

Лабораторная работа — Зависимые и вычисляемые элементы данных

Цель работы

Научиться создавать и использовать зависимые элементы данных (Dependent Items) в Zabbix для извлечения множества метрик из одного HTTP-ответа на примере Apache server-status?auto.

Ход работы:

Зависимый элемент данных (Dependent Item) — это элемент, который не опрашивает источник данных напрямую, а получает значение путём извлечения (через предобработку) из данных другого элемента — **Основной элемент данных**.

mod_status — модуль Apache, который отображает статистику веб-сервера в реальном времени. С параметром ?auto возвращает данные в машиночитаемом текстовом формате. Пример вывода с http://127.0.0.1/server-status?auto:

```
Total Accesses: 57621
Total kBytes: 891234
CPULoad: .045678
Uptime: 1234567
ReqPerSec: .0467
BytesPerSec: 1234.56
BytesPerReq: 26432.1
BusyWorkers: 2
IdleWorkers: 8
Scoreboard: _W__K__K__K__RR_.....
```

Преимущество использования зависимых элементов данных:

Преимущество	Описание
Один HTTP-запрос	Вместо 10+ отдельных элементов данных — один Основной элемент данных
Мгновенная консистентность	Все метрики получены в один момент времени
Снижение нагрузки	1 запрос вместо N к Apache
Гибкость извлечения	Через RegExp можно достать любую строку/число

2.2. Задание 1. Включение Apache mod_status

Если mod_status ещё не включён, выполнить настройку.

На Debian/Ubuntu:

```
# Включить модуль
sudo a2enmod status

# Отредактировать конфигурацию
sudo nano /etc/apache2/mods-available/status.conf

<Location "/server-status">
    SetHandler server-status
    Require local
    Require ip 127.0.0.1
    Require ip ::1
</Location>

# Перезапустить Apache
sudo systemctl restart apache2

# Проверить доступность
curl http://127.0.0.1/server-status?auto
```

2.3. Задание 2. Создание Основной элемент данных (HTTP агент)

Создать Основной элемент данных, который будет получать данные с http://127.0.0.1/server-status?auto.

Перейти: **Сбор данных** → **Узлы сети** → выбрать узел с Apache → **Элементы данных**

Нажать **Создать элемент данных**

Заполнить поля:

Поле	Значение
Имя	Apache server-status raw
Тип	HTTP агент
Ключ	apache.server.status.raw
Тип информации	Текст (Text)
URL	http://127.0.0.1/server-status?auto
Тип запроса	GET
Тип тела запроса	Сырые данные
Интервал обновления	1m

Перейти на вкладку **Предобработка (Preprocessing)** и добавить шаг:

Шаг	Параметры
-----	-----------

Отбрасывать не изменившееся	—
-----------------------------	---

Нажать **Добавить**

Проверить в **Мониторинг → Последние данные**, что элемент получает текст с сервера

2.4. Задание 3. Зависимый элемент — общее количество запросов

Создать зависимый элемент, извлекающий **Total Accesses** (общее количество обработанных запросов с момента запуска Apache).

Шаги выполнения:

Нажать **Создать элемент данных**

Заполнить поля:

Поле	Значение
Имя	Apache total accesses
Тип	Зависимый элемент данных
Ключ	apache.total.accesses
Тип информации	Числовой (Целое положительное)
Основной элемент данных	Apache server-status raw

Перейти на вкладку **Предобработка (Preprocessing)** и добавить шаги:

Имя	Параметры	Вывод
Регулярное выражение	Total Accesses: ([0-9]+)	\1

Нажать **Добавить**

Проверить значение в **Мониторинг → Последние данные**

Пояснение RegExp: Total Accesses: ([0-9]+) — находит строку Total Accesses: 57621 и извлекает число 57621 в первую группу захвата \$1.

2.5. Задание 4. Зависимые элементы — трафик и пропускная способность

Создать зависимые элементы для метрик трафика.

4.1. Total kBytes

Поле	Значение
Имя	Apache total kBytes

Тип	Зависимый элемент данных
Ключ	apache.total.kbytes
Тип информации	Числовой (Целое положительное)
Единицы измерения	KB
Основной элемент данных	Apache server-status raw

Перейти на вкладку **Предобработка (Preprocessing)** и добавить шаги:

Имя	Параметры	Вывод
Регулярное выражение	Total kBytes: ([0-9]+)	\1

4.2. Bytes per second

Поле	Значение
Имя	Apache bytes per sec
Тип	Зависимый элемент данных
Ключ	apache.bytes.per.sec
Тип информации	Числовой (С плавающей точкой)
Единицы измерения	B/s
Основной элемент данных	Apache server-status raw

Перейти на вкладку **Предобработка (Preprocessing)** и добавить шаги:

Имя	Параметры	Вывод
Регулярное выражение	BytesPerSec: ([0-9.]+)	\1

4.3. Bytes per request

Поле	Значение
Имя	Apache bytes per req
Тип	Зависимый элемент данных
Ключ	apache.bytes.per.req
Тип информации	Числовой (С плавающей точкой)
Единицы измерения	B/req
Основной элемент данных	Apache server-status raw

Перейти на вкладку **Предобработка (Preprocessing)** и добавить шаги:

Имя	Параметры	Вывод
-----	-----------	-------

Регулярное выражение	BytesPerReq: ([0-9.]*)	\1
----------------------	------------------------	----

2.9. Задание 8. Вычисление Uptime через зависимые элементы

Создать зависимый элемент для извлечения аптайма Apache и вычисляемый элемент для перевода в дни.

8.1. Uptime в секундах

Поле	Значение
Имя	Apache uptime
Тип	Зависимый элемент данных
Ключ	apache.uptime
Тип информации	Числовой (Целое положительное)
Единицы измерения	s
Основной элемент данных	Apache server-status raw

Перейти на вкладку **Предобработка (Preprocessing)** и добавить шаги:

Имя	Параметры	Вывод
Регулярное выражение	Uptime: ([0-9.]*)	\1

8.2. Uptime в днях (через вычисляемый элемент)

Создать вычисляемый элемент (Calculated Item), который делит аптайм на 86400:

Поле	Значение
Имя	Apache uptime days
Тип	Вычисляемое
Ключ	apache.uptime.days
Формула	last(/host/apache.uptime) / 86400
Тип информации	Числовой (С плавающей точкой)
Единицы измерения	d
Интервал обновления	1m

Вычисляемый элемент данных — это элемент, значение которого вычисляется на сервере Zabbix на основе значений других элементов данных. Вычисления выполняются по формуле, задаваемой в конфигурации элемента.

Особенности:

- Не требуют агента — вычисление происходит на стороне сервера
- Не потребляют сетевой трафик
- Могут агрегировать данные с разных узлов сети
- Поддерживают математические операции и функции

Поддерживаемые функции вычисляемых элементов.

Функция	Описание	Пример
avg	Среднее арифметическое	avg(/host/key, 5m)
sum	Сумма значений	sum(/host/key, 1h)
min	Минимальное значение	min(/host/key, 10m)
max	Максимальное значение	max(/host/key, 10m)
last	Последнее значение	last(/host/key)
count	Количество значений	count(/host/key, 5m, "error")
nodata	Проверка отсутствия данных	nodata(/host/key, 5m)
diff	Разница с предыдущим значением	diff(/host/key)
change	Изменение за период	change(/host/key)
percentile	Процентиль	percentile(/host/key, 5m, 95)
forecast	Прогнозирование	forecast(/host/key, 1h, "linear")
timeleft	Время до достижения порога	timeleft(/host/key, 1h, 100)
trend_avg	Среднее по трендам	trend_avg(/host/key, 1d)
trend_count	Количество по трендам	trend_count(/host/key, 1d)
trend_max	Максимум по трендам	trend_max(/host/key, 1d)
trend_min	Минимум по трендам	trend_min(/host/key, 1d)

Поддерживаемые операторы:

- Арифметические: +, -, *, /, % (остаток от деления)
- Сравнения: =, <>, >, <, >=, <=
- Логические: and, or, not
- Скобки: (,) — для группировки
- Числовые константы

