Санкт-Петербургское государственное бюджетное
профессиональное образовательное учреждение
«Академия управления городской средой, градостроительства и печати»

УТВЕРЖДАЮ
Заместитель директора
по учебно-производственной работе
О.В. Фомичева
«16» 12 2025 г.

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ
по выполнению практических работ
по МДК.04.01 Конфигурирование и поддержка сетевой инфраструктуры
ПМ.04 СОПРОВОЖДЕНИЕ МОДЕРНИЗАЦИИ СЕТЕВОЙ ИНФРАСТРУКТУРЫ

для специальности

09.02.06 Сетевое и системное администрирование

Санкт-Петербург
2023 г.

Методические рекомендации рассмотрены на заседании методического совета СПб ГБПОУ «АУГСГиП»
Протокол № 2 от «19» 11 202 5 г.

Методические рекомендации одобрены на заседании цикловой комиссии информационных технологий
Протокол № 4 от «21» 11 202 3 г.

Председатель цикловой комиссии: Караченцева М.С. _____

Разработчики: преподаватели СПб ГБПОУ «АУГСГиП»

С О Д Е Р Ж А Н И Е

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Рабочая тетрадь по выполнению практических работ предназначена для организации работы на практических занятиях по МДК.04.01 «Конфигурирование и поддержка сетевой инфраструктуры», которая является важной составной частью в системе подготовки специалистов среднего профессионального образования по специальности 09.02.06 «Сетевое и системное администрирование».

Практические занятия являются неотъемлемым этапом изучения учебной дисциплины и проводятся с целью:

- формирования практических умений в соответствии с требованиями к уровню подготовки обучающихся, установленными рабочей программой учебной дисциплины;
- обобщения, систематизации, углубления, закрепления полученных теоретических знаний;
- готовности использовать теоретические знания на практике.

Практические занятия по МДК.04.01 «Конфигурирование и поддержка сетевой инфраструктуры» способствуют формированию в дальнейшем при изучении профессиональных модулей, следующих общих и профессиональных компетенций:

ОК 1. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам;

ОК 2. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;

ОК 3. Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях;

ОК 4. Эффективно взаимодействовать и работать в коллективе и команде;

ОК 5. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 6. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;

ОК 7. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;

ОК 8. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;

ОК 9. Пользоваться профессиональной документацией на государственном и иностранном языках.

ПК 4.1 Принимать участие в конфигурировании и поддержки сетевой инфраструктуры.

В рабочей тетради предлагаются к выполнению практические работы, предусмотренные учебной рабочей программой МДК.04.01 «Конфигурирование и поддержка сетевой инфраструктуры».

При разработке содержания практических работ учитывался уровень сложности освоения студентами соответствующей темы, общих и профессиональных компетенций, на формирование которых направлена дисциплина.

Выполнение практических работ в рамках МДК.04.01 «Конфигурирование и поддержка сетевой инфраструктуры» позволяет освоить комплекс работ по выполнению практических заданий по всем темам МДК.04.01 «Конфигурирование и поддержка сетевой инфраструктуры».

Рабочая тетрадь по МДК.04.01 «Конфигурирование и поддержка сетевой инфраструктуры» имеют практическую направленность и значимость. Формируемые в процессе практических занятий умения могут быть использованы студентами в будущей профессиональной деятельности.

Рабочая тетрадь предназначена для студентов колледжа, изучающих МДК.04.01 «Конфигурирование и поддержка сетевой инфраструктуры».

Оценки за выполнение практических работ выставляются по пятибалльной системе. Оценки за практические работы являются обязательными текущими оценками и выставляются в журнале теоретического обучения.

1. Перечень практических работ по МДК.04.01 «Конфигурирование и поддержка сетевой инфраструктуры»

№ раздела, темы	Освоение умений в процессе занятия	Формируемые ОК и ПК	Тема практического занятия	Кол-во часов
Тема 1.1. Модернизация локальных сетей	проводить аудит сетевой инфраструктуры; выявлять компоненты сетевой инфраструктуры, требующие модернизации; подбирать компоненты сетевой инфраструктуры в соответствии с выявленными потребностями;	ОК 1 – ОК 9 ПК 4.1	Практическая работа 1 Сбор информации о сети организации	2
			Практическая работа 2 Анализ исходных данных сети	2
			Практическая работа 3 Модернизация физических устройств	2
			Практическая работа 4 Замена оборудования	2
			Практическая работа 5 Настройка оборудования модернизированной сети	2
			Практическая работа 6 Повышение безопасности сети	2
			Практическая работа 7 Настройка программного обеспечения	2
			Практическая работа 8 Тестирование модернизированной сети	2
Тема 1.2. Мониторинг оборудования в локальной вычислительной	проводить аудит сетевой инфраструктуры; выявлять компоненты сетевой инфраструктуры, требующие модернизации;		Практическая работа 9 Мониторинг сети на сетевом уровне tcp/ip	2
			Практическая работа 10 Сравнительный анализ средств мониторинга сетей	2
			Практическая работа 11 Установка и настройка сервера Zabbix	2
			Практическая работа 12 Установка и настройка агента Zabbix на Windows	2
			Практическая работа 13 Установка и настройка агента Zabbix на Linux	2
			Практическая работа 14 Установка и настройка прокси сервера Zabbix	2
			Практическая работа 15 Создание шаблонов в	2

№ раздела, темы	Освоение умений в процессе занятия	Формируемые ОК и ПК	Тема практического занятия	Кол-во часов
			Zabbix	
			Практическая работа 16 Настройка оповещений Zabbix	2
			Практическая работа 17 Запуск скриптов на удаленных хостах	2
			Практическая работа 18 Мониторинг сетевого оборудования	2
			Практическая работа 19 Управление пользователями Zabbix	2
Тема 1.3. Обслуживание и поддержка компьютеров и рабочих мест	подбирать компоненты сетевой инфраструктуры в соответствии с выявленными потребностями;		Практическая работа 20 Настройка средств удаленного администрирования	2
			Практическая работа 21 Инвентаризация рабочих станций	2
			Практическая работа 22 Выявление неполадок в работе сети	2
			Практическая работа 23 Физическое обслуживание сетевого оборудования	2
			Практическая работа 24 Физическое обслуживание кабельной системы	2
			Практическая работа 25 Физическое обслуживание рабочих станций	2
			Практическая работа 26 Проверка работы мониторов	2
			Практическая работа 27 Обновление системного программного обеспечения	2
			Практическая работа 28 Установка прикладного программного обеспечение	2
			Практическая работа 29 Обновление прикладного программного обеспечения	2
			Практическая работа 30 Проверка	2

№ раздела, темы	Освоение умений в процессе занятия	Формируемые ОК и ПК	Тема практического занятия	Кол-во часов
			работоспособности периферийных устройств	
			Практическая работа 31 Обновление политик безопасности	2
			Практическая работа 32 Проверка учетных записей в AD	2
			Практическая работа 33 Проверка учетных записей в ALD	2
			Практическая работа 34 Перенос учетных записей из AD в ALD PRO	2
			Практическая работа 35 Настройка политик безопасности ALD PRO	2
Тема 1.4. Автоматизация управления сетью	проводить аудит сетевой инфраструктуры; выявлять компоненты сетевой инфраструктуры, требующие модернизации;		Практическая работа 36 Сравнительный анализ средств управления сетью	2
			Практическая работа 37 Управления сетью с помощью ИКС	2
			Практическая работа 38 Установка Ansible	2
			Практическая работа 39 Использование Ansible для установки пакетов, модулей и расширений	2
			Практическая работа 40 Использование Ansible для управления системой	2
			Практическая работа 41 Использование Ansible для сбора информации	2
			Практическая работа 42 Использование Ansible для работы с файлами	2

2. Описание порядка выполнения практических работ

Практическая работа №1 Сбор информации о сети организации

Задание:

1. Изучите сеть учебного класса.
2. Постройте физическую и логическую топологию сети.
3. Перечислите используемое активное сетевое оборудование.
4. Изучите состав и состояние пассивных компонентов локальной сети.
5. Опишите вид и форму получения внешних сервисов

Практическая работа № 2 Анализ исходных данных сети

Задание:

1. Проанализируйте информацию, полученную в ходе выполнения практической работы №1.
2. Дайте описание центральной точки.
3. Дайте описание пользовательских рабочих мест.
4. Дайте описание системы безопасности.
5. Выявите проблемы сети организации.
6. Предложите способы решения выявленных проблем.

Практическая работа № 3 Модернизация физических устройств

Задание:

1. Проанализируйте информацию, полученную в ходе выполнения практических работ №1, №2.
2. Опишите недостатки рабочих мест.
3. Предложите способы модернизации физических устройств рабочих мест.
4. Опишите процесс модернизации.
5. Предложите комплектующие, с помощью которых можно выполнить модернизацию.

Практическая работа № 4 Замена оборудования

Задание:

1. Проанализируйте информацию, полученную в ходе выполнения практических работ №1, №2.
2. Перечислите перечень оборудования, подлежащего замене.
3. Предложите оборудование на замену.
4. Опишите процесс замены оборудования.

5. Опишите преимущества замены выбранного оборудования.

Практическая работа № 5 Настройка оборудования модернизированной сети

Задание:

1. Для оборудования, которое было установлено в практической работе №4 произведите необходимые действия для стабильной работы сети.

   Все этапы настройки зафиксировать скриншотами.

Практическая работа № 6 Повышение безопасности сети

Задание:

1. Проанализируйте информацию, полученную в ходе выполнения практических работ №1, №2.
2. Предложите меры по повышению безопасности сети.
3. Реализуйте предложенные меры.

Практическая работа № 7 Настройка программного обеспечения

Задание:

1. Произведите обновление всего системного и программного обеспечения.
2. Обеспечьте переход на отечественное программное обеспечение.
3. Зафиксируйте замену ПО.
4. Произведите замену ПО на отечественное.

Практическая работа № 8 Тестирование модернизированной сети

Задание:

1. Проверьте работоспособность сети путем отправки эхо-запросов на разные рабчие станции.
2. Проверьте доступ к внешним ресурсам и работоспособность сети Интернет.
3. Проведите повторный аудит сети для проверки устранения выявленных недостатков.

Практическая работа №9 Мониторинг сети на сетевом уровне tcp/ip

Задание 1 Выполните команду ping в командной строке с различными значениями параметров -t, -n, -l, -i,-w. Какие наблюдения и выводы вы сделали?

ping www.seun.ru

ping www.sgu.ru

ping www.microsoft.com

ping www.sun.com

ping212.193.38.83

_____

_____

Выполните ping к тем же хостам с параметром –f, увеличивая параметр -l size. При каком значении размера перестают получаться ответы?

_____

_____

Задание 2. Tracert

Выполните команду tracert в командной строке с различными значениями параметров. Какие наблюдения и выводы вы сделали?

Используйте, например,

tracertwww.seun.ru

tracert www.sgu.ru

tracert www.microsoft.com

tracert www.sun.com

tracert 212.193.38.83

Задание 3. Поисковые сервисы Европейского и Российского ip-регистров

Определите, кому принадлежат сети 194.85.33.0, 217.23.64.0, 212.193.38.0. Для этого используйте поисковые аппараты http://www.ripe.net/db/whois/whois.htmlиhttp://www.ripn.net:8080/nic/whois/index.html.

Пользуясь данными этих информационных систем, попробуйте определите географическое расположение сетей. Попробуйте изобразить топологическую схему соединения этих сетей.

Задание 4. Использование программы ping для исследования параметров сети.

1. Приведите сравнительные результаты выполнения команд ping по адресам 194.85.33.29, 194.85.33.30, 217.23.64.2, 212.193.38.248, 212.193.35.10 по параметрам «время отклика», TTL в форме таблицы. Объясните полученные различия.

2. Соберите средние времена прохождения 10 пакетов на указанные адреса. Сравните с р4езультатами, полученными при использовании сервиса ping в интерфейсе Looking Glass на сайте http://noc.runnet.ru. Объясните полученные различия.

3. Соберите усредненные времена прохождения 10 пакетов увеличивающегося размера по указанным адресам. Начните с 64 байт и каждый раз удваивайте размер пакета. При каком размере пакета потери превышают 50 %. Как влияет время ожидания отклика на процент прохождения пакетов этого размера. При каком времени ожидания отклика потери для пакетов зафиксированного размера не возникают?

Представьте результаты измерений в форме таблиц, наилучшим образом проявляющим, по вашему мнению, обнаруженные зависимости.

4. Используя программу ping, оцените вклад разных сетевых участков, по которым проходит эхо-пакет между вашей рабочей станцией и интерфейсом 194.85.33.29.

Задание 5. Использование программы tracert для анализа соединений в сети.

1. Приведите сравнительные результаты выполнения команд tracert по адресам 194.85.33.29, 194.85.33.30, 217.23.64.2, 212.193.38.248, 212.193.35.10. Объясните полученные различия.

2. Выполните трассировку к адресу 212.193.38.248 и к адресу 217.23.64.2 со стороны сайта http://noc.runnet.ru. Приведите полученные результаты.

3. Используя данные, полученные в результате выполнения трассировки и отправки эхо-пакетов между интерфейсами 212.193.38.248 и 194.85.35.100, оцените вклад разных участков сетей, соединяющих эти интерфейсы, в среднее время прохождения пакетов между ними.

4. Используя полученную в ходе выполнения всех заданий информацию, уточните схему задания 1, изобразите на ней обнаруженные вами промежуточные интерфейсы и линки сети, объединяющей подсети 194.85.33.0, 217.23.64.0, 212.193.38.0.

Практическая работа № 10 Сравнительный анализ средств мониторинга сетей

Задание:

Произведите сравнительный анализ 3 средств управления сетью по критериям:

1. Автообнаружение
2. WMI
3. Без агента/с агентом
4. Триггеры

5. Доступ через веб-аккаунт
6. Метод хранения данных
7. Определение имени хоста по его адресу через сервер DNS
8. Максимальное рекомендуемое число обслуживаемых узлов
9. Распознавание сетевых топологий
10. Распределенное управление
11. Поддержка протокола SNMP
12. Поддержка ОС
13. Стоимость

Оформите отчет в удобной форме.

Практическая работа № 11 Установка и настройка сервера Zabbix

Задание:

Перед установкой Zabbix необходимо выполнить подготовительные процедуры.

1. Правильное время

Для получения актуальной информации необходимо, чтобы на сервере было правильное время.

Для этого сначала нужно задать правильную временную зону:

timedatectl set-timezone Europe/Moscow

** в данном примере задается московское время.*

Затем установить и запустить сервис синхронизации времени:

apt install chrony

systemctl enable chrony

systemctl start chrony

2. Настройка брандмауэра

Для работы сервера, открываем следующие порты:

iptables -I INPUT -p tcp --match multiport --dports 80,443 -j ACCEPT

iptables -I INPUT -p tcp --match multiport --dports 10050,10051 -j ACCEPT

iptables -I INPUT -p udp --match multiport --dports 10050,10051 -j ACCEPT

** где **80** — порт для http запросов (веб-интерфейс); **443** — для https запросов (веб-интерфейс); **10050** — порты для получения информации от zabbix агентов.*

Для сохранения правил используем

apt install iptables-persistent

netfilter-persistent save

3. Обновляем список пакетов в репозитории:

apt update

Если наша система чистая и на ней не работают критически важные сервисы, стоит обновить установленные пакеты:

apt upgrade

Подготовка системы закончена.

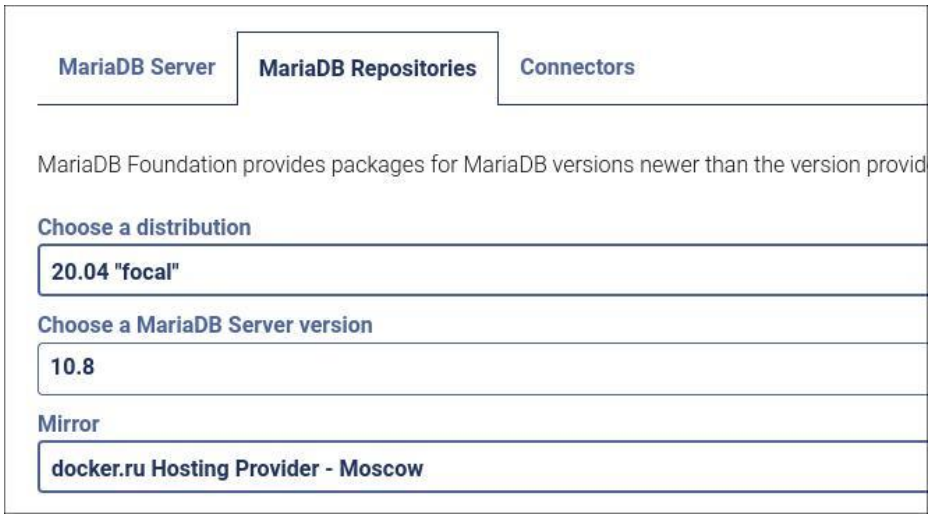
Установка веб-сервера

Управление сервером Zabbix будет осуществляться посредством веб-интерфейса. Для этого необходимо установить и настроить веб-сервер, СУБД и PHP.

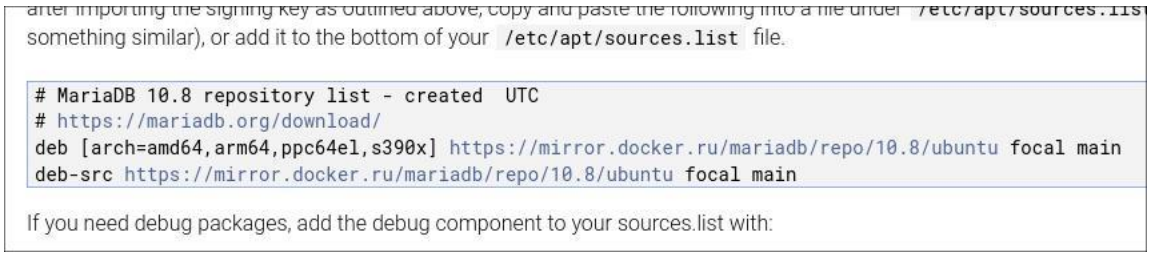
СУБД

В данному инструкции мы будем использовать сервер баз данных mariadb. Zabbix версии 6 требует MariaDB версии 10.05.00 и выше. Но слишком свежая версия также может не поддерживаться сервером заббикс. Необходимо уточнить системные требования на официальной странице.

В репозитории Ubuntu может не оказаться нужной версии СУБД, поэтому мы подключим репозиторий разработчика. Для этого переходим по ссылке downloads.mariadb.org/mariadb/repositories и выбираем нашу версию операционной системы, последний стабильный и поддерживаемый релиз mariadb и геолокацию репозитория, например:



Ниже появится инструкция по добавлению репозитория и установке СУБД:



Согласно инструкции, создаем файл:

vi /etc/apt/sources.list.d/matiadb.list

MariaDB 10.8 repository list - created UTC
https://mariadb.org/download/
deb [arch=amd64,arm64,ppc64el,s390x] https://mirror.docker.ru/mariadb/repo/10.8/ubuntu focal main
deb-src https://mirror.docker.ru/mariadb/repo/10.8/ubuntu focal main

Импортируем ключ репозитория:

apt-key adv --fetch-keys 'https://mariadb.org/mariadb_release_signing_key.asc'

Обновляем кэш пакетов:

apt update

Устанавливаем СУБД:

apt install mariadb-server

Разрешаем автозапуск сервера баз данных и запускаем mariadb:

systemctl enable mariadb

systemctl start mariadb

Задаем пароль для суперпользователя СУБД:

mysqladmin -u root password

** после ввода команды система потребует ввести пароль два раза.*

Веб-сервер

Для наших целей будем использовать веб-сервер NGINX.

Для его установки вводим команду:

apt install nginx

Запускаем nginx и разрешаем его автозапуск:

systemctl enable nginx

systemctl start nginx

Открываем веб-браузер и переходим по ссылке http://<IP-адрес сервера>/ — мы должны увидеть окно приветствия:

PHP и PHP-FPM

Интерфейс zabbix разработан на PHP — наш веб-сервер должен обрабатывать скрипты, написанные на нем. Смотрим на странице с системными требованиями, какая версия PHP рекомендована и создаем системную переменную для нашего удобства:

export PHP_VER=8.2

** в моем примере для Zabbix рекомендована версия 8.2.*

С помощью страницы Установка разных версий PHP на Linux Ubuntu или Установка разных версий PHP на Linux Debian устанавливаем нужную нам версию PHP.

Устанавливаем php компоненты. В зависимости от того, нативная у нас версия PHP или установленная из дополнительного репозитория, команды будут отличаться.

а) Для нативной версии:

apt install php php-fpm php-mysql php-pear php-cgi php-common php-ldap php-mbstring php-snmp php-gd php-xml php-bcmath

б) Установленной из репозитория:

apt install php${PHP_VER}-fpm php${PHP_VER}-mysql php${PHP_VER}-pear php${PHP_VER}-cgi php${PHP_VER}-common php${PHP_VER}-ldap php${PHP_VER}-mbstring php${PHP_VER}-snmp php${PHP_VER}-gd php${PHP_VER}-xml php${PHP_VER}-bcmath

Установка PHP и компонентов завершены. Переходим к настройке.

Для настройки php, открываем файл:

vi /etc/php/${PHP_VER}/fpm/php.ini

Редактируем следующие параметры:

date.timezone = "Europe/Moscow"

...

```
max_execution_time = 300
...
post_max_size = 16M
...
max_input_time = 300
...
max_input_vars = 10000
```

** где:*

- ***date.timezone* —** *временная зона. В нашем примере выставлено московское время.*
- ***max_execution_time* —** *разрешенное время выполнения скрипта. Если последний будет выполняться дольше, система прервет его работу.*
- ***post_max_size* —** *максимальные объем передачи данных методом POST.*
- ***max_input_time* —** *время в секундах, за которое PHP должен разобрать входные данные GET и POST.*
- ***max_input_vars* —** *ограничение на количество входных переменных, которые могут быть переданы на сервер в одном запросе.*

Разрешим запуск php-fpm и перезапустим его:

```
systemctl enable php${PHP_VER}-fpm

systemctl restart php${PHP_VER}-fpm
```

NGINX + PHP

Для того, чтобы NGINX обрабатывал PHP, открываем конфигурационный файл:

```
vi /etc/nginx/sites-enabled/default
```

В секции **location** добавляем параметр **index**:

```
        ...
 index  index.php;
 ...
```

Внутри секции **server** добавим следующее:

```
        location ~ \.php$ {
         set $root_path /var/www/html;
         fastcgi_buffer_size 32k;
         fastcgi_buffers 4 32k;
         fastcgi_pass unix:/run/php/php8.2-fpm.sock;
         fastcgi_index index.php;
         fastcgi_param SCRIPT_FILENAME $root_path$fastcgi_script_name;
         include fastcgi_params;
         fastcgi_param DOCUMENT_ROOT $root_path;
 }
```

** где* **/var/www/html** *— корневой путь хранения скриптов;* **/run/php/php8.2-fpm.sock** *— путь до сокетного файла php-fpm (точное расположение файла можно посмотреть в конфигурационном файле* ***/etc/php/8.2/fpm/pool.d/www.conf****).*

Проверяем настройки nginx и перезагружаем его:

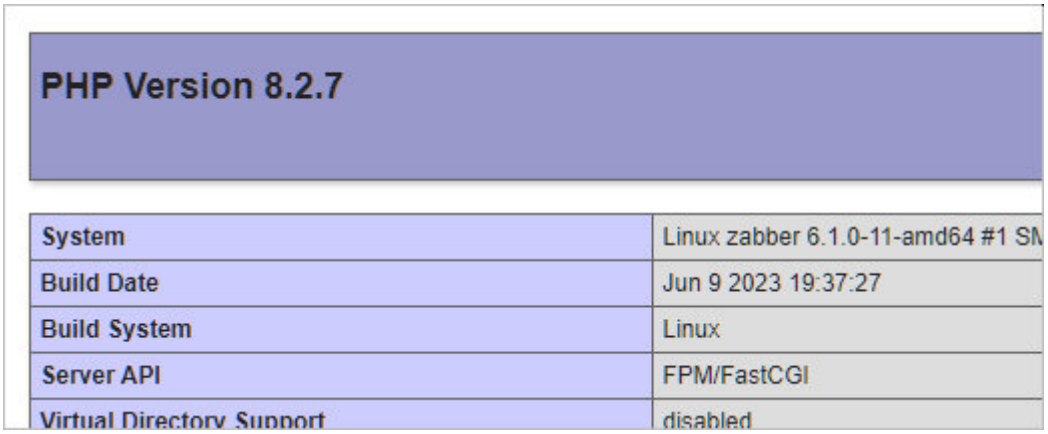
nginx -t && nginx -s reload

Создаем index.php со следующим содержимым:

vi /var/www/html/index.php

<?php phpinfo(); ?>

Открываем веб-браузер и переходим по ссылке http://<IP-адрес сервера>/ — теперь мы должны увидеть сводную информацию по PHP и его настройкам:



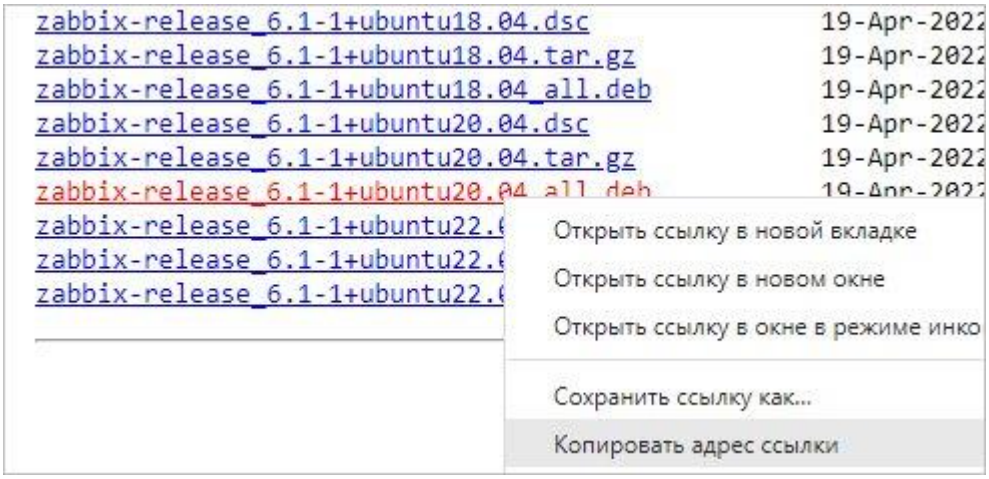
Веб-сервер готов для работы с Zabbix Web.

Установка и настройка сервера Zabbix

Переходим к установке самого Zabbix сервера.

Установка

Сначала установим репозиторий последней версии Zabbix. Для этого переходим на страницу https://repo.zabbix.com/zabbix/ и переходим в раздел с самой последней версией пакета - затем переходим в **ubuntu/pool/main/z/zabbix-release/** - копируем ссылку на последнюю версию релиза:



Обратите внимание, что необходимо скопировать ссылку на файл для своей версии Ubuntu — в нашем примере это 20.04. Посмотреть свою версию операционной системы можно командой:

cat /etc/os-release | grep VERSION_ID

Скачиваем файл репозитория командой:

wget https://repo.zabbix.com/zabbix/6.1/ubuntu/pool/main/z/zabbix-release/zabbix-release_6.1-1%2Bubuntu20.04_all.deb

Устанавливаем его:

dpkg -i zabbix-release_*.deb

Обновляем списки пакетов:

apt update

Устанавливаем сервер, вводя команду:

apt install zabbix-server-mysql zabbix-frontend-php zabbix-get zabbix-sql-scripts

Настройка базы данных

Входим в оболочку ввода sql-команд:

mysql -uroot -p

Создаем базу данных:

> CREATE DATABASE zabbix DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_bin;

** мы создали базу **zabbix**.*

Создаем пользователя для подключения и работы с созданной базой:

> GRANT ALL PRIVILEGES ON zabbix.* TO zabbix@localhost IDENTIFIED BY 'zabbixpassword';

** в данном примете мы создали пользователя **zabbix** с доступом к базе **zabbix** и паролем **zabbixpassword**.*

Выходим из sql-оболочки:

> quit

В составе zabbix идет готовая схема для СУБД MySQL/MariaDB или postgreSQL. В нашем случае, нам нужен MySQL.

Для применения схемы переходим в каталог:

cd /usr/share/zabbix-sql-scripts/mysql

В предыдущих версиях Zabbix путь до дампа базы был **/usr/share/doc/zabbix-sql-scripts/mysql**. Если мы не смогли найти нужный каталог с дампом, можно попробовать выполнить поимск командой:

find / -type f -iname server.sql.gz

Распаковываем архив с дампом базы:

gunzip server.sql.gz

Восстанавливаем базу их дампа:

mysql -u root -p zabbix < server.sql

** после ввода команды система запросит пароль. Необходимо ввести пароль, который мы задали после установки mariadb.*

Настройка zabbix

Открываем конфигурационный файл zabbix:

vi /etc/zabbix/zabbix_server.conf

Добавляем строку:

DBPassword=zabbixpassword

** мы настраиваем портал на подключение к базе с паролем **zabbixpassword**, который задали при создании базы для zabbix.*

И проверяем следующие строки:

...
DBName=zabbix
...
DBUser=zabbix
...

** имя базы и пользователя должны быть **zabbix** (как мы и создали в mariadb).*

Создаем каталог для инклудов конфигурационных файлов (по какой-то причине, он может быть не создан при установке):

mkdir /etc/zabbix/zabbix_server.conf.d

Запуск zabbix-server

Разрешаем автозапуск сервера мониторинга:

systemctl enable zabbix-server

После запускаем сам сервер zabbix:

systemctl start zabbix-server

Настройка nginx

При установке zabbix-web файлы портала копируются в каталог /usr/share/zabbix. Наш веб-сервер работает с каталогом /var/www/html.

Меняем это — открываем конфигурационный файл nginx:

vi /etc/nginx/sites-enabled/default

Редактируем параметры root и set $root_path:

```
...
root /usr/share/zabbix;
...
location ~ \.php$ {
    set $root_path /usr/share/zabbix;
    ...
}
...
```

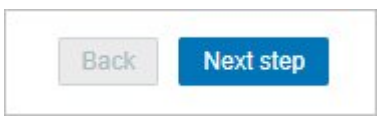
Перезапускаем nginx:

systemctl restart nginx

Установка портала для управления Zabbix

Открываем браузер и переходим по адресу http://<IP-адрес сервера>/ — откроется страница установки Zabbix Web.

Выбираем нужный язык установки и кликаем по **Next Step**:



Если в списке языков не окажется нужного нам, устанавливаем локаль:

dpkg-reconfigure locales

** например, для русского языка выбираем в открывшемся окне **ru_RU.UTF-8 UTF-8**.*

И перезапускаем обработчик PHP, в нашем случае:

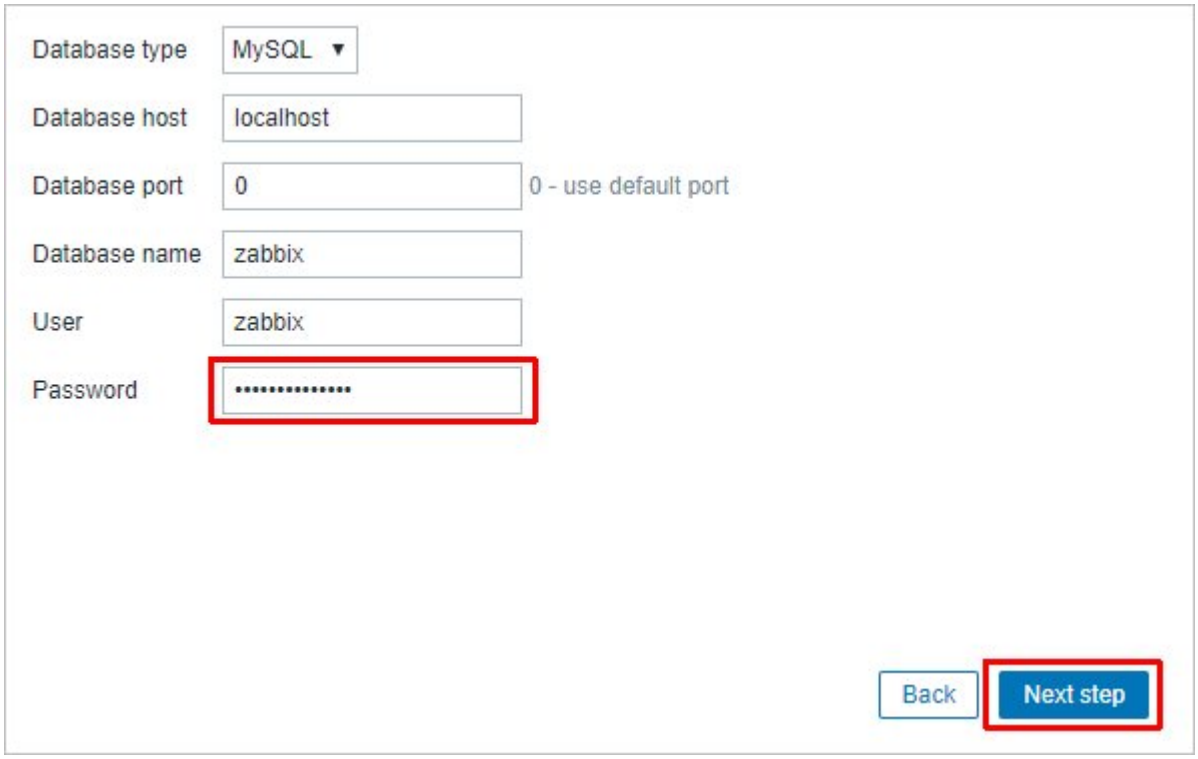
systemctl restart php${PHP_VER}-fpm

Перезагружаем страницу установки и выбираем нужный язык.

В следующем окне внимательно смотрим на результаты проверки нашего веб-сервера — справа мы должны увидеть все **OK**. Если это не так, проверяем настройки и исправляем предупреждения и ошибки, после перезапускаем страницу F5 для повторной проверки настроек.

Когда все результаты будут **OK**, кликаем по **Next Step**:

В следующем окне мы оставляем настройки подключения к базе как есть — дополнительно прописываем пароль, который задали при создании пользователя zabbix. После нажимаем **Next Step**:



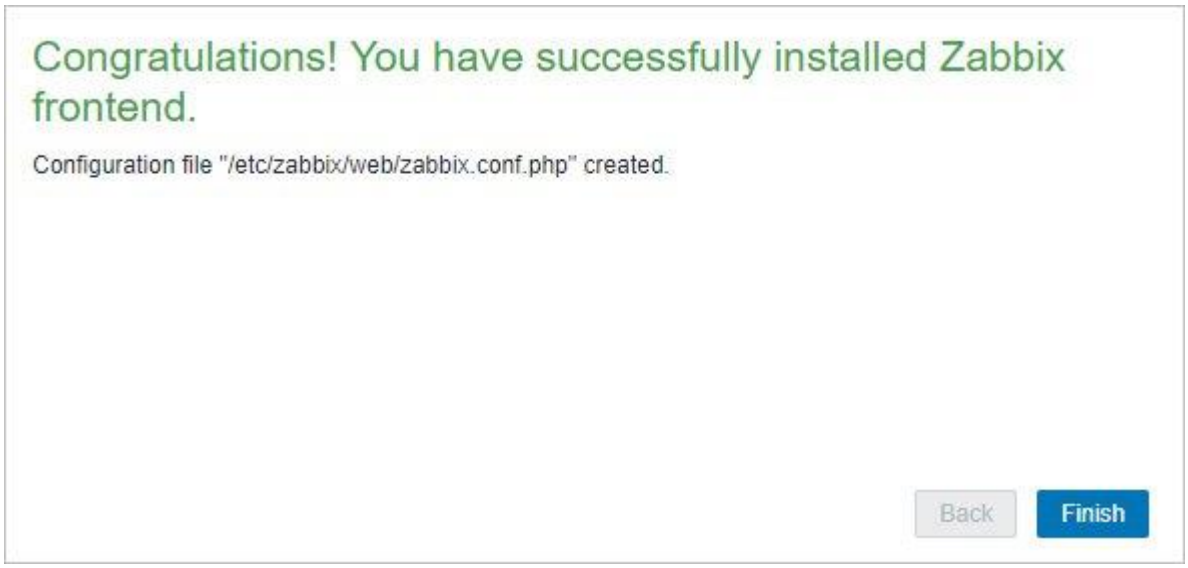
** в нашем случае, пароль был **zabbixpassword**;*

В следующем окне оставляем все как есть:

... и нажимаем **Next Step**.

В последнем окне мы проверяем настройки и кликаем **Next Step**.

Установка завершена — нажимаем **Finish**:



В открывшемся окне вводим логин **Admin** и пароль **zabbix** (по умолчанию) — откроется окно со сводной информацией по мониторингу:

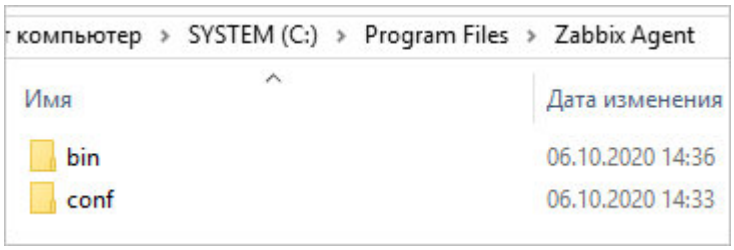
Практическая работа № 12 Установка и настройка агента Zabbix на Windows

Задание:

Установите агент Zabbix на Windows.

Данный метод требует дополнительных манипуляций, но с его помощью можно автоматизировать процесс. Мы рассмотрим только установку.

Распаковываем содержимое скачанного архива в каталог, где будут находиться файлы программы, например в **C:\Program Files\Zabbix Agent**. В итоге, у нас получится:



Открываем WordPad от администратора и в нем открываем конфигурационный файл **conf/zabbix_agentd.conf** и правим опцию для сервера zabbix — находим строку:

Server=127.0.0.1

... и меняем ее на:

Server=zabbix-server.dmosk.local

* где ***zabbix-server.dmosk.local*** *— имя сервера Zabbix.*

Теперь открываем командную строку от администратора и выполняем команду для установки агента:

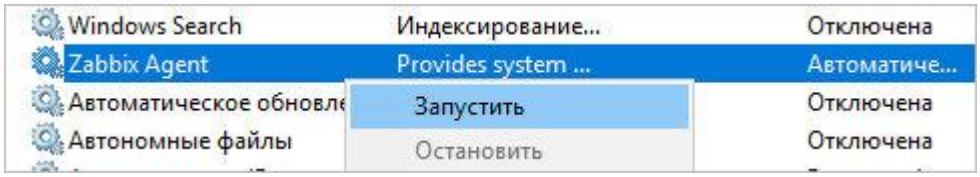
"C:\Program Files\Zabbix Agent\bin\zabbix_agentd.exe" --config "C:\Program Files\Zabbix Agent\conf\zabbix_agentd.conf" --install

** где **C:\Program Files\Zabbix Agent** — папка, куда мы распаковали архив.*

Мы должны увидеть что-то на подобие:

zabbix_agentd.exe [468]: service [Zabbix Agent] installed successfully
zabbix_agentd.exe [468]: event source [Zabbix Agent] installed successfully

Теперь открываем службы Windows и находим «Zabbix Agent» — кликаем по ней правой кнопкой мыши и выбираем **Запустить**:



Установка завершена.

Настройка брандмауэра

Если в системе работает брандмауэр Windows, необходимо разрешить порт 10050 или приложение zabbix agent.

Для этого переходим в **Панель управления - Система и безопасность - Брандмауэр Windows** (или вводим команду **control /name Microsoft.WindowsFirewall**). Кликаем по ссылке **Разрешение взаимодействия с приложением или компонентов в брандмауэре Windows** - добавляем наше приложение **zabbix_agentd** в исключение.

Это же действие можно выполнить в powershell — запускаем ее от администратора и вводим команду:

New-NetFirewallRule -DisplayName "Разрешить приложение Zabbix Agent" -Direction Inbound -Action Allow -EdgeTraversalPolicy Allow -Program "C:\Program Files\Zabbix Agent\bin\zabbix_agentd.exe"

Или можно добавить в исключение порт. Это можно сделать также из командной строки poweshell:

New-NetFirewallRule -DisplayName "Разрешить порт 10050 для Zabbix" -Direction Inbound -Action Allow -EdgeTraversalPolicy Allow -Protocol TCP -LocalPort 10050

Проверка работы

Чтобы убедиться в работоспособности агента, можно зайти на сервер zabbix и выполнить подключение по telnet, например, командой:

telnet 192.168.1.15 10050

** где **192.168.1.15** — IP-адрес компьютера с установленным Zabbix.*

Connected to nr-fs-06.
Escape character is '^]'.

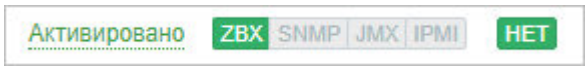
.. и через небольшой интервал времени:

Connection closed by foreign host.

Но если агент не запущен или не работает, выпадет ошибка:

telnet: connect to address ...: Connection refused

В панели сервера в узлах сети при корректной установке и настройке, мы также должны увидеть доступность компьютера по агенту:



Практическая работа № 13 Установка и настройка агента Zabbix на Linux

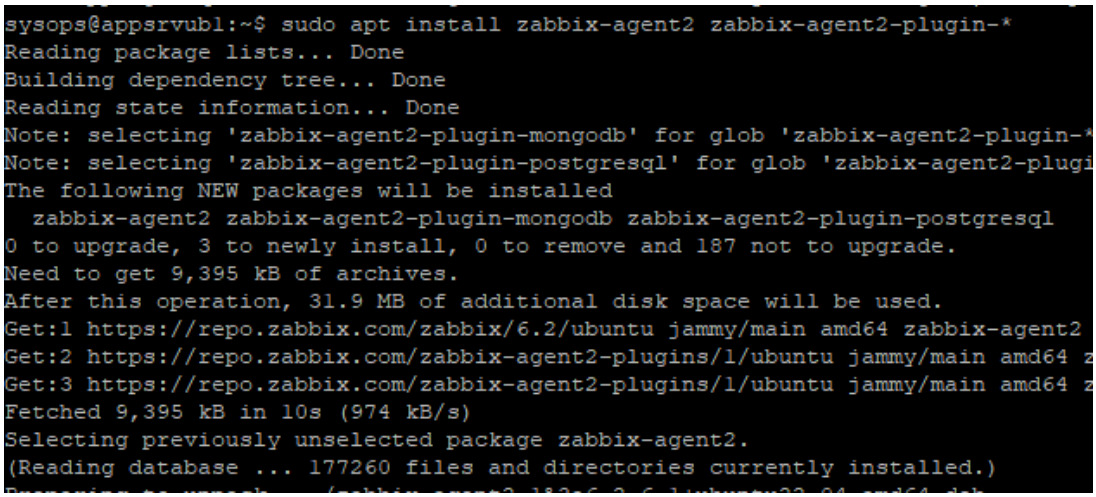
Задание:

Установить репозиторий Zabbix и обновить кеш менеджера пакетов:

$ wget https://repo.zabbix.com/zabbix/6.2/ubuntu/pool/main/z/zabbix-release/zabbix-release_6.2-4%2Bubuntu22.04_all.deb
$ sudo dpkg -i zabbix-release_6.2-4+ubuntu22.04_all.deb
$ sudo apt update

Теперь можно установить Zabbix agent2:

$ sudo apt install zabbix-agent2

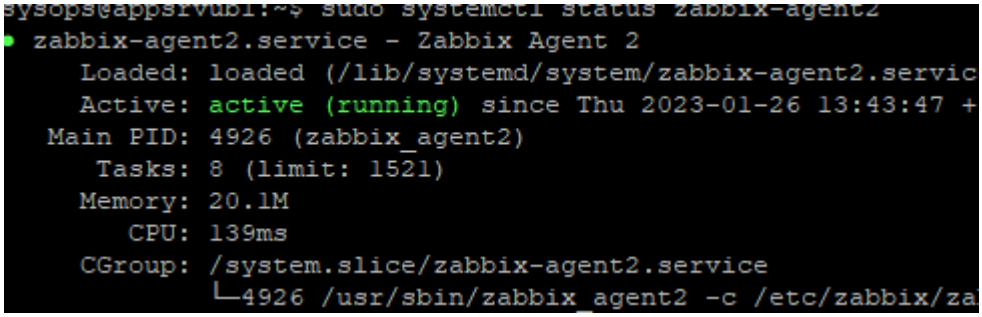


Запустите службу агента Zabbix и добавьте его в автозагрузку.

$ sudo systemctl restart zabbix-agent2
$ sudo systemctl enable zabbix-agent2

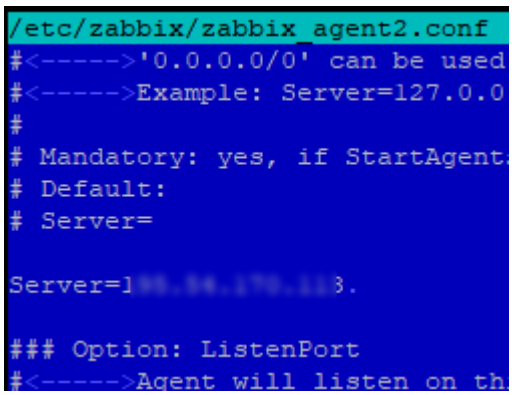
Проверьте, что агент Zabbix запущен:

$ sudo systemctl status zabbix-agent2



Теперь нужно отредактировать конфигурационной файл агента:

$ sudo mcedit /etc/zabbix/zabbix_agent2.conf



Server=ИМЯ_или_IP_Zabbix_Server

ServerActive=ИМЯ_или_IP_Zabbix_Server

Hostname= appsrvub1

appsrvub1 – это вашего имя узла, который мы добавим далее на сервер мониторинга Zabbix.

В Zabbix доступны два режима проверки агентов:

- Пассивный режим – данные запрашиваются сервером Zabbix
- Активный режим – агент сам отправляет данные на сервер Zabiix (в моем случае сервер с агентом находится за NAT, поэтому для него я буду использовать активный режим)

Можно полностью отключить пассивные проверки:

StartAgents=0

Перезапустите агент Zabbix.

$ sudo systemctl restart zabbix-agent2

Проверьте, что агент успешно запустился:

$ cat /var/log/zabbix/zabbix_agent2.log

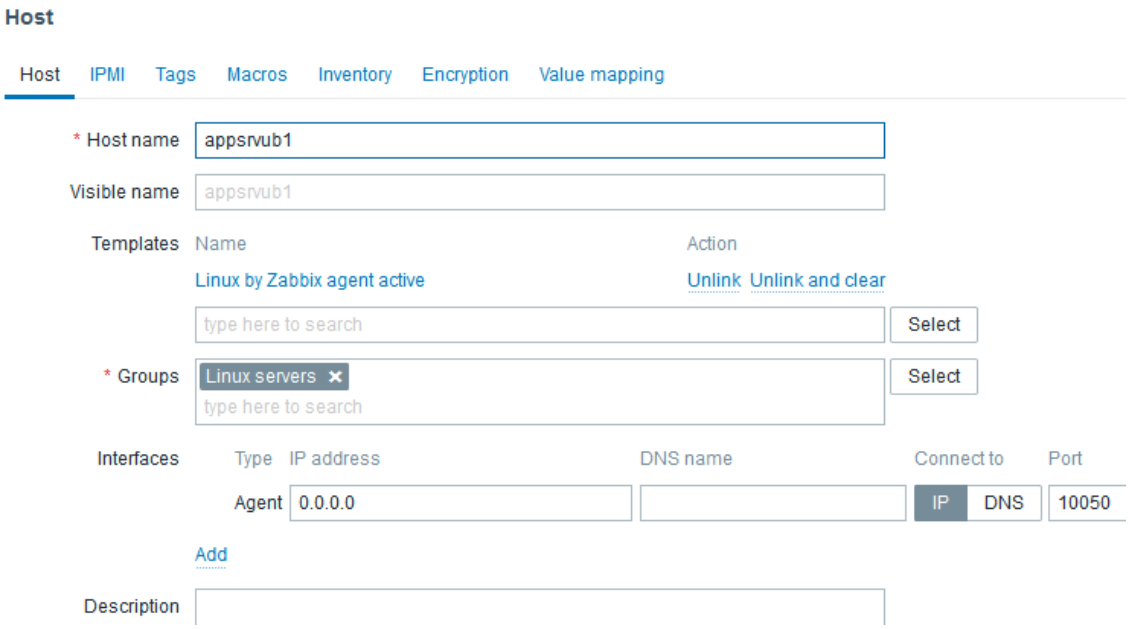
Агент подключается к серверу на порт TCP/10051. Поэтому порт не должен блокироваться файерволами. Вы можете проверить доступность Zabbix Server с хоста с помощью netcat:

$ nc -zv zabbixsrv1 10051

Теперь нужно добавить новый агент через веб-интерфейс Zabbix.

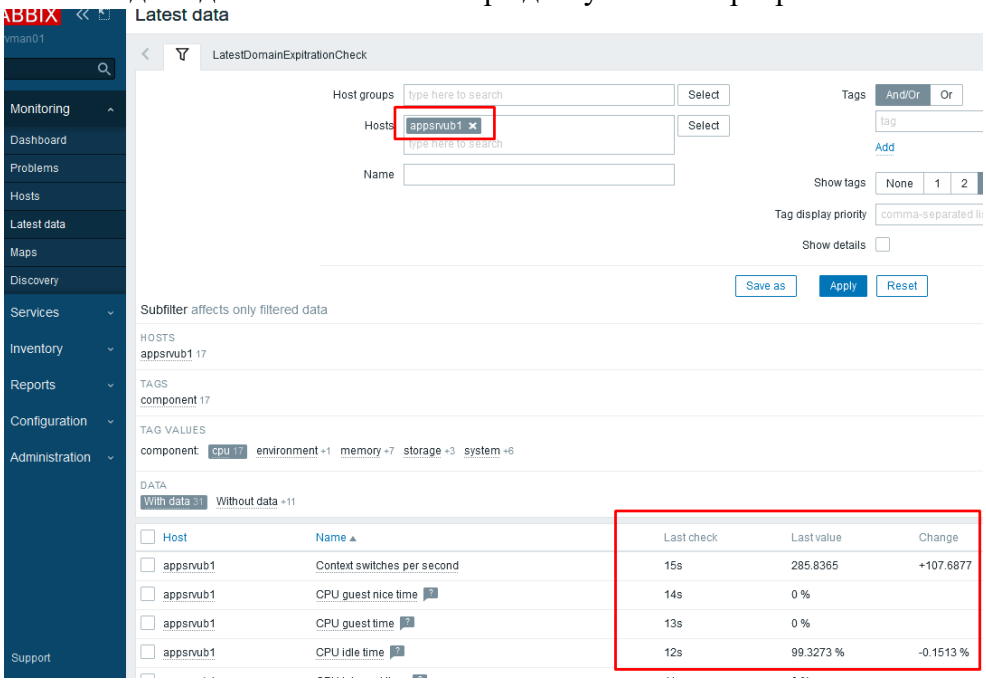
1. Перейдите в раздел **Configuration -> Hosts** и нажмите **Create host;**
2. Укажите имя хоста (должно соответствовать значению в конфигурационном файле агента);
3. Назначьте шаблон (в моем случае это Linux by Zabbix agent active) и группу хостов;

4. Т.к. мой агент Zabbix находится за NAT, для интерфейса агента не нужно указывать IP адрес. Просто укажите
0.0.0.0.



5. Сохраните изменения.

6. Теперь проверьте, что сервер Zabbix получает данные с агента. Перейдите в **Monitoring** -> **Latest Data,** выберите в поле **Hosts** нужный вам хост и нажмите Apply. Как вы видите данные с агента теперь доступны на сервере Zabbix.



По умолчанию агент Zabbix передает данные на сервер в открытом виде. Если данные от агентов поступают на сервере через публичную сеть Интернет, нужно настроить шифрование с помощью pre-shared ключей (PSK).

Подключитесь к агенту и сгенерируйте 256 битный PSK ключ с помощью openssl:

openssl rand -hex 32 > /etc/zabbix/zabbix_agent.psk
chown zabbix:zabbix /etc/zabbix/zabbix_agent.psk
chmod 400 /etc/zabbix/zabbix_agent.psk

Теперь добавьте информацию о PSK ключе шифрования в конфигурационный файл:

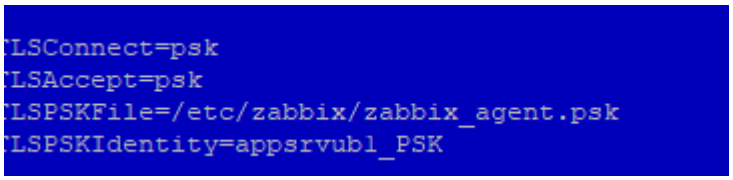
mcedit /etc/zabbix/zabbix_agent2.conf

TLSConnect=psk

TLSAccept=psk

TLSPSKFile=/etc/zabbix/zabbix_agent.psk

TLSPSKIdentity=appsrvub1_PSK



Сохраните файл и перезапустите агент Zabbix:

$ sudo systemctl restart zabbix-agent2

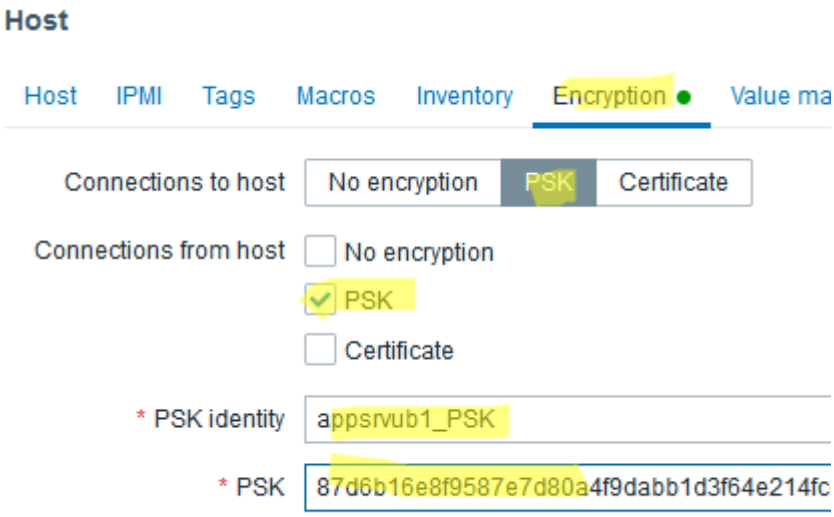
Скопируйте значение PSK ключа:

$ cat /etc/zabbix/zabbix_agent.psk

Теперь нужно указать ваш PSK ключ в настройках хоста на сервере Zabbix.

1. Откройте настройки хоста и перейдите на вкладку **Encryption**;
2. Включите опцию **PSK**;
3. Вставьте значение TLSPSKIdentity из конфигурационного файла агента в **PSK Identity**;
4. Вставьте ваш PSK ключ в следующее поле;

5. Нажмите кнопку



Update.

6. Если вы все настроили правильно, в информации о хосте в Zabbix будет указано, что агент использует соединение с PSK шифрованием.

Практическая работа № 14 Установка и настройка прокси сервера Zabbix

Задание:

Подключите репозиторий zabbix:

$ sudo rpm -Uvh https://repo.zabbix.com/zabbix/5.4/rhel/8/x86_64/zabbix-release-5.4-1.el8.noarch.rpm
$ sudo dnf clean all

Установите пакеты:

$ sudo dnf install -y zabbix-proxy-sqlite3 zabbix-agent policycoreutils-python-utils nano

Создайте каталог для базы данных SQLite:

$ sudo mkdir /var/lib/zabbix/
$ sudo chown -R zabbix. /var/lib/zabbix/

Отредактируйте конфигурационный файл:

$sudo nano /etc/zabbix/zabbix_proxy.conf

Server=10.1.15.10 #адрес Zabbix Server

Hostname=sp-zbprx1 # имя Zabbix Proxy

LogFile=/var/log/zabbix/zabbix_proxy.log

LogFileSize=1024

PidFile=/var/run/zabbix/zabbix_proxy.pid

SocketDir=/var/run/zabbix

DBName=/var/lib/zabbix/zabbix_proxy

DBUser=zabbix

SNMPTrapperFile=/var/log/snmptrap/snmptrap.log

Timeout=4

LogSlowQueries=3000

StatsAllowedIP=127.0.0.1

Запустите zabbix-proxy (обязательно, т.к. это создаст правила для selinux):

$ sudo systemctl start zabbix-proxy

Если на хосте включен Selinux, нужно добавить разрешающие правила:

$ cd /tmp
$ sudo grep zabbix_proxy /var/log/audit/audit.log | grep denied | audit2allow -m zabbix_proxy > zabbix_proxy.te
$ sudo grep zabbix_proxy /var/log/audit/audit.log | grep denied | audit2allow -M zabbix_proxy
$ sudo semodule -i zabbix_proxy.pp

Включите службу zabbix-proxy:

$ sudo systemctl start zabbix-proxy
$ sudo systemctl enable zabbix-proxy
$ sudo systemctl status zabbix-proxy

Логи Zabbix прокси можно вывести так:

$ sudo tail -f /var/log/zabbix/zabbix_proxy.log

Для защиты подключений между прокси и сервером Zabbix с помощью шифрования, мы дополнительно установим pre-shared key (PSK).

Сгенерируйте PSK ключ:

$ openssl rand -hex 32 | sudo tee /var/lib/zabbix/proxy.psk
$ sudo chown zabbix. /var/lib/zabbix/proxy.psk

Добавьте информацию о PSK в конфигурационный файл zabbix proxy:

$ sudo nano /etc/zabbix/zabbix_proxy.conf

####### TLS-RELATED PARAMETERS #######

TLSConnect=psk

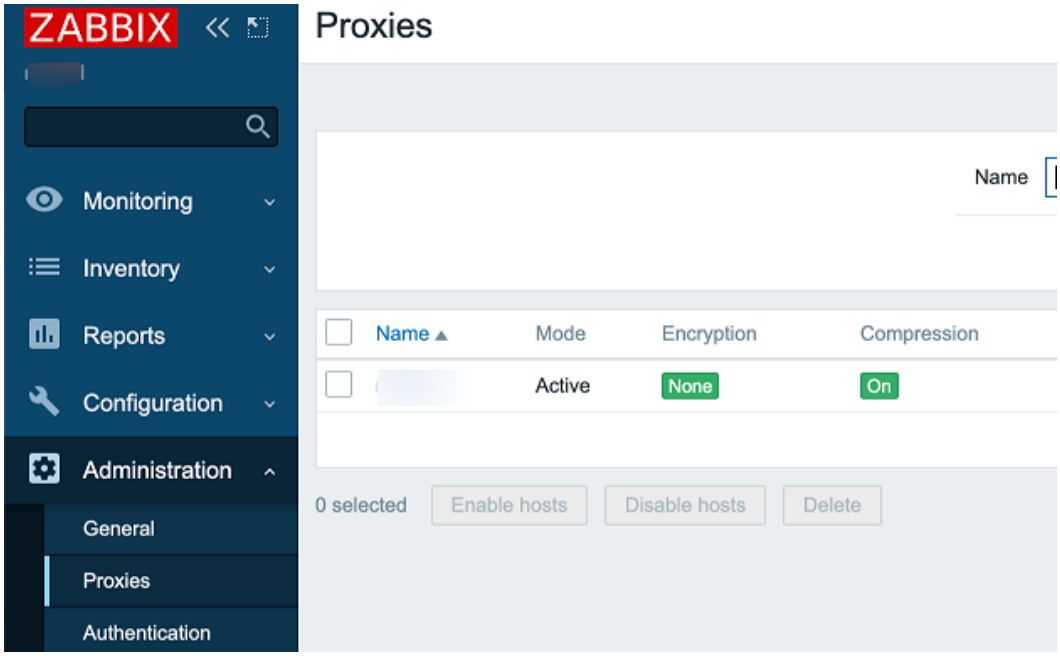
TLSPSKIdentity=sp-zbprx1

TLSPSKFile=/var/lib/zabbix/proxy.psk

Перезапустите сервис:

$ sudo systemctl restart zabbix-proxy

Зарегистрируйте ваш прокси в веб-интерфейсе Zabbix Server и укажите ваш PSK ключ.



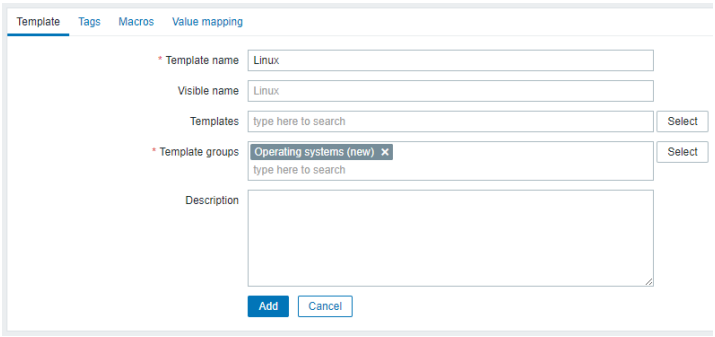
Практическая работа № 15 Создание шаблонов в Zabbix

Задание:

Для создания шаблона, выполните следующее:

- Перейдите в *Настройка → Шаблоны*
- Нажмите на *Создать шаблон*
- Измените атрибуты шаблона

Вкладка **Шаблон** содержит общие атрибуты шаблона.

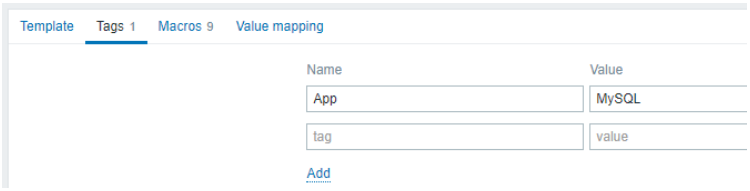


Все обязательные поля ввода отмечены красной звёздочкой.

Атрибуты шаблонов:

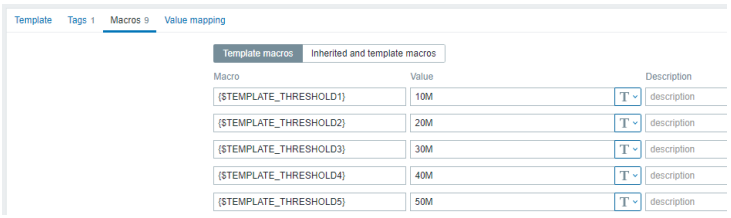
Параметр	Описание
Имя шаблона	Уникальное имя шаблона. Разрешены буквенно-цифровые символы, пробелы, точки, тире и символы подчеркивания. Однако пробелы в начале и конце имени запрещены.
Видимое имя	Если вы укажите это имя, то только оно будет видимо в списках, картах и прочем.
Шаблоны	Присоедините один или более "вложенных" шаблонов к этому шаблону. Все объекты (элементы данных, триггеры, графики и т.д.) будут унаследованы с присоединенных шаблонов. Для присоединения нового шаблона, начните печатать в поле *Присоединить новые шаблоны*. Появится список совпадающих шаблонов; прокрутите список и выберите. Кроме того, вы можете нажать *Выбрать* рядом с полем и выбрать шаблоны из списка во всплывающем окне. Шаблоны, выбранные в поле *Привязать новые шаблоны*, будут связаны с шаблоном при сохранении или обновлении формы конфигурации шаблона. Для отсоединения шаблона, воспользуйтесь одной из двух опций в блоке *Присоединенных шаблонов*: *Отсоединить* - отсоединить шаблон, но оставить его элементы данных, триггеры и графики *Отсоединить и очистить* - отсоединить шаблон и удалить все его элементы данных, триггеры и графики
Группы	Группы узлов сети / шаблонов к которым принадлежит этот шаблон.
Описание	Введите описание шаблона.

Вкладка **Теги** позволяет вам задать теги на уровне шаблона. Все проблемы узлов сети, присоединенных к этому шаблону, будут отмечены тегами с их значениями, введенными здесь.

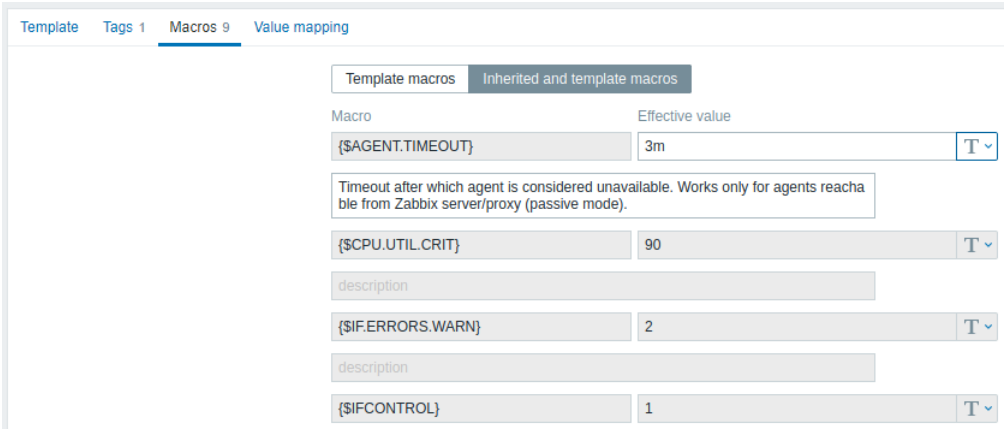


В тегах поддерживаются пользовательские макросы, все макросы {INVENTORY.*}, а также {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} и {HOST.ID}.

Вкладка **Макросы** позволяет вам задать пользовательские макросы уровня шаблона в виде пар имени и значения. Обратите внимание, что значения макросов могут храниться в виде обычного текста, скрытого текста или секрета Хранилища. Также поддерживается добавление описания.



Вы также можете здесь просмотреть макросы из присоединенных шаблонов и глобальные макросы, если вы выберите опцию *Унаследованные и макросы из шаблонов*. Это то место, где отображаются все определенные пользовательские макросы для этого шаблона со своими раскрытыми значениями, а также информация о том откуда эти макросы.



Для удобства имеются ссылки на настройку соответствующих шаблонов и глобальных макросов. Также имеется возможность изменить макрос уровня шаблона/глобальный на уровне этого шаблона, фактически создав копию этого макроса у шаблона.

Вкладка **Преобразования значений** позволяет настроить удобные для человека представления данных элемента данных в преобразованиях значений.

Кнопки:

Add	Добавление шаблона. Добавленный шаблон должен появиться в списке.

Update	Обновление свойств существующего шаблона.
Clone	Создание другого шаблона основанного на свойствах текущего шаблона, включая все объекты (элементы данных, триггеры и т.п.) унаследованные от присоединенных шаблонов.
Full clone	Создание другого шаблона основанного на свойствах текущего шаблона, включая все объекты (элементы данных, триггеры и т.п.) как унаследованные от присоединенных шаблонов, так и напрямую присоединенные к текущему шаблону.
Delete	Удаление шаблона; объекты из шаблона (элементы данных, триггеры и прочее) останутся присоединенными к узлам сети.
Delete and clea	Удаление шаблона и всех его объектов из присоединенных узлов сети.
Cancel	Отмена изменения свойств шаблона.

Когда шаблон создан, самое время добавить в него какие-нибудь объекты.

В шаблон необходимо сначала добавить элементы данных. Триггеры и графики нельзя добавлять без наличия соответствующих элементов данных.

Добавление элементов данных, триггеров, графиков

Для добавления элементов данных в шаблон, сделайте следующее:

- Перейдите в *Настройка* → *Узлы сети* (или *Шаблоны*)
- Нажмите на *Элементы данных* в строке с требуемым узлом сети / шаблоном
- Отметьте элементы данных которые вы хотите добавить в шаблон
- Выберите *Копировать* ниже списка элементов данных
- Выберите шаблон (или группу шаблонов) в который необходимо скопировать элементы данных и нажмите на *Копировать*

Все выбранные элементы данных должны будут скопироваться в шаблон.

Добавление триггеров и графиков осуществляется похожим образом (из списка триггеров и графиков соответственно), опять же, имейте ввиду, что их можно добавить только после того, как сначала будут добавлены требуемые элементы данных.

Добавление панелей

Для добавления панелей в шаблон из *Настройка* → *Шаблоны*, сделайте следующее:

- Нажмите на *Панели* в строке с шаблоном

- Настройте панель следуя обычному способу настройки панелей

Виджеты, которые можно добавить к панелям шаблонов: простые графики, прототипы графиков, часы, простой текст, URL.

Для получения информации касательно доступа к панелям узлов сети, которые созданы с панелей шаблонов, смотрите раздел панель узлов сети.

Настройка правил низкоуровневого обнаружения

Смотрите раздел по низкоуровневому обнаружению в этом руководстве.

Добавление веб-сценариев

Чтобы добавить веб-сценарии в шаблон в *Настройка* → *Шаблоны*, сделайте следующее:

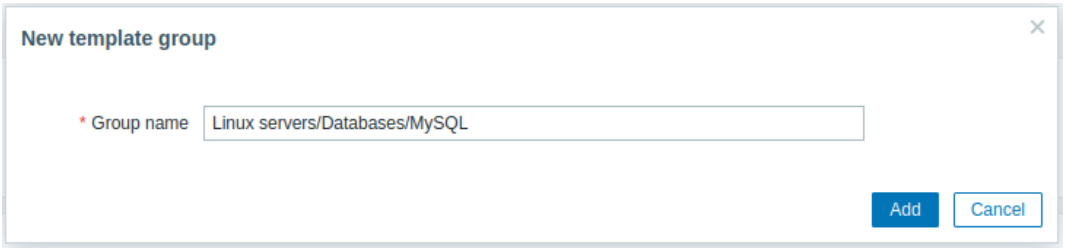
- Нажмите на *Веб* в строке с шаблоном
- Настройте веб-сценарий следуя обычному способу настройки веб-сценариев

Creating a template group

Only Super Admin users can create template groups.

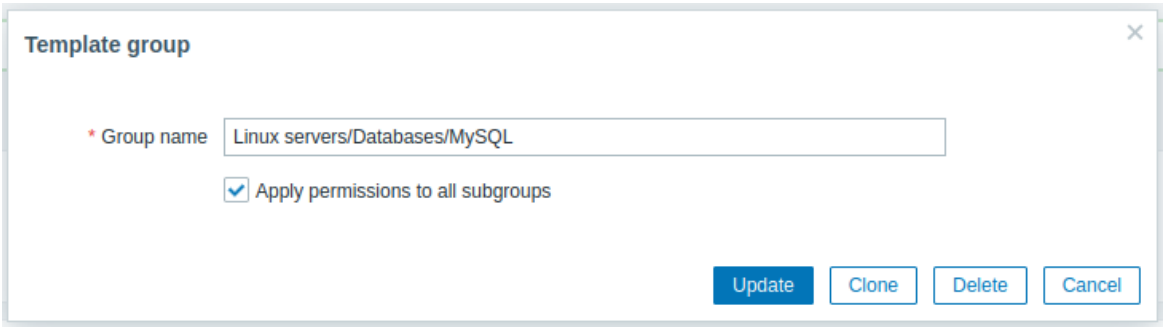
To create a template group in Zabbix frontend, do the following:

- Go to: *Configuration* → *Template groups*
- Click on *Create template group* in the upper right corner of the screen
- Enter the group name in the form



To create a nested template group, use the '/' forward slash separator, for example Linux servers/Databases/MySQL. You can create this group even if none of the two parent template groups (Linux servers/Databases/) exist. In this case creating these parent template groups is up to the user; they will not be created automatically.
Leading and trailing slashes, several slashes in a row are not allowed. Escaping of '/' is not supported.

Once the group is created, you can click on the group name in the list to edit group name, clone the group or set additional option:



Практическая работа № 16 Настройка оповещений Zabbix

Задание:

Для того чтобы Zabbix мог отправлять сообщения по электронной почте, необходимо сделать следующее:

- настроить способы оповещений Media Types;
- назначить Media с типом Email пользователю Zabbix;
- добавить действие при срабатывании триггера Trigger Action

Расскажем об этом подробнее.

Настройка способов оповещений Media Types

Если в Web-интерфейсе Zabbix выбрать **Media Types** из меню **Administration**, вы увидите многочисленные способы оповещений, доступные для настройки в Zabbix. Часть из них показана на рис. 1.

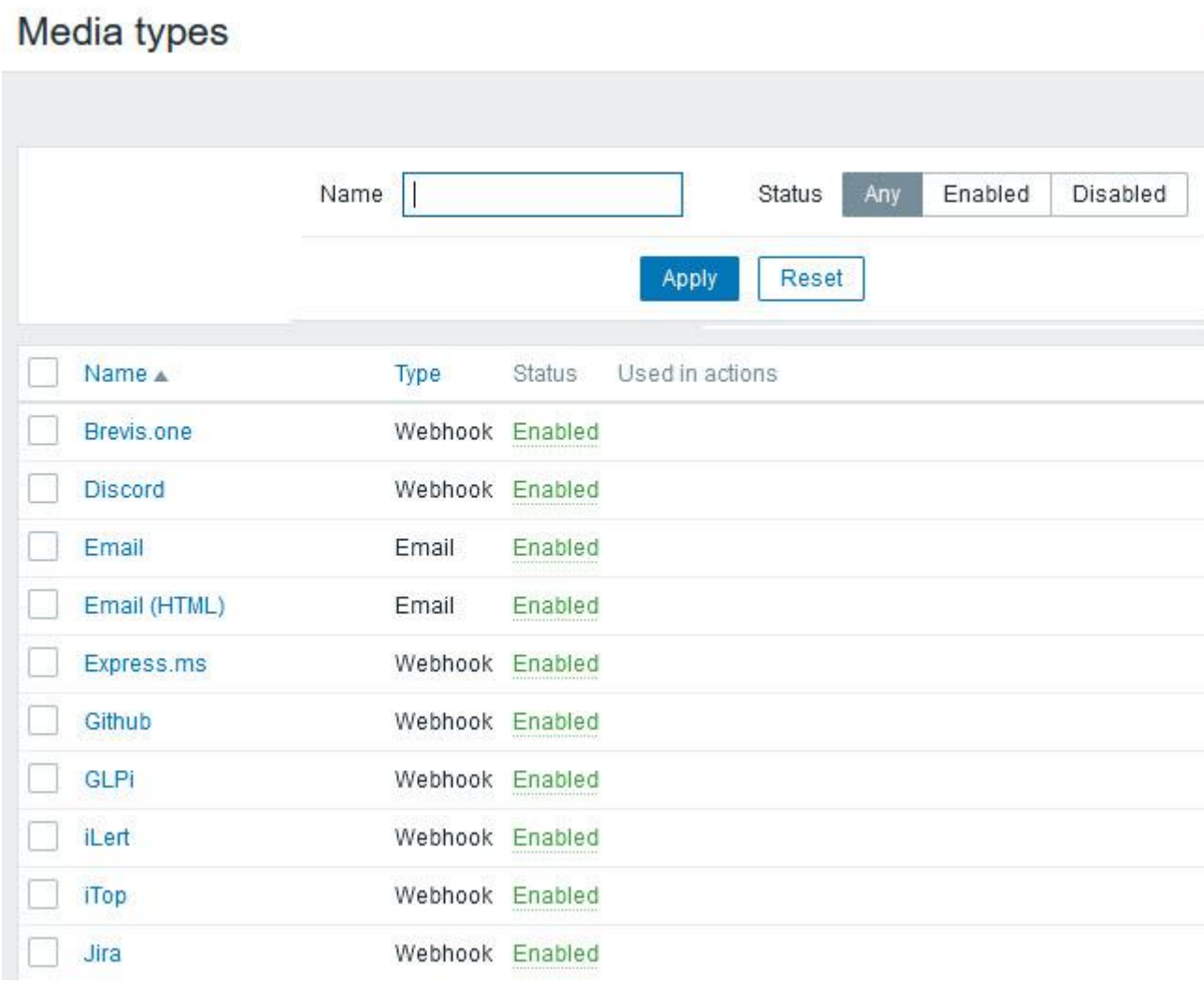


Рис. 1. Способы оповещений в Zabbix

Мы будем использовать готовые способы оповещений **Email** и **Telegram**, а также создадим собственные — **p1sms.ru** (для отправки SMS) и **Zvonobot** (для голосовых сообщений по телефону).

Если вы находитесь там, где есть интернет, то сможете получать сообщения через Email и Telegram. Но бывает и так, что доступны только SMS и голосовые звонки. Чтобы не пропустить важные сообщения от Zabbix, пригодятся все эти способы.

Чтобы настроить отправку электронной почты, щелкните на странице **Media Types** строку **Email**. Далее вам нужно будет заполнить форму, указав в ней параметры исходящего почтового сервера (рис. 2).

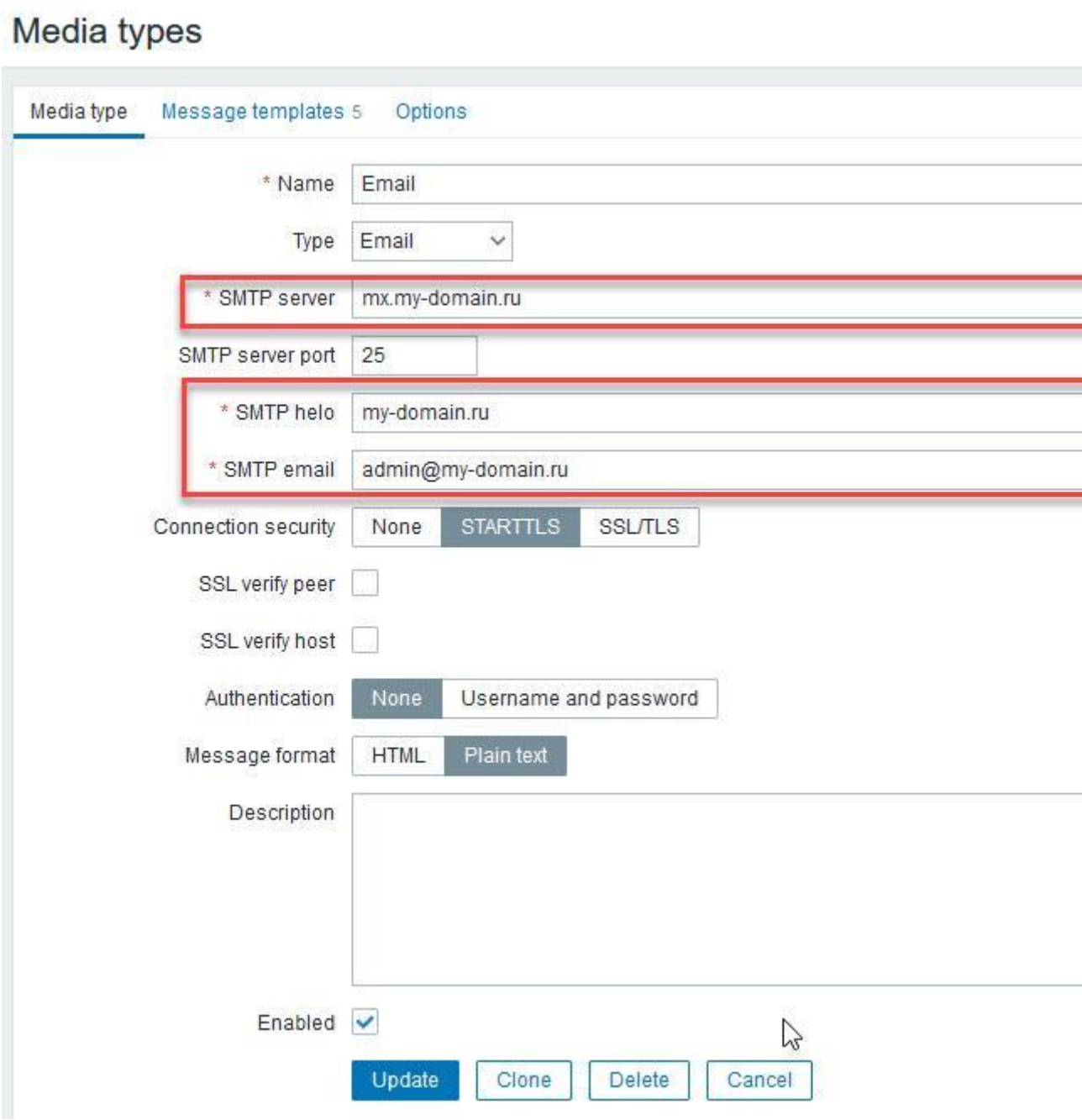


Рис. 2. Настройка способа оповещения Email (домен указан только для примера)

Мы показали случай, когда используется собственный почтовый сервер, доступный на порту 25 с использованием STARTTLS. Сообщения будут отправлены на адрес admin@my-domain.ru (имя домена приведено только для примера).

В ответственных случаях безопаснее использовать собственный почтовый сервер, который находится под вашим контролем. При необходимости можно настроить отправку почты и через публичные почтовые сервисы, такие как Google Mail или Яндекс Почту. Помимо официальной документации в интернете есть инструкции по настройке, например, здесь.

Настроив способ оповещения Email, выполните проверку с помощью ссылки **Test**, расположенной справа напротив Email в списке способов оповещений, показанном на рис. 1.

Щелкните эту ссылку и заполните форму, указав в ней адрес получателя **Send to**, тему сообщения **Subject** и текст сообщения **Message**. Затем щелкните кнопку **Test** (рис. 3).

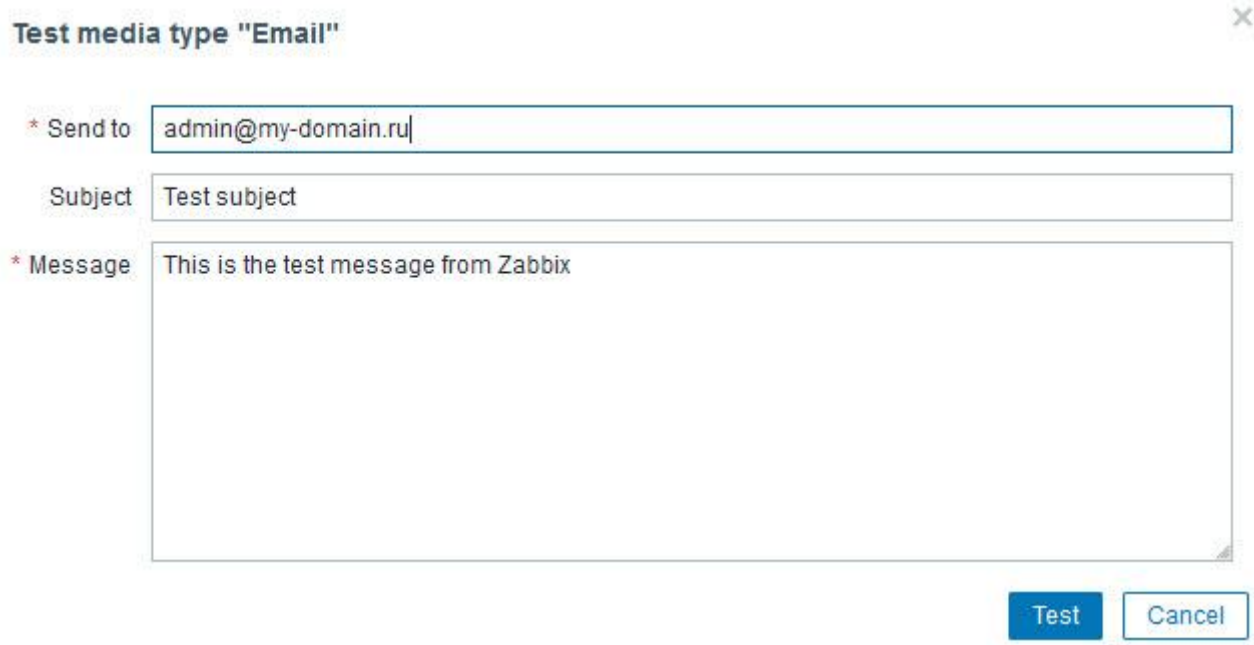


Рис. 3. Отправка тестового сообщения через способ оповещения Email

Если настройки почтового сервера указаны правильно, вы получите сообщение на указанный адрес электронной почты.

На вкладке **Message Templates** можно изменить стандартные шаблоны сообщений, отправляемых по электронной почте.

Назначение Media с типом Email пользователю Zabbix

Чтобы способ оповещения заработал, его нужно назначить пользователю Zabbix, например, Admin.

Выберите в меню **Administration** раздел **Users**, щелкните имя пользователя **Admin**, а затем откройте вкладку **Media**. На рис. 4 мы показали ситуацию, когда для пользователя было добавлено несколько способов оповещения, в том числе и способ **Email**.

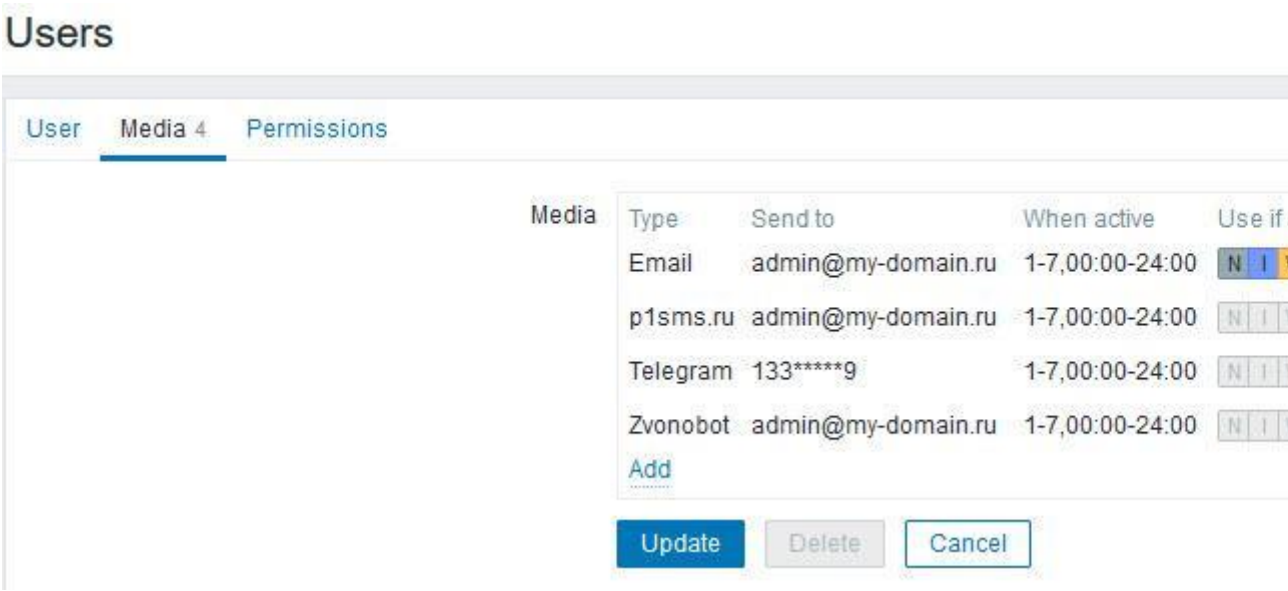


Рис. 4. Способы оповещения для пользователя Admin

Для добавления способа оповещения Email щелкните ссылку **Add**. В списке **Type** выберите строку **Email**.

В поле **Send to** укажите адрес электронной почты, по которому должно отправляться оповещение. С помощью поля **When active** можно ограничить дни и часы отправки сообщений.

Набор флажков **Use if severity** позволяет задать уровни серьезности триггеров, при срабатывании которых необходимо отправить сообщение по указанному адресу (рис. 5).

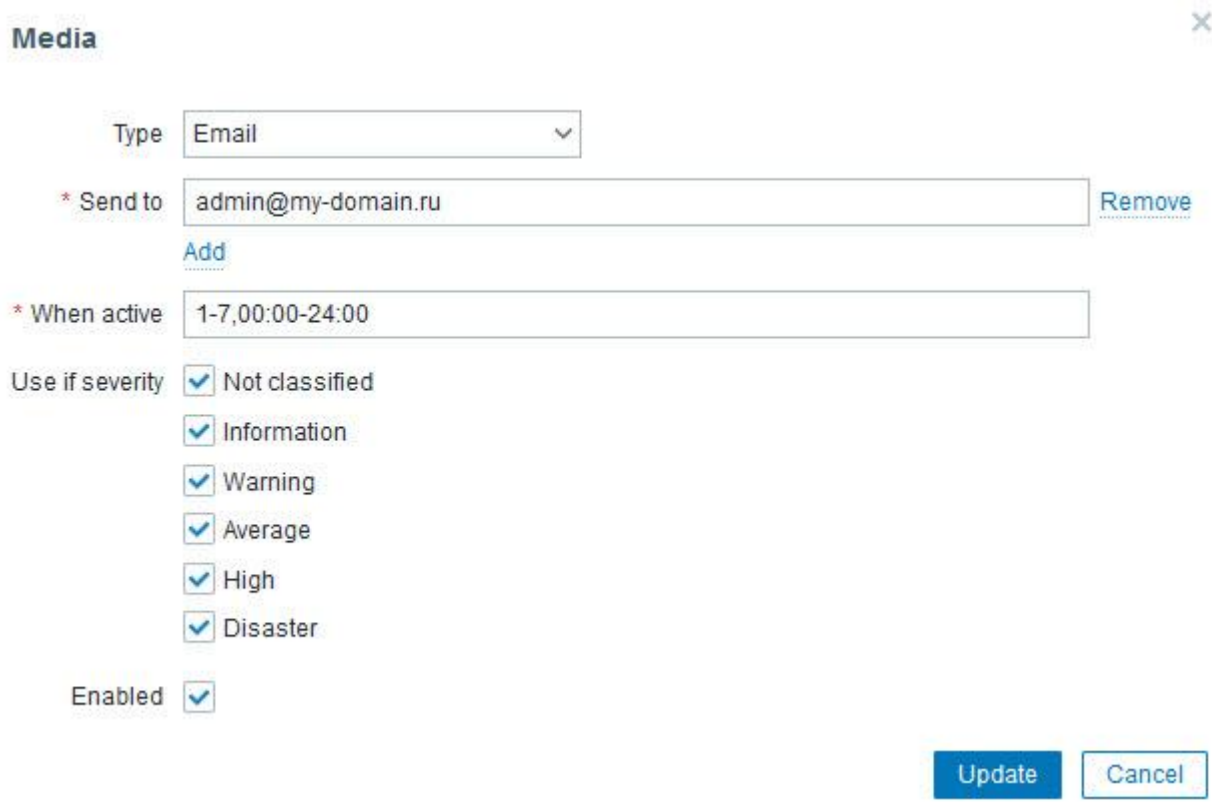


Рис. 5. Добавление способа оповещения

В нашем случае сообщения отправляются по электронной почте круглосуточно, причем при срабатывании триггера с любым уровнем серьезности.

Если нужно отредактировать или удалить способ оповещения, воспользуйтесь ссылкой **Edit** или **Remove**, соответственно (рис. 4).

Добавление действий при срабатывании триггера Trigger Actions

Итак, мы настроили и проверили способ оповещения Email, а также добавили его для пользователя Admin.

Теперь нужно добавить действие при срабатывании триггера. Для этого в меню **Configuration** выберите меню **Action**, а из него — строку **Trigger Actions**.

На рис. 6 показан уже сформированный список действий для различных способов оповещений.

Рис. 6. Список действий для различных способов оповещений

Здесь действие **zabbix03 action**, заключающееся в отправке сообщения по электронной почте, выполняется для всех уровней серьезности, кроме Disaster. Действие **SMS Notify** выполняется при срабатывании триггеров с уровнем серьезности High и Disaster. И, наконец, действие **DISASTER_ALERT_Phone_CALL** определено для звонков по телефону при срабатывании триггеров уровня Disaster.

Чтобы создать новое действие, воспользуйтесь кнопкой **Create action** на странице **Trigger Actions**. Вам будет предложено добавить действие на вкладке **Action** (рис. 7).

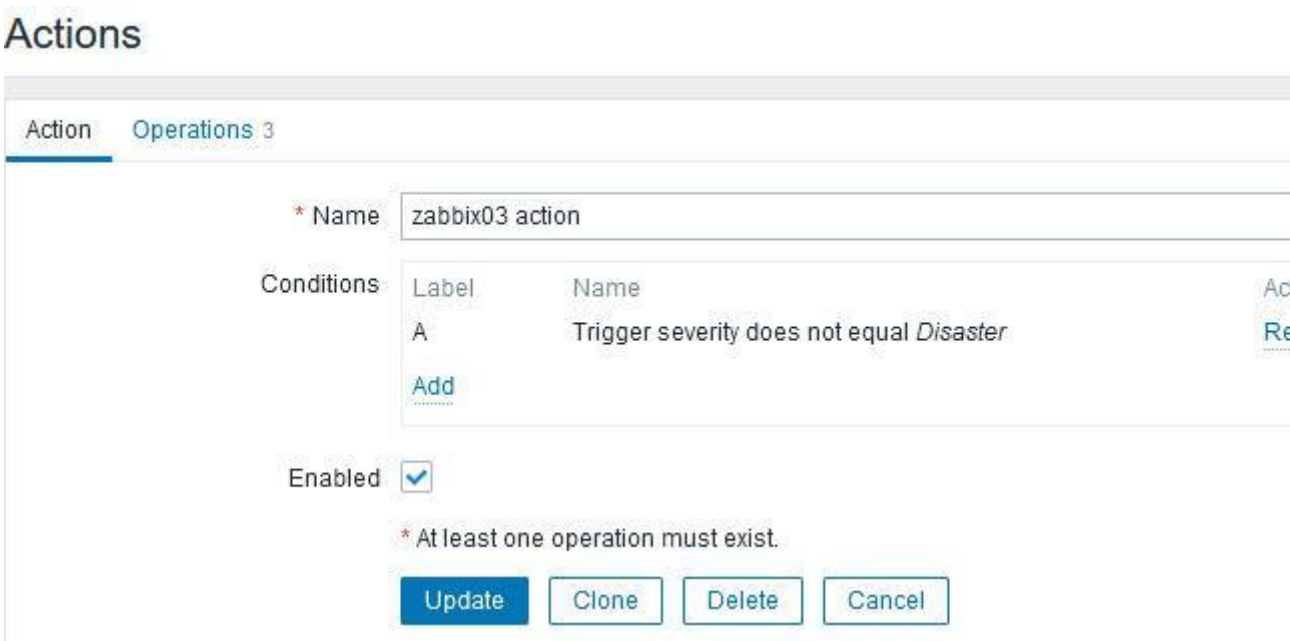


Рис. 7. Редактирование добавленного действия

Задайте имя действия, а затем щелкните ссылку **Add**, чтобы добавить операцию. При добавлении выберите в списке **Type** уровень серьезности триггера **Trigger severity**. Чтобы сообщение электронной почты отправлялось для любого уровня серьезности, кроме Disaster, выберите значения полей **Operator** и **Severity**, как это показано на рис. 8.

Рис. 8. Добавление нового условия выполнения действия

Далее для добавленного действия на вкладке **Operations** необходимо добавить операции. На рис. 9 три операции уже добавлены.

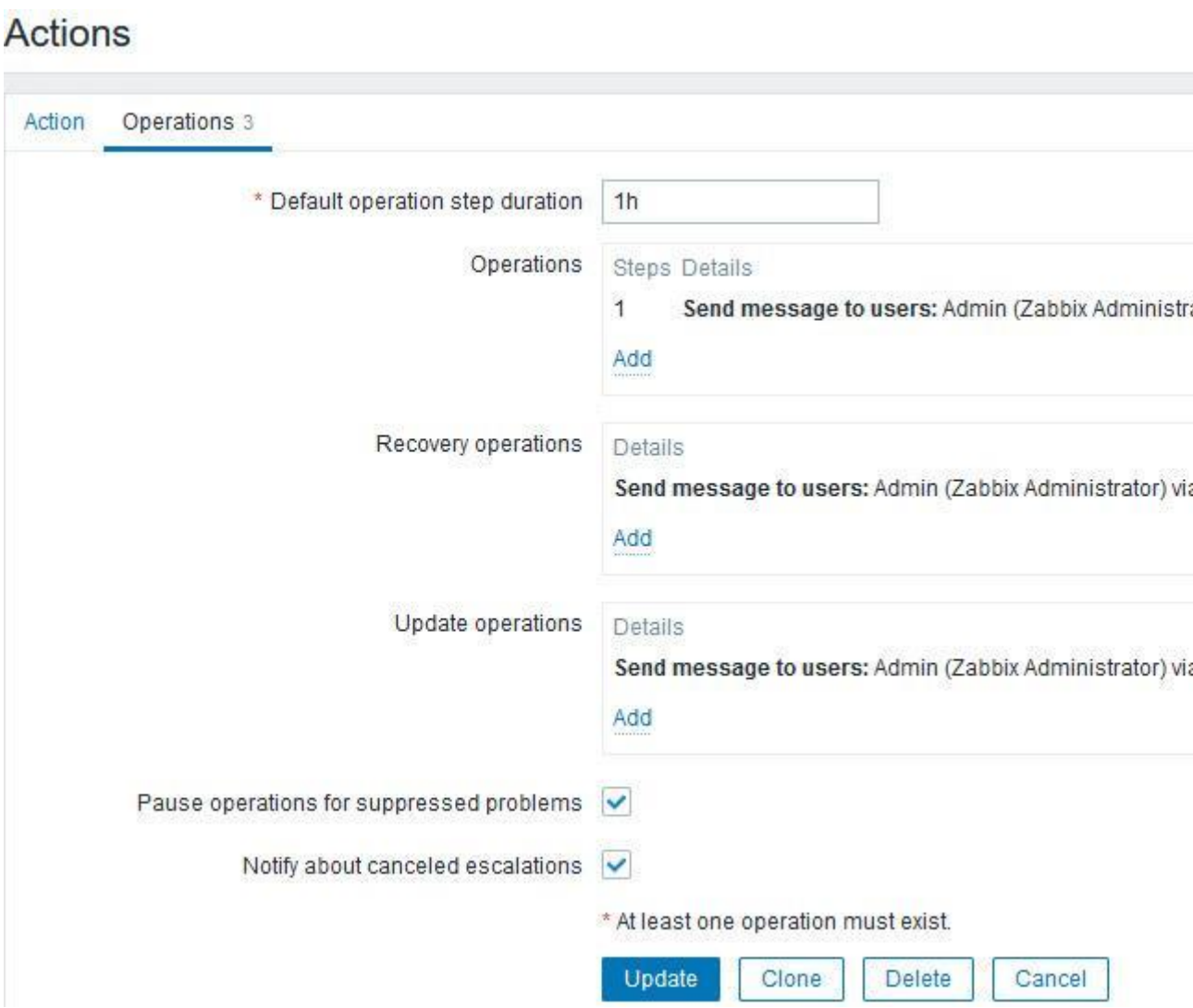


Рис. 9. Добавление операций

Для добавления операции воспользуйтесь ссылкой **Add**. Здесь можно добавить операции, которые будут выполнены при срабатывании триггера, при восстановлении его состояния (поле **Recovery operations**), а также при обновлении состояния (поле **Update operations**).

На рис. 10 форма редактирования уже добавленной операции.

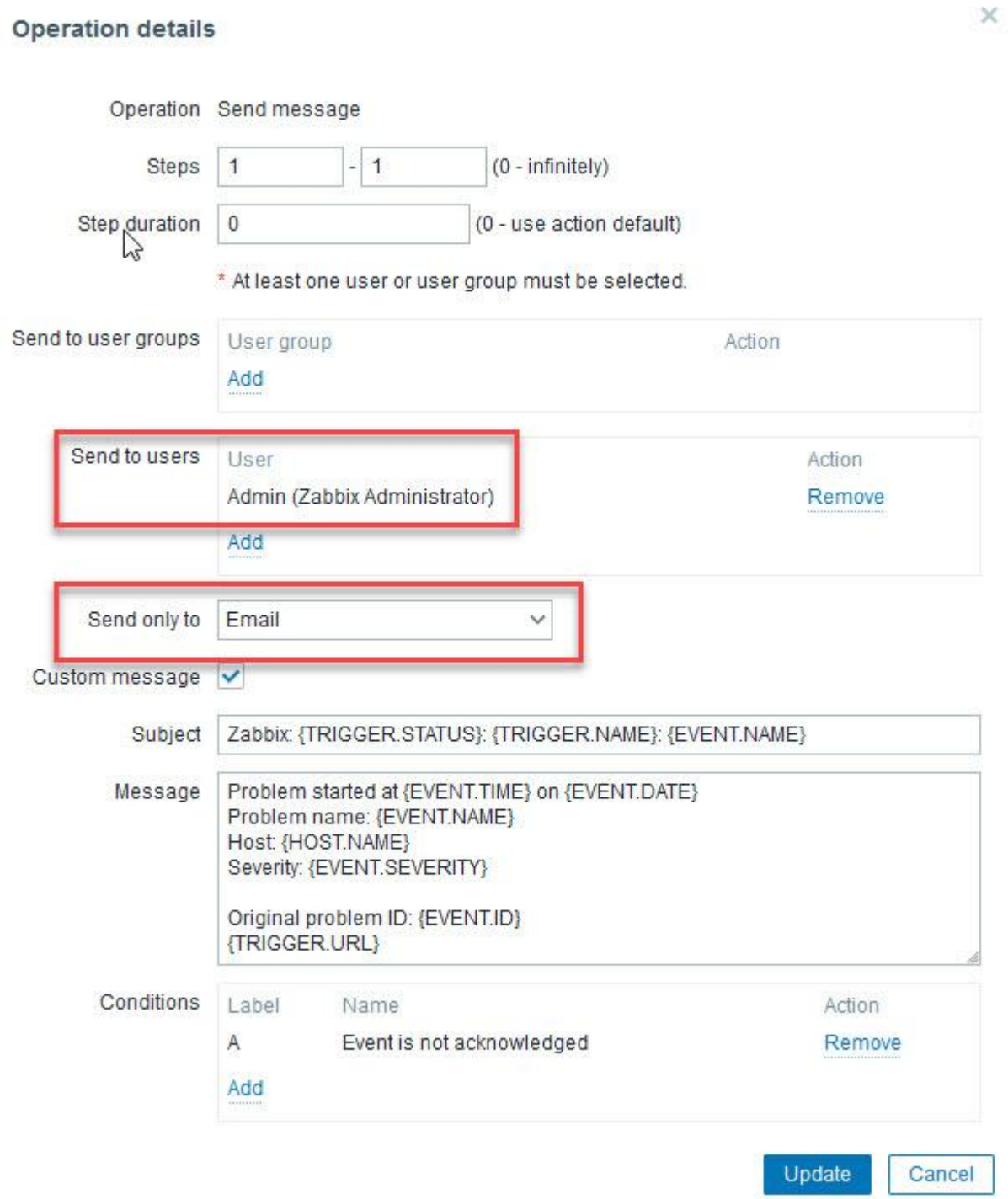


Рис. 10. Редактирование операции

Здесь как минимум необходимо выбрать пользователя в поле **Send to users** и способ извещения **Email** в поле **Send only to**.

Вы также можете отредактировать шаблон темы сообщения **Subject** и шаблон сообщения **Message**, отметив флажок **Custom message**. Если этого не сделать, будут использованы шаблоны сообщения по умолчанию, определенные в способе оповещения Email.

Официальная документация, имеющая отношение к настройке отправки сообщений по электронной почте, находится здесь:

- конфигурирование способа оповещения Email (на русском языке для версии 6.0);
- настройка действий (на русском языке)

Практическая работа № 17 Запуск скриптов на удаленных хостах

Задание:

Сначала нам понадобится настроить агент Zabbix для работы со скриптами:

1. Войдите на хост, на котором запущен агент.
2. Отредактируйте файл zabbix_agentd.conf:
3. Пропишите 1 в параметр “EnableRemoteCommands”, чтобы агент мог выполнять удаленные команды.
4. Так же поменяйте “UnsafeUserParameters” на 1, чтобы позволить агенту
5. Zabbix выполнять пользовательские скрипты.
6. Можно ещё включить “LogRemoteCommands”, если нужно вести журналы.

Перезапустите службу агента Zabbix, чтобы применить изменения.

Создание скриптов оповещения Zabbix

Чтобы создать скрипт оповещения в Zabbix, выполните следующие действия:

Перейдите в раздел “Administration” → “Scripts” в левом меню. Нажмите на кнопку “Create Script” в правом верхнем углу экрана.
Дайте скрипту имя, выберите нужный параметр области действия, заполните необходимую информацию, которая может включать параметры скрипта, аргументы команды или настройки типа медиа, и нажмите “Add”.

Теперь более подробно рассмотрим некоторые из вариантов настройки, такие как область действия и тип скрипта.

Направления скриптов

Скрипты операций действий:
Это скрипты, которые выполняются в рамках операции. Скрипты операций действий выполняются автоматически и могут использоваться для выполнения задач, таких как отправка уведомлений, запуск скриптов на удаленных хостах или обновление данных во внешних системах. Эти скрипты связаны с конкретным действием и выполняются при каждом его срабатывании.

Скрипты действий на хосте:
Это скрипты, которые можно выполнять вручную на конкретном хосте. Они связаны с

конкретным хостом и могут использоваться для выполнения задач, таких как перезапуск сервиса, запуск диагностической команды или выполнение резервного копирования. Скрипты действий на хосте выполняются вручную пользователем из веб-интерфейса Zabbix или API и могут быть выполнены только на хосте, с которым они связаны.

Скрипты действий на событии:
Это скрипты, которые можно выполнять вручную на конкретном событии. Они связаны с конкретным событием и могут использоваться для выполнения задач, таких как отправка пользовательского уведомления, обновление системы учета инцидентов или запуск рабочего процесса. Скрипты действий на событии выполняются вручную пользователем из веб-интерфейса Zabbix или API и могут быть выполнены только на событии, с которым они связаны.

Типы скриптов

Скрипты webhook: webhook – это способ для одного приложения предоставлять данные другому приложению в режиме реального времени. Скрипты webhook в Zabbix могут использоваться для отправки данных во внешнюю систему, такую как чат-приложение или платформа управления инцидентами. Чтобы настроить скрипт webhook в Zabbix, вам нужно создать новый тип медиа с соответствующими настройками (например, URL webhook и любые необходимые заголовки или аутентификация), а затем связать этот тип медиа с действием.

Скрипты: Zabbix также поддерживает настраиваемые скрипты, которые могут быть написаны на различных языках программирования (таких как Bash, Perl, Python или PowerShell) и выполняться в различных областях действия в зависимости от того, где предполагается запуск скрипта. Существуют три разных области действия для настраиваемых скриптов, которые определяют, где они могут быть использованы:

- Агент Zabbix
- Прокси-сервер Zabbix
- Сервер Zabbix

Чтобы настроить настраиваемый скрипт в Zabbix, вам нужно создать новый скрипт с соответствующими настройками кода и языка. Затем вы сможете использовать этот скрипт в различных контекстах, таких как в действии или в настраиваемом ключе элемента.

Скрипты SSH: скрипты SSH в Zabbix могут использоваться для выполнения команд или скриптов на удаленном хосте через SSH. Чтобы настроить скрипт SSH в Zabbix, вам нужно создать новую пару ключей SSH и добавить открытый ключ в файл authorized_keys на удаленном хосте. Затем вам нужно создать новый скрипт в Zabbix, который указывает команду для выполнения на удаленном хосте вместе с необходимыми деталями подключения SSH.

Скрипты Telnet: Telnet – это протокол сетевого взаимодействия, который обеспечивает виртуальное терминальное соединение с удаленным хостом. Чтобы настроить скрипт Telnet в Zabbix, вам нужно создать новый элемент Telnet в конфигурации Zabbix (на любом выбранном вами хосте, это проверка без агента, которая может указывать на любой IP-адрес, что может быть полезно для мониторинга хостов, не способных запускать агент), указав имя хоста или IP-адрес удаленного хоста, а также номер порта и любые

необходимые учетные данные для входа. Затем вам нужно создать новый скрипт в Zabbix, который указывает команду для выполнения на удаленном хосте.

Скрипты IPMI: IPMI (Intelligent Platform Management Interface) – это стандарт для удаленного управления сервером, который позволяет администраторам мониторить и управлять аппаратными компонентами, такими как блоки питания, вентиляторы и датчики температуры. Скрипты IPMI в Zabbix могут использоваться для выполнения IPMI-команд на удаленном сервере. Чтобы настроить скрипт IPMI в Zabbix, вам нужно включить IPMI на удаленном сервере и настроить необходимые сетевые настройки. Затем вам нужно создать новый скрипт в Zabbix, который указывает IPMI-команду для выполнения вместе с необходимыми деталями подключения.

Примеры скриптов оповещения Zabbix

После того, как вы создали свой скрипт оповещения, вы можете настроить его для выполнения конкретных действий при генерации оповещения. Например, вы можете настроить скрипт для перезапуска сервиса, отправки уведомления по электронной почте или выполнения команды на удаленном сервере.

Вот несколько примеров скриптов оповещения из базовой конфигурации Zabbix:

Обнаружение операционной системы: этот скрипт можно использовать для определения версии операционной системы на хосте

sudo /usr/bin/nmap -O/usr/bin/traceroute {HOST.CONN} {HOST.CONN}

Копировать

Ping: говорит сам за себя

ping -c 3 {HOST.CONN}; case $? in [01]) true;; *) false;; esac

Копировать

Перезапуск сервиса: перезапускает службу на хосте Windows, обнаруженном функцией автообнаружения

net start {TRIGGER.DESCRIPTION}

Практическая работа № 18 Мониторинг сетевого оборудования

Задание:

Мониторинг сетевого оборудования предполагает одинаковую настройку протокола SNMPv3 и на сервере мониторинга, и на наблюдаемом объекте.

Начнем с настройки сетевого устройства Cisco, его минимально необходимая конфигурация выглядит следующим образом (для конфигурирования используем CLI, имена и пароли я упростил во избежание путаницы):

```
snmp-server group snmpv3group v3 priv read snmpv3name

snmp-server user snmpv3user snmpv3group v3 auth md5 md5v3v3v3 priv des des56v3v3v3

snmp-server view snmpv3name iso included
```

Первая строка snmp-server group – определяет группу SNMPv3-пользователей (snmpv3group), режим чтения (read), и право доступа группы snmpv3group на просмотр определенных веток MIB-дерева объекта мониторинга (snmpv3name далее в конфигурации задает, к каким веткам MIB-дерева группа snmpv3group сможет получить доступ).

Вторая строка snmp-server user – определяет пользователя snmpv3user, его принадлежность к группе snmpv3group, а так же применение аутентификации md5 (пароль для md5 — md5v3v3v3) и шифрования des (пароль для des — des56v3v3v3). Разумеется, вместо des лучше использовать aes, здесь я его привожу просто для примера. Так же при определении пользователя можно добавить список доступа (ACL), регламентирующий IP-адреса серверов мониторинга, имеющих право осуществлять мониторинг данного устройства – это так же best practice, но я не буду усложнять наш пример.

Третья строка snmp-server view определяет кодовое имя, которое задает ветки MIB-дерева snmpv3name, чтобы их могла запрашивать группа пользователей snmpv3group. ISO, вместо строгого определения какой-то одной ветки, позволяет группе пользователей snmpv3group получать доступ ко всем объектам MIB-дерева объекта мониторинга.

Аналогичная настройка оборудования Huawei (так же в CLI) выглядит следующим образом:

```
snmp-agent mib-view included snmpv3name iso

snmp-agent group v3 snmpv3group privacy read-view snmpv3name

snmp-agent usm-user v3 snmpv3user group snmpv3group

snmp-agent usm-user v3 snmpv3user authentication-mode md5
    md5v3v3v3

snmp-agent usm-user v3 snmpv3user privacy-mode des56
    des56v3v3v3
```

После настройки сетевых устройств, необходимо проверить наличие доступа с сервера мониторинга по протоколу SNMPv3, я воспользуюсь snmpwalk:

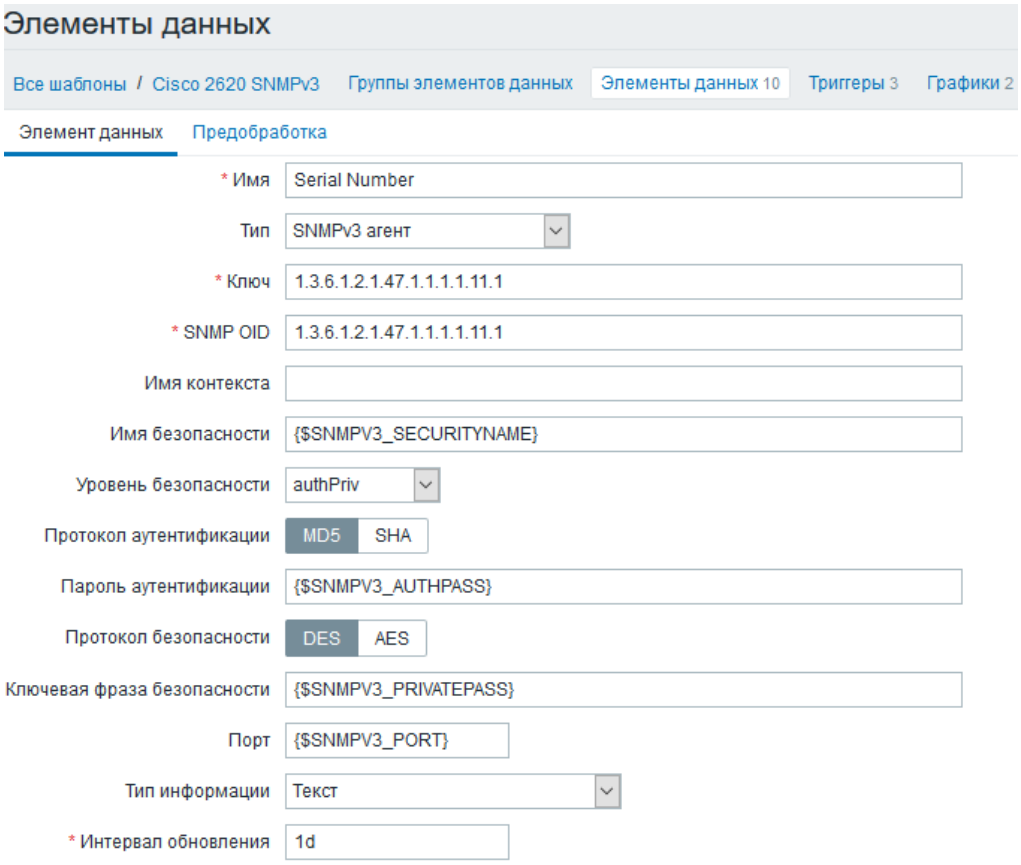
snmpwalk -v 3 -u snmpv3user -l authPriv -A md5v3v3v3 -a md5 -x des -X des56v3v3v3 10.10.10.252

```
zabbix@zabbix:~$ snmpwalk -v 3 -u snmpv3user -l authPriv -A md5v3v3v3 -a md5 -x des -X des56v3v3v3 10.10.10.252
iso.3.6.1.2.1.1.1.0 = STRING: "Cisco IOS Software"
iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.9.1.467
iso.3.6.1.2.1.1.3.0 = Timeticks: (29633529) 3 days, 10:18:55.29
iso.3.6.1.2.1.1.7.0 = INTEGER: 78
iso.3.6.1.2.1.1.8.0 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.1.9.1.2.1 = OID: iso.3.6.1.4.1.9.7.129
iso.3.6.1.2.1.1.9.1.2.2 = OID: iso.3.6.1.4.1.9.7.115
iso.3.6.1.2.1.1.9.1.2.3 = OID: iso.3.6.1.4.1.9.7.265
iso.3.6.1.2.1.1.9.1.2.4 = OID: iso.3.6.1.4.1.9.7.112
iso.3.6.1.2.1.1.9.1.2.5 = OID: iso.3.6.1.4.1.9.7.106
iso.3.6.1.2.1.1.9.1.2.6 = OID: iso.3.6.1.4.1.9.7.47
iso.3.6.1.2.1.1.9.1.2.7 = OID: iso.3.6.1.4.1.9.7.122
iso.3.6.1.2.1.1.9.1.2.8 = OID: iso.3.6.1.4.1.9.7.135
iso.3.6.1.2.1.1.9.1.2.9 = OID: iso.3.6.1.4.1.9.7.43
iso.3.6.1.2.1.1.9.1.2.10 = OID: iso.3.6.1.4.1.9.7.37
^C
zabbix@zabbix:~$
```

Более наглядный инструмент для запроса конкретных OID-объектов, с использованием MIB-фалов – snmpget:

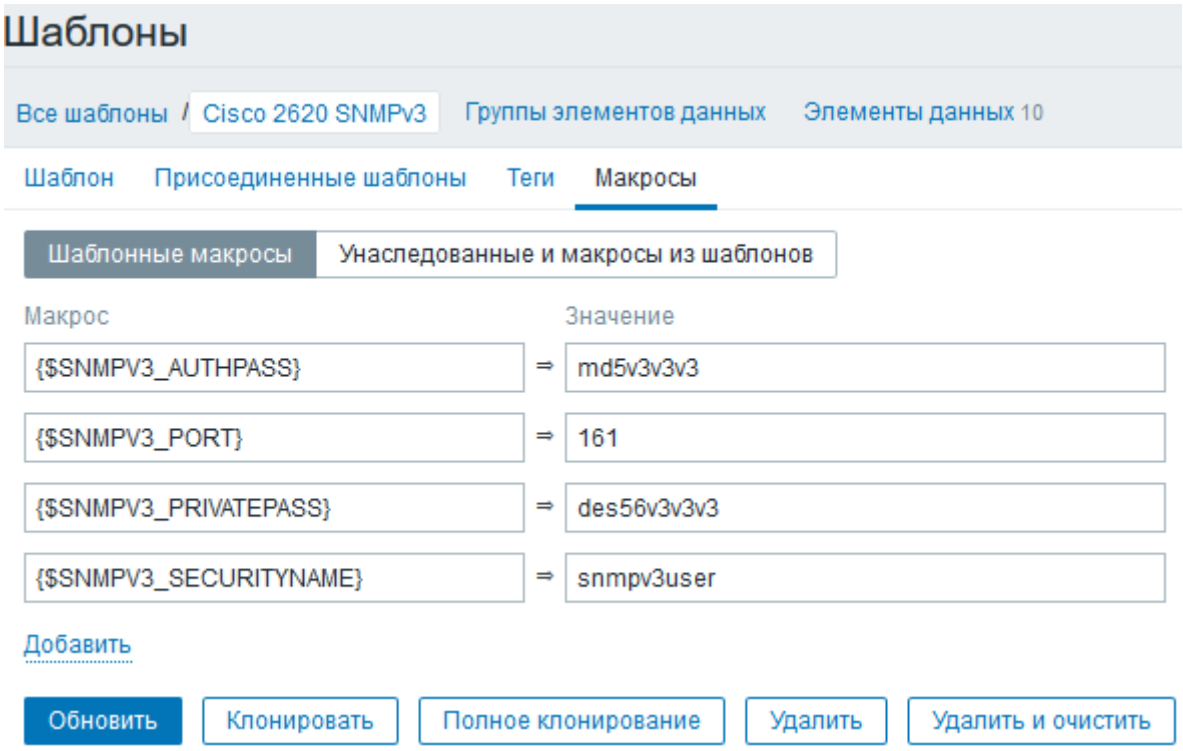
```
zabbix@zabbix:~$ snmpget -v 3 -u snmpv3user -l authPriv -A md5v3v3v3 -a md5 -x des -X des56v3v3v3
10.10.10.253 SNMPv2-MIB::system.sysUpTime.0
SNMPv2-MIB::sysUpTime.0 = Timeticks: (172253589) 19 days, 22:28:55.89
zabbix@zabbix:~$
```

Теперь перейдем к настройке типового элемента данных для SNMPv3, в рамках Zabbix-шаблона. Для простоты и независимости от MIB, я использую цифровые OID:



Я использую в ключевых полях пользовательские макросы, поскольку они будут одинаковы для всех элементов данных в шаблоне. Задавать их можно в рамках шаблона, если в Вашей сети у всех сетевых устройств параметры SNMPv3 одинаковы, или в рамках

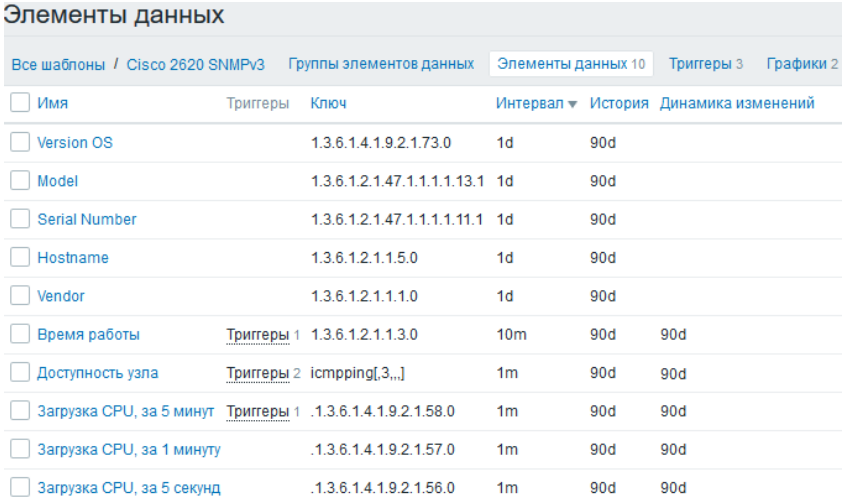
узла сети, если параметры SNMPv3 для разных объектов мониторинга отличаются:



Обратите внимание, система мониторинга располагает только именем пользователя, и паролями для аутентификации и шифрования. Группа пользователей и область MIB-объектов, к которым разрешен доступ, задается на объекте мониторинга.
Теперь перейдем к наполнению шаблона.

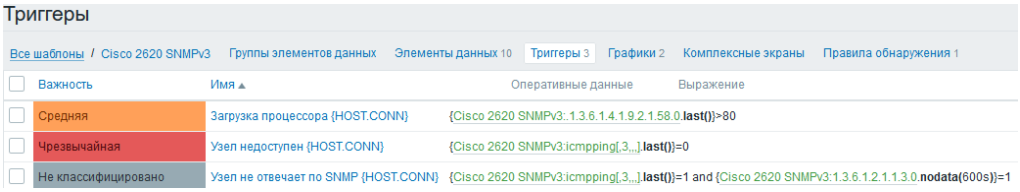
Шаблон опроса в Zabbix

Простое правило при создании любых шаблонов опроса – делать их максимально подробными:



Я уделяю большое внимание инвентаризации, чтобы с большой сетью было удобнее

работать. Об этом немного позднее, а пока – триггеры:

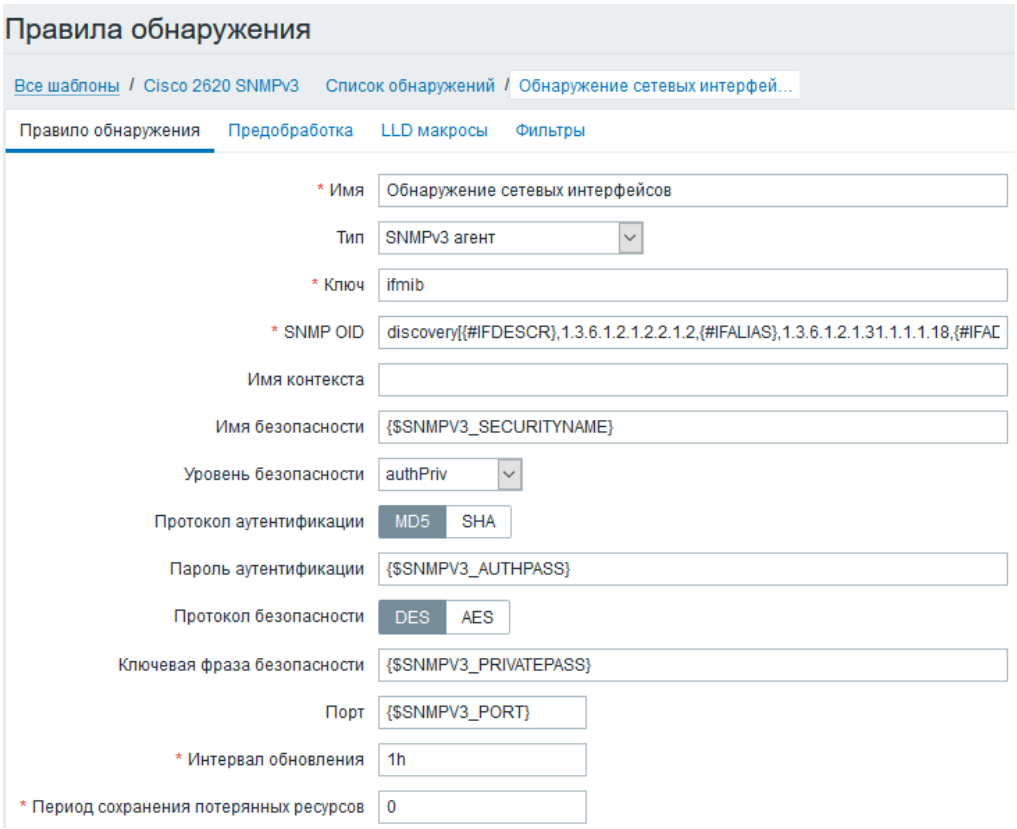


Для удобства визуализации триггеров в их названия заложены системные макросы {HOST.CONN}, чтобы на дашборде в разделе алёртинга выводились не только имена устройств, но и IP-адреса, хотя это больше вопрос удобства, чем необходимости. Для определения недоступности устройства, помимо обычного echo-запроса, я использую проверку на недоступность узла по протоколу SNMP, когда объект доступен по ICMP, но не отвечает на SNMP-запросы – такая ситуация возможна, например, при дублировании IP-адресов на разных устройствах, из-за некорректно настроенных межсетевых экранов, или неверных настроек SNMP на объектах мониторинга. Если использовать проверку доступности узлов только по ICMP, в момент расследования инцидентов в сети, данных мониторинга может не оказаться, поэтому их поступление нужно контролировать.

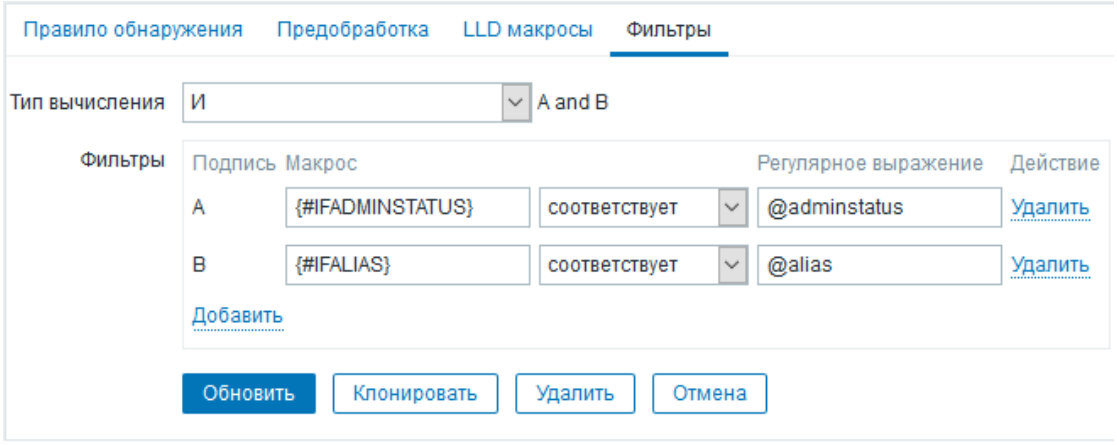
Перейдем к обнаружению сетевых интерфейсов – для сетевого оборудования это самая важная функция мониторинга. Поскольку на сетевом устройстве могут быть сотни интерфейсов, необходимо фильтровать ненужные, чтобы не загромождать визуализацию и не захламлять базу данных.

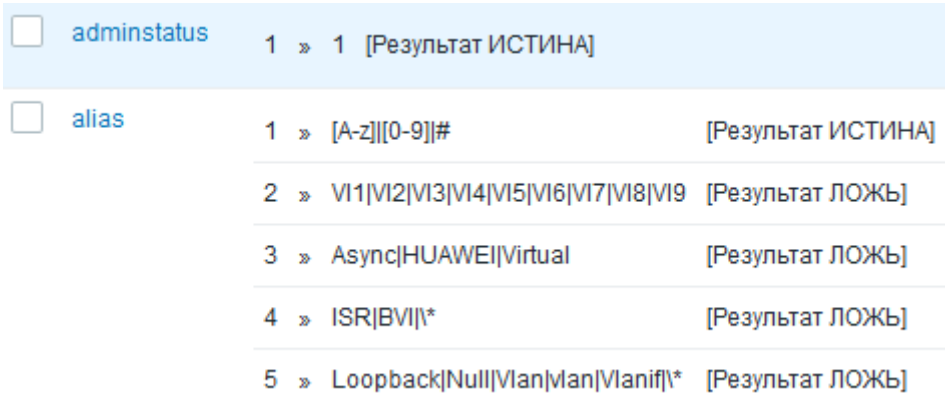
Я использую стандартную функцию обнаружения для SNMP, с большим количеством обнаруживаемых параметров, для более гибкой фильтрации:

discovery[{#IFDESCR},1.3.6.1.2.1.2.2.1.2,{#IFALIAS},1.3.6.1.2.1.31.1.1.1.18,{#IFADMINSTATUS},1.3.6.1.2.1.2.2.1.7]



При таком обнаружении, можно фильтровать сетевые интерфейсы по их типам, пользовательским описаниям «description», и административным статусам портов. Фильтры и регулярные выражения для фильтрации в моем случае выглядят следующим образом:





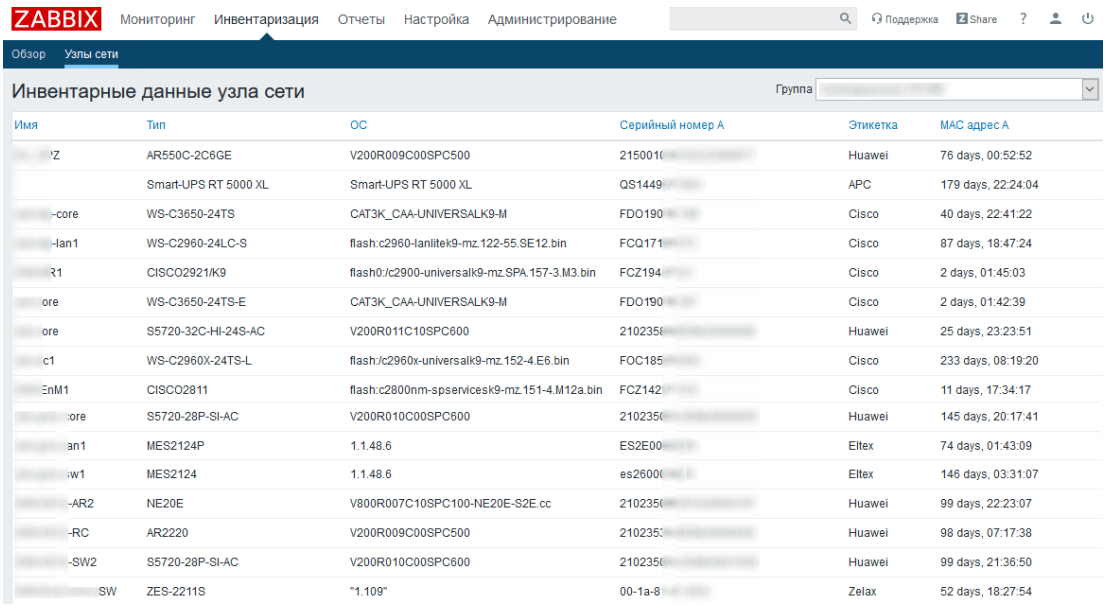
При обнаружении будут исключены следующие интерфейсы:

- выключенные вручную (adminstatus<>1), благодаря IFADMINSTATUS;
- не имеющие текстового описания, благодаря IFALIAS;
- имеющие в текстовом описании символ *, благодаря IFALIAS;
- являющиеся служебными или техническими, благодаря IFDESCR (в моем случае, в регулярных выражениях IFALIAS и IFDESCR проверяются одним регулярным выражением alias).

Шаблон для сбора данных по протоколу SNMPv3 почти готов. Не будем подробнее останавливаться на прототипах элементов данных для сетевых интерфейсов, перейдем к результатам.

Итоги мониторинга

Для начала – инвентаризация небольшой сети:



Имя	Тип	ОС	Серийный номер A	Этикетка	MAC адрес A
[illegible]Z	AR550C-2C6GE	V200R009C00SPC500	215001[illegible]	Huawei	76 days, 00:52:52
	Smart-UPS RT 5000 XL	Smart-UPS RT 5000 XL	QS1449[illegible]	APC	179 days, 22:24:04
[illegible]-core	WS-C3650-24TS	CAT3K_CAA-UNIVERSALK9-M	FDO190[illegible]	Cisco	40 days, 22:41:22
[illegible]-lan1	WS-C2960-24LC-S	flash:c2960-lanlitek9-mz.122-55.SE12.bin	FCQ171[illegible]	Cisco	87 days, 18:47:24
[illegible]21	CISCO2921/K9	flash0:/c2900-universalk9-mz.SPA.157-3.M3.bin	FCZ194[illegible]	Cisco	2 days, 01:45:03
[illegible]ore	WS-C3650-24TS-E	CAT3K_CAA-UNIVERSALK9-M	FDO190[illegible]	Cisco	2 days, 01:42:39
[illegible]ore	S5720-32C-HI-24S-AC	V200R011C10SPC600	210235[illegible]	Huawei	25 days, 23:23:51
[illegible]c1	WS-C2960X-24TS-L	flash:/c2960x-universalk9-mz.152-4.E6.bin	FOC185[illegible]	Cisco	233 days, 08:19:20
[illegible]EnM1	CISCO2811	flash:c2800nm-spservicesk9-mz.151-4.M12a.bin	FCZ142[illegible]	Cisco	11 days, 17:34:17
[illegible]:ore	S5720-28P-SI-AC	V200R010C00SPC600	210235[illegible]	Huawei	145 days, 20:17:41
[illegible]an1	MES2124P	1.1.48.6	ES2E00[illegible]	Eltex	74 days, 01:43:09
[illegible]w1	MES2124	1.1.48.6	es2600[illegible]	Eltex	146 days, 03:31:07
[illegible]-AR2	NE20E	V800R007C10SPC100-NE20E-S2E.cc	210235[illegible]	Huawei	99 days, 22:23:07
[illegible]-RC	AR2220	V200R009C00SPC500	210235[illegible]	Huawei	98 days, 07:17:38
[illegible]-SW2	S5720-28P-SI-AC	V200R010C00SPC600	210235[illegible]	Huawei	99 days, 21:36:50
[illegible]SW	ZES-2211S	"1.109"	00-1a-8[illegible]	Zelax	52 days, 18:27:54

Если подготовить шаблоны для каждой серии сетевых устройств – можно добиться удобной для анализа компоновки сводных данных по актуальному ПО, серийным номерам, и оповещении о приходе в серверную уборщицы (по причине малого Uptime). Выдержка моего списка шаблонов ниже:

Cisco Switches 3650 Series

Cisco Switches 3850 Series

Eltex Switch MES2124

Eltex Switch MES2124P

Eltex Switch MES2348B

Eltex Switch MES2348P

EnteIUPS

HP iLO

Huawei AR550C Switch

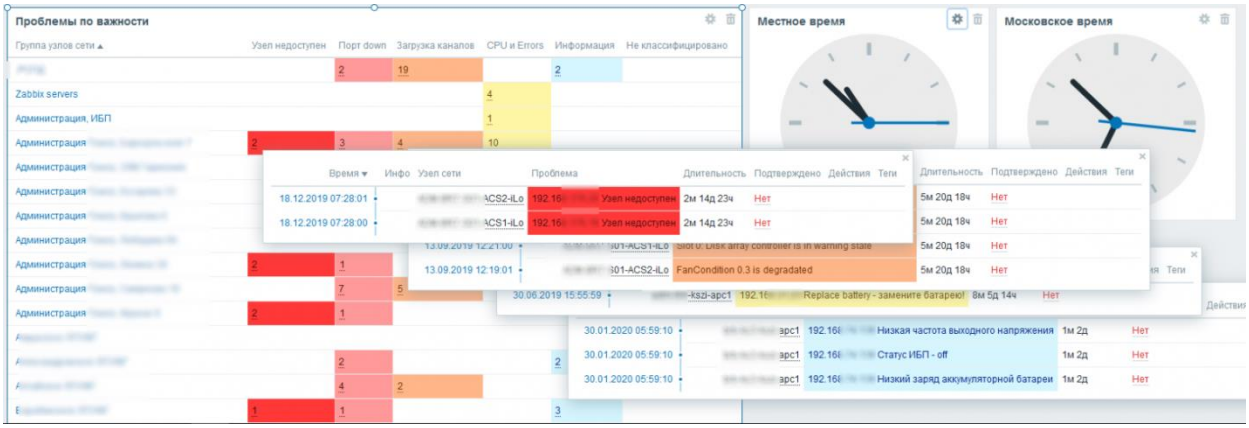
Huawei NQA

Huawei Router AR2200

Huawei Router NE08E

Huawei Router NE20E

А теперь – главная панель мониторинга, с распределенными по уровням важности триггерами:



Благодаря комплексному подходу к шаблонам для каждой модели устройств в сети, можно добиться того, что в рамках одной системы мониторинга будет организован инструмент для прогнозирования неисправностей и аварий (при наличии соответствующих датчиков и метрик). Zabbix хорошо подходит для мониторинга сетевых, серверных, сервисных инфраструктур, и задача обслуживания сетевого оборудования наглядно демонстрирует её возможности.

Практическая работа № 19 Управление пользователями Zabbix

Задание:

Нажмите на вкладку «Администрирование» в меню навигации, затем нажмите «Пользователи».

Нажмите на кнопку «Создать пользователя» или выберите пользователя, которого хотите изменить.

Заполните поля Имя пользователя и Пароль.

В разделе Группа выберите группу пользователей, к которой должен принадлежать пользователь. Если группа ещё не существует, вы можете создать ее, используя инструкции ниже.

В разделе разрешения вы должны выбрать предварительно созданную роль для вашего пользователя, которая указывает, какие элементы пользовательского интерфейса доступны для этого пользователя. Также устанавливаются разрешения на конкретные действия, например, на подтверждение проблем или удаление событий.

Создание групп пользователей

Группы пользователей в Zabbix служат способом управления и организации пользователей, которые выполняют сходные обязанности в системе мониторинга. Группы пользователей позволяют администраторам назначать разрешения группе пользователей, вместо того, чтобы назначать разрешения каждому пользователю индивидуально. Это помогает упростить управление доступом пользователей и облегчает обеспечение того, что все пользователи с похожими обязанностями имеют соответствующий уровень доступа.

Меню «Администрирование», нажимаем «Группы пользователей».

Нажмите кнопку «Создать группу пользователей» или группу, которую вы хотите изменить.

В разделе «Детали группы пользователей» укажите название группы и добавьте пользователей для этой группы.

В разделе «Разрешения» вы можете указать, на какие группы узлов сети и шаблоны, а также на другие связанные с ними объекты группа пользователей может иметь доступ и читать и/или записывать.

Вкладка «Фильтр по тегам» позволяет давать доступ на основе тегов. Напишите комбинацию необходимых групп хостов и тегов.

Роли пользователей

«Роли пользователей» во вкладке «Администрирование». Как обычно, создайте роль пользователя или измените существующую.

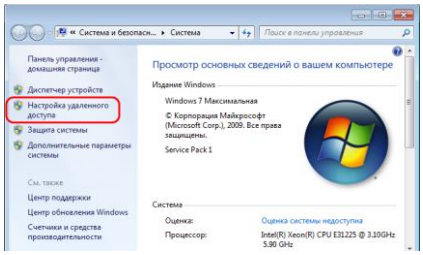
Дайте ей имя и выберите, какие элементы пользовательского интерфейса доступны для этой роли (вы можете отталкиваться от трех типов пользователей, которые она предлагает). Затем вы можете выбрать доступ к модулям, службам и действиям.

Сохраните изменения, и готово.

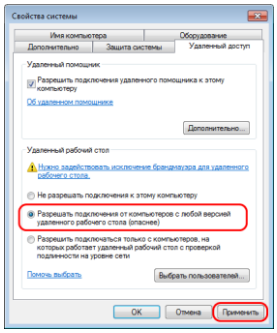
Практическая работа № 20 Настройка средств удаленного администрирования

Задание:

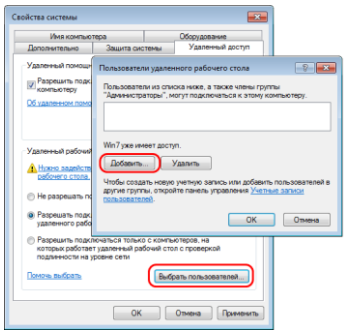
Чтобы на удалённом компьютере возможно было подключение по RDP, в его среде Windows необходимо разрешить это дело. Жмём клавиши Win+Pause, выбираем «Настройка удалённого доступа».



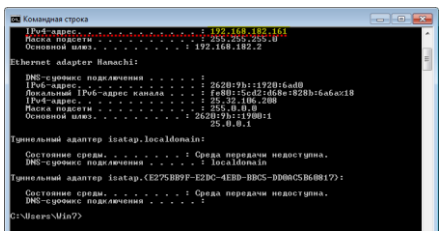
Активируем галочку разрешения подключения от компьютеров с любой версией удалённого рабочего стола. Жмём «Применить».



Возможность удалённого доступа активирована. Но такая вот обычная активация открывает удалённый доступ только к учётным записям администратора. Если нужно подключаться к учётным записям обычных стандартных пользователей, таких, войдя в систему с учётки администратора, необходимо добавить дополнительно. Жмём «Выбрать пользователей», далее «Добавить» и указываем имена стандартных учёток.



Далее нам нужно получить IP удалённого компьютера. Запускаем командную строку, вводим:
ipconfig



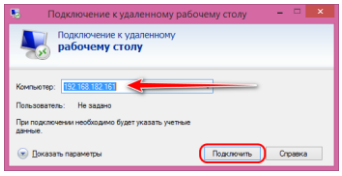
Смотрим IP-адрес компьютера. Кроме IP-адреса, нам для подключения также потребуется, как упоминалось, имя учётной записи Windows и пароль.

↥ Удалённое подключение

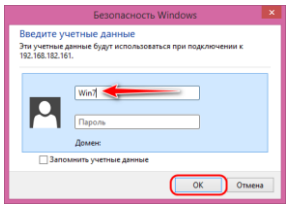
Как теперь удалённо подключиться? Жмём Win+R, вводим:

mstsc.exe

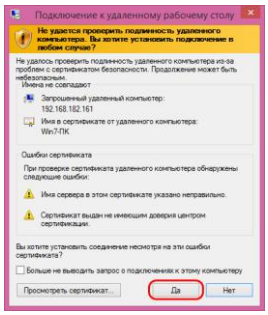
Откроется окошко штатной утилиты RDP-подключения. Вводим здесь IP-адрес удалённого компьютера. Жмём «Подключить».



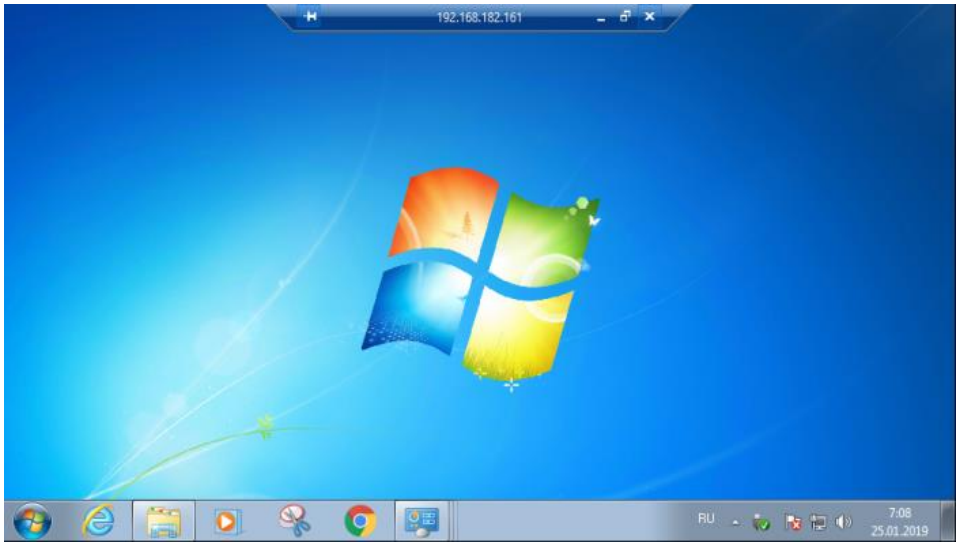
Далее вводим имя учётной записи удалённого компьютера и пароль. Для обычных локальных учётных записей Windows вводится их пароль, а для учётных записей Microsoft – соответственно, их. Но важно: не задаваемый на каждом отдельном компьютере местный пин-код, а именно пароль от подключённого веб-аккаунта Microsoft. В качестве имени пользователя при подключённой учётке Microsoft вводится электронная почта аккаунта.



Жмём «Ок» в окне проверки безопасности подключения.



Ну и, собственно, подключаемся.



Практическая работа № 21 Инвентаризация рабочих станций

Задание:

1. Произведите инвентаризацию всех рабочих станций кабинета с помощью утилит: systeminfo, msconfig, msinfo32, ipconfig, Netstat. Route, tracert.
2. Создайте таблицу сети с указанием устройств (ip и mac адресов), характеристик рабочей станции, версии ОС.

Практическая работа № 22 Выявление неполадок в работе сети

Задание:

Используя заданные команды определите неполадки в работе виртуальной сети:

Ping - диагностическая утилита, которая проверяет возможность соединения с удаленным компьютером.

Pathping - усовершенствованная утилита ping, которая также отражает маршрут прохож-дения и предоставляет статистику потери пакетов на промежуточных маршрутизаторах.

Route - показывает и позволяет изменять конфигурацию локальной таблицу маршрутиза-ции.

Tracert - отслеживает маршрут, по которому пакеты перемещаются на пути к пункту назначения.

Netstat - показывает текущую информацию сетевого соединения TCP/IP. Например, ин-формацию о подключенном хосте и номера используемых портов.

Ipconfig - показывает текущую конфигурацию TCP/IP на локальном компьютере.

Hostname - показывает локально настроенное имя узла TCP/IP ..

Arp - показывает и позволяет изменять кэш протокола ARP (Address Resolution Protocol), где хранится информация о соответствии IP - адресов - MAC - адресам локальных узлов.

Nslookup - утилита командной строки - распознаватель для запросов DNS сервера.

Утилиты выполняются из командной строки.

Чтобы определить причину неполадок, попытайтесь выполнить обмен пакетами (утилита ping) с IP-адресом другого компьютера. Таким компьютером может быть компьютер, с которым вы пытаетесь соединиться, или основной шлюз.

Чтобы определить IP-адрес основного шлюза: наберите в командной строке ipconfig и нажмите клавишу ENTER. Если требуемая информация уходит с экрана, то для просмотра экранов по очереди введите ipconfig | more и нажмите клавишу ENTER. В отображаемых ре-зультатах найдите строку Основной шлюз и запишите соответствующий IP-адрес.

Чтобы выполнить обмен пакетами (ping) с другим компьютером: наберите в командной строке: ping адрес, где адрес представляет IP-адрес другого компьютера, и нажмите клавишу ENTER.

Утилита Ipconfig показывает текущую конфигурацию TCP/IP на локальном компьютере.

Ключи утилиты:

/release - освобождает полученный от DHCP IP - адрес. /renew - получает от DHCP новый IP - адрес. /all - показывает всю информацию о TCP/IP конфигурации. /flushdns - очищает кэш локального распознавателя DNS. /regsiterdns - обновляет адрес в DHCP и перерегистрирует его в DNS. /displaydns - показывает содержание кэша распознавателя DNS

Практическая работа № 23 Физическое обслуживание сетевого оборудования

Задание:

Задание 1: Последовательно подключая тестовые сетевые кабели 1-3 к сетевой карте ПК выполнить, используя омметр, прозвонку кабеля, результаты измерения занести в таблицу:

Таблица 1

Результаты измерений

Номер тестового сетевого кабеля	Результат измерения пар кабеля, Ом				Вывод о состоянии кабеля
	1	2	3	4	

Задание 2

Используя сетевой тестер проверить правильность разделки кабеля и определить вариант разделки кабеля результаты измерения занести в таблицу:

Таблица 2

Результаты измерений

Номер тестового сетевого кабеля	Тип разделки
1	
2	

Запуская утилиты из командной строки меню выполнить следующие проверки:

Проверить с помощью утилиты ping соединение компьютера рабочей станцией с IP-адресом 192.168.1.12 и сервером «Neptun». Записать результаты работы программы.

Проверить с помощью утилиты Pathping соединение компьютера рабочей станцией с IP-адресом 192.168.1.12 и сервером «Neptun». Записать результаты работы программы.

С помощью утилиты Route просмотреть таблицу маршрутизации сервера. Записать результаты работы программы.

С помощью утилиты Tracert просмотреть маршрут до сервера «Neptun» и до рабочей станции с IP-адресом 192.168.1.12. Записать результаты работы программы.

С помощью утилиты Ipconfig выполнить следующие действия:

получить всю информацию о TCP/IP конфигурации;

Получить новый адрес для ПК;

Просмотреть на сервере содержание кэша DNS;

Записать результаты работы программы.

Контрольные вопросы:

1. Назначение ЛВС?

2.Каковы основные элементы ЛВС и каково их назначение?

3.Какие существуют методы поиска неисправностей ЛВС?

4.Каков принцип работы сетевого сканера, какие типы неисправностей можно с помощью его определить и как?

5.Где охраняться соответствия IP И MAC адресов ?

6.Как проверить линию связи между ПК и сервером?

7.Как выполнить принудительную смену IP адреса ПК?

Практическая работа № 24 Физическое обслуживание кабельной системы

Задание:

1. Произведите проверку СКС в заданном помещении.
2. Замените поврежденный кабель.
3. Организуйте правильный кабель-менеджмент.

Практическая работа №25 Физическое обслуживание рабочих станций

Задание:

1. Проверьте техническое состояние рабочих станций.
2. Замените комплектующие, если это необходимо.
3. Замените термопасту.
4. Продуйте систему охлаждения.

Практическая работа № 26 Проверка работы мониторов

Задание:

1. С помощью тестового приложения произведите проверку работы мониторов в заданном помещении.
2. Опишите выявленные недостатки.
3. Исправьте и зафиксируйте результат.

Тест для проверки монитора | Официальный сайт Kuycon Россия

Практическая работа № 27 Обновление системного программного обеспечения

Задание:

На всех рабочих станциях обновите системное программное обеспечение. Зафиксируйте выполнение скриншотами. Зафиксируйте обновления в журнале, указав версию и время обновления.

Практическая работа № 28 Установка прикладного программного обеспечение

Задание:

Установите следующее прикладное программное обеспечение на все рабочие станции:

1. Текстовый процессор (указать какой и почему)
2. Табличный процессор (указать какой и почему)
3. Графический редактор (указать какой и почему)
4. Просмотрщик PDF-файлов (указать какой и почему)
5. Мультимедийный проигрыватель (указать какой и почему)
6. Архиватор (указать какой и почему)
7. Яндекс Браузер

Практическая работа № 29 Обновление прикладного программного обеспечения

Задание:

Обновите базу сигнатур антивируса на всех рабочих станциях и проверьте каждую рабочую станцию на наличие вирусов.

Практическая работа № 30 Проверка работоспособности периферийных устройств

Задание:

1. Установите удаленный доступ к сетевым МФУ.
2. Проверьте печать, распечатав любой документ.
3. Проверьте сканирование, отсканировав распечатанный докумет

Практическая работа № 31 Обновление политик безопасности

Задание:

1. Проверьте работоспособность всех настроенных политик.
2. Обновите парольную политику. Примените ее для всех пользователей.
3. Добавьте политики безопасности, если есть необходимость.

Практическая работа № 32 Проверка учетных записей в AD

Задание:

1. Сверьте список пользователей системы со списком работающих сотрудников организации.
2. Удалите лишние учетные записи в домене AD.
3. Проверьте права доступа каждого сотрудника. Сверьте права доступа с матрицей.

Практическая работа №33 Проверка учетных записей в ALD

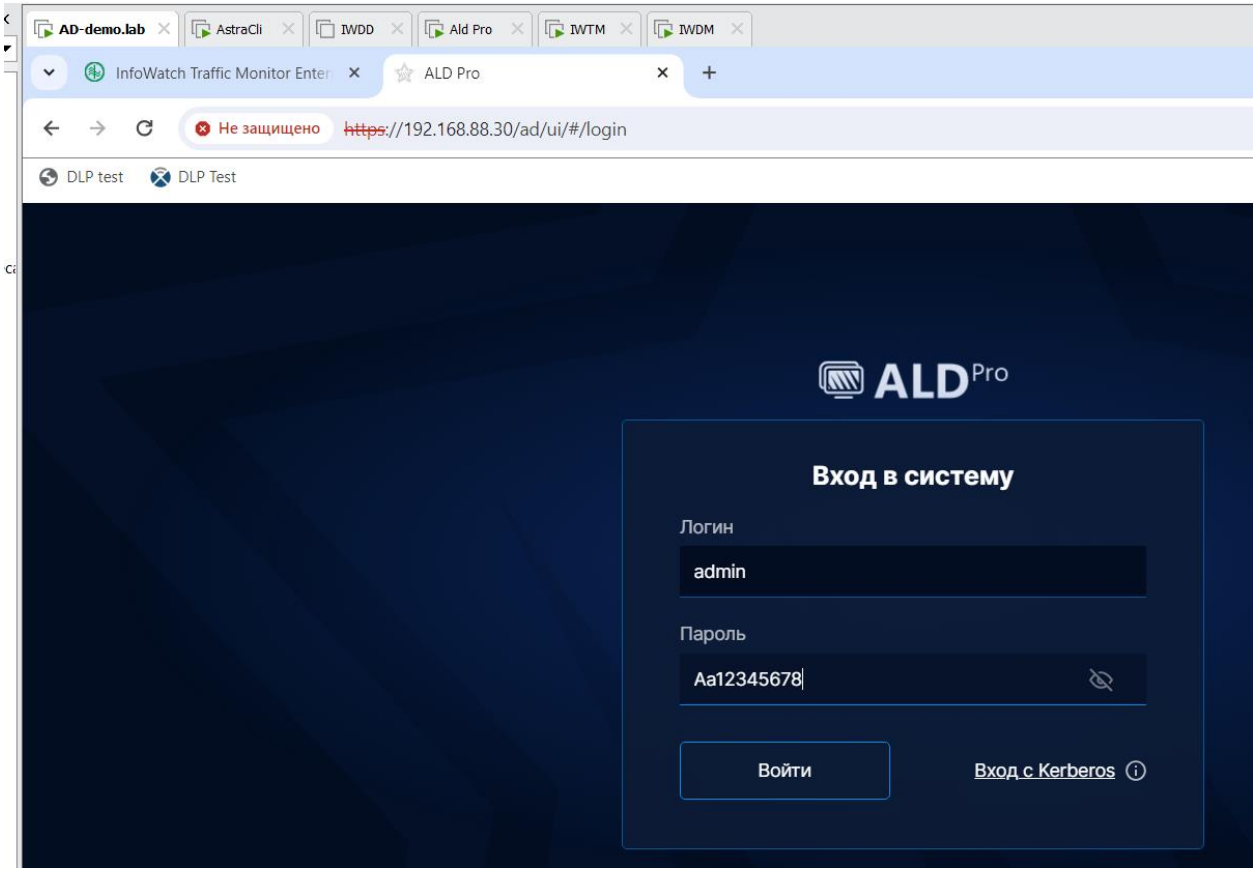
Задание:

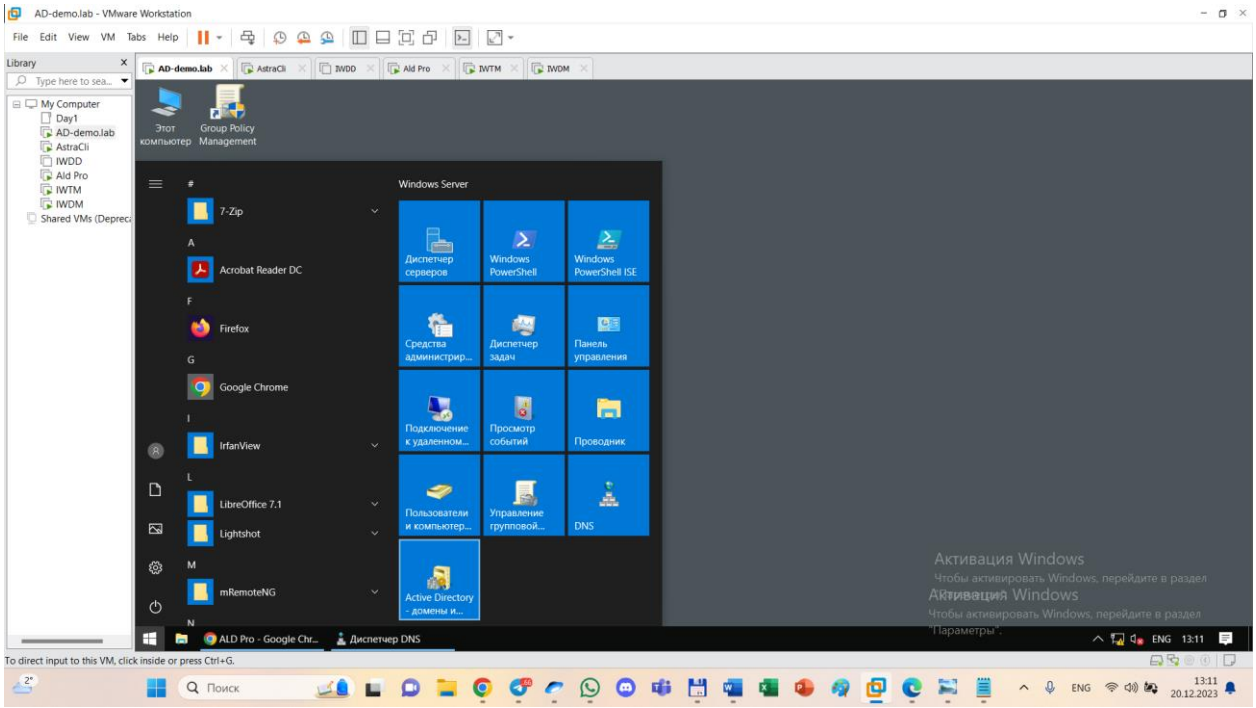
1. Сверьте список пользователей системы со списком работающих сотрудников организации.
2. Удалите лишние учетные записи в домене ALD.
3. Проверьте права доступа каждого сотрудника. Сверьте права доступа с матрицей.

Практическая работа № 34 Перенос учетных записей из AD в ALD PRO

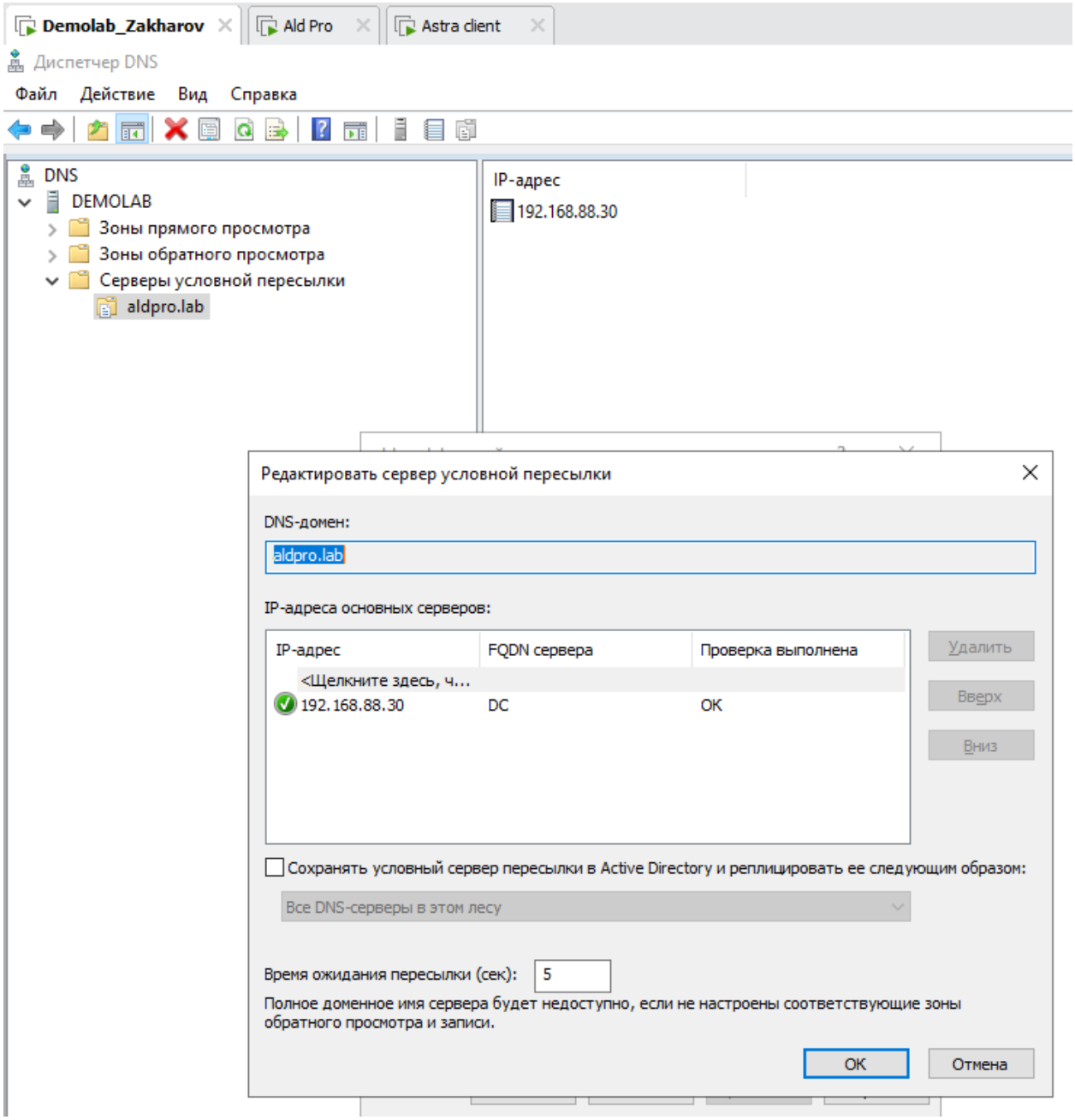
Задание:

Заходим в ALDPro:

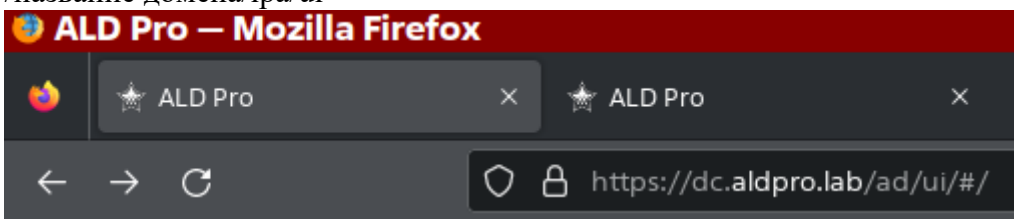




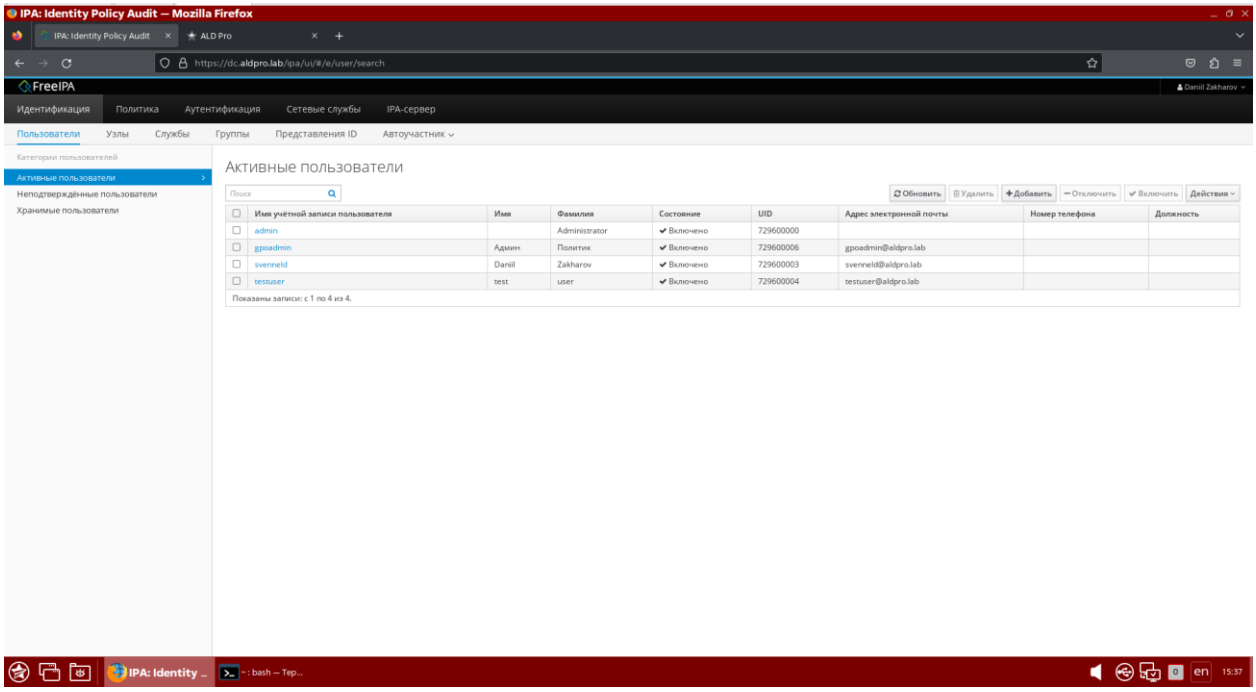
Диспетчер DNS > ПКМ по Серверы условной пересылки > Создать сервер условной пересылки с названием домена aldpro.
Вписываем ip адрес и ждем подключения, которое должно через время стать зеленым.



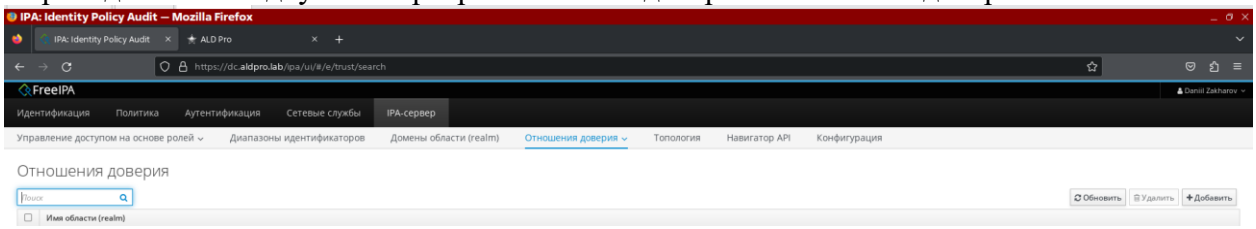
Переходим в ALDPRO. Меняем адрес с /название домена/ad на /название домена/ipa/ui



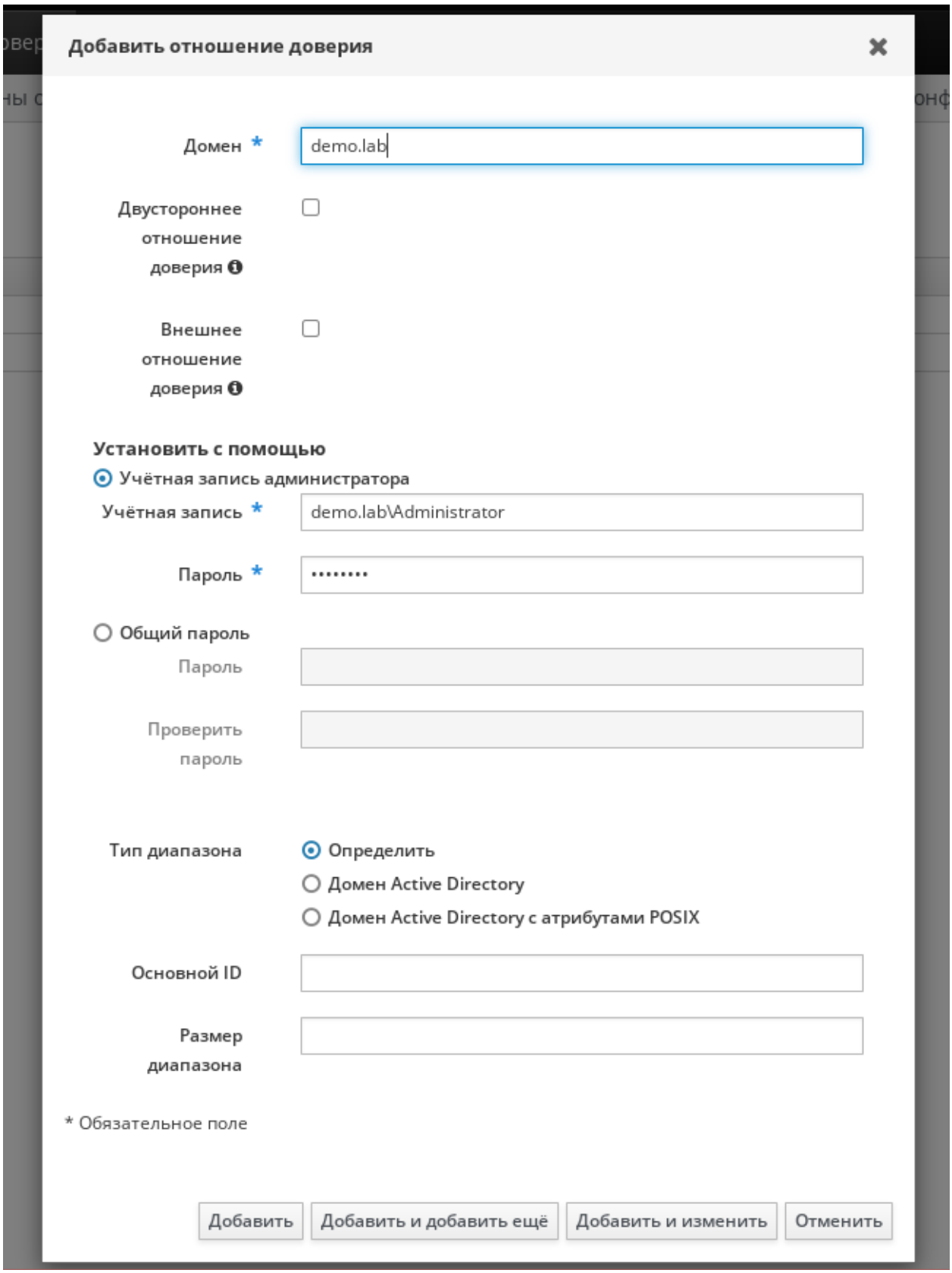
Откроется окно IPA



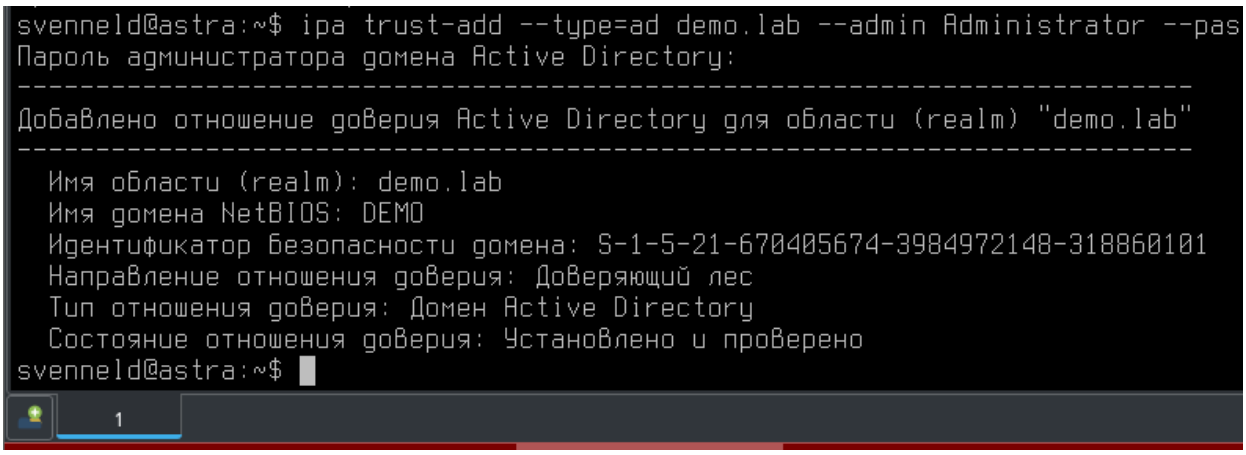
Переходим на вкладку IPA-сервер-отношения доверия-отношения доверия.



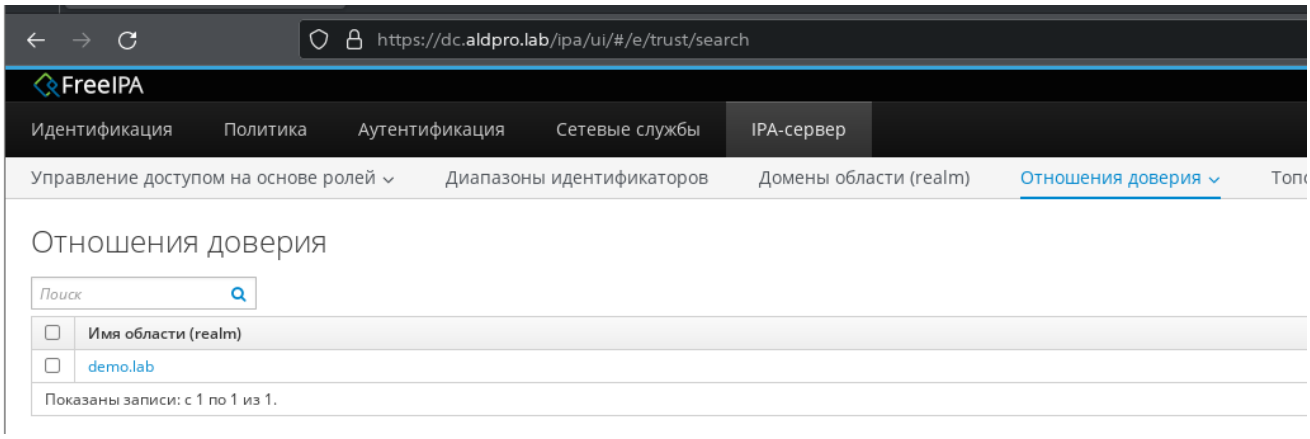
Нажимаем +Добавить. И заполняем необходимые поля.



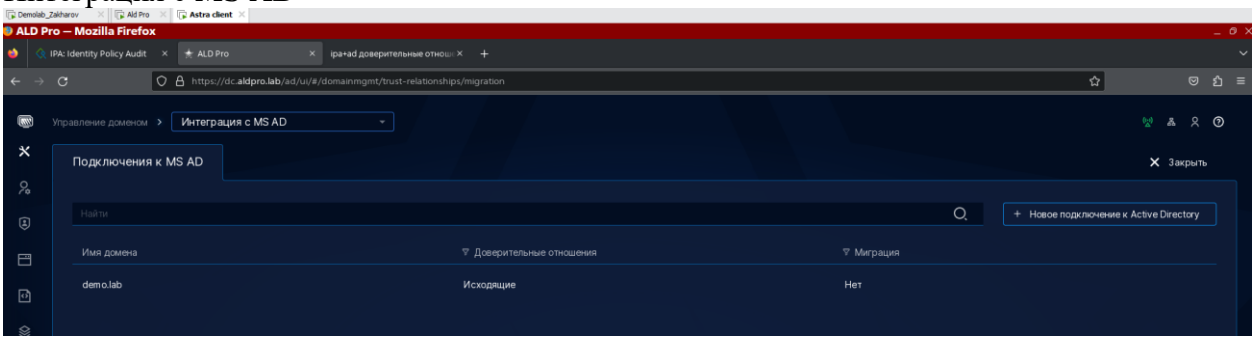
Нажимаем добавить, после чего операция закончится ошибкой, закрываем окно отношений.
Далее переходим в терминал астра, где заходим под доменным пользователем с помощью команды kinit
kinit admin
и вводим пароль. Следующей командой вводим:
ipa trust-add --type=ad demo.lab --admin Administrator --password
где demo.lab название вашего AD домена

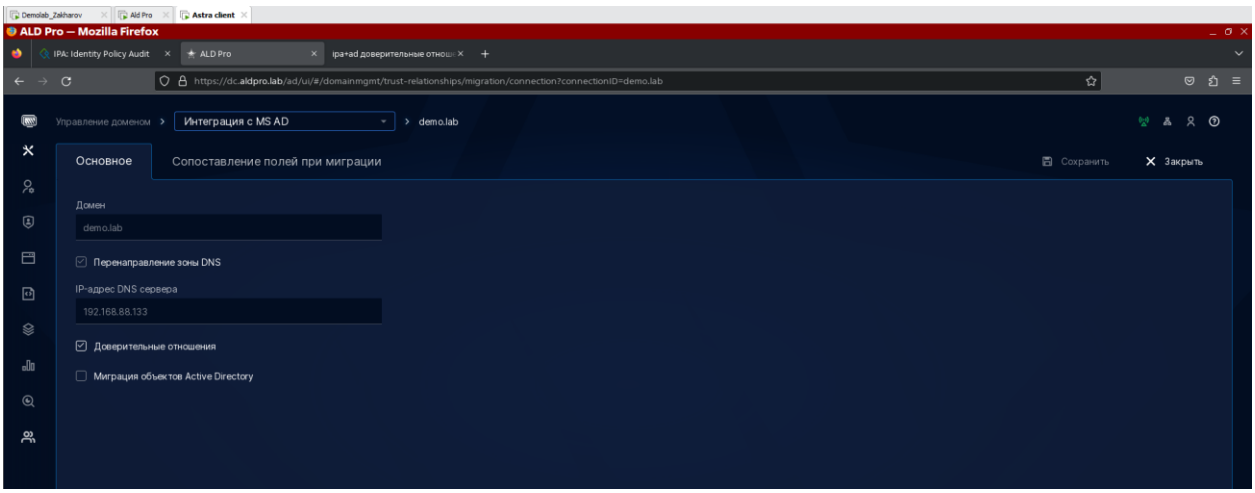


После этого автоматически создается запись в веб интерфейсе IPA.



И автоматическая запись в интерфейсе управления ALDPRO. Управление доменом > Интеграция с MS AD





Практическая работа № 35 Настройка политик безопасности ALD PRO

Задание:

В настроенном домене ALD PRO перенесите все ранее установленные политики AD (которые возможно реализовать)

Зафиксируйте процесс снимками экрана.

Практическая работа № 36 Сравнительный анализ средств управления сетью

Задание:

Произведите сравнительный анализ 5 средств управления сетью по критериям:

14. Автообнаружение
15. Ограничение по числу промежуточных маршрутизаторов
16. Возможность модификации присвоенного имени хоста
17. Определение имени хоста по его адресу через сервер DNS
18. Максимальное рекомендуемое число обслуживаемых узлов
19. Распознавание сетевых топологий
20. Распределенное управление
21. Поддержка баз данных
22. Собственная карта сети у клиента
23. Поддержка протокола SNMP через другие протоколы
24. Поддержка ОС

Оформите отчет в удобной форме.

Практическая работа № 37 Управления сетью с помощью межсетевого экрана

Задание:

На виртуальной машине «Сервер» настроить межсетевой экран по схеме 4 «Усиленная защита сервера».

Подготовка стенда

1.На виртуальную машину «Сервер» установим OpenSSH сервер и службу автоматической синхронизации системных часов systemd-timesyncd:

```
apt install ssh systemd-timesyncd
```

2.В файле настроек службы синхронизации времени /etc/systemd/timesyncd.conf раскомментируем строки NTP и FallbackNTP. Строку NTP запишем в следующем виде:

```
NTP=0.europe.pool.ntp.org 1.europe.pool.ntp.org 2.europe.pool.ntp.org
3.europe.pool.ntp.org
```

3.Активируем автоматическую синхронизацию времени, выполнив заклинание:

```
timedatectl set-ntp true
systemctl enable --now systemd-timesyncd.service
systemctl restart systemd-timesyncd.service
```

4.Для проверки синхронизации времени нужно перезагрузиться, а затем выполнить заклинание:

```
timedatectl timesync-status
```

5.Настроим статический IP адрес на виртуальной машине «Сервер». Для этого содержимое файла /etc/network/interfaces заменим на следующее:

```
auto lo
iface lo inet loopback
# The primary network interface
auto ens33
iface ens33 inet static
address 172.30.0.40
mask 255.255.255.0
gateway 172.30.0.2
nameserver 172.30.0.2
```

Примечание. В Debian по умолчанию при описании сетевых интерфейсов используется ключевое слово «allow-hotplug». Мы же поменяем его на «auto». Это сделано для того, чтобы была возможность менять сетевые настройки с помощью заклинания:

```
service networking restart
```

6.Проверим работоспособность стенда. С гипервизора должна быть возможность установить SSH сессию до «Сервера», а с «Сервера» — возможность скачивать обновления с официальных репозиториев и обновлять время по сети.

Ход выполнения работы

1.Логика организации дистрибутива Debian такова, что межсетевой экран nftables установлен в нем по умолчанию. В этом можно убедиться, выполнив заклинание:

```
nft -v
```

2.Для автоматической загрузки правил межсетевого экранирования после старта системы создана служба nftables (по сути являющаяся systemd юнитом), но по умолчанию она деактивирована. В этом можно убедиться, выполнив заклинание:

```
service nftables status
```

Служба организована таким образом, что при запуске она считывает правила, записанные в файле /etc/nftables.conf. Соответственно, цель нашей работы — внести в данный файл необходимые правила и активировать службу.

3.Уточним схему разграничения сетевого трафика. Как часто бывает, те схемы, которые мы описывали ранее, не могут учитывать особенности эксплуатации конкретных серверов, поэтому и требуют уточнения. В частности, сделаем следующее:

- Разрешим «пинговать» (использовать команду проверки сетевой связности ping) защищаемый сервер и разрешим серверу «пинговать» другие узлы. Несмотря на то, что каждый открытый поток сетевого трафика снижает защищенность, открытие «пингов» существенно улучшает эксплуатационные свойства сервера.
  Для этого разрешим входящие и исходящие ICMP echo-request.
- Разрешим отправлять запросы и получать ответы по протоколу DNS.
  Для этого разрешим исходящие соединения по портам: TCP 53 и UDP 53 в адрес явно указанного DNS сервера: 172.30.0.2.
- Разрешим автоматически обновлять системные часы по протоколу NTP.
  Для этого разрешим исходящие соединения по портам: TCP 123 и UDP 123 в адрес серверов, указанных в файле /etc/systemd/timesyncd.conf.
- Разрешим скачивать обновления с репозиториев, указанных в файлах настройки менеджеров пакетов.
  Для этого разрешим исходящие http соединения (TCP 80) в адрес репозиториев, указанных в файле /etc/apt/sources.list.
- Будем явно отбрасывать входящий трафик, превышающий скоростные ограничение в:
  - 5 пакетов в секунду для ICMP;
  - 10 попыток установки соединений в минуту для SSH.

•Узлы, занимающиеся подборкой паролей к SSH, будем определять с помощью утилиты fail2ban. Утилита сама будет конфигурировать nftables для блокирования и разблокирования атакующих узлов.

4.В файл /etc/nftables.conf запишем следующее содержимое:

```
#!/usr/sbin/nft -f
flush ruleset
table ip firewall {
# Список разрешенных DNS серверов
        set allowed-dns-servers {
                type ipv4_addr
                elements = { 172.30.0.2 }
        }
# Список разрешенных узлов, с которых можно подключаться по SSH
        set allowed-ssh-clients {
                type ipv4_addr
                elements = { 172.30.0.1 }
        }
# Список разрешённых NTP-серверов из файла /etc/systemd/timesyncd.conf
        set allowed-ntp-servers {
                type ipv4_addr
        }
# Список разрешённых серверов репозиториев из файла /etc/apt/sources.list
        set allowed-repos {
                type ipv4_addr
        }
# Цепочка правил фильтрации входящего трафика. Запрещено все, кроме того, что
# явно разрешено (policy drop)
        chain fw_input {
                type filter hook input priority filter; policy drop;
# Разрешен трафик с петлевого (loopback) интерфейса
                iifname "lo" accept
# Явно отбрасываются IP-пакеты с обратным адресом локальной петли,
# но не относящиеся к петлевому интерфейсу
                ip saddr 127.0.0.0/8 drop
# Явно отбрасываем ICMP трафик, превышающий скоростной лимит
                ip protocol icmp limit rate over 5/second drop
# Явно отбрасываем запросы на соединение по SSH, превышающие скоростной лимит
                tcp dport 22 ct state new limit rate over 10/minute drop
# Разрешаем трафик по уже установленным соединениям
                ct state established,related accept
# Разрешаем получение ICMP-эхо запросов (чтобы узел можно было пингануть)
                icmp type echo-request accept
# Разрешаем подключение по SSH избранным клиентам
                ip saddr @allowed-ssh-clients tcp dport 22 accept
        }
# Цепочка правил фильтрации исходящего трафика. Запрещено все, кроме того, что
# явно разрешено (policy drop)
        chain fw_output {
                type filter hook output priority filter; policy drop;
# Разрешаем трафик на петлевой интерфейс
                oifname "lo" accept
# Разрешаем трафик по уже установленным соединениям
                ct state established,related accept
# Разрешаем отправку ICMP эхо-запросов (чтобы узел мог пингануть).
                icmp type echo-request accept
# Разрешаем отправку DNS-запросов по UDP
                ip daddr @allowed-dns-servers udp dport 53 accept
# Разрешаем отправку DNS-запросов по TCP
                ip daddr @allowed-dns-servers tcp dport 53 accept
# Разрешаем отправку запросов по протоколу NTP с помощью TCP
```

```
                ip daddr @allowed-ntp-servers tcp dport 123 accept
# Разрешаем отправку запросов по протоколу NTP с помощью UDP
                ip daddr @allowed-ntp-servers udp dport 123 accept
# Разрешаем установке HTTP-соединений с серверами репозиториев, указанными в
файле
# /etc/apt/sources.list
                ip daddr @allowed-repos tcp dport 80 accept
        }
}
```

Примечание. Опытные администраторы наверно заметили, что в цепочке fw_input мы явно отбрасываем icmp пакеты, превышающие установленный скоростной лимит (ip protocol icmp limit rate over 5/second drop), а затем пишем правило, разрешающее получать icmp эхо-запросы (icmp type echo-request accept). Резонный вопрос: зачем мы это делаем? Ведь политика цепочки — отбрасывать все, что явно не разрешено, и, по идее, кажется, что эти два правила можно заменить одним icmp type echo-request limit rate 5/second accept, но в данном случае это не так. Дело в том, что в цепочке присутствует правило ct state established,related accept, и из-за него icmp type echo-request limit rate 5/second accept не будет ограничивать скорость. Это происходит потому, что nftables считает «ping» как сетевое соединение. Это можно увидеть с помощью заклинания

```
conntrack -L
```

Соответственно, пакеты, не попавшие в правило icmp type echo-request limit rate 5/second accept, будут пропущены правилом ct state established,related accept. Вот поэтому и нужно явно отбрасывать пакеты, превышающие скорость, и делать это перед правилом ct state established,related accept.

5.Активируем автоматический запуск nftables после загрузки системы. Для этого выполним заклинания:

```
systemctl daemon-reload
systemctl enable nftables
```

6.Для дальнейших работ нам потребуется утилита dig, входящая в пакет dnsutils, и утилита fail2ban, входящая в одноименный пакет. Установим их, выполнив заклинание:

```
apt install dnsutils fail2ban
```

7.Рассмотрим процесс формирования динамических списков allowed-ntp-servers и allowed-repos. Данные списки должны содержать в себе IP-адреса:

- репозиториев, указанных в файле /etc/apt/sources.list;
- NTP-серверов, указанных в файле /etc/systemd/timesyncd.conf.

Если вы просмотрите эти файлы, то никаких IP-адресов там не заметите. Вместо них указаны лишь FQDN-имена требуемых серверов. Проблема в том, что nftables не умеет фильтровать по FQDN-именам, он работает только по IP. Соответственно, необходимо, чтобы кто-то ему перевел из FQDN в IP. Также следует помнить, что FQDN-имена — вещь не постоянная, и процесс перевода их в IP нужно делать периодически. Для извлечения и перевода из данных файлов FQDN в IP-адреса можно воспользоваться скриптом:

```
nft flush set ip firewall allowed-ntp-servers
nft flush set ip firewall allowed-repos
grep -oP '(?<=^deb http://)[^ /]*' /etc/apt/sources.list | uniq  | xargs dig
+short | xargs -r -I IP nft add element ip firewall allowed-repos { IP }
grep -oP '(?<=^NTP=).+$' /etc/systemd/timesyncd.conf | xargs dig +short | xargs
-r -I IP nft add element ip firewall allowed-ntp-servers { IP }
grep -oP '(?<=^FallbackNTP=).+$' /etc/systemd/timesyncd.conf | xargs dig +short
| xargs -r -I IP nft add element ip firewall allowed-ntp-servers { IP }
```

Алгоритм работы данного скрипта заключается в том, что вначале очищаются соответствующие списки nftables, затем с помощью регулярных выражений из файлов извлекаются FQDN-имена серверов, которые с помощью утилиты dig преобразуются в IP-адреса, а затем добавляются к требуемым спискам.

8.Проблема в том, что приведенный выше скрипт нужно выполнять периодически. Для ее решения преобразуем скрипт в systemd юнит /etc/systemd/system/nft-dns.service:

```
# /etc/systemd/system/nft-dns.service
[Unit]
Description=nftables DNS resolve service
Requires=nftables.service
Wants=nft-dns.timer
After=network.target
[Service]
Type=oneshot
ExecStart=bash -c "nft flush set ip firewall allowed-ntp-servers ; nft flush
set ip firewall allowed-repos ; grep -oP '(?<=^deb http://)[^ /]*'
/etc/apt/sources.list | uniq  | xargs dig +short | xargs -r -I IP nft add
element ip firewall allowed-repos { IP } ; grep -oP '(?<=^NTP=).+$'
/etc/systemd/timesyncd.conf | xargs dig +short | xargs -r -I IP nft add element
ip firewall allowed-ntp-servers { IP } ; grep -oP '(?<=^FallbackNTP=).+$'
/etc/systemd/timesyncd.conf | xargs dig +short | xargs -r -I IP nft add element
ip firewall allowed-ntp-servers { IP } "
StandardOutput=journal
[Install]
WantedBy=multi-user.target
```

9.Затем для ежечасного запуска этого юнита создадим systemd таймер /etc/systemd/system/nft-dns.timer:

```
# /etc/systemd/system/nft-dns.timer
```

```
[Unit]
Description=nftables DNS resolve timer
Requires=nftables.service
[Timer]
Unit=nft-dns.service
OnCalendar=hourly
[Install]
WantedBy=timers.target
```

10.После создания юнита и таймера активируем их, выполнив:

```
systemctl daemon-reload
systemctl enable nft-dns.service
systemctl enable nft-dns.timer
```

11.Последним этапом данной задачи будет настройка fail2ban на анализ журналов работы OpenSSH сервера и блокирование всех тех, кто совершил большое количество неудачных попыток авторизации по ssh.

На самом деле fail2ban по умолчанию защищает SSH-сервер сразу после установки. Единственное, что необходимо сделать, так это поменять действия, выполняемые при блокировке. По умолчанию они рассчитаны на iptables (предшественника nftables), нам же требуется их поменять для nftables.

Сделать это очень просто. В конфигурационном файле /etc/fail2ban/jail.conf в секции [DEFAULT] значение параметров banaction и banaction_allports необходимо поменять на nftables-multiport и nftables-allports соответственно. Затем перезапустить службу заклинанием

```
service fail2ban restart
```

Для блокировки нарушителей fail2ban добавляет к правилам фильтрации nftables новую таблицу f2b-table. Эта таблица содержит единственную цепочку f2b-chain, имеющую более низкий приоритет и соответственно срабатывающую раньше, чем тем цепочки, что мы создавали в файле /etc/nftables.com. Единственным правило цепочки f2b-chain является блокировка доступа к порту ssh (tcp 22) для IP-адресов, включенных в список addr-set-sshd. Пример рассмотренных списков и правил фильтрации, при добавлении туда нарушителя, выглядит следующим образом :

Текущее состояние блокировок можно посмотреть, выполнив :

```
fail2ban-client status sshd
```

Практическая работа № 38 Установка Ansible

Задание:

Настройка инвентарного файла

В данном файле хранится информация о хостах и/или группах хостов. Также в нем могет храниться переменные, определенные для конкретной группы хостов или конкретного компьютера.

Есть два варианта написания инвентарного файла — в формате yml или ini. Рассмотрим оба.

1. Файл формата ini.

Даннай формат используется по умолчанию. Откроем на редактирование файл с серверами, которыми хотим управлять:

vi /etc/ansible/hosts

и приведем его к следующему виду:

[test_servers]
192.168.1.100
192.168.1.101

** в данном примере создана группа серверов **test_servers,** в которую добавлены два сервера с IP-адресами **192.168.1.100** и **192.168.1.101.***

2. Файл формата yml.

Создадим отдельный каталог:

mkdir /etc/ansible/inventory

И создадим файл:

vi /etc/ansible/inventory/test_servers.yml

```
test_servers:
 vars:
  ansible_python_interpreter: /usr/bin/python3
 hosts:
  server01:
   ansible_ssh_host: 192.168.1.100
   ansible_ssh_port: 22
  server02:
```

ansible_ssh_host: 192.168.1.101
ansible_ssh_port: 22

** в данном примере также создана группа серверов **test_servers,** в которую добавлены два сервера **server01** и **server02** с IP-адресами **192.168.1.100** и **192.168.1.101**. Адреса не обязательно писать, если имя машины разрешается с помощью DNS. Также не обязательно указывать порты, если они стандартные (22).*
*** обратите внимание, что мы также добавили переменную **ansible_python_interpreter** с указанием пути для запуска python.*

Настройка ansible

Открываем конфигурационный файл ansible:

vi /etc/ansible/ansible.cfg

Для более новых версий ansible конфигурационный файл не создается. Создаем для него каталог и сам файл:

mkdir /etc/ansible

vi /etc/ansible/ansible.cfg

Снимаем комментарий с опции **host_key_checking**, приведя ее к виду:

host_key_checking = False

** данная настройка позволит нашему серверу управления автоматически принимать ssh fingerprint, избавляя нас от необходимости постоянно вводить **yes**, когда мы впервые конфигурируем новый сервер.*

Также в секцию **defaults** добавим:

[defaults]
...
interpreter_python = auto_silent

** данная опция указывает, чтобы ansible автоматически искал python на целевом хосте без показа предупреждений.*

Тестовый запуск

Теперь выполним проверку доступности добавленных серверов:

ansible -m ping test_servers -u root -kK

** данная команда проверит доступность по сети двух серверов из группы **test_servers** от учетной записи **root**.*

Если мы создали и хотим использовать инвентарный файл yml, то вводим:

ansible -i /etc/ansible/inventory/test_servers.yml -m ping test_servers -u root -kK

** мы должны указать путь до файла inventory с помощью опции -i.*

Будет запрошен пароль от учетной записи (в нашем случае, root). После будет запрошен пароль суперпользователя на серверах.

На экране должно появиться, примерно, следующее:

```
    192.168.1.100 | SUCCESS => {
  "ansible_facts": {
     "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
192.168.1.101 | SUCCESS => {
  "ansible_facts": {
     "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
```

Наш сервер управления готов к работе.

Подключение без пароля

В нашем примере аутентификация на узлах выполняется с помощью пароля. Это не очень удобно и безопасно. Рассмотрим вариант использования ssh ключа.

На компьютере с ansible сгенерируем пару ключей следующей командой:

```
ssh-keygen -t ed25519
```

После нажатия **Enter** система попросит ввести параметры размещения ключа и пароль. Ничего не меняем, нажимая ввод и соглашаясь со значениями по умолчанию.

Мы увидим что-то на подобие:

```
    Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_ed25519
Your public key has been saved in /root/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:FTDGR6Gz92WI33/ywMSImcqfx2IRQjrD20tE/qzqZ5Y root@ansible.dmosk.ru
The key's randomart image is:
+--[ED25519 256]--+
|      .+o+.     |
|      .+o..     |
|    . =o..      |
|     = ++.= +   |
|      *S+*.o =  |
|     ..oo+o *   |
```

```
|      .o+ oo +  |
|       E.o.o .o.|
|    .o= .oo   o+|
+----[SHA256]-----+
```

** обратите внимание, что мы сгенерировали ключи под пользователем root и из местоположение в каталоге **/root/.ssh**.*

Теперь скопируем публичный ключ на все серверы, к которым будем подключаться:

ssh-copy-id -i /root/.ssh/id_ed25519.pub root@192.168.1.100

ssh-copy-id -i /root/.ssh/id_ed25519.pub root@192.168.1.101

** в нашем примере мы копируем наш ключ для пользователя **root** на удаленной системе.*

Мы также можем скопировать ключ вручную. Для этого содержимое публичного файла (в нашем случае, **id_ed25519.pub**) добавить в файл **authorized_keys** для нужного пользователя. Например:

mkdir /root/.ssh

vi /root/.ssh/authorized_keys

** в данном примере мы сможем подключиться к компьютеру с нашим сертификатом под пользователм **root**.*

Готово. Попробуем подключиться по SSH без пароля к любому из серверов:

ssh root@192.168.1.100

Мы должны подключиться к серверу без запроса пароля.

Теперь можно попробовать запустить наш ансибл без ввода паролей:

ansible -m ping test_servers -u root

Или с использованием инвентарного файла yml:

ansible -i /etc/ansible/inventory/test_servers.yml -m ping test_servers -u root

Команда должна выполниться без запросов пароля.

Практическая работа № 39 Использование Ansible для установки пакетов, модулей и расширений

Задание:

1. Установка/удаление пакетов в систему.

Выполняется с помощью универсального модуля **package** или с помощью специализированных **yum** и **apt**. Последние имеют больше полезных опций.

а) package:

```
- name: Install NTP-client
  package:
    name: chrony
    state: latest
```

** по данной инструкции в нашей системе должен быть установлен пакет **chrony** последней версии.*

Для установки пакета конкретной версии, ее нужно указать в названии пакета:

```
- name: Install NTP-client
  package:
    name: chrony-3.4
    state: present
```

О package: https://docs.ansible.com/ansible/latest/collections/ansible/builtin/package_module.html.

б) yum.

Установка:

```
- name: Install NTP-client
  yum:
    name: chrony
    state: present
```

Для удаления:

```
- name: Remove NTP-client
  yum:
    name: chrony
    state: absent
```

А так можно сделать даунгрейд:

```
- name: Downgrade NTP-client
  yum:
    name: chrony-1.0.0
    state: present
    allow_downgrade: true
```

О yum: https://docs.ansible.com/ansible/latest/collections/ansible/builtin/yum_module.html.

в) apt.

Установка:

```
- name: Install NTP-client
  apt:
```

```
    name: chrony
    state: present
```

Если нам нужно установить пакет из файла deb, синтаксис будет следующий:

```
    - name: "Install lib for postgresql"
  apt:
    deb: http://ftp.ru.debian.org/debian/pool/main/l/llvm-toolchain-7/libllvm7_7.0.1-
8+deb10u2_amd64.deb
    state: present
```

** в данном примере мы устанавливаем пакет **libllvm7** из файла deb, который доступен по url **http://ftp.ru.debian.org/debian/pool/main/l/llvm-toolchain-7/libllvm7_7.0.1-8+deb10u2_amd64.deb**.*

Об apt: https://docs.ansible.com/ansible/latest/collections/ansible/builtin/apt_module.html.

2. Работа с репозиториями RPM.

Ведется с помощью модуля yum_repository.

О yum_repository: https://docs.ansible.com/ansible/latest/collections/ansible/builtin/yum_repository_module.html.

а) Добавление репозитория:

```
    - name: "Add YUM repository"
  yum_repository:
    name: new_yum_repo
    description: "New Yum Repo Description"
    file: new_repo_file
    baseurl: https://download.fedoraproject.org/pub/epel/$releasever/$basearch/
    enabled: yes
    gpgcheck: no
    validate_certs: no
```

б) Для удаления репозитория также используем yum_repository:

```
    - name: "Remove YUM repositories"
  yum_repository:
    name: pgsql95
    state: absent
```

** в нашем примере будет удален репозиторий с названием **pgsql95**.*

Ansible может не распознавать некоторые репозитории, которые настроена в пользовательских файлах. В таком случае необходимо указать на эти файлы с помощью опции file:

```
    - name: "Remove YUM repositories"
  yum_repository:
    name: pgsql95
```

```
    state: absent
    file: pgsql
```

в) Установка пакетов из dnf-модулей. Начиная с версии 8 дистрибутивов на основе RPM пакеты могут устанавливаться из модульных репозиториев, в которых идет четкое разграничение между версиями пакетов. В ansible, если необходимо установить пакет из определнного модуля, используем сценарий на подобие:

```
    - name:
  dnf:
    name:
      - @postgresql:13/client
      - @postgresql:13/server
    state: present
```

** в данном примере мы установим пакеты postgresql-client и postgresql-server из модульного репозитория postgresql:13.*

3. Работа с репозиториями DEB.

а) Добавление репозитория:

```
    - name: "Add DEB repository"
  apt_repository:
    repo: deb http://dl.google.com/linux/chrome/deb/ stable main
    filename: new_deb_repo
```

О apt_repository: https://docs.ansible.com/ansible/latest/collections/ansible/builtin/apt_repository_module.html.

б) Для удаления репозитория нужно воспользоваться директивой **state** со значением **absent**, например:

```
    - name: "Remove Debian default repository"
  apt_repository:
    repo: deb https://ftp.debian.org/debian stretch main contrib
    state: absent
```

** в данном примере будет удален репозиторий, который идет по умолчанию для Debian Stretch.*

в) Импорт ключа.

Рассмотрим примеры с добавлением gpg-ключа по URL и с сервера ключей:

```
     - name: Import postgresql repo key
    apt_key:
      url: https://www.postgresql.org/media/keys/ACCC4CF8.asc
      state: present
```

** импорт ключа из файла* ***https://www.postgresql.org/media/keys/ACCC4CF8.asc***.

```
- name: Import keyserver.ubuntu.com repo keys
  apt_key:
    keyserver: keyserver.ubuntu.com
    id: 648ACFD622F3D138
    state: present
```

** импорт ключа с идентификатором **648ACFD622F3D138** из сервера ключей **keyserver.ubuntu.com**.*

О apt_key: https://docs.ansible.com/ansible/latest/collections/ansible/builtin/apt_key_module.html.

г) Обновление кэша репозитория.

Выполняется с помощью модуля apt и опции update_cache:

```
- name: Update repositories cache
  apt:
    update_cache: yes
```

Об apt: https://docs.ansible.com/ansible/latest/collections/ansible/builtin/apt_module.html.

4. Установка модуля в nodejs.

Установка модулей в nodejs выполняется с помощью npm. Для него в ansible есть отдельная функция:

```
- name: Install nodejs modules.
  npm:
    name: newman
    global: yes
```

** в данном примере будет выполнена установка **newman**, которая будет доступна всем проектам (опция **global**).*

О nodejs npm: https://docs.ansible.com/ansible/latest/collections/community/general/npm_module.html.

5. Установка расширений для python.

Используем модуль pip. Рассмотрим несколько примеров.

а) установка пакета python:

```
- name: Pip install psycopg2
  pip:
    name: psycopg2
    state: present
```

** в данном примере будет установлен **psycopg2**.*

б) обновление пакетов:

```
    - name: Upgrade pip and wheel
  pip:
    name: "{{ item }}"
    extra_args: --upgrade
    executable: pip3
  loop:
    - pip
    - wheel
```

** в нашем примере будут обновлены пакеты **pip** и **wheel**.*

в) использовать определенную версию pip:

```
    - name: Install python modules with pip3
  pip:
    name: patroni[consul]
    executable: pip3
    state: present
```

О pip: https://docs.ansible.com/ansible/latest/collections/ansible/builtin/pip_module.html.

6. Распаковка архива.

Выполняется с помощью **unarchive**:

```
    - name: Unpacking Nginx Source
  unarchive:
    src: "http://nginx.org/download/nginx-{{ nginx_ver }}.tar.gz"
    dest: /tmp/
    remote_src: yes
    creates: /tmp/nginx-{{ nginx_ver }}.tar.gz
```

** в данном примере мы распакуем исходник для nginx в каталог **/tmp**. Обратите внимание на две вещи:*

- *Мы используем переменную **nginx_ver**. Данная переменная должна быть определена при запуске плейбука, или в инвентарном файле, или в var, или в default. Подробнее в соответствующем разделе выше.*
- *Опция **creates** позволит не выполнять операцию, если существует файл **/tmp/nginx-{{ nginx_ver }}.tar.gz**.*

Практическая работа № 40 Использование Ansible для управления системой

Задание:

1. Добавить задание в cron.

Выполняется с помощью модуля cron:

```
    - name: Add Job for Run Command
  cron:
    name: Start Script
    job: "/scripts/command.sh"
    user: root
    minute: "0"
    hour: "*/6"
    day: "*"
    month: "*"
    weekday: "*"
```

** в данном примере мы создадим задание для запуска команды **/scripts/command.sh** каждый день, каждые 6 часов.*

О cron: https://docs.ansible.com/ansible/latest/collections/ansible/builtin/cron_module.html.

2. Создание учетной записи.

Для этого есть модуль user. У него много опций, рассмотрим некоторые из них.

а) Простая учетная запись:

```
    - name: Create User1
  user:
    name: user1
    shell: /bin/bash
    create_home: yes
```

** в данном примере мы создадим пользователя **user1** с домашней директорией. Также мы указали для использования командную оболочку **/bin/bash**.*

б) Для создания системной учетной записи нам достаточно:

```
    - name: Create User Consul
  user:
    name: consul
    system: yes
    comment: "Consul Agent"
```

** в данном примере будет создана учетная запись **consul**.*

в) Создаем пользователя с паролем:

```
    - name: Create User2
  user:
    name: user2
    shell: /bin/bash
    create_home: yes
    password: "{{ 'my_passw0rd' | password_hash('sha512') }}"
```

** будет создан пользователь **user2** с паролем **my_passw0rd**.*

г) Добавляем пользователя в группу:

```
- name: Добавляем пользователя clamav в группу amavis
  user:
    name: vmail
    groups: mail
    append: yes
```

** в данном примере пользователь **vmail** будет добавлен в группу **mail**.*

О user: https://docs.ansible.com/ansible/latest/collections/ansible/builtin/user_module.html.

д) Создание группы:

```
- name: Создаем группу vmail
  group:
    name: vmail
    state: present
```

О group: https://docs.ansible.com/ansible/latest/collections/ansible/builtin/group_module.html.

3. Работа с systemd.

Для данной настройки есть одноименный модуль systemd. Рассмотрим варианты его использования.

а) перечитать конфигурацию (необходимо делать каждый раз, когда мы меняем настройки юнита):

```
- name: systemd reload
  systemd:
    daemon_reload: yes
```

б) разрешить сервис (автозапуск):

```
- name: mysql enable
  systemd:
    name: mysql
    enabled: yes
```

** для сервиса **mysql**.*

в) перезапустить сервис:

```
- name: mysql reload
  systemd:
    name: mysql
    state: restarted
```

г) остановить сервис:

```
- name: mysql stoped
  systemd:
```

```
  name: mysql
  state: stopped
```

О systemd: https://docs.ansible.com/ansible/latest/collections/ansible/builtin/systemd_module.html.

4. Настройка брандмауэра.

Выполняется разными модулями в зависимости от используемой системы управления netfilter:

- firewalld
- iptables
- ufw

Рассмотрим небольшие примеры.

а) firewalld:

```
- name: permit traffic in default zone for https service
  firewalld:
  service: https
  permanent: yes
  state: enabled
```

Подробнее: https://docs.ansible.com/ansible/latest/collections/ansible/posix/firewalld_module.html.

б) iptables:

```
- name: Block specific IP
  iptables:
  chain: INPUT
  source: 8.8.8.8
  jump: DROP
```

Подробнее: https://docs.ansible.com/ansible/latest/collections/ansible/builtin/iptables_module.html.

в) UFW.

Добавить 80 порт:

```
- name: Allow all access to tcp port 80
  ufw:
  rule: allow
  port: '80'
  proto: tcp
```

Добавить порты с циклом:

```
    - name: Allow Ports in Firewall
  ufw:
    rule: allow
    port: "{{ item.port }}"
    proto: "{{ item.proto }}"
    comment: "{{ item.comment }}"
  loop:
    - { port: 5432, proto: tcp, comment: 'PostgreSQL' }
```

Подробнее: https://docs.ansible.com/ansible/latest/collections/community/general/ufw_module.html.

5. Имя компьютера.

Для указания имени компьютера можно использовать модуль hostname:

```
    - name: Задаем имя компьютера
  hostname:
    name: myweb
    use: systemd
```

** в данном примере мы задаем имя **myweb**. Обратите внимание на опцию **use** — в зависимости от операционной системы или ее версии, ее значение должно отличаться.*

О hostname: https://docs.ansible.com/ansible/latest/collections/ansible/builtin/hostname_module.html.

6. Часовой пояс.

Часовой пояс можно настроить с помощью модуля **timezone**, например:

```
    - name: Задаем часовой пояс
  timezone:
    name: Europe/Moscow
```

О timezone: https://docs.ansible.com/ansible/latest/collections/community/general/timezone_module.html.

Практическая работа № 41 Использование Ansible для работы с файлами

Задание:

Изучите задачи, которые помогут создавать, копировать и работать с файлами.

1. Создание каталогов и файлов.

Создание файлов и каталогов выполняется с помощью модуля **file**.

а) для каталога в качестве **state** указываем **directory**:

```
    - name: Create Directories
  file:
    path: "{{ item }}"
    state: directory
    owner: www-data
    group: www-data
    mode: 0755
  loop:
    - '/var/www/site1'
    - '/var/www/site2'
```

** в данном примере мы создадим 2 каталога: **site1** и **site2** в каталоге **/var/www**.*

б) для создания файла убираем опцию **state** (или даем ей значение **touch**):

```
    - name: Create File
  file:
    path: "/var/www/site1/index.php"
    state: touch
    owner: www-data
    group: www-data
    mode: 0644
```

** в данном примере мы созданим файл **index.php** в каталоге **/var/www/site1**.*

в) для создания симлинка используем **state** со значением **link**:

```
    - name: Create a symbolic link from foo to bar
  file:
    src: /usr/bin/foo
    dest: /usr/sbin/bar
    state: link
```

О file: https://docs.ansible.com/ansible/latest/collections/ansible/builtin/file_module.html.

2. Задать права.

Это можно выполнить с помощью модуля создания файла или каталога:

```
    - name: Set File Rights
  file:
    path: "/var/www/site1/index.php"
    owner: www-data
    group: www-data
    mode: 0664
```

** обратите внимание, это пример из предыдущено раздела. Для созданного файла мы просто немного изменили права.*

3. Копирование файлов из каталога.

Для копирования данных мы используем модуль copy:

```
    - name: Copy Cert File If Different
  copy:
    src: "{{ item }}"
    dest: /etc/ssl/dmosk
    remote_src: no
    mode: 0644
    owner: root
    group: root
  with_fileglob:
    - files/*
```

** в данном примере мы прочитаем все содержимое каталога **files** на компьютере с ansible, и скопируем его в каталог **/etc/ssl/dmosk** на целевом компьютере.*

О copy: https://docs.ansible.com/ansible/latest/collections/ansible/builtin/copy_module.html.

4. Используем шаблон.

Копирование из шаблона отличается от копирования из файла тем, что в шаблоне могут использоваться переменные, которые будет заменяться их значениями в момент копирования. Для самого процесса копирования из шаблона используется модуль **template**:

```
    - name: Create Config for Consul Agent
  template:
    src: templates/consul/config.json.j2
    dest: /etc/consul.d/config.json
```

** в данном примере мы возьмом шаблон **templates/consul/config.json.j2** на компьютере ansible и разместим его в по пути **/etc/consul.d/config.json** на целевом компьютере.*

Мы можем вывести в консоль результат обработки шаблона следующим образом:

```
    - name: Show Templating Results
  debug:
    msg: "{{ lookup('template', './config.json.j2') }}"
```

О template: https://docs.ansible.com/ansible/latest/collections/ansible/builtin/template_module.html.

5. Архивирование.

Создать архив из файла или каталога можно с помощью модуля archive:

```
    - name: "Use gzip to compress folder"
  archive:
    path: /etc/raddb
    dest: "/tmp/raddb.gz"
    format: gz
```

** в данном примере мы создадим архив из каталога **/etc/raddb** и сохраним его в файл **/tmp/raddb.gz**.*

О archive: https://docs.ansible.com/ansible/latest/collections/community/general/archive_module.html.

Для распаковки архивов используется модуль unarchive, о котором мы говорили выше.

6. Поиск файлов и папок.

Выполняется с помощью модуля find. Особый интерес представляет в контексте поиска файлов и выполнение над ними определенных действий. Разберем несколько примеров.

а) Удалить последние 30 файлов. Задача решается в два этапа:

- ищем содержимое целевого каталога.
- сотритуем список найденных по времени изменения файлов и удаляем все, что идут после определенного числа объектов.

Поиск выполняем с помощью модуля **find**, удаление — **file**:

```
- name: "Get list of backup files"
  find:
    paths: "/backup"
    file_type: file
  register: founds

- name: "Delete last 30 Copies"
  file:
    path: "{{ item }}"
    state: absent
  loop: "{{ (founds.files | sort(attribute='mtime', reverse=True) | map(attribute='path') | list )[30:] }}"
```

** в данном примере мы ищем файлы в каталоге **/backup**, после чего сортируем найденное и удаляем по списку все файлы, которые идут после 30-го.*

б) Удаление архивов для логов. Также выполняем в два этапа — более сложный поиск и удаление с помощью file:

```
- name: "Get a list of logs to be deleted"
  find:
    paths: "/var/log"
    file_type: file
    patterns: '*.gz,*.log-*,*.old,*.[0-9].log,*.log.[0-9],*-[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'
    recurse: yes
  register: logs_to_delete

- name: "Delete logs"
  file:
    path: "{{ item }}"
    state: absent
  loop: "{{ logs_to_delete.files | map(attribute='path') | list }}"
```

** в данном примере мы применили регулярные выражения для поиска различных вариантов логов, которые являются архивными. После мы сохраняем результат поиска в переменную **logs_to_delete**, которую используем для получения списка путей до найденных файлов.*

О find: https://docs.ansible.com/ansible/latest/collections/ansible/builtin/find_module.html.

7. Скачать файл с помощью curl.

Для этого используется модуль uri. Простой пример:

```
- name: CURL simple download file
  uri:
    url: https://www.dmosk.ru/files/winsetupfromusb.zip
    dest: /tmp
```

** в данном примере мы загрузим файл **https://www.dmosk.ru/files/winsetupfromusb.zip** в каталог **/tmp**.*

Пример посложнее:

```
- name: CURL download file with token auth
  uri:
    url:
https://gitlab.dmosk.ru/api/v4/projects/555/repository/files/folder%2Fpath%2Fdata.sql/raw?ref=m
aster
    dest: /tmp/data.sql
    owner: dmosk
    group: dmosk
    mode: 640
    headers:
      PRIVATE-TOKEN: access-token
```

** в данном примере мы скачаем файл с ресурса, где требуется аутентификация по токену, который передается в заголовке. Заголовки мы передаем с помощью параметра **headers**. Также мы задаем права на загруженный файл и делаем в качестве владельца пользователя и группу **dmosk**.*

Об uri: https://docs.ansible.com/ansible/latest/collections/ansible/builtin/uri_module.html.

8. Создание временных файлов и папок.

Иногда, для работы нужно временное хранилище файлов, которое после работы можно будет удалить. Для работы с данным хранилишем в ansible можно использовать модуль **tempfile**.

Пример создания каталога:

```
- name: Create temporary ansible directory
  tempfile:
    state: directory
    suffix: ansible
  register: tmp
```

Путь до созданного каталога будет записан в переменную **tmp.path**.

После не забываем удалить созданную папку:

```
- name: Remove temporary ansible directory
 file:
  path: "{{ tmp.path }}"
  state: absent
 when: tmp.path is defined
```

О tempfile: https://docs.ansible.com/ansible/latest/collections/ansible/builtin/tempfile_module.html.

9. Работа с GIT.

О Git: https://docs.ansible.com/ansible/latest/collections/ansible/builtin/git_module.html.

а) Клонирование проекта из гита.

Выполняется с помощью модуля git.

```
- name: Clone docker-compose
 git:
  repo: "https://github.com/docker/compose.git"
  dest: /tmp/docker-compose
```

** в данном примере мы сделаем клон репозитория в каталог **/tmp/docker-compose**.*

б) Изменение содержимого для файла.

Модуль ansible не поддерживает отпавку изменений в git, но можно воспользоваться API. Например, для gitlab обновить контент файла можно с помощью модуля uri:

```
- name:
 uri:
  url: "https://gitlab.dmosk.ru/api/v4/projects/666/repository/files/folder%2Ffilename"
  method: PUT
  return_content: false
  body_format: json
  body:
    branch: "main"
    author_email: "master@dmosk.ru"
    author_name: "Dmitriy Mosk"
    content: "Text for file"
    commit_message: "update filename"
  headers:
   PRIVATE-TOKEN: 00000_1111111_33333
   Content-Type: application/json
  validate_certs: no
```

** где:*

- ***url** — полный путь до файла. Обратите внимание на:*

- ***gitlab.dmosk.ru*** *— адрес нашего сервера gitlab.*
- ***666*** *— идентификатор проекта. Его можно посмотреть в настройках самого проекта.*
- ***folder%2Ffilename*** *— путь до файла. В нормальном формате, это folder/filename.*

- ***body*** *— содержит данные, которые будут отправлены на сервер для смены контента.*
- ***headers PRIVATE-TOKEN*** *— токен доступа к API. Его можно создать в настройках профиля учетной записи Gitlab.*

Подробнее о работе API в гиблабе: https://docs.gitlab.com/ee/api/repository_files.html.

Практическая работа № 42 Использование Ansible для сбора информации

Задание:

а) Список виртуальных машин на хосте. Для получения информации о виртуальных машинах, которые находятся на хосте виртуализации нужно использовать API Proxmox. Для этого в ansible мы будем применять модуль URI:

```
- name: Get vms from pve
  uri:
    url: "https://pve.dmosk.local:8006/api2/json/cluster/resources?type=vm"
    headers:
      Authorization: PVEAPIToken=ansible@pve!Ansible=e94d5627-1f8d-36a7-37e2-7ad6fad65ab7
    follow_redirects: all
  register: vm_list
```

** где:*

- ***pve.dmosk.local*** *— адрес веб-интерфейса сервера виртуализации.*
- ***8006*** *— порт для подключения в веб-интерфейсу.*
- ***ansible@pve*** *— имя учетной записи, для которой создан токен доступа. Он создается в консоле управления Proxmox (**Датацентр - Разрешения - API Tokens**).*
- ***Ansible=e94d5627-1f8d-36a7-37e2-7ad6fad65ab7*** *— имя токена и сам токен.*

В результате мы сохраним список виртуальных машин в переменной **vm_list**.

б) Подробная информация о виртуальной машине или хосте виртуализации. Если мы хотим собрать побольше информации, нам также понадобится API:

```
- name: Get vm info
  uri:
```

```
  url: "https://pve.dmosk.local:8006/api2/json/<запрос>"
  headers:
    Authorization: PVEAPIToken=ansible@pve!Ansible=e94d5627-1f8d-36a7-37e2-
7ad6fad65ab7
  follow_redirects: all
 register: vm_info
```

Для получения различной информации о виртуальной машине или сервере мы используем различные запросы (отмечено как <запрос>). С полным списком того, что мы можем получить предлагаю ознакомиться на странице pve.proxmox.com/pve-docs/api-viewer.

Например, для получения конфигурации виртуальной машины, используем запрос:

nodes/<имя ноды>/qemu/<VMID>/config

Чтобы узнать hostname:

nodes/<имя ноды>/qemu/<VMID>/agent/get-host-name

И так далее.

в) Информация о машине с помощью модуля proxmox_kvm:

```
- name: "Get VMID"
 proxmox_kvm:
  node: "{{ pve_node }}"
  api_user: "{{ pve_user }}"
  api_password: "{{ pve_password }}"
  api_host: "{{ pve_host }}"
  name: "myvm"
  state: current
 register: vm_info
```

2. Операции с ВМ

Рассмотрим наиболее популярные действия над виртуальными машинами.

***) Остановить:**

```
- name: "Останавливаем виртуальную машину Proxmox"
 proxmox_kvm:
  api_host    : "{{ pve_host }}"
  api_user    : "{{ pve_user }}"
  api_password: "{{ pve_password }}"
  node        : "{{ pve_node }}"
  name        : "myvm"
  state       : stopped
  force       : yes
  timeout     : 300
```

***) Удалить:**

```
    - name: "Удаляем виртуальную машину Proxmox"
proxmox_kvm:
  api_host     : "{{ pve_host }}:{{ pve_port }}"
  api_user     : "{{ pve_user }}"
  api_password : "{{ pve_password }}"
  node         : "{{ pve_node }}"
  name         : "myvm"
  state        : absent
```