

# Base IMIS Deployment Manual (Linux Environment)

- [Basic Requirements](#)
- [Secured Server Setup](#)

# Basic Requirements

The deployment manual is based on the specific system requirements mentioned below and any deviations will require adjustments to the deployment process. The specific system requirements mentioned below are recommended to ensure a smooth and error-free deployment process. The web-application and database are recommended to be deployed and maintained in two separate servers of identical specifications. However, both the web application and database can be maintained on the same server with minor modification to the deployment process.

## 1.1 Server Requirements

The minimum server requirements recommended for the successful deployment and implementation of IMIS are mentioned below:

### Operating System

The deployment of Base IMIS is carried out on a Linux Server running Ubuntu 22.04 LTS “Jammy Jellyfish”. This is the recommended server and operating system specifications.

### Disk Space

The recommended minimum disk space for IMIS is 100 GB during the initial deployment. However, depending on the volume of the data and media files (images, GIS information, etc.) of the system, this requirement can be scaled up or scaled down as required.

### RAM

The recommended minimum RAM is 8 GB during the initial deployment. However, depending on the size and complexity of the GIS data and spatial processing, expected traffic and number of users, this requirement can be further scaled up. Additionally, further scaling down the RAM is not recommended.

### CPU

A minimum of 4 cores CPU is recommended for Base IMIS. However, depending on the size and complexity of the GIS data and spatial processing, expected traffic and number of users, this requirement can be further scaled up. Additionally, further scaling down the CPU cores is not recommended.

### Network Bandwidth

A minimum network bandwidth of 100 Mbps speed with 500 GB to 1 TB monthly data transfer is recommended. However, depending on the number of users of the system and expected traffic,

this requirement can be further scaled up or down as required.

## 1.2 Software Requirements

The software requirements for IMIS are mentioned below and its corresponding installation procedures are mentioned in sections below.

### Web Server

The webserver recommended and currently used in this deployment manual is Nginx (V 1.22.0). However, the Apache web servers can also be used with modifications.

### PHP

IMIS is developed using PHP version  $\geq 8$ . This is the recommended version of PHP and any upgrades/downgrades require modifications to both the deployment process and source code.

### Database

IMIS is designed and developed with PostgreSQL (V 14) database. For GIS data storage and processing, the PostGIS extension (V3) is used.

### Geoserver

IMIS currently uses Geoserver (V2.21.0) for rendering and displaying spatial data maintained in the system. This specific version of Geoserver is recommended to ensure bugs/issues do not arise in the system.

## 1.3 Deployment Engineer Requirements

The recommended skillset for the deployment engineer conducting the deployment of IMIS is mentioned below. These skillsets are critical to ensure smooth and error free deployment and maintenance of IMIS.

- Linux Server with Ubuntu OS

- Docker Container Concepts
- Docker-Compose tool for multi-container applications
- Networking concepts and Port configurations
- Firewall concepts and rules
- PostgreSQL
- PostGIS extension of PostgreSQL
- Geoserver
- Shell scripting knowledge

Apart from the deployment engineer, the minimum recommended skillset for the development/maintenance team is mentioned below:

- PHP language

- Laravel Framework
- PostgreSQL

- PostGIS extension of PostgreSQL
- NPM package manager
- Git version control
- Geoserver
- JavaScript
- jQuery
- Open Layers (OL)
- Technical Documentation

# Secured Server Setup

The following steps are recommended to be carried out to ensure the server is secure and protected from external access/ intrusions and disruptions.

## 2.1 User Access Management

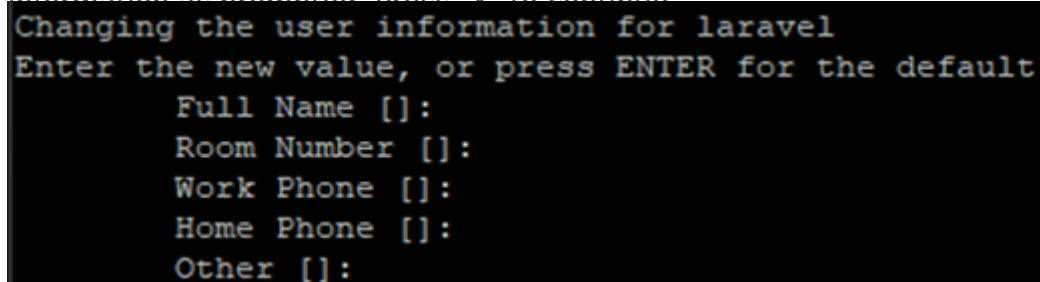
The root user is not recommended to be used for the server setup. Hence, a new user must be created. To add a new user and grant them superuser privileges, these steps are to be followed:

```
# adduser <<username>>
```

If root access is not available:

```
# sudo adduser <<username>>
```

Password prompt is displayed. Re-type the password to confirm password. Then fill out the user information as prompted. Press "Y" to continue.

A terminal window with a black background and yellow text. The text reads: "Changing the user information for laravel", "Enter the new value, or press ENTER for the default", "Full Name []:", "Room Number []:", "Work Phone []:", "Home Phone []:", and "Other []:".

```
Changing the user information for laravel
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
```

Add the user "<<username>>" to the "sudo" group to grant them superuser privileges:

```
usermod -aG sudo <<username>>
```

If root access is not available:

```
sudo usermod -aG sudo <<username>>
```

[Note: Ensure all commands are carried out through this user, as the docker build and docker compose commands require a UID and GID of 1000, which is assigned to the first user created in the OS]

## 2.2 Firewall Setup

To configure the firewall (ufw) to allow SSH connections:  
See which applications are registered with ufw/firewall.

```
ufw app list
```

There should be "OpenSSH" in the list.  
Allow OpenSSH through the firewall and enable the firewall.

[Note: Since the firewall blocks all connections except those explicitly allowed in the firewall rules, it is crucial to verify that the rules are accurate before enabling the firewall. Incorrect or missing rules could result in losing access to the server, so proceed with caution. For more information: [UFW Essentials guide](#). ]

```
ufw allow OpenSSH
ufw enable
```

Type Y and press ENTER to proceed. This will activate the firewall.  
To check the status of the firewall and ensure that SSH connections are allowed, type:

```
ufw status
```

## 2.3 Secure SSH Setup

It is recommended to configure all servers used for the deployment of the IMIS to use secure SSH (Secure Shell) authentication for remote access and server management. To enhance security, password-based login should be disabled, and only SSH key-based authentication should be allowed. This approach reduces the risk of unauthorized access by relying on cryptographic key pairs, which are significantly more secure than traditional passwords, thereby safeguarding the system against brute-force attacks and other vulnerabilities.

## 2.4 SSL Setup

SSL is recommended to ensure data security and prevent attacks. To secure your application with SSL, an SSL certification is required with the following certificates:

- private.pem (private key)
- fullchain.pem (full certificate)

SSL is required for both the application and the geoserver as well, thus two SSL certificates are required for IMIS. Additionally, to implement SSL, the corresponding domain/sub-domain names are also required for the web application and geoserver, that is mapped to the corresponding IP addresses of the servers. This procedure should be carried out at the end of the deployment process, after the deployment process is completed.

#### Configure Nginx for SSL in Docker

Modify Nginx configuration in Dockerfile to enable HTTPS. For more details refer to Annex: SSL Configuration in Docker section below.

#### Configure GeoServer to Use SSL

Configure GeoServer to use SSL; otherwise, it may create issues while displaying layers and styles. For more details refer to Annex: SSL Configuration in Geoserver below.

### 2.5 Data Backup Recommendation

The 3-tier backup strategy is recommended to ensure protection of data and ensure quick recovery in case of data loss. The 3-tier backup strategy is mentioned below:

#### Tier 1: Primary Backup (On-Site Daily Backup)

The Primary Backup is to be carried out daily, which is stored on-site (local server or external storage device). This provides quick access to the data for immediate recovery.

#### Tier 2: Secondary Backup (Off-site or Cloud Backup)

The Secondary Backup is to be carried out weekly, which is stored in a remote location or a separate cloud server. This protects the data from local disasters and data loss from main servers. This ensures a copy of the data is present even when the main server and primary backups are lost.

#### Tier 3: Tertiary Backup (Archival)

The Tertiary Backup is to be carried out monthly, which is stored in a highly secure and often offline or low-access environment. This enables long term retention of historical data, geo-redundancy of data and resilience against large scale incidents.