# Container 'mariadb'

## Ressourcen

- 1 GB RAM
- 2 Cores
- 8 GB HDD (root-fs)

## System

- interne IPs
  - 10.2.0.100, fd00:10:2:0::100
  - 10.3.0.100, fd00:10:3:0::100 (MariaDB)

## Dienste

- MariaDB 10

## Datenbanken

| Datenbank | Benutzer | Verwendung |
|---|---|---|
| nextcloud | nc_user | Nextcloud |
| wordpress | wp_user | Wordpress |

## Betrieb

### Datenbank und Benutzer anlegen

1. Zur Datenbank verbinden
   - **sudo mysql**
2. Datenbank anlegen und Benutzer mit Passwort zuweisen

   MySQL-Konsole "MariaDB [(none)]"

```
CREATE DATABASE databasename;
GRANT ALL PRIVILEGES ON databasename.* TO 'username'@'%' IDENTIFIED BY 'password';
FLUSH PRIVILEGES;
```

### Passwort für Benutzer ändern

1. Zur Datenbank verbinden
   - **sudo mysql**
2. Benutzer neues Passwort zuweisen

   MySQL-Konsole "MariaDB [(none)]"

```
ALTER USER 'username'@'%' IDENTIFIED BY 'password';
FLUSH PRIVILEGES;
```

## Installation

- Standard-Template mit Benutzern

### MariaDB

1. MariaDB-Server installieren
   - **apt-get install mariadb-server**
2. MariaDB - Erstkonfiguration

- **mysql_secure_installation**
  - Set root password? [Y/n]: **Y**
  - New password: **PASSWORT**
  - Re-enter new password: **PASSWORT**
  - 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**

3. Benutzerdefinierte Konfiguration anlegen

   /etc/mysql/mariadb.conf.d/99-bytecluster.cnf

```
[mysqld]
# An lokale IP binden
bind-address                    = 10.3.0.100

# Binlog deaktivieren
skip-log-bin

# InnoDB verwenden
default_storage_engine          = InnoDB

# InnoDB-Optimierungen
innodb_buffer_pool_size         = 256M
innodb_log_buffer_size          = 8M
innodb_log_file_size            = 128M

innodb_log_files_in_group       = 2
innodb_flush_log_at_trx_commit  = 2
innodb_flush_method             = O_DIRECT
innodb_file_per_table           = 1
```

4. MariaDB neustarten
   - **systemctl restart mariadb.service**

## Backup mit Borgmatic

1. Debian Testing-Repo „Bullseye) integrieren (für Borgmatic 1.5, da Debian 10 noch Borgmatic 1.2 beinhaltet)
   - Standard-Installationsquelle auf „stable“ stellen

     /etc/apt/apt.conf.d/99defaultrelease

```
APT::Default-Release "stable";
```

   - Testing-Repo integrieren

     /etc/apt/sources.list.d/hetzner-mirror-testing.list

```
deb http://mirror.hetzner.de/debian/packages bullseye main contrib
deb http://mirror.hetzner.de/debian/packages bullseye-updates main contrib
deb http://mirror.hetzner.de/debian/packages bullseye-backports main contrib
```

2. Borgmatic aus Testing installieren
   - **sudo apt-get update**
   - **sudo apt-get -t testing install borgmatic**
3. Borgmatic-Konfiguration in der Datei /etc/borgmatic/config.yaml erzeugen
   - **sudo generate-borgmatic-config**
4. SSH-Key erzeugen
   - **sudo ssh-keygen -a100 -t ed25519 -f /root/.ssh/id_borgbackup**
5. SSH-Key bei Backupserver hinterlegen
6. Zufälliges Passwort in Konfiguration erzeugen
   - **sudo sed -i -e "s|^\( encryption_passphrase: \"\).*\(\"\)$|\1PASS_TO_REPLACE\2|"**

**/etc/borgmatic/config.yaml**
- **sudo sed -i "s|PASS_TO_REPLACE|$(pwgen -cnysB -1 32 -r \"\^\|\\\\)|" /etc/borgmatic/config.yaml**

7. Konfiguration anpassen (USERNAME, SERVERNAME, SSH-PORT ersetzen)

   /etc/borgmatic/config.yaml

```
location:
    ...
    source_directories:
        - /etc
        - /home
        - /root
        - /usr/local
        - /var/log
    ...
    repositories:
        - USERNAME@SERVERNAME:~/borg
    ...
    one_file_system: true
    ...
    exclude_caches: true
...
storage:
    ...
    encryption_passphrase: "ENCRYPTION-PASSPHRASE"
    ...
    compression: zlib,9
    ...
    ssh_command: ssh -i /root/.ssh/id_borgbackup -p SSH-PORT
...
retention:
    ...
    keep_daily: 7
    ...
    keep_weekly: 4
    ...
    keep_monthly: 6
    ...
    keep_yearly: 1
...
consistency:
    ...
    checks:
        - repository
        - archives
...
hooks:
    before_backup:
        - dpkg-query -f '${binary:Package}\n' -W > /root/package.list
    ...
    mysql_databases:
        - name: all
    ...
```

8. Borg-Repository initialisieren
   - **sudo borgmatic init –encryption keyfile**
9. Verschlüsselungsinformationen sicher verwahren
   1. Verschlüsselungs-Passwort ermitteln
      - **sudo grep „encryption_passphrase:“ /etc/borgmatic/config.yaml**
   2. Verschlüsselungs-Schlüssel ermitteln
      - **sudo cat /root/.config/borg/keys/SERVERNAME-MIT-UNTERSTRICHEN__borg**
10. Erstes Backup initialisieren

- **sudo borgmatic create -progress -stats**

```
------------------------------------------------------------------------------
Archive name: mariadb-2020-12-05T17:36:48.502653
Archive fingerprint:
9a9c7f769dcd9af1e7f28158e4a3b0d05cb3faae25c45a6e9930591a1414eaa3
Time (start): Sat, 2020-12-05 17:36:49
Time (end):   Sat, 2020-12-05 17:36:55
Duration: 6.57 seconds
Number of files: 472
Utilization of max. archive size: 0%
------------------------------------------------------------------------------
                       Original size      Compressed size    Deduplicated size
This archive:               71.08 MB              2.72 MB              2.53 MB
All archives:               71.08 MB              2.72 MB              2.53 MB

                       Unique chunks         Total chunks
Chunk index:                     437                  464
------------------------------------------------------------------------------
```

11. Backupinhalt nochmal prüfen
    - **sudo borgmatic list -archive latest**

```
USERNAME@SERVERNAME:~/borg: Listing archives
drwxr-xr-x root   root          0 Wed, 2020-11-25 19:49:45 etc
-rw-r--r-- root   root        767 Fri, 2016-03-04 11:00:00 etc/profile
...
```

12. Cronjob einrichten
    - **echo -e "0 3 * * *\troot\t$(which borgmatic) --syslog-verbosity 1" | sudo tee /etc/cron.d/borgmatic > /dev/null**