

InvoiceNinja

- [How to Install InvoiceNinja on Ubuntu 20.04 Server with Apache/Nginx](#)

How to Install InvoiceNinja on Ubuntu 20.04 Server with Apache/Nginx

This tutorial will be showing you how to install InvoiceNinja on Ubuntu 20.04 with Apache or Nginx web server. [InvoiceNinja](#) is an open-source, self-hosted invoice software, a low-cost alternative to commercial online invoice platforms such as Freshbooks. InvoiceNinja provides hosted invoice service, but if you like to self host the software, you can follow the instructions below.

InvoiceNinja Features

- With InvoiceNinja, you can send invoices to your clients by using your own domain name and brand.
- Manage invoicing for multiple businesses all under one account.
- Save time by automatically billing long-term clients with recurring invoices.
- Easily create and send beautiful proposals to your customers.
- Attach 3rd Party Files to Invoices.
- Create Project Tasks & Track Time.
- Organize and plan your client work with a visual project management tool.
- Allow your clients to see all their transactions with you in one glance.
- Zapier automation allows you to transfer data between your invoicing account and popular apps including Gmail, Google Sheets, QuickBooks Online, Slack, Pipeline, MailChimp, and hundreds more.
- Request deposits & partial payments using the same invoice again and again.
- Use a pre-written auto-reminder email sequence to remind clients your invoice needs to be paid.
- Receive notifications when a client views and pays your invoice.
- And many more

Prerequisites

First, you need a Linux server with at least 512MB RAM. You can click [this special link](#) to get \$100

free credit on DigitalOcean. (For new users only). If you are already a DigitalOcean user, then you can click [this special link](#) to get \$50 free credit on Vultr (for new users only). Once you have an account at DigitalOcean or Vultr, install Ubuntu 20.04 on your server and follow the instructions below.

InvoiceNinja requires PHP and MySQL/MariaDB. To follow this tutorial, you should have already set up a LAMP stack or LEMP stack. If you haven't already done so, please use one of the following guides.

- [How to Install LAMP \(Apache, MariaDB, PHP\) stack on Ubuntu 20.04](#)
- [How to Install LEMP \(Nginx, MariaDB, PHP\) stack on Ubuntu 20.04](#)

And you also need a domain name, so your clients can see the invoice via your domain name. I registered my domain name at [NameCheap](#) because the price is low and they give whois privacy protection free for life.

Now let's install InvoiceNinja.

Step 1: Download InvoiceNinja Install Zip File on Ubuntu 20.04 Server

Login into your Ubuntu 20.04 server via [SSH](#). Then run the following command to download the latest version of InvoiceNinja zip file onto your server.

```
wget -O invoice-ninja.zip https://download.invoiceninja.com/
```

Once downloaded, extract the archive to the `/var/www/` directory with `unzip`.

```
sudo apt install unzip

sudo mkdir -p /var/www/

sudo unzip invoice-ninja.zip -d /var/www/
```

The `-d` option specifies the target directory. InvoiceNinja web files will be extracted to `/var/www/ninja`. To better identify each directory, you can rename it to `invoice-ninja`.

```
sudo mv /var/www/ninja /var/www/invoice-ninja
```

Then we need to change the owner of this directory to `www-data` so that the web server can write to this directory.

```
sudo chown www-data:www-data /var/www/invoice-ninja/ -R
```

We also need to change the permission of the `storage` directory.

```
sudo chmod 755 /var/www/invoice-ninja/storage/ -R
```

Step 2: Create a Database and User in MariaDB

Log into MariaDB database server with the following command. Since MariaDB is now using `unix_socket` plugin to authentication user login, there's no need to enter MariaDB root password. We just need to prefix the `mysql` command with `sudo`.

```
sudo mysql
```

Then create a database for Invoice Ninja. This tutorial names the database `invoiceninja`. You can use whatever name you like.

```
create database invoiceninja;
```

Create the database user. Again, you can use your preferred name for this user. Replace `your-password` with your preferred password.

```
create user ninja@localhost identified by 'your-password';
```

Grant this user all privileges on the `invoiceninja` database.

```
grant all privileges on invoiceninja.* to ninja@localhost;
```

Flush privileges and exit.

```
flush privileges;
```

```
exit;
```

Step 3: Install PHP Modules

Note that InvoiceNinja currently doesn't support PHP7.4. For best compatibility, it's recommended to add the PHP7.3 PPA and install PHP7.3.

```
sudo add-apt-repository ppa:ondrej/php
```

Then run the following commands to install PHP modules required or recommended by InvoiceNinja.

```
sudo apt install php-imagick php7.3-fpm php7.3-mysql php7.3-common php7.3-gd php7.3-json  
php7.3-curl php7.3-zip php7.3-xml php7.3-mbstring php7.3-bz2 php7.3-intl php7.3-gmp
```

If you use Apache web server, then you need to disable the PHP module for Apache.

```
sudo a2dismod php7.4
```

You also need to disable the prefork MPM module.

```
sudo a2dismod mpm_prefork
```

Now you need to run the following command to enable three modules in order to use PHP-FPM, regardless of whether `mod_php` is installed on your server.

```
sudo a2enmod mpm_event proxy_fcgi setenvif
```

Then restart Apache.

```
sudo systemctl restart apache2
```

Step 4: Setting Up Web Server

We can use Apache or Nginx web server.

Apache

If you prefer Apache, create a virtual host file for Invoice Ninja.

```
sudo nano /etc/apache2/sites-available/invoice-ninja.conf
```

Put the following text into the file. Replace `invoice.yourdomain.com` with your own sub-domain for Invoice Ninja. Don't forget to set A record for the domain name in your DNS manager. (Note that the webroot is set to `/var/www/invoice-ninja/public/`, not `/var/www/invoice-ninja/`)

```
<VirtualHost *:80>
    ServerName invoice.yourdomain.com
    DocumentRoot /var/www/invoice-ninja/public

    <Directory /var/www/invoice-ninja/public>
        DirectoryIndex index.php
        Options +FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/invoice-ninja.error.log
    CustomLog ${APACHE_LOG_DIR}/invoice-ninja.access.log combined

    Include /etc/apache2/conf-available/php7.3-fpm.conf
</VirtualHost>
```

Save and close the file. Then enable this virtual host.

```
sudo a2ensite invoice-ninja.conf
```

We need to enable the rewrite module.

```
sudo a2enmod rewrite
```

Restart Apache for the changes to take effect.

```
sudo systemctl restart apache2
```

Now visit `invoice.yourdomain.com` and you will be redirected to the Invoice Ninja setup wizard page

(`invoice.yourdomain.com/setup`). If you see the default Apache page instead of the setup wizard, then you need to disable the default virtual host.

```
sudo a2dissite 000-default.conf
```

And restart Apache.

Before entering any information in the setup wizard, we need to enable HTTPS, which is explained in step 4.

Nginx

If you prefer Nginx, create a `invoice-ninja.conf` file in `/etc/nginx/conf.d/` directory.

```
sudo nano /etc/nginx/conf.d/invoice-ninja.conf
```

Put the following text into the file. Replace `invoice.yourdomain.com` with your own sub-domain for Invoice Ninja. Don't forget to set A record for the domain name in your DNS manager. (Note that the webroot is set to `/var/www/invoice-ninja/public/`, not `/var/www/invoice-ninja/`)

```
server {
    listen 80;
    listen [::]:80;
    server_name invoice.yourdomain.com;

    root /var/www/invoice-ninja/public;
    index index.php index.html index.htm;
    charset utf-8;

    location / {
        try_files $uri $uri/ /index.php?$query_string;
    }

    location = /favicon.ico { access_log off; log_not_found off; }
    location = /robots.txt { access_log off; log_not_found off; }

    access_log /var/log/nginx/invoiceninja.access.log;
    error_log /var/log/nginx/invoiceninja.error.log;

    location ~ \.php$ {
        fastcgi_split_path_info ^(.+\.(php))(/.+)$;
```

```
fastcgi_pass unix: /run/php/php7.3-fpm.sock;
fastcgi_index index.php;
include fastcgi_params;
fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
fastcgi_intercept_errors off;
fastcgi_buffer_size 16k;
fastcgi_buffers 4 16k;
}

location ~ /\.ht {
    deny all;
}
}
```

Save and close the file. Then test Nginx configuration.

```
sudo nginx -t
```

If the test is successful, reload Nginx for the changes to take effect.

```
sudo systemctl reload nginx
```

Now visit `|invoice.yourdomain.com|` and you will be redirected to the Invoice Ninja setup wizard page (`|invoice.yourdomain.com/setup|`). Before entering any information in the setup wizard, we need to enable HTTPS.

Step 5: Enabling HTTPS

To encrypt the HTTP traffic, we can enable HTTPS by installing a free TLS certificate issued from Let's Encrypt. Run the following command to install Let's Encrypt client (certbot) on Ubuntu 20.04 server.

```
sudo apt install certbot
```

If you use Nginx, then you also need to install the Certbot Nginx plugin.

```
sudo apt install python3-certbot-nginx
```

Next, run the following command to obtain and install TLS certificate.

```
sudo certbot --nginx --agree-tos --redirect --hsts --staple-ocsp --email you@example.com -d invoice.yourdomain.com
```

If you use Apache, install the Certbot Apache plugin.

```
sudo apt install python3-certbot-apache
```

And run this command to obtain and install TLS certificate.

```
sudo certbot --apache --agree-tos --redirect --hsts --staple-ocsp --email you@example.com -d invoice.yourdomain.com
```

Where

- `--nginx`: Use the nginx plugin.
- `--apache`: Use the Apache plugin.
- `--agree-tos`: Agree to terms of service.
- `--redirect`: Force HTTPS by 301 redirect.
- `--hsts`: Add the Strict-Transport-Security header to every HTTP response. Forcing browser to always use TLS for the domain. Defends against SSL/TLS Stripping.
- `--staple-ocsp`: Enables OCSP Stapling. A valid OCSP response is stapled to the certificate that the server offers during TLS.

The certificate should now be obtained and automatically installed.

IMPORTANT NOTES:

- Congratulations! Your certificate and chain have been saved at:
/etc/letsencrypt/live/invoice.████████.com/fullchain.pem
Your key file has been saved at:
/etc/letsencrypt/live/invoice.████████.com/privkey.pem
Your cert will expire on 2020-10-26. To obtain a new or tweaked version of this certificate in the future, simply run certbot again with the "certonly" option. To non-interactively renew *all* of your certificates, run "certbot renew"
- If you like Certbot, please consider supporting our work by:

Donating to ISRG / Let's Encrypt: <https://letsencrypt.org/donate>
Donating to EFF: <https://eff.org/donate-le>

Step 6: Finish Installation with the Setup Wizard

Now go to `invoice.yourdomain.com` and enter the database information, SMTP settings and create an admin user.

Application Settings

URL

HTTPS Require

Debug Enable

Database Connection

Driver MySQL

Host

Database

Username

Password

Success

If you would like to use your own mail server to send emails to clients, please check out the following article to set up your own mail server. **Note** that I highly recommend running iRedMail mail server on a fresh clean OS. Installing iRedMail on an OS that has other web applications can fail, and likely break existing applications.

- [How to easily set up a full-featured mail server on Ubuntu 20.04 with iRedMail](#)

When using your own email server, there are two drivers to choose from: `SMTP` and `Sendmail`.

- Choose `SMTP` if your mail server and Invoice Ninja runs on two different hosts.
- Choose `Sendmail` if your mail server and Invoice Ninja runs on the same host.

If you choose `SMTP`, use settings like below.

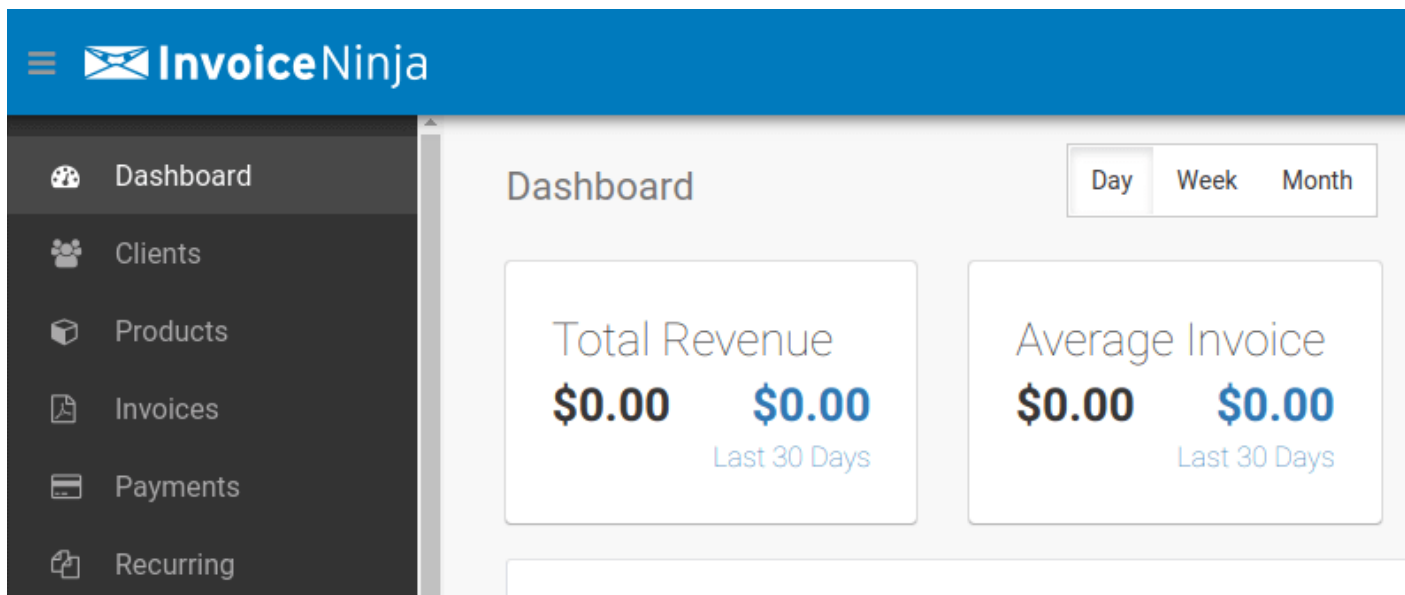
- **host:** mail.yourdomain.com **port:** 587 **Encryption:** TLS. And enter your username and

password.

If you choose `Sendmail`, use settings like below.

- **host:** 127.0.0.1 **port:** 25 **Encryption:** none. You don't need to enter username or password.

After creating the admin user, you can log into InvoiceNinja.



Set Up Cron Jobs

We need to set up Cron jobs to send recurring invoices and email reminders. Edit the root user's crontab file.

```
sudo crontab -e
```

Add the following lines at the end of this file.

```
#InvoiceNinja
0 8 * * * /usr/bin/php7.3 /var/www/invoice-ninja/artisan ninja: send-invoices > /dev/null
0 8 * * * /usr/bin/php7.3 /var/www/invoice-ninja/artisan ninja: send-reminders > /dev/null
```

Save and close the file. The two Cron jobs will run at 8 AM every day. You can also manually run a job like below.

```
sudo /usr/bin/php7.3 /var/www/invoice-ninja/artisan ninja: send-invoices
```