Git and Gitlab

For Version Control

Git and Gitlab: Introduction

What is Version Control?

Quite simply, it is a method used to keep a software system that can consist of many versions and/or configurations, well organized.

There have been a ton of version control systems throughout the lifetime of software, some are:

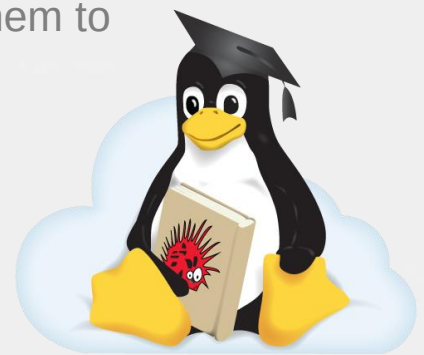
- CVS – kind of the grandfather of source control
- PVCS – commercialized CVS
- Subversion – inspired by CVS
- Perforce
- Microsoft Visual SourceSafe
- Mercurial
- TeamSite
- Vault
- Bitkeeper – Used to manage the Linux kernel before...
- Git – created by our favorite Linux author and creator: Linus!

*** From Wikipedia – http://en.wikipedia.org/wiki/application_server

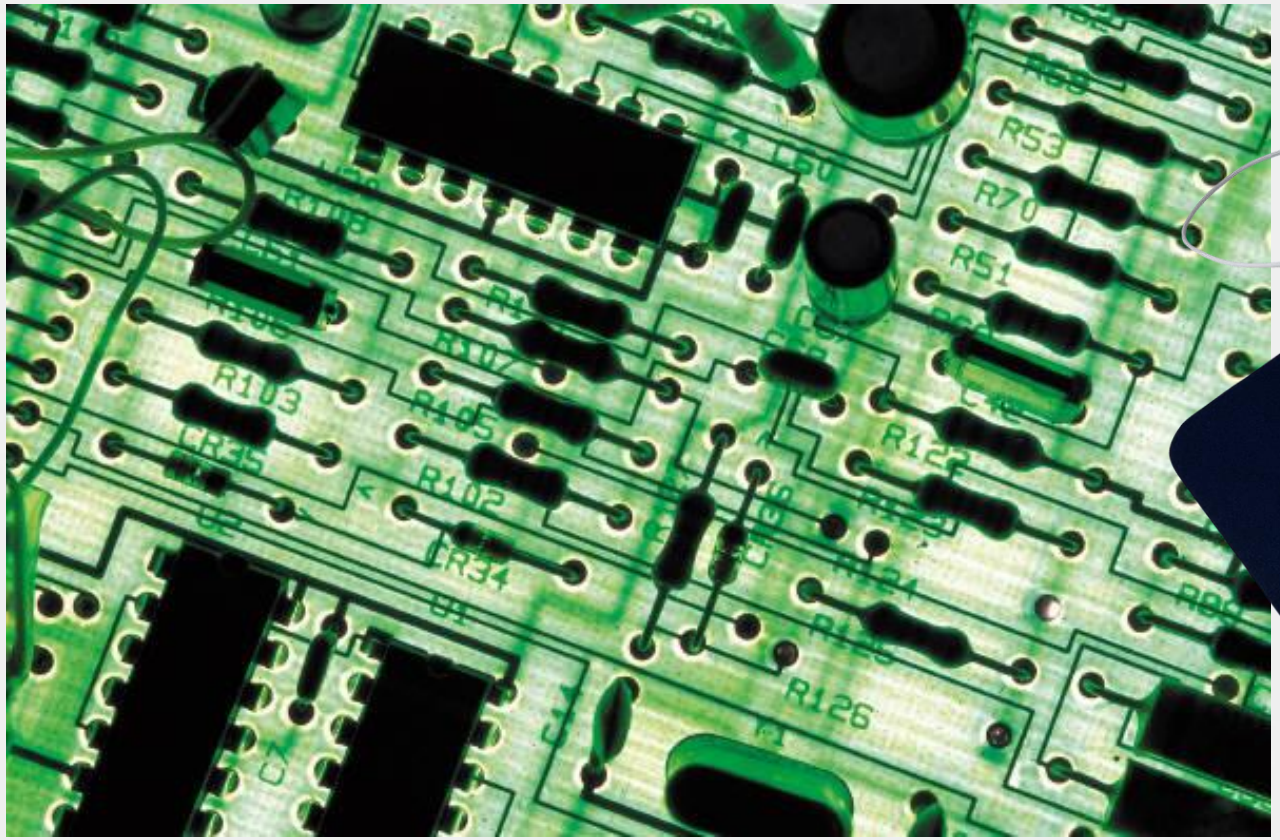
Git and Gitlab: Introduction

Common Source Tasks:

- Initialization
  - Creating the empty repository for use
- Clone
  - Making a local full copy on your workstation
- Checking Out
  - Locking a copy of one or more files for exclusive use
- Branching
  - Allowing a set of files to be developed concurrently and at different speeds for different reasons
- Merging
  - Taking different branches or sets of changes and integrating into one set or branch
- Resolving
  - Taking conflicting changes from multiple people on the same file and manually addressing
- Commit
  - Taking changes from the local system and committing them to the branch
- Push/Pull
  - Taking changes locally or remotely and merging into one or more branches



Git and Gitlab: Introduction



Covering:

We will be covering Git from start to finish, including touching on some of the public Git hosting services. In addition, we will also cover installing Gitlab, a full web application like Github, that allows you to work with multiple complex projects and teams. Don't worry if things seem a bit confusing now, we are going to make a source control expert out of you yet!