

Changing Test Parameters with Gerrit Commit Messages

- [Introduction](#)
- [Build Parameters](#)
- [General Parameters](#)
- [Node Parameters](#)
- [Versions](#)
- [Test Groups](#)

Introduction

There may be times that you wish to influence the building and testing carried out on your change. You might for example be fixing an issue that affects a particular distribution, CPU architecture, interoperability with a particular Lustre version, or fails only intermittently and needs multiple test runs to confirm it is fixed. To address these needs, you can change or add the tests carried out by indicating in the commit message the changes you require. Test - Parameter sessions are normally ***in addition*** to the normal test sessions that would be run against a patch. This allows a patch with specific or unusual testing requirements to ensure that sufficient *additional* testing is run to gain confidence in the change being made. For patches that are of an experimental nature (i.e. developer is not sure of functionality, or only wants a limited set of tests to be run just to try something), it is also possible to submit a patch with the [fortestonly](#) parameter.

To run additional sessions for your patch, add Test - Parameters: with space-separated name value pairs to your commit message (order within the line does not matter).

For example:

```
Test-Parameters: testlist=conf-sanity env=ONLY=61,ONLY_REPEAT=20
```

This will cause Autotest to run the normal test sessions plus one **additional** session where only `conf-sanity.sh test_61` will run, and it will be repeated 20 times. The parameters specified in Test - Parameters **do not** affect the existing sessions that are run by default.

Multiple 'Test-Parameters:' lines can be defined, with each Test - Parameters line triggering a **separate** test session to run in **parallel** with the default test sessions (parameters are **not** carried over across multiple lines):

```
Test-Parameters: testgroup=review-ldiskfs clientdistro=sles12sp3 serverdistro=el9.3
Test-Parameters: testgroup=review-ldiskfs clientdistro=ubuntu1804 serverdistro=el7.9
```

Long lists can be catered for by escaping the carriage return, but are otherwise not subject to the 70-character limit for commit messages:

```
Test-Parameters: ostcount=2 clients=1 ostsizegb=2 mdssizegb=2 env=SLOW=yes \
testlist=sanity,liblustre
```

Quotations can be used when spaces are necessary in a value:

```
Test-Parameters: testlist=sanity env=SANITY_EXCEPT="101g 102i"
```

The test parameter sessions can be influenced in many ways, see the [General Parameters](#) and [Node Parameters](#) sections below for all of the options.

Build Parameters

Below are build parameters read by Jenkins, and the Lustre Janitor.

forjanitoronly | flag

Signals to all components the patch is intended for Lustre Janitor testing only. Jenkins will not build the patch (the status will be **FAILURE**) and therefore, Autotest will not test it. Your patch will not receive a +/- 1 from Maloo.

```
Test-Parameters: forjanitoronly
```

ignore | flag

Signals to all components the patch should be completely ignored. Jenkins will not build the patch (the status will be **FAILURE**) and as a result it will not be tested by Autotest or the Lustre Janitor. Your patch will not receive a +/- 1 from Maloo.

```
Test-Parameters: ignore
```

General Parameters

Below is the list of general test parameters that can be used to run custom test sessions. These parameters differ from the [node_parameters](#) in that these do not need to be specified with a node type prefix.

For all of the examples below the 'Test-Parameters:' marker has been omitted to simplify the examples.

austeroptions | String

Pass through options for auster.

Valid values: See "Auster usage help" section on the [Setting up a Lustre Test Environment](#) wiki.

```
austeroptions=-R
```

combinedmdsmgs | Boolean

When true, the MGT will share a partition with an MDT. When false, AT will create an additional partition to be used by just the MGT.

NOTE: mdfilesystemtype will override mgfilesystemtype when combinedmdsmgs is true. If standalonemgs is set to true, this option will be ignored.

Valid values: true, false (no value is the same as true)

```
# creates a shared partition to be used by the MGT and a MDT - typically not necessary since this is the
default
combinedmdsmgs
# or
combinedmdsmgs=true

# creates an additional partition to be used by the MGT
combinedmdsmgs=false
```

env, envdefinitions | String

Comma separated environment definitions passed to the test environment. For definitions requiring spaces, enclose them in quotations.

NOTE: Be very careful setting environment variables directly (for example OSTFSTYPE=zfs) because Autotest creates a config file based on the environment it builds. If you ask for something at odds with Autotest's expectations you will see failure instead of success. In this case, for example, you should use the filesystemtype keyword described on this page. Autotest will then create the appropriate environment variables, the same is true for other things like ostcount instead of OSTCOUNT, ostsizegb instead of OSTSIZE, etc. Only use the env parameter when a direct variable is not listed here.

```
env=SLOW=yes
envdefinitions=SLOW=yes

# multiple definitions
env=ONLY=123,HONOR_EXCEPT=y

# one with spaces that must be enclosed in quotations
env=SLOW=yes,SANITY_EXCEPT="101g 102i"
```

facet | String

Used to specify which node should run the test framework. Default is client1.

Valid values: mds1, oss1, client2, etc.

```
# specify mds1 as the facet
facet=mds1

# invalid facet value, will cause an error message to be posted to the Gerrit review
mdscount=1 facet=mds2
```

failover | Boolean

Setup cluster in failover configuration

Valid values: true, false (no value is the same as true)

```
# enable failover
failover
# or
failover=true

# disable failover setup
failover=false
```

forbuildonly | Boolean

Patch will be built by Jenkins, but Autotest will not run any testing.

Valid values: true, false (no value is the same as true)

```
# mark build as for build only
forbuildonly
# or
forbuildonly=true

# disable forbuildonly - typically not necessary since the default is false
forbuildonly=false
```

fortestonly | Boolean

The fortestonly parameter marks that the patch is ***not intended for landing***. Any testlist= specified with Test-Parameters: fortestonly will **replace** the default tests that will be run, so it is possible to run only a subset of tests. Patches marked with fortestonly will **not** receive the Verified +1 label from Maloo and **cannot be landed**. In order to land such a patch, it should be rebased and submitted **without** the fortestonly keyword once the patch is known to be good, or the commit comment in Gerrit should be edited to remove the fortestonly label which will rebuild and retest the patch and will preserve any reviews that the patch has received).

Valid values: true, false (no value is the same as true)

```
# mark build as for test only
fortestonly
# or
fortestonly=true

# mark patch as experimental and run only a custom list of test
fortestonly testlist=conf-sanity

# disable fortestonly - typically not necessary since the default is false
fortestonly=false
```

fstype | String

Sets the file system type for all server nodes (MDS, MGS and OSS).

NOTE: fstype should not be used in combination with <mdt/mgt/ost>filesystemtype.

Valid values: ldiskfs, zfs

```
# set all server nodes to use zfs
fstype=zfs

# set all server nodes to use ldiskfs
fstype=ldiskfs

# this may result in an unexpected configuration
fstype=zfs mdtfilesystemtype=ldiskfs
```

iscsi | boolean

Use iSCSI for failover testing

Valid values: 0 (no iSCSI), 1 (iSCSI)

```
# do not setup iscsi
iscsi=0

# setup iscsi
iscsi=1
```

kerberos | boolean

Set the cluster up to use Kerberos

Valid values: true, false (no value is the same as true)

```
# setup kerberos
kerberos
# or
kerberos=true

# do not setup kerberos
kerberos=false
```

livedebug

For DDN employees only: [How to: livedebug](#)

mdscount | Integer

The total number of MDS nodes in the cluster.

Valid values: 1 - 8 MDS nodes

```
# create 2 MDS nodes with 4 MDTs each
mdscount=2 mdtcount=8

# create 4 MDS nodes with 1 MDT each
mdscount=4 mdtcount=4
```

mdssizegb | Integer

Sets the size of the partitions on the MDS in GB.

Valid values: > 0 and must be an integer. When setting this value, take into consideration your mdtcount value and that most test nodes have ~90GB of disk space.

```
# configure 2GB partitions on the MDS
mdssizegb=2
```

mdtcount | Integer

The total number of MDTs spread across **all** of the MDSs.

Valid values: 1 - 4 per MDS

```
# create 4 MDTs per MDS
mdscount=2 mdtcount=8

# create 1 MDT per MDS
mdscount=4 mdtcount=4
```

mdtfilesystemtype | String

Configure the file system type to use on the MDTs.

NOTE: <mdt/mgt/ost>filesystemtype should not be used in combination with fstype.

Valid values: ldiskfs, zfs

```
# configure the MDTs to use zfs
mdtfilesystemtype=zfs

# configure the MDTs to use ldiskfs
mdtfilesystemtype=ldiskfs

# this may result in an unexpected configuration
fstype=zfs mdtfilesystemtype=ldiskfs
```

mgssizegb | Integer

Sets the size of the partition on the MGS in GB. This value is only applicable when the session is using a stand alone MGS.

Valid values: > 0 and must be an integer. When setting this value, take into consideration that most test nodes have ~90GB of disk space.

```
# configure a 2GB partition on the MGS
mgssizegb=2
```

mgtfilingssystemtype | String

Configure the file system type to use on the MGTs.

NOTE: The default value for [combinedmdsmgs](#) is true, therefore it must be set to false in order to configure a different file system type for the MGT. Also, <mdt/mgt/ost>filesystemtype should not be used in combination with [fstype](#).

Valid values: ldiskfs, zfs

```
# configure the MGTs to use zfs
mgtfilingssystemtype=zfs combinedmdsmgs=false

# configure the MGTs to use ldiskfs
mgtfilingssystemtype=ldiskfs combinedmdsmgs=false

# since combinedmdsmgs defaults to true, mgtfilingssystemtype will be ignored and
# the MGT will use ldiskfs
mdtfilingssystemtype=ldiskfs mgtfilingssystemtype=zfs

# do not do this - may result in an unexpected configuration
fstype=zfs mgtfilingssystemtype=ldiskfs
```

optional | Boolean

Marks the test session as optional: does not impact the verified value from Maloo and is only run if resources are immediately available.

Valid values: true, false (no value is the same as true)

```
# mark a session as optional
optional
# or
optional=true

# mark a session as required - typically not necessary since false is the default
optional=false
```

ostcount | Integer

The number of OSTs per OSS.

Valid values: 1 - 8

```
# create 8 OSTs per OSS
osscount=2 ostcount=8

# create 4 OSTs per OSS
osscount=4 ostcount=4
```

ostfilesystemtype | String

Configure the file system type to use on the OSTs.

NOTE: <mdt/mgt/ost>filesystemtype should not be used in combination with [fstype](#).

Valid values: ldiskfs, zfs

```
# configure the OSTs to use zfs
ostfilesystemtype=zfs

# configure the OSTs to use ldiskfs
ostfilesystemtype=ldiskfs

# do not do this - may result in an unexpected configuration
fstype=zfs ostfilesystemtype=ldiskfs
```

ostsizegb | Integer

Sets the size of the partitions on the OSS in GB.

Valid values: > 0 and must be an integer. When setting this value, take into consideration your ostcount value and that most test nodes have ~90GB of disk space.

```
# configure 2GB partitions on the OSS
ostsizegb=2
```

resumeaftercrash | Boolean

When true, Autotest will continue the session after a suite crashes. The session will continue with the suite following the one that crashed. When false, Autotest will stop the session and upload the results to Maloo immediately.

Valid values: true, false (no value is the same as true)

```
# configure the session to continue after a suite crashes - typically not necessary since the default is true
resumeaftercrash
# or
resumeaftercrash=true

# configure the session to stop after a suite crashes
resumeaftercrash=false
```

signofftest | Boolean

Marks the session as enforcing: is used to determine the verified value from Maloo.

Valid values: true, false (no value is the same as true)

```
# mark the session as enforcing - typically not needed since the default is true
signofftest
# or
signofftest=true

# mark the session as not enforcing
signofftest=false
```

standalonemgs | Boolean

Provisions a separate test node as the MGS. The MGS will be setup with the same parameters as the MDS unless they're overwritten.

Valid values: true, false (no value is the same as true)

```
# configure the session to run with a stand alone MGS
standalonemgs
# or
standalonemgs=true

# configure the session to run with the MGS on the MDS but separate partitions
standalonemgs=false combinedmdsmgs=false

# configure the session to run with the MGS on the MDS and have the MGT share
# a partition with a MDT
standalonemgs=false combinedmdsmgs=true

# configure the session to run with a stand alone MGS with a different file
# system type than the OSS and MDS
standalonemgs mgtfilesystemtype=zfs mdtfilesystemtype=ldiskfs ostfilesystemtype
```

subproject, @ | Boolean

Specifies the subproject of the patch. Subprojects must be added by Autotest Admin, currently the only configured subproject is lnet.

Valid values: lnet

```
# configure the session to run the lnet subproject test sessions
@lnet
# or
subproject=lnet
```

testgroup | String

Configure the test session to run a specific test group. Specifying a test group makes it easy to run a typical test grouping with small modifications since the session will inherit all of the values from the base test with any overrides applied (see the code block below for examples).

NOTE: testgroup can be combined with [testlist](#) to run a test group plus additional suites

Valid values: See the [Test Groups](#) section for a complete list

```
# run a typical review-ldiskfs session with a stand alone MGS
testgroup=review-ldiskfs standalonemgs

# run failover with a specific server version
testgroup=failover serverversion=2.11.50

# run review-dne-part-1 with additional suites
testgroup=review-dne-part-1 testlist=sanity-lfsck, sanity-sec
```

testlist | String

Configure the test session to run a specific list of suites. testlist can be combined with [testgroup](#) to run a test group plus additional suites.

Valid values: sanity, sanitytn, conf-sanity, mmp, replay-single, replay-dual, lnet-selftest, etc.

```
# run a custom list of suites
testlist=sanity,sanity-sec,sanity-hsm

# run review-dne-part-1 with additional suites
testlist=sanity-lfsck,sanity-sec testgroup=review-dne-part-1
```

timeoutreprovision | Boolean

Configures the session to re-provision all of the nodes instead of simply reboot them.

Valid values: true, false (no value is the same as true)

```
# re-provision nodes after a timeout
timeoutreprovision
# or
timeoutreprovision=true

# reboot the nodes after a timeout - typically not necessary since reboot is the default
timeoutreprovision=false
```

trivial | Boolean

The trivial keyword can be used to reduce the testing time (both wall-clock time as well as total test system hours) for patches that do not affect code functionality, such as changes to whitespace, comments, man pages, and test scripts (in conjunction with the testlist keyword to ensure the modified test script is run if it isn't already). Patches marked with trivial in the Test-Parameters: list will run the review-ldiskfs test session instead of the regular tests, so it currently runs only **sanity** and **lnet-selftest** to ensure basic functionality. If changes are only being made to a test script outside of those already run by default, an additional testlist=<test-script(s)>keyword should be added with a comma-separated list of modified tests to ensure they are run to validate the changes. The trivial keyword will reduce the test completion time from approximately 10h elapsed and 30h of total test system time to approximately 3h elapsed/system time. This saves test system resources that may be better spent on other patches.

Valid values: true, false (no value is the same as true)

```
# mark the patch as trivial
trivial
# or
trivial=true

# mark the patch as non-trivial - typically not necessary since this is the default
trivial=false
```

Node Parameters

Node parameters are used to change how specific node types are configured. They must be prefixed with the node type being changed. Valid node types are **client**, **mds**, **mgs**, **oss** and **server**. Server is an alias that allows users to modify a value for all server node types (mds, mgs and oss). For example instead of writing

```
mdsdistro=e17 mgsdistro=e17 ossdistro=e17
```

Users can simply write

```
serverdistro=e17
```

arch | String or Array

Sets the architecture for the specified node type

Valid values: x86_64, ppc64, aarch64 (architecture must have been built for patch)

```
# clients will use x86_64
clientarch=x86_64
```

buildno | Integer

Used in conjunction with [job](#) to install a specific build on the specified node. [job](#) must be specified with buildno and [version](#) cannot be specified with buildno.

Valid values: Any valid Jenkins build number for the specified job

```
# install the server with a specific job/build combination
serverbuildno=343 serverjob=lustre-master

# this would fail since serverjob is not specified
serverbuildno=343
```

count | Integer

The number of nodes to use for the specified node type.

NOTE: For MDS and OSS nodes, it's best to also set [mdtcount](#) / [ostcount](#) to ensure you have the expected number of targets.

Valid values:

clientcount	2 - 4
mdscount	1 - 6
osscount	1 - 4

```
# configure session to run with 4 clients, 2 MDSs and 1 OSS
clientcount=4 mdscount=2 osscount=1
```

distro | String

Distribution to use for the specified node type.

Valid values: el7, el7.5, el7.6, ubuntu1604, ubuntu1804, sles12sp3, etc. (The distribution specified must have been built for the patch being tested. The distros built can be seen on the patch's build page in [Jenkins](#).)

```
# specify the client to use el7 and all servers sles12sp3
clientdistro=el7 serverdistro=sles12sp3

# specify the client to use ubuntu1804, MDS to use el7 and OSS to use sles12sp3
clientdistro=ubuntu1804 mdsdistro=el7 ossdistro=sles12sp3
```

ibstack | String

The IB stack to use for the specified node type.

Valid values: inkernel, ofa (The IB stack type must have been built for the build being tested. The stack types can be seen on the patch's build page in [Jenkins](#).)

```
clientibstack=inkernel
```

job | String

Used with buildno to install a specific build on the specified node. [buildno](#) must be specified with job. version cannot be specified with job.

Valid values: Any valid Jenkins job, such as lustre-reviews, lustre-master, etc.

```
# install the server with a specific job/build combination
serverjob=lustre-b2_10 serverbuildno=123

# this would fail since buildno is not specified for client
serverjob=lustre-b2_10 serverbuildno=123 clientjob=lustre-master
```

selinux | Boolean

Enables selinux on the specified node type.

Valid values: true, false (no value is the same as true)

```
# enables selinux on clients
clientselinux
# or
clientselinux=true

# disables selinux on clients - not typically necessary since false is the default
clientselinux=false
```

version | String

The version of Lustre to use for the specified node type.

Version **cannot** be specified with job and build.

If a distro is not specified in the test parameters, Autotest will use the highest **el** version available for the specified version.

Valid values: For a list of valid versions see the [Versions](#) section.

```
# specify patch level versions
clientversion=2.10.3 serverversion=2.11.0

# specify minor versions which uses the highest patch level version
clientversion=2.10 serverversion=2.11

# specify Exascaler tags in the -ddn[0-9] format
clientversion=2.12.2-ddn2 serverversion=2.14.0-ddn23

# specify Exascaler tags in the EXAx.x.x format
clientversion=EXA5.0.1 serverversion=EXA6.0.0

# specify Exascaler tags in the EXAx format which uses the highest minor version
clientversion=EXA5 serverversion=EXA6
```

Note that in addition to the more commonly used serverversion= option, it is possible (to a limited extent) to request separate ossversion= and mdsversion= if necessary.

Versions

Versions are pointers to job/build combinations and simplify using a specific Lustre version on a test node. Versions can be specified in the test parameters using the [version](#) node parameter.

The versions in parenthesis auto-update as new versions are added. For example 1.9 will always point to the highest 1.9.x version. EXA1 will always point to the highest EXA1.x.x version.

If there is a version missing, send an email to charlie@whamcloud.com

Version	Build
2.7.0 (2.7)	https://build.whamcloud.com/job/lustre-b2_7/29/
2.8.0 (2.8)	https://build.whamcloud.com/job/lustre-b2_8/12/
2.9.0 (2.9)	https://build.whamcloud.com/job/lustre-b2_9/22/
2.10.0	https://build.whamcloud.com/job/lustre-b2_10/5/
2.10.1	https://build.whamcloud.com/job/lustre-b2_10/30/
2.10.2	https://build.whamcloud.com/job/lustre-b2_10/52/
2.10.3	https://build.whamcloud.com/job/lustre-b2_10/69/
2.10.4	https://build.whamcloud.com/job/lustre-b2_10/116/
2.10.5	https://build.whamcloud.com/job/lustre-b2_10/136/
2.10.6	https://build.whamcloud.com/job/lustre-b2_10/152/
2.10.7	https://build.whamcloud.com/job/lustre-b2_10/168/
2.10.8 (2.10)	https://build.whamcloud.com/job/lustre-b2_10/170/
2.11.0 (2.11)	https://build.whamcloud.com/job/lustre-b2_11/2/

2.12.0	https://build.whamcloud.com/job/lustre-b2_12/2/
2.12.1	https://build.whamcloud.com/job/lustre-b2_12/13/
2.12.2	https://build.whamcloud.com/job/lustre-b2_12/18/
2.12.3	https://build.whamcloud.com/job/lustre-b2_12/53/
2.12.4	https://build.whamcloud.com/job/lustre-b2_12/66/
2.12.5	https://build.whamcloud.com/job/lustre-b2_12/87/
2.12.6	https://build.whamcloud.com/job/lustre-b2_12/112/
2.12.7	https://build.whamcloud.com/job/lustre-b2_12/135/
2.12.8	https://build.whamcloud.com/job/lustre-b2_12/152/
2.12.9 (2.12)	https://build.whamcloud.com/job/lustre-b2_12/164/
2.13.0 (2.13)	https://build.whamcloud.com/job/lustre-b2_13/6/
2.14.0 (2.14)	https://build.whamcloud.com/job/lustre-b2_14/2/
2.15.0	https://build.whamcloud.com/job/lustre-b2_15/13/
2.15.1	https://build.whamcloud.com/job/lustre-b2_15/28/
2.15.2	https://build.whamcloud.com/job/lustre-b2_15/48/
2.15.3	https://build.whamcloud.com/job/lustre-b2_15/65/
2.15.4 (2.15)	https://build.whamcloud.com/job/lustre-b2_15/81
EXAScaler Versions	

Test Groups

Test groups are set lists of Lustre test suites managed by Autotest.

Name	Suites
basic	conf-sanity
failover-part-1	replay-vbr, replay-dual, replay-single, mmp, replay-ost-single, recovery-small, recovery-double-scale
failover-part-2	recovery-random-scale
failover-part-3	recovery-mds-scale
failover-zfs-part-1	replay-vbr, replay-dual, replay-single, mmp, replay-ost-single, recovery-small, recovery-double-scale
failover-zfs-part-2	recovery-random-scale
failover-zfs-part-3	recovery-mds-scale
full-dne-part-1	sanity-scrub, replay-single, obdfilter-survey, replay-ost-single, large-scale, insanity, parallel-scale, runtests, replay-dual, sanity-flr, sanity-lsnapshot, mmp, sanity-dom, mds-survey, sanity-lfscck, sanity-lnet, pjdfstest, ost-pools, recovery-small
full-dne-part-2	lnet-selftest, sanity, sanity-hsm, lustre-rsync-test, sanity-sec, replay-vbr, parallel-scale-nfsv3, sanity-quota, sanity-pcc, sanity-lipe-find3, racer
full-dne-part-3	sanity-pfl, performance-sanity, sanity-benchmark, conf-sanity, sanityn, parallel-scale-nfsv4, hot-pools, sanity-lipe, sanity-lipe-scan3

full-dne-zfs-part-1	sanity-scrub, replay-single, obdfilter-survey, replay-ost-single, large-scale, insanity, parallel-scale, runtests, replay-dual, sanity-flr, sanity-lsnapshot, mmp, sanity-dom, mds-survey, sanity-lfsck, sanity-lnet, pjdfstest, ost-pools, recovery-small
full-dne-zfs-part-2	lnet-selftest, sanity, sanity-hsm, lustre-rsync-test, sanity-sec, replay-vbr, parallel-scale-nfsv3, sanity-quota, sanity-pcc, sanity-lipe-find3, racer
full-dne-zfs-part-3	sanity-pfl, performance-sanity, sanity-benchmark, conf-sanity, sanityn, parallel-scale-nfsv4, hot-pools, sanity-lipe, sanity-lipe-scan3
full-part-1	sanity-scrub, replay-single, obdfilter-survey, replay-ost-single, large-scale, insanity, parallel-scale, runtests, replay-dual, sanity-flr, sanity-lsnapshot, mmp, sanity-dom, mds-survey, sanity-lfsck, sanity-lnet, pjdfstest, ost-pools, recovery-small
full-part-2	lnet-selftest, sanity, sanity-hsm, lustre-rsync-test, sanity-sec, replay-vbr, parallel-scale-nfsv3, sanity-quota, sanity-pcc, sanity-lipe-find3, racer
full-part-3	sanity-pfl, performance-sanity, sanity-benchmark, conf-sanity, sanityn, parallel-scale-nfsv4, hot-pools, sanity-lipe, sanity-lipe-scan3
full-patchless	lnet-selftest, runtests, sanity, sanity-scrub, sanity-benchmark, sanity-lfsck, sanityn, sanity-hsm, sanity-flr, sanity-dom, sanity-lsnapshot, insanity, sanity-quota, sanity-sec, sanity-pfl, lustre-rsync-test, ost-pools, mds-survey, mmp, performance-sanity, parallel-scale, large-scale, obdfilter-survey, parallel-scale-nfsv3, parallel-scale-nfsv4, pjdfstest, sanity-lnet, racer
full-zfs-dkms	sanity-scrub, replay-single, obdfilter-survey, replay-ost-single, large-scale, insanity, parallel-scale, runtests, replay-dual, sanity-flr, sanity-lsnapshot, mmp, sanity-dom, mds-survey, sanity-lfsck, sanity-lnet, pjdfstest, ost-pools, recovery-small
full-zfs-part-1	sanity-scrub, replay-single, obdfilter-survey, replay-ost-single, large-scale, insanity, parallel-scale, runtests, replay-dual, sanity-flr, sanity-lsnapshot, mmp, sanity-dom, mds-survey, sanity-lfsck, sanity-lnet, pjdfstest, ost-pools, recovery-small
full-zfs-part-2	lnet-selftest, sanity, sanity-hsm, lustre-rsync-test, sanity-sec, replay-vbr, parallel-scale-nfsv3, sanity-quota, sanity-pcc, sanity-lipe-find3, racer
full-zfs-part-3	sanity-pfl, performance-sanity, sanity-benchmark, conf-sanity, sanityn, parallel-scale-nfsv4, hot-pools, sanity-lipe, sanity-lipe-scan3
lnet-review-ldiskfs-dne	lnet-selftest, sanity, sanity-lnet, sanity-sec
review	lnet-selftest, runtests, sanity, sanityn, replay-single, conf-sanity, recovery-small, replay-ost-single, insanity, sanity-quota, lustre-rsync-test, ost-pools, sanity-lfsck, sanity-hsm, sanity-lnet
review-dne-part-1	sanity, sanity-pfl
review-dne-part-2	mds-survey, replay-dual, runtests, sanity-lfsck, sanity-sec
review-dne-part-3	conf-sanity
review-dne-part-4	insanity, mmp, replay-ost-single, sanity-dom, sanity-flr, sanity-hsm, sanity-quota
review-dne-part-5	lustre-rsync-test, recovery-small, sanity-scrub, sanityn
review-dne-part-6	replay-single, ost-pools
review-dne-part-7	large-scale, hot-pools, sanity-pcc, sanity-lipe
review-dne-part-8	parallel-scale, replay-vbr, replay-dual, racer
review-dne-part-9	obdfilter-survey, performance-sanity, sanity-benchmark, parallel-scale-nfsv4, parallel-scale-nfsv3

review-dne-selinux-ssk-part-1	sanity
review-dne-selinux-ssk-part-2	recovery-small, sanity-sec, sanity-selinux
review-dne-zfs	runtests, sanity, sanityn, replay-single, conf-sanity, recovery-small, replay-ost-single, insanity, sanity-quota, lustre-rsync-test, ost-pools, sanity-lfsck, sanity-hsm
review-dne-zfs-part-1	sanity, sanity-pfl
review-dne-zfs-part-2	mds-survey, replay-dual, runtests, sanity-lfsck, sanity-sec
review-dne-zfs-part-3	conf-sanity
review-dne-zfs-part-4	insanity, mmp, replay-ost-single, sanity-dom, sanity-flr, sanity-hsm, sanity-quota
review-dne-zfs-part-5	lustre-rsync-test, recovery-small, sanity-scrub, sanityn
review-dne-zfs-part-6	replay-single, ost-pools
review-dne-zfs-part-7	large-scale, hot-pools, sanity-pcc, sanity-lipe
review-exa6	sanity-pcc, sanity-sec
review-lldiskfs	lnet-selftest, sanity, sanity-lnet
review-lldiskfs-arm	lnet-selftest, sanity, sanity-lnet
review-lldiskfs-dne	lnet-selftest, sanity, sanity-lnet
review-lldiskfs-dne-arm	lnet-selftest, sanity, sanity-lnet
review-lldiskfs-ubuntu	lnet-selftest, sanity, sanity-lnet
review-zfs	sanity-quota, sanity-flr, replay-single, replay-ost-single, insanity
review-zfs-part-1	runtests, sanity, sanityn, sanity-quota, ost-pools, sanity-lfsck, sanity-hsm, sanity-flr
review-zfs-part-2	replay-single, conf-sanity, recovery-small, replay-ost-single, insanity, lustre-rsync-test, large-scale, mds-survey
tiny	sanity, mmp