

Proxmox

- LXC
 - Templates
 - Проброс видеокарты
 - Уменьшение размера диска контейнера
- QEMU
 - Принудительная смена разрешения
- Кластер
 - Удаление кластера
- Ошибки
 - AppArmor enabled on system but the docker-default profile could not be loaded: Unable to replace "docker-default". Permission denied; attempted to load a profile while confined?
 - AVC apparmor="DENIED" operation="mount" info="failed flags match" error=-13 profile="lxc-100_</var/lib/lxc>" name="/run/systemd/unit-root/" pid=16153 comm="(ionclean)" srcname="/" flags="rw,
 - Detected Hardware Unit Hang
 - ipcc_send_rec failed: Connection refused
 - RRDC update error /var/lib/rrdcached/db/pve-storage/pve/local-lvm: -1
 - vgremove Failed to update pool
 - Восстановление LVM
 - Зависание хоста при проброшенной карте от AMD
 - Зависла виртуальная машина
 - Неизвестный статус хоста, виртуалок и дисков

- Система
 - Bonding
 - GVT-g
 - Nvidia драйвер для хоста
 - Trim
 - Включение rc.local
 - Изменение swap раздела
 - Смена имени
- Установка гостевых систем
 - Alpine
 - CentOS
 - Ubuntu

LXC

LXC

Templates

Обновить список

```
pveam update
```

Отобразить список

```
pveam available  
pveam available --section system
```

Отобразить установленные

```
pveam list local
```

Скачать XXX

```
pveam download local XXX
```

LXC

Проброс видеокарты

Создать привилегированный контейнер

Отредактировать файл `/etc/pve/lxc/<id>.conf`

```
lxc.cgroup.devices.allow: c 226:0 rwm
lxc.cgroup.devices.allow: c 226:128 rwm
lxc.cgroup.devices.allow: c 29:0 rwm
lxc.mount.entry: /dev/dri dev/dri none bind,optional,create=dir
lxc.mount.entry: /dev/fb0 dev/fb0 none bind,optional,create=file
```

Установить драйвера видеокарты в контейнер

```
apt install vainfo i965-va-driver ffmpeg
```

LXC

Уменьшение размера диска контейнера

Остановить контейнер

```
lxc-stop --name 101
```

Изменить размер диска в файле конфигурации

```
nano /etc/pve/lxc/101.conf
```

Проверить диск на ошибки

```
e2fsck -f /dev/pve/vm-101-disk-0
```

Перераспределить пространство

```
fstrim -v /dev/pve/vm-101-disk-0
```

Изменить размер диска

```
resize2fs /dev/pve/vm-101-disk-0 8G
```

Запустить контейнер

```
lxc-start --name 101
```

В примере используется контейнер номер 101 и новый размер диска - 8Gb

QEMU

QEMU

Принудительная смена разрешения

Virtual-1 - имя виртуального монитора

```
xrandr --newmode "1920x1080_60.00" 173.00 1920 2048 2248 2576 1080 1083 1088 1120 -hsync +vsync  
xrandr --addmode Virtual-1 1920x1080_60.00  
xrandr --output Virtual-1 --mode 1920x1080_60.00
```


Кластер

Кластер

Удаление кластера

Остановить кластер:

```
rm -fr /etc/pve/nodes/*  
systemctl stop pve-cluster  
systemctl stop corosync
```

Запустить кластер снова в одиночном режиме:

```
pmxcfs -l
```

Удалить файлы конфигов:

```
rm /etc/pve/corosync.conf  
rm -r /etc/corosync/*
```

Убить одиночный процесс, лучше несколько раз:

```
killall pmxcfs
```

Удалить список нод с сервера, иначе они будут болтаться там "неживые":

```
rm -fr /etc/pve/nodes
```

Включить кластер:

```
systemctl start corosync  
systemctl start pve-cluster
```

Рестарт веб морды:

```
service pveproxy restart
```

Если необходимо сохранить конфигурацию виртуалок, за backup перед процедурой директорию `/etc/pve/nodes`

Ошибки

AppArmor enabled on system but the docker-default profile could not be loaded: Unable to replace "docker-default". Permission denied; attempted to load a profile while confined?

Остановить контейнер, внести правки, запустить контейнер

XXX - ID нужного LXC контейнера

```
nano /etc/pve/lxc/XXX.conf
```

```
## lxc.apparmor.profile: unconfined
lxc.cgroup.devices.allow: a
lxc.cap.drop:
```

```
AVC apparmor="DENIED"  
operation="mount"  
info="failed flags match"  
error=-13 profile="lxc-  
100_</var/lib/lxc>"  
name="/run/systemd/unit-  
root/" pid=16153  
comm="(ionclean)"  
srcname="/" flags="rw,
```

```
nano /etc/apparmor.d/lxc/lxc-default-cgns
```

```
“ mount fstype=nfs*,  
mount fstype=rpc_pipefs,  
mount options=(rw, nosuid, noexec, remount, relatime, ro, bind, rbind),
```

```
systemctl reload apparmor
```

Detected Hardware Unit Hang

```
apt install ethtool  
nano /etc/default/grub
```

```
GRUB_CMDLINE_LINUX_DEFAULT="pcie_aspm=off"
```

```
update-grub  
ethtool -K eth0 gso off gro off tso off  
nano /etc/network/if-up.d/hangfix-ifup && chmod +x /etc/network/if-up.d/hangfix-ifup
```

```
#!/bin/sh -e  
if [ "$IFACE" = "YOUR-INTERFACE-NAME-HERE" ]; then  
  /sbin/ethtool -K $IFACE gso off gro off tso off tx off rx off  
  pcie_aspm=off  
fi  
exit 0
```

Ошибки

ipcc_send_rec failed: Connection refused

```
/etc/init.d/ntpd restart  
/etc/init.d/pve-cluster restart  
pvecm nodes  
pvecm status
```

RRDC update error

/var/lib/rrdcached/db/pve-storage/pve/local-lvm: -1

```
cd /var/lib/  
systemctl stop rrdcached  
mv rrdcached rrdcached.bck  
systemctl start rrdcached  
systemctl restart pve-cluster
```


Ошибки

vgremove Failed to update pool

```
vg
```

```
vgremove -ff vg
```

Восстановление LVM

Если вы сделали это только что, то удалены только метаданные LVM. У вас должна сохраниться архивная копия метаданных. Обычно их находят в директории `/etc/lvm/archive/[имя vg]_[0-9]+.vg`, выбрать из них нужный (это обычные текстовые файлы в которых записывается конфигурация LVM) и восстановить группу томов с помощью команды

```
vgcfgrestore -f /etc/lvm/archive/<имя vg>_<цифровое значение>.vg <имя vg>
```

Например

```
vgcfgrestore -f /etc/lvm/archive/main_00007-587941649.vg main
```

Затем нужно активировать группу

```
lvm vgchange -ay
```

Зависание хоста при проброшенной карте от AMD

```
cd /opt
apt install pve-headers
apt install git dkms build-essential
git clone https://github.com/gnif/vendor-reset.git
cd vendor-reset
dkms install .
echo "vendor-reset" >> /etc/modules
update-initramfs -u
modprobe vendor-reset
reboot
```

Зависла виртуальная машина

Посмотреть список виртуальных машин

```
qm list
```

Остановить виртуальную машину (VMID поменять на ID виртуальной машины из списка)

```
qm stop VMID
```

Посмотреть PID процесса виртуальной машины (VMID поменять на ID виртуальной машины из списка)

```
ps aux | grep "/usr/bin/kvm -id VMID"
```

Убить виртуальную машину (PID поменять на процесс виртуальной машины)

```
kill -9 PID
```

Неизвестный статус хоста, виртуалок и дисков

```
systemctl status -l pvestatd  
journalctl -r  
systemctl restart pvestatd
```

Система

Bonding

- Round-robin (balance-rr): Передача сетевых пакетов в последовательном порядке от первого доступного связанного сетевого интерфейса (NIC) до последнего. Этот режим обеспечивает балансировку нагрузки и отказоустойчивость.
- Active-backup (active-backup): Только один из связанных в bond сетевых интерфейсов активен. Другой связанный интерфейс становится активным, если и только если активный интерфейс выходит из строя. MAC-адрес одного логического связанного интерфейса виден извне только на одном сетевом адаптере (порту), чтобы избежать искажений в сетевом коммутаторе. Этот режим обеспечивает отказоустойчивость.
- XOR (balance-xor): Передача сетевых пакетов на основе [(source MAC address XOR'd with destination MAC address) modulo NIC slave count]. Это выбирает тот же ведомый сетевой адаптер для каждого MAC-адреса назначения. Этот режим обеспечивает балансировку нагрузки и отказоустойчивость.
- Broadcast (broadcast): Передает сетевые пакеты по всем связанным сетевым интерфейсам. Этот режим обеспечивает отказоустойчивость.
- IEEE 802.3ad Dynamic link aggregation (802.3ad)(LACP): Создает группы агрегации с одинаковыми настройками скорости и дуплекса. Использует все связанные сетевые интерфейсы в активной группе агрегаторов в соответствии со спецификацией 802.3 ad.
- Adaptive transmit load balancing (balance-tlb): Linux bonding driver mode, который не требует специальной поддержки сетевых коммутаторов. Исходящий сетевой пакетный трафик распределяется в соответствии с текущей нагрузкой (вычисленной относительно скорости) на каждом ведомом устройстве сетевого интерфейса. Входящий трафик получен одним в настоящее время назначенным ведомым сетевым интерфейсом. Если этот принимающий интерфейс отказывает, другой связанный интерфейс берет на себя MAC-адрес отказавшего принимающего интерфейса.
- Adaptive load balancing (balance-alb): Включает balance-tlb plus receive load balancing (rlb) для трафика IPV4 и не требует специальной поддержки сетевых коммутаторов. Балансировка нагрузки на прием достигается путем согласования ARP. Драйвер связи перехватывает ответы ARP, отправленные локальной системой на их выходе, и перезаписывает исходный аппаратный адрес с уникальным аппаратным адресом одного из подчиненных сетевых карт в едином логическом связанном интерфейсе так, чтобы различные сетевые узлы использовали различные MAC-адреса для своего сетевого пакетного трафика.

Система

GVT-g

Загружаем модули

```
echo "vfio" >> /etc/modules-load.d/modules.conf
echo "vfio_iommu_type1" >> /etc/modules-load.d/modules.conf
echo "vfio_pci" >> /etc/modules-load.d/modules.conf
echo "vfio_virqfd" >> /etc/modules-load.d/modules.conf
echo "kvmgt" >> /etc/modules-load.d/modules.conf
```

Вносим изменения в загрузчик

```
nano /etc/default/grub
```

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet intel_iommu=on kvm.ignore_msrs=1
i915.enable_gvt=1 i915.enable_execlists=0 drm.debug=0"
```

Обновляем конфиги и модули

```
update-grub && update-initramfs -u -k all
```

Перезапускаем сервер

В свойствах виртуальной машины (предварительно ее отключив) раздел процессора приводим к виду

```
cpu: host,hidden=1
```

Теперь добавляем PCI устройство с видеокартой и отключаем виртуальную

Система

Nvidia драйвер для хоста

Добавить дополнительные репозитории в /etc/apt/sources.list

```
deb http://ftp.us.debian.org/debian bullseye main contrib non-free
deb http://ftp.us.debian.org/debian bullseye-updates main contrib non-free
deb http://security.debian.org bullseye-security main contrib
```

Обновить пакеты и систему

```
apt update
apt upgrade
```

Установить заголовки ядра

```
apt install pve-headers
```

Установить драйвер и необходимые пакеты

```
apt install libnvidia-cfg1 nvidia-kernel-source nvidia-kernel-common nvidia-driver
```

Перезагрузить систему

```
reboot
```

Проверить установку драйвера

```
nvidia-smi
```

Система

Trim

Для автоматизации операции trim для физических дисков, а также для lxc контейнеров выполняем ряд операций ниже.

Создаем скрипт:

```
nano /opt/pve_trim.sh
```

Вносим в него:

```
#!/bin/bash

FSTRIM=/sbin/fstrim

for i in $(pct list | awk '/^[0-9]/ {print $1}'); do
    pct fstrim "$i" 2>&1 | logger -t "pct fstrim [$i]"
done

$FSTRIM -av 2>&1 | logger -t "fstrim [$i]"
```

После создаем задание:

```
nano /etc/cron.d/pve_trim
```

Вносим в него следующее:

```
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

0 6 * * 1 /opt/pve_trim.sh
```

Все. Теперь каждый понедельник в 6 утра будет запускаться сначала трим по всем контейнерам, после чего трим физических дисков сервера.

P.S. Не ставить задание чаще, чем раз в неделю!

Включение rc.local

```
nano /etc/systemd/system/rc-local.service
```

```
“ [Unit]
   Description=/etc/rc.local Compatibility
   ConditionPathExists=/etc/rc.local

   [Service]
   Type=forking
   ExecStart=/etc/rc.local start
   TimeoutSec=0
   StandardOutput=tty
   RemainAfterExit=yes
   SysVStartPriority=99

   [Install]
   WantedBy=multi-user.target
```

```
nano /etc/rc.local
```

```
“ #!/bin/bash
   exit 0
```

```
chmod +x /etc/rc.local
systemctl enable --now rc-local
```

Система

Изменение swar раздела

Отключить подкачку

```
swapoff -v /dev/pve/swap
```

Увеличить размер раздела

```
lvm lvresize /dev/pve/swap -L +8G
```

Отметить раздел как swar

```
mkswap /dev/pve/swap
```

Включить swar

```
swapon -va
```

Смена имени

Отредактировать файлы

```
“ /etc/hostname  
  /etc/hosts  
  /etc/postfix/main.cf  
  /etc/mailname (if you're using this)  
  /etc/pve/storage.cfg  
  /var/lib/rrdcached/db/pve2-storage  
  /var/lib/rrdcached/db/pve2-node  
  /etc/pve/nodes/
```

Перезапустить сервер

Установка гостевых систем

Alpine

```
sudo apk add udev
sudo rc-update add udev sysinit
sudo rc-update add udev-trigger
sudo rc-update add udev-settle
sudo rc-update add udev-postmount
sudo apk add qemu-guest-agent
sudo rc-update add qemu-guest-agent
```

CentOS

```
sudo yum upgrade  
sudo yum install qemu-guest-agent open-vm-tools rng-tools  
sudo systemctl enable qemu-guest-agent  
sudo systemctl start qemu-guest-agent  
sudo reboot
```


Ubuntu

```
sudo apt update
sudo apt install qemu-guest-agent open-vm-tools rng-tools
sudo systemctl enable qemu-guest-agent
sudo systemctl start qemu-guest-agent
sudo reboot
```