

Запуск команд внутри гостевых ОС в гипервизоре KVM на примере Proxmox

 ittraveler.org/zapusk-komand-vnutri-gostevyx-os-v-gipervizore-kvm-na-primere-proxmox

18 марта 2019 г.

Дата: 18.03.2019 Автор Admin

В VMware, с помощью Powercli, есть возможность запускать команды внутри гостевых ОС с помощью командлета Invoke-VMScript, это очень удобно, ведь с помощью этого механизма можно выполнить необходимые команды на сотне VM, не открывая на них консоль.

Работая с KVM мне захотелось найти аналог данного механизма, чтобы запускать команды из консоли гипервизора прямо на гостевых ОС, по аналогии с VMware.

В данной статье мы рассмотрим как использовать qemu агент для выполнения задуманного.

Для начала установите qemu агент, взять его можно по [ссылке](#)

Для установки в Linux используйте следующие команды:

```
1 apt-get install qemu-guest-agent
```

```
1 yum install qemu-guest-agent
```

После установки агента внутри виртуальной машины, нужно активировать его поддержку в настройках VM, вот так активированный агент для VM выглядит в Proxmox:

Summary	Edit	Revert
> Console	Name	sys
Hardware	Start at boot	No
Cloud-Init	Start/Shutdown order	order=any
Options	OS Type	Microsoft Windows 10/2016
Task History	Boot Order	Disk 'virtio0', CD-ROM, Network
Monitor	Use tablet for pointer	Yes
Backup	Hotplug	Disk, Network, USB
Replication	ACPI support	Yes
Snapshots	SCSI Controller	VirtIO SCSI
Firewall	BIOS	Default (SeaBIOS)
Permissions	KVM hardware virtualization	Yes
	Freeze CPU at startup	No
	Use local time for RTC	No
	RTC start date	now
	SMBIOS settings (type1)	uuid=ed4e291d-4644-4859-a729-37b8f0dc9072
	Qemu Agent	Enabled
	Protection	No

Если вы используете libvirt, то xml код будет примерно такой:

```

1 <channel type='unix'>
2   <source mode='bind' path='/var/lib/libvirt/qemu/f16x86_64.agent'/>
3   <target type='virtio' name='org.qemu.guest_agent.0'/>
4 </channel>

```

Теперь включите VM, с помощью команды `qm ping`, можно убедиться что агент внутри VM функционирует нормально.

Если команда

```
1 qm ping 101
```

где 101 — это ID вашей VM, не вернула ничего — значит агент функционирует нормально.

Теперь рассмотрим как запускать команды внутри VM с windows из консоли гипервизора KVM (он же в данном случае Proxmox)

Чтобы подключиться к сессии VM, выполните команду, где 101 — ID вашей VM

```
1 socat /var/run/qemu-server/101.qga -
```

Теперь нам нужно передать команду внутрь гостевой ОС, команды передаются в формате JSON.

Для примера отправим команду на перезапуск гостевой ОС

```
1 {"execute":"guest-exec", "arguments":{"path":"cmd.exe","arg":["/c","shutdown", "-r", "-f"]}}
```

Данная команда перезагрузит вашу VM.

По аналогии можно перезагрузить VM без использования cmd:

```
1 {"execute":"guest-exec", "arguments":{"path":"shutdown.exe","arg":["-r", "-f"]}}
```

или например та же перезагрузка, только с помощью powershell:

```
1 {"execute":"guest-exec", "arguments":{"path":"powershell.exe","arg":["-command","restart-computer", "-force"]}}
```

Вот так к примеру можно запустить sysprep:

```
1 {"execute":"guest-exec", "arguments":{"path":"c:\\windows\\system32\\sysprep\\sysprep.exe","arg":["/oobe", "/generalize", "/reboot"]}}
```

Если вы не хотите подключаться к сессии агента, а просто отправить команду одной строкой — используйте echo и отправляйте команду в socat:

```
1 echo '{"execute":"guest-exec", "arguments":{"path":"c:\\windows\\system32\\sysprep\\sysprep.exe","arg":["/oobe", "/generalize", "/reboot"]}}' | socat /var/run/qemu-server/101.qga -
```

Обратите внимание что слеш и спец символы в JSON экранируются, поэтому, перед тем как отправлять команду гостю — проверьте что синтаксис корректный.

Если не уверены, то всегда можно воспользоваться командой ConvertTo-Json в Powershell, например:

```
1 'cmd.exe "c:\\windows\\system32\\sysprep.exe /oobe"' | ConvertTo-Json
```

на выходе получите строку, с экранированным выводом:

```
1 "cmd.exe \"c:\\windows\\system32\\sysprep.exe /oobe\""
```

На этом все.

В следующей статье рассмотрим как можно автоматизировать процесс деплоя и ввода в домен AD, VM на базе KVM.

Удачной настройки!