Конкурсное задание

ИТ Сетевое и системное

администрирование

WSC2022_TP39_EN_Mod_C

Модуль C – Программируемость и автоматизация инфраструктуры

Представленный:
Thushjandan Ponnudurai CH
Jun Tian CN

Anthony Lebarbanchon FR
Hamed Kargarzadeh IR
John Leong SG
Janos Csoke HU
Gen Le EE

Содержание

Введение в тестовый проект

Ниже приводится список разделов или информации, которые должны быть включены во все предложения по конкурсным заданиям, представляемые в WorldSkills.

- Содержание, включая список всех документов, рисунков и фотографий, составляющих Конкурсное задание.
- Введение/обзор
- Краткое описание проекта и задач
- Инструкция для участника
- Оборудование, машины, установки и материалы, необходимые для выполнения Конкурсного задания.
- Схема выставления оценок (включая критерии оценки)
- Другое

Введение

Это предложение по тестовому заданию состоит из следующей документации/файлов:

- WSC2022_TP39_EN_Mod_C.docx
- hosts
- users.csv
- .vault_pass
- customers.json

Точная и актуальная документация всегда была проблемой в ИТ. Когда несколько инженеров работают над одной и той же системой, трудно отследить, кто что изменил. Корпорация Applix решила решить эту проблему и наняла вас для модернизации, укрепления и расширения их инфраструктуры.

Описание проекта и задач

Вы будете переносить виртуальные машины в инфраструктуру как код (IaC) и упростите процесс создания новых сервисов. У вас есть доступ к виртуальным машинам разработки (DEV-LIN и DEV-WIN). Эти виртуальные машины можно использовать для разработки и тестирования вашей работы.

Вход для всех ВМ и Устройств:

Имя пользователя Linux: root / appadmin
Имя пользователя Windows: Administrator / appadmin
Имя пользователя Cisco: appadmin
Имя пользователя CML: admin

Пароль: Incheon-2022

Все виртуальные машины и устройства подключены к сети управления (10.22.0.0/24) и имеют статически настроенный IP-адрес. Все IP-адреса в сетевой таблице не будут изменены для маркировки. Сеть управления будет использоваться для настройки различных хостов. Вы можете войти в систему, используя имя пользователя и пароль через SSH, HTTPS или WinRM.

Вы можете установить любые дополнительные необходимые пакеты и функции на виртуальные машины.

Инструкция для участника

Часть 1. Линукс

Используйте Ansible для настройки хостов Linux LIN[1-5] с HOST VM. Существует предварительно сконфигурированный файл hosts, расположенный в /etc/ansible/hosts. Не изменяйте этот файл. Перед оценкой все виртуальные машины Linux будут сброшены в исходное состояние, а виртуальные машины Linux будут случайным образом удалены и добавлены в разные группы в файле hosts. Для маркировки все плейбуки будут запущены по порядку с помощью команды «ansible-playbook playbookname.yml» в каталоге /data/ansible/linux. Такие переменные, как «имя хоста», «веб-порт» и «веб-цвет» в /etc/ansible/host могут быть изменены для маркировки.

На всех хостах LIN подключены Debian 11.3 BD ISO 1-4.

Общее

- Создайте каталог /data/ansible/linux
  - Все плейбуки должны находиться в корне этого каталога.
  - Вы можете создавать папки/файлы в этом каталоге для запуска плейбуков.
  - Запустите репозиторий git и регулярно вносите свои изменения, чтобы иметь рабочую версию в git.
  - Все задачи должны иметь состояние «OK» или «Пропущено» после повторного запуска плейбуков.
  - Убедитесь, что имя пользователя и пароль ansible зашифрованы с помощью файла /etc/ansible/.vault_pass и автоматически расшифровываются при запуске плейбуков.

Имя хоста

- Создайте playbook с именем 1-hostname.yml для настройки имени хоста.
  - Все хосты должны получить имя хоста на основе переменной «hostname» в файле /etc/ansible/hosts.

IP-таблицы

- Создайте playbook с именем 2-iptables.yml для фильтрации входящего трафика на всех хостах LIN.
  - Входящие пакеты должны отбрасываться по умолчанию
  - Разрешить минимальный трафик для работы сервисов
  - Разрешить трафик SSH и ICMP от виртуальных машин HOST и DEV к узлам LIN.
  - Убедитесь, что iptables сохраняется после перезагрузки

DNS

- Создайте playbook с именем 3-dns-server.yml для настройки двух или более DNS-серверов.
- Установите службу DNS на все хосты в группе «dns»
  - Первый хост в группе «dns» должен быть авторитетным для домена «applix.com».
  - Все остальные хосты в этой группе должны быть подчиненными DNS-серверами для домена «applix.com».
  - Убедитесь, что все хосты в /etc/ansible/hosts имеют запись A для <hostname>.applix.com.
  - intranet.applix.com должен разрешаться в 10.22.0.51
- Создайте плейбук с именем 4-dns-client.yml.
  - Убедитесь, что все LIN-серверы и HOST используют первый DNS-хост в качестве основного DNS-сервера, а все подчиненные DNS-серверы — в качестве вторичного сервера имен.
  - Настройте DNS-суффикс «applix.com»

Интернет

- Создайте playbook с именем 5-web-server.yml для настройки двух или более веб-серверов.
- Установить веб-сервис на все хосты в группе «web»
  - Локальный веб-сайт должен прослушивать переменную порта «webport» в /etc/ansible/hosts.
  - Отображение следующего содержимого с цветом текста на основе переменной «webcolor» в файле /etc/ansible/hosts.

    «Hello from <hostname> !»
  - Создайте виртуальный хост, прослушивающий порт 8081, с именем «intranet.applix.com», отображающий следующее содержимое.

    “Welcome to the intranet of Applix”
    “This site was served by <hostname>”

High availability intranet

- Создайте сценарий под названием 6-ha-intranet.yml для настройки двух или более серверов высокой доступности.
- Установите keepalived и HAProxy на все хосты в группе «ha»
  - Настройте 10.22.0.51 как плавающий IP-адрес и используйте последний хост в группе «ha» в качестве мастера VRRP.
  - Если мастер выйдет из строя, второй последний сервер в группе ha должен взять на себя управление и т.д. (приоритет в обратном порядке)
  - Использовать аутентификацию по паролю
  - Настройте HAProxy для балансировки нагрузки «http://intranet.applix.com» между всеми доступными веб-серверами с использованием циклического перебора.
    - Добавьте заголовок «x-haproxy-host» с именем текущего хоста HAProxy.

Пользователи

- Создайте playbook с именем 7-users.yml для импорта пользователей на все хосты LIN.
  - Импортируйте пользователей из /etc/ansible/users.csv на всех хостах LIN.
  - Убедитесь, что этот пароль не изменен, если уже существует пользователь с таким же именем пользователя и UID.

Часть 2. Windows

Используйте Ansible для настройки хостов Windows WIN[1-5] с HOST VM. Существует предварительно сконфигурированный файл hosts, расположенный в /etc/ansible/hosts. Не изменяйте этот файл. Перед оценкой все виртуальные машины Windows будут сброшены в исходное состояние, а виртуальные машины Windows будут случайным образом перемещены в разные группы в файле hosts. Для оценки все плейбуки будут запущены по порядку с помощью команды «ansible-playbook playbookname.yml» в каталоге /data/ansible/windows. Такие переменные, как «hostname», «RootCAPriv» в /etc/ansible/host, могут быть изменены.

Общее

- Создайте каталог /data/ansible/windows
  - Все плейбуки должны находиться в корне этого каталога.
  - Вы можете создавать папки/файлы в этом каталоге для запуска плейбуков.
  - Запустите репозиторий git и регулярно вносите свои изменения, чтобы иметь рабочую версию в git.
  - Убедитесь, что доступное имя пользователя, пароль и секреты сертификата зашифрованы и автоматически расшифрованы при запуске плейбуков.

Имя хоста

- Создайте playbook с именем 1-hostname.yml для настройки имени хоста.
  - Все хосты должны получить имя хоста на основе переменной «hostname» в файле /etc/ansible/hosts.

Корневой сертификат

- Создайте playbook с именем 2-cert.yml для настройки сертификатов.
  - Создайте корневой сертификат со следующими свойствами
    - Common Name = Applix Root CA
    - Organization Name = Applix Corporation
    - Country Code = KR
  - Распространите корневой сертификат на все хосты Windows как доверенный корневой ЦС в хранилище компьютера.
    - Убедитесь, что соответствующие хосты Windows на основе переменной «RootCAPriv» в файле /etc/ansible/hosts имеют сертификат в хранилище персонального компьютера с закрытым ключом.

Безопасность и ведение журнала

- Создайте playbook с именем 3-sec-log.yml для настройки параметров безопасности.
  - Остановите и отключите службу удаленного рабочего стола на всех хостах Windows.

- Создайте и запланируйте задачу с именем LogUptime для добавления текущего времени безотказной работы в C:\uptime.txt в формате, указанном ниже, каждые 30 секунд

```
XX days XX hours XX minutes XX seconds
XX days XX hours XX minutes XX seconds
XX days XX hours XX minutes XX seconds
```

Инструменты управления

- Создайте playbook с именем 4-tools.yml для установки инструментов Windows.
  - Установите клиент telnet на серверы с установкой Windows с графическим интерфейсом.

Развертывания клиентов

- Создайте playbook с именем 5-environment.yml для установки клиентской среды.
  - Настройте все серверы в группе «dc» как DNS-серверы на всех хостах WIN.
  - Настройте все серверы в группе «dc» как контроллеры домена
    - Использовать customers.com в качестве доменного имени
    - Используйте «Incheon-2022» в качестве надежного пароля
  - Присоединить все хосты Windows, не относящиеся к DC, к домену
  - Установите роль IIS на все серверы в группе «iis»
- Создайте playbook с именем 6-customers.yml для развертывания клиентских веб-сред.
  - Для каждого клиента в файле /etc/ansible/customers.json
    - Создайте OU на основе атрибута «name»
    - Создайте пользователя AD на основе атрибута «username» и «password» в этом подразделении.
    - Создайте запись DNS для domain_prefix.customers.com, указывающую на случайный сервер IIS, и используйте domain_prefix в качестве рандомизации.
    - Создайте виртуальный хост, прослушивающий порт 80 на выбранном сервере IIS, который отображает атрибут «message».

Часть 3. Сеть

Используйте Ansible для настройки маршрутизаторов Cisco RTR[1-8] с HOST VM. Существует предварительно сконфигурированный файл hosts, расположенный в /etc/ansible/hosts. Не изменяйте этот файл. Перед оценкой все маршрутизаторы Cisco будут очищены загрузочной конфигурацией. Для оценки все плейбуки будут запущены по порядку с помощью команды «ansible-playbook playbookname.yml» в каталоге /data/ansible/cisco. Такие переменные, как «bgp_as_internal», «bgp_as_external» и «loop_net» в /etc/ansible/host, могут быть изменены.

Общее

- Создайте каталог /data/ansible/cisco
  - Все плейбуки должны находиться в корне этого каталога.
  - Вы можете создавать папки/файлы в этом каталоге для запуска плейбуков.
  - Запустите репозиторий git и регулярно вносите свои изменения, чтобы иметь рабочую версию в git.
  - Все задачи должны иметь состояние «OK» или «Пропущено» после повторного запуска плейбуков.
  - Убедитесь, что доступное имя пользователя, пароль и секрет пиринга eBGP зашифрованы и автоматически расшифрованы при запуске плейбуков.

Имя хоста

- Создайте playbook с именем 1-hostname.yml для настройки имени хоста.
  - Все маршрутизаторы должны получить имя хоста на основе переменной hostname в файле /etc/ansible/hosts.

Безопасность

- Создайте playbook с именем 2-security.yml для настройки доступа к интерфейсу управления.
  - Разрешить трафик SSH, ICMP и HTTPS только от виртуальных машин HOST и DEV к Gig1

Петлевые интерфейсы

- Создайте playbook с именем 3-loopback.yml для настройки петлевых интерфейсов.
  - Все маршрутизаторы должны получать один IP-адрес из подсети «loop_net» в /etc/ansible/hosts.
  - Используйте интерфейс loopback 0 и длину префикса /32.
  - Используйте описание «Ansible — Routing Loopback Interface»
  - Первый маршрутизатор в /etc/ansible/hosts должен получить первый доступный IP-адрес «loop-net» и т. д.

Внутренняя маршрутизация

- Создайте playbook с именем 4-igp.yml для настройки внутренней динамической маршрутизации.
  - Вы можете использовать любой современный протокол маршрутизации IGP.
  - Все внутренние и пограничные маршрутизаторы должны участвовать в процессе маршрутизации.
  - Объявить петлевой интерфейс 0 и сеть LAN (10.0.0.0/24)

Внутренняя BGP-маршрутизация

- Создайте playbook с именем 5-ibgp-lan.yml для настройки внутренней динамической маршрутизации.
  - Все внутренние (internal) и пограничные (edge) маршрутизаторы должны участвовать в полносвязной внутренней AS на основе переменной «bgp_as_internal» в файле /etc/ansible/hosts.
  - Используйте loopback 0 для пиринга
  - Объявите loopback 0 и сеть LAN (10.0.0.0/24) на каждом маршрутизаторе
- Создайте playbook с именем 6-ibgp-wan.yml для настройки внутренней динамической маршрутизации внешних маршрутизаторов.
  - Все внешние (external) маршрутизаторы должны участвовать в полносвязной внутренней AS на основе переменной «bgp_as_external» в файле /etc/ansible/hosts.
  - Объявите loopback 1 и 99 на каждом маршрутизаторе

Пиринг eBGP

- Создайте playbook с именем 7-ebgp-peering.yml для настройки пиринга eBGP.
  - Создайте полноценную сеть пиринга eBGP между пограничными и внешними маршрутизаторами.
  - Используйте аутентификацию по паролю со строкой «EBGP$Secret»
  - Убедитесь, что пиринги eBGP сбалансированы по нагрузке.

Тест подключения

- Убедитесь, что все внутренние (internal) и пограничные (edge) маршрутизаторы могут пинговать петлевые интерфейсы loopback внешних маршрутизаторов.

Бэкап конфигурации

- Создайте пустой репозиторий git в /data/cisco/backup на виртуальной машине HOST.
- Создайте playbook с именем 8-backup.yml для резервного копирования текущей конфигурации и отслеживания изменений.
  - Назовите файлы <RTR ID>.cfg в папке резервной копии.
  - Создайте коммит git с комментарием «Automated Ansible backup YYYY-mm-dd hh:mm:ss»
  - Коммит должен быть создан только тогда, когда какой-либо файл конфигурации изменился.

Часть 4. API

Создайте API Python для запроса данных об инфраструктуре. API должен размещаться на VM **HOST ,** а все файлы API должны располагаться в папке /data/api. Вы можете свободно использовать любые модули или фреймворки, доступные в ISO-образах Debian.

- API должен прослушивать HTTPS через порт 443 на IP 10.22.0.50.
  - Убедитесь, что ваш API доступен по адресу https://api.applix.com/ с виртуальных машин HOST и DEV.
  - В браузерах Edge и Firefox на виртуальных машинах DEV не должно быть предупреждений о сертификатах.
- Создайте сервис systemd под названием «applix-api».
  - Служба должна запускаться при загрузке
  - API должен управляться с помощью команд systemd «start, restart, stop».
- Общие сведения о конечных точках (endpoint)
  - Убедитесь, что API возвращает информацию о формате контента
  - Аутентификация токена выполняется с использованием заголовка «токен» в HTTP-запросе.

- **Endpoint /stats (GET)**
  - Эта конечная точка должна возвращать статистику использования с HOST VM в формате JSON.

```
{
        "ram_free": "X",
        "uptime": "XX:XX:XX:XX,
        "api_pid": "X",
        "api_prio": "X",
        "total_proc": "X"
}
```

  - ram_free = объем свободной памяти в мегабайтах
  - uptime = время безотказной работы в формате дд:чч:мм:сс
  - api_pid = pid текущего процесса API
  - api_prio = текущий приоритет ЦП процесса API
  - total_proc = общее количество процессов, запущенных в системе

- **Endpoint /login (POST)**
  - Конечная точка должна использовать обычную аутентификацию и должна сверять имя пользователя и пароль с файлом /etc/ansible/users.csv. Если имя пользователя и пароль верны, верните 32-символьный буквенно-цифровой токен в формате JSON.

    ```
    {
            "токен": "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
    }
    ```
  - Сохраните токен в локальной базе данных Redis и убедитесь, что токен удален через 90 секунд.
  - После 3 неудачных попыток входа исходный IP-адрес клиента должен быть заблокирован для доступа к API на 1 минуту.

- **Endpoint /logout (POST)**
  - Конечная точка должна использоваться для выхода пользователя из системы.
  - Этой конечной точке требуется токен для аутентификации.
  - Нет данных, кроме токена, отправленного на конечную точку

- **Endpoint /whoami (GET)**
  - Этой конечной точке требуется токен для аутентификации.
  - Вернуть имя пользователя в формате JSON

    ```
    {
            "username": "X"
    }
    ```

- **Endpoint /customers/ (PUT)**
  - Эта конечная точка используется для развертывания новых клиентов.
  - Для этой конечной точки требуются следующие параметры в теле HTTP в виде JSON.
    - name = имя клиента
    - domain_prefix = префикс для домена
    - username = имя пользователя Active Directory
    - password = пароль Active Directory
    - message = текст, отображаемый на веб-сайте
  - Эта конечная точка должна добавить нового клиента в файл /etc/ansible/customers.json.
    - Убедитесь, что имя, префикс домена и имя пользователя уникальны.
    - Если это не так, вернуть ошибку HTTP Forbidden.
  - После добавления клиента плейбук 6-customers.yml должен запускаться автоматически.

- **Endpoint /network/devices/stats (GET)**
  - Эта конечная точка должна возвращать статистику всех активных (восходящих) интерфейсов от сетевых устройств cisco в формате JSON.
  - Запрашивайте данные с маршрутизаторов с помощью RESTCONF (используйте одноадресные пакеты для pkts-in/pkts-out)

```
{
       "devices": [{
              "hostname": "X",
              "management_ip": "XXX.XXX.XXX",
              "ios_version": "XX.XX",
              "interfaces": [
              {
                 "name": "GigabitEthernet1",
                 "mac": "XX:XX:XX:XX:XX:XX",
                 "ip": "XXX.XXX.XXX.XXX",
                 "pkts-in": "XX",
                 "pkts-out": "XX"
              },
              {
                 "name": "Loopback0",
                 "mac": "XX:XX:XX:XX:XX:XX",
                 "ip": "XXX.XXX.XXX.XXX",
                 "pkts-in": "XX",
                 "pkts-out": "XX"
              }, ....
         ]
       }, {
              "hostname": "X",
              "management_ip": "XXX.XXX.XXX.XXX",
              "ios_version": "X", ....
       }, .... ]
}
```

  - Должна быть возможность фильтровать выходные данные с помощью параметра запроса management_ip=XXX.XXX.XXX.XXX.
    - Если устройство не найдено, API должен вернуть ошибку HTTP not found.

- **Endpoint /network/devices/bgp (GET)**
  - Эта конечная точка должна возвращать активных соседей BGP всех сетевых устройств cisco в формате JSON.
  - Запросите данные с маршрутизаторов с помощью RESTCONF.

```
{
        "devices": [{
                "management_ip": "XXX.XXX.XXX.XXX",
                "bgp_neighbors": [
                {
                   "id": "XXX.XXX.XXX.XXX",
                   "as": "XXX",
                   "uptime": "hh:mm:ss",
                   "keepalives-sent": "X",
                   "keepalives-received": "X"
                },
                {
                   "id": "XXX.XXX.XXX.XXX",
                   "as": "XXX",
                   "uptime": "hh:mm:ss",
                   "keepalives-sent": "X",
                   "keepalives-received": "X"
                }, ....
          ]
        }, {
                "management_ip": "XXX.XXX.XXX.XXX",
        }, .... ]
}
```

  - Должна быть возможность фильтровать выходные данные с помощью параметра запроса management_ip=XXX.XXX.XXX.XXX.
    - Если устройство не найдено, API должен вернуть ошибку http not found.

- **Endpoint /network/devices/routes (GET)**
  - Эта конечная точка должна возвращать таблицу маршрутизации всех сетевых устройств cisco в формате JSON.
  - Запросите данные с маршрутизаторов с помощью RESTCONF.

```
{
        "devices": [{
                "management_ip": "XXX.XXX.XXX.XXX",
                "routes": [
                {
                   "subnet": "XXX.XXX.XXX.XXX/XX",
                   "type": "<bgp / eigrp / ospf / local / connected>",
                   "ad": "X",
                   "next-hop": "XXX.XXX.XXX.XXX",
                   "interface": "X"
                },
                {
                   "subnet": "XXX.XXX.XXX.XXX/XX",
                   "type": "<bgp / eigrp / ospf / local / connected>",
                   "ad": "X",
                   "next-hop": "XXX.XXX.XXX.XXX",
                   "interface": "X"
                }, ....
          ]
        }, {
                "management_ip": "XXX.XXX.XXX.XXX",
        }, .... ]
}
```

  - Должна быть возможность фильтровать выходные данные с помощью параметра запроса management_ip=XXX.XXX.XXX.XXX.
    - Если устройство не найдено, API должен вернуть ошибку http not found.

- **Endpoint /network/loopback (PUT/DELETE)**
  - Эта конечная точка должна создавать и удалять петлевой интерфейс на маршрутизаторе cisco с помощью RESTCONF.
  - Для этой конечной точки требуется действительный токен проверки подлинности.
  - Для этой конечной точки требуются следующие параметры в теле HTTP в виде JSON.
    - number = номер петлевого интерфейса
    - ip = IP-адрес петлевого интерфейса в формате XXX.XXX.XXX.XXX/XX
    - desc = Описание
    - management_ip = IP-адрес сетевого коммутатора для настройки
  - Существующий петлевой интерфейс должен быть перезаписан, если уже существует интерфейс с таким же номером

- **Endpoint /healthz (GET)**
  - Конечная точка должна возвращать код состояния HTTP 2xx, если служба API работает правильно, и 5xx, если есть какие-либо проблемы.
  - Конечная точка должна возвращать количество неудачных и успешных запросов за последние 5 минут и за последний час.
  - Токен аутентификации не требуется

```
{
        "successful_requests_5mins ": X,
        "failed_request_5mins": X,
        "successful_requests_1h ": X,
        "failed_request_1h": X,
}
```

Часть 5. Мониторинг

Внедрите мониторинг, который каждую минуту отслеживает конечную точку API /healthz и инициирует оповещения, если в приложении API возникают какие-либо проблемы. Вы можете написать сценарий и/или использовать любое программное обеспечение, доступное на ISO-образах Debian, для выполнения этой задачи.

Общее

- Создайте каталог /data/ansible/monitoring
  - Все плейбуки должны находиться в корне этого каталога.
  - Вы можете создавать папки/файлы в этом каталоге для запуска плейбуков.
  - Запустите репозиторий git и регулярно вносите свои изменения, чтобы иметь рабочую версию в git.
  - Все задачи должны иметь состояние «ok» или «skipped» после повторного запуска плейбуков.

Мониторинг

- Создайте playbook с именем 1-monitoring.yml для развертывания и настройки системы мониторинга.
  - Измерение SLI на основе успешных и неудачных запросов
  - Запускать оповещения, если количество неудачных запросов выходит за пределы SLO 99,99%
  - Мониторинг всех хостов с помощью ICMP
    - Инициировать оповещение, если есть проблема с подключением, которая нарушает SLO 99,99%
  - Установите службу MariaDB на все хосты в группе «monitoring» и создайте кластер MariaDB со всеми членами этой группы.

- Сохраняйте оповещения и время срабатывания в формате (RFC 3339) (например, 2002-10-02T10:00:00-05:00)
    - Мы должны иметь отказоустойчивость, когда один узел мониторинга выходит из строя
    - Пример предупреждения: 2002-10-02T10:00:00-05:00|"$server has been down for more than 1 minute."

Необходимое оборудование, машины, установки и материалы

Сетевая таблица

ID	IP	ОС	ОПИСАНИЕ
DEV-LIN	10.22.0.251	Debian 11.3 (GNOME)	Development VM with the following software installed: - Postman - Python3
DEV-WIN	10.22.0.252	Windows 10 Pro	Development VM with the following software installed: - Postman - Python3
HOST	10.22.0.50	Debian 11.3	Host Server
LIN1	10.22.0.1	Debian 11.3	Dynamic Config
LIN2	10.22.0.2	Debian 11.3	Dynamic Config
LIN3	10.22.0.3	Debian 11.3	Dynamic Config
LIN4	10.22.0.4	Debian 11.3	Dynamic Config
LIN5	10.22.0.5	Debian 11.3	Dynamic Config
WIN1	10.22.0.101	Windows Server 2019	Dynamic Config
WIN2	10.22.0.102	Windows Server 2019 (Core)	Dynamic Config
WIN3	10.22.0.103	Windows Server 2019 (Core)	Dynamic Config
WIN4	10.22.0.104	Windows Server 2019	Dynamic Config
WIN5	10.22.0.105	Windows Server 2019 (Core)	Dynamic Config
RTR1	Gig1 - 10.22.0.201 Gig2 - 10.0.0.1	Cisco CSR1000v	Internal - Dynamic Config
RTR2	Gig1 - 10.22.0.202 Gig2 - 10.0.0.2	Cisco CSR1000v	Internal - Dynamic Config
RTR3	Gig1 - 10.22.0.203 Gig2 - 10.0.0.3	Cisco CSR1000v	Internal - Dynamic Config
RTR4	Gig1 - 10.22.0.204 Gig2 - 10.0.0.4	Cisco CSR1000v	Internal - Dynamic Config
RTR5	Gig1 - 10.22.0.205 Gig2 - 10.0.0.5 Gig3 - 192.168.55.1	Cisco CSR1000v	Edge - Dynamic Config
RTR6	Gig1 - 10.22.0.206 Gig2 - 10.0.0.6	Cisco CSR1000v	Edge - Dynamic Config

	Gig3 - 192.168.55.2		
RTR7	Gig1 - 10.22.0.207 Gig3 - 192.168.55.3 Loop1 - 1.1.1.1 Loop99 - 99.99.99.99	Cisco CSR1000v	External - Dynamic Config
RTR8	Gig1 - 10.22.0.208 Gig3 - 192.168.55.4 Loop1 - 8.8.8.8 Loop99 - 99.99.99.99	Cisco CSR1000v	External - Dynamic Config
CML	10.22.0.240	Cisco Modelling Lab	Web Access

Схема сети (логическая)

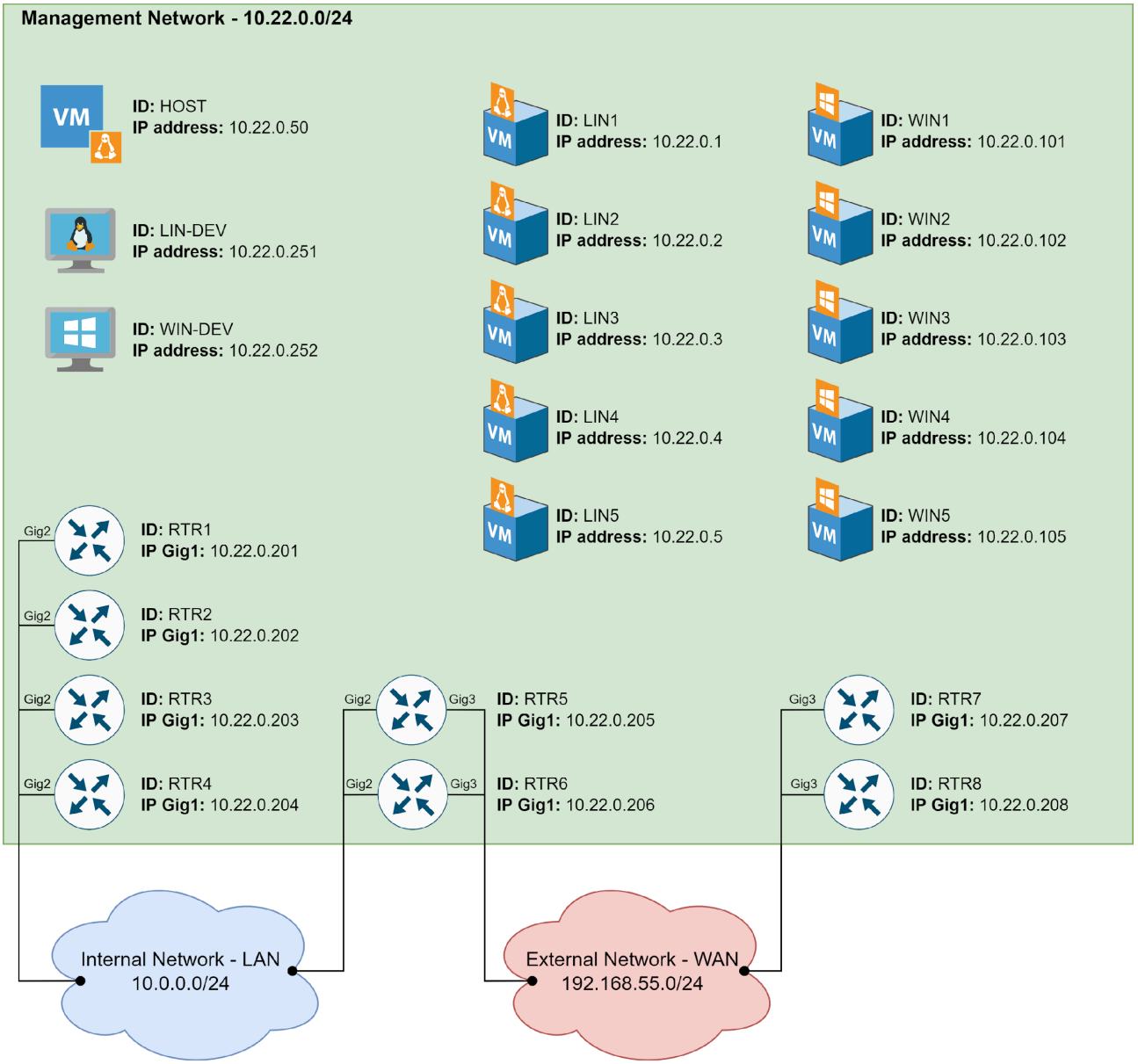

Схема сети (физическая)