Installing and publishing Tiki Wiki on Debian 13 (2026 Edition)

Bernard Sfez - 2026-04-20 09:04

This guide explains how to set up a Debian 13 server with the Virtualmin hosting control panel and deploy a Tiki Wiki CMS solution using exclusively open-source software.

Built with security and digital sovereignty in mind, you will learn how to configure essential server components (ports, MariaDB, PHP versions), install Webmin and Virtualmin, secure your server with SSL certificates, and publish your Tiki Wiki site efficiently.
This tutorial is designed for intermediate system administrators and full-stack developers seeking a reliable, scalable, and sovereignty-friendly approach to deploying modern web applications.

In this tutorial, you will learn how to set up and configure:

- AWS Lightsail instance (also applicable to EC2 and most cloud providers)
- Linux operating system: Debian 13 with Apache2
- MariaDB (InnoDB storage engine)
- Webmin and Virtualmin web hosting control panel
- PHP (version management and configuration)
- Git and the Tiki source code from GitLab
- Tiki Wiki CMS and website builder

⚠ Sovereignty & Hosting Considerations

Given how rapidly legislation evolves, it is important to restate our position: we prioritize sovereignty, control, and transparency over private data, notably through Open Source software and, whenever possible, by keeping data in-house or with a trusted provider.
We used an Amazon Lightsail free-tier instance to validate and test this guide. This choice is purely practical for demonstration purposes. The same setup can be reproduced with minimal adjustments on European cloud providers (OVHcloud, Scaleway, Infomaniak) or on your own infrastructure.

If you rely on an external hosting provider (AWS, Google Cloud, Azure, OVH, etc.), carefully review your contract—especially terms of service, jurisdiction clauses, and applicable law. The physical location of servers does not always guarantee exclusive legal protection under that country’s regulations.

Hosting in Europe can improve data residency, latency, and operational control, but may not fully eliminate exposure to extraterritorial regulations. For more context, see our analysis on digital dependence on U.S. technology.

Before starting, a basic familiarity with the Linux shell and server administration is recommended. That said, you can absolutely learn along the way—just make sure to work in a test environment without sensitive data.

Most cloud providers offer flexible instances and free tiers, allowing you to experiment, rebuild, and refine your setup safely.
If you prefer to save time or ensure a production-ready deployment, you can contact us directly or explore the Tiki Consultants list. Working with a specialist helps ensure a secure, optimized, and fully sovereign-friendly setup.

Server Preparation

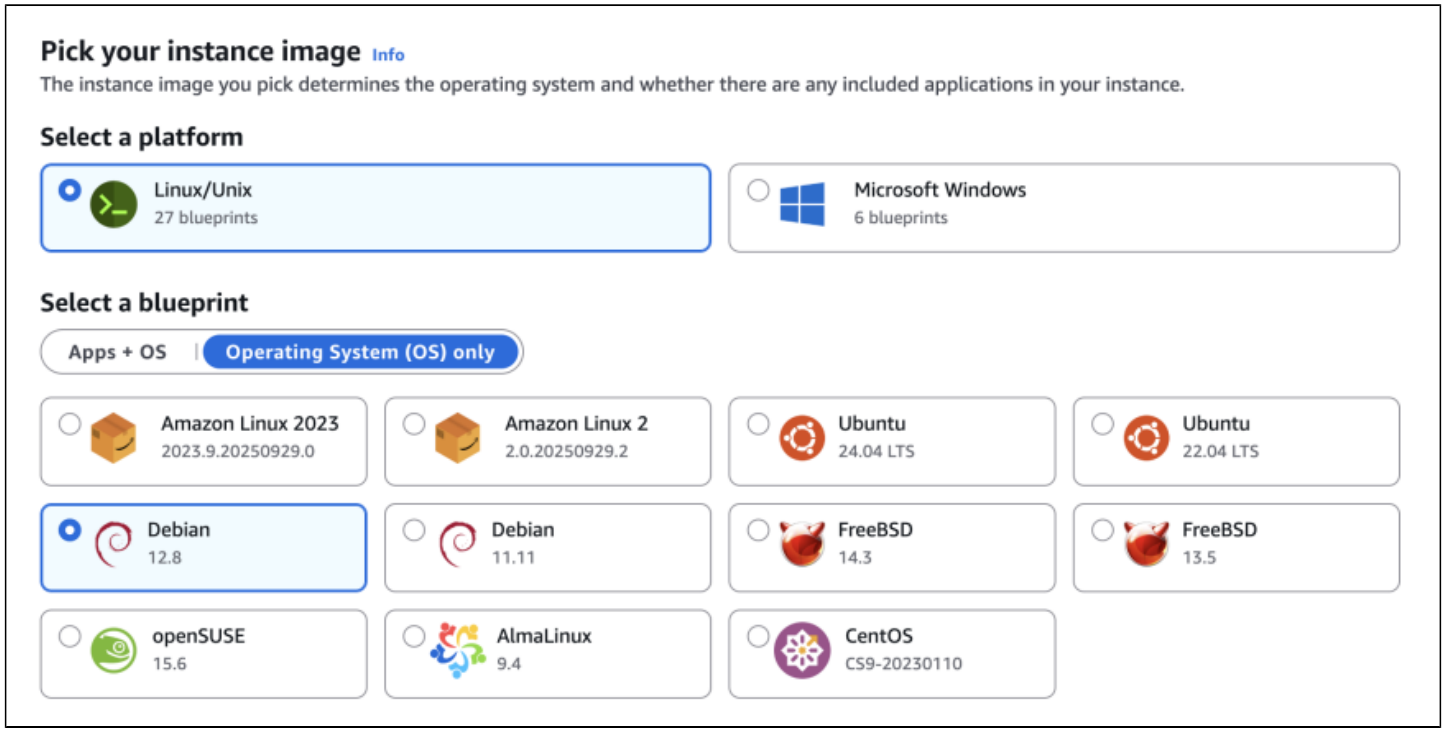
Install an Amazon Lightsail Instance

Region Selection

Log into your AWS Console (or create an account) and search for "Lightsail" in the search bar.

Select your instance location by choosing a region closest to your target users for optimal latency and performance. Then choose your instance image by selecting a platform.
Note: Lightsail instances are not available in all AWS regions.

For this tutorial, select the "Linux/Unix" blueprint, choose "OS Only", and then select Debian 13.x. Important: Virtualmin requires a clean, minimal system. Avoid pre-configured stacks or additional packages beyond a basic OS installation.



SSH Key for Instance Access

In the SSH key section, you can use the default SSH key or specify an existing one.

If this is your first instance and you prefer simplicity, select the default key generated by AWS and download it to your `~/.ssh` directory for future access.

If you already have an SSH key, you can upload and associate it with the instance. You may also create a new key pair by following AWS documentation.

Select Your Plan

Choose your instance plan carefully. While entry-level plans may work for older Tiki versions, Tiki 26+ requires at least 2 GB of RAM, and newer or development versions (such as Tiki 29/30) may require 4 GB RAM or more.

Note: Lightsail does not support in-place upgrades. To scale up, you must create a new instance from a snapshot.

To avoid unnecessary migrations, select a plan aligned with your expected usage. If needed, consider scalable and reproducible deployment approaches such as those provided by Open Source Solutions.

If you're unsure, you can contact our team for guidance.

Save Your AWS Instance

Give your instance a clear and descriptive name to stay organized—especially if you plan to manage multiple environments (development, staging, production). Tags can also help with classification and cost tracking.

Finally, click the orange "Create Instance" button. Within a minute, your server will be running. You can access it directly using the integrated SSH console via the terminal icon.

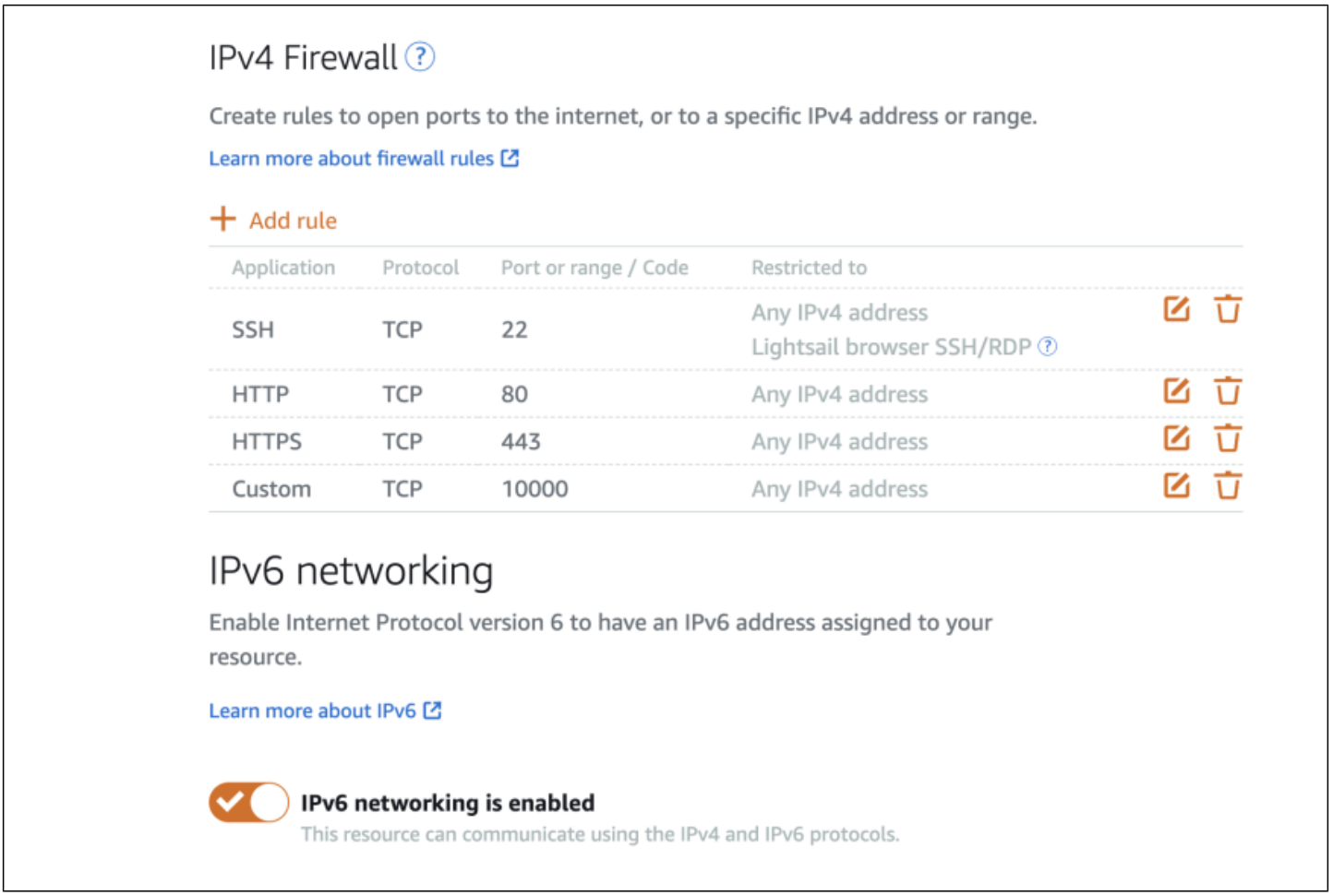
Lightsail Networking and Security Group Setup

Configuring access to your site and the Virtualmin control panel is essential for server security. Begin by navigating to your Lightsail instance dashboard and selecting Networking to manage access rules.

By default, ports 22 (SSH) and 80 (HTTP) should already be open.

1. Configure Firewall Rules
Add the required ports by clicking "Add Rule":

- 443 → HTTPS (SSL-secured web traffic)
- 10000 → Virtualmin/Webmin control panel



2. Apply Access Restrictions (Recommended)
For improved security:

- Keep ports 80 and 443 open to all traffic (public access required for websites)
- Restrict ports 22 (SSH) and 10000 (Virtualmin) to your IP address only
- During setup, you can use the Lightsail browser-based SSH console if needed
- After setup, consider changing the default SSH port to reduce automated attacks

> At Open Source Solutions, we apply a strict server port management policy to reduce attack surfaces and enhance infrastructure security.

3. Set Up a Static IP
By default, your public IPv4 address changes when the instance is restarted.

To avoid this:

- Allocate a Static IP from the IPv4 networking section
- Attach it to your instance

This ensures your server always keeps the same public IP address.

You can also point your domain name to this static IP (DNS configuration), although detailed DNS setup is outside the scope of this guide.

4. Apply and Test

- Reboot your instance to ensure all changes are applied

- Verify:
  - Static IP is correctly attached
  - Required ports are open and accessible

This setup provides a secure baseline while maintaining full functionality for web and admin access.

⚠ Enabling root login

With your AWS instance default comes an "admin" account member of the sudoers group and therefor able to switch to the "root" super user (su). Enable root login only if your server or hosting configuration requires it—and we strongly recommend against doing so, except possibly on a temporary basis.

Sudo or Set and use root login☒✓

Sudo or Set and use root login

Depending of your account settings, you may be able to use the root user or use the sudo command. If you can use the root user or sudo that's ok, you can skip this.
If you want to enable root login anyway... this is the way.

Logged as the default "admin" user. (I use nano but it is the same with VI, VIM or any editor)

Switch for the Super user (root)

```
sudo su
```

Modifier le fichier de configuration SSH

```
nano /etc/ssh/sshd_config
# Change
PermitRootLogin no to PermitRootLogin yes
# PasswordAuthentication no to
PasswordAuthentication yes
```

Of course save your changes.
You will also need to remove a small script that forbid root to use his SSH key

Edit the authorized_keys file

```
nano ~/.ssh/authorized_keys
```

The script begins with "no-port-fowarding..." you need to remove this script up to the ssh key itself that starts with "ssh-rsa...".
That way the ssh key will be accessible to the ssh login.

Restart SSHD to validate changes

```
systemctl restart sshd
```

Set the root password

```
passwd root
```

Reboot your instance

```
sudo reboot
```

You will be disconnected, reconnect and check everything is working fine when you login as root.

Login as root

```
ssh root@xx.xx.xx.xx (your instance IP)
```

Note: It is a good practice, and I highly recommend it, to disable root access on production servers once everything is in place. This can be done by setting the "PermitRootLogin" and "PasswordAuthentication" parameters to "no" in the "/etc/ssh/sshd_config" file.

Connect to Your Server Using SSH

Connect Using the AWS Browser-Based SSH/RDP

Once your instance is running, you can begin managing it. A simple way to access your server is through Lightsail's browser-based SSH/RDP, which provides direct, secure SSH access via an AWS terminal.

For most management tasks, this browser-based method is sufficient. However, you should also be able to log in using an external terminal application, and later, we'll use the Virtualmin integrated terminal for more advanced tasks.

Connect Using SSH from an External Terminal

Alternatively, you can connect to your server using your preferred terminal application (e.g., PuTTY, macOS Terminal, or others) for shell access. This approach requires the public IP of your instance and your locally stored SSH key. The default administrative user set by AWS is admin.

The command to connect (assuming your SSH key is stored as LightsailDefaultKeyPair-us-east-1.pem) would look like this:

SSH using the stored default key

```
ssh -i ~/.ssh/LightsailDefaultKeyPair-us-east-1.pem admin@xx.xx.xx.xx
```

Replace xx.xx.xx.xx with your instance's public IP address. This method offers greater flexibility, particularly for advanced configuration and software installations.
If you didn't set the SSH key during instance creation, configuring the SSH key pair may be a bit challenging if you're not familiar with IT administration. Here’s how to proceed:

- Replace LightsailDefaultKeyPair-us-east-1.pem with your key filename.
- Download the SSH key to your computer.
- Move the key to your ~/.ssh directory.
- Set the correct file permissions:

Set per file permissions

```
sudo chmod 600 ~/.ssh/LightsailDefaultKeyPair-us-east-1.pem
```

After this, you should be able to connect with the following command:

SSH using the stored default key

```
ssh -i ~/.ssh/LightsailDefaultKeyPair-us-east-1.pem admin@xx.xx.xx.xx
```

For further documentation, visit the Amazon Lightsail SSH Using Terminal guide.

Checking the Hostname

To ensure smooth operation of Virtualmin/Webmin, it's essential to properly configure your server's hostname and Fully Qualified Domain Name (FQDN). By default, your server may auto-generate the hostname, but it’s important to verify and correct it if necessary.

Step 1: Check the Current Hostname
Once logged into your server via SSH, use the following command to check the current hostname:

Check hostname

```
hostnamectl
```

In the output, the "Static hostname" will typically show the instance's local IP. Confirm this by running the following command to output only the hostname:

Check hostname

```
hostname
```

Step 2: Set the Hostname to Your FQDN
You need to update the hostname to match your FQDN (e.g., yourdomain.com) before proceeding with the next step (Installing Virtualmin). To do this, edit the hostname configuration file:

Edit the hostname

```
sudo nano /etc/hostname
```

Change the existing entry to your desired FQDN (e.g., subdomain.domain.com).
Step 3: Edit the Hosts File
Next, verify and update the /etc/hosts file to include your FQDN. It should look like this:
-+ 127.0.1.1 subdomain.domain.com subdomain+- and you need to set it.

To check the current configuration, run:

Check hostname

```
cat /etc/hosts
```

To edit the hosts file and set the correct FQDN, use:

Edit host

```
sudo nano /etc/hosts
```

Add or update the line as shown above (replace subdomain.domain.com and subdomain with your actual domain and subdomain).
Step 4: Reboot the Server
After making these changes, reboot your server to ensure the changes take effect:

Reboot server

```
sudo reboot
```

Step 5: Verify Changes
Once the server reboots, check that everything is set up correctly by running:

Check hostname

```
hostname
```

If your FQDN is displayed correctly, you are ready to proceed with installing Virtualmin. If not, double-check the configuration files or refer to the troubleshooting options below.

Fix host file and make it permanent☒✓
You need also to check and add your host to the hosts file if your providing company didn't set it right.
But if you do it directly (editing /etc/hosts) the changes will be reverted on the next reboot.
To make it permanent, you need to modify the template used to recreate on each reboot the hosts file.

Some instance may use a Cloud configuration file other may use a Debian host file.
You will find useful information in the top comments of -+/etc/hosts+-

If the file start with: "Your system has configured 'manage_etc_hosts' as True.", it means your server is using cloud-init and you need to change the value of "preserve_hostname" to true at /etc/cloud/cloud.cfg

Edit hosts.debian.tmpl

```
sudo nano /etc/cloud/cloud.cfg
preserve_hostname: true
```

Reboot the server and check changes are permanent.

Create a swap file (swapfile)

Since the introduction of the new Built system in Tiki (starting with Tiki 27), installing from Git has become more resource-intensive than installing from the official release package (standard installation).

With Tiki 29, you generally need more than 2 GB of RAM to complete the installation properly because NPM packages are required. If your server has limited physical memory, we recommend creating and enabling a swap file (swapfile) to avoid out-of-memory issues (processes being killed, build/installation failures, etc.).

It is best to set up swap as early as possible—ideally before starting the installation—so the entire deployment (and the server more broadly) can benefit from it.

See the full procedure (Debian/Ubuntu) that we provide.

Install Webmin and Virtualmin

Virtualmin will install everything you need and it is critical the installation of Virtualmin comes first.
Else you will be warned previous tools or libraries have been installed and sometimes it means you will have to clean them manually before proceding.

We will download Virtualmin automated install script and run it. It is basically a shell script that will handle rest of the installation once executed.

Download Virtualmin install script

```
sudo wget https://download.virtualmin.com/virtualmin-install.sh
```

sudo You can do a full install but also you can do a minimal install which will spare the instance resources. For exemple where I don't need a mail server I do minimal install.

Full install of Virtualmin

```
sudo sh install.sh
```

Minimal install of Virtualmin

```
sudo sh virtualmin-install.sh --type mini
```

```
admin@fwtestserver:~$ sudo sh install.sh --minimal
[INFO] Log will be written to: /home/admin/virtualmin-install.log

  Welcome to the Virtualmin GPL installer, version 7.4.0

  This script must be run on a freshly installed supported OS. It does not
  perform updates or upgrades (use your system package manager) or license
  changes (use the "virtualmin change-license" command).

  The systems currently supported by the install script are:

    Red Hat Enterprise Linux and derivatives
      - RHEL 8 and 9 on x86_64
      - Alma and Rocky 8 and 9 on x86_64
      - CentOS 7 on x86_64

    Debian Linux and derivatives
      - Debian 10, 11 and 12 on i386 and amd64
      - Ubuntu 20.04 LTS, 22.04 LTS and 24.04 LTS on i386 and amd64

  If your OS/version/arch is not listed, installation will fail. More
  details about the systems supported by the script can be found here:

    https://www.virtualmin.com/os-support

  The selected package bundle is LAMP and the size of install is
  minimal. It will require up to 1 GB of disk space.

  Exit and re-run this script with --help flag to see available options.
```

You should see now the different components being installed one by one.
This is taking some times... Just wait for the process to complete.

It should end with:
"SUCCESS to configure at https://xxx-xxx-xxx-xxx:10000 (or https://yourFQDN:10000)."
You should be ready to login to your Virtualmin panel using your root access ===if your firewall is set correctly===.
''It happens your hosting services open only certain ports by default like 80,443,22. For virtualmin you need to also open port 10000.''

Setup your root access to Virtualmin

To be able to use Virtualmin as root without enabling root login I run a new webmin passwd command to change the password and set Webmin password authentication.

Set your Webmin root user

```
sudo webmin passwd --user root
```

Use it to login to your new Virtualmin.
https://xxx-xxx-xxx-xxx:10000 (or https://yourFQDN:10000

Virtualmin setup and post-installation optional features.

Once you've logged into your Virtualmin web interface, you may need to trust the SSL certificate (depending on your browser and system configuration).

Follow the Virtualmin Post-Installation Wizard. This wizard is straightforward and provides step-by-step guidance to complete the setup. While the questions may vary based on your system and the version of Virtualmin you're using, in most cases, it is safe to proceed with the default settings suggested by the wizard.
Database servers

- Run MariaDB database server?
- I use MariaDB for my Tiki Wiki installs so, yes.
- ''You will have to validate or change the MariaDB root password, enter it and save it somewhere if needed later.''
- Run PostgreSQL database server?__
- As stated above it will be a no.

You will be asked to set / change the MariaDB password.

DNS configuration
Primary nameserver should show the hostname set previously.

System email address
Set your admin email.

You can stop now and have enough to launch your first Virtual Server (site) or continue with additional wizard settings.

Password storage
We don't want to store in clear the passwords used on the server so I select "Only store hashed passwords".

MariaDB configuration size
This depend of your usage of the Tiki Wiki instance but selecting the "suggested" option by the virtualmin wizard is fine.

SSL key directory
Unless you know what you do keep the default. (Per-domain)

At this stage, the process should be complete. However, you may encounter a prompt asking if you'd like to create a default virtual server. If this happens, I recommend selecting "No" to manually configure the settings for your domain.

After making any necessary changes, I always recheck the configuration to ensure everything is in order.

Once satisfied, proceed to a last reboot of the server, and check all our settings didn't change so you can create your first Virtual Server—essentially a website.

Create your first Virtual Server (your website)

- Set your domain name.
- Give it a description.
- Set the Administration password (even if you use an SSH key keep a second access is wise.
- Add an SSH public key (copy your Public key)
- Set the Administration username (or keep default)

In the Advanced panel I usually change only the Default database name.
In the Enabled features panel if you won't use Email and the Internet AWStats I suggest to disable it (Mail for domain and Enable AWStats reporting)

Don't change the rest and Create your server.
Once it has been created Return to the Virtual Server screen and in the Quotas and limits panel you should check the Server Quota. During Tiki Wiki installation from git, the setup process may require more than 2Gb so you can set to unlimited during the setup and limit the quota once everything is running.

Install Let's Encrypt SSL Certificate

Note: By default, Virtualmin attempts to set up SSL for all domains in the "Domains associated with this server" list. If some of these domains aren’t set up correctly or are inaccessible, the certificate request will fail. To avoid this, I recommend manually specifying only the domains you’re using in the "Domain names listed here" field to install the certificates only for the relevant domains.

To install the Let's Encrypt SSL certificate, follow these steps:

- Go to the Virtual Server List, select your Virtual Server, and in the left-hand menu, navigate to Manage Virtual Server > Setup SSL Certificate. Then, select the Let's SSL Providers.
- In the Request certificate for field, select and enter the domain names for which you need the SSL certificate, using the "Domain names listed here" option.
- Ensure that the "Automatically renew certificate?" option is set to Yes, then click Request Certificate.

Before starting, make sure your domain is properly configured at your domain registrar.

Checking Server Status and Managing Running Services

In your Virtualmin dashboard, you’ll find a panel displaying the status of your servers and applications. Here, you can verify which services are running and take necessary actions:

- Check PHP Version: Confirm the PHP version currently in use to ensure compatibility with your applications.

- Mail and Mailbox Applications:
  ** Keep Postfix running as it is essential for sending emails, such as system notifications or application-generated messages.
  ** Disable Dovecot if your server does not need to process or receive incoming emails.

However, stopping Dovecot via the dashboard only halts the service temporarily. To disable it permanently:
In the left-hand menu, navigate to Webmin > System > Bootup and Shutdown.
Locate any services labeled "Dovecot" in the list.
Select the service, and at the bottom of the page, click Disable On Boot.

Additional Recommendations

- Enable Fail2Ban: Protect your server from brute-force attacks by enabling this intrusion prevention tool.
- Configure a Firewall: Enhance your server’s security by enabling and configuring a firewall to restrict unauthorized access.

By carefully managing these services and implementing security measures, you ensure your server runs efficiently and remains protected.

Installing Multiple Versions of PHP

''Note: This information was accurate at the time of writing this tutorial but it is easy to change depending of your environment.''

Debian 12 comes with PHP 8.2 by default, but this version may not meet your specific requirements. If you need PHP 8.1 or other versions, you can install and configure them on your server.

Adding PHP8.1 from the sury repository
You can configure which one is the default PHP version used on your new Virtual Servers. You can change that default in System Settings -> Server Templates -> Default -> PHP Options. To set your host or domain PHP version go to Virtualmin -> Server configuration -> PHP version and on that screen you can set the PHP version for your domain.

Navigate to System Settings > Server Templates > Default > PHP Options.

Set PHP Version for a Specific Domain:
Go to Virtualmin, select domain > Web Configuration > PHP Options.
On this screen, you can select the PHP version for a particular domain.
Installing Libraries for Additional PHP Versions
If you need to support multiple PHP versions, install the required libraries as follows:

Additional libraries for Tiki Wiki

Don't forget that anyway you have to install a few additional libraries and they should be accessible to complete the setup.
That's why these have been added above but you can also add them for your other PHP versions. (change the version number)

Install PHP8.4

```
sudo apt-get install php8.4-gd php8.4-intl php8.4-curl php8.4-zip php8.4-bcmath
```

By following this process, you can ensure your server has the required PHP versions and libraries for various applications.

Secure the server (basic)

Enhancing server security requires skills, knowledge, and expertise beyond the scope of this article. However, it is crucial to implement basic measures to protect your server and data. Here, we will review the minimum steps you should take. For a comprehensive security review of your Tiki and server, please contact me.

Secure your MySQL database (version before MariaDB 11.8.6)
Done, let's check MariaDB is running:

Test mariaDB status

```
sudo systemctl status mariadb
```

''Some errors may appear since we did not use the root password for this command. However, MariaDB should indicate that it is "Active," confirming that the service is running successfully.''

Installing Tiki from the Official Tiki Repository (Anonymous)

Navigate to Your HTML Directory
Begin by navigating to the html directory. If you are unsure of its location, check the Virtual Server Summary in Virtualmin.

Clone the Tiki Repository

Visit Tiki's website for detailed instructions, or refer to the complete installation guide. For a quick setup, use the following command to clone the desired Tiki branch (in this example, 29.x) without the repository history:

Download Tiki from the git repo

```
git clone --depth=1 --branch=29.x https://gitlab.com/tikiwiki/tiki.git .
```

Setting SSH to connect to Gitlab

To use the SSH key associated with your GitLab account, you need to create a config file in your ~/.ssh directory. Here's how to do it:

Go to the .ssh directory in your home folder (e.g., /home/domain/.ssh). If the directory does not exist, create it:

Create and set permissions for .ssh directory

```
mkdir -p ~/.ssh
chmod 700 ~/.ssh
```

Create and Configure the SSH Config File
Inside the .ssh directory, create a config file and add the following content:

Create config file and copy inside

```
# GitLab.com
Host gitlab.com
PreferredAuthentications publickey
IdentityFile ~/.ssh/id_rsa
```

Replace id_rsa with the filename of your private SSH key.
Set File Permissions
Ensure the config file has the correct ownership and permissions:

Set config file permission

```
chmod 600 ~/.ssh/config
chown your_user:your_user ~/.ssh/config
```

Test the connection to GitLab to confirm that your SSH key is being used:

test ssh connection to Gitlab

```
ssh -T -vvv git@gitlab.com
```

You should see a success message confirming the connection.
Once the SSH key and configuration are set, proceed with cloning your repository into the html directory.

Installing Tiki using ssh

Navigate into your html directory (public_html). If you don’t know where it is located on your new server on Virtualmin check the Virtual Server Summary.

Download Tiki from your git repo

```
git clone --branch=target_branch --depth=1 git@gitlab.com:youruser/your_repo .
```

Then it is required to run tiki setup.sh to install packages and fix the files and directories

permissions.

From version 27, Tiki uses a different method to complete the setup

From Tiki27 Tiki uses the Tiki 27 plus Build System.

This includes the integration of tools like Composer for PHP dependencies and Node.js for JavaScript and CSS dependencies. Please refer to another article, How to upgrade to Tiki Wiki 27 new Build System

Tiki setup.sh for version prior to Tiki27
From here follow the regular Tiki install process (setup.sh (see additional notes below), database creation) and you have a Tiki ready to be installed !