PROYECTOS DE ENAIRE (SACTA, COMETA e ICARO)

RESUMEN PROCEDIMIENTOS XR12

(IT)

(Procedimientos XR12)

Nº Exp.: xxx

Doc. Nº: xxx

Ed./Rv.: 1

Fecha: 23/07/24

Nº LDEC: xxx

Preparado para:

Enaire - División de Automatización
Parque Empresarial Las Mercedes
Edificio 7 – Planta baja
Avda. de Aragón, 330
28022 - Madrid

Preparado por:

Indra Sistemas
Parque Empresarial San Fernando (Edificio Kenia)
Avenida de Castilla, 2
28830 San Fernando de Henares
Madrid

ÍNDICE

1. INTRODUCCIÓN

1.1. OBJETIVO: DESPLIEGUE DE MUEBLES IFOCUS

Tal como muestra la figura 1.1-1, los muebles iFocus son controlados desde ordenadores XR12 (dos para planificador, dos para ejecutivo) que contienen las aplicaciones de tráfico aéreo encapsuladas dentro de máquinas virtuales. En otras palabras, el ordenador iFocus son varias máquinas anfitrionas XR12 especializadas en gestionar instancias de máquinas virtuales que encapsulan cada aplicación de tráfico aéreo, con todas sus dependencias (sistema operativo y librerías).

Como indica la imagen, consola y ordenador están en distintas salas, conectadas a través de cables de fibra óptica. Detrás de cada mueble iFOCUS, hay un armario con agregadores que multiplexan las señales de las distintas conexiones que viene del mueble sobre una misma fibra óptica. En la sala de cómputo, hay un armario por mueble iFOCUS que tiene sus cuatro XR12 en la parte inferior y los disgregadores en la parte superior que demultiplexan esas señales de la fibra óptica para llevarlas a los ordenadores XR12.

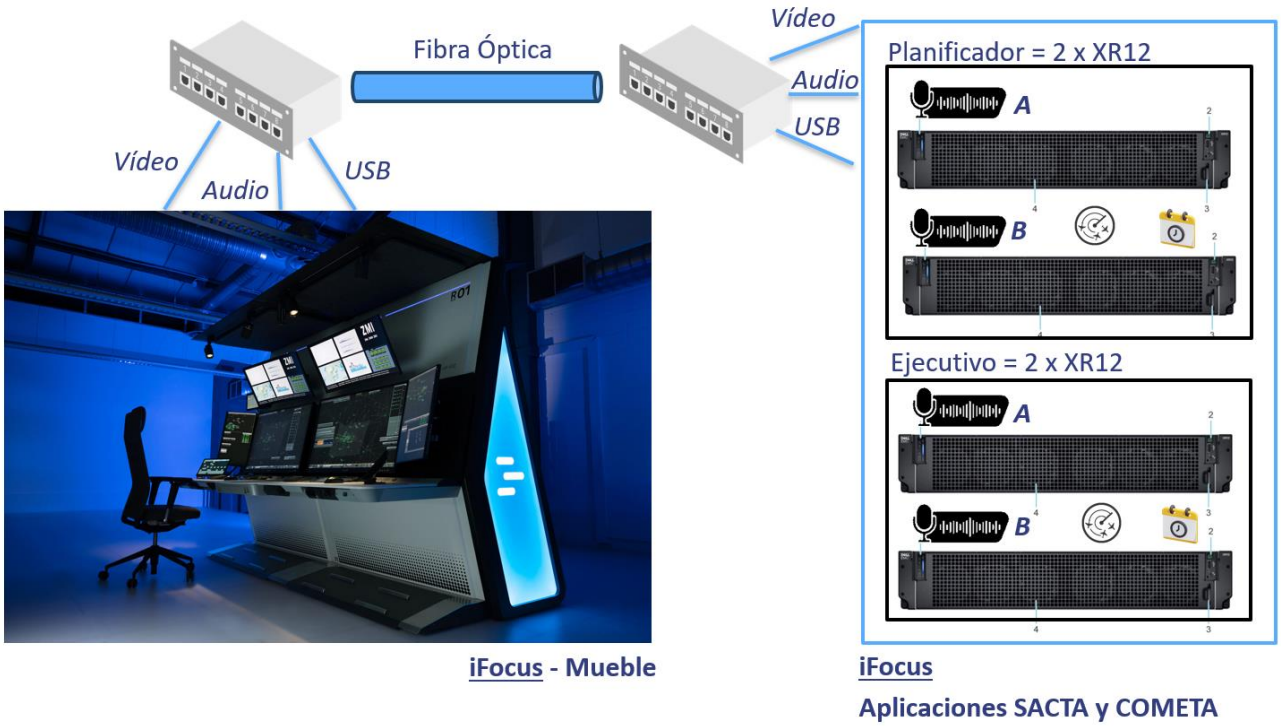


Figura 1.1-1. Esquema Simplificado de los Muebles iFocus

Tal como muestra la figura 1.1-2, Enaire libera versiones tanto de aplicativo como de máquina virtual, **los centros de control son responsables de realizar un proceso de puesta en marcha**, es decir, inyectarles aplicación a las máquinas virtuales, **y a partir de ahí, propagarlas por todos los muebles** que gestione el centro de control. Has tres tipos de chasis virtual, cada uno con su versión de RedHat Enterprise Linux (RHEL): RHEL6 para POS, RHEL7 para OUCS y RHEL8 para CWP. El anfitrión trabaja en RHEL 8.6

Ha de llevarse un control de versiones de las instancias de máquina virtual que están ejecutándose en cada mueble iFocus del centro.

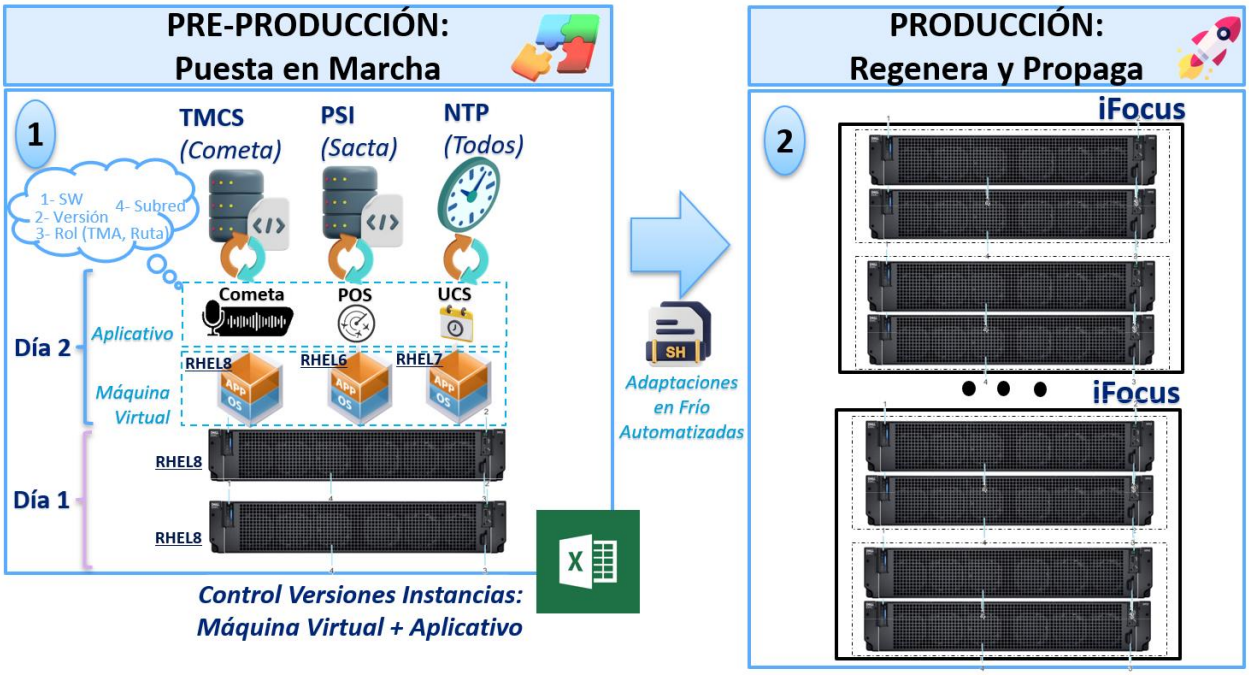


Figura 1.1-2. Proceso de Despliegue de los Muebles iFocus

1.2. TAREAS DE MANTENIMIENTO DE MUEBLES IFOCUS

Dos son las tareas de mantenimiento asociadas al mueble iFOCUS:

- TAREA 1 – CONTROL DEL MUEBLE IFOCUS: consiste en automatizar los diagnósticos de los periféricos para hacer verificaciones periódicas de su estado e ir clasificando y guardando los distintos errores que vayan surgiendo, o sea, ir creando una base de datos errores del mueble iFOCUS clasificada de tal forma que sea fácil encontrar cómo resolver cada caso (ejemplo de cómo clasificar errores: por periférico y por puntos de sondeo dentro de su cadena de eventos, identificando planes de acción para cada error). Clave es protocolizar los diagnósticos, tanto para su automatización como para la apertura de casos a través de los cuales ir creando esa base de datos de errores, diagnósticos y resoluciones para el mueble iFOCUS fácil de consultar (RedHat tiene la suya, pero es genérica).

- TAREA 2 – MANTENIMIENTO DE INSTANCIAS DE MÁQUINA VIRTUAL: consiste en tener un repositorio de máquinas virtuales con aplicativo instalado (tanto TMA como Ruta), listas para ser propagadas. Además, hay que tener controlados los parámetros de instancia en cada posición de cada mueble iFOCUS. Esos parámetros de instancia son los que permiten la instanciación automatizada de esas máquinas virtuales de referencia en cada mueble del centro de control (las instancias no guardan datos, sino que se sincronizan con servidores externos). Clave es la gestión de versiones, tanto el repositorio de máquinas virtuales de referencia, como parámetros instancia en cada mueble; tal vez un sistema de control de versiones pueda facilitar estas tareas de mantenimiento.

2. HERRAMIENTAS DE ANÁLISIS LINUX: CREANDO PROTOCOLOS DE DIAGNÓSTICO

2.1. ÍNDICE GRÁFICO.

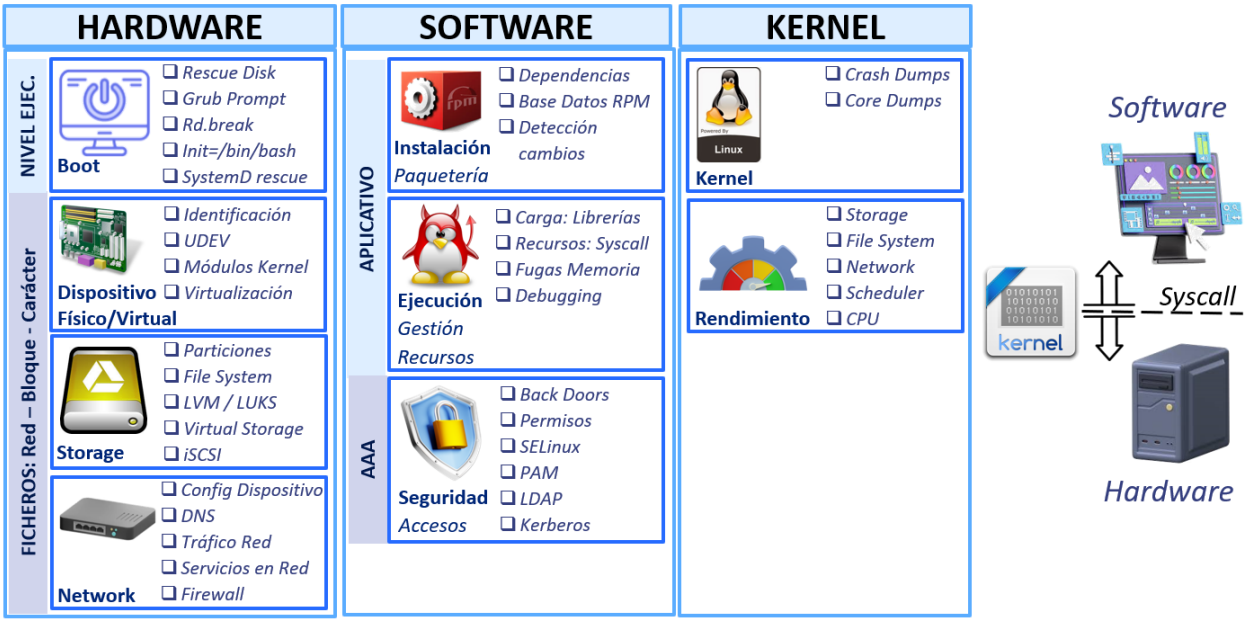


Figura 2.1-1. Herramientas de Análisis según establece RedHat en sus planes formativos (EX342).

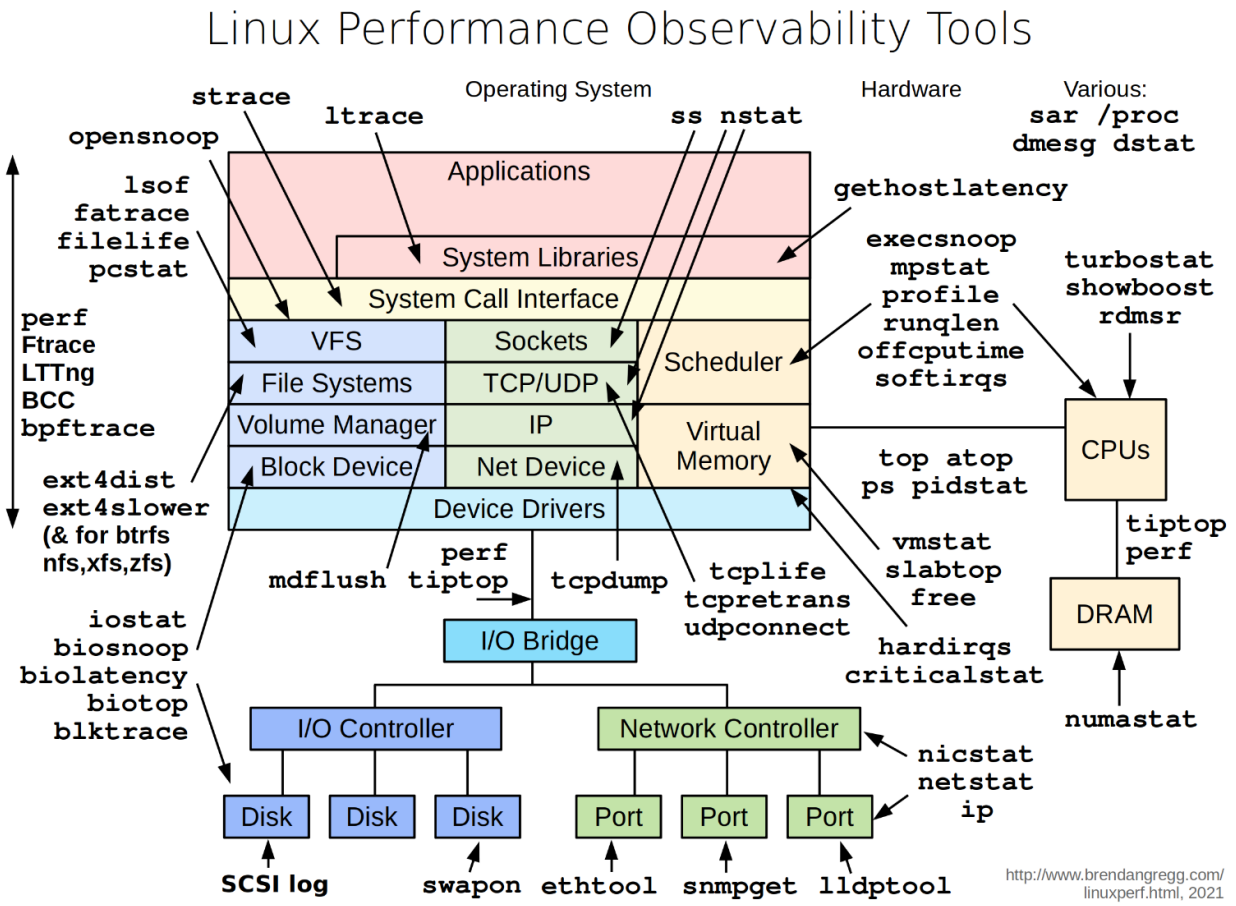


Figura 2.1-2. Herramientas de Análisis de Rendimiento según *brendangregg.com*.

2.2. HERRAMIENTAS COMUNES.

Tabla 2.2-1: Herramientas de Análisis Comunes.

COMUNES:

Problemas Generales

Journal

Journal Persistente

```
mkdir /var/log/journal
chown root:systemd-journal /var/log/journal
chmod 2755 /var/log/journal
killall -USR1 systemd-journal                    # or reboot
```

Ejemplos Journal Comunes

```
journalctl -ef                                   # end & follow
journalctl _SYSTEMD_UNIT=sshd.service            # generated by sshd service
journalctl -u sshd.service                       # generated and about sshd service
journalctl -p emerg..err                         # priority between emergency and error
journalctl -b -1                                 # only from the last boot
journalctl --since "2019-01-01 20:30:00" --until "2019-02-02 12:00:00"
journalctl -o verbose                            # show all fields
```

Herramientas

Recursos RedHat – Apertura Casos

```
yum -y install sos
sosreport --help | less
sosreport -l | less       # view currently enabled/disabled plugins and plugin options
sosreport -o <PLUGIN(S)> # enable these plugins only (it will only run these plugins)
sosreport -n <PLUGIN(S)> # skip plugin (it will run all of the plugins, except for these)
sosreport -e <PLUGIN(S)> # enable previously disabled plugins
sosreport -k xfs.logprint  # xfs module and logprint option enabled
redhat-support-tool
```

Insights

```
yum -y install redhat-access-insights
redhat-access-insights --register
```

Cockpit (Interfaz de Web de Control)

```
yum -y install cockpit
systemctl start cockpit
firewall-cmd -add-service=cockpit --permanent
firewall-cmd --reload                            # http://localhost:9090
```

Co-Pilot (Análisis de Rendimiento)

```
yum -y install pcp            # performance co-pilot
systemctl start pmcd          # performance metrics collector daemon
systemctl enable pmcd
pmstat -s 5                   # 5 samples
pmatop                        # machine stats and data
pminfo                        # obtain list of metrics
pminfo -dt proc.nprocs        # understand specific metric
pmval -s 5 proc.nprocs        # gather sample data about the metric 5x times
pmval -T 1minute kernel.percpu.cpu.idle          # per-CPU idle time for one minute
```

Datos Históricos

```
systemctl start pmlogger     # ability to store metrics data to logs (-a <ARCHIVE>)
systemctl enable pmlogger
pcp | grep 'primary logger' # location of the log that pmlogger is writing to
ls /var/log/pcp/pmlogger/<HOSTNAME> # collects data every second to this location
pmval -a <ARCHIVE.xz> -f 3 <METRIC> # metrics dump from archive 3 digits precision
pmval -a /var/log/pcp/pmlogger/serverX.example.com/20190101.00.10.0 kernel.all.load
pmval -a  /var/log/pcp/pmlogger/serverX.example.com/20190101.00.10.0 kernel.all.load \
  -S '@ Tue Feb 01 12:00:00 2019' -T '@ Tue Feb 01 13:00:00 2019'
```

Intrusion Detection

```
yum -y install aide                # intrusion detection
vim /etc/aide.conf
  # Configuration lines:
  PERMS = p+i+u+g+acl+selinux  # file (p)ermissions,(i)node,(u)ser/(g)roup, acl, selinux

  # Selection lines:
  /dir1 PERMS                  # group check on dir1 and all files and dirs below it
  =/dir2 PERMS                 # group check in dir2, but not recursively
  !/dir3                       # excludes dir3 and all files below it from any checks

  # Macro lines:
  @@define VAR value           # @@{VAR} is reference to the macro defined previously
aide --init                   # creates /var/lib/aide/aide.db.new.gz every time
mv -v /var/lib/aide/aide.db.new.gz /var/lib/aide/aide.db.gz
aide -check
```

Gestión Centralizada Logs

Registro Central Logs

```
systemctl is-active rsyslog
systemctl is-enabled rsyslog
man rsyslog.conf          # man pages
vim /etc/rsyslog.conf
  $ModLoad imudp.so        # for UDP
  $UDPServerRun 514

  $ModLoad imtcp.so        # for TCP
  $InputTCPServerRun 514

  $template DynamicFile,"/var/log/loghost/%HOSTNAME%/cron.log"
  cron.* ?DynamicFile      # 'DyamicFile' here is arbitrary template name

  $template DynamicFile,"/var/log/loghost/%HOSTNAME%/%syslogfacility-text%.log"
  *.* -?DynamicFile         # minus is turn off syncing of the log file after each write
systemctl restart rsyslog
firewall-cmd --add-port=514/udp --permanent
firewall-cmd --add-port=514/tcp --permanent
firewall-cmd --reload

# Enable log rotation:
vim /etc/logrotate.d/syslog
  /var/log/loghost/*/*.log   # must not be at the end, but before curly braces {...}
```

Redirección Logs al Registro Central

```
vim /etc/rsyslog.conf
  *.info @loghost.example.com[:PORT] # all info msg to loghost.example.com via UDP
  *.* @@loghost.example.com          # all msg to loghost.example.com via TCP
```

Auditoría Ficheros

```
man audit.rules                          # manpages
auditctl -w /etc/passwd -p wa -k <KEY> # watch: write, attribute changes with key
auditctl -w /etc/sysconfig -p rwa -k <KEY> # recursive watch: all files and dirs
auditctl -w /bin -p x                    # all executions in bin
auditctl -W <PATH>                       # remove watch rule(s) at path
auditctl -d <RULE>                       # remove previous -a or -A rule(s)
auditctl -D                  # remove all rules (or they get removed by reboot)
vim /etc/audit/rules.d/audit.rules # persistent rules, without auditctl at the beginning
 -w /etc -p w -k etc_content # w=path, p=permission((r)ead,(w)rite,e(x)ecute,(a)ttribute)
 -w /etc -p a -k etc_attribute
```

Auditoría Llamadas al Sistema

```
auditctl -a always,exit -F arch=b64 -S open \   # list names:task,exit,user,exclude
          -F success=0                         #            actions:never,always
                                               # system call open() that failed
cat /var/log/audit/audit.log
ausearch -i --raw -a <EVENT-ID> --file <FILENAME>
         -k <KEY> --start <START-TIME> --end <END-TIME>
ausearch -k etc_content                          # audit search for specific key/tag
ausearch -k etc_attribute

# Some fields (-F):
```

- **auid=**The original ID the user logged in with. Its an abbreviation of audit uid. Sometimes its referred to as loginuid.
- **egid=**Effective Group ID.
- **euid=**Effective User ID.
- **sgid=**Saved Group ID.
- **suid=**Saved User ID.
- **uid=**User ID.

2.3. ANÁLISIS DEL HARDWARE.

2.3.1. Análisis del Arranque del Sistema.

Tabla 2.3-1: Análisis del Arranque del Sistema.

HARDWARE:

Arranque del Sistema

GRUB

Configuración GRUB

```
# Configuring Grub2:
vim /etc/default/grub
  GRUB_TIMEOUT = seconds the menu is displayed
  GRUB_DEFAULT = starts counting from 0, what is the default entry
  GRUB_CMDLINE_LINUX = list of extra kernel params, e.g "rhgb quiet"
grub2-mkconfig -o /boot/grub2/grub.cfg          # this is *.cfg and NOT *.conf
```

Entrada Menú GRUB

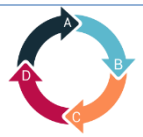
```
# An example of the 'linux16' menu entry line:
linux16 /vmlinuz-3.8.0-0.40.el7.x86_64 root=/dev/mapper/rhel-root ro rd.md=0 rd.dm=0
rd.lvm.lv=rhel/swap       crashkernel=auto       rd.luks=0       vconsole.keymap=us
rd.lvm.lv=rhel/root rhgb quiet
```

Reinstalación GRUB en MBR (se debe reiniciar en modo rescate, Anaconda)

```
chroot /mnt/sysimage
ls -l /boot
grub2-install /dev/vda                # rewrite the boot loader sections of the MBR
```

Reinstalación GRUB en UEFI

```
yum reinstall grub2-efi shim                        # if the files /boot/efi have been
removed
grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg # if the cfg file has been removed
yum -y install efibootmgr
efibootmgr                        # manage list of available UEFI boot targets
efibootmgr -b 1E -B               # delete entry 1E from UEFI boot targets completely
# Selecting a temporary boot target:
efibootmgr -n 2c                  # override normal boot ordering for a next single boot
# Add an entry (/dev/sda2 as the ESP with the application /EFI/yippie.efi on the ESP):
efibootmgr -c -d /dev/sda -p 2 -L "Yippie" -l "\EFI\yippie.efi"
                                                  # (c)reate,(d)evice,(p)artition,(L)abel
```



SYSTEMD

SystemD

```
# SystemD and failing services:
/etc/systemd/system/<UNITNAME>.d/            # symlinks to /usr/lib/systemd/system/
/etc/systemd/system/<UNITNAME>.requires/<SYMLINK>
/etc/systemd/system/<UNITNAME>.wants/<SYMLINK>
Requires=
    stopping listed unit will also stop this unit as well
RequiresOverridable=
    failed requirements will not cause the unit to fail when explicitly started
Requisite=,RequisiteOverridable=
    the unit will fail if required unit is not already running
After=
    listed units have to have finished starting before this unit can be started
Before=
    listed units will be delayed
Wants=
    when wante unit fails to start, this unit itself will still start
Conflicts=
    starting this unit will stop the conflicting units
systemctl daemon-reload                          # this is required after each change
systemctl list-dependencies <UNITNAME>
systemctl list-unit-files
systemctl status              # shows tree of services and corresponding PIDs
# To obtain a root shell during startup (/dev/tty9):
systemctl enable debug-shell.service # do not leave enabled after finished debugging!
systemctl list-jobs                     # troubleshoot startup tasks
```

Resetear Contraseña Root

```
# Resetting a root password:
1. Interrupt countdown
2. Press ["e"] on the highlighted entry    # changes made like this in the Grub2
menu screen are only temporary
3. Find kernel arguments line (starts with "linux16" or "linuxefi")
4. Add "rd.break" at the end of this line by pressing ["End"]
5. ["Ctrl"]+["X"]
6. mount -oremount,rw /sysroot
7. chroot /sysroot
8. echo <PASSWORD> | passwd --stdin root
9. touch ./autorelabel (or "load_policy -i; restorecon -Rv /etc")
10. ["Ctrl"]+["D"]
```

2.3.2. Análisis de Dispositivos Hardware.

Tabla 2.3-2: Análisis de Dispositivos Hardware.

HARDWARE: Dispositivos	
	Identificación de Dispositivos
	`lscpu` # identifying processor `cat /proc/cpuinfo` # identifying what flags CPU supports `dmidecode -t memory` # identifying memory `lsscsi -v` # identifying disks `hdparm -I /dev/sda` # more information about individual disks `lspci` # identifying PCI hardware `lsusb` # identifying USB hardware
	Errores de Memoria

	# Older:
	`yum -y install mcelog` # framework for catching and logging exceptions
	# Newer:
	`yum -y install rasdaemon` # modern replacement for mcelog
	`systemctl enable rasdaemon`
	`systemctl start rasdaemon`
	`ras-mc-ctl --status` # what does subsystem know about memory
	`ras-mc-ctl --errors`

	Pruebas de Memoria
	# Test memory: `yum -y install memtest86+` `memtest-setup` # this adds new template to Grub2 (/etc/grub.d/) `grub2-mkconfig -o /boot/grub2/grub.cfg` # update Grub2 config
	Módulos del Kernel
	`ls /lib/modules/<KERNEL_VERSION>/..` # all possible drivers `lsmod` # view currently loaded Kernel modules (same as /sys/module/..) `modprobe -v <MODULE>` # load the module manually `modprobe -r <MODULE>` # unload the module manually `modinfo -p <MODULE>` # list of supported options for the module `cat /sys/module/<MODULE>/parameters/<PARAMETER>` # active value of the module option `modprobe -v st buffer_kbs=64` # set option buffer_kbs for the st module to 64 when loaded `vim /etc/modprobe.d/00-st.conf` # automatically set every time, parsed alphabetically options st buffer_kbs=64 max_sg_segs=512 # permanent, needs unload & reload to take effect
	Soporte del Hardware de Virtualización
	`modprobe -v kvm-intel` # or 'kvm-amd' `virsh capabilities` # usually on the host, not guest
	Libvirt XML Config
	`virsh define <FILENAME.xml>` # attempt to create VM `xmllint --noout <FILENAME.xml>` # verify XML syntax `virt-xml-validate <FILENAME.xml>` # verify if it matches libvirt XML schema `/etc/libvirt/qemu/networks` # network definitions
	Consumo de Recursos

	`virsh nodecpustats`
	`virsh nodememstats`
	`virsh dommemstats <DOMAIN>`

2.3.3. Análisis del Almacenamiento.

Tabla 2.3-3: Análisis del Almacenamiento.

HARDWARE:

Almacenamiento

Mapeo de Dispositivos

```
dmsetup ls                    # list top level devices (e.g. VolGroup00-LogVol01 [253:1])
ls -l /dev/mapper/<LOGICAL_VOL> --> /dm-0 # symlink to device mapper#0 in /dev/<VG_NAME>
dmsetup table /dev/mapper/<LOGICAL_VOL>   # shows block device's minor/major number
ls -l /dev/vdb*                  # or lsblk -r | awk '{ print $1, $2 }', disks & partitions
yum -y install device-mapper-multipath
multipath -v 2 <DEVICE>          # device mapper target autoconfig
```

Scheduler I/O

```
cat /sys/block/<DEVICE>/queue/scheduler    # e.g. noop deadline [cfq]
yum -y install e2fsprogs
# e2fsck options:                     # 'df -TH' shows filesystems
e2fsck -b <LOCATION>                  # use alternative superblock
e2fsck -p                             # automatically repair, only prompt un-safe problems
e2fsck -v                             # verbose
e2fsck -y                             # non-interactive mode, answer yes to all
```

Recuperación Sistema de Ficheros Corrupto

```
# ext3/ext4:
umount /dev/<DEVICE>
e2fsck -n /dev/<DEVICE>                # dry-run (read-only + answer no to everything)
# If corrupt supeblock (bad magic number):  # magic number = where superblock starts
dumpe2fs /dev/<DEVICE> | grep 'Backup superblock'
e2fsck [-n] /dev/<DEVICE> -b <NUMBER>  # alternative superblock to use from previous cmd

# XFS:
yum -y install xfsprogs
umount /dev/<DEVICE>            # re-mount on systems where journal corruption suspected
xfs_repair -n /dev/<DEVICE>     # perform only check
xfs_repair [-o force_geometry] /dev/<DEVICE>     # perform all corrective actions, shows
invalid inodes
mount /dev/<DEVICE> /mountpoint
ls /mountpoint/lost+found       # unreferenced files
find /mountpoint -inum <NUMBER> # locate directory with the inode number
diff -s /file/from/backup /mountpoint/lost+found/<NUMBER>
# If corrupt journal log:
xfs_repair -L /dev/<DEVICE>     # zeros out the journal log, potentially dangerous
```

Recuperación de un Volumen Lógico (LVM)

```
# Config file:
vim /etc/lvm/lvm.conf
  dir                          # scan for physical volumes (/dev)
  obtain_device_list_from_udev # should udev be used (1)
  preferred_names              # which path name to display for block device
  filter                       # which devices to scan for presence of PV signature
  backup                       # save text-based metadata before each disk change (1)
  backup_dir                   # where the backup of VG metadata should be stored
  archive                      # should old configurations be also archived (1)
  archive_dir                  # where the archives will be stored
  retain_min                   # minimum number of archives to store
  retain_days                  # minimum number of days for archive to be kept
```

```
# Reverting LVM changes:
ls /etc/lvm/backup/<VG_NAME>
cat /etc/lvm/archive/<VG_NAME>_timestamp.vg | grep 'description ='
vgcfgrestore -l <VG_NAME>    # list descriptions of each archives of the volume group
umount ALL FS CREATED ON THE LOGICAL VOLUME
vgcfgrestore -f /etc/lvm/archive/<VG_NAME>_timestamp.vg
lvchange -an /dev/<VG_NAME>/<LV_NAME>  # activate no (deactivate)
lvchange -ay /dev/<VG_NAME>/<LV_NAME>  # activate yes (reactivate)
xfs_growfs /dev/<VG_NAME>/<LV_NAME>    # eventually grow the filesystem if needed
mount ALL FS CREATED ON THE LOGICAL VOLUME
```

LUKS

```
dmsetup ls --target crypt        # e.g. 'luks-0123456789-abcde-987654321-fghij (253,0)'
cat /etc/crypttab                # may contain UUID instead of ENCDEVICE
  <NAME> /dev/<ENCDEVICE> /path/keyfile or none # if none, you will be asked for
password on boot
/dev/mapper/<NAME>                    # decrypted device mapper location
cryptsetup luksDump /dev/<ENCDEVICE> # display LUKS header info. for encrypted device

# LUKS header backup:
cryptsetup luksHeaderBackup /dev/<ENCDEVICE> \
                    --header-backup-file /path/to/backup_file
# Trial decryption with header backup file:
#          use 'cryptsetup luksClose' when you make a typo
cryptsetup luksOpen /dev/<ENCDEVICE> <NAME> [--header /path/to/backup_file]
cryptsetup luksOpen <FILE.img> <NAME> --key-file <EXISTING_KEY_FILE.key>
# Add key to the key slot:
cryptsetup luksAddKey /dev/<ENCDEVICE> --key-file <EXISTING_KEY_FILE.key>
                      --key-slot <ID> [<key file with new key or pswd>]
# Restore header:
cryptsetup luksHeaderRestore /dev/<ENCDEVICE> \
                     --header-backup-file /path/to/backup_file
```

iSCSI Initiator/Client

```
yum -y install iscsi-initiator-utils # 'systemctl enable iscsi --now'
iscsiadm -m node                # see already discovered targets/node records
iscsiadm -m session [-P 3]      # validate sessions or connections, P=print level
vim /etc/iscsi/iscsid.conf       # restart iscsi/iscsid every time you change this file
  discovery.sendtargets.auth.<authmetod|username|password|username_in|password_in>
  node.session.auth.<authmetod|username|password|username_in|password_in>
vim /etc/iscsi/initiatorname.iscsi              # this needs iscsid restart
  InitiatorName=iqn.2016-01.com.example.lab:servera
systemctl restart iscsid
iscsiadm -m discovery -t st \
         -p <TARGET>:<PORT>  # discovery & sendtargets for portal -> /var/lib/iscsi/nodes
iscsiadm  -m  node  -T  iqn.2016-01.com.example.lab:iscsistorage  --login  [-d8]  #  -
d8=debug
# Disable CHAP authentication:
iscsiadm -m node -T iqn.2016-01.com.example.lab:iscsistorage \
         -o update -n node.session.auth.authmethod \
         -v None [-p <TARGET>:<PORT>]    # o=overwrite previous config,n=name,v=value
# Purge all node information from cache, recommended when server's setting change:
iscsiadm -m node -T iqn.2016-01.com.example.lab:iscsistorage \
         -o delete [-p <TARGET>:<PORT>]
# Purge all know nodes from cache:
iscsiadm -m node -o delete [-p <TARGET>:<PORT>] # default port 3260/tcp
lsblk --scsi
```

Troubleshooting

```
ip addr show dev <DEVICE_NAME>
ip route
nmcli con                                          # display connection information
nmcli dev                                          # display device information
nmcli conn show '<CONNECTION_NAME>' | grep ipv  # all config settings
  ipv4.method                          # auto=dhcp, manual=static (needs addresses,gateway)
  ipv6.method
ncmli conn mod '<CONNECTION_NAME>' ipv4.dns '<IPv4>' # good alternative: 'nmtui',
restart affected services
nmcli conn reload                    # after you manually edit network-scripts
nmcli conn down '<CONNECTION_NAME>'# changes are not applied to already active interface
nmcli conn up '<CONNECTION_NAME>'  # ...also updates /etc/resolv.conf
firewall-cmd --list-all-zones [--permanent] # compare active/permanent to identify
problems
firewall-cmd --runtime-to-permanent        # quick convert of runtime rules to permanent
host -v -t aaaa <HOSTNAME> <DNS>           # query DNS for hostname's IPv6
```

2.3.4. Análisis de las Comunicaciones.

Tabla 2.3-4: Análisis de las Comunicaciones.

HARDWARE:

Comunicaciones

Común

```
ping -c 1 -W 3 <IPv4>            # send single echo request and wait 3s for reply
ping6 [-I <INTERFACE>] <IPv6>   # -I is not needed when routable IPv6 is used
```

Escaneo de la Red

```
yum -y install nmap
nmap -n <IPv4>/<SUBNET>         # -n means don't use DNS, discover all ports on all hosts
nmap -n -sn <IPv4>/<SUBNET>     # -sn means disable port scanning, only discover hosts
nmap -n -sU <IPv4>              # perform UDP scan on the host
nmap <HOSTNAME>                 # perform IPv4 port scan on hostname
nmap -6 <HOSTNAME>              # scan ports of the IPv6 address
```

Prueba de Servicios Activos en los Nodos

```
yum -y install nmap-ncat
nc <HOSTNAME> <PORT>            # client/connect mode
nc -6 <HOSTNAME> <PORT>         # connect to port of hostname using IPv6
nc -l [-k] <PORT>               # server mode, -k means keep listening for >1 connections
nc -l <PORT> -e <COMMAND>       # pass incoming traffic to the command
```

IPTraf – Monitorización de Red

```
yum -y install iptraf-ng
iptraf-ng
```

Interfaces de Red

```
cat /etc/udev/rules.d/80-net-name-slot.rules     # udev rules with persistent naming
vim /etc/udev/rules.d/70-persistent-net.rules    # these custom rules overwrite defaults
cat /etc/sysconfig/network-scripts/ifcfg-eth1    # filename should match device name
  DEVICE=<NAME>
  HWADDR=<MAC_ADDRESS>
  BOOTPROTO=STATIC            # static/none (needs IPADDR0,PREFIX0) or dhcp/bootp
  ONBOOT=yes
  TYPE=Ethernet
  USERCTL=yes
  PEERDNS=no                  # define entries in /etc/resolv.conf (needs DNS1,DNS2)
  IPV6INIT=no                 # use IPv6 (needs IPV6ADDR/MASK,IPV6_AUTOCONF)
  IPADDR=<IPv4>
  NETMASK=<MASK>
```

Inspeccionando Tráfico de Red

```
yum -y install wireshark-gnome
wireshark &
wireshark -r <FILE> &            # analyze captured packets previously saved in file
```

Captura de Tráfico

```
tcpdump -c <NUMBER> -w <FILE.pcap>  # capture number of packets to the file
tcpdump -r <FILE.pcap>           # read from a capture file
tcpdump 'host <HOSTNAME>'        # coming to/from host
tcpdump 'src <HOSTNAME>'         # from host
tcpdump 'port <NUMBER>'

# icmp to/from host tcpdump 'ip host <HOSTNAME1> and not <HOSTNAME2>'
tcpdump 'icmp and host <IPv4>'
tcpdump -x                       # display packet header and hexadecimal values
tcpdump -X                       # display data as hexadecimal and ASCII values
tcpdump -X -r <FILE.pcap> 'host <HOSTNAME>' | grep -i 'pass' # display plaintext
passwords
```

2.4. ANÁLISIS DEL SOFTWARE.

2.4.1. Análisis de Paquetería.

Tabla 2.4-1: Análisis de la Paquetería.

SOFTWARE:

Paquetería

Dependencias entre Paquetes

```
# Display package dependencies:
yum deplist <PACKAGE>                # same as rpm -q -R <PACKAGE>
# Resolving package dependencies:
yum downgrade <PACKAGE>               # same as rpm -U --oldpackage <PACKAGE>
rpm -U --force <PACKAGE>             # same as --oldpackage --replacepkgs --replacefiles
# Using YUM to lock package versions:
# 'yum list available yum-plugin*'
yum -y install yum-plugin-versionlock
yum versionlock list               # display list of locked package versions
yum versionlock add <PATTERN>      # lock current versions of packages matched by wildcard
yum versionlock delete <PATTERN>   # delete locks matched by wildcard
yum versionlock clear              # clear all package version locks
yum list --showduplicates <PACKAGE>\*     # find all available versions of a package
```

Reparando base de Datos RPMs corrupta

```
lsof | grep /var/lib/rpm
rm /var/lib/rpm/__db*                          # remove database indexes
tar cjvf rpmdb-$(date +%Y%m%d-%H%M).tar.bz2 /var/lib/rpm
cd /var/lib/rpm
/usr/lib/rpm/rpmdb_verify Packages             # verify RPM database integrity
mv Packages Packages.bad
/usr/lib/rpm/rpmdb_dump Packages.bad | /usr/lib/rpm/rpmdb_load Packages
/usr/lib/rpm/rpmdb_verify Packages
rpm -v --rebuilddb # rebuild database indexes, 'rpm -qa > /dev/null' shouldn't show
anything
```

	Verificación Cambios RPM

	`# Verifying changed files with RPM:`
	`rpm -qf <PATH>  # what package does the file belong to`
	`rpm -ql <PACKAGE>  # list files in package`
	`tail /var/log/yum.log  # see what was installed recently`
	`rpm -V <PACKAGE(S)>  # verify package (S,M,5,L,U,G,T), shows file types (c,d,l,r) for some`
	`rpm -Va  # verify files of all installed packages`
	`# Verifying changes with YUM:`
	`yum -y install yum-plugin-verify  # works like 'rpm -V'`
	`yum verify <PACKAGE>  # does not show configuration files changes`
	`yum verify-rpm <PACKAGE>  # includes configuration files diff from original`
	`# Recovering changed files:`
	`rpm --setperms <PACKAGE>  # resets the permissions of files in a package`
	`rpm --setguids <PACKAGE>  # resets the user/group ownership of files`
	`yum reinstall <PACKAGE>  # repair installed package`

2.4.2. Análisis de Recursos de las Aplicaciones.

Tabla 2.4-2: Análisis de las Recursos de las Aplicaciones.

SOFTWARE: **Recursos**	
	Linkado contra librerías compartidas (.so=shared libraries)

	`objdump -p /usr/lib64/<LIBRARY>-<VER>.so | grep SONAME`
	`ls -l /usr/lib64/<LIBRARY>*  # shared library has symbolic link to DT_SONAME field`
	`ls -l /lib64/ld-linux-x86-64.so*  # default 64-bit runtime linker on RHEL7`
	`ls -l /lib/ld-linux.so*  # default 32-bit runtime linker on RHEL7`
	`ldconfig [-v]  # updates the runtime linker cache`
	`ldconfig -p  # list of libraries in /etc/ld.so.cache`
	`ldd <PATH/TO/EXECUTABLE>  # required shared libraries by executable (| grep 'not found')`
	`yum whatprovides '*lib/<LIBRARY>.so.0'  # identify package that provides shared library`
	`rpm -q --requires -p <FILE>.rpm  # required runtime libraries are stored in RPM metadata`

	Diagnosticando Fugas de Memoria

	`yum -y install valgrind`
	`valgrind --tool=memcheck --leak-check=full <PROGRAM>`
	`watch -d -n1 'free -mh; grep -i commit /proc/meminfo'`

	Mostrando Llamadas al Sistema

	`# launch executable with strace, -o=to file,-e=show only specific events`
	`strace [-o <FILE> -e <SYSCALLS>] <EXECUTABLE>`
	`strace -f <EXECUTABLE>  # also follow the execution of forks (child processes)`
	`strace -p <PID>  # trace a process already executed`

	Mostrando Llamadas a Librerías
	`ltrace -S <EXECUTABLE>  # strace + ltrace (needs at least read access to executable)` `ltrace -p <PID>  # trace a process already executed`

2.4.3. Análisis de Seguridad.

SOFTWARE:

Seguridad

SElinux: Trazado

```
ausearch -m avc -ts recent        # display Access Vector Control messages, last 10mins
yum -y install setools-console    # provides "seinfo", "sesearch"
sesearch -D                       # full list of all active "dontaudit" rules
sesearch --allow -b <BOOLEAN>       # view allow rules enabled by the boolean
# disable "dontaudit" rules until turned 'dontaudit on'-blocks events, but does not log them
semanage dontaudit off
seinfo -t httpd_sys_content_t       # or -b to show booleans
seinfo --portcon=443 --protocol=tcp

# An example of USER_AUTH entry in the audit log:
type=USER_AUTH  msg=audit(1564120484.274:12873):  pid=22300  uid=0  auid=1000  ses=2
subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
msg='op=PAM:authentication    grantors=pam_rootok    acct="root"    exe="/usr/bin/su"
hostname=localhost.localdomain addr=? terminal=pts/0 res=success'
```

SELinux: Troubleshooting

```
yum -y install setroubleshoot-server # provides 'sealert', 'sedispatch' plugin for auditd
service auditd restart   # auditd's sedispatch plugin requires restart, don't use systemd
sealert -l <UUID_OF_DENIAL>
sealert -a /var/log/audit/audit.log  # parse all denial messages out of file
```

SELinux: Problemas Recurrentes

```
semanage fcontext -a -t <TYPE> '<PATH>(/.*)?' # add path to list of standard file contexts
restorecon -Rv <PATH>                         # apply the new file contexts
# perform automatic relabel of all files after disabled->enforcing, which causes all files to be 'unlabeled_t'
touch /.autorelabel
semanage boolean --list      # current/default state + description of all SELinux toggles
# boolean values updated permanently, without -P only in memory
setsebool -P <BOOLEAN=ON/OFF>

# label an unlabeled port (e.g. http_port_t on port 8001 etc.)
semanage port -a -t <TYPE> -p tcp <PORT>

# vsftpd public directory, where anonymous are allowed to write:
semanage fcontext -a -t public_content_rw_t '/var/ftp/pub(/.*)?'
setsebool -P allow_ftpd_anon_write=1
```

PAM (Pluggable Authentication Modules): Configuración

```
vim /etc/pam.d/<SERVICE>      # if none is found, /etc/pam.d/other will block all access
  type control module-path [module-arguments]   # shouldn't be configured by hand,
                                                #     but by "authconfig" tools
```

PAM (Pluggable Authentication Modules): Troubleshooting

```
tail -f /var/log/secure
journalctl -u <PROBLEMATIC_SERVICE> # good start logging issues: 'journalctl _COMM=login'

# did files belonging to service change, especially PAM config?
rpm -V <PROBLEMATIC_SERVICE>

# rename broken PAM config, otherwise reinstall will not touch it
mv /etc/pam.d/<PROBLEMATIC_FILE>{,.broken}
yum reinstall <PROBLEMATIC_SERVICE>
```

```
# compare good and bad PAM config of problematic service
diff -u /etc/pam.d/<PROBLEMATIC_FILE>{,.broken}
authconfig
authconfig-tui
authconfig-gtk

# recreate all configuration files + re-apply configuration
#              as stored in /etc/sysconfig/authconfig
authconfig --updateall
yum -y install pam_krb5     # when the module for Kerberos is missing
```

LDAP: Troubleshooting

```
yum -y install openldap-clients # set of tools

# LDAP defaults (BASE,URI etc.), usually port TCP/389 with STARTTLS
cat /etc/openldap/ldap.conf

# when CAs mismatches are happening, this needs to be done
mv <CRT> /etc/openldap/cacerts;cacertdir_rehash

# -x simple auth, -ZZ enforce TLS, -LL disable comments in output,
#         only return canonical name and home directory
ldapsearch -x -ZZ -LL '(uid=ldapsuer)' \
  cn homeDirectory
getent passwd <LDAP_USER>     # uses nsswitch.conf to query backend password systems
```

Kerberos: Troubleshoot

```
# Solving Kerberos issues:
kinit <USER>      # obtain TGT (ticket granting ticket), time must match everywhere!

# is the [domain_realm] section correct in Kerberos5 config?
cat /etc/krb5.conf | grep -A 1 domain_realm
yum -y install sssd-common
man sssd-krb5   # when System Security Services Daemon is used (krb5_server,krb5_realm)
/etc/sssd/sssd.conf # this cache may contain KRB5 as well. Change needs restart of sssd
yum -y install krb5-workstation
klist                      # check if the user received TGT

# inspect keytabs, KVNO shows version of the password stored.
# When you overwrite Keytab file with a new one, dependant services (e.g. NFS)
#     must be restarted
klist -ek /etc/krb5.keytab
# sec=krb5i vs. sec=krb5p = they must match everywhere, needs autofs restart
cat /etc/exports.d/* /etc/auto.guests /etc/fstab

# when testing LDAP instead of SSH key auth
ssh -o PreferredAuthentications=keyboard-interactive,password ldapuser@server
```

2.5. ANÁLISIS DEL KERNEL.

2.5.1. Análisis del Kernel.

SOFTWARE:
Kernel

Kdump y kexec

```
# provides graphical configuration tool for kdump
yum -y install kexec-tools system-config-kdump
cat /etc/default/grub | grep GRUB_CMDLINE_LINUX
  ... crashkernel=auto ...            # after adding this, run "grub2-mkconfig -o ..."

# by default crash dumps go to /var/crash/<IP>-<DATE> (raw,nfs,ssh is possible)
cat /etc/kdump.conf | grep ^path

# by default collection is done by "makedumpfile" utility
cat /etc/kdump.conf | grep ^core_collector
vim /etc/kdump.conf
  core_collector scp         # collection of crash dumps using SSH dump targets (needs
ssh,ssh_key)

# lzo compression, only progress indicator, exclude some pages (-d=dump level)
makedumpfile -l --message-level 1 -d 31
man 8 makedumpfile      # -c=zlib, -l=lzo, -p=snappy, message level
    # 1=Only include progress indicator, 4=Only include error messages, 31=Include all
messages

# enables and starts, must be restarted when config file is changed
systemctl enable --now kdump
kdumpctl status
kdumpctl showmem           # how much memory is reserved for crash kernel
kdumpctl propagate         # simplify setup of SSH key (sshkey in kdump.conf) authentication
```

Kernel Crash Dumps

```
echo "vm.panic_on_oom=1" >> /etc/sysctl.conf        # panic on OOM-killer events permanently
echo "kernel.hung_task_panic=1" >> /etc/sysctl.conf # panic on hung process perm
cat /proc/sys/kernel/hung_task_timeout_secs         # hung task timeout

# soft lockups (kernel loops in kernel mode) perm
echo "kernel.softlockup_panic=1" >> /etc/sysctl.conf
echo "kernel.panic_on_io_nmi=1" >> /etc/sysctl.conf # nonrecoverable HW failure (NMI) perm

# enable all magic sysrq (key sequence in case of unresponsive system) perm
echo "kernel.sysrq=1" >> /etc/sysctl.conf

# initiate a system crash (other sysrq keys: m,t,p,c,s,u,b,9,f,w)
echo "c" > /proc/sysrq-trigger
sysctl -p                          # load in Kernel parameters
```

Analizar Crash Dumps

```
yum -y install kernel-debuginfo
strings vmcore | head                            # same info as vmcore-dnesg.txt

# it needs debug version of the kernel image and crash dump
crash /usr/lib/debug/modules/<KERNEL_VER>/vmlinux \
  /var/crash/<IP_ADDRESS-DATE-TIME>
```

Depuración del Kernel con SystemTap

```
# Install software needed to compile SystemTap modules:
subscription-manager repos --enable rhel-7-server-debug-rpms
yum -y install kernel-debuginfo kernel-devel systemtap

# checks current kernel and install matching devel & debuginfo
stap-prep

# useful *.stp scripts from systemtap package
ls /usr/share/doc/systemtap-client-*/examples
stap -v /usr/share/doc/systemtap-client-*/examples/process/syscalls_by_proc.stp

# Compile a kernel stap module to a specific dir:
#      generates *.ko in the current dir, '-p 4'=only first 4 steps, -m=filename
stap -p 4 -v -m syscalls_by_proc \
  /usr/share/doc/systemtap-client-*/examples/process/syscalls_by_proc.stp

# Make module available for users in "stapusr" group:
#     Folder must be owned by root and not be world writable
mkdir /lib/modules/$(uname -r)/systemtap
ls -ld /lib/modules/$(uname -r)/systemtap
cp /root/syscalls_by_proc.ko /lib/modules/$(uname -r)/systemtap

# run the module, doesn't need to specify extension here
staprun syscalls_by_proc

# For only the SystemTap runtime environment you need a single package:
yum -y install systemtap-runtime

# see the PERMISSIONS section of the stap manpage for all the details
man -P 'less +/PERMISSIONS' stap

# User in "stapdev" & "stapusr" groups can run the module from anywhere
#    (do this on the destination machine):
#      can run SystemTap modules, but only if they exist
#        in the /lib/modules/$(uname -r)/systemtap dir
usermod -aG stapusr <USER>

# may compile their own SystemTap instrumentation kernel modules using stap,
# if they are also in 'stapusr', they may use staprun to load a module,
# even if it does not reside in the /lib/modules/$(uname -r)/systemtap directory
usermod -aG stapdev <USER>
```