



Highlights of Zabbix 2.2

ZABBIX 2013
Conference

General look and feel

The web interface has become
more responsive and intuitive

Server performance and data
processing speed has greatly
increased



Web monitoring

Improved web monitoring

You can now configure web scenarios for hosts and templates

| <u>Templates</u> ↓↑ | <u>Applications</u> | <u>Items</u> | <u>Triggers</u> | <u>Graphs</u> | <u>Screens</u> | <u>Discovery</u> | <u>Web</u> | <u>Lin</u> |
|--|----------------------------------|----------------------------|-------------------------------|-----------------------------|-----------------------------|-------------------------------|-------------------------|------------|
| Template App Agentless | Applications (1) | Items (12) | Triggers (12) | Graphs (0) | Screens (0) | Discovery (0) | Web (0) | - |
| Template App MySQL | Applications (1) | Items (14) | Triggers (1) | Graphs (2) | Screens (1) | Discovery (0) | Web (0) | - |
| Template App Zabbix Agent | Applications (1) | Items (3) | Triggers (3) | Graphs (0) | Screens (0) | Discovery (0) | Web (0) | - |
| Template App Zabbix Proxy | Applications (1) | Items (21) | Triggers (19) | Graphs (4) | Screens (1) | Discovery (0) | Web (0) | - |
| Template App Zabbix Server | Applications (1) | Items (29) | Triggers (25) | Graphs (5) | Screens (1) | Discovery (0) | Web (0) | - |
| Template IPMI Intel SR1530 | Applications (3) | Items (8) | Triggers (11) | Graphs (2) | Screens (0) | Discovery (0) | Web (0) | - |
| Template IPMI Intel SR1630 | Applications (3) | Items (11) | Triggers (21) | Graphs (2) | Screens (0) | Discovery (0) | Web (0) | - |
| Template JMX Generic | Applications (8) | Items (47) | Triggers (20) | Graphs (11) | Screens (0) | Discovery (0) | Web (0) | - |
| Template JMX Tomcat | Applications (5) | Items (32) | Triggers (5) | Graphs (4) | Screens (0) | Discovery (0) | Web (0) | - |

Template web scenarios 

New scenario settings

You can set the number of times to run each step and an HTTP proxy to use

The screenshot shows the 'Steps' configuration tab for a scenario named 'Homepage test'. The form includes the following fields:

- Name: Homepage test
- Application: (dropdown menu)
- New application: (text input field)
- Authentication: None (dropdown menu)
- Update interval (in sec): 60
- Retries: 3
- Agent: Internet Explorer 10.0 (dropdown menu)
- HTTP proxy: http://[username[:password]@]proxy.example.com[:port]
- Variables: (text area)

Handwritten annotations include:

- An arrow pointing from the word 'Retries' to the 'Retries' input field containing the value '3'.
- An arrow pointing from the text 'HTTP proxy' to the 'HTTP proxy' input field.

Regular expression matching

Variables can extract data from the response so it can be used in the following steps

Step of scenario

| | |
|-----------|--|
| Name | Welcome page |
| URL | http://company.com |
| Post | |
| Variables | {username}=Welcome, ([a-zA-Z]+) <i>Regexps in variables</i> |
| Timeout | 15 |



**Virtual machine
monitoring**

Automatic VM discovery

LLD can be used to automatically discover hypervisors and virtual machines

Discovery rules

Displaying 1 to 3 of 3 found

« [Template list](#) **Template:** [Template Virt VMware vCenter](#) [Applications \(2\)](#) [Items \(1\)](#) [Triggers \(0\)](#) [Graphs \(0\)](#) [Hosts \(0\)](#)

| <input type="checkbox"/> | Name ↑ | Items | Triggers | Graphs | Hosts | Key |
|--------------------------|--|-------------------------------------|--|--------------------------------------|-------------------------------------|---------------------|
| <input type="checkbox"/> | Discover vCenter clusters | Item prototypes (1) | Trigger prototypes (0) | Graph prototypes (0) | Host prototypes (0) | vmware: |
| <input type="checkbox"/> | Discover vCenter hypervisors | Item prototypes (0) | Trigger prototypes (0) | Graph prototypes (0) | Host prototypes (1) | vmware: |
| <input type="checkbox"/> | Discover vCenter VMs | Item prototypes (0) | Trigger prototypes (0) | Graph prototypes (0) | Host prototypes (1) | vmware: |

Enable selected ▾ Go (0)

New host prototype section



Built-in VMware support

You can use a number of built-in checks to monitor your VMware installations

`vmware.vcenter.vm.cpu.usage`

`vmware.vcenter.vm.cpu.num`

`vmware.vcenter.vm.memory.size.private`

`vmware.vcenter.vm.memory.size.ballooned`

`vmware.vcenter.vm.hv.name`

`vmware.vcenter.vm.uptime`

New VMware templates

Zabbix 2.2 comes with a set of templates to monitor VMware vCenters and vSpheres

Ready-to-use templates

| | | | | | | | |
|--------------------------|---|----------------------------------|----------------------------|------------------------------|----------------------------|-----------------------------|-------------------------------|
| <input type="checkbox"/> | Template Virt VMware vCenter | Applications (2) | Items (1) | Triggers (0) | Graphs (0) | Screens (0) | Discovery (3) |
| <input type="checkbox"/> | Template Virt VMware vCenter Guest | Applications (8) | Items (17) | Triggers (0) | Graphs (0) | Screens (0) | Discovery (3) |
| <input type="checkbox"/> | Template Virt VMware vCenter Hypervisor | Applications (4) | Items (17) | Triggers (0) | Graphs (0) | Screens (0) | Discovery (0) |
| <input type="checkbox"/> | Template Virt VMware vSphere | Applications (5) | Items (17) | Triggers (0) | Graphs (0) | Screens (0) | Discovery (1) |
| <input type="checkbox"/> | Template Virt VMware vSphere Guest | Applications (8) | Items (16) | Triggers (0) | Graphs (0) | Screens (0) | Discovery (3) |

Preconfigured LLD rules



Loadable modules

The idea of loadable modules

A faster and more powerful alternative to user parameters and external checks

Modules are libraries written in C used by Zabbix server and agents

Using modules

Modules are placed in the directory specified by the `LoadModulePath` parameter and enabled using the `LoadModule` parameter

```
LoadModulePath=/usr/local/lib/zabbix/agent/
```

```
LoadModule=mariadb.so
```

```
LoadModule=apache.so
```

```
LoadModule=kernel.so
```

```
LoadModule=dummy.so
```

Using modules

After restarting the server or agent you can create items that will use the new checks from the modules

Item

| Name | <input type="text" value="Dymmy item"/> | | | | | | | | |
|--------------------------------|--|---------------------------------------|--|----------|--------|--------|--------------------------------|--|--|
| Type | <input type="text" value="Zabbix agent"/> | | | | | | | | |
| Key | <input type="text" value="dummy.echo[Hello world]"/> | <input type="button" value="Select"/> | | | | | | | |
| Host interface | <input type="text" value="127.0.0.1 : 10050"/> | | | | | | | | |
| Type of information | <input type="text" value="Text"/> | | | | | | | | |
| Update interval (in sec) | <input type="text" value="30"/> | | | | | | | | |
| Flexible intervals | <table border="1"><thead><tr><th>Interval</th><th>Period</th><th>Action</th></tr></thead><tbody><tr><td colspan="3">No flexible intervals defined.</td></tr></tbody></table> | | | Interval | Period | Action | No flexible intervals defined. | | |
| Interval | Period | Action | | | | | | | |
| No flexible intervals defined. | | | | | | | | | |

← Custom item

Module structure

Modules are written in C and must implement the following functions:

- A function to run when a module is initialized and shut down
- A function to return the list of supported checks
- A separate function that implements a specific check

Sharing modules

Modules can and should be developed and shared by the community independently of the Zabbix code base



Internal events

What is an internal event?

Internal events are generated when an item, trigger or low-level discovery rule goes into a “Not supported” or “Unknown” state

Handling internal events

You can configure actions to notify you if an item or trigger stops working

Create action

New event source  Event source Internal

| Conditions | Operations | Status |
|---|--|-------------------------|
| Event type = <i>Item in "not supported" state</i> | Send message to user groups: Zabbix administrators via all media | Enabled |
| Event type = <i>Low-level discovery rule in "not supported" state</i> | Send message to user groups: Zabbix administrators via all media | Enabled |
| Event type = <i>Trigger in "unknown" state</i> | Send message to user groups: Zabbix administrators via all media | Enabled |



**Web interface
improvements**

Access to disabled hosts

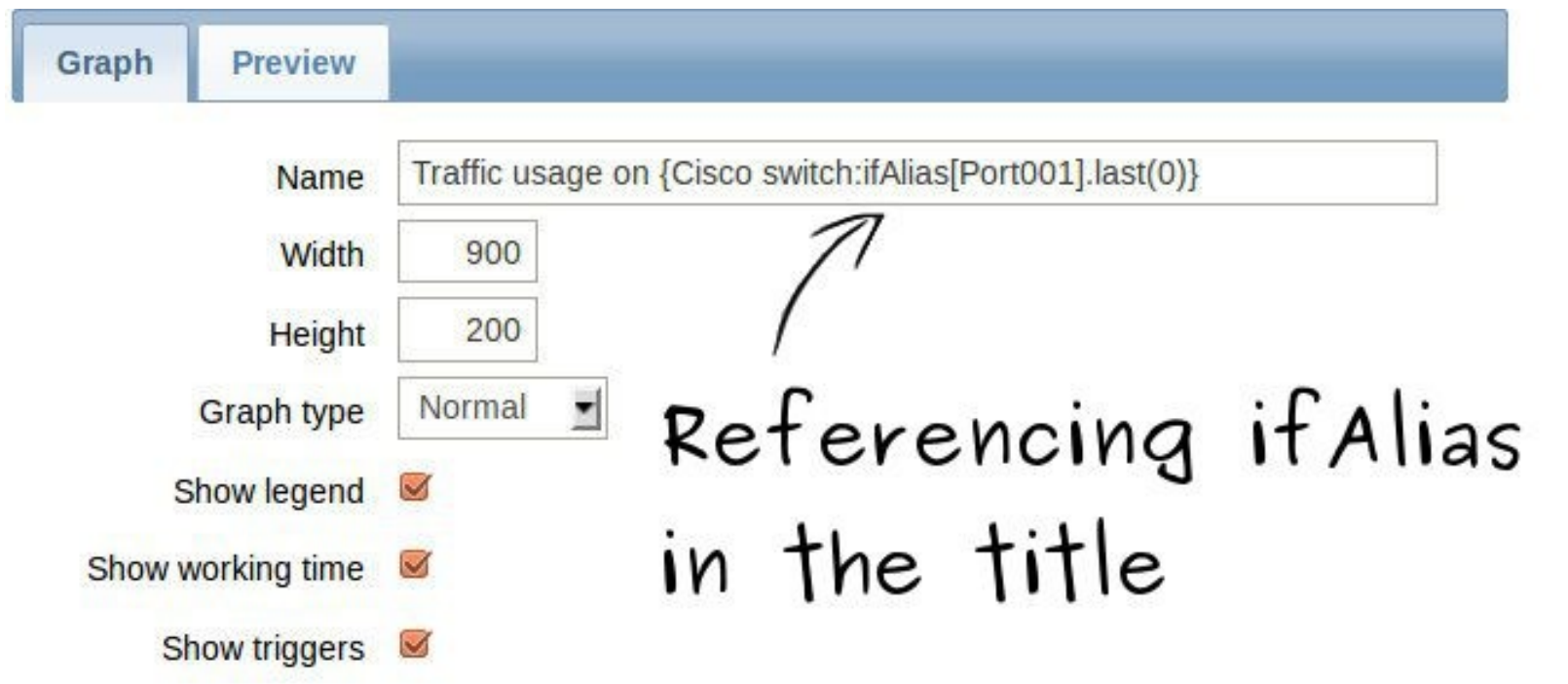
Users will now have access to historical data of disabled hosts

| + <u>Host</u> | <u>Name</u> ↑ |
|---------------------------------|--------------------------|
| + Zabbix server | CPU (13 Items) |
| + Zabbix server | General (5 Items) |
| + Zabbix server | Memory (5 Items) |
| + Zabbix server | OS (8 Items) |
| + Zabbix server | Performance (13 Items) |
| + Zabbix server | Processes (2 Items) |
| + Zabbix server | Security (2 Items) |
| + Zabbix server | Zabbix agent (3 Items) |
| + Zabbix server | Zabbix server (29 Items) |

Disabled host

Macros in graph names

Simple macros like `{host:key.func(param)}` are now supported in graph names



The screenshot shows the Zabbix configuration interface for a graph. The 'Preview' tab is selected. The 'Name' field contains the text 'Traffic usage on {Cisco switch:ifAlias[Port001].last(0)}'. The 'Width' is set to 900, 'Height' to 200, and 'Graph type' to 'Normal'. There are three checked checkboxes: 'Show legend', 'Show working time', and 'Show triggers'. A handwritten arrow points from the text 'Referencing ifAlias in the title' to the macro in the 'Name' field.

| | |
|-------------------|--|
| Name | Traffic usage on {Cisco switch:ifAlias[Port001].last(0)} |
| Width | 900 |
| Height | 200 |
| Graph type | Normal |
| Show legend | <input checked="" type="checkbox"/> |
| Show working time | <input checked="" type="checkbox"/> |
| Show triggers | <input checked="" type="checkbox"/> |

Referencing ifAlias
in the title

Improved value mappings

Value mappings can now be created for string and float values

Value mapping

Name


Mappings

| Value | | Mapped to | |
|--------------------------------|---|---|------------------------|
| <input type="text" value="F"/> | ⇒ | <input type="text" value="Full"/> | Remove |
| <input type="text" value="D"/> | ⇒ | <input type="text" value="Differential"/> | Remove |
| <input type="text" value="I"/> | ⇒ | <input type="text" value="Incremental"/> | Remove |
| Add | | | |

Text value mapping

Template applications

You can now link multiple templates with the same application

| | |
|--------------------------|--|
| <input type="checkbox"/> | <u>Application</u>  |
| <input type="checkbox"/> | <u>Template OS Linux</u> : CPU |
| <input type="checkbox"/> | <u>Template OS Linux</u> : Filesystems |
| <input type="checkbox"/> | <u>Template OS Linux</u> , <u>Template Web Server</u> : General |
| <input type="checkbox"/> | <u>Template OS Linux</u> , <u>Template Web Server</u> : Memory |
| <input type="checkbox"/> | <u>Template OS Linux</u> : Network interfaces |
| <input type="checkbox"/> | <u>Template OS Linux</u> : OS |
| <input type="checkbox"/> | <u>Template OS Linux</u> , <u>Template Web Server</u> : Performance |
| <input type="checkbox"/> | <u>Template OS Linux</u> , <u>Template Web Server</u> : Processes |

Applications from
two templates

Better housekeeper control

The housekeeper can now be configured from the web interface

Housekeeper

| | |
|---------------------------|--|
| Events and alerts | Enable housekeeping <input checked="" type="checkbox"/> |
| Flexible event settings → | Keep trigger data for (in days) <input type="text" value="365"/> |
| | Keep internal data for (in days) <input type="text" value="365"/> |
| | Keep network discovery data for (in days) <input type="text" value="365"/> |
| | Keep auto-registration data for (in days) <input type="text" value="365"/> |
| IT services | Enable housekeeping <input checked="" type="checkbox"/> |
| | Keep data for (in days) <input type="text" value="365"/> |
| Audit | Enable housekeeping <input checked="" type="checkbox"/> |
| | Keep data for (in days) <input type="text" value="365"/> |



Server improvements

Automatic database upgrades

The database will now be automatically upgraded when starting the Zabbix server



Better SNMPv3 support

SNMPv3 items now support context names, SHA authentication protocol and AES privacy protocol

| | |
|---------------------------|---|
| Type | <input type="text" value="SNMPv3 agent"/> |
| Key | <input type="text"/> <input type="button" value="Select"/> |
| SNMP OID | <input type="text" value="interfaces.ifTable.ifEntry.ifInOctets.1"/> |
| Context name | <input type="text"/> |
| Security name | <input type="text"/> |
| Security level | <input type="text" value="authPriv"/> |
| Authentication protocol | <input type="button" value="MD5"/> <input checked="" type="button" value="SHA"/> ← SHA authentication |
| Authentication passphrase | <input type="text"/> |
| Privacy protocol | <input type="button" value="DES"/> <input checked="" type="button" value="AES"/> ← AES privacy |
| Privacy passphrase | <input type="text"/> |

Zabbix proxy monitoring

Zabbix 2.2 comes with a set of new checks for monitoring the health of a Zabbix proxy. Some of them are:

```
zabbix[proxy_history]
```

```
zabbix[boottime]
```

```
zabbix[host, <type>, available]
```

```
zabbix[hosts]
```

```
zabbix[items]
```

```
zabbix[items_unsupported]
```



Follow the latest news on
Zabbix 2.2 at zabbix.com