Guía de Instalación

Sistema de Monitorización de Red BigLan

Contenido

Introducción

Este documento le guía a través de los pasos de instalación del Sistema de Monitorización de Red BigLan.

> **NOTA: La instalación y configuración correctas requieren experiencia y conocimientos de servidores Linux y de componentes de servidor web (como Apache).**

La aplicación también incluye una interfaz web basada en Laravel, por lo que también se recomienda encarecidamente tener experiencia con Laravel para una resolución de problemas y una personalización más eficaces.

> **¡IMPORTANTE! Para un funcionamiento óptimo y seguro de la aplicación, se recomienda utilizarla dentro de una red de área local (LAN) o intranet. Este entorno minimiza los riesgos de seguridad, ya que los datos de red sensibles no salen de la red interna. Aunque la aplicación puede comunicarse a través de Internet, esta opción solo debe utilizarse mediante un canal cifrado (por ejemplo, VPN) y con las reglas de firewall adecuadas configuradas. El uso a través de Internet abierto supone un mayor riesgo de seguridad, incluyendo la posibilidad de acceso no autorizado e interceptación de datos.**
>
> **¡ADVERTENCIA! Aunque el sistema puede funcionar con un certificado autofirmado, útil para pruebas o en una red interna, esta solución no es segura. Los navegadores pueden mostrar una advertencia y la comunicación no está correctamente cifrada. Se recomienda encarecidamente crear su propia autoridad de certificación interna (CA local). Este método permite que el servidor web que opera en la red interna disponga de un certificado de confianza. Los certificados emitidos por su propia CA deben instalarse en todos los dispositivos de la red (por ejemplo, los equipos cliente) para que sean de confianza.**
>
> **¡ADVERTENCIA! El usuario es responsable de la configuración de seguridad y de los riesgos derivados del uso del producto en una red externa.**
>
> **¡IMPORTANTE! Para que la aplicación funcione correctamente, el servidor debe tener acceso a las VLAN en las que se encuentran las estaciones de trabajo y servidores monitorizados.**
>
> **¡IMPORTANTE! Para que la aplicación funcione correctamente, el servidor debe tener al menos una dirección IP en cada rango de IP en el que se encuentren las estaciones de trabajo y servidores monitorizados. Esto garantiza que la aplicación pueda acceder a los distintos segmentos de red.**

Requisitos de hardware

Los requisitos de hardware básicos de la aplicación son los siguientes, en función del número de dispositivos monitorizados:

- Procesador: CPU de cuatro núcleos
- Memoria: 4 GB de RAM
- Almacenamiento: 40 GB de espacio libre

Estos valores son estimaciones aproximadas y se aplican a un entorno de red típico donde la carga consiste en:

- Monitorización de 500 estaciones de trabajo/servidores
- Monitorización de 100 impresoras de red
- Monitorización de 100 cambios de estado de supervisión por minuto

¡IMPORTANTE! Para redes más grandes y cargas mayores, los requisitos de hardware también pueden aumentar.

Instalación Automatizada (Recomendada)

El servidor BigLan se puede instalar automáticamente con un único script que realiza por usted todos los pasos descritos en la sección Instalación Manual — incluyendo la configuración de la base de datos, la configuración de PHP, la puesta en marcha de la aplicación Laravel y la tarea programada en segundo plano. Este es el método de instalación recomendado para la mayoría de los usuarios.

Requisito previo: una instalación limpia de Ubuntu Server 24.04 LTS con acceso SSH y privilegios de sudo.

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

El script le hará algunas preguntas al principio (carpeta de instalación, nombre/usuario/contraseña de la base de datos, zona horaria, dirección del servidor, idioma predeterminado y si desea habilitar un certificado SSL autofirmado), y después se ejecutará sin supervisión, mostrando su progreso paso a paso.

> Al final, el script muestra una MASTER_KEY generada aleatoriamente. Guárdela en un lugar seguro — no se puede recuperar posteriormente sin el archivo .env, y es necesaria para la función de ejecución remota cifrada (Consola) descrita en la Guía del Usuario.

Una vez finalizado, abra la dirección mostrada en su navegador y registre el primer usuario — véase «Carga del sitio web y registro de un usuario» al final de esta guía.

Si desea entender, personalizar o solucionar problemas de algún paso concreto, la sección Instalación Manual describe exactamente lo que automatiza el script.

Instalación Manual (Avanzada)

1. Instalación de Ubuntu Server 24.04 LTS

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

2. Instalación de Apache 2

```
sudo apt install apache2
```

3. Instalación de MySQL

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

Después de la instalación, es recomendable ejecutar el script de seguridad de MySQL.

```
sudo mysql_secure_installation
```

> En Ubuntu, la cuenta root de MySQL utiliza autenticación por socket (auth_socket) de forma predeterminada — solo permite conectarse localmente como el usuario root de Linux, nunca con contraseña a través de una conexión normal. En el siguiente paso, NO utilice root como usuario de base de datos de la aplicación — cree siempre un usuario dedicado, como se muestra a continuación.

4. Creación de la base de datos y configuración del usuario

```
mysql -u [nombre_usuario] -p
CREATE DATABASE [nombre_base_datos ej.: biglan] CHARACTER SET utf8mb4
COLLATE utf8mb4_unicode_ci;
CREATE USER 'admin'@'localhost' IDENTIFIED BY 'contraseña'; GRANT ALL
PRIVILEGES ON *.* TO 'admin'@'localhost' WITH GRANT OPTION; FLUSH
PRIVILEGES;
exit;
```

5. Instalación de SNMP

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

6. Instalación y configuración de PHP

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

> Laravel y Composer requieren los paquetes php-mbstring y php-bcmath. Sin ellos, el paso 11 (composer install) fallará.

7. Configuración del firewall

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

> El puerto 8080 se utiliza para el canal cifrado de comandos remotos hacia las estaciones de trabajo (Consola). Los puertos 80/443 son necesarios para la propia interfaz web, si ufw está activo.

8. Git (si es necesario)

```
sudo apt install git
```

9. Creación de una carpeta para los archivos

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

Permiso para que el servidor web pueda escribir en la carpeta

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

10. Descarga del proyecto desde GitHub a la carpeta

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

11. Instalación de Composer y las dependencias del proyecto

```
sudo apt install composer
```

> Esto solo instala el propio programa Composer. A continuación también debe instalar las dependencias PHP del proyecto — este paso es fácil de omitir, pero obligatorio, ya que sin él la aplicación no funcionará:

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

12. Configuración de la carpeta public

```
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. Habilitación de 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. Configuración del archivo .env de Laravel

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

APP_NAME=BigLan
APP_ENV=production
APP_TIMEZONE=[Su zona horaria]
APP_URL=https://[Dirección IP del servidor]
MASTER_KEY=[Token de 32 caracteres generado con letras minúsculas,
mayúsculas y números]
APP_LOCALE=es
APP_FALLBACK_LOCALE=en
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=biglan
DB_USERNAME=[Nombre de usuario]
DB_PASSWORD=[Contraseña]
```

> El archivo .env debe existir antes de generar la clave de la aplicación en el siguiente paso — cópielo siempre primero, como se muestra arriba.

15. Generación de la clave APP de Laravel

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

16. Reinicio del servidor web Apache

```
sudo systemctl restart apache2
```

17. Migración de la base de datos

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

> No ejecute «php artisan cache:table» — la migración de la tabla de caché ya está incluida en el proyecto. Si la vuelve a ejecutar, fallará con «Migration already exists».

18. Limpieza de la caché de Laravel

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

19. Población de la base de datos con datos predeterminados

```
php artisan db:seed
```

20. Programación de una tarea CronJob para Laravel

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

Elija la opción 1 (editar con nano).

Añada esta línea al documento:

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

¡IMPORTANTE! Programe esta tarea bajo el usuario www-data, no bajo root, como se muestra arriba. Si se ejecuta como root, los archivos de registro creados por el planificador quedarán en propiedad de root, y el servidor web (que se ejecuta como www-data) no podrá escribir en ellos — lo que provoca errores de «Permission denied» en el sitio web.

21. Configuración de permisos de carpetas

```
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
```

Esto garantiza que tanto el servidor web como la tarea programada puedan escribir siempre registros y archivos de caché, sin importar cuál de los dos los toque primero.

22. Registro de la subred local

Sin al menos una subred registrada, BigLan no puede determinar que una estación de trabajo está en la red local, y la pestaña Consola (comandos remotos) nunca aparecerá para ninguna estación de trabajo, independientemente de la topología de red real. Después de cargar el sitio web (paso siguiente) e iniciar sesión, vaya a Tabla IP -> Nueva Subred y registre la red en la que se encuentran sus estaciones de trabajo, por ejemplo:

- Nombre: LAN
- ID: 192.168.1.0
- Máscara: 24
- Puerta de enlace: 192.168.1.1

23. Carga del sitio web en el navegador y registro de un usuario

Tras la instalación, el primer usuario registrado se confirma automáticamente y recibe privilegios administrativos completos.

Encuentre la Guía del Usuario de BigLan en el menú Documentos.

Solución de problemas

Reinicie el servidor web después de cada operación que lo afecte.

```
sudo systemctl restart apache2
```

Ejecute una limpieza de la caché de configuración después de cada modificación del archivo .env de Laravel. Inicie siempre este proceso desde la carpeta del proyecto (/var/www/biglan).

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

«Access denied for user 'root'@'localhost'» incluso con la contraseña correcta

Esto ocurre porque la cuenta root de MySQL en Ubuntu utiliza autenticación por socket, que ignora por completo las contraseñas en una conexión normal. Utilice en su archivo .env el usuario de base de datos dedicado creado en el paso 4, en lugar de root.

El sitio web muestra «Permission denied» para storage/logs/laravel.log

Esto significa que el archivo de registro fue creado por un usuario distinto al del servidor web (normalmente porque la tarea programada del paso 20 se configuró bajo root en lugar de www-data). Corrija la entrada de crontab para que se ejecute bajo www-data, y vuelva a aplicar los comandos de propiedad/permisos del paso 21.

La pestaña Consola no aparece en ninguna estación de trabajo

Registre al menos una subred en Tabla IP -> Nueva Subred, que coincida con la red en la que realmente se encuentran las estaciones de trabajo (véase el paso 22). La pestaña Consola solo se muestra para las estaciones de trabajo cuya dirección IP esté dentro de una subred registrada.

La pestaña Consola aparece, pero los comandos nunca reciben respuesta

Esto casi siempre lo provoca el Firewall de Windows en la estación de trabajo, que bloquea silenciosamente la conexión entrante en el puerto 8080. Añada una regla de entrada en la estación de trabajo (ejecute como Administrador en PowerShell):

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

Detección de errores de base de datos con phpMyAdmin

Para detectar errores de base de datos, se recomienda instalar 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
```

No inicie sesión en phpMyAdmin como «root» con contraseña en blanco — fallará por el mismo motivo de auth_socket explicado arriba. Utilice el usuario de base de datos dedicado del paso 4.