LXC Toolbox für Proxmox

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

- Zamba Fileserver
- Die schlimmsten Fehler im Umgang mit Proxmox VE und ZFS
- Piler Mailarchiv (>= 1.4.5)

Zamba Fileserver

Trojanersicherer Fileserver mit und ohne Active Directory Integration auf Proxmox VE Systemen mit ZFS

Proxmox bietet seit Version 3.4 eine ausgezeichnete Unterstützung für das ZFS Dateisystem / Filestorage an.

Installationen die mittels Hostbusadaptern oder lokalen SATA Controllern, wie auch USB Raids durchgeführt wurden sind in der Lage Dateisystem und Raid in einem Kernel zu betreiben.

Mittels der Software zfs-auto-snapshot generiert das System alle 15 Minuten einen Snapshot, also einen eingefrorenen Zustand des Dateisystems.

Die Snapshots greifen für alle Datasets (Dateisystem) und ZVOLs (virtuelle Disks).

Zu beachten ist hier nur dass das System niemals über 80% Auslastung kommt, da ZFS dann anfängt bis ca. 95% immer härter zu drosseln, was einen Überlauf verhindern soll.

Das Snapshotten und die Zerstörung von Snapshots

ZFS bietet die Möglichkeit mittels Verzeichnisstruktur auf diese Snapshots zuzugreifen, sprich dort auf ältere Dateien zuzugreifen.

Voraussetzung Nr.1 – Proxmox VE System V6/7 mit installiertem zfs-auto-snapshot

~# apt install zfs-auto-snapshot

Danach finden sich Cronjobs unter

/etc/cron.d/zfs-auto-snapshot (viertelstündig mal vier als Standard)

/etc/cron.hourly/zfs-auto-snapshot (stündlich mal 24 als Standard)

/etc/cron.daily/zfs-auto-snapshot (täglich mal dreissig als Standard)

/etc/cron.weekly/zfs-auto-snapshot (wöchentlich mal acht als Standard)

/etc/cron.monthly/zfs-auto-snapshot (monatlich mal zwölf als Standard)

Damit wir unter den 80% bleiben empfehlen wir erst mal

/etc/cron.monthly/zfs-auto-snapshot

auf zwölf Monate zu reduzieren.

Besonders komfortabel geht das mit unserem Postinstallskript

https://github.com/bashclub/proxmox-zfs-postinstall

Die Prüfung auf die Dateibelegung geht wie folgt:

NAME SIZE ALLOC FREE CKPOINT EXPANDSZ FRAG CAP DEDUP HEALTH ALTROOT

rpool 1.73T 501G 1.24T - - 44% **28%** 1.00x ONLINE -

Dieses System ist mit 28% im grünen Bereich

Als komfortable Erweiterung für ZFS per Webgui könnte man auch noch Cockpit mit ZFS installieren

wget https://raw.githubusercontent.com/bashclub/proxmox-zfs-postinstall/main/install-cockpit-zfs-manager

bash install-cockpit-zfs-manager

diesen erreicht man danach unter https://IPVOMPVE:9090

Bitte daran denken dass Cockpit nichts vom PVE weiß und anders herum.

Vor einem Rollback auf einen alten Stand einfach VM/LXC ausschalten.

Nun zum Fileserver

Die LXC-Toolbox ist das Schweizer Taschenmesser von sysops.tv.

Darüber wird einmalig eine best Practice Installation verschiedener Dienste ausgeführt.

Diese Systeme werden danach manuell durch den Admin weiter gepflegt.

https://github.com/bashclub/zamba-lxc-toolbox

So installieren Sie den besten Fileserver mit ZFS in wenigen Minuten

apt update

apt -y install git

Wir klonen das Repository nach /root

git clone https://github.com/bashclub/zamba-lxc-toolbox

cd zamba-lxc-toolbox

Die Konfigurationsvorlage wird kopiert, so kann man später die Standardwerte sehen.

cp conf/zamba.conf.example conf/zamba.conf

Wir passen die Konfigurationsdatei an

nano conf/zamba.conf

Dort werden folgende Informationen benötigt um einen AD-Member-Server zu installieren, also falls abweichend bitte anpassen

LXC_TEMPLATE_STORAGE="local" #ist der Dateispeicher für die Linuxvorlage

LXC_ROOTFS_SIZE="32"# Größe des Linuxcontainer Betriebssystems, kann jederzeit angepasst werden, da nur Limit

LXC_ROOTFS_STORAGE="local-zfs" #Standard Speicherort von PVE mit ZFS rpool/data

LXC_SHAREFS_SIZE="100" #Größe des Fileserverbereichs, kann jederzeit angepasst werden, da nur Limit

LXC_SHAREFS_STORAGE="local-zfs" #Speicherort für Fileserverbereich

LXC_MEM="1024" #RAM für Fileserver, hier ggf. mal auf 2048 oder 4096 bei größeren Systemen. Im Betrieb selten mehr als 100MB benötigt, jedoch als Cache kann es knapp werden

LXC_HOSTNAME="${service}" #ggf. nach dem = eigenen Namen für PVE vergeben, empfohlen wäre hier der hostname im netz, also zmb oder fs oder wunschname

LXC_DOMAIN="zmb.rocks" #Der Domänensuffix für PVE, hier empfohlen Windows DNS domain, also z. B. sysops.local

LXC_IP="192.168.100.200/24" #feste freie IP im LAN

LXC_GW="192.168.100.254" #Router im Netz, bitte ausgehend 80,443 und 123 erlauben

LXC_DNS="192.168.100.254" #Hier „muss“ ein Domaincontroller rein, damit die Windows Domäne aufgelöst werden kann.

LXC_BRIDGE="vmbr0"# die virtuelle Brücke ins LAN

LXC_PWD='Start!123'# Passwort für den Container Linux Seite zur Wartung

############### Zamba-Server-Section ###############

Bereich für den Fileserverdienst, die anderen Bereiche nicht ausfüllen.

ZMB_REALM="ZMB.ROCKS"#Windows DNS Name (Großbuchstaben sind Pflicht!).

ZMB_DOMAIN="ZMB" #Windows Netbios Name (Großbuchstaben sind Pflicht!).

ZMB_ADMIN_USER="administrator"#User für die Aufnahme in Domäne. Falls Fehler bei Installation Groß- und Kleinschreibung des Users beachten.

ZMB_ADMIN_PASS='Start!123' #Passwort des ZMB_ADMIN_USER in der Active Directory

###

Speichern Sie die Datei

Installation des Systems, mehrfach möglich!

bash install.sh

Das System legt einen LXC in der GUI sichtbar an, startet ihn und installiert die Software.

Am Ende sehen Sie als Erfolgskontrolle alle User und Gruppen.

Falls Fehler beim User angemahnt werden die Groß- und Kleinschreibung des ZMB_ADMIN prüfen.

Löschen einer Fehlgeschlagenen Installationen

pct stop LXCNUMMER

pct destroy LXCNUMMER

Kontrolle Funktion

pct enter LXCNUMMER

wbinfo -u && wbinfo -g

Ausgabe aller User und Gruppen

Reparatur gestörte Domänenmitgliedschaft

host sysops.local # sollte auf Domaincontroller verweisen

kinit -V administrator #gefolgt von Passworteingabe

net ads join -U administrator createcomputer=Computers #gefolgt von Passworteingabe

Ab hier verhält sich der Zamba Fileserver wie ein Windows Server

#! Über den Explorer können Sie nun auf alle Snapshots als Vorgängerversion wiederherstellen zugreifen!!!

So setzen Sie Berechtigungen

a) Wie gewohnt über Explorer

b) Deutlich schneller via Kommandos im LXC

Löschen alle erweiterten Rechte

setfacl -Rb /tank/share

Zugriff setzen für einen Anwender plus Unterordner

setfacl -Rm u:administrator:rwx /tank/share/administrator # Rechte

setfacl -Rdm u:administrator:rwx /tank/share/administrator # Standard für neue Ordner und Dateien, wichtig!!!

Zugriff setzen für eine Gruppe plus Unterordner

setfacl -Rm g:verwaltung:rwx /*tank*/share/verwaltung

setfacl -Rdm g:verwaltung:rwx /*tank*/share/verwaltung

Weitere Videos:

https://www.youtube.com/watch?v=HP3zaRnNGLE

https://www.youtube.com/watch?v=yFN9Ykr7s5I&t=732s #englisch, more recent

Kurse zum Thema

https://cloudistboese.de #ZFS für Firmen

Die schlimmsten Fehler im Umgang mit Proxmox VE und ZFS

- Vergessen Sie nicht die Seriennummern der Platten in den Einschüben zu notieren, sonst muss man im dümmsten Fall den Server herunterfahren bevor man sicher eine Disk tauschen kann
- LVM bringt im Vergleich zu ZFS nur Nachteile und sollte wegen dem gernigen Leistungsumfang vermieden werden
- Hardware Raid Controller als HBA missbrauchen und einzelne Platten als Raid0 definieren im ZFS Verbund führt zu Datenverlust, nachweislich!
- Raid 5/6, bzw. ZFS RaidZ1+2 ist für virtuelle Maschinen nur für lazy Daten geeignet
- ZFS im Einsatz mit RaidZ1+Z2 führt bei einer Blockgröße von default 8k auf local-zfs Datastore zu einem Overhead und Verlangsamung von 50-100%
- Updates ohne Subskription oder PVE-NO-SUBSCRIPTION Maßnahme macht keine Proxmox, lediglich Ubuntu Updates
- SSH Passwort Login sollte durch Public Key Login ersetzt werden
- Weblogin ohne 2fa ist riskant
- alte Installation aktualisieren den Kernel nicht mehr auf der nicht gebooteten Version, also Efi Boot macht nur noch EFI Updates, Grub Boot macht nur noch Grub Updates!
- ZFS ist zu komplex um es nur über die PVE Konsole auf Platz und Funktion zu prüfen, was auch für Smart gilt
- Automatisches Trimming der SSDs ist dem Wöchentlichen vorzuziehen
- Die wöchentlichen Trimming und Scrub Jobs sollten auf Verträglichkeit mit Arbeit und Backups geprüft werden
- Der primäre ZFS Cache ARC sollte via unserem Postinstaller manuell festgelegt werden, da 50% zu viel für VMs ist und für SAN zu wenig
- Bootfestplatten müssen bei Ausfall vorpartitioniert und mit Boot-Tool und ZFS gefüllt werden
- /etc/pve läuft aus einer Datenbank und sollte versioniert gesichert werden
- KVM Konsolen sollten auf Sonderzeichen beim Login geprüft werden
- KVM Konsolen sollten jährlich auf Clientkompatibilität geprüft werden
- Mailversand muss mühselig über Postfix eingerichtet werden, ist nicht zuverläsig
- ZFS Directory Datastores sind zu vermeiden, da PVE im dümmstem fall den Ordner schneller anlegt als ZFS. Dann erscheint der Ordner leer
- Die eingebaute ZFS Replikation ist simpel und baut keine Historie auf

- ZFS-Auto-Snapshot unbedingt nutzen, jedoch nicht 80% Plattenplatz vom ZPOOL überschreiten
- CEPH ist nichts für Anfänger und kleine Firmen
- Im Idealfall CPU Typ Host in VMs auswählen, da sonst nicht alle Funktionen der CPU zur Verfügung stehen
- Guest Treiber immer nur 'stable' verwenden, nicht 'latest'
- Backups niemals auf local Store ausführen, da dies der PVE ist
-

Unser Video zum Beitrag

https://www.youtube.com/watch?v=R-1Z6iqRNr0

Unser Github Repo mit z. B. dem PVE Postinstaller

https://github.com/bashclub

Piler Mailarchiv (>= 1.4.5)

Installation via zamba-lxc-toolbox (dev) auf Proxmox VE

Autor: Thorsten Spille

Supporte mich mit einem Kaffee:

git installieren (einmalig)

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

Toolbox klonen und Piler LXC-Container erstellen

```
git clone -b dev https://github.com/bashclub/zamba-lxc-toolbox
cd zamba-lxc-toolbox
# Konfiguration erstellen
cp conf/zamba.conf.example conf/piler.conf
# Mit favorisiertem Texteditor die Datei conf/piler.conf anpassen
```

```
# Hier ist es nur notwendig, Variablen beginnend mit LXC_ zu bearbetien
# Wird vom DHCP Server im Netzwerk ein DNS-Suffix mitgegeben, so sollte der Piler Hostname eine statische IP-Konfiguration verwendet werden
# Der Piler Hostname wird aus dem FQDN Des Containers erzeugt
# Nach Anpassung der conf/piler.conf den Container erzeigen:
bash install.sh -c conf/piler.conf [-s piler] [-i <id>]
# Optional kann man den Container-Typ / Service angeben und die gewünschte LXC Container ID
```

Nach erfolgter Installation ist die Mailpiler Website über den FQDN des Containers erreichbar, dabei ist wichtig, dass der FQDN des Containers vom Client auflösbar ist.

Erstkonfiguration

Passwörter ändern (Web UI)

Standard-Logins:
- admin@local:pilerrocks
- auditor@local:auditor

Die Passwörter der Standard-Benutzer sollten auf sichere Passwörter geändert werden unter *administration -> users -> Edit/view* für den jeweiligen Benutzer.

SMTP (Smarthost), LDAP und IMAP Login Settings (CLI via pct enter oder ssh)

In der Datei */etc/piler/config-site.php* gibt es bereits vorbereitete Konfigurationsblöcke für Logins und Mailversand:
- SMTP (Smarthost mit optionaler Authentifizierung)
- IMAP (Serverdaten für User-Login am Mailarchiv)

  https://www.mailpiler.org/authenticating-against-an-imap-server/
- LDAP (für User-Login am Mailarchiv)

  https://www.mailpiler.org/authenticating-against-an-ldap-directory/
- UCS LDAP (für User-Login am Mailarchiv)
- POP3 (User-Login)

https://www.mailpiler.org/authenticating-against-a-pop3-server/

TLS-Zertifikat

Nach dem Ausrollen des Containers sind die piler Dienste mit dem snakeoil Zertifikat (Paket ssl-cert) konfiguriert. Dieses sollte durch ein Zertifikat aus einer zugehörigen PKI oder mit einem letsencrypt Zertifikat ersetzt werden, damit es für den Client signiert ist.

Dateien:
nginx: */etc/nginx/ssl/fullchain.pem* und */etc/nginx/ssl/privkey.pem* sind jeweils Links auf das Snakeoil Zertifikat
piler: */etc/piler/piler.pem* = Kopie des Snakeoil Zertifikats, Zertifikatkette und Private Key in einer Datei

Austausch des Zertifikats

1. Links von nginx Zertifikat entfernen

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

2. Zertifikat (komplette Kette nach */etc/nginx/ssl/fullchain.pem,* root:root 0640) und Private Key (nach */etc/nginx/ssl/privkey.pem*, root:root 0600) auf das Mailpiler-System (z.B. via SFTP) kopieren.
3. *renew-piler-cert* ausführen (beendet Dienste, erzeugt auf Basis des nginx Zertifikats ein piler Zertifikat, startet Dienste)

POP-/IMAP-Import via WebUI

POP-/IMAP-Postfächer können für den initialen Import via WebUI unter *administration - import* konfiguriert werden.

Mailempfang vorbereiten

Auf dem Mailpiler läuft ein integrierter SMTP-Server, der Mails für das Archiv auf TCP-Port 25 annimmt. Es muss dafür gesorgt werden, dass der Mailserver das Archiv erreichen kann.

SMTP Acl List

Ist der SMTP des Mailarchivs öffentlich erreichbar, sollte eine ACL mit erlaubten Mailservern für den Mailempfang am Archiv definiert werden. Hierzu wird die Datei */etc/piler/smtp.acl* erstellt und darin IP-Netze im CIDR Format und einem permit / reject dehinter. Beispiel:

```
# MS 365 Server
40.92.0.0/15 permit
40.107.0.0/16 permit
52.100.0.0/14 permit
104.47.0.0/17 permit
```

Ist die Liste vollständig, wird die Datei gespeichert, die ACL aktiviert und der piler-smtp neu gestartet:

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

Konfiguration des Mailservers

Auf dem Mailserver sollte nun BCC-Maps definiert werden, damit alle eingehende Mails (der zu archivierenden Domains) an die Adresse archive@<fqdn-des-archivservers> gesendet werden.

Postfix

Ein Beispiel ist in der Installationsanleitung zu finden: https://www.mailpiler.org/installation/

To archive emails, piler must receive them somehow. So you have to configure your mail server to send a copy of each received emails to piler via smtp. Since piler is actually an SMTP server, you should not put postfix, exim, ... on the archive itself. If you need it for some reason, then put it to 127.0.0.1:25/tcp, and set the listen_addr variable in piler.conf to listen on eth0 or similar.

If you have MS Exchange, then turn on journaling.

If you have postfix (including zimbra), then add the following to main.cf:

/etc/postfix/main.cf:

```
  smtpd_recipient_restrictions = reject_non_fqdn_recipient, ..., \
                check_recipient_access pcre:$config_directory/x-add-envelope-to, ...

  always_bcc = archive@piler.yourdomain.com
```

/etc/postfix/x-add-envelope-to:

```
  /(.*)/   prepend X-Envelope-To: $1
```

Note that such configuration might reveal Bcc addresses to the recipients in the To/Cc fields. To prevent it happening piler features the HEADER_LINE_TO_HIDE config.php variable to automatically hide the X-Envelope-To: line.

When set (and the default is as seen below) it will hide such header lines from regular users on the GUI, only auditors are allowed to see all recipients, including the Bcc addresses.

```
$config['HEADER_LINE_TO_HIDE'] = 'X-Envelope-To:';
```

If you have Exim, then add the following at the beginning of the routers-section:

```
begin routers

mailarchive:
  debug_print = "R: mailarchive for $local_part@$domain"
  driver = manualroute
  domains = *
  transport = remote_smtp
  # piler listening on port 25:
  route_list = * "piler.yourdomain.com::25"
  self = send

  unseen
```

Mailcow

Im Mailcow WebUI unter *E-Mail - Konfiguration - Addressumschreibung* können pro Domain BCC-Maps für eingehende und ausgehende Mails definiert werden.
Sollte eine DNS-Auflösung des Mailarchivs nicht verfügbar, das Archiv aber über die IP-Adresse erreichbar sein, so kann unter *System - Konfiguration - Routing* ein Transport Mapping angelegt werden.

Weitere Mailserver

Mehr gibts in der offiziellen Doku: https://www.mailpiler.org/docs.html

Angepasste Pfade im LXC-Container

Konfigurations-Verzeichnis: */etc/piler*
Piler-Cronjobs: */etc/cron.d/piler*
Piler CLI-Tools: */usr/bin* (ausführbar ohne Pfadangabe)
Piler-Dienste (systemd units):
- */etc/systemd/system/piler.service (Hauptprozess)*
- */etc/systemd/system/pilersearch.service (Suchdienst)*
- */etc/systemd/system/piler-smtp.service (Piler SMTP)*
Verzeichnis für vom WebUI genutzte Skripte: */usr/libexec/piler*

Datenverzeichnis und Home-Directory des piler Users: */var/piler*
Mountpoint *mp0* ist ein separates Dataset mit sparsamer *recordsize=16K*, welches auf */var/piler* eingehängt wird.

Supportkontakt

Spille IT Solutions

Thorsten Spille <thorsten@spille-edv.de>