ghettoVCB.sh - Free alternative for backing up VM's for ESX(i) 3.5, 4.x & 5.x

Version 81

Created by lamw on Nov 17, 2008 7:04 PM. Last modified by lamw on Jan 26, 2013 3:58 PM.

Table of Contents:

Description:

This script performs backups of virtual machines residing on **ESX(i) 3.5/4.x/5.x** servers using methodology similar to VMware's VCB tool. The script takes snapshots of live running virtual machines, backs up the master VMDK(s) and then upon completion, deletes the snapshot until the next backup. The only caveat is that it utilizes resources available to the Service Console of the ESX server or Busybox Console (Tech Support Mode) of the ESXi server running the backups as opposed to following the traditional method of offloading virtual machine backups through a VCB proxy.

This script has been tested on **ESX 3.5/4.x/5.x and ESXi 3.5/4.x/5.x** and supports the following backup mediums: **LOCAL STORAGE**, **SAN** and **NFS**. The script is non-interactive and can be setup to run via cron. Currently, this script accepts a text file that lists the display names of virtual machine(s) that are to be backed up. Additionally, one can specify a folder containing configuration files on a per VM basis for granular control over backup policies.

Additionally, for ESX(i) environments that don't have persistent NFS datastores designated for backups, the script offers the ability to automatically connect the ESX(i) server to a NFS exported folder and then upon backup completion, disconnect it from the ESX(i) server. The connection is established by creating an NFS datastore link which enables monolithic (or thick) VMDK backups as opposed to using the usual *nix mount command which necessitates breaking VMDK files into the 2gbsparse format for backup. Enabling this mode is self-explanatory and will evidently be so when editing the script (Note:**VM_BACKUP_VOLUME** variable is ignored if **ENABLE_NON_PERSISTENT_NFS=1**).

In its current configuration, the script will allow up to 3 unique backups of the Virtual Machine before it will overwrite the previous backups; this however, can be modified to fit procedures if need be. Please be diligent in running the script in a test or staging environment before using it on production live Virtual Machines; this script functions well within our environment but there is a chance that it may not fit well into other environments.

If you have any questions, you may post in the dedicated ghettoVCB VMTN community group.

If you have found this script to be useful and would like to contribute back, please click here to donate.

Please read ALL documentation + FAQ's before posting a question about an issue or problem. Thank You

Features

- Online back up of VM(s)
- Support for multiple VMDK disk(s) backup per VM
- Only valid VMDK(s) presented to the VM will be backed up
- Ability to shutdown guestOS and initiate backup process and power on VM afterwards with the option of hard power timeout
- Allow spaces in VM(s) backup list (not recommended and not a best practice)
- Ensure that snapshot removal process completes prior to to continuing onto the next VM backup
- VM(s) that intially contain **snapshots** will not be backed up and will be ignored
- Ability to specify the number of backup rotations for VM
- Output back up VMDK(s) in either **ZEROEDTHICK** (default behavior) or **2GB SPARSE** or **THIN** or**EAGERZEROEDTHICK** format
- Support for both SCSI and IDE disks
- Non-persistent NFS backup
- Fully support VMDK(s) stored across multiple datastores
- Ability to compress backups (**Experimental Support - Please refer to FAQ #25**)
- Ability to configure individual VM backup policies
- Ability to include/exclude specific VMDK(s) per VM (requires individual VM backup policy setup)
- Ability to configure logging output to file
- Independent disk awareness (will ignore VMDK)
- New timeout variables for shutdown and snapshot creations
- Ability to configure snapshots with both memory and/or quiesce options
- Ability to configure disk adapter format
- Additional debugging information including dry run execution
- Support for VMs with both virtual/physical RDM (pRDM will be ignored and not backed up)
- Support for global ghettoVCB configuration file
- Support for VM exclusion list
- Ability to backup all VMs residing on a specific host w/o specifying VM list
- Implemented simple locking mechenism to ensure only 1 instance of ghettoVCB is running per host
- Updated backup directory structure - rsync friendly
- Additional logging and final status output
- Logging of ghettoVCB PID (proces id)
- Email backup logs (**Experimental Suppport**)
- Rsync "Link" Support (**Experimental Suppport**)
- Enhanced "dryrun" details including configuration and/or VMDK(s) issues
- New storage debugging details pre/post backup
- Quick email status summary
- Updated ghettoVCB documentation
- ghettoVCB available via github
- **Support for ESXi 5.1** NEW!
- **Support for individual VM backup via command-line** NEW!
- **Support VM(s) with existing snapshots NEW!**
- **Support mulitple running instances of ghettoVCB** NEW! (**Experimental Suppport**)
- **Configure VM shutdown/startup order NEW!**
- **Support changing custom VM name during restore NEW!**

Requirements:

- VMs running on ESX(i) 3.5/4.x+/5.x
- SSH console access to ESX(i) host

Setup:

1) Download **ghettoVCB** from github by clicking on the ZIP button at the top and upload to either your ESX or ESXi system (use scp or WinSCP to transfer the file)

2) Extract the contents of the zip file (filename will vary):

```
# unzip ghettoVCB-master.zip

Archive:  ghettoVCB-master.zip
   creating: ghettoVCB-master/
  inflating: ghettoVCB-master/README
  inflating: ghettoVCB-master/ghettoVCB-restore.sh
  inflating: ghettoVCB-master/ghettoVCB-restore_vm_restore_configuration_template
  inflating: ghettoVCB-master/ghettoVCB-vm_backup_configuration_template
  inflating: ghettoVCB-master/ghettoVCB.conf
  inflating: ghettoVCB-master/ghettoVCB.sh
```

3) The script is now ready to be used and is located in a directory named **ghettoVCB-master**

```
# ls -l

-rw-r--r--    1 root     root          281 Jan  6 03:58 README
-rw-r--r--    1 root     root        16024 Jan  6 03:58 ghettoVCB-restore.sh
-rw-r--r--    1 root     root          309 Jan  6 03:58 ghettoVCB-restore_vm_restore_configura
-rw-r--r--    1 root     root          356 Jan  6 03:58 ghettoVCB-vm_backup_configuration_temp
-rw-r--r--    1 root     root          631 Jan  6 03:58 ghettoVCB.conf
-rw-r--r--    1 root     root        49375 Jan  6 03:58 ghettoVCB.sh
```

4) Before using the scripts, you will need to enable the execute permission on both **ghettoVCB.sh** and**ghettoVCB-restore.sh** by running the following:

```
chmod +x ghettoVCB.shchmod +x ghettoVCB-restore.sh
```

Configurations:

The following variables need to be defined within the script or in VM backup policy prior to execution.

Defining the backup datastore and folder in which the backups are stored (if folder does not exist, it will automatically be created):

```
VM_BACKUP_VOLUME=/vmfs/volumes/dlgCore-NFS-bigboi.VM-Backups/WILLIAM_BACKUPS
```

Defining the backup disk format (zeroedthick, eagerzeroedthick, thin, and 2gbsparse are available):

```
DISK_BACKUP_FORMAT=thin
```

Note: If you are using the 2gbsparse on an ESXi 5.1 host, backups may fail. Please download the latest version of the ghettoVCB script which automatically resolves this or take a look at this article for the details.

Defining the backup rotation per VM:

```
VM_BACKUP_ROTATION_COUNT=3
```

Defining whether the VM is powered down or not prior to backup (1 = enable, 0 = disable):

Note: VM(s) that are powered off will not require snapshoting

```
POWER_VM_DOWN_BEFORE_BACKUP=0
```

Defining whether the VM can be hard powered off when "POWER_VM_DOWN_BEFORE_BACKUP" is enabled and VM does not have VMware Tools installed

```
ENABLE_HARD_POWER_OFF=0
```

If "ENABLE_HARD_POWER_OFF" is enabled, then this defines the number of (60sec) iterations the script will before executing a hard power off when:

```
ITER_TO_WAIT_SHUTDOWN=3
```

The number (60sec) iterations the script will wait when powering off the VM and will give up and ignore the particular VM for backup:

```
POWER_DOWN_TIMEOUT=5
```

The number (60sec) iterations the script will wait when taking a snapshot of a VM and will give up and ignore the particular VM for backup:

Note: Default value should suffice

```
SNAPSHOT_TIMEOUT=15
```

Defining whether or not to enable compression (1 = enable, 0 = disable):

```
ENABLE_COMPRESSION=0
```

NOTE: With ESXi 3.x/4.x/5.x, there is a limitation of the maximum size of a VM for compression within the unsupported Busybox Console which should not affect backups running classic ESX 3.x,4.x or 5.x. On ESXi 3.x the largest supported VM is 4GB for compression and on ESXi 4.x the largest supported VM is 8GB. If you try to compress a larger VM, you may run into issues when trying to extract upon a restore.**PLEASE TEST THE RESTORE PROCESS BEFORE MOVING TO PRODUCTION SYSTEMS!**

Defining the adapter type for backed up VMDK (DEPERCATED - NO LONGER NEEDED):

```
ADAPTER_FORMAT=buslogic
```

Defining whether virtual machine memory is snapped and if quiescing is enabled (1 = enable, 0 = disable):

Note: By default both are disabled

```
VM_SNAPSHOT_MEMORY=0
VM_SNAPSHOT_QUIESCE=0
```

NOTE: VM_SNAPSHOT_MEMORY is only used to ensure when the snapshot is taken, it's memory contents are also captured. This is only relevant to the actual snapshot and it's not used in any shape/way/form in regards to the backup. All backups taken whether your VM is running or offline will result in an offline VM backup when you restore. This was originally added for debugging purposes and in generally should be left disabled

Defining VMDK(s) to backup from a particular VM either a list of vmdks or "all"

```
VMDK_FILES_TO_BACKUP="myvmdk.vmdk"
```

Defining whether or not VM(s) with existing snapshots can be backed up. This flag means it willCONSOLIDATE ALL EXISTING SNAPSHOTS for a VM prior to starting the backup (1 = yes, 0 = no):

```
ALLOW_VMS_WITH_SNAPSHOTS_TO_BE_BACKEDUP=0
```

Defining the order of which VM(s) should be shutdown first, especially if there is a dependency between multiple VM(s). This should be a comma seperate list of VM(s)

```
VM_SHUTDOWN_ORDER=vm1,vm2,vm3
```

Defining the order of VM(s) that should be started up first after backups have completed, especially if there is a dependency between multiple VM(s). This should be a comma seperate list of VM(s)

```
VM_STARTUP_ORDER=vm3,vm2,vm1
```

Defining NON-PERSISTENT NFS Backup Volume (1 = yes, 0 = no):

```
ENABLE_NON_PERSISTENT_NFS=0
```

NOTE: This is meant for environments that do not want a persisted connection to their NFS backup volume and allows the NFS volume to only be mounted during backups. The script expects the following 5 variables to be defined if this is to be used: UNMOUNT_NFS, NFS_SERVER, NFS_MOUNT, NFS_LOCAL_NAME and NFS_VM_BACKUP_DIR

Defining whether or not to unmount the NFS backup volume (1 = yes, 0 = no):

```
UNMOUNT_NFS=0
```

Defining the NFS server address (IP/hostname):

```
NFS_SERVER=172.51.0.192
```

Defining the NFS export path:

```
NFS_MOUNT=/upload
```

Defining the NFS datastore name:

```
NFS_LOCAL_NAME=backup
```

Defining the NFS backup directory for VMs:

```
NFS_VM_BACKUP_DIR=mybackups
```

NOTE: Only supported if you are running vSphere 4.1 and this feature is experimental. If you are having issues with sending mail, please take a look at **Email Backup Log** section

Defining whether or not to email backup logs (1 = yes, 0 = no):

```
EMAIL_LOG=1
```

Defining whether or not to email message will be deleted off the host whether it is successful in sending, this is used for debugging purposes. (1 = yes, 0 = no):

```
EMAIL_DEBUG=1
```

Defining email server:

```
EMAIL_SERVER=auroa.primp-industries.com
```

Defining email server port:

```
EMAIL_SERVER_PORT=25
```

Defining email delay interval (useful if you have slow SMTP server and would like to include a delay in netcat using -i param, default is 1second):

```
EMAIL_DELAY_INTERVAL=1
```

Defining recipient of the email:

```
EMAIL_TO=auroa@primp-industries.com
```

Defining from user which may require specific domain entry depending on email server configurations:

```
EMAIL_FROM=root@ghettoVCB
```

Defining to support RSYNC symbolic link creation (1 = yes, 0 = no):

```
RSYNC_LINK=0
```

Note: This enables an automatic creation of a generic symbolic link (both a relative & absolution path) in which users can refer to run replication backups using rsync from a remote host. This does not actually support rsync backups with ghettoVCB. Please take a look at the Rsync Section of the documentation for more details.

- A sample global ghettoVCB configuration file is included with the download called **ghettoVCB.conf**. It contains the same variables as defined from above and allows a user to customize and define multiple global configurations based on a user's environment.

```
# cat ghettoVCB.conf
VM_BACKUP_VOLUME=/vmfs/volumes/dlgCore-NFS-bigboi.VM-Backups/WILLIAM_BACKUPS
DISK_BACKUP_FORMAT=thin
VM_BACKUP_ROTATION_COUNT=3
POWER_VM_DOWN_BEFORE_BACKUP=0
ENABLE_HARD_POWER_OFF=0
ITER_TO_WAIT_SHUTDOWN=3
POWER_DOWN_TIMEOUT=5
ENABLE_COMPRESSION=0
VM_SNAPSHOT_MEMORY=0
VM_SNAPSHOT_QUIESCE=0
ALLOW_VMS_WITH_SNAPSHOTS_TO_BE_BACKEDUP=0
ENABLE_NON_PERSISTENT_NFS=0
UNMOUNT_NFS=0
NFS_SERVER=172.30.0.195
NFS_MOUNT=/nfsshare
NFS_LOCAL_NAME=nfs_storage_backup
NFS_VM_BACKUP_DIR=mybackups
SNAPSHOT_TIMEOUT=15
EMAIL_LOG=0
EMAIL_SERVER=auroa.primp-industries.com
EMAIL_SERVER_PORT=25
EMAIL_DELAY_INTERVAL=1
EMAIL_TO=auroa@primp-industries.com
EMAIL_FROM=root@ghettoVCB
WORKDIR_DEBUG=0
VM_SHUTDOWN_ORDER=
VM_STARTUP_ORDER=
```

To override any existing configurations within the ghettoVCB.sh script and to use a global configuration file, user just needs to specify the new flag -g and path to global configuration file (For an example, please refer to the sample execution section of the documenation)

Running multiple instances of ghettoVCB is now supported with the latest release by specifying the working directory (-w) flag.

By default, the working directory of the ghettoVCB instance is /tmp/ghettoVCB.work and you can run another instance by providing an alternate working directory. You should try to minimize the number of ghettoVCB instances running on your ESXi host as it does consume some amount of resources when running in the ESXi Shell. This is considered an experimental feature, so please test in a development environment to ensure everything is working prior to moving to production system.

Ensure that you do not edit past this section:

```
########################## DO NOT MODIFY PAST THIS LINE ##########################
```

Usage:

```
# ./ghettoVCB.sh
###############################################################################
#
# ghettoVCB for ESX/ESXi 3.5, 4.x+ and 5.x
# Author: William Lam
# http://www.virtuallyghetto.com/
# Documentation: http://communities.vmware.com/docs/DOC-8760
# Created: 11/17/2008
# Last modified: 2012_12_17 Version 0
#
###############################################################################

Usage: ghettoVCB.sh [options]

OPTIONS:
   -a     Backup all VMs on host
   -f     List of VMs to backup
   -m     Name of VM to backup (overrides -f)
   -c     VM configuration directory for VM backups
   -g     Path to global ghettoVCB configuration file
   -l     File to output logging
   -w     ghettoVCB work directory (default: )
   -d     Debug level [info|debug|dryrun] (default: info)

(e.g.)

Backup VMs stored in a list
    ./ghettoVCB.sh -f vms_to_backup

Backup a single VM
    ./ghettoVCB.sh -m vm_to_backup

Backup all VMs residing on this host
    ./ghettoVCB.sh -a

Backup all VMs residing on this host except for the VMs in the exclusion list
    ./ghettoVCB.sh -a -e vm_exclusion_list

Backup VMs based on specific configuration located in directory
    ./ghettoVCB.sh -f vms_to_backup -c vm_backup_configs

Backup VMs using global ghettoVCB configuration file
    ./ghettoVCB.sh -f vms_to_backup -g /global/ghettoVCB.conf

Output will log to /tmp/ghettoVCB.log (consider logging to local or remote datastore to persist
    ./ghettoVCB.sh -f vms_to_backup -l /vmfs/volume/local-storage/ghettoVCB.log

Dry run (no backup will take place)
    ./ghettoVCB.sh -f vms_to_backup -d dryrun
```

The input to this script is a file that contains the display name of the virtual machine(s) separated by a newline. When creating this file on a non-Linux/UNIX system, you may introduce ^M character which can cause the script to miss-behave. To ensure this does not occur, plesae create the file on the ESX/ESXi host.

Here is a sample of what the file would look like:

```
[root@himalaya ~]# cat vms_to_backup
vCOPS
vMA
vCloudConnector
```

Sample Execution:

- Dry run Mode
- Debug Mode
- Backup VMs stored in a list
- Backup Single VM using command-line
- Backup All VMs residing on specific ESX(i) host
- Backup VMs based on individual VM backup policies

Dry run Mode (no backup will take place)

Note: This execution mode provides a qucik summary of details on whether a given set of VM(s)/VMDK(s) will be backed up. It provides additional information such as VMs that may have snapshots, VMDK(s) that are configured as independent disks, or other issues that may cause a VM or VMDK to not backed up.

- Log verbosity: dryrun
- Log output: stdout & /tmp (default)
  - Logs by default will be stored in /tmp, these log files may not persist through reboots, especially when dealing with ESXi. You should log to either a local or remote datastore to ensure that logs are kept upon a reboot.

```
[root@himalaya ghettoVCB]# ./ghettoVCB.sh -f vms_to_backup -d dryrun
Logging output to "/tmp/ghettoVCB-2011-03-13_15-19-57.log" ...
2011-03-13 15:19:57 -- info: ============================== ghettoVCB LOG START ===============

2011-03-13 15:19:57 -- info: CONFIG - VERSION = 2011_03_13_1
2011-03-13 15:19:57 -- info: CONFIG - GHETTOVCB_PID = 30157
2011-03-13 15:19:57 -- info: CONFIG - VM_BACKUP_VOLUME = /vmfs/volumes/dlgCore-NFS-bigboi.VM-Ba
2011-03-13 15:19:57 -- info: CONFIG - VM_BACKUP_ROTATION_COUNT = 3
2011-03-13 15:19:57 -- info: CONFIG - VM_BACKUP_DIR_NAMING_CONVENTION = 2011-03-13_15-19-57
2011-03-13 15:19:57 -- info: CONFIG - DISK_BACKUP_FORMAT = thin
2011-03-13 15:19:57 -- info: CONFIG - POWER_VM_DOWN_BEFORE_BACKUP = 0
2011-03-13 15:19:57 -- info: CONFIG - ENABLE_HARD_POWER_OFF = 0
2011-03-13 15:19:57 -- info: CONFIG - ITER_TO_WAIT_SHUTDOWN = 3
2011-03-13 15:19:57 -- info: CONFIG - POWER_DOWN_TIMEOUT = 5
2011-03-13 15:19:57 -- info: CONFIG - SNAPSHOT_TIMEOUT = 15
2011-03-13 15:19:57 -- info: CONFIG - LOG_LEVEL = dryrun
2011-03-13 15:19:57 -- info: CONFIG - BACKUP_LOG_OUTPUT = /tmp/ghettoVCB-2011-03-13_15-19-57.lo
2011-03-13 15:19:57 -- info: CONFIG - VM_SNAPSHOT_MEMORY = 0
2011-03-13 15:19:57 -- info: CONFIG - VM_SNAPSHOT_QUIESCE = 0
2011-03-13 15:19:57 -- info: CONFIG - VMDK_FILES_TO_BACKUP = all
2011-03-13 15:19:57 -- info: CONFIG - EMAIL_LOG = 0
2011-03-13 15:19:57 -- info:
2011-03-13 15:19:57 -- dryrun: ###############################################
2011-03-13 15:19:57 -- dryrun: Virtual Machine: scofield
2011-03-13 15:19:57 -- dryrun: VM_ID: 704
2011-03-13 15:19:57 -- dryrun: VMX_PATH: /vmfs/volumes/himalaya-local-SATA.RE4-GP:Storage/scofi
2011-03-13 15:19:57 -- dryrun: VMX_DIR: /vmfs/volumes/himalaya-local-SATA.RE4-GP:Storage/scofie
2011-03-13 15:19:57 -- dryrun: VMX_CONF: scofield/scofield.vmx
```

```
2011-03-13 15:19:57 -- dryrun: VMFS_VOLUME: himalaya-local-SATA.RE4-GP:Storage
2011-03-13 15:19:57 -- dryrun: VMDK(s):
2011-03-13 15:19:58 -- dryrun:  scofield_3.vmdk 3 GB
2011-03-13 15:19:58 -- dryrun:  scofield_2.vmdk 2 GB
2011-03-13 15:19:58 -- dryrun:  scofield_1.vmdk 1 GB
2011-03-13 15:19:58 -- dryrun:  scofield.vmdk   5 GB
2011-03-13 15:19:58 -- dryrun: INDEPENDENT VMDK(s):
2011-03-13 15:19:58 -- dryrun: TOTAL_VM_SIZE_TO_BACKUP: 11 GB
2011-03-13 15:19:58 -- dryrun: ###############################################

2011-03-13 15:19:58 -- dryrun: ###############################################
2011-03-13 15:19:58 -- dryrun: Virtual Machine: vMA
2011-03-13 15:19:58 -- dryrun: VM_ID: 1440
2011-03-13 15:19:58 -- dryrun: VMX_PATH: /vmfs/volumes/himalaya-local-SATA.RE4-GP:Storage/vMA/v
2011-03-13 15:19:58 -- dryrun: VMX_DIR: /vmfs/volumes/himalaya-local-SATA.RE4-GP:Storage/vMA
2011-03-13 15:19:58 -- dryrun: VMX_CONF: vMA/vMA.vmx
2011-03-13 15:19:58 -- dryrun: VMFS_VOLUME: himalaya-local-SATA.RE4-GP:Storage
2011-03-13 15:19:58 -- dryrun: VMDK(s):
2011-03-13 15:19:58 -- dryrun:  vMA-000002.vmdk 5 GB
2011-03-13 15:19:58 -- dryrun: INDEPENDENT VMDK(s):
2011-03-13 15:19:58 -- dryrun: TOTAL_VM_SIZE_TO_BACKUP: 5 GB
2011-03-13 15:19:58 -- dryrun: Snapshots found for this VM, please commit all snapshots before
2011-03-13 15:19:58 -- dryrun: THIS VIRTUAL MACHINE WILL NOT BE BACKED UP DUE TO EXISTING SNAPS
2011-03-13 15:19:58 -- dryrun: ###############################################

2011-03-13 15:19:58 -- dryrun: ###############################################
2011-03-13 15:19:58 -- dryrun: Virtual Machine: vCloudConnector
2011-03-13 15:19:58 -- dryrun: VM_ID: 2064
2011-03-13 15:19:58 -- dryrun: VMX_PATH: /vmfs/volumes/himalaya-local-SATA.RE4-GP:Storage/vClou
2011-03-13 15:19:58 -- dryrun: VMX_DIR: /vmfs/volumes/himalaya-local-SATA.RE4-GP:Storage/vCloud
2011-03-13 15:19:58 -- dryrun: VMX_CONF: vCloudConnector/vCloudConnector.vmx
2011-03-13 15:19:58 -- dryrun: VMFS_VOLUME: himalaya-local-SATA.RE4-GP:Storage
2011-03-13 15:19:58 -- dryrun: VMDK(s):
2011-03-13 15:19:59 -- dryrun:  vCloudConnector.vmdk    3 GB
2011-03-13 15:19:59 -- dryrun: INDEPENDENT VMDK(s):
2011-03-13 15:19:59 -- dryrun:  vCloudConnector_1.vmdk  40 GB
2011-03-13 15:19:59 -- dryrun: TOTAL_VM_SIZE_TO_BACKUP: 3 GB
2011-03-13 15:19:59 -- dryrun: Snapshots can not be taken for indepdenent disks!
2011-03-13 15:19:59 -- dryrun: THIS VIRTUAL MACHINE WILL NOT HAVE ALL ITS VMDKS BACKED UP!
2011-03-13 15:19:59 -- dryrun: ###############################################

2011-03-13 15:19:59 -- info: ###### Final status: OK, only a dryrun. ######

2011-03-13 15:19:59 -- info: ============================== ghettoVCB LOG END =================
```

In the example above, we have 3 VMs to be backed up:

- scofield has 4 VMDK(s) that total up to 11GB and does not contain any snapshots/independent disks and this VM should backup without any issues
- vMA has 1 VMDK but it also contains a snapshot and clearly this VM will not be backed up until the snapshot has been committed
- vCloudConnector has 2 VMDK(s), one which is 3GB and another which is 40GB and configured as an independent disk. Since snapshots do not affect independent disk, only the 3GB VMDK will be backed up for this VM as denoted by the "**TOTAL_VM_SIZE_TO_BACKUP**"

Debug backup mode

Note: This execution modes provides more in-depth information about environment/backup process including additional storage debugging information which provides information about both the source/destination datastore pre and post backups. This can be very useful in troubleshooting backups

- Log verbosity: debug
- Log output: stdout & /tmp (default)
  - Logs by default will be stored in /tmp, these log files may not persist through reboots, especially when dealing with ESXi. You should log to either a local or remote datastore to ensure that logs are kept upon a reboot.

```
[root@himalaya ghettoVCB]# ./ghettoVCB.sh -f vms_to_backup -d debug
Logging output to "/tmp/ghettoVCB-2011-03-13_15-27-59.log" ...
2011-03-13 15:27:59 -- info: ============================== ghettoVCB LOG START ==============

2011-03-13 15:27:59 -- debug: Succesfully acquired lock directory - /tmp/ghettoVCB.lock

2011-03-13 15:27:59 -- debug: HOST VERSION: VMware ESX 4.1.0 build-260247
2011-03-13 15:27:59 -- debug: HOST LEVEL: VMware ESX 4.1.0 GA
2011-03-13 15:27:59 -- debug: HOSTNAME: himalaya.primp-industries.com

2011-03-13 15:27:59 -- info: CONFIG - VERSION = 2011_03_13_1
2011-03-13 15:27:59 -- info: CONFIG - GHETTOVCB_PID = 31074
2011-03-13 15:27:59 -- info: CONFIG - VM_BACKUP_VOLUME = /vmfs/volumes/dlgCore-NFS-bigboi.VM-Ba
2011-03-13 15:27:59 -- info: CONFIG - VM_BACKUP_ROTATION_COUNT = 3
2011-03-13 15:27:59 -- info: CONFIG - VM_BACKUP_DIR_NAMING_CONVENTION = 2011-03-13_15-27-59
2011-03-13 15:27:59 -- info: CONFIG - DISK_BACKUP_FORMAT = thin
2011-03-13 15:27:59 -- info: CONFIG - POWER_VM_DOWN_BEFORE_BACKUP = 0
2011-03-13 15:27:59 -- info: CONFIG - ENABLE_HARD_POWER_OFF = 0
2011-03-13 15:27:59 -- info: CONFIG - ITER_TO_WAIT_SHUTDOWN = 3
2011-03-13 15:27:59 -- info: CONFIG - POWER_DOWN_TIMEOUT = 5
2011-03-13 15:27:59 -- info: CONFIG - SNAPSHOT_TIMEOUT = 15
2011-03-13 15:27:59 -- info: CONFIG - LOG_LEVEL = debug
2011-03-13 15:27:59 -- info: CONFIG - BACKUP_LOG_OUTPUT = /tmp/ghettoVCB-2011-03-13_15-27-59.lo
2011-03-13 15:27:59 -- info: CONFIG - VM_SNAPSHOT_MEMORY = 0
2011-03-13 15:27:59 -- info: CONFIG - VM_SNAPSHOT_QUIESCE = 0
2011-03-13 15:27:59 -- info: CONFIG - VMDK_FILES_TO_BACKUP = all
2011-03-13 15:27:59 -- info: CONFIG - EMAIL_LOG = 0
2011-03-13 15:27:59 -- info:
2011-03-13 15:28:01 -- debug: Storage Information before backup:
2011-03-13 15:28:01 -- debug: SRC_DATASTORE: himalaya-local-SATA.RE4-GP:Storage
2011-03-13 15:28:01 -- debug: SRC_DATASTORE_CAPACITY: 1830.5 GB
2011-03-13 15:28:01 -- debug: SRC_DATASTORE_FREE: 539.4 GB
2011-03-13 15:28:01 -- debug: SRC_DATASTORE_BLOCKSIZE: 4
2011-03-13 15:28:01 -- debug: SRC_DATASTORE_MAX_FILE_SIZE: 1024 GB
2011-03-13 15:28:01 -- debug:
2011-03-13 15:28:01 -- debug: DST_DATASTORE: dlgCore-NFS-bigboi.VM-Backups
2011-03-13 15:28:01 -- debug: DST_DATASTORE_CAPACITY: 1348.4 GB
2011-03-13 15:28:01 -- debug: DST_DATASTORE_FREE: 296.8 GB
2011-03-13 15:28:01 -- debug: DST_DATASTORE_BLOCKSIZE: NA
2011-03-13 15:28:01 -- debug: DST_DATASTORE_MAX_FILE_SIZE: NA
2011-03-13 15:28:01 -- debug:
2011-03-13 15:28:02 -- info: Initiate backup for scofield
2011-03-13 15:28:02 -- debug: /usr/sbin/vmkfstools -i "/vmfs/volumes/himalaya-local-SATA.RE4-GP
Destination disk format: VMFS thin-provisioned
Cloning disk '/vmfs/volumes/himalaya-local-SATA.RE4-GP:Storage/scofield/scofield_3.vmdk'...
Clone: 37% done.
2011-03-13 15:28:04 -- debug: /usr/sbin/vmkfstools -i "/vmfs/volumes/himalaya-local-SATA.RE4-GP
Destination disk format: VMFS thin-provisioned
Cloning disk '/vmfs/volumes/himalaya-local-SATA.RE4-GP:Storage/scofield/scofield_2.vmdk'...
Clone: 85% done.
2011-03-13 15:28:05 -- debug: /usr/sbin/vmkfstools -i "/vmfs/volumes/himalaya-local-SATA.RE4-GP

2011-03-13 15:28:06 -- debug: /usr/sbin/vmkfstools -i "/vmfs/volumes/himalaya-local-SATA.RE4-GP
Destination disk format: VMFS thin-provisioned
Cloning disk '/vmfs/volumes/himalaya-local-SATA.RE4-GP:Storage/scofield/scofield.vmdk'...
Clone: 78% done.
2011-03-13 15:29:52 -- info: Backup Duration: 1.83 Minutes
2011-03-13 15:29:52 -- info: Successfully completed backup for scofield!

2011-03-13 15:29:54 -- debug: Storage Information after backup:
2011-03-13 15:29:54 -- debug: SRC_DATASTORE: himalaya-local-SATA.RE4-GP:Storage
2011-03-13 15:29:54 -- debug: SRC_DATASTORE_CAPACITY: 1830.5 GB
2011-03-13 15:29:54 -- debug: SRC_DATASTORE_FREE: 539.4 GB
2011-03-13 15:29:54 -- debug: SRC_DATASTORE_BLOCKSIZE: 4
2011-03-13 15:29:54 -- debug: SRC_DATASTORE_MAX_FILE_SIZE: 1024 GB
2011-03-13 15:29:54 -- debug:
```

```
2011-03-13 15:29:54 -- debug: DST_DATASTORE: dlgCore-NFS-bigboi.VM-Backups
2011-03-13 15:29:54 -- debug: DST_DATASTORE_CAPACITY: 1348.4 GB
2011-03-13 15:29:54 -- debug: DST_DATASTORE_FREE: 296.8 GB
2011-03-13 15:29:54 -- debug: DST_DATASTORE_BLOCKSIZE: NA
2011-03-13 15:29:54 -- debug: DST_DATASTORE_MAX_FILE_SIZE: NA
2011-03-13 15:29:54 -- debug:
2011-03-13 15:29:55 -- debug: Storage Information before backup:
2011-03-13 15:29:55 -- debug: SRC_DATASTORE: himalaya-local-SATA.RE4-GP:Storage
2011-03-13 15:29:55 -- debug: SRC_DATASTORE_CAPACITY: 1830.5 GB
2011-03-13 15:29:55 -- debug: SRC_DATASTORE_FREE: 539.4 GB
2011-03-13 15:29:55 -- debug: SRC_DATASTORE_BLOCKSIZE: 4
2011-03-13 15:29:55 -- debug: SRC_DATASTORE_MAX_FILE_SIZE: 1024 GB
2011-03-13 15:29:55 -- debug:
2011-03-13 15:29:55 -- debug: DST_DATASTORE: dlgCore-NFS-bigboi.VM-Backups
2011-03-13 15:29:55 -- debug: DST_DATASTORE_CAPACITY: 1348.4 GB
2011-03-13 15:29:55 -- debug: DST_DATASTORE_FREE: 296.8 GB
2011-03-13 15:29:55 -- debug: DST_DATASTORE_BLOCKSIZE: NA
2011-03-13 15:29:55 -- debug: DST_DATASTORE_MAX_FILE_SIZE: NA
2011-03-13 15:29:55 -- debug:
2011-03-13 15:29:55 -- info: Snapshot found for vMA, backup will not take place

2011-03-13 15:29:57 -- debug: Storage Information before backup:
2011-03-13 15:29:57 -- debug: SRC_DATASTORE: himalaya-local-SATA.RE4-GP:Storage
2011-03-13 15:29:57 -- debug: SRC_DATASTORE_CAPACITY: 1830.5 GB
2011-03-13 15:29:57 -- debug: SRC_DATASTORE_FREE: 539.4 GB
2011-03-13 15:29:57 -- debug: SRC_DATASTORE_BLOCKSIZE: 4
2011-03-13 15:29:57 -- debug: SRC_DATASTORE_MAX_FILE_SIZE: 1024 GB
2011-03-13 15:29:57 -- debug:
2011-03-13 15:29:57 -- debug: DST_DATASTORE: dlgCore-NFS-bigboi.VM-Backups
2011-03-13 15:29:57 -- debug: DST_DATASTORE_CAPACITY: 1348.4 GB
2011-03-13 15:29:57 -- debug: DST_DATASTORE_FREE: 296.8 GB
2011-03-13 15:29:57 -- debug: DST_DATASTORE_BLOCKSIZE: NA
2011-03-13 15:29:57 -- debug: DST_DATASTORE_MAX_FILE_SIZE: NA
2011-03-13 15:29:57 -- debug:
2011-03-13 15:29:58 -- info: Initiate backup for vCloudConnector
2011-03-13 15:29:58 -- debug: /usr/sbin/vmkfstools -i "/vmfs/volumes/himalaya-local-SATA.RE4-GP
Destination disk format: VMFS thin-provisioned
Cloning disk '/vmfs/volumes/himalaya-local-SATA.RE4-GP:Storage/vCloudConnector/vCloudConnector.
Clone: 97% done.
2011-03-13 15:30:45 -- info: Backup Duration: 47 Seconds
2011-03-13 15:30:45 -- info: WARN: vCloudConnector has some Independent VMDKs that can not be b

2011-03-13 15:30:45 -- info: ###### Final status: ERROR: Only some of the VMs backed up, and so

2011-03-13 15:30:45 -- debug: Succesfully removed lock directory - /tmp/ghettoVCB.lock

2011-03-13 15:30:45 -- info: ============================== ghettoVCB LOG END ================
```

Backup VMs stored in a list

```
[root@himalaya ~]# ./ghettoVCB.sh -f vms_to_backup
```

Backup Single VM using command-line

```
# ./ghettoVCB.sh -m MyVM
```

Backup All VMs residing on specific ESX(i) host

```
/ghettoVCB # ./ghettoVCB.sh -a
```

Backup All VMs residing on specific ESX(i) host and exclude the VMs in the exclusion list

```
/ghettoVCB # ./ghettoVCB.sh -a -e vm_exclusion_list
```

Backup VMs based on individual VM backup policies and log output to /tmp/ghettoVCB.log

- Log verbosity: info (default)
- Log output: /tmp/ghettoVCB.log
    - Logs by default will be stored in /tmp, these log files may not persist through reboots, especially when dealing with ESXi. You should log to either a local or remote datastore to ensure that logs are kept upon a reboot.

1. Create folder to hold individual VM backup policies (can be named anything):

```
[root@himalaya ~]# mkdir backup_config
```

2. Create individual VM backup policies for each VM that ensure each file is named exactly as the display name of the VM being backed up (use provided template to create duplicates):

```
[root@himalaya backup_config]# cp ghettoVCB-vm_backup_configuration_template scofield
[root@himalaya backup_config]# cp ghettoVCB-vm_backup_configuration_template vCloudConnector
```

Listing of VM backup policy within backup configuration directory

```
[root@himalaya backup_config]# ls
ghettoVCB-vm_backup_configuration_template  scofield  vCloudConnector
```

Backup policy for "scofield" (backup only 2 specific VMDKs)

```
[root@himalaya backup_config]# cat scofield
VM_BACKUP_VOLUME=/vmfs/volumes/dlgCore-NFS-bigboi.VM-Backups/WILLIAM_BACKUPS
DISK_BACKUP_FORMAT=thin
VM_BACKUP_ROTATION_COUNT=3
POWER_VM_DOWN_BEFORE_BACKUP=0
ENABLE_HARD_POWER_OFF=0
ITER_TO_WAIT_SHUTDOWN=4
POWER_DOWN_TIMEOUT=5
SNAPSHOT_TIMEOUT=15
ENABLE_COMPRESSION=0
VM_SNAPSHOT_MEMORY=0
VM_SNAPSHOT_QUIESCE=0
VMDK_FILES_TO_BACKUP="scofield_2.vmdk,scofield_1.vmdk"
```

Backup policy for VM "vCloudConnector" (backup all VMDKs found)

```
[root@himalaya backup_config]# cat vCloudConnector
VM_BACKUP_VOLUME=/vmfs/volumes/dlgCore-NFS-bigboi.VM-Backups/WILLIAM_BACKUPS
DISK_BACKUP_FORMAT=thin
VM_BACKUP_ROTATION_COUNT=3
POWER_VM_DOWN_BEFORE_BACKUP=0
```

```
ENABLE_HARD_POWER_OFF=0
ITER_TO_WAIT_SHUTDOWN=4
POWER_DOWN_TIMEOUT=5
SNAPSHOT_TIMEOUT=15
ENABLE_COMPRESSION=0
VM_SNAPSHOT_MEMORY=0
VM_SNAPSHOT_QUIESCE=0
VMDK_FILES_TO_BACKUP="vCloudConnector.vmdk"
```

Note: When specifying -c option (individual VM backup policy mode) if a VM is listed in the backup list but **DOES NOT** have a corresponding backup policy, the VM will be backed up using the default configuration found within the ghettoVCB.sh script.

Execution of backup

```
[root@himalaya ~]# ./ghettoVCB.sh -f vms_to_backup -c backup_config -l /tmp/ghettoVCB.log

2011-03-13 15:40:50 -- info: ============================== ghettoVCB LOG START ===============

2011-03-13 15:40:51 -- info: CONFIG - USING CONFIGURATION FILE = backup_config//scofield
2011-03-13 15:40:51 -- info: CONFIG - VERSION = 2011_03_13_1
2011-03-13 15:40:51 -- info: CONFIG - GHETTOVCB_PID = 2967
2011-03-13 15:40:51 -- info: CONFIG - VM_BACKUP_VOLUME = /vmfs/volumes/dlgCore-NFS-bigboi.VM-Ba
2011-03-13 15:40:51 -- info: CONFIG - VM_BACKUP_ROTATION_COUNT = 3
2011-03-13 15:40:51 -- info: CONFIG - VM_BACKUP_DIR_NAMING_CONVENTION = 2011-03-13_15-40-50
2011-03-13 15:40:51 -- info: CONFIG - DISK_BACKUP_FORMAT = thin
2011-03-13 15:40:51 -- info: CONFIG - POWER_VM_DOWN_BEFORE_BACKUP = 0
2011-03-13 15:40:51 -- info: CONFIG - ENABLE_HARD_POWER_OFF = 0
2011-03-13 15:40:51 -- info: CONFIG - ITER_TO_WAIT_SHUTDOWN = 4
2011-03-13 15:40:51 -- info: CONFIG - POWER_DOWN_TIMEOUT = 5
2011-03-13 15:40:51 -- info: CONFIG - SNAPSHOT_TIMEOUT = 15
2011-03-13 15:40:51 -- info: CONFIG - LOG_LEVEL = info
2011-03-13 15:40:51 -- info: CONFIG - BACKUP_LOG_OUTPUT = /tmp/ghettoVCB.log
2011-03-13 15:40:51 -- info: CONFIG - VM_SNAPSHOT_MEMORY = 0
2011-03-13 15:40:51 -- info: CONFIG - VM_SNAPSHOT_QUIESCE = 0
2011-03-13 15:40:51 -- info: CONFIG - VMDK_FILES_TO_BACKUP = scofield_2.vmdk,scofield_1.vmdk
2011-03-13 15:40:51 -- info: CONFIG - EMAIL_LOG = 0
2011-03-13 15:40:51 -- info:
2011-03-13 15:40:53 -- info: Initiate backup for scofield
Destination disk format: VMFS thin-provisioned
Cloning disk '/vmfs/volumes/himalaya-local-SATA.RE4-GP:Storage/scofield/scofield_2.vmdk'...
Clone: 100% done.

Destination disk format: VMFS thin-provisioned
Cloning disk '/vmfs/volumes/himalaya-local-SATA.RE4-GP:Storage/scofield/scofield_1.vmdk'...
Clone: 100% done.

2011-03-13 15:40:55 -- info: Backup Duration: 2 Seconds
2011-03-13 15:40:55 -- info: Successfully completed backup for scofield!

2011-03-13 15:40:57 -- info: CONFIG - VERSION = 2011_03_13_1
2011-03-13 15:40:57 -- info: CONFIG - GHETTOVCB_PID = 2967
2011-03-13 15:40:57 -- info: CONFIG - VM_BACKUP_VOLUME = /vmfs/volumes/dlgCore-NFS-bigboi.VM-Ba
2011-03-13 15:40:57 -- info: CONFIG - VM_BACKUP_ROTATION_COUNT = 3
2011-03-13 15:40:57 -- info: CONFIG - VM_BACKUP_DIR_NAMING_CONVENTION = 2011-03-13_15-40-50
2011-03-13 15:40:57 -- info: CONFIG - DISK_BACKUP_FORMAT = thin
2011-03-13 15:40:57 -- info: CONFIG - POWER_VM_DOWN_BEFORE_BACKUP = 0
2011-03-13 15:40:57 -- info: CONFIG - ENABLE_HARD_POWER_OFF = 0
2011-03-13 15:40:57 -- info: CONFIG - ITER_TO_WAIT_SHUTDOWN = 3
2011-03-13 15:40:57 -- info: CONFIG - POWER_DOWN_TIMEOUT = 5
2011-03-13 15:40:57 -- info: CONFIG - SNAPSHOT_TIMEOUT = 15
2011-03-13 15:40:57 -- info: CONFIG - LOG_LEVEL = info
```

```
2011-03-13 15:40:57 -- info: CONFIG - BACKUP_LOG_OUTPUT = /tmp/ghettoVCB.log
2011-03-13 15:40:57 -- info: CONFIG - VM_SNAPSHOT_MEMORY = 0
2011-03-13 15:40:57 -- info: CONFIG - VM_SNAPSHOT_QUIESCE = 0
2011-03-13 15:40:57 -- info: CONFIG - VMDK_FILES_TO_BACKUP = all
2011-03-13 15:40:57 -- info: CONFIG - EMAIL_LOG = 0
2011-03-13 15:40:57 -- info:
2011-03-13 15:40:59 -- info: Snapshot found for vMA, backup will not take place

2011-03-13 15:40:59 -- info: CONFIG - USING CONFIGURATION FILE = backup_config//vCloudConnector
2011-03-13 15:40:59 -- info: CONFIG - VERSION = 2011_03_13_1
2011-03-13 15:40:59 -- info: CONFIG - GHETTOVCB_PID = 2967
2011-03-13 15:40:59 -- info: CONFIG - VM_BACKUP_VOLUME = /vmfs/volumes/dlgCore-NFS-bigboi.VM-Ba
2011-03-13 15:40:59 -- info: CONFIG - VM_BACKUP_ROTATION_COUNT = 3
2011-03-13 15:40:59 -- info: CONFIG - VM_BACKUP_DIR_NAMING_CONVENTION = 2011-03-13_15-40-50
2011-03-13 15:40:59 -- info: CONFIG - DISK_BACKUP_FORMAT = thin
2011-03-13 15:40:59 -- info: CONFIG - POWER_VM_DOWN_BEFORE_BACKUP = 0
2011-03-13 15:40:59 -- info: CONFIG - ENABLE_HARD_POWER_OFF = 0
2011-03-13 15:40:59 -- info: CONFIG - ITER_TO_WAIT_SHUTDOWN = 4
2011-03-13 15:40:59 -- info: CONFIG - POWER_DOWN_TIMEOUT = 5
2011-03-13 15:40:59 -- info: CONFIG - SNAPSHOT_TIMEOUT = 15
2011-03-13 15:40:59 -- info: CONFIG - LOG_LEVEL = info
2011-03-13 15:40:59 -- info: CONFIG - BACKUP_LOG_OUTPUT = /tmp/ghettoVCB.log
2011-03-13 15:40:59 -- info: CONFIG - VM_SNAPSHOT_MEMORY = 0
2011-03-13 15:40:59 -- info: CONFIG - VM_SNAPSHOT_QUIESCE = 0
2011-03-13 15:40:59 -- info: CONFIG - VMDK_FILES_TO_BACKUP = vCloudConnector.vmdk
2011-03-13 15:40:59 -- info: CONFIG - EMAIL_LOG = 0
2011-03-13 15:40:59 -- info:
2011-03-13 15:41:01 -- info: Initiate backup for vCloudConnector
Destination disk format: VMFS thin-provisioned
Cloning disk '/vmfs/volumes/himalaya-local-SATA.RE4-GP:Storage/vCloudConnector/vCloudConnector.
Clone: 100% done.

2011-03-13 15:41:51 -- info: Backup Duration: 50 Seconds
2011-03-13 15:41:51 -- info: WARN: vCloudConnector has some Independent VMDKs that can not be b

2011-03-13 15:41:51 -- info: ###### Final status: ERROR: Only some of the VMs backed up, and so

2011-03-13 15:41:51 -- info: ============================== ghettoVCB LOG END ================
```

Enable compression for backups (EXPERIMENTAL SUPPORT)

Please take a look at **FAQ #25** for more details before continuing

To make use of this feature, modify the variable **ENABLE_COMPRESSION** from 0 to 1. Please note, do not mix uncompressed backups with compressed backups. Ensure that directories selected for backups do not contain any backups with previous versions of ghettoVCB before enabling and implementing the compressed backups feature.

Email Backup Logs (EXPERIMENTAL SUPPORT)

nc (netcat) utility must be present for email support to function, this utility is a now a default with the release of vSphere 4.1 or greater, previous releases of VI 3.5 and/or vSphere 4.0 does not contain this utility. The reason this is listed as experimental is it may not be compatible with all email servers as the script utlizes **nc** (netcat) utility to communicate to an email server. This feature is provided as-is with no guarantees. If you enable this feature, a separate log will be generated along side any normal logging which will be used to email recipient. If for whatever reason, the email fails to send, an entry will appear per the normal logging mechanism.

Users should also make note due to limited functionality of netcat, it uses SMTP pipelining which is not the most ideal method of communicating with an SMTP server. Email from ghettoVCB may not work if your email server does not support this feature.

You can define an email recipient in the following two ways:

```
EMAIL_TO=william@virtuallyghetto.com
```

OR

```
EMAIL_TO=william@virtuallyghetto.com,tuan@virtuallyghetto.com
```

If you are running ESXi 5.1, you will need to create a custom firewall rule to allow your email traffic to go out which I will assume is default port 25. Here are the steps for creating a custom email rule.

Step 1 - Create a file called **/etc/vmware/firewall/email.xml** with contains the following:

```
<ConfigRoot>
  <service>
    <id>email</id>
    <rule id="0000">
      <direction>outbound</direction>
      <protocol>tcp</protocol>
      <porttype>dst</porttype>
      <port>25</port>
    </rule>
    <enabled>true</enabled>
    <required>false</required>
  </service>
</ConfigRoot>
```

Step 2 - Reload the ESXi firewall by running the following ESXCLI command:

```
~ #
```

```
esxcli network firewall refresh
```

Step 3 - Confirm that your email rule has been loaded by running the following ESXCLI command:

```
~ # esxcli network firewall ruleset list | grep email
email                  true
```

Step 4 - Connect to your email server by usingn nc (netcat) by running the following command and specifying the IP Address/Port of your email server:

```
~ # nc 172.30.0.107 25
220 mail.primp-industries.com ESMTP Postfix
```

You should recieve a response from your email server and you can enter Ctrl+C to exit. This custom ESXi firewall rule will not persist after a reboot, so you should create a custom VIB to ensure it persists after a system reboot. Please take a look at this article for the details.

Rsync Support (EXPERIMENTAL SUPPORT)

To make use of this feature, modify the variable **RSYNC_LINK** from 0 to 1. Please note, this is an experimental feature request from users that rely on rsync to replicate changes from one datastore volume to another datastore volume. The premise of this feature is to have a standardized folder that rsync can monitor for changes to replicate to another backup datastore. When this feature is enabled, a symbolic link will be generated with the format of "<VMNAME>-symlink" and will reference the latest successful VM backup. You can then rely on this symbolic link to watch for changes and replicate to your backup datastore.

Here is an example of what this would look like:

```
[root@himalaya ghettoVCB]# ls -la /vmfs/volumes/dlgCore-NFS-bigboi.VM-Backups/WILLIAM_BACKUPS/v
total 0
drwxr-xr-x 1 nobody nobody 110 Sep 27 08:08 .
drwxr-xr-x 1 nobody nobody  17 Sep 16 14:01 ..
lrwxrwxrwx 1 nobody nobody  89 Sep 27 08:08 vcma-symlink -> /vmfs/volumes/dlgCore-NFS-bigboi.VM
drwxr-xr-x 1 nobody nobody  58 Sep 27 08:04 vcma-2010-09-27_08-04-26
drwxr-xr-x 1 nobody nobody  58 Sep 27 08:06 vcma-2010-09-27_08-05-55
drwxr-xr-x 1 nobody nobody  58 Sep 27 08:08 vcma-2010-09-27_08-07-37
```

FYI - This feature has not been tested, please provide feedback if this does not work as expected.

Restore backups (ghettoVCB-restore.sh):

To recover a VM that has been processed by ghettoVCB, please take a look at this document: Ghetto Tech Preview - ghettoVCB-restore.sh - Restoring VM's backed up from ghettoVCB to ESX(i) 3.5, 4.x, and 5.x

Stopping ghettoVCB Process:

There may be a situation where you need to stop the ghettoVCB process and entering Ctrl+C will only kill off the main ghettoVCB process, however there may still be other spawn processes that you may need to identify and stop. Below are two scenarios you may encounter and the process to completely stop all processes related to ghettoVCB.

Interactively running ghettoVCB:

Step 1 - Press Ctrl+C which will kill off the main ghettoVCB instance

Step 2 - Search for any existing ghettoVCB process by running the following:

```
# ps -c | grep ghettoVCB | grep -v grep
3360136 3360136 tail                  tail -f /tmp/ghettoVCB.work/ghettovcb.Cs1M1x
```

Step 3 - Here we can see there is a tail command that was used in the script. We need to stop this process by using the kill command which accepts the PID (Process ID) which is identified by the first value on the far left hand side of the command. In this example, it is 3360136.

```
# kill -9 3360136
```

Note: Make sure you identify the correct PID, else you could accidently impact a running VM or worse your ESXi host.

Step 4 - Depending on where you stopped the ghettoVCB process, you may need to consolidate or remove any existing snapshots that may exist on the VM that was being backed up. You can easily do so by using the vSphere Client.

Non-Interactively running ghettoVCB:

Step 1 - Search for the ghettoVCB process (you can also validate the PID from the logs)

```
~ # ps -c | grep ghettoVCB | grep -v grep
3360393 3360393 busybox              ash ./ghettoVCB.sh -f list -d debug
3360790 3360790 tail                 tail -f /tmp/ghettoVCB.work/ghettovcb.deGeB7
```

Step 2 - Stop both the main ghettoVCB instance & tail command by using the kill command and specifying their respective PID IDs:

```
kill -9 3360393
kill -9 3360790
```

Step 3 - If a VM was in the process of being backed up, there is an additional process for the actual vmkfstools copy. You will need to identify the process for that and kill that as well. We will again use ps -c command and search for any vmkfstools that are running:

```
# ps -c | grep vmkfstools | grep -v grep
3360796 3360796 vmkfstools           /sbin/vmkfstools -i /vmfs/volumes/himalaya-temporary/VC-Wi
```

Step 4 - In case there is someone manually running a vmkfstools, make sure you take a look at the command itself and that it maps back to the current VM that was being backed up before kill the process. Once you have identified the proper PID, go ahead and use the kill command:

```
# kill -9 3360796
```

Step 5 - Depending on where you stopped the ghettoVCB process, you may need to consolidate or remove any existing snapshots that may exist on the VM that was being backed up. You can easily do so by using the vSphere Client.

Cronjob FAQ:

Please take a moment to read over what is a cronjob and how to set one up, before continuing

The task of configuring cronjobs on classic ESX servers (with Service Console) is no different than traditional cronjobs on *nix operating systems (this procedure is outlined in the link above). With ESXi on the other hand, additional factors need to be taken into account when setting up cronjobs in the limited shell console called Busybox because changes made do not persist through a system reboot. The following document will outline steps to ensure that cronjob configurations are saved and present upon a reboot.

Important Note: Always redirect the ghettoVCB output to /dev/null and/or to a log when automating via cron, this becomes very important as one user has identified a limited amount of buffer capacity in which once filled, may cause ghettoVCB to stop in the middle of a backup. This primarily only affects users on ESXi, but it is good practice to always redirect the output. Also ensure you are specifying the FULL PATH when referencing the ghettoVCB script, input or log files.

e.g.

```
0 0 * * 1-5 /vmfs/volumes/dlgCore-NFS-bigboi.VM-Backups/ghettoVCB.sh -f /vmfs/volumes/dlgCore-N
```

or

```
0 0 * * 1-5 /vmfs/volumes/dlgCore-NFS-bigboi.VM-Backups/ghettoVCB.sh -f /vmfs/volumes/dlgCore-N
```

Task: Configure ghettoVCB.sh to execute a backup five days a week (M-F) at 12AM (midnight) everyday and send output to a unique log file

Configure on ESX:

1. As root, you'll install your cronjob by issuing:

```
[root@himalaya ~]# crontab -e
```

2. Append the following entry:

```
0 0 * * 1-5 /vmfs/volumes/dlgCore-NFS-bigboi.VM-Backups/ghettoVCB.sh -f /vmfs/volumes/dlgCore-N
```

3. Save and exit

```
[root@himalaya dlgCore-NFS-bigboi.VM-Backups]# crontab -e
no crontab for root - using an empty one
crontab: installing new crontab
```

4. List out and verify the cronjob that was just created:

```
[root@himalaya dlgCore-NFS-bigboi.VM-Backups]# crontab -l
0 0 * * 1-5 /vmfs/volumes/dlgCore-NFS-bigboi.VM-Backups/ghettoVCB.sh -f /vmfs/volumes/dlgCore-N
```

You're ready to go!

Configure on ESXi:

1. Setup the cronjob by appending the following line to /var/spool/cron/crontabs/root:

```
0 0 * * 1-5 /vmfs/volumes/simplejack-local-storage/ghettoVCB.sh -f /vmfs/volumes/simplejack-loc
```

If you are unable to edit/modify /var/spool/cron/crontabs/root, please make a copy and then edit the copy with the changes

```
cp /var/spool/cron/crontabs/root /var/spool/cron/crontabs/root.backup
```

Once your changes have been made, then "mv" the backup to the original file. This may occur on ESXi 4.x or 5.x hosts

```
mv /var/spool/cron/crontabs/root.backup /var/spool/cron/crontabs/root
```

You can now verify the crontab entry has been updated by using "cat" utility.

2. Kill the current crond (cron daemon) and then restart the crond for the changes to take affect:

On ESXi < 3.5u3

```
kill $(ps | grep crond | cut -f 1 -d ' ')
```

On ESXi 3.5u3+

```
~ # kill $(pidof crond)
~ # crond
```

On ESXi 4.x/5.0

```
~ # kill $(cat /var/run/crond.pid)
~ # busybox crond
```

On ESXi 5.1

```
~ # kill $(cat /var/run/crond.pid)
~ # crond
```

3. Now that the cronjob is ready to go, you need to ensure that this cronjob will persist through a reboot. You'll need to add the following two lines to **/etc/rc.local** (ensure that the cron entry matches what was defined above). In ESXi 5.1, you will need to edit **/etc/rc.local.d/local.sh** instead of /etc/rc.local as that is no longer valid.

On ESXi 3.5

```
/bin/kill $(pidof crond)
/bin/echo "0 0 * * 1-5 /vmfs/volumes/simplejack-local-storage/ghettoVCB.sh -f /vmfs/volumes/sim
crond
```

On ESXi 4.x/5.0

```
/bin/kill $(cat /var/run/crond.pid)
/bin/echo "0 0 * * 1-5 /vmfs/volumes/simplejack-local-storage/ghettoVCB.sh -f /vmfs/volumes/sim
/bin/busybox crond
```

On ESXi 5.1

```
/bin/kill $(cat /var/run/crond.pid)
/bin/echo "0 0 * * 1-5 /vmfs/volumes/simplejack-local-storage/ghettoVCB.sh -f /vmfs/volumes/sim
crond
```

Afterwards the file should look like the following:

```
~ # cat /etc/rc.local
#! /bin/ash
export PATH=/sbin:/bin

log() {
   echo "$1"
   logger init "$1"
}

#execute all service retgistered in /etc/rc.local.d
if [http:// -d /etc/rc.local.d |http:// -d /etc/rc.local.d ]; then
   for filename in `find /etc/rc.local.d/ | sort`
      do
         if [ -f $filename ] && [ -x $filename ]; then
            log "running $filename"
            $filename
         fi
      done
fi

/bin/kill $(cat /var/run/crond.pid)
/bin/echo "0 0 * * 1-5 /vmfs/volumes/simplejack-local-storage/ghettoVCB.sh -f /vmfs/volumes/sim
/bin/busybox crond
```

This will ensure that the cronjob is re-created upon a reboot of the system through a startup script

2. To ensure that this is saved in the ESXi configuration, we need to manually initiate an ESXi backup by running:

```
~ # /sbin/auto-backup.sh
config implicitly loaded
local.tgz
etc/vmware/vmkiscsid/vmkiscsid.db
etc/dropbear/dropbear_dss_host_key
etc/dropbear/dropbear_rsa_host_key
etc/opt/vmware/vpxa/vpxa.cfg
etc/opt/vmware/vpxa/dasConfig.xml
etc/sysconfig/network
etc/vmware/hostd/authorization.xml
etc/vmware/hostd/hostsvc.xml
etc/vmware/hostd/pools.xml
etc/vmware/hostd/vmAutoStart.xml
etc/vmware/hostd/vmInventory.xml
etc/vmware/hostd/proxy.xml
etc/vmware/ssl/rui.crt
etc/vmware/ssl/rui.key
etc/vmware/vmkiscsid/initiatorname.iscsi
etc/vmware/vmkiscsid/iscsid.conf
etc/vmware/vmware.lic
etc/vmware/config
etc/vmware/dvsdata.db
```

```
etc/vmware/esx.conf
etc/vmware/license.cfg
etc/vmware/locker.conf
etc/vmware/snmp.xml
etc/group
etc/hosts
etc/inetd.conf
etc/rc.local
etc/chkconfig.db
etc/ntp.conf
etc/passwd
etc/random-seed
etc/resolv.conf
etc/shadow
etc/sfcb/repository/root/interop/cim_indicationfilter.idx
etc/sfcb/repository/root/interop/cim_indicationhandlercimxml.idx
etc/sfcb/repository/root/interop/cim_listenerdestinationcimxml.idx
etc/sfcb/repository/root/interop/cim_indicationsubscription.idx
Binary files /etc/vmware/dvsdata.db and /tmp/auto-backup.31345.dir/etc/vmware/dvsdata.db differ
config implicitly loaded
Saving current state in /bootbank
Clock updated.
Time: 20:40:36   Date: 08/14/2009   UTC
```

Now you're really done!
If you're still having trouble getting the cronjob to work, ensure that you've specified the correct parameters and there aren’t any typos in any part of the syntax.

Ensure crond (cron daemon) is running:

ESX 3.x/4.0:

```
[root@himalaya dlgCore-NFS-bigboi.VM-Backups]# ps -ef | grep crond | grep -v grep
root      2625     1  0 Aug13 ?        00:00:00 crond
```

ESXi 3.x/4.x/5.x:

```
~ # ps | grep crond | grep -v grep
5196 5196 busybox              crond
```

Ensure that the date/time on your ESX(i) host is setup correctly:

ESX(i):

```
[root@himalaya dlgCore-NFS-bigboi.VM-Backups]# date
Fri Aug 14 23:44:47 PDT 2009
```

Note: Careful attention must be noted if more than one backup is performed per day. Backup windows should be staggered to avoid contention or saturation of resources during these periods.

FAQ:

0Q: I'm getting error X when using the script or I'm not getting any errors, the backup didn't even take place. What can I do?
0A: First off, before posting a comment/question, please thoroughly read through the **ENTIRE**documentation including the FAQs to see if your question has already been ansered.

1Q: I've read through the entire documentation + FAQs and still have not found my answer to the problem I'm seeing. What can I do?
1A: Please join the ghettoVCB Group to post your question/comment.

2Q: I've sent you private message or email but I haven't received a response? What gives?
2A: I do not accept issues/bugs reported via PM or email, I will reply back, directing you to post on the appropriate VMTN forum (that's what it's for). If the data/results you're providing is truely senstive to your environment I will hear you out, but 99.99% it is not, so please do not messsage/email me directly. I do monitor all forums that contain my script including the normal VMTN forums and will try to get back to your question as soon as I can and as time permits. Please do be patient as you're not the only person using the script (600,000+ views), thank you.

3Q: Can I schedule backups to take place hourly, daily, monthly, yearly?
3A: Yes, do a search online for **crontab**.

4Q: I would like to setup cronjob for ESX(i) 3.5 or 4.0?
4A: Take a look at the Cronjob FAQ section in this document.

5Q: I want to schedule my backup on Windows, how do I do this?
5A: Do a search for **plink**. Make sure you have paired SSH keys setup between your Windows system and ESX/ESXi host.

6Q: I only have a single ESXi host. I want to take backups and store them somewhere else. The problem is: I don't have NFS, iSCSI nor FC SAN. What can I do?
6A: You can use local storage to store your backups assuming that you have enough space on the destination datastore. Afterwards, you can use scp (WinSCP/FastSCP) to transfer the backups from the ESXi host to your local desktop.

7Q: I'm pissed; the backup is taking too long. My datastore is of type X?
7A: YMMV, take a look at your storage configuration and make sure it is optimized.

8Q: I noticed that the backup rotation is occurring after a backup. I don't have enough local storage space, can the process be changed?
8A: This is primarily done to ensure that you have at least one good backup in case the new backup fails. If you would like to modify the script, you're more than welcome to do so.

9Q: What is the best storage configuration for datastore type X?
9A: Search the VMTN forums; there are various configurations for the different type of storage/etc.

10Q: I want to setup an NFS server to run my backups. Which is the best and should it be virtual or physical?
10A: Please refer to answer 7A. From experience, we've seen physical instances of NFS servers to be faster than their virtual counterparts. As always, YMMV.

11Q: I have VMs that have snapshots. I want to back these things up but the script doesn't let me do it. How do I fix that?
11A: VM snapshots are not meant to be kept for long durations. When backing up a VM that contains a snapshot, you should ensure all snapshots have been committed prior to running a backup. No exceptions will be made…ever.

12Q: I would like to restore from backup, what is the best method?
12A: The restore process will be unique for each environment and should be determined by your backup/recovery plans. At a high level you have the option of mounting the backup datastore and registering the VM in question or copy the VM from the backup datastore to the ESX/ESXi host. The latter is recommended so that you're not running a VM living on the backup datastore or inadvertently modifying your backup VM(s). You can also take a look at ghettoVCB-restore which is experimentally supported.

13Q: When I try to run the script I get: **"-bash: ./ghettoVCB.sh: Permission denied"**, what is wrong?
13A: You need to change the permission on the script to be executable, chmod +x ghettoVCB.sh

14Q: Where can I download the latest version of the script?
14A: The latest version is available on on github - https://github.com/lamw/ghettoVCB/downloads

15Q: I would like to suggest/recommend feature X, can I get it? When can I get it? Why isn't it here, what gives?
15A: The general purpose of this script is to provide a backup solution around VMware VMs. Any additional features outside of that process will be taken into consideration depending on the amount of time, number of requests and actual usefulness as a whole to the community rather than to an individual.

16Q: I have found this script to be very useful and would like to contribute back, what can I do?
16A: To continue to develop and share new scripts and resources with the community, we need your support. You can donate here Thank You!

17Q: What are the different type of backup uses cases that are supported with ghettoVCB?
17A: 1) Live backup of VM with the use of a snapshot and 2) Offline backup of a VM without a snapshot. These are the only two use cases supported by the script.

18Q: When I execute the script on ESX(i) I get some funky errors such as ": not found.sh" or "command not found". What is this?
18A: Most likely you have some ^M characters within the script which may have come from either editing the script using Windows editor, uploading the script using the datastore browser OR using wget. The best option is to either using WinSCP on Windows to upload the script and edit using vi editor on ESX(i) host OR Linux/UNIX scp to copy the script into the host. If you still continue to have the issue, do a search online on various methods of removing this Windows return carriage from the script

19Q: My backup works fine OR it works for a single backup but I get an error message "Input/output error" or "-ash: YYYY-MM-DD: not found" during the snapshot removal process. What is this?
19A: The issue has been recently identified by few users as a problem with user's NFS server in which it reports an error when deleting large files that take longer than 10seconds. VMware has recently released a KB article http://kb.vmware.com/kb/1035332 explaining the details and starting with **vSphere 4.1 Update 2** or **vSphere 5.0**, a new advanced ESX(i) parameter has been introduced to increase the timeout. This has resolved the problem for several users and maybe something to consider if you are running into this issue, specifically with NFS based backups.

20Q: Will this script function with vCenter and DRS enabled?
20Q: No, if the ESX(i) hosts are in a DRS enabled cluster, VMs that are to be backed up could potentially be backed up twice or never get backed up. The script is executed on a per host basis and one would need to come up a way of tracking backups on all hosts and perhaps write out to external file to ensure that all VMs are backed up. The main use case for this script are for standalone ESX(i) host

21Q: I'm trying to use WinSCP to manually copy VM files but it's very slow or never completes on huge files, why is that?
21A: WinSCP was not designed for copying VM files out of your ESX(i) host, take a look at Veeam's FastSCP which is designed for moving VM files and is a free utility.

22Q: Can I use setup NFS Server using Windows Services for UNIX (WSFU) and will it work?
22A: I've only heard a handful of users that have successfully implemented WSFU and got it working, YMMV. VMware also has a KB article decribing the setup process here:http://kb.vmware.com/kb/1004490 for those that are interested. Here is a thread on a user's experience between Windows Vs. Linux NFS that maybe helpful.

23Q: How do VMware Snapshots work?
23A: http://kb.vmware.com/kb/1015180

24Q: What files make up a Virtual Machine?
24A: http://virtualisedreality.wordpress.com/2009/09/16/quick-reminder-of-what-files-make-up-a-virtual-machine/

25Q: I'm having some issues restoring a compressed VM backup?
25A: There is a limitation in the size of the VM for compression under ESXi 3.x & 4.x, this limitation is in the unsupported Busybox console and should not affect classic ESX 3.x/4.x. On ESXi 3.x, the maximum largest supported VM is 4GB for compression and on ESXi 4.x the largest supported VM is 8GB. If you try to compress a larger VM, you may run into issues when trying to extract upon a restore. **PLEASE TEST THE RESTORE PROCESS BEFORE MOVING TO PRODUCTION SYSTEMS!**

26Q: I'm backing up my VM as "thin" format but I'm still not noticing any size reduction in the backup? What gives?
2bA: Please refer to this blog post which explains what's going on: http://www.yellow-bricks.com/2009/07/31/storage-vmotion-and-moving-to-a-thin-provisioned-disk/

27Q: I've enabled **VM_SNAPSHOT_MEMORY** and when I restore my VM it's still offline, I thought this would keep it's memory state?
27A: VM_SNAPSHOT_MEMORY is only used to ensure when the snapshot is taken, it's memory contents are also captured. This is only relavent to the actual snapshot itself and it's not used in any shape/way/form in regards to the backup. All backups taken whether your VM is running or offline will result in an offline VM backup when you restore. This was originally added for debugging purposes and in generally should be left disabled

28Q: Can I rename the directories and the VMs after a VM has been backed up?
28A: The answer yes, you can ... but you may run into all sorts of issues which may break the backup process. The script expects a certain layout and specific naming scheme for it to maintain the proper rotation count. If you need to move or rename a VM, please take it out of the directory and place it in another location

29Q: Can ghettoVCB support CBT (Change Block Tracking)?
29A: No, that is a functionality of the vSphere API + VDDK API (vSphere Disk Development Kit). You will need to look at paid solutions such as VMware vDR, Veeam Backup & Recovery, PHD Virtual Backups, etc. to leverage that functionailty.

30Q: Does ghettoVCB support rsync backups?
30A: Currently ghettoVCB does not support rsync backups, you either obtain or compile your own static rsync binary and run on ESXi, but this is an unsupported configuration. You may take a look at this blog post for some details.

31Q: How can I contribute back?
31A: You can provide feedback/comments on the ghettoVCB Group. If you have found this script to be useful and would like to contribute back, please click here to donate.

32Q: How can select individual VMDKs to backup from a VM?
32A: Ideally you would use the "**-c**" option which requires you to create individual VM configuration file, this is where you would select specific VMDKs to backup. Note, that if you do not need to define all properties, anything not defined will adhere from the default global properties whether you're editing the ghettoVCB.sh script or using ghettoVCB global configuration file. It is not recommended that you edit the ghettoVCB.sh script and modify the **VMDK_FILES_TO_BACKUP** variable, but if you would like to keep everything in one script, you may add the extensive list of VMDKs to backup but do know this can get error prone as script may be edited frequently and lose some flexibility to support multiple environments.

33Q: Why is email not working when I'm using ESXi 5.x but it worked in ESXi 4.x?
33A: ESXi 5.x has implemented a new firewall which requires the email port that is being used to be opened. Please refer to the following articles on creating a custom firewall rule for email:
http://www.virtuallyghetto.com/2012/09/creating-custom-vibs-for-esxi-50-51.html
How to Create Custom Firewall Rules in ESXi 50
How to Persist Configuration Changes in ESXi 4.x/5.x Part 1
How to Persist Configuration Changes in ESXi 4.x/5.x Part 2

34Q: How do I stop the ghettoVCB process?
34A: Take a look at the **Stopping ghettoVCB Process** section of the documentation for more details.

Our NFS Server Configuration

Many have asked what is the best configuration and recommendation for setting up a cheap NFS Server to run backups for VMs. This has been a question we've tried to stay away from just because the possiblities and solutions are endless. One can go with physical vs. virtual, use VSA (Virtual Storage Appliances) such as OpenFiler or Lefthand Networks, Windows vs. Linux/UNIX. We've not personally tested and verify all these solutions and it all comes down to "it depends" type of answer. Though from our experience, we've had much better success with a physical server than a virtual.

It is also well known that some users are experiencing backup issues when running specifically against NFS, primarily around the rotation and purging of previous backups. The theory from what we can tell by talking to various users is that when the rotation is occuring, the request to delete the file(s) may take awhile and does not return within a certain time frame and causes the script to error out with unexpected messages. Though the backups were successful, it will cause unexpected results with directory structures on the NFS target. We've not been able to isolate why this is occuring and maybe due to NFS configuration/exports or hardware or connection not being able to support this process.

We'll continue to help where we can in diagonising this issus but we wanted to share our current NFS configuration, perhaps it may help some users who are new or trying to setup their system. (**Disclaimer:**These configurations are not recommendations nor endorsement for any of the components being used)

UPDATE: Please also read FAQ #19 for details + resolution

Server Type: Physical
Model: HP DL320 G2
OS: Arch linux 2.6.28
Disks: 2 x 1.5TB
RAID: Software RAID1
Source Host Backups: ESX 3.5u4 and ESX 4.0u1 (We don't run any ESXi hosts)

uname -a output

```
Linux XXXXX.XXXXX.ucsb.edu 2.6.28-ARCH #1 SMP PREEMPT Sun Jan 18 20:17:17 UTC 2009 i686 Intel(R
```

NICs:

```
00:05.0 Ethernet controller: Broadcom Corporation NetXtreme BCM5702X Gigabit Ethernet (rev 02)
00:06.0 Ethernet controller: Broadcom Corporation NetXtreme BCM5702X Gigabit Ethernet (rev 02)
```

NFS Export Options:

```
/exports/vm-backups XXX.XXX.XXX.XXX/24(rw,async,all_squash,anonuid=99,anongid=99)
```

*One important thing to check is to verify that your NFS exportion options are setup correctly, **"async"**should be configured to ensure that all IO requests are processed and reply back to the client before waiting for the data to be written to the storage.

*Recently VMware released a KB article describing the various "Advanced NFS Options" and their meanings and recommendations: http://kb.vmware.com/kb/1007909 We've not personally had to touch any of these, but for other vendors such as EMC and NetApp, there are some best practices around configuring some of these values depending on the number of NFS volumes or number of ESX(i) host connecting to a volume. You may want to take a look to see if any of these options may help with NFS issue that some are seeing

*Users should also try to look at their ESX(i) host logs during the time interval when they're noticing these issues and see if they can find any correlation along with monitoring the performance on their NFS Server.

*Lastly, there are probably other things that can be done to improve NFS performance or further optimization, a simple search online will also yield many resources.

Useful Links:

Windows utility to email ghettoVCB Backup Logs - http://www.waldrondigital.com/2010/05/11/ghettovcb-e-mail-rotate-logs-batch-file-for-vmware/
Windows front-end utility to ghettoVCB - http://www.magikmon.com/mkbackup/ghettovcb.en.html

Note: Neither of these tools are supported, for questions or comments regarding these utilities please refer to the author's pages.

Change log:

01/13/13 -

Enhancements:

- ghettoVCB & ghettoVCB-restore supports ESXi 5.1
- Support for individual VM backup via command-line and added new -m flag
- Support VM(s) with existing snapshots and added new configuration variable called**ALLOW_VMS_WITH_SNAPSHOTS_TO_BE_BACKEDUP**
- Support multiple running instances of ghettoVCB running and added a new -w flag
- Configure VM shutdown/startup order and added two new configuration variables called**VM_SHUTDOWN_ORDER** and **VM_STARTUP_ORDER**
- Support changing custom VM name during restore
- Documentation updates

Fixes:

- Fixed tab/indentation for both ghettoVCB/ghettoVCB-restore
- Temp email files and email headers
- Fixed "whoami" command as it is no longer valid in ESXi 5.1 to check for proper user
- Added 2gbsparse check in sanity method to auto-load VMkernel module
- Various typos, for greater detail, you can refer to the "diff" in github repo

----

11/19/11 -

Enhancements:

- ghettoVCB & ghettoVCB-restore is now packaged together and both scripts are versioned on github
- ESXi 5 firewall check for email port (Check **FAQ #33** for more details)
- New **EMAIL_DELAY_INTERVAL** netcat variable to control slow SMTP servers
- ADAPTER_TYPE (buslogic,lsilogic,ide) no longer need to manually specified, script will auto-detect based on VMDK descriptor file
- Using symlink -f parameter for quicker unlink/re-link for RSYNC use case
- Updated documentation, including NFS issues (Check **FAQ #19** for more details including new VMware KB article)

Fixes:

- vSphere 4.1 Update 2 introduced new vim-cmd snapshot.remove param, this has now been updated in script to detect this new param change

----

06/28/11 -

Enhancements:

- Support for vSphere 5.0 - ESXi 5.0

05/22/11 -

Enhancements:

- Support for multiple email recipients
- Support for individual VMDK backup within ghettoVCB.sh script - **FAQ #33**

Fixes:

- Minor fix in additional validation prior to VM rotation

03/14/11 -

Enhancements:

- Enhanced "dryrun" details including configuration and/or VMDK(s) issues
  - Warning messages about physical RDM and Independent VMDK(s)
  - Warning messages about VMs with snapshots
- New storage debugging details
  - Datastore details both pre and post backups
  - Datstore blocksize miss-match warnings
- Quick email status summary is now included in the title of the email, this allows a user to quickly verify whether a backup was successful or had complete/partial failure without having to go through the logs.
- Updated ghettoVCB documentation
- ghettoVCB going forward will now be version tracked via github and previous releases will not be available for download

Fixes:

- Updated absolute sym link path for RSYNC_LINK variable to relative path
- Enhanced logging and details on warning/error messages

Big thanks to Alain Spineux and his contributions to the ghettoVCB script and helping with debugging and testing.

09/28/10 -

Enhancements:

- Additional email support for Microsoft IIS and email debugging functionality (Experimental Support)
- ghettoVCB PID is now captured in the logs
- Rsync support, please take a look at the above documentation for **Rsync Support** (Experimental Support)

Fixes:

- Fixed a few typos in the script
- Trapping SIG 13

07/27/10 -

Enhancements:

- Support for emailing backup logs (Experimental Support)

07/20/10 -

Enhancements:

- Support for vSphere 4.1 (ESX and ESXi)
- Additional logging information for debugging purposes

05/12/10 -

Enhancements:

- Thanks to user **Rodder** who submitted a patch for a workaround to handle the NFS I/O issue. The script will check to see if the return code of the "rm" operation for VMs that are to be rotated. If the return code has not returned right away, we may be running into the NFS I/O issue, the script will not sleep and check perodically to see if NFS volume is responsive and then continue to the next VM for backup.

Fixes:

- Resolved the problem when trying to specify ghettoVCB global configuration file with the fullpath

---

05/11/10 -

- Updated useful links to 2 utilties that were written by users for ghettoVCB

05/05/10 -

Fixes:

- Resolved an issue where VMs with spaces were not being properly rotated. Thanks to user **chrb** for finding the bug

---

04/24/10 -

Enhancements:

- Added the ability to include an exclusion list of VMs to not backup

Fixes:

- Resolved persistent NFS configuration bug due to the addition of the global ghettoVCB conf

---

04/23/10 -

Fixes:

- Resolved a bug in the VM naming directory which may not delete backups properly

04/20/10 -

- Support for global ghettoVCB configuration file. Users no longer need to edit main script and can use multiple configuration files based on certain environment configurations
- Ability to backup all VMs residing on a specific host w/o specifying VM list
- Implemented simple locking mechenism to ensure only 1 instance of ghettoVCB is running per host
- Updated backup directory structure - rsync friendly. All backup VM directories will now have the format of "VMNAME-YYYY-MM-DD_HH_MM_SS" which will not change once the backup has been completed. The script will keep N-copies and purge older backups based on the configurations set by the user.
- Additional logging and final status output has been added to the script to provide more useful error/warning messages and an additoinal status will be printed out at the end of each backup to provide an overall report

Big thanks goes out to the community for the suggested features and to those that submitted snippet of their modifications.

03/27/10 -

- Updated FAQ #0-1 & #25-29 for common issues/questions.
- For those experiencing NFS issue, please take a look at FAQ #29
- Re-packaged ghettoVCB.sh script within a tarball (ghettoVCB.tar.gz) to help assist those users having the "Windows affect" when trying to execute the script

02/13/10 -

Updated FAQ #20-24 for common issues/questions. Also included a new section about our "personal" NFS configuration and setup.

01/31/10 -

Fix the crontab section to reflect the correct syntax + updated FAQ #17,#18 and #19 for common issues.

11/17/09 -

The following enhancements and fixes have been implemented in this release of ghettoVCB. Special thanks goes out to all the ghettoVCB BETA testers for providing time and their environments to test features/fixes of the new script!

Enhancements:

- Individual VM backup policy
- Include/exclude specific VMDK(s)
- Logging to file
- Timeout variables
- Configur snapshot memory/quiesce
- Adapter format
- Additional logging + dryrun mode
- Support for both physical/virtual RDMs

Fixes:

- Independent disk awareE