

# SMB Share der NAS in einen Unprivileged LXC Container bringen

Nachdem ich zahlreiche Projekte mit Proxmox umgesetzt habe, ist es für mich die vielseitigste Virtualisierungsplattform überhaupt. Dank der großen Anzahl an Community-Skripten und der klaren Ausrichtung auf Home-Lab-Setups lässt sich mit Proxmox und etwas Geduld nahezu alles realisieren – von virtualisierten Entwicklungsumgebungen und produktiven Alltags-VMs bis hin zu vollständigen Hackintosh-Installationen.

Ein NAS habe ich bislang bewusst nicht auf meinen PVE-Systemen betrieben und Storage-Server stets von meinen Proxmox-Workstations getrennt gehalten. Nach mehreren Gesprächen mit einigen Menschen im IRC und XMPP ob es möglich sei einen virtualisierten NAS-Ansatz zu betreiben, kamen wir zu dem Schluss, dass es zwar geht, aber dennoch keine optimale Sache sei und es zum testen und ausprobieren ok, aber für Produktivsysteme nicht geeignet ist. Allerdings fanden wir bei der Diskussion eine elegante Möglichkeit, Netzwerkfreigaben aus dieser VM mit unprivilegierten LXCs zu verbinden – ohne den sonst üblichen, aufwendigen Terminal-Einsatz. Und darum geht es hier.

## Warum macht man das?

### Nützlich, wenn ausschließlich unprivilegierte LXCs zum Einsatz kommen

Zwischen meiner ausgeprägten Neigung, möglichst viele Serverkomponenten anzuschaffen, betreibe ich sowohl ein primäres als auch ein sekundäres NAS, wobei Letzteres für Storage-Experimente genutzt wird. Wer zum Testen kein separates NAS sein eigenen nennt, kann unter Proxmox auch ein OpenMediaVault oder ein TrueNAS in einer VM einrichten. Damit funktioniert es genauso gut.

## Überblick

Im Hinblick auf die Netzwerkfreigaben sieht mein Setup wie folgt aus: Auf meinem Test-NAS läuft eine SMB-Freigabe, die zunächst auf dem PVE-Host eingebunden wird. Anschließend wird per Terminal ein Mountpunkt auf den Node für den LXC eingerichtet, sodass die Freigabe dort wie ein lokales Laufwerk erscheint. Dieser Umweg ist notwendig, da unprivilegierte Container häufig

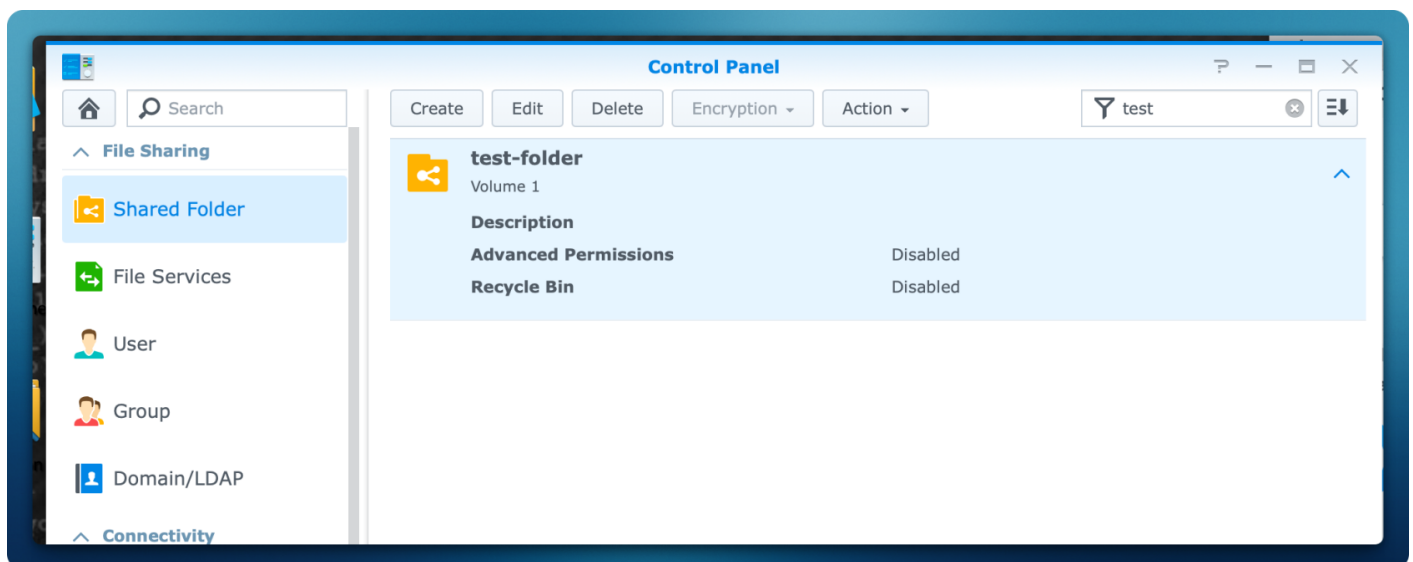
Berechtigungsprobleme beim direkten Einbinden von Netzwerkfreigaben verursachen. Aus Sicherheitsgründen sind unprivilegierte LXC's zwar vorzuziehen, sie erschweren jedoch genau dieses Szenario. Der beschriebene Ansatz umgeht diese Einschränkung und ermöglicht die Nutzung von SMB-Freigaben in sicheren, unprivilegierten Containern, ohne sich mit komplexen UID-/GID-Zuordnungen befassen zu müssen.

## Annahmen

Außerdem gehen wir davon aus, dass auf dem NAS ein Nutzer namens **"Testuser"** mit Passwort existiert und die Freigabe unter `smb://192.168.10.20/test-folder` erreichbar ist.

Außerdem verfügen wir über einen beliebigen LXC Container den wir zum Testen verwenden wollen. Ich habe hierzu einfach einen neuen LXC mit einer ID von 500 erstellt mit minimal CPU und RAM Angaben.

## Synology Nutzer und Share



**testuser**

**Info** | User Groups | Permissions | Quota | Applications | Speed Limit

Name \*:

Description:

Email:

Password \*:

Confirm password \*:

Disallow the user to change account password

Password is always valid

Disable this account

Immediately

After:

\* This field is required.

## Einbinden des Share in Proxmox Node

**PROXMOX** Virtual Environment 9.1.4

Server View | Datacenter

Search

▼ Datacenter

- ▼ pve
  - 100 (lxc-netbird)
  - 105 (lxc-npm)
  - 106 (lxc-adguard)
  - 108 (lxc-authelia)
  - 111 (lxc-homer)
  - 112 (lxc-uptimekuma)
  - 113 (lxc-myspeed)
  - 114 (lxc-pulse)
  - 120 (lxc-bookstack)
  - 121 (lxc-docmost)
  - 122 (lxc-paperless-ngx)
  - 123 (lxc-memos)
  - 124 (lxc-monica)
  - 125 (lxc-baikal)
  - 126 (lxc-jotty)
  - 130 (lxc-syncthing)
  - 131 (lxc-resiliosync)
  - 132 (lxc-gokapi)
  - 140 (lxc-immich)
  - 150 (lxc-restic-backups)
  - 207 (blocky)
  - 220 (lxc-homepage)

Search | Summary | Notes | Cluster | Ceph | Options | **Storage** | Backup | Replication | Permissions | Users | API Tokens | Two Factor | Groups | Pools | Roles | Realms

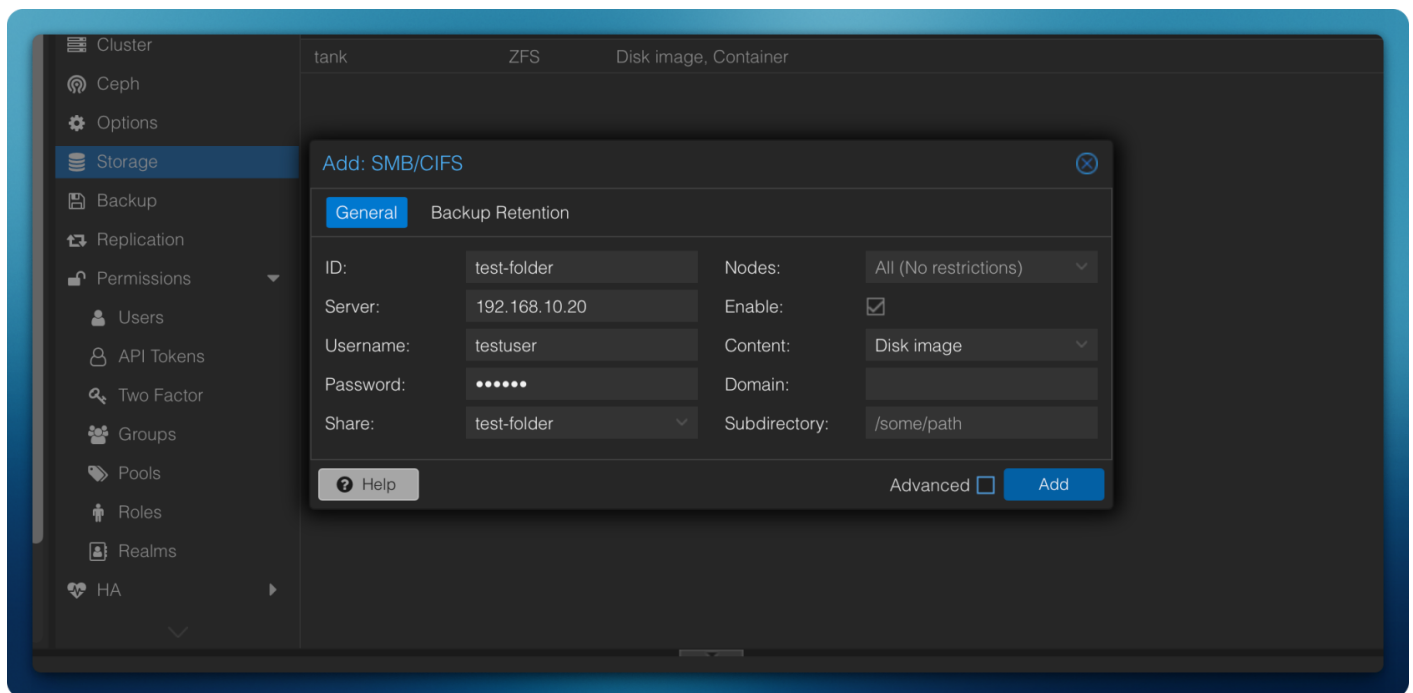
Add | Remove | Edit

- Directory
- LVM
- LVM-Thin
- BTRFS
- NFS
- SMB/CIFS**
- iSCSI
- CephFS
- RBD
- ZFS over iSCSI
- ZFS
- Proxmox Backup Server
- ESXi

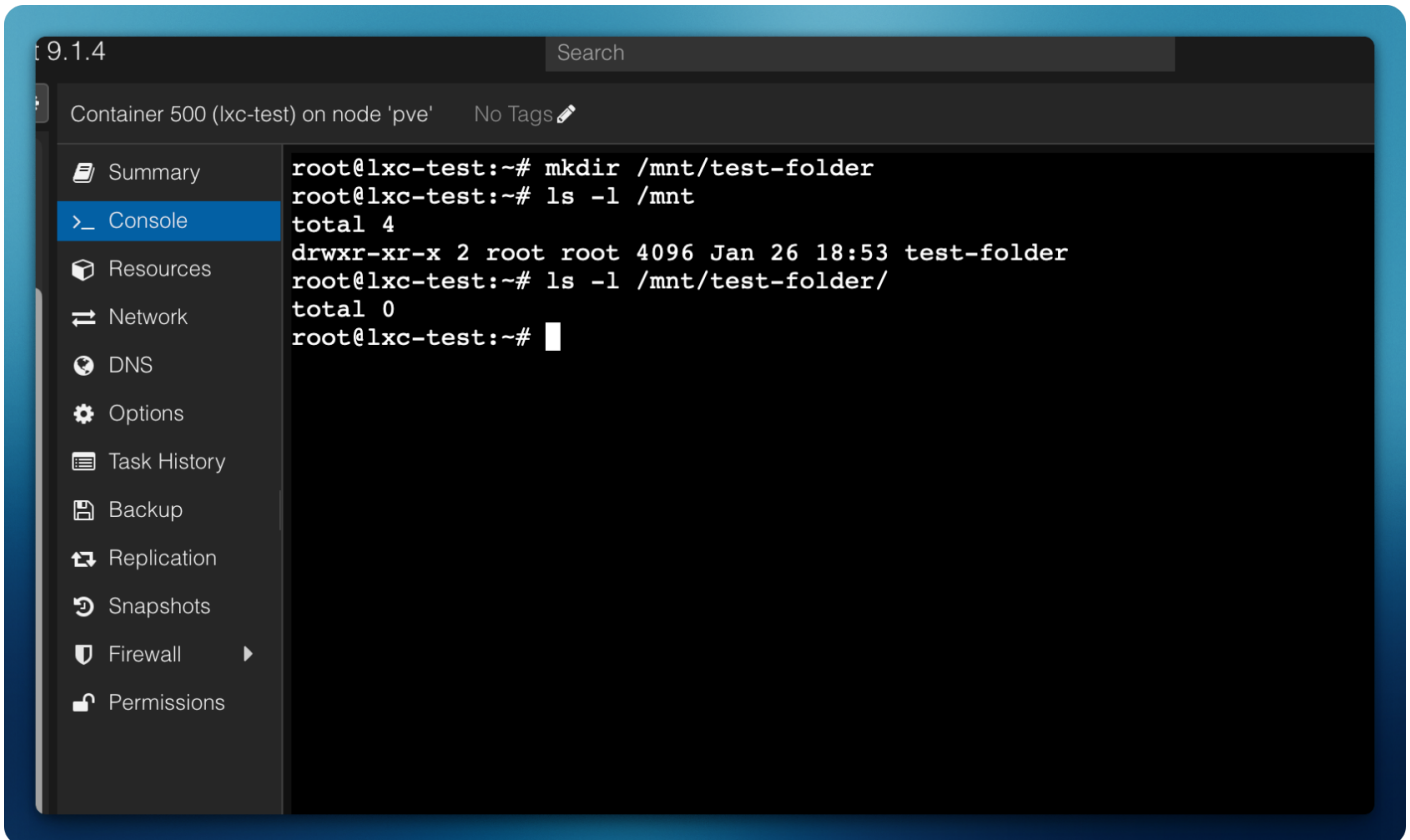
Content	Content
Directory	Disk image, Import, ISO image, Container, Snippets
CIFS	Backup, Disk image, Import, ISO image, Container
	Disk image, Container

Nachdem die SMB-Freigabe auf der Synology eingerichtet war, wechselte ich zu meinem Proxmox-Host. Vor dem Anlegen des LXC's öffnete ich in der Proxmox-Web Oberfläche den

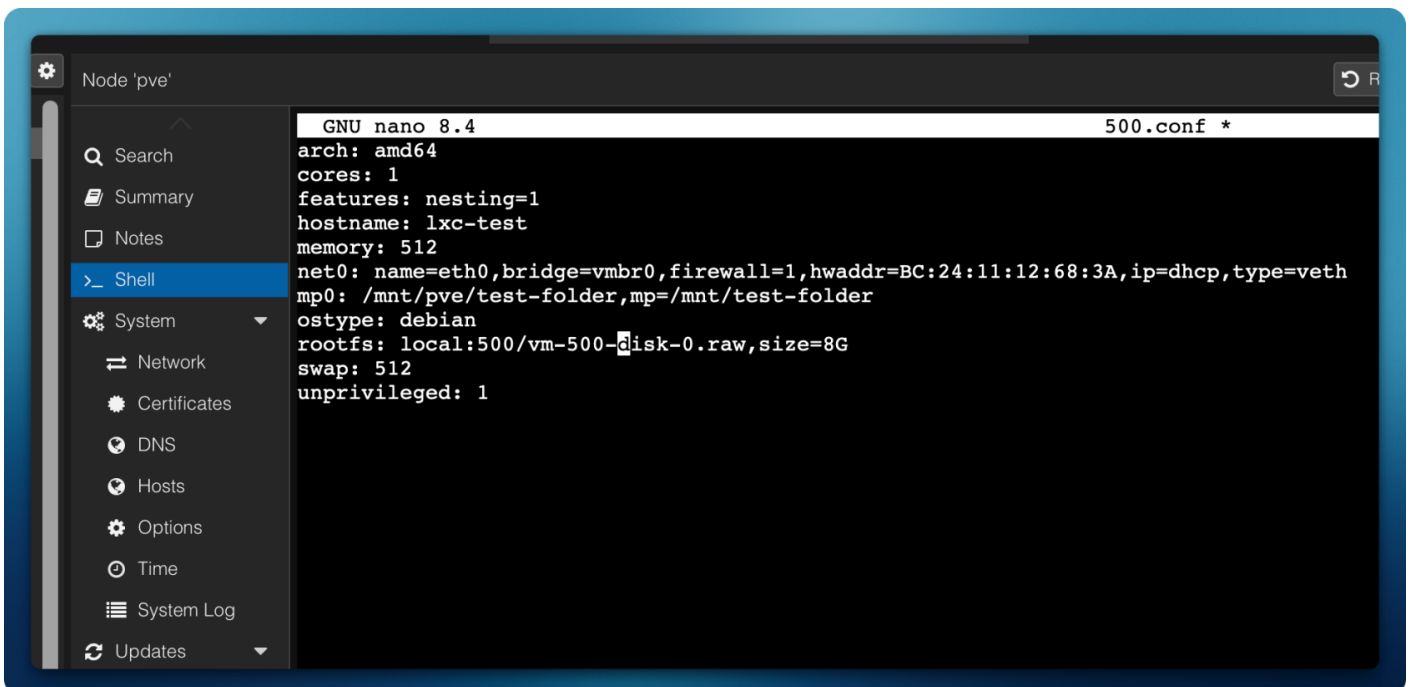
**Datastore**, ging auf den Reiter **Storage** und wählte über **Add ? CIFS/SMB** eine neue Netzwerkfreigabe aus. Als ID genügt ein beliebiger Name, als Server trug ich die IP-Adresse der Synology ein und hinterlegte Benutzername sowie Passwort des SMB-Kontos. Anschließend wurde die Freigabe automatisch erkannt und hinzugefügt.



Als Nächstes erstellte ich ein Verzeichnis, das als Mountpunkt im LXC dienen sollte. Dafür wechselte ich in die Shell meines **debianbasierten unprivilegierten Container** den ich zu Testzwecken mit der **ID 500** angelegt hatte. Anstatt zunächst Anwendungen zu konfigurieren, wechselte ich in der Proxmox-Oberfläche direkt in die **Console** des Containers und legte mit `mkdir -p /mnt/test-folder` das entsprechende Verzeichnis an.

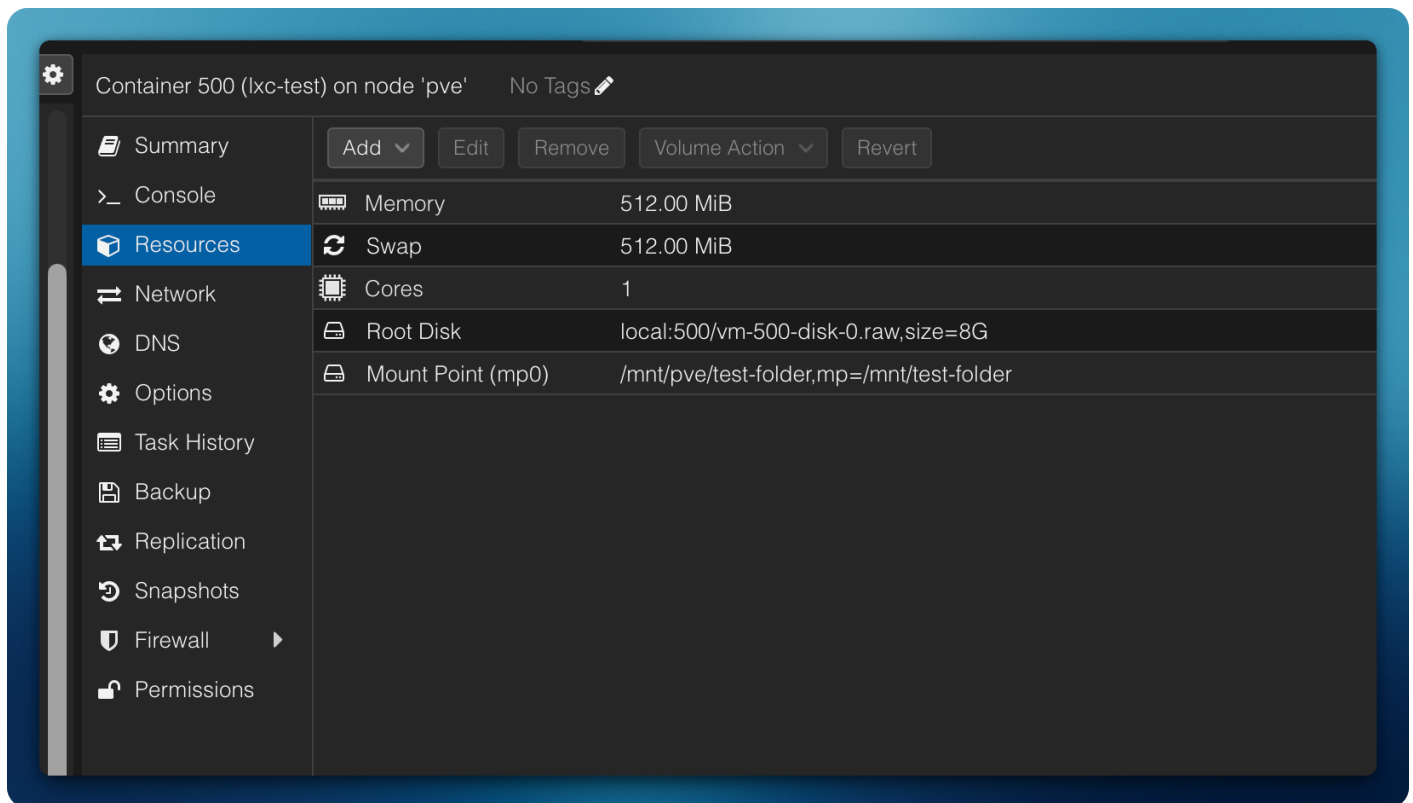


Zum Schluss wechselte ich auf dem Proxmox-Host in die Shell, navigierte nach `/etc/pve/lxc`, öffnete die Konfigurationsdatei des entsprechenden Containers (`500.conf`) und ergänzte dort die erforderliche Mount-Konfiguration, um die SMB-Freigabe im LXC verfügbar zu machen.



Hier verweist die Variable **test-folder** auf die ID, die ich der SMB-Freigabe in Proxmox zugewiesen hatte. Auch ohne einen Neustart sollte es funktionieren, zur Sicherheit aber, ist es sinnvoll einen

Neustart des Containers durchzuführen und dann zu schauen, ob das eingebundene Verzeichnis im LXC sichtbar wird. In meinem Fall hat es geklappt.



## Test der Verbindung und erweitern der Rechte

Zur Überprüfung lud ich eine Datei auf die Synology NAS hoch, die anschließend unmittelbar im Container erschien. Damit war bestätigt, dass die Einbindung der SMB-Freigabe korrekt funktionierte.

Das tut es auch soweit so gut. Innerhalb des LXC kann ich nun **lesen** auf die Dateien zugreifen. Schreiben jedoch nicht, da die Rechte nicht weitergereicht werden.

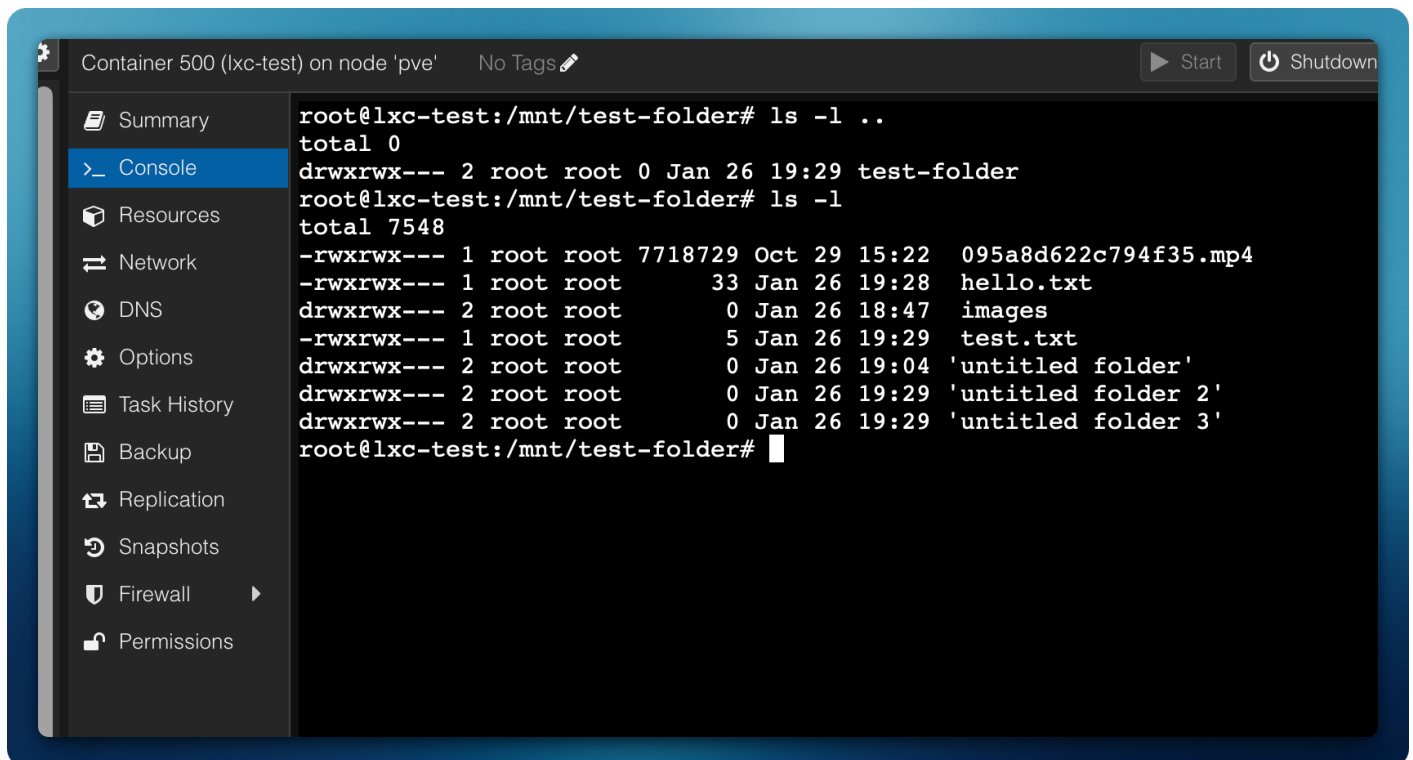
Um es dem lokalen `root` Nutzer des LXC zu ermöglichen auch Daten zu schreiben, ist noch ein kleiner Handgriff von Nöten.

Und zwar müssen wir auf dem Proxmox Node dazu einmal die `/etc/pve/storage.cfg` öffnen und folgende Zeile hinzufügen

```
options uid=100000,gid=100000,file_mode=0770,dir_mode=0770
```

```
cifs: test-folder
  path /mnt/pve/test-folder
  server 192.168.10.20
  share test-folder
  content images
  options uid=100000,gid=100000,file_mode=0770,dir_mode=0770
  prune-backups keep-all=1
  username testuser
```

Wenn wir das gemacht haben, speichern wir die Datei und starten die LXC neu. Danach sollte der Ordner nun Root gehören, und die korrekten Berechtigungen haben.



The screenshot shows a terminal window for a container named 'lxc-test' on node 'pve'. The terminal output shows the following commands and results:

```
root@lxc-test:/mnt/test-folder# ls -l ..
total 0
drwxrwx--- 2 root root 0 Jan 26 19:29 test-folder
root@lxc-test:/mnt/test-folder# ls -l
total 7548
-rwxrwx--- 1 root root 7718729 Oct 29 15:22 095a8d622c794f35.mp4
-rwxrwx--- 1 root root 33 Jan 26 19:28 hello.txt
drwxrwx--- 2 root root 0 Jan 26 18:47 images
-rwxrwx--- 1 root root 5 Jan 26 19:29 test.txt
drwxrwx--- 2 root root 0 Jan 26 19:04 'untitled folder'
drwxrwx--- 2 root root 0 Jan 26 19:29 'untitled folder 2'
drwxrwx--- 2 root root 0 Jan 26 19:29 'untitled folder 3'
root@lxc-test:/mnt/test-folder#
```

Revision #2

Created 2026-01-26 18:13:38 UTC by Carsten

Updated 2026-02-14 14:52:42 UTC by Carsten