**Инструкции для участника**

Прочитайте задание полностью.

Обратите внимание на то, что задание написано не в хронологическом порядке; последовательность действий для выполнения задания остаётся на ваше усмотрение.

Выполнение задания на 100% за отведённое время теоретически возможно, но от вас не требуется выполнять задание полностью.

Распределите время работы таким образом, чтобы заработать максимальное количество баллов. Во время выполнения задания руководствуйтесь следующими полезными советами:

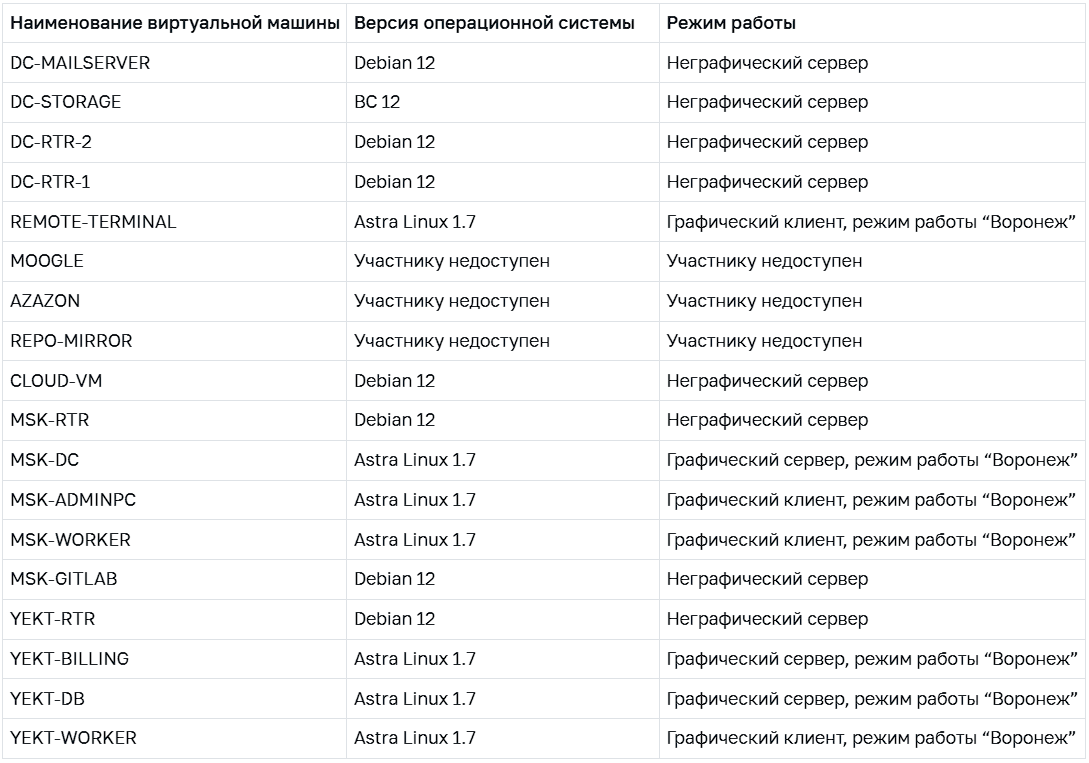
* Если вам необходимо установить пароль, но в задании не указано, какой именно — установите P@ssw0rd.
* Если вы по каким-либо причинам хотите использовать свои собственные учетные данные/адреса/DNS имена — не забудьте указать их в черновике и оставить его на вашем рабочем месте.
* Внимательно проверяйте свою работу. Если эксперты не смогут подключиться к чему-либо по какой-либо причине (сетевая недоступность, не подходят учетные данные) — проверка этого устройства производится не будет.
* Любые дополнительные комментарии и пожелания на проверку вы можете оставить для экспертов в черновике. Ваши комментарии гарантированно будут прочитаны перед проверкой или во время проверки.
* Если вы столкнулись с непониманием пункта задания — поднимите руку и спросите.
* Вы можете расходовать ваше время так, как вам этого хочется. Помните, что время, затраченное на ваши нужды, не компенсируется, за исключением обеда и случаев проблем со стороны площадки.

**Легенда конкурсного задания**

По легенде конкурсного задания ваша организация - Холдинг “CoolCompany”, занимается аудитом систем информационной безопасности, а также проводит разработку собственного коммерческого продукта - “Afisha Today”. Это платформа для малых предприятий и частных лиц в индустрии развлечений и досуга. Платформа за фиксированную комиссию предлагает разместить объявление на открытом ресурсе. Платформа берёт на себя обязательства помочь с регистрацией слушателей и полное информационное сопровождение мероприятий.

Компания имеет серверную стойку в ЦОД, офис в Москве, аренду виртуальной машины в публичном облаке “Cloud Happy” а также недавно Холдинг вышел на новый рынок, завершив покупку ГК “Минералы Урала” - вся ИТ-инфраструктура горнодобывающей компании, как и все её процессы, теперь перешли в ответственность “CoolCompany”. Ранее в организации “Минералы Урала” работала другая ИТ-компания и совершенно не оставила документации после себя. Необходимо будет исправить текущие процессы и проверить работу.

**Используемые операционные системы:**

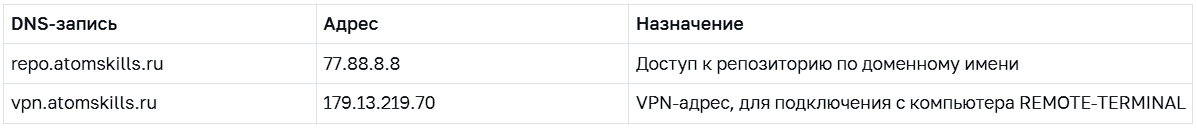


Машина REPO-MIRROR представляет собой прокси-сервер, выполняющий трансляцию запросов на официальные репозитории Debian 12 (mirror.yandex.ru), Ubuntu 24.04 (mirror.yandex.ru), Astra Linux 1.7 (dl.astralinux.ru).

Также, на машине REPO-MIRROR сохранены в папке /configs полезные документы и готовые файлы конфигураций, которыми будет удобно пользоваться при выполнении задания.

В инфраструктуре задания выход в “настоящий” интернет непредусмотрен, система лишь эмулирует работы реального Интернета, но напрямую выход недоступен.

На провайдере ISP (77.88.8.1) настроен DNS с следующими записями:



DNS-запись Адрес Назначение

repo.atomskills.ru 77.88.8.8 Доступ к репозиторию по доменному имени

vpn.atomskills.ru 179.13.219.70 VPN-адрес, для подключения с компьютера REMOTE-TERMINAL

**Работы в ЦОД**

# [На всех устройствах, указанных на топологии, создайте и настройте L3-адреса.](#_На_всех_устройствах,_1)

# [1. Все устройства (за исключением DC-STORAGE) должны прослушивать 22 порт SSH исключительно на внутреннем интерфейсе, по протоколу IPv4. Логин доступа - cod\_admin:](#_Все_устройства_(за)

* + Доступ исключительно по ключу. Ключ сохраните на сервере DC-STORAGE в каталоге - /ssh\_keys, у пользователя cod\_admin в домашнем каталоге никаких ключей быть не должно;
  + Для DC-STORAGE доступ по SSH с пользователем cod\_admin по паролю для подключенных пользователей через OpenConnect - пароль для подключения по SSH- At0mSk1lls;
  + На всех целевых машинах в датацентре пользователь cod\_admin должен иметь доступ к команде sudo без запроса пароля.

# [Между DC-RTR-2, DC-RTR-1, MSK-RTR, YEKT-RTR настройте удалённый защищённый туннель:](#_Между_DC-RTR-2,_DC-RTR-1,)

* + В качестве инструмента защиты используйте любое решение;
  + Важно организовать следующую логику работы соединений:
    - В случае, если соединение пропадает с Московского офиса до ЦОД - канал связи автоматически перестраивается на соединение с Екатеринбургом и перенаправляет трафик через резервного провайдера;
    - В случае, если соединение пропадает из офиса в Екатеринбурге - канал связи аналогично переключается на Московский офис и проходит по другому каналу связи;
    - При восстановлении соединения, система обратно переключается на основной канал связи;
    - Необходимо обеспечить бесперебойную работу сервера почты для всех филиалов, используйте этот сервис в качестве проверки доступности;
    - Основное соединение, ровно, как и резервные обязательно должны быть зашифрованы;
    - Вы можете самостоятельно проверить механизм отключения “провайдера”, под пользователем test с паролем P@ssw0rd в домашнем каталоге расположены скрипты, отвечающие за данную настройку;
    - При нормальной работе системы, туннели должны быть настроены в строгом соответствии с Топологией VPN. Дополнительные интерфейсы использовать запрещено.

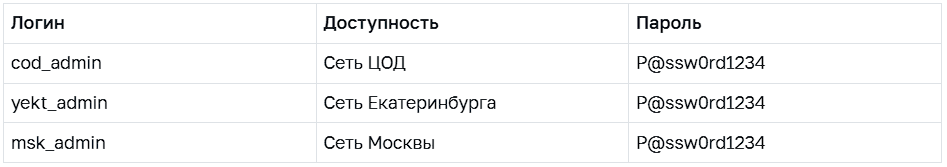
# **[3. Для обеспечения сетевого взаимодействия настройте протокол динамической маршрутизации OSPF на роутерах:](#_3._Для_обеспечения_1)**

* + Обеспечьте работу OSPF только на туннельных интерфейсах;
  + Обеспечьте аутентификацию через в OSPF по ключу - C00lCompanY;

# **[4. Для выхода в сеть «Интернет» используйте PAT, настроенный на пограничных роутерах соответственно.](#_4._Для_выхода_1)**

1. На базе сервера CLOUD-VM1 необходимо реализовать OpenConnect со следующей логикой работы:
   * Машина CLOUD-VM1 является публичным ресурсом, и компания не скрывает её адрес, а вот адреса роутеров в ЦОДе компания не планирует опубликовывать, именно поэтому CLOUD-VM1 является брокером подключений;
   * Машина CLOUD-VM1 подключается только к DC-RTR-1. Отказоустойчивое соединение настраивать не планируется;
   * Для OpenConnect используйте сертификат, выданный основным ЦС в городе Москва.
2. Клиентский компьютер REMOTE-TERMINAL подключается на CLOUD-VM1 через OpenConnect-client. В зависимости от логина, меняется сетевая доступность внутренних ресурсов компании, таблица доступности объясняет это подробнее.

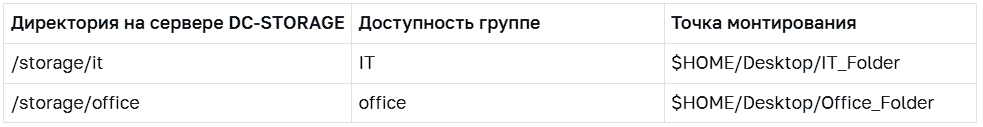
**Таблица доступности:**



# [На машине DC-STORAGE реализуйте файловый сервер на базе NFS:](#_На_машине_DC-STORAGE_1)

* + Обеспечьте автомонтирование NFS-сервера на компьютер MSK-ADMINPC и MSK-WORKER через pam\_mount;
  + Монтирование происходит на основании членства пользователя в указанной группе;
  + Планируется, что все папки будут на рабочем столе пользователя;
  + Настройте каждую папку так, чтобы пользователи даже при наличии максимального доступа к файлам не имели права на удаление тех файлов и директорий, владельцами которых они не являются;
  + Права на директорию можете назначить любые, главное сохранить условие монтирование папки в нужную группу.

**Таблица опубликованных файловых ресурсов:**



Первое:

Директория на сервере DC-STORAGE - /storage/it

Доступность группе – IT

Точка монтирования - $HOME/Desktop/IT\_Folder

Второе:

Директория на сервере DC-STORAGE - /storage/office

Доступность группе – office

Точка монтирования - $HOME/Desktop/Office\_Folder

1. Реализуйте систему резервного копирования важных данных:
   * Необходимо сделать так, что после создания пользователем из группы OFFICE в каталоге его группы файла с именем “SAVE” в любом регистре и в любом месте наименования файла (например, saveme.txt; SAVE\_Customer.docx;sAvE\_billing.pdf;booking\_save\_plz.xslx и т.д.) файл автоматически сохранялся на отдельно смонтированный Crypto-LVM массив на сервере DC-STORAGE /crypto-folder.
   * В случае, если файл был удалён пользователем, то он не удаляется с места резервного копирования.

# **[Реализуйте LVM-массив на DC-STORAGE по следующим требованиям:](#_Реализуйте_LVM-массив_на_1)**

* + LVM-массив из трёх дисков, объемом в 1ГБ ОЗУ;
  + Обеспечьте шифрование тома средствами dm-crypt. Диск должен монтироваться при загрузке ОС без запроса пароля;
  + Смонтируйте в каталог /crypto-folder;

1. На машине DC-MAILSERVER уже развернут почтовый сервер организации.
   * Почтовая система является частью критической инфраструктуры, используйте адрес этой машины для проверки 100% доступности с филиалов.

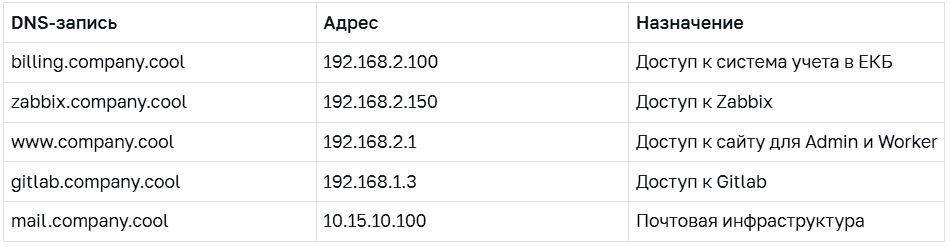
# [Реализуйте отказоустойчивый шлюз по умолчанию в инфраструктуре ЦОД:](#_Реализуйте_отказоустойчивый_шлюз_1)

* + Используйте протокол VRRP (или его аналоги);
  + Адрес шлюза - 10.15.10.1 на всех устройствах в ЦОД.

**Настройка офиса в Москве**

# [Убедитесь, что домен FreeIPA на MSK-DC1 развернут корректно:](#_Убедитесь,_что_домен_1)

* + Имя домена — company.cool;
  + На сервере настроен DNS со следующими записями, вы можете добавить собственные при необходимости.



1. Импортируйте пользователей в домен. Используйте для этого файл по пути - /root/users.csv;
   * В качестве логина используйте фамилию пользователя;
   * В качестве пароля - P@ssw0rd;
   * Сделайте так, чтобы пароль не требовал смены для всех пользователей.
2. Добавьте компьютеры MSK-ADMINPC и MSK-WORKER в домен FreeIPA.

# [На компьютере MSK-DC1 разверните Основной Центр сертификации:](#_На_компьютере_MSK-DC1_1)

* + Все веб-сервисы должны быть доступны только по протоколу HTTPS, OpenConnect также.
  + Все сертификаты должны быть выпущены Центром сертификации в городе Москва и добавлены в доверенные на всех компьютерах инфраструктуры, если в задании не сказано обратное;
  + Каталог для центра сертификации - /root/ca;
  + Имя центра сертификации - Cool CA;
  + Страна “Россия”;
  + Город “Москва”;
  + Область “Московская”.

# [Настройте ограничение на работу компьютера в нерабочее время:](#_Настройка_временного_ограничения)

* Компьтер MSK-WORKER позволяет заходить на компьютер только в промежуток с 09:00 до 18:00 по МСК. Если пользователь пытается зайти вне этого времени, то после ввода логина и пароля система его не пустит. При этом активную сессию прерывать не нужно.
* Локальному пользователю administrator доступ разрешить
* Настройке изображение на экране блокировки таким образом, чтобы было указано “В случае необходимости доступа к рабочему месту вне регламентированных работ, напишите на почту - admin@company.cool”. Или организуйте любой другой способ информирования пользователя о необходимости связи с администраторами для получения доступа к ПК.

# [MSK-ADMINPC позволяет входить в систему только пользователям из группы IT. Остальным вход запрещен](#_MSK-ADMINPC_позволяет_входить_1)

* + Настройте sudo для данной группы через парольную авторизацию.

1. На машине MSK-GITLAB:
   * На машине GITLAB необходимо настроить SAST.

# [Настройте DHCP-сервер на машине MSK-RTR:](#_Настройте_DHCP-сервер_на_1)

* + Диапазон адресов - 192.168.1.50 - 192.168.1.100;
  + DNS-сервер - 192.168.1.2 и 77.88.8.1;
  + Домен поиска - company.cool.

**Настройка офиса в Екатеринбурге**

1. На сервере YEKT-BILLING развернута самописная система контроля за добычей горных пород, система работает с базой данных на сервере YEKT-DB, необходимо исправить следующие особенности платформы:
   * Сервер работает медленно, выявите причину и устраните неисправность;
   * Согласно внутреннему регламенту компании, информация о добытых горных породах храниться только в течение последних 15 лет. Данные старше следует отправить на холодное хранение на сервер DC-STORAGE в каталог /crypto-folder, а из базы данных удалить.

# **[На сервере YEKT-DB уже есть сервер Zabbix:](#_На_сервере_YEKT-DB)**

* + Подключите к серверу HTTPS;
  + Настройте метрики Zabbix так, что в случае перегрузки CPU больше чем на 80% и переполнения диска более чем на 90%, Zabbix автоматически писал уведомления на почту - **admin@company.cool**.
  + почтовый аккаунт admin@company.cool подключен к MSK-ADMINPC для доменного пользователя - admin\_infra (импортируемый из CSV-файла)

1. Сбор логов организуйте на машине YEKT-WORKER, на рабочем столе локального пользователя - yekt\_admin с паролем P@ssw0rd. Директория $HOME/Desktop/Logs/, где NAME имя сервера, с которого был взят лог файл:
   * Логи собирать только веб-сервера на YEKT-BILLING и YEKT-DB;
   * Синхронизация логов происходит мгновенно;
   * Используйте любое техническое решение для реализации данной задачи.

# [На сервере YEKT-RTR разверните подчиненный Центр сертификации:](#_На_сервере_YEKT-RTR_1)

* + Подпишите подчиненный сертификат вашим основном ЦС в городе Москва;
  + Каталог для центра сертификации - /sub/ca;
  + Имя центра сертификации - Yekaterinburg CA;
  + Остальные сведения наследуются с основного ЦС.

1. Испольуя подчиненный Центр сертификации обеспечьте корректную работу YEKT-BILLING приложения по HTTPS.

# **[Реализуйте веб-сайт, на базе YEKT-RTR:](#_Реализуйте_веб-сайт,_на_1)**

* + веб-сайт с поддержкой авторизации согласено таблице пин-кодов;
  + пользователь вводит свой уникальный ПИН-код (исходя из таблицы пин-кодов);
  + при переходе на корневую страницу сайта пользователю предлагается на выбор две ссылки:
    - click here for admin;
    - click here for worker.
  + после выбора нужной ссылки для пользователя запрашивается авторизация на сайте, пользователь наблюдает HTML-страницу:
    - Если зашел как /admin, страница пишет - Hello Admins!
    - Если зашел как /worker, страница пишет - Hello Workers!

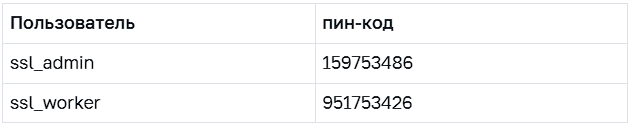
Пин-код:

Пользователь – ssl\_admin

Пин-код – 159753486

Пользователь – ssl\_worker

Пин-код - 951753426



# **[На всех устройствах, указанных на топологии, создайте и настройте L3-адреса.](#_На_всех_устройствах,)**

**DC-RTR-1 (пограничный роутер):**

enp1s0 – 200.100.100.20/24

enp6s0 – 10.15.10.3/24 , 10.15.10.1/30

gre1 – 10.7.7.1/30

gre2 – 10.6.6.1/30

**DC-RTR-2 (пограничный роутер):**

enp1s0 -100.200.100.20/24

enp6s0 – 10.15.10.2/24

gre1 – 10.5.5.1/30

gre2 – 10.8.8.1/30

**MSK-RTR (пограничный роутер):**

enps1s0 – 188.121.90.2/30

enp6s0 – 192.168.1.1/24

gre1 – 10.7.7.2/30

gre2 – 10.5.5.2/30

**YEKT-RTR (пограничный роутер):**

enp1s0 – 88.8.8.27/27

enp6s0 – 192.168.2.1/24

gre1 – 10.6.6.2/30

gre2 – 10.8.8.2/30

# **[Настройте ограничение на работу компьютера в нерабочее время:](#_Настройте_ограничение_на)**

Используем модуль **PAM (pam\_time)** для контроля времени входа.

1. Создаем пользователей для эксперимента:

**useradd worker1 -m -s /bin/bash**

**passwd worker1**

**useradd worker2 -m -s /bin/bash**

**passwd worker2**

**useradd philipp -m -s /bin/bash**

**passwd philipp**

**useradd administrator -m -s /bin/bash**

**passwd administrator**

1. Ставим таймзону MSK:

Проверить:

**timedatectl**

Поставить:

**timedatectl set-timezone Europe/Moscow**

1. vim /etc/security/time.conf

# Разрешить administrator всегда

**\*;\*;administrator;Al0000-2400**

# Остальные пользователи — только с 09:00 до 18:00

**\*;\*;!administrator;Al0900-1800**

1. vim /etc/pam.d/login

**account required pam\_time.so**

1. vim /etc/pam.d/common-auth

**account required pam\_time.so**

1. apt-get install gettext
2. msgunfmt /usr/share/locale/ru/LC\_MESSAGES/Linux-PAM.mo -o pam\_ru.po
3. vim pam\_ru.po
4. Ищем строку “Доступ запрещен” и исправляем ее на нужный нам текст

В случае необходимости доступа к рабочему месту вне регламентированных работ, напишите на почту - admin@company.cool

1. msgfmt pam\_ru.po -o Linux-PAM\_new.mo
2. sudo cp Linux-PAM\_new.mo /usr/share/locale/ru/LC\_MESSAGES/Linux-PAM.mo
3. sudo chmod 644 /usr/share/locale/ru/LC\_MESSAGES/Linux-PAM.mo

# **[MSK-ADMINPC позволяет входить в систему только пользователям из группы IT. Остальным вход запрещен](#_MSK-ADMINPC_позволяет_входить)**

Для теста добавляем группу и пользаков

groupadd IT

**useradd worker1 -m -s /bin/bash**

**passwd worker1**

**useradd worker2 -m -s /bin/bash**

**passwd worker2**

**useradd philipp -m -s /bin/bash**

**passwd philipp**

usermod -aG IT philipp

**Меняем editor для visudo на vim:**

sudo update-alternatives --config editor

sudo update-alternatives --install /usr/bin/editor editor /usr/bin/vim.basic 100

sudo visudo:

Добавить строку

%IT ALL=(ALL:ALL) ALL

**Нужно убедиться, что нет таких строк:**

%IT ALL=(ALL:ALL) NOPASSWD: ALL

Далее блокировка входа всем тем, кто не в группе IT:

vim /etc/security/access.conf

В конец файла добавляем строчки:

+:IT:ALL

-:ALL:ALL

В vim /etc/pam.d/login

Добавялем:

account required pam\_access.so

В vim /etc/pam.d/common-auth

Добавляем:

account required pam\_access.so

# **[Между DC-RTR-2, DC-RTR-1, MSK-RTR, YEKT-RTR настройте удалённый защищённый туннель:](#_Между_DC-RTR-2,_DC-RTR-1,_1)**

**Настраиваем туннели**

**На роутере DC-RTR-1:**

# Туннель до MSK-RTR

ip tunnel add gre1 mode gre local <IP-DC-RTR-1> remote <IP-MSK-RTR> ttl 255

ip addr add 10.7.7.1/30 dev gre1

ip link set gre1 up

# Туннель до YEKT-RTR

ip tunnel add gre2 mode gre local <IP-DC-RTR-1> remote <IP-YEKT-RTR> ttl 255

ip addr add 10.6.6.1/30 dev gre2

ip link set gre2 up

**На роутере DC-RTR-2:**

# Туннель до MSK-RTR

ip tunnel add gre1 mode gre local <IP-DC-RTR-2> remote <IP-MSK-RTR> ttl 255

ip addr add 10.5.5.1/30 dev gre1

ip link set gre1 up

# Туннель до YEKT-RTR

ip tunnel add gre2 mode gre local <IP-DC-RTR-2> remote <IP-YEKT-RTR> ttl 255

ip addr add 10.8.8.1/30 dev gre2

ip link set gre2 up

**На роутере MSK-RTR:**

# Туннель до DC-RTR-1

ip tunnel add gre1 mode gre local <IP-MSK-RTR> remote <IP-DC-RTR-1> ttl 255

ip addr add 10.7.7.2/30 dev gre1

ip link set gre1 up

# Туннель до DC-RTR-2

ip tunnel add gre2 mode gre local <IP-MSK-RTR> remote <IP-DC-RTR-2> ttl 255

ip addr add 10.5.5.2/30 dev gre2

ip link set gre2 up

**На роутере YEKT-RTR:**

# Туннель до DC-RTR-1

ip tunnel add gre1 mode gre local <IP-YEKT-RTR> remote <IP-DC-RTR-1> ttl 255

ip addr add 10.6.6.2/30 dev gre1

ip link set gre1 up

# Туннель до DC-RTR-2

ip tunnel add gre2 mode gre local <IP-YEKT-RTR> remote <IP-DC-RTR-2> ttl 255

ip addr add 10.8.8.2/30 dev gre2

ip link set gre2 up

**Чтобы добавить скрипт в автозапуск:**

chmod + x /path/to/script.sh

vim /etc/crontab

@reboot root /bin/bash /path/to/script.sh

**Настройка IPsec VPN:**

**На роутере DC-RTR-1:**

**/etc/ipsec.conf:**

config setup

uniqueids=no

conn %default

keyexchange=ikev2

ike=aes256-sha2\_256-modp2048

esp=aes256-sha2\_256

dpdaction=restart

dpddelay=30s

dpdtimeout=120s

keyingtries=%forever

authby=secret

# Туннель с MSK-RTR

conn dc1-msk

left=<IP-DC-RTR-1>

right=<IP-MSK-RTR>

auto=start

type=transport

# Туннель с YEKT-RTR

conn dc1-yekt

left=<IP-DC-RTR-1>

right=<IP-YEKT-RTR>

auto=start

type=transport

**/etc/ipsec.secrets:**

<IP-DC-RTR-1> <IP-MSK-RTR>: PSK "C00lCompanY"

<IP-DC-RTR-1> <IP-YEKT-RTR> : PSK "C00lCompanY"

systemctl restart ipsec

systemctl enable --now ipsec

**На роутере DC-RTR-2:**

**/etc/ipsec.conf:**

config setup

uniqueids=no

conn %default

keyexchange=ikev2

ike=aes256-sha2\_256-modp2048

esp=aes256-sha2\_256

dpdaction=restart

dpddelay=30s

dpdtimeout=120s

keyingtries=%forever

authby=secret

# Туннель с MSK-RTR

conn dc2-msk

left=<IP-DC-RTR-2>

right=<IP-MSK-RTR>

auto=start

type=transport

# Туннель с YEKT-RTR

conn dc2-yekt

left=<IP-DC-RTR-2>

right=<IP-YEKT-RTR>

auto=start

type=transport

**/etc/ipsec.secrets:**

<IP-DC-RTR-2> <IP-MSK-RTR>: PSK "C00lCompanY"

<IP-DC-RTR-2> <IP-YEKT-RTR>: PSK "C00lCompanY"

systemctl restart ipsec

systemctl enable --now ipsec

**На роутере MSK-RTR:**

**/etc/ipsec.conf:**

config setup

uniqueids=no

conn %default

keyexchange=ikev2

ike=aes256-sha2\_256-modp2048

esp=aes256-sha2\_256

dpdaction=restart

dpddelay=30s

dpdtimeout=120s

keyingtries=%forever

authby=secret

# Туннель с DC-RTR-1

conn msk-dc1

left=<IP-MSK-RTR>

right=<IP-DC-RTR-1>

auto=start

type=transport

# Туннель с DC-RTR-2

conn msk-dc2

left=<IP-MSK-RTR>

right=<IP-DC-RTR-2>

auto=start

type=transport

**/etc/ipsec.secrets:**

<IP-MSK-RTR> <IP-DC-RTR-1>: PSK "C00lCompanY"

<IP-MSK-RTR> <IP-DC-RTR-2>: PSK "C00lCompanY"

**На роутере YEKT-RTR:**

**/etc/ipsec.conf:**

config setup

uniqueids=no

conn %default

keyexchange=ikev2

ike=aes256-sha2\_256-modp2048

esp=aes256-sha2\_256

dpdaction=restart

dpddelay=30s

dpdtimeout=120s

keyingtries=%forever

authby=secret

# Туннель с DC-RTR-1

conn yekt-dc1

left=<IP-YEKT-RTR>

right=<IP-DC-RTR-1>

auto=start

type=transport

# Туннель с DC-RTR-2

conn yekt-dc2

left=<IP-YEKT-RTR>

right=<IP-DC-RTR-2>

auto=start

type=transport

**/etc/ipsec.secrets:**

<IP-YEKT-RTR> <IP-DC-RTR-1>: PSK "C00lCompanY"

<IP-YEKT-RTR> <IP-DC-RTR-2>: PSK "C00lCompanY"

**Конфигурация Keepalived на каждом маршрутизаторе:**

**DC-RTR-1**:

**apt-get install keepalived**

**vim /etc/keepalived/keepalived.conf**

vrrp\_script chk\_gre1 {

script "ping -c 2 10.7.7.2" # Проверка туннеля до MSK-RTR

interval 2

weight -10

}

vrrp\_script chk\_gre2 {

script "ping -c 2 10.6.6.2" # Проверка туннеля до YEKT-RTR

interval 2

weight -10

}

vrrp\_instance GRE1 {

state MASTER

interface gre1

virtual\_router\_id 51

priority 100

advert\_int 1

authentication {

auth\_type PASS

auth\_pass C00lCompanY

}

track\_script {

chk\_gre1

}

}

vrrp\_instance GRE2 {

state MASTER

interface gre2

virtual\_router\_id 52

priority 100

advert\_int 1

authentication {

auth\_type PASS

auth\_pass C00lCompanY

}

track\_script {

chk\_gre2

}

}

**systemctl restart keepalived**

**systemctl enable --now keepalived**

**DC-RTR-2**:

**apt-get install keepalived**

**vim /etc/keepalived/keepalived.conf**

vrrp\_script chk\_gre1 {

script "ping -c 2 10.5.5.2" # Проверка туннеля до MSK-RTR

interval 2

weight -10

}

vrrp\_script chk\_gre2 {

script "ping -c 2 10.8.8.2" # Проверка туннеля до YEKT-RTR

interval 2

weight -10

}

vrrp\_instance GRE1 {

state MASTER

interface gre1

virtual\_router\_id 53

priority 100

advert\_int 1

authentication {

auth\_type PASS

auth\_pass C00lCompanY

}

track\_script {

chk\_gre1

}

}

vrrp\_instance GRE2 {

state MASTER

interface gre2

virtual\_router\_id 54

priority 100

advert\_int 1

authentication {

auth\_type PASS

auth\_pass C00lCompanY

}

track\_script {

chk\_gre2

}

}

**systemctl restart keepalived**

**systemctl enable --now keepalived**

**MSK-RTR:**

**apt-get install keepalived**

**vim /etc/keepalived/keepalived.conf**

vrrp\_script chk\_gre1 {

script "ping -c 2 10.7.7.1" # Проверка туннеля до DC-RTR-1

interval 2

weight -10

}

vrrp\_script chk\_gre2 {

script "ping -c 2 10.5.5.1" # Проверка туннеля до DC-RTR-2

interval 2

weight -10

}

vrrp\_instance GRE1 {

state MASTER

interface gre1

virtual\_router\_id 55

priority 100

advert\_int 1

authentication {

auth\_type PASS

auth\_pass C00lCompanY

}

track\_script {

chk\_gre1

}

}

vrrp\_instance GRE2 {

state MASTER

interface gre2

virtual\_router\_id 56

priority 100

advert\_int 1

authentication {

auth\_type PASS

auth\_pass C00lCompanY

}

track\_script {

chk\_gre2

}

}

**systemctl restart keepalived**

**systemctl enable --now keepalived**

**YEKT\_RTR:**

**apt-get install keepalived**

**vim /etc/keepalived/keepalived.conf**

vrrp\_script chk\_gre1 {

script "ping -c 2 10.6.6.1" # Проверка туннеля до DC-RTR-1

interval 2

weight -10

}

vrrp\_script chk\_gre2 {

script "ping -c 2 10.8.8.1" # Проверка туннеля до DC-RTR-2

interval 2

weight -10

}

vrrp\_instance GRE1 {

state MASTER

interface gre1

virtual\_router\_id 57

priority 100

advert\_int 1

authentication {

auth\_type PASS

auth\_pass C00lCompanY

}

track\_script {

chk\_gre1

}

}

vrrp\_instance GRE2 {

state MASTER

interface gre2

virtual\_router\_id 58

priority 100

advert\_int 1

authentication {

auth\_type PASS

auth\_pass C00lCompanY

}

track\_script {

chk\_gre2

}

}

**systemctl restart keepalived**

**systemctl enable --now keepalived**

З

# **[3. Для обеспечения сетевого взаимодействия настройте протокол динамической маршрутизации OSPF на роутерах:](#_3._Для_обеспечения)**

**НА ВСЕХ РОУТЕРАХ:**

**vim /etc/frr/daemons:**

**ospfd=yes**

**systemctl restart frr**

**systemctl enable --now frr**

Также на всех роутерах включаем forwarding

**vim /etc/sysctl.conf**

**sysctl -p**

**DC-RTR-1:**

configure terminal

router ospf

router-id 10.10.10.10

network 10.7.7.0/30 area 0

network 10.6.6.0/30 area 0

network 10.15.10.0/24 area 0

area 0 authentication message-digest

interface gre1

ip ospf message-digest-key 1 md5 C00lCompanY

ip ospf authentication message-digest

exit

interface gre2

ip ospf message-digest-key 1 md5 C00lCompanY

ip ospf authentication message-digest

exit

interface enp6s0

ip ospf message-digest-key 1 md5 C00lCompanY

ip ospf authentication message-digest

exit

exit

write

**DC-RTR-2:**

configure terminal

router ospf

router-id 20.20.20.20

network 10.5.5.0/30 area 0

network 10.8.8.0/30 area 0

network 10.15.10.0/24 area 0

area 0 authentication message-digest

interface gre1

ip ospf message-digest-key 1 md5 C00lCompanY

ip ospf authentication message-digest

exit

interface gre2

ip ospf message-digest-key 1 md5 C00lCompanY

ip ospf authentication message-digest

exit

interface enp6s0

ip ospf message-digest-key 1 md5 C00lCompanY

ip ospf authentication message-digest

exit

exit

write

**MSK-RTR:**

configure terminal

router ospf

router-id 30.30.30.30

network 10.7.7.0/30 area 0

network 10.5.5.0/30 area 0

network 192.168.1.0/24 area 0

area 0 authentication message-digest

interface gre1

ip ospf message-digest-key 1 md5 C00lCompanY

ip ospf authentication message-digest

exit

interface gre2

ip ospf message-digest-key 1 md5 C00lCompanY

ip ospf authentication message-digest

exit

interface enp6s0

ip ospf message-digest-key 1 md5 C00lCompanY

ip ospf authentication message-digest

exit

exit

write

**YEKT-RTR:**

configure terminal

router ospf

router-id 40.40.40.40

network 10.6.6.0/30 area 0

network 10.8.8.0/30 area 0

network 192.168.2.0/24 area 0

area 0 authentication message-digest

interface gre1

ip ospf message-digest-key 1 md5 C00lCompanY

ip ospf authentication message-digest

exit

interface gre2

ip ospf message-digest-key 1 md5 C00lCompanY

ip ospf authentication message-digest

exit

interface enp6s0

ip ospf message-digest-key 1 md5 C00lCompanY

ip ospf authentication message-digest

exit

exit

write

# [4. Для выхода в сеть «Интернет» используйте PAT, настроенный на пограничных роутерах соответственно.](#_4._Для_выхода)

**DC-RTR-1:**

**vim /etc/nftables.conf**

table inet my\_nat {

chain my\_masquerade {

type nat hook postrouting priority srcnat;

oifname "enp1s0" masquerade

}

}

**systemctl restart nftables**

**systemctl enable --now nftables**

**DC-RTR-2:**

**vim /etc/nftables.conf**

table inet my\_nat {

chain my\_masquerade {

type nat hook postrouting priority srcnat;

oifname "enp1s0" masquerade

}

}

**systemctl restart nftables**

**systemctl enable --now nftables**

**MSK-RTR:**

**vim /etc/nftables.conf**

table inet my\_nat {

chain my\_masquerade {

type nat hook postrouting priority srcnat;

oifname "enp1s0" masquerade

}

}

**systemctl restart nftables**

**systemctl enable --now nftables**

**YEKT-RTR:**

**vim /etc/nftables.conf**

table inet my\_nat {

chain my\_masquerade {

type nat hook postrouting priority srcnat;

oifname "enp1s0" masquerade

}

}

**systemctl restart nftables**

**systemctl enable --now nftables**

# **[Реализуйте веб-сайт, на базе YEKT-RTR:](#_Реализуйте_веб-сайт,_на)**

**1. Структура проекта**

project/

├── app/

│ ├── app.py

**mkdir project**

**mkdir app**

**cd app**

**vim app.py**

import os

import http.cookies

from http.server import BaseHTTPRequestHandler, HTTPServer, SimpleHTTPRequestHandler

import urllib.parse

from io import BytesIO

# База данных пользователей с пин-кодами

USERS = {

'ssl\_admin': {'pin': '159753486', 'role': 'admin'},

'ssl\_worker': {'pin': '951753426', 'role': 'worker'}

}

# Хранение сессий

sessions = {}

class SimpleHTTPRequestHandler(BaseHTTPRequestHandler):

def do\_GET(self):

"""Обработка GET-запросов"""

path = self.path

if path == '/':

self.show\_index\_page()

elif path.startswith('/login'):

self.show\_login\_page(path)

elif path == '/admin':

self.show\_admin\_page()

elif path == '/worker':

self.show\_worker\_page()

else:

self.send\_error(404, "Page not found")

def do\_POST(self):

"""Обработка POST-запросов"""

path = self.path

if path.startswith('/login'):

self.handle\_login\_request(path)

def send\_html(self, html\_content):

"""Отправка HTML контента в ответе"""

self.send\_response(200)

self.send\_header('Content-type', 'text/html')

self.end\_headers()

self.wfile.write(html\_content.encode('utf-8'))

def show\_index\_page(self):

"""Показ главной страницы с выбором роли"""

html\_content = '''

<html>

<head><title>YEKT-RTR</title></head>

<body>

<h1>Welcome to YEKT-RTR</h1>

<p>Select your role:</p>

<ul>

<li><a href="/login/admin">click here for admin</a></li>

<li><a href="/login/worker">click here for worker</a></li>

</ul>

</body>

</html>

'''

self.send\_html(html\_content)

def show\_login\_page(self, path):

"""Показ страницы входа для выбранной роли"""

role = path.split('/')[2]

html\_content = f'''

<html>

<head><title>Login for {role}</title></head>

<body>

<h1>Login for {role}</h1>

<form method="POST">

<label>Username:</label><input type="text" name="username"><br>

<label>PIN:</label><input type="password" name="pin"><br>

<button type="submit">Login</button>

</form>

</body>

</html>

'''

self.send\_html(html\_content)

def handle\_login\_request(self, path):

"""Обработка формы логина"""

content\_length = int(self.headers['Content-Length'])

post\_data = self.rfile.read(content\_length)

post\_data = urllib.parse.parse\_qs(post\_data.decode('utf-8'))

username = post\_data.get('username', [''])[0]

pin = post\_data.get('pin', [''])[0]

role = path.split('/')[2]

# Проверка логина

if username in USERS and USERS[username]['pin'] == pin and USERS[username]['role'] == role:

session\_id = os.urandom(16).hex()

sessions[session\_id] = {'username': username, 'role': role}

self.send\_cookie(session\_id)

if role == 'admin':

self.send\_redirect('/admin')

elif role == 'worker':

self.send\_redirect('/worker')

else:

self.send\_error\_page(role)

def send\_cookie(self, session\_id):

"""Отправка сессионного cookie"""

cookie = http.cookies.SimpleCookie()

cookie['session\_id'] = session\_id

self.send\_header('Set-Cookie', cookie.output(header='', sep=''))

def send\_redirect(self, location):

"""Перенаправление пользователя на другую страницу"""

self.send\_response(303)

self.send\_header('Location', location)

self.end\_headers()

def send\_error\_page(self, role):

"""Показ страницы с ошибкой авторизации"""

html\_content = f'''

<html>

<head><title>Login Error</title></head>

<body>

<h1>Invalid credentials for {role}!</h1>

<a href="/login/{role}">Try again</a>

</body>

</html>

'''

self.send\_html(html\_content)

def show\_admin\_page(self):

"""Показ страницы для администратора"""

session\_id = self.get\_session\_id()

if session\_id and sessions.get(session\_id, {}).get('role') == 'admin':

html\_content = '''

<html>

<head><title>Admin</title></head>

<body>

<h1>Hello Admins!</h1>

</body>

</html>

'''

self.send\_html(html\_content)

else:

self.send\_redirect('/login/admin')

def show\_worker\_page(self):

"""Показ страницы для рабочего"""

session\_id = self.get\_session\_id()

if session\_id and sessions.get(session\_id, {}).get('role') == 'worker':

html\_content = '''

<html>

<head><title>Worker</title></head>

<body>

<h1>Hello Workers!</h1>

</body>

</html>

'''

self.send\_html(html\_content)

else:

self.send\_redirect('/login/worker')

def get\_session\_id(self):

"""Получение session\_id из cookies"""

cookie\_header = self.headers.get('Cookie', '')

cookies = http.cookies.SimpleCookie(cookie\_header)

return cookies.get('session\_id', None).value if cookies.get('session\_id') else None

def run(server\_class=HTTPServer, handler\_class=SimpleHTTPRequestHandler, port=5000):

server\_address = ('', port)

httpd = server\_class(server\_address, handler\_class)

print(f'Starting server on port {port}...')

httpd.serve\_forever()

if \_\_name\_\_ == "\_\_main\_\_":

run(port=5000)

**python app.py**

**ТУТ НУЖЕН ИНЕТИК ПОЭТОМУ ДАЛЬШЕ СТАРАЯ ВЕРСИЯ НАПИСАНА**

**Установка необходимых компонентов**

apt-get update

apt-get install -y python3 python3-pip python3-venv docker.io docker-compose

**Структура проекта:**

yekt-rtr-website/

├── app/

│ ├── app.py

│ ├── requirements.txt

│ └── templates/

│ ├── index.html

│ ├── login.html

│ ├── admin.html

│ └── worker.html

├── Dockerfile

└── docker-compose.yml

**Создаем проект**

mkdir yekt-rtr-website

cd yekt-rtr-website

**Создаем Dockerfile: (Secret-key делается командой вначале)**

**openssl rand -hex 32**

FROM python:3.9-slim

WORKDIR /app

COPY app/requirements.txt .

RUN pip install --no-cache-dir -r requirements.txt

COPY app .

ENV FLASK\_SECRET\_KEY=dd8669310940d0d42b7659bc009d5bfab74427775144b99d8ded4823872e7353

EXPOSE 5000

CMD ["python", "app.py"]

**Создаем docker-compose.yml:**

version: '3.8'

services:

web:

build: .

ports:

- "5000:5000"

environment:

- FLASK\_SECRET\_KEY=dd8669310940d0d42b7659bc009d5bfab74427775144b99d8ded4823872e7353

**mkdir app**

**cd app**

**Создаем requirements.txt:**

flask

**Создаем app.py:**

from flask import Flask, render\_template, request, redirect, url\_for, session

import os

app = Flask(\_\_name\_\_)

app.secret\_key = os.environ.get('FLASK\_SECRET\_KEY', 'dev-secret-key')

# База данных пользователей

USERS = {

'ssl\_admin': {'pin': '159753486', 'role': 'admin'},

'ssl\_worker': {'pin': '951753426', 'role': 'worker'}

}

@app.route('/')

def index():

return render\_template('index.html')

@app.route('/login/<role>', methods=['GET', 'POST'])

def login(role):

if request.method == 'POST':

username = request.form.get('username')

pin = request.form.get('pin')

user = USERS.get(username)

if user and user['pin'] == pin and user['role'] == role:

session['authenticated'] = True

session['role'] = role

return redirect(url\_for(role))

return render\_template('login.html', role=role, error="Invalid credentials")

return render\_template('login.html', role=role)

@app.route('/admin')

def admin():

if not session.get('authenticated') or session.get('role') != 'admin':

return redirect(url\_for('login', role='admin'))

return render\_template('admin.html')

@app.route('/worker')

def worker():

if not session.get('authenticated') or session.get('role') != 'worker':

return redirect(url\_for('login', role='worker'))

return render\_template('worker.html')

if \_\_name\_\_ == '\_\_main\_\_':

app.run(host='0.0.0.0', port=5000)

**mkdir templates**

**cd templates**

**Создаем файл index.html:**

<!DOCTYPE html>

<html>

<head>

<title>YEKT-RTR</title>

</head>

<body>

<h1>Welcome to YEKT-RTR</h1>

<p>Select your role:</p>

<ul>

<li><a href="{{ url\_for('login', role='admin') }}">click here for admin</a></li>

<li><a href="{{ url\_for('login', role='worker') }}">click here for worker</a></li>

</ul>

</body>

</html>

**Создаем файл login.html:**

<!DOCTYPE html>

<html>

<head>

<title>Login</title>

</head>

<body>

<h1>Login for {{ role }}</h1>

{% if error %}

<p style="color: red;">{{ error }}</p>

{% endif %}

<form method="POST">

<div>

<label>Username:</label>

<input type="text" name="username" required>

</div>

<div>

<label>PIN:</label>

<input type="password" name="pin" required>

</div>

<button type="submit">Login</button>

</form>

</body>

</html>

**Создаем файл admin.html:**

<!DOCTYPE html>

<html>

<head>

<title>Admin</title>

</head>

<body>

<h1>Hello Admins!</h1>

</body>

</html>

**Создаем файл worker.html:**

<!DOCTYPE html>

<html>

<head>

<title>Worker</title>

</head>

<body>

<h1>Hello Workers!</h1>

</body>

</html>

**cd /root/yekt-rtr-website**

**docker-compose up -d**

# [На сервере YEKT-DB уже есть сервер Zabbix:](#_На_сервере_YEKT-DB_1)

Сначала поднимаем Zabbix (если его не будет также как на стенде):

1. apt-get update
2. apt install apache2 libapache2-mod-php7.3 -y
3. vim /etc/php/7.3/apache2/php.ini

Необходимо настроить параметр:

date.timezone = Europe/Moscow

1. vim /etc/apache2/apache2.conf

Необходимо настроить параметр:

AstraMode off

1. systemctl restart apache2
2. Проверяем что апач работает:

http://10.40.29.90/

1. apt install postgresql-11 php7.3-pgsql -y
2. vim /etc/postgresql/11/main/pg\_hba.conf

Добавляем параметры доступа:

local zabbix          zabbix trust

host     zabbix          zabbix          127.0.0.1/32 trust

1. Запускаем консольный клиент psql для управления СУБД Postgresql от имени пользователя postgres. В терминале:

sudo -u postgres psql

1. Создаем базу данных zabbix и тд:

CREATE DATABASE zabbix;

CREATE ROLE zabbix with login password 'P@ssw0rd';

GRANT ALL ON DATABASE zabbix to zabbix;

\q

1. apt install zabbix-frontend-php zabbix-server-pgsql -y
2. zcat /usr/share/zabbix-server-pgsql/{schema,images,data}.sql.gz | psql -h localhost zabbix Zabbix

P@ssw0rd

P@ssw0rd

1. sudo a2enconf zabbix-frontend-php
2. sudo systemctl reload apache2
3. sudo cp /usr/share/zabbix/conf/zabbix.conf.php.example /etc/zabbix/zabbix.conf.php
4. vim /etc/zabbix/zabbix.conf.php

Меняем параметры:

$DB[**'TYPE'**]                             = **'POSTGRESQL'**;

$DB[**'PASSWORD'**]                 = **'z@BBix'**;

1. sudo chown www-data:www-data /etc/zabbix/zabbix.conf.php
2. sudo systemctl reload apache2
3. sudo systemctl enable zabbix-server
4. sudo systemctl start zabbix-server
5. Авторизуемся на сервере:

Username: Admin

Password: zabbix

1. apt install apache2 openssl -y
2. mkdir /etc/apache2/ssl
3. Вот так вот

sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \

-keyout /etc/apache2/ssl/zabbix.key \

-out /etc/apache2/ssl/zabbix.crt \

-subj "/CN=YEKT-DB/O=zabbix.company.cool/C=RU"

25. sudo a2enmod ssl

26. sudo a2ensite default-ssl

27. vim /etc/apache2/sites-available/zabbix-ssl.conf

Туда пишем:

<VirtualHost \*:80>

ServerName YEKT-DB

Redirect permanent / https://YEKT-DB/

</VirtualHost>

<VirtualHost \*:443>

ServerName YEKT-DB

ServerAdmin admin@company.cool

DocumentRoot /usr/share/zabbix

SSLEngine on

SSLCertificateFile /etc/apache2/ssl/zabbix.crt

SSLCertificateKeyFile /etc/apache2/ssl/zabbix.key

<Directory "/usr/share/zabbix">

Options FollowSymLinks

AllowOverride None

Require all granted

</Directory>

ErrorLog ${APACHE\_LOG\_DIR}/zabbix\_error.log

CustomLog ${APACHE\_LOG\_DIR}/zabbix\_access.log combined

</VirtualHost>

28. sudo a2dissite 000-default.conf

sudo a2ensite zabbix-ssl.conf

29. sudo apache2ctl configtest

sudo systemctl restart apache2

ДОДЕЛАТЬ НАСТРОЙКУ МОНИТОРИНГА **Настройка мониторинга CPU и диска в Zabbix**

# [Убедитесь, что домен FreeIPA на MSK-DC1 развернут корректно:](#_Убедитесь,_что_домен)

# [На компьютере MSK-DC1 разверните Основной Центр сертификации:](#_На_компьютере_MSK-DC1)

1. Создаём структуру каталогов для CA

mkdir -p /root/ca

cd /root/ca

1. Генерация ключа CA:

openssl genrsa -out ca.key.pem 4096

1. Генерация самоподписанного сертификата CA:

openssl req -new -x509 -days 3650 -key ca.key.pem -out ca.cert.pem \

-subj "/C=RU/ST=Московская/L=Москва/O=Cool CA/CN=Cool CA Root Certificate Authority"

# [На сервере YEKT-RTR разверните подчиненный Центр сертификации:](#_На_сервере_YEKT-RTR)

**mkdir -p /sub/ca**

**cd /sub/ca**

**openssl genrsa -out sub.ca.key.pem 4096**

**openssl req -new -key sub.ca.key.pem -out sub.ca.csr.pem -subj "/C=RU/ST=Sverdlovsk/L=Yekaterinburg/O=Yekaterinburg CA/CN=Yekaterinburg CA"**

**На сервере MSK-DC1, где находится основной ЦС, подпишите запрос:**

**openssl x509 -req -in /sub/ca/sub.ca.csr.pem -CA /root/ca/ca.cert.pem -CAkey /root/ca/ca.key.pem -CAcreateserial -out /sub/ca/sub.ca.cert.pem -days 3650**

**Проверьте сертификат подчинённого ЦС:**

**openssl verify -CAfile /root/ca/ca.cert.pem /sub/ca/sub.ca.cert.pem**

# **[Настройте DHCP-сервер на машине MSK-RTR:](#_Настройте_DHCP-сервер_на)**

**На машине MSK-RTR:**

**apt-get install isc-dhcp-server -y**

**vim /etc/dhcp/dhcpd.conf**

ddns-update-style none;

authoritative;

option domain-name "company.cool";

option domain-name-servers 192.168.1.2, 77.88.8.1;

subnet 192.168.1.0 netmask 255.255.255.0 {

range 192.168.1.50 192.168.1.100;

option routers 192.168.1.1; # Адрес шлюза (маршрутизатора)

option domain-name-servers 192.168.1.2, 77.88.8.1; # DNS-серверы

option domain-name "company.cool"; # Домен поиска

default-lease-time 600;

max-lease-time 7200;

}

**vim /etc/default/isc-dhcp-server**

INTERFACESv4="enp6s0"

systemctl restart isc-dhcp-server

systemctl enable --now isc-dhcp-server

# [Реализуйте отказоустойчивый шлюз по умолчанию в инфраструктуре ЦОД:](#_Реализуйте_отказоустойчивый_шлюз)

На **DC-RTR-1** (Master):

**apt install keepalived -y**

**vim /etc/keepalived/keepalived.conf**

vrrp\_instance VI\_1 {

state MASTER

interface enp6s0

virtual\_router\_id 101

priority 100

advert\_int 1

authentication {

auth\_type PASS

auth\_pass C001CompanY

}

virtual\_ipaddress {

10.15.10.1

}

}

systemctl restart keepalived

systemctl enable --now keepalived

На **DC-RTR-2** (Backup):

**apt install keepalived -y**

**vim /etc/keepalived/keepalived.conf**

vrrp\_instance VI\_1 {

state BACKUP

interface enp6s0

virtual\_router\_id 101

priority 90

advert\_int 1

authentication {

auth\_type PASS

auth\_pass C001CompanY

}

virtual\_ipaddress {

10.15.10.1

}

}

systemctl restart keepalived

systemctl enable --now keepalived

# [Все устройства (за исключением DC-STORAGE) должны прослушивать 22 порт SSH исключительно на внутреннем интерфейсе, по протоколу IPv4. Логин доступа - cod\_admin:](#_1._Все_устройства)

На машинах DC-MAILSERVER**,** MSK-DC1, MSK-GITLAB, MSK-ADMINPC, MSK-WORKER **(MOSCOW),** YEKT-BILLING, YEKT-DB, YEKT-WORKER(не получается зайти в рут)) **(EKB).**

Создаем пользака:

**useradd -m -s /bin/bash cod\_admin**

**passwd cod\_admin**

**apt-get update**

**apt-get install ssh -y**

**vim /etc/ssh/sshd\_config**

PasswordAuthentication yes

**ТУТ НУЖНО С НАСТРОЕННОГО DC\_STORAGE КОПИРНУТЬ КЛЮЧ**

**passwd -l cod\_admin**

**vim /etc/ssh/sshd\_config**

Port 22

ListenAddress <внутренний\_IP\_устройства>

AddressFamily any

PermitRootLogin no

PasswordAuthentication no

PubkeyAuthentication yes

AuthorizedKeysFile %h/.ssh/authorized\_keys

mkdir -p /ssh\_keys/authorized\_keys

**systemctl restart ssh**

**systemctl enable --now ssh**

**На DC-STORAGE:**

**useradd -m -s /bin/bash cod\_admin**

**passwd -l cod\_admin**

**apt-get install sudo ssh -y**

**Настройка SSH на DC-STORAGE**

**vim /etc/ssh/sshd\_config:**

PasswordAuthentication yes

PermitRootLogin no

Match User cod\_admin

PasswordAuthentication yes

AuthenticationMethods password

**update-alternatives --config editor**

**visudo**

**echo "cod\_admin ALL=(ALL) NOPASSWD:ALL" | sudo tee /etc/sudoers.d/cod\_admin**

**mkdir -p /ssh\_keys**

**chmod 700 /ssh\_keys**

**ssh-keygen -t rsa -b 4096 -f /ssh\_keys/cod\_admin -N ""**

**chmod 600 /ssh\_keys/cod\_admin.pub**

**ssh-copy-id -i /ssh\_keys/cod\_admin.pub cod\_admin@<внутренний\_IP>**

10.15.10.100

192.168.1.2

192.168.1.50

192.168.1.51

192.168.1.52

192.168.2.100

192.168.2.150

**ssh -i /ssh\_keys/cod\_admin cod\_admin@10.15.10.100**

**vim ~/.ssh/config:**

Host 10.15.10.\*

User cod\_admin

IdentityFile /ssh\_keys/cod\_admin

IdentitiesOnly yes

**systemctl restart ssh**

# **[На машине DC-STORAGE реализуйте файловый сервер на базе NFS:](#_На_машине_DC-STORAGE)**

**На DC-STORAGE:**

**apt-get update**

**apt-get install nfs-kernel-server**

**Создание директорий и настройка прав**

**mkdir -p /storage/it**

**mkdir -p /storage/office**

**groupadd IT**

**groupadd office**

**chown root:IT /storage/it**

**chown root:office /storage/office**

**chmod 1775 /storage/it**

**chmod 1775 /storage/office**

**vim /etc/exports:**

/storage/it \*(rw,sync,no\_root\_squash,no\_subtree\_check)

/storage/office \*(rw,sync,no\_root\_squash,no\_subtree\_check)

**exportfs -ra**

**systemctl restart nfs-kernel-server**

**Настройка клиентов (MSK-ADMINPC и MSK-WORKER)**

**apt-get update**

**apt install nfs-common libpam-mount -y**

**vim /etc/security/pam\_mount.conf.xml:**

<?xml version="1.0" encoding="utf-8"?>

<!DOCTYPE pam\_mount SYSTEM "pam\_mount.conf.xml.dtd">

<pam\_mount>

<debug enable="1" />

<volume

user="\*"

fstype="nfs"

server="10.15.10.150"

path="/storage/it"

mountpoint="/home/%(USER)/Desktop/IT\_Folder"

options="vers=4,rw,noatime,nolock"

sgrp="IT" />

<volume

user="\*"

fstype="nfs"

server="10.15.10.150"

path="/storage/office"

mountpoint="/home/%(USER)/Desktop/Office\_Folder"

options="vers=4,rw,noatime,nolock"

sgrp="office" />

</pam\_mount>

**vim /etc/pam.d/common-auth:**

Добавляем эту строчку в конец:

auth optional pam\_mount.so

**vim /etc/pam.d/common-session:**

Строку добавляем ВНАЧАЛО

session optional pam\_mount.so

**vim /etc/profile:**

**В конец пишем:**

# Create mount points if they don't exist

if [ "$PS1" ] && [ -n "$USER" ] && [ "$(id -u)" -ge 1000 ]; then

# Only for interactive login shells and regular users

# For IT group

if groups "$USER" | grep -qw "IT"; then

mkdir -p "$HOME/Desktop/IT\_Folder" 2>/dev/null

chown "$USER:" "$HOME/Desktop/IT\_Folder" 2>/dev/null

fi

# For office group

if groups "$USER" | grep -qw "office"; then

mkdir -p "$HOME/Desktop/Office\_Folder" 2>/dev/null

chown "$USER:" "$HOME/Desktop/Office\_Folder" 2>/dev/null

fi

fi

**vim /etc/security/pam\_mount.conf.xml**

**showmount -e 10.15.10.150**

**mount -t nfs 10.15.10.150:/storage/it ~/Desktop/IT\_Folder**

**journalctl -xe | grep pam\_mount**

**cat /var/log/auth.log | grep pam\_mount**

**pmvarrun -u $(whoami) -o**

**mount -t nfs DC-STORAGE:/storage/it /home/administrator/Desktop/IT\_Folder**

**mount -t nfs -o vers=4,rw,noatime DC-STORAGE:/storage/it /home/<username>/Desktop/IT\_Folder**

# [Реализуйте LVM-массив на DC-STORAGE по следующим требованиям:](#_Реализуйте_LVM-массив_на)