

Red Hat Virtualization 4.1基础架构与命令汇总

文档说明：

- 使用VMware环境部署RH318v4.1环境，每次启动foundation0时都会同步时间。
- 需手动更改时间为2019年，并同步classroom与utility节点的时间，否则无法使用IdM服务！
- 添加RHHV节点时，网口可能down掉而无法与RHVM通信，可尽快将该网口up以加入RHHV集群。

RHV 4.1概述：

- Red Hat Enterprise Virtualization 4.1: RHV 4.1
- Red Hat Enterprise Virtualization Manager: RHVM
- Red Hat Enterprise Virtualization Hypervisor: RHHV

- RHV 4.1基于开源版本 **oVirt 4.1** 开发。
- RHVM不能与Red Hat IPA服务（IdM）安装在同一个物理主机上，两者的 **mod_ssl** 模块

存在冲突。

- RHV可以添加已安装有RHHV（RHV Host）与RHEL物理主机至数据中心的集群中，但是RHHV

（RHV Host）与RHEL物理主机必须安装 **vdsm** 软件包，使用 **vdsm** 守护进程作为代理与RHVM间连接。

- 使用RHEL作为集群主机将提供更多的软件包，包括超融合技术。
- RHV集群架构概览：

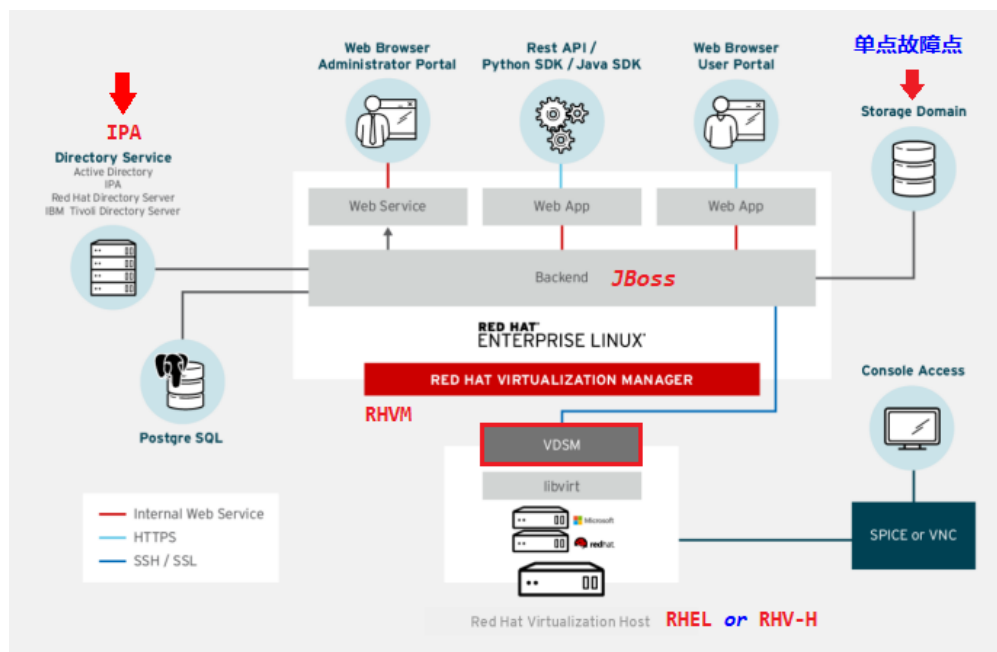


Figure 1.1: Overview of Red Hat Virtualization

RHV 4.1集群启停操作:

- 关闭RHV环境:

```
cluster hosts > maintainance > poweroff cluster hosts > poweroff
```

RHVM

- 启动RHV环境:

```
poweron RHVM > poweron cluster hosts > cluster hosts active
```

- 由于RHVH (hypervisor) 主机上可能跑着虚拟机, 升级RHVH必须设置成维护模式。

RHV逻辑网络架构 (logical network):

- 数据中心的一个逻辑网络可被一个或多个集群的主机共享。
- 不同数据中心间的逻辑网络可使用同一个网段, 并且可相互通信。
- 生产环境中集群物理主机的网口可桥接至数据中心的各个逻辑网络的Linux bridge

上, 根据业务需求, 将分配不同的网段对应不同的网口 (可使用VLAN划分), 如业务网络、存储网络、心跳网络、管理网络等。

- 每台物理主机中的虚拟机可具有多张虚拟网卡, 并将其桥接至对应的Linux bridge上,

即不同的逻辑网络上, 以实现不同类型的逻辑网络流量负载。

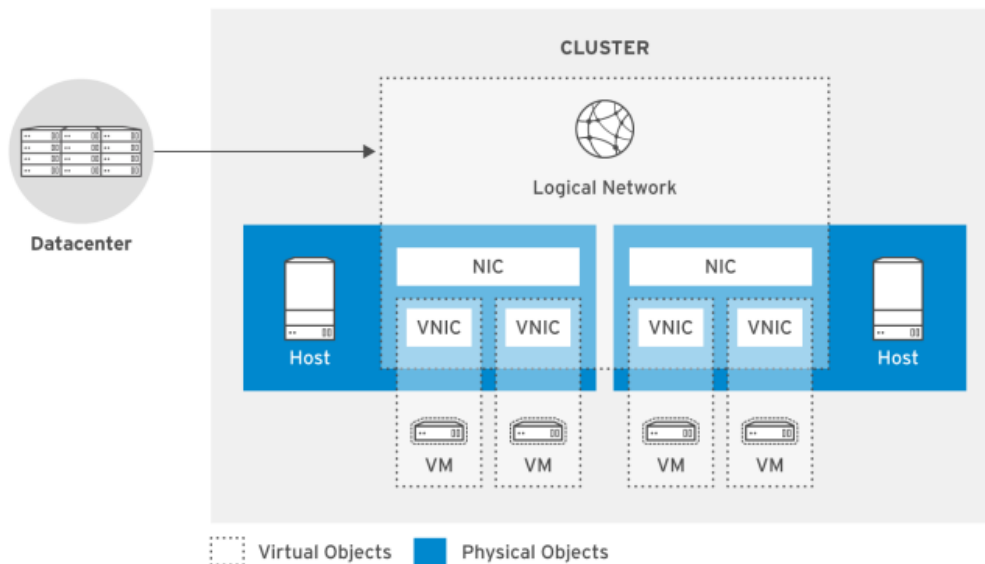
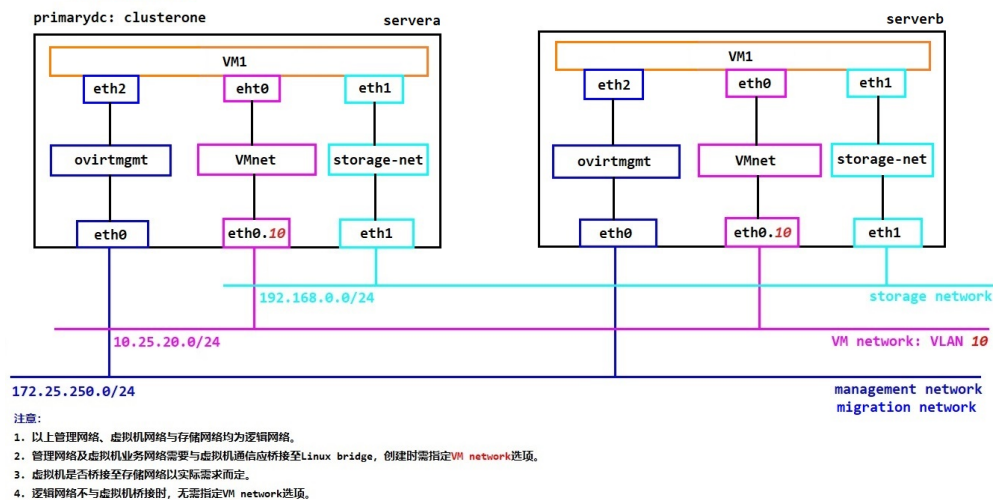


Figure 6.1: RHV networking

RHV 4.1逻辑网络拓扑示意



* 注意:

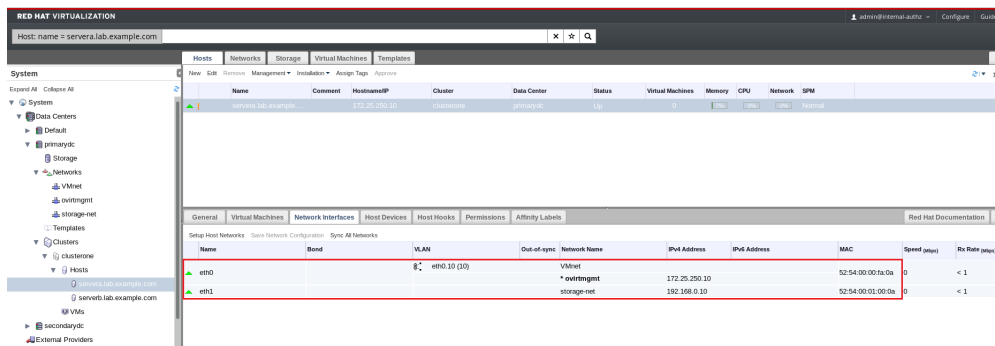
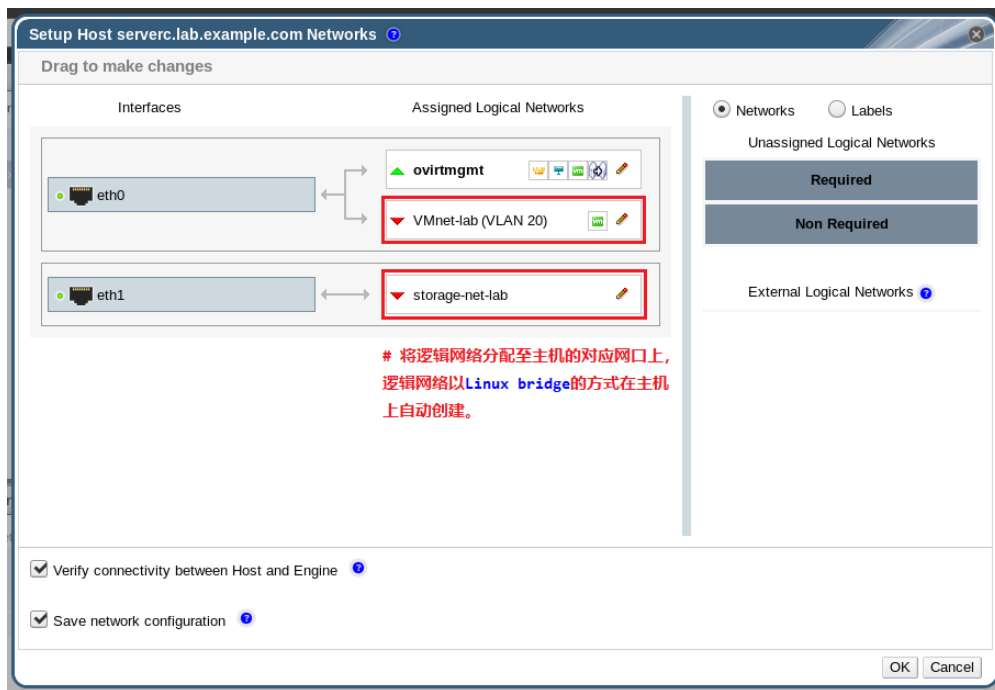
1. 在Web界面上创建逻辑网络并将其分配至RHVH的对应网口后, 将自动在RHVH上创建

Linux bridge并将物理网口桥接其上。

2. OpenStack Neutron作为RHV的逻辑网络仍处于技术预览阶段!

• 创建RHV逻辑网络:

1. 创建带VLAN tag的逻辑网络需要有外部网络支持 (该示例在serverb上抓包验证)。



```
[root@servera ~]# ip address show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master ovirtmgmt state UP qlen 1000
    link/ether 52:54:00:00:fa:0a brd ff:ff:ff:ff:ff:ff
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 52:54:00:01:00:0a brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.10/24 brd 192.168.0.255 scope global eth1 # 存储网络 (storage network), 该示例中存储
        valid_lft forever preferred_lft forever                                网络未与虚拟机桥接, 不创建相应Linux bridge.
    inet6 fe80::5054:ff:fe01:a/64 scope link
        valid_lft forever preferred_lft forever
19: ovirtmgmt: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP qlen 1000
    link/ether 52:54:00:00:fa:0a brd ff:ff:ff:ff:ff:ff
    inet 172.25.250.10/24 brd 172.25.250.255 scope global dynamic ovirtmgmt # 虚拟机管理与迁移的逻辑网络
        valid_lft 470sec preferred_lft 470sec
    inet6 fe80::5054:ff:fe00:fa0a/64 scope link
        valid_lft forever preferred_lft forever
20: vdsmdummy:: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN qlen 1000
    link/ether 6a:14:dd:12:f0:a8 brd ff:ff:ff:ff:ff:ff
22: eth0.10@eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master VMnet state UP qlen 1000
    link/ether 52:54:00:00:fa:0a brd ff:ff:ff:ff:ff:ff
23: VMnet: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP qlen 1000 # 虚拟机的业务网络
    link/ether 52:54:00:00:fa:0a brd ff:ff:ff:ff:ff:ff
    inet 10.25.20.10/24 scope global VMnet
        valid_lft forever preferred_lft forever

[root@servera ~]# brctl show
bridge name      bridge id        STP enabled      interfaces
vdsmdummy;       8000.000000000000  no                no
VMnet             8000.52540000fa0a  no                eth0.10 # VLAN ID: 10
ovirtmgmt         8000.52540000fa0a  no                eth0
```

```

[root@serverb ~]# tcpdump -i eth0 -e -p icmp -vv
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
13:56:06.126472 52:54:00:00:fa:0a (oui Unknown) > 52:54:00:00:fa:0b (oui Unknown), ethertype 802.1Q (0x8100), length 102: vlan 10, p 0, et
hertype IPv4, (tos 0x0, ttl 64, id 11384, offset 0, flags [DF], proto ICMP (1), length 84)
10.25.20.10 > serverb.lab.example.com: ICMP echo request, id 13186, seq 42, length 64
13:56:07.127389 52:54:00:00:fa:0a (oui Unknown) > 52:54:00:00:fa:0b (oui Unknown), ethertype 802.1Q (0x8100), length 102: vlan 10, p 0, et
hertype IPv4, (tos 0x0, ttl 64, id 11986, offset 0, flags [DF], proto ICMP (1), length 84)
10.25.20.10 > serverb.lab.example.com: ICMP echo request, id 13186, seq 43, length 64
13:56:08.127319 52:54:00:00:fa:0a (oui Unknown) > 52:54:00:00:fa:0b (oui Unknown), ethertype 802.1Q (0x8100), length 102: vlan 10, p 0, et
hertype IPv4, (tos 0x0, ttl 64, id 12288, offset 0, flags [DF], proto ICMP (1), length 84)
10.25.20.10 > serverb.lab.example.com: ICMP echo request, id 13186, seq 44, length 64
13:56:09.127139 52:54:00:00:fa:0a (oui Unknown) > 52:54:00:00:fa:0b (oui Unknown), ethertype 802.1Q (0x8100), length 102: vlan 10, p 0, et
hertype IPv4, (tos 0x0, ttl 64, id 12374, offset 0, flags [DF], proto ICMP (1), length 84)
10.25.20.10 > serverb.lab.example.com: ICMP echo request, id 13186, seq 45, length 64
13:56:10.127123 52:54:00:00:fa:0a (oui Unknown) > 52:54:00:00:fa:0b (oui Unknown), ethertype 802.1Q (0x8100), length 102: vlan 10, p 0, et
hertype IPv4, (tos 0x0, ttl 64, id 12752, offset 0, flags [DF], proto ICMP (1), length 84)
10.25.20.10 > serverb.lab.example.com: ICMP echo request, id 13186, seq 46, length 64
13:56:11.126989 52:54:00:00:fa:0a (oui Unknown) > 52:54:00:00:fa:0b (oui Unknown), ethertype 802.1Q (0x8100), length 102: vlan 10, p 0, et
hertype IPv4, (tos 0x0, ttl 64, id 13303, offset 0, flags [DF], proto ICMP (1), length 84)
10.25.20.10 > serverb.lab.example.com: ICMP echo request, id 13186, seq 47, length 64
^C

```

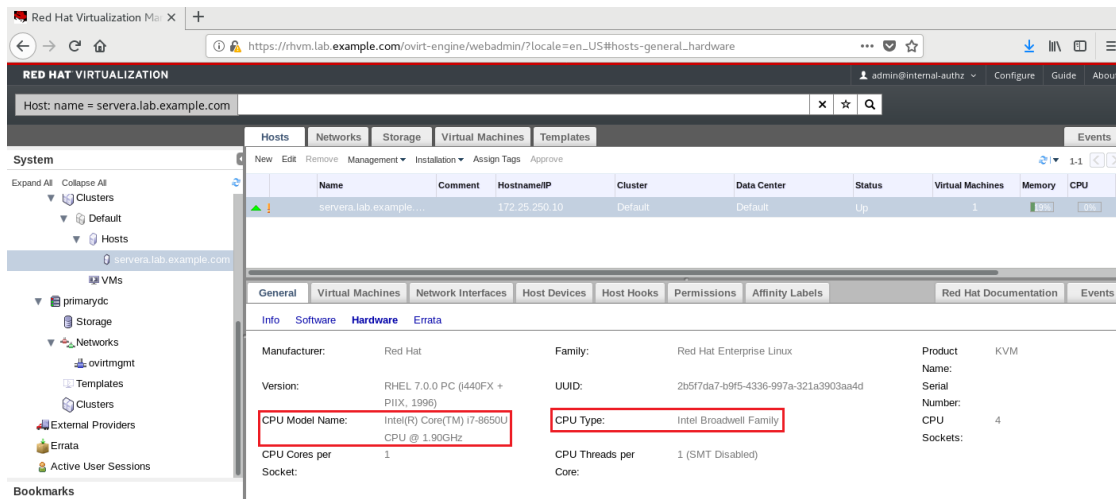
创建RHV数据中心（data center）与存储域（storage domain）：

- 数据中心中Storage、Networks（Logical Networks）与Clusters处于同一级别。
- 在数据中心中创建Cluster时，应注意之后在其中创建的物理主机的CPU型号，应保证

Cluster的CPU型号与前者的兼容性。

- 若集群中RHVH或RHEL主机具有不同模式的CPU类型（model），整个集群将降级（downgrade）

来满足基本的vCPU的模式，这将保证虚拟机能在集群中的所有主机上的实时迁移及兼容性。



- 能够将物理主机先添加入默认创建的Default数据中心，查看其CPU类型后，再将其转移

至新创建的数据中心集群中。

- 集群（cluster）也就是一个迁移域（migration domain），集群内的虚拟机只能在同

一个集群内迁移。

- 创建存储域（storage domain）：

1. 由于需要RHVH或RHEL主机作为SPM，在数据中心中添加data domain前，必须先添加至少

一台RHVH或RHEL主机，并且处于Up状态。

2. 集群中的主机可手动指定为 **storage pool manager (SPM)**，也可由RHVM根据优先级

自动选择主机为SPM，并且SPM可跨集群。

3. 存储域概要：

a. 数据存储域 (data domain)：

- 1) data domain存储虚拟机磁盘镜像与模板。
- 2) data domain可供同一数据中心中的所有集群共享使用 (shared)。
- 3) data domain可由 NFS、iSCSI、FCP (FC SAN)、local disk 提供。
- 4) **必须先创建 data domain (Master)，才能再创建ISO domain与export domain。**

b. ISO存储域 (ISO domain)：

- 1) ISO domain存储操作系统与应用部署的ISO文件。
- 2) ISO domain只能由 NFS 提供，并且可在不同数据中心间共享。
- 3) ISO domain在共享前需设置为维护状态，再attach至其他数据中心。

c. 导出存储域 (export domain)：

- 1) export domain作为备份与迁移的虚拟机临时存储。
- 2) export domain只能由 NFS 提供，作为非必须存储域 (optional)，

在需要时

创建，作为虚拟机的备份与虚拟机导出或导入数据中心的过度存储。

4. 创建各个存储域时，将会有短暂的 Locked 状态，必须等待为 Up 状态后，才能继续创建

其他存储域。

5. 同一个数据中心的Storage中可以包含一个或多个data domain、一个export domain与

一个ISO domain。

6. 只有创建好存储域后，数据中心才能从 **Uninitialized** 状态切换到 **Up** 状态。

安装RHEL虚拟机：

- VMware环境中，安装RHEL虚拟机前，必须关闭VMware Workstation的自定义网络中的

DHCP功能，否则将干扰Kickstart的安装过程。

- 根据虚拟机所在的逻辑网络，配置虚拟机的MAC地址池范围 (**MacPoolRanges**)。
- 使用Kickstart脚本部署虚拟机需要跑安装过程，占用较高资源，而使用虚拟机模板

(template) 快速部署减少资源占用。

- RHV中默认的5种虚拟机实例类型：

- **Tiny**: 1 vCPU, 512 MB RAM
- **Small**: 1 vCPU, 2048 MB RAM
- **Medium**: 2 vCPUs, 4096 MB RAM
- **Large**: 2 vCPUs, 8192 MB RAM
- **XLarge**: 4 vCPUs, 16384 MB RAM

- 由RHV管理的虚拟机使用本地的引导磁盘，其中包含完整的操作系统与应用二进制文件，

额外的应用数据存储于虚拟机可直接访问的iSCSI或NFS中。



IMPORTANT

A recommended practice is to give your virtual machine a local boot disk managed by RHV that contains the operating system and application binaries. Any additional disks for application data are probably best managed externally and accessed directly by the virtual machine using iSCSI or NFS.

The advantage of this method is that virtual machines can be managed and backed up separately from application data. A virtual machine can be rebuilt quickly from a snapshot or template, and then the application data can be remounted from the SAN or NAS. Performance may also be better for the application by not having the Red Hat Virtualization host translate storage traffic for the virtual machine.

创建与恢复虚拟机快照（snapshot）：

- 建议在虚拟机关闭的情况下创建快照，虽然RHV 4.1版本支持虚拟机开机状态下创建快照。
- 还原（restore）虚拟机快照时，选中需还原的快照后，先进行preview预览该虚拟机状态，

若确定无误后，再点击commit还原该快照。

- KVM虚拟机快照缺点：还原快照后，该快照后的所有快照将被自动清除！**
- 可使用已存在的快照克隆（clone）虚拟机，克隆是在新硬件上的虚拟机拷贝，但是克隆

的虚拟机包含有原虚拟机的个性化信息。

- 可使用已存在的快照创建模板（template）来部署不具有个性化信息的虚拟机。

导入与导出虚拟机磁盘镜像：

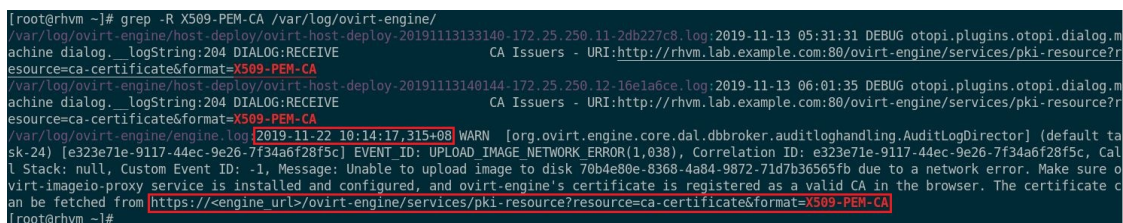
- 虚拟机磁盘镜像存储于data domain中。
- data domain在同一时刻只能attach至一个数据中心，而数据中心可同时具有多个data domain。

- 使用export domain可以将虚拟机从一个数据中心导出，并导入到另一个数据中心。

*** 注意：** export domain在RHV 4.1中为**deprecated**状态！

- 导入外部虚拟机镜像至data domain中：

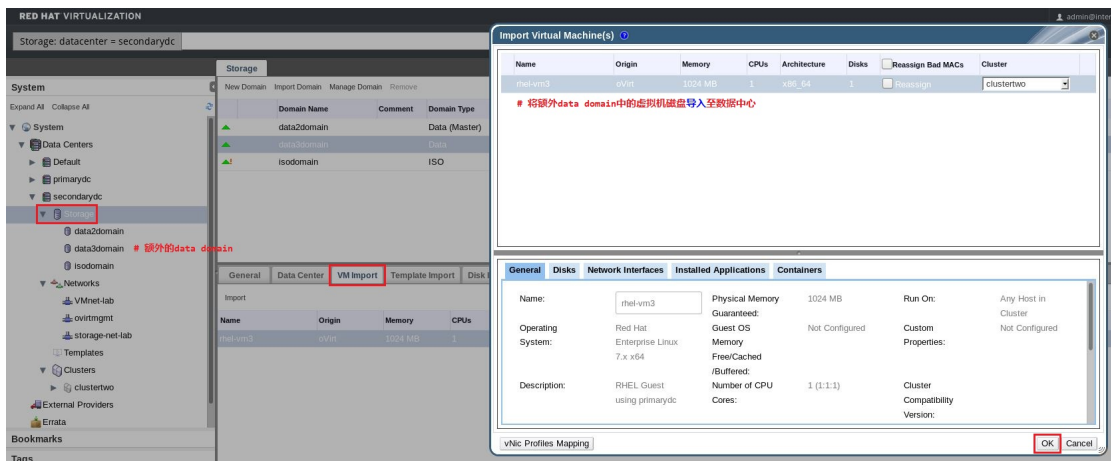
- 选中指定数据中心的data domain > Disks > Upload（RAW或QCOW2镜像） > Start
- Upload镜像时，网络报错需打开浏览器安装相应的CA证书即可。



- RHV虚拟机导入至export domain，需以OVA（Open Virtual Appliance）文件的方式存储，

格式为RAW或QCOW2。

- RHVM将虚拟机以**OVF**包的目录结构形式导出至**export domain**，OVF包的扩展名为**.ovf**。
- **OVA**文件是**OVF**包的目录结构的**tar**归档。
- 虚拟机磁盘在数据中心间的导入与导出使用额外的**data domain**来实现。



* 注意:

使用**export domain**实现虚拟机在数据中心间导出与导入: **RHV**旧版本使用的方法

1. 若不存在**export domain**, 需创建**export domain**, 并且只能以**NFS**服务的方式提供存储。

将新建或已存在的**export domain**与一个数据中心关联, 该数据中心必须先创建**data**

domain作为**Master**, 并且为**Active**状态。

2. 将虚拟机导入到数据中心的**export domain**中, 虚拟机必须为关闭状态, 并且虚拟机的

所有快照将全部合并 (merge)。

3. 将**export domain**配置为 **maintenance** 状态, 再将其配置为 **Detach** 状态, 与数据

中心解除关联, 最后选中 **System/Storage** 后删除**export domain**。

4. 选中另一个数据中心, 导入**export domain**。

在导入到另一个数据中心前, 在**export domain**中, 可使用 **libguestfs-tools** 软件

包中的**guestfish**命令在虚拟机关机状态下进行镜像修改, 此时虚拟机镜像以**OVA**文件

QCOW2格式存储。

5. 在另一个数据中心中从**export domain**中导入虚拟机。

模板 (template) 与虚拟机池:

- 模板: 预配置虚拟机的拷贝, 用于简化多个虚拟机的再次创建。
- 封装模板 (seal template) 可去除模板中的虚拟机个性化信息, 如主机名、SSH主机公钥

对、网口的**MAC**地址等。

- 虚拟机池 (pool): 来源于同一个虚拟机模板克隆的一组虚拟机
- 当**pool**中的部分虚拟机运行于同一物理主机上时, 由于其来源于同一模板, 物理主机可以使用

内核同页合并 (KSM) 机制, 合并虚拟机使用的相同内存页表生成共享内存, 提高物理主机的

内存使用率与**CPU**命中率, 降低资源占用。

- **pool**中的虚拟机在关闭时将丢失开机状态下的全部数据，返回为基准模板，即**pool**中的虚拟机

为无状态虚拟机。

- **pool**中的虚拟机能够被分配相应权限的IPA域中的用户，用户能够登录 **User Portal** 界面来

使用虚拟化桌面，在教学环境下非常实用。

- **pool**中的虚拟机通过 **Detach** 与**pool**解除关联后成为普通虚拟机（虚拟机图标发生变化），

才能将其**Remove**删除，否则将报错。

- 必须先创建虚拟机模板，才能再创建**pool**！

迁移（migration）与调度策略（scheduling policy）：

- 在虚拟机迁移传输完成时，与虚拟机通信的客户端应该注意到不超过几毫秒（**millisecond**）

的网络暂停。

* **注意：**虚拟机实时迁移时，可以使用**ping**来测试迁移的瞬断与丢包。

- 集群为虚拟机的迁移域，其中的主机具有相同的CPU架构及特性，并且使用共享存储。
- 虚拟机实时迁移只能从集群中的一台主机迁移至另一台主机。
- 虚拟机实时迁移的条件：

1. 迁移的源主机与目标主机均为**Up**状态。

2. 迁移的源主机与目标主机可访问相同的虚拟网络、VLAN与数据域（**Shared**类型的数据中心）。

3. 目标主机必须有足够的CPU负载与RAM来支持虚拟机的迁移。

- **ovirtmgmt**逻辑网络可作为虚拟机的管理网络与迁移网络！
- 可限制每个实时迁移的最大传输速率及迁移的最大并发数目。
- 虚拟机迁移是一种网络敏感操作。
- 运行**10**个或更多虚拟机的主机，迁移所有的虚拟机将是一个漫长且资源消耗的过程。
- 当集群中的主机设置为**Maintenance**状态时，在该主机上运行的所有虚拟机将迁移至集群内

的其他主机。

Edit Cluster ⓘ

General
Optimization
Migration Policy
Scheduling Policy
Console
Fencing Policy
MAC Address Pool

Migration Policy # 迁移策略

Minimal downtime
A policy that lets the VM migrate in typical situations. The VM should not experience any significant downtime. If the VM migration is not converging for a long time, the migration will be aborted. The guest agent hook mechanism is enabled.

Bandwidth
Migration bandwidth limit (Mbps) ⓘ Auto

Resilience Policy ⓘ

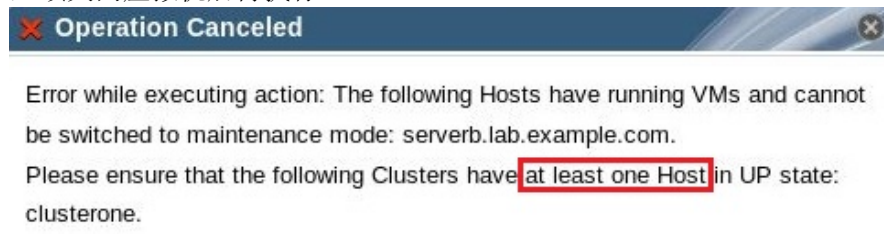
☒ Migrate Virtual Machines
☐ Migrate Only Highly Available Virtual Machines
☐ Do Not Migrate Virtual Machines

Additional Properties
Auto Converge migrations Inherit from global setting
Enable migration compression Inherit from global setting

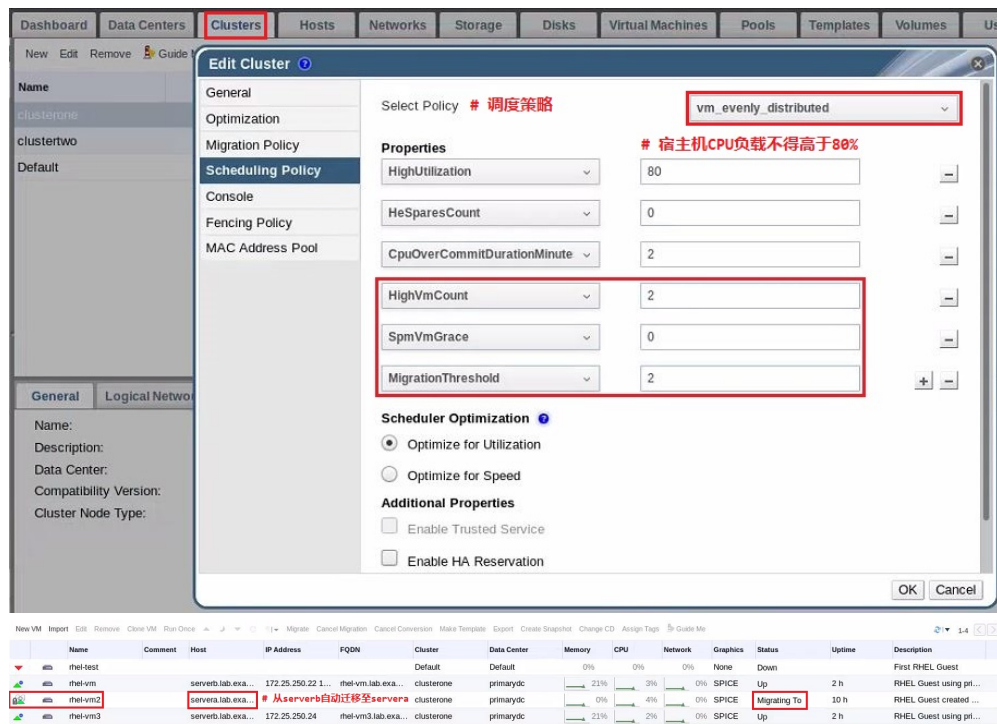
OK Cancel

- 若集群中只有一台主机运行，并且该主机上运行有虚拟机，则该主机无法设置为维护状态，

必须关闭虚拟机后再执行。



- 三种虚拟机迁移方式：
 1. 冷迁移
 2. 手动实时迁移：不影响业务
 3. 增加host主机CPU负载来实现自动迁移：调整虚拟机调度策略
- 默认情况下，RHVM配置的虚拟机调度策略为None。
- 可设置宿主机超过2分钟CPU负载高于80%时，迁移其上的虚拟机。



部署RHV环境：BIOS中必须开启物理CPU的相关特性

- `$ grep -E --color=auto 'lm' /proc/cpuinfo`
查看CPU是否支持64 bit长模式（long mode）
- `$ grep -E --color=auto 'vmx|svm' /proc/cpuinfo`
查看CPU是否支持Intel VT-x硬件虚拟化技术
- `$ grep -E --color=auto 'nx' /proc/cpuinfo`
查看CPU是否关闭eXecute（No eXecute），CPU标记为nx。
该CPU标记用于防止内存特定区段溢出，用于防病毒。
- `$ wget -P /etc/yum.repos.d/ http://materials.example.com/rhvm.repo`
下载RHV相关软件源repo文件
`/* ANOTHER WAY */`
`$ curl -o`
`/etc/yum.repo.d/rhvm.repo http://materials.example.com/rhvm.repo`
下载RHV相关软件源repo文件
- `$ yum update -y`
升级系统软件包
- `$ yum install -y rhevm`
安装RHV相关软件包，其中RHV 4.1版本软件包名为rhevm。
* 注意：正式安装时需注册Red Hat虚拟化软件订阅频道

- **\$ engine-setup --generate-answer=/root/answer.txt**

根据交互式提示配置RHV环境与生成answer应答文件，并完成RHV的部署。

```
[ INFO ] Stage: Transaction setup
[ INFO ] Stopping engine service
[ INFO ] Stopping ovirt-fence-kdump-listener service
[ INFO ] Stopping dwh service
[ INFO ] Stopping Image I/O Proxy service
[ INFO ] Stopping vmconsole-proxy service
[ INFO ] Stopping websocket-proxy service
[ INFO ] Stage: Misc configuration
[ INFO ] Stage: Package installation
[ INFO ] Stage: Misc configuration
[ INFO ] Upgrading CA
[ INFO ] Initializing PostgreSQL
[ INFO ] Creating PostgreSQL 'engine' database
[ INFO ] Configuring PostgreSQL
[ INFO ] Creating PostgreSQL 'ovirt_engine_history' database
[ INFO ] Configuring PostgreSQL
[ INFO ] Creating CA
[ INFO ] Creating/refreshing Engine database schema
[ INFO ] Creating/refreshing DWH database schema
[ INFO ] Configuring Image I/O Proxy
[ INFO ] Setting up ovirt-vmconsole proxy helper PKI artifacts
[ INFO ] Setting up ovirt-vmconsole SSH PKI artifacts
[ INFO ] Configuring WebSocket Proxy
[ INFO ] Creating/refreshing Engine 'internal' domain database schema
[ INFO ] Generating post install configuration file '/etc/ovirt-engine-setup.conf.d/20-setup-ovirt-post.conf'
[ INFO ] Stage: Transaction commit
[ INFO ] Stage: Closing up
[ INFO ] Starting engine service
[ INFO ] Starting dwh service
[ INFO ] Restarting ovirt-vmconsole proxy service

--== SUMMARY ==--

[ INFO ] Restarting httpd
Please use the user 'admin@internal' and password specified in order to login
Web access is enabled at:
    http://rhvm.lab.example.com:80/ovirt-engine
    https://rhvm.lab.example.com:443/ovirt-engine
Internal CA EB:9B:6F:2A:24:C2:10:45:C6:04:20:D7:17:32:4C:B8:E3:50:601
SSH fingerprint: f9:a7:42:0e:22:c8:2d:99:a8:db:50:9f:83:27:37:5b
[WARNING] Less than 16384MB of memory is available

--== END OF SUMMARY ==--

[ INFO ] Stage: Clean up
Log file is located at /var/log/ovirt-engine/setup/ovirt-engine-setup-20201110024232-jok13n.log
[ INFO ] Generating answer file '/var/lib/ovirt-engine/setup/answers/20201110030256-setup.conf'
[ INFO ] Generating answer file '/root/answer.txt'
[ INFO ] Stage: Pre-termination
[ INFO ] Stage: Termination
[ INFO ] Execution of setup completed successfully
[root@rhvm ~]#
```

* 注意:

1. /var/log/ovirt-engine: ovirt-engine日志文件存储路径
2. /etc/ovirt-engine: ovirt-engine配置文件存储路径
3. engine-setup将自动配置firewall防火墙规则，不与iptables规则冲突。

```
[root@rhvm ~]# cd /etc/ovirt-engine/firewalld/
[root@rhvm firewalld]# ls -lh
total 28K
-rw-r--r--. 1 root root 216 Nov 10 20:44 ovirt-fence-kdump-listener.xml
-rw-r--r--. 1 root root 185 Nov 10 20:44 ovirt-https.xml
-rw-r--r--. 1 root root 182 Nov 10 20:44 ovirt-http.xml
-rw-r--r--. 1 root root 192 Nov 10 20:44 ovirt-imageio-proxy.xml
-rw-r--r--. 1 root root 192 Nov 10 20:44 ovirt-postgres.xml
-rw-r--r--. 1 root root 207 Nov 10 20:44 ovirt-vmconsole-proxy.xml
-rw-r--r--. 1 root root 206 Nov 10 20:44 ovirt-websocket-proxy.xml
[root@rhvm firewalld]# cat ovirt-postgres.xml
<?xml version="1.0" encoding="utf-8"?>
<service>
  <short>ovirt-postgres</short>
  <description>oVirt configured postgres service</description>
  <port protocol="tcp" port="5432"/>
</service>
[root@rhvm firewalld]#
```

- **\$ grep " ERROR " /var/log/ovirt-engine/setup/ovirt-engine-setup-*.log**

- # 查看ovirt-engine配置过程是否报错
- \$ systemctl status ovirt-engine.service
- # 查看ovirt-engine守护进程状态
 - * 注意:
 - \$ cd /var/log/ovirt-engine/setup
 - \$ tail -f /var/log/yum.log
 - # 查看RHV安装过程的两种方法
 - \$ engine-setup --config-append=/root/answers.txt
- # 使用已有的应答文件重新配置部署RHV环境，常用于RHV备份与故障恢复。

RHV远程升级RHHV:

- \$ yum install -y rhv-hypervisor7
- # RHV安装 **rhv-hypervisor7** 软件包用于远程升级物理主机RHHV
- # 升级物理主机RHHV，选择最新的RHHV iso镜像进行升级。

RHV添加集群物理主机:

- \$ yum update -y
- # 升级物理主机所有软件包
- \$ systemctl stop NetworkManager.service
- # 停止NetworkManager服务，否则将干扰RHV逻辑网络部署，导致物理主机无法加入集群。
- \$ systemctl disable NetworkManager.service
- # 取消NetworkManager服务开机自启动
- \$ wget -P /etc/yum.repos.d/ <http://materials.example.com/rhvh.repo>
- # 下载RHHV相关软件源
- \$ yum install -y vdsm
- # vdsm软件包的vdsmd服务提供RHVM的管理代理，RHVM通过该代理管理RHHV或RHEL主机。
- # vdsmd服务默认不自动启动，在物理主机被成功添加加入集群后将自动启动，并能够开机自启。
- # 在RHV的Web界面上将物理主机添加加入相应的数据中心集群中。

RHV环境配置与更改:

- \$ yum install -y spice-xpi
- # spice-xpi软件包作为Firefox插件扩展，能在Web浏览器上使用spice协议登陆查看虚拟机终端。

该软件包安装于使用Web界面管理RHV的主机上。

- **\$ engine-config -a**

查看RHV全部可配置的参数与默认值

engine-config命令常用选项与参数:

-l, --list	列出可配置的键以及说明
-a, --all	列出可配置的键以及已生效的值
-s, --set	配置RHV环境参数

MacPoolRanges

RHV创建虚拟机的MAC地址池范围

UserSessionTimeoutInterval

非活动用户的登录退出超时时间间隔

(min)

ServerRebootTimeout

物理主机重启的超时时间 (s)

AdminPassword

oVirt引擎internal域的管理员密码

MaxVmsInPool

虚拟机池中的最大虚拟机数量

- **\$ engine-manage-domains add **

--domain=<domain_name> --provider=<provider_name> --user=

<user_name>

/* EXAMPLE */

**\$ engine-manage-domains add **

--domain=[example.com](#) --provider=IPA --user=rhvadmin

将RHV与IPA域关联，添加IPA域用户至RHV中。

- **\$ ovirt-aaa-jdbc-tool user unlock admin**

admin@internal用户多次登录验证失败后（默认为5次），该用户将被锁定无法再登录。

在rhvm节点上执行以上命令以解锁该用户。

- **\$ ovirt-aaa-jdbc-tool user password-reset admin **

--password-valid-to="2021-12-25 12:00"

交互模式重置admin@internal内部域管理员用户的密码

- **\$ engine-config -s MacPoolRanges=52:54:00:00:00:10-52:54:00:00:00:FF**

配置RHV环境中物理主机中的虚拟机MAC地址池范围

由于使用PXE网络引导方式安装虚拟机系统，其IP地址由DHCP服务器根据相应的MAC地址进行分配。

- **\$ systemctl restart ovirt-engine.service**

配置好RHV环境后需重启ovirt-engine守护进程，使配置生效。

- **\$ engine-iso-upload **

--iso-domain=<iso_domain_name> upload /path/of/<iso_name>.iso

上传物理主机或虚拟机安装引导时所需的ISO镜像至ISO存储域中，也可直接拷贝该iso

镜像至ISO存储域的NFS共享目录中。

* 注意：

1. 上传的ISO镜像需更改其属组与权限，UID与GID均为36，分别为vds用户与kvm组。

2. 如下所示：

```
$ chown 36:36 /path/of/<iso_name>.iso
```

```
$ chmod 0640 /path/of/<iso_name>.iso
```

```
[root@utility ~]# cd /exports/iso/37f5fbec-0c06-4d81-96db-9aa587904954/images/
[root@utility images]# ls -l
total 0
drwxr-xr-x. 2 36 36 92 Nov 12 12:56 11111111-1111-1111-1111-111111111111
[root@utility images]# cd 11111111-1111-1111-1111-111111111111/
[root@utility 11111111-1111-1111-1111-111111111111]# ls -lh
total 4.0G
-rw-r-----. 1 36 36 451M Sep  2 2019 rhel-7-server-supplementary-20170731.iso
-rw-r-----. 1 36 36 3.6G Nov 12 12:49 rhel-server-7.3-x86_64-dvd.iso
[root@utility 11111111-1111-1111-1111-111111111111]#
```

- \$ guestfish -i -w -a \

/exports/rhviso0/<export_uuid>/images/<vm_uuid>/<disk_uuid>

使用guestfish命令在虚拟磁盘镜像文件不开启的状态下进行更改配置