Руководство для администраторов:

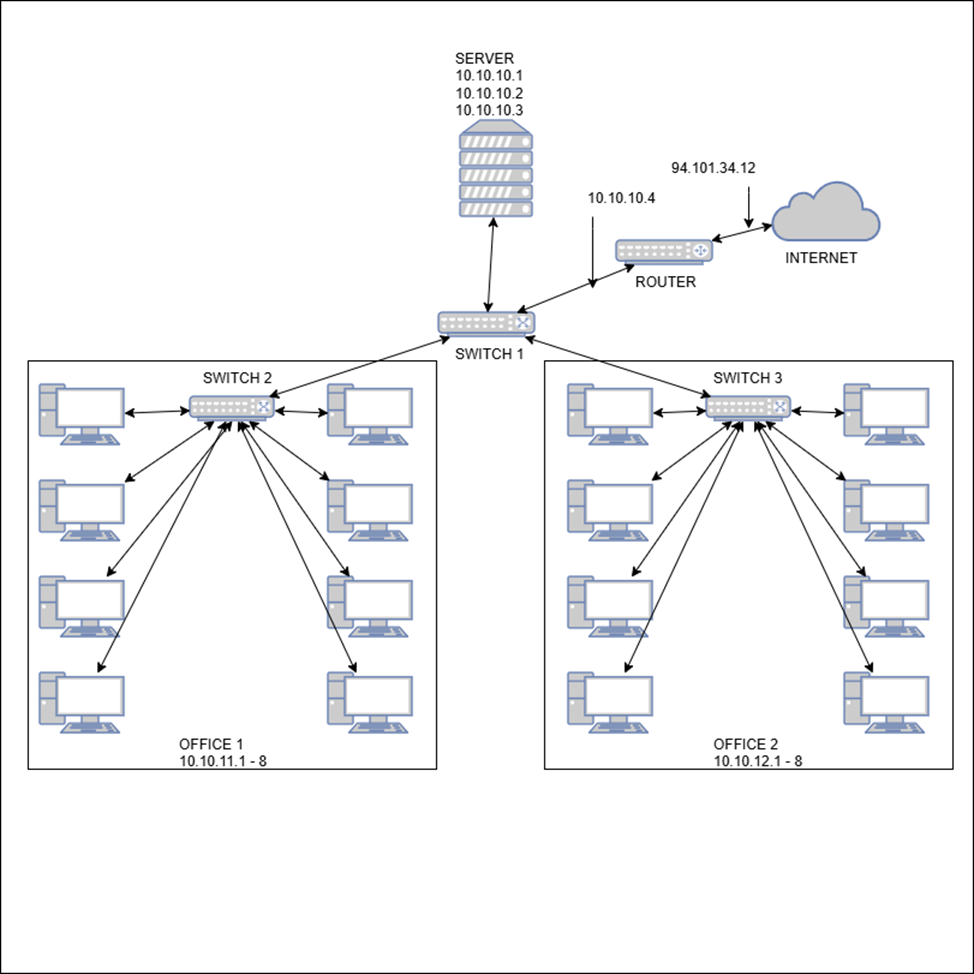
Местоположение: локальное в здании компании.

Группы:

Admin-vlan – имеют доступ ко всем ресурсам сети. В данной группе состоят две учетные записи admin1 и admin2.

Staff – в ней состоят сотрудники имеющие ограниченный доступ к сетевым ресурсам. Учетные записи имеют в названии фамилии сотрудников.

Схема инфраструктуры:



В данной сети есть один сервер с ОС Ubuntu на котором находятся две виртуальные машины. Одна виртуальная машина является vpn-сервером, с которой производится подключение к другим компьютерам в офисах с помощью Anydesk, вторая нужна для сбора метрик серверов с помощью Prometheus. На основном сервере происходит хранение данных, бэкапов и скриптов в gitlab.

Сервисы:

Vpn - 10.10.10.3:1194

Prometheus - 10.10.10.2:9090

Prometheus-node-exporter - 10.10.10.1–3:9100

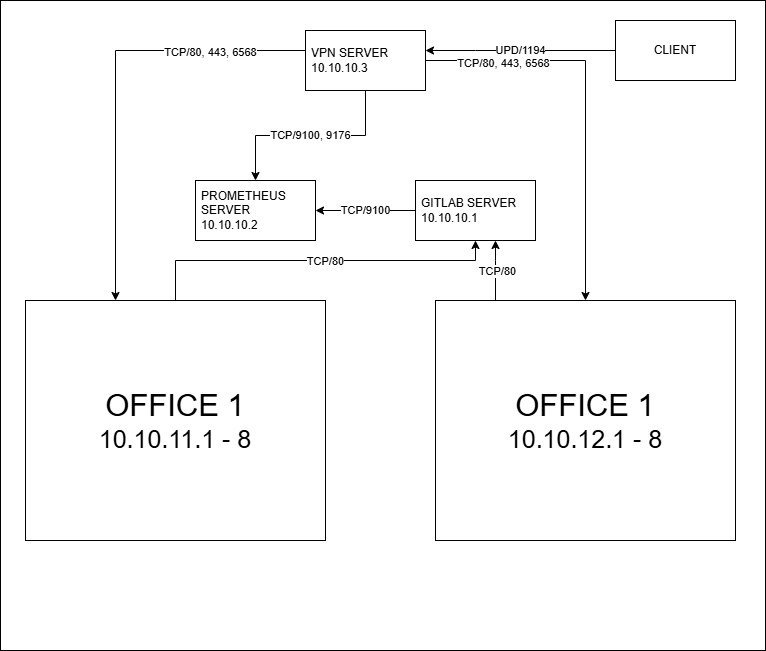
Prometheus-alertmanager - 10.10.10.2:9093, 25

Prometheus-vpn-exporter - 10.10.10.3: 9176

Gitlab - 10.10.10.1:80, 443, 22

Anydesk - 10.10.10.3: 80, 443, 6568 | 10.10.11.1–8: 80, 443, 6568 | 10.10.12.1–8: 80, 443, 6568

Схема взаимодействия компонентов сети:



Характеристики для двух виртуальных машин в virtualbox:

Образ ОС: Ubuntu Server 22.04 LTS

Размер ВМ: 4 vCPU, 4 GB RAM, 50 GB HDD

Сеть: Сетевой мост

**Настройка виртуальных машин на сервере, на случай если они будут утеряны или повреждены без возможности восстановления**.

**Для настройки vpn сервера(10.10.10.3):**

1. Необходимо клонировать из github репозиторий командой «git clone https://github.com/referenc2/My\_final\_job.git»
2. Запустить deb-пакет командой «dpkg -i ./My\_final\_job/task\_1/easy-rsa-config\_0.1-4\_all.deb»
3. Перейти в папку /home/root/easy-rsa и запустить скрипт введя «./easyrsa gen-req server nopass»
4. Скопировать ключ с помощью команды «cp /home/root/easy-rsa/pki/private/server.key /etc/openvpn/server/»
5. Выполнить подписание сертификата с помощью «./easyrsa sign-req server server»
6. Скопировать созданные в итоге файлы в /etc/openvpn/server/
7. Запустить скрипт script\_for\_vpn.sh, он находится в папке ./My\_final\_job/task\_2/
8. Перейти в директорию /etc/openvpn/server/ и запустить команду «openvpn --genkey --secret ta.key».
9. В итоге в /etc/openvpn/server/ должны быть:

ca.crt

server.conf

server.crt

server.key

ta.key

1. Перезапустить службу openvpn.

После настройки openvpn нужно настроить резервное копирование. Для этого нужно установить duplicity с помощью команды «sudo apt-get install duplicity», далее нужно вставить строку «0 8 \* \* 0 duplicity –no-encryption /etc/openvpn sftp://admin1@10.10.10.1//backup-from-vpn/backup-$(date +%Y-%m-%d)» в crontab.

**Для создания сертификата клиента openvpn необходимо:**

1. Перейти в папку /home/root/easy-rsa и запустить скрипт введя «./easyrsa gen-req client-1 nopass»
2. И скопировать client-1.key из папки /home/root/easy-rsa/pki/private/ в /etc/openvpn/client/
3. Подписать запрос с помощью команды «./easyrsa sign-req client client-1»
4. Скопировать необходимые файлы с помощью череды команд:

cp /home/root/easy-rsa/pki/issued/client1.crt /etc/openvpn/client/

cp /home/root/easy-rsa/ta.key /etc/openvpn/client/

cp /etc/openvpn/server/ca.crt /etc/openvpn/client/

1. Перейти в директорию /etc/openvpn/client/ запустить скрипт командой «./make\_config.sh client-1»

В итоге этих действий будет создан сертификат client-1.ovpn в папке /etc/openvpn/client/.

**Для настройки сервера сбора метрик(10.10.10.2):**

1. Нужно клонировать репозиторий из gitlab командой «git clone https://github.com/referenc2/My\_final\_job.git»
2. Запустить скрипт script\_for\_metrics.sh, он находится в папке ./My\_final\_job/task\_3/
3. Скопировать файл ./My\_final\_job/task\_3/config-for-prometheus-1.1/alerts.yml в /etc/prometheus/
4. Отредактировать файл /etc/prometheus/prometheus.yml вставив по строчкой «rule\_files:» строчку «– “alerts.yml”».

Это должно выглядеть так:

rule\_files:

- "alerts.yml"

1. Отредактировать /etc/prometheus/alertmanager.yml пользуясь официальной документацией по ссылке: https://prometheus.io/docs/alerting/latest/configuration/
2. Перезапустить службу prometheus и prometheus-alertmanager

После настройки prometheus нужно настроить резервное копирование. Для этого нужно установить duplicity с помощью команды «sudo apt-get install duplicity», далее нужно вставить строку «0 8 \* \* 0 duplicity –no-encryption /etc/prometheus sftp://admin1@10.10.10.1//backup-from-prometcheus/backup-$(date +%Y-%m-%d)» в crontab.

**Бэкапы происходят в воскресенье в восемь часов.**

**Чтобы восстановить данные из бэкапа для prometcheus** необходимо ввести «duplicity sftp://admin1@10.10.10.1//backup-from-prometcheus/backup-(и нужную дату) /etc/prometcheus»

**Чтобы восстановить данные из бэкапа для openvpn** необходимо ввести «duplicity sftp://admin1@10.10.10.1//backup-from-openvpn/backup-(и нужную дату) /etc/openvpn»

**Компоненты системы мониторинга:**

* Prometheus - сбор метрик и визуализация
* Node Exporter - системные метрики
* OpenVPN Exporter - метрики VPN
* Alertmanager - обработка и отправка уведомлений

**Критические алерты и меры для устранения ошибок:**

1. NodeDown (Узел недоступен)

Диагностика:

Сначала проверяем доступность узла через пинг. Если узел не отвечает, проверяем порт SSH. Затем через консоль VirtualBox смотрим, запущена ли виртуальная машина и в каком она состоянии. Проверяем, не произошел ли сбой питания или не исчерпаны ли ресурсы хоста.

Восстановление:

Если виртуальная машина остановлена, запускаем ее через консоль VirtualBox. Если она зависла, выполняем принудительный перезапуск. После восстановления подключения проверяем работу критических сервисов, особенно node\_exporter. Убеждаемся, что узел снова появляется в мониторинге.

2. HighCPUUsage (Высокая загрузка процессора)

Диагностика:

Определяем процессы с наибольшей загрузкой процессора. Проверяем общую нагрузку системы и утилизацию по ядрам. Если используются контейнеры, анализируем потребление ресурсов каждым контейнером. Смотрим историю нагрузки, чтобы понять, является ли это разовой ситуацией или постоянной проблемой.

Оптимизация:

Временно останавливаем некритичные сервисы, если это возможно. Для контейнеров ограничиваем выделенные ресурсы процессора. Повышаем приоритет критически важным процессам. Анализируем, можно ли оптимизировать работу ресурсоемких приложений.

3. OutOfMemory (Недостаток памяти)

Диагностика:

Проверяем общее использование памяти и доступную память. Ищем процессы с аномально высоким потреблением памяти. Анализируем использование swap-раздела. Проверяем, не вызвана ли проблема утечкой памяти в каком-либо приложении.

Немедленные действия:

Останавливаем процессы с утечкой памяти, если они не критические. Очищаем кэш памяти. Перезапускаем проблемные сервисы. В экстренных случаях временно создаем дополнительный swap-файл, чтобы выиграть время для решения проблемы.

4. DiskSpaceLow (Мало места на диске)

Диагностика:

Анализируем использование дискового пространства по разделам. Ищем крупные файлы и директории. Проверяем логи на предмет аномального роста. Если используются контейнеры, анализируем их использование диска. Смотрим, какие приложения генерируют много временных данных.

Очистка:

Очищаем старые логи, оставляя только актуальные данные. Удаляем кэш пакетов и временные файлы. Очищаем Docker от неиспользуемых образов и контейнеров. Архивируем и перемещаем старые данные на внешнее хранилище при необходимости.

5. HighNetworkErrors (Сетевые ошибки)

Диагностика:

Проверяем статистику сетевых интерфейсов на предмет ошибок приема и передачи. Анализируем, на каком именно интерфейсе возникают проблемы. Смотрим статистику повторной передачи пакетов. Мониторим сетевую активность в реальном времени для выявления аномалий.

Решение:

Перезапускаем проблемный сетевой интерфейс. Проверяем настройки сети в VirtualBox. Анализируем, не связана ли проблема с конфигурацией сетевого адаптера виртуальной машины.

Процесс обработки инцидента:

* Получить уведомление по email
* Проверить статус в Prometheus
* Зайти на проблемный сервер по SSH
* Выполнить диагностику согласно инструкциям выше
* После решения - отметить инцидент как обработанный в Alertmanager