**New Server Procedure**

1. Create Server
   1. Configured to the Development Guidelines specifications for this class
2. SSH Into the Server
   1. ssh root@Server\_IP\_Address
   2. Enter password
3. Create Non-Root User
   1. adduser Username
   2. adduser Username sudo
4. End SSH Session
   1. exit
5. Login as Non-Root User
   1. ssh Username@Server\_IP\_Address
   2. Enter password
6. Update Package System
   1. sudo apt-get update
   2. Enter password
7. Upgrade Package System
   1. sudo apt-get upgrade
8. Update Packages for Newly Installed Version
   1. sudo apt-get update
9. Update System level Packages
   1. sudo aptitude update
   2. sudo aptitude safe-upgrade
   3. sudo reboot
10. Install Packages
    1. Git
       1. See Git: Install & Config
    2. Apache
       1. See Apache Install & Config

**Git: Install & Config**

1. Install Git
   1. sudo apt-get install git-core
2. Configure Git
   1. git config --global user.name “NAME”
   2. git config --global user.email [Your@Email.com](mailto:Your@Email.com)
3. Confirm Settings
   1. git config --list
4. Create SSH Keys for Github Access
   1. ssh-keygen -t rsa -C ”[YourEmail@example.com](mailto:YourEmail@example.com)”
      1. This will ask if you want to customize the name, you don’t. Just press enter.
   2. Enter a passphrase
   3. Re-enter the passphrase
5. Put the RSA key on file with github.com so this server is treated as a trusted machine.
   1. less ~/.ssh/id\_rsa.pub
      1. Copy ALL of the contents of this file to your clipboard.
   2. Add new SSH Key to your Github account under the [Account Settings](https://github.com/settings/ssh).
      1. Click the Add Key button
      2. Enter a title defining the server
      3. Paste the RSA file’s contents into text area marked “Key”
      4. Click Add Key

**Apache Install & Config**

1. Install Apache 2
   1. sudo apt-get install apache2
2. Configure ServerName
   1. Restart Server
      1. sudo service apache2 restart
   2. Failed to Restart
      1. apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1 for ServerName
   3. Configure ServerName
      1. sudo pico /etc/apache2/conf.d/security
         1. Add
            1. ServerName localhost
      2. sudo service apache2 restart
      3. Apache should report a successful restart
3. Restrict Access
   1. sudo pico /etc/apache2/conf.d/security
      1. Uncomment <Directory />
      2. Add
         1. Options FollowSymLinks
      3. sudo service apache2 restart
4. Change Permissions to Allow Access
   * + 1. sudo chown -R UserName /var/www
5. Further Reading (optional)
   1. <https://www.digitalocean.com/community/articles/how-to-configure-the-apache-web-server-on-an-ubuntu-or-debian-vps>

#### Configure Git Deployment

1. Create Space for Repo on Live Server
   1. sudo mkdir /var/repos/
   2. sudo chown **YourUserName** /var/repos
      1. Changing ownership of the Repos folder to your user instead of the Root user allows for your further manipulation of it’s content without superuser permissions.
   3. cd /var/repos && mkdir **SiteName**.git && cd **SiteName**.git
   4. git init --bare
      1. A bare git init means this folder will have no source files, only the version control structure of git.
2. Configure Server Post Hook
   1. cd /var/repos/**SiteName**.git/hooks/
   2. pico post-receive
      1. Save the file with the contents below. It separates the working tree and git dir into separate locations.
         1. #!/bin/sh GIT\_WORK\_TREE=/var/www git checkout -f
   3. chmod +x post-receive
      1. Give permissions to make the hook executable by the server.
3. Configure Local Dev Environment
   1. It’s important to have a centralized space for your Projects and Deployments even if you’re not deploying multiple sites on this server.
   2. Open a new Terminal Window
   3. mkdir ~/Projects
   4. mkdir ~/Projects/SiteName.com && cd ~/Projects/SiteName.com
      1. Even if you did not purchase a domain name use a .com or .net etc of what you would like to use. In linux adding a .com to a folder name does not complicate things as it would on your laptop.
   5. git init
   6. git remote add prodServer ssh://YourUserName@IPAddress/var/repos/SiteName.git
4. Further Reading
   1. <http://toroid.org/ams/git-website-howto>