

# ПК Скаут

---

Развертывание в виртуальной среде

## Оглавление

---

1. Развертывание	4
1.1 KVM	4
1.1.1 Развертывание ПК Скаут в среде KVM в ручном режиме	4
1.1.2 Развертывание ПК Скаут в среде KVM с применением cloud-init	10

---

ООО "Гарда Технологии"

Все права сохраняются за правообладателем.

ООО "Гарда Технологии" оставляет за собой право вносить изменения в содержащуюся в данном документе информацию без предварительного уведомления.

#### ИНФОРМАЦИЯ О ПРАВЕ СОБСТВЕННОСТИ

Информация, содержащаяся в данном документе, является собственностью ООО "Гарда Технологии". Никакая часть этого документа не может быть воспроизведена или заимствована в какой бы то ни было форме или каким-либо способом - графическом, электронном виде или механическим путем, включая фотокопирование, запись, в том числе и на магнитные носители, или любые другие устройства, предназначенные для хранения информации - без письменного разрешения ООО "Гарда Технологии". Подобное разрешение не может быть выдано третьей стороной, будь то организация или частное лицо.

# 1. Развертывание

## 1.1 KVM

### 1.1.1 Развертывание ПК Скаут в среде KVM в ручном режиме

#### Общее описание

Для развертывания ПК Скаут в среде виртуальных машин KVM, необходимо использовать образ виртуального диска ПК Скаут в формате qcow2. В ручном режиме, ПК Скаут развертывается путем создания виртуальной машины, выполнения однократных действий по интеграции виртуальной машины в локальную сеть, а также базовых настроек ПК Скаут.

#### Создание и запуск ПК Скаут в среде KVM

##### СОЗДАНИЕ ВИРТУАЛЬНЫХ СЕТЕЙ

ПК Скаут взаимодействует с локальной сетью посредством трех сетевых интерфейсов:

- 1-й интерфейс - используется для управления, отправки уведомлений и административных задач;
- 2-й интерфейс - используется как входящий интерфейс для трафика защищаемой сети;
- 3-й интерфейс - используется как исходящий интерфейс для трафика защищаемой сети.

Порядок интерфейсов определяется по их расположению на шине PCI от меньшего к большему.

**Для каждого интерфейса необходимо создать виртуальную сеть. Поскольку Скаут представляет собой устройство типа L2-мост, входящий и исходящий интерфейсы для трафика должны также размещаться в отдельных виртуальных сетях.**

Для создания и активации виртуальной сети управления устройствами Скаут в среде KVM, необходимо использовать команды

```
virsh net-define --file net.xml
virsh net-start --network scout-management
virsh net-autostart --network scout-management
```

где в файле net.xml определена конфигурация виртуальной сети, например:

```
<network>
  <name>scout-management</name>
  <forward mode='nat'>
    <nat>
      <port start='1024' end='65535'>
    </nat>
  </forward>
  <bridge stp='off' delay='0'>
  <ip address='10.123.2.1' netmask='255.255.255.0'>
  </ip>
</network>
```

В приведенном примере, определяется сеть управления устройствами Скаут 10.123.2.0/24, использующая для доступа к внешним сетям трансляцию адресов.

Аналогичные команды используются для создания виртуальных сетей, через которые проходит фильтруемый трафик:

```
virsh net-define --file tc_in.xml
virsh net-start --network TC_IN
virsh net-autostart --network TC_IN
virsh net-define --file tc_out.xml
virsh net-start --network TC_OUT
virsh net-autostart --network TC_OUT
```

в этом случае целесообразно использовать изолированные виртуальные сети, например, для файла tc\_in.xml :

```
<network>
  <name>TC_IN</name>
```

```
<bridge stp='off' delay='0' />
</network>
```

и для файла `tc_out.xml` :

```
<network>
  <name>TC_OUT</name>
  <bridge stp='on' delay='0' />
</network>
```

Альтернативным способом создания виртуальных сетей, является графическая утилита `virt-manager` . Для создания новой виртуальной сети, необходимо:

- перейти в меню `Edit / Connection details` ;
- активировать вкладку `Virtual Networks` ;
- нажать на кнопку `+` и следовать подсказкам мастера добавления сети.

#### СОЗДАНИЕ ВИРТУАЛЬНОЙ МАШИНЫ

Для работы ПК Скаут необходимо создать виртуальную машину со следующими аппаратными возможностями:

- 6xvCPU model Skylake-Server-IBRS+
- 32Gb RAM
- Controller SCSI (model VirtIO SCSI)
- Virtual Disk VirtIO qcow2
- NIC VirtIO (management interface)
- NIC VirtIO (traffic in)
- NIC VirtIO (Traffic out)
- Serial 1 (text console on `/dev/ttyS0` )

Для создания виртуальной машины с именем `new-scout`, необходимо:

- 1) Скопировать исходный образ ПК Скаут в хранилище виртуальных дисков (по умолчанию `/var/lib/libvirt` с именем `new-scout.qcow2` )
- 2) Создать файл конфигурации виртуальной машины по шаблону (необходимо изменить имя виртуальной машины, названия виртуальных сетей и расположение и имя виртуального диска `qcow2`):

```
<domain type='kvm'>
  <name>new-scout</name>
  <metadata>
    <libosinfo:libosinfo xmlns:libosinfo="http://libosinfo.org/xmlns/libvirt/domain/1.0">
      <libosinfo:os id="http://debian.org/debian/10"/>
    </libosinfo:libosinfo>
  </metadata>
  <memory unit='GiB'>32</memory>
  <currentMemory unit='GiB'>32</currentMemory>
  <vcpu placement='static'>6</vcpu>
  <os>
    <type arch='x86_64' machine='pc-q35-3.1'>hvm</type>
  </os>
  <features>
    <acpi/>
    <apic/>
    <vmport state='off' />
  </features>
  <cpu mode='host-model' check='partial'>
    <model fallback='allow' />
  </cpu>
  <clock offset='utc'>
    <timer name='rtc' tickpolicy='catchup' />
    <timer name='pit' tickpolicy='delay' />
    <timer name='hpet' present='no' />
  </clock>
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>destroy</on_crash>
  <pm>
    <suspend-to-mem enabled='no' />
    <suspend-to-disk enabled='no' />
  </pm>
  <devices>
    <emulator>/usr/bin/qemu-system-x86_64</emulator>
    <disk type='file' device='disk'>
      <driver name='qemu' type='qcow2' />
```

```

<source file='/var/lib/libvirt/images/new-scout.qcow2'/>
<target dev='vda' bus='virtio'/>
<boot order='1'/>
<address type='pci' domain='0x0000' bus='0x07' slot='0x00' function='0x0'/>
</disk>
<controller type='usb' index='0' model='qemu-xhci' ports='15'>
  <address type='pci' domain='0x0000' bus='0x02' slot='0x00' function='0x0'/>
</controller>
<controller type='sata' index='0'>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x1f' function='0x2'/>
</controller>
<controller type='pci' index='0' model='pcie-root'/>
<controller type='pci' index='1' model='pcie-root-port'>
  <model name='pcie-root-port'/>
  <target chassis='1' port='0x10'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x0' multifunction='on'/>
</controller>
<controller type='pci' index='2' model='pcie-root-port'>
  <model name='pcie-root-port'/>
  <target chassis='2' port='0x11'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x1'/>
</controller>
<controller type='pci' index='3' model='pcie-root-port'>
  <model name='pcie-root-port'/>
  <target chassis='3' port='0x12'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x2'/>
</controller>
<controller type='pci' index='4' model='pcie-root-port'>
  <model name='pcie-root-port'/>
  <target chassis='4' port='0x13'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x3'/>
</controller>
<controller type='pci' index='5' model='pcie-root-port'>
  <model name='pcie-root-port'/>
  <target chassis='5' port='0x14'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x4'/>
</controller>
<controller type='pci' index='6' model='pcie-root-port'>
  <model name='pcie-root-port'/>
  <target chassis='6' port='0x15'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x5'/>
</controller>
<controller type='pci' index='7' model='pcie-root-port'>
  <model name='pcie-root-port'/>
  <target chassis='7' port='0x16'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x6'/>
</controller>
<controller type='pci' index='8' model='pcie-root-port'>
  <model name='pcie-root-port'/>
  <target chassis='8' port='0x17'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x7'/>
</controller>
<controller type='pci' index='9' model='pcie-to-pci-bridge'>
  <model name='pcie-pci-bridge'/>
  <address type='pci' domain='0x0000' bus='0x04' slot='0x00' function='0x0'/>
</controller>
<controller type='virtio-serial' index='0'>
  <address type='pci' domain='0x0000' bus='0x03' slot='0x00' function='0x0'/>
</controller>
<controller type='scsi' index='0' model='lsiilogic'>
  <address type='pci' domain='0x0000' bus='0x09' slot='0x01' function='0x0'/>
</controller>
<controller type='pci' index='10' model='pcie-root-port'>
  <model name='pcie-root-port'/>
  <target chassis='10' port='0x18'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
</controller>
<interface type='network'>
  <source network='scout-management'/>
  <model type='virtio'/>
  <address type='pci' domain='0x0000' bus='0x01' slot='0x00' function='0x0'/>
</interface>
<interface type='network'>
  <source network='TC_IN'/>
  <model type='virtio'/>
  <driver queues='3'/>
  <address type='pci' domain='0x0000' bus='0x08' slot='0x00' function='0x0'/>
</interface>
<interface type='network'>
  <source network='TC_OUT'/>
  <model type='virtio'/>
  <driver queues='3'/>
  <address type='pci' domain='0x0000' bus='0x0a' slot='0x00' function='0x0'/>
</interface>
<serial type='pty'>
  <target type='isa-serial' port='0'>
    <model name='isa-serial'/>
  </target>
</serial>
<console type='pty'>
  <target type='serial' port='0'>
</console>
<console type='pty'>
  <target type='virtio' port='1'/>

```

```

</console>
<channel type='unix'>
  <target type='virtio' name='org.qemu.guest_agent.0' />
  <address type='virtio-serial' controller='0' bus='0' port='1' />
</channel>
<channel type='spicevmc'>
  <target type='virtio' name='com.redhat.spice.0' />
  <address type='virtio-serial' controller='0' bus='0' port='2' />
</channel>
<input type='mouse' bus='ps2' />
<input type='keyboard' bus='ps2' />
<graphics type='spice' autoport='yes'>
  <listen type='address' />
  <image compression='off' />
</graphics>
<video>
  <model type='qxl' ram='65536' vram='65536' vgamem='16384' heads='1' primary='yes' />
  <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x0' />
</video>
<redirdev bus='usb' type='spicevmc'>
  <address type='usb' bus='0' port='1' />
</redirdev>
<redirdev bus='usb' type='spicevmc'>
  <address type='usb' bus='0' port='2' />
</redirdev>
<memballoon model='virtio'>
  <address type='pci' domain='0x0000' bus='0x05' slot='0x00' function='0x0' />
</memballoon>
<rng model='virtio'>
  <backend model='random'>/dev/urandom</backend>
  <address type='pci' domain='0x0000' bus='0x06' slot='0x00' function='0x0' />
</rng>
</devices>
</domain>

```

### 3) выполнить команду

```
virsh define new-scout.xml
```

### Запуск виртуальной машины

Для запуска созданной виртуальной машины, необходимо выполнить команду

```
virsh start new-scout
```

При необходимости, можно добавить ключ `--console` для подключения к локальной консоли запускаемой виртуальной машины.

Локальная консоль доступна через графический интерфейс утилиты `virt-manager` или через текстовый интерфейс утилиты `virsh`.

```
virsh console new-scout
```

Альтернативным способом создания виртуальных сетей, является графическая утилита `virt-manager`. Для создания новой виртуальной сети, необходимо:

- перейти в меню `File / New Virtual Machine`;
- выбрать вариант `Import existing disk image`;
- указать на расположение копии исходного образа ПК Скаут, который будет использоваться виртуальной машиной;
- указать Debian 10 в качестве операционной системы;
- указать параметры CPU и RAM;
- установить флаг `Customize configuration before install`;
- указать имя новой виртуальной машины;
- добавить 2-й и 3-й сетевой адаптеры в режиме `virtio`;
- нажать на кнопку `+` и следовать подсказкам мастера добавления сети.

### Первоначальный вход в систему через локальную консоль

ПК Скаут поставляется с установленным по умолчанию паролем пользователя `root`. Для входа в локальную консоль необходимо указать имя пользователя `root` и пароль, полученный от производителя ПК.

### ВАЖНО!!! Обязательно смените пароль пользователя root !!!

Для смены пароля необходимо использовать команду

```
passwd
```

после чего дважды ввести новый пароль.

### Конфигурирование IP адреса интерфейса управления

Образ виртуальной машины по умолчанию настроен на автоматическое получение IP адреса на 1-м сетевом интерфейсе по протоколу DHCP. При необходимости ручного конфигурирования сетевого интерфейса управления, необходимо:

1) подключиться к консоли виртуальной машины, используя предоставляемые системой виртуализации средства; 2) войти в консольный интерфейс, используя логин и пароль администратора; 3) внести изменения в файл `/etc/network/interfaces`:

```
# The primary network interface
allow-hotplug enp1s0
iface enp1s0 inet static
    address 1.2.3.4/31
    gateway 1.2.3.3
```

### Конфигурирования сервера DNS

Образ виртуальной машины по умолчанию настроен на автоматическое получение IP адреса сервера разрешения имен по протоколу DHCP. При необходимости ручного конфигурирования сервера DNS, необходимо:

1) подключиться к консоли виртуальной машины, используя предоставляемые системой виртуализации средства; 2) войти в консольный интерфейс, используя логин и пароль администратора; 3) внести изменения в файл `/etc/resolv.conf`:

```
nameserver 5.6.7.8
```

### Конфигурирование сервера NTP

ПК Скаут настроено на использование серверов времени debian (`0.debian.pool.ntp.org 1.debian.pool.ntp.org 2.debian.pool.ntp.org 3.debian.pool.ntp.org`).

Для просмотра состояния синхронизации введите команду:

```
timedatectl
```

Проверьте, что поле `System clock synchronized` имеет значение `yes`.

При необходимости ручного конфигурирования серверов имен, внесите изменения в файл `/etc/systemd/timesyncd.conf`, например, если ваш сервер времени имеет адрес `1.2.3.4`, добавьте строку:

```
NTP=1.2.3.4
```

и перезапустите сервис `systemd-timesyncd`.

### Смена паролей администратора ПК Скаут


Доступ к веб-интерфейсу ПК Скаут осуществляется по протоколу HTTPS при предъявлении логина и пароля. Для входа в веб-интерфейс перейдите в браузере на `https://{IP_АДРЕС_ИНТЕРФЕЙСА_УПРАВЛЕНИЯ}`.

Используйте `admin` для логина и пароля в качестве учетных данных по умолчанию.

### ВНИМАНИЕ!!! Обязательно измените пароль по умолчанию после первого входа в систему!!!



Для смены пароля пользователя `admin`, необходимо:

- войти в веб-интерфейс с учетными данным по умолчанию;
- перейти в верхнее меню `Администрирование` ;
- выбрать в левой группе `Настройки` пункт `Пользователи` или прокрутить экран вниз до блока `Пользователи` ;
- нажать на пиктограмму  в строке пользователя `admin` ;
- в открывшейся форме ввода, ввести новый пароль в поля `Пароль` и `Подтверждение пароля` ;
- нажать на кнопку `Сохранить` .

### Проверка работоспособности

ЧЕРЕЗ ВЕБ-ИНТЕРФЕЙС ПК СКАУТ

Для проверки нормального функционирования ПК Скаут, необходимо:

- войти в веб-интерфейс с учетными данным администратора;
- перейти в меню `Администрирование` ;
- выбрать в левой группе `Настройки` пункт `Журналы` или прокрутить экран вниз до блока `Журналы` ;
- нажать на кнопку `Журнал событий` ;
- убедиться в отсутствии ошибок.

ЧЕРЕЗ ЛОКАЛЬНУЮ КОНСОЛЬ

Для проверки работоспособности компонентов ПК Скаут, необходимо:

- выполнить вход в локальную консоль с правами администратора ПК Скаут;
- выполнить команду `systemctl is-active scout-* redis-server@*` . Вывод должен содержать только строчки `active` .

## 1.1.2 Развертывание ПК Скаут в среде KVM с применением cloud-init

### Общее описание

Для развертывания ПК Скаут в среде виртуальных машин KVM, необходимо использовать образ виртуального диска ПК Скаут в формате qcow2. Образ содержит пакет cloud-init, который позволяет настроить ПК Скаут при первом запуске создаваемой виртуальной машины. Поддерживаются следующие операции:

- присвоение имени хоста (hostname);
- настройка параметров сетевого интерфейса управления;
- настройка сервера разрешения имен DNS;
- настройка сервера точного времени NTP;
- создание пользователей и групп в операционной системе;

### Подготовка метаданных cloud-init для экземпляра виртуальной машины

СОЗДАНИЕ ФАЙЛОВ КОНФИГУРАЦИИ ЭКЗЕМПЛЯРА ВИРТУАЛЬНОЙ МАШИНЫ.

Метаданные cloud-init состоит из двух файлов:

- network-config.cfg - файл содержит параметры сетевого интерфейса управления

```
version: 1
config:
- type: physical
  name: enp1s0
  subnets:
  - type: static
    control: hotplug
    address: 10.123.2.71/24
    gateway: 10.123.2.1
    dns_nameservers:
    - 10.123.2.1
```

Примечание. Название сетевого интерфейса зависит от конфигурации виртуальной машины KVM. Если в состав входит PCIe контроллер, то название имеет вид enpXsY, где X и Y - номера устройства и функции PCIe соответственно. Если в составе виртуальной машины нет контроллер PCIe, то сетевой интерфейс располагается на шине PCI, а название имеет вид ensX, где X - номер устройства на шине

Полная информация о формате файла и правилах его использования доступна в документации cloud-init (<https://cloudinit.readthedocs.io/en/latest/topics/network-config-format-v1.html>)

- cloud-config.cfg - файл содержит параметры настройки экземпляра виртуальной машины

```
# настройка пользователей операционной системы
chpasswd:
  list: |
    root:<PASSWORD_HASH>
  disable_root: false

# настройка сервера времени
ntp:
  pools: []
  servers: [ '10.123.2.1' ]

# дополнительные настройки
resize_rootfs: false
ssh:
  emit_keys_to_console: false

#

final_message: "Scout is finally up after $SUPTIME seconds"
```

где: - <PASSWORD HASH> - хэш пароля, получаемый командой:

```
mkpasswd -m sha-512 "my password"
```

Полная информация о формате файла и правилах его использования доступна в документации cloud-init (<https://cloudinit.readthedocs.io/en/latest/topics/examples.html>)

#### ЗАГРУЗКА ПРЕДВАРИТЕЛЬНО НАСТРОЕННОЙ КОНФИГУРАЦИИ ПК СКАУТ

Предварительно настроенная конфигурация готовится на эталонном комплексе ПК Скаут. Для её создания, необходимо:

- выполнить развертывание ПК Скаут с настройками по умолчанию;
- внести необходимые изменения в конфигурацию ПК Скаут;
- экспортировать конфигурацию в файл через меню `Администрирование / Обслуживание / Резервное копирование веб-интерфейса`;
- преобразовать содержимое файла в кодировку base64.

В файл `cloud-config.cfg` необходимо добавить:

```
write_files:
- encoding: base64
  content: <CONFIG ARCHIVE IN BASE64>
  path: /run/cloud_import_config.gz
```

где,

- `<CONFIG ARCHIVE IN BASE64>` - конфигурация в кодировке base64.

#### ПОДГОТОВКА ОБРАЗА, СОДЕРЖАЩЕГО МЕТАДААННЫЕ CLOUD-INIT

Для подключения метаданных к виртуальной машине в среде KVM (тип облака NoCloud), используется seed образ, подключаемый к виртуальной машине как дополнительный CD. Для создания образа необходимо выполнить команду:

```
cloud-localds -v --network-config=network-config.cfg seed.img cloud-config.cfg
```

где:

- `network-config.cfg` - полный путь к файлу с настройками сетевых интерфейсов;
- `cloud-config.cfg` - полный путь к файлу с настройками экземпляра виртуальной машины;
- `seed.img` - полный путь к файлу образа, в который будут записаны метаданные cloud-init

### Создание и запуск ПК Скаут в среде KVM

#### СОЗДАНИЕ ВИРТУАЛЬНЫХ СЕТЕЙ

ПК Скаут взаимодействует с локальной сетью посредством трех сетевых интерфейсов:

- 1-й интерфейс - используется для управления, отправки уведомлений и административных задач;
- 2-й интерфейс - используется как входящий интерфейс для трафика защищаемой сети;
- 3-й интерфейс - используется как исходящий интерфейс для трафика защищаемой сети.

Порядок интерфейсов определяется по их расположению на шине PCI от меньшего к большему.

**Для каждого интерфейса необходимо создать виртуальную сеть. Поскольку Скаут представляет собой устройство типа L2-мост, входящий и исходящий интерфейсы для трафика должны также размещаться в отдельных виртуальных сетях.**

Для создания и активации виртуальной сети управления устройствами Скаут в среде KVM, необходимо использовать команды

```
virsh net-define --file net.xml
virsh net-start --network scout-management
virsh net-autostart --network scout-management
```

где в файле `net.xml` определена конфигурация виртуальной сети, например:

```
<network>
  <name>scout-management</name>
  <forward mode='nat'>
```

```
<nat>
  <port start='1024' end='65535' />
</nat>
</forward>
<bridge stp='off' delay='0' />
<ip address='10.123.2.1' netmask='255.255.255.0'>
</ip>
</network>
```

В приведенном примере, определяется сеть управления устройствами Скаут 10.123.2.0/24, использующая для доступа к внешним сетям трансляцию адресов.

Аналогичные команды используются для создания виртуальных сетей, через которые проходит фильтруемый трафик:

```
virsh net-define --file tc_in.xml
virsh net-start --network TC_IN
virsh net-autostart --network TC_IN
virsh net-define --file tc_out.xml
virsh net-start --network TC_OUT
virsh net-autostart --network TC_OUT
```

в этом случае целесообразно использовать изолированные виртуальные сети, например, для файла tc\_in.xml:

```
<network>
  <name>TC_IN</name>
  <bridge stp='off' delay='0' />
</network>
```

и для файла tc\_out.xml

```
<network>
  <name>TC_OUT</name>
  <bridge stp='on' delay='0' />
</network>
```

Альтернативным способом создания виртуальных сетей, является графическая утилита virt-manager. Для создания новой виртуальной сети, необходимо:

- перейти в меню Edit / Connection details;
- активировать вкладку Virtual Networks;
- нажать на кнопку + и следовать подсказкам мастера добавления сети.

#### СОЗДАНИЕ ВИРТУАЛЬНОЙ МАШИНЫ

Для работы ПК Скаут необходимо создать виртуальную машину со следующими аппаратными возможностями:

- 6xvCPU model Skylake-Server-IBRS+
- 32Gb RAM
- Controller SCSI (model VirtIO SCSI)
- Virtual Disk VirtIO qcow2
- Virtual Disk cdrom raw
- NIC VirtIO (management interface)
- NIC VirtIO (traffic in)
- NIC VirtIO (Traffic out)
- Serial 1 (text console on /dev/ttyS0)

Для создания виртуальной машины с именем new-scout, необходимо:

1) Скопировать исходный образ ПК Скаут в хранилище виртуальных дисков (по умолчанию /var/lib/libvirt с именем new-scout.qcow2)

2) Создать файл конфигурации виртуальной машины по шаблону (необходимо изменить имя виртуальной машины, названия виртуальных сетей, а также расположение и имя виртуального диска qcow2):

```
<domain type='kvm'>
  <name>new-scout</name>
```

```

<metadata>
<libosinfo:libosinfo xmlns:libosinfo="http://libosinfo.org/xmlns/libvirt/domain/1.0">
  <libosinfo:os id="http://debian.org/debian/10"/>
</libosinfo:libosinfo>
</metadata>
<memory unit='GiB'>32</memory>
<currentMemory unit='GiB'>32</currentMemory>
<vcpu placement='static'>6</vcpu>
<os>
  <type arch='x86_64' machine='pc-q35-3.1'>hvm</type>
</os>
<features>
  <acpi/>
  <apic/>
  <vmport state='off'/>
</features>
<cpu mode='host-model' check='partial'>
  <model fallback='allow'/>
</cpu>
<clock offset='utc'>
  <timer name='rtc' tickpolicy='catchup'/>
  <timer name='pit' tickpolicy='delay'/>
  <timer name='hpet' present='no'/>
</clock>
<on_poweroff>destroy</on_poweroff>
<on_reboot>restart</on_reboot>
<on_crash>destroy</on_crash>
<pm>
  <suspend-to-mem enabled='no'/>
  <suspend-to-disk enabled='no'/>
</pm>
<devices>
<emulator>/usr/bin/qemu-system-x86_64</emulator>
<disk type='file' device='cdrom'>
  <driver name='qemu' type='raw'/>
  <source file='/var/lib/libvirt/images/seed.img'/>
  <target dev='sda' bus='sata'/>
  <readonly/>
  <address type='drive' controller='0' bus='0' target='0' unit='0'/>
</disk>
<disk type='file' device='disk'>
  <driver name='qemu' type='qcow2'/>
  <source file='/var/lib/libvirt/images/new-scout.qcow2'/>
  <target dev='vda' bus='virtio'/>
  <boot order='1'/>
  <address type='pci' domain='0x0000' bus='0x07' slot='0x00' function='0x0'/>
</disk>
<controller type='usb' index='0' model='qemu-xhci' ports='15'>
  <address type='pci' domain='0x0000' bus='0x02' slot='0x00' function='0x0'/>
</controller>
<controller type='sata' index='0'>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x1f' function='0x2'/>
</controller>
<controller type='pci' index='0' model='pcie-root'/>
<controller type='pci' index='1' model='pcie-root-port'>
  <model name='pcie-root-port'/>
  <target chassis='1' port='0x10'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x0' multifunction='on'/>
</controller>
<controller type='pci' index='2' model='pcie-root-port'>
  <model name='pcie-root-port'/>
  <target chassis='2' port='0x11'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x1'/>
</controller>
<controller type='pci' index='3' model='pcie-root-port'>
  <model name='pcie-root-port'/>
  <target chassis='3' port='0x12'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x2'/>
</controller>
<controller type='pci' index='4' model='pcie-root-port'>
  <model name='pcie-root-port'/>
  <target chassis='4' port='0x13'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x3'/>
</controller>
<controller type='pci' index='5' model='pcie-root-port'>
  <model name='pcie-root-port'/>
  <target chassis='5' port='0x14'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x4'/>
</controller>
<controller type='pci' index='6' model='pcie-root-port'>
  <model name='pcie-root-port'/>
  <target chassis='6' port='0x15'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x5'/>
</controller>
<controller type='pci' index='7' model='pcie-root-port'>
  <model name='pcie-root-port'/>
  <target chassis='7' port='0x16'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x6'/>
</controller>
<controller type='pci' index='8' model='pcie-root-port'>
  <model name='pcie-root-port'/>
  <target chassis='8' port='0x17'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x7'/>
</controller>

```

```

<controller type='pci' index='9' model='pcie-to-pci-bridge'>
  <model name='pcie-pci-bridge'>
    <address type='pci' domain='0x0000' bus='0x04' slot='0x00' function='0x0'>
  </controller>
<controller type='virtio-serial' index='0'>
  <address type='pci' domain='0x0000' bus='0x03' slot='0x00' function='0x0'>
</controller>
<controller type='scsi' index='0' model='lsiilogic'>
  <address type='pci' domain='0x0000' bus='0x01' slot='0x01' function='0x0'>
</controller>
<controller type='pci' index='10' model='pcie-root-port'>
  <model name='pcie-root-port'>
    <target chassis='10' port='0x18'>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'>
</controller>
<interface type='network'>
  <source network='scout-management'>
  <model type='virtio'>
  <address type='pci' domain='0x0000' bus='0x01' slot='0x00' function='0x0'>
</interface>
<interface type='network'>
  <source network='TC_IN'>
  <model type='virtio'>
  <driver queues='3'>
  <address type='pci' domain='0x0000' bus='0x08' slot='0x00' function='0x0'>
</interface>
<interface type='network'>
  <source network='TC_OUT'>
  <model type='virtio'>
  <driver queues='3'>
  <address type='pci' domain='0x0000' bus='0x0a' slot='0x00' function='0x0'>
</interface>
<serial type='pty'>
  <target type='isa-serial' port='0'>
  <model name='isa-serial'>
  </target>
</serial>
<console type='pty'>
  <target type='serial' port='0'>
</console>
<console type='pty'>
  <target type='virtio' port='1'>
</console>
<channel type='unix'>
  <target type='virtio' name='org.qemu.guest_agent.0'>
  <address type='virtio-serial' controller='0' bus='0' port='1'>
</channel>
<channel type='spicevmc'>
  <target type='virtio' name='com.redhat.spice.0'>
  <address type='virtio-serial' controller='0' bus='0' port='2'>
</channel>
<input type='mouse' bus='ps2'>
<input type='keyboard' bus='ps2'>
<graphics type='spice' autoport='yes'>
  <listen type='address'>
  <image compression='off'>
</graphics>
<video>
  <model type='qxl' ram='65536' vram='65536' vgamem='16384' heads='1' primary='yes'>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x0'>
</video>
<redirdev bus='usb' type='spicevmc'>
  <address type='usb' bus='0' port='1'>
</redirdev>
<redirdev bus='usb' type='spicevmc'>
  <address type='usb' bus='0' port='2'>
</redirdev>
<memballoon model='virtio'>
  <address type='pci' domain='0x0000' bus='0x05' slot='0x00' function='0x0'>
</memballoon>
<rng model='virtio'>
  <backend model='random'>/dev/urandom</backend>
  <address type='pci' domain='0x0000' bus='0x06' slot='0x00' function='0x0'>
  </rng>
</devices>
</domain>

```

### 3) выполнить команду

```
virsh define new-scout.xml
```

### Запуск виртуальной машины

Для запуска созданной виртуальной машины, необходимо выполнить команду

```
virsh start new-scout
```

При необходимости, можно добавить ключ `--console` для подключения к локальной консоли запускаемой виртуальной машины.

Локальная консоль доступна через графический интерфейс утилиты `virt-manager` или через текстовый интерфейс утилиты `virsh`

```
virsh console new-scout
```

### Проверка работоспособности

При использовании `cloud-init`, запускаемая виртуальная машина будет иметь задаваемые в конфигурационных файлах настройки.

#### ЧЕРЕЗ ВЕБ-ИНТЕРФЕЙС ПК СКАУТ

Для проверки нормального функционирования ПК Скаут, необходимо:

- войти в веб-интерфейс с учетными данными администратора (если использовался импорт конфигурации, то учетные данные для входа будут соответствовать импортированной конфигурации);
- перейти в меню `Администрирование` ;
- выбрать в левой группе `Настройки` пункт `Журналы` или прокрутить экран вниз до блока `Журналы` ;
- нажать на кнопку `Журнал событий` ;
- убедиться в отсутствии ошибок.

#### ЧЕРЕЗ ЛОКАЛЬНУЮ КОНСОЛЬ

Для проверки работоспособности компонентов ПК Скаут, необходимо:

- выполнить вход в локальную консоль с правами администратора ПК Скаут (если использовались модули управления пользователями операционной системы, то учетные данные для входа будут соответствовать заданным в конфигурационных файлах);
- выполнить команду `systemctl is-active scout-* redis-server@*` Вывод должен содержать только строчки `active`.