Subject: Re: vzdump snapshot problem and workaround for it
Posted by jarcher on Thu, 12 Apr 2007 06:23:43 GMT

View Forum Message <> Reply to Message

I did some more checking, and have this to add:

I think I found the trouble, which is in the get_device function. Now, just as a warning, my PERL is really, really rusty. I did some many years ago and quickly decided I'll stick to C/C++. But I'll do my best to provide helpful information to the developer of vzdump.

My VPS private areas live on a LVM2 LV, which is ultimately on an iSCSI SAN.

I run this:

```
actual:/home/jim# vzdump --dumpdir /var/local/vps-bu --snapshot 108
```

As soon as get_device is entered, $dir is set to:

```
/mnt/compvps/vz/private/108
```

So running df against the private area for the VPS results in:

```
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/mapper/actual--vg001-vps--storage
                     100756920   6463816  89174948   7% /mnt/compvps
```

So of course $out becomes:

```
/dev/mapper/actual--vg001-vps--storage
```

Parsing $out with the specified regex results in:

```
res[0]: /dev/mapper/actual--vg001-vps--storage
res[5]:
```

At this point, I don't follow this:

```
($vg, $lv) = @{$devmapper->{$dev}} if defined $devmapper->{$dev};
```

I see $devmapper is defined as a global but I can't find $dev, so I think what is happening is that since $devmapper->$dev is undefined, $vg and $lv are never being set, hence the errors I see reported.

Also, I added a few print statements and I see that the get_device function is being called 3 times, at least on my system.

I hope this helps someone.

```
sub get_device {
    my $dir = shift;

    open (TMP, "df '$dir'|");
    <TMP>; #skip first line
    my $out = <TMP>;
    close (TMP);

    my @res = split (/\s+/, $out);

    my $dev = $res[0];
    my $mp = $res[5];
    my ($vg, $lv);

    ($vg, $lv) = @{$devmapper->{$dev}} if defined $devmapper->{$dev};

    return wantarray ? ($dev, $mp, $vg, $lv) : $dev;
}
```