Lecture 01 - Working with Linux Servers and Git

CS 1XA3

Jan. 9, 2018

Working with Linux Servers: SSH

- SSH (named for Secure SHell) is a protocol commonly used for remote login. You can use it from a command line interface with the following syntax

  ```
  ssh username@server_url
  ```

- Example: I can log into the McMaster Server: mills.mcmaster.ca with my MacID: dalvescb

  ```
  ssh dalvescb@mills.mcmaster.ca
  ```

Try logging into mills.mcmaster.ca with your MacID

Working with Linux Servers: SCP

- SCP (Secure Copy) allows Remote Transfer of files
- To copy a file from a server to */copypath*

```
scp username@server_url:/path/to/file.txt /copypath
```

- To copy a file from a server to local machine

```
scp /path/to/file.txt username@server_url:/path/to/
```

Working with Linux Servers: Browsing File Directories

- File Paths are denoted using a *backslash* like so

  `/path/to/file.txt`

- Basic Commands for file browsing include
  - **cd** */path/to*
    Change Directory to */path/to*
  - **ls**
    List current directory contents
  - **pwd**
    show *Parent Working Directory* (where am I?)

Try entering **pwd** after logging in with ssh, it should list your *HOME* directory

Working with Linux Servers: Manipulating Files/Directories

Basic Commands for file manipulation

- **cp** *file1.txt file2.txt*
  Copy file1.txt to file2.txt
- **cp -r** *dir /path/to*
  Copy a directory *dir* and all its contents to */path/to/*
- **mv** *file.txt /path/to*
  Move file or directory to */path/to*
- **mv** *file1.txt file2.txt*
  Rename file1.txt to file2.txt

Working with Linux Servers: Manipulating Files/Directories

Basic Commands for file manipulation

- **mkdir** *Directory1*
  Make a new directory named *Directory1*
- **rm** *file.txt*
  Removes a file - Warning: no undo
- **rm -r** *dir*
  Removes a directory and its contents - Warning: no undo

Working with Linux Servers: Text Editors

- GNU Emacs - https://www.gnu.org/software/emacs/
  Complicated but very powerful
- Nano - https://www.nano-editor.org
  Simple and very popular
- Vim - https://github.com/vim/vim
  Recommended for this course, feature-ful and not too complicated

Working with Linux Servers: Built-in Manuals

If you ever want to refresh your memory on how a certain command works, you can access the command's Manual with the following command

```
man command
```

Press **q** to quit

Warning: most linux/unix commands, even the very basic ones, are extremely feature-full. The built-in manuals are often fairly intimidating for beginners

Version Control Systems

Version Control gives you a means to manage your source code, and is essential for multi-developer projects. There are really only two commonly used version control systems amongst software developers

- Subversion (SVN)
  Simpler design, revolving around a single centralized repository
- Git
  A distributed revision control system, a bit more complicated but with more powerful features

Version Control with Git: Simple Git Commands

- **git clone** *repo-url*
  Download the code from Remote Repo to current directory
- **git pull**
  Merges code from remote repository to current directory
- **git add** *file* / **git rm** *file*
  First step to adding or removing a file or directory
- **git reset** *file*
  Undo local changes so far (opposite of git add)
- **git commit** *-m "comment"*
  Commit changes to Local Repo
- **git push**
  Push changes in Local Repo to the Remote Repo
- **git –help**
  List git commands with descriptions

Version Control with Git: Repository Control Flow

Git version control manages code between two Repo's:
a Local Repo and a Remote Repo