LXC Toolbox für Proxmox

Maßgeschneiderte und in der Praxis gereifte Software in der speziellen Installerversion für Proxmox VE mit ZFS

- **Die schlimmsten Fehler im Umgang mit Proxmox VE und ZFS**
- **Mailpiler auf Proxmox VE installieren – E-Mail-Archiv im LXC**

Die schlimmsten Fehler im Umgang mit Proxmox VE und ZFS

Proxmox VE Best Practices – wichtige Tipps für Installation und Betrieb

Hinweis zum Alter des Beitrags: Diese Sammlung ist über mehrere Jahre aus unserer täglichen Arbeit mit **Proxmox VE, ZFS, Backup, Storage und Virtualisierung** entstanden. Einzelne Hinweise stammen daher ursprünglich aus älteren Proxmox-Versionen.

Die grundlegenden Empfehlungen und Erfahrungen sind weiterhin aktuell. Wo sich Proxmox VE technisch weiterentwickelt hat, sollte die jeweilige Konfiguration mit der aktuell eingesetzten PVE-Version abgeglichen werden.

Beim Aufbau einer **Proxmox VE Infrastruktur** gibt es zahlreiche Details, die über eine reine Standardinstallation hinausgehen. Viele Probleme zeigen sich erst im produktiven Betrieb – beispielsweise beim Austausch einer Festplatte, bei ZFS, Backup, Replikation, Updates oder nach einem Boot-Disk-Ausfall.

Die folgenden Punkte basieren auf unseren Erfahrungen aus realen **Proxmox-Installationen und Kundenumgebungen**.

Festplatten und Hardware dokumentieren

Seriennummern der Festplatten dokumentieren

Notiert bei der Installation die **Seriennummern der Festplatten und deren Position in den Einschüben**.

Bei einem späteren Ausfall muss eindeutig erkennbar sein, welche physische Disk zu einem ausgefallenen ZFS-Gerät gehört. Fehlt diese Dokumentation, kann im ungünstigsten Fall eine zusätzliche Prüfung oder sogar ein Herunterfahren des Servers notwendig werden, bevor die richtige Festplatte sicher getauscht werden kann.

Hardware-RAID und ZFS nicht vermischen

Wichtig bei ZFS: Hardware-RAID-Controller sollten nicht dadurch als vermeintlicher HBA verwendet werden, dass für jede einzelne Festplatte ein eigenes RAID0 angelegt wird.

Andere haben mit solchen Konstruktionen schlechte Erfahrungen bis hin zu **Datenverlust** gemacht.

Für ZFS sollte ein geeigneter **HBA bzw. JBOD-Modus** verwendet werden, bei dem ZFS möglichst direkten Zugriff auf die einzelnen Datenträger erhält.

ZFS statt LVM für viele Proxmox-Szenarien

Wir bevorzugen bei unseren Proxmox-Installationen in vielen Fällen **ZFS gegenüber klassischem LVM**.

ZFS bietet unter anderem Prüfsummen, Snapshots, Replikation, Kompression, Mirrors und RAID-Z sowie umfangreiche Möglichkeiten zur Überwachung des Storage.

Das bedeutet allerdings nicht, dass LVM technisch grundsätzlich ungeeignet ist. Die Wahl des Storage-Systems sollte zum jeweiligen Einsatzzweck passen.

RAID-Z für virtuelle Maschinen sorgfältig planen

RAID5/RAID6 bzw. ZFS RAIDZ1 und RAIDZ2 sind nicht automatisch die beste Wahl für Storage mit vielen virtuellen Maschinen.

Bei I/O-intensiven VMs sind Mirrors häufig interessanter. RAID-Z eignet sich dagegen besonders für Workloads, bei denen Kapazität und sequenzielle Zugriffe wichtiger als maximale IOPS sind.

Achtung: Bei ZFS RAIDZ1/RAIDZ2 können Blockgröße, ZVOL-Konfiguration und der Workload erheblichen Einfluss auf Speicherbedarf und Performance haben. Eine unpassende Kombination kann zu deutlichem Overhead führen.

Insbesondere die Einstellungen von **volblocksize**, ZFS-Pool, RAID-Z-Geometrie und VM-Workload sollten deshalb gemeinsam betrachtet werden.

Proxmox Updates und Repositories prüfen

Bei Proxmox VE sollte geprüft werden, welche **Proxmox-Repositories** tatsächlich konfiguriert sind.

Je nach Subscription und gewünschtem Update-Kanal stehen unterschiedliche Repositories zur Verfügung. Wer keine Subscription verwendet, muss darauf achten, dass ein geeigneter Proxmox-No-Subscription-Kanal korrekt konfiguriert ist.

Es reicht nicht aus, lediglich die zugrunde liegenden Debian-Paketquellen zu aktualisieren und davon auszugehen, damit automatisch sämtliche Proxmox-Komponenten aktuell zu halten.

SSH mit Public-Key-Authentifizierung absichern

Für administrative SSH-Zugänge sollte nach Möglichkeit **Public-Key-Authentifizierung** anstelle eines reinen Passwort-Logins verwendet werden.

Zusätzlich sollte geprüft werden, ob Passwort-Login für administrative Benutzer vollständig deaktiviert werden kann.

Proxmox Webinterface mit Zwei-Faktor-Authentifizierung schützen

Ein administrativer Zugang zum **Proxmox VE Webinterface ohne Zwei-Faktor-Authentifizierung (2FA)** erhöht unnötig das Risiko eines Account-Missbrauchs.

Für administrative Konten empfehlen wir daher grundsätzlich die Einrichtung eines zweiten Faktors.

Bootloader und alte Proxmox-Installationen prüfen

Bei älteren Proxmox-Installationen sollte bekannt sein, ob das System über **GRUB oder systemd-boot/EFI** gestartet wird und wie die Boot-Partitionen verwaltet werden.

Insbesondere nach Upgrades über mehrere Proxmox-Versionen sollte kontrolliert werden, ob Kernel, EFI-Systempartitionen und Bootloader korrekt gepflegt werden.

ZFS nicht nur über die Proxmox GUI überwachen

Die Proxmox-Oberfläche allein reicht unserer Erfahrung nach nicht aus, um den Zustand eines produktiven **ZFS-Storage** vollständig zu beurteilen.

Zusätzlich sollten regelmäßig unter anderem folgende Bereiche überwacht werden:

- ZPOOL-Status
- SMART-Werte der Datenträger
- Kapazität und tatsächliche Pool-Auslastung
- ZFS-Fehler und Checksum Errors

- Scrub-Ergebnisse
- SSD-Verschleiß
- Temperaturen
- ARC und Speicherverbrauch

SSD TRIM konfigurieren und überwachen

Bei SSD-basierten ZFS-Systemen sollte die verwendete **TRIM-Strategie** bewusst festgelegt und überprüft werden.

Ob kontinuierliches ZFS-Autotrim oder ein zeitgesteuerter Trim sinnvoller ist, hängt auch von der eingesetzten Hardware und dem Workload ab.

Scrub und TRIM nicht mit Backup-Zeiten kollidieren lassen

Automatische **ZFS-Scrubs, TRIM-Vorgänge und Backups** sollten zeitlich aufeinander abgestimmt werden.

Bei größeren Storage-Systemen können diese Prozesse erhebliche I/O-Last erzeugen. Sie sollten daher möglichst nicht gleichzeitig mit großen Backups oder besonders I/O-intensiven produktiven Arbeiten laufen.

ZFS ARC bewusst dimensionieren

Der **ZFS ARC** verwendet Arbeitsspeicher als Cache. Die optimale Größe hängt stark vom Einsatzzweck des Servers ab.

Ein Proxmox-Host mit vielen virtuellen Maschinen benötigt eine andere Speicheraufteilung als ein dediziertes Storage-System.

Wir empfehlen deshalb, den ARC nicht einfach aufgrund pauschaler Prozentwerte zu dimensionieren, sondern anhand von RAM-Ausstattung, VM-Bedarf, Storage-Größe und tatsächlichem Workload festzulegen und anschließend zu überwachen.

Ausfall einer Proxmox Boot-Disk vorbereiten

Bei gespiegelten Proxmox-Bootlaufwerken reicht es im Fehlerfall nicht immer aus, lediglich die defekte Festplatte physisch zu ersetzen.

Die neue Disk muss passend **partitioniert, in den ZFS-Pool eingebunden und gegebenenfalls wieder bootfähig gemacht** werden.

Der Ablauf für einen Boot-Disk-Ausfall sollte deshalb bereits vor dem Ernstfall dokumentiert und getestet sein.

/etc/pve zusätzlich sichern

Das Verzeichnis `/etc/pve` ist kein gewöhnliches lokales Konfigurationsverzeichnis. Proxmox stellt die Cluster-Konfiguration über **pmxcfs** bereit.

Wichtige Proxmox-Konfigurationen sollten deshalb zusätzlich **regelmäßig und versioniert gesichert** werden.

KVM- und Remote-Konsolen regelmäßig testen

Remote-Konsolen sollten nicht erst getestet werden, wenn der Server nicht mehr über das Netzwerk erreichbar ist.

Wir empfehlen insbesondere:

- Tastaturbelegung und Sonderzeichen beim Login prüfen
- Passwörter mit Sonderzeichen testen
- Browser- und Client-Kompatibilität kontrollieren

- Remote-KVM bzw. IPMI/iLO/iDRAC regelmäßig testen

Eine jährliche Funktionsprüfung kann verhindern, dass die Notfallkonsole ausgerechnet im Ernstfall nicht verwendbar ist.

E-Mail und Benachrichtigungen testen

Ein Proxmox-System sollte wichtige Ereignisse und Fehler zuverlässig melden können.

Die konkrete Einrichtung des Mailversands hat sich über die verschiedenen Proxmox-Versionen weiterentwickelt. Entscheidend ist deshalb weniger die konkrete Versandmethode als ein **regelmäßig getesteter Benachrichtigungsweg**.

Testet insbesondere, ob Warnungen tatsächlich einen Administrator erreichen.

ZFS-Mountpoints und Directory Storage beachten

Vorsicht bei ZFS und Directory Storage: Ein Verzeichnis sollte nicht unkontrolliert als Storage verwendet werden, wenn an dieser Stelle eigentlich ein ZFS-Dataset gemountet werden soll.

Ist das Dataset nicht gemountet und wird das Verzeichnis trotzdem verwendet, können Daten versehentlich auf dem darunterliegenden Dateisystem landen. Mountpoints und Storage-Verfügbarkeit sollten deshalb überwacht werden.

Proxmox ZFS Replication ersetzt keine Backup-Historie

Die integrierte **Proxmox ZFS Replication** ist eine praktische Möglichkeit, VM-Daten auf einen anderen Proxmox-Node zu replizieren.

Eine Replikation ist jedoch **kein Ersatz für ein Backup mit Historie**. Fehler oder unerwünschte Änderungen können ebenfalls repliziert werden.

Für eine belastbare Datensicherung empfehlen wir zusätzlich ein separates Backup-System, beispielsweise **Proxmox Backup Server**.

ZFS Snapshots sinnvoll einsetzen

Regelmäßige **ZFS-Snapshots** sind eine sinnvolle Ergänzung für Storage und Wiederherstellung.

Snapshots benötigen zunächst wenig zusätzlichen Speicher, können bei vielen Änderungen jedoch erheblich wachsen. Die tatsächliche Pool-Auslastung muss deshalb überwacht werden.

ZFS-Pools nicht vollständig füllen: Für produktive Systeme sollte ausreichend freier Speicher vorhanden bleiben. Sehr hohe Pool-Auslastungen können Performance und administrativen Handlungsspielraum erheblich verschlechtern.

Ceph nur mit passender Infrastruktur einsetzen

Ceph ist eine leistungsfähige Lösung für verteilten Storage, aber kein Selbstzweck.

Für kleine Installationen kann ein sauber aufgebautes ZFS-Konzept einfacher, wirtschaftlicher und leichter zu betreiben sein.

Ceph erfordert Erfahrung sowie eine passende Anzahl an Nodes, Datenträgern, Netzwerkbandbreite und Monitoring. Vor einer Ceph-Einführung sollten deshalb Anforderungen und vorhandene Infrastruktur genau geprüft werden.

CPU-Typ virtueller Maschinen bewusst auswählen

Der CPU-Typ einer KVM-VM beeinflusst, welche Funktionen der physischen CPU innerhalb der virtuellen Maschine zur Verfügung stehen.

`host` kann eine hohe CPU-Feature-Abdeckung und Performance ermöglichen. Bei Clustern und insbesondere bei **Live-Migration zwischen unterschiedlichen CPU-Generationen** muss die Auswahl jedoch sorgfältig geplant werden.

Für Cluster kann ein gemeinsamer CPU-Typ sinnvoller sein als pauschal `host` für jede VM zu verwenden.

VirtIO- und Windows-Gasttreiber

Bei produktiven Windows-VMs empfehlen wir, **stabile und getestete VirtIO-Treiber** einzusetzen.

Neue Treiberversionen sollten vor einem flächendeckenden Rollout zunächst getestet werden.

Backups niemals ausschließlich auf dem Proxmox-Host speichern

Ein Backup auf demselben Server ist kein ausreichendes Backup-Konzept.

Backups sollten nicht ausschließlich auf `local` bzw. auf demselben physischen Proxmox-Host gespeichert werden, auf dem die produktiven VMs laufen.

Bei einem Defekt, Dateisystemproblem oder vollständigen Ausfall des Hosts wären ansonsten sowohl Produktivsystem als auch Backup betroffen.

Für produktive Umgebungen empfehlen wir einen getrennten **Proxmox Backup Server oder ein anderes physisch getrenntes Backup-Ziel**.

Video: Proxmox VE Praxistipps

Zu diesen Erfahrungen haben wir auch ein Video veröffentlicht:

Proxmox VE Praxistipps auf YouTube ansehen

Proxmox Tools und Postinstaller auf GitHub

Viele unserer Werkzeuge für Proxmox VE stellen wir als Open Source zur Verfügung. Dazu gehört unter anderem unser **Proxmox VE Post Install Script**.

bashclub – Proxmox Tools und Postinstaller auf GitHub

Stand der Dokumentation: Diese Seite ist aus einer älteren Sammlung unserer Proxmox-Praxiserfahrungen entstanden und wird inhaltlich weiterhin als technische Orientierung gepflegt. Da sich Proxmox VE, Debian, ZFS, Kernel, Bootloader und Hardwareunterstützung weiterentwickeln, sollten konkrete Befehle und versionsabhängige Einstellungen immer gegen die Dokumentation der eingesetzten Proxmox-Version geprüft werden.

Mailpiler auf Proxmox VE installieren – E-Mail-Archiv im LXC

Mailpiler auf Proxmox VE installieren – E-Mail-Archivierung im LXC-Container

Hinweis: Diese Anleitung beschreibt die Installation von **Mailpiler** mit der **zamba-lxc-toolbox** auf **Proxmox VE**. Für die Installation wird hier der `dev`-Branch der Toolbox verwendet. Dieser kann Änderungen enthalten, die noch nicht Bestandteil einer stabilen Version sind.

Mit **Mailpiler** lässt sich eine zentrale Open-Source-Lösung zur **E-Mail-Archivierung** aufbauen. In dieser Anleitung installieren wir Mailpiler automatisiert als **LXC-Container auf Proxmox VE**.

Die Installation erfolgt mit der **zamba-lxc-toolbox** aus dem bashclub-Projekt. Nach der Grundinstallation konfigurieren wir unter anderem **SMTP, LDAP, IMAP, TLS-Zertifikate, Mailcow und Postfix**.

Autor: Thorsten Spille

Mailpiler mit der zamba-lxc-toolbox installieren

Git auf dem Proxmox-Host installieren

Falls Git auf dem Proxmox-Host noch nicht vorhanden ist:

```
apt update
apt -y install git
```

zamba-lxc-toolbox herunterladen

Anschließend wird der `dev`-Branch der zamba-lxc-toolbox geklont:

```
git clone -b dev https://github.com/bashclub/zamba-lxc-toolbox
cd zamba-lxc-toolbox
```

Für Mailpiler erstellen wir zunächst eine eigene Konfigurationsdatei:

```
cp conf/zamba.conf.example conf/piler.conf
```

Anschließend wird `conf/piler.conf` mit dem bevorzugten Texteditor angepasst. Für eine einfache Installation müssen insbesondere die mit `LXC_` beginnenden Variablen geprüft und angepasst werden.

DNS und FQDN beachten: Der Mailpiler-Hostname wird aus dem FQDN des Containers erzeugt. Der FQDN sollte deshalb sowohl vom Proxmox-Host als auch von den späteren Clients korrekt über DNS auflösbar sein.

Falls der DHCP-Server ein DNS-Suffix verteilt, empfehlen wir für den Mailpiler-Container eine passende **statische IP- und DNS-Konfiguration**.

Mailpiler LXC-Container erstellen

Nach der Konfiguration kann die Installation gestartet werden:

```
bash install.sh -c conf/piler.conf [-s piler] [-i <id>]
```

Optional können der gewünschte Service `piler` und eine bestimmte LXC-Container-ID angegeben werden.

Nach erfolgreicher Installation ist das **Mailpiler Webinterface über den FQDN des Containers** erreichbar. Voraussetzung ist eine funktionierende DNS-Auflösung.

Mailpiler Erstkonfiguration

Standard-Passwörter sofort ändern

Wichtig: Nach der Installation müssen die Standard-Zugangsdaten unverzüglich geändert werden. Insbesondere sollte ein System mit Standard-Passwörtern niemals öffentlich erreichbar sein.

Die Installation verwendet zunächst folgende Standard-Logins:

```
admin@local
auditor@local
```

Die Passwörter werden im Mailpiler Webinterface unter **Administration → Users → Edit/View** für den jeweiligen Benutzer geändert.

SMTP, LDAP und IMAP konfigurieren

Die zentrale Konfigurationsdatei befindet sich im LXC-Container unter:

```
/etc/piler/config-site.php
```

Dort sind bereits Konfigurationsbereiche für verschiedene Dienste und Authentifizierungsmethoden vorbereitet:

- **SMTP / Smarthost** – optional mit Authentifizierung
- **IMAP** – Benutzeranmeldung am Mailarchiv über einen IMAP-Server
- **LDAP** – Benutzeranmeldung über ein LDAP-Verzeichnis
- **UCS LDAP** – Anbindung an Univention Corporate Server
- **POP3** – Benutzeranmeldung über einen POP3-Server

Offizielle Mailpiler-Dokumentation:

- **Authentifizierung über IMAP**
- **Authentifizierung über LDAP**
- **Authentifizierung über POP3**

TLS-Zertifikat für Mailpiler einrichten

Nach dem Ausrollen des Containers sind die Piler-Dienste zunächst mit einem **Snakeoil-Zertifikat** aus dem Paket `ssl-cert` konfiguriert.

Für den produktiven Betrieb sollte dieses Zertifikat durch ein gültiges **TLS-Zertifikat** ersetzt werden. Hierfür kann beispielsweise ein Zertifikat aus der eigenen PKI oder ein **Let's-Encrypt-Zertifikat** verwendet werden.

Verwendete Zertifikatsdateien

Dienst	Datei
nginx Zertifikatskette	`/etc/nginx/ssl/fullchain.pem`
nginx Private Key	`/etc/nginx/ssl/privkey.pem`
Mailpiler	`/etc/piler/piler.pem`

Mailpiler Zertifikat austauschen

Zunächst werden die vorhandenen Links auf die Snakeoil-Zertifikate entfernt:

```
unlink /etc/nginx/ssl/fullchain.pem
unlink /etc/nginx/ssl/privkey.pem
```

Anschließend werden die neuen Zertifikatsdateien kopiert:

- `/etc/nginx/ssl/fullchain.pem` – vollständige Zertifikatskette, `root:root 0640`
- `/etc/nginx/ssl/privkey.pem` – Private Key, `root:root 0600`

Danach wird ausgeführt:

```
renew-piler-cert
```

Das Skript beendet die entsprechenden Dienste, erzeugt auf Grundlage des nginx-Zertifikats das benötigte Piler-Zertifikat und startet die Dienste anschließend neu.

Bestehende E-Mails über POP3 oder IMAP importieren

Bereits vorhandene Postfächer können für einen initialen **E-Mail-Import in Mailpiler** über das Webinterface konfiguriert werden:

Administration → Import

Dort können entsprechende POP3- oder IMAP-Postfächer für den Import hinterlegt werden.

Mailempfang für das Archiv konfigurieren

Mailpiler verfügt über einen integrierten SMTP-Dienst, der die zu archivierenden Nachrichten auf **TCP-Port 25** entgegennimmt.

Der produktive Mailserver muss den Mailpiler-Server daher über SMTP erreichen können.

SMTP ACL für Mailpiler

Besonders wichtig bei öffentlich erreichbarem SMTP: Der Zugriff auf den SMTP-Dienst des Archivs sollte auf die tatsächlich berechtigten Mailserver bzw. Netze eingeschränkt werden.

Hierfür kann die Datei:

```
/etc/piler/smtp.acl
```

angelegt werden. Dort werden erlaubte IP-Netze im CIDR-Format definiert.

Beispiel:

```
# Beispiel Microsoft 365
40.92.0.0/15 permit
40.107.0.0/16 permit
52.100.0.0/14 permit
104.47.0.0/17 permit
```

Hinweis: IP-Netze externer Anbieter wie Microsoft 365 können sich ändern. Die aktuell gültigen Netze sollten deshalb vor der Übernahme anhand der Dokumentation des jeweiligen Anbieters geprüft werden.

Nach Fertigstellung der ACL wird diese aktiviert und der SMTP-Dienst neu gestartet:

```
sed -i -e "s|smtp_access_list=.*|smtp_access_list=1|g" /etc/piler/piler.conf
systemctl restart piler-smtp
```

Mailserver für die E-Mail-Archivierung konfigurieren

Nach der Installation von Mailpiler muss der produktive Mailserver so konfiguriert werden, dass die zu archivierenden Nachrichten zusätzlich an den **Mailpiler Archivserver** übermittelt werden.

Ein mögliches Ziel ist beispielsweise:

```
archive@<fqdn-des-archivservers>
```

Wie diese Weiterleitung bzw. Journalisierung eingerichtet wird, hängt vom eingesetzten Mailserver ab.

Mailpiler mit Postfix

Bei **Postfix** kann eine Kopie ein- und ausgehender Nachrichten über entsprechende BCC- bzw. Recipient-Konfigurationen an Mailpiler übertragen werden.

Da sich die konkrete Postfix-Konfiguration nach Umgebung und gewünschtem Archivierungsumfang richtet, empfehlen wir die jeweils aktuelle Mailpiler-Dokumentation:

Mailpiler – Installation und Mailserver-Konfiguration

BCC-Adressen beachten: Bei bestimmten Postfix-Konfigurationen können zusätzliche Envelope-Informationen sichtbar werden. Mailpiler bietet mit `HEADER_LINE_TO_HIDE` eine Möglichkeit, entsprechende Header für normale Benutzer auszublenden.

Mailpiler mit Mailcow

Auch **Mailcow** lässt sich mit Mailpiler als zentralem E-Mail-Archiv kombinieren.

Im Mailcow Webinterface können unter:

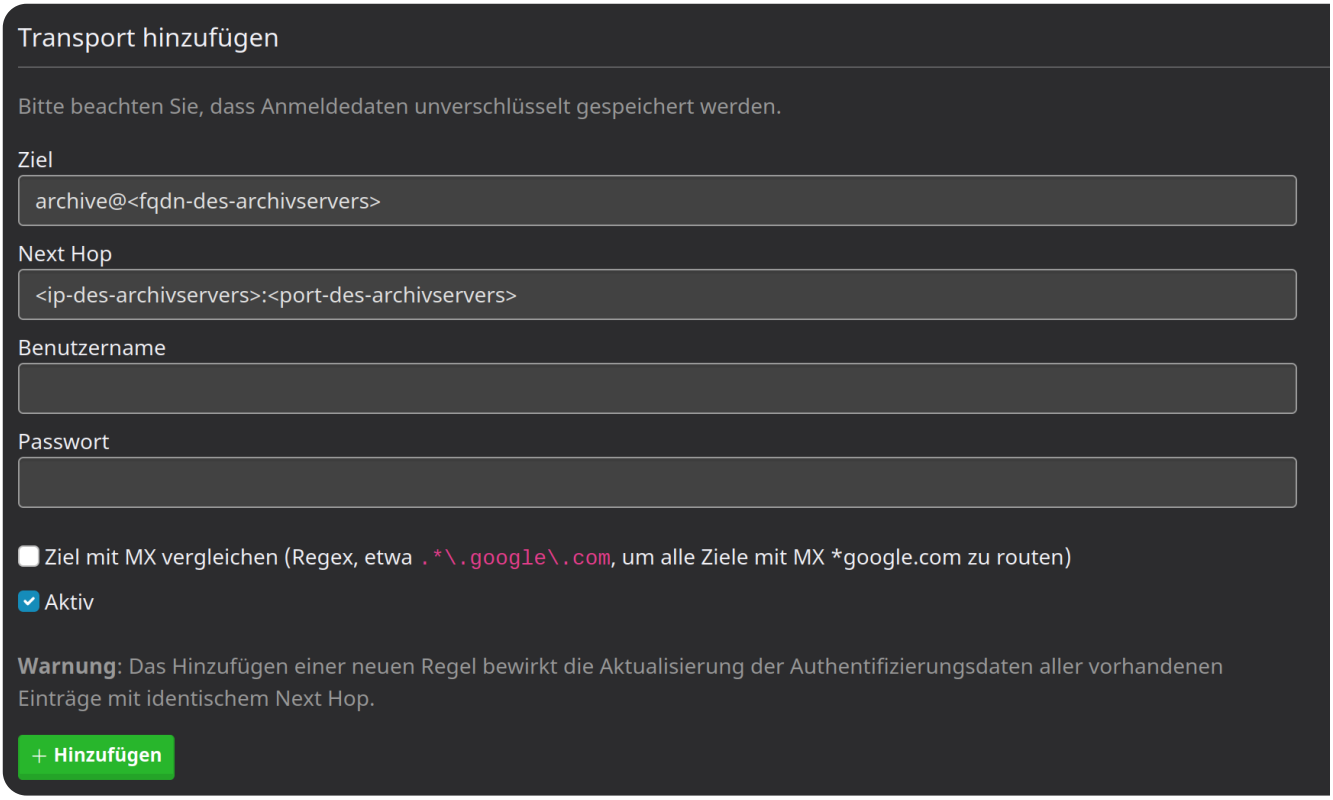
E-Mail → Konfiguration → Addressumschreibung

für die gewünschten Domains BCC-Zuordnungen für eingehende und ausgehende Nachrichten definiert werden.

Ist der Archivserver nicht über DNS erreichbar, aber über seine IP-Adresse verfügbar, kann zusätzlich unter:

System → Konfiguration → Routing

ein entsprechendes Transport-Mapping eingerichtet werden.



Exchange und weitere Mailserver

Bei anderen Mailservern unterscheidet sich die Konfiguration entsprechend. Bei Microsoft Exchange bzw. Microsoft-Umgebungen kann beispielsweise **Journaling** zum Einsatz kommen.

Weitere Informationen befinden sich in der offiziellen Dokumentation:

Mailpiler Dokumentation

Mailpiler Pfade im LXC-Container

Die Installation über die zamba-lxc-toolbox verwendet unter anderem folgende Pfade:

Funktion	Pfad
Konfiguration	`/etc/piler`
Cronjobs	`/etc/cron.d/piler`
CLI-Tools	`/usr/bin`

Funktion	Pfad
Hauptdienst	`/etc/systemd/system/piler.service`
Suchdienst	`/etc/systemd/system/pilersearch.service`
SMTP-Dienst	`/etc/systemd/system/piler-smtp.service`
WebUI-Skripte	`/usr/libexec/piler`
Datenverzeichnis	`/var/piler`

ZFS Dataset für Mailpiler

Das Datenverzeichnis und Home-Verzeichnis des Piler-Benutzers befindet sich unter:

```
/var/piler
```

In unserer Installation wird der LXC-Mountpoint `mp0` als separates **ZFS-Dataset** auf `/var/piler` eingebunden.

Das Dataset verwendet eine für diesen Einsatzzweck angepasste `recordsize=16K`.

Die optimale ZFS-Konfiguration hängt von Storage, Mailaufkommen und Workload ab. Die Einstellung `recordsize=16K` sollte daher als Konfiguration dieser Installation und nicht als allgemeingültige Vorgabe für jedes Mailpiler-System verstanden werden.

Support und Autor

Diese Mailpiler-Integration für die zamba-lxc-toolbox wurde dokumentiert von:

Thorsten Spille
Spille IT Solutions

thorsten@spille-edv.de

Wer die Arbeit unterstützen möchte:

Thorsten Spille auf Ko-fi unterstützen

Technischer Hinweis: Mailpiler, Proxmox VE, die zamba-lxc-toolbox und die angebundenen Mailserver werden kontinuierlich weiterentwickelt. Prüft deshalb insbesondere bei produktiven Mailarchiven die jeweils aktuelle Dokumentation sowie die Kompatibilität der eingesetzten Versionen.