BookStack @OL9 (Oracle Linux 9)

Let's deploy BookStack web application to the isolated environment, but with repository server available:

- Solution: BookStack (www, github)
- Resources: VM in the laptop lab.
- Stack:
  - lt58dox1: OL9 + nginx + MariaDB

Create VM. Install Oracle Linux 9, perform recommended post-install.

Preparations:

We need to enable package module to ensure fresh PHP will be installed

```
dnf module list php
```

```
Last metadata expiration check: 0:54:24 ago on Fri 18 Jul 2025 02:09:59 PM EEST.
Oracle Linux 9 Application Stream Packages
Name                Stream              Profiles                                Summary
php                 8.1                 common [d], devel, minimal              PHP scripting
language
php                 8.2                 common [d], devel, minimal              PHP scripting
language
php                 8.3                 common [d], devel, minimal              PHP scripting
language
```

```
Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled
```

Configure to use v8.3 by defining the version of the module

```
dnf module enable php:8.3
```

```
sudo su
dnf install \
    tmux \
    git \
    php \
    php-gd \
    php-zip \
    php-mysqlnd \
```

Install MariaDB and perform recommended post-install

```
dnf install mariadb-server
systemctl enable --now mariadb
systemctl status mariadb
```

Secure the fresh setup

```
mariadb-secure-installation
```

Set up root password and write it down to your password manager.

```
Enter current password for root (enter for none):
Switch to unix_socket authentication [Y/n]: Y
Change the root password? [Y/n] Y
  New password:
  Re-enter new password:
Password updated successfully!
```

```
Remove anonymous users? [Y/n] Y
Disallow root login remotely? [Y/n] Y
Remove test database and access to it? [Y/n] Y
Reload privilege tables now? [Y/n] Y
```

Create database and application user

```
export db_pass="Very-Strong-Password123"

mariadb -u root --execute="CREATE DATABASE bookstack;"
mariadb -u root --execute="CREATE USER 'bookstack'@'localhost' IDENTIFIED BY '${db_pass}';"
mariadb -u root --execute="GRANT ALL ON bookstack.* TO 'bookstack'@'localhost'; FLUSH PRIVILEGES;"
```

Setup nginx, enable SSL (we have isolated environment and will use self-signed certificate)

```
dnf install nginx
systemctl enable --now nginx
systemctl status nginx
ss -ntap | grep nginx
```

```
LISTEN    0      511          0.0.0.0:80         0.0.0.0:*
users:(("nginx",pid=14387,fd=6),("nginx",pid=14386,fd=6))
LISTEN    0      511             [::]:80            [::]:*     users:(("nginx",pid=14387,fd=7),("nginx",pid=14386,fd=7))
```

Create of self-signed certificates and Enable SSL

```
sudo su
mkdir -p  /etc/pki/nginx/private/

# default cert
openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
    -subj "/C=FI/ST=State/L=City/O=Home/OU=IT/CN=$(hostname)" \
    -out    /etc/pki/nginx/server.crt \
    -keyout /etc/pki/nginx/private/server.key

# application-specific cert
export app="dox.$(hostname)"
openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
    -subj "/C=FI/ST=State/L=City/O=Home/OU=IT/CN=${app}" \
    -out    /etc/pki/nginx/${app}.crt \
    -keyout /etc/pki/nginx/private/${app}.key
```

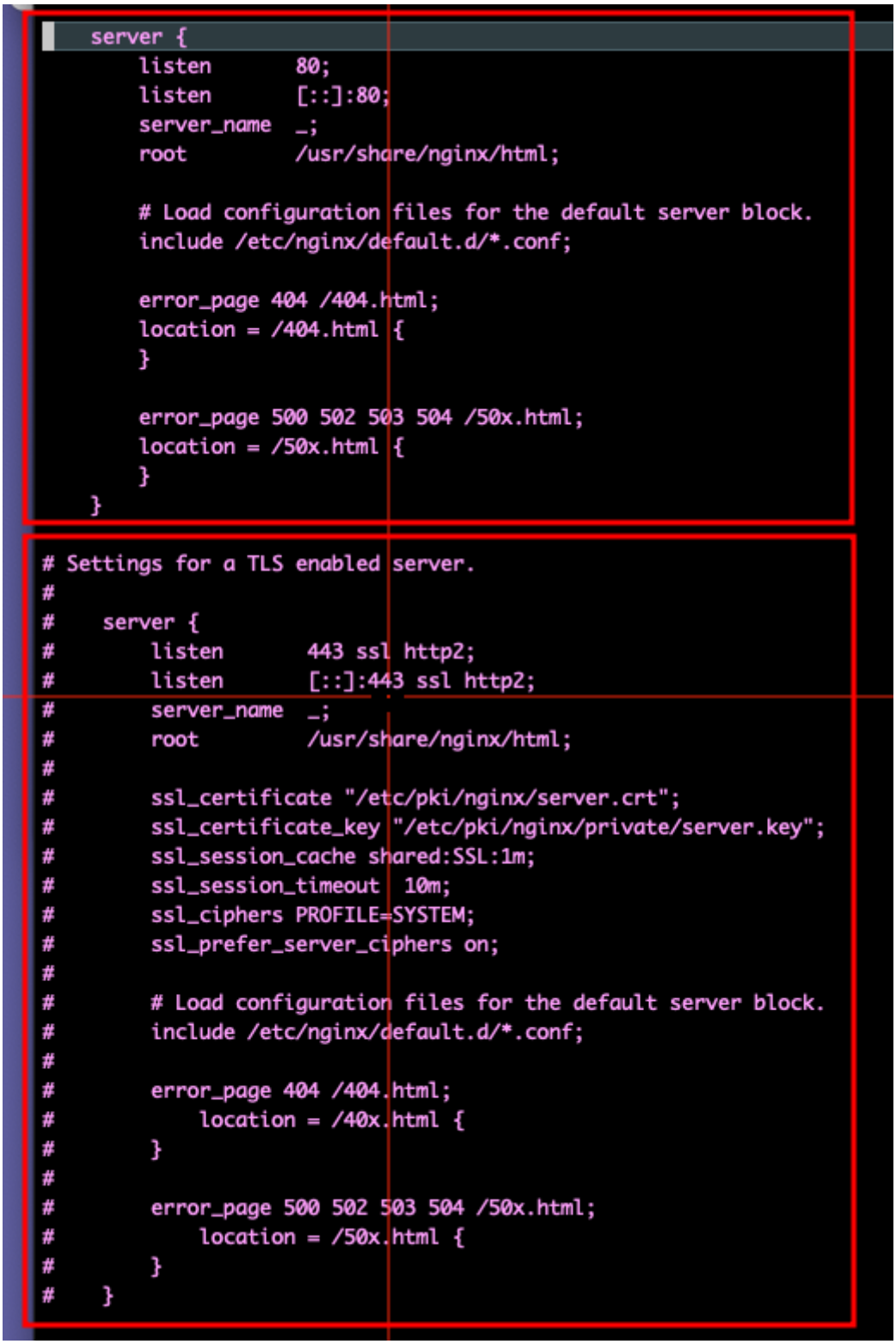
Modify Nginx webserver's config

Open Nginx's default config file

```
vi /etc/nginx/nginx.conf
```

It is highly recommended to stop using and promote usage of unencrypted HTTP.

Disable (=comment or remove) "Server { Listen 80; }" definition (upper block) and enable (=uncomment) "Server { Listen 443 ssl http2; }" definition lower block.



Shortly, config should look like this

```
server {
    listen       443 ssl http2;
    listen       [::]:443 ssl http2;
    server_name  _;
    root         /usr/share/nginx/html;
```

```
        ssl_certificate     "/etc/pki/nginx/server.crt";
        ssl_certificate_key "/etc/pki/nginx/private/server.key";

        ssl_session_cache shared:SSL:1m;
        ssl_session_timeout  10m;
        ssl_ciphers PROFILE=SYSTEM;
        ssl_prefer_server_ciphers on;

        # Load configuration files for the default server block.
        include /etc/nginx/default.d/*.conf;
    }
```

Test config and reload it:

```
nginx -t
nginx -s reload
ss -ntap | grep nginx
```

```
LISTEN 0      511          0.0.0.0:443        0.0.0.0:*
users:(("nginx",pid=14513,fd=11),("nginx",pid=14386,fd=11))
LISTEN 0      511             [::]:443           [::]:*     users:(("nginx",pid=14513,fd=12),("nginx",pid=14386,fd=12))
```

Test it, requesting to show headers only, '-k' - will skip cert checks, as they will fail for not matching the CN (common name). Using local address to avoid DNS lookup.

```
curl -I https://127.0.0.1 -k
```

```
HTTP/2 200
server: nginx/1.20.1
date: Fri, 18 Jul 2025 10:07:36 GMT
content-type: text/html
content-length: 4395
last-modified: Tue, 13 May 2025 20:26:12 GMT
etag: "6823aae4-112b"
accept-ranges: bytes
```

Open firewall (create security policies to pass the traffic in)

```
# It is highly recommended to stop using and promote usage of unencrypted HTTP


firewall-cmd --remove-service=http  --permanent
firewall-cmd    --add-service=https --permanent
firewall-cmd --reload
firewall-cmd --list-all
```

Foundation is done. We can proceed with application setup.

Install PHP-FPM

```
dnf install php-fpm
systemctl enable --now php-fpm
systemctl status php-fpm
fgrep -irn sock /etc/php-fpm.d/ | grep run
```

We need to know, where PHP-FPM is listening, to point requests correctly from webserver below:

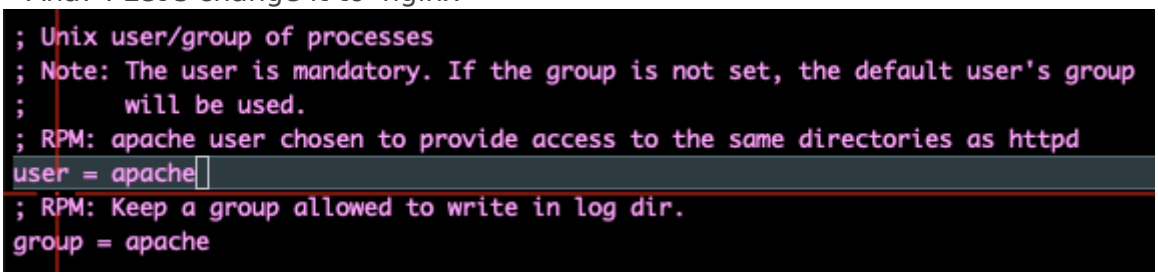
```
/etc/php-fpm.d/www.conf:38:listen = /run/php-fpm/www.sock
```

Due to historical leftovers, we have to change a configuration to let php-fpm access the files properly. If you investigate which username is used during operation:

```
fgrep -irn user /etc/php-fpm.d/ | grep -v \;
```

```
/etc/php-fpm.d/www.conf:24:user = apache
/etc/php-fpm.d/www.conf:55:listen.acl_users = apache,nginx
```

-"Aha!". Let's change it to 'nginx'




```
/etc/php-fpm.d/www.conf
```

Should be like this, using 'nginx' user:

```
; RPM: apache user chosen to provide access to the same directories as httpd
user = nginx
; RPM: Keep a group allowed to write in log dir.
group = nginx
```

Remember to restart the service

```
systemctl restart php-fpm
```

Install Composer and make available globally.

```
# as a normal user, not root
mkdir -p ~/utils/composer
cd ~/utils/composer/
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "if (hash_file('sha384', 'composer-setup.php') ===
'dac665fdc30fdd8ec78b38b9800061b4150413ff2e3b6f88543c636f7cd84f6db9189d43a81e5503cda447da73c7e
5b6') { echo 'Installer verified'; } else { echo 'Installer corrupt'; unlink('composer-setup.php'); } echo PHP_EOL;"
php composer-setup.php
php -r "unlink('composer-setup.php');"
php composer.phar
sudo mv composer.phar /usr/local/bin/composer
```

Prepare home for application

Naming convention: application.host, in my case it is. Always bear in mind, that one host may have more than one application. If there are direct FQDN, then use it to make management easier.

```
whoami
sudo su

# privileged user (you)
export pu="anton"

export host="$(hostname)"
export app="dox.${host}"
export dir="/var/www/${app}"
mkdir -p ${dir}

# permit priviledged user to own app directory
chown -R ${pu}:${pu} ${dir}
exit

# as priviledged user, not as root
export host="$(hostname)"
export app="dox.${host}"
export dir="/var/www/${app}"
cd ${dir}
```

Create application-specific configuration file in the webserver

```
cat << _EOF_ > /etc/nginx/conf.d/${app}.conf
server {
    listen 443 ssl;

    server_name ${app};

    access_log /var/log/nginx/${app}_access.log;
    error_log  /var/log/nginx/${app}_error.log;

    ssl_certificate     "/etc/pki/nginx/${app}.crt";
    ssl_certificate_key "/etc/pki/nginx/private/${app}.key";
```

```
    ssl_protocols TLSv1.2;
    ssl_prefer_server_ciphers on;
    ssl_session_cache shared:SSL:1m;
    ssl_session_timeout 10m;
    ssl_ciphers PROFILE=SYSTEM;

    root ${dir}/BookStack/public;
    index index.php;

    client_max_body_size 1G;
    fastcgi_buffers 64 4K;

    location / {
        try_files \$uri \$uri/ /index.php?\$query_string;
    }

    location ~ \.php(?:$|/) {
        fastcgi_split_path_info ^(.+\.php)(/.+)$;
        include fastcgi_params;
        fastcgi_param SCRIPT_FILENAME \$document_root\$fastcgi_script_name;
        fastcgi_param PATH_INFO \$fastcgi_path_info;
        fastcgi_pass unix:/run/php-fpm/www.sock;
    }

    location ~* \.(?:jpg|jpeg|gif|bmp|ico|png|css|js|swf)$ {
        expires 30d;
        access_log off;
    }

    location ~ ^/(?:\.htaccess|data|config|db_structure\.xml|README) {
        deny all;
    }

}
_EOF_
```

Prepare hostnames to match certificates CNs (for local usage)

```
vi /etc/hosts
```

```
127.0.0.1   localhost lt58dox1 dox.lt58dox1
```

Test and reload the webserver:

```
nginx -t
nginx -s reload
curl https://dox.lt58dox1 -k
curl https://lt58dox1 -k
```

Installation of BookStack application

Change directory before cloning the repo

```
# as a normal user, not root
export host="$(hostname)"
export app="dox.${host}"
export dir="/var/www/${app}/"

cd ${dir}
pwd
```

```
/var/www/dox.lt58dox1
```

Clone repo

```
git clone https://github.com/BookStackApp/BookStack.git --branch release --single-branch
ls -la
```

```
total 8
drwxr-xr-x.  3 root root   41 Jul 18 14:02 .
drwxr-xr-x.  5 root root   53 Jul 18 13:32 ..
drwxr-xr-x. 15 root root 4096 Jul 18 14:02 BookStack
-rw-r--r--.  1 root root   14 Jul 18 13:24 index.html
```

Build an application

```
cd Bookstack
which composer
composer install --no-dev
```

Should be built successfully:

```
[...]
> @php artisan cache:clear
  INFO  Application cache cleared successfully.
> @php artisan view:clear
  INFO  Compiled views cleared successfully.
```

Creating missing directories

issue

```
#3 /var/www/dox.lt58dox1/BookStack/vendor/la...; PHP message: PHP Fatal error:  Uncaught
UnexpectedValueException: The stream or file "/var/www/dox.lt58dox1/BookStack/storage/logs/laravel.log"
could not be opened in append mode: Failed to open stream: Permission denied
```

```
sudo su

export host="$(hostname)"
export app="dox.${host}"
export dir="/var/www/${app}/"

# Create an empty log and directory for views
touch ${dir}/BookStack/storage/logs/laravel.log
```

Set least necessary permissions for application

```
export host="$(hostname)"
export app="dox.${host}"
export dir="/var/www/${app}"

vi fixperm.sh
```

```
#!/bin/bash

export host="$(hostname)"
export app="dox.${host}"
export dir="/var/www/${app}/BookStack/"

# priviledged user
export pu="anton"

# webserver's user
export wsu="nginx"

chown -R ${wsu}:${pu} ${dir}
subdirs=("" "/storage/" "/bootstrap/cache/" "/public/uploads/")
for subdir in "${subdirs[@]}" ; do
    echo "--[ subdir: ${subdir} ]--"
    chown -R ${wsu}:${pu} ${dir}/${subdir}
done
for subdir in "${subdirs[@]}" ; do
    echo "--[ subdir: ${subdir} ]--"
    find ${dir}/${subdir} -type d -exec chmod 0770 {} \+
    find ${dir}/${subdir} -type f -exec chmod 0660 {} \+
done
```

```
chmod +x ./fixperm.sh
./fixperm.sh
```

Configure application

Copy template, generate salt and configure .env

```
sudo su

export host="$(hostname)"
export app="dox.${host}"
export dir="/var/www/${app}"
cd ${dir}/BookStack/
pwd

cp .env.example .env
php artisan key:generate
```

Are you sure you want to run this command? Yes [Enter]

```
vi .env
```

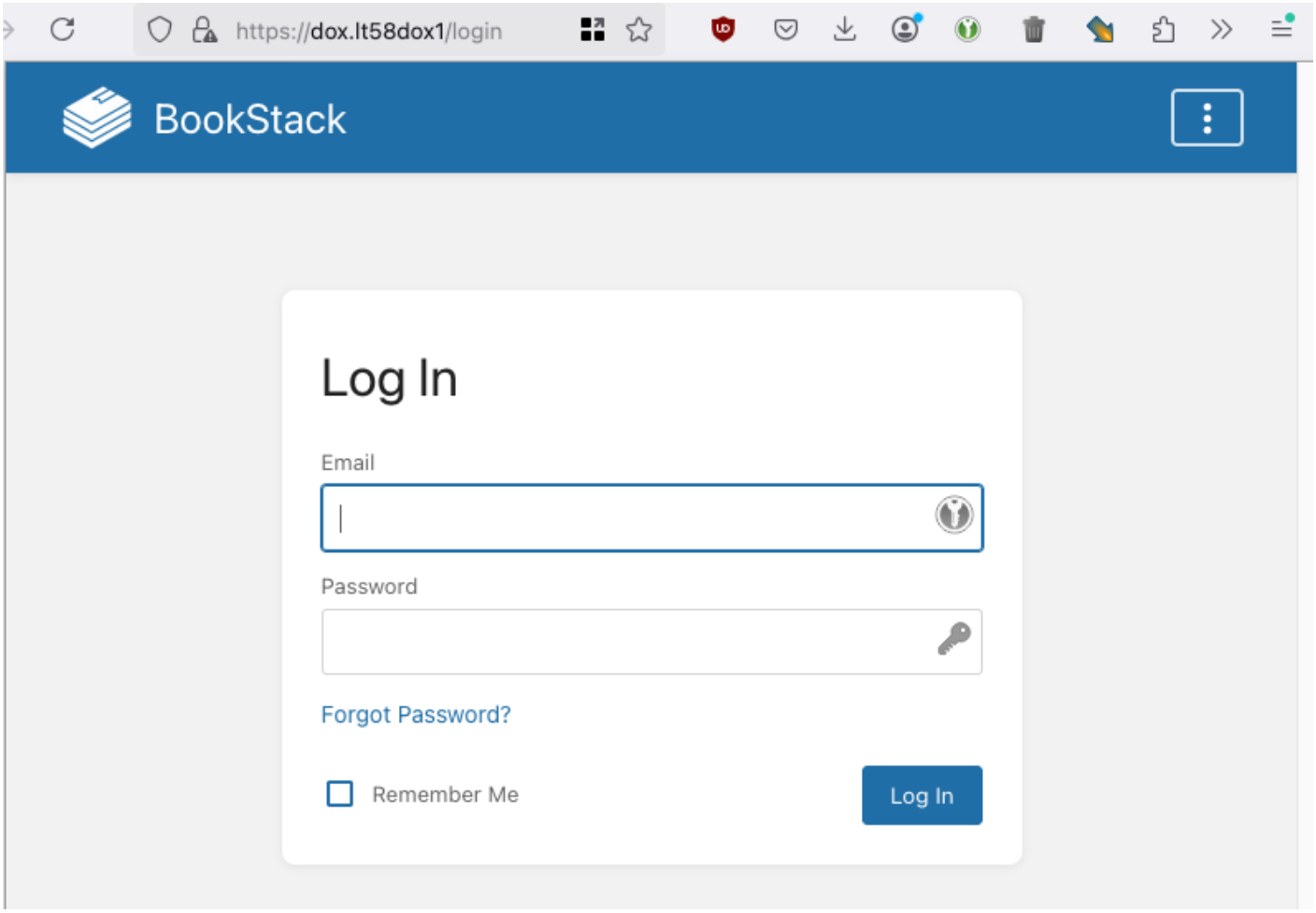
Should be simple as

```
APP_KEY=base64:goKeJWQsFFboBGOSF5+eti2Yv1auP4rXvxVbQ4Iupgc=
APP_URL=https://dox.lt58dox1

DB_HOST=localhost
DB_DATABASE=bookstack
DB_USERNAME=bookstack
DB_PASSWORD=Very-Strong-Password123

# not in use
MAIL_DRIVER=smtp
MAIL_FROM_NAME="BookStack"
MAIL_FROM=bookstack@example.com
MAIL_HOST=localhost
MAIL_PORT=587
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null
```

Create database schema

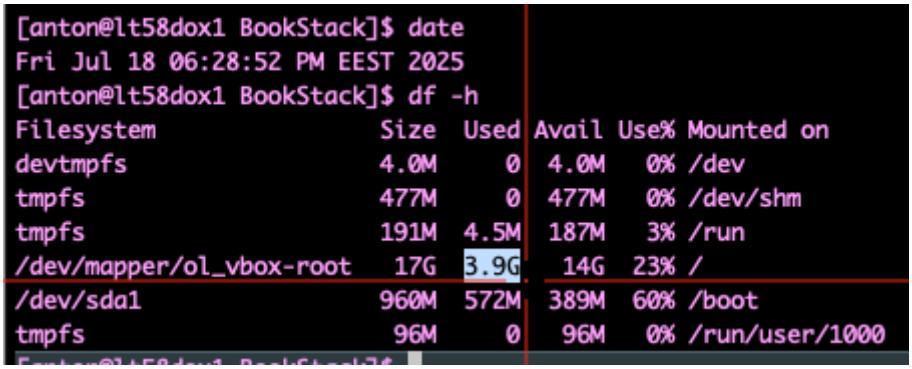
```
php artisan migrate
```

Application is ready



Used space:

Minimal OL9 + webserver + PHP-FPM + MariaDB




```
[anton@lt58dox1 BookStack]$ date
Fri Jul 18 06:33:49 PM EEST 2025
```

```
[anton@lt58dox1 BookStack]$ export dir="/var/www/${app}"
[anton@lt58dox1 BookStack]$ du -h ${dir} --max-depth=1
0       /var/www/cgi-bin
0       /var/www/html
176M    /var/www/dox.lt58dox1
176M    /var/www/
[anton@lt58dox1 BookStack]$ df -h ${dir}
Filesystem              Size  Used Avail Use% Mounted on
/dev/mapper/ol_vbox-root   17G  3.9G   14G  23% /
```

Revision #5
Created 18 July 2025 08:16:38 by Anton
Updated 21 July 2025 11:41:17 by Anton