LAMP Stack Projektdokumentation

Inhalt

[1. Projektübersicht 2](#_Toc187044962)

[1.1 Zielsetzung 2](#_Toc187044963)

[1.2 Funktionsumfang 2](#_Toc187044964)

[2. Systemarchitektur 2](#_Toc187044965)

[2.1 Komponenten 2](#_Toc187044966)

[2.2 Verzeichnisstruktur 3](#_Toc187044967)

[3. Datenbank-Design 3](#_Toc187044968)

[3.1 Datenbankschema 3](#_Toc187044969)

[3.2 Datenbankbenutzer 4](#_Toc187044970)

[4. Sicherheitsarchitektur 4](#_Toc187044971)

[4.1 Apache-Sicherheit 4](#_Toc187044972)

[4.2 PHP-Sicherheit 5](#_Toc187044973)

[4.3 Backup-System 5](#_Toc187044974)

[4.4 SSH-Konfiguration 5](#_Toc187044975)

[4.5 Firewall (UFW) 5](#_Toc187044976)

[5. Systemüberwachung 6](#_Toc187044977)

[5.1 Automatische Überwachung 6](#_Toc187044978)

[5.2 Protokollierung 6](#_Toc187044979)

[6. Installation und Wartung 6](#_Toc187044980)

[6.1 Erstinstallation 6](#_Toc187044981)

[6.2 Wartungsaufgaben 6](#_Toc187044982)

[6.3 Notfallwiederherstellung 7](#_Toc187044983)

[7. Fehlerbehebung 7](#_Toc187044984)

[7.1 Häufige Probleme 7](#_Toc187044985)

[7.2 Debugging-Werkzeuge 7](#_Toc187044986)

[8. Weiterentwicklung 7](#_Toc187044987)

[8.1 Mögliche Erweiterungen 7](#_Toc187044988)

[8.2 Bekannte Einschränkungen 8](#_Toc187044989)

## 1. Projektübersicht

### 1.1 Zielsetzung

Dieses Projekt implementiert eine vollständige LAMP-Stack Umgebung (Linux, Apache, MySQL, PHP) mit erweiterten Sicherheitsfunktionen und automatisierter Systemüberwachung. Die Lösung ist speziell für Raspberry Pi optimiert, kann aber auf anderen Linux-Systemen eingesetzt werden.

### 1.2 Funktionsumfang

* Vollautomatisierte Installation und Konfiguration
* Mehrbenutzersystem mit Dateiupload-Funktion
* Umfassendes Backup- und Wiederherstellungssystem
* Erweiterte Sicherheitsmaßnahmen
* Automatisierte Systemüberwachung
* Webbasierte Datenbankadministration

## 2. Systemarchitektur

### 2.1 Komponenten

#### Linux

* Basis-Betriebssystem: Raspberry Pi OS oder kompatibles Linux
* Kernel-Version: 5.x oder höher empfohlen
* Mindestanforderungen:
  + CPU: 1 GHz
  + RAM: 2 GB
  + Speicherplatz: 8 GB

#### Apache2

* Aktuellste Version
* MPM: prefork (für PHP-Kompatibilität)
* Aktivierte Module:
  + mod\_rewrite (URL-Umschreibung)
  + mod\_headers (Sicherheitsheader)
  + mod\_php (PHP-Integration)

#### MariaDB

* Aktuellste Version
* Storage Engine: InnoDB
* Charaktersatz: UTF-8MB4

#### PHP

* Version: 8.4 oder höher
* Wichtige Erweiterungen:
  + mysql (Datenbankverbindung)
  + json (Datenaustausch)
  + mbstring (Unicode-Unterstützung)
  + zip (Dateiverarbeitung)

### 2.2 Verzeichnisstruktur

/var/www/html/ # Webroot  
├── index.php # Haupteinstiegspunkt  
├── .htaccess # Apache-Konfiguration  
├── css/ # Stylesheets  
├── js/ # JavaScript-Dateien  
├── img/ # Bilder  
└── inc/ # PHP-Includes  
  
/etc/  
├── apache2/ # Apache-Konfiguration  
├── php/ # PHP-Konfiguration  
└── mysql/ # MariaDB-Konfiguration  
  
/root/backups/ # Backup-Verzeichnis

## 3. Datenbank-Design

### 3.1 Datenbankschema

#### Users-Tabelle

CREATE TABLE `users` (  
 `id` int(10) UNSIGNED NOT NULL AUTO\_INCREMENT,  
 `nachname` varchar(255) NOT NULL,  
 `vorname` varchar(255) NOT NULL,  
 `email` varchar(255) NOT NULL,  
 `passwort` varchar(255) NOT NULL,  
 PRIMARY KEY (`id`),  
 UNIQUE KEY `email` (`email`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

Erklärung der Felder: - id: Eindeutige Benutzer-ID - nachname: Familienname des Benutzers - vorname: Vorname des Benutzers - email: E-Mail-Adresse (dient als Benutzername) - passwort: Gehashtes Passwort

#### Files-Tabelle

CREATE TABLE `files` (  
 `id` int(10) UNSIGNED NOT NULL AUTO\_INCREMENT,  
 `user\_id` int(10) UNSIGNED NOT NULL,  
 `file\_name` varchar(255) NOT NULL,  
 `file\_path` varchar(255) NOT NULL,  
 `uploaded\_at` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP,  
 PRIMARY KEY (`id`),  
 FOREIGN KEY (`user\_id`) REFERENCES `users`(`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

Erklärung der Felder: - id: Eindeutige Datei-ID - user\_id: Referenz zum Benutzer (Fremdschlüssel) - file\_name: Originaler Dateiname - file\_path: Speicherort im Dateisystem - uploaded\_at: Zeitstempel des Uploads

### 3.2 Datenbankbenutzer

1. Root-Benutzer
   * Voller Zugriff auf alle Datenbanken
   * Nur für Administration
   * Passwort: “c” (nach Installation ändern!)
2. Pi-Benutzer
   * Voller Zugriff auf php\_projekt-Datenbank
   * Für Anwendungszugriffe
   * Passwort: “c” (nach Installation ändern!)

## 4. Sicherheitsarchitektur

### 4.1 Apache-Sicherheit

#### .htaccess-Konfiguration

Options -Indexes  
php\_flag display\_errors off  
  
<FilesMatch "\.inc$">  
 Require all denied  
</FilesMatch>  
  
<IfModule mod\_headers.c>  
 Header set X-Content-Type-Options "nosniff"  
 Header set X-Frame-Options "SAMEORIGIN"  
 Header set X-XSS-Protection "1; mode=block"  
</IfModule>

Sicherheitsmaßnahmen: - Verzeichnislisting deaktiviert - PHP-Fehlermeldungen versteckt - Zugriff auf .inc-Dateien blockiert - Security-Header aktiviert - Cross-Site-Scripting-Schutz

### 4.2 PHP-Sicherheit

Wichtige php.ini-Einstellungen:

display\_errors = Off  
log\_errors = On  
error\_reporting = E\_ALL & ~E\_DEPRECATED & ~E\_STRICT  
session.use\_strict\_mode = 1  
session.cookie\_httponly = 1  
session.cookie\_samesite = Strict

### 4.3 Backup-System

#### Backup-Skript (backup-lamp)

* Tägliche Ausführung um 3:00 Uhr
* Inkrementelle Backups
* 7-Tage-Rotation
* Komponenten:
  1. Datenbank-Dump
  2. Konfigurationsdateien
  3. Webroot-Verzeichnis
  4. Komprimierung als tar.gz

#### Wiederherstellung (restore-lamp)

restore-lamp [backup.tar.gz]

Prozess: 1. Dienste stoppen 2. Daten extrahieren 3. Konfigurationen wiederherstellen 4. Datenbank importieren 5. Dienste neustarten

### 4.4 SSH-Konfiguration

Port 22222  
PermitRootLogin no  
PasswordAuthentication no  
PubkeyAuthentication yes  
Protocol 2

Sicherheitsmerkmale: - Nicht-Standard-Port - Nur Public-Key-Authentifizierung - Root-Login deaktiviert - SSHv2-Protokoll erzwungen

### 4.5 Firewall (UFW)

Regelsatz:

ufw default deny incoming  
ufw default allow outgoing  
ufw allow 80/tcp # HTTP  
ufw allow 443/tcp # HTTPS  
ufw allow 22222/tcp # SSH

## 5. Systemüberwachung

### 5.1 Automatische Überwachung

Skript: check-system Ausführung: Alle 5 Minuten via Cron Überwachte Metriken: - Dienststatus (Apache, MySQL, SSH) - Speicherauslastung - Festplattennutzung - Aktive Netzwerkverbindungen - Firewall-Status

### 5.2 Protokollierung

Wichtige Logdateien:

/var/log/apache2/error.log # Apache-Fehler  
/var/log/apache2/access.log # Apache-Zugriffe  
/var/log/php\_errors.log # PHP-Fehler  
/var/log/system-status.log # Systemstatus  
/var/log/mysql/error.log # MySQL-Fehler

## 6. Installation und Wartung

### 6.1 Erstinstallation

1. Repository klonen:

* git clone [repository-url]

1. Script ausführen:

* sudo /Gruppe-C/setup.sh

1. Nach Installation:
   * Passwörter ändern
   * SSH-Schlüssel sichern
   * Firewall-Status prüfen

### 6.2 Wartungsaufgaben

#### Tägliche Aufgaben:

* Protokolle prüfen
* Backup-Status verifizieren
* Systemstatus kontrollieren

#### Wöchentliche Aufgaben:

* Systemaktualisierungen durchführen
* Backup-Rotation prüfen
* Festplattenplatz kontrollieren

#### Monatliche Aufgaben:

* Sicherheitsaudits durchführen
* Benutzerkonten überprüfen
* Datenbankoptimierung

### 6.3 Notfallwiederherstellung

1. Letztes Backup identifizieren
2. System stoppen:
3. systemctl stop apache2 mysql
4. Backup wiederherstellen:
5. restore-lamp /root/backups/[datum].tar.gz
6. System testen:
7. check-system

## 7. Fehlerbehebung

### 7.1 Eventuelle Probleme

1. Apache startet nicht:

* systemctl status apache2  
  journalctl -xe  
  systemctl restart apache2

1. Datenbank-Verbindungsfehler:

* systemctl status mysql  
  tail -f /var/log/mysql/error.log  
  systemctl restart mysql

1. PHP-Fehler:

* tail -f /var/log/php\_errors.log

### 7.2 Debugging-Werkzeuge

* htop: Prozessüberwachung
* iftop: Netzwerküberwachung
* mysqlshow: Datenbankstruktur anzeigen
* apache2ctl -S: Apache-Konfiguration prüfen

## 8. Weiterentwicklung

### 8.1 Mögliche Erweiterungen

1. HTTPS-Unterstützung mit Let’s Encrypt
2. Erweiterte Benutzerrollen
3. Datei-Versionierung
4. Erweiterte Monitoring-Funktionen

### 8.2 Bekannte Einschränkungen

1. Keine HTTPS-Konfiguration
2. Einfaches Benutzermodell
3. Lokale Backups
4. Grundlegende Überwachung