

```

#!/usr/bin/perl

# Disk Space monitor for OVM
# Monitor usage of the Oracle VM Repositories
# Require SSH keys for the admin port on the OVM CLI interface
# (usually port 10000), setup here https://wikis.oracle.com/display/oraclevm/Oracle+VM
+Manager+3+CLI
#
# Kevin Creason ckevinj@gmail.com
#
# 20150210 ver 1.0 of OVM Repository Monitor

# Global Basics to tweak for your installation:
my $OVMSHOST='admin@hostname:10000'; # should be in the notation user@host:port for the OVM
CLI
my $MAILTO='your best friends email address'; # who receives the alerts?
my $MAILFROM='no-reply'; # who do the alerts come from?
my $MAILHOST="smtpforwarder"; # who can forward the email alerts?
my $DEBUG=0;
my $PCused="87"; # percent used threshold
my $FSfree="200"; # gigs free of file system threshold

# Dependencies
use MIME::Lite;
use HTML::HashTable;
use Net::OpenSSH;

# Global variables
my $ssh;
my $sendalert=0;

# Get the Repo List and hash it out
# It has DOS carriage returns! who does that ?
&ssh_connect;
my $rawrepolist=$ssh->capture("list repository"); $rawrepolist=~s/\r//g;
if ($DEBUG>1){print STDERR "Repolist capture: \"$rawrepolist\"\n"; }
my @repolist=split(/\n/, $rawrepolist);
if ($DEBUG>1)
{ my $rp=scalar @repolist; my $count=0;
print STDERR "\tFound $rp elements in array repolist.\n";
while ($count < $rp){ print STDERR "\t\t$count of $rp: \"$repolist[$count]\"\n"; $count
++;}
}

# Build the hash of repository UUID and friendly name
my %repos;
my $globalrepointex=0;
foreach $r (@repolist)
{ #each repo line should begin with 'id'
$r=~s/^ //g; $r=~s/ / /g;
if ("$r" =~ /^id/ )
{ if ($DEBUG>0){print STDERR "\tFound raw data repository name \"$r\" as index
$globalrepointex\n";}
my ($oneth,$twoth)=split(/ /, $r); my $discard;
($discard,$repos{$globalrepointex}{uuid})=split(/:/,$oneth);
($discard,$repos{$globalrepointex}{name})=split(/:/,$twoth);
$globalrepointex++;
}
else { if ($DEBUG>1){print STDERR "\tFound raw line \"$r\"\n"; } }
}

```

```

}

# Debug section on OpenSSH
if ($DEBUG>2)
{ &ssh_connect;
  my $checkconn=$ssh->capture("list tag");
  print STDERR "\tSSH connection is open if there is data: $checkconn.\n\n";
}

# Now let's get space info for each repo
my $rii=0; #RoleInfoIndex counter
until ($rii == $globalrepoindex)
{ if ($DEBUG>1){print STDERR "\tRetreiving detailed information on repo $repos{$rii}
{name} ($repos{$rii}{uuid}).\n"; }
  my $rilindex=0; # Repo Info Line Index
  &ssh_connect;
  #local $/="\r"; #?\n|\r";
  my $rrinfo=$ssh->pipe_out("show repository id=$repos{$rii}{uuid}");
  while (<$rrinfo>)
  {
    $_=~tr/\r\n//d; #remove carriage returns and linefeeds
    $_=~tr/ //s; #squeezes spaces
    $_=~s/^ //; # deletes leading space
    if ($DEBUG>1){print STDERR "\t\tRepo($rii) line($rilindex): \"$_\".\n"; }
    my($name,$data)=split(/ = /,$_);
    if ($DEBUG>1){print STDERR "\t\t\tSplit to name=\"$name\" and data=\"$data\".
\n"; }
    if ("$_" =~/^Used/) { $repos{$rii}{percentused}=$data; if ($data > $PCused)
{$sendalert++;} }
    if ("$_" =~/^File System Free/) { $repos{$rii}{fsfree}=$data; if ($data <
$FSfree){$sendalert++;} }
    if ("$_" =~/^File System Total/) { $repos{$rii}{fstotal}=$data; }
    if ("$_" =~/^File System Used/) { $repos{$rii}{fsused}=$data; }

    $rilindex++;
  }
  $rii++;
}

# Data is gathered, did we hit send alert function?
if ($sendalert >0)
{
  if ($DEBUG>1){print STDERR "\tAt least one item hit alert threshold, preparing to send
alert.\n"; }

  # Build the $report
  my $report="<h2> Oracle VM Repository Report</h2>\n<p>One or more thresholds was hit and
initiated this report.<p>\n";

  my $repind=0;
  until ($repind==$globalrepoindex)
  {
    $report=$report."<n<h3>$repos{$repind}{name}</h3>\n";
    $report=$report. tablify({ BORDER=>1, DATA=>$repos{$repind}, SORTBY=>'name',
ORDER=>'asc' } );
    $repind++;
  }

  if ($DEBUG>0){print STDERR "\n\tHTML Table:\n$report\n";}

  # send the $report
  if (($MAILTO ne "")&&($MAILHOST ne ""))

```

```
{
  my $msg = MIME::Lite->new(
    From    =>"$MAILFROM",
    To      =>"$MAILTO",
    Subject =>"OVM Repository Alert",
    Type    =>'multipart/related'
  );
  $msg->attach(
    Type => 'text/html',
    Data => qq{
      <body>
        $report
      </body>
    },
  );
  $msg->send('smtp', $MAILHOST, Debug=>$DEBUG);
}
}
```

```
#####
```

```
# Open the SSH connection
```

```
sub ssh_connect
```

```
{
  if ($DEBUG>1){print STDERR "\t\tConnecting to host \"$OVMHOST\".\n"; }
  $ssh = Net::OpenSSH->new("$OVMHOST"); #, async=>1, master_stderr_discard => 1
  $ssh->error and die "Couldn't establish SSH connection: ". $ssh->error;
}
```