

第一块 5B 已经上岗营业，没办法停机，所以等第二块到手了，才有机会再次做一遍。

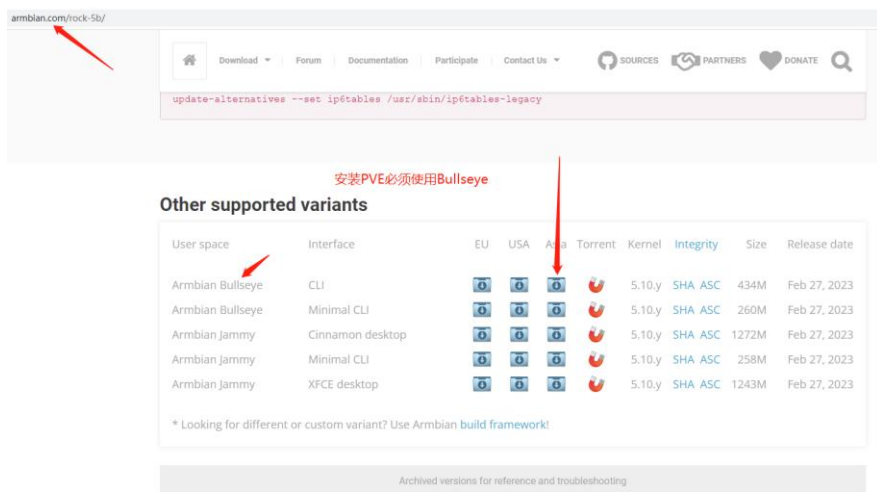
以下内容仅供参考。还是和之前一样，因 radxa 官方论坛有图片和链接发布数量限制，再次以文本方式发布该内容。

第二块 5B，嗯貌似——安慰奖,应该没有比这更惨的吧

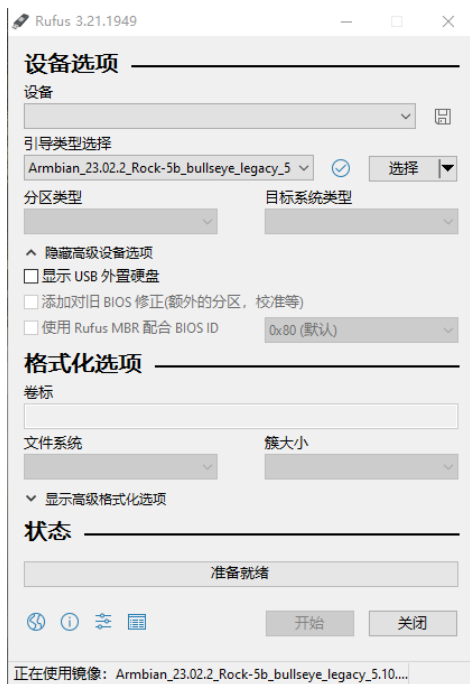
```
root@rock-5b:~# dmesg|grep pvtm
[ 7.700474] rockchip-pvtm fda40000.pvtm: pvtm@0 probed
[ 7.700531] rockchip-pvtm fda50000.pvtm: pvtm@1 probed
[ 7.700583] rockchip-pvtm fda60000.pvtm: pvtm@2 probed
[ 7.700634] rockchip-pvtm fdaf0000.pvtm: pvtm@3 probed
[ 7.700682] rockchip-pvtm fdb30000.pvtm: pvtm@4 probed
[ 7.992625] cpu cpu0: pvtm=1446
[ 7.992714] cpu cpu0: pvtm-volt-sel=2
[ 8.001433] cpu cpu4: pvtm=1680
[ 8.005408] cpu cpu4: pvtm-volt-sel=4
[ 8.014802] cpu cpu6: pvtm=1692
[ 8.018751] cpu cpu6: pvtm-volt-sel=4
[ 8.106079] mali fb000000.gpu: pvtm=889
[ 8.106136] mali fb000000.gpu: pvtm-volt-sel=4
[ 8.291415] RKNPU fdab0000.npu: pvtm=888
[ 8.295570] RKNPU fdab0000.npu: pvtm-volt-sel=4
```

下载 Armbian

用 armbian 官方的比较省事，不用自己再编译



写盘



插 TF 卡、网线、插电启动

查到 IP 地址，ssh 进去

```
Create root password: ****
Repeat root password: ****

Warning: Weak password, it is too short!

Support status: community support (looking for a dedicated maintainer)

Choose default system command shell:

1) bash
2) zsh

Shell: ZSH

Creating a new user account. Press <Ctrl-C> to abort
```

禁用 NetworkManager

systemctl stop NetworkManager

systemctl disable NetworkManager

```
root@rock-5b:~# systemctl stop NetworkManager
systemctl disable NetworkManager
Removed /etc/systemd/system/dbus-org.freedesktop.nm-dispatcher.service.
Removed /etc/systemd/system/multi-user.target.wants/NetworkManager.service.
root@rock-5b:~#
```

配置网络

```
root@rock-5b:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enP4p65s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:e0:4c:68:01:fe brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.93/24 brd 192.168.0.255 scope global dynamic noprefixroute enP4p65s0
        valid_lft 84991sec preferred_lft 84991sec
    inet6 fd6f:1bef:e5dc:0:bd3:b669:4488:6330/64 scope global noprefixroute
        valid_lft forever preferred_lft forever
```

如果是 5B，这个网卡名称应该都是这个名字

nano /etc/network/interfaces

```
GNU nano 5.4 /etc/network/interfaces *
#source /etc/network/interfaces.d/*
# Network is managed by Network manager
auto lo
iface lo inet loopback

auto enP4p65s0
iface enP4p65s0 inet static
    hardware ether 00:e0:4c:68:01:fe
    address 192.168.0.93
    netmask 255.255.255.0
    gateway 192.168.0.2
    dns-nameservers 192.168.0.11
```

###仅供参考，这个设置只是为了保障一会安装 PVE 用，安装 PVE 后使用网桥，这个配置不再使用。

#####

```
auto enP4p65s0
iface enP4p65s0 inet static
    hwaddress ether 00:e0:4c:68:01:fe
    address 192.168.0.93
    netmask 255.255.255.0
    gateway 192.168.0.2
    dns-nameservers 192.168.0.11
```

编辑/etc/hostname

nano /etc/hostname

这个名字要与/etc/hosts 内保持一致，不然 WEB 页面可能会打不开

```
GNU nano 5.4 /etc/hostname *
cooip
```

nano /etc/hosts

```
GNU nano 5.4 /etc/hosts *
127.0.0.1 localhost.localdomain localhost
192.168.0.93 cooip.rock.5b cooip
#::1 localhost rock-5b ip6-localhost ip6-loopback
#fe00::0 ip6-localnet
#ff00::0 ip6-mcastprefix
#ff02::1 ip6-allnodes
#ff02::2 ip6-allrouters
```

换源(根据自己喜好，不过我还是推荐清华源)

nano /etc/apt/sources.list

```
GNU nano 5.4 /etc/apt/sources.list
# 默认注释了源码镜像以提高 apt update 速度，如有需要可自行取消注释
deb https://mirrors.tuna.tsinghua.edu.cn/debian/ bullseye main contrib non-free
# deb-src https://mirrors.tuna.tsinghua.edu.cn/debian/ bullseye main contrib non-free

deb https://mirrors.tuna.tsinghua.edu.cn/debian/ bullseye-updates main contrib non-free
# deb-src https://mirrors.tuna.tsinghua.edu.cn/debian/ bullseye-updates main contrib non-free

deb https://mirrors.tuna.tsinghua.edu.cn/debian/ bullseye-backports main contrib non-free
# deb-src https://mirrors.tuna.tsinghua.edu.cn/debian/ bullseye-backports main contrib non-free

deb https://mirrors.tuna.tsinghua.edu.cn/debian-security bullseye-security main contrib non-free
# deb-src https://mirrors.tuna.tsinghua.edu.cn/debian-security bullseye-security main contrib non-free
```

# 默认注释了源码镜像以提高 apt update 速度，如有需要可自行取消注释

deb https://mirrors.tuna.tsinghua.edu.cn/debian/ bullseye main contrib non-free

deb https://mirrors.tuna.tsinghua.edu.cn/debian/ bullseye-updates main contrib non-free

deb https://mirrors.tuna.tsinghua.edu.cn/debian/ bullseye-backports main contrib non-free

deb https://mirrors.tuna.tsinghua.edu.cn/debian-security bullseye-security main contrib non-free

<Cancel>

@不知道

@cooip

```
Created symlink /etc/systemd/system/multi-user.target.wants/pvedaemon.service → /lib/systemd/system/pvedaemon.service.
Created symlink /etc/systemd/system/multi-user.target.wants/pveproxy.service → /lib/systemd/system/pveproxy.service.
Created symlink /etc/systemd/system/multi-user.target.wants/spiceproxy.service → /lib/systemd/system/spiceproxy.service.
Created symlink /etc/systemd/system/multi-user.target.wants/pvestatd.service → /lib/systemd/system/pvestatd.service.
Created symlink /etc/systemd/system/getty.target.wants/pvebanner.service → /lib/systemd/system/pvebanner.service.
Created symlink /etc/systemd/system/multi-user.target.wants/pvescheduler.service → /lib/systemd/system/pvescheduler.service.
Created symlink /etc/systemd/system/timers.target.wants/pve-daily-update.timer → /lib/systemd/system/pve-daily-update.timer.
Created symlink /etc/systemd/system/sysinit.target.wants/pvenetcommit.service → /lib/systemd/system/pvenetcommit.service.
Created symlink /etc/systemd/system/pve-manager.service → /lib/systemd/system/pve-guests.service.
Created symlink /etc/systemd/system/multi-user.target.wants/pve-guests.service → /lib/systemd/system/pve-guests.service.
Backing up lvm.conf before setting pve-manager specific settings ..
'/etc/lvm/lvm.conf' → '/etc/lvm/lvm.conf.bak'
Setting 'global_filter' in /etc/lvm/lvm.conf to prevent zvols from being scanned:
global_filter=["a|.*|"] ⇒ global_filter=["r|/dev/zd.*|"]
Setting up proxmox-ve (7.2-1) ...
Processing triggers for fontconfig (2.13.1-4.2) ...
Processing triggers for initramfs-tools (0.140) ...
update-initramfs: Generating /boot/initrd.img-5.10.110-rockchip-rk3588
update-initramfs: Converting to u-boot format
Processing triggers for libc-bin (2.31-13+deb11u5) ...
Processing triggers for rsyslog (8.2102.0-2+deb11u1) ...
Processing triggers for man-db (2.9.4-2) ...
Processing triggers for pve-ha-manager (3.3-4) ...
root@rock-5b:~#
```

reboot

以上主体安装完成



下面是选装

(配置网络用)

apt install ifupdown2

网桥配置

节点 'cooip'

创建 还原 编辑 删除 应用配置

名称 ↑	类别	活动	自动启动	VLAN...	端口/从属	Bond模式	CIDR	网关
enP4p65s0	网络设备	是	是	否			192.168.0.93/24	192.168.0.2

创建: Linux Bridge

名称:  自动启动: ☒

IPv4/CIDR:  VLAN感知: ☐

网关 (IPv4):  桥接端口:

IPv6/CIDR:  备注:

网关 (IPv6):

帮助 高级 ☐ 创建

将物理接口地址配置到网桥上。

节点 'cooip'

创建 还原 编辑 删除 应用配置

名称 ↑	类别	活动	自动启动	VLAN...	端口/从属	Bond模式	CIDR	网关
enP4p65s0	网络设备	是	是	否				
vmbr0	Linux Bridge	否	是	否	enP4p65s0		192.168.0.93/24	192.168.0.2

安装 LXC OpenWRT

Openwrt 模板文件拜托“暴躁老哥”帮忙打包

下载地址

<https://github.com/breakings/OpenWrt/releases/download/ARMv8Docker/openwrt-armvirt-64-default-rootfs.tar.gz>

手动下载或者使用 PVE 自带的下载工具下载都可以（需要科学）

PROXMOX Virtual Environment 7.2-7 搜索

服务器视图

- 数据中心
  - cooip
    - local (cooip)

存储 'local' 在节点 'cooip' 上

概要 备份 VM磁盘 CT卷 ISO镜像 CT模板 片段 权限

上传 从URL下载 模板 删除

从URL下载

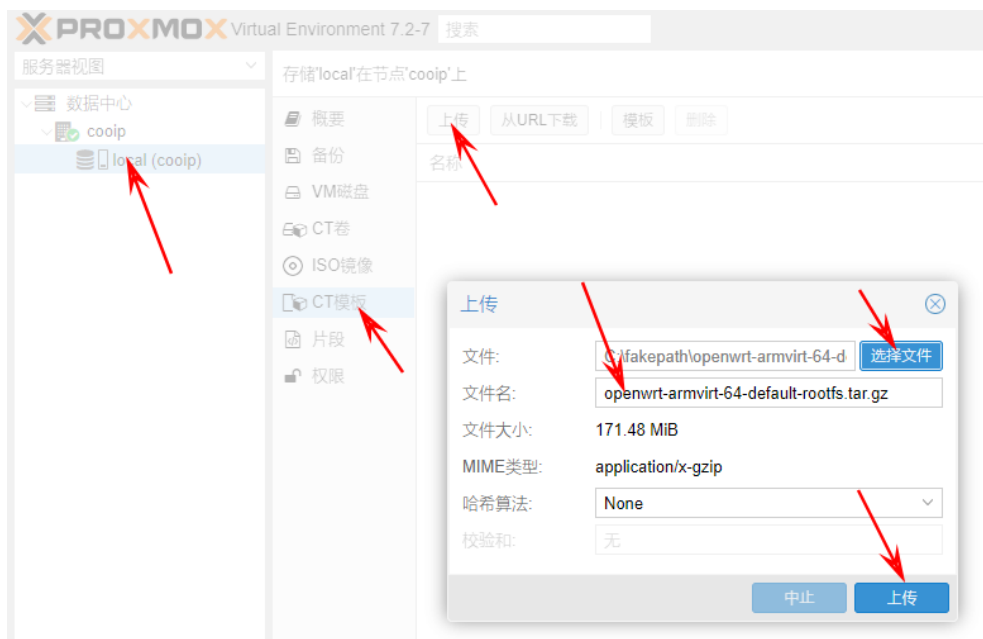
URL:  查询网址

文件名:

文件大小: 171.48 MiB MIME类型: application/octet-stream

高级 ☐ 下载





## 安装 LXC OpenWRT

```
pct create 119 \
  local:vztmpl/openwrt-armvirt-64-default-rootfs.tar.gz \
  --rootfs local:4 \
  --ostype unmanaged \
  --hostname openwrt \
  --arch arm64 \
  --cores 8 \
  --memory 512 \
  --swap 0 \
  --net0 bridge=vmbri0,name=eth0
```

```
cooip:~:~#
cooip:~:~#
cooip:~:~#
cooip:~:~#
cooip:~:~# pct create 119 \ ID 号, 纯数字, 不可重复
  local:vztmpl/openwrt-armvirt-64-default-rootfs.tar.gz \ 模板文件local为磁盘名称
  --rootfs local:4 \ 分配该LXC磁盘大小, 之后可以在PVE界面扩大, 根据自己需求来
  --ostype unmanaged \
  --hostname openwrt \
  --arch arm64 \
  --cores 8 \ 内核数量, 内存大小
  --memory 512 \
  --swap 0 \
  --net0 bridge=vmbri0,name=eth0
Formatting '/var/lib/vz/images/119/vm-119-disk-0.raw', fmt=raw size=4294967296 preallocation=off
Creating filesystem with 1048576 4k blocks and 262144 inodes
Filesystem UUID: 2fd59fe9-1fd8-4e42-a975-1dc853d2ac46
Superblock backups stored on blocks:
  32768, 98304, 163840, 229376, 294912, 819200, 884736
extracting archive '/var/lib/vz/template/cache/openwrt-armvirt-64-default-rootfs.tar.gz'
Total bytes read: 495216640 (473MiB, 101MiB/s)
cooip:~:~#
```

修改 LXC OP 配置文件

nano /etc/pve/lxc/119.conf

```
GNU nano 5.4 /etc/pve/lxc/119.conf *
arch: arm64
cores: 8
features: nesting=1
hostname: openwrt
memory: 512
net0: name=eth0,bridge=vbr0,hwaddr=2A:B7:0C:DD:86:55,type=veth
ostype: unmanaged
rootfs: local:119/vm-119-disk-0.raw,size=4G
swap: 0
lxc.cgroup2.devices.allow: c 108:0 rwm
lxc.cgroup2.devices.allow: c 10:200 rwm
lxc.mount.entry: /dev/ppp dev/ppp none bind,create=file #旁网关这个PPP拨号可以不要
lxc.mount.entry: /dev/net/tun dev/net/tun none bind,create=file

#####直通第二块网卡
lxc.net.1.type: phys
lxc.net.1.link: enP2p33s0
lxc.net.1.flags: up
lxc.net.1.name: eth1
```

打开嵌套



启动 LXC Op

pct start 119

进入

lxc-attach 119

修改 IP 地址

nano /etc/config/network



```

extracting archive '/var/lib/vz/template/cache/openwrt-armvirt-64-default-rootfs.tar.gz'
Total bytes read: 495216640 (473MiB, 101MiB/s)
cooip:~:~# nano /etc/pve/lxc/119.conf
cooip:~:~# pct start 119
cooip:~:~# lxc-attach 119

BusyBox v1.36.0 (2023-03-24 14:28:07 UTC) built-in shell (ash)

~ # nano /etc/config/network

```

IP 地址不要与目前局域网其他设备冲突

```

GNU nano 7.2 /etc/config/network
1
2 config interface 'loopback'
3     option ifname 'lo'
4     option proto 'static'
5     option ipaddr '127.0.0.1'
6     option netmask '255.0.0.0'
7
8 config globals 'globals'
9     option ula_prefix 'fd4d:089f:19fb::/48'
10
11 config interface 'lan'
12     option type 'bridge'
13     option ifname 'eth0'
14     option proto 'static'
15     option ipaddr '192.168.0.22'
16     option netmask '255.255.255.0'
17     option ip6assign '60'
18
19 config interface 'vpn0'
20     option ifname 'tun0'
21     option proto 'none'
22
23

```

修改 root 密码

Passwd

重启网络

/etc/init.d/network restart

```

~ # passwd
Changing password for root
New password:
Retype password:
passwd: password for root changed by root
~ # /etc/init.d/network restart
~ #

```

Web 访问刚才修改好的 IP 和密码

# OpenWrt®

- 状态
  - 概览
  - 防火墙
  - 路由表
  - 系统日志
  - 内核日志
  - 系统进程
  - 实时信息
  - 实时监控
  - 释放内存
- 系统
- 服务
- 网络存储
- VPN
- 网络
- 退出

## 状态

### 系统

主机名	openwrt
主机型号	Radxa ROCK 5B (CpuMark : 98320.360508 Scores)
架构	ARMv8 Processor rev 0 x 8
固件版本	OpenWrt R23.3.3 / LuCI Master (git-23.082.27304-adda8b2)
内核版本	5.10.110-rockchip-rk3588
本地时间	Sun Mar 26 15:03:17 2023
运行时间	0h 42m 56s
平均负载	3.45, 2.56, 1.14
CPU 使用率 (%)	0 %

### 内存

可用数	5582 MB / 7689 MB (72%)
已缓存	28 MB / 7689 MB (0%)

哎，安慰奖 10W 都上不去。

本安装过程不限于 3588, 已经成功在 N1、905X3、s922x 等设备上测试，都可以顺畅安装  
使用的是 OPHUB 这个仓库的 Armbian\_bullseye 镜像  
<https://github.com/ophub/amlogic-s9xxx-armbian/releases>

## 日志优化

/var/log/ 目录下 syslog daemon.log 总是写被满，LXC 容器不断输出大量日志。

应该是 BUG，但不影响使用。直接关闭日志。

先释放被写满的硬盘空间

```
echo "" > /var/log/syslog
```

```
echo "" > /var/log/daemon.log
```

编辑配置文件

```
nano /etc/rsyslog.conf
```

下面两行注释掉

```
#,auth,authpriv.none -/var/log/syslog
```

```
#daemon.* -/var/log/daemon.log
```

重启 syslog 服务器

```
service syslog restart
```

```
#####  
#### RULES ####  
#####  
  
#  
# First some standard log files.  Log by facility.  
#  
auth,authpriv.*          /var/log/auth.log  
#*.*;auth,authpriv.none  -/var/log/syslog  
#cron.*                  /var/log/cron.log  
#daemon.*                -/var/log/daemon.log  
kern.*                    -/var/log/kern.log  
lpr.*                     -/var/log/lpr.log  
mail.*                    -/var/log/mail.log  
user.*                   -/var/log/user.log
```



























其他 LXC 模板下载

这里比较多（没有精力和能力全部测试）

<https://us.lxd.images.canonical.com/images/>

← → ↻ 🔒 us.lxd.images.canonical.com/images/

## Index of /images

Name	Last modified	Size
 <a href="#">Parent Directory</a>		-
 <a href="#">almalinux/</a>	2022-06-01 00:11	-
 <a href="#">alpine/</a>	2022-12-03 15:40	-
 <a href="#">alt/</a>	2021-08-02 03:26	-
 <a href="#">amazonlinux/</a>	2021-12-03 15:40	-
 <a href="#">apertis/</a>	2022-03-05 15:40	-
 <a href="#">archlinux/</a>	2021-01-20 23:31	-
 <a href="#">busybox/</a>	2021-11-18 15:40	-
 <a href="#">centos/</a>	2022-01-30 15:40	-
 <a href="#">debian/</a>	2022-07-18 15:40	-
 <a href="#">devuan/</a>	2021-10-27 20:52	-
 <a href="#">fedora/</a>	2023-01-19 15:40	-
 <a href="#">funtoo/</a>	2023-01-23 17:11	-
 <a href="#">gentoo/</a>	2021-01-21 00:17	-
 <a href="#">kali/</a>	2021-11-01 19:10	-
 <a href="#">mint/</a>	2023-01-04 09:17	-
 <a href="#">openeuler/</a>	2022-12-09 16:32	-
 <a href="#">opensuse/</a>	2023-01-19 15:40	-
 <a href="#">openwrt/</a>	2022-11-11 15:40	-
 <a href="#">oracle/</a>	2022-08-25 08:10	-
 <a href="#">plamo/</a>	2021-11-01 19:10	-
 <a href="#">pld/</a>	2021-01-21 00:59	-
 <a href="#">rockylinux/</a>	2022-07-20 03:27	-
 <a href="#">springdalelinux/</a>	2022-11-18 08:19	-
 <a href="#">ubuntu/</a>	2023-03-07 18:24	-
 <a href="#">voidlinux/</a>	2021-11-01 19:10	-

Apache/2.4.29 (Ubuntu) Server at us.lxd.images.canonical.com Port 443

现在以 Debian 的 LXC 模板举例安装 Adguardhome 和 OpenMediaVault,其它的自行研究。

## Index of /images/debian/bullseye/arm64/default

Name	Last modified	Size
Parent Directory	-	-
SHA256SUMS	2023-03-26 06:27	732
SHA256SUMS.asc	2023-03-26 06:27	833
build.log	2023-03-26 06:25	74K
build.log.asc	2023-03-26 06:27	833
delta-20230325_05:49.qcow2.vcdiff	2023-03-26 06:27	191M
delta-20230325_05:49.qcow2.vcdiff.asc	2023-03-26 06:27	833
delta-20230325_05:49.vcdiff	2023-03-26 06:25	1.0M
delta-20230325_05:49.vcdiff.asc	2023-03-26 06:27	833
disk.qcow2	2023-03-26 06:25	284M
disk.qcow2.asc	2023-03-26 06:27	833
image.yaml	2023-03-26 06:25	61K
image.yaml.asc	2023-03-26 06:27	833
lxd.tar.xz	2023-03-26 06:24	708
lxd.tar.xz.asc	2023-03-26 06:27	833
meta.tar.xz	2023-03-26 06:24	1.0K
meta.tar.xz.asc	2023-03-26 06:27	833
rootfs.squashfs	2023-03-26 06:24	81M
rootfs.squashfs.asc	2023-03-26 06:27	833
rootfs.tar.xz	2023-03-26 06:24	73M
rootfs.tar.xz.asc	2023-03-26 06:27	833

Apache/2.4.29 (Ubuntu) Server at us.lxd.images.canonical.com Port 443

和上面下载 OpenWrt 的模板一样，下载 rootfs.tar.xz 文件

PROXMOX Virtual Environment 7.2-7 搜索

服务器视图

数据中心

cooip

119 (openwrt)

local (cooip)

存储'local'在节点'cooip'上

概要

备份

VM磁盘

CT卷

ISO镜像

CT模板

片段

权限

上传

从URL下载

模板

删除

搜索: 命名, 格式

名称	日期	格式
openwrt-armvirt-64-default-rootfs.tar.gz	2023-03-26 14:39:49	tgz

从URL下载

修改文件名，以区别

URL: <https://us.lxd.images.canonical.com/images/debian/bullseye/arm64> 查询网址

文件名: [debian-rootfs.tar.xz](#)

文件大小: 73.29 MiB

MIME类型: application/x-xz

高级 ☐ 下载

创建 CT

PROXMOX Virtual Environment 7.2-7 搜索

文档

创建虚拟机

创建CT

root@pam

存储'local'在节点'cooip'上

帮助

概要

备份

VM磁盘

上传

从URL下载

模板

删除

搜索: 命名, 格式

名称	日期	格式	大小
debian-rootfs.tar.xz	2023-03-26 16:18:18	txz	76.85 MB

创建: LXC容器

常规 模板 磁盘 CPU 内存 网络 DNS 确认

节点: coolp 资源池:   
CT ID: 120 密码:   
主机名: AdguardHome 确认密码:   
无特权的容器: ☒ SSH公钥:   
嵌套: ☒ [加载SSH密钥文件](#)

密码必须填写  
或者使用SSH密钥

帮助 高级 ☐ 返回 下一步

创建: LXC容器

常规 模板 磁盘 CPU 内存 网络 DNS 确认

存储: local   
模板: debian-rootfs.tar.xz

磁盘大小自己定，这个作为演示只分 1G，因为一会还要打包镜像

创建: LXC容器

常规 模板 磁盘 CPU 内存 网络 DNS 确认

rootfs   
存储: local   
磁盘大小 (GiB): 1

创建: LXC容器

常规 模板 磁盘 CPU 内存 网络 DNS 确认

核心: 8

## 创建: LXC容器

常规 模板 磁盘 CPU 内存 网络 DNS 确认

内存 (MiB): 256

交换分区 (MiB): 0

## 创建: LXC容器

常规 模板 磁盘 CPU 内存 网络 DNS 确认

名称: eth0 IPv4: ☒ 静态 ☐ DHCP


MAC地址: auto IPv4/CIDR: 无

桥接: vmbr0 网关 (IPv4):

VLAN标签: no VLAN IPv6: ☒ 静态 ☐ DHCP ☐ SLAAC

速率限制 (MB/s): unlimited IPv6/CIDR: 无

防火墙: ☒ 网关 (IPv6):

 建议关闭

## 数据中心

## cooip

- 119 (openwrt)
- 120 (AdguardHome)
- local (cooip)

## 概要

上传 从URL下载 模板 删除

## 备份

名称

## VM磁盘

debian-rootfs.tar.xz

## Task viewer: CT 120 - 创建

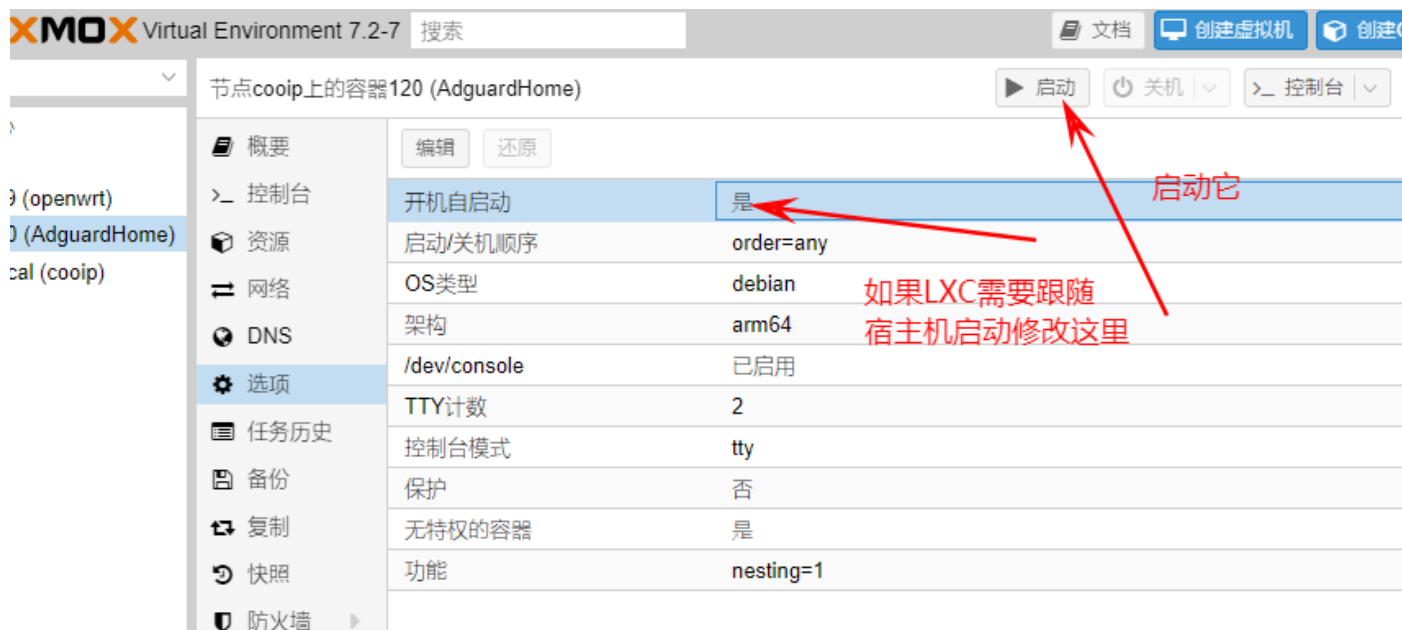
输出 状态

停止

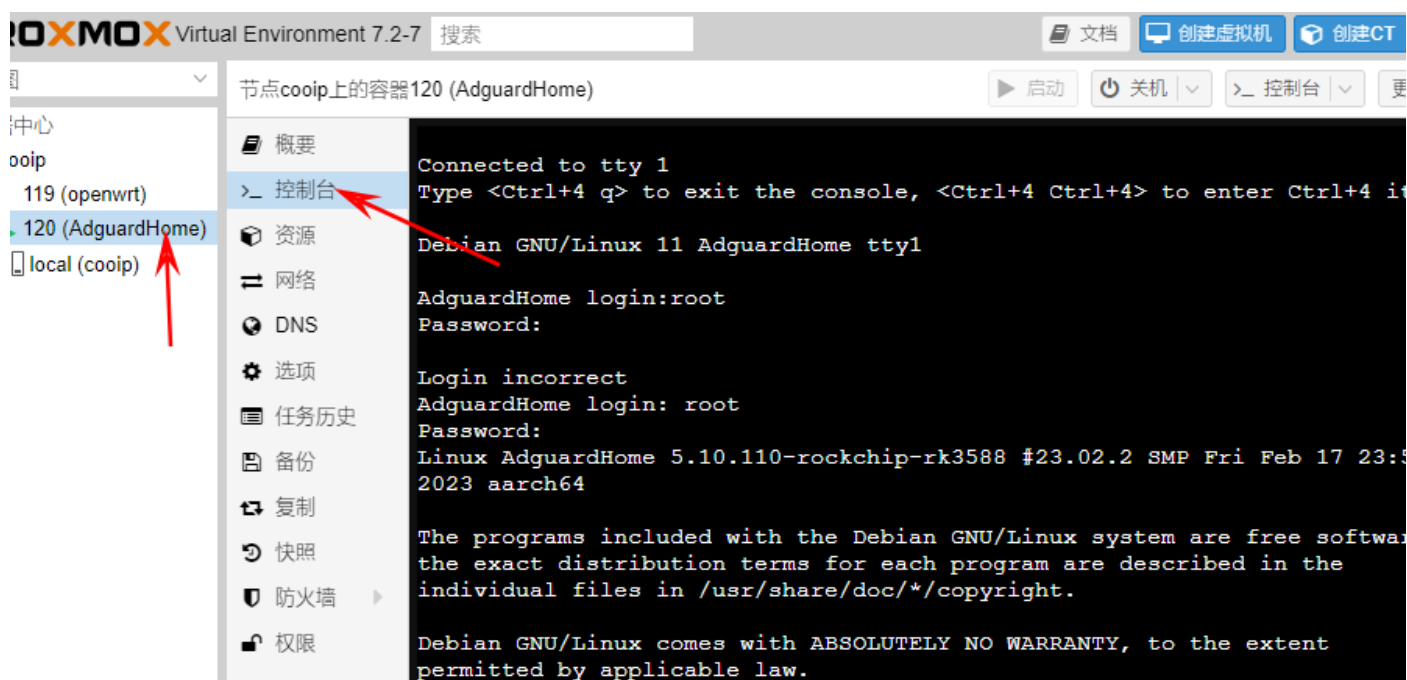
```
Formatting '/var/lib/vz/images/120/vm-120-disk-0.raw', fmt=raw size=1073741824 preallocation=off
Creating filesystem with 262144 4k blocks and 65536 inodes
Filesystem UUID: 560e6512-2ecc-4693-8a65-9fd7ee1daa25
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376
extracting archive '/var/lib/vz/template/cache/debian-rootfs.tar.xz'
Total bytes read: 344668160 (329MiB, 50MiB/s)
Detected container architecture: arm64
Creating SSH host key 'ssh_host_ecdsa_key' - this may take some time ...
done: SHA256:vAp2+yPsh5tOTUdlyx8AxWq/LDdxd+L9Xbrl1SV5rbk root@AdguardHome
Creating SSH host key 'ssh_host_ed25519_key' - this may take some time ...
done: SHA256:edaNP7iJOuU3Iyun40NFazdNQkBp0hkiAFi2jVfHKE4 root@AdguardHome
Creating SSH host key 'ssh_host_rsa_key' - this may take some time ...
done: SHA256:OZSSWlGkHVx72HWy1oLoq8m0W+G6gpIejA1NMhbwnNw root@AdguardHome
Creating SSH host key 'ssh_host_dsa_key' - this may take some time ...
done: SHA256:mS6XIwZc9K5r7iz1mVg1jD3zKbILiHDuG9fRvOxmADA root@AdguardHome
TASK OK
```



LXC-Debian 到此创建完成。



可以在控制台管理该 LXC，也可以通过宿主机的 lxc-attach 120 进入该 LXC，建议用第二种方式



第二种

```
cooip:~:~#
cooip:~:~# lxc-attach 120
root@AdguardHome:~#
```

+++++

换源

vi /etc/apt/sources.list

deb http://mirrors.tuna.tsinghua.edu.cn/debian/ bullseye main contrib non-free

deb http://mirrors.tuna.tsinghua.edu.cn/debian/ bullseye-updates main contrib non-free

deb http://mirrors.tuna.tsinghua.edu.cn/debian/ bullseye-backports main contrib non-free

deb http://mirrors.tuna.tsinghua.edu.cn/debian-security bullseye-security main contrib non-free

更新、安装一些顺手的工具和打开 SSHD 服务

```
apt update && apt install -y openssh-server nano curl wget apt-transport-https ca-certificates
```

```
nano /etc/ssh/sshd_config
```

根据自己使用实际情况调整——这里只是为了演示方便，不代表安全

```
GNU nano 5.4 /etc/ssh/sshd_config
# This sshd was compiled with PATH=/usr/bin:/bin:/usr/sbin:/sbin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes
#prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

配置容器网络-仅供参考

```
nano /etc/systemd/network/eth0.network
```

```
GNU nano 5.4 /etc/systemd/network/eth0.network *
[Match]
Name=eth0
[Network]
Address=192.168.0.95/24
DNS=192.168.0.11
IPForward=yes
IPv6AcceptRA=true
[Route]
Gateway=192.168.0.2
```

```
[Match]
```

```
Name=eth0
```

```
[Network]
```

```
Address=192.168.0.95/24
```

```
DNS=192.168.0.11
```

```
IPForward=yes
```

```
IPv6AcceptRA=true
```

[Route]

Gateway=192.168.0.2

# 重启网络服务

systemctl restart systemd-networkd

# 设置开机启动网络服务

systemctl enable systemd-networkd

```

root@AdguardHome:~# systemctl enable systemd-networkd
root@AdguardHome:~# systemctl restart systemd-networkd
root@AdguardHome:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue s
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00:
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0@if7: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500
    link/ether aa:a8:e0:c7:be:aa brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.95/24 brd 192.168.0.255 scope global
        valid_lft forever preferred_lft forever
    inet6 fd6f:1bef:e5dc:0:a8a8:e0ff:fec7:beaa/64 scope
        valid_lft forever preferred_lft forever

```

安装 AdguardHome

其官方 Github 地址 <https://github.com/AdguardTeam/AdGuardHome>

安装脚本

curl -s -S -L <https://raw.githubusercontent.com/AdguardTeam/AdGuardHome/master/scripts/install.sh> | sh -s -- -v

```

root@AdguardHome:~# curl -s -S -L https://raw.githubusercontent.com/AdguardTeam/AdGuardHome/master/scripts/install.sh | sh -s -- -v
starting AdGuard Home installation script
channel: release
operating system: linux
cpu type: arm64
AdGuard Home will be installed into /opt/AdGuardHome
checking tar
script is executed with root privileges
no need to uninstall
downloading package from https://static.adtidy.org/adguardhome/release/AdGuardHome_linux_arm64.tar.gz → AdGuardHome_linux_arm64.tar.g
z
successfully downloaded AdGuardHome_linux_arm64.tar.gz
unpacking package from AdGuardHome_linux_arm64.tar.gz into /opt
successfully unpacked, contents:
total 26632
-rwxrwxrwx 1 root root 27131904 Mar  9 13:04 AdGuardHome
-rw-rw-rw- 1 root root    587 Mar  9 13:04 AdGuardHome.sig
-rw-r--r-- 1 root root  70640 Mar  9 13:04 CHANGELOG.md
-rw-r--r-- 1 root root  35149 Mar  9 13:04 LICENSE.txt
-rw-r--r-- 1 root root  21563 Mar  9 13:04 README.md
2023/03/26 08:54:09 [info] AdGuard Home, version v0.107.26
2023/03/26 08:54:09 [info] service: control action: install
2023/03/26 08:54:10 [info] service: started
2023/03/26 08:54:10 [info] Almost ready!
AdGuard Home is successfully installed and will automatically start on boot.
There are a few more things that must be configured before you can use it.
Click on the link below and follow the Installation Wizard steps to finish setup.
AdGuard Home is now available at the following addresses:
2023/03/26 08:54:10 [info] go to http://127.0.0.1:3000
2023/03/26 08:54:10 [info] go to http://[::1]:3000
2023/03/26 08:54:10 [info] go to http://192.168.0.95:3000
2023/03/26 08:54:10 [info] go to http://[fd6f:1bef:e5dc:0:a8a8:e0ff:fec7:beaa]:3000
2023/03/26 08:54:10 [info] go to http://[:::3000
2023/03/26 08:54:10 [info] go to http://[fe80::a8a8:e0ff:fec7:beaa%eth0]:3000
2023/03/26 08:54:10 [info] service: action install has been done successfully on linux-systemd
AdGuard Home is now installed and running
you can control the service status with the following commands:
sudo /opt/AdGuardHome/AdGuardHome -s start|stop|restart|status|install|uninstall
root@AdguardHome:~#

```

安装完成

根据提示访问 <http://192.168.0.95:3000>



# 欢迎使用 AdGuard Home!

AdGuard Home 是一个可在特定网络范围内拦截所有广告和跟踪器的 DNS 服务器。它的目的是让您控制整个网络和您的所有设备，且不需要使用任何客户端程序。

开始配置

步骤 1/5



## 网页管理界面

监听接口

端口

eth0 - 192.168.0.95

80

您可以通过以下地址访问您的 AdGuard Home 网页管理界面：

- <http://192.168.0.95>

## DNS 服务器

监听接口

端口

eth0 - 192.168.0.95

53

您将需要使用以下地址来设置您的设备或路由器的 DNS 服务器：

- 192.168.0.95



AdguardHome 安装完毕。



如果有该提示，添加玩上游 DNS 该提示消失

仅供参考，根据自己的网络换进配置

我目前使用的上游 DNS

<https://dns.alidns.com/dns-query>

[tls://dns.alidns.com](https://dns.alidns.com)

[tls://dns.ipv6dns.com](https://dns.ipv6dns.com)

<https://dns.ipv6dns.com/dns-query>

<https://1.1.1.1/dns-query>

Bootstrap DNS 服务器

223.5.5.5

9.9.9.10

149.112.112.10

2620:fe::10

2620:fe::fe:10

配置好 Adguard 之后，可以使用该服务作为所有设备的上游 DNS。

比如上面看到的 192.168.0.11 就是 Adguardhome

相同的套路，现在来安装 **OpenMediaVault**

官方地址：<https://www.openmediavault.org/>

创建: LXC容器

常规 模板 磁盘 CPU 内存 网络 DNS 确认

存储: local

模板: debian-rootfs.tar.xz

创建: LXC容器

常规 模板 磁盘 CPU 内存 网络 DNS 确认

rootfs	存储: local
	磁盘大小 (GiB): 4

可以自己挂载更大的硬盘  
这里是演示，做NAS这个空间肯定太小了



内容可以分的稍微大那么一点点



创建: LXC容器

常规 模板 磁盘 CPU 内存 网络 DNS 确认

内存 (MiB): 2048

交换分区 (MiB): 0

和上面安装 adguardhome 一样，换源、更新、安装工具、修改网络，这里不再赘述。  
或者配置一个基础的 debian 系统，使用 PVE 的克隆功能复制出新的 LXC，这样可以减少重复性工作。



概要 控制台

Clone CT 150

目标节点: cooip 目标存储: local

CT ID: 188

主机名: base-tools

资源池:

帮助 克隆

丰俭由人，根据自己喜好来

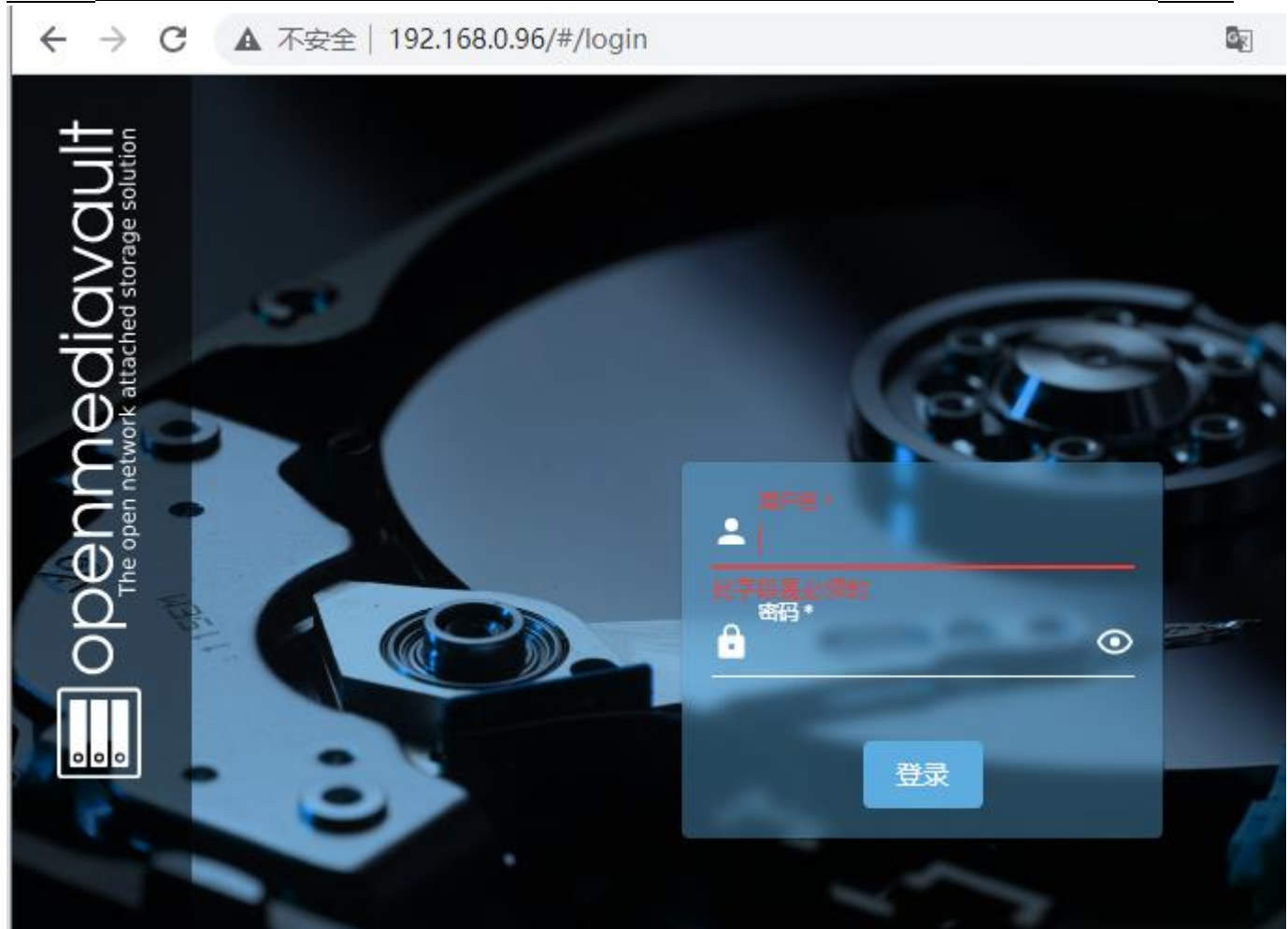
++++++

正式安装

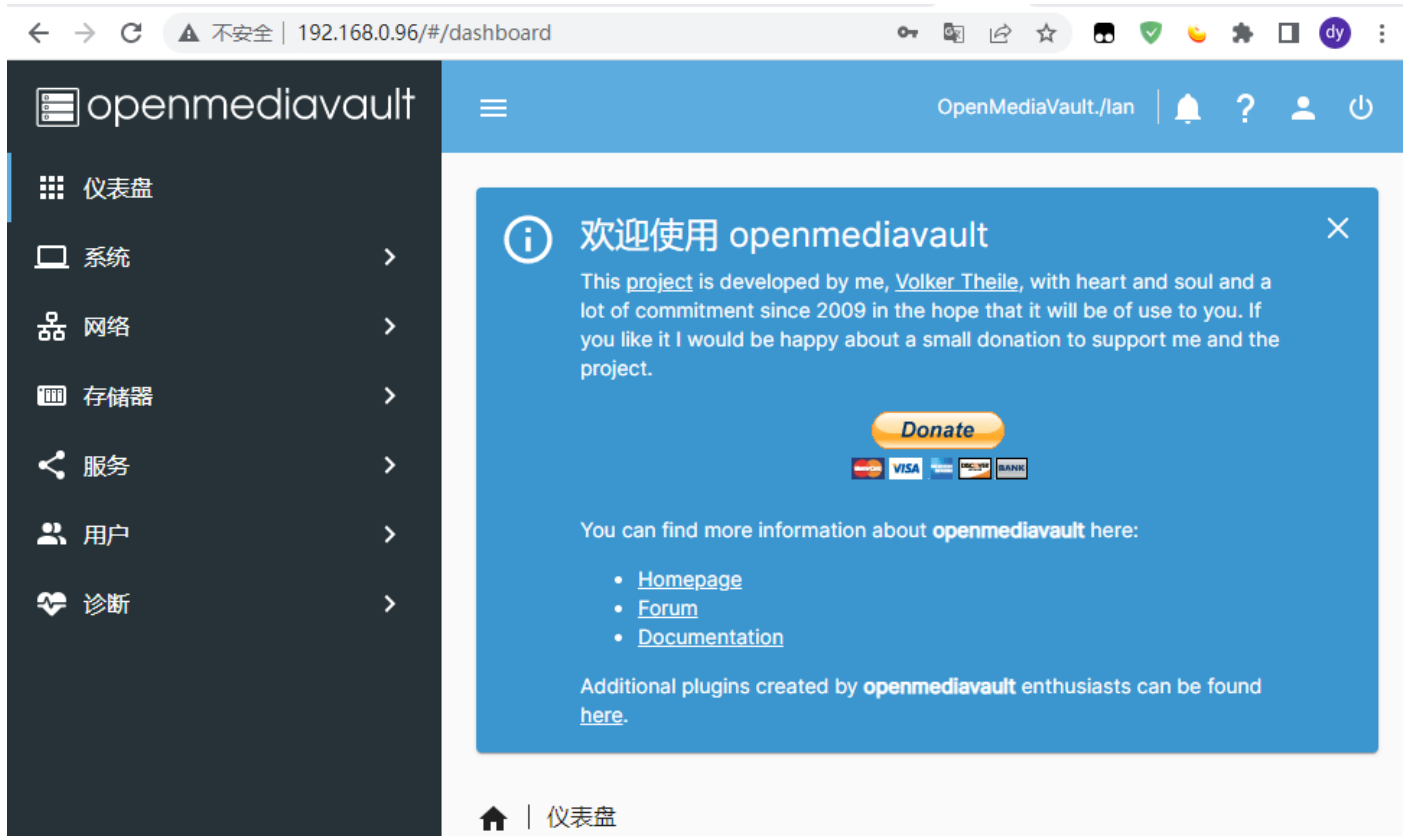
++++++

```
wget -O - https://github.com/OpenMediaVault-Plugin-Developers/installScript/raw/master/install | sudo bash
```

该系统安装时间较长，务必确保科学网络的可靠，不然无法正常安装!!!!



默认 web 管理密码用户名: admin 密码: openmediavault



PVE: <https://192.168.0.93:8006>

用户名: root 密码: 1234

Openwrt:192.168.0.22

用户名: root 密码:password

Adguardhome: 192.168.0.22

User:Rock Password:Rock-5b-Radxa

没用过 PVE 的注意:



### 您的连接不是私密连接

攻击者可能会试图从 **192.168.0.93** 窃取您的信息（例如：密码、通讯内容或信用卡信息）。[了解详情](#)

NET::ERR\_CERT\_AUTHORITY\_INVALID

[高级](#)

[返回安全连接](#)



### 您的连接不是私密连接

攻击者可能会试图从 **192.168.0.93** 窃取您的信息（例如：密码、通讯内容或信用卡信息）。[了解详情](#)

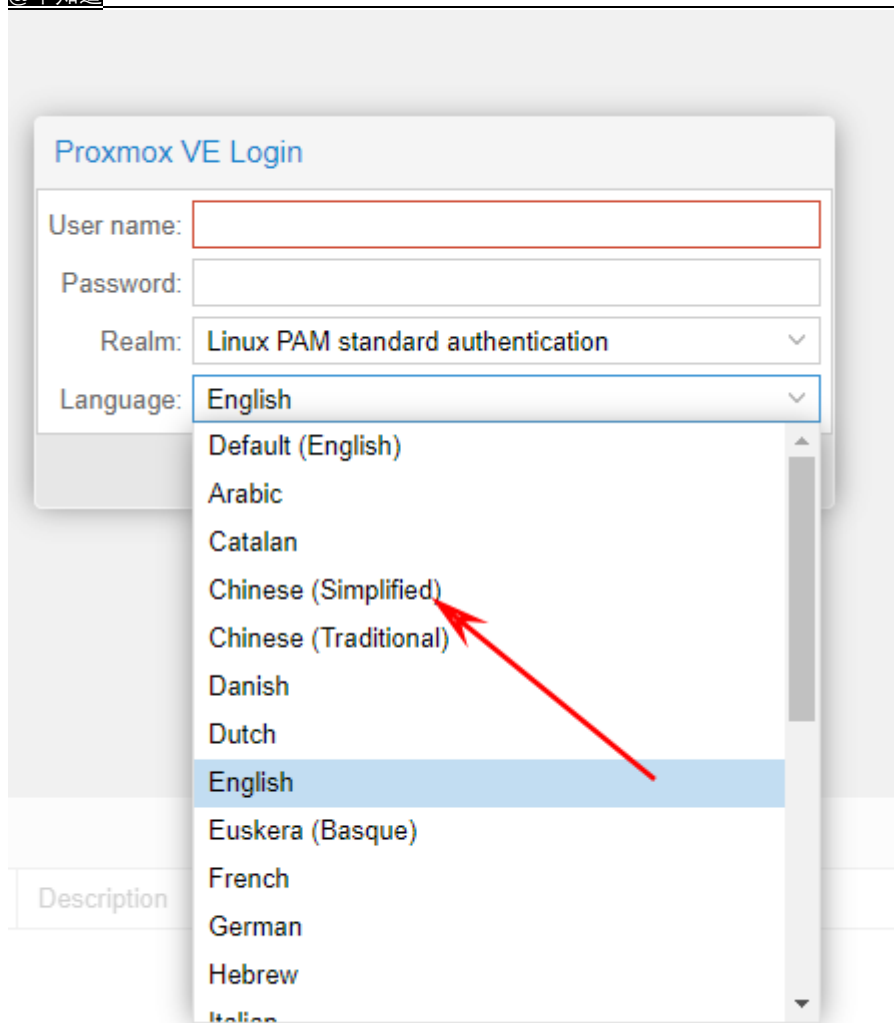
NET::ERR\_CERT\_AUTHORITY\_INVALID

[隐藏详情](#)

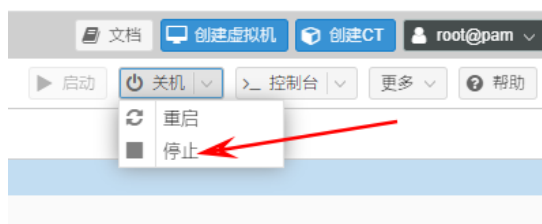
[返回安全连接](#)

此服务器无法证明它是**192.168.0.93**；您计算机的操作系统不信任其安全证书。出现此问题的原因可能是配置有误或您的连接被拦截了。

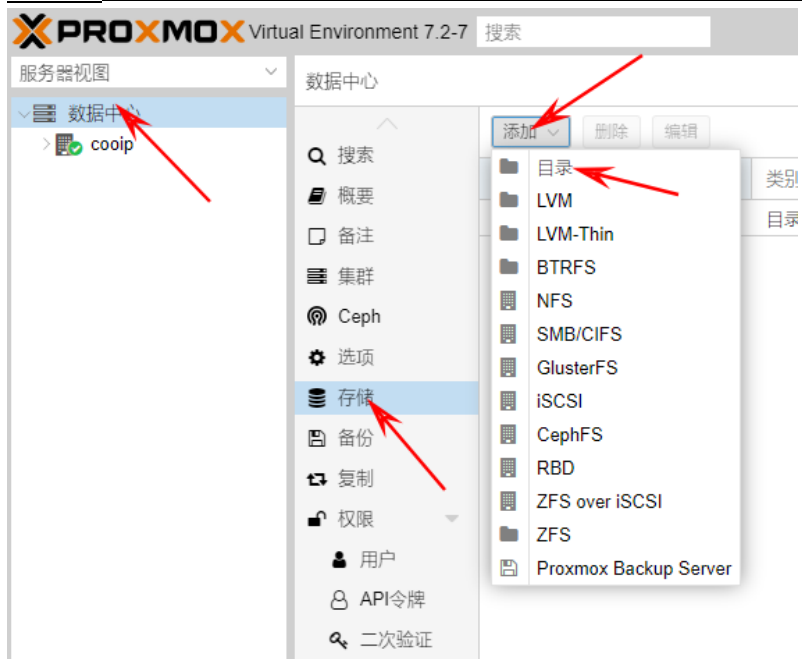
[继续前往192.168.0.93 \(不安全\)](#)



如果需要强制关闭 LXC  
使用

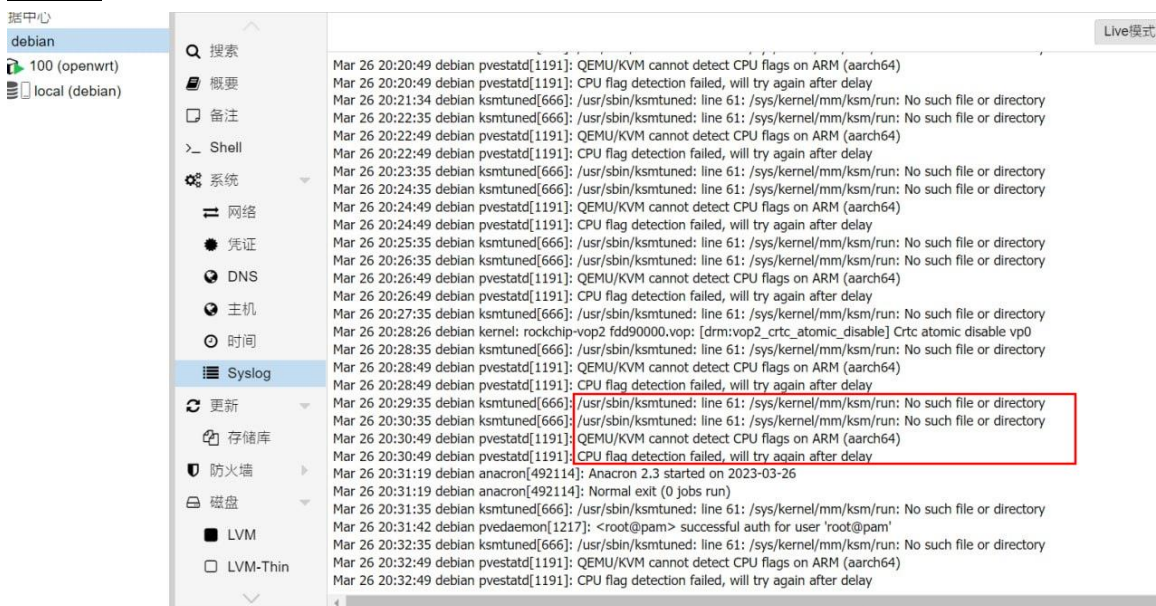


如需要挂载其他磁盘



这些功能模块根据需求添加





Live模式

Mar 26 20:20:49 debian pvestatd[1191]: QEMU/KVM cannot detect CPU flags on ARM (aarch64)  
Mar 26 20:20:49 debian pvestatd[1191]: CPU flag detection failed, will try again after delay  
Mar 26 20:21:34 debian ksmtuned[666]: /usr/sbin/ksmtuned: line 61: /sys/kernel/mm/ksm/run: No such file or directory  
Mar 26 20:22:35 debian ksmtuned[666]: /usr/sbin/ksmtuned: line 61: /sys/kernel/mm/ksm/run: No such file or directory  
Mar 26 20:22:49 debian pvestatd[1191]: QEMU/KVM cannot detect CPU flags on ARM (aarch64)  
Mar 26 20:22:49 debian pvestatd[1191]: CPU flag detection failed, will try again after delay  
Mar 26 20:23:35 debian ksmtuned[666]: /usr/sbin/ksmtuned: line 61: /sys/kernel/mm/ksm/run: No such file or directory  
Mar 26 20:24:35 debian ksmtuned[666]: /usr/sbin/ksmtuned: line 61: /sys/kernel/mm/ksm/run: No such file or directory  
Mar 26 20:24:49 debian pvestatd[1191]: QEMU/KVM cannot detect CPU flags on ARM (aarch64)  
Mar 26 20:24:49 debian pvestatd[1191]: CPU flag detection failed, will try again after delay  
Mar 26 20:25:35 debian ksmtuned[666]: /usr/sbin/ksmtuned: line 61: /sys/kernel/mm/ksm/run: No such file or directory  
Mar 26 20:26:35 debian ksmtuned[666]: /usr/sbin/ksmtuned: line 61: /sys/kernel/mm/ksm/run: No such file or directory  
Mar 26 20:26:49 debian pvestatd[1191]: QEMU/KVM cannot detect CPU flags on ARM (aarch64)  
Mar 26 20:26:49 debian pvestatd[1191]: CPU flag detection failed, will try again after delay  
Mar 26 20:27:35 debian ksmtuned[666]: /usr/sbin/ksmtuned: line 61: /sys/kernel/mm/ksm/run: No such file or directory  
Mar 26 20:28:26 debian kernel: rockchip-vop2 fdd90000.vop: [drm:vop2\_crtc\_atomic\_disable] Crtc atomic disable vp0  
Mar 26 20:28:35 debian ksmtuned[666]: /usr/sbin/ksmtuned: line 61: /sys/kernel/mm/ksm/run: No such file or directory  
Mar 26 20:28:49 debian pvestatd[1191]: QEMU/KVM cannot detect CPU flags on ARM (aarch64)  
Mar 26 20:28:49 debian pvestatd[1191]: CPU flag detection failed, will try again after delay  
Mar 26 20:29:35 debian ksmtuned[666]: /usr/sbin/ksmtuned: line 61: /sys/kernel/mm/ksm/run: No such file or directory  
Mar 26 20:30:35 debian ksmtuned[666]: /usr/sbin/ksmtuned: line 61: /sys/kernel/mm/ksm/run: No such file or directory  
Mar 26 20:30:49 debian pvestatd[1191]: QEMU/KVM cannot detect CPU flags on ARM (aarch64)  
Mar 26 20:30:49 debian pvestatd[1191]: CPU flag detection failed, will try again after delay  
Mar 26 20:31:19 debian anacron[492114]: Anacron 2.3 started on 2023-03-26  
Mar 26 20:31:19 debian anacron[492114]: Normal exit (0 jobs run)  
Mar 26 20:31:35 debian ksmtuned[666]: /usr/sbin/ksmtuned: line 61: /sys/kernel/mm/ksm/run: No such file or directory  
Mar 26 20:31:42 debian pvedaemon[1217]: <root@pam> successful auth for user 'root@pam'  
Mar 26 20:32:35 debian ksmtuned[666]: /usr/sbin/ksmtuned: line 61: /sys/kernel/mm/ksm/run: No such file or directory  
Mar 26 20:32:49 debian pvestatd[1191]: QEMU/KVM cannot detect CPU flags on ARM (aarch64)  
Mar 26 20:32:49 debian pvestatd[1191]: CPU flag detection failed, will try again after delay

日志里如果出现这个，不用担心因为 arm CPU 没那个标志

叨叨点乱七八糟的

为什么使用 LXC，而不使用 VM



Proxmox Virtual Environment 7.2-7 搜索

服务器视图

节点 'rock-5b'

软件包版本

rock-5b (运行时间: 31 天 09:06:53)

CPU利用率 2.13%的8 CPU(s) IO延迟 0.00%

平均负荷 0.58,0.80,0.58

内存使用率 20.78% (1.56 GiB的7.51 GiB) KSM共享 0 B

硬盘空间 9.52% (5.57 GiB的58.51 GiB) SWAP使用率 9.21% (354.00 MiB的3.75 GiB)

CPU(s) 8 x unknown (1 插槽)

内核版本 Linux 5.10.110-rockchip-rk3588 #trunk.0025 SMP Thu Dec 8 15:14:22 UTC 2022

PVE管理器版本 pve-manager/7.2-7/f1b67e2b

存储库状态 未启用Proxmox VE存储库!

CPU利用率

CPU利用率 IO延迟

这个是目前已经上线营业的 5B，它其实已经稳定运行很长时间了。

LXC 比 VM 资源占用少，启动速度和 Docker 一样甚至更快；

宿主机的资源是共享方式给所有 LXC 容器使用，也就是可以最大限度的利用设备的资源，

每个 LXC 等于一个单独的设备，你可以用它做交换机、路由器、服务器、甚至 KODI 播放器（3588 目前不可以，缺



少了一点点东西)

另外一个特点就是可以非常方便的直通各个接口，包括网卡、显卡、键盘、鼠标等；这是 vm 或者 docker 不具备的能力，而且随心可控。

限制使用核心数+指定内核：

cores: 4

cpulimit: 4

lxc.cgroup2.cpuset.cpus: 0,1,2,3 （指定使用处理器 0 ， 1 ， 2 ， 3 内核）

直通硬盘：

mp0: /物理机目录 0,mp=/容器机目录 0,backup=0,replicate=0

mp1: /物理机目录 1,mp=/容器机目录 1,backup=0,replicate=0

mp2: /物理机目录 2,mp=/容器机目录 2,backup=0,replicate=0 （直接挂载硬盘 /dev 路径也行）

直通网卡：

lxc.net.0.link: (http://lxc.net.0.link/) enp1s0f1 （物理机网卡名 1 ）

lxc.net.0.type: phys

lxc.net.1.link: (http://lxc.net.1.link/) enp1s0f2 （物理机网卡名 2 ）

lxc.net.1.type: phys

直通显卡：

lxc.cgroup2.devices.allow: c 226:0 rwm

lxc.cgroup2.devices.allow: c 226:128 rwm

lxc.cgroup2.devices.allow: c 4:7 rwm

lxc.cgroup2.devices.allow: c 29:0 rwm

lxc.mount.entry: /dev/dri/card0 dev/dri/card0 none bind,optional,create=file

lxc.mount.entry: /dev/dri/renderD128 dev/dri/renderD128 none bind,optional,create=file

lxc.mount.entry: /dev/tty7 dev/tty7 none bind,optional,create=file

lxc.mount.entry: /dev/fb0 dev/fb0 none bind,optional,create=file

直通键鼠：

lxc.cgroup2.devices.allow = c 13:\* rwm

lxc.mount.entry: /dev/input dev/input none bind,optional,create=dir

直通声卡：

lxc.cgroup2.devices.allow = c 116:\* rwm

lxc.mount.entry: /dev/snd dev/snd none bind,optional,create=dir

直通之后 LXC 容器内需要安装对应依赖包，比如显卡声卡等

7.2 以后版本使用 cgroup2

---

关于镜像的下载和使用

镜像比较大，强烈建议自行安装。实在安装不下去，或者不会安装的。。。还是用镜像吧。

由于 TG 最大限制 2G 的文件上传，所以我把镜像进行了二次分卷压缩

下载确保

Zip、z01、z02 这三个文件都完全下载。解压后得到 Rock5B-3588-PVE-32G-Tf-Cooip.pmf

按照后面的图示进行镜像文件还原。

Rock5B-3588-PVE-32G-Tf-Cooip.pmf	2023/3/26 20:17	PMFX 文件	2,321,916...
Rock5B-3588-PVE-32G-Tf-Cooip.z01	2023/3/26 20:28	Z01 文件	1,048,576...
Rock5B-3588-PVE-32G-Tf-Cooip.z02	2023/3/26 20:28	Z02 文件	1,048,576...
Rock5B-3588-PVE-32G-Tf-Cooip.zip	2023/3/26 20:28	WinRAR ZIP 压缩...	165,824 KB

由于我使用的是 32G tf 卡作为安装介质，所以。。。。

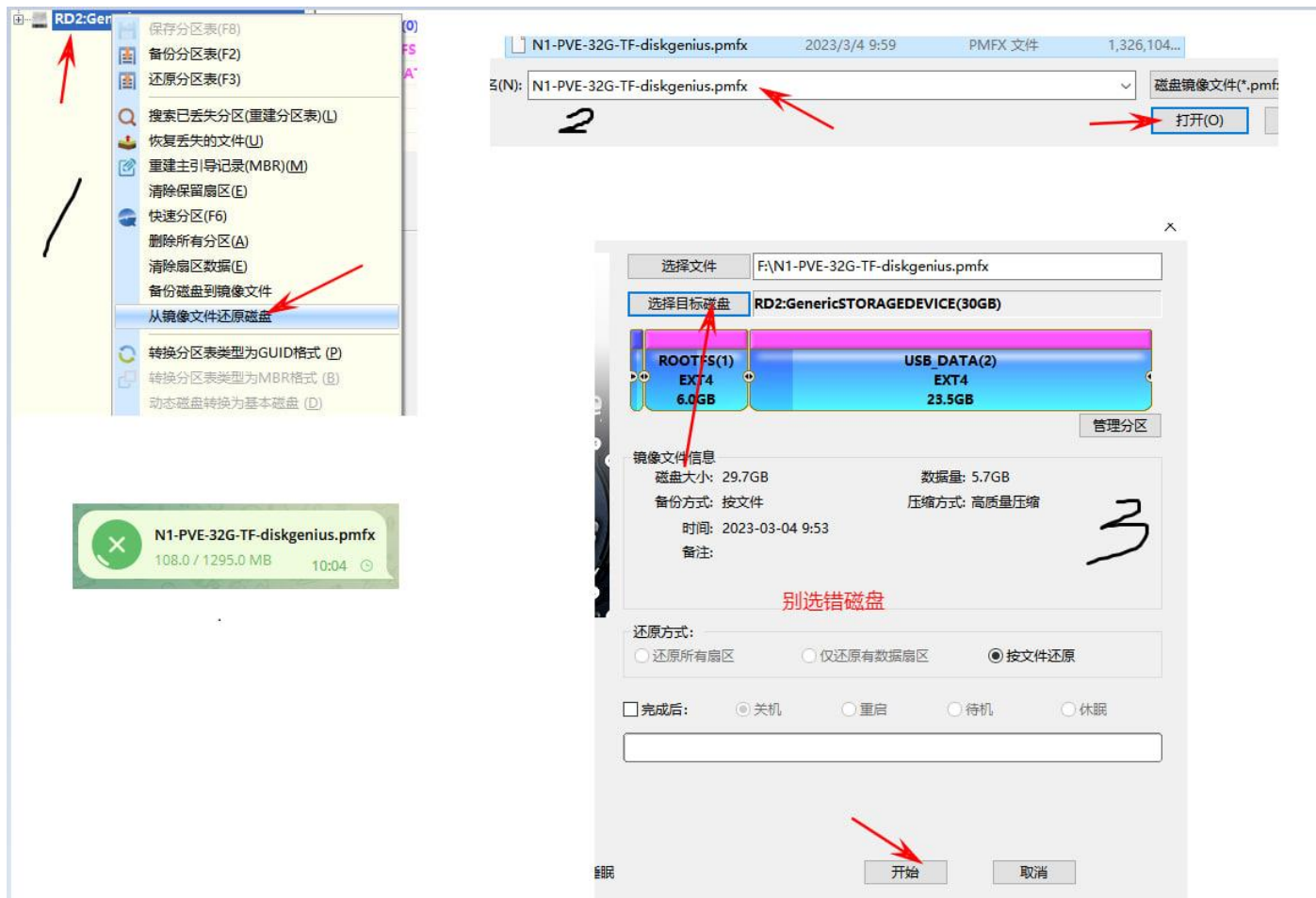
关于如何写入 emmc 或者 nvme 以及如何使用 Nvme 启动，这里不做说明。

平时泡在 OpenWrt-flippy 群

连接: <https://t.me/joinchat/lnJrhXPcuJJiMDdl>

群文件中有我发的 3588 镜像 关键字 PVE

下面的截图是之前给 N1 做镜像的说明，3588 镜像还原也一样，文件名不同，自己注意一下



好了，就到这为止

能力、精力有限，不包“售后”，如果你觉得对你有帮助

