

Some commands

command	discription
hostname + name	Изменяет нынешнее имя
interface + name	Режим настройки интерфейса name
no shutdown	Активировать интерфейс
line vty + fst_num + last_num	Зайти в настройку линии
login local	Создание базы данных пользователей (только в настройке линии)
username + name + secret + password	Создаём нового пользователя

command	discription
show version	Всякая инфа про загрузку (ОС, версия)
show inventory	Показывает установленные модули
show env + ...	Показывает температуру и другие характеристики
show processes + ...	Показывает процессы
show processes cpu history	Показывает графики загруженности
Сессионные	
reload	Перезагружает
show running-config	Показывает содержание запущенного конфига
show startup-config	Показывает содержание стартового конфига
copy running-config startup-config	Перезапись из первого конфига во второй
write	Записывает из рана в стартap
show history	Показывает историю введённых команд
Инфо про интерфейсы	
debug + ...	Включает дебаг
undebug + ... / no debug + ...	Выключает дебаг
show ip interface brief	Показать все интерфейсы
show cdp neighbour	Показать все соседние устройства (подключённые напрямую)
show cdp entry + name	Все данные об устройстве рядом
clear line + port_name	Очистить все процессы
show interface + interface name	Подробная информация про инрефейс

command	discription
ping + ip name ± repeat + amount of repeat	Пинг ip-адреса
show inventory	Информация про подключённые интерфейсы (с нумерацией)
Инфо про рантайм	
show run \ \ incl dhcp	Показывает настройки именно DHCP
Инфо про ACL	
sh ip access	Показать все ACL'ы
Инфо про протоколы	
sh ip protocols	Инфо про протоколы
sh ip arp	
sh ip ospf	

Настройка Telnet

На клиенте:

```
telnet + ip_num + port (можно без порта. HTTP - 80 порт)
pass (при установке соединения попросит пароль)
```

На сервере:

1 способ

```
line vty 0 1869
login local
```

In Global config mode (not in config-line)

```
username usr secret pwd
username cisco secret cisco // -инициализация пользователя
enable secret cisco // -разрешения входа в привелегированный режим
```

2 способ

```
line vty 0 1869
password pass
login
```

Настройка VLAN

```
conf t
int fa0/0
no sh

int fa0/0.20
enc dot 20
ip add 192.168.20.1 255.255.255.0

int fa0/0.13
enc dot 13 nat
ip add 192.168.13.1 255.255.255.0
```

Команда	Описание
int fa0/0.13	Переходим на виртуальный интерфейс
enc dot 20	Инкапсулируем в dot1q (т. е. превращаем в trunk)
enc dot 13 nat	Инкапсулируем в dot1q и говорим, что native VLAN = 13
ip add 192.168.20.1 255.255.255.0	Добавляем виртуальному интерфейсу ip-адресс

Настройка Роутера как хоста

```
conf t
int fa0/0
no sh

int fa0/0.20
ip add 192.168.20.2 255.255.255.0
ip route 0.0.0.0 0.0.0.0 192.168.20.1
```

DHCP

На сервере:

```
conf t
int fa0/0
ip add 192.168.0.1 255.255.255.0
no sh

ip dhcp pool Name
netw 192.168.0.0 /24
default-router 192.168.0.1
```

```
lease N N N
```

```
ip dhcp excl 192.168.0.1 192.168.0.15
```

На клиенте:

```
ip add dhcp
```

Команда	Описание
<code>ip dhcp pool + Name</code>	Инициализируем пул
<code>netw 192.168.1.0 /24</code>	Задаём доступные адреса
<code>default-router 192.168.0.1</code>	Дефолт гейтвей
<code>lease N N N</code>	Настройка времени аренды
<code>ip dhcp excl 192.168.0.1 192.168.0.15</code>	Исключить адреса
<code>ip route + адрес сети + маска сети + default gateway + индекс приоритетности (меньше - приоритетнее)</code>	Настроить default gateway
<code>ip route 0.0.0.0 0.0.0.0 192.168.20.1</code>	Настроить так, чтобы все запросы перенаправлялись на 192.168.20.1

ACL basic (составляем таблицу)

```
ip access-list standart 1  
permit/deny + source_IP
```

```
int + int_name  
ip access-group 1 out
```

ACL extended (составляем таблицу)

```
ip access-list extended 100  
permit/deny + TEG + source_IP + destination_IP + wildcard  
exit
```

```
int + int_name  
ip access-group 1 out
```

Создание виртуальных соседей

```
int lo0
```

```
ip add 192.168.20.2 255.255.255.0
```

IPv6

Comand	Description
<code>ipv6 enable</code>	Запустить функции IPv6. На интерфейсе сразу выдаётся link local address. Не забыть включить интерфейс!
<code>do sh ipv6 int br</code>	Показать краткую инфу про интерфейсы
<code>do sh ipv6 route</code>	Показать таблицу маршрутизации
<code>ipv6 unicast-routing</code>	Из Global config mode запускает маршрутизацию (по умолчанию машина - хост)
<code>ipv6 cef</code>	cef = Cisco Express Forwarding
<code>ipv6 address fe80::2 link-local</code>	Задаём локальный адрес
<code>ipv6 address + prefix + mask + eui-64</code>	Настроить адрес по EUI-64

DHCP IPv6

```
ipv6 dhcp pool Name
address prefix 2001:234::/64
dns-server 2001:234::1
domain-name cisco.com
```

```
##Далее на интерфейсе
ipv6 nd managed-config-flag
ipv6 nd other-config-flag
```

На клиенте:

```
ipv6 add dhcp
```

или для stateless dhcp:

```
ipv6 add autoconfig
```

Команда	Описание
<code>ipv4 dhcp pool + Name</code>	Инициализируем пул
<code>address prefix 2001:234:0 /64</code>	Настройка подсети, в которой работает сервер
<code>ipv6 nd managed-config-flag</code>	Поднять флаг о передаче адреса клиенту (для stateful)
<code>ipv6 nd other-config-flag</code>	Поднять флаг о передаче всей информации кроме адреса

R4

```
conf t

ipv6 unicast-routing
int f0/0
ipv6 address 2001:14::4/64
no sh

do sh ipv6 int br

ipv6 route ::/0 2001:14::1
```

R1

```
conf t
ipv6 unicast-routing

int f0/0
ipv6 address 2001:14::1/64
no sh

int f4/0
ipv6 address 2001:12::1/64
no sh

int g1/0
ipv6 address 2001:13::1/64
no sh
do sh ipv6 int br

ipv6 route 2001:23::/64 2001:13::3
ipv6 route 2001:25::/64 2001:13::3
```

R2

```
conf t
ipv6 unicast-routing

int f0/0
ipv6 address 2001:25::2/64
no sh

int f4/0
```

```
ipv6 address 2001:12::2/64
no sh

int g1/0
ipv6 address 2001:23::2/64
no sh
do sh ipv6 int br

ipv6 route 2001:13::/64 2001:13::3
ipv6 route 2001:14::/64 2001:13::3
```

R3

```
conf t
int g5/0
ipv6 address 2001:23::3/64
no sh

int g1/0
ipv6 address 2001:13::3/64
no sh
do sh ipv6 int br

ipv6 route 2001:14::/64 2001:13::1
ipv6 route 2001:25::/64 2001:23::2
ipv6 route 2001:12::/64 2001:13::1
```

R5

```
conf t

ipv6 unicast-routing
int f0/0
ipv6 address 2001:15::5/64
no sh

do sh ipv6 int br
ipv6 route ::/0 2001:25::2
```

NAT/PAT

R1

```
ip access-list extended 100
permit ip 192.168.134.0 0.0.0.255 any
```

```

int g1/0
ip nat inside

int f0/0
ip nat outside

// Далее в priveleged mode
ip nat inside source list 100 interface f0/0 overload
do sh ip nat tra
do sh ip nat stat

```

ПО ЧАСТЯМ	ЗАЧЕМ
ip nat	командуем для nat
inside	говорим, что начинаем разбирать логику трансляции
source list 100	Source, который проходит правила ACL 100
interface f0/0	Source выше заменяется на IP на интерфейсе f0/0
overload	Указание, чтобы использовалось PAT

PAT+NAT

```

\\ in global config
\\ инициализировали пул
ip nat pool CiscoPool 10.0.0.1 10.0.0.10 netmask 255.0.0.0
ip nat inside source list 100 pool CiscoPool overload

```

NAT

Настраивается как PAT+NAT, но без **overload**

Tunnel setting

Comand	Description
int tun0	поднимаем интерфейс для туннеля
tunnel mode + mode_name	Настраиваем туннель
tunnel mode gre ip	Настраиваем туннель IP через GRE
tunnel source e2/3	Отправитель через физический интерфейс (плохой дизайн)
tunnel destination 192.168.45.5	Получатель через IP
keepalive	Мониторинг состояния туннеля


```
int tun0
ip add 192.168.25.2 255.255.255.0 // логический адрес туннеля
tunnel mode gre ip
tunnel source Ethernet2/3
tunnel destination 192.168.45.5 // адрес физического интерфейса
keepalive
```

OSPF setting

Command	Description
router ospf 1	инициализируем процесс 1. каждый процесс - одно дерево
router-id 3.3.3.3	Ставим ID на роутере
network 0.0.0.0 255.255.255.255 area 0	Ставим сети, участвующие в OSPF. Маска wildcard
sh ip ospf data	Показать всю информацию по OSPF
clear ip ospf process	Очистить таблицу маршрутизации

R2

```
router ospf 1
network 192.168.23.0 0.0.0.255 area 0
passive-int e2/1
```

R3

```
network 0.0.0.0 255.255.255.255 area 0
router-id 3.3.3.3
```

R4

```
router ospf 1
network 0.0.0.0 255.255.255.255 area 0
int lo0
ip add 4.4.4.4 255.255.255.255
int lo1
ip add 44.44.44.44 255.255.255.255
```

R5

```
router ospf 1
network 192.168.45.0 0.0.0.255 area 0
```

HDLC

command	discriptoin
<code>encapsulation hdlc</code>	Выставить нужную инкапсуляцию
<code>show controllers + serial number</code>	Lists whether a cable is connected to the interface, and if so, whether it is a DTE or DCE cable
<code>clock rate</code>	Установить на DCE скорость передачи (в GNS3 все считают, что они DCE)

R1

```
conf t

int s3/4
enc hdlc - по умолчанию
ip add 192.168.0.1 255.255.255.0
no sh
```

R2

```
conf t

int s3/4
enc hdlc - по умолчанию
ip add 192.168.0.2 255.255.255.0
no sh
```

PPP

R1- host, R2 - client

CHAP setting:

R1:

```
username Cisco3 password Pwd3
int s3/3
enc ppp
no sh
ip add 192.168.13.1 255.255.255.254

ppp authe chap
```

```
ppp chap hostn Cisco1
ppp chap password Pwd3
```

R2:

```
conf t

int s3/1
enc ppp
no sh
ip add 192.168.13.0 255.255.255.254

ppp chap host Cisco3
ppp chap pass Pwd3
```

PAP setting

R1:

```
username Cisco2 password Pwd2
int s3/2
enc ppp
no sh
ip add 192.168.12.1 255.255.255.252
ppp authe pap
```

R2:

```
conf t

int s3/1
enc ppp
no sh
ip add 192.168.12.2 255.255.255.252

ppp pap sent Cisco2 pass Pwd2
```

Different networks

Разные подсети успешно подключаются:

R1:

```
int s3/4
enc ppp
```

```
no sh
ip add 192.168.14.1 255.255.255.252
```

R2:

```
conf t

int s3/1
enc ppp
no sh
ip add 192.168.41.1 255.255.255.252
```

Address assignment

R1:

```
int s3/5
enc ppp
no sh
ip add 192.168.15.1 255.255.255.254
peer default ip address 192.168.15.0
```

R2:

```
conf t

int s3/1
enc ppp
no sh
ip add negotiated
```

MLPPP

Как LAG, то есть повторное соединение

R1:

```
##Создаём новый логический интерфейс, с которого будем общаться
int multilink 1
enc ppp
ppp multilink group 1
ip add 192.168.21.1 255.255.255.252

int s3/1
enc ppp
ppp multilink group 1
```

```
no sh
```

```
int s3/0  
enc ppp  
ppp multilink group 1  
no sh
```

R2

```
##Создаём новый логический интерфейс, с которого будем общаться
```

```
int multilink 1  
enc ppp  
ppp multilink group 1  
ip add 192.168.21.2 255.255.255.252
```

```
int s3/1  
enc ppp  
ppp multilink group 1  
no sh
```

```
int s3/0  
enc ppp  
ppp multilink group 1  
no sh
```

PPP unnumbered

Создаём loopback и назначаем все интерфейсы на него (то есть с этого адреса будут слаться пакеты)

R1:

```
int lo0  
ip add 1.1.1.1 255.255.255.255
```

```
int s3/2  
ip unnumbered lo0
```

```
int s3/3  
ip unnumbered lo0
```

```
int s3/4  
ip unnumbered lo0
```

```
int s3/5  
ip unnumbered lo0
```

PPP neighbour route

Позволяет очистить таблицу маршрутизации, чтобы не было видно соседей. Лучше не делать, если подключён сосед из известной подсети, ведь иначе не сможет пройти ping.

```
no peer neig
do sh ip route
do clear ip route *
do sh ip route
```

Frame relay

command	discription
serial 0/1/1.1 + multipoint / point-to-point	Переключиться на интерфейс (сабинтерфейс). multipoint - если на него забинжены несколько dcli. Иначе - point-to-point
enc frame-r	Включить инкапсуляцию. Если связаны устройства cisco и не cisco, то в конце можно дописать ietf
frame-relay interface-dlci + dlci_num	Настроить на интерфейсе (или сабинтерфейсе номер DLCI)
do sh frame pvc	посмотреть инфу про статистику и FR
do sh frame map	посмотреть инфу про подключения
frame-relay map ip + ip_num + dlci_num + broadcast ex: frame-relay map ip 199.1.1.1 51 broadcast	Настроить статический маршрут до ip_num через некоторую линку с dlci_num Нужно, если выключен inverse-ARP, который по DLCI узнаёт IP
debug frame-relay lmi	Посмотреть информацию по соединению линка
frame-rel switching	включить режим FR-switch
frame intf + dte / dce	DTE - клиент, DCE - провайдер. в GNS3 в сторону тглупого FR-Switch'a нужно всегда ставить dte
frame-relay route + dlci_in_num + interface + int_out_name + dlci_out_num ex: frame-relay route 135 interface s4/1 101	Настройка маршрута на входном интерфейсе роутера, настроенного под FR-switch. Причём int_name может быть и туннелем!

R4

```
conf t
int se4/0
no sh
```

```
enc fr
```

```
int s4/0.102 point-to-point
frame-relay interface-dlci 102
exit
ip add 192.168.24.4 255.255.255.0
```

```
do sh frame pvc
```

R2

```
conf t
int se4/0
no sh
```

```
enc fr
```

```
int s4/0.104 point-to-point
frame-relay interface-dlci 104
exit
ip add 192.168.24.2 255.255.255.0
```

Важно! Чтобы [OSPF](#) работал корректно, включать нужно на каждом multicast интерфейсе прописать настройку: `ip ospf network point-to multipoint`. Подробнее в записи об [OSPF](#)

HSRP setting

command	discription
do sh standby br	Посмотреть информацию
standby + group_num + ip +ip_num ex: standby 1 ip 192.168.10.123	Присязать к виртуальному ip
standby + group_num + preempt	Настроить preempt (перехват управления active при рабочем active)
standby + group_num + priority + priority_num	Изменить приоритет роутера
standby + group_num + name name_str	Назначить имя
standby version + 1 2	Назначить версию (на всех роутерах в группе должны быть одинаковые версии)

Важно, что виртуальный IP не совпадает с IP на интерфейсе. HSRP нужно настраивать на каждом задейственном интерфейсе (или сабинтерфейсе, если дело касается VLAN)
На каждом VLAN можно сделать свою группу и по приоритету сделать так, что каждую подсеть

обслуживает свой роутер (остальные для неё - запасные)

R1

```
int f0/0
ip add 192.168.10.1 255.255.255.0
no sh

standby 1 ip 192.168.10.123
standby 1 preempt
standby 1 timers 3 10

/// 1 - номер группы
```

Можно использовать `standby 1 track 1 decrement 20`, где [track](#) это некоторое отслеживаемое условие

Track

`track 1 ip route 3.3.3.3/24 reachability` отслеживает в данном случае доступность (присутствие в таблице маршрутизации) маршрута

VRRP setting

command	discription
<code>do sh vrrp br</code>	Посмотреть информацию
<code>vrrp + group_num + ip +ip_num</code> ex: <code>vrrp 1 ip 192.168.10.123</code>	Присязать к виртуальному ip
<code>vrrp + group_num + preempt</code>	Настроить preempt (перехват управления active при рабочем active)
<code>vrrp + group_num + priority + priority_num</code>	Изменить приоритет роутера

Main router upstairs

```
conf t
int f0/0
no sh

int fa0/0.204
enc dot 204
ip add 192.168.204.20 255.255.255.0

int fa0/0.203
enc dot 203
```



```
ip add 192.168.203.20 255.255.255.0
```

Main in 201 subn, backup in 202 subn

```
conf t
int fa0/0
no sh

int fa0/0.202
enc dot 202
ip add 192.168.202.21 255.255.255.0

int fa0/0.201
enc dot 201
ip add 192.168.201.21 255.255.255.0

int fa0/0.203
enc dot 203
ip add 192.168.203.21 255.255.255.0

int f0/0.201
vrrp 201 ip 192.168.201.1
vrrp 201 priority 200
vrrp 201 preempt
int f0/0.202
vrrp 202 ip 192.168.202.1
vrrp 202 preempt
```

Main in 202 subn, backup in 201 subn

```
conf t
int fa0/0
no sh

int fa0/0.202
enc dot 202
ip add 192.168.202.41 255.255.255.0

int fa0/0.201
enc dot 201
ip add 192.168.201.41 255.255.255.0

int fa0/0.204
```

```
enc dot 204
ip add 192.168.204.41 255.255.255.0

int f0/0.201
vrrp 201 ip 192.168.201.1
vrrp 201 preempt
int f0/0.202
vrrp 202 ip 192.168.202.1
vrrp 202 priority 200
vrrp 202 preempt
```

All daughter routers

```
ip route 0.0.0.0 0.0.0.0 192.168.201.1 - default gateway to virtual router
```

Можно использовать `standby 1 track 1 decrement 20`, где [track](#) это некоторое отслеживаемое условие

GLBP

Всё как в [VRRP](#) и [HSRP](#)

R1:

```
int f1/0
ip add 192.168.123.1 255.255.255.0
no sh

glbp 1 ip 192.168.123.123
glbp 1 preempt
glbp 1 load-balancing weighted
```

Zone-based firewall

command	discription
zone security + zone_name ex: zone security IN	Создать защищённую зону
zone-pair secu + pair_name + source zone_source_name dest zone_dest_name ex: zone-pair secu IN2DMZ source IN dest DMZ	Создать пару зон (правило в одну сторону может отличаться от правила в другую)
class-map type inspect + match- all / match-any + class_map_name	Создать карту класса (её нужно будет применять в карту политики) match-all = если входят все условия

command	discription
ex: class-map type inspect Telnet11	match-any = если подходит хотя бы одно условие Можно не писать, по умолчанию match-all
match protocol + protocol_name ex: match protocol telnet match access-group name ACL1	Внутри настройки карты класса добавить протоколы, входящие в класс Для второго примера необходимо создать отдельно ACL (например в ячейке ниже)
ip access-list extended VLAN2 permit ip 192.168.12.0 0.0.0.255 any	ACL Для подсети из конкретного VLAN
policy-map type inspect + policy_map_name ex: policy-map type inspect IN2DMZ	Создать карту политики
class + class_map_name ex: class Telnet11	Внутри настройки карты политики можно привязать карту класса
inspect - туда-сюда pass - только туда drop - без уведомлений отбросить весь трафик, попавший в этот класс, в обоих направлениях	Для добавленной карты класса нужно выбрать действие - одно из указанных
zone-pair secu + pair_name serv type inspect policy_map_name ex: zone-pair secu IN2DMZ serv type inspect IN2DMZ	В настройках пары зон указать, какой политики придерживаться
int + int_name zone-member sec + zone_name ex: int g1/0.12 zone-member sec IN	Добавляем в зону интерфейс, навешивая на него необходимую зону (спасибо, капитан очевидность)

SLB

command	discription
do show ip slb reals	Посмотреть реальные сервера, соединённые по SLB
do show ip slb vservers	Посмотреть информацию про то, как нас видят клиенты
do show ip slb serverfarm detail	Посмотреть дополнительные данные по серверам

L3:

command	discription
ip slb serverfarm + serv_farm_name	создать набор серверов
nat server	сделать так, чтобы IP подменялся на виртуальный
real + ip_num + port_num	инициализируем сервер и порт, который нужно будет обслуживать (один сервер может добавляться несколько раз с разными портами)
maxconns + num	заявить максимальную загрузку сервера
weight + num	заявить, как много нагрузки помещать на сервере за один круг round-robin
inservice	добавляем сервер в набор
ip slb vserver virt_serv_name	создать виртуальный сервер
serverfarm + serv_farm_name	инициализировать в виртуальном сервере набор из серверов
virtual + virt_ip_num + tcp / udp + port_num	привязка виртуального ip и типа соединения к виртуальному серверу
inservice	добавляем набор серверов в виртуальный сервер

```
ip slb serverfarm sf1
nat server
real 192.168.1.2 23 -- заходим в сервер R2, 23 - это порт телнет
inservice -- добавить сервис в ферму
ip slb serverfarm sf1
real 192.168.1.3 23
inservice
exit
```

```
ip slb vserver SF1
serverfarm sf1
virtual 1.1.1.1 tcp 23
inservice
exit
```

L2:

R1

```
ip slb serverfarm SF2
real 192.168.1.5
inservice
real 192.168.1.6
```

```
inervice
```

```
ip slb vserver VS1
virtual 1.1.1.1 tcp telnet
no inervice - пока перенастраиваем лучше выключить
serverfarm SF2
inervice
```

На серверах:

```
int lo0
ip add 1.1.1.1 255.255.255.255
```

Protected VLAN

R1:

```
// Настроим VLAN
vlan 10
name Users
exit

// Назначим порты в VLAN
interface range fa0/1 - 2
switchport mode access
switchport access vlan 10
switchport protected
spanning-tree portfast
no shutdown

interface fa0/24
switchport mode access
switchport access vlan 10
spanning-tree portfast
no shutdown
```

В данном случае fa0/1 и fa0/2 - protected. Они не могут общаться друг с другом, но каждый из них может общаться с non-protected fa0/24

PVLAN

Configuring a VLAN as a PVLAN

```
configure terminal
vlan 202
private-vlan primary
```

```
end
do sh vlan private-vlan
```

```
configure terminal
vlan 303
private-vlan community
end
show vlan private-vlan
```

```
configure terminal
vlan 440
private-vlan isolated
end
show vlan private-vlan
```

Associating a Secondary [VLAN](#) with a Primary [VLAN](#)

```
configure terminal
vlan 202
private-vlan association 303-307,309,440
end
show vlan private-vlan
```

Configuring a Layer 2 Interface as a PVLAN Promiscuous Port

```
configure terminal
interface fastethernet 5/2
switchport mode private-vlan promiscuous
switchport private-vlan mapping 200 2 // 200 - primary, 2 - secondary vlan
end
```

Configuring a Layer 2 Interface as a PVLAN Host Port

```
configure terminal
interface fastethernet 5/1
switchport mode private-vlan host
switchport private-vlan host-association 200 2 // 200 - primary, 2 - secondary
end
```

Configuring a Layer 2 Interface as a PVLAN Trunk Port

```
configure terminal
interface fastethernet 5/1
switchport private-vlan association trunk 200 2 // 200 - primary, 2 - secondary
```

```
switchport mode private-vlan trunk
end
```

Permitting Routing of Secondary [VLAN](#) Ingress Traffic

```
configure terminal
interface vlan 202
private-vlan mapping add 303-307,309,440
end
show interfaces private-vlan mapping
```

VRF

command	discription
do show ip vrf	показать инфу про VRF
ip vrf + vrf_name	создать VRF
description + text	добавить описание
ip vrf forwarding + vrf_name	определить интерфейс (или сабинтефейс) в VRF
router ospf 1 vrf + vrf_name	Запустить OSPF в VRF

R1

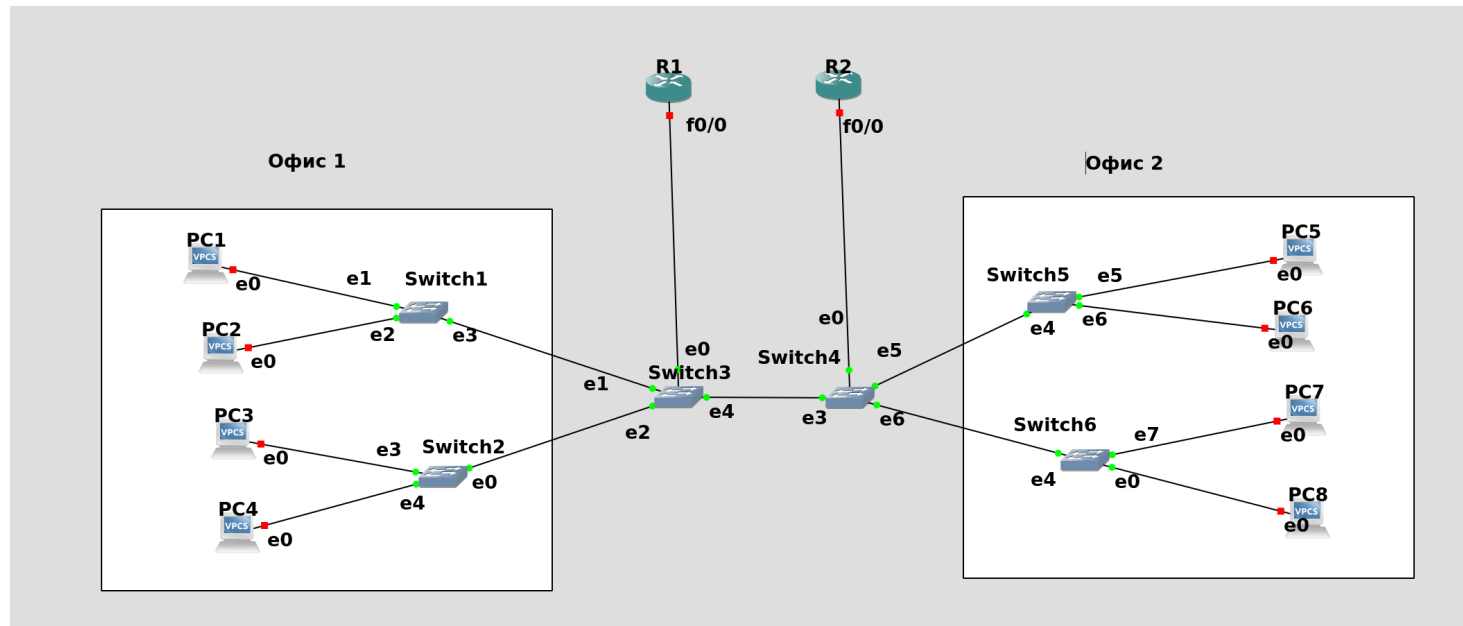
```
ip vrf Cust1
description Customer 1 with OSPF
ip vrf Cust2
description Customer 2 with statics

int g2/0
ip vrf forwarding Cust1 -- если настраиваем вrf после настройки адреса, то его
надо перенастроить
ip address 192.168.12.1 255.255.255.0
no sh

router ospf 1 vrf Cust1
net 192.168.12.0 0.0.0.255 area 0
red con sub
```

QnQ

Чтобы можно было связывать вот такие схемы



Где **VLAN** у компов из разных офисов совпадают:

ПК	VLAN	IP-адрес
PC1	2	192.168.0.2
PC2	3	192.168.1.2
PC3	2	192.168.2.2
PC4	3	192.168.3.2
PC5	2	192.168.0.3
PC6	3	192.168.1.3
PC7	2	192.168.2.3
PC8	3	192.168.3.3

Нужно настроить свитчи так:

1. SW1,2,5,6 - access с нужным VLAN в сторону клиента и trunk в сторону провайдерского оборудования

2. SW3,4:

Switch3 configuration

General

Name: Switch3

Console type: none

Settings

Port: 8

VLAN: 1

Type: dot1q

QinQ EtherType: 0x8100

Ports

Port	VLAN	Type
0	1	dot1q
1	11	qinq
2	12	qinq
3	1	access
4	1	dot1q
5	1	access

Switch4 configuration

General

Name: Switch4

Console type: none

Settings

Port: 8

VLAN: 12

Type: qinq

QinQ EtherType: 0x8100

Ports

Port	VLAN	Type
2	1	access
3	1	dot1q
4	1	access
5	11	qinq
6	12	qinq
7	1	access

3. Ha R1:

```
conf t
int f0/0
no sh

int f0/0.112
encapsulation dot1q 11 second-dot1q 2
ip add 192.168.0.1 255.255.255.0

int f0/0.113
encapsulation dot1q 11 second-dot1q 3
```

```
ip add 192.168.1.1 255.255.255.0
```

```
int f0/0.122
```

```
encapsulation dot1Q 12 second-dot1q 2
```

```
ip add 192.168.2.1 255.255.255.0
```

```
int f0/0.123
```

```
encapsulation dot1Q 12 second-dot1q 3
```

```
ip add 192.168.3.1 255.255.255.0
```

command	discription
encapsulation dot1Q + QnQ_VLAN_num second-dot1q + VLAN_num	Настроить на декапсуляцию сначала внешнего тега от QnQ (QnQ_VLAN_num), а потом - внутреннего от обычного VLAN (VLAN_num)