

# Development Environment

## Setting up your Development Environment

The HUBzero Platform is comprised of many software packages woven together into an a single system. The minimum requirements for developing HUBzero web code can be a simple Linux Apache MySQL PHP (LAMP) server matching the currently supported Linux distribution. At the time of writing, Centos 7. Although the a simple LAMP server should suffice, it should not The HUBzero maintainers try to ensure that system administrators do not have to deviate too much from the stock distribution in order to install HUBzero.

## Setup Development Environment via VMware Image

The Hubzero maintainers package a VMware Image available for evaluation and development purposes available for download at <https://hubzero.org/download>.

## Setup Development Environment via Packages

If you are confident in your Linux system administration skills, the Hub can be built via Debian and RedHat packages. Instructions can be found [in our documentation under System Administrators](#). Once the hub is operational, you will want to clone the hubzero-cms repository for development.

See the Getting the Latest Code section for instructions on how to use git to grab the latest code.

## Getting the Latest Code

In the pre-packaged development environments listed above, the hubzero-cms code is a snapshot of when the package was created. To obtain the latest code, we will use git to clone the hubzero-cms repository.

In order to get the latest code you will need:

- SSH access to the machine
- sudo privileges
- git

A couple conventions for this section, in particular:

- this text indicates an incantation to be typed in the terminal.
- <something> indicates something that should be replaced with value you have set.
- [!] indicates something that should be interpreted, or have your judgment applied.

1. `ssh <username>@<hostname>`
  1. Log into your development server
2. `cd /var/www [!]`
  1. Change into the webroot directory.
  2. [!]: This may be /www in some configurations.
3. `mv <hubname> <hubname>.bak [!]`
  1. Backup the existing hubzero-cms installation.
  2. [!]: The name of the directory may change depending on [what <hubname> is used for hzcms install](#). In the documentation, 'example' is used. Therefore the directory will be /var/www/example.
4. `git clone https://github.com/hubzero/hubzero.git <hubname>`
  1. This will clone the latest release into your webroot. The 'dev' branch is the default. You may switch to another available at this step if desired.
  2. This will create the shell of the application, we will need to (re)-populate it with configuration values and pull in external dependencies later on.
5. `cp <hubname>.bak/app <hubname>/.`
  1. This moves the configuration files back into the application. If prompted, overwrite the new app/ directory.
6. `cd <hubname>/core`
  1. Navigate to the core directory.
7. `php ./bin/composer install`
  1. Pull in external dependencies using composer.
8. `cd ../`
  1. Navigate back to the /var/www/<hubname> directory.
9. `php muse migration -i -f`
  1. Run database migrations. This helps maintain the database's schema between versions. Note: Reversing database schema changes can be extremely difficult if needed.
10. `cd /var/www`
  1. Navigate back to the directory containing the hub installation.
11. `chown apache:apache <hubname> -R [!]`
  1. Change the owner and the group of the hub installation directory.
  2. [!] The user may be *apache* OR *www-data*, depending on the distribution.
12. `chmod 775 <hubname> -R`
  1. Change the file permissions to allow the group to read and write permissions.
13. `sudo usermod -aG apache <username>`
  1. Add your user to the apache group.
14. `git pull --rebase`
  1. Pull the latest code. This should be done periodically.
  2. Use the rebase flag if you already have some commits that you haven't pushed yet. This puts your work on top of the changes that come downstream.

**Note:** [!] The permissions scheme used here is NOT production safe. It's used for development purposes. File permissions and your security scheme depend on your needs.

