From LLD to SuperDiscovery

How to involve developers in monitoring process

Ilya Ableev · 16th of September

badoo

badoo tech

Who am I?

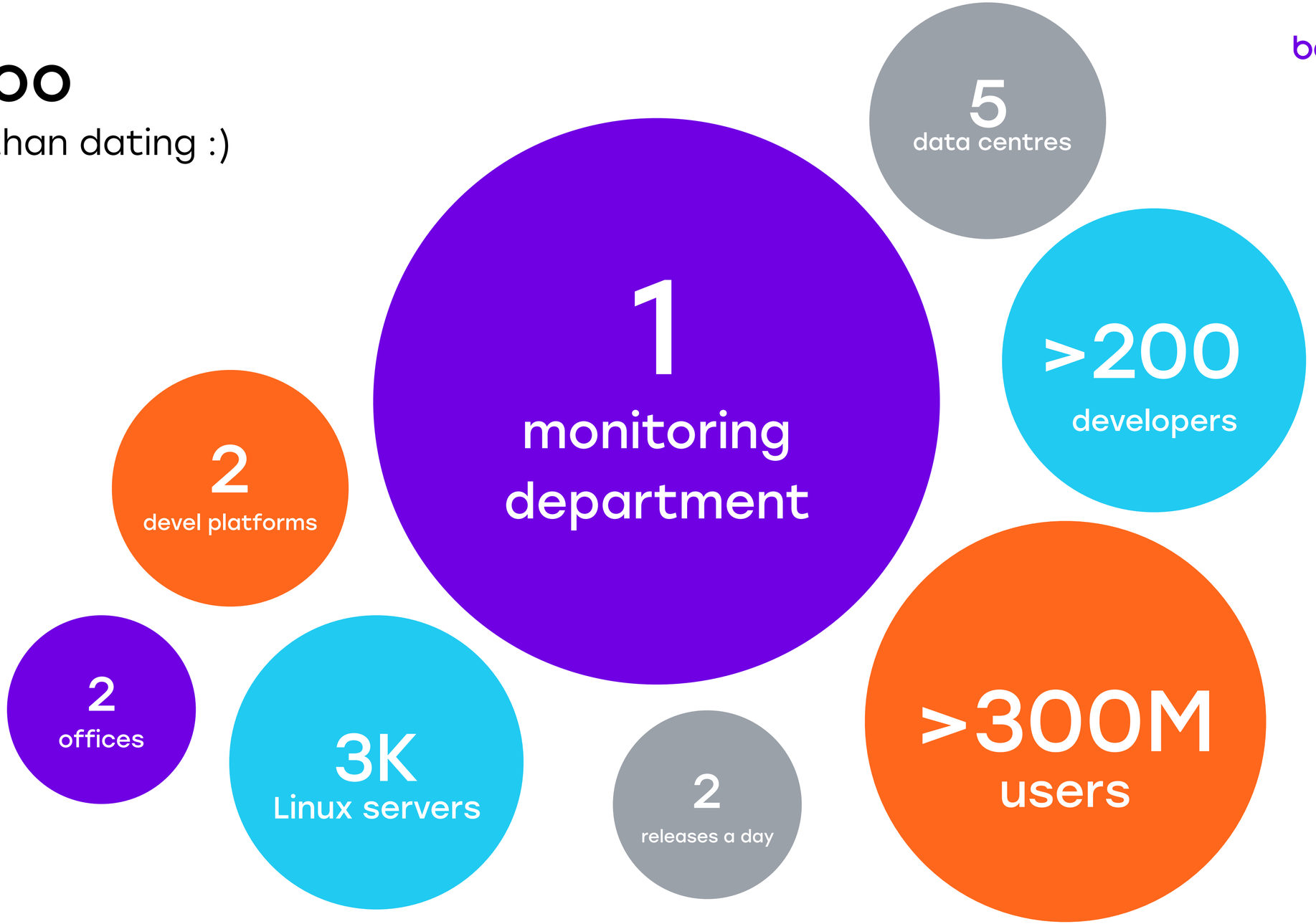
badoo_tech

- **Ilya Ableev**, Head of Monitoring Department in Badoo
- Zabbix experience – 7 years (certified specialist)
- Zabbix Moscow Meetup leader for 3 years
- Author of “Zabbix-in-Telegram” tool

Badoo

Bigger than dating :)

badoo_tech



badoo_tech

Monitoring Department in Badoo

is:

- Controlling everything’s state
- Detecting problems, researching these
- **Notifying** responsible people
- Escalating ongoing problems
- Keeping an eye on things

And doing something with Zabbix :)

Monitoring Department in Badoo

badoo_tech

badoo_tech

Zabbix in Badoo

monitors:

Availability

Servers, network equipment (simple checks)

Hardware statistics

Free space, memory, CPU, load average (standard agent check)

Services

Daemons, MySQL, Nginx, PHP-FPM, Docker, Tarantool, a lot of in-house services

Application metrics

Queues, RPS, response time, consistency, data freshness

User activity

Online users, registrations, messages count, notifications deliverability

And everything we want to know about...

How things became monitored?

badoo_tech

When do we need it:

- Thought about monitoring before releasing to production
- We faced a problem, decided to check it constantly

How things became monitored?

badoo_tech

What to do:

- Describe requirements (metrics, thresholds, severities, docs)
- Write monitoring code
- Implement it to Zabbix – templates, hosts, items, formulas, triggers
- Enable **notifications** – users media, actions

Who is being notified?

badoo_tech

Operations team:

- One primary engineer on duty (“many to one”)
- Quite standard environment (N metrics multiply amount of servers)
- Easy way of managing thresholds (templates)
- No notifications: our team investigates problems, escalates those

Who is being notified?

badoo_tech

Developers (large departments and small teams):

- A lot of different applications with completely different logic
- Even more metrics with different behaviour
- Reviews required, something changes or becomes old
- List of responsible people should be up to date: ex-employees, newcomers, vacations, collaboration
- Notifications should remain “fresh”

How was this handled before?

badoo_tech

- Need to get info from developers about app's logic
- Write scripts on our own: bash, perl, php, python, sql, telnet, grep...
- Deploy scripts
- Add items to Zabbix: `system.run[script.php]`
- Clarify limits for triggers, fix them after flapping (**continuously**!)
- Create notifications for everything and everyone, change those from time to time by request from developers (a bit annoying)
- **Click**, click, click, every time we change something

What could have been improved?

badoo_tech

- We have a squad of developers who could help us write the code
- **No need to write code**
- **No need to deploy the code**
- **No need to dig inside an app**

} **saving tons of time**

- Still should deploy written code
- Still change it in Zabbix
- Still support notifications
- Still clicking

Initial results

badoo_tech

- We got rid of maintaining monitoring code
- Developers could create and change it code by themselves

- But we are still supporting items, triggers, notifications – not fast, not flexible

How to become faster and more flexible?

badoo_tech

- Provide **admin** access to Zabbix web for developers
- Share knowledge about Zabbix API (allocate another instance)

Both were rejected :)

- Needed to learn what are... hosts, items, triggers, functions, actions...
- Too hard to understand interface
- New universe of documentation about API

Tried low level discovery as an experiment

badoo_tech

- Got task to monitor queue processor
- Perfectly fit LLD

Item:

count[{#QUEUE}] – queue size (trapper)

Function:

{queue:**count[[{#QUEUE}]**.last(0)}>**{#LIMIT}**

Trigger:

Queue size of **{#NAME}** > **{#LIMIT}**

```
{
  "data": [{
    "{#QUEUE}": 1,
    "{#NAME}": "name 1",
    "{#LIMIT}": 1000
  }, {
  ...
  }, {
    "{#QUEUE}": 99,
    "{#NAME}": "name 99",
    "{#LIMIT}": 5000
  }]
}
```

Expanding the experiment

badoo_tech

- **Added contacts to LLD!!!** – easily changeable by developers
- Split limits by two severities, used only as indicator
- Item “status[]” – to allow developers to mute trigger

```
{
    "data": [{
        "{#ID}": 1,
        "{#NAME}": "app stuff",
        "{#RESPONSIBLE}": "user1,user2,manager1",
        "{#PHONES}": "7915123456789,7901987654321"
    }]
}
```

badoo_tech

Unifying solution: problems to solve

Hosts

- **Problem**: we can’t create items or trigger without hosts
- **Resolution**: create “fake” hosts (“virtual”) to attach metric to them; developers should provide list of available hosts; hosts could bring some valuable information, e.g. domain zone

Example:

pushopens.dc1 – push opening rates in datacenter 1

emailclicks.dc2 – clicks from emails in datacenter 2

badoo_tech

Unifying solution: problems to solve

Metrics (LLD keys, items)

- **Problem**: zabbix must generate items based on LLD's JSON
- **Resolution**: developers are maintaining "core" code to provide JSON arrays by host

  array keys: id, item name, trigger name, emails of responsible people, their numbers (for sms)

Example:

php /opt/www/getJson.php --host=**pushopens.dc1**

php /opt/www/getJson.php --host=**emailclicks.dc2**

badoo_tech

Unifying solution: problems to solve

Data (history)

- **Problem**: we need to get data somehow
- **Resolution**: developers are maintaining code which pushes new data to us from our cloud of scripts using **zabbix_sender**;

  items: count, status, limit_{warning, high, disaster} (used for notifications)

How it looks

badoo_tech

Discovery rules

Create discovery rule

All templates / TZabbix_SuperDiscovery_TEMPLATE | Applications 1 | Items | Triggers | Graphs | Screens | Discovery rules 1 | Web scenarios

Name ▲	Items	Triggers	Graphs	Hosts	Key	Interval	Type	Status
SuperDiscovery	Item prototypes 5	Trigger prototypes 3	Graph prototypes	Host prototypes	system.run["php /opt/www/getJson.php --host={HOST.DNS}"]	5m	Zabbix agent	Enabled

Displaying 1 of 1 found

How it looks

badoo_tech

Item prototypes

Create item prototype

All templates / TZabbix_SuperDiscovery_TEMPLATE Discovery list / SuperDiscovery Item prototypes 5 Trigger prototypes 3 Graph prototypes
Host prototypes

Name ▲	Key	Interval	History	Trends	Type	Applications	Create enabled
count_of_{#NAME}	count[{#ID}]		1d	0d	Zabbix trapper	SuperDiscovery	Yes
limit_disaster_{#NAME}	limit_disaster[{#ID}]		1d	0d	Zabbix trapper	SuperDiscovery	Yes
limit_fail_{#NAME}	limit_fail[{#ID}]		1d	0d	Zabbix trapper	SuperDiscovery	Yes
limit_warning_{#NAME}	limit_warning[{#ID}]		1d	0d	Zabbix trapper	SuperDiscovery	Yes
status_of_{#NAME}	status[{#ID}]		1d	0d	Zabbix trapper	SuperDiscovery	Yes

Displaying 5 of 5 found

How it looks

badoo_tech

Trigger prototypes

Create trigger prototype

All templates / TZabbix_SuperDiscovery_TEMPLATE Discovery list / SuperDiscovery Item prototypes 5 Trigger prototypes 3 Graph prototypes
Host prototypes

Severity	Name ▲	Expression	Create enabled
Average	{#TRIGGER} @e: [{#RESPONSIBLE}] @t: [{#PHONES}]	{TZabbix_SuperDiscovery_TEMPLATE:status[{#ID}].**count**(#5,1)}=5 and {TZabbix_SuperDiscovery_TEMPLATE:count[{#ID}].**date**(0)}>0 and {TZabbix_SuperDiscovery_TEMPLATE:limit_warning[{#ID}].**date**(0)}>0	Yes
High	{#TRIGGER} @e: [{#RESPONSIBLE}] @t: [{#PHONES}]	{TZabbix_SuperDiscovery_TEMPLATE:status[{#ID}].**count**(#5,2)}=5 and {TZabbix_SuperDiscovery_TEMPLATE:count[{#ID}].**date**(0)}>0 and {TZabbix_SuperDiscovery_TEMPLATE:limit_fail[{#ID}].**date**(0)}>0	Yes
Disaster	{#TRIGGER} @e: [{#RESPONSIBLE}] @t: [{#PHONES}]	{TZabbix_SuperDiscovery_TEMPLATE:status[{#ID}].**count**(#5,3)}=5 and {TZabbix_SuperDiscovery_TEMPLATE:count[{#ID}].**date**(0)}>0 and {TZabbix_SuperDiscovery_TEMPLATE:limit_disaster[{#ID}].**date**(0)}>0	Yes

Displaying 3 of 3 found

Example

badoo_tech

Too few amount of pushes @e: [ableev,zorin,lobashev] @t: [7915123456789,7900987654321]

Trigger name: “Too few amount of pushes”

Email: ableev@..., zorin@... and lobashev@... – **one** email with three recipients (hello, ZBXNEXT-3126)

SMS: 7915123456789 and 7900987654321

Summary: flow

badoo_tech

- **Developers** are responsible for metrics and thresholds
- They can measure “problem”, they can switch something off for awhile
- **Monitoring** provides them with a semi-API, one dedicated template for each team
- No need to configure actions, those rules apply by developers as well
- Ex-employees/newcomers removed and added automatically
- **No need to create zabbix users with media for one action anymore!**

Summary: technical details

badoo_tech

- Alert script parses **@t** and **@e** arrays from trigger description
- Only one email – killer feature for non-zabbix users
- There is a limitation for amount of keys (def. 200), in case if someone goes “crazy”
- Not only one template: we can disable “bad” templates (flapping, etc)

Questions?

Me: @ableev (Tg, GitHub, Twitter, FB)

Zabbix Moscow Meetup: https://zabbix.moscow

https://techblog.badoo.com

badoo

badoo tech