

RHEV API / Python SDK Examples

PREPARED FOR - Volkswagen IT Group Cloud

Adrian Bradshaw

Version 0.0.1

Table of Contents

1. History and Revisions	1
2. Preface	2
2.1. Confidentiality, Copyright, and Disclaimer	2
2.2. Additional Background and Related Documents	2
2.3. Terminology	2
3. RHEV API/SDK Documentation	3
3.1. Summary	3
3.2. Background	3
4. Requirements	4
5. Example 1 - VM Inventory	5
6. Example 2 - VM Creation	7
7. Example 3 - Insert CDROM & Boot	10
8. oVirt.py	11
9. Resources	12

1. History and Revisions

Version	Authors	Changes
0.0.1	Adrian Bradshaw adrian@redhat.com	Initial Draft

2. Preface

2.1. Confidentiality, Copyright, and Disclaimer

Copyright 2015 © Red Hat, Inc. All Rights Reserved. No part of the work covered by the copyright herein may be reproduced or used in any form or by any means- graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems without permission in writing from Red Hat except as is required to share this information as provided with the aforementioned confidential parties.

2.2. Additional Background and Related Documents

This document also references additional information that can be found on Red Hat's documentation site at <https://access.redhat.com/knowledge/docs/> and specifically at https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Virtualization/3.5/

Documents specific to products covered in this solution include the following Guides

- [RHEV 3.5 Installation Guide](#)
- [RHEV 3.5 Administration Guide](#)
- [RHEV 3.5 User Guide](#)
- [RHEV 3.5 Technical Guide](#)

Additional information can be found on the RHEV upstream projects website (oVirt):

<http://www.ovirt.org/Documentation>

2.3. Terminology

Some of the acronyms using in this document are included in the table below

Table 1. Terminology Table

Term	Definition
RHEV	Red Hat Enterprise Virtualisation
RHEV-M	Red Hat Enterprise Virtualisation Manager
RHEL-H	Red Hat Enterprise Linux Hypervisor

3. RHEV API/SDK Documentation

3.1. Summary

The purpose of this document is to provide examples of RHEV API calls, to enable VW group Linux admins to be able to add RHEV into their existing environment and be able to manage it in the same way they currently manage other systems such as VMware

3.2. Background

Currently the VW group Linux admins use API calls to their VMware setup to enable fast and efficient VM provisioning. In a nutshell their scripts first create a custom ISO for each VM to be deployed (as there is no PXE available). It then creates the VM inside vSphere (via API calls) adding network card(s) & disk(s). It then inserts the custom ISO into the VMs virtual CDROM drive and starts the VM

They also have a daily script/scripts that, via querying the vSphere, provides a complete inventory of each VM, listing CPUs/Memory/Disks etc

In order to facilitate the admins to adjust their existing scripts to do the same with VMware, this document will list examples of how to achieve similar results using the RHEV API and specifically the python SDK

4. Requirements

In order to use the following scripts you will need, as a minimum, to install the RHEVM SDK

Make sure that the machine has access to the following channel or local replica of it

```
rhel-6-server-rhev-3.5-rpms
```

On the VW network, this is available on the install servers as a repo called

```
rhel6u6-rhev
```

This should be enabled already but the following repo file can be used to access this

```
[rhel6u6-rhev]
name=rhel6u6-rhev
baseurl=http://10.208.1.61/packages/redhat/yum/rhev/rhel6u6/x86_64
enabled=1
gpgcheck=0
```

Then you simply install the package

```
yum install rhvm-sdk-python
```

If you wish to specify a certificate to be used, inside your scripts, then you will first need to get this from the RHEV-M server. The following command will do that

```
wget https://<rhvm-servername>/ca.crt --no-check-certificate
```

5. Example 1 - VM Inventory

This first example shows how to use the python SDK to generate an inventory of the VMs, showing CPU / Memory / Network / Disks connected.

```
#!/usr/bin/python
from ovirtsdk.api import API
from ovirtsdk.xml import params
from time import sleep

def main():
    URL='https://lxf01tpa01.wob.sec.vw.vwg:443/api'
    USERNAME='admin@internal'
    PASSWORD='redhat01'

    api = API(url=URL, username=USERNAME, password=PASSWORD,insecure=True)
    vm_list=api.vms.list()
    for vm in vm_list:
        print "VM Name:\t", vm.name
        print "Status:\t", vm.status.state
        print "Memory:\t", vm.memory
        print "Sockets:\t", vm.cpu.topology.sockets, " Cores:\t", vm.cpu.topology.cores

        #Let's enumerate the NICs
        print "NICs:"
        nic_list=vm.nics.list()
        for nic in nic_list:
            print nic.name
            #if the server is up, lets see what addresses it has on each nic
            address=[]
            if vm.status.state == 'up' and vm.get_guest_info():
                vmnics= vm.get_nics().list()
                ips = vm.get_guest_info().get_ips().get_ip()
                for card in vmnics:
                    print 'MacAddress:%s \t Interface:%s \t Plugged:%s \t Linked:%s ' % (card.mac.address, card.interface, card.plugged,
card.linked)
                    for ip in ips:
                        address.append(ip.get_address())
                        print '\t IP : %s' % ( ip.get_address())

        print "Disks:"
        disks=vm.disks.list()
        for disk in disks:
            #check if it's not a direct LUN
            if disk.interface != 'virtio_scsi':
                print 'Name:%s \t Interface:%s \t\t Actual Size:%s \t Size:%s' % (disk.name, disk.interface, disk.actual_size,
disk.size)
            else:
                luns=disk.lun_storage.get_logical_unit()
                for lun in luns:
                    print 'Name:%s \t Interface:%s \t\t Size:%s ' % (disk.name, disk.interface, lun.size)

        print "\n"
        api.disconnect()

if __name__ == '__main__':
    main()
```

The above script results in output similar to this:

```
...
VM Name:      lxf01tp1003-wob-vw-vwg
Status: up
Memory: 17179869184
Sockets:      4 Cores:      4
NICs:
eth0
MacAddress:00:1a:4a:44:74:01  Interface:virtio    Plugged:True    Linked:True
  IP : 10.186.226.33
  IP : 10.116.96.33
MacAddress:00:1a:4a:44:74:0c  Interface:virtio    Plugged:True    Linked:True
  IP : 10.186.226.33
  IP : 10.116.96.33
eth1
MacAddress:00:1a:4a:44:74:01  Interface:virtio    Plugged:True    Linked:True
  IP : 10.186.226.33
  IP : 10.116.96.33
MacAddress:00:1a:4a:44:74:0c  Interface:virtio    Plugged:True    Linked:True
  IP : 10.186.226.33
  IP : 10.116.96.33
Disks:
Name:lxf02tp1002-wob-vw-vwg_Disk5  Interface:virtio_scsi    Size:162135015424
Name:lxf01tp1003.wob.vw.vwg_Disk1  Interface:virtio         Actual Size:20401094656    Size:144955146240
Name:lxf02tp1002-wob-vw-vwg_Disk2  Interface:virtio_scsi    Size:108447924224
Name:lxf02tp1002-wob-vw-vwg_Disk3  Interface:virtio_scsi    Size:53687091200
Name:lxf02tp1002-wob-vw-vwg_Disk4  Interface:virtio_scsi    Size:162135015424
...
```


6. Example 2 - VM Creation

This second example shows how to use the python SDK to create a VM, add a disk, add a network card and then power up the VM.

Its taken from a github project as can be seen in the comments at the top of the script.

There are two files in this example.

Listing 1. api_ovirt_include.py

```

#!/usr/bin/python

# Created by Jens Depuydt
# http://www.jensd.be
# http://github.com/jensdepuysdt

#this script requires ovirt-engine-sdk-python

from ovirtsdk.api import API
from ovirtsdk.xml import params
from time import sleep

def connectToHost(host,host_user,host_pw):
    apiurl="https://" + host + "/api"
    #insecure -> skips SSL check
    api = API(url=apiurl,username=host_user,password=host_pw,insecure=True)
    return api

def createGuest(api,guest_cluster,guest_name,guest_description,guest_mem,guest_cpu,guest_disk_gb,guest_domain,guest_network):
    cpu_params = params.CPU(topology=params.CpuTopology(cores=guest_cpu))
    try:
        api.vms.add(params.VM(name=guest_name,memory=guest_mem*1024*1024,cluster=api.clusters.get(guest_cluster),template=api.templates.get('Blank'),cpu=cpu_params,type_='server',description=guest_description))

        api.vms.get(guest_name).nics.add(params.NIC(name='nic1', network=params.Network(name=guest_network), interface='virtio'))

        api.vms.get(guest_name).disks.add(params.Disk(storage_domains=params.StorageDomains(storage_domain=[api.storagedomains.get(guest_domain)]),size=guest_disk_gb*1024*1024*1024,status=None,interface='virtio',format='cow',sparse=True,bootable=True))
        while api.vms.get(guest_name).status.state != 'down':
            sleep(1)

    except Exception as e:
        print 'Failed to create VM with disk and NIC\n%s' % str(e)

    disk_name=guest_name+"_Disk1"
    print "Waiting for "+disk_name+" to reach ok status"
    while api.vms.get(guest_name).disks.get(name=disk_name).status.state != 'ok':
        sleep(1)

    return "Successfully created guest: "+guest_name

def getMac(api,guest_name):
    return api.vms.get(guest_name).nics.get("nic1").mac.address

def powerOnGuest(api,guest_name):
    try:
        if api.vms.get(guest_name).status.state != 'up':
            print 'Starting VM'
            api.vms.get(guest_name).start()
            print 'Waiting for VM to reach Up status'
            while api.vms.get(guest_name).status.state != 'up':
                sleep(1)
        else:
            print 'VM already up'
    except Exception as e:
        print 'Failed to Start VM:\n%s' % str(e)

```

Listing 2. ovirt_create-vm.py

```
#!/usr/bin/python
import api_ovirt_include

def main():
    #connection properties
    #change these to match your installation
    host="<rhev-servername>"
    host_user="admin@internal"
    host_pw="<password>"

    #properties of the new VM:
    guest_name="pleasedeleteme"    #name of the VM
    guest_description="testvm"    #name of the VM
    guest_mem=1024                #memory in MB
    guest_cpu=1                   #number of virtual CPU
    guest_space=2                  #space in GB
    storage_domain="GlusterStore" #name of the storage domain
    guest_cluster="Default"       #cluster name
    guest_network="ovirtmgmt"     #network-name

    #connect to the host
    host_con=api_ovirt_include.connectToHost(host,host_user,host_pw)

    #create the new VM
    res=api_ovirt_include.createGuest(host_con,guest_cluster,guest_name,guest_description,guest_mem,guest_cpu,guest_space,storage_domain
    ,guest_network)
    print "Result:",res

    #start the new VM
    api_ovirt_include.powerOnGuest(host_con,guest_name)

    #disconnect from host
    host_con.disconnect()

if __name__ == '__main__':
    main()
```

7. Example 3 - Insert CDROM & Boot

This final example demonstrates how to specify a CDROM to boot from once and power on the VM:

```
#!/usr/bin/python
from ovirtsdk.api import API
from ovirtsdk.xml import params
from time import sleep

def main():
    URL='https://destination:443/api'
    USERNAME='admin@internal'
    PASSWORD='blah'

    api = API(url=URL, username=USERNAME, password=PASSWORD, insecure=True)

    vm = api.vms.get(name="pleasedelete")
    vm.start(
        action=params.Action(
            vm=params.VM(
                cdroms=params.CdRoms(
                    cdrom=[
                        params.CdRom(
                            file=params.File(
                                id="rhel-server-7.1-x86_64-dvd.iso"
                            )
                        )
                    ]
                ),
                os=params.OperatingSystem(
                    boot=[
                        params.Boot(
                            dev="cdrom"
                        )
                    ]
                )
            )
        )
    )

if __name__ == '__main__':
    main()
```

8. oVirt.py

There is a very detailed python script called ovirt.py that was developed by a Red Hat guy, its available from github at the address below. Be aware that there are a few additional requirements for this to work but it's just two RPMs

```
https://github.com/karmab/ovirt
```

Here are some examples, from the **readme.md**, of what it can do

CREATE VIRTUAL MACHINE V0 100 BASED ON PROFILE BE6 FOR CLIENT BIGCOMPANY, PROVIDING IPS FOR COBBLER TOO

```
ovirt.py -ZC bigcompany -n v0100 -p be6 -1 192.168.1.100 -2 192.168.10.100
```

DELETE VIRTUAL MACHINE V0100 FROM CLIENT NYSE

```
ovirt.py -C nyse -K v0100
```

GET A CONSOLE FOR VIRTUAL MACHINE HENDRIX (IN DEFAULT CLIENT)

```
ovirt.py -o hendrix
```

START A VM KIPA02 DIRECTLY THROUGH THE HYPERVISOR WITH

```
hypervisor.py -T ~/vdsm_certs -H 192.168.6.1 -s kipa02
```

CREATE THE MACHINE DIRECTLY IN FOREMAN (specifying foremanip, name and dns of the vm, the hostgroup and compute resource to use,and build mode)

```
foreman.py -H 192.168.8.223 -n satriani -d xxx.org -X base6 -b -c bumblefoot
```

To get started using this script, simply clone the repo

```
git clone https://github.com/karmab/ovirt.git
```

9. Resources

Listed below are some resources that you should find helpful

First off some official documentation from Red Hat

https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Virtualization/3.2/html/Developer_Guide/chap-Python_Quick_Start_Example.html

There is also some useful docs upstream

<http://www.ovirt.org/Python-sdk> <http://www.ovirt.org/Testing/PythonApi>

Internal Red Hat guys github

<https://github.com/karmab/ovirt>

3rd Party Resources

Nice pythons sdk docs here

<http://www.kernel-panic.it/linux/rhev/> <https://website-humblec.rhcloud.com/ovirt-list-vms-ip-addresses-nic-details-etc-using-python-sdk-part-4>

cURL based API examples

https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Virtualization/3.5/html-single/Technical_Guide/index.html#chap-REST_API_Quick_Start_Example