

Bookstack

[Link zu BookStack](#)

- [BookStack - Installation unter Debian 13 \(Trixie\)](#)
- [BookStack - Backup / Restore unter Debian 13 \(Trixie\)](#)
- [BookStack - Upgrade unter Debian 13 \(Trixie\)](#)

BookStack - Installation unter Debian 13 (Trixie)

```
wget https://codeberg.org/bookstack/devops/raw/branch/main/scripts/installation-debian-13.sh
```

```
#!/bin/bash

echo "THIS SCRIPT IS NOT CONSIDERED OFFICIALLY SUPPORTED!"
echo "Only our Ubuntu LTS scripts are considered supported. This is community"
echo "maintained and provided for convenience. It may not be up-to-date or may"
echo "have unresolved issues. Use with caution."
echo ""

echo "This installs a new BookStack instance on a fresh Debian 13 (Trixie) server."
echo "This script does not ensure system security."
echo ""

# Generate a path for a log file to output into for debugging
LOGPATH=$(realpath "bookstack_install_$(date +%s).log")

# Get the current user running the script
SCRIPT_USER="${SUDO_USER:-$USER}"

# Get the current machine IP address
CURRENT_IP=$(ip addr | grep 'state UP' -A4 | grep 'inet ' | awk '{print $2}' | cut -f1 -d'/')

# Generate a password for the database
DB_PASS="$(head /dev/urandom | tr -dc A-Za-z0-9 | head -c 13)"

# The directory to install BookStack into
BOOKSTACK_DIR="/var/www/bookstack"

# Get the domain from the arguments (Requested later if not set)
DOMAIN=$1
```

```

# Prevent interactive prompts in applications
export DEBIAN_FRONTEND=noninteractive

# Echo out an error message to the command line and exit the program
# Also logs the message to the log file
function error_out() {
    echo "ERROR: $1" | tee -a "$LOGPATH" 1>&2
    exit 1
}

# Echo out an information message to both the command line and log file
function info_msg() {
    echo "$1" | tee -a "$LOGPATH"
}

# Run some checks before installation to help prevent messing up an existing
# web-server setup.
function run_pre_install_checks() {
    # Check we're running as root and exit if not
    if [[ $EUID -gt 0 ]]
    then
        error_out "This script must be ran with root/sudo privileges"
    fi

    # Check if Apache appears to be installed and exit if so
    if [ -d "/etc/apache2/sites-enabled" ]
    then
        error_out "This script is intended for a fresh server install, existing apache config
found, aborting install"
    fi

    # Check if MySQL/MariaDB appears to be installed and exit if so
    if [ -d "/var/lib/mysql" ]
    then
        error_out "This script is intended for a fresh server install, existing MySQL data found,
aborting install"
    fi
}

# Fetch domain to use from first provided parameter,

```

```

# Otherwise request the user to input their domain
function run_prompt_for_domain_if_required() {
    if [ -z "$DOMAIN" ]
    then
        info_msg ""
        info_msg "Enter the domain (or IP if not using a domain) you want to host BookStack on and
press [ENTER]."

```

```

function run_bookstack_download() {
    cd /var/www || exit
    git clone https://github.com/BookStackApp/BookStack.git --branch release --single-branch
bookstack
}

# Install BookStack composer dependencies
function run_download_bookstack_vendor_files() {
    cd "$BOOKSTACK_DIR" || exit
    php bookstack-system-cli download-vendor
}

# Copy and update BookStack environment variables
function run_update_bookstack_env() {
    cd "$BOOKSTACK_DIR" || exit
    cp .env.example .env
    sed -i.bak "s@APP_URL=.*\@$@APP_URL=http://$DOMAIN@" .env
    sed -i.bak 's/DB_DATABASE=.*$/DB_DATABASE=bookstack/' .env
    sed -i.bak 's/DB_USERNAME=.*$/DB_USERNAME=bookstack/' .env
    sed -i.bak "s/DB_PASSWORD=.*\$/DB_PASSWORD=$DB_PASS/" .env
    # Generate the application key
    php artisan key:generate --no-interaction --force
}

# Run the BookStack database migrations for the first time
function run_bookstack_database_migrations() {
    cd "$BOOKSTACK_DIR" || exit
    php artisan migrate --no-interaction --force
}

# Set file and folder permissions
# Sets current user as owner user and www-data as owner group then
# provides group write access only to required directories.
# Hides the `.env` file so it's not visible to other users on the system.
function run_set_application_file_permissions() {
    cd "$BOOKSTACK_DIR" || exit
    chown -R "$SCRIPT_USER":www-data ./
    chmod -R 755 ./
    chmod -R 775 bootstrap/cache public/uploads storage
    chmod 740 .env

```

```

# Tell git to ignore permission changes
git config core.fileMode false
}

# Setup apache with the needed modules and config
function run_configure_apache() {
    # Enable required apache modules and config
    a2enmod rewrite proxy_fcgi setenvif
    a2enconf php8.4-fpm

    # Set-up the required BookStack apache config
    cat >/etc/apache2/sites-available/bookstack.conf <<EOL
<VirtualHost *:80>
    ServerName ${DOMAIN}

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/bookstack/public/

    <Directory /var/www/bookstack/public/>
        Options -Indexes +FollowSymLinks
        AllowOverride None
        Require all granted
        <IfModule mod_rewrite.c>
            <IfModule mod_negotiation.c>
                Options -MultiViews -Indexes
            </IfModule>

            RewriteEngine On

            # Handle Authorization Header
            RewriteCond %{HTTP:Authorization} .
            RewriteRule .* - [E=HTTP_AUTHORIZATION:%{HTTP:Authorization}]

            # Redirect Trailing Slashes If Not A Folder...
            RewriteCond %{REQUEST_FILENAME} !-d
            RewriteCond %{REQUEST_URI} (.+)/$
            RewriteRule ^ %1 [L,R=301]

            # Handle Front Controller...

```

```

        RewriteCond %{REQUEST_FILENAME} !-d
        RewriteCond %{REQUEST_FILENAME} !-f
        RewriteRule ^ index.php [L]
    </IfModule>
</Directory>

ErrorLog \${APACHE_LOG_DIR}/error.log
CustomLog \${APACHE_LOG_DIR}/access.log combined

</VirtualHost>
EOL

# Disable the default apache site and enable BookStack
a2dissite 000-default.conf
a2ensite bookstack.conf

# Restart apache to load new config
systemctl restart apache2
# Ensure php-fpm service has started
systemctl start php8.4-fpm.service
}

info_msg "This script logs full output to $LOGPATH which may help upon issues."
sleep 1

run_pre_install_checks
run_prompt_for_domain_if_required
info_msg ""
info_msg "Installing using the domain or IP \"${DOMAIN}\""
info_msg ""
sleep 1

info_msg "[1/8] Installing required system packages... (This may take several minutes)"
run_package_installs >> "$LOGPATH" 2>&1

info_msg "[2/8] Preparing MySQL database..."
run_database_setup >> "$LOGPATH" 2>&1

info_msg "[3/8] Downloading BookStack to ${BOOKSTACK_DIR}..."
run_bookstack_download >> "$LOGPATH" 2>&1

```

```
info_msg "[4/8] Downloading PHP dependency files..."
run_download_bookstack_vendor_files >> "$LOGPATH" 2>&1

info_msg "[5/8] Creating and populating BookStack .env file..."
run_update_bookstack_env >> "$LOGPATH" 2>&1

info_msg "[6/8] Running initial BookStack database migrations..."
run_bookstack_database_migrations >> "$LOGPATH" 2>&1

info_msg "[7/8] Setting BookStack file & folder permissions..."
run_set_application_file_permissions >> "$LOGPATH" 2>&1

info_msg "[8/8] Configuring apache server..."
run_configure_apache >> "$LOGPATH" 2>&1

info_msg "-----"
info_msg "Setup finished, your BookStack instance should now be installed!"
info_msg "- Default login email: admin@admin.com"
info_msg "- Default login password: password"
info_msg "- Access URL: http://$CURRENT_IP/ or http://$DOMAIN/"
info_msg "- BookStack install path: $BOOKSTACK_DIR"
info_msg "- Install script log: $LOGPATH"
info_msg "-----"
```

BookStack - Backup / Restore unter Debian 13 (Trixie)

Backup

Backup Datenbank

```
mysqldump -u root bookstack > bookstack.backup.sql
```

Backup Dateien

```
tar -czvf /root/bookstack-files-backup.tar.gz /var/www/bookstack/.env  
/var/www/bookstack/public/uploads /var/www/bookstack/storage/uploads /var/www/bookstack/themes
```

Restore

Restore Datenbank

```
mysql -u root -p bookstack < bookstack.backup.sql
```

Restore der Dateien

Wird im Verzeichnis **/var/www/bookstack** ausgeführt!

```
cd /var/www/bookstack && tar -xvzf /root/bookstack-files-backup.tar.gz
```

Änderungen Dateisystem

Nach den Restore müssen die Dateiberechtigungen korrigiert werden.

Dateibesitz ändern:

```
chown -R www-data:www-data /var/www/bookstack
```

Dateiberechtigungen setzen:

```
chmod -R 755 /var/www/bookstack  
chmod -R 775 /var/www/bookstack/storage /var/www/bookstack/bootstrap/cache  
/var/www/bookstack/public/uploads  
chmod 640 /var/www/bookstack/.env
```

Änderungen Datenbank

Falls der Restore nach einer Neuinstallation geschieht, muss das Passwort des BookStack-Datenbankbenutzers korrigiert werden.

Das Datenbankpasswort aus der **.env** Datei in der Mariadb setzen:

```
mysql -u root
```

```
ALTER USER 'bookstack'@'localhost' IDENTIFIED BY 'passwort_aus_der_.env';  
exit;
```

BookStack - Upgrade unter Debian 13 (Trixie)

Vorbereitung

Vor einem Upgrade wird ein Backup der Datenbank und einiger Dateien empfohlen. Die Schritte dazu finden sich [hier](#).

Upgrade

Dieses Script erstellt Backups, löscht das BookStack Verzeichnis, klonst die aktuelle Version und stellt die benötigten Dateien und deren Berechtigungen wieder her.

```
cat > /root/upgrade-boockstak-on-debian-13.sh << EOF

echo "THIS SCRIPT IS NOT CONSIDERED OFFICIALLY SUPPORTED!"

echo "This upgrade a BookStack instance on a allready running Debian 13 (Trixie) server."
echo "This script does not ensure system security."
echo "This script is a modification from the install script found at"
echo "https://codeberg.org/bookstack/devops/src/branch/main/scripts/installation-debian-13.sh"

# Get date & Time
SCRIPTTIME=$(date +%s)

# Generate a path for a log file to output into for debugging
LOGPATH=$(realpath "bookstack_install_$SCRIPTTIME.log")

# The directory to install BookStack into
BOOKSTACK_DIR="/var/www/bookstack"

# Prevent interactive prompts in applications
export DEBIAN_FRONTEND=noninteractive
```

```

# Echo out an error message to the command line and exit the program
# Also logs the message to the log file
function error_out() {
    echo "ERROR: $1" | tee -a "$LOGPATH" 1>&2
    exit 1
}

# Echo out an information message to both the command line and log file
function info_msg() {
    echo "$1" | tee -a "$LOGPATH"
}

# Run some checks before installation
function run_pre_install_checks() {
    # Check we're running as root and exit if not
    if [[ $EUID -gt 0 ]]
    then
        error_out "This script must be ran with root/sudo privileges"
    fi
}

# Backup of database & files
function run_bookstack_backups () {
    mysqldump -u root bookstack > bookstack-database-backup-$SCRIPTTIME.sql || exit
    cd /var/www/bookstack || exit
    tar -czvf /root/bookstack-files-backup-$SCRIPTTIME.tar.gz .env public/uploads
storage/uploads themes || exit
}

# Remove old BookStack directory
function run_bookstack_removal() {
    cd /var/www || exit
    rm bookstack -R
}

# Download BookStack
function run_bookstack_download() {
    cd /var/www || exit
    git clone https://github.com/BookStackApp/BookStack.git --branch release --single-branch
bookstack

```

```

}

# Install BookStack composer dependencies
function run_download_bookstack_vendor_files() {
    cd "$BOOKSTACK_DIR" || exit
    php bookstack-system-cli download-vendor
}

# Restore BooStack files from backup
function run_restore_files () {
    cd "$BOOKSTACK_DIR" || exit
    tar -xvzf /root/bookstack-files-backup-$SCRIPTTIME.tar.gz
}

# Set file and folder permissions
# Sets current user as owner user and www-data as owner group then
# provides group write access only to required directories.
# Hides the `.env` file so it's not visible to other users on the system.
function run_set_application_file_permissions() {
    cd "$BOOKSTACK_DIR" || exit
    chown -R www-data:www-data ./
    chmod -R 755 ./
    chmod -R 775 bootstrap/cache public/uploads storage
    chmod 640 .env

    # Tell git to ignore permission changes
    git config core.fileMode false
}

info_msg "This script logs full output to $LOGPATH which may help upon issues."
sleep 1

run_pre_install_checks
#run_prompt_for_domain_if_required
#info_msg ""
#info_msg "Installing using the domain or IP \"${DOMAIN}\""
#info_msg ""
#sleep 1
info_msg "[1/6] Backup BookStack database and files from ${BOOKSTACK_DIR}..."
run_bookstack_backups >> "$LOGPATH" 2>&1

```

```
info_msg "[2/6] Remove BookStack from ${BOOKSTACK_DIR}..."
run_bookstack_removal >> "$LOGPATH" 2>&1
```

```
info_msg "[3/6] Downloading BookStack to ${BOOKSTACK_DIR}..."
run_bookstack_download >> "$LOGPATH" 2>&1
```

```
info_msg "[4/6] Downloading PHP dependency files..."
run_download_bookstack_vendor_files >> "$LOGPATH" 2>&1
```

```
info_msg "[5/6] Restore files from backup..."
run_restore_files >> "$LOGPATH" 2>&1
```

```
info_msg "[6/6] Setting BookStack file & folder permissions..."
run_set_application_file_permissions >> "$LOGPATH" 2>&1
```

```
info_msg "-----"
info_msg "Setup finished, your BookStack instance should now be upgraded!"
info_msg "-----"
```

EOF