

Санкт-Петербургский Государственный Университет
Математико-механический факультет

Кафедра системного программирования

Кижнеров Павел Александрович

Реализация тестового стенда Network Function Virtualization

Курсовая работа

Научный руководитель:
ст. преп. Зеленчук И. В.

Санкт-Петербург
2019

Оглавление

Введение	3
1. Постановка задачи	5
2. Обзор	6
2.1. Введение в предметную область	6
2.1.1. Network Function Virtualization	6
2.1.2. Примеры VNF	9
2.2. Обзор существующих решений	9
2.2.1. Virtual Infrastructure Manager	10
2.2.2. NVF Orchestration	11
2.2.3. VNF	12
2.3. Выбор оборудования и технические требования	12
3. Реализация	14
3.1. Выбор платформы	14
3.2. Virtual Infrastructure Manager	14
3.2.1. OPNFV	14
3.2.2. Kernel-based Virtual Machine	15
3.2.3. Preboot eXecution Environment	17
Заключение	21
Список литературы	22

Введение

В современном мире существует тенденция к увеличению количества специализированных аппаратных устройств в сети крупных компаний. Внедрение нового сетевого сервиса подразумевает добавление новых устройств, каждое из которых требует:

- место в помещении;
- энергоснабжение;
- климатические ресурсы;
- время для настройки;
- квалифицированный обслуживающий персонал.

Более того, аппаратные решения подвержены "моральному" устареванию, что влечет потребность в замене оборудования и его дальнейшей настройке. Все это является причиной нерентабельности расширения сети из-за колоссальных издержек в перспективе.

Виртуализация физических устройств способна решить часть вышеупомянутых проблем.

- устройствам, которых не существует в аппаратной реализации, не требуется место в аппаратных комнатах;
- они не потребляют энергии (вместо них это делает устройство, на котором происходит виртуализация);
- им не требуются определенные климатические условия для бесперебойной работы;
- при должной развитости технологии виртуализации время настройки существенно меньше, т.к работа через CLI происходит быстрее чем физическое взаимодействие с устройством.

В рамках курсовой работы рассматривается задача создания тестового стенда с использованием технологии Network Function Virtualization (далее NFV), которая позволяет виртуализировать физические сетевые элементы посредством предоставления возможности сетевым функциям исполняться программными модулями на серверах или на виртуальных машинах (далее VM) в них. Мы считаем данную задачу актуальной на сегодняшний день, так как концепция виртуализации сетевой аппаратуры возникла относительно недавно и только начинает развиваться, в частности на кафедре системного программирования такие исследования до сих пор не велись.

1. Постановка задачи

Целью работы является создание и настройка тестового стенда технологии NFV, для достижения которой были выделены следующие задачи:

- проанализировать и выбрать готовое решения NFV;
- подобрать необходимое оборудование;
- установить Virtual Infrastructure Manager.

Развертывание NFV предполагает наличие некоторых навыков системного администратора, в связи с чем были выделены подзадачи:

- установка и настройка гипервизора KVM;
- установка через PXE операционной системы CentOS 7 с нужной конфигурацией.

2. Обзор

2.1. Введение в предметную область

2.1.1. Network Function Virtualization

Network Function Virtualization (NFV) – технология, позволяющая виртуализировать аппаратные реализации сетевых сервисов. Предложена в 2012 году ETSI [4].

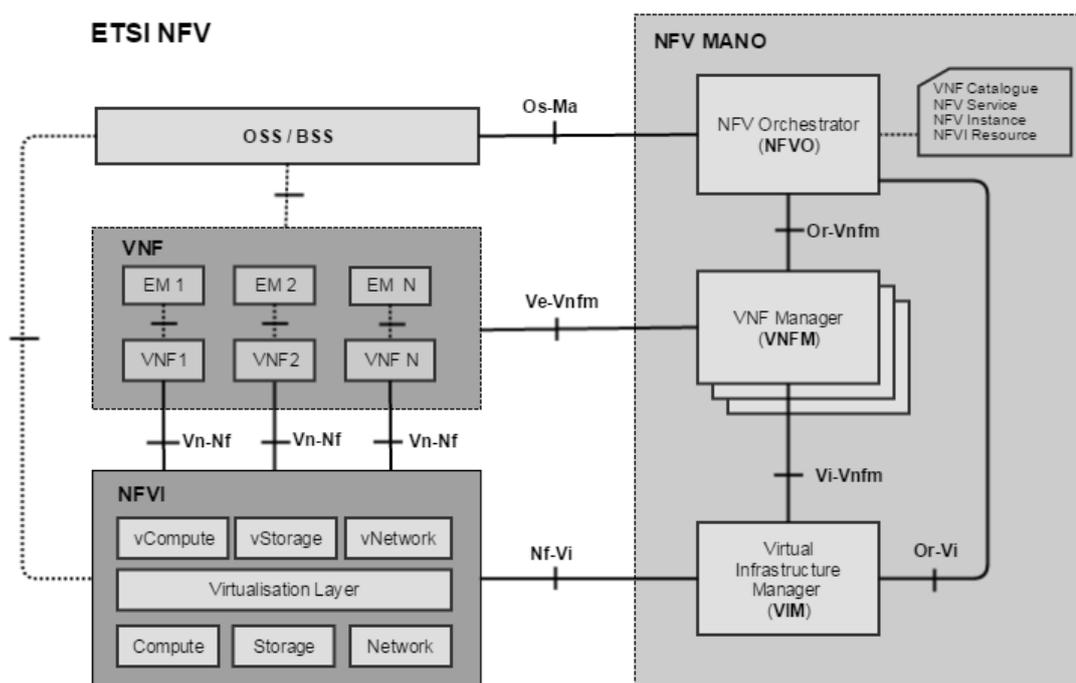


Рис. 1: Архитектура NFV, разработанная ETSI

Референсная точка – граница взаимодействия между двумя функциональными блоками. Это определение стоит отдельного внимания, так как оно похоже на определение интерфейса. Разница в том, что интерфейс – общая граница, через которую взаимодействуют два объекта, а референсная точка – общая граница именно функциональных блоков, которые могут состоять из множества аппаратных и программных объектов [7].

Virtual Network Function (VNF) – виртуализированные функции сетевых элементов (например: firewall, маршрутизатор). Как правило одному сетевому элементу соответствует одна VNF. Однако воз-

возможен случай, когда VNF виртуализирует часть функционала устройства, тогда одному сетевому элементу могут соответствовать несколько VNF.

Element Manager (EM) – система, управляющая работой виртуализированных функций. Отвечает за задание параметров операций виртуализированных функций. Управление происходит через закрытые интерфейсы, поэтому референсная точка не регламентирована. Можно провести параллель с Network Management System (NMS). NMS управляет физическими устройствами, EM – виртуализированными.

Virtual Network Function Manager (VNFM) – система, занимающаяся запуском, обслуживанием, и прекращением работы виртуализированных функций. Способна выполнять те же функции что и EM, однако через референсную точку, предложенную ETSI (Ve-Vnfm).

Виртуальные ресурсы – виртуализированные физические ресурсы, которые используются для работы VNF.

Аппаратные ресурсы – физическая часть, на которой запускаются виртуальные ресурсы (например: сервер, коммутатор, контроллер Software Defined Network).

Плоскость виртуализации – часть, ответственная за размещение виртуальных ресурсов на аппаратных ресурсах (например гипервизор).

Network Function Virtualization Infrastructure (NFVI) – инфраструктура, в которой работают виртуализированные функции. Включает в себя аппаратные ресурсы, виртуальные ресурсы а также плоскость виртуализации.

Virtualized Infrastructure Manager (VIM) – система администрирования инфраструктуры виртуализированных функций сетевых элементов. Отвечает за контроль и управление вычислениями, памятью и сетевыми ресурсами в пределах одного домена инфраструктуры, то есть в архитектуре могут быть несколько VIM, каждый из которых управляет своей частью инфраструктуры. Также собирает данные о производительности.

NFV Orchestrator – часть архитектуры NFV, занимающаяся управлением и администрированием виртуализированных сетевых функций,

но не напрямую, а косвенно через VIM и VNFM.

Operation Support System/Business Support System (OSS/BSS)

– система поддержки операций и бизнеса. Управление сетью: управление при отказах, управление конфигурацией и управление услугами - сфера деятельности OSS. BSS отвечает за управление клиентами, управление продуктами и управление заказами.

Репозитории – сущности, которые хранят различную информацию.

NS Catalog – каталог используемых сетевых услуг. Также хранит шаблоны развертывания для сетевых услуг.

Virtual Network Function Descriptor (VNFD) – шаблон, описывающий сценарий развертывания и использования виртуализованных сетевых функций.

VNF Catalog – каталог всех используемых VNFD.

NFV Instances – репозиторий: содержащий все детали об экземплярах сетевых услуг.

NFVI Resources – список ресурсов инфраструктуры, задействованных в работе сетевых услуг.

Management and Orchestration (MANO) – архитектурная единица: включающая в себя:

- Virtualized Infrastructure Manager;
- Virtual Network Function Manager;
- NFV Orchestrator;
- репозитории:
 - NS Catalog;
 - VNF Catalog;
 - NFV Instances;
 - NFVI Resources.

2.1.2. Примеры VNF

Межсетевой экран – устройство обеспечения безопасности сети, которое осуществляет мониторинг входящего и исходящего сетевого трафика и на основании установленного набора правил безопасности принимает решения, пропустить или заблокировать конкретный трафик [11]. Аппаратура с интегрированным сетевым экраном стоит в разы дороже, чем аналогичные устройства без этой функции, а изменение топологии сети может привести как к недостатку, так и к избытку такого оборудования, что нерентабельно. Виртуальные межсетевые экраны решают эту проблему.

QoS (Quality of service) – способность сетевой инфраструктуры предоставлять улучшенное обслуживание определенному виду передаваемого трафика. Архитектура NFV дает возможность добавлять виртуальные сетевые функции, маркирующие и проходящие пакеты и маршрутизирующие их в порядке приоритета.

Маршрутизатор – устройство выполняющее поиск пути доставки пакета между сетями. Виртуальный маршрутизатор позволит из одного пункта управления редактировать таблицу маршрутизации с целью добавления и удаления хостов из сети.

DPI (Deep Packet Inspection) – технология, которая обнаруживает, распознает, классифицирует, перенаправляет или блокирует пакеты с конкретными данными или полезной нагрузкой, которые обычная фильтрация пакетов (которая проверяет только заголовки пакетов) не может обнаружить [12]. В сети NFV трафик проходит через слой инфраструктуры, что позволяет настроить виртуальное устройство, выполняющее глубокий анализ трафика.

2.2. Обзор существующих решений

На сегодняшний день ситуация с готовыми решениями для NFV обстоит следующим образом: VIM реализуется с помощью набора компонентов OpenStack, который зависит от конкретных нужд; NFVO предоставляют такие проекты как Open Source MANO, ONAP, Open Baton,

Sonata; VNF + VNFM реализованы такими организациями как A10 vThunder, Alpha Networks Inc., Anritsu MasterClaw vProbe и многими другими. [9]

2.2.1. Virtual Infrastructure Manager

VMWare vDirector – проприетарное решение от VMWare, предоставляющее удобные возможности инициализации и использования облачных ресурсов. В это решение встроены такие услуги, как защита данных, аварийное восстановление, управление несколькими средами и другие.

OpenStack – комплекс проектов с открытым исходным кодом, позволяющий создавать облачные сервисы. Также может быть использован для реализации инфраструктуры и VIM для NFV. Известно несколько проектов, позволяющих произвести комплексную установку и настройку программного окружения для виртуальной инфраструктуры, используя компоненты OpenStack: Open Platform for NFV (OPNFV), Canonical OpenStack, Red Hat OpenStack Platform (RHOSP), VMWare vCloud NFV OpenStack Edition [8] [2] [10]. Другой путь настройки виртуального инфраструктурного менеджера – установка и конфигурация необходимых компонентов OpenStack вручную.

Open Platform of NFV – интегрированная платформа с открытым исходным кодом для виртуализации сетевых функций, распространяющаяся с лицензией Apache. Является коллаборативной разработкой организаций, входящих в The Linux Foundation, также тесно связана с ETSI.

Canonical OpenStack – проприетарное решение от частной компании Canonical, занимающейся коммерциализацией продуктов с открытым исходным кодом; заключается в установке и дальнейшем сопровождении компонентов OpenStack третьими лицами на аппаратуру под управлением Ubuntu Linux.

Red Hat OpenStack Platform – проприетарное решение от Red Hat, предлагающее платформу облачных вычислений с EPA (Enhanced Platform Awareness). EPA – методология, направленная на улучшение

производительности виртуальных машин путем предоставления им информации о состоянии физического оборудования. Также платформа содержит дистрибутив OpenDayLight для работы с SDN контроллерами.

VMWare vCloud NFV OpenStack Edition – проприетарное решение от VMWare, которое, по заявлению производителя, является наикратчайшим путем для создания служб, работающих на компонентах OpenStack.

	OPNFV	Canonical	Red Hat	VMWare	Pure OpenStack
Open source	Yes	Yes	No	No	Yes
Auto deploy	Yes	Yes	Yes	Yes	No
Clear documentation	Partially		Yes	Yes	Partially
Clear documentation	Partially		Yes	Yes	Partially
Installation difficulty	Medium		Easy	Easy	Hard
Free	Yes	No	No	No	Yes

Рис. 2: Решения VIM

2.2.2. NVF Orchestration

Open Source MANO – проект с открытым исходным кодом, разрабатываемый рабочей группой под руководством ETSI. Туда входят такие имена как Telefonica, Intel, Ubuntu, VMWare [1]. Предоставляет NFV Orchestrator. На этот проект возложены большие надежды, так как продвигается самим ETSI.

Open Network Automation Platform (ONAP) – разработка The Linux Foundation, предоставляющая всеобъемлющую платформу для управления и автоматизации физических и виртуальных сетевых функций. Open-source программный продукт.

Cloudify NFVO – open-source разработка компании Cloudify. Является не настолько популярным решением для оркестрации NFV, судя по количеству упоминаний в сети.

Sonata – open-source проект, имеющий академическую направленность. Его цель – не скорейшая коммерческая эксплуатация, а глубокое изучение и проработка молодой концепции NFV. В GitHub можно наблюдать контрибьюцию кода в текущую версию, из чего можно сделать вывод, что этот релиз нельзя назвать стабильным.

2.2.3. VNF

Удалось найти несколько готовых виртуальных сетевых функций, совместимых с оркестратором OSM [9].

A10 vThunder Application Delivery Controller – виртуальный балансировщик нагрузки для сервера, работающий на уровнях модели OSI выше транспортного.

A10 vThunder Carrier Grade NAT – виртуальный шлюз трансляции сетевых адресов операторского уровня, допускающий миграцию с IPv4 на IPv6.

Empirix Virtual Agent – виртуальная сетевая функция для сбора и анализа пакетов в режиме реального времени.

Hillstone CloudEdge – Виртуальный межсетевой экран, встроенный в Hillstone Networks StoneOS. Развертывается как виртуальная машина и предоставляет расширенные службы безопасности для приложений и пользователей в любой виртуализированной среде.

2.3. Выбор оборудования и технические требования

Бесперебойная работа всех элементов архитектуры NFV требует наличия существенных вычислительных ресурсов, особенно это касается узлов, занимающихся оркестрацией.

Для более подробного изучения этого вопроса было принято решение связаться с sdn-lab Санкт-Петербургского государственного университета телекоммуникаций им. проф. М. А. Бонч-Бруевича.

Коллеги из sdn-lab Санкт-Петербургского государственного университета телекоммуникаций им. проф. М. А. Бонч-Бруевича поделились техническими характеристиками своего стенда NFV + SDN, созданного с помощью компонентов OpenStack.

Наименование устройства	Технические параметры
Сервер (Контроллер SDN)	<ul style="list-style-type: none"> • описание: Материнская плата • продукт: P8Z77-V PRO • производитель: ASUSTeK COMPUTER INC. • производитель: American Megatrends Inc. • описание: ЦПУ • продукт: Intel(R) Xeon(R) CPU E3-1220 V2 @ 3.10GHz • производитель: Intel Corp. • физический ID: 4 • версия: Intel(R) Xeon(R) CPU E3-1220 V2 @ 3.10GHz • слот: LGA1155 • размер: 2394MHz • capacity: 3800MHz • разрядность: 64 bits • частота: 100MHz • возможности: x86 • RAM: 10 Gb. • ROM: 300 Gb.
Сервер (сервисы и приложения).	<ul style="list-style-type: none"> • описание: Материнская плата • продукт: P8Z77-V PRO • производитель: ASUSTeK COMPUTER INC. • производитель: American Megatrends Inc. • описание: ЦПУ • продукт: Intel(R) Xeon(R) CPU E3-1220 V2 @ 3.10GHz • производитель: Intel Corp. • физический ID: 4 • версия: Intel(R) Xeon(R) CPU E3-1220 V2 @ 3.10GHz • слот: LGA1155 • размер: 2394MHz • capacity: 3800MHz • разрядность: 64 bits • частота: 100MHz • возможности: x86 • RAM: 16 Gb. • ROM: 300 Gb.
<ul style="list-style-type: none"> • SDN коммутатор (Mikrotik) 	<ul style="list-style-type: none"> • Mikrotik RB 201 1UI AS-RM
<ul style="list-style-type: none"> • SDN коммутатор (Zelax) 	<ul style="list-style-type: none"> • ZES v.3028GPX

Рис. 3: Технические характеристики

Название	OpenStack (платформа)			SDN
	OpenStack Controller	OpenStack Compute	OpenStack Network	SDN Controller
Число ядер	8	4	2	4
Оперативная память, ГБ	16	8	8	12
Жёсткий диск, ГБ	100	100	80	80

Рис. 4: Минимальные требования компонентов OpenStack

В силу того, что все решения с открытым исходным кодом используют компоненты OpenStack, можно опираться на приведенную техническую спецификацию при выборе оборудования.

3. Реализация

3.1. Выбор платформы

Из существующих решений VIM для наших нужд был выбран OPNFV, так как это проект с открытым исходным кодом, распространяющийся с лицензией Apache; установка компонентов OpenStack производится в автоматическом режиме; на момент исследования готовых решений в сети Интернет больше всего информации было именно по OPNFV.

Для оркестрации подходят OSM, ONAP, Cloudify NFVO и Sonata. В первую очередь необходимо исследовать область VIM, так как выбор оркестратора без настроенной инфраструктуры не имеет смысла. Предварительно OSM удовлетворяет нашим нуждам, так как этот проект находится под контролем ETSI.

3.2. Virtual Infrastructure Manager

3.2.1. OPNFV

Существуют несколько релизов OPNFV, из которых был выбран самый новый стабильный – Gambia 7.2.

В контексте проекта OPNFV существует такое понятие как сценарий. Сценарий – это совокупность из установщика, конфигурации сети NFV и программных компонентов, необходимых для работы VIM в конкретной конфигурации сети NFV.

Для наших целей подходит тривиальная конфигурация сети и минимальный набор программных компонентов для VIM, поэтому выбор сценария был сделан в пользу `os-nosdn-nofeature-noha`. `os` – OpenStack, `nosdn` – нет поддержки SDN, `nofeature` – нет дополнительного функционала по типу мониторинга состояния физического оборудования, `noha` – физическое оборудование не является отказоустойчивым кластером.

Такой сценарий поддерживают два установщика – TripleO (Apex) и MCP (Fuel)[3]. Выбор был сделан в пользу MCP, так документация к этому установщику показалась более ясной и полной, чем к TripleO [5]

Scenario	Installer	Owner	Jenkins Job Created (Y/N)	Intent to release 7.0 (Y/N) (1,2)	Intent to release 7.1 (Y/N) (1,2)	Intent to release 7.2 (Y/N) (1,2)
os-nosdn-nofeature-noha	Fuel@x86	@Michael Polenchuk	Y	Y	Y	Y
os-nosdn-nofeature-ha	Fuel@x86 Fuel@aarch64	@Michael Polenchuk	Y	Y	Y	Y
os-odl-nofeature-noha	Fuel@x86	@Michael Polenchuk	Y	Y	Y	Y
os-odl-nofeature-ha	Fuel@x86 Fuel@aarch64	@Michael Polenchuk	Y	Y	Y	Y
os-nosdn-ovs-noha	Fuel@x86	@Michael Polenchuk	Y	Y	Y	Y
os-nosdn-ovs-ha	Fuel@x86 Fuel@aarch64	@Michael Polenchuk	Y	Y	Y	Y
os-ovn-nofeature-noha	Fuel@x86	@Michael Polenchuk	Y	Y	Y	Y
os-ovn-nofeature-ha	Fuel@aarch64	@Michael Polenchuk	Y	N	Y	Y

Рис. 5: Сценарии OPNFV

[6].

Установка и настройка VIM – нетривиальная задача, требующая опыта работы с технологиями телекоммуникаций и навыков системного администрирования. Ввиду того, что инфраструктуры NFV может содержать виртуальные машины и ее настройка отчасти выполняется с помощью установки операционных систем по сети, было принято решение изучить технологии Kernel-based Virtual Machine и Preboot eXecution Environment.

3.2.2. Kernel-based Virtual Machine

Выбранный нами установщик в качестве плоскости виртуализации NFVI поддерживает гипервизор KVM, обеспечивающий виртуализацию в операционных системах на базе ядра Linux.

В процессе изучения технологии виртуализации была поставлена задача установки операционной системы на виртуальную машину, созданную с помощью гипервизора KVM.

Настройка окружения заключается в установке минимальной сбор-

ки операционной системы CentOS 7 и конфигурирования BIOS с целью включения поддержки процессором аппаратной виртуализации.

Необходимые пакеты:

- `qemu-kvm` – гипервизор;
- `libvirt` – библиотека управления виртуализацией;
- `virt-install` – управление виртуальными машинами;
- `bridge-utils` – для работы с сетевым мостом;
- `tiger-vnc` – для запуска VNC-сервера.

Виртуальные машины для доступа к внешней сети используют Network Address Translation, где шлюзом является физический интерфейс хоста. Для установки операционной системы на виртуальную машину необходимо объединить хост и виртуальную машину в одну сеть с помощью сетевого моста, чтобы потом подключиться к ней с помощью клиента VNC и завершить настройку.

Сетевой мост настраивается путем создания конфигурационного файла сетевого моста (`ifcfg-br*`) и добавления его названия в конфигурационный файл сетевой карты (`ifcfg-*`).

Для того, чтобы к виртуальной машине можно было обращаться из сети, нужно включить перенаправление пакетов между сетевым интерфейсом виртуальной машины и физическим сетевым интерфейсом хоста, раскомментировав строчку `"net.ipv4.ip_forward=1"` в файле `sysctl.conf`.

По умолчанию в операционной системе CentOS 7 закрыты порты 5900-5905, необходимые для подключения клиентов VNC. Требуется открыть эти порты для TCP соединения, сконфигурировав межсетевой экран операционной системы командой `firewall-cmd --permanent --add-port=5900-5905/tcp`.

В качестве сетевого интерфейса виртуальной машины выступает временно созданный и настроенный к работе сетевой мост, то есть указывается флаг `--network=bridge:br*`.

После ввода команды `virt-install` виртуальная машина будет находиться в состоянии ожидания завершения установки операционной системы. На этом этапе требуется подключиться к ней с помощью VNC клиента и завершить установку.

3.2.3. Preboot eXecution Environment

PXE – среда для загрузки компьютера с помощью сетевой карты без использования локальных носителей данных. Эта технология используется VIM для установки операционных систем на виртуальные или физические машины, входящие в инфраструктуру NFV.

В процессе изучения данной технологии была поставлена задача установки по сети операционной системы CentOS 7 с желаемой конфигурацией на виртуальную машину с помощью сценариев установки `kickstart`.

Суть работы классического PXE-сервера: настроенный на загрузку по сети клиент отправляет в сеть широковещательный запрос на получение сетевой конфигурации; DHCP сервер отсылает конфигурацию клиенту, с помощью которой последний узнает свой IP адрес в сети, а также IP адрес TFTP сервера; зная IP адрес TFTP сервера клиент с помощью протокола TFTP загружает по сети PXE загрузчик, временную файловую систему, а также ядро операционной системы и передает ему управление;

Kickstart PXE сервер отличается от классического тем, что после того, как ядро получило управление, начинается выполнение сценария установки.

В рамках нашей задачи TFTP и DHCP серверы запускались на одной виртуальной машине.

Необходимые пакеты:

- `tftp-server` – TFTP сервер;
- `vsftpd` – FTP сервер
- `xinetd` – служба, управляющая сетевыми соединениями;

- `syslinux` – набор загрузчиков.

Настройку PXE сервера условно можно разделить на четыре части: настройка TFTP сервера; настройка DHCP сервера; настройка FTP сервера; настройка сценариев установки kickstart.

Держать программы в оперативной памяти сервера, когда они не используются – не рациональное решение, поэтому используется сервис `xinetd`, позволяющий запускать другие сервисы по запросу клиентов.

Для того, чтобы TFTP сервер запускался при обращении клиентов, нужно разрешить в конфигурации `xinetd` `tftp` в качестве сервиса по требованию, изменив в файле `/etc/xinet.d/tftp` строчку `"disable = yes"` на `"disable = no"`.

В корневую директорию `tftp` сервера нужно перенести файлы, необходимые для запуска операционной системы на машине клиента: `pxelinux.0` – системный загрузчик; `vmlinuz` – ядро операционной системы; `initrd.img` – временная файловая система, используемая ядром Linux при начальной загрузке. Загрузчик `pxelinux.0` доступен после установки `syslinux` и располагается в директории `/usr/lib/syslinux/`; `vmlinuz` и `initrd.img` находятся в директории `/mnt/images/pxeboot/` относительно смонтированного образа предполагаемой для установки операционной системы.

Настройка DHCP сервера заключается в редактировании конфигурационного файла `dhcpd.conf`, находящегося в директории `/etc/`. В нашем случае конфигурационный файл выглядел следующим образом:

```
ddns-update-style interim;
ignore client-updates;

subnet 192.168.0.0 netmask 255.255.255.0 {
    option routers 192.168.0.1;
    option subnet-mask 255.255.255.0;
    option domain-name "local";
    option domain-name-servers 8.8.8.8;
    option time-offset 10800; # GMT+3
```

```

    range dynamic-bootp 192.168.0.4 192.168.2.254;
    default-lease-time 21600;
    max-lease-time 43200;
    next-server 192.168.0.2;
    filename 'pxelinux.0';
}

```

Где **routers** – шлюз по умолчанию; **verb|subnet-mask|** – маска подсети; **domain-name** – домен, который будет добавляться при обращении клиента к ресурсу по имени, которое не имеет домена; **domain-name-servers** – DNS серверы; **time-offset** – разница во времени с Гринвичем; **range dynamic** – диапазон выдаваемых IP адресов; **default-lease-time** – время аренды IP адреса по умолчанию; **max-lease-time** – максимальное время аренды IP адреса; **next-server** – адрес TFTP сервера; **filename** – путь к файлу PXE загрузчика.

В процессе установки операционных систем на клиентские машины инсталляторам понадобится доступ к дистрибутиву операционной системы, который удобно располагать на FTP сервере. Для этого в корневом каталоге FTP сервера `/var/ftp/` создаются директории для дистрибутивов операционных систем, а затем образы дистрибутивов монтируются в эти директории.

Чтобы операционная система устанавливалась полностью автоматически, необходимо создать т.н kickstart файл, который содержит в определенном формате ответы на все вопросы инсталлятора, такие как путь до дистрибутива, язык системы, тип клавиатуры, настройки сетевой карты и так далее. Ниже приведен существенный фрагмент нашего файла сценария установки, указывающий тип установки (чистая установка), расположение дистрибутива операционной системы и установочный номер.

```

install
url --url ftp://192.168.0.2/Centos/
key 1

```

Этот файл нужно сохранить в корневом каталоге FTP сервера.

Заключительной стадией настройки kickstart PXE сервера является настройка загрузчика путем добавления и редактирования конфигурационного файла `pxelinux.cfg` в корневой директории TFTP сервера. В нашем случае он выглядел следующим образом:

```
default linux
prompt 1
timeout 100
label linux
kernel vmlinuz
append initrd=initrd.img ramdisk_size=9216\
noapic acpi=offo ks=ftp://192.168.0.2/ks.cfg
```

Где `default` – метка записи, загружаемая по умолчанию; `prompt` – флаг, отвечающий за возможность выбора операционной системы, `timeout` – время, после которого будет выбрана метка по умолчанию; `label` – метка конфигурации; `kernel` – путь к файлу ядра относительно корневой директории tftp сервера; `append` – опции, передаваемые ядру, в том числе временная файловая система `initrd` и `kickstart` файл, который завершит установку в автоматическом режиме.

Заключение

Ввиду того, что в распоряжении не оказалось нужного оборудования, комплексная установка и настройка программного окружения для стенда не могла быть закончена. В ходе работ были достигнуты следующие результаты:

- было подобрано необходимое оборудование;
- был сделан выбор готового решения на основании сравнительной таблицы;
- был установлен и настроен гипервизор KVM;
- была установлена операционная система CentOS 7 с нужной конфигурацией с помощью PXE.

Список литературы

- [1] [www.etsi.org](https://osm.etsi.org/). — 2019. — URL: <https://osm.etsi.org/> (дата обращения: 20.05.2019).
- [2] Building Open Source NFV // www.opnfv.org. — 2019. — URL: <https://www.opnfv.org/end-users/building-open-source-nfv> (дата обращения: 10.10.2018).
- [3] Gambia scenario status // wiki.opnfv.org. — 2019. — URL: <https://wiki.opnfv.org/display/SWREL/Gambia+Scenario+Status> (дата обращения: 3.12.2018).
- [4] Network Functions Virtualisation(NFV); Architectural Framework // www.etsi.org. — 2014. — URL: https://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.02.01_60/gs_NFV002v010201p.pdf (дата обращения: 17.10.2018).
- [5] OPNFV Fuel Installation Instruction // opnfv-fuel.readthedocs.io. — 2018. — URL: <https://opnfv-fuel.readthedocs.io/en/stable-gambia/release/installation/index.html#fuel-installation> (дата обращения: 22.02.2019).
- [6] OPNFV Installation instructions (Apex) // opnfv-apex.readthedocs.io. — 2018. — URL: <https://opnfv-apex.readthedocs.io/en/stable-gambia/release/installation/index.html> (дата обращения: 22.02.2019).
- [7] Rajendra Chayapathi Syed F. Hassan Paresh Shah. The Journey to Network Functions Virtualization (NFV) Era // [informIt.com](http://informit.com). — 2017. — URL: <http://www.informit.com/articles/article.aspx?p=2755705&seqNum=2> (дата обращения: 29.10.2018).
- [8] VIMs // osm.etsi.org. — 2019. — URL: <https://osm.etsi.org/wikipub/index.php/VIMs> (дата обращения: 16.05.2019).

- [9] VNFs // osm.etsi.org. — 2019. — URL: <https://osm.etsi.org/wikipub/index.php/VNFs> (дата обращения: 16.05.2019).
- [10] vCloud NFV // www.vmware.com. — URL: <https://www.vmware.com/ru/products/network-functions-virtualization.html> (дата обращения: 10.06.2019).
- [11] Межсетевые экраны нового поколения Cisco ASA серии 5500-X // www.cisco.com. — 2018. — URL: https://www.cisco.com/c/ru_ru/products/security/asa-5500-series-next-generation-firewalls/index.html (дата обращения: 10.05.2019).
- [12] ЧТО ТАКОЕ DPI? // wiki.merionet.ru. — 2019. — URL: <https://wiki.merionet.ru/seti/20/что-такое-технология-dpi/> (дата обращения: 10.05.2019).