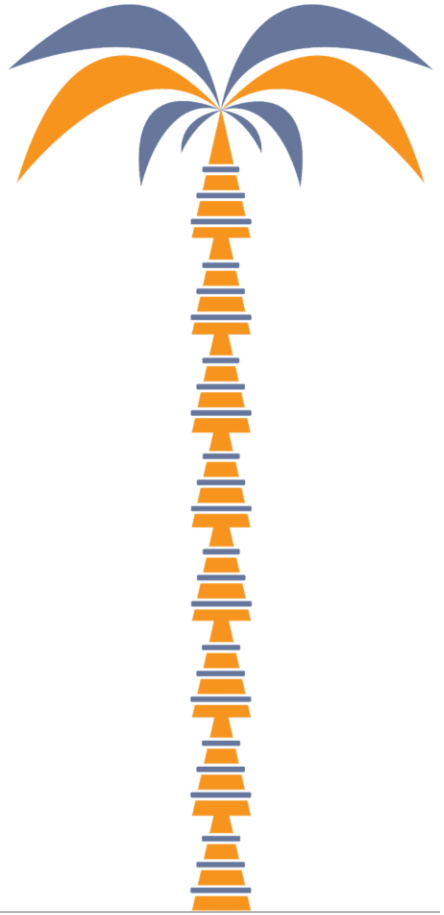


DevOps Bootcamp

Version Control with Git - Lecture #1

Dan Morgenstern

danm@sela.co.il

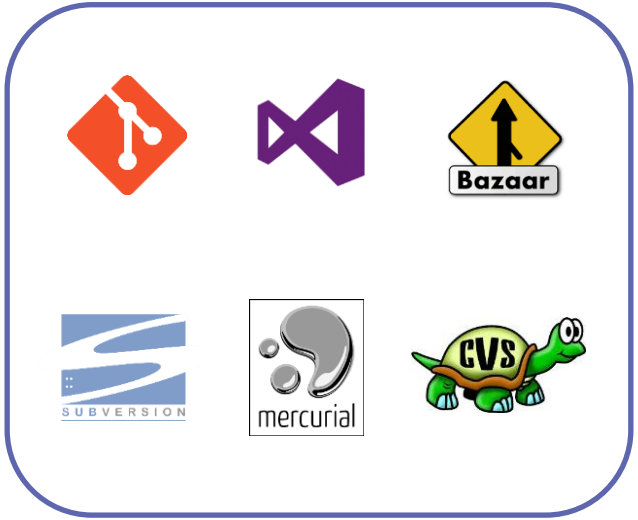
22/05/2022

(Please write "I am here (your name)" in the zoom chat to register your attendance)

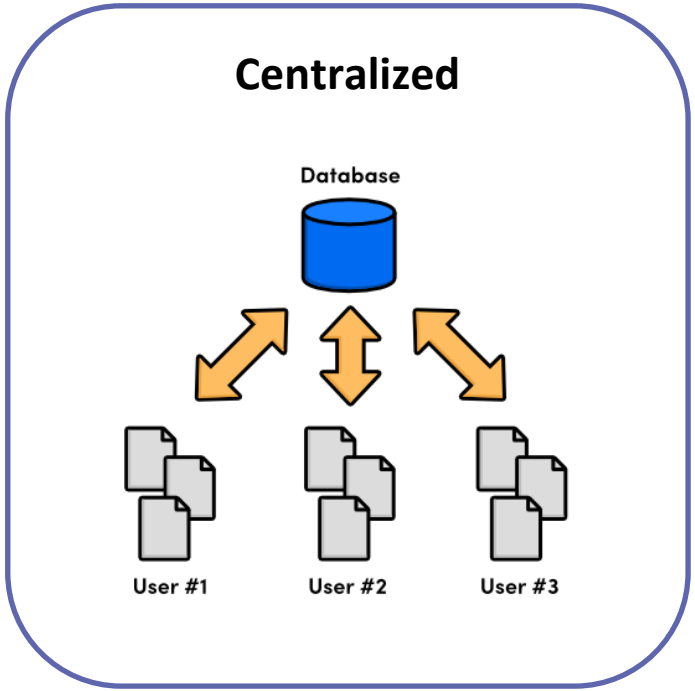
Version Control Systems

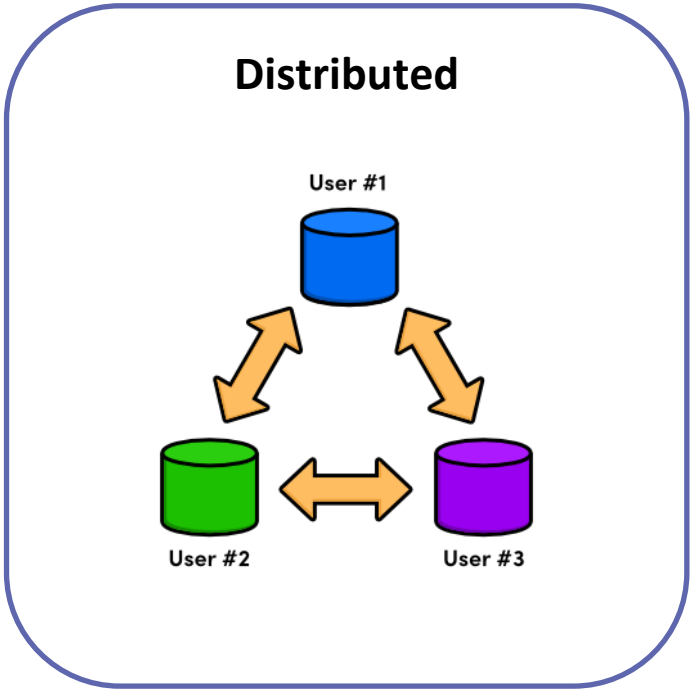
Version Control Systems (VCS)

- In general, is a kind of “database” which record changes to a file or set of files over the time.
  - Teamwork
  - Store Versions Properly
  - Show differences between versions
  - Restore previous versions
  - Understand project history
  - Backup

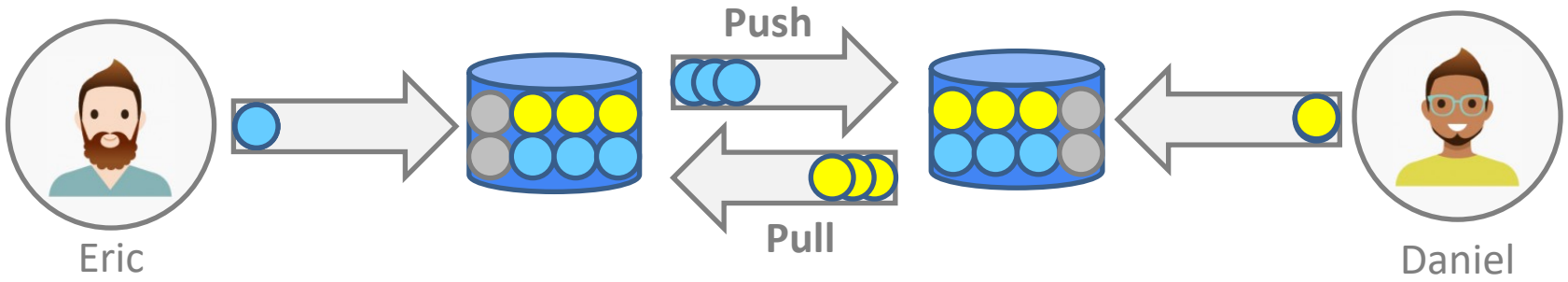


Centralized VS Distributed





Distributed VCS - Explained



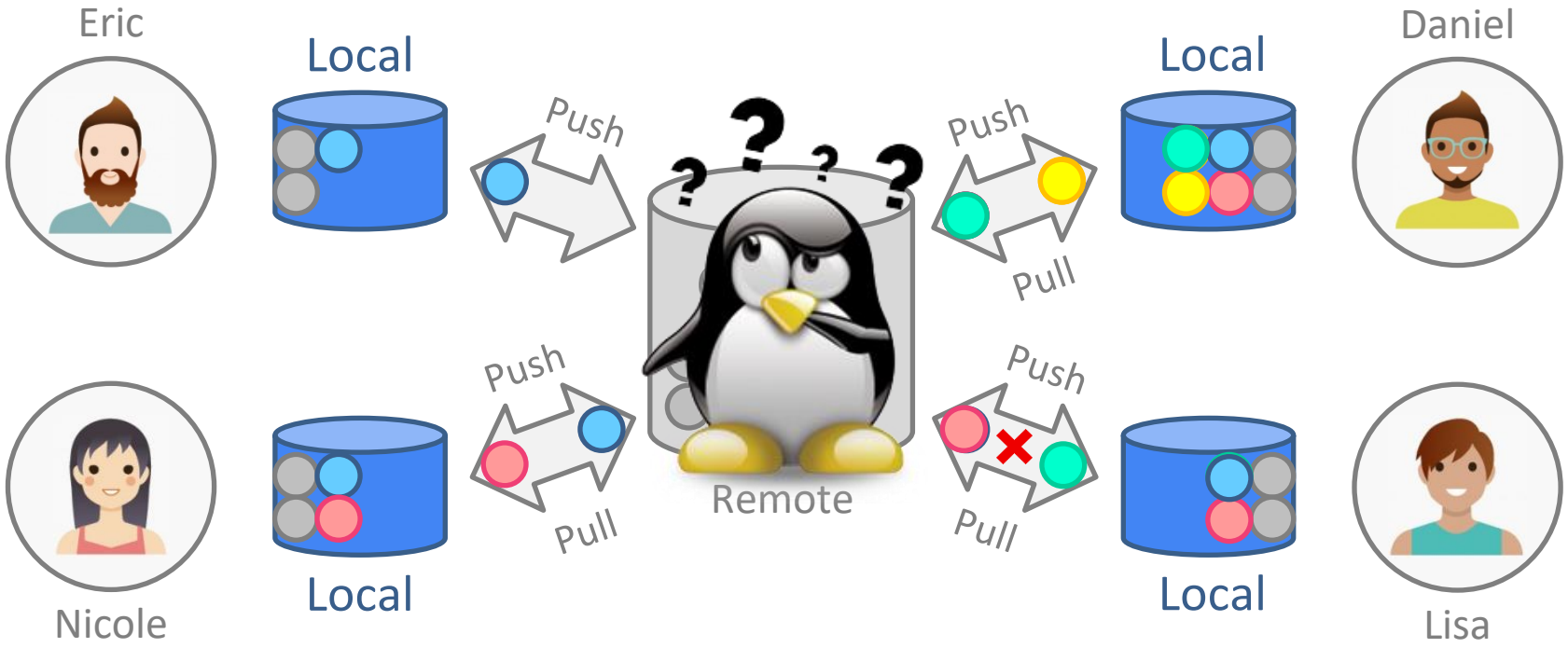
Introduction to Git

What is Git?

- The stupid content tracker
- Random 3 letter combination (not used in Unix)
- Stupid, contemptible and despicable (slang)
- Global Information Tracker
- Goddamn Idiotic Truckload of sh*t

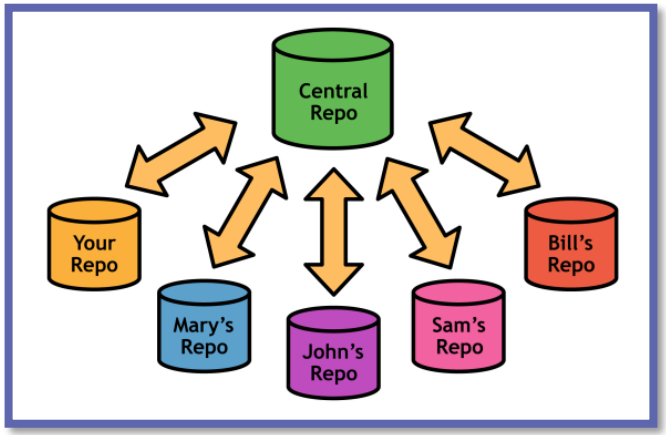
Git is a free and open source distributed version control system designed with performance, security and flexibility in mind

Git – Distributed but Centralized



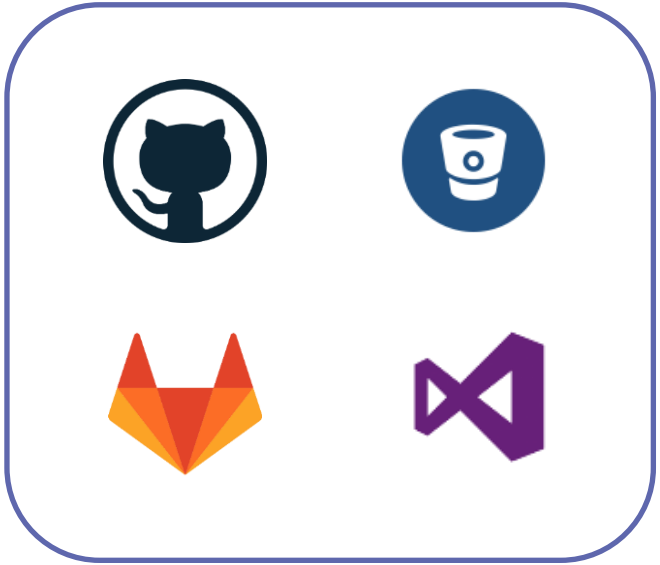
Git Server

- A Git server is just a machine that has Git installed that you and your team can push and pull changes from a Git repository.
- One Git Server can store several repositories.
- Central repositories are often “bare” (no working directory)



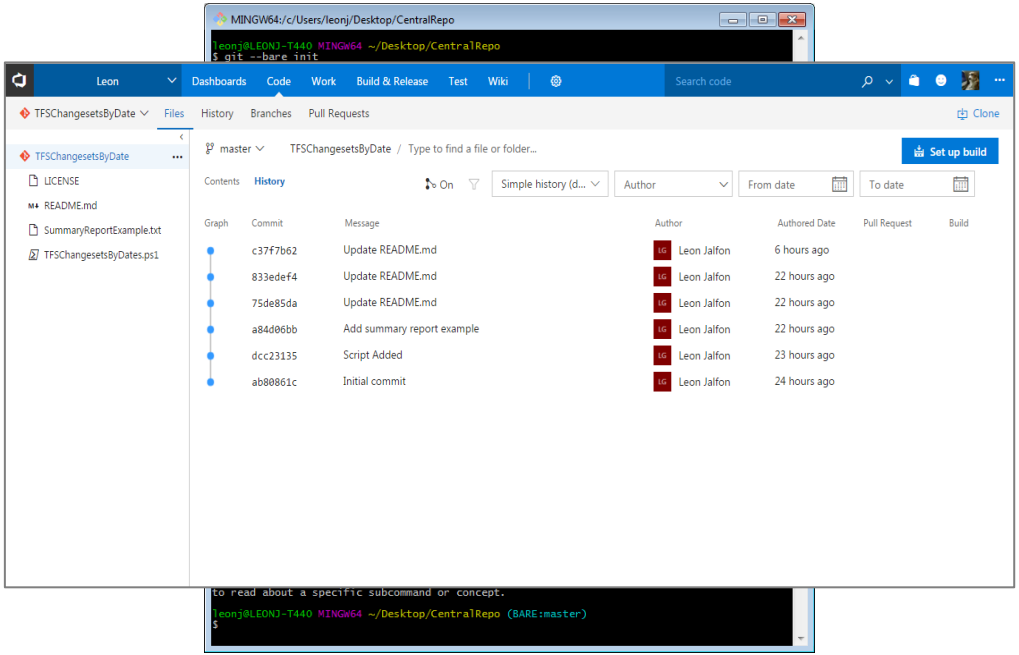
Repository Managers

- Instead of setting up your own server, you can also use a hosting service such as:
  - GitHub
  - GitLab
  - Bitbucket
  - Azure Repos
  - Perforce



Repository Managers

- Manage Security
- Manage Backups
- High Availability
- Manage Repositories
- Groups and Teams
- UI Management Tools
- Issue Tracking
- Code Review Process
- Integrations



Git Basics

- Git stores snapshots instead of deltas
- Each developer has a copy of the entire repository
- You can continue your work while been offline
- Branches are part of everyday development process
- Merging is central to Git (don't be afraid of conflicts)
- Git is based on the key-value model

Introduction Summary

Git is a free and open source distributed version control system designed with performance, security and flexibility in mind

Version Control with Git

Demo

Git Structure - Why do I need to understand it?

- Git is not designed to be user friendly

NAME

git-push - Update remote refs along with associated objects

NAME

git-rm - Remove files from the working tree and from the index

NAME

git-fetch - Download objects and refs from another repository

NAME

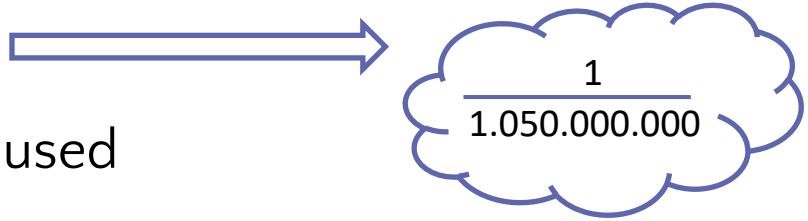
git-reset - Reset current HEAD to the specified state

Git Structure - Meeting the SHA1

- Is a hash function that convert an long string of data into a 40 character hexadecimal number

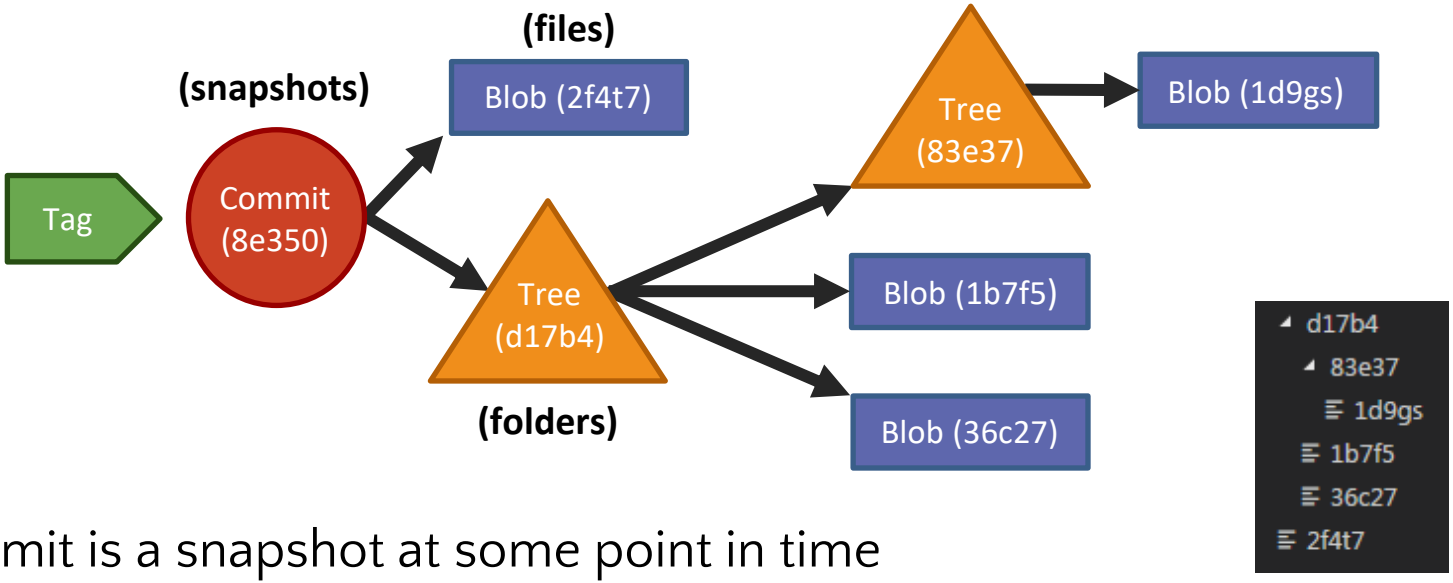
SHA1 = **e8964**2b96685d5f22ee7044e05b9e6566e69b7a5

- Every object in Git have its own SHA1 (used as key)
- Each SHA1 is unique (or almost)



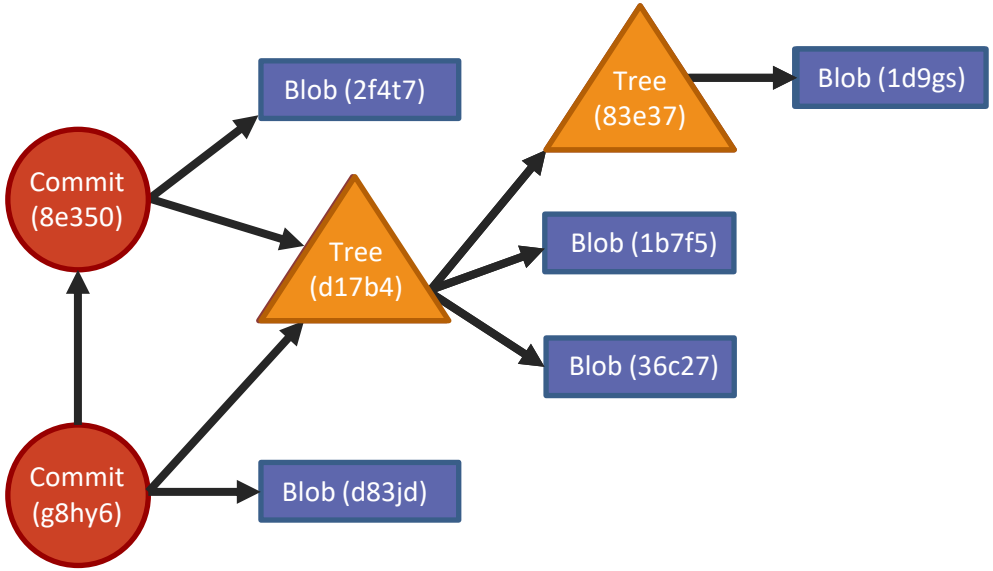
- Usually only the first 5 digits are used

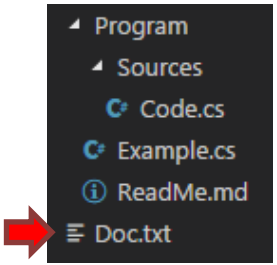
Git Structure - Objects

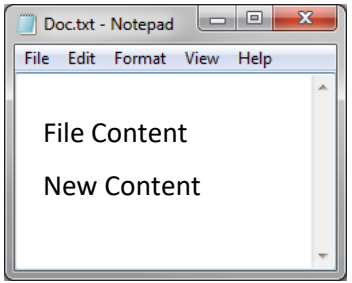


- A commit is a snapshot at some point in time
- A tag is a reference to a commit

Git Structure – Save Snapshots, Store Deltas



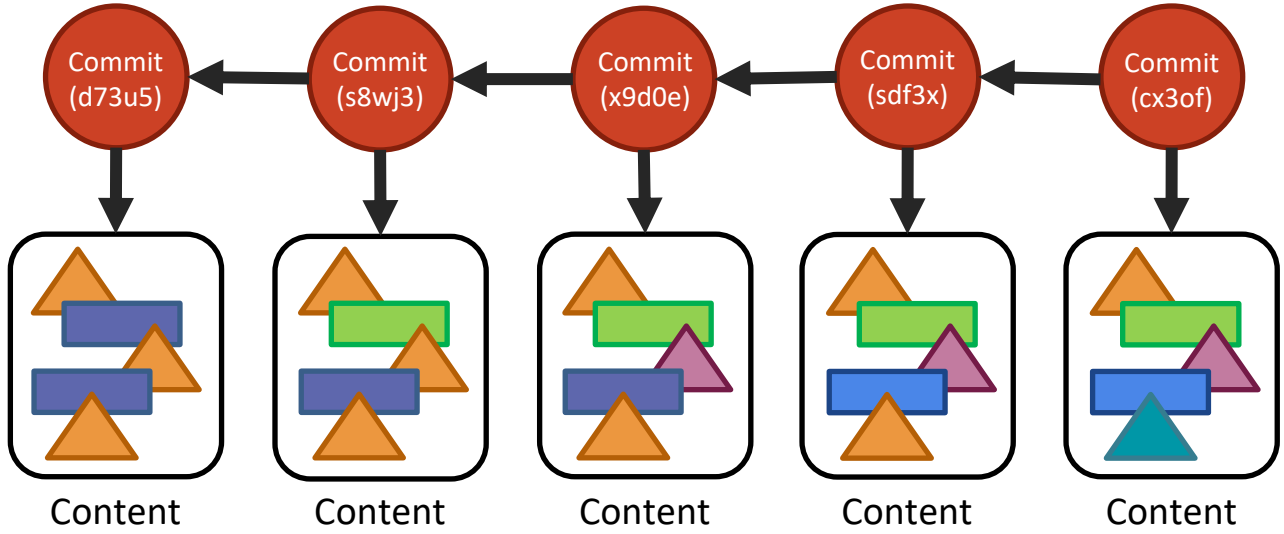




- Git creates objects only for updated content

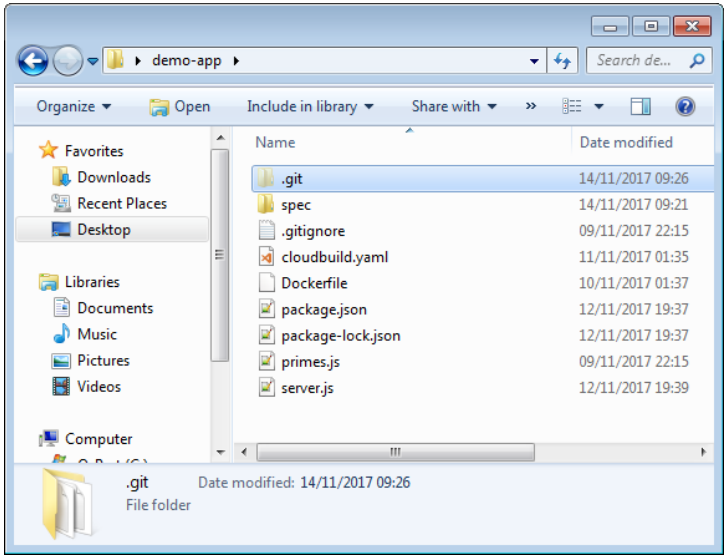
Git Structure - History

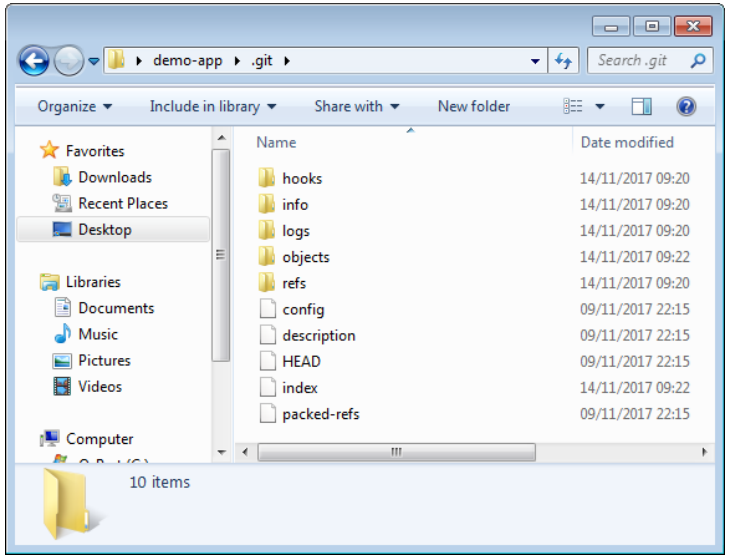
- The history is a set of interconnected commits



Git Structure – How things are stored

- The whole repository is stored under the `.git` folder

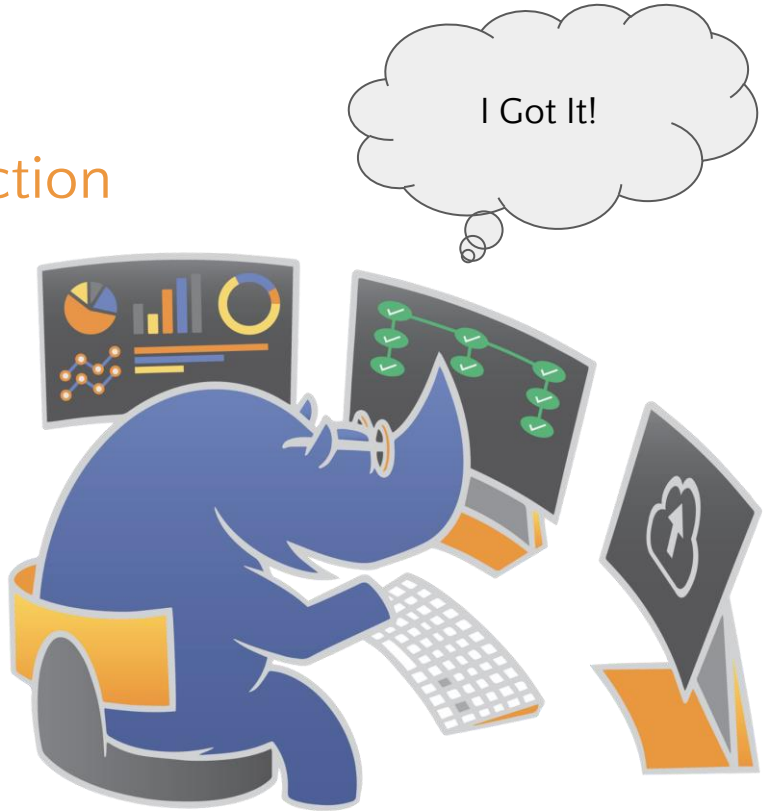




Questions

DevOps Bootcamp

Version Control with Git - Introduction



Natali Cutic
natalic@sela.co.il

Noam Amrani
noama@sela.co.il

Pushing YOU forward