

# Ansible et VMware

État des lieux - Ansible  
Montréal

# Gonéri Le Boudier

- développeur dans l'équipe "Cloud" de Ansible
- collections VMware principalement
- infrastructure de test
- Gonéri sur Matrix

<https://github.com/goneri> - <https://goneri.leboudier.net>

# community.vmware

- Commencé en 2016.
- Maintenu par la communauté
  - Une release tout les mois
- Utilise pyVmomi principalement
  - Interface SOAP

# community.vmware

- Peut être utilisé avec vCenter
  - et souvent directement avec un ESXI
- Les couvertures fonctionnels
  - cluster
  - datacenter
  - content library
  - Distributed Resource Scheduler (DRS)
  - guest (a.k.a VM)
  - Distributed and regular Virtual Switch
  - host (a.k.a ESXi)
  - resource pool
  - VSAN
  - and more

# community.vmware

- avantages
  - large collection de modules (150+)
  - couvre beaucoup de fonctionnalités
- inconvénients
  - qualité des modules variables
  - pas supporté par Red Hat

## pyVmomi et SOAP

- dépendance pour la majorité des modules `community.vmware`
- pas supporté par VMware
- et en 2019...

# vmware.vmware\_rest

- Nouvelle collection
- Développement commencé en 2019
- Première publication en 2020 (1.0.0)
- Basé sur l'interface REST de vSphere
- Identification des ressources basées sur le MOID

## vmware.vmware\_rest

- appliance\_: modules dédiés à la gestion de l'appliance vSphere (VCSA)
- vcenter\_: cycle de vie d'une machine virtuelle



# vmware.vmware\_rest

- branche 1.0.0
  - Supporte vSphere 6.7.0 à 7.0.1
- depuis la version 2
  - Supporte vSphere 7.0.2

# Identification des ressources (1/3)

- community.vmware
  - nom, un dossier peut être ajouté ainsi qu'un datacenter

- vmware\_guest:

**folder:** "vm"

**name:** test\_vm1

**datacenter:** dc1

## Identification des ressources (2/3)

- set\_fact:

  - folder\_name: "f1"

- vmware\_guest:

  - state: absent

  - name: mon-server

  - folder: "{{ folder\_name }}"

## Identification des ressources (3/3)

- set\_fact:

folder\_name: "" ⚠ un champ texte vide!

- vmware\_guest:

state: absent

name: mon-server

folder: "{{ folder\_name }}"

## vmware.vmware\_rest - MOID (1/2)

- Les ressources sont identifiés par le MOID
  - ex: host-1014 pour un ESXI
- Trois façons de passer un nom à un MOID
  - avec les modules `_info` et un filtre

- name: get all the clusters called my\_cluster

vcenter\_cluster\_info:

**filter\_names:**

- my\_cluster

## vmware.vmware\_rest - MOID (2/2)

- Sinon avec les greffons de filtre:

```
"{{ lookup('vmware.vmware_rest.cluster_moid',  
  '/my_dc/host/my_cluster' ) }}"
```

- Enfin, avec `register` suite à une création de ressource

- name: **Attach a VM to a dvs switch**

vmware.vmware\_rest.vcenter\_vm\_hardware\_ethernet:

vm: '{{ test\_vm1\_info.id }}'

pci\_slot\_number: 4

backing:

type: DISTRIBUTED\_PORTGROUP

network: '{{ my\_portgroup.value[o].network }}'

start\_connected: false

register: **vm\_hardware\_ethernet\_1**

- name: **Turn the NIC's start\_connected flag on**

vmware.vmware\_rest.vcenter\_vm\_hardware\_ethernet:

nic: '{{ **vm\_hardware\_ethernet\_1**.id }}'

start\_connected: true

vm: '{{ test\_vm1\_info.id }}'

# vmware.vmware\_rest

- avantages
  - paramètres consistents
  - supporté par Red Hat
    - dans l'Execution Environment par défaut
  - plus rapide que `community.vmware`
- inconvénients
  - couverture fonctionnel plus faible



**Demo**

- name: Create a VM

vmware.vmware\_rest.vcenter\_vm:

placement:

cluster: "{{ lookup('vmware.vmware\_rest.cluster\_moid', '/my\_dc/host/my\_cluster') }}"

datastore: "{{ lookup('vmware.vmware\_rest.datastore\_moid', '/my\_dc/datastore/local') }}"

folder: "{{ lookup('vmware.vmware\_rest.folder\_moid', '/my\_dc/vm') }}"

resource\_pool: "{{ lookup('vmware.vmware\_rest.resource\_pool\_moid', '/my\_dc/host/my\_cluster/Resources') }}"

name: test\_vm1

guest\_OS: RHEL\_7\_64

hardware\_version: VMX\_11

memory:

hot\_add\_enabled: true

size\_MiB: 1024

(...)

Merci