

# РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ МОНИТОРИНГА ДЛЯ АНАЛИЗА СОСТОЯНИЯ СЕРВЕРА

Докладчик: студент 942 гр. Воронцов М. Ю.

Научный руководитель: Власенко В. Д.



- ▶ В связи с автоматизацией и компьютеризацией учёта бизнес-процессов различными организациями возникает необходимость обеспечения стабильной работы серверных платформ на основе постоянного мониторинга их функционирования и своевременного устранения неполадок. Это обусловило актуальность данной темы.

АКТУАЛЬНОСТЬ



- ▶ Цель работы заключается в анализе существующих систем мониторинга и разработке мобильного мониторинга, обеспечивающего оптимизацию серверных платформ

ЦЕЛЬ ИССЛЕДОВАНИЯ

- ▶ Объектом исследования являются различные ИТ инфраструктуры

ОБЪЕКТ ИССЛЕДОВАНИЯ

- ▶ Способы мониторинга ИТ инфраструктур.

ПРЕДМЕТ ИССЛЕДОВАНИЯ

- ▶ Мониторинг - система сбора/регистрации, хранения и анализа ключевых (явных или косвенных) признаков/параметров описания данного объекта для вынесения суждения о поведении/состоянии данного объекта в целом.

ОПРЕДЕЛЕНИЕ

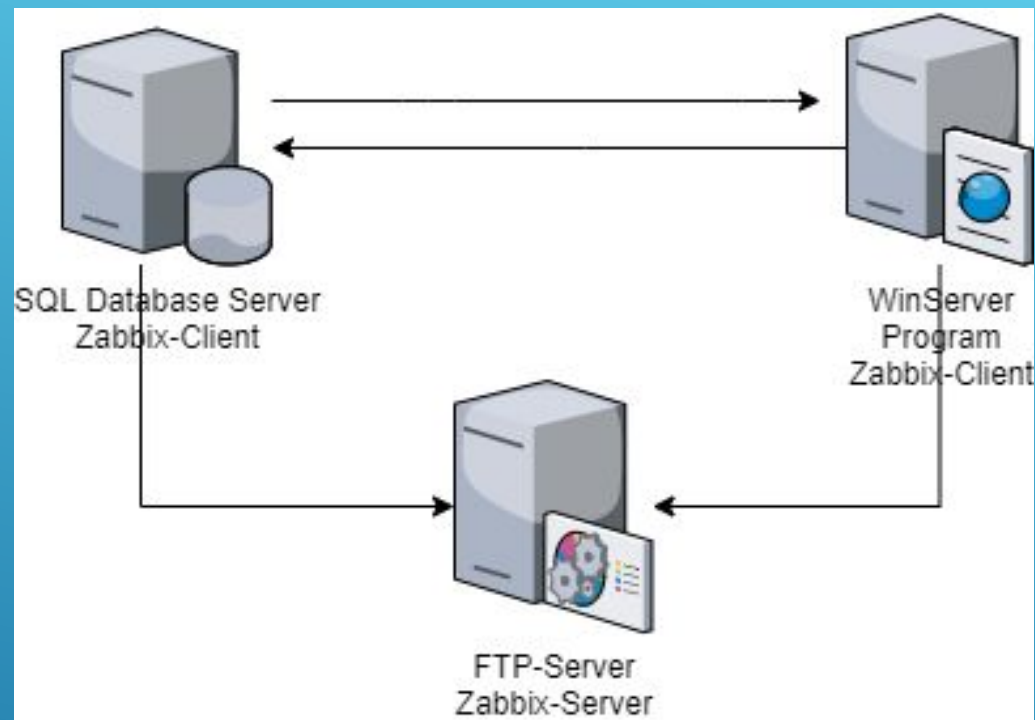
- ▶ В целях повышения скорости информирования системных администраторов, был выбран способ мобильного оповещения.
- ▶ После изучения существующих систем мониторинга за основу разрабатываемой системы была выбрана система Zabbix.
- ▶ В качестве инструментов также использовались:
  - Telegram API, позволяющий передавать информацию от сервера на мобильное устройство.
  - Язык программирования Python для автоматизации операций над базой данных.



На имеющийся Linux сервер, на котором находится база данных и Windows сервер, на котором расположено клиент-серверное приложение устанавливается Zabbix Client.

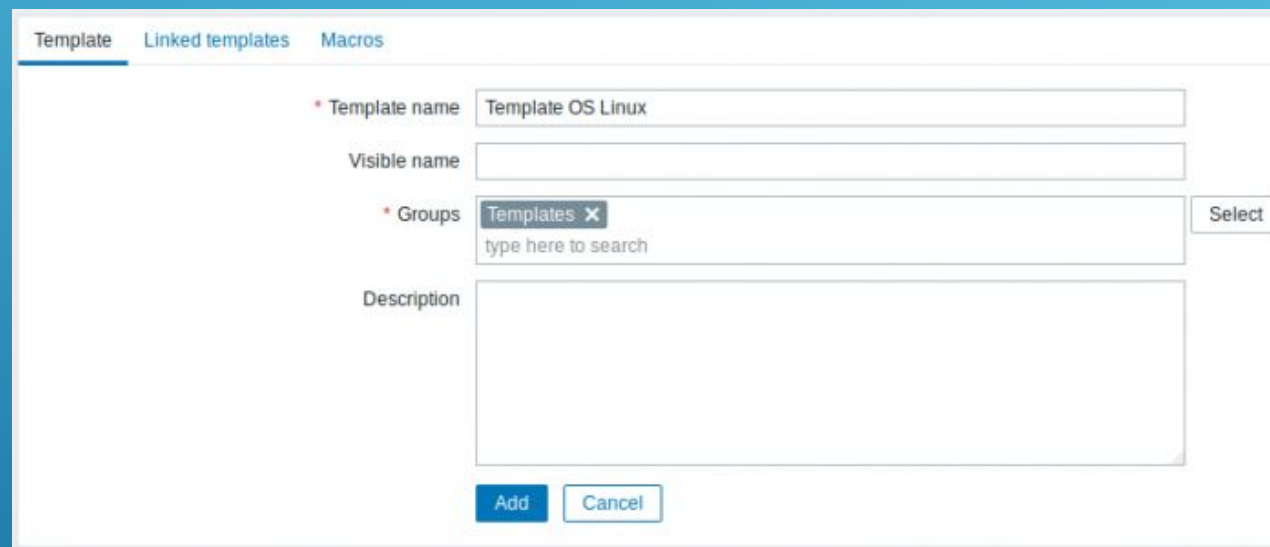
Дополнительно в эту цепочку включаем File Transfer Server с UNIX системой, на который устанавливаем Zabbix Server.

FTP сервер должен получать и обрабатывать информацию, поступающую от основных серверов и посылать соответствующие сообщения при возникновении проблемы и хранить бэкапы.





- ▶ **Шаблоном** является набор объектов, который можно удобно и просто применить к нескольким узлам сети
- ▶ Для настройки шаблона необходимо сначала создать его, указав общие параметры, и только затем добавлять объекты (триггеры) к этому шаблону.



The screenshot shows a configuration window for creating a template. The window has three tabs: "Template", "Linked templates", and "Macros". The "Template" tab is selected. The form contains the following fields:

- Template name:** A text input field containing "Template OS Linux".
- Visible name:** An empty text input field.
- Groups:** A dropdown menu showing "Templates" with a close button (X). Below it is a search box with the placeholder text "type here to search". A "Select" button is located to the right of the search box.
- Description:** A large empty text area.

At the bottom of the form, there are two buttons: "Add" (highlighted in blue) and "Cancel".

Создание шаблона

# ШАБЛОНЫ

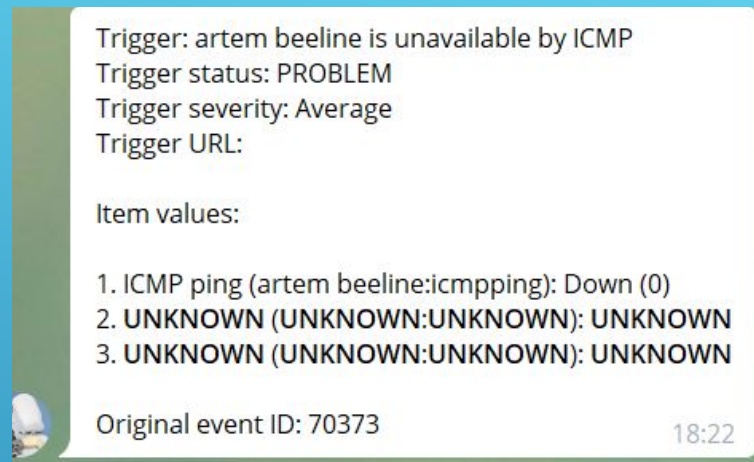
- ▶ **Триггеры это** логические выражения, которые “оценивают” данные собранные элементами данных и отражают текущее состояние системы. Именно по состоянию триггеров выявляются возникшие проблемы и запускается оповещение.

Для мониторинга были выбраны следующие критерии:

- Доступность хоста
- Процент потери пакетов
- Время задержки
- Свободное место на диске
- Загруженность процессора

# ТРИГГЕРЫ

- ▶ Сообщения передаются системным администраторам по токену мессенджера Telegram с помощью программы написанной на Python, используя двойное хеширование для проверки подлинности отправителя сообщения.



# ПЕРЕДАЧА СООБЩЕНИЯ

- ▶ С помощью скриптов на языке Python также были автоматизированы процессы обслуживания БД:
- Удаление ненужных файлов (так как PostgreSQL не выполняет физическое удаление данных с диска при удалении записей БД посредством системы управления).
- Перестройка индексов базы данных.

```
ts_server_bot
bot

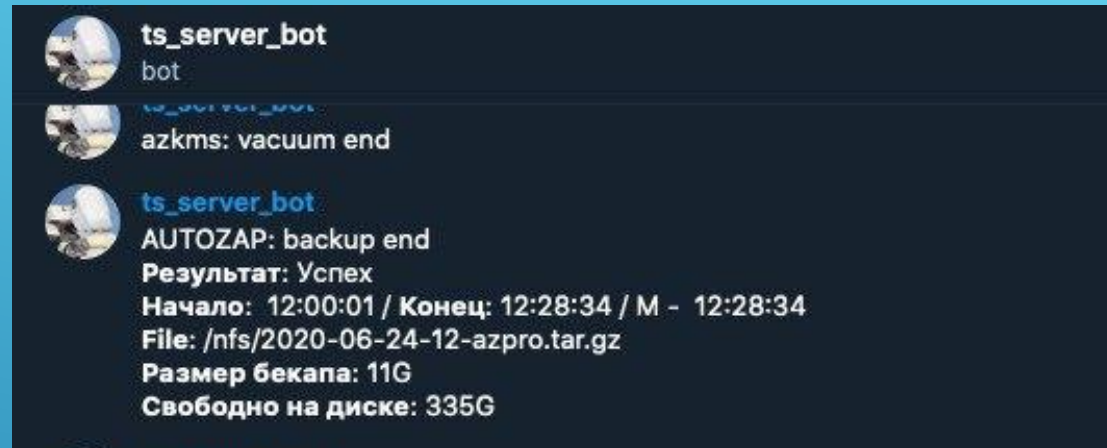
Сб июн 27 05:00:01 +10 2020
pgsql (vac.sh) : Начинаем vacuum + analyse
Сессий:      1
Размер базы: 103 GB

Сб июн 27 05:04:46 +10 2020
pgsql (vac.sh) : Закончил vacuum + analyse
vacuumdb: очистка базы данных "azpro"
vacuumdb: очистка базы данных "mamonsu"
vacuumdb: очистка базы данных "postgres"
vacuumdb: очистка базы данных "template1"
Сессий:      1
Размер базы: 103 GB
Начинаем перестройку индексов

Сб июн 27 05:45:30 +10 2020
pgsql (vac.sh) : Индексы перестроены 5:45
```

# АВТОМАТИЗАЦИЯ ОБСЛУЖИВАНИЯ БД

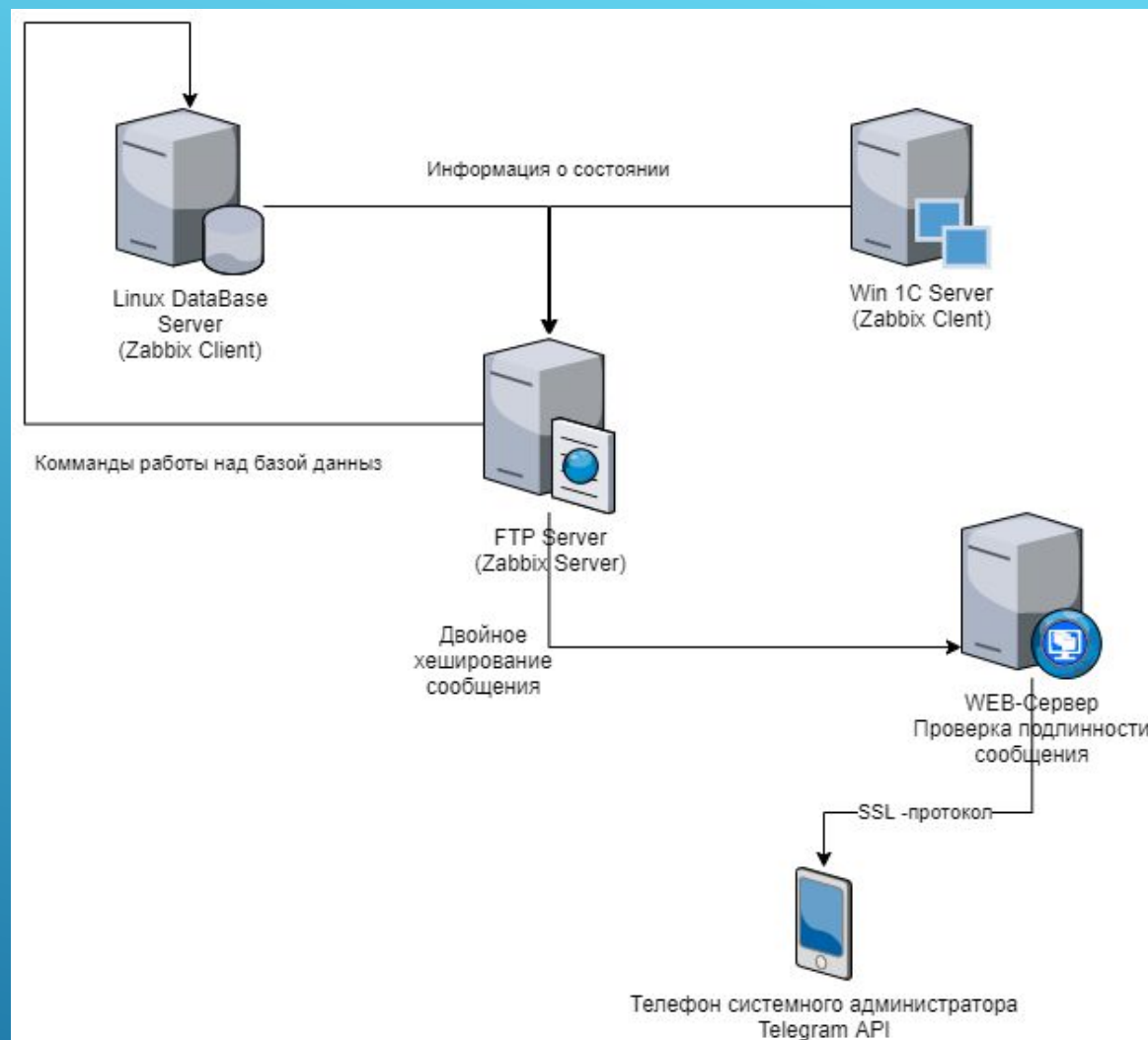
- ▶ Также на языке Python был разработан алгоритм автоматического и ежедневного формирования бэкапов с последующим уведомлением специалиста о успехе или провале операции.



```
ts_server_bot
bot
azkms: vacuum end
ts_server_bot
AUTOZAP: backup end
Результат: Успех
Начало: 12:00:01 / Конец: 12:28:34 / М - 12:28:34
File: /nfs/2020-06-24-12-azpro.tar.gz
Размер бекапа: 11G
Свободно на диске: 335G
```

# АВТОМАТИЗАЦИЯ ФОРМИРОВАНИЯ БЭКАПОВ

- ▶ Из-за того, что руководство мессенджера Telegram и Роскомнадзора до недавнего времени не могли прийти к компромиссу для бесперебойной работы системы оповещения потребовалось пустить трафик через сервер, расположенный в Латвии. Именно на этом сервере проверяется проверка подлинности отправителя. Итоговая архитектура мониторинга представлена на схеме:



- ▶ Изучены информационные структуры и способы их мониторинга
- ▶ Изучены возможности интегрирования мессенджеров в системы мониторинга
- ▶ Разработана мобильная система мониторинга сервера
- ▶ Автоматизирована часть работы, выполняемая над базами данных
- ▶ Система интегрирована в работу предприятия

## ЗАКЛЮЧЕНИЕ

**Акт**  
**О внедрении результатов выпускной квалификационной работы на тему**  
**«Разработка мобильного приложения для мониторинга состояния сервера»**

Автор: Воронцов Михаил Юрьевич, номер группы 942  
Руководитель выпускной квалификационной работы: Власенко Виктор Дмитриевич, доцент, к. ф-м.наук

Наименование организации: ООО «Техносоюз»

Сущность внедряемой разработки: проектирование комплекса мобильного мониторинга состояния сервера, обеспечивающий оперативное оповещение соответствующих специалистов

Форма внедрения: разработан комплекс мобильного мониторинга состояния сервера, обеспечивающий оперативное оповещение соответствующих специалистов о возникших проблемах. Автоматизирована часть работы, которую необходимо выполнять регулярно для корректной работы базы данных.

Эффективность внедрения: повышение стабильности работы сервера и базы данных, увеличена скорость решения проблем, связанных с серверной платформой.

Дата внедрения: 25 мая 2020 года

\_\_\_\_\_  
Девятко К. А.

(ФИО руководителя структурного подразделения или организации)

\_\_\_\_\_  
Воронцов М. Ю.

(ФИО автора)



(подпись)



СПАСИБО,  
УВАЖАЕМАЯ КОМИССИЯ

