IrisPanel: Deployment and Administration Manual

Alice WinFrame

December 7th, 2025

Contents

1 Introduction

IrisPanel is a program product designed to assist you in basic maintenance and administration of a World Wide Web server. It provides facilities for managing Virtual Hosts on multiple domains, database creation and management, remote site file access, and Web Server configuration. IrisPanel centralises configuration of several separate programs into a unified control plane.

2 How to provide feedback on this document

Comments, suggestions for improvements, or corrections of inaccuracies are welcomed. You can send feedback about this document to the e-mail address: `feedback@aliceisvery.gay`

3 Understanding the IrisPanel architecture

IrisPanel is composed of separate components, several of which are off-the-shelf free software. It is important that you understand the architecture of the system you are installing, as it will make you to be more effective when administering the installation and troubleshooting issues.

As you can see in Figure 1, IrisPanel itself is not the web server showing the pages to your visitors, nor is it the PHP runtime, nor any other part of the infrastructure of actually serving each request to clients. This is by design - it provides a separation of concerns between the management tooling (in this case IrisPanel), and the operational web server. It also means that anything that would affect the function of the websites is handled by standard components (Apache HTTPd, MariaDB, et cetera), which allows for both existing administrator familiarity, and for possible integration of third-party extensions (though this would not be a supported configuration).

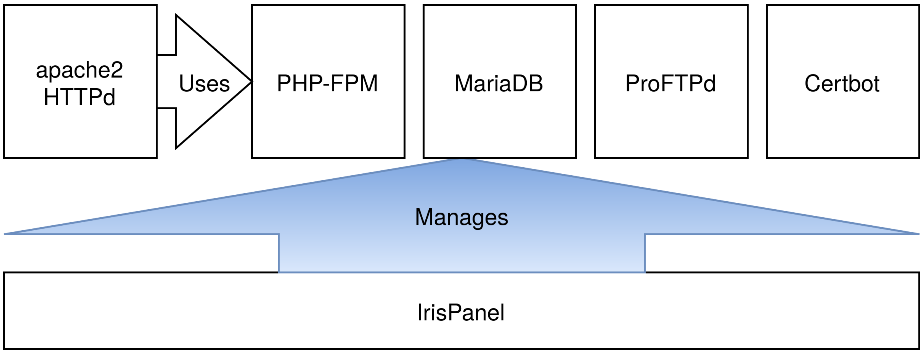

Figure 1: Management architecture of IrisPanel

3.1 Basic operational model

For reasons of both practicality and security, IrisPanel takes pains to isolate users from each other and from the system where possible. This has informed the way that certain parts of IrisPanel are designed, and can account for certain decisions that may seem odd at first glance.

All IrisPanel behaviour starts with a *account*. A account in IrisPanel parlance describes an account which is both registered in IrisPanel's own database, and has a user created on the underlying Linux system. The reason that all IrisPanel accounts also have system users created is for two reasons: to provide a context to run user provided PHP code in, and to allow for jailed FTP access.

A account has 4 main properties which are relevant to this manual: Account UUID, Panel Username, System Username, Account Prefix. The account UUID is used to identify things for which only one exists per account, such as the account's PHP-FPM pool. The panel username is the username which can be used to log into the account on IrisPanel itself, the system username is the name of the Linux user which is assigned to the user, and the account prefix is used to prefix things which must exist outside of IrisPanel's database, for which multiple accounts may have the same name - an example of this may be a database, multiple accounts on the same IrisPanel server may wish to create a MariaDB database called "*wordpress*", and the prefix prevents there from being a conflict.

3.2 How IrisPanel manages external components

While IrisPanel uses 5 external components to accomplish it's function, only three of those need to be reconfigured at runtime - those being the Apache HTTPd, PHP-FPM and MariaDB. Certbot needs no actual configuration, and ProFTPd only needs to be reconfigured once at install time to perform it's function with regard to IrisPanel use. For those which need to be reconfigured at runtime, there are two techniques for altering their state that are used: Configuration file generation, and directly interfacing with the program in question. Apache HTTPd and PHP-FPM both use Configuration file generation, and MariaDB is connected and controlled directly.

3.2.1 PHP-FPM

Each account gets it's own PHP-FPM pool configured, meaning every account also gets it's own PHP worker processes. This is important as it means that PHP code executes in the context of the account's system user. Generated pool configuration files are located in `/etc/php/8.4/fpm/pool.d/`. The configuration file names are `<account uuid>.conf`. Therefore, an example pool configuration path would be
`/etc/php/8.4/fpm/pool.d/ccf90010-ac48-438c-ba4a-7860f245ad46.conf`.

3.2.2 Apache HTTPd

For every domain you configure in IrisPanel, an Apache VirtualHost configuration file is added to */etc/apache2/sites-enabled*. The configuration file names are structured as follows:
`/etc/apache2/sites-enabled/`*<account prefix>_domain-com*`.conf`, so an example full path for the domain "aliceisvery.gay" might be
`/etc/apache2/sites-enabled/29a1_aliceisvery-gay.conf`
This file is regenerated whenever a user makes a change to the configuration of the domain, creates a redirect on the domain, or changes certain account-wide configuration options (such as directory indexing).

```
<VirtualHost *:80>
  ServerName example.com
  RewriteEngine on
  RewriteCond %{SERVER_NAME} =example.com
  RewriteRule ^ https://%{SERVER_NAME}%{REQUEST_URI} [END,NE,R=permanent]
</VirtualHost>

<VirtualHost *:443>
  ServerName example.com
  DocumentRoot /var/web/29a1/public_html/
  <FilesMatch "\.php$">
    SetHandler "proxy:unix:/run/php-fpm-67f141be-20eb-46ab-
b687-20e0d12c2902.sock|fcgi://localhost/"
  </FilesMatch>
  SSLCertificateFile /etc/letsencrypt/live/example.com/fullchain.pem
  SSLCertificateKeyFile /etc/letsencrypt/live/example.com/privkey.pem
  SSLProtocol all -SSLv2 -SSLv3 -TLSv1 -TLSv1.1
  SSLCipherSuite ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-
SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-
ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-AES128-GCM-
SHA256:DHE-RSA-AES256-GCM-SHA384
  SSLHonorCipherOrder off
  SSLSessionTickets off
  SSLOptions +StrictRequire
</VirtualHost>
```

Figure 2: Example VirtualHost configuration file for domain “example.com” with TLS enabled configured to redirect HTTP to HTTPS

It is worth noting the “FilesMatch” directive for PHP files in Figure 2. This hands off responsibility for executing PHP code to the account’s PHP-FPM worker pool, therefore ensuring it executes with the correct user context.

While this example configuration is generally representative of the VirtualHost configurations that IrisPanel will generate, it’s worth noting that the example shown in Figure 2 does not use any of the extra features that IrisPanel provides. Many of those (namely: Redirects, Custom Error Pages, Directory Indexing and Clean URLs) will cause extra directives to be added to the VirtualHost; Similarly, if TLS is not enabled, most of those lines will be present in the port 80 listener, and the 443 listener will not be present, and TLS is enabled but not required, those lines will be shared between the 80 and 443 listeners, instead of the 80 listener being a redirect to HTTPS.

3.2.3 ProFTPd

ProFTPd is used by IrisPanel to provide the FTP service for uploading and managing website files. IrisPanel only actually alters the ProFTPd configuration once, at install time. The configuration changes made are as follows:

- Jail users in home directories (means users only see relevant files, and cannot see the files of other users)
- Only allow users in the group `irispanel-ftp` to connect (the significance of this will be explained shortly)
- Do not require users to have a valid shell to connect (because the system users created by IrisPanel have `/usr/bin/nologin` set as the shell)
- Enable and require TLS for connections

That's all well and good, but you may be wondering how users can enable or disable their FTP access, if ProFTPd doesn't get reconfigured at runtime. The answer is that `irispanel-ftp` group mentioned earlier, when a user enables or disables FTP access for their account, that adds or removes them from the `irispanel-ftp` group. TLS is set up with a self-signed certificate generated by the installer,

3.3 How IrisPanel runs on your server

IrisPanel is a WSGI application, which means it needs a WSGI capable server to run. This role is handled by the excellent Gunicorn WSGI server, which is then reverse proxied through Apache. This means then when you access the IrisPanel interface, your request is proxied by Apache through to Gunicorn, which actually then allows IrisPanel to process the request. The installer program installs a systemd service which starts Gunicorn pointing at IrisPanel, and restarts it if it fails. IrisPanel stores application data in a database in the MariaDB database server that runs on the machine.

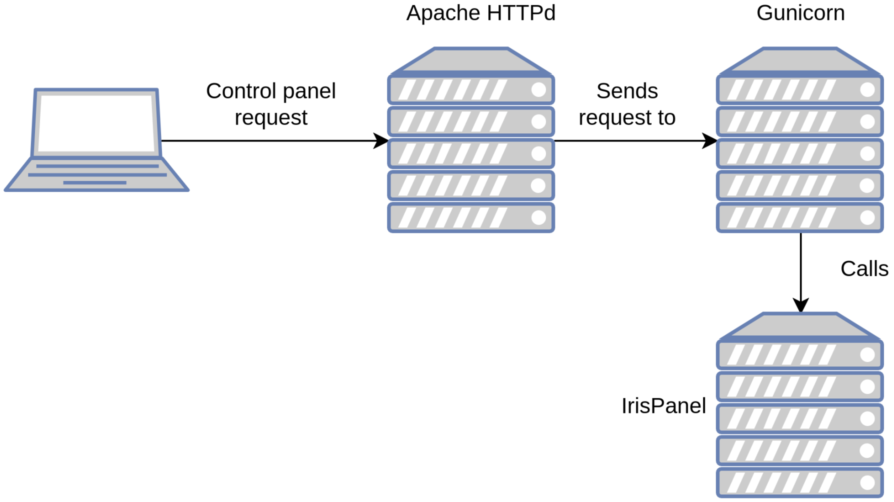

Figure 3: How a request for the control panel is processed

4 Planning your IrisPanel deployment

4.1 System Considerations

You should first ensure that your target environment will be able to accept an IrisPanel installation. The IrisPanel installation procedure requires a freshly installed copy of the target operating system, as installing onto an already in use system could lead to conflict and an unsupported system state. Therefore, it is recommended to re-install the server's operating system prior to IrisPanel installation. IrisPanel currently only supports installation on Debian 13. The installer will exit if you attempt to begin the installation program on any other distribution.

A minimum of 512 megabytes of RAM is recommended for the installation of IrisPanel, and nearly any processor you are likely to be using will probably suffice. These requirements are just a guide for IrisPanel itself, you must consider how much memory and CPU any ISV applications you may choose to install will use (such as the popular *WordPress* CMS package), and take into consideration the level of load that any websites you host with IrisPanel will be under. While IrisPanel itself may use a consistently low amount of system resources, websites with heavy PHP code that are under heavy load will naturally require more system resources than a lightweight static site under light load.

4.2 Deployment Security

To install IrisPanel, you must either be running as root or be able to elevate to root privileges on the target server (such as with *sudo)*. IrisPanel itself will install all of it's own components in a secure configuration, but does not do anything to secure the wider server's operating system and configuration. It is therefore recommended that you secure the server with basic security measures *prior* to the installation of IrisPanel, immediately after system installation. While they will not be covered by this document, general good practice would be to disable password authentication and use exclusively SSH-key based authentication, disable root login over SSH, set up Fail2Ban, and configure a firewall such as UFW or firewalld.

If you are in an environment with no internet access on the server (such as in an air-gapped environment), you will not be able to follow standard IrisPanel installation procedures, as the installer relies on being able to download the latest version of IrisPanel from the repository. This is something that may be addressed in a future version of IrisPanel.

4.3 Networking Considerations

For full functionality, your IrisPanel web server should be the machine exposed to the internet, not behind a reverse proxy. This is to enable the IrisPanel certificate management features to be able to automatically retrieve and configure TLS for your domains. Deployments behind reverse proxies are supported and possible, but it will require you to configure your reverse proxying HTTP server, which is out of scope for this manual.

5 Installation

Before beginning the installation please again ensure that you are running a clean copy of your Debian 13 operating system. Please also note that at present, IrisPanel does not have an uninstallation facility, the only supported uninstall method is to reinstall the machine.

The actual installation procedure for IrisPanel is very simple. First, use wget to download the installation program:

```
wget https://codeberg.org/confusionunknown/irispanel/raw/branch/main/install.sh
```

Now, make the install program executable:

```
chmod +x install.sh
```

And finally execute the installer as root:

```
sudo ./install.sh
```

With those commands, IrisPanel will configure the services, install IrisPanel itself, and set up the database.

The installer will ask you for your email address (this is only used to enable Lets Encrypt), the domain that the panel is running on (this is used to self-sign TLS certificates for the IrisPanel control panel itself and the FTP server), and once it is complete it will ask you to create an initial user for your IrisPanel system.

Once this is complete, you can log in by visiting the server's public address on port 5252 (remember, *only* HTTPS is supported for connections to the control panel), and begin to use the system.

Note: before logging in to the control panel, you should compare the fingerprint of the certificate presented to your web browser with the one stored on the system at `/opt/irispanel/certs/irispanel.crt`*, in order to ensure that the connection is secure.*

6 Administration and Upkeep of your IrisPanel system

6.1 Managing accounts

6.1.1 Adding accounts

As root/with sudo, execute the following command to add an account:

```
/usr/bin/add-irispanel-user
```

This program will create an IrisPanel account *and* create a corresponding Linux system account, meaning this command is all that is required to allow a new user to use the IrisPanel system

As of the writing of this document, this program requires interactive input - this will be fixed at a later date to allow it to be scripted and this document updated.

6.1.2 Removing accounts

As root/with sudo, execute the following command to remove an account:
`/usr/bin/remove-irispanel-user <username>`
This will remove the Linux user, remove the IrisPanel account from the database, delete any databases, domains and backups the user may have created and reload Apache HTTPd.

6.1.3 Changing an account password

As of the writing of this document, there is no supported procedure for changing account credentials.

6.2 Backups

IrisPanel system data is spread out across several key locations in the filesystem, all of which should be backed up regularly. The main things that you will want to back up are:

- The database server (the MariaDB server holds both databases created by users of the system, and IrisPanel configuration data)
- The account files
- Backups taken by users themselves of their accounts

The database server itself can be backed up with any supported method for backing up MariaDB. The database will also allow you to reconstruct the Linux users in the case that you have to rebuild the entire system.
Account files can be backed up by simply making backups of `/var/web`, which is the directory that contains IrisPanel account files (the Linux account home directories are located there, and so therefore is any file that any user could create).
Finally, you can back up `/opt/irispanel/backups` to keep backups of the backups that users have taken of their accounts.

7 Security Considerations

7.1 Networks

While IrisPanel does make an effort to prevent users from interfering with the system itself or other users on the same machine, this does not extend to the network. If you are running IrisPanel on an internal LAN, then anyone with an account that IrisPanel server will be able to reach out over the network from that computer. This is much more of a consideration if you host mail from the same IP address as the IrisPanel server - someone with an account on the IrisPanel server could use the PHP `mail()` function to send an email, which would pass through spam filters due to it originating from the correct IP address. (incidentally, the possibility of disabling the `mail()` function in IrisPanel's PHP environment is under consideration, if you have any feedback on this, please send it to the address mentioned in *Section 2: How to provide feedback on this document*)