How-To Linux Password Recovery and Rescue Mode

Version: 1.0

Published: February 2020

Author: Gideon Asanji

**Purpose**

The purpose of this document is to outline the steps Which may be used to recover a Linux server where the root password is unknown but the server has not fully booted allowing login via ssh.

**Disclaimer**: *Information contained in this document is provided on a “Subject to Contract” basis and is intended by CTL to be solely for CTL reference. This document is not, therefore, contractually binding on CTL and it may be subject to change from time to time as CTL deems appropriate. CTL assumes no liability, whether in contract, tort or any other legal theory, in respect of this document, its contents, or any use to which CTL may put the same. CTL disclaims in particular any express or implied warranties, whether statutory or otherwise, relating to fitness for a particular purpose, satisfactory quality, merchantability or infringement of any patent, copyright or other intellectual property with respect to this document.*

Document Details

This is an internal document intended for CenturyLink Technology Solutions (CTL) Global Operations Centre personnel.

Document Version History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version | Date | Change | Author/Input | Reviewed By | Approved |
| **1.0** | 15/02/2020 | Initial Draft | Gideon Asanji |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Document Approval Authority

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Organisation | Title | Signature | Date |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Table of Contents

[1 Introduction 6](#_Toc439771262)

[2 Notes on the Console 7](#_Toc439771263)

[3 Notes on Boot Modes 8](#_Toc439771264)

[4 Boot into single user mode 9](#_Toc439771265)

[4.1 Red Hat Enterprise Linux 5 or 6 9](#_Toc439771266)

[4.2 Red Hat Enterprize Linux 7 9](#_Toc439771267)

[4.2.1 Grub2 Menu Editing During Boot 9](#_Toc439771268)

[4.2.2 Boot into rescue mode 9](#_Toc439771269)

[5 Boot into emergency mode 10](#_Toc439771270)

[5.1 Red Hat Enterprize Linux 5 or 6 10](#_Toc439771271)

[5.2 Red Hat Enterprize Linux 7 10](#_Toc439771272)

[6 Boot into a bash shell 11](#_Toc439771273)

[6.1 Red Hat Enterprize Linux 5 or 6 11](#_Toc439771274)

[6.2 Red Hat Enterprize Linux 7 12](#_Toc439771275)

[7 Resetting the Root Password Using rd.break 13](#_Toc439771276)

[8 How to Boot Linux Rescue Mode from a Jumpkick Server 14](#_Toc439771277)

[9 Additional notes for VMware and SOS report 20](#_Toc439771278)

Table of Tables

[**Table 1 – Comparison of recovery modes** 8](#_Toc439771279)

Table of Figures

[**Figure 1 – MAC address and Vlan setting on a virtual Machine** 16](#_Toc439771462)

[Figure 2 – Booting the Network Device on an HP Proliant 17](#_Toc439771463)

[Figure 3 – The “running install” message 17](#_Toc439771464)

[Figure 4 – Mounting your Linux installation 17](#_Toc439771465)

[Figure 5 – Entering the shell on HP Proliant 18](#_Toc439771466)

[**Figure 6 – Entering the shell on a Virtual Machine** 18](#_Toc439771467)

[Figure 7 – Accessing the root file system 19](#_Toc439771468)

# Introduction

From time to time we encounter a situation where a server has failed to reboot and once the console is successfully connect to it is found to be requesting the root password before it is possible to perform a manual fsck (file system check) or other recovery actions. This becomes an issue if the root password is not available or for some reason simply does not work. This document describes a number of strategies that may be used to reset the root password of a server by the use of different boot modes. It is intended for guidance and should not be thought of as exhaustive.

NOTE: While not all off the boot modes that are discussed are suitable for password recovery they may prove useful for other recovery actions outside the scope of this document and so have been included completeness.

# Notes on the Console

Occasionally we see a situation where a server has failed to reboot and the console is blank or appears to “hang” part of the way into the boot process. While this situation is occasionally the result of an actual hardware fault it is most often the result of incorrect configuration.

Unfortunately the first action most often taken in an attempt to “resolve” this issue is to power cycle the machine, often many times which will NOT resolve the issue and frequently leads to file system damage.

This issue is the result of multiple “console=” parameters within the /boot/grub.conf similar to the example shown in below:-

title SAVVIS Linux [multi user mode serial console]

root (hd0,0)

kernel /vmlinuz-2.6.18-53.1.13.el5 ro root=/dev/vg\_root/lv\_root **console=ttyS0 console=tty0**

initrd /initrd-2.6.18-53.1.13.el5.img

Within Linux it is possible to output Boot loader and kernel messages to multiple consoles simultaneously, however Init script messages can only be output to the “master” console. The last console that is declared in the grub.conf is the master console (tty0 in this case). So in this case if you are viewing the server boot from the Serial console via a Cyclades connection the boot will appear to hang with the last message displayed looking something like the below:-

EXT3-fs: mounted filesystem with ordered data mode.

type=1403 audit(1402905086.993:2): policy loaded auid=4294967295 ses=4294967295

warningIt is important to realise that messages are still being sent to the “master” console at this point and any issues encountered during the boot will only appear there.

* VMware - does not provide a serial console, ensure that there is only one console entry in /boot/grub.conf e.g. console=tty0
* HP ILO – this is a simulated VGA console ensure grub.conf has a single console=tty0
* HP serial ilo (Cyclades) - this is a serial console ensure that grub.conf reads console=ttyS0 and optionally specify speed, stop bits and flow control (console=ttyS0 9600,8,1)

Note: in Redhat 6 and 7 we often see “rhgb” and “quiet” these should be removed if present

* rhgb = Redhat graphical boot - This is a GUI mode booting screen with most of the information hidden
* quiet = hides the majority of boot messages before rhgb starts. These are supposed to make the common user more comfortable.

# Notes on Boot Modes

Red Hat Enterprise Linux 6 offers three standard system recovery modes, single-user mode, emergency mode and rescue mode that can be used to repair malfunctioning systems.

Red Hat Enterprise Linux 7 also offers three standard system recovery modes, rescue mode, emergency mode and the debug shell, however while the names are similar these modes do not have the same functionality. The table below relates these modes and provides a short description of each.

Note that it is also possible to boot directly to a shell, this is not an official recovery mode but it is supported in all versions of Red Hat Enterprise Linux and can be of considerable use especially when resetting a lost root password. However for Red Hat Enterprise Linux 7 the preferred option is to use the rd.break option.

Red Hat Enterprise Linux 6 rescue mode i.e. booting from external media is supported in Red Hat Enterprise Linux 7.

|  |  |  |
| --- | --- | --- |
| RHEL 5/6 | RHEL 7 | Description |
| single-user mode | rescue mode | The system attempts to mount all local file systems and start some important system services, but it does not activate network interfaces or allow more users to be logged into the system at the same time. **Requires the root password.** |
| emergency mode | emergency mode | provides the most minimal environment possible, the system mounts the root file system only for reading, does not attempt to mount any other local file systems, does not activate network interfaces, and only starts few essential services. **Requires the root password**. |
| - | debug shell | The systemd debug shell provides a shell very early in the startup process that can be used to diagnose systemd related boot-up problems. **Requires the root password.** |
| /bin/bash | /bin/bash | Loads the kernel and the specified shell. **Does NOT require the root password.** |
| - | rd.break | The initramfs will stop before passing control to the Linux kernel, enabling you to work with the root file system. **Does NOT require the root password.** |
| rescue mode | - | Provides the ability to boot a small Red Hat Enterprise Linux environment entirely from external media. It contains command-line utilities for repairing a wide variety of issues. **Does NOT require the root password.** |

**Table 1 – Comparison of recovery modes**

# Boot into single user mode

## Red Hat Enterprise Linux 5 or 6

In single-user mode, your computer boots to run level 1. Your local file systems are mounted, but your network is not activated. You have a usable system maintenance shell. Single-user mode automatically tries to mount your file system. Do not use single-user mode if your file system cannot be mounted successfully. You cannot use single-user mode if the run level 1 configuration on your system is corrupted.

1. At the GRUB splash screen at boot time, press any key to enter the GRUB interactive menu.
2. Select “**Red Hat Enterprise Linux”** with the version of the kernel that you wish to boot and type a to append the line.
3. Go to the end of the line and type **single** as a separate word (press the **Spacebar** and then type **single**). Press **Enter** to exit edit mode.
4. Press the **b** key to boot the system with this option.

## Red Hat Enterprise Linux 7

### Grub2 Menu Editing During Boot

Menu entries can be modified and arguments passed to the kernel on boot. This is done using the menu entry editor interface, which is triggered when pressing the **e** key on a selected menu entry in the boot loader menu. The **Esc** key discards any changes and reloads the standard menu interface. The **c** key loads the command line interface.

The command line interface is the most basic GRUB interface, but it is also the one that grants the most control. The command line makes it possible to type any relevant GRUB commands followed by the **Enter** key to execute them. This interface features some advanced features similar to **shell**, including **Tab** key completion based on context, and **Ctrl**+**a** to move to the beginning of a line and **Ctrl**+**e** to move to the end of a line. In addition, the **arrow**, **Home**, **End**, and **Delete** keys work as they do in the bash shell.

### Boot into rescue mode

Rescue mode provides a convenient single-user environment and allows you to repair your system in situations when it is unable to complete a normal booting process. In rescue mode, the system attempts to mount all local file systems and start some important system services, but it does not activate network interfaces or allow more users to be logged into the system at the same time. In Red Hat Enterprise Linux 7, rescue mode is equivalent to single user mode and requires the root password.

1. To enter rescue mode during boot, on the GRUB 2 boot screen, press the **e** key for edit.
2. Add the following parameter at the end of the Linux line on 64-Bit IBM Power Series, the linux16 line on x86-64 BIOS-based systems, or the linuxefi line on UEFI systems:

**systemd.unit=rescue.target**

Press **Ctrl**+**a** and **Ctrl**+**e** to jump to the start and end of the line, respectively. On some systems, **Home** and **End** might also work.

Note that equivalent parameters, *1*, *s*, and *single*, can be passed to the kernel as well.

1. Press **Ctrl**+**x** to boot the system with the parameter.

# Boot into emergency mode

## Red Hat Enterprise Linux 5 or 6

In emergency mode, you are booted into the most minimal environment possible. The root file system is mounted read-only and almost nothing is set up. The main advantage of emergency mode over single-user mode is that the init files are not loaded. If init is corrupted or not working, you can still mount file systems to recover data that could be lost during a re-installation.

1. At the GRUB splash screen at boot time, press any key to enter the GRUB interactive menu.
2. Select **Red Hat Enterprise Linux** with the version of the kernel that you wish to boot and type a to append the line.
3. Go to the end of the line and type **emergency** as a separate word (press the **Spacebar** and then type **emergency**). Press **Enter** to exit edit mode.
4. Press the **b** key to boot the system with this option.

## Red Hat Enterprise Linux 7

Emergency mode provides the most minimal environment possible and allows you to repair your system even in situations when the system is unable to enter rescue mode. In emergency mode, the system mounts the root file system only for reading, does not attempt to mount any other local file systems, does not activate network interfaces, and only starts few essential services. In Red Hat Enterprise Linux 7, emergency mode requires the root password.

1. To enter emergency mode, on the GRUB 2 boot screen, press the **e** key for edit.
2. Add the following parameter at the end of the Linux line on 64-Bit IBM Power Series, the linux16 line on x86-64 BIOS-based systems, or the linuxefi line on UEFI systems:

systemd.unit=emergency.target

Press **Ctrl**+**a** and **Ctrl**+**e** to jump to the start and end of the line, respectively. On some systems, **Home** and **End** might also work.

Note that equivalent parameters, *emergency* and *-b*, can be passed to the kernel as well.

1. Press **Ctrl**+**x** to boot the system with the parameter.

# Boot into a bash shell

This feature can be used as the last option before having to perform a recovery boot and has the advantage that none of the system init scripts are run.

In all Unix-like systems the system scheduler known as init or systemd in later releases of Redhat is normally the first process to be run, and the ultimate ancestor of all processes ever run. It's responsible for running all the init/startup scripts.

So here we are instructing the kernel to load the bash shell rather than init. Note that no file systems will be mounted or processes started other than the bash shell.

## Red Hat Enterprise Linux 5 or 6

1. At the GRUB splash screen at boot time, press any key to enter the GRUB interactive menu.
2. Select **Red Hat Enterprise Linux** with the version of the kernel that you wish to boot and type a to append the line.
3. Go to the end of the line and type **init=/bin/bash** as a separate word (press the **Spacebar** and then type **init=/bin/bash**). Press **Enter** to exit edit mode.
4. Press the **b** key to boot the system with this option.
5. Once booted to the bash prompt check that the root file system is mounted read and write

# mount | grep root

# mount -o remount,rw /

Note: The system will be booted from the “system disk” so there is no requirement to use the chroot command.

1. At this point only the root file system and (/) and swap will be mounted which means that the passwd command (/usr/bin/passwd) will not be available until the /usr file system is mounted.

In the standard CenturyLink build all file systems with the exception of /boot are created on LVM Logical Volumes, however, as init and so udev were not executed the root volume group is not active and the device nodes have not been created . You will find that any attempt to mount the /usr file system will result in an error. Similar to :

mount: special device /dev/mapper/vg\_root-lv\_usr does not exist

To work around this you will need to activate the volume group containing the volumes you want to mount, in most cases vg\_root.

Note: if LVM is not in use you should be able to mout the partition as seen in /etc/fstab.

Note: you may need to carry out a file system chack on /usr before you can mount it.

1. Activate the root volume group

# vgchange –ay vg\_root

1. If you don’t see any entries under /dev/vg\_root you may need to do a vgscan

# vgscan –-mknodes

# mount /usr

1. Now we can simply use the passwd command and follow the prompts

# passwd

1. If SElinux is enabled (or permissive) don’t forget to relabel the SELinux contexts

# touch /.autorelabel

1. And finally reboot the system.

# cd /

# umount /usr

# vgchange –an vg\_root

# mount -o remount,ro /

# exec /sbin/init

Note: using the “reboot or “/sbin/shutdown” commands will not work when booted directly to a shell.

Note: In step 9 of the above example you can just execute /sbin/init, but you may see a number of error messages relating to the root volume group and the /usr file system. These can safely be ignored.

## Red Hat Enterprise Linux 7

1. To enter emergency mode, on the GRUB 2 boot screen, press the **e** key for edit.
2. Add the following parameter at the end of the Linux line on 64-Bit IBM Power Series, the linux16 line on x86-64 BIOS-based systems, or the linuxefi line on UEFI systems:

Replace **rhgb quiet** with **init=/bin/bash**

Press **Ctrl**+**a** and **Ctrl**+**e** to jump to the start and end of the line, respectively. On some systems, **Home** and **End** might also work.

1. Press **Ctrl**+**x** to boot the system with the parameter.
2. Once booted to the bash prompt check that the root file system is mounted read and write

# mount | grep root

# mount -o remount,rw /

1. Now we can simply use the password command and follow the prompts

# passwd

1. If SElinux is enabled (or permissive)don’t forget to relabel the SELinux contexts

# touch /.autorelabel

1. And finally reboot the system.

# exec /sbin/init

Note: using the “reboot or “/sbin/shutdown” commands will not work when booted directly to a shell.

# ****Resetting the Root Password Using rd.break****

1. Start the system and, on the GRUB 2 boot screen, press the **e** key for edit.
2. Remove the rhgb and quiet parameters from the end, or near the end, of the linux16 line, or linuxefi on UEFI systems.
3. Press **Ctrl**+**a** and **Ctrl**+**e** to jump to the start and end of the line, respectively. On some systems, **Home** and **End** might also work.

append **rd.break**

1. Press **Ctrl**+**x** to boot the system with the changed parameters.

The initramfs switch\_root prompt appears.

1. The file system is mounted read-only on /sysroot/. You will not be allowed to change the password if the file system is not writable.
2. Remount the file system as writable:

switch\_root:/# mount -o remount,rw /sysroot

1. The file system is remounted with write enabled, Change the file system's root as follows:

switch\_root:/# chroot /sysroot

The prompt changes to

sh-4.2#.

1. Enter the passwd command and follow the instructions displayed on the command line to change the root password.

Note: that if the system is not writable, the **passwd** tool fails with the following error:

Authentication token manipulation error

1. Updating the password file results in a file with the incorrect SELinux security context. To relabel all files on next system boot, enter the following command:

sh-4.2# touch /.autorelabel

1. Remount the file system as read only:

sh-4.2# mount -o remount,ro /

1. Enter the exit command to exit the chroot environment.
2. Enter the exit command again to resume the initialization and finish the system boot.

# How to Boot Linux Rescue Mode from a Jumpkick Server

## Overview

This process is similar to the process used to build servers and makes use of the Red Hat “kickstart” process. In each CenturyLink data centre there is a combined Oracle “Jumpstart” and Red Hat “kickstart” server known as the “Jumpkick” server. In each data centre the jumpkick server is connected to a dedicated jumpkick VLAN (VLAN 640) which provides services such as DHCP, tftp and nfs.

Using the kickstart process it is possible to use The Preboot Execution Environment (**PXE**) to boot a server into “rescue mode” without requiring physical access to the server, however, extreme care must be taken to avoid re-installing the problem server.

To perform this process the following prerequisite must be met.

* A network interface in the “jumpkick VLAN (640)
* The MAC Address of the network interface to be used for the PXE boot.
* A valid PXE configuration file
* A valid ks.cfg file.

The steps to perform the recovery boot are as follows:-

## The Process

1. Obtain the MAC address for the client interface which will be used for the PXE boot. This may be obtained from one or more of the following sources depending on the platform/configuration.

* The Integrated Lights Out (ILO) manager.
* The ROM-Based Setup Utility (RBSU) on an HP ProLiant.
* The vSphere Client – Virtual Machine Properties for VMware

1. Run the ksadd script on the Jumpkick server, specifying the MAC address obtained in step 1, the version and architechture (32 or 64 bit) of Red hat Linux to be booted, the architechture of the server and which region the server resides in as shown in the below example. This will create the configuration files that would be used to install the server with the standard CenturyLink build. This script will create a kickstart configuration file and a file named for the MAC address of the server which is effectively a grub.conf file used by the PXE boot process.

Usage: ksadd <*hostname*> <*MAC*> <*version*> <*platform*> <*region*>

For example,

[root@jumpbox ~]# **ksadd s216920rgsl901 00-15-60-0E-2E-28 RHEL5.3\_64 hp EMEA**

Registering s216920rgsl901 with version RHEL5.3\_64/hp at 00-15-60-0e-2e-28.

**/data01/jumpkick/clients/s216920rgsl901/ks.cfg** Generated.

PXE Configuration Written.

[root@jumpbox ~]#

1. Modify the Kickstart configuration file created by the ksadd script in step 2, **/data01/jumpkick/clients/<server name>/ks.cfg** as shown in the below examples.

**Note: the ks.cfg file as created by the ksadd script contains may hundreds of configuration directives. You must either edit or replace this file before proceeding.**

**Example ks.cfg for Red Hat Enterprise Linux** **5**

#SVVS|s216920rgsl901|00-15-60-0e-2e-28|RHEL5.3\_64|hp/{}|EMEA|1402048670|svadmin

#

lang en\_US.UTF-8

langsupport --default=en\_US.UTF-8 en\_US.UTF-8

keyboard us

mouse none

nfs --server=192.168.10.2 --dir=/data01/kickstart/builds/RHEL5.3\_64/

network --bootproto=dhcp

**Example ks.cfg for Red Hat Enterprise Linux 6**

#SVVS|S265575ch3vl66t\_recovery|00-50-56-93-5a27|RHEL6.5\_64|vmware/{}|Americas|1447315168|svadmin

nfs --server 192.168.10.2 --dir /data01/kickstart/builds/RHEL6.5\_64/

network --noipv6

network --bootproto=dhcp

lang en\_US.UTF-8

keyboard us

timezone --utc Etc/UTC

**Example ks.cfg for Red Hat Enterprise Linux** **7**

#SVVS|2604764tjvl711|00-50-56-a1-74-43|RHEL7.1\_64|vmware/{}|EMEA|1452000700|svadmin^M

nfs --server 192.168.10.2 --dir /data01/kickstart/builds/RHEL7.1\_64/

network --bootproto=dhcp --noipv6

timezone --utc Etc/UTC

lang en\_US.UTF-8

keyboard --vckeymap=us --xlayouts='us'

1. Modify the PXE boot configuration file.

This file will reside in the /tftpboot/pxelinux.cfg directory and its name comprises the MAC address used in the ksadd command in step 2 with each octet hyphen separated and prefixed with 01.

The change is more or less the same for all versions of Red Hat Enterprise Linux i.e. replace “network” with “text rescue”

For example,

[root@jumpbox ~]# **cat -n /tftpboot/pxelinux.cfg/01-00-15-60-0e-2e-28**

1 default linux

2 serial 0,9600n8

3 label linux

4 kernel kickstart/vmlinuz-rhel5.3\_64

5 ipappend 2

6 append ramdisk\_size=16384 load\_ramdisk=1 initrd=kickstart/initrd-rhel5.3\_64.img network ks=http://192.168.10.2/clients/s216920rgsl901/ks.cfg ksdevice=bootif console=ttyS0 pci=nobfsort

[root@jumpbox ~]#

Notice that the 6th line contains the **network** keyword. This should be replaced with the keywords **text rescue** as shown below. This file is the same for all versions of Linux as it relates to the initial PXE boot environment.

[root@jumpbox ~]# **cat -n /tftpboot/pxelinux.cfg/01-00-15-60-0e-2e-28**

1 default linux

2 serial 0,9600n8

3 label linux

4 kernel kickstart/vmlinuz-rhel5.3\_64

5 ipappend 2

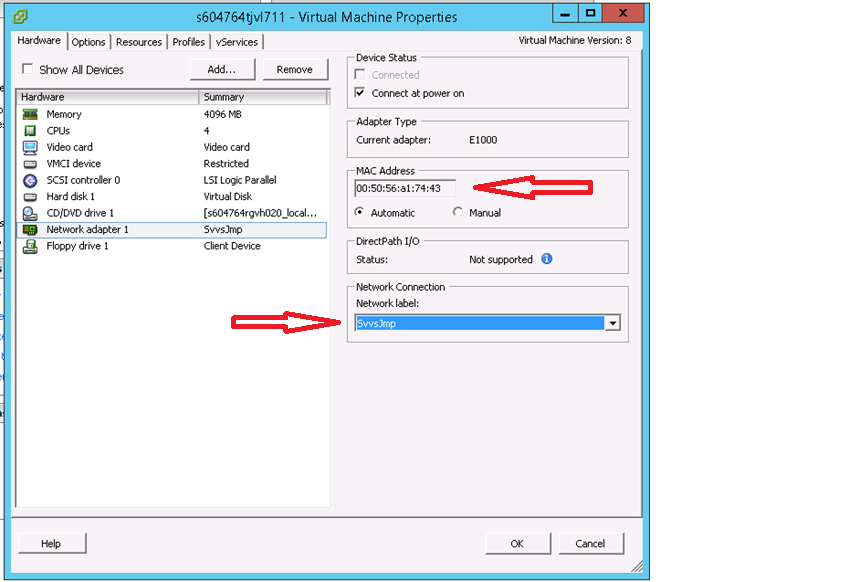
6 append ramdisk\_size=16384 load\_ramdisk=1 initrd=kickstart/initrd-rhel5.3\_64.img text rescue ks=http://192.168.10.2/clients/s216920rgsl901/ks.cfg ksdevice=bootif console=ttyS0 pci=nobfsort

[root@jumpbox ~]#

informationAlso note the **console** parameter; ttyS0 should be specified for a serial console such as a Cyclades device but you may need to change this to tty0 if accessing the console via the ILO.

1. The machine to be recovered will need to have one Network interface moved to the “build VLAN (640) before we can proceed. The method used to achieve this varies between architectures, for example for a physical machine you will need to have the Network team place the client in the build VLAN, but for VMware you can do this yourself via the Virtual Machine settings – Network Connection as shown below.

NOTE: You will need to make a note of the original VLAN configuration so that the configuration can be restored later.



**Figure 1 – MAC address and Vlan setting on a virtual Machine**

1. reboot the machine using the network interface that was placed into the “build” VLAN in step 5 above as the boot device. Figure 1 shows an example of how to do this on an HP ProLiant server, however, once again this varies with Vendor, BIOS version, and architecture.

Note: This document does not detail how to achieve this.

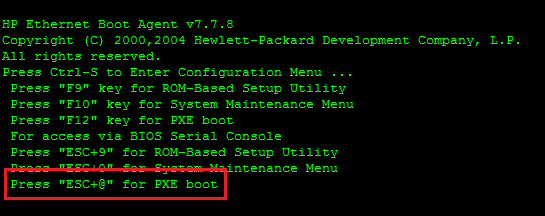


Figure – Booting the Network Device on an HP Proliant

warningDo not be alarmed when you see a message similar to the following appear on the console.

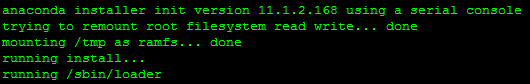


Figure – The “running install” message

*If you have followed the steps above correctly the OS will* ***not*** *be re-installed.*

1. When you see the message shown in figure 4 appear select **Continue** using the <*Tab*> key and press <*Return*>.

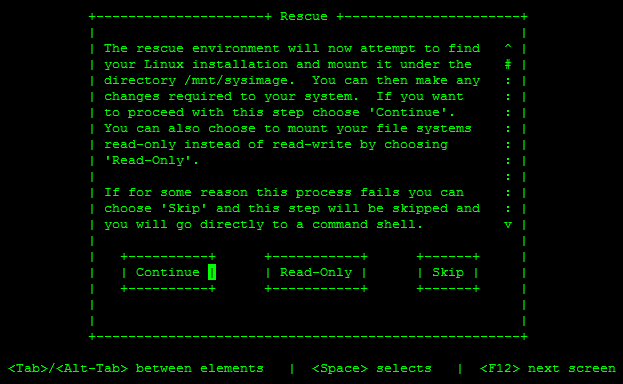


Figure – Mounting your Linux installation

1. Note the message shown and press <*Return*> to enter the shell, see figure 5 below.

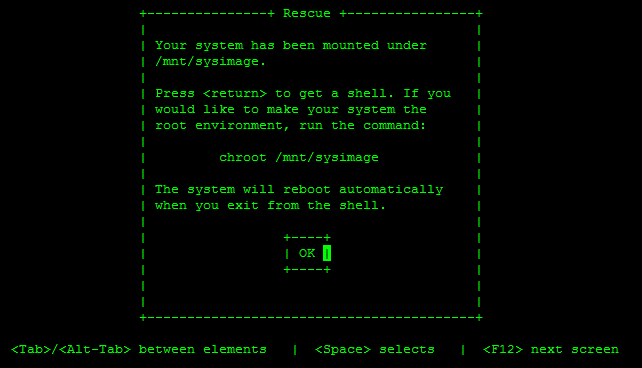
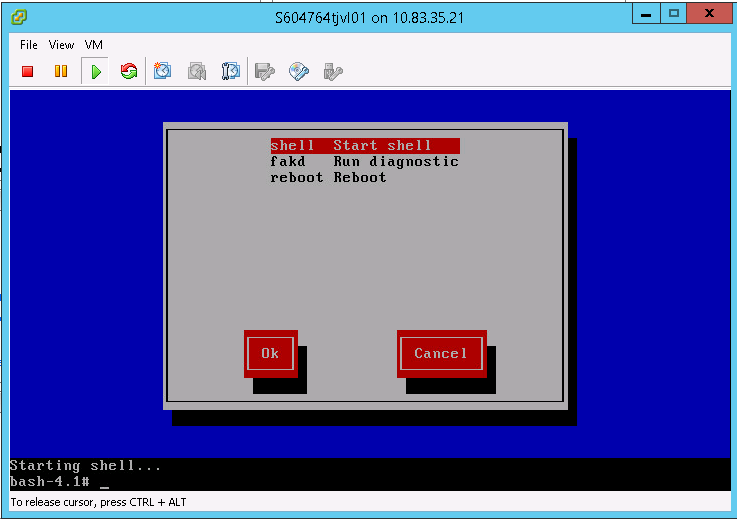
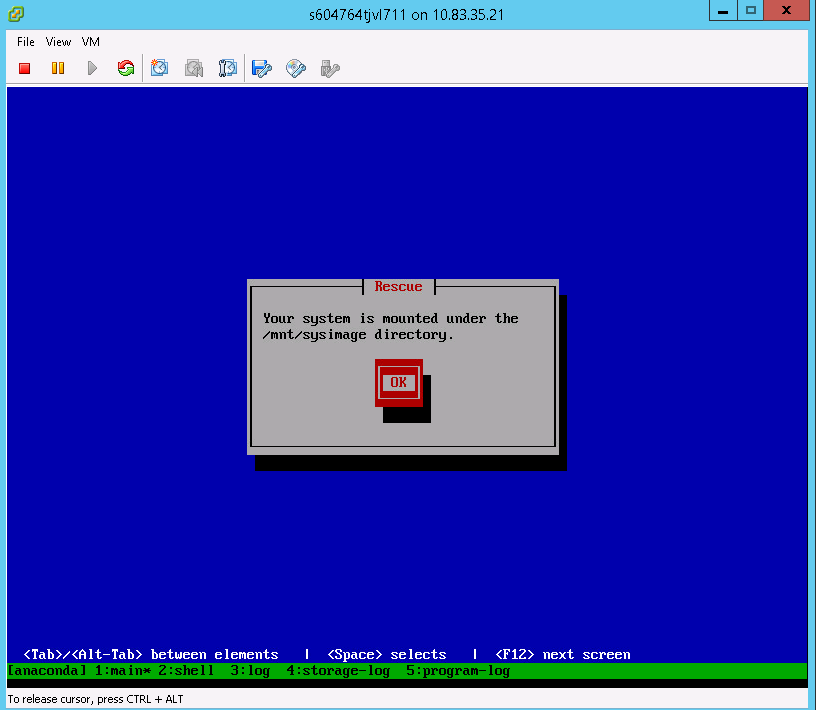


Figure – Entering the shell on HP Proliant



**Figure 6 – Entering the shell on a RHEL 6 Virtual Machine**



**Figure 7 – Entering the shell on a RHEL 7 Virtual Machine**

1. You should now see the root ( / ) file system mounted under /mnt/sysimage, similar to that shown in figure 7 below.

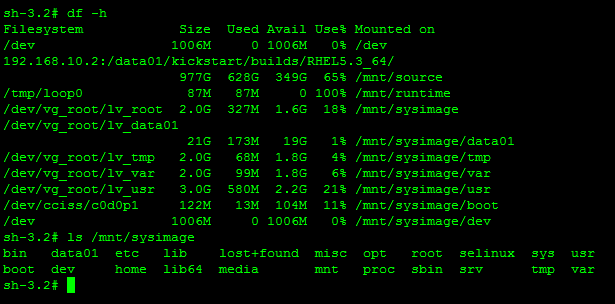


Figure – Accessing the root file system

1. Change to the root of the real file system.

# chroot /mnt/sysimage

1. Proceed as before to rest the password.

# Passwd root

1. Exit the chroot environment

# exit

Note: The Network interface will need to be restored to its original vlan, either by changing the Virtual Machine settings or by a member of the Networks team

Note: any BIOS settings that were changed, such as to allow for PXE boot, should be reverted prior to rebooting the server.

# Additional notes for VMware and SOS report

* Boot unresponsive server from the network as detailed in How to Boot Linux Rescue Mode from a Jumpkick Server above

Below are example configuration files for reference

**Working example /data01/jumpkick/clients/hostname/ks.cfg**

#SVVS|S265575ch3vl66t\_recovery|00-50-56-93-5a27|RHEL6.5\_64|vmware/{}|Americas|1447315168|svadmin

nfs --server 192.168.10.2 --dir /data01/kickstart/builds/RHEL6.5\_64/

network --noipv6

network --bootproto=dhcp

lang en\_US.UTF-8

keyboard us

timezone --utc Etc/UTC

**Working example /tftpboot/pxelinux.cfg/<mac address>**

default linux

serial 0,9600n8

label linux

kernel kickstart/vmlinuz-rhel6.5\_64

ipappend 2

append ramdisk\_size=16384 load\_ramdisk=1 edd=off initrd=kickstart/initrd-rhel6.5\_64.img **text rescue** ks=http://192.168.10.2/clients/S

265575ch3vl66t\_recovery/ks.cfg ksdevice=bootif

* Run the sosreport, this is for a Linux machine booted from the recovery media.

# chroot /mnt/sysimage

# sosreport

The files will be written to /tmp (this will become /mnt/sysimage/tmp )

* Once the sosreport has completed exit the chroot env

# exit

Now we need to be able to get the sosreport onto another working server. One way to do this is by using nfs on the jumpkick server (the O/S was booted via nfs, networking is active and in a private vlan with an IP address assigned by the jumpkick server).

* On the kickstart server become root and then temporarily export an additional share e.g. /tmp

#share -F nfs -o anon=0 /tmp

* Back on the "broken" system, mount the newly created nfs export

#mkdir /mnt2

#mount -t nfs 192.168.10.2:/tmp /mnt2

Note: the IP address of the jumpkick server is the same in all data centres.

* Now we can copy the files over**...**

# cp /mnt/sysimage/tmp/sosreport\* /mnt2

* Having confirmed that this worked unmount the nfs mount point

# umount /mnt2

* **Do not forget to “unshare” /tmp from the jumpkick server**

# unshare /tmp

# showmount -e localhost

export list for localhost:

/data01 (everyone)