ZABBIX 5.0 LTS
Freedom and integrity of monitoring
Alexei Vladishev
Founder and CEO of Zabbix
@avladishev
Quick recap of Zabbix 4.2 and 4.4
**Zabbix 4.2**

**April, 2019**

- High frequency monitoring with throttling
- Data collection: HTTP agent, Prometheus
- Preprocessing: validation and JavaScript!
- Preprocessing by Proxies
- Enhanced tag management
Zabbix 4.4
September, 2019

New Zabbix Agent: plugins, scheduler and more

Web hooks for alerting and notifications

Support of TimescaleDB

Built-in knowledge base for metrics and triggers

Standard for Zabbix Templates
5.0 LTS release
May 12, 2020
Available templates for monitoring & integrations

| 7 | https://www.zabbix.com/integrations | 7 |
Making a platform for high quality solutions
Monitoring

redis
MySQL
PostgreSQL
NGINX
ClickHouse
Windows
HAProxy
elasticsearch
Easy to contribute!

3 simple steps

Sign Zabbix Contributor Agreement (ZCA)
https://www.zabbix.com/developers

Make Zabbix Pull Request
https://git.zabbix.com

Zabbix Dev Team will review and accept if everything is fine
Available everywhere!

Linux distributions and containers

- RHEL and CentOS 6, 7 and 8
- Debian 8, 9, 10
- SuSE 12, 15
- 16.04 (Xenial), 18.04 (Bionic) and 20.04 (Focal Fossa)
- Raspbian 9 (Stretch), 10 (Buster)
- Docker

Linux appliance images

- ISO
- VMWare, VirtualBox
- Microsoft Hyper-V
- KVM
- XEN
- LiveCD

Public clouds

- AWS Zabbix Server 4.4 Mysql + Nginx
- Azure Zabbix Server 4.4 Mysql + Nginx
- DigitalOcean Zabbix Server 4.4 Mysql + Nginx
- Google Cloud Zabbix Server 4.4 Mysql + Nginx

OpenShift

OpenStack
Official support of Zabbix Agent2 for Linux and Windows
Most advanced monitoring agent on the market!

- Plugin infrastructure
- Support of long running scripts
- Parallel active checks
- Support of flexible intervals for all checks
- Support of persistent connections (DB connections)
- Accepting incoming traps and events (MQTT subscribe, listening TCP/UDP ports, etc)
- Monitoring of systemd services out of the box
- Drop-in replacement of the existing agent!
Persistent storage for Agent2

- Use cases
  - Unstable communications
  - Monitoring of critical data
  - Bursts of data

Your data is safe!

EnablePersistentBuffer=1
PersistentBufferFile=/var/spool/zabbix/agent.db
PersistentBufferPeriod=1d
Secure by design
Usability improvements
Build dashboards faster

Copy

Paste to the same or different dashboard
Export graphs as PNG image
Filter by tags for some widgets

Problem by severity & Problem hosts

Edit widget

Type: Problems by severity
Name: New York datacenters
Refresh interval: Default (1 minute)
Host groups: type here to search
Exclude host groups: type here to search
Hosts: type here to search
Problem: 
Severity: Not classified, Warning, High, Information
Tags: And/Or, Or
Service: Contains, Equals value
Datacenter: Contains, Equals NY
Add
Show: Host groups, Totals
Layout: Horizontal, Vertical
Show operational data: None, Separately, With problem name
Show suppressed problems: 
Hide groups without problems:

New York datacenters
0 Disaster 0 High 2 Average 43 Warning 1 Information 0 Not classified
Frankfurt datacenters
0 Disaster 2 High 0 Average 0 Warning 1 Information 0 Not classified

Production services
Host group: Disaster, High, Average, Warning, Information, Not classified
Cloud/AWS
Cloud/Azure
End user services
HPC Cluster
Internal infrastructure
R&D Lab2
Region/Australia
Region/China
Region/USA
UI modules

Create new menu entries
Create new pages
Extend existing functionality
Use and share 3rd party extensions
Permission control
Example:

Structure

manifest.json

```
{
   "manifest_version": 1.0,
   "id": "demo.report",
   "version": "1.0",
   "name": "Custom report",
   "namespace": "Demo",
   "author": "Zabbix",
   "url": "http://www.zabbix.com",
   "description": "Demo report module",
   "actions": {
      "demo.report": {
         "class": "DemoReportAction",
         "view": "demo.report"
      }
   }
}
```

module.php

```
<?php declare(strict_types = 1);
namespace Modules\Demo;
use APP;
use Core\CModule as BaseModule;

class Module extends BaseModule {
    public function init(): void {
        (APP::Component()->get('menu.main'))
          ->find(_('Reports'))
          ->add('Custom report', [
            'action' => 'demo.report'
          ]);    
    }
}
```
List of monitored devices

Monitoring->Hosts

<table>
<thead>
<tr>
<th>Name</th>
<th>Interface</th>
<th>Availability</th>
<th>Tags</th>
<th>Problems</th>
<th>Status</th>
<th>Latest data</th>
<th>Problems</th>
<th>Graphs</th>
<th>Screens</th>
<th>Web</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS N30</td>
<td>127.0.0.1:10050</td>
<td>ZBX, SNMP, JMX, IPMI</td>
<td>OS: Linux, Region: us-east-1, Service: Oracle Cluster</td>
<td>1</td>
<td>Enabled</td>
<td>Latest data</td>
<td>Problems 1</td>
<td>Graphs 14</td>
<td>Screens 2</td>
<td>Web</td>
</tr>
<tr>
<td>AWS N34</td>
<td>127.0.0.1:10050</td>
<td>ZBX, SNMP, JMX, IPMI</td>
<td>OS: Linux</td>
<td>1</td>
<td>Enabled</td>
<td>Latest data</td>
<td>Problems 2</td>
<td>Graphs 14</td>
<td>Screens 2</td>
<td>Web</td>
</tr>
<tr>
<td>AWS N90</td>
<td>127.0.0.1:10050</td>
<td>ZBX, SNMP, JMX, IPMI</td>
<td>OS: Linux</td>
<td>1</td>
<td>Enabled</td>
<td>Latest data</td>
<td>Problems 3</td>
<td>Graphs 14</td>
<td>Screens 2</td>
<td>Web</td>
</tr>
<tr>
<td>AZ M08</td>
<td>127.0.0.1:10050</td>
<td>ZBX, SNMP, JMX, IPMI</td>
<td>OS: Linux</td>
<td>1</td>
<td>Enabled</td>
<td>Latest data</td>
<td>Problems 2</td>
<td>Graphs 14</td>
<td>Screens 2</td>
<td>Web</td>
</tr>
<tr>
<td>AZ M10</td>
<td>127.0.0.1:10050</td>
<td>ZBX, SNMP, JMX, IPMI</td>
<td>OS: Linux</td>
<td>1</td>
<td>Enabled</td>
<td>Latest data</td>
<td>Problems 2</td>
<td>Graphs 14</td>
<td>Screens 2</td>
<td>Web</td>
</tr>
<tr>
<td>AZ M18</td>
<td>127.0.0.1:10050</td>
<td>ZBX, SNMP, JMX, IPMI</td>
<td>OS: Linux</td>
<td>1</td>
<td>Enabled</td>
<td>Latest data</td>
<td>Problems 2</td>
<td>Graphs 14</td>
<td>Screens 2</td>
<td>Web</td>
</tr>
<tr>
<td>Linux001</td>
<td>127.0.0.1:10050</td>
<td>ZBX, SNMP, JMX, IPMI</td>
<td>OS: Linux</td>
<td>1</td>
<td>Enabled</td>
<td>Latest data</td>
<td>Problems 2</td>
<td>Graphs 14</td>
<td>Screens 2</td>
<td>Web</td>
</tr>
</tbody>
</table>

No more Monitoring->WEB and Monitoring->Graphs
Easy navigation to host related resources
New preprocessing operator: **Replace**

**Configuration->Items**

- **Typical use cases**
  - mappings (text -> numeric, numeric -> text)
  - removing characters and strings
  - replacing characters and strings
  - in many cases easier than dealing with regular expressions!
New operator for JSONPath: ~

It returns property names of matching elements

```json
{
    "consul": [],
    "content": [
        "2020.01.05",
        "golang"
    ],
    "login": [
        "2019.11.02",
        "java"
    ],
    "mail": [
        "2020.01.02",
        "golang"
    ]
}
```

\$.*~

```json
[
    "consul",
    "content",
    "login",
    "mail"
]
```
Threading for email notifications

Grouped by event ID + media
Mass update of user macros for hosts & templates
Mass update of user macros for hosts & templates

<table>
<thead>
<tr>
<th>Host</th>
<th>Templates</th>
<th>IPMI</th>
<th>Tags</th>
<th>Macros</th>
<th>Inventory</th>
<th>Encryption</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Add</strong></td>
<td><strong>Update</strong></td>
<td><strong>Remove</strong></td>
</tr>
<tr>
<td>Macros</td>
<td>([$DATACENTER])</td>
<td></td>
<td>Remove</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add</td>
<td></td>
<td></td>
<td>Except selected</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Update</strong></td>
<td><strong>Cancel</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Host</th>
<th>Templates</th>
<th>IPMI</th>
<th>Tags</th>
<th>Macros</th>
<th>Inventory</th>
<th>Encryption</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Add</strong></td>
<td><strong>Update</strong></td>
<td><strong>Remove</strong></td>
</tr>
<tr>
<td>Macros</td>
<td></td>
<td></td>
<td><strong>Remove all</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>I confirm to remove all macros</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Update</strong></td>
<td><strong>Cancel</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
CLI tool to test JS scripts
Typical use cases

Test JavaScript code from command line:
- webhooks
- complex preprocessing scripts
shell> zabbix_js -help

Execute script using Zabbix embedded scripting engine.

General options:
- \texttt{-s,\texttt{--}script \textit{script-file}} Specify the filename of script to execute. Specify for standard input.
- \texttt{-i,\texttt{--}input \textit{input-file}} Specify input parameter file name. Specify for standard input.
- \texttt{-p,\texttt{--}param \textit{input-param}} Specify input parameter
- \texttt{-l,\texttt{--}loglevel \textit{log-level}} Specify log level
- \texttt{-t,\texttt{--}timeout \textit{timeout}} Specify timeout in seconds
- \texttt{-h --help} Display this help message
- \texttt{-V --version} Display version number
Example 1

```
shell> cat test.js

return Math.log(value)

shell> zabbix_js -s test.js -p 10

2.302585092994046
```

Example 2

```
shell> zabbix_js -s test.js -i my.json # reading input from file
```

Example 3

```
shell> cat test.js

Zabbix.Log(3, value) // use Zabbix.Log(log level, text) for debug purposes

return Math.log(value)
```
Triggers support text operations
Text data

Typical use cases

Working with software versions

Log file monitoring

Comparing string values of different items

Comparing last and previous values

Supported operators: $=<>$
Comparing with text constant

{host:zabbix.version.last()}="5.0.0"
{host:zabbix.version.last()}="{$ZABBIX.VERSION}"

Comparing last value with previous one

{host:text.last()}<>{host:text.prev()}

OR

{host:text.last(#1)}<>{host:text.last(#2)}

Comparing values of different items

{hostA:textA.last()}={hostB:textB.last()}
Automation &
Discovery
Discovery for JMX counters

New JMX checks

jmx.get[]

jmx.discovery[]

jmx.get[] is similar to the jmx.discovery[] item, but does not does not turn Java object properties into low-level discovery macro names and therefore can return values without limitations that are associated with LLD macro name generation such as hyphens or non-ASCII characters.

jmx.get[beans,"com.example:type=*,*"]

```
{
  "object": "com.example:type=Hello.data-src=data-base.,ключ=значение",
  "domain": "com.example",
  "properties": {
    "data-src": "data-base",
    "ключ": "значение",
    "type": "Hello"
  }
},
{
  "object": "com.example:type=Atomic",
  "domain": "com.example",
  "properties": {
    "type": "Atomic"
  }
}
```

jmx.discovery[...]

```
[
  {
    "{#JMXDOMAIN}":"java.lang",
    "{#JMXTYPE}":"GarbageCollector",
    "{#JMOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
    "{#JMXNAME}":"PS Scavenge"
  }
]
Discovery for Windows perf counters

Zabbix Agent and Agent2

perf_counter.discovery[object]

perf_counter_en.discovery[object]

[ "{"{#INSTANCE}":":0"},
  "{"{#INSTANCE}":":1"},
  "{"{#INSTANCE}":":_Total"}"
]
Discovery of IPMI sensors

Typical use cases

Simpler templates

```javascript
const ipmi = require('ipmi-client');

const ipmiGet = ipmi.get;

// Example of getting sensor data
ipmiGet({
  id: "SubTemp12",
  name: "(7.1).SubTemp12",
  sensor: {
    type: "1",
    text: "temperature"
  },
  reading: {
    type: "1",
    text: "threshold"
  },
  state: {
    state: "3",
    text: "lower Critical - going high"
  },
  value: 32,
  units: "C",
  threshold: {
    low: {
      "non_crit": 48,
      "crit": 32,
      "non_recover": 16
    },
    up: {
      "non_crit": 112,
      "crit": 144,
      "non_recover": 160
    }
  }
}).then(data => {
  console.log(data);
}).catch(err => {
  console.error(err);
});

// Example of getting another sensor data
ipmiGet({
  id: "1.8V Switch",
  name: "(7.1).1.8V Switch",
  sensor: {
    type: "2",
    text: "voltage"
  },
  reading: {
    type: "2",
    text: "discrete_usage"
  },
  state: {
    state: "1",
    text: "transition to active"
  }
}).then(data => {
  console.log(data);
}).catch(err => {
  console.error(err);
});
```
Support of **user macros** for host prototypes
User macros for host prototypes

Use LLD macros in macro value and description!
Support of **Float64** data types
Float64

Benefits

Compatible with Float64 returned by Prometheus

Execute to upgrade existing installation:

MySQL: database/mysql/double.sql
PostgreSQL: database/postgresql/double.sql
MySQL: database/oracle/double.sql
Scalability improvements
Improvements made

No drop-downs for host selections anymore, replaced with host search control

Hardcoded maximum size of the Overview grid

Redesigned Monitoring->Hosts->Graphs (multiselection of graphs, displaying of all graphs, pattern matching)

Introduced paging whenever possible (Monitoring ->Hosts->Web)

Zabbix UI

Ready to handle

Millions of devices
Compression for efficiency
Manage LLD rules globally
Filter for discovery rules

Useful for: troubleshooting (find all not supported or disabled), mass operations
Ability to unacknowledge event
Unacknowledge it!

Useful for

fixing mistakes

better workflow between various teams
Overrides for LLD rules
Discovery of filesystems

Special treating of Oracle related filesystems
Do not discover temporary filesystems

Override

Name: Do not monitor temporary filesystems

Stop processing next overrides if matches: 

Filters

Type of calculation: And/Or
Label Macro: A
macro: (#FSNAME)
Matches: tmp
Action: Remove

Operations

Condition: Trigger prototype equals
New operation

Object: Item prototype
Condition: equals
Pattern: pattern
Discover: Yes
Create enabled: 
Update interval: 
History storage period: 
Trend storage period: 
Original:
Add
Cancel
New macros

**ZBXNEXT-1797** support of macro `{HOST.ID}` in notifications

Can be used to build URLs to Zabbix UI. For example, Latest data:

```
{$ZABBIX.URL}/zabbix.php?action=latest.view&filter_set=1&filter_hostids%5B0%5D={HOST.ID}
```

**ZBXNEXT-5369** support of macro `{EVENT.TAGSJSON}` in notifications

Easier to pass all tags to webhooks

**ZBXNEXT-252** support of macro `{EVENT.DURATION}` in notifications.

Recovery subject “Resolved in 5m: Service Nginx is down.”
Other improvements

ZBXNEXT-5848  increased size of acknowledge messages to 4K (was 256)
ZBXNEXT-5690  added support of LIBSSH to support newer platforms like RHEL 8
ZBXNEXT-5825  support of ElasticSearch 7.x (7.4, 7.6)
ZBXNEXT-5720  Latest data displays data if filter is not set
ZBXNEXT-1561  increased zabbix_sender time resolution to nanoseconds
ZBXNEXT-1234  Monitoring->Latest data: show data if filter is empty
ZBXNEXT-5734  Base64 processing in JavaScript, functions atob() and btoa()
ZBXNEXT-5604  Do not log system.run[] for local use
ZBXNEXT-4584  New API method to get auditlog
ZBXNEXT-5851  Remote monitoring of versions of Zabbix components
And more

ZBXNEXT-1989 Increased size of item key to 2048 characters (was 255)
ZBXNEXT-3940 Ability to flush SNMP cache, SNMPv3 context changes
ZBXNEXT-5829 Faster hash function for internal operations
ZBXNEXT-2081 Documented how to do filtering for vmware.event monitoring
ZBX-15914 Improved consistency of map labels
Removing legacy to make a better product, faster

ZBXNEXT-5697  No support of Internet Explorer 11 anymore

ZBXNEXT-5592  Dropped support of IBM DB2 database

ZBXNEXT-5716  mbedTLS (former polarSSL) is no longer supported for encryption. Only OpenSSL and GnuTSL libraries

Minimum supported version for PHP is now 7.2: safer and more strict code
Upgrade!

Alexander will cover it in his talk today! :-}
Thank you!