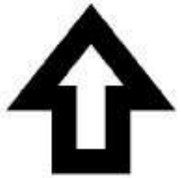
THE EPIC GIT 💣

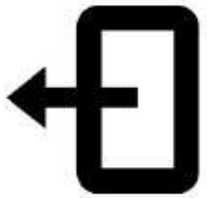
By Chen Hui Jing / @hj_chen

In case of fire

1. git commit

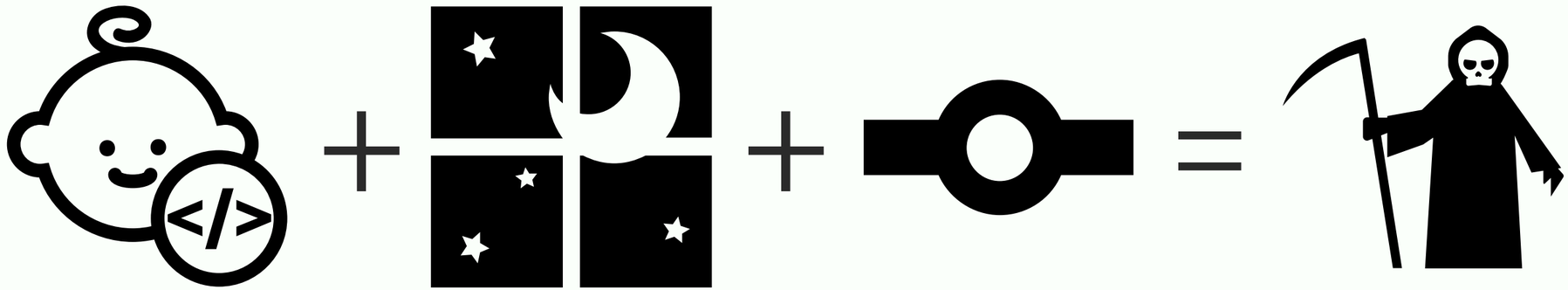


2. git push

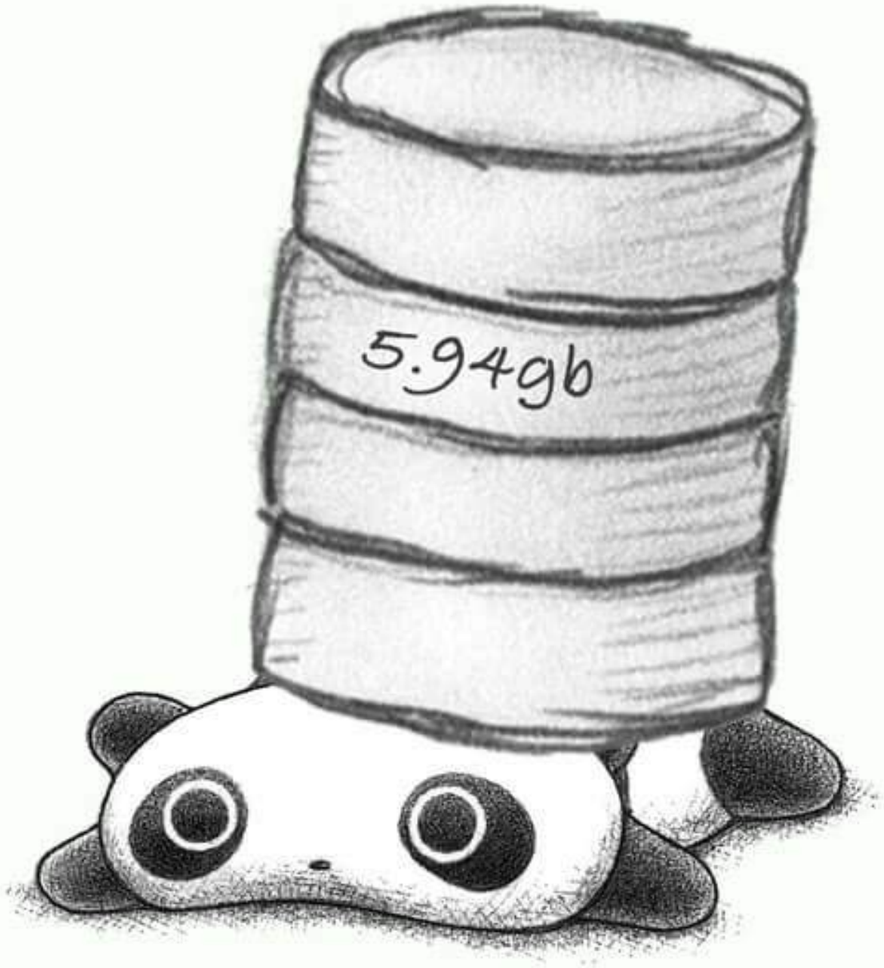


3. leave building

A LITTLE BACK-STORY



THE REPO DOES NOT FORGET



STEP 1: GIT FILTER-BRANCH

```
git filter-branch --index-filter 'git
```

```
git filter-branch --index-filter 'git rm -
rf --cached --ignore-unmatch FOLDER_NAME'
--prune-empty --tag-name-filter cat -- --
all
```

Hey, it's a pretty long line. ¯_(ツ)_/¯

`filter-branch --index-filter`

- The **`filter-branch`** command allows you to rewrite the Git history
- Can apply filters to modify information about each commit
- **`--index-filter`** rewrites the index
- Possible to use **`--tree-filter`** but **`--index-filter`** is faster because it does not check out the tree

```
'git rm -rf --cached --ignore-unmatch
```

- Used with `--index-filter` for optimal results
- Removes file(s) recursively (`rm`) and forcefully(`-rf`)
- `--cached` is used to unstage and remove paths from the index
- `--ignore-unmatch` will prevent the command from failing if the file is absent from the tree of a commit

```
--prune-empty --tag-name-filter cat
```

- `--prune-empty` allows the filter-branch command to ignore empty commits generated by the filters applied
- `-tag-name-filter cat` will update the relevant tags by rewriting them

```
-- --all
```

- `--` simply separates the filter-branch options from the revision options
- `--all` will rewrite ALL branches and tags
- Official documentation for `git filter-branch`

STEP 2: GIT PRUNE

```
git prune
```

- Prunes all unreachable objects from the object database
- Removes objects that are no longer being referenced
- Official documentation for `git prune`

STEP 3: REMOVE OLD REFERENCES

```
rm -rf .git/refs/original/
```

- Removes any old references to the unwanted folder/file
- Branches, remote-tracking branches, and tags are all references to commits
- All references are named with a slash-separated path name starting with "refs"

STEP 4: GIT REFLOG

```
git reflog expire --expire=now --all
```

- To manage reference log information
- `expire` is used to prune older reflog entries
- `--expire=now` specifies how far behind these older entries should be, in this case, right now
- Official documentation for `git reflog`

STEP 5: GIT GC

```
git gc --prune=now
```

- Cleans up unnecessary files and optimises the local repository
- `--prune=now` prunes objects older than the date specified, in this case, right now
- Official documentation for `git gc`

STEP 6: CLONE TO NEW REPO

```
git clone --no-hardlinks file://PATH/T
```

- Gives a "clean" repository with the history rewritten to remove target folder, with all other commits intact

HOUSTON, WE GOT A PROBLEM

- Force pushing these changes up to remote didn't work
- Create a new remote repository and push the clean repo up there
- Asked everyone to change their remote origin
- Only works if nobody pulled from the time you screwed up and the time you fixed things

TO FIND OUT MORE...

- BFG Repo-Cleaner
- Official Git documentation
- The Git Community Book

THE END

 http://www.chenhuijing.com

 @hj_chen

 @hj_chen

 @huijing