Chapter 2 – SCM with GitHub

Learning Topics

- Launching Server in Cloud
- Introduction to SCM
- Overview of GitHub
- GitHub Terminologies
- GitHub Workflow

Lab – Launch Your Server in Cloud

- Navigate to …

https://bit.ly/2nhKGAQ

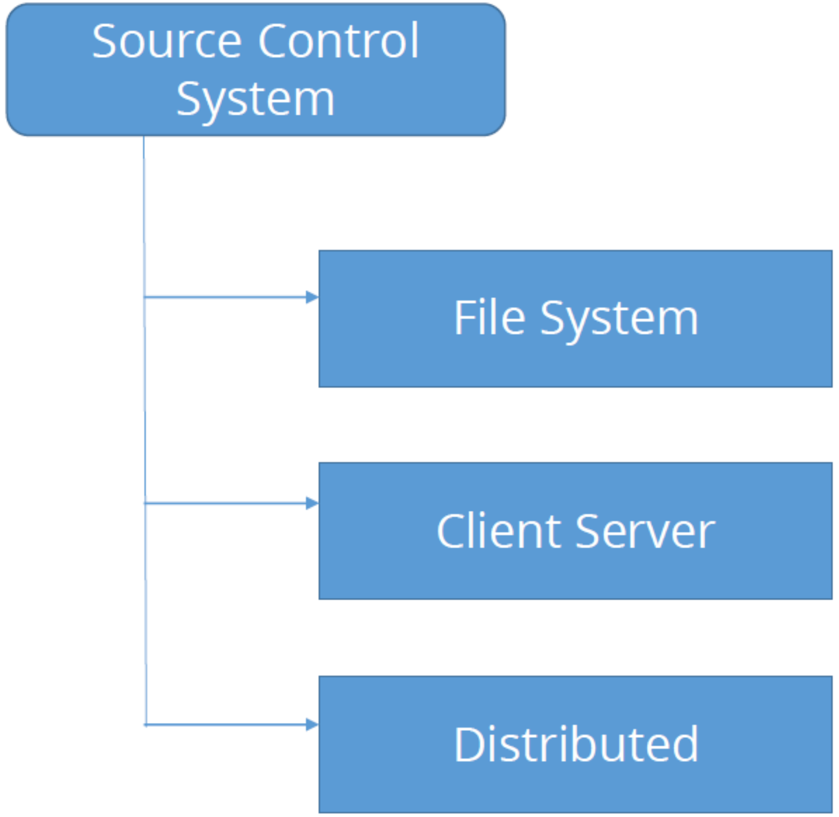
User-devops

Pwd-Dev0p$!!/

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

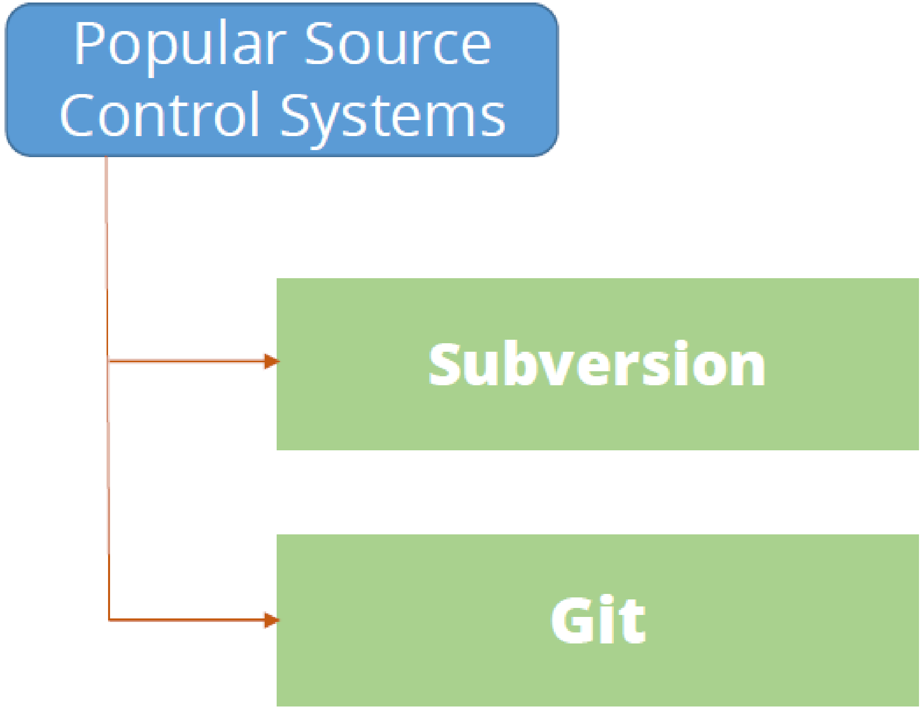


Distributed Source Control Systems

- 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

Can be a project or part of project

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
3. https://github.com/albertlovescloud/rildevopstraining

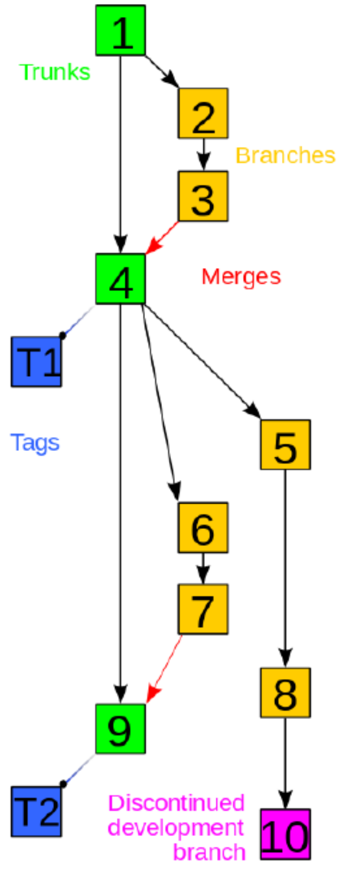
Lab 1 – Set Global Configuration Locally

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

Lab 2 – Create and Clone Repository

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

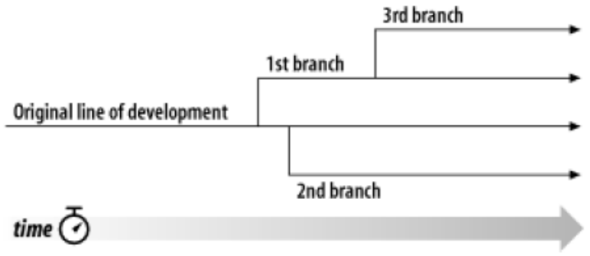
Better tools for replicating data which doesn't change

- 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

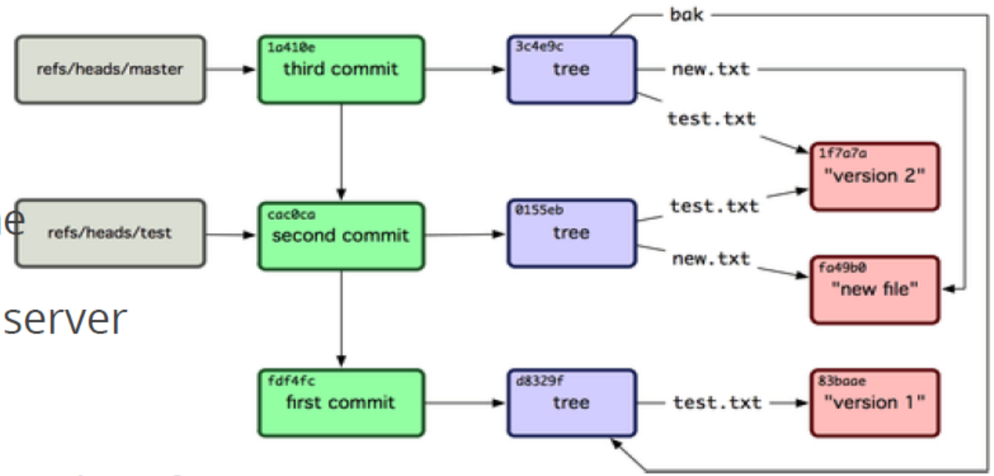
- Branches are when the the head of the repository is split for parallel development
    - The main branch is usually known as the trunk
- Branches are controversial and confusing
- There are many ways to use branches but fundamentally the rule is – create a branch when there is a change in policy
    - 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



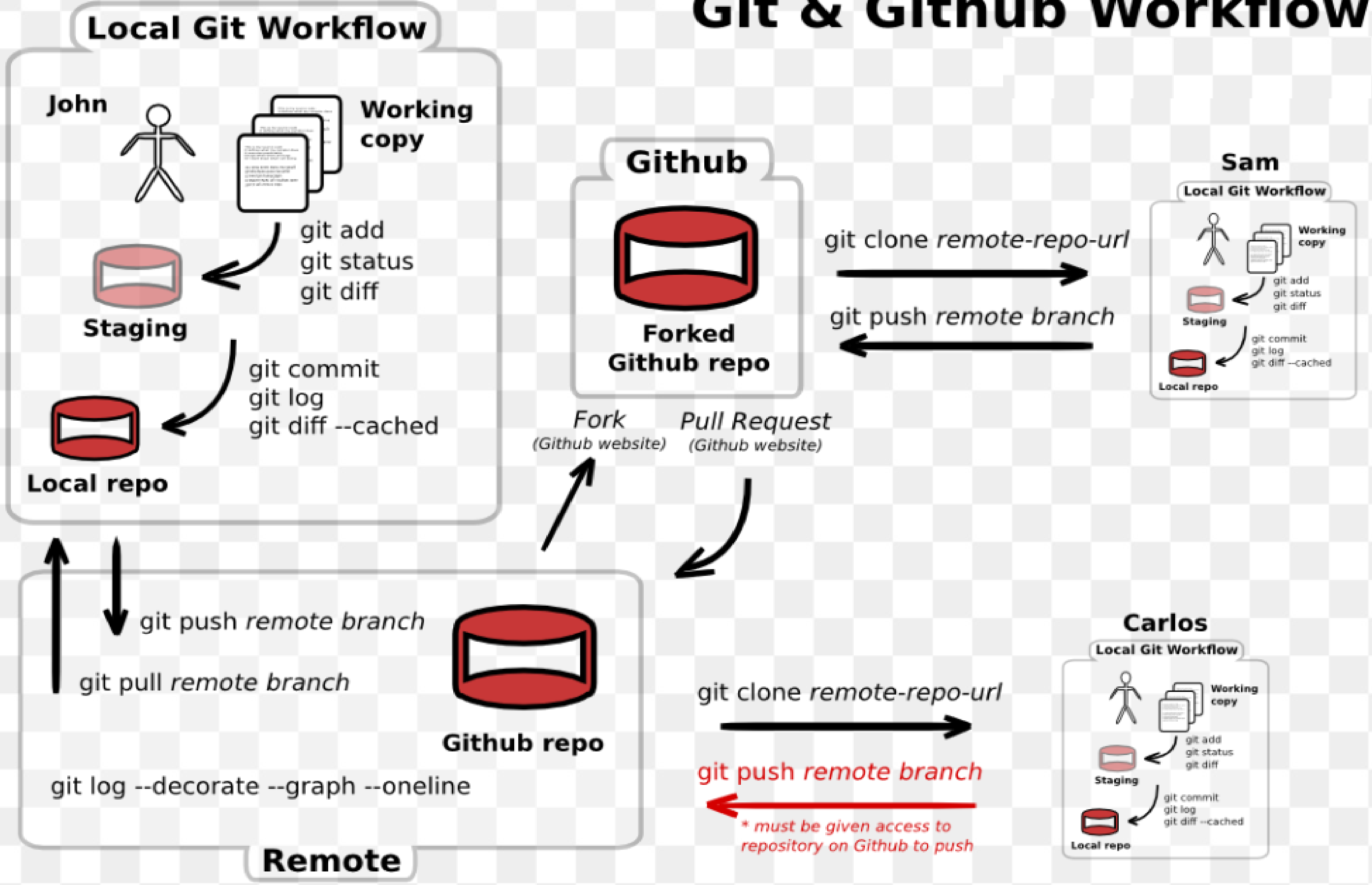
Lab 4 – Change a File & Save Locally

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



Git & Github Workflow

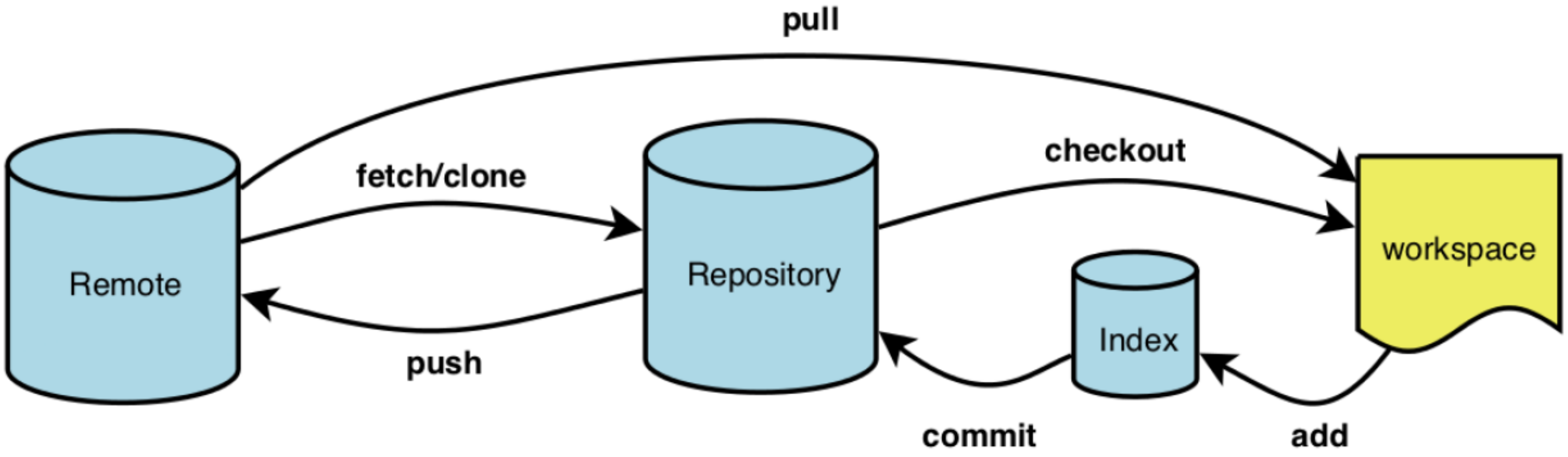


```
git add
git status
git diff

git commit
git log
git diff --cached

git clone remote-repo-url
git push remote branch

git push remote branch
git pull remote branch

git log --decorate --graph --oneline
```

Lab 5 – Push Changes to GitHub Repo

Git Workflow

- Repositories are shared by a clone operation
- Local changes can be undone using checkout
- Workflow is pull-edit-add-commit-push



Lab 6 – Pull Request & Merge

Lab 7 – Collaboration

Group Assignment – GitHub

1. Team of 4-5
2. Team Leader and Team Members
3. Team Leader to create repo
4. Add Team Members as Collaborators
5. Team Members to clone repo
6. Create their own branch (use your name)
7. Create New File
8. Push to Their Own Branch
9. Create Pull Request
10. Team Leader to Merge

Conflicts

- When two user changes overlap, it is called a conflict
- When second user merges, a conflict will be flagged
- Resolving conflicts can be difficult
  - Software can't do this automatically
  - Can see both sets of conflicting changes
- Must be manually resolved, with discussion
- Copy-Modify-Merge advantages outweigh conflict problems
  - Resolving conflicts is faster than waiting for locks
- The worst case scenario is that the conflicts are so bad that one person has to reapply changes to the latest version

Github Structure

Personal user account

PUBLIC

PRIVATE

Organization account

PUBLIC

PRIVATE

REMOTE REPOSITORY

- Unlimited public repositories and collaborators on all plans
- Limited Private repositories
- Ability to add unlimited repository collaborators
- Public repositories are open to view and copy but not commit changes.

SYNC

- Organizations are great for that need multiple owners & admins.
- Limited private repositories (> Personal)
- Team-based access permissions
- Unlimited owners, admins, & collaborators using teams

CLONE TO GET LOCAL REPOSITORY

This concludes Chapter 2

Let us move to Chapter 3

<u>CI-CD with Jenkins</u>