1. ОБЩИЕ СВЕДЕНИЯ

Платформа C2 – Cloud Conductor – это облачная сервис-ориентированная платформа с поддержкой концепции NFV для создания, управления и мониторинга виртуальной инфраструктуры и жизненного цикла виртуальных сетевых сервисов.

Платформа C2 разработана в соответствии с международными стандартами ETSI MANO.

Для программной реализации использовались языки:

- интерпретируемый язык Python,
- компилируемый язык C++,
- компилируемый язык TypeScript (с использованием Angular).

Описание виртуальной сетевой функции для ее использования в платформе C2 составляется на языке TOSCA.

Инсталляция платформы C2

Для инсталляции платформы C2 и ее основных модулей используется описание для утилиты ansible, которое хранится под именем C2-installer.

В процессе установки платформы на целевом хосте создается конфигурация для утилиты docker-compose, с помощью которой запускаются контейнеры с модулями.

C2-installer поддерживает установку следующих компонент:

- Модуль C2-sky;
- Модуль C2-orch,
- Модуль C2-VIM,
- Модуль C2-monitoring,
- Модуль C2-VNFM,
- Модуль C2-CPEM,
- Модуль C2-admin
- Модуль C2-customer.
- Подмодуль C2-vim-agent
- Модуль C2-billing
- Модуль C2-cpem

Инсталляция модуля C2-cube выполняется отдельно.

1.1. Необходимые условия

C2-installer поддерживает следующие операционные системы:

- Ubuntu;
- CentOS.

Необходима утилита ansible (пример установки для Ubuntu):

```
sudo apt update
sudo apt install software-properties-common
sudo apt-add-repository --yes --update ppa:ansible/ansible
sudo apt install ansible
```

Необходимо также установить пакеты python3, python3-pip.

В hosts создать секцию [python:vars], добавить строку ansible_python_interpreter=/usr/bin/python3.

C2-installer может быть запущен как с целевого, так и с удаленного хоста:

1. С целевого хоста из папки `/opt/c2-installer` (позволяет использовать разные версии C2-installer на разных хостах);
2. С удаленного хоста через SSH (позволяет производить установку модулей сразу на несколько целевых хостов из одного места).

Рекомендованное место - целевой хост.

1.2. Инсталляция

Примечание. Утилите ansible необходимы SSH ключи для доступа к git репозиториям модулей платформы. Ключи можно добавить на целевой хост либо пробросить через SSH соединение при помощи опции `-A` (опция позволяет использование локальных SSH ключей на удаленном хосте).

Для того чтобы провести инсталляцию платформы C2 выполните следующие действия:

1. Создайте папку `/opt/c2-installer`, дайте пользователю права клонирования c2-installer в `/opt` (по умолчанию содержимое `/opt` доступно только пользователю с правами администратора).

Команды bash для клонирования c2-installer:

```
sudo mkdir /opt/c2-installer
```

```
cd /opt/
sudo chown soc c2-installer
git clone git@gitlab.arccn.ru:soc/c2-installer.git
```

2. Заполните /opt/c2-installer/ssh.config

Этот файл хранит информацию о том, как связаться через SSH со всеми целевыми хостами. Формат файла является стандартным форматом SSH конфигурации (как в local `~/.ssh/config`).

Для одного целевого хоста в файле надо произвести следующие замены:

```
Host example => Host <hostname>
  HostName 10.10.10.10 => HostName <ip>
```

Если целевых хостов более одного, необходимо добавить конфигурацию SSH для каждого. В SSH конфигурации хоста следует указать IP адрес (поле `HostName`), имя пользователя (поле User) и порт SSH (поле Port). Дополнительная информация о конфигурации содержится в руководстве по конфигурационным файлам OpenSSH (https://linux.die.net/man/5/ssh_config).

3. Скопируйте содержимое папки inventories/example в папку `inventories/<farm_name>`, где `<farm_name>` - имя, описывающее группу целевых хостов, на которые будет производиться инсталляция платформы.
4. Заполните конфигурацию хостов в файле `inventories/<farm_name>/hosts` (этот файл называется файлом инвентаря, инструкции по его заполнению приведены ниже).

Примечание. Принадлежность хостов группам описана в файле inventories/<farm_name>/hosts, роли для групп описаны в файлах сценариев в директории playbooks. Задачи, выполняющиеся для роли, описаны в файле roles/<rolename>/tasks/main.yml.

Внесите имена серверов из ssh.config в группы хостов (c2host/zabbix/...) в зависимости от предназначения хостов. Например, на группу хостов zabbix можно установить совместимую с платформой версию zabbix.

Пример файла hosts для одного сервера с именем `tiny`:

```
[c2host]
tiny
[zabbix]
tiny
[graylog]

[os_allinone]
tiny
[os_control]

[os_compute]

[vim_agent:children]
os_allinone
os_control
os_compute
```

Группы могут формировать иерархию: например, группы `os_allinone`, `os_control` и `os_compute` являются подгруппами группы `vim_agent`. Это означает, что все задачи для группы `vim_agent` будут выполняться для хостов всех подгрупп, но каждая подгруппа может иметь уникальную конфигурацию, хранимую в файле `inventories/<farm_name>/group_vars/<subgroup_name>.yml`.

5. В файле `inventories/<farm_name>/group_vars/all.yml` установите значения переменных , параметров и настроек:

- Значение переменной `dc_name` должно совпадать с именем хоста из файла ssh.config (шаг 2). Оно используется модулями C2-Orch, C2-VIM и C2-VIM-agent для корректного взаимодействия друг с другом.

Параметры Openstack:

- переменные `openstack_user`, `openstack_password`, `openstack_project`, `openstack_auth_url` копируются из файла /etc/kolla/admin-openrc.sh.

Адрес сервера API:

- `backend_ip` - IP адрес хоста, где будет работать модуль C2-Sky.

Параметры Zabbix:

- `zabbix_host` - это IP адрес хоста, где zabbix будет инсталлирован
- `zabbix_port` = 8090
-
- `zabbix_username`=Admin
- `zabbix_password`=zabbix

Примечание. Только переменная zabbix_port используется при инсталляции самого Zabbix. Другие переменные используются модулем мониторинга для доступа к API Zabbix.

Настройки CPE контроллера. Возможные значения:

- `cpe_control_username: admin_nfv`
- `cpe_control_password: password_nfv`
- `cpe_control_address: https://172.30.10.1`

Параметры RMQ:

- `rmq_ext_host: localhost`
- `rmq_ext_port: 5673`
- `rmq_user: soc`
- `rmq_pass: s0c`
- `rmq_host: c2-rabbitmq`
- `rmq_port: 5672`

Параметры SDN контроллера. Возможные значения:

- `controller_host` - IP адрес хоста, где будет тоже инсталлирован Runos;

- `controller_port` = 6653.

6. Запустите инсталляцию платформы следующими командами:

```
cd /opt/c2-installer
ansible-playbook -i inventories/<farm_name>
    playbooks/c2host.yml --ask-pass --ask-become-pass
```

Примечание. Рекомендуется запускать инсталляцию без запуска контейнеров, а затем запускать контейнеры вручную. Для запуска контейнеров вручную вызовите утилиту docker-compose по alias без указания абсолютного пути в файловой системе. Данный путь является оптимальным, т.к. запуск контейнеров обычно продолжается около 1 часа, при этом прогресс выполнения этой задачи не очевиден.

Пример:

```
cd /opt/c2-installer
ansible-playbook -i inventories/<farm_name>
    playbooks/c2host.yml --ask-pass --ask-become-pass -
    -skip-tags "startup"
```

```
    ansible-playbook -i inventories/<farm_name>
    playbooks/c2host.yml --ask-pass --ask-become-pass -
    -tags "migrate_db"
    cd /etc/soc/staging/docker/
sudo docker-compose up --build -d
```

7. Выполните инсталляцию zabbix

Для того чтобы инсталлировать zabbix выполните следующие команды:

```
cd /opt/c2-installer
ansible-playbook -i inventories/<farm_name>
    playbooks/zabbix.yml --ask-pass --ask-become-pass
```

Примечание. Если на хосте установлена операционная система CentOS, возможен сбой.

В случае сбоя следует закомментировать в файле `/opt/c2-installer/roles/zabbix/tasks/main.yml` следующие строки:

```
# TODO: use role dependency instead
- name: Setup compose
  include_role:
    name: common
    tasks_from: ubuntu_setup_compose
```

После чего надо запустить инсталляцию заново.

8. Выполните инсталляцию подмодуля C2-VIM-agent

Для того чтобы инсталлировать C2-VIM-agent выполните следующие команды:

```
cd /opt/c2-installer
```

```
ansible-playbook -i inventories/<farm_name>
    playbooks/vim_agent.yml --ask-pass --ask-become-
    pass
```

После завершения инсталляции контейнеры с модулями платформы C2 будут запущены и появится возможность зайти в web-интерфейсы по адресам:

`ip:8082`(c2-admin) и `ip:8083`(c2-customer), где ip - IP адрес сервера, который был добавлен в группу c2host.

1.3. Группы хостов

Предназначение групп хостов:

- c2host – содержит платформу C2;
- zabbix - содержит zabbix для мониторинга виртуальных машин;
- graylog - содержит сборщик логов;
- os_allinone - содержит C2-VIM-agent на хосте, который является одновременно контролирующим и вычислительным для Openstack;
- os_control - содержит C2-VIM-agent на хосте, который является контролирующим для Openstack;
- os_compute - содержит C2-VIM-agent на хосте, который является вычислительным для Openstack.

1.4. Переменные

Переменные могут быть настроены в файлах папки `inventories/<farm_name>/group_vars`. Большинство из них расположено в файле `all.yml`. Эти переменные перезаписывают локальные

переменные, определенные в файлах `defaults/all.yml` внутри папок ролей. Назначение переменной соответствует ее названию либо приведено в сопутствующем комментарии.

1.5. Запуск ansible

Имя утилиты: `ansible-playbook`

Необходимый аргумент – имя файла сценария. Файлы сценариев хранятся в папке `playbooks` и представляют собой отношение групп хостов к ролям, где каждая роль в свою очередь определяет набор задач, связанных с установкой.

```
ansible-playbook playbooks/c2host.yml
```

Также могут быть полезны следующие опциональные аргументы:

- `-i` – позволяет выбрать файл инвентаря. Аргумент необходим, если в корне проекта нет инвентаря по умолчанию.

  Пример: ansible-playbook -i inventories/example

- `--tags` – список тегов, разделенных запятыми, задачи с которыми надо запускать;

  Пример: ansible-playbook --tags `"update-docker-env,startup"`

- `--skip-tags` – список тегов, разделенных запятыми, задачи с которыми не надо запускать;

  Пример: ansible-playbook --skip-tags `"update-docker-env,startup"`

- `--ask-pass` – потребовать ручной ввод пароля пользователя, обычно используется с `--ask-become-pass`, который требует пароль для получения прав администратора.

Пример: `ansible-playbook --ask-pass --ask-become-pass`

1.6. Конфигурация контейнеров

Если инсталляция завершена успешно, в папке платформы C2 создается папка `docker` с релевантной конфигурацией контейнеров.

Не рекомендуется проводить какие-либо изменения в папке `docker` самостоятельно: если она будет обновлена через c2-installer, то локальные изменения будут потеряны.

1.7. Инсталляция модуля c2-cube

Настройка конфигурации

Запустить контейнер cube из заранее подготовленного образа:

sudo docker run -itd --name c2-cube c2-cube

Зайти внутрь контейнера:

sudo docker exec -it -uroot c2-cube /bin/ash

открыть файл /etc/soc/c2-cube/runos/network-settings.json.

В секции rmq_settings выставить актуальные параметры rabbitmq сервера так, как если бы доступ осуществлялся с хостовой системы.

Выйти из контейнера и запустить команду:

sudo docker commit c2-cube c2-cube-with-params

Удалить созданный контейнер:

sudo docker rm -fv c2-cube

Запуск

Cube запускается из созданного на этапе выше образа docker контейнера с помощью следующей команды:

sudo docker run -itd --name c2-cube --network host c2-cube-with-params nix-shell --run "/etc/soc/c2-cube/runos/build/runos -c /etc/soc/c2-cube/runos/network-settings.json" /etc/soc/c2-cube/runos/default.nix

2. Конфигурация и предназначение ролей

Изменение конфигурации платформы выполняется преимущественно в файле `inventories/<farm_name>/group_vars/all.yml`.

Переменные роли хранятся в файле `roles/role/defaults/main.yml`. Их значения можно переопределить через переменные групп хостов в файле `inventories/<farm_name>/group_vars/all.yml`.

Примечание. Следует избегать изменения конфигурации докер-контейнеров на хосте (в папке платформы C2) напрямую, так как она может быть перезаписана после вызова ansible с тегом "update-docker-env". Возможные опции: изменение переменных окружения докер-контейнеров в ansible инсталляции, используемой для контроля фермы или запрос изменений в репозитории, затем обновление ansible инсталляции.

2.1. Роли

2.1.1.1. common

Роль common содержит задачи, которые используются в других ролях. В текущей версии роль предоставляет задачи для установки утилит docker и docker-compose для операционной системы Ubuntu.

2.1.1.2. C2-installer

Данная роль устанавливает следующие модули платформы:

- C2-sky;
- C2-orch,

- C2-VIM,
- C2-monitoring,
- C2-VNFM,
- C2-CPEM,
- C2-admin,
- C2-customer,
- C2-billing.

Задачи роли также копируют конфигурацию докер-контейнеров для всех включенных модулей в директорию `/etc/soc/staging/docker`.

Примечание. Для использования утилиты docker-compose необходимо, чтобы существовали папки всех модулей. Если платформа C2 установлена частично (не все модули), то для некоторых пропущенных при установке модулей необходимо вручную создать папку. Например,: ansible вызвана с аргументом --skip-tags "c2-monitoring", после чего утилита docker-compose не сможет запустить контейнеры из-за отсутствия папки модуля c2-monitoring (так как docker-compose проверяет наличие всех папок описанных в файле конфигурации). Решение - создать папку модуля c2-monitoring рядом с папками остальных модулей.

2.1.1.3. Zabbix

Роль zabbix содержит задачи для установки Zabbix со скриптами мониторинга.

Параметры конфигурации:

- zabbix_port – порт, через который zabbix интерфейс связан с 0.0.0.0 хоста.

Примечание. Переменные конфигурации *zabbix_host*, *zabbix_username* и zabbix_password не

используются в задачах этой роли. Для учетных данных по умолчанию установлено значение `Admin/zabbix`.

2.1.1.4. VIM-agent

Роль VIM-agent используется для инсталляции подмодуля C2-VIM-agent.

2.1.1.5. Graylog

Роль Graylog используется для инсталляции Graylog (после инсталляции требуется выполнить конфигурирование вручную).

2.2. ПАТЧИ OPENSTACK

Для функционирования платформы C2 необходима установка патчей на OpenStack. В текущей версии установка выполняется вручную . Cinder

Патч реализует функциональность размещения volume на конкретный cinder хост. По умолчанию место для volume выбирается автоматически. Патч имеет смысл только для lvm хранилища.

2.2.1.1. DHCP

Патч позволяет произвольному Mac-адресу получить IP-адрес от DHCP сервера openstack (по умолчанию это могут сделать только созданные в openstack порта). Патч необходим для работы vCPE.

Также патч устанавливает необходимое значение MTU, раздаваемое DHCP. MTU выставляется меньше 1500 из-за использования vxlan туннелей.

2.2.1.2. SFC

Патч применяется к библиотеке построения цепочек networking-sfc и делает следующее:

- Изменяет логику валидации классификаторов в sfc: по умолчанию это логика очень строгая и нужна для предотвращения коллизий. Однако VIM имеет собственный алгоритм предотвращения коллизий классификаторов, и такая строгая логика не нужна.

− Позволяет создавать пары портов, проходящих через порты, не привязанные к устройству. Это необходимо для того, чтобы строить цепочки через ovs bridge, которые управляются openflow контроллером в С2. Так как openstack ничего не знает об этих ovs bridge, такие порты считаются не привязанными к устройству.

− Позволяет создавать более одной цепочки, состоящей из одной пары портов (одинаковой в разных цепочках), причем у этой пары портов in и out порты совпадают. По неизвестной причине в стандартном sfc это запрещено.

− Заставляет neutron ovs agent выставлять дополнительные правила, всегда пропускающие определенный трафик, например dhcp трафик и трафик от zabbix агентов. Это необходимо для упрощения и значительного ускорения задания классификаторов при построении цепочек в С2.

− Позволяет задавать приоритет классификаторов. Это расширяет возможности управления трафиком в цепочках.

− Позволяет создавать `dummy` классификаторы. Такие классификаторы, вместо направления трафика в цепочку, пропускают его вне цепочки. Это необходимо для создания правил вида "отправить в цепочку всё, кроме трафика с конкретного IP адреса".

3.2 Конфигурация OpenStack

Для работы sfc необходимо внести изменения в следующие конфиги:

- neutron.conf (/etc/kolla/neutron_server/neutron.conf+/etc/kolla/neutron_openvswitch_agent/neutron.conf, на всех узлах):
    - в строке service_plugins через запятую добавить в конец networking_sfc.services.sfc.plugin.SfcPlugin,networking_sfc.services.flowclassifier.plugin.FlowClassifierPlugin
    - Добавить в конец секцию с именем sfc, добавить в неё параметр drivers = ovs

    - Добавить в конец секцию с именем flowclassifier, добавить в неё параметр drivers = ovs
- ml2_conf.ini (/etc/kolla/neutron-openvswitch-agent/ml2_conf.ini на всех нодах):
    - в секцию agent добавить extensions = sfc, проверить, что l2_population = true, arp_responder = true
    - ml2_conf.ini (/etc/kolla/neutron-server/ml2_conf.ini либо /etc/neutron/plugins/ml2/openvswitch_agent.ini):
    - в секции ml2 проверить, что в mechanism_drivers есть openvswitch

Теги

Для частичной инсталляции C2 и дальнейшего управления платформой в C2-Installer используются теги.

В текущей версии поддерживаются следующие теги:

- `update-docker-env` - обновляет конфигурацию docker-compose;
- `c2-admin` - инсталляция/обновление C2-admin;
- `c2-customer` - инсталляция/обновление C2-customer;
- `c2-monitoring` - инсталляция/обновление C2-monitoring;
- `c2-cpe-manager` - инсталляция/обновление C2-CPEM;
- `startup` – запуск контейнеров.

С помощью тегов можно запускать или пропускать задачи, используя указанные ранее опции утилиты ansible-playbook.