Post Installation Setup

Please follow these instructions which are accurate as of 2015-12-02.

Post Installation OS Updates

- `sudo apt-get update` updates the aptitude repository sources and identifies update canditates
- `sudo apt-get upgrade` apples updates
- `sudo apt-get install php5-dev` installs developer and command line tools for PHP. *ESSENTIAL*.
- `sudo apt-get install build-essential` installs compilers and libraries for the C language and many other tools we like
- `sudo apt-get install git`

You may be prompted about software that is outdated or no longer after running `upgrade`, if so, run `apt-get auto remove`.

After running auto-remove you may be prompted to update GRUB, the boot-loader. If so, run `update-grub`, then reboot.

Installing a LAMP stack.

There are many ways. This way is very simple:

`sudo apt-get install tasksel`

Then run `tasksel` on the command line, choose the LAMP Server option and be done.

You will be be prompted for the root MSYQL password; which is always `EDITEDOUT` for development environments.

Please see the separate section on setting up Apache virtual hosts, but installation is complete, you will have a fully functioning website which is PHP enabled running at

`/var/www/html`, or `/var/www` (this varies sometimes with each release of Ubuntu).

Installing Protocol Buffer Support

You need to first install all the libraries and tools for using Protocol Buffers including it's compiler into the OS, and then you need to install an apache module for supporting Protocol Buffers into the web server. Once this is done you can download and use the Allegro Protobuf libraries from GitHub.

`sudo apt-get install google-protobuf` `sudo apt-get install protobuf-compiler`

Then run `protoc --version` and confirm you have `libprotoc 2.5.0` or higher.

Now, make sure you are in the root user's home folder:

`cd ~` should put you here: `/root`

Then download the Allegro library and set it up.

- `git clone https://github.com/allegro/php-protobuf.git` (checkout the files)
- `cd php-protobuf` move into the folder
- run `phpize` to prepare the extension we need for the PHP environment. There is more information on phpize here if needed.

Now do a standing configure/make/install:

- `./configure`
- `make`
- `make install`

The result will tell you where the .so file is located. Copy this to your clipboard or a notepad.

Now that the exertion (an .SO file, basically equivalent to a DLL in windows) is built it needs to be installed into the PHP configuration for both the command line PHP interpreter and the CGI extension for PHP for Apache.

For CLI Support (needed by the Allegro library)

```
nano /etc/php5/cli/php.ini
```

Locate the “Dynamic Extensions” section (use CTRL + V to jump down by page and CTRL + Y to jump up by page): Also, CTRL + W to search for text, CTRL + X to leave w/o saving.

It looks like this:

;;;;;;;;;;;;;;;;;;;;;; ; Dynamic Extensions ; ;;;;;;;;;;;;;;;;;;;;;;

Put this under it, assuming that the resulting path of `make install` had a folder named “20121212”. Adjust accordingly.

```
extension_dir = "/usr/lib/php5/20121212/"
extension="/usr/lib/php5/20121212/protobuf.so"
```

`CTRL + X` to save, then press Enter/Return

Now restart Apache:

```
sudo service apache2 restart
```

Then confirm the module is loaded with

```
php -m
```

If you don’t see ‘protobuf’ in the output list in the “P” section something is wrong. Look for path errors in the php.ini file or for errors you might have overlooked when restarting Apache.

For Apache Support

This procedure is essentially the same:

`sudo nano /etc/php5/apache2/php.ini`

;;;;;;;;;;;;;;;;;;;;;; ; Dynamic Extensions ; ;;;;;;;;;;;;;;;;;;;;;;

Put this under it, assuming that the resulting path of `make install` had a folder named “20121212”. Adjust accordingly.

`extension_dir = "/usr/lib/php5/20121212/"` `extension="/usr/lib/php5/20121212/protobuf.so"`

`CTRL + X` to save, then press Enter/Return

Now restart Apache:

`sudo service apache2 restart`

Use a `phpinfo()` function to confirm there is a section called `protobuf` and it’s version reads `0.9`

You are now ready to use PHP / ProtoBuf on Apache.