

????????? SDN ? Proxmox

В **Proxmox VE SDN (Software Defined Networking)** — это подсистема для управления виртуальными сетями, добавленная начиная с версии **7**. Она упрощает работу с сетевой инфраструктурой внутри кластера и позволяет организовывать гибкие, изолированные и масштабируемые сети для виртуальных машин и контейнеров.

????????? ?????????? ??? SDN ? Proxmox

1. ?????? ?????? SDN

- Упрощает создание частных сетей внутри кластера Proxmox.
- Позволяет изолировать трафик разных пользователей или проектов.
- Дает возможность масштабировать сеть на несколько узлов кластера.
- Автоматизирует настройку маршрутизации, NAT, DHCP и DNS.

2. ?????????????? SDN ? Proxmox

- **Zones (зоны)** — логические сегменты сети (например, для разных проектов или клиентов).
- **VNet (виртуальные сети)** — создаются внутри зон и определяют топологию L2/L3.
- **Controllers (контроллеры)** — управляют сетевыми настройками (например, через EVPN, VXLAN, BGP).
- **IPAM/DNS** — встроенная система для управления IP-адресами и именами.

3. ?????????????????????? ????? ????

- **Simple (VLAN-aware bridge)** — обычная сеть на базе bridge + VLAN.
- **VXLAN** — оверлейная сеть с туннелированием поверх IP.
- **EVPN** — продвинутый вариант для интеграции с “датацентровой” сетью через BGP.
- **QoS zones** — поддержка ограничений пропускной способности.

4. ??????? ??????????????????

1. У вас есть кластер Proxmox на 3 узла.
2. Вы создаете SDN-зону типа **VXLAN**, чтобы виртуалки на разных хостах оказались в одной L2-сети.
3. Proxmox сам поднимает туннели между узлами, и VM/контейнеры видят друг друга, как будто они в одной локальной сети.

????????????

SDN в Proxmox позволяет строить облачные сценарии наподобие **OpenStack/VMware NSX**, но в более простом виде.

????????????????????????????

В данной статье, которая идет как дополнение к видео в начале текста, я постараюсь рассмотреть базовую конфигурацию SDN, которая даст нам понимание как работает SDN в Proxmox

?????????

SDN в Proxmox по умолчанию появилась начиная с версии `Proxmox 8.1`. В случае если у вас система более старая, то сначала вам надо будет ввести в шелле вашей ноды следующую команду, чтобы установить необходимые пакеты:

```
apt update
apt install libpve-network-perl
```

после установки необходимо убедиться, что следующая строка присутствует в конце файла конфигурации `/etc/network/interfaces` на всех нодах (в случае если у вас кластер), чтобы конфигурация SDN была включена и активирована

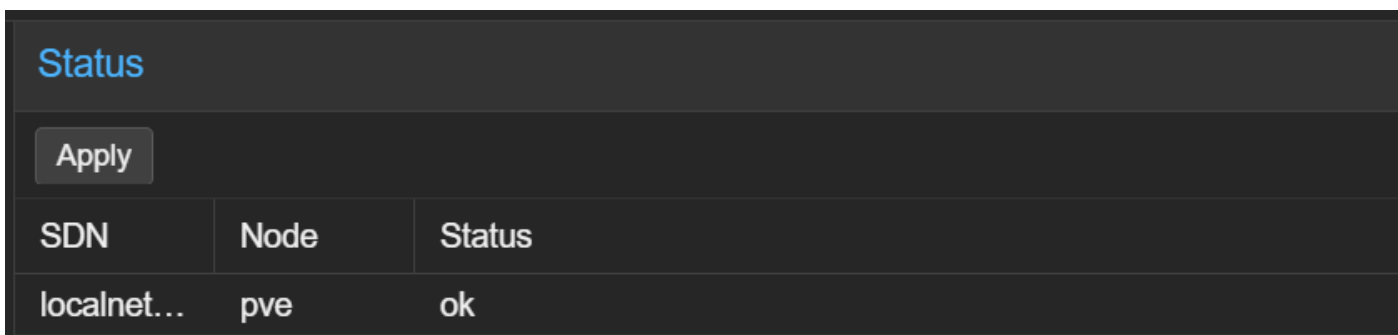
```
source /etc/network/interfaces.d/*
```

Интеграция DHCP во встроенный стек управления IP-адресами PVE в настоящее время использует `dnsmasq` для выдачи DHCP-аренды. Чтобы использовать эту функцию, вам необходимо установить пакет `dnsmasq` на каждой ноды.

```
apt update
apt install dnsmasq
# disable default instance
systemctl disable --now dnsmasq #отключаем dhcp сервер, чтобы он не конфликтовал с вашим роутером
```

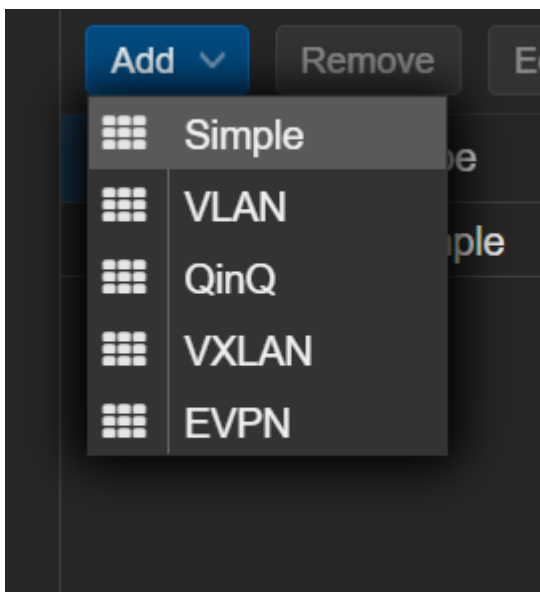
???????????????????????????????? SDN

В меню **Datacenter** кликаете на вкладку **SDN**. Там вы видите список сетей, которые существуют на данный момент. Количество сетей будет зависеть от количество нод. В моем случае будет только одна локальная сеть.



SDN	Node	Status
localnet...	pve	ok

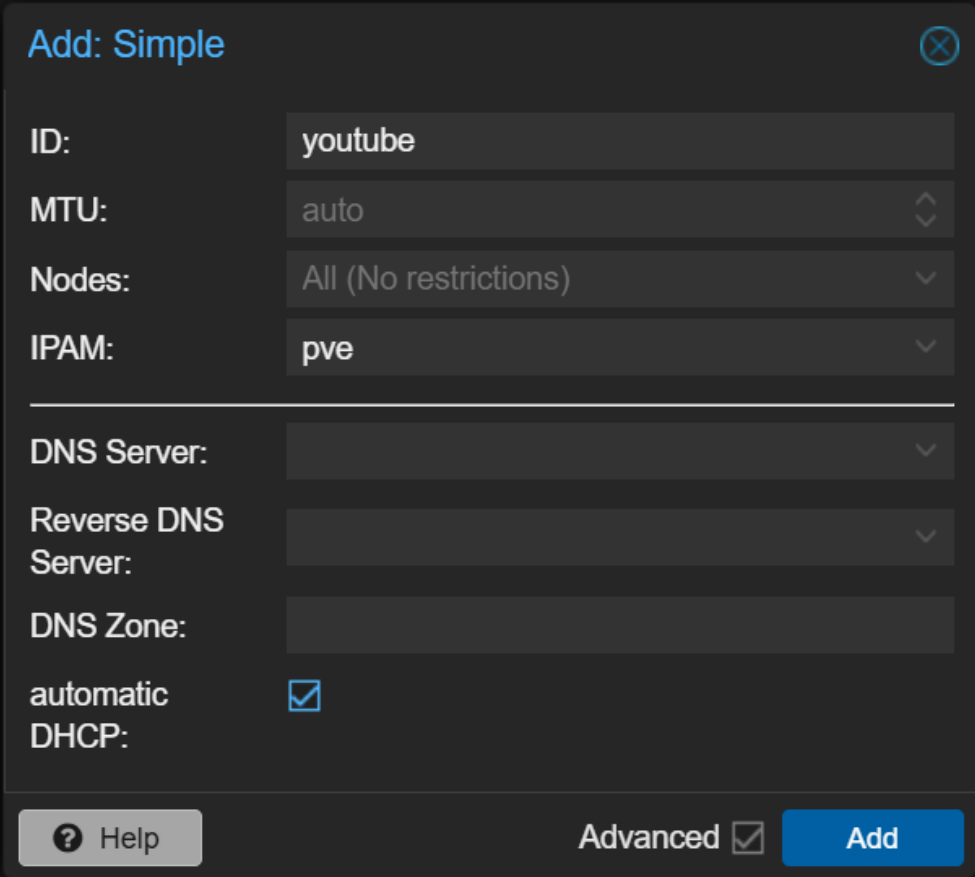
Теперь давайте создадим свою первую (так-то она конечно вторая по счету) локальную сеть. Идем в раздел **Zones** жмем кнопку **add** и видим, что можем выбрать разные типы зон.



Так как у нас с вами будет базовая/простая конфигурация, - то выбираем **simple**.

В качестве ID я указываю название `youtube`. Значения MTU я оставляю по умолчанию. Обратите внимание, иногда регионы рапортуют, что с дефолтным значение MTU которое

равняется 1500 могут быть проблемы. Можете попробовать значение 1460 как вариант. В этом же меню в качестве DHCP я выбираю автонастройку. Жмем **add**.



Add: Simple ✕

ID:

MTU:

Nodes:

IPAM:

DNS Server:

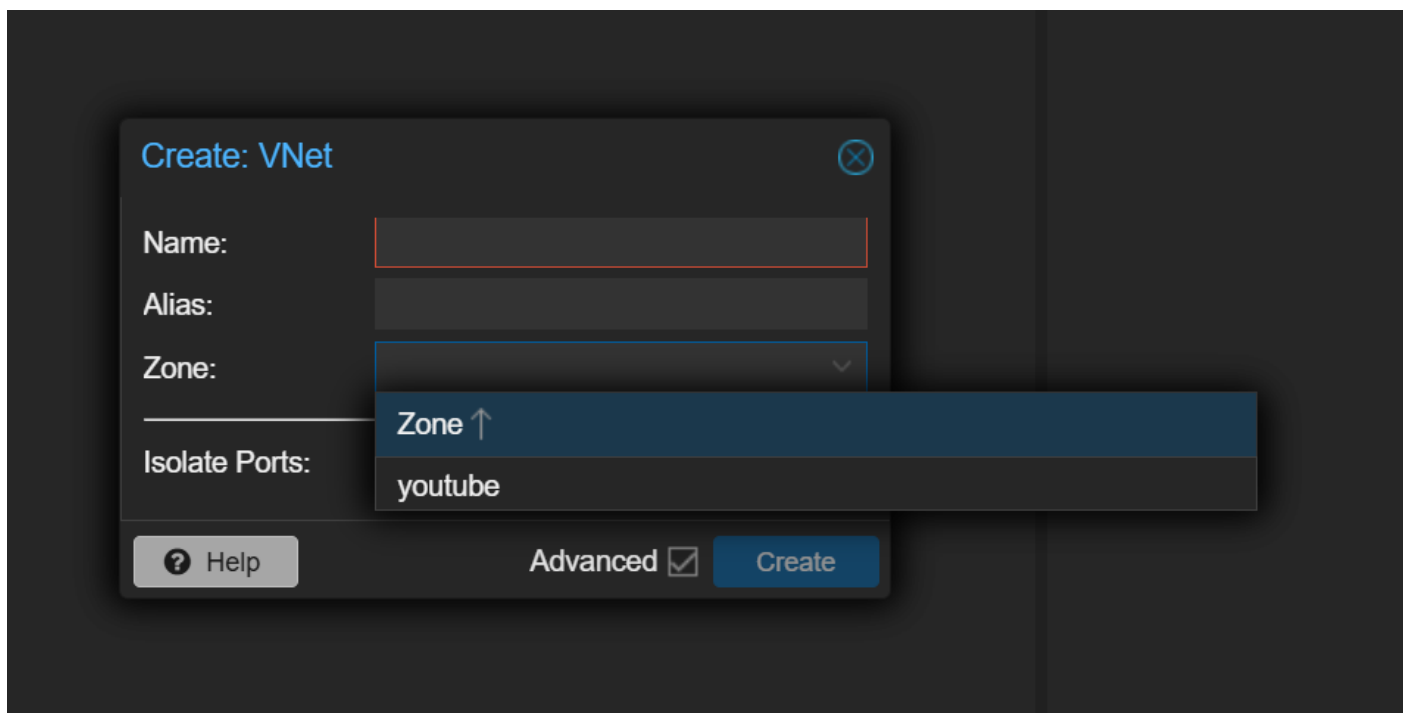
Reverse DNS Server:

DNS Zone:

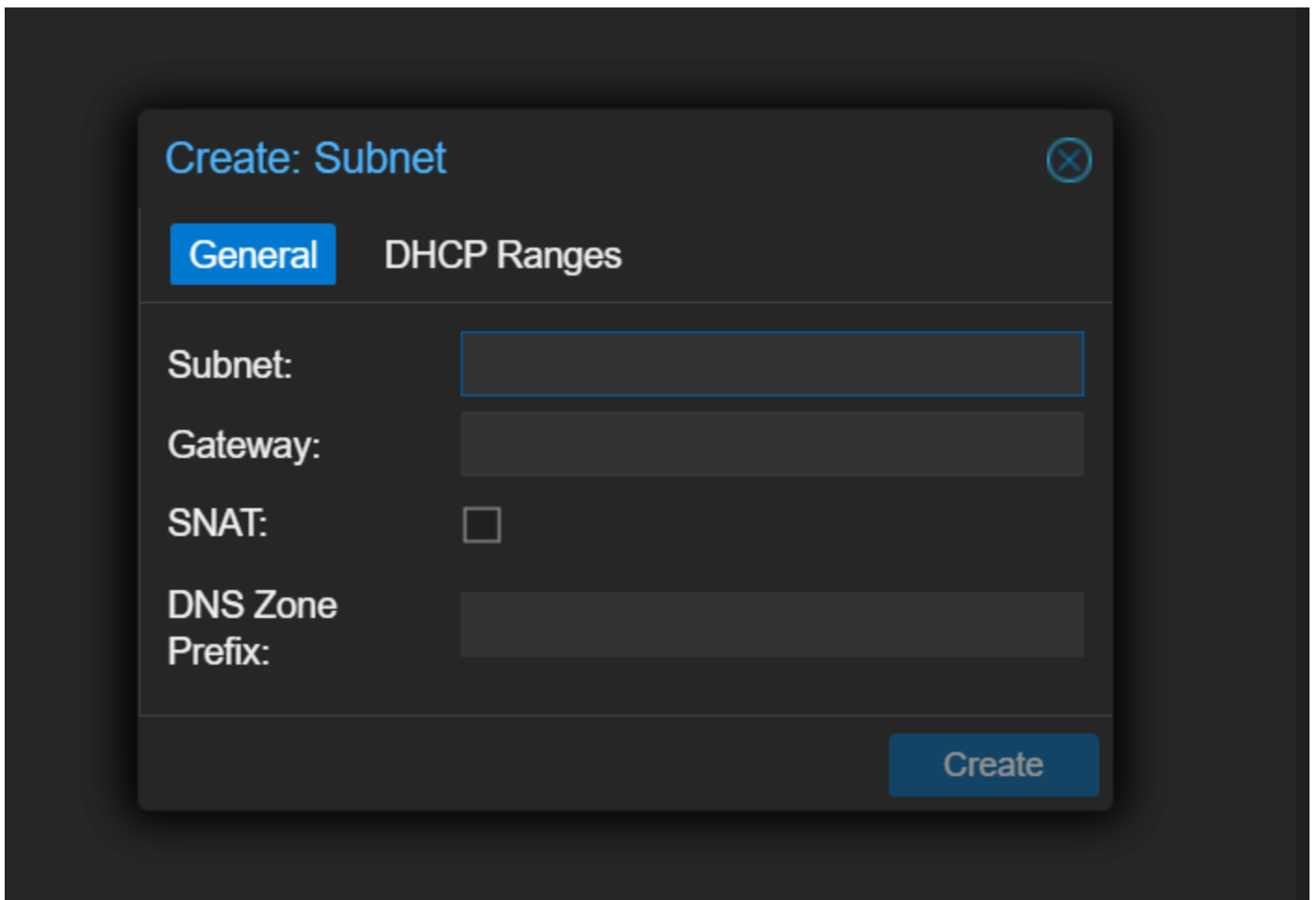
automatic DHCP:

? Help Advanced Add

В итоговом окошке видим настройки нашей зоны. Теперь идем в раздел VNET, жмем кнопку create.



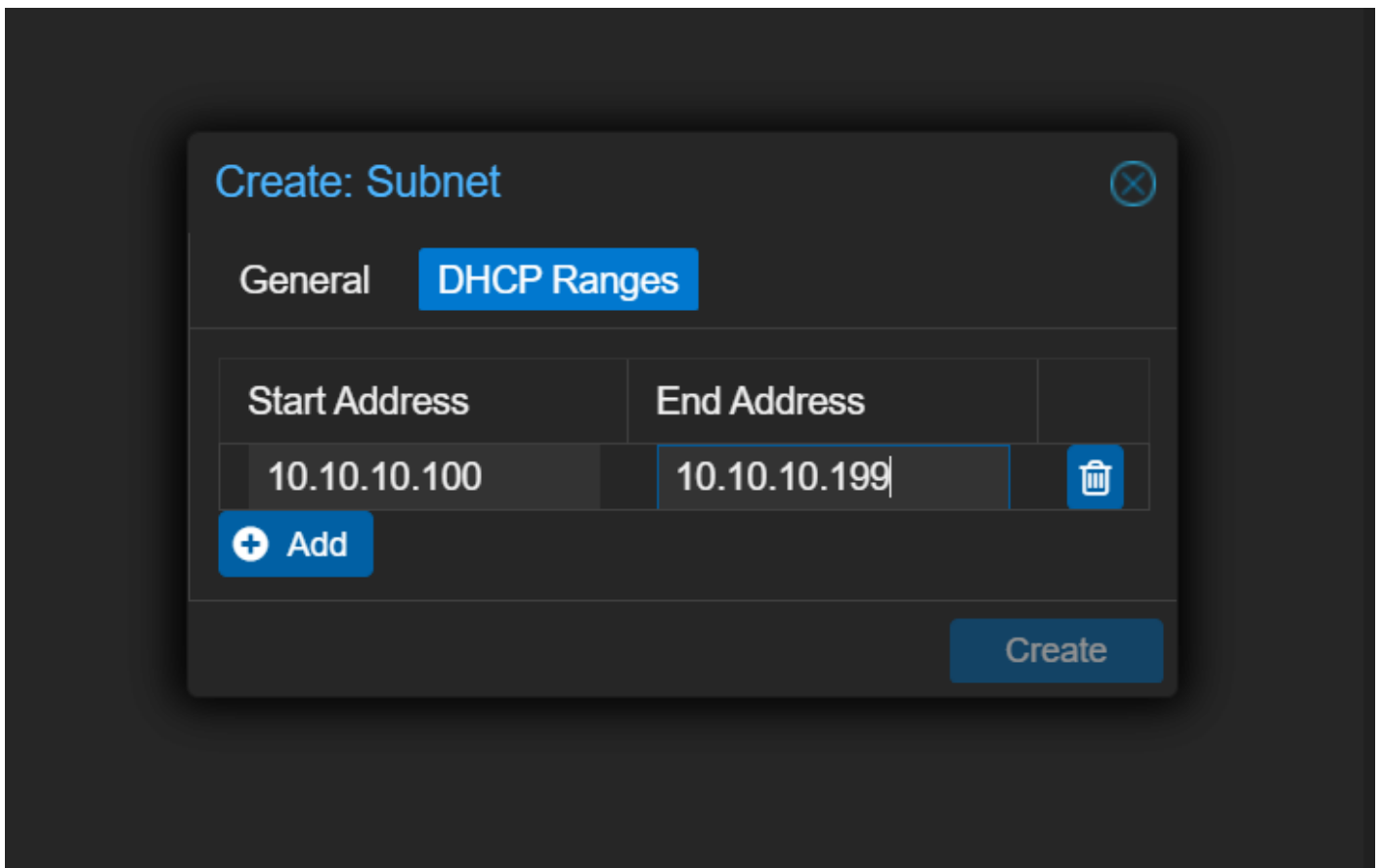
Выбираем имя и alias который вы хотите. Теперь выбираете вновь созданную зону. Все остальное оставляем по умолчанию, потому что в этой статье, как сказано выше, мы делаем только базовую настройку, никаких vlan и прочего. Мы с вами создали нашу первую виртуальную сеть. После того как мы с вами ее создали, у нас появилось новое меню подсеть. В меню vnet выбираем нашу вновь созданную сеть, переходим в меню подсеть `subnet`. Жмем создать. Сейчас мы с вами создадим нашу подсеть, которая будет изолирована от нашей домашней сети.



Задаем адрес нашей подсети. Я люблю нечетные числа, поэтому задам адрес 11.11.11.0/24 (но это неправильно с точки зрения использования в production среде). Адрес шлюза 11.11.11.1 нам нужно поставить галку в разделе snat. Это позволит всем устройствам подсети иметь единый внешний айпи адрес, чтобы они имели выход во внешний мир. Днс не трогаем.

Warning! Правильно создавать сети в пределах диапазона частных сетей для избежания разных неприятных сюрпризов! Поэтому лучше создавать все в пределах 10.10.10.0/24. Но так как наша вновь созданная подсеть должна быть изолирована от доступа из внешнего мира без отдельного на то разрешения, то и так сойдет.

Теперь нам надо задать диапазон адресов в меню DHCP. Не будем изобретать велосипед, зададим диапазон от 100 до 199. Жмем create.



Мы с вами создали зону, vnet, subnet. Теперь чтобы изменения были применены идем в меню **SDN** и жмем **apply**. Теперь когда вы перейдете в список сетей вашей ноды, вы увидите что рядом с **localnetwork** появилась вновь созданная сеть. Все, мы с вами сделали первые шаги, чтобы создать основу для нашей маленькой SDN.

Чтобы проверить как все оно работает в реальности - создадим для тестовых задач простой и пустой LXC контейнер на базе `Ubuntu`. Все делаем как обычно, кроме одного маленького, но очень важного, момента. Когда мы с вами выбираем мост (bridge) который будет использовать наш контейнер в разделе `bridge`

Add: Network Device (veth) ✕

Name: <input style="width: 90%;" type="text" value="(e.g., eth0)"/>	IPv4: <input checked="" type="radio"/> Static <input type="radio"/> DHCP
MAC address: <input style="width: 90%;" type="text" value="auto"/>	IPv4/CIDR: <input style="width: 90%;" type="text" value="None"/>
Bridge: <input style="width: 90%;" type="text" value="youtube1"/>	Gateway (IPv4): <input style="width: 90%;" type="text"/>
VLAN Tag: <input style="width: 90%;" type="text" value="no VLAN"/>	IPv6: <input checked="" type="radio"/> Static <input type="radio"/> DHCP <input type="radio"/> SLAAC
Firewall: <input checked="" type="checkbox"/>	IPv6/CIDR: <input style="width: 90%;" type="text" value="None"/>
	Gateway (IPv6): <input style="width: 90%;" type="text"/>
<hr/>	
Disconnect: <input type="checkbox"/>	Rate limit (MB/s): <input style="width: 90%;" type="text" value="unlimited"/>
MTU: <input style="width: 90%;" type="text" value="Same as bridge"/>	

Advanced

выбираем не `vmbr0`, к которому мы с вами привыкли, а вновь созданную сеть. Все остальное оставляем как обычно. Уже сейчас вы можете вернуться в **Datacenter**, и в подменю **IPAM** увидеть сетевые данные вновь созданного контейнера.

Name / VMID ↑	IP Address ↑	MAC	Gateway	Actions										
<div style="display: flex; align-items: flex-start;"> <div style="margin-right: 10px;"> <ul style="list-style-type: none"> youtube └─ youtube1 └─ 11.11.11.0/24 └─ Gateway └─ 100 </div> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">11.11.11.1</td> <td style="padding: 5px;">BC:24:11:E2:92:49</td> <td style="padding: 5px;">1</td> <td style="padding: 5px;"></td> <td style="padding: 5px; text-align: center;">+</td> </tr> <tr> <td style="padding: 5px;">11.11.11.100</td> <td style="padding: 5px;"></td> <td style="padding: 5px;"></td> <td style="padding: 5px;"></td> <td style="padding: 5px; text-align: center;">✎ ✕</td> </tr> </table> </div>					11.11.11.1	BC:24:11:E2:92:49	1		+	11.11.11.100				✎ ✕
11.11.11.1	BC:24:11:E2:92:49	1		+										
11.11.11.100				✎ ✕										

В нашем случае у него теперь ip адрес `11.11.11.100` После создания и запуска контейнера мы должны проверить - имеет ли наш контейнер выход во внешний мир. Вы конечно можете это сделать простой командой `ping нужный ресурс`, но достаточно будет просто запустить `art update`, и в случае если все в порядке, то контейнер проверит наличие обновлений и выдаст вам результат. Уже это подтверждает, что у вас полностью рабочий контейнер, который имеет доступ во внешний мир. Теперь удостоверимся, что наш контейнер не доступен из внешнего мира. С любой машины внутри вашей локальной сети, но не из шелла Proxmox пропингуем наш контейнер `ping 11.11.11.100` и ответом нам будет тишина. Все потому, что, как я и указывал в начале статьи, наш с вами контейнер находится в изолированной от внешнего мира сети (SDN). Необходимо отметить, что если вы будете пинговать контейнер из шелла вашей ноды, то в данном случае контейнер будет доступен. Это связано с тем, что

сеть создана внутри нашей ноды, каких-либо правил firewall мы еще не настраивали, поэтому с ноды наш контейнер замечательно пингуется. Однако настройка firewall внутри Proxmox - это тема уже другой статьи.

<https://prohomelab.com/posts/proxmox-sdn-guide/>

<https://youtu.be/bKO9lyKR-ms?si=Qz8B9sjqckKv4Ef>

Revision #3

Created 3 December 2025 07:59:47 by itimokhin

Updated 3 December 2025 08:14:38 by itimokhin