

# Welcome to the InternHub installation guide.

---

Please read along to set up your own **InternHub** instance.

## Before you start

---

There are some things you will need before you can get started developing with **InternHub**.

- Apache/Nginx (This guide was built using Nginx on Ubuntu Server 22.04. You can also use [MAMP](#) or [XAMPP](#), which are available for both macOS and Windows)
- MySQL server (database server)
- [VS Code](#) (code editor)
- [Beekeeper Studio](#) (database management software)
- [Git](#) (version control management)
- [Composer](#) (PHP package manager)
- [NPM](#) (JS package manager)

## Preparing your server

---

### Install all the necessary dependencies:

---

```
sudo apt install -y composer npm git php-mbstring  
php-imagick php-bcmath php-xml php-fpm php-zip  
php-intl php-gd php-common php-fpm php-cli unzip  
curl php-curl nginx redis php-redis  
mysql-server php-mysql;
```

## Getting InternHub up and running in your server

---

1. Clone the repository into your development environment.

```
git clone https://github.com/elvisblanco1993/intern-hub.git
```

2. Move into the project directory.

```
cd intern-hub
```

### 3. Create the environment file.

`cp .env.example .env` for UNIX based systems, and `copy .env.example .env` on MS Windows

### 4. Install back-end dependencies (this includes all packages InternHub depends on).

```
composer install
```

### 5. Install front-end dependencies.

```
npm install
```

### 6. Generate application key (this will help with encryption and security).

```
php artisan key:generate
```

### 7. Create database.

i. Open a terminal window, and access your MySQL server

```
sudo mysql -u root -p;
```

ii. Create your database and assign permissions

```
CREATE DATABASE internhub;  
CREATE USER 'internhub'@'localhost' IDENTIFIED BY '{YOUR_PASSWORD}';  
ALTER USER 'internhub'@'localhost' IDENTIFIED WITH mysql_native_password BY '{YOUR_PASSWORD}';  
GRANT ALL PRIVILEGES ON internhub.* to 'internhub'@'localhost' WITH GRANT OPTION;  
FLUSH PRIVILEGES;  
EXIT;
```

iii.\*Replace {YOUR\_PASSWORD} with a strong, secure password.\*

### 8. Add your database credentials to InternHub.

Now that you created your database, database username and password, it is time to connect your instance to it.

To do so, open your `.env` file, and modify the following lines.

```
DB_CONNECTION=mysql  
DB_HOST=127.0.0.1  
DB_PORT=3306
```

```
DB_DATABASE=internhub
DB_USERNAME=internhub
DB_PASSWORD=SET_YOUR_PASSWORD_HERE
```

## 9. Run migrations (this will create your database tables).

```
php artisan migrate && php artisan db:seed
```

## 10. Generate front-end assets.

Run `npm run build` if you are deploying on production, or `npm run dev` if you are deploying on a staging site want live reload

## 11. Fix filesystem permissions

```
sudo chgrp -R www-data . ;
sudo chown -R www-data:www-data storage;
sudo chown -R www-data:www-data bootstrap/cache;
chmod -R 775 ./storage;
chmod -R 775 bootstrap/cache;
```

## 12. Add InternHub to your web server

### Create Nginx File

```
sudo nano /etc/nginx/sites-available/internhub
```

```
server {
    listen 80;
    server_name internhub.localhost;
    root /var/www/internhub/public;

    add_header X-Frame-Options "SAMEORIGIN";
    add_header X-XSS-Protection "1; mode=block";
    add_header X-Content-Type-Options "nosniff";

    index index.php;

    charset utf-8;

    client_max_body_size 100M;

    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; }

error_page 404 /index.php;

location ~ /\.php$ {
    fastcgi_pass unix:/var/run/php/php8.1-fpm.sock; # Replace with correct PHP
    fastcgi_param SCRIPT_FILENAME $realpath_root$fastcgi_script_name;
    include fastcgi_params;
}

location ~ /\.(!well-known).* {
    deny all;
}

# Enable gzip compression
gzip on;
gzip_comp_level 5;
gzip_min_length 256;
gzip_proxied any;
# Compress all output labeled with one of the following MIME-types.
gzip_types
application/atom+xml
application/javascript
application/json
application/ld+json
application/manifest+json
application/rss+xml
application/vnd.geo+json
application/vnd.ms-fontobject
application/x-font-ttf
application/x-web-app-manifest+json
application/xhtml+xml
application/xml
font/opentype
image/bmp
image/svg+xml
image/x-icon
text/cache-manifest
text/css
text/plain
text/vcard
text/vnd.rim.location.xloc
text/vtt
text/x-component
text/x-cross-domain-policy;
}

```

---

## Enable NGINX Site

```
sudo ln -s /etc/nginx/sites-available/internhub /etc/nginx/sites-enabled/;
```

```
sudo rm /etc/nginx/sites-enabled/default;
```

## Restart Nginx Server

```
sudo systemctl restart nginx;
```

13. **Setup automated daily diggest emails** Lastly, since we will be sending a daily diggest email, we need to set up a cron job in our server. We will do this like so:

Open your cron file by running `crontab -e` and add the following line at the end of the file:

```
*/10 * * * * cd /path-to-your-project && php artisan schedule:run >> /dev/null  
2>&1
```

Make sure you replace '/path-to-your-project' with the actual path to your project.

Save your changes.

## You are all set!

---

Now your InternHub site should be up and running.

### What's next?

If you are deploying your site on a production environment, you will need to enable SSL certificates to ensure all traffic from and to your server is fully secure. You can follow [this guide](#) on how to get a free certificate from Let's Encrypt.