Подготовка PNETLab

Образы и ресурсы

Узлы	Образ	Ресурсы на один узел
SPINE1/2, LEAF1/2	`nxosv9k-9300v-10.5.2.F`	4 vCPU, 12288 МБ RAM, 5 NIC
PC1/2	`vios-adventerprisek9-m.spa.159-3.m9`	1 vCPU, 1024 МБ RAM

В сумме: 18 назначенных vCPU и 50 ГиБ гостевой RAM; отдельно требуется память для PNETLab. Проверенный хост: 24 vCPU и около 63 ГиБ RAM. Это одна полная одновременно работающая топология, не несколько студенческих экземпляров.

Клиенты IOSv моделируют конечные узлы, не участвуют в маршрутизации underlay. Для ICMP и ARP достаточно их консолей. Для `curl` можно заменить клиентов Linux, сохранив адреса и MTU 1500.

Импорт и доступ

1. Используйте static_initial.zip либо evpn_initial.zip. Альтернатива — преподаватель размещает `.unl` в `/opt/unetlab/labs/VXLAN-Course/`.
2. Создайте персональную копию с **новым UUID** средствами PNETLab. Простое переименование файла не создаёт независимую сессию.
3. Убедитесь, что четыре Nexus имеют по 4 vCPU и 12 ГиБ RAM. Не используйте старые значения шаблона 2 vCPU / 8 ГиБ.
4. Запускайте узлы по одному. Первичная загрузка Nexus и применение bootstrap занимают несколько минут. Дождитесь приглашений `SPINE1#`, `LEAF1#` и т. д.
5. Учебная учётная запись Nexus: **`admin / LabVxlan2026!`**. Она публичная, только для изолированной работы. В стартовых конфигурациях уже заданы hostname, loopback, IP/MTU физических uplink; OSPF и VXLAN студент настраивает самостоятельно. PC уже имеют IP.
6. Проверьте `show version`, `show interface brief`, `show running-config`. Если hostname/адреса не совпадают — импорт не завершился либо не сработал. Не считайте наличие `.unl` доказательством применения конфигурации.
7. После выполнения сохраните конфигурацию: Nexus — `copy running-config startup-config`, IOSv — `write memory`.

Печатные методички: Lab 1 PDF, Lab 2 PDF. Ссылки и загрузки удобнее открывать из Markdown-версии.

После проверки старта переходите к общему underlay с командами и объяснениями, затем к пошаговому заданию нужной работы. Оба задания начинаются со своего initial-стенда; готовую Lab 1 не следует использовать как скрытую основу Lab 2.

`Mgmt0` на Nexus — NIC 0, не подключён. `Ethernet1/1` — NIC 1; ошибочная нумерация на один порт нарушает топологию. Интернет/Cloud/pnet0 не нужны.

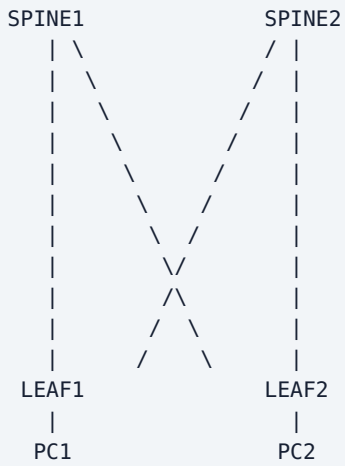
Стартовые и эталонные лаборатории имеют разные UUID. Эталоны помечены admin-only; проверка реальных прав веб-доступа требует отдельной непривилегированной учётной записи.

Захват

Используйте Capture на uplink Leaf и отдельно на access-порту. При ECMP один конкретный поток может выбрать любой Spine: отсутствие пакетов на одном uplink не доказывает отсутствие VXLAN. Захватите оба либо временно отключите второй uplink **в своей копии** и затем восстановите его.

Источник требований: Cisco Nexus 9000v Guide 10.5(x). Виртуальный dataplane не воспроизводит производительность и все аппаратные функции физического Nexus.

Общая топология и адресация



Каждый Leaf связан с обоими Spine. **Физического линка Leaf–Leaf и Spine–Spine нет.** Точные связи — в таблице:

Сегмент	Сторона A	IP A	Сторона B	IP B
S1-L1	SPINE1 E1/1	10.0.11.1/30	LEAF1 E1/1	10.0.11.2/30
S1-L2	SPINE1 E1/2	10.0.12.1/30	LEAF2 E1/1	10.0.12.2/30
S2-L1	SPINE2 E1/1	10.0.21.1/30	LEAF1 E1/2	10.0.21.2/30
S2-L2	SPINE2 E1/2	10.0.22.1/30	LEAF2 E1/2	10.0.22.2/30
ACCESS-L1	LEAF1 E1/3	access VLAN 10	PC1 Gi0/0	192.168.10.11/24
ACCESS-L2	LEAF2 E1/3	access VLAN 10	PC2 Gi0/0	192.168.10.12/24

Узел	Loopback0: router-id/BGP	Loopback1: VTEP
SPINE1	10.255.0.1/32	—
SPINE2	10.255.0.2/32	—
LEAF1	10.255.0.11/32	10.255.1.11/32

Узел	Loopback0: router-id/BGP	Loopback1: VTEP
LEAF2	10.255.0.12/32	10.255.1.12/32

- Underlay: OSPF `UNDERLAY`, area `0.0.0.0`, point-to-point на routed uplink, MTU 9216.
- Overlay: VLAN **10** ↔ L2VNI **10010**, NVE1 source `loopback1`.
- EVPN: iBGP AS **65000**, соседи по Loopback0; оба Spine — route reflector, оба Leaf — RR clients.
- Spine не имеют tenant VLAN, VNI или NVE. Передают outer IP и отражают EVPN-маршруты.
- Шлюз клиентам не требуется: они в одной подсети. Не настраивайте SVI или маршруты `192.168.10.0/24` в underlay вместо VXLAN.

Теория: VXLAN BGP EVPN

1. Зачем нужен EVPN, если VXLAN уже работает

В Lab 1 VTEP вручную знает удалённые replication peers и изучает remote MAC из поступающего VXLAN-трафика. При росте fabric поддерживать такие списки на всех Leaf неудобно. Кроме того, оператору нужна распределяемая информация: какой VTEP обслуживает MAC и какие VTEP входят в конкретный VNI.

EVPN (Ethernet VPN) — семейство BGP-маршрутов для распространения такой информации. В этой работе транспортом клиентских кадров остаётся VXLAN. EVPN не заменяет UDP-инкапсуляцию и не переносит каждый пользовательский кадр внутри BGP UPDATE.

Уровень	Задача	Протокол/объект
Underlay control plane	Узнать маршруты до инфраструктурных IP	OSPF
Overlay control plane	Распространить MAC и членство VTEP в VNI	BGP EVPN, TCP/179
Overlay data plane	Перенести клиентские кадры	VXLAN, UDP/4789

Established BGP не гарантирует работоспособный VXLAN. BGP может быть установлен, а маршрут не импортирован из-за RT; маршрут может быть импортирован, а access VLAN или MTU — настроен неверно.

2. Два loopback — две разные роли

На Leaf адреса разделены намеренно:

Роль	LEAF1	LEAF2
Loopback0, router-id и BGP update-source	10.255.0.11	10.255.0.12
Loopback1, NVE source/VTEP	10.255.1.11	10.255.1.12

BGP-соседи LEAF1 — Loopback0 обоих Spine: `10.255.0.1` и `.2`. Но **EVPN next-hop** remote MAC PC2 — `10.255.1.12`, VTEP LEAF2. Именно на этот адрес LEAF1 отправляет VXLAN-пакет.

Отсюда следуют две проверки: OSPF должен обеспечивать достижимость и BGP loopback, и VTEP loopback. Ping до RR не доказывает маршрут до удалённого NVE source.

3. Какие EVPN-маршруты нужны в этой работе

BGP EVPN передаёт NLRI разных типов. Мы используем два из них:

Route type	Название	Что сообщает
2	MAC/IP Advertisement	«Этот MAC, и при наличии IP binding — этот IP, достижим за данным VTEP»
3	Inclusive Multicast Ethernet Tag, IMET	«Этот VTEP участвует в broadcast domain; включите его в доставку BUM»

Type 2: доставка известному MAC

LEAF2 изучает MAC PC2 на E1/3 и объявляет его через EVPN. После импорта маршрута LEAF1 знает, что кадр к этому MAC нужно отправить через NVE к `10.255.1.12`.

В нашей **L2-only** конфигурации type 2 может быть **MAC-only**: IP address length равен 0, а в подробном выводе присутствует `0.0.0.0`. Это не означает, что клиент получил адрес `0.0.0.0` или что EVPN сломан. IP binding/ARP suppression здесь не настраиваются и для переноса Ethernet не обязательны.

Локальный MAC не появляется из ничего. Пока PC2 не отправил кадр, LEAF2 может не иметь локальной записи для объявления. Поэтому type 2 следует искать **после генерации клиентского трафика**.

Type 3: кому копировать BUM

Для доставки первого ARP одного remote MAC-маршрута недостаточно: ARP — broadcast. Type 3 сообщает о членстве VTEP в VNI и содержит информацию, используемую для ingress replication. В нашем случае это unicast replication, а не multicast underlay.

Проверка только type 2 неполна: без корректной BUM-доставки новые клиенты или очищенные ARP-кэши могут обнаружить проблему, которую скрывал уже изученный unicast.

Другие EVPN-типы, например type 1/4 для multihoming и type 5 для IP prefix advertisement, не входят в эту работу. Их отсутствие здесь не является ошибкой.

4. RD, RT и VNI: три разных понятия

Объект	Для чего нужен	Пример на стенде
VNI	Идентификатор сегмента в VXLAN data plane	10010
RD, Route Distinguisher	Делает BGP NLRI уникальным, в том числе при одинаковых MAC в разных VPN	`10.255.0.11:32777`, `10.255.0.12:32777` в проверенном выводе

Объект	Для чего нужен	Пример на стенде
RT, Route Target	Определяет политику экспорта и импорта маршрутов через extended community	`65000:10010`

RD не выбирает, кто импортирует маршрут. Эту роль выполняет RT. На разных Leaf RD должны обеспечивать уникальность, а совместимые import/export RT позволяют включить маршруты в один логический сегмент.

`rd auto` и `route-target ... auto` вычисляют значения автоматически. В нашей одно-AS схеме с одинаковым VNI результатом является согласованный RT `65000:10010`. Не переносите это без проверки в multi-AS-дизайн: автоматическое вычисление RT зависит от платформы и схемы AS/VNI. Число `32777` в RD — наблюдение этой VLAN-based конфигурации, не обязательная константа любого EVPN.

RT передаётся как **extended community**. Поэтому у BGP-соседа нужна передача extended communities; команда `send-community both` включает обычные и extended communities. Потеря RT может нарушить импорт, даже если сессия установлена и NLRI виден у RR.

5. Зачем два route reflector

В обычном iBGP маршрут, полученный от одного iBGP-соседа, не передаётся другому iBGP-соседу. Без дополнительного механизма потребовался бы full mesh между Leaf.

Route reflector (RR) снимает это ограничение для настроенных клиентов. SPINE1 и SPINE2 получают EVPN-маршруты Leaf и отражают их другим Leaf. На каждом Spine оба Leaf помечены `route-reflector-client` в EVPN address-family.

Spine не имеют локальных tenant VNI, но должны сохранять и отражать маршруты соответствующих RT. Для этого в данном NX-OS дизайне у RR задано `retain route-target all` в глобальном `address-family l2vpn evpn`.

RR не обязан быть VTEP. На наших Spine нет `interface nve1`, VLAN 10 и client-facing портов. Не настраивайте `next-hop-self` для EVPN «по привычке»: remote next-hop должен оставаться адресом Leaf VTEP. Отражатель распространяет информацию, но не становится точкой декапсуляции клиентского кадра.

Два RR дают два источника control-plane информации. Здесь каждый Leaf имеет две iBGP-сессии, но между Spine отдельной BGP-сессии нет. Не следует распространять эту минимальную топологию на любое количество Leaf/Spine без отдельного проектирования.

6. Как формируется доставка от PC1 к PC2

1. OSPF обеспечивает маршруты до Loopback0 и Loopback1 всех нужных узлов.
2. Leaf устанавливают EVPN BGP-сессии с двумя RR по Loopback0.
3. Активный VNI объявляет type 3. После импорта remote type 3 формируется список VTEP для BUM.
4. ARP PC1 передаётся через VXLAN к участникам VNI; его клиентский broadcast остаётся внутри unicast VXLAN-пакета.
5. Каждый Leaf изучает MAC локального клиента и объявляет type 2 через RR.
6. LEAF1 импортирует type 2 PC2 с подходящим RT и устанавливает remote MAC → `10.255.1.12`.

7. Известный unicast отправляется напрямую к VTEP LEAF2 через выбранный underlay-путь. BGP не участвует в обработке каждого ICMP-пакета.

На LEAF1 цепочку можно проследить командами `show bgp l2vpn evpn route-type 2` → `show l2route evpn mac all` → `show mac address-table vlan 10`. В проверенной конфигурации remote запись имеет producer **BGP**, а MAC-таблица указывает `nve1(10.255.1.12)`.

7. Что меняется по сравнению со статическим вариантом

Свойство	Lab 1: static	Lab 2: EVPN
Список remote VTEP для BUM	Ручные `peer-ip`	По EVPN type 3
Изучение remote MAC	Из data plane	По EVPN type 2
Изучение local MAC	На access-порту	На access-порту
NVE host-reachability	По умолчанию flood-and-learn	`protocol bgp`
Ingress replication	`protocol static`	`protocol bgp`
Отображение NVE на стенде	DP, UnicastStatic	CP, UnicastBGP
Перенос клиентских кадров	VXLAN UDP/4789	VXLAN UDP/4789

Переключение `ingress-replication protocol static` на `bgp` без настройки BGP/EVPN/RD/RT не создаёт рабочий control plane. Поэтому в методичке EVPN настраивается последовательно, а не одной командой.

8. Резервирование и время сходимости

Отказ одного uplink Leaf проверяет OSPF/ECMP. BGP-соседство с RR1 может остаться доступным обходным путём: LEAF1 → SPINE2 → LEAF2 → SPINE1. Поэтому потеря прямого линка к Spine не равна потере route reflector.

Чтобы действительно изолировать RR1 в нашей схеме, выключают оба его uplink. После истечения BGP timers сессия с RR1 перестаёт быть установленной, а с RR2 остаётся. Remote EVPN-маршруты и клиентский трафик продолжают обслуживаться через оставшийся путь.

Сходимость не мгновенна. Допускаются потери во время смены маршрутов; проверяют восстановившуюся передачу и фактические состояния, а не обещают zero-loss. Таймеры в работе не сокращаются искусственно ради красивого результата.

9. Чтение вывода и типичные ошибки

- В `show bgp l2vpn evpn summary` число в State/PfxRcd означает установленную сессию с данным количеством принятых префиксов. Слово `Established` может не выводиться.
- Сессия с RR и импорт маршрута — разные этапы. Проверяйте RT, next-hop, наличие записи в L2RIB и MAC-таблице.
- Статус VNI Up и peer Up — необходимые, но не достаточные признаки. Нужны ping и корректные VXLAN-пакеты.

- `is not in HW` в подробном выводе виртуального Nexus не является самостоятельным доказательством отказа: на стенде при таком выводе dataplane был проверен реальными пакетами.
- На NX-OS 10.5(2) для EVPN address-family на **Spine тоже нужен `nv overlay evpn`**, хотя NVE на нём не создаётся. Это включение EVPN control plane, а не назначение Spine роли VTEP.

10. Границы практикума и вопросы

Мы используем L2VNI. L3VNI, symmetric IRB, anycast gateway, EVPN multihoming/vPC и Multi-Site здесь не настраиваются. Нельзя заявлять их работоспособность на основании одного L2 ping. Как и статический VXLAN, VXLAN EVPN сам по себе не шифрует трафик.

Перед практикой объясните:

1. Почему TCP/179 в Capture не является доказательством клиентского VXLAN?
2. Чем RD отличается от RT, и почему RD у Leaf может различаться?
3. Что будет, если MAC-маршрут виден у RR, но его RT не импортируется на Leaf?
4. Почему next-hop remote MAC — Loopback1 Leaf, а BGP neighbor — Loopback0 Spine?
5. Зачем type 3, если уже есть type 2?

Далее: underlay и ход Lab 2 с объяснением команд.

Источники

- RFC 7432 — BGP MPLS-Based Ethernet VPN — базовые EVPN механизмы и типы маршрутов; наша инкапсуляция — VXLAN, не MPLS.
- RFC 8365 — Network Virtualization Overlay Solution Using EVPN.
- Cisco Nexus 9000 NX-OS VXLAN Configuration Guide, Release 10.5(x).

Общий этап: underlay и OSPF пошагово

Эту последовательность выполняют **перед настройкой VXLAN в каждой работе**. В Lab 2 используйте отдельную `evpn_initial.unl`, а не стенд с оставшимися static peers из Lab 1. Полная адресация одинакова.

1. Проверьте, что стартовые конфигурации применились

Стартовый `.unl` уже задаёт hostname, учётную запись, IP loopback, IP/MTU routed uplink и адреса PC. OSPF, VLAN 10/VNI 10010, NVE и BGP студент добавляет сам. Примеры ниже повторяют IP/MTU намеренно: их можно сверить со стартом и восстановить правильные значения, не угадывая контекст.

На каждом Nexus:

```
show version
show interface brief
show running-config
```

На PC1 и PC2:

```
show ip interface brief
```

Ожидается C9300v с NX-OS 10.5(2), правильные имена узлов, PC1 `192.168.10.11/24`, PC2 `192.168.10.12/24`. У Nexus `Mgmt0` не подключён; рабочие порты начинаются с E1/1. Если вместо нужного имени видно `switch`, дождитесь окончания bootstrap и проверьте импорт, а не продолжайте на случайной конфигурации.

2. Разберитесь с режимами CLI

Команда	Что меняет
`configure terminal`	Переходит из EXEC LEAF1# в глобальную конфигурацию
`feature ospf`	Включает функцию OSPF в модульном NX-OS CLI
`router ospf UNDERLAY`	Создаёт/открывает процесс OSPF; UNDERLAY — локальное имя процесса
`router-id ...`	Задаёт стабильный уникальный идентификатор OSPF-маршрутизатора
`interface ...`	Открывает конфигурацию конкретного интерфейса
`exit`	Возвращается на один уровень CLI выше
`end`	Возвращается из любого режима конфигурации в EXEC

Копируйте только содержимое блока для указанного устройства. Приглашения вида `LEAF1(config-if)#` вводить не нужно. Отступы помогают читать иерархию, но **режим CLI задают команды**, а не количество пробелов. Поэтому `exit` и `end` важны.

3. Настройте SPINE1

```
configure terminal
feature ospf
router ospf UNDERLAY
  router-id 10.255.0.1
exit
interface loopback0
  ip address 10.255.0.1/32
  ip router ospf UNDERLAY area 0.0.0.0
exit
interface Ethernet1/1
  no switchport
  mtu 9216
  ip address 10.0.11.1/30
  ip ospf network point-to-point
  ip router ospf UNDERLAY area 0.0.0.0
  no shutdown
exit
interface Ethernet1/2
  no switchport
  mtu 9216
  ip address 10.0.12.1/30
  ip ospf network point-to-point
  ip router ospf UNDERLAY area 0.0.0.0
  no shutdown
end
```

Объяснение интерфейсных команд, общее для всех четырёх Nexus:

Команда	**Зачем нужна**
`interface loopback0`	Создаёт логический интерфейс, не зависящий от состояния одного uplink
`ip address .../32`	Задаёт адрес одного инфраструктурного узла; /32 — host prefix
`no switchport`	Делает Ethernet-порт маршрутизируемым, а не access/trunk
`mtu 9216`	Даёт underlay место для клиентского кадра и VXLAN overhead
`ip address .../30`	Задаёт адрес физического L3-линка; адреса на концах должны быть из одной /30
`ip ospf network point-to-point`	Убирает выбор DR/BDR на двухточечном линке; задаётся согласованно на обоих концах
`ip router ospf UNDERLAY area 0.0.0.0`	Включает интерфейс в процесс OSPF и area 0; анонсирует его сеть
`no shutdown`	Снимает административное отключение интерфейса

Spine не получает Loopback1: в этой схеме он не VTEP.

4. Настройте SPINE2

```
configure terminal
feature ospf
router ospf UNDERLAY
  router-id 10.255.0.2
exit
interface loopback0
  ip address 10.255.0.2/32
  ip router ospf UNDERLAY area 0.0.0.0
exit
interface Ethernet1/1
  no switchport
  mtu 9216
  ip address 10.0.21.1/30
  ip ospf network point-to-point
  ip router ospf UNDERLAY area 0.0.0.0
  no shutdown
exit
interface Ethernet1/2
  no switchport
  mtu 9216
  ip address 10.0.22.1/30
  ip ospf network point-to-point
  ip router ospf UNDERLAY area 0.0.0.0
  no shutdown
end
```

От SPINE1 отличаются router-id, Loopback0 и адреса двух линков. Не копируйте `.1` Loopback0 на оба Spine: идентификаторы должны быть уникальны.

5. Настройте LEAF1

```
configure terminal
feature ospf
router ospf UNDERLAY
  router-id 10.255.0.11
exit
interface loopback0
  ip address 10.255.0.11/32
  ip router ospf UNDERLAY area 0.0.0.0
exit
interface loopback1
  ip address 10.255.1.11/32
  ip router ospf UNDERLAY area 0.0.0.0
exit
interface Ethernet1/1
  no switchport
  mtu 9216
  ip address 10.0.11.2/30
  ip ospf network point-to-point
  ip router ospf UNDERLAY area 0.0.0.0
  no shutdown
exit
interface Ethernet1/2
  no switchport
  mtu 9216
  ip address 10.0.21.2/30
  ip ospf network point-to-point
  ip router ospf UNDERLAY area 0.0.0.0
  no shutdown
end
```

Loopback0 используется как router-id и позже BGP source. **Loopback1 — отдельный VTEP source**. Оба должны быть известны underlay: без маршрута к Loopback1 клиентский VXLAN не заработает, даже если BGP по Loopback0 установлен.

6. Настройте LEAF2

```
configure terminal
feature ospf
router ospf UNDERLAY
  router-id 10.255.0.12
exit
interface loopback0
  ip address 10.255.0.12/32
  ip router ospf UNDERLAY area 0.0.0.0
exit
interface loopback1
  ip address 10.255.1.12/32
  ip router ospf UNDERLAY area 0.0.0.0
exit
interface Ethernet1/1
  no switchport
  mtu 9216
  ip address 10.0.12.2/30
  ip ospf network point-to-point
  ip router ospf UNDERLAY area 0.0.0.0
  no shutdown
exit
interface Ethernet1/2
  no switchport
  mtu 9216
  ip address 10.0.22.2/30
  ip ospf network point-to-point
  ip router ospf UNDERLAY area 0.0.0.0
  no shutdown
end
```

E1/3 обоих Leaf пока не трогаем: его access VLAN будет настроен в конкретной работе. Не превращайте uplink E1/1 и E1/2 в trunk.

7. Контрольная точка: OSPF и ECMP

На всех четырёх Nexus:

```
show ip ospf neighbors
show ip route ospf
```

Должно быть **два соседа FULL** на каждом устройстве. `FULL/ -` нормально для point-to-point: выбор DR/BDR не выполняется. Кратковременные INIT/EXSTART во время сходимости допустимы, но не являются готовым результатом.

На LEAF1:

```
show ip route 10.255.1.12/32
ping 10.255.1.12 source 10.255.1.11 count 5
```

На LEAF2:

```
show ip route 10.255.1.11/32
ping 10.255.1.11 source 10.255.1.12 count 5
```

`source` выбирает локальный адрес-источник — в данном случае будущий VTEP. `count 5` ограничивает число echo request. В NX-OS это синтаксис `count`, а на IOSv-клиентах далее используется `repeat`.

При обоих исправных путях LEAF1 должен иметь next-hop **10.0.11.1 и 10.0.21.1** до remote VTEP; LEAF2 — **10.0.12.1 и 10.0.22.1**. В проверенном выводе это `ubest/mbest: 2/0`. Точная стоимость маршрута зависит от reference bandwidth и параметров образа; важны достижимость и два равностоимостных пути.

Если FULL нет, сначала проверьте адреса/маски, administrative state, area, MTU и network type на обоих концах. Если соседство есть, но remote Loopback1 отсутствует, проверьте включение именно этого loopback в OSPF. **Не переходите к NVE при неработающем underlay.**

8. Сохраните этот этап и продолжите свою работу

На каждом Nexus:

```
copy running-config startup-config
```

`running-config` — текущее состояние в памяти, `startup-config` — сохранённая конфигурация для следующего запуска. Выполненная команда настройки без сохранения и без проверки состояния ещё не является завершённой работой.

- Продолжить Lab 1: VLAN/VNI и статический NVE.
- Продолжить Lab 2: RR и BGP EVPN.

Lab 2. VXLAN BGP EVPN

Время: ориентировочно 120–150 минут, без учёта загрузки. Рекомендуется предварительно выполнить Lab 1. Ниже приведён последовательный ход работы с командами и объяснением их назначения.

Материалы: подготовка, адресация, теория, проверки, Wireshark, стартовый ZIP.

Цель

Заменить статические remote peers на EVPN control plane. Подтвердить двусторонний клиентский трафик, remote MAC через BGP и работу при отказе одного Spine-пути.

Шаг 1. Подготовьте чистый старт и underlay

Создайте отдельную персональную копию `evpn_initial.unl` с новым UUID по инструкции подготовки. Не используйте активную Lab 1 со статическим `peer-ip`. Прочитайте теорию EVPN, затем выполните общую пошаговую настройку underlay на всех четырёх Nexus.

Контрольная точка: по два OSPF FULL на каждом Nexus; LEAF1/2 достигают `10.255.0.1`, `.2` и remote Loopback1; до remote VTEP есть два next-hop, uplink routed, MTU 9216. Не продолжайте при неработающем underlay.

Шаг 2. Настройте route reflector SPINE1

```
configure terminal
feature bgp
nv overlay evpn
router bgp 65000
  router-id 10.255.0.1
  address-family l2vpn evpn
    retain route-target all
  exit
  neighbor 10.255.0.11
    remote-as 65000
    update-source loopback0
    address-family l2vpn evpn
      send-community both
      route-reflector-client
    exit
  exit
  neighbor 10.255.0.12
    remote-as 65000
    update-source loopback0
    address-family l2vpn evpn
      send-community both
      route-reflector-client
    exit
  exit
end
```

Команда	Назначение
`feature bgp`	Включает BGP в NX-OS
`nv overlay evpn`	Включает EVPN control plane; в NX-OS 10.5(2) нужна и на RR без NVE
`router bgp 65000`	Создаёт BGP process в AS 65000; все четыре узла здесь в одной AS, то есть используется iBGP
`router-id ...`	Задаёт уникальный BGP identifier, в работе равный Loopback0
Глобальный `address-family l2vpn evpn`	Включает EVPN address-family процесса, не обычную IPv4 unicast AF
`retain route-target all`	Сохраняет EVPN-маршруты RT, для которых у RR нет локального tenant VNI
`neighbor ...`	Открывает конфигурацию конкретного удалённого BGP-соседа — Loopback0 Leaf
`remote-as 65000`	Задаёт AS соседа; равенство local/remote AS делает сессию iBGP
`update-source loopback0`	Выбирает локальный адрес-источник BGP TCP-сессии; оба конца должны ожидать именно loopback друг друга
`address-family l2vpn evpn` у соседа	Активирует обмен EVPN NLRI с данным соседом

Команда	Назначение
`send-community both`	Передаёт standard и extended communities, в том числе RT
`route-reflector-client`	Помечает соседа как RR client **в этой address-family**; позволяет отражать iBGP-маршруты

Не создавайте на Spine NVE/VNI и не добавляйте `next-hop-self`. EVPN next-hop должен указывать на VTEP Leaf, а не на route reflector.

Шаг 3. Настройте второй route reflector SPINE2

```
configure terminal
feature bgp
nv overlay evpn
router bgp 65000
  router-id 10.255.0.2
  address-family l2vpn evpn
    retain route-target all
  exit
  neighbor 10.255.0.11
    remote-as 65000
    update-source loopback0
    address-family l2vpn evpn
      send-community both
      route-reflector-client
    exit
  exit
  neighbor 10.255.0.12
    remote-as 65000
    update-source loopback0
    address-family l2vpn evpn
      send-community both
      route-reflector-client
    exit
  exit
end
```

Соседи — те же два Leaf, но локальные router-id и update-source IP принадлежат SPINE2. Отдельную BGP-сессию SPINE1 ↔ SPINE2 в этой минимальной топологии не добавляем.

Шаг 4. Включите EVPN и iBGP на Leaf

На **LEAF1**:

```
configure terminal
feature bgp
feature nv overlay
feature vn-segment-vlan-based
nv overlay evpn
router bgp 65000
  router-id 10.255.0.11
  address-family l2vpn evpn
  exit
  neighbor 10.255.0.1
    remote-as 65000
    update-source loopback0
    address-family l2vpn evpn
      send-community both
    exit
  exit
  neighbor 10.255.0.2
    remote-as 65000
    update-source loopback0
    address-family l2vpn evpn
      send-community both
    exit
  exit
end
```

На **LEAF2**:

```
configure terminal
feature bgp
feature nv overlay
feature vn-segment-vlan-based
nv overlay evpn
router bgp 65000
  router-id 10.255.0.12
  address-family l2vpn evpn
  exit
  neighbor 10.255.0.1
    remote-as 65000
    update-source loopback0
    address-family l2vpn evpn
      send-community both
    exit
  exit
  neighbor 10.255.0.2
    remote-as 65000
    update-source loopback0
    address-family l2vpn evpn
      send-community both
    exit
  exit
end
```

`feature nv overlay` разрешает NVE, `feature vn-segment-vlan-based` — связь VLAN/VNI, `nv overlay evpn` — EVPN control plane. Это разные функции. На Leaf не задаём `route-reflector-client`: роль RR выполняют Spine.

Проверьте **на всех четырёх Nexus**:

```
show bgp l2vpn evpn summary
```

Контрольная точка: у Leaf два соседа `.1` и `.2`, у Spine — `.11` и `.12`. На этом этапе установленная сессия может показывать **0** в State/PfxRcd: VNI ещё не настроен, маршрутов нет. `Idle`, `Active`, `Connect` — не установленная сессия. При ошибке сначала проверьте IP-достижимость Loopback0, `remote-as`, `update-source`, `nv overlay evpn` и AF у соседа.

Шаг 5. Создайте локальный VLAN/VNI и правила импорта EVPN

Выполните одинаковый блок **на обоих Leaf**:

```
configure terminal
vlan 10
  name TENANT-A
  vn-segment 10010
exit
interface Ethernet1/3
  switchport
  switchport mode access
  switchport access vlan 10
  spanning-tree port type edge
  no shutdown
exit
evpn
  vni 10010 l2
    rd auto
    route-target import auto
    route-target export auto
end
```

VLAN/access-команды разобраны в Lab 1. Здесь их смысл тот же: локальный клиент попадает в VLAN 10, который отображён в VNI 10010.

Команда	Объяснение
`evpn`	Открывает глобальную конфигурацию EVPN instances
`vni 10010 l2`	Описывает L2 EVPN instance для этого VNI; это не L3VNI/VRF
`rd auto`	Автоматически формирует RD для уникальности NLRI
`route-target import auto`	Определяет, какие RT импортировать в instance
`route-target export auto`	Назначает RT локально объявляемым EVPN-маршрутам

В этой одно-AS схеме оба Leaf используют RT `65000:10010`; RD различаются. Не заменяйте RD на RT и не считайте их одним идентификатором.

Шаг 6. Подключите VNI к NVE

Одинаково **на LEAF1 и LEAF2**:

```
configure terminal
interface nve1
  no shutdown
  source-interface loopback1
  host-reachability protocol bgp
  member vni 10010
    ingress-replication protocol bgp
end
```

Команда	Объяснение
`interface nve1` / `no shutdown`	Создаёт и включает VXLAN-интерфейс
`source-interface loopback1`	Выбирает local VTEP `.1.11` или `.1.12`, а не BGP Loopback0
`host-reachability protocol bgp`	Использует BGP EVPN для удалённой информации о конечных узлах
`member vni 10010`	Включает конкретный VNI в NVE
`ingress-replication protocol bgp`	Строит BUM replication list по EVPN, без ручного peer-IP

Проверьте:

```
show nve vni
show nve peers
show bgp l2vpn evpn route-type 3
```

Контрольная точка: VNI 10010 Up, CP, UnicastBGP, L2 VLAN 10; remote VTEP — Loopback1 другого Leaf, Up/CP. Найдите type 3 remote VTEP и RT. Если BGP установлен, но peer не появился, проверьте VNI, RD/RT, импорт и маршрут до remote Loopback1.

В этой работе не должно быть статического `peer-ip`. Если по ошибке начали со стенда Lab 1, не смешивайте два режима: предпочтительно вернитесь к отдельной initial-копии Lab 2. Простое добавление команд поверх static NVE не является проверенным переходом.

Шаг 7. Изучите MAC и докажите реальную передачу

Включите Capture на обоих uplink Leaf. **На PC1**:

```
ping 192.168.10.12 repeat 5
show arp
ping 192.168.10.12 size 1500 df-bit repeat 5
```

На PC2:

```
ping 192.168.10.11 repeat 5
show arp
```

`repeat` — IOSv-синтаксис числа запросов; `size 1500 df-bit` проверяет большой пакет без фрагментации. Повторные ping должны давать 5/5 в обе стороны. Первый может потерять пакет при изучении ARP/MAC.

Теперь **на обоих Leaf**:

```
show bgp l2vpn evpn route-type 2
show bgp l2vpn evpn route-type 3
show l2route evpn mac all
show mac address-table vlan 10
show nve peers
```

Найдите цепочку доказательств:

1. Type 2 удалённого клиентского MAC с RT `65000:10010`, label/VNI 10010 и next-hop remote **Loopback1**.
2. Тот же MAC в L2RIB с producer **BGP**.
3. Тот же MAC в MAC-таблице через `nve1(<remote-VTEP>)`, а local MAC — через E1/3.
4. В PCAP — **UDP/4789, VNI 10010**, outer VTEP IP и inner IP клиентов; для DF-теста — inner length 1500.

Type 2 до клиентского трафика может отсутствовать. MAC-only type 2 допустим: IP binding/ARP suppression не настраивались. Не требуйте `is in HW` от виртуального Nexus вместо фактической передачи.

Для анализа BGP используйте `tcp.port == 179`; для VXLAN — `udp.port == 4789 && vxlan.vni == 10010`. Keepalive не доказывает наличие UPDATE, а OSPF не доказывает EVPN. Подробный разбор — в заданиях Wireshark.

Шаг 8. Проверьте отказ одного uplink

Только в своей копии **на LEAF1**:

```
configure terminal
interface Ethernet1/1
  shutdown
end
show ip ospf neighbors
show ip route 10.255.1.12/32
show bgp l2vpn evpn summary
```

После OSPF-сходимости клиентский ping и DF 1500 должны проходить через E1/2. BGP к SPINE1 по loopback может **остаться установленным через SPINE2 и LEAF2** — это отказ пути, а не доказанная потеря RR.

Восстановите LEAF1:

```
configure terminal
interface Ethernet1/1
  no shutdown
end
show ip ospf neighbors
```

Дождитесь двух FULL перед следующим опытом.

Шаг 9. Изолируйте RR1 и восстановите его

На **SPINE1**, только в своей копии:

```
configure terminal
interface Ethernet1/1
  shutdown
exit
interface Ethernet1/2
  shutdown
end
```

Теперь RR1 действительно изолирован от обоих Leaf. На LEAF1 и LEAF2 наблюдайте `show bgp l2vpn evpn summary` до истечения таймеров: сессия с `10.255.0.1` перестаёт быть установленной, с `10.255.0.2` остаётся. Проверьте remote MAC, клиентский ping и DF 1500. В Capture трафик идёт через оставшийся Spine.

Затем **обязательно восстановите SPINE1**:

```
configure terminal
interface Ethernet1/1
  no shutdown
exit
interface Ethernet1/2
  no shutdown
end
show ip ospf neighbors
```

Дождитесь восстановления обоих BGP-соседей на обоих Leaf и повторите ping. Не заявляйте мгновенную или zero-loss сходимость: сохраняйте реальные состояния до, во время и после отказа.

Дополнительный опыт с RT выполняйте по согласованию с преподавателем: замените import RT на одном Leaf, **не оставляя правильный auto RT параллельно неправильному**. Объясните, почему сессия BGP остаётся установленной, а нужный remote MAC не импортируется. После опыта верните исходный import RT и проверьте доставку. Этот негативный опыт не входит в автоматическую сетевую приёмку практикума.

Шаг 10. Сохраните и оформите результат

На всех Nexus после восстановления:

```
copy running-config startup-config
```

Если меняли конфигурацию PC, сохраните её на IOSv через `write memory`. По чек-листу проверки убедитесь, что не осталось выключенных uplink, статических peers и ошибочных RT. В отчёте объясните, что сделал OSPF, что распространил BGP и какие кадры перенёс VXLAN.

Отчёт и оценка

Underlay/ECMP — 15%; RR и два EVPN-соседства — 20%; VNI/NVE и type 2/type 3 с next-hop/RT — 25%; двусторонний трафик и PCAP — 25%; отказ RR/восстановление и объяснения — 15%. Только конфиг без реальных выводов не подтверждает работоспособность. Не требуйте IP в type 2 там, где конфигурация даёт MAC-only advertisement.

Проверка Lab 2

На всех Nexus:

```
show ip ospf neighbors
show bgp l2vpn evpn summary
show bgp l2vpn evpn
```

На LEAF1 (на LEAF2 — обратные адреса):

```
show ip route 10.255.1.12/32
ping 10.255.1.12 source 10.255.1.11 count 5
show nve peers
show nve vni
show interface nve1
show bgp l2vpn evpn route-type 2
show bgp l2vpn evpn route-type 3
show l2route evpn mac all
show mac address-table vlan 10
```

Ожидается:

- OSPF: два FULL, два пути до remote VTEP при полностью исправной топологии.
- BGP: Leaf видит `.1` и `.2`, Spine видит `.11` и `.12`; AS 65000. Число в столбце State/PfxRcd — установленное соседство с принятыми префиксами, а не код ошибки.
- NVE peer: удалённый Loopback1, состояние Up, control-plane learning; VNI 10010 Up, L2, VLAN 10.
- EVPN type 3: членство remote VTEP. Type 2 после клиентского трафика: MAC удалённого клиента, next-hop `10.255.1.12` на LEAF1, не Loopback0 Spine.
- `show l2route evpn mac all`: remote MAC, producer BGP; таблица MAC указывает удалённую NVE-доставку.
- PC1 → `.12` и PC2 → `.11`: повторный `ping ... repeat 5` даёт 100%, 5/5.

Если peer/VNI Up, но ping не проходит, проверьте access VLAN/port, локальный MAC, импорт RT, соответствие VNI, путь и MTU до VTEP, а затем dataplane в PCAP. BGP Established не исключает ошибку RT.

У виртуального Nexus в подробном EVPN-выводе может встречаться `is not in HW`. Это не самостоятельный критерий отказа виртуального dataplane: проверяйте реальные ping, remote MAC и захват, не переносите аппаратные критерии ASIC на эмулятор.

После изоляции SPINE1: OSPF идёт через SPINE2; после истечения BGP timers сосед RR1 не установлен, RR2 установлен, повторный ping проходит. После восстановления оба соседа установлены. Допустима потеря пакетов во время сходимости; это не zero-loss fabric.

Wireshark: VXLAN EVPN

Снимайте underlay на обоих uplink Leaf. Для BGP UPDATE начните захват до установления сессий либо согласованно перезапустите **только собственную** BGP-сессию; постоянные keepalive не содержат новых маршрутов.

Фильтр	Назначение
`tcp.port == 179`	BGP между Loopback0
`bgp.type == 2`	BGP UPDATE
`udp.port == 4789 && vxlan.vni == 10010`	Tenant data plane
`ospf`	Underlay routing

1. В BGP OPEN найдите Multiprotocol capability AFI 25 (L2VPN), SAFI 70 (EVPN).
2. В UPDATE найдите type 3 IMET и type 2 MAC/IP Advertisement. Выпишите RD, RT, next-hop, VNI/label и MAC. Учитывайте, что MAC-only type 2 имеет IP address length 0.
3. Сравните IP BGP-соседа Spine с next-hop EVPN-маршрута Leaf: они выполняют разные роли.
4. Найдите VXLAN echo request/reply. Докажите, что Spine не декапсулирует overlay: IP destination — remote VTEP, а не Spine.
5. Сопоставьте один remote MAC из BGP UPDATE с `show l2route evpn mac all` и inner Ethernet header.
6. Во время отказа сохраните отдельный PCAP; объясните изменение физического пути при сохранении VNI и VTEP адресов.

Приложите PCAP, номера пакетов и выводы CLI с временем/точкой захвата. Не подменяйте UPDATE скриншотом BGP keepalive, а VXLAN — обычным underlay ping.