# ZABBIX 5.4

**Alexei Vladishev**
ZABBIX Founder & CEO

zabbix

Zabbix is an <span style="color:red">Universal</span> Open Source enterprise-level monitoring solution

# Теги для всего!

# ТЕГИ

✓ **ГОТОВО:** Теги для шаблонов, хостов, прототипов хостов, триггеров

✓ **Zabbix 5.4:** Теги заменят applications

| Time ▲ | | Severity | Recovery time | Status | Info | Host | Problem | Duration | Ack | Actions | Tags |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2019-05-23 16:37:00 | ☐ | Average | | PROBLEM | | Linux907 | Zabbix agent on Linux907 is unreachable for 5 minutes | 1y 4m 19d | No | | Service: Zabbix agent |
| September ○ | | | | | | | | | | | |
| 2019-09-30 12:45:45 | ☐ | Information | | PROBLEM | | AZ M08 | Low CPU utilization on host machines | 1y 9d | Yes | 1 ⤳1 | Service: Kubernetes   Datacenter: FR2 |
| 2019-09-30 12:45:45 | ☐ | Information | | PROBLEM | | AZ M18 | Slow query execution time | 1y 9d | No | 1 ⤳ | Service: AWS Dynam…   Datacenter: NY1   Env: Production |
| 2019-09-30 13:45:45 | ☐ | Average | | PROBLEM | | AWS N30 | Too many queries per second ? | 1y 9d | Yes | 1 ⤳2 | Service: HTTP balancer   Datacenter: NY1   Env: Production |
| 2019-09-30 13:45:45 | ☐ | Average | | PROBLEM | | AZ M10 | Too many queries per second ? | 1y 9d | Yes | 2 ⤳3 | Service: HTTP balancer   Datacenter: NY1   Env: Production |
| October ○ | | | | | | | | | | | |
| 2019-10-01 12:25:11 | ☐ | High | | PROBLEM | | AWS N90 | Service Redis stopped | 1y 8d | Yes | 1 ⤳1 | Service: Redis   Datacenter: FR2   Env: Staging |
| 2019-10-01 12:25:11 | ☐ | High | | PROBLEM | | AWS N34 | Service Redis stopped | 1y 8d | Yes | 1 ⤳2 | Service: Redis   Datacenter: FR2   Env: Staging |
| Today ○ | | | | | | | | | | | |
| 10:48:11 | ☐ | Information | | PROBLEM | | Zabbix server | Too many processes running on Zabbix server | 3m 39s | No | | Service: Zabbix   OS: Linux   Performance |

# Единый синтаксис

# СУЩЕСТВУЮЩИЕ ОГРАНИЧЕНИЯ

- ⊘ Различный синтаксис для триггерных выражений, вычисляемых и агрегированных проверок

- ⊘ Недоступны некоторые выражения

  - ⊘ min(item1, item2)

  - ⊘ min(item1, yesterday)

- ⊘ Агрегации ограничены лишь группами хостов, нет поддержки тегов

  - ⊘ avg(все "Response time" имеющих tag="Service:API") > 1

  - ⊘ sum(/host/vfs.fs.size[*,free]) < 1T

- ⊘ Лучшие возможности по агрегации позволят более крутой уровень автоматизации для мониторинга кластеров, облаков, K8S, и прочих систем

# СУЩЕСТВУЮЩИЙ СИНТАКСИС

{host:key.func(params)}=0

# НОВЫЙ СИНТАКСИС

func(/host/key, params)

func(/host1/key1, /host2/key2)

func(/host/key[abc,*], params)

func(/host/key?[tag="Service:DB"], period, params)

# ТРИГГЕРНЫЕ ВЫРАЖЕНИЯ

| Сейчас | Новый синтаксис |
|---|---|
| {host:key.func(params)}=0 | func(/host/key, period, params) |
| {host:key.min(#3)}>0 and {host:key.max(1h)}>100 | min(/host/key,#3)>0 and max(/host/key,1h)>100 |
| {host:key.last()}>0 | last(/host/key)>0 |
| {host:key.prev()}>0 | last(/host/key,#2) |
| {host:vfs.fs.size[/,pfree].last()}<10 | last(/host/vfs.fs.size[/,pfree])<10 |
| {host:vfs.fs.size["/var/log",pfree].last()}<10 | last(/host/vfs.fs.size["/var/log",pfree])<10 |
| {host:key.min(1h, 24h)}<10 | min(/host/key, 1h:now-24h)<10 |
| {host:key.trendavg(1M, now/M)} > 1.2*{host:key.trendavg(1M, now/M-1y)} | trendavg(/host/key,1M:now/M) > 1.2*trendavg(/host/key,1M:now/M-1y) |
| N/A: Aggregate function with parameters as expressions | min(min(/host/key,1h), min(/host/key2,1h), 25) |
| N/A: Aggregate function with many parameters of various types, another example | min(min(/host/key,1h), avg(/host/key2,1h), 25) |
| N/A: Using expressions as parameters | min(min(/host/key,1h), avg(/host/key2,1h)*100, 25) |
| N/A: Calculate func() of multiple data sources or expressions | min(min(/host1/key1, 1h),min(/host2/key2,1h)) |
| N/A: Absolute time periods | min(/host/key, 1d:now/d+1d)<br>min(/host/key, 1d:now/d) or trendmin(/host/key,1d:now/d) |

# ВЫЧИСЛЯЕМЫЕ И АГРЕГИРОВАННЫЕ

| | Сейчас | Новый синтаксис |
|---|---|---|
| **Вычисляемые** | func(host:key, 30m) | func(/host/key, 30m) |
| | func1(host1:key1, 30m, param) + func1(host2:key2, 30m, param) | func1(/host1/key1, 30m, param) + func1(/host2/key2, 30m, param) |
| | 100*last("vfs.fs.size[/,free]/last("vfs.fs.size[/,total]") | 100*last(//vfs.fs.size[/,free]) / last(//vfs.fs.size[/,total]) |
| | avg("Zabbix Server:zabbix[wcache,values]",600) | avg(/Zabbix Server/zabbix[wcache,values], 10m) |
| | last("net.if.in[eth0,bytes]")+last("net.if.out[eth0,bytes]") | last(//net.if.in[eth0,bytes])+last(//net.if.out[eth0,bytes]) |
| | N/A: Calculate aggregate function using filter | sum(/host/vfs.fs.size[*,free], 10m) |
| | N/A: Complex filters for data sources | sum(/*/vfs.fs.size[*,free]?[tag="Service" or tag="Importance:High") and (group="Production" or group="Preproduction"]) |
| | N/A: Absolute time periods for trend functions | trendavg(/host/key,1M:now/M) > 1.2*trendavg(/host/key,1M:now/M-1y) |
| **Агрегированные** | grpmin["Servers",qps,avg,5m] | min(avg_foreach(/*/qps?[grpoup="Servers"], 5m)) |
| | grpsum["MySQL Servers","vfs.fs.size[/,total]",last] | sum(last_foreach(/*/qps?[group="MySQL Servers"])) |
| | grpavg[["Servers A","Servers B"],system.cpu.load,last] | avg(last_foreach(/*/qps?[group="Servers A" or group="Servers B"])) |

# ZABBIX ROADMAP

## Roadmap

This page contains an incomplete list of planned functionality that can be updated anytime without prior notice.

V 5.4                                    V 6.0 LTS

### Zabbix 5.4

ETA: Q2, 2021

**General directions**

- Synthetic monitoring
- Advanced event correlation engine
  Ability to perform event filtering, aggregation and root cause analysis
- Keep extending Zabbix monitoring, automatic problem detection, alerting and visualization capabilities
- Keep making Zabbix as modular and flexible as possible
- Make Zabbix truly horizontally scalable monitoring solution ready for the most demanding workloads
- Make it simpler for end users by improving UI

**Synthetic monitoring**

- Support of data collection using complex custom logic
  A new metric type "Script" will allow multi-step scenarios for data collection and service checks written in JavaScript.

**Smart problem and anomaly detection**

- Baseline monitoring
  New algorithms for data processing to allow automatic problem and anomaly detection based on analysis of historical data. No manual setting of

**Enhanced discovery for efficient monitoring of cloud services**

https://www.zabbix.com/roadmap

ZABBIX

Спасибо!