

РЕД  
СЛЁРМ

+

  
Southbridge

# Удалённый мониторинг: Zabbix



# Евгений Терешков

Инженер компании Southbridge

# План

- Зачем мониторинг, какой мониторинг?
- Устройство
- Настройка и эксплуатация
- Проблемы

# Зачем мониторинг, какой мониторинг?

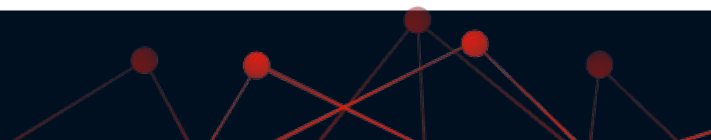
# Хотелки

- Нужно узнавать о проблемах, пока они не стали факапами
- Мониторинг объектов из разных локаций, клиенты по всему миру, ведь сети не идеальны
- Нужны тренды, планировать нагрузку/рост
- Нужны сложные проверки, зависимые друг от друга, упрощать поиск причин проблем
- Нужна автоматическая адаптация к среде выполнения, автоматизировать рутину
- Нужна возможность сопоставлять данные разных хостов чтобы искать причины, планировать рост
- Нужно информировать клиентов, идя им навстречу

# Требования к МОНИТОРИНГУ

- Масштабируемый
- Распределённый
- С контролем доступа и разделением прав
- Богатый функционалом
- Гибко и просто настраиваемый
- Автоматизируемый

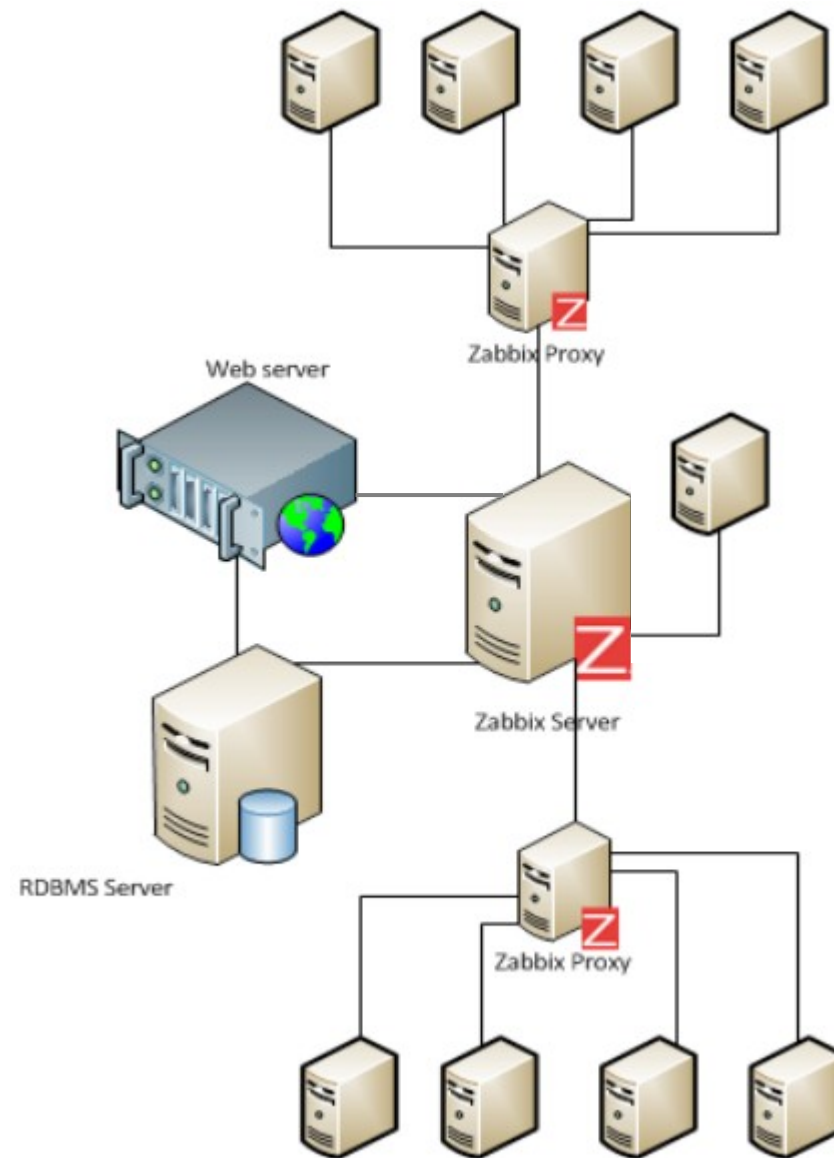
# Устройство





# Общая архитектура

- Агенты
- Сервер
- Прокси
- БД
- ФРОНТЭНД
- Хосты
- Итемы
- Триггеры
- Действия





# Сводка в цифрах

- Zabbix: prev LTS (3.4)
- Хосты: ~ 2k
- Итемы: ~ 300k
- Прокси: ~ 30
- Пользователей: ~ 200
- Производительность: ~ 2.5 knvps
- БД: ~ 0.6 Тб (300Гб SSD, 300Гб HDD)

| Status of Zabbix                                   |        |                           | ...               |
|--|--------|---------------------------|-------------------|
| Parameter  | Value  | Details                   |                   |
| Zabbix server is running                           | Yes    | localhost:10051           |                   |
| Number of hosts (enabled/disabled/templates)       | 2021   | 1946 / 12 / 63            |                   |
| Number of items (enabled/disabled/not supported)   | 286275 | 259780 / 15415 / 11080    |                   |
| Number of triggers (enabled/disabled [problem/ok]) | 78671  | 46660 / 32011 [8 / 46652] |                   |
| Number of users (online)                           | 210    | 9                         |                   |
| Required server performance, new values per second | 2486.9 |                           |                   |
|  |        |                           | Updated: 09:31:26 |

# Оборудование

- Xeon E5 3.5ГГц, 6 ядер (LA 4)
- 252 Гб (занято меньше половины)
- SSD RAID1, HDD RAID1
- Сеть 1 Гбит/с

# Настройка и эксплуатация



# Конфигурация агента

- SCM кладёт в `/etc/zabbix/zabbix_agentd.conf` общую для всех конфигурацию
- В `/etc/zabbix/00_sb_zabbix_agentd.conf` SCM кладёт часть специфичную для клиента
- В `/etc/zabbix/01_sb_zabbix_agentd.conf` scm кладёт hostmetadata и **ТОЛЬКО** сюда можно добавлять специфические для клиента/хоста параметры (через SCM)

# Пример конфигурации агента

```
[root@ds01.slurm.io /etc/zabbix]# cat zabbix_agentd.conf
```

```
PidFile=/var/run/zabbix/zabbix_agentd.pid
```

```
LogFile=/var/log/zabbix/zabbix_agentd.log
```

```
LogFileSize=0
```

```
Server=127.0.0.1
```

```
ServerActive=127.0.0.1
```

```
Include=/etc/zabbix/zabbix_agentd.d/*.conf
```

```
Include=/etc/zabbix/00_sb_zabbix_agentd.conf
```

```
Include=/etc/zabbix/01_sb_zabbix_agentd.conf
```

```
[root@ds01.slurm.io /etc/zabbix]# cat 00_sb_zabbix_agentd.conf
```

```
StartAgents=5
```

```
Server=127.0.0.1,172.20.1.10
```

```
ServerActive=127.0.0.1
```

```
EnableRemoteCommands=1
```

```
LogRemoteCommands=1
```

```
Timeout=30
```

```
[root@ds01.slurm.io /etc/zabbix]# cat 01_sb_zabbix_agentd.conf
```

```
HostMetadata=ds.slurm.io
```

# Конфигурация сервера

```
StartPollers=1000  
StartEscalators=60  
StartPollersUnreachable=300  
StartPingers=200  
StartHTTTPollers=150  
StartProxyPollers=100  
StartDBSyncers=50  
CacheSize=3G  
HistoryCacheSize=2G  
HistoryIndexCacheSize=2G  
TrendCacheSize=2G  
ValueCacheSize=3G  
Timeout=30  
UnreachablePeriod=120  
UnavailableDelay=60  
UnreachableDelay=60
```

# UserParameter

- Подключение внешнего кода в качестве итема агента
- Кладём конфигурацию агента в */etc/zabbix/zabbix\_agentd.d/*
- Внешние скрипты кладём в */etc/zabbix/scripts/*



# Пример каталогов агента

```
[root@ds01.slurm.io /etc/zabbix]# ls -la scripts/ zabbix_agentd.d/
scripts/:
итого 168
drwxr-xr-x. 3 root root 4096 июн 22 13:35 ./
drwxr-xr-x. 4 root root 4096 окт 4 19:27 ../
-rwxr-xr-x 1 root root 3338 map 2 2018 apache-stats.sh*
-rwxr-xr-x 1 root root 2469 map 2 2018 elasticsearch.py*
-rwxr-xr-x 1 root root 538 map 2 2018 find_dbname.sh*
-rwxr-xr-x 1 root root 1750 map 2 2018 find_dbname_table.sh*
-rwxr-xr-x 1 root root 1771 map 2 2018 gearman.py*
-rwxr-xr-x 1 root root 1237 map 2 2018 get_mysql_stats_wrapper.sh*
-rwxr-xr-x 1 root root 1929 map 2 2018 iostat.sh*
-rwxr-xr-x 1 root root 1522 map 2 2018 nginx-stats.sh*
-rwxr-xr-x 1 root root 13865 июн 21 02:33 ntplib.py*
-rwxr-xr-x 1 root root 1011 июн 21 02:33 ntpoffset.py*
-rwxr-xr-x 1 root root 1068 map 2 2018 php-fpm-check.sh*
-rwxr-xr-x 1 root root 385 map 2 2018 php_fpm_discovery.sh*
-rwxr-xr-x 1 root root 1805 map 2 2018 postfix.sh*
drwxr-xr-x. 2 root root 4096 июн 26 17:28 rabbitmq/
-rwxr-xr-x 1 root root 1825 map 2 2018 rdiff_backup_discovery.sh*
-rwxr-xr-x 1 root root 3551 map 2 2018 rdiff_backup_stat.sh*
-rwxr-xr-x 1 root root 6081 map 2 2018 redis.sh*
-rwxr-xr-x 1 root root 262 map 2 2018 remote_fping.sh*
-rw-r--r-- 1 root root 58079 map 2 2018 ss_get_mysql_stats.php
-rwxr-xr-x 1 root root 1221 map 2 2018 supervisord_trapper.sh*
-rwxr-xr-x 1 root root 4747 май 28 12:10 zbx-highload-report.sh*
```

```
zabbix_agentd.d/:
итого 136
drwxr-xr-x. 2 root root 4096 окт 4 19:27 ./
drwxr-xr-x. 4 root root 4096 окт 4 19:27 ../
-rw-r--r-- 1 root root 174 map 2 2018 ansible_role_discover.conf
-rw-r--r-- 1 root root 122 map 2 2018 apache-status.conf
-rw-r--r-- 1 root root 136 map 2 2018 crond.conf
-rw-r--r-- 1 root root 121 map 2 2018 elasticsearch.conf
-rw-r--r-- 1 root root 439 map 2 2018 gearman.conf
-rw-r--r-- 1 root root 121 map 2 2018 iface_error.conf
-rw-r--r-- 1 root root 115 map 2 2018 iostat.conf
-rw-r--r-- 1 root root 147 map 2 2018 memcached.conf
-rw-r--r-- 1 root root 119 map 2 2018 nginx-status.conf
-rw-r--r-- 1 root root 107 июн 21 02:33 ntpoffset.conf
-rw-r--r-- 1 root root 25044 map 2 2018 percona.conf
-rw-r--r-- 1 root root 236 map 2 2018 php-fpm-status.conf
-rw-r--r-- 1 root root 106 map 2 2018 postfix.conf
-rw-r--r-- 1 root root 15398 map 2 2018 postgresql.conf
-rw-r--r-- 1 root root 532 map 2 2018 rabbitmq.conf
-rw-r--r-- 1 root root 375 map 2 2018 rdiff-backup.conf
-rw-r--r-- 1 root root 357 июл 23 16:05 redis.conf
-rw-r--r-- 1 root root 110 map 2 2018 remote_fping.conf
-rw-r--r-- 1 root root 1001 map 2 2018 sockstat.conf
-rw-r--r-- 1 root root 519 map 2 2018 supervisor.conf
-rw-r--r-- 1 root root 9816 map 2 2018 userparameter_pgsql.conf
```

# Административные скрипты

Для работы со скриптами нужно клонировать репозиторий:

```
git clone git@gitlab.slurm.io:red/zabbix.git
```

Далее можно запускать программы из их подкаталогов. При каждом запуске проверяется наличие обновлений и если они есть, выполняется команда **git pull**. После обновления нужно перезапустить скрипт.

Список подкаталогов скриптов:

- `zbx_add_proj` – добавление нового проекта
- `zbx_remove_project` – удаление проекта
- `zbx_create_sslvalid_check` – создание проверки срока ssl сертификата
- `zbx_create_domain_check` – создание проверка срока делегирования домена
- `zbx_create_web_check` – создание веб-проверки
- `zbx_add_maintenance` – постановка хостов на обслуживание

# Авторегистрация агентов

| <input type="checkbox"/> Name ▲                        | Conditions                            | Operations  |
|--|---------------------------------------|---|
| <input type="checkbox"/> Auto registration ds.slurm.io | Host metadata like <i>ds.slurm.io</i> | Add host<br>Add to host groups: slurm.io<br>Link to templates: IOstat_agent, SMART_agent, Template OS Linux |
| <input type="checkbox"/> Auto registration vs.slurm.io | Host metadata like <i>vs.slurm.io</i> | Add host<br>Add to host groups: slurm.io<br>Link to templates: Template OS Linux                            |

- Скрипт `zbx_add_proj.py --prname "имя.проекта"` создаёт группу хостов, **действия авторегистрации**, пингователь, пользователя
- SCM устанавливает на хост агента, прописывая **HostMetadata/Server/ServerActive**
- Агент сам регистрируется на сервере/прокси
- Действия авторегистрации создают хост в мониторинге
- Хост можно править вручную

# Низкоуровневое обнаружение

## Discovery rules

All templates / Template OS Linux Applications 11 Items 49 Triggers 23 Graphs 7 Screens 1 **Discovery rules 2** Web scenarios

| <input type="checkbox"/> Name ▲                       | Items             | Triggers             | Graphs             | Hosts           | Key              | Interval |
|---|-------------------|----------------------|--------------------|-----------------|------------------|----------|
| <input type="checkbox"/> Mounted filesystem discovery | Item prototypes 5 | Trigger prototypes 3 | Graph prototypes 1 | Host prototypes | vfs.fs.discovery | 10m      |
| <input type="checkbox"/> Network interface discovery  | Item prototypes 3 | Trigger prototypes 3 | Graph prototypes 1 | Host prototypes | net.if.discovery | 10m      |

- Опрашиваем список смонтированных ФС
- Создаём итемы/триггеры/графики по шаблону, если надо
- Собираем данные/зажигаем триггеры
- Если ФС исчезла, со временем забываем её

# Прокси

| Name  | Applications   | Items    | Triggers | Graphs | Discovery   |
|---|----------------|----------|----------|--------|-------------|
| <input type="checkbox"/> zabbix-proxy.southbridge.ru: icmp_rus.southbridge.ru | Applications 2 | Items 62 | Triggers | Graphs | Discovery 1 |

| Name ▲  | Triggers | Key          | Interval | History | Trends |
|---|----------|--------------|----------|---------|--------|
| ICMP_HOST_DISCOVERY: ds01.southbridge.ru              |          | icmpping     | 1m       | 90d     | 365d   |
| ICMP_HOST_DISCOVERY: ICMP loss on ds01.southbridge.ru |          | icmppingloss | 1m       | 90d     | 365d   |

- Скрипт заведения проекта создаёт хост-пингователь. Задача в кроне создаёт пингователь по необходимости (при ручных вмешательствах). Это пустой хост с шаблоном **icmp\_rus** и макросом **\$HG** (имя проекта)
- Шаблонный LLD пингователя (внешний скрипт ) по **\$HG** ищет публичные пингуемые хосты и начинает их пинговать (без триггеров)
- При срабатывании триггера о недоступности, скрипт действия подтягивает данные с пингователя и создаёт тикет (если частично недоступен, с задержкой) или аварию (если полностью недоступен):

**Описание**

Хост: [redacted]  
Текущее значение с основного сервера: Недоступен  
☐ График с основного сервера  
Текущее значение с резервного сервера: Недоступен  
☐ График с резервного сервера  
Дата: 2018.05.11 13:52:51  
☐ MTR  
☐ КБ

# Оповещения/фабрика

Вызов серверного скрипта из действия триггера Zabbix:

**ZABBIX** Monitoring Inventory Reports Configuration Administration

Host groups Templates Hosts Maintenance **Actions** Event correlation Discovery Services

Actions

Action Operations Recovery operations Acknowledgement operations

Name: 01\_Create\_task\_In\_Redmine!!!

Type of calculation: And/Or A and (B or C)

| Label | Name   | Action                 |
|-------|--|------------------------|
| A     | Maintenance status not in <i>maintenance</i> | <a href="#">Remove</a> |
| B     | Trigger severity = <i>Disaster</i>           | <a href="#">Remove</a> |
| C     | Trigger severity = <i>High</i>               | <a href="#">Remove</a> |

New condition: Trigger name like

[Add](#)

Enabled ☒

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

Default operation step duration: 15m

Default subject:

Default message:

Pause operations while in maintenance ☒

| Operations | Steps                               | Details     | Start in | Duration | Action |
|------------|-------------------------------------|-------------|----------|----------|--------|
| 1          | Run remote commands on current host | Immediately | Default  | Edit     | Remove |

Run custom commands on Zabbix server:

```
/var/lib/zabbix/zbx_scripts/zbx_create_task/main.py --hostgroup '{TRIGGER.HOSTGROUP.NAME}' --status '{TRIGGER.STATUS}' --severity '{TRIGGER.SEVERITY}' --hostname '{HOST.NAME}' --triggername '{TRIGGER.NAME}' --itemid '{ITEM.ID}' --value '{ITEM.VALUE}' --valueadditional '{ITEM.VALUE2}' --date '{EVENT.DATE}' --time '{EVENT.TIME}' --tag '{TRIGGER.DESRIPTION}' --url '{TRIGGER.URL}' >> /var/log/zabbix/redmine_api_create_task.log 2>&1
```

```
/var/lib/zabbix/zbx_scripts/zbx_create_task/main.py --hostgroup '{TRIGGER.HOSTGROUP.NAME}' --status '{TRIGGER.STATUS}' --severity '{TRIGGER.SEVERITY}' --hostname '{HOST.NAME}' --triggername '{TRIGGER.NAME}' --itemid '{ITEM.ID}' --value '{ITEM.VALUE}' --valueadditional '{ITEM.VALUE2}' --date '{EVENT.DATE}' --time '{EVENT.TIME}' --tag '{TRIGGER.DESRIPTION}' --url '{TRIGGER.URL}' --escalation >> /var/log/zabbix/redmine_api_create_task.log 2>&1
```

КБ

# Веб-проверки/домены/ сертификаты

- При создании проекта создаётся пустой хост с именем **`check_content.имя.проекта`** куда потом привязываются проверки (без прокси)
- `zbx_create_domain_check/domain_check.py --prname "southbridge.ru" --domain "southbridge.ru"`
- `zbx_create_sslvalid_check/zbx_create_sslvalid_check.py --prname "southbridge.ru" --domain "southbridge.ru"`
- `zbx_create_web_check/zbx_create_web_check.py --url "https://southbridge.ru" --prname "southbridge.ru" --pattern "TEST" --desc "Test task (#12345)"` (автоматически создаст проверку сертификата для схемы HTTPS).
- Регулярно выполняем групповую проверки итемов, для которых не созданы триггеры
- Создаём тикеты при истечении срока или "падении" сайта



# БД для прокси

- Для низкого потребления ресурсов прокси нужно использовать БД SQLite3
- Проблема: БД прокси регулярно повреждается при пропадании питания
- Решение:
  - 1) `tmpfs /var/lib/zabbix/db tmpfs size=50m 0 0`
  - 2) `DBName=/var/lib/zabbix/db/zabbix_proxy.sqlite`

# Партиционирование БД

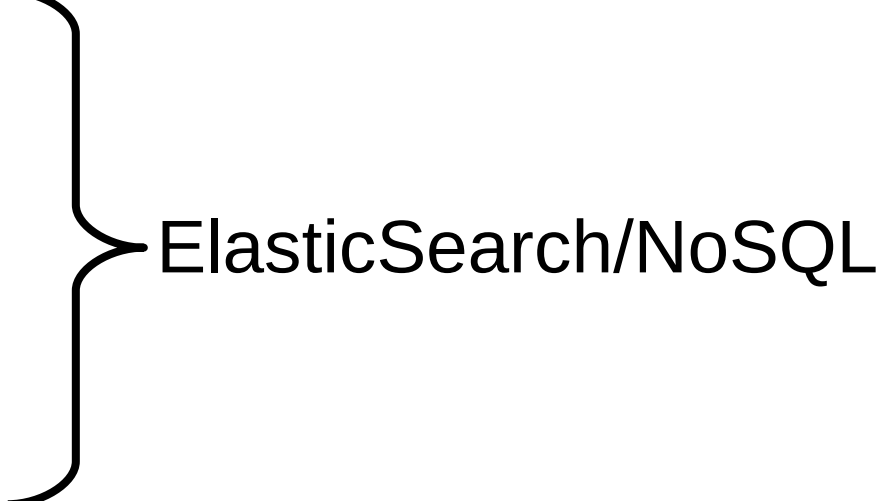
- РСУБД не разрабатывалась для временных рядов
- Удалять много данных часто - боль
- Делим входящие данные в РСУБД на "куски" по признаку времени
- Выключили встроенный хаускипер для истории/трендов, удаляем "куски" сами
- Получаем ускорение поиска по времени - всегда известно, в каком "куске" смотреть
- Разнесли "куски": старые на HDD, свежие на SSD

[Подробнее про партиционирование](#)

# Проблемы



# Текущие проблемы

- Медленные выборки
  - Требуется много места
  - Партиционирование официально не поддерживается
  - Страшненькие не интерактивные графики
  - Сложность/негибкость анализа/сопоставления данных
  - Нет типа итема для HTTP-запроса (4.x!)
  - Не очень удобное API
  - Нарушения связности ДЦ с мониторингом — БОЛЬ :-)
- 
- ElasticSearch/NoSQL

# Источники информации

- Официальная документация
- КБ, что делать по конкретным триггерам
- КБ, административные скрипты
- КБ в целом

# Вопросы?

