

## bytecluster0002

---

bytecluster0002 ist ein Virtualisierungsserver, der Kommunikationsdienste für den Verein bereitstellt. Er löst [bytecluster0001](#) ab.

## Virtueel Hardware

---

\* 4 Kerne \* 16 GB RAM \* Volumes

- 160 GB (inkludiert)
- 100 GB (zusätzlich)

## Migration

---

- mehr zur Migration unter [migration](#)

## Festlegungen (Diskussionsgrundlage)

---

- Kein produktiver Container ohne Dokumentation.
- Der erste Schritt beim Installieren und Testen ist immer einer leere Dokumentation.
- Container ohne Dokumentation werden ohne Rückfrage heruntergefahren und bei Kapazitätsengpässen entfernt.
- Spiel- und Testcontainer sind eindeutig im Namen gekennzeichnet.
- Mit Ressourcen (Disk/RAM) wird verantwortungsvoll umgegangen - Upscaling geht später immer.

## Container

---

- [wiki](#)
- [traefik](#)
- [mariadb](#)
- [nextcloud](#)
- [web](#)
- [wordpress](#)

## Administratoren

---

- [mape2k](#)
- [maddi](#)
- [suicider](#)
- [chaos](#)

## IPs /DNS

---

### extern

---

- [bytecluster0002.bytespeicher.org](https://bytecluster0002.bytespeicher.org)
  - 138.201.246.25
  - 2a01:4f8:c17:cf64::1

### intern

---

- Vergabe siehe [host-netzwerke](#)
- Netzwerk für Internetzugang und Traefik
  - 10.2.0.254/24
  - fd00:10:2:0::0/64
- Netzwerk für Datenbankserver und -clients

- 10.3.0.0/24
- fd00:10:3:0::0/64
- IPs liegen am Host nicht an

## Betrieb

### Benutzer anlegen

1. Benutzer anlegen
  1. Normaler Benutzer ohne sudo-Rechte
    - **useradd --create-home --shell /bin/bash --comment "Max Mustermann" mustermann**
  2. Benutzer mit sudo-Rechten
    - **useradd --create-home --shell /bin/bash --comment "Max Mustermann" --groups sudo mustermann**
2. SSH-Key hinterlegen
  1. SSH-Verzeichnis anlegen
    - **mkdir /home/mustermann/.ssh**
  2. SSH-Schlüssel in Datei authorized\_keys hinterlegen  
  
/home/mustermann/.ssh/authorized\_keys  
  

```
ssh-rsa AAAA... KOMMENTAR
```
3. Berechtigungen und Rechte anpassen
  - **chown --recursive mustermann:mustermann /home/mustermann/.ssh**
  - **chmod 700 /home/mustermann/.ssh**
  - **chmod 644 /home/mustermann/.ssh/authorized\_keys**
4. Passwort setzen
  - Das Passwort ist für den Nutzung von sudo und für die Proxmox-Weboberfläche gültig und sollte vom Benutzer dann geändert werden!
  - **passwd mustermann**

### Benutzer-Zugang zu Proxmox als Admin gewähren

1. Benutzer als Admin hinzufügen zuweisen
  - **pveum user add mustermann@pam -groups admin -enable 1 -firstname „Max“ -lastname „Mustermann“**
2. Login des Benutzers
  - **passwd** - Passwort ändern
  - **pve\_generate\_oath**
    - QR-Code mit geeignetem 2FA-Client scannen und nach Enter Ausführung mit eigenem Passwort (für sudo) bestätigen

## Installation

### Betriebssystem

- Debian 10 minimal (vorinstalliert)

### Vorkonfiguration

1. Vorgeschlagene Pakete nicht mit installieren (bereits im Standard vom Provider vorhanden)

/etc/apt/apt.conf.d/00InstallRecommends

```
APT::Install-Recommends "false";
```

## Grundeinrichtung

# Migration

1. System aktualisieren
  - **apt-get update**
  - **apt-get dist-upgrade**
2. Notwendige Standardsoftware installieren
  - vim (Editor)
  - mc (Dateimanager)
  - debian-goodies (Debian-Systemtools)
  - needrestart (Prüfung von Diensteneustarts nach Update)
  - net-tools (Netzwerktools)
  - **apt-get install vim mc debian-goodies needrestart net-tools**
3. Suche in der Konsole mit Bild-ab/Bild-auf aktivieren

/etc/inputrc

```
...
# alternate mappings for "page up" and "page down" to search the history
"\e[5~": history-search-backward
"\e[6~": history-search-forward
...
```

## Absicherung

1. NFS / rpcbind deaktivieren und beenden (wird nicht benötigt, offene Ports schließen)
  - **systemctl disable --now rpcbind.service rpcbind.socket**
2. sudo installieren und konfigurieren
  - **apt-get install sudo**
  - Konfiguration prüfen, so dass sudo von Nutzern der Gruppe sudo genutzt werden kann

/etc/sudoers

```
# Allow members of group sudo to execute any command
%sudo    ALL=(ALL:ALL) ALL
```

3. SSH - Login als root und mit Passwort deaktivieren
  - Vorher mindestens einen Benutzer einrichten, der einen SSH-Schlüssel hinterlegt hat!
1. Konfiguration anpassen

/etc/ssh/sshd\_config

```
...
PermitRootLogin no
...
PasswordAuthentication no
...
ChallengeResponseAuthentication no
...
```

2. SSH-Daemon neustarten
  - **systemctl restart sshd**

## Netzwerk

### Bridges für Netzwerk(e) einrichten

Die Einrichtung von Bridges sollte nicht über die Web-GUI erfolgen, da dabei u.U. bestehende Konfigurationen aus dem Ordner /etc/network/interfaces.d nicht mehr funktionieren. Die Bridges werden in /etc/network/interfaces angelegt, damit

sie in der Proxmox-GUI sichtbar sind.

1. Bridge für Internetzugang in Containern und Datenbanknetzwerk anlegen

/etc/network/interfaces

```
...
auto vmbr0
iface vmbr0 inet static
    address 10.2.0.254
    netmask 255.255.255.0
    bridge_ports none
    bridge_stp off
    bridge_fd 0
#Frontend-Netzwerk (Traefik) mit Internetzugang
iface vmbr0 inet6 static
    address fd00:10:2:0::0
    netmask 64

auto vmbr1
iface vmbr1 inet manual
    bridge_ports none
    bridge_stp off
    bridge_fd 0
#Datenbanken
```

2. Bridges starten
  - **ifup vmbr0**
  - **ifup vmbr1**

## Paketfilter einrichten

**Hinweis:** Es wurde iptables statt nftables (Standard bei Debian Buster) verwendet, weil nftables noch kein NETMAP unterstützt.

1. iptables-persistent installieren, um iptables-Regeln für Neustarts zu speichern
  - **apt-get install iptables-persistent**
  - Aktuelle Regeln nicht speichern
2. NAT (portbasiert) für IPv4
  - **iptables -t nat -A POSTROUTING -o eth0 -s 10.2.0.0/24 -j MASQUERADE**
3. NAT-Reflection für IPv4 (Zugriff von intern auf externe IP-Adresse für Ports 80/443)
  - **iptables -t nat -A PREROUTING -i vmbr0 -s 10.2.0.0/24 -d 138.201.246.25/32 -p tcp -m tcp --dport 80 -j DNAT --to-destination 10.2.0.1:80**
  - **iptables -t nat -A PREROUTING -i vmbr0 -s 10.2.0.0/24 -d 138.201.246.25/32 -p tcp -m tcp --dport 443 -j DNAT --to-destination 10.2.0.1:443**
  - **iptables -t nat -A POSTROUTING -o vmbr0 -s 10.2.0.0/24 -j SNAT --to-source 138.201.246.25**
4. NAT (prefixbasiert) für IPv6
  - **ip6tables -t nat -A POSTROUTING -o eth0 --to 2a01:4f8:c17:cf64:ffff::/80 -s fd00:10:2:0::/64 -j NETMAP**
  - **ip6tables -t nat -A PREROUTING -i eth0 -d 2a01:4f8:c17:cf64:ffff::/80 --to fd00:10:2:0::/64 -j NETMAP**
5. NAT-Reflection für IPv6 (Zugriff von intern auf externe IP-Adresse)
  - **ip6tables -t nat -A POSTROUTING -o vmbr0 -s fd00:10:2:0::/64 -j SNAT --to-source 2a01:4f8:c17:cf64::1**
6. Regeln speichern
  - **netfilter-persistent save**

## Forwarding aktivieren

1. sysctl-Konfiguration erstellen

/etc/sysctl.d/99-forward.conf

```
# Forwarding aktivieren
net.ipv4.conf.eth0.forwarding = 1
net.ipv4.conf.vmr0.forwarding = 1
net.ipv6.conf.all.forwarding = 1
```

2. sysctl-Konfiguration übernehmen
  - **sysctl --system -a**

## Eingerichtete Forwards

- Port 80/443 für Container [traefik](#)
  - **iptables -t nat -A PREROUTING -d 138.201.246.25/32 -p tcp -m tcp -dport 80 -j DNAT -to-destination 10.2.0.1:80**
  - **iptables -t nat -A PREROUTING -d 138.201.246.25/32 -p tcp -m tcp -dport 443 -j DNAT -to-destination 10.2.0.1:443**

- NGINX default im web-container

```
iptables -t nat -A PREROUTING -d 138.201.246.25/32 -p tcp --dport 8089 -j DNAT --to 10.2.0.10:8089
```

- Dokuwiki im web-container

```
iptables -t nat -A PREROUTING -d 138.201.246.25/32 -p tcp --dport 8088 -j DNAT --to 10.2.0.10:8088
```

- Nextcloud im nextcloud.test container

```
iptables -t nat -A PREROUTING -d 138.201.246.25/32 -p tcp --dport 8087 -j DNAT --to 10.2.0.20:8087
```

- Wordpress im wordpress.test container

```
iptables -t nat -A PREROUTING -d 138.201.246.25/32 -p tcp --dport 8086 -j DNAT --to 10.2.0.30:8086
```

## Proxmox

- nach Anleitung: [https://pve.proxmox.com/wiki/Install\\_Proxmox\\_VE\\_on\\_Debian\\_Buster](https://pve.proxmox.com/wiki/Install_Proxmox_VE_on_Debian_Buster)

## Vorbereitung

1. Hosts-Datei anpassen
  - IP-Adresse des internen Netzes nutzen, so dass später ein Proxmox-Cluster möglich ist
  - Konfiguration

/etc/hosts

```
...
# 127.0.1.1 bytecluster0002 bytecluster0002
127.0.0.1 localhost
10.10.0.2 bytecluster0002.bytespeicher.org bytecluster0002 pvelocalhost
...
```

## Installation

1. Installation nach Anleitung:
  - [https://pve.proxmox.com/wiki/Install\\_Proxmox\\_VE\\_on\\_Debian\\_Buster#Install\\_Proxmox\\_VE](https://pve.proxmox.com/wiki/Install_Proxmox_VE_on_Debian_Buster#Install_Proxmox_VE)
  - bei **apt full-upgrade** mit „install the package maintainer's version“ die Konfiguration für grub-efi-amd64 übernehmen
  - für den Punkt „Install Proxmox VE packages“ nur **apt install proxmox-ve postfix** ausführen, da open-iscsi

nicht benötigt wird

- Modify smb.conf to use WINS settings from DHCP? **No**
- Postfix
  - Postfix Configuration: **Local only**
  - System Name: **bytecluster0002**

## Anpassung der Update-Repository

Proxmox richtet das Repository für die Enterprise-Version mit ein. Ohne Subskription schlägt das Update der Quelle aber fehl und sie muss daher deaktiviert werden.

1. Enterprise-Repository deaktivieren
  - **sed -i -e 's/^/# ' /etc/apt/sources.list.d/pve-enterprise.list**

## 2FA Grundeinrichtung

1. Skript anlegen

/usr/local/bin/pve\_generate\_oath

```
#!/bin/bash

clear

USERNAME=$USER
HOSTNAME=$(hostname --fqdn)
OATHKEY=$(oathkeygen)

qrencode -t ANSIUTF8 -o - "$(echo otpauth://totp/Proxmox $HOSTNAME?secret=$OATHKEY)"

read -p "Scan QR code in your application and press enter to activate. Otherwise press Ctrl+C" -n1 -s
sudo pveum user modify $USER@pam -keys $OATHKEY
```

2. Berechtigungen anpassen und ausführbar machen
  - **chown root:root /usr/local/bin/pve\_generate\_oath**
  - **chmod 755 /usr/local/bin/pve\_generate\_oath**
3. 2FA für PAM-Anmeldungen verpflichtend machen
  - **pveum realm modify pam -tfa type=oath,digits=6 -default 1**

## Admin-Gruppe und ersten Benutzer anlegen

1. Admin-Gruppe anlegen
  - **pveum group add admin -comment „Administrators“**
  - **pveum aclmod / -group admin -role Administrator**
2. ersten Benutzer zuweisen und root sperren
  - **pveum user add mustermann@pam -groups admin -enable 1 -firstname „Max“ -lastname „Mustermann“**
  - **pveum user modify root@pam -enable 0**
3. 2FA für ersten Benutzer aktivieren
  - **ALS BENUTZER AUSFÜHREN** - vorher also **su mustermann** (falls als root eingeloggt)
  - **pve\_generate\_oath**
    - QR-Code scannen und nach Enter ggf. Ausführung mit eigenem Passwort für sudo bestätigen

## SSL mit Let's Encrypt

Quelle: [https://pve.proxmox.com/wiki/Certificate\\_Management](https://pve.proxmox.com/wiki/Certificate_Management)

1. Mail-Account für Let's Encrypt registrieren
  - **pvenode acme account register default xxxxxxxxxx@bytespeicher.org**

```

Directory endpoints:
0) Let's Encrypt V2 (https://acme-v02.api.letsencrypt.org/directory)
1) Let's Encrypt V2 Staging
  (https://acme-staging-v02.api.letsencrypt.org/directory)
2) Custom
Enter selection: 0

Attempting to fetch Terms of Service from
'https://acme-v02.api.letsencrypt.org/directory'..
Terms of Service:
https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf
Do you agree to the above terms? [y|N]: y

Attempting to register account with
'https://acme-v02.api.letsencrypt.org/directory'..
Generating ACME account key..
Registering ACME account..
Registration successful, account URL:
'https://acme-v02.api.letsencrypt.org/acct/XXXXXXXX'
Task OK

```

2. Domain hinterlegen
  - **pvenode config set --acme domains=\$(hostname --fqdn)**
3. Erstes Zertifikat initialisieren
  - **pvenode acme cert order**

```

...
Task OK

```

## SSH-Forward-Hook einrichten

1. Skript hinterlegen

/var/lib/vz/snippets/ssh\_forward.sh

```

#!/usr/bin/sh
# Hook script to automatically forward SSH-Port to Container

VMID=$1
PHASE=$2

NETWORK_INTERFACE_EXTERNAL=eth0
NETWORK_INTERFACE_INTERNAL=vbr0
SSH_DESTINATION_PORT=22
SSH_FORWARD_BASE_PORT=22000

# Determine container details
IP_ADDRESS_INTERNAL=$(pct config $VMID | grep "bridge=$NETWORK_INTERFACE_INTERNAL" |
sed --regexp-extended 's/.*ip=([0-9\.]*)\./\1/')
# IP_LAST_OCTET=$(pct config $VMID | grep "bridge=$NETWORK_INTERFACE_INTERNAL" | sed --
regexp-extended 's/.*ip=[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+\./\1/')
SSH_FORWARD_PORT=$(expr $SSH_FORWARD_BASE_PORT + $VMID)

echo "GUEST HOOK on VMID $VMID for phase $PHASE"

echo "Internal IP address: $IP_ADDRESS_INTERNAL"
echo "SSH-Port on host: $SSH_FORWARD_PORT"

case "$PHASE" in

```

```
# First phase 'pre-start' will be executed before the guest
# ist started. Exiting with a code != 0 will abort the start
pre-start) echo "$VMID is starting, doing preparations.\n"
            iptables -t nat -A PREROUTING -i $NETWORK_INTERFACE_EXTERNAL -p tcp --
dport $SSH_FORWARD_PORT -j DNAT --to $IP_ADDRESS_INTERNAL:$SSH_DESTINATION_PORT
            ;;

# Second phase 'post-start' will be executed after the guest
# successfully started.
post-start) echo "$VMID started successfully.\n"
            ;;

# Third phase 'pre-stop' will be executed before stopping the guest
# via the API. Will not be executed if the guest is stopped from
# within e.g., with a 'poweroff'
pre-stop) echo "$VMID will be stopped.\n"
          ;;

# Last phase 'post-stop' will be executed after the guest stopped.
# This should even be executed in case the guest crashes or stopped
# unexpectedly.
post-stop) echo "$VMID stopped. Doing cleanup.\n"
            iptables -t nat -D PREROUTING -i $NETWORK_INTERFACE_EXTERNAL -p tcp --
dport $SSH_FORWARD_PORT -j DNAT --to $IP_ADDRESS_INTERNAL:$SSH_DESTINATION_PORT
            ;;

*) echo "Got unknown phase '$PHASE'\n"
   exit 1
   ;;
esac
```

2. Skript ausführbar machen
  - **chmod +x /var/lib/vz/snippets/ssh\_forward.sh**

## Zusätzliches Volume einrichten

1. Volume vorbereiten als /mnt/data
2. Volume in Proxmox einbinden
  - **pvesm add dir data -content rootdir,images,backup -path /mnt/data/**

## Anpassung des Standard-Templates auf Debian-Basis

1. Systemd-Container installieren
  - **apt-get install systemd-container**
2. Liste der verfügbaren Template aktualisieren
  - **pveam update**
3. Verfügbare Images anzeigen
  - **pveam available --section system | grep debian**

system	debian-10.0-standard_10.0-1_amd64.tar.gz
system	debian-8.0-standard_8.11-1_amd64.tar.gz
system	debian-9.0-standard_9.7-1_amd64.tar.gz

4. Debian 10 Image herunterladen
  - **pveam download local debian-10.0-standard\_10.0-1\_amd64.tar.gz**
5. Template in neuen Ordner entpacken
  - **mkdir /tmp/template**
  - **cd /tmp/template**
  - **tar --numeric-owner --extract --verbose --file=/var/lib/vz/template/cache/debian-10.0-standard\_10.0-1\_amd64.tar.gz --directory=/tmp/template**
6. In das Template-System wechseln
  - **systemd-nspawn -D /tmp/template**



## Ausgabe

```
Spawning container template on /tmp/template.
Press ^] three times within 1s to kill container.
root@template:~#
```

## 7. Template: Konfiguration und Software anpassen

## 1. APT-Quellen auf Hetzner festlegen

- **echo „deb <http://mirror.hetzner.de/debian/security> buster/updates main contrib non-free“ > /etc/apt/sources.list.d/hetzner-security-updates.list**
- **echo „deb <http://mirror.hetzner.de/debian/packages> buster main contrib non-free“ > /etc/apt/sources.list.d/hetzner-mirror.list**
- **echo „deb <http://mirror.hetzner.de/debian/packages> buster-updates main contrib non-free“ » /etc/apt/sources.list.d/hetzner-mirror.list**
- **echo „deb <http://mirror.hetzner.de/debian/packages> buster-backports main contrib non-free“ » /etc/apt/sources.list.d/hetzner-mirror.list**

## 2. Alle Änderungen aus Betriebssystem von bytecluster0002 vornehmen

- Ausnahmen: NFS deaktivieren und SSH neustarten

## 3. Template bereinigen

- **apt-get clean**
- **history -c**

## 4. Aus Template ausloggen

- **logout**

## 8. Template packen und temporären Ordner entfernen

- **tar --numeric-owner --create --gzip --verbose --file=/var/lib/vz/template/cache/debian-10-\$(hostname).tar.gz .**
- **cd**
- **rm --recursive /tmp/template**

## Neuen Container anlegen

### Anmelden bei Proxmox Web-Gui

## 1. Vorbedingung: 2FA aktivieren

- Auf Console einloggen (via hinterlegtem Public Key)
- **pve\_generate\_oath** ausführen (als normaler Nutzer)
- Angezeigten QR-Code mit Smartphone scannen und an beliebiges 2FA tool (z.B. Google Authenticator) weiterleiten
- Wichtig: QR-Code-Anzeige in der Konsole mit **Enter** schliessen

2. <https://bytecluster0002.bytespeicher.org:8006> aufrufen

- Username + Passwort wie im Linux
- 2FA-Token aus der Smartphone-App

### Container anlegen und konfigurieren

1. **Create CT** Button oben rechts
2. **hostname** frei wählen (z.B. 'web'), **CT ID** zählt automatisch hoch, **SSH public key** vom eigenen Rechner hochladen (derselbe, der für den Login zum host-server benutzt wird), **Next** klicken
3. **template** *debian-10-bytecluster0002-with-users* wählen, **Next** klicken
4. Plattengröße, Cores und RAM wählen
5. Unter **network** statische IPs vergeben: IPv4: 10.2.0.x/24 Gateway 10.2.0.0 IPv6: fd00:10:2:0::x/64 Gateway fd00:10:2:0::0 mit freiem x (0 = host, 10 = web, ...);
6. Im letzten Tab bestätigen und ggfs. Container sofort starten.

### Automatischen SSH-Forward für Container konfigurieren

1. Per SSH auf dem Host einloggen
2. Hook-Skript an Container binden
  - **sudo pct set <CT ID> -hookscript local:snippets/ssh\_forward.sh**
3. Container ist nach dem (Neu)Start per SSH auf Port 22000 + <CT ID> (z.B. 22149 für Container mit ID 149) direkt erreichbar

## In Container einloggen

- Vor weiteren Konfigurationen ist der Container nicht von aussen erreichbar, sondern nur über den Host
- Die Benutzer-Accounts des Hosts sind auch im Container angelegt, das Passwort liegt in einer `password.txt` im Home des Benutzers im Container
- Ein Login ist über mittels SSH-Key möglich, die Keys sind in den Containern vorab hinterlegt
  - Es empfiehlt sich, SSH Agent forwarding zu nutzen, statt seinen private key auf den Host zu kopieren

### Disclaimer

Wird SSH Agent Forwarding genutzt, kann potentiell jeder mit root/sudo-Rechten solange ihr eingeloggt seid euren Key benutzen, um sich in eurem Namen lokal oder auf weiteren (auch externen) Servern einzuloggen, auf denen der selbe Key benutzt wird!

- Dazu am eigenen Rechner einmal **ssh-add** (je nach system einmalig oder nach jedem Login), dann **ssh -A bytecluster0002.bytespeicher.org**; von dort dann **ssh 10.2.0.x** zum Container
- Zum Passwort ändern kann man sich zur Konsole des Containers verbinden
- Von der Konsole des Hosts aus mit **sudo pct console <CT ID>**
- Über das Web-Frontend kann man die „Console“ nutzen
- In der Konsole kann man sich als root mit dem festgelegten Passwort einloggen und mit **passwd <nutzernamen>** das Benutzer-Passwort ändern
- Alle bekannten Nutzer sind sudo-fähig

#### Dauerhafter Link zu diesem Dokument:

<https://wiki.technikkultur-erfurt.de/dienste:bytecluster0002?rev=1674394467>

Dokument zuletzt bearbeitet am: **22.01.2023 14:34**

**Verein zur Förderung von Technikkultur in Erfurt e.V**

<https://wiki.technikkultur-erfurt.de/>

