

Version Control System (vcs)

COMP 3104 DEVOPS

What is a Version Control System (VCS)?

Version control systems are a category of software tools that helps record changes to files by keeping track of modifications done to the code.

Use of Version Control System:

- ❖ **A repository:**

It can be thought as a database of changes. It contains all the edits and historical versions (snapshots) of the project.

- ❖ **Copy of Work (sometimes called as checkout):**

It is the personal copy of all the files in a project. You can edit to this copy, without affecting the work of others and you can finally commit your changes to a repository when you are done making your changes.

Benefits of Version Control System

- Software version control systems support an overall Agile development.
- Allow users to identify root causes for mistakes or bugs, or revert to an earlier version that may reduce development errors and conflicts.
- The ability to prevent overwriting by providing code review and merger conflict resolution benefits.
- Changes are historic and kept over the life of the software solution, developers can track the changes.
- Improve team productivity and enable collaboration.
- Enhance team communication with a reliable solution.
- Improve customer satisfaction with reliable software versions.

Benefits of Version Control System

Enables a safe workflow - A developer can commit code in small and easily testable increments. If any commit causes a problem... it can easily be reset to a stable state.

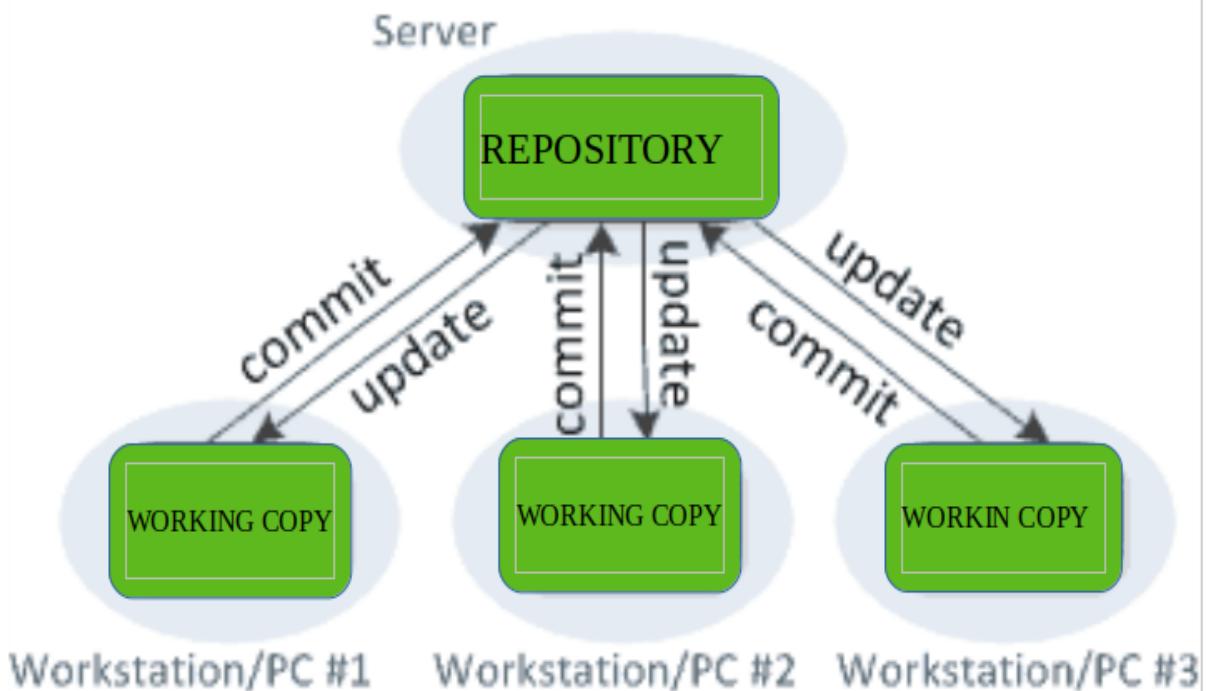
Enables experimentation without fear – If you'd like to try a change that is at risk for the introduction of unintended behavior, a developer can simply revert their work in progress back to a clean copy.

Rolling Back – A “roll back” or “rolling back” allows a developer to recover older versions which may not have that bug which was not caught before reaching production. With VCS, you can roll back to a working build with ease.

Isolating Problems - Sometimes problems can arise and it may be difficult to isolate where the problem stems from. VCS provide the luxury of traveling back in time to old code commits. Rolling back to a previous commit allows a developer to isolate and identify when the problem was introduced. This is really helpful as finding the source of the problem often times is 90% of the work.

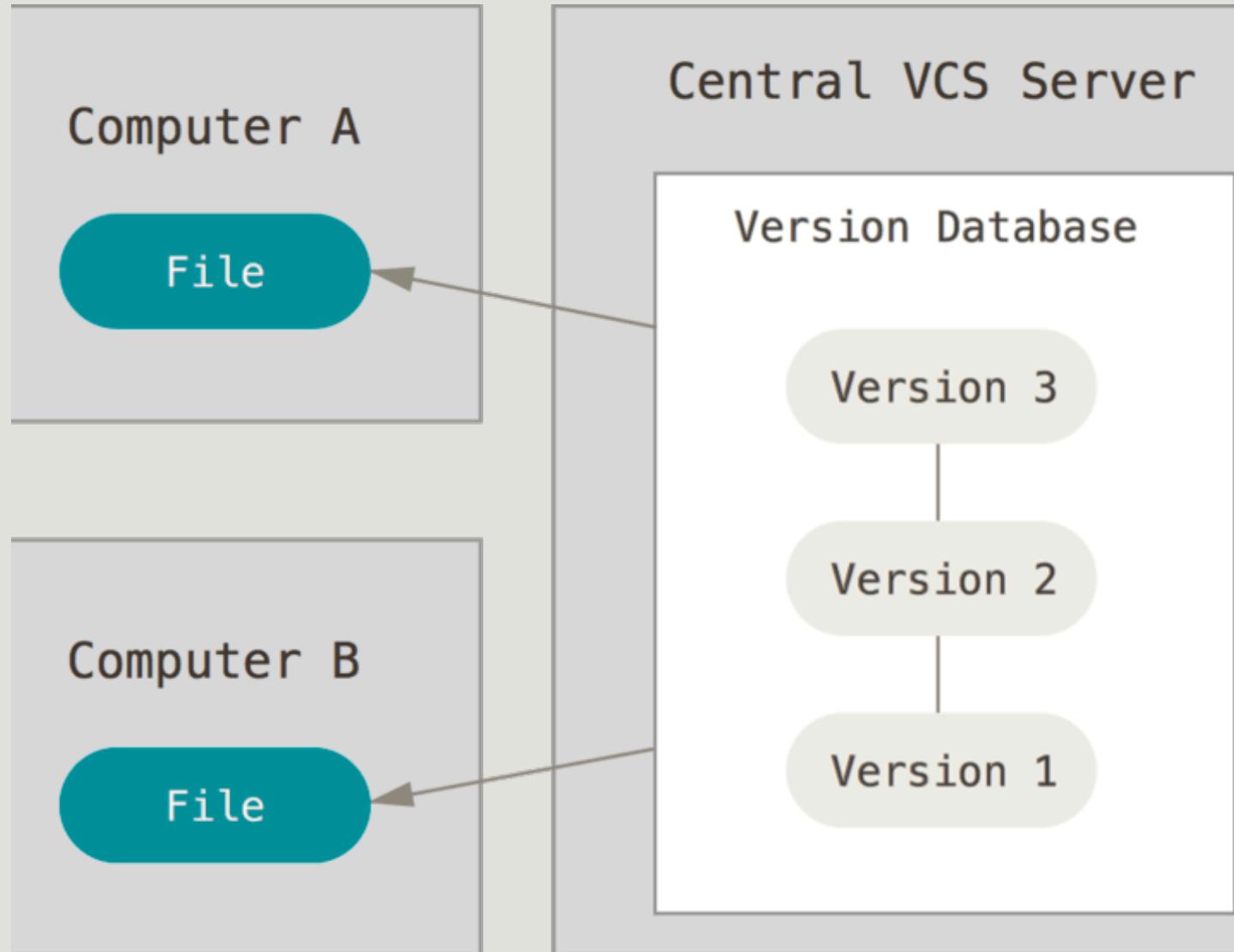
Enables CI/CD - As we will learn throughout this course, CI/CD is crucial for scaling up a development team and automating builds, testing and deployment. VCS is a fundamental enabler for these advanced techniques.

Centralized version control



What is CVCS?

- Centralized version control systems contain just one repository.
- Individual developer/user gets their own working copy on there local machine.
- You need to commit to reflect your changes in the CVCS repository.
- It is possible for others to see your changes by updating your local repository.



Pros and Cons of CVCS

Pros

- Allows to collaborate with developers on other systems.
- Providing clear insight on what everyone else is doing on the project.
- It allows administrators to fine-grained control over who can do what.

Cons

- The single point of failure
- During failure the collaboration and saving versioned changes is not possible.
- Hardware failure of the central database becomes corrupted, and proper backups haven't been kept? You lose absolutely everything.

What is DVCS ?

Distributed version control systems contain multiple repositories. Each user has their own repository and working copy.

Only committing your changes will not give others access to your changes.

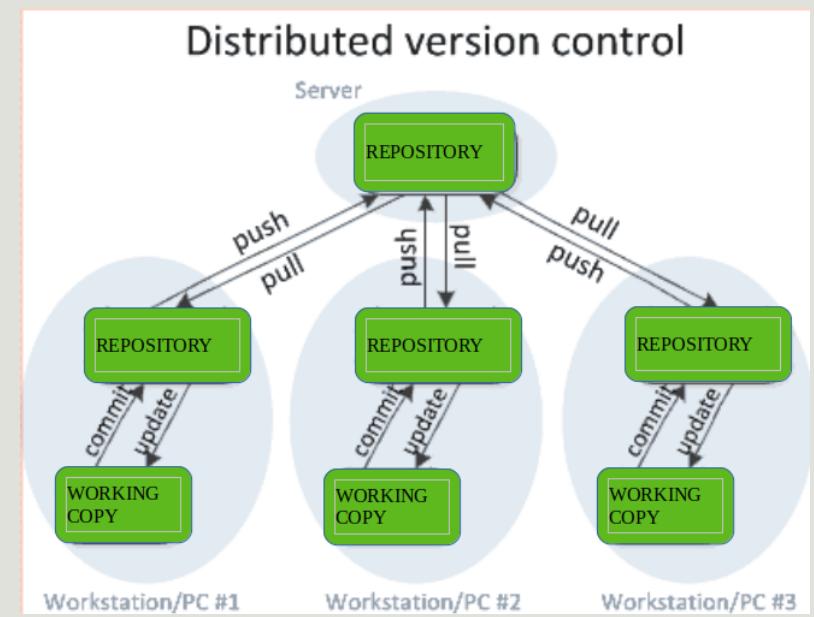
- *commit will reflect those changes in your local repository and you need to **push** them in order to make them visible on the central repository/*

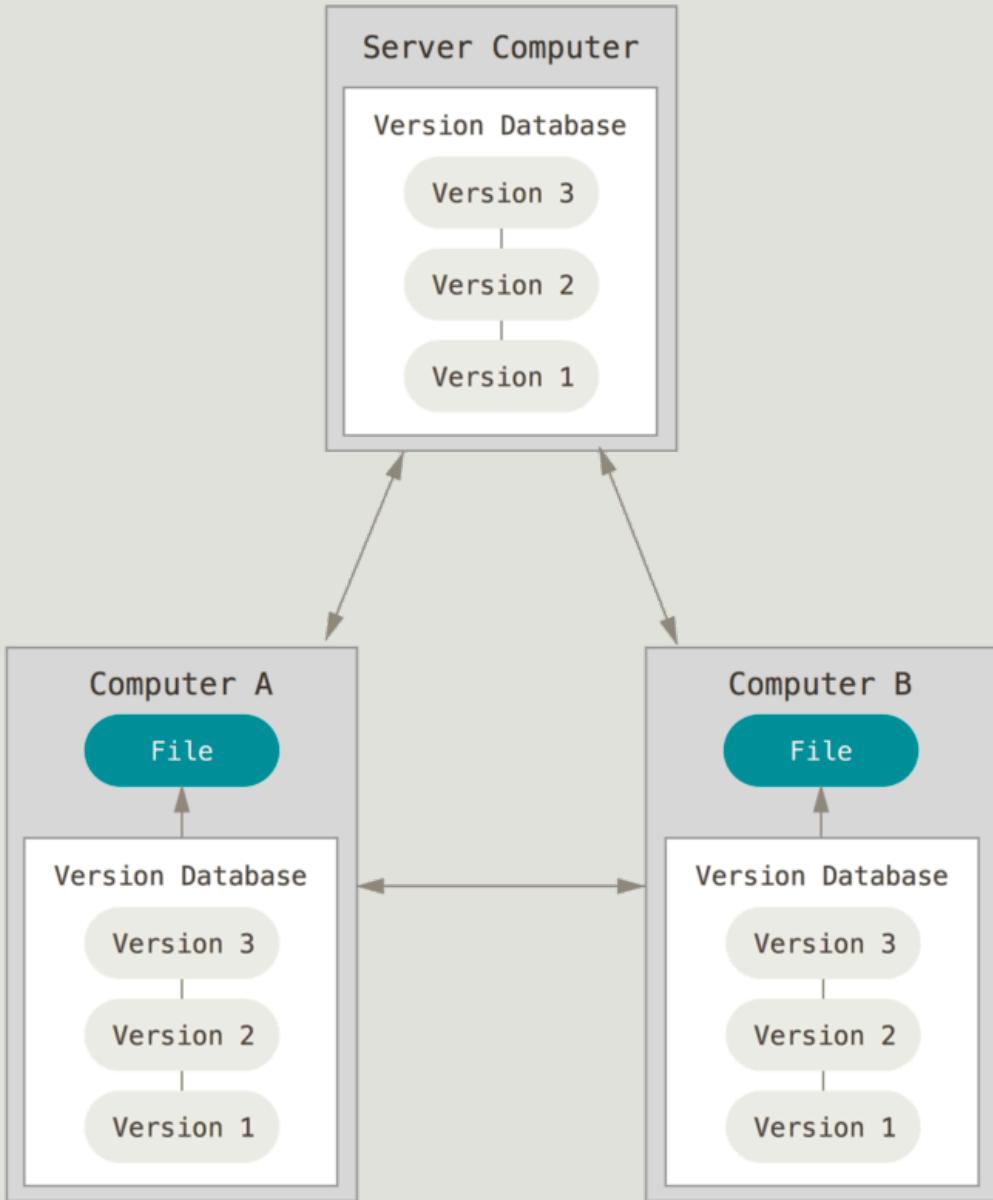
When you update, you do not get other's changes unless you have first **pulled** those changes into your repository.

To make your changes visible to others, 4 things are required:

- ❖ You commit
- ❖ You push
- ❖ They pull
- ❖ They update

The most popular distributed version control systems are Git, Mercurial. They help us overcome the problem of single point of failure.





Pros and Cons of DVCS ?

Pros

- Due to maintaining local commits, the full history is always available.
- No need to access a remote server all the time (faster access)
- Ability to push your changes continuously
- Saves time, especially with SSH keys
- Good for projects with off-shore developers

Cons

- It may not always be obvious who did the most recent change
- File locking doesn't allow different developers to work on the same piece of code simultaneously. - Helps to avoid merge conflicts, but slows down development
- DVCS enables you to clone the repository – this could mean a security issue
- Managing non-mergeable files is contrary to the DVCS concept
- Working with a lot of binary files requires a huge amount of space, and developers can't do diffs



Thank You
