Installation Guide

BigLan Network Monitoring System

Contents

Introduction 3

Hardware requirements 3

Automated Installation (Recommended) 4

Manual Installation (Advanced) 5

1. Installing Ubuntu Server 24.04 LTS 5

2. Installing Apache 2 5

3. Installing MySQL 5

4. Create a database and set up a user 5

5. Installing SNMP 5

6. Installing and configuring PHP 5

7. Firewall settings 5

8. Git (if needed) 5

9. Create a folder for your files 6

10. Download project from GitHub to folder 6

11. Installing Composer and project dependencies 6

12. public folder setup 6

13. Enable SSL 6

14. Laravel .env file settings 6

15. Laravel APP key generation 7

16. Apache restart web server 7

17. Database migration 7

18. Laravel clear cache 7

19. Populate database with default data 7

20. Schedule a CronJob for Laravel 7

21. Set folder permissions 8

22. Register the local subnet 8

23. Create the first administrator account from the terminal 8

Troubleshooting 9

"Access denied for user 'root'@'localhost'" even with the correct password 9

Website shows "Permission denied" for storage/logs/laravel.log 9

The Console tab does not appear on any workstation 9

The Console tab appears, but commands never get a response 9

Detecting database errors with phpMyAdmin 9

Introduction

This document guides you through the steps of installing the BigLan Network Monitoring System.

> **NOTE: Successful installation and configuration requires experience and knowledge of Linux server and web server components (such as Apache).**

The application also includes a Laravel-based web interface, so Laravel experience is also highly recommended for more efficient troubleshooting and customization.

> **IMPORTANT! For optimal and secure operation of the application, it is recommended to use it within a local area network (LAN) or intranet. This environment minimizes security risks, as sensitive network data does not leave the internal network. Although the application may be able to communicate over the Internet, this option should only be used over an encrypted channel (e.g. VPN) and with appropriate firewall rules in place. Use over the open Internet poses increased security risks, including the possibility of unauthorized access and data interception.**
>
> **WARNING! Although the system can operate with a self-signed certificate, which can be used for testing purposes or on an internal network, this solution is not secure. Browsers may display a warning and the communication is not properly encrypted. It is strongly recommended to create your own internal certificate authority (Local CA). This method allows the web server operating on the internal network to have a trusted certificate. Certificates issued by your own CA must be installed by all devices on the network (for example, client computers) to be reliable.**
>
> **WARNING! The user is responsible for the security settings and risks encountered when using the product on an external network.**
>
> **IMPORTANT! For the application to function properly, the server must have access to the VLANs in which the monitored workstations and servers are located.**
>
> **IMPORTANT! For the application to function properly, the server must have at least one IP address in each IP range where the monitored workstations and servers are located. This ensures that the application can access different network segments.**

Hardware requirements

The basic hardware requirements for the application are as follows, depending on the number of monitored devices:

- Processor: Quad-core CPU
- Memory: 4GB RAM
- Storage: 40 GB free space

These values are rough estimates and apply to a typical network environment where the load consists of:

- Monitoring of 500 workstations/servers
- Monitor 100 network printers
- Monitor 100 monitoring status changes per minute

> **IMPORTANT! For larger networks and higher loads, hardware requirements may also increase.**

Automated Installation (Recommended)

BigLan Server can be installed automatically with a single script that performs every step described in the Manual Installation section for you - including database setup, PHP configuration, the Laravel application setup, and the scheduled background task. This is the recommended installation method for most users.

Prerequisite: a fresh Ubuntu Server 24.04 LTS installation with SSH access and sudo privileges.

```
curl -O https://raw.githubusercontent.com/atlantisguru/biglan-server/main/install-biglan.sh
chmod +x install-biglan.sh
sudo bash install-biglan.sh
```

The script will ask you a few questions up front (installation folder, database name/username/password, timezone, server address, default language, and whether to enable a self-signed SSL certificate), then runs unattended, printing its progress step by step.

> At the end, the script prints a randomly generated MASTER_KEY. Store it somewhere safe - it cannot be recovered later without the .env file, and it is required for the encrypted remote command feature (Console) described in the User Guide.

If you would like to understand, customize, or troubleshoot an individual step, the Manual Installation section below describes exactly what the script automates.

Manual Installation (Advanced)

1. Installing Ubuntu Server 24.04 LTS

```
https://ubuntu.com/tutorials/install-ubuntu-server
```

2. Installing Apache 2

```
sudo apt install apache2
```

3. Installing MySQL

```
sudo apt install mysql-server
```

After installation, it is worth running the MySQL security script.

```
sudo mysql_secure_installation
```

> On Ubuntu, the MySQL root account uses socket authentication (auth_socket) by default - it can only connect locally as the Linux root user, and never with a password over a normal connection. Do not use root as the application's database user in the next step; always create a dedicated user, as shown below.

4. Create a database and set up a user

```
mysql -u [username] -p
CREATE DATABASE [database_name e.g.: biglan] CHARACTER SET utf8mb4
COLLATE utf8mb4_unicode_ci;
CREATE USER 'admin'@'localhost' IDENTIFIED BY 'password'; GRANT ALL
PRIVILEGES ON *.* TO 'admin'@'localhost' WITH GRANT OPTION; FLUSH
PRIVILEGES;
exit;
```

5. Installing SNMP

```
sudo apt update
sudo apt install snmp snmpd
```

6. Installing and configuring PHP

```
sudo apt install php libapache2-mod-php php-mysql php-snmp php-xml php-
zip php-curl php-mbstring php-bcmath
```

> php-mbstring and php-bcmath are required by Laravel and Composer. Without them, step 11 (composer install) will fail.

7. Firewall settings

```
sudo ufw allow 8080/tcp
sudo ufw allow 80/tcp
sudo ufw allow 443/tcp
```

> Port 8080 is used for the encrypted remote command channel to workstations (Console). Ports 80/443 are needed for the web interface itself, if ufw is active.

8. Git (if needed)

```
sudo apt install git
```

9. Create a folder for your files

```
sudo mkdir /var/www/biglan
```

Permission for the web server to write to the folder

```
sudo chown -R $USER:www-data /var/www/biglan
```

10. Download project from GitHub to folder

```
cd /var/www/biglan
git clone https://github.com/atlantisguru/biglan-server.git
```

11. Installing Composer and project dependencies

```
sudo apt install composer
```

> This only installs the Composer program itself. You must also install the project's PHP dependencies - this step is easy to miss but required, since the application will not run without it:

```
cd /var/www/biglan
composer install --no-interaction --optimize-autoloader
```

12. public folder setup

```
sudo nano /etc/apache2/sites-available/000-default.conf

<VirtualHost *:80>
 ServerAdmin webmaster@localhost
 DocumentRoot /var/www/biglan/public
<Directory /var/www/biglan/public>
 AllowOverride All
 Require all granted
</Directory>
 ErrorLog ${APACHE_LOG_DIR}/error.log
 CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

13. Enable SSL

```
sudo a2enmod ssl
sudo a2ensite default-ssl
sudo nano /etc/apache2/sites-available/default-ssl.conf

<VirtualHost *:443>
 ServerAdmin webmaster@localhost
 DocumentRoot /var/www/biglan/public
<Directory /var/www/biglan/public>
 AllowOverride All
 Require all granted
</Directory>
 ErrorLog ${APACHE_LOG_DIR}/error.log
 CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>

sudo a2enmod rewrite
```

14. Laravel .env file settings

```
cd /var/www/biglan
sudo cp .env.example .env
sudo nano .env

APP_NAME=BigLan
APP_ENV=production
APP_TIMEZONE=[Your time zone]
APP_URL=https://[Server IP Address]
MASTER_KEY=[A 32-character token generated from lowercase and
uppercase letters and numbers]
APP_LOCALE=en
APP_FALLBACK_LOCALE=en
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=biglan
DB_USERNAME=[Username]
DB_PASSWORD=[Password]
```

> The .env file must exist before generating the application key in the next step - always copy it first, as shown above.

15. Laravel APP key generation

```
cd /var/www/biglan
php artisan key:generate
```

16. Apache restart web server

```
sudo systemctl restart apache2
```

17. Database migration

```
php artisan config:clear
php artisan migrate
```

> Do not run "php artisan cache:table" - the cache table migration is already included in the project. Running it again will fail with "Migration already exists."

18. Laravel clear cache

```
php artisan cache:clear
php artisan config:cache
```

19. Populate database with default data

```
php artisan db:seed
```

20. Schedule a CronJob for Laravel

```
sudo crontab -u www-data -e
```

Choose option 1 (edit with nano).

Add this line to the document:

```
* * * * * cd /var/www/biglan && php artisan schedule:run >> /dev/null
2>&1
```

IMPORTANT! Schedule this under the www-data user, not root, as shown above. If it runs as root, log files created by the scheduler become owned by root, and the web server (which runs as www-data) will then be unable to write to them - causing "Permission denied" errors on the website.

21. Set folder permissions

```
sudo chown -R www-data:www-data /var/www/biglan/storage
/var/www/biglan/bootstrap/cache
sudo chmod -R 775 /var/www/biglan/storage /var/www/biglan/bootstrap/cache
```

This ensures both the web server and the scheduled task can always write logs and cache files, regardless of which one touches them first.

22. Register the local subnet

Without at least one subnet registered, BigLan cannot determine that a workstation is on the local network, and the Console (remote command) tab will never appear for any workstation - regardless of actual network topology. After loading the website (next step) and logging in, go to IP Table -> New Subnet and register the network your workstations are on, for example:

- Name: LAN
- ID: 192.168.1.0
- Mask: 24
- Gateway: 192.168.1.1

23. Create the first administrator account from the terminal

```
sudo -u www-data php artisan biglan:create-admin
```

The command will ask for your full name, your email address, and a password, entered twice to confirm (minimum 8 characters). Note: public registration is disabled by default, so this is the only way to create the first account.

Find the BigLan User Guide in the Documents menu.

Troubleshooting

Restart the web server after every operation that affects the web server.

```
sudo systemctl restart apache2
```

Run a config cache flush after every modification to Laravel's .env file. Always initiate this from the project folder (/var/www/biglan).

```
php artisan cache:clear
php artisan config:cache
```

"Access denied for user 'root'@'localhost'" even with the correct password

This happens because Ubuntu's MySQL root account uses socket authentication, which ignores passwords entirely over a normal connection. Use the dedicated database user created in step 4 instead of root in your .env file.

Website shows "Permission denied" for storage/logs/laravel.log

This means the log file was created by a different user than the web server (usually because the scheduled task in step 20 was set up under root instead of www-data). Fix the crontab entry to run under www-data, then re-apply step 21's ownership/permission commands.

The Console tab does not appear on any workstation

Register at least one subnet under IP Table -> New Subnet, matching the network the workstations are actually on (see step 22). The Console tab is only shown for workstations whose IP address falls inside a registered subnet.

The Console tab appears, but commands never get a response

This is almost always caused by the Windows Firewall on the workstation silently blocking the inbound connection on port 8080. Add an inbound rule on the workstation (run as Administrator in PowerShell):

```
New-NetFirewallRule -DisplayName "BigLanService Console" -Direction
Inbound -Protocol TCP -LocalPort 8080 -Action Allow
```

Detecting database errors with phpMyAdmin

To detect database errors, it is recommended to install phpMyAdmin.

```
sudo apt install phpmyadmin
sudo nano /etc/apache2/conf-available/phpmyadmin.conf

#
# Apache config file for phpMyAdmin
#
Alias /phpmyadmin /usr/share/phpmyadmin
<Directory /usr/share/phpmyadmin>
 Options Indexes FollowSymLinks
 DirectoryIndex index.php
<IfModule mod_authz_core.c>
# Apache 2.4
 Require all granted
</IfModule>
<IfModule !mod_authz_core.c>
```

```
# Apache 2.2
 Order Deny,Allow
 Deny from All
 Allow from 127.0.0.1
 Allow from ::1
</IfModule>
</Directory>

sudo a2enconf phpmyadmin
sudo systemctl restart apache2
```

Do not log in to phpMyAdmin as "root" with a blank password - this will fail for the same auth_socket reason described above. Use the dedicated database user from step 4.