SwimAdmin 0623 system administration

If you are running a server which you installed yourself from a downloadable image, then you will be responsible for administering that server. This document provides the information that you will need to carry out any administration that may be required. In practice, however, it is likely that your server should be able to run for extended periods without intervention.

If you have a maintenance contract, you can largely ignore this manual. However, if you enable the WordPress site, whoever is responsible for the site will need the information in Section 4.

1 INTRODUCTION

Your server is set up to run automatically, without any external intervention. However, this does require some external monitoring to confirm that the server is running as expected. The server will email you regular status updates and, in practice, the job of the system administrator (or 'sysadmin') is simply to:

1. Confirm that these status emails arrive, and carry on arriving. The most important of these are the 'heartbeat' emails (the server will send you two of these every day, to confirm that it is live and functioning); and
2. To carry out backups.

Historically, SwimAdmin servers have run for several years without rebooting, and without any failures. However, more recent versions have more aggressive security update policies, and additional functionality, and will generally automatically reboot every few weeks. This process should be entirely transparent to you, but you will need to keep an eye on the status emails to confirm that the server is operating normally.

Section 2 below lists the emails that you will receive from the server during normal operation. Note that these are not all related to 'system administration': in some circumstances, you will also receive reports of payment runs for gala entries, which you will probably need to forward on to your fixtures secretary or club treasurer.

Section 3 covers backup procedures.

Section 4 covers WordPress administration. If you have WordPress enabled (it is disabled by default) you will need to find somebody who is familiar with WordPress to develop or port your website. This section covers the information required to let this person log into your server, or to copy files to it, and relevant security information.

The remaining sections are a more detailed look at the implementation of the system. You don't need this information, and can generally ignore it. However, this may be of interest if you have any Linux administration experience, and need to change anything.

1.1 WHAT IS THE 'SYSADMIN', ANYWAY?

During club configuration, you entered the name and email address of a club official in the 'Site administrator' section. The SwimAdmin app was set up with a single user, with username `admin`, with these details. However, the details are also re-used during the setup of the Linux server itself, in such a way that this email address receives any internal emails generated by the server. This effectively makes this user the sysadmin.

Note that the sysadmin is not a real Linux user; he or she cannot log in. The only user accounts are for `root`, `swimadmin`, and `wpuser`.

If you change the identity or the email address of the club `admin` user in the normal way (see section 1.4.1 of the Reference Manual), the status emails will be sent to the new email address. The sysadmin is therefore effectively tied to the club `admin` user.

2 STATUS EMAILS

You will receive at least two emails a day, and potentially more. The full list is given below.

2.1 HEARTBEATS

The server sends two emails a day, at approximately 7AM and 7PM. These emails confirm that the system is running, and include basic information about the server status. **If you do not receive a heartbeat email, then there is an issue which must be investigated**. See section 8 below.

2.2 SECURITY CERTIFICATES

The server requires a security certificate in order to use the 'https' protocol. These certificates are issued by Let's Encrypt, and are valid for 3 months. In practice, the server automatically renews the certificate one month before the renewal date, so renewal is on a two-monthly cycle. You will get 3 emails during the renewal process, which should arrive within a minute or two of each other. These should inform you that (1) a certificate renewal is being initiated, and that (2) the certificate has been renewed, and that (3) the new certificate has been installed.

2.3 SECURITY UPDATES

The server is set up to automatically install security updates, and will email you when it has done so. With the current security settings, this happens fairly frequently (every 8 or 9 days). Some security updates will automatically reboot the server; if so, the reboot is scheduled for approximately 3AM. A security update which requires a reboot will contain 'reboot required' in the subject line.

2.4 REBOOTS

The server will email you whenever it reboots. This should only happen after relevant security updates (currently, about every 20 days). This email will generally arrive the day after the relevant security update email. **You should investigate if you receive a reboot email which you were not expecting**; this probably means that there has been a problem at your datacentre.

2.5 GALA PAYMENT RUNS

If you have enabled Stripe payment processing, *and* you have set a deferred payment method (methods Stripe 2 and Stripe 3), then the server will email you reports from gala payment runs. The server carries out a check just after midnight, every night, for online entry windows which have just closed. If one is found, the server will carry out a payment run. **You will need to be aware of the closing dates for gala entries, and you will need to confirm that you have received a payment report.** You will probably also need to forward the report to your fixtures secretary or club treasurer.

2.6 OTHERS

You will receive an email if the server internally generates an email to a specific user (`postmaster`, `webmaster`, `root`, or `swimadmin`). This should not happen in normal usage.

3 BACKUPS

A number of different backup strategies are described below. None of these carry out a 'complete' backup of your server (in other words, a backup which could be restored to a bare-metal server). The `rsync` (3.2) and `burp` (3.3) procedures are intended to save only the data which is required to customise a newly-installed VPS (in other words, about 100MB of data, rather than 4GB). In the worst-case scenario, if you lose the entire server, the restore procedure is therefore to install a new VPS, and then restore your `rsync` or `burp` backup to the new image. You will be able to restore an 0623 backup to any current or future 0623 image.

3.1 DATABASE BACKUP

The club database is a MySQL file, which is stored at `/var/www1/ABCD/sqlite/ABCD.db` (for club 'ABCD'). This file can be downloaded from the Administrator menu, by selecting `Administration > File download`, and then `SQL database`. A previously downloaded database can be uploaded (*overwriting* the current club database) from the `Administration > File upload` page.

This is not, by any means, a comprehensive backup. However, this file contains your member and official databases, your swim times, and your club records, among other things. You should consider downloading the database before carrying out any large-scale club changes. If you make any errors, you can then simply revert to the old database version by uploading your saved file.

The club database does *not* include your WordPress or Wiki files, or any emails stored on the mail server. To back these up, you will need to use one of the two methods described in 3.2 and 3.3 below.

The database is a standard MySQL database. If necessary, you can edit it using the `sqlite3` program. You can find more details at the SQLite website (`https://www.sqlite.org/index.html`).

3.2 RSYNC

`rsync` is a Linux file copy program which is widely used for backing up large amounts of data over a network. Backups are initiated by running `rsync` on a remote (Linux) computer, which communicates with `rsync` on your club server.

The remote `rsync` must be run as root, and requires key-based `ssh` authentication in order to communicate with your server. To enable this, you will have to add your own root public key to the `authorized_keys` file in `/root/.ssh` on your club server. By default, this file will already contain the public key for Maia EDA; you can simply replace this key, or append your own.

In principle, this method is simple and straightforward. However, backing up a running server is problematical. If the web server (Apache) is running, the databases (both MySQL and SQLite) are not guaranteed to be in a consistent state. Similarly, if Dovecot is running, your mailboxes may not be in a consistent state at the point at which `rsync` reads them. To handle this, you must log in to your server, and stop these services before the backup, and then restart them after the backup, using the procedure below. Note that this is not necessary for either 3.1 above, or 3.3 below.

1. On the club server, shut down the relevant services prior to the backup:

```
swimadmin@swimvps $ sudo backup-prepost pre
```

2. Wait 2 minutes, and run rsync on the backup computer:

```
root@backup # rsync -avriXPz --delete \
  --files-from=swimadmin-files.txt -e 'ssh -p7901' \
  myswimclub.uk:/ /backups/myswimclub.uk
```

3. On the club server, restore the relevant services post-backup:

```
swimadmin@swimvps $ sudo backup-prepost post
```

If you do not complete step (3) your server will show a 'maintenance' page rather than the normal SwimAdmin front page.

In this example, the required SwimAdmin files will be stored at `/backups/myswimclub.uk` on your backup computer; change this as required. The files to be backed up are listed in `swimadmin-files.txt`. You can copy this from the corresponding `burp` file, which is located at `/etc/burp/clientconfdir/incexc/swimadmin-includes`. However, you must remove the `include=` prefix at the start of each line.

About 100MB must be backed up, plus any Dovecot mailboxes you may have (which could potentially amount to several GB). `rsync` backups are differential, so backups will normally complete in a few seconds.

3.3 BURP

Burp ('BackUp and Restore Program') is a network backup utility. If you are maintaining your own VPS (the 'client'), one suitable client configuration is already present at `/etc/burp/burp.conf`. To enable this configuration, you will need to change 'enabled' from '0' to '1', and enter the address and port numbers for your own burp server; see the file comments for details.

You will also need to install a burp server on another computer. An example server configuration is given in `/etc/burp/burp-server.conf`, together with the `clientconfdir` directory (these are of course unused by your VPS, which is the burp *client*; however, they can be copied to your burp *server*).

To enable the server, you should change the 'listen' and 'listen_status' port numbers as required, and set 'directory' to the location of your backup directory on the server. `clientconfdir` contains a file named `swimvps.example.com`. You should change this for your own domain (`swimvps.myswimclub.uk`, for example).

The list of files and directories to back up is in `clientconfdir/incexc/swimadmin-includes`. You can re-use this file for `rsync` (3.2), but you will need to manually remove the `include=` prefix at the start of each line.

A cron job runs `burp` every 20 minutes. If you have enabled the client configuration, it will attempt to communicate with the server, which then decides whether or not to initiate a backup. This

process is completely automatic and transparent. With the default configuration settings, a backup will take place every night at some point between 4 and 5AM. The server will keep 7 daily backups, and then 4 weekly backups, and then 6 monthly backups. From the client, you can see a list of backups by running `burp -a l` as root; the status monitor will give additional information.

When the client first attempts to contact the server, the server will create SSL keys, and carry out a key exchange with the client. This process is potentially complex and should be monitored to confirm that it completes correctly. You can find further details at `https://burp.grke.org/`.

3.4 OTHERS

This section is included for completeness only. It should not be considered to be a viable alternative to 3.2 or 3.3 above.

A large part of your system can be backed up simply by copying `/var/www1/` to your local system, using `rsync` or a Windows file copy client such as WinSCP. This includes the SwimAdmin files, DokuWiki, and Roundcube. This is not a particularly useful backup, but may be sufficient while you are developing another backup strategy. The files in this directory are all owned by the swimadmin user, so you do not need root permissions, and can carry out the copy as user `swimadmin`, with swimadmin's password.

If you are running on Linux, this command will synchronise your remote VPS `/var/www1/` directory to your local `www1` directory, deleting anything in the local `www1` which no longer exists on the remote system:

```
$ rsync -avriXPz --delete -e 'ssh -l swimadmin -p 7901' \
        myswimclub.uk:/var/www1/ www1
```

Note that the `rsync` '-X' option preserves extended file attributes; gala files use Linux attributes to set gala dates. You must make sure that these dates are not lost if you back up a file locally, and then restore it to the server. If you use a Windows client to back up your club directory you should ensure that it handles Linux extended attributes. A loss of the date attribute will cause some confusion in gala listings, but is not a major problem. Note also that your club MySQL database may be inconsistent if a club administrator is changing it while you are attempting to copy it.

Note that this backup does *not* handle:

- Your Dovecot mailbox files and sieve scripts, which are in `/var/mail/`. You need root privileges to access these files
- The Wordpress installation, which is at `/home/wpuser`. These files are all owned by `wpuser`, and so can be copied by `wpuser`
- The MySQL databases, which are at `/var/lib/mysql/`. You will need root permissions to access these files. These include the Wordpress, Roundcube, and Stripe databases.

4 WORDPRESS ADMINISTRATION

The server has an additional regular user named `wpuser`. This user has restricted permissions, and owns all the WordPress files, which are located at `/home/wpuser/wordpress`. The password for this user was set during initial system configuration. You should be careful not to lose this password: the Linux passwords for `root`, `swimadmin`, and `wpuser` cannot be set or changed from the SwimAdmin front-end, and you will have to log into the server to do this (an example is given below for `wpuser`).

4.1 CONFIGURATION

A SwimAdmin administrator can enable or disable the WordPress site from the front-end menus (from `Others > WordPress`):

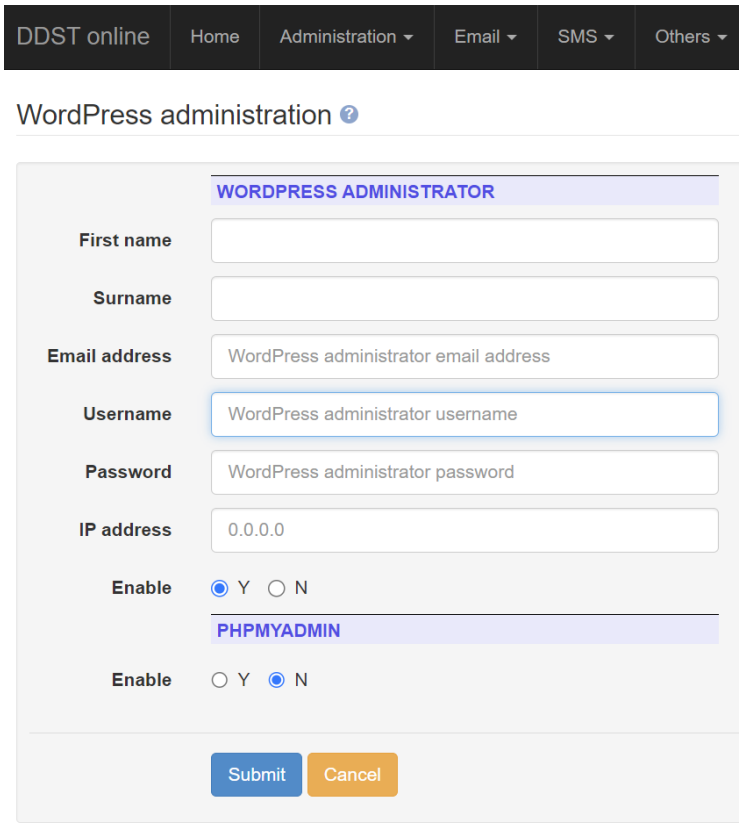

Figure 1: WordPress configuration

During website development, WordPress should be disabled (by setting 'Enable' to 'N'), so that it is not visible at `myswimclub.uk`. However, the site *will* be visible from the single IPv4 address specified under 'IP address', even when WordPress is 'disabled'. If you are responsible for developing the website, you should ask an admin to enter your IP address here. You should have a fixed IP address, if possible; this may or may not work with a temporary address.

Note that you should *not* use the WordPress administration back-end to change the identity of the WordPress administrator. This is maintained in SwimAdmin itself, which also handles any required MySQL updates.

4.2 FILE TRANSFER

To transfer WordPress files to the server, you should use `scp` or a GUI equivalent such as WinSCP. You will need to set the port number to 7901, and set the user to `wpuser`. You will technically be able to write files anywhere in `/home/wpuser`, but you should restrict yourself to WordPress files, in the `wordpress` subdirectory.

If you have access to `scp`, you can copy local file `myfile.php` to `/home/wpuser/wordpress` as follows:

```
$ scp -P7901 myfile.php wpuser@myswimclub.uk:/home/wpuser/wordpress
```

And you can then copy it back as follows:

```
$ scp -P7901 wpuser@myswimclub.uk:/home/wpuser/wordpress/myfile.php .
```

After transferring any files, you should ensure that the files have the correct ownerships and permissions. You can do this by running `wpress-fixup`; see below.

4.3 FILE OWNERSHIPS AND PERMISSIONS

The file ownerships and permissions in the `/home/wpuser/wordpress` directory are maintained by a script called `wpress-fixup`. This requires a single argument, which must be `lock`, or `unlock`. To run this, you will first need to `ssh` into the server. You can do this as follows, from a command line on your local computer:

```
$ ssh -p7901 wpuser@myswimclub.uk
```

When you are logged in, run `wpress-fixup` as follows:

```
wpuser@swimvps $ sudo wpress-fixup lock
```

This will take a few seconds to complete, and will report nothing on success. You can, if necessary, confirm that it has been successful in the normal way (run `echo $?` immediately after running `wpress-fixup`; if this reports `0`, then `wpress-fixup` completed successfully).

You should note that `wpuser` does *not* have sudo permissions, with this one exception.

The policy for WordPress file ownerships and permissions is, briefly, that all files are owned by `wpuser`, with group ownership `www-data`. Regular files are given mode `0644`, while directories are given mode `0755`. There are a number of exceptions which are handled by the script; the script contains documentation if you need any further details.

Note that you should never need to run `wpress-fixup` unless you have changed the WordPress installation is some way (by carrying out an update, for example).

4.4 WORDPRESS UPDATES

In normal circumstances, files are owned by `wpuser`, and are not writeable by WordPress ('WordPress', in this context, actually means the Apache web server, which runs as user `www-data`). This means that you cannot run a WordPress update.

If you need to update WordPress, the simplest solution is to temporarily grant WordPress write permissions on the relevant files. To do this, you will first need to `ssh` into the server (as above), and then run this command:

```
wpuser@swimvps $ sudo wpress-fixup unlock
```

On completion, WordPress will be able to run any updates. **You should not leave WordPress in this state; this is a security risk**. When you have completed any updates, run `wpress-fixup` again.

4.5 PHPMYADMIN

`phpMyAdmin` is installed, and can be enabled as shown in Figure 1 above (it is disabled by default). When enabled, it is visible at `https://myswimclub.uk/phpmyadmin/`. You should log in as user `wpuser`. This user is a MySQL user, and *not* the Linux `wpuser` user, and so has a different password which was set during configuration. The password can be changed by a SwimAdmin administrator, from the `Others > MySQL passwords` page.

`phpMyAdmin` is a security risk, in the sense that anyone can attempt to log in simply by guessing passwords. You should therefore leave it disabled unless you specifically require it during WordPress development.

Note that, in the 0623 release, the back-end database is actually MariaDB, and not MySQL.

4.6 BACKUPS

If you have a maintenance contract (or you have your own administrator who has configured club backups), then your MySQL wordpress table will be backed up remotely every night. However, you should not rely on this during development. You should carry out your own backups with `phpMyAdmin` or `mysqldump`, in the normal way.

4.7 OTHER COMMANDS

When logged in as `wpuser`, you can change your own password as follows:

```
wpuser@swimvps $ passwd
```

'passwd' will ask for your current password, and then prompt for a new password.

5 GENERAL INFORMATION

The 0623 release is based on a standard Ubuntu Server 22.04 LTS install. The SwimAdmin management software is accessed from a browser, with content served by Apache.

The server image also includes a wiki (DokuWiki), WordPress, phpMyAdmin, and Roundcube. These are not required by SwimAdmin and are not supported; they are included simply for user convenience. DokuWiki is always enabled, and can be used for collaborative documentation. The WordPress installation can be enabled or disabled as required (it is disabled by default), but the code itself is always installed.

When WordPress is enabled, the WordPress site is visible at the top level of your domain. If your domain is 'myswimclub.uk', for example, then the WP site will be at `https://myswimclub.uk`, while the WP administration page will be at `https://myswimclub.uk/wordpress/wp-admin`.

The SwimAdmin app is always accessed using your 4-letter club code. If this is ABCD, for example, then the app is at `https://myswimclub.uk/ABCD`, while the administration front-end is at `https://myswimclub.uk/ABCD/admin`.

The wiki is always accessible at `https://myswimclub.uk/wiki`, and the Roundcube webmail interface is always visible at `https://myswimclub.uk/mail.`

When WordPress is disabled, your domain top level (`https://myswimclub.uk`) will show an 'Invalid address' page which directs the user to the app, the wiki, or the webmail interface. However, the WordPress site will still be visible to a single IP address, if so configured (see Figure 1 above).

5.1 IMAGE RESOURCES

The 0623 disk image is currently 20.0 GB, with approximately 3.4 GB used. A SwimAdmin site is unlikely to be heavily loaded, and a single vCPU should be sufficient to run it. The required RAM size is more difficult to define, however. 0.5 GB is certainly insufficient, and applications will report that they have run out of memory. 1 GB is sufficient in testing, but you should use 2 GB on a live site.

5.2 NETWORK

Network management is carried out by systemd-networkd; no legacy network software is installed, and netplan is unused. The network configuration is `/etc/systemd/network/static.network`.

The only ports which are open are:

25	Postfix email routing (SMTP)
80	Apache (HTTP)
443	Apache (HTTPS)
587	Secure SMTP mail submission
993	Secure IMAP mail retrieval
7901	ssh

If you are selecting a VPS provider, you will need to ensure that all of these ports can be opened.

5.3 WORDPRESS

The WordPress install is a default 6.2 install, with no customisation, except that:

1. A password plugin is used to enforce the use of more secure 'bcrypt' passwords. Some other basic security precautions have been taken; the table prefix has been changed from 'wp_', for example. As a WordPress administrator you can, of course, change these defaults.
2. The site is configured so that it is one level down from Apache's `DOCUMENT_ROOT`, in the 'wordpress' directory.

Note that the keys and salts defined in `wp-config.php` should be changed before use. The file contains instructions for running a key generator; this generates the 8 relevant lines which need to be replaced.

6 SSH ACCESS AND PASSWORDS

6.1 REMOTE ACCESS

The vast majority of attacks on public-facing servers occur to ports 22 (for ssh) and 25 (for mail servers). Port 25 cannot be changed, but ssh can be configured to use a different port; SwimAdmin uses 7901.

If you have a terminal program and an ssh installation, you can therefore connect to your server (which is assumed to be 'myswimclub.uk') as user 'swimadmin' as follows:

```
$ ssh -p7901 swimadmin@myswimclub.uk
```

Or you can transfer a file to an arbitrary destination (as long as user 'swimadmin' has write access to that directory) as follows (note the use of the upper-case 'P', rather than lower-case):

```
$ scp -P7901 myfile1 swimadmin@myswimclub.uk:/path/to/destination
```

A file can be read back as follows:

```
$ scp -P7901 swimadmin@myswimclub.uk:/path/to/source myfile2
```

If you are using a GUI file transfer client (such as WinSCP, for example), then you should select the SFTP protocol, on port 7901, with user swimadmin.

When you log in with ssh, you will be told how many updates can be applied. ***Do not be tempted to update the system***. Security updates are installed automatically (see 7 below), and installing any other updates risks causing issues with the basic SwimAdmin functionality.

6.2 ROOT ACCOUNT

It is common on Ubuntu installations for there to be no password on the 'root' account. This prevents users from logging in as root, and gives the impression that the root user has been disabled. These systems require another user to have sudo privileges, to allow that user to carry out administrative operations. However, this is not practical when a great deal of administrative work has to be carried out, and particularly so if it has to be remotely automated.

SwimAdmin therefore creates user-supplied passwords for `root`, `swimadmin`, and `wpuser` during configuration. All three users can potentially ssh into the system, depending on the ssh configuration defined in `/etc/ssh/sshd_config`. This is set up as follows on a new system:

- Users `swimadmin` and `wpuser` can log in with a password. The `ssh` and `scp` commands above, together with any file transfer clients you use, therefore require a password to be supplied
- User `root` must login using *only* public-key authentication

Any remote user who needs to log in as root must have their cryptographic public key stored on the server in `/root/.ssh/authorized_keys`. Remote root logins are not recommended for normal use; this should be allowed only for automated remote system administration. There is, by default, one key in this file (starting with 'AAAAB3Nz'), to allow Maia EDA to carry out remote updates. You can remove this file, if preferred, to completely disable *remote* root access (however, you should keep a backup, in case you want to enable updates in the future).

6.3 SYSTEM USERS

There are three users:

1. `root`. Do not give anyone the root password unless you have a good reason to do so
2. `swimadmin`, who has uid 1000. This user does *not* have sudo privileges (in others words, this user is not in the `sudo` group). Any privileged operations should instead be carried out by `root`. There are entries in the sudoers file `(/etc/sudoers.d/swimadmin`), however, that allow this user to carry out certain administrative actions which require root privileges
3. `wpuser`, who has uid 1001. This account should be used by whoever is responsible for your WordPress site. The site is located at `/home/wpuser/wordpress`. This user has only the privileges required to modify the WordPress site.

7 SECURITY UPDATES

SwimAdmin is configured to download and install Ubuntu security (and *only* security) updates. If a restart is required, this will take place at about 3AM. You will be notified by email when updates have been installed, and if a restart is scheduled. This should not affect server operation, and you should be able to ignore these emails.

The notification email will list any packages that were upgraded. If you need further details, you should look on the Ubuntu Security Notices page (which is currently at `https://ubuntu.com/security/notices?order=newest&release=jammy&details=`), filtering for release 22.04 LTS.

Ubuntu 22.04 was released in April 2022, and will reach end-of-life in April 2032. However, security updates after April 2027 will require the purchase of an 'Ubuntu Advantage' subscription. You may find it more convenient to simply install a later release of SwimAdmin before this date.

8 HEARTBEAT

SwimAdmin is configured to send the sysadmin a 'heartbeat' email twice a day (a little after 7AM, and a little after 7PM). If this email arrives, then the server can be considered to be functional.

The heartbeat is intended to test all major aspects of the server's operation. It is initiated by a cron job, which collects some basic information about the system loading, and then submits this to a (hidden) webpage on the server. SwimAdmin then sends that information to an *external* server. The external server sends an email to the club's admin user, containing the information which has been collected. The SwimAdmin mail server then forwards this mail on to the sysadmin's real address. This procedure confirms that both the website, and the mail relay, are functional.

Historically, the software instructed Gmail to send the heartbeat email, by connecting to the SMTP server at Google (in other words, the 'external server' was gmail.com). However, Google has recently made this increasingly difficult. The software now connects to swimadmin.co.uk instead, which generates the returned email. The information supplied to swimadmin.co.uk is non-identifiable, and simply gives system statistics. This is returned to the admin user in the email. The statistics are presented as follows (the details will depend on the version of SwimAdmin which is running):

Date	26-Apr-2023 07:00:04 AM
Uptime	1 day, 3 hours
Load average	0% (2 CPUs)
RAM available	1.435 GB
Disk used	4.016 GB (48.4%)
Disk available	4.276 GB
Mails sent	29624
Mails received	29707
SwimAdmin version	0522-220

Figure 2: Heartbeat statistics

The uptime is unlikely to exceed 4 weeks for 0623, because of security update reboots. This VPS is running on 2 vCPUs, so the load average is shown over both, and can exceed 100%. However, loading is always likely to be low, particularly at this time of day (the load average is for the previous 15 minutes).

If a heartbeat does not arrive, you should go through this checklist:

1. Is swimadmin.co.uk up? If not, no emails will be sent back to your server
2. Can you directly access the member and administrator pages at your server?
3. Can you ssh to your server?
4. If (2) and (3) fail, you will need to go to your VPS provider's console, and determine whether or not the server is still running. If not, you can issue a reboot from the console.