# Описание системы:

Проект consaltingua находится на Yandex.cloud; идентификатор облака: b1gtpok2mfu5dds9lgqo.

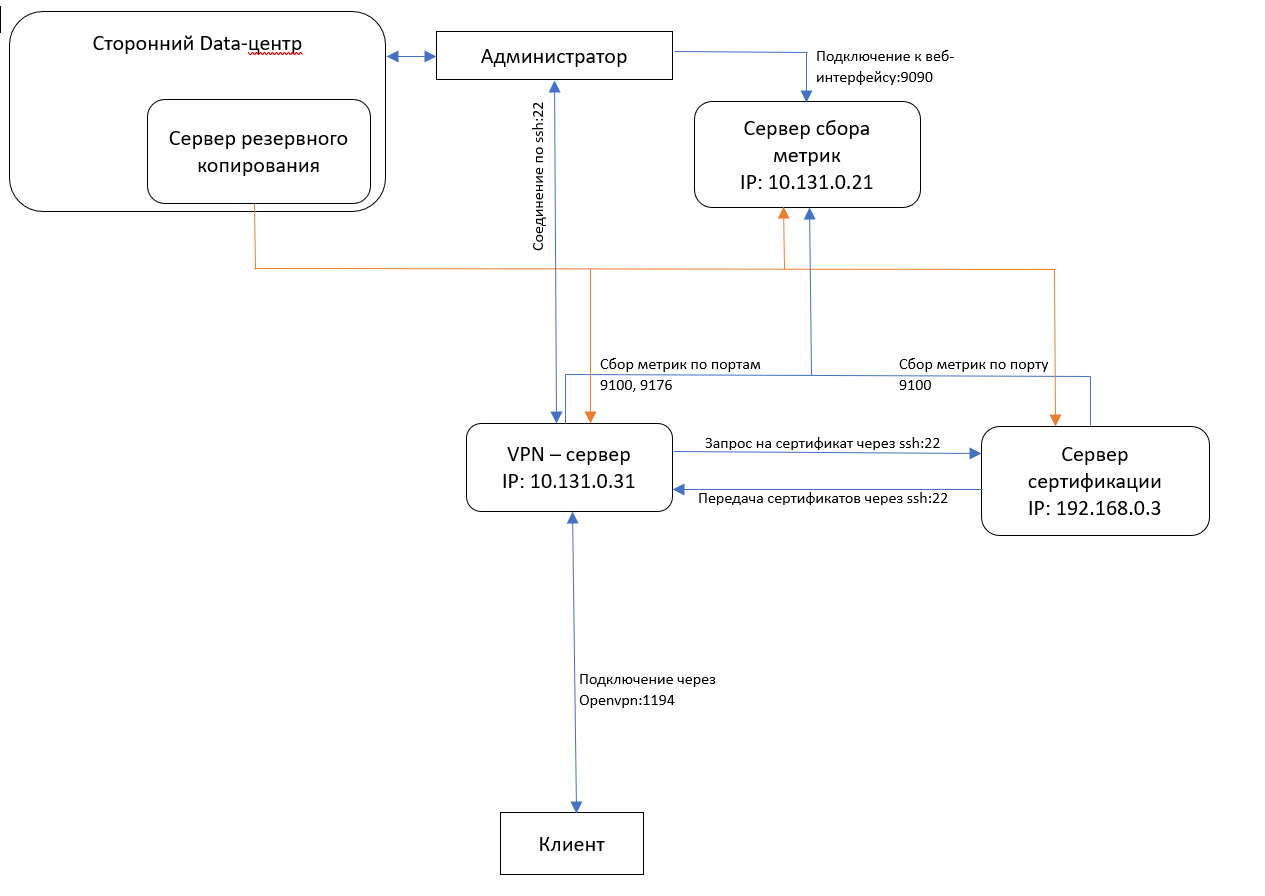
# Текущие аккаунты:

Алексей Коптев, права доступа owner

# Состав облака:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Вид элемента | Имя | Описание | Внутренний IPv4 | Публичный IPv4[[1]](#footnote-1) |
| Виртуальная машина | ca | Сервер сертификации | 192.168.0.3 |  |
| Виртуальная машина | vpn | VPN-сервер | 10.131.0.31 | 158.160.144.239 |
| Виртуальная машина | prometheus | Сервер сбора метрик | 10.131.0.21 | 158.160.143.112 |

# Схема инфраструктуры:



# Развертывание серверов:

## Сервер сертификации:

1. Для сервера сертификации создается отдельная подсеть 192.168.0.0/28, в группе безопасности перекрываем все порты для подсети кроме 22, 443, 53, 80, 9100. Локальный IP адрес устанавливаем 192.168.0.3.
2. С корпоративного гитлаба закачиваем на сервер пакет configcaserver и скрипт Install\_CA.sh. Для развертывания сервера используем скрипт Install\_CA.sh:

Sudo ./Install\_CA.sh configcaserver\_0.1-4\_all.deb

В качестве параметра используется имя пакета. Скрипт установит easy-rsa, утилиту для iptables, а также экспортер. Помимо этого, в папку /opt/scripts скопируются скрипты для автоматизации некоторых работ на сервере.

1. Запускаем скрипт с правами рута /opt/scripts/IptablesCA.sh для закрытия всех портов на сервере кроме 22 и 9100 (только для Prometheus сервера):

Sudo /opt/scripts/IptablesCA.sh

1. Запускаем скрипт /opt/scripts/CreateCertificate.sh для создания корневого сертификата:

/opt/scripts/CreateCertificate.sh

Если сертификат создался, развертывание прошло успешно.

Состав deb пакета:

* /opt/scripts/CreateCertificate.sh – скрипт создания корневого сертификата
* /opt/scripts/IptablesCA.sh – скрипт настроек iptables для сервера сертификации
* /opt/easy-rsa/vars – прописанный конфиг файл easy-rsa
* /opt/scripts/signserver.sh – скрипт создания серверных сертификатов (используют заранее прописанные адреса серверов)
* /opt/scripts/signclient.sh – скрипт создания клиентских сертификатов (используют заранее прописанные адреса серверов)

## VPN-сервер:

1. Для vpn-сервера используется дефолтная подсеть default-ru-central1-d 10.131.0.0/24. Локальный IP адрес устанавливаем 10.131.0.31.
2. С корпоративного гитлаба закачиваем на сервер пакет vpnserver и скрипт installvpnserver.sh. Для развертывания сервера используем скрипт installvpnserver.sh:

Sudo ./installvpnserver.sh vpnserver\_0.1-4\_all.deb

В качестве параметра используется имя пакета. Скрипт установит easy-rsa, openvpn, утилиту для iptables, экспортер для узла и openvpn, а также golang необходимый для работы экспортера openvpn. Помимо этого, в папку /opt/scripts скопируются скрипты для автоматизации некоторых работ на сервере.

1. Дальнейшая настройка сервера требует настройку ssh соединения между vpn-сервером и ca-сервером, проводящаяся вручную. Предполагается что соединение будет настроено для текущего пользователя.
2. Запускаем скрипт /opt/scripts/serverkey.sh:

/opt/scripts/serverkey.sh

Скрипт скачает корневой сертификат, создаст и отправит запрос на подписание на сервер сертификации. После этого на сервере сертификации надо будет запустить скрипт создания серверного сертификата. Сертификат автоматически будет отправлен на сервер VPN

1. Запускаем с правами рута скрипт /opt/scripts/configclient.sh:

Sudo /opt/scripts/configclient.sh 158.160.144.239

В качестве параметра использует внешний IP адрес vpn-сервера. Скрипт перенесет необходимые конфигурационные файлы.

1. Запускаем с правами рута скрипт /opt/scripts/startupvpn.sh

Sudo /opt/scripts/startupvpn.sh

Скрипт сохранит полученный серверный сертификат и запустит [openvpn-server@server.service](mailto:openvpn-server@server.service)

1. Запускаем с правами рута скрипт /opt/scripts/iptables.sh

Sudo /opt/scripts/iptables.sh udp 1194

Скрипт требует в параметрах протокол и порт, прописывает туннелирование и закрытие всех портов кроме 22, 80, 443, 1194, 9100 (только для Prometheus сервера), 9176 (только для Prometheus сервера) в iptables.

Если все прошло успешно, то можно установить впн-туннель к серверу с помощью клиента openvpn.

Создание сертификатов пользователей будет в разделе ниже.

Состав deb пакета:

* /opt/clients/base.conf – конфигурационный файл клиента Openvpn
* /etc/openvpn/server/server.conf – конфигурационный файл сервера Openvpn
* /opt/scripts/sysctl.conf – конфигурационный файл системы с настройками сети
* /opt/scripts/iptables.sh – скрипт настроек iptables для сервера vpn
* /opt/scripts/clientkey.sh – скрипт создания запроса сертификата для клиента (используют заранее прописанные адреса серверов)
* /opt/scripts/startupvpn.sh – скрипт запуска сервиса openvpn
* /opt/scripts/serverkey.sh – скрипт создания запроса сертификата для сервера (используют заранее прописанные адреса серверов)
* /opt/scripts/configclient.sh – скрипт прописывающий в конфигурационные файлы актуальный адрес сервера
* /opt/scripts/makeconfig.sh – скрипт создающий .ovpn для клиента
* /etc/systemd/system/openvpn\_exporter.service – Юнит для экспортера openvpn

## Prometheus сервер:

1. Для prometheus-сервера используется дефолтная подсеть default-ru-central1-d 10.131.0.0/24. Локальный IP адрес устанавливаем 10.131.0.21.
2. С корпоративного гитлаба закачиваем на сервер пакет prometheusserver и скрипт installprometheus.sh. Для развертывания сервера используем скрипт installprometheus.sh:

Sudo ./ installprometheus.sh prometheusserver\_0.1-3\_all.deb

В качестве параметра используется имя пакета. Скрипт установит prometheus, утилиту для iptables, alertmanager, экспортер для узла, а также python3-bcrypt необходимый для работы шифрования пароля. Помимо этого, в папку /opt/scripts скопируются скрипты для автоматизации некоторых работ на сервере.

1. Запускаем с правами рута скрипт /opt/scripts/iptables.sh

Sudo /opt/scripts/iptables.sh

Скрипт закроет все порты кроме 22, 80, 443, 9090

1. Запускаем скрипт с правами рута /opt/scripts/makepass.sh:

Sudo /opt/scripts/makepass.sh

Скрипт предложит придумать пароль для учетной записи admin и установит его.

Если все прошло успешно, то по адресу Prometheus:9090 будет доступна web версия сбора метрик со снимаемыми со всех серверов метрик.

Состав deb пакета:

* /opt/scripts/configprometheus.sh – скрипт копирует конфигурационные файлы Prometheus и перезапускает сервис.
* /opt/scripts/makepass.sh – запрашивает и устанавливает пароль на сайт с метриками
* /opt/scripts/web.yml – конфигурационный файл авторизации на сайте Prometheus
* /opt/scripts/gen-pass.py – скрипт запрашивает пароль и шифрует его
* /opt/scripts/prometheus.service – Юнит Prometheus с подключенным конфигурационным файлом авторизации
* /opt/scripts/iptables.sh – скрипт настроек iptables для сервера Prometheus
* /opt/scripts/rules.yml – конфигурационный файл c alerts
* /opt/scripts/prometheus.yml – конфигурационный файл Prometheus (используют заранее прописанные адреса серверов)
* /opt/scripts/alertmanager.yml – конфигурационный файл алертменеджера с установленной для рассылки почты (по умолчанию прописана почта владельца)

# Подпись сертификатов:

## Подпись серверного сертификата:

1. Запускаем на vpn-сервере скрипт /opt/scripts/serverkey.sh:

/opt/scripts/serverkey.sh

Скрипт скачает корневой сертификат, создаст и отправит запрос на подписание на сервер сертификации.

1. Запускам на ca-сервере скрипт /opt/scripts/signserver.sh

/opt/scripts/signserver.sh

Скрипт создаст сертификат и отправит его сервер vpn

## Подпись клиентского сертификата и создание файла .ovpn:

1. Запускаем на vpn-сервере скрипт /opt/scripts/clientkey.sh:

/opt/scripts/clientkey.sh ivan

Скрипт требует в параметрах указать имя клиента. Он создаст и отправит запрос на подписание на сервер сертификации.

1. Запускам на ca-сервере скрипт /opt/scripts/signclient.sh

/opt/scripts/signclient.sh ivan

Скрипт требует в параметрах указать имя клиента. Он создаст сертификат и отправит его сервер vpn

1. Запускаем на vpn-сервере /opt/scripts/makeconfig.sh

/opt/scripts/makeconfig.sh ivan

Скрипт требует в параметрах указать имя клиента. Он создаст файл настроек ivan.ovpn с именем пользователя, который необходимо передать пользователю для подключения его к vpn.

# Описание alerts:

alert: InstanceDown – Критический алерт. Срабатывает если какой-то сервис не доступен. Требуется выяснить причину неработоспособности. В зависимости от причины могут потребоваться такие серьезные шаги как резервное восстановление, повторное развертывание серверов.

alert: HostUnusualNetworkThroughputIn – Предупреждающий алерт. Слишком большая загруженность входящего трафика. Возможно, сервер под DDoS атакой

alert: HostUnusualNetworkThroughputOut – Предупреждающий алерт. Слишком большая загруженность исходящего трафика. Возможно, стоит проверить сервер на вредоносное ПО

alert: HostOutOfDiskSpace – Предупреждающий алерт. На сервере заканчивается место. Требуется либо почистить его, либо увеличить размер

alert: HostHighCpuLoad – Предупреждающий алерт. На сервере слишком загружен CPU. Требуется уточнить причину перегрузки, возможно нужно заменить процессор на более мощный.

alert: HostClockSkew – Предупреждающий алерт. Обнаружен рассинхрон времени на сервере с глобальным. Требуется синхронизировать его.

alert: HostOutOfMemory – Предупреждающий алерт. Недостаточно памяти на сервере. Требуется уточнить загрузку памяти, возможно нужно ее увеличить.

alert: ManyConnections – Предупреждающий алерт. Слишком много vpn соединений. Требуется уточнить что за соединения есть на сервере, возможно ключ клиента был скомпрометирован. Правило нужно обновлять под реалии соединений

# Описание резервного копирования:

Для сохранения скриптов, пакетов для развертывания и т.д. используется гитлаб корпорации. Не забудьте делать копии к себе на физический носитель.

Для сохранения инфраструктуры используется два бэкап сервера – инструмент предлагаемый облаком яндекс, а также сторонним сервисом облачного бэкапа. На стороннем сервисе для бэкапа используется Duplicati 2, и ставится для мониторинга контейнер Zabbix.

1. Поскольку за статичный публичный IPv4 в яндекс клауд требуется платить даже при выключенных машинах по факту в работе использовались динамические адреса, но предполагается, что они статичные. [↑](#footnote-ref-1)