ZABBIX

Reacting to Zabbix Events with Event Driven Ansible

Aleksandr Kotsegubov

Integration engineer

What are we going to discuss today?

ZABBIX

How it works?

Rulebooks

How to configure?

Actions and conditions

Examples

Why is this integration needed?

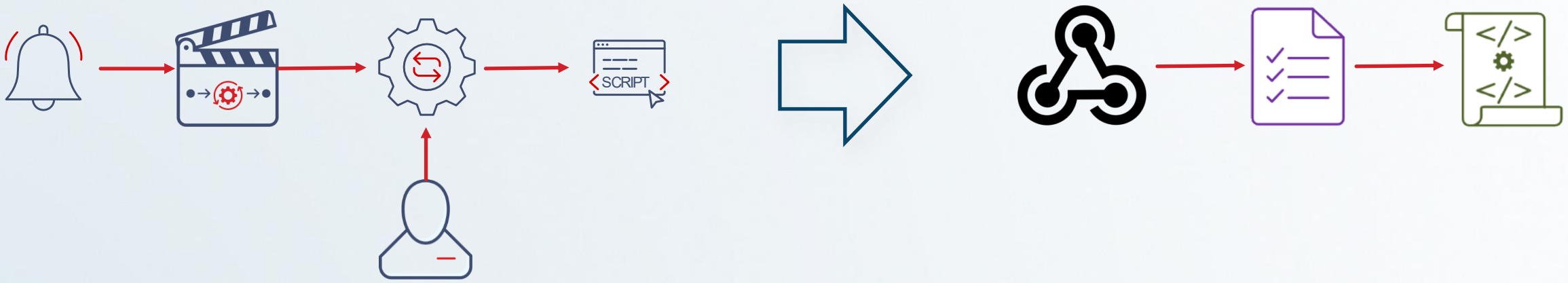
- Automate your IT processes
- React to non-scheduled events
- Managing your IT infrastructure

How it works?

ZABBIX

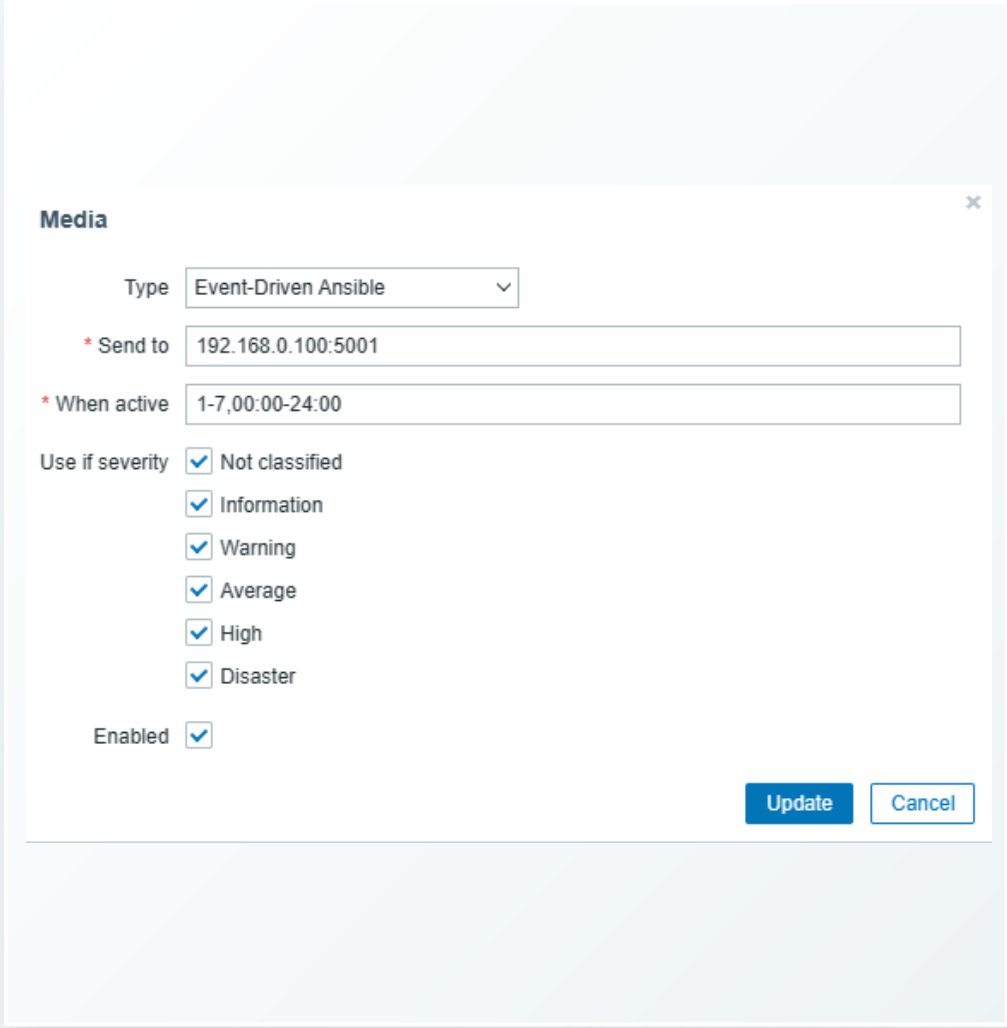
ZABBIX

ANSIBLE



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



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

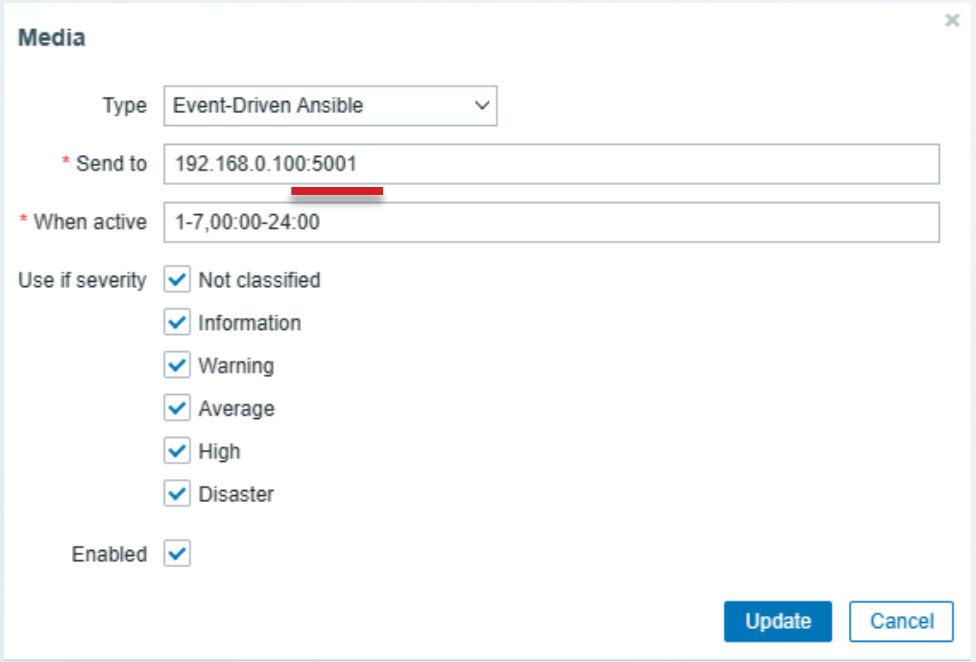
```yaml
---
- name: My first rulebook
  hosts: all
  sources:
    - ansible.eda.webhook:
        host: 0.0.0.0
        port: 5001
  rules:
    - name: debug
      condition: event.payload is defined
      action:
        debug:
          msg: It works!
```

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

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

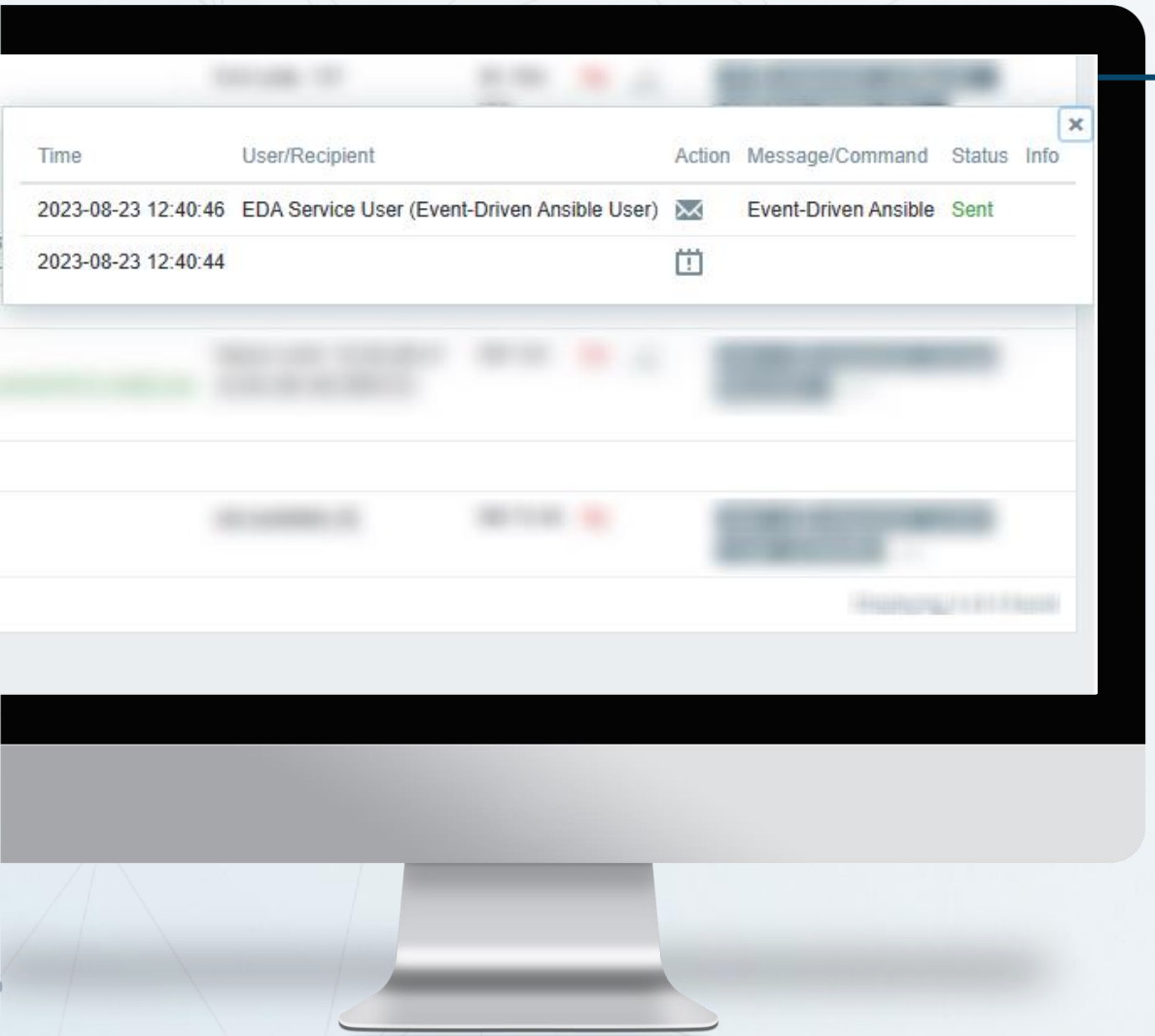
Ports must match!

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



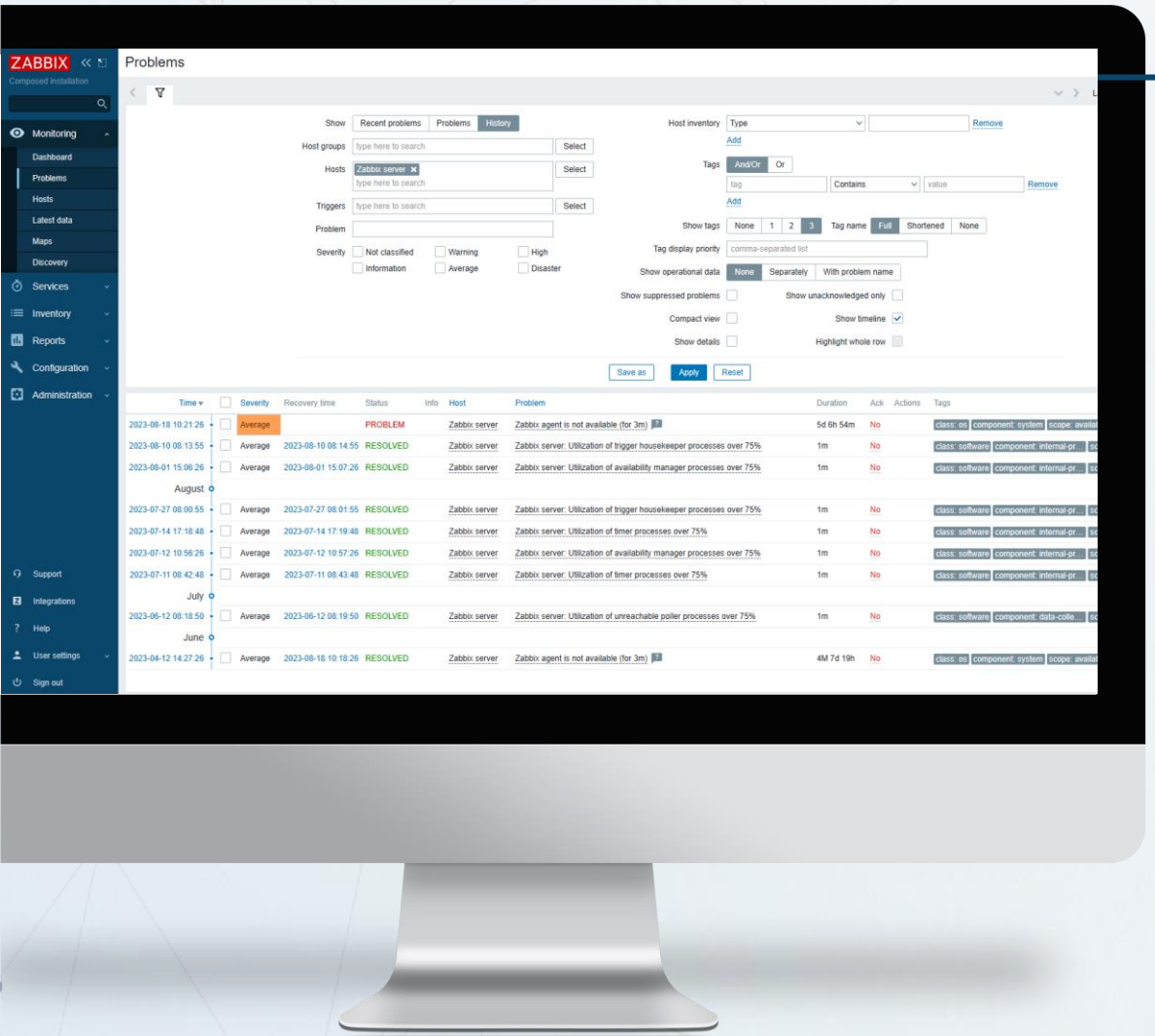

```yaml
---
- name: My first rulebook
  hosts: all
  sources:
    - ansible.eda.webhook:
        host: 0.0.0.0
        port: 5001
  rules:
    - name: debug
      condition: event.payload is defined
      action:
        debug:
          msg: It works!
```

Event has status “Sent”



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

Only problem events



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

```
{
    ...
    "event_tags": {
        "Location": ["Berlin", "Central office"],
        "component": ["configuration"],
        "container": ["/web-portal"],
        "scope": ["security"]
    }
    ...
}
```

Host groups

- Berlin
- Linux servers
- Web servers

```
{
    ...
    "host_groups": [
        "Berlin",
        "Linux servers",
        "Web servers"
    ]
    ...
}
```

Most interesting fields

- event_nseverity
- event_severity
- acknowledged
- event_name
- trigger_name

```
{
    ...
    "event_nseverity": 4,
    "event_severity": "High",
    "acknowledged": "No",
    "event_name": "Container '/web-portal': Container has been stopped with error code",
    "trigger_name": "Container '/web-portal: Container has been stopped with error code"
    ...
}
```

How to view value?

ZABBIX

- print_event

```yaml
---
- name: Print all variables
  hosts: all
  sources:
    - ansible.eda.webhook:
        host: 0.0.0.0
        port: 5001
  rules:
    - name: print event
      condition: event.payload is defined
      action:
        print_event:
          pretty: true
```

- debug

```yaml
---
- name: Print all variables
  hosts: all
  sources:
    - ansible.eda.webhook:
        host: 0.0.0.0
        port: 5001
  rules:
    - name: print event
      condition: event.payload is defined
      action:
        debug:
```


```json
{
    "payload": {
        "HTTPProxy": "",
        "acknowledged": "No",
        "endpoint": "/endpoint",
        "event_date": "2023.08.23",
        "event_datetime_timestamp": 1692817297,
        "event_id": 448,
        "event_name": "Container '/web-portal': Container has been stopped with error code",
        "event_nseverity": 4,
        "event_object": 0,
        "event_severity": "High",
        "event_source": 0,
        "event_tags": {"Location": ["Berlin", "Central office"],
                       "component": ["configuration"],
                       "container": ["/web-portal"],
                       "scope": ["security"]},
        "event_time": "22:01:37",
        "event_value": 1,
        "host_groups": ["Berlin", "Linux servers", "Web servers"],
        "host_host": "Web portal",
        "host_id": 10559,
        "host_ip": "127.0.0.1",
        "host_port": "10050",
        "monitoring_source": "Zabbix sever",
        "operation_data": "Exit code: 137",
        "send_to": "192.168.56.102:5001",
        "subject": "Event ID: 448, Host: Web portal, Problem: Container '/web-portal': Container has been stopped with error  code",
        "trigger_description": "",
        "trigger_id": 23133,
        "trigger_name": "Container '/web-portal': Container has been stopped with error code"
    }
}
```

Rulebooks

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

- Sources
- Rules
  - ✓ condition
  - ✓ action

The order of the rules is important for the rulebook

```yaml
---
- name: Print all variables
  hosts: all
  sources:
    - ansible.eda.webhook:
        host: 0.0.0.0
        port: 5001
  rules:
    - name: print event
      condition: event.payload is defined
      action:
        debug:
```

Rule conditions

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

```yaml
---
- name: Debug condition
  hosts: all
  sources:
    - ansible.eda.webhook:
        host: 0.0.0.0
        port: 5001
  rules:
    - name: Debug condition
      condition: >-
        event.payload.host_groups is select("==","Web servers") and
        event.payload.host_groups is select("==","Berlin") and
        event.payload.event_tags.component is select("==","configuration")
      action:
        debug:
          msg: 'It works!'
```

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

Examples of conditions

```
{
    ...
    "event_tags": {
        "Location": ["Berlin", "Central office"],
        "component": ["configuration"],
        "container": ["/web-portal"],
        "scope": ["security"]
    }
    ...
}
```

```
---
- name: Debug condition
  hosts: all
  sources:
    - ansible.eda.webhook:
        host: 0.0.0.0
        port: 5001
  rules:
    - name: Debug condition
      condition: event.payload.event_tags.component is select("==","configuration")
      action:
        debug:
          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

```
{
    ...
    "host_groups": [
        "Berlin",
        "Linux servers",
        "Web servers"
    ]
    ...
}
```

```
---
- name: Debug condition
  hosts: all
  sources:
    - ansible.eda.webhook:
        host: 0.0.0.0
        port: 5001
  rules:
    - name: Debug condition
      condition: event.payload.host_groups is select("==","Web servers")
      action:
        debug:
          msg: 'It works!'
```

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

Examples of conditions

```
{
    ...
    "event_nseverity": 4,
    "event_severity": "High",
    "acknowledged": "No",
    "event_name": "Container '/web-portal': Container has been stopped with error code",
    "trigger_name": "Container '/web-portal: Container has been stopped with error code"
    ...
}
```

```
---
- name: Debug condition
  hosts: all
  sources:
    - ansible.eda.webhook:
        host: 0.0.0.0
        port: 5001
  rules:
    - name: Debug condition
      condition: >-
        event.payload.event_nseverity > 3 and
        event.payload.acknowledged == 'No' and
        event.payload.event_name is regex(".*Container has been stopped with error code")
      action:
        debug:
          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

```yaml
---
- name: Debug condition
  hosts: all
  sources:
    - ansible.eda.webhook:
        host: 0.0.0.0
        port: 5001
  rules:
    - name: Debug condition
      condition: >-
        event.payload.event_nseverity > 3 and
        event.payload.acknowledged == 'No'
      action:
        run_playbook:
          name: /root/ansible/playbooks/playbook_zabbix_example.yml
```

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

Use data from event

```yaml
---
- name: Zabbix -- Print the value
  hosts: localhost
  gather_facts: false
  tasks:

  - name: Print container name
    ansible.builtin.debug:
      msg: "{{ ansible_eda.event.payload.event_tags.container }}"
```

"{{ ansible_eda.event.payload }}"

Complex example

Tags

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

Host groups

- Berlin
- Linux servers
- Web servers

```yaml
---
- name: Debug condition
  hosts: all
  sources:
    - ansible.eda.webhook:
        host: 0.0.0.0
        port: 5001
  rules:
    - name: Debug condition
      condition: >-
        event.payload.host_groups is select("==","Web servers") and
        event.payload.host_groups is select("==","Berlin") and
        event.payload.event_tags.component is select("==","configuration")
      action:
        run_playbook:
          name: /root/ansible/playbooks/reload_nginx.yml
```

```json
{
    ...
    "event_tags": {
        "target": ["nginx"],
        "class": ["software"],
        "component": ["configuration"]
    },
    "host_groups": ["Berlin", "Linux servers", "Web servers"]
    ...
}
```

ZABBIX

Thank you!

Aleksandr Kotsegubov

Integration engineer