Настраиваем сеть в Proxmox Virtual Environment

Записки IT специалиста

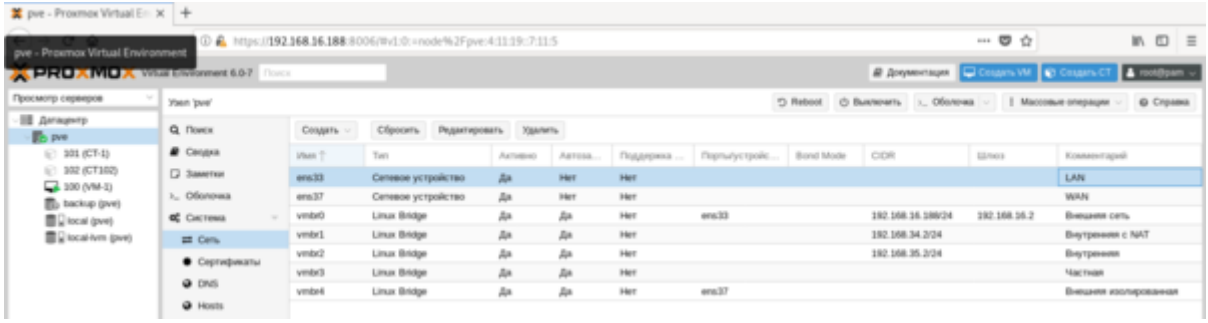
Технический блог специалистов ООО"Интерфейс"

- Главная
- Настраиваем сеть в Proxmox Virtual Environment

Настройка сетевой конфигурации системы виртуализации - одна из самых главных задач, она же вызывает наибольшие затруднения у начинающих. Поэтому начиная цикл статей о Proxmox мы сразу решили подробно разобрать этот вопрос. Тем более, что официальная документация довольно скупо освещает эту тему и может сложиться впечатление, что Proxmox ограничен в сетевых возможностях по сравнению с другими гипервизорами. Однако это не так, скорее даже наоборот, потому что перед нами открытое ПО и мы можем конфигурировать его именно так, как считаем нужным, даже если этих возможностей не было из коробки.

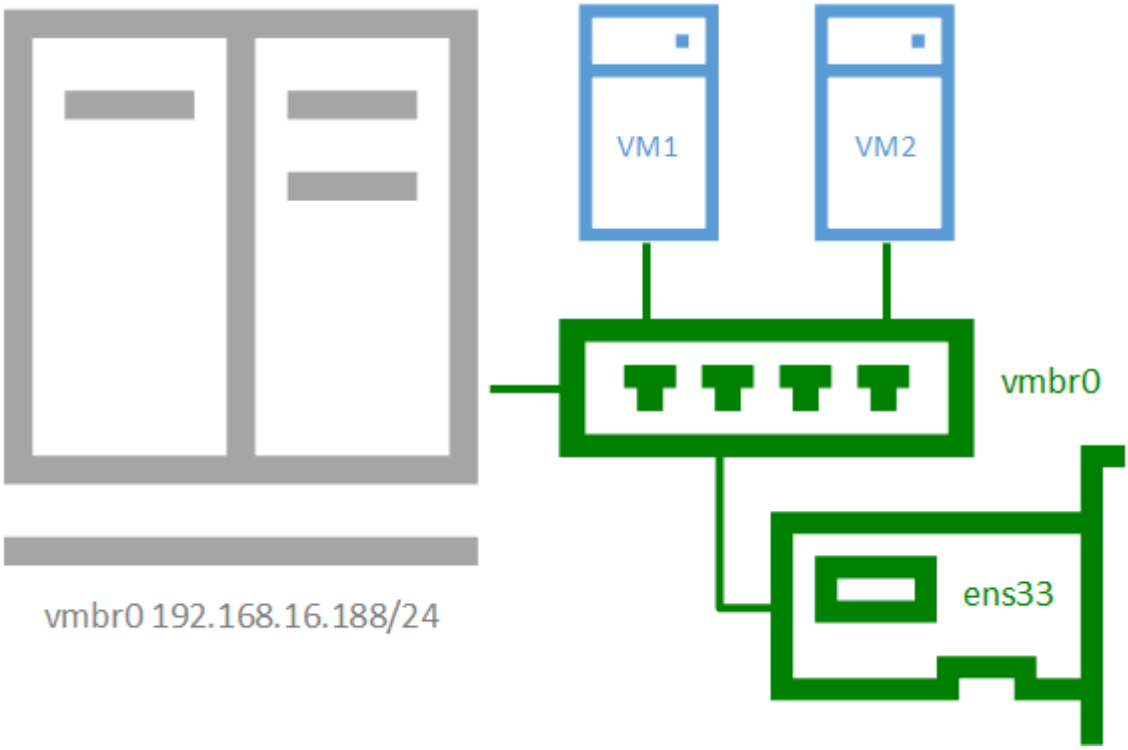
Если обратиться к официальной документации, то там будет рассказано о двух основных сетевых конфигурациях: с использованием моста и маршрутизации. Приведенные примеры покрывают основные сценарии использования и не углубляются в подробности, но различные комбинации настроек для этих вариантов позволяют реализовывать самые разнообразные сетевые конфигурации. В данном материале мы рассмотрим базовые возможности Proxmox, не касаясь объединения сетевых адаптеров или использования Open vSwitch, потому как это отдельные темы, лежащие за рамками базовой настройки.

Все сетевые параметры настраиваются на уровне ноды, для этого перейдите на нужный сервер и раскройте **Система - Сеть**. Ниже показан пример нашего тестового сервера, где реализованы все те сетевые конфигурации, о которых мы будем говорить ниже.



Внешняя сеть

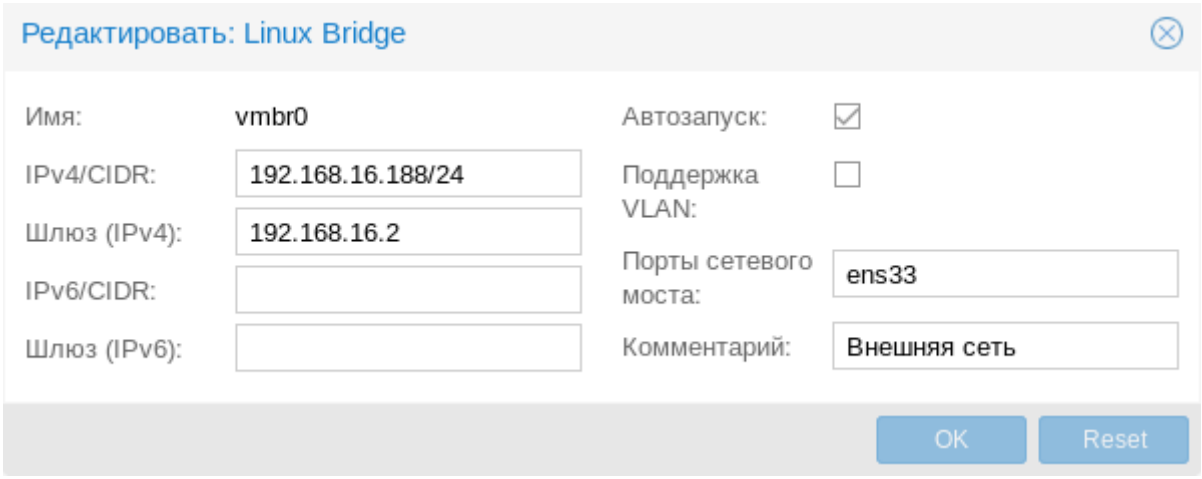
Сетевая конфигурация, создаваемая по умолчанию, когда и виртуальные машины, и гипервизор получают прозрачный доступ к вешней сети, подключенной через физический сетевой адаптер. Она же самая часто используемая, так как позволяет организовать простой доступ к виртуальным машинам, как к самым обычным узлам локальной сети.



В основе всех виртуальных сетей в Proxmox лежит **сетевой мост** (*Linux Bridge*) - **vmbr**, допускается создание до 4095 таких устройств. Сетевой мост может включать в себя как физические, так и виртуальные адаптеры, выполняя для них роль неуправляемого коммутатора. Физическая сетевая карта, подключенная к мосту, не имеет настроек и используется как физический Ehternet-интерфейс для данного виртуального коммутатора. Все сетевые настройки производятся внутри виртуальных машин, которые через мост и физический адаптер прозрачно попадают во внешнюю сеть.

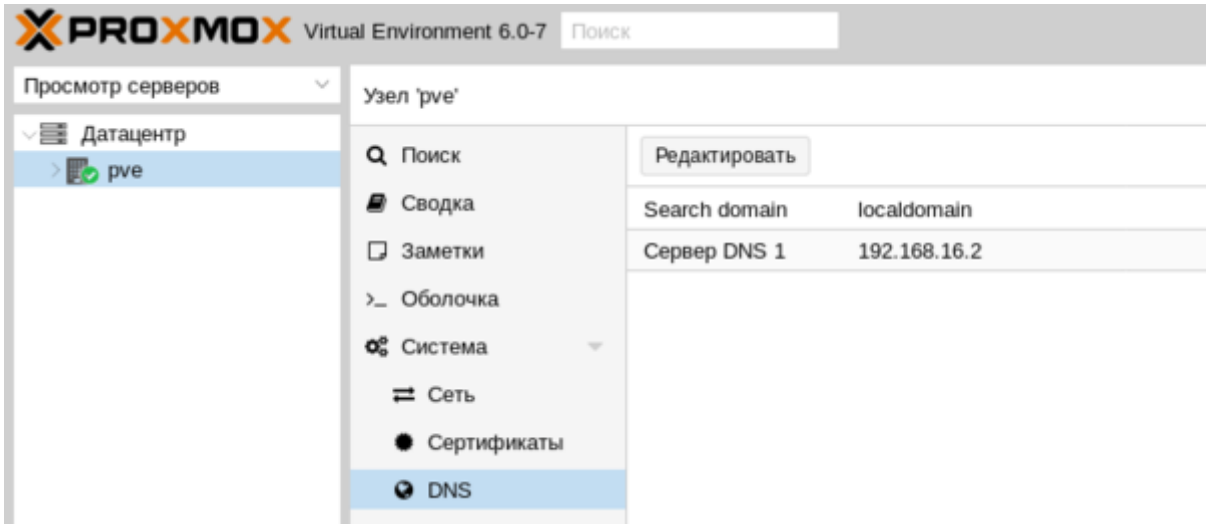
Присвоение интерфейсу моста IP-адреса фактически подключает к виртуальному коммутатору сам хост, т.е. гипервизор, который также прозрачно попадет во внешнюю сеть. Если в Hyper-V для подключения гипервизора к сети на хосте

создавался еще один виртуальный сетевой адаптер, то в Proxmox для этого следует назначить IP-адрес интерфейсу моста. Ниже показан пример такой настройки:

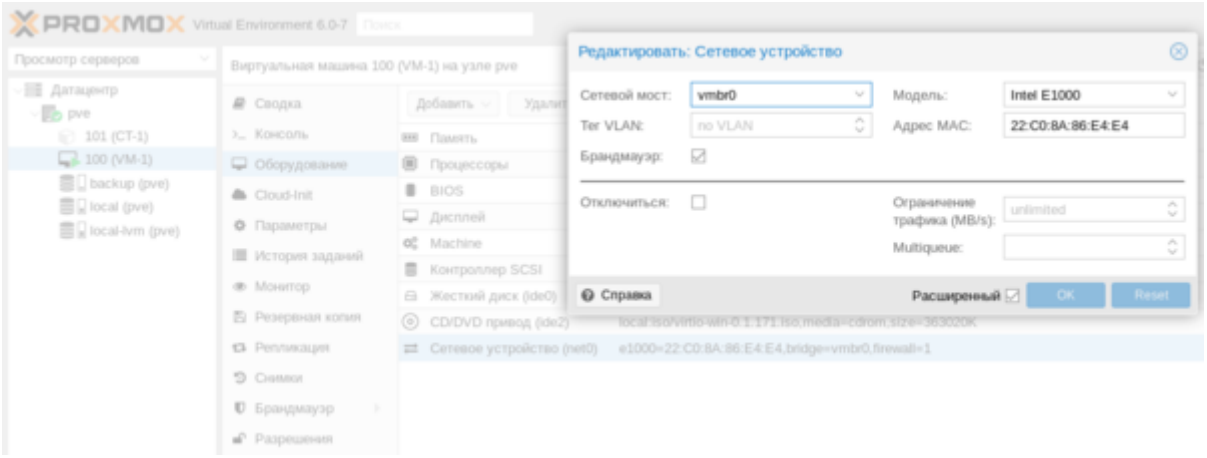


В настройках указывается адрес и шлюз, опция автозапуска и привязанный к мосту физический адаптер. Также мы советуем в поле комментарий оставлять осмысленное описание сетевого устройства, чтобы всегда было понятно, что это и зачем.

Фактически это сетевые настройки самого гипервизора. Обратите внимание, что сервера DNS указываются отдельно, в **Система - DNS**:



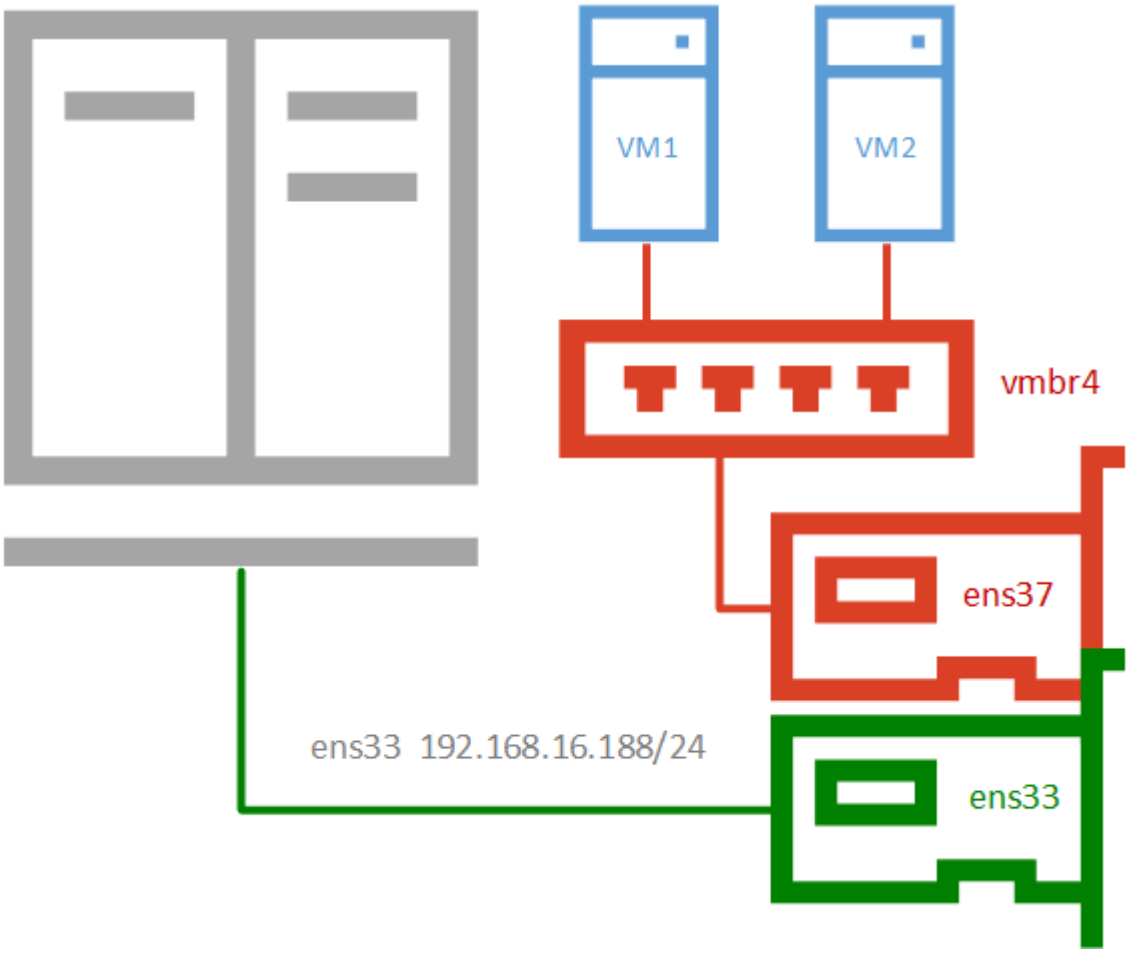
Для того, чтобы подключить к такой сети виртуальную машину в настройках ее сетевого адаптера следует выбрать нужный мост (виртуальный коммутатор):



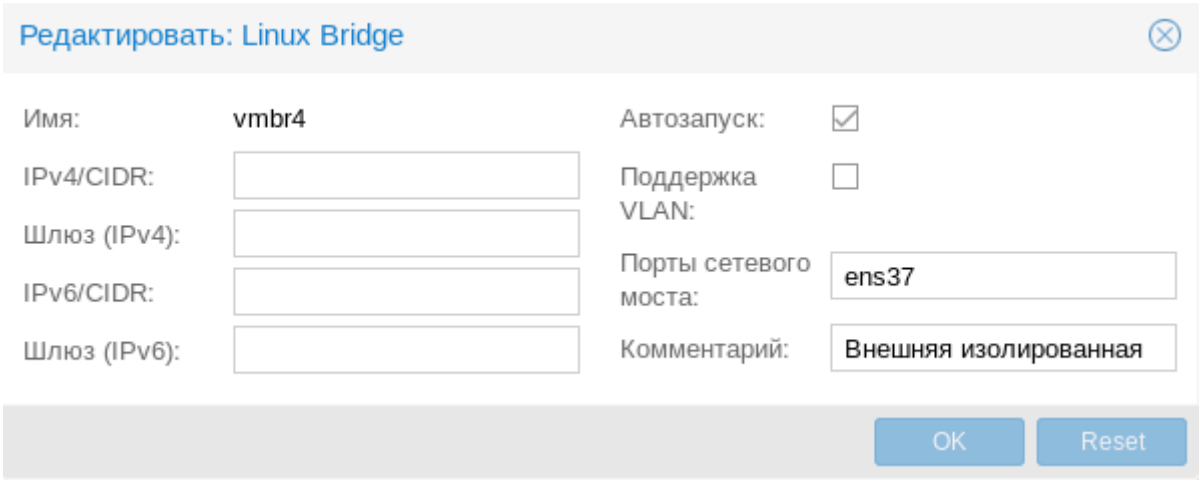
Сетевые настройки виртуальной машины либо задаются вручную, либо могут быть получены от DHCP-сервера внешней сети.

Внешняя изолированная сеть

Данная конфигурация требует минимум двух сетевых адаптеров и предусматривает изоляцию гипервизора от внешней сети и виртуальных машин. Это может быть полезно при виртуализации пограничных устройства, например, шлюза. Либо когда виртуальные машины арендуются третьими лицами, либо находятся вне доверенной сети и доступ к гипервизору оттуда должен быть закрыт.

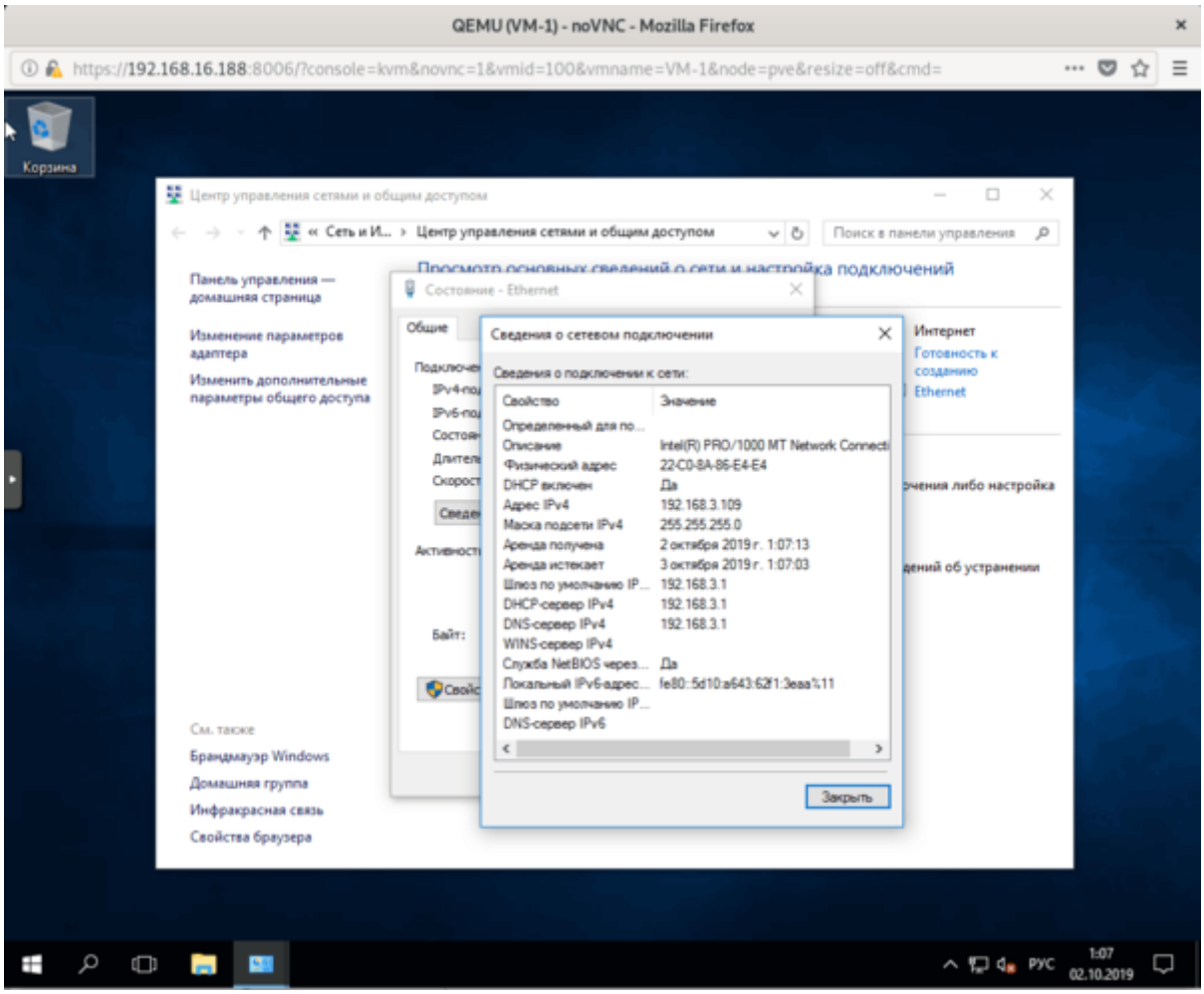


Для создания изолированной внешней сети нам потребуется создать новый сетевой мост без сетевых настроек и привязать к нему физический адаптер (тоже без настроек), таким образом будет обеспечен доступ виртуальных машин во внешнюю сеть с изоляцией этой сети от гипервизора.



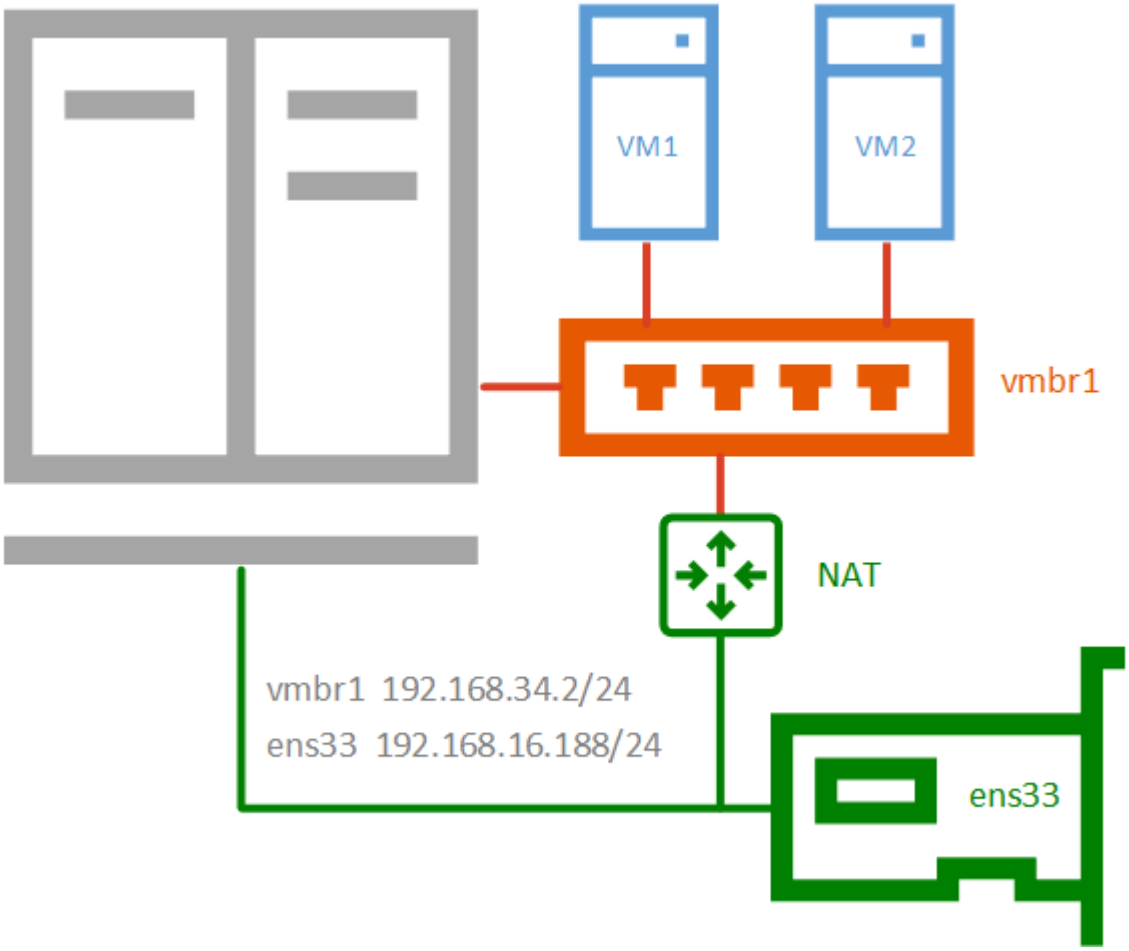
Для доступа к самому гипервизору может быть использован либо другой сетевой адаптер (как показано на нашей схеме), так и созданная по умолчанию внешняя сеть с сетевым мостом. Оба варианта имеют право на жизнь, а во втором случае вы сможете также подключать виртуальные машины к разным виртуальным сетям. Поэтому не следует рассматривать приведенную нами схему как догму, это только один из возможных вариантов и выбран нами в целях упрощения схемы.

Для примера мы подключили к такой сети виртуальную машину, которая тут же получила по DHCP адрес из внешней сети, никак не связанной с гипервизором.

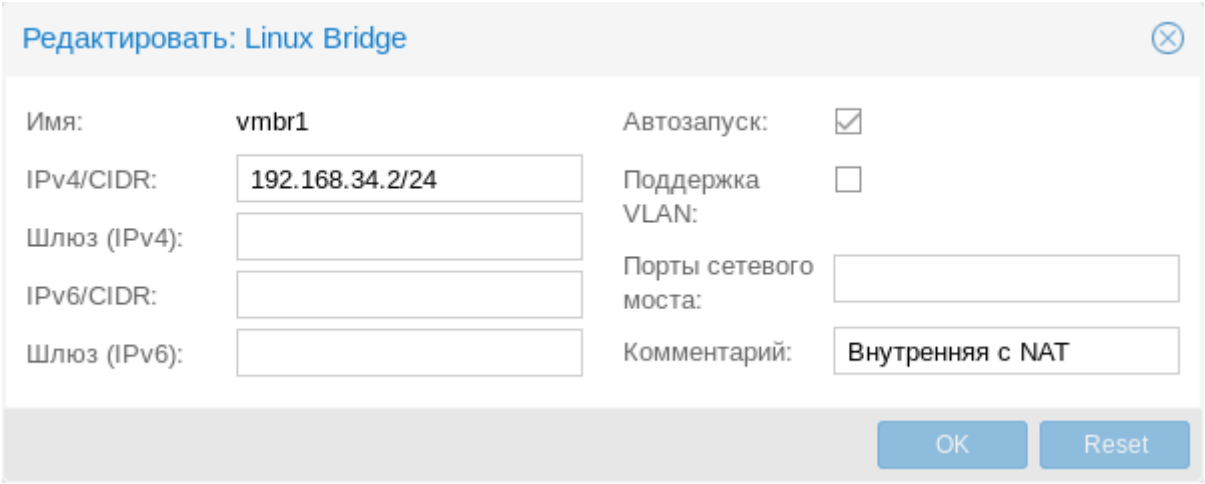


Внутренняя сеть с NAT

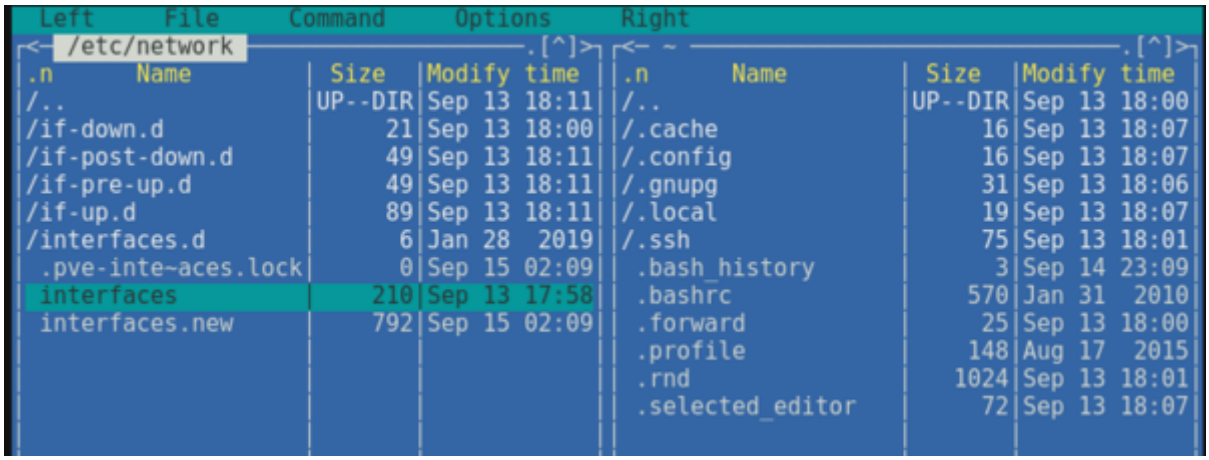
Применяется в тех случаях, когда нужно изолировать виртуальные машины в собственной сети, но в тоже время обеспечить им доступ в интернет, а также доступ из внешней сети к некоторым из них (или отдельным сетевым службам). Широко используется в лабораторных сценариях, а также при работе с контейнерами.



Обратите внимание, данная конфигурация не может быть изолирована от хоста, так как именно хост предоставляет ей службу **трансляции сетевых адресов** (NAT) и выступает шлюзом для виртуальных машин. Для настройки такой сети создайте новый сетевой мост без привязки к физическому адаптеру и назначьте ему IP-адрес из произвольной сети, отличной от внешней.



Все изменения сетевой конфигурации требуют перезагрузки узла гипервизора, поэтому, чтобы не перезагружать узел дважды перейдем в консоль сервера и перейдем в директорию **/etc/network**, в котором будут присутствовать файлы **interfaces** - с текущей сетевой конфигурацией и **interfaces.new** - с новой, которая вступит в силу после перезагрузки.



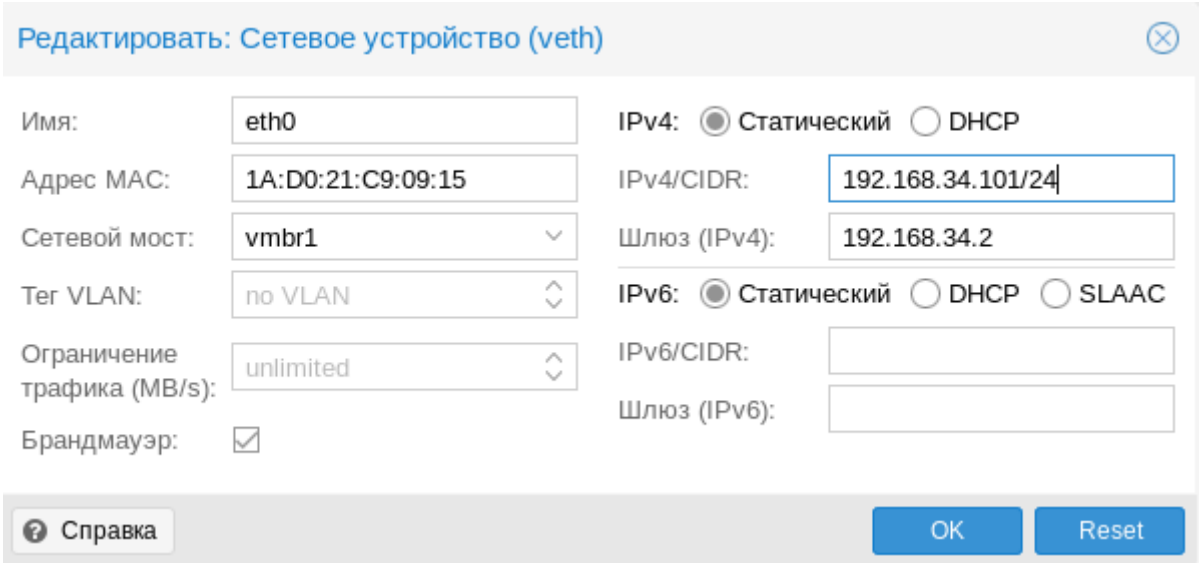
Откроем именно **interfaces.new** и внесем в конец следующие строки:

```
post-up echo 1 > /proc/sys/net/ipv4/ip_forward
post-up   iptables -t nat -A POSTROUTING -s '192.168.34.0/24' -o ens33 -j
MASQUERADE
post-down iptables -t nat -D POSTROUTING -s '192.168.34.0/24' -o ens33 -j
MASQUERADE
```

В качестве сети, в нашем случае **192.168.34.0/24**, укажите выбранную вами сеть, а вместо интерфейса **ens33** укажите тот сетевой интерфейс, который смотрит во внешнюю сеть с доступом в интернет. Если вы используете сетевую конфигурацию по умолчанию, то это будет не физический адаптер, а первый созданный мост **vmbr0**, как на скриншоте ниже:

```
auto lo
iface lo inet loopback

iface ens33 inet manual

auto vmbr0
iface vmbr0 inet static
<------>address  192.168.16.188
<------>netmask  255.255.255.0
<------>gateway  192.168.16.2
<------>bridge-ports ens33
<------>bridge-stp off
<------>bridge-fd 0

auto vmbr1
iface vmbr1 inet static
<------>address  192.168.34.2
<------>netmask  24
<------>bridge-ports none
<------>bridge-stp off
<------>bridge-fd 0

post-up echo 1 > /proc/sys/net/ipv4/ip_forward
post-up <------>iptables -t nat -A POSTROUTING -s '192.168.34.0/24' -o vmbr0 -j MASQUERADE
posr-down<------>iptables -t nat -D POSTROUTING -s '192.168.34.0/24' -o vmbr0 -j MASQUERADE
```

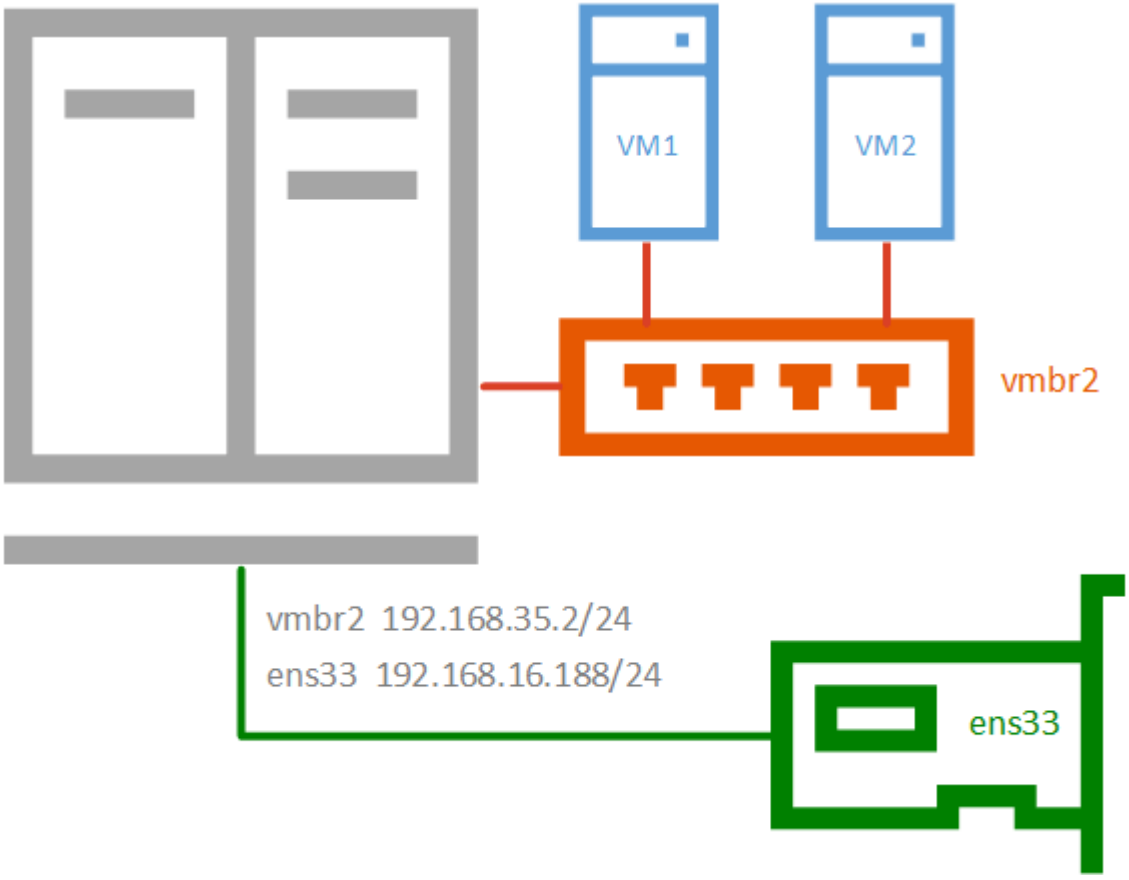
Перезагрузим узел и назначим виртуальной машине или контейнеру созданную сеть (vmbr1), также выдадим ей адрес из этой сети, а шлюзом укажем адрес моста.



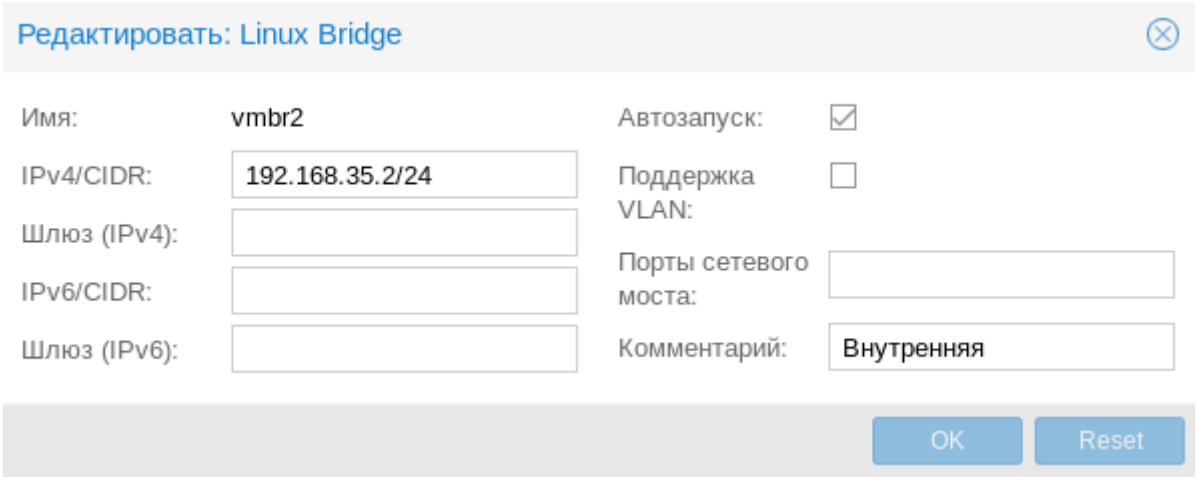
Не забудьте указать доступный адрес DNS-сервера и убедитесь, что виртуальная машина имеет выход в интернет через NAT.

Внутренняя сеть

Позволяет изолировать виртуальные машины от внешней сети и не предоставляет им доступ в интернет, используется в основном в лабораторных целях, когда в качестве шлюза будет выступать одна из виртуальных машин и обычно сочетается на хосте с одной из сетей, имеющих выход в интернет.

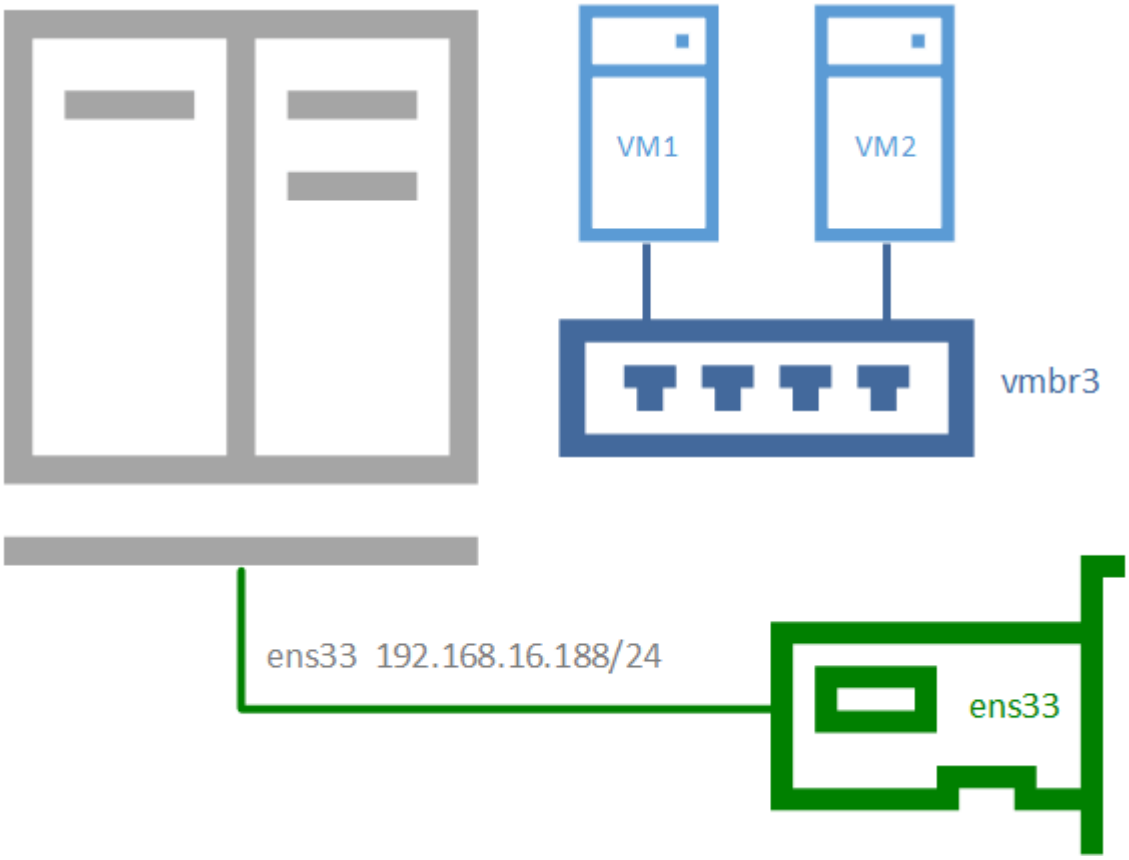


Чтобы получить такую сеть, просто создайте еще один мост без привязки к адаптеру и назначьте ему IP-адрес из любой отличной от используемых сети.

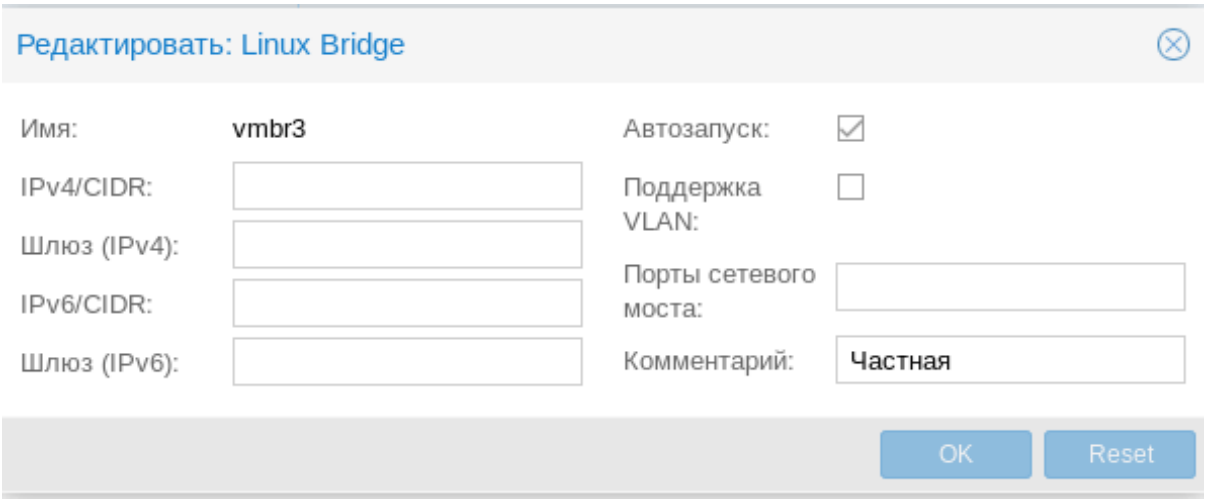


Частная сеть

Разновидность внутренней сети, которая подразумевает изоляцию не только от внешней сети, но и от хоста. Что позволяет получить полностью независимую сеть между виртуальными машинами, может быть полезна в лабораторных условиях, когда нужно смоделировать сеть, адресация которой пересекается с используемыми вами сетями.



Для такой сети просто создайте еще один сетевой мост без каких-либо настроек:



Подобные сети также обычно используются не самостоятельно, а в сочетании с иными типами сетей на хосте.

Организуем службы DNS и DHCP для внутренних сетей

Как вы уже могли заметить все адреса для виртуальных машин во внутренних сетях мы назначали вручную. Но можно это делать автоматически, сняв с себя еще одну заботу, это удобно, особенно в лабораторных и тестовых средах, где виртуальных машин много и назначать им адреса вручную может быть достаточно затруднительно.

В нашем примере мы организуем службы DNS и DHCP для внутренней сети с NAT и просто внутренней сети. Для первой мы должны будет выдавать адрес, шлюз и сервера DNS, для второй просто адрес. Данная конфигурация не является реальной, а создана нами исключительно в учебных целях.

В качестве серверов DNS и DHCP мы будем использовать уже известный нашим читателям пакет **dnsmasq**, который является простым и легким кеширующим DNS и DHCP-сервером. Установим его:

```
apt install dnsmasq
```

Затем перейдем в конфигурационный файл **/etc/dnsmasq.conf** и найдем и приведем к следующему виду параметры:

```
interface= vmbrl, vmbr2
listen-address= 127.0.0.1, 192.168.34.2, 192.168.35.2
```

Здесь мы явно указали интерфейсы и адреса, на которых будет работать наш сервер. С одной стороны, присутствует некоторая избыточность, но лучше так, чем потом, при изменении сетевых настроек в вашей сети неожиданно появится неавторизованный DHCP-сервер.

Затем укажем выдаваемые клиентам диапазоны адресов:

```
dhcp-range=interface:vmbr1,192.168.34.101,192.168.34.199,255.255.255.0,12h
dhcp-range=interface:vmbr2,192.168.35.101,192.168.35.199,255.255.255.0,12h
```

Обратите внимание на формат записи, перед каждой настройкой мы указываем сетевой интерфейс к которой она применяется.

Аналогичным образом зададим нужные DHCP-опции, в нашем случае это Option 3 и 6 (шлюз и DNS-сервер).

```
dhcp-option=interface:vmbr1,3,192.168.34.2
dhcp-option=interface:vmbr1,6,192.168.34.2
dhcp-option=interface:vmbr2,3
dhcp-option=interface:vmbr2,6
```

Если настройки для моста **vmbr1** не вызывают вопросов, то настройки для второй сети следует пояснить. Так как нам нужно передавать ей только IP-адрес и маску, без шлюза и серверов имен, то соответствующие опции следует передать пустыми, иначе будут переданы опции по умолчанию, где в качестве этих узлов будет передан адрес сервера.

Сохраняем конфигурационный файл и перезапускаем службу

```
service dnsmasq restart
```

После чего в виртуальных машинах, подключенных к внутренним сетям, мы можем установить настройки для получения адреса через DHCP и убедиться, что все работает как надо.

- **Категории:**
- Виртуализация,
- Сети и интернет,
- Системному администратору
- **Теги:**
- Dnsmasq,
- Proxmox,
- Виртуализация,
- Сетевые технологии
