



## Required Software

- Linux (Ubuntu Server 12.04 recommended) <http://www.ubuntu.com/download>
- Server (Apache2 recommended)
- Mysql (PhpMyAdmin recommended)
- Php5
- Node.js
- Redis
- Haproxy
- Drush
- Git

Only for games managers:

- Python
- Sqlite 3
- Flask
- Php5 modules

This guide presume you are using the recommended configuration and explains how to set up Experimental Tribe Server (ETS) using the software listed before.

## Packages

- apache2
- mysql-server, mysql-common, phpmyadmin
- php5 (php5-common, php5-curl, php5-dev, php5-idn, php5-pear, php5-recode, php5-mysql, php5-pspell, php5-gd, php5-snmp, php5-sqlite, php5-xmlrpc, php5-mcrypt, php5-memcache, php5-mhash, php5-ming, php5-ps)
- nodejs, nodejs-legacy, npm
- redis-server
- haproxy
- drush
- git

Only for games managers:

- python-pip
- sqlite3 libsqlite3-dev
- php5 (php5-imagick, phantomjs, curl)
- Flask (pip install Flask)
- Pymongo (pip install pymongo)
- Datetime (pip install datetime)
- mcrypt, json (php5enmod mcrypt, php5enmod json)
- forever (npm install -g forever)



## Verify your configuration

- Node.js: commands reply with software version

```
node --version  
npm --version
```

- Redis: command reply with *pong*.

```
redis-cli  
ping
```

## Install Experiment Tribe Server (ETS)

ETS consists of a Drupal front-end, in charge of the whole user interaction, and a core back-end, realized in Node.js. First step of the installation procedure, involves copying data. ETS files are all contained in a directory *xtribe.eu* that you can copy inside your server main directory (*/var/www*, in case you are using Apache). In the same way, you can import ETS database, as it is. Create a user 'xtribe' to access it and give him all privileges (beside GRANT). This user will be adopted by the Drupal front-end to access the database.

**ETS directories** contain data organized, neatly, in subdirectory:

- **drupal**: link to "drupal-7.22" directory
  - **drupal-7.22**: original core Drupal data, as you can download from Drupal website. The directories "drupal/sites" and "drupal/sites/default/files" are link to corresponding directories located in "/xtribe", in order to maintain customizations separated from Drupal installation, which allows the administrator to update Drupal cleanly and easily.
  - **etsweb**: Drupal directory "drupal/sites". Contains all the customized part of a Drupal installation. Here you can find, for example, ETS theme and customized ETS module.
  - **files**: Drupal directory "drupal/sites/default/files". In our case, contains, mainly, game's logos and design.
  - **etsman**: your game's managers. Managers can be kept in a separated server.
  - **ets**: ETS core. "Node" directory contains the core code, realized in Node.js. "Conf" contains all the configuration files you will need to install ETS. "Bin" contains scripts needed to install and run ETS.
  - **etsinstall**: how-to install ETS and an automatic installation script working on Ubuntu.
- ETS database** contains data related to the Drupal front-end.

Once you've copied files, take a moment to change permissions: all files must be owned by the user *www-data*.

```
sudo chown -R www-data:www-data xtribe.eu
```

Change Drupal settings, contained in *drupal/sites/default/settings.php*, in order to reflect the database name, user name and password you chose for the database used to import the original ETS database.

Drupal needs mod-rewrite module enabled for its clean-url to work. If you have a fresh installation of Apache, this module should be disabled, by default. Verify its presence, eventually install it, enable it and restart Apache.

```
sudo apachectl -M //show installed modules
sudo a2enmod rewrite // enable module rewrite
sudo service apache2 restart
```

If you don't own a *www* domain, comment these three lines, adding an hashmark before them, in the file *.htaccess* that you can find, hidden, in Drupal directory.

```
# RewriteCond %{HTTP_HOST} !^dev\. [NC]
# RewriteCond %{HTTP_HOST} !^www\. [NC]
# RewriteRule ^ http://www.%{HTTP_HOST}%{REQUEST_URI} [L,R=301]
```

Define a Virtual Host for your ETS. Copy the file *default* located in */etc/apache2/sites-available* and paste it in the same directory. Rename it *xtribe*. Create a link to it and move it to */etc/apache2/sites-enabled*. Add, to the file you created, these lines, right after the first (*<VirtualHost \*:8080>*):

```
ServerName YourServerName
DocumentRoot /var/www/xtribe.eu/drupal
ErrorLog /var/log/apache2/xtribe-error.log
CustomLog /var/log/apache2/xtribe-access.log combined
<Directory /var/www>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
```

Modify your *DocumentRoot* and *Directory* directive, accordingly to the location of your ETS site, and feel free to put the logs wherever you want to. The *Directory* directive is meant to assure that the *.htaccess* contained in Drupal directory will be considered: otherwise it could be ignored.

If you don't own a domain, add your server name to your host file (*/etc/host*).

Enable the site just defined and disable the default, so ETS will be the main. Restart your server.

```
sudo a2ensite xtribe
sudo a2dissite default
sudo service apache2 restart
```

Check the enabled sites to be sure the configuration is correct.

```
sudo apache2ctl -S
```

ETS server listen on 8080 port, due to configuration issues about Haproxy, so please maintain this configuration. Check it in your *sites-available* file and in Apache */etc/apache2/ports.conf* file.

Assuming Redis is correctly installed, copy */xtribe.eu/ets/conf/redis-server* configuration file in */etc/init.d*. This adds to the default configuration only one line, needed to notify to the system when redis has started, for ETS needs to launch some processes when redis is already up.

```
Initctl emit redis-server-started
```

Make sure that the directive *daemonize* in *etc/redis/redis.conf* is set to *yes*, so that Redis will be run as a daemon.

Some ETS processes need to be started on boot. Assuming Upstart is installed. Copy in your */etc/init* directory */xtribe.eu/ets/conf/ets-server.conf*. Check if the variables declared here are correctly set. For example, *NODE\_BIN* defines the directory where Node.js is installed. Replace it with the right one.

Copy in your */etc/init* directory */xtribe.eu/ets/conf/ets-haproxy.conf*. Haproxy will be started on boot, using configuration contained in */xtribe.eu/ets/conf/haproxy.conf*.

Copy in your */etc/init* directory */xtribe.eu/ets/conf/ets-managers.conf* to start all your managers during boot, if your managers reside on the same machine ETS is running.

Execute the installation script, contained in */xtribe.eu/ets/bin/server.sh*:

```
sudo bash server.sh install
```

The script will create a file *config.json* and prompt its content for you to modify it. Fill the entries, according to your server configuration and change the ownership of this file to *www-data*.

This configuration must be reported to the Drupal front-end. Open ETS Drupal website and log in with administrative permissions. Access the administrative menu *Configuration*, then *ETS Service Setup*. Fill the entries, so that this configuration reflects your *config.json*.

Now, ETS is Up and Running.