

# 第 7 章 Proxmox VE 存储

Proxmox VE 提供了非常灵活的存储配置模型。虚拟机镜像既可以保存在一种或多种服务器本地存储上，也可以保存在多种共享存储上，例如 NFS 或 iSCSI ( NAS , SAN )。你可以自由地配置多种存储池，想配多少就配多少，完全没有任何限制。事实上，Debian Linux 支持的所有存储技术都可以拿过来用。

使用共享存储保存虚拟机镜像的最大好处就是可以在线迁移虚拟机，只要集群的所有节点都能直接访问虚拟机磁盘镜像，那么就无需关机随意迁移，而且迁移时无需复制虚拟机镜像数据，这样也大大提高了迁移的速度。

Proxmox VE 的存储库 ( libpve-storage-perl 包 ) 具有非常灵活的插件式设计，并对所有存储技术提供了统一接口。这使 Proxmox VE 能够轻松兼容未来出现的新存储技术。

## 7.1 存储类型

Proxmox VE 将存储分为两种基本类型：

### 块存储

可用于存储 raw 格式的虚拟机镜像。但不可用于存储其他文件 ( ISO , 虚拟机备份 , ... )。大部分较新的块存储方案自带了快照和克隆功能。RADOS , sheepdog 和 GlusterFS 是分布式存储，并将数据分散在多个节点保存。

### 文件存储

指允许以全功能 ( POSIX ) 文件系统方式访问的存储方案。这类存储方案更加灵活，允许保存所有类型的数据。ZFS 大概是目前最先进的文件存储方案，并且完全支持快照和克隆功能。

表 7.1 Proxmox VE 支持的存储类型

名称	PVE 名称	级别	支持共享	支持快照	是否稳定
ZFS(local)	zfspool	文件	否	是	是
目录	dir	文件	否	否	是
NFS	nfs	文件	是	否	是
GlusterFS	glusterfs	文件	是	否	是
LVM	lvm	块	否	否	是
LVM-thin	lvmthin	块	否	是	是
iSCSI/kernel	iscsi	块	是	否	是
iSCSI/libiscsi	iscsidirect	块	是	否	是
Ceph/RBD	rbd	块	是	是	是
Sheepdog	sheepdog	块	是	是	测试中
ZFS over iSCSI	zfs	块	是	是	是

---

➤ 提示

可以在 iSCSI 存储上创建 LVM，这样就可以共享方式使用 LVM。

---

### 7.1.1 薄模式存储

一些存储方案，以及 Qemu 镜像文件 qcow2，支持薄模式（thin provisioning）。在薄模式下，只有虚拟机实际写入的数据才会占用物理存储空间。

例如，你创建了一个带有 32GB 磁盘的虚拟机，安装操作系统后，虚拟机根目录下有 3GB 数据。这时，薄模式存储上虚拟机磁盘仅使用 3GB 空间，而非你在虚拟机内查看磁盘容量时看到的 32GB。通过这种方式，薄模式存储允许你分配远大于当前实际可用存储空间的虚拟磁盘镜像。你可以为你的虚拟机创建很大的虚拟磁盘，当虚拟磁盘占用空间变大时，再向你的存储中增加物理硬盘设备，而无需重新调整虚拟磁盘的容量。

---

✓ 警告

当薄模式存储空间耗尽时，会造成其上所有虚拟机 IO 错误，进而导致文件系统不一致，甚至数据被破坏。建议不要超配存储空间，或者随时监控剩余空间，避免出错。

---

## 7.2 存储配置

Proxmox VE 所配置的存储配置信息全部保存在/etc/pve/storage.cfg 中。鉴于该文件在 /etc/pve 目录下，该文件会自动分发到集群的所有节点，所以所有节点都使用同样的存储配置信息。

共享存储配置信息对于共享存储非常有意义，因为共享存储本身就需要被所有节点访问。但对于本地存储来说也是很有用的，特别是在所有节点都配置了同一类本地存储的时候，尽管每个节点的本地存储都是不同的物理设备，其保存的数据也完全不一样。

### 7.2.1 存储池

每个存储池都有一个类型<type>，并唯一地被<STORAGE\_ID>标识。存储池的配置示例如下：

```
<type>: <STORAGE_ID>
    <property> <value>
    <property> <value>
    ...
```

其中，<type>: <STORAGE\_ID>是存储池配置的开始部分，其后是一组属性信息。大部分属性都需要配置参数值，但也有一部分使用默认值。使用默认值的情况下，可以省去<value>值。

以 Proxmox VE 安装后的默认存储配置文件为例，其中包含一个名为 local 的本地存储池，其路径为本地文件系统/var/lib/vz，并默认处于启用状态。Proxmox VE 安装程序也会根据安装时设置的存储类型创建其他存储服务。

默认存储配置文件（/etc/pve/storage.cfg）

```
dir: local

    path /var/lib/vz

    content iso,vztmpl,backup

# default image store on LVM based installation

lvmthin: local-lvm
```

```
thinpool data
vgname pve
content rootdir,images
```

```
# default image store on ZFS based installation
```

```
zfspool: local-zfs
    pool rpool/data
    sparse
    content images,rootdir
```

### 7.2.2 公共存储服务属性

在 Proxmox VE 中，有一些公共的存储服务属性，在不同类型的存储服务中都有。

- nodes

用于配置能够使用/访问当前存储服务的节点名列表。通过该属性可将存储服务配置为仅能由部分节点访问。

- content

存储服务可用于保存多种不同类型的数据，例如虚拟磁盘镜像，光盘 ISO 镜像，容器模板或容器根文件系统。不是所有存储服务都可以存储所有类型的数据。可通过该属性设置存储服务所要保存的数据类型。可设置的属性值如下：

- images      KVM-Qemu 虚拟机镜像
- rootdir      容器镜像数据
- vztpl      容器模板
- backup      虚拟机备份文件
- iso      ISO 镜像

- shared

用于标示存储服务是共享存储服务。

- disable

设置该属性值可禁用该存储服务。

- maxfiles

用于设置每个虚拟机备份文件最大数量。设为 0 表示不限制备份文件数量。

- format

用于设置默认的虚拟机镜像格式 ( raw|qcow2|vmdk ) 。

---

#### ✓ 警告

建议不要在不同 Proxmox VE 集群之间共享同一存储服务。由于某些存储服务访问操作有排他性，需要通过锁机制来防止并发访问。一个集群内可以通过锁机制防止并发访问，但两个集群之间就没办法禁止并发访问了。

---

## 7.3 存储卷

我们专门设计了一套存储空间命名规范。当你从存储池中分配了一块存储空间时，Proxmox VE 将返回一个存储卷标示符。存储卷标示符由多个部分组成，开头是存储服务标识<STORAGE\_ID>，其后是冒号，最后是基于存储数据类型命名的卷名称。如下是一些合法卷标识符<VOLUME\_ID>的示例：

local:230/example-image.raw

local:iso/debian-501-amd64-netinst.iso

local:vztmpl/debian-5.0-joomla\_1.5.9-1\_i386.tar.gz

iscsi-storage:0.0.2.scsi-14 f504e46494c4500494b5042546d2d646744372d31616d61

可用如下命令获取<VOLUME\_ID>对应的文件系统路径：

pvesm path <VOLUME\_ID>

### 7.3.1 存储卷从属关系

每个 image 类型的存储卷都有一个属主。每个 iamge 类型的存储卷，都属于一个虚拟机或容器。例如存储卷 local:230/example-image.raw 由 230 号虚拟机拥有。大部分后端存储都会把这种从属关系用于编码生成存储卷名称。

当你删除虚拟机或容器时，Proxmox VE 会同时删除其拥有的全部存储卷。

## 7.4 命令行界面使用方法

建议你熟悉并掌握 Proxmox VE 中存储池和存储卷的概念，但实际应用中，你不一定非要在命令行界面去实践基于这些概念的底层操作。通常情况下，使用虚拟机和容器管理工具分配或删除存储卷更加方便。

尽管如此，Proxmox VE 还是提供了一个名为 pvesm ( “Proxmox VE Storage Manager” ) 的命令行工具，可用于基本的存储服务管理操作。

### 7.4.1 示例

添加存储池

```
pvesm add <TYPE> <STORAGE_ID> <OPTIONS>
```

```
pvesm add dir <STORAGE_ID> --path <PATH>
```

```
pvesm add nfs <STORAGE_ID> --path <PATH> --server <SERVER> --export  
<EXPORT>
```

```
pvesm add lvm <STORAGE_ID> --vgname <VGNAME>
```

```
pvesm add iscsi <STORAGE_ID> --portal <HOST[:PORT]> --target <TARGET>
```

禁用存储池

```
pvesm set <STORAGE_ID> --disable 1
```

启用存储池

```
pvesm set <STORAGE_ID> --disable 0
```

修改/设置存储属性

```
pvesm set <STORAGE_ID> <OPTIONS>
```

```
pvesm set <STORAGE_ID> --shared 1
```

```
pvesm set local --format qcow2
```

```
pvesm set <STORAGE_ID> --content iso
```

删除存储池。该操作并不删除任何数据，也不断开任何连接或卸载任何文件系统，而仅仅是删除配置文件中相关内容。

```
pvesm remove <STORAGE_ID>
```

### 分配存储卷

```
pvesm alloc <STORAGE_ID> <VMID> <name> <size> [--format <raw|qcow2>]
```

在 local 存储中分配 4GB 的存储卷。如果设置<name>为空，系统将自动生成一个存储卷名称。

```
pvesm alloc local <VMID> " 4G
```

### 释放存储卷

```
pvesm free <VOLUME_ID>
```

---

### ✓ 警告

该操作将删除存储卷上的所有数据。

---

### 列出存储池状态

```
pvesm status
```

### 列出存储池中的存储卷

```
pvesm list <STORAGE_ID> [--vmid <VMID>]
```

### 列出某个虚拟机拥有的存储卷

```
pvesm list <STORAGE_ID> --vmid <VMID>
```

### 列出 iso 镜像

```
pvesm list <STORAGE_ID> --iso
```

列出容器模板

```
pvesm list <STORAGE_ID> --vztmpl
```

显示某个存储卷的文件系统路径

```
pvesm path <VOLUME_ID>
```

## 7.5 基于目录的后端存储

存储池类型：dir

Proxmox VE 可以使用本地目录或挂载在本地文件系统的共享存储作为存储服务。目录是文件系统级的存储服务，你可以在目录中保存任何类型的数据，包括虚拟机镜像，容器，模板，ISO 镜像或虚拟机备份文件。

---

### ✓ 注意

你可以通过 linux 配置文件/etc/fstab 挂载新增存储设备，然后将相应挂载点定义为目录存储服务，用这种方法就可以使用 Linux 支持的任意类型的文件系统。

---

Proxmox VE 对目录后端存储的唯一要求是兼容 POSIX 标准。这意味着你不能直接在目录存储服务上创建虚拟机快照，但可以使用 qcow2 格式自带的快照功能为保存在目录后端存储的虚拟机镜像创建快照。

---

### ✓ 提示

有些存储服务不支持 O\_DIRECT，所以你不能在这些存储服务上配置使用 none 模式的缓存，而需要设置缓存模式为 writeback。

---

Proxmox VE 会在目录后端存储上自动创建预先定义好的子目录，以便存储不同类型的数据。



表 7.2 目录后端存储子目录

数据类型	子目录
虚拟机镜像	images/<VMID>/
ISO 镜像	template/iso/
容器模版	template/cache/
备份文件	dump/

### 7.5.1 配置方法

目录后端存储支持全部的公共存储服务属性，此外还支持名为 path 的附加属性，以指定路径。配置 path 属性时需要使用绝对路径。

配置示例 ( /etc/pve/storage.cfg )

```
dir: backup
    path /mnt/backup
    content backup
    maxfiles 7
```

以上配置定义了名为 backup 的存储池。该存储池可以用来保存最多 7 个虚拟机备份文件（指每个虚拟机最多 7 个备份）。备份文件的绝对路径为/mnt/backup/dump/....

### 7.5.2 文件命名规范

目录后端存储有一套专门设计的虚拟机镜像文件命名规范，文件名格式如下：

vm-<VMID>-<NAME>.<FORMAT>

- <VMID>

镜像文件所属的虚拟机 ID.

- <NAME>

可以是任何不包含空白字符的字符串（ascii）。目录后端存储默认设置为 disk-[N]，其中[N]是一个不重复的整数序号。

- <FORMAT>

标识虚拟机镜像文件格式（ raw|qcow2|vmdk ）。

当你将一个虚拟机转换为虚拟机模版 时，Proxmox VE 会重新命名虚拟机镜像文件，以标明其处于只读状态，并仅供基础镜像或克隆使用。

base-<VMID>-<NAME>.<FORMAT>

✓ 注意

像虚拟机模版这样的基础镜像文件仅供用于克隆生成新的虚拟机。所以确保这类文件的只读属性非常重要。目录后端存储会将基础镜像文件的访问权限修改为 0444，并在文件系统支持的情况下设置不可修改标记（ chattr +i ）。

7.5.3 存储功能

如上所述，绝大部分文件系统本身不支持快照功能。如果要创建虚拟机快照，只能利用 qcow2 文件格式自带的快照功能。

同理，对于链接克隆操作，目录后端存储服务利用 qcow2 的基础镜像功能实现以链接克隆方式创建新虚拟机。

表 7.3 目录后端存储功能

数据类型	镜像格式	支持共享	支持快照	支持链接克隆
虚拟机镜像	raw	否	qcow2	qcow2
容器镜像	qcow2			
容器模版	vmdk			
ISO 镜像	subvol			
虚拟机备份				

7.5.4 示例

如下命令用于在 local 存储池上创建一个 4GB 的磁盘镜像：

# pvesm alloc local 100 vm-100-disk10.raw 4G

```
Formatting ' /var/lib/vz/images/100/vm-100-disk10.raw' , fmt=raw
size=4294967296

successfully created 'local:100/vm-100-disk10.raw'
```

---

✓ 注意

虚拟机镜像文件必须按照如前所述的规范进行命名。

---

如下命令用于查看镜像文件路径：

```
# pvesm path local:100/vm-100-disk10.raw
/var/lib/vz/images/100/vm-100-disk10.raw
```

如下命令用于删除镜像文件：

```
# pvesm free local:100/vm-100-disk10.raw
```

## 7.6 基于 NFS 的后端存储

存储池类型：nfs

基于 NFS 的后端存储服务实际上建立在目录后端存储之上，其属性和目录后端存储非常相似。其中子目录布局 and 文件命名规范完全一致。NFS 后端存储的优势在于，你可以通过配置 NFS 服务器参数，实现 NFS 存储服务自动挂载，而无需编辑修改/etc/fstab 文件。NFS 存储服务能够自动检测 NFS 服务器的在线状态，并自动连接 NFS 服务器输出的共享存储服务。

### 7.6.1 配置方法

NFS 后端存储支持全部的公共存储服务属性，但 shared 标识例外，因为 NFS 后端存储的 shared 属性值总是设置为启用状态。此外，NFS 后端存储还具有以下属性，以便于配置 NFS 服务器：

- server

设置 NFS 服务器的 IP 地址或 DNS 域名。建议直接配置为 IP 地址，以避免 DNS 查询带来的额外延迟——除非你的 DNS 服务器非常强大，或者以本地/etc/hosts 文件方式解析 DNS 域名。

- export

设置 NFS 服务器输出的共享存储路径（可用 `pvesm nfsscan` 命令扫描查看）。

此外，你还可以通过如下属性设置 NFS 存储挂载点：

- path

NFS 后端存储在 Proxmox VE 服务器上的挂载点(默认为/mnt/pve/<STORAGE\_ID>/ )。

- options

NFS 挂载选项（可查看 `man nfs` 获取更多信息）。

配置示例（`/etc/pve/storage.cfg`）

nfs: iso-templates

path /mnt/pve/iso-templates

server 10.0.0.10

export /space/iso-templates

options vers=3,soft

content iso,vztmpl

---

✓ 提示

在 NFS 连接请求超时后，NFS 默认会持续尝试建立连接。这有可能导致 NFS 客户端一侧的意外死机。对于保存只读数据的 NFS 存储，可以考虑使用 `soft` 选项，以限制尝试连接次数为 3。

---

## 7.6.2 存储功能

NFS 本身不支持快照功能，但可利用 `qcow2` 文件格式的支持进行虚拟机快照和链接克隆。

表 7.4 NFS 后端存储功能

数据类型	镜像格式	支持共享	支持快照	支持链接克隆
虚拟机镜像	raw	是	qcow2	qcow2
容器镜像	qcow2			
容器模版	vmdk			
ISO 镜像	subvol			
虚拟机备份				

### 7.6.3 示例

可用如下命令列出 NFS 共享路径：

```
# pvesm nfsscan <server>
```

## 7.7 基于 GlusterFS 的后端存储

存储池类型：glusterfs

GlusterFS 是一个可水平扩展的网络文件系统。GlusterFS 具有模块化设计，兼容常见硬件等优点，是一种低成本的高可用企业级存储解决方案。GlusterFS 能够支持扩容到数 P 字节容量，并可同时支持数千客户端连接。

---

#### ✓ 注意

在遭遇节点/brick 故障时，GlusterFS 会通过 rsync 重新同步数据，而大文件同步往往会需要很长时间，所以 GlusterFS 不适宜用于虚拟机镜像存储。

---

### 7.7.1 配置方法

GlusterFS 后端存储支持全部的公共存储服务属性，以及如下的 GlusterFS 特有属性：

- server

GlusterFS 存储服务器 IP 或 DNS 域名。

- server2

GlusterFS 备用存储服务器 IP 或 DNS 域名。

- volume

GlusterFS 卷名称。

- transport

GlusterFS 网络传输协议：tcp，unix 或 rdma。

配置示例 ( /etc/pve/storage.cfg )

```
glusterfs: Gluster
    server 10.2.3.4
    server2 10.2.3.5
    volume glustervol
    content images,iso
```

7.7.2 文件命名规范

GlusterFS 后端存储的子目录布局 and 文件命名规范与目录后端存储完全一致。

7.7.3 存储功能

NFS 本身不支持快照功能，但可利用 qcow2 文件格式的支持进行虚拟机快照和链接克隆。

表 7.5 GlusterFS 后端存储功能

数据类型	镜像格式	支持共享	支持快照	支持链接克隆
虚拟机镜像 容器模版 ISO 镜像 虚拟机备份	raw qcow2 vmdk	是	qcow2	qcow2

## 7.8 基于本地 ZFS 的后端存储

存储池类型：zfspool

该类型后端存储基于本地 ZFS 存储池（或 ZFS 存储池中的 ZFS 文件系统）建立。

### 7.8.1 配置方法

ZFS 后端存储支持公共存储服务属性 content、nodes、disable，以及如下的 ZFS 特有属性：

- pool

用于设置所使用的 ZFS 存储池/文件系统名称。所有的 Proxmox VE 存储卷都将在指定的存储池分配。

- blocksize

用于设置 ZFS 数据块大小。

- sparse

用于设置启用 ZFS 的薄存储模式。薄模式下，一个存储卷的大小等于其内部数据所占用的实际空间，而非分配给它的总空间。

配置示例（/etc/pve/storage.cfg）

```
zfspool: vmdata
    pool tank/vmdata
    content rootdir,images
    sparse
```

### 7.8.2 文件命名规范

ZFS 后端存储采用如下虚拟机镜像文件命名规范：

vm-<VMID>-<NAME>      // 普通虚拟机镜像

base-<VMID>-<NAME>      // 虚拟机模版（只读）

subvol-<VMID>-<NAME>    // 容器镜像（使用 ZFS 文件系统存储容器）

- <VMID>

镜像文件所属的虚拟机 ID.

- <NAME>

可以是任何不包含空白字符的字符串 ( ascii ) 。目录后端存储默认设置为 disk-[N] , 其中[N]是一个不重复的整数序号。

### 7.8.3 存储功能

在快照功能和克隆功能方面，ZFS 大概是最强大的后端存储方案。ZFS 后端存储同时支持虚拟机镜像 ( raw 格式 ) 和容器镜像 ( subvol 格式 ) 的存储。ZFS 的配置继承自上级存储池，所以你只需配置上级存储池使用默认属性值即可。

表 7.6 ZFS 后端存储功能

数据类型	镜像格式	支持共享	支持快照	支持链接克隆
虚拟机镜像	raw	否	是	是
容器镜像	subvol			

### 7.8.4 示例

推荐创建另外的 ZFS 文件系统以存储虚拟机镜像：

```
# zfs create tank/vmdata
```

如下命令在新建文件系统开启数据压缩功能：

```
# zfs set compression=on tank/vmdata
```

如下命令用于列出可用的 ZFS 文件系统：

```
# pvesm zfsscan
```

## 7.9 基于 LVM 的后端存储

存储池类型：lvm



LVM 是建立在硬盘设备和分区之上的一个轻量级存储层软件。LVM 可将硬盘空间划分为多个小的逻辑卷。LVM 在 Linux 上得到了广泛应用，并大大简化了硬盘管理操作。

另一种方式使用 LVM 管理大的 iSCSI LUN。这样可以轻松实现 iSCSI LUN 的空间分配，否则，在 iSCSI 本身不提供空间分配接口的情况下，这将是一个不可能完成的任务。

### 7.9.1 配置方法

LVM 后端存储支持公共存储服务属性 content、nodes、disable，以及如下的 LVM 特有属性：

- vname

用于设置 LVM 的卷组（VG）名称。必须设置为已有卷组的名称。

- base

用于标识基本卷。基本卷必须在访问存储之前就自动激活。该属性常用在远端 iSCSI 服务器上的 LVM 卷组。

- saferemove

用于标识删除逻辑卷时同步擦除数据。设置该属性后，删除逻辑卷时，LVM 将确保所有数据被物理擦除。

- saferemove\_throughput

用于设置擦除数据块大小。（即 cstream -t 参数值）。

配置示例（/etc/pve/storage.cfg）

```
lvm: myspace
```

```
    vname myspace
```

```
    content rootdir,images
```

### 7.9.2 文件命名规范

LVM 后端存储的命名规范与 ZFS 后端存储基本一致。

```
vm-<VMID>-<NAME>      //普通虚拟机镜像
```

### 7.9.3 存储功能

LVM 是典型的块存储解决方案，但 LVM 后端存储本身不支持快照和链接克隆功能。更不幸的是，在创建普通 LVM 快照期间，整个卷组的写操作都会受到影响而变得非常低效。

最大的好处是你可以在共享存储上建立 LVM 后端存储服务。例如可以在 iSCSI LUN 上建立 LVM。LVM 后端存储自带 Proxmox VE 集群锁以有效防止并发访问冲突。

✓ 提示

最新的 LVM-thin 后端存储提供了快照和链接克隆功能，但不支持在共享存储上使用。

表 7.7 LVM 后端存储功能

数据类型	镜像格式	支持共享	支持快照	支持链接克隆
虚拟机镜像 容器镜像	raw	可能	否	否

### 7.9.4 示例

以下命令列出所有可用的 LVM 后端存储：

```
# pvesm lvmscan
```

## 7.10 基于 LVM-thin 的后端存储

存储池类型：lvm-thin

LVM 是在逻辑卷创建时就按设置的卷容量大小预先分配所需空间。LVM-thin 存储池是在向卷内写入数据时按实际写入数据量大小分配所需空间。LVM-thin 所用的存储空间分配方式允许创建容量远大于物理存储空间的存储卷，因此也称为“薄模式”。

创建和管理 LVM-thin 存储池的命令和 LVM 命令完全一致（参见 man lvmthin）。假定你已经有一个 LVM 卷组 pve，如下命令可以创建一个名为 data 的新 LVM-thin 存储池（容量 100G）：

```
lvcreate -L 100G -n data pve  
lvconvert --type thin-pool pve/data
```

### 7.10.1 配置方法

LVM-thin 后端存储支持公共存储服务属性 content、nodes、disable，以及如下的 LVM 特有属性：

- vname

用于设置 LVM 的卷组（VG）名称。必须设置为已有卷组的名称。

- thinpool

LVM-thin 存储池名称。

配置示例（/etc/pve/storage.cfg）

```
lvmthin: local-lvm  
    thinpool data  
    vname pve  
    content rootdir,images
```

### 7.10.2 文件命名规范

LVM-thin 后端存储的命名规范与 ZFS 后端存储基本一致。

vm-<VMID>-<NAME>        //普通虚拟机镜像

### 7.10.3 存储功能

LVM-thin 属于块存储解决方案，同时支持快照和链接克隆功能。新创建的卷数据默认为全 0。

必须强调，LVM-thin 存储池不能被多个节点同时共享使用，只能用于节点本地存储。

表 7.8 LVM-thin 后端存储功能

数据类型	镜像格式	支持共享	支持快照	支持链接克隆
虚拟机镜像 容器镜像	raw	否	是	是

#### 7.10.4 示例

以下命令列出卷组 pve 上所有可用的 LVM-thin 后端存储：

```
# pvesm lvmthinscan pve
```

### 7.11 基于 [Open-iSCSI](#) 的后端存储

存储池类型：iscsi

iSCSI 是一种广泛应用于服务器和存储设备之间连接的协议。几乎所有的存储厂商都有兼容 iSCSI 的设备。目前也有多种开源的 iSCSI target 解决方案，例如基于 Debian 系统的 [OpenMeidaVault](#)。

你需要先手工安装 open-iscsi 软件包才可以使用基于 Open-iSCSI 的后端存储服务。Proxmox VE 默认不安装该 Debian 软件包。

```
# apt-get install open-iscsi
```

底层的 iscsi 管理任务可以通过 iscsiadm 命令完成。

#### 7.11.1 配置方法

Open-iSCSI 后端存储支持公共存储服务属性 content、nodes、disable，以及如下的 iSCSI 特有属性：

- portal

用于设置 iSCSI Portal（可设置为 IP 地址或 DNS 域名）。

- target

用于设置 iSCSI target。

配置示例 ( /etc/pve/storage.cfg )

```
iscsi: mynas
    portal 10.10.10.1
    target iqn.2006-01.openfiler.com:tsn.dcb5aaadd
    content none
```

✓ 提示

如果需要在 iSCSI 上创建 LVM 存储服务，最好启用 content none。这样就防止直接在 iSCSI LUN 上创建虚拟机镜像。

7.11.2 文件命名规范

iSCSI 协议本身未定义分配空间或删除数据的接口，而是将这部分工作交由各存储厂商自行实现。一般情况下，iSCSI 上分配的存储卷都以 LUN 序号的形式输出，所以 Proxmox VE 就以 Linux 内核获取的 LUN 信息命名 iSCSI 存储卷。

7.11.3 存储功能

iSCSI 属于块存储解决方案，但并未提供任何管理接口。所以，最佳时间是配置并输出一个很大的 iSCSI LUN，然后再配置创建 LVM 进行管理。你可以使用 Proxmox VE 的 LVM 插件直接管理 iSCSI LUN 的存储空间。

表 7.9 iscsi 后端存储功能

数据类型	镜像格式	支持共享	支持快照	支持链接克隆
虚拟机镜像 none	raw	是	否	否

7.11.4 示例

以下命令用于扫描远端的 iSCSI portal，并列出的 target：

```
pvesm iscsiscan -portal <HOST[:PORT]>
```

# 7.12 基于用户空间 iSCSI 的后端存储

存储池类型：iscsidirect

用户空间 iSCSI 和 Open-iSCSI 后端存储功能相近，其主要区别在于使用用户空间库（libiscsi2）实现。

需要强调的是，iscsidirect 未使用内核组件。由于省去了内核空间切换，所以其性能更加优秀，但代价是不能其创建的 iSCSI LUN 上配置使用 LVM，你只能在存储服务器端完成 iSCSI LUN 的划分和管理。

## 7.12.1 配置方法

用户空间 iSCSI 后端存储的属性和 Open-iSCSI 后端存储完全一致。

```
iscsidirect: faststore
    portal 10.10.10.1
    target iqn.2006-01.openfiler.com:tsn.dcb5aaaddd
```

## 7.12.2 存储功能

✓ 提示

用户空间 iSCSI 后端存储仅能用于 KVM 虚拟机镜像存储，不能存储容器镜像。

表 7.10 iscsidirect 后端存储功能

数据类型	镜像格式	支持共享	支持快照	支持链接克隆
虚拟机镜像	raw	是	否	否

# 7.13 基于 Ceph RADOS 块设备的后端存储

存储池类型：rbd

Ceph 是一种同时支持对象存储和文件存储的高性能分布式存储解决方案，其设计兼顾高性能、可靠性、可扩展性。RADOS 块设备是一种功能强大的块级别存储设备，优势如下：

- 薄模式存储
- 存储卷容量可调
- 分布式存储及多副本存储（基于多个 OSD 的条带）
- 支持快照和链接式克隆
- 数据自修复
- 无单点故障
- 容量可扩展至数 E 字节。
- 支持内核空间 and 用户空间实现

系统用户空间 iSCSI 和 Open-iSCSI 后端存储功能相近，其主要区别在于使用用户空间库（libiscsi2）实现。

需要强调的是，iscsidirect 未使用内核组件。由于省去了内核空间切换，所以其性能更加优秀，但代价是不能在其创建的 iSCSI LUN 上配置使用 LVM，你只能在存储服务器端完成 iSCSI LUN 的划分和管理。

---

#### ✓ 注意

小规模部署场景下，也可以直接在 Proxmox VE 服务器上运行 Ceph 存储服务。近些年服务器的 CPU 和内存配置足以支持同时运行存储服务和虚拟机应用。

---

### 7.13.1 配置方法

rbd 后端存储支持公共存储服务属性 content、nodes、disable，以及如下的 rbd 特有属性：

- monhost

用于设置监控服务绑定的 IP 地址。

- pool

用于设置 Ceph 存储池名称。

- username

Ceph 用户 ID。

- krbd

用于设置通过内核模块 krbd 访问 rbd 存储服务。如需要使用 Ceph 存储容器镜像，需要启用该设置。

配置示例 ( /etc/pve/storage.cfg )

rbd: ceph3

monhost 10.1.1.20 10.1.1.21 10.1.1.22

pool ceph3

content images

username admin

---

✓ 提示

Ceph 底层管理任务可以使用 rbd 命令完成。

---

### 7.13.2 认证方式

如选择使用 cephx 认证方式，需要将密钥文件从 Ceph 服务器复制到 Proxmox VE 服务器。首先运行如下命令创建目录/etc/pve/priv/ceph

```
mkdir /etc/pve/priv/ceph
```

然后复制密钥文件

```
scp <cephserver>:/etc/ceph/ceph.client.admin.keyring  
/etc/pve/priv/ceph/<STORAGE_ID>.keyring
```

密钥文件名称需要和<STORAGE\_ID>一致。注意复制操作需要 root 权限才能完成。



### 7.13.3 存储功能

rbid 属于块存储解决方案，并支持快照和链接克隆。

表 7.11 rbd 后端存储功能

数据类型	镜像格式	支持共享	支持快照	支持链接克隆
虚拟机镜像 容器镜像	raw	是	是	是