Upgrading Zabbix for a better sleep
Who am I?

Worked at Zabbix for some wonderful time

Two books on Zabbix

- Zabbix 1.8 network monitoring
- Zabbix network monitoring second edition

Writing articles on www.zabbixbook.com

OpenStreetMap mapper
This experience based on...

Zabbix usage in one division of Nokia

Disclaimer: usage is not endorsement
Our environment

Development

Staging

Production 1

Production 2
Why upgrade?
Why upgrade?

New features, improvements
Support running out (both LTS and non-LTS)
Security improvements
Reality kicks in

No time for support systems (think Zabbix)
Non-production systems upgraded
Production moving to another DC -> new install
Chasing shiny things

Zabbix team done a great job on 3.2 + 3.4
Decided to go for 3.4 (non-LTS)
Commitment to move to 4.0
Upgrade process
What does an upgrade involve

Selling the upgrade
Actual upgrade
Potential problems
Selling the update

Awesome management understands the need
A carrot has never hurt anybody
No escalation, please
Improved Jira integration

**Acknowledge**
- Only selected event
- Selected and all unacknowledged PROBLEM events (7 events)
- Selected and all unacknowledged events (13 events)

**Jira integration**

- Search JIRA
- Create JIRA issue

**Actions**
- Acknowledge
- Cancel
# Actionsim by Volker


### Messages
<table>
<thead>
<tr>
<th>Action</th>
<th>Time passed</th>
<th>Time</th>
<th>Step</th>
<th>Step config</th>
<th>Operation</th>
<th>Type</th>
<th>Recipient</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>test action</td>
<td>0</td>
<td>09/12/2017 01:30:00 PM</td>
<td>1</td>
<td>1</td>
<td>To user</td>
<td>Any</td>
<td>user (User Name)</td>
<td>[Error Message]</td>
</tr>
</tbody>
</table>

### Remote commands
<table>
<thead>
<tr>
<th>Action</th>
<th>Time passed</th>
<th>Time</th>
<th>Step</th>
<th>Step config</th>
<th>Operation</th>
<th>Command</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

No remote commands

### Action condition matches

<table>
<thead>
<tr>
<th>Match</th>
<th>Label</th>
<th>Condition type</th>
<th>Trigger Value</th>
<th>Operator</th>
<th>Condition Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRUE</td>
<td>test action</td>
<td>test action A and B</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRUE</td>
<td>A</td>
<td>Maintenance status</td>
<td>not in</td>
<td>is</td>
<td>not in</td>
</tr>
<tr>
<td>TRUE</td>
<td>B</td>
<td>Triggervalue</td>
<td>PROBLEM</td>
<td>=</td>
<td>PROBLEM</td>
</tr>
<tr>
<td>DISABLED</td>
<td>Report problems to Zabbix administrators</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Potential concerns

Zabbix proxies – not yet
Java GW (protocol changed in 3.4)
Agents – backwards compatible, yay
What could go wrong

DB upgrade could fail - not very common
SystemD could restart server during the upgrade
DB upgrade

SQL scripts long time ago, server now

7344:20170630:183622.330 required mandatory version: 03000000
7344:20170630:183622.330 starting automatic database upgrade
7344:20170630:183622.335 completed 0% of database upgrade
7344:20170630:183622.372 completed 1% of database upgrade
...
7344:20170630:183623.698 completed 99% of database upgrade
7344:20170630:183623.698 completed 100% of database upgrade
7344:20170630:183623.698 database upgrade fully completed
DB upgrade magic

More powerful, a bit harder to debug

```c
static int DBpatch_3030007(void)
{
    DB_ROW row;
    DB_RESULT result;
    zbx_vector_uint64_t dserviceids;
    zbx_uint64_t dserviceid;
    int ret = SUCCEED;

    zbx_vector_uint64_create(&dserviceids);

    /* After dropping fields type and key_ from table dservices there is no guarantee that a unique
     * index with fields dcheckid, ip and port can be created. To create a unique index for the same
     * fields later this will delete rows where all three of them are identical only leaving the latest. */
    result = DBselect(
        "select ds.dserviceid"
        " from dservices ds"
        " where not exists (" 
        " select null"
        " from dchecks dc"
        " where ds.dcheckid = dc.dcheckid"
        " and ds.type = dc.type"
        " and ds.key_ = dc.key_
        ")");

    while (NULL != (row = DBfetch(result)))
```
OS reinstall, uh

Historical baggage

- cronjobs
- local script changes, not committed to svn
- Perl, Python modules used by various scripts
- SNMP MIB files
- ...forgotten things
Other changes, waiting to bite

Notification content updates
Latest data, acknowledge, other URLs changed

-- default message
update actions set def_longdata=
  replace(def_longdata, 'oldstring', 'newstring');

-- default recovery message; was recovery_msg in older versions
update actions set r_longdata=
  replace(r_longdata, 'oldstring', 'newstring');

-- operation message override
update opmessage set message=
  replace(message, 'oldstring', 'newstring');
Getting along
Delayed alerts

Alerts delayed – hard to debug on 2.2
3.4 might help, but until then...
Mail roundtrip monitoring

Send a timestamped email

From: zabbix
To: public_mailbox
Subject: Zabbix email roundtrip monitoring

1505307315

Retrieve with getmail

[destination]
type = MDA_external
path = /home/zabbix/bin/roundtrip_email

Timestamp in the body <-> header data
Email delays exposed

Email roundtrip times (3d 12h 10m)

- Email roundtrip, time to local relay: [max] 1s, min 0, avg 0, max 1s
- Email roundtrip, time to relay: [max] 1s, min 0, avg 0, max 30s
- Email roundtrip, time to remote: [max] 1s, min 0, avg 0, max 31s

Data from trends. Generated in 0.21 sec

Zabbix Conference 2017
Alert count monitoring

How to know how many are we sending?
Query API for all historical data
  • Send in 10 minute steps
  • Further runs pick up from the previous time

LLD to create items for all media types

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZBX_ALERT_PENDING</td>
<td>0</td>
</tr>
<tr>
<td>ZBX_ALERT_SUCCEEDED</td>
<td>1</td>
</tr>
<tr>
<td>ZBX_ALERT_FAILED</td>
<td>2</td>
</tr>
<tr>
<td>Zabbix alerts</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>Failed alerts to &quot;Email&quot;</td>
<td></td>
</tr>
<tr>
<td>Failed alerts to &quot;Send to Pager (short)&quot;</td>
<td></td>
</tr>
<tr>
<td>Failed alerts to &quot;Send to Prowl&quot;</td>
<td></td>
</tr>
<tr>
<td>Failed alerts to &quot;Send to Pushover&quot;</td>
<td></td>
</tr>
<tr>
<td>Pending alerts to &quot;Email&quot;</td>
<td></td>
</tr>
<tr>
<td>Pending alerts to &quot;Send to Pager (short)&quot;</td>
<td></td>
</tr>
<tr>
<td>Pending alerts to &quot;Send to Prowl&quot;</td>
<td></td>
</tr>
<tr>
<td>Pending alerts to &quot;Send to Pushover&quot;</td>
<td></td>
</tr>
<tr>
<td>Succeeded alerts to &quot;Email&quot;</td>
<td></td>
</tr>
<tr>
<td>Succeeded alerts to &quot;Send to Pager (short)&quot;</td>
<td></td>
</tr>
</tbody>
</table>
Staging alerts

Succeeded alerts to "Email" (2m 6d 23h)

Data from trends. Generated in 0.28 sec

Zabbix Conference 2017
Too many alerts

-- determine the number of pending alerts
select count(*) from alerts where status=0 and alerttype=0;
Set them all to failed

-- determine the number of pending alerts
select count(*) from alerts where status=0 and alerttype=0;
-- set all the unsent alerts to failed
update alerts set status=2 where status=0 and alerttype=0;
Possible values

-- determine the number of pending alerts
select count(*) from alerts where status=0 and alerttype=0;
-- set all the unsent alerts to failed
update alerts set status=2 where status=0 and alerttype=0;

Status*:
ALERT_STATUS_NOT_SENT => 0
ALERT_STATUS_SENT => 1
ALERT_STATUS_FAILED => 2

Type:
ALERT_TYPE_MESSAGE => 0 - Message (Email, SMS, Jabber, script etc)
ALERT_TYPE_COMMAND => 1 - Remote command

* 3.4 also has ALERT_STATUS_NEW
Sprinkle a bit of security

Auto-lock inactive user accounts

Sessions not available in the API -> use DB

• `sessions.status` – 0 active, 1 inactive

Housekeeper removes old sessions

• `config.hk_sessions_mode` – 0 disabled, 1 enabled
• `config.hk_sessions` – retention, in days
$ perl lock_users.pl
4 accounts past the threshold

<table>
<thead>
<tr>
<th>User</th>
<th>Last login</th>
</tr>
</thead>
<tbody>
<tr>
<td>mhare</td>
<td>No login info, first seen more than a week ago</td>
</tr>
<tr>
<td>hatter</td>
<td>No login info, first seen more than a week ago</td>
</tr>
<tr>
<td>khearts</td>
<td>No login info, not seen before</td>
</tr>
<tr>
<td>ccat</td>
<td>Thu Jan 5 10:23:50 2017</td>
</tr>
<tr>
<td>qhearts</td>
<td>Fri Jan 27 11:20:24 2017</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>dutchess</td>
<td>Mon Sep 11 20:38:34 2017</td>
</tr>
<tr>
<td>alice</td>
<td>Mon Sep 11 20:38:35 2017</td>
</tr>
</tbody>
</table>
Lessons reinforced

The more versions omitted, the more work

- Split the problem

Don't complicate things for a temporary need

- Make the 'old' method more involved
- Make the transition method more involved

Zabbix professional services for upgrade*

* bias – I like them
Let's get in touch

IRC – #zabbix on Freenode

Contact form –
http://www.zabbixbook.com/about/

Conference – ask me pointed questions

NEW: https://zabbix.org/wiki/Docs/DB_schema