Red Hat Enterprise Linux 5.x iSCSI

and Device Mapper Multipath HOWTO

Marshall University
Systems Infrastructure Team

Abstract

This document outlines in a detailed step-by-step fashion, how to properly configure iSCSI initiator, and multi-path I/O software on Red Hat Enterprise Linux version 5.

Author: Eric G. Wolfe
Contributor: Jaymz Mynes
Editor: Tim Calvert

Marshall University

400 Hal Greer Blvd.

Huntington, WV 25755

I. Introduction

The following configuration was documented and tested on a Dell Poweredge R710, with RedHat Enterprise Linux 5.0 x64. The R710 server has 4 Broadcom NeteXtreme II BCM5709 Gigabit Network Interface Cards (NIC), and is connected by Ethernet to an EqualLogic PS6000 storage array.

This step-by-step guide is directly applicable to these software and hardware components. However, much of the software configuration would remain the same for any Linux distribution on either X86 or AMD64 based hardware. The most dynamic variable in any case will be specific storage solutions. Consider any factors that your storage vendor may support, or recommend, before proceeding.

II. Hardware Preparation

Appropriate hardware preparation is beyond the scope of this document. See vendor documentation for connecting Ethernet cables to the storage array, and appropriate configuration of the storage array. According to Dell EqualLogic (2008) some recommended guidelines are as follows:

- Do...
  - Use Cat 6 or Cat 5E TSB95 cables.
  - Use redundant switches and network paths.
  - Use Flow Control on switches and NICs.
  - Use Jumbo Frames on switches and NICs
  - Use VLANs, if physically separate switches are not available.
- Do NOT...
  - Use Spanning Tree Protocol on switch ports for iSCSI connections.
  - Use Unicast storm control on a switch that handles iSCSI traffic.

See *PS Series Array Network Performance Guidelines*, or relevant storage vendor's documentation for more information.

III. Manual Installation

The following section outlines manual installation and configuration of Device Mapper Multipath and Open-iSCSI initiator on Red Hat Enterprise Linux version 5.x. If this is your first time installing and configuring Device Mapper Multipath, or Open-iSCSI software, then this is the recommended method of installation and configuration. It is possible to automate a portion of this installation and configuration with Red Hat's Kickstart, or configuration management engines such as Puppet. However, that is beyond the scope of this document.

The methods outlined in this guide apply directly to Red Hat Enterprise Linux, though a major portion of the configuration items are generic and could be applied to other Linux distributions. The author of this HOWTO assumes these steps will, for the most part, remain constant for subsequent versions of Red Hat Enterprise Linux.

1. Package Installation

After performing a base install of Red Hat Enterprise v5, make sure to upgrade the system to the latest patch-level.

Change this line in */etc/yum.conf* as illustrated below.
exclude=kernel*

Comment out the exclude options, so the kernel may be upgraded.
exclude=#kernel*

Run the following to upgrade to the latest patch-level.
yum -y upgrade

Run the following to install needed software.
yum -y install iscsi-initiator-utils device-mapper-multipath

Dell Equallogic (2009) recommends using RHEL 5.4 (version 5 update 4) or newer for best performance and interoperability with PS Series storage arrays. Ensure the following requirements are met before proceeding.

Run the following to check for minimum package versions.
rpm -qa iscsi-initiator-utils device-mapper-multipath
device-mapper-multipath-<version>-<patch-level>.el5
iscsi-initiator-utils-<version>-<patch-level>.el5

- Minimum package version for PS Series arrays
  - **iscsi-initiator-utils-6.2.0.742-0.6.el5**
- Minimum package version for GbE NICs
  - **iscsi-initiator-utils-6.2.0.868-0.7.el5**
- With SELinux enabled, you may not be able to login to iSCSI targets. You may have to create a policy for iSCSI traffic, or disable SELinux.

After updating the kernel, you will need to reboot the system, before proceeding. This would be a good time to pick an appropriate I/O scheduler. Available I/O schedulers, and the kernel option to select them, are as follows.

- Completely Fair Queueing – elevator=cfq
- Deadline – **elevator=deadline**
- NOOP – **elevator=noop**
- Anticipatory – elevator=as

According to a citation in Dell Equallogic (2009), the Open-iSCSI group reports that sometimes the NOOP scheduler works best for iSCSI server environments. However, if this server will be used as an Oracle database or application server, it is standard best practice to use the Deadline scheduler for optimal performance.

Run the following command to enable the NOOP scheduler, before rebooting.
sed -e 's!/vmlinuz.*!& elevator=noop!g' -i /boot/grub/menu.lst

If you are provisioning an Oracle server, then run the following to select Deadline.
sed -e 's!/vmlinuz.*!& elevator=deadline!g' -i /boot/grub/menu.lst

Finally, reboot the server after upgrading the system and kernel. Do this before proceeding with the configuration steps that follow so the proper I/O scheduler will be active and the newer kernel will be available. You may also want to go back and uncomment the *exclude=#kernel** line in */etc/yum.conf*, at your own discretion.

IV. Configuring Open-iSCSI

1. Configuring the NIC cards for iSCSI use

In the following example, it is assumed you will be using the third (eth2) and fourth (eth3) Network Interface Cards in the system for iSCSI traffic. If this is not the case, then adjust the configuration for different NICs as necessary.

Add the following bold lines to each of the *ifcfg-ethX* scripts for each iSCSI NIC. Ensure that *IPADDR*, and *NETMASK* are correctly set for your iSCSI subnet. Make sure *ONBOOT* is set to *yes*. This is set so that each interface will be automatically initialized when the system is booted. The *MTU* variable should be set to *9000*, for each iSCSI NIC, this MTU setting will enable jumbo frames.

/etc/sysconfig/network-scripts/ifcfg-eth2

```
# Broadcom Corporation NetXtreme II BCM5709 Gigabit Ethernet
DEVICE=eth2
HWADDR=00:11:22:33:44:aa
ONBOOT=yes
BOOTPROTO=none
NETMASK=255.255.255.0
IPADDR=10.1.2.3
TYPE=Ethernet
MTU=9000
```

/etc/sysconfig/network-scripts/ifcfg-eth3

```
# Broadcom Corporation NetXtreme II BCM5709 Gigabit Ethernet
DEVICE=eth3
HWADDR=00:11:22:33:44:bb
ONBOOT=yes
BOOTPROTO=none
NETMASK=255.255.255.0
IPADDR=10.1.2.4
TYPE=Ethernet
MTU=9000
```

Dell EqualLogic (2009) recommends enabling Flow Control, and disabling Auto Negotiation on each iSCSI NIC. For these settings to be automatically applied upon each boot, add the following code, in boldface, to the */etc/rc.local* file.

```
/etc/rc.local
# iSCSI Interface Settings for Equallogic
ISCSI_IF="eth2 eth3"
ETHTOOL_OPTS="autoneg off rx on tx on"
ETHTOOL=`which ethtool`

for i in $ISCSI_IF;
do
        echo "$ETHTOOL -A $i $ETHTOOL_OPTS"
        $ETHTOOL -A $ISCSI_IF $ETHTOOL_OPTS
done
```

1.1. Configuring Open-iSCSI initiator utilities

You need to tune a few lines in the */etc/iscsi/iscid.conf* file, per vendor recommendations. You can grep out blank and comment lines in this file for quick inspection. The lines we are most concerned about will appear in bold. Values set per specific vendor recommendations are highlighted in red.

1.1.1. Notes about iscsid.conf configuration

As I understand it, the configuration item, *node.session.timeo.replacement_timeout* is for specifying the length of time, in seconds, after which the iSCSI layer will fail back to the Device Mapper Multipath application layer. By default, this is 120 seconds, or 2 minutes. According to Red Hat (2008), it is preferable to pass the path failure quickly from the SCSI layer to your Multipath software, which would be necessary for quick fail-over. With the default configuration, two minutes of I/O would be queued before failing the path. If you are familiar with Fiber Channel path fail-over, it tends to be instantaneous. Most people configuring iSCSI fail-over certainly don't want a failed path to be retried for two whole minutes. Using the default configuration with a production database server could lead to massive amounts of I/O being queued instead of the expected behavior of failing over to another working path quickly. Changing this value to 15 will allow the path to failover much more quickly.

Red Hat (2008) also recommends setting *node.conn[0].timeo.noop_out_interval* to *5,* and *node.conn[0].timeo.noop_out_timeout* to 10. I recommend changing only the *node.session.timeo.replacement_timeout* to *15*. I found the best results during my testing with the default value as shipped with RHEL, for both timeo.noop_out_interval and timeo.noop_out_timeout that default value was 5.In either case, test your configuration to see how it affects Device Mapper Multipath.

According to Dell (2010), if the Broadcom bnx2i interface fails to login to an Equallogic iSCSI storage volume, the *node.session.initial_login_retry_max* should be changed from the default value of *8* to *12*.

According to Dell Equallogic (2009) the options *node.session.cmds_max*, and *node.session.queue_depth* should be changed for Equallogic arrays. The recommendation for *node.session.cmds_max* is to change the value from *128 to 1024.* The recommendation for *node.session.queue_depth* is to change the value from *32* to *128*.

Finally, according to the in-line comments in the default iscsid.conf, and Dell (2010) Equallogic arrays should have *node.session.isci.FastAbort* set to *No*. If using software such as iSCSI Enterprise Target (IET), instead of an Equallogic Array, leave the *FastAbort* value set to the default, which is *Yes*.

1.1.2. iscsid.conf

egrep -v "^(#|$)" /etc/iscsi/iscid.conf

node.startup = automatic
node.session.timeo.replacement_timeout = 15 # default 120; RedHat recommended
node.conn[0].timeo.login_timeout = 15
node.conn[0].timeo.logout_timeout = 15
node.conn[0].timeo.noop_out_interval = 5
node.conn[0].timeo.noop_out_timeout = 5
node.session.err_timeo.abort_timeout = 15
node.session.err_timeo.lu_reset_timeout = 20
node.session.initial_login_retry_max = 12 # default 8; Dell recommended
node.session.cmds_max = 1024 # default 128; Equallogic recommended
node.session.queue_depth = 128 # default 32; Equallogic recommended
node.session.iscsi.InitialR2T = No
node.session.iscsi.ImmediateData = Yes
node.session.iscsi.FirstBurstLength = 262144
node.session.iscsi.MaxBurstLength = 16776192
node.conn[0].iscsi.MaxRecvDataSegmentLength = 262144
discovery.sendtargets.iscsi.MaxRecvDataSegmentLength = 32768
node.conn[0].iscsi.HeaderDigest = None
node.session.iscsi.FastAbort = No # default Yes; Dell / Open-iSCSI recommended

IV. Targeting and Logging in with iscsiadm

1. Create iSCSI interfaces

The iSCSI interfaces are separate pseudo-devices used by Open-iSCSI. From Open-iSCSI's point-of-view, these are not the same as physical Ethernet devices. What you'll need to do is bind each of your ethX devices to an iSCSI interface. Technically speaking, you could call your iSCSI interfaces eth2 and eth3, and bind them to the corresponding physical devices in the /dev folder. To avoid any confusion about physical NICs and iSCSI interfaces, it's recommended to give the interfaces different names like iface0 and iface1.

You can create your iSCSI interfaces with the following commands.

```
iscsiadm -m iface -I iface0 -o new
iscsiadm -m iface -I iface1 -o new
```

Second, let's take a look at the iSCSI interface properties for one of our new interfaces. This will come in handy to view the currently active configuration in cases where you need to troubleshoot.

This is how you can look at the iSCSI interface properties for one of the new interfaces. Being able to view the currently active configuration can come in handy in cases where you need to troubleshoot.

```
iscsiadm -m iface -I iface0
```

```
# BEGIN RECORD 2.0-871
iface.iscsi_ifacename = iface0
iface.net_ifacename = <empty>
iface.ipaddress = <empty>
iface.hwaddress = <empty>
iface.transport_name = tcp
iface.initiatorname = <empty>
# END RECORD
```

2. Bind iSCSI interfaces

Now that we have iSCSI interfaces, we need to bind them to a physical Ethernet device. You can use either a device name or MAC address. Open-iSCSI won't even let you bind the interfaces with both a device name and MAC address.

Binding by physical device name

```
iscsiadm -m iface -o update -I iface0 -n iface.net_ifacename -v eth2
iscsiadm -m iface -o update -I iface1 -n iface.net_ifacename -v eth3
```

Binding by MAC address

```
iscsiadm -m iface -o update -I iface0 -n iface.hwaddress -v 00:aa:bb:cc:dd:ee
iscsiadm -m iface -o update -I iface1 -n iface.hwaddress -v 00:aa:bb:cc:dd:ff
```

An important note about alternate transport drivers for iSCSI offload: By default, your new iSCSI interfaces will use TCP as the iSCSI transport. There are other offload transport drivers available to use, such as **bnx2i**, **iser**, and **cxgb3i**. According to the iscsiadm manual page and personal conversation with a Dell storage engineer, these are experimental drivers which are not supported or considered stable.

2.1. Updating iSCSI interfaces

If you bind an interface by MAC address, and have a hardware replacement which changes the MAC address. Then the iSCSI interface bindings will need to be updated to reflect such system changes. You can do this with an update command:

```
iscsiadm -m iface -o update -I iface0 -n iface.hwaddress -v 00:aa:bb:cc:dd:11
iscsiadm -m iface -o update -I iface1 -n iface.hwaddress -v 00:aa:bb:cc:dd:33
```

3. Connecting to the iSCSI array

The file */etc/iscsi/initiatorname.iscsi* should contain an initiator name for your iSCSI client host. You need to include this initiator name on your iSCSI array's configuration for this specific iSCSI client host.

If you haven't yet started the iSCSI daemon, run the following command before we commence with discovering targets.

```
service iscsid start
```

3.1 Discovering targets

Once the iscsid service is running and the client's initiator name is configured on the iSCSI array, then you may proceed with the following command to discover available targets. Assuming our iSCSI array had an IP address of 10.1.2.10, the following command would return the available targets.

```
iscsiadm -m discovery -t st -p 10.1.2.10:3260
10.1.2.10:3260,1 iqn.2001-05.com.equallogic:0-8a0906-7008ec504-23d000000204bad0-hostname-vol0
10.1.2.10:3260,1 iqn.2001-05.com.equallogic:0-8a0906-7008ec504-23d000000204bad0-hostname-vol0
```

3.2. Login to target

We're finally ready to login to our target. To log in to all targets, use the following command. This should return successful if everything is working correctly.

```
iscsiadm -m node -l
```

Individual targets can be logged into by specifying a whole target name.

```
iscsiadm -m node -T iqn.2001-05.com.equallogic:0-8a0906-7008ec504-23d000000204bad0-hostname-vol0 -l -p 10.1.2.10:3260
```

3.3. Logoff a target

Logging off is basically the same as logging into a target, except you use *-u* instead of *-l.*

```
iscsiadm -m node -u
```

3.4. Session status

The session command can be used to print the basic status or more verbose output for debugging and troubleshooting.

Basic command:

```
iscsiadm -m session
tcp: [10] 10.1.2.10:3260,1 iqn.2001-05.com.equallogic:0-8a0906-7008ec504-
23d000000204bad0-hostname-vol0
tcp: [9] 10.1.2.10:3260,1 iqn.2001-05.com.equallogic:0-8a0906-7008ec504-
23d000000204bad0-hostname-vol0
```

Troubleshooting information with -P (print) flag and verbosity level 0-3:

```
iscsiadm -m session -P3
iSCSI Transport Class version 2.0-871
version 2.0-871
Target: iqn.2001-05.com.equallogic:0-8a0906-7008ec504-23d000000204bad0-hostname-
vol0
Current Portal: 10.1.2.25:3260,1
Persistent Portal: 10.1.2.10:3260,1
**********
Interface:
**********
Iface Name: iface0
Iface Transport: tcp
Iface Initiatorname: iqn.1994-05.com.redhat:13c39f80866f
Iface IPaddress: 10.1.2.3
Iface HWaddress: <empty>
Iface Netdev: eth2
SID: 10
iSCSI Connection State: LOGGED IN
iSCSI Session State: LOGGED_IN
Internal iscsid Session State: NO CHANGE
************************
Negotiated iSCSI params:
************************
HeaderDigest: None
DataDigest: None
MaxRecvDataSegmentLength: 262144
MaxXmitDataSegmentLength: 65536
FirstBurstLength: 65536
MaxBurstLength: 262144
ImmediateData: Yes
InitialR2T: No
MaxOutstandingR2T: 1
************************
Attached SCSI devices:
************************
Host Number: 27 State: running
scsi27 Channel 00 Id 0 Lun: 0
Attached scsi disk sdc          State: running
```

V. Configuring Device Mapper Multipath

1. Notes about the Device Mapper Multipath example

The final major step is to configure Device Mapper Multipath using the configuration file */etc/multipath.conf*. By default there will be a *devnode "*"* line in the *blacklist* section, thereby disabling Multipath for all devices in the system. If you want to assign persistently bound names to iSCSI devices, be sure to set *user_friendly_names* to yes, as seen in the example on the following page. If you do not use friendly names with Device Mapper, you'll end up with device names such as */dev/mapper/36090affffffffffffffffffffffffff*. When using friendly names, you'll need to specify an *alias* line in the *multipaths* section after Open-iSCSI is configured and working correctly with your iSCSI target or array.

You will want to blacklist any local devices that do not have multiple paths to be managed in the *blacklist* section. It is recommended to blacklist the local SCSI disks by World Wide Identifier (WWID). The WWID is a unique identifier for any block device and may be retrieved with the command *scsi_id -g -u -s /block/sdX*, where X is the letter identifier of the local SCSI disk. It cannot hurt to also blacklist local devices by physical device name with a *devnode* line and a Perl-compatible regular expression string. The reason for blacklisting a second time with a regular expression is in case the *scsi_id* program fails to read the WWID from sector zero of a local device. See the blacklist section in the configuration example. The WWID line and *devnode "^sd[a]$"* line both serve as a blacklist for device sda.

Once you have discovered targets and logged in to the iSCSI array, you can use the '*iscsiadm -m session -P3'* command to find the physical device names of your iSCSI volumes. Then you can use the '*scsi_id -g -u -s /block/sdX'* command to find the WWID of your iSCSI volumes. Once you have the WWIDs of your iSCSI volumes, you can configure friendly name aliases in the *multipaths* section, as seen in the example on the next page.

The device section in the configuration example is pertinent to an Equallogic PS Series array. The lines in boldface type are of particular importance, and are the recommended defaults for Equallogic devices. You can override the parameters on a per-volume basis in the *multipaths* section.

For best performance, the value for *rr_min_io* should be in the range of *10-20* for database environments. For more sequential loads, which may be seen on file servers, performance might be better if *rr_min_io* is set in the range *100-512*. Any *rr_min_io* value over *200* will require max commands and queue depths parameters in *iscsid.conf* to be increased.

1.1. multipath.conf

```
defaults {
    user_friendly_names yes
}

blacklist {
    wwid 360924ffffffffffffffffffffffff
    devnode "^sd[a]$"
    devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
    devnode "^hd[a-z][[0-9]*]"
    devnode "^cciss!c[0-9]d[0-9]*[p[0-9]*]"
}

devices {
    device {
        vendor "EQLOGIC"
        product "100E-00"
        path_grouping_policy multibus
        getuid_callout "/sbin/scsi_id -g -u -s /block/%n"
        features "1 queue_if_no_path"
        no_path_retry fail
        path_checker readsector0
        failback immediate
        path_selector "round-robin 0"
        rr_min_io 10
        rr_weight priorities
    }
}

multipaths {
    multipath {
        wwid 36090affffffffffffffffffffffff
        alias u02
    }
}
```

1.2. Start and test Device Mapper Multipath

Device Mapper Multipath can be started with the following command once it has been configured.

```
service multipathd start
```

You can list the Multipath topology with the following command to verify everything is working correctly.

```
multipath -ll -v2
u02 (36090a04850ec0870d0ba04020000d023) dm-0 EQLOGIC,100E-00
[size=100G][features=0][hwhandler=0][rw]
\_ round-robin 0 [prio=2][active]
 \_ 10:0:0:0 sdb 8:16  [active][ready]
 \_ 11:0:0:0 sdc 8:32  [active][ready]
```

1.3. Reload udev to test friendly alias names

Once Multipath has been verified to be working correctly, you may need to run the following *udev* command to create devices with friendly names. This only applies if you have chosen to use persistent Multipath binding by defining *alias* lines in the *multipaths* section of *multipath.conf*. Reloading udev will automagically create your aliased devices in the */dev/mapper* folder.

Reload udev command:

```
udevcontrol reload_rules
```

You may proceed to *fdisk* and format the aliased device, just as you would any other SCSI disk device appearing in the /dev directory.

VI. Final steps

1. Set services to start automatically.

Once you have everything up and running, with regards to Open-iSCSI and Device Mapper Multipath, run the following commands to ensure services get started automatically during server boot up.

Set the iscsid daemon to start automatically:

```
chkconfig iscsid on
```

Set the iscsi service to log in to targets automatically:

```
chkconfig iscsi on
```

Set the multipathd daemon to start automatically:

```
chkconfig multipathd on
```

Edit */etc/fstab*, and add a line with the *_netdev* keyword for all your volumes to be mounted. Using the *_netdev* keyword ensures this device will not be mounted before the networking subsystem has started.

We've observed Open-iSCSI failing both paths temporarily while updating firmware on the group node. This could result in a machine remounting a filesystem in read-only mode upon error unless an '*errors=continue*' option is added to /etc/fstab.

```
LABEL=/              /           ext3    defaults        1 1
LABEL=/u01           /u01        ext3    defaults        1 2
/dev/mapper/u02p1    /u02        ext3    _netdev,defaults,errors=continue 0 0
tmpfs                /dev/shm    tmpfs   defaults        0 0
devpts               /dev/pts    devpts  gid=5,mode=620  0 0
sysfs                /sys        sysfs   defaults        0 0
proc                 /proc       proc    defaults        0 0
LABEL=SWAP-sda5      swap        swap    defaults        0 0
```

2. Final testing

2.1. Reboot

Reboot your server, and make sure everything comes up. The iscsi initialization script should log the server in to the iSCSI targets. Multipath should be started, and it should see both of its paths to the iSCSI array. If everything is working correctly, the *fstab* entry should automatically mount any iSCSI volumes.

2.2. Test your Multipath software

The easiest way to test your Multipath software is to take down one of the Ethernet devices manually. To do this, run the following command.

```
ifdown eth3
```

After about 15 seconds you should see something like this in the messages log.

```
kernel: device-mapper: multipath: Failing path X:XX
multipathd: <alias>: remaining active paths: 1
```

Then, bring the device back up manually:

```
ifup eth3
```

Again, after about 15 seconds you should see something like this in the messages log.

```
multipathd: X:XX: reinstated
multipathd: <alias>: remaining active paths: 2
```

Repeat this test for every other NIC being used for iSCSI and make sure every iSCSI Network Interface fails-over from faulty paths, and reinstate all active iSCSI paths gracefully.

References and recommended reading

Aizman, Alex, and Dmitry Yusupov. 2007. Open-iSCSI – RFC3720 architecture and implementation. *Open-iSCSI project*. http://www.open-iscsi.org/index.html#docs.

Anon. Device-mapper Resource Page. *Device-Mapper-Multipath Project*. http://sources.redhat.com/dm/.

Dell EqualLogic, Inc. 2008. *PS Series Array Network Performance Guidelines*. 3rd ed. Nashua, NH: Dell, Inc, June. http://www.equallogic.com/uploadedfiles/Resources/Tech_Reports/tr-network-guidelines-TR1017.pdf.

———. 2009. *Red Hat Linux v5.x Software iSCSI Initiator Configuration, MPIO and tuning Guide*. Nashua, NH: Dell, Inc, December. http://www.equallogic.com/resourcecenter/assetview.aspx?id=8727.

———. 2010. *Dell Poweredge Systems Red Hat Enterprise Linux 5.4 (x86_64, x86) Installation Instructions and Important Information,* Rev. A02. March. http://support.dell.com/support/edocs/software/rhel_mn/rhel5_4/iig_en.pdf.

Red Hat, Inc. 2007. How do I configure the iscsi-initiator in Red Hat Enterprise Linux 5? *Red Hat Knowledgebase*. June 25. https://access.redhat.com/kb/docs/DOC-6388.

———. 2008. How can I improve the failover time of a faulty path when using device-mapper-multipath over iSCSI? *Red Hat Knowledgebase*. October 1. https://access.redhat.com/kb/docs/DOC-2877.

———. 2010a. *DM Multipath*. http://www.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5.5/html/DM_Multipath/index.html.

———. 2010b. Kickstart Installations. In *Installing Red Hat Enterprise Linux 5 for all architectures*, Chap. 31. Research Triangle Park, NC: Red Hat, Inc. http://www.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5.5/html/Installation_Guide/index.html.