Docker

does not support file handles

```
overlayfs: fs on '/var/lib/docker/overlay2/l/EDSZIPXO5E6HIZ5EIOYLLFCEW6'
does not support file handles, falling back to xino=off.
```

https://forum.proxmox.com/threads/docker-lxc-unprivileged-container-on-proxmox-7-with-zfs.99796/page-2#post-451845

docker inside unprivileged LXC

Some hints:

- Docker is recommended to be used inside VM.
- `overlay`/`overlay2` is not possible on ZFS (as documented by docker).
- In priviledged container it works with ZFS using AUFS. See below.
  - From LXC-NEWS: The aufs storage driver has been deprecated since LXC 2.1 and is now officially removed.
- ZFS over ZFS is not possible (ZFS nesting)

From documentation: https://pve.proxmox.com/wiki/Linux_Container#pct_configuration

- edit LXC container config

  /etc/pve/local/lxc/contained_id.conf

  ```
  features:  keyctl=1,nesting=1
  ```

- stop/start LXC container

  ```
  docker run hello-world
  ```

VFS FS is used by docker.

```
The vfs backend is a very simple fallback that has no copy-on-write support.
Each layer is just a separate directory. Creating a new layer based on
another layer is done by making a deep copy of the base layer into a new
directory.
Since this backend doesn’t share diskspace use between layers, and since
creating a new layer is a slow operation this is not a very practical
backend. However, it still has its uses, for instance to verify other
backends against, or if you need a super robust (if slow) backend that works
```

```
everywhere.
```

```
docker info
...
 Server Version: 19.03.8
 Storage Driver: vfs
...
```

When restarted in privileged container: NOTE: restarting in privileged container do mess with user permission. Make backup/clone before.

```
docker info
...
 Storage Driver: aufs
  Root Dir: /var/lib/docker/aufs
  Backing Filesystem: zfs
  Dirs: 0
  Dirperm1 Supported: true
...
```

Use RAW image for Docker

Edit existing `local` storage and add following content types:

- `Disk image`
- `Container`

Add mount point to LXC:

- Resources -> Add -> Mount Point
  - Storage: `local`
  - Backup: NO
  - Path: `/var/lib/docker`
  - Mount options: `noatime`

And new RAW disc will be created, with EXT4 FS. Docker will use `overlay2` driver with this FS.

Bind mount host btrfs subvolume

- Only directory hierarchy under `/mnt/bindmounts` are allowed to be bind-mounterd inside LXC containers.
- Host permission and ACL will be used. To play with permission to shared folder please read: Unprivileged LXC containers
- Container Settings)
- CONS: no disk size control inside guest. Possible to use btrfs subvolume quotas

Login to host using SSH node

```
btrfs subvol create /mnt/bindmounts
```

```
btrfs quota enable /mnt/bindmounts
btrfs subvol create /mnt/bindmounts/<cid>-docker
btrfs qgroup limit 50G /mnt/bindmounts/<cid>-docker

# Disable COW for performance:
chattr +C -f -R /mnt/bindmounts

# Give unprivilged container right to write
chown 100000.100000 /mnt/bindmounts/<cid>-docker
# setfacl -Rm user:100000:rwx,default:user:100000:rwx

pct set <cid> -mp0 /mnt/bindmounts/<cid>-docker,mp=/var/lib/docker
```

```
btrfs quota rescan /
btrfs qgroup show -pcre
```

Enabling ZFS tools and access - no success

```
dockerd -D --storage-driver zfs
...
zfs command is not available: exec: "zfs": executable file not found in
$PATH  storage-driver=zfs
...
apt-get install zfsutils-linux --no-install-recommends

dockerd -D --storage-driver zfs
...
DEBU[2020-05-06T16:37:51.502473451Z] cannot open /dev/zfs: open /dev/zfs: no
such file or directory  storage-driver=zfs
...
```

/etc/pve/lxc/${container_id}.conf

```
lxc.mount.entry: /dev/zfs dev/zfs none bind,create=file
```

Exposing `/dev/zfs` works. NOTE: it gives too wide permisions, like `zfs list` shows ALL info about host ZFS. After this trick, Docker starts and detects ZFS without any additional configuration. But usage is not possible. Docker cannot create additional subvolumes

```
~# docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
0e03bdcc26d7: Extracting
[==================================================>]  2.529kB/2.529kB
docker: failed to register layer: exit status 1: "/sbin/zfs fs create -o
mountpoint=legacy rpool/data/subvol-803-
disk-0/007d22d205263c9f89d2e53ab75787430a20a1b1b3b0270acf4eb67459de09ed" =>
cannot create 'rpool/data/subvol-803-
```

```
disk-0/007d22d205263c9f89d2e53ab75787430a20a1b1b3b0270acf4eb67459de09ed':
permission denied
```

Try to add permission on host (note, not possible to use UID which not exists in /etc/passwd)

```
zfs allow -u 100000 create,destroy,mount rpool/data/subvol-803-disk-0
```

Doesn't help. Give up and switch to RAW EXT4 image.

Trying to run on ZFS - without success

Solution:

```
cp /etc/apparmor.d/lxc/lxc-default-with-nesting /etc/apparmor.d/lxc/lxc-
default-with-nesting-docker
```

Edit new file and update `profile` name and add some mount permissions:

```
# Do not load this file.  Rather, load /etc/apparmor.d/lxc-containers, which
# will source all profiles under /etc/apparmor.d/lxc

profile lxc-container-default-with-nesting-docker
flags=(attach_disconnected,mediate_deleted) {
  #include <abstractions/lxc/container-base>
  #include <abstractions/lxc/start-container>

  deny /dev/.lxc/proc/** rw,
  deny /dev/.lxc/sys/** rw,
  mount fstype=proc -> /var/cache/lxc/**,
  mount fstype=sysfs -> /var/cache/lxc/**,
  mount options=(rw,bind),
  mount fstype=cgroup -> /sys/fs/cgroup/**,
  mount fstype=cgroup2 -> /sys/fs/cgroup/**,
  mount fstype=aufs,
  mount fstype=overlay,
}
```

```
systemctl reload apparmor
```

Edit `/etc/pve/lxc/${container_id}.conf` and append this line:

/etc/pve/lxc/${container_id}.conf

```
lxc.apparmor.profile: lxc-container-default-with-nesting-docker
```

Disabling apparmor

/etc/pve/lxc/${container_id}.conf

```
lxc.apparmor.profile = unconfined
```

```
systemctl reload apparmor
```

Forcing ''aufs''

Error: AUFS cannot be used in non-init user namespace

Forcing ''overlay2''

On host log:

```
kernel: overlayfs: filesystem on '/var/lib/docker/check-overlayfs-
support244358035/upper' not supported as upperdir
kernel: overlayfs: filesystem on '/var/lib/docker/check-overlayfs-
support445538983/upper' not supported as upperdir
```

There are some requirements to use overlayfs. It is not possible to use it over ZFS.