

Развёртывание и настройка oVirt 4.0. Часть 1.

Создание кластера виртуализации в конфигурации Hosted Engine

 blog.it-kb.ru/2016/09/10/install-ovirt-4-0-part-1-create-two-node-hosted-engine-cluster-with-shared-fc-san-storage

Автор:Алексей Максимов



oVirt версии **4.0** в конфигурации **Hosted Engine** с учётом тех граблей, по которым мне удалось походить. Но для начала пару слов о том, каким образом я пришёл к решению о необходимости развёртывания такого продукта, как oVirt.

Моя практика использования виртуальных машин на базе **Linux**, в доступной мне среде виртуализации на базе **Microsoft Hyper-V**, показала, что стабильность работы таких гостевых систем в некоторых сценариях, мягко говоря, оставляет желать лучшего. Например, неоднократно были замечены проблемы с некоторыми версиями **RHEL**, **Ubuntu**, **Debian**, когда банальная перезагрузка гостевой системы приводила к невозможности последующей загрузки ОС, или чехарда с компонентами интеграции. Некоторые проблемы вытекают из нестабильной работы всё тех же компонент интеграции, например, во время горячего резервного копирования виртуальных машин с **Ubuntu** из **System Center DPM**, **shadow copy** провайдер внутри ВМ порой делает что-то такое, после чего файловая система **ext4** на каком-нибудь из дисков переходит в режим **Read Only** и, как следствие, сервисы внутри гостевой системы ломаются. При этом я ничего плохого не хочу сказать про Hyper-V, как про платформу виртуализации, ведь все ОС от Microsoft работают в Hyper-V вполне достойно. Что же касается Linux-систем, то проблема, на мой взгляд, заключается в том, что, на самом деле, Microsoft не уделяет должного внимания вопросу контроля за средствами интеграции Hyper-V с операционными средами, отличными от Windows. Видимо для этих ребят гораздо важнее заявить миру о том, что теперь PowerShell можно запустить в Linux, а bash теперь работает в Windows. В общем по личному опыту, пока у меня есть устоявшееся мнение о том, что Hyper-V, как гипервизор, не может претендовать на звание решения для гетерогенных сред. Вы скажете, есть же **VMware**! Но к полноценной среде виртуализации от VMware с поддержкой кластеризации и прочих "плюшек" у меня доступа нет, поэтому про эту платформу я сказать не могу ничего. В общем, исходя из соображений того, что с виртуальными Linux-системами, так как я живу, жить больше нельзя, я начал своё движение в сторону изучения средств виртуализации серверов в мире открытых и свободно-распространяемых систем. Перечитав кучу разных "холиваров" на тему выбора лучшего средства управления гипервизором **KVM**, как наиболее популярным средством программной виртуализации для Linux-систем, меня заинтересовали два решения – это Proxmox VE и oVirt. Исходя из информации, доступной в открытых источниках, я сравнил функциональные возможности обоих продуктов и пришёл к выводу, что oVirt имеет больше перспектив развития, так как по сути является более масштабируемым решением,

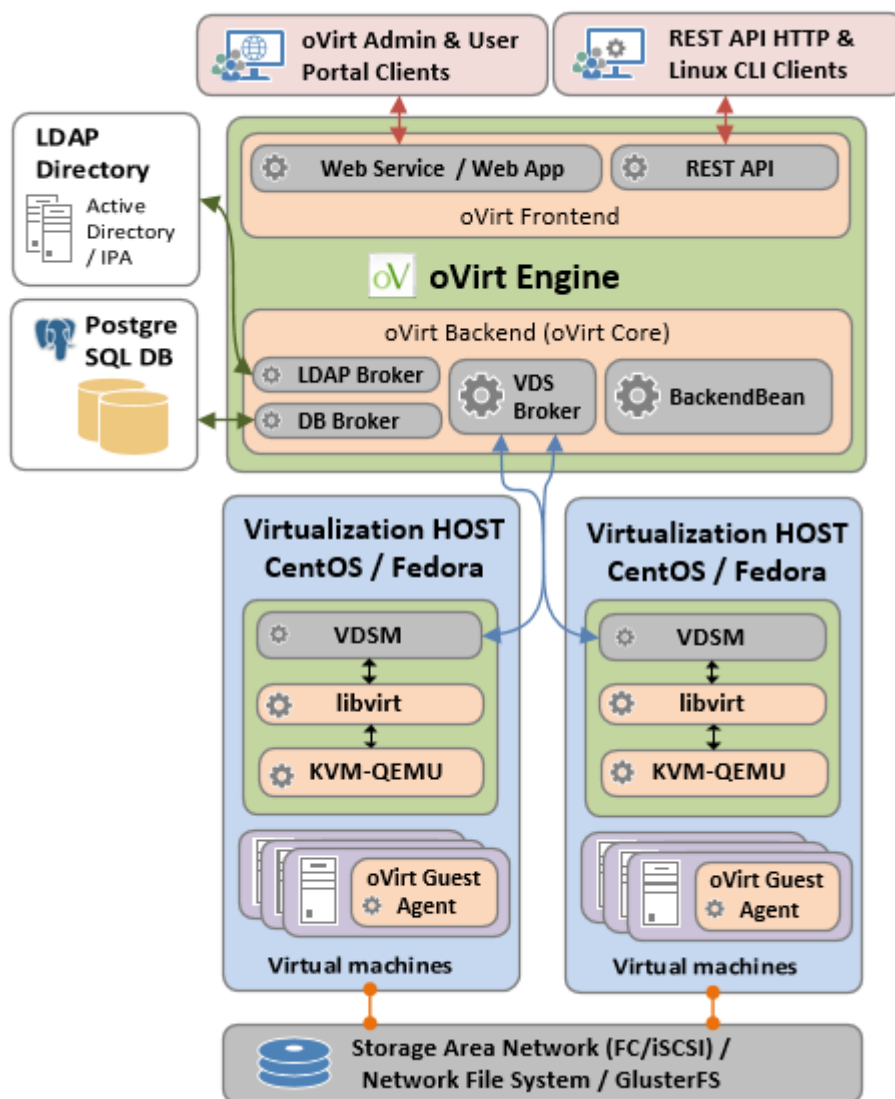
чем Proxmox VE. Несмотря на некоторые свои явные недостатки с точки зрения "юзабилити", например отсутствие бэкапа виртуальных машин штатными средствами администрирования (чего не скажешь о Proxmox), oVirt мне показался более интересным в том плане, что он является upstream-решением для платформы виртуализации **Red Hat Enterprise Virtualization (RHEV)**. А как известно, компания **Qumranet**, в недрах которой родилась технология **KVM**, в 2008 году была куплена именно компанией **Red Hat**. Поэтому кто, как не ребята из Red Hat, "умеют правильно готовить" обвязку для KVM.

Итак, **oVirt**. Сразу хочу определиться с терминологией и архитектурой oVirt, чтобы не было в дальнейшем каких-то недопониманий того, что и зачем делается. Сам по себе, oVirt не является гипервизором, а является средством управления гипервизором KVM. По сути oVirt в сегодняшнем его состоянии уже можно ставить в одну линейку с такими продуктам, как **VMware vCenter** или **Microsoft System Center Virtual Machine Manager**, делая при этом скидку на то, что это всё-таки не коммерческое решение, и ждать от него всего богатства функционала, как у коммерческих решений, не приходится. Однако, как я вижу, oVirt сегодня очень активно развивается и поддерживается компанией **Red Hat**. И это не может не радовать.

Архитектура oVirt

Сердцем oVirt является сервер с компонентами **oVirt Engine**. С этого сервера происходит управление всеми хостами виртуализации, общими дисковыми ресурсами и виртуальными сетями. В ранних версиях oVirt Engine требовал наличия выделенного сервера под свою управляющую роль и для построения высоко-доступной инфраструктуры oVirt требовалось, как минимум, 3 физических сервера – один сервер под oVirt Engine и минимум два хоста виртуализации. В настоящее же время oVirt Engine не требует наличия выделенного физического сервера, а может быть развёрнут в виде виртуальной машины прямо на том хосте виртуализации, которым он будет в последствии управлять. Такой вариант развёртывания в терминологии oVirt называется **Hosted Engine**. Таким образом, теперь для развёртывания высоко-доступной инфраструктуры oVirt требуется наличие лишь двух физических серверов. Обратите внимание на то, что говоря про oVirt, я сразу завёл речь о высокой доступности, так как я считаю, что заниматься внедрением oVirt имеет смысл лишь тогда, когда вы планируете использовать функционал кластеризации и балансировки нагрузки с использованием **нескольких хостов виртуализации** (от двух и более). Что же касается более мелких вариантов виртуальных сред (1-2 хоста), то для них более эффективным, с точки зрения максимального упрощения развёртывания, может оказаться использование таких продуктов, как ранее упомянутый Proxmox VE.

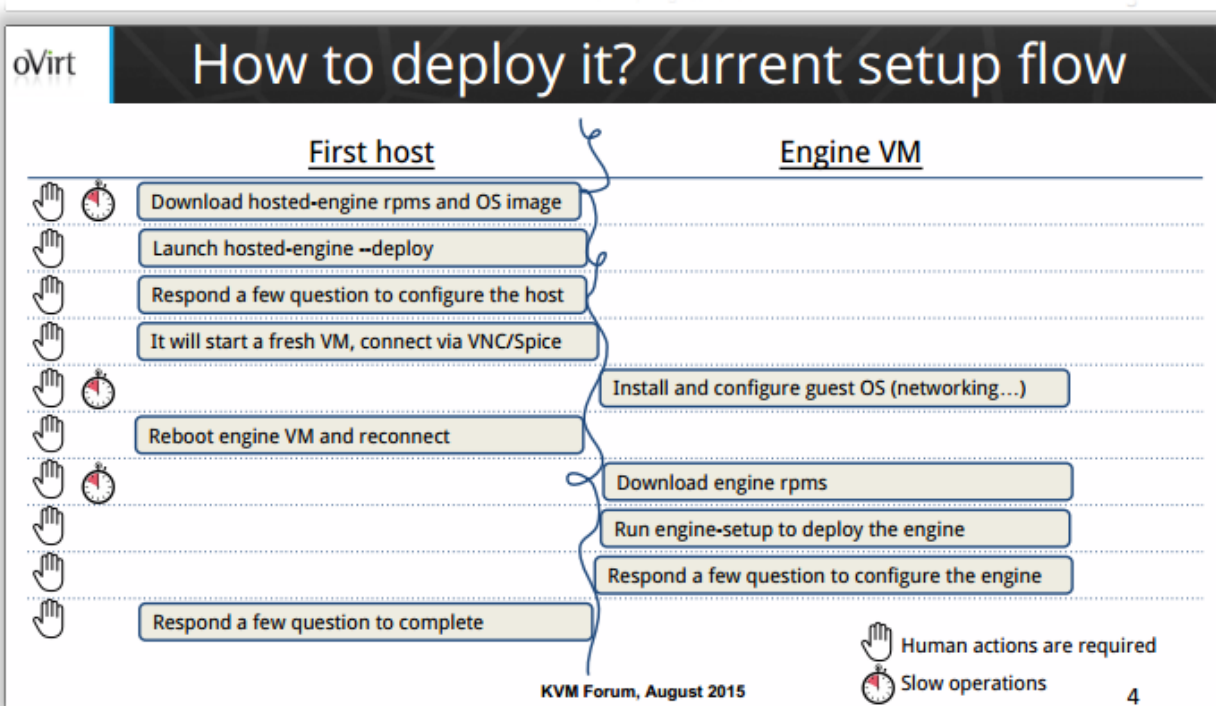
Основная информация об составе компонент и архитектуре oVirt собрана на Вики-странице [oVirt Architecture](#). Здесь я постарался в одной схеме собрать информацию из разных схем:



Дополнительную информацию по архитектуре oVirt можно найти и в других источниках, например не будет лишним просмотреть презентацию [Yaniv Bronhaim \(Red Hat Senior Software Engineer\) - March 2016 The oVirt Way General Product Overview](#). О назначении основных компонент архитектуры неплохо расписал на русском языке в своём блоге [Александр Руденко](#).

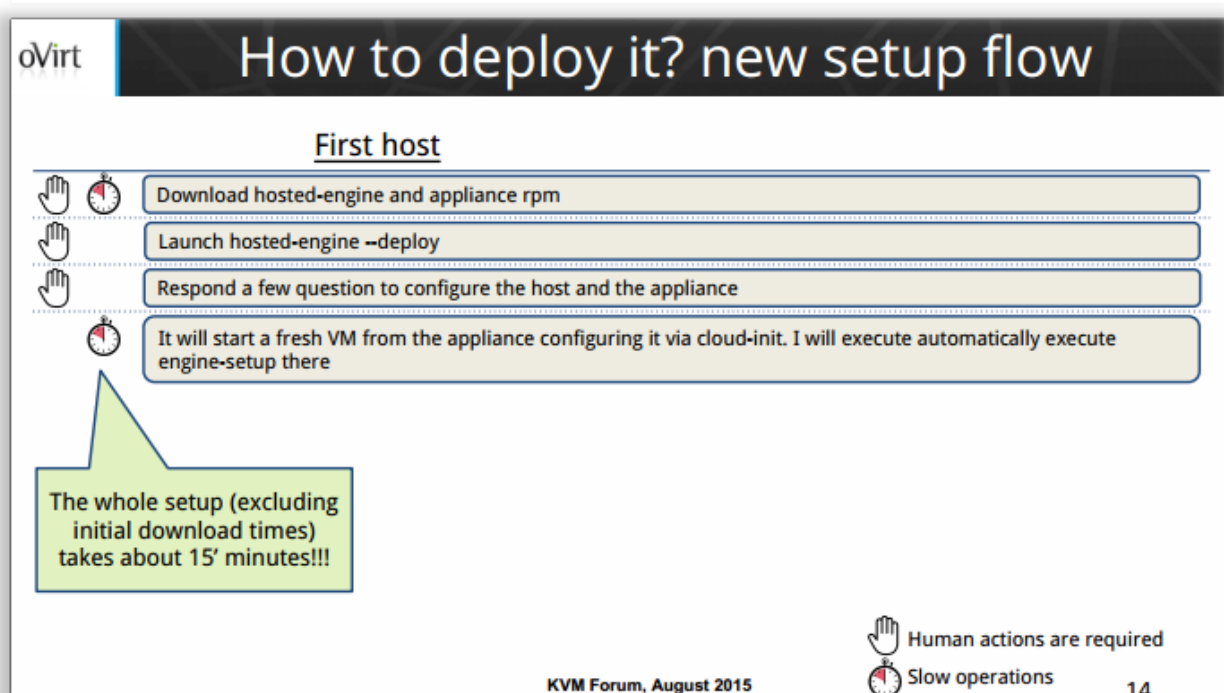
План развёртывания oVirt Hosted Engine

Для развёртывания **oVirt** версии **4.0** в конфигурации **Hosted Engine** в качестве опорного руководства будем использовать [oVirt Hosted Engine Howto](#). Перед тем, как приступить к процедуре развёртывания, для лучшего понимания последовательности этапов развёртывания, посмотрим на очень наглядную, на мой взгляд, схему из презентации **Simone Tiraboschi** (Software Engineer, Red Hat):



Видео, где Simone сам комментирует свою презентацию, можно найти здесь: [oVirt self-hosted engine seamless deployment by Simone Tiraboschi](#)

Кстати в этой же презентации Simone представляет нам и альтернативные варианты развёртывания oVirt, в том числе, применительно к варианту Hosted Engine, в презентации уделяется внимание такому варианту развёртывания, как [oVirt Hosted Engine Appliance](#). Слайд презентации также наглядно показывает нам то, что данный вариант с готовой виртуальной машиной oVirt Hosted Engine в виде **Appliance** может сократить количество этапов развёртывания Hosted Engine и выглядит более привлекательно с точки зрения ускорения процесса развёртывания.



Однако меня несколько смутил тот факт, что грп-пакеты, необходимые для такого варианта развёртывания, постоянно пересобираются и доступны для загрузки в виде последней успешной сборки. К тому же, мне хотелось самостоятельно установить ОС внутри виртуальной машины Hosted Engine так, как мне это нужно, контролируя процесс происходящего, а не получать уже готовую, кем-то и как-то сконфигурированную, Linux-систему. Поэтому я остановился на варианте развёртывания из первого слайда, и далее в этой статье будут последовательно рассмотрены все этапы такого развёртывания.

Подготовка хостов к развёртыванию oVirt Hosted Engine

Нам потребуется подготовить **2** физических сервера с ОС **CentOS Linux 7.2** под будущие хосты виртуализации. В моём случае используется два "олд-скульных" сервера **HP ProLiant DL360 G5** одинаковой аппаратной конфигурации (особенности установки CentOS 7 на такой сервер описывались ранее) :

- 2 Quad-Core процессора Intel Xeon X5460 3.16GHz (с поддержкой Intel Virtualization Technology (VT-x))
- 2 встроенных сетевых адаптера NC373i Multifunction Gigabit Server Adapter, объединённых в LACP Bond для агрегации пропускной способности и повышения доступности.
- двух-портовый FC HBA (HP FC1242SR / HP FC2242SR), подключённый к SAN.

С СХД каждому из хостов презентовано (с использованием multipath) общее хранилище через FC SAN в виде двух дисков. Один выделенный диск - под виртуальную машину oVirt Hosted Engine, второй - под все остальные виртуальные машины будущего кластера oVirt.

Хосты будут иметь имена **KOM-AD01-VM31** и **KOM-AD01-VM32**. В процессе развёртывания oVirt нами будет создана виртуальная машина с именем **KOM-AD01-OVIRT1**, так же с ОС **CentOS Linux 7.2**

Проверяем на обоих хостах наличие поддержки виртуализации на процессоре сервера, например, утилитой lscpu:

```
root@KOM-AD01-VM31:~  
[root@KOM-AD01-VM31 ~]#  
[root@KOM-AD01-VM31 ~]# lscpu  
Architecture:          x86_64  
CPU op-mode(s):        32-bit, 64-bit  
Byte Order:             Little Endian  
CPU(s):                 8  
On-line CPU(s) list:    0-7  
Thread(s) per core:     1  
Core(s) per socket:     4  
Socket(s):              2  
NUMA node(s):          1  
Vendor ID:              GenuineIntel  
CPU family:             6  
Model:                 23  
Model name:             Intel(R) Xeon(R) CPU           X5460  @ 3.16GHz  
Stepping:               6  
CPU MHz:               3166.744  
BogoMIPS:               6333.35  
Virtualization:         VT-x  
L1d cache:              32K  
L1i cache:              32K  
L2 cache:               6144K  
NUMA node0 CPU(s):      0-7
```

Другой вариант, посмотреть вывод команды:

```
# cat /proc/cpuinfo
```

Среди флагов процессора должен присутствовать флаг **vmx** (для процессоров **Intel**), либо флаг **svm** (для процессоров **AMD**).

```
root@KOM-AD01-VM31:~  
[root@KOM-AD01-VM31 ~]# cat /proc/cpuinfo  
processor       : 0  
vendor_id      : GenuineIntel  
cpu family     : 6  
model          : 23  
model name     : Intel(R) Xeon(R) CPU           X5460  @ 3.16GHz  
stepping       : 6  
microcode      : 0x612  
cpu MHz        : 3166.744  
cache size     : 6144 KB  
physical id    : 0  
siblings       : 4  
core id        : 0  
cpu cores      : 4  
apicid         : 0  
initial apicid : 0  
fpu            : yes  
fpu_exception  : yes  
cpuid level    : 10  
wp             : yes  
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx lm constant_tsc arch_perfmon pebs  
bts rep_good nopl aperfmpperf pni dtes64 monitor ds_cpl vmx est tm2 ssse3 cx16 xtpr pdcm dc
```

Поддержка есть, двигаемся дальше.

Останавливаем и отключаем службу **NetworkManager**, так как oVirt берёт на себя управление сетевыми профилями в системе. Когда я пробовал выполнять развёртывание oVirt 3.6, с запущенной службой NetworkManager, oVirt попросту не

устанавливался. В oVirt 4.0 ситуация изменилась, и теперь процесс установки работает с запущенной службой NetworkManager, однако я предпочёл всё же отключить и остановить данную службу и сейчас:

```
# systemctl stop NetworkManager
# systemctl disable NetworkManager
```

Чтобы избежать граблей описанных в баге [Red Hat Bugzilla – Bug 1160423 - hosted-engine --deploy doesn't copy DNS config to ovirtmgmt](#), задействуем использование в качестве основного источника настроек DNS клиента файл resolv.conf

```
# cat /etc/resolv.conf
```

```
nameserver 10.1.0.9
nameserver 10.1.6.8
search holding.com
```

В файлах, описывающих сетевые интерфейсы (/etc/sysconfig/network-scripts/ifcfg*), уберём значения параметров DNS, там где они указаны в явном виде (удалим параметры **DNS1**, **DNS2** и т.д.) и добавим параметр **PEERDNS=no**. Например, в моём случае, конфигурационный файл интерфейса управления (vlan-интерфейс поверх bond-интерфейса), через который я подключаюсь к серверу выглядит следующим образом:

```
# cat /etc/sysconfig/network-scripts/ifcfg-bond0.4
```

```
TYPE=Vlan
VLAN_ID=4
DEVICE=bond0.4
NAME=bond0-vlan4
BOOTPROTO=none
ONBOOT=yes
USERCTL=no
NM_CONTROLLED=no
VLAN=yes
MTU=1500
IPV4_FAILURE_FATAL=yes
IPV6INIT=no
IPADDR=10.1.0.231
PREFIX=24
GATEWAY=10.1.0.1
PEERDNS=no
DOMAIN=holding.com
```

<

```
p align="center">***
```

Далее. Если ранее на сервере была полностью отключена поддержка **ipv6**, то, как минимум, потребуется её включить в явном виде для интерфейсов **lo** и **ovirtmgmt** (будет создан в системе дальнейшем с процессе развёртывания oVirt) . Это нужно, чтобы процесс установки смог подключиться к службе **vdsm** (это грабли версии 4.0,

которые мусолились в ветке [\[ovirt-users\] oVirt 4 Hosted Engine deploy on fc storage - \[ERROR\] Failed to execute stage 'Misc configuration': \[Errno 101\] Network is unreachable](#) и по информации [Bug_1350883](#) были пофиксены в версии 4.0.2)

Добавим в системный конфигурационный файл `/etc/sysctl.conf` пару строчек:

```
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 0

net.ipv6.conf.ovirtmgmt.disable_ipv6 = 0
```

Чтобы сразу задействовать включение `ipv6`, например на интерфейсе `lo`, можно сделать так:

```
# cat /proc/sys/net/ipv6/conf/lo/disable_ipv6
1
# echo 0 > /proc/sys/net/ipv6/conf/lo/disable_ipv6
# cat /proc/sys/net/ipv6/conf/lo/disable_ipv6
0
```

После проделанных изменений перезагрузим сервер и проверим, чтобы на Loopback-интерфейсе был назначен адрес `ipv6 ::1`

```
# ip addr show lo

1: lo: mtu 65536 qdisc noqueue state UNKNOWN
    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

***
```

Добавляем информацию о репозитории **oVirt** и **Epel**. Это обязательный шаг.

```
# yum -y install http://resources.ovirt.org/pub/yum-repo/ovirt-release40.rpm
# yum -y install epel-release
```

Развёртывание oVirt Hosted Engine на первом хосте

Загружаем минимальный установочный образ **CentOS 7** во временную папку на первом хосте (KOM-AD01-VM31), который мы будем использовать для развёртывания oVirt. Этот образ потребуется нам в дальнейшем для установки виртуальной машины с oVirt Hosted Engine:

```
# wget http://mirror.yandex.ru/centos/7/isos/x86_64/CentOS-7-x86_64-NetInstall-1511.iso -P /tmp/
```

Репозитории oVirt и Epel уже были подключены ранее на этапе подготовки хоста. Устанавливаем пакеты, необходимые для развёртывания oVirt Hosted Engine:

```
# yum install ovirt-hosted-engine-setup
```


В процессе установки в систему дополнительно будет установлено более трёхсот пакетов-зависимостей.

Дополнительно установим утилиту **screen**:

```
# yum install screen
```

Перед запуском процедуры развёртывания запустим утилиту screen, как это рекомендует онлайн-руководство, создав screen-сессию и подключившись к ней.

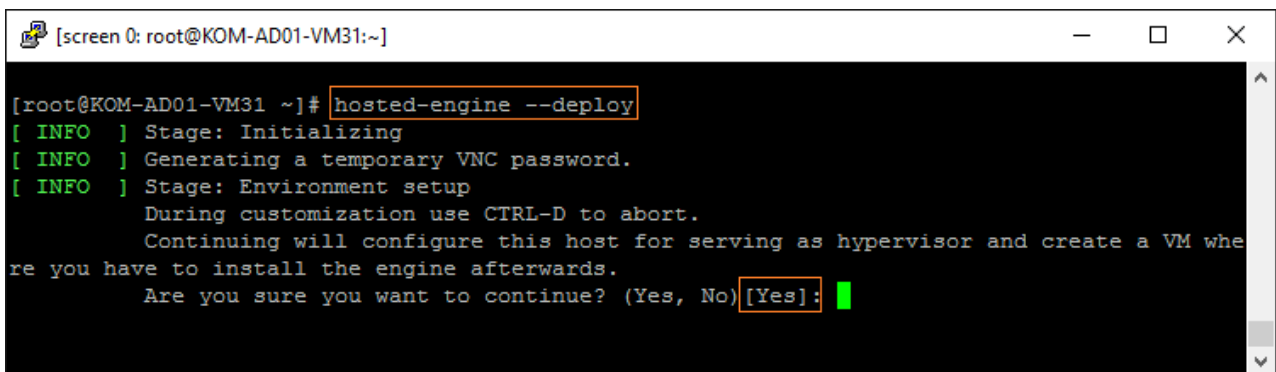
```
# screen -RD
```



В screen-сессии запускаем процедуру развёртывания oVirt Hosted Engine:

```
# hosted-engine --deploy
```

В процессе развёртывания будет задан ряд вопросов, на которые мы должны ответить. На первый вопрос о том, что мы действительно хотим сделать из данного сервера хост виртуализации и после этого создать виртуальную машину для развёртывания Hosted Engine, отвечаем утвердительно (можно просто Enter, так как в фигурных скобках отражается предлагаемый по умолчанию ответ):



Конфигурация хранилища для виртуальной машины Hosted Engine

На начальном этапе развёртывания нас спросят о том, какой тип хранилища мы будем использовать для размещения файлов виртуальной машины с oVirt Hosted Engine. В нашем случае выбран тип хранилища - **fc**

```
[screen 0: root@KOM-AD01-VM31:~]
Are you sure you want to continue? (Yes, No) [Yes]:
[ INFO ] Hardware supports virtualization
Configuration files: []
Log file: /var/log/ovirt-hosted-engine-setup/ovirt-hosted-engine-setup-2016072114
0146-y8cyzu.log
Version: otopi-1.5.1 (otopi-1.5.1-1.el7.centos)
[ INFO ] Stage: Environment packages setup
[ INFO ] Stage: Programs detection
[ INFO ] Stage: Environment setup
[ INFO ] Generating libvirt-spice certificates
[WARNING] Cannot locate gluster packages, Hyper Converged setup support will be disabled.
[ INFO ] Please abort the setup and install vdsmd-gluster, gluster-server >= 3.7.2 and rest
art vdsmd service in order to gain Hyper Converged setup support.
[ INFO ] Stage: Environment customization

---== STORAGE CONFIGURATION ===--

Please specify the storage you would like to use (glusterfs, iscsi, fc, nfs3, nfs
4) [nfs3]: fc
```

Следующим шагом, программа установки опросит систему на предмет доступных для сервера хранилищ выбранного типа и выдаст перечень возможных вариантов. Выбираем **LUN**, в который будет размещена виртуальная машина с Hosted Engine. Чтобы LUN можно было использовать для первичного развёртывания oVirt, он не должен содержать никаких разделов (должен иметь статус **free**):

```
[screen 0: root@KOM-AD01-VM31:~]
4) [nfs3]
The following luns have been found on the requested target:
[1] 360002ac0000000000000000160000cec9 2000GiB 3PARdata VV
status: free, paths: 4 active
[2] 360002ac00000000000000001e0000cec9 90GiB 3PARdata VV
status: free, paths: 4 active
Please select the destination LUN (1, 2) [1]: 2
```

Конфигурация сети хоста для создания bridge-интерфейса

Следующий этап - конфигурация сети. Выбираем тот интерфейс, у которого есть фактическая привязка к сети. Как я уже ранее сказал, в нашем случае используется Bond-интерфейс **bond0**, поверх которого создан тегированный VLAN-интерфейс **bond0.4**. С этого интерфейса должен быть доступен указанный далее **IP-адрес шлюза** (в противном случае установка завершится ошибкой). На базе указанного интерфейса в процессе развёртывания будет создан bridge-интерфейс с именем **ovirtmgmt**, к которому в свою очередь будет выполнена привязка виртуального сетевого адаптера виртуальной машины Hosted Engine.

На вопрос об автоматической настройке брандмауэра отвечаем утвердительно, чтобы скрипты установки создали нужные для функционирования oVirt правила в **iptables** и **firewalld**

```
[screen 0: root@KOM-AD01-VM31:~]
[ INFO ] Installing on first host

---== SYSTEM CONFIGURATION ===

---== NETWORK CONFIGURATION ===

Please indicate a nic to set ovirtmgmt bridge on: (bond0, bond0.4) [bond0]: bond0
.4
iptables was detected on your computer, do you wish setup to configure it? (Yes,
No) [Yes]:
Please indicate a pingable gateway IP address [10.1.0.1]:
```

Конфигурация виртуальной машины для роли Hosted Engine

Следующий этап – задание параметров будущей виртуальной машины Hosted Engine.

- Сначала нужно указать источник загрузки дистрибутива Linux, который будет использоваться в процессе установки ОС внутрь виртуальной машины Hosted Engine. Так, как мы планируем установку ОС с ранее загруженного ISO-образа, выбираем виртуальный привод - **cdrom**.
- Затем нужно выбрать протокол подключения к консоли ВМ – оставляю предлагаемый по умолчанию **vnc**.
- Затем нас спросят о том, какого типа процессоры должны использоваться для ВМ – оставляю предлагаемый по умолчанию тип **model_Penryn** (у вас он может быть другим в зависимости от физических процессоров хоста).
- Количество виртуальных процессоров - минимум **2**.
- Размер диска ВМ в большинстве случаев тоже можно оставить предлагаемый по умолчанию (**25 GB**). Однако в нашем примере, мы сделаем размер больше предлагаемого, так как на этой ВМ мы в дальнейшем сделаем NFS-шару, на которой будут размещаться ISO-образы, которые в случае необходимости будут монтироваться на виртуальные машины (oVirt ISO Domain) – **40 GB**
- Адрес MAC для виртуальной сетевой карты оставляем сгенерированный автоматически.
- Размер оперативной памяти ВМ оставляю предлагаемый минимум **4096 MB**. При необходимости позже его можно будет изменить. Рекомендуемым значением при этом является 16 GB.
- В конце задаём путь к ранее загруженному ISO-образу установочного диска CentOS 7.

```
[screen 0: root@KOM-AD01-VM31:~]
--== VM CONFIGURATION ==--

Bootimg from cdrom on RHEL7 is ISO image based only, as cdrom passthrough is disabled (BZ760885)
Please specify the device to boot the VM from (choose disk for the oVirt engine appliance)
(cdrom, disk, pxe) [disk]: cdrom
Please specify the console type you would like to use to connect to the VM (vnc, spice) [vnc]:
The following CPU types are supported by this host:
- model_Penryn: Intel Penryn Family
- model_Conroe: Intel Conroe Family
Please specify the CPU type to be used by the VM [model_Penryn]:
Please specify the number of virtual CPUs for the VM (Defaults to minimum requirement): [2]:
Please specify the disk size of the VM in GB (Defaults to minimum requirement): [25]: 40
You may specify a unicast MAC address for the VM or accept a randomly generated default [00:16:3e:29:8e:29]:
Please specify the memory size of the VM in MB (Defaults to minimum requirement): [4096]:
Please specify path to installation media you would like to use [None]: /tmp/CentOS-7-x86_64-NetInstall-1511.iso
```

Конфигурация Hosted Engine.

Следующий этап – задание параметров, относящихся к работе Hosted Engine.

- Укажем пароль для пользователя **admin**. Пароль, который зададим здесь, будет в дальнейшем использоваться для доступа к portalу администрирования oVirt (чтобы в дальнейшем не запутаться, для примера, будем считать что этот пароль у нас Pa\$sw0rD). Этот же пароль нам потребуется ввести в дальнейшем в процессе развёртывания управляющего ПО в виртуальной машине для oVirt Engine. Пароль вводим два раза.
- Укажем имя хоста, с которого производится установка виртуальной машины. В нашем случае это хост **KOM-AD01-VM31**.
- Укажем полное доменное имя для виртуального сервера Hosted Engine. В нашем случае это хост **KOM-AD01-OVIRT1.holding.com**.
- Укажем адрес и порт почтового сервера SMTP, на который будут отправляться почтовые уведомления. По умолчанию в этих полях установлен вариант **localhost** и порт **25**. Чтобы такая конфигурация работала, можно в дальнейшем будет настроить локальную службу SMTP-сервера, например так, как это описано в [здесь](#). Я выбрал другой вариант – сразу указал адрес корпоративного почтового SMTP сервера.
- В конце укажем e-mail адрес отправителя и адрес получателя уведомлений (в моём случае это адрес группы рассылку в которую входят администраторы oVirt).

```
[screen 0: root@KOM-AD01-VM31:~]

---- HOSTED ENGINE CONFIGURATION ----

Enter engine admin password:
Confirm engine admin password:
Enter the name which will be used to identify this host inside the Administrator
Portal [hosted_engine_1]: KOM-AD01-VM31
Please provide the FQDN for the engine you would like to use.
This needs to match the FQDN that you will use for the engine installation within
the VM.
Note: This will be the FQDN of the VM you are now going to create,
it should not point to the base host or to any other existing machine.
Engine FQDN: []: KOM-AD01-OVIRT1.holding.com
Please provide the name of the SMTP server through which we will send notificatio
ns [localhost]: 10.1.0.18
Please provide the TCP port number of the SMTP server [25]:
Please provide the email address from which notifications will be sent [root@loca
lhost]: KOM-AD01-OVIRT1@holding.com
Please provide a comma-separated list of email addresses which will get notificat
ions [root@localhost]: DST-oVirt-Administrators@holding.com
```

Установка на хост компонент oVirt и создание виртуальной машины

После того как все параметры будущей виртуальной машины Hosted Engine заданы, ещё раз их просматриваем и жмём Enter для подтверждения запуска процесса установки:

```
[screen 0: root@KOM-AD01-VM31:~]

[ INFO ] Stage: Setup validation

---- CONFIGURATION PREVIEW ----

Bridge interface           : bond0.4
Engine FQDN                : KOM-AD01-OVIRT1.holding.com
Bridge name               : ovirtmgmt
Host address               : kom-ad01-vm31.holding.com
SSH daemon port           : 22
Firewall manager           : iptables
Gateway address            : 10.1.0.1
Host name for web application : KOM-AD01-VM31
Storage Domain type        : fc
Host ID                    : 1
LUN ID                     : 360002ac00000000000000001e0000cec9
Image size GB              : 40
Console type               : vnc
Memory size MB             : 4096
MAC address                : 00:16:3e:29:8e:29
Boot type                  : cdrom
Number of CPUs              : 2
ISO image (cdrom boot/cloud-init) : /tmp/CentOS-7-x86_64-NetInstall-1511.iso
CPU Type                   : model_Penryn

Please confirm installation settings (Yes, No) [Yes]:
```

В ходе установки будет выполнена настройка **libvirt** и **VDSM**, настройка дискового хранилища, которое мы выделили под виртуальную машину Hosted Engine, создание виртуальной машины...

```
[screen 0: root@KOM-AD01-VM31:~]
Please confirm installation settings (Yes, No)[Yes]:
[ INFO ] Stage: Transaction setup
[ INFO ] Stage: Misc configuration
[ INFO ] Stage: Package installation
[ INFO ] Stage: Misc configuration
[ INFO ] Configuring libvirt
[ INFO ] Configuring VDSM
[ INFO ] Starting vdsmd
[ INFO ] Configuring the management bridge
[ INFO ] Creating Volume Group
[ INFO ] Creating Storage Domain
[ INFO ] Creating Storage Pool
[ INFO ] Connecting Storage Pool
[ INFO ] Verifying sanlock lockspace initialization
[ INFO ] Creating Image for 'hosted-engine.lockspace' ...
[ INFO ] Image for 'hosted-engine.lockspace' created successfully
[ INFO ] Creating Image for 'hosted-engine.metadata' ...
[ INFO ] Image for 'hosted-engine.metadata' created successfully
[ INFO ] Creating VM Image
[ INFO ] Destroying Storage Pool
[ INFO ] Start monitoring domain
[ INFO ] Configuring VM
[ INFO ] Updating hosted-engine configuration
[ INFO ] Stage: Transaction commit
[ INFO ] Stage: Closing up
[ INFO ] Creating VM
```

Затем на экран будет выведено предложение подключиться к консоли созданной виртуальной машины для установки внутрь этой VM гостевой ОС. Консоль виртуальной машины будет нам доступна на нашем хосте (KOM-AD01-VM31) по протоколу **VNC** на порту **5900** с временным паролем доступа указанным в сообщении.

Появившееся в самом конце сообщения меню выбора действий, мы пока оставим без выбора, то есть ничего больше пока не нажимаем в этом сеансе.

Предполагается, что первый пункт будет выбран позднее, в том случае, если дальнейшая процедура развертывания ОС в виртуальную машину для Hosted Engine пройдет штатно. Если же в процессе установки CentOS внутрь виртуальной машины возникнут проблемы, то можно будет вернуться к этому меню и выбрать пункт 3 для того, чтобы гипервизору хоста была дана команда на перезапуск этой VM. Пункты 2 и 4 нужны в том, случае если вы решили "на всё плюнуть" и пойти играть в пинг-понг.

```
[screen 0: root@KOM-AD01-VM31:~]
[ INFO ] Creating VM
You can now connect to the VM with the following command:
/bin/remote-viewer vnc://localhost:5900
Use temporary password "8512gNcF" to connect to vnc console.
Please note that in order to use remote-viewer you need to be able to run graphic
al applications.
This means that if you are using ssh you have to supply the -Y flag (enables trust
ed X11 forwarding).
Otherwise you can run the command from a terminal in your preferred desktop enviro
nment.
If you cannot run graphical applications you can connect to the graphic console fr
om another host or connect to the serial console using the following command:
socat UNIX-CONNECT:/var/run/ovirt-vmconsole-console/3c6cfc11-9af1-4e68-89b8-100c2
b005832.sock,user=ovirt-vmconsole STDIO,raw,echo=0,escape=1
Please ensure that your Guest OS is properly configured to support serial console
according to your distro documentation.
Follow http://www.ovirt.org/Serial_Console_Setup#I_need_to_access_the_console_the
_old_way for more info.
If you need to reboot the VM you will need to start it manually using the command
:
hosted-engine --vm-start
You can then set a temporary password using the command:
hosted-engine --add-console-password

The VM has been started.
To continue please install OS and shutdown or reboot the VM.

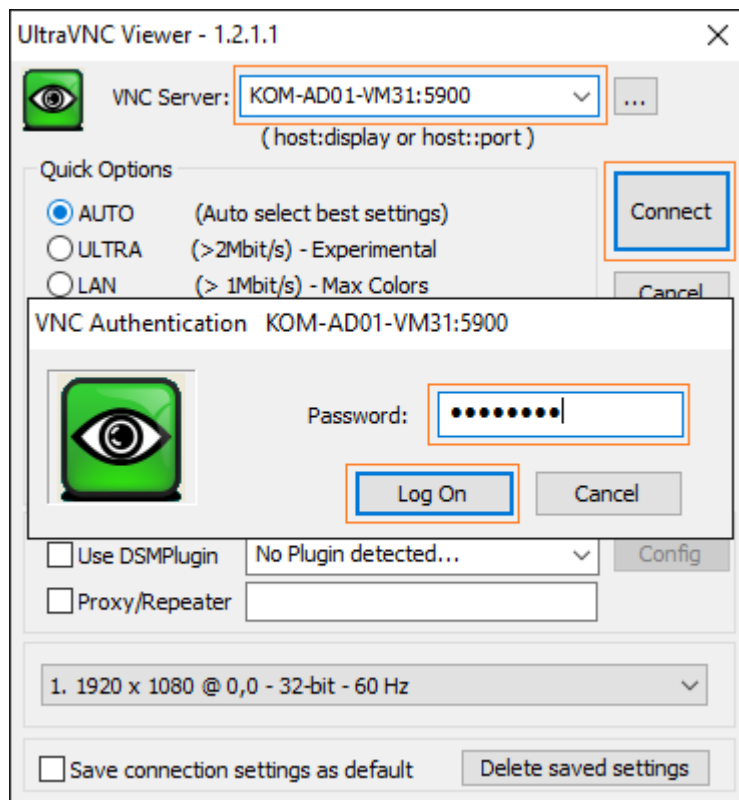
Make a selection from the options below:
(1) Continue setup - OS installation is complete
(2) Abort setup
(3) Power off and restart the VM
(4) Destroy VM and abort setup

(1, 2, 3, 4) [1]: █
```

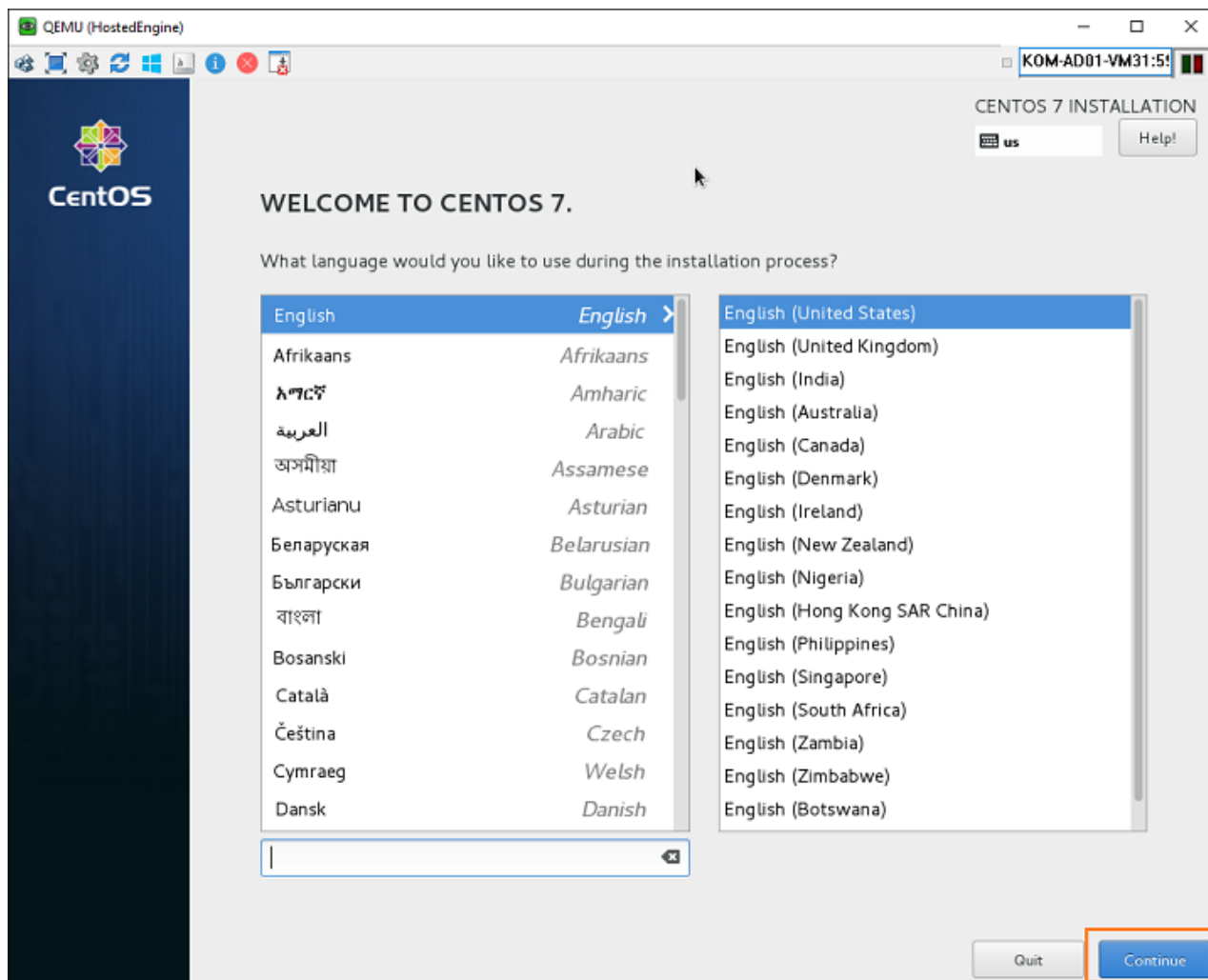
Итак, оставим текущий сеанс в ожидании и создадим новое подключение к консоли только что созданной и запущенной виртуальной машины Hosted Engine по протоколу **VNC**.

Установка гостевой ОС в виртуальную машину Hosted Engine

Так как наш хост не имеет установленных **X**-компонент, то подключение к консоли виртуальной машины можно сделать с любого другого компьютера локальной сети, имеющего графическую среду и средства работы с протоколом VNC. Я использую для этой задачи рабочую станцию на базе **Microsoft Windows** и утилиту **UltraVNC Viewer** (загрузить можно [отсюда](#)). Интерфейс утилиты "прост до безобразия", и всё, что нам нужно указать это адрес подключения и пароль, который мы только что увидели в screen-сессии на хосте:



Подключившись к консоли виртуальной машины, выполняем стандартную установку **CentOS 7.2**. Пример установки рассматривался ранее. Отличие этой установки в том, что она несколько проще, так как сейчас не потребуется дополнительных действий с подгрузкой драйверов.



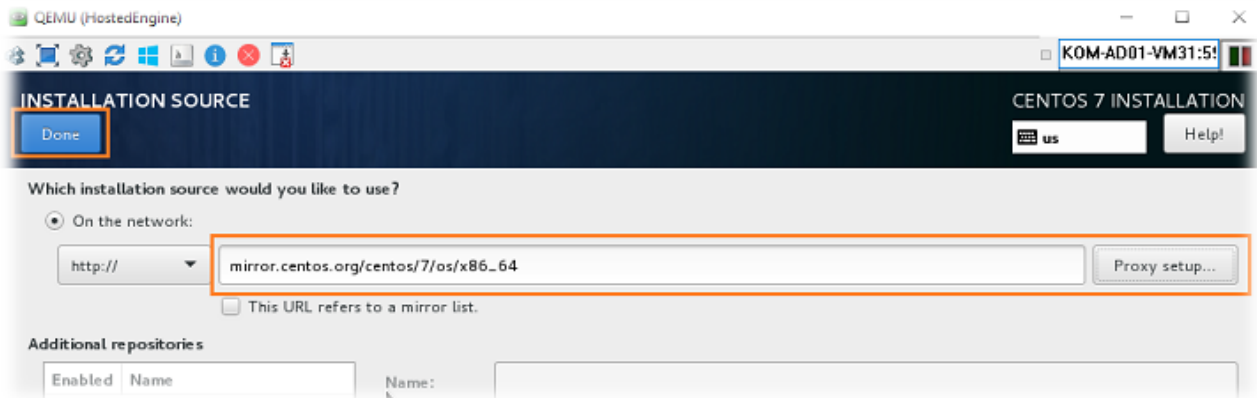
В моём случае для установки CentOS был выбран **NetInstall**-дистрибутив, который подразумевает загрузку необходимых для установки ОС пакетов из онлайн-репозитория CentOS. Поэтому нам придётся дополнительно его прописать в разделе установщика **INSTALLATION SOURCE**



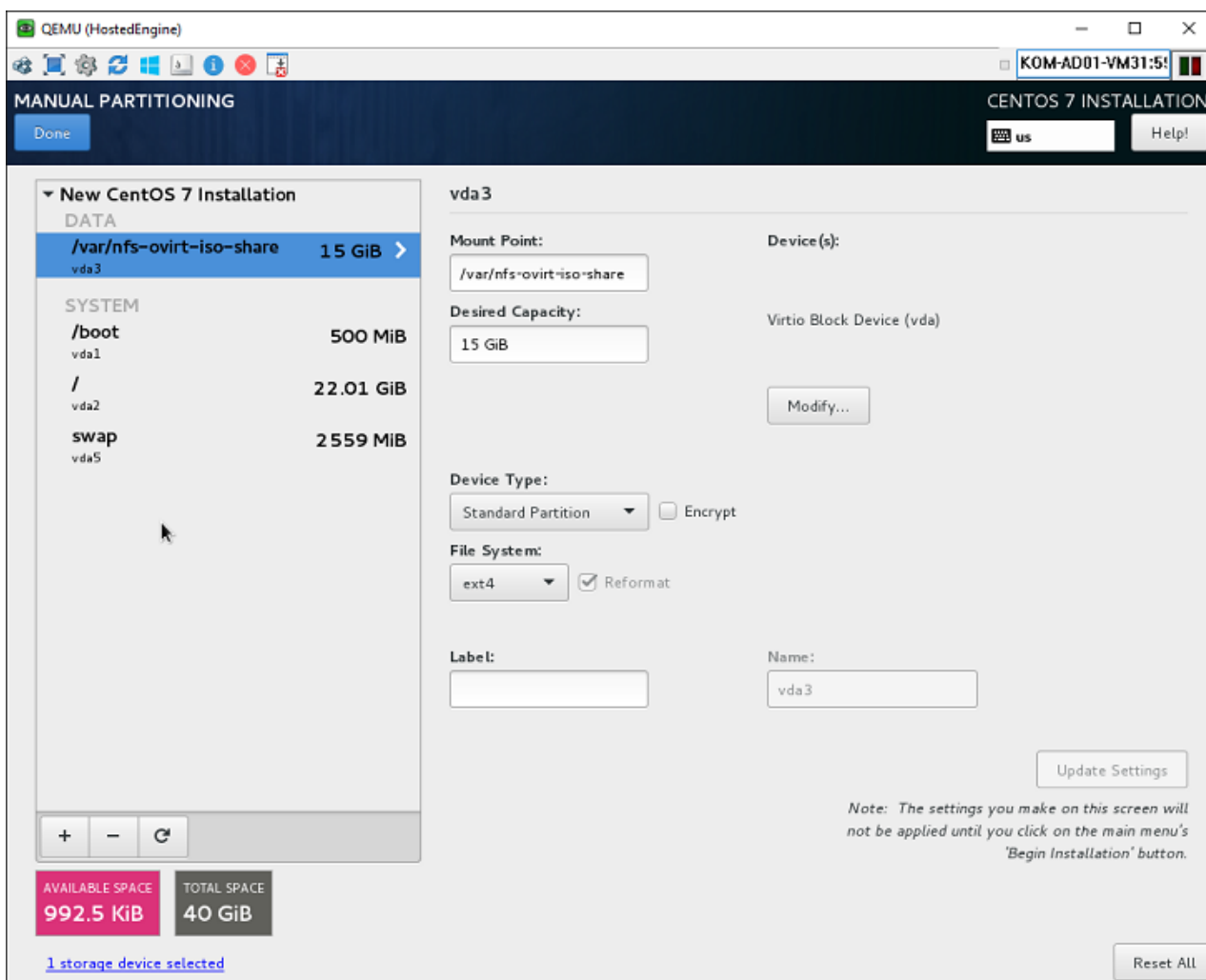
Добавим онлайн-репозиторий для CentOS 7 64-bit:

`http://mirror.centos.org/centos/7/os/x86_64/`

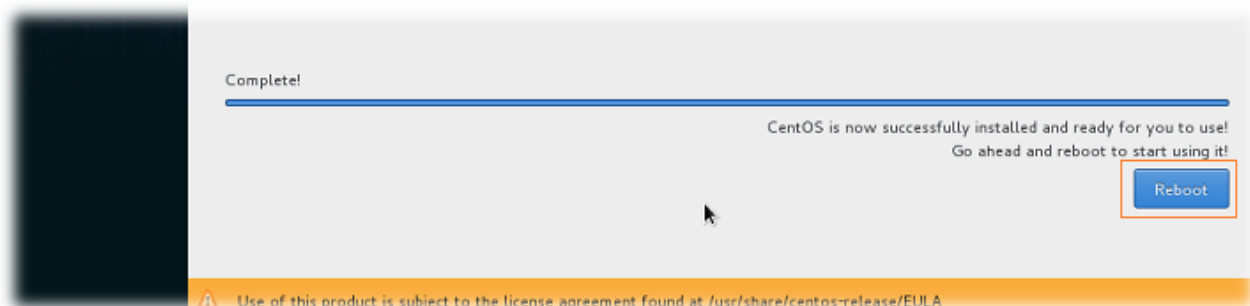
При этом не забываем указать данные прокси сервера, если он используется:



Разделы диска я оставил предложенные режимом автоматической разметки, только отказался от использования LVM и заменил файловую систему на **ext4**. Из общей ёмкости виртуального диска в **40 GB** под систему я оставил **25 GB**, и сделал отдельную точку монтирования размером в **15 GB** под **NFS**-шару для **oVirt ISO Domain**, как и планировал ранее.



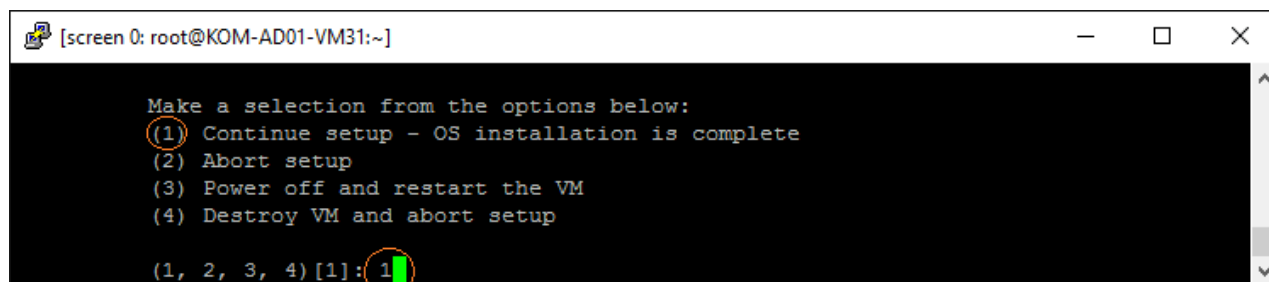
Дождёмся конца установки CentOS в виртуальную машину и в завершении выполним её перезагрузку:



Как только виртуальная система уйдёт в перезагрузку, наша VNC-сессия к этой виртуальной машине будет автоматически закрыта.

Перезагрузка виртуальной машины

Отправив виртуальную машину в перезагрузку, отключив тем самым VNC-сессию, возвращаемся в screen-сессию на нашем хосте виртуализации (KOM-AD01-VM31) и в ожидающем меню процедуры развёртывания выбираем **пункт 1**, инициируя тем самым процедуру повторной загрузки, только что отключенной виртуальной машины.



После этого программа развёртывания oVirt дожждётся от гипервизора хоста сигнал того, что виртуальная машина загружена и предложит нам снова подключиться к этой VM через VNC с использованием всё того же временного пароля.

Появившееся в самом конце меню выбора действий мы снова пока оставим в ожидании. Предполагается, что первый пункт будет выбран позднее, в том случае, если дальнейшая процедура развёртывания самого управляющего набора пакетов oVirt Engine внутри только что развёрнутой виртуальной CentOS будет завершена успешно. И опять же, если в процессе установки пакетов oVirt Engine внутри виртуальной машины возникнут проблемы, то можно будет вернуться к этому меню и выбрать пункт 3 для того, чтобы гипервизор хоста перезапустил эту VM, и мы смогли начать этот шаг заново. Пункты 2 и 4 могут пригодиться тем, кто решил всё бросить и пойти на рыбалку.

```
[screen 0: root@KOM-AD01-VM31:~]
(1, 2, 3, 4)[1]: 1

Please reboot or shutdown the VM.

Verifying shutdown...
[ INFO ] Creating VM
You can now connect to the VM with the following command:
/bin/remote-viewer vnc://localhost:5900
Use temporary password "8512gNcF" to connect to vnc console.
Please note that in order to use remote-viewer you need to be able to run graphic
al applications.
This means that if you are using ssh you have to supply the -Y flag (enables trus
ted X11 forwarding).
Otherwise you can run the command from a terminal in your preferred desktop enviro
nment.
If you cannot run graphical applications you can connect to the graphic console f
rom another host or connect to the serial console using the following command:
socat UNIX-CONNECT:/var/run/ovirt-vmconsole-console/3c6cfc11-9af1-4e68-89b8-100c2
b005832.sock,user=ovirt-vmconsole STDIO,raw,echo=0,escape=1
Please ensure that your Guest OS is properly configured to support serial console
according to your distro documentation.
Follow http://www.ovirt.org/Serial_Console_Setup#I_need_to_access_the_console_the
_old_way for more info.
If you need to reboot the VM you will need to start it manually using the command
:
hosted-engine --vm-start
You can then set a temporary password using the command:
hosted-engine --add-console-password
Please install and setup the engine in the VM.
You may also be interested in installing ovirt-guest-agent-common package in the
VM.

The VM has been rebooted.
To continue please install oVirt-Engine in the VM
(Follow http://www.ovirt.org/Quick_Start_Guide for more info).

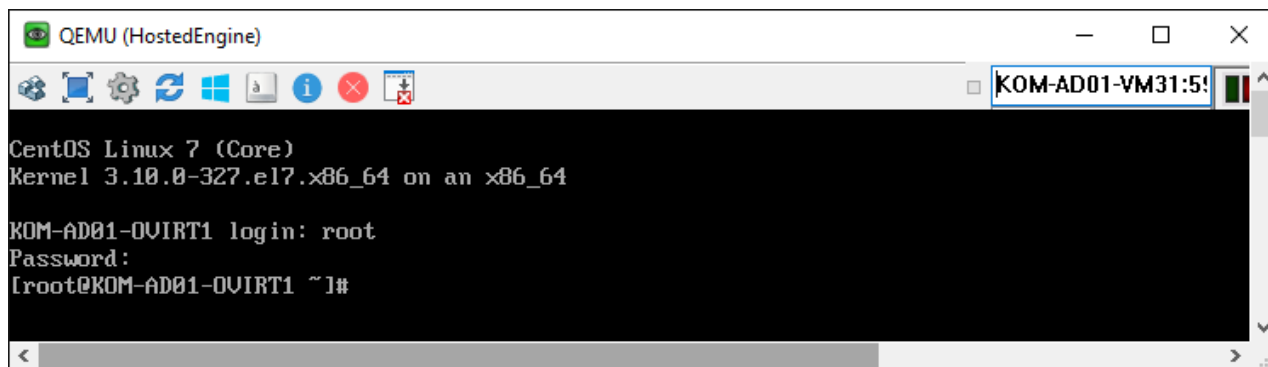
Make a selection from the options below:
(1) Continue setup - oVirt-Engine installation is ready and ovirt-engine service
is up
(2) Abort setup
(3) Power off and restart the VM
(4) Destroy VM and abort setup

(1, 2, 3, 4)[1]: █
```

Итак, снова оставим текущий сеанс в ожидании и создадим новое подключение к нашему хосту по протоколу VNC.

После-установочная настройка гостевой ОС в виртуальной машине

На этот раз подключившись к консоли виртуальной машины мы уже увидим приглашение входа только что установленной CentOS 7. Входим в систему с учётной записью пользователя **root** и паролем, который мы задали в ходе установки ОС на эту виртуальную машину.

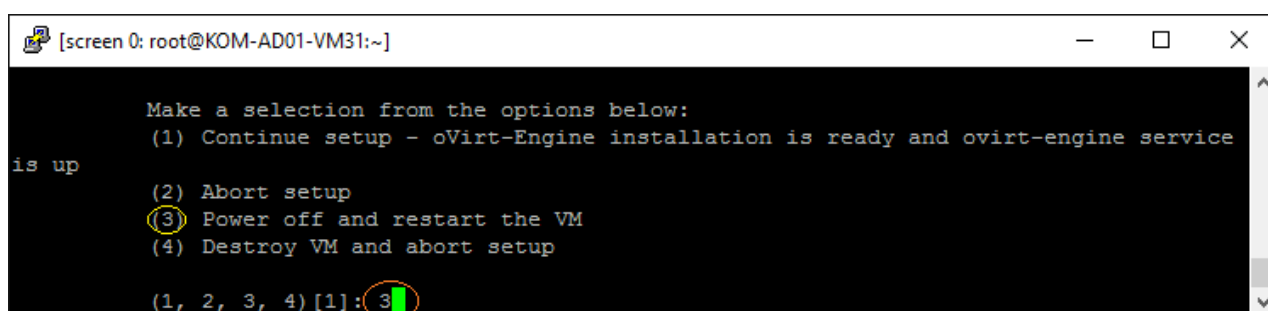


```
QEMU (HostedEngine)
KOM-AD01-VM31:5
CentOS Linux 7 (Core)
Kernel 3.10.0-327.el7.x86_64 on an x86_64
KOM-AD01-OVIRT1 login: root
Password:
[root@KOM-AD01-OVIRT1 ~]#
```

Кстати, теперь уже мы можем использовать **SSH** для доступа к нашему виртуальному серверу.

В гостевой ОС обновим все пакеты, отключим при необходимости поддержку IPv6 и выполним прочие пост-установочные процедуры, некоторые из которых рассматривались ранее.

После этого нам желательно будет перезагрузить ОС внутри виртуальной машины. Для этого внутри виртуальной машины вызовем команду **reboot**, а когда отключится VNC-сессия, вернёмся в screen-сессию на нашем хосте виртуализации и выберем пункт рестарта виртуальной машины – **Power off and restart the VM**.



```
[screen 0: root@KOM-AD01-VM31:~]
Make a selection from the options below:
(1) Continue setup - oVirt-Engine installation is ready and ovirt-engine service
is up
(2) Abort setup
(3) Power off and restart the VM
(4) Destroy VM and abort setup
(1, 2, 3, 4) [1]: 3
```

После того, как виртуальная машина снова станет доступна, мы можем подключиться к ней по VNC или SSH (кому как удобно) и выполнить процедуру развёртывания управляющих пакетов oVirt Engine. Однако в нашем примере, как вы наверное помните, мы хотим дополнительно организовать **NFS-шару** для возможности монтирования ISO-образов из этой шары к будущим виртуальным машинам (**oVirt ISO Domain**). Поэтому перед тем, как приступить к установке управляющих пакетов oVirt Engine в виртуальной машине, настроим в ней **NFS-сервер**.

Установка NFS-сервера в гостевой ОС виртуальной машины

Снова подключившись к виртуальной машине, устанавливаем NFS-сервер, который в дальнейшем потребуется нам в качестве **ISO Domain**. Можно конечно использовать для **oVirt ISO Domain** и выделенный NFS-сервер, но в моём случае его нет, поэтому все загрузочные/инсталляционные ISO-образы мы будем держать непосредственно в NFS-шаре внутри самого oVirt Engine сервера.

Устанавливаем пакеты для поддержки NFS:

```
# yum install nfs-utils
```

Проверяем наличие каталога, который мы указали в процессе установки ОС:

```
# ls -la /var/nfs-ovirt-iso-share/
```

```
total 24
drwxr-xr-x.  3 root root  4096 Jul 26 15:55 .
drwxr-xr-x. 20 root root  4096 Jul 26 15:52 ..
drwx-----.  2 root root 16384 Jul 26 15:15 lost+found
```

Создаём каталог под общую сетевую папку и выставляем на него разрешения (36:36 это UID/GID vds:m:kvm) в соответствии с документом [oVirt - Troubleshooting NFS Storage Issues](#):

```
# mkdir -p /var/nfs-ovirt-iso-share/files
# chown 36:36 /var/nfs-ovirt-iso-share/files
# chmod -R 755 /var/nfs-ovirt-iso-share/files
```

Запускаем службы **NFS** и включаем их автозагрузку:

```
# systemctl start rpcbind
# systemctl start nfs-server
# systemctl enable rpcbind
# systemctl enable nfs-server
```

Включаем разрешающие правила брандмауэра **firewalld** для возможности удалённого подключения к NFS-серверу:

```
# firewall-cmd --permanent --zone=public --add-service=nfs
# firewall-cmd --permanent --zone=public --add-service=mountd
# firewall-cmd --permanent --zone=public --add-service=rpc-bind
# firewall-cmd --reload
```

Теперь можно переходить к развёртыванию управляющих пакетов oVirt Engine на гостевую ОС внутри нашей виртуальной машины.

Установка компонент oVirt Engine в гостевой ОС виртуальной машины

Добавляем информацию о репозиториях oVirt и Epeel, а затем сразу выполняем установку и запуск развёртывания управляющих компонент oVirt Engine:

```
# yum install http://resources.ovirt.org/pub/yum-repo/ovirt-release40.rpm
# yum install epel-release
# yum install ovirt-engine
# engine-setup
```

Будет запущен мастер развёртывания компонент oVirt Engine, где нам тоже потребуется ответить на ряд вопросов. На первые четыре вопроса из секции **PRODUCT OPTIONS** можно оставить предложенные по умолчанию утвердительные ответы

```
root@KOM-AD01-OVIRT1:~  
[root@KOM-AD01-OVIRT1 ~]# engine-setup  
[ INFO ] Stage: Initializing  
[ INFO ] Stage: Environment setup  
          Configuration files: ['/etc/ovirt-engine-setup.conf.d/10-packaging-jboss.conf', '  
/etc/ovirt-engine-setup.conf.d/10-packaging.conf']  
          Log file: /var/log/ovirt-engine/setup/ovirt-engine-setup-20160726161720-r4zvik.lo  
g  
          Version: otopi-1.5.1 (otopi-1.5.1-1.el7.centos)  
[ INFO ] Stage: Environment packages setup  
[ INFO ] Stage: Programs detection  
[ INFO ] Stage: Environment setup  
[ INFO ] Stage: Environment customization  
  
--== PRODUCT OPTIONS ==--  
  
Configure Engine on this host (Yes, No) [Yes]:  
Configure WebSocket Proxy on this host (Yes, No) [Yes]:  
Please note: Data Warehouse is required for the engine. If you choose to not conf  
igure it on this host, you have to configure it on a remote host, and then configure the en  
gine on this host so that it can access the database of the remote Data Warehouse host.  
Configure Data Warehouse on this host (Yes, No) [Yes]:  
Configure VM Console Proxy on this host (Yes, No) [Yes]:  
  
--== PACKAGES ==--  
  
[ INFO ] Checking for product updates...  
[ INFO ] No product updates found
```

В разделе вопросов о сетевой конфигурации оставим предложенное по умолчанию FQDN имя нашего виртуального сервера и разрешим настройку необходимых правил брандмауэра:

```
root@KOM-AD01-OVIRT1:~  
  
--== NETWORK CONFIGURATION ==--  
  
Host fully qualified DNS name of this server [kom-ad01-ovirt1.holding.com]:  
Setup can automatically configure the firewall on this system.  
Note: automatic configuration of the firewall may overwrite current settings.  
Do you want Setup to configure the firewall? (Yes, No) [Yes]:  
[ INFO ] firewalld will be configured as firewall manager.
```

В секции конфигурации базы данных оставим предложенные по умолчанию значения, чтобы программа установки развернула и сконфигурировала локальный экземпляр **postgresql**:

```
root@KOM-AD01-OVIRT1:~  
  
--== DATABASE CONFIGURATION ==--  
  
Where is the DWH database located? (Local, Remote) [Local]:  
Setup can configure the local postgresql server automatically for the DWH to run.  
This may conflict with existing applications.  
Would you like Setup to automatically configure postgresql and create DWH databas  
e, or prefer to perform that manually? (Automatic, Manual) [Automatic]:  
Where is the Engine database located? (Local, Remote) [Local]:  
Setup can configure the local postgresql server automatically for the engine to r  
un. This may conflict with existing applications.  
Would you like Setup to automatically configure postgresql and create Engine data  
base, or prefer to perform that manually? (Automatic, Manual) [Automatic]:
```


В конфигурации Engine у нас будет запрошен пароль учётной записи **admin**. Здесь важно указать тот же пароль, который мы указывали ранее в начале процедуры развёртывания Hosted Engine на хосте (ранее он фигурировал в нашем описании как Pa\$sw0rd). В противном случае Engine может не запуститься. Значения в следующих паре вопросов я также оставил предложенные по умолчанию.

```
root@KOM-AD01-OVIRT1:~  
  
---- OVIRT ENGINE CONFIGURATION ----  
  
Engine admin password:  
Confirm engine admin password:  
[WARNING] Password is weak: it is based on a dictionary word  
Use weak password? (Yes, No) [No]: Yes  
Application mode (Virt, Gluster, Both) [Both]:
```

```
root@KOM-AD01-OVIRT1:~  
  
---- STORAGE CONFIGURATION ----  
  
Default SAN wipe after delete (Yes, No) [No]:
```

Далее последует вопрос о конфигурации **PKI**. Как я понял, в процессе установки oVirt разворачивает локальную службу сертификации и выдает из неё сертификаты для внутренних нужд компонент oVirt, в частности для защиты соединения с веб-порталами oVirt. Имя организации для сертификатов предлагается при этом взятое из доменной части FQDN-имени сервера Hosted Engine. Оставляем значение по умолчанию, так как менять его у нас особых причин нету.

```
root@KOM-AD01-OVIRT1:~  
  
---- PKI CONFIGURATION ----  
  
Organization name for certificate [holding.com]:
```

В секции конфигурации Apache оставим предложенные по умолчанию значения вопросов о необходимости установки и автоматического запуска веб-сервера.

```
root@KOM-AD01-OVIRT1:~  
  
---- APACHE CONFIGURATION ----  
  
Setup can configure the default page of the web server to present the application  
home page. This may conflict with existing applications.  
Do you wish to set the application as the default page of the web server? (Yes, No)  
o) [Yes]:  
Setup can configure apache to use SSL using a certificate issued from the interna  
l CA.  
Do you wish Setup to configure that, or prefer to perform that manually? (Automat  
ic, Manual) [Automatic]:
```

В секции **SYSTEM CONFIGURATION** нам будет задан вопрос хотим ли мы создать **ISO Domain** для oVirt на этой виртуальной машине. Утвердительно ответим на этот вопрос, так как ранее мы подготовили службу NFS-сервера в этой системе. Затем

укажем путь к каталогу, который мы ранее подготовили под NFS-шару (/var/nfs-ovirt-iso-share/files) и установим разрешения на доступ к этой шаре. В нашем примере разрешается доступ на запись (rw) всем NFS-клиентам из нашей серверной подсети (10.1.0.0/24). Имя ISO domain можно задать любое или оставить предложенное (так оно будет отображаться в дальнейшем в консоли oVirt).

```
root@KOM-AD01-OVIRT1:~  
  
---- SYSTEM CONFIGURATION ----  
  
Configure an NFS share on this server to be used as an ISO Domain? (Yes, No) [No]  
Yes  
Local ISO domain path [/var/lib/exports/iso]: /var/nfs-ovirt-iso-share/files  
  
Please provide the ACL for the Local ISO domain.  
See the exports(5) manpage for the format.  
Examples:  
- To allow access for host1, host2 and host3, input: host1(rw) host2(rw) host3(rw)  
- To allow access to the entire Internet, input: *(rw)  
  
For more information, see: http://www.ovirt.org/Troubleshooting_NFS_Storage_Issue  
  
Local ISO domain ACL: 10.1.0.0/24(rw)  
Local ISO domain name [ISO_DOMAIN]:
```

В следующей секции нас попросят выбрать режим хранения сведений в исторической базе данных **Data Warehouse**. Чтобы понять различие между двумя предложенными вариантами можно заглянуть в документ [Red Hat Virtualization 4.0 - Data Warehouse Guide](#). Так как мы устанавливаем базу данных DWH на том же сервере, где будет работать и оперативная база данных oVirt и даже сами компоненты oVirt Engine, рекомендуемым вариантом будет использование упрощённого формата хранения исторических данных (**Basic**), которые, к слову, используются при построении отчётов. А на тему развёртывания и настройки отчётности в oVirt мы поговорим отдельно в одной из следующих частей цикла заметок о развёртывании oVirt.

```
root@KOM-AD01-OVIRT1:~  
  
---- MISC CONFIGURATION ----  
  
Please choose Data Warehouse sampling scale:  
(1) Basic  
(2) Full  
(1, 2) [1]:
```

В сводной секции посмотрим все выбранные параметры развёртывания и нажмём Enter для запуска процесса установки oVirt Engine:

```
root@KOM-AD01-OVIRT1:~  
  
----- END OF CONFIGURATION -----  
[ INFO ] Stage: Setup validation  
[WARNING] Less than 16384MB of memory is available  
  
----- CONFIGURATION PREVIEW -----  
  
Application mode : both  
Default SAN wipe after delete : False  
Firewall manager : firewalld  
Update Firewall : True  
Host FQDN : kom-ad01-ovirt1.holding.com  
Engine database secured connection : False  
Engine database host : localhost  
Engine database user name : engine  
Engine database name : engine  
Engine database port : 5432  
Engine database host name validation : False  
DWH database secured connection : False  
DWH database host : localhost  
DWH database user name : ovirt_engine_history  
DWH database name : ovirt_engine_history  
DWH database port : 5432  
DWH database host name validation : False  
Engine installation : True  
NFS setup : True  
PKI organization : holding.com  
NFS export ACL : 10.1.0.0/24(rw)  
NFS mount point : /var/nfs-ovirt-iso-share/files  
Configure local Engine database : True  
Set application as default page : True  
Configure Apache SSL : True  
DWH installation : True  
Configure local DWH database : True  
Engine Host FQDN : kom-ad01-ovirt1.holding.com  
Configure VMConsole Proxy : True  
Configure WebSocket Proxy : True  
  
Please confirm installation settings (OK, Cancel) [OK]:
```

Дождёмся успешного выполнения всех основных этапов установки oVirt Engine:

Теперь перебираемся в screen-сессию на наш первый хост виртуализации (KOM-AD01-VM31), где ранее была запущена процедура развёртывания oVirt Hosted Engine и в ожидающем ответа меню выбираем пункт **(1) Continue setup ...**, подтверждая тем самым факт того, что компоненты oVirt Engine внутри виртуальной машины успешно установлены.

```
[screen 0: root@KOM-AD01-VM31:~]

Make a selection from the options below:
(1) Continue setup - oVirt-Engine installation is ready and ovirt-engine service
is up
(2) Abort setup
(3) Power off and restart the VM
(4) Destroy VM and abort setup

(1, 2, 3, 4) [1]: 1

Checking for oVirt-Engine status at KOM-AD01-OVIRT1.holding.com...
[ INFO ] Engine replied: DB Up!Welcome to Health Status!
[ INFO ] Acquiring internal CA cert from the engine
[ INFO ] The following CA certificate is going to be used, please immediately interrupt if
not correct:
[ INFO ] Issuer: C=US, O=holding.com, CN=kom-ad01-ovirt1.holding.com.11885,
Subject: C=US, O=holding.com, CN=kom-ad01-ovirt1.holding.com.11885, Fingerprint (SHA-1):
B2FB0...4B
[ INFO ] Connecting to the Engine
[ INFO ] Waiting for the host to become operational in the engine. This may take several m
inutes...
[ INFO ] Still waiting for VDSM host to become operational...
[ INFO ] The VDSM Host is now operational
[ INFO ] Saving hosted-engine configuration on the shared storage domain
Please shutdown the VM allowing the system to launch it as a monitored service.
The system will wait until the VM is down.
```

Процесс развёртывания попросит нас выключить ОС в виртуальной машине с Engine. Выключаем виртуальный сервер, выполнив, например, команду **poweroff**

```
PuTTY (inactive)

[ INFO ] Stage: Pre-termination
[ INFO ] Stage: Termination
[ INFO ] Execution of setup completed successfully
[root@KOM-AD01-OVIRT1 ~]# poweroff
```

После этого процесс развёртывания oVirt Hosted Engine в screen-сессии на нашем первом хосте виртуализации (KOM-AD01-VM31) будет автоматически завершён, а виртуальная машина Engine снова запустится.

```
[screen 0: root@KOM-AD01-VM31:~]

[ INFO ] Saving hosted-engine configuration on the shared storage domain
Please shutdown the VM allowing the system to launch it as a monitored service.
The system will wait until the VM is down.
[ INFO ] Enabling and starting HA services
[ INFO ] Stage: Clean up
[ INFO ] Generating answer file '/var/lib/ovirt-hosted-engine-setup/answers/answers-201607
26165944.conf'
[ INFO ] Generating answer file '/etc/ovirt-hosted-engine/answers.conf'
[ INFO ] Stage: Pre-termination
[ INFO ] Stage: Termination
[ INFO ] Hosted Engine successfully set up
[root@KOM-AD01-VM31 ~]#
```

Итак, процесс развёртывания oVirt Hosted Engine на первом хосте завершён и теперь нужно выполнить развёртывание ещё на одном хосте, чтобы сделать конфигурацию Engine высоко-доступной.

Развёртывание oVirt Hosted Engine на втором хосте

Для развёртывания мы будем использовать второй хост с именем **KOM-AD01-VM32** с аналогичными настройками доступа к общим дискам на СХД и сетевой конфигурацией. Выполняем на хосте все те же предварительные настройки, что ранее делали на первом хосте (см. пункт "Подготовка хостов к развёртыванию oVirt Hosted Engine").

После того, как хост подготовлен, выполняем установку пакета **ovirt-hosted-engine-setup** и запускаем развёртывание **hosted-engine** в screen-сессии:

```
# yum install ovirt-hosted-engine-setup screen -y
# screen -RD
# hosted-engine --deploy
```

Здесь снова нам нужно будет ответить на ряд вопросов. Сначала в разделе конфигурации хранилища выберем тот же диск на котором ранее мы развернули виртуальную машину Hosted Engine. Статус этого диска в этот раз будет отображаться как **used**.

```
[root@KOM-AD01-VM32 ~]# hosted-engine --deploy
/usr/lib/python2.7/site-packages/ovirt_hosted_engine_ha/lib/storage_backends.py:15: DeprecationWarning: vdscli uses xmlrpc. since ovirt 3.6 xmlrpc is deprecated, please use vdsm.jsonrpcvdscli
import vdsm.vdscli
[ INFO ] Stage: Initializing
[ INFO ] Generating a temporary VNC password.
[ INFO ] Stage: Environment setup
During customization use CTRL-D to abort.
Continuing will configure this host for serving as hypervisor and create a VM where you have to install the engine afterwards.
Are you sure you want to continue? (Yes, No) [Yes]:
[ INFO ] Hardware supports virtualization
Configuration files: []
Log file: /var/log/ovirt-hosted-engine-setup/ovirt-hosted-engine-setup-20160727101526-rvogk6.log
Version: otopi-1.5.1 (otopi-1.5.1-1.el7.centos)
[ INFO ] Stage: Environment packages setup
[ INFO ] Stage: Programs detection
[ INFO ] Stage: Environment setup
[ INFO ] Generating libvirt-spice certificates
[WARNING] Cannot locate gluster packages, Hyper Converged setup support will be disabled.
[ INFO ] Please abort the setup and install vdsm-gluster, gluster-server >= 3.7.2 and restart vdsmd service in order to gain Hyper Converged setup support.
[ INFO ] Stage: Environment customization

--- STORAGE CONFIGURATION ---

Please specify the storage you would like to use (glusterfs, iscsi, fc, nfs3, nfs4) [nfs3]:fc
The following luns have been found on the requested target:
    [1] 360002ac000000000000000160000cec9      2000GiB  3PARdata
VV
    status: free, paths: 4 active
    [2] 360002ac0000000000000001e0000cec9      90GiB   3PARdata
VV
    status: used, paths: 4 active

Please select the destination LUN (1, 2) [1]: 2
```

Затем программа установки обнаружит на выбранном диске так называемый **Data Domain** и спросит, хотим ли мы присоединить данный хост к нему. Ответим утвердительно, а на следующий вопрос о присвоении идентификатора хоста **Host ID** оставим вычисленное автоматически предлагаемое значение – **2**. Основную массу других параметров, которые мы задали для настройки Hosted Engine в процессе развёртывания первого хоста, программа установки считает из конфигурационного файла на диске с виртуальной машиной Hosted Engine.

```
[screen 0: root@KOM-AD01-VM32:~]

The specified storage location already contains a data domain. Is this an additional host setup (Yes, No) [Yes]?
[ INFO ] Installing on additional host
Please specify the Host ID [Must be integer, default: 2]:

---- SYSTEM CONFIGURATION ----

[WARNING] A configuration file must be supplied to deploy Hosted Engine on an additional host.
[ INFO ] Answer file successfully loaded

---- NETWORK CONFIGURATION ----

[ INFO ] Additional host deployment, firewall manager is 'iptables'
The following CPU types are supported by this host:
- model_Penryn: Intel Penryn Family
- model_Conroe: Intel Conroe Family
```

Далее нам потребуется два раза ввести пароль учётной записи **admin**, который мы задали в процессе развертывания первого хоста и повторно использовали в процессе развертывания oVirt Engine (ранее он фигурировал в нашем описании как Pa\$sw0rd). Затем введём имя нашего второго хоста, под которым этот хост будет отображаться в дальнейшем в веб-консоли oVirt и его полное доменное имя FQDN:

```
[screen 0: root@KOM-AD01-VM32:~]

---- HOSTED ENGINE CONFIGURATION ----

Enter engine admin password:
Confirm engine admin password:
Enter the name which will be used to identify this host inside the Administrator Portal [hosted_engine_2]: KOM-AD01-VM32
[ INFO ] Stage: Setup validation
Please provide the fully qualified domain name of this host.
Note: The engine VM and all the other hosts should be able to correctly resolve it.
Host FQDN: [kom-ad01-vm32.holding.com]:
```

Далее будет выполнен процесс непосредственной настройки компонент oVirt на данном хосте с обновлением текущей конфигурации Hosted Engine. В конце мы должны получить сообщение об успешной установке Hosted Engine.


```
--== CONFIGURATION PREVIEW ==--

Engine FQDN                : KOM-AD01-OVIRT1.holding.com
Bridge name                 : ovirtmgmt
Host address                : kom-ad01-vm32.holding.com
SSH daemon port             : 22
Firewall manager            : iptables
Gateway address             : 10.1.0.1
Host name for web application : KOM-AD01-VM32
Storage Domain type         : fc
Host ID                     : 2
LUN ID                      : 360002ac000000000000000001e0000cec9
Image size GB               : 40
Console type                : vnc
Memory size MB              : 4096
MAC address                 : 00:16:3e:29:8e:29
Boot type                   : disk
Number of CPUs              : 2
CPU Type                    : model_Penryn
```

```
[ INFO ] Stage: Transaction setup
[ INFO ] Stage: Misc configuration
[ INFO ] Stage: Package installation
[ INFO ] Stage: Misc configuration
[ INFO ] Configuring libvirt
[ INFO ] Configuring VDSM
[ INFO ] Starting vdsm
[ INFO ] Configuring VM
[ INFO ] Updating hosted-engine configuration
[ INFO ] Stage: Transaction commit
[ INFO ] Stage: Closing up
[ INFO ] Acquiring internal CA cert from the engine
[ INFO ] The following CA certificate is going to be used, please immediately interrupt if
not correct:
[ INFO ] Issuer: C=US, O=holding.com, CN=kom-ad01-ovirt1.holding.com.11885,
Subject: C=US, O=holding.com, CN=kom-ad01-ovirt1.holding.com.11885, Fingerprint (SHA-1):
B2C...
[ INFO ] Connecting to the Engine
[ INFO ] Waiting for the host to become operational in the engine. This may take several m
inutes...
[ INFO ] Still waiting for VDSM host to become operational...
[ INFO ] The VDSM Host is now operational
[ INFO ] Enabling and starting HA services
[ INFO ] Stage: Clean up
[ INFO ] Generating answer file '/var/lib/ovirt-hosted-engine-setup/answers/answers-201607
27104018.conf'
[ INFO ] Generating answer file '/etc/ovirt-hosted-engine/answers.conf'
[ INFO ] Stage: Pre-termination
[ INFO ] Stage: Termination
[ INFO ] Hosted Engine successfully set up
```

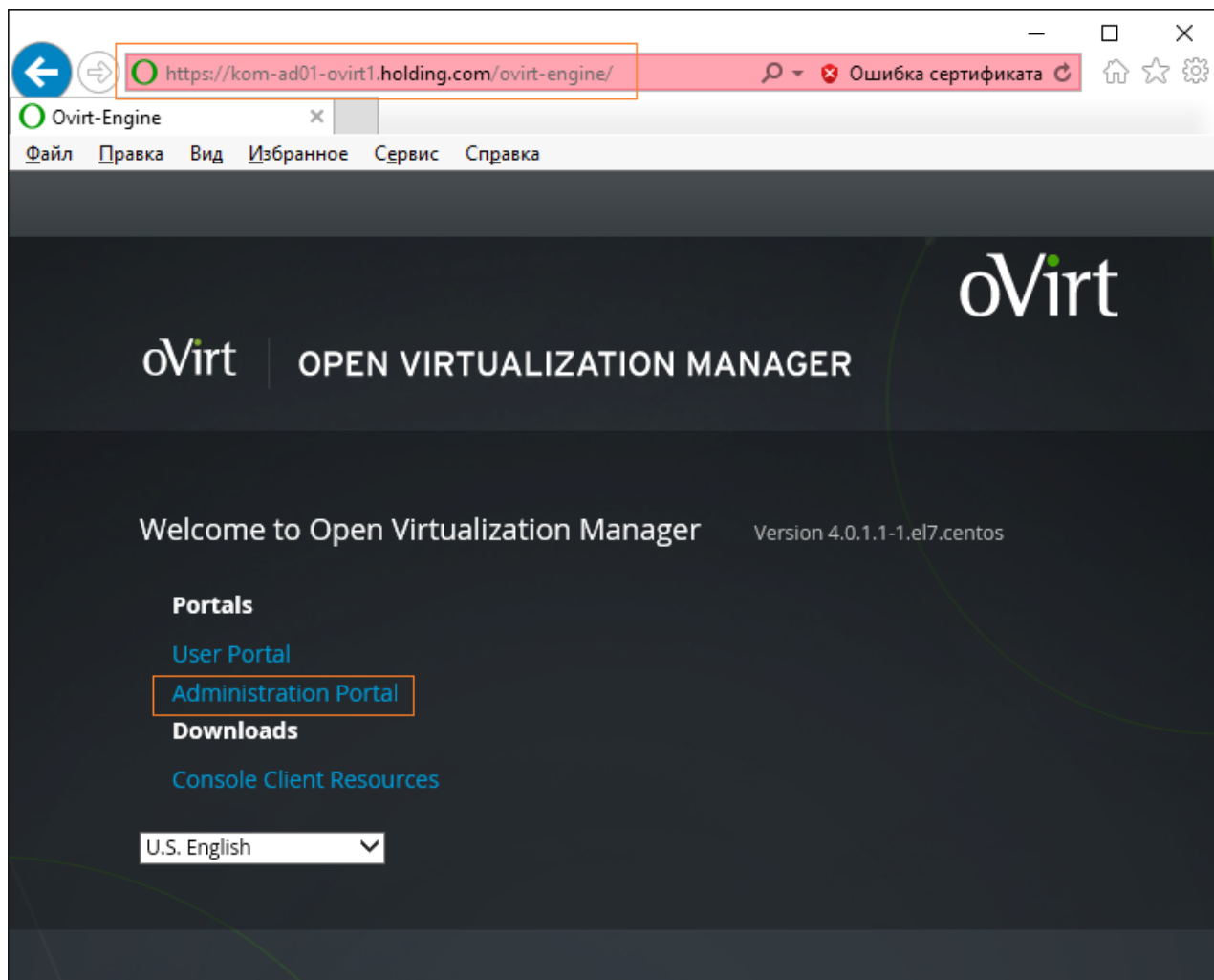
```
[root@KOM-AD01-VM32 ~]#
```

Теперь наш экземпляр oVirt Hosted Engine стал высоко-доступным, то есть при отключении хоста, на котором запущена виртуальная машина с Hosted Engine, будет происходить автоматический перезапуск виртуальной машины Hosted Engine на втором доступном хосте.

Портал администрирования oVirt Engine

Проверим доступ к основному инструменту администрирования инфраструктуры oVirt – веб-порталу. Для этого перейдём в веб-браузере по ссылке, которую мы получили ранее в завершении процесса развёртывания oVirt Engine на виртуальной машине. В нашем случае это:

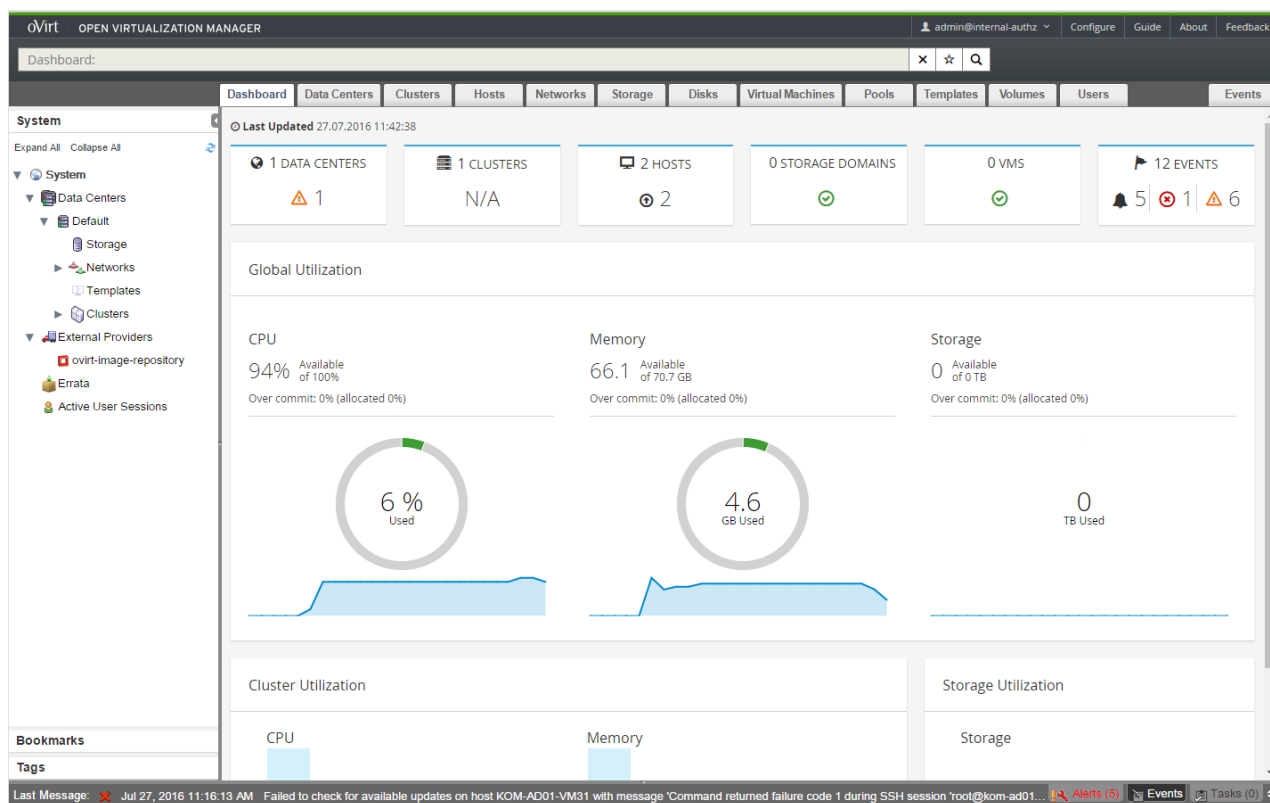
<https://kom-ad01-ovirt1.holding.com/ovirt-engine/>



На данной стартовой веб-странице oVirt Engine мы сможем увидеть текущую версию oVirt Engine и ссылки на портал администрирования oVirt (**Administration Portal**) и пользовательский портал (**User Portal**). Для входа на портал администрирования нам потребуется ввести учётные данные учётной записи **admin**, пароль для которой мы задали ранее в процессе развертывания первого хоста (ранее он фигурировал в нашем описании как Pa\$sw0rD).



Перед нами появится сводная статусная информация о состоянии ключевых показателей нашей виртуализации oVirt на открывающейся по умолчанию закладке **Dashboard**.



Кстати, замечено, что в IE11 Dashboard иногда может не отображаться (бесконечный статус Loading...), при этом, например в Firefox или Chrome страница загружается сразу. Также могу отметить, что на десктопе Ubuntu 16.04 в Firefox веб-консоль периодически может подтормаживать, вызывая сильную нагрузку на процессор.

Послеустановочная модификация конфигурации multipathd

Если в процессе подготовки физических серверов под хосты oVirt ранее использовалась настройка multipath-подключений к СХД, то нужно учесть тот факт, что в процессе развёртывания компонент oVirt на эти серверы происходит модификация конфигурационного файла `/etc/multipath.conf`. При этом простая обратная правка этого файла не имеет смысла, так как управляющий код **VDSM** всё равно будет переписывать содержимое этого файла. Причём в прежних версиях oVirt такая ситуация имела безусловный характер ([Red Hat Bugzilla - Bug 1076531 - vdsmd overwrites multipath.conf at every startup if vdsmd installed file was replaced](#)). По началу такая ситуация меня несколько смутила, но задав вопрос в официальной мейл-группе oVirt, я достаточно оперативно получил информацию о возможном варианте решения этой проблемы. Вообще инженер Red Hat дал понять то, что без веских на то причин, модификация файла `multipath.conf` в сторону отклонения от варианта генерируемого VDSM - не самая лучшая идея. Если же всё же вы решили что в `multipath.conf` должны быть ваши настройки, то для того, чтобы VDSM не изменял контент файла после ваших правок, во второй строке файла (сразу после строки типа `# VDSM REVISION 1.3`) нужно добавить строку `# VDSM PRIVATE`, то есть начало файла должно выглядеть следующим образом:

```
# VDSM REVISION 1.3
# VDSM PRIVATE
...
```

После чего заставить службу **multipathd** перечитать настройки:

```
# systemctl reload multipathd
```

А ещё лучше выполнить проверочную перезагрузку сервера, чтобы убедиться в том, что в процессе запуска системы и старта всех служб настроенная нами multipath-конфигурация работает штатно. Аналогичным образом нужно настроить конфигурационный файл на всех хостах, входящих в кластер oVirt и подключенных к дисковым ресурсам одной и той же СХД.

Ручной запуск VM Hosted Engine

В некоторых ситуациях может получиться так, что веб-интерфейс управления oVirt недоступен, так как виртуальная машина с oVirt Hosted Engine выключена (например VM была выключена непосредственно из гостевой ОС). В таком случае потребуется выполнить ручной запуск VM с Hosted Engine. Сделать это можно с помощью утилиты управления oVirt на любом из хостов, на которых были развёрнуты компоненты Hosted Engine. Для того, чтобы запустить виртуальную машину с oVirt Hosted Engine выполним на хосте команду:

```
# hosted-engine --vm-start
```

Дополнительно, чтобы выяснить какие виртуальные машины выполняются на хосте, можно воспользоваться утилитой **vdsClient**:

```
# vdsClient -s 127.0.0.1 list table
```

661bafca-e9c3-4191-99b4-285ff8553488	30897	KOM-AD01-TEST1	Up	10.1.0.25
3c5cfc11-9af1-4e68-89b8-101c2b005832	13331	HostedEngine	Down	

Как видим, на данном хосте в текущий момент времени хостится две виртуальных машины - KOM-AD01-TEST1 (запущена) и HostedEngine (выключена). Посмотреть статус виртуальных машин на другом хосте можно указав имя хоста в ключе -s.

Обратите внимание на то, что после запуска VM с Hosted Engine, веб-портал администрирования oVirt доступен не сразу, так как для требуется некоторое время на запуск необходимых веб-служб.

На данном этапе можно считать, что минимальные действия по развёртыванию инфраструктуры oVirt в конфигурации Hosted Engine выполнены, и теперь у нас есть базовый инструментарий для администрирования виртуальных машин на базе открытого гипервизора KVM. В следующей части мы рассмотрим процедуру замены цифрового сертификата, используемого для защиты HTTP соединений с порталами oVirt, чтобы избежать предупреждений безопасности веб-браузеров.

