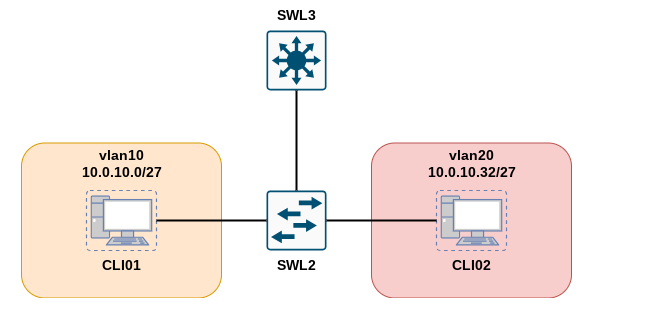
**Open vSwitch - маршрутизация между VLAN (SVI)**

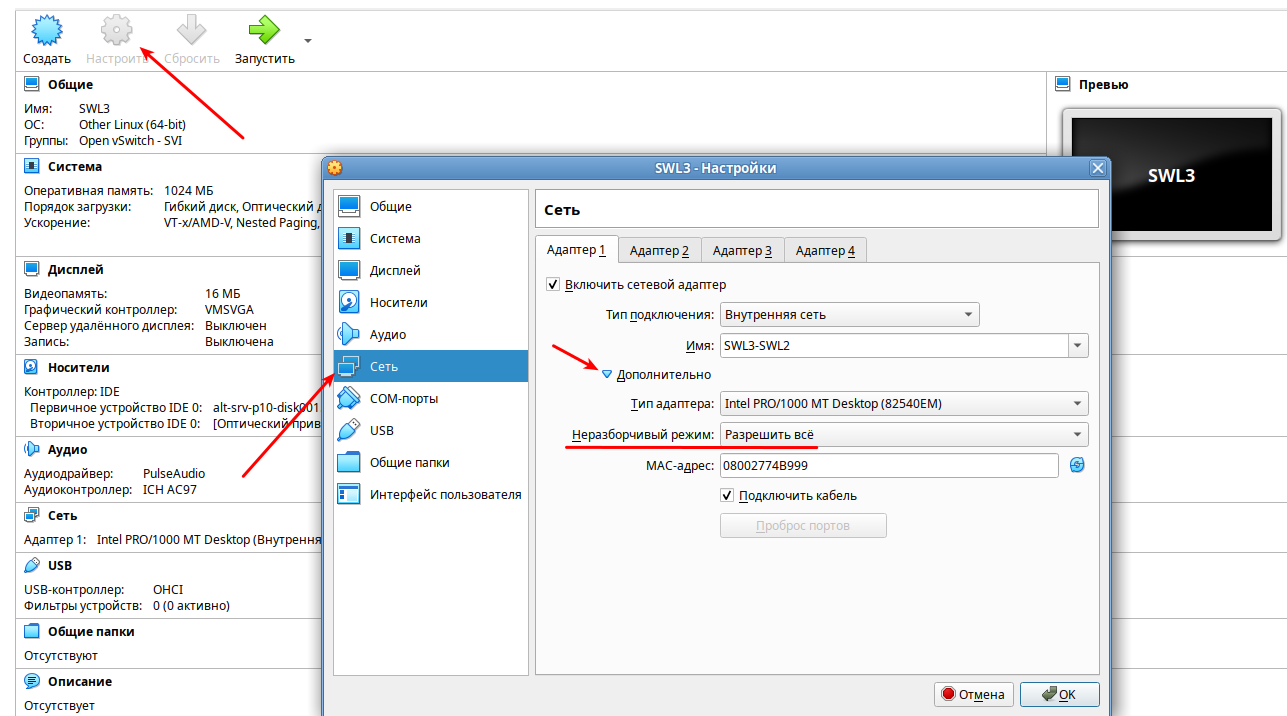
**Топология:**

**Виртуальный стенд:**

* **SWL3 | SWL2** - Альт Сервер;
* **CLI01 | CLI02**- Starterkit (LiveCD)

**Важно!**

* Перевести все сетевые адаптеры на виртуальной машине которая является коммутатором - в неразборчивый режим:
  1. Перейти в настройки ВМ - **SWL3 | SWL2**
  2. Перейти на вкладку **Сеть**
  3. Развернуть флажок **Дополнительно**
  4. В параметры "Неразборчивый режим" - выбрать **Разрешить всё**
  5. Аналогично выполнить и на всех остальных сетевых адаптерах:



***А также зарание установить пакет "openvswitch" на ВМ SWL3 и SWL2  (подключить адаптер NAT, обновить списки пакетов и установить данный пакет)***

**Задача:**

Используя функционал **Open vSwitch** - реализовать маршрутизацию между VLAN с использованием **SVI**

**Реализация:**

**SWL2:**

Выполняем настройку коммутатора 2-го уровня:

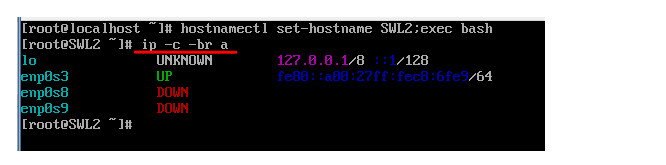
* Задаём имя в соответствие с топологией:

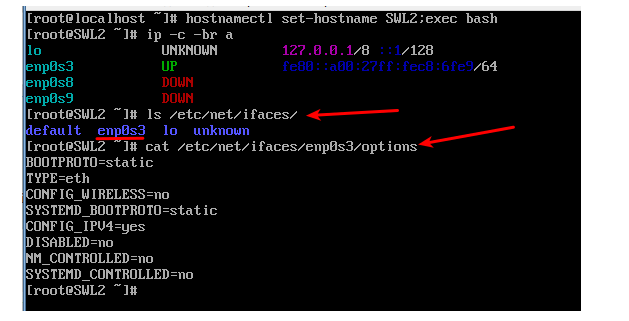
hostnamectl set-hostname SWL2; exec bash

* Смотри на имеющиеся физические интерфейсы:

ip -c -br a

* + Результат:

Так как, интерфей **enp0s3** находится в состоянии **UP** - а в Альт Сервер в минимальной установки в качестве сетевой подсистемы используется **etcnet** - можно сделать вывод, что для интерфейса **enp0s3** - как минимум создана директория по пути **/etc/net/ifaces/<INTERFACE\_NAME>** в которой описан файл **options**:

Таким образом, для добавления физических интерфейсов в bridge коммутатора, они должны быть в состоянии **UP**, можно создать вручную директорию для **enp0s8**и **enp0s9**и описать для каждого файл **options**, или можно использовать "шаблон", который уже присутствует в системе для интерфейса **enp0s3** если он удовлетворяет требования (не должен ожидать адреса по dhcp: BOOTPROTO=static)

* Поднимаем физические интерфейсы: **enp0s8**и **enp0s9**:

cp -r /etc/net/ifaces/enp0s3/ /etc/net/ifaces/enp0s8

cp -r /etc/net/ifaces/enp0s3/ /etc/net/ifaces/enp0s9

* Перезапускаем службу **network**:

systemctl restart network

* + Проверяем:
* Включаем и добавляем в автозагрузку **openvswitch** (должен быть установлен заранее):

systemctl enable --now openvswitch

* Создаём мостовой интерфейс (**bridge**) с именем, например: **ovs0**:

ovs-vsctl add-br ovs0

* Добавляем в мостовой интерфейс физические интерфейсы, которые идут к оконечным устройствам и назначаем их портами доступа "**access**" с "помещением" в соответствующий **VLAN**:
  + **enp0s8**- интерфейс в сторону **CLI01:**
  + **enp0s9**- интерфейс в сторону **CLI02:**

ovs-vsctl add-port ovs0 enp0s8

ovs-vsctl set port enp0s8 tag=10

*ИЛИ*: можно выполнить в одну команду, добавляя интерфейс в bridge и присвоение тега:

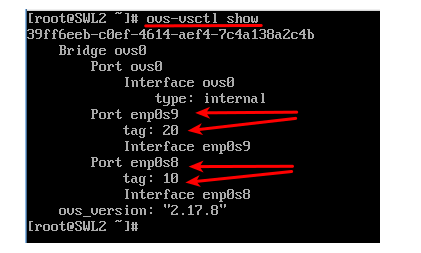
ovs-vsctl add-port ovs0 enp0s8 tag=10

Аналогично выполняем для интерфейса **enp0s9**:

ovs-vsctl add-port ovs0 enp0s9 tag=20

* Проверить:

ovs-vsctl show

* + Результат:
* Настраиваем магистральный канал (**trunk**) в сторону **SWL3**:
  + *будем разрешать пропускать только используемые****VLAN-ы***

ovs-vsctl add-port ovs0 enp0s3

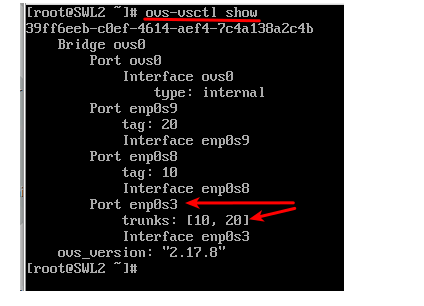
ovs-vsctl set port enp0s3 trunk=10,20

*ИЛИ:*в одну команду

ovs-vsctl add-port ovs0 enp0s3 trunk=10,20

* Проверить:

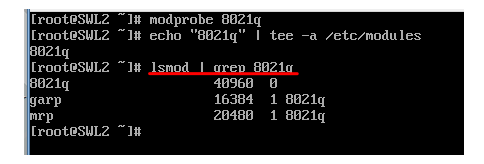
ovs-vsctl show

* + Результат:
* Включаем модуль ядра для поддержки стандарта **IEEE 802.1Q**:

modprobe 8021q

echo "8021q" | tee -a /etc/modules

* + Проверяем:



**SWL3:**

Выполняем настройку коммутатора 3-го уровня для реализации маршрутизации между VLAN:

* Задаём имя в соответствие с топологией:

hostnamectl set-hostname SWL3; exec bash

* Включаем и добавляем в автозагрузку **openvswitch** (должен быть установлен заранее):

systemctl enable --now openvswitch

* Создаём мостовой интерфейс (**bridge**) с именем, например: **ovs0**:

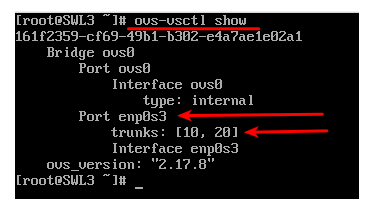
ovs-vsctl add-br ovs0

* Настраиваем магистральный канал (**trunk**) в сторону **SWL2**:
  + *будем разрешать пропускать только используемые****VLAN-ы***

ovs-vsctl add-port ovs0 enp0s3 trunk=10,20

* Проверить:

ovs-vsctl show

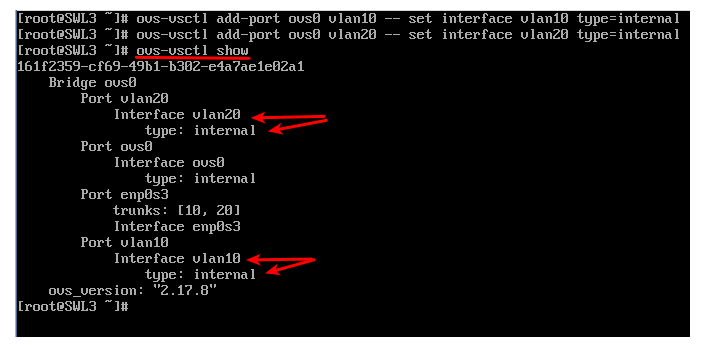
* + Результат:
* 
* Настраиваем так называемые **SVI** - интерфейсы:
  + *с точки зрения Open vSwitch - такие интерфейсы должны быть****type=internal***
    - *создадим виртуальный интерфейс с именем****vlan10****- для подсети****10.0.10.0/27;***
    - *создадим виртуальный интерфейс с именем****vlan20****- для подсети****10.0.10.32/27;***

ovs-vsctl add-port ovs0 vlan10 -- set interface vlan10 type=internal

ovs-vsctl add-port ovs0 vlan20 -- set interface vlan20 type=internal

* Проверить:

ovs-vsctl show

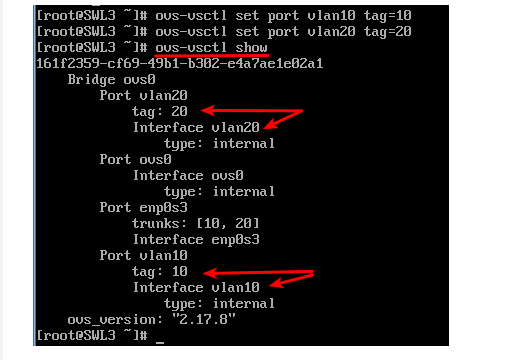
* + Результат:
* Также добавляем для каждого виртуального интерфейса тег с указанием соответствующего VLAN-а:

ovs-vsctl set port vlan10 tag=10

ovs-vsctl set port vlan20 tag=20

* Проверить:

ovs-vsctl show

* + Результат:
* Назначаем на виртуальные интерфейсы IP-адреса из соответствующих подсетей:
  + *пока назначаем средствами****iproute2****- т.е. временно (до перезагрузки), в случае необходимости закрепить IP-адреса на постоянной основе - можно подружить****openvswitch****с****etcnet****;*

ip addr add 10.0.10.1/27 dev vlan10

ip addr add 10.0.10.33/27 dev vlan20

* Включаем виртуальные интерфейсы:

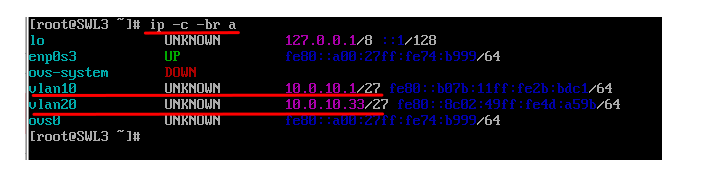
ip link set up vlan10

ip link set up vlan20

ip link set up ovs0

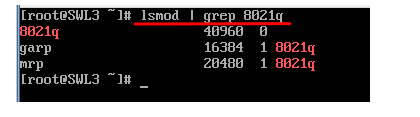
* Проверить:

ip -c -br a

* + Результат:
* Включаем модуль ядра для поддержки стандарта **IEEE 802.1Q**:

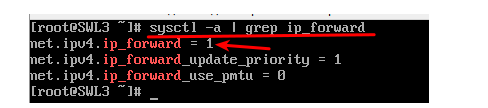
modprobe 8021q

echo "8021q" | tee -a /etc/modules

* + Проверяем:
* 
* Включаем перессылку пакетов (**forwarding**):
  + *временно - но можно и через****etcnet****на постоянной основе;*

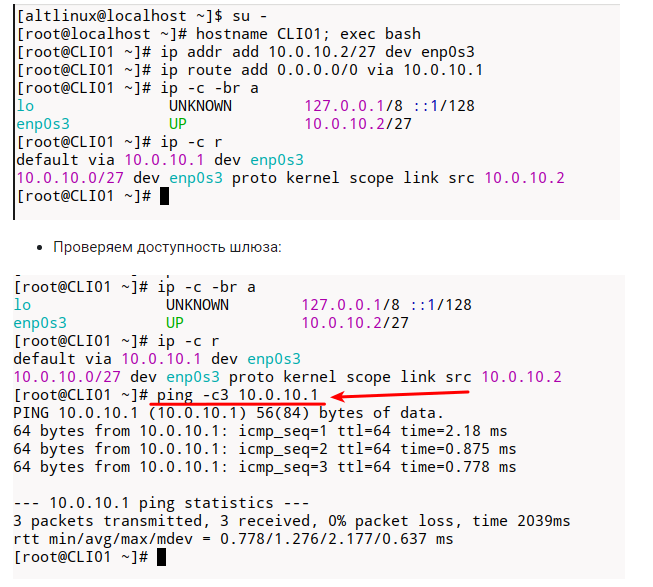
sysctl -w net.ipv4.ip\_forward=1

* + Проверяем:

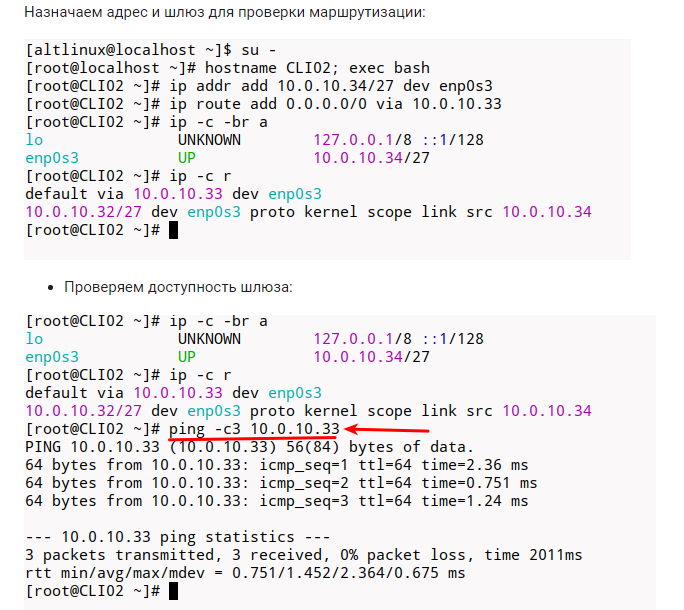


**CLI01:**

Назначаем адрес и шлюз для проверки маршрутизации:

* Проверяем доступность шлюза:

**CLI02:**

**Проверяем маршрутизацию между VLAN:**

