

#ANSIBLEAUTOMATES



ANSIBLE

HYBRID CLOUD AUTOMATION WITH ANSIBLE TOWER

Alessandro Arrichiello
ale@redhat.com



Michele Naldini
maldini@redhat.com



Solution Architects - Red Hat

WHAT IS ANSIBLE?

Ansible is a simple agentless idempotent **automation solution**

- **Playbooks** can contain one or many **plays**
- **Tasks** are grouped together via **plays**
 - **Play** operates on a set of **hosts**
 - **Hosts** are grouped/managed via an **inventory**
- **Task** functionality provided by ad hoc **modules** or **custom commands**
- **Roles** are bundled re-usable sets of content, variables, and tasks

Inventory

- Defines the infrastructure
- Can be sourced from a static text file or dynamically

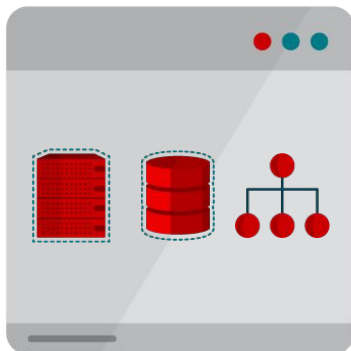
```
[web]  
web01.example.com  
web02.example.com
```

```
- name: install bash
  hosts: web
  become: yes
  vars:
    install_packages:
      - bash

  tasks:
    - name: install packages with yum
      yum: name="{{ install_packages }}" state=present
```

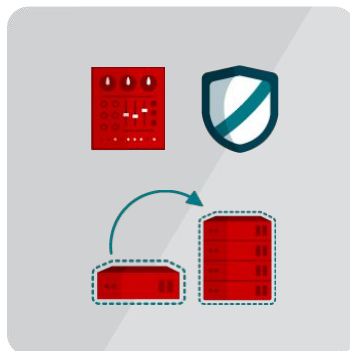
PROVISIONING RH-V VMs WITH ANSIBLE TOWER

Red Hat Virtualization is an easy to use software-defined platform for virtualized Linux and Windows built on Red Hat Enterprise Linux and Kernel-based Virtual Machine (KVM) technologies.



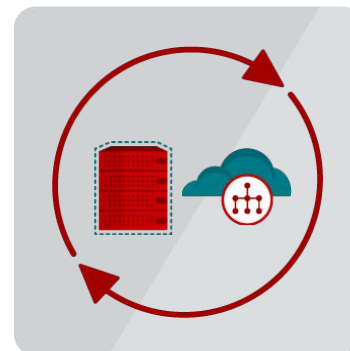
CENTRALIZED MANAGEMENT

Virtualized compute, network, & storage resources using the open source Kernel-based Virtual Machine (KVM) hypervisor



AUTOMATED WORKLOAD

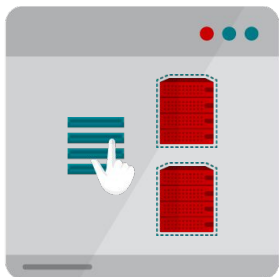
Management, scalability, and security features for virtualized applications



OPTIMIZES CURRENT I.T.

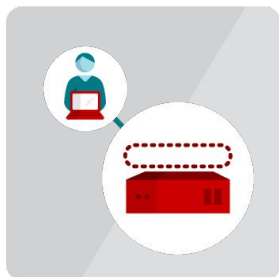
Integrates with future technologies using RESTful application programming interface (API)

EASY TO OPERATIONALIZE, EASY TO AUTOMATE, EASY ON THE BUDGET, NO VENDOR LOCK-IN



RED HAT VIRTUALIZATION MANAGER

- Designed for large scale (500+ hosts and 5,000+ VMs)
- REST API to integrate with Red Hat portfolio, third-party applications, backup and recovery software
- Can be integrated with existing infrastructure
Active directory, Red Hat CloudForms®, OpenStack, etc.

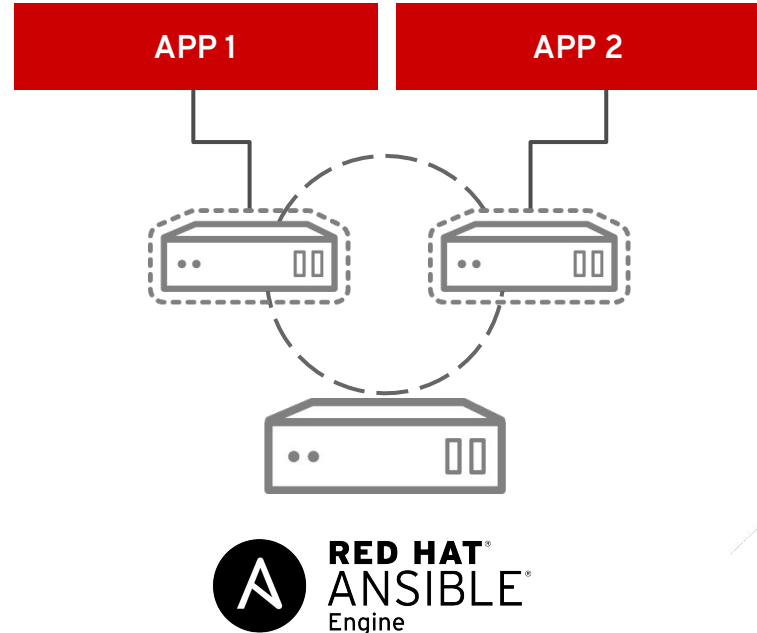


RH-V HYPERVISOR + COCKPIT

- Small footprint hypervisor to run your VMs
- Cockpit included inside Red Hat Virtualization Host
- Useful to configure networking, storage, subscriptions, and other aspects of the virtualization host
- Can be used to deploy Red Hat Virtualization in high availability

RED HAT VIRTUALIZATION INCLUDES ANSIBLE AUTOMATION ROLE

- Removes manual steps from deployment and reconfiguration.
- Streamlines operations, frees up resources to focus on strategic initiatives
- Includes and supports **Ansible Automation roles** for Red Hat Virtualization:
 - Virtual DCs
 - Clusters
 - Virtual machines
 - Virtual networks
 - Virtual storage
 - Configuration Mgmt



```
- name: Rh-v build VMs
hosts: localhost
connection: local
gather_facts: false
vars_files:
  # Contains encrypted `engine_password` variable using ansible-vault
  - passwords.yml
```

```
vars:
  engine_url: https://rhvm-5c04.rhpd.s.opentlc.com/ovirt-engine/api
  engine_user: admin@internal
  wait_for_ip: true
```

```
http_vm:
  cluster: Production
  domain: rhpd.s.opentlc.com
  template: rhel-7.5-template
  memory: 2GiB
  cores: 2
  username: root
  root_password: password
  state: running
```

disks:

- size: 50GiB
name: data
storage_domain: glusterSD1
interface: virtio

nics:

- name: nic1
network: vm public net
profile: vm public net

cloud_init:

- nic_on_boot: yes

vms:

- name: postgresql-vm-0
tag: postgresqlvm
profile: "{{ db_vm }}"
- name: apache-vm-0
tag: httpvm
profile: "{{ http_vm }}"
- name: apache-vm-1
tag: httpvm
profile: "{{ http_vm }}"

roles:

- ovirt.vm-infra

PROVISIONING AZURE VMS WITH ANSIBLE TOWER

Create a credentials file for Ansible on your Cloud / Shell.

```
$ az ad sp create-for-rbac
```

To find out what your subscription ID is, type in:

```
$ az account show --query "{ subscription_id: id }"
```

Copy output into a text file so that you can copy/paste it later:

```
"subscription_id": "854c5e9a-ed49-687e-bc7a-96ed7315095"
```

Then, type this command in:

```
$ az ad sp create-for-rbac --query '{"client_id": appId, "secret": password, "tenant": tenant}'
```

Output like this should show up:

```
{  
  "client_id": "eec5624a-90f8-4386-8a87-02730b5410d5",  
  "secret": "531dcffa-3aff-4488-99bb-4816c395ea3f",  
  "tenant": "72f988bf-86f1-41af-91ab-2d7cd011db47"  
}
```

```
$ cd ~/.azure  
$ vi ~/.azure/credentials
```

Format for the credentials file:

```
[default]  
subscription_id=<your-subscription_id>  
client_id=<security-principal-appid>  
secret=<security-principal-password>  
tenant=<security-principal-tenant>
```

Create an Ansible playbook named *azure_create_vm.yml*

The following slides will show you the content that should be in that playbook and how it works.

*Note: The text in **red** indicate arbitrary names for things that you can change/customize.*


```
- name: Create Azure VM
  hosts: localhost
  connection: local
  tasks:
    - name: Create virtual network
      azure_rm_virtualnetwork:
        resource_group: webinar-test
        name: webinarVnet
        address_prefixes: "10.0.0.0/16"
```

```
- name: Add subnet
  azure_rm_subnet:
    resource_group: webinar-test
    name: webinarSubnet
    address_prefix: "10.0.1.0/24"
    virtual_network: webinarVnet
```

```
- name: Create public IP address
  azure_rm_publicipaddress:
    resource_group: webinar-test
    allocation_method: Static
    name: myPublicIP
```

```
- name: Create Network Security Group that allows SSH
  azure_rm_securitygroup:
    resource_group: webinar-test
    name: webinarNetworkSecurityGroup
    rules:
      - name: SSH
        protocol: Tcp
        destination_port_range: 22
        access: Allow
        priority: 1001
        direction: Inbound
```

```
- name: Create virtual network interface card
  azure_rm_networkinterface:
    resource_group: webinar-test
    name: myNIC
    virtual_network: webinarVnet
    subnet: webinarSubnet
    public_ip_name: myPublicIP
    security_group: webinarNetworkSecurityGroup
```

```
- name: Create VM
  azure_rm_virtualmachine:
    resource_group: webinar-test
    name: WebinarVM
    vm_size: Standard_DS1_v2
    admin_username: azureuser
    ssh_password_enabled: false
    ssh_public_keys:
      - path: /home/azureuser/.ssh/authorized_keys
        key_data: "ssh-rsa AAAAB3Nz{snip}hwhqT9h"
    network_interfaces: myNIC
    image:
      offer: CentOS
      publisher: OpenLogic
      sku: '7.3'
      version: latest
```

AZURE RESOURCES READY!

ANSIBLE

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the Microsoft Azure logo, a search bar, and user information for 'mperz@AnsibleWork...'. The main content area displays the 'webinar-test' resource group. On the left, there is a sidebar with navigation options like 'Create a resource', 'All services', and 'FAVORITES'. The central pane shows the 'Overview' section with a search bar and a list of settings. The right pane shows a table of resources under the 'Essentials' section. A red box highlights the following resources:

	NAME	TYPE	LOCATION
<input type="checkbox"/>	myNIC	Network interface	East US
<input type="checkbox"/>	myPublicIP	Public IP address	East US
<input type="checkbox"/>	webinarNetworkSecurityGroup	Network security group	East US
<input type="checkbox"/>	WebinarVM	Virtual machine	East US
<input type="checkbox"/>	webinarvm2989	Storage account	East US
<input type="checkbox"/>	webinarVnet	Virtual network	East US

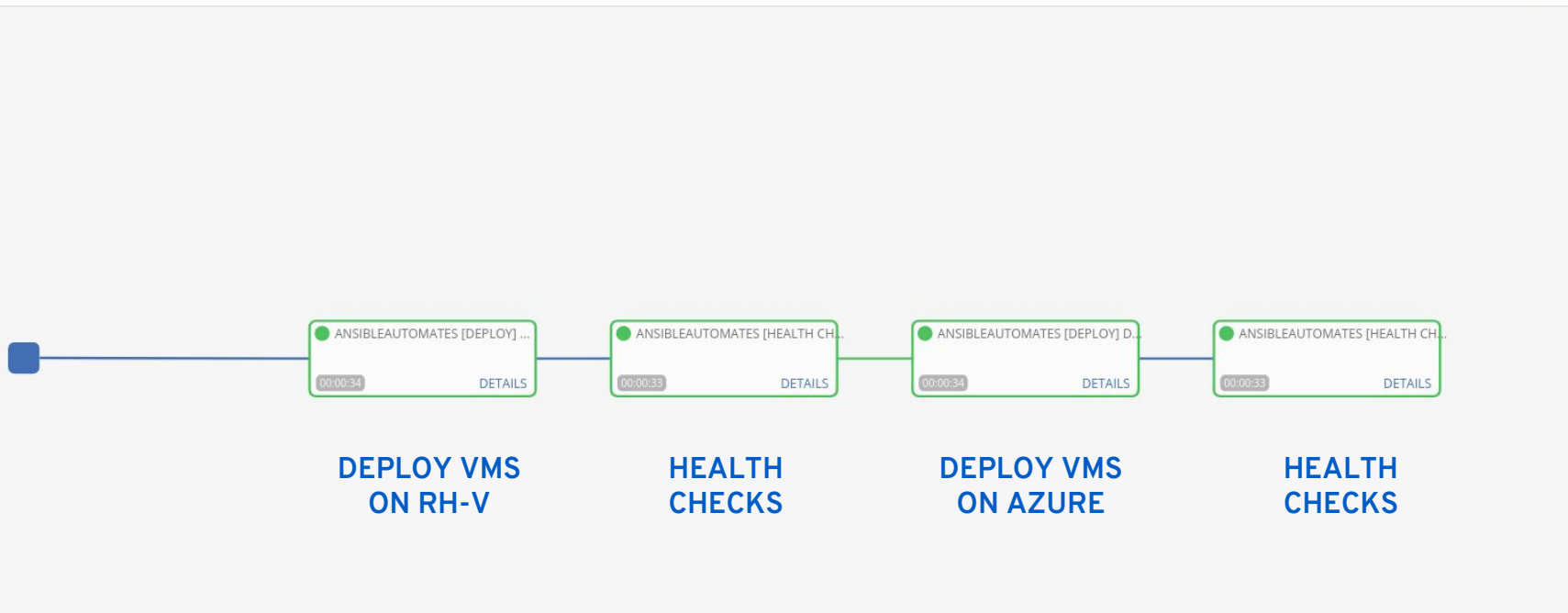
DEMO TIME

SCALE OUT TO AZURE WITH TOWER WORKFLOW

ANSIBLEAUTOMATES [WORKFLOW] Deploy VMs on RH-V + Scale Out on Azure

TOTAL JOBS 4 ELAPSED 00:02:39

KEY: ● On Success ● On Fail ■ Always P Project Sync I Inventory Sync



https://youtu.be/3KT_Xd6uGIA

Ansible Docs (Azure):

https://docs.ansible.com/ansible/latest/scenario_guides/guide_azure.html

Ansible Docs (RH-V modules):

https://docs.ansible.com/ansible/latest/modules/list_of_cloud_modules.html#ovirt

ANSIBLE



AUTOMATION EVERYWHERE



ANSIBLE