

Очередная статья на тему использования современной универсальной системы мониторинга. Расскажу, как с помощью zabbix настроить мониторинг времени на серверах и предупреждать при слишком больших расхождениях. На мой взгляд, полезная метрика, которой почему-то нет в базовых проверках.

Если у вас есть желание научиться работать с роутерами микротик и стать специалистом в этой области, рекомендую по программе, основанной на информации из официального курса **MikroTik Certified Network Associate**. Курс стоящий, все подробности читайте по ссылке. Есть бесплатные курсы.

#### Содержание:

- 1 Цели статьи
- 2 Введение
- 3 Элемент данных для мониторинга за временем
- 4 Триггер для оповещения о неправильном времени
- 5 Проверка работы
- 6 Заключение

#### Цели статьи

1. Рассказать о разных способах мониторинга локального времени на сервере.
2. Описать выбранный способ.
3. Предоставить готовый шаблон для zabbix сервера.

## Введение

Задача по мониторингу времени на сервере на первый взгляд кажется тривиальной, но с ней не все так просто. Я несколько раз брался за эту задачу и не доводил ее до конца. Сейчас уже не помню всех подробностей, так как поделиться решил финальной версией, которая уже некоторое время у меня функционирует. Основная проблема там в том, что время с агента берется в один момент, а время заббикс сервера берется в другой момент. И когда они сравниваются, может быть большое расхождение именно потому, что бралось время для сравнения в разные моменты, а не потому, что время реально расходится.

Сравнение времени идет между временем сервера zabbix и локальным временем на сервере, где установлен агент, с помощью встроенной функции **fuzzytime**. Я для себя решил, что мне достаточно проверять время раз в минуту. Если расхождение будет более 60 секунд в трех последних проверках, то срабатывает триггер.

Теоретически, к этому триггеру можно прицепить action, который будет что-то исполнять на сервере. К примеру, запуск утилиты ntpdate для синхронизации времени. Но я не стал это настраивать, так как на всех серверах у меня всегда работает служба ntpd или chrony. Так что если время неактуальное, то надо идти на сервер и проверять, что со службой синхронизации времени.

Приступаем к настройке.

Если у вас еще нет своего сервера для мониторинга, то рекомендую материалы на эту тему. Для тех, кто предпочитает систему CentOS:

1. Установка CentOS 7.
2. Настройка CentOS 7.
3. Установка и настройка zabbix сервера.

То же самое на Debian 9, если предпочитаете его:


1. Установка Debian 9.
2. Базовая настройка Debian 9.
3. Установка и настройка zabbix на debian.

## Элемент данных для мониторинга за временем

В финальном варианте я предлагаю следующий элемент данных для мониторинга локального времени на сервере.



Элемент данных [Предобработка](#)

\* Имя  

Тип

\* Ключ

\* Формула

Тип информации

Единица 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](#)

\* Период хранения истории

\* Период хранения динамики изменений

Отображение значения  [показать преобразования значений](#)

Новая группа элементов данных

Группы элементов данных

Это **вычисляемый элемент** данных. Формула:

```
fuzzytime(system.localtime,60)
```

В данном случае функция `fuzzytime` сравнивает время на сервере `zabbix` с локальным временем наблюдаемого сервера. Если разница больше 60 секунд, функция возвратит 0, если меньше, то 1.

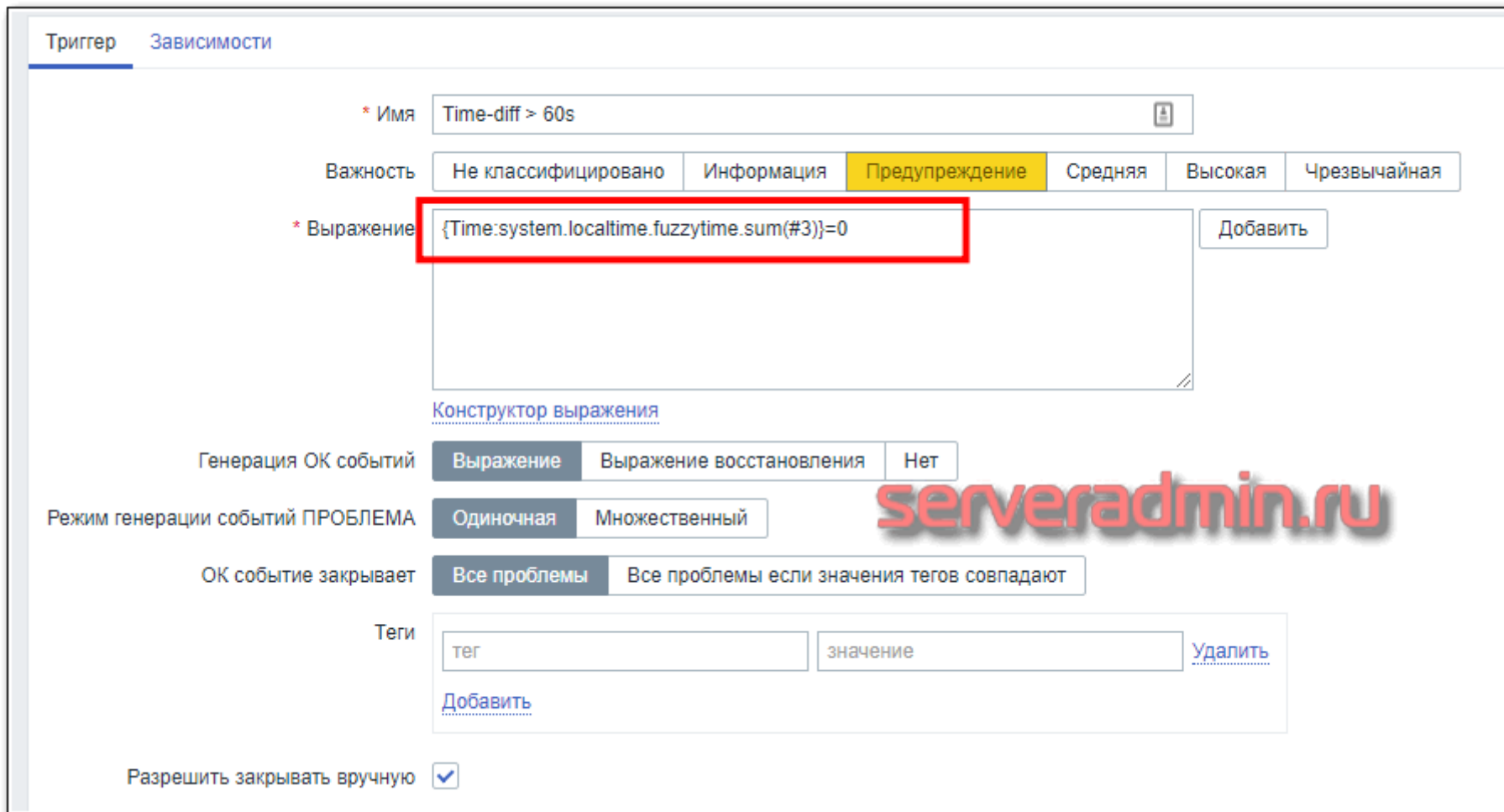
С помощью вычисляемого элемента в базу сразу идет конкретный результат проверки на текущий момент времени, который потом легко использовать в триггере. Не нужно делать каких-то вычислений и сравнений в нем. Я просто беру 3 последних проверки и если они все три вернули значение 0, значит триггер сработает.

Вместо вычисляемого элемента можно было бы сделать обычный с ключом `system.localtime`, а затем функцию `fuzzytime` использовать уже в триггере. Но там может получаться большая погрешность, если интервал обновления будет большой (несколько минут), а проверяемая разница небольшая (десятки секунд). Мне не понравилось, как это работало, поэтому переделал на вычисляемое значение сразу в итеме.

## Триггер для оповещения о неправильном времени

Вот пример самого триггера для оповещений.





Триггер    Зависимости

\* Имя

Важность  Не классифицировано  Информация  Предупреждение  Средняя  Высокая  Чрезвычайная

\* Выражение

[Конструктор выражения](#)

Генерация ОК событий  Выражение  Выражение восстановления  Нет

Режим генерации событий ПРОБЛЕМА  Одиночная  Множественный

ОК событие закрывает  Все проблемы  Все проблемы если значения тегов совпадают

Теги

[Добавить](#)

Разрешить закрывать вручную

Если 3 последние проверки были с ошибкой, то их сумма будет равно 0. После этого шлем уведомление. Как только время будет синхронизировано, триггер сразу же перейдет в нормальное состояние.



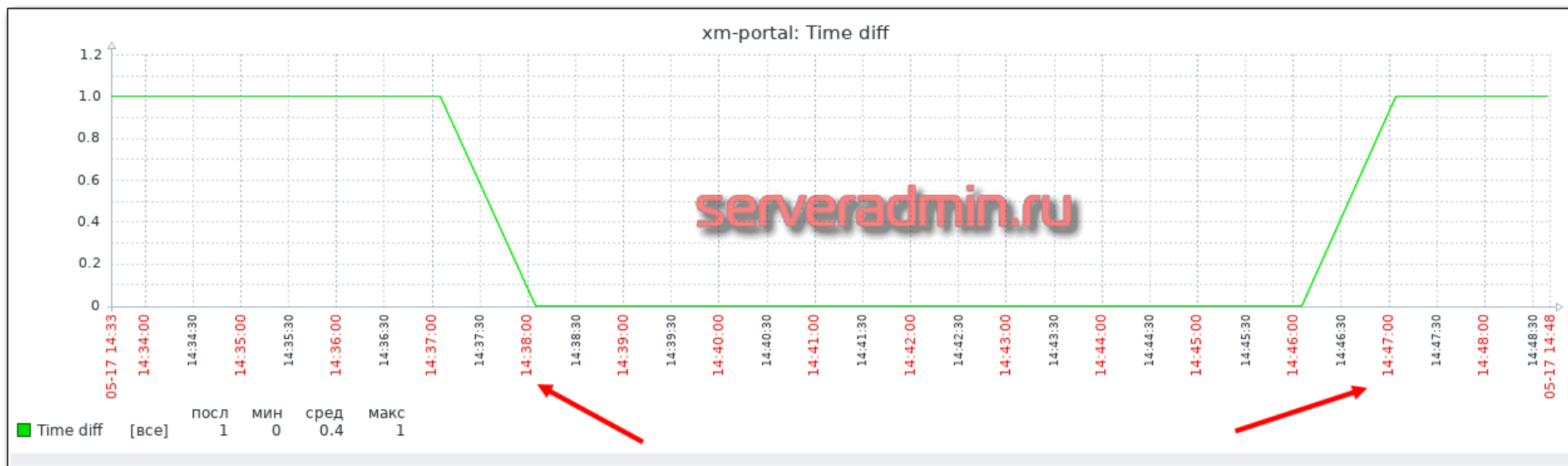
Вот готовый шаблон с элементом и триггером `zabbix_time.xml`. Проверки работают как на linux, так и windows серверах. Шаблон выгружен с сервера версии 4.0.

Важно следить, чтобы на самом сервере мониторинга было всегда правильное время, иначе в мониторинге не будет никакого смысла. Сравнение идет конкретно со временем на сервере zabbix. Нужно это помнить и следить за ним.

## Проверка работы

Для проверки работы мониторинга за временем, достаточно на каком-то сервере установить время с отставанием или опережением более чем на 60 секунд. После этого подождать 3 неудачные проверки.





После этого сработает триггер.



Время ▼	<input type="checkbox"/> Важность	Время восстановления	Состояние	Инфо	Узел сети	Проблема	Длительность
14:40:06	<input type="checkbox"/>		РЕШЕНО		xm-portal	Time-diff > 60s	7м

*serveradmin.ru*

#### Интересные записи:

Шутки для сисадминов

Мои программы для системного администрирования

Монетизация ИТ блога, сколько можно заработать на информационном сайте

Как видно по графику, 3 неверных значения пришли к 14:40, в этот момент сработал триггер. В 14:47 проверка вернула значение 1 и триггер перешел в состояние **РЕШЕНО**. В это время я сначала остановил ntpd, потом задал неправильное время командой:

```
# date +%T -s "14:35:13"
```

Этим я создал отставание на полторы минуты. Подождал несколько минут, потом обратно запустил службу ntpd и дождался, когда она проведет синхронизацию.

## Заключение

Не понравилась статья и хочешь научить меня администрировать? Пожалуйста, я люблю учиться. Комментарии в твоём распоряжении. Расскажи, как сделать правильно!

Такая проверка времени серверов работает нормально и ложных срабатываний почти нет. Если вам нужны более жесткие интервалы проверки, то могут быть трудности. К примеру, если вам надо сделать за временем и предупредить, когда разница становится 5 секунд. Если сделать, как показано у меня, то будет работать не надежно.

Точно не знаю, с чем это связано. Возможно сетевые задержки, или в работе вычисляемых значений. Тут надо внимательно смотреть в реализацию fuzzytime. Мне это было не нужно. Если кто-то решал подобную задачу, прошу поделиться в комментариях, как это делали вы.

Читайте мои остальные статьи по мониторингу с помощью zabbix.

## Онлайн курсы по Mikrotik

Если у вас есть желание научиться работать с роутерами микротик и стать специалистом в этой области, рекомендую пройти курсы по программе, основанной на информации из официального курса **MikroTik Certified Network Associate**. Помимо официальной программы, в курсах будут лабораторные работы, в которых вы на практике сможете проверить и закрепить полученные знания. Все подробности на сайте . Стоимость обучения весьма демократична, хорошая возможность получить новые знания в актуальной на сегодняшний день предметной области. Особенности курсов:

- Знания, ориентированные на практику;
- Реальные ситуации и задачи;
- Лучшее из международных программ.

Помогла статья? Есть возможность отблагодарить автора