Version Control

BOOSTER
boost your career

Agenda

- Understanding Version Control
- Getting to Know Git
- Staging & Remote
- Cloning & Branching
- Rebasing
- History & Configurations

BOOSTER
boost your career

Understanding Version Control

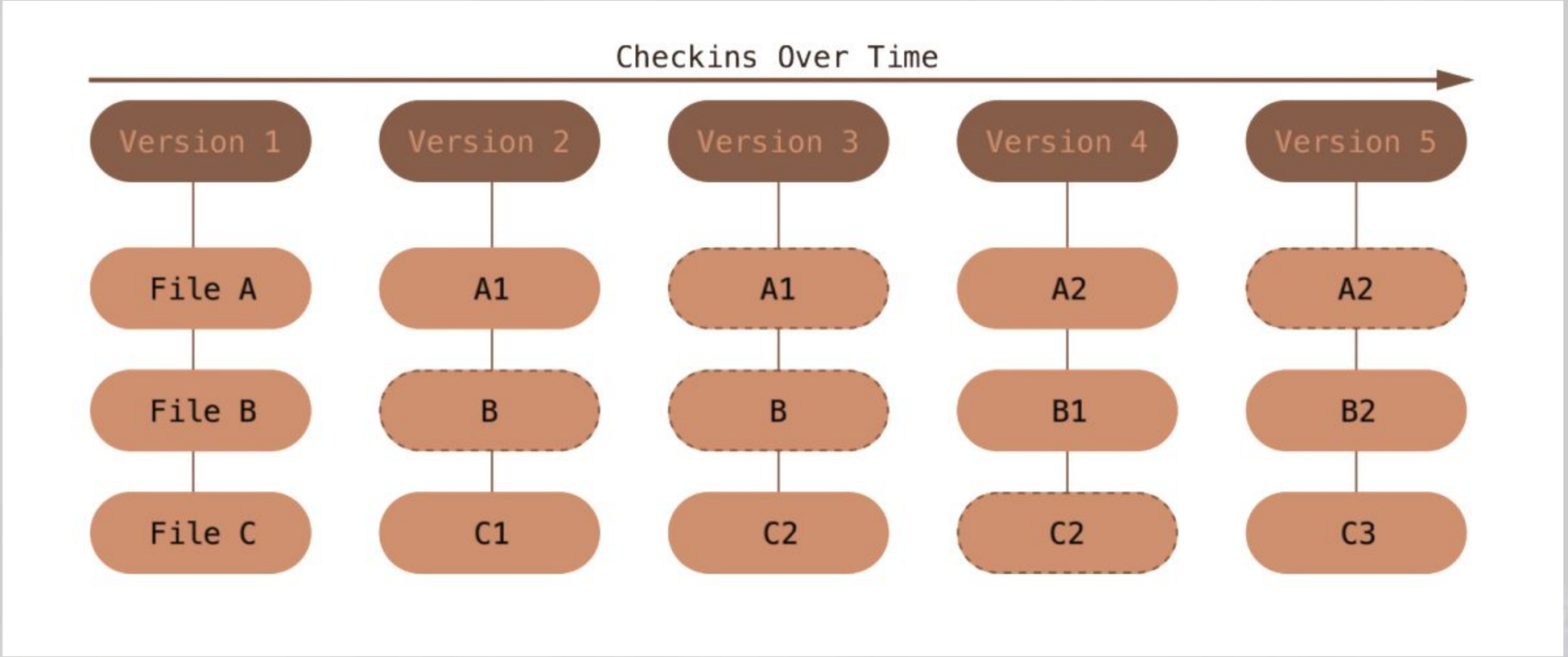
BOOSTER
boost your career

What is Version Control?

- Most VCS store information as a list of file-based changes. These systems think of the information they keep as a set of files and the changes made to each file over time..

- Git thinks of its data more like a set of snapshots of a miniature filesystem, Every time you commit in Git, it basically takes a picture of what all your files look like at that moment and stores a reference to that snapshot.

BOOSTER
boost your career



BOOSTER
boost your career

Every operation is local

- Most operations in Git only need local files and resources to operate – generally no information is needed from another computer on your network.

- For example, to browse the history of the project, Git doesn't need to go out to the server to get the history and display it for you – it simply reads it directly from your local database

BOOSTER
boost your career

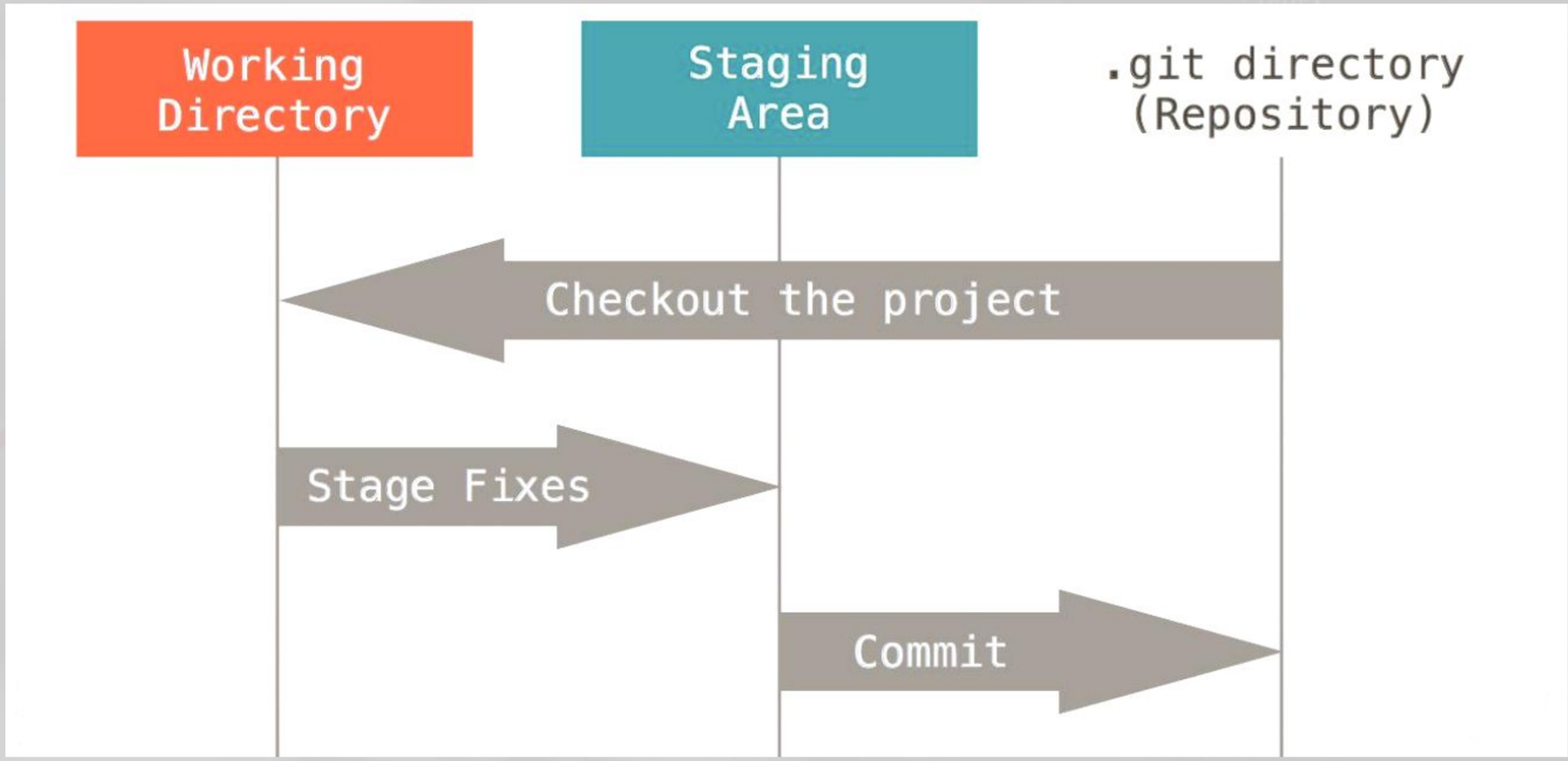
The three states

Git has three main states that your files can reside in:
committed, modified, and staged.

- Committed means that the data is safely stored in your local database.
- Modified means that you have changed the file but have not committed it to your database yet.
- Staged means that you have marked a modified file in its current version to go into your next commit snapshot

BOOSTER
boost your career

The three states



BOOSTER
boost your career

Installing Git

- CentOS

  sudo yum install git-all
- Ubuntu

  sudo apt-get install git-all

BOOSTER
boost your career

Git Setup

The first thing you should do when you install Git is to set your user name and email address.

```
git config --global user.name “user_name”
git config --global user.email “user_email”
```

This is important because every Git commit uses this information, and it’s immutably baked into the commits you start creating.

BOOSTER
boost your career

Getting help

If you ever need help while using Git, just use any of these commands:

- git help <verb>
- git <verb> —help

BOOSTER
boost your career

Starting a repo

- mkdir git_test
- cd git_test
- git init

Initialized empty Git repository in /home/user_name/git_test/.git/

- ls -a

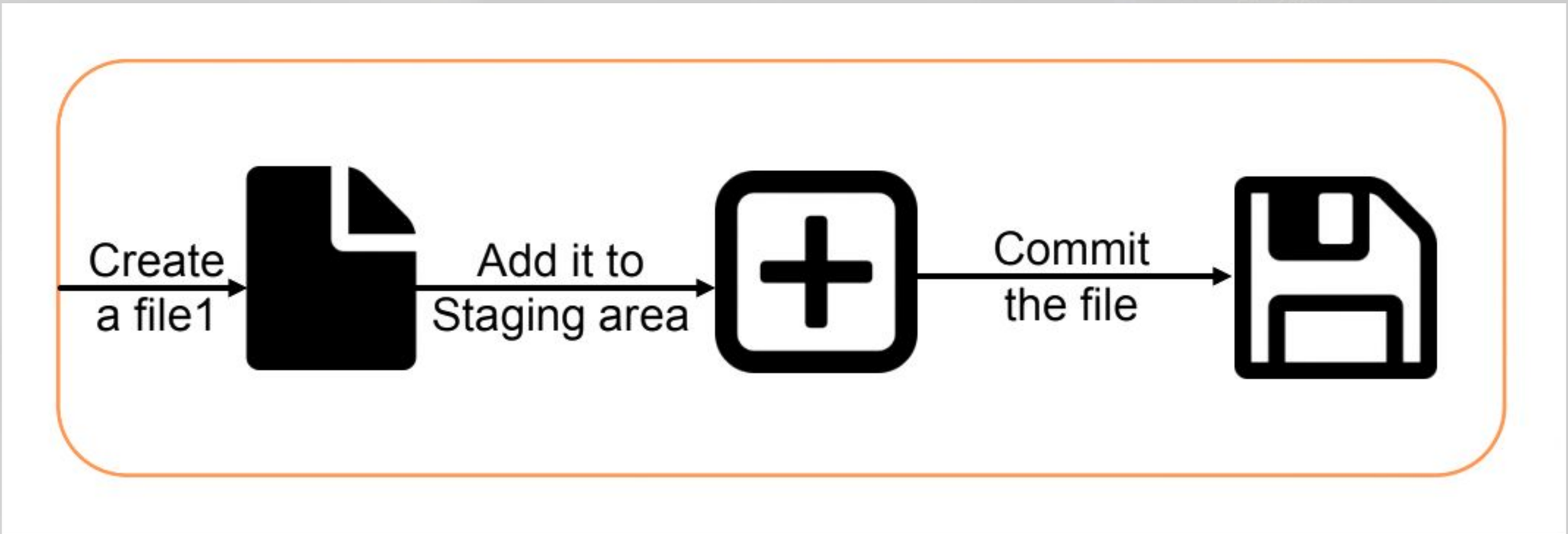
.git

Git directory is created that’s contains the meta data

BOOSTER
boost your career

Git Flow

Let’s assume the following scenario:



BOOSTER
boost your career

Simple practice

```
touch file1
git status
# On branch master
#
# Initial commit
#
# Untracked files:
# (use "git add <file>..." to include in what will
be committed)
#
# file1
nothing added to commit but untracked files present
(use "git add" to track)
```


BOOSTER
boost your career

Simple practice

```
git add file1
git status
# On branch master
#
# Initial commit
#
# Changes to be committed:
# (use "git rm --cached <file>..." to unstage)
#
# new file: file1
```


BOOSTER
boost your career

Simple practice

```
git commit -m "Create file1."
[master abe28da] Create file1.
1 files changed, 1 insertions(+), 0 deletions(-)
create mode 100644 file1

git status
# On branch master
nothing to commit (working directory clean)
```


BOOSTER
boost your career

Git log

- git log

commit 5acaf86b04aaf9cbbb8ebb9042a20a46d0b9ce76

Author: user_name <user_email>

Date: Thu Apr 7 22:00:46 2020 -0400

Add file1.

BOOSTER
boost your career

Git diff

- Show unstaged differences since last commit

  git diff
- Show staged differences since last commit

  git diff —staged

BOOSTER
boost your career

Unstaging files

- To unstage file

git reset HEAD file1

- Short Status

git status –s

M file1

BOOSTER
boost your career

Discard Changes

- git status –s

M file1

- git checkout –- file1
- git status

nothing to commit (working directory clean)

BOOSTER
boost your career

Undoing a commit

- Move to commit before ‘HEAD’ (last commit)

git reset --soft HEAD^

- Undo last 2 commits

git reset --soft HEAD^^

- Undo the last commit and all changes

git reset --hard HEAD^

BOOSTER
boost your career

Adding & removing a remote

- To add remote repository

  git remote add origin url
- To list the remote repositories

  git remote –v
- To delete or to rename a remote repository:

  git remote rm origin

  git remote rename origin neworigin

BOOSTER
boost your career

Pushing & Pulling from remote

- To add your files to the remote repo use this command (Note: you need to create an account on github.com)

  git push origin master

- To get the changes made by others

  git pull

- this command will:
  - Fetch (or Sync) our local repository with the remote one.
  - Merge the origin/master with master

BOOSTER
boost your career

Dealing with conflict

- git pull

remote: Counting objects: 5, done.

remote: Compressing objects: 100% (1/1), done.

remote: Total 3 (delta 1), reused 3 (delta 1)

Unpacking objects: 100% (3/3), done.

From https://github.com/user_name/gitSandbox

ee47baa..4e76d35 master -> origin/master

Auto-merging file1

CONFLICT (content): Merge conflict in file1

Automatic merge failed; fix conflicts and then commit the result

BOOSTER
boost your career

Dealing with conflict

- We have to edit the file and fix the conflict

<<<<<<< HEAD

This my line.

=======

No, it’s mine!

>>>>>>>

4e76d3542a7eee02ec516a47600002a90a4e4b48

- Then commit to save the merge, then the editor will pop up to enter your commit message

git commit -a

BOOSTER
boost your career

Cloning a Repo

- To clone a repo

  git clone repo_url
- This will:
  - Download the entire repository into a new gitSandbox directory.
  - Add the ‘origin’ remote, pointing it to the clone URL
  - get all branches

BOOSTER
boost your career

Branching out

- If you need to work on a feature that will take some time, it’s preferred to make a branch

git branch new-branch

- To list all branches:

git branch new-branch

* master

new-branch

- To switch to a branch:

git checkout newbranch

- Then you have to make you branch available remotely:

git checkout -b newbranch

git push origin newbranch

- To list the remote branches

git branch -r

BOOSTER
boost your career

Removing a remote branch

- To delete a remote branch

  git push origin :newbranch

  git branch -d newbranch

  git remote show origin

Remote branches:

master tracked

refs/remotes/origin/new-branch stale (use ‘git remote prune' to remove)

  git remote prune origin

BOOSTER
boost your career

Merging branches

• After finishing your work on the branch you have to merge it with master branch (assuming no changes were made to the master)

git checkout master

git merge newbranch

Updating 1191ceb..ab48a3f Fast-forward newfile | 1 + 1 file changed, 1 insertion(+) create

mode 100644 newfile

BOOSTER
boost your career

Tagging

- A tag is a reference to a commit (used mostly for release versioning)
- Types of tagging:
    - A lightweight tag is very much like a branch that doesn’t change – it’s just a pointer to a specific commit.
    - Annotated tags, however, are stored as full objects in the Git database. They’re checksummed, contain the tagger name, email, and date; have a tagging message.
- To create a lightweight tag.

  git tag v1.4
- To switch to a tag

  git checkout v1.4
- To push tags

  git push --tags

BOOSTER
boost your career

History and configuration

- To show the commits history

  git log
- To show the history in one line

  git log --pretty=oneline
- To show the log with changes

  git log --oneline -p
- To show the log with stats only

  git log --oneline --stat
- To See visual representation of the branch merging into master

  git log --oneline —graph
- play with time

  git log --until=1.minute.ago

  git log --since=1.day.ago

BOOSTER
boost your career

Blame

- Display the changes and committer name with each line of the file

git blame file1--date short

BOOSTER
boost your career

Ignore

- Often, you'll have a class of files that you don't want Git to automatically add or even show you as being untracked.
  - In such cases, you can create a file listing patterns to match them named .gitignore

```
cat .gitignore
cache/
logs/*.log
```


BOOSTER
boost your career

Untracking & removing files

- To stop tracking a file

  git rm --cached file3
- To delete a file

  git rm file2

BOOSTER
boost your career

Thanks

BOOSTER
boost your career