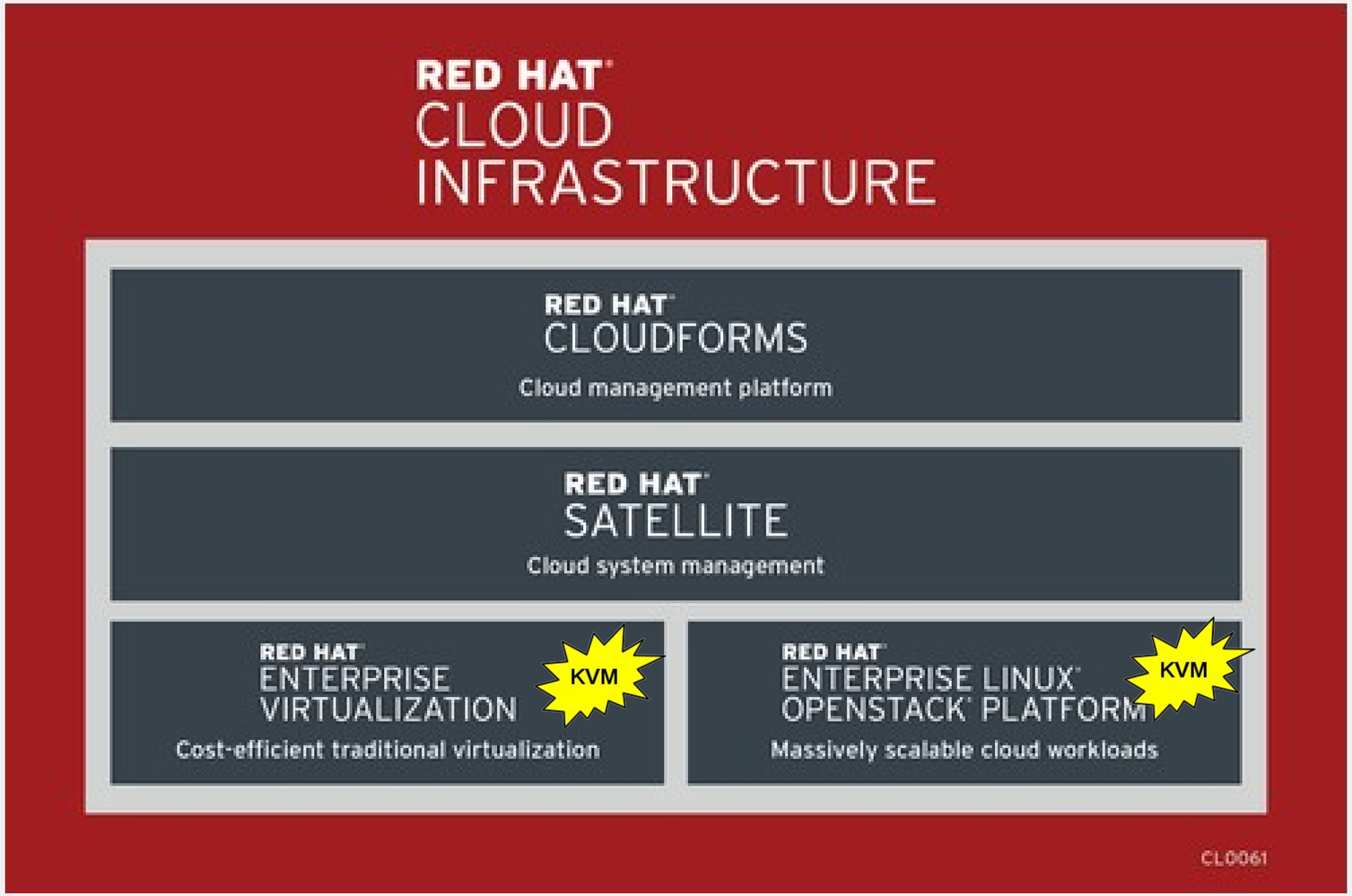
Features & Futures: Red Hat Enterprise Virtualization Hypervisor (KVM)

Karen Noel – Senior Software Engineering Manager
Andrea Arcangeli – Sr. Principal Software Engineer
June 2016

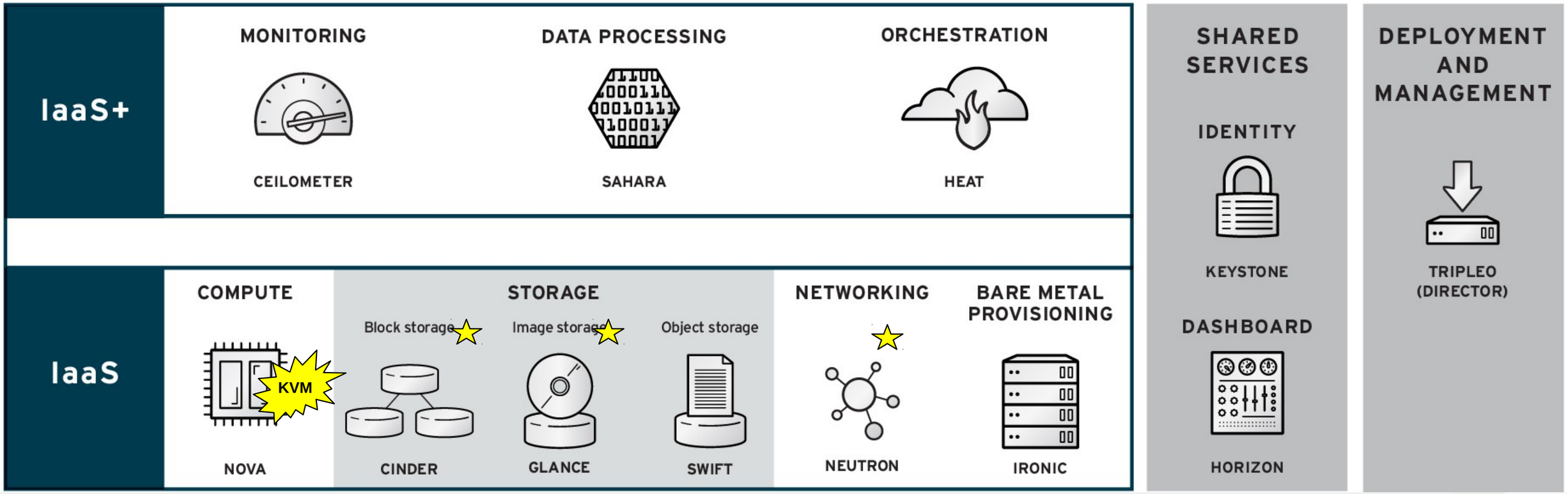
Features & Futures:

- Red Hat's KVM Hypervisor
- Virtual CPU and memory hot-plug
- Real-time KVM
- Post-copy live migration
- More futures...

Red Hat Virtualization - KVM



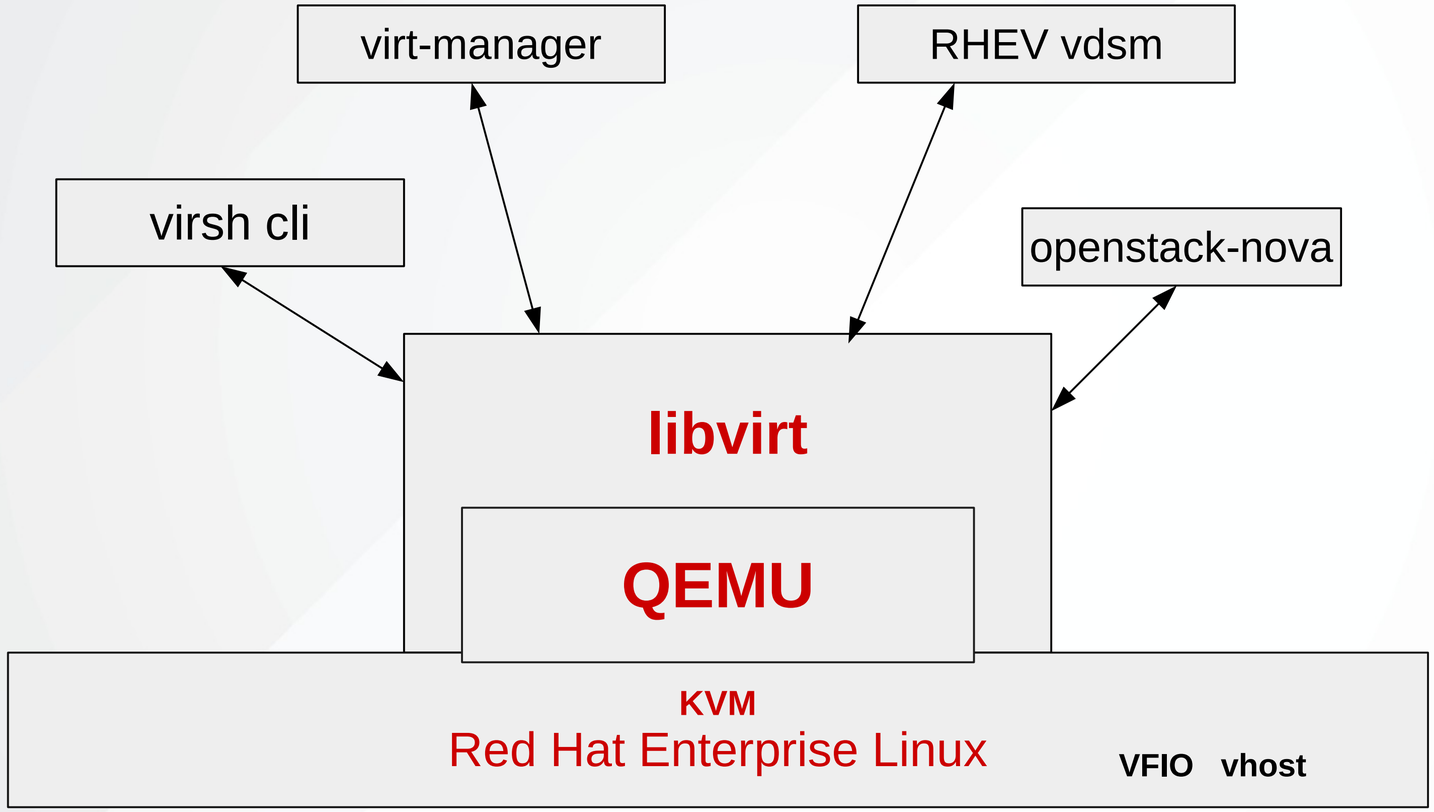
Cloud Infrastructure for Cloud Workloads



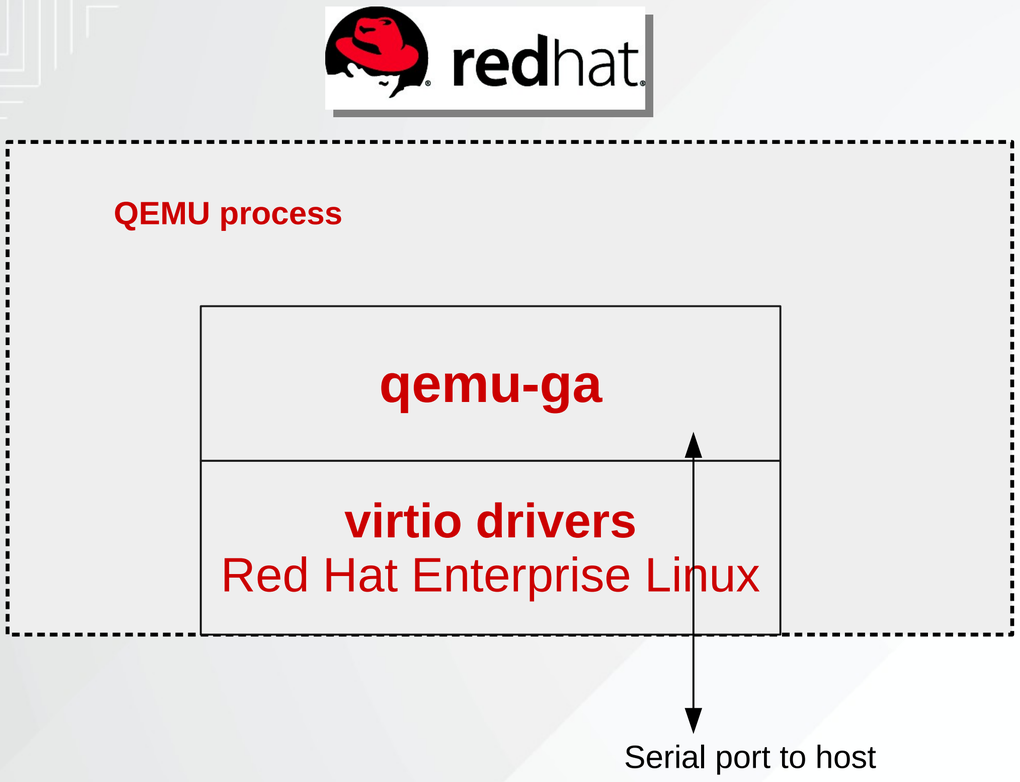
RHEV and Red Hat OpenStack Platform

TRADITIONAL: SCALE UP (RHEV)	CLOUD: SCALE OUT (OpenStack)	MIXED/HYBRID
Big stateful VM	Small stateless VMs	Combination of traditional scale-up and cloud scale-out workloads.
1 Application → 1 VM	1 Application → Many VMs	For example: Database may be hosted on traditional workloads, web front-end and logic layers on cloud workloads.
Lifecycle in years	Lifecycle hours to months	
Increased user demand = Scale up (VM gets bigger)	Increased user demand = Scale out (add VMs)	
Not designed to tolerate failure of VM, so you need features that keep VMs up	If a VM dies, application kills it and creates a new one, app stays up	
Application SLA requires enterprise virtualization features (migration, HA, etc.) to keep applications available	Application SLA requires adding/removing VM instances to application cloud to maintain application availability	

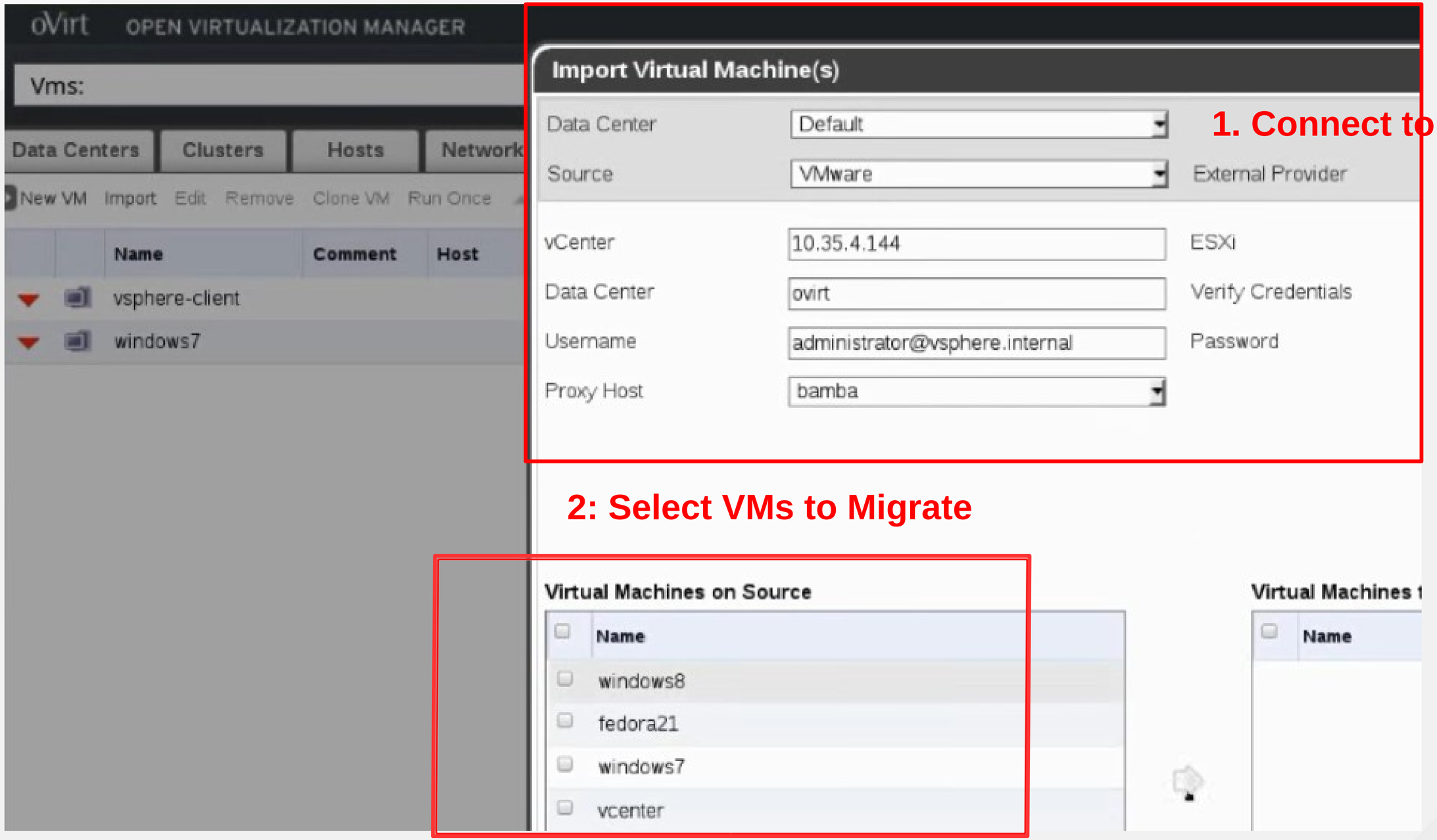
KVM Virtualization Stack - host



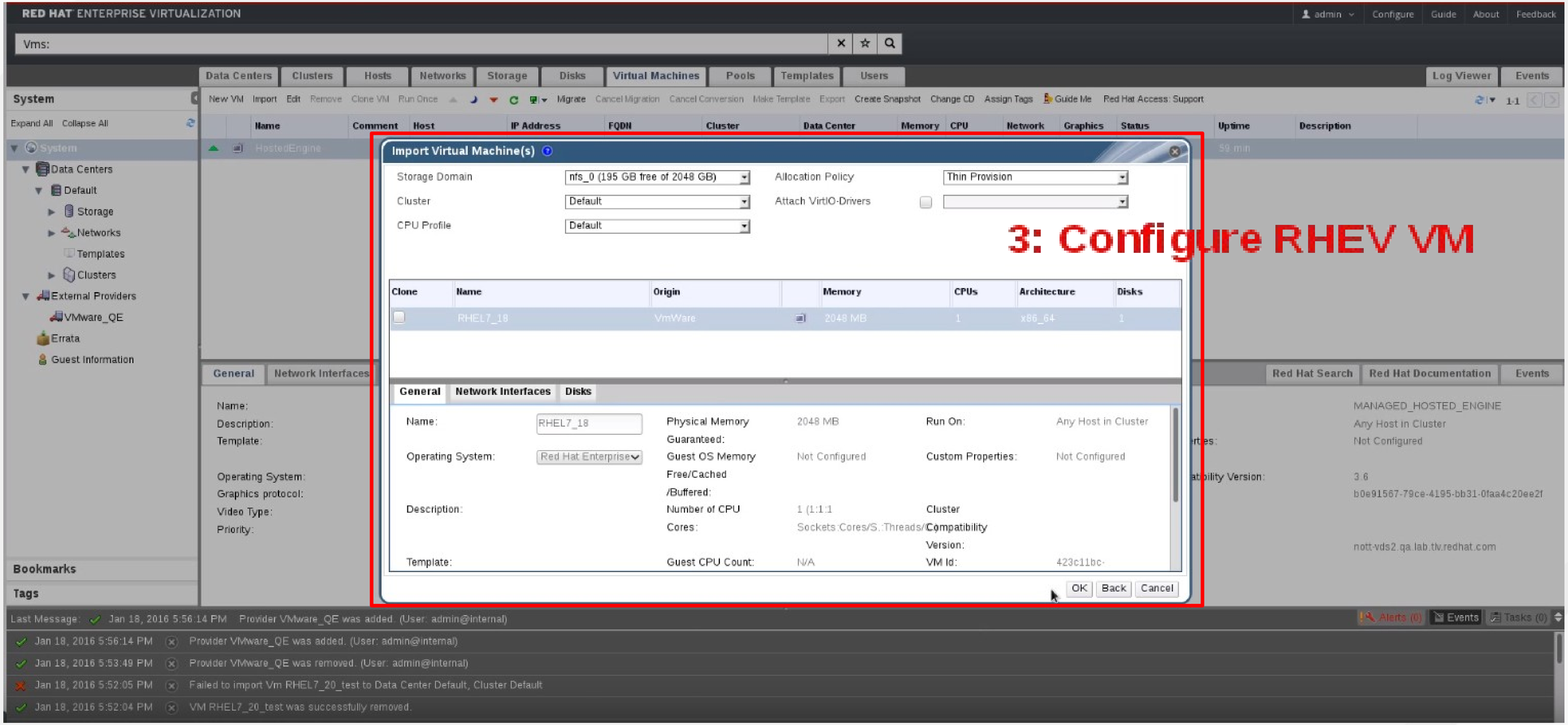
KVM Virtualization Stack - guest

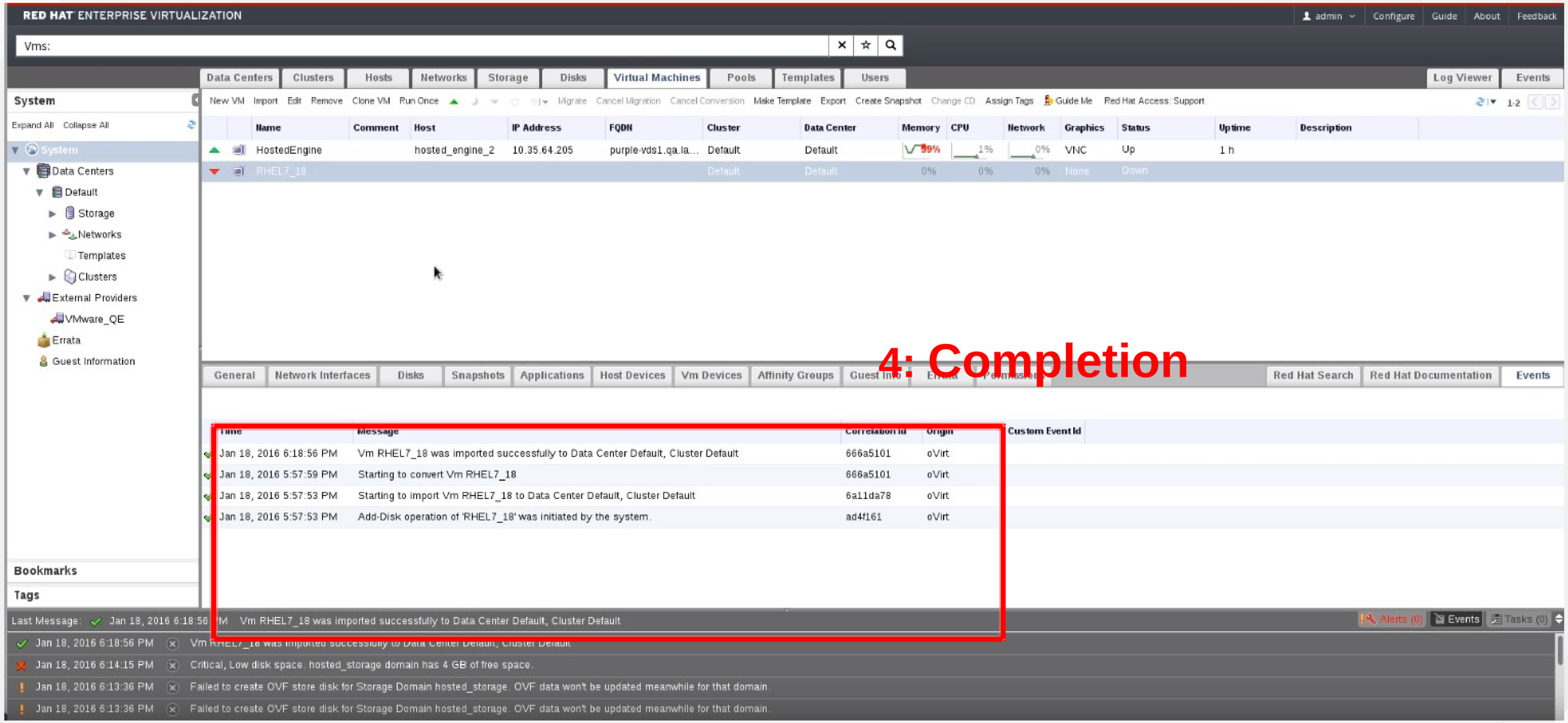


Who is running RHEL?
RHEV?
OpenStack?
VMware?

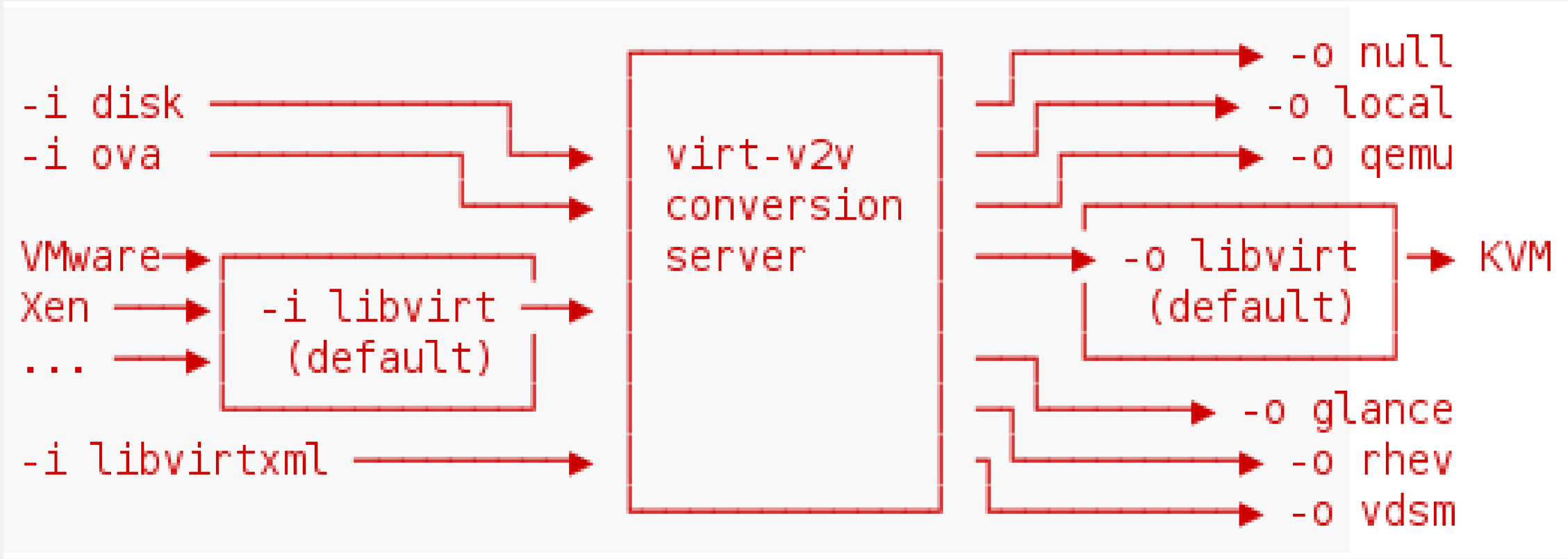


https://www.youtube.com/watch?v=yEjckA-FriU


RED HAT ENTERPRISE VIRTUALIZATION
admin
Configure
Guide
About
Feedback
Vms:
Data Centers
Clusters
Hosts
Networks
Storage
Disks
Virtual Machines
Pools
Templates
Users
Log Viewer
Events
System
New VM
Import
Edit
Remove
Clone VM
Run Once
Migrate
Cancel Migration
Cancel Conversion
Make Template
Export
Create Snapshot
Change CD
Assign Tags
Guide Me
Red Hat Access: Support
Expand All
Collapse All
Name
Comment
Host
IP Address
FQDN
Cluster
Data Center
Memory
CPU
Network
Graphics
Status
Uptime
Description
HostedEngine
59 min
System
Data Centers
Default
Storage
Networks
Templates
Clusters
External Providers
VMware_QE
Errata
Guest Information
Import Virtual Machine(s)
Storage Domain
nfs_0 (195 GB free of 2048 GB)
Allocation Policy
Thin Provision
Cluster
Default
Attach VirtIO-Drivers
CPU Profile
Default
3: Configure RHEV VM
Clone
Name
Origin
Memory
CPUs
Architecture
Disks
RHEL7_18
VmWare
2048 MB
1
x86_64
1
General
Network Interfaces
Disks
Name:
RHEL7_18
Physical Memory Guaranteed:
2048 MB
Run On:
Any Host in Cluster
Operating System:
Red Hat Enterprise
Guest OS Memory Free/Cached /Buffered:
Not Configured
Custom Properties:
Not Configured
Description:
Number of CPU Cores:
1 (1:1:1 Sockets:Cores/S.:Threads/C.)
Cluster Compatibility Version:
Template:
Guest CPU Count:
N/A
VM Id:
423c11bc-
OK
Back
Cancel
Red Hat Search
Red Hat Documentation
Events
MANAGED_HOSTED_ENGINE
Any Host in Cluster
Not Configured
3.6
b0e91567-79ce-4195-bb31-0faa4c20ee2f
nott-vds2.qa.lab.tlv.redhat.com
Bookmarks
Tags
Last Message:
Jan 18, 2016 5:56:14 PM
Provider VMware_QE was added. (User: admin@internal)
Alerts (0)
Events
Tasks (0)
Jan 18, 2016 5:56:14 PM
Provider VMware_QE was added. (User: admin@internal)
Jan 18, 2016 5:53:49 PM
Provider VMware_QE was removed. (User: admin@internal)
Jan 18, 2016 5:52:05 PM
Failed to import Vm RHEL7_20_test to Data Center Default, Cluster Default
Jan 18, 2016 5:52:04 PM
VM RHEL7_20_test was successfully removed.



4: Completion



http://libguestfs.org/virt-v2v.1.html#convert-from-vmware-vcenter-server-to-local-libvirt

From VMware to KVM...

RHEV:

```
# virt-v2v -ic vpx://vcenter.example.com/Datacenter/esxi vmware_guest \
   -o rhev -os rhev.nfs:/export_domain --network rhevm
```

RHOSP - Glance:

```
# virt-v2v -i disk disk.img -o glance
```

RHEL:

```
# virt-v2v -ic vpx://vcenter.example.com/Datacenter/esxi vmware_guest
```

How does V2V work? Windows Guest

- Check for Group Policy Objects → WARNING!
- Check for Anti-Virus → WARNING!
- Insert RHEV guest agent - add firstboot script to install
- Disable Windows services, intelppm.sys, processor.sys
- Disable autoreboot... just in case
- Upload virtio drivers
  - Modify HKLM\SOFTWARE registry, locate virtio-blk at boot
  - Other drivers – use PCI/driver discovery

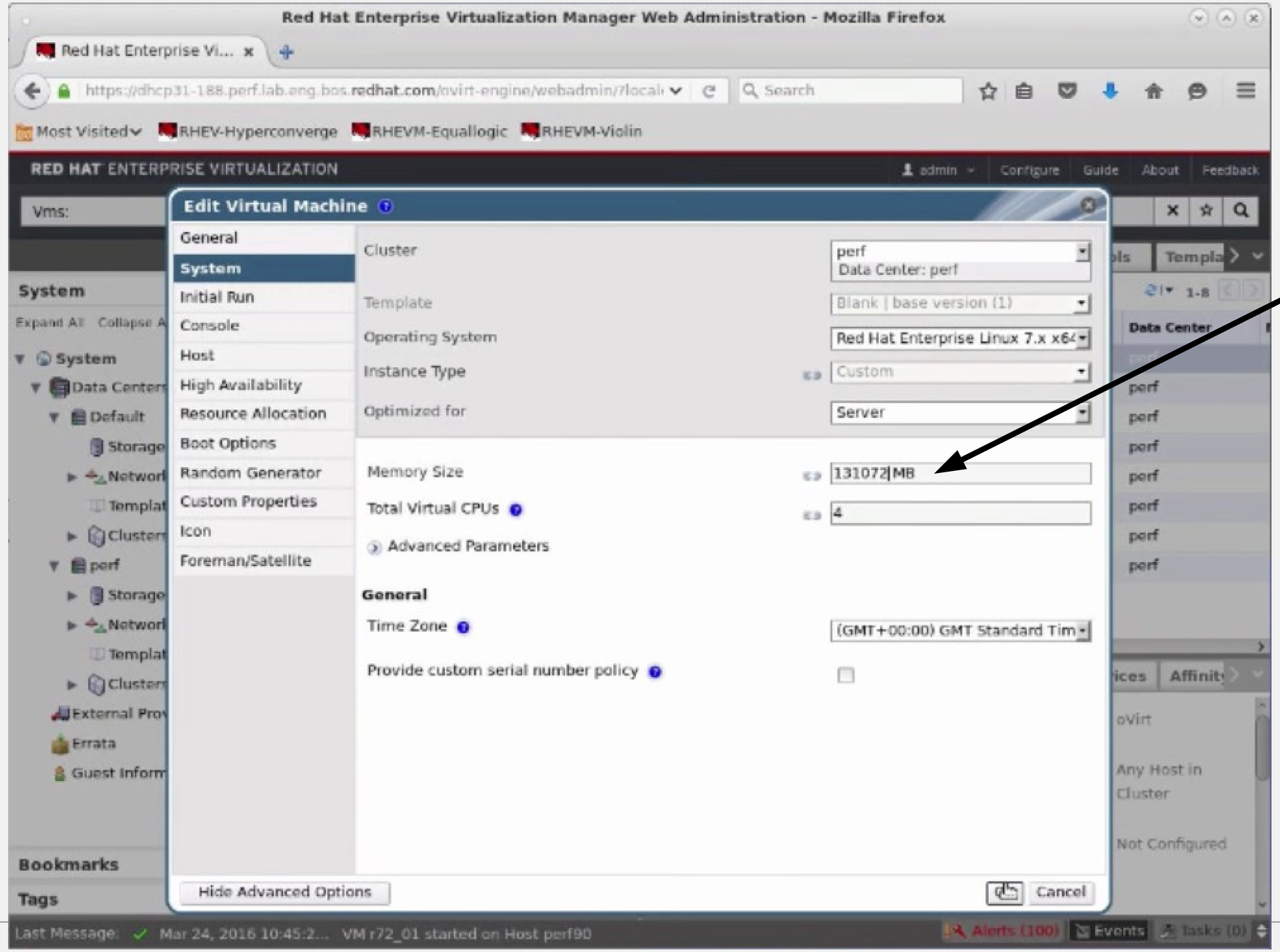
Windows Guest – v2v support

- RHEL 7.3 adds Win8+
  - Tech preview? Supported?
  - New Windows driver installation model
  - Being very cautious, needs lots of testing ←volunteer!

How does V2V work? RHEL Guest

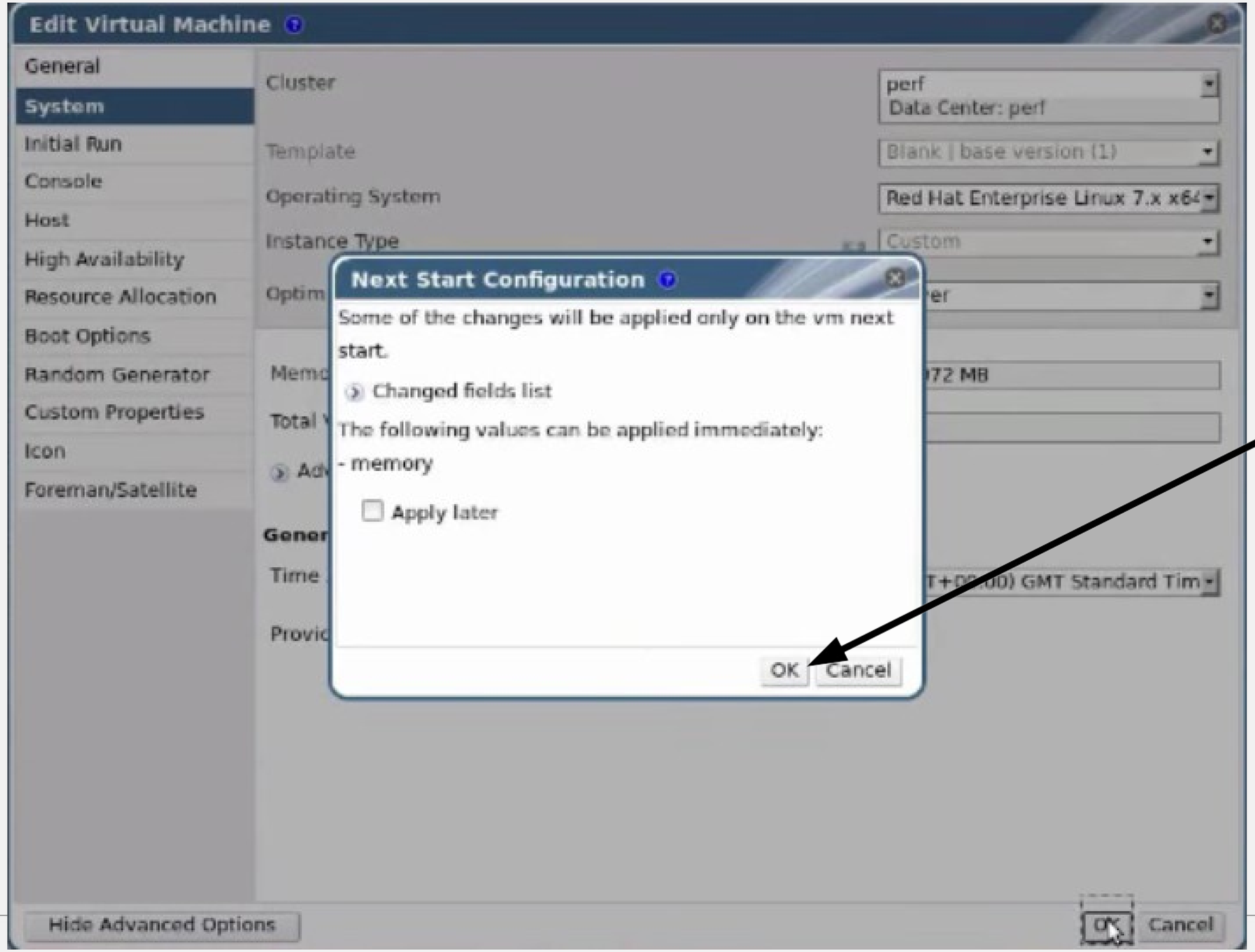
- Clean RPM database
- Check kernels available: virtio supported? Which drivers?
- Touch /.autorelabel – SELinux will relabel file system on next boot
- /etc/X11/xorg.conf – change to use QXL or Cirrus
- Rebuild initrd – so virtio drivers are available

Virtual CPU and memory hot-plug



Memory size increase

http://captainkvm.com/2016/04/hot-adding-memory-rhev/



OK
Apply immediately

http://captainkvm.com/2016/04/hot-adding-memory-rhev/

Memory hot-plug

- Configure guest with:
  - maxMemory > currentMemory
  - NUMA topology
    - node0 is enough

```
<maxMemory slots='16' unit='KiB'>4194304</maxMemory>

<cpu>
  <numa>
     <cell id='0' cpus='0-3' memory='524288' unit='KiB'/>
  </numa>
</cpu>
```

Memory hot-plug

- Prepare a device memory.xml to attach with:

```
# virsh attach-device memory.xml
```

```
<memory model='dimm'>
  <target>
    <size unit='KiB'>524288</size>
    <node>0</node>
  </target>
</memory>
```

Memory hot-plug

- Memory unplug not supported, yet*
- Use balloon to adjust guest memory down/up

* Currently targeted for a future release of RHEV

Virtual CPU hot-plug

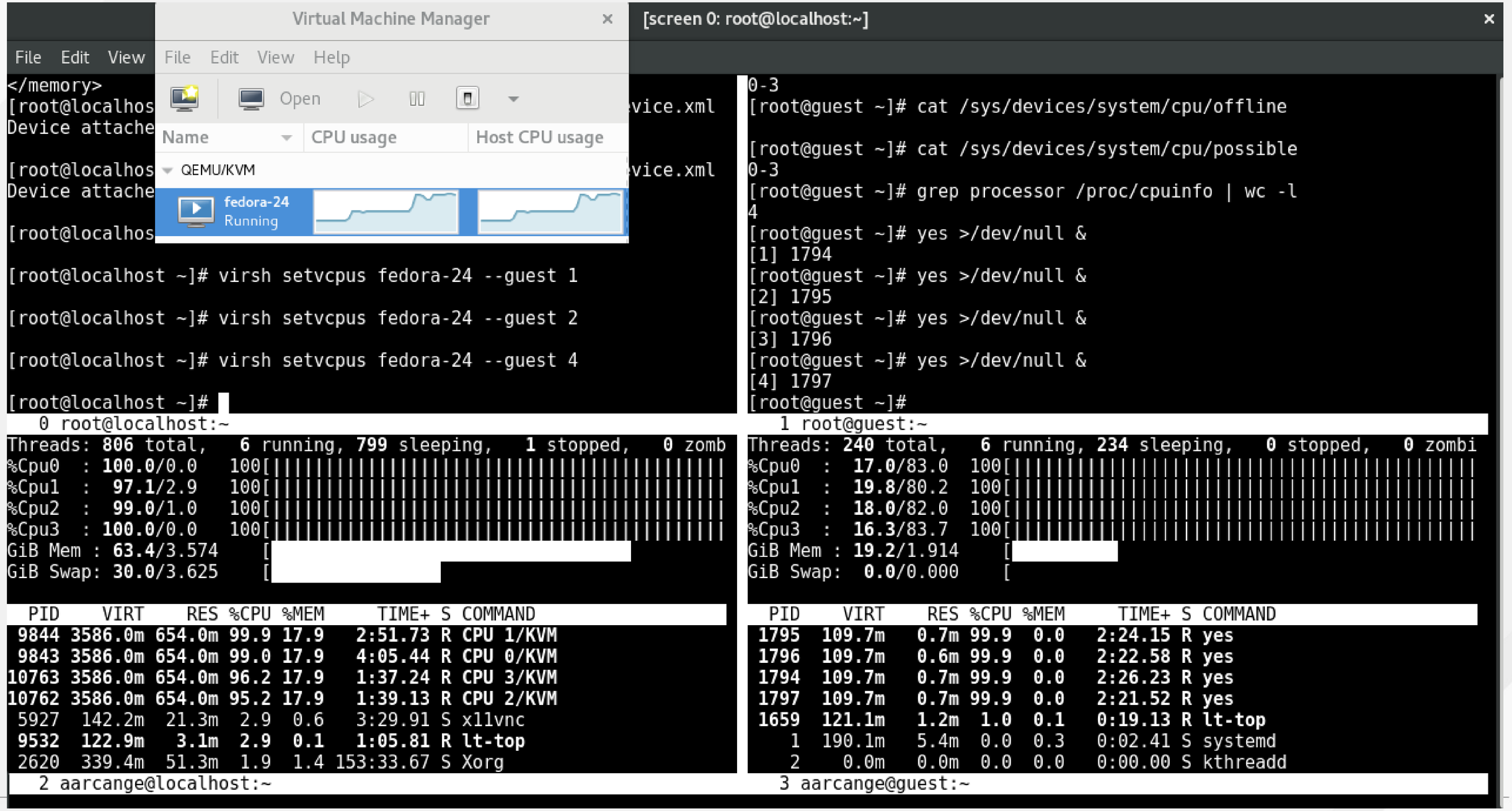
- Configure guest with max vcpus > current vcpus

```
<domain type='kvm'>
  <vcpu placement='static' current='2'>16</vcpu>
```

Virtual CPU hot-plug

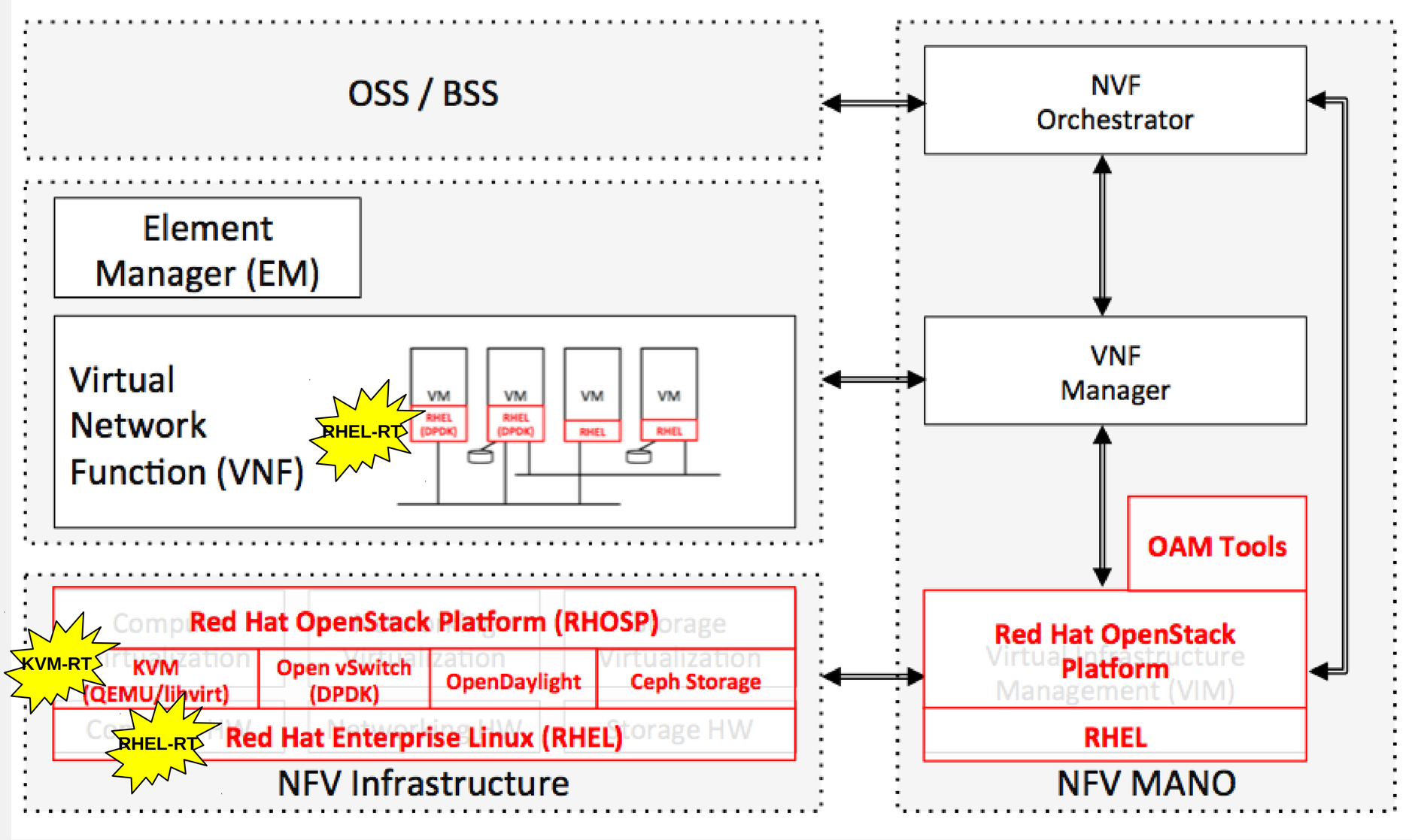
- Specify total number of vcpus for guest
  - Unplug is not supported yet*
- RHEL: udev rule brings cpus online
- Fedora: use –guest or add udev rule
  - Configure QEMU guest agent

```
# virsh setvcpus fedora-24  4
# virsh setvcpus fedora-24  4  --guest
```


Virtual Machine Manager
File Edit View Help
Open
Name
CPU usage
Host CPU usage
QEMU/KVM
fedora-24
Running
[screen 0: root@localhost:~]
File Edit View
</memory>
[root@localhos
Device attache
[root@localhos
Device attache
[root@localhos
[root@localhost ~]# virsh setvcpus fedora-24 --guest 1
[root@localhost ~]# virsh setvcpus fedora-24 --guest 2
[root@localhost ~]# virsh setvcpus fedora-24 --guest 4
[root@localhost ~]#
0 root@localhost:~
Threads: 806 total, 6 running, 799 sleeping, 1 stopped, 0 zomb
%Cpu0 : 100.0/0.0
%Cpu1 : 97.1/2.9
%Cpu2 : 99.0/1.0
%Cpu3 : 100.0/0.0
GiB Mem : 63.4/3.574
GiB Swap: 30.0/3.625
PID VIRT RES %CPU %MEM TIME+ S COMMAND
9844 3586.0m 654.0m 99.9 17.9 2:51.73 R CPU 1/KVM
9843 3586.0m 654.0m 99.0 17.9 4:05.44 R CPU 0/KVM
10763 3586.0m 654.0m 96.2 17.9 1:37.24 R CPU 3/KVM
10762 3586.0m 654.0m 95.2 17.9 1:39.13 R CPU 2/KVM
5927 142.2m 21.3m 2.9 0.6 3:29.91 S x11vnc
9532 122.9m 3.1m 2.9 0.1 1:05.81 R lt-top
2620 339.4m 51.3m 1.9 1.4 153:33.67 S Xorg
2 aarcange@localhost:~
0-3
[root@guest ~]# cat /sys/devices/system/cpu/offline
[root@guest ~]# cat /sys/devices/system/cpu/possible
0-3
[root@guest ~]# grep processor /proc/cpuinfo | wc -l
4
[root@guest ~]# yes >/dev/null &
[1] 1794
[root@guest ~]# yes >/dev/null &
[2] 1795
[root@guest ~]# yes >/dev/null &
[3] 1796
[root@guest ~]# yes >/dev/null &
[4] 1797
[root@guest ~]#
1 root@guest:~
Threads: 240 total, 6 running, 234 sleeping, 0 stopped, 0 zombi
%Cpu0 : 17.0/83.0
%Cpu1 : 19.8/80.2
%Cpu2 : 18.0/82.0
%Cpu3 : 16.3/83.7
GiB Mem : 19.2/1.914
GiB Swap: 0.0/0.000
PID VIRT RES %CPU %MEM TIME+ S COMMAND
1795 109.7m 0.7m 99.9 0.0 2:24.15 R yes
1796 109.7m 0.6m 99.9 0.0 2:22.58 R yes
1794 109.7m 0.7m 99.9 0.0 2:26.23 R yes
1797 109.7m 0.7m 99.9 0.0 2:21.52 R yes
1659 121.1m 1.2m 1.0 0.1 0:19.13 R lt-top
1 190.1m 5.4m 0.0 0.3 0:02.41 S systemd
2 0.0m 0.0m 0.0 0.0 0:00.00 S kthreadd
3 aarcange@guest:~

RHEL for Real-Time with KVM

Red Hat OpenStack Platform 8*



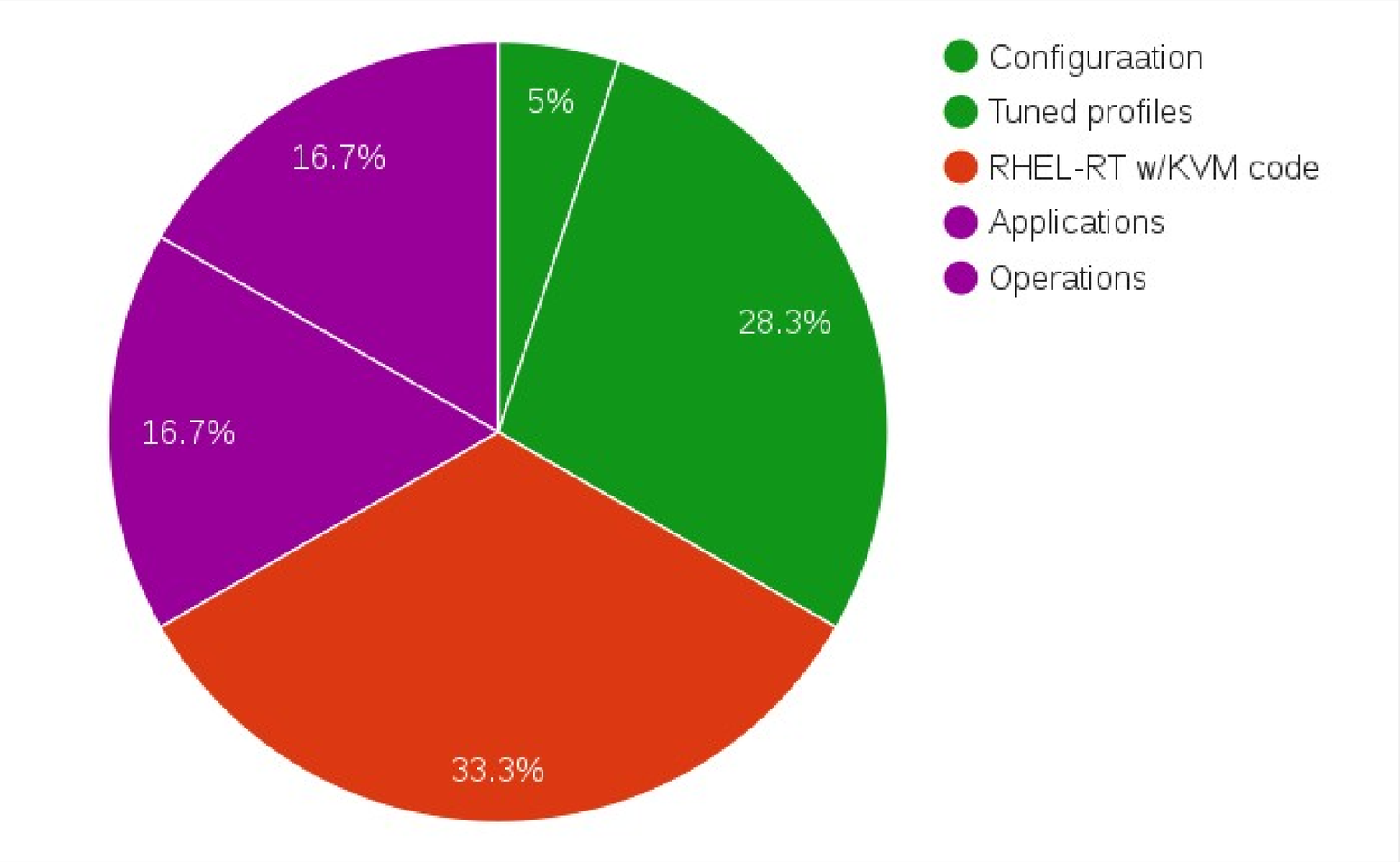
* KVM-RT is Tech Preview in RHOSP 8

Real-time KVM

- 1/3 Code
- 1/3 Tuning & Config
- 1/3 Apps & Ops

Collaboration:

- Open source
- HW partners
- NFV partners



KVM-RT - code

- 56% Red Hat contribution as of June 2015 (48 of 86)
- 58% Red Hat contribution as of June 2016 (63 of 109)
- Kernel: mm, core, vmstat, sched, memcontrol, workqueue, timer, cpusets/isolcpus, lib/vsprintf, rcu/nohz/kvm, tracing, kvm/x86
- Ftrace: x86
- RT kernel: rt/kvm
- Libvirt: qemu

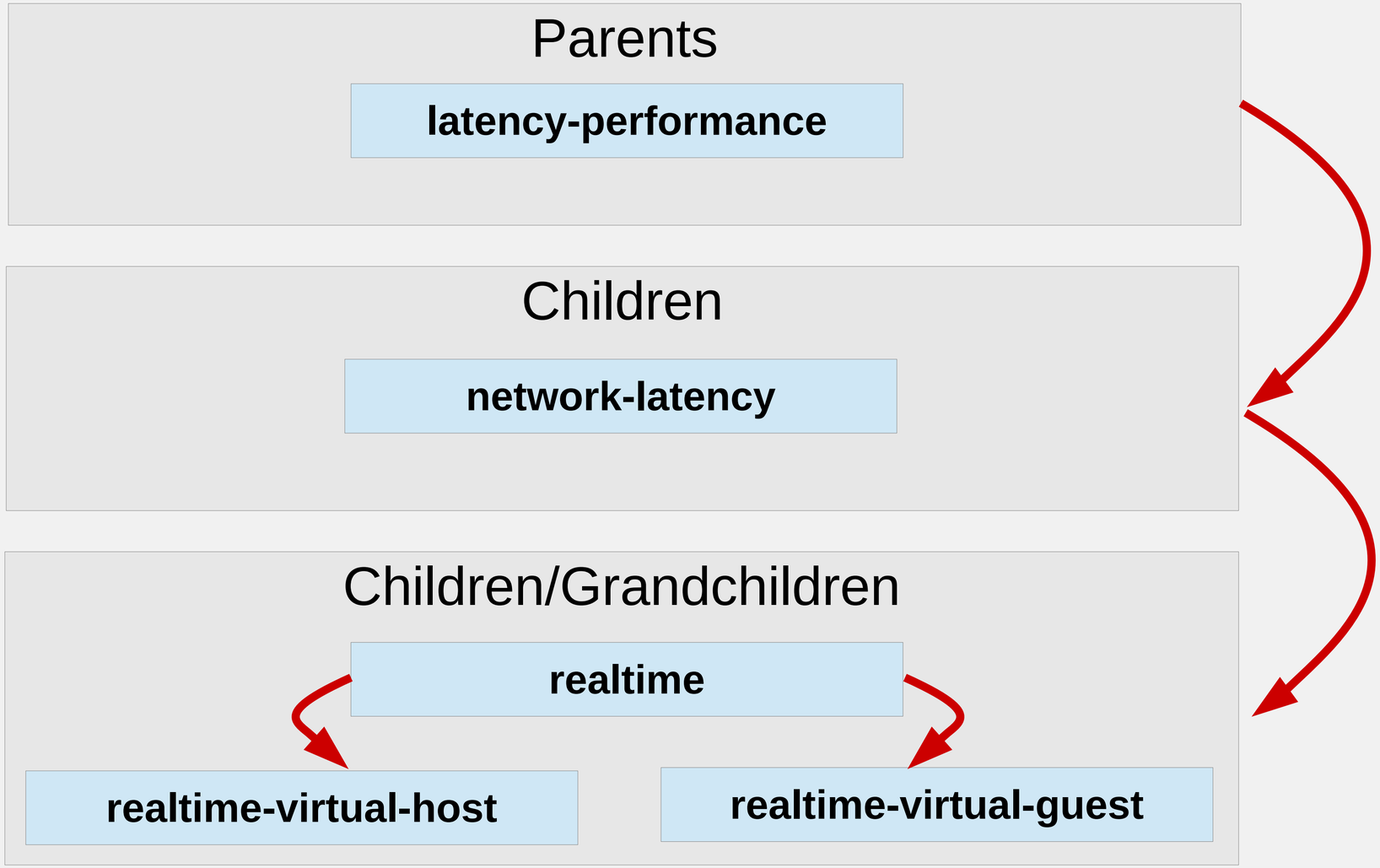
KVM-RT – tuning/confg host

```
[root@localhost ~]# hwlatdetect
hwlatdetect:  test duration 120 seconds
   parameters:
        Latency threshold: 10us
        Sample window:     1000000us
        Sample width:      500000us
     Non-sampling period:  500000us
        Output File:       None

Starting test
test finished
Max Latency: 11us
Samples recorded: 6
Samples exceeding threshold: 7
1466464193.0724182328   0       11
1466464211.0725493769   11      0
1466464234.0724915467   0       11
1466464276.0725694788   11      0
1466464280.0725769009   0       11
1466464286.0724895787   11      0
```

```
[root@localhost ~]# hwlatdetect
hwlatdetect:  test duration 120 seconds
   parameters:
        Latency threshold: 10us
        Sample window:     1000000us
        Sample width:      500000us
     Non-sampling period:  500000us
        Output File:       None

Starting test
test finished
Max Latency: 0us
Samples recorded: 0
Samples exceeding threshold: 0
```

* https://rt.wiki.kernel.org/index.php/HOWTO:_Build_an_RT-application
* https://access.redhat.com/ecosystem/search/#/category/Server

KVM-RT - tuned profiles



KVM-RT – tuning/config host

```
# yum install tuned-profiles-realtime tuned-profiles-nfv

# echo “isolated_cores=3,5,7” >> /etc/tuned/realtime-virtual-
host-variables.conf

# systemctl enable tuned
# systemctl start tuned
# tuned-adm profile realtime-virtual-host
```

KVM-RT – tuning/config host

In /etc/default/grub add:

```
default_hugepagesz=1G
```

Update the bootloader:

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

Set hugepage reservation:

```
# echo 2 >
/sys/devices/system/node/nodeY/hugepages/hugepages-
1048576kB/nr_hugepages
```

```
[root@virtlab502 proc]# cat /proc/cmdline
BOOT_IMAGE=/vmlinuz-3.10.0-327.18.2.rt56.223.el7_2.x86_64 root=/dev/mapper/rhel_virtlab502-root ro
crashkernel=auto rd.lvm.lv=rhel_virtlab502/root rd.lvm.lv=rhel_virtlab502/swap
console=ttyS1,115200 default_hugepagesz=1G isolcpus=3,5,7 nohz=on nohz_full=3,5,7
intel_pstate=disable nosoftlockup
[root@virtlab502 proc]#

[root@virtlab502 proc]# tuned-adm active
Current active profile: realtime-virtual-host
[root@virtlab502 proc]#



[root@virtlab502 proc]# cat /usr/lib/tuned/realtime-virtual-host/lapic_timer_adv_ns
5000 <= non matching 0
[root@virtlab502 proc]# cat /sys/module/kvm/parameters/lapic_timer_advance_ns
5000 <= non matching 0
[root@virtlab502 proc]#

[root@virtlab502 ~]# cat /sys/devices/system/node/node1/hugepages/hugepages-1048576kB/nr_hugepages
2
[root@virtlab502 ~]#
```

```
[root@compute ~]# vi /etc/nova/nova.conf
....
       # Defines which pcpus that instance vcpus can use. For example, "4-12,^8,15"
       # (string value)
       #vcpu_pin_set=<None>
       vcpu_pin_set=3,5,7....
```

```
[root@controller ~]#  . keystonerc_admin
[root@controller ~(keystone_admin)]# nova flavor-create realTime.medium 99 1024 100 4
+----+----------+-----------+------+-----------+------+-------+-------------+-----------+
| ID | Name     | Memory_MB | Disk | Ephemeral | Swap | VCPUs | RXTX_Factor | Is_Public |
+----+----------+-----------+------+-----------+------+-------+-------------+-----------+
| 99 | r1.small | 1024      | 10   | 0         |      | 4     | 1.0         | True      |
+----+----------+-----------+------+-----------+------+-------+-------------+-----------+
[root@controller ~(keystone_admin)]#
[root@controller ~(keystone_admin)]# nova flavor-key 99 set hw:cpu_policy=dedicated
[root@controller ~(keystone_admin)]# nova flavor-key 99 set hw:cpu_realtime=yes
[root@controller ~(keystone_admin)]# nova flavor-key 99 set hw:cpu_realtime_mask="^0-1"
[root@controller ~(keystone_admin)]# nova flavor-key 99 set hw:mem_page_size=1GB
```

KVM-RT – tuning/config guest

- Install kernel-rt in guest, too!
- Use same default_hugepagesz as host
- Install tuned profile: tuned-profiles-nfv

```
# echo “isolated_cores=2,3” >> /etc/tuned/realtime-virtual-guest-
variables.conf

# tuned-adm profile realtime-virtual-guest

# grep tuned_params= /boot/grub2/grub.cfg
set tuned_params=”isolcpus=2,3 nohz=on nohz_full=2,3
intel_pstate=disable nosoftlockup
```

KVM-RT – apps/ops

- Target applications are NFV networking workloads
- Types of operations to avoid
  - Disk IO
  - Video or Sound
  - Page faults or swapping
  - CPU hot-plug
  - Live migration

KVM-RT – testing

Run cyclictest: confirm guest latencies within expected limits

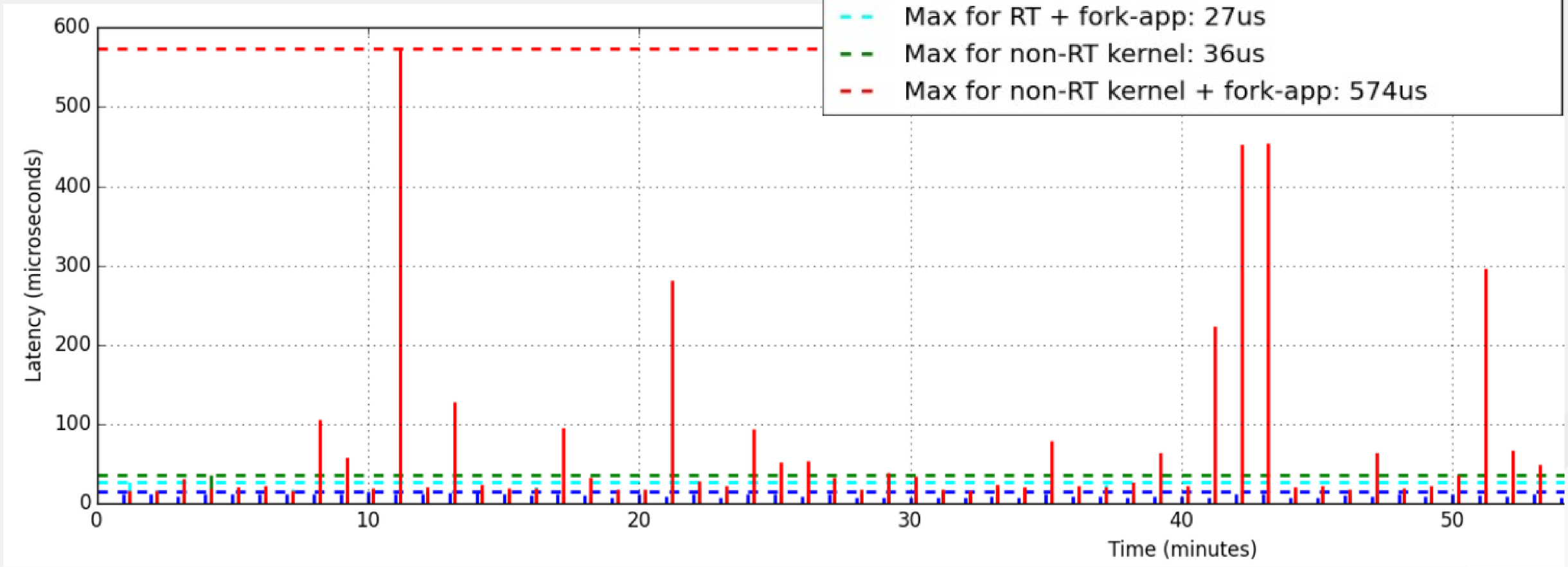
```
# taskset -c 2 <application>

# taskset -c 2 cyclictest -m -n -q -p95 -D 24h -h100 -i 200 >
cyclictest.out
```

KVM-RT – testing

- Tuning: real-time tuned profiles host and guest
- Fork app:
  - Simple application executing fork() repeatedly
  - Task that sends signal to fork app

KVM-RT – testing

cyclictest -m -n -q -p95 -D 60s -h60 -i 200 -a 1



KVM Live Migration

Precopy vs Auto Converge vs Postcopy

KVM Live Migration

- Software Dependability
  - Is Live Migration guaranteed to succeed?
    - Emergency Evacuation
    - Recoverable machine checks
    - Hardware maintainance
- Guest performance during live migration
  - Minimize CPU performance impact on the guest
- Live migration time
  - Take as little time as possible
    - To reduce network load as well
- Downtime latency
  - Minimize the downtime with source & destination both paused

Precopy

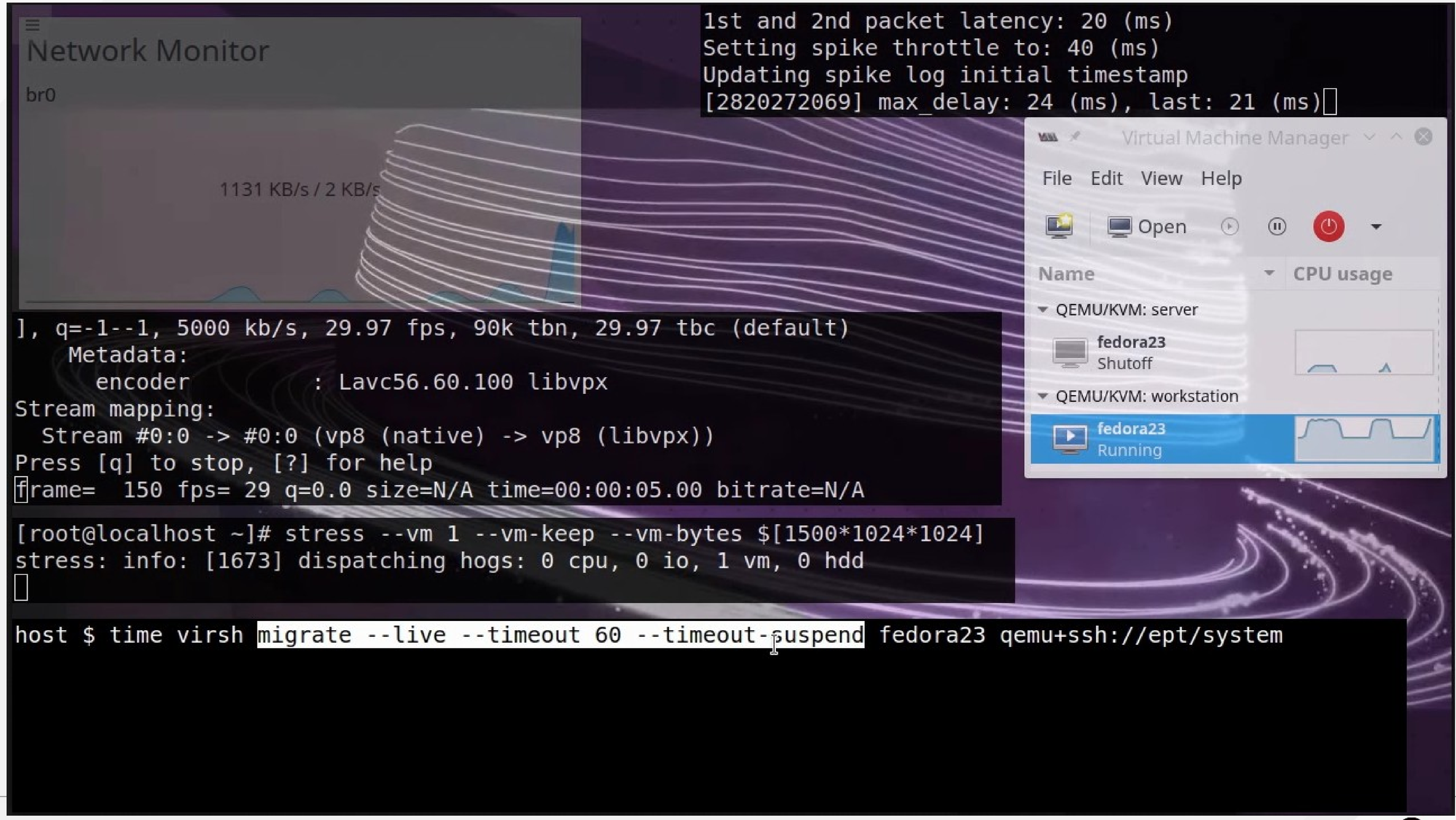
- Software Dependability
  - ***No***
- Guest performance during live migration
  - Good
- Live migration time
  - Bad, could never end/converge
- Downtime latency
  - Low, if we sacrifice dependability & migration time

Auto Converge

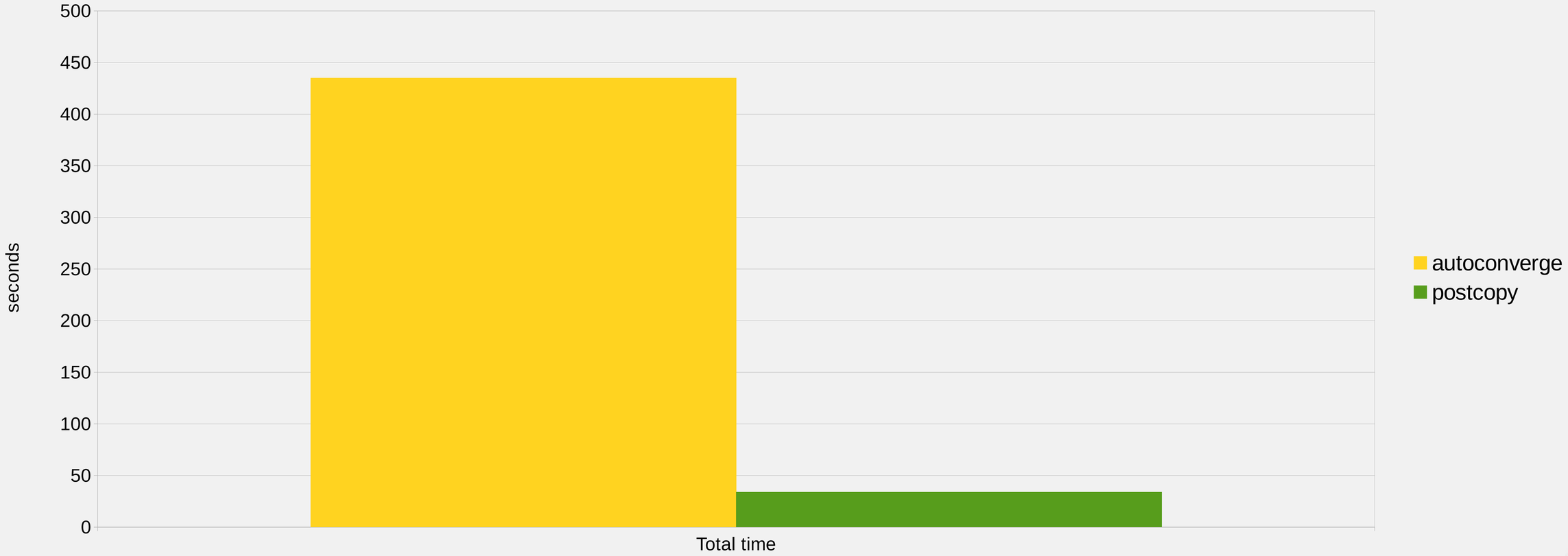
- Software Dependability
  - ***Yes***
- Guest performance during live migration
  - ***Bad, guest vCPUs may be throttled down heavily***
- Live migration time
  - Bad, the CPU throttling process takes time
- Downtime latency
  - Low, same as precopy
    - Artificial “latency” created during the auto-converge phase

Postcopy after Precopy

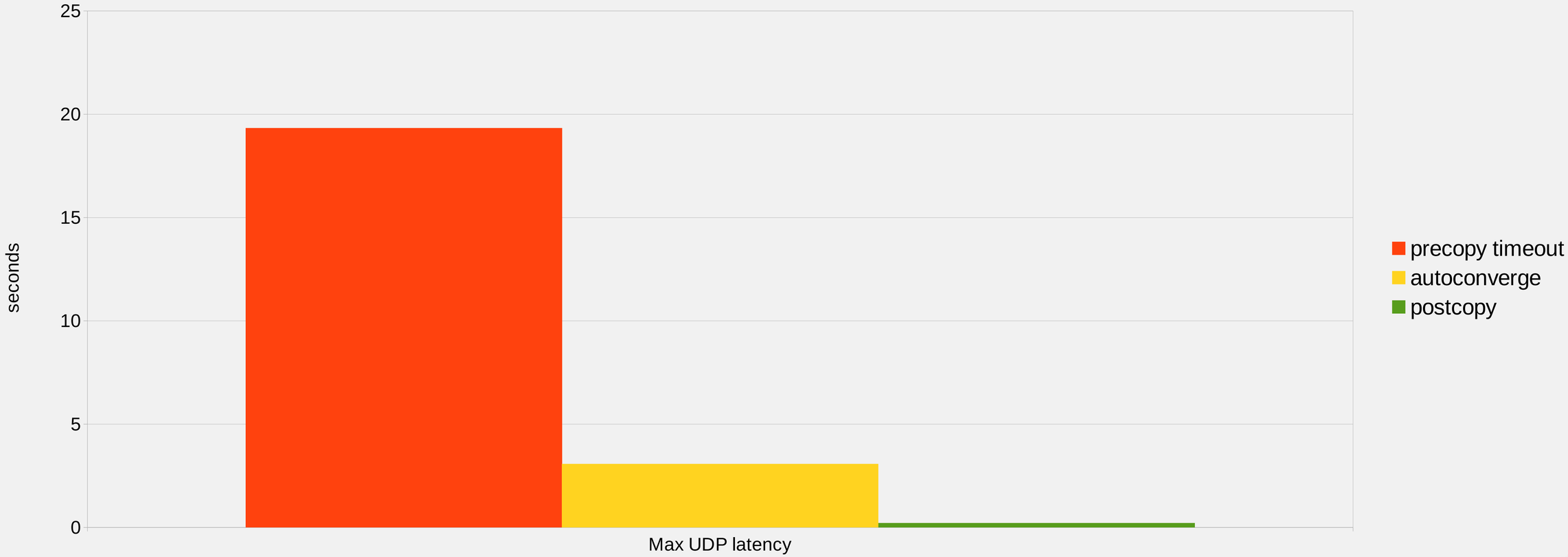
- Software Dependability
  - ***Yes***
    - Guest memory accesses might block waiting for network I/O *if the network hardware fails*
- Guest performance during live migration
  - ***Guest vCPUs performance is never throttled down***
  - The first access to some memory page in the destination may be delivered at network I/O bandwidth instead of RAM bandwidth, similar to a disk swapin
- Live migration time
  - Lower than precopy & auto converge and ***deterministic***
- Downtime latency
  - Lower than precopy & auto converge artificial latencies


Network Monitor
br0
1131 KB/s / 2 KB/s
1st and 2nd packet latency: 20 (ms)
Setting spike throttle to: 40 (ms)
Updating spike log initial timestamp
[2820272069] max_delay: 24 (ms), last: 21 (ms)
Virtual Machine Manager
File Edit View Help
Open
Name
CPU usage
QEMU/KVM: server
fedora23
Shutoff
QEMU/KVM: workstation
fedora23
Running
], q=-1--1, 5000 kb/s, 29.97 fps, 90k tbn, 29.97 tbc (default)
Metadata:
encoder : Lavc56.60.100 libvpx
Stream mapping:
Stream #0:0 -> #0:0 (vp8 (native) -> vp8 (libvpx))
Press [q] to stop, [?] for help
frame= 150 fps= 29 q=0.0 size=N/A time=00:00:05.00 bitrate=N/A
[root@localhost ~]# stress --vm 1 --vm-keep --vm-bytes $[1500*1024*1024]
stress: info: [1673] dispatching hogs: 0 cpu, 0 io, 1 vm, 0 hdd
host $ time virsh migrate --live --timeout 60 --timeout-suspend fedora23 qemu+ssh://ept/system

Live migration total time



Live migration max UDP guest delivery latency



Everything Else

“Meet The Experts” - Thurs 5:45-7 PM Free - Soda/Beer/Wine