

# Zabbix

- [Общая идея](#)
- [Установка с использованием docker compose](#)
- [Источник клиент \(Zabbix trapper\)](#)
- [Источник сервер \(Zabbix agent + script\)](#)
- [Панели](#)
- [Настройка агента на alt linux](#)

????? ?????

Мониторинг критичен для любой компании.

## Prometheus

База данных временных рядов.

	Zabbix	Prometheus
Тип данных	Любой	Только Время
Период хранения	Не ограничивается	Оперативный мониторинг
Скорость обработки данных	Падает с объемом	Слабо зависит от объема

Состав системы:

- Prometheus
- Exporter: инструмент для предобработки/получения метрик
- Alertmanager: уведомления

Модели получения данных: Pull (обращается к машине и забирает накопившиеся данные) и Push (машина отдает данные на pushgateway, откуда потом prometheus забирает данные).

Типы метрик:

Счетчик	Количество элементов за определенный период (сколько раз падал сервер за месяц или сколько человек посетило сайт за день). Только положительный.
Измеритель	Положительный и отрицательный. Но только текущее значение.
Гистограмма	
Сводка	Улучшенная гистограмма

Alertmanager

Postmortem

[12.4 Постмортем пример.pdf](#)



# ????????? ? ??????????????????????

## docker compose

### Первичная настройка сервера

Добавление пользователя

```
adduser sergey
usermod -aG sudo sergey
```

Установка Docker

Одного ядра для запуска все-таки мало. Тормозит.

### Установка сервера

Клонируем репозиторий Zabbix

```
git clone https://github.com/zabbix/zabbix-docker.git
```

При необходимости переключаемся на нужную версию

```
git checkout 7.0
```

Запускаем нужный файл, например Alpine+MySQL:

```
docker compose -f ./docker-compose_v3_alpine_mysql_latest.yaml up
```

Структура названий файлов: docker-compose\_v3\_<distr>\_<db>\_<тип сборки>

distr:

alpine	Alpine Linux
ol	Oracle Linux
ubuntu	Ubuntu Linux

db:

mysql	MySQL
-------	-------

pgsql	PostgreSQL
-------	------------

тип сборки:

latest	Предсобранные образы
local	Локально собираемые образы

По-умолчанию логин Admin, пароль zabbix

Для версии 7.0 на момент августа 2024 г. для использования оповещений с использованием curl (например, Telegram) необходимо в файл env\_vars/.env\_srv добавить строку.

```
ZBX_SOURCEIP=0.0.0.0
```

Планируется исправить в следующих релизах. Проблема в том, что для работы последней версии curl требуется данный параметр.

Для работы web мониторинга добавил

Сборка и запуск в фоновом режиме:

```
docker compose -f ./docker-compose_v3_alpine_mysql_latest.yaml up -d
```

Источники:

[Официальная документация](#)

## Установка агента

Для Linux:

```
sudo apt-get install zabbix-agent
```

Изменяем настройки в файле /etc/zabbix/zabbix\_agentd.conf

Hostname=	Имя хоста,
Server=	
ServerActive=	
AllowKey=system.run[*]	

# ????????? ??????? (Zabbix trapper)

## Обновление данных по инициативе клиента

Создать элемент данных Zabbix trapper

[Item](#) [Tags](#) [Preprocessing](#)

\* Name

Type

\* Key

Type of information

Units

\* History

\* Trends

Value mapping

Allowed hosts

Populates host inventory field

Description

Enabled

### [документация](#)

При помощи утилиты zabbix\_sender:

```
zabbix_sender -z <server IP address> -p 10051 -s "New host" -k trap -o "test value"
```

-s

имя хоста, существующего в hosts на сервере zabbix и к которому привязан элемент данных, в кавычках

-k	ключ, без кавычек
-o	значение, в кавычках если тип данных строка и без кавычек если число

При помощи python:

### [Документация](#)

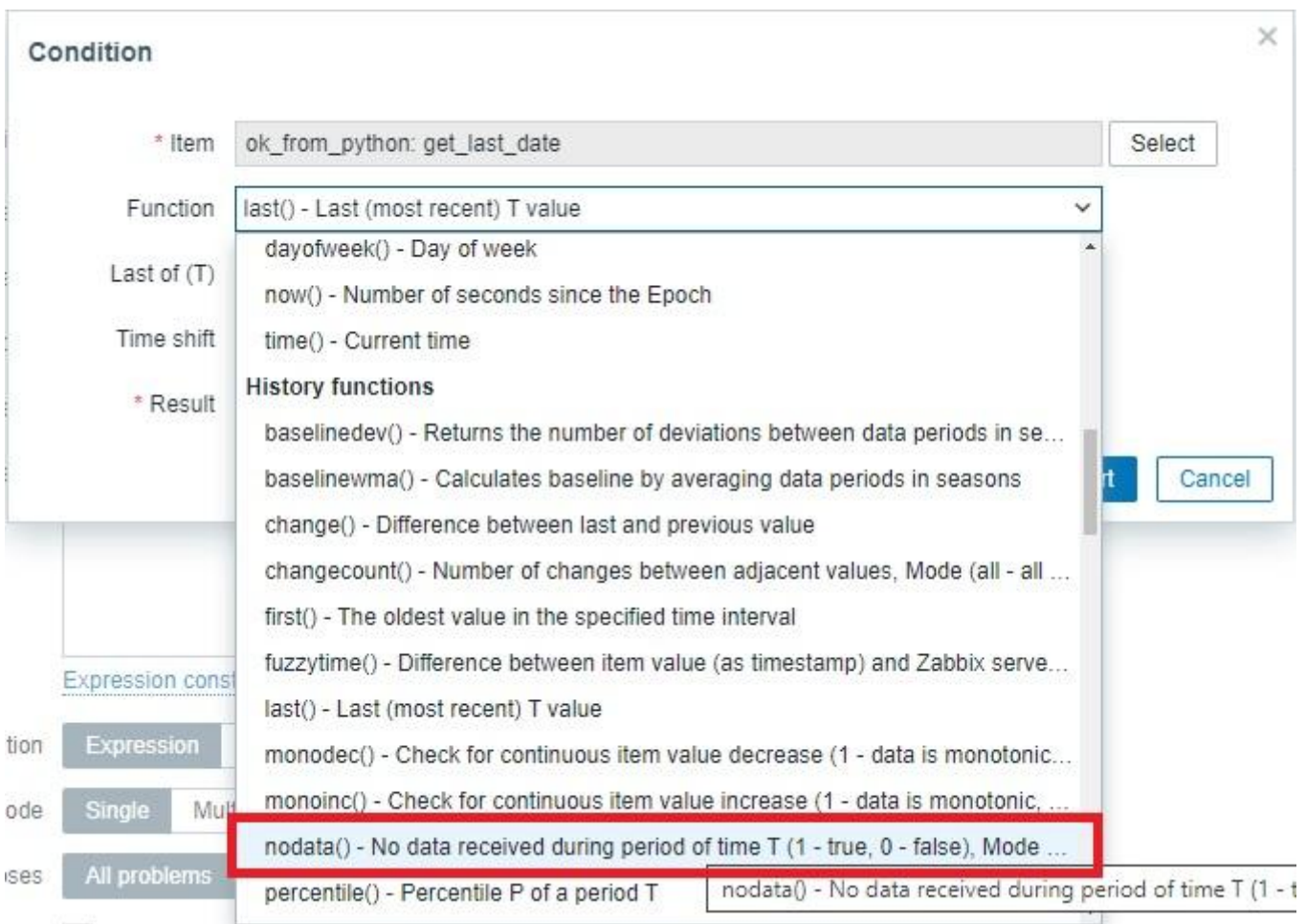
```
pip install zabbix_utils
```

Добавляем этот код в нужный скрипт и начинаем получать информацию о времени выполнения скрипта. Отличие от параметров утилиты - ключ нужно указывать в кавычках.

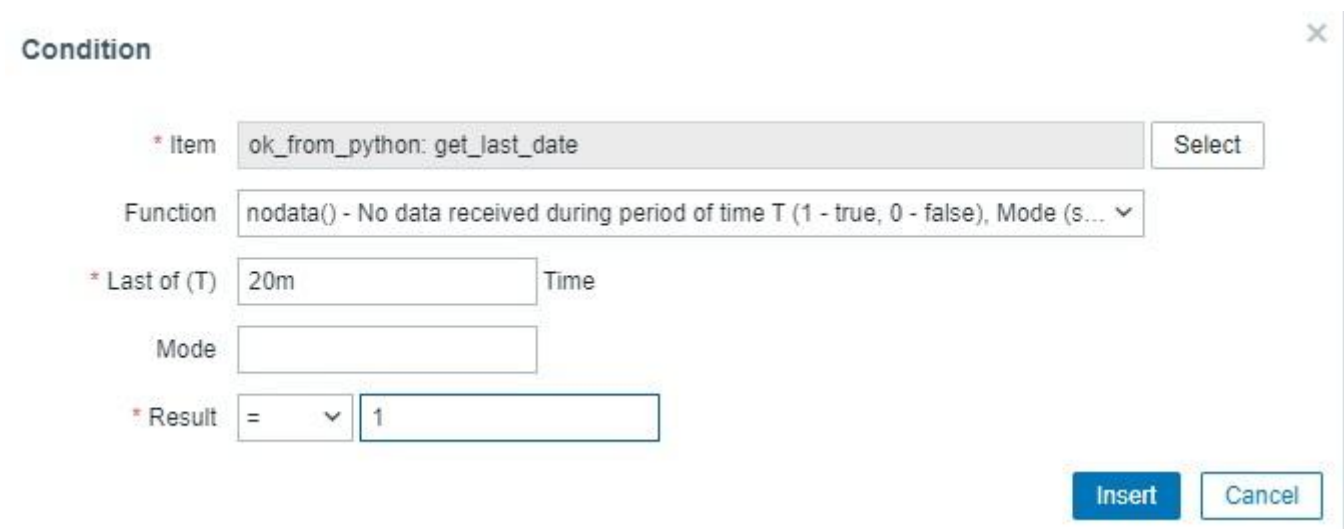
```
from zabbix_utils import Sender
import time
import datetime

sender = Sender(server='192.168.1.198', port=10051)
# Parameters: (host, key, value, clock)
d = datetime.datetime.now()
unix_time = int(time.mktime(d.timetuple()))
resp = sender.send_value('moodle', 'lastupdate', 0, unix_time)
if resp.failed == 0:
    # Print a success message along with the response time
    print(f"Value sent successfully in {resp.time}")
else:
    # Print a failure message
    print("Failed to send value")
    print(resp.details)
```

Добавляем триггер, в условиях для анализа последних данных функция nodata



Вид условия, срабатывающего при отсутствии данных более 20 минут. Поддерживаются окончания s, m, h, d, w, m



Общий вид триггера:

В поле Event name можно указать текст, который будет выводиться в dashboard при оповещении. Этот текст будет отображаться при следующем срабатывании триггера, если триггер сейчас активен, то текст не изменится.

\* Name

Event name

Operational data

Severity Not classified Information Warning Average High Disaster

\* Expression  Add

[Expression constructor](#)

OK event generation Expression Recovery expression None

PROBLEM event generation mode Single Multiple

OK event closes All problems All problems if tag values match

Allow manual close

Menu entry name

Menu entry URL

Description

Enabled

Также для связанных триггеров (например 1минута-3минуты-5минут) нужно зайти в Зависимости и указать следующий по уровню триггер, т е для указанного примера нужно в триггере 1минута указать зависимость от 3минуты, в триггере 3минуты - зависимость 5минут.

Пример для триггера 1минута.

### Trigger

Dependencies	Name	Action
	<a href="#">ok_from_python: Задержка более 3 минут</a>	<a href="#">Remove</a>
	<a href="#">Add</a> <a href="#">Add host trigger</a>	

Добавляем отображение на дашборде

## Edit widget

? X

Type **Problems**  Show header

Name

Refresh interval

Show **Recent problems** **Problems** History

Host groups

Exclude host groups

Hosts **moodle**

Problem

Severity  Not classified  Warning  High  
 Information  Average  Disaster

Problem tags **And/Or** Or

[Add](#)

Show tags **None** 1 2 3

Tag name **Full** Shortened None

Tag display priority

Show operational data **None** Separately With problem name

Show symptoms

Show suppressed problems

Acknowledgement status **All** Unacknowledged Acknowledged By me

Sort entries by

Show timeline

Графики

Ссылка на скриншоты графиков

<http://192.168.1.198/chart2.php?graphid=2845&from=now-3h&to=now>



# ????????? ??????? (Zabbix agent + script)

В этом случае создается источник данных Zabbix agent и устанавливается ключ system.run[размещение скрипта на клиенте].

Элемент данных Теги Предобработка 1

Родительские элементы данных **abctest**

\* Имя

Тип

\* Ключ

Тип информации

\* Интерфейс узла сети

Единицы измерения

\* Интервал обновления

Пользовательские интервалы

Тип	Интервал	Период	Действие
<input checked="" type="checkbox"/> Переменный	<input type="text" value="По расписанию"/>	<input type="text" value="50s"/>	<input type="text" value="1-7,00:00-24:00"/>

[Добавить](#) [Удалить](#)

\* Время ожидания

\* История

\* Динамика изменений

Преобразование значений

Заполнение поля инвентаря узла сети

Описание

Активировано

Можно установить предобработку данных перед итоговым занесением в БД

Шаги предобработки ?

Имя	Параметры	Другое при ошибке	Действия
1: Регулярное выражение	All ok 1	<input checked="" type="checkbox"/>	<a href="#">Тест</a> <a href="#">Удалить</a>
Другое при ошибке	<input type="button" value="Отбросить значение"/> <input type="button" value="Задать значением"/> <input type="button" value="Задать ошибку в"/>	<input type="text" value="0"/>	

[Добавить](#) [Тест всех шагов](#)

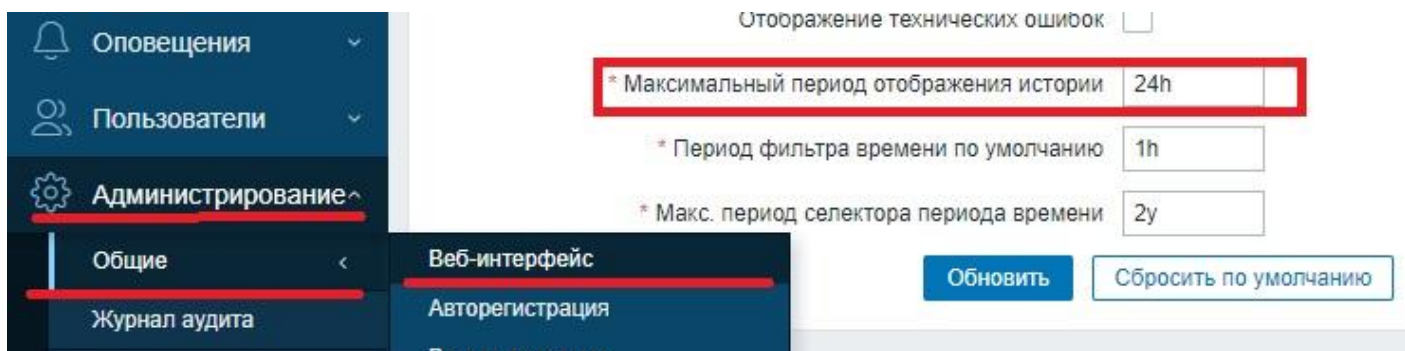
Тип информации:

Дальше - стандартно триггеры, ...

??????

### Отображение в виджете при больших периодах получения данных

При отображении часто используются данные "Последнее значение". Однако, оно по умолчанию использует максимальный период в 24 часа. Соответственно, если последние данные пришли позже этого периода, то считается, что последних данных нет. Поэтому отображаемые элементы будут глючить при большем периоде. Для увеличения периода нужно перейти в Администрирование->Общие->Веб-интерфейс и изменить параметр "Максимальный период отображения истории"



Этот период также можно изменить через функцию агрегации для каждого виджета. Там появляется элемент "период...".

????????? ?????? ?? alt linux

B