

# Services

- [Concretete5](#)
- [Docker and Docker-compose](#)
- [npm / nodejs](#)

# Concrete5

## DataBase

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

```
CREATE DATABASE concrete_db;
```

```
CREATE USER 'concrete_user'@'localhost' IDENTIFIED BY 'ChangeME';
```

```
GRANT ALL PRIVILEGES ON concrete_db.* TO 'concrete_user'@'localhost';
```

```
FLUSH PRIVILEGES;
```

## Install PHP-mods

```
sudo apt install -y php8.2-curl php8.2-xml php8.2-gd php8.2-mbstring php8.2-mysql php8.2-cli php8.2-zip
```

## Install CMS

Get Downlod Link: <https://www.concretecms.org/download>

```
cd /var/www
```

```
sudo wget https://www.concretecms.org/download_file/9cbfcc11-37cc-433e-b18d-e57adb5c1c9f/2658 -O  
concrete.zip  
sudo unzip concrete.zip
```

```
sudo rm -rfv concrete.zip
```

```
sudo vim /etc/apache2/sites-available/000-default.conf
```

```
<VirtualHost *:80>
```

```
## The ServerName directive sets the request scheme, hostname and port that  
## the server uses to identify itself. This is used when creating  
## redirection URLs. In the context of virtual hosts, the ServerName  
## specifies what hostname must appear in the request's Host: header to  
## match this virtual host. For the default virtual host (this file) this  
## value is not decisive as it is used as a last resort host regardless.  
## However, you must set it for any further virtual host explicitly.  
##ServerName www.example.com
```

```
##ServerAdmin webmaster@localhost
```

```
##DocumentRoot /var/www/concrete-cms-9.2.1/
```

```
##<Directory /var/www/concrete-cms-9.2.1/>
```

```
##Options Indexes FollowSymLinks MultiViews
```

```
##AllowOverride All
```

```
##Order allow,deny
```

```
##allow from all
```

```
##</Directory>
```

```
# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,  
# error, crit, alert, emerg.  
# It is also possible to configure the loglevel for particular  
# modules, e.g.  
LogLevel info ssl:warn
```

```
ErrorLog ${APACHE_LOG_DIR}/error.log  
CustomLog ${APACHE_LOG_DIR}/access.log combined
```

```
# For most configuration files from conf-available/, which are  
# enabled or disabled at a global level, it is possible to  
# include a line for only one particular virtual host. For example the  
# following line enables the CGI configuration for this host only  
# after it has been globally disabled with "a2disconf".  
Include conf-available/serve-cgi-bin.conf  
</VirtualHost>
```

```
sudo chown -R www-data:www-data /var/www/
```

```
sudo systemctl restart apache2.service
```

# Backup

```
/opt/scripts/backup.sh
```

```
#!/bin/bash  
  
# Setzen Sie die Variablen  
BACKUP_DIR="/backup"  
WEB_DIR="/var/www/concrete-cms-9.2.1"  
DB_NAME="concrete_db"  
DB_USER="concrete_user"
```

```
DB_PASS="ChangeME"
```

```
# Generieren Sie den Dateinamen mit Datum und Uhrzeit
```

```
echo "Generate data name...."
```

```
DATE=$(date +%Y%m%d_%H%M%S)
```

```
FILE_NAME="backup_$(DATE).tar.gz"
```

```
DB_DUMP_NAME="db_dump_$(DATE).sql"
```

```
# Erstellen Sie die Unterordner, falls sie nicht existieren
```

```
echo "checking folders...."
```

```
mkdir -p $BACKUP_DIR/web
```

```
mkdir -p $BACKUP_DIR/DB
```

```
# Erstellen Sie das tar.gz-Archiv
```

```
echo "compressing...."
```

```
tar -czf $BACKUP_DIR/web/$FILE_NAME $WEB_DIR
```

```
# Erstellen Sie den Datenbank-Dump
```

```
echo "dumping...."
```

```
mysqldump -u $DB_USER -p$DB_PASS $DB_NAME > $BACKUP_DIR/DB/$DB_DUMP_NAME
```

```
# Funktion zum Löschen der ältesten Backups, wenn mehr als 30 vorhanden sind
```

```
cleanup() {
```

```
    echo "Checking if cleanup is necessary..."
```

```
    while [ $(ls -1qA $BACKUP_DIR/web | wc -l) -gt 30 ]
```

```
    do
```

```
        OLDEST_BACKUP=$(ls -tr $BACKUP_DIR/web | head -1)
```

```
        echo "Deleting oldest backup: $OLDEST_BACKUP"
```

```
        rm -f $BACKUP_DIR/web/$OLDEST_BACKUP
```

```
    done
```

```
    while [ $(ls -1qA $BACKUP_DIR/DB | wc -l) -gt 30 ]
```

```
    do
```

```
        OLDEST_BACKUP=$(ls -tr $BACKUP_DIR/DB | head -1)
```

```
        echo "Deleting oldest backup: $OLDEST_BACKUP"
```

```
        rm -f $BACKUP_DIR/DB/$OLDEST_BACKUP
```

```
    done
```

```
}
```

```
# Führen Sie die Bereinigungsfunktion aus
```

```
cleanup
```

```
echo "Backup erfolgreich erstellt in $BACKUP_DIR/web/$FILE_NAME und $BACKUP_DIR/DB/$DB_DUMP_NAME"
```

## Crontab

```
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow  command

0 2 * * * /opt/scripts/backup.sh
```

# Docker and Docker-compose

```
# Add Docker's official GPG key:
sudo apt update; \
sudo apt install -y ca-certificates curl gnupg; \
sudo install -m 0755 -d /etc/apt/keyrings; \
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg; \
sudo chmod a+r /etc/apt/keyrings/docker.gpg; \

# Add the repository to Apt sources:
echo \
  "deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/debian \
  "$(. /etc/os-release && echo "$VERSION_CODENAME")" stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt update
```

```
sudo apt install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

## If Failure like this:

failed to register layer: lsetxattr user.overlay.impure /etc: operation not supported

## Do on vhost:

```
vim /etc/pve/lxc/xxx.conf
```

and add:

```
unprivileged: 0 # only change value
lxc.apparmor.profile: unconfined
lxc.cap.drop:
lxc.cgroup.devices.allow: a
```





# npm / nodejs

Follow instructions:

<https://github.com/nodesource/distributions#using-debian-as-root-4>