veeam, backup, vmware, mysql

Veeam Backup

La instalación puede ser en virtual o en físico, en principio da igual sólo depende de los recursos que tengamos disponibles .

Hay que tener cuidado con donde seleccionamos la ubicación del **NFsDatastore** ya que si hacemos un instant recovery es en esa ubicación donde se crea temporalmente la máquina recuperada

Recomendaciones

1. Realizar una copia a otro repositorio del backup que realiza de su propio catálogo
2. Activar CBT (Change block truck). Antes de aplicarlos no tener snapshots abiertos previamente

Backup consistente de una base de datos mysql/mariadb

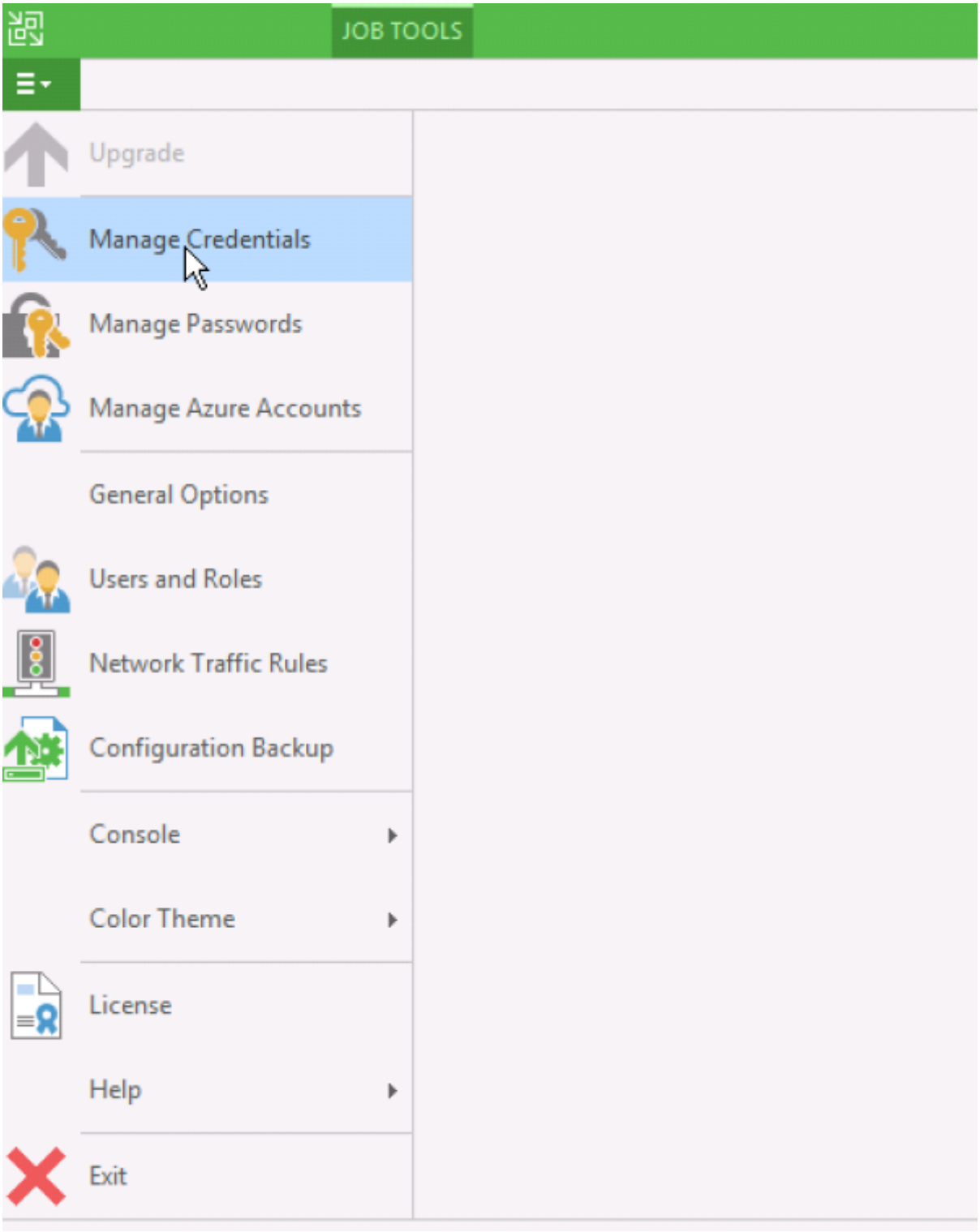
Para hacer la copia de seguridad podemos optar por tres métodos distintos, previos a la realización del snapshot :

- Ejecutando un script que bloquea la BDD para hacer un volcado de la misma,
- Parando el servicio de BDD por completo
- Poniendo la BDD en modo lectura para hacer el volcado de la misma

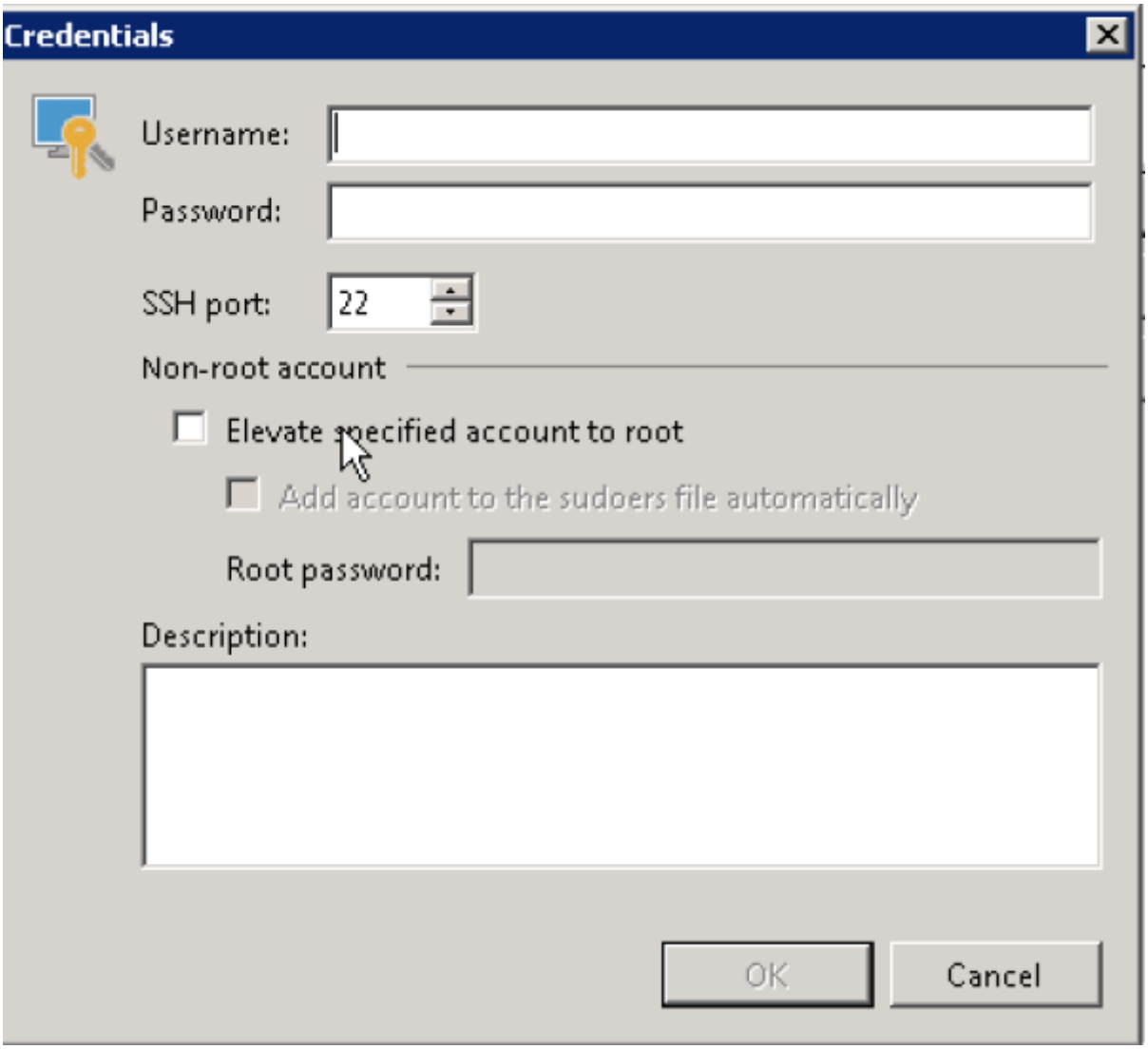
Nosotros vamos a optar por el último ya que implica no parar el recurso, aunque tampoco podemos realizar modificaciones mientras la bdd está bloqueada en modo lectura. Además el tiempo de congelación de la BDD es corto ya que sólo dura el tiempo que veeam backup tarda en hacer el snapshot previo a la copia.

Paso 1

Lo primero que tenemos que hacer es definir una cuenta que tenga permisos en la mv destino para poder ejecutar el script. Para eso vamos al Veeam Backup → Manage Credentials



Añadimos una nueva, en mi caso del tipo Linux Account



Paso 2

Una vez creada la cuenta que vamos a utilizar para ejecutar los scripts, lo siguiente será crear los scripts. Recomiendo crearlos con el Notepad++ ya que tenemos que guardarlos aplicando la compatibilidad para linux. Es decir, antes de guardar el fichero deberemos ir al Notepad+++→Edit→EOL Conversion y elegir la opción **UNIX/OSF Format** Crearemos dos scripts uno para bloquear la BDD y el otro para desbloquearla una vez creado el snapshot

precongelarmysql.sh

```
#!/bin/bash
# config:
# Por defecto usamos el usuario y la clave que hay que definir en
/etc/my.cnf
# En caso contrario, descomentar la línea inferior y poner usuario y
contraseña para la bdd
#use_credentials="-uroot -pcontraseña"
defaults_file="/etc/my.cnf"
```

```
timeout=300
lock_file=/tmp/mysql_tables_read_lock
###
if [ -f $defaults_file ]; then
opts="--defaults-file=$defaults_file"
fi
if [ -n $use_credentials ]; then
opts="$opts $use_credentials"
fi
sleep_time=$((timeout+10))
rm -f $lock_file
echo "$0 executing FLUSH TABLES WITH READ LOCK" | logger
mysql $opts -e "FLUSH TABLES WITH READ LOCK; system touch $lock_file; system
nohup sleep
$sleep_time; system echo\ lock released|logger; " > /dev/null &
mysql_pid=$!
echo "$0 child pid $mysql_pid" | logger
c=0
while [ ! -f $lock_file ]
do
# check if mysql is running
if ! ps -p $mysql_pid 1>/dev/null ; then
echo "$0 mysql command has failed (bad credentials?)" | logger
exit 1
fi
sleep 1
c=$((c+1))
if [ $c -gt $timeout ]; then
echo "$0 timed out waiting for lock" | logger
touch $lock_file
kill $mysql_pid
fi
done
echo $mysql_pid > $lock_file
exit 0
```

Referencias

- https://www.virtuallifestyle.nl/2013/03/back-up-mysql-on-linux-without-stopping-services-or-dumping-the-database/
- https://www.veeam.com/wp-consistent-protection-mysql-mariadb.html

From:
http://wiki.intrusos.info/ - **LCWIKI**

Permanent link:
http://wiki.intrusos.info/doku.php?id=aplicaciones:backup:veeam&rev=1526562448

Last update: **2023/01/18 14:12**