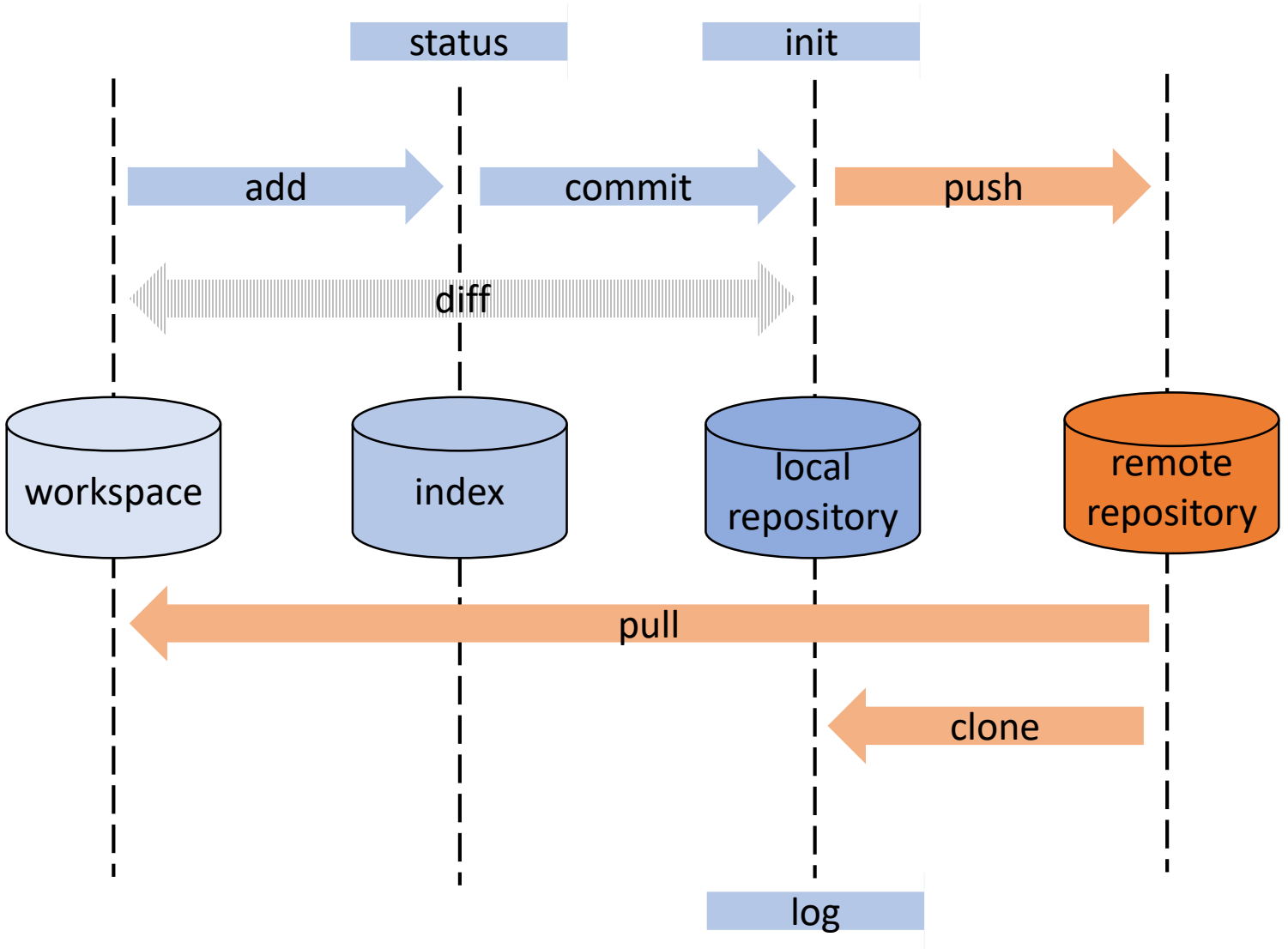
Compare previous versions

```
$ git diff
```

- Shows differences between the current file (workspace) and the most recent commit (repository)

```
git diff HEAD~n file.txt
```

- Shows diff between current file and n^{th} commit back



Checkout previous version

```
$ git checkout
```

- Revert to a previous version

```
git checkout HEAD file.txt
```

```
git checkout 5501ec5 file.txt
```