

# Building Lustre on Centos 7.3

## 1.1. Purpose

Describe the steps you need to build and test a Lustre system (MGS, MDT, MDS, OSS, OST, client) with ZFS from the `master` branch on a x86\_64, RHEL /CentOS 7.3 machine.

## 1.2. Prerequisite

- A newly installed RHEL/CentOS 7.3 x86\_64 machine connected to the internet.
- [EPEL Repository](#): this is a convenient source for git.
- **NOTE** It is suggested that you have at least 1GB of memory on the machine you are using for the build.
- **NOTE** Verify that selinux is disabled.

## 1.3. Overview

This document walks through the steps of building ZFS, building Lustre and running a basic test of the complete system.

## 1.4. Procedure

The procedure requires that a OS is setup for development - this includes Lustre sources, and build tools.

### 1.4.1. Provision machine and installing dependencies.

Once RHEL 7.3 is newly installed on `rhel-master` login as user `root`.

1. Install the kernel development tools:

```
# yum -y groupinstall "Development Tools"
```

Problem with installing 'Development Tools'

If the Development Tools group is not be available for some reason, you may find the following list if individual packages necessary to install.

```
# yum -y install automake xmlto asciidoc elfutils-libelf-devel zlib-devel binutils-devel newt-devel python-devel hmaccalc perl-ExtUtils-Embed rpm-build make gcc redhat-rpm-config patchutils git
```

2. Install additional dependencies:

```
# yum -y install xmlto asciidoc elfutils-libelf-devel zlib-devel binutils-devel newt-devel python-devel hmaccalc perl-ExtUtils-Embed bison elfutils-devel audit-libs-devel kernel-devel libattr-devel libuuid-devel libblkid-devel libsasl-devel libselinux-devel libudev-devel
```

3. Install EPEL 7:

```
# rpm -ivh http://download.fedoraproject.org/pub/epel/7/x86\_64/e/epel-release-7-9.noarch.rpm
```

4. Install additional packages:

```
# yum -y install pesign numactl-devel pciutils-devel ncurses-devel libselinux-devel fio
```

### 1.4.2. Preparing the Lustre source.

1. Create a user `build` with the home directory `/home/build`:

```
# useradd -m build
```

2. Switch to the user `build` and change to the build `$HOME` directory:

```
# su build
$ cd $HOME
```

3. Get the MASTER branch from git:

```
$ git clone git://git.whamcloud.com/fs/lustre-release.git
$ cd lustre-release
```

4. Run `sh ./autogen.sh` to generate the configure script.
5. Resolve any outstanding dependencies until `autogen.sh` completes successfully. Success will look like:

```
$ sh ./autogen.sh
configure.ac:10: installing 'config/config.guess'
configure.ac:10: installing 'config/config.sub'
configure.ac:12: installing 'config/install-sh'
configure.ac:12: installing 'config/missing'
libcfs/libcfs/automakefile.am: installing 'config/depcomp'
$
```

### 1.4.3. Prepare ZFS for Lustre

1. Get SPL and ZFS sources from Github:

```
$ cd $HOME
$ git clone https://github.com/zfsonlinux/spl.git
$ git clone https://github.com/zfsonlinux/zfs.git
$
$ cd spl
$ ./autogen.sh
$ cd ../zfs
$ ./autogen.sh
$ cd $HOME
```

2. Build SPL for kabi-tracking kmod:

```
$ cd $HOME/spl
$ ./configure --with-spec=redhat
$ make pkg-utils pkg-kmod
```

This will end with:

```
...
...
Checking for unpackaged file(s): /usr/lib/rpm/check-files /tmp/spl-build-build-FxdXBfej/BUILDROOT/spl-
kmod-0.7.0-rc3.el7.centos.x86_64
Wrote: /tmp/spl-build-build-FxdXBfej/RPMS/x86_64/kmod-spl-0.7.0-rc3.el7.centos.x86_64.rpm
Wrote: /tmp/spl-build-build-FxdXBfej/RPMS/x86_64/kmod-spl-devel-0.7.0-rc3.el7.centos.x86_64.rpm
Wrote: /tmp/spl-build-build-FxdXBfej/RPMS/x86_64/spl-kmod-debuginfo-0.7.0-rc3.el7.centos.x86_64.rpm
Executing(%clean): /bin/sh -e /tmp/spl-build-build-FxdXBfej/TMP/rpm-tmp.NhMrJh
+ umask 022
+ cd /tmp/spl-build-build-FxdXBfej/BUILD
+ cd spl-0.7.0
+ rm -rf /tmp/spl-build-build-FxdXBfej/BUILDROOT/spl-kmod-0.7.0-rc3.el7.centos.x86_64
+ exit 0
Executing(--clean): /bin/sh -e /tmp/spl-build-build-FxdXBfej/TMP/rpm-tmp.FDCzZG
+ umask 022
+ cd /tmp/spl-build-build-FxdXBfej/BUILD
+ rm -rf spl-0.7.0
+ exit 0
```

At this point you should have SPL package in `~build/spl`

3. Install the SPL packages. They are required dependencies to build ZFS:

```
As root, go to ~build/spl and install the packages
# yum -y localinstall *.x86_64.rpm
```

4. Build ZFS:

```
$ su build
$ cd $HOME/zfs
$ ./configure --with-spec=redhat
$ make pkg-utils pkg-kmod
```

This will end with:

```

...
...
Requires(rpmlib): rpmlib(FileDigests) <= 4.6.0-1 rpmlib(PayloadFilesHavePrefix) <= 4.0-1 rpmlib
(CompressedFileNames) <= 3.0.4-1
Checking for unpackaged file(s): /usr/lib/rpm/check-files /tmp/zfs-build-build-OhM03R8P/BUILDROOT/zfs-
kmod-0.7.0-rc3_10_gcc923db.el7.centos.x86_64
Wrote: /tmp/zfs-build-build-OhM03R8P/RPMS/x86_64/kmod-zfs-0.7.0-rc3_10_gcc923db.el7.centos.x86_64.rpm
Wrote: /tmp/zfs-build-build-OhM03R8P/RPMS/x86_64/kmod-zfs-devel-0.7.0-rc3_10_gcc923db.el7.centos.x86_64.
rpm
Wrote: /tmp/zfs-build-build-OhM03R8P/RPMS/x86_64/zfs-kmod-debuginfo-0.7.0-rc3_10_gcc923db.el7.centos.
x86_64.rpm
Executing(%clean): /bin/sh -e /tmp/zfs-build-build-OhM03R8P/TMP/rpm-tmp.Z1fVso
+ umask 022
+ cd /tmp/zfs-build-build-OhM03R8P/BUILD
+ cd zfs-0.7.0
+ rm -rf /tmp/zfs-build-build-OhM03R8P/BUILDROOT/zfs-kmod-0.7.0-rc3_10_gcc923db.el7.centos.x86_64
+ exit 0
Executing(--clean): /bin/sh -e /tmp/zfs-build-build-OhM03R8P/TMP/rpm-tmp.aBx3yh
+ umask 022
+ cd /tmp/zfs-build-build-OhM03R8P/BUILD
+ rm -rf zfs-0.7.0
+ exit 0

```

5. Install the ZFS packages. They are required dependencies to build Lustre:

```

As root, go to ~build/zfs and install the packages
# yum -y localinstall *.x86_64.rpm

```

## 1.4.4. Configure and build Lustre

1. Configure Lustre source:

```

$ su build
$ cd ~/lustre-release/
$ ./configure
...
...
config.status: executing depfiles commands
config.status: executing libtool commands

CC:          gcc
LD:          /usr/bin/ld -m elf_x86_64
CPPFLAGS:   -include /localhome/build/lustre-release/undef.h -include /localhome/build/lustre-release
/config.h -I/localhome/build/lustre-release/libcfs/include -I/localhome/build/lustre-release/lnet/include
-I/localhome/build/lustre-release/lustre/include
CFLAGS:     -g -O2 -Wall -Werror
EXTRA_KCFLAGS: -include /localhome/build/lustre-release/undef.h -include /localhome/build/lustre-release
/config.h -g -I/localhome/build/lustre-release/libcfs/include -I/localhome/build/lustre-release/lnet
/include -I/localhome/build/lustre-release/lustre/include

Type 'make' to build Lustre.

```

2. Make RPMs:

```

$ make rpms
...
...
Checking for unpackaged file(s): /usr/lib/rpm/check-files /tmp/rpmbuild-lustre-build-184Ao2RN/BUILDROOT
/lustre-2.9.51_45_g3b3eeeb-1.x86_64
Wrote: /tmp/rpmbuild-lustre-build-184Ao2RN/RPMS/x86_64/lustre-2.9.51_45_g3b3eeeb-1.el7.centos.x86_64.rpm
Wrote: /tmp/rpmbuild-lustre-build-184Ao2RN/RPMS/x86_64/kmod-lustre-2.9.51_45_g3b3eeeb-1.el7.centos.x86_64.
rpm
Wrote: /tmp/rpmbuild-lustre-build-184Ao2RN/RPMS/x86_64/kmod-lustre-osd-zfs-2.9.51_45_g3b3eeeb-1.el7.
centos.x86_64.rpm
Wrote: /tmp/rpmbuild-lustre-build-184Ao2RN/RPMS/x86_64/lustre-osd-zfs-mount-2.9.51_45_g3b3eeeb-1.el7.
centos.x86_64.rpm
Wrote: /tmp/rpmbuild-lustre-build-184Ao2RN/RPMS/x86_64/lustre-tests-2.9.51_45_g3b3eeeb-1.el7.centos.
x86_64.rpm
Wrote: /tmp/rpmbuild-lustre-build-184Ao2RN/RPMS/x86_64/kmod-lustre-tests-2.9.51_45_g3b3eeeb-1.el7.centos.
x86_64.rpm
Wrote: /tmp/rpmbuild-lustre-build-184Ao2RN/RPMS/x86_64/lustre-iokit-2.9.51_45_g3b3eeeb-1.el7.centos.
x86_64.rpm
Wrote: /tmp/rpmbuild-lustre-build-184Ao2RN/RPMS/x86_64/lustre-debuginfo-2.9.51_45_g3b3eeeb-1.el7.centos.
x86_64.rpm
Executing(%clean): /bin/sh -e /tmp/rpmbuild-lustre-build-184Ao2RN/TMP/rpm-tmp.qsc8Zw
+ umask 022
+ cd /tmp/rpmbuild-lustre-build-184Ao2RN/BUILD
+ cd lustre-2.9.51_45_g3b3eeeb
+ rm -rf /tmp/rpmbuild-lustre-build-184Ao2RN/BUILDROOT/lustre-2.9.51_45_g3b3eeeb-1.x86_64
+ rm -rf /tmp/rpmbuild-lustre-build-184Ao2RN/TMP/kmp
+ exit 0
Executing(--clean): /bin/sh -e /tmp/rpmbuild-lustre-build-184Ao2RN/TMP/rpm-tmp.vzndd5
+ umask 022
+ cd /tmp/rpmbuild-lustre-build-184Ao2RN/BUILD
+ rm -rf lustre-2.9.51_45_g3b3eeeb
+ exit 0

```

3. You should now have build the following, similarly named, RPMs:

```

$ ls *.rpm
kmod-lustre-2.9.51_45_g3b3eeeb-1.el7.centos.x86_64.rpm
kmod-lustre-osd-zfs-2.9.51_45_g3b3eeeb-1.el7.centos.x86_64.rpm
kmod-lustre-tests-2.9.51_45_g3b3eeeb-1.el7.centos.x86_64.rpm
lustre-2.9.51_45_g3b3eeeb-1.el7.centos.x86_64.rpm
lustre-2.9.51_45_g3b3eeeb-1.src.rpm
lustre-debuginfo-2.9.51_45_g3b3eeeb-1.el7.centos.x86_64.rpm
lustre-iokit-2.9.51_45_g3b3eeeb-1.el7.centos.x86_64.rpm
lustre-osd-zfs-mount-2.9.51_45_g3b3eeeb-1.el7.centos.x86_64.rpm
lustre-tests-2.9.51_45_g3b3eeeb-1.el7.centos.x86_64.rpm

```

## 1.4.5. Installing Lustre.

Change to root and Change directory into ~build/lustre-release/:

```
# yum localinstall *.x86_64.rpm
```

## 1.4.6. Disable SELinux (Lustre Servers)

SELinux, which is on by default in RHEL/CentOS, will prevent the format commands for the various Lustre targets from completing. Therefore you must either disable it or adjust the settings. These instructions explain how to disable it.

1. Run `getenforce` to see if SELinux is enabled. It should return 'Enforcing' or 'Disabled'.
2. To disable it, edit `/etc/selinux/config` and change the line 'selinux=enforcing' to 'selinux=disabled'.
3. Finally, reboot your system.

```
# vi /etc/selinux/config

----
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
# enforcing - SELinux security policy is enforced.
# permissive - SELinux prints warnings instead of enforcing.
# disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of these two values:
# targeted - Only targeted network daemons are protected.
# strict - Full SELinux protection.
SELINUXTYPE=targeted
---
# shutdown -r now
```

## 1.4.7. Testing

Run "FSTYPE=zfs /usr/lib64/lustre/tests/llmount.sh" to start a test mount with the new modules:

```
# FSTYPE=zfs /usr/lib64/lustre/tests/llmount.sh
Stopping clients: onyx-2lvm5.onyx.whamcloud.com /mnt/lustre (opts:)
Stopping clients: onyx-2lvm5.onyx.whamcloud.com /mnt/lustre2 (opts:)
Loading modules from /usr/lib64/lustre/tests/..
detected 1 online CPUs by sysfs
libcfs will create CPU partition based on online CPUs
debug=vfstrace rpctrace dlmtrace neterror ha config          iocctl super lfsck
subsystem_debug=all
gss/krb5 is not supported
Formatting mgs, mds, osts
Format mds1: lustre-mdt1/mdt1
Format ost1: lustre-ost1/ost1
Format ost2: lustre-ost2/ost2
Checking servers environments
Checking clients onyx-2lvm5.onyx.whamcloud.com environments
Loading modules from /usr/lib64/lustre/tests/..
detected 1 online CPUs by sysfs
libcfs will create CPU partition based on online CPUs
debug=vfstrace rpctrace dlmtrace neterror ha config          iocctl super lfsck
subsystem_debug=all
gss/krb5 is not supported
Setup mgs, mdt, osts
Starting mds1:  lustre-mdt1/mdt1 /mnt/lustre-mds1
Commit the device label on lustre-mdt1/mdt1
Started lustre-MDT0000
Starting ost1:  lustre-ost1/ost1 /mnt/lustre-ost1
Commit the device label on lustre-ost1/ost1
Started lustre-OST0000
Starting ost2:  lustre-ost2/ost2 /mnt/lustre-ost2
Commit the device label on lustre-ost2/ost2
Started lustre-OST0001
Starting client: onyx-2lvm5.onyx.whamcloud.com:  -o user_xattr,flock onyx-2lvm5.onyx.whamcloud.com@tcp:/lustre
/mnt/lustre
UUID                1K-blocks      Used  Available Use% Mounted on
lustre-MDT0000_UUID  130688         1792   126848    1% /mnt/lustre[MDT:0]
lustre-OST0000_UUID  343936         1664   340224    0% /mnt/lustre[OST:0]
lustre-OST0001_UUID  343936         1408   340480    0% /mnt/lustre[OST:1]

filesystem_summary:  687872         3072   680704    0% /mnt/lustre

Using TIMEOUT=20
setting jobstats to procname_uid
Setting lustre.sys.jobid_var from disable to procname_uid
Waiting 90 secs for update
disable quota as required
```

You will now have a Lustre filesystem available at /mnt/lustre for testing. Note that this is just a temporary filesystem for testing purposes and will be reformatted the next time llmount.sh is run.