LEMP Stack Management System

Complete set of bash scripts for managing Linux + Nginx + MySQL/MariaDB + PHP-FPM server stack.

📋 Table of Contents

✨ Features

- **Complete LEMP Stack Installation** - Automated setup of Nginx, MySQL/MariaDB, PHP-FPM
- **phpMyAdmin in Docker** - Isolated phpMyAdmin container
- **User Management** - Create and manage system users with web directories
- **Disk Quota** - Set and monitor disk usage limits
- **Virtual Hosts** - Easy Nginx virtual host configuration
- **SSL Certificates** - Let's Encrypt SSL with auto-renewal
- **DNS Management** - BIND9 DNS server configuration
- **FTP Server** - vsftpd with user isolation
- **MySQL Management** - Database, user, and permission management
- **Interactive Menu** - User-friendly TUI for all operations

🔧 Requirements

- **OS**: Ubuntu 20.04+, Debian 10+, CentOS 8+, Rocky Linux 8+, AlmaLinux 8+
- **Access**: Root or sudo privileges
- **Resources**: 2GB RAM minimum, 10GB disk space
- **Network**: Internet connection for package installation

📦 Installation

1. Download and Extract

```bash
# Download the ZIP file
wget https://your-download-link/lemp-scripts.zip

# Extract
unzip lemp-scripts.zip
cd lemp-scripts

# Make all scripts executable
chmod +x *.sh
```

2. Initial Setup

```bash
# Run the master menu
sudo ./lemp-manager.sh

# Or setup LEMP stack directly
sudo ./setup-lemp.sh
```

📜 Scripts Overview

Core Scripts

Script	Purpose
`setup-lemp.sh`	Install complete LEMP stack
`setup-phpmyadmin.sh`	Deploy phpMyAdmin in Docker
`manage-user.sh`	System user management
`manage-quota.sh`	Disk quota management
`manage-vhost.sh`	Nginx virtual host management
`manage-ssl.sh`	Let's Encrypt SSL certificates
`manage-dns.sh`	BIND9 DNS zone management
`manage-ftp.sh`	FTP user and server management
`manage-mysql-db.sh`	MySQL database operations
`manage-mysql-user.sh`	MySQL user management
`manage-mysql-permissions.sh`	MySQL permission control
`lemp-manager.sh`	Interactive master menu

🚀 Quick Start Guide

Setting Up a New Website

```bash
# 1. Install LEMP stack (first time only)
sudo ./setup-lemp.sh

# 2. Create a system user
sudo ./manage-user.sh create john

# 3. Create virtual host
sudo ./manage-vhost.sh create example.com john

# 4. Install SSL certificate
sudo ./manage-ssl.sh install example.com

# 5. Create MySQL database
sudo ./manage-mysql-db.sh create example_db

# 6. Create MySQL user
sudo ./manage-mysql-user.sh create john_db password123

# 7. Grant permissions
sudo ./manage-mysql-permissions.sh grant john_db example_db
```

Using the Master Menu

```bash
sudo ./lemp-manager.sh
```

The interactive menu provides access to all functions with guided prompts.

📖 Detailed Usage

1. LEMP Stack Setup

```bash
# Install complete stack
sudo ./setup-lemp.sh

# What it installs:
# - Nginx web server
# - MariaDB/MySQL database
# - PHP-FPM with extensions
# - Docker for phpMyAdmin
# - Quota tools
# - SSL tools (certbot)
```

2. phpMyAdmin Setup

```bash
# Install phpMyAdmin in Docker
sudo ./setup-phpmyadmin.sh 8080 your_mysql_password

# Access at: http://your-server-ip:8080
# Username: root
# Password: your_mysql_password

# Management commands:
cd /opt/phpmyadmin
docker-compose start
docker-compose stop
docker-compose restart
docker-compose logs -f
```

3. User Management

```bash
# Create user
sudo ./manage-user.sh create username

# Delete user
sudo ./manage-user.sh delete username

# Modify user
sudo ./manage-user.sh modify username

# List all users
sudo ./manage-user.sh list

# Show user info
sudo ./manage-user.sh info username
```

4. Quota Management

```bash
# Setup quota system (first time only)
sudo ./manage-quota.sh setup

# Set quota (soft: 5GB, hard: 6GB)
sudo ./manage-quota.sh set username 5000 6000

# Check quota
sudo ./manage-quota.sh check username

# View quota report
sudo ./manage-quota.sh report

# Remove quota
sudo ./manage-quota.sh remove username
```

5. Virtual Host Management

```bash
# Create virtual host
sudo ./manage-vhost.sh create example.com username

# Delete virtual host
sudo ./manage-vhost.sh delete example.com

# Enable virtual host
sudo ./manage-vhost.sh enable example.com

# Disable virtual host
sudo ./manage-vhost.sh disable example.com

# List all virtual hosts
sudo ./manage-vhost.sh list

# Show virtual host info
sudo ./manage-vhost.sh info example.com
```

6. SSL Certificate Management

```bash
# Install SSL certificate
sudo ./manage-ssl.sh install example.com

# Renew specific certificate
sudo ./manage-ssl.sh renew example.com

# Renew all certificates
sudo ./manage-ssl.sh renew

# Remove certificate
sudo ./manage-ssl.sh remove example.com

# List all certificates
sudo ./manage-ssl.sh list

# Show certificate info
sudo ./manage-ssl.sh info example.com

# Setup auto-renewal
sudo ./manage-ssl.sh auto-renew
```

7. DNS Management

```bash
# Setup BIND9 (first time only)
sudo ./manage-dns.sh setup

# Add DNS zone
sudo ./manage-dns.sh add-zone example.com

# Add DNS record
sudo ./manage-dns.sh add-record example.com A 192.168.1.100
sudo ./manage-dns.sh add-record example.com MX mail.example.com

# Remove DNS record
sudo ./manage-dns.sh remove-record example.com A

# List all zones
sudo ./manage-dns.sh list

# Check zone configuration
sudo ./manage-dns.sh check example.com

# Remove zone
sudo ./manage-dns.sh remove-zone example.com
```

8. FTP Management

```bash
# Setup FTP server (first time only)
sudo ./manage-ftp.sh setup

# Add FTP user
sudo ./manage-ftp.sh add username

# Remove FTP user
sudo ./manage-ftp.sh remove username

# Modify FTP user
sudo ./manage-ftp.sh modify username

# List FTP users
sudo ./manage-ftp.sh list

# Set user quota (5GB)
sudo ./manage-ftp.sh quota username 5000

# Show user info
sudo ./manage-ftp.sh info username
```

9. MySQL Database Management

```bash
# Create database
sudo ./manage-mysql-db.sh create myapp_db

# Delete database
sudo ./manage-mysql-db.sh delete myapp_db

# List databases
sudo ./manage-mysql-db.sh list

# Show database info
sudo ./manage-mysql-db.sh info myapp_db

# Backup database
sudo ./manage-mysql-db.sh backup myapp_db
sudo ./manage-mysql-db.sh backup myapp_db /path/to/backup.sql

# Restore database
sudo ./manage-mysql-db.sh restore myapp_db /path/to/backup.sql.gz

# Import SQL file
sudo ./manage-mysql-db.sh import myapp_db schema.sql
```

10. MySQL User Management

```bash
# Create user
sudo ./manage-mysql-user.sh create dbuser password123

# Delete user
sudo ./manage-mysql-user.sh delete dbuser

# Modify user
sudo ./manage-mysql-user.sh modify dbuser

# List users
sudo ./manage-mysql-user.sh list

# Show user info
sudo ./manage-mysql-user.sh info dbuser

# Change password
sudo ./manage-mysql-user.sh password dbuser newpassword
```

11. MySQL Permission Management

```bash
# Grant permissions (interactive)
sudo ./manage-mysql-permissions.sh grant dbuser myapp_db

# Revoke permissions
sudo ./manage-mysql-permissions.sh revoke dbuser myapp_db

# Show user permissions
sudo ./manage-mysql-permissions.sh show dbuser

# Apply permission template
sudo ./manage-mysql-permissions.sh template dbuser myapp_db readonly
sudo ./manage-mysql-permissions.sh template dbuser myapp_db readwrite
sudo ./manage-mysql-permissions.sh template dbuser myapp_db full
sudo ./manage-mysql-permissions.sh template dbuser myapp_db admin

# Enable remote access
sudo ./manage-mysql-permissions.sh remote dbuser 192.168.1.100
sudo ./manage-mysql-permissions.sh remote dbuser %
```

🔒 Security Notes

Important Security Practices

1. **MySQL Root Password**: Store securely or create `/root/.my.cnf`:

```bash
cat > /root/.my.cnf <<EOF
[client]
password=your_mysql_root_password
EOF
chmod 600 /root/.my.cnf
```

2. **Firewall Configuration**:

```bash
# UFW (Ubuntu/Debian)
ufw allow 22/tcp    # SSH
ufw allow 80/tcp    # HTTP
ufw allow 443/tcp   # HTTPS
ufw allow 21/tcp    # FTP
ufw allow 40000:40100/tcp  # FTP passive
ufw enable

# firewalld (CentOS/Rocky)
firewall-cmd --permanent --add-service=ssh
firewall-cmd --permanent --add-service=http
firewall-cmd --permanent --add-service=https
firewall-cmd --permanent --add-service=ftp
firewall-cmd --reload
```

3. **Regular Updates**:

```bash
# Ubuntu/Debian
apt update && apt upgrade -y

# CentOS/Rocky
yum update -y
```

4. **Backup Strategy**:

- Automated MySQL backups are created before database deletion
- Zone files are backed up before DNS changes
- User data is backed up before deletion
- Store backups in: `/root/mysql_backups/`, `/root/user_backups/`, `/root/dns_backups/`

5. **SSL/TLS**:

- Auto-renewal is configured for Let's Encrypt certificates
- Certificates are checked daily at 2:30 AM
- Renewal logs: `/var/log/certbot-renew.log`

🐛 Troubleshooting

Common Issues

1. Nginx won't start

```bash
# Check configuration
nginx -t

# View error logs
tail -f /var/log/nginx/error.log

# Check if port 80 is in use
netstat -tlnp | grep :80
```

2. MySQL connection issues

```bash
# Check MySQL status
systemctl status mariadb

# View MySQL logs
tail -f /var/log/mysql/error.log

# Test connection
mysql -u root -p
```

3. PHP-FPM not working

```bash
# Check PHP-FPM status
systemctl status php*-fpm

# View PHP-FPM logs
tail -f /var/log/php*-fpm.log

# Restart PHP-FPM
systemctl restart php*-fpm
```

4. SSL certificate issues

```bash
# Test certificate
certbot certificates

# Manual renewal
certbot renew --dry-run
certbot renew --force-renewal

# Check logs
tail -f /var/log/letsencrypt/letsencrypt.log
```

5. Quota not working

```bash
# Check if quota is enabled
mount | grep usrquota

# Recreate quota files
quotaoff -a
quotacheck -augmn
quotaon -a

# Check user quota
quota -vs username
```

6. FTP connection issues

```bash
# Check vsftpd status
systemctl status vsftpd

# View FTP logs
tail -f /var/log/vsftpd.log

# Test FTP
ftp localhost
```

Log Locations

Service	Log Location
Nginx	`/var/log/nginx/`
MySQL	`/var/log/mysql/` or `/var/log/mariadb/`
PHP-FPM	`/var/log/php*-fpm.log`
SSL/Certbot	`/var/log/letsencrypt/`
FTP	`/var/log/vsftpd.log`
DNS	`/var/log/syslog` or `/var/log/messages`

Getting Help

1. Check script usage: `./script-name.sh` (without arguments)
2. Check system logs: `journalctl -xe`
3. Check service status: `systemctl status service-name`
4. Enable debug mode: Add `set -x` at the beginning of any script

Directory Structure

```
/home/username/
├── www/                    # Web files for virtual hosts
│   └── domain.com/
│       └── public/
├── public_html/            # Alternative web directory
├── logs/                   # User-specific logs
│   └── domain.com/
│       ├── access.log
│       └── error.log
└── tmp/                    # Temporary files

/root/
├── mysql_backups/          # MySQL database backups
├── user_backups/           # User data backups
├── dns_backups/            # DNS zone backups
└── vhost_backups/          # Virtual host config backups

/opt/
└── phpmyadmin/             # phpMyAdmin Docker setup
    └── docker-compose.yml

/etc/nginx/
├── sites-available/        # All virtual host configs
├── sites-enabled/          # Enabled virtual host configs
└── snippets/               # Reusable config snippets

/etc/bind/zones/            # DNS zone files
```

🔄 Update & Maintenance

Regular Maintenance Tasks

```bash
# Check system status
sudo ./lemp-manager.sh
# Select option 12 for system status

# Update system packages
apt update && apt upgrade -y  # Ubuntu/Debian
yum update -y                 # CentOS/Rocky

# Check disk usage
df -h

# Check quota usage
sudo ./manage-quota.sh report

# View SSL certificate status
sudo ./manage-ssl.sh list

# Backup all databases
for db in $(mysql -Nse 'SHOW DATABASES' | grep -v -E 'information_schema|performance_schema|mysql|sys'); do
    sudo ./manage-mysql-db.sh backup $db
done
```

📄 License

These scripts are provided as-is for server management purposes. Use at your own risk.

🤝 Support

For issues or questions:

1. Check the troubleshooting section
2. Review log files
3. Verify system requirements
4. Test with minimal configuration

📚 Additional Resources

- Nginx Documentation
- MySQL Documentation
- PHP Documentation
- Let's Encrypt Documentation
- BIND9 Documentation

Version: 1.0
Last Updated: 2025-12
Tested On: Ubuntu 22.04, Debian 11, Rocky Linux 8