


Развёртывание и настройка oVirt 4.0. Часть 11.

Резервное копирование oVirt Hosted Engine

 blog.it-kb.ru/2016/10/06/install-ovirt-4-0-part-11-hosted-engine-4-0-4-backup-script-with-engine-backup-tool-and-theoretical-recovery-plan

Автор:Алексей Максимов

06.10.2016



Рассматривая тему резервного копирования в **oVirt 4.0**, выделю два основных направления – это резервное копирование данных самой системы управления **oVirt Engine** и резервное копирование виртуальных машин. В этой части мы рассмотрим конкретный практический пример автоматизации регулярного резервного копирования данных **oVirt Hosted Engine 4.0.4** и с теоретической точки зрения поговорим о восстановлении.

Утилита engine-backup

Резервное копирование данных oVirt Engine представляет собой процедуру сохранения всех настроенных конфигурационных файлов и баз данных postgresql с помощью утилиты **engine-backup**. Утилита имеет встроенную подсказку, которая вызывается соответствующей опцией `--help`:

```
# engine-backup --help
```

```
engine-backup: back up and restore ovirt-engine environment
```

```
USAGE:
```

```
    /usr/bin/engine-backup [--mode=MODE] [--scope=SCOPE] [--file=FILE] [--log=FILE]
```

```
    MODE is one of the following:
```

backup	back up system into FILE
restore	restore system from FILE
verify	verify FILE

```
...
```

К сожалению man-страница у этой утилиты имеет неактуальную информацию, поэтому на данный момент лучше пользоваться именно встроенной подсказкой. Также дополнительную информацию о работе утилиты можно найти в документе [oVirt Develop - oVirt-engine-backup](#). На данный момент утилита имеет три режима работы, которые задаются опцией `--mode`, - это режим создания резервной копии, режим восстановления и режим проверки файла резервной копии.

Особенности резервного копирования oVirt Hosted Engine

Когда я изучал вопрос автоматизации резервного копирования данных oVirt Engine с помощью утилиты **engine-backup**, то из более или менее готовых решений нашёл только скрипт [Learn Linux - Ovirt Manager Backup Script](#), но в предложенном виде скрипт не подходит для конфигурации oVirt в развёртывании **Hosted Engine**, так как здесь имеются свои особенности, которые нужно учитывать в процессе создания

резервной копии. При резервном копировании oVirt Hosted Engine желательно использовать определённую последовательность действий, которая изложена в документе [oVirt Documentation - oVirt Hosted Engine Backup and Restore](#).

Всю процедуру можно разделить на три высокоуровневых этапа:

Этап 1. Выполняется команда включения глобального режима обслуживания **hosted-engine** на хосте oVirt (не на любом хосте, а на одном из хостов, где был развёрнут hosted-engine):

```
# hosted-engine --set-maintenance --mode=global
```

Этап 2. На виртуальной машине oVirt Hosted Engine выполняются команды остановки службы **ovirt-engine**, резервного копирования (**engine-backup**) и последующего запуска службы **ovirt-engine**:

```
# service ovirt-engine stop
# engine-backup --mode=backup --scope=all --file=backup.tar --log=backup.log
# service ovirt-engine start
```

Этап 3. Выполняется команда отключения глобального режима обслуживания **hosted-engine** на хосте oVirt (не на любом хосте, а на одном из хостов, где был развёрнут hosted-engine):

```
# hosted-engine --set-maintenance --mode=none
```

Таким образом задача усложняется тем, что вся процедура резервного копирования требует последовательного выполнения команд на разных серверах oVirt. Чтобы "одним выстрелом убить всех зайцев", мы будем запускать скрипт резервного копирования на виртуальной машине oVirt Hosted Engine и в нём использовать вызов команд на удалённом сервере (хосте oVirt) по протоколу **SSH**.

Сделаю небольшое отступление о том, каким образом будет обеспечена аутентификация в SSH-сессии между серверами в рамках нашей задачи. В процессе добавления хоста в инфраструктуру oVirt между хостом и сервером управления oVirt Engine устанавливаются доверительные отношения для возможности создания SSH сессий без лишних запросов авторизации. Как я понял, в частности реализовано это путём добавления публичного ключа **ssh-rsa** сервера oVirt Engine в файл `/root/.ssh/authorized_keys` на хостах oVirt. В свою очередь на сервере oVirt Engine в файле `/etc/pki/ovirt-engine/keys/engine_id_rsa` можно найти приватный ключ, который связан с выше обозначенным публичным ключом. Таким образом, всё что нам нужно сделать, это использовать этот самый приватный ключ в целях аутентификации при подключении сервера oVirt Engine к любому из хостов по протоколу SSH.

Для того, чтобы проверить то, что такая аутентификация будет работать, попробуем выполнить произвольную команду на удалённом хосте oVirt (например KOM-AD01-VM31) с сервера oVirt Engine:

```
# ssh -i /etc/pki/ovirt-engine/keys/engine_id_rsa root@KOM-AD01-VM31.holding.com  
"hostname"
```

KOM-AD01-VM31.holding.com

Как видим, команда выполнена на удалённом хосте успешно без каких-либо запросов аутентификации. Соответственно, мы сможем использовать такой вызов команд в скрипте резервного копирования, чтобы соблюсти требуемую последовательность операций до и после процедуры резервного копирования oVirt Hosted Engine.

Подготовка инфраструктуры

Настроим **NFS** сервер для хранения резервных копий oVirt Engine. Создадим отдельную NFS-шару, к которой ограничим доступ - разрешим доступ на чтение и запись в шару только для NFS-клиента с сервера oVirt Engine.

На NFS-сервере создаём каталог для сохранения резервных копий oVirt Engine:

```
# mkdir -p /mnt/mdadm-vv1/nfs/ovirt-engine-backup  
# chmod -R 600 /mnt/mdadm-vv1/nfs/ovirt-engine-backup
```

Настраиваем экспорт NFS-шары, добавив в файл /etc/exports строку вида:

```
/mnt/mdadm-vv1/nfs/ovirt-engine-backup KOM-AD01-  
OVIRT1.holding.com(rw, sync, no_root_squash, no_all_squash)
```

Заставляем службу **nfs-server** пересчитать конфигурацию и проверяем результат:

```
# exportfs -r  
# exportfs  
  
/mnt/mdadm-vv1/nfs/ovirt-engine-backup  
KOM-AD01-OVIRT1.holding.com  
  
***
```

Переходим на сервер oVirt Engine и для начала создаём папку, в которую будем монтировать ранее настроенную NFS-шару для сохранения резервных копий:

```
# mkdir -p /mnt/nfs/ovirt-engine-backup  
  
***
```

Так как скрипт, который мы будем использовать для резервного копирования, имеет возможность формировать письма для отсылки уведомлений по электронной почте, то дополнительно на сервере oVirt Engine нам желательно настроить почтовый клиент. Сделать это можно, например, настроив службу **postfix**, по аналогии с тем, так как это описано в нашей Вики-статье: [Как настроить отсылку уведомлений на внешний почтовый сервер с помощью postfix в CentOS](#)

Скрипт резервного копирования oVirt Hosted Engine

Создаём скрипт резервного копирования:

```
# nano /usr/local/sbin/ovirt-engine-backup.sh
```

Наполняем скрипт следующим содержимым:

```

#!/bin/bash
#
#
oVirtHEHost="KOM-AD01-VM31.holding.com"
oVirtEngineKey="/etc/pki/ovirt-engine/keys/engine_id_rsa"
#
NFSShare="KOM-FS03.holding.com:/mnt/mdadm-vv1/nfs/ovirt-engine-backup/";
NFSPPoint="/mnt/nfs/ovirt-engine-backup";
BackupAge="30"
#
MailMe=true;
MailMeOnlyErrors=false;
MailTo="DST-oVirt-Admins@holding.com";
MailFrom="KOM-AD01-OVIRT1@holding.com";
#
(
echo "Mount share $NFSShare to $NFSPPoint ..."
mount -t nfs $NFSShare $NFSPPoint/

if mount | grep $NFSPPoint > /dev/null; then

    echo "Delete the old backups from $NFSPPoint ..."
    if [ $(find $NFSPPoint/ -name '*.xz' -mtime -${($BackupAge-1)} | wc -l) -eq 0
]; then
        echo "No files to delete."
    else
        echo "Delete backups greater than $BackupAge days old"
        find $NFSPPoint/ -name '*.xz' -mtime +$BackupAge -delete
        find $NFSPPoint/ -name '*.log' -mtime +$BackupAge -delete
    fi

    echo "oVirt Hosted Engine backup operations started `date`...";

    echo "Start oVirt Hosted Engine Global Maintenance Mode on host $oVirtHEHost
...";
    ssh -i $oVirtEngineKey root@$oVirtHEHost "hosted-engine --set-maintenance --
mode=global 2>/dev/null"

    echo "Stop oVirt Engine (ovirt-engine) Service ...";
    /bin/systemctl stop ovirt-engine.service

    echo "Run engine-backup..."
    BcpTimeStamp=$(date +%Y.%m.%d-%H:%M:%S");
    BcpFileName=$NFSPPoint/ovirt.engine.backup.$BcpTimeStamp;
    /usr/bin/engine-backup --mode=backup --scope=all --file=$BcpFileName.xz --
log=$BcpFileName.log --archive-compressor=xz --files-compressor=None
    echo "Created files:";echo "`ls -la $BcpFileName*`"

    echo "Start oVirt Engine (ovirt-engine) Service ...";
    /bin/systemctl start ovirt-engine.service

    echo "Stop oVirt Hosted Engine Global Maintenance Mode on host $oVirtHEHost
...";
    ssh -i $oVirtEngineKey root@$oVirtHEHost "hosted-engine --set-maintenance --
mode=none 2>/dev/null"

```

```

    echo "Unmount share $NFSShare from $NFSPPoint ..."
    umount $NFSPPoint/
    echo "oVirt Hosted Engine backup operations completed `date`"

else
    echo "Error. Share $NFSShare can not be mounted to $NFSPPoint"
fi
)1>/tmp/ovirt-engine-backup-last-output.txt 2>/tmp/ovirt-engine-backup-last-
error.txt

if $MailMe; then

if [ -s /tmp/ovirt-engine-backup-last-error.txt ]
then
(
    echo "To: $MailTo"
    echo "From: $MailFrom"
    echo "Subject: [Alert] $0 script Error `hostname`"
    echo -e "Dear oVirt Administrator,\n\n"
    echo -e "Contents of /tmp/ovirt-engine-backup-last-error.txt :\n\n"
    cat /tmp/ovirt-engine-backup-last-error.txt
    echo -e "\n\n*** THIS IS A AUTO GENERATED ALERT GENERATED AT `date` ***\n"
)| /usr/sbin/sendmail -t -f $MailTo
else
    if ! $MailMeOnlyErrors; then
        (
            echo "To: $MailTo"
            echo "From: $MailFrom"
            echo "Subject: [INFO] oVirt Engine backup completed on `date +%d-%B-%Y`"
            echo -e "Dear oVirt Administrator,"
            echo -e "\n\noVirt Engine backup completed on `date +%d-%B-%Y`"
            echo -e "\nScript running in host: `hostname`"
            echo -e "\nScript name: $0"
            echo -e "\nScript output in /tmp/ovirt-engine-backup-last-output.txt
:\n\n"
            cat /tmp/ovirt-engine-backup-last-output.txt
            echo -e "\n\n*** THIS IS A AUTO GENERATED ALERT GENERATED AT `date` ***\n"
            )| /usr/sbin/sendmail -t -f $MailTo
        fi
    fi
fi
fi
fi

```

Общая логика работы скрипта будет следующей:

- В локальный каталог (NFSPPoint) монтируем NFS-шапу (NFSShare) для хранения резервных копий.
- Удаляем старые резервные копии (старше BackupAge дней). При этом если создание новых резервных копий по какой-то причине ранее перестало работать и свежих файлов, то удаление старых копий не производится.
- Удалённо включаем режим обслуживания hosted-engine на хосте oVirt (oVirtHEHost)
- Останавливаем службу oVirt Engine

- Создаём резервную копию oVirt Engine с сохранением файлов в смонтированную NFS-шапу
- Запускаем службу oVirt Engine
- Удалённо отключаем режим обслуживания hosted-engine на хосте oVirt (oVirtHEHost)
- Отключаем NFS-шапу с резервной копией.

При необходимости можно включить режим оповещений на электронную почту о результате работы скрипта (MailMe). Включённая переменная MailMeOnlyErrors означает то, что оповещения будут отсылаться только в случае возникновения проблем. Переменные MailTo и MailFrom определяют адреса получателя и отправителя сообщений.

Обратите внимание на то, что при вызове команды **hosted-engine** для включения режима обслуживания используется подавление вывода "2>/dev/null", что само по себе не очень правильно. Связано это с тем, что текущая версия утилиты hosted-engine при каждом вызове ключей управления режимом обслуживания выплёвывает сообщение:

```
/usr/lib/python2.7/site-
packages/ovirt_hosted_engine_ha/lib/storage_backends.py:15: DeprecationWarning:
vdscli uses xmlrpc. since ovirt 3.6 xmlrpc is deprecated, please use
vdsm.jsonrpcvdscli
import vsdm.vdscli
```

Логика работы нашего скрипта такова, что появление вывода от работы вызываемых нами команд свидетельствует о какой-то нештатной ситуации, о чём формируется предупреждающее письмо. По сути, вышеуказанное сообщение не несёт никакой смысловой нагрузки для тех, кто эксплуатирует в повседневной работе утилиту hosted-engine, а предназначено больше для разработчиков. После обсуждения этой мелкой неприятности в mail-группе был зарегистрирован баг, где разработчики пообещали убрать это сообщение в одном из ближайших обновлений VDSM, после чего можно будет удалить подавление вывода "2>/dev/null" из скрипта.

Наполнив скрипт содержимым, делаем его исполняемым и ограничиваем к нему доступ:

```
# chmod 700 /usr/local/sbin/ovirt-engine-backup.sh
```

Пробуем выполнить скрипт с правами супер-пользователя (если работаем из под **sudo**, то перед запуском скрипта выполним команду "sudo su -"):

```
# /usr/local/sbin/ovirt-engine-backup.sh
```

Скрипт должен отработать без ошибок и отправить подробный отчёт о работе по электронной почте.

Добавляем задание резервного копирования в планировщик **cron**:

```
# nano /etc/cron.d/ovirt-engine-backup
```

Содержимое файла задания на ежедневное выполнение в 01:00:

```
# Daily oVirt Engine backup
00 01 * * * root /usr/local/sbin/ovirt-engine-backup.sh
```

Процедура восстановления oVirt Hosted Engine

Процедура восстановления oVirt Hosted Engine из резервной копии в зависимости от обстоятельств, при которых требуется восстановление, может представлять собой весьма нетривиальный процесс, и, вполне возможно, не без подводных камней, как выяснилось в последней беседе в mail-группе oVirt.

Наглядного примера восстановления я привести не смогу, так как пока не имел практического опыта такой работы, а проводить эксперименты на уже работающей инфраструктуре oVirt, с учётом даже минимальной возможности неприятных сюрпризов не очень хочется. Однако, если получится так, что мне придётся столкнуться с этой задачей на практике, то я, по мере возможности, постараюсь дополнительно описать этот процесс. Также буду признателен всем эксплуатирующим oVirt за любые дельные практические советы и комментарии по этому поводу. Возможно у кого-то даже появится желание написать в наш блог свою статью по этому поводу на базе собственного практического примера.

Если же говорить о процедуре восстановления с теоретической точки зрения, то найти информацию по ней на данный момент мне удалось в двух документах:

Причём первый документ, как я понял, уже несколько отстал от текущих возможностей утилиты engine-backup, а второй документ не даёт полной картины возможных сценариев восстановления. В конечном итоге, с учётом обоих выше обозначенных документов, а также с учётом комментариев Simone Tiraboschi, был сформирован некий теоретический план, которым можно будет попробовать воспользоваться при необходимости восстановления oVirt Hosted Engine из резервной копии.

Предварительные условия:

- Виртуальная машина с oVirt Hosted Engine более недоступна.
- Подготовлено чистое дисковое хранилище (NFS/iSCSI/FC), на котором будет размещена новая виртуальная машина с oVirt Hosted Engine.
- Имеется полная резервная копия oVirt Hosted Engine, сделанная ранее с помощью утилиты engine-backup.

Порядок восстановления состоит из трёх высокоуровневых этапов:

Этап 1. Выполняется на всех хостах oVirt (если они всё ещё доступны).

Останавливаем и удаляем службы, отвечающие за высокую доступность Hosted Engine:

```
# service ovirt-ha-broker stop
# service ovirt-ha-agent stop
# chkconfig --del ovirt-ha-broker
# chkconfig --del ovirt-ha-agent
```

Этап 2. На одном из хостов (это будет первый хост в повторном развёртывании oVirt Hosted Engine) выполняется:

```
# hosted-engine --deploy
```

Процесс развёртывания будет схож с тем, что делается при первом развёртывании с некоторыми важными нюансами:

- Используйте прежнее имя **FQDN** для создания виртуальной машины **oVirt Hosted Engine**.
- Используйте **чистое** дисковое хранилище (NFS/iSCSI/FC) для размещения новой виртуальной машины с oVirt Hosted Engine
- Используйте тот же пароль для учётной записи **admin**, что использовался ранее.

- Установите **чистую ОС на ВМ** и подтвердите то, что ОС была установлена.

На этом этапе внутри виртуальной машины **oVirt Hosted Engine**:

- Установите на ВМ пакеты ovirt-engine, но НЕ запускайте **engine-setup**:

```
# yum install http://resources.ovirt.org/pub/yum-repo/ovirt-release40.rpm
# yum install epel-release
# yum install ovirt-engine
```

- Разверните архив с резервной копией во временную папку и выполните команду восстановления oVirt Engine:

```
# engine-backup --mode=restore --file=file_name --log=log_file_name --provision-db --provision-dwh-db --restore-permissions
```

Здесь Simone предупреждает нас о том, что на данном этапе по нам потребуется удалить из восстановленной базы данных информацию о старом **Storage Domain**, на котором ранее размещалась виртуальная машина **Hosted Engine**. Однако на текущий момент нет точно отлаженной процедуры для этого действия и ведётся разработка дополнительной утилиты под эту задачу, вызов которой в перспективе может быть организован путём добавления дополнительной опции к команде **engine-backup**. Ждём этой опции с выходом oVirt 4.1.

- Выполните команду:

```
# engine-setup
```

- Откройте веб-портал администрирования oVirt с только что развёрнутого сервера Hosted Engine и **удалите** информацию о хостах, участвовавших ранее в развёртывании Hosted Engine
- В утилите развёртывания **hosted-engine** подтвердите том, что установка engine была завершена (выбрав пункт **1** в ожидающем меню).
- Завершите процедуру развёртывания **hosted-engine**.
- Дождитесь запуска служб высокой доступности на хосте и зайдя на веб-портал администрирования oVirt **активируйте** вновь добавленный хост.

Этап 3. На дополнительных хостах выполните, как при обычной первичной установке:

```
# hosted-engine --deploy
```

При этом с версии **oVirt 4.1** возможность добавления хостов через указанную команду будет исключена. В качестве альтернативного, и единственно возможного, варианта будет добавление хоста через веб-портал администрирования oVirt.
