Необходимо разработать и настроить инфраструктуру информационно коммуникационной системы согласно предложенной топологии (см. Рисунок 3).

Задание Модуля 3 содержит миграцию пользователей, развёртывание и настройку центра сертификации, выдачу сертификатов веб серверам для шифрования трафика, настройку шифрованного туннеля, настройку межсетевого экрана, принт-сервера, сервера логирования и мониторинга, автоматизации на основе инфраструктуры открытых ключей, настройку защиты протокола ssh от перебора, настройку программного обеспечения для создания архивных копий

В ходе проектирования и настройки сетевой инфраструктуры следует заносить записи в отчет о своих действиях, когда это требуется в задании.

Отчет по окончании работы следует сохранить на диске рабочего места и задать имя файла без учёта расширения - ФамилияУчастникаМодуль3

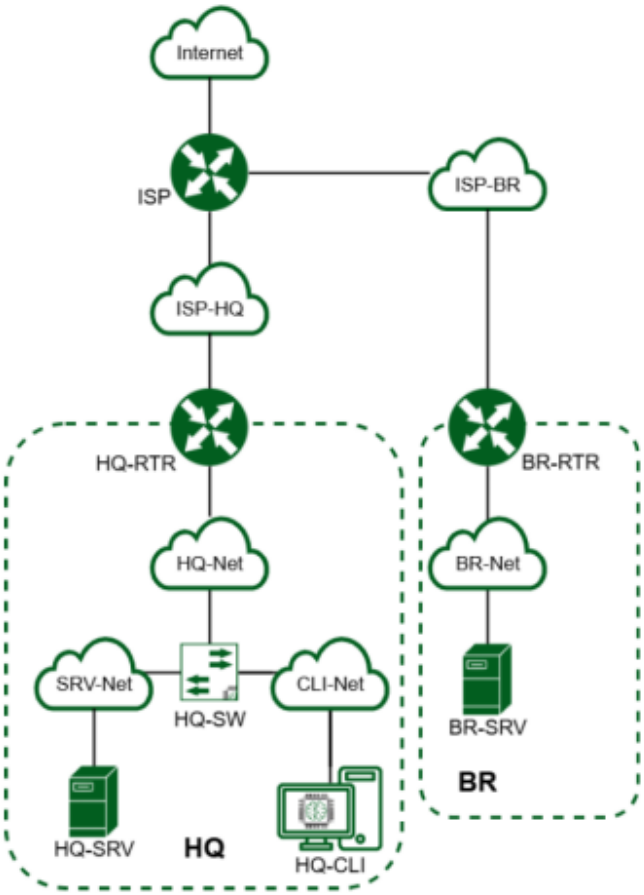


Рисунок 3. Топология сети

1. Выполните импорт пользователей в домен au-team.irpo:

— В качестве файла источника выберите файл users.csv располагающийся в образе Additional.iso

— Пользователи должны быть импортированы со своими паролями и другими атрибутами

— Убедитесь, что импортированные пользователи могут войти на машину HQ-CLI

Выполняем импортирование пользователей на сервере BR-SRV:

Монтируем образ Additional.iso:

mount /dev/cdrom /mnt

Проверяем, что users.csv имеет данные:

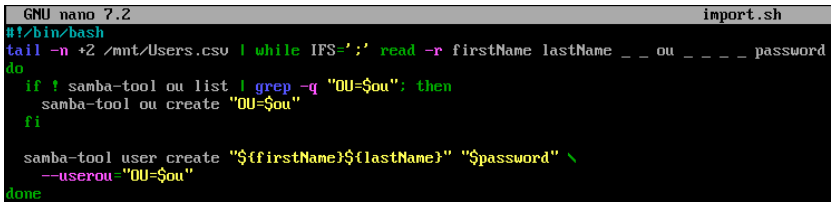
cat /mnt/Users.csv

Создаём файл import.sh и пишем скрипт для быстрого импорта данных:

nano import.sh

```
#!/bin/bash
tail -n +2 /mnt/Users.csv | while IFS=';' read -r firstName lastName _ _ ou _ _ _ _ password
do
  if ! samba-tool ou list | grep -q "OU=$ou"; then
    samba-tool ou create "OU=$ou"
  fi

  samba-tool user create "${firstName}${lastName}" "$password" \
    --userou="OU=$ou"
done
```



Выдаём файлу права на исполнение:

chmod +x import.sh

./import.sh

На HQ-CLI попробовать зайти по данным:

Логин: MalachiAlexander@au-team.irpo

Пароль: P@ssw0rd1

2. Выполните настройку центра сертификации на базе HQ-SRV:

— Необходимо использовать отечественные алгоритмы шифрования

— Сертификаты выдаются на 30дней

— Обеспечьте доверие сертификату для HQ-CLI

— Выдайте сертификаты веб серверам

— Перенастройте ранее настроенный реверсивный прокси nginx на протокол https

— При обращении к веб серверам https://web.au-team.irpo и https://docker.au-team.irpo у браузера клиента не должно возникать предупреждений.

Центра сертификации необходимо развернуть на HQ-SRV:

Устанавливаем пакет openssl-gost-engine, обеспечивающий поддержку алгоритмов ГОСТ:

dnf install openssl-gost-engine

Активируем поддержку ГОСТ в openssl:

openssl-switch-config gost

Применяем модуль политики, определяющий используемые методы шифрования ГОСТ:

update-crypto-policies --set GOST-ONLY:GOST

Проверяем используемую в системе политики шифрования:

update-crypto-policies --show

Генерируем закрытый ключ для удостоверяющего центра:

openssl genpkey -algorithm gost2012_256 -pkeyopt paramset:A -out CA.key

Создаем самоподписанный корневой сертификат (Root CA):

openssl req -new -x509 -md_gost12_256 -days 365 -key CA.key -out CA.crt -subj "/C=RU/ST=Russia/L=Kazan/O=MCK-KTITS/OU=MCK-KTITS CA/CN=MCK-KTITS CA Root"

Генерируем закрытый ключ для веб-серверов:

openssl genpkey -algorithm gost2012_256 -pkeyopt paramset:A -out au-team.irpo.key

Создаём файл расширений:

nano au-team.irpo.ext

И добавляем туда следующее:

authorityKeyIdentifier=keyid,issuer

basicConstraints=CA:FALSE

keyUsage = digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment

subjectAltName = @alt_names

[alt_names]

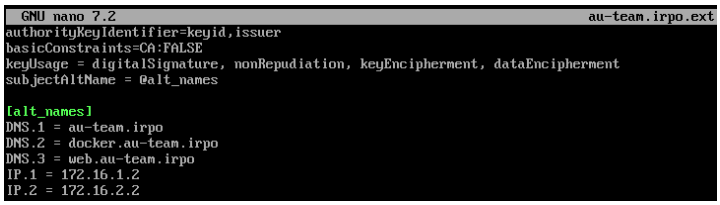
DNS.1 = au-team.irpo

DNS.2 = docker.au-team.irpo

DNS.3 = web.au-team.irpo

IP.1 = 172.16.1.2

IP.2 = 172.16.2.2



Создаем запрос на сертификат (CSR):

openssl req -new -md_gost12_256 -key au-team.irpo.key -out au-team.irpo.csr -subj "/C=RU/L=Kazan/O=AU-TEAM Site GOST/CN=*.au-team.irpo"

Выпускаем сертификат для веб-серверов:

openssl x509 -req -in au-team.irpo.csr -CA CA.crt -CAkey CA.key -CAcreateserial -out au-team.irpo.crt -days 30 -extfile au-team.irpo.ext

Создаем «цепочку сертификатов»:

cat au-team.irpo.crt CA.crt > fullchain.crt

Возвращаем модуль политики, определяющий используемые методы шифрования:

openssl-switch-config default

update-crypto-policies --set DEFAULT

Создаём папку для хранения и дальнейшего переноса сертификатов:

mkdir -p /home/sshuser/certs

Копируем в папку файлы:

cp fullchain.crt /home/sshuser/certs

cp au-team.irpo.key /home/sshuser/certs

cp CA.crt /home/sshuser/certs

Редактируем права на файлы:

chmod 755 -R /home/sshuser/certs

Теперь необходимо произвести настройку nginx на ISP, чтобы сайты работали по https:

Устанавливаем пакет openssl-gost-engine, обеспечивающий поддержку алгоритмов ГОСТ:

dnf install openssl-gost-engine

Активируем поддержку ГОСТ в openssl:

openssl-switch-config gost

Применяем модуль политики, определяющий используемые методы шифрования ГОСТ:

update-crypto-policies --set GOST-ONLY:GOST

Проверяем используемую в системе политики шифрования:

update-crypto-policies –show

Создаём папку для хранения сертификатов:

mkdir -p /etc/ssl/site

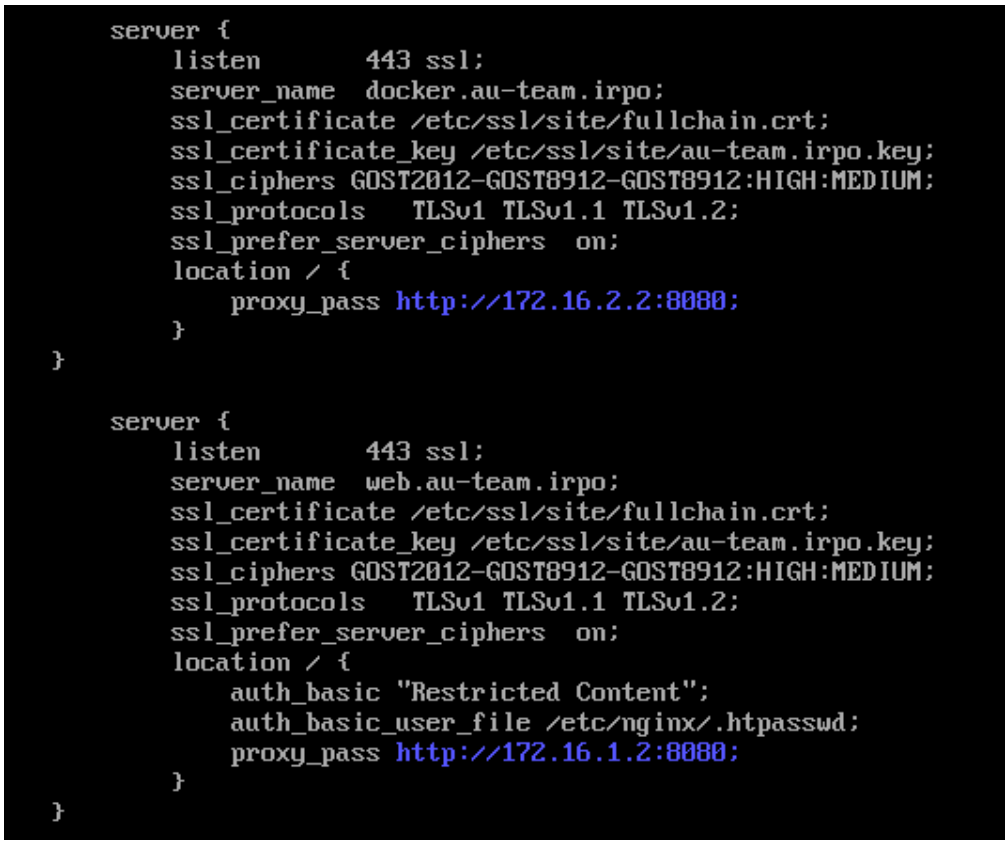
Копируем сертификаты:

scp -P 2026 sshuser@172.16.1.2:/home/sshuser/certs/* /etc/ssl/site

Редактируем файл nginx.conf и добавляем серверы https после http серверов:

```
        server {
            listen      443 ssl;
            server_name  docker.au-team.irpo;
            ssl_certificate /etc/ssl/site/fullchain.crt;
            ssl_certificate_key /etc/ssl/site/au-team.irpo.key;
            ssl_ciphers GOST2012-GOST8912-GOST8912:HIGH:MEDIUM;
            ssl_protocols   TLSv1 TLSv1.1 TLSv1.2;
            ssl_prefer_server_ciphers  on;
            location / {
               proxy_pass http://172.16.2.2:8080;
            }
    }
```

```
        server {
            listen       443 ssl;
            server_name  web.au-team.irpo;
            ssl_certificate /etc/ssl/site/fullchain.crt;
            ssl_certificate_key /etc/ssl/site/au-team.irpo.key;
            ssl_ciphers GOST2012-GOST8912-GOST8912:HIGH:MEDIUM;
            ssl_protocols   TLSv1 TLSv1.1 TLSv1.2;
            ssl_prefer_server_ciphers  on;
            location / {
                auth_basic "Restricted Content";
                auth_basic_user_file /etc/nginx/.htpasswd;
               proxy_pass http://172.16.1.2:8080;
            }
}
```



Проверяем, что конфиг без ошибок:

nginx -t

Если команда не выдала ошибок, то перезапускаем сервис:

systemctl restart nginx

Теперь необходимо произвести настройку на HQ-CLI:

Монтируем Additional.iso:

mount /dev/cdrom /mnt

Копируем файлы КриптоПро для открытия сайтов с сертификатами ГОСТ:

cp -r /mnt/cryptopro/linux-amd64/ .

Копируем корневой сертификат для дальнейшей установки для доверии браузера работы по https:

scp -P 2026 sshuser@192.168.100.2:/home/sshuser/certs/CA.crt /home/username

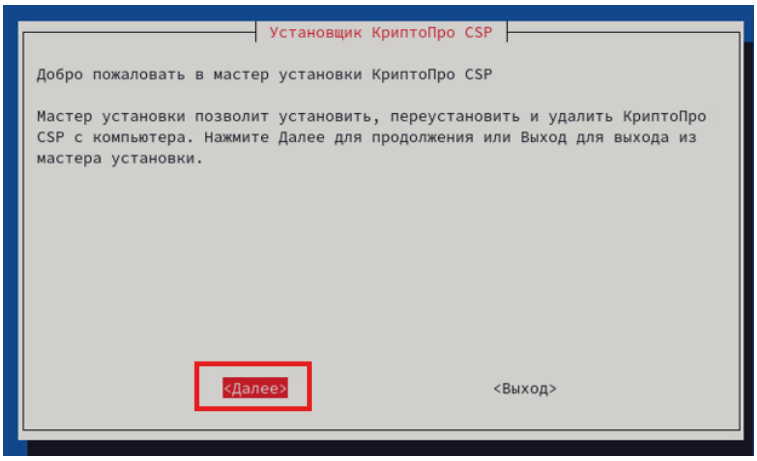
Выдаём права на исполнение исполняемых файлов:

chmod +x linux-amd64/*.sh

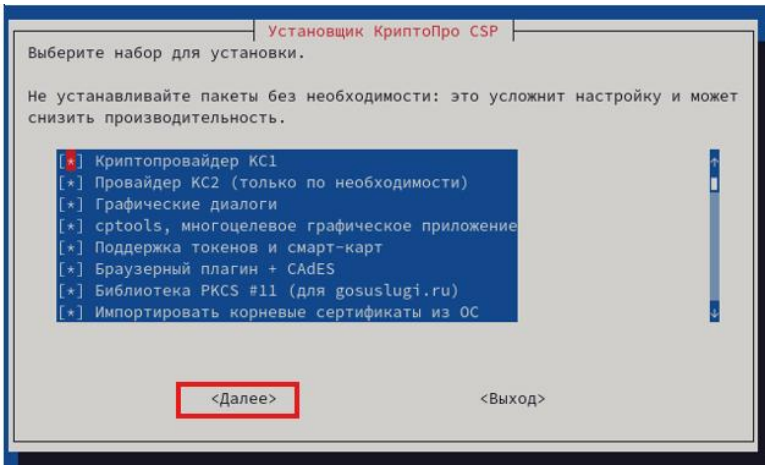
Запускаем установку графической версии КриптоПро:

./linux-amd64/install_gui.sh

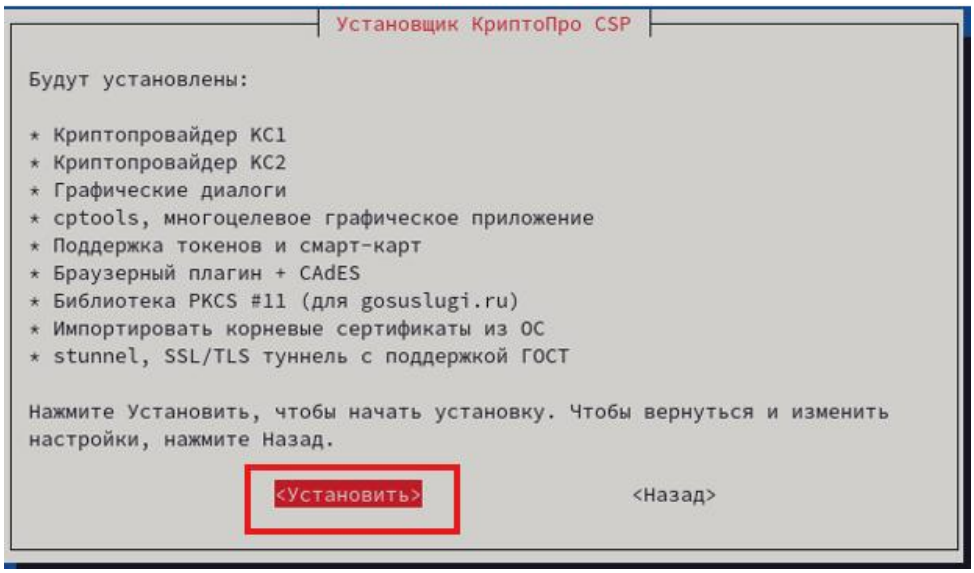
Открывается установщик КриптоПро, нажимаем далее:



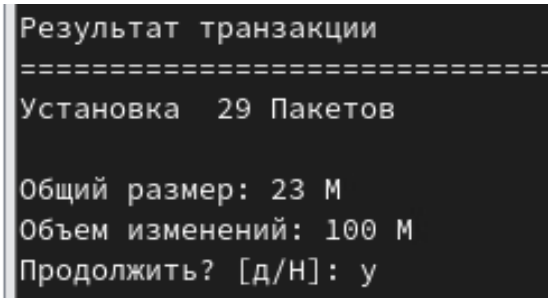
Здесь необходимо выбрать все пакеты с помощью пробел и нажимаем далее:



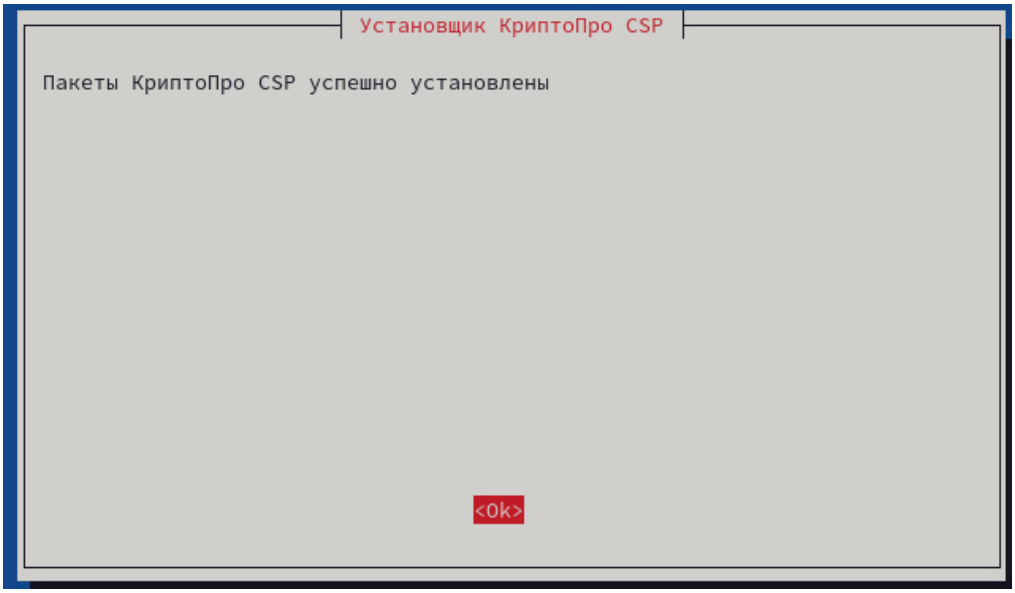
Начинаем установку пакетов КриптоПро:



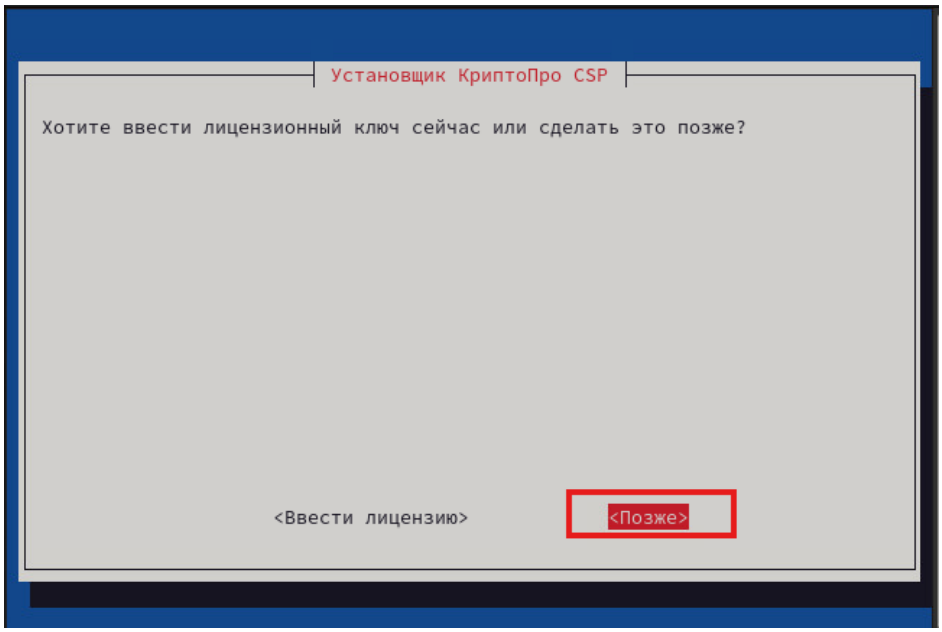
В процесс установки необходимо подтвердить установку пакетов:



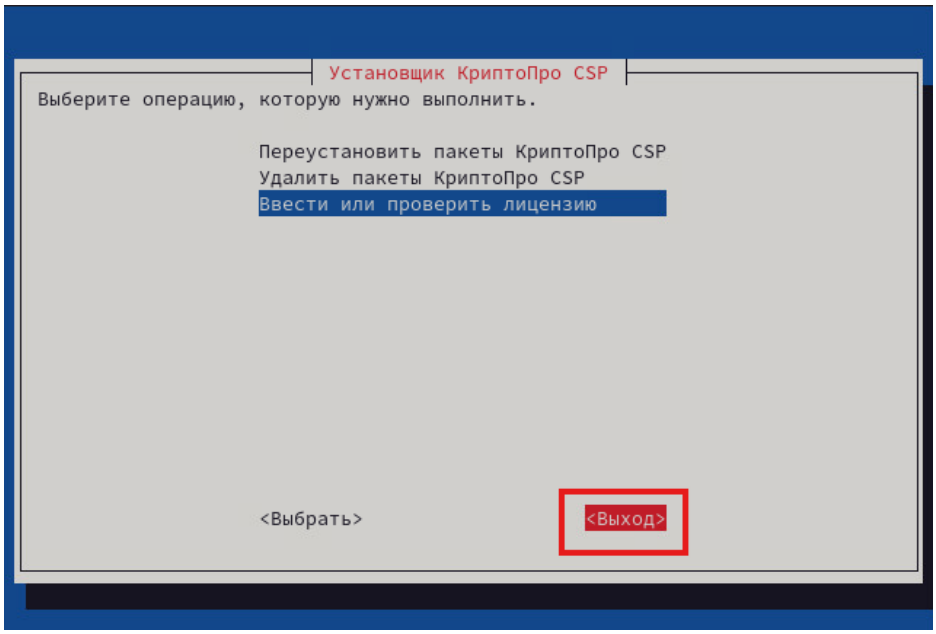
После окончательной установки пакетов, КриптоПро будет успешно установлен.



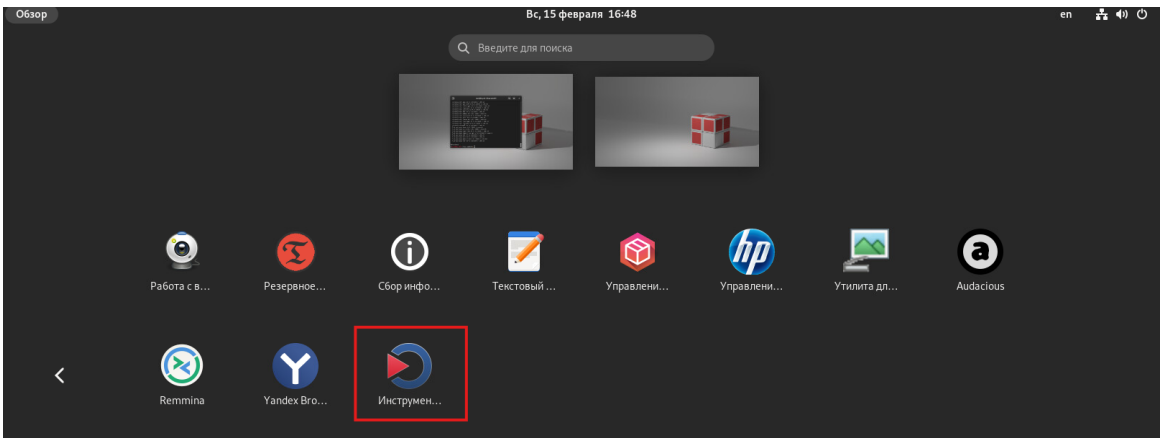
При запросе лицензионного ключа, выбираем позже:



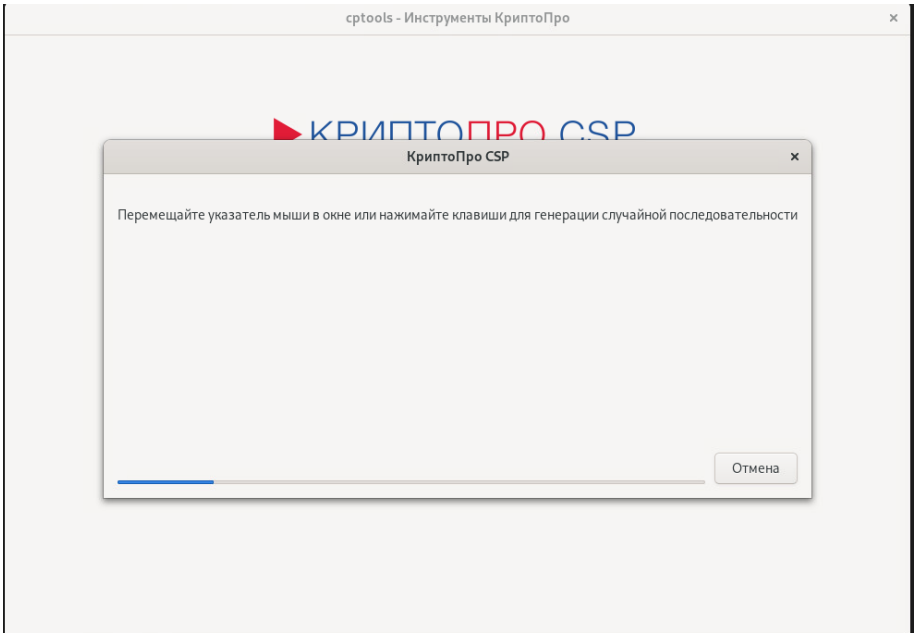
Выходим из установщика:



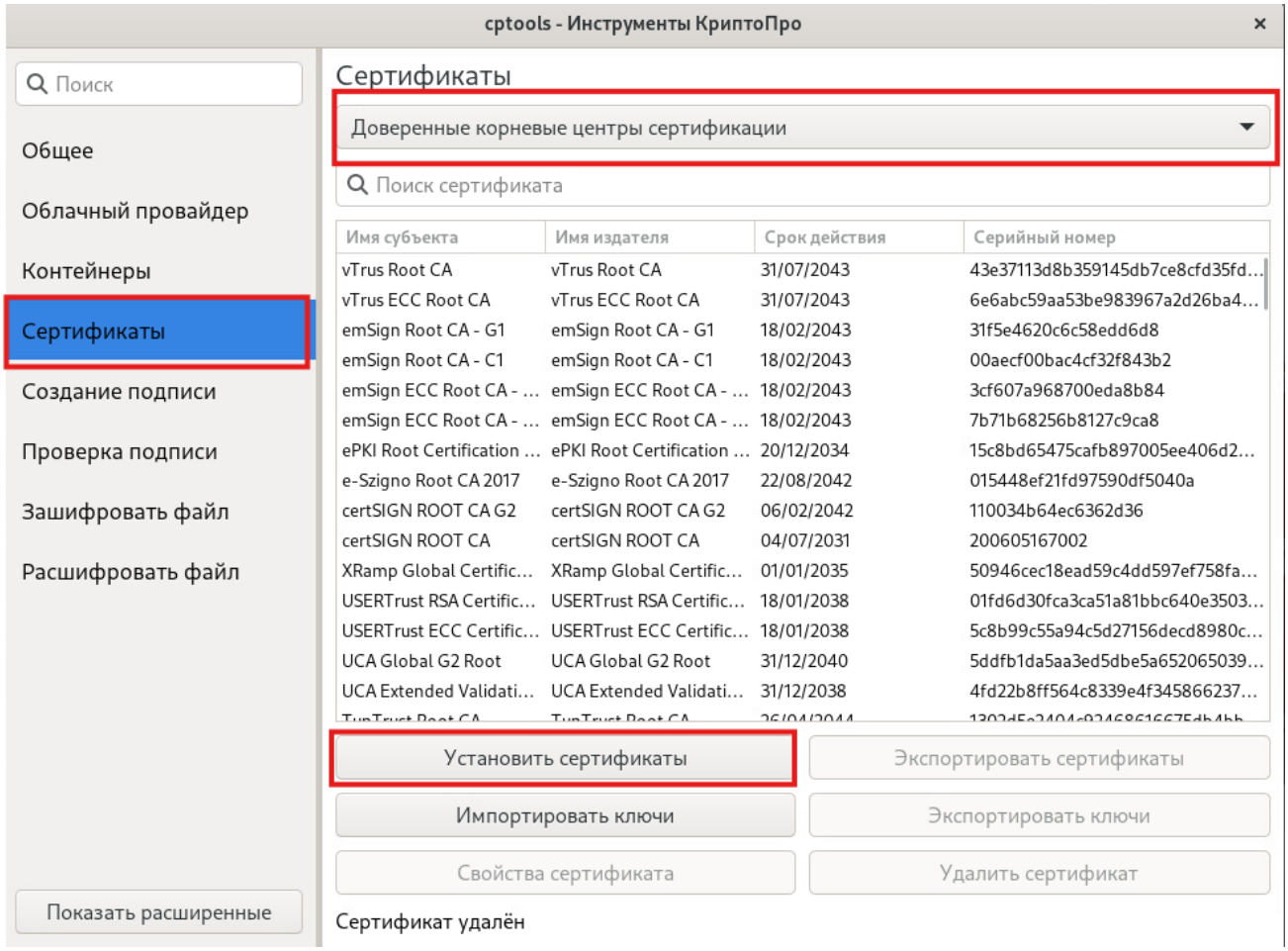
Находим приложение «Инструменты КриптоПро» и запускаем его:



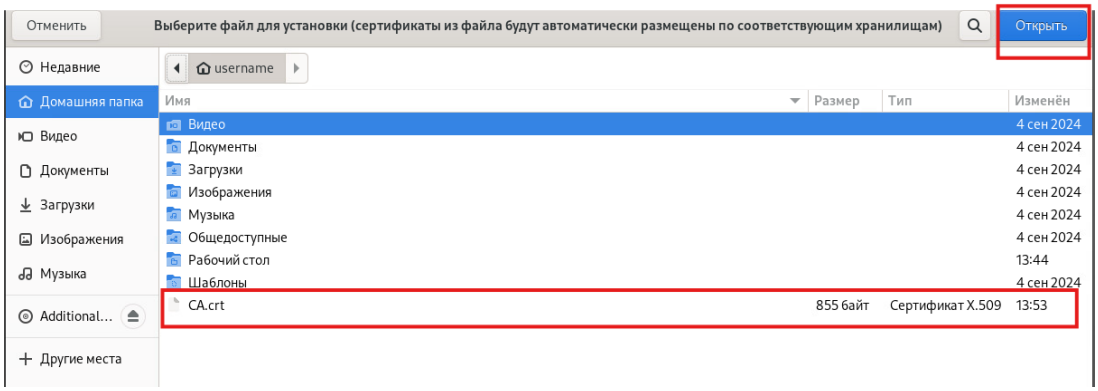
После открытия приложения необходимо перемещать мышь в окне или нажимать клавиши для генерации случайной последовательности:



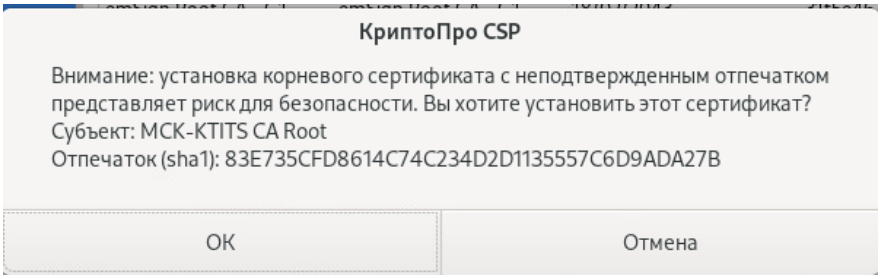
Добавляем корневой сертификат во вкладку «Сертификаты», выбрав хранилище сертификатов «Доверенные корневые центры сертификации» и «Устанавливаем сертификат».



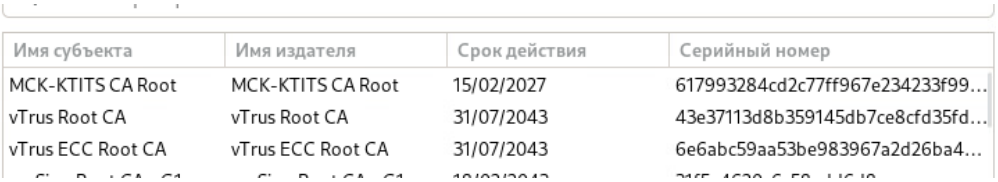
И добавляем сертификат центра, который копировали ранее:



Выйдет уведомление о добавление сертификата, нажимаем «ОК».

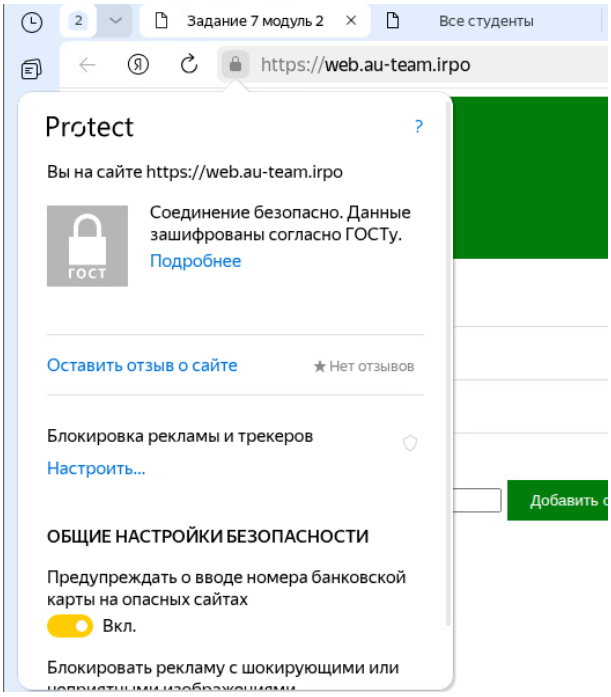


Проверяем наличие сертификата в хранилище:



Имя субъекта	Имя издателя	Срок действия	Серийный номер
MCK-KTITS CA Root	MCK-KTITS CA Root	15/02/2027	617993284cd2c77ff967e234233f99...
vTrus Root CA	vTrus Root CA	31/07/2043	43e37113d8b359145db7ce8cfd35fd...
vTrus ECC Root CA	vTrus ECC Root CA	31/07/2043	6e6abc59aa53be983967a2d26ba4...

Запускаем браузер Яндекс.Браузер и проверяем сайты https://web.au-team.irpo и https://web.au-team.irpo, что подключения защищены:



3. Перенастройте ip-туннель с базового до уровня туннеля, обеспечивающего шифрование трафика

— Настройте защищенный туннель между HQ-RTR и BR-RTR

— Внесите необходимые изменения в конфигурацию динамической маршрутизации, протокол динамической маршрутизации должен возобновить работу после перенастройки туннеля

— Выбранное программное обеспечение, обоснование его выбора и его основные параметры, изменения в конфигурации динамической маршрутизации отметьте в отчёте.

Устанавливаем strongswan на HQ-RTR и BR-RTR:

dnf install strongswan -y

Добавляем strongswan в автозагрузку:

systemctl enable --now strongswan

Создаём конфигурационный файл на HQ-RTR:

nano /etc/strongswan/swanctl/conf.d/swanctl.conf

Добавляем конфигурацию IPSec:

```
connections {
  my-tunnel {
    local_addrs  = 172.16.1.2
    remote_addrs = 172.16.2.2
    local {
      auth = psk
    }
    remote {
      auth = psk
    }
    children {
      net {
        mode = transport
        esp_proposals = aes256-sha256
      }
    }
  }
}
secrets {
  ike-1 {
    secret = "P@ssw0rd"
  }
}
```

Перезапускаем службу strongswan:

systemctl restart strongswan

Создаём конфигурационный файл на BR-RTR:

nano /etc/strongswan/swanctl/conf.d/swanctl.conf

Добавляем конфигурацию IPSec:

```
connections {
  my-tunnel {
    local_addrs  = 172.16.2.2
    remote_addrs = 172.16.1.2
    local {
      auth = psk
    }
    remote {
      auth = psk
    }
    children {
      net {
        mode = transport
        esp_proposals = aes256-sha256
      }
    }
  }
}
secrets {
  ike-1 {
    secret = "P@ssw0rd"
  }
}
```

Перезапускаем службу strongswan:

systemctl restart strongswan

Принудительно инициируем соединение (BR-RTR или HQ-RTR):

swanctl --initiate --child net

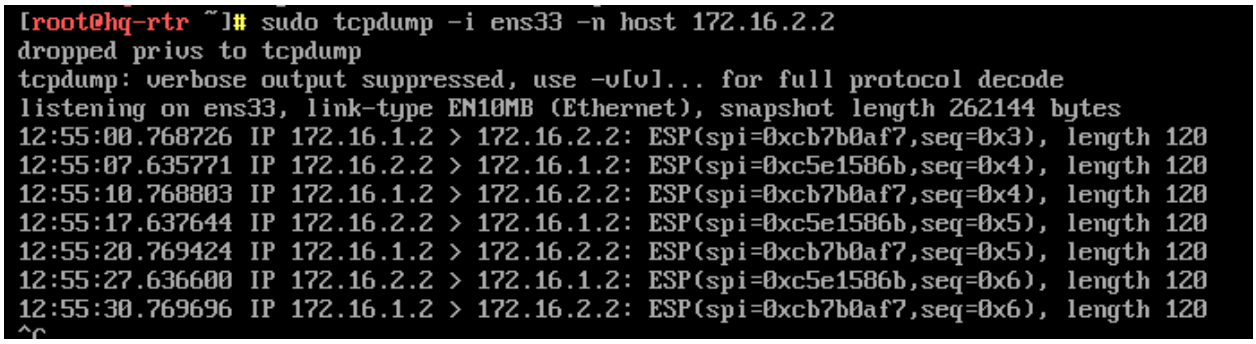
Проверяем работу:

swanctl --list-conns

(Вывод должен быть приблизительно таким)

Также дополнительно проверяем, например, с HQ-RTR, что пакеты имеют заголовок ESP, что подтверждает работу IPSec:

tcpdump -i ens33 -n host 172.16.2.2



4. Настройте межсетевой экран на маршрутизаторах HQ-RTR и BR-RTR на сеть в сторону ISP

— Обеспечьте работу протоколов http, https, dns, ntp, icmp или дополнительных нужных протоколов

— Запретите остальные подключения из сети Интернет во внутреннюю сеть.

На HQ-RTR:

firewall-cmd --permanent --zone=external --change-interface=ens33

firewall-cmd --permanent --zone=internal --change-interface=ens34

firewall-cmd --permanent --zone=internal --change-interface=ens34.100

firewall-cmd --permanent --zone=internal --change-interface=ens34.200

firewall-cmd --permanent --zone=internal --add-interface=tun0

firewall-cmd --permanent --new-policy int-to-ext

firewall-cmd --permanent --policy int-to-ext --add-ingress-zone=internal

firewall-cmd --permanent --policy int-to-ext --add-egress-zone=external

firewall-cmd --permanent --policy int-to-ext --set-target=ACCEPT

firewall-cmd --permanent --zone=external --add-service=http

firewall-cmd --permanent --zone=external --add-service=https

firewall-cmd --permanent --zone=external --add-service=dns

firewall-cmd --permanent --zone=external --add-service=ntp

firewall-cmd --permanent --zone=external --add-port=2026/tcp

firewall-cmd --permanent --zone=external --add-port=8080/tcp

firewall-cmd --permanent --zone=external --add-protocol=gre

```
firewall-cmd --permanent --zone=external --add-port=500/udp
firewall-cmd --permanent --zone=external --add-port=4500/udp
firewall-cmd --permanent --zone=external --add-protocol=esp
firewall-cmd --permanent --zone=external --add-protocol=ah
firewall-cmd --permanent --zone=internal --add-protocol=ospf
firewall-cmd --permanent --zone=external --add-forward-port=port=2026:proto=tcp:toport=2026:toaddr=192.168.100.2
firewall-cmd --permanent --zone=external --add-forward-port=port=8080:proto=tcp:toport=80:toaddr=192.168.100.2
firewall-cmd --permanent --direct --add-passthrough ipv4 -t mangle -A FORWARD -p tcp --tcp-flags SYN,RST SYN -j TCPMSS --clamp-mss-to-pmtu
firewall-cmd --reload
```

Ha BR-RTR:

```
firewall-cmd --permanent --zone=external --change-interface=ens33
firewall-cmd --permanent --zone=internal --change-interface=ens34
firewall-cmd --permanent --zone=internal --add-interface=tun0
firewall-cmd --permanent --new-policy int-to-ext
firewall-cmd --permanent --policy int-to-ext --add-ingress-zone=internal
firewall-cmd --permanent --policy int-to-ext --add-egress-zone=external
firewall-cmd --permanent --policy int-to-ext --set-target=ACCEPT
firewall-cmd --permanent --zone=external --add-service=http
firewall-cmd --permanent --zone=external --add-service=https
firewall-cmd --permanent --zone=external --add-service=dns
firewall-cmd --permanent --zone=external --add-service=ntp
firewall-cmd --permanent --zone=external --add-port=2026/tcp
firewall-cmd --permanent --zone=external --add-port=8080/tcp
firewall-cmd --permanent --zone=external --add-protocol=gre
firewall-cmd --permanent --zone=external --add-port=500/udp
firewall-cmd --permanent --zone=external --add-port=4500/udp
firewall-cmd --permanent --zone=external --add-protocol=esp
firewall-cmd --permanent --zone=external --add-protocol=ah
```

firewall-cmd --permanent --zone=internal --add-protocol=ospf

firewall-cmd --permanent --zone=external --add-forward-port=port=2026:proto=tcp:toport=2026:toaddr=172.30.100.2

firewall-cmd --permanent --zone=external --add-forward-port=port=8080:proto=tcp:toport=8080:toaddr=172.30.100.2

firewall-cmd --permanent --direct --add-passthrough ipv4 -t mangle -A FORWARD -p tcp --tcp-flags SYN,RST SYN -j TCPMSS --clamp-mss-to-pmtu

firewall-cmd --reload

5. Настройте принт-сервер cups на сервере HQ-SRV:

— Опубликуйте виртуальный pdf-принтер

— На клиенте HQ-CLI подключите виртуальный принтер как принтер по умолчанию.

Разворачиваем принт-сервер CUPS на HQ-SRV:

dnf install cups cups-pdf -y –nogpgcheck

Добавляем службу в автозагрузку:

systemctl enable –-now cups

Вносим изменения в файл:

nano /etc/cups/cupsd.conf

В строке Listen localhost:631 меняем localhost на *:

```
# Only listen for connections from the local machine.
Listen *:631
Listen /run/cups/cups.sock
```

В строках с доступом на сервер и с доступом на страницу админа добавляем строки Allow all:

```
# Restrict access to the server...
<Location />
  Order allow,deny
  Allow all
</Location>

# Restrict access to the admin pages...
<Location /admin>
  AuthType Default
  Require user @SYSTEM
  Order allow,deny
  Allow all
</Location>
```

Перезапускаем службу cups:

systemctl restart cups

Создаём принтер на HQ-SRV:

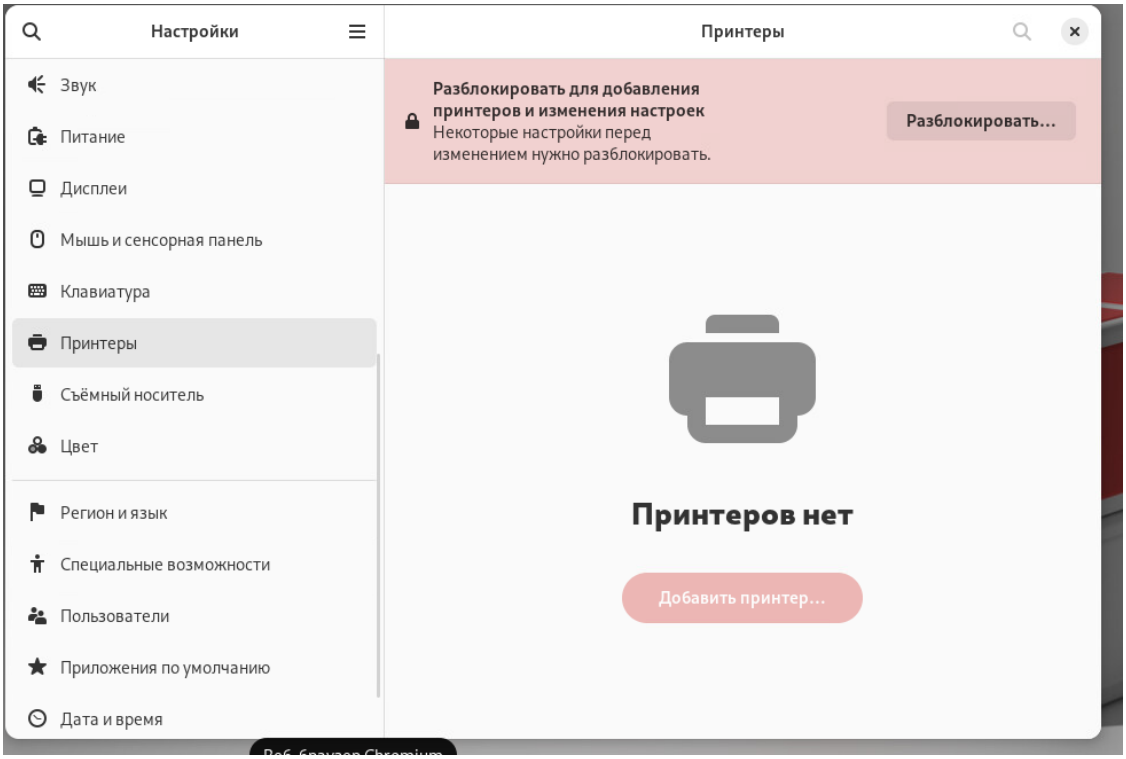
lpadmin -p Virtual_PDF -E -v cups-pdf:/ -m raw

Проверяем создание принтера на HQ-SRV:

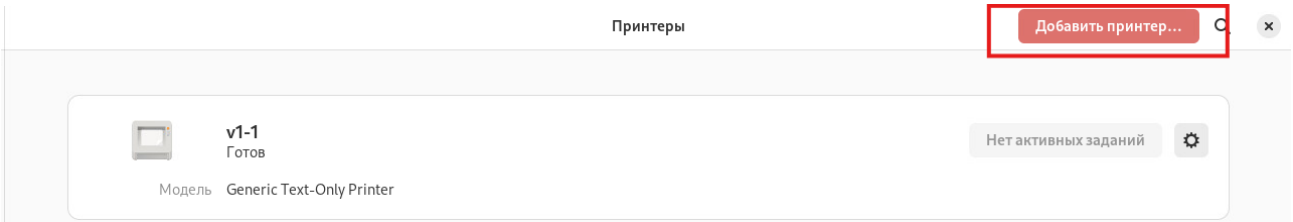
lpstat -p

```
[root@hq-srv ~]# lpstat -p
принтер Cups-PDF свободен. Включен с момента Вс 15 фев 2026 15:02:17
[root@hq-srv ~]#
```

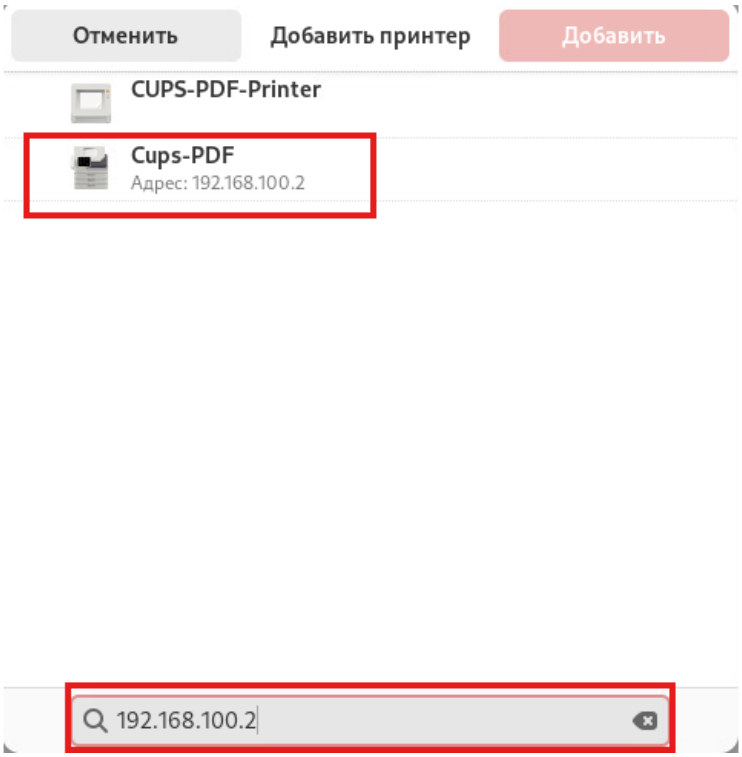
Переходим в настройки на HQ-CLI и добавляем принтер. Для этого сначала необходимо разблокировать доступ к добавлению и изменению настроек:



Добавляем принтер:

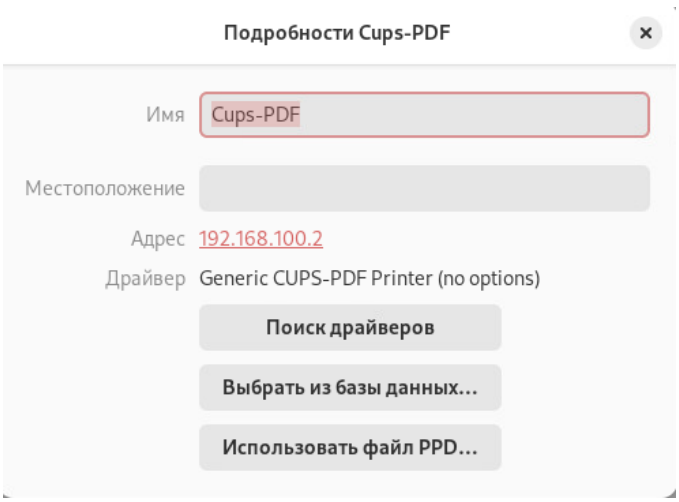


Вводим IP-адрес сервера HQ-SRV и выбираем принтер с HQ-SRV:



После добавления принтера, настраиваем принтер для использования по умолчанию, необходимо, что появилась галочка у пункта «Использовать по умолчанию».

После этого, дополнительно проверяем, выбрав пункт «Сведения о принтере», что принтер действительно с HQ-SRV. Должен быть IP HQ-SRV.



6. Реализуйте логирование при помощи rsyslog на устройствах HQ-RTR, BR-RTR, BR-SRV:

— Сервер сбора логов расположен на HQ-SRV, убедитесь, что сервер не является клиентом самому себе

— Приоритет сообщений должен быть не ниже warning

— Все журналы должны находиться в директории /opt. Для каждого устройства должна выделяться своя поддиректория, которая совпадает с именем машины

— Реализуйте ротацию собранных логов на сервере HQ-SRV:

— Ротируются все логи, находящиеся в директории и поддиректориях /opt

— Ротация производится один раз в неделю

— Логи необходимо сжимать

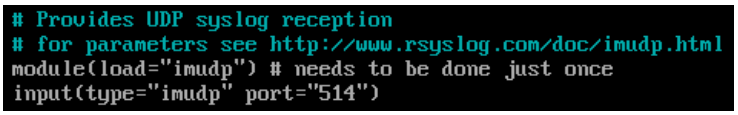
— Минимальный размер логов для ротации – 10МБ.

Развернём сервер сбора логов на HQ-SRV:

Необходимо включить прием логов по UDP (или TCP) и настроить шаблоны для записи в /opt. Открываем и редактируем файл /etc/rsyslog.conf.

nano /etc/rsyslog.conf

Необходимо расскоментировать строки связанные с udp (module и input):



В конце добавляем шаблон для записи логов с удаленных серверов в /opt и исключения hq-srv из записи в папку /opt:

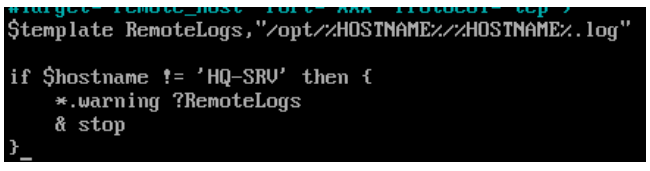
$template RemoteLogs,"/opt/%HOSTNAME%/%HOSTNAME%.log"

if $hostname != 'hq-srv' then {

*.warning ?RemoteLogs

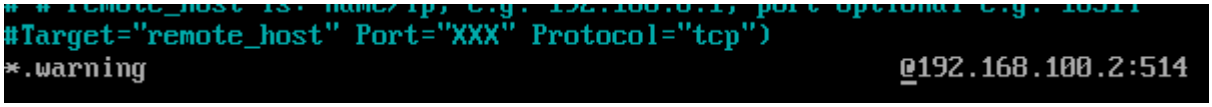
& stop

}



На клиентах HQ-RTR, BR-RTR, BR-SRV в файле /etc/rsyslog.conf добавляем запись, для отправки логов на сервер логов (можно также в конце):

nano /etc/rsyslog.conf



На HQ-RTR, BR-RTR, BR-SRV, HQ-SRV перезапускаем службу rsyslog:

systemctl restart rsyslog

На HQ-SRV настраиваем ротацию логов. Для этого создаём файл /etc/logrotate.d/remote_logs.

nano /etc/logrotate.d/remote_logs

Добавляем в него конфигурацию:

```
/opt/*/*.log {
    weekly
    rotate 4
    compress
    minsize 10M
    missingok
    notifempty
    sharedscripts
    postrotate
        /usr/bin/systemctl kill -s HUP rsyslog.service >/dev/null 2>&1 || true
    endscript
}
```

На HQ-RTR, BR-RTR, BR-SRV, HQ-SRV перезапускаем службу logrotate:

systemctl restart logrotate

7. На сервере HQ-SRV реализуйте мониторинг устройств с помощью открытого программного обеспечения

— Обеспечьте доступность по URL - http://mon.au-team.irpo для сетей офиса HQ, внесите изменения в инфраструктуру разрешения доменных имён

— Мониторить нужно устройства HQ-SRV и BR-SRV

— В мониторинге должны визуально отображаться нагрузка на ЦП, объем занятой ОП и основного накопителя

— Логин и пароль для службы мониторинга admin P@ssw0rd

— Организуйте доступ к мониторингу для HQ-CLI, без внешнего доступа

— Выбор программного обеспечения, основание выбора и основные параметры с указанием порта, на котором работает мониторинг, отметьте в отчёте

Разворачиваем на сервере HQ-SRV Grafana, Prometheus и node_exporter:

dnf install -y grafana prometheus prometheus-node_exporter –nogpgcheck

После установки редактируем файл prometheus.yml, добавляем адреса клиентов node_exporter для сбора метрик:

nano /etc/prometheus/prometheus.yml

Редактируем там строчку, добавив адреса 192.168.100.2:9100 и 172.30.100.2:9100:

static_configs:

- targets: ['localhost:9090','192.168.100.2:9100','172.30.100.2:9100']

```
static_configs:
  - targets: ["localhost:9090","192.168.100.2:9100","172.30.100.2:9100"]
    # The label name is added as a label `label_name=<label_value>` to any timeseries scraped from this config.
    labels:
      app: "prometheus"
```

Добавляем в автозагрузку сервисы:

systemctl enable --now grafana-server

systemctl enable --now prometheus

systemctl enable --now node_exporter

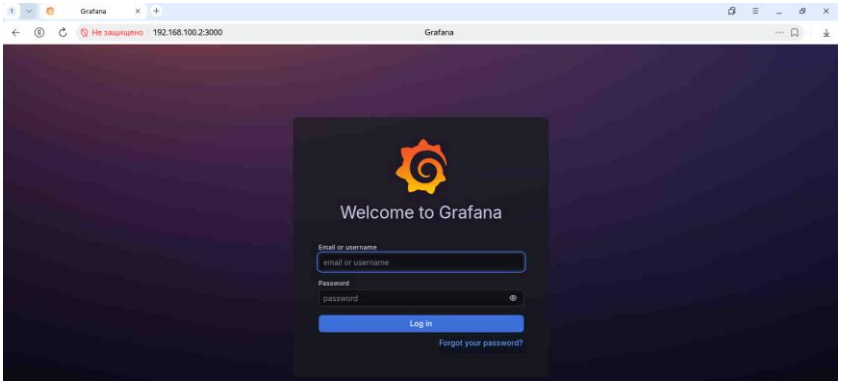
Разворачиваем на BR-SRV node_exporter:

dnf install -y prometheus-node_exporter

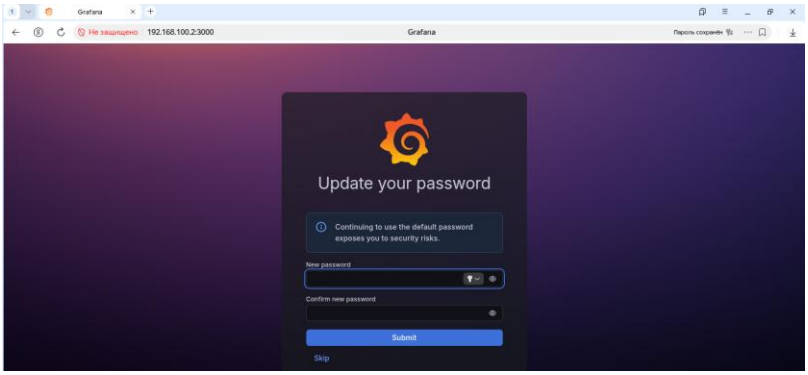
Добавляем в автозагрузку сервис node_exporter:

systemctl enable --now node_exporter

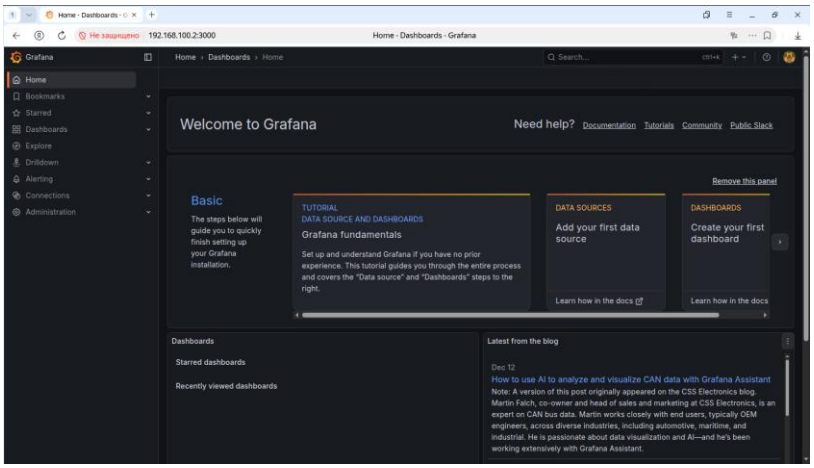
Переходим на HQ-CLI для настройки Grafana. Открываем браузер и переходим по адресу http://192.168.100.2:3000. Откроется веб-интерфейс Grafana.



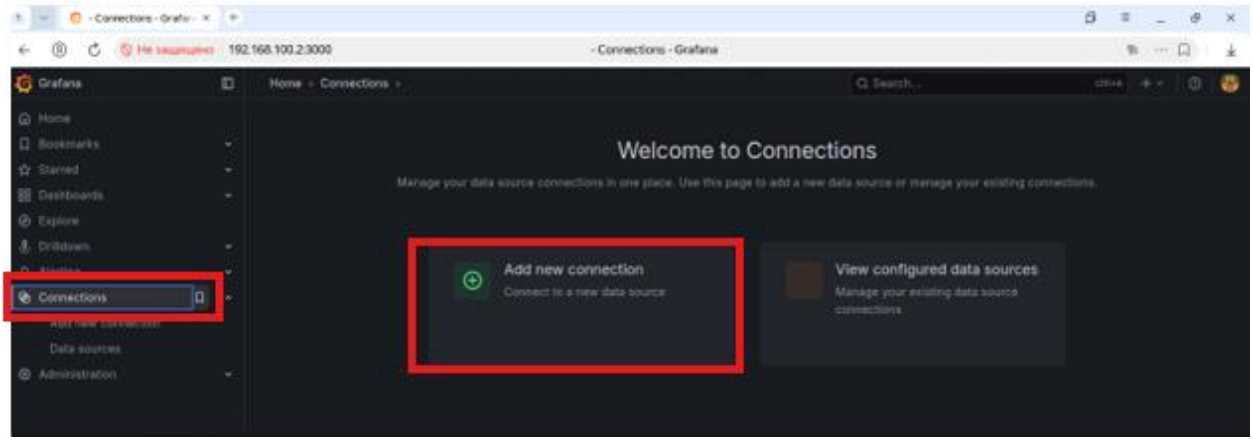
Авторизуемся с помощью логина и пароля admin:admin. После этого Grafana запросит новый пароль для учетной записи admin. Задаём новый пароль P@ssw0rd.



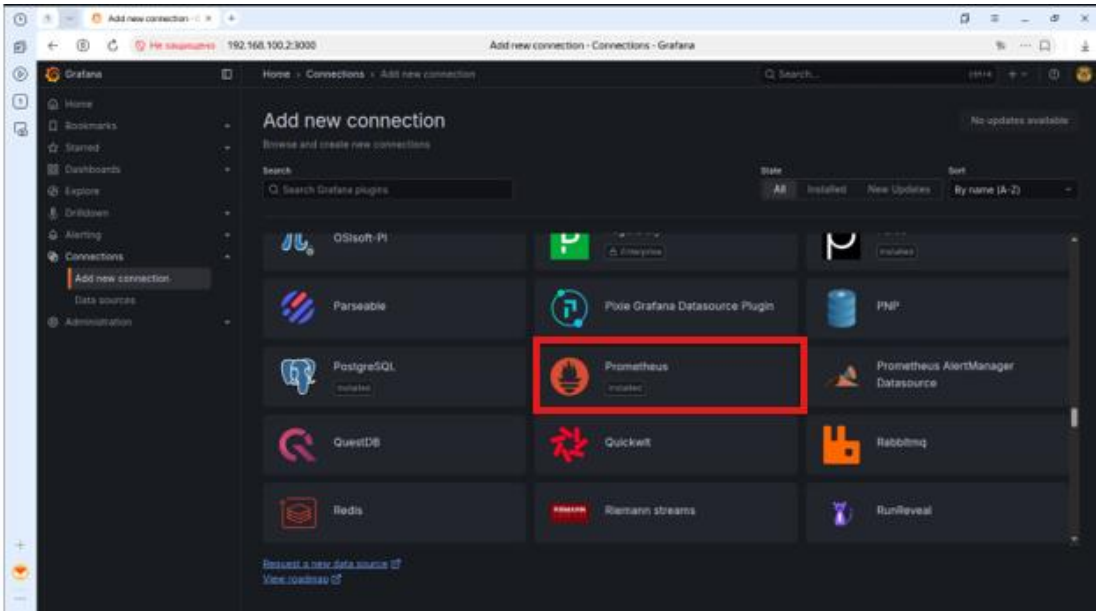
После чего, открывается веб-интерфейс Grafana.



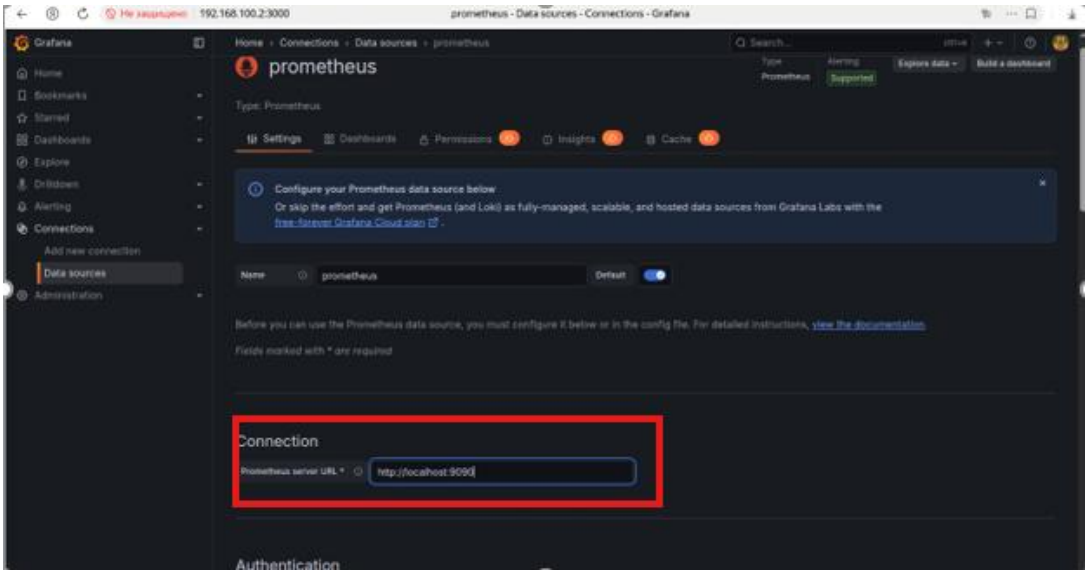
Добавляем подключение к Prometheus. Для этого переходим во вкладку Connections и добавляем новое подключение через «Add new connection»



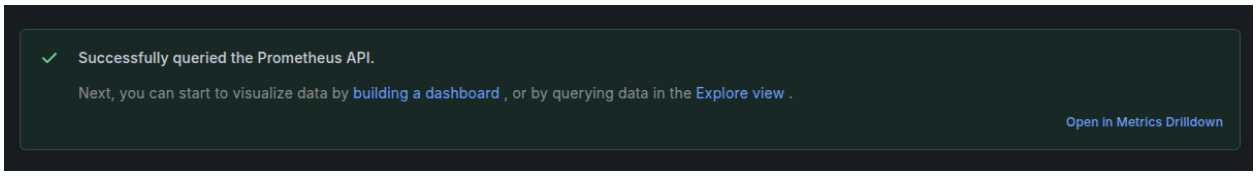
Находим в DataSources Prometheus



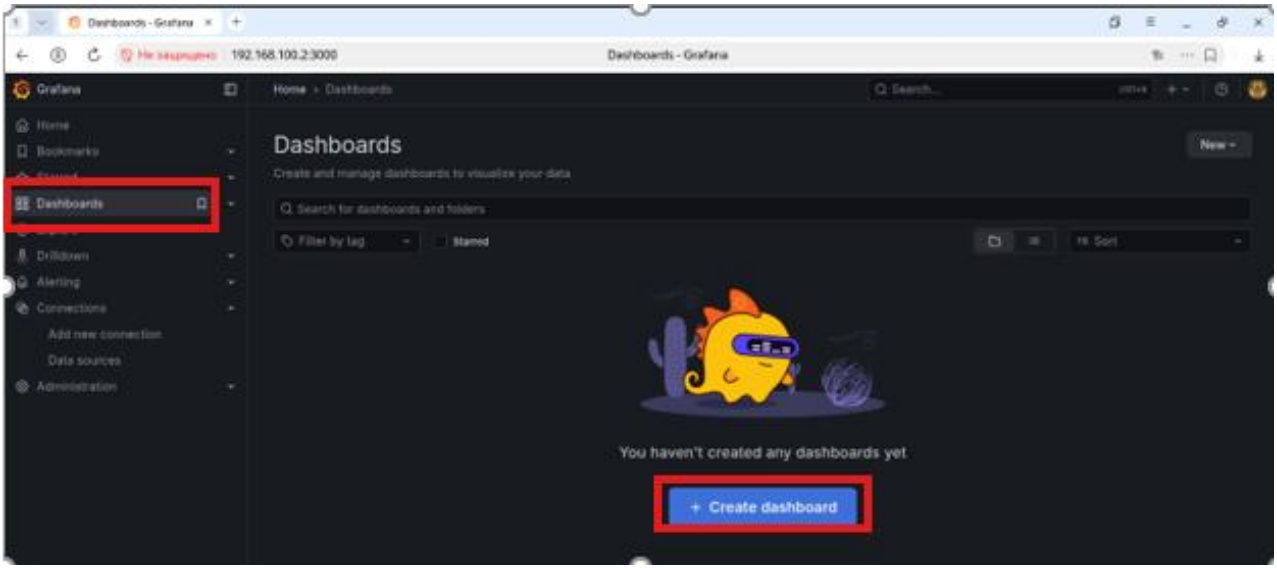
В connection пишем адрес Prometheus – http://localhost:9090



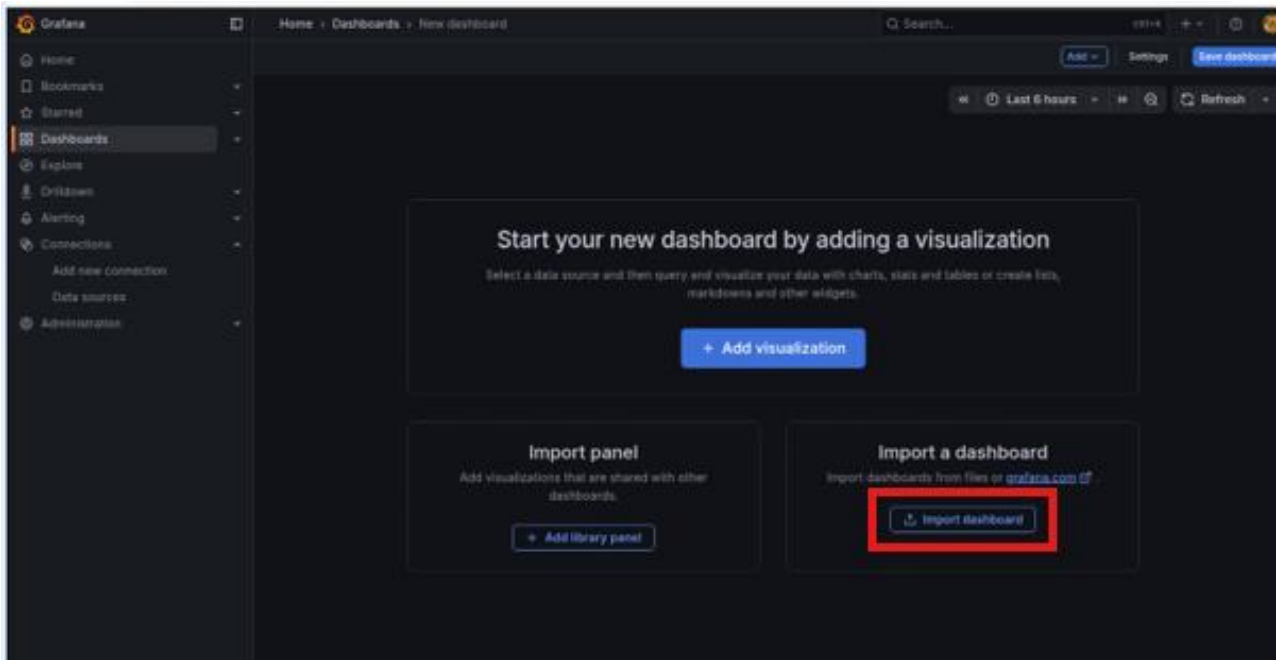
После чего Datasource Prometheus должно успешно добавиться.



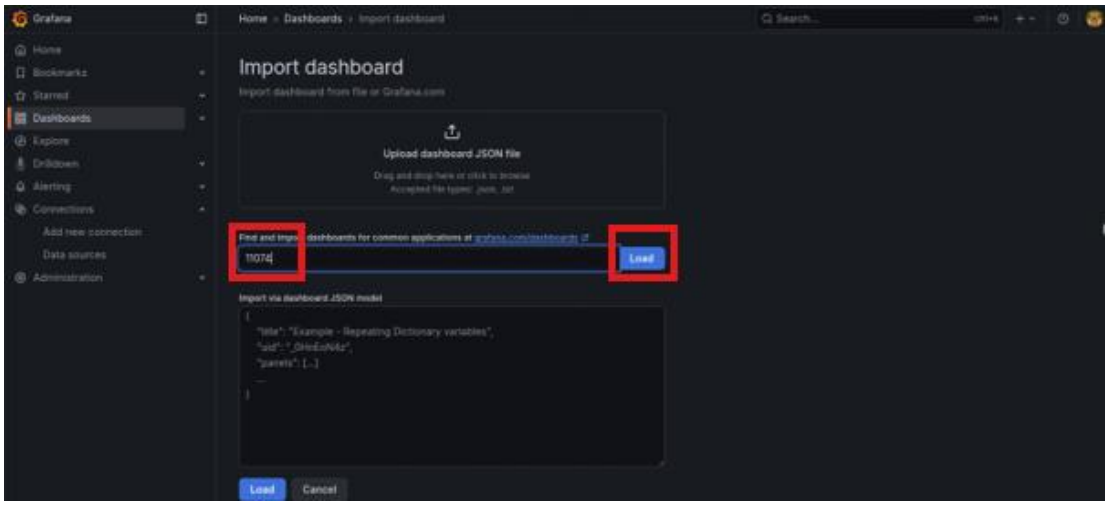
Добавляем Dashboard, выбрав вкладку Dashboards и создаём дашборд «Create Dashboard»



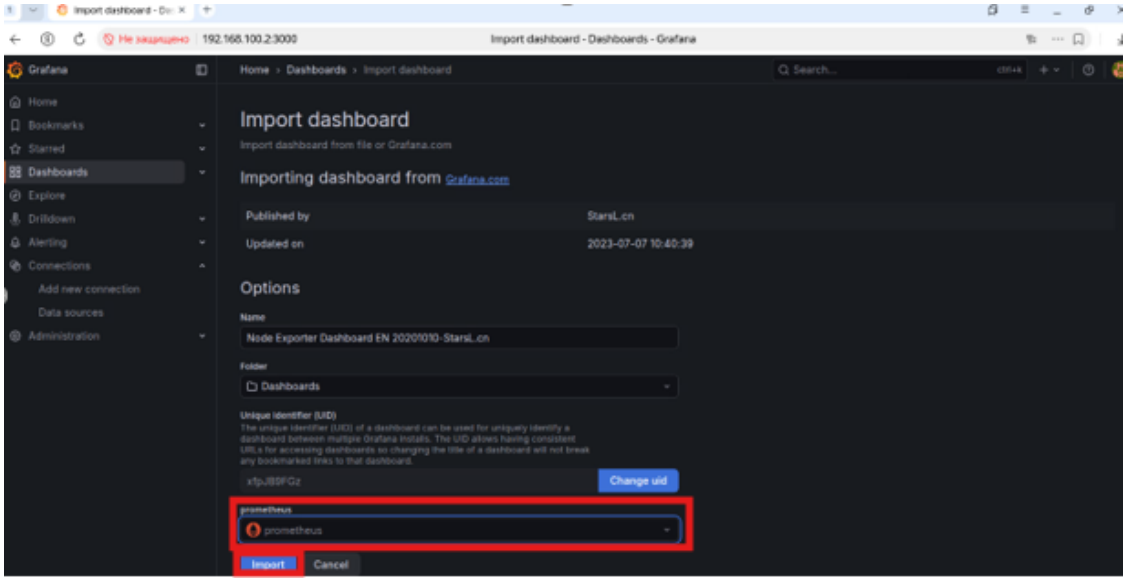
Импортируем Dashboard, выбрав «Import Dashboard»



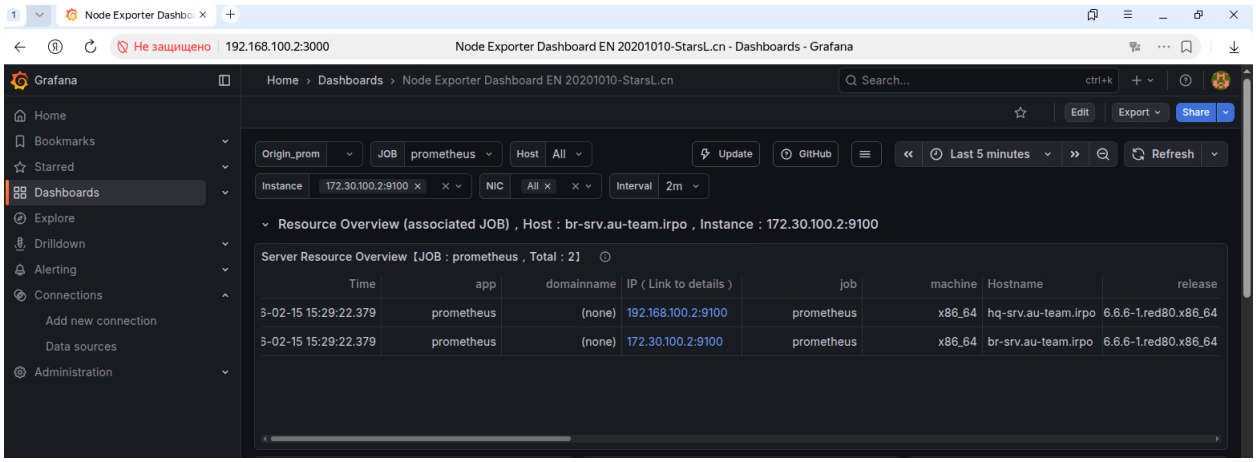
Добавляем номер dashboard 11074 и загружаем его «Load».



После этого выбираем Prometheus, который ранее был подключен к Grafana и импортирует дашборд.

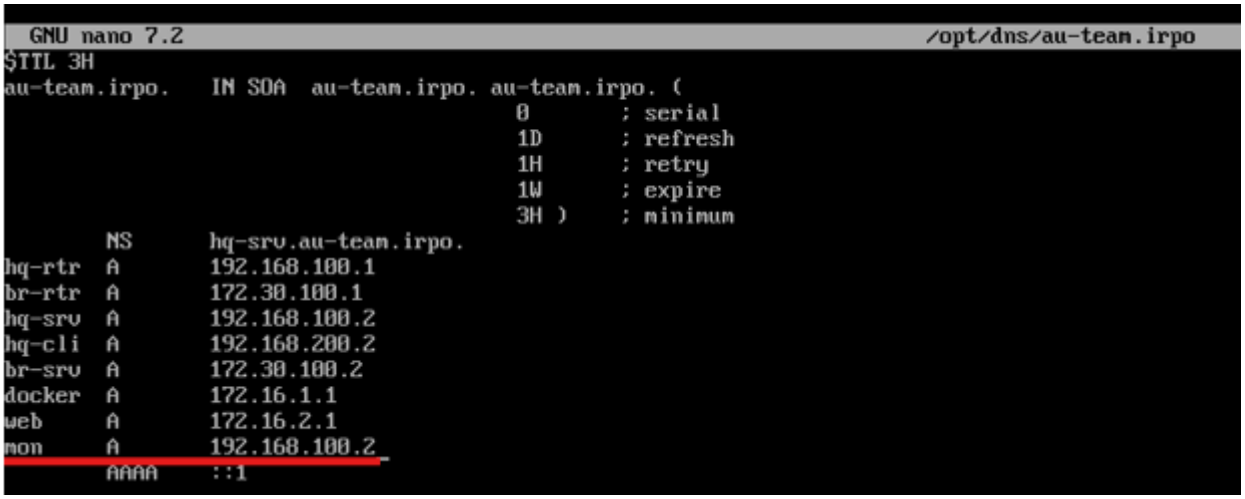


В результате получаем дашборд с мониторингом серверов HQ-SRV и BR-SRV.



После этого необходимо настроить доступ к мониторингу по доменному имени http://mon.au-team.irpo. Для этого переходим на HQ-SRV и открываем файл /opt/dns/au-team.irpo и добавляем A запись для mon.au-team.irpo, указав адрес HQ-SRV.

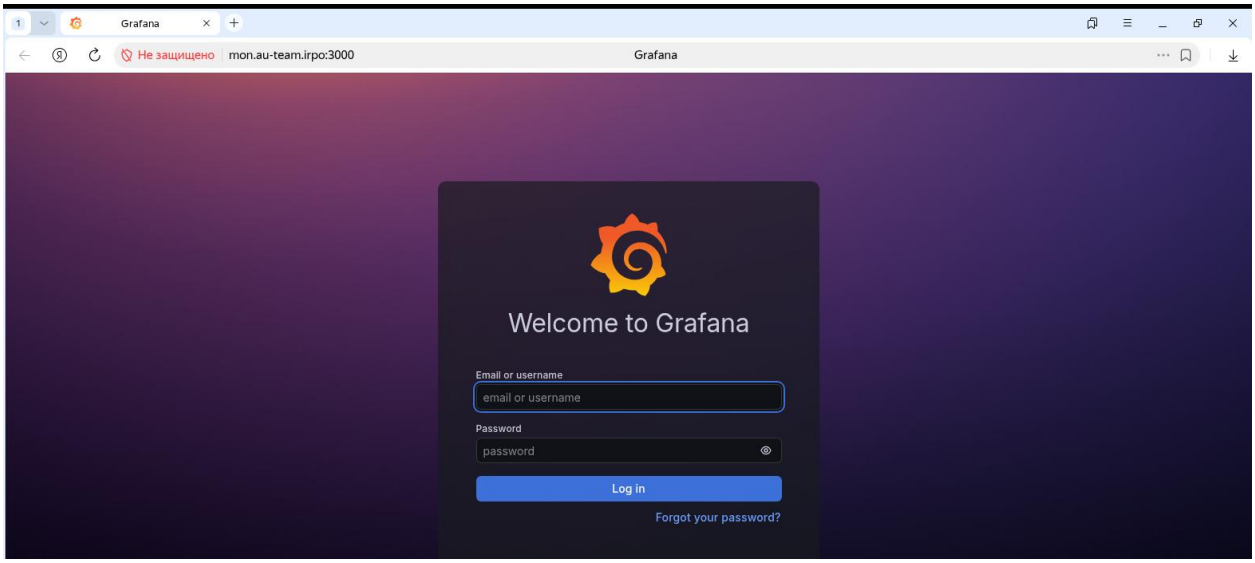
nano /opt/dns/au-team.irpo



После этого перезапускаем службу named.

systemctl restart named

Теперь открываем мониторинг на HQ-CLI через адрес http://mon.au-team.irpo:3000.



8. Реализуйте механизм инвентаризации машин HQ-SRV и HQ-CLI через Ansible на BR-SRV:

— Плейбук должен собирать информацию о рабочих местах:

Имя компьютера

IP-адрес компьютера

— Плейбук, должен быть размещен в директории /etc/ansible, отчёты в поддиректории PC-INFO, в формате .yml. Файлы должны называется именем компьютера, который был инвентаризирован

— Файл плейбука располагается в образе Additional.iso в директории playbook

Монтируем образ Additional.iso:

mount /dev/cdrom /mnt

Копируем playbook из образа:

cp /mnt/playbook/get_hostname_address.yml /etc/ansible

Добавяем группу хостов inventory с HQ-SRV и HQ-CLI в файл инвентаризации demo.ini:

nano /etc/ansible/demo.ini

```
[inventory]
hq-cli ansible_host=192.168.200.2 ansible_user=username
hq-srv ansible_host=192.168.100.2 ansible_port=2026 ansible_user=sshuser
```

Редактирует плейбук и приводим к виду:

```
---
- name: Inventory of HQ-SRV and HQ-CLI
  hosts: inventory
  gather_facts: yes
  tasks:
    - name: получение данных с хоста
      copy:
        dest: /etc/ansible/PC-INFO/{{ ansible_hostname }}.yml
        content: |
          Hostname: {{ ansible_hostname }}
          IP_Address: {{ ansible_default_ipv4.address }}
      delegate_to: localhost
```

Создаём папку для хранения файлов:

mkdir -p /etc/ansible/PC-INFO

Переходим в директорию ansible:

cd /etc/ansible

Запускаем исполнение плейбука:

ansible-playbook get_hostname_address.yml -i demo.ini

После этого проверяем, что файлы были созданы:

ls /etc/ansible/PC-INFO

Проверяем содержимое:

cat /etc/ansible/PC-INFO/hq-cli.yml

cat /etc/ansible/PC-INFO/hq-srv.yml

9. На HQ-SRV настройте программное обеспечение fail2ban для защиты ssh

— Укажите порт ssh

— При 3 неуспешных авторизациях адрес атакующего попадает в бан

— Бан производится на 1минуту

Выполняем установку fail2ban на HQ-SRV:

dnf install fail2ban -y --nogpgcheck

Создаём файл с конфигурацией:

nano /etc/fail2ban/jail.local

Пишем следующие конфигурацию:

[sshd]

enabled = true

port = 2026

filter = sshd

maxretry = 3

bantime = 60

findtime = 120

Перезапускаем службу fail2ban:

systemctl restart fail2ban

Добавляем службу в автозагрузку:

systemctl enable --now fail2ban

Проверяем статус настроек:

fail2ban-client status sshd