Linux LVM Snapshot VM Backup Script ...

 bradsjm 3 posts since

Jul 6, 2006 I thought it might be useful to someone so I'm posting the perl script I use to do daily backups of my Virtual Machines.

The script handles running machines by suspending them, taking an LVM snapshot, then resuming the VM and doing a backup of the snapshot. This gives me a window of less than 60 seconds of downtime per running virtual machine. You will need to customize this to your own LVM volume names, paths, etc.

Note: The editor seems to modify the text so I have attached the script.

```
#!/usr/bin/perl -w
#
```

1. VM Server Virtual Machine Snapshot Backups

```
#

use VMware::VmPerl;
use VMware::VmPerl::VM;
use VMware::VmPerl::Server;
use VMware::VmPerl::ConnectParams;
use File::Basename;
use strict;

my $backupdir = "/vm/Backups/daily";
my $log_file = "/var/log/vmware/vmbackup.log";

my ($server, $vm);

my $server_name = undef;
my $user = undef;
my $passwd = undef;
```

1. Use the default port of 902. Change this if your port is different.

```
my $port = 902;

open LOG, ">>$log_file";
log_stuff( "Starting backups" );

#
```

1. Connect to VMware Server

```
#
my $connect_params = VMware::VmPerl::ConnectParams::new($server_name,$port,$user,$passwd);
```

```
$server = VMware::VmPerl::Server::new();

if (!$server->connect($connect_params)) {
my ($error_number, $error_string) = $server->get_last_error();
log_stuff( "Connect to server: $error_number: $error_string\n" );
die "Could not connect to server: Error $error_number: $error_string\n";
}

#
```

1. Get list of Virtual Machines

```
#
my @vmlist = $server->registered_vm_names();
if (!defined($vmlist[0])) {
my ($error_number, $error_string) = $server->get_last_error();
log_stuff( "get_vms: $error_number: $error_string" );
die "Could not get list of VMs from server: Error $error_number: $error_string\n";
}

#
```

1. Virtual Machine Backup

```
#
foreach my $conf (@vmlist) {
my ($name,$vmdir,$suffix) = fileparse($conf, '\..*' );
```

1. connect to virtual machine

```
$vm = VMware::VmPerl::VM::new();
if (!$vm->connect($connect_params, $conf)) {
my ($error_number, $error_string) = $vm->get_last_error();
log_stuff( "Connect to $conf: $error_number: $error_string\n" );
die "Could not connect to $conf: Error $error_number: $error_string\n";
}
```

1. Get the virtual machine display name

```
my $displayName = $vm->get_config("displayName");
$displayName =~ s/ \()//g; # remove special characters
```

1. Build the path for the backup filename

```
my $backup = "$backupdir/$displayName.tar.gz";
```

1. Get the current state for the virtual machine

```
my $state = $vm->get_execution_state();
```

1. Log what we are doing

```
log_stuff( "$displayName: Starting backup" );
```

1. handle virtual machines that are running by using lvm snapshot with a suspend

```
if ( $state eq VM_EXECUTION_STATE_ON )
{
log_stuff( "$displayName: Suspending Virtual Machine" );
if (!$vm->suspend()) {
my ($error_number, $error_string) = $vm->get_last_error();
log_stuff( "suspend $conf: $error_number: $error_string\n" );
}
else
{
log_stuff( "$displayName: Creating LVM Snapshot" );
log_stuff( `lvremove -f /dev/data/vmsnapshot` );
log_stuff( `lvcreate -L9.85G -s -n vmsnapshot /dev/data/vmware` );
log_stuff( "$displayName: Resuming Virtual Machine" );
if (!$vm->start()) {
my ($error_number, $error_string) = $vm->get_last_error();
log_stuff( "start $conf: $error_number: $error_string\n" );
}
log_stuff( "$displayName: Mounting LVM Snapshot" );
log_stuff( `mount -t ext3 -v -r /dev/data/vmsnapshot /mnt` );
```

1. fix directory to use the snapshot instead of the actual file

```
$vmdir =~ s!/vm/Machines!/mnt!g;
log_stuff( "$displayName: Backup $vmdir -> $backup.tmp" );
log_stuff( `tar czpf $backup.tmp -C $vmdir .` );
log_stuff( "$displayName: Removing LVM Snapshot" );
log_stuff( `umount -v /mnt` );
log_stuff( `lvremove -f /dev/data/vmsnapshot` );
}
}
else
{
```

1. regular backup of suspended or offline machines

```
log_stuff( "$displayName: Backup $vmdir -> $backup.tmp" );
log_stuff( `tar czpf $backup.tmp -C $vmdir .` );
}
log_stuff( "$displayName: Moving $backup.tmp to $backup" );
log_stuff( `mv -vf $backup.tmp $backup` );
log_stuff( "$displayName: Backup Completed" );
$vm->disconnect();
}
#
```

1. Logging

```
#
sub log_stuff
{
my $msg = shift();
if ( $msg )
{
chomp($msg);
my ($sec,$min,$hour,$mday,$mon,$year,$wday,$yday,$isdst)=localtime(time);
printf LOG "%4d-%02d-%02d %02d:%02d:%02d $msg\n", $year+1900,$mon+1,$mday,$hour,$min,$sec;
}
}
```

1. Destroys the server object, thus disconnecting from the server.

```
$server->disconnect();
undef $server;
log_stuff( "Completed backups" );
close LOG;
```

Attachments:

- vmbackup.pl (4.3 K)

Tags: linux, lvm, backup, snapshot

arman68 *11 posts since Jun 10, 2006* 1. **Re: Linux LVM Snapshot VM Backup Script** Mar 13, 2008 7:46 AM

Thanks for your script. I have been meaning to do something like this for a while now, but never came round to it. Could you please give us some more details on what it does? I have had a quick look at the script and this is what I understand (correct me if I am wrong):

- the local server name, username and password need to be filled in
- it automatically enumerates all the vm

- vm which are running are first suspended (via vmware), a lvm snapshot is taken, and then the vm is resumed via vmware
- the backup uses the vmware files directly, or from the lvm snapshot (when needed)
- backups with tar+gzip goes to a local file
- lvm snapshots are deleted once the backup is completed

A few comments/requests:

- using lzop compression instead of gzip would put much less stress on the server by compressing faster (~80%) and with only ~10% extra space required; an option to choose between lzop and gzip would be useful
- can the backups go somewhere else than the local server, and if so, how?
- an exclude list of vm would be useful
- same for the possibilty to run the script to backup only 1 specific vm


gordho *10 posts since*
Jun 15, 2006 2. **Re: Linux LVM Snapshot VM Backup Script** Apr 1, 2008 7:59 AM

in response to: arman68 You should be able to backup the files to any location by either modifying the **my $backupdir = "/vm/Backups/daily";** variable or by mounting an nfs or cifs share to /vm/Backups/daily.


postonra *22 posts since*
Jul 14, 2005 3. **Re: Linux LVM Snapshot VM Backup Script** Jun 4, 2008 2:34 PM

in response to: gordho

Is this a script tied to VM Server or is this a tool that could be used with ESX ? Where are the LVM snapshots being created inside the VMs or on VM Server or ESX ?


allenby *5 posts since*
Jul 18, 2008 4. **Re: Linux LVM Snapshot VM Backup Script** Jul 18, 2008 2:16 PM

is it possible to do the snapshot with out suspending the vm?


Peter_vm *9,058 posts since*
Feb 1, 2006 5. **Re: Linux LVM Snapshot VM Backup Script** Jul 18, 2008 5:28 PM

in response to: allenby Yes.

It also possible to power-off VM without running shutdown.


allenby *5 posts since*

Jul 18, 2008 6. **Re: Linux LVM Snapshot VM Backup Script** Jul 19, 2008 8:54 AM

in response to: Peter_vm so I can ignore the "suspending" part of that scrip and it still should work right


Peter_vm *9,058 posts since*
Feb 1, 2006 7. **Re: Linux LVM Snapshot VM Backup Script** Jul 19, 2008 11:47 AM

in response to: allenby It all depends what you mean by "should work right". Snapshot would complete successfully. Would that snapshot be worth anything? That depends.


bcnx *3 posts since*
May 8, 2007 8. **Re: Linux LVM Snapshot VM Backup Script** Oct 8, 2009 6:59 AM

Hi,

can anyone give me an indication on how much extra space you need for snapshot for a machine (in percentages that is). I guess it's related to how much the files change in that timespan, but this is really hard to predict with VMware.

Thank you,

Bart