| Студент: | Бардадым С. О. | Допуск: |  |
| --- | --- | --- | --- |
| **Отчет по работе №3 (часть 1)**  Диагностика сети и настройка NAT | | | |
| Дата выполнения | 15.04.2025 | Дата защиты | 19.04.2025 |
| Оценка |  | Подпись |  |

Цель работы: познакомиться с технологиями NAT, протоколом DHCP, DNS.

Задачи работы: построить топологию сети, получить доступ в Интернет, развернуть DNS сервер, поднять веб-сервер, провести трассировку.

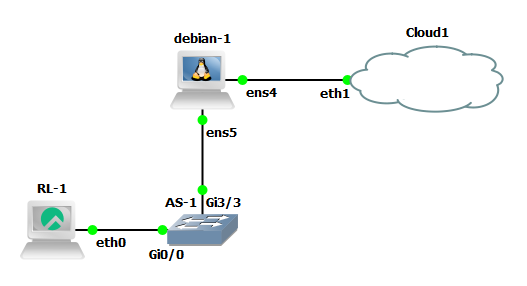
Краткий конспект теоретической части:

| NAT (Network Address Translation) – технология, применяемая в сетевых устройствах для преобразования локальных (частных) IP-адресов в глобальные (публичные) и наоборот. Это решение позволяет множеству устройств в частной сети совместно использовать один публичный IP-адрес для выхода в интернет, что особенно актуально при ограниченном количестве доступных глобальных IP-адресов.  Port Forwarding – это технология, которая позволяет обращаться из Интернет к компьютеру во внутренней сети за маршрутизатором, использующим NAT (NAPT).  Тройное рукопожатие TCP – это процесс, с помощью которого TCP устанавливает соединение между клиентом и сервером. Этот процесс состоит из трех этапов: SYN, SYN-ACK и ACK.  DNS – компьютерная распределённая система для получения информации о доменах.  DHCP Dynamic Host Configuration Protocol – это протокол клиент/сервер, который автоматически предоставляет хост Интернет-протокола (IP) с его IP-адресом и другой соответствующей информацией о конфигурации, такой как маска подсети и шлюз по умолчанию.  Firewall – программный или программно-аппаратный элемент компьютерной сети, осуществляющий контроль и фильтрацию проходящего через него сетевого трафика в соответствии с заданными правилами. |
| --- |

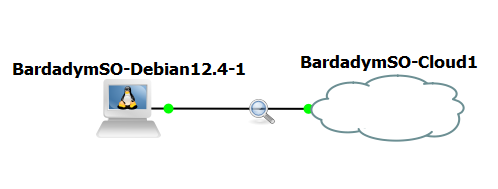
В этой и следующих работах следует придерживаться простых правил:

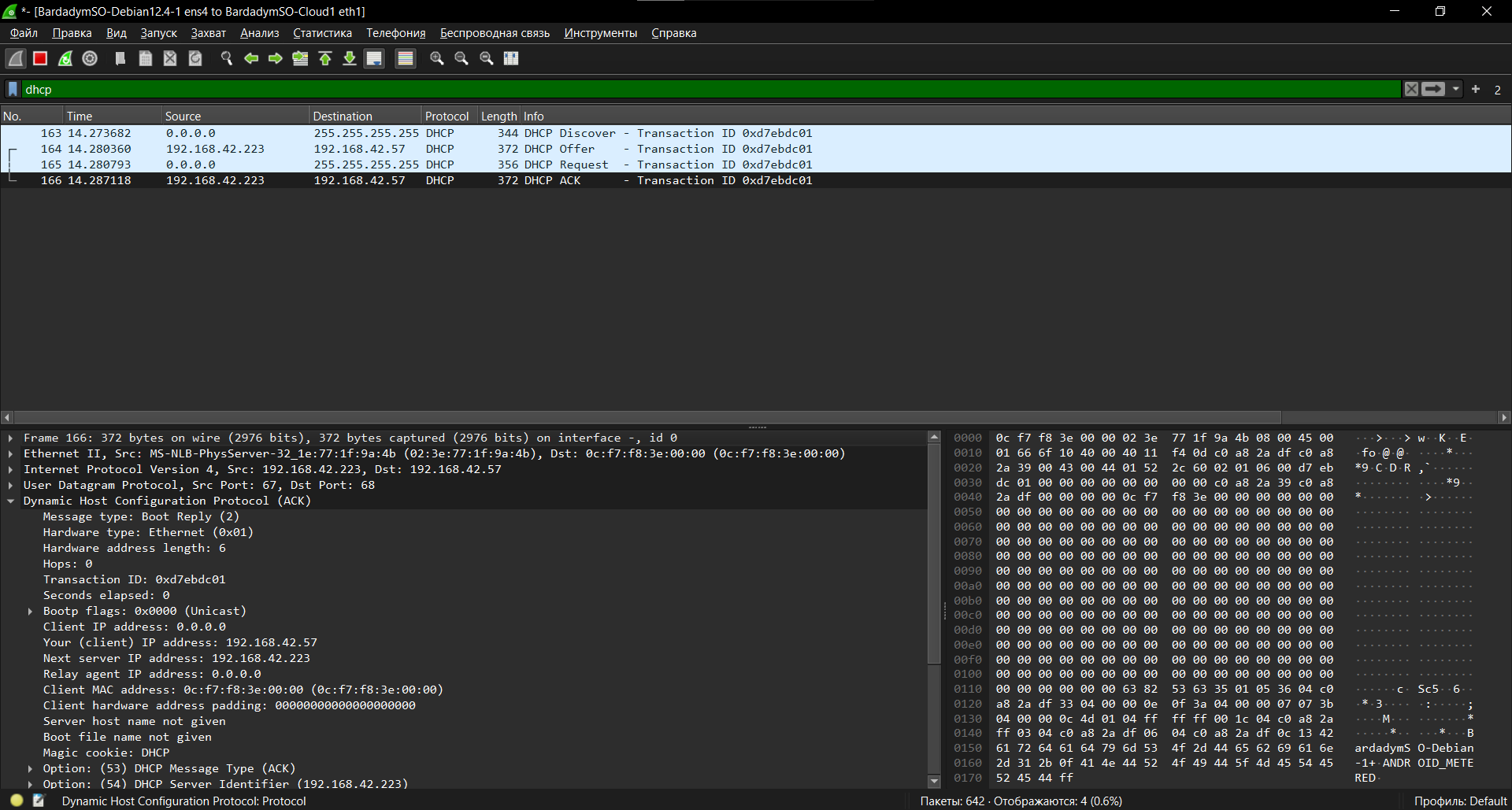
- hostname устройств должен содержать фамилию и инициалы студента, если не указано иное,

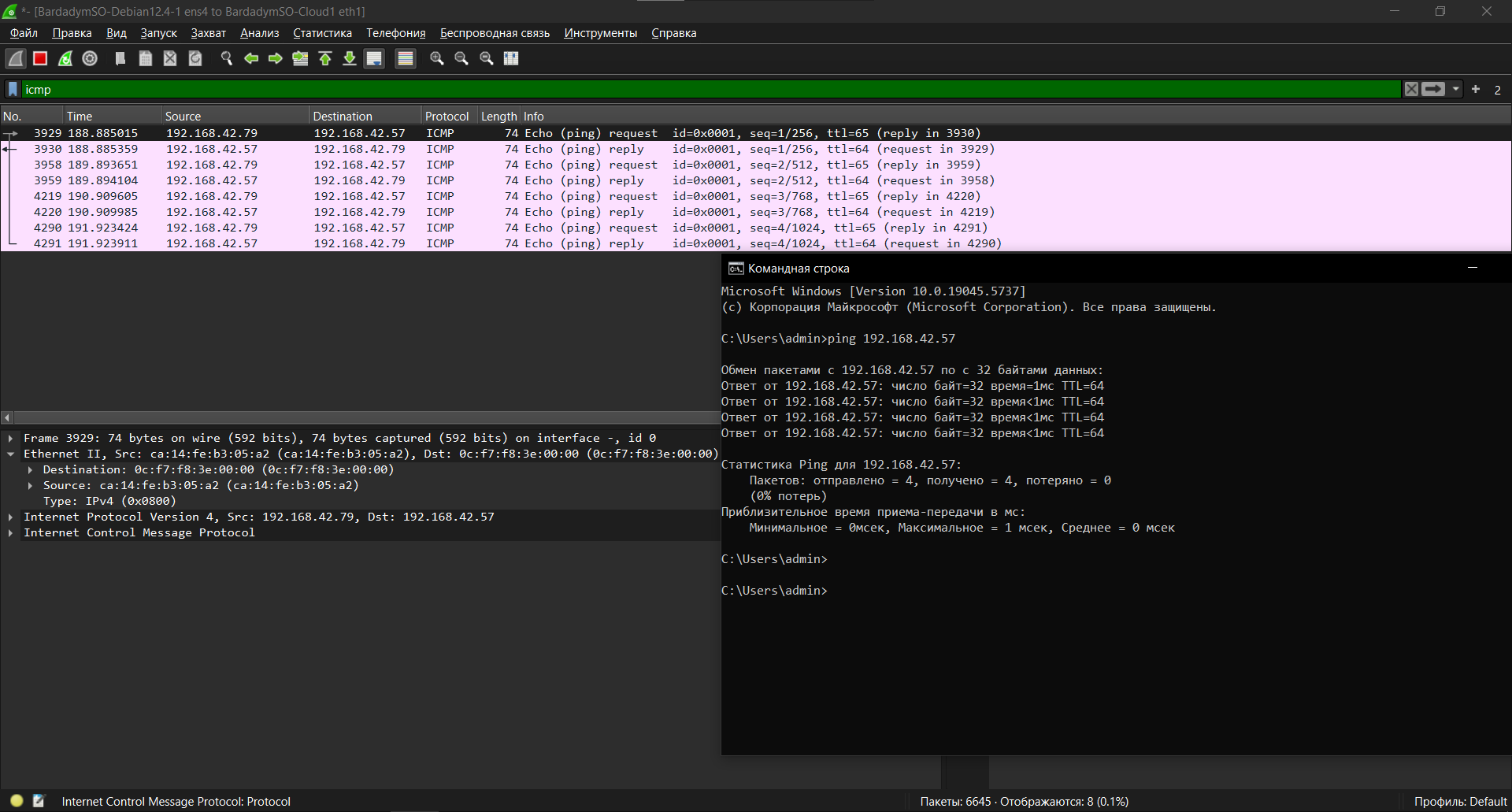
- вместо LN должен быть подставлен персональный Lucky Number из таблички

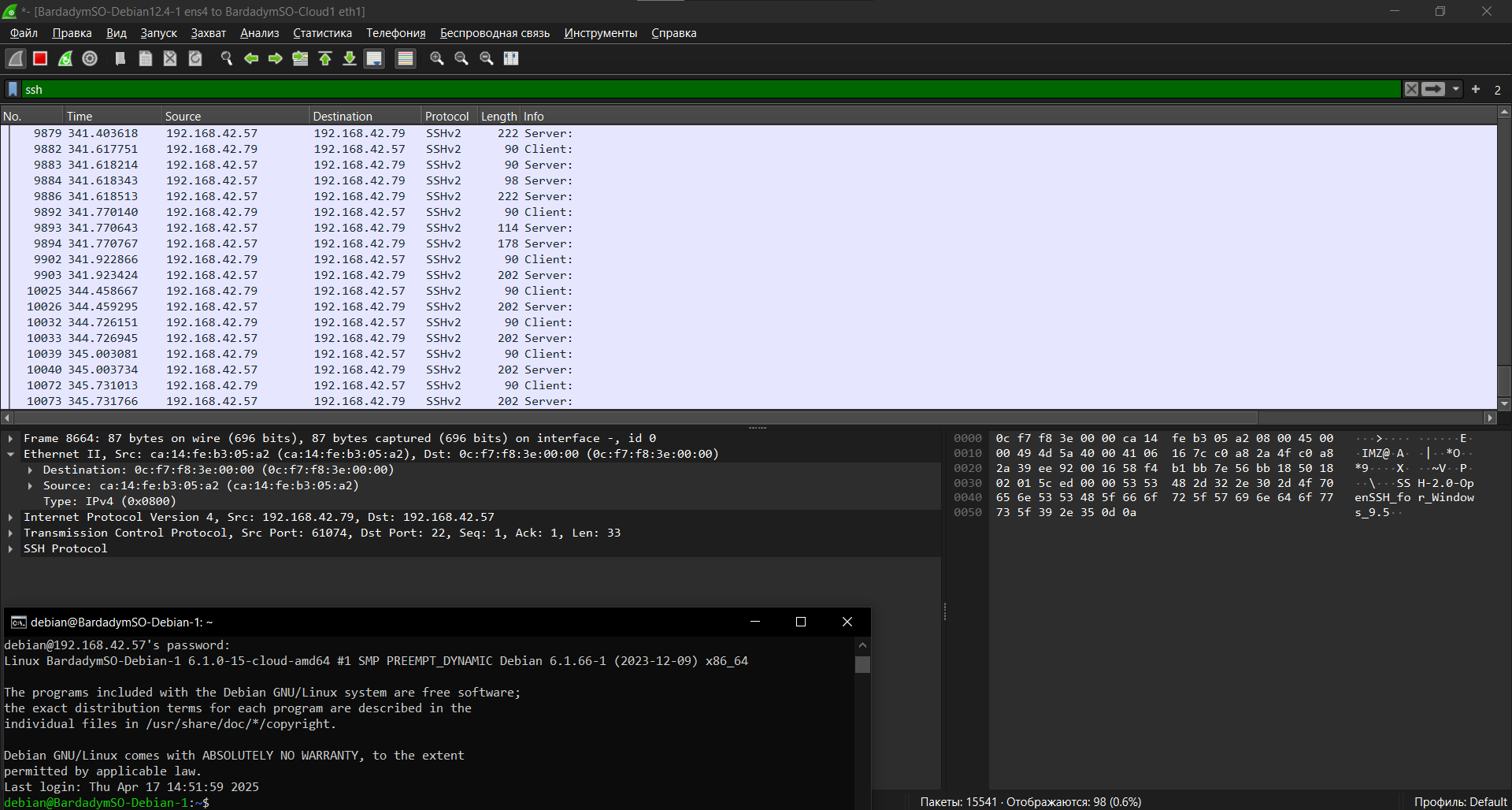


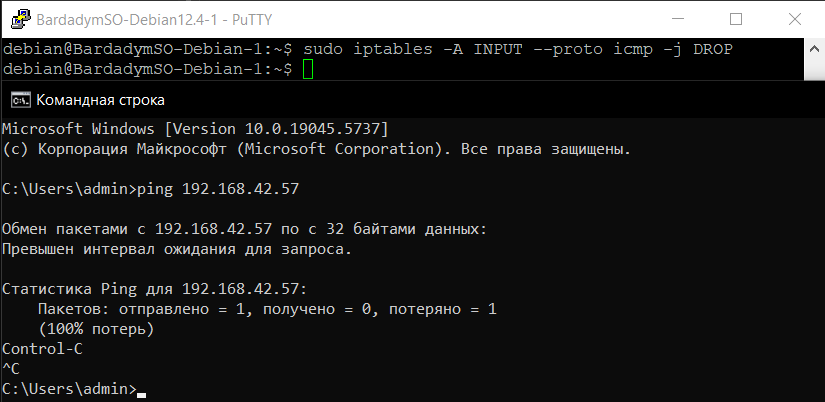
Задание 1. В GNS3 построить топологию из клиентского узла (ВМ debian) и облака, позволяющего тем или иным способом получить доступ в интернет. Показать процесс получения сетевых настроек по DHCP. Показать доступ с хоста к debian по протоколам ICMP и ssh. Написать правило iptables, отбрасывающее входящие ICMP-запросы на интерфейсе, продемонстрировать его работу.



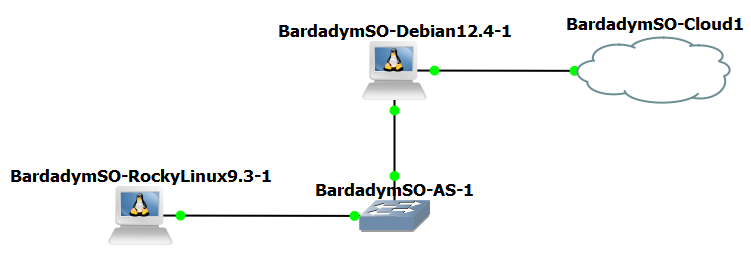


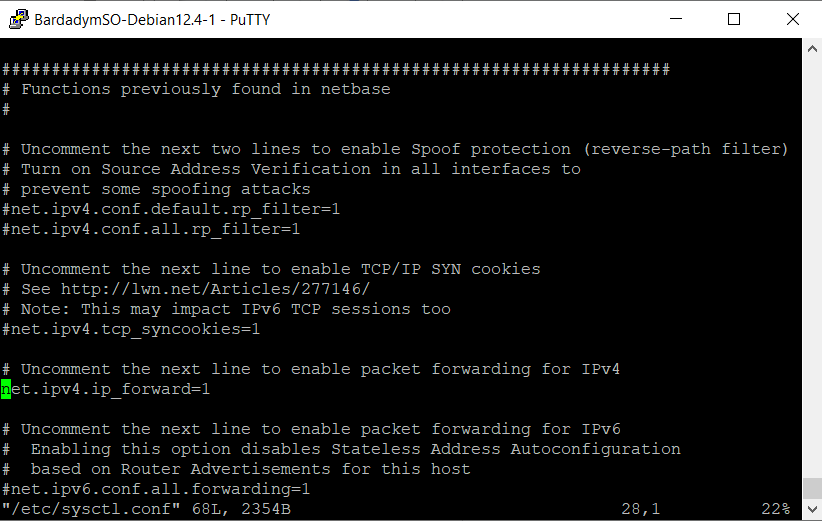


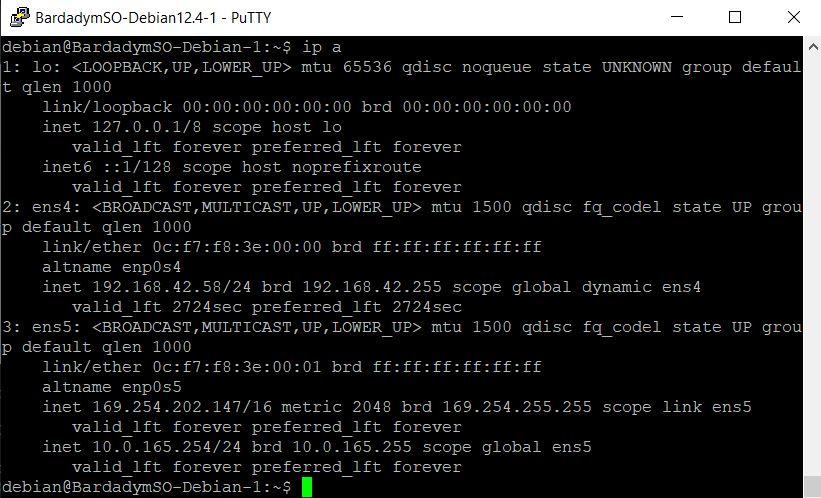


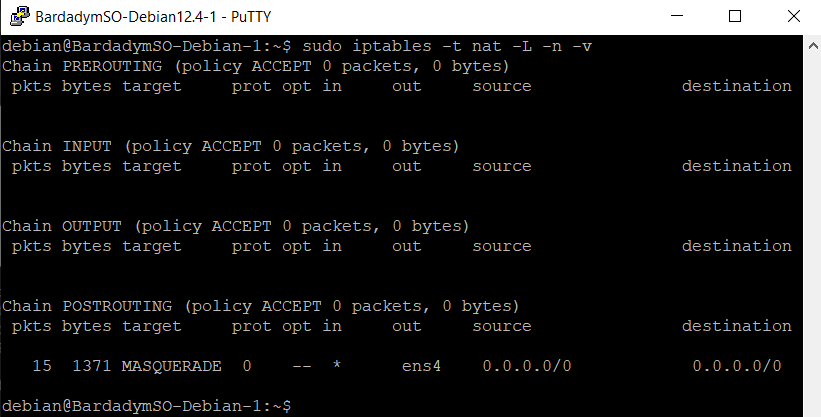


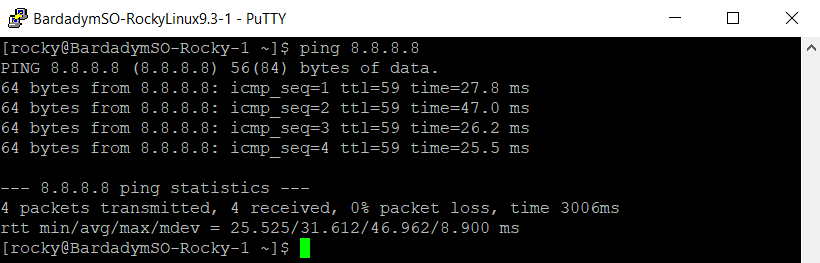
Задание 2. Добавить на debian второй интерфейс, задать на нем статический адрес из сети 10.0.LN.0/24. Разместить коммутатор, подключить к нему второй интерфейс debian. Развернуть вторую машину со RHEL-based операционной системой (CentOS, Alma Linux, Rocky Linux…). Настроить debian для работы в качестве шлюза, продемонстрировать необходимые настройки debian, факт работы второй машины через debian как шлюз.



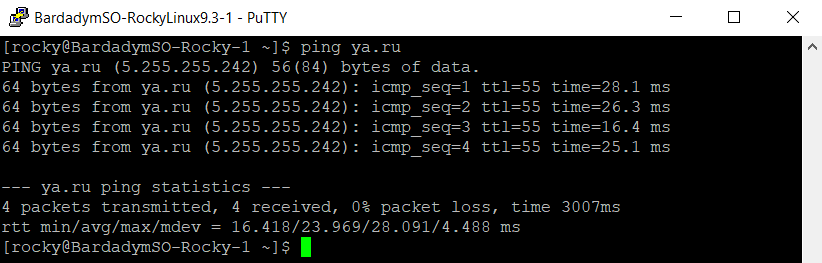


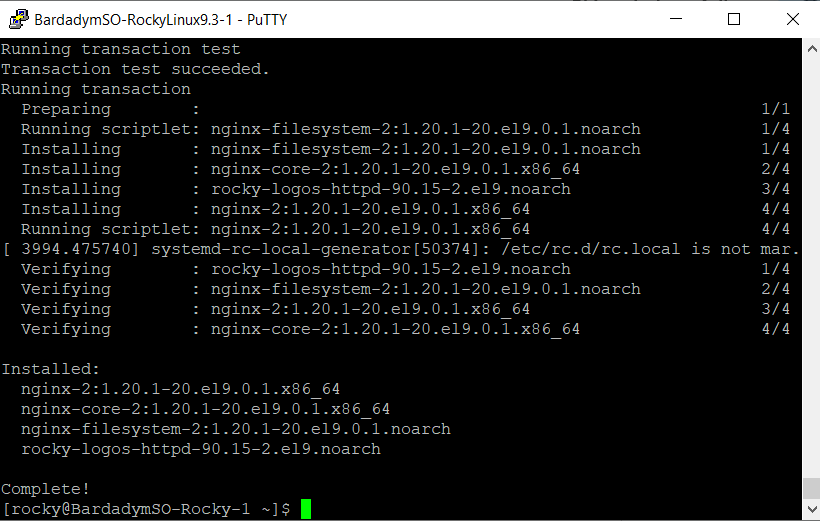






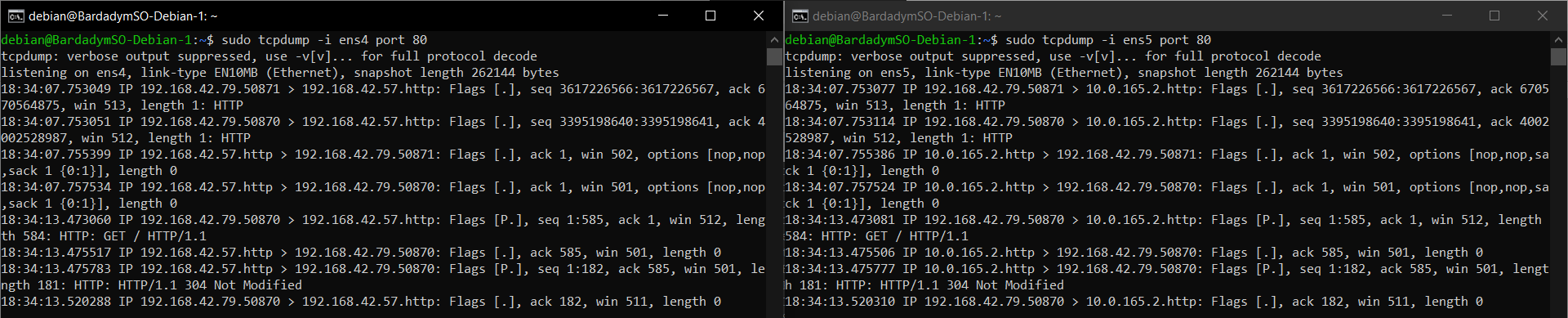
Задание 3. Развернуть на debian DNS-сервер. Настроить вторую машину на использование debian в качестве DNS-сервера. Показать работоспособность конфигурации, обновить все пакеты на второй виртуальной машине.



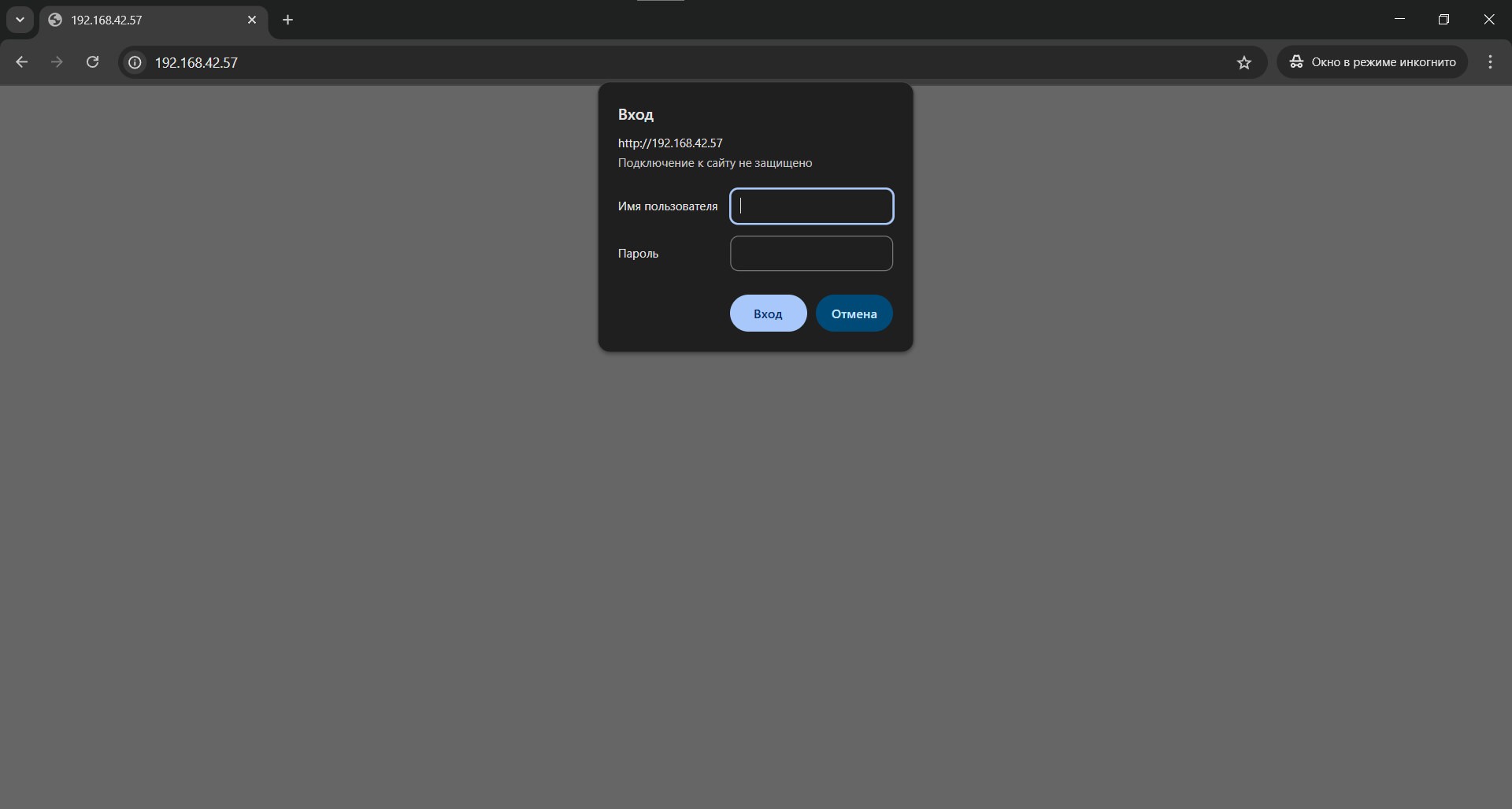


Задание 4. Установить на вторую машину веб-сервер. Настроить на debian port forwarding. Показать доступность веб-страниц с хоста. Показать работу технологии DNAT.

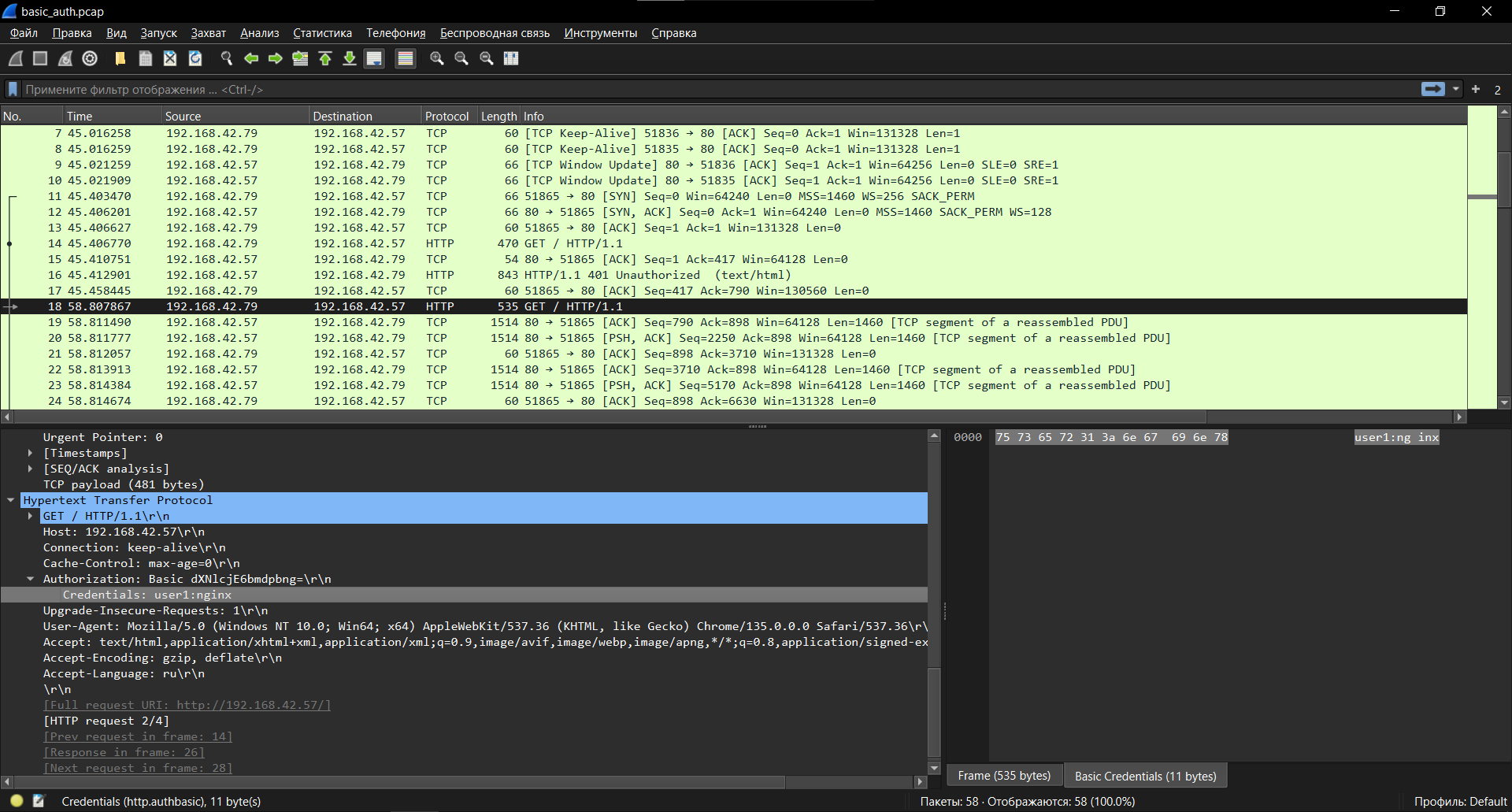




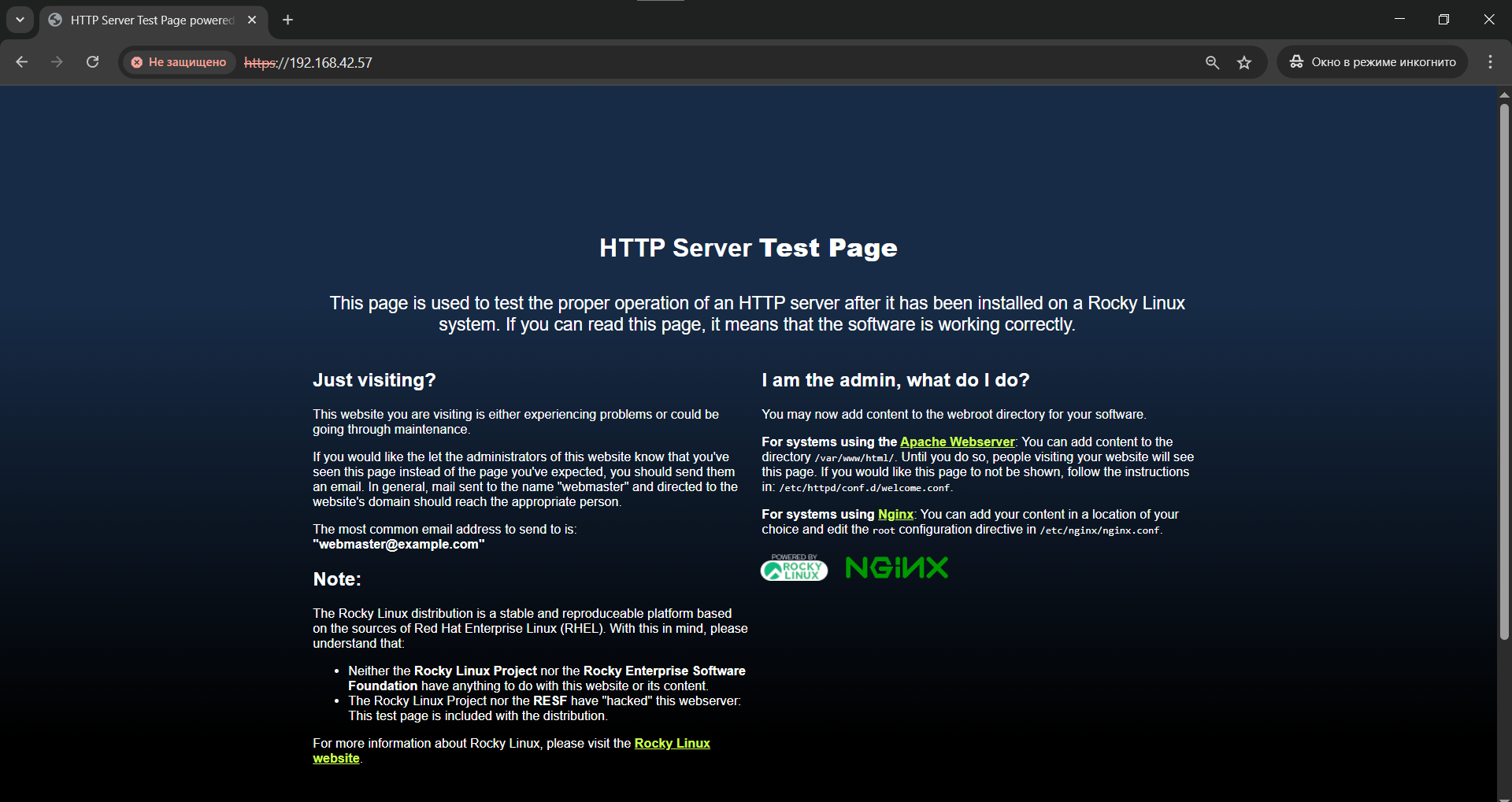
Задание 5. Настроить Basic Authorization. Перехватить пользовательский пароль. Настроить TLS. Показать пакеты обмена сообщениями в рамках TLS Handshake.

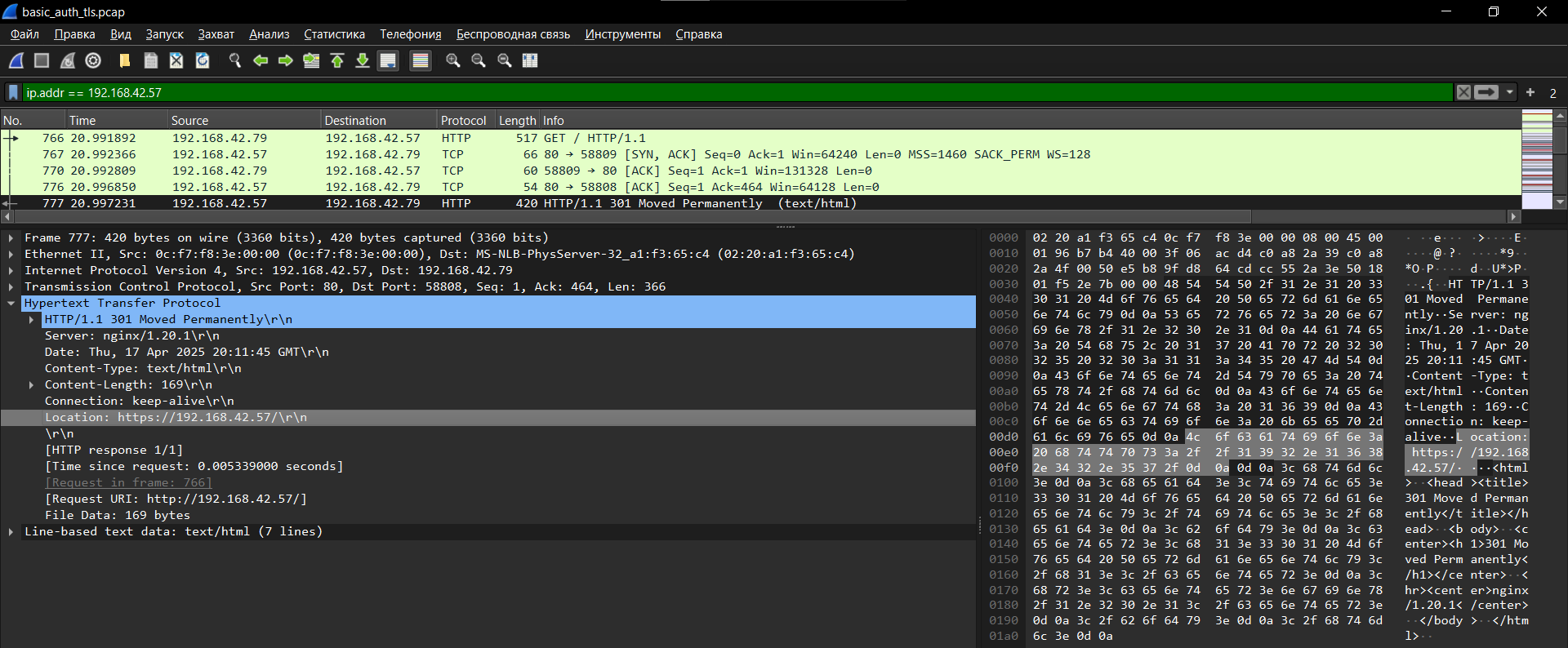


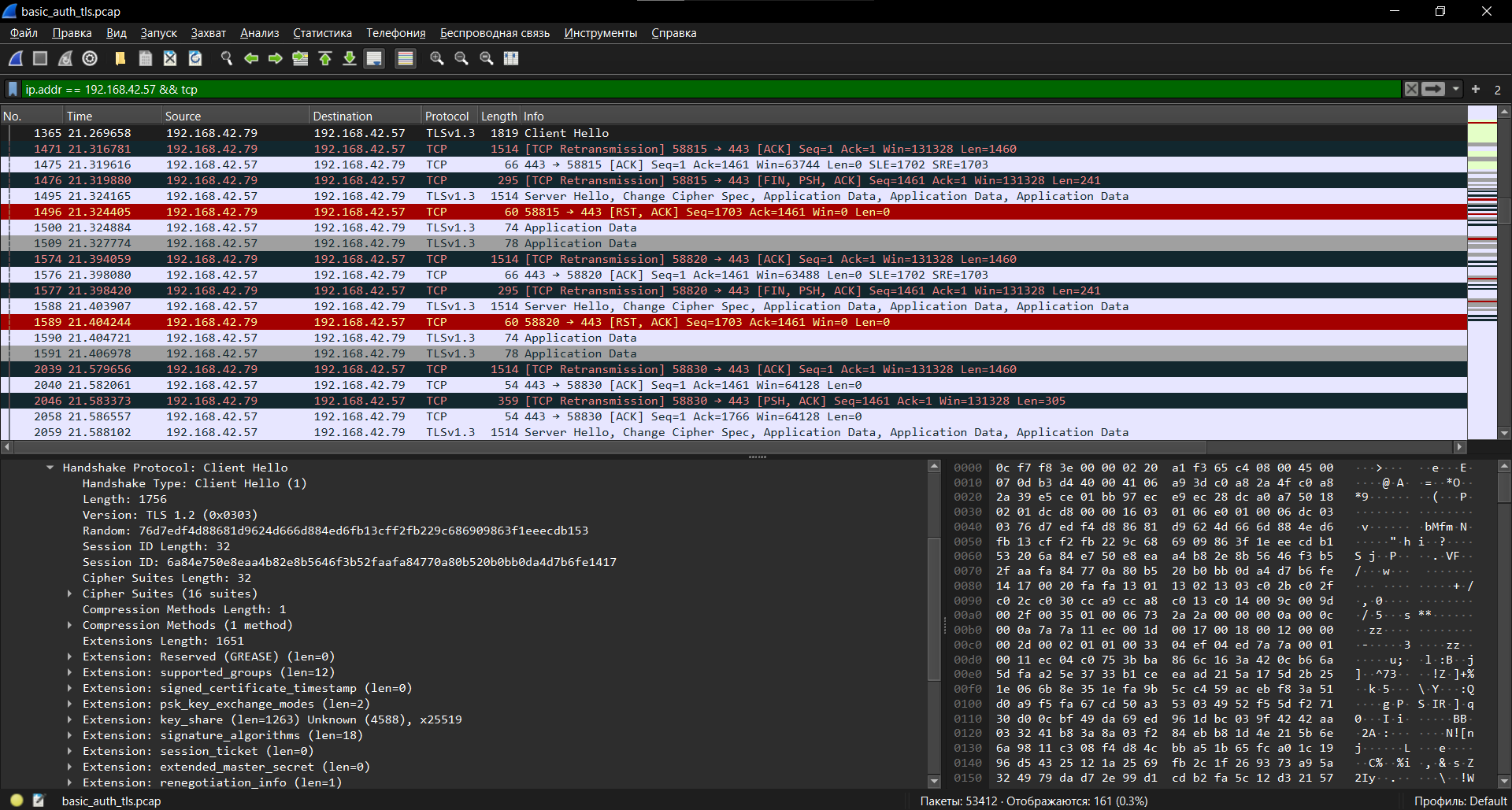




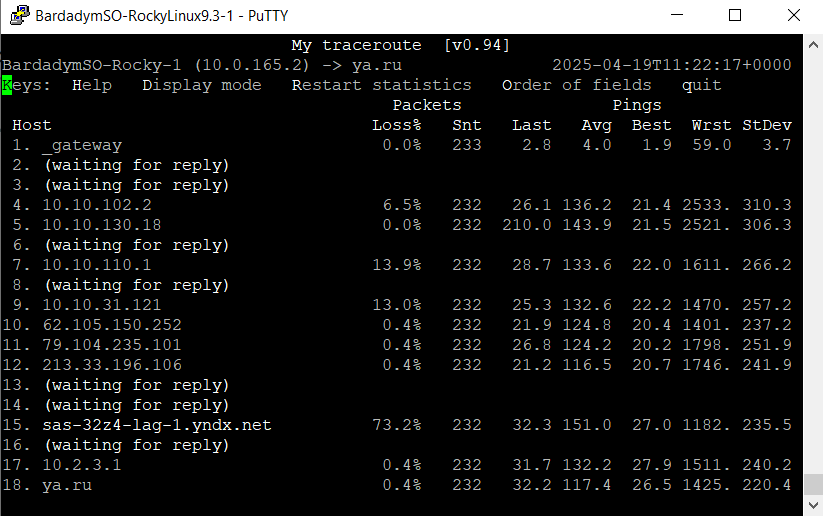


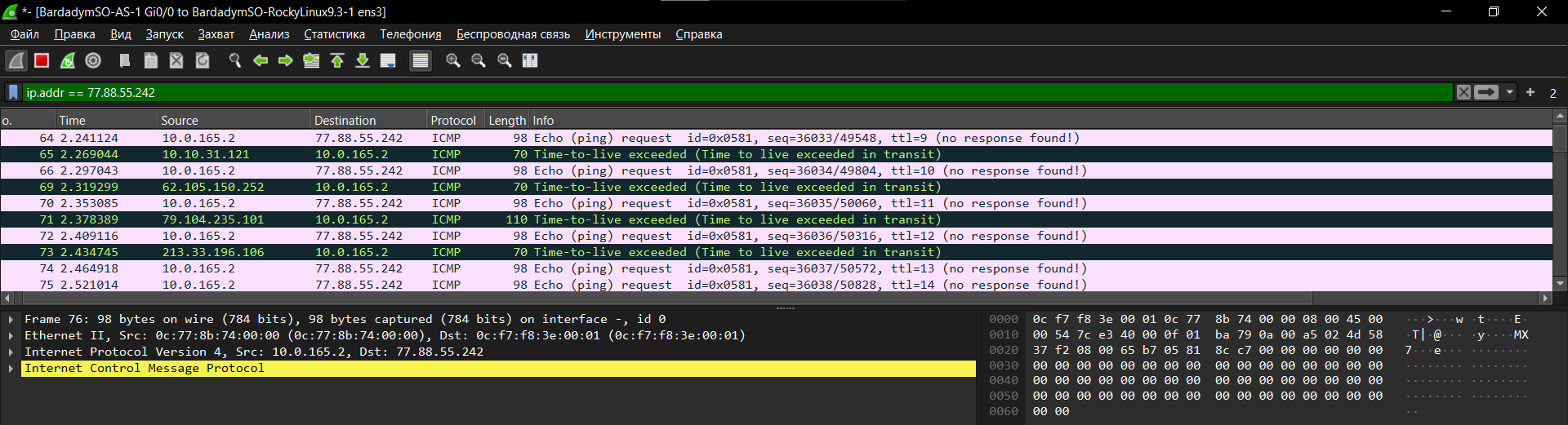




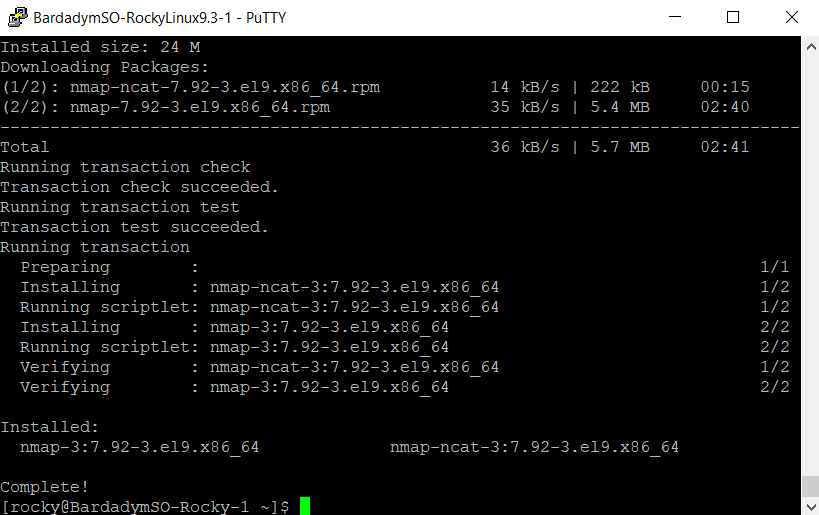


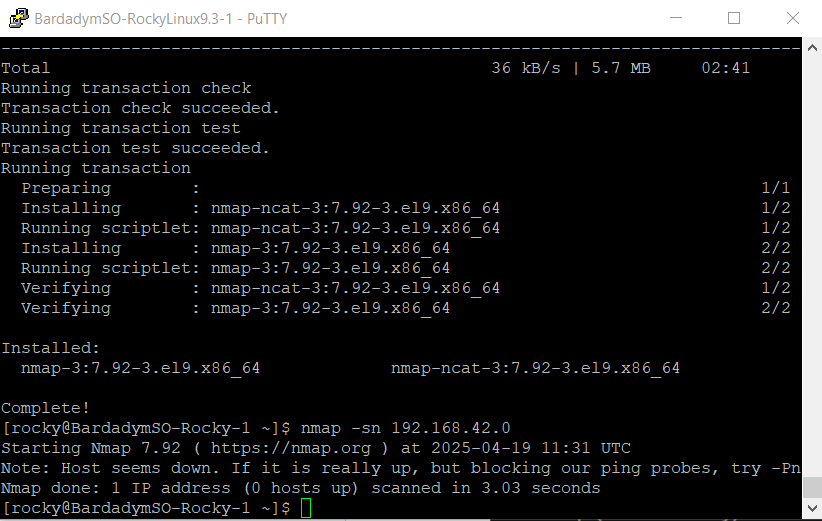
Задание 6. Провести трассировку любым удобным способом, захватывая трафик. На основе дампа захваченного трафика объяснить принцип алгоритма трассировки.

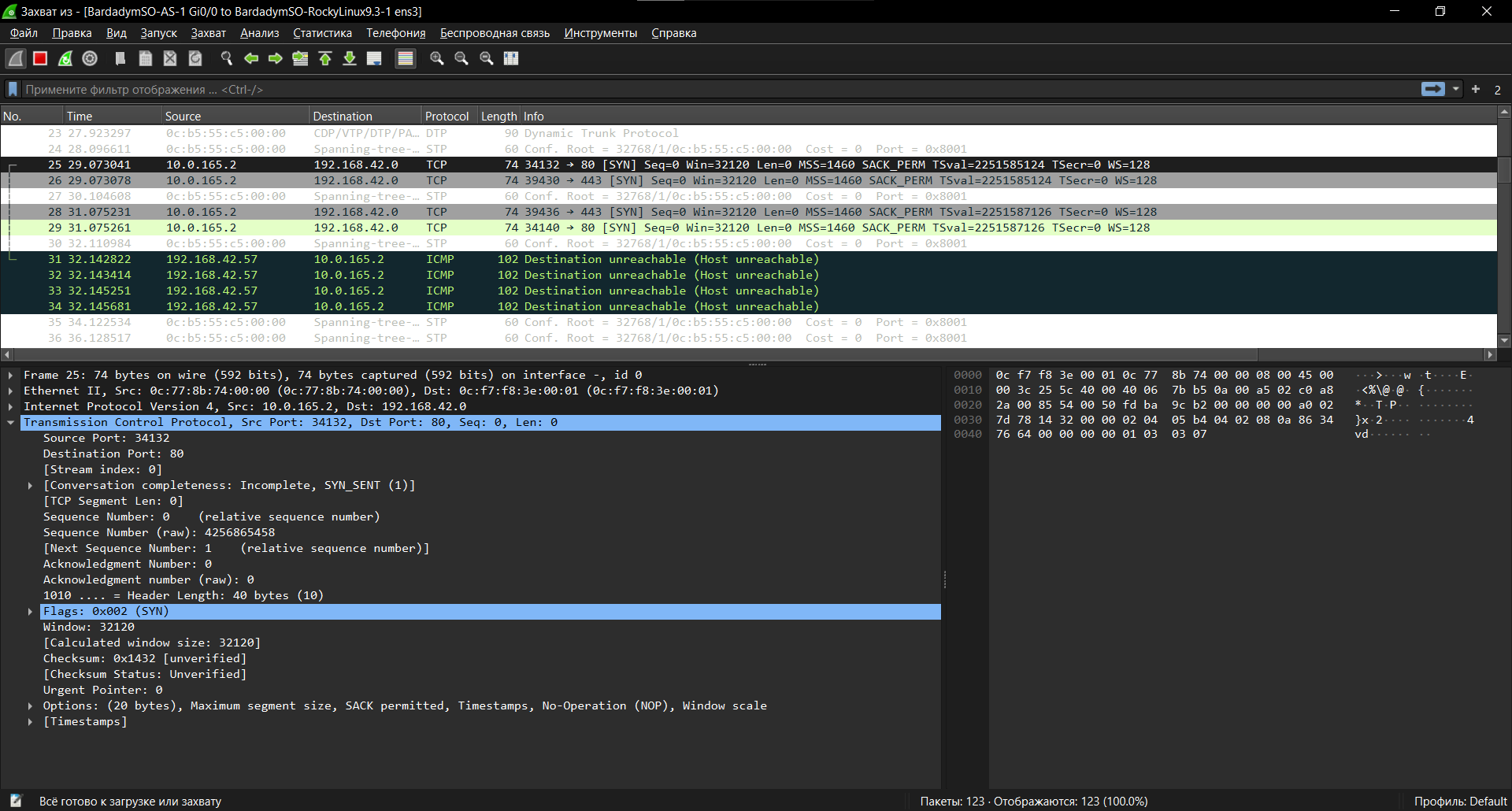




Задание 7. Установить на вторую машину nmap. Просканировать один и тот же диапазон активных хостов различными способами сканирования. Показать трафик в каждом случае, на его основе объяснить принцип сканирования. Сравнить результаты.







Сохраните проект в файл .gns3 и приложите с отчетом на диск.

**Контрольные вопросы 1:**

1. Какие основные сетевые настройки необходимо сконфигурировать на сетевом интерфейсе в любой операционной системе?

2. Перечислите способы конфигурации сети в зависимости от ОС.

3. Опишите принцип работы прокси-сервера.

4. Опишите принцип работы межсетевого экрана.

5. Опишите принцип процесса masquerade. Чем он отличается от source NAT?

**Контрольные вопросы 2:**

1. Что такое iptables, netfilter, firewalld?

2. Какими способами можно сохранить правила iptables, чтобы не потерять их после перезагрузки?

3. Опишите принцип работы утилит traceroute, tracert.

4. Опишите принципы ping- и SYN- сканирования.

5. В чем концептуальное различие между фильтрами захвата и отображения в wireshark?

| Студент: | Бардадым С. О. | Допуск: |  |
| --- | --- | --- | --- |
| **Отчет по работе №3 (часть 2)**  Организация сетей в контейнерной инфраструктуре | | | |
| Дата выполнения | 24.04.25 | Дата защиты | 17.05.25 |
| Оценка |  | Подпись |  |

Цель работы: получить опыт работы с сетями в контейнерах.

Задачи работы: развернуть виртуальную машину с RHEL-based системой, настроить сети в пространствах имен, настроить сетевую связность, установить docker, написать docker-compose.yaml, на основе которого будут развертываться 5 контейнеров: 4 веб-сервера и 1 балансировщик.

Краткий конспект теоретической части:

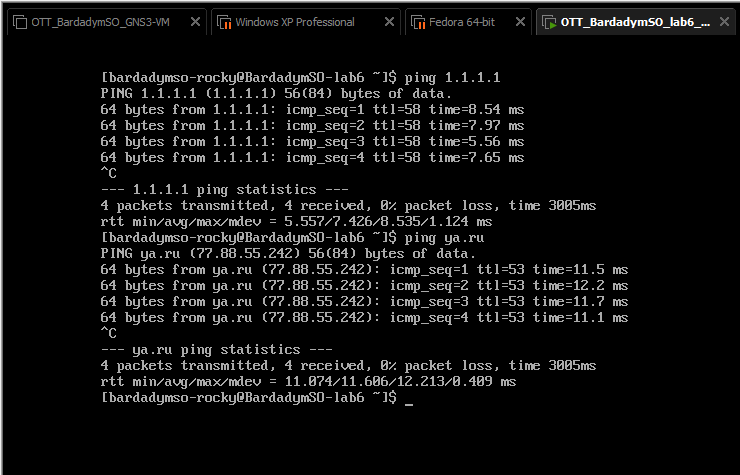
| Различие между виртуализацией и контейнеризацией  Виртуализация создаёт полное изолированное окружение, которое включает собственную операционную систему, тогда как контейнеризация использует общую операционную систему, изолируя только программные уровни.  Контейнер  Контейнеры – это легкие программные пакеты, содержащие все зависимости, необходимые для запуска автономного программного приложения.  Пространство имен Linux  Пространство имён (от англ. namespaces) – функция ядра Linux, позволяющая изолировать и виртуализировать глобальные системные ресурсы множества процессов.  Сетевые интерфейсы в Linux  Сетевой интерфейс (сетевой адаптер) – это часть аппаратного устройства, которая позволяет операционным системам и программам передавать данные через компьютерную сеть.  Виртуальная память – это технология управления памятью, используемая операционными системами  iptables – это утилита командной строки в Linux, предназначенная для настройки встроенного брандмауэра Netfilter. |
| --- |

В этой и следующих работах следует придерживаться простых правил:

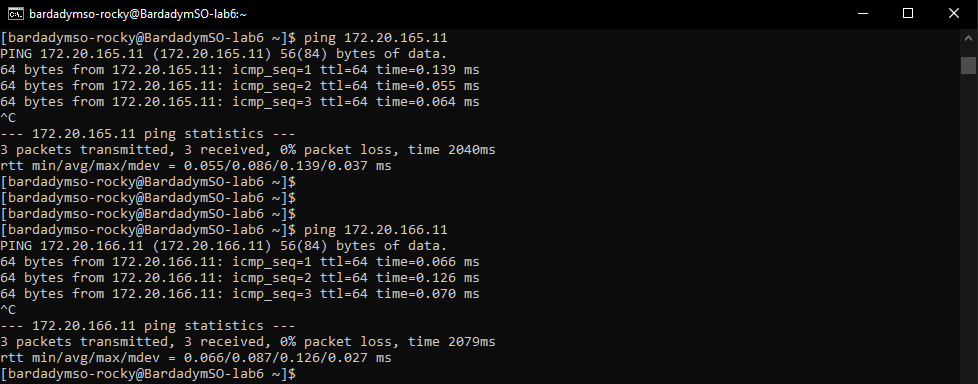
- hostname устройств должен содержать фамилию и инициалы студента, если не указано иное,

- вместо LN должен быть подставлен персональный Lucky Number из таблички.

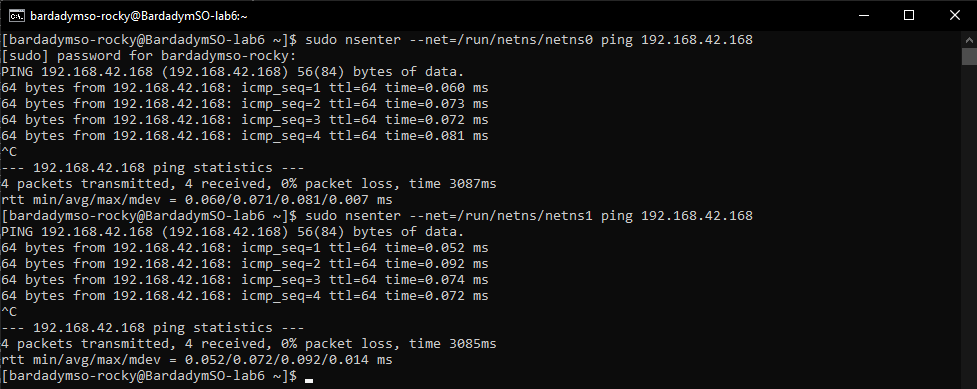
Задание 1. Развернуть новую виртуальную машину желательно с RHEL-based операционной системой без графического интерфейса. Создать пользовательскую учетную запись, состоящую из фамилии и инициалов студента, все задания выполнять из-под этой учетной записи. Предоставить виртуальной машине доступ в глобальную сеть. Продемонстрировать учетную запись и доступ в интернет.



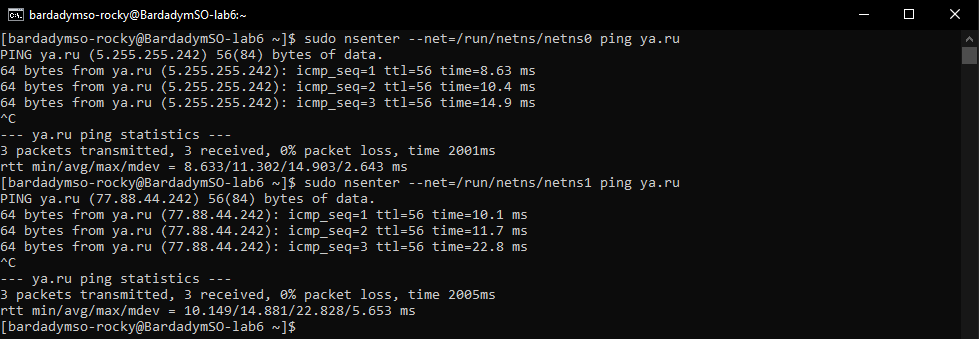
Задание 2. Создать два сетевых пространства имен Linux. Создать парные виртуальные интерфейсы, связать с их помощью основное пространство имен с созданными. Назначить IP-адреса из сетей 172.20.LN.0/24, 172.20.LN+1.0/24, продемонстрировать сетевую связность.

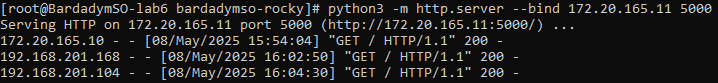


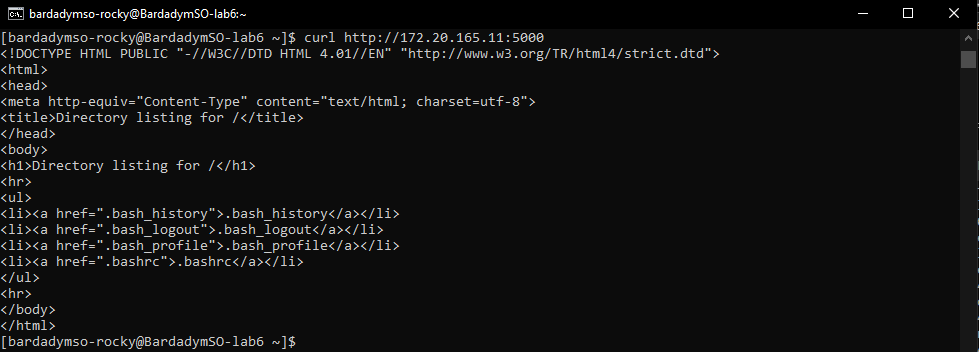
Задание 3. Создать в основном сетевом пространстве имен интерфейс bridge, связать с ним виртуальные сетевые интерфейсы. Настроить сетевой доступ из созданных пространств имен до внешнего интерфейса ВМ.

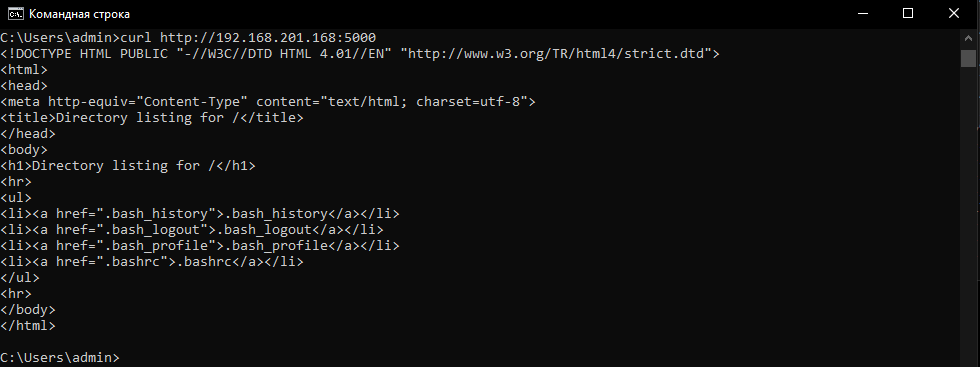


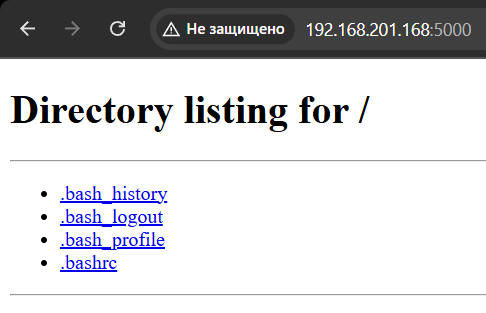
Задание 4. Настроить доступ в глобальную сеть из созданных пространств имен. Продемонстрировать работоспособность. В одном из пространств имен развернуть веб-сервер. Продемонстрировать доступ к нему из другого пространства имен, из виртуальной машины, с хост-компьютера.



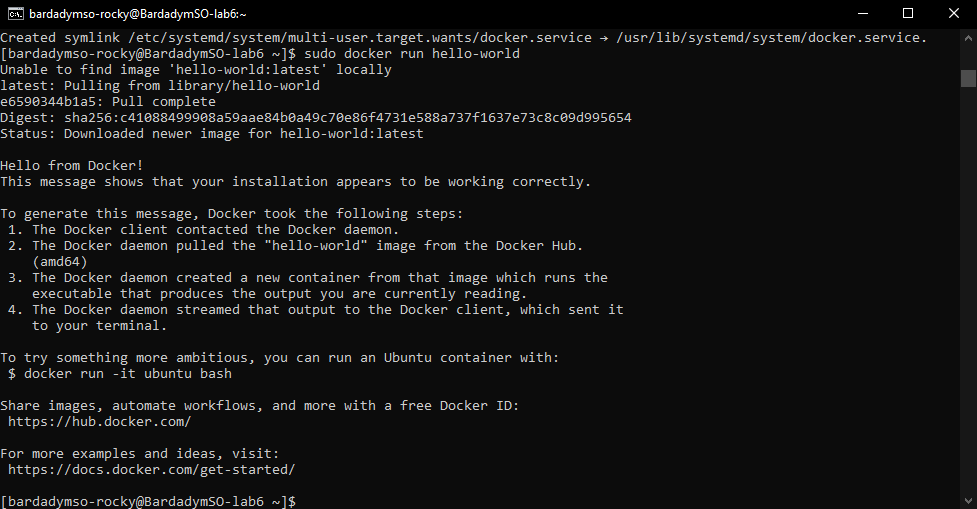


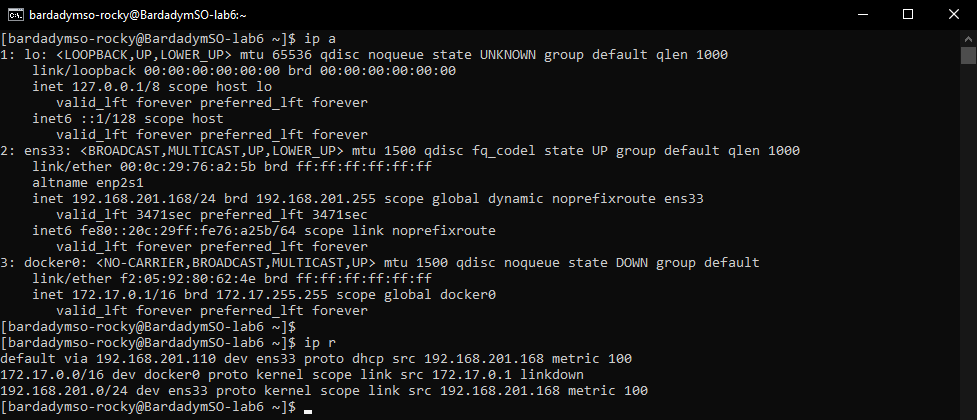


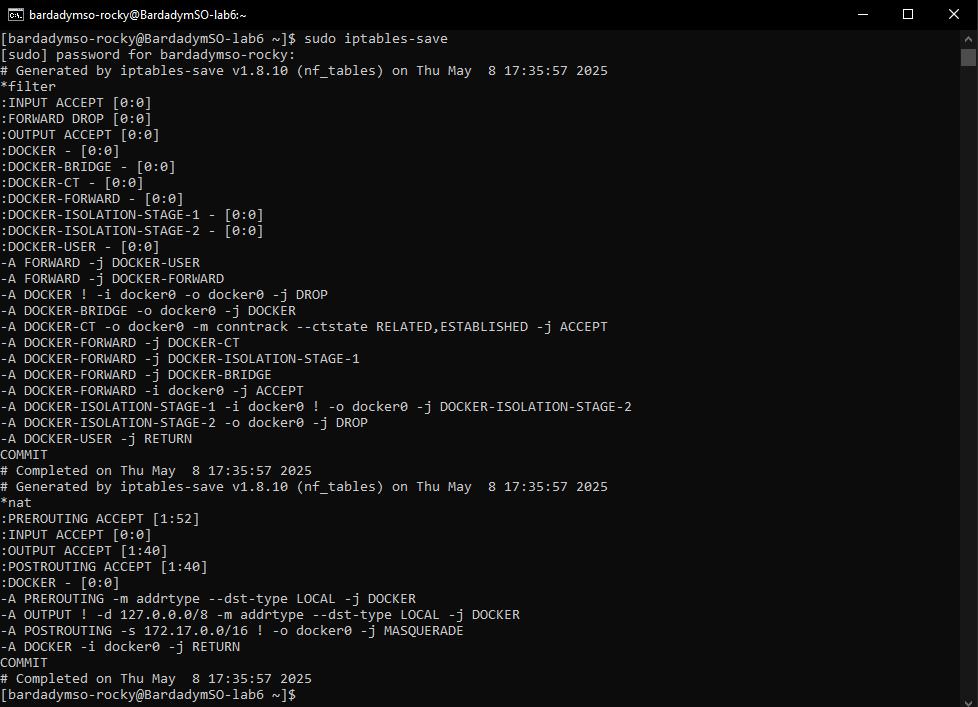




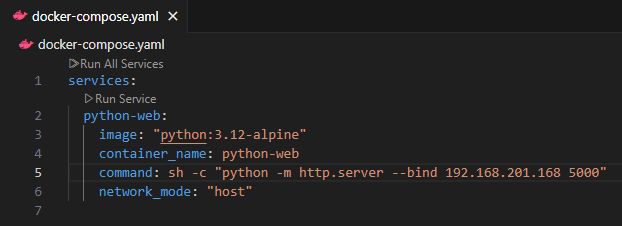
Задание 5. Установить docker. Показать созданные по умолчанию сетевые интерфейсы, правила iptables.

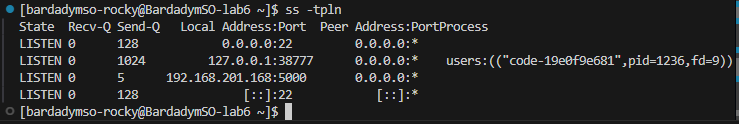


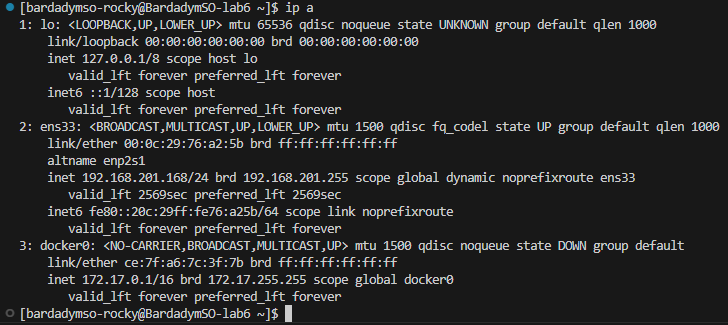


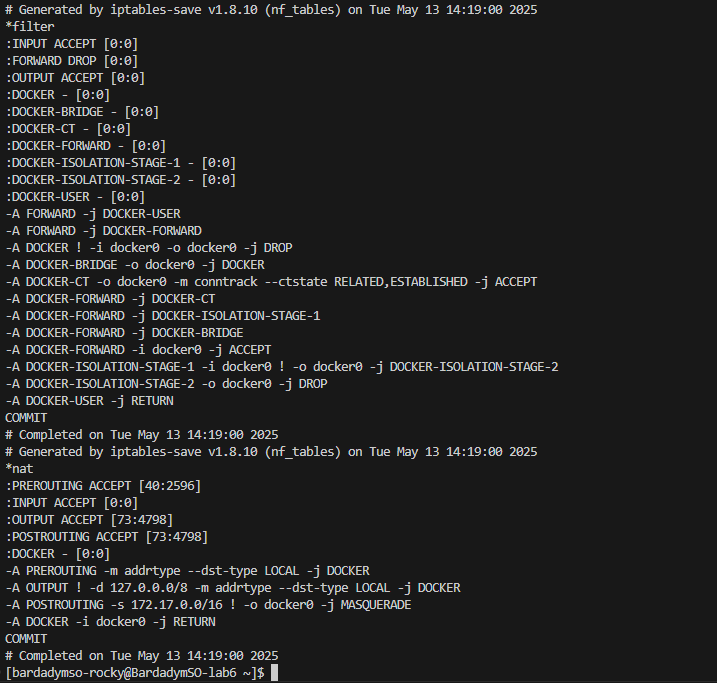


Задание 6. Написать docker-compose.yml файл, позволяющий развернуть docker-контейнер с веб-сервером с типом сети host. Показать используемые им интерфейсы, порты, правила iptables для доступа к нему.



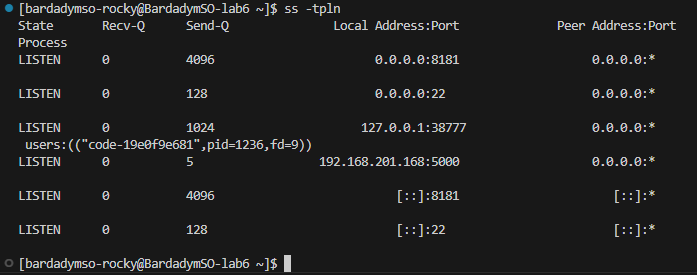


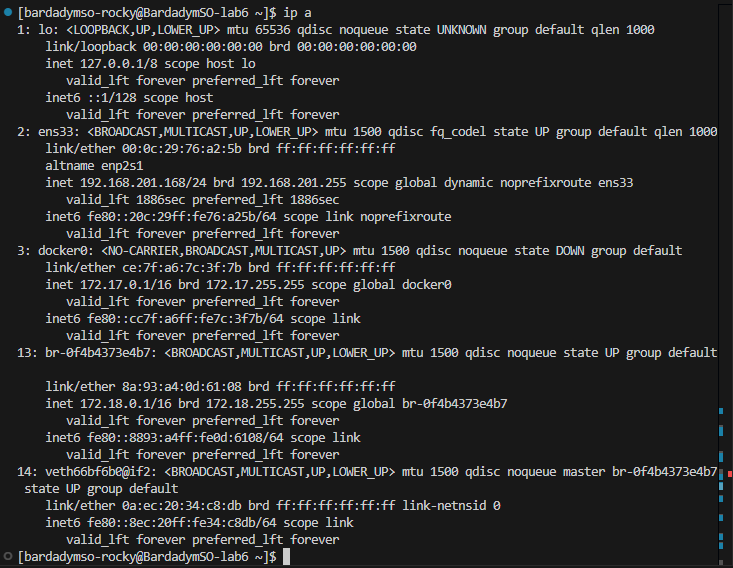


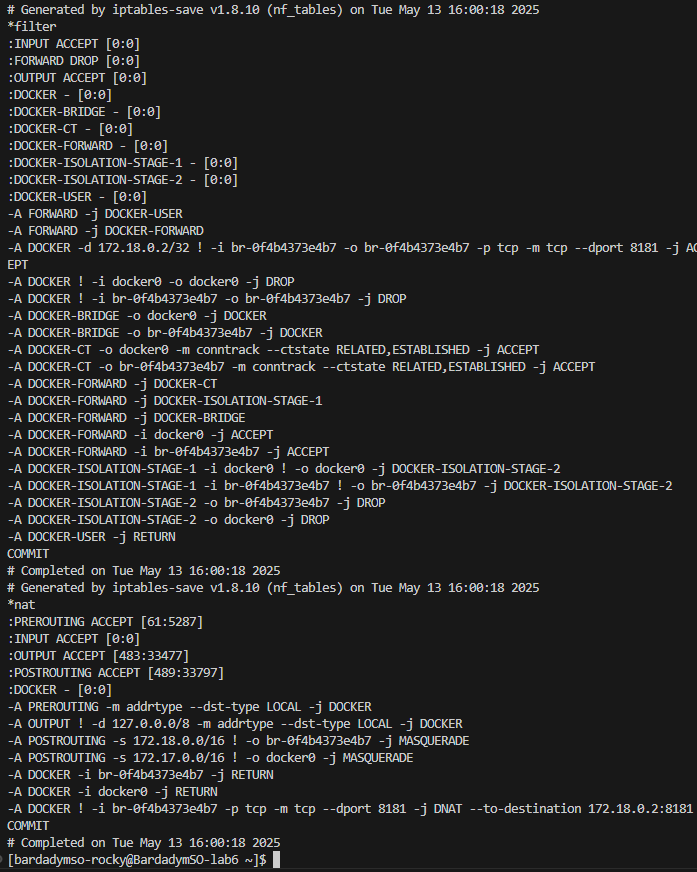


Задание 7. Добавить в docker-compose.yml еще один веб-сервер с новой сетью с типом bridge. Показать используемые им интерфейсы, порты, правила iptables для доступа к нему.

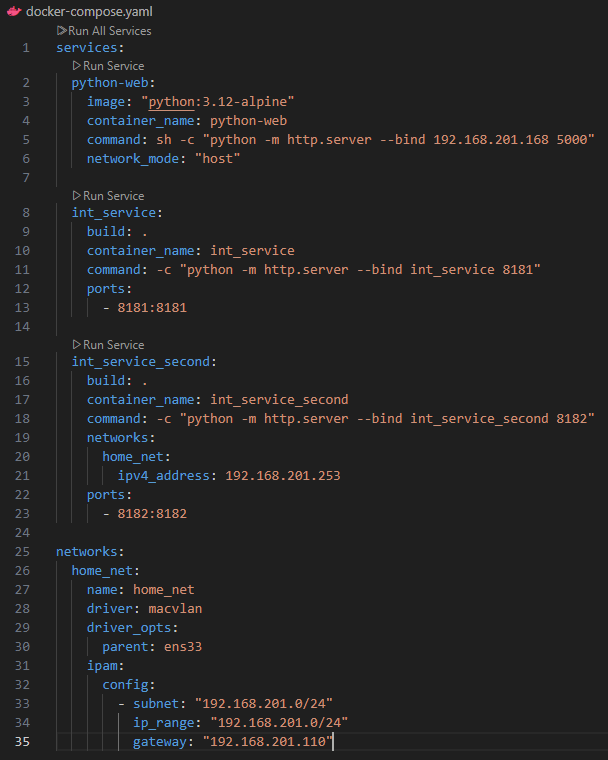


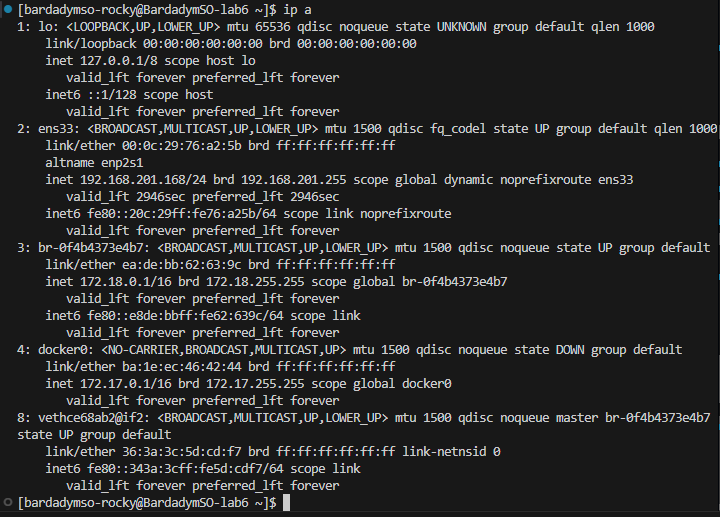


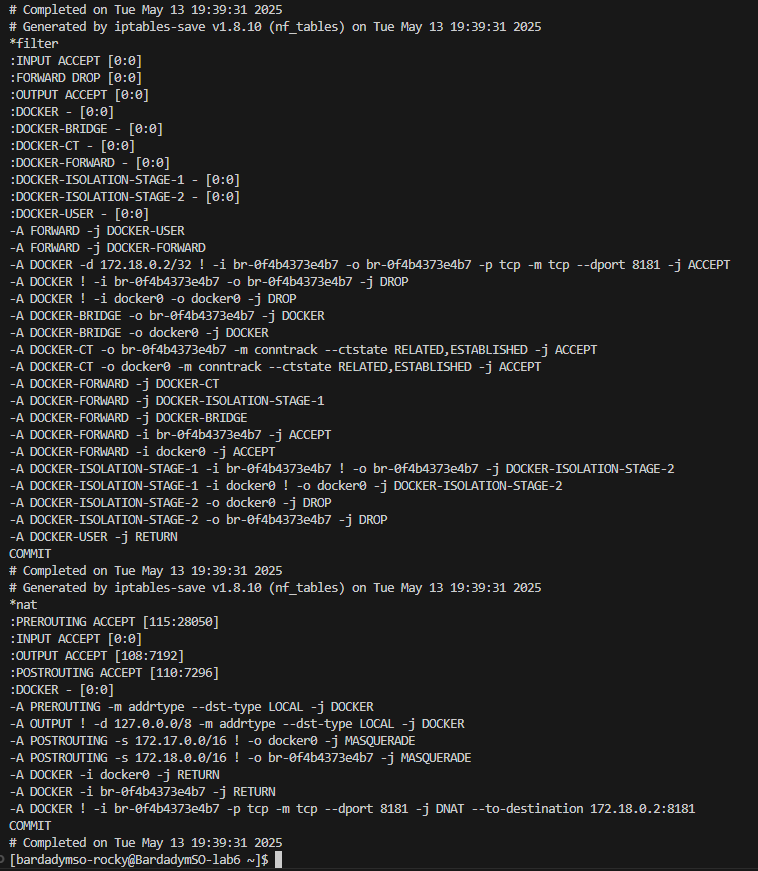




Задание 8. Добавить в docker-compose.yml еще один веб-сервер с новой сетью с типом macvlan. Показать используемые им интерфейсы, порты, правила iptables для доступа к нему.







Задание 9. Добавить в docker-compose.yml nginx в роли балансировщика и веб-сервер. Настроить сети и балансировщик так, чтобы веб-сервер был доступен только через него и недоступен при обращении к виртуальной машине напрямую.

