

Реализация тестового стенда Network Function Virtualization

результаты

Кижнеров Павел Александрович
научный руководитель: ст. преп И.В Зеленчук

СПбГУ

22 мая 2019 г.

Network Function Virtualization - концепция, предлагающая виртуализировать аппаратные сетевые устройства посредством предоставления возможности сетевым функциям исполняться программными модулями.

Актуальность обусловлена стремлением крупных компаний избавиться от физических устройств, так как каждое из них:

- Занимает место в помещении
- Потребляет электроэнергию
- Нуждается в климатических ресурсах
- Имеет свойство морально устаревать
- Зачастую сильно зависит от вендора

Цель – начать создание тестового стенда NFV

Были поставлены следующие задачи:

- Изучить предметную область
- Провести сравнительный анализ существующих решений
- Начать реализацию одного из решений

Основа архитектуры – VIM, VNFM и NFVO

VIM (Virtual Infrastructure Manager) –

контролирует и управляет вычислениями, памятью и сетевыми ресурсами. Собирает данные о производительности

VNFM (Virtual Network Function Manager) –

запускает, обслуживает, и прекращает работы виртуализированных функций

NFVO (NFV Orchestrator) – управляет и

администрирует виртуализированные сетевые функции через VIM и VNFM

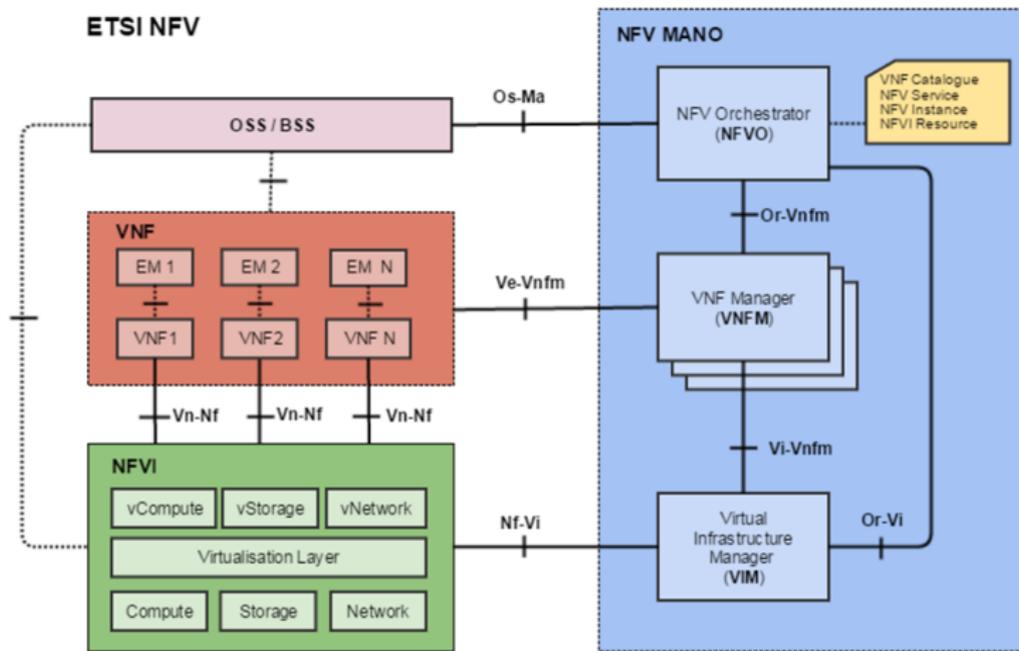


Рис. 1: Архитектура NFV, разработанная ETSI

Существующие решения: VIM

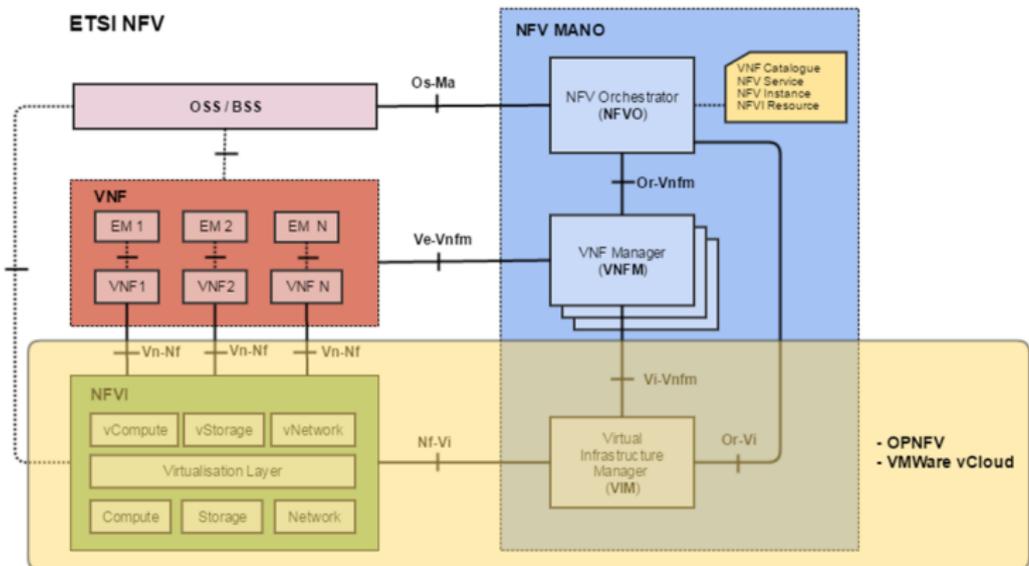


Рис. 2: решения для VIM

vCloud NFV – проприетарная интегрированная платформа от VMWare

OPNFV – open-source интегрированная платформа от Linux Foundation

Для реализации VIM была выбрана OPNFV в силу проприетарности vCloud NFV

Существующие решения: NFVO

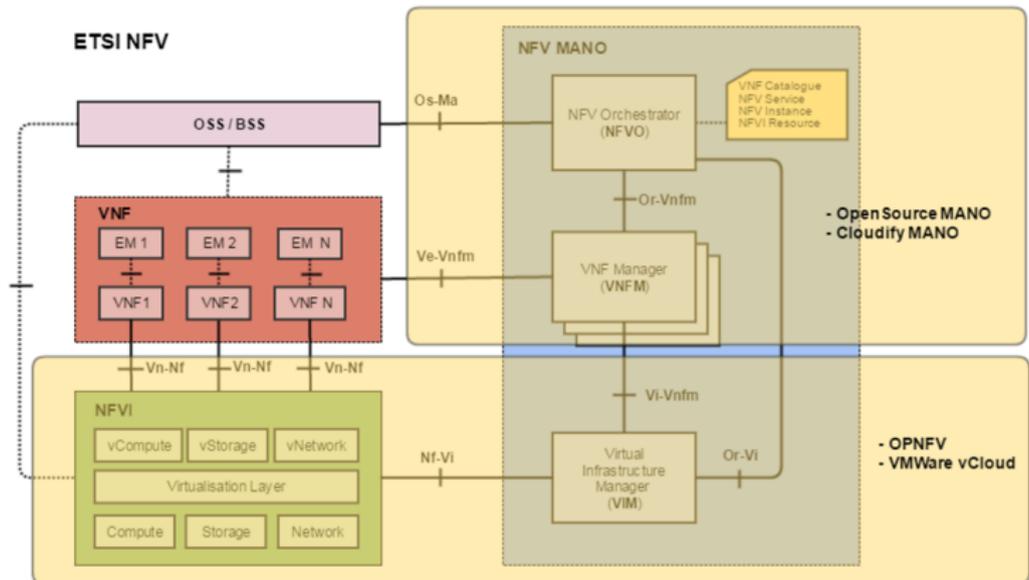


Рис. 3: решения для NFVO

- Open Source MANO
- ONAP
- OpenBaton
- Sonata

Предварительно – OSM, так как разрабатывается самим ETSI

- Установка и настройка окружения
- Конфигурация сети
- Конфигурация программного обеспечения
OPNFV: PDF (POD Descriptor File), IDF (Installer
Descriptor File)
- Клонирование репозитория релиза Gambia и
запуск `deploy.sh`

Gambia 7.2 – последний стабильный релиз OPNFV.

Установщики:

- Apex
- Fuel
- Compass4NFV

Apex – относительно легкая настройка сети, но есть неопределенности с конфигурационными файлами

Fuel – сложная настройка сети, но более ясная документация

Compass4NFV – скудная документация, но более производительный результат

В качестве установщика был выбран Fuel, так как разницы с Arех на небольшой сети быть не должно, но зато есть подробная инструкция.

Предполагается, что имеется опыт работы с PXE и KVM, поэтому было принято решение выполнить следующие задания:

- С помощью PXE установить ОС на виртуальную машину
- Поставить на хост несколько ОС, используя KVM

Сценарий – установщик + конфигурация +
upstream компоненты
Для наших нужд требуется тривиальный вариант.

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-odl-nofeature-ha	Compass	@Harry Huang	Y	Y	Y	Y
K8-nosdn-stor4nfv-ha	Compass	@Harry Huang	Y	Y	Y	Y
os-nosdn-bar-ha	Compass	@Harry Huang	Y	Y	Y	Y
os-nosdn-stor4nfv-ha	Compass	@Harry Huang	N	Y	Y	Y
os-nosdn-calipso-noha	Apex	@Koren Lev	N	Y	Y	Y
os-nosdn-nofeature-noha	Fuel@x86	@Michael Polenчук	Y	Y	Y	Y
os-nosdn-nofeature-ha	Fuel@x86 Fuel@aarch64	@Michael Polenчук	Y	Y	Y	Y

Рис. 4: сценарии OPNFV

Возникла проблема удаленного доступа к стенду из за NAT.

Решена с помощью настройки сервера VPN и установки ПО на роутер для поддержки VPN туннелирования.

В качестве протокола VPN был выбран OPNVPN ради экономия времени, так как уже имеется опыт работы.

ПО для роутера – DD-WRT, так как на слуху, а выбор прошивки не является основной целью данной курсовой работы.

Достигли:

- Изучена архитектура NFV
- Проанализированы существующие решения
- Начата реализация инфраструктурного менеджера
 - Изучен гипервизор KVM
 - Подготовлено ПО для удаленного доступа к тестовому стенду