

Перечень проделанных работ по восстановлению работы сервера в ООО «Рога и копыта»

Все наименования, рассмотренные в нашем примере вымышленные, случаи возможных совпадений с какими-либо настоящими юр. лицами не преднамерены.

Некоторые не большие фирмы, как рассмотренная в нашем примере **ООО «Рога и копыта»** не представляют в полной мере значимости сервера для жизни предприятия (ведения проводок, сдачи отчётности и, обслуживание сложных и дорогих программных комплексов, как скажем *Лоцман:PLM 2018, компании «Аскон»*) и обслуживающей его штатной единицы – **системного администратора**, поэтому при его утрате или увольнении не ищут замены, а стараются экономить средства и возлагать всю нагрузку на хрупкие плечи девочки программиста, переименовывая её профессию в универсала. Вся эта система продолжает поддерживаться в течении некоторого времени до наступления полного краха ввиду того, что как выясняется позднее та самая девочка *не понимала изначально основных принципов и азов той профессии, что на неё возложили в довесок к основной работе*. Девочка, конечно, старается, и успешно ведёт журнал куда заботливо вписывает *все пароли пользователей и админские* в продолжении тех, полученных от предшественников. Но зачем они нужны, не ясно – вот не задача... Зачем разграничивать права доступа, зачем создавать сетевые учётные записи, когда *проще было бы пустить всех админами и пусть делают что хотят*. Даже удобнее; попросил кто-то пароль – **вон тетрадка, возьми и посмотри**. Где уж там служба безопасности... Вроде бы сервер находится под управлением не безизвестной всем операционной системы *Microsoft Windows*, и пароли в тетрадке, не надо ничего придумывать: нужны права администратора домена – да пожалуйста. **Великолепно**. И всё бы ничего, да ключевая проблема *Microsoft Windows* в том, что она как *подвержена вирусам, так и большинство прошивальщиков работают в ней безотказно для любого оборудования*.

В случае сбоев, при методах работы, построенных схожим образом, как и в нашей маленькой фирме, никого обвинять не стоит; это **только рассуждение**, но попав в руку **неразборчивого пользователя прошивальщик** может оказать **непоправимый урон** тому оборудованию, на которое он установлен будь то *RAID-контроллер* или *материнская плата* или что угодно, а тем более, если его когда-то забыл удалить из учётной записи администратора предшественник нашего программиста.

Это было лирическое отступление, но суть проблемы, поведанная мне оказалась глубже: Со слов специалиста (*нашей девочки универсала*), на предприятии отключили свет, работа сервера была завершена корректно, на работающих ИБП, а после восстановления электричества ничего не стартует, как быть? Приезжай помоги, заплатим за работу.

На вопрос, что произошло вразумительного ответа нет; подключите монитор и пришлите фотографии того, что происходит на экране приходит следующее:

**RAID Adapter
Firmware Failed Validation!!!
Adapter needs to be reflashed.
Press Any Key to Continue...**

Пересборка *RAID-массива* и его переинициализацией грозит утратой всего...

Подготовительные работы

1) Этап:

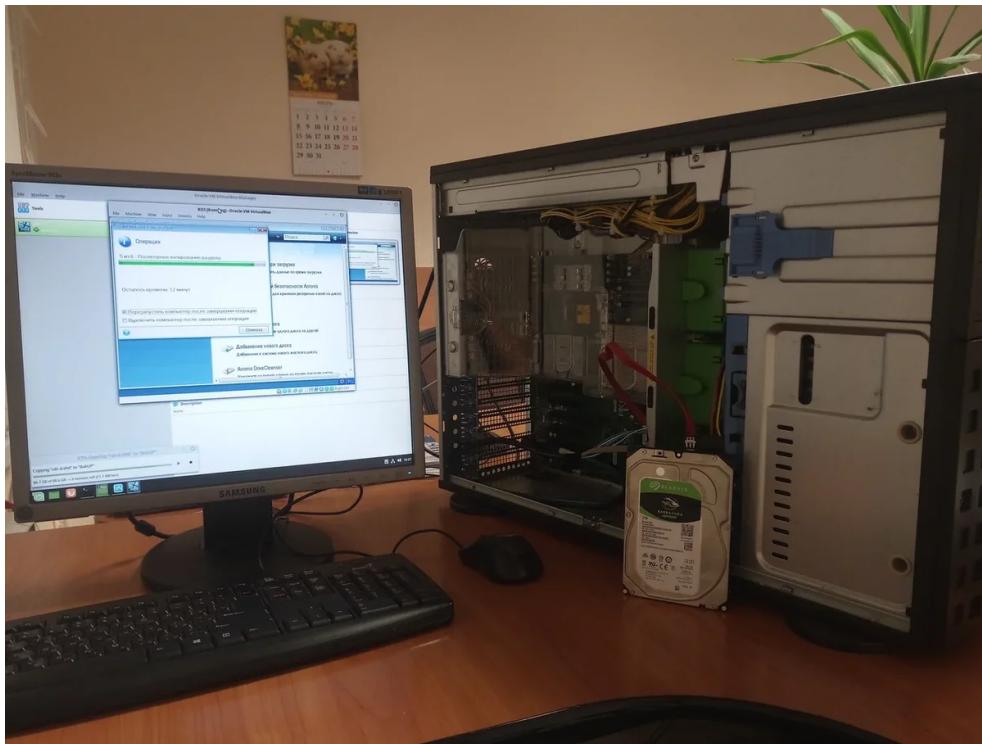
К счастью с *HDD*, включенных в состав **RAID** массива извлечь данные труда не составило, стоила эта операция порядка 12000 рублей в ценах этого календарного года. Информация

даже оказалась избыточной. Затем пришлось приобрести дополнительный *HDD* куда было отклоноировано содержимое **RAID-10**. Ну а остальное дело техники.

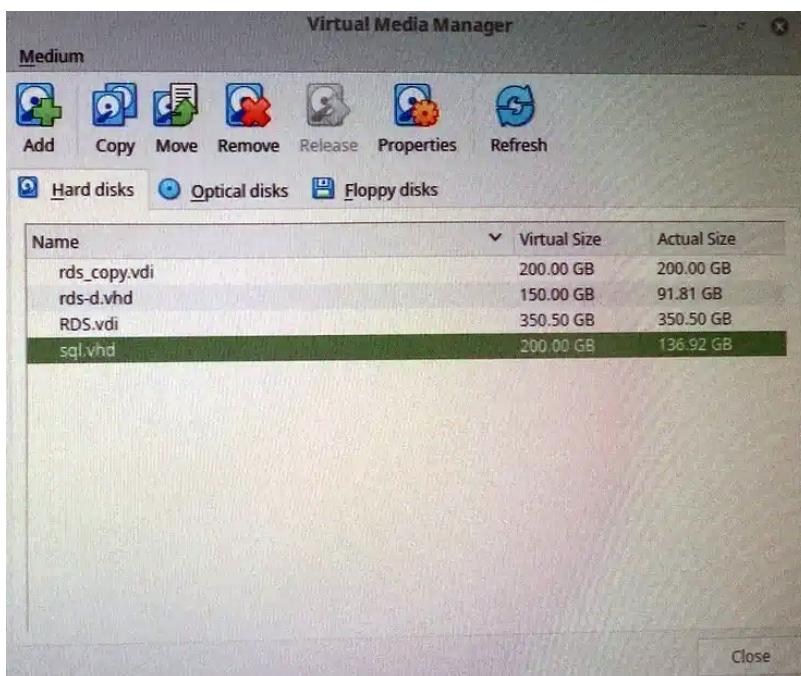
Дабы не наступать на грабли девушки универсала-программиста, принял решение о переводе сервера на *Датацентр ProxmoxVE*. Как известно, *ProxmoxVE* «из коробки» умеет работать с виртуальными машинами в нескольких режимах. Мы рассматриваем вариант побочного переноса виртуальных *HDD* на *RAID10*, как устройство.

Подготовка образов дисков *Microsoft Hyper-V* к переносу в Датацентр *ProxmoxVE v5.4-3*

а) объединение 2-х физических файлов жёстких дисков динамического размера *.vhd в 1 дисковый файл фиксированного размера *.vdi (*сервер Remote Desktop Server*) Диски *RDS* зачем-то были представлены 2-я независимыми файлами. Каждый имел *MBR*, поэтому содержимое второго из них приходилось переносить без неё, и выверять после объединения. Для этой операции использовал *Acroins TrueImage 2016*



б) конвертация динамического файла жёсткого диска (далее по тексту **HDD**) *.vhd → *.vdi (фиксированного размера) (*сервер SQL*) через функции копирования менеджера виртуальных носителей *VirtualBox*.



в) Конвертация файлов **HDD RDS.VDI → RDS.RAW, SQL.VDI → SQL.RAW**

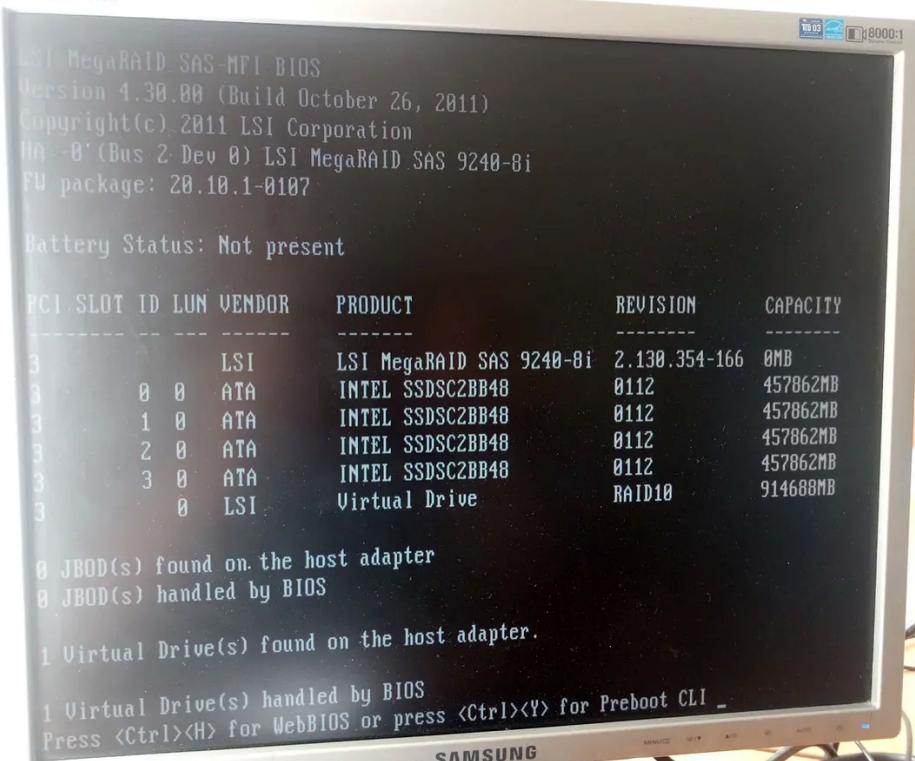
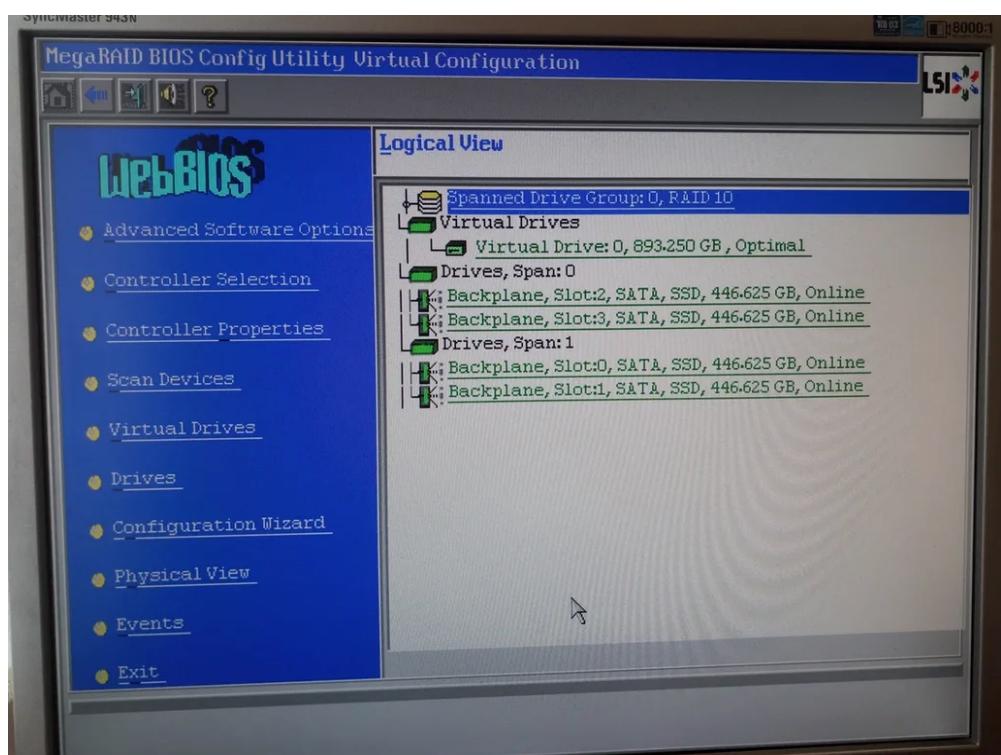
Эта процедура легче всего осуществима в Linux {Ubuntu,Debian,Mint} с использованием **QEMU** конвертора. Пример использования:

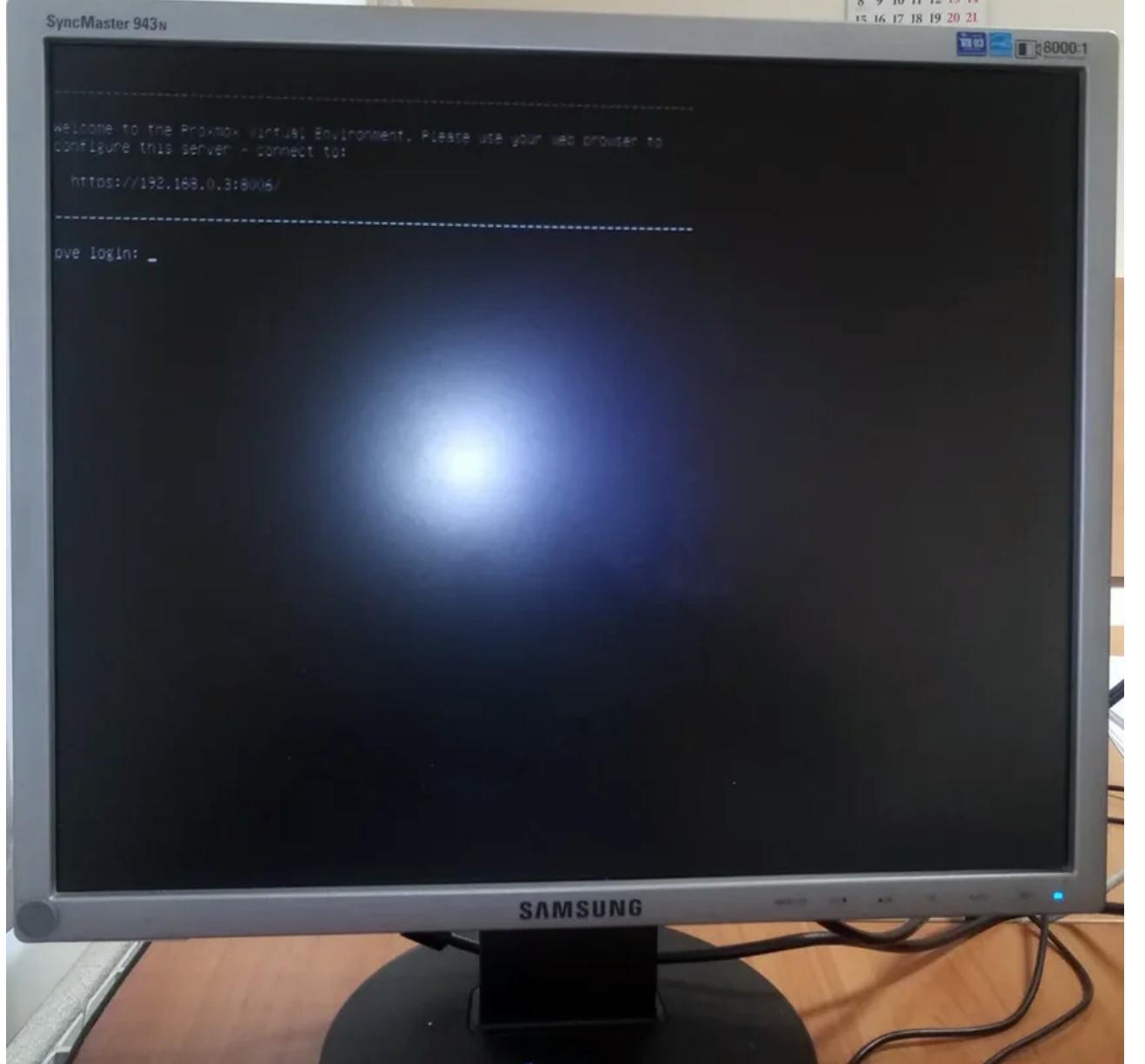
```
sudo apt-get install qemu-utils  
qemu-img convert f vdi vm-100-disk1.vdi -O raw vm-100-disk-1.raw
```

На этом подготовительный этап переноса сервера завершён, переходим ко 2 этапу.

2) Этап:

- Установка в сервер нового **RAID-контроллера LSI MegaRAID SAS 9240-8i**.
- Настройка и инициализация RAID-массива 10 (1+0) из имеющихся в сервере 4-х SSD.
- установка корневой операционной системы **Датацентр ProxmoxVE 5.4-3** на проинициализированный RAID10.





После того, как сервер завёлся остаётся выполнить ещё несколько шагов по его полному восстановлению:

г) Создание шаблонов и серверов и распаковка заготовленных образов жёстких дисков в них. (см. пункт в этапа 1)

На этом этапе нужно отметить, что хоть ProxmoxVE и хорош, но при своей штатной разметке монтирует директорию `/var/lib/vz/dump` на основной системный раздел, за которым резервирует не более 103 ГБ, что негативно сказывается на размере размещённых в нём дампов серверов, тем более если речь идёт о побочном копировании Microsoft Windows серверов, а новая установка исключена, ввиду использования дорогостоящего программного обеспечения как Лоцман:Полином от компании Аскон.

Но простой вариант решения – Выделить отдельный HDD или его раздел большой ёмкости, разметить его в `ext4` либо `xfs` разместить в него заготовленные в первом этапе образы `*.raw`; монтировать его через `.bashrc` из консоли `root` в директорию `/var/lib/vz/dump` либо через `fstab` (но для ежедневной непрерывной работы он мне не нужен, поэтому я остановился на `.bashrc`) и выполнять с ним операции по бэкапированию/восстановлению серверов. Если не все корзины сервера заполнены дисками «до отказа», его можно разместить во 2 корзину и расключить в матплату. Как мне кажется, подобный подход к делу положительно сказался бы на отказоустойчивости системы в целом, т. к. при выходе из строя RAID-контроллера, на отдельном HDD останутся дампы виртуальных машин (далее по тексту ВМ); а в случае отказа HDD, просто придётся разметить новый, промонтировать и выполнить резервное копирование.

```
/root/.bashrc      [-M--] 38 L:[ 1+18 19/ 20] *(608 / 609b) 0010 0x00A
# ``.bashrc` executed by bash(1) for non-login shells.

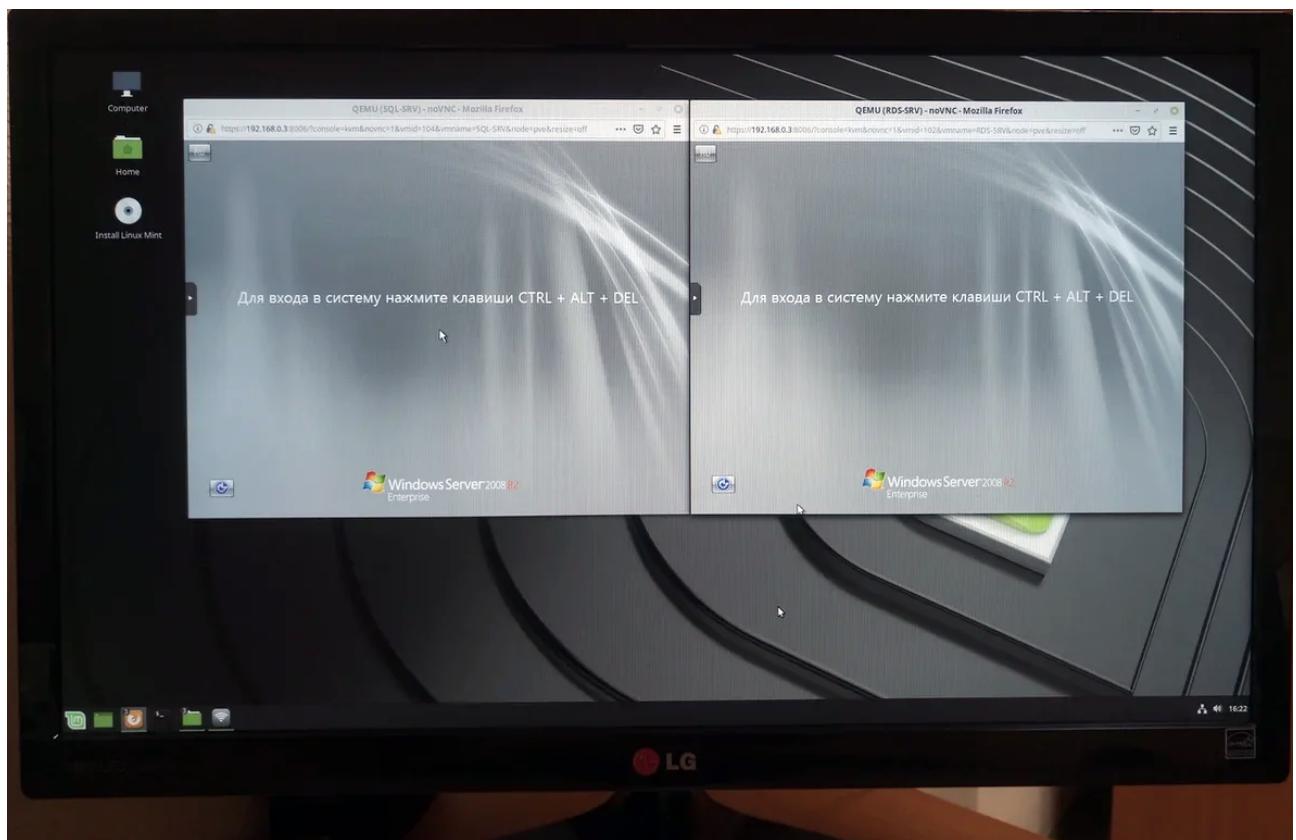
# Note: PS1 and umask are already set in /etc/profile. You should not
# need this unless you want different defaults for root.
# PS1='${debian_chroot:+($debian_chroot)}\n:\n\$ '
# umask 022

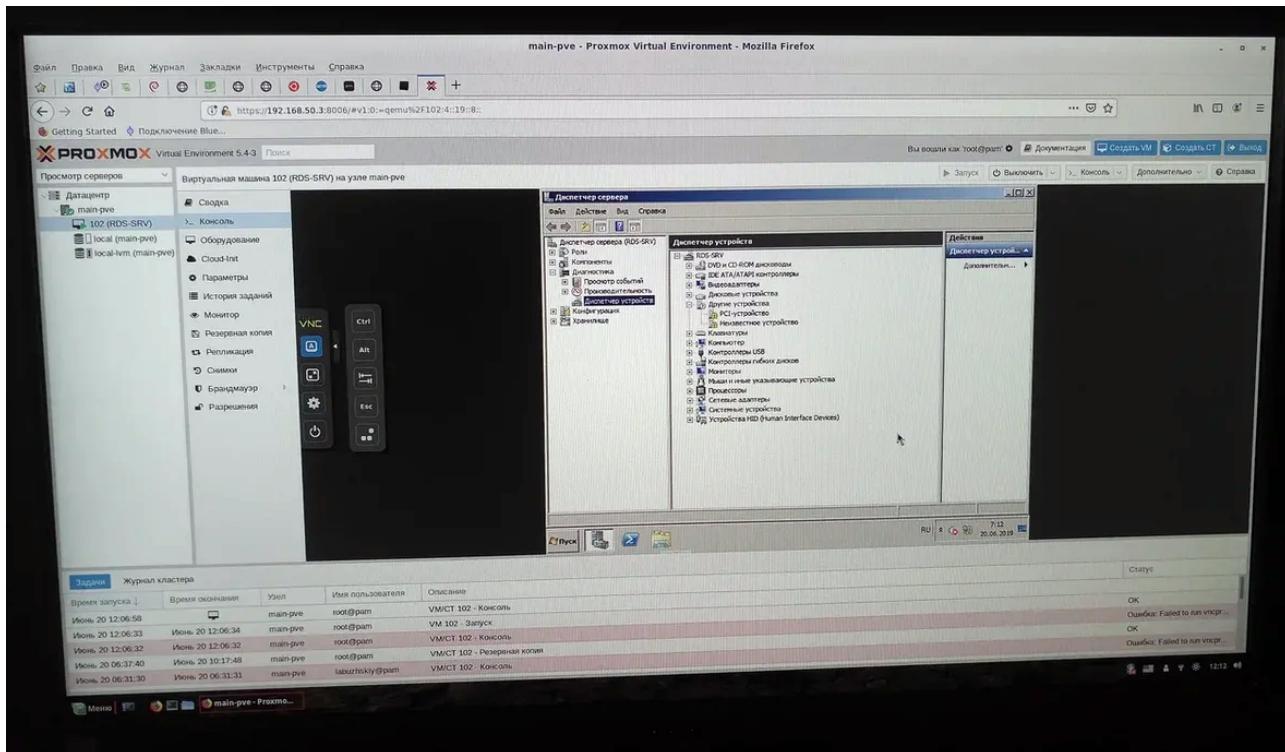
# You may uncomment the following lines if you want `ls` to be colorized:
# export LS_OPTIONS='--color=auto'
# eval "$(dircolors)"
# alias ls='ls $LS_OPTIONS'
# alias ll='ls $LS_OPTIONS -l'
# alias l='ls $LS_OPTIONS -la'

# Some more alias to avoid making mistakes
# alias rm='rm -i'
# alias cp='cp -i'
# alias mv='mv -i'

mount -o rw /dev/sdb4 /var/lib/Vz/dump
```

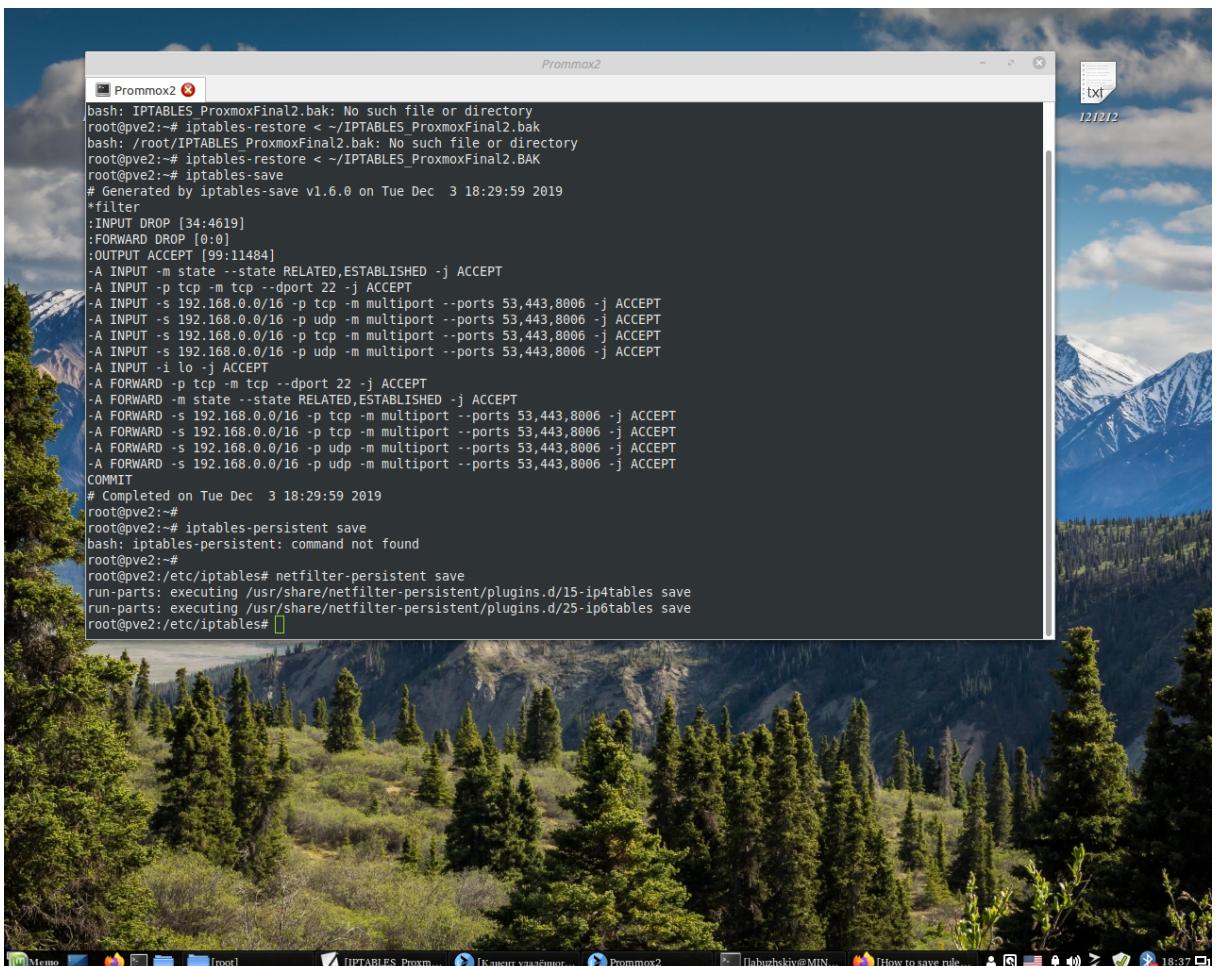
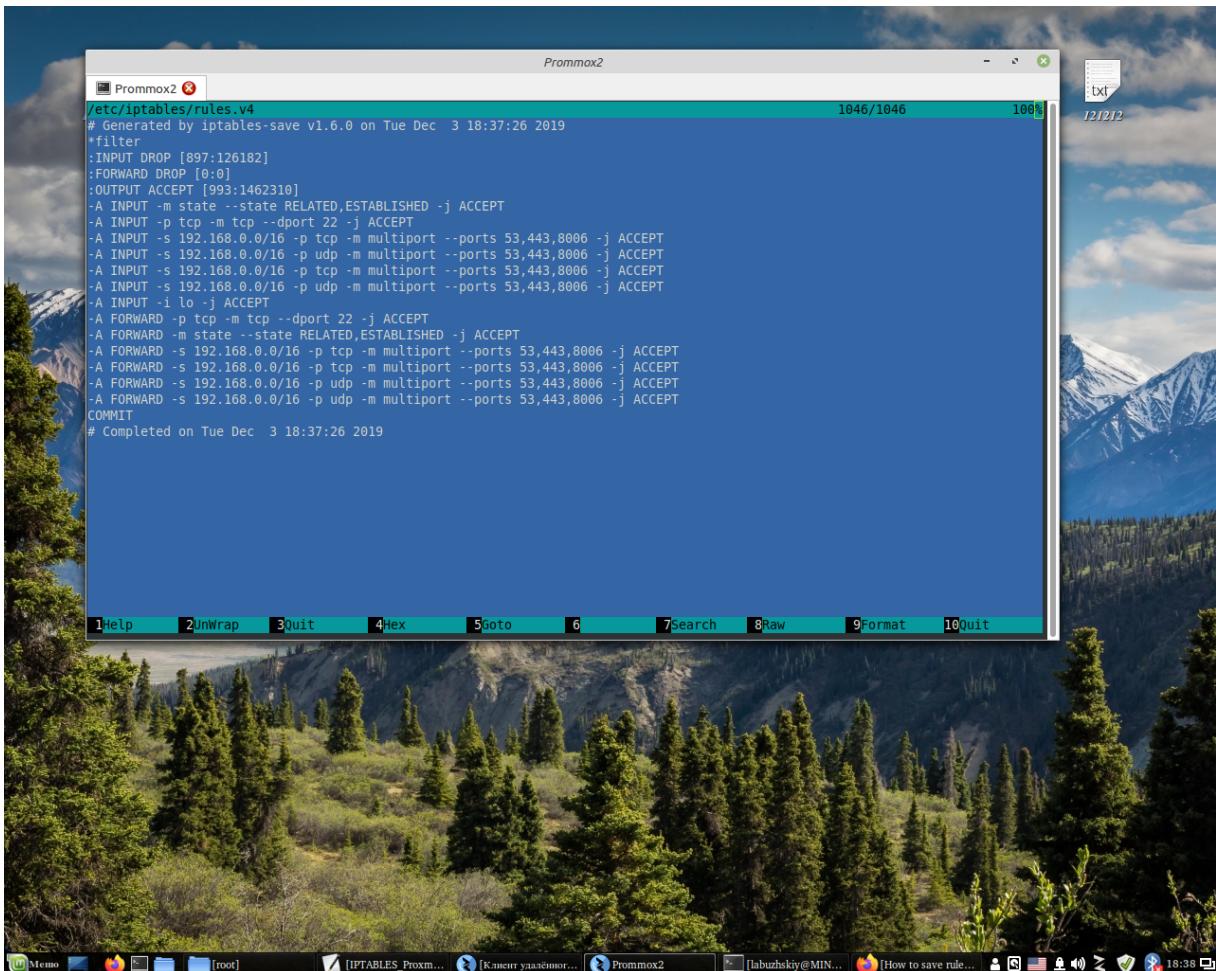
д) пробный старт серверов, установка драйверов контролеров с диска **Virtio-Win.iso**

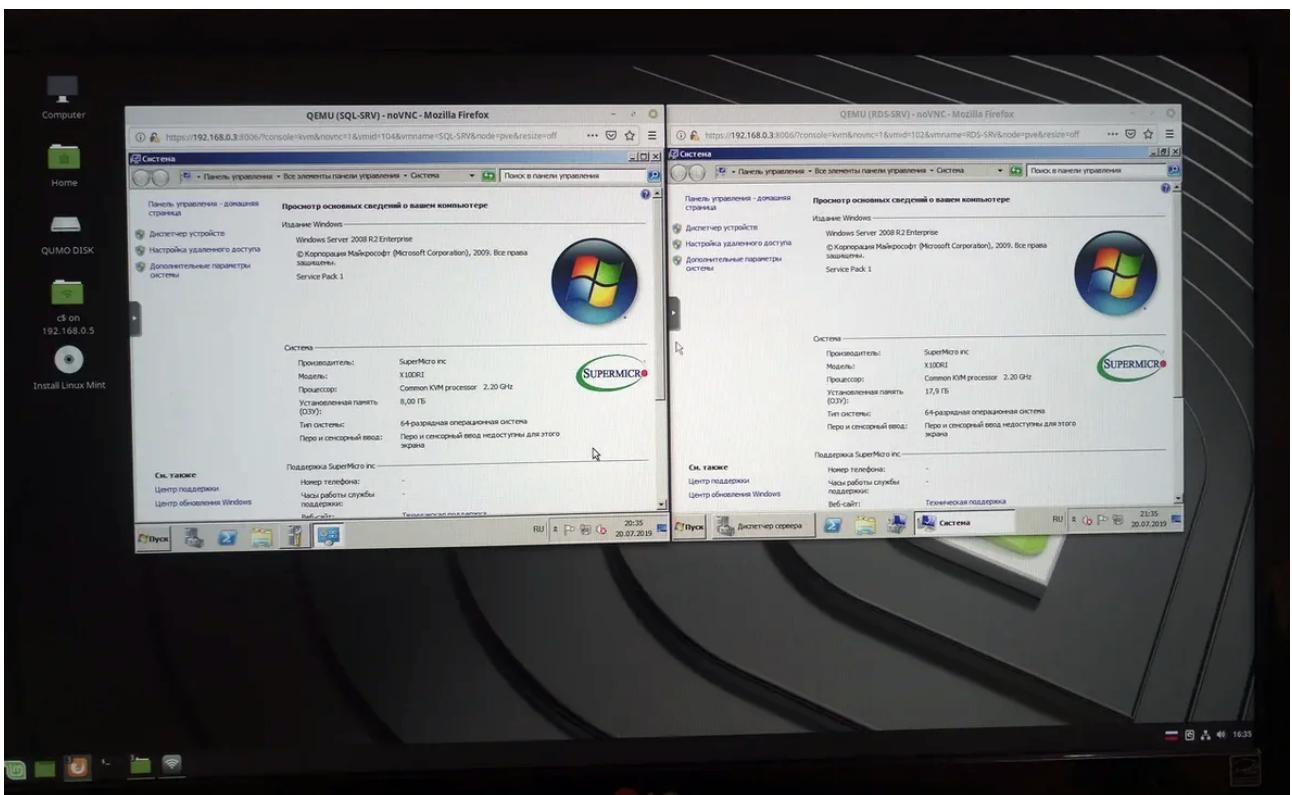
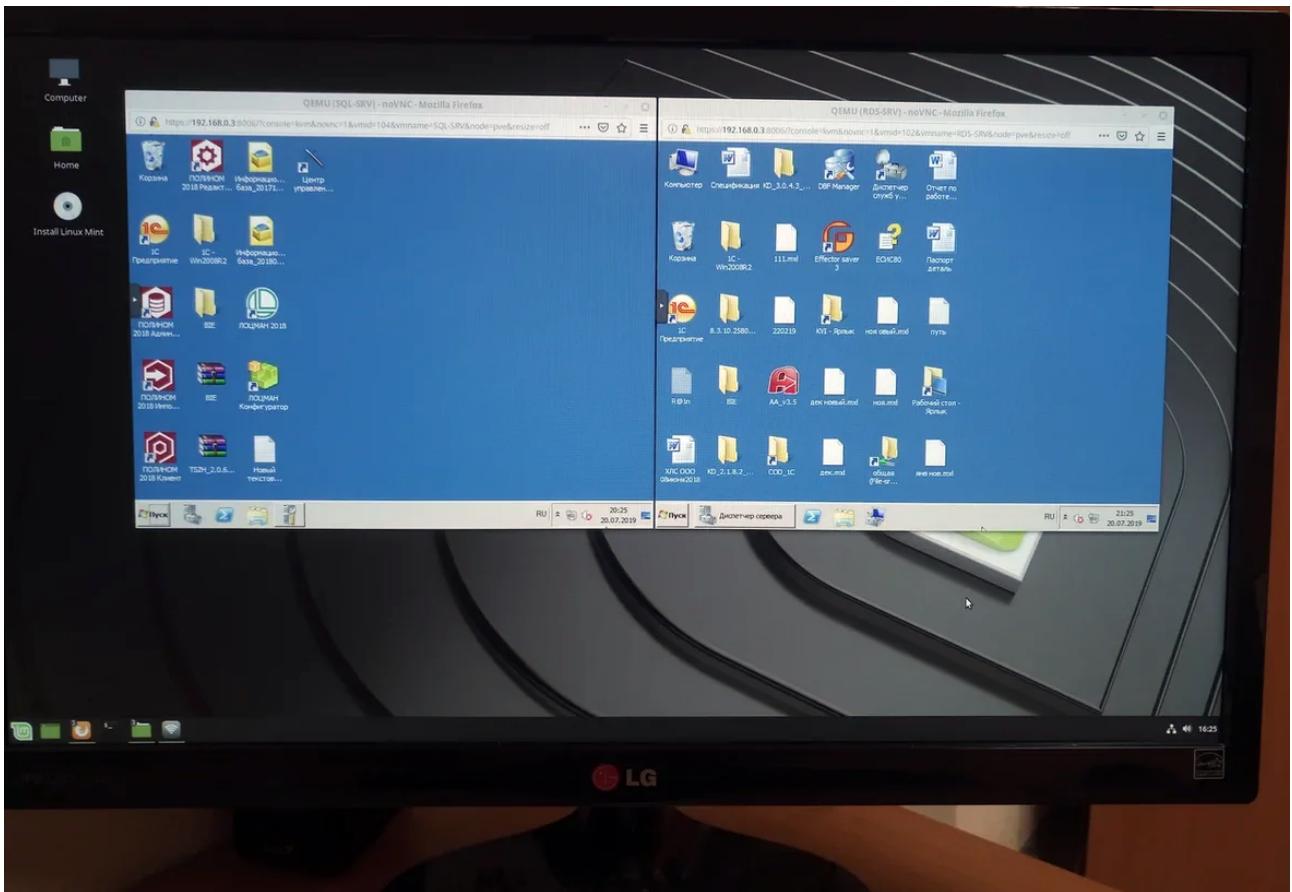




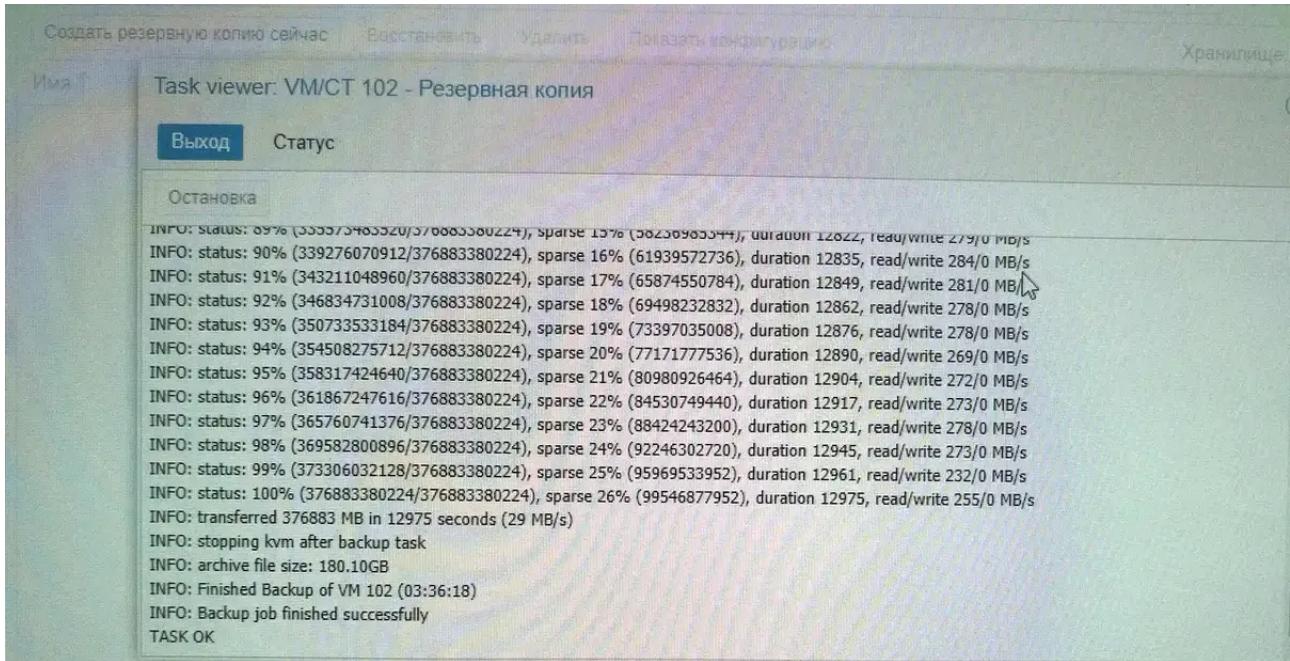
е) настройка правил *iptables* для подключений в *ProxmoxVE*.

```
# Generated by iptables-save v1.4.21 on Tue Dec 3 11:14:10 2019
*filter
:INPUT DROP [191:18834]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [106:32283]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A INPUT -s 192.168.0.0/16 -p tcp -m multiport --ports 53,443,8006 -j ACCEPT
-A INPUT -s 192.168.0.0/16 -p udp -m multiport --ports 53,443,8006 -j ACCEPT
-A INPUT -s 192.168.10.0/16 -p tcp -m multiport --ports 53,443,8006 -j ACCEPT
-A INPUT -s 192.168.10.0/16 -p udp -m multiport --ports 53,443,8006 -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A FORWARD -p tcp -m tcp --dport 22 -j ACCEPT
-A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -s 192.168.0.0/16 -p tcp -m multiport --ports 53,443,8006 -j ACCEPT
-A FORWARD -s 192.168.10.0/16 -p tcp -m multiport --ports 53,443,8006 -j ACCEPT
-A FORWARD -s 192.168.0.0/16 -p udp -m multiport --ports 53,443,8006 -j ACCEPT
-A FORWARD -s 192.168.10.0/16 -p udp -m multiport --ports 53,443,8006 -j ACCEPT
COMMIT
# Completed on Tue Dec 3 11:14:10 2019
```





3) создание резервных копий серверов RDS и SQL через встроенную функцию резервирования в веб-интерфейсе **Датацентра ProxmoxVE**.



и) создание шаброна сервера *pfSense* и перенос **вм** на рейд-массив основного сервера.
к) физическая (отвёртачная) сборка получившегося сервера, наведение порядка на рабочих местах.

Наконец-то Всё!!! Рад что всё работает; деньги на изучение курса «Безопасность Linux» честно заработаны, получены и потрачены от директора наших «Рогов и копыт».
Спасибо за внимание.