Chapter 3 – SCM with GitHub

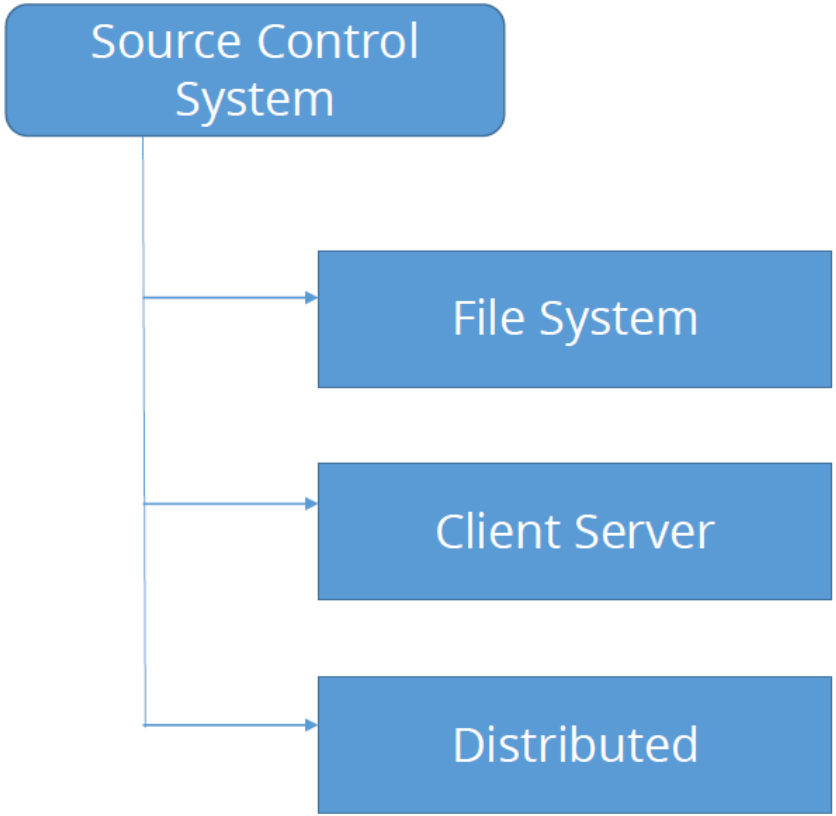
Learning Topics

- Introduction to SCM
- Overview of GitHub
- GitHub Terminologies
- GitHub Workflow

Source Control Systems

- Source control systems provide the necessary “grip” to allow you to stay in control
- Source control systems manage changes to documents so that their state is consistent
  - Also known as version control or revision control systems
- They can be used to store anything
  - They work best for storing changing text documents
  - They are usually associated with source code

Types of Source Control Systems

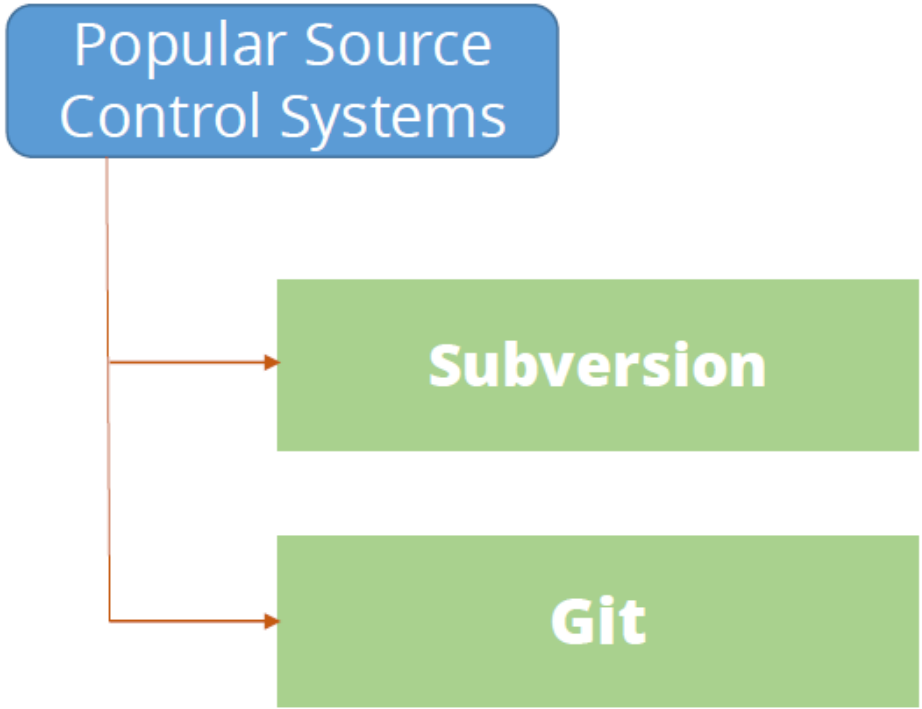


Distributed Source Control Systems

- Distributed Source Control Systems create replicas of the repository on each computer
  - Each user works on a full replica locally and can do so while disconnected from the network
- Are based on immutable snapshots of state with mutable tree structure
- Conceptually complex
- Are extremely fast
- Supports a wide variety of workflows
- Easy to use once standards are set

Popular Open Source Control Systems

SVN and Git are the two most popular source control systems



Git

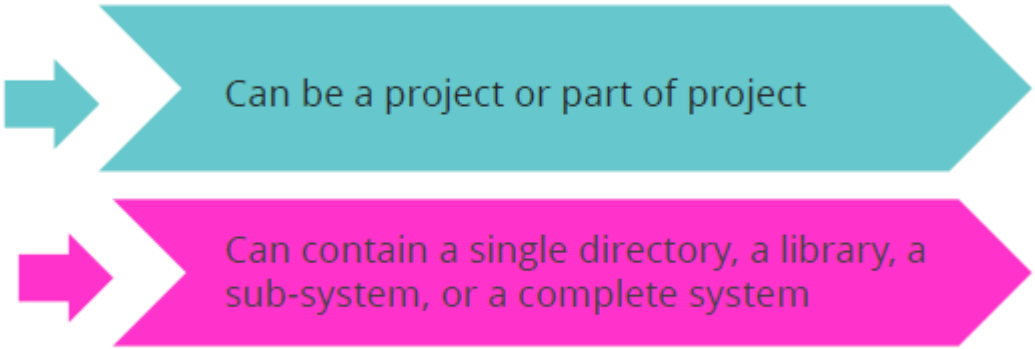
- Git was developed in 2005 by Linus Torvalds for Linux Kernel development
- Torvalds wanted a fast distributed open source version control system
- Nothing suitable existed so he wrote his own with a few criteria
  - It must be fast and be patchable in 3 seconds
  - It must do the opposite of what CVS does
  - It must protect against data corruption

Git Repositories

- There are several Git repositories for use on the Internet
- Github provides public and private repositories, and issue management
- Gitlab is similar but provides more project management features

Repositories

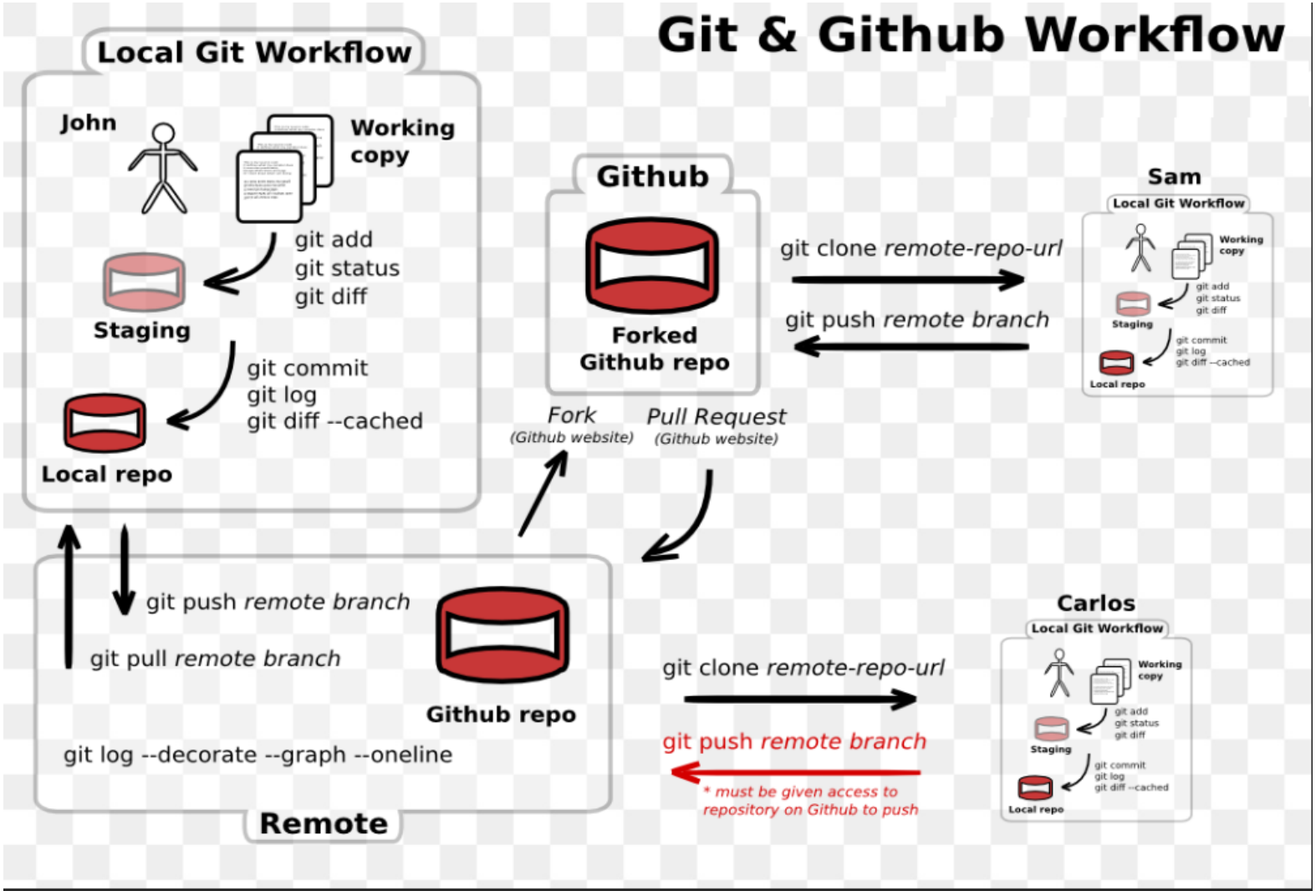
- Source control systems are organized into repositories or repos
- A repository is the unit of implementation and has to be explicitly created
- A repository can contain a single directory, a library, a sub-system, or a complete system



Assignment – Create Your Account on GitHub

1. Create your account on www.github.com
2. Remember credentials, we'll need them in labs

Git & Github Workflow



Lab 1 – Set Global Configuration Locally

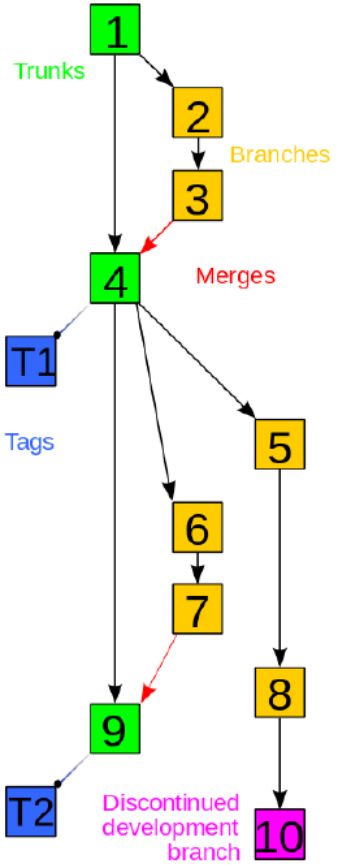
Lab 2 – Create and Clone Repository

Individual Assignment

1. Create a New Repository on GitHub.com
2. Perform Actions on Lab 2
3. Perform Actions on Lab 3
4. Perform Actions on Lab 4
5. Show New File on GitHub.Com

Repositories and DevOps

- Repositories are a key tool for DevOps management
  - They provide a mechanism to reach stability in an ever changing environment
  - Committed resources are immutable and stable
- Repositories can become very complex
- Branches and merges can have intricate relationships
- History always flows forward
  - Graph is a directed acyclic graph of state changes
- Unused branches should be deleted but many are afraid to do so
  - Nothing is truly deleted



Repositories Best Practices

A repository stores files and tracks changes

- Any file can be stored in a repository!
- Stored files which need to change over time

Inappropriate use of a repository

- Storing files which never change
  Example: a photograph or music library

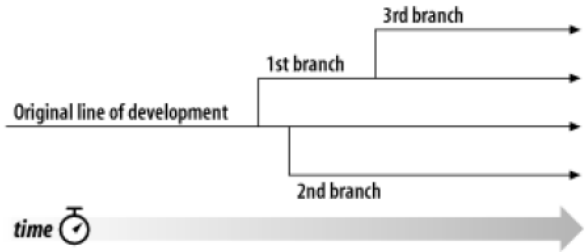
Better tools for replicating data which doesn't change

- rsync for directory synchronization
- TimeMachine for Apple computers
- Arq for "time machine" like backup into a Cloud storage location

Lab 3 – Change a File & Save Locally

Branches

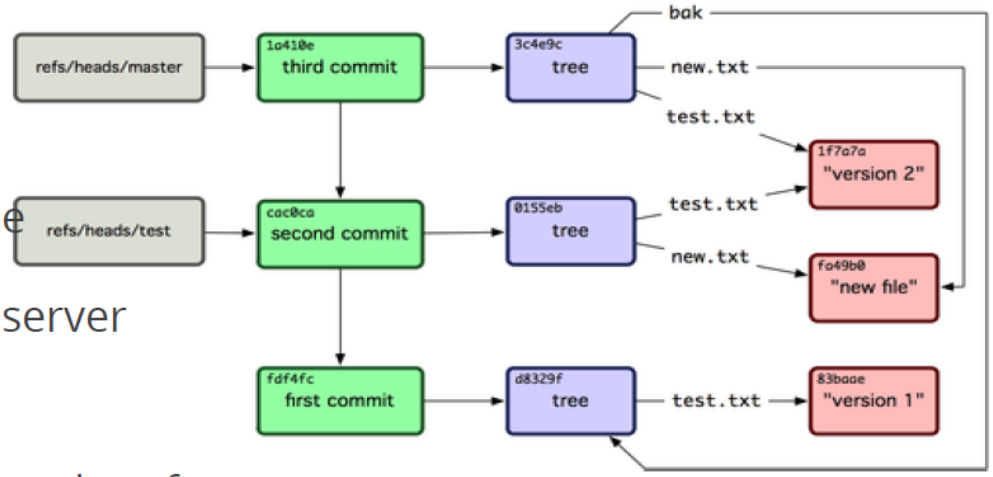
- Branches are when the the head of the repository is split for parallel development
  - The main branch is usually known as the trunk
- Branches are controversial and confusing
- There are many ways to use branches but fundamentally the rule is – create a branch when there is a change in policy
  - One branch is main code development
  - One branch is the code that passes all tests and is ready for release at any time
  - One branch captures a specific release (e.g. 1.1) and any bug fixes



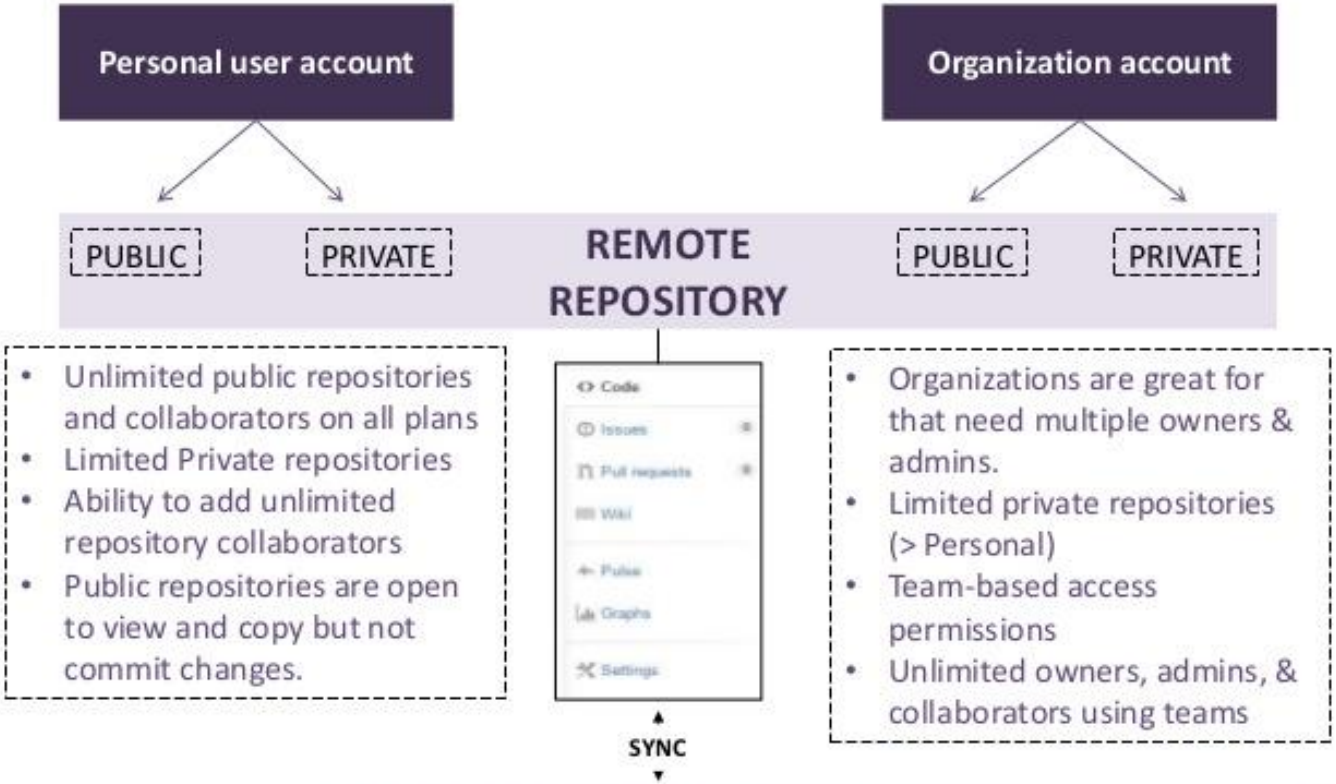
Lab 4 – Push Changes to Master from Local

Git Design

- Git is designed for frequent branching
  - A branch is just a reference to a single commit
- Merging is equally easy
- Each developer has a local copy of the repository
- Git can emulate CVS and Subversion server and can be accessed by CVS clients
- Git is fast and scalable and can be an order of magnitude faster than other systems



Github Structure



This concludes Chapter 3

Let us move to Chapter 4

CI-CD with Jenkins