

ООО «С-Терра СиЭсПи»
124498, г. Москва, Зеленоград, Георгиевский проспект,
дом 5, помещение I, комната 33
Телефон/Факс: +7 (499) 940 9061
Эл.почта: information@s-terra.com
Сайт: <http://www.s-terra.com>



Программный комплекс С-Терра Шлюз. Версия 4.1

Руководство администратора

Развертывание виртуального шлюза на KVM

РЛКЕ.00009-01 90 03

22.05.2015

Содержание

Развертывание виртуального шлюза на KVM	3
Введение	3
Используемые версии ПО	4
Настройка интерфейсов	4
Импорт в KVM	5
Карта интерфейсов	7
Управление виртуальной машиной при помощи virsh	7
Монтирование USB Flash	8

Развертывание виртуального шлюза на KVM

Введение

Данный документ относится только к «Программному комплексу С-Терра Шлюз. Версия 4.1», работающему в виртуальной среде.

В документе описывается развертывание «Программного комплекса С-Терра Шлюз» из OVA-шаблона на KVM (OVA – часть стандарта OVF, содержит все файлы виртуальной машины в одном файле).

В данном документе описана работа KVM, развернутого на базе Debian 7.

Для работы потребуется компакт-диск, входящий в комплект поставки, на котором находится образ виртуальной машины с инсталлированным «Программным комплексом С-Терра Шлюз. Версия 4.1» в формате OVA – «С-Терра Виртуальный Шлюз CP KC1. Версия 4.1. Релиз 14905» или «С-Терра Виртуальный Шлюз ST KC1. Версия 4.1. Релиз 14905».

После развертывания виртуальной машины из OVA-шаблона и установки необходимых настроек выполните инициализацию «Программного комплекса С-Терра Шлюз 4.1» в соответствии с документом «Инициализация S-Terra Gate на вычислительных системах архитектуры Intel x86/x86-64» ([Initialization_gate_guide.pdf](#)).

Дальнейшие настройки шлюза выполняются в соответствии с документацией на «Программный комплекс С-Терра Шлюз. Версия 4.1. Руководство администратора».

Используемые версии ПО

На хосте используется следующее программное обеспечение:

1. Информация о системе:

```
root@debian:~# uname -a
Linux debian 3.2.0-4-amd64 #1 SMP Debian 3.2.65-1 x86_64 GNU/Linux
root@debian:~# cat /etc/debian_version
7.8
```

2. Используемые версии ПО для работы гипервизора:

```
root@debian:~# virsh version
Compiled against library: libvir 0.9.12
Using library: libvir 0.9.12
Using API: QEMU 0.9.12
Running hypervisor: QEMU 1.1.2
root@debian:~# dpkg -l | grep qemu-kvm
ii  qemu-kvm                    1.1.2+dfsg-6+deb7u6
amd64  Full virtualization on x86 hardware
root@debian:~# dpkg -l | grep virt-manager
ii  virt-manager                 0.9.1-4
all    desktop application for managing virtual machines
root@debian:~# dpkg -l | grep libvirt
ii  libvirt-bin                 0.9.12.3-1+deb7u1
amd64  programs for the libvirt library
ii  libvirt0                    0.9.12.3-1+deb7u1
amd64  library for interfacing with different virtualization systems
ii  python-libvirt              0.9.12.3-1+deb7u1
amd64  libvirt Python bindings
```

3. Версии дополнительного ПО:

```
root@debian:~# dpkg -l | grep bridge-utils
ii  bridge-utils                1.5-6
amd64  Utilities for configuring the Linux Ethernet bridge
```

Настройка интерфейсов

Необходимо настроить сетевые мосты для подключения виртуальных машин к сети:

1. Настройте бриджи на хосте для работы виртуальных машин:

```
root@debian:~# vi /etc/network/interfaces
auto br0
iface br0 inet static
    address 192.168.1.254
    netmask 255.255.255.0
```

```

gateway 192.168.1.1
bridge_ports eth0
bridge_fd 0
bridge_waitport 0
bridge_stp off
auto br1
iface br1 inet manual
    bridge_ports eth1
    bridge_fd 0
    bridge_waitport 0
    bridge_stp off
auto br2
iface br2 inet manual
    bridge_ports eth2
    bridge_fd 0
    bridge_waitport 0
    bridge_stp off

```

Где:

`bridge_ports <eth>` – перечисление всех интерфейсов бриджа;

`bridge_stp off` – отключить Spanning Tree Protocol;

`bridge_fd 0` – отключить задержку переправки;

`bridge_waitport 0` – отключить задержку до того как порт станет доступен.

Настройку сетевой части для сопряженных с бриджами интерфейсов (`eth0`, `eth1`, `eth2`) уберите.

В данном случае создаем 3 интерфейса; один – для SSH (`br0` на базе `eth0`, он так же используется для доступа на хост), два других (`br1`, `br2`) – для сетевого взаимодействия.

2. Перегрузите сетевую часть:

```
root@debian:~# /etc/init.d/networking restart
```

Импорт в KVM

Чтобы установить виртуальную машину с инсталлированным «Программным комплексом С-Терра Шлюз. Версия 4.1» на KVM выполните следующие действия:

1. Скопируйте на хост виртуальную машину в формате `*.ova`.
2. OVA – это архив, из которого необходимо извлечь файл жесткого диска в формате `*.vmdk` (остальные файлы можно удалить):

```

root@debian:~# tar -xvf /tmp/ag41_cp39_kc1_4G_virtual_0.ova -C /tmp
sterragate-cp39.ovf
sterragate-cp39.mf
sterragate-cp39-disk1.vmdk

```

3. Vmdk и img форматы имеют ряд ограничений при использовании в KVM, поэтому рекомендуется использовать формат `qcow2`.

Необходимо переконвертировать файл формата vmdk в файл формата qcow2.

```
root@debian:~# qemu-img convert -f vmdk -O qcow2 /tmp/sterragate-cp39-disk1.vmdk /home/gate1.qcow2
```

4. Создайте виртуальную машину:

```
root@debian:~# virt-install --connect=qemu:///system -n gate -r 2048
--vcpus=2 --import --disk
path=/home/gate1.qcow2,format=qcow2,bus=virtio --vnc --noautoconsole
--os-type=linux --accelerate --network=bridge:br0,model=virtio --
network=bridge:br1,model=virtio --network=bridge:br2,model=virtio
```

```
Starting install...
```

```
Creating domain...
```

```
| 0 B 00:01
```

```
Domain creation completed. You can restart your domain by running:
```

```
virsh --connect qemu:///system start gate
```

Где:

- connect=qemu:///system – URL, по которому происходит подключение к KVM;
- n <имя> – название виртуальной машины;
- r <количество> – количество памяти в Мб, выделенных для виртуальной машины;
- vcpus=<количество> – количество процессорных ядер, выделенных для виртуальной машины; так же можно задать следующие опции: --vcpus 5, --vcpus 5,maxcpus=10, --vcpus sockets=2,cores=4,threads=2;
- import – использовать для виртуальной машины имеющийся образ диска;
- disk path=<путь до образа диска> – путь до диска (совместное использование с --import указывает на существующий диск); format=<формат образа>, bus=virtio – обязательные параметры для образов vmdk и qcow2;
- os-type=<ОС> – тип операционной системы виртуальной машины;
- vnc – запуск vnc для доступа к консоли, без этой команды через команду virsh console будет не подключиться;
- noautoconsole – не пытаться автоматически подсоединиться к консоли после создания виртуальной машины;
- accelerate – работа через /dev/kvm;
- network=bridge:<название интерфейса> – создание сетевого адаптера, с привязкой к конкретному бриджу; в данном случае br0 будет в виртуальной машине интерфейсом eth0, br1 – eth1, br2 – eth2; так же указывается тип адаптера virtio (model=virtio), иначе будет применяться стандартный тип адаптера, который имеет меньшую производительность.

5. Для доступа к консоли виртуальной машины выполните команду (выход из консоли осуществляется при помощи сочетания Ctrl +]):

```
root@debian:~# virsh console gate
```

```
Connected to domain gate
```

```
Escape character is ^]
```

```
Debian GNU/Linux 6.0 sterragate ttyS0
```

```
sterragate login:
```

Карта интерфейсов

При первом старте или при добавлении/удалении сетевых адаптеров после включения виртуального шлюза необходимо обновить карту интерфейсов.

1. Создайте карту интерфейсов:

```
root@sterragate:~# /bin/netifcfg enum > /home/map1
```

2. Отредактируйте созданный файл:

```
root@sterragate:~# vim.tiny /home/map1
```

```
0000:02:01.0 eth0 GigabitEthernet0/0
```

```
0000:02:03.0 eth2 GigabitEthernet0/2
```

```
0000:02:02.0 eth1 GigabitEthernet0/1
```

Где:

0000:02:01.0 – адрес устройства из lspci;

eth0 – наименование интерфейса в системе;

GigabitEthernet0/0 – наименование интерфейса в конфигурации S-Terra Gate.

3. Примените отредактированную карту интерфейсов:

```
root@sterragate:~# /bin/netifcfg map /home/map1
```

```
SUCCESS: Operation was successful.
```

4. Перегрузите шлюз.

Управление виртуальной машиной при помощи virsh

Команды virsh:

- `virsh edit <VM name>` – изменить настройки виртуальной машины;
- `virsh start <VM name>` – запустить виртуальную машину;
- `virsh shutdown <VM name>` – выключить виртуальную машину;
- `virsh reboot <VM name>` – перезагрузить виртуальную машину;
- `virsh console <VM name>` – открыть консоль виртуальной машины; выход из консоли осуществляется при помощи сочетания `Ctrl +]`;
- `virsh list --all` – вывести список всех виртуальных машин;
- `virsh destroy <VM name>` – уничтожает (останавливает) виртуальную машину (когда `shutdown` не работает);
- `virsh undefine <VM name>` – удалить виртуальную машину из списка (необходимо применять после `destroy`).

Монтирование USB Flash

В случае, если вам необходимо подключить USB Flash к виртуальному шлюзу (например, доставить на виртуальный шлюз сертификаты или политику безопасности) выполните следующие действия:

1. Подключите USB Flash к серверу, на котором развернут KVM.
2. Информация о подключенном устройстве отобразится в `dmesg`:

```
[ 5413.273010] usb 2-4: new high-speed USB device number 3 using ehci_hcd
[ 5413.405600] usb 2-4: New USB device found, idVendor=8644, idProduct=800e
[ 5413.405604] usb 2-4: New USB device strings: Mfr=1, Product=2,
SerialNumber=3
[ 5413.405607] usb 2-4: Product: USB Flash Disk
[ 5413.405610] usb 2-4: Manufacturer: General
[ 5413.405613] usb 2-4: SerialNumber: 12340000200027DB
[ 5413.406359] scsi5 : usb-storage 2-4:1.0
[ 5414.402739] scsi 5:0:0:0: Direct-Access      General  USB Flash Disk
1.0   PQ: 0 ANSI: 2
[ 5414.404574] sd 5:0:0:0: Attached scsi generic sg4 type 0
[ 5414.405336] sd 5:0:0:0: [sdd] 3915776 512-byte logical blocks: (2.00
GB/1.86 GiB)
[ 5414.406212] sd 5:0:0:0: [sdd] Write Protect is off
[ 5414.406216] sd 5:0:0:0: [sdd] Mode Sense: 03 00 00 00
[ 5414.407056] sd 5:0:0:0: [sdd] No Caching mode page found
[ 5414.407166] sd 5:0:0:0: [sdd] Assuming drive cache: write through
[ 5414.411285] sd 5:0:0:0: [sdd] No Caching mode page found
[ 5414.411394] sd 5:0:0:0: [sdd] Assuming drive cache: write through
[ 5414.412634]  sdd: sdd1
[ 5414.415965] sd 5:0:0:0: [sdd] No Caching mode page found
[ 5414.416073] sd 5:0:0:0: [sdd] Assuming drive cache: write through
[ 5414.416182] sd 5:0:0:0: [sdd] Attached SCSI removable disk
```

3. Подключите диск к виртуальной машине. Не рекомендуется подключать все устройство целиком (в данном случае – `dev/sdd`). Рекомендуется подключить только нужный раздел (в данном случае – `/dev/sdd1`).

```
root@debian:~# virsh attach-disk gate /dev/sdd1 vdb
Disk attached successfully
```

Где:

`gate` – название виртуальной машины;

`/dev/sdd1` – подключаемый к виртуальной машине раздел диска `sdd`;

`vdb` – наименование диска в виртуальной машине (по умолчанию в виртуальной машине только один используемый диск `vda`).

4. Перегрузите виртуальную машину. Диск будет подключен только при включении виртуальной машины.
5. Просмотрите все диски в системе:

```
root@sterragate:~# fdisk -l
```

```
...
```

```
Disk /dev/vdb: 2004 MB, 2004845056 bytes  
16 heads, 63 sectors/track, 3884 cylinders  
Units = cylinders of 1008 * 512 = 516096 bytes  
Sector size (logical/physical): 512 bytes / 512 bytes  
I/O size (minimum/optimal): 512 bytes / 512 bytes  
Disk identifier: 0x20ac7dda
```

```
This doesn't look like a partition table  
Probably you selected the wrong device.
```

Device	Boot	Start	End	Blocks	Id	System
/dev/vdb1	?	3198908	3628344	216435558+	7	HPFS/NTFS
Partition 1 does not end on cylinder boundary.						
/dev/vdb2	?	3246053	5184009	976730017	16	Hidden FAT16
Partition 2 does not end on cylinder boundary.						
/dev/vdb3	?	1	1	0	6f	Unknown
Partition 3 does not end on cylinder boundary.						
/dev/vdb4		49803	966802	462167897	0	Empty
Partition 4 does not end on cylinder boundary.						

```
Partition table entries are not in disk order
```

6. Создайте папку, куда будет подмонтирован диск:

```
mkdir /flash
```

7. Подмонтируйте диск:

```
mount /dev/vdb /flash
```

8. Устройство готово к использованию.