Setting up a framadate server on FreeBSD

Introduction

Framadate is an online service for planning an appointment or making a decision quickly and easily. No registration is required. It is an open source equivalent to services such as Doodle.

This document details the steps required to set up a **framadate** server on FreeBSD. It was written because there is quite a lot involved, and there are a lot of small things which existing instructions either don't explain at all, or which are explained in terms of Linux systems! This was done on FreeBSD 13.2; earlier systems should be fine as long as the various packages are reasonably recent.

See the end of this document for license details.

The emphasis here (mostly) is not just to be a recipe, but to explain *why* something is done. The setup is actually quite quick when it is all in one place.

This is a long document, because:

- It contains a lot of detail
- It contains explanations as well as directions
- It includes some optional enhancements
- Many pathnames are given in full, to avoid ambiguity

Please note the changelog at the end of the document.

Packages required

The following packages (and their dependencies) are all required, so install them, or build from ports:

- archivers/unzip (unzip utility, needed for the **framadate** release). [you can also use `tar` for this if you know how]
- lang/php81 (PHP, including the FastCGI Process Manager). Needs to be at least version 5.6. [the package **php81** is used here, but other versions should be fine]
- some PHP extensions:
  - security/php81-filter
  - devel/php81-intl
  - converters/php81-mbstring
  - databases/php81-mysqli
  - databases/php81-pdo
  - databases/php81-pdo_mysql
  - www/php81-session
- databases/mysql57-server (database server); this will install the client (databases/mysql57-client) as a dependency. This assumes the database is on the same system as **framadate**. [the package **mysql57-server** is used here, but other versions should be fine]
- www/nginx (HTTP server) [in principle, another server could be used, but that is outside the scope of this document]
- security/py39-htpasswd (htpasswd generator). [this may vary in terms of the Python version]

It is strongly suggested that all of these packages are installed **now**, as there are one or two cross dependencies. There are no port options to worry about; in all cases, the defaults should be OK. Naturally, a number of dependencies may also be installed, e.g. openssl and python.

You can check that all of the PHP packages have been picked up by entering the command:

```
$ php -m
```

Also required is the latest **framadate** release, which can be downloaded from:

https://framagit.org/framasoft/framadate/framadate/-/releases

Be sure to download the release ZIP file, *not* the source code. A quick check should be made, using `unzip -v`, to ensure that the file is the right one, is to see that there is a `vendor` subdirectory; that will be present if it is the correct file. Don't actually unzip the file yet.

Overview

The operation of **framadate** is fairly simple. The web server handles the user interface, passing requests to and from the php-fpm process for CGI processing. Database access from php-fpm is handled by the **mysql** server.

All this means that configuration is very distributed; hence, each component will be set up separately below. Work steadily through it, and all should be fine.

In all cases, pathnames specific to a normal FreeBSD installation have been used, in accordance with `hier(7)`.

The whole process can be done as root, or using `sudo`.

Domain names

First, a fully qualified domain name must be chosen for the server.

The base domain name used here is the well-worn `example.com`, and we choose to call the server `fram.example.com`. Other names can, of course, be used.

Conventions

The following conventions are used to save space and multiple explanations below:

Abbreviation	Meaning	Example
BASEDN	Base domain name	example.com
FQDN	Fully qualified domain name	fram.example.com
ADMIN	Username for **framadate** admin operations	framadmin
USER	Username for application access to database	framadate
DATABASE	Name of database used by **framadate**	framadate
SECRET1	Password for use by `root` on `mysql`	Xyzjkr13+-99
SECRET2	Password for use by **USER**	ab6$z/lsp9,y
SECRET3	Password for use by **ADMIN**	Df7xz/$to8,w
ADMIN-EMAIL	Email address for administrator	root@**BASEDN**

The passwords should be *secure*. Reasonable passwords can be generated in many ways; here is a suitable shell script:

```
#!/bin/sh
# generate random password
dd if=/dev/random count=1 bs=12 2>/dev/null | b64encode - | \
      sed -e 's/=*$//' -e '/^begin/d' -e '/^$/d'
```

To save confusion, there is no reason why you cannot add stuff to the passwords to indicate which is which, e.g.:

```
SECRET2-ab6$z/lsp9,y
```

One point of notation; some examples of file contents have a shaded background. Groups of shaded lines should be kept on one line in the actual file.

SSL/TLS

A certificate must be obtained for the **FQDN** web server; how to do this is outside the scope of this document.

One easy way is to use Lets Encrypt for free certificates; see https://letsencrypt.org. The FreeBSD package for this is `security/py-acme`.

Whatever you use, you *must* make sure that the certificate is renewed and replaced before it expires! In the case of Let's Encrypt, this is especially important as they only last three months. See the Let's Encrypt documentation for details on how to do this (of course, if you are not using Let's Encrypt you will have to make other arrangements; even a `cron` job sending mail, or the FreeBSD `calendar` program, will work for reminders). The instructions are not replicated here because it's better to have one up to date source.

Setup

Setup is a little involved, because there are four packages to configure. These are described separately. All configuration is done as root, or using `sudo`.

mysql

Installation

First, ensure that the mysql server (and thus client) are installed, either from packages or from ports. Use the default port options if this is done from ports.

Initial configuration

Edit the file `/usr/local/etc/mysql/my.cnf`. Add the following line to the `[mysqld]` stanza, perhaps just after the `log-bin` line:

```
Log-error = mysqld.err
```

Note that log-error contains a *minus sign*. This merely gives the error log a more meaningful name. Also disable the slow query log (it isn't really needed here) by changing `1` to `0` on this line:

```
slow-query-log = 0
```

There are some further changes that are probably useful. Reduce the valiue of `table_open_cache`, perhaps to 8000 or less. You may also wish to include this line under `[mysqld]`:

```
tls_version = TLSv1.2
```

(or even `TLSv1.3`, if supported).

Startup

Enable and start the server, noting that the character between `mysql` and `server` differs between the two lines:

```
$ sysrc mysql_enable="YES"
$ service mysql-server start
```

The first time the server is started, it may take a little while; do not panic.

The initial password for first time use of the mysql server may now be found in `/root/.mysql_secret`, or it may be empty. You will set or change this soon.

Further configuration

It is now a good idea to secure the installation; a command is provided for this. You will need the new password that you are going to use (**SECRET1**):

```
$ mysql_secure_installation
```

Recommended settings are:

- enable the password validator, preferable at the strongest level; the password to use is **SECRET1**
- remove anonymous users
- disallow remote root login
- remove the test database
- reload the privilege tables

Now connect to the server:

```
$ mysql -u root -p
Enter password:
```

Use the password from `/root/.mysql_secret` here; this password has now expired, so a new one is necessary (ignore this if you already changed it during validation), this is **SECRET1**. Enter the following commands to `mysql`; note that all statements should be terminated by a semicolon, or you will get a puzzling continuation prompt (in such a case, just enter the semicolon). The first line may not be needed but does no harm:

```
ALTER USER 'root'@'localhost' IDENTIFIED BY 'SECRET1';
ALTER USER 'root'@'localhost' PASSWORD EXPIRE NEVER;
```

The second line is optional but is probably convenient. Note that `/root/.mysql_secret` is not updated!

Now create the database used by **framadate**. Enter this command (still in `mysql`):

```
CREATE DATABASE IF NOT EXISTS DATABASE DEFAULT CHARACTER SET = utf8mb4
      COLLATE = utf8mb4_unicode_ci;
```

Enter these commands, taking care not to omit the `.*` after the database name:

```
CREATE USER 'USER'@'localhost' IDENTIFIED BY 'SECRET2';
GRANT ALL PRIVILEGES ON DATABASE.* TO 'USER'@'localhost';
```

The database setup is now complete; enter the `QUIT` command to quit `mysql`.

It may be useful to create the file ~/.my.cnf, containing the following; it will provide autologon when starting the client, although it is less secure:

```
[client]
host = localhost
user = root
password = SECRET1
```

Logging

The only enabled log will be the error log, which is now named `/var/db/mysql/mysqld.err`.

Before setting up log rotation, it is necessary to create the file `/root/.mylogin.cnf`, which contains obfuscated login information required by the `mysqladmin` command (if it is to be used non-interactively). Enter the following command, and respond to the `Enter password:` prompt with **SECRET1**:

```
mysql_config_editor set --host=localhost --user=root --password
Enter password:
```

Now put this short shell script somewhere (e.g. `/root/bin/mysqld-rotate`, which is assumed below):

```
#!/bin/sh
mysqladmin flush-logs error
# end
```

then enter the command:

```
$ chmod +x /root/bin/mysqld-rotate   # using the appropriate pathname
```

Create the file `/usr/local/etc/newsyslog.conf.d/mysqld`, containing:

```
/var/db/mysql/mysqld.err mysql:mysql 600 7 30 * JRC /root/bin/mysqld-rotate
```

Modify this as desired. Then enter the command:

```
$ newsyslog -C /var/db/mysql/mysqld.err
```

to create the logfile with the correct ownership and permissions. Yes, this really is in `/var/db`.

Now restart the server to pick up the changes, with:

```
$ service mysql-server restart
```

Check `mysqld.err` for any errors.

PHP

Installation

First, ensure that `PHP` is installed, either from a package or from ports. Use the default port options if this is done from ports.

Next, ensure that the following extensions are also installed (all of the names will start with `php`*`xx`*`-`):
`filter, hash, intl, json, mbstring, mysqli, openssl, PDO, pdo_mysql, session.`

Use this command to check that all of the correct modules are active:

```
$ php -m
```

Configuration

On FreeBSD, the php configuration directory is `/usr/local/etc`, and the work takes place there. The configuration file is named `php.ini`; create this by copying directly from `php.ini-production`:

```
$ cd /usr/local/etc
$ cp php.ini-production php.ini
```

There are some necessary minor changes to `php.ini`. First, un-comment the line containing `date.timezone`, and enter a suitable value. Valid values may be found at:

https://www.php.net/manual/en/timezones.php

(an example might be `Europe/London`).

Still in `php.ini`, find the line containing `session.cookie_httponly`, and ensure that it reads:

```
session.cookie_httponly = 1
```

That completes the basic PHP configuration.

We will be using `php-fpm`, which is the FastCGI Process Manager (automatically installed with `php`). Its configuration files are in `/usr/local/etc/php-fpm.conf` and `/usr/local/etc/php-fpm.d` but the defaults should be fine. However, you can limit access to `php-fpm` by adding this line to `/usr/local/etc/php-fpm.d/www.conf`:

```
listen.allowed_clients = 127.0.0.1
```

Logging

`php-fpm` logs by default to `/var/log/php-fpm.log`; see `php-fpm.conf` to change this. The default logfile name is assumed below.

Create the file `/usr/local/etc/newsyslog.conf.d/php-fpm`, containing:

```
/var/log/php-fpm.log 600  7  50  *   JC  /var/run/php-fpm.pid  SIGUSR1
```

Modify this as desired. Then enter the command:

```
$ newsyslog –C /var/log/php-fpm.log
```

to create the logfile with the correct ownership and permissions.

Starting up

Enable and start the PHP FastCGI Process Manager. Enter the following commands, noting that the character between `php` and `fpm` differs between the two lines (but the other way round to `mysql`!):

```
$ sysrc php_fpm_enable=”YES”
$ service php-fpm start
```

Check the logfile for any errors.

nginx

Installation

It is assumed that nginx has just been installed; modify these instructions appropriately if it is already running. There are many port options, but it is safe to take the default settings for now.

Kernel module

FreeBSD includes a kernel module that optimises HTTP requests; it is advisable to use this. Add the following line to `/boot/loader.conf`:

```
accf_http_load=”YES”
```

This will load the module when the system is booted. Meanwhile, to save a reboot right now, load the module manually with the command:

```
$ kldload accf_http
```

Configuration

On FreeBSD, the nginx configuration directory is `/usr/local/etc/nginx`, and the work takes place there. The configuration file is named `nginx.conf`.

What follows is a *complete* configuration file; modify as necessary, or merge it with any existing configuration. It is assumed that a certificate has been obtained from Lets Encrypt for the server.

```
user                            www www;

worker_processes                auto;
pid                             /var/run/nginx.pid;

events {
      use                       kqueue;
      worker_connections        256;
}

http {
      include                   mime.types;
      default_type              application/octet-stream;
      sendfile                  on;
      keepalive_timeout         65;
      log_format                hostcombined '$remote_addr - $remote_user host'
                                ' [$time_local] "$request" $status'
                                ' $body_bytes_sent "$http_referer" "http_user_agent"';
      access_log                /var/log/nginx/access.log hostcombined;
      error_log                 /var/log/nginx/error.log error;

# FQDN HTTP server – redirects to HTTPS
server {
      listen                    80 default_server;
      return                    301 https://$host$request_uri;
}

# FQDN HTTPS server
server {
      server_name               FQDN;
      listen                    443 ssl http2 accept_filter=httpready;
      root                      /usr/local/www/framadate;
      index                     index.php;

      add_header        Content-Security-Policy "default-src 'self';
            script-src 'self' 'unsafe-inline' 'unsafe-eval'; object-src 'none';
            style-src 'self' 'unsafe-inline'; font-src 'self'; img-src 'self'";
      add_header                Referrer-Policy "strict-origin";
      # Avoid XSS attacks through incorrect MIME types
      add_header                X-Content-Type-Options "nosniff";
      # Avoid XSS attacks for older browsers
      add_header                X-XSS-Protection "1; mode=block";

      ssl_certificate   /usr/local/etc/letsencrypt/live/fram.tavi.co.uk/fullchain.pem;

      ssl_certificate_key /usr/local/etc/letsencrypt/live/fram.tavi.co.uk/privkey.pem;

      location ~ ^/(\.git)/ {
            deny                all;
      }

      location ~ /\. {
            deny                all;
      }

      location ~ ^/composer\.json.*$|^/composer\.lock.*$|^/php\.ini.*$|^/.*\.sh {
            deny                all;
      }

      location /admin/ {
            auth_basic          "Restricted access";
            auth_basic_user_file /usr/local/etc/nginx/htpasswd;
```

```
        location ~ \.php$ {
              fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
              include     fastcgi_params;
              fastcgi_pass      127.0.0.1:9000;
        }
        try_files         $uri $uri/ =401;
    }

    location / {
        rewrite "^/admin$" "/admin/" permanent;
        # Clean URL
        rewrite "^/([a-zA-Z0-9-]+)$" "/studs.php?poll=$1" last;
        rewrite "^/([a-zA-Z0-9-]+)/action/([a-zA-Z_-]+)/(.+)$"
        "/studs.php?poll=$1&$2=$3" last;

        rewrite "^/([a-zA-Z0-9-]+)/vote/([a-zA-Z0-9]{16})$"
        "/studs.php?poll=$1&vote=$2" last;

        rewrite "^/([a-zA-Z0-9]{24})/admin$" "/adminstuds.php?poll=$1" last;

        rewrite "^/([a-zA-Z0-9]{24})/admin/vote/([a-zA-Z0-9]{16})$"
        "/adminstuds.php?poll=$1&vote=$2" last;

        rewrite "^/([a-zA-Z0-9]{24})/admin/action/([a-zA-Z_-]+)(/([A-Za-z0-9]+))?$"
        "/adminstuds.php?poll=$1&$2=$4" last;
        try_files $uri /index.php;
    }

    location ~ \.php$ {
        fastcgi_param     SCRIPT_FILENAME $document_root$fastcgi_script_name;
        fastcgi_index     index.php;
        include           fastcgi_params;
        fastcgi_pass      127.0.0.1:9000;
    }
} # server
} # http
```

Access to the admin area is restricted by password. The password file is in the *htaccess* format used by Apache, but there is no need to install Apache to create a password file;. Use the `htpasswd.py` command, which is provided in the `py39-htpasswd` package. It is suggested that the password file be placed outside the web directories; a convenient place is the `nginx` configuration directory. Enter the command:

```
$ htpasswd.py -bc /usr/local/etc/nginx/htpasswd ADMIN SECRET3
$ chmod 600 /usr/local/etc/nginx/htpasswd
$ chown www:www /usr/local/etc/nginx/htpasswd
```

Check the configuration with the command:

```
$ nginx –t
```

Logging

Logging is defined in `nginx.conf` using the `access_log` and `error_log` directives. It is possible to change the verbosity level of the access log; see the nginx documentation for details.

Create the file `/usr/local/etc/newsyslog.conf.d/nginx`, containing:

```
/var/log/nginx/*  600  7     *    @T03  JGC   /var/run/nginx.pid SIGUSR1
```

Modify this as desired. Then issue the commands:

```
$ newsyslog –C /var/log/nginx/access.log
$ newsyslog –C /var/log/nginx/error.log
```

to create the logfiles with the correct ownership and permissions.

Starting up

Enable and start nginx:

```
$ sysrc nginx_enable="YES"
$ service nginx start
```

Check the logfile for any errors.

More information about nginx may be found at https://www.nginx.com.

framadate

Installation

framadate is not yet a FreeBSD port, so it must be installed manually.

Make `/usr/local/www` the current directory, and unzip (or use `tar` if you know how) the **framadate** distribution into it. Set the owner and group of the newly created `framadate/` directory (and its contents) to `www` (the same user as used at the start of `nginx.conf`):

```
$ cd /usr/local/www
$ unzip framadate.1.19.1       # or whatever version
$ chown -R www:www framadate
```

Then make sure that the file `admin/stdout.log` exists, and set the mode of that file to owner access only by `www`.

```
$ cd framadate
$ touch admin/stdout.log
$ chmod 600 admin/stdout.log
$ chown www:www admin/stdout.log
```

Configuration

First, restart `php-fpm` to pick up any `php.ini` changes:

```
$ service php-fpm restart
```

Configuration is partly done inside **framadate** itself, but part of it has to be done manually. Start by browsing to https://FQDN; after being prompted for credentials (**ADMIN** and **SECRET3**) you should see a welcome screen from **framadate**, indicating the results of a configuration check. Correct any items flagged (if you wish) and go to the next screen. There are a number of things to enter:

- The displayed name for this installation, for example `fram`.
- The administrator email address (**ADMIN-EMAIL**).
- The respond-to email address (optional); this is used when users answer email messages sent by the system. It can be **ADMIN-EMAIL** too, or something else.
- The language to use; there are several options.
- Clean URL option; clean URLs displayed in the browser's address bar.
- Connection string (mandatory): The string used to configure the connection to the database. In the example shown on screen, replace `<HOST>` by the host name of your database server (for instance, `localhost`). Replace <SCHEMA> with **DATABASE**. A real example might be:
  `mysql:host=localhost;dbname=framadate;port=3306`
- Database user (normally `root`).

- Password; this is the password of the database user `root` (**SECRET1**).
- The prefix of the tables used in the database. Leave this alone, set to `fd_`.
- Migration table; the table used for database migrations. Leave this alone too.

Once the form has been submitted, the file `app/inc/config.php` will be generated, and you will be redirected to the migration page. This handles:

- Installing the whole database
- Updating the database when **framadate** has new updates. Initially, there is nothing to do.

That concludes the first part of the configuration. The second part involves editing the file `app/inc/config.php` to make some manual changes; typically these will be

- Set `APP_URL='`**FQDN**`'` (mandatory)
- Set the SMTP options to suit your email setup (mandatory).
- Setting `PURGE_DELAY`, `MAX_SLOTS_PER_POLL`, and `TIME_EDIT_LINK_EMAIL`. The defaults are reasonable, but the comments in `config.php` will explain these if you wish to alter them.
- Lastly, the general configuration section at the end of `config.php` allows the omission of some items on the main **framadate** page, as well as a few other useful settings. These setting are advised for production:

```
show_the_software = false
show_cultivate_your_garden = false
```

That is the end of the **framadate** configuration. Restart `php-fpm` with:

```
$ service php-fpm restart
```

Logging

By default, **framadate** logs to the file `/usr/local/www/framadate/admin/stdout.log`. Create the file `/usr/local/etc/newsyslog.conf.d/framadate`, containing:

```
/usr/local/www/framadate/admin/stdout.log www:www 600 3  30 *  JNC
```

Modify this as desired. Then issue the command:

```
$ newsyslog –C /usr/local/www/framadate/admin/stdout.log
```

to create the logfile with the correct ownership and permissions.

Starting up

There is no startup to do, as these are merely pages loaded by `nginx`.

Testing

To test, open a browser and go to **http://FQDN**. You should see the **framadate** main screen!

Further work

You may find that the defaults for optional parameters (when creating a poll) are not what you generally want. These can be changed, but it involves changing a file which will be overwritten when any upgrade to **framadate** is applied. It is suggested that the following procedure is adopted (the file `Form.php` can be found in `app/classes/Framadate`):

```
$ cp Form.php Form.php.orig
$ # edit Form.php
$ diff -u Form.php.orig Form.php > /root/Form.diff
```

After reinstallation or upgrade, enter:

```
$ patch < Form.diff
```

to re-apply the edits. Be sure to check and fix the ownership and permissions on `Form.php` after doing this.

The optional parameters are part of the class definition for the Form class, which can be found in `/usr/local/www/app/classes/Framadate/Form.php`. They are:

- `$receiveNewVotes` (notify poll administrator when new vote is made)
- `$receiveNewComments` (notify poll administrator when new comment is posted)
- `$use_ValueMax` (impose a limit of voters per option; the limit is left open for filling in)
- `$hidden` (only the poll maker can see the poll's results)
- `$use_customized_url` (author wants to customize the URL; the URL is left open for filling in))
- `$use_password` (password will be needed to access the poll)
- $results_publicly_visible

See `Forms.php` for more details on the meaning of each of these. All of them are declared without an initial value, so they take a `NULL` value, which is interpreted as `false`. To change this, add '`= true`' before the semicolon, e.g.:

```
$hidden = true;
```

There is also the setting of the three way option for permissions, which is set a little further down in the line:

```
$this->editable = Editable::EDITABLE_BY_OWN;
```

The possible values here are `EDITABLE_BY_OWN`, `EDITABLE_BY_ALL`, and `NOT_EDITABLE`.

License

Changelog

2024-10-06	Revision 3: Minor clarifications and corrections
2024-03-08	Revision 2: Minor clarifications, and improvements to security
2023-12-18	Revision 1: For PHP 8.1 and other port updates
2021-01-23	Added hint about using `tar` instead of `unzip`
2021-01-11	Initial version

Bob Eager
bob@eager.cx
October 2024