

Установка и настройка Proxmox VE 6.0

 interface31.ru/tech_it/2019/09/ustanovka-i-nastroyka-proxmox-ve-60.html

Записки IT специалиста

Технический блог специалистов ООО "Интерфейс"

- [Главная](#)
- Установка и настройка Proxmox VE 6.0

Proxmox Virtual Environment (Proxmox VE) - система виртуализации с открытым исходным кодом на базе Debian, использующая в качестве гипервизора KVM для виртуальных машин и LXC для контейнеров. Это позволяет запускать в виртуальной среде Linux-системы без потери производительности и остальные, поддерживаемые KVM ОС с минимальными потерями. Кроме того, Proxmox VE позволяет создавать высокодоступные конфигурации с несколькими серверами и распределенными системами хранения.



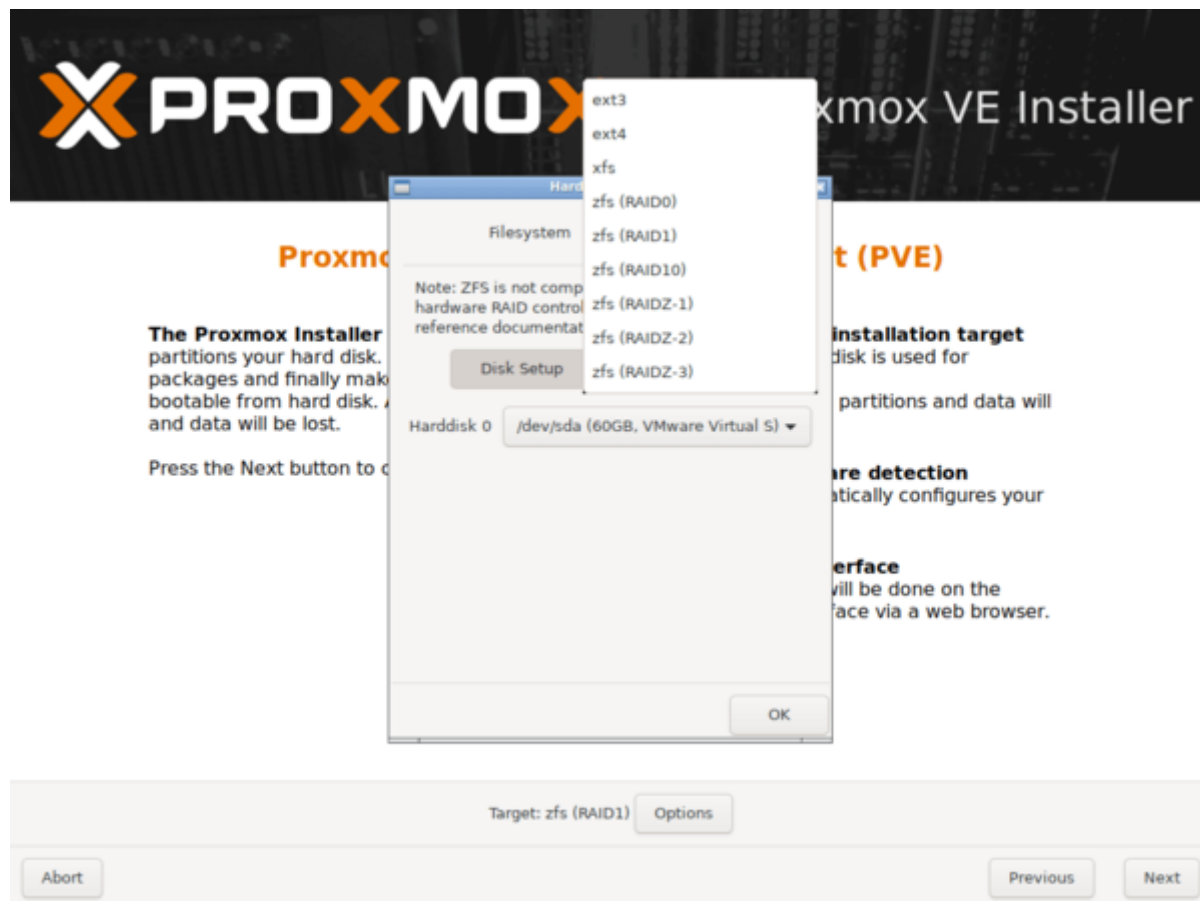
Онлайн-курс по устройству компьютерных сетей

На углубленном курсе "[Архитектура современных компьютерных сетей](#)" вы с нуля научитесь работать с Wireshark и «под микроскопом» изучите работу сетевых протоколов. На протяжении курса надо будет выполнить более пятидесяти лабораторных работ в Wireshark.

ССС

Получить Proxmox VE можно на [официальном сайте](#), сейчас доступна новая версия 6.0, которую мы и будем устанавливать. Да, это действительно open-source и это бесплатно, корпоративная подписка предоставляет доступ к закрытому репозиторию Proxmox VE Enterprise, который содержит стабильные обновления ПО и обновления безопасности, а также техническую помощь и поддержку. Никаких ограничений функциональности, если вы откажетесь от подписки, нет.

Для установки используется собственный инсталлятор и весь процесс предельно прост, от вас потребуется указать конфигурацию дисковой подсистемы:



А также пароль суперпользователя, настройки сети, страну и часовой пояс. Потом останется только откинуться в кресле и немного подождать, пока система установится на ваш компьютер. По завершении процесса установки вас встретит приглашение входа в систему, над которым будет указан адрес, который следует набрать в браузере для доступа к консоли управления.

Но не будем спешить. Прежде всего отключим корпоративный репозиторий, на который у нас все равно нет подписки, и подключим вместо него некоммерческий репозиторий Прокмох. Для отключения просто удалим файл репозитория из источников apt:

```
rm -f /etc/apt/sources.list.d/pve-enterprise.list
```

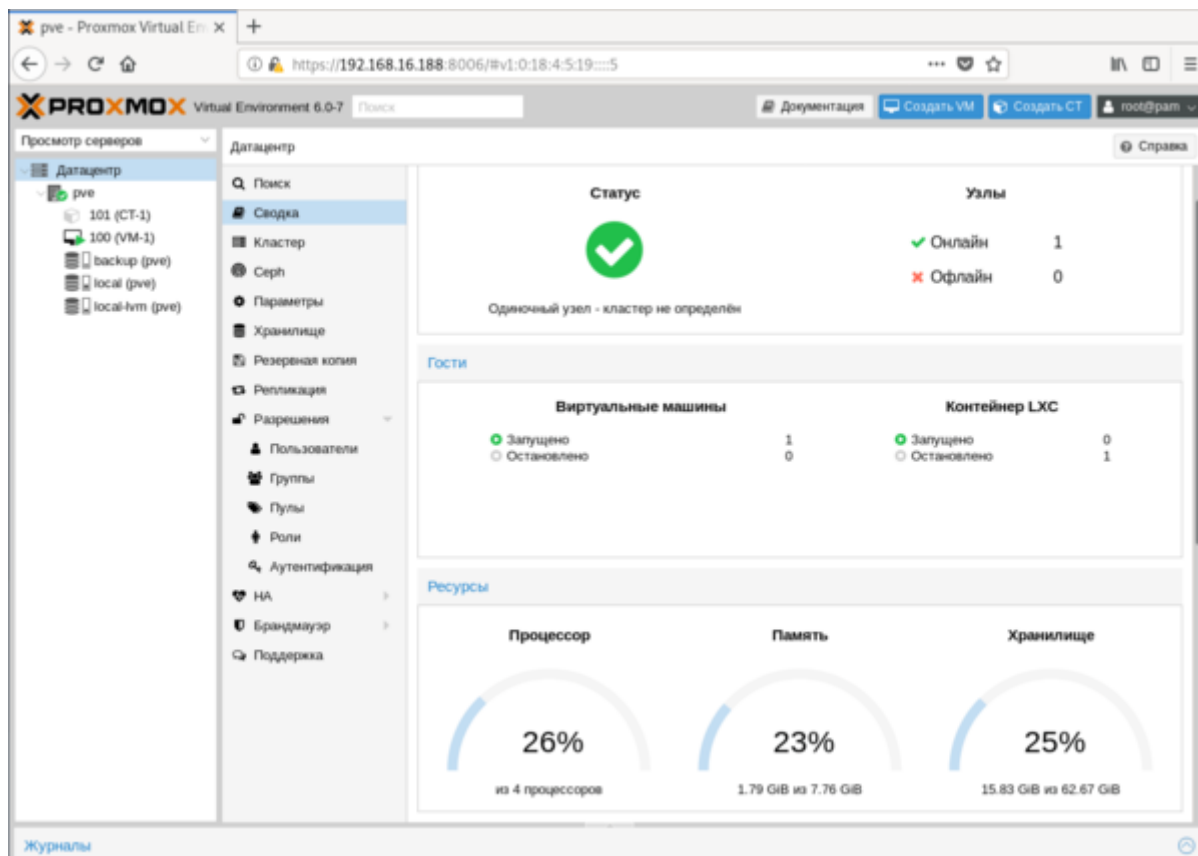
Затем создадим свой список:

```
touch /etc/apt/sources.list.d/pve-no-subscription.list
```

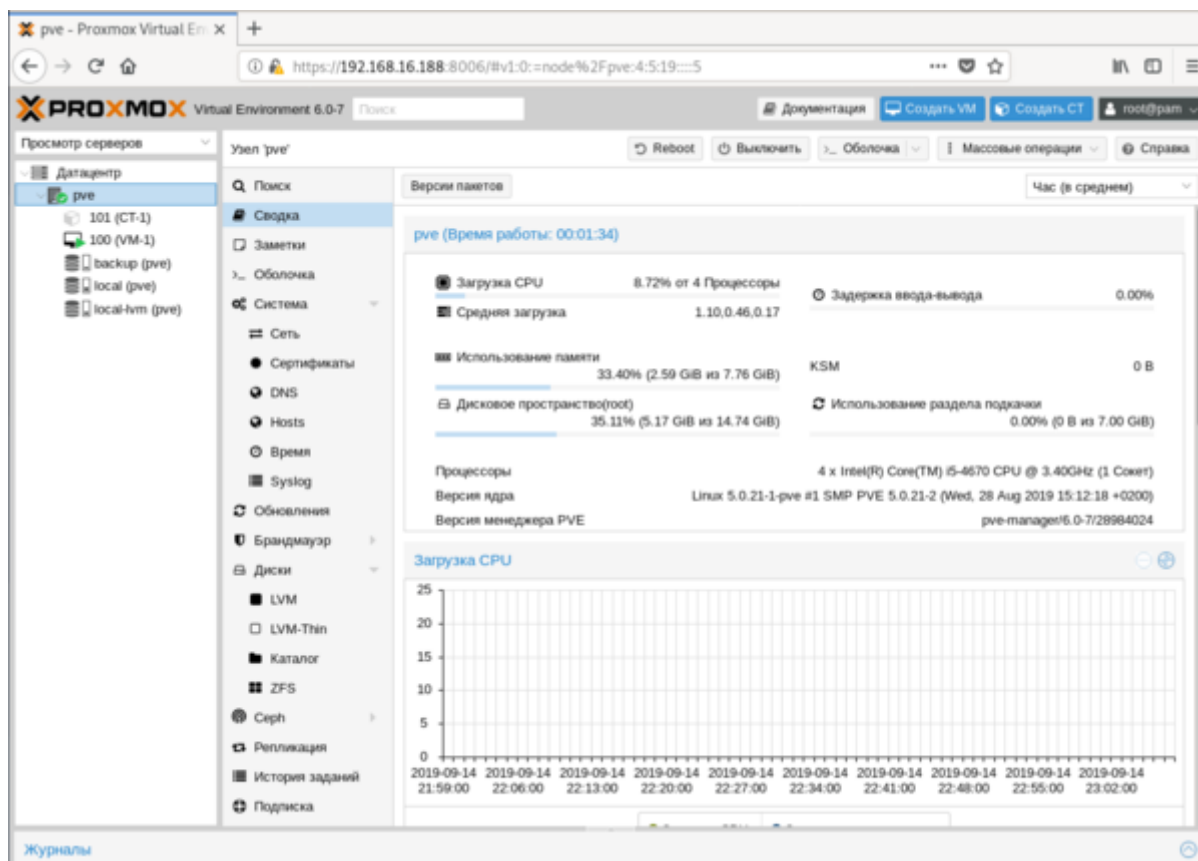
В который внесем следующие строки:

```
deb http://download.proxmox.com/debian/pve buster pve-no-subscription
```

Теперь можно переходить к консоли управления, откроем браузер и наберем предложенный адрес, консоль управления продуктами Прокмох работает на порту 8006, что следует помнить.

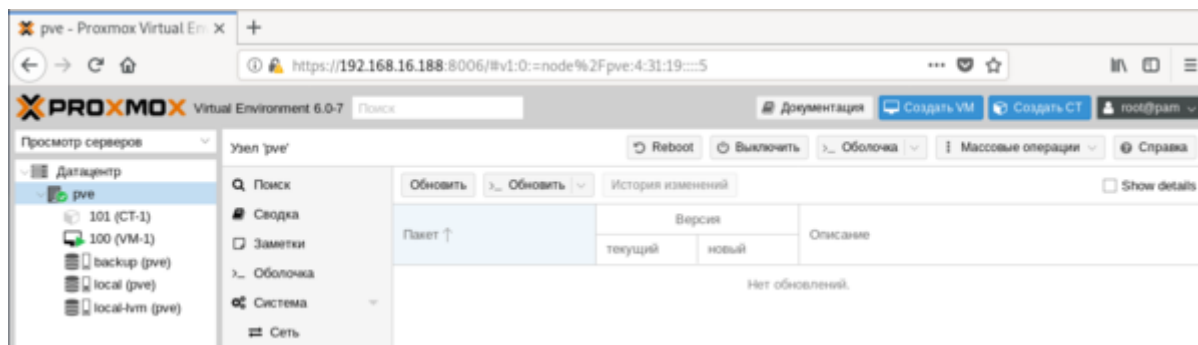


Структура Proxmox VE иерархична, высший уровень составляет **Датациентр**, здесь есть настройки, применяемые сразу ко всем узлам, например, управление пользователями или хранилищами, настройки резервного копирования и т.д.

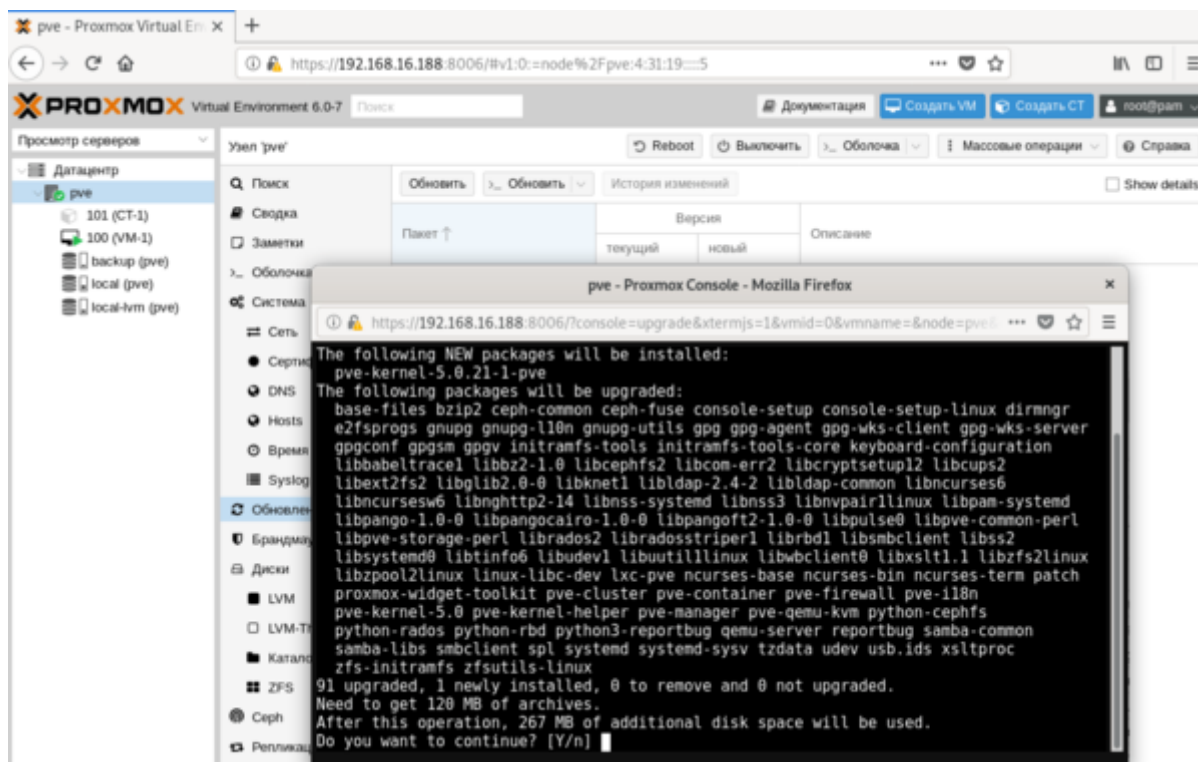


Следующий уровень - настройки узла, здесь доступна подробная статистика и большое количество настроек, которые относятся именно к этому серверу. Мы не будем останавливаться на них подробно, для администратора с достаточным уровнем подготовки разобраться с ними не составит труда.

Но есть и "сложности перевода", так в разделе **Обновления**, нас ждут две одинаковые кнопки **Обновить**.

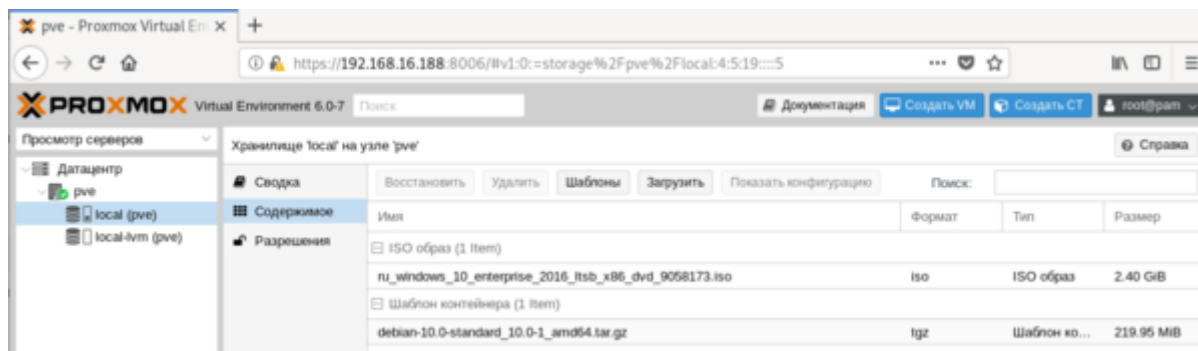


В английской версии это были **Update** и **Upgrade**, а вот здесь перевели как перевели. Первая из них запускает обновление списка пакетов, а вторая запускает само обновление, открывая в отдельном окне консоль сервера.



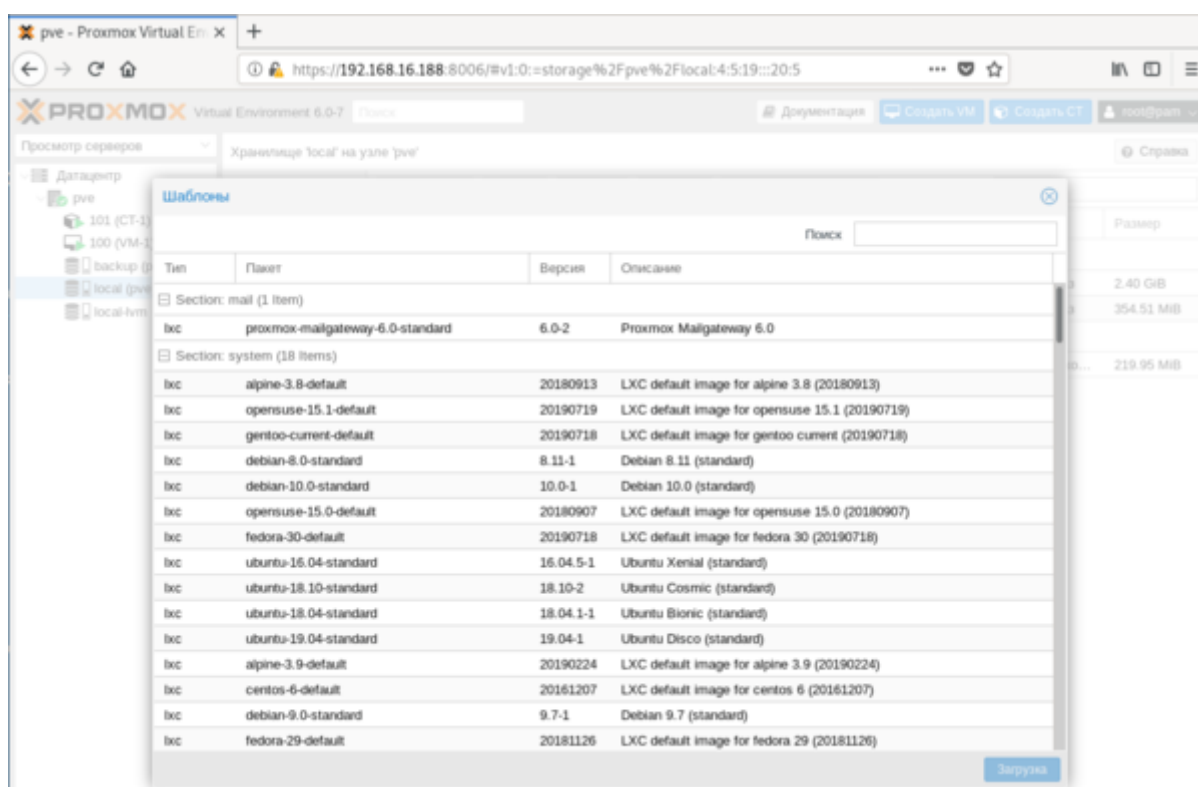
Если вам нужно просто попасть в консоль сервера, можно нажать кнопку **Оболочка** и перед вами в отдельном окне браузера откроется poVNC-сессия, при этом не требуется ни Flash, ни Java, что позволяет подключаться как к серверу, так и к виртуальным машинам с любого устройства, был бы браузер.

Перед тем, как создавать виртуальные машины и контейнеры, обратимся к **Хранилищам**, по умолчанию создается два хранилища: **local** и **local-lvm**.

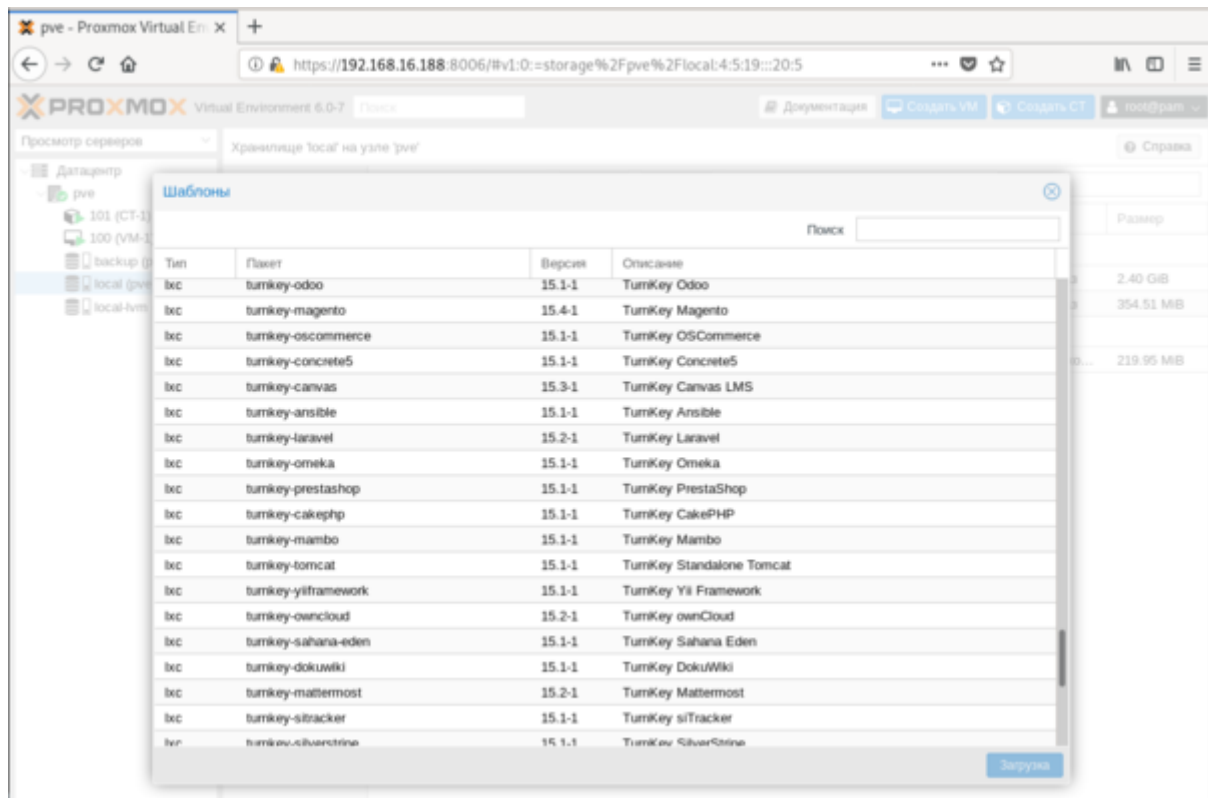


Первое предназначено для хранения ISO-образов и шаблонов контейнеров, а во втором хранятся сами виртуальные машины. Поэтому перед тем, как создать виртуальную машину или контейнер, нам нужно загрузить в хранилище нужный образ диска или шаблон. С образами все просто, нажимаем кнопку **Загрузить** и указываем нужный нам ISO-образ.

Для получения шаблонов следует нажать кнопку **Шаблоны** и мы попадем в официальное хранилище шаблонов, здесь мы можем сразу загрузить шаблон другого продукта Proxmox - Mail Gateway или шаблоны стандартных Linux-систем.



Прокрутив ниже, мы обнаружим шаблоны предварительно настроенных систем практически на все случаи жизни. Использовать их, либо настраивать все самостоятельно - решать вам, в ряде случаев такие шаблоны способны хорошо сэкономить время, особенно при развертывании большого количества однотипных систем.



Чтобы загрузить выбранный шаблон достаточно нажать кнопку **Загрузка** внизу окна.

Для создания виртуальной машины нажмем **Создать VM**. На первом экране укажем имя виртуальной машины:

Создать: Виртуальные машины (VM)

Общие | ОС | Система | Жесткий диск | Процессор | Память | Сеть | Подтверждение

Узел: Пул ресурсов:

VM ID:

Имя:

Справка ☐ Расширенный

Затем укажем тип и версию гостевой ОС и подключим нужный образ из хранилища.

Создать: Виртуальные машины (VM)

Общее **ОС** Система Жесткий диск Процессор Память Сеть Подтверждение

☒ Использовать файл с образом CD/DVD

Хранилище: local

ISO образ: ru_windows_10_enterprise_201

Гостевая ОС:

Тип: Microsoft Windows

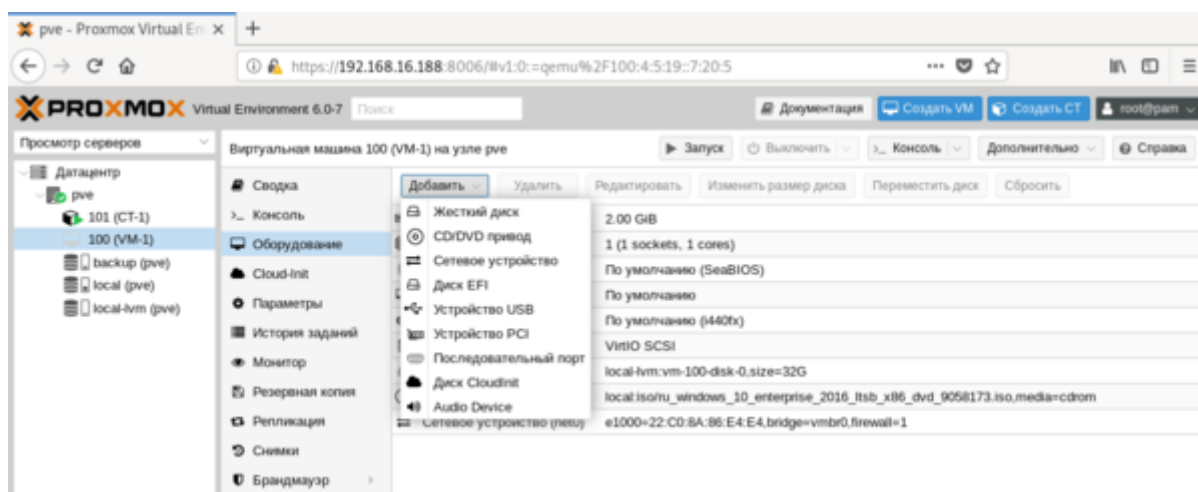
Версия: 10/2016/2019

☐ Использовать привод CD/DVD

☐ Нет носителя

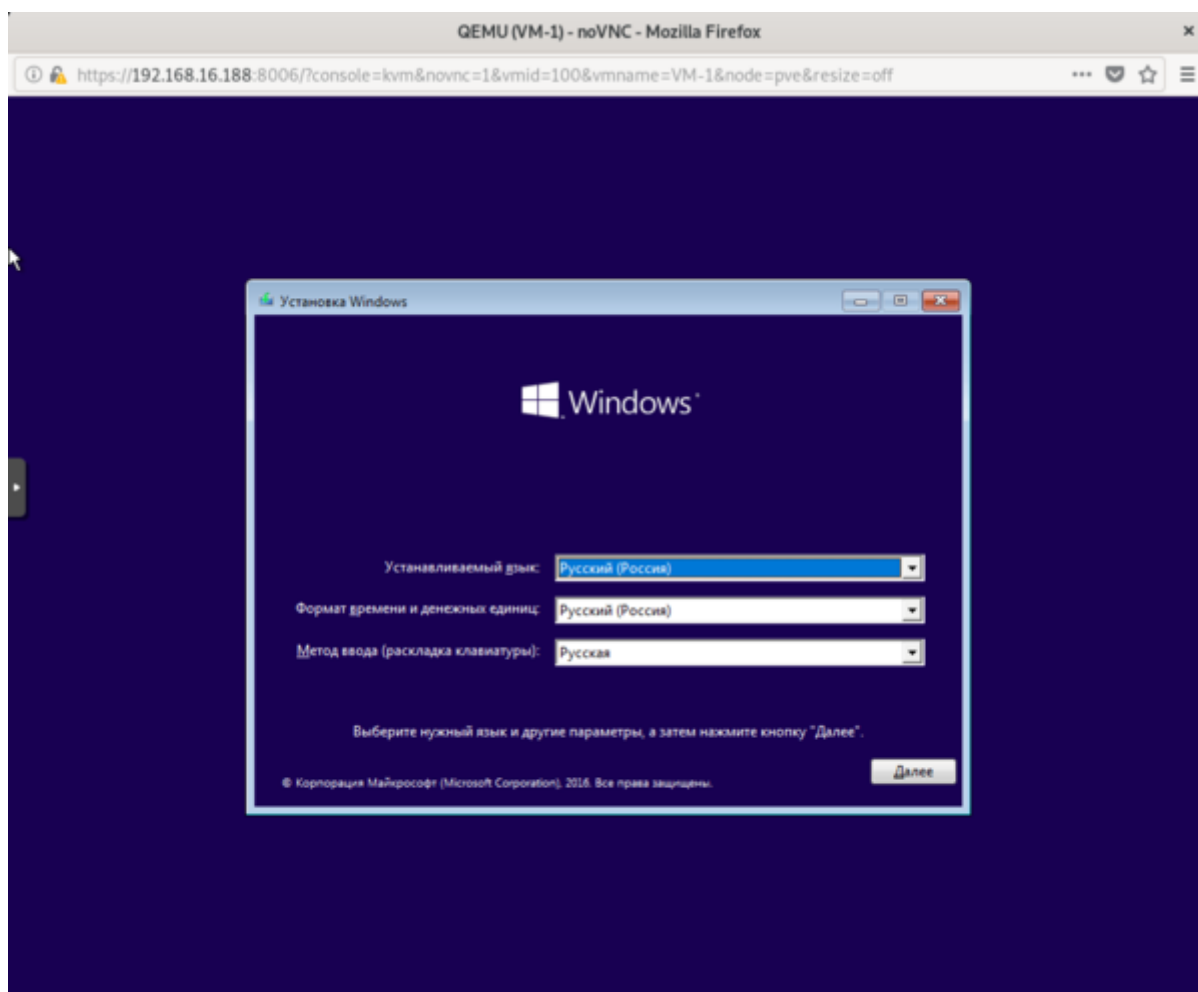
Расширенный ☐ Назад Далее

После чего, последовательно перемещаясь по пунктам следует настроить все остальные параметры виртуалки, особых сложностей это составить не должно. Но создав виртуальную машину не спешите ее запускать. Прежде всего перейдем в раздел **Оборудование**. Здесь можно не только настроить уже подключенное оборудование, но и добавить новое. В отличие от Hyper-V, Proxmox позволяет пробрасывать внутрь виртуальных машин USB-устройства, что может быть полезным, если вам нужно работать с ключами защиты или USB-токенами.



Также не забудьте заглянуть в **Параметры**, где стоит включить **Запуск при загрузке** и **Агент QEMU**. И если с первым пунктом все понятно, то Агент позволяет управлять системой со стороны гипервизора, без него вы, например, не сможете сделать резервную копию без остановки работы машины.

Итак, все настройки закончены, включаем виртуальную машину и переключаемся в консоль. Да, прямо в браузере вы увидите экран вашей виртуалки, при этом ничего не тормозит, работать с системой достаточно комфортно.



Пока идет установка системы создадим еще новый контейнер, для этого следует нажать кнопку **Создать СТ** в правом верхнем углу консоли управления. В качестве контейнеров могут быть только Linux-системы, на первом экране укажите имя контейнера и задайте пароль root или загрузите публичный SSH-ключ.

Создать: Контейнер LXC

Общее

Шаблон

Корневой диск

Процессор

Память

Сеть

DNS

Подтверждение

Узел:

pve

Пул ресурсов:

CT ID:

101

Пароль:

••••••••

Имя хоста:

CT-1

Подтвердить пароль:

••••••••

Непривилегированный контейнер:

☒

Открытый ключ SSH:

Загрузить файл ключа SSH

Справка

Расширенный ☐

Назад

Далее

Затем укажите шаблон и следующими шагами сконфигурируйте виртуальное железо.

Создать: Контейнер LXC

Общее

Шаблон

Корневой диск

Процессор

Память

Сеть

DNS

Подтверждение

Хранилище:

local

Шаблон:

debian-10.0-standard_10.0-1_am

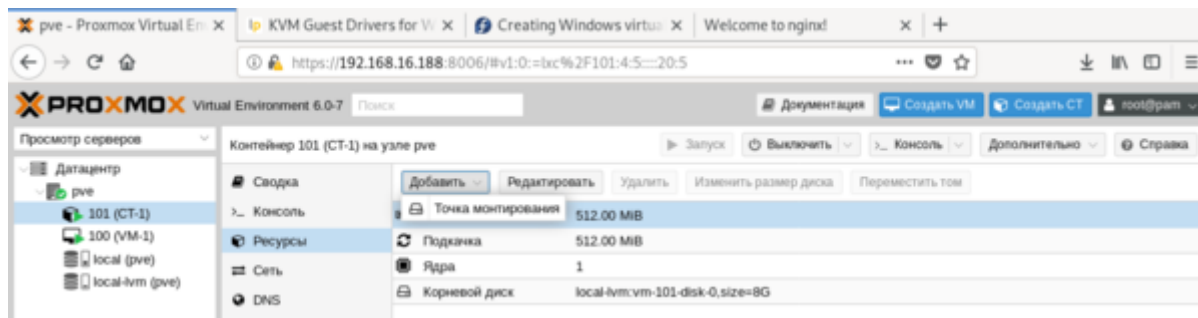
Справка

Расширенный ☐

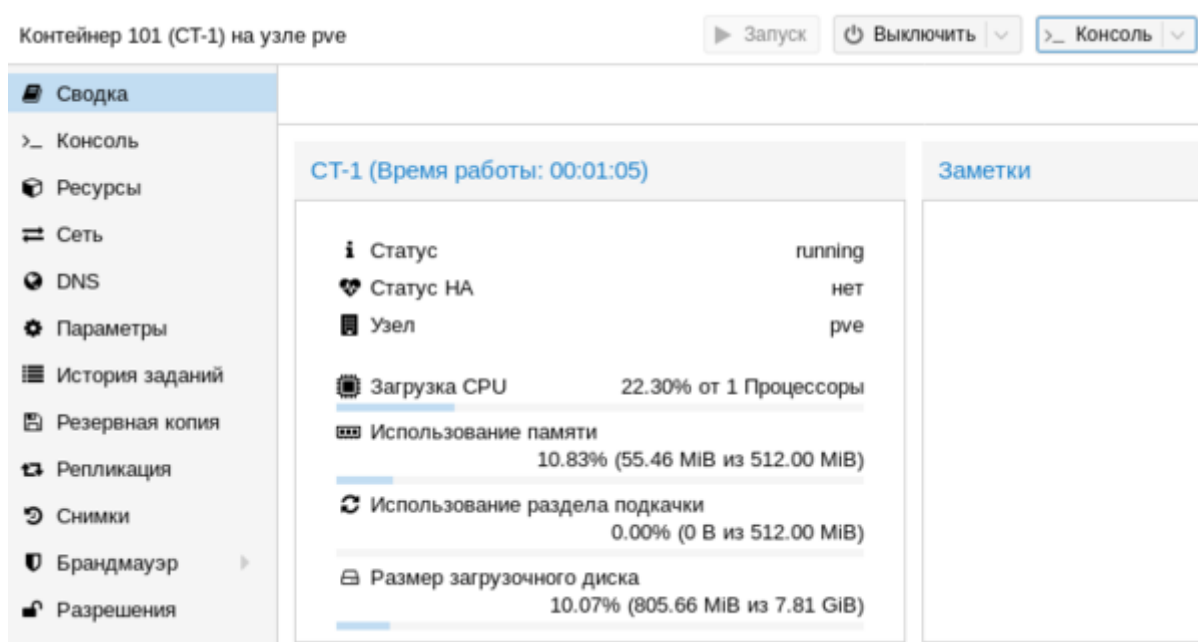
Назад

Далее

Доступных настроек у контейнеров гораздо меньше, зато они гораздо более экономно расходуют ресурсы.



Обратите внимание, что запущенный контейнер с Debian 10 потребляет всего лишь 55,46 МБ оперативной памяти.

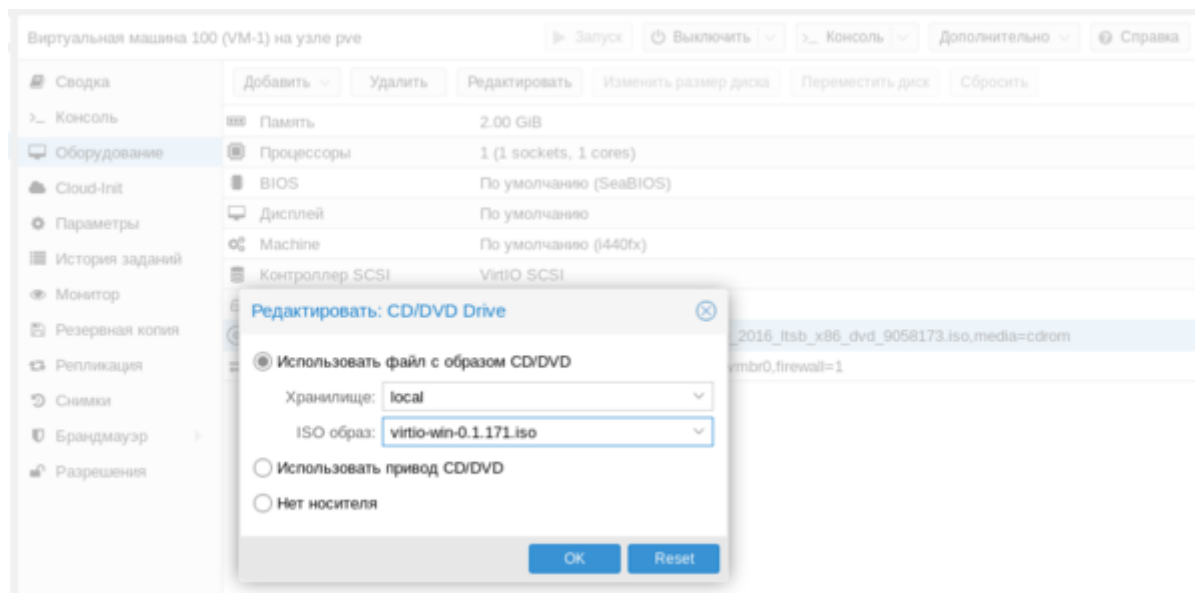


Тем временем, пока мы создавали контейнер, должна была установиться наша виртуальная машина. После установки ВМ потребуется установить в нее QEMU-агента для полноценного взаимодействия с гипервизором. Для Linux систем все просто, на **deb-based** системах достаточно выполнить:

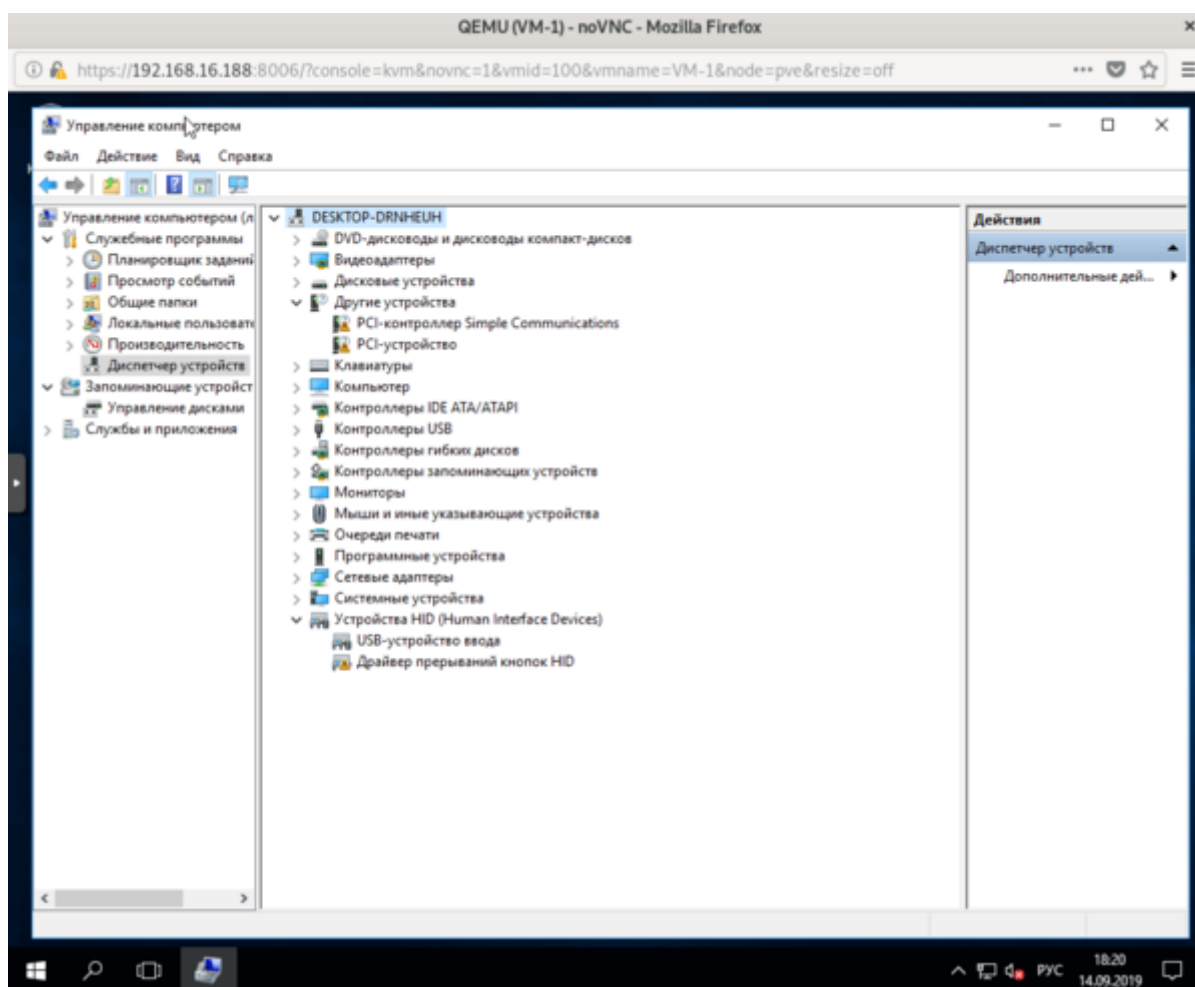
```
apt install qemu-guest-agent
```

Для других дистрибутивов следует установить пакет **qemu-guest-agent** штатным пакетным менеджером.

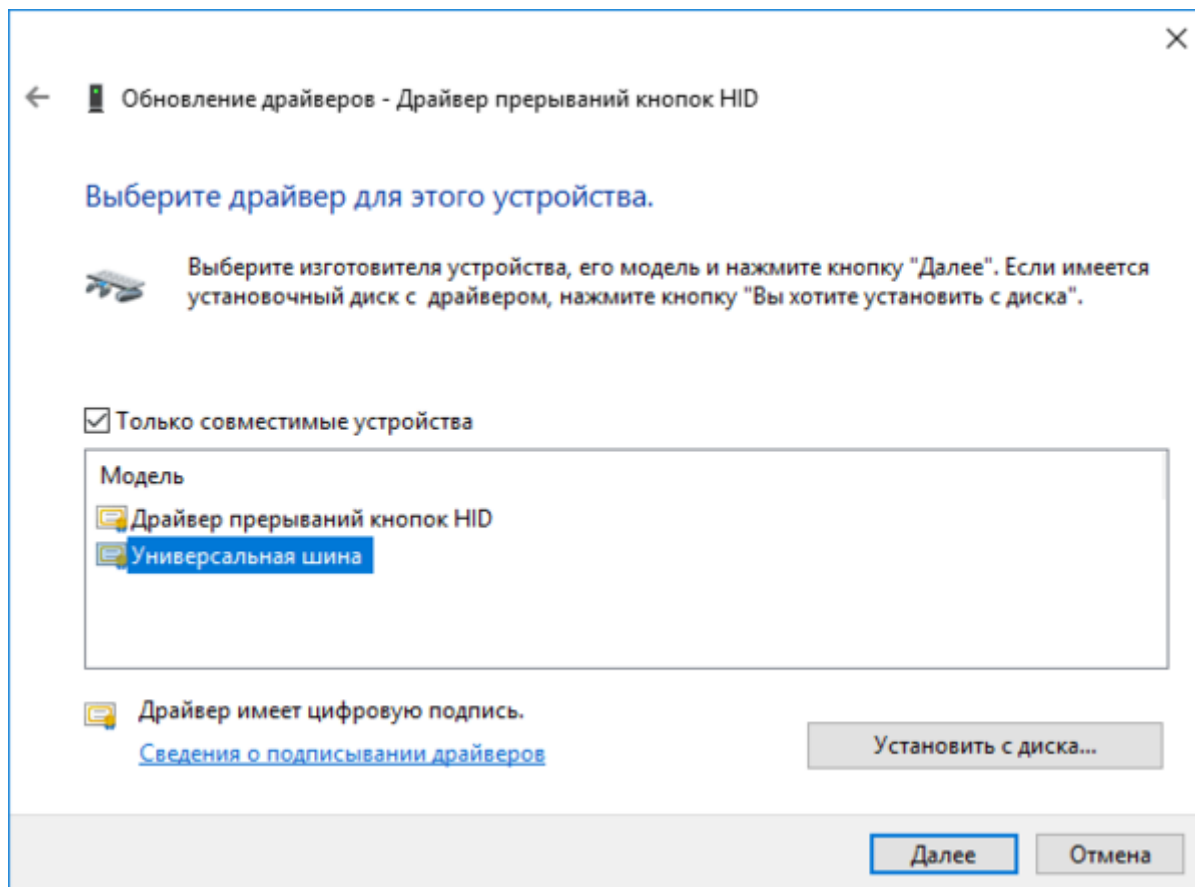
С Windows все несколько сложнее, прежде всего скачаем и поместим в хранилище Proxmox ISO-образ с virtIO драйверами. Получить его можно со страницы [Fedora Project](https://www.fedoraproject.org/iso/2019/01/20190120-100x64/). Затем подключим скачанный образ к виртуальной машине.



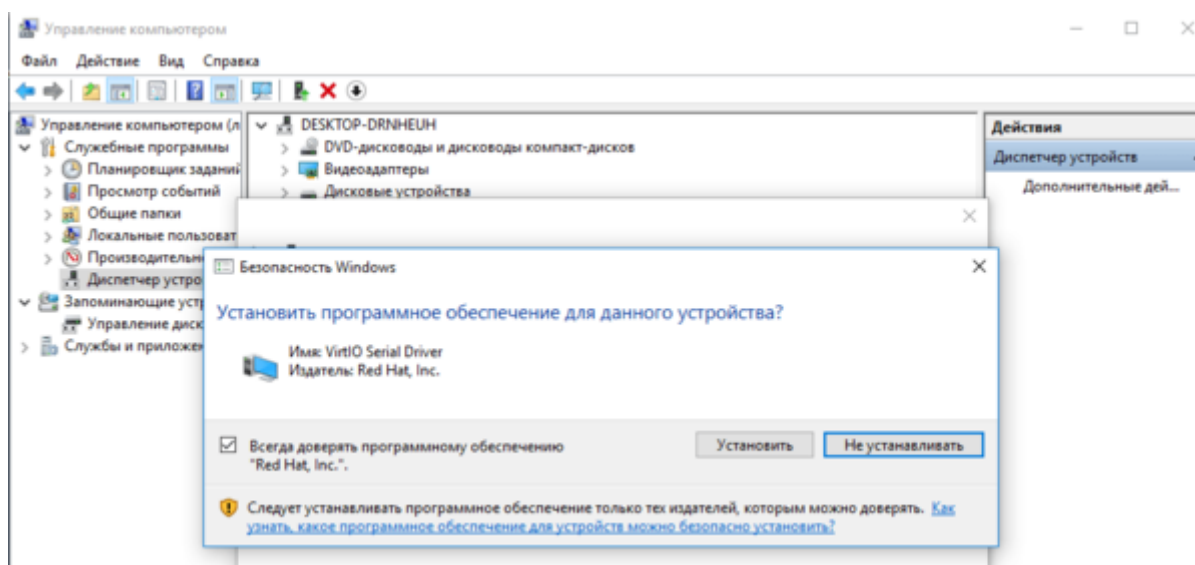
При включенном QEMU-агенте мы увидим три неустановленных устройства.



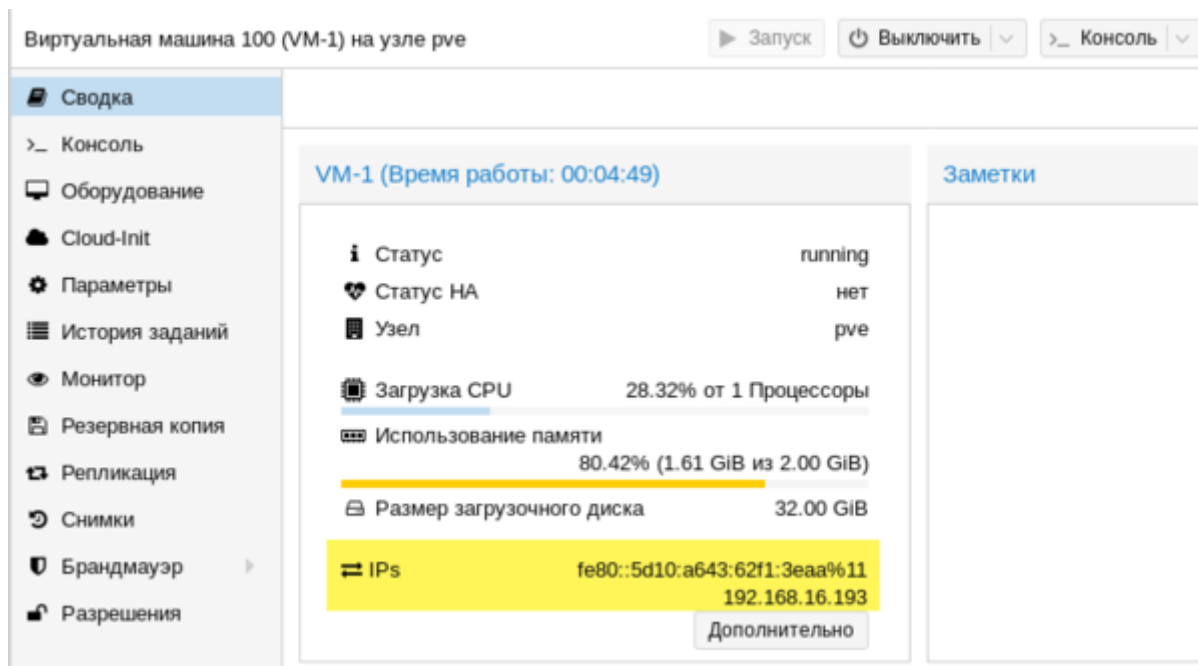
Начнем с последнего, для **Драйвера прерываний кнопок HID** перейдем в **выбор из списка уже установленных драйверов** и укажем драйвер **Универсальной шины**.



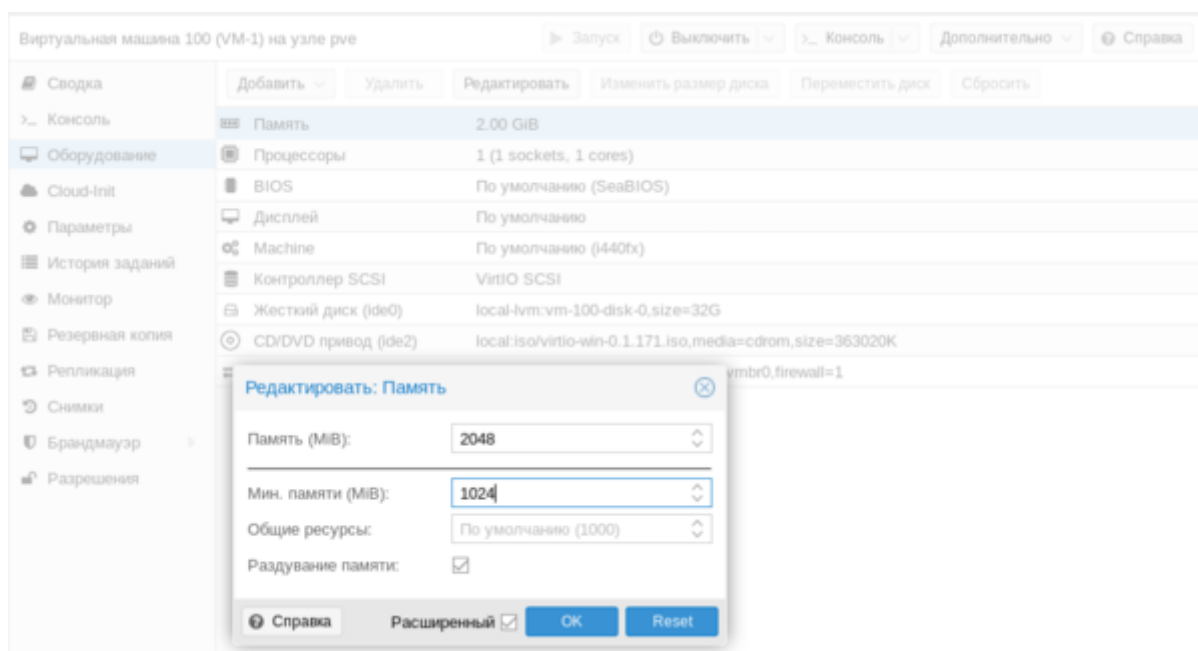
Для двух оставшихся выполним поиск драйверов на смонтированном диске и устанавливаем драйвер **VirtIO Serial Driver**, который отвечает за работу QEMU-агента:



И **VirtIO Balloon Driver**, который отвечает за динамическое управление памятью. При работающем QEMU-агенте внизу окошка со сведениями о виртуальной машине вы увидите ее IP-адрес, в противном случае там будет сообщение, что QEMU-агент не установлен.

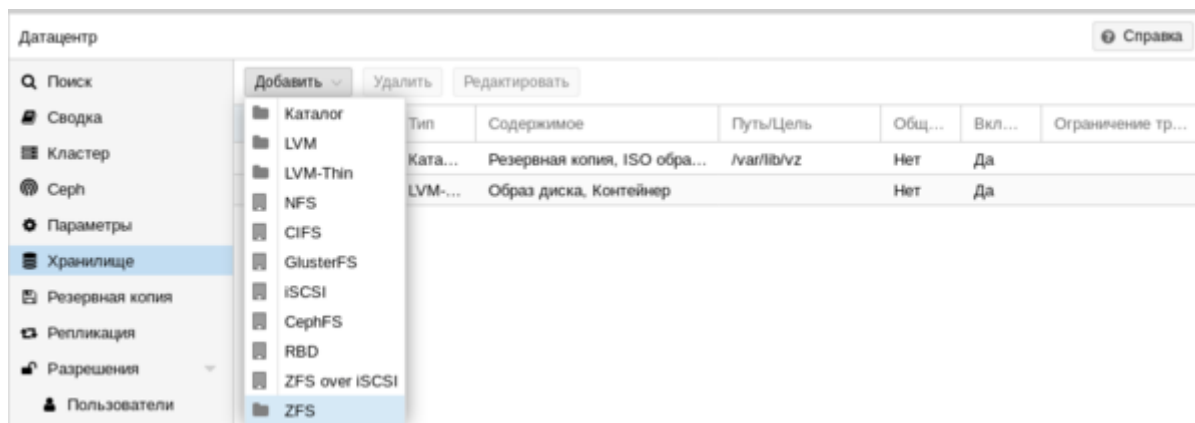


Также вы можете теперь включить для виртуальной машины динамическую память, для этого установите флаг **Раздувание памяти** и укажите минимальный размер оперативки, с которой начнет работу виртуалка:

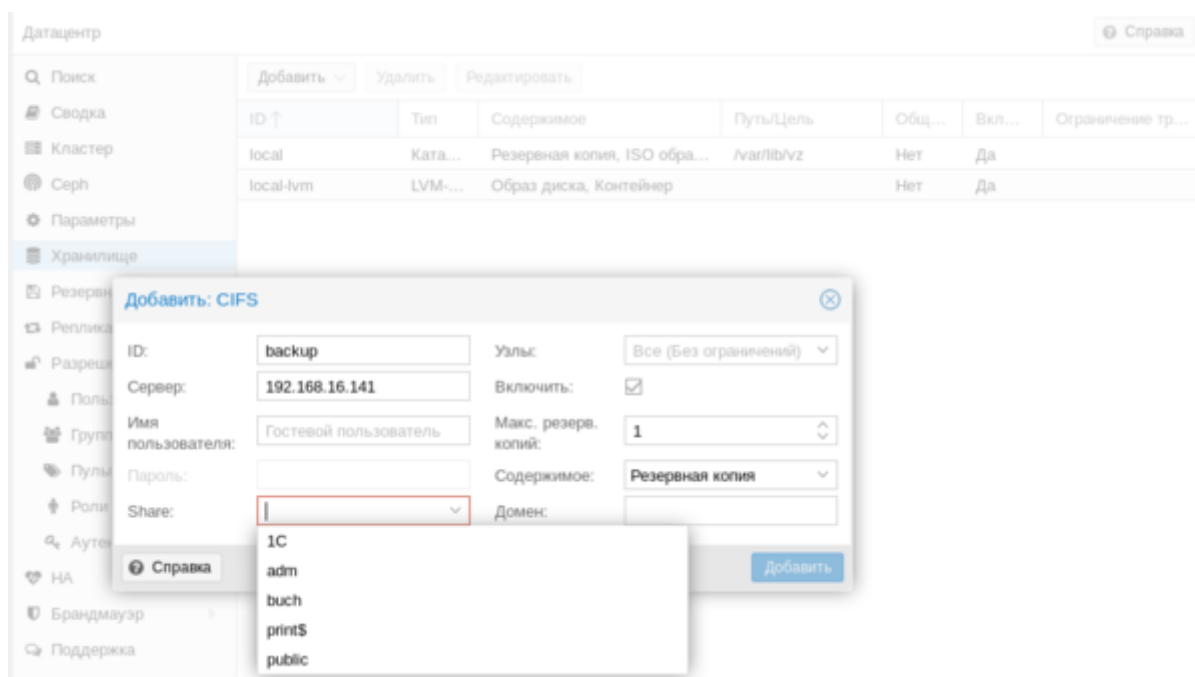


Но придерживайтесь при этом разумных пределов, так вы можете столкнуться с невозможностью запустить современные версии Windows если укажете минимальный размер памяти менее 1 ГБ.

Еще одна важная функция, которая доступна в Proxmox VE - это резервное копирование. Существует хорошая практика - хранить резервные копии за пределами хоста. Поэтому подключим к Proxmox сетевое хранилище, для этого перейдем в **Датацентр - Хранилище**. Выбор здесь достаточно богатый: iSCSI, NFS, CIFS и прочее.

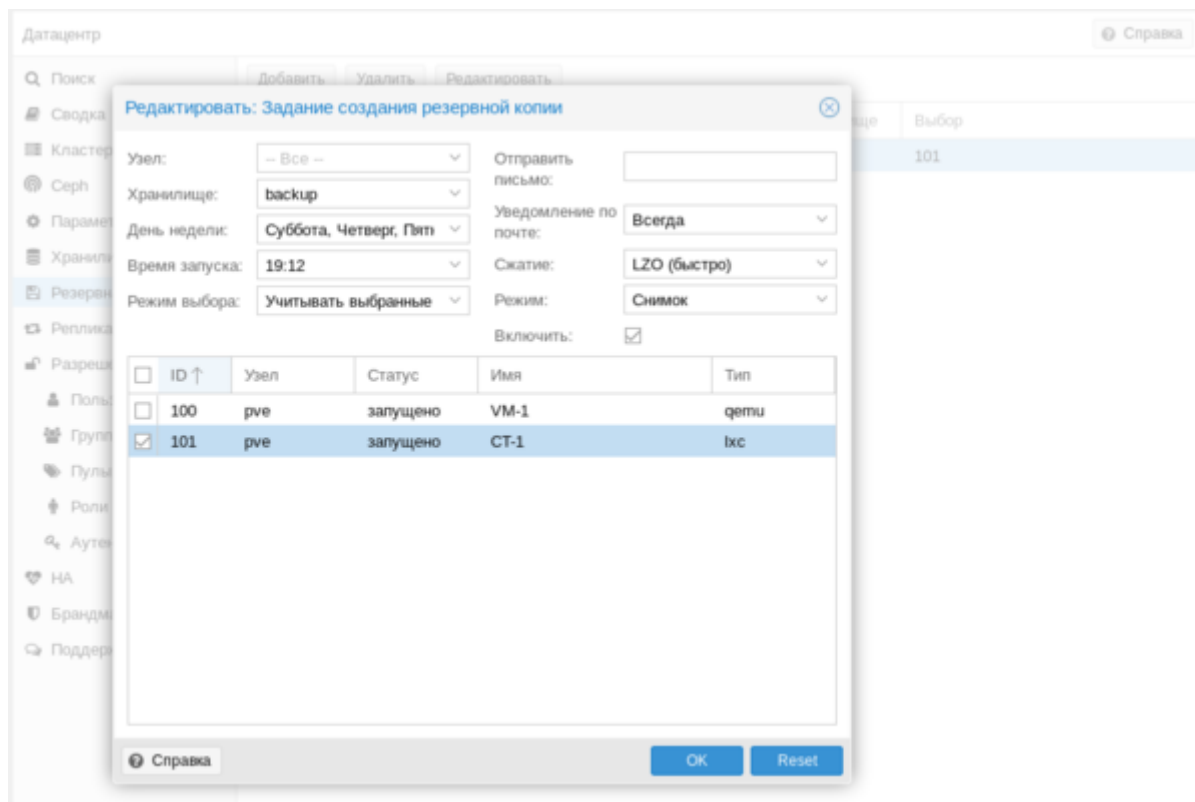


В нашем случае мы подключим CIFS (SMB) хранилище.



Укажите имя хранилища, в нашем случае **backup**, его IP-адрес или FQDN-имя, доступные ресурсы будут просканированы автоматически и вам останется только выбрать их из списка, в поле **Содержимое** укажите **Резервная копия**, а также не забудьте указать максимальное количество резервных копий в хранилище.

Затем перейдем в раздел **Резервная копия** и создадим новый сценарий резервного копирования:



Особых сложностей там нет, выбираете нужные виртуальные машины или контейнеры и задаете расписание копирования. После чего вам только останется проверить, что резервные копии действительно создаются.

Как видим, Proxmox представляет собой вполне серьезный и самодостаточный продукт, способный стать основой вашей системы виртуализации, предоставляя возможности на уровне дорогих коммерческих гипервизоров.

- **Категории:**
- Виртуализация,
- Системному администратору.
- **Теги:**
- Linux,
- Proxmox,
- Виртуализация