

The ZABBIX logo consists of the word "ZABBIX" in a bold, white, sans-serif font, centered within a solid red rectangular background. The background of the entire slide is a dark blue gradient with a faint, glowing network of white lines and dots, and a subtle world map outline in the background.

ZABBIX

Reacting to Zabbix Events with Event Driven Ansible

Aleksandr Kotsegubov

Integration engineer

What are we going to discuss today?

How it works?

Rulebooks

How to configure?

Actions and conditions

Examples

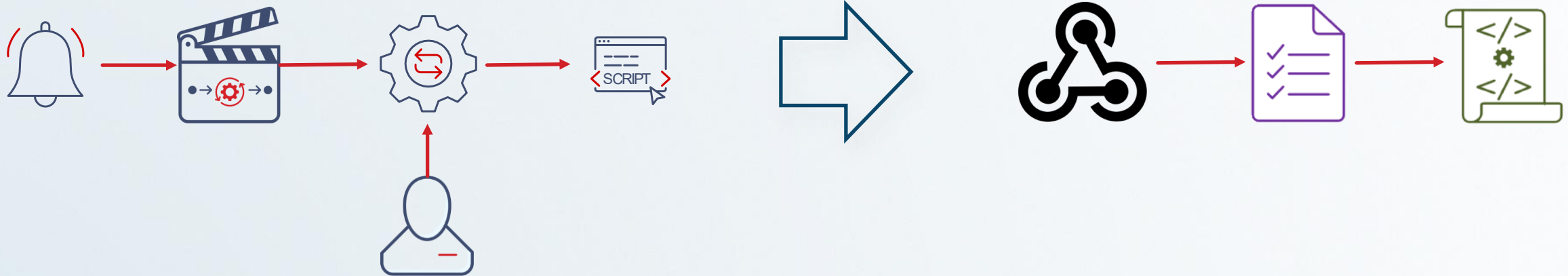
Why is this integration needed?

- ⚡ Automate your IT processes
- ⚡ React to non-scheduled events
- ⚡ Managing your IT infrastructure



How it works?

ZABBIX



How to configure in Zabbix?

- ⚡ Import the media type
 - ✓ Zabbix GIT
 - ✓ Integrations page
- ⚡ Create a service user
 - ✓ Select type “Event-Driven Ansible”
 - ✓ Specify address of EDA Server
- ⚡ Configure an action to send notifications
 - ✓ Specify sending via “Event-Drive Ansible.”
 - ✓ Specify the service user

The screenshot shows the 'Media' configuration window in Zabbix. The 'Type' is set to 'Event-Driven Ansible'. The 'Send to' field contains '192.168.0.100:5001' and the 'When active' field contains '1-7,00:00-24:00'. Under 'Use if severity', all checkboxes are checked: Not classified, Information, Warning, Average, High, and Disaster. The 'Enabled' checkbox is also checked. 'Update' and 'Cancel' buttons are at the bottom right.

Field	Value
Type	Event-Driven Ansible
* Send to	192.168.0.100:5001
* When active	1-7,00:00-24:00
Use if severity	<input checked="" type="checkbox"/> Not classified <input checked="" type="checkbox"/> Information <input checked="" type="checkbox"/> Warning <input checked="" type="checkbox"/> Average <input checked="" type="checkbox"/> High <input checked="" type="checkbox"/> Disaster
Enabled	<input checked="" type="checkbox"/>

<https://www.zabbix.com/integrations/ansible>

How to configure in Ansible?

⚡ Install

- ✓ Eda-server
- ✓ Ansible-rulebook
- ✓ Ansible collections “ansible.eda”
ansible-galaxy collection install ansible.eda

⚡ Create a rulebook

⚡ Run “ansible-rulebook”

- ✓ ansible-rulebook my-first-rulebook.yaml

<https://github.com/ansible/eda-server/blob/main/docs/deployment.md>

<https://ansible.readthedocs.io/projects/rulebook/en/stable/installation.html>

```
1 ---
2 - name: My first rulebook
3   hosts: all
4   sources:
5     - ansible.eda.webhook:
6       host: 0.0.0.0
7       port: 5001
8   rules:
9     - name: debug
10      condition: event.payload is defined
11      action:
12        debug:
13          msg: It works!
14
```

Ports must match!



Make sure you are using the same port as specified in user settings.

Media ✕

Type

* Send to

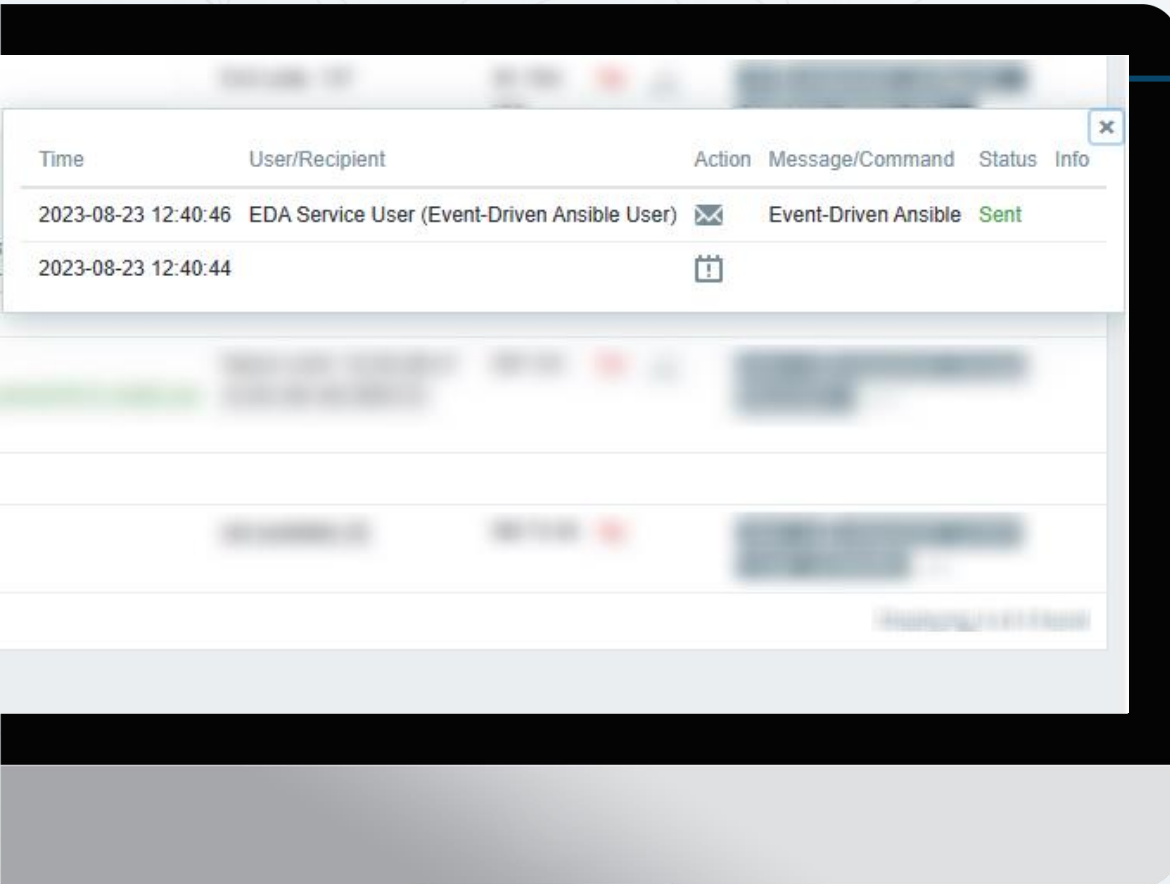
* When active

Use if severity Not classified
 Information
 Warning
 Average
 High
 Disaster

Enabled

```
1 ---
2 - name: My first rulebook
3   hosts: all
4   sources:
5     - ansible.eda.webhook:
6       host: 0.0.0.0
7       port: 5001
8   rules:
9     - name: debug
10      condition: event.payload is defined
11      action:
12        debug:
13          msg: It works!
```

Event has status "Sent"



If the event is marked as sent, it doesn't mean that rulebook has taken action!

Only problem events

The screenshot shows the Zabbix 'Problems' page. The left sidebar contains navigation options: Monitoring, Dashboard, Problems, Hosts, Latest data, Maps, Discovery, Services, Inventory, Reports, Configuration, and Administration. The main content area includes filters for 'Recent problems', 'Problems', and 'History'. There are input fields for 'Host groups', 'Hosts', and 'Triggers'. A 'Host inventory' section has a 'Type' dropdown and an 'Add' button. A 'Tags' section has an 'Add/Or' button and a 'tag' input field with a 'Contains' dropdown and a 'value' input field. Below these are 'Show tags' (None, 1, 2, 3), 'Tag name' (Full, Shortened, None), and 'Tag display priority' (comma-separated list). There are also checkboxes for 'Show operational data', 'Show suppressed problems', 'Show unacknowledged only', 'Compact view', 'Show details', 'Show timeline', and 'Highlight whole row'. At the bottom of the filters are 'Save as', 'Apply', and 'Reset' buttons. The main table lists problem events with columns for Time, Severity, Recovery time, Status, Info, Host, Problem, Duration, Ack, Actions, and Tags. The first row is highlighted in orange and shows a 'PROBLEM' status for 'Zabbix agent is not available (for 3m)'. Other rows show 'RESOLVED' status for various utilization issues.

Time	Severity	Recovery time	Status	Info	Host	Problem	Duration	Ack	Actions	Tags
2023-08-18 10:21:26	Average		PROBLEM		Zabbix server	Zabbix agent is not available (for 3m)	5d 6h 54m	No		class: os component: system scope: availa
2023-08-10 08:13:55	Average	2023-08-10 08:14:55	RESOLVED		Zabbix server	Zabbix server: Utilization of trigger housekeeper processes over 75%	1m	No		class: software component: internal-pr
2023-08-01 15:06:26	Average	2023-08-01 15:07:26	RESOLVED		Zabbix server	Zabbix server: Utilization of availability manager processes over 75%	1m	No		class: software component: internal-pr
August										
2023-07-27 08:00:55	Average	2023-07-27 08:01:55	RESOLVED		Zabbix server	Zabbix server: Utilization of trigger housekeeper processes over 75%	1m	No		class: software component: internal-pr
2023-07-14 17:18:48	Average	2023-07-14 17:19:48	RESOLVED		Zabbix server	Zabbix server: Utilization of timer processes over 75%	1m	No		class: software component: internal-pr
2023-07-12 10:56:26	Average	2023-07-12 10:57:26	RESOLVED		Zabbix server	Zabbix server: Utilization of availability manager processes over 75%	1m	No		class: software component: internal-pr
2023-07-11 08:42:48	Average	2023-07-11 08:43:48	RESOLVED		Zabbix server	Zabbix server: Utilization of timer processes over 75%	1m	No		class: software component: internal-pr
July										
2023-06-12 08:18:50	Average	2023-06-12 08:19:50	RESOLVED		Zabbix server	Zabbix server: Utilization of unreachable poller processes over 75%	1m	No		class: software component: data-colle
June										
2023-04-12 14:27:26	Average	2023-08-16 10:18:26	RESOLVED		Zabbix server	Zabbix agent is not available (for 3m)	4M 7d 19h	No		class: os component: system scope: availa



Only trigger-based problem events will be sent

What data will be sent?



- acknowledged
- endpoint
- event_date
- event_datetime_timestamp
- event_id
- event_name
- event_nseverity
- event_object
- event_severity
- event_source
- event_tags
- event_time
- event_value
- host_groups
- host_host
- host_id
- host_ip
- host_port
- monitoring_source
- operation_data
- send_to
- Subject
- trigger_description
- trigger_id
- trigger_name

Most interesting fields



Tags

- 📶 Location: Berlin
- 📶 Location: Central office
- 📶 container: /web_portal
- 📶 scope: security
- 📶 component: configuration



Host groups

- 📶 Berlin
- 📶 Linux servers
- 📶 Web servers

```
1  {
2      ...
3      "event_tags": {
4          "location": ["Berlin", "Central office"],
5          "component": ["configuration"],
6          "container": ["/web-portal"],
7          "scope": ["security"]
8      }
9      ...
10 }
11
```

```
1  {
2      ...
3      "host_groups": [
4          "Berlin",
5          "Linux servers",
6          "Web servers"
7      ]
8      ...
9  }
10
```

Most interesting fields

- 📡 event_nseverity
- 📡 event_severity
- 📡 acknowledged
- 📡 event_name
- 📡 trigger_name

```
1  {
2    ...
3    "event_nseverity": 4,
4    "event_severity": "High",
5    "acknowledged": "No",
6    "event_name": "Container '/web-portal': Container has been
7    stopped with error code",
8    "trigger_name": "Container '/web-portal: Container has
9    been stopped with error code"
10 }
```

How to view value?

print_event

```

1  ---
2  - name: Print all variables
3    hosts: all
4    sources:
5      - ansible.eda.webhook:
6        host: 0.0.0.0
7        port: 5001
8    rules:
9      - name: print event
10     condition: event.payload is defined
11     action:
12       print_event:
13         pretty: true
14

```

debug

```

1  ---
2  - name: Print all variables
3    hosts: all
4    sources:
5      - ansible.eda.webhook:
6        host: 0.0.0.0
7        port: 5001
8    rules:
9      - name: print event
10     condition: event.payload is defined
11     action:
12       debug:
13

```

```

1  {
2    "payload": {
3      "HTTPProxy": "",
4      "acknowledged": "No",
5      "endpoint": "/endpoint",
6      "event_date": "2023.08.23",
7      "event_datetime_timestamp": 1692817297,
8      "event_id": 448,
9      "event_name": "Container '/web-portal': Container has been stopped
10     with error code",
11     "event_nseverity": 4,
12     "event_object": 0,
13     "event_severity": "High",
14     "event_source": 0,
15     "event_tags": {"Location": ["Berlin", "Central office"],
16                   "component": ["configuration"],
17                   "container": ["/web-portal"],
18                   "scope": ["security"]},
19     "event_time": "22:01:37",
20     "event_value": 1,
21     "host_groups": ["Berlin", "Linux servers", "Web servers"],
22     "host_host": "Web portal",
23     "host_id": 10559,
24     "host_ip": "127.0.0.1",
25     "host_port": "10050",
26     "monitoring_source": "Zabbix sever",
27     "operation_data": "Exit code: 137",
28     "send_to": "192.168.56.102:5001",
29     "subject": "Event ID: 448, Host: Web portal, Problem: Container '/
30     web-portal': Container has been stopped with error code",
31     "trigger_description": "",
32     "trigger_id": 23133,
33     "trigger_name": "Container '/web-portal': Container has been
34     stopped with error code"
35   }
36 }

```

Rulebooks

Similar to Ansible Playbooks, but more oriented to “if-then” scenarios

📶 Sources

📶 Rules

✓ condition

✓ action

```
1 ---
2 - name: Print all variables
3   hosts: all
4   sources:
5     - ansible.eda.webhook:
6       host: 0.0.0.0
7       port: 5001
8   rules:
9     - name: print event
10      condition: event.payload is defined
11      action:
12        debug:
13
```



The order of the rules is important for the rulebook

Rule conditions

- ⚡ in
- ⚡ contains
- ⚡ is defined
- ⚡ is search(pattern ,ignorecase=true)
- ⚡ is regex(pattern, ignorecase=true)
- ⚡ is select(operator, value)
- ⚡ and / or

```
1 ---
2 - name: Debug condition
3   hosts: all
4   sources:
5     - ansible.eda.webhook:
6       host: 0.0.0.0
7       port: 5001
8   rules:
9     - name: Debug condition
10      condition: >-
11        event.payload.host_groups is select("=", "Web servers") and
12        event.payload.host_groups is select("=", "Berlin") and
13        event.payload.event_tags.component is select("=", "configuration")
14      action:
15        debug:
16          msg: 'It works!'
```

<https://ansible.readthedocs.io/projects/rulebook/en/stable/conditions.html>

Examples of conditions

```
1  {
2    ...
3    "event_tags": {
4      "Location": ["Berlin", "Central office"],
5      "component": ["configuration"],
6      "container": ["/web-portal"],
7      "scope": ["security"]
8    }
9    ...
10 }
11
```

```
1  ---
2  - name: Debug condition
3    hosts: all
4    sources:
5      - ansible.eda.webhook:
6        host: 0.0.0.0
7        port: 5001
8    rules:
9      - name: Debug condition
10     condition: event.payload.event_tags.component is select("=", "configuration")
11     action:
12       debug:
13         msg: 'It works!'
```

event.payload

- ⚡ event.payload.event_tags.component **is select**("=", "configuration")
- ⚡ event.payload.event_tags.Location **contains** "Central office"
- ⚡ event.payload.event_tags.scope **is defined**
- ⚡ event.payload **is defined**

Examples of conditions

```
1  {
2    ...
3    "host_groups": [
4      "Berlin",
5      "Linux servers",
6      "Web servers"
7    ]
8    ...
9  }
10
```

```
1  ---
2  - name: Debug condition
3    hosts: all
4    sources:
5      - ansible.eda.webhook:
6          host: 0.0.0.0
7          port: 5001
8    rules:
9      - name: Debug condition
10     condition: event.payload.host_groups is select("=", "Web servers")
11     action:
12         debug:
13             msg: 'It works!'
```

- ⚡ event.payload.host_groups is select("=", "Web servers")
- ⚡ event.payload.host_groups contains "Web servers"

Examples of conditions

```
1  {
2    ...
3    "event_nseverity": 4,
4    "event_severity": "High",
5    "acknowledged": "No",
6    "event_name": "Container '/web-portal': Container has been
7    stopped with error code",
8    "trigger_name": "Container '/web-portal: Container has
9    been stopped with error code"
10  }
```

```
1  ---
2  - name: Debug condition
3    hosts: all
4    sources:
5      - ansible.eda.webhook:
6        host: 0.0.0.0
7        port: 5001
8    rules:
9      - name: Debug condition
10     condition: >-
11       event.payload.event_nseverity > 3 and
12       event.payload.acknowledged == 'No' and
13       event.payload.event_name is regex(".*Container has been stopped with error code")
14     action:
15       debug:
16         msg: 'It works!'
```

⚡ event.payload.event_nseverity > 3

⚡ event.payload.acknowledged == 'No'

⚡ event.payload.event_name is regex(".*Container has been stopped with error code")

Rules actions

- run_playbook
- run_module
- run_job_template
- run_workflow_template
- set_fact
- post_event
- retract_fact
- print_event
- shutdown
- debug
- none

```
1 ---
2 - name: Debug condition
3   hosts: all
4   sources:
5     - ansible.eda.webhook:
6       host: 0.0.0.0
7       port: 5001
8   rules:
9     - name: Debug condition
10      condition: >-
11        event.payload.event_nseverity > 3 and
12        event.payload.acknowledged == 'No'
13      action:
14        run_playbook:
15          name: /root/ansible/playbooks/playbook_zabbix_example.yml
```

<https://ansible.readthedocs.io/projects/rulebook/en/stable/actions.html>

Use data from event

```
1 ---
2 - name: Zabbix -- Print the value
3   hosts: localhost
4   gather_facts: false
5   tasks:
6
7   - name: Print container name
8     ansible.builtin.debug:
9       msg: "{{ ansible_edata.event.payload.event_tags.container }}"
```

"{{ ansible_edata.event.payload }}"

Complex example



Tags

- target: nginx
- class: software
- component: configuration



Host groups

- Berlin
- Linux servers
- Web servers

```

1 ---
2 - name: Debug condition
3   hosts: all
4   sources:
5     - ansible.eda.webhook:
6       host: 0.0.0.0
7       port: 5001
8   rules:
9     - name: Debug condition
10      condition: >-
11        event.payload.host_groups is select("==","Web servers") and
12        event.payload.host_groups is select("==","Berlin") and
13        event.payload.event_tags.component is select("==","configuration")
14      action:
15        run_playbook:
16          name: /root/ansible/playbooks/reload_nginx.yml
  
```

```

1 {
2   ...
3   "event_tags": {
4     "target": ["nginx"],
5     "class": ["software"],
6     "component": ["configuration"]
7   },
8   "host_groups": ["Berlin", "Linux servers", "Web servers"]
9   ...
10 }
  
```

The ZABBIX logo consists of the word "ZABBIX" in a bold, white, sans-serif font, centered within a solid red rectangular background. The background of the entire slide is a dark blue gradient with a faint, glowing network of white lines and dots, and a semi-transparent world map in the center-right.

ZABBIX

Thank you!

Aleksandr Kotsegubov

Integration engineer