

Продолжаю серию рассказов на тему настройки различных метрик в системе мониторинга заббикс. Сегодня расскажу, как быстро, легко и удобно передавать данные в json формате в zabbix и там их обрабатывать. Развитие функционала в последних версиях позволяют работать с json без лишних костылей и самописных скриптов.

Содержание:

- 1 Передача json данных в zabbix
- 2 Настройка zabbix server для мониторинга по json
- 3 Заключение

Передача json данных в zabbix

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.

Для примера возьму мониторинг работы ноды Bitcoin. В данном случае это совершенно не принципиально, так как подход к парсингу json в zabbix

одинаковый везде. Формат json сейчас очень популярен. Большое число сервисов его поддерживают для вывода информации или логов. У биткоин ноды есть команды для вывода информации о состоянии ноды. Вывод идет в формате json. Вот пример:

```
# bitcoin-cli getblockchaininfo
```

```
{
  "chain": "test",
  "blocks": 1447503,
  "headers": 1447503,
  "bestblockhash": "506a6952d47b939e9d7baa34fc290b4356c446",
  "difficulty": 12691409.16925421,
  "mediantime": 1544607906,
  "verificationprogress": 0.9999915635029165,
  "initialblockdownload": false,
  "chainwork": "00000000000000000000000000000000db56207f5987ad3",
  "size_on_disk": 23159068639,
  "pruned": false,
  "softforks": [
    {
      "id": "bip34",
      "version": 2,
      "reject": {
        "status": true
      }
    },
    {
      "id": "bip66",
      "version": 3,
      "reject": {
        "status": true
      }
    }
  ],
}
```

```
{
  "id": "bip65",
  "version": 4,
  "reject": {
    "status": true
  }
},
"bip9_softforks": {
  "csv": {
    "status": "active",
    "startTime": 1456790400,
    "timeout": 1493596800,
    "since": 770112
  },
  "segwit": {
    "status": "active",
    "startTime": 1462060800,
    "timeout": 1493596800,
    "since": 834624
  }
},
"warnings": "Warning: unknown new rules activated (versionbit 28)"
}
```

В своем примере я покажу, как из этого вывода парсить значения blocks и headers, хранить их в отдельных итемах в zabbix, чтобы дальше использовать в триггерах. По аналогии можно будет работать с другими значениями.

Настройка парсинга json в zabbix будет состоять из следующих шагов:

1. Передаем всю json строку в исходном виде в итем в заббиксе.
2. С помощью зависимых элементов данных формируем отдельные итемы с нужными значениями.
3. Используем полученные итемы по назначению.

Для начала делаем простой скрипт для получения json — `/etc/zabbix/scripts/getblockchaininfo.sh`

```
#!/bin/bash
/usr/bin/bitcoin-cli -rpcuser=user -rpcpassword=password -rpcport=8332 getblockchaininfo
```

Проверьте, что скрипт выдает корректную строку json. Далее создаем **UserParameter**. Я предпочитаю их хранить в разных конфигах, разделяя по функционалу. Создаю файл `/etc/zabbix/zabbix_agentd.d/btc.conf` следующего содержания:

```
UserParameter=blockchaininfo,/etc/zabbix/scripts/getblockchaininfo.sh
```

Сохраняем и перезапускаем zabbix-agent. Проверяем, что json корректно передается в нужном виде, без ошибок и прочих проблем:

```
# zabbix_agentd -t blockchaininfo
```

На выходе должны увидеть тот же самый json.

Zabbix агент подготовили. Для продолжения настройки, переходим на сервер.

Настройка zabbix server для мониторинга по json

На сервере создаем шаблон и добавляем в него элемент данных.

All templates / Template BTC Applications 1 Items 5 Triggers 2 Graphs Screens Discovery rules Web scenarios

Item Preprocessing

* Name

Type

* Key

Type of information

* Update interval

Custom intervals

| Type | Interval | Period | Action |
|----------------------------------------------|-----------------------------------------|----------------------------------|----------------------------------------------|
| <input checked="" type="checkbox"/> Flexible | <input type="text" value="Scheduling"/> | <input type="text" value="50s"/> | <input type="text" value="1-7,00:00-24:00"/> |

* History storage period

New application

Applications

- None-
- BTC

Populates host inventory field


Description

Enabled


Подключите этот шаблон к хосту, где настраивали zabbix агент и подождите поступления данных. После того, как убедитесь, что json строка поступает на сервер, возвращайтесь в шаблон и создавайте еще один **зависимый** элемент данных.

Item Preprocessing

* Name

Type 

* Key

* Master item 

Type of information

Units

* History storage period

* Trend storage period

Show value


New application

Applications

Populates host inventory field

Description

Enabled



В моем случае значение, которое я буду парсить из json, числовое, поэтому я выбираю соответствующий тип. У вас он может быть другой.

Дальше переходим на вкладку **Preprocessing (предобработка)**:

| Preprocessing steps | Name | Parameters | Action |
|---------------------|-----------|------------|------------------------|
| | JSON Path | \$.blocks | Remove |

[Add](#)

[Update](#) [Clone](#) [Delete](#) [Cancel](#)

В моем случае путь json очень простой. Если у вас более сложные выражения, то используйте удобный json парсер. Можете добавить туда свой json вывод и указать путь для парсинга, проверив таким образом его правильность.

Inputs

Output paths

JSONPath Syntax

Example '\$.phoneNumbers[*].type' See also [JSONPath expressions](#)

JSON

```
1 {
2   "chain": "test",
3   "blocks": 1447504,
4   "headers": 1447504,
5   "bestblockhash": "7ee1a5ddb2791a0cf3320504d720749",
6   "difficulty": 12691409.16925421,
7   "mediantime": 1544608633,
8   "verificationprogress": 0.999997537727991,
9   "initialblockdownload": false,
10  "chainwork": "0000000000000000db56e227ec4ab6",
11  "size_on_disk": 23159125386,
12  "pruned": false,
13  "softforks": [
14    {
15      "id": "bip34",
16      "version": 2,
17      "reject": {
18        "status": true
19      }
20    },
21    {
22      "id": "bip66",
23      "version": 3,
24      "reject": {
25        "status": true
26      }
27    },
28    {
29      "id": "bip65"
```

Evaluation Results

```
1 [
2   1447504
3 ]
```

serveradmin.ru

Я убедился, что все верно. Сохраняем этот итем и ждем, когда он будет обработан. В latest data увидите значения из json строки.

| <input type="checkbox"/> Host | Name ▲ | Last check | Last value | Change |
|-------------------------------|----------------|------------------------|--------------------------------|--------------------------|
| ▼ <u>yoch-BTC</u> | BTC (5 Items) | | | |
| <input type="checkbox"/> | Blockchaininfo | 12/12/2018 02:35:07 PM | { "chain": "test", "blocks"... | History |
| <input type="checkbox"/> | Blocks | 12/12/2018 02:35:07 PM | 1447508 | +1 Graph |
| <input type="checkbox"/> | Headers | 12/12/2018 02:35:07 PM | 1447508 | +1 Graph |
| <input type="checkbox"/> | Networkactive | 12/12/2018 02:35:11 PM | true | History |
| <input type="checkbox"/> | Networkinfo | 12/12/2018 02:35:11 PM | { "version": 170000, "su... | History |

serveradmin.ru

По аналогии добавляете остальные элементы. Таким образом вы можете распарсить полностью json в отдельные итемы и использовать их дальше в триггерах и на графиках.

Заключение

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

Я показал простейший пример работы с json для тех, кто никогда не использовал его в zabbix. Благодаря тому, что в какой-то не очень давней версии появились зависимые элементы, работа с мониторингом сильно упростилась. Первое, что мне пришло в голову, когда нужно было выбрать отдельные значения из json вывода — написать баш скрипт. Это не сложно, я бы быстро распарсил и вытащил те данные, что мне нужны. Но вспомнив про зависимые элементы и возможность обработки данных в самом заббиксе, решил посмотреть в эту сторону и не ошибся. Так действительно проще и удобнее. За нас уже все сделали разработчики.

Скорее всего вам понадобится не просто парсинг json файла, но и автообнаружение на основе данных из него и автоматическое создание итемов. Но это уже тема отдельной статьи. А на сегодняшний момент это все, что я хотел рассказать, пока выдалась возможность поделиться информацией.

Помогла статья? Есть возможность отблагодарить автора