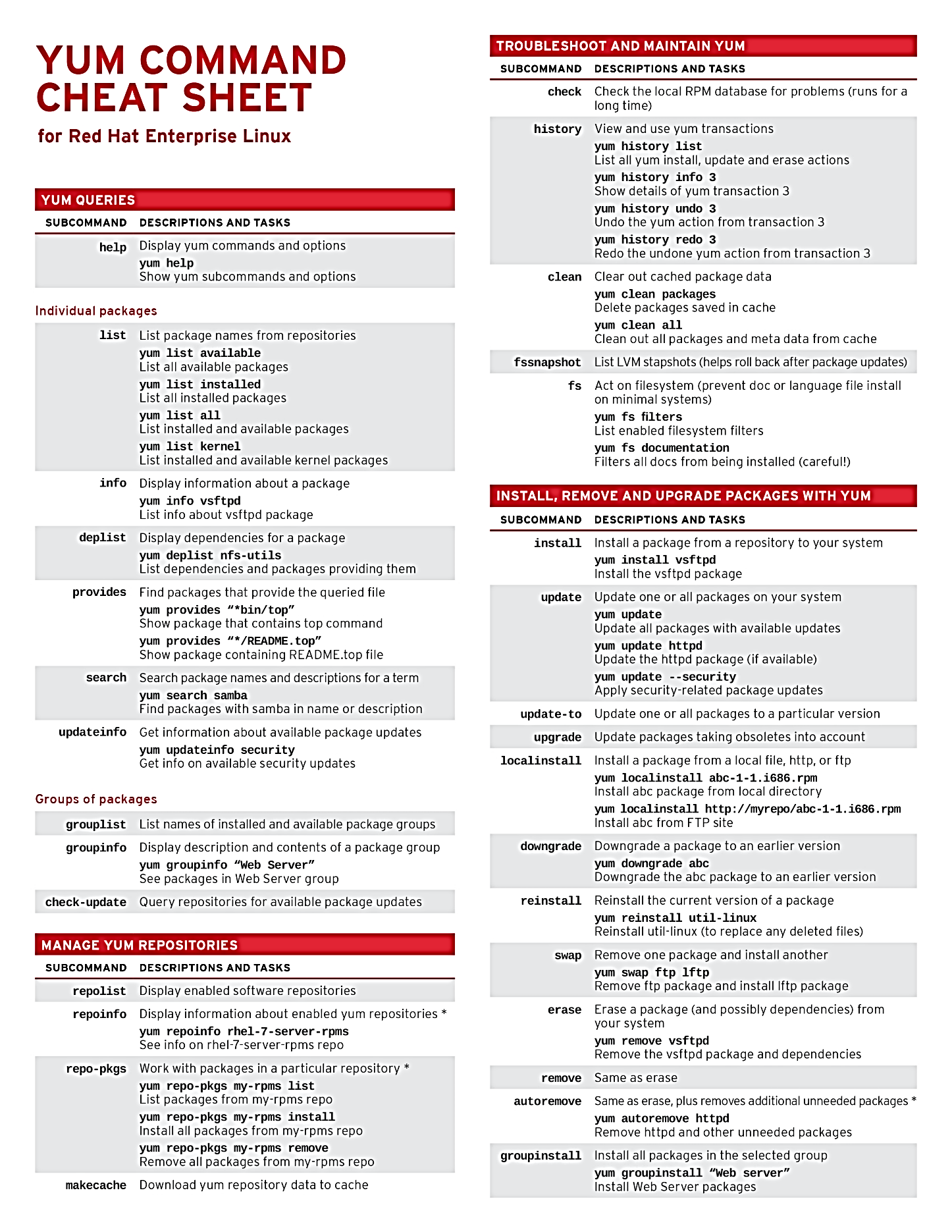
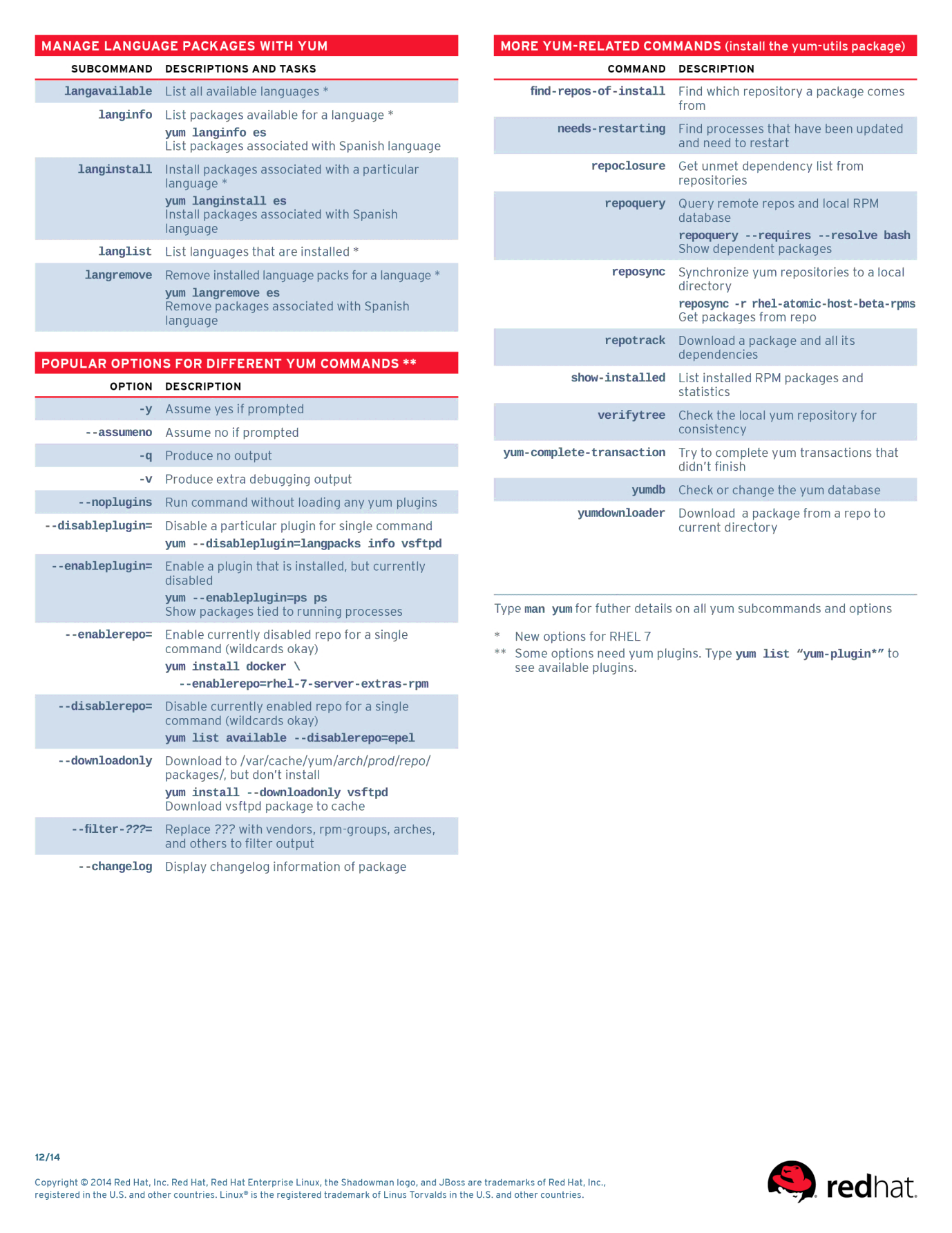
|  |  |
| --- | --- |
| 1 – SYSTEM INFORMATION  # Display Linux system information  uname -a  # Display kernel release information  uname -r  # Show which version of redhat installed  cat /etc/redhat-release  # Show how long the system has been running + load  uptime  # Show system host name  hostname  # Display the IP addresses of the host  hostname -I  # Show system reboot history  last reboot  # Show the current date and time  date  # Show this month's calendar  cal  # Display who is online  w  # Who you are logged in as  Whoami  2 – HARDWARE INFORMATION  # Display messages in kernel ring buffer  dmesg  # Display CPU information  cat /proc/cpuinfo  # Display memory information  cat /proc/meminfo  # Display free and used memory ( -h for human readable, -m for MB, -g for GB.)  free -h  # Display PCI devices  lspci -tv  # Display USB devices  lsusb -tv  # Display DMI/SMBIOS (hardware info) from the BIOS  dmidecode  # Show info about disk sda  hdparm -i /dev/sda  # Perform a read speed test on disk sda  hdparm -tT /dev/sda  # Test for unreadable blocks on disk sda  badblocks -s /dev/sda | 3 – PERFORMANCE MONITORING AND STATISTICS  # Display and manage the top processes  top  # Interactive process viewer (top alternative)  htop  # Display processor related statistics  mpstat 1  # Display virtual memory statistics  vmstat 1  # Display I/O statistics  iostat 1  # Display the last 100 syslog messages  (Use /var/log/syslog for Debian based systems.)  tail 100 /var/log/messages  # Capture and display all packets on interface eth0  tcpdump -i eth0  # Monitor all traffic on port 80 ( HTTP )  tcpdump -i eth0 'port 80'  # List all open files on the system  lsof  # List files opened by user  lsof -u user  # Display free and used memory ( -h for human readable, -m for MB, -g for GB.)  free -h  # Execute "df -h", showing periodic updates  watch df –h  4 – USER INFORMATION AND MANAGEMENT  # Display the user and group ids of your current user.  id  # Display the last users who have logged onto the system.  last  # Show who is logged into the system.  who  # Show who is logged in and what they are doing.  w  # Create a group named "test".  groupadd test  # Create an account named john, with a comment of "John Smith" and create the user's home directory.  useradd -c "John Smith" -m john  # Delete the john account.  userdel john  # Add the john account to the sales group  usermod -aG sales john |
| 5 – FILE AND DIRECTORY COMMANDS  # List all files in a long listing (detailed) format  ls -al  # Display the present working directory  pwd  # Create a directory  mkdir directory  # Remove (delete) file  rm file  # Remove the directory and its contents recursively  rm -r directory  # Force removal of file without prompting for confirmation  rm -f file  # Forcefully remove directory recursively  rm -rf directory  # Copy file1 to file2  cp file1 file2  # Copy source\_directory recursively to destination. If destination exists, copy source\_directory into destination, otherwise create destination with the contents of source\_directory.  cp -r source\_directory destination  # Rename or move file1 to file2. If file2 is an existing directory, move file1 into directory file2  mv file1 file2  # Create symbolic link to linkname  ln -s /path/to/file linkname  # Create an empty file or update the access and modification times of file.  touch file  # View the contents of file  cat file  # Browse through a text file  less file  # Display the first 10 lines of file  head file  # Display the last 10 lines of file  tail file  # Display the last 10 lines of file and "follow" the file as it grows.  tail -f file | 6 – PROCESS MANAGEMENT  # Display your currently running processes  ps  # Display all the currently running processes on the system.  ps -ef  # Display process information for processname  ps -ef | grep processname  # Display and manage the top processes  top  # Interactive process viewer (top alternative)  htop  # Kill process with process ID of pid  kill pid  # Kill all processes named processname  killall processname  # Start program in the background  program &  # Display stopped or background jobs  bg  # Brings the most recent background job to foreground  fg  # Brings job n to the foreground  fg n  7 – FILE PERMISSIONS  Linux chmod example          PERMISSION      EXAMPLE           U   G   W          rwx rwx rwx     chmod 777 filename          rwx rwx r-x     chmod 775 filename          rwx r-x r-x     chmod 755 filename          rw- rw- r--     chmod 664 filename          rw- r-- r--     chmod 644 filename  # NOTE: Use 777 sparingly!          LEGEND          U = User          G = Group          W = World          r = Read          w = write          x = execute          - = no access |

|  |  |
| --- | --- |
| 8 – NETWORKING  # Display all network interfaces and ip address  ifconfig -a  # Display eth0 address and details  ifconfig eth0  # Query or control network driver and hardware settings  ethtool eth0  # Send ICMP echo request to host  ping host  # Display whois information for domain  whois domain  # Display DNS information for domain  dig domain  # Reverse lookup of IP\_ADDRESS  dig -x IP\_ADDRESS  # Display DNS ip address for domain  host domain  # Display the network address of the host name.  hostname -i  # Display all local ip addresses  hostname -I  # Download http://domain.com/file  wget http://domain.com/file  # Display listening tcp and udp ports and corresponding programs  netstat -nutlp    9 – ARCHIVES (TAR FILES)  # Create tar named archive.tar containing directory.  tar cf archive.tar directory  # Extract the contents from archive.tar.  tar xf archive.tar  # Create a gzip compressed tar file name archive.tar.gz.  tar czf archive.tar.gz directory  # Extract a gzip compressed tar file.  tar xzf archive.tar.gz  # Create a tar file with bzip2 compression  tar cjf archive.tar.bz2 directory  # Extract a bzip2 compressed tar file.  tar xjf archive.tar.bz2  10 – INSTALLING PACKAGES  # Search for a package by keyword.  yum search keyword  # Install package.  yum install package  # Display description and summary information about package.  yum info package  # Display total disk usage off the current directory  du –sh  15 – DIRECTORY NAVIGATION  # To go up one level of the directory tree.  (Change into the parent directory.)  cd .. | # Install package from local file named package.rpm  rpm -i package.rpm  # Remove/uninstall package.  yum remove package  # Install software from source code.  tar zxvf sourcecode.tar.gz  cd sourcecode  ./configure  make  make install  11 – SEARCH  # Search for pattern in file  grep pattern file  # Search recursively for pattern in directory  grep -r pattern directory  # Find files and directories by name  locate name  # Find files in /home/john that start with "prefix".  find /home/john -name 'prefix\*'  # Find files larger than 100MB in /home  find /home -size +100M  12 – SSH LOGINS  # Connect to host as your local username.  ssh host  # Connect to host as user  ssh user@host  # Connect to host using port  ssh -p port user@host  13 – FILE TRANSFERS  # Secure copy file.txt to the /tmp folder on server  scp file.txt server:/tmp  # Copy \*.html files from server to the local /tmp folder.  scp server:/var/www/\*.html /tmp  # Copy all files and directories recursively from server to the current system's /tmp folder.  scp -r server:/var/www /tmp  # Synchronize /home to /backups/home  rsync -a /home /backups/  # Synchronize files/directories between the local and remote system with compression enabled  rsync -avz /home server:/backups/  14 – DISK USAGE  # Show free and used space on mounted filesystems  df -h  # Show free and used inodes on mounted filesystems  df -i  # Go to the $HOME directory  cd  # Change to the /etc directory  cd /etc  # Display disks partitions sizes and types  fdisk -l  # Display disk usage for all files and directories in human readable format  du –ah  16- The proc Filesystem  The proc "file system" is a directory that contains a hierarchy of files that represent the current state  of the Linux kernel.  It allows applications and users to see the kernel's view of the system.  The proc directory also contains information about the hardware of the system, and any currently running processes.  Most of these files are read-only, but some files (primarily those in /proc/sys) can be manipulated by users and applications to communicate configuration changes to the kernel. |





Kernel and kdump

Kdump reserves a small portion of the memory for the secondary kernel called crashkernel. This secondary or crash kernel is used to capture the core dump image whenever the system crashes.

Kdump is an utility used to capture the system core dump in the event of system crashes.These captured core dumps can be used later to analyze the exact cause of the system failure and implement the necessary fix to prevent the crashes in future.

Kdump is a kernel crash dumping mechanism and is very reliable because the crash dump is captured from the context of a freshly booted kernel and not from the context of the crashed kernel. Kdump uses kexec to boot into a second kernel whenever system crashes. This second kernel, often called the crash kernel, boots with very little memory and captures the dump image.

The first kernel reserves a section of memory that the second kernel uses to boot. Kexec enables booting the capture kernel without going through the BIOS, so contents of the first kernel's memory are preserved, which is essentially the kernel crash dump.

How to Use Kdump

**Step 1: Configuring Kdump**

a)First, install the kexec-tools, crash and kernel-debuginfo packages. Use following command line to install the packages.

NOTE: The `crash` and `kernel-debuginfo` packages are only required if you are planning on looking at the resulting kernel vmcore yourself. Most often this is the case, however if you are setting up kdump on a machine simply to capture a vmcore that will be analyzed by someone else or on a different machine, you can skip those packages.

b)Next, edit /boot/grub/grub.conf and add the "crashkernel=128M" command line option. Ex:-

linux /vmlinuz-3.1.4-1.fc16.x86\_64 ro root=/dev/VolGroup00/LogVol00 rhgb LANG=en\_US.UTF-8 crashkernel=128M

It is recommended to use minimum of 128M for a machine with 2G memory or higher.

c)Next, consider editing the kdump configuration file /etc/kdump.conf. This will allow you to write the dump over the network or to some other location on the local system, rather than to the default location of /var/crash. For additional information, consult the mkdumprd man page and the comments in /etc/kdump.conf.

d)Next, reboot your system .Finally, active the kdump system service

# service kdump restart

Stopping kdump: [ OK ]

Starting kdump: [ OK ]

# service kdump status

Kdump is operational

**Considerations:**

Above shown parameter reserves 128MB of physical memory. This reserved memory is used to preload and run the capture kernel. Init scripts take care of pre-loading the capture kernel at system boot time.

It is recommended to either set up a serial console or switch to run level 3 (init 3) for testing purposes. The reason being that kdump does not reset the console if you are in X or framebuffer mode, and no message might be visible on console after system crash. You may also see screen corruption in graphics mode during capture.

Capturing a crash dump can take a long time, especially if the system has a lot of memory. Be patient. The system will reboot after the dump is captured.

**Step 2: Capturing the Dump**

Normally kernel panic() will trigger booting into capture kernel but for testing purposes one can simulate the trigger in one of the following ways.

1. Enable [SysRq](https://fedoraproject.org/wiki/QA/Sysrq) then trigger a panic through /proc interface

* echo 1 > /proc/sys/kernel/sysrq
* echo c > /proc/sysrq-trigger

1. Trigger by inserting a module which calls panic().

The system will boot into the capture kernel. A kernel dump will be automatically saved in /var/crash/<dumpdir> and the system will boot back into the regular kernel. The name of the dump directory will depend on date and time of crash. For example, /var/crash/2017-10-13-13:02/vmcore.

**Step 3: Dump Analysis**

Once the system has returned from recovering the crash, you may wish to analyse the kernel dump file using the crash tool.

1. First, locate the recent vmcore dump file:

*find /var/crash -type f -mtime -1*

1. One you have located a vmcore dump file, call crash:

*crash /*/var/crash/2017-10-13-13:02/vmcore */usr/lib/debug/lib/modules/`uname -r`/vmlinux*

sosreport:

The sosreport command is a tool that collects configuration details, system information and diagnostic information from a Red Hat Enterprise Linux system. For instance: the running kernel version, loaded modules, and system and service configuration files. The command also runs external programs to collect further information, and stores this output in the resulting archive.

The output of sosreport is the common starting point for Red Hat support engineers when performing an initial analysis of a service request for a Red Hat Enterprise Linux system.

**Installing sosreport:**

To run sosreport the sos package must be installed. The package is part of the default group and will be installed automatically on most systems.

*# yum install sos*

**How to run sosreport:**

Once the sos package has been [installed](https://access.redhat.com/solutions/3592#install), issue the following command to run sosreport: **Note**: sosreport requires **root** permissions to run.

*# sosreport*

The command will normally complete within a few minutes on Red Hat Enterprise Linux 6. Older versions may take longer to complete. Depending on local configuration and the options specified in some cases the command may take longer to finish.

Once completed, sosreport will generate a compressed a file under /tmp (for RHEL6 and earlier) or under /var/tmp (for RHEL7 and later). Different versions use different compression schemes (gz, bz2, or xz). The file should be provided to your support representative (normally as an attachment to an open support case).

The size of the archive varies depending on system configuration and any optional sosreport features that are enabled (for example specifying the "all\_logs" option of the general module to collect all syslog log files may greatly increase the size of the archive).

To post sosreport, or any other file, to a existing support case you can use the [redhat-support-tool](https://access.redhat.com/solutions/2112#supporttool) command line option, the [Red Hat Portal UI](https://access.redhat.com/solutions/2112#customerportal) or several different methods using [FTP](https://access.redhat.com/solutions/2112).

If the collected *sosreport* file is too big to upload to the case, it could be uploaded to the [Red Hat ftp *dropbox.redhat.com* site](https://access.redhat.com/site/solutions/2112). The sosreport command has a modular structure and allows the user to enable and disable modules and specify module options via the command line. To list available modules (plug-ins) use the following command:

# sosreport -l

Jumbo Frame:-

A jumbo frame is an Ethernet frame with a payload greater than the standard maximum transmission unit (MTU) of 1,500 bytes. Jumbo frames are used on local area networks that support at least 1 Gbps and can be as large as 9,000 bytes.

This can improve the performance. First, make sure your network driver supports custom MTU. Second you need to have a compatible gigabit NIC and switch.Jumbo frames can reduce server overhead such as a big decrease in CPU usage when transferring larger file. Also you should see some increase in network throughput.

**Configure eth0 interface for Jumbo Frames:-**

Simply type the following command at a shell prompt to set new MTU (Jumbo Frames):

*# ifconfig eth0 mtu 9000*

**Make changes permanent**

Edit the network configuration file for eth0 interface – for example, /etc/sysconfig/network-script/ifcfg-eth0 (CentOS / RHEL / Fedora Linux):

*# vi /etc/sysconfig/network-script/ifcfg-eth0*

Append the following configuration directive, which specifies the size of the frame in bytes:

*MTU 9000*

To confirm the MTU used between two specific devices. use ip command as follows:

*# ip route get {IP-address}*

*# ip route get 192.168.1.1*

Output:

*192.168.1.1 dev eth0 src 192.168.1.100*

*cache mtu 9000 advmss 1460 hoplimit 64*

tcpdump

tcpdump is a package analyzer tool which is used to capture or filter TCP/IP packets that received or transferred over a network on a specific interface. Tcpdump also gives us a option to save captured packets in a file for future analysis. It saves the file in a pcap format, that can be viewed by tcpdump command.Many of Linux distributions already shipped with tcpdump tool, if in case you don’t have it on systems, you can install it using following Yum command.

*# yum install tcpdump*

Examples:

1. Capture Packets from Specific Interface

The command screen will scroll up until you interrupt and when we execute tcpdump command it will captures from all the interfaces, however with -i switch only capture from desire interface.

*# tcpdump -i eth0*

2. Capture Only N Number of Packets

When you run tcpdump command it will capture all the packets for specified interface, until you Hit cancel button. But using -c option, you can capture specified number of packets. The below example will only capture 6 packets.

*# tcpdump -c 5 -i eth0*

3. Print Captured Packets in ASCII

The below tcpdump command with option -A displays the package in ASCII format. It is a character-encoding scheme format.

*# tcpdump -A -i eth0*

4. Display Available Interfaces

To list number of available interfaces on the system, run the following command with -D option.

*# tcpdump -D*

5. Display Captured Packets in HEX and ASCII

The following command with option -XX capture the data of each packet, including its link level header in HEX and ASCII format.

*# tcpdump -XX -i eth0*

6. Capture and Save Packets in a File

As we said, that tcpdump has a feature to capture and save the file in a .pcap format, to do this just execute command with -w option.

*# tcpdump -w 0001.pcap -i eth0*

7. Read Captured Packets File

To read and analyze captured packet 0001.pcap file use the command with -r option, as shown below.

*# tcpdump -r 0001.pcap*

8. Capture IP address Packets

To capture packets for a specific interface, run the following command with option -n.

*# tcpdump -n -i eth0*

9. Capture only TCP Packets.

To capture packets based on TCP port, run the following command with option tcp.

*# tcpdump -i eth0 tcp*

10. Capture Packet from Specific Port

Let’s say you want to capture packets for specific port 22, execute the below command by specifying port number 22 as shown below.

*# tcpdump -i eth0 port 22*

11. Capture Packets from source IP

To capture packets from source IP, say you want to capture packets for 192.168.0.2, use the command as follows.

*# tcpdump -i eth0 src 192.168.0.2*

12. Capture Packets from destination IP

To capture packets from destination IP, say you want to capture packets for 50.116.66.139, use the command as follows.

*# tcpdump -i eth0 dst 50.116.66.139*

Nmap:-

The Nmap or Network Mapperis used for exploring networks, perform security scans, network audit and finding open ports on remote machine. It scans for Live hosts, Operating systems, packet filters and open ports running on remote hosts.

Examples:-

1. Scan a System with Hostname and IP Address

*nmap server10.sanjeevi.net*

*or*

*nmap 192.168.1.60*

*or*

*nmap -v server10.sanjeevi.net*

*where -v is for detailed output*

2. You use multiple hosts or multiple ipaddresses along with nmap command

*nmap server10 server9 server8*

3.You can scan a whole subnet or IP range with Nmap by providing \* wildcard with it

*nmap 192.168.1.\**

*or*

Scan Multiple Servers using last octet of IP address

*nmap 192.168.1.60,61,62,64,70*

*or*

Scan an IP Address Range

*nmap 192.168.1.51-71*

4.Scan list of Hosts from a File

*nmap -iL /path/to/file\_withHostnames*

5.Scan Network Excluding Remote Hosts

*nmap 192.168.1.\* --exclude 192.168.1.50*