IT'S A VIRTUAL SERVERWORLD

From his caravan in a field in North Wales, **David Rutland** takes you on a touring holiday through the world of virtual private servers.

The future is in the cloud, they say. But whose cloud? Google? Microsoft? AWS? Why not your own? There was once a time when people ran their own servers from home – under the stairs, in the back bedroom or from a lovingly kitted out homelab in the stripped-down carcass of the garage, meaning the vintage BMW restoration project is left to moulder on the drive.

A server is a simple beast at heart. It's a computer (usually running Linux), with an internet connection, which receives requests for resources such as pages and serves them back to the machine that made the original request. Server. It's in the name, innit.

But servers serve far more than web pages these days. There are progressive web apps, content management systems, databases and a whole host (geddit?) of other toys you may want to play with.

Some of these demand serious investment in terms of time, hardware, power consumption, and if you or your partner are light sleepers, noise becomes an issue, too. In fact a properly specced home server with adequate cooling often sounds like a jet taking off – so it's time to take your server virtually off the premises.

What is a VPS?

If you're determined to run your server away from home in order to avoid unnecessary concerns about power bills, antisocial noise, and so on, there are a few options available:

1 Rent out a dedicated private server in someone else's data centre. This option gives you an actual physical machine, equipped to your exacting specifications, and on which you can run whatever software you want. It's accessible only by you (making it private). Opting for a dedicated private server with an eight core Xeon and 64GB of RAM may get you the best possible performance, but it's far beyond the needs of most home users, and the financial cost will run to thousands of pounds per year.

2 Virtual Private Server. As with the dedicated server option, it's yours exclusively for as long as you choose to rent it, but the key thing here is that it's virtual. Your virtual machine will be one of several being run on one physical machine. You will be given a dedicated IP address through which you can log in and install your own software, and many providers offer some degree of pre-configuration through pre-built images. For example, you can opt for a machine which will have Linux Apache, PHP and MariaDB all set up and ready to go.

3 The cheapest option is to buy space on a non-VPS web host which is already running Linux. As an example we bought a standard, low-cost hosting package from **Godaddy.com** at a cost of £12 for a year, and while it was technically possible to deploy and use Nextcloud through the web interface, it didn't run well at all. Even the basic Nextcloud text editor (used to write this article) failed to function, save edits or reliably load documents. It also generated repeated warnings that "Your hosting account is reaching or exceeding its resource limits." Basically a standard webhost isn't configured in the same way as a VPS.

What's a VPS for?

So glad you asked! A VPS can be whatever you want it to be, and you can use it for whatever you choose. Generally you would use it for hosting software which has a front-end which can be accessed through a web browser.

The internet isn't what it was a decade ago, and thanks to the wonders of browser development, it's relatively simple to run beautiful applications which replicate, replace or augment services you would normally have to pay for – either with cold hard cash or indirectly with your personal data which is harvested, stored, analysed and sold every time you go online without protection.

We've been a fan of private servers (both dedicated and virtual) for some time, and deploy our own services to help us keep control of our own data, while enjoying all of the amenity the modern web has to offer. This is some of the self-hosted software we use which should give you some idea of what its all about:

>> **Nextcloud** This should probably be your starting point, as it has basic versions of almost everything you could ever need. At its heart it's a drop-in replacement for Google Drive and Dropbox. Where it excels is in its apps, which allow you to further expand the software to cover multiple use cases. There are full-fat office suites,

While it is absolutely within the LFX discretionary budget to spin up a $2,400 per month monster VPS, we decided that it would be egregious and excessive for the purposes of this feature.

music and video players, sync clients, mapping and location software, recipe managers, news aggregators… the list goes on and on. But while Nextcloud seemingly does everything, it isn't the best at any one thing, and as time goes on you'll be swapping Nextcloud functions for other services which are dedicated to a single task.

>> **Jellyfin** A FOSS project to manage and play your multimedia library. Jellyfin handles videos, music, photographs, ebooks, and audiobooks, meaning you can listen, watch or read at any time and on any device.

>> **FreshRSS** RSS isn't dead and it is still this writer's favourite way of reading digital news and magazines. *FreshRSS* will pull in RSS feeds from sources you select, and display them for you in a browser or through one of the many available mobile clients. Even better, with a little jiggery-pokery, *FreshRSS* will pull full articles rather than stubs – meaning that you never need to visit another website again.

QUICK TIP

While it is possible to run web apps in a pseudo–VPS using standard web hosting, it isn't worth the trouble unless the resources usage is exceptionally light.

Your hosting account is reaching or exceeding its resource limits.

You can sign in to your account at any time to view your usage or move to a bigger plan.

Review Your Account

To make sure you've always got the speed and storage you need, we recommend switching to a Business Hosting plan. It comes with dedicated resources, and you can manage it with the cPanel you already know. Get more details.

Here's an article to help you learn more about resource limits and why you might be using more than you expect. You can call also our hosting support team at 020

This is what you can expect if you go for £12 per year non–VPS hosting.

>> WHAT ABOUT AWS?

The one-line explanation of a Virtual Private Server is pretty simple. It's a virtual machine running on a physical machine, alongside other people's virtual machines. You are responsible for the virtual machine, its uptime, its security updates and so on. Your VPS provider isn't going to be interested if your VPS gets hacked, it won't provide security or even updates for you. Case in point, my VPS required downloading around 900MB in updates immediately after initialising the VPS on DigitalOcean. Your resources are strictly limited to what you've paid for and if you max out the RAM on your VPS, it will crash.

Amazon Web Services is slightly different. It is virtual cloud computing – a phrase which means a lot more than the three words would suggest. Virtual cloud computing means that you won't have root access to the system, but security and updates are handled on your behalf. It's also infinitely scaleable, and if your computing needs outgrow what you've paid for, Amazon will still allow you to access more and more resources – meaning that your monthly bill can quickly grow out of hand. Thousands of pounds out of hand in some cases, so be careful…

» Photoprism Do you like being able to back up, sync and view your photos automagically, but don't trust Google and Apple not to check out your personals? You need a self-hosted photo manager hosted on a VPS. *Photoprism* is our choice. Other people may have strong feelings on other software.

» The Fediverse The social network you've probably never heard of. It has millions of users and thousands of private servers which federate together to create a distributed social platform which is surprisingly usable. If you want to be part of it without signing up to yet another monolith administered and moderated by other people, you'll need to set up an instance of your own using Pleroma, Mastodon or Soapbox. A VPS is the perfect place to do it!

There are many more potential uses for a VPS, but space necessitates brevity. We'll be coming back to the software I've mentioned (and more!) in coming issues.

Choosing the right VPS

With great power comes great responsibility. So said Spider-Man, or possibly Voltaire. When choosing a VPS, the adage shifts slightly to "With great power comes great cost", so it's important to know roughly what resources you need your remote virtual machine to have, and an idea of what you want to spend on it.

To have an idea of the specs you require, make a list of the services you'd like to try out then check out the project GitHub pages to see the minimum requirements. You'll also want to read about other users' experiences to see how the software performs in the real world. While Nextcloud will run happily (albeit slowly) on a VPS with an ARM processor and 512K of RAM, *Photoprism* demands a minimum of 4GB of RAM in order to index your epic 20GB collection of holiday photos amassed on the beaches of Europe over the last three decades.

It's a good idea to start small with a low-cost VPS and then add in more memory, storage space and processing power as the need arises.

Being the cheapskates we are, we chose a $5 per month DigitalOcean droplet with an allotted 1GB RAM, a 25GB share of an SSD, and a 1TB monthly data transfer limit. Naturally, this is scaleable and we can upgrade at any time. At the top end of the scale, DigitalOcean offers a VPS with a staggering 256GB RAM, 32 virtual CPUs, 7TB storage and 10TB transfer limit. At $2,480 per month, this is probably not the VPS you are looking for.

Get up and running

Virtual Private Servers provided by DigitalOcean are called droplets. It's a pun: your computing requirements are a mere droplet in their digital ocean. It hints at their limitless resources and your insignificance as both a

Every machine needs an awesome hostname – even virtual ones.

The first thing you'll want to do is set up a user who isn't root – trashing the system by accident is a very real risk. Then again, it's only a *virtual* system.

human being (*I feel insignificant enough, thanks–Ed*) and as a user of their services However, as this section can be broadly applied to providers other than DigitalOcean, I'll be referring to the droplet simply as a VPS.

After account creation, which involves verifying your identity by way of adding a $5 balance to your account, you'll be presented with an array of options, including deploying a web app, hosting a website or static site, deploying a container-based app or deploying a database. The option you want, of course, is 'Deploy a Virtual Machine.' This allows you to set up a VPS, from inside which you can do all of the other stuff anyway.

Clicking through to the next screen enables you to configure the specs of your new VPS, and you can choose from a selection of Ubuntu, FreeBSD, Fedora, Debian, CentOS and Rocky Linux. Each of these distros has a range of recent versions available.

'bunt it up

We chose Ubuntu 20.04 LTS because it's what we use at home and because it's the most popular Linux distro, while there is excellent support available from a range of sources. We stayed with the basic single CPU configuration and selected a datacentre in London – a closer datacentre means less latency. The next choice you'll need to make is whether to use SSH keys or simply a username and password. SSH keys are considerably more secure, but the choice, as always, is yours.

Finally, you will be asked to choose a hostname. By default this will be long, complicated and nonsensical. We chose something long, complicated and sensible: **the-awesome-lxf-virtual-private-server**.

With that, it's time to click Create Droplet or the equivalent button for your VPS provider, and go and make a cup of tea.

Hot brew in hand, you should come back to your machine after ten minutes or so. You'll see an IP address through which you'll be able to SSH into your shiny new VPS. Note that you'll be SSHing as root, eg: **root@159.65.216.202**, so the first thing you'll want to do is to create a new, non-root user with its own home directory skeleton using the *useradd* command.

```
useradd -m new_username
```

followed by a new password for your new user using the *passwd* command:

```
passwd new_username
```

Grant your new user *sudo* powers so you can actually get things done by typing:

```
usermod -a -G sudo new_username
```

Finalize and create

How many Droplets?
Deploy multiple Droplets with the same configuration.

Choose a hostname
Give your Droplets an identifying name you will remember them by. Your Droplet name can only contain alphanumeric characters, dashes, and periods.

1 Droplet   +   the-awesome-lfx-virtual-private-server

Add tags
Use tags to organize and relate resources. Tags may contain letters, numbers, colons, dashes, and underscores.

Type tags here

After this you should log out as root and log back in as your new user. Now you can start using your VPS just like your regular machine at home, update and upgrade the already installed packages using `sudo apt update` and `sudo apt upgrade` and start adding packages which will make your VPS journey possible.

As a bare minimum, you should have server software such as Apache or Nginx (always choose Apache because then you can say "It's a 'patchy' server"). You'll also need PHP and a database such as MariaDB.

You can install Apache by typing `sudo apt install apache2`, PHP by typing `sudo apt install php`, and MariaDB by typing `sudo apt install mariadb-server` then `sudo mysqlsecureinstallation` and following the prompts.

A domain of your own

Finally, you probably want a domain name through which you can access your services from out in the world. As we already own **lxf.by**, we chose to create a subdomain of that for our first service: **vps1.lxf.by**. If you don't already have a domain name, they can be picked up for a small cost from a provider such as Namecheap, or for free from Freenom.

You need to make sure that the nameservers at your registrar are pointed towards the IP address given to you by the VPS provider. Fortunately, this is as simple as changing the standard registrar nameserver to those of, in our case, DigitalOcean. From within your registrar's website, find the option for nameserver and change it from standard to custom. In the resulting field, you will want to add something like **ns1.digitalocean.com**, **ns2. digitalocean.com** and **ns3.digitalocean.com**. Your VPS provider will handle the rest.

There is no point in us going into the exact details here as the location of the nameserver settings will vary from registrar to registrar. Most will have a help document available.

After the DNS changes have propagated (you can check this using **dnschecker.org**), you should be ready to go. Typing in the address of your new domain into your browser should result in you seeing the Apache2 Ubuntu Default Page.

Take a sip of your cooling tea and wipe your brow. The hardest part is over. You have a working virtual machine which you can access through an IP address.

Deploying VPS services

There's no point having a VPS if all you're going to do with it is host a static webpage – especially if that static webpage is the most boring and ubiquitous in existence.
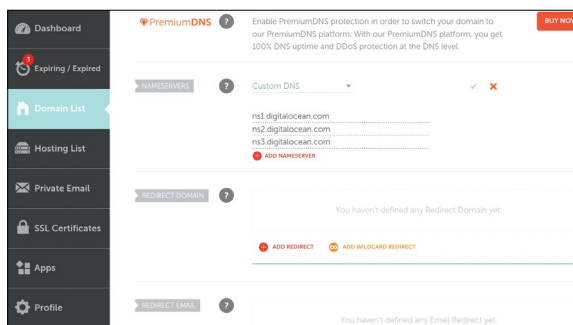
This is where you set custom nameservers using Namecheap. The exact location will vary according to your own registrar.
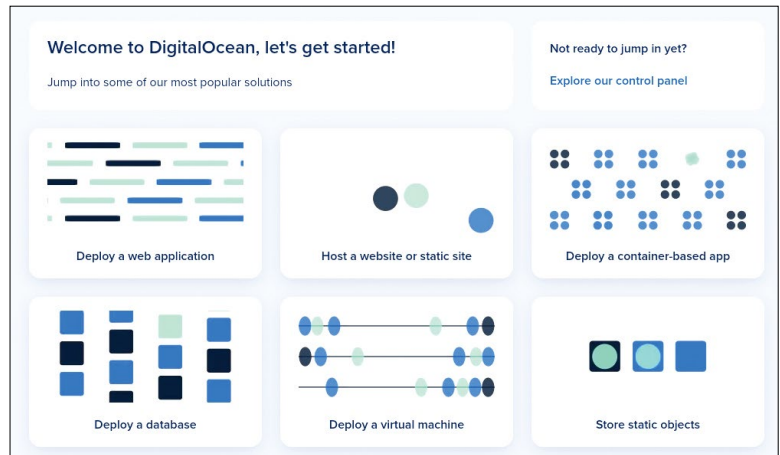
It's tempting to deploy a web app without setting up your VPS first, but this will only lead to tears. Select the middle option on the bottom row instead.

A VPS is all about useful services. Search Google (other search engines are available) for 'Awesome self hosted' and you should end up on a curated GitHub page with a list of highly rated self-hosted FOSS software. Have a browse and see if there's anything you fancy. You'll come across some which require or use Docker and *docker-compose*. These are outside the scope of this article, but you'll find plenty of Docker tutorials in previous issues, the most recent in **LXF272**.

As a super-quick demonstration piece, I chose Sharlii (**https://github.com/shaarli/Shaarli**) which is a 'personal, minimalist, super-fast, database free, bookmarking service.'

To install it, download the package into your VPS home directory using:

`wget <https://github.com/shaarli/Shaarli/releases/download/v0.11.1/shaarli-v0.11.1-full.zip>`

Unzip by typing:

`unzip shaarli-v0.11.1-full.zip`

(Sidenote: you'll need to actually install unzip first with `sudo apt install unzip`.)

`cd` into Shaarli, them move the contents of the directory to where it can be seen by Apache.

`cd Shaarli`

`sudo mv \* /var/www/html/`

remove the original Apache index page:

`sudo rm /var/www/html/index.html`

lastly give ownership of the html directory to apache:

`sudo chown -R www–data:www-data /var/www/html`

That's it. You've installed your first web service on your very own VPS. Have a hunt round and find some more – it's addictive, and before you know it, you'll be wanting to upgrade your server. **LXF**
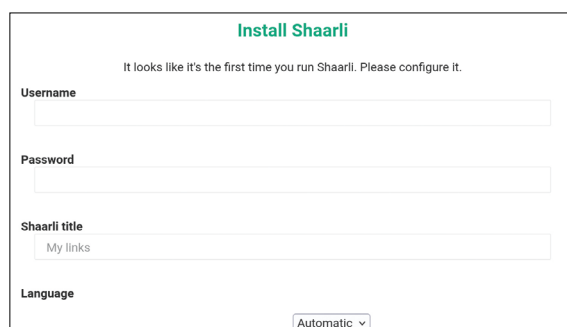
Congratulations! You've deployed your first web app on your very own VPS. Awesome Self Hosted has plenty of other great suggestions.