CUNY Tech Prep

Edgardo Molina, PhD

Summer 2016 – Session 1

Welcome

- What is CUNY Tech Prep? (The technical part)
  - Learn Full Stack Application Development
  - Project Based Course
  - Open Source Project
  - Industry Practices
  - Industry Guest Lecturers

Course Overview and Expectations

- Develop several Full Stack Web Applications
- Work in groups
- Collaborate across groups
- Regular project presentations
- Quizzes via HackerRank
- Commit substantial time outside of class

Student Introductions

- Name
- School
- Which **software product** or **open-source project** would you LOVE to work on?

Outline

- Let’s create some accounts
  - Slack, Github
- Install some software
  - Git, Sublime Text 3, Virtual Box, Vagrant
- Learn about Linux OS and Bash Shell
- Let’s use Git and Github
- Finally, let’s install Node.js and run some JavaScript

Subjects in Gray we did not get to in the first session. We will cover in the second session.

Slack - Communication

- We will use Slack for class/team collaboration:
  - https://ctp2016.slack.com

- For personal/private matters
  - Contact me via email: molina@cs.ccny.cuny.edu
  - Or Ms. Jessica Hill, or Ms. Dariela Estrella

What is Git

- Git is a Version Control System
- Developers use a Version Control System:
  - to keep a history (versions) of their changes to code
  - to collaborate with multiple people on an application
  - to resolve conflicts that emerge when multiple people change the same code
- Git is local and distributed
  - Works completely locally, you have the entire history
  - It can be pushed to a remote service to share with multiple developers

- Github is a remote service https://github.com

Install Git

- https://git-scm.com/

- Download the installer for your platform

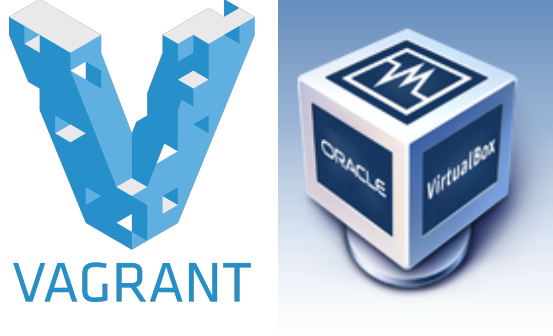
- If you are on Windows,
  - The default installation choices are fine.
  - This will install an SSH client. You'll need it for Vagrant to work properly.

Install Sublime Text

- An extensible text editor

- https://www.sublimetext.com/3

Virtual Box / Vagrant



- Vagrant provides us with configurable lightweight virtual machines
- Benefits
  - Our host operating system does not matter
    - No Windows/Mac/Linux compatibility issues
  - All developers will be able to run the same software
  - We can mirror our deployment environment
- Cons
  - Can consume a lot of memory/CPU (not good for older computers)

Install Virtual Box and Vagrant

- Install the Virtual Box (Host)
  - https://www.virtualbox.org/wiki/Downloads
- Install Vagrant
  - https://www.vagrantup.com/downloads.html

- Open terminal and run
  - `vagrant init ubuntu/trusty64`
  - `vagrant up`

- Logging in
  - `vagrant ssh`

Before Linux, Let's talk applications…

- Local
  - Run on a single computer/device
  - Take input (from keyboard or files, STDIN)
  - Produce output (to screen, STDOUT and files)

- Distributed
  - Runs on more than one machine (called servers).
    - Can be a set of applications (system) working together
  - Takes input from a remote machine, over a network
  - Performs computations on the server
  - Sends output back to a remote machine, over a network

Which local and distributed applications have we just used?

Linux OS - Ubuntu

- “Linux” based Operating Systems are the most popular OS for Servers
- Linux is actually just the Kernel
- **Ubuntu** is a Linux Distribution that packages the kernel along with the base GNU applications for a GNU/Linux Operating System to make up a UNIX-like Operating System
- Why?
  - Open Source, Customizable, Standard tools

https://en.wikipedia.org/wiki/GNU_Project#GNU.2FLinux

Distributions (commercial and open source)

- Redhat Enterprise
- CentOS
- Fedora
- Ubuntu
- Amazon Linux
- Arch Linux
- … many many more!

- All use same Linux kernel and GNU applications
- They differ in package managers, configuration, licenses, and community

Ubuntu

- Different Variations/Flavors
  - Desktop
  - Mobile
  - Server
  - etc…
- LTS – Long Term Support versions released every 2 years
- We will use ‘Ubuntu Server’

Ubuntu Package Management

- Apt-Get

- A package manager queries a package repository for available software that can be installed. It also manages **updating** software installed via the package manager, and **resolving dependencies**

- Ubuntu provides a vetted repository of applications. You are free to add additional repositories (at your own risk!)

- Package Managers are also available for programming languages and their libraries
  - Examples: …

Let’s dive in

Filesystem Hierarchy Standard

- System Directories (Typically don’t modify)
  - **/boot** contains files needed to start up the system, including the Linux kernel, a RAM disk image and bootloader configuration files.
  - **/dev** contains all *device files*, which are not regular files but instead refer to various hardware devices on the system, including hard drives.
  - **/sys** is a virtual filesystem that can be accessed to set or obtain information about the kernel's view of the system.
  - **/proc** is a virtual filesystem that provides a mechanism for kernel to send information to processes.
  - **/root** is the superuser's home directory, not in /home/ to allow for booting the system even if /home/ is not available.
  - **/bin** is a place for most commonly used terminal commands, like ls, mount, rm, etc.
  - **/lib** contains very important dynamic libraries and kernel modules
  - **/sbin** contains important administrative commands that should generally only be employed by the superuser.
- External Storage
  - **/media** is intended as a mount point for external devices, such as hard drives or removable media (floppies, CDs, DVDs).
  - **/mnt** is also a place for mount points, but dedicated specifically to "temporarily mounted" devices, such as network filesystems.

Only modify if you are dealing with custom hardware, filesystems, etc

Source: https://help.ubuntu.com/community/LinuxFilesystemTreeOverview

Filesystem Hierarchy Standard

- Server Application Directories
  - **/var** is dedicated to variable data, such as logs, databases, websites, and temporary spool (e-mail etc.) files that persist from one boot to the next. A notable directory it contains is /var/log where system log files are kept.
  - **/srv** can contain data directories of services such as HTTP (/srv/www/) or FTP.
  - **/etc** contains system-global configuration files, which affect the system's behavior for all users.
- Of interest to us
  - **/home** home sweet home, this is the place for users' home directories.
  - **/opt** can be used to store additional software for your system, which is not handled by the package manager.
  - **/tmp** is a place for temporary files used by applications.
  - **/usr** contains the majority of user utilities and applications, and partly replicates the root directory structure, containing for instance, among others, /usr/bin/ and /usr/lib.

Source: https://help.ubuntu.com/community/LinuxFilesystemTreeOverview

JavaScript (officially ECMAScript)

- ECMAScript is usually abbreviated ES
  - Latest version ES6, most supported ES5 (at the moment)
- Client-Side
  - JavaScript is supported by all web browsers without requiring a plugin
  - Allows developers to add dynamic content to web pages
  - Within the browser the language is sandboxed, it cannot access the filesystem and other host machine resources. Only what the browser allows.
- Server-Side
  - Node.js popularized the use of JavaScript on the server side for backend and full application development. On the server-side the language has access to all host machine resources.

JavaScript

- Is an Interpreted and untyped language
- Allows for procedural and OO programming
- Syntax is similar to C/C++ and Java
- Known as a “Prototype” language

JavaScript Tools

- nvm – node version manager
- npm – node package manager
- node – JavaScript Runtime (interpreter)

- Chrome and Firefox (client-side)
  - Developer tools console

Homework

- JavaScript Course
  - https://www.codecademy.com/learn/javascript
    - Units 1-6
  - Free Text: http://eloquentjavascript.net/
    - Chapters 1-4
- HTML+CSS Course
  - https://www.codecademy.com/learn/web
    - Units 1-5 (try 6)
  - Reference:
    - http://www.w3schools.com/html/default.asp
    - http://www.w3schools.com/css/default.asp
- JavaScript HackerRank Problem Set

- *Skip "Pro" sections*