Buzz

Setup, and projects that run off of Buzz.deome.net/Buzz.deome.loc

- Plex
  - Moving Plex Library
- Web Server Setup & Applications
  - MariaDB Installation and Configuration
  - Apache Web Server Installation and Configuration
  - BookStack Installation and Configuration on wiki.deome.net
  - Piwigo Installation and Configuration on gallery.deome.net
- Monitoring
  - Monitoring Stack Installation on buzz.deome.loc

Plex

Moving Plex Library

Moving Plex Library to a Different Drive

If you need to move your Plex library to a different drive to free up space, follow these steps. This guide will ensure a smooth transfer without losing your data.

1. Stop the Plex Service

To prevent changes to the database while moving the files, stop the Plex Media Server service:

```
sudo systemctl stop plexmediaserver
```

2. Create the New Library Location

Make sure your new drive is mounted, and create a directory for the Plex library:

```
sudo mkdir -p /new/mount/plex_library
```

Replace `/new/mount/plex_library` with the path to the new location on your drive.

3. Copy the Plex Library

Use `rsync` to copy the library files to the new location:

```
sudo rsync -av --progress /var/lib/plexmediaserver/Library /new/mount/plex_library/
```

- `-a`: Archive mode to preserve permissions and attributes.
- `-v`: Verbose mode to show file names being copied.
- `--progress`: Shows progress during the copy.

4. Backup the Old Library (Optional)

Keep a backup of the original library just in case:

```
sudo mv /var/lib/plexmediaserver/Library /var/lib/plexmediaserver/Library_backup
```

5. Create a Symlink to the New Location

Create a symbolic link from the original library path to the new one:

```
sudo ln -s /new/mount/plex_library/Library /var/lib/plexmediaserver/Library
```

This allows Plex to continue accessing the library as if it were in the original location.

6. Adjust Permissions

Make sure the new directory has the correct permissions for Plex to access it:

```
sudo chown -R plex:plex /new/mount/plex_library/Library
```

7. Start the Plex Service

Restart the Plex Media Server to verify the move:

```
sudo systemctl start plexmediaserver
```

8. Verify the Move

Open Plex in your browser and verify that your media library is intact. If everything works properly, you can delete the old backup to free up space:

```
sudo rm -rf /var/lib/plexmediaserver/Library_backup
```

Summary

- Stop Plex.
- Copy the library to the new drive.
- Create a symlink to the new location.
- Restart Plex and verify.

Using these steps ensures a safe move while keeping Plex operational without reconfiguration. If you encounter issues, feel free to ask for help!

Web Server Setup & Applications

MariaDB Installation and Configuration

MariaDB Installation and Configuration

Overview

This page provides step-by-step instructions for installing MariaDB on *buzz.deome.net* and configuring it to work with applications like BookStack and the photo gallery.

---

1. Installing MariaDB

1. **Update the System**:
   - Ensure all packages are up-to-date.

     ```
     sudo apt update && sudo apt upgrade -y
     ```

2. **Install MariaDB Server**:
   - Use the following command to install MariaDB:

     ```
     sudo apt install mariadb-server -y
     ```

3. **Start and Enable MariaDB**:
   - Start the MariaDB service and enable it to start on boot:

     ```
     sudo systemctl start mariadb
     sudo systemctl enable mariadb
     ```

4. **Verify Installation**:
   - Check the MariaDB version to confirm it is installed:

     ```
     mysql --version
     ```

   - You should see output confirming the MariaDB version, as shown in your initial check.

2. Initial MariaDB Configuration

1. **Run the Security Script**:
   - MariaDB provides a security script to help you configure common security settings.

     ```
     sudo mysql_secure_installation
     ```

   - This script will prompt you to:
     - Set a root password.
     - Remove anonymous users.
     - Disallow remote root login.
     - Remove test databases.
   - Answer each prompt as needed to secure the installation.
2. **Log in as the Root User**:
   - After configuring security, log in to MariaDB:

     ```
     sudo mysql -u root -p
     ```

3. **Create a New Database and User for Each Application**:
   - **Database for BookStack**:

     ```
     CREATE DATABASE bookstackdb;
     CREATE USER 'bookstackuser'@'localhost' IDENTIFIED BY 'your_password';
     GRANT ALL PRIVILEGES ON bookstackdb.* TO 'bookstackuser'@'localhost';
     FLUSH PRIVILEGES;
     ```

   - **Database for Photo Gallery**:

     ```
     CREATE DATABASE gallerydb;
     CREATE USER 'galleryuser'@'localhost' IDENTIFIED BY 'your_password';
     GRANT ALL PRIVILEGES ON gallerydb.* TO 'galleryuser'@'localhost';
     FLUSH PRIVILEGES;
     ```

   - Replace `your_password` with secure passwords.
4. **Exit MariaDB**:
   - Once databases and users are set up, exit MariaDB:

     ```
     EXIT;
     ```

3. Basic MariaDB Configuration Options

1. **Optimize Settings (Optional)**:
   - Open the MariaDB configuration file:

```
sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf
```

- Modify settings under `[mysqld]` as needed. Example optimizations:

```
max_connections = 100
innodb_buffer_pool_size = 256M
```

- Save and close the file.

2. **Restart MariaDB**:
   - Apply changes by restarting the MariaDB service:

```
sudo systemctl restart mariadb
```

---

4. Testing and Verification

1. **Verify Databases and Users**:
   - Log in as root and list the databases to confirm they were created:

```
sudo mysql -u root -p
SHOW DATABASES;
```

2. **Connect Using Each Application's Credentials**:
   - Use the created usernames and passwords to confirm access for BookStack and the gallery:

```
mysql -u bookstackuser -p
mysql -u galleryuser -p
```

3. **Connection from Applications**:
   - Ensure that *BookStack* and the photo gallery application can connect to MariaDB using the specified database names and credentials.

---

5. Maintenance and Backup

1. **Automate Backups**:
   - Create a cron job to back up each database:

```
mysqldump -u root -p bookstackdb > /path/to/backup/bookstackdb.sql
mysqldump -u root -p gallerydb > /path/to/backup/gallerydb.sql
```

2. **Update MariaDB**:
   - Periodically check for updates to keep MariaDB secure and optimized:

```
sudo apt update && sudo apt upgrade mariadb-server
```

Apache Web Server Installation and Configuration

Apache Web Server Installation and Configuration

Overview

This page provides instructions for setting up and configuring Apache on *buzz.deome.net*. It includes steps for installing Apache, configuring virtual hosts for multiple sites, securing with SSL certificates, and basic performance tuning.

---

1. Apache Installation

1. **Update System Packages**:
   - Start by updating the package list and upgrading existing packages:

     ```
     sudo apt update && sudo apt upgrade -y
     ```

2. **Install Apache**:
   - Install Apache with the following command:

     ```
     sudo apt install apache2 -y
     ```

3. **Start and Enable Apache**:
   - Start the Apache service and enable it to run on boot:

     ```
     sudo systemctl start apache2
     sudo systemctl enable apache2
     ```

4. **Verify Installation**:
   - Check the Apache status to ensure it’s running:

```
sudo systemctl status apache2
```

---

2. Basic Apache Configuration

1. **Main Configuration File**:
   - The main Apache configuration file is located at:

     ```
     /etc/apache2/apache2.conf
     ```

2. **Enable Recommended Modules**:
   - Enable commonly used modules:

     ```
     sudo a2enmod rewrite ssl headers
     ```

   - Restart Apache to apply the changes:

     ```
     sudo systemctl restart apache2
     ```

---

3. Setting Up Virtual Hosts

1. **Create a New Virtual Host**:
   - For each site, create a configuration file in `/etc/apache2/sites-available/`. For example, for *wiki.deome.net*:

     ```
     sudo nano /etc/apache2/sites-available/wiki.deome.net.conf
     ```

   - Add the following configuration:

     ```
     <VirtualHost *:80>
       ServerName wiki.deome.net
       DocumentRoot /var/www/wiki
       <Directory /var/www/wiki>
           AllowOverride All
       </Directory>
       ErrorLog ${APACHE_LOG_DIR}/wiki_error.log
       CustomLog ${APACHE_LOG_DIR}/wiki_access.log combined
     </VirtualHost>
     ```

2. **Enable the Virtual Host**:
   - Use `a2ensite` to enable each virtual host:

     ```
     sudo a2ensite wiki.deome.net.conf
     ```

- Reload Apache to apply changes:

```
sudo systemctl reload apache2
```

3. **Repeat for Additional Sites**:
   - Create similar configuration files for *gallery.deome.net*, *buzz.deome.net*, and any other sites you plan to host.

---

4. SSL Configuration with Let’s Encrypt

1. **Install Certbot**:
   - Install Certbot to automate SSL certificate generation:

```
sudo apt install certbot python3-certbot-apache -y
```

2. **Generate SSL Certificates**:
   - Use Certbot to obtain SSL certificates for each virtual host:

```
sudo certbot --apache -d wiki.deome.net
sudo certbot --apache -d gallery.deome.net
```

   - Certbot will automatically configure SSL for the specified domains.
3. **Set Up Auto-Renewal**:
   - To ensure certificates renew automatically:

```
sudo certbot renew --dry-run
```

---

5. Performance and Optimization

1. **Adjust Apache Settings**:
   - Open the main Apache configuration file:

```
sudo nano /etc/apache2/apache2.conf
```

   - Consider modifying these settings under the `mpm_prefork_module` or `mpm_worker_module`:

```
KeepAlive On
MaxKeepAliveRequests 100
KeepAliveTimeout 5
```

2. **Enable Caching**:
   - Install and enable Apache caching modules if desired:

```
sudo a2enmod cache
sudo a2enmod cache_disk
sudo systemctl restart apache2
```

3. **Optimize Logging**:
   - Adjust log levels or configure centralized logging if needed. Logging configuration is in each virtual host file and `/etc/apache2/apache2.conf`.

---

6. Maintenance and Monitoring

1. **Monitor Apache Status**:
   - Check Apache's status periodically:

```
sudo systemctl status apache2
```

2. **Check Logs**:
   - Access logs for each virtual host in `/var/log/apache2/`, e.g.:

```
sudo tail -f /var/log/apache2/wiki_access.log
sudo tail -f /var/log/apache2/wiki_error.log
```

3. **Restarting Apache**:
   - Restart Apache after any configuration changes:

```
sudo systemctl restart apache2
```

This wiki page provides a thorough guide for Apache installation, virtual host setup, SSL configuration, and performance tuning. Let me know if you'd like any additional details or further customization for your setup!

BookStack Installation and Configuration on wiki.deome.net

Overview

This page provides a complete guide to setting up BookStack on *wiki.deome.net*, including installation, configuration, database setup, and SSL configuration with Let's Encrypt.

---

1. Prerequisites

1. **Web Server**: Apache (refer to the Apache wiki page for installation).
2. **Database**: MariaDB (refer to the MariaDB wiki page for installation and configuration).
3. **PHP**: BookStack requires PHP 7.3 or higher. Install necessary PHP modules.

```
sudo apt install php php-mysql php-xml php-mbstring php-zip php-gd -y
```

---

2. Download and Install BookStack

1. **Create the Document Root Directory**:
   - Create a directory for BookStack:

```
sudo mkdir -p /var/www/wiki
```

2. **Download BookStack**:
   - Navigate to the web root and download the latest version:

```
cd /var/www/wiki
sudo git clone https://github.com/BookStackApp/BookStack.git ./
sudo git checkout release
```

3. **Set File Permissions**:
   - Ensure Apache can read/write to the BookStack files:

```
sudo chown -R www-data:www-data /var/www/wiki
sudo chmod -R 755 /var/www/wiki
```

---

3. Configure the Database

1. **Log in to MariaDB**:

   ```
   sudo mysql -u root -p
   ```

2. **Create Database and User**:
   - Create a database for BookStack:

     ```
     CREATE DATABASE bookstackdb;
     CREATE USER 'bookstackuser'@'localhost' IDENTIFIED BY 'your_password';
     GRANT ALL PRIVILEGES ON bookstackdb.* TO 'bookstackuser'@'localhost';
     FLUSH PRIVILEGES;
     EXIT;
     ```

---

4. Configure Environment Variables

1. **Copy the Environment File**:
   - BookStack uses an `.env` file for configuration. Copy the example:

     ```
     sudo cp .env.example .env
     ```

2. **Edit the .env File**:
   - Open the `.env` file to configure your database and site settings:

     ```
     sudo nano .env
     ```

   - Update the following lines:

     ```
     # Database configuration
     DB_HOST=localhost
     DB_DATABASE=bookstackdb
     DB_USERNAME=bookstackuser
     DB_PASSWORD=your_password

     # Application URL
     APP_URL=https://wiki.deome.net
     ```

3. **Save and Close**: Press `Ctrl + X`, then `Y`, and `Enter` to save changes.

5. Set Up Apache Virtual Host for wiki.deome.net

1. **Create the Virtual Host File**:

```
sudo nano /etc/apache2/sites-available/wiki.deome.net.conf
```

2. **Configure the Virtual Host**:
   - Add the following configuration:

```
<VirtualHost *:80>
  ServerName wiki.deome.net
  DocumentRoot /var/www/wiki/public
  <Directory /var/www/wiki/public>
     AllowOverride All
     Require all granted
  </Directory>
  ErrorLog ${APACHE_LOG_DIR}/wiki_error.log
  CustomLog ${APACHE_LOG_DIR}/wiki_access.log combined
</VirtualHost>
```

3. **Enable the Site and Rewrite Module**:

```
sudo a2ensite wiki.deome.net.conf
sudo a2enmod rewrite
sudo systemctl reload apache2
```

---

6. Set Up SSL with Let’s Encrypt

1. **Install Certbot**:
   - If Certbot is not already installed:

```
sudo apt install certbot python3-certbot-apache -y
```

2. **Generate SSL Certificate**:

```
sudo certbot --apache -d wiki.deome.net
```

3. **Auto-Renewal Setup**:
   - Certbot will auto-renew SSL certificates by default. To verify:

```
sudo certbot renew --dry-run
```

---

7. Complete BookStack Setup

1. **Run the BookStack Installation Commands**:
   - Navigate to the BookStack directory and run the following commands to finish the setup:

```
cd /var/www/wiki
sudo -u www-data php artisan key:generate --force
sudo -u www-data php artisan migrate --force
```

2. **Set Permissions (again, if necessary)**:
   - Ensure permissions are still correct after setup:

```
sudo chown -R www-data:www-data /var/www/wiki
sudo chmod -R 755 /var/www/wiki
```

---

8. Testing and Verification

1. **Access BookStack**:
   - Go to `https://wiki.deome.net` in your web browser.
   - Follow the on-screen instructions to set up the admin account.
2. **Check Logs**:
   - Monitor Apache logs if there are any issues:

```
sudo tail -f /var/log/apache2/wiki_access.log
sudo tail -f /var/log/apache2/wiki_error.log
```

---

9. Ongoing Maintenance

1. **Database Backups**:
   - Set up a cron job to regularly back up the BookStack database:

```
mysqldump -u root -p bookstackdb > /path/to/backup/bookstackdb.sql
```

2. **Updating BookStack**:
   - Periodically pull the latest updates from the BookStack repository:

```
cd /var/www/wiki
sudo -u www-data git pull origin release
sudo -u www-data php artisan migrate --force
```

This setup guide for BookStack covers installation, configuration, and basic maintenance. Let me know if you'd like additional details or specific troubleshooting steps included!

Piwigo Installation and Configuration on gallery.deome.net

Overview

This page provides step-by-step instructions to install and configure Piwigo as a photo gallery on *gallery.deome.net*. This includes setting up the Apache virtual host, database configuration, and SSL with Let's Encrypt.

---

1. Prerequisites

1. **Web Server**: Apache (see Apache wiki page).
2. **Database**: MariaDB (refer to MariaDB setup page).
3. **PHP**: Ensure PHP is installed along with the necessary modules:

```
sudo apt install php php-mysql php-xml php-mbstring php-gd php-curl -y
```

---

2. Download and Install Piwigo

1. **Create the Document Root Directory**:
   - Create a directory for Piwigo:

```
sudo mkdir -p /var/www/gallery
```

2. **Download Piwigo**:
   - Navigate to the gallery directory and download the latest Piwigo release:

```
cd /var/www/gallery
sudo wget https://piwigo.org/download/dlcounter.php?code=latest -O piwigo.zip
sudo unzip piwigo.zip
sudo mv piwigo/* ./
sudo rm -rf piwigo piwigo.zip
```

3. **Set File Permissions**:
   - Make sure Apache has the correct permissions to access and modify the files:

```
sudo chown -R www-data:www-data /var/www/gallery
sudo chmod -R 755 /var/www/gallery
```

---

3. Configure the Database

1. **Log in to MariaDB**:

```
sudo mysql -u root -p
```

2. **Create a Database and User for Piwigo**:

```
CREATE DATABASE piwigodb;
CREATE USER 'piwigouser'@'localhost' IDENTIFIED BY 'your_password';
GRANT ALL PRIVILEGES ON piwigodb.* TO 'piwigouser'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

---

4. Configure Apache Virtual Host for gallery.deome.net

1. **Create the Virtual Host File**:
   - Open a new virtual host configuration file for *gallery.deome.net*:

```
sudo nano /etc/apache2/sites-available/gallery.deome.net.conf
```

2. **Add the Virtual Host Configuration**:
   - Use the following configuration:

```
<VirtualHost *:80>
  ServerName gallery.deome.net
  DocumentRoot /var/www/gallery
  <Directory /var/www/gallery>
     AllowOverride All
     Require all granted
  </Directory>
  ErrorLog ${APACHE_LOG_DIR}/gallery_error.log
  CustomLog ${APACHE_LOG_DIR}/gallery_access.log combined
```

```
</VirtualHost>
```

3. **Enable the Site and Rewrite Module**:
   - Enable the virtual host and rewrite module, then reload Apache:

```
sudo a2ensite gallery.deome.net.conf
sudo a2enmod rewrite
sudo systemctl reload apache2
```

---

5. Set Up SSL with Let’s Encrypt

1. **Install Certbot**:
   - If Certbot is not already installed:

```
sudo apt install certbot python3-certbot-apache -y
```

2. **Generate SSL Certificate**:
   - Use Certbot to obtain an SSL certificate for *gallery.deome.net*:

```
sudo certbot --apache -d gallery.deome.net
```

3. **Enable Auto-Renewal**:
   - Certbot will automatically handle renewal. Test the setup with:

```
sudo certbot renew --dry-run
```

---

6. Complete Piwigo Setup

1. **Access the Piwigo Installation Wizard**:
   - Open a browser and go to `https://gallery.deome.net`. You should see the Piwigo setup wizard.
2. **Follow the Setup Wizard**:
   - Enter the database information:
     - **Database Name**: `piwigodb`
     - **Database User**: `piwigouser`
     - **Database Password**: `your_password`
   - Configure the admin account and complete the setup.

---

7. Testing and Verification

1. **Access Piwigo**:
   - Verify that Piwigo loads correctly at `https://gallery.deome.net`.

2. **Check Logs**:
   - Monitor the Apache logs if any issues arise:

```
sudo tail -f /var/log/apache2/gallery_access.log
sudo tail -f /var/log/apache2/gallery_error.log
```

---

8. Maintenance and Updates

1. **Database Backup**:
   - Set up regular database backups:

```
mysqldump -u root -p piwigodb > /path/to/backup/piwigodb.sql
```

2. **Updating Piwigo**:
   - Periodically check for updates in the Piwigo admin dashboard.

This setup should get Piwigo running smoothly on *gallery.deome.net*. Let me know if you need any specific configurations or additional settings!

Monitoring

Monitoring Stack Installation on buzz.deome.loc

Monitoring Stack Installation on buzz.deome.loc

Overview

This guide details the installation and configuration of a monitoring stack on *buzz.deome.loc* using Node Exporter, Prometheus, and Grafana. This setup will allow for system metrics collection, visualization, and alerting.

---

1. Node Exporter Installation (for System Metrics)

1. **Download and Install Node Exporter**:
   - Navigate to the desired installation directory and download the latest Node Exporter release:

```
cd /usr/local/bin
wget https://github.com/prometheus/node_exporter/releases/download/v1.6.0/node_exporter-1.6.0.linux-amd64.tar.gz
tar xvfz node_exporter-1.6.0.linux-amd64.tar.gz
mv node_exporter-1.6.0.linux-amd64/node_exporter .
rm -rf node_exporter-1.6.0.linux-amd64*
```

2. **Create a Systemd Service for Node Exporter**:
   - Create a new service file:

```
sudo nano /etc/systemd/system/node_exporter.service
```

   - Add the following configuration:

```
[Unit]
Description=Node Exporter
```

```
After=network.target

[Service]
User=node_exporter
ExecStart=/usr/local/bin/node_exporter

[Install]
WantedBy=default.target
```

3. **Start and Enable Node Exporter**:
   - Enable and start Node Exporter:

```
sudo systemctl daemon-reload
sudo systemctl enable node_exporter
sudo systemctl start node_exporter
```

4. **Verify Node Exporter**:
   - Node Exporter should now be running on port `9100`. Check by navigating to `http://buzz.deome.loc:9100/metrics` in a web browser.

---

2. Prometheus Installation and Configuration

1. **Download and Install Prometheus**:
   - Download the latest Prometheus release:

```
cd /usr/local/bin
wget https://github.com/prometheus/prometheus/releases/download/v2.47.0/prometheus-2.47.0.linux-amd64.tar.gz
tar xvfz prometheus-2.47.0.linux-amd64.tar.gz
mv prometheus-2.47.0.linux-amd64/prometheus .
mv prometheus-2.47.0.linux-amd64/promtool .
rm -rf prometheus-2.47.0.linux-amd64*
```

2. **Configure Prometheus**:
   - Create the configuration file:

```
sudo nano /usr/local/bin/prometheus.yml
```

   - Add the following configuration:

```
global:
  scrape_interval: 15s
```

```
scrape_configs:
  - job_name: 'node_exporter'
    static_configs:
      - targets: ['localhost:9100']
```

3. **Create a Systemd Service for Prometheus**:
   - Create a new service file:

     ```
     sudo nano /etc/systemd/system/prometheus.service
     ```

   - Add the following configuration:

     ```
     [Unit]
     Description=Prometheus
     After=network.target

     [Service]
     User=prometheus
     ExecStart=/usr/local/bin/prometheus --config.file=/usr/local/bin/prometheus.yml --
     storage.tsdb.path=/var/lib/prometheus

     [Install]
     WantedBy=default.target
     ```

4. **Start and Enable Prometheus**:
   - Enable and start Prometheus:

     ```
     sudo systemctl daemon-reload
     sudo systemctl enable prometheus
     sudo systemctl start prometheus
     ```

5. **Verify Prometheus**:
   - Access Prometheus at `http://buzz.deome.loc:9090`.

---

3. Grafana Installation and Configuration

1. **Install Grafana**:
   - Add the Grafana repository and install Grafana:

     ```
     sudo apt update
     sudo apt install -y software-properties-common
     sudo add-apt-repository "deb https://packages.grafana.com/oss/deb stable main"
     sudo apt update
     ```

```
sudo apt install grafana -y
```

2. **Start and Enable Grafana**:
   - Enable and start Grafana:

```
sudo systemctl start grafana-server
sudo systemctl enable grafana-server
```

3. **Access Grafana**:
   - Grafana will be accessible at `http://buzz.deome.loc:3000`. Default login is `admin` / `admin` (you will be prompted to change the password upon first login).
4. **Add Prometheus Data Source in Grafana**:
   - In Grafana:
     - Go to **Configuration > Data Sources > Add Data Source**.
     - Select **Prometheus** and set the URL to `http://localhost:9090`.
     - Save and test the data source.

---

4. Setting Up Basic Dashboards and Alerts

1. **Import Node Exporter Dashboard**:
   - Grafana provides pre-made dashboards. For Node Exporter metrics, import dashboard ID `1860` from Grafana's dashboard library:
     - Go to **Create > Import** in Grafana.
     - Enter `1860` and follow the prompts to import the Node Exporter dashboard.
2. **Create Alerts in Prometheus (Optional)**:
   - Alerts can be set up in Prometheus for key metrics (e.g., CPU usage, memory).
   - Edit the Prometheus configuration file to add alert rules, and restart Prometheus for changes to take effect.

---

5. Maintenance and Updates

1. **Update the Stack**:
   - Periodically check for updates to Node Exporter, Prometheus, and Grafana:

```
sudo apt update && sudo apt upgrade -y
```

2. **Data Backup**:
   - Regularly back up Grafana dashboards and Prometheus data as needed.

This setup guide provides a full configuration for a monitoring stack on *buzz.deome.loc* using Node Exporter, Prometheus, and Grafana. Let me know if there are additional steps or customizations you'd like!