Ansible:
automation for everyone

Simone Tiraboschi
Principal Software Engineer
Red Hat EMEA

October 2019
THE WORLD IS AUTOMATING
Those who succeed in automation will win
AUTOMATION IN ENTERPRISE IT TODAY
AUTOMATION IN ENTERPRISE IT TODAY

AUTOMATED SILOS ARE STILL SILOS
Isolated scripts

AUTOMATES FUNCTIONS

A culture of automation

AUTOMATES ORGANIZATIONS
AUTOMATION FOR EVERYONE

Designed around the way people work and the way people work together.
### WHY ANSIBLE IS VIRALLY ADOPTED

<table>
<thead>
<tr>
<th>SIMPLE</th>
<th>POWERFUL</th>
<th>AGENTLESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human readable automation</td>
<td>App deployment</td>
<td>Agentless architecture</td>
</tr>
<tr>
<td>No special coding skills needed</td>
<td>Configuration management</td>
<td>Uses OpenSSH &amp; WinRM</td>
</tr>
<tr>
<td>Tasks executed in order</td>
<td>Workflow orchestration</td>
<td>No agents to exploit or update</td>
</tr>
<tr>
<td>Usable by every team</td>
<td>Network automation</td>
<td>Get started immediately</td>
</tr>
</tbody>
</table>

**GET PRODUCTIVE QUICKLY**

**ORCHESTRATE THE APP LIFECYCLE**

**MORE EFFICIENT & MORE SECURE**
8

1900+
Ansible modules

31,000+
Stars on GitHub

500,000+
Downloads a month
REDIS HAT ANSIBLE TOWER
Scale + operationalize your automation

CONTROL

KNOWLEDGE

DELEGATION

REDIS HAT ANSIBLE ENGINE
Support for your Ansible automation

SIMPLE

POWERFUL

AGENTLESS

FUELED BY AN INNOVATIVE OPEN SOURCE COMMUNITY
It is a simple automation language that can perfectly describe an IT application infrastructure in Ansible Playbooks.

It’s an automation engine that runs Ansible Playbooks.

Ansible Tower/AWX is an enterprise framework for controlling, securing and managing your Ansible automation with a UI and a RESTful API.
**Why Ansible?**

**Agentless** - Ansible relies on OpenSSH. Ansible does not require any remote agents: it delivers all modules to remote systems and executes tasks, as needed, to enact the desired configuration.

**Idempotency** - An operation is idempotent if the result of performing it once is exactly the same as the result of performing it repeatedly without any intervening actions. The core idea here is that you only do things if they are needed and that things are repeatable without side effects.

**Declarative Not Procedural** - Ansible features an state-driven resource model that describes the desired state of computer systems and services, not the paths to get them to this state. No matter what state a system is in, Ansible understands how to transform it to the desired state (and also supports a "dry run" mode to preview needed changes).

**Tiny Learning Curve** - Ansible, YAML based, is quite easy to learn: it doesn’t require any extra knowledge.

**De facto standard**: Ansible 2.8 ships more than 2800 ready to use modules, 30.000+ github stars, 3.500 community contributors.
oVirt Ansible modules

- ovirt_affinity_group - Module to manage affinity groups in oVirt/RHV
- ovirt_affinity_label - Module to manage affinity labels in oVirt/RHV
- ovirt_affinity_label_facts - Retrieve facts about one or more oVirt/RHV affinity labels
- ovirt_api_facts - Retrieve facts about the oVirt/RHV API
- ovirt_auth - Module to manage authentication to oVirt/RHV
- ovirt_cluster - Module to manage clusters in oVirt/RHV
- ovirt_cluster_facts - Retrieve facts about one or more oVirt/RHV clusters
- ovirt_datacenter - Module to manage data centers in oVirt/RHV
- ovirt_datacenter_facts - Retrieve facts about one or more oVirt/RHV datacenters
- ovirt_disk - Module to manage Virtual Machine and floating disks in oVirt/RHV
- ovirt_disk_facts - Retrieve facts about one or more oVirt/RHV disks
- ovirt_event - Create or delete an event in oVirt/RHV
- ovirt_event_facts - This module can be used to retrieve facts about one or more oVirt/RHV events
- ovirt_external_provider - Module to manage external providers in oVirt/RHV
- ovirt_external_provider_facts - Retrieve facts about one or more oVirt/RHV external providers
- ovirt_group - Module to manage groups in oVirt/RHV
- ovirt_group_facts - Retrieve facts about one or more oVirt/RHV groups
- ovirt_host - Module to manage hosts in oVirt/RHV
- ovirt_host_facts - Retrieve facts about one or more oVirt/RHV hosts
- ovirt_host_network - Module to manage host networks in oVirt/RHV
- ovirt_host_pools - Module to manage power management of hosts in oVirt/RHV
- ovirt_host_storage_facts - Retrieve facts about one or more oVirt/RHV HostStorages (applicable only for block storage)
- ovirt_instance_type - Module to manage Instance Types in oVirt/RHV
- ovirt_mac_pool - Module to manage MAC pools in oVirt/RHV
- ovirt_network - Module to manage logical networks in oVirt/RHV
- ovirt_network_facts - Retrieve facts about one or more oVirt/RHV networks
- ovirt_nic - Module to manage network interfaces of Virtual Machines in oVirt/RHV
- ovirt_nic_facts - Retrieve facts about one or more oVirt/RHV network ports
How ansible works
Ansible playbook example

---
- name: Setup oVirt environment
  hosts: ovirt
  tasks:
    - block:
      - name: Include oVirt password
        no_log: true
        include_vars: ovirt_password.yml
      - name: Obtain SSO token
        no_log: false
        ovirt_auth:
          url: "{{ url }}"
          username: "{{ username }}"
          password: "{{ password }}"
          ca_file: "{{ ca_file }}"
      - name: Create datacenter
        ovirt_datacenters:
          auth: "{{ ovirt_auth }}"
          name: "{{ datacenter }}"
          description: mydatacenter
      - name: Create logical network
        ovirt_networks:
          auth: "{{ ovirt_auth }}"
          name: mynetwork
          datacenter_name: mydatacenter
          vm_network: false
      - name: Create cluster
        ovirt_clusters:
          auth: "{{ ovirt_auth }}"
          datacenter_name: "{{ datacenter }}"
          name: "{{ cluster }}"
          cpu_type: Intel Nehalem Family
          description: mycluster
          compatibility_version: 4.1
      - name: Add host using public key
        ovirt_hosts:
          auth: "{{ ovirt_auth }}"
          public_key: true
          cluster: "{{ cluster }}"
          name: "{{ host }}"
          address: "{{ host_address }}"
oVirt Ansible Roles

oVirt maintains multiple Ansible roles that can be deployed to easily configure and manage various parts of the oVirt infrastructure. Ansible roles provide a method of modularizing your Ansible code, in other words; it enables you to break up large playbooks into smaller reusable files. This enables you to have a separate role for each component of the infrastructure, and allows you to reuse and share roles with other users. For more information about roles, see Creating Reusable Playbooks in the Ansible Documentation.

Currently we have implemented following Ansible roles:

- **oVirt.cluster-upgrade** - easily upgrade your oVirt clusters, host by host.
- **oVirt.disaster-recovery** - plan, failover and fallback oVirt in Disaster Recovery scenarios.
- **oVirt.engine-setup** - setup your oVirt Engine via Ansible.
- **oVirt.hosted-engine-setup** - setup your oVirt Hosted-Engine via Ansible.
- **oVirtinfra** - setup a complete oVirt setup (data centers, clusters, hosts, networks...) via this role.
- **oVirt.image-template** - easily create VM templates (via Glance or QCOW2 download)
- **oVirt.manageriq** - install and configure a ManagerIQ (or CloudForms) VM appliance on your oVirt!
- **oVirt.repositories** - set up the required oVirt repositories on your hosts.
- **oVirt.vm-infra** - configure a complete VM setup (create and configure VMs and their properties)
- **oVirt.v2v-conversion-host** - define a host as a target for VMware to oVirt migration.
- **oVirt.shutdown-env** - shutdown the whole environment in a clean and ordered way.
Install a role

```
rpm -i oVirt.infra
```

```
# yum install ovirt-ansible-infra
```

```
# ansible-galaxy install oVirt.infra
```

Name and summary matches only, use "search all" for everything.