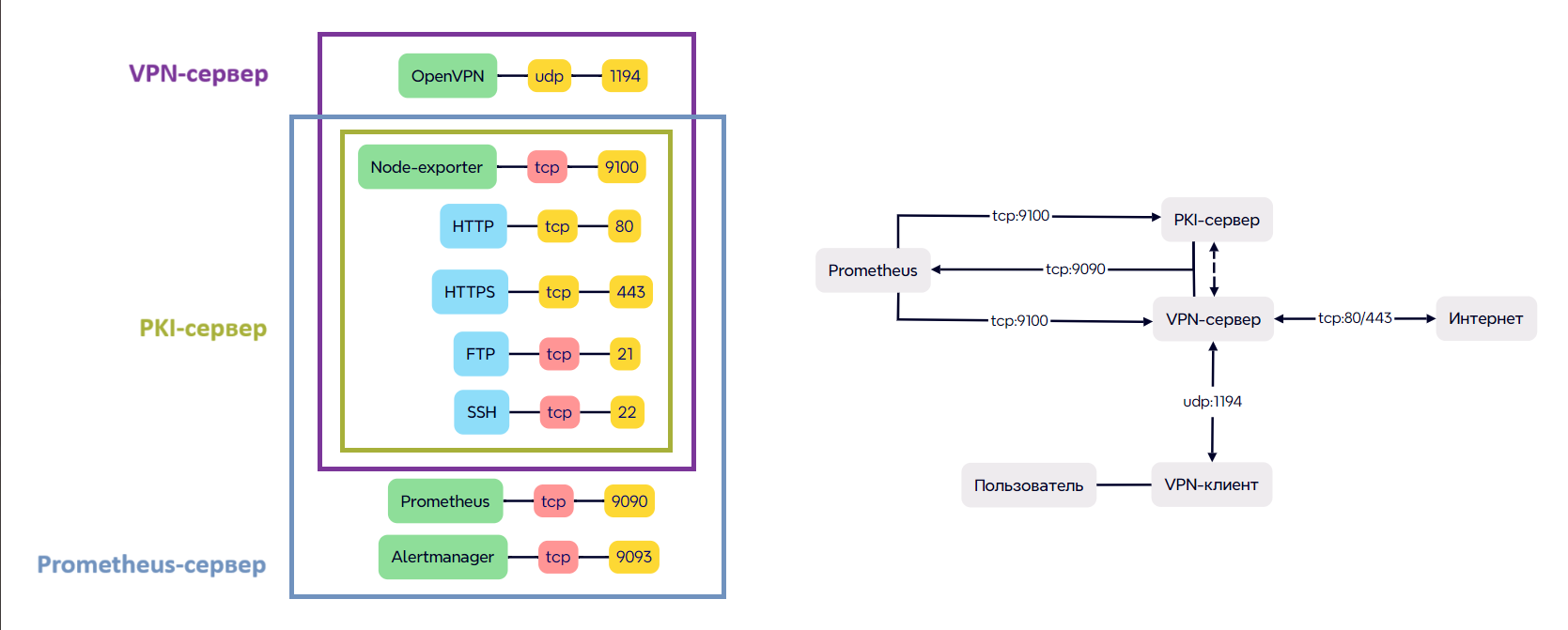
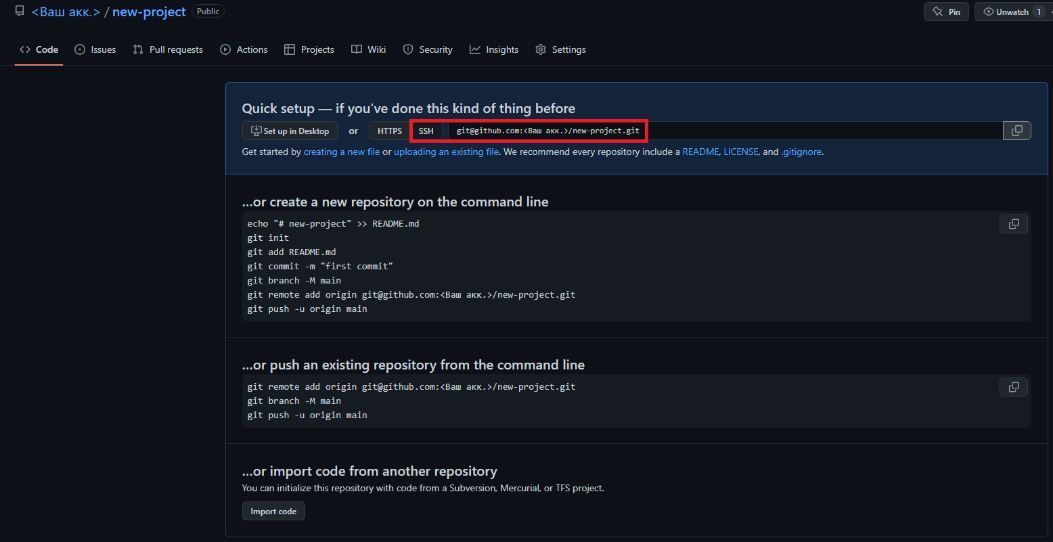
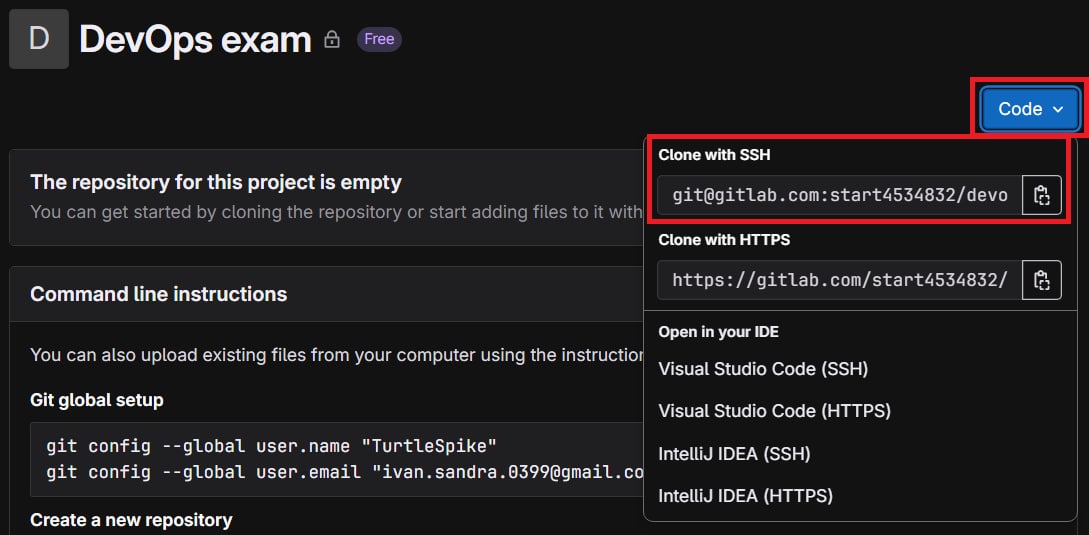
Проект представляет собой набор пакетов для развертки VPN сервера и его мониторинга (pki – для удостоверяющего центра, vpn – для самого vpn-сервера, prom – для сервера мониторинга).



(Синим цветом отмечены используемые протоколы; зеленым – используемые сервисы; красным – то, что нельзя изменить; желтым – то, что изменить можно)

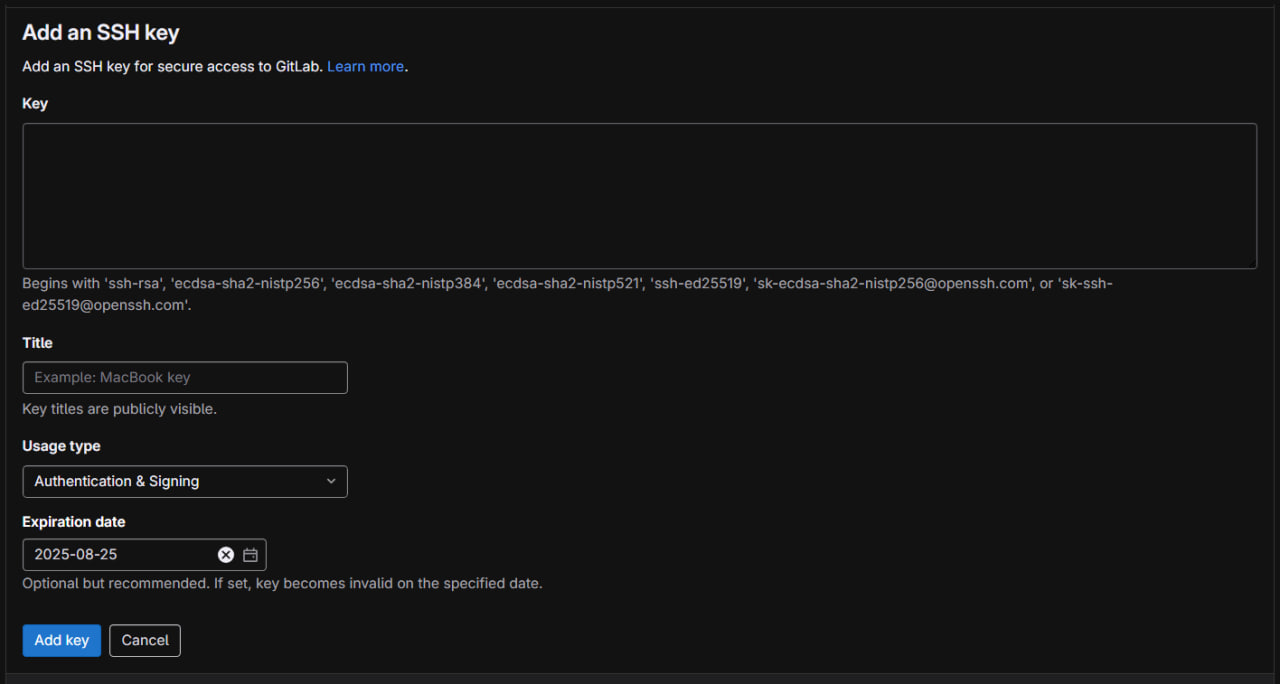
Жестких требований к создаваемым серверам нет, они только должны быть больше 10 ГБ чтобы вместить 7 резервных копий и чтобы не сработал алерт на количество дискового пространства. Ничего предварительно устанавливать не надо: при установке пакетов все необходимые приложения и утилиты сами установятся, если их нет на сервере. Необходимо только подготовить SSH-ссылки: одну для переброса файлов между серверами при установке и три уникальных для резервного копирования каждого из 3 серверов. Инструкция по получению SSH-ссылки в GitHub и GitLab соответственно расположена ниже

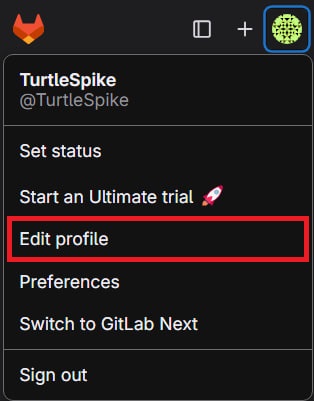
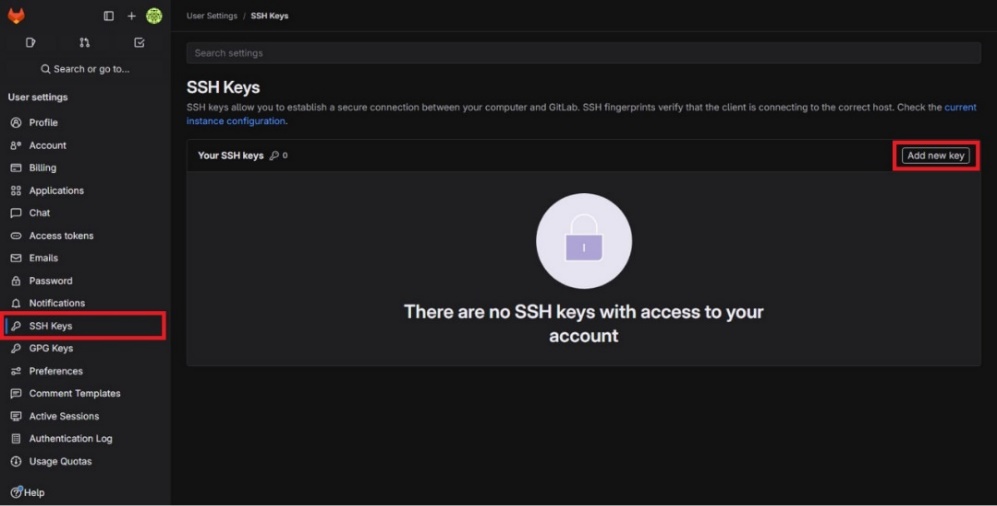


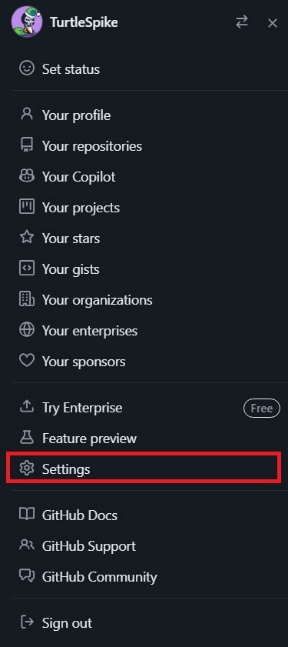
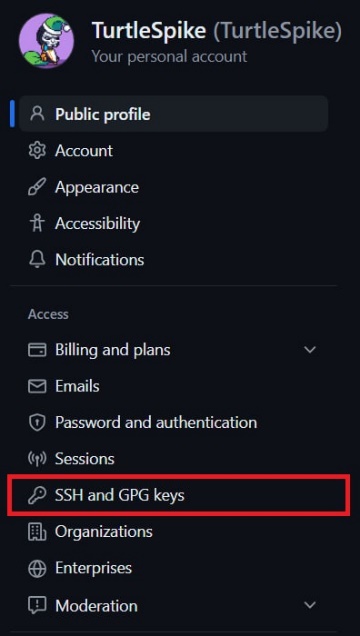


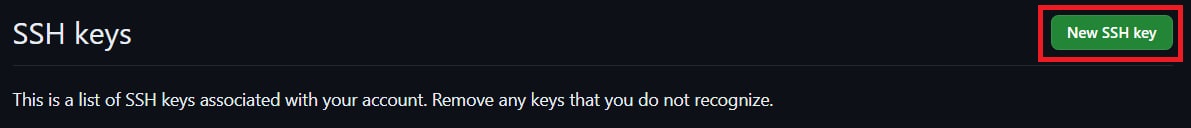
Желательно устанавливать пакеты в такой последовательности: pki > vpn > prom

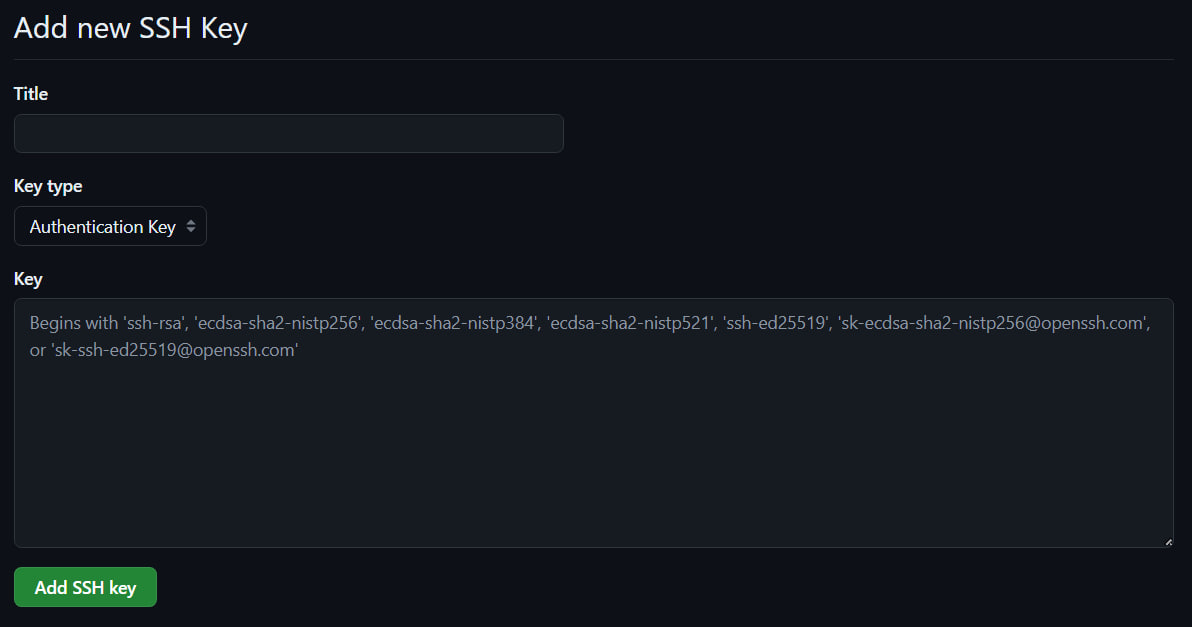
Во время установки трех пакетов система перед запросом SSH-ссылок пакет выведет на экран SSH-ключ, который необходимо ввести в сервис удаленных репозиториев для синхронизации с локальным репозиторием. Инструкция для ввода выглядит так:

— GitLab



— GitHub





Далее в консоли будет указаны действия для создания системы оповещений в Телеграм

Также во время установки пакетов будут созданы 2 директории: “/backup” для хранения резервных копий и “~/transfer” для переноса нужных системе файлов между серверами

После установки пакетов VPN-сервера и сервера удостоверяющего центра в домашней директории пользователя появится папка vpn(или pki)-setup-scripts. Цифра в названии скриптов обозначает порядковый номер запуска этого скрипта:

* 1-create-pki.sh
* инициализирует PKI (Public Key Infrastructure) на сервере Удостоверяющего Центра и создает его открытый и закрытый ключ
* параметр один: место для расположения папки “easy-rsa”
* 2-setup-firewall.sh
* настраивает фаерволл
* параметров нет
* 3-create-vpn.sh
* инициализирует PKI на VPN-сервере
* параметр один: место для расположения папки “easy-rsa”
* 4-create-server-req.sh
* создает запрос на подписание сертификата будущего сервера и отправляет его в удаленный репозиторий
* параметр один: желаемое название сервера
* 5-create-client-req.sh
* создает запрос на подписание сертификата будущего клиента сервера и отправляет его в удаленный репозиторий
* параметров 1 или 2 в зависимости от ситуации: если папка “clients” уже существует – параметр один: желаемое название клиента. Иначе первый параметр – желаемое расположение папки, второй параметр – желаемое название клиента
* используется для создания нового клиента
* 6-create-crypt-key.sh
* используется для создания криптоключа с целью повышения уровня безопасности
* параметров нет
* 7-sign-server-req.sh
* подписывает сертификат сервера
* параметр один: название сервера, запрос которого необходимо подписать
* 8-sign-client-req.sh
* подписывает сертификат клиента
* параметр один: название клиента, запрос которого необходимо подписать
* используется для создания нового клиента
* 9-pack-keys.sh
* собирает необходимые ключи и сертификаты в одной директории
* параметр один: название недавно подписанного клиента
* используется для создания нового клиента
* 10-setup-server-conf.sh
* подготавливает конфиг сервера и настраивает фаерволл
* параметр один: название недавно подписанного сервера
* 11-create-client-configs.sh
* подготавливает заготовку конфига для клиента
* параметров нет
* 12-make-ovpn-file.sh
* создает пользовательский “.ovpn” файл
* параметр один: название недавно подписанного клиента
* используется для создания нового клиента

Проверить результат создания сервера можно с помощью команды «sudo systemctl status openvpn-server@\*название сервера\*.service» на VPN-сервере

Помимо скриптов для развертки VPN’а, на эти 2 сервера еще устанавливаются и сразу активируются 2 systemd-юнита резервного копирования: auto-backup.timer и auto-backup.service – а также скрипт самого копирования: auto-backup.sh.

Список устанавливаемых файлов на сервер мониторинга немного отличается. Помимо 3 файлов резервного копирования, пакет устанавливает:  
- ovpn.yml – файл с правилами мониторинга для Prometheus  
- alertmanager-tg.yml – файл с кодом для настройки оповещений об алертах в Телеграм, который будет вставляться в уже существующий файл /etc/Prometheus/alertmanager.yml  
- prometheus-node-paste.txt – файл с текстом, который будет вставляться в уже существующий файл /etc/prometheus/prometheus.yml

Ovpn.yml содержит следующие метрики:  
- CPU\_Overload – загруженность процессора за 5 минут больше количества ядер  
- LowDiskSpace – количество места на диске меньше 10%  
- LowMemory – количество оперативной памяти меньше 15%  
- OpenVPN\_Down – OpenVPN неактивен (метрика эксклюзивна для VPN-сервера)  
- auto\_backup\_timerDown – таймер срабатывания сервиса резервного копирования auto\_backup неактивен

Последние 2 файла содержат по 2 строки, начинающиеся с нескольких знаков “#”. Сами строки трогать запрещается, т.к. они используются для удаления кода между ними. Между самими строками можно писать что угодно, однако необходимо учесть, что при удалении пакета код между строками будет удален вместе со всеми строками

При удалении любого из пакетов система будет спрашивать пользователя хочет ли он удалить то или иное приложение. Если ответ “да”, то скрипт удалит как сам указанный пакет, так и все пользовательские файлы, созданные при развертке инфраструктуры, в корневой папке пакета. Если ответ “нет”, то скрипт только откатит все внесенные в ключевые файлы изменения.