An introduction to version control with Git

Sam Fearn
(s.m.fearn@durham.ac.uk)

November 5th, 2019

Outline

What is Version Control?

- A Version Control System (VCS, sometimes also SCMS) is a tool for managing a changing collection of files, allowing you to get back to a particular version at any time.

What is Version Control?

- A Version Control System (VCS, sometimes also SCMS) is a tool for managing a changing collection of files, allowing you to get back to a particular version at any time.
- E.g. myfile041119.txt, myfile051119.txt, ...

What is Version Control?

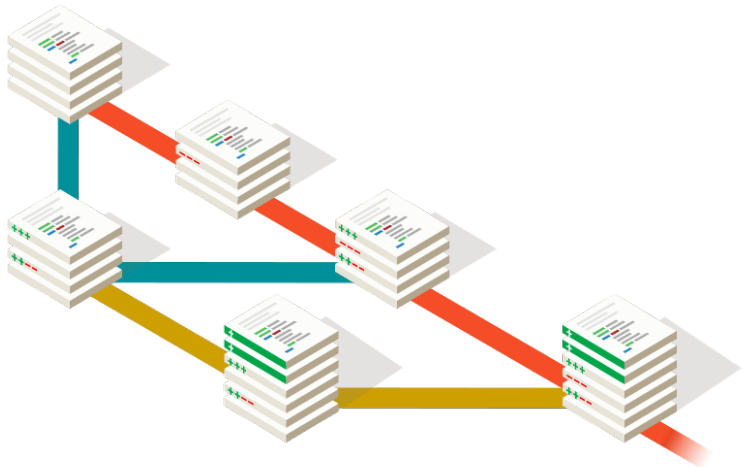
- A Version Control System (VCS, sometimes also SCMS) is a tool for managing a changing collection of files, allowing you to get back to a particular version at any time.
- E.g. myfile041119.txt, myfile051119.txt, ...
- The method above has many problems, including duplication, naming errors, easy to accidentally overwrite the old file, ...

What is Version Control?

- A Version Control System (VCS, sometimes also SCMS) is a tool for managing a changing collection of files, allowing you to get back to a particular version at any time.
- E.g. myfile041119.txt, myfile051119.txt, ...
- The method above has many problems, including duplication, naming errors, easy to accidentally overwrite the old file, ...
- Also, if you're collaborating on a project, this can get very tricky to manage. Which version did I last send? Have I included the latest changes from my collaborator? What do I do if they make changes to a section I've also changed since we last compared?

What is Version Control?

- A Version Control System (VCS, sometimes also SCMS) is a tool for managing a changing collection of files, allowing you to get back to a particular version at any time.
- E.g. myfile041119.txt, myfile051119.txt, ...
- The method above has many problems, including duplication, naming errors, easy to accidentally overwrite the old file, ...
- Also, if you're collaborating on a project, this can get very tricky to manage. Which version did I last send? Have I included the latest changes from my collaborator? What do I do if they make changes to a section I've also changed since we last compared?



Different VCS

- Many different VCS have been created to address the challenges of managing different versions of files.

Different VCS

- Many different VCS have been created to address the challenges of managing different versions of files.
- Most of the popular systems in use today are either **Centralised Version Control Systems**, or **Distributed Version Control Systems**.

Different VCS

- Many different VCS have been created to address the challenges of managing different versions of files.
- Most of the popular systems in use today are either **Centralised Version Control Systems**, or **Distributed Version Control Systems**.
- With a CVCS, there is a single server which manages the versioned files – the main downside is that this type of system has a single point of failure.

Different VCS

- Many different VCS have been created to address the challenges of managing different versions of files.
- Most of the popular systems in use today are either **Centralised Version Control Systems**, or **Distributed Version Control Systems**.
- With a CVCS, there is a single server which manages the versioned files – the main downside is that this type of system has a single point of failure.
- Examples of CVCS you may have heard of include SVN and Subversion.

Different VCS

- Many different VCS have been created to address the challenges of managing different versions of files.
- Most of the popular systems in use today are either **Centralised Version Control Systems**, or **Distributed Version Control Systems**.
- With a CVCS, there is a single server which manages the versioned files – the main downside is that this type of system has a single point of failure.
- Examples of CVCS you may have heard of include SVN and Subversion.
- In a DVCS, the entire history of each project is mirrored across multiple clients – this protects against the single point of failure.

Different VCS

- Many different VCS have been created to address the challenges of managing different versions of files.
- Most of the popular systems in use today are either **Centralised Version Control Systems**, or **Distributed Version Control Systems**.
- With a CVCS, there is a single server which manages the versioned files – the main downside is that this type of system has a single point of failure.
- Examples of CVCS you may have heard of include SVN and Subversion.
- In a DVCS, the entire history of each project is mirrored across multiple clients – this protects against the single point of failure.
- Common examples of DVCS include Git and Mercurial.

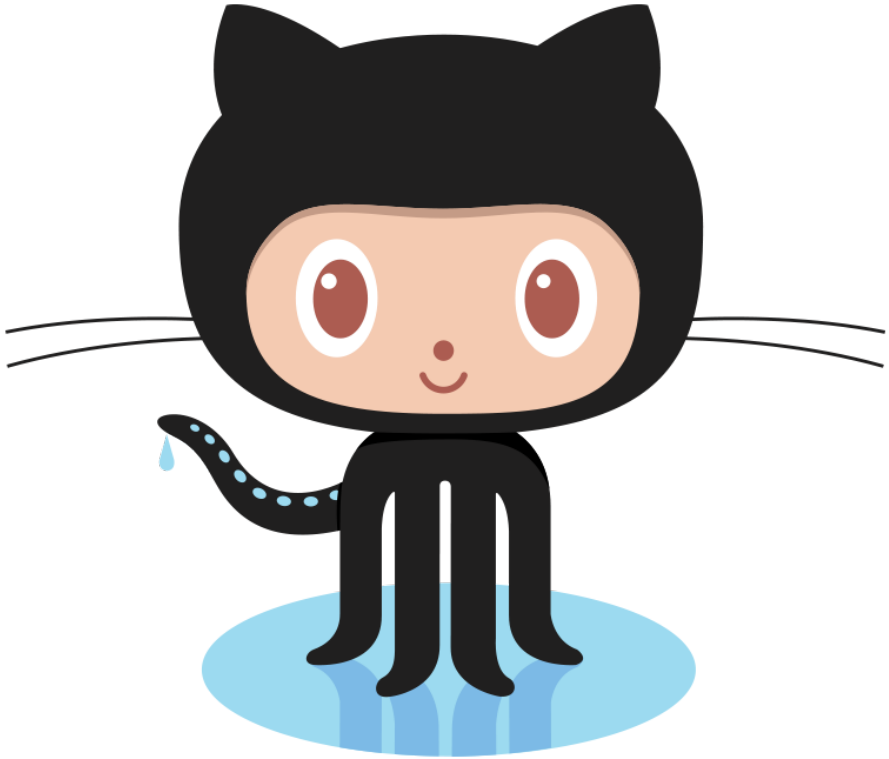
Different VCS

- Many different VCS have been created to address the challenges of managing different versions of files.
- Most of the popular systems in use today are either **Centralised Version Control Systems**, or **Distributed Version Control Systems**.
- With a CVCS, there is a single server which manages the versioned files – the main downside is that this type of system has a single point of failure.
- Examples of CVCS you may have heard of include SVN and Subversion.
- In a DVCS, the entire history of each project is mirrored across multiple clients – this protects against the single point of failure.
- Common examples of DVCS include Git and Mercurial.
- Of all the systems mentioned above, Git is probably the most common and widely used.

Different VCS

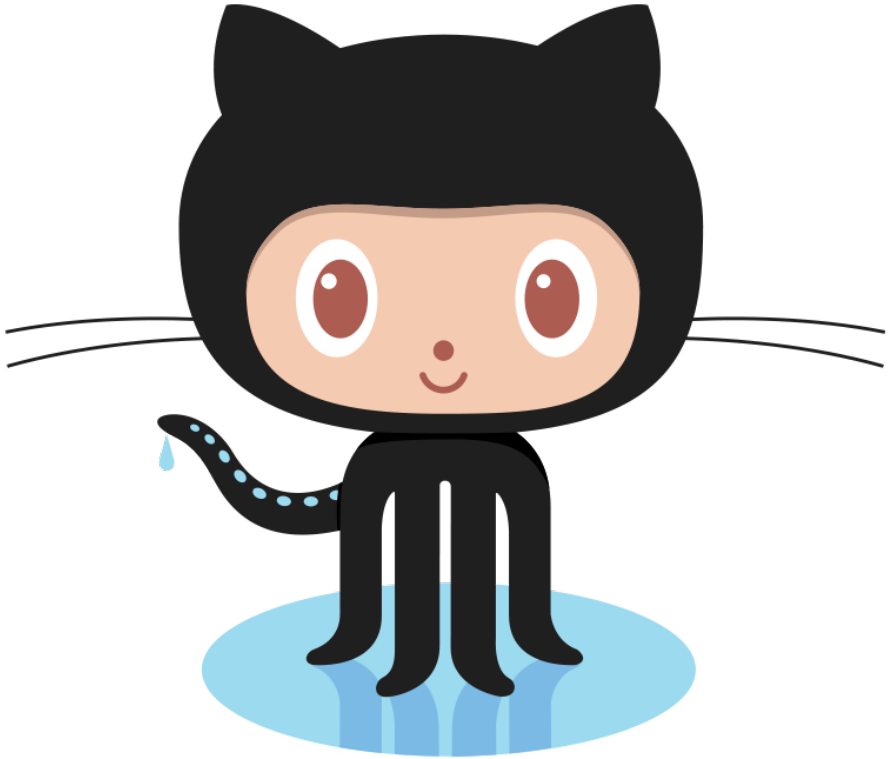
- Many different VCS have been created to address the challenges of managing different versions of files.
- Most of the popular systems in use today are either **Centralised Version Control Systems**, or **Distributed Version Control Systems**.
- With a CVCS, there is a single server which manages the versioned files – the main downside is that this type of system has a single point of failure.
- Examples of CVCS you may have heard of include SVN and Subversion.
- In a DVCS, the entire history of each project is mirrored across multiple clients – this protects against the single point of failure.
- Common examples of DVCS include Git and Mercurial.
- Of all the systems mentioned above, Git is probably the most common and widely used.

Git and Github



- Github is a free git hosting service, allowing you to share repositories with other people (or keep them private) – alternatives do exist (Bitbucket, Kiln?).

Git and Github



- Github is a free git hosting service, allowing you to share repositories with other people (or keep them private) – alternatives do exist (Bitbucket, Kiln?).
- Git can be used without Github. Your repository is always stored locally, and you can share this however you wish – ssh, ftp, (email, memory stick,...!) etc.

Using Git at the command line

Git GUIs

Useful Resources