Table of Contents

InstallingCentOS7LocalSite

RED: Not done yet or not working

CERN 7

Installation

1. Prepare boot media (you will need a single recordable CD or USB memory key), and copy the OS image available at: http://linuxsoft.cern.ch/cern/centos/7/os/x86_64/images/boot.iso

2. Select in "Installation source" the "http" method with the following address: http://linuxsoft.cern.ch/cern/centos/7/os/x86_64/

3. Time zone: America/Santiago. Enable Network time protocol and add the NTP servers: ntp1.puc.cl, ntp.shoa.cl, removing the rest.

4. Follow the rest of the instructions from: http://linux.web.cern.ch/linux/centos7/docs/install.shtml

5. Disable: kDump SELinux and Firewall

Fix BIOS/RHEL problem after installation

After a fresh install of CentOS 7 (or any RHEL OS), HP DL 380/360 G5/G6 servers shows the following error:

```
BIOS HAS CORRUPT [hw] PMU RESOURCES....
```

Then the server does not boot, hangs after grub, it gets a blank screen without anymore details.

One way to solve this problem is to modify the grub. Another is to disable the "Processor Power and Utilization Monitoring" in the BIOS following the next steps:

1. Enter the BIOS

2. By pressing CTRL+A, the power utilisation option suddently appears at the end of the menu, select it.

3. Select the option "Processor Power and Utilization Monitoring" and disable it.

4. When the system start and the screen goes dark after trying to load the OS for the first time, press CTRL+ALT+F2/3/ and follow the instructions on the screen.

More information about this issue: redhat: https://bugzilla.redhat.com/show_bug.cgi?id=787126 hp: https://community.hpe.com/t5/ProLiant-Servers-ML-DL-SL/BIOS-HAS-CORRUPT-hw-PMU-RESOURCES/td-p/600

Configuration

1. Create a root user and install&update the packages according to: http://linux.web.cern.ch/linux/centos7/docs/install.shtml

2. Assign a static IP address to the site with the corresponding DNS servers, by using the network manager or editing /etc/sysconfig/network-scripts/ifcfg-eno1 (in the case of Nyx):

- IP: 146.155.46.166
- DNS: 146.155.47.15, 146.155.1.155
- Netmask: 255.255.255.0

3. Restart network:

```
/etc/init.d/network restart
```

Server Configuration

1. Install Openssh Server:

```
yum install openssh-server
```

2. After the Installation is finished, start the sshd service using systemctl command:

```
systemctl start sshd.service
systemctl enable sshd.service
```

After any network change, you should restart the sshd service by using the following command:

```
systemctl start|stop|restart|status network
```

Add users to the server

1. To create a new user account, type:

```
adduser username
```

2. Set a password for the new user. You'll be prompted to input and confirm a password.

```
passwd username
```

3. If your new user should have the ability to execute commands with root (administrative) privileges, you will need to give the new user access to sudo. We can do this by adding the user to the wheel group (which gives sudo access to all of its members by default) through the usermod command.

```
usermod -aG wheel usernamel
```

4. If you want to delete a user home directory along with the user account itself, type this command as root:

```
userdel -r username
```

5. Disallow Root login and password authentication

Customizing welcome message

You can have the MOTD (message of the day) display messages that may be unique to the machine. One way to do this is to create a text file that runs when a user logs on to the system.

```
emacs -nw /etc/motd
```

and add the message you want (In Nyx we have the commands to setup ROOT and DevTools2 7).

Upgrade CMake

The default CMake version available in CentOS7 (and SLC 6) is 2.8.11, which is too old to install the last version (6.X) of Root Framework. Given that it is impossible to update the version, we would have to delete cmake completely and install a newer version (3.6.2).

1. In order to install version 3.6.2 or newer, first you must uninstall it with yum remove:

```
sudo yum remove cmake -y
```

If you don't perform the above step to remove the old CMake version, you may see an error after the final step.

2. Download, extract, compile and install the CMake from https://cmake.org/download/:

```
wget https://cmake.org/files/v3.6/cmake-3.6.2.tar.gz
tar -zxvf cmake-3.6.2.ta
cd cmake-3.6.2
sudo ./bootstrap --prefix=/usr/local
sudo make
sudo make install
```

3. Edit the configuration file: ~/.bash_profile, including:

```
PATH=/usr/local/bin:$PATH:$HOME/bin
```

4. You can check the new version by typing:

```
cmake --version
```

Upgrade GCC by installing development tools

Developer Toolset 7.0 provides current versions of the GNU Compiler Collection (7.2), GNU Debugger, and other development, debugging, and performance monitoring tools. These do not replace the Red Hat Enterprise Linux system versions of these tools, nor will they be used in preference to those system versions unless explicitly invoked using the scl utility.

In case your scripts were made using a GCC version higher than 4.8.5 we recommend you to install and use these tools.

1. Install a package with the repository for your system:

```
sudo yum install centos-release-scl
```

2. Enable RHSCL repository for you system

```
sudo yum-config-manager --enable rhel-server-rhscl-7-rpms
```

3. Install the collection:

```
sudo yum install devtoolset-7
```

4. In order to use the GNU C++ compiler, you need to add DTS to your environment with scl enable in a Terminal window (you need yo do this each time you want to use it, you may also want to add it to your .bashrc in case you want to activate it by default).

```
scl enable devtoolset-7 bash
```

5. You can check the new version by typing:

```
gcc --version
```

More details can be found at: https://developers.redhat.com/products/developertoolset/hello-world/

ROOT Framework Installation

Nowadays ROOT uses the CMake (link is external) cross-platform build-generator tool as a primary build system. CMake does not build the project, it generates the files needed by your build tool (GNU make, Ninja, Visual Studio, etc) for building ROOT. In order to work with a compile version of ROOT you must download a source version.

1. First of all, ROOT needs prerequisite packages to be installed to be able to configure and to build basic ROOT. If more advanced ROOT plugins are required look at the cmake output and add the desired third party packages before configuring again. The required packages can be installed by using the following command:

```
sudo yum install git cmake gcc-c++ gcc binutils \
libX11-devel libXpm-devel libXft-devel libXext-devel
```

It is recommended to install the optional packages too:

```
sudo yum install gcc-gfortran openssl-devel pcre-devel \
mesa-libGL-devel mesa-libGLU-devel glew-devel ftgl-devel mysql-devel \
fftw-devel cfitsio-devel graphviz-devel \
avahi-compat-libdns_sd-devel libldap-dev python-devel \
libxml2-devel gsl-static
```

2. Download and unpack the ROOT's sources from the download area:

```
wget https://root.cern.ch/download/root_<version>.source.tar.gz
tar -zxf root_<version>.source.tar.gz
```

3. Create a directory for containing the build. It is not supported to build ROOT on the source directory. cd to this directory:

```
mkdir <builddir>
cd <builddir>
```

4. Execute the cmake command on the shell replacing path/to/source with the path to the top of your ROOT source tree:

```
cmake path/to/source
```

CMake will detect your development environment, perform a series of test and generate the files required for building ROOT. CMake will use default values for all build parameters. See the Build Options and Variables sections for fine-tuning your build This can fail if CMake can t detect your toolset, or if it thinks that the environment is not sane enough. On this case make sure that the toolset that you intend to use is the only one reachable from the shell and that the shell itself is the correct one for you development environment. You can force CMake to use a given build tool, see the Usage section.

5. After CMake has finished running, proceed to use IDE project files or start the build from the build directory. On unix systems (with make or ninja) you can speedup the build with:

```
cmake --build . -- -jN
```

where N is the number of available cores.

6. ROOT build options are boolean variables that can be turned ON or OFF. The current value is recorded in the CMake cache (CMakeCache.txt file on the build directory) and therefore it is not needed to be specified on the cmake command each time. Please note that some of the options might be turned OFF automatically for some platforms or if the required external library or component can not be satisfied. The user can view and

edit the full list of options using the ccmake utility in the source directory. Python3 and RooFit are two recommended options to activate before building the software.

```
cd path/to/source
ccmake
```

You need to execute cmake and compile again to install the new libraries:

```
cmake path/to/source
cmake --build . -- -jN
```

7. Setup the environment to run:

```
source /path/to/install-or-build/dir/bin/thisroot.sh
```

in the case of Nyx:

```
source /opt/RootFramework_6.10.08/root-6.10.08/bin/thisroot.sh
```

8. Start ROOT interactive application

```
root
```

Slurm Installation

To submit a job using more than one cpu by using an executable script to a batch server

You can install MariaDB to store the accounting that Slurm provides. If you want to store accounting, here s the time to do so. I only install this on the server node, buhpc3. I use the server node as our SlurmDB node.

```
  yum install mariadb-server mariadb-devel -y
```

Slurm and Munge require consistent UID and GID across every node in the cluster. For all the nodes, before you install Slurm or Munge:

```
  export MUNGEUSER=991
  sudo groupadd -g $MUNGEUSER munge
  sudo useradd  -m -c "MUNGE Uid 'N' Gid Emporium" -d /var/lib/munge -u $MUNGEUSER -g munge  -s
  export SLURMUSER=970
  sudo groupadd -g $SLURMUSER slurm
  sudo useradd  -m -c "SLURM workload manager" -d /var/lib/slurm -u $SLURMUSER -g slurm  -s /bin
```

Since I m using CentOS 7, I need to get the latest EPEL repository.

```
  sudo yum install epel-release
  sudo yum install munge munge-libs munge-devel -y
```

After installing Munge, I need to create a secret key on the Server. My server is on the node with hostname, buhpc3. Choose one of your nodes to be the server node.

First, we install rng-tools to properly create the key.

```
  yum install rng-tools -y
  rngd -r /dev/urandom
```

Now, we create the secret key. You only have to do the creation of the secret key on the server.

```
  /usr/sbin/create-munge-key -r
```

```
dd if=/dev/urandom bs=1 count=1024 > /etc/munge/munge.key
chown munge: /etc/munge/munge.key
chmod 400 /etc/munge/munge.key
```

After the secret key is created, you will need to send this key to all of the compute nodes.

```
scp /etc/munge/munge.key root@10.0.0.202:/etc/munge
scp /etc/munge/munge.key root@10.0.0.203:/etc/munge
scp /etc/munge/munge.key root@10.0.0.204:/etc/munge
scp /etc/munge/munge.key root@10.0.0.205:/etc/munge
```

Now, we SSH into every node and correct the permissions as well as start the Munge service.

```
sudo chown -R munge: /etc/munge/ /var/log/munge/
sudo chmod 0700 /etc/munge/ /var/log/munge/

systemctl enable munge
systemctl start munge
systemctl restart munge
```

To test Munge, we can try to access another node with Munge from our server node, buhpc3.

```
munge -n
munge -n | unmunge
munge -n | ssh 10.0.0.202 unmunge
remunge
```

Install Slurm

Slurm has a few dependencies that we need to install before proceeding.

```
sudo yum install openssl openssl-devel pam-devel numactl numactl-devel hwloc hwloc-devel lua l
```

Now, we download the latest version of Slurm preferably in our shared folder. The latest version of Slurm may be different from our version.

```
cd /opt/
wget https://download.schedmd.com/slurm/slurm-18.08.0.tar.bz2

sudo rpmbuild -ta slurm-18.08.0.tar.bz2
```

We will check the rpms created by rpmbuild.

```
cd /root/rpmbuild/RPMS/x86_64

scp * mhaacke@10.0.0.202:~/
scp * mhaacke@10.0.0.203:~/
scp * mhaacke@10.0.0.204:~/
scp * mhaacke@10.0.0.205:~/
```

on every machine

```
yum --nogpgcheck localinstall slurm-18.08.0-1.el7.x86_64.rpm slurm-libpmi-18.08.0-1.el7.x86_64
```

Now in /etc/slurm/

```
cp cgroup.conf.example cgroup.conf
```

We configure the slurm config file in slurm.conf.

```
# slurm.conf file generated by configurator easy.html.
# Put this file on all nodes of your cluster.
# See the slurm.conf man page for more information.
#

ControlMachine=nyx
ControlAddr=10.0.0.200

#MailProg=/bin/mail
MpiDefault=none
#MpiParams=ports=#-#
ProctrackType=proctrack/cgroup
ReturnToService=1
SlurmctldPidFile=/var/run/slurmctld.pid
#SlurmctldPort=6817
SlurmdPidFile=/var/run/slurmd.pid
#SlurmdPort=6818
SlurmdSpoolDir=/var/spool/slurmd
SlurmUser=slurm
#SlurmdUser=root
StateSaveLocation=/var/spool/slurmctld
SwitchType=switch/none
TaskPlugin=task/affinity
#
#
# TIMERS
#KillWait=30
#MinJobAge=300
#SlurmctldTimeout=120
#SlurmdTimeout=300
#
#
# SCHEDULING
FastSchedule=0
SchedulerType=sched/backfill
SelectType=select/cons_res
SelectTypeParameters=CR_Core
#
#
# LOGGING AND ACCOUNTING
AccountingStorageType=accounting_storage/none
ClusterName=nyxcluster
#JobAcctGatherFrequency=30
JobAcctGatherType=jobacct_gather/none
#SlurmctldDebug=3
#SlurmctldLogFile=
#SlurmdDebug=3
#SlurmdLogFile=
#
#
# COMPUTE NODES
NodeName=nyxwn20 CPUs=16 Boards=1 SocketsPerBoard=2 CoresPerSocket=8 ThreadsPerCore=1 RealMemo
NodeName=nyxwn19 CPUs=16 Boards=1 SocketsPerBoard=2 CoresPerSocket=8 ThreadsPerCore=1 RealMemo
NodeName=nyxwn18 CPUs=16 Boards=1 SocketsPerBoard=2 CoresPerSocket=8 ThreadsPerCore=1 RealMemo
NodeName=nyxwn17 CPUs=16 Boards=1 SocketsPerBoard=2 CoresPerSocket=8 ThreadsPerCore=1 RealMemo
NodeName=nyx CPUs=32 Boards=1 SocketsPerBoard=2 CoresPerSocket=8 ThreadsPerCore=2 RealMemory=1
PartitionName=Principal Nodes=nyxwn[17-20] Default=YES MaxTime=INFINITE State=UP
```

After this we copy this to every working node

```
scp slurm.conf root@10.0.0.202/etc/slurm/slurm.conf
```

in the server node

```
mkdir /var/spool/slurmctld
chown slurm: /var/spool/slurmctld
chmod 755 /var/spool/slurmctld
touch /var/log/slurmctld.log
chown slurm: /var/log/slurmctld.log
touch /var/log/slurm_jobacct.log /var/log/slurm_jobcomp.log
chown slurm: /var/log/slurm_jobacct.log /var/log/slurm_jobcomp.log
```

and in every working node

```
mkdir /var/spool/slurmd
chown slurm: /var/spool/slurmd
chmod 755 /var/spool/slurmd
touch /var/log/slurmd.log
chown slurm: /var/log/slurmd.log
```

Use the following command to make sure that slurmd is configured properly.

```
slurmd -C
```

The firewall will block connections between nodes, so I normally disable the firewall on the compute nodes except for nyx head.

```
systemctl stop firewalld
systemctl disable firewalld
```

On the server node, nyx, I usually open the default ports that Slurm uses:

```
firewall-cmd --permanent --zone=public --add-port=6817/udp
firewall-cmd --permanent --zone=public --add-port=6817/tcp
firewall-cmd --permanent --zone=public --add-port=6818/tcp
firewall-cmd --permanent --zone=public --add-port=6818/tcp
firewall-cmd --permanent --zone=public --add-port=7321/tcp
firewall-cmd --permanent --zone=public --add-port=7321/tcp
firewall-cmd --reload
```

If the port freeing does not work, stop the firewalld for testing. Next, we need to check for out of sync clocks on the cluster. On every node:

```
yum install ntp -y
chkconfig ntpd on
ntpdate pool.ntp.org
systemctl start ntpd
```

The clocks should be synced, so we can try starting Slurm! On all the compute nodes,

```
systemctl enable slurmd.service
systemctl start slurmd.service
systemctl status slurmd.service
```

Now, on the server node

```
systemctl enable slurmd.service
systemctl start slurmd.service
systemctl status slurmd.service
systemctl enable slurmctld.service
systemctl start slurmctld.service
systemctl status slurmctld.service
```

When you check the status of slurmd and slurmctld, we should see if they successfully completed or not. If problems happen, check the logs!

```
   Compute node bugs: tail /var/log/slurmd.log
   Server node bugs: tail /var/log/slurmctld.log
```

For more information, I followed this wiki
https://www.slothparadise.com/how-to-install-slurm-on-centos-7-cluster/

Hard drives mounting

After physically installing the new hard drives to the cluster, the OS need to recognize them.

1. You can check the installed hard drives, with their mounting points, by:

```
lsblk
```

2. Select the new disk drive by using fdisk (for example hdb):

```
fdisk /dev/hdb
```

3. Delete any unwanted partitions that may already be present on the new disk drive. This is done using the d command in fdisk (use default - ENTER):

```
Command (m for help): d
Partition number (1-4): 1
```

4. Create the new partition(s), being sure to specify the desired size and file system type. Using fdisk, this is a two-step process -- first, creating the partition (using the n command):

```
Command (m for help): n
Command action e extended p primary partition (1-4) p
Partition number (1-4): 1 First cylinder (1-767): 1
Last cylinder or +size or +sizeM or +sizeK: +512M
```

To use the entire disk just press ENTER to all the options (default)

5. Second, by setting the file system type (using the t command):

```
Command (m for help): t
Partition number (1-4): 1
Hex code (type L to list codes): 83
```

Partition type 83 represents a Linux partition.

6. Save your changes and exit the partitioning program. This is done in fdisk by using the w command:

```
Command (m for help): w
```

7. The partition still does not have a file system. Create the file system:

```
mkfs -t ext3 /dev/sdb1
```

ext3 format is chosen, but you can choose any other.

8. You can label the new partition (for example as data1):

```
e2label /dev/sdb1 /data1
```

9. As root, create the mounting point:

```
cd ..
mkdir /data1
```

10. As root, edit the /etc/fstab file to include the new partition. The new line should look similar to the following:

```
LABEL=/data1            /data1                  ext3    defaults        1 2
```

11. Reboot. To mount the partition without rebooting (recommended for testing), as root, type the command:

```
mount /data1
```

Set Hostname

There are two methods that are going to be explain here to setup the server's hostname (I used the first one, which may be wrong):

1. You can check the server hostname by using the following command:

```
hostnamectl status
```

2. Where the first line of the output tells the hostname:

```
Static hostname: nyx.fis.puc.cl
Icon name: computer
Machine ID: f9216fe8b6d04ff3ac2797a51ea8d365
Boot ID: 6d4b934c157748809d969e021d436fff
Operating System: CentOS Linux 7 (Core)
CPE OS Name: cpe:/o:centos:centos:7
Kernel: Linux 3.10.0-693.5.2.el7.x86_64
Architecture: x86-64
```

3. [METHOD 1] You can set host name using nmtui command which has text user interface for new users:

- Use the Down arrow key > select the Set system hostname menu option > Press the Ok button:

- You will see the confirmation box

3. [METHOD 2] Using a text editor, open the server s /etc/sysconfig/network file. Modify the HOSTNAME= value to match your FQDN hostname (if the file is empty, copy the following lines):

```
NETWORKING=yes
HOSTNAME=nyx.fis.puc.cl
NTPSERVERARGS=iburst
```

4. Finally, restart hostnamed service by typing the following command

```
systemctl restart systemd-hostnamed
```

5. If you used the second method, you need to restart the network to ensure that changes will persist on reboot:

```
/etc/init.d/network restart
```

6. To verify changes, re-enter:

```
hostnamectl status
```

or

```
hostname
```

Mounting EOS via FUSE

The FUSE client allows to access EOS as a mounted file system, which is a service already installed at the server. To check all the available puppet modules with their current state

```
/usr/bin/locmap --list
```

Current available modules are:

- kerberos
- sendmail
- cernbox
- ntp
- gpg
- cvmfs
- ssh
- lpadmin
- nscd
- afs
- sudo
- eosclient

You have to notice that modules may be enabled but not configured (check afs).

To enable one of the modules:

```
/usr/bin/locmap --enable module_name
```

Then to setup:

```
/usr/bin/locmap --configure module_name
```

1. In case EOS is not enable and configure, you need to do it by (NOTE: I have an error with configure because it does not recognize the name of the server):

```
/usr/bin/locmap --enable eosclient
/usr/bin/locmap --configure eosclient
```

2. To install EOS-fuse, you need to modify CERN YUM repository, adding a file /etc/yum.repos.d/eos7-stable.repo, with the following content:

```
[eos-tag]
name=tagged EOS releases from EOS project
baseurl=http://storage-ci.web.cern.ch/storage-ci/eos/citrine/tag/el-$releasever/$basearch/
enabled=1
gpgcheck=0
priority=10
```

3. As root, run:

```
yum install eos-fuse
service eosd start
```

Kerberos

Kerberos is shared-secret networked authentication system. Its use at CERN serves a dual purpose:

- user convenience: once signed in, authentication is handled "automagically" on the users' behalf.
- security: by reducing the need to enter passwords into potentially compromised machines, the impact of such a compromise can be greatly reduced.

In order for Kerberos to work, the user needs to be registered in a central database (KDC). In CERN parlance, this means a CERN account is required for the user.

1. The authentication needs a working config file (with CERN as default realm) , which default location is /etc/krb5.conf. You can edit the krb5.conf file using the following syntax:

```
includedir /etc/krb5.conf.d/
[logging]
 default = FILE:/var/log/krb5libs.log
 kdc = FILE:/var/log/krb5kdc.log
 admin_server = FILE:/var/log/kadmind.log

[libdefaults]
 dns_lookup_realm = false
 ticket_lifetime = 24h
 renew_lifetime = 7d
 forwardable = true
 rdns = false
 default_realm = CERN.CH
 default_ccache_name = KEYRING:persistent:%{uid}

[realms]
  CERN.CH = {
    default_domain = cern.ch
    kpasswd_server = cerndc.cern.ch
    admin_server = cerndc.cern.ch
    kdc = cerndc.cern.ch
  }

[domain_realm]
  cern.ch = CERN.CH
  .cern.ch = CERN.CH
```

2. Check whether your TGT is "forwardable" for CERN.CH

```
klist -f
```

3. To login to Kerberos and mount the afs share on your lxplus system:

```
kinit username@CERN.CH
aklog
```

File System (CVMFS)

The CernVM File System provides a scalable, reliable and low-maintenance software distribution service. It was developed to assist High Energy Physics (HEP) collaborations to deploy software on the worldwide-distributed computing infrastructure used to run data processing applications. CernVM-FS is implemented as a POSIX read-only file system in user space (a FUSE (link is external) module). Files and directories are hosted on standard web servers and mounted in the universal namespace /cvmfs.

CernVM-FS is actively used by small and large HEP collaborations. In many cases, it replaces package managers and shared software areas on cluster file systems as means to distribute the software used to process experiment data. For the experiments at the LHC, CernVM-FS hosts several hundred million files and directories that are distribute to the order of hundred thousand client computers.

1. Create a new partition where CVMFS will be mounted (please refer to Section "Hard drives mounting"). In our case we re-created the filesystem of sdb, creating two partitions: a 50GB CVMFS partition and a 880GB data partition. * The size of your partition is defined in the second step of the filesystem option: "Last cylinder or +size or +sizeM or +sizeK: +52G" * To define the size of your partition, you need to in the filesystem step: "Last cylinder or +size or +sizeM or +sizeK: +52G"

Non cc installation

Fetch and install the repo information (only needed first time):

```
yum install https://ecsft.cern.ch/dist/cvmfs/cvmfs-release/cvmfs-release-latest.noarch.rpm
```

Install or update cvmfs:

```
yum install cvmfs cvmfs-config-default cvmfs-auto-setup
```

Follow step 3 in cc installation

cc instalation

2. In case CVMFS is not enable and configure, you need to do it by:

```
 /usr/bin/locmap --enable cvmfs
 /usr/bin/locmap --configure cvmfs
```

1. CVMFS should be installed in the server, if not:

```
 sudo yum install cvmfs cvmfs-config-default
```

3. For the base setup, run

```
 cvmfs_config setup
```

4. Create /etc/cvmfs/default.local and open the file for editing

5. Select the desired repositories, for instance ATLAS uses the following:

```
 CVMFS_REPOSITORIES=atlas.cern.ch,atlas-condb.cern.ch,grid.cern.ch
```

6. Specify the HTTP proxy servers on your site with (NOTE: The ip represents the squid proxy of your site):

```
CVMFS_HTTP_PROXY="10.0.0.99:3128;DIRECT"
```

or use direct connection:

```
 CVMFS_HTTP_PROXY=DIRECT
```

7. Check if CernVM-FS mounts the specified repositories by (NOTE: It does not find the servers):

```
cvmfs_config probe
```

If the probe fails, try to restart autofs with:

```
sudo service autofs restart
```

Useful commands

- check cpu: lscpu

- check space: df -h

- check ram: cat /proc/meminfo

- lsblk: check partitions with mountpoint

-- SebastianOlivares - 2017-10-30

Major updates:
-- Main.Sebastian Olivares - 30 Oct 2017

RESPONSIBLE SebastianOlivares

This topic: Main > InstallingCentOS7LocalSite
Topic revision: r12 - 2020-06-18 - MaximKonyushikhin