

Развёртывание и настройка oVirt 4.0. Часть 3.

Базовые операции в oVirt

blog.it-kb.ru/2016/09/14/install-ovirt-4-0-part-2-about-data-center-iso-domain-logical-network-vlan-vm-settings-console-guest-agent-live-migration

Автор: Алексей Максимов

14.09.2016

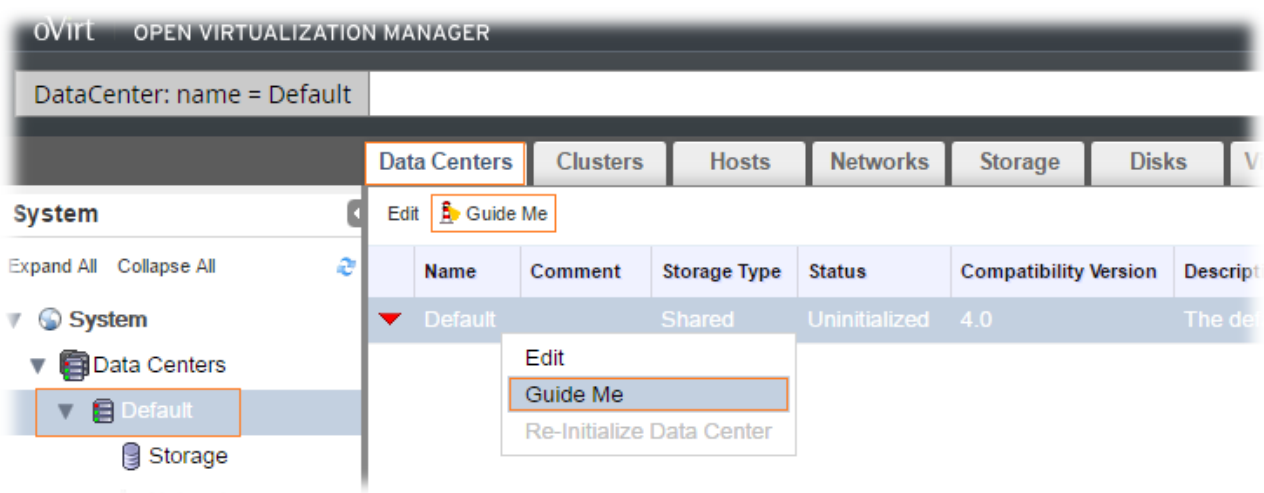


В этой заметке мы рассмотрим некоторые базовые приёмы работы с **oVirt**

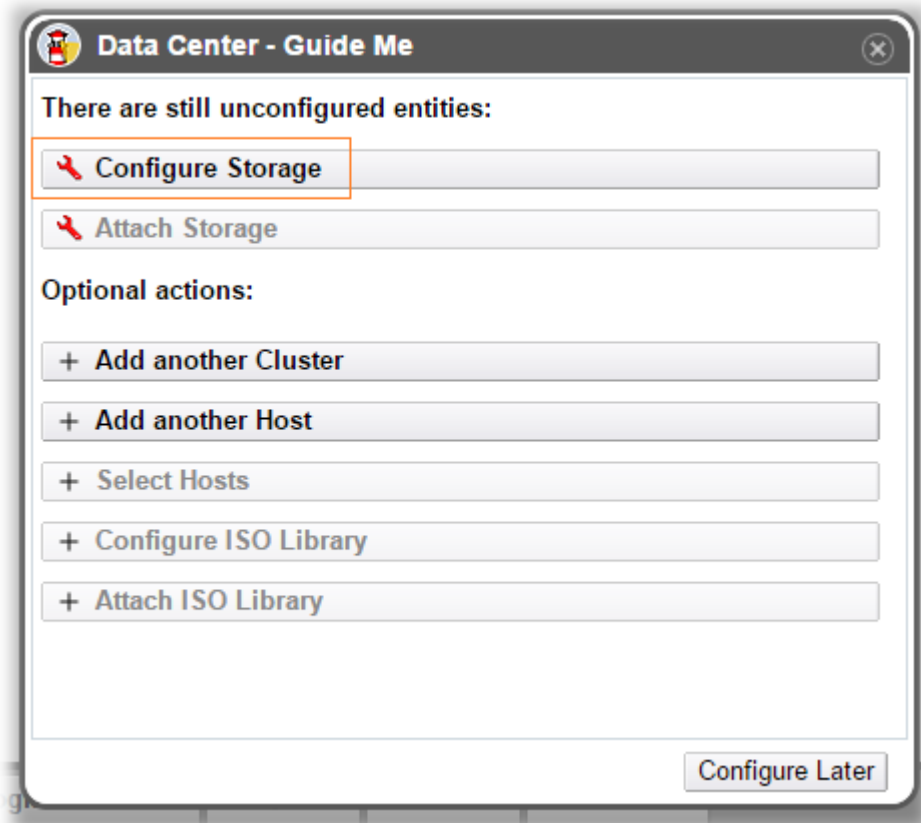
4.0 в конфигурации **Hosted Engine**, то есть основные действия, которые может потребоваться выполнить администратору после развёртывания oVirt. Чтобы начать работу с oVirt, нужно хотя бы на базовом уровне понимать значение компонент инфраструктуры этого продукта. Довольно простым и доступным языком описал основные компоненты oVirt Александр Руденко в своём блоге: [oVirt часть 4. Настройка инфраструктуры](#). Рекомендую к предварительному прочтению.

Инициализация Data Center

После того, как мы начнём ознакомление с веб-консолью портала администрирования oVirt на свежей инсталляции, мы можем обнаружить, что в консоли не отображается информация о присутствии виртуальной машины с oVirt Hosted Engine, а также о дисковом разделе из FC SAN на СХД, на который была установлена эта виртуальная машина. Как я понял, причина заключается в том, что наш **Data Center** (верхний уровень иерархии oVirt) не проинициализирован. Для того, чтобы выполнить его инициализацию, можно воспользоваться функцией подсказки **Guide Me** на панели кнопок вкладки **Data Centers** или в контекстном меню нашего дата-центра по умолчанию **Default**, который был автоматически создан в процессе развёртывания oVirt.



Инициализация Дата-Центра невозможна без предоставления в распоряжение oVirt дискового хранилища, на котором будут располагаться создаваемые виртуальные машины. Поэтому, в первую очередь, в открывшемся окне выбираем пункт **Configure Storage**



Откроется форма добавления хранилища к oVirt. Так как мы намереваемся подключить дисковое хранилище для размещения виртуальных машин, выберем в поле **Domain Function** значение **Data**. Тип хранилища **Storage Type** выбираем тот, который у нас планируется использовать (в нашем случае это **Fibre Channel**). В поле **Use Host** выбираем хост, с которого будет произведена проверка доступа к хранилищу и зададим понятное нам имя хранилища в поле Name. Поле **Description** по моим наблюдениям в **oVirt 4.0.1** при добавлении FC Data Domain не сохраняется, возможно этот недочёт исправят в будущих версиях. После выбора хоста в табличной части формы появится список доступных на этом хосте **LUN**-ов. Малый LUN размером 90GB в нашем случае уже используется под виртуальную машину Hosted Engine, поэтому здесь мы выбираем второй свободный LUN размером 2000GB.

New Storage

Data Center: Default Name: 3PAR-VOLUME2

Domain Function: Data Description: oVirt VM Storage

Storage Type: Fibre Channel Comment:

Use Host: KOM-AD01-VM31

<input type="checkbox"/> LUN ID	Dev. Size	Additional Size	#path	Vendor ID	Product ID	Serial
<input checked="" type="checkbox"/> 360002ac00000000	2000GB		4	3PARdata	VV	S3PARdataVV_168
<input type="checkbox"/> 360002ac00000000	90GB		4	3PARdata	VV	S3PARdataVV_168

Advanced Parameters

Warning Low Space Indicator (%) 10

Critical Space Action Blocker (GB) 5

Format V3

Wipe After Delete ☐

OK Cancel

Через несколько секунд после того, как я добавил в oVirt новый чистый FC LUN для хранения виртуальных машин, в веб-консоли появилась информация и о LUN-е под виртуальную машину oVirt Engine, а статус Дата-Центра изменился на **Up**

DataCenter: name = Default

Data Centers Clusters Hosts Networks Storage Disks Virtual Machines

Edit Guide Me

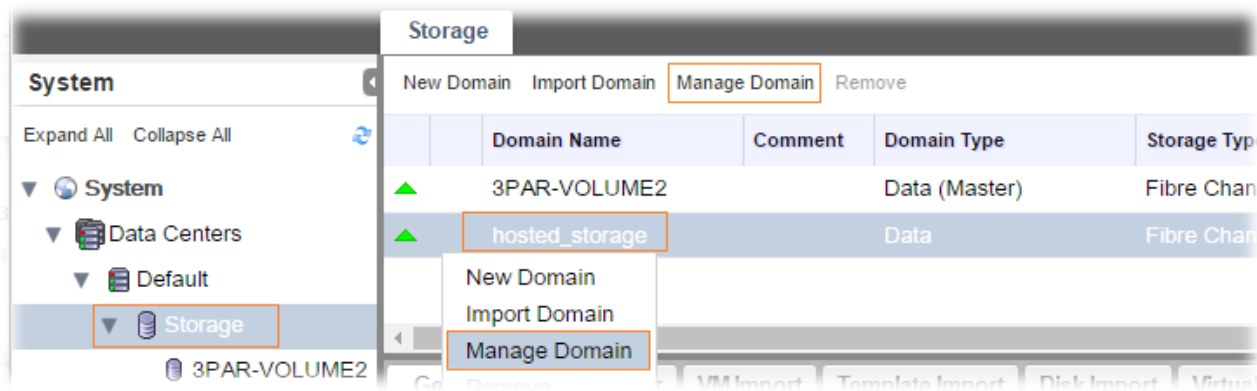
Name	Comment	Storage Type	Status	Compatibility Version	Description
Default		Shared	Up	4.0	The default

Storage Logical Networks QoS Clusters Permissions

Attach Data Attach ISO Attach Export Detach Activate Maintenance

Domain Name	Domain Type	Status	Free Space	Used Space	Total Space
hosted_storage	Data	Active	44 GB	45 GB	89 GB
3PAR-VOLUME2	Data (Master)	Active	1995 GB	4 GB	1999 GB

Кстати, присваиваемое по умолчанию имя Data Domain "**hosted_storage**" можно при желании переименовать (в **System > Data Centers > Default > Storage** – выбрать имя нужного **Data Domain** и нажать **Manage Domain**)



После инициализации Дата-Центра в веб-консоли должна появиться информация и о работающей виртуальной машине Hosted Engine.

Подключение ISO Domain

После первичной инициализации Дата-Центра и Кластера указанное в процессе развёртывания oVirt Engine хранилище NFS, предназначенное для хранения ISO-образов (**ISO Domain**), в веб-консоли может быть не видно. Если так, то в первую очередь на BM Engine стоит проверить содержимое файла `/etc/exports.d/ovirt-engine-iso-domain.exports`. При необходимости можно подправить этот файл, например так:

```
# cat /etc/exports.d/ovirt-engine-iso-domain.exports

# /var/nfs-ovirt-iso-share/files
10.1.0.0/24(rw, sync, no_root_squash, no_subtree_check)
/var/nfs-ovirt-iso-share/files KOM-AD01-
VM3*.holding.com(rw, sync, no_subtree_check, all_squash, anonuid=36, anongid=36)
```

После правки файла экспорта, заставим NFS сервер пересчитать новые параметры:

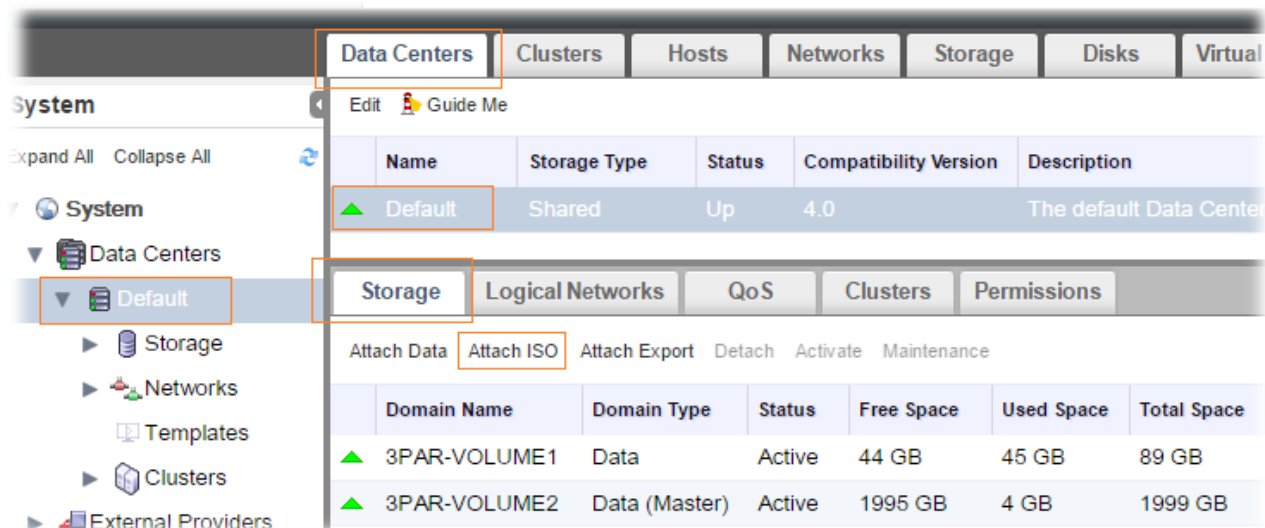
```
# exportfs -rv
```

После этого проверим, какие шары может отдавать NFS-сервер:

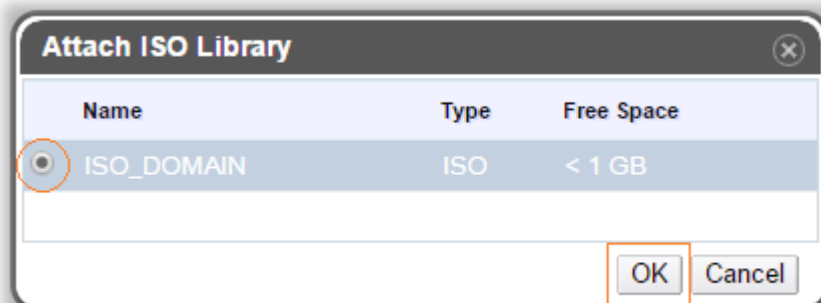
```
# exportfs

/var/nfs-ovirt-iso-share/files
KOM-AD01-VM3*.holding.com
```

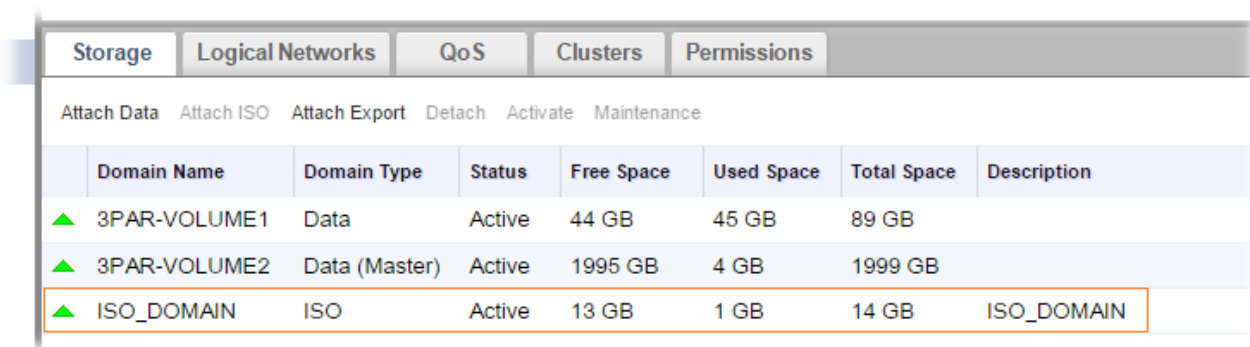
Будем считать, что наш NFS сервер настроен так как надо, и теперь можно попробовать присоединить ISO Domain в oVirt. Для этого в веб-консоли переходим к **System > Data Centers > Default >** на закладке **Storage** жмём кнопку **Attach ISO**



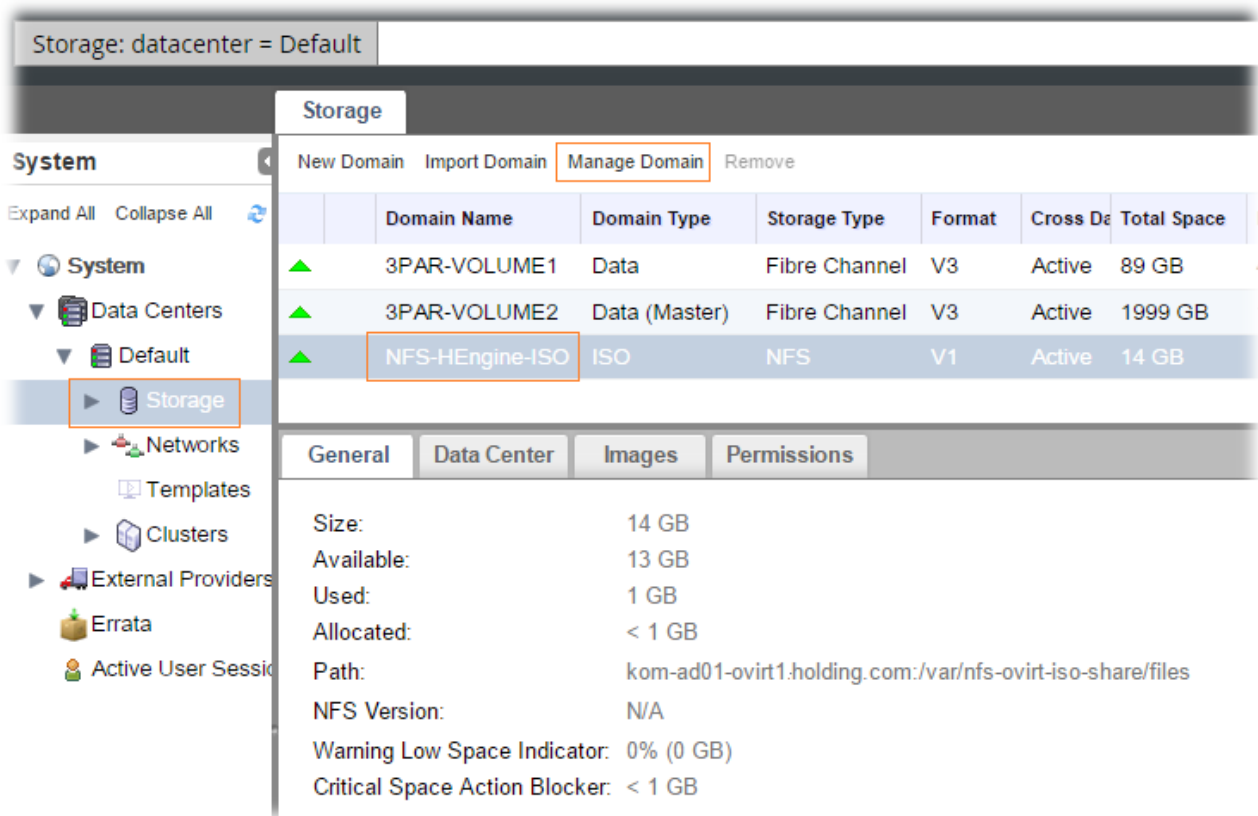
В открывшейся форме выбираем ранее описанный ISO Domain



После этого в веб-консоли нам станет доступно это хранилище для размещения ISO-образов.



Опять же, при необходимости можно переименовать только что присоединённый ISO Domain. Сделать это можно в **System > Data Centers > Default > Storage** - выбрать нужный ISO Domain и нажать **Manage Domain**. Для большей наглядности я переименую заданное в процессе развёртывания oVirt имя ISO_DOMAIN, например, в NFS-HEngine-ISO:



Загрузка ISO образов в ISO Domain

Возможности прямой загрузки ISO образов в ISO Domain через веб-интерфейс oVirt нет. Загрузить ISO образы можно непосредственно на виртуальном сервере Hosted Engine. Для примера я добавлю загрузочный инсталляционный образ Ubuntu Server 16.04 LTS. Как я понял, для этого действия предполагается иметь на локальной файловой системе сервера заранее скачанный ISO-образ и использовать утилиту **ovirt-iso-uploader** для его копирования в ISO Domain.

```
# wget http://releases.ubuntu.com/16.04/ubuntu-16.04.1-server-amd64.iso -P /tmp/
# ovirt-iso-uploader --iso-domain=NFS-HEngine-ISO upload /tmp/ubuntu-16.04.1-
server-amd64.iso --force
```

Please provide the REST API password for the admin@internal oVirt Engine user (CTRL+D to abort):

Uploading, please wait...

INFO: Start uploading /tmp/ubuntu-16.04.1-server-amd64.iso

Uploading: [#####] 100%

INFO: /tmp/ubuntu-16.04.1-server-amd64.iso uploaded successfully

```
# rm /tmp/ubuntu-16.04.1-server-amd64.iso
```

Хотя, в нашем случае, можно воспользоваться более простым вариантом добавления образов - загружая их из Интернета прямо в каталог, где ISO Domain хранит образы. Это каталог вида:

```
/var/nfs-ovirt-iso-share/files/<идентификатор>/images/11111111-1111-1111-1111-
111111111111/
```

```
# wget http://mirror.yandex.ru/centos/7/isos/x86_64/CentOS-7-x86_64-Minimal-1511.iso -P /var/nfs-ovirt-iso-share/files/b6773c29-5c80-4825-b345-9bf84ae48bdf/images/11111111-1111-1111-1111-111111111111/
# chmod 640 /var/nfs-ovirt-iso-share/files/b6773c29-5c80-4825-b345-9bf84ae48bdf/images/11111111-1111-1111-1111-111111111111/CentOS-7-x86_64-Minimal-1511.iso
# chown 36:36 /var/nfs-ovirt-iso-share/files/b6773c29-5c80-4825-b345-9bf84ae48bdf/images/11111111-1111-1111-1111-111111111111/CentOS-7-x86_64-Minimal-1511.iso
```

Обратите внимание на то, что после загрузки желательно выставить на файл разрешения по аналогии с тем, как их выставляет утилита **ovirt-iso-uploader**.

Теперь проверим состояние ISO Domain в веб-консоли oVirt (**System > Data Centers > Default > Storage** > выберем соответствующий ISO Domain и на закладке **Images** проверим доступность загруженных ранее ISO-образов)

The screenshot shows the oVirt web console interface. On the left, the 'System' menu is expanded, showing 'Data Centers' > 'Default' > 'Storage'. The 'Storage' tab is active, and the 'Images' sub-tab is selected. The main area displays a table of storage domains and a table of images.

Domain Name	Domain Type	Storage Type	Format	Cross D.	Total Sp
3PAR-VOLUME1	Data	Fibre Channel	V3	Active	89 GB
3PAR-VOLUME2	Data (Master)	Fibre Channel	V3	Active	1999 GB
NFS-HEngine-ISO	ISO	NFS	V1	Active	14 GB

File Name	Type	Actual Size
.keep	Unknown	0 Bytes
CentOS-7-x86_64-Minimal-1511.iso	ISO	603 MB
ubuntu-16.04.1-server-amd64.iso	ISO	667 MB

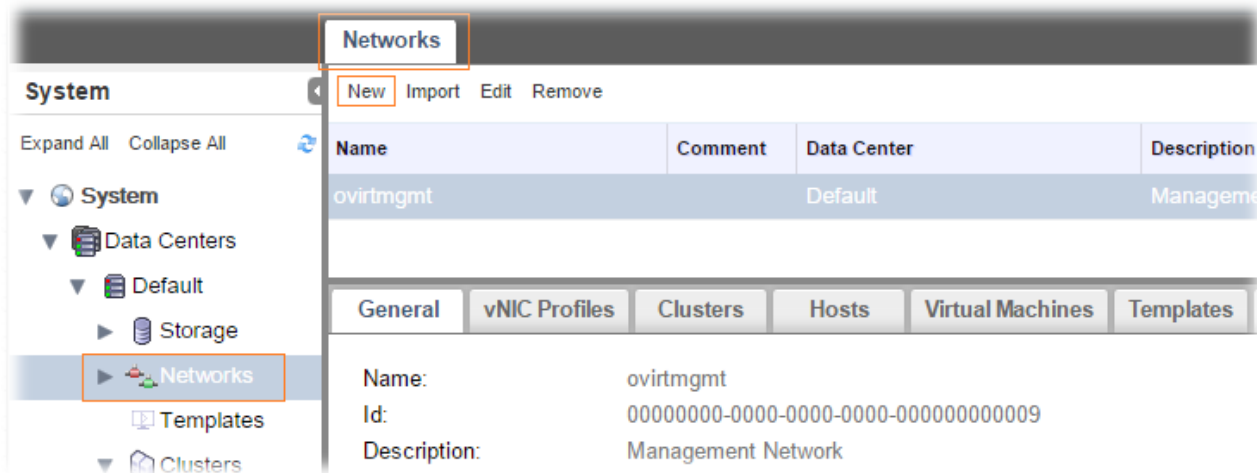
Как видим, оба образа отображаются и могут быть использованы в дальнейшем для подключения к виртуальным машинам.

Настройка сетей

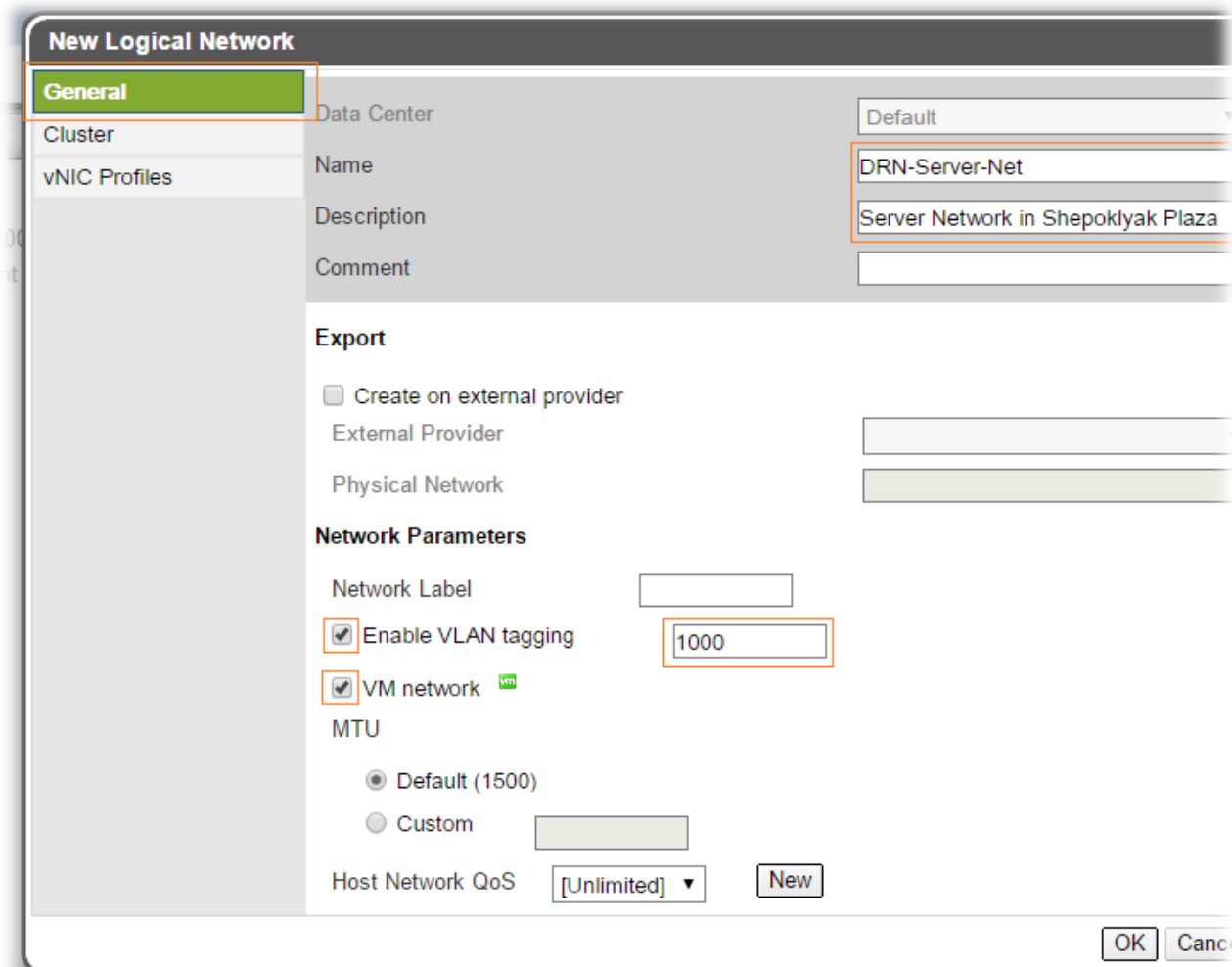
Создавать логические сети в oVirt имеет смысл в том случае, если, например, в вашей сети используется изоляция разнородного трафика с помощью **VLAN**.

Например, если физические сетевые интерфейсы наших хостов виртуализации подключены к транковым портам коммутаторов, имеющих доступ к множеству VLAN-ов, то с помощью создания виртуальных сетей с определённым VLAN с последующей привязкой этих сетей к сетевым адаптерам виртуальных машин, мы сможем организовать необходимое разделение трафика. В следующем примере мы создадим логическую сеть с номером VLAN 1000. Перейдём в веб-консоли в **System > Data Centers > Default > Networks**. Здесь мы увидим список доступных сетей нашего Дата-Центра. В данный момент здесь присутствует только сеть

ovirtmgmt, которая была автоматически создана в процессе развёртывания oVirt (к этой сети привязан vNIC виртуальной машины Hosted Engine). Чтобы создать новую сеть используем кнопку **New**



В открывшейся форме создания логической сети на вкладке **General** зададим понятное для нас имя с описанием, укажем номер VLAN-а и включим признак того, что эта сеть может использоваться для виртуальных машин – **VM network**



Так как создаваемые в oVirt логические сети - это объекты уровня Дата-Центра, а в Дата-Центре может быть не один Кластер, на вкладке **Cluster** мы можем при необходимости настроить привязку этих сетей к конкретным Кластерам. В данном случае мы не меняем предложенных по умолчанию настроек.

The screenshot shows the 'New Logical Network' dialog with the 'Cluster' tab selected. The 'Attach/Detach Network to/from Cluster(s)' section contains a table with the following data:

Name	Attach All	Required All
Default	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Профиль vNIC по умолчанию создаётся для сети с одноимённым названием и при желании можно изменить его имя.

The screenshot shows the 'New Logical Network' dialog with the 'vNIC Profiles' tab selected. The 'vNIC Profiles' section contains a table with the following data:

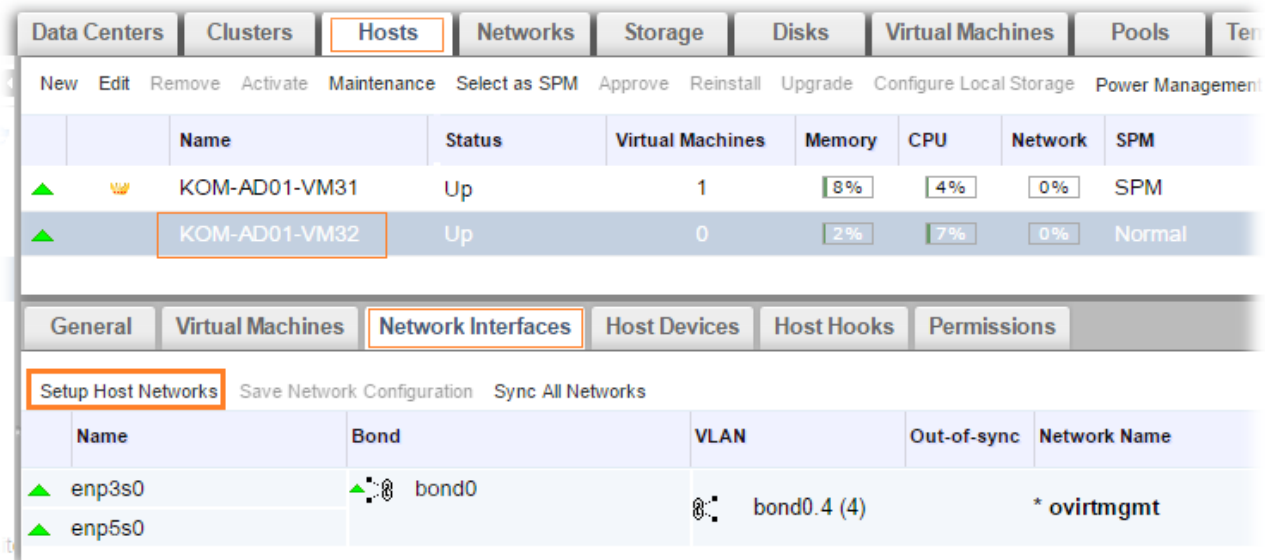
Name	Public	QoS
DRN-Server-Net-vNIC-Pr	<input checked="" type="checkbox"/>	[Unlimited]

Подробнее о том, для чего могут использоваться профили vNIC можно почитать [здесь](#). Если кратко, то с помощью множества профилей vNIC в рамках одной логической сети можно реализовать разные возможности виртуальных сетевых адаптеров, например, выбрать различные политики QoS (сами политики QoS настраиваются на уровне Дата-Центра), настроить различные опции сетевой фильтрации, включить Port Mirroring и т.д. Полные настройки профилей доступны в **System > Data Centers > Default > Networks** на закладке **vNIC Profiles** при выборе той или иной логической сети.

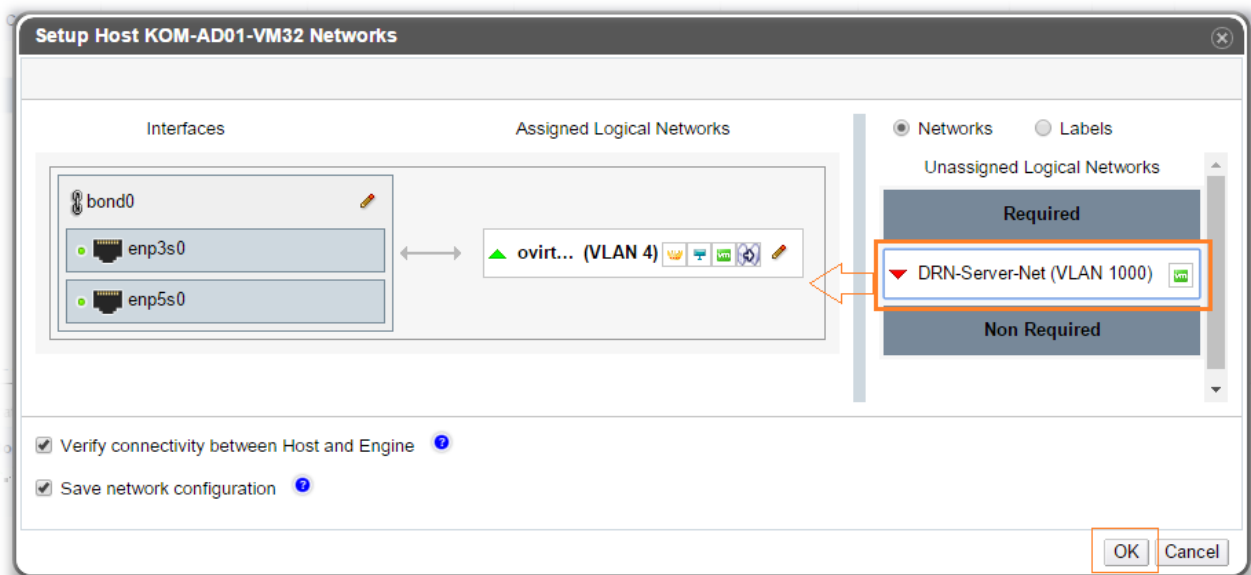
The screenshot shows the 'VM Interface Profile' dialog. The 'Network' dropdown is set to 'DRN-Server-Net'. The 'Name' field contains 'DRN-Server-Net-vNIC-Profile'. The 'QoS' dropdown is set to '[Unlimited]'. The 'Network Filter' dropdown is set to 'vdsim-no-mac-spoofing'. The 'Passthrough' and 'Port Mirroring' checkboxes are unchecked. The 'Please select a key...' dropdown is also visible.

После создания сети, для того чтобы виртуальные машины могли использовать подключение к этой сети, необходимо выполнить привязку этих сетей к физическим сетевым интерфейсам хостов. Перейдём в веб-консоли в **System > Data Centers >**

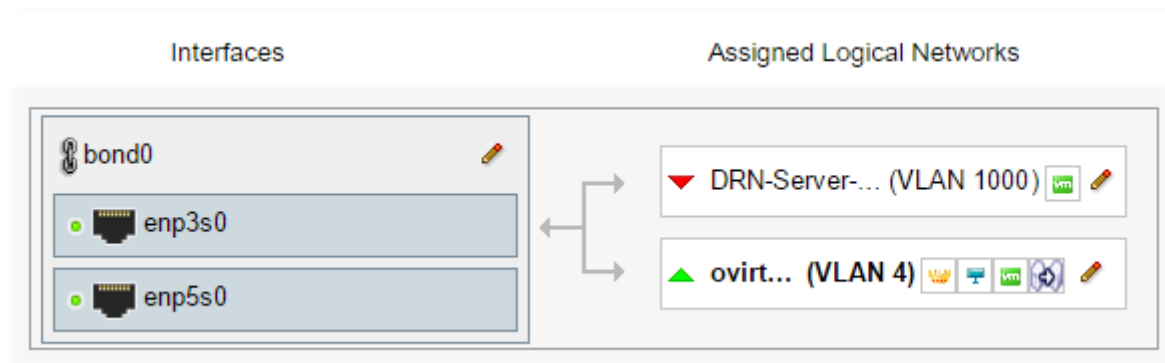
Default, выберем вкладку **Hosts**, выберем какой-нибудь хост, к которому хотим выполнить привязку сети, и в нижней части экрана перейдём на закладку **Network Interfaces**. Здесь мы увидим информацию о сетевых интерфейсах хоста. Нажмём кнопку **Setup Host Networks**.



Откроется форма привязки логических сетей к хосту. В этой форме справа будут перечислены все имеющиеся в oVirt, но не привязанные к хосту сети, а слева текущая сетевая конфигурация хоста. С помощью манипулятора, типа "мышь", перетаскиваем интересующую нас сеть на интерфейс хоста...



В нашем случае используется привязка ранее созданной сети с VLAN 1000 к интерфейсу bond0:



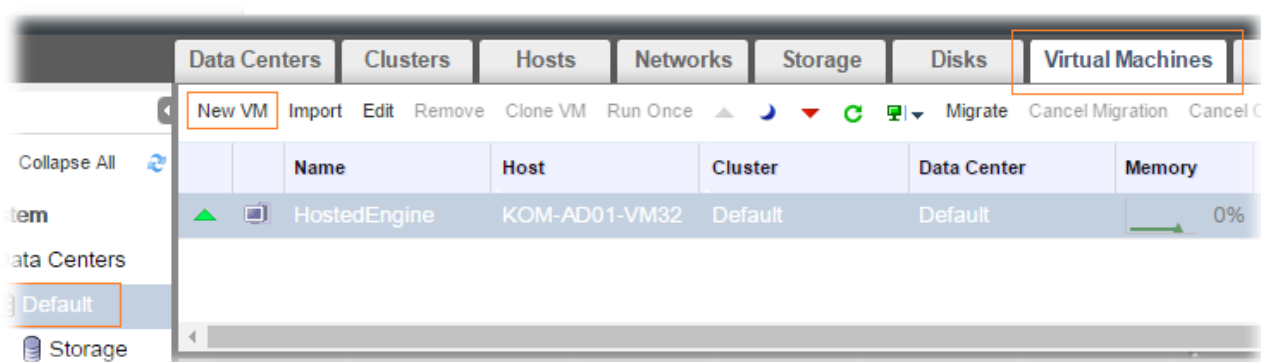
После этого на хосте будет создан интерфейс с соответствующим номером VLAN и в веб-консоли эта информация будет отображена соответственно:

KOM-AD01-VM32 Up 1 4% 5% 0% SPM					
General Virtual Machines Network Interfaces Host Devices Host Hooks Permissions					
Setup Host Networks Save Network Configuration Sync All Networks					
Name	Bond	VLAN	Out-of-sync	Network Name	
enp3s0	bond0	bond0.4 (4)		* ovirtmgmt	
enp5s0		bond0.1000 (1000)		DRN-Server-Net	

Аналогичным образом нужно выполнить привязку созданной нами логической сети ко всем хостам, где могут оказаться, например в процессе миграции, виртуальные машины, привязанные к данной сети.

Создание новой виртуальной машины

Создадим новую виртуальную машину и установим на неё ОС **Ubuntu Server 16.04 LTS** с ранее загруженного в ISO Domain установочного образа. Переходим в **System > Data Centers > Default > закладка Virtual Machines > кнопка New VM**.



В открывшейся веб-форме создания виртуальной машины сразу нажимаем кнопку **Show Advanced Options**, чтобы включить расширенный режим настроек.

Назначение полей нетрудно понять из их названий. Задаём тип гостевой ОС VM, имя VM, так как оно будет отображаться в oVirt, описание. Для виртуального

интерфейса **nic1** выбираем ранее созданный **vNIC Profile**, который в свою очередь привязан в ранее созданной логической сети. Чтобы создать новый диск для виртуальной машины используем кнопку **Create**

New Virtual Machine

General

Cluster: Default
Data Center: Default

Template: Blank | (0)

Operating System: Linux

Instance Type: Custom

Optimized for: Server

Name: KOM-AD01-PBX02

VM ID:

Description: FreePBX Server (Asterisk)

Comment:

☐ Stateless ☐ Start in Pause Mode ☐ Delete Protection

Instance Images: Attach **Create** + -

Instantiate VM network interfaces by picking a vNIC profile.

nic1: DRN-Server-Net-vNIC-Profile/DRN- + -

Hide Advanced Options OK Cancel

Поверх формы создания виртуальной машины откроется форма создания нового виртуального диска, где мы укажем его размер, понятное имя с описанием, выберем хранилище для размещения файла диска (**Storage Domain**) и тип распределения данных (**Allocation Policy**). Не забудем включить опцию **Bootable**, так планируем устанавливать на этот диск ОС.

New Virtual Disk

☒ Image ☐ Direct LUN ☐ Cinder

Size(GB)

Alias

Description

Interface

Storage Domain

Allocation Policy

Disk Profile

☐ Wipe After Delete

☒ Bootable

☐ Shareable

☐ Read Only

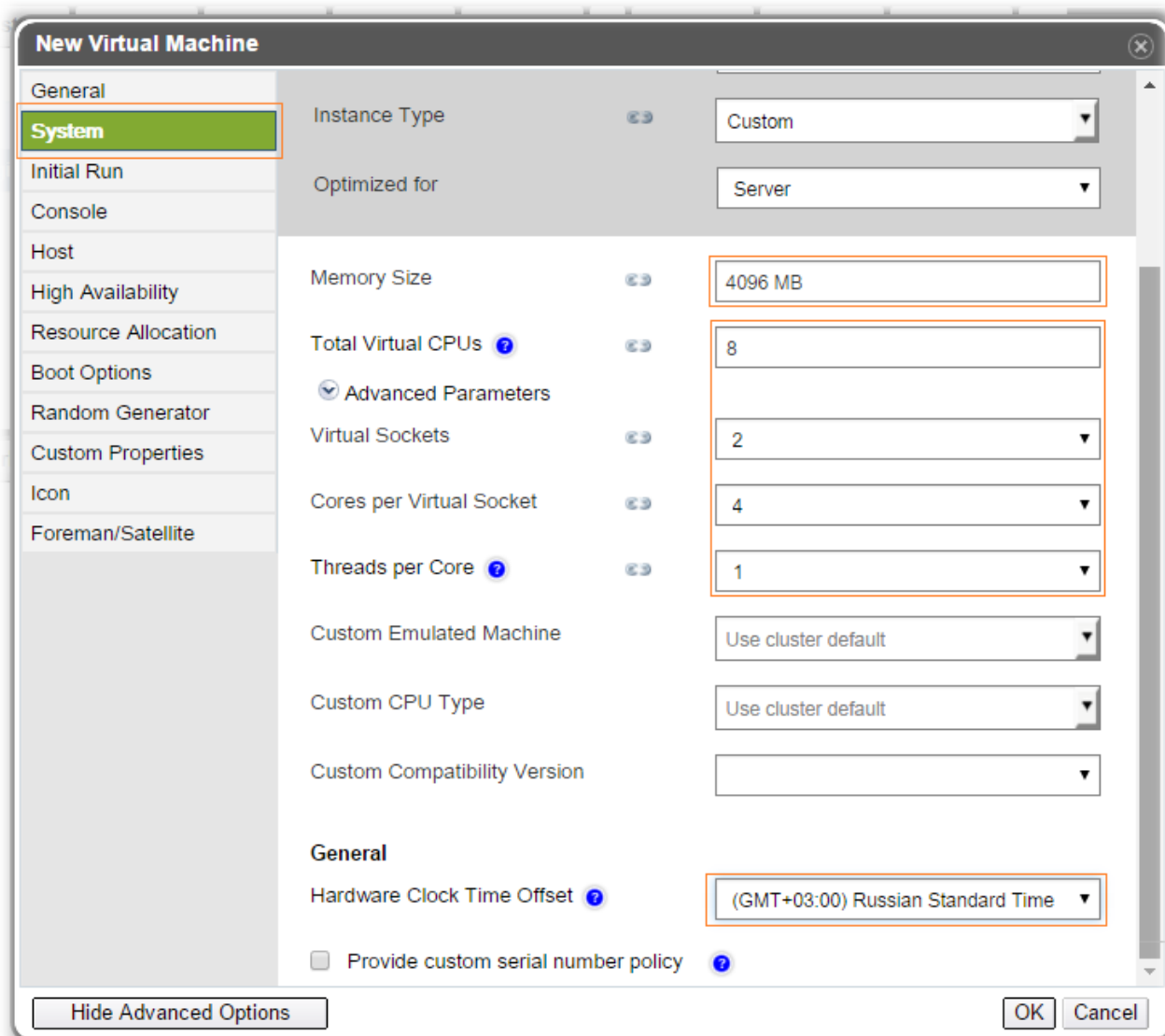
Таким образом, для примера, я создал первый диск (под гостевую ОС) и дополнительный (под данные приложений).

Instance Images

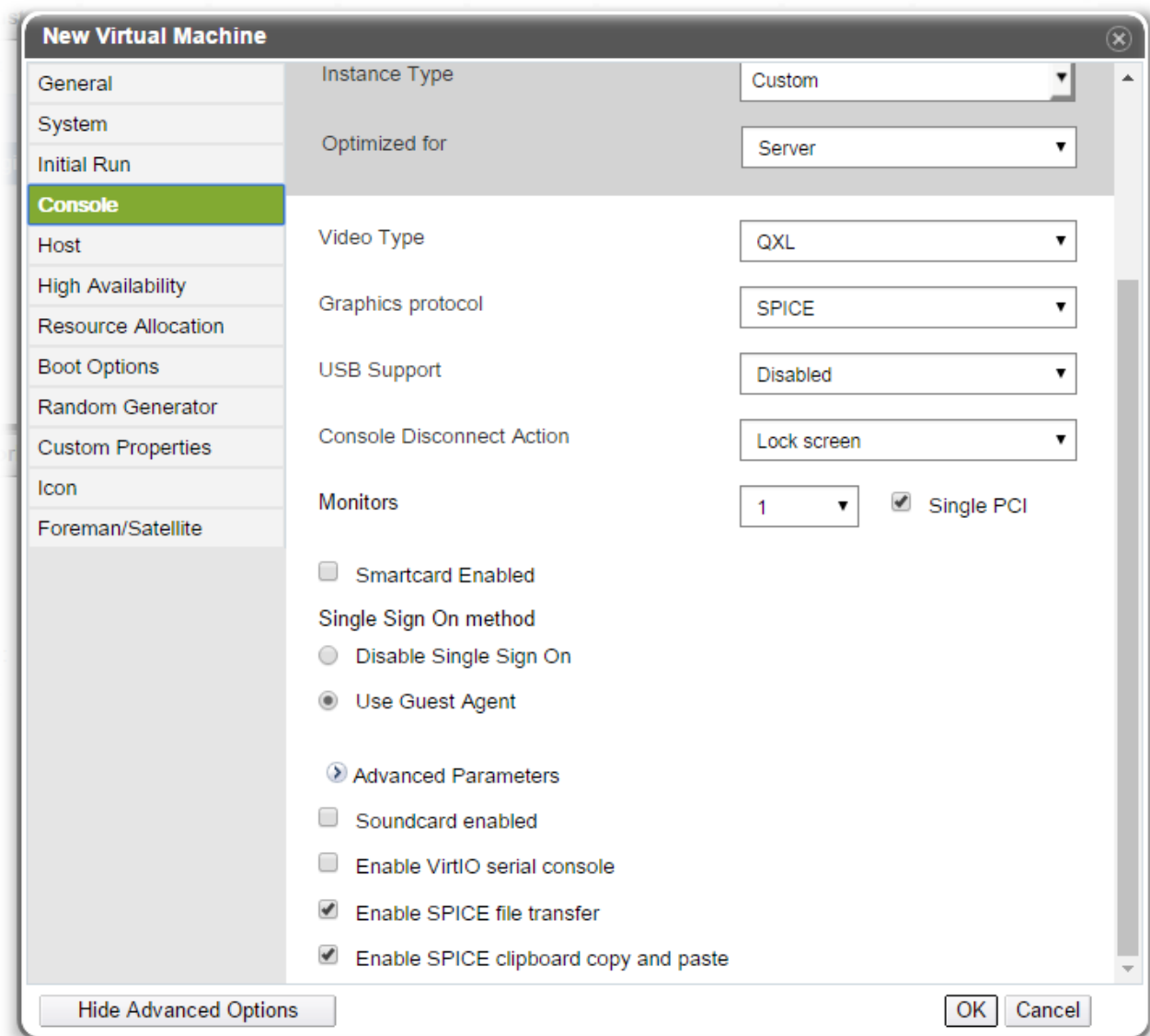
KOM-AD01-PBX02_Disk1: (40 GB) creating (boot)

KOM-AD01-PBX02_Disk2: (20 GB) creating

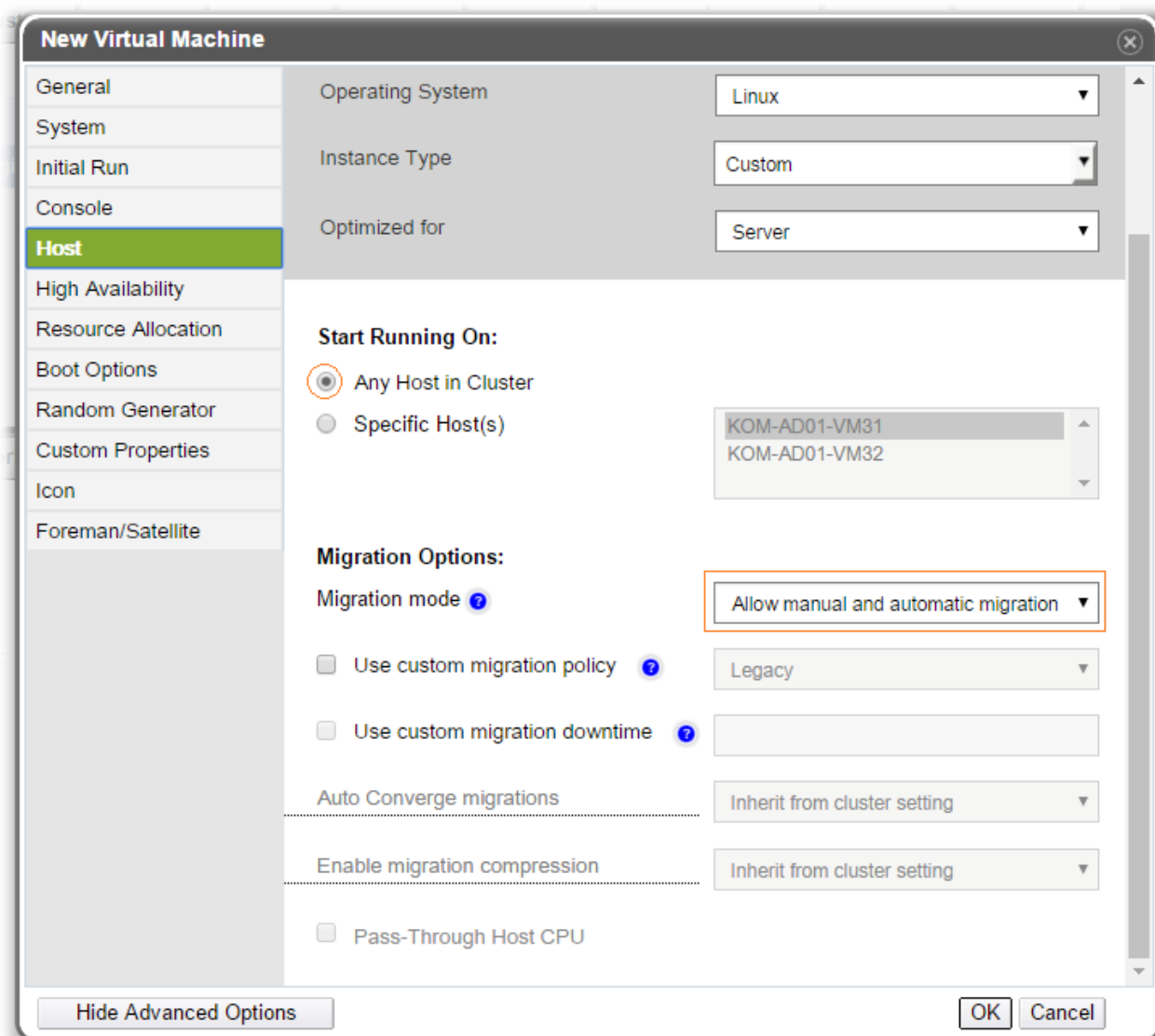
Переходим на вкладку **System** и настраиваем параметры оперативной памяти, процессора, часовой пояс:



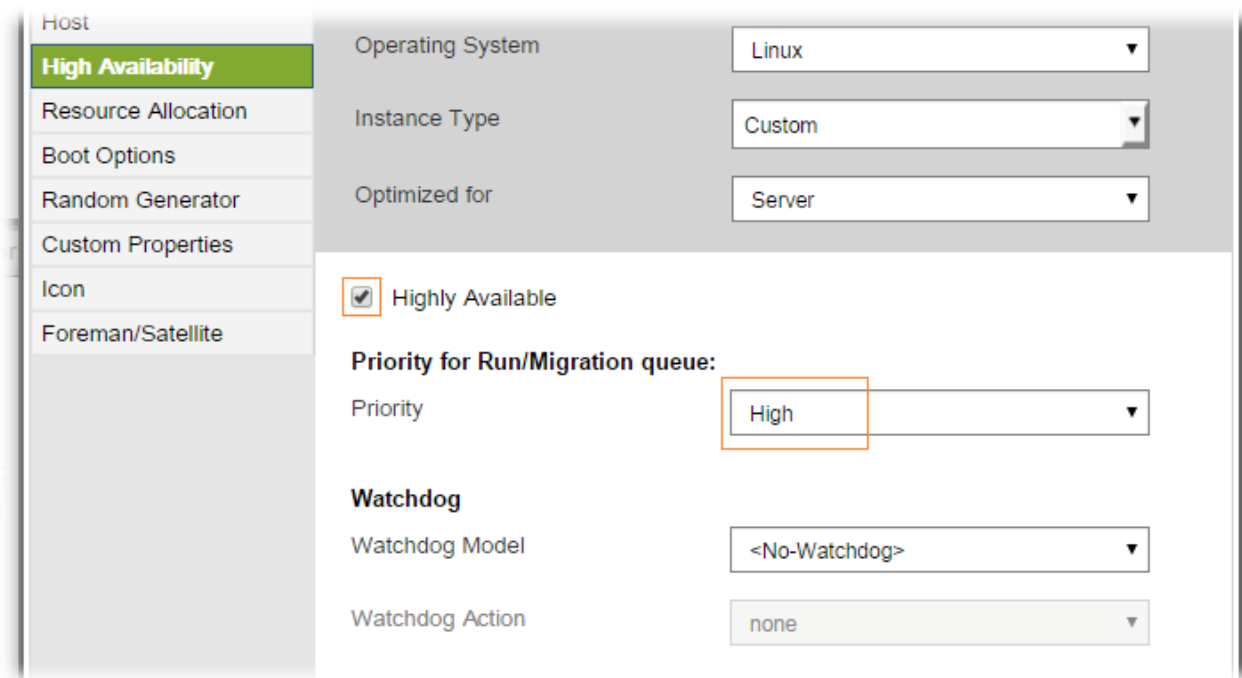
На вкладке **Console** выбираем параметры подключения к консоли виртуальной машины. В данном случае нам доступны два основных графических протокола **VNC** и **SPICE** (к особенностям их использования мы вернёмся чуть позже), так как в свойствах ОС выбран тип **Linux**.



На вкладке **Host** при необходимости можно настроить привязку виртуальной машины к какому-то конкретному хосту в кластере, а также изменить параметры миграции VM между хостами.



На вкладке **High Availability** можно настроить приоритет перезапуска виртуальной машины в контексте выполнения операций по автоматическому восстановлению доступности виртуальных машин в кластере.



Host

High Availability

Resource Allocation

Boot Options

Random Generator

Custom Properties

Icon

Foreman/Satellite

Operating System: Linux

Instance Type: Custom

Optimized for: Server

☒ Highly Available

Priority for Run/Migration queue:

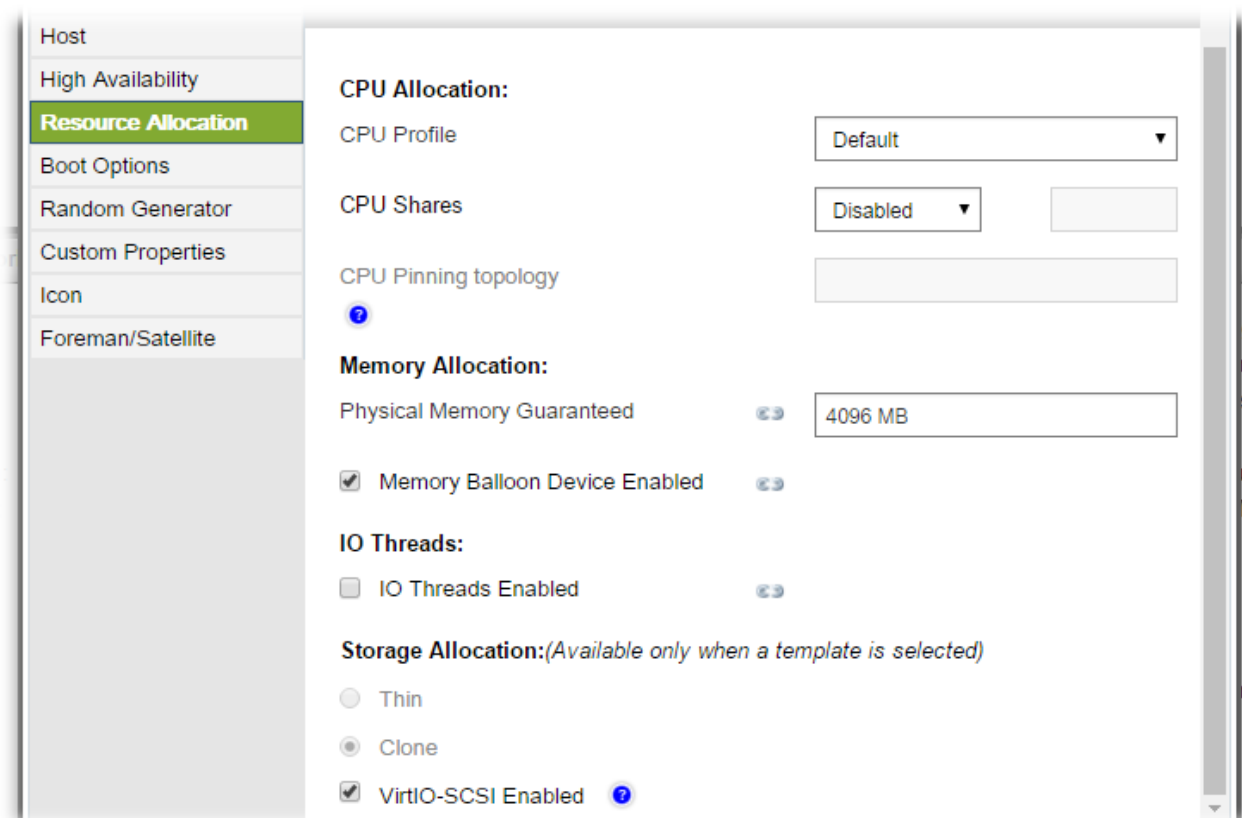
Priority: High

Watchdog

Watchdog Model: <No-Watchdog>

Watchdog Action: none

На вкладке **Resource Allocation** настраиваются дополнительные опции, которыми можно контролировать производительность той или иной виртуальной машины, путём определения порядка выделения физических ресурсов этой машине.



Host

High Availability

Resource Allocation

Boot Options

Random Generator

Custom Properties

Icon

Foreman/Satellite

CPU Allocation:

CPU Profile: Default

CPU Shares: Disabled

CPU Pinning topology

Memory Allocation:

Physical Memory Guaranteed: 4096 MB

☒ Memory Balloon Device Enabled

IO Threads:

☐ IO Threads Enabled

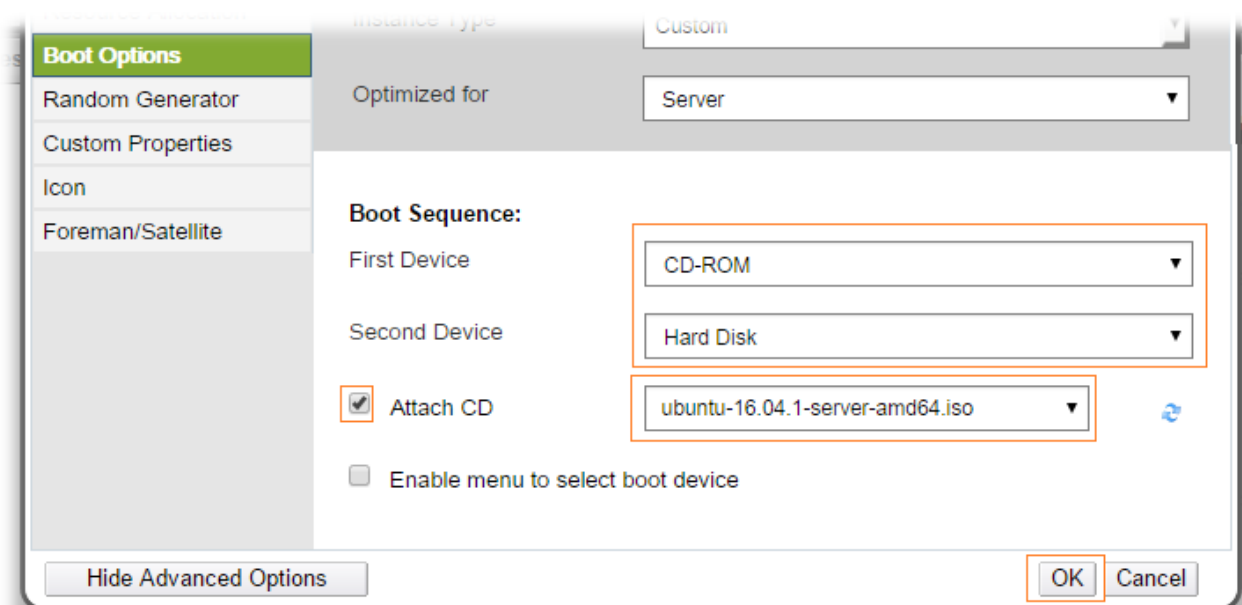
Storage Allocation: (Available only when a template is selected)

☐ Thin

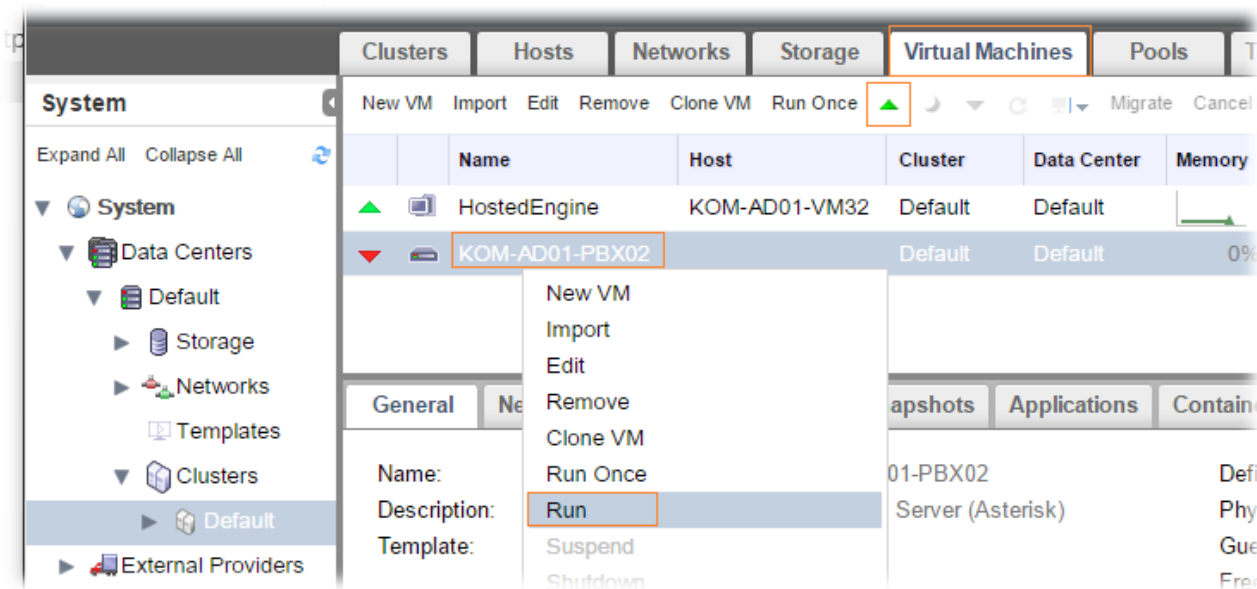
☒ Clone

☒ VirtIO-SCSI Enabled

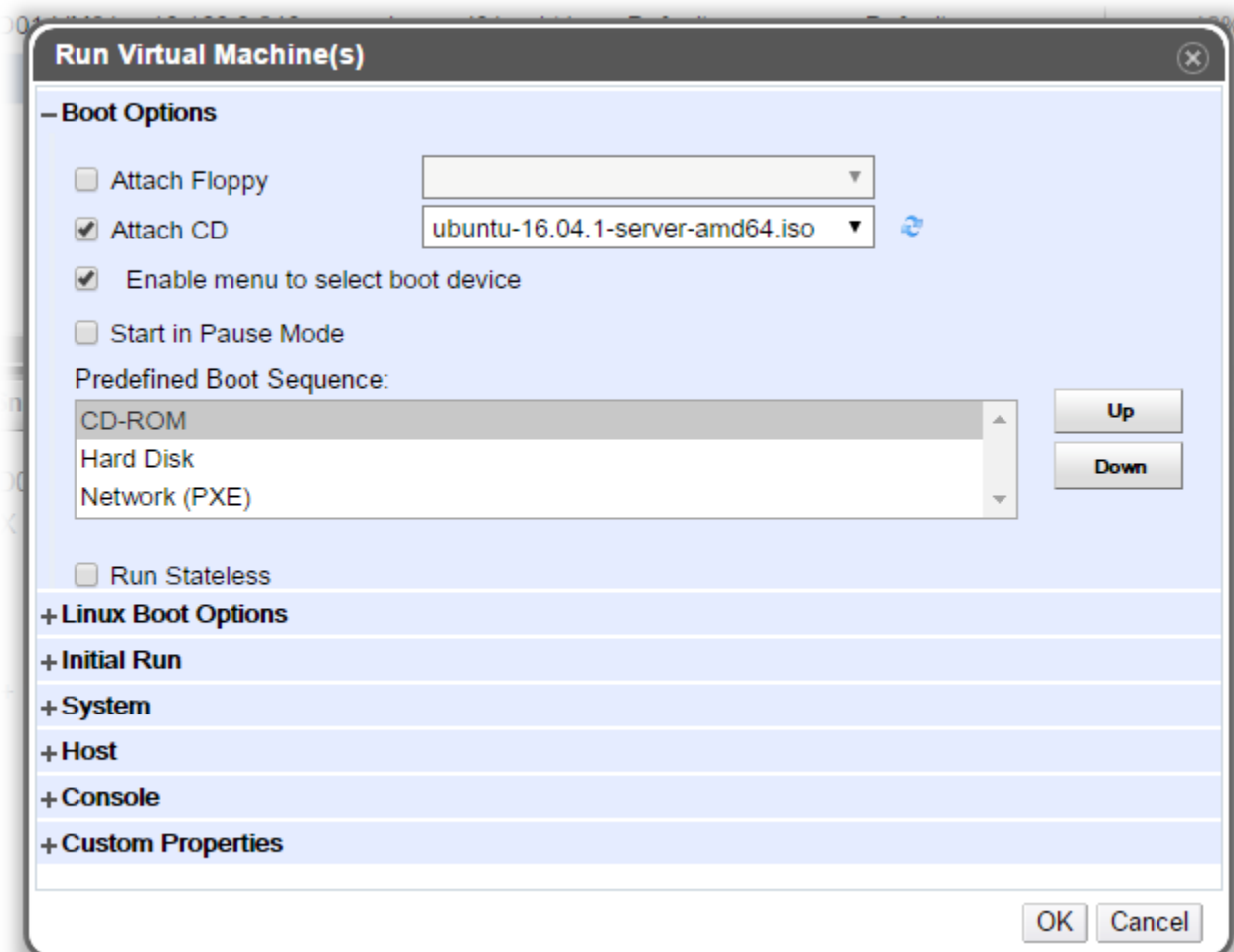
На вкладке **Boot Options** определяем порядок использования устройств для загрузки. Здесь мы можем присоединить к виртуальному **CD-ROM** ранее загруженный в **ISO Domain** установочный ISO-образ операционной системы Linux, которую мы планируем установить на диск VM.



Сохраняем настроенные свойства виртуальной машины и пробуем выполнить её запуск из консоли используя пункты меню или кнопки **Run** или **Run Once**.



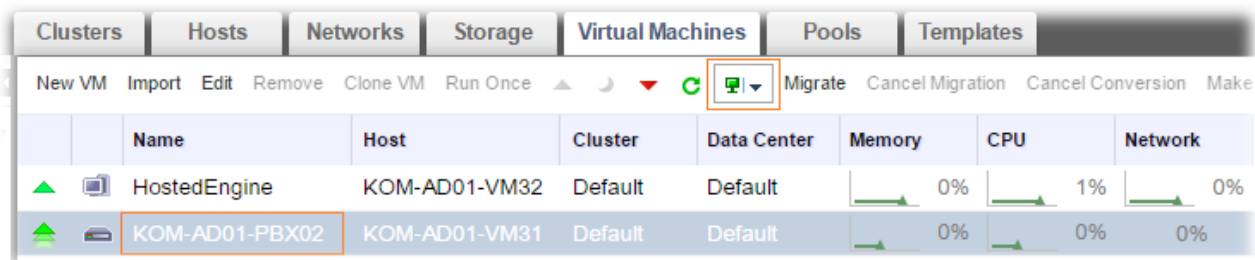
Функция **Run Once** позволяет выполнять запуск виртуальной машины с изменением ряда её параметров, которые не сохраняются в конфигурации этой VM при следующем перезапуске, например, именно здесь можно указать единожды монтируемый ISO образ, который нужен для установки ОС и уже не потребуется при следующей перезагрузке VM.



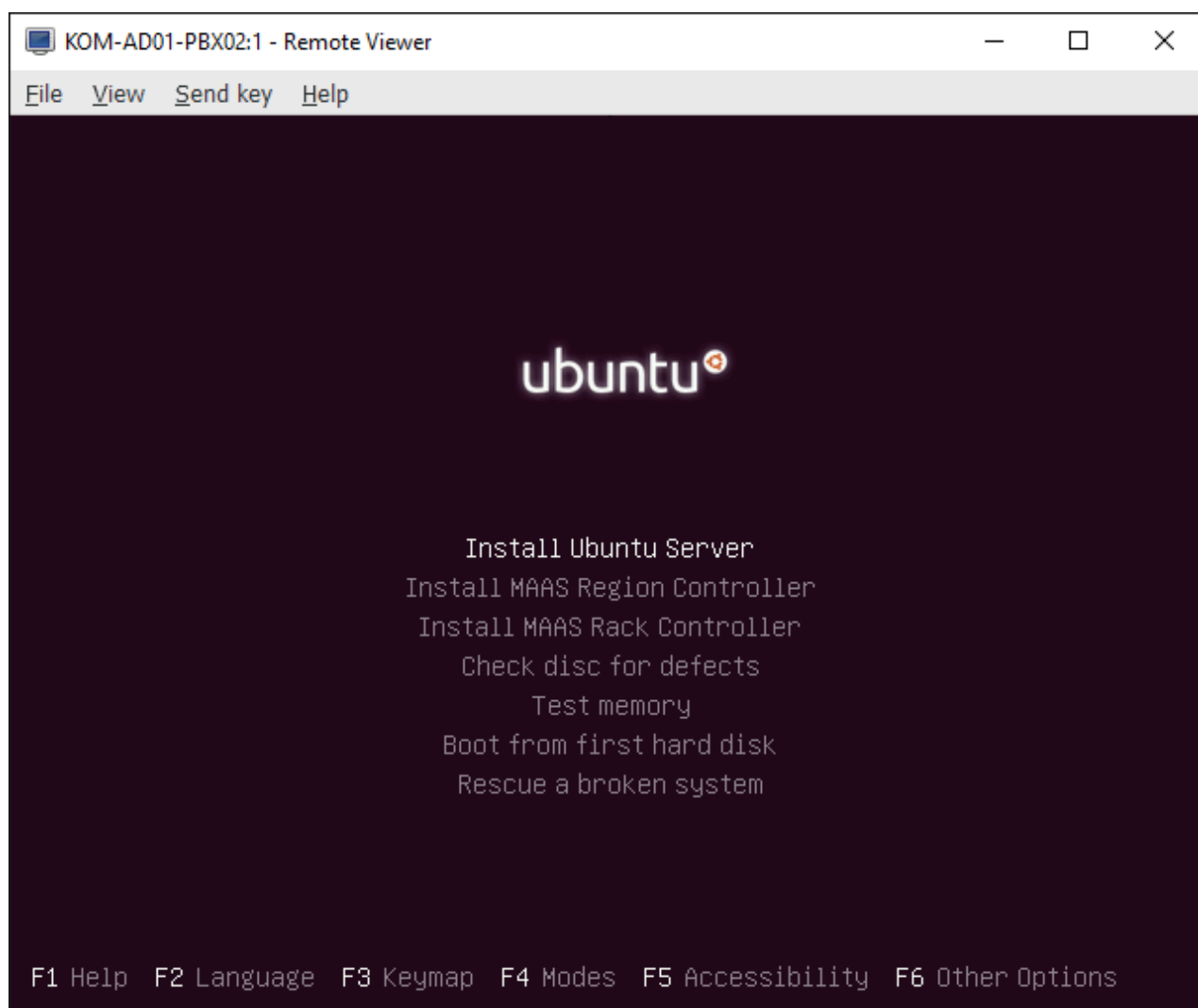
Подключение к консоли виртуальных машин

После того, как виртуальная машина запущена, мы сможем подключиться к её консоли разными способами. Поддерживаемые варианты подключения к консоли виртуальной машины описаны в онлайн-документе [Console Clients Resources](#). На данный момент поддерживаются 3 протокола **SPICE**, **VNC** и **RDP**. Протокол RDP можно использовать только для виртуальных машин с гостевой ОС Windows. Протокол VNC требует наличия VNC-клиента. Протокол SPICE может работать как через Native SPICE-клиент, так и как HTML5-апплет.

В качестве **Native SPICE**-клиента можно использовать, например, программу **Virt-Viewer**. Все доступные для загрузки версии можно найти [здесь](#). Если установка Virt-Viewer планируется на Windows, то на данный момент [здесь рекомендуют](#) использовать версию 3 вместо 4-ой, так как в последней имеются какие-то проблемы. Скачиваем и устанавливаем Virt-Viewer, после чего в консоли oVirt, выбрав нужную нам виртуальную машину, нажимаем кнопку с зелёным монитором для подключения к консоли ВМ.



Будет скачан файл **console.vv** и запущен в ассоциированном приложении Virt-Viewer, после чего мы получим доступ к консоли VM:

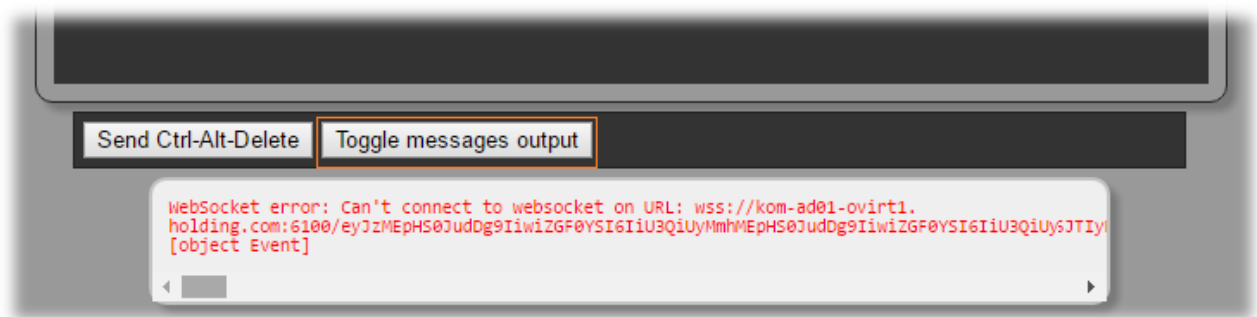


При попадании курсора в консоль, управление курсором переходит в виртуальную машину. Для возврата курсора обратно используем сочетание клавиш **Shift+F12**.

Помимо Native SPICE-клиента можно использовать возможности **HTML5 SPICE**-клиент, то есть консоль виртуальной машины при этом будет открываться непосредственно в интернет-браузере. Тесты показали, что в **Internet Explorer v11** HTML5 SPICE-клиент не работает. В **Firefox** и **Chrome** работает, но не всегда стабильно и при условии, что SSL сертификат, используемый в веб-консоли oVirt является доверенным (этого можно добиться либо добавив само-подписанный сертификат oVirt в хранилище доверенных сертификатов пользователя в системе,

либо заменить сертификат на веб-сайте oVirt, так как мы сделали это [ранее](#)). Однако не смотря на то, что даже если требование доверия браузера сертификату соблюдено мы можем получить ошибку типа:

```
WebSocket error: Can't connect to websocket on URL: wss://kom-ad01-ovirt1.holding.com:6100
```



В [прошлой заметке](#) (в последнем абзаце) был описан способ решения этой проблемы, подсказанный в [мейл-группе oVirt](#). Напомню, что смысл его в том, чтобы откорректировать конфигурационный файл `/etc/ovirt-engine/ovirt-websocket-proxy.conf.d/10-setup.conf`, изменив в нём значение переменных **SSL_CERTIFICATE** и **SSL_KEY**.

Для более комфортной работы при подключении по протоколу SPICE, в гостевую ОС виртуальной машины можно дополнительно установить такие компоненты, как агент **SPICE vdagent** и видео драйвер **SPICE QXL**. Загрузить их можно по ссылке [SPICE Download](#) (секция "**Guest**"). В некоторых дистрибутивах GNU/Linux видеодрайвер SPICE QXL уже входит в базовый состав ОС.

Установка компонент интеграции в виртуальные машины

Основную информацию о гостевых агентах oVirt можно найти в документе [Understanding Guest Agents and Other Tools](#). В этом же документе можно найти таблицу совместимости компонент интеграции с разными типами гостевых систем Linux и Windows.

Компоненты интеграции гостевых ОС в среду виртуализации oVirt состоят из дополнительно устанавливаемых в гостевую ОС агентов **oVirt Guest Agent**, **SPICE Agent** и драйверов **VirtIO Drivers**.

oVirt Guest Agent

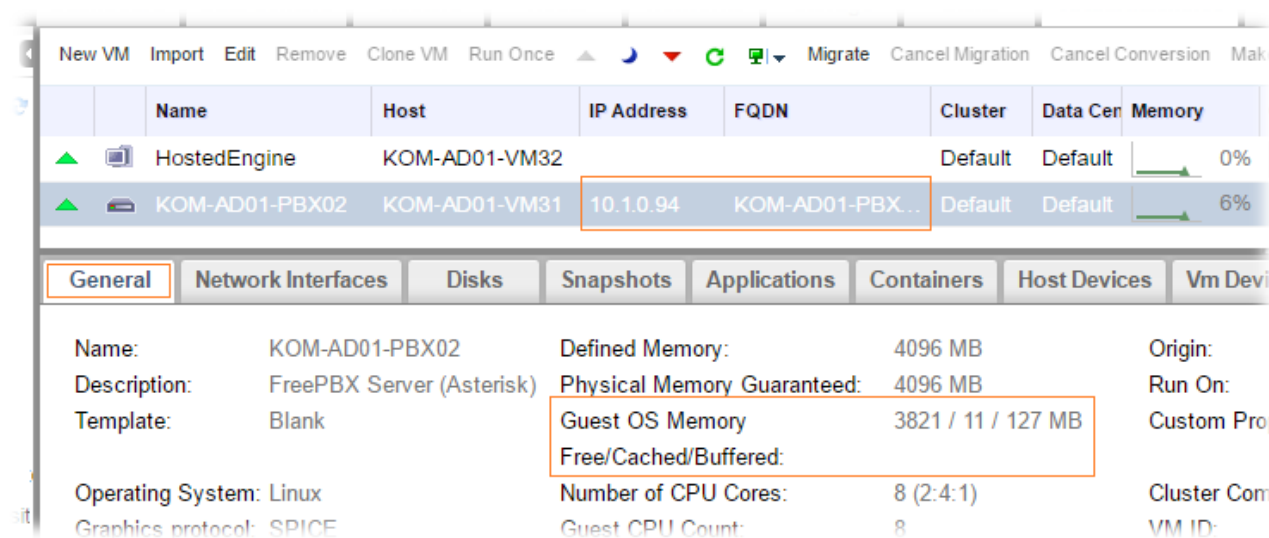
Информация о том, какие преимущества даёт установка **oVirt Guest Agent** доступна в документе [Guest Agent](#). Примеры установки гостевого агента на разные дистрибутивы Linux есть в серии [How to install the guest agent in](#) Например, в нашей виртуальной машине с гостевой ОС **Ubuntu 16.04 LTS** агент установился из официальных репозиториев, однако после его установки потребовалась корректировка прав на каталог для логов службы **ovirt-guest-agent**, иначе служба отказывалась стартовать:

```
$ sudo apt-get install ovirt-guest-agent
$ sudo chown ovirtagent:ovirtagent -R /var/log/ovirt-guest-agent
$ sudo systemctl enable ovirt-guest-agent
$ sudo reboot
$ sudo service ovirt-guest-agent status
```

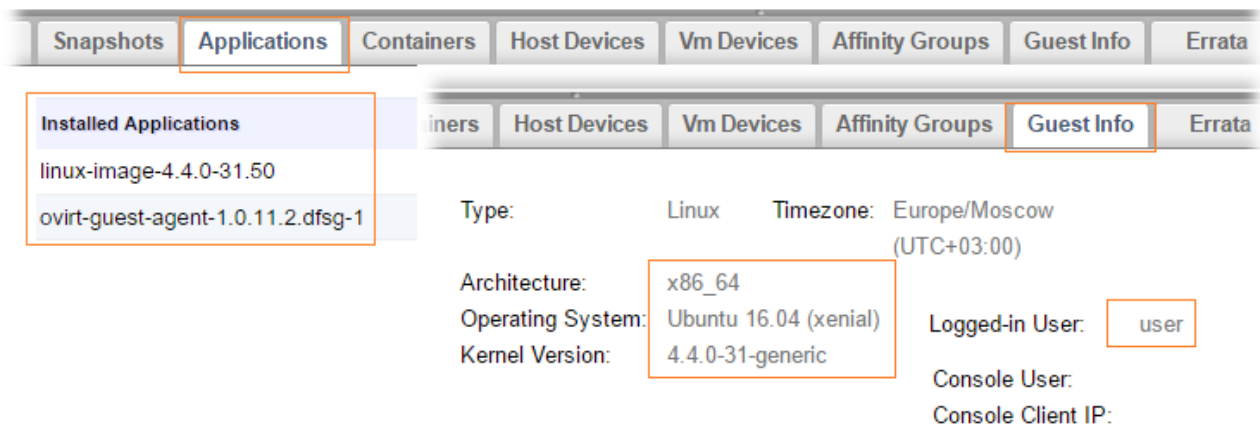
```
user@KOM-AD01-PBX02:~$ sudo service ovirt-guest-agent status
[sudo] password for user:
● ovirt-guest-agent.service - oVirt Guest Agent
   Loaded: loaded (/lib/systemd/system/ovirt-guest-agent.service; enabled; vendor preset: e
   Active: active (running) since Sat 2016-07-30 15:19:07 MSK; 54s ago
     Process: 2678 ExecStartPre=/bin/chown ovirtagent:ovirtagent /run/ovirt-guest-agent.pid (c
     Process: 2668 ExecStartPre=/bin/touch /run/ovirt-guest-agent.pid (code=exited, status=0/S
     Process: 2627 ExecStartPre=/sbin/modprobe virtio_console (code=exited, status=0/SUCCESS)
    Main PID: 2700 (python)
      Tasks: 3
     Memory: 63.2M
        CPU: 987ms
    CGroup: /system.slice/ovirt-guest-agent.service
            └─2700 /usr/bin/python /usr/share/ovirt-guest-agent/ovirt-guest-agent.py

Jul 30 15:19:06 KOM-AD01-PBX02 systemd[1]: Starting oVirt Guest Agent...
Jul 30 15:19:07 KOM-AD01-PBX02 systemd[1]: Started oVirt Guest Agent.
Jul 30 15:19:08 KOM-AD01-PBX02 python[2700]: [90B blob data]
Jul 30 15:19:08 KOM-AD01-PBX02 python[2700]: [155B blob data]
Jul 30 15:19:08 KOM-AD01-PBX02 python[2700]: [100B blob data]
user@KOM-AD01-PBX02:~$
```

После того, как служба агента запущена в виртуальной машине, в веб-консоли oVirt нам станут доступны дополнительные сведения о текущем состоянии гостевой ОС. В частности, на закладке **General** появится информация о текущем состоянии утилизации оперативной памяти внутри VM.



Так же появится дополнительная информация на закладках **Applications**, **Guest Info** и других



На другую нашу виртуальную машину с **Hosted Engine** на базе **CentOS 7.2** агент был установлен из ранее подключённого репозитория **EPEL** (**yum install epel-release**) следующей последовательностью команд:

```
# yum -y install ovirt-guest-agent
# systemctl enable ovirt-guest-agent
# systemctl start ovirt-guest-agent
# service ovirt-guest-agent status
```

По поводу установки агента в гостевые системы на базе ОС Windows мы останавливаться не будем, так как в нашем случае виртуализация oVirt разворачивается исключительно под Linux/Unix системы, а для Windows систем есть более дружелюбная среда на базе гипервизора Hyper-V. Однако, для желающих эксплуатировать гостевые системы на базе Windows, в oVirt есть статья с примером установки гостевого агента в записи блога [Open Source Community - How to Install oVirt's Windows Guest Tools](#).

SPICE Agent

Простая инструкция по установке агента **SPICE vdagent**, расширяющего возможности работы с консолью гостевых ОС Linux есть в документе [How to install the spice guest agent](#). На разных дистрибутивах Linux в общем случае процедура сводится к установке пакета **spice-vdagent**. Например на нашей виртуальной машине с **Ubuntu 16.04 LTS** это делается так:

```
$ sudo apt-get install spice-vdagent
```

После установки служба автоматически запускается и настраивается на автозагрузку при старте системы:

```
user@KOM-AD01-PBX02: ~  
user@KOM-AD01-PBX02:~$ service spice-vdagent status  
● spice-vdagentd.service - Agent daemon for Spice guests  
   Loaded: loaded (/lib/systemd/system/spice-vdagentd.service; enabled; vendor preset: enab  
   Active: active (running) since Sat 2016-07-30 17:15:04 MSK; 3min 5s ago  
   Main PID: 4969 (spice-vdagentd)  
   CGroup: /system.slice/spice-vdagentd.service  
           └─4969 /usr/sbin/spice-vdagentd  
  
Jul 30 17:15:04 KOM-AD01-PBX02 systemd[1]: Starting Agent daemon for Spice guests...  
Jul 30 17:15:04 KOM-AD01-PBX02 systemd[1]: Started Agent daemon for Spice guests.  
user@KOM-AD01-PBX02:~$
```

После установки агента **SPICE vdagent** я заметил улучшение удобства работы с консолью ВМ в **Virt-Viewer**, в частности перестало дёргать курсор мыши и само перемещение курсора мыши между моей клиентской системой и окном консоли Virt-Viewer теперь стало работать прозрачно, без потери управления при попадании в окно консоли (Shift+F12 теперь стал не нужен для возврата управления курсором). А то, как работает функция общего буфера обмена, я так и не постиг.

На гостевых системах **RHEL/CentOS/Fedora** установка агента **SPICE** выполняется также просто:

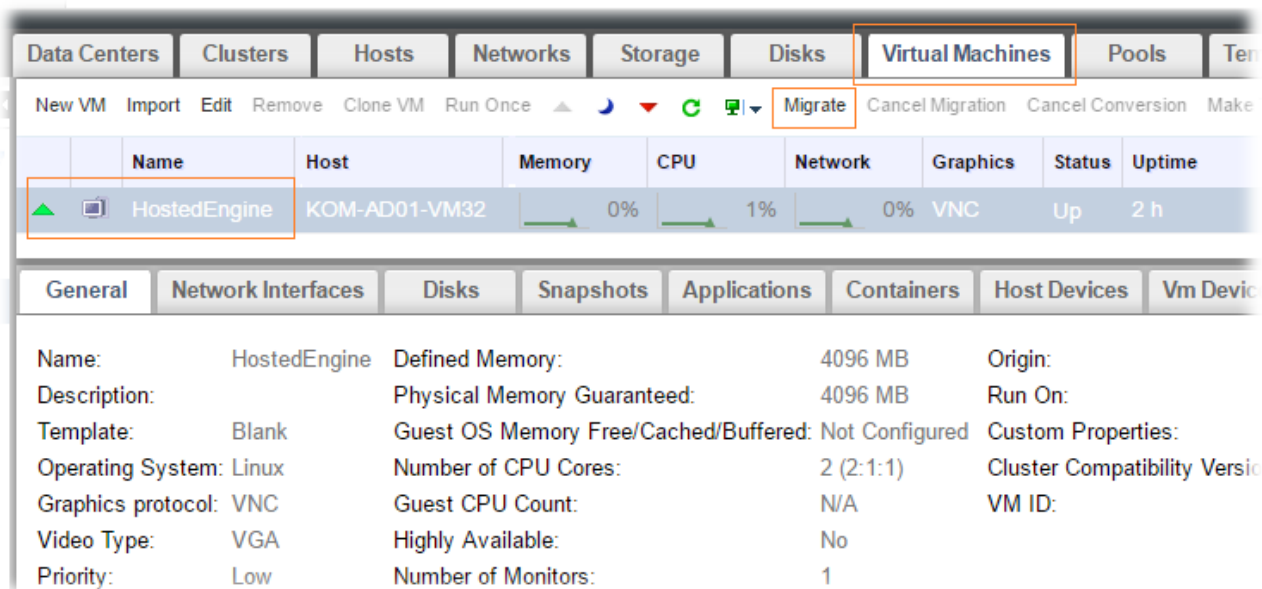
```
$ sudo yum install spice-vdagent
```

VirtIO Drivers

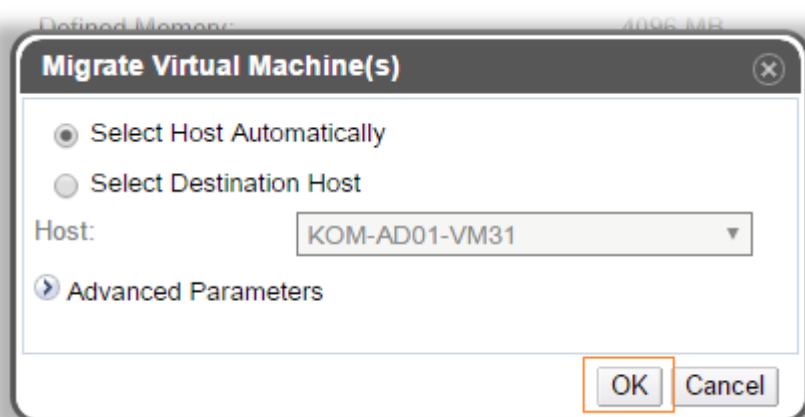
Согласно вышеупомянутого документа драйверы **VirtIO Drivers** уже входят в состав всех гостевых ОС Linux, а для Windows систем потребуется их отдельная установка (можно воспользоваться руководствами [How to create a Windows ... Virtual Machine](#)).

Проверка живой миграции

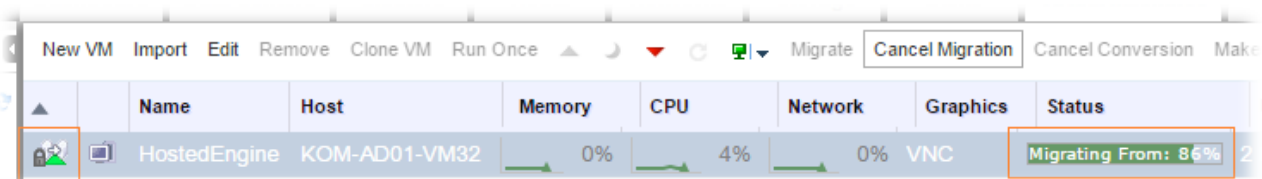
Так как у нас в кластере уже есть 2 хоста, мы можем попробовать вызвать процедуру живой миграции виртуальной машины между хостами. Для примера выполним процедуру живой миграции виртуальной машины Hosted Engine. Для проверки постоянной доступности виртуальной машины в процессе живой миграции перед запуском процедуры миграции запустим ping до виртуального сервера Engine, затем в веб-консоли oVirt выберем эту виртуальную машину и нажмём кнопку **Migrate...**



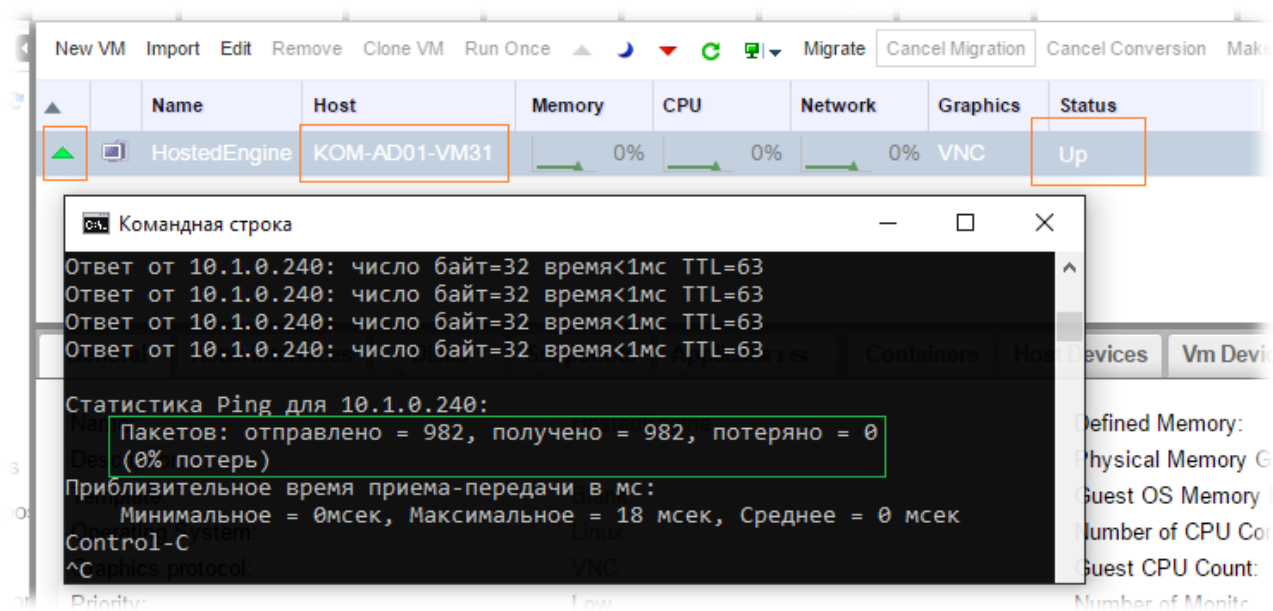
Откроется окно, где мы можем выбрать хост, на который нужно мигрировать виртуальную машину. В нашем примере используется автоматическое определение хоста назначения



После нажатия **OK** в консоли увидим изменение статуса виртуальной машины...



А спустя несколько секунд виртуальная машина уже будет работать на другом хосте, при этом доступной виртуальной машины останется на прежнем уровне. Проверяем статистику ring и видим, что потерь соединения с нашей виртуальной машиной в процессе миграции не было. И это хорошо.



В ранних версиях oVirt миграция виртуальной машины **Hosted Engine** подразумевала предварительное включение **Global HA Maintenance**, но современная версия **oVirt 4.0**, как следует, в частности, из документа [How to perform live migration of Hosted Engine VM](#) уже этого не требует.

На этом пока всё. В следующей части мы рассмотрим механизмы **Fencing** в oVirt, как средство достижения **High Availability**.