목차

Host USB Passthrough to RHEV Guests

RHEV 3.6 이후 버전

3.6 이후 버전부터는 정식적으로 지원한다. 아래 링크 참조

- https://access.redhat.com/documentation/en-us/red_hat_enterprise_virtualization/3.6/html/virtual_machine_management_guide/sect-host_devices

RHEV 3.2

이 솔루션은 RHEL 기반 호스트에만 적용됩니다. (Baremetal 방식 지원 안됨) 그리고 VM을 해당호스트에 고정해야 합니다. (마이그레이션 불가)

- RHEL 호스트에서 rhev의 게스트로 USB 패스 스루를 수행하려면 다음 단계를 수행해야합니다.

USB 스틱을 RHEL 호스트에 넣고 `lsusb` 명령을 수집하여 `vendor:product id` 를 얻습니다 .

```
# lsusb
Bus 002 Device 008: ID 0781:5567 SanDisk Corp. Cruzer Blade
```

hostusb 링크를 방문 하여 rhel 호스트에 `before_vm_start.py` 파일을 가져 와서 `/usr/libexec/vdsm/hooks/before_vm_start/` 위치에 `60_hostusb` 이름으로 755 권한을 주어 넣으십시오.

before_vm_start.py

```
#!/usr/bin/python

import re
import os
import sys
import grp
import pwd
import traceback

import hooking

'''
host usb hook
=============
```

```
          !!! Disclaimer !!!
*****************************************
The host side usb support wasn't thoroughly
tests in kvm!
*****************************************

add hosts usb device/s to VM:

<hostdev mode='subsystem' type='usb'>
    <source>
        <vendor id='0x1234'/>
        <product id='0xbeef'/>
    </source>
</hostdev>

syntax:
    hostusb=0x1234:0xbeef&0x2222:0xabaa
    i.e.
    hostusb=VendorId:ProductId (can add more then one with '&' separator)

Note:
    The VM must be pinned to host and this hook will
    fail any migration attempt.
'''

HOOK_HOSTUSB_PATH = '/var/run/vdsm/hooks/hostusb-permissions'

def log_dev_owner(devpath, user, group):
    entry = devpath + ":" + str(user) + ":" + str(group)

    if not os.path.isdir(os.path.dirname(HOOK_HOSTUSB_PATH)):
        os.mkdir(os.path.dirname(HOOK_HOSTUSB_PATH))

    if os.path.isfile(HOOK_HOSTUSB_PATH):
        f = file(HOOK_HOSTUSB_PATH, 'r')
        for line in f:
            if entry == line:
                f.close()
                return

    f = file(HOOK_HOSTUSB_PATH, 'a')
    f.writelines(entry)
    f.close()

#!TODO:
# merge chown with after_vm_destroy.py
# maybe put it in hooks.py?
def chown(vendorid, productid):

    # remove the 0x from the vendor and product id
```

```
    devid = vendorid[2:] + ':' + productid[2:]
    command = ['lsusb', '-d', devid]
    retcode, out, err = hooking.execCmd(command, sudo=False, raw=True)
    if retcode != 0:
        sys.stderr.write('hostusb: cannot find usb device: %s\n' % devid)
        sys.exit(2)

    # find the device path:
    # /dev/bus/usb/xxx/xxx
    devpath = '/dev/bus/usb/' + out[4:7] + '/' + out[15:18]
    stat = os.stat(devpath)

    group = grp.getgrnam('qemu')
    gid = group.gr_gid
    user = pwd.getpwnam('qemu')
    uid = user.pw_uid

    # we don't use os.chown because we need sudo
    owner = str(uid) + ':' + str(gid)
    command = ['/bin/chown', owner, devpath]
    retcode, out, err = hooking.execCmd(command, sudo=True, raw=True)
    if retcode != 0:
        sys.stderr.write('hostusb: error chown %s to %s, err = %s\n' %
(devpath, owner, err))
        sys.exit(2)

    log_dev_owner(devpath, stat.st_uid, stat.st_gid)

def create_usb_device(domxml, vendorid, productid):
    hostdev = domxml.createElement('hostdev')
    hostdev.setAttribute('mode', 'subsystem')
    hostdev.setAttribute('type', 'usb')

    source = domxml.createElement('source')
    hostdev.appendChild(source)

    vendor = domxml.createElement('vendor')
    vendor.setAttribute('id', vendorid)
    source.appendChild(vendor)

    product = domxml.createElement('product')
    product.setAttribute('id', productid)
    source.appendChild(product)

    return hostdev

if os.environ.has_key('hostusb'):
    try:
        regex = re.compile('^0x[\d,A-F,a-f]{4}$')
        domxml = hooking.read_domxml()
        devices = domxml.getElementsByTagName('devices')[0]
```

```
        for usb in os.environ['hostusb'].split('&'):
            vendorid, productid = usb.split(':')
            if len(regex.findall(vendorid)) != 1 or
len(regex.findall(productid)) != 1:
                sys.stderr.write('hostusb: bad input, expected 0x0000 format
for vendor and product id, input: %s:%s\n' % (vendorid, productid))
                sys.exit(2)

            hostdev = create_usb_device(domxml, vendorid, productid)
            devices.appendChild(hostdev)
            chown(vendorid, productid)

        hooking.write_domxml(domxml)
    except:
        sys.stderr.write('hostusb: [unexpected error]: %s\n' %
traceback.format_exc())
        sys.exit(2)
```

vdsm 후크에 대한 자세한 정보는 VDSM_Hooks 링크를 참조하십시오 .

- rhev 관리자 시스템에서 아래 명령을 실행하여 필요한 변경을 수행하십시오.

아래 명령을 사용하여 hostusb를 활성화하십시오.

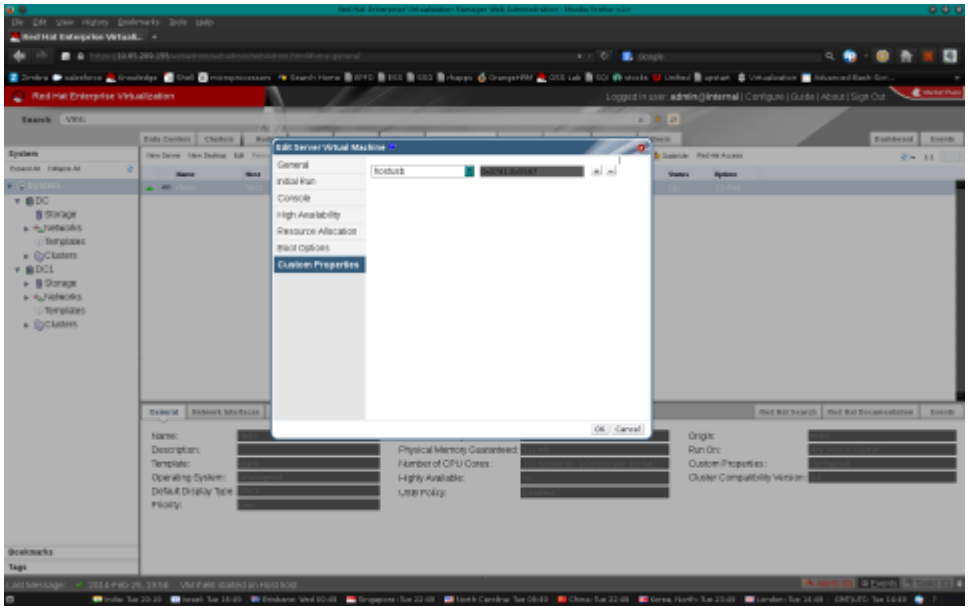
```
# rhevm-config -s UserDefinedVMProperties='hostusb=[\w:&]+' --cver=3.2
# /etc/init.d/ovirt-engine restart
```

아래 명령을 사용하여 변경 사항을 확인할 수 있습니다.

```
# rhevm-config -g UserDefinedVMProperties
```

그런 다음 rhevm GUI로 이동하여 게스트를 종료하고 '편집'버튼을 선택하여 Custom Properties탭 에서 필요한 수정을 수행 하십시오.

1. hostusb 를 선택하십시오
2. 속성으로 0x0781 : 0x5567 을 사용 하고 확인을 클릭하십시오.



참고 : 공급 업체 ID(0781), 제품 ID(5567) 는 `lsusb` 명령으로 수집한 정보입니다.

이제 VM을 시작하여 매핑 된 USB 스틱을 사용하십시오.

중요 : 기존 사용자 정의 후크가 있으면 `rhevm-config -s`를 사용하여 `hostusb`를 설정하는 동안 알려줘야합니다. 그렇지 않으면 이전 항목이 지워지고 새 `hostusb` 만 추가됩니다.

```
# rhevm-config -g UserDefinedVMProperties
UserDefinedVMProperties: copypaste=^(yes|no)$ version: 3.2

# rhevm-config -s
UserDefinedVMProperties='copypaste=^(yes|no)$;hostusb=[\w:&]+' --cver=3.2

# rhevm-config -g UserDefinedVMProperties
UserDefinedVMProperties: copypaste=^(yes|no)$;hostusb=[\w:&]+ version: 3.2
```

참조링크

- https://access.redhat.com/solutions/146793
- https://access.redhat.com/solutions/474373
- https://access.redhat.com/solutions/734323