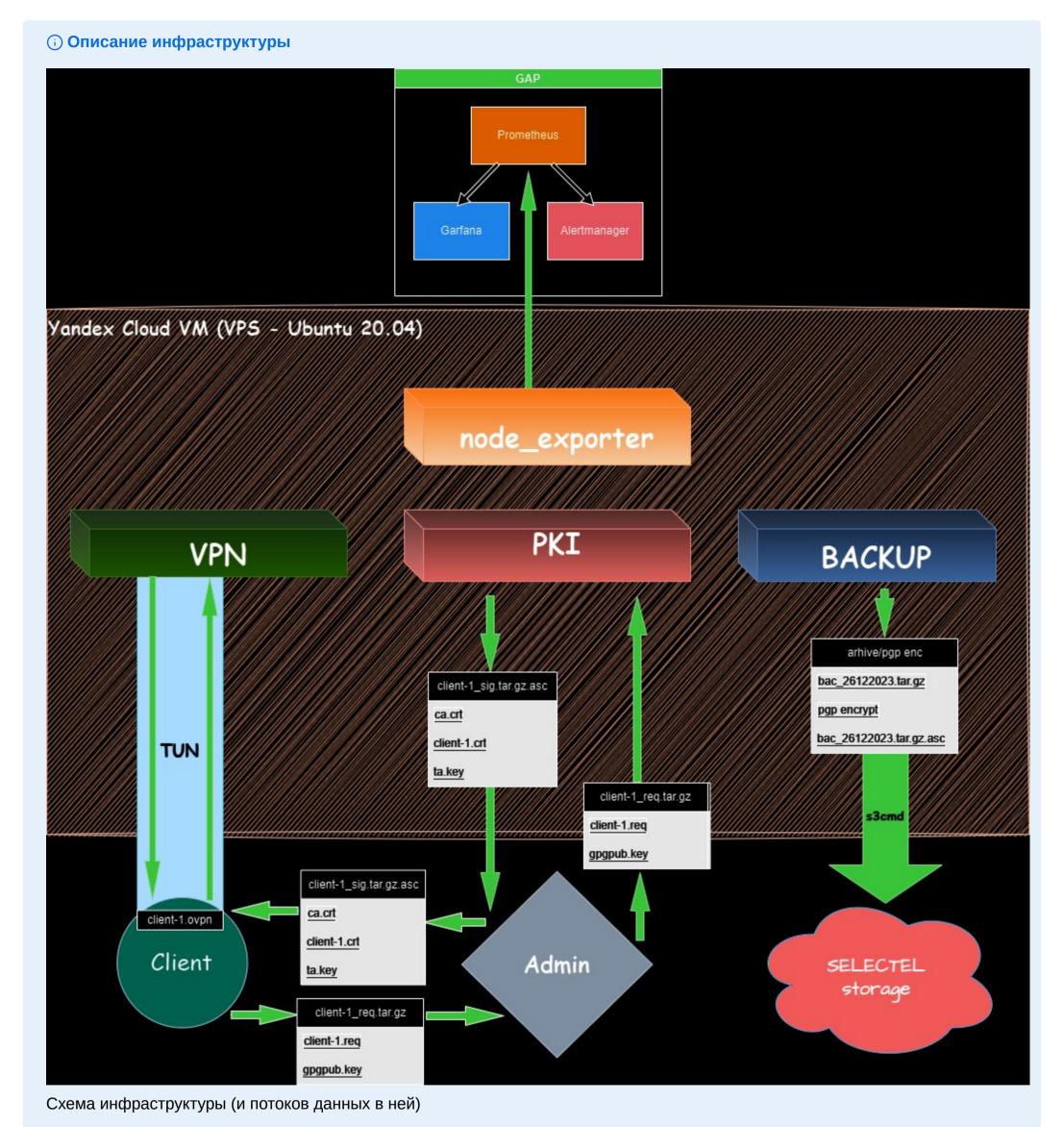
infrastructure_doc



Описание инфраструктуры

Система может быть развернута на ресурсах любого облачного провайдера с OS Ubuntu 20.04 (на других версиях не тестировалось).

В демонстрационном скринкасте, система разворачивается на виртуально машине YandexCloud.

Исходные файлы пакетов, а также собранные пакеты для загрузки и установки, находятся в репозитарии <u>GitHub</u>.

Пакеты

• **pki_0.1-1_all.deb** - пакет разворачивающий центр для выдачи сертификатов - Public Key Infrastructure (PKI). (для работы требует следующие зависимости: easy-rsa, gpg)

- ovpn_0.1-1_all.deb пакет развертывающий OpenVPN сервер (для работы требует следующие зависимости: gnupg, prometheus-node-exporter, iptables-persistent, net-tools, openvpn, easy-rsa, pki)
- bcp_0.1-1_all.deb пакет добавляющий сервис, позволяющий организовать резервирование критически важных файлов и директорий системы, шифрование этого архива и отправку его в контейнер сетевого хранилища SELECTEL (для работы требует следующие зависимости: gpg, gpg-agent, s3cmd). При установки пакета в систему добавляются такие файлы:
 - /usr/bin/bac.sh скрипт организующий упаковку файлов в архив, шифрование архива и отправку его в облако SELECTEL
 - /etc/bac.conf конфигурационный файл содержащий перечень директорий и файлов для добавления их в архив рез.
 копии
 - /etc/systemd/system/bac.service systemd unit запускающий скрипт bac.sh
 - /etc/systemd/system/bac.timer systemd unit типа timer запускающийся один раз в сутки, и вызывающий unit bac.service

? Резервное копирование >

Резервное копирование организовано двумя способами

- 1 Cloud Backup позволяет автоматически создавать резервные копии виртуальных машин. Копии ВМ создаются по принципу application consistent: сохраняются не только данные на дисках, но и промежуточные данные запущенных приложений память и текущие операции чтения и записи на диски. Это позволяет быстро возобновить работу приложений после восстановления ВМ. (Требуется подключение и настройка услуги в YandexCloud панели)
- 2 bac.timer сервис который добавляется при установке пакета bcp_0.1-1_all.deb

Система мониторинга организована на с помощью стека GAP (Grafana, Alertmanager, Prometheus). Доступ к Prometheus и Grafana через веб интерфейс открыть по адресам (и доступен только для конкретных IP, через блокировку в iptables). Grafana запущена в docker контейнере. Логин и пароль предоставляет администратор сервера по запросу на email toorrp4@gmail.com

- http://toorrp4.fvds.ru:9090/
- http://toorrp4.fvds.ru:3000/

Alertmanager сконфигурирован так, что уведомления о срабатывании алертов приходят на email, а правила для них разнесены по файлам с логическими названиями соответствующие метрикам.

Окрипты автоматизации установки .deb пакетов >

Там же в корне проекта находятся два bash скрипта, которые позволяют автоматизировать пересборку пакетов при внесении изменения в них, с автоматической выгрузкой этих пакетов на сервер, и запуском их установки

- push.sh скрипт для развертывания РКІ на сервере
- push_client.sh скрипт установки клиентского пакета на машину клиента

ВАЖНО: эти скрипты являются не обязательными, а служат лишь для ускорения тестирования изменений в пакетах. Вы можете скачать весть проект полностью, внести изменения в пакет, и затем вызвать скрипт:

```
./push.sh
```

BAЖHO: push.sh отвечает только за пакеты для PKI, а push_client.sh только за пакет для клиента. Соответственно для установки на сервер клиента, требуется запуск скрипта

```
./push_client.sh
```

Для использования этих скриптов, предварительно нужно настроить SSH доступ к серверу по ключу (на который будет выполнятся установка)

Копируем публичный ключ со своей машины на сервер

```
ssh-copy-id -i ~/.ssh/id_rsa.pub admin@REMOTE_SERVER_IP
```

Закрываем доступ по паролю и использование root логина В файле /etc/ssh/sshd_config выставляем такие значения директивам PubkeyAuthentication yes
AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2
PermitRootLogin No
PasswordAuthentication No
PermitEmptyPasswords No

Перезагружаем сервер sshd

sudo systemctl restart sshd

Скачиваем весь проект из репозитария https://github.com/Cleverscript/pki
Переходим в директорию (путь к вашей директории куда вы загрузите может отличатся!)

cd ~/WORK/DEB/PKI

Запускаем скрипт, которые пересоздаст .deb пакеты, выгрузит их на указанный сервер и запустит их установку

./push.sh

(i) Инструкция для администратора

Инструкция для администратора

○ Инструкция для пользователей VPN

Инструкция для клиента