# Финальный проект

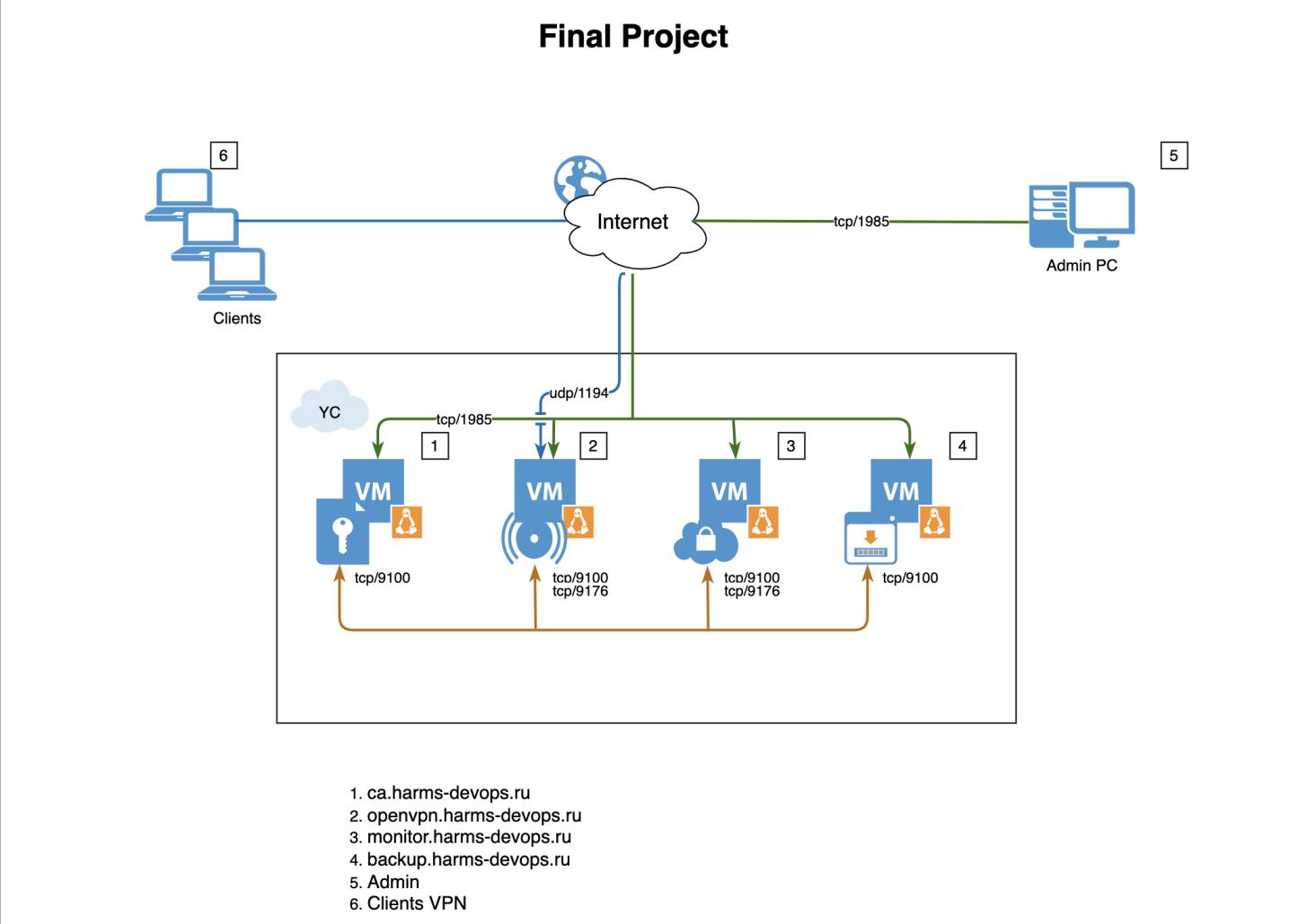
# Руководство администратора инфрастуктуры

1. **Общие сведения**

Создание начальной инфрастуктуры для централизованного управления учетными записями для Финального проекта (далее, ИФП)

1. **Архитектура ИФП**

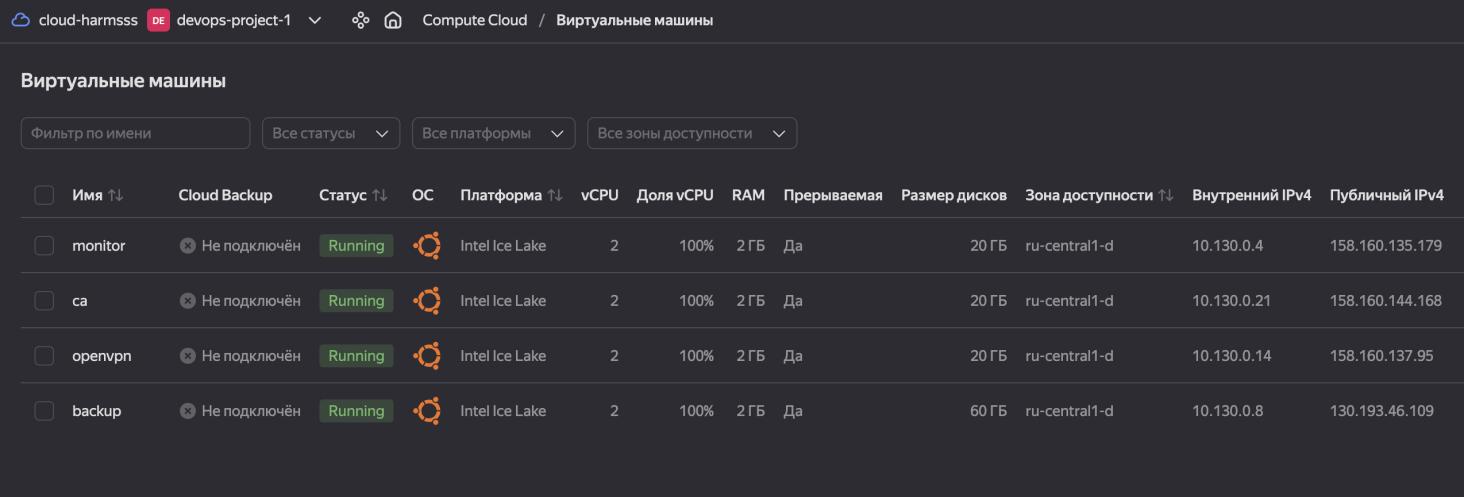
Архитектура представлена на следующем рисунке



1. **Состав программно-технический средств**

Система ИФП размещается на вычислительных мощностях облачного провайдера YandexCloud. Система ИФП состоит из следующих подсистем развернутых на отдельных виртуальных серверах:

* Сервер Сертификации (ca);
* Сервер VPN (openvpn);
* Сервер мониторинга (monitor);
* Сервер создания и хранения резервных копий (backup).



На хостинг-провайдере Рег.ру зарегистрирован домен harms-devops.ru и каждому серверу присвоены имена (A-запись для Subdomain)

* 1. **Сервер сертификации**

|  |  |
| --- | --- |
| **Параметр** | **Значение** |
| Имя сервера | **ca.harms-devops.ru** |
| Публичный IPv4-адрес (статический) | 158.160.144.168 |
| Внутренний IPv4-адрес | 10.130.0.21 |
| OS | Ubuntu 22.04 LTS |
| Платформа | Intel Ice Lake |
| vCPU | 2 |
| RAM | 2 ГБ |
| Объём дискового пространства | 20 ГБ |

* 1. **Сервер OpenVPN**

|  |  |
| --- | --- |
| **Параметр** | **Значение** |
| Имя сервера | **openvpn.harms-devops.ru** |
| Публичный IPv4-адрес (статический) | 158.160.137.95 |
| Внутренний IPv4-адрес | 10.130.0.30 |
| OS | Ubuntu 22.04 LTS |
| Платформа | Intel Ice Lake |
| vCPU | 2 |
| RAM | 2 ГБ |
| Объём дискового пространства | 20 ГБ |

* 1. **Сервер мониторинга**

|  |  |
| --- | --- |
| **Параметр** | **Значение** |
| Имя сервера | **monitor.harms-devops.ru** |
| Публичный IPv4-адрес (статический) | 158.160.135.179 |
| Внутренний IPv4-адрес | 10.130.0.4 |
| OS | Ubuntu 22.04 LTS |
| Платформа | Intel Ice Lake |
| vCPU | 2 |
| RAM | 2 ГБ |
| Объём дискового пространства | 20 ГБ |

* 1. **Сервер Backup**

|  |  |
| --- | --- |
| **Параметр** | **Значение** |
| Имя сервера | **backup.harms-devops.ru** |
| Публичный IPv4-адрес | 130.193.46.109 |
| Внутренний IPv4-адрес | 10.130.0.8 |
| OS | Ubuntu 22.04 LTS |
| Платформа | Intel Ice Lake |
| vCPU | 2 |
| RAM | 2 ГБ |
| Объём дискового пространства | 20 ГБ + 40 ГБ (BTFRS) |

##### Автоматизация установки и настройки

4.1 Перечень основных скриптов необходимых для установки:

**vm-start.sh** – скрипт для первоначальной настройки сервера (устанавливается на каждый сервер).

Скрипт выполняет следующие действия:

- меняет timezone, проверяет, и при необходимости, устанавливает необходимые недостающие пакеты, изменяет/устанавливает пароль для пользователя, создает нового пользователя, добавляя его в группу sudo;

- редактирует файл настроек */etc/ssh/sshd\_config*: изменяет порт подключения по ssh, запрещает подключение под root, разрешает аутентификацию по ключу, запрещает пустые пароли, запрещает аутентификацию по паролю, задает время жизни ssh подключения. Отключает ipv6;

- добавляет глобальные переменные для сборки/пересборки пакетов;

- настраивает правила iptables: закрывает все, кроме необходимых портов (80, 53, 123, 443, ssh, lo).

- создает необходимые папки.

**Важно!** После выпонения скрипта, при изменении стандартного порта 22 для подключения по ssh на новый, необходимо повторно подключиться к серверу для продолжения настройки и конфигурирования.

**easy-rsa.sh** – скрипт для автоматизированной настройки сервера сертификации.

Скрипт выполняет следующие действия:

- проверяет наличие установленного ПО Easy-rsa, и при необходимости, скачивает пакет для установки/переустановки Easy-rsa;

- cоздает корневой сертификат удостоверяющего центра.

- создает симвалическую ссылку */home/$username/easy-rsa*, изменяет права *700*, назначает владельцем группу и пользователя *$username*

**ca-crt-key.sh** - скрипт для автоматизации выпуска, копирования сертификатов и ключей для серверов и клиентов (используется на сервере CA)

Имеется меню для выбора действий:

- копирование корневого сертификата на заданный сервер;

- создание пары ключа и сертификата для заданного сервера;

- копирование серверной пары на заданный сервер;

- создание и копирование клиентской пары с заданым именем на сервер vpn;

- копирование пары выбранного сервера на сервер мониторинга.

**openvpn.sh** - скрипт для автоматизированной настройки сервера vpn

Скрипт выполняет следующие действия:

- проверяет наличие установленного ПО Openvpn, и при необходимости, скачивает пакет для установки/переустановки Openvpn. Запращивает файлы сертифаката, ключа сервера, корневого сертификата и перемещает в необходимые директории, генерирует tls-crypt ключ для клиентов;

- настраивает server.conf: задает пути до сертификатов и ключей. Включает маршрутизацию для ipv4. Устанавливает порт подключения;

- настраивает *base.conf*: устанавливает внешний IP и порт;

- добавляет правила для iptables: открывает порт для подключения, разрешает переадресацию трафика и настраивает маскарадинг для nat;

- создает конфигурационный файл для клиента и копирует его в директорию для хранения конфигурационный файлов клиентов.

**make-config.sh** - скрипт для создания openvpn конфига для клиента (используется на openvpn сервере). Запускается из скрипта *openvpn.sh*, при необходимости можно запускать отдельно.

**prometheus.sh** - скрипт для автоматизированной настройки сервера мониторинга

Скрипт выполняет следующие действия:

- проверяет наличие установленного ПО Prometheus и Alertmanager, и при необходимости, скачивает пакеты для установки/переустановки Prometheus и Alertmanager;

- добавляет правила iptables: 9090 - Prometheus, 9093 - alertmanager, 587 - smtp, 9100 - exporter, 9176 - openvpn\_exporter;

- добавляет сертификаты для сервера и эксопртеров в нужные директории;

- редактирует конфигурационный файл prometeus.yml: устанавливает имя пользователя, пароль, сертификат, имя хоста.

- настраивает dns в файле */etc/hosts* для подключения к экспортерам через приватную сеть по доменным именам.

**backup-server.sh** - скрипт для автоматизированной настройки сервера backup

Скрипт выполняет следующие действия:

- создает, при необходимости, новый раздел btrfs на дополнительном диске;

- монтирует диск в директорию */backup*, если отсутствует папка, создает ее;

- проверяет наличие установленного ПО UrBackup, и при необходимости, устанавливает/переустанавливает из собственного репозитория *ppa:uroni/urbackup*;

- добавляет правила iptables: input / 55413 - fastcgi, 55414 - http, 55415 - internet client; output / 35621 - send files, 35622 (udp) - udp broadcasts, 35623 - commands and image backup.

4.2 Дополнителные скрипты для донастройки серверов:

**exporters.sh** - скрипт для добавления экспортеров для каждого сервера (устанавливается на каждый сервер)

Имеется меню для выбора действий:

1. Установка node\_exporter

- проверяет наличе установленного экспортера и при необходимости устанавливает/переустанавливает его;

- копирует сертификат и ключ в нужную дерикторию, устанавливает права *640* и задает владельца и группу *node\_exporter*;

- устанавливает пароль для пользователя *admin* в конфиге *web.yml*;

- добавляет правила iptables: 9100 - node exporter.

1. Установка openvpn\_exporter (устанавливается только на vpn сервер)

- проверяет наличе установленного openvpn\_экспортера и при необходимости устанавливает/переустанавливает его;

- добавляет правила iptables: 9176 - openvpn\_exporter.

**backup-client.sh** - скрипт для установки UrBackup клиента (устанавливается на каждый сервер, кроме backup сервера)

- проверяет наличе установленного клиента urbackup и при необходимости устанавливает/переустанавливает его;

- отключает в конфигурационном файле internet подключения (при необходимости можно не отключать);

- добавляет папки для бэкапа;

- добавляет правила для iptables: output / 35621, 35622 (udp), 35623

4.3 Перечень пакетов

Для автоматизации настройки серверов созданы deb-пакеты, которые размещены в созданном репозитории на *github.com* (*https://github.com/harms-danil/Devops\_final\_project\_1*):

**[easy-rsa-harms\_0.2\_all.deb](https://github.com/harms-danil/Devops_final_project_1/blob/main/deb/easy-rsa-harms_0.2_all.deb" \o "easy-rsa-harms_0.2_all.deb)** - пакет для установки Easy-rsa

Содержит:

- ~/easy-rsa/vars - конфигурационный файл

- /home/nikolay/easy-rsa/pki - созданная инфрастуктура pki очищенная от лишних файлов

**[openvpn-harms\_2.5.5\_amd64.deb](https://github.com/harms-danil/Devops_final_project_1/blob/main/deb/openvpn-harms_2.5.5_amd64.deb" \o "openvpn-harms_2.5.5_amd64.deb)** - пакет для установки OpenVPN.

Содержит:

- /etc/openvpn/server/server.conf - готовый конфигурационный файл для OpenVPN-сервера

- /etc/openvpn/clients\_conf - среду для создания клиентских конфигурационных файлов со скриптом make\_config.sh и файлом base.conf

**[prometheus-harms\_2.55.1.linux-amd64\_all.deb](https://github.com/harms-danil/Devops_final_project_1/blob/main/deb/prometheus-harms_2.55.1.linux-amd64_all.deb" \o "prometheus-harms_2.55.1.linux-amd64_all.deb)** - пакет для установки Prometheus.

Содержит:

*- /etc/prometheus/console\_libraries/*

*- /etc/prometheus/consoles/*

*- /etc/prometheus/prometheus.yml - конфигурационный файл*

*- /etc/prometheus/web.yml - настройка https*

*- /usr/bin/prometheus*

*- /usr/bin/promtool*

*- /etc/systemd/system/prometheus.service - systemd-юнит*

А так же *postinst* скрипт для добавления технического пользователя и группы *prometheus, установки прав, включения/запуска сервиса, создания симвалических ссылок* и *prerm* скрипты для остановки/выключения сервиса, удаления поьзователя и группы, симвалических ссылок

**[alertmanager-harms\_0.27.0.linux-amd64\_all.deb](https://github.com/harms-danil/Devops_final_project_1/blob/main/deb/alertmanager-harms_0.27.0.linux-amd64_all.deb" \o "alertmanager-harms_0.27.0.linux-amd64_all.deb)** - пакет для установки Alertmanager для Prometheus

Содержит:

*- /usr/bin/alertmanager*

*- /usr/bin/amtool*

- */etc/prometheus/alertmanager.yml - конфигурационный файл*

*- /etc/prometheus/rules.yml - файл с правилами*

*- /etc/systemd/system/prometheus-alertmanager.service - systemd-юнит*

А так же *postinst* скрипт для добавления технического пользователя и группы *prometheus, установки прав, включения/запуска сервиса, создания симвалических ссылок* и *prerm* скрипты для остановки/выключения сервиса, удаления поьзователя и группы, симвалических ссылок

**[node-exporter-harms\_1.8.2-1\_all.deb](https://github.com/harms-danil/Devops_final_project_1/blob/main/deb/node-exporter-harms_1.8.2-1_all.deb" \o "node-exporter-harms_1.8.2-1_all.deb)** - пакет для установки node exporter

Содержит:

- */opt/node\_exporter/web.yml* - настройки https

- */usr/bin/node\_exporter*

*- /etc/systemd/system/node\_exporter.service* - *systemd-юнит*

А так же *postinst* скрипт для добавления технического пользователя и группы *node\_exporter, установки прав, включения/запуска сервиса* и *prerm* скрипты для остановки/выключения сервиса, удаления поьзователя и группы

**[openvpn-exporter-harms\_0.3.0\_all.deb](https://github.com/harms-danil/Devops_final_project_1/blob/main/deb/openvpn-exporter-harms_0.3.0_all.deb" \o "openvpn-exporter-harms_0.3.0_all.deb)** - пакет для установки openvpn exporter

Содержит:

- /usr/bin/openvpn\_exporter

- /etc/systemd/system/openvpn\_exporter.service - *systemd-юнит*

А так же *postinst* скрипт для добавления технического пользователя и группы *openvpn\_exporter, установки прав, включения/запуска сервиса* и *prerm* скрипты для остановки/выключения сервиса, удаления поьзователя и группы

4.4 Порядок настройки серверов

Для разворачивания инфраструктуры используем облачные сервисы от YandexCloud. Создание виртуальных машин производим из вэб-интерфейса (можно использовать Интерфейс командной строки Yandex Cloud (CLI)). Перед созданием необходимо сгенерировать ключи для соединения по ssh, публичный ключ добавляется при создании виртуальных машин.

*ssh-keygen -t ed25519 -C DevOps\_1\_finalproject*

После запуска сервера необходимо подключиться по ssh и продолжить настройку. Для автоматизированной настройки необходимо загрузить скрипты на сервер:

*wget -P ~/ \   
[https://raw.githubusercontent.com/harms-danil/Devops\_final\_project\_1/refs/heads/main/"$name\_script](https://raw.githubusercontent.com/harms-danil/Devops_final_project_1/refs/heads/main/$name_script.sh)”.sh*

*где, “$name\_script” -* имя нужного скрипта

и после, их запустить, предварительно добавив права на запуск (chmod +x \*.sh)

**4.4.1 Сервер ca.harms-devops.ru**

Сервер сертификации построен на базе ПО Easy-rsa

*sudo ./vm-start.sh*

*sudo ./easy-rsa.sh*

*sudo ./exporters.sh* (в скрипте выбрать node exporters)

*sudo ./backup-client.sh* (выбрать 5 пункт, без создания snapshot)

**4.4.2 Сервер openvpn.harms-devops.ru**

Сервер vpn построен на базе ПО OpenVpn

*sudo ./vm-start.sh*

*sudo ./openvpn.sh*

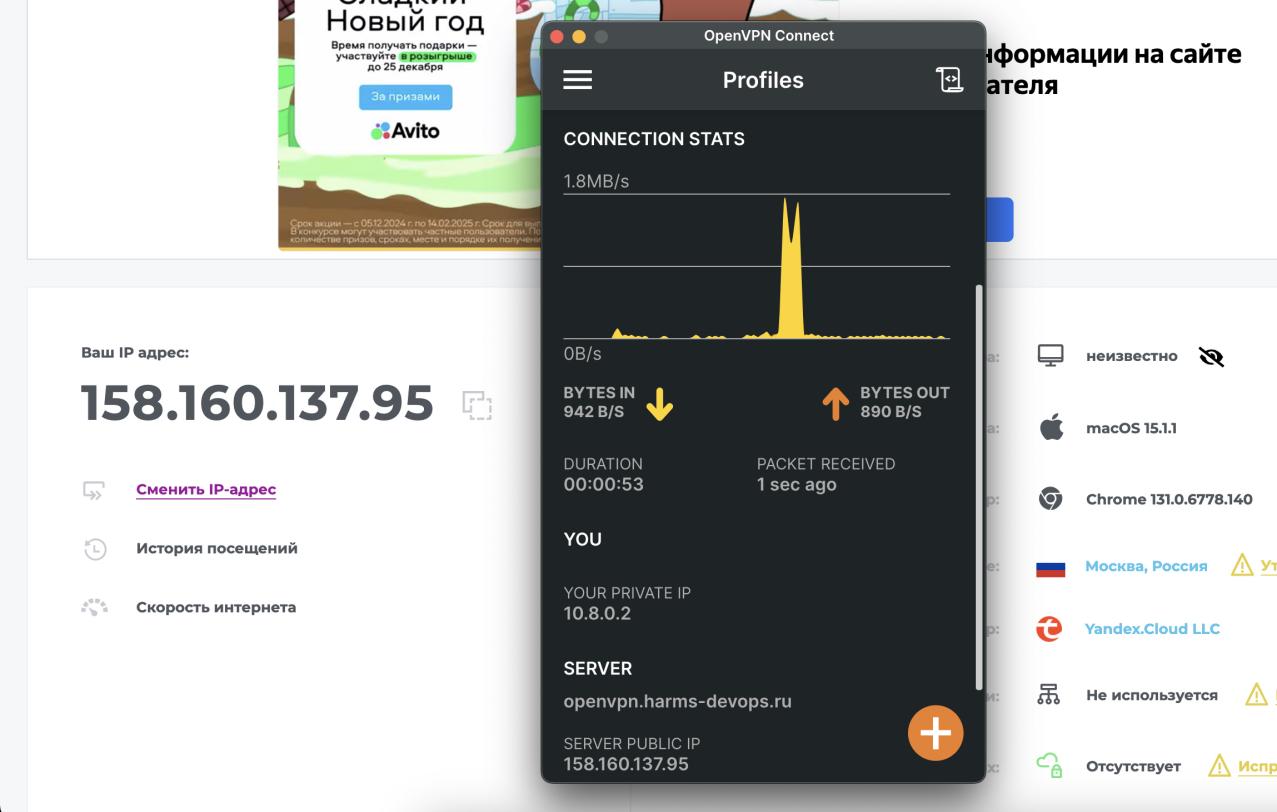
*sudo ./exporters.sh* (в скрипте выбрать node exporters и vpn exporters)

*sudo ./backup-client.sh* (выбрать 5 пункт, без создания snapshot)

Для выпуска клиентского конфигурационного фала необходимо предварительно выпустить сертификат и ключ пользователя и передать их на данный сервер.

Для подключения клиентов к серверу необходимо передать созданный файл *$client\_name.opvn* (*$HOME/client\_config/$client\_name.ovpn*) на клиентскую машину.

Работа подключенного vpn клиента с адресом vpn сервера представлена на рисунке



**4.4.3 Сервер monitor.harms-devops.ru**

Сервер мониторинга построен на базе ПО Prometheus и Prometheus Alertmanager

*sudo ./vm-start.sh*

*sudo ./prometheus.sh*

*sudo ./exporters.sh* (в скрипте выбрать node exporters)

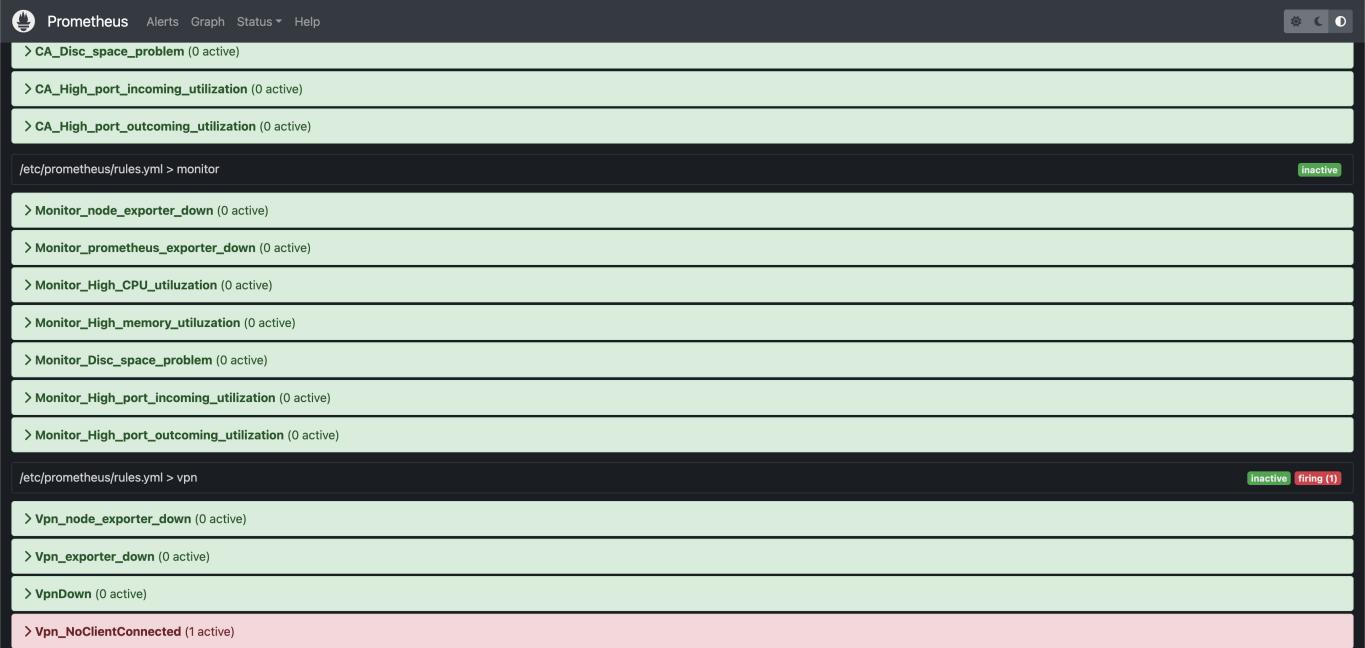
*sudo ./backup-client.sh* (выбрать 5 пункт, без создания snapshot)

Prometheus доступен по адресу *[https://monitor.harms-devops.ru:9090/](https://sb-mon.ppb-sb.ru:9090/)*

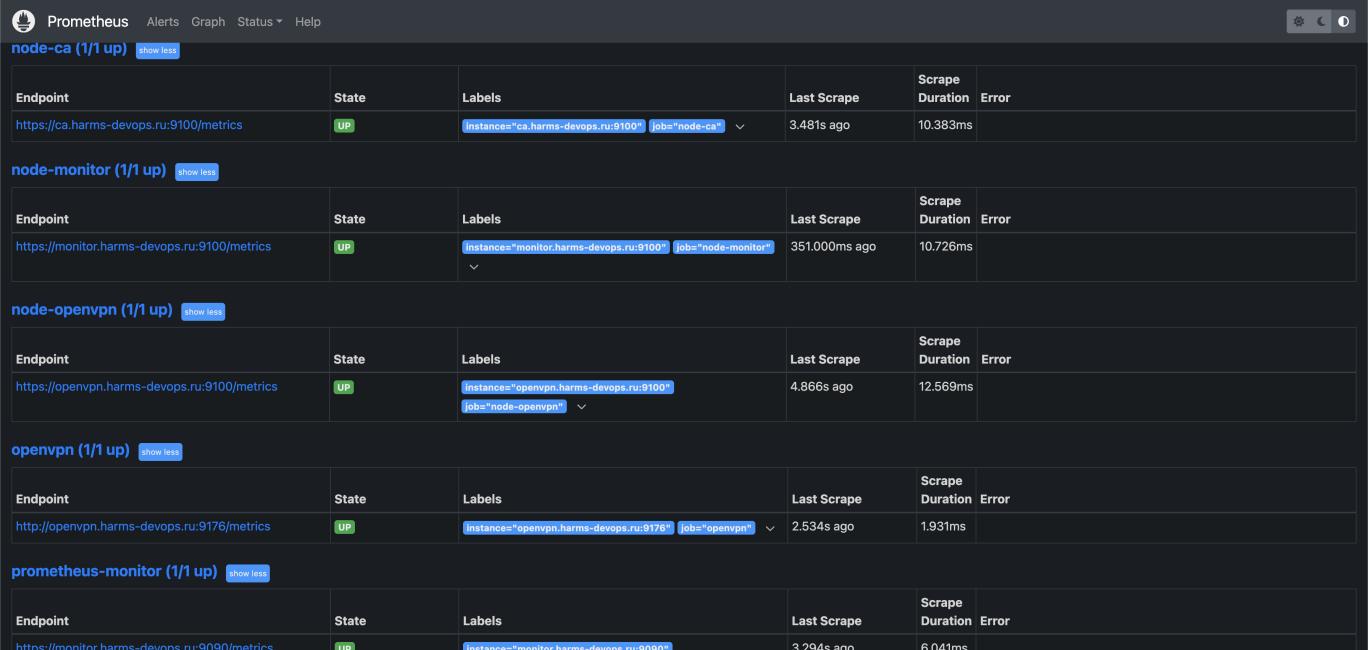
Доступ осуществляется по логину/паролю *admin/\*\*\**

Алерты отправляются на почту *[harmsss@yandex.ru](mailto:ggenoptik@yandex.ru)*

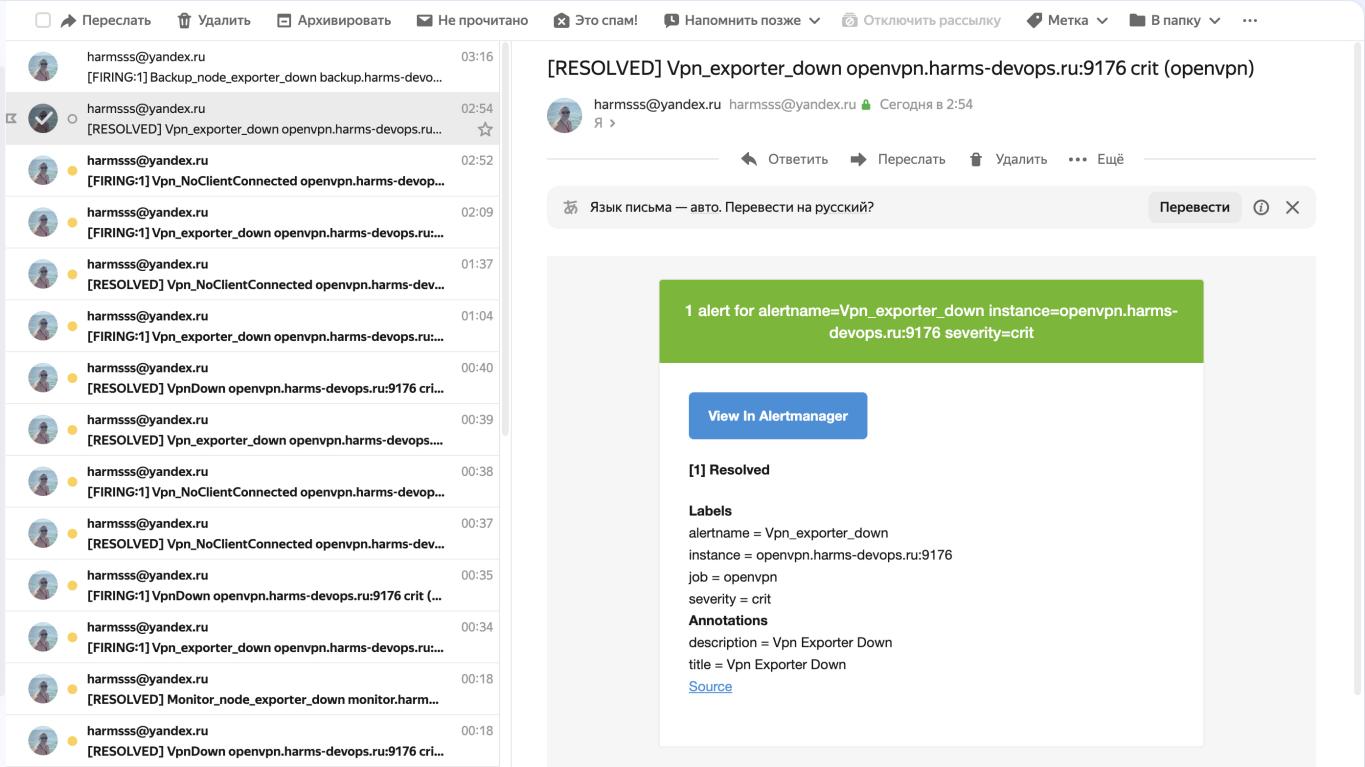
Алерты в Prometheus



Targets в Prometheus



Оповещения на почте



**4.4.4 Сервер backup**

Сервер резервного копирования построен на базе ПО UrBackup

*sudo ./vm-start.sh*

*sudo ./backup-server.sh*

*sudo ./exporters.sh* (в скрипте выбрать node exporters)

Далее донастройка производится в вэб-интерфейсе, расположеном по адресу *<http://backup.harms-devops.ru:55414/>*

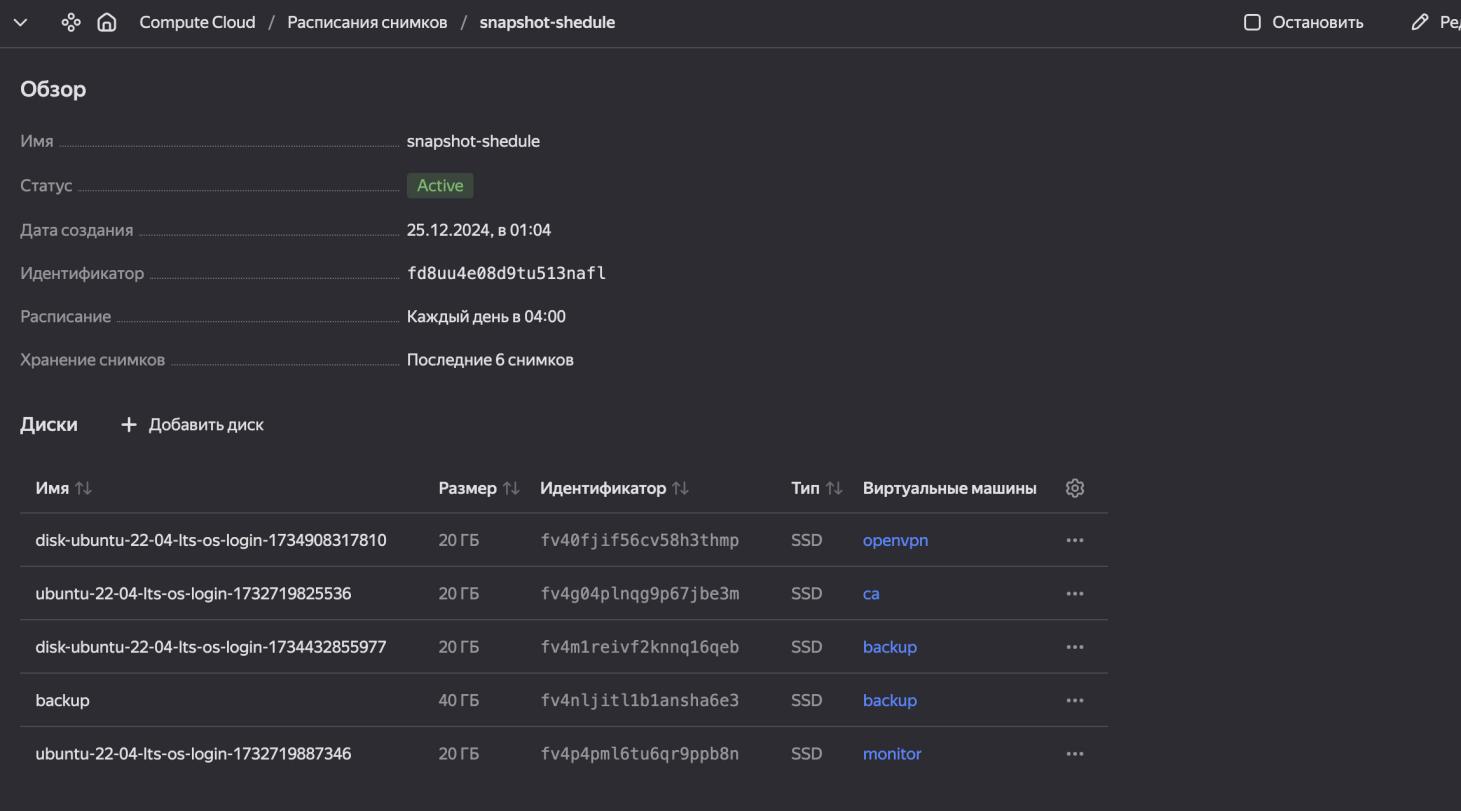
При первом входе необходимо создать пользователя для дальнейшего входа по паролю. Для того, чтобы клинты автоматически определялись, необходимо добавить подсказки для обнаружения клиентов на главной странице. Клиенты внутри подсети определяются автоматически. При необходимости, можно дополнительно доваить клиента за приделами NAT.

Клиенты добавлены в созданную группу для общего редактирования правил. Для клиентов выбраны пути для сохранения резервных копий /, создание snapshots и образов отключено.

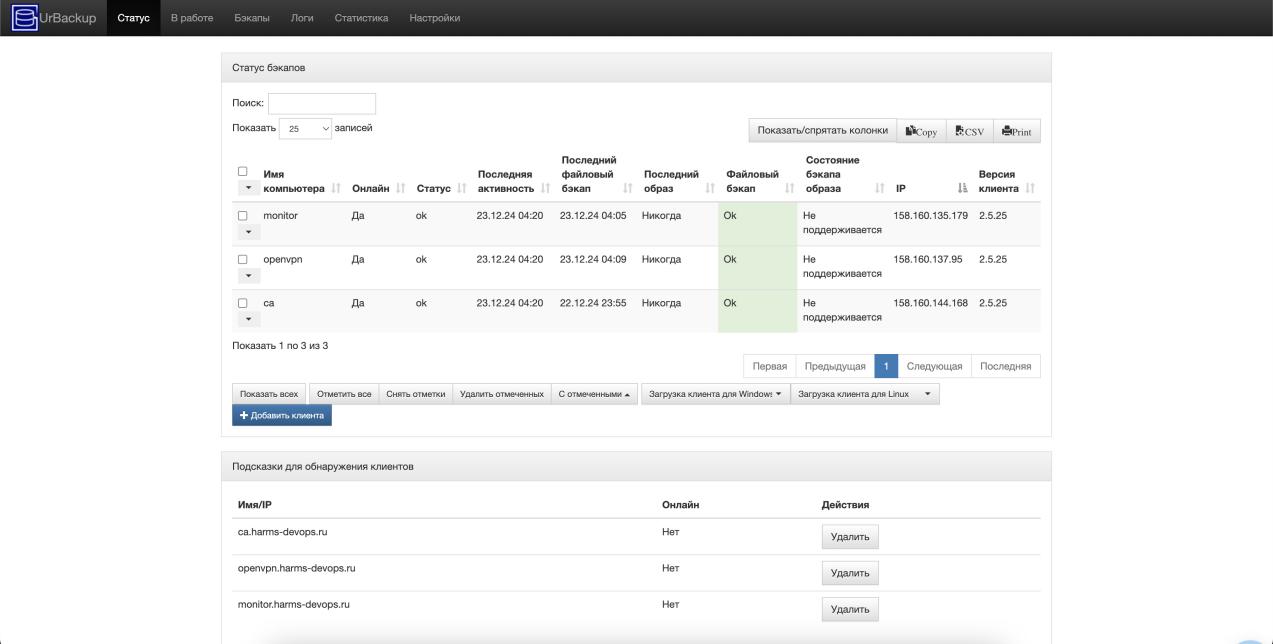
Описание настроек сервера *<https://www.urbackup.org/administration_manual.html>*

Дополнительно, для резервного копирования инфраструктуры можно использовать внутренние механизмы yandex cloud, но для этого необходимо производить заморозку файловой системы на время создания снимка.

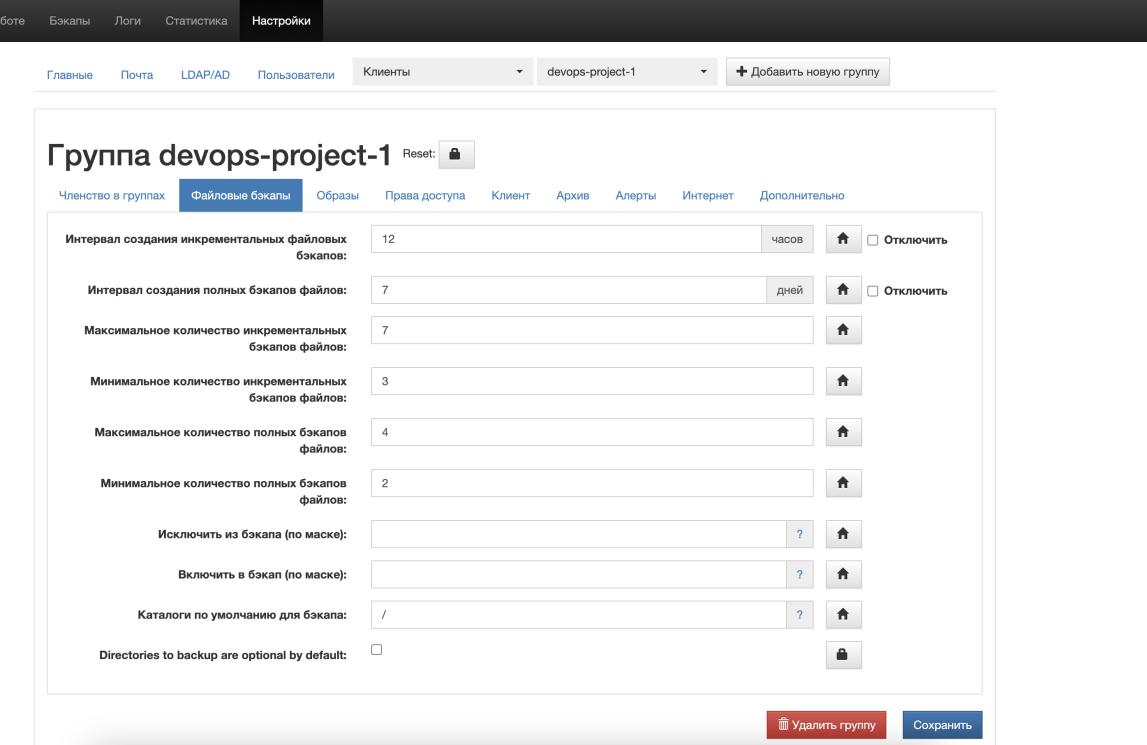
В нашем случае мы создаем такую сущность как snapshot-schedule. Это правило по которому будет происходить бэкапирование (делатся snapshot) наших дисков на серверах.

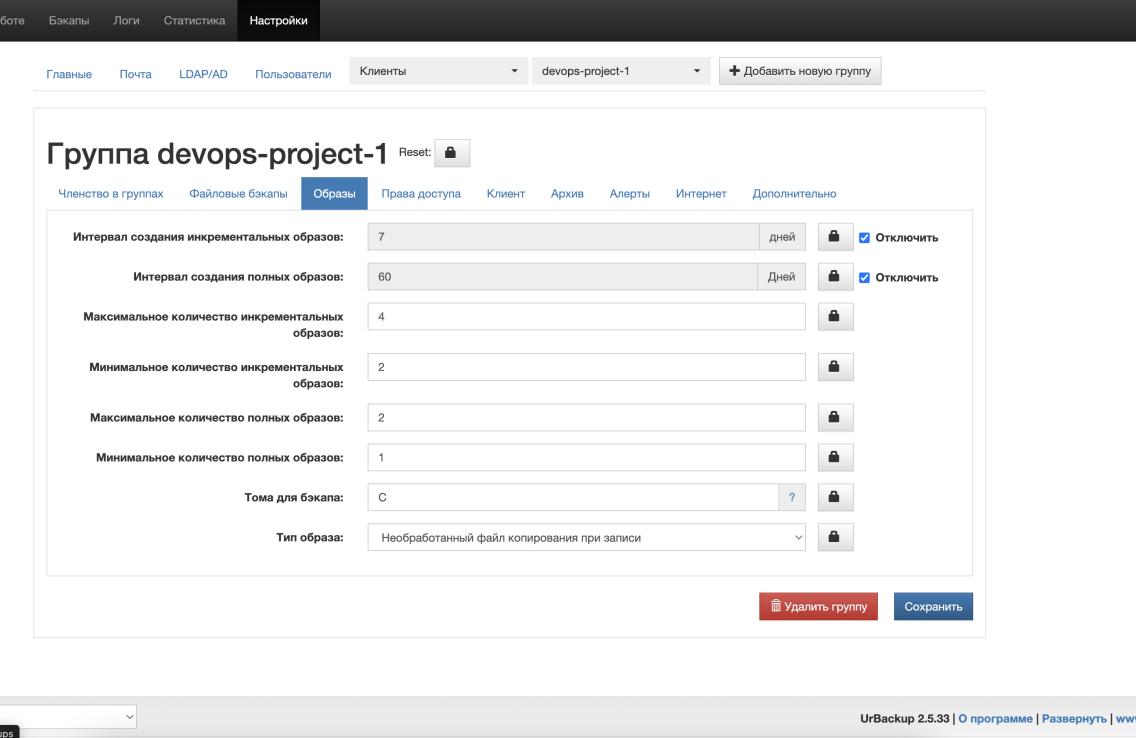


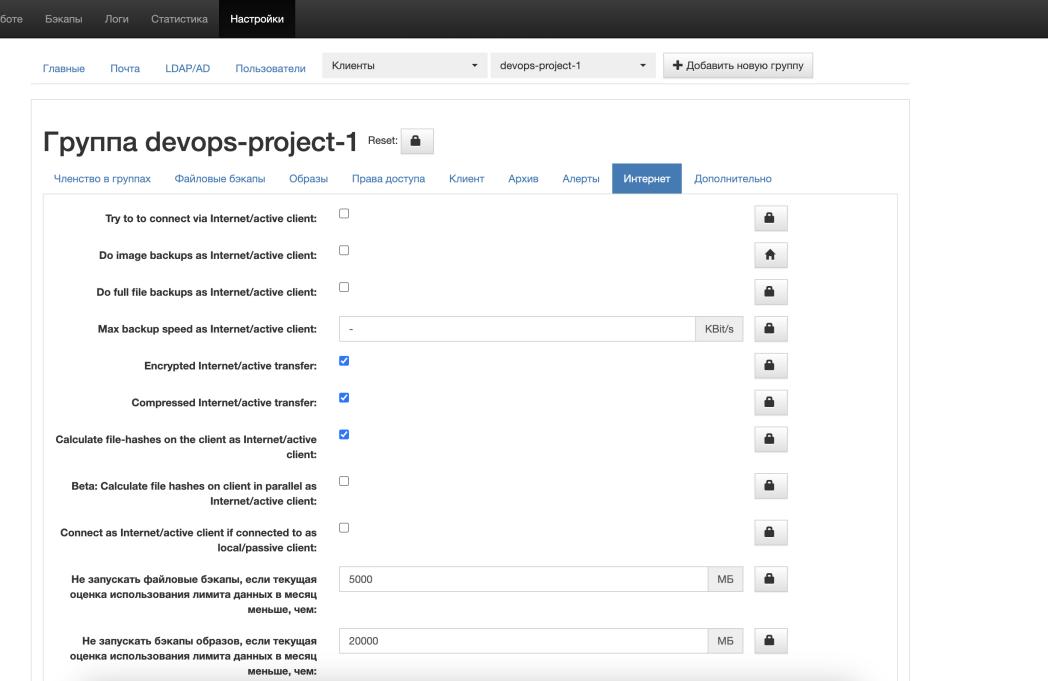
Общий обзор внешнего вида веб-интерфейса



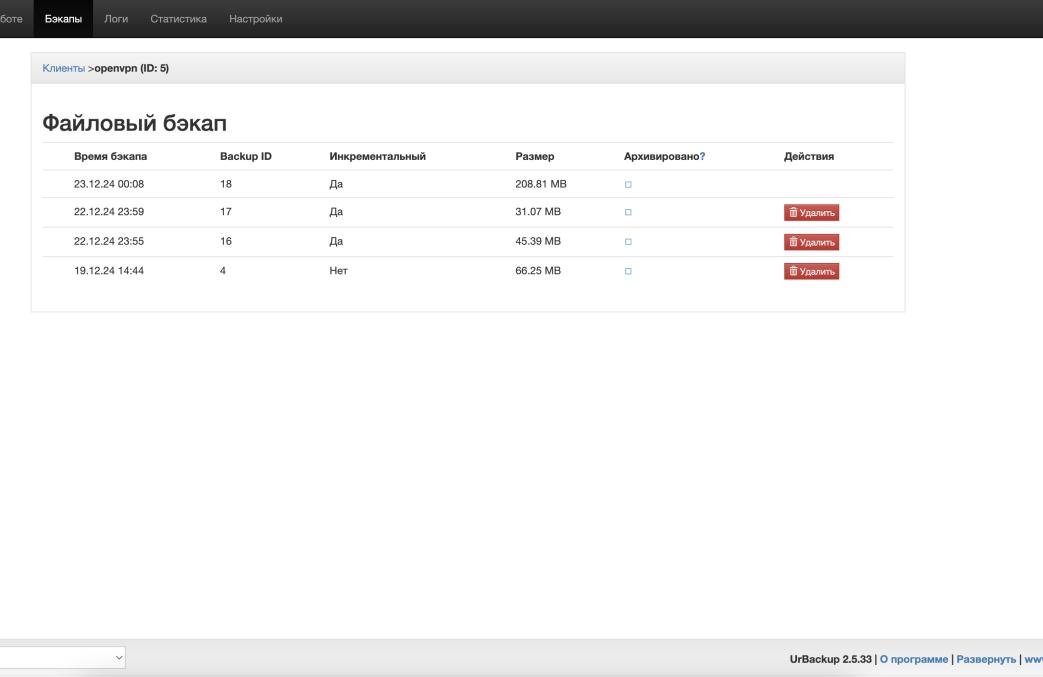
Настройки для группы







Доступные бэкапы, для примера, для сервера сертификации



Пример работы создания полного файлового бэкапа

