Zabbix and Ansible

Like Wine and Cheese

Riga, 2022

About Me

Zabbix user for almost 20 years

Author of multiple Zabbix related programs

Linux user for almost 30 years.

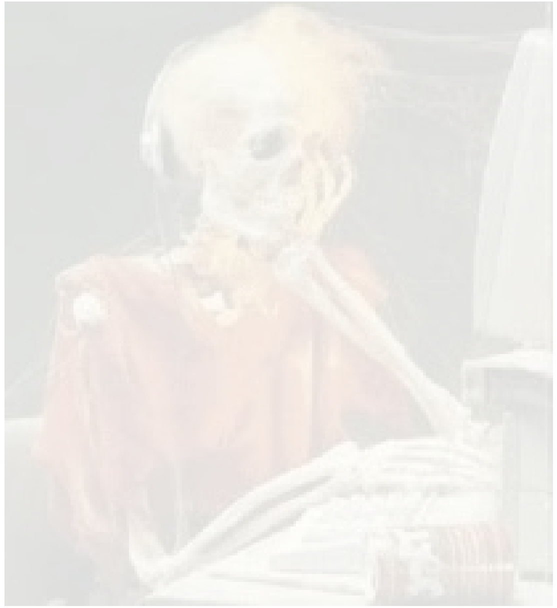
In the beginning

Build servers

In the beginning

Build servers

Install Zabbix, Database and Web packages



In the beginning

Build servers

Install Zabbix, Database and Web packages

Configure Database

In the beginning

Build servers

Install Zabbix, Database and Web packages

Configure Database

Configure Zabbix Server

In the beginning

Build servers

Install Zabbix, Database and Web packages

Configure Database

Configure Zabbix Server

Configure Web Server

In the beginning

Build servers

Install Zabbix, Database and Web packages

Configure Database

Configure Zabbix Server

Configure Web Server

Configure Zabbix agent on Zabbix Server

In the beginning

Build servers

Install Zabbix, Database and Web packages

Configure Database

Configure Zabbix Server

Configure Web Server

Configure Zabbix agent on Zabbix Server

Configure Zabbix agent on Host N

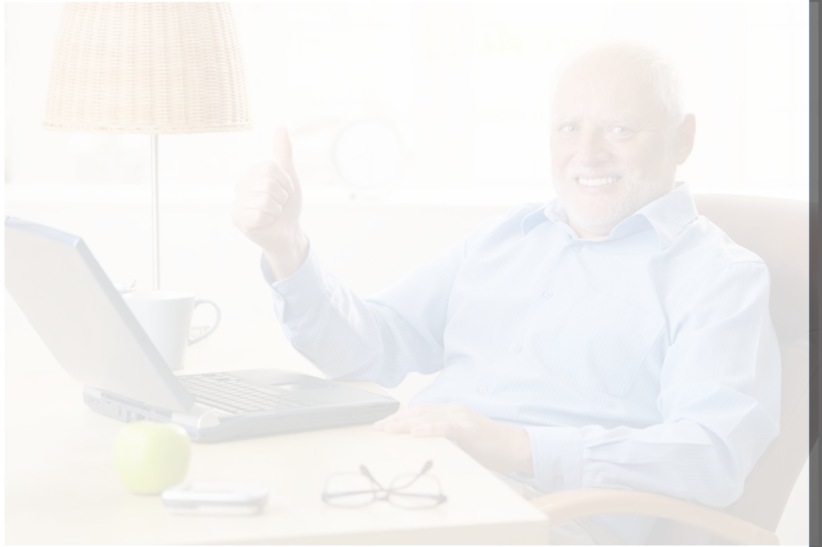
The Ansible Way

Build servers

The Ansible Way

Build servers

Write Ansible Inventory



The Ansible Way

Build servers

Write Ansible Inventory

Write Ansible Playbook using collections

The Ansible Way

Build servers

Write Ansible Inventory

Write Ansible Playbook using collections

Run Playbook

The Ansible Way

Build servers

Write Ansible Inventory

Write Ansible Playbook using collections

Run Playbook

Fix typo

The Ansible Way

Build servers

Write Ansible Inventory

Write Ansible Playbook using collections

Run Playbook

Fix typo

Rerun on hundreds of nodes

The Ansible Way

Build servers

Write Ansible Inventory

Write Ansible Playbook using collections

Run Playbook

Fix typo

Rerun on hundreds of nodes

Take a nap

Inventories, the foundation of Ansible

Inventories define the work

Allow for grouping of hosts

Allow for hierarchies

Can be designed for scale

Inventories, the foundation of Ansible

Static Inventory

- Can be one large file
- Can be multiple YAML files in a directory heirarchy
- Extremely flexible

```
---
all:
  hosts:
    zbxs1.lab.example.com:
    zbxp1.lab.example.com:
    zbxc1.lab.example.com:
  children:
    zabbix:
      hosts:
        zbxs1.lab.example.com:
        zbxp1.lab.example.com:
        zbxc1.lab.example.com:
    zabbix_server:
      hosts:
        zbxs1.lab.example.com:
    zabbix_proxy:
      hosts:
        zbxp1.lab.example.com:
    zabbix_client:
      hosts:
        zbxc1.lab.example.com:
```

Inventories, the foundation of Ansible

Zabbix Inventory

- Group hosts by status
- Get interface stats
- Use first Zabbix agent Interface as ansible hostname

```
---
plugin: community.zabbix.zabbix_inventory
server_url: https://zbxsrvr.example.com
login_user: zabconf
login_password: 10years2022
host_zapi_query:
  selectInterfaces: 'extend'
validate_certs: false
groups:
  enabled: zbx_status == "0"
  disabled: zbx_status == "1"
compose:
  ansible_host: (zbx_interfaces |
selectattr('type','==','1') | first).ip
```

Inventories, the foundation of Ansible

```
"Zabbix server": {
                "ansible_host": "127.0.0.1",
                "zbx_active_available": "1",
                "zbx_auto_compress": "1",
                "zbx_custom_interfaces": "0",
                "zbx_description": "",
                "zbx_flags": "0",
                "zbx_host": "Zabbix server",
                "zbx_hostid": "10084",
                "zbx_interfaces": [{
                        "available": "2",
                        "details": [],
                        "disable_until": "1664867100",
                        "dns": "zbxsrvr.example.com",
                        "error": "Received empty response from Zabbix Agent at [127.0.0.1]. Assuming that agent dropped connection because of access permissions.",
                        "errors_from": "1664782020",
                        "hostid": "10084",
                        "interfaceid": "1",
                        "ip": "127.0.0.1",
                        "main": "1",
                        "port": "10050",
                        "type": "1",
                        "useip": "1"}],
                "zbx_inventory_mode": "1",
                "zbx_ipmi_authtype": "-1",
                "zbx_ipmi_password": "",
                "zbx_ipmi_privilege": "2",
                "zbx_ipmi_username": "",
                "zbx_maintenance_from": "0",
                "zbx_maintenance_status": "0",
                "zbx_maintenance_type": "0",
                "zbx_maintenanceid": "0",
                "zbx_name": "Zabbix server",
                "zbx_proxy_address": "",
                "zbx_proxy_hostid": "0",
                "zbx_status": "0",
                "zbx_templateid": "0",
                "zbx_tls_accept": "1",
                "zbx_tls_connect": "1",
                "zbx_tls_issuer": "",
                "zbx_tls_subject": "",
                "zbx_uuid": ""
            }
```

Inventories, the foundation of Ansible

Google Compute Inventory

- Group hosts by
  - Status
  - Labels
- Configure variables from metadata and other sources
- Default everything to a ‘Zabbix’ group

```
---
plugin: gcp_compute
projects:
- zabconf-2022
auth_kind: serviceaccount
service_account_file: /home/nelsonab/ansible/zabconf2022/zabconf-2022-
a96622a75d0b.json
hostnames: name
keyed_groups:
- prefix: gcp
  key: labels
- prefix: status
  key: status
compose:
  ansible_user: nelsonab
  ansible_host: networkInterfaces[0].accessConfigs[0].natIP
  zabbix_agent_server: metadata.zabbix_agent_server | default(omit)
  zabbix_agent_serveractive: metadata.zabbix_agent_serveractive | default(omit)
  zabbix_proxy: metadata.zabbix_proxy | default(omit)
groups:
  zabbix_server: '"server" in labels["role"]'
  zabbix_proxy: '"proxy" in labels["role"]'
  zabbix_client: '"client" in labels["role"]'
  zabbix: true
```

Inventories, the foundation of Ansible

You can combine the two!

- Allows for per-host data override
- Allows for per-group data override
- Sometimes easier than adding metadata

BUT!

- ‘ansible_group_priority’ does not work

```
gcp_inventory
+- google_inventory.gcp.yml
+- group_vars
|  +- zabbix.yml
|  +- zabbix_proxy.yml
+- host_vars
   +- server-1
   +- client-1
   +- client-2
```

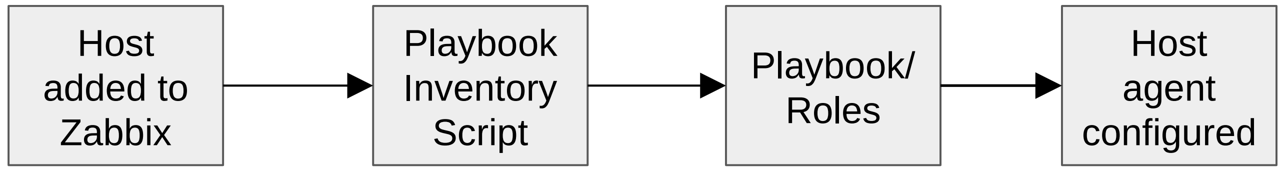
Let’s install Zabbix!

```
---
- hosts: zabbix_server
  become: true

  pre_tasks:
 - name: Install needed packages
    dnf:
      name: "{{ item }}"
      state: present
    loop:
    - python3-netaddr
    - python3-libsemanage

  - name: allow execmem selinux
    ansible.posix.seboolean:
      name: httpd_execmem
      state: yes
      persistent: yes
```

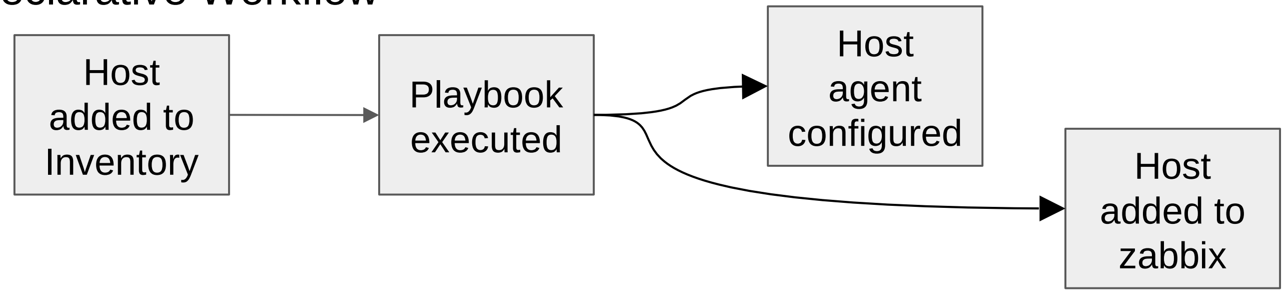
```
- name: Copy SSL certs
  copy:
    src: "certs/{{ item.name }}"
    dest: "{{ item.dest }}"
    owner: "{{ item.owner | default('root') }}"
    group: "{{ item.group | default('root') }}"
    mode: "{{ item.mode | default('0644') }}"
  loop:
  - name: self-signed.key
    dest: /etc/pki/tls/private/
    mode: '0600'
  - name: self-signed.crt
    dest: /etc/pki/tls/certs/
  register: certs_copy

roles:
  - role: geerlingguy.apache
  - role: geerlingguy.php
  - role: community.zabbix.zabbix_server
  - role: community.zabbix.zabbix_web
  - role: community.zabbix.zabbix_agent
```

Workflow Considerations

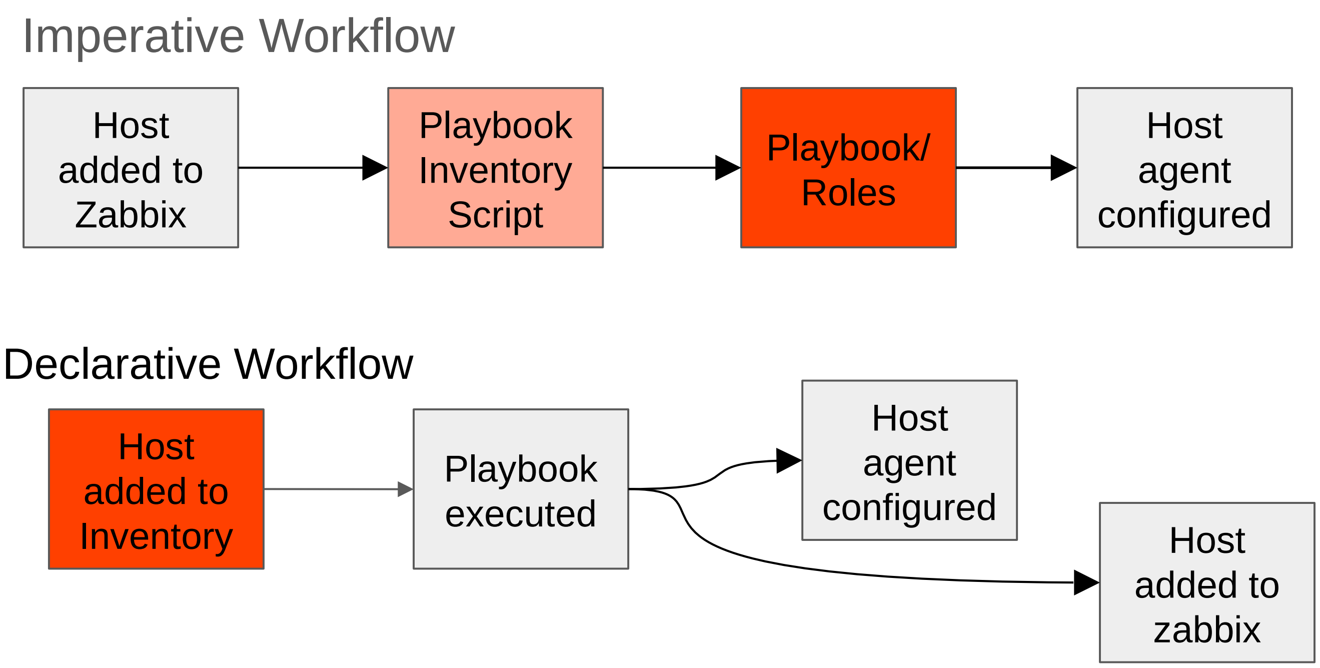
Imperative Workflow



Declarative Workflow



Where’s the control of the logic?



Questions