

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 overlaid on a subtle world map.

**ZABBIX**

# Zabbix Ansible collection

Certified by RedHat

---

**Andrey Tochko**

Integration engineer

# Collection objectives

## Automate monitoring agent maintenance:

- ▶ Deploy / Remove
- ▶ Reconfigure / Extend
- ▶ Restart / Reload
- ▶ Upgrade / Downgrade

## Automatically represent devices in Zabbix:

- ▶ Add / Remove / Modify host
  - Setup interfaces
  - Assign templates, proxy, hostgroups
- ▶ Add / Remove hostgroups

# Certified collection

- ✓ Available from Automation platform hub
- ✓ Support from Redhat and collection maintainer
- ✓ Stronger quality and security requirements
- ✓ Built on top of certified content only

<https://access.redhat.com/support/articles/ansible-automation-platform-certified-content>



main

ansible-collection / README.md

↑ Top

Preview

Code

Blame

58 lines (35 loc) · 3.56 KB

Raw



## Collection components

### Roles:

- [zabbix.zabbix.agent](#) - the role to deploy, configure and maintain Zabbix agent on a target device.
- [zabbix.zabbix.host](#) - this role represents target device on Zabbix server.

### Plugins:

- [HTTP API](#) - Zabbix API interface for Ansible.
- [Inventory](#) - the tool to synchronize Zabbix monitoring instance hosts with Ansible inventory.

### Modules:

- [zabbix\\_hostgroup](#) - Ansible module for Zabbix host groups management (uses [HTTP API](#) plugin).
- [zabbix\\_host](#) - Ansible module for Zabbix hosts management (uses [HTTP API](#) plugin).

### Rulebooks:

- [zabbix.zabbix.example](#) - Ansible rulebook example for remediation of issues, detected by Zabbix.

```
$ ansible-playbook -i inventory/ playbook.yml
```

Static

### inventory/hosts.yml

```
aws:
  hosts:
    host[01:08].ubuntu.aws.lids:
  vars:
    ansible_host: '{{ inventory_hostname }}'
    ansible_ssh_private_key_file: ~/.ssh/aws.pem

oci:
  hosts:
    host101.ubuntu.oci.lids:
      ansible_host: 158.179.206.1
    host301.alma.oci.lids:
      ansible_host: 129.151.195.145
      ansible_user: opc
  vars:
    ansible_ssh_private_key_file: ~/.ssh/id_rsa

ubuntu:
  hosts:
    host101.ubuntu.oci.lids:
  children:
    aws:
  vars:
    ansible_user: ubuntu
    ansible_become_method: sudo
```

### inventory/group\_vars/all.yml

```
---
agent_param_tlsaccept: ["psk"]
agent_param_tlsconnect: "psk"

host_templates: ["Linux by Zabbix agent"]
```

### playbook.yml

```
---
- name: example showcase
  hosts: all
  roles:
    - role: zabbix.zabbix.agent
    - role: zabbix.zabbix.host
    ### Zabbix API properties
    host_zabbix_api_server: zabbix.api.example
    host_zabbix_api_port: 8070
    host_zabbix_api_user: Admin
    host_zabbix_api_password: zabbix
```

```
$ ansible-playbook -i inventory/ playbook.yml
```

Hosts

Create host Import

Filter

Name	Items	Triggers	Graphs	Discovery	Web	Interface	Proxy	Templates	Status	Availability	Agent encryption	Info	Tags
<input type="checkbox"/> host01.ubuntu.aws.lids	Items 64	Triggers 23	Graphs 13	Discovery 3	Web	host01.ubuntu.aws.lids:10050		Linux by Zabbix agent	Enabled	ZBX	PSK None PSK CERT		managed
<input type="checkbox"/> host02.ubuntu.aws.lids	Items 64	Triggers 23	Graphs 13	Discovery 3	Web	host02.ubuntu.aws.lids:10050		Linux by Zabbix agent	Enabled	ZBX	PSK None PSK CERT		managed
<input type="checkbox"/> host03.ubuntu.aws.lids	Items 64	Triggers 23	Graphs 13	Discovery 3	Web	host03.ubuntu.aws.lids:10050		Linux by Zabbix agent	Enabled	ZBX	PSK None PSK CERT		managed
<input type="checkbox"/> host04.ubuntu.aws.lids	Items 64	Triggers 23	Graphs 13	Discovery 3	Web	host04.ubuntu.aws.lids:10050		Linux by Zabbix agent	Enabled	ZBX	PSK None PSK CERT		managed
<input type="checkbox"/> host05.ubuntu.aws.lids	Items 64	Triggers 23	Graphs 13	Discovery 3	Web	host05.ubuntu.aws.lids:10050		Linux by Zabbix agent	Enabled	ZBX	PSK None PSK CERT		managed
<input type="checkbox"/> host06.ubuntu.aws.lids	Items 64	Triggers 23	Graphs 13	Discovery 3	Web	host06.ubuntu.aws.lids:10050		Linux by Zabbix agent	Enabled	ZBX	PSK None PSK CERT		managed
<input type="checkbox"/> host07.ubuntu.aws.lids	Items 64	Triggers 23	Graphs 13	Discovery 3	Web	host07.ubuntu.aws.lids:10050		Linux by Zabbix agent	Enabled	ZBX	PSK None PSK CERT		managed
<input type="checkbox"/> host08.ubuntu.aws.lids	Items 64	Triggers 23	Graphs 13	Discovery 3	Web	host08.ubuntu.aws.lids:10050		Linux by Zabbix agent	Enabled	ZBX	PSK None PSK CERT		managed
<input type="checkbox"/> host101.ubuntu.oci.lids	Items 64	Triggers 23	Graphs 13	Discovery 3	Web	158.179.206.1:10050		Linux by Zabbix agent	Enabled	ZBX	PSK None PSK CERT		managed
<input type="checkbox"/> host301.alma.oci.lids	Items 68	Triggers 27	Graphs 14	Discovery 3	Web	129.151.195.145:10050		Linux by Zabbix agent	Enabled	ZBX	PSK None PSK CERT		managed

0 selected Enable Disable Export Mass update Delete

Displaying 10 of 10 found

# General settings

Variable	Type	Default	Description
agent_variant	int	1	The variant of Zabbix agent (1: Zabbix agentd, 2: Zabbix agent 2).
agent_major_version	string	6.0	The major version of Zabbix agent. Defaults to the latest LTS.
agent_minor_version	string		Zabbix agent minor version customization.
agent_https_proxy	string		Defines HTTPS proxy address for the packager.
agent_repository_mirror	string	<a href="https://repo.zabbix.com/">"https://repo.zabbix.com/"</a>	Defines repository mirror URL. You can override it to use self-hosted Zabbix repo mirror.

# User settings

Variable	Type	Default	Description
agent_service_user	string	zabbix	The user to run Zabbix agent.
agent_service_group	string	zabbix	User group for the custom user.
agent_service_uid	string		User ID for the custom user.
agent_service_gid	string		User group ID for the custom user group.

- Creates group.
- Creates user with home folder (defaults to `/home/{{ agent_service_user }}`).
- Adds systemd overrides to manage Zabbix agent pid file (`/run/zabbix_agent[d | 2]/zabbix_agent[d | 2].pid`).
- Changes Zabbix agent 2 sockets paths (to the folder of pid file).
- Changes logging path (to `/var/log/zabbix_agent[d | 2]/zabbix_agent[d | 2].log`).



# Firewall settings

Variable	Type	Default	Description
agent_apply_firewalld_rule	string	false	Defines application of firewalld rule. Possible options: ["auto", "force"]. Undefined or any other string will skip the rule application.
agent_apply_iptables_rule	boolean	false	Defines application of iptables rule. Possible options: [true, false].
agent_firewalld_zone	string	default	Firewalld zone for the rule application.
agent_firewall_allow_from	string		Limits source address of passive check using the firewall rule. For firewalld, this setting will change the rule from simple to rich rule.

The role allows adding simple firewall rules on the target machine to accept passive checks. Advanced firewall configuration is out of the scope of Zabbix agent role.

Firewalld is a recommended way of applying firewall rule as it works with both iptables and nftables. Note that iptables does not work in Ubuntu since 22.04. Firewalld should be installed on target machines. It is supported on RHEL- and Debian-based distributions.

# Logrotate settings

You can modify rotation options of **Zabbix agent[2]** log file. It requires default option overriding in **agent\_logrotate\_options** variable. This is a list type variable, and it defaults to the list of the following options:

- weekly
- maxsize 5M
- rotate 12
- compress
- delaycompress
- missingok
- notifempty
- create 0640 {{ agent\_service\_user }} {{ agent\_service\_group }}

Note that most distributions execute **logrotate** jobs on a daily basis by default. If you wish to change rotation calendar, modify cronjob or **systemd** timer accordingly or add separate cronjob/timer to process only Zabbix agent[2] log.

# Local path variable table

Variable	Type	Default	Description
agent_source_conf_dir	string		Path to the configuration folder on Ansible execution environment that needs to be transferred to the target machine and included in Zabbix agent configuration. For example, a folder with Userparameters.
agent_source_scripts_dir	string		Path to the scripts folder on Ansible execution environment. Will be copied under the service_user home folder. Scripts can be utilized by UserParameters.
agent_source_tlscafile	string		Path to the file on Ansible execution environment containing the top-level CA(s) certificates for peer certificate verification. Will be placed under service_user home folder and added to Zabbix agent configuration automatically.
agent_source_tlscertfile	string		Path to the file on Ansible execution environment containing the agent certificate or certificate chain. Will be placed under service_user home folder and added to Zabbix agent configuration automatically.
agent_source_tlscrlfile	string		Path to the file on Ansible execution environment containing revoked certificates. Will be placed under service_user home folder and added to Zabbix agent configuration automatically.
agent_source_tlskeyfile	string		Path to the file containing the agent private key. Will be placed under service_user home folder and added to Zabbix agent configuration automatically.

# SELinux settings

Variable	Type	Default	Description
agent_apply_seport	boolean	true	Adds custom agent port defined in param_listenport to SELinux port type zabbix_agent_port_t. Enabled by default and triggers when <code>agent_param_listenport</code> is not equal to 10050.
agent_apply_semodule	boolean	false	Adds SELinux policy extension to make a transition of Zabbix agent 2 to SE domain zabbix_agent_t. Additionally, it allows socket usage for the same domain. It also enables read permission on RPM database for <code>system.sw.packages.get</code> key.

host

role

# Host configuration

Variable	Type	Default	Description
host_name	string	{{ inventory_hostname }}	Unique name of the host.
host_hostgroups	list	{{ group_names }}	List of <b>hostgroup</b> names assigned to the host. At least one hostgroup needed. By default, uses the list of groups assigned to the host in Ansible inventory.
host_templates	list	[]	List of template names that should be linked to the host.
host_get_cert_info	boolean	False	Extract issuer and subject info from certificates, defined in <b>agent_source_tlscertfile</b> . Requires Openssl installation on Ansible execution environment.

**host\_interfaces:**

- **type:** agent

**ip:** '{{ ansible\_host if ansible\_host | ansible.utils.ipaddr else omit }}'

**dns:** '{{ ansible\_host if not ansible\_host | ansible.utils.ipaddr else omit }}'

**useip:** '{{ true if ansible\_host | ansible.utils.ipaddr else false }}'

**port:** '{{ agent\_param\_listenport | default(10050) }}

# Execution control with tags

Configuration tasks of agent role only:

```
ansible-playbook -i inventory playbook.yml -t config
```

Userparameter modification and reload (no agent restart):

```
ansible-playbook -i inventory playbook.yml -t userparam
```

REMOVE agent and host according to variables set:

```
ansible-playbook -i inventory playbook.yml -t remove
```

Reassure agent restart after deploy (handlers):

```
ansible-playbook -i inventory playbook.yml -t all,restart
```

Allow unsupported OS or Zabbix versions:

```
ansible-playbook -i inventory playbook.yml --skip-tags assert
```

# No Inventory yet? - Try our `zabbix_inventory` plugin

```
---
plugin: zabbix.zabbix.zabbix_inventory

zabbix_api_url: 'http://zabbix.api.example:8070'
zabbix_user: 'Admin'
zabbix_password: 'zabbix'

#####

query:
  selectHostGroups: ['name']
  selectInterfaces: ['ip', 'dns', 'useip']

keyed_groups:
  - key: zabbix_hostgroups | map(attribute='name')
    separator: ""

compose:
  ansible_host: 'zabbix_interfaces[0].ip if zabbix_interfaces[0].useip | bool
                else zabbix_interfaces[0].dns'
```

```
[project]$ ansible-inventory -i zabbix_inventory.yml --graph
@all:
|--@ungrouped:
|--@Zabbix_servers:
| |--Zabbix server
|--@oci:
| |--host101.ubuntu.oci.lds
| |--host301.alma.oci.lds
|--@ubuntu:
| |--host101.ubuntu.oci.lds
| |--host01.ubuntu.aws.lds
| |--host02.ubuntu.aws.lds
| |--host03.ubuntu.aws.lds
| |--host04.ubuntu.aws.lds
| |--host05.ubuntu.aws.lds
| |--host06.ubuntu.aws.lds
| |--host07.ubuntu.aws.lds
| |--host08.ubuntu.aws.lds
|--@aws:
| |--host01.ubuntu.aws.lds
| |--host02.ubuntu.aws.lds
| |--host03.ubuntu.aws.lds
| |--host04.ubuntu.aws.lds
| |--host05.ubuntu.aws.lds
| |--host06.ubuntu.aws.lds
| |--host07.ubuntu.aws.lds
| |--host08.ubuntu.aws.lds
```

```
- name: 'Zabbix API : Host presence'
  throttle: 1
  delegate_to: 'my.zabbix.server'
  vars:
    ansible_connection: httpapi
    ansible_network_os: zabbix.zabbix.zabbix
    ansible_httpapi_port: 80
    zabbix_api_token: '{{ vault_secret_token }}'
  zabbix.zabbix.host:
    host_state: 'present'
    host_name: '{{ inventory_hostname }}'
    host_description: 'Managed by ansible'
    host_hostgroups: '{{ group_names }}'
    host_templates: ["Cisco IOS SNMP"]
    host_interfaces:
      - type: 'snmp'
        version: '2'
        ip: '{{ ansible_host }}'
        community: 'readonly'
    host_tags: [{"name": "snmpv", "value": "2"},
{"name": "managed"}]
    host_inventory_mode: 'automatic'
    host_status: 'enabled'
    host_proxy: 'proxy_aws_stockholm'
```

# Add **agent-less** devices with **host** module

- ✓ Use "delegate\_to" for zabbix api connection
- ✓ Use "throttle" to limit API call pressure on Zabbix



```
- name: 'SET FACT : prepare list of hostgroups'  
delegate_to: 'localhost'  
run_once: true  
ansible.builtin.set_fact:  
  full_hostgroup_list: '{{  
    ((full_hostgroup_list | default([]))  
    + hostvars[item]["zabbix_host_hostgroups"]) | unique  
  }}'  
loop: '{{ ansible_play_hosts }}'
```

```
- name: 'Zabbix API : Hostgroup presence'  
delegate_to: 'my.zabbix.server'  
run_once: true  
vars:  
  ansible_connection: httpapi  
  ansible_network_os: zabbix.zabbix.zabbix  
  ansible_httpapi_port: 80  
  zabbix_api_token: '{{ vault_secret_token }}'  
zabbix.zabbix.zabbix_hostgroup:  
  name: '{{ full_hostgroup_list }}'  
  state: present
```

# Speed up

## hostgroup module

- ✓ Prepare list of all hostgroups
- ✓ Add full list once

# Execution Environment

Run ansible and all dependencies from prebuild containers.  
Supports: Docker & Podman

```
---
version: 3
dependencies:
  ansible_core:
    package_pip: ansible-core==2.15
  ansible_runner:
    package_pip: ansible-runner
galaxy:
  collections:
    - name: zabbix.zabbix
      version: '>=1.3.6'
    - name: ansible.utils
      version: '>=2.8.0'
    - name: ansible.posix
      version: '>=1.4.0'
    - name: ansible.netcommon
      version: '>=3.1.1'
python_interpreter:
  package_system: "python311"
  python_path: "/usr/bin/python3.11"
python:
  - netaddr>=0.8
  - Jinja2==3.1.2
system:
  - openssl
images:
  base_image:
    name: quay.io/centos/centos:stream9
```

## 1. Install navigator

```
python3 -m pip install ansible-navigator
```

## 2. Create file "execution-environment.yml" in empty folder:

## 3. Build your image:

```
ansible-builder build -t custom-ee -v3
```

## 4. Run your playbooks with local image:

```
ansible-navigator run playbook.yml -i inventory/ --eei custom-ee  
--pull-policy missing --mode stdout
```

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

**ZABBIX**

**Thank you!**

---

**Andrey Tochko**

Integration engineer