

Использование VBoxManage

Использование VBoxManage

VBoxManage list vms - список имеющихся машин

VBoxManage startvm - команда запуска машины

VBoxManage list runningvms - список запущенных машин

Управление VirtualBox из консоли с помощью vboxmanage

Просмотреть настройки VirtualBox:

Изменение настроек, например, пути до каталога, где хранятся виртуалки:

Список поддерживаемых типов виртуалок:

Создаем новую виртуалку:

Меняем параметры VM — указываем количество памяти и тд:

Создаем жесткий диск размером 10 Гб (место по дефолту не преаллоцируется):

Цепляем созданный жесткий диск к виртуалке:

Цепляем к VM ISO-образ, с которого будет происходить установка системы:

Включаем доступ к VM по VNC:

Я лично для подключения использую TigerVNC (форк TightVNC):

Чтобы каждый раз не вводить пароль, можно сказать:

Посмотреть свойства виртуальной машины (узнать номер порта VNC):

Запустить виртуалку...

Запуск на десктопе

Запуск на сервере и будем подключаться по VNC

Для некоторых систем установочный диск отключается от VM автоматически по окончанию установки. Если этого не произошло, говорим:

Pause, Resume, Poweroff:

Проброс портов настраивается так (VM должна быть остановлена):

После этого в виртуалку можно будет зайти по SSH таким образом:

Просмотр списка правил перенаправления портов:

Удаление правила:

Список всех виртуалок:

Список работающих виртуалок:

Переименовать VM (при этом переименоывается и ее каталог):

Создать полный клон VM:

Список NAT-сетей:

Создание новой сети:

Редактирование сети:

Удаление сети:

Подключение виртуалок к NAT-сети:

Удалить виртуалку и все ассоциированные с ней файлы, в том числе и диски:

Экспорт:

Импорт:

Или, с переопределением имени:

Установить гостевую ОС

VBoxManage list vms - список имеющихся машин

команда

```
1 | VBoxManage list vms
```

результат

```
1 | "Windows 10" {3d355328-a9ff-4514-acf2-5e35f321b328}
2 | "Ubuntu 18.04 server" {3ad61441-1a24-4809-b1b9-cf62fd6bc3e3}
3 | "<inaccessible>" {6b79ef35-7335-437f-b8d4-da9846c77ac6}
4 | "<inaccessible>" {9258d297-c143-4426-ac52-539e5a374998}
5 | "<inaccessible>" {fcff38f7-f4a3-4eb6-80de-26571a6de7d8}
6 | "Ubuntu-18.04.3-live-server" {07d85140-5d86-4d4d-8964-d30f066d33cb}
```

VBoxManage startvm - команда запуска машины

команда

```
1 | VBoxManage startvm 07d85140-5d86-4d4d-8964-d30f066d33cb --type headless
```

результат

```
1 | Waiting for VM "07d85140-5d86-4d4d-8964-d30f066d33cb" to power on...
2 | VM "07d85140-5d86-4d4d-8964-d30f066d33cb" has been successfully started.
```

VBoxManage list runningvms - список запущенных машин

команда

```
1 | VBoxManage list runningvms
```

ВЫВОД

```
1 | "Ubuntu-18.04.3-live-server" {07d85140-5d86-4d4d-8964-d30f066d33cb}
```

Управление VirtualBox из консоли с помощью vboxmanage

Такие решения, как LXC и KVM, не всегда удобны, потому что они работают только под Linux. Используя их, вы не можете передать виртуалки пользователям каких-нибудь MacOS или Windows. По этой причине, а также потому что на практике у еще не возникало необходимости запускать больше пяти VM одновременно, и все еще предпочитаю VirtualBox. Им можно управлять из консоли при помощи Vagrant, но Vagrant всегда делал чуть-чуть не то, что на самом деле было нужно. Например, чтобы по умолчанию все виртуалки всегда были в одной NAT-сети, без какой-либо правки Vagrantfile'ов. Все это сподвигло к изучению «родных» утилит VirtualBox, в частности, **vboxmanage**.

Давайте же посмотрим, как все, что вы когда-либо можете захотеть от виртуалок, делается с помощью **vboxmanage**.

Просмотреть настройки VirtualBox:

```
1 | vboxmanage list systemproperties
```

Изменение настроек, например, пути до каталога, где хранятся виртуалки:

```
1 | vboxmanage setproperty machinefolder /home/eax/virtualbox
```

Список поддерживаемых типов виртуалок:

```
1 | vboxmanage list ostypes
2 | vboxmanage list ostypes | egrep ^ID
```

Создаем новую виртуалку:

```
1 | vboxmanage createvm --name ubuntu1604 --ostype Ubuntu_64 --register
```

Меняем параметры VM — указываем количество памяти и тд:

```
1 | vboxmanage modifyvm ubuntu1604 --cpus 1 --memory 512 --audio none \
2 |   --usb off --acpi on --boot1 dvd --nic1 nat
```

Создаем жесткий диск размером 10 Гб (место по дэфолту не преаллоцируется):

```
1 | vboxmanage createhd \
2 |   --filename /home/eax/virtualbox/ubuntu1604/ubuntu1604.vdi \
3 |   --size 10000
```

Цепляем созданный жесткий диск к виртуалке:

```
1 | vboxmanage storagectl ubuntu1604 --name ide-controller --add ide
2 | vboxmanage storageattach ubuntu1604 --storagectl ide-controller \
3 |   --port 0 --device 0 --type hdd \
4 |   --medium /home/eax/virtualbox/ubuntu1604/ubuntu1604.vdi
```

Цепляем к VM ISO-образ, с которого будет происходить установка системы:

```
1 | vboxmanage storageattach ubuntu1604 --storagectl ide-controller \
2 |   --port 0 --device 1 --type dvddrive \
3 |   --medium /home/eax/data/iso/ubuntu-16.04.1-server-amd64.iso
```

Включаем доступ к VM по VNC:

```
1 | vboxmanage modifyvm ubuntu1604 --vrde on
2 | vboxmanage modifyvm ubuntu1604 --vrdeaddress 127.0.0.1
3 | vboxmanage modifyvm ubuntu1604 --vrdeport 3001
4 | vboxmanage modifyvm ubuntu1604 --vrdeproperty VNCPassword="secret"
```

Я лично для подключения использую TigerVNC (форк TightVNC):

```
1 | vncviewer localhost:3001
```

Чтобы каждый раз не вводить пароль, можно сказать:

```
1 | vncpasswd ./vnc.password
```

... и всегда использовать этот файл:

```
vncviewer localhost:3001 PasswordFile=/home/eax/temp/vnc.password
```

Посмотреть свойства виртуальной машины (узнать номер порта VNC):

```
1 | vboxmanage showvminfo ubuntu1604 | grep VRDE
```

Запустить виртуалку...

Запуск на десктопе

```
1 | vboxmanage startvm ubuntu1604
```

Запуск на сервере и будем подключаться по VNC

```
1 | vboxmanage startvm --type headless ubuntu1604
```

Для некоторых систем установочный диск отключается от ВМ автоматически по окончании установки. Если этого не произошло, говорим:

```
1 | vboxmanage storageattach archlinux1 --storagectl ide-controller \  
2 | --port 0 --device 1 --type dvddrive --medium emptydrive
```

Pause, Resume, Poweroff:

```
1 | vboxmanage controlvm archlinux1 pause  
2 | vboxmanage controlvm archlinux1 resume  
3 | vboxmanage controlvm archlinux1 poweroff
```

Проброс портов настраивается так (ВМ должна быть остановлена):

```
1 | vboxmanage modifyvm ubuntu1604 --natpf1 "ssh-forwarding,tcp,,22001,,22"
```

После этого в виртуалку можно будет зайти по SSH таким образом:

```
1 | ssh -p 22001 username@localhost
```

Просмотр списка правил перенаправления портов:

```
1 | vboxmanage showvminfo ubuntu1604 | grep Rule
```

Удаление правила:

```
1 | vboxmanage modifyvm ubuntu1604 --natpf1 delete ssh-forwarding
```

Список всех виртуалок:

```
1 | vboxmanage list vms
2 | vboxmanage list vms --long | egrep '^(Name|State)'
```

Список работающих виртуалок:

```
1 | vboxmanage list runningvms
```

Переименовать ВМ (при этом переименоывается и ее каталог):

```
1 | vboxmanage modifyvm ubuntu1604 --name ubuntu1
```

Создать полный клон ВМ:

```
1 | vboxmanage clonevm ubuntu1 --name ubuntu2 --register
```

Список NAT-сетей:

```
1 | vboxmanage natnetwork list
```

Создание новой сети:

```
1 | vboxmanage natnetwork add --netname UbuntuNat --network 10.0.4.0/24 \
2 | --enable --dhcp on --ipv6 off
```

Редактирование сети:

```
1 | vboxmanage natnetwork modify --netname UbuntuNat --ipv6 on
```

Удаление сети:

```
1 | vboxmanage natnetwork remove --netname UbuntuNat
```

Подключение виртуалок к NAT-сети:

```
1 | vboxmanage modifyvm ubuntu1 --nic2 natnetwork --nat-network2 UbuntuNat
2 | vboxmanage modifyvm ubuntu2 --nic2 natnetwork --nat-network2 UbuntuNat
```

Важно! Скорее всего, гостевая операционная система окажется не настроена для использования второго интерфейса, если вы добавили его после установки. В Ubuntu для добавления нового интерфейса нужно отредактировать файл `/etc/network/interfaces`.

Удалить виртуалку и все ассоциированные с ней файлы, в том числе и диски:

```
1 | vboxmanage unregistervm ubuntu2 --delete
```

Экспорт:

```
1 | vboxmanage export ubuntu2 --output ubuntu2.ova
```

Импорт:

```
1 | vboxmanage import ubuntu2.ova
```

Или, с переопределением имени:

```
1 | vboxmanage import ubuntu2.ova --vsys 0 --vmname ubuntu2
```

Это, разумеется, далеко не полный список возможностей **vboxmanage**. Например, мы не рассмотрели использование снапшотов, соединение машин при помощи последовательного порта (например, для отладки ядра операционной системы), работу с разделяемыми каталогами, и многие другие.

Подробности вы найдете в официальной документации и `vboxmanage --help`.

Установить гостевую ОС

```
1 | apt-get install dkms build-essential linux-headers-generic
```

Подключить образ диска в дисковод гостевой ОС: Устройства — Подключить образ диска дополнений гостевой ОС. Этот образ входит в поставку основного дистрибутива — скачивать дополнительно его не нужно.

Смонтировать устройство в файловую систему, если это не произошло автоматически.

```
1 | mkdir /media/cdrom  
2 | mount /dev/sr0 /media/cdrom
```

Запустить скрипт-установщик.

```
1 | cd /media/cdrom  
2 | ./VBoxLinuxAdditions.run
```