


Развёртывание и настройка oVirt 4.0. Часть 5.

Watchdog как средство повышения доступности гостевых систем виртуальных машин

 blog.it-kb.ru/2016/09/16/install-ovirt-4-0-part-5-how-to-setup-virtual-watchdog-device-and-guest-os-ubuntu-server-16-04-service-watchdog-linux-daemon-for-high-availability-and-restarting-unresponsive-vm

Автор:Алексей Максимов

16.09.2016

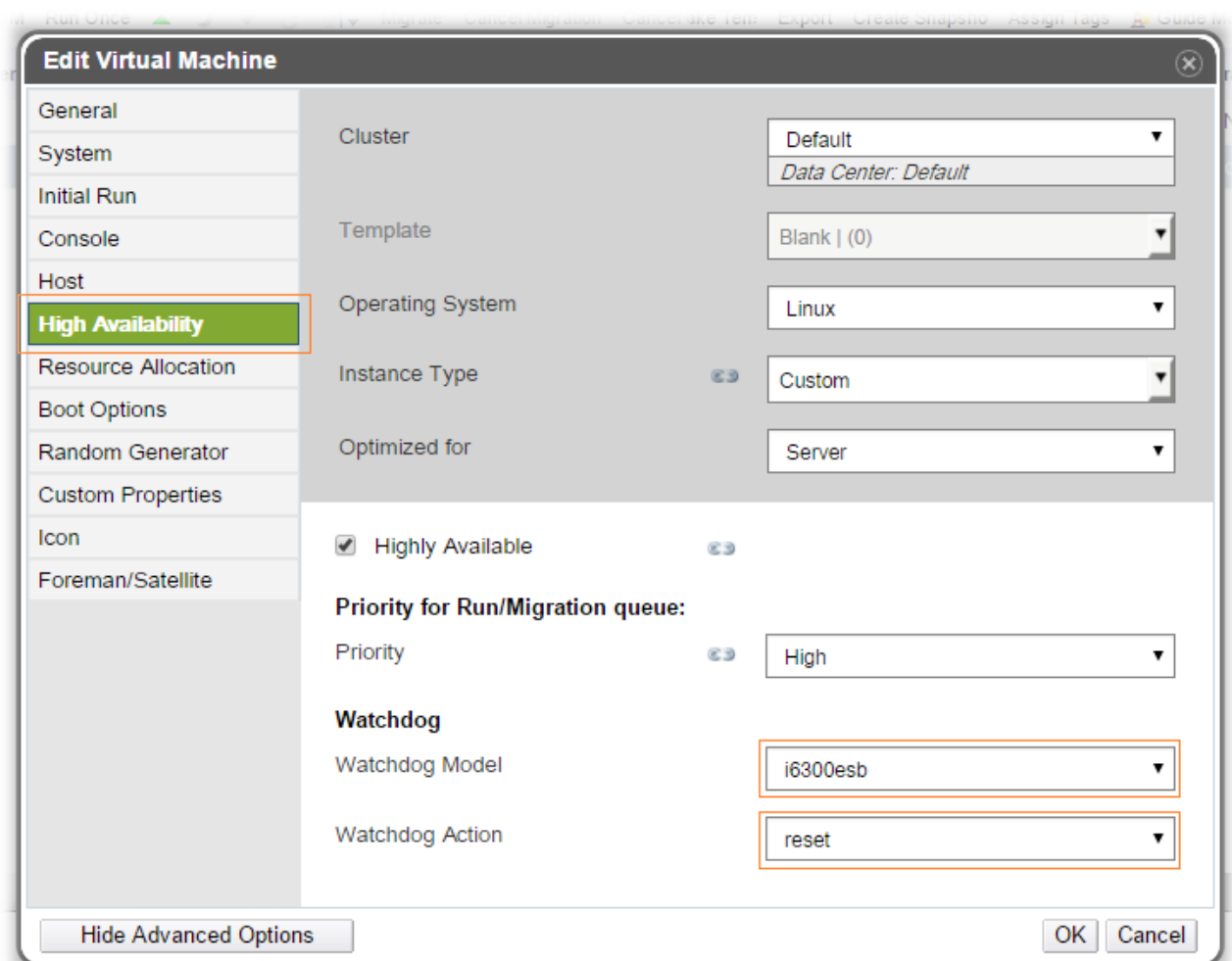


Продолжая тему возможностей обеспечения высокой доступности (**High Availability**) в **oVirt 4.0** нельзя упустить из виду такой функционал, как поддержка виртуальных устройств **Watchdog**. В этой заметке мы рассмотрим практический пример настройки Watchdog-устройства в виртуальной машине с гостевой ОС **Ubuntu Linux 16.04 LTS**.

Рассмотренные в прошлой части механизмы **Fencing** в oVirt реализуют высокую доступность на уровне хостов виртуализации и виртуальных машин, но это механизмы более низкого уровня и они не отвечают за разрешение проблемных ситуаций в плане доступности гостевой ОС внутри самих виртуальных машин. На этот случай, для особо критичных виртуальных машин, в oVirt имеется возможность активировать виртуальное **Watchdog**-устройство в конфигурации ВМ. Об этом типе устройств упоминается в документах [Watchdog Device](#) и [oVirt Administration Guide](#).

Итак, рассмотрим пример настройки Watchdog для ВМ с гостевой ОС **Ubuntu Linux 16.04.1 LTS**.

В веб-консоли oVirt откроем свойства виртуальной машины и перейдём на вкладку **High Availability**. Здесь настроим использование **Watchdog**-устройства. Смысл в том, что в конфигурацию ВМ добавляется новое виртуальное устройство **i6300esb**, для обеспечения работы которого, в гостевую ОС необходимо установить и запустить специальную службу. Эта служба будет обмениваться данными с этим виртуальным Watchdog-устройством. В случае если с гостевой ОС произойдёт какой-то серьёзный сбой и, как следствие, служба станет не доступна, то Watchdog-устройство инициирует перезапуск данной виртуальной машины. В окне **Watchdog Action**, мы должны определить действие, которое oVirt должен выполнять в том случае, если Watchdog-устройство перестанет получать сигналы доступности от Watchdog-службы внутри гостевой ОС. В большинстве сценариев может использоваться **reset**, то есть форсированный перезапуск ВМ. Но есть и другие варианты действий, которые вы сможете использовать в зависимости от своих потребностей - **none**, **poweroff**, **dump**, **pause**...



После загрузки нашей виртуальной машины с гостевой ОС **Ubuntu Linux** убедимся в том, что в системе появилось новое Watchdog-устройство:

```
$ sudo lspci | grep watchdog -i
```

```
user@KOM-AD01-PBX02: ~  
user@KOM-AD01-PBX02:~$  
user@KOM-AD01-PBX02:~$ sudo lspci | grep watchdog -i  
00:09.0 System peripheral: Intel Corporation 6300ESB Watchdog Timer  
user@KOM-AD01-PBX02:~$
```

Установим соответствующую службу из пакета **watchdog**

```
$ sudo install watchdog
```

Сделаем минимальную корректировку конфигурационного файла `/etc/watchdog.conf`, то есть уберём комментарий в одной строке (другие параметры настраиваются при необходимости в зависимости от ваших потребностей):

```
watchdog-device = /dev/watchdog
```

Для проверки подгружаем модуль ядра с драйвером для поддержки нашей модели Watchdog-устройства:

```
$ sudo modprobe i6300esb  
$ sudo modinfo i6300esb
```

```
user@KOM-AD01-PBX02: ~  
user@KOM-AD01-PBX02:~$  
user@KOM-AD01-PBX02:~$ sudo modprobe i6300esb  
user@KOM-AD01-PBX02:~$ sudo modinfo i6300esb  
filename:      /lib/modules/4.4.0-36-generic/kernel/drivers/watchdog/i6300esb.ko  
license:      GPL  
description:   Watchdog driver for Intel 6300ESB chipsets  
author:       Ross Biro and David Härdeman  
srcversion:   D3A534F60D8A3EB0F5F69C3  
alias:       pci:v00008086d000025ABsv*sd*bc*sc*i*  
depends:        
intree:      Y  
vermagic:     4.4.0-36-generic SMP mod_unload modversions  
parm:        heartbeat:Watchdog heartbeat in seconds. (1<heartbeat<2046, default=30) (int)  
parm:        nowayout:Watchdog cannot be stopped once started (default=0) (bool)  
user@KOM-AD01-PBX02:~$
```

После этого попробуем включить и запустить службу **watchdog**:

```
$ sudo systemctl enable watchdog  
$ sudo systemctl start watchdog  
$ sudo systemctl status watchdog
```

Служба, как минимум, должна запуститься без ошибок:

```
user@KOM-AD01-PBX02: ~  
user@KOM-AD01-PBX02:~$ sudo systemctl enable watchdog  
Synchronizing state of watchdog.service with SysV init with /lib/systemd/systemd-sysv-install...  
Executing /lib/systemd/systemd-sysv-install enable watchdog  
user@KOM-AD01-PBX02:~$ sudo systemctl start watchdog  
user@KOM-AD01-PBX02:~$ sudo systemctl status watchdog  
● watchdog.service - watchdog daemon  
   Loaded: loaded (/lib/systemd/system/watchdog.service; static; vendor preset: enabled)  
   Active: active (running) since Thu 2016-09-15 18:58:54 MSK; 16min ago  
     Main PID: 3414 (watchdog)  
       CGroup: /system.slice/watchdog.service  
               └─3414 /usr/sbin/watchdog  
  
Sep 15 18:58:54 KOM-AD01-PBX02 watchdog[3414]: ping: no machine to check  
Sep 15 18:58:54 KOM-AD01-PBX02 watchdog[3414]: file: no file to check  
Sep 15 18:58:54 KOM-AD01-PBX02 watchdog[3414]: pidfile: no server process to check  
Sep 15 18:58:54 KOM-AD01-PBX02 watchdog[3414]: interface: no interface to check  
Sep 15 18:58:54 KOM-AD01-PBX02 watchdog[3414]: temperature: no sensors to check  
Sep 15 18:58:54 KOM-AD01-PBX02 watchdog[3414]: test=none(0) repair=none(0) alive=/dev/watch  
Sep 15 18:58:54 KOM-AD01-PBX02 watchdog[3414]: watchdog now set to 60 seconds  
Sep 15 18:58:54 KOM-AD01-PBX02 watchdog[3414]: hardware watchdog identity: i6300ESB timer  
Sep 15 19:07:36 KOM-AD01-PBX02 systemd[1]: Started watchdog daemon.  
Sep 15 19:14:48 KOM-AD01-PBX02 systemd[1]: Started watchdog daemon.  
user@KOM-AD01-PBX02:~$
```

Теперь нам нужно обеспечить правильную загрузку службы **watchdog** во время загрузки гостевой ОС. Для этого в файл `/etc/default/watchdog` в строку `watchdog_module="none"` вносим имя нашего модуля и при необходимости добавляем параметры `watchdog_options`. В итоге файл должен принять следующий вид:

```
# Start watchdog at boot time? 0 or 1
run_watchdog=1
# Start wd_keepalive after stopping watchdog? 0 or 1
run_wd_keepalive=1
# Load module before starting watchdog
watchdog_module=i6300esb
# Specify additional watchdog options here (see manpage).
watchdog_options="-s -c /etc/watchdog.conf"
```

В файле блэк-листа `/etc/modprobe.d/blacklist-watchdog.conf` среди множества строк находим и комментируем строку с названием нужного нам модуля ядра, чтобы разрешить его загрузку:

```
...
#blacklist i6300esb
...
```

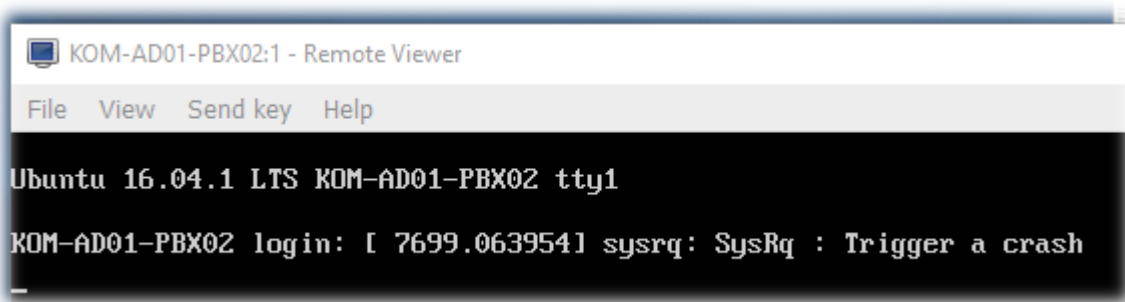
В целом проделанных действий должно быть достаточно, однако, как показывает практика, в некоторых случаях при загрузке системы служба **watchdog** не стартует, поэтому в конец файла `/etc/rc.local` можно добавить излюбленный "костыль", который после запуска системы будет выполнять сначала проверку наличия в системе watchdog-устройства, а затем, если это устройство есть, будет выполнять проверку состояния службы **watchdog**, и если необходимо инициировать запуск этой службы.

```
...
sleep 5
if [ "$(lspci | grep watchdog -i)" ]; then
    if [ -z "$(service watchdog status | grep 'Active: active')" ]; then
        service watchdog start
    fi
fi
exit 0
```

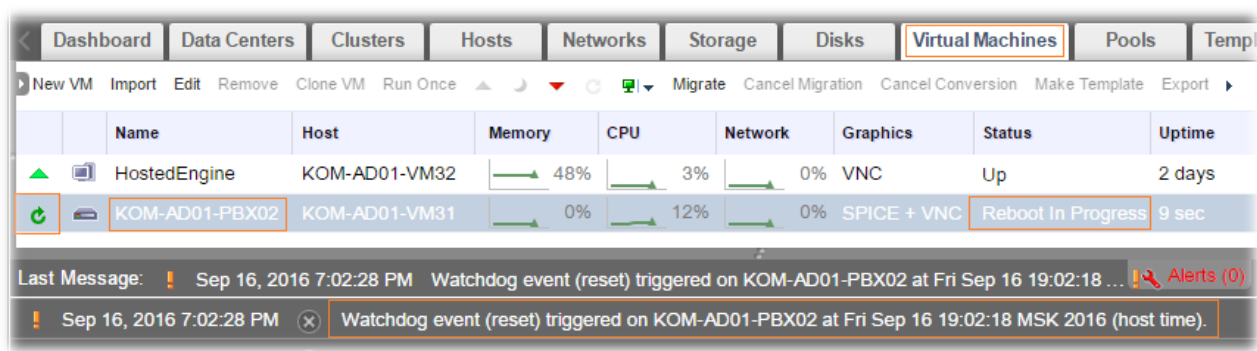
В конце проделанной настройки перезагружаем наш виртуальный сервер с Ubuntu (можно для проверки несколько раз) и убеждаемся в том, что после каждого запуска системы служба **watchdog** успешно стартует. Если с авто-запуском службы проблем нет, можем переходить к проверке совместной работы виртуального watchdog-устройства и watchdog-службы.

Для проверки из под `root` выполним команду, которая вызовет "краш" нашей гостевой Linux-системы:

```
$ sudo su -
# echo c > /proc/sysrq-trigger
```



После выполнения этой команды гостевая ОС "намертво" повиснет и, как следствие, виртуальное watchdog-устройство потеряет связь с watchdog-службой внутри ОС. Спустя 60 секунд (интервал опроса по умолчанию) watchdog-устройство инициирует перезагрузку ВМ, о чём будет выведено соответствующее сообщение во вкладке отображения событий в еб-консоли oVirt



После перезагрузки наша гостевая ОС снова в работе, а пара watchdog-устройство/watchdog-служба снова будет нести для нас свою "нелёгкую вахту".

В связи с тем, что не так давно появилась новая версия **oVirt 4.0.3**, в следующей части серии заметок посвящённых oVirt, мы рассмотрим процедуру обновления **oVirt Hosted Engine** до актуальной новой версии внутри 4 ветки.