Installing Ushahidi v2.X on Ubuntu 10.4

The following instructions describe the process of setting up an Ushahidi instance on Ubuntu. The instructions are based on a clean install of Ubuntu 10.04 Lucid Lynx, so they include how to install all of the prerequisites that you will need.

This article is geared towards installing for development purposes. Therefore, it describes the process of setting up access to the GitHub repository and downloading the latest development version of Ushahidi. There are a couple of other points where things are done which would not be fully appropriate in a 'production' environment.

Note: There are a couple of points in this guide where the wiki host decided that the text being entered might be an attack on the server. Notably sudo commands that mentioned the apache path. Look out for the note Replace | with / which indicates where a command can't just be copied and pasted into your command line.

Step 1 : Install pre-requisites

First there are a few packages that you need to install in order to follow the instructions below.

- xclip - Enables you to do a command line copy of text to the clipboard.
- git - The Ushahidi application files are stored on GitHub. The Git client gives you easy access to download them.
- lamp-server - This is a handy package that will pull in almost all of the Apache, MySQL and PHP packages you will need.
- libapache2-mod-auth-mysql - This seems to be essential for the application to connect to MySQL.
- mcrypt - Kohana, which is used by Ushahidi, needs this for some features.
- sudo apt-get install xclip
- sudo apt-get install git-core
- sudo tasksel install lamp-server
- sudo apt-get install libapache2-mod-auth-mysql
- sudo apt-get install php5-mcrypt
- sudo apt-get update
- sudo apt-get install php5-curl
  NB: See the forum discussion warning that “sudo tasksel install lamp-server” may be a dangerous way to install lamp-server.

You can now check out what you have just installed:

```
dpkg -l | grep php
dpkg -l | grep mysql
```

Step 2 : Configure your Git installation

Set up your global GitHub settings:

```
git config --global user.name "Your Name"
git config --global user.email "your.name@youremail.com"
git config --global github.user "your github name"
git config --global github.token "your github token here"
```

ssh-keygen -t rsa -C "your.name@youremail.com"Copy your new SSH key to the clipboard:
cat ~/.ssh/id_rsa.pub | xclip -sel clipGo to github “Account Settings”. Under SSH public keys, paste the key you just sent to the clipboard.

At this point it would be a good idea to create your own GitHub fork of Ushahidi.

Step 3 : Configure Apache

The mod_rewrite Apache module must be enabled for the installation to work. This is enabled as follows:

```
sudo a2enmod rewrite
```

Create an apache configuration file defining your Ushahidi folder as its base.

Replace | with /

```
sudo cp |etc|apache2/sites-available/default |etc|apache2/sites-available/ushahidi-dev
```

Open this file in gedit as follows in order to edit it.

Replace | with /

```
gksudo gedit |etc|apache2/sites-available/ushahidi-dev
```

You can replace the contents as shown below. This enables web access to your Ushahidi directory and configures logging specific to Ushahidi.

```
<VirtualHost \*:80>
  ServerAdmin webmaster@localhost

  DocumentRoot /var/www/ushahidi-dev

  <Directory /var/www/ushahidi-dev>
    Options FollowSymLinks MultiViews
    AllowOverride All
    Order allow,deny
    Allow from all
  </Directory>

  ErrorLog /var/log/apache2/error-ushahidi-dev.log
  LogLevel warn

  CustomLog /var/log/apache2/access-ushahidi-dev.log combined

</VirtualHost>
```

Save and quit gedit, then run the following command to enable this new application:

```
sudo a2dissite default && sudo a2ensite ushahidi-dev
```

Run the following command to update the running apache configuration with your changes.

Replace | with /

```
sudo |etc|init.d|apache2 reload
```

Step 4: Create a fresh Ushahidi instance

Create a directory to hold the Ushahidi files.

```
sudo mkdir /var/www/ushahidi-dev
```

Saving the files here creates them as owned by your Ubuntu 'root' account. For convenience later, set the ownership of this directory to your Ubuntu user account. This is useful if you are the only developer, as it enables you to complete the rest of this process with minimal use of sudo. It also means that you will be able to modify files later in your text editor with minimal hassle. In a shared environment you should probably set up group permissions instead.

```
sudo chown your_user /var/www/ushahidi-dev
```

Enable your Ushahidi folder for source control using Git, and download all the files you need from the GitHub remote repository. This assumes that you have created your own branch of Ushahidi.

```
cd /var/www/ushahidi-dev/
git init
git remote add origin git@github.com:your_github_user/Ushahidi_Web.git
git pull origin master
```

After downloading, the Ushahidi files don't have sufficient privileges to run the installer, so you need to alter a number of files as follows.

```
chmod 777 application/config/config.php
chmod 777 application/config
chmod 777 application/cache
chmod 777 application/logs
chmod 777 media/uploads
chmod 777 .htaccess
```

Create a MySQL database to support your Ushahidi instance. You will be prompted to set up a root password.

```
sudo mysqladmin -u root --password create ushahidi-dev
```

Step 5 : Run the Ushahidi installer

All of the prerequisites are in place now, so you are ready to run the installer. If you are running on a virtual machine, now would be a good time to take a snapshot. It will make it much easier to start again if you missed something above.

Browse to your new web application to start the install. If you installed locally, following the instructions above, this link should work:

http://127.0.0.1/ushahidi-dev