

15: BookStack - Documentation Platform / Wiki

BookStack is a self-hosted documentation and wiki platform built with PHP and Laravel. It provides a hierarchical content structure (Books → Chapters → Pages), WYSIWYG editing with Markdown support, full-text search, user roles, and export capabilities. For API documentation, advanced customization, and development guides, see the [official BookStack documentation](#).

Prerequisites

- **Docker installed** (Chapter 3)
- **Docker Compose** (Chapter 3)
- **Optional: Traefik installed** (Chapter 4) for HTTPS with Let's Encrypt
- **Optional: Domain configured** (Chapter 4.5), e.g., `docs.example.com`

Installation via Infinity Tools

Menu Installation

```
☐☐APPLICATIONS → BookStack → Install
```

CLI Installation

```
sudo bash /opt/InfinityTools/Solutions/setup-bookstack.sh --install

# With domain (Traefik mode)
export BS_DOMAIN="docs.example.com"
export BS_EMAIL="admin@example.com"
sudo -E bash /opt/InfinityTools/Solutions/setup-bookstack.sh --install

# Standalone mode with custom port
export BS_USE_TRAEFIK=false
```

```
export BS_PORT=8092
sudo -E bash /opt/InfinityTools/Solutions/setup-bookstack.sh --install

# Fresh install (delete all data)
sudo bash /opt/InfinityTools/Solutions/setup-bookstack.sh --install --deleteall
```

Deployment Modes

Traefik Mode (Default)

Uses Traefik for SSL termination and domain routing:

- Automatic Let's Encrypt certificate provisioning
- Domain-based access: `https://docs.example.com`
- HTTP → HTTPS redirect configured
- Security headers (X-Frame-Options, CSP, etc.)
- Requires: Traefik running, DNS A record configured

Standalone Mode

Direct access with optional HTTPS (self-signed):

- HTTP: `http://SERVER_IP:8092`
- HTTPS: `https://SERVER_IP:8092` (self-signed cert via nginx proxy)
- Default port: 8092 (configurable)
- No domain required

Architecture

Containers

- **bookstack** - Main BookStack application (lscr.io/linuxserver/bookstack:latest)
- **bookstack-db** - MariaDB 10.11 database
- **bookstack-ssl-proxy** - Nginx SSL proxy (standalone HTTPS mode only)

Data Persistence

- **Database:** `/opt/speedbits/bookstack/db_data/` (MariaDB data)

- **Config:** `/opt/speedbits/bookstack/config/` (BookStack config, uploads)
- **SSL:** `/opt/speedbits/bookstack/ssl/` (standalone mode certificates)

Networks

- **Traefik network:** Joins Traefik's proxy network (Traefik mode)
- **bookstack-internal:** Isolated bridge network (standalone mode)
- **borgmatic-db:** Database backup network (for Borgmatic integration)

Installation Process

Configuration Steps

1. **SSL Mode Selection:** Choose Traefik (default) or Standalone
2. **If Traefik:** Provide domain name and email
3. **If Standalone:** Specify port (default: 8092) and SSL mode
4. **Database Setup:** Random passwords generated and saved
5. **App Key:** Encryption key generated for BookStack

What Gets Created

- **Directory:** `/opt/speedbits/bookstack`
- **Containers:** `bookstack`, `bookstack-db`
- **Docker Compose:** `/opt/speedbits/bookstack/docker-compose.yml`
- **Password Files:** `db_password.txt`, `db_root_password.txt`, `app_key.txt`
- **Volumes:** Database data, config, SSL certificates

Database Configuration

Database Details

- **Type:** MariaDB 10.11
- **Database:** bookstack
- **User:** bookstack
- **Password:** Randomly generated, saved in `db_password.txt`
- **Root Password:** Saved in `db_root_password.txt`
- **Character Set:** utf8mb4
- **Collation:** utf8mb4_unicode_ci
- **Max Packet Size:** 256MB

Accessing Database

```
# View database password
cat /opt/speedbits/bookstack/db_password.txt

# Connect to database
docker exec -it bookstack-db mysql -u bookstack -p bookstack
# Enter password from db_password.txt

# Backup database
docker exec bookstack-db mysqldump -u bookstack -p bookstack > backup.sql
```

Access Methods

Traefik Mode

```
https://docs.example.com
```

Direct web access after DNS propagation and SSL certificate generation (30-60 seconds).

Standalone Mode

HTTP:

```
http://SERVER_IP:8092
```

HTTPS:

```
https://SERVER_IP:8092
```

Accept self-signed certificate warning (Advanced → Proceed).

Authentication

Default Credentials

- **Email:** `admin@admin.com`

- **Password:** `password`

⚠ **CRITICAL:** Change these immediately after first login! These are public defaults.

User Roles

- **Admin** - Full system access, user management, settings
- **Editor** - Can create/edit content, manage own content
- **Viewer** - Read-only access
- **Guest** - Public access (if enabled)

Environment Variables

BookStack Container

- `APP_URL` - Base URL (`https://docs.example.com` or `http://SERVER_IP:8092`)
- `APP_KEY` - Encryption key (base64 encoded, auto-generated)
- `DB_HOST` - Database host (`bookstack-db:3306`)
- `DB_DATABASE` - Database name (`bookstack`)
- `DB_USERNAME` - Database user (`bookstack`)
- `DB_PASSWORD` - Database password (from `db_password.txt`)
- `MAIL_DRIVER` - Email driver (`smtp`)
- `MAIL_HOST` - SMTP server
- `MAIL_PORT` - SMTP port
- `MAIL_FROM` - From email address

Key Features

Content Management

- Hierarchical structure: Books → Chapters → Pages
- WYSIWYG editor with Markdown support
- Full-text search across all content
- Image uploads and attachments
- Version history and revisions
- Tags and categorization

User Management

- Role-based access control (RBAC)
- User invitations via email
- LDAP/Active Directory integration (advanced)
- OAuth providers (GitHub, Google, etc.)

Export & Integration

- Export to PDF, HTML, Markdown, Plain Text
- REST API for programmatic access
- Webhook support
- RSS feeds

Security Configuration

Access Security

- Traefik mode uses Let's Encrypt SSL (production-ready)
- Standalone HTTPS uses self-signed certificates (acceptable for internal use)
- Security headers configured (X-Frame-Options, CSP, etc.)
- Password hashing (bcrypt)
- [△](#) Change default admin credentials immediately

Container Security

- Runs as non-root user (PUID/PGID: 1000)
- Database isolated in separate container
- Network isolation (internal networks)
- Volume mounts for data persistence

Configuration Persistence

- **Database Volume:** `db_data` persists all content
- **Config Volume:** `config` persists settings and uploads
- **Password Files:** Saved in `/opt/speedbits/bookstack/`
- All settings survive container restarts

Backup & Restore

Backup Strategy

```
# Full backup
cd /opt/speedbits
tar czf bookstack-backup-$(date +%Y%m%d).tar.gz bookstack/

# Database-only backup
docker exec bookstack-db mysqldump -u bookstack -p bookstack > bookstack-db-$(date +%Y%m%d).sql

# Config-only backup
tar czf bookstack-config-$(date +%Y%m%d).tar.gz -C /opt/speedbits/bookstack config/
```

Restore Process

1. Stop containers: `cd /opt/speedbits/bookstack && docker compose down`
2. Restore data: Extract backup to `/opt/speedbits/bookstack/`
3. Restore database: `docker exec -i bookstack-db mysql -u bookstack -p bookstack < backup.sql`
4. Start containers: `docker compose up -d`

Troubleshooting

Container Not Starting

```
docker logs bookstack
docker logs bookstack-db
docker ps -a | grep bookstack
```

Database Connection Issues

- Verify database is running: `docker ps | grep bookstack-db`
- Check database logs: `docker logs bookstack-db`
- Test connection: `docker exec bookstack-db mysqladmin ping -h localhost`
- Verify password: `cat /opt/speedbits/bookstack/db_password.txt`

SSL Certificate Issues

- **Traefik mode:** Check Traefik logs: `docker logs traefik`
- **Traefik mode:** Verify DNS: `dig docs.example.com`
- **Standalone mode:** Check nginx proxy logs: `docker logs bookstack-ssl-proxy`
- **Standalone mode:** Verify certificates: `ls -la /opt/speedbits/bookstack/ssl/`

Performance Issues

- Check container resources: `docker stats bookstack bookstack-db`
- Review database performance: `docker exec bookstack-db mysqladmin processlist`
- Optimize images before upload
- Consider database indexing for large content

Production Considerations

- **Access Method:** Use Traefik mode for production (trusted SSL)
- **Password Policy:** Enforce strong passwords via user management
- **Backup Strategy:** Implement automated backups (integrate with Borgmatic)
- **User Management:** Use LDAP/OAuth for enterprise environments
- **Monitoring:** Monitor container health and resource usage
- **Updates:** Re-run install script periodically for updates
- **Email:** Configure SMTP for password resets and notifications

Integration with Infinity Tools

BookStack complements Infinity Tools by providing:

- Documentation platform for Infinity Tools guides
- Knowledge base for server configurations
- Team collaboration on documentation
- Export capabilities for offline access

Note: The Infinity Tools documentation sync script (`sync-bookstack.sh`) can automatically upload HTML documentation files to BookStack.

SMTP Configuration

After installation, you can configure SMTP for email functionality:

- Run SMTP config wizard: `sudo bash Infrastructure/bookstack-smtp-config.sh`
- Or edit `docker-compose.yml` and set `MAIL_HOST`, `MAIL_PORT`, etc.

- Restart container: `docker restart bookstack`

Next Steps

BookStack is now operational. Use it to:

- Create documentation for your infrastructure
- Organize knowledge in books and chapters
- Collaborate with team members
- Export content for offline access
- Integrate with other tools via API

For advanced features, API usage, custom themes, and development guides, refer to the [official BookStack documentation](#).

Revision #1

Created 17 November 2025 16:39:27 by bjoern

Updated 17 November 2025 16:39:46 by bjoern