Coding

ERIN KEITH

CS 491 – TESTING AND DEVOPS

Pipeline

- Coding
  - Code development and reviews
  - Source code management
  - Code merging
  - *not IDEs (personal development environment)
- Building
  - Continuous integration tools
  - Build status
- Testing
  - Continuous testing tools

Source Control

AKA

- Version Control
- Revision Control
- Source Code Management

A system that records changes to a file or set of files over time.

Generally changes include

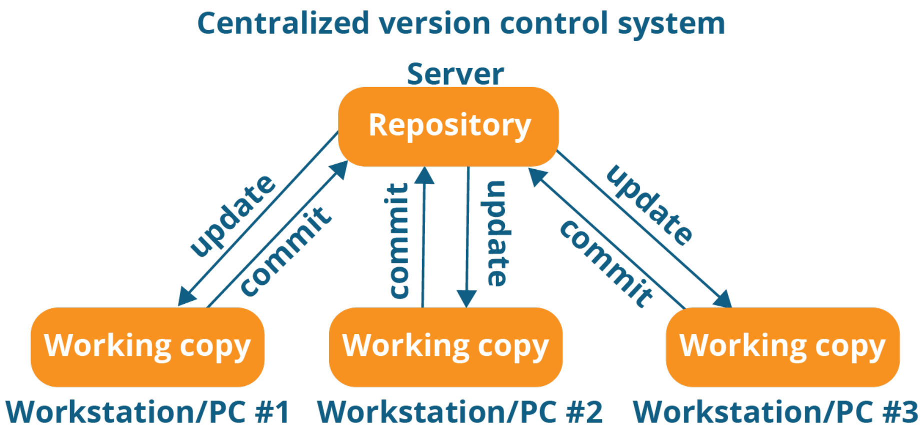
- timestamp
- author
- ID number

A change may also include

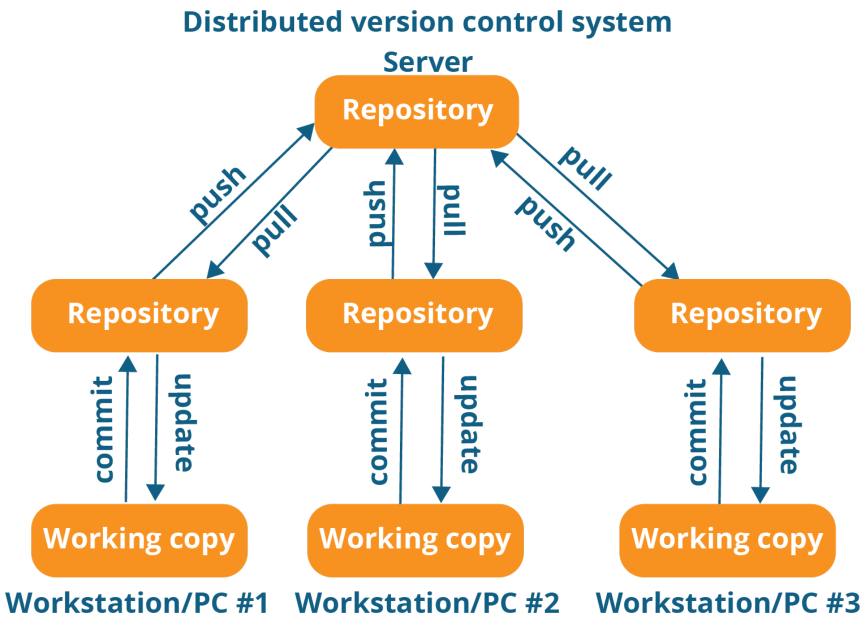
- the difference

Source Control - Models

SERVER-CLIENT



DISTRIBUTED



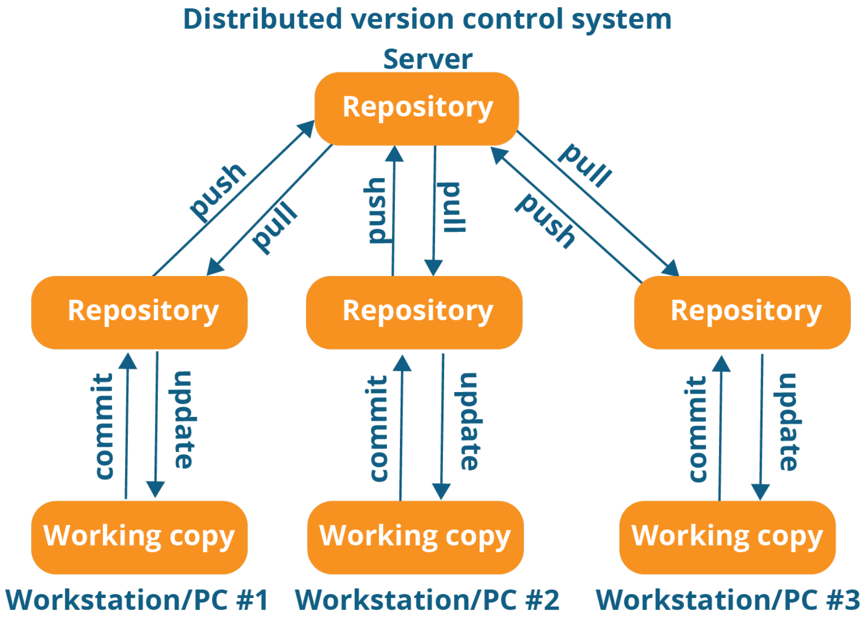
https://www.edureka.co/blog/what-is-git/

Source Control - Models

- Operations (except push & pull) are very fast
  - accesses the hard drive, not a remote server
  - internet connection not necessary
- Committing new change-sets can be done locally without manipulating the data on the main repository
  - once you have a group of change-sets ready, you can push them all at once
- Every contributor has a full copy of the project repository
  - share changes for feedback before affecting changes in the main repository
  - If the central server crashes, the lost data can be easily recovered from any contributor's local repositories.

DISTRIBUTED

git



https://www.edureka.co/blog/what-is-git/

Source Control - git

commands

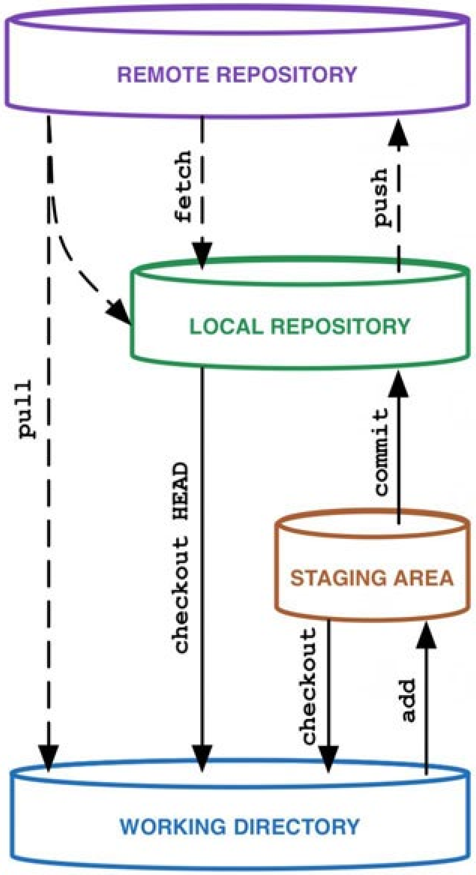
status
  - Checking the train schedule in your hand
  - Checks status of local files

pull
  - Grabbing the most up to date train schedule
  - You have the latest versions of the files

add
  - Standing on the platform in a train station waiting for the train
  - Files that are going to the same destination

commit -m
  - Walking toward the open door of a train
  - The files are ready to go to their destination

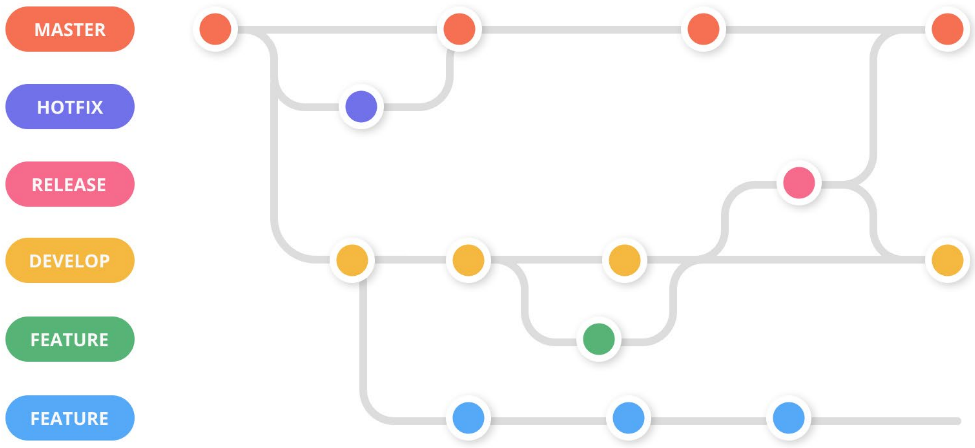
push
  - Someone shoves you into the train
  - Files on their way to destination



Source Control - Branches

Duplication of an object (file or directory)

- Each object can thereafter be modified separately and in parallel so that the objects become different
- “merging” integrate changes back into parent branch
  - resolve any merge “conflict”
- teams must determine their “workflow”
- common best practices
  - Features get their own branch
  - The main branch (trunk) is ALWAYS DEPLOYABLE
  - Require “pull requests” to force a code review before merging

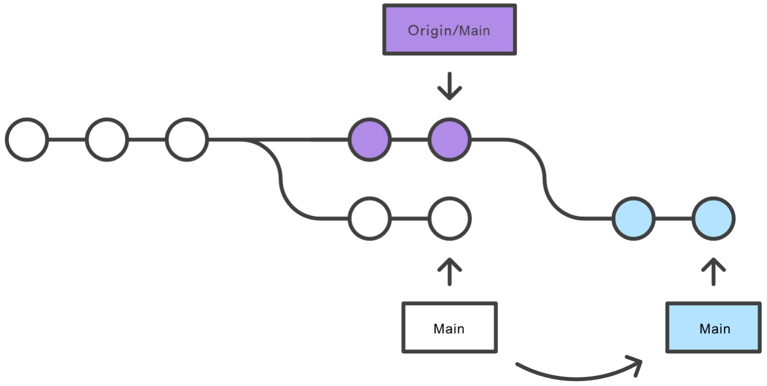


Source Control - An Individual Workflow

Begin development for the day:

- checkout main branch
- pull
- checkout your feature branch
- work
- pull main branch
- rebase main branch

(this allows for handling merge conflicts on your feature branch instead of when you're trying to merge into main)



When you're done:

- checkout main branch
- pull main branch
- merge feature branch

https://www.atlassian.com/git/articles/simple-git-workflow-is-simple

Questions?

Let’s Do Something!