

CloudLinux Documentation

© 2015 Cloud Linux Inc



Note:

To change the product logo for your own print manual or PDF, click "Tools > Manual Designer" and modify the print manual template.

CloudLinux Documentation

© 2015 Cloud Linux Inc

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: February 2015 in (wherever you are located)

Table of Contents

Foreword	0
Part I Installation	8
1 Converting existing servers.....	8
Advanced options for cldploy	8
Explanation Of Changes	9
2 Installing new servers.....	10
3 Installation on Amazon EC2.....	10
4 VMware Images.....	10
5 Xen Images.....	10
6 KVM Images.....	11
7 Net Install.....	11
8 Installing on H-Sphere Server.....	12
Converting from mod_fastcgi to mod_fcgid	13
9 Virtuozzo and OpenVZ.....	15
10 Getting Trial License	16
11 Registering CloudLinux Server.....	17
12 CloudLinux on DigitalOcean.....	17
13 Servers with LILO boot loader.....	17
14 Uninstalling CloudLinux.....	18
Part II Limits	19
1 Understanding LVE.....	20
2 Command Line Tools.....	22
lvectl	22
lveps	23
lvetop	23
cldetect	24
lve-stats	24
Storing statistics in MySQL.....	26
Storing statistics in PostgreSQL.....	30
Compacting in multi-server settings.....	33
3 SPEED Limits.....	33
4 CPU Limits.....	34
5 Memory Limits.....	34
6 IO Limits.....	35
7 IOPS Limits.....	35
8 Number of Processes Limit.....	35
9 Entry Processes Limit.....	35
10 Compatibility Matrix.....	36

11	Integration Components.....	36
	LVEPAM module	37
	LVEWrappers	37
	MPM ITK support	38
	HostingLimits module for Apache	38
	Redis Support for HostingLimits.....	43
	cPanel/WHM JSON API	44
Part III LVE Manager		45
1	cPanel LVE Manager.....	45
	LVEExtensions for cPanel	50
Part IV CageFS		51
1	Installation.....	51
2	Uninstalling CageFS.....	52
3	Managing Users.....	52
4	Command line tools.....	53
5	Running Command Inside CageFS.....	54
6	CageFS Quirks.....	54
7	Configuration.....	55
	File System Templates	55
	Excluding files	55
	Excluding Users	56
	Mount Points	56
	Per user virtual mount points.....	56
	Split by username.....	57
	Base Home Directory	58
	PostgreSQL support	59
	PAM configuration	59
	Executing By Proxy	59
	Custom /etc direcotry	60
	Moving cagefs-skeleton directory	60
	Moving /var/cagefs directory	60
	TMP directories	61
	Syslog	61
8	Control Panel Integration.....	61
	cPanel	62
	Plesk	63
	ISPManager	64
Part V MySQL Governor		66
1	Installation.....	66
2	Removing MySQL Governor.....	66
3	Modes Of Operation.....	67
4	Configuration.....	67
5	Starting And Stopping.....	69
6	User to Database mapping.....	69

7	Log Files.....	70
8	Change MySQL version.....	70
9	Command Line Tools.....	71
	dbtop	71
	dbctl	72
	lveinfo --dbgov	73
	dbgovchart	75
10	Backing Up MySQL.....	77
11	abrt plugin.....	77
Part VI PHP Selector		79
1	Installation.....	79
	LiteSpeed support	79
	ISPmanager support	80
2	Configuration.....	81
	Setting default version and modules	81
	Individual PHP.ini files	81
	Substitute global php.ini for individual customer	82
	Managing interpreter version	82
	Including PHP Selector only with some packages (cPanel)	83
3	Command Line Tools.....	84
	selectorctl	84
	cl-selector	88
	piniset - php.ini options	89
	Integrating With Control Panels	89
4	Removing PHP Selector.....	91
5	Using PHP Selector.....	91
6	Custom PHP.ini options.....	93
7	End user directories.....	95
8	Compiling your own extensions.....	95
9	Roll your own PHP.....	96
10	Detect User's PHP Version.....	96
11	Bundled PHP Extensions.....	97
	PHP 4.4 Extensions	97
	PHP 5.1 Extensions	97
	PHP 5.2 Extensions	98
	PHP 5.3 Extensions	98
	PHP 5.4 Extensions	99
	PHP 5.5 Extensions	100
12	Disabling PHP extensions.....	100
Part VII Python and Ruby Selector		101
1	Installation.....	101
2	End User Access.....	101
3	Command Line	104

Part VIII inodes Limits	105
1 Command Line Tool.....	107
Part IX Kernel Settings	108
1 Virtualized /proc filesystem.....	108
2 SecureLinks.....	109
3 ptrace Block.....	109
4 Xen XVDA detection.....	110
5 TPE Extension.....	110
6 IOLimits latency.....	110
7 Hybrid Kernel.....	111
8 Reading LVE usage.....	111
9 flashcache.....	112
Part X Apache mod_isapi	113
1 Installation.....	116
2 Uninstall.....	118
3 Troubleshooting mod_isapi.....	118
Part XI OptimumCache	122
1 Installation.....	122
2 Using without ploop.....	124
3 Marking Directories.....	124
4 Configuration File.....	126
5 Command Line Interface.....	126
6 Uninstall OptimumCache.....	127
7 Troubleshooting.....	128
Part XII Additional Packages	130
1 Git for cPanel.....	130
Part XIII Integration Guide	131
1 Common Questions.....	131
2 Displaying CPU, Memory & IO limits.....	131
3 Integrating LVE Limits with Packages.....	132
Part XIV Partner Portal	134
1 IP Reseller Partner UI.....	134
Part XV Hardware Compatibility	140

Part XVI Downloading Documentation**141****Index****142**

1 Installation

1.1 Converting existing servers

It is easy to switch server from CentOS 5.x or 6.x to CloudLinux. The process takes a few minutes and replaces just a handful of RPMs.

- Get <activation_key> either by getting [trial subscription](#) or by [purchasing subscription](#).
- Download script: [cldeploy](#)
- Execute `sh cldeploy -k <activation_key>` (if you have IP based license, execute `sh cldeploy -i`)
- Reboot

```
$ wget http://repo.cloudlinux.com/cloudlinux/sources/cln/cldeploy
$ sh cldeploy -k <activation_key> # if you have activation key
or
$ sh cldeploy -i # if you have IP based license
$ reboot
```

Once you have rebooted, you are running CloudLinux kernel with LVE enabled.

The script automatically detects and supports following control panels: cPanel, Plesk, ISPmanager, DirectAdmin, InterWorx. It will install CloudLinux kernel, [Apache module](#), [PAM module](#), [command line tools](#) as well as LVE Manager

** Note: If you are converting Hyper-V server, please, make sure you upgrade to latest CentOS 5.9 or CentOS 6.4 first*

1.1.1 Advanced options for cldeploy

```
# ./cldeploy --help
```

Usage:

```
-h, --help          Print this message
-k, --key <key>    Update your system to CloudLinux with activation key
-i, --byip         Update your system to CloudLinux and register by IP
-c, --uninstall    Convert CloudLinux back to CentOS
--components-only  Install control panel components only
--conversion-only  Do not install control panel components after converting
--hostinglimits    Install mod_hostinglimits rpm
```

The script will install following to the server:

1. Register server with CLN
2. Install CloudLinux kernel, lve libraries, lve-utils, lve-stats and pam_lve packages
3. It will attempt to detect control panel and do following actions
 - a. For cPanel & DirectAdmin
 - i. recompile Apache to install mod_hostinglimits
 - ii. install lve manager
 - b. For Plesk, ISPManager & InterWorx
 - i. Updates httpd and installs mod_hostinglimits
 - ii. install lve manager

To disable installation of LVE Manager and mod_hostinglimits, please use `--conversion-only` option
To disable installation of kernel & CLN registration, please use `--components-only` option

To install `mod_hostinglimits` only, use `--hostinglimits` option

Examples:

```
$ cldeploy --key xx-xxxxxx # convert RHEL/CentOS to CL by using
activation key, install control panel components
$ cldeploy --byip --conversion-only # convert RHEL/CentOS to CL by ip,
don't install control panel components
$ cldeploy --components-only # install control panel components on
already converted system
$ cldeploy --hostinglimits # update httpd and install
mod_hostinglimits
```

1.1.2 Explanation Of Changes

CloudLinux uses the fact that it is very close to CentOS and RHEL to convert systems in place, requiring just one reboot. Our conversion script does following actions:

- Backup of original repository settings into `/etc/cl-converted-saved`
- Backup of RHEL system id into `/etc/cl-converted-save` (RHEL systems only)
- Installs CL repository settings & imports CL RPM key
- Replaces `redhat/centos-release`, `redhat-release-notes`, `redhat-logos` with CL version
- Removes `cpuspeed` RPM (as it conflicts with CPU limits)
- Re-installs CL version of `rhnlib/rhnplugin`
- Checks for binary kernel modules, finds replacement if needed
- Detects OVH servers and fixes `mkinitrd` issues
- Detects Linode servers, and fixes `grub` issues
- Checks if `LES` is installed
- checks that `/etc/fstab` has correct `/dev/root`
- checks for `efi`
- Installs CL kernel, `lve-utils`, `liblve`, `lve-stats` RPMs
- Installs LVE Manager for `cPanel`, `Plesk`, `DirectAdmin`, `ISPManager` & `InterWorx`
- Installs `mod_hostinglimits` apache module
 - RPM install for `Plesk`, `ISPManager` & `InterWorx`
 - On `Plesk`, replaces `psa-mod_fcgid*` with `mod_fcgid`
 - `EasyApache` rebuild for `cPanel`
 - `custombuild` for `DA`

Script for converting back:

- Restores CentOS repositories, and `centos-release/release-notes/logos`
- Removes `lve`, `mod_hostinglimits`, `lve-stats`, `lvemanager`
- `mod_hostinglimits` RPM is removed

The kernel is not removed - to prevent condition when server has no kernels and wouldn't boot. The command line to remove the kernel is provided.

On `cPanel` servers, rebuild of Apache with `easyapache` will complete the conversion back, but doesn't have to be performed immediately

On `DirectAdmin` servers, rebuild of Apache with `custombuild` will complete the conversion back, but doesn't have to be performed immediately

1.2 Installing new servers

You can download latest CloudLinux ISO and use it to install CloudLinux on your server:

Latest stable CloudLinux 6.6 ISO:

x86_64 version: http://repo.cloudlinux.com/cloudlinux/6.6/iso/x86_64/CloudLinux-6.6-x86_64-DVD.iso

i386 version: <http://repo.cloudlinux.com/cloudlinux/6.6/iso/i386/CloudLinux-6.6-i386-DVD.iso>

Last Updated: Oct 28, 2014

Latest stable CloudLinux 5.11 ISO:

x86_64 version: http://repo.cloudlinux.com/cloudlinux/5.11/iso/x86_64/CloudLinux-5.11-x86_64-DVD.iso

i386 version: <http://repo.cloudlinux.com/cloudlinux/5.11/iso/i386/CloudLinux-5.11-i386-DVD.iso>

Last Updated: Oct 10, 2014

** Important: Once you install server from the ISO, make sure you run yum update, and then register your system.*

1.3 Installation on Amazon EC2

We have prepared images of CloudLinux, CloudLinux with cPanel and CloudLinux with Plesk that are available from Amazon marketplace:

CloudLinux 6.4 Minimal: <https://aws.amazon.com/marketplace/pp/B00ENMWBP8>

CloudLinux 6.4 with cPanel: <https://aws.amazon.com/marketplace/pp/B00C2EOLBG>

CloudLinux 6.4 for Parallels Plesk 11.5: <https://aws.amazon.com/marketplace/pp/B00C2DZ1WK>

1.4 VMware Images

To start using CloudLinux Images with VMware, create RH Enterprise 6 Virtual Machine. Set it to use one of the following virtual disks.

Root password: cloudlinux

Disk Images:

CloudLinux Minimal: <http://dropbox.cloudlinux.com/images/cl6/aws-cl6-hvm-base.img.tgz>

CloudLinux + cPanel: <http://dropbox.cloudlinux.com/images/cl6/aws-cl6-hvm-cpanel.vmdk>

CloudLinux + Parallels Plesk: <http://dropbox.cloudlinux.com/images/cl6/aws-cl6-plesk.vmdk>

CloudLinux + DirectAdmin: <http://dropbox.cloudlinux.com/images/cl6/aws-cl6-hvm-da.vmdk>

1.5 Xen Images

To start using Xen image:

Decompress xen image to: `/var/lib/xen/images/` (depends on your setup)

Create a config file in `/etc/xen`

Like:

```
name = "cl6-sample"
uuid = "4230bccf-5882-2ac6-7e1c-0e2a60208001"
maxmem = 1024
memory = 1024
vcpus = 1
bootloader = "/usr/bin/pygrub"
on_poweroff = "destroy"
on_reboot = "restart"
on_crash = "restart"
vfb = [ "type=vnc,vncunused=1,key=en-us" ]
disk = [ "tap:aio:/var/lib/xen/images/cl6-sample.img,sda,w" ]
vif = [ "mac=00:16:3e:23:09:10,bridge=xenbr0,script=vif-bridge" ]
```

where:

name = "cl6-sample" - unique name of the server

disk = ["tap:aio:/var/lib/xen/images/cl6-sample.img,sda,w"] - path to image file

uuid = "4230bccf-5882-2ac6-7e1c-0e2a60208001" - unique id for that server

vif = ["mac=00:16:3e:23:09:10,bridge=xenbr0,script=vif-bridge"] - unique MAC

[maxmem = 1024 memory = 1024 vcpus = 1] resources

Root password: cloudlinux

Disk Images

CloudLinux Minimal: <http://dropbox.cloudlinux.com/images/cl6/aws-cl6-hvm-base.img.tgz>

CloudLinux + cPanel: <http://dropbox.cloudlinux.com/images/cl6/aws-cl6-hvm-cpanel.img.tgz>

CloudLinux + Parallels Plesk: <http://dropbox.cloudlinux.com/images/cl6/aws-cl6-plesk.img.tgz>

CloudLinux + DirectAdmin: <http://dropbox.cloudlinux.com/images/cl6/aws-cl6-hvm-da.img.tgz>

1.6 KVM Images

To start using CloudLinux Images with KVM, create a VM using `virt-manager`, and add one of the images as a disk

Root password: cloudlinux

Disk Images

CloudLinux Minimal: <http://dropbox.cloudlinux.com/images/cl6/aws-cl6-hvm-base.qcow2>

CloudLinux + cPanel: <http://dropbox.cloudlinux.com/images/cl6/aws-cl6-hvm-cpanel.qcow2>

CloudLinux + Parallels Plesk: <http://dropbox.cloudlinux.com/images/cl6/aws-cl6-plesk.qcow2>

CloudLinux + DirectAdmin: <http://dropbox.cloudlinux.com/images/cl6/aws-cl6-hvm-da.qcow2>

1.7 Net Install

To install CloudLinux over network:

1. Download & boot from netboot image from: http://repo.cloudlinux.com/cloudlinux/6.6/iso/x86_64/CloudLinux-6.6-x86_64-netboot.iso. It will boot into CloudLinux installer.

Alternatively you can configure your PXE server using following folder as reference: http://repo.cloudlinux.com/cloudlinux/6.6/install/x86_64/images/pxeboot/

2. During the CloudLinux installation select URL as installation source and enter URL: <http://>

repo.cloudlinux.com/cloudlinux/6.6/install/x86_64/ and continue with installation. To install CloudLinux 5.10 instead of 6.6 use following URL: http://repo.cloudlinux.com/cloudlinux/5.10/netinstall/x86_64/
Same URLs can be used to install para-virtualized Xen using either command-line or virt manager.

1.8 Installing on H-Sphere Server

For H-Sphere 3.5+

Requirements

1. CloudLinux with liblve 0.8 or later
2. Apache 2.2.x or 1.3
3. mod_suexec should be enabled

To achieve optimal performance, we recommend to [convert from mod_fastcgi to mod_fcgid](#)

Installing CloudLinux Enhancement

There is no need to install mod_hostinglimits -- it comes built in with H-Sphere. Once you load kernel from CloudLinux with liblve 0.8 or later -- it will get enabled.

You can check if LVE is enabled by running:

```
$ ps aux | grep httpd | grep DLIBLVE
```

If you see no output, it means that Apache didn't pick up LVE. Try checking file:

```
/hsphere/shared/scripts/apache-get-env.sh
```

Following lines should be there

```
if [ -e /usr/lib64/liblve.so.0 -o -e /usr/lib/liblve.so.0 ]; then
    APENV_DSSL="$APENV_DSSL -DLIBLVE"
fi
```

If those strings are absent, you should add it, after:

```
else
    APENV_DSSL='-DSSL'
fi
###
```

and before:

```
# this is used by apacheGetEnv.pm perl module
if [ "$1" = 'show' ] ; then
    set | egrep "^APENV_"
fi
```

strings. Restart apache afterward.

* don't forget to [convert from mod_fastcgi to mod_fcgid](#)

1.8.1 Converting from mod_fastcgi to mod_fcgid

To achieve the best results in productivity and stability we recommend converting from mod_fastcgi to mod_fcgid .

H-Sphere 3.6.3 and newer

Step 1:

Download our fcgi.conf file:

```
$ wget -O /hsphere/local/config/httpd2/fcgi.conf http://repo.cloudlinux.com/cloudlinux/sources/mo
```

Step 2:

Edit ~httpd2/conf/extra/httpd-hostinglimits.conf to the following state:

```
#####
LoadModule hostinglimits_module /hsphere/shared/apache2/modules/mod_hostinglimits.so

<IfModule mod_hostinglimits.c>
SkipErrors Off
AllowedHandlers cgi-script %php% fcgid-script application/x-miva-compiled
DenyHandlers hs-php5-script hs-php53-script hs-php54-script
Include /hsphere/local/config/httpd2/fcgi.conf

</IfModule>
#####
```

Step 4:

Go to P.Servers > web server [Config] and be sure to have enable:

- apache_version=2
- apache_mpm=prefork
- apache_fastcgi
- apache_fcgid
- PHP version/mode: php_fastcgi*

** No changes needed to httpd.conf.tpl.custom or usermodule.phpmode as this version provides its own mod_fcgid.*

Older Versions of H-Sphere

Step 1:

Compile mod_fcgid module

```
$ yum install gcc liblve-devel zlib-devel openssl-devel
$ wget http://apache.osuosl.org/httpd/mod_fcgid/mod_fcgid-2.3.9.tar.gz
$ tar zxvf mod_fcgid-2.3.9.tar.gz
$ cd mod_fcgid-2.3.9/
$ APXS=/hsphere/shared/apache2/bin/apxs ./configure.apxs
$ make
$ mv modules/fcgid/.libs/mod_fcgid.so /hsphere/shared/apache2/modules
```

Step 2:

Download and apply patch http://repo.cloudlinux.com/cloudlinux/sources/mod_fcgid-hsphere/usermodule.phpmode.patch to /hsphere/local/config/scripts/usermodule.phpmode

```
$ wget http://repo.cloudlinux.com/cloudlinux/sources/mod_fcgid-hsphere/usermodule.phpmode.patch
$ patch /hsphere/local/config/scripts/usermodule.phpmode usermodule.phpmode.patch
```

Step 3:

If /hsphere/local/config/httpd2/httpd.conf.tpl.custom do not exists - create it:

```
$ cp -rp /hsphere/local/config/httpd2/httpd.conf.tpl /hsphere/local/config/httpd2/httpd.conf.tpl
download and apply patch http://repo.cloudlinux.com/cloudlinux/sources/mod\_fcgid-hsphere/httpd.conf.tpl.patch to /hsphere/local/config/httpd2/httpd.conf.tpl.custom
```

```
$ wget http://repo.cloudlinux.com/cloudlinux/sources/mod\_fcgid-hsphere/httpd.conf.tpl.patch
$ patch --fuzz=3 /hsphere/local/config/httpd2/httpd.conf.tpl.cusom httpd.conf.tpl.patch
```

Step 4:

download pre-defined config file http://repo.cloudlinux.com/cloudlinux/sources/mod_fcgid-hsphere/fcgi.conf to /hsphere/local/config/httpd2

```
$ wget -O /hsphere/local/config/httpd2/fcgi.conf http://repo.cloudlinux.com/cloudlinux/sources/mod\_fcgid-hsphere/fcgi.conf
```

Step 5:

download our wrapper file http://repo.cloudlinux.com/cloudlinux/sources/mod_fcgid-hsphere/php-wrapper into /hsphere/shared/php5/bin/ and make it executable

```
$ wget -O /hsphere/shared/php5/bin/php-wrapper http://repo.cloudlinux.com/cloudlinux/sources/mod\_fcgid-hsphere/php-wrapper
$ chmod 755 /hsphere/shared/php5/bin/php-wrapper
```

Step 6:

change permissions for /hsphere/local/home to 755

```
$ chmod 755 /hsphere/local/home
```

Step 7:

Edit ~httpd2/conf/extra/httpd-hostinglimits.conf and add DenyHandlers, so section will looks like:

```
<IfModule mod_hostinglimits.c>
SkipErrors Off
AllowedHandlers cgi-script %php% fcgid-script application/x-miva-compiled
DenyHandlers hs-php5-script hs-php53-script hs-php54-script
</IfModule>
```

Step 8:

configure physical server from H-Sphere admin > E.Manager > P.Servers > server_name [parameters] icon, settings should be:

```
apache_version = 2
apacha_fastcgi = yes
apache_status = yes
```

common	
apache_version ?	2 ▾
apache_mpm ?	prefork ▾
web servers	
Apache Modules	
apache_ssl ?	<input checked="" type="checkbox"/> (* Apache module mod_ssl)
apache_fastcgi ?	<input checked="" type="checkbox"/> (* Apache module mod_fastcgi)
apache_scgi ?	<input type="checkbox"/> (* Apache module mod_scgi)
apache_throttle ?	<input type="checkbox"/> (* Apache module mod_throttle)
apache_frontpage ?	<input type="checkbox"/> (* Apache module mod_frontpage)
apache_status ?	<input checked="" type="checkbox"/> (* Apache module mod_status)

Step 9:

Set PHP configuration to:

```
php_libphp5 enabled but not default
php_fastcgi5 enabled and is default
```

PHP Configuration ?			
Version	Mode	Enabled	Default
4	php_libphp4 ?	<input type="checkbox"/>	<input type="radio"/>
	php_fastcgi4 ?	<input type="checkbox"/>	<input type="radio"/>
	php_cgi4 ?	<input type="checkbox"/>	<input type="radio"/>
5	php_libphp5 ?	<input checked="" type="checkbox"/>	<input type="radio"/>
	php_fastcgi5 ?	<input checked="" type="checkbox"/>	<input checked="" type="radio"/>
	php_cgi5 ?	<input type="checkbox"/>	<input type="radio"/>

Other options could be configured according to personal needs.
When done - click SUBMIT to apply changes.

Note: After updating H-Sphere software on web server with CloudLinux you need to re-apply step 2 (patch usemodule.phpmode) and restart apache with /hsphere/shared/scripts/apache-restart script

1.9 Virtuozzo and OpenVZ

[beta]

* *Kernel 2.6.32-042stab088.4 or later required*

CloudLinux provides limited support for OpenVZ and Virtuozzo. At this stage only following functionality works:

CageFS

PHP Selector

max entry processes

No other limits work at this stage.

Installation

VZ Node (needs to be done once for the server):

*** Make sure all containers are stopped prior to doing this operation. Or reboot the server after the install.**

```
$ wget -P /etc/yum.repos.d/ http://repo.cloudlinux.com/vzlive/vzlive.repo
$ yum install lve-kernel-module
```

This will setup LVE module for VZ kernel, as well as DKMS to update that module each time VZ kernel is updated.

After this is done, you can add LVE support for any container on a node, at any time.

For CloudLinux to work inside VZ container, VZ node has to be enabled. This should be done for any container where LVE support needs to be added.

```
$ vzctl set CT_ID --devnodes lve:rw --save
```

To disable LVE support for Container:

```
$ vzctl set CT_ID --devnodes lve:none --save
```

Inside container, follow standard CL installation procedures, with one difference:

You will be downloading cldeploy from different place, as it is a different version with VZ support. In the future we will have one version of cldeploy for all cases.

```
$ wget http://repo.cloudlinux.com/vzlive/cldeploy
$ sh cldeploy -i
# or
$ sh cldeploy -k KEY
```

Follow up with (see CageFS & PHP Selector instructions for more info):

```
$ cagefsctl --init
$ yum groupinstall alt-php # (if desigred)
```

CloudLinux license is required for each VZ container.

1.10 Getting Trial License

You will need a trial activation key to be able to convert your CentOS server to CloudLinux. The trial subscription will work for 30 days.

If you have any issues getting activation key or if you have any questions regarding using your trial subscription -- contact sales@cloudlinux.com and we will help.

To get the activation key:

1. Register with CloudLinux Network: <https://cln.cloudlinux.com/clweb/register.html> (skip it if you already registered)

2. You will receive an email with activation link.
3. Login at: <https://cln.cloudlinux.com/clweb/login.html>
4. Click on Get Trial Activation Key

You should get a key that looks like: 12314-d34463a182fede4f4d7e140f1841bcf2

Use it to register your system or to [convert CentOS server to CloudLinux](#) server.

1.11 Registering CloudLinux Server

To register your server with CloudLinux Network using activation key:

```
$ yum install rhn-setup --enablerepo=cloudlinux-base  
$ /usr/sbin/rhnreg_ks --activationkey=<activation key> --force
```

Where activation key is like 1231-2b48feedf5b5a0e0609ae028d9275c93

If you have IP based license, use `clnreg_ks` command:

```
$ yum install rhn-setup --enablerepo=cloudlinux-base  
$ /usr/sbin/clnreg_ks --force
```

1.12 CloudLinux on DigitalOcean

How to make CloudLinux work on DigitalOcean:

DigitalOcean doesn't support custom kernels. The droplet (VM) always runs DigitalOcean's kernel. CloudLinux requires its own kernel. To enable CloudLinux work on DigitalOcean droplets, we provide ability to boot into CloudLinux kernel using `kexec` functionality.

How does this work:

- `cldeploy` script checks for presence of `/etc/digitalocean`. If the file detected, we assume that this is DigitalOcean droplet;
- `kexec-tools` are installed;
- `kexec` script will be created in `/etc/rc.d/init.d/` and set to run right after `rc.sysinit`.

When executed, script `/etc/rc.d/init.d/kexec` detects latest installed CloudLinux kernel, and loads that kernel.

If the system cannot boot into CloudLinux kernel (due to any reason), subsequent reboot will skip `kexec`, allow droplet to boot into DigitalOceans' kernel.

To disable booting into Cloudlinux kernel, run: `chkconfig --del kexec`

To re-enable booting into CloudLinux kernel, run: `chkconfig --add kexec`

1.13 Servers with LILO boot loader

CloudLinux can be deployed on servers that don't have `grub` installed, by installing `grub` first.

To do that:

1. Make sure `grub` and kernel packages are not excluded. Edit file `/etc/yum.conf` and check `exclude=`

line for presence of kernel* grub*

2. Backup lilo config file:

```
mv /etc/lilo.conf /etc/lilo.conf.bak
```

3. Convert to CloudLinux using [deploy2cl](#) utility

4. Check `grub.conf` -- it should be configured automatically:

```
# cat /boot/grub/grub.conf
default=0
timeout=5
    title CloudLinux Server (2.6.18-294.8.1.el5.lve0.7.33)
        kernel /boot/vmlinuz-2.6.18-294.8.1.el5.lve0.7.33 root=/dev/sda1 ro
        root (hd0,0)
        initrd /boot/initrd-2.6.18-294.8.1.el5.lve0.7.33.img
    title linux centos5_64
        kernel /boot/bzImage-2.6.33.5-xxxx-grs-ipv4-64 root=/dev/sda1 ro
        root (hd0,0)
```

5. Install grub to master boot record:

```
/sbin/grub-install /dev/sda
```

6. Reboot and check that you are running CloudLinux. `uname -r` should show something like: 2.6.18-294.8.1.el5.lve0.7.33

1.14 Uninstalling CloudLinux

You can always uninstall CloudLinux. In this case, we will 'convert' the system back to CentOS. Even if the original system was RHEL -- we will still convert to 'CentOS' state.

Following this will be done:

1. LVE related packages will be removed
2. CloudLinux repositories & yum plugin will be removed
3. CentOS repositories will be setup

At the end, script will provide instructions on how to finish the conversion back to CentOS. That will require removal of CloudLinux kernel (manual step), and installation of CentOS kernel (if needed).

To uninstall CloudLinux, do:

```
$ wget -O cldeploy http://repo.cloudlinux.com/cloudlinux/sources/cln/cldeploy
$ sh cldeploy -c
```

Please, note that some of the packages from CloudLinux repo will still be present. They are same as CentOS packages, and don't have to be removed. They will be updated in the future from CentOS repositories, as new versions come out.

2 Limits

CloudLinux has support for following limits:

Limits	Units	Default Value	Description	Supported Kernels / OS
SPEED	% of a core, or HZ	100%	CPU speed limit, relative to a single core, or specified in HZ (portable across CPUs)	all
CPU [deprecated]	% of CPU	25%	CPU Limit (smallest of CPU & NCPU is used)	all
NCPU [deprecated]	number of cores	1 CORE	Max number of cores (smallest of all CPU & NCPU used)	
PMEM	KB	1024MB	Physical memory limit (RSS field in ps/RES in top). Also includes shared memory and disk cache	CL5 hybrid kernel, CL5 lve1.x+ kernel & CL6
VMEM	KB	1024MB	Virtual memory limit (VSZ field in ps/VIRT in top)	all
IO	KB/sec	1024KB/sec	IO throughput - combines both read & write operations	CL6 lve1.1.9+ kernel, CL5 hybrid kernel
IOPS [requires lve1.3+]	Operations per second	0	Restricts total number of read/write operations per second.	CL6 and CL5 hybrid kernels lve1.3+
NPROC	number	100	Max number of processes within LVE	CL5 hybrid kernel, CL5 lve1.x+ kernel & CL6
EP	number	20	Limit on entry processes. Usually all represents max number of concurrent connections to apache dynamic scripts as well as SSH and cron jobs running simultaneously.	

Bellow you can find recommendations for your typical shared hosting setup. The recommendations don't depend on the power of your server. They only depend on how "fast" you want your hosting accounts to be.

Typical Hosting Account

SPEED=100%

```
PMEM=256MB
VMEM=0
IO=1024KB/s
NPROC=100
EP=20
```

High End Hosting Account

```
SPEED=200%
PMEM=512MB
VMEM=0
IO=1024KB/s
NPROC=100
EP=40
```

2.1 Understanding LVE

LVE is a kernel level technology developed by the CloudLinux team. The technology has common roots with container based virtualization and uses cgroups in its latest incarnation. It is lightweight, and transparent. The goal of LVE is to make sure that no single web site can bring down your web server. Today, a single site can consume all CPU, IO, Memory resources or apache processes -- and bring the server to a halt. LVE prevents that. It is done via collaboration of apache module, PAM module and kernel.

[mod_hostinglimits](#) is apache module that:

- Detects VirtualHost from which the request came.
- Detects if it was meant for cgi or PHP script.
- Puts apache process used to serve that request into LVE for the user determined via SuexecUserGroup directive for that virtual host.
- Lets apache to serve the request.
- Removes apache process from user's LVE.

The kernel makes sure that all LVEs get fair share of the server's resources, and that no customer can use more then the limits set for that customer.

Today we can limit CPU, Memory (virtual and physical), IO, number of processes as well as number of entry processes (concurrent connections to apache).

Each LVE limits amount of entry processes (Apache processes entering into LVE) to prevent single site exhausting all Apache processes. If the limit is reached -- mod_hostinglimits will not be able to place Apache process into LVE, and will return error code 508. This way very heavy site would slow down and start returning 508 errors, without affecting other users.

If the site is limited by CPU or IO -- the site will start responding slower.

If the site is limited by memory or number of processes limits -- user will be 500 or 503 errors that server cannot execute the script

Checking if LVE is installed

To use LVE you have to have CloudLinux kernel installed, and LVE module loaded you can check the kernel by running the following command:

```
$ uname -r
```

You should see something like 2.6.18-294.8.1.el5.lve0.8.60. The kernel should have **lve** in its name. To see if lve kernel module is loaded do

```
$ lsmod|grep lve
lve                46496  0
```

To see if iolimits module is enabled (**e16.lve1.1.9** kernels and later):

```
$ lsmod|grep iolimits
iolimits
```

You can toggle LVE on/off by editing `/etc/sysconfig/lve` and setting `LVE_ENABLE` variable to yes or no. Setting it to yes will enable LVE, setting it to no will disable LVE.

You can toggle IO limits by editing `/etc/sysconfig/iolimits` and setting `IO_LIMITS_ENABLED` variable to yes or no. You need to reboot the server, after you set this option to make the changes live

Controlling LVE Limits

The best way to control LVE limits is using LVE Manager in your favorite control panel. Alternatively, you can use command line tool [lvectl](#) to control limits.

The limits are saved in `/etc/container/ve.cfg`

Example:

```
<?xml version="1.0" ?>
<lveconfig>
  <defaults>
    <cpu limit="25"/>
    <ncpu limit="1"/>
    <io limit="1024"/>
    <mem limit="262144"/>
    <other maxentryprocs="200"/>
    <pmem limit="262144"/>
    <nproc limit="0"/>
  </defaults>
  <lve id="532">
    <cpu limit="30"/>
    <ncpu limit="5"/>
  </lve>
</lveconfig>
```

Sets CPU limit to 25%, IO limit to 1024KB/s, virtual memory limit to 1GB (memory limit is set as a number of 4096 bytes pages), physical memory limit to 1GB, CPU cores per LVE to 1, maximum entry processes to 200 and no limit for number of processes for all LVEs. It also sets the limit of 30% and number of processes limit to 5 for LVE with ID 532

Checking LVE Usage

One of the best way to monitor current usage is [lvetop](#)

```
$ lvetop
  ID      EP      PNO      TNO      CPU      MEM      I/O
  test    1        2        2        2%      728      0
```

You can also check the content of `/proc/lve/list` file that has all the data about LVE usage for all LVEs

```
[root@localhost tests]$ cat /proc/lve/list
4:LVE EP lCPU lIO CPU MEM IO lMEM lEP nCPU fMEM fEP
0 0 75 25 0 0 0 262144 20 2 0 0
500 0 75 25 0 0 0 4294967 20 3 2 1
700 1 75 25 1403247 202 0 262144 20 2 0 0
```

Additionally you can use tool `lveps` to see CPU usage, and processes within LVE

2.2 Command Line Tools

2.2.1 lvectl

`lvectl` is the primary tool for LVE management. To use it, you have to be administrator. `lvectl` is part of `lve-utils` package. The syntax of `lvectl` is:

Usage: `lvectl` command [veid] [options]

commands:

```
apply          apply config settings to specified LVE
apply all     apply config settings to all the LVEs
apply-many    to apply LVE limits to multiple distinct LVEs (uids of users are
read from stdin)
set          set parameters for a LVE and/or create a LVE
set-user     same as set, but get LVE id as user id from user name
list        list loaded LVEs
list-user    same as list, but show username instead of LVE id whenever possible
limits      show limits for loaded LVEs
delete       delete LVE and set configuration for that LVE to defaults
delete-user  same as delete, but retrieve uid of user from username
destroy     destroy LVE (configuration file remains unchanged)
destroy all  destroy all LVE (configuration file remains unchanged)
destroy-many to destroy LVE limits to multiple distinct LVEs (uids of users are
read from stdin)
package-set  set LVE parameters for a package
package-list list LVE parameters for packages
package-delete delete LVE parameters for a package
paneluserslimits show current user's limits for control panel
help (-h)   show this message
version (-v) version number
lve-version lve version number
```

options:

```
--speed=XX%      limit CPU usage relative to single core. As such 50% would
mean 1/2 core, 100% would mean 1 core, and 200 would mean 2 cores
--speed=XXmhz/ghz limit CPU usage relative to processor speed in mhz or ghz.
--cpu=N          limit CPU usage (deprecated since lve-utils 1.4)
--ncpu=N        limit VCPU usage (deprecated since lve-utils 1.4)
--io=N          define io limit, in KB/s
--nproc=N       limit number of processes
--pmem=N        limit physical memory usage for applications inside LVE
--vmem=N        limit virtual memory for applications inside LVE
--maxEntryProcs=N limit number of entry processes
--save-all-parameters save all parameters, even if they match default settings
--json          return results in json format [lve-utils 1.2-10+]
--unlimited      removes limits for cpu, ncpu, io, nproc, pmem, vmem and
entry processes
```

[lve-utils 1.3-15]

Examples

Reset all LVEs settings based on configuration in `/etc/container/ve.cfg`

```
$ lvectl apply all
```

Set new default CPU & Physical memory limit

```
$ lvectl set default --speed=100% --pmem=256m
```

Reset all LVE's killing processes inside them.

```
$ lvectl destroy all
```

Show list of LVEs and their limits:

```
$ lvectl list
```

2.2.2 lveps

`lveps` tool show information about running LVEs.

Usage: `lveps [-p] [-n] [-o <fmt1:width1,...>] [-d] [-c <time>] [-s <style>] [-t] [-h]`

Options

- p to print per-process/per-thread statistics*
- n to print LVE ID instead of username*
- o to use formatted output (fmt=id,ep,pid,tid,cpu,mem,io)*
- d to show dynamic cpu usage instead of total cpu usage*
- c to calculate average cpu usage for <time> seconds (used with -d)*
- r to run under realtime priority for more accuracy (needs privileges)*
- s to sort LVEs in output (cpu, process, thread, mem, io)*
- t to run in the top-mode*
- h to print this brief help message*

Command like `lveps -p` will display processes running inside 'active' LVEs.

2.2.3 lvetop

`lvetop` utility allows to monitor LVE usage

```
$ lvetop
ID      EP      PNO      TNO      CPU      MEM      I/O
90      1       1        1        6%      13K      0
user1   17      18       18       2%      56M      0
user2   8       8        16       18%     72M      0
```

lvetop fields

ID user name if LVE id matches user id in `/etc/passwd`, or LVE id
 EP number of entry processes (concurrent scripts executed)
 PNO number of processes within LVE
 TNO number of threads within LVE
 CPU CPU usage by LVE, relative to total CPU resources of the server
 MEM Memory usage by LVE, in KB
 I/O I/O usage (currently not implemented)

2.2.4 cldetect

[lve-utils 1.2-10+]

`cldetect` is used to detect installed software, and adjust CloudLinux options accordingly

Usage: `/usr/bin/cldetect [--options]`

<code>-h --help</code>	show this message
<code>--detect-cp</code>	prints control panel and its version (CP_NAME,CP_VERSION)
<code>--detect-cp-nameonly</code>	prints control panel name (CP_NAME)
<code>--cxs-installed</code>	check if CXS is installed. Returns 0 if installed, 1 otherwise
<code>--cpanel-suphp-enabled</code>	check if suPHP is enabled in cPanel. Returns 0 if enabled, 1 otherwise
<code>--detect-litespeed</code>	check if LiteSpeed is installed. Returns 0 if installed, 1 otherwise
<code>--detect-postgresql</code>	check if PostgreSQL is installed. Returns 0 if installed, 1 otherwise
<code>--print-apache-gid</code>	prints current apache gid
<code>--print-da-admin</code>	prints DirectAdmin admin user
<code>--set-securelinks-gid</code>	changes <code>/etc/sysctl.conf</code> if apache gid != 48 (default)

2.2.5 lve-stats

Package `lve-stats` collects LVE usage statistics, and allows to query the data.

To install, run:

```
$ yum install lve-stats
```

If you are already running `lve-stats` (in case you are running cPanel LVE plugin), do:

```
$ yum update lve-stats
```

This should also be updated automatically next time your system runs system wide update.

The package installs `lvestats-server`. You can re-start the server by running

```
$ service lvestats restart
```

The package creates sqlite database `/var/lve/lveinfo.db` that holds historical information about LVE usage. Up to two months of hourly info is stored for each client. The data for the last hour is stored with 5 minutes interval, and the data for the past 10 minutes is stored with 1 minute interval.

LVE Stats updates `/var/lve/info` every few seconds. That info is used by LVE Manager plugin

Package consists of `lveinfo` utility to query LVE usage, and `lvechart` that allows you to chart usage for individual LVE.

To query historical LVE info, `lveinfo` command provided. It is located at `/usr/sbin/lveinfo`


```

# /usr/sbin/lveinfo [OPTIONS]
-h --help           : this help screen
-v, --version       : version number
-d, --display-username : try to convert LVE id into username when possible
-f, --from=         : run report from date and time in YYYY-MM-DD HH:MM format
                    : if not present last 10 minutes are assumed
-t, --to=          : run report up to date and time in YYYY-MM-DD HH:MM format
                    : if not present, reports results up to now
-o, --order-by=    : orders results by one of the following:
  cpu_avg          : average CPU usage
  cpu_max          : max CPU usage
  mep_avg          : average number of entry processes (concurrent connections)
  mep_max          : max number of entry processes (concurrent connections)
  vmem_avg         : average virtual memory usage
  vmem_max         : max virtual memory usage
  pmem_avg         : average physical memory usage
  pmem_max         : max physical memory usage
  nproc_avg        : average number of processes usage
  nproc_max        : max number of processes usage
  io_avg           : average IO usage
  io_max           : max IO usage
  total_mem_faults : total number of out of virtual memory faults (deprecated
since 0.8-6)
  total_vmem_faults: total number of out of virtual memory faults (since 0.8-6)
  total_pmem_faults: total number of out of physical memory faults (since 0.8-6)
  total_mep_faults : total number of entry processes faults (deprecated since 0.8-
6)
  total_ep_faults  : total number of entry processes faults (since 0.8-6)
  total_nproc_faults: total number of number of processes faults (since 0.8-6)
  any_faults       : total number of any types of faults (since 0.8-6)
  --id=            : LVE id -- will display record only for that LVE id
-u, --user=        : Use username instead of LVE id, and show only record for that
user
-l, --limit=       : max number of results to display, 10 by default
-c, --csv          : display output in CSV format
-b, --by-usage     : show LVEs with usage (averaged or max) within 90% percent of
the limit
  available values:
  cpu_avg          : average CPU usage
  cpu_max          : max CPU usage
  mep_avg          : average number of entry processes (concurrent connections)
  ep_avg           : average number of entry processes (since 0.8-6)
  mep_max          : max number of entry processes (concurrent connections)
  ep_max           : max number of entry processes (since 0.8-6)
  mem_avg          : average virtual memory usage
  mem_max          : max virtual memory usage
  vmem_avg         : average virtual memory usage
  vmem_max         : max virtual memory usage
  pmem_avg         : average physical memory usage
  pmem_max         : max physical memory usage
  nproc_avg        : average number of processes
  nproc_max        : max number of processes
  io_avg           : average IO usage
  io_max           : max IO usage
-p, --percentage   : defines percentage for --by-usage option
-f, --by-fault     : show LVEs which failed on max entry processes limit or memory
limit
  available values: mem, mep.
  since 0.8-6      : vmem, pmem, ep, nproc
--show-all        : since 0.8-6 only columns for enabled limits will show up.
-r, --threshold    : in combination with --by-fault, shows only LVEs with number

```

```
of faults above threshold specified
--server_id      : used in combination with centralized storage, to access info
from any server
--show-all      : full output (show all limits); brief output by default
```

Output

```
ID          LVE Id or username
aCPU        Average CPU usage
mCPU        Max CPU usage
lCPU        CPU Limit
aEP         CPU Limit
mEP         Max Entry Processes
lEP         Entry Proc limit
aNPROC      Average Number of Processes
mNPROC      Max Number of Processes
lNPROC      Number of Processes limit
aVMEM       Average virtual Memory Usage
mVMEM       Max virtual Memory Usage
lVMEM       Virtual Memory Limit
aPMEM       Average physical Memory Usage
mPMEM       Max physical Memory Usage
lPMEM       Physical Memory Limit
aIO         Average IO usage
mIO         Max IO usage
lIO         IO Limit
fVMEM       Out Of Virtual Memory Faults
fPMEM       Out Of Physical Memory Faults
fEP         Entry processes faults
fNPROC      Number of processes faults
```

** only enabled limits will show up*

Examples

Display top 10 users, by max CPU usage, from Oct 10, 2010 to Oct 15, 2010. Display username if possible

```
$ lveinfo --from='2010-10-10' --to='2010-10-15' -o cpu_max --display-username
ID   aCPU  mCPU  lCPU  aEP  mEP  lEP  aMem  mMem  lMem  MemF  MepF
777   7     9     10    0    0    25   10M   15M   1G    0     0
300   2     8     10    0    1    25   1M    3M    1G    0     0
web2  1     6     10    0    0    25   17K   18M   1G    0     0
web1  0     0     10    0    0    25   204K  1M    1G    0     0
```

Display LVE info about user web2, from Oct 10, 2010 to Oct 15, 2010.

```
$ lveinfo --from='2010-10-10' --to='2010-10-15' --user=web2 --display-username
ID   aCPU  mCPU  lCPU  aEP  mEP  lEP  aMem  mMem  lMem  MemF  MepF
web2  1     6     10    0    0    25   10M   15M   1G    0     0
```

2.2.5.1 Storing statistics in MySQL

You have to install MySQL-python rpm to store lve-stats on centralized server

```
$ yum install MySQL-python
```

If you have MySQL 5.3+ installed on CloudLinux 5 server, and there is no libmysqlclient_r.so.15 on the server, do:

```
$ yum --enablerepo=cloudlinux-updates-testing install mysqlclient15
```

A typical procedure to configure the MySQL database for storing information about multiple servers for lve-stats services looks as follows:

Create of database and user. You can do it by executing following commands:

```
create database <database>;  
grant all on <database>.* to <user> identified by 'password';  
flush privileges;
```

Create database schema:

```

CREATE TABLE history (id INTEGER,
    cpu INTEGER, cpu_limit INTEGER,
    cpu_max INTEGER,
    ncpu INTEGER,
    mep INTEGER, mep_limit INTEGER,
    mep_max INTEGER,
    io INTEGER, io_limit INTEGER,
    mem INTEGER, mem_limit INTEGER,
    mem_max INTEGER,
    mem_fault INTEGER, mep_fault INTEGER,
    created TIMESTAMP, weight INTEGER, server_id CHAR(10),
    lmemphy INTEGER, memphy INTEGER, memphy_max INTEGER, memphy_fault INTEGER,
    lnproc INTEGER, nproc INTEGER, nproc_max INTEGER, nproc_fault INTEGER,
    lcpuw INTEGER, io_max INTEGER,
    iops INTEGER, liops INTEGER, iops_max INTEGER );
CREATE INDEX idx_history_id ON history(id);
CREATE INDEX idx_history_created ON history(created);
CREATE INDEX idx_history_weight ON history(weight);
CREATE INDEX idx_history_server_id ON history(server_id);
CREATE TABLE last_run (hourly TIMESTAMP, daily TIMESTAMP, server_id CHAR(10),
lve_version INTEGER);
CREATE TABLE users (server_id CHAR(10), id INTEGER, username CHAR(20));
CREATE INDEX idx_users_server_id ON users(server_id);
CREATE INDEX idx_users_id ON users(id);

CREATE TABLE history_gov ( ts INTEGER,
    username CHAR(64),
    max_simultaneous_requests INTEGER,
    sum_cpu FLOAT,
    sum_write FLOAT,
    sum_read FLOAT,
    number_of_iterations INTEGER,
    max_cpu FLOAT,
    max_write FLOAT,
    max_read FLOAT,
    number_of_restricts INTEGER,
    limit_cpu_on_period_end INTEGER,
    limit_read_on_period_end INTEGER,
    limit_write_on_period_end INTEGER,
    cause_of_restrict INTEGER,
    weight INTEGER,
    server_id char(10));

CREATE INDEX idx_history_gov_ts ON history_gov(ts);
CREATE INDEX idx_history_gov_cause_of_restrict ON history_gov(cause_of_restrict);
CREATE INDEX idx_history_gov_number_of_restricts ON history_gov(number_of_restricts);
CREATE INDEX idx_history_gov_max_simultaneous_requests ON
history_gov(max_simultaneous_requests);
CREATE INDEX idx_history_gov_server_id ON history_gov(server_id);
CREATE INDEX idx_history_gov_weight ON history_gov(weight);

CREATE TABLE last_run_gov (hourly TIMESTAMP, daily TIMESTAMP, server_id CHAR(10),
lve_version INTEGER);

* Execute following SQL command for each remote server for which you want to store
statistics in this database (make sure you substitute _SERVER_NAME_ with the same
servername as used in lvestats config file on remote server:

INSERT INTO last_run(hourly, daily, server_id, lve_version) VALUES (UTC_TIMESTAMP(),

```

```
UTC_TIMESTAMP(), '_SERVER_NAME_', 4);
```

On each server edit file `/etc/sysconfig/lvestats` & `/etc/sysconfig/lvestats.readonly` as follows:

```
db_type = mysql
connect_string = host:database:user:password
server_id = _SERVER_NAME_
db_port = _port
```

* Note: `lvestats.readonly` should have a user that has read only access to all tables from `lvestats` database.

* Note: `_SERVER_NAME_` should be at most 10 characters

* Note: `db_port` is an optional parameter. Default port would be used.

Select server responsible for compacting database on regular bases by setting **COMPACT=master** in `/etc/sysconfig/lvestats` for that server. Set **COMPACT=slave** on all other servers.

Make sure that `/etc/sysconfig/lvestats` is readable only by root (`chmod 600 /etc/sysconfig/lvestats`), `lvestats.readonly` should be readable by anyone

restart service:

```
service lvestats restart
```

If you use central database to store `lvestats` data, on each server, execute:

```
$ /usr/share/lve-stats/save_users_to_database.py
```

You just need to execute it once, as it will be later executed via cron job. That script will store usernames from each server, so that `lve-stats` would later be able to correctly identify each user.

Updating MySQL & PostgreSQL schema for lve-stats 0.8+

If you are using MySQL or PostgreSQL server for `lve-stats` older then 0.8, make sure to do following steps to upgrade to latest version:

Stop `lvestats` service on all your servers

Connect to your database server, and execute following commands:

```
ALTER TABLE history ADD lmemphy INTEGER;
ALTER TABLE history ADD memphy INTEGER;
ALTER TABLE history ADD memphy_max INTEGER;
ALTER TABLE history ADD memphy_fault INTEGER;
ALTER TABLE history ADD lnpoc INTEGER;
ALTER TABLE history ADD nproc INTEGER;
ALTER TABLE history ADD nproc_max INTEGER;
ALTER TABLE history ADD nproc_fault INTEGER;
ALTER TABLE history ADD lcpuw INTEGER;
ALTER TABLE history ADD io_max INTEGER;
UPDATE history SET lmemphy = 0, memphy = 0, memphy_max = 0, memphy_fault = 0,
    lnpoc = 0, nproc = 0, nproc_max = 0, nproc_fault = 0,
    lcpuw = 0, io_max = 0;

ALTER TABLE last_run ADD lve_version INTEGER;
UPDATE last_run SET lve_version = 4;
CREATE TABLE last_run_gov (hourly TIMESTAMP, daily TIMESTAMP, server_id CHAR(10), lve_version
INTEGER);
```

To upgrade scheme to support MySQL Governor:

```
CREATE TABLE history_gov ( ts INTEGER,
  username char(64),
  max_simultaneous_requests INTEGER,
  sum_cpu float,
  sum_write float,
  sum_read float,
  number_of_iterations INTEGER,
  max_cpu float,
  max_write float,
  max_read float,
  number_of_restricts INTEGER,
  limit_cpu_on_period_end INTEGER,
  limit_read_on_period_end INTEGER,
  limit_write_on_period_end INTEGER,
  cause_of_restrict INTEGER,
  server_id char(10));

CREATE INDEX idx_history_gov_ts ON history_gov(ts);
CREATE INDEX idx_history_gov_cause_of_restrict ON history_gov(cause_of_restrict);
CREATE INDEX idx_history_gov_number_of_restricts ON history_gov(number_of_restricts);
CREATE INDEX idx_history_gov_max_simultaneous_requests ON
history_gov(max_simultaneous_requests);
CREATE INDEX idx_history_gov_server_id ON history_gov(server_id);
```

Upgrading from lve-stats < 0.9-20

```
ALTER TABLE history_gov ADD weight INTEGER;
CREATE INDEX idx_history_gov_weight ON history_gov(weight);
CREATE TABLE last_run_gov (hourly TIMESTAMP, daily TIMESTAMP, server_id CHAR(10), lve version INTE
```

Update lve-stats RPM on all your servers

If you use central database to store lvestats data, execute following commands:

```
CREATE TABLE users (server_id CHAR(10), id INTEGER, username CHAR(20));
CREATE INDEX idx_users_server_id ON users(server_id);
CREATE INDEX idx_users_id ON users(id);
```

On each server, execute:

```
$ /usr/share/lve-stats/save_users_to_database.py
```

You just need to execute it once, as it will be later executed via cron job. That script will store usernames from each server, so that lve-stats would later be able to correctly identify each user.

2.2.5.2 Storing statistics in PostgreSQL

You have to install postgresql-python rpm to store lve-stats on centralized server

```
$ yum install postgresql-python
```

A typical procedure to configure the PostgreSQL database for storing information about multiple servers for lve-stats services looks as follows:

Create of database and user. You can do it by executing following commands:

```
createdb <database>  
createuser <user>
```

Create database schema:

```
CREATE TABLE history (id INTEGER,
    cpu INTEGER, cpu_limit INTEGER,
    cpu_max INTEGER,
    ncpu INTEGER,
    mep INTEGER, mep_limit INTEGER,
    mep_max INTEGER,
    io INTEGER, io_limit INTEGER,
    mem INTEGER, mem_limit INTEGER,
    mem_max INTEGER,
    mem_fault INTEGER, mep_fault INTEGER,
    created TIMESTAMP, weight INTEGER, server_id CHAR(10),
    lmemphy INTEGER, memphy INTEGER, memphy_max INTEGER, memphy_fault INTEGER,
    lnproc INTEGER, nproc INTEGER, nproc_max INTEGER, nproc_fault INTEGER,
    lcpuw INTEGER, io_max INTEGER);

CREATE INDEX idx_history_id ON history(id);
CREATE INDEX idx_history_created ON history(created);
CREATE INDEX idx_history_weight ON history(weight);
CREATE INDEX idx_history_server_id ON history(server_id);
CREATE TABLE last_run (hourly TIMESTAMP, daily TIMESTAMP, server_id CHAR(10),
    lve_version INTEGER);
CREATE TABLE users (server_id CHAR(10), id INTEGER, username CHAR(20));
CREATE INDEX idx_users_server_id ON users(server_id);
CREATE INDEX idx_users_id ON users(id);

CREATE TABLE history_gov ( ts INTEGER,
    username char(64),
    max_simultaneous_requests INTEGER,
    sum_cpu float,
    sum_write float,
    sum_read float,
    number_of_iterations INTEGER,
    max_cpu float,
    max_write float,
    max_read float,
    number_of_restricts INTEGER,
    limit_cpu_on_period_end INTEGER,
    limit_read_on_period_end INTEGER,
    limit_write_on_period_end INTEGER,
    cause_of_restrict INTEGER,
    weight INTEGER,
    server_id char(10));

CREATE INDEX idx_history_gov_ts ON history_gov(ts);
CREATE INDEX idx_history_gov_cause_of_restrict ON history_gov(cause_of_restrict);
CREATE INDEX idx_history_gov_number_of_restricts ON history_gov(number_of_restricts);
CREATE INDEX idx_history_gov_max_simultaneous_requests ON
    history_gov(max_simultaneous_requests);
CREATE INDEX idx_history_gov_server_id ON history_gov(server_id);
CREATE INDEX idx_history_gov_weight ON history_gov(weight);

CREATE TABLE last_run_gov (hourly TIMESTAMP, daily TIMESTAMP, server_id CHAR(10),
    lve_version INTEGER);

    * Execute following SQL command for each remote server for which you want to store
    statistics in this database (make sure you substitute _SERVER_NAME_ with the
same
    servername as used in lvestats config file on remote server:

INSERT INTO last_run(hourly, daily, server_id, lve_version) VALUES (now() AT TIME ZONE
'UTC', now() AT TIME ZONE 'UTC', '_SERVER_NAME_', 4);
```


On each server edit file `/etc/sysconfig/lvestats` and `/etc/sysconfig/lvestats` as follows:

```
db_type = postgresql
connect_string = host:database:user:password
server_id = _SERVER_NAME_
db_port = _port
```

* Note: `lvestats.readonly` should have a user that has read only access to history table.

* Note: `_SERVER_NAME_` should be at most 10 characters

* Note: `db_port` is optional, default PostgreSQL port will be used

Select server responsible for compacting database on regular bases by setting **COMPACT=master** in `/etc/sysconfig/lvestats` for that server. Set **COMPACT=slave** on all other servers.

Make sure that `/etc/sysconfig/lvestats` is readable only by root (`chmod 600 /etc/sysconfig/lvestats`), `lvestats.readonly` should be readable by anyone

restart service:

```
service lvestats restart
```

If you use central database to store `lvestats` data, on each server, execute:

```
$ /usr/share/lve-stats/save_users_to_database.py
```

You just need to execute it once, as it will be later executed via cron job. That script will store usernames from each server, so that `lve-stats` would later be able to correctly identify each user.

You are done!

2.2.5.3 Compacting in multi-server settings

[`lve-stats` 0.10+]

When you have multiple servers storing LVE statistics to a central database -- you will need to pick one server responsible for compacting data.

In that server, edit file: `/etc/sysconfig/lvestats`, and change option **COMPACT** to **master**

On all other servers, change that option to **slave**

Default: **single** -- should be used when `lve-stats` stores data to a single database

2.3 SPEED Limits

Requires: lve-utils 1.4+

CPU SPEED limit allows to set CPU limit in terms of % of a single core, or as a fixed number of Hz.

`--speed=XX%` would set performance relative to one core. For example:

`--speed=50%` would mean 1/2 core.

`--speed=100%` would mean 1 core,

`--speed=150%` would mean 1.5 cores

`--speed=XXmhz` would automatically detect CPU speed of each core, and adjust the CPU scheduler to make sure user cannot go over that limit.

For example on 1ghz CPU, setting of `--speed=2ghz` would mean 2 cores, while on 4ghz CPU same setting would mean 1/2 of a core.

This should allow hosting companies to set same approximate performance level limits across different hardware using single setting..

2.4 CPU Limits

[deprecated]

This limit is no longer used, and [SPEED](#) is used instead

CPU limits before lve-utils 1.4

CPU Limits are set by **CPU** and **NCPU** parameters. **CPU** specifies the % of total CPU of the server available to LVE. **NCPU** specifies the number of cores available to LVE. The smallest of the two is used to define how much CPU power will be accessible to the customer. For example:

1 core,

Cores Per Server	CPU Limit	NCPU Limit	Real limit
1	25%	1	25% of 1 core
2	25%	1	50% of 1 core
2	25%	2	50% of 1 core
4	25%	1	100% of 1 core (full core)
4	25%	2	1 core
4	50%	1	1 core
4	50%	2	2 cores
8	25%	1	1 core
8	25%	2	2 cores
8	50%	2	2 cores
8	50%	3	3 cores

When user hits CPU limit, processes within that limit are slowed down. For example, if you set your CPU limit to 10%, and processes inside LVE want to use more then 10% they will be throttled (put to sleep) to make sure they don't use more then 10%. In reality, processes don't get CPU time above the limit, and it happens much more often then 1 second interval, but the end result is that processes are slowed down so that their usage is never above the CPU limit set.

2.5 Memory Limits

Memory is controlled using virtual (VMEM) and physical (PMEM) memory limits.

Virtual Memory Limit

Virtual memory limit corresponds to the amount of memory processes can allocate within LVE. You can see individual process virtual memory usage by monitoring VIRT column in *top* output for the process. When process tries to allocate more memory, CloudLinux checks if the new total virtual memory used by all processes within LVE is more then a limit set. In such case CloudLinux will prevent memory from being allocated and increments fVMEM counter. In most cases, but not all of them - this causes process to fail. For CGI/PHP scripts it will usually cause 500 and 503 error.

Physical Memory Limit

Physical memory limit corresponds to the amount of memory actually used by end customer's processes. You can see individual process physical memory usage by monitoring RES column in *top* output for the process. Because similar processes (like PHP) share a lot of their memory, physical memory usage is often much lower than virtual memory usage.

Additionally physical memory includes shared memory used by the customer, as well as disk cache.

In case of disk cache – if user is starting to lack physical memory, the memory used for disk cache will be freed up, without causing any memory faults.

When LVE goes over physical memory limit, CloudLinux will first free up memory used for disk cache, and if that is not enough, it will kill some of the processes within that LVE, and increment `fpMEM` counter. This will usually cause web server to serve 500 and 503 errors. Physical memory limit is a much better way to limit memory for shared hosting.

2.6 IO Limits

IO limits restrict the data throughput for the customer. They are in KB/s. When limit is reached, the processes are throttled (put to sleep). This makes sure that processes within LVE cannot go over the limit. Yet don't stop working, nor getting killed – they just work slower when the limit is reached. IO limits are available with kernels `el6.lve1.x` and higher.

The IO limits will only affect DISK IO, and will have no effect on network. It also doesn't take into consideration any disk cache accesses. So, even if file is loaded from disk cache 1000 times – it will not be counted towards IO limits.

2.7 IOPS Limits

IOPS limits restrict the total number of read/write operations per second. When the limit is reached the read/write operations stop until current second expires.

2.8 Number of Processes Limit

`NPROC` controls the total number of processes within LVE. Once the limit is reached, no new process can be created (until another one dies). When that happens `fNPROC` counter is incremented. Apache might return 500 or 503 errors in such case.

URL of this page: http://docs.cloudlinux.com/index.html?number_of_processes_limit.html

2.9 Entry Processes Limit

Entry processes limit control the number of entries into LVE. Each time a process 'enters' into LVE, we increment the counter. Each time process exits LVE, we decrement the counter. We don't count processes that are created inside LVE itself. It is also known as 'Apache concurrent connections' limit.

The process enters into LVE when there is a new HTTP request for CGI/PHP, when new SSH session is created, or when new cron job is started.

This limit was created to prevent DoS attacks against web server. One of the fairly popular attacks is to tie up all the Apache connections by hitting some slow page on a server. Once all Apache slots are used up, no one else will be able to connect to the web server, causing it to appear to be down. The issue is worsened by CPU limits, as once site starts to get slow due to CPU limit – it will respond to requests slower and slower, causing more and more connections to be tied up.

To solve that, we have created entry processes (often called concurrent connections) limit. It will limit the number of concurrent connections to Apache, causing web server to serve error 508 page (Resource Limit Reached), once there number of concurrent requests for the site goes above the limit.

2.10 Compatibility Matrix

Web Server / PHP	CP U	Virtual & Physical Memory	EP	NPROC	IO	CageF S	PHP Selector
Apache / suPHP	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Apache / FCGID	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Apache / CGI	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Apache / PHP-FPM	Yes ³	Yes	Yes	Yes	Yes	Yes ³	No
Apache / mod_php	Yes	No	Yes	Yes	Yes	no	No
Apache / mod_ruid2	Yes	No	Yes	Yes	Yes	no	No
Apache / MPM ITK	Yes	No	Yes	Yes	Yes	Yes ¹	No
LiteSpeed	Yes	Yes ²	Yes	Yes	Yes	Yes	Yes
NGINX / PHP-FPM	Yes ³	Yes	Yes	Yes	Yes	Yes	No
SSH	Yes	Yes	Yes	Yes	Yes	Yes ³	Yes
Cron Jobs	Yes	Yes	Yes	Yes	Yes	Yes	Yes

1. Requires patched version of MPM-ITK. CL httpd RPM has ITK worker with the patch. Patch is also available at: <http://repo.cloudlinux.com/cloudlinux/sources/da/cl-apache-patches.tar.gz>

2. CloudLinux 6 or CloudLinux 5 hybrid kernels only

3. The DirectAdmin and CloudLinux PHP provide patched version. For other PHP distributions, please, use patches available here: <http://repo.cloudlinux.com/cloudlinux/sources/da/cl-apache-patches.tar.gz>

2.11 Integration Components

CloudLinux uses various ways to integrate with existing system. By default we can integrate with:

- PAM - using pam_lve

- Apache - using mod_hostinglimits, apr library, patched suexec
- LiteSpeed - built in integration

2.11.1 LVE PAM module

pam_lve.so is a PAM module that sets up LVE environment. It provides easy way to setup LVE for SSH sessions, as well as other PAM enabled applications, such as crontab, su, etc...

pam_lve.so is installed by default when you convert existing server.

Installation:

```
# yum install pam_lve
```

After you install RPM, add following line to PAM config file for the required application:

```
session required pam_lve.so 500 1 wheel,other
```

In this line:

- **500** stands for minimum UID for which LVE will be setup. For any user with UID < 500, LVE will not be setup. **If CageFS is installed, use:**
cagefsctl --set-min-uid UID to setup minimum UID. The parameter in PAM files will be ignored in that case.
- **1** stands for CageFS enabled (0 -- cagefs disabled)
- 3rd optional argument defines group of users that will not be placed into LVE or CageFS. Starting with pam_lve **0.3-7** you can specify multiple groups, coma separated

IT IS CRUCIAL TO PLACE ALL USERS THAT SU OR SUDO TO ROOT INTO THAT GROUP. OTHERWISE, ONCE SUCH USER GAINS ROOT, USER WILL BE INSIDE LVE, AND ALL APPLICATIONS RESTARTED BY THAT USER WILL BE INSIDE THAT USER LVE AS WELL.

For example, to enable LVE for SSH access, add that line to `/etc/pam.d/sshd`. To enable LVE for SU, add that line to `/etc/pam.d/su`

By default module will not place users with group wheel into lve. If you want to use different group to define users that will not be placed into LVE by pam_lve - pass it as 3rd argument.

WARNING: BE CAREFUL WHEN YOU TEST IT, AS IF YOU INCORRECTLY ADD THIS LINE TO /ETC/PAM.D/SSHD, IT WILL LOCK YOU OUT SSH. DON'T LOG OUT OF YOUR CURRENT SSH SESSION, UNTIL YOU SURE IT WORKS.

2.11.2 LVE Wrappers

LVE wrappers are the set of tools that allow system administrator to run various users, programs & daemons within Lightweight Virtual Environment. This allows system administrator to have control over system resources such program can have. Additionally it prevents misbehaving programs running within LVE to drain system resources and slow down or take down the whole system. The tools are provided by lve-wrappers RPM.

You can install them by running:

```
$ yum install lve-wrappers
```

Placing programs inside LVE

LVE Wrappers provide two tools for placing programs inside LVE: lve_wrapper and lve_suwrapper

`/bin/lve_wrapper` – can be used by any non-root user, as long as that user is in group lve (see `/etc/`

groups file).

Syntax:

```
lve_wrapper <command_to_run>
```

Example:

```
$ lve_wrapper make install
```

The program will be executed within LVE with ID matching user's id.

/bin/lve_suwrapper – can be used by root user or any user in group lve (see /etc/groupfile) to execute command within specified LVE

Syntax:

```
lve_suwrapper LVE_ID <command_to_run>
```

Example:

```
# lve suwrapper 10000 /etc/init.d/postgresql start
```

2.11.3 MPM ITK support

CloudLinux httpd RPM comes with MPM ITK built in. Yet, if you would like to build your own Apache, you need to apply our patch for MPM ITK

Download file: <http://repo.cloudlinux.com/cloudlinux/sources/da/cl-apache-patches.tar.gz>

Extract: `apache2.2-mpm-itk-securelve12.patch`

And apply this patch to your Apache source code.

When running MPM ITK, you should disable `mod_hostinglimits`. All the functionality needed by MPM ITK is already built into the patch.

Directives which can be used by Apache with ITK patch:

- `AssignUserID` - uses ID as LVE ID
- `LVEErrorCodeITK` - Error code to display on LVE error (508 by default)
- `LVERetryAfterITK` - same as `LVERetryAfter` - respond with Retry-After header when LVE error 508 occurs
- `LVEId` - overrides id used for LVE ID instead of `AssignUserID`
- `LVEUser` - overrides user to use to retrieve LVE ID, instead of `AssignUserID`

2.11.4 HostingLimits module for Apache

`mod_hostinglimits` works with existing cgi/php modules, to put them into LVE context. In most cases the cgi/php process will be placed into LVE with ID of user that sites belongs to. **`mod_hostinglimits`** detects the user from **`SuexecUserGroup`** (`suexec` module), `SuPHP_UserGroup` (from `mod_suphp`), **`AssignUserID`** (MPM ITK), **`RUIDGid`** (`mod_ruid2`) directives.

This can be overwritten via **`LVEId`** or **`LVEUser`** parameter on the Directory level. Note, that those parameters will not work with `mod_fcgid` and `mod_cgid`. The order of detection is like following:

- `LVEId`
- `LVEUser`
- `SuexecUserGroup`
- `suPHP_UserGroup`
- `RUIDGid`
- `AssignUserID`

LVE doesn't work for `mod_include` #include due to its "filter" nature.

Example:

```
LoadModule hostinglimits_module modules/mod_hostinglimits.so
<IfModule mod_hostinglimits.c>
    AllowedHandlers cgi-script php5-script php4-script
    SecureLinks On
</IfModule>
```

Installation

cPanel Installed by default during EasyApache build. Requires lve-stats & lve-utils packages to be installed.

DirectAdmin Can be built using `custombuild`

```
$ yum install liblve-devel
$ cd /usr/local/directadmin/custombuild
$ ./build update
$ ./build set cloudlinux yes
$ ./build apache
$ ./build rewrite_confs
if you run suphp, then run the following:
$ ./build suphp
```

Plesk `$ yum install mod_hostinglimits`

ISPmanager `$ yum install mod_hostinglimits`

InterWorx `$ yum install mod_hostinglimits`

H-Sphere Included by default in H-Sphere 3.5+

Standard Apache from RPM `$ yum install mod_hostinglimits`

Custom Apache installation Compile from source: http://repo.cloudlinux.com/cloudlinux/sources/mod_hostinglimits.tar.gz

```
$ wget http://repo.cloudlinux.com/cloudlinux/sources/mod_hostinglimits.tar.gz
$ yum install cmake
$ tar -zxvf mod_hostinglimits*.tar.gz
$ cd mod_hostinglimits*
$ cmake .
$ make
$ make install
```

Apache Module `hostinglimits_module`
 Identifier:
 Source Files: `mod_hostinglimits.c`
 Compatibility: MPM prefork, worker, event, ITK

Directives

SecureLinks

Description: Makes sure that for any virtual hosts, only files owned by user specified via SuexecUserGroup or other ways as described above are served. For files owned by any other user apache will return Access Denied error. The directive will not affect VirtualHost without user id specified, or with uid < 100

Syntax: SecureLinks On

Default: SecureLinks Off

Context: server config

Prevents apache from serving files not owned by user, stopping symlink attacks against php config files.

Example:

```
SecureLinks On
```

SkipErros

Description: Allow apache to continue if LVE is not available

Syntax: SkipErrors On

Default: SkipErrors On

Context: server config

Prevents apache from existing if LVE is not available.

Example:

```
SkipErrors Off
```

AllowedHandlers

Description: List of handlers that should be placed into LVE, support regexp

Syntax: AllowedHandlers cgi-script %^php% my-script

Default: none

Context: server config

This directive allows to list handlers which will be intercepted and placed into LVE.

Example:

Match requests handled by cgi-script handler

```
AllowedHandlers cgi-script
```

Match all requests

```
AllowedHandlers *
```

Match all requests that handled by handler that contains php

```
AllowedHandlers %php%
```

Match all requests handled by handler that starts with php

```
AllowedHandlers %^php%
```

DenyHandlers

Description: List of handlers that should not be placed into LVE, support regexp

Syntax: DenyHandlers text/html

Default: none

Context: server config

This directive works together with AllowHandlers, to exclude some handlers from being allowed in LVE

Example:

Match all requests, but text/*

```
AllowedHandlers *
```

```
DenyHandler %text/*%
```


LVEErrorCode

Description: Error code to display once entry is rejected due to maxEntryProcs

Syntax: values from 500 to 510

Default: 508

Context: directory config

Specifies ErrorCode to use on LVE error (like too many concurrent processes running). The message that will be displayed by default is:

```
Resource Limit Is Reached
The website is temporarily unable to server your request as it exceeded
resource limit.
Please try again later.
You can redefine error message using ErrorDocument directive
```

Example:

```
LVEErrorCode 508
ErrorDocument 508 508.html
```

LVEid

Description: Allows to setup separate LVE id on per directory level. If not set, user id of corresponding user is used

Syntax: LVEid number

Default: User Id is used

Context: directory config

Specifies LVE id for particular directory

Example:

```
<Directory "/home/user1/domain.com/forums">
  LVEid 10001
</Directory>
```

LVEUser

Description: Allows to setup separate LVE id on per directory level.

Syntax: LVEUser username

Default: none

Context: directory config

Specifies LVE id for particular directory

Example:

```
<Directory "/home/user1/domain.com/forums">
  LVEUser user1
</Directory>
```

LVEUserGroupID

Description: Use group ID instead of user ID for lve container number

Syntax: LVEUserGroupID On/Off

Default: User Id is used

Context: global config only

If option enabled, group id will be used instead of user id. Apache will display following string in error logs:

```
mod_hostinglimits: use GroupID instead of UID
```

```
mod_hostinglimits: found apr extension version 2
mod_hostinglimits: apr_lve_environment_init_group check ok
```

If compatible apr library is not found, following error message will display in error logs

```
mod_hostinglimits: apr_lve_* not found!!!
```

Example:

```
<Directory "/home/user1/domain.com/forums">
  LVEUserGroupID On
</Directory>
```

LVERetryAfter

Description: Returns Retry-After header when LVE error 508 occurs.

Syntax: LVERetryAfter MINUTES

Default: 240 minutes

Context: directory config

Specifies interval for Retry-After header. The Retry-After response-header field can be used w to indicate how long the service is expected to be unavailable to the requesting client.

Example:

```
LVERetryAfter 180
```

LVEsitesDebug

Description: Provides extended debug info for listed sites

Syntax: LVEsitesDebug test.com test2.com

Default: none

Context: directory config

Specifies virtual hosts for which to provide extra debugging information

Example:

```
<Directory "/home/user1/domain.com/forums">
  LVEsitesDebug abc.com yx.cnet
</Directory>
```

LVEParseMode

Description: Determines the way LVE id will be extracted. In Conf

Syntax: LVEParseMode CONF|PATH|OWNER|[REDIS](#)

Default: CONF

Context: directory config

In CONF mode, standard way to extract LVE id is used (SuexecUserGroup, LVEid, or similar directives.

In PATH mode, username is extracted from the home directory path. The default way to match username is via following regexp `/home/([^\s]*)/`. Custom regexp can be specified in LVEPathRegexp

In OWNER mode, owner of the file used as a LVE id

In [REDIS](#) mode, LVE id is retrieved from Redis database

Example:

```
LVEParseMode CONF
```

LVEPathRegexp

Description: Regexp used to extract username from the path. Used in conjunction with LVEParseMode PATH

Syntax: LVEPathRegexp regexp

Default: `/home/([^\s]*)/`

Context: directory config

Used to extract user's name via path.

Example:

```
LVEPathRegexp /home/([^\s]*)/
```

2.11.4.1 Redis Support for HostingLimits

Redis support provides a way to query Redis database for LVE id, based on domain in the HTTP request. Given a database like:

```
xyz.com 10001
bla.com 10002
....
```

the module will retrieve corresponding LVE id from the database.

To enable Redis support, compile from source: http://repo.cloudlinux.com/cloudlinux/sources/mod_hostinglimits.tar.gz

The compilation requires hiredis library

```
$ wget http://repo.cloudlinux.com/cloudlinux/sources/mod_hostinglimits.tar.gz
$ yum install cmake
$ tar -zxvf mod_hostinglimits*.tar.gz
$ cd mod_hostinglimits*
$ cmake -DREDIS:BOOL=TRUE .
$ make
$ make install
```

To enable Redis mode, specify:

```
LVEParseMode REDIS
```

LVERedisSocket

Description: Socket to use to connect to Redis database

Syntax: LVERedisSocket path

Default: /tmp/redis.sock

Context: server config

Used to specify location of Redis socket.

Example:

```
LVERedisSocket /var/run/redis.sock
```

LVERedisAddr

Description: IP/port used to connect to Redis database instead of socket

Syntax: LVERedisAddr IP PORT

Default: none

Context: server config

Used to specify IP & port to connect to Redis instead of using Socket

Example:

```
LVERedisAddr 127.0.0.1 6993
```

LVERedisTimeout

Description: Number of seconds to wait before attempting to re-connect to redis

Syntax: LERetryAfter SECONDS

Default: 60 seconds

Context: server config

Number of seconds to wait before attempting to reconnect to Redis after last unsuccessful attempt to connect

Example:

```
LVERedisTimeout 120
```

2.11.5 cPanel/WHM JSON API

CloudLinux offers JSON API for [lvectl](#) via WHM. You can access it using following URL:
`https://IP:2087/cpsess_YOURTOKEN/cgi/CloudLinux.cgi?cgiaction=jsonhandler&handler=list`
output will be like this:

```
{"data":[{"ID":"default","CPU":"30","NCPU":"1","PMEM":"1024M","VMEM":"1024M","EP":"28"}] "NPROC":"0"}
```

Parameters

cgiaction always *jsonhandler*
handler should match [lvectl](#) command

for commands like set, destroy & delete where you need to specify LVE (user) ID, like `lveid=500`
(matches user id 500)

Example:

```
https://IP:2087/cpsess_YOURTOKEN/cgi/CloudLinux.cgi?  
cgiaction=jsonhandler&handler=set&lveid=500&cpu=30&io=2048
```

output:

```
{"status":"OK"}
```

to do 'set default', use `lveid=0`, like:

```
https://IP:2087/cpsess_YOURTOKEN/cgi/CloudLinux.cgi?  
cgiaction=jsonhandler&handler=set&lveid=0&cpu=30&io=2048
```

for commands like apply all, destroy all, use:

```
handler=apply-all  
handler=destroy-all
```

3 LVE Manager

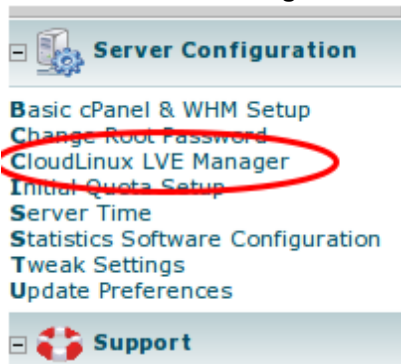
LVE Manager is a plugin for most popular control panels including cPanel, Plesk, DirectAdmin and ISPmanager (InterWorx coming soon). It allows you to control and monitor limits, and set limits on per package bases.

LVE Manager is installed by default on most servers. If it is missing you can always install it by running:

```
$ yum install lvemanager
```

3.1 cPanel LVE Manager

LVE Manager can be accessed through WHM interface. It is located under **Server Configuration -> CloudLinux LVE Manager**



LVE Manager provides an easy way to monitor current, real time usage

The screenshot shows the LVE Manager interface with the 'Current Usage' tab selected. At the top, there are controls for enabling/disabling the service, a refresh interval of 0 seconds, and a 'refresh page' button. Below this is a table showing usage for two accounts: 'ctest3' and 'ctest2'. Annotations with arrows point to the '3' in the refresh interval, the table, and the '0' in the refresh interval.

ID	EP	PNO	TNO	CPU	MEM	I/O
ctest3	1	2	2	0	1	0
ctest2	1	2	2	0	1	204

Adjust settings for individual accounts

Sortable headers Filtering by username

Current Usage Settings Statistics Packages Options Alternatives

▼ LVE id	▼ Username	▼ Domain	▼ CPU	▼ nCPU	▼ vMEM(MB)	▼ pMEM(MB)	▼ EP	▼ nPROC	▼ IO(KB/s)	▼ PACKAGE			
501	user11	user1.test1	25	2	-	-	20	-	700	test1	Edit	Reset	History
502	user12	user2.test1	44	1	-	1024	20	20	3333	test1	Edit	Reset	History
503	user21	user3.test2	33	1	-	512	30	35	1024	test2	Edit	Reset	History
504	user22	user4.test2	33	1	-	-	30	35	1024	test2	Edit	Reset	History
506	user6	user6.test	44	1	-	1024	20	20	1024	default	Edit	Reset	History
510	user31	user5.test3	55	1	1024	1024	20	12	1000	test3	Edit	Reset	History

Non-default values are in red Change individual account settings

Reset to use default package settings View usage history

View individual account usage history

Statistics Packages Options Alternatives LVE History

History for LVE 519 (cltest8 - cltest8.com)

Timeframe: Last 30 minutes ▼ Show

Last 10 minutes

Last 30 minutes

Last hour

Last 4 hours

Today

Yesterday

Last 7 days

Last 30 days

You can select time interval for the historical statistics

Current Usage Settings Statistics Packages Options Alternatives LVE History

History for LVE 507

Timeframe: Last 10 minutes Show

Sortabe Headers Average, Max & Limit for each resource Change timeframe at any moment Faults - if they are none zero, user's site might experience issues

From	To	CPU	MEM(MB)	pMEM(MB)	EP	nPROC	IO(KB/s)	VMemF	PMemF	EPf	NprocF
09-17 15:00	09-17 16:00	0 0 100	2 4 -	8 17 1024	0 1 25 -	1 24 3	1008 1024	0	0	0	0
09-17 16:00	09-17 17:00	0 0 100	- 4 -	4 18 1024	0 1 25 -	1 24 1	874 1024	0	0	0	0
09-18 13:00	09-18 14:00	0 4 100	- 26 -	1 37 1024	0 1 25 -	1 24 3	610 1024	0	0	0	0
09-18 14:00	09-18 15:00	0 29 100	42 95 -	58 111 1024	0 3 25 1	5 24 9	1003 1024	0	0	0	0
09-18 15:00	09-18 16:00	0 6 100	12 106 -	27 65 1024	0 1 25 -	1 24 7	1024 1024	0	0	0	0
09-18 16:00	09-18 17:00	0 4 100	- 18 -	5 32 1024	0 1 25 -	1 24 3	940 1024	0	0	0	0
09-18 17:00	09-18 18:00	0 0 100	- - -	- - - 1024	0 0 25 -	- 24 0	0 1024	0	0	0	0
09-18 18:00	09-18 19:00	0 0 100	- - -	- - - 1024	0 0 25 -	- 24 0	0 1024	0	0	0	0
09-18 19:00	09-18 20:00	0 0 100	- - -	- - - 1024	0 0 25 -	- 24 0	0 1024	0	0	0	0
09-18 20:00	09-18 21:00	0 0 100	- - -	- - - 1024	0 0 25 -	- 24 0	0 1024	0	0	0	0
09-18 21:00	09-18 22:00	0 0 100	- - -	- - - 1024	0 0 25 -	- 24 0	0 1024	0	0	0	0
09-18 22:00	09-18 23:00	0 0 100	- - -	- - - 1024	0 0 25 -	- 24 0	0 1024	0	0	0	0
09-18 23:00	09-19 00:00	0 0 100	- - -	- - - 1024	0 0 25 -	- 24 0	0 1024	0	0	0	0

Usage is displayed for particular time intervals

Statistics tab allows you to get historical data about LVE usage.

You can get accounts with highest number of faults, or highest average/max CPU/Memory/IO/etc.. usage by selecting Top LVEs. You can adjust number of LVEs in the report by selecting the number from Limit drop box.

You can get accounts that are using X% (by specifying Usage) of their Limit, by average or max CPU/ Memory/etc... by selecting LVE Approaching Limits.

Fault LVE lets you select accounts that experiences issues due to hitting particular limits. You can specify minimum number of faults in the report using Threshold

List top accounts based on resource usage

Current Usage Settings Statistics Packages Options Alternatives

Number of top accounts

Show accounts approaching limits

Show accounts that filed due to limits

Average, Max & Limit

Timeframe: Last 30 days

Limit: 100

Using: 50%

Threshold: 50

Accounts % close to limit

Number of faults

Time period for the faults

ID	Username	Domain	CPU	pMEM(MB)	EP	nPROC	IO(KB/s)	PMemF	EPf	NprocF
507	blissie	blissiest.com	0 13 44	2 111 1024	0 3 26 -	5 26 0	1024 1024	0	0	0
512	cltest2	cltest2.com	0 4 44	- 37 1024	0 17 20 -	9 20 0	1024 1024	0	0	0
513	cltest3	cltest3.com	0 0 33	- - 512	0 1 30 -	1 35 0	0 1024	0	0	0
32006	mailman	-	0 1 44	- 9 1024	0 0 20 -	1 20 0	272 1024	0	0	0

vMEM - unlimited

If limit not set for Resource, it will not show up / unlimited

See charts

From the *Statistics* results you can get charts for resource usage for particular LVE.



Packages tab lets you manage limits for packages. Each account belonging to a package will adhere to those limits



Current Usage Settings Statistics Packages Options Alternatives

Show resellers packages *Select if you want to see reseller plans in the list*

Package ID	CPU	nCPU	vMEM(MB)	pMEM(MB)	EP	nPROC	IO(KB/s)	
Business 1	10	1	-	256	10	100	1024	Edit
Business 2	15	1	-	368	15	100	1524	Edit
Business 3	20	2	-	738	20	100	2048	Edit
Personal 1	7	1	-	128	10	50	1024	Edit
Personal 2	10	1	-	256	15	50	1024	Edit
Personal 3	15	1	-	512	20	100	1524	Edit
Professional 1	50	2	-	512	25	100	2048	Edit
Professional 2	75	3	-	1024	50	200	4096	Edit
DEFAULT	44	1	-	1024	20	20	1024	

Sortable headers

Edit Plan Settings

Default settings can be edited from "Settings" menu

Editing Package Limits is very similar to editing limits for individual account:

Statistics Packages Options Alternatives Edit Package

Settings for package BUSINESS 3

CPU usage (CPU) *Current values are pre-set*

Number of cores for LVE (nCPU)

Virtual memory (vMEM) MB (0 - unlimited)

Physical memory (pMEM) MB (0 - unlimited)

Concurrent connections (EP)

Number of processes (nPROC) (0 - unlimited)

I/O limit (IO) KB/s (0 - unlimited)

Apply Cancel

vMEM, pMEM, nPROC & IO can be set unlimited by setting them to 0

Options tab lets you define if LVE usage & limits will show up to end customers in cPanel interface.

Settings Statistics Packages Options Alternatives

Hide LVE end user usage statistics

Apply Cancel

3.1.1 LVE Extensions for cPanel

When you need to change LVE Manager options in cPanel config file on big amount of servers, you don't have to edit file manually, therefore there is no need to login into cPanel on each server. Just go to WHM, choose CloudLinux and click on Options - and you will be able to change settings from here.

```
root@toaster [~]# grep lve /var/cpanel/cpanel.config
```

<code>lve_hideextensions</code>	Hides (when =1) range of php extensions for user in Select PHP version.
<code>lve_hideuserstat</code>	Hides (when =1) LVE statistics in cPanel Stats Bar (UI).
<code>lve_showinodeusage</code>	Displays (when =1) used inodes in cPanel (UI).
<code>lve_hide_selector</code>	Turns off UI PHP Selector (Select PHP Version option).

4 CageFS

CageFS is a virtualized file system and a set of tools to contain each user in its own 'cage'. Each customer will have its own fully functional CageFS, with all the system files, tools, etc...

The benefits of CageFS are:

- Only safe binaries are available to user
- User will not see any other users, and would have no way to detect presence of other users & their user names on the server
- User will not be able to see server configuration files, such as Apache config files.
- User's will have limited view of /proc file system, and will not be able to see other' users processes

At the same time, user's environment will be fully functional, and user should not feel in any way restricted. No adjustments to user's scripts are needed. CageFS will cage any scripts execution done via:

- Apache (suexec, suPHP, mod_fcgid, mod_fastcgi)
- LiteSpeed Web Server
- Cron Jobs
- SSH
- Any other PAM enabled service

* Note: *mod_php* is not supported, *MPM ITK* requires custom patch

4.1 Installation

Minimum Requirements:

- *kernel: CL5 with lve0.8.54 or later, CL6 with lve1.2.17.1 or later*
- *7GB of disk space*

Depending on your setup, and number of users, you might also need:

- *Up to 8MB per customer in /var directory (to store custom /etc directory)*
- *5GB to 20GB in /usr/share directory (to store safe skeleton of a filesystem)*

Warning: If at any time you decide to uninstall CageFS, please, make sure you follow [uninstall instructions](#)

To install CageFS:

```
$ yum install cagefs
$ /usr/sbin/cagefsctl --init
```

That last command will create skeleton directory that might be around 7GB in size. If you don't have enough disk space in /usr/share, use following commands to have cagefs-skeleton being placed in a different location:

```
$ mkdir /home/cagefs-skeleton
$ ln -s /home/cagefs-skeleton /usr/share/cagefs-skeleton
```

On cPanel servers, if you will be placing skeleton into /home directory, you must configure the following option in:

cPanel WHM WHM -> Server Configuration -> Basic cPanel/WHM Setup -> Basic Config -> Additional home directories

Change the value to blank (not default "home")

Without changing this option, cPanel will create new accounts in incorrect places.

CageFS will automatically detect and configure all necessary files for:

- cPanel
- Plesk
- DirectAdmin
- ISPmanager
- Interworx
- MySQL
- PostgreSQL
- LiteSpeed

Web interface to manage CageFS is available for cPanel, Plesk 10+, DirectAdmin, ISPmanager & Interworx. Command line tool would need to be used for other control panels.

Once you initialized the template you can start enabling users. By default CageFS is disabled for all users.

4.2 Uninstalling CageFS

To uninstall CageFS, start by disabling & removing all directories

```
$ /usr/sbin/cagefsctl --remove-all
```

That command will: Disable CageFS for all customers, unmount CageFS for all users, removes */usr/share/cagefs-skeleton* & */var/cagefs* directories. It will not remove */etc/cagefs* directory

Remove CageFS RPM:

```
$ yum remove cagefs
```

4.3 Managing Users

CageFS provides for two modes of operations:

1. Enabled for all, except those that are disabled
2. Disabled for all, except those that are enabled

Mode #1 is convenient for production operation, where you want all new users to automatically be added to CageFS.

Mode #2 is convenient while you test CageFS, as it allows you to enable it on one by one for your customers.

To start using CageFS you have to select one of the mode of operations.

```
$ /usr/sbin/cagefsctl --enable-all
```

or

```
$ /usr/sbin/cagefsctl --disable-all
```

or

```
$ /usr/sbin/cagefsctl --toggle-mode
```

That will switch the operation mode, preserving current disabled/enabled users.

To enable individual user do:

```
$ /usr/sbin/cagefsctl --enable [username]
```

To disable individual user:

```
$ /usr/sbin/cagefsctl --disable [username]
```

To list all enabled users:

```
$ /usr/sbin/cagefsctl --list-enabled
```

To list all disabled users

```
$ /usr/sbin/cagefsctl --list-disabled
```

To see current mode of operation:

```
$ /usr/sbin/cagefsctl --display-user-mode
```

4.4 Command line tools

cagefsctl is used to manage CageFS. It allows you to initialize and update CageFS, as well as enable/disable CageFS for individual users.

Usage: /usr/sbin/cagefsctl [OPTIONS]

Options:

- i | --init initialize CageFS (create CageFS if it does not exist)
- r | --reinit reinitialize CageFS (make backup and recreate CageFS)
- u | --update update files in CageFS (add new and modified files to CageFS, remove unneeded files)
- update-etc update /etc template only
- f | --force recreate CageFS (do not make backup, overwrite existing files)
- d | --dont- do not delete any files from skeleton (use with --update option)
- clean
- k | --hardlinkuse hardlinks if possible
- create-mp Creates /etc/cagefs/cagefs.mp file
- mount-skel mount CageFS skeleton directory and start cagefs-fuse service (if not started)
- unmount- unmount CageFS skeleton directory and stop cagefs-fuse service (if started)
- skel
- remove-all disable CageFS, remove templates and /var/cagefs directory
- addrpm add rpm-packages in CageFS (run "cagefsctl --update" in order to apply changes)
- delrpm remove rpm-packages from CageFS (run "cagefsctl --update" in order to apply changes)
- list-rpm list rpm-packages that are installed in CageFS
- e | --enter enter into user's CageFS as root
- enable- enable CageFS
- cagefs
- disable- disable CageFS
- cagefs
- set-min-uid Set min UID
- get-min-uid Display current MIN_UID setting
- do-not-ask assume "yes" in all queries (should be the first option in command)
- set-update- set minimum period for doing update of a skeleton (default 1 day)
- period
- force- forces the update even if min period is yet to be reached
- update
- tmpwatch forces clean up of all user's tmp directories
- set- set command to run to clean up end user tmp directories, like: --set-tmpwatch='/usr/sbin/
- tmpwatch tmpwatch -umclq 720'
- m | -- remount specified user(s)

```

remount
-M | -- restart cagefs-fuse service, remount CageFS skeleton directory and all users, (use this
remount-all each time you have changed cagefs.mp file)
-w | -- unmount specified user(s)
unmount
-W | -- stop cagefs-fuse service, unmount CageFS skeleton directory and all users
unmount-all
--create-virt- creates virt.mp file for Plesk user
mp
USERNAME
--create-virt- creates virt.mp file for all Plesk users
mp-all
-l | --list list users that entered in CageFS
--enable enable CageFS for the user
--disable disable CageFS for the user
--enable-all enable all users, except specified in /etc/cagefs/users.disabled
--disable-all disable all users, except specified in /etc/cagefs/users.enabled
--display- display current mode ("Enable All" or "Disable All")
user-mode
--toggle- toggle mode saving current lists of users, (lists of enabled and disabled users remain
mode unchanged)
--list-enabled list enabled users
--list- list disabled users
disabled
--getprefix display prefix for user

```

4.5 Running Command Inside CageFS

[lve-wrappers 0.6-1+]

Sometimes you will want to execute a command, as user, inside CageFS.

If user has shell enabled - you can simply use

```
$ /bin/su - $USERNAME -c "command "
```

Yet, if user has shell disabled, it wouldn't work. To solve this issue, we have added command:

```
$ /sbin/cagefs_enter user $USERNAME "command "
```

4.6 CageFS Quirks

Due to the nature of CageFS, some things will not work as before or require some changes:

- lastlog will not work (/var/log/lastlog)
- PHP will load php.ini from /usr/selector/php.ini. That file is actually a link to a real php.ini file from your system. So the same php.ini will be loaded in the end
- You have to run cagefsctl --update any time you have modified php.ini, or you want to get new / updated software inside CageFS

4.7 Configuration

4.7.1 File System Templates

CageFS creates a filesystem template in `/usr/share/cagefs-skeleton` directory. CageFS template will be mounted for each customer. The template is created by running:

```
# /usr/sbin/cagefsctl --init
```

To update the template, you should run:

```
$ /usr/sbin/cagefsctl --update
```

The behavior of the commands (and the files copied into `/usr/share/cagefs-skeleton` directory) depends on the configuration files in `/etc/cagefs/conf.d`

You can add additional files, users, groups and devices into CageFS template by adding `.cfg` file, and running:

```
$ /usr/sbin/cagefsctl --update
```

To delete files from CageFS template, remove corresponding `.cfg` file, and run:

```
$ /usr/sbin/cagefsctl --update
```

Here is an example `openssh-clients.cfg` file:

```
[openssh-clients]
comment=OpenSSH Clients
paths=/etc/ssh/ssh_config, /bin/hostname, /usr/bin/scp, /usr/bin/sftp, /usr/bin/
slogin, /usr/bin/ssh, /usr/bin/hoststat, /usr/bin/ssh-add, /usr/bin/ssh-agent, /usr/bin/ssh-copy-id, /
usr/bin/.ssh.hmac, /usr/bin/ssh-keyscan, /usr/libexec/openssh/sftp-server, /etc/
environment, /etc/security/pam_env.conf
devices=/dev/ptmx
```

Example `mail.cfg` file

```
[mail]
comment=Mail tools
paths=/bin/mail, /etc/aliases.db, /etc/mail, /etc/mailcap, /etc/mail.rc, /etc/
mime.types, /etc/pam.d/smtp.sendmail, /etc/rc.d/init.d/sendmail, /etc/smrsh, /etc/
sysconfig/sendmail, /usr/bin/hoststat, /usr/bin/Mail, /usr/bin/mailq.sendmail, /usr/
bin/makemap, /usr/bin/newaliases.sendmail, /usr/bin/purgestat, /usr/bin/
rmail.sendmail, /usr/lib64/sasl2/Sendmail.conf, /usr/lib/mail.help, /usr/lib/
mail.tildehelp, /usr/lib/sendmail.sendmail, /usr/sbin/mailstats, /usr/sbin/makemap, /
usr/sbin/praliases, /usr/sbin/sendmail.sendmail, /usr/sbin/smrsh, /var/log/mail, /var/
spool/clientmqueue, /var/spool/mqueue
users=smmisp
groups=smmisp
```

There is an easy way to add/delete files from particular RPMs into CageFS. That can be done by using `--addrpm` and `--delrpm` options in `cagefsctl`. Like:

```
$ cagefsctl --addrpm ffmpeg
$ cagefsctl --update
```

Please, note that `ffmpeg` RPM should be installed on the system already.

4.7.2 Excluding files

To exclude files and directories from CageFS, edit file:

```
/etc/cagefs/black.list
```

And add files or directories in there, one per line.

4.7.3 Excluding Users

To exclude users from CageFS, create a file (any name would work) inside

```
/etc/cagefs/exclude
```

folder, and list users that you would like to exclude from CageFS in that file.

4.7.4 Mount Points

CageFS creates individual namespace for each user, making it impossible for users to see each other's files and creating high level of isolation. The way namespace is organized:

1. `/usr/share/cagefs-skeleton` with safe files is created
2. Any directory from the server that needs to be shared across all users is mounted into `/usr/share/cagefs-skeleton`
 - a. list of such directories is defined in `/etc/cagefs/cagefs.mp`
3. `/var/cagefs/[prefix]/username` directory for each user. Prefix is defined as last two digits of user id. User id is taken from `/etc/passwd` file.
4. Separate `/etc` directory is created and populated for each user inside `/var/cagefs/[prefix]/username`
5. `/tmp` directory is mounted for each user separately into `~username/.cagefs-tmp directory`
6. Additional custom directories can be mounted for each user by defining them in `/etc/cagefs/cagefs.mp`
7. You can define custom directories per user using [virt.mp](#) files [CageFS 5.1 and higher]

To define individual custom directories in `/etc/cagefs/cagefs.mp` following format is used:

`@/full/path/to/directory,permission notation`

This is useful when you need to give each user its own copy of a particular system directory, like:

`@/var/run/screen,777`

Such entry would create separate `/var/run/screen` for each user, with permissions set to `777`

To modify mount points, edit `/etc/cagefs/cagefs.mp`. Here is an example of `cagefs.mp`:

```
/var/lib/mysql
/var/lib/dav
/var/www/cgi-bin
/var/spool
/dev/pts
/usr/local/apache/domlogs
/proc
/opt
@/var/spool/cron,700
@/var/run/screen,777
```

If you want to change mount points, make sure you re-initialize mount points for all customers:

```
$ cagefsctl --remount-all
```

This command will kill all current processes and reset mount points.

4.7.4.1 Per user virtual mount points

[CageFS 5.1 and higher]

* Please, see [Split by username](#) feature, as it might be more simpler to implement in some cases.

Starting with CageFS 5.1 you can specify additional directories to be mounted inside user's CageFS. This can be specified for each user.

To specify virtual mount points for a user, create a file:

```
/var/cagefs/[prefix]/[user]/virt.mp
```

Inside that file, you can specify mount points in the following format:

```
virt_dir1,mask
@sub_dir1,mask
@sub_dir2,mask
virt_dir2,mask
@sub_dir3,mask
@sub_dir4,mask
^virt_dir3,mask
@sub_dir5,mask
@sub_dir6,mask
# comments
```

- mask is always optional, if missing 0755 is used
- Create virtual directory sub_dir/virt_dir, mount it to:
 - skeleton jail_dir/virt_dir
 - inside virtual directory, create directories sub_dir1, sub_dir2
 - mount virt_dir1/sub_dir1 to sub_dir/virt_dir/sub_dir1
 - if virt_dir is started with ^, create directory sub_dir/virt_dir, but don't mount it into jail_dir. This is needed for cases when virt_dir is inside home base dir.
- if file /var/cagefs/[prefix]/[user]/virt.mp is missing -- no virt directories are loaded for that user.

Note that CageFS will automatically create those files for Plesk 10 & higher.

For example if we have plesk11.5 with two users cltest1, and cltest2

```
cltest1 uid 10000 has domains: cltest1.com, cltest1-addon.com and sub1.cltest1.com
cltest2 uid 10001 has domains: cltest2.com, cltest2-addon.com
```

In such case we would have file

/var/cagefs/00/cltest1/virt.mp:

```
>/var/www/vhosts/system,0755
@cltest1-addon.com,0755
@cltest1.com,0755
@sub1.cltest1.com,0755
```

and file: /var/cagefs/01/cltest2/virt.mp

```
>/var/www/vhosts/system
@cltest2-addon.com
@cltest2.com
```

4.7.4.2 Split by username

[CageFS 5.3.1+]

Sometimes you might need to make sure that directory containing all users would show up as containing just that user inside CageFS. For example, if you have directory structure like:

```
/home/httpd/fcgi-bin/user1
/home/httpd/fcgi-bin/user2
```

Then we can add the following line to /etc/cagefs/cagefs.mp file:

```
%/home/httpd/fcgi-bin
```

and execute:

```
cagefsctl --remount-all
```

After that each subdirectory of `/home/httpd/cgi-bin` will be mounted for appropriate user in CageFS: `/home/httpd/cgi-bin/user1` will be mounted for `user1` and `/home/httpd/cgi-bin/user2` will be mounted for `user2`.

4.7.5 Base Home Directory

If you have a custom setup where home directories are in a special format, like: `/home/$USERNAME/data`, you can specify it using regular expressions. This is needed by CageFS to create safe home space for end user, where no other users are visible. We will create empty: `/var/cagefs/[prefix]/$USERNAME/home`, and then mount `/home/$USERNAME` in that directory

To do that, create file:
`/etc/cagefs/cagefs.base.home.dirs`

With content like:

```
^/home/  
^/var/www/users/
```

If there is no such file, the home directory without last component will be considered as a base dir, like with `/home/$USERNAME` we would create `/var/cagefs/[prefix]/$USERNAME/home`, and then mount `/home/$USERNAME` in there

With `/home/$USERNAME/data` as a home dir, we would assume that `/home/$USERNAME` is the base directory, and we would create `/var/cagefs/[prefix]/$USERNAME/home/$USERNAME/data` and then we would mount `/home/$USERNAME/data` -- which would cause each user to see empty base directories for other users, exposing user names.

Sharing home directory structure among users

When you want to share directory structure among multiple users, you can add following line at the top of the `/etc/cagefs/cagefs.base.home.dirs` file. This is useful on the systems that support sites with multiple users, with different home directories inside main 'site' directory.

```
mount_basedir=1
```

For example:

user1 has home directory `/var/www/vhosts/sitename.com/web_users/user1`
user2 has home directory `/var/www/vhosts/sitename.com/web_users/user2`
site admin has home directory `/var/www/vhosts/sitename.com`

So, content of `/etc/cagefs/cagefs.base.home.dirs` should be the following:

```
mount_basedir=1  
^/var/www/vhosts/[^/]+
```

Directory structure in `/var/www/vhosts/sitename.com` will be mounted in CageFS for appropriate users. Each user will have access to whole directory structure in `/var/www/vhosts/sitename.com` (according to their permissions).

* Note: you should execute `cagefsctl --remount-all` in order to apply changes to CageFS (i.e. remount home directories).

4.7.6 PostgreSQL support

CageFS provides separate `/tmp` directory for each end user. Yet, PostgreSQL keeps its Unix domain socket inside server's main `/tmp` directory. In addition to that -- the location is hard coded inside PostgreSQL libraries.

To resolve the issue, CloudLinux provides version of PostgreSQL with modified start up script that can store PostgreSQL's socket in `/var/run/postgres`. The script automatically creates link from `/tmp` to that socket to prevent PostgreSQL dependent applications from breaking.

In addition to that, CageFS knows how to correctly link this socket inside end user's `/tmp` directory.

To enable PostgreSQL support in CageFS:

1. make sure you have updated to latest version of PostgreSQL
2. Edit file `/etc/sysconfig/postgres`, and uncomment `SOCK_DIR` line
3. Restart PostgreSQL by running:

```
$ service postgresql restart
```

If you are using cPanel, you would also need to modify file: `/etc/cron.daily/tmpwatch`

And update line

```
flags=-umc
```

to:

```
flags=-umcl
```

To prevent symlink from being removed.

4.7.7 PAM configuration

CageFS depends on `pam_lve` module for PAM enabled services. When installed the module is automatically installed for following services:

- sshd
- crond
- su

Following line is added to corresponding file in `/etc/pam.d/`

```
session    required    pam_lve.so    100    1
```

Where 100 stands for minimum UID to put into CageFS & LVE, and 1 stands for CageFS enabled.

4.7.8 Executing By Proxy

Some software has to run outside of CageFS to be able to complete its job. This includes such programs as `passwd`, `sendmail`, etc...

CloudLinux uses **proxyexec** technology to accomplish such goal. You can define any program to run outside CageFS, by specifying it in `/etc/cagefs/proxy.commands`

Once program is defined, execute:

```
$ cagefsctl --update
```

To populate the skeleton

All the cPanel scripts located in `/usr/local/cpanel/cgi-sys/` that user might need to execute should be added to `proxy.commands`

4.7.9 Custom /etc direcotry

[4.0-5 and later]

To create custom file in /etc directory for end user, create a directory:

```
/etc/cagefs/custom.etc/[username]
```

Put all custom files, and sub-directories into that direcotry.

For example, if you want to create custom /etc/hosts file for USER1, create a directory:

```
/etc/cagefs/custom.etc/USER1
```

Inside that directory, create a file hosts, with the content for that user.

After that execute:

```
$ cagefsctl --update-etc USER1
```

If you are making changing for multiple users, you can run:

```
$ cagefsctl --update-etc
```

To remove custom file, remove it from /etc/cagefs/custom.etc/[USER] directory, and re-run

```
$ cagefsctl --update-etc
```

4.7.10 Moving cagefs-skeleton directory

Sometimes you might need to move cagefs-skeleton from /usr/share to another partition. There are two ways:

If */usr/share/cagefs-skeleton* is not created yet (cagefsctl --init wasn't executed yet), execute:

```
$ mkdir /home/cagefs-skeleton
$ ln -s /home/cagefs-skeleton /usr/share/cagefs-skeleton
$ cagefsctl --init
```

If */usr/share/cagefs-skeleton* already exists:

```
$ cagefsctl --disable-cagefs
$ cagefsctl --unmount-all
# To ensure that the following command prints empty output:
$ cat /proc/mounts | grep cagefs
# if you see any cagefs entries, execute "cagefsctl --unmount-all" again.
$ mv /usr/share/cagefs-skeleton /home2/cagefs-skeleton
$ ln -s /home2/cagefs-skeleton /usr/share/cagefs-skeleton
cagefsctl --enable-cagefs
```

4.7.11 Moving /var/cagefs directory

To move /var/cagefs to another location:

```
$ cagefsctl --disable-cagefs
$ cagefsctl --unmount-all
```

Verify that `/var/cagefs.bak` directory does not exist (if it exists - change name "cagefs.bak" to something else)

```
$ cp -rp /var/cagefs /new/path/cagefs
$ mv /var/cagefs /var/cagefs.bak
$ ln -s /new/path/cagefs /var/cagefs
$ cagefsctl --enable-cagefs
$ cagefsctl --remount-all
```

Verify that the following command gives empty output:

```
$ cat /proc/mounts | grep cagefs.bak
```

Then you can safely remove `/var/cagefs.bak`:

```
$ rm -rf /var/cagefs.bak
```

4.7.12 TMP directories

CageFS makes sure that each user has its own `/tmp` directory, and that directory is part of end-user's quota. The actual location of the directory is `$USER_HOME/.cagefs/tmp`

Once a day, using cron job, CageFS will clean up user's `/tmp` directory from all the files that haven't been modified in 30 days.

This can be changed by running:

```
$ cagefsctl --set-tmpwatch='/usr/sbin/tmpwatch -umclq 720'
```

Where 720 is how old file should be before it is removed. The period is in hours.

You can also force the clean up of all user's `/tmp` directories without waiting for a cron job, by running:

```
$ cagefsctl --tmpwatch
```

4.7.13 Syslog

By default, `/dev/log` should be available inside end user's CageFS. This is needed so that user's cronjobs and other things that user `/dev/log` would get recorded in the system log files.

This is controlled using file `/etc/rsyslog.d/schroot.conf` with the following content:

```
$AddUnixListenSocket /usr/share/cagefs-skeleton/dev/log
```

To remove presence of `/dev/log` inside CageFS, remove that file, and restart rsyslog service.

4.8 Control Panel Integration

CageFS comes with a plugin for various control panels. The plugins allow to:

- Initialize CageFS
- Select [mode of operation](#)
- See & modify list of enabled/disabled users
- Update CageFS skeleton

4.8.1 cPanel

CageFS plugin for cPanel is located in *Plugins* section of WHM. It allows to initialize CageFS, select for which users CageFS will be enabled as well as update CageFS skeleton.

Change Log
 Enable/Disable Outlook
 AutoConfig
 Install cPAddons
 Manage cPAddons
 Manage Plugins
 Modify cPanel & WHM News
 Reset a Mailman Password
 Shopping Cart Reset
 Synchronize FTP Passwords
 Upgrade to Latest Version

SSL/TLS

Generate a SSL Certificate and Signing Request
 Install a SSL Certificate and Setup the Domain
 Manage SSL Hosts
 Purchase & Install SSL Certificate
 SSL Key/Crt Manager

Restart Services

DNS Server (BIND/NSD)
 E-Commerce Server (Interchange)
 FTP Server (ProFTPD/PureFTPd)
 HTTP Server (Apache)
 IMAP Server (Courier/Dovecot)
 Mail Server (Exim)
 Mailing List Manager (Mailman)
 POP3 Server (cPOP)
 SQL Server (MySQL)
 SSH Server (OpenSSH)

Plugins

CageFS

CageFS Init Page:

CageFS was Initialized. See below for more information
[Download Init Log](#)

```

42154. Copying /usr/local/cpanel/whostmgr/docroot/templates/downloadloglist.tmpl to /usr/share/cagefs-
skeleton/usr/local/cpanel/whostmgr/docroot/templates/downloadloglist.tmpl
42155. Copying /usr/local/cpanel/whostmgr/docroot/templates/suspendreseller.tmpl to /usr/share/cagefs-
skeleton/usr/local/cpanel/whostmgr/docroot/templates/suspendreseller.tmpl
42156. Copying /usr/local/cpanel/whostmgr/docroot/templates/getacctlist.tmpl to /usr/share/cagefs-
skeleton/usr/local/cpanel/whostmgr/docroot/templates/getacctlist.tmpl
42157. Copying /usr/local/cpanel/whostmgr/docroot/templates/savebackup.tmpl to /usr/share/cagefs-
skeleton/usr/local/cpanel/whostmgr/docroot/templates/savebackup.tmpl
42158. Create directory /usr/share/cagefs-skeleton/usr/local/cpanel/etc
42159. Create directory /usr/share/cagefs-skeleton/usr/local/cpanel/etc/suspended.page
42160. Copying /usr/local/cpanel/etc/suspended.page/index.html to /usr/share/cagefs-skeleton/usr/local/cpanel/etc/susp
42161. Create directory /usr/share/cagefs-skeleton/usr/local/cpanel/etc/webtemplates
42162. Create directory /usr/share/cagefs-skeleton/usr/local/cpanel/etc/webtemplates/english
42163. Copying /usr/local/cpanel/etc/webtemplates/english/default.tmpl to /usr/share/cagefs-skeleton/usr/local/cpanel/e
42164. Copying /usr/local/cpanel/etc/webtemplates/english/moving.tmpl to /usr/share/cagefs-skeleton/usr/local/cpanel/e
42165. Copying /usr/local/cpanel/etc/webtemplates/english/redirect.tmpl to /usr/share/cagefs-skeleton/usr/local/cpanel/e
42166. Copying /usr/local/cpanel/etc/webtemplates/english/suspended.tmpl to /usr/share/cagefs-
skeleton/usr/local/cpanel/etc/webtemplates/english/suspended.tmpl
42167. Source file(s) /usr/local/safe-bin do not exist
42168. Creating symlink /usr/share/cagefs-skeleton/usr/bin/php5 to /usr/local/bin/php
42169. /usr/share/cagefs-skeleton/usr/lib/libssl.so.6 already exists, will not touch it
42170. /usr/share/cagefs-skeleton/usr/lib/libcrypto.so.6 already exists, will not touch it
42171. /usr/share/cagefs-skeleton/usr/lib/libz.so.1 already exists, will not touch it
  
```

The screenshot shows the WHM Accelerated 2 interface. At the top, there is a navigation bar with links for Home, News, Change Log, Secure (with a lock icon), and Logout (root). Below the navigation bar, the title "CageFS User Manager:" is displayed. A toggle switch is present, currently set to "New users will be in CageFS".

The interface is divided into two main sections: "Enabled Users (1)" and "Disabled Users (4)".

- Enabled Users (1):** A list box containing the user "cltest1".
- Disabled Users (4):** A list box containing the users "secvle", "testuser", "user2", and "user3". The "testuser" entry is highlighted in blue.

Between the two list boxes are two buttons: "<<" (left arrow) and ">>" (right arrow), used for moving users between the enabled and disabled states.

At the bottom of the interface, there are two buttons: "Disable CageFS" and "Update CageFS Skeleton".


4.8.2 Plesk

CageFS comes with a plugin for Plesk 10.x. It allows you to initialize and update CageFS template, as well as manage users and mode of operation for CageFS


Home ▶


Modules


Tools

 Manage Modules

Modules

 CageFS

 LVE Manager

 Products from Parallels Partners

Home ▶ Modules ▶

CageFS User Manager

New users will **not** be in CageFS by default (disable all)

Disabled:

ferrary3

pupkin2

<<

>>










Enabled:

ferrary2

pupkin

4.8.3 ISPManager

CageFS comes with plugin for ISP Manager to enable/disable CageFS on per user bases (via users' list, Edit->Permissions tab)

Name	Owner	Preset	Properties	Disk quota	Bandwidth	CageFS status
anton	root	custom	   	0 / 0	0 / 100000000	Enabled
mgrtest	root	custom	    	0 / 100	0 / 0	Disabled



Edit user - anton




User Permissions Limits Resources Notes

- Shell access
- SSL
- CGI
- SSI
- PHP as an Apache module
- PHP as CGI
- PHP safe_mode
- CageFS User Mode

Ok Cancel

Or you can manage global CageFS settings via CageFS menu

193.170.230.219




CageFS Information

New users will be in CageFS

Enabled Users (2):
anton; test;

Disabled Users (1):
mgrtest;

5 MySQL Governor

MySQL Governor 0.8-32+

MySQL governor is software to monitor and restrict MySQL usage in shared hosting environment. The monitoring is done via resource usage statistics per each MySQL thread. MySQL governor can also kill off slow SELECT queries.

MySQL Governor has multiple modes of operations, depending on the configuration. It can work in monitor only mode, or it can use different throttling scenarios.

MySQL Governor allows to restrict customers who use too much resources. It supports following limits:

CPU	%	CPU speed relative to one core. 150% would mean one and a half cores
READ	bytes	bytes read. Cached reads are not counted, only those that were actually read from disk will be counted
WRITE	bytes	bytes written. Cached writes are not counted, only once data is written to disk, it is counted

You can set different limits for different periods: current, short, med, long. By default those periods are defined as 1 second, 5 seconds, 1 minute and 5 minutes. They can be re-defined using [configuration file](#). The idea is to use larger acceptable values for shorter periods. Like you could allow a customer to use two cores (200%) for one second, but only 1 core (on average) for 1 minute, and only 70% within 5 minutes. That would make sure that customer can burst for short periods of time.

When customer is restricted, the customer will be placed into special LVE with ID 3. All restricted customers will be placed into that LVE, and you can control amount of resources available to restricted customers. Restricted customers will also be limited to only 30 concurrent connections. This is done so they wouldn't use up all the MySQL connections to the server.

5.1 Installation

To install MySQL governor on your server:

```
$ yum remove db-governor db-governor-mysql # you can ignore errors if you don't have those packages installed
$ yum install governor-mysql
$ /usr/share/lve/dbgovernor/mysqlgovernor.py --install
```

If you are installing CloudLinux on a server running MariaDB already, do instead:

```
$ yum install governor-mysql
$ /usr/share/lve/dbgovernor/db-select-mysql --mysql-version=mariadb55
$ /usr/share/lve/dbgovernor/mysqlgovernor.py --install
```

* The installation currently supports only cPanel, Plesk, DirectAdmin, ISPmanager, InterWorx as well as servers without control panel

** Please, note that MySQL/MariaDB will be updated from CloudLinux repositories

*** MySQL Governor is compatible only with MySQL 5.x & MariaDB

5.2 Removing MySQL Governor

To remove MySQL governor:

```
$ /usr/share/lve/dbgovernor/mysqlgovernor.py --delete
```

The script will install original MySQL server, and remove MYSQL Governor.

5.3 Modes Of Operation

[MySQL governor 1.0+]

MySQL Governor has multiple modes of operation. Some of them are experimental at this moment.

Mode:

off -- Monitor Only: In this mode MySQL governor will not throttle customer's queries, instead it will let you monitor the MySQL usage to see the abusers at any given moment of time (and historically). This mode is good when you are just starting and want to see what is going on

single -- Single restricted's LVE for all restricted customers (deprecated): In that mode once customer reaches the limits specified in the MySQL governor, all customer's queries will be running inside LVE with id 3. This means that when you have 5 customers restricted at the same time, all queries for all those 5 customers will be sharing same LVE. The larger the number of restricted customers - the less resources per restricted customer will be available

abusers - Use LVE for a user to restrict queries (default mode): In that mode, once user goes over the limits specified in the MySQL governor, all customer's queries will execute inside that user's LVE. We believe this mode will help with the condition when the site is still fast, but MySQL is slow (restricted) for that user. If someone abuses MySQL, it will cause queries to share LVE with PHP processes, and PHP processes will also be throttled, causing less of a new queries being sent to MySQL. *Requires dbuser-map file*

all - Always run queries inside user's LVE: This way there are no need for separate limits for MySQL. Depending on overhead we see in the future, we might decide to use it as a primary way of operating MySQL Governor. The benefits of this approach is that limits are applied to both PHP & MySQL at the same time, all the time, preventing any spikes what so ever. *Requires dbuser-map file*

If dbuser-map file is absent on the server, "abusers" mode works emulate "single".

With **single** and **abusers** mode, once user is restricted, the queries for that user will be limited as long as user is using more than limits specified. After a minute that user is using less, we will unrestricted that user.

You can specify modes of operation using [dbctl](#) or by changing [configuration file](#). dbuser-map file is located at `/etc/container/dbuser-map`

5.4 Configuration

MySQL Governor configuration is located in `/etc/container/mysql-governor.xml`
It is best to modify it using [dbctl](#) tool.

Once configuration file is updated, please, restart the governor using:

```
$ service db governor restart
```

Example configuration:

```
<governor>
<!-- 'off' - do not throttle anything, monitoring only -->
<!-- 'abusers' - when user reaches the limit, put user's queries into LVE for that user -->
<!-- 'all' - user's queries always run inside LVE for that user -->
```

```

<!-- 'single' - single LVE=3 for all abusers. -->
<!-- 'on' - deprecated (old restriction type) -->
<!-- To change resource usage of restricted user in LVE mode use command /usr/sbin/lvectl set 3 --
cpu=<new value> --ncpu=<new value> --io=<new value> --save-all-parameters -->
<lve use="on|single|off|abusers|all"/>

<!-- connection information -->
<!-- If host, login and password are not present, this information is taken from /etc/my.cnf and ~root/
.my.cnf -->
<!-- Use symbol specified in prefix to figure out hosting accounts (mysql username will be split using
prefix_separator, and first part will be used as account name). If prefix is not set, or empty -- don't use
prefixes/accounts -->

<!-- db governor will try to split MySQL user names using prefix separator (if present) and statistics will
be aggregated for the prefix (account name) -->
<connector host="..." login="..." password=".." prefix_separator="_"/>

<!-- Intervals define historical intervals for burstable limits. In seconds -->
<intervals short="5" mid="60" long="300"/>

<!-- log all errors/debug info into this log -->
<log file="/var/log/dbgovernor-error.log" mode="DEBUG|ERROR"/>

<!-- s -- seconds, m -- minutes, h -- hours, d -- days -->
<!-- on restart, restrict will disappear -->
<!-- log file will contain information about all restrictions that were take -->
<!-- timeout - penalty period when user not restricted, but if he hit his limit during this period he will be
restricted with higher level of restrict (for more long time) -->
<!-- level1, level2, level3, level4 - period of restriction user for different level of restriction. During this period
all user's requests will be placed into LVE container -->
<!-- if user hits any of the limits during period of time specified in timeout, higher level of restrict will be
used to restrict user. If user was already on level4, level4 will be applied again -->
<!-- attribute format set an restrict log format:
SHORT - restrict info only
MEDIUM - restrict info, _all_tracked_values_
LONG - restrict info, _all_tracked_values_, load average and vmstat info
VERYLONG - restrict info, _all_tracked_values_, load average and vmstat info, slow query info
-->
<!-- script -- path to script to be triggered when account is restricted -->
<!-- user_max_connections - The number of simultaneous connections of blocked user (in LVE mode) -->

<!-- restriction levels/format are deprecated -->
<restrict level1="60s" level2="15m" level3="1h" level4="1d" timeout="1h"
log="/var/log/dbgovernor-restrict.log" format="SHORT|MEDIUM|LONG|VERYLONG"
  script="/path/to/script"
  user_max_connectins="30"/>

<!-- period (deprecated) - period based restriction that has multiple levels (see above) -->
<!-- limit (by default) - when user hits limits, the account will be marked as restricted and if user does not
hit limit again during "unlimit=1m" account will be unrestricted. This mode doesn't have any additional
levels/penalties. -->
<restrict_mode use="period|limit" unlimit="1m"/>

```

```
<!-- killing slow SELECT queries (no other queries will be killed) -->
<!-- if "log" attribute was set all killed queries will be saved in log file -->
<slow_queries run="on|off" log="/var/log/dbgovernor-kill.log"/>

<!-- Enable or disable saving of statistics for lve-stats - On - enabled, Off-disabled -->
<statistic mode="on|off"></statistic>
<!-- Enable logging user queries on restrict, can be On or Off -->
<!-- Files saves in /var/lve/dbgovernor-store and be kept here during 10 days -->
<logqueries use="on|off"></logqueries>
<default>
<!-- -1 not use limit(by default, current - required) -->
<limit name="cpu" current="150" short="100" mid="90" long="65"/>
<limit name="read" current="100000000" short="90000000" mid="80000000" long="70000000"/>
<limit name="write" current="100000000" short="90000000" mid="80000000" long="70000000"/>
<!-- Time to kill slow SELECT queries for account, can be different for accounts in seconds -->
<!-- enabled only when slow_queries run="on" -->
<limit name="slow" current="30"/>
</default>
<!-- name will matched account name, as extracted via prefix extraction -->

<!-- mysql_name will match exact MySQL user name. If both name and mysql_name are present,
system will produce error -->
<!-- mode restrict -- default mode, enforcing restrictions -->
<!-- mode norestrict -- track usage, but don't restrict user -->
<!-- mode ignore -- don't track and don't restrict user -->
<user name="xxx" mysql_name="xxx" mode="restrict|norestrict|ignore">
<limit...>
</user>

<!-- debug mode for particular user. The information logged to restrict log. -->
<debug_user name="xxx"/>

</governor>
```

5.5 Starting And Stopping

To start:

```
$ service db_governor start
```

To stop:

```
$ service db_governor stop
```

5.6 User to Database mapping

[MySQL Governor 1.x]

Traditionally MySQL Governor used prefixes to map user to database. With the latest version, we automatically generate user -> database user mapping for cPanel and DirectAdmin control panels (other panels will follow).

The mapping file is located at: `/etc/container/dbuser-map`

The format of the file:

```
[dbuser_name1] [account_name1] [UID1]
...
[dbuser_nameN] [account_nameN] [UIDN]
```

For example:

```
pupkinas_u2 pupkinas 502
pupkinas_u1 pupkinas 502
pupkinas_u3 pupkinas 502
pupkin2a_uuu1 pupkin2a 505
pupkin10_p10 pupkin10 513
pupkin5a_u1 pupkin5a 508
pupkin3a_qq1 pupkin3a 506
pupkin3a_test22 pupkin3a 506
pupkin3a_12 pupkin3a 506
```

This would specify that db users: pupkinas_us2, pupkinas_u1, pupkinas_u3 belong to user pupkinas with uid (lve id) 502

db user pupkin2a_uuu1 belongs to user pupkin2a with uid 505, etc...

This file is checked for modifications every 5 minutes.

If you need to force reload of that file, run:

```
service db governor restart
```

5.7 Log Files

Error_log

MySQL Governor error log is used to track any problems that MySQL governor might have.

Restrict_log

Restrict log is located at /var/log/dbgovernor-restrict.log

Restrictions:

```
_timestamp_ _username_ LIMIT_ENFORCED _limit_setting_ __current_value_ _restrict_level_
...
SERVER_LO
```

- TRACKED_VALUES_DUMP=busy_time:xx,cpu_time:xx,...
- SERVER_LOAD= load averages followed by output of vmstat
- TRACKED_VALUES_DUMP is available with MEDIUM & LONG format
- SERVER_LOAD is available with LONG format

5.8 Change MySQL version

If you would like to change to a different MySQL version, or switch to MariaDB you have to start by backing up existing databases.

IMPORTANT:

Please make full database backup(including system tables) before you will do upgrade of MySQL or switch to MariaDB. This action will prevent data losing in case if something goes wrong.

```
$ /usr/share/lve/dbgovernor/db-select-mysql --mysql-version=MYSQL_VERSION
$ /usr/share/lve/dbgovernor/mysqlgovernor.py --install
```

* If you are using cPanel or DirectAdmin -- recompile apache

To install beta version of MySQL:

```
$ /usr/share/lve/dbgovernor/mysqlgovernor.py --install-beta
```

MYSQL_VERSION can be one of the following:

auto	default version of MySQL for given OS release (or cPanel settings)
mysql50	MySQL v5.0
mysql51	MySQL v5.1
mysql55	MySQL v5.5
mysql56	MySQL v5.6
mariadb55	MariaDB v5.5
mariadb100	MariaDB v10.0
mariadb101	MariaDB v10.1

* We don't recommend to downgrade from MySQL v5.6, MariaDB 10.x

5.9 Command Line Tools

dbtop -- monitor MySQL usage on per user bases. [More info...](#)

dbctl -- command line tool to manage DB Governor configuration. [More info...](#)

lveinfo --dbgov -- provides historical information about usage and customer restrictions. [More info...](#)

dbgovchar -- generate charts for MySQL usage. [More info...](#)

5.9.1 dbtop

Utility to monitor MySQL usage. Requires db_governor to be running. It shows usage for the current, mid and long intervals.

Options:

-c show one time user list (no interactive mode)
-r interval refresh interval for interactive mode (in seconds)

Control keys:

z toggle color mode and two-color mode
q F10, Ctrl-c - quit program
u sort table by username
c sort table by cpu column
r sort table by read column
w sort table by write column
l sort by restriction level
t sort by time before restrictions will be lifted.

Control keys, that sort table, displays into header of table bold and underlined symbol.

Sorted field will be highlighted by *.

CAUSE field shows current stage, reason for restriction and number of seconds before restriction will be lifted:

Values of column 'CAUSE' - cause of restriction or freezing:

Possible stages: -- OK, 1 - Restriction 1, 2 - Restriction 2, 3 - Restriction 3, 4 -- restriction level 4

c - current (current value of parameter)

s - short (average value of 5 last values of parameter)

m - middle (average value of 15 last values of parameter)

l - long (average value of 30 last values of parameter)

and parameter which is cause of restriction

1/s:busy_time/12 - first level restricted account with short average restriction by busy_time with 12 seconds left before re-enabled.

Display fields:

- cpu - number in %, shows cpu usage by user
- read - number of bytes (kbytes, mbytes, gbytes) which user reads per second
- write - number of bytes (kbytes, mbytes, gbytes) write user reads per second

Color conventions:

Accounts highlighted in **red** color means that the account is restricted.

Accounts highlighted in **blue** color are in cool down period

Command line parameters of dbtop utility:

-r - dbtop refresh period in seconds(dbtop -r12)

5.9.2 dbctl

usage: dbctl command [parameter] [options]

commands:

set	set parameters for a db_governor
list	list users & their limits. It will list all users who had been active since governor
restart,	
	as well as those for who explicit limits were set
list-restricted	list restricted customers, with their limits, restriction reason, and time period they
will still be restricted	
ignore	ignore particular user
watch	start observing particular user again
delete	remove limits for user/use defaults
restrict	restrict user using lowest level (or if --level specified, using the specified level)
unrestrict	unrestrict username (configuration file remains unchanged)
unrestrict-all	unrestrict all restricted users (configuration file remains unchanged)
--help	show this message
--version	version number
--lve-mode	set DB Governor mode of operation. Available values: off abusers all single on
	off - monitor only, don't throttle
	abusers - when user reaches the limit, put user's queries into LVE for that user
(experimental)	
	all - user's queries always run inside LVE for that user (experimental)
	single - single LVE for all abusers.
	on - same as single (deprecated)

parameters:

default set default parameter
usrename set parameter for user

options:

--cpu=N limit CPU (pct) usage
--read=N limit READ (MB/s) usage
--write=N limit WRITE (MB/s) usage
--level=N level (1,2,3 or 4) specified (**deprecated**)
--slow=N limit time (in seconds) for long running SELECT queries

Examples:

```
$ dbctl set test2 --cpu=150,100,70,50 --read=2048,1500,1000,800
```

sets individual limits for cpu(current, short, middle period) and read(current, short, middle, long periods) for user test2

```
$ dbctl set default --cpu=70,60,50,40
```

changes default cpu limits.

All new limits will be applied immediately

To unrestrict user:

```
$ dbctl unrestrict username
```

To unrestrict all users:

```
$ dbctl unrestrict-all
```

To restrict user:

```
$ dbctl restrict dbgov
```

To restrict user to level 2 restriction:

```
$ dbctl restrict dbgov --level=2
```

To make governor to ignore user:

```
$ dbctl ignore username
```

Delete user's limits, and use defaults instead

```
$ dbctl delete username
```

5.9.3 lveinfo --dbgov

lveinfo tool is part of lve-stats package. It was extended to collect historical information about MySQL usage.

```
$ lveinfo --dbgov --help
```

```

Displays information about historical Db Governor usage
Usage: lveinfo [OPTIONS]

-h --help                : this help screen
-v, --version            : version number
-f, --from=              : run report from date and time in YYYY-MM-DD HH:MM format
                        if not present last 10 minutes are assumed
-t, --to=                : run report up to date and time in YYYY-MM-DD HH:MM format
                        if not present, reports results up to now
--period=                : time period
  usage                  : specify minutes with m, h - hours, days with d, and values: today, yesterday,
                        : 5m - last 5 minutes, 4h -- last four hours, 2d - last 2 days, as well as
-o, --order-by=         : orders results by one of the following:
  con                    : average connections
  cpu                    : average CPU usage
  read                   : average READ usage
  write                  : average WRITE usage
-u, --user=              : mysql username
-l, --limit=             : max number of results to display, 10 by default
-c, --csv                : display output in CSV format
-b, --format             : show only specific fields into output
  available values:
  ts                     : timestamp records
  username               : user name
  con                    : average connections
  cpu                    : average CPU usage
  read                   : average READ usage
  write                  : average WRITE usage
  lcpu                   : CPU limit
  lread                  : READ limit
  lwrite                 : WRITE limit
  --show-all            : full output (show all limits); brief output is default

-o, --order-by=         : orders results by one of the following:
  ts                     : timestamp records
  username               : user name
  max_sim_req            : max simultaneous requests
  sum_cpu                : average CPU usage
  sum_write              : average WRITE usage
  sum_read               : average READ usage
  num_of_restrict       : number of restricts
  limit_cpu_end          : limit CPU on period end
  limit_read_end         : limit READ on period end
  limit_write_end       : limit WRITE on period end
--id=                   : LVE id -- will display record only for that LVE id
-u, --user=              : Use username instead of LVE id, and show only record for that user
-l, --limit=             : max number of results to display, 10 by default
-c, --csv                : display output in CSV format
-b, --by-usage           : show LVEs with usage (averaged or max) within 90% percent of the limit
  available values:
  sum_cpu                : average CPU usage
  sum_write              : average WRITE usage
  sum_read               : average READ usage
  num_of_restrict       : number of restricts
  limit_cpu_end          : limit CPU on period end
  limit_read_end         : limit READ on period end
  limit_write_end       : limit WRITE on period end
  --show-all            : full output (show all limits); brief output is default

TS                       : timestamp records
USER                     : user name

```

```

CPU           : average CPU usage
READ          : average READ usage
WRITE         : average WRITE usage
CON           : average connections
lCPU          : CPU limit
lREAD         : READ limit
lWRITE        : WRITE limit
RESTRICT      : C-cpu restrict, R- read restrict, W- write restrict

```

Example:

```

root@cpanell1 [~/ttdttt]# lveinfo --dbgov --user=dbgov --period=1d --limit=10
TS                USER          CPU    READ    WRITE   CON    lCPU    lREAD  lWRITE  RES
2012-12-06 11:14:49  dbgov      9     0.0    0.0    1     90     1000   1000
2012-12-06 11:13:49  dbgov      9     0.0    0.0    1     90     1000   1000
2012-12-06 11:12:49  dbgov      9     0.0    0.0    1     90     1000   1000
2012-12-06 11:11:49  dbgov      9     0.0    0.0    1     90     1000   1000
2012-12-06 11:10:49  dbgov      9     0.0    0.0    1     90     1000   1000
2012-12-06 11:09:49  dbgov     90     0.0    0.0    1     90     1000   1000    C
2012-12-06 11:08:49  dbgov      0     0.0    0.0    0     400    1000   1000
2012-12-06 11:07:49  dbgov      0     0.0    0.0    0     400    1000   1000
2012-12-06 11:06:49  dbgov      0     0.0    0.0    0     400    1000   1000

```

5.9.4 dbgovchart

dbgovchart is analog of lvechart tool to create charts representing customer's to MySQL usage

Usage: /usr/sbin/dbgovchart [OPTIONS]

Acceptable options are:

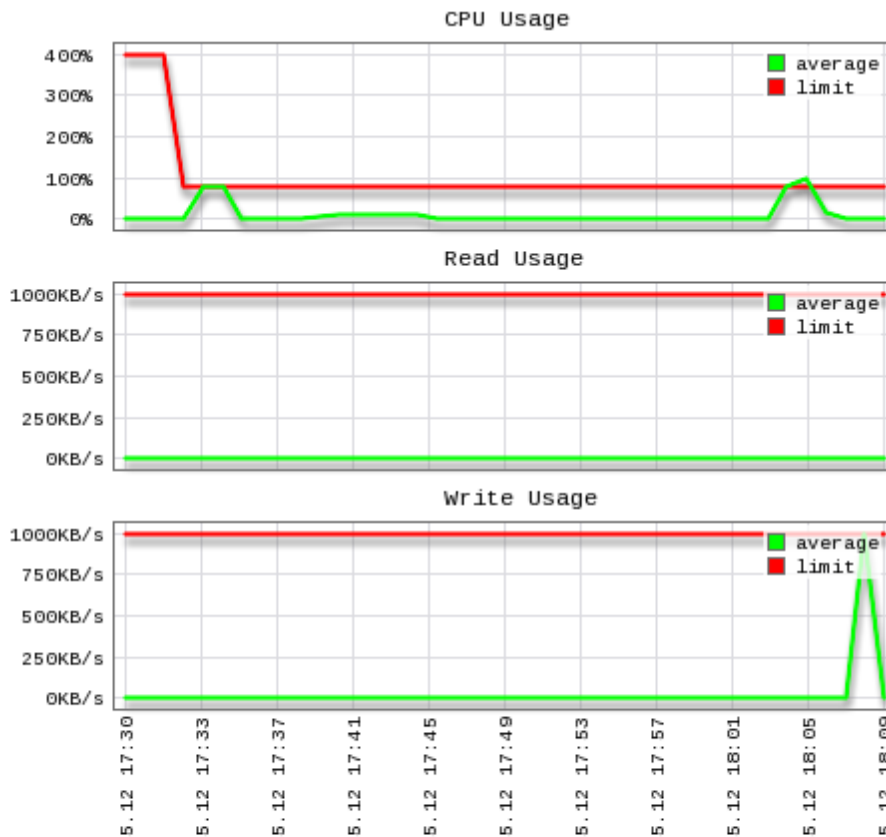
```

--help          This help screen
--version       Version number
--from=         Run report from date and time in YYYY-MM-DD HH:MM format
                 if not present last 10 minutes are assumed
--to=           Run report up to date and time in YYYY-MM-DD HH:MM format
                 if not present, reports results up to now
--period=       Time period
                 specify minutes with m, h - hours, days with d, and values: today, yesterday
                 5m - last 5 minutes, 4h - last four hours, 2d - last 2 days, as well as today
--user=         mysql username
--output=       Filename to save chart as, if not present, output will be sent to STDOUT
--show-all     Show all graphs (by default shows graphs for which limits are set)

```

Charts examples:





5.10 Backing Up MySQL

This operation will take some time. Execute as root:

```
$ mkdir -p ~/mysqlbkp
$ service mysql restart --skip-networking --skip-grant-tables
$ mysql_upgrade
$ mysqldump --all-databases --routines --triggers > ~/mysqlbkp/dbcopy.sql
$ service mysql stop
$ cp -r /var/lib/mysql/mysql ~/mysqlbkp/
$ service mysql start
```

5.11 abrt plugin

We have created a plugin for abrt tool to automatically upload core dumps in case MySQL Governor crashes.

To install the plugin:

```
$ yum install cl-abrt-plugin --enablerepo=cloudlinux-updates-testing
```

It will monitor crash reports for `/usr/sbin/db_governor`, `/usr/sbin/dbtop` and `/usr/sbin/dbctl`

You can modify `/etc/libreport/plugins/dropbox.conf` to monitor other software as well by adding them to `AppList`.

```
AppLists=/usr/sbin/db_governor,/usr/sbin/dbtop,/usr/sbin/dbctl
```

6 PHP Selector

PHP Selector is a CloudLinux component that sits on top of CageFS. It allows each user to select PHP version & module based on their needs. PHP Selector requires account to have CageFS enabled to work.

PHP Selector is compatible with following technologies:
suPHP, mod_fcgid, CGI (suexec), LiteSpeed

It is not compatible with mod_php/DSO, including mod_ruid2 and MPM ITK
PHP Selector is currently not compatible with PHP-FPM, though we are planning to support it in the future.

6.1 Installation

The installation of PHP Selector presumes that you already have [CageFS](#) & [LVE Manager](#) installed.

Installation of different versions of PHP & modules:

```
$ yum groupinstall alt-php
```

Update CageFS & LVE Manager with support for PHP Alternatives

```
$ yum update cagefs lvemanager
```

cPanel/WHM: Make sure 'Select PHP version' is enabled in Feature Manager

IMPORTANT: Please, do not use settings like *SuPHP_ConfigPath*, *PHPRC*, *PHP_INI_SCAN_DIR*. Do not redefine path to php.ini and ini-files for php modules. Doing that can break PHP Selector functionality.

For example, alternative php5.2 versions should load */opt/alt/php52/etc/php.ini* file and scan */opt/alt/php52/etc/php.d* directory for modules:

Configuration File (php.ini) Path	/opt/alt/php52/etc
Loaded Configuration File	/opt/alt/php52/etc/php.ini
Scan this dir for additional .ini files	/opt/alt/php52/etc/php.d
additional .ini files parsed	/opt/alt/php52/etc/php.d/alt_php.ini

Those are default locations for alt-php.

If you need custom PHP settings per user, please change them via "Edit PHP settings" feature of PHP Selector.

6.1.1 LiteSpeed support

To enable PHP Selector with LiteSpeed Web Server follow [PHP Selector installation guide](#), and then adjust following settings in LiteSpeed:

1. CloudLinux (Admin Console --> Configuration --> Server --> General): CageFS
2. Enable SuExec: Server-> General -> PHP SuEXEC -> Yes
3. LSPHP5 external app runs in SUEXEC non-daemon mode ONLY (Run On Start Up --> Yes or No)

4. In lsfhp5 external app (Admin Console --> Configuration --> Server --> External App --> lsfhp5)

Change

```
command => $SERVER_ROOT/cgi-bin/lsphp5
```

To

```
command => /usr/local/bin/lsphp
```

See screen shot below:

The screenshot shows the LiteSpeed WebServer Admin Console interface. The top navigation bar includes Home, Actions, Configuration, Web Console, and Help. A message indicates that configuration changes have been made and a graceful restart is required. The main content area shows the configuration for the 'lsphp5' external app under the 'Server' section. The 'Command' field is highlighted with a red circle.

LiteSpeed API App Definition		Edit	Delete	Back
Name	lsphp5			
Address	uds://tmp/shhttpd/lsphp5.sock			
Notes	Not Set			
Max Connections	35			
Environment	PHP_LSAPI_MAX_REQUESTS=500 PHP_LSAPI_CHILDREN=35			
Initial Request Timeout (secs)	60			
Retry Timeout (secs)	0			
Persistent Connection	Yes			
Connection Keepalive Timeout	Not Set			
Response Buffering	No			
Auto Start	Yes			
Command	/usr/local/bin/lsphp			
Back Log	100			
Instances	1			
suEXEC User	Not Set			
suEXEC Group	Not Set			

* In order to use PHP Selector and custom php.ini, lsfhp5 needs to be in SuEXEC non-daemon mode
 ** Some PHP configurations require more memory for SuExec to work properly. If you are getting error 500 after switching suEXEC to non-daemon mode, try to increase Memory Soft Limit and Memory Hard Limit for external App to at least 650/800M.

6.1.2 ISPmanager support

As of July 2013, PHP Selector support for ISPmanager is limited to command line utilities. You should still be able to use it.

As always, PHP Selector requires CGI, FCGI or suPHP to work.

You will need to do following modifications:

Create new file `/usr/local/bin/php-cgi-etc`


```
#!/bin/bash
/usr/bin/php-cgi -c /etc/php.ini "$@"
```

Make that file executable:

```
$ chmod +x /usr/local/bin/php-cgi-etc
```

Edit file `/usr/local/ispmgr/etc/ispmgr.conf`

Add line:

```
path phpcgibinary /usr/local/bin/php-cgi-etc
```

Make sure there is no other lines with `path phpcgibinary` defined in the file

Restart ISPmanager

```
$ killall ispmgr
```

After that FCGID wrappers (`/var/www/[USER]/data/php-bin/php`) for new users will be like this:

```
#!/usr/local/bin/php-cgi-etc
```

You might need to edit/modify wrappers for existing users if you want them to be able to use PHP Selector. You can leave them as is for users that don't need such functionality.

6.2 Configuration

6.2.1 Setting default version and modules

Administrator can set default interpreter version and extensions all users. All file operations are actually done by CageFS. CageFS takes settings from `/etc/cl.selector/defaults.cfg`. Currently the `/etc/cl.selector/defaults.cfg` is created and handled by CloudLinux PHP Selector scripts. It has the following format:

```
--
[global]
selector=enabled

[versions]
php=5.4

[php5.4]
modules=json,phar

[php5.3]
modules=json,zip,fileinfo
```

6.2.2 Individual PHP.ini files

For each customer, inside CageFS, file `alt_php.ini` is located in `/etc/cl.php.d/alt-phpXX` (XX - version of PHP, like 52 or 53). The file contains PHP extension settings and extension directives selected by customer. This file exists for each customer, for each PHP version. Note, that this is 'local' to CageFS, and different users will have different files. The file is not visible in `/etc/cl.php.d` outside CageFS. If you would like to view that file, use:

```
# cagefsctl -e USERNAME
```

to enter into CageFS for that user. Then type: `exit`; to exit from CageFS

This file has to be updated using `cagefsctl --rebuild-alt-php-ini` after updating `alt-php` RPMs

Admin can change individual settings for PHP extensions by changing that extension's ini file, like editing

```
/opt/alt/php54/etc/php.d.all/eaccelerator.ini
```

and then running

```
cagefsctl --rebuild-alt-php-ini
```

to propagate the change.

6.2.3 Substitute global php.ini for individual customer

Sometimes you might want to have a single customer with a different php.ini, than the rest of your customers.

To do that, you will use [custom.etc directory functionality](#):

1. Move default php.ini into /etc directory and create a symlink to it:

```
$ mv /usr/local/lib/php.ini /etc/php.ini
$ ln -fs /etc/php.ini /usr/local/lib/php.ini
```

2. Change path to *php.ini* in */etc/cl.selector/native.conf* file to:

```
php.ini=/etc/php.ini
```

3. For each user that needs custom *php.ini* file, create directory */etc/cagefs/custom.etc/USER_NAME/php.ini*.

For example if you want to create custom *php.ini* for USER1 and USER2 you would create files:

```
/etc/cagefs/custom.etc/USER1/php.ini
```

```
/etc/cagefs/custom.etc/USER2/php.ini
```

Create such files for each user that should have custom *php.ini* file

4. Execute

```
$ cagefsctl --force-update
```

Notes:

1. Users will be able to override settings of those *php.ini* files (global or custom) via PHP Selector. if you want to prevent that, you should disable PHP Selector feature.
2. Even if PHP Selector is disabled, user can override php settings by using *ini_set()* php function in php script, or by "php -c" command line option.
3. If you modify anything in */etc/cagefs/custom.etc* directory, you should execute

```
$ cagefsctl --update-etc
```

in order to apply changes to CageFS for all users or

```
$ cagefsctl --update-etc user1 user2
```

to apply changes to CageFS for specific users.

6.2.4 Managing interpreter version

Managing interpreter versions is done by means of manipulating a set of symbolic links that point to different versions of interpreter binaries. For example, if default PHP binary is */usr/local/bin/php*:

- First we move the default binary inside CageFS to */usr/share/cagefs-skeleton/usr/selector*, and make */usr/local/bin/php* a symlink pointing to */etc/cl.selector/php*. This operation is done as part of CageFS deployment.
- Next suppose we have additional PHP version, say 5.4.2. The information about all additional interpreter binaries and paths for them is kept in */etc/cl.selector/selector.conf*. This config file is updated by RPM package manager each time alternative PHP package is added, removed or updated
- */usr/bin/cl-selector --list=php* will get us list of all available PHP interpreter versions out of /

`etc/cl.selector/selector.conf` file.

Next we want to know which PHP version is active for a given user (to supply a selected option in options list). We type:

- `/usr/bin/cl-selector --current=php --user=USERNAME` will retrieve PHP version set for a particular user. The script gets the path from `/var/cagefs/USERID/USERNAME/etc/cl.sel/php` symlink, compares it with contents of `/etc/cl.selector/selector.conf` file and if path is valid, prints out the current interpreter version.
- `/usr/bin/cl-selector --select=php --version=5.3 --user=USERNAME` sets the current PHP version for particular user by creating symlink in `/var/cagefs/USERID/USERNAME/etc/cl.selector` directory. All old symlinks are removed, and new symlinks are set.

6.2.5 Including PHP Selector only with some packages (cPanel)

cPanel has a 'Feature Manager' in WHM that allows you to disable PHP Selector for some of the packages that you offer.

In reality it only disables the icon in cPanel interface. Yet, in most cases it should be enough in shared hosting settings.

You can find more info on 'Feature Manager' here: http://docs.cpanel.net/twiki/bin/view/11_30/WHMDocs/FeatureManager

Once PHP Selector is enabled, you can find it in the Feature Manager. Disabling it in Feature Manager, will remove the icon for users that are using that particular 'Feature List'

Home » Packages » Feature Manager

Feature Manager

Working with Feature List: basic

Edit cPAddons Site Software Feature list

- Ability to Change MX
- Addon Domain Manager
- Advanced DNS Zone Editor
- Advanced Guestbook
- Agora Shopping Cart
- Analog Stats
- Apache Handlers Manager

-
- Ruby on Rails
 - SSH Connection Window
 - SSL Host Installer
 - SSL Manager
 - See PHP Configuration
 - Select PHP version
 - Server Status Viewer

6.3 Command Line Tools

<code>/usr/bin/cl-selector</code>	tool is used to select version of PHP interpreter inside CageFS
<code>/usr/bin/piniset</code>	Allows to adjust individual php.ini settings
<code>/usr/bin/alt-php-mysql-reconfigure</code>	Reconfigures alt-php extensions to use correct MySQL library, based on the one installed in the system

6.3.1 selectorctl

selectorctl is a new tool that replaces cl-selector & piniset. It is available starting with CageFS 5.1.3. All new features will be implemented as part of selectorctl.

Common Options:

`--interpreter (-i):` chooses the interpreter to work with. Currently only PHP is supported. If omitted, `--interpreter=php` is implied.

- `--version (-v):` specifies alternatives version to work with.
- `--user (-u):` specifies user to take action upon.
- `--show-native-version (-V):` prints the version of native interpreter

Global Options:

The global options modify settings in `/etc/cl.selector/defaults.cfg` file.

- `--list (-l):` lists all available alternatives for an interpreter. For instance on server with `alt-php` installed it produces the following output. Columns are: short alternative version, full alternative version and path to `php-cgi` binary.

```
$ selectorctl --list
5.2 5.2.17 /opt/alt/php52/usr/bin/php-cgi
5.3 5.3.28 /opt/alt/php53/usr/bin/php-cgi
5.4 5.4.23 /opt/alt/php54/usr/bin/php-cgi
5.5 5.5.7 /opt/alt/php55/usr/bin/php-cgi
```

- `--summary (-S):` prints alternatives state summary. Output format: alternative version, state ('e' for 'enabled', '-' otherwise), chosen as default one or not ('d' for 'default', '-' otherwise). For example:

```
$ selectorctl --summary
5.2 e -
5.3 e -
5.4 e -
5.5 e -
native e d
```

if used with `--show-native-version` displays version for native interpreter

```
$ selectorctl --summary --show-native-version
5.2 e -
5.3 e -
5.4 e -
5.5 e -
native(5.3) e d
```

- `--current (-C):` prints currently globally selected default version (it is stored in `/etc/cl.selector/defaults.cfg` file)

```
$ selectorctl --current
native native /usr/bin/php
```

as well. If used with `--show-native-version`, native interpreter version is displayed

```
--current --show-native-version
native(5.3) native(5.3.19) /usr/bin/php
```

- `--set-current (-B):` sets specified version as globally default one (in `/etc/cl.selector/defaults.cfg` file). For example to set current default version of PHP to 5.4, use:

```
$ selectorctl --set-current=5.4
```

- `--disable-alternative (-N):` adds `state=disabled` option to alternative section. With it a corresponding alternative gets removed from user alternatives selection list. For instance to disable PHP 5.2, run:

```
$ selectorctl --disable-alternative=5.2
```

- `--enable-alternative (-Y):` Enables alternative version, removes `state=disabled` option, if present, from alternative section. For example to enable PHP 5.2:

```
$ selectorctl --enable-alternative=5.2
```

`--enable-extensions (-E)`: enables extensions for particular PHP version by adds comma-separated list of extensions of modules for alternative in `/etc/cl.selector/defaults.cfg`. Requires `--version` option. For example:

```
$ selectorctl --enable-extensions=pdo,phar --version=5.2
```

`--disable-extensions (-D)`: removes extensions for a particular PHP version. Comma-separated list of extensions will be removed from `/etc/cl.selector/defaults.cfg`. Requires `--version`. Example:

```
$ selectorctl --disable-extensions=pdo,phar --version=5.2
```

`--replace-extensions (-R)`: replaces all extensions for particular PHP version to the list of comma separated extensions. Requires `--version` option. Example:

```
$ selectorctl --replace-extensions=pdo,phar --version=5.2
```

`--list-extensions (-G)`: lists extensions for an alternative for a particular version. Requires `--version`. Example:

```
$ selectorctl --list-extensions --version=5.3
~ xml
- xmlreader
- xmlrpc
- xmlwriter
- xrange
+ xsl
```

Plus sign stands for 'enabled', minus – for 'disabled', tilde (~) means compiled into interpreter. Enabled and disabled state relates to presence in `/etc/cl.selector/defaults.cfg` file.

End User Options

All end-user settings are contained in individual user's `alt_php.ini` files and controlled using `selectorctl` command.

`--user-summary (-s)`: prints user alternatives state summary. Example:

```
$ selectorctl --summary --user=user1
5.2 e - -
5.3 e - -
5.4 e - -
5.5 e - -
native e d s
```

Columns are: alternative version, state ('e' for 'enabled', '-' otherwise), chosen as default one or not ('d' for 'default', '-' otherwise), selected as user default one or not ('s' for 'selected', '-' otherwise). If used with `--show-native-version`, version for native interpreter is shown in parenthesis:

```
$ selectorctl --summary --user=user1 --show-native-version
5.2 e - -
5.3 e - -
5.4 e - -
5.5 e - -
native(5.3) e d s
```

`--user` option is required.

`-current (-c)` prints currently globally selected default version (in `/etc/cl.selector/defaults.cfg` file)

```
$ selectorctl --current
5.3 5.3.28 /opt/alt/php53/usr/bin/php-cgi
```

If used with `--show-native-version` to display native version

```
$ selectorctl --user-current --user=user1
5.3 5.3.28 /opt/alt/php53/usr/bin/php-cgi
```

--user option is required.

--set-user-current (-b)

sets specified version as the one to use for this end user

```
$ selectorctl --set-user-current=5.4 --user=user1
```

changes user symlinks for the PHP interpreter to point to alternative

5.4.

--user option is required.

--enable-user-extensions (-e)

Enables comma-separated list of extensions for the user user.

Information is saved to alt_php.ini file. Requires --version and --user options.

```
$ selectorctl --enable-user-extensions=pdo,phar --version=5.2 --user=u
```

--disable-user-extensions (-d)
version and --user options.

Disables extensions provided as comma-separated list. Requires --

```
$ selectorctl --disable-user-extensions=pdo,phar --version=5.2 --user=
```

--replace-user-extensions (-r)

Replaces extensions with a provided comma-separated list of

extensions Requires --version and --user options:

```
$ selectorctl --replace-user-extensions=pdo,phar --version=5.2 --user=
```

--reset-user-extensions (-t)

Resets extensions for end user to default list of extensions as defined

in default.cfg. Requires --version and --user options.

```
$ selectorctl --reset-user-extensions --version=5.2 --user=user1
```

--list-user-extensions (-g)
--user options.

lists enabled user extensions for an alternative. Requires --version and

```
$ selectorctl --list-user-extensions --version=5.3 --user=user1
xml
xmlreader
xmlrpc
```

if --all option present, command will list all alternatives extensions

marked enabled or disabled for given user. For example:

```
$ selectorctl --list-user-extensions --version=5.3 --user=user1 --all
- xmlreader
- xmlrpc
- xmlwriter
- xrange
+ xsl
```

Plus sign stands for 'enabled', minus – for 'disabled'. Enabled and disabled state relates to presence or absence of corresponding extensions in user alt_php.ini file.

--add-options (-k)

adds options (as in php.ini) to user alt_php.ini file. For example:

```
$ selectorctl --add-options=log_errors:on,display_errors:on --version=
```

alt_php.ini file overwriting default values for a user. Requires --version and --user options.

--replace-options (-m)

replaces all options in user alt_php.ini file with specified ones. Requires

--version and --user options.

```
$ selectorctl --replace-options=log_errors:on,display_errors:on --vers
```

--delete-options (-x)
and --user options.

removes custom options from user alt_php.ini file. Requires --version

```
$ selectorctl --delete-options=log_errors,display_errors --version=5.2
```

--print-options (-P)

print options from /etc/cl.selector/php.conf file with default values or

ones overwritten in user's alt_php.ini file

```
$ selectorctl --print-options --version=5.2 --user=user1
TITLE:allow_url_fopen
DEFAULT:On
COMMENT:Allows PHP file functions to retrieve data from remote
locations over FTP or HTTP. This option is a great security risk,
thus do not turn it on without necessity.
TYPE:bool
...
```

Requires `--version` and `--user` options. By default outputs as plain test. If `--json`, `--csv`, `--perl` is specified, outputs data in corresponding format. For example, with `--perl` option, the output is perl hash structure that can be evaluated.

`--reset-options (-z)` removes custom options from alt_php.ini files for ALL users and versions. Backup files in home folders are cleared.

```
$ selectorctl --reset-options
```

The ranges of affected customers or versions can be narrowed with `--version` or `--user` options:

```
$ selectorctl --reset-options --user=user1,user2 --version=5.3,5.4
```

`--list-users (-L)` list users that use particular version of interpreter, specified with `--version` option. For example, to see all users that use PHP version 5.3

```
$ selectorctl --list-users --version=5.3
```

`--change-to-version (-T)` changes all (or particular user) from one interpreter version to another

```
$ selectorctl --change-to-version=5.2 --version=5.3
```

Additional Options:

`--base64 (-Q)` Sometimes PHP options values can contain commas and other symbols that break command line formatting. In such a case convert a key:value pair into base64 and pass it as value for option-related arguments. For example, to add `disable_functions=exec,popen,system` and `display_errors=on` to user options, do the following:

```
$ selectorctl --add-options=`echo disable_functions:exec,popen,system`
```

Option `-w 0` of base64 executable stands for 'disable wrapping of lines'.

Without it base64 output will break the command.

`--quiet` makes selectorctl continue when it encounter option not found in php.conf. Without it selectorctl exits with error.

6.3.2 cl-selector

`/usr/sbin/cl-selector` - tool is used to select version of PHP interpreter inside CageFS

```
-l | --list           : List available alternatives for item specified
-L | --list-extensions : List available extensions for a user
-e | --enable       : Enable an extension for a user
-i | --interpreter   : Specify an interpreter for an extension (e.g.
php)
-d | --disable      : Disable an extension for a user
-a | --all          : Show available extensions for a user
-c | --current      : Print alternative currently in use for a user
-u | --user         : Specify a user
-v | --version      : Specify a version for an alternative
-s | --select       : Select an alternative to be used
-p | --prove        : Print if CageFS enabled for a given user
-r | --reload       : Reload specified processes for a given user
-b | --backup       : backup linkage configuration for a user
-q | --quiet        : Suppress error messages
```


-h | --help : Print this message

It was meant to be used by control panels (like cPanel) to configure interpreter for a particular customer. It can:

- Managing interpreter versions.
- Managing interpreter versions extensions (modules).

`/usr/bin/cagefsctl` -- has a set of extensions to administer PHP Selector

`--cl-selector-reset-versions [u1 u2 ...]` : switches all or specified users to default

version of PHP

`--cl-selector-reset-modules [u1 u2 ...]` : switches all or specified users to default

set of PHP extensions

`--rebuild-alt-php-ini [u1 u2 ...]` : rebuilds `alt_php.ini` for all specified (or all) users

6.3.3 piniset - php.ini options

`/usr/bin/piniset` Allows to adjust individual `php.ini` settings

Customer's can adjust individual `php.ini` settings using user interface.

Admin can manage the settings by running:

```
/usr/bin/piniset --replace="session.save_path:/tmp/php,upload_max_size:1G" --version=5.3 --user=USERNAME
```

`/usr/bin/piniset` -- is used to manage PHP ini settings

Usage: `/usr/bin/piniset [OPTIONS] --version=PHPVERSION --user=USERNAME`

Options:

-r | --replace : replaces user `php.ini` options with supplied comma-separated

list of colon-separated key:value pairs

--append : appends options to user's `php.ini` with supplied comma-separated

list of colon-separated key:value pairs [lvermanager

0.7-1.29]

--delete : deletes options from user `php.ini` file, accepts comma-separated

list of options. [lvermanager 0.7-1.29]

-j | --json : return data as JSON

-p | --perl : return data as perl "hash"

-u | --user : specify the user

-v | --version : specify php version

-h | --help : print this message

```
/usr/bin/piniset --json --version=5.3 --user=USERNAME
```

Will display custom `php.ini` options for end customer

6.3.4 Integrating With Control Panels

This is a list of commands that we are using to integrate PHP Selector with control panels. If you need to integrate with custom control panel, you might find all the commands here:

PHP summary:

`/usr/bin/cl-selector --summary=php`

Result:

```
5.1 - -
5.2 e -
5.3 e d
5.4 e -
5.5 e -
4.4 e -
native - -
```

First column: php version

Second: enabled or not (e -- enabled)

Third: if selected as default (d -- default)

Set default version:

```
/usr/bin/sudo /usr/bin/cl-selector --interpreter=php --version=_VERSION_
```

Disable version

```
/usr/bin/cl-selector --interpreter=php --set-disabled=_VERSION_
```

Enable version

```
/usr/bin/cl-selector --interpreter=php --set-enabled=_VERSION_
```

List Extensions for a version:

```
/usr/bin/cl-selector --list-extensions=php --version=5.2
```

Result:

```
- apc
- bcmath
- big_int
- bitset
- bloomy
~ bz2
- bz2_filter
~ calendar
- coin_acceptor
- crack
~ ctype
+ curl
```

+ -- enabled

~ -- included in php binary (cannot be disabled)

-- disabled

Update Default Extensions:

```
/usr/bin/cl-selector --interpreter=php --version=_VERSION_ --update-default-extensions=EXT_LIST
```

EXT_LIST is comma separated list of PHP extensions to be enabled by default for this version

Select PHP version for a user:

```
/usr/bin/cl-selector --select=php --version=_VERSION_ --user=_USER_
```

List Enabled extensions for a user:

```
cl-selector --list-extensions=php --version=_VERSION_ --user=_USER_ --all
```

Reset user's extensions to defaults:

```
/usr/bin/cl-selector --set-default-extensions=php --user=_USER_ --version=_VERSION_
```

Update extensions for the user:

```
/usr/bin/cl-selector --interpreter=php --update=EXT_LIST --user=_USER_ --version=_VERSION_ --with-reload --backup
```

EXT_LIST is comma separated list of PHP extensions to be enabled by default for this version

List available options for php.ini editing:

```
/usr/bin/piniset --version=_VERSION_ --user=_USER_ [--json]
```

Set php.ini options for end user:

```
/usr/bin/piniset --version=_VERSION_ --user=_USER_ --replace=OPTIONS --base64
```

Here is an example how you can generate OPTIONS in base64 format:

```
OPTIONS=`echo disable_functions:exec,syslog|base64 -w 0`,`echo display_errors:off|base64 -w 0`,`echo post_max_size:128M|base64 -w 0`  
echo $OPTIONS
```

6.4 Removing PHP Selector

Once alternative versions of PHP are removed, PHP Selector will be disabled. To do that:

```
$ yum groupremove alt-php
```

6.5 Using PHP Selector

Once PHP Selector is installed you will see "Selector" tab in LVE Manager

The screenshot shows the LVE Manager interface with the 'Selector' tab active. The interface displays the status 'Selector is: enabled' and 'Default PHP version is: native'. A dropdown menu shows 'Choose default modules for: 5.4'. A grid of PHP modules is listed with checkboxes for enabling or disabling them. At the bottom, there are 'Save', 'Reset', and 'Default' buttons. Annotations with arrows point to various elements: 'Enable PHP Selector in end user interface' points to the 'enabled' status; 'List of modules enabled by default' points to the 'zip' checkbox; 'Save modules that will be enabled by default' points to the 'Save' button; 'Reset module selection' points to the 'Reset' button; 'Match modules as in cPanel's PHP' points to the 'Default' button; and 'select modules for particular php version' points to the '5.4' dropdown.

PHP Selector lets you select default PHP version, as well as modules that will be available to user out of the box.

Inside cPanel, User will be able to change PHP version they would have



User can select PHP version

as well as extensions that they want to use:

Change Version *Current Version* *Change php.ini settings*

Current PHP version: **5.2**

PHP Version: **5.2** | Set as current | Show PHP Settings

apc	<input type="checkbox"/>	gender	<input type="checkbox"/>	mcrypt	<input type="checkbox"/>	posix	<input checked="" type="checkbox"/>	tidy	<input type="checkbox"/>
bcmath	<input checked="" type="checkbox"/>	geoip	<input type="checkbox"/>	memcache	<input type="checkbox"/>	pspell	<input type="checkbox"/>	timezonedb	<input type="checkbox"/>
big_int	<input type="checkbox"/>	hidef	<input type="checkbox"/>	memcached	<input type="checkbox"/>	quickhash	<input type="checkbox"/>	translit	<input type="checkbox"/>
bitset	<input type="checkbox"/>	htscanner	<input type="checkbox"/>	mongo	<input type="checkbox"/>	radius	<input type="checkbox"/>	uploadprogress	<input type="checkbox"/>
bloomy	<input type="checkbox"/>	huffman	<input type="checkbox"/>	msgpack	<input type="checkbox"/>	recode	<input type="checkbox"/>	uuid	<input type="checkbox"/>
bz2_filter	<input type="checkbox"/>	idn	<input type="checkbox"/>	mssql	<input type="checkbox"/>	rsync	<input type="checkbox"/>	wddx	<input type="checkbox"/>
coin_acceptor	<input type="checkbox"/>	igbinary	<input type="checkbox"/>	mysql	<input checked="" type="checkbox"/>	snmp	<input type="checkbox"/>	xcache	<input type="checkbox"/>
crack	<input type="checkbox"/>	imagick	<input type="checkbox"/>	mysqli	<input type="checkbox"/>	soap	<input type="checkbox"/>	xdebug	<input type="checkbox"/>
curl	<input checked="" type="checkbox"/>	imap	<input checked="" type="checkbox"/>	ncurses	<input type="checkbox"/>	sourceguardian	<input type="checkbox"/>	xmlreader	<input checked="" type="checkbox"/>
dba	<input type="checkbox"/>	includ	<input type="checkbox"/>	oauth	<input type="checkbox"/>	spl_types	<input type="checkbox"/>	xmlrpc	<input type="checkbox"/>
dbase	<input type="checkbox"/>	inotify	<input type="checkbox"/>	odbc	<input type="checkbox"/>	ssh2	<input type="checkbox"/>	xmlwriter	<input checked="" type="checkbox"/>
dbx	<input type="checkbox"/>	intl	<input type="checkbox"/>	pdo	<input type="checkbox"/>	stats	<input type="checkbox"/>	xrange	<input type="checkbox"/>
dom	<input checked="" type="checkbox"/>	ioncube_loader	<input type="checkbox"/>	pdo_mysql	<input type="checkbox"/>	stem	<input type="checkbox"/>	xsl	<input type="checkbox"/>
doublemetaphone	<input type="checkbox"/>	json	<input checked="" type="checkbox"/>	pdo_odbc	<input type="checkbox"/>	stomp	<input type="checkbox"/>	yaf	<input type="checkbox"/>
eaccelerator	<input type="checkbox"/>	ldap	<input type="checkbox"/>	pdo_pgsql	<input type="checkbox"/>	suhosin	<input type="checkbox"/>	zend_optimizer	<input type="checkbox"/>
enchant	<input type="checkbox"/>	lzf	<input type="checkbox"/>	pdo_sqlite	<input type="checkbox"/>	sysvmsg	<input type="checkbox"/>	zip	<input type="checkbox"/>
fileinfo	<input type="checkbox"/>	mailparse	<input type="checkbox"/>	pgsql	<input type="checkbox"/>	sysvsem	<input type="checkbox"/>		
gd	<input type="checkbox"/>	mbstring	<input type="checkbox"/>	phar	<input checked="" type="checkbox"/>	sysvshm	<input type="checkbox"/>		

Save | Use Defaults

Save selected extensions *Use default set of extensions* *Enable PHP extensions*

and php.ini settings

Current PHP version: **5.2**

PHP Version

allow_url_fopen	On	<i>Switch back to extensions</i>
display_errors	Off	
error_reporting	E_ALL & -E_NOTICE	
file_uploads	On	
include_path	./usr/share/pear:/opt/alt/php52/usr/share/php	
log_errors	On	
magic_quotes_gpc	Off	
mail.force_extra_parameters	no value	
max_execution_time	0	<i>Change PHP Options</i>
max_input_time	-1	
memory_limit	512M	
open_basedir	no value	
post_max_size	<input type="text" value="2M"/> <input type="button" value="Apply"/>	
register_globals		
safe_mode		
safe_mode_exec_dir		
safe_mode_include_dir		
session.save_path	/tmp	
short_open_tag	On	
upload_max_filesize	2M	

Save php.ini options

6.6 Custom PHP.ini options

[Requires LVE Manager 0.6+]

PHP Selector allows customer to edit php.ini settings. Admin has a full control over which settings can be modified.

To allow settings to be modifiable, it has to be whitelisted in:

/etc/cl.selector/php.conf

Here are some of the examples of allowed directives:

```
Directive = safe_mode
Default   = Off
Type      = bool
Remark    = <5.4.0
Comment   = Enables PHP safe mode. This mode puts a number of restrictions on scripts (say, access
```

```

Directive = safe_mode_include_dir
Type      = value
Remark    = <5.4.0
Comment   = If PHP is in the safe mode and a script tries to access some files, files from this di
e directory must also be in include path. For example: /dir/inc

```

Directive	php.ini setting
Default	Default value
Type	bool, value (any text), list
Range	list of values for list Type
Comment	explanation of the setting to display in UI

Default values, that are shown in PHP-Selector web interface, are taken from '/opt/alt/phpXX/usr/bin/php -i' runtime values, if directive is not there, it will use 'default' value that was set in php.conf . So, if you wish to change default value of any option for "alternative" php version, please modify /opt/alt/phpXX/etc/php.ini files (where XX = 55, 54, 53, etc according to php version).

Admin can modify the settings using [piniset](#) command.
Users can use web interface to modify php.ini settings

Current PHP version: **5.2**

PHP Version

allow_url_fopen	On	<i>Switch back to extensions</i>
display_errors	Off	
error_reporting	E_ALL & -E_NOTICE	
file_uploads	On	
include_path	./usr/share/pear:/opt/alt/php52/usr/share/php	
log_errors	On	
magic_quotes_gpc	Off	
mail.force_extra_parameters	no value	
max_execution_time	0	<i>Change PHP Options</i>
max_input_time	-1	
memory_limit	512M	
open_basedir	no value	
post_max_size	2M <input type="button" value="Apply"/>	
register_globals		
safe_mode		
safe_mode_exec_dir		
safe_mode_include_dir		
session.save_path	/tmp	
short_open_tag	On	
upload_max_filesize	2M	

Save php.ini options

6.7 End user directories

Following files and directories are created inside CageFS for each customer

/etc/cl.selector -> php binaries symbolic links

/usr/selector/php -> native PHP binaries

/etc/cl.php.d/alt-php* --> links to enabled modules.

/home/user/.cl.selector/alt_phpXX.cfg -- config file for custom php options

like:

/etc/cl.php.d/alt-php54/fileinfo.ini -> /opt/alt/php54/etc/php.d.all/fileinfo.ini

6.8 Compiling your own extensions

Sometimes you might want to compile your own PHP extension for your users to use. In most cases it is better to contact our support at <https://helpdesk.cloudlinux.com>. We will try to provide such extension

for you via regular updates within 5-7 days.

If you have decided that you want to build it on your own, you would need to build it for each and every supported version of PHP that you have installed. The module installation process is a bit different from standard - you would need to use the version of phpize and php-config binaries that comes with particular alt-php version.

The full process for php 5.X would be:

1. download and unpack extension, cd into it's directory

2. execute our version of phpize for necessary:

```
/opt/alt/php5X/usr/bin/phpize
```

3. execute configure with our binary:

```
./configure --with-php-config=/opt/alt/php5X/usr/bin/php-config
```

4. make the .so file:

```
make
```

5. copy it to modules directory (on 32bit server, use usr/lib/php/modules)

```
cp -rp modules/*.so /opt/alt/php5X/usr/lib64/php/modules/
```

6. add ini file for module to /opt/alt/php5X/etc/php.d.all

7. register new alt-php version with:

```
$ cagefsctl --setup-cl-selector
```

6.9 Roll your own PHP

To add your own PHP version in PHP Selector

- create directory in /opt/alt (like: /opt/alt/php51), and mimic directory structure inside to be similar to the one for PHP versions bundled by CloudLinux.
- Put all the ini files for all the modules into /opt/alt/php51/etc/php.d.all
- Create symbolic link /opt/alt/php51/etc/php.d -> /etc/cl.php.d/alt-php51

Place all so files into /opt/alt/php51/usr/lib/php/modules

Add absolute path to PHP binaries into /etc/cl.selector/selector.conf using following format:

```
php      5.1 5.1.2 /opt/alt/php51/usr/bin/php-cgi
php-cli  5.1 5.1.2 /opt/alt/php51/usr/bin/php
php-fpm  5.1 5.1.2 /opt/alt/php51/usr/sbin/php-fpm
  ^      ^      ^                ^----- absolute path
  |      |      |----- real version
  |      | ----- version to display
  |----- binary to 'substitute'
```

Execute cagefsctl --setup-cl-selector

New version of PHP should be available now for selection in PHP Selector.

6.10 Detect User's PHP Version

[requires lve-manager 0.5-63 or higher]

PHP Selector provides an easy way to figure out which versions are available and selected for end user from command line. You can get this information by running:

```
$ /usr/bin/cl-selector --summary php --user USERNAME
```

The output:


```

5.2 e - -
5.3 e - s
5.4 e - -
5.5 e - -
native e d -

```

The first column defines the PHP version. Native -- means native PHP version, like the one installed by cPanel with EasyApache.

Second column will contain either **e** or -. If **e** is present, it means that given version is enabled, and can be selected by end user.

Third column can have values **d** or -. If **d** is present, that version is considered a 'default' version. Only one PHP version will have **d** indicator

Fourth column can have values **s** or -. If **s** is present, that is the selected version, currently being used by end user. Only one PHP version will have **s** indicator.

In case user is not inside CageFS, and as such doesn't use PHP Selector, you will see following error message:

```
ERROR: User is not in cagefs
```

6.11 Bundled PHP Extensions

Large number of PHP extensions are bundled with each version of PHP:

- [PHP 4.4](#)
- [PHP 5.1](#)
- [PHP 5.2](#)
- [PHP 5.3](#)
- [PHP 5.4](#)
- [PHP 5.5](#)

6.11.1 PHP 4.4 Extensions

bcmath	fileinfo	mbstring	posix	sysvsem
bz2	ftp	mcrypt	pspell	sysvshm
calendar	gd	mhash	recode	tokenizer
ctype	gettext	mysql	session	wddx
curl	gmp	ncurses	shmop	xml
dba	iconv	odbc	snmp	xmlrpc
dbase	imap	overload	sockets	zlib
dbx	ioncube_loader	pcntl	sourceguardian	
domxml	json	pcre	standard	
exif	ldap	pgsql	sysvmsg	

6.11.2 PHP 5.1 Extensions

bcmath	gd	lzf	pgsql	stem
big_int	geoip	mbstring	posix	sysvmsg
bitset	gettext	mcrypt	pspell	sysvsem

bz2 bz2_filter calendar coin_acceptor crack ctype curl date dba dbase dom doublemetaphone exif ftp	gmp haru hash huffman iconv idn igbinary imagick imap included inotify ioncube_loader ldap libxml	memcache msgpack mysql mysqli ncurses odbc openssl pcntl pcre pdo pdo_mysql pdo_odbc pdo_pgsql pdo_sqlite	quickhash radius reflection session shmap simplexml snmp soap sockets sourceguardian spl ssh2 standard stats	sysvshm tidy tokenizer translit wddx xdebug xml xmlreader xmlrpc xmlwriter xsl zlib
---	--	--	---	--

6.11.3 PHP 5.2 Extensions

apc bcmath big_int bitset bloomy bz2 bz2_filter calendar coin_acceptor crack ctype curl date dba dbase dbx dom doublemetaphone eaccelerator enchant exif ffmpeg fileinfo filter	ftp gd gender geoip gettext gmp haru hash hidef htscanner huffman iconv idn igbinary imagick imap included inotify intl ioncube_loader json ldap libxml lzf	magickwand mailparse mbstring mcrypt memcache memcached mhash mongo msgpack mssql mysql mysqli ncurses oauth odbc opcache openssl pcntl pcre pdo pdo_mysql pdo_odbc pdo_pgsql pdo_sqlite	pgsql phar posix pspell quickhash radius recode reflection rsync session shmap simplexml snmp soap sockets sourceguardian spl spl_types sqlite ssh2 standard stats stem stomp	suhosin sysvmsg sysvsem sysvshm tidy timezonedb tokenizer translit uploadprogress uri_template uuid wddx xcache xcache_3 xdebug xml xmlreader xmlrpc xmlwriter xrange xsl yaf zend_optimizer zip zlib
--	--	---	--	---

6.11.4 PHP 5.3 Extensions

apc bcmath big_int	filter ftp functional	magickwand mailparse mbstring	posix pspell quickhash	sysvshm tidy timezonedb
--------------------------	-----------------------------	-------------------------------------	------------------------------	-------------------------------

bitset	gd	mcrypt	radius	tokenizer
bloomy	gender	memcache	recode	trader
bz2	geoip	memcached	reflection	translit
bz2_filter	gettext	mhash	rsync	uploadprogress
calendar	gmp	mongo	session	uri_template
coin_acceptor	haru	msgpack	shmop	uuid
core	hash	mssql	simplexml	wddx
crack	hidef	mysql	snmp	weakref
ctype	htscanner	mysqli	soap	xcache
curl	huffman	ncurses	sockets	xcache_3
date	iconv	oauth	sourceguardian	xdebug
dba	idn	odbc	spl	xml
dbase	igbinary	opcache	spl_types	xmlreader
dbx	imagick	openssl	sqlite	xmlrpc
dom	imap	pcntl	sqlite3	xmlwriter
doublemetaphone	included	pcre	ssh2	xrange
eaccelerator	inotify	pdo	standard	xsl
eio	intl	pdo_mysql	stats	yaf
enchant	ioncube_loader	pdo_odbc	stem	zend_guard_loader
ereg	json	pdo_pgsql	stomp	zip
exif	ldap	pdo_sqlite	suhosin	zlib
ffmpeg	libxml	pgsql	sysvmsg	
fileinfo	lzf	phar	sysvsem	

6.11.5 PHP 5.4 Extensions

apc	functional	mcrypt	quickhash	timezonedb
bcmath	gd	memcache	radius	tokenizer
big_int	gender	memcached	recode	trader
bitset	geoip	mhash	reflection	translit
bz2	gettext	mongo	rsync	uploadprogress
bz2_filter	gmp	msgpack	session	uri_template
calendar	haru	mssql	shmop	uuid
core	hash	mysql	simplexml	wddx
ctype	hidef	mysqli	snmp	weakref
curl	htscanner	ncurses	soap	xcache
date	iconv	oauth	sockets	xcache_3
dba	igbinary	odbc	sourceguardian	xdebug
dbase	imagick	opcache	spl	xml
dbx	imap	openssl	spl_types	xmlreader
dom	included	pcntl	sqlite3	xmlrpc
doublemetaphone	inotify	pcre	ssh2	xmlwriter
eaccelerator	intl	pdo	standard	xrange
eio	ioncube_loader	pdo_mysql	stats	xsl
enchant	json	pdo_odbc	stem	yaf
ereg	ldap	pdo_pgsql	stomp	zend_guard_loader
exif	libxml	pdo_sqlite	suhosin	zip
ffmpeg	lzf	pgsql	sysvmsg	zlib
fileinfo	magickwand	phar	sysvsem	

filter ftp	mailparse mbstring	posix pspell	sysvshm tidy	
---------------	-----------------------	-----------------	-----------------	--

6.11.6 PHP 5.5 Extensions

apcu bcmath big_int bitset bz2 bz2_filter calendar core ctype curl date dba dbase dbx dom doublemetaphone eio enchant ereg exif ffmpeg fileinfo filter	ftp gd gender geoip gettext gmp haru hash hidef htscanner iconv igbinary imagick imap inotify intl json ldap libxml lzf magickwand mailparse mbstring	mcrypt memcache memcached mhash mongo msgpack mssql mysql mysqli ncurses oauth odbc opcache openssl pcntl pcre pdo pdo_mysql pdo_odbc pdo_pgsql pdo_sqlite pgsql phar	posix pspell quickhash radius recode reflection rsync session shmop simplexml snmp soap sockets spl spl_types sqlite3 ssh2 standard stats stem stomp sysvmsg sysvsem	sysvshm tidy timezonedb tokenizer trader translit uploadprogress uri_template uuid wddx weakref xcache_3 xdebug xml xmlreader xmlrpc xmlwriter xrange xsl yaf zip zlib
--	---	---	--	---

6.12 Disabling PHP extensions

If you want to disable PHP extension globally, you don't need to remove file `/opt/alt/phpXX/etc/php.d.all/$EXTENSION.ini`. You should just comment out "extension=" directives in it.

The extension will be visible in PHP Selector interface, but selecting it in users's interface will take no effect - extension will be disabled in fact.

Reinstalling of alt-php packages will not reset settings (will not enable extension again).

7 Python and Ruby Selector

We have the ability to deploy Python and Ruby applications via application server. Python and Ruby Selector uses mod_passenger to host Python and Ruby.

This feature is available for CloudLinux 6 or later and requires LVE Manager 0.9-1 or later.

7.1 Installation

To install Python and Ruby Selector run:

```
yum install lve-manager alt-python-virtualenv alt-mod-passenger
```

To use Python Selector you should install alternative Python packages:

```
yum groupinstall alt-python
```

To use Ruby Selector install alternative Ruby packages:

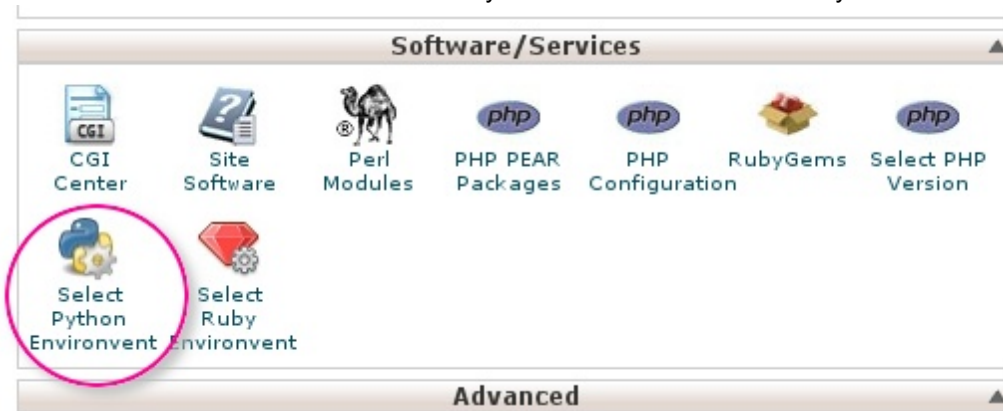
```
yum groupinstall alt-ruby
```

To use MySQL database you should install alt-python27-devel package:

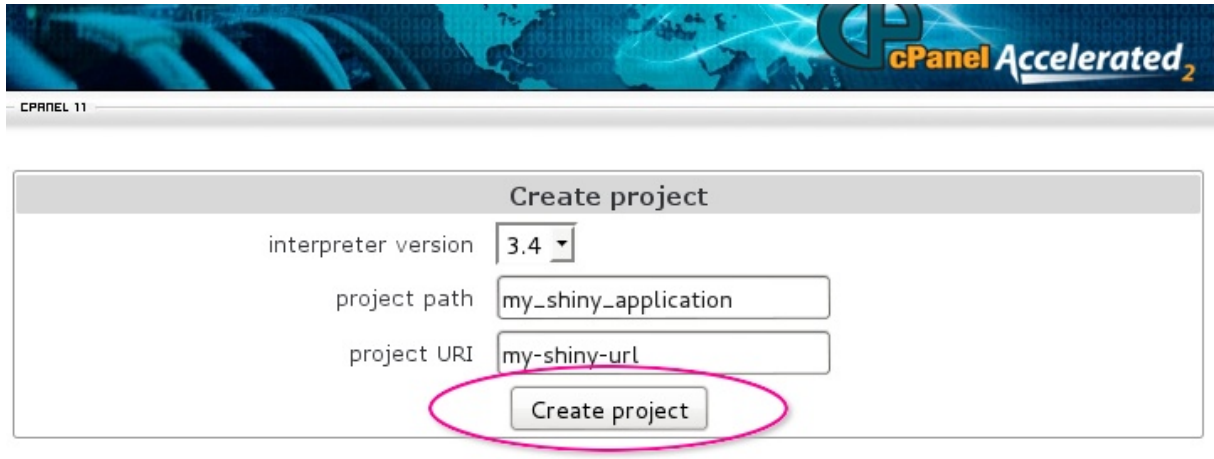
```
yum install alt-python27-devel
```

7.2 End User Access

1. In Software/Services area choose Select Python Environment/Select Ruby Environment.



2. Create project form will appear. Choose interpreter version for your application, application folder name (project path) and url for accessing your application (project URL). Click "Create project" to create an application.



CPANEL 11

Create project

interpreter version

project path

project URI

After a little while a new application entry will be appended to the web-page.

path	my_shiny_application	Edit	<input type="button" value="Remove"/> <input type="button" value="Update"/> <input type="button" value="Reset"/>
uri	clman1.com/my-shiny-url	Edit	
wsgi		Edit	
version	<input type="text" value="3.4"/>		
modules	show		

Home ■ Trademarks ■ Help ■ Documentation ■ Contact ■ Logout

3. You can edit path (folder name in homedir, for example /home/clman1), uri for application, wsgi handler. If you click Edit - the value is converted to input field and thus becomes editable. When editing is complete, click Save.

path	my_shiny_application	Edit
uri	<input type="text" value="clman1.com/my-shiny-url-edited"/>	<input type="button" value="Save"/>
wsgi		Edit

path	my_shiny_application	Edit
uri	clman1.com/my-shiny-url-edited	Edit
wsgi		Edit
version	<input type="text" value="3.4"/>	
modules	show	

4. Wsgi entry is to specify python wsgi application entry point. It must be specified as filename, must be callable and separated by colon. If your app is running from file flask/run.py by calling callable app, set flask/run.py:app.

path	my_shiny_application
uri	clman1.com/my-shiny-url-edited
wsgi	
version	3.4 ▾
modules	<input type="text" value="Flask,sqlalc"/> Add SQLAlchemy SQLAlchemy-AutoSlug sqlalchemy-batteries sqlalchemy-bitcoin sqlalchemy-citext SQLAlchemy-Continuum SQLAlchemy-Dao

4. When Show control is clicked, python extensions section will be expanded. It gives the ability to add or remove python modules. When start typing in input field, appropriate hints are shown in drop-down list. Choose the entry you want from drop-down and click Add.

path	my_shiny_application	Edit
uri	clman1.com/my-shiny-url-edited	Edit
wsgi		Edit
version	3.4 ▾	
modules	<input type="text"/> Add pip 1.5.6 delete setuptools 3.6 delete Flask - delete SQLAlchemy - delete hide	

If you click Delete, the corresponding module entry will disappear.

In addition to setting path, uri and wsgi, the interpreter version can be changed as well by changing the value in select drop-down.

5. No changes are applied to application environment until Update button is clicked. Before the Update button is clicked, all changes can be reverted with Reset button.

The newly created application will be supplied with stub only. A real application ought to be put into application folder. After application is placed into application folder, the wsgi parameter can be set.

Click Remove to delete the application - the application folder itself will remain untouched.

7.3 Command Line

All the actions mentioned in Deploy and Settings chapter can be done from the command line:

To create application run:

```
/usr/bin/selectorctl --interpreter=<python|ruby> --version=VERSION [--user=USER] [--print-summary] [--json] --create-webapp <FOLDER_NAME> <URI>
```

To delete application:

```
/usr/bin/selectorctl --interpreter=<python|ruby> [--user=USER] [--print-summary] [--json] --destroy-webapp <FOLDER_NAME>
```

To change application folder name:

```
/usr/bin/selectorctl --interpreter=<python|ruby> [--user=USER] [--print-summary] [--json] --relocate-webapp <FOLDER_NAME> <NEW_FOLDER_NAME>
```

To change application URI:

```
/usr/bin/selectorctl --interpreter=<python|ruby> [--user=USER] [--print-summary] [--json] --transit-webapp <FOLDER_NAME> <NEW_URI>
```

To change application interpreter version:

```
/usr/bin/selectorctl --interpreter=<python|ruby> [--user=USER] [--print-summary] [--json] --set-user-current --version=<NEW_VERSION> <FOLDER_NAME>
```

To set application WSGI handler (python only):

```
/usr/bin/selectorctl --interpreter=python [--user=USER] [--print-summary] [--json] --setup-wsgi=<file_path:callable> <FOLDER_NAME>
```

To install modules to application environment:

```
/usr/bin/selectorctl --interpreter=python [--user=USER] [--print-summary] [--json] --enable-user-extensions=<module1[,module2...]> <FOLDER_NAME>
```

To remove modules from application environment:

```
/usr/bin/selectorctl --interpreter=python [--user=USER] [--print-summary] [--json] --disable-user-extensions=<module1[,module2...]> <FOLDER_NAME>
```

To list modules installed in application environment:

```
/usr/bin/selectorctl --interpreter=python [--user=USER] [--print-summary] [--json] --list-user-extensions <FOLDER_NAME>
```

To print applications summary for a user:

```
/usr/bin/selectorctl --interpreter=python [--user=USER] [--json] --user-summary
```

To list available interpreters:

```
/usr/bin/selectorctl --interpreter=python [--user=USER] [--json] --list
```


8 inodes Limits

[cPanel Only]

inodes Limits extension to LVE Manager allows you to set inodes limits for your customers. An inode is a data structure on a file system used to keep information about a file or a folder. The number of inodes indicates the number of files and folders an account has. inodes limits work on the level of disk quota, and will be enabled on /home partition only.

LVE Manager allows to set soft & hard IO limit.

- Hard limit prevents user from writing data to disk
- Soft limit can be exceeded for a period of time. The grace period can be set using: `edquota -t`

* Please, note that we don't collect statistical information related to inodes like we do for other LVE limits.

You can set inodes limits using LVE Manager, same way you would set other LVE Limits



Current Usage Settings Statistics Packages Options Selector Edit LVE

Settings for LVE 502 (YAKHZ1 - YAKHZTEST.COM)

CPU usage (CPU)	<input type="text" value="25"/>	
Number of cores for LVE (nCPU)	<input type="text" value="1"/>	
Virtual Memory (vMEM)	<input type="text" value="1024"/>	MB (0 - unlimited)
Physical memory (pMEM)	<input type="text" value="1024"/>	MB (0 - unlimited)
Concurrent connections (EP)	<input type="text" value="20"/>	
Number of processes (nPROC)	<input type="text" value="0"/>	(0 - unlimited)
I/O limit (IO)	<input type="text" value="1024"/>	KB/s (0 - unlimited)
Number of inodes (soft hard)	<input type="text" value="768"/> <input type="text" value="1024"/>	(0 - unlimited) <i>inodes</i>

Apply Cancel

The limits can be set on the level of individual account, and package:

LVE Manager

Current Usage Settings Statistics Packages Options Selector

Show resellers packages

Package ID	CPU	NCPU	VMEM (MB)	PMEM (MB)	EP	NPROC	IO (kBps)	INODES	
_yakhz_package_standard	25	1	1024	1024	20	-	1024	3072 4096	Edit
DEFAULT	25	1	1024	1024	20	-	1024	- -	

Sometimes disk quota breaks, so do inodes limits. You can reset them through 'Options' tab, in LVE Manager:

LVE Manager

Current Usage Settings Statistics Packages Options Selector

Hide LVE end user usage statistics

Show end-user inodes usage

Reset inode limits

Apply Cancel

End users can monitor their inodes usage using cPanel:

Stats	
Main Domain	yakhztest.com
Home Directory	/home/yakhz1
Last login from	172.17.0.33
CPU Usage	0 / 100 %
inodes	80 / 768
Disk Space Usage	0.31 / 10%
Monthly Bandwidth Transfer	0 / ∞

End user can also see their usage inside resource usage menu:

Current Usage			
Description	Usage	Limit	Fault
CPU Usage	0.0%	100%	-
inodes usage	80	768	-
I/O usage	0	1024	-
Entry Processes	0	20	0
Physical Memory Usage	0.00k	1.00G	0
Virtual Memory Usage	0.00k	1.00G	0

Timeframe:



8.1 Command Line Tool

`/usr/bin/cl-quota` -- tool to manage inodes limits

-u	--user	: print inode limits for all users
-U	--user-id	: print limits for a particular user
-N	--print-names	: print limit for a particular user id
-P	--package-limits	: print user names instead of user ids
-p	--package	: print limits for all packages
-S	--soft-limit	: print limits for a given package
-H	--hard-limit	: set soft limit for user or package
-Y	--sync	: set hard limit for user or package
-C	--cache	: re-set inodes limits for all accounts (fix quota)
-M	--mount-point	: generate cache file
-h	--help	: set partition for which to apply inodes limits
		: print help message

By default `/home` mountpoint is considered as having quota (or `/` if single partitioned system)

9 Kernel Settings

9.1 Virtualized /proc filesystem

You can prevent user from seeing processes of other users (via ps/top command) as well as special files in /proc file system by setting `fs.proc_can_see_other_uid` sysctl.

To do that, edit `/etc/sysctl.conf`

```
fs.proc_can_see_other_uid=0
fs.proc_super_gid=600
```

And do:

```
# sysctl -p
```

`fs.proc_can_see_other_uid=0`

If `fs.proc_can_see_other_uid` is set to 0, users will not be able to see special files. If it is set to 1 - user will see other processes IDs in /proc filesystem.

`fs.proc_super_gid=XX`

The `fs.proc_super_gid` sets group ID which will see system files in /proc , add any users to that group so they will see all files in /proc . Usually needed by some monitoring users like nagios or zabbix .

Virtualized /proc filesystem will only display following files (as well as directories for PIDs for the user) to unprivileged users:

```
/proc/cpuinfo
/proc/version
/proc/stat
/proc/uptime
/proc/loadavg
/proc/filesystems
/proc/stat
/proc/cmdline
/proc/meminfo
/proc/mounts
/proc/tcp
/proc/tcp6
/proc/udp
/proc/udp6
/proc/assocs
/proc/raw
/proc/raw6
/proc/unix
/proc/dev
```

9.2 SecureLinks

Starting with kernel **Ive1.1.95.1** and **Ive0.8.62** and higher, *SecureLinks* are implemented on the kernel level. They are set via kernel level parameters and can be overwritten using `sysctl`.

Defaults:

```
fs.enforce_symlinksifowner = 1
fs.symlinkown_gid = 48
```

```
fs.enforce_symlinksifowner == 0 -> do not check symlink ownership
fs.enforce_symlinksifowner == 1 -> deny if gid == symlinkown_gid
fs.enforce_symlinksifowner == 2 -> deny if gid > symlinkown_gid [since
kernel 2.6.32-379.19.1.Ive1.2.8]
```

When `fs.enforce_symlinksifowner` set to 1, processes with GID 48 will not be able to follow symlinks if they are owned by user1, but point to file owned user2.

When `fs.enforce_symlinksifowner` set to 2, processes with GID > 48 will not be able to follow symlinks if they are owned by user1, but point to file owned user2. **[since kernel 2.6.32-379.19.1.Ive1.2.8]**

***fs.enforce_symlinksifowner=2 should only be used with mod_ruid2. This option is not needed when used with suPHP, mod_fcgid, CGI or LiteSpeed. It will also break PHP Selector.**

On standard RPM Apache installation, apache is usually running under GID 48

On cPanel servers, Apache is running under user nobody, GID 99.

To change GID of processes that cannot follow symlink, edit file `/etc/sysctl.conf`, add line:

```
fs.symlinkown_gid = XX
```

And execute

```
$ sysctl -p
```

To disable SecureLinks feature, set `fs.enforce_symlinksifowner = 0` in `/etc/sysctl.conf`, and execute

```
$ sysctl -p
```

9.3 ptrace Block

Starting with kernel **Ive1.2.12** and **Ive0.8.62** and higher we implemented *ptrace block* to protect against ptrace family of vulnerabilities such as <http://seclists.org/oss-sec/2013/q1/326>. It prevents end user from using any ptrace related functionality, including such commands as **strace**, **lsof** & **gdb**

By default, CloudLinux doesn't prevent ptrace functionality.

Defaults:

```
kernel.user_ptrace = 1
```

```
kernel.user_ptrace == 0 -> user is blocked from using ptrace
kernel.user_ptrace == 1 -> user is allowed to use ptrace, default
```

To disable ptrace for end users, edit file `/etc/sysctl.conf`, add line:

```
kernel.user_ptrace = 0
```

And execute

```
$ sysctl -p
```

* **ptrace protection is known to brake PSA service for Plesk 11**

9.4 Xen XVDA detection

2.6.32 kernels have different mode of naming Xen XVDA drives.

By adding `xen_blkfront.sda_is_xvda=0` to kernel boot line in `grub.conf` you will make sure no naming translation is done, and the drives will be identified as `xvde`

By default, this option is set to 1 in the kernel, and drives are detected as `xvda`

This is needed only for CL6 and Hybrid kernels.

9.5 TPE Extension

[TPE Extension will removed in the next version of CloudLinux 5.x kernel]

CloudLinux 5.x (kernel 2.6.18) has limited support for trusted path execution extension.

CloudLinux 6.x (kernel 2.6.32) and CloudLinux 5.x with hybrid kernel don't have TPE extension

TPE (Trusted Path Execution)

The kernel supports TPE feature out of the box. You can configure it using following files:

- `/proc/sys/kernel/grsecurity/grsec_lock`
- `/proc/sys/kernel/grsecurity/tpe`
- `/proc/sys/kernel/grsecurity/tpe_gid`
- `/proc/sys/kernel/grsecurity/tpe_restrict_all`

To enable TPE feature in a standard way just add following to the end of your `/etc/sysctl.conf`

```
#GRsecurity
kernel.grsecurity.tpe = 1
kernel.grsecurity.tpe_restrict_all = 1
kernel.grsecurity.grsec_lock = 1
```

And do:

```
# sysctl -p
```

*Note: Once you set **grsec_lock** to 1, you will not be able to change TPE options without reboot.*

This Trusted Path Execution feature was adopted from grsecurity

9.6 IOLimits latency

[lve1.2.29+]

When customer reaches IO Limit, the processes that are waiting for IO will be placed to sleep to make sure they don't go over the limit. That could make some processes sleep for a very long time.

By defining IO latency, you can make sure that no process sleeps due to IO limit for more then X nanoseconds. By doing so, you will also let customers to burst through the limits, and use up more than they were limited too in some instances.

This option is OFF by default.

To enable IOLimits latency and set it to 10 seconds

```
# echo 10000 > /sys/module/iolimits/**parameters/latency
```

To disable latency:

```
# echo 2000000000 > /sys/module/iolimits/**parameters/latency
```

9.7 Hybrid Kernel

Hybrid Kernel is CloudLinux 6 (2.6.32-lve1.x) kernel compiled for CloudLinux 5 OS. It brings new features available in CloudLinux 6 kernel, like IO limits, to CloudLinux 5 servers. When you switch to hybrid kernel, you will also switch to a new channel, that has all other components compatible with the new kernel. The kernel is as stable as the CloudLinux 6 production kernel and has same features.

While the kernel itself is fully production quality, the process of switching to hybrid kernel is not perfect. This is mostly due to differences in naming for the devices and modules between 2.6.32 and 2.6.18 kernels. As the result, your server might not boot with new kernel. To solution that, boot into previous (CL5) kernel, and convert back. After that -- notify our support department at <https://helpdesk.cloudlinux.com>. In most cases we can resolve all the naming issues, and make sure you can boot into hybrid kernel

Switching to hybrid kernel:

```
# yum update rhn-setup
# /usr/sbin/normal-to-hybrid
# reboot
```

To convert back to original CloudLinux 5 kernel:

```
# yum update rhn-setup
# /usr/sbin/hybrid-to-normal
# reboot
```

9.8 Reading LVE usage

CloudLinux kernel provides real time usage data in `/proc/lve/list` file.

All the statistics can be read from that file in real time. Depending on your kernel version you will get either Version 6 of the file, or version 4 of the file.

You can detect the version by reading the first line of the file. It should look like:

6:LVE... for version 6

4:LVE... for version 4

First line presents headers for the data.

Second line shows default limits for the server, with all other values being 0.

The rest of the lines present limits & usage data on per LVE bases.

Version 6 (CL6 & hybrid kernels)

6:LVE	EP	lCPU	lIO	CPU	MEM	IO	lMEM	lEP	nCPU	fMEM	fEP	lMEMPHY	PHYS
0	0	25	1024	0	0	0	262144	20	1	0	0	262144	100
300	0	25	1024	1862407		0	0	262144	20	1	0	0	262144

Version 4 (CL 5 kernel)

4:LVE	EP	lCPU	lIO	CPU	MEM	IO	lMEM	lEP	nCPU	fMEM	fEP
0	0	25	25	0	0	0	262144	20	1	0	0
300	0	25	25	15103019		0	0	262144	20	1	0

Label	Description	Value	Supported versions
LVE	LVE ID	number	
EP	Number of entry processes	number	
ICPU	CPU Limit	% relative to total CPU power	
IIO	IO limits for CL6 or IO priority for CL5	KB/s for v6, from 1 to 100 for v4	
CPU	CPU usage since reboot	in nanoseconds for v6, hertz for v4	
MEM	Virtual memory usage	number of 4k pages	
IO	IO usage	KB/s for v6, 0 for v4	
IMEM	Virtual memory limit	number of 4k pages	
IEP	Entry Processes limit	number	
nCPU	Number of cores limit	number of cores	
mMEM	Virtual memory faults	number of faults	
mEP	Entry Processes faults	number of faults	v6+
IMEMPHY	Physical memory limit	number	v6+
ICPUW	CPU weight (not used)	from 1 to 100	v6+
INPROC	Number of processes limit	number	v6+
MEMPHY	Physical memory usage	number of 4k pages	v6+
mMEMPHY	Physical memory faults	number of faults	v6+
NPROC	Number of processes	number	v6+
mNPROC	Number of processes faults	number of faults	v6+

9.9 flashcache

** Available only for x86_64, CL6 and hybrid servers*

Flashcache is a module originally written and released by Facebook (Mohan Srinivasan, Paul Saab and Vadim Tkachenko) in April of 2010. It is a kernel module that allows Writethrough caching of a drive on another drive. This is most often used for caching a rotational drive on a smaller solid-state drive for performance reasons. This gives you the speed of an SSD and the size of a standard rotational drive for recently cached files. Facebook originally wrote the module to speed up database I/O, but it is easily extended to any I/O.

To install on CL6 & hybrid servers:

```
$ yum install flashcache
```

More info on flashcache: <https://github.com/facebook/flashcache/>

ArchLinux has a good page explaining how to use flashcache:
<https://wiki.archlinux.org/index.php/Flashcache>

10 Apache mod_lsapi

[beta]

Apache mod_lsapi is a module based on LiteSpeed Technologies API for PHP, Ruby and Python. It offers excellent PHP performance, low memory footprint coupled with great security and support for opcode caching.

Requirements

CageFS (installed and initialized)

Alt-PHP

Apache with SuExecUserGroup directive for each user's VirtualHost,

mod_ruid2 disabled

Configuration Options

Options	Description	Level
php_value, php_admin_value, php_flag, php_admin_flag	mod_php emulation	httpd.conf, virtualhost, htaccess
lsapi_backend_connect_timeout	number of usec to wait while lsPHP starts (if not started on request)	httpd.conf
lsapi_backend_connect_tries	number of retries to connects to lsPHP daemon	httpd.conf
lsapi_terminate_backends_on_exit	httpd.conf, On - stop lsphp services on apache restart, Off - leave live started lsphp services on apache restart (for php+opcache). The lsphp will not restart, even if Apache gets restarted.	httpd.conf
lsapi_backend_children	sets env variable LSAPI_CHILDREN # lsphp also try to read PHP_LSAPI_CHILDREN var # Required and should be >0 in order to enter into self-managed mode # min value is 1; max value is 10000. if var value is more, 10000 will be used.	httpd.conf
lsapi_backend_max_process_time	env variable LSAPI_MAX_PROCESS_TIME # Optional. Default value is 3600 # Timeout to kill runaway processes	httpd.conf
lsapi_backend_pgrp_max_idle	sets env variable LSAPI_PGRP_MAX_IDLE, in seconds controls how long an control process will wait for # a new request before it exits. # Optional, default value is 0 -> infinite	httpd.conf

	# export LSAPI_PGRP_MAX_IDLE=0	
lsapi_debug	enable debugging for mod_lsapi, acceptable values: on/off	httpd.conf
lsapi_socket_path	Path to back end lsphp sockets. By default /tmp/lshttpd	httpd.conf
lsapi_phpirc	Sets PHPRC env variable	httpd.conf, virtualhost
lsapi_user_group	Set user & group for requests	httpd.conf, virtualhost, directory
lsapi_uid_gid	Set user id & group id for requests	httpd.conf, virtualhost, directory
lsapi_use_default_uid	Use default apache UID/GID if no uid/gid set. Values: On/Off. If Off, and no UID/GID set, error 503 will be returned. Default - Off	httpd.conf
lsapi_target_perm	check target PHP script permissions. If set to On, lsapi will check that script is owned by the same user, as user under which it is being executed. Return 503 error if they don't match. Default: Off	httpd.conf
lsapi_selfstarter	Use or not separate process for starting lsphp. By default is - On. For apache prefork can be used parameter Off (because of low level virtual memory usage by apache prefork), but for event and worker should be - On (because high level of virtual memory usage by event and worker). Acceptable values: on/off.	httpd.conf
lsapi_poll_timeout	By default - 0 (infinity). For preventing long running processes which can use EP (limit number of entry processes). In seconds - time to wait response from lsphp daemon.	httpd.conf
lsapi_mutex_mech	Default value is "default" (experimental option yet, for checking problem with semaphores). Values: default, fcntl, flock, posixmem, pthread, syssem.	httpd.conf
lsapi_backend_coredump	env variable LSAPI_ALLOW_CORE_DUMP (On or Off). Pass	httpd.conf

	<p>LSAPI_ALLOW_CORE_DUMP to lsphp or not. If it will be passed - core dump on lsphp crash will be created.</p> <p># Off by default</p> <p># By default a LSAPI application will not leave a core dump file when crashed. If you want to have # LSAPI PHP dump a core file, you should set this environment variable. If set, regardless the # value has been set to, core files will be created under the directory that the PHP script in.</p> <p>LSAPI ALLOW CORE DUMP</p>	
--	--	--

Example configuration

```
LoadModule lsapi_module modules/mod_lsapi.so

<IfModule lsapi_module>
  AddType application/x-httpd-lsphp .php
  lsapi_backend_connect_timeout 100000
  lsapi_backend_connect_tries 10
  lsapi_backend_children 20
  lsapi_backend_pgrp_max_idle 30
  lsapi_backend_max_process_time 300
  lsapi_debug Off
</IfModule>
```

Secret File

When installed, liblsapi will automatically create secret file used by mod_lsapi to communicate with backend:

```
/etc/sysconfig/modlsapi.secret
owner root:root
perms: 400
```

Command Line Tools (cPanel only):

```
/usr/bin/switch_mod_lsapi [OPTIONS]
```

Options:

- setup - setup mod_lsapi configurations for apache
- uninstall - uninstall mod_lsapi from apache
- enable-domain - enable mod_lsapi for individual domain
- disable-domain - disable mod_lsapi for individual domain
- enable-global - sets up mod_lsapi as a default way to serve PHP, making it enabled for all domains. Once that mode is enabled, you cannot disable mod_lsapi for individual domain
- disable-global - disable mod_lsapi as a default way to serve PHP, disabling mod_lsapi for all domains, including those selected previously using --enable-domain
- build-native-lsphp - build native lsphp for cPanel

This tool:

- At the moment, works only with cPanel
- creates native lsphp (if it doesn't exist) by doing: `cp /opt/alt/php54/usr/bin/lspsh /usr/local/bin/`
- creates `/tmp/lshhttpd` and adds it to `/etc/cagefs/cagefs.mp`
- Removes config template for `mod_ruid2`
- Configures Apache handler `application/x-httpd-lsphp`
- Switch domain to lsphp or enable global lsphp
- For cPanel can build native lsphp

Once enabled, you can enable `mod_lsapi` for a site by adding to `.htaccess` for that site:

```
AddType application/x-httpd-lsphp .php5 .php4 .php .php3 .php2 .phtml .php
```

To enable `mod_lsapi` for all sites, edit cPanel's Apache configuration to use `application/x-httpd-lsphp` for `.php` extensions

10.1 Installation

[beta]

For all control panels - `SuExecUserGroup` should be present for each virtual host. `CageFS` and `PHP Selector` will be installed by dependencies (for `lsphp` binaries)

If `CageFS` is not initialized:

```
$ cagefsctl --init
$ cagefsctl --enable-all
```

Installing on cPanel servers

```
$ yum install liblsapi liblsapi-devel --enablerepo=cloudlinux-updates-testing
$ yum install cpanel-mod-lsapi --enablerepo=cloudlinux-updates-testing
```

```
$ /usr/bin/switch_mod_lsapi --setup
# Enable for a single domain:
$ /usr/bin/switch_mod_lsapi --enable-domain [domain]
# or globally
$ /usr/bin/switch_mod_lsapi --enable-global
$ service httpd restart
```

Installing on DirectAdmin servers

```
$ cd /usr/local/directadmin/custombuild
$ ./build update
$ ./build set php1_mode lsphp
$ ./build php n
$ ./build apache
```

Installing on ISPManager servers

```
$ yum install liblsapi liblsapi-devel --enablerepo=cloudlinux-updates-testing
$ yum install mod_lsapi --enablerepo=cloudlinux-updates-testing
$ /usr/bin/switch_mod_lsapi --setup
```

uncomment string LoadModule lsapi_module modules/mod_lsapi.so from file /etc/httpd/conf.d/mod_lsapi.conf

disable php support for needed domain (this action comment out AddHandler or AddType for VirtualHost) or for all domains.

remove from /etc/httpd/conf/httpd.conf strings:

```
<Directory /var/www/*/data/>
php_admin_flag engine off
</Directory>
```

Alternatively:

add to needed (where mod_lsapi should be enabled) VirtualHost such strings:

```
<Directory /var/www/[username]/data/www/[domain]>
Options -ExecCGI -Includes
php_admin_flag engine on
</Directory>
```

uncomment string AddType application/x-httpd-lsphp .php5 .php4 .php .php3 .php2 .phtml in file /etc/httpd/conf.d/mod_lsapi.conf

service httpd restart

RPM Installation

```
$ yum install liblsapi liblsapi-devel --enablerepo=cloudlinux-updates-testing
$ yum install mod_lsapi --enablerepo=cloudlinux-updates-testing
$ /usr/bin/switch_mod_lsapi --setup
```

Disable php.conf or any other php handler and uncomment AddType application/x-httpd-lsphp .php .php4 .php3 .phtml in /etc/httpd/conf.d/mod_lsapi.conf and restart Apache

```
$ service httpd restart
```

Building from source:

This steps are needed for installation of lsphp binaries needed for mod_lsapi

```
$ yum install cagefs lvmanager cmake gcc httpd-devel apr-devel
$ yum groupinstall alt-php
```

if lsphp already exists, copy it to /usr/local/bin/lsphp (this step allows you to avoid installing alt-php)

Compile mod_lsapi

```
$ yum install liblsapi liblsapi-devel --enablerepo=cloudlinux-updates-testing
$ wget http://repo.cloudlinux.com/cloudlinux/sources/da/mod_lsapi.tar.gz
$ tar zxvf mod_lsapi.tar.gz
$ cd mod-lsapi-0.1-37
$ cmake .
$ make
$ make install
$ cp conf/mod_lsapi.conf /etc/httpd/conf/extra/ #(or another httpd conf directory)
$ service httpd restart
```

This will:

- Install: /usr/lib/apache/mod_lsapi.so (or to another correct httpd modules path)
- Install: /usr/sbin/sulspsh

if you want lsapi as global php handler, uncomment #AddType application/x-httpd-lsphp .php and disable current PHP handler. If server uses suPHP, you can enable lsphp for single hosts. Just add AddType application/x-httpd-lsphp .php5 .php4 .php .php3 .php2 .phtml to site's .htaccess

10.2 Uninstall

[beta]

cPanel Servers

```
$ /usr/bin/switch_mod_lsapi --uninstall
```

DirectAdmin servers

```
$ cd /usr/local/directadmin/custombuild
$ ./build update
$ ./build set php1_release [any other php type]
$ ./build php n
$ ./build apache
```

RPM:

```
$ yum erase mod_lsapi
$ rm [path to mod_lsapi.conf]
# restore standard php handler
$ service httpd restart
```

10.3 Troubleshooting mod_lsapi

[beta]

1. Non-standard apache user

if apache runs under a username other than "apache" or "nobody", you should rebuild sulspsh (where username is built in for security reasons) with corresponding username:

```
$ yum install liblsapi liblsapi-devel --enablerepo=cloudlinux-updates-testing
$ cd ~
$ wget http://repo.cloudlinux.com/cloudlinux/sources/da/mod\_lsapi.tar.gz
$ tar zxvf mod_lsapi.tar.gz
$ cd mod-lsapi-0.1-37
$ cmake -DHOTTPD_USER=<new user name> .
$ make
$ make install
```

This will:

- Install: /usr/lib/apache/mod_lsapi.so (or to another correct httpd modules path)
- Install: /usr/sbin/sulspsh

2. Isphp started under user apache/nobody

Check if SuExecUserGroup specified for virtual hosts. This parameter is used by mod_lsapi for user identification.

3. Could not connect to Isphp backend: connect(/tmp/lshttpd/lsapi_application-x-httpd-lsphp_XXX.sock) failed: 111 Connection refused

- switch in lsapi.conf or mod_lsapi.conf value to: lsapi_terminate_backends_on_exit Off
- check if empty: `cat /etc/cron.d/kill_orphaned_php-cron | grep lsphp` - then `yum install lve-utils --enablerepo=cloudlinux-updates-testing` and restart cron service

4. Running PHP for users with UID < 99

If you need to run PHP using mod_lsapi using users with UID < 99, you would need to re-compile sulspsh:

- `yum install liblsapi liblsapi-devel --enablerepo=cloudlinux-updates-testing`
- `cd ~`
- `wget http://repo.cloudlinux.com/cloudlinux/sources/da/mod_lsapi.tar.gz`
- `tar zxvf mod_lsapi.tar.gz`
- `cd mod-lsapi-0.1-XX`
- `cmake -DUID_MIN=80 -DGID_MIN=80 .`
- `make`
- `make install`

will be installed

- Installing: /usr/lib/apache/mod_lsapi.so (or another httpd modules path)
- Installing: /usr/sbin/sulspsh

5. Apache binary called not httpd (httpd.event, httpd.worker)

- `yum install liblsapi liblsapi-devel --enablerepo=cloudlinux-updates-testing`
- `cd ~`
- `wget http://repo.cloudlinux.com/cloudlinux/sources/da/mod_lsapi.tar.gz`
- `tar zxvf mod_lsapi.tar.gz`
- `cd mod-lsapi-0.1-XX`
- `cmake -DPARENT_NAME="<apache binary name>".`
- `make`
- `make install`

will be installed

- Installing: /usr/lib/apache/mod_lsapi.so (or another httpd modules path)
- Installing: /usr/sbin/suexec

6. WHMCS Status page not accessible after installing CL and mod_lsapi (cPanel).

- add user: useradd userstat
- add to file (to the end of file before </IfModule>) /usr/local/apache/conf/conf.d/lsapi.conf: <Directory /usr/local/apache/htdocs/>
lsapi_user_group userstat userstat
</Directory>
- service httpd restart

This is safe solution for easyapache rebuilding and cpanel-mod-lsapi updating.

7. PHP page with Suhosin return 503 error.

Make php.ini for suhosin as recommended below:

```
[suhosin]
suhosin.simulation = Off
suhosin.mail.protect = 1
suhosin.cookie.disallow_nul = Off
suhosin.cookie.max_array_depth = 1000
suhosin.cookie.max_array_index_length = 500
suhosin.cookie.max_name_length = 500
suhosin.cookie.max_totalname_length = 500
suhosin.cookie.max_value_length = 200000
suhosin.cookie.max_vars = 16384
suhosin.get.disallow_nul = Off
suhosin.get.max_array_depth = 1000
suhosin.get.max_array_index_length = 500
suhosin.get.max_name_length = 500
suhosin.get.max_totalname_length = 500
suhosin.get.max_value_length = 1000000
suhosin.get.max_vars = 16384
suhosin.post.disallow_nul = Off
suhosin.post.max_array_depth = 1000
suhosin.post.max_array_index_length = 500
suhosin.post.max_name_length = 500
suhosin.post.max_totalname_length = 500
suhosin.post.max_value_length = 1000000
suhosin.post.max_vars = 16384
suhosin.request.disallow_nul = Off
suhosin.request.max_array_depth = 1000
suhosin.request.max_array_index_length = 500
suhosin.request.max_totalname_length = 500
suhosin.request.max_value_length = 1000000
suhosin.request.max_vars = 16384
suhosin.request.max_varname_length = 524288
suhosin.upload.max_uploads = 300
suhosin.upload.disallow_elf = Off
suhosin.session.cryptua = Off
```

```
suhosin.session.encrypt = Off
suhosin.session.max_id_length = 1024
suhosin.executor.allow_symlink = Off
suhosin.executor.disable_eval = Off
suhosin.executor.disable_emodifier = Off
suhosin.executor.include.max_traversal = 8
```

8. PHP page with APC return 503 error.

Make php.ini for apc as recommended below:

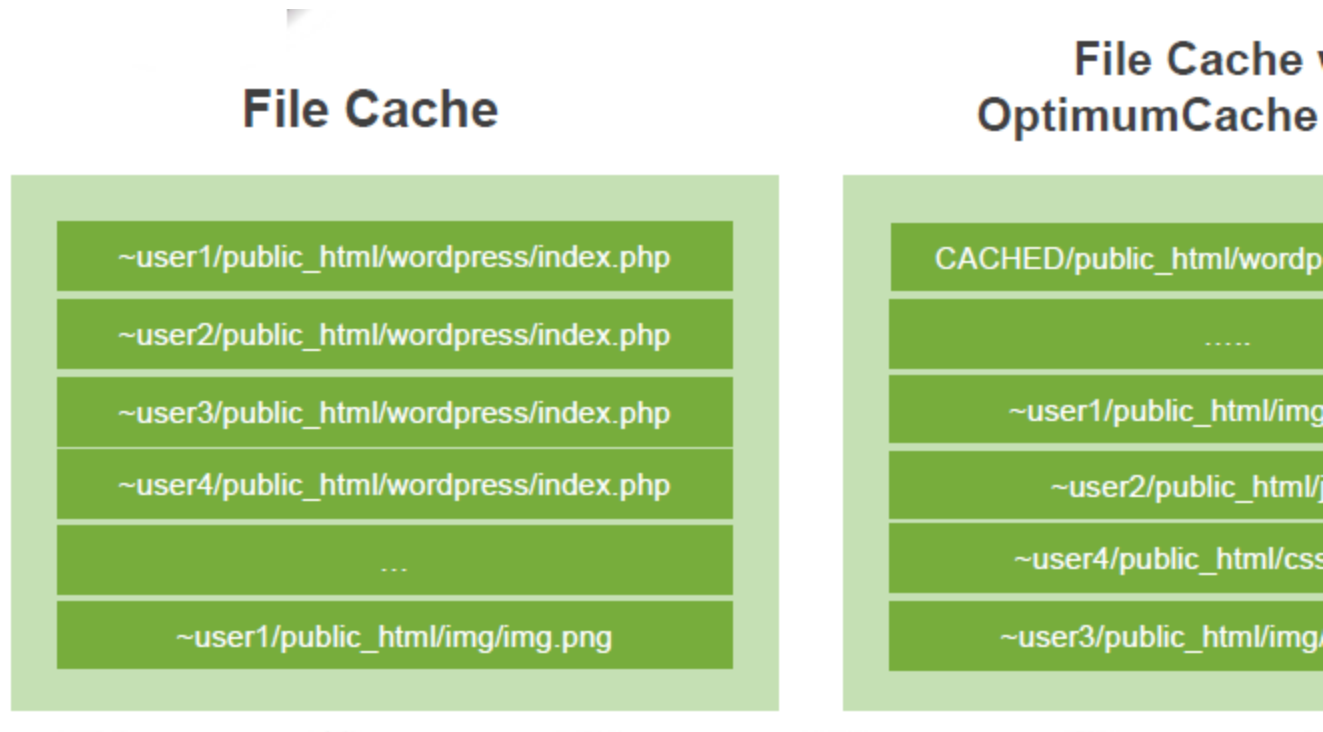
```
[apc]
...
apc.shm_segments=1
apc.shm_size=32
...
```

shared memory should be not less than 32MB

11 OptimumCache

OptimumCache 0.2+

OptimumCache is a de-duplicating file cache optimized specifically for shared hosting. Typical shared hosting server runs a number of sites with WordPress and Joomla as well as other popular software. This usually means that there are hundreds of duplicate files that are constantly being read into file cache - both wasting precious disk IO operations as well as memory. OptimumCache creates a cache of such duplicated files and de-duplicates file cache.



With OptimumCache, if a duplicate of an already loaded file is requested, the file gets loaded from filesystem cache. By doing that, system bypasses disk IO, significantly improving the speed of reading that file, while lowering load on the hard disk. As the file had been read from disk just once, it is cached by filesystem cache just once, minimizing amount of duplicates in file system cache and improving overall cache efficiency. This in turn reduces memory usage, decreases the number of disk operations - all while improving the websites response time.

11.1 Installation

Requirements:

64bit CloudLinux 6.x or higher
ext4 filesystem
kernel lve1.2.55 or later.

Installation:

```
# yum install optimumcache --enablerepo=cloudlinux-updates-testing
```

Allocating Disk Space for OptimumCache:

By default OptimumCache will attempt to setup 5GB ploop (high efficiency loopback disk) to be used for the cache in `/var/share/optimumcache/optimumcache.image`

That ploop will be mounted to: `/var/cache/optimumcache`

The ploop image will be located at `/var/share/optimumcache/optimumcache.image`

Allocating OptimumCache disk space for ploop on a fast drives (like SSD) will provide additional performance improvement as more duplicated files would be loaded from fast disks into memory.

Moving ploop image to another location:

```
# occtl --move-ploop /path/to/new/image/location [new size[KMG]]
```

If 'new size' is not mentioned, then value from `/etc/sysconfig/optimumcache` is used. If `/etc/sysconfig/optimumcache` does not mention anything regarding ploop image size, then default 5GB is used.

Enabling and disabling ploop:

To turn on ploop:

```
# occtl --init-ploop
```

To disable ploop:

```
# occtl --disable-ploop
```

If ploop image has been mounted in `/etc/fstab` for OpimumCache-0.1-21 and earlier, you may consider removing this fstab entry in OpimumCache 0.2+. That is because since 0.2+ ploop is mounted automatically at service start.

If you prefer leave that fstab mount point as is, you may see some warnings when you decide to move ploop later via `'occtl --move-ploop'`.

Resizing ploop:

To resize ploop:

```
# occtl --resize-ploop [new size[KMG]]
```

For the case when resize cannot be done due to "Unable unmount ploop" issue, there is a workaround in "Troubleshooting" section.

11.2 Using without ploop

On servers with kernel prior to lve1.2.55 ploop will not be used (due to ploop related issues in the kernel). Instead cached files will be stored in `/var/cache/optimumcache`.

The cache will be cleaned (shrunk) by 20% once partition on which `OPTIMUMCACHE_MNT` resides has only 10% of free space. You can change that by changing `PURGEAHEAD` param in `/etc/sysconfig/optimumcache`, and restarting `optimumcache` service.

The cache is cleaned `/etc/cron.d/optimumcache_cron` script `optimumcache_purge`, which runs every minute:

```
0-59 * * * * root /usr/share/optimumcache/optimumcache_purge
```

11.3 Marking Directories

Marking directories to be cached:

```
# occtl --mark-dir /path/to/site/on/filesystem --recursive
```

By default OptimumCache marks `/home` directory to be cached.

Ignoring particular files & directories:

OptimumCache tracks files & directories that need to be cached. Once file is modified, it will no longer be tracked by OptimumCache (as there is very little chance that it will have a duplicate). Yet, all new files created in tracked directories are checked for duplicates.

Sometimes you might want to ignore such checks for directories where large number of temporary or new files are created, that will not have duplicates - as such checks are expensive. Directories like mail queue, and tmp directories should be ignored.

You can set a regexp mask for directories that you would like to ignore using:

```
$ occtl --add-skip-mask REGEX
```

To list skip masks:

```
$ occtl --list-skip-mask
```

To remove skip mask:

```
$ occtl --remove-skip-mask ID|Tag
```

By default, OptimumCache sets up following skip masks:

id	tag	regex
1	all_dot_files	/\...*
2	cagefs	^/home/cagefs-skeleton\$
3	cagefs	^/home/cagefs-skeleton/
4	cpanel	^/home[^]*/cPanelInstall

```

5 cpanel      ^/home[^]*/cpeasyapache
6 cpanel      ^/home[^]*/aquota
7 cpanel      ^/home[^]*/jailshell
8 cpanel      ^/home[^]*/[^]+/mail$
9 cpanel      ^/home[^]*/[^]+/mail/*
10 cpanel     ^/home[^]*/[^]+/logs$
11 cpanel     ^/home[^]*/[^]+/logs/*
12 cpanel     ^/home[^]*/[^]+/\..cpanel$
13 cpanel     ^/home[^]*/[^]+/\..cpanel/*
14 cpanel     ^/home[^]*/[^]+/\..cagefs
15 cpanel     ^/home[^]*/[^]+/\..cagefs/*
16 cpanel     ^/home[^]*/virtfs
17 cpanel     ^/home[^]*/virtfs/*
18 not_a_userdi^/home/tmp/
   r
19 not_a_userdi^/home/tmp$
   r
20 not_a_userdi^/home/ftp/
   r
21 not_a_userdi^/home/ftp$
   r
22 not_a_userdi^/home/admin/
   r
23 not_a_userdi^/home/admin$
   r
24 quota      ^/home[^]*/quota.user$
25 usermisc   /quota.user$
26 users_home ^/home/[^]+/backups$
27 users_home ^/home/[^]+/backups/
28 users_home ^/home/[^]+/imap$
29 users_home ^/home/[^]+/imap/
30 users_home ^/home/[^]+/Maildir$
31 users_home ^/home/[^]+/Maildir/
32 users_home ^/home/[^]+/domains/[^]+/logs$
33 users_home ^/home/[^]+/domains/[^]+/logs/
34 users_home ^/home/[^]+/domains/[^]+/public_ftp$
35 users_home ^/home/[^]+/domains/[^]+/public_ftp/
36 users_home ^/home/[^]+/domains/[^]+/stats$
37 users_home ^/home/[^]+/domains/[^]+/stats/

```

This information is stored in `/etc/container/optimumcache/ignore.d/`

Skip Mask syntax

Skip masks use following regexp syntax: <http://www.greenend.org.uk/rjk/tech/regexp.html>

For example, to disable caching all directories that contain `*/cache/*`, you should use skip masks like:

```

/cache/
/cache$

```

This information is stored in `/etc/container/optimumcache/ignore.d/`

11.4 Configuration File

OptimumCache configuration file:

```
/etc/sysconfig/optimumcache
```

```
# Location of cache
```

```
OPTIMUMCACHE_MNT=/var/cache/optimumcache
```

```
# Valency to cache
```

```
COUNT=0
```

```
# Minimal file size to cache, default - cache all files
```

```
# MINSIZE=0
```

```
# Minimal page number in file to start caching, default - 1
```

```
PAGEMIN=0
```

```
# Maximum file size to cache, 2147483648 (2GB) by default
```

```
# MAXSIZE
```

```
# Interval between caching attempts, default - 5 seconds
```

```
# TIMEOUT=7
```

```
# Limit IO bandwidth of pfcached, in bps, default - unlimited
```

```
# OPTIMUMCACHE_IOLIMIT=10485760
```

```
# Limit IOPS of pfcached, default - unlimited
```

```
# OPTIMUMCACHE_IOPSLIMIT=200
```

```
# Extra space in %% of requested to purge, default 20%
```

```
# PURGEAHEAD=20
```

```
# Logging verbosity, default - 1, verbose
```

```
# LOGLEVEL=1
```

```
# Y to use PLOOP
```

```
USE_PLOOP=Y
```

```
# Eliminates superflows fsync() calls in OptimumCache operation
```

```
# NOIMMSYNC=1
```

11.5 Command Line Interface

OptimumCache is controlled using occtl command line utility.

Usage:

```
occtl [-h] [--move-ploop param [param ...]] [--check] [--verbose]
      [--init-ploop [param [param ...]]] [--resize-ploop New Size]
      [--disable-ploop] [--enable-ploop] [--mount-ploop]
      [--unmount-ploop] [--delete-ploop] [--unmark-all]
```

```
[--mark-dir Path] [--unmark-dir Path] [--recursive]
[--add-skip-mask Regex] [--remove-skip-mask Id|Tag]
[--list-skip-mask] [--silent] [--ignore-unmount-failure]
```

Check status:

```
optimumcache stat /home/
```

Optional Arguments:

```
-h, --help          show this help message and exit
--move-ploop param  Move cache from one ploop image to /path/to/new/image/location [New
[param ...]         Size[KMGTT]]
--check            Check marked files for errors
--verbose         List what is being checked
--init-ploop [param Create ploop image for the cache [/path/to/ploop/image [ploop_size] |
[param ...]]       ploop_size] - if only one parameter is given, it is considered to be ploop size.
                   Size should be a NUMBER[KMGTT]
--resize-ploop New Size New Size NUMBER[KMGTT]
--disable-ploop    Disable ploop
--enable-ploop     Enable ploop
--mount-ploop      Mount ploop image
--unmount-ploop    Unmount ploop image
--delete-ploop     Delete ploop image. Implies disable ploop, if was enabled.
--unmark-all      Unmark all marked directories
--mark-dir Path    Mark directory for caching
--unmark-dir Path  Unmark directory for caching
--recursive        Is used with mark/unmark dir
--add-skip-mask Regex Regexp to skip files/directories for caching
--remove-skip-mask Id| Tag Remove regexp to skip files/directories by id or tag
--list-skip-mask   List regexp to skip files/directories
--silent          Do not echo status to stdout / syslog
--ignore-unmount-failure Ignore cannot unmount ploop problem
```

11.6 Uninstall OptimumCache

To uninstall OptimumCache run:

```
# rpm --erase optimumcache
```

For OptimumCache version prior 0.2-11, uninstalling via rpm package manager does not automatically removes away ploop image. That is because not always possible to unmount it properly due to kernel dependency. If there is no luck with unmounting ploop, then the server will have to be rebooted and will need to remove ploop files manually:

```
# rm /var/share/optimumcache/optimumcache.image
# rm /var/share/optimumcache/DiskDescriptor.xml
# rm /var/share/optimumcache/DiskDescriptor.xml.lck
```

or

```
# rm /path/to/ploop/image/file
# rm /path/to/ploop/image/DiskDescriptor.xml
# rm /path/to/ploop/image/DiskDescriptor.xml.lck
```

For OptimumCache version 0.2-11 and later, ploop image will be removed automatically during uninstall. If ploop unmount issue prevents doing that, ploop image clean up will be scheduled after next server reboot.

11.7 Troubleshooting

Installing for FS is different from Ext4:

For now Ext4 is the only supported file system type. If a host has no Ext4 filesystem mounted, OptimumCache package installation will be abandoned:

```
Preparing packages for installation...
Cannot continue: Ext4 partition is the only supported by OptimumCache, there is no one in
fstab
error: %pre(optimumcache-0.1-22.el6.cloudlinux.x86_64) scriptlet failed, exit status 1
error: install: %pre scriptlet failed (2), skipping
```

Also, an attempt to add for caching directory, which does not reside on Ext4, will fail:

```
# occtl --mark-dir /home --recursive
mount: / not mounted already, or bad option
optimumcache: Can not mount device. rc[8192]
Error: mark[1]: /usr/bin/optimumcache mark --recursive /home
```

Ploop: Cannot unmount old ploop image

This is well-known ploop problem, which may result in failing such actions as resizing or moving ploop in OptimumCache. To workaround this problem use '--ignore-unmount-failure' with --move-ploop:

```
# occtl --move-ploop --ignore-unmount-failure
```

As for resizing ploop, use flavor of '--move-ploop' command instead:

```
# occtl --move-ploop /path/to/new/image/file [size GB] --ignore-unmount-failure
```

For your changes to take effect, the server has to be rebooted. Upon reboot, you may clean up manually old ploop image file and DiskDescriptor.xml file, which resides in the same directory along with old image.

High IO rate

High IO problem was fixed in latest version of OptimumCache (version 0.2-6). The fix is to eliminate superflows fsync() calls in OptimumCache operations. To activate this fix in existing installation, flag NOIMMSYNC=1 has to be manually set in /etc/syscoconfig/optimumcache.

To ensure that this parameter is set ON in the config, set LOGLEVEL=2 and execute 'service optimumcache restart'. You will see something like this:

```
optimumcache[1770]: Hash-size: 100000000 min-size: 0 max-size: 18446744071562067968
optimumcache[1770]: Count: 0 Timeout: 5
optimumcache[1770]: Max Timeout: 160 Adaptive Timeout Mul/Div: 2/4
optimumcache[1770]: Iolimit: 0 iopslimit: 0
optimumcache[1770]: No immediate fsync: Yes
optimumcache[1771]: Starting OptimumCache monitor
```

To update to version 0.2-6 run:

```
# yum update optimumcache --enablerepo=cloudlinux-updates-testing
```

High CPU Utilization

Once it is detected that OptimumCache overuses CPU, it is useful to check, whether checksums reindexing process is running. When reindexing is running, high CPU usage is ok, as far it will certainly drop down after reindexing finished.

Can be checked in /var/log/messages -

```
# grep Reindexing /var/log/messages
Feb  4 17:00:55 CL-default-2 occtl[2654]: Reindexing started
```

If the last line from the output is not 'Reindexing finished...', than indexing is in progress.

Also, can be checked via command 'occtl --report', watch if PFL_REINDEX_NUM_FILES and PFL_REINDEX_THROUGHPUT_KB identifiers are present in the last series of data:

```
# occtl --report
- Period starts at: 2015-02-04 17:00
  Period Stat:
PFL_ATTACHED:                170318
PFL_CREATED:                  161583
PFL_ERR_BAD_CSUM:              176
PFL_ERR_INODES:                879
PFL_FAILED_TO_ATTACH_PEER:     791
PFL_FAILED_TO_ATTACH_PEER_EBUSY: 791
PFL_INODE_IN:                  406167
PFL_PAGEMIN_FILTERED_OUT:     233418
PFL_PAGEMIN_USED:              136082
PFL_REINDEX_NUM_FILES:        192810
PFL_REINDEX_THROUGHPUT_KB:    2904007
PFL_RESTART:                   1
```

12 Additional Packages

CloudLinux will package additional software needed by hosters for your convenience.

12.1 Git for cPanel

Please, note this package is no longer needed, as since cPanel 11.38, you can install git without any issues on cPanel by running

```
$ yum install git
```

To install [git](#) on cPanel servers

```
$ yum install git-cpanel
```

13 Integration Guide

Here you will find instructions and common techniques used to integrate your software with CloudLinux

13.1 Common Questions

Detect if system is running CloudLinux / CloudLinux kernel:

```
$ uname -r|grep lve
```

if you get an output, it means the system runs CloudLinux kernel. CloudLinux kernels have lve in its name, like: 2.6.32-458.18.1.lve1.2.44.el6.x86_64

Alternatively you can check for presence of file /proc/lve/list

Check if CageFS is enabled (as root):

```
$ /usr/sbin/cagefsctl --cagefs-status
```

Check if CageFS is enabled for particular user (as root):

```
$ /usr/sbin/cagefsctl --user-status _USER_NAME_
```

Check if you are inside CageFS:

Check for presence of file: /var/.cagefs/.cagefs.token

Presence of such file means that you are inside CageFS.

13.2 Displaying CPU, Memory & IO limits

Most control panels choose to display CloudLinux usage & limits to end customers. To simplify that, we lve-stats exports a file that can be easily read & processed by a control panel to display necessary information.

The information is located in the /var/lve/info file. This information is updated every 5 minutes, and contains default limits (first line), as well as usage & limits for all customers. If customer is not present in the file, it means that customer is not active (no scripts were executed recently for the customer), and customer has default limits (so you can display no usage, and default limits in the control panel for that customer).

The data is stored in a form of one line per customer, with coma separated values.

0	user id
1	entry processes
2	entry processes limit
3	CPU
4	CPU limit
5	Virtual Memory
6	Virtual Memory Limit
7	Number of virtual memory faults
8	Number of entry processes faults

9	Physical Memory Limit
10	Physical Memory
11	Number of Physical memory faults
12	Number of processes limit
13	Number of processes
14	Number of processes fault
15	Reserved
16	IO Usage
17	IO Limit

With LVE version 4 (CloudLinux lve0.x) only first 9 parameters are available. You can check the the version by reading first byte of /proc/lve/list

On version 6 all 15 parameters should be available.

There is only 2 LVE versions currently used in production. Future versions might add more fields, but will not alter order of existing fields.

Memory is defined in 4KB pages (so, 1024 would mean 1024 4KB pages, or 4MB)

IO is defined as KB/s

CPU is defined as % of total number of cores on a server

13.3 Integrating LVE Limits with Packages

[lve-utils 1.4+]

CloudLinux can automatically detect most popular control panels, like cPanel -- and allows to set different limits for users in different packages.

This simplifies management -- as you don't have to choose between one limit that fits all your customers on the server, or individual limits for the customers.

If you have a custom made control panel, with your own 'package' implementation, you can still use CloudLinux framework to manage limits for your packages.

To do that, you would need:

1. Implement script that would map users <-> packages
2. Configure lvectl to use your script.

Implementing script

Script can be written in any language, and it has to be executable.

It should accept following arguments:

--list-all prints <userid package> pairs

Output should look like a list of space separate pairs of user linux IDs, and package names.

```
100 package1
101 package1
102 package2
103 package3
```

--userid=id prints package for a user specified

Output should contain package name, like:

```
package1
```

--package="package" prints users for a package specified.

Output should look like a list of user linux IDs.

```
100  
101
```

--list-packages prints list of packages list

Output contains a list of names of packages, like:

```
package1  
package2  
package3
```

Configuring lvectl to use your custom script

Edit file `/etc/sysconfig/cloudlinux`

Edit or modify parameter `CUSTOM_GETPACKAGE_SCRIPT`, and set it to point to your script, like:

`CUSTOM_GETPACKAGE_SCRIPT=/absolute/path/to/your/script`

For script example and please visit the following article: <http://kb.cloudlinux.com/2015/02/integrating-lve-limits-with-packages-for-unsupported-control-panel/>

14 Partner Portal

14.1 IP Reseller Partner UI

To become CloudLinux reseller partner you should first register your account following this link: <https://cln.cloudlinux.com/clweb/login.xhtml> and contact us to apply for your access status.

Once you have got the reseller partner access, in IP Reseller Partner UI you can view and manage IP licenses, billing options, profile details. Here you can track your money balance, licenses count and licenses prices as well as using IP address search to find customers.

Server Section

As soon as you have added funds (*See **Billing Info/Add Funds below***) to your account you can immediately add new licenses for clients. To add license:

1. Enter IP address in **Add IP License** field, choose license type in pull-down menu (CloudLinux or KernelCare) and click **Add license**.

The screenshot displays the CloudLinux Partner UI. At the top left is the CloudLinux logo with the tagline "The Cloud Ready OS". To the right are navigation links: [API], [Servers], [Profile], [Billing Info/Add Funds], and [Logout]. Below the navigation, the account status is shown: "CloudLinux Licenses: 0 Price per License: \$12.0/per month" and "KernelCare Licenses: 0 Price per License: \$2.95/per month". The current balance is "\$100.00" with a link to "(Add Funds)".

Below the balance information is a form to "Add IP License". It includes an input field for the IP address, a dropdown menu for "License type", and an "Add license" button.

Underneath the form is a table with tabs for "CloudLinux", "KernelCare IPs", and "KernelCare Keys". The "CloudLinux" tab is active. The table has a search filter by "Name" and "Find" and "Reset" buttons. The table header includes "ip", "Hostname", and "Added / Last Check In". The table content shows "Current customer have no licenses".

2. To delete license click **Delete** in front of the needed IP address.

Billing Info/Add Funds

To add funds:

1. Click on **Add Funds** near your balance or go to **Billing Info/Add Funds** on the top of the starting page of your account.
2. Click Add to add credit card details, then enter funds amount and click **TopUp** or **Process to**

Checkout to pay via PayPal.

The screenshot shows the CloudLinux Partner Portal interface. At the top, there is a navigation bar with the CloudLinux logo and the tagline "The Cloud Ready OS". To the right of the logo are links for [API], [Servers], [Profile], [Billing Info/Add Funds], and [Logout]. Below the navigation bar, the main heading is "Billing".

The "Billing Info" section contains the text: "You don't have any credit card. Do you want add one?" with an "Add" button below it. At the bottom of this section, it displays "Balance: \$100.00" and an "Invoices" button.

The "Add Funds via Credit Card" section has a form with "Amount: *" and an input field, followed by a "TopUp" button.

The "Add Funds via PayPal" section features the PayPal logo and a "Process to checkout >>" button.

While adding credit card details, you can also choose **Auto add funds** option - the funds amount you choose in pull down menu will be automatically added when your balance is below \$100.

If you choose **Auto repay**, your card will be automatically charged when your balance becomes negative. Minimal charge is \$20 (E.g. for balance -\$15 - you'll be charged at \$20, for balance -\$134.2 - you'll be charged at \$134.2).

Billing

Billing Info

Edit Credit Card

Payment Type	VISA ▼
Name on the card	<input style="width: 95%;" type="text"/>
Card number	<input style="width: 95%;" type="text"/>
Expiration date	1 ▼ 14 ▼
CVV2	<input style="width: 95%;" type="text"/>
<input type="radio"/> Auto add funds When balance below \$100.00	\$500 ▼
<input type="radio"/> Auto repay (more details?)	
<input checked="" type="radio"/> Do not add funds automatically	

Balance: \$100.00

Add Funds via Credit Card

Amount: *

Add Funds via PayPal

Note: If your balance is shown as negative, it means that you have to deposit more funds.

API Section

CloudLinux and KernelCare IP licenses adding and removing is compatible with different hosting and domain management and billing systems and platforms. You can find comprehensive information on all possible CloudLinux modules and plug-ins APIs in API Section.

[\[API\]](#)[\[Servers\]](#)[\[Profile\]](#)[\[Billing Info/Add Funds\]](#)[Logout](#)

Cloudlinux partner API

CloudLinux provides a simple way to add & remove IPs for partners.

REST API	XMLRPC API
Cloudlinux primary remote API interface. API Documentation	Old remote API interface, there no new features will be provided. API Documentation Perl Examples PHP Examples Shell Examples
Cloudlinux plugin for WHMCS CloudLinux & KernalCare For WHMCS (PDF) CloudLinux & KernalCare WHMCS plugin	
Cloudlinux plugin for Blesta CloudLinux & KernalCare For Blesta (PDF) CloudLinux & KernalCare Blesta plugin	

Profile

You can edit your profile information by clicking on **Profile** section. Edit the necessary info and click **Update Account**.

Edit your CloudLinux profile

Login Information	
Login	naugolnyi
E-Mail	<input type="text" value="mnaugolnyi@cloudlinux.com"/>
IPs registration token *	<input type="text" value="uVd1mF7exKvQJhWx"/>
I would like to receive the latest information via email	<input type="checkbox"/>

[Change password](#)

Billing Contact Information	
First name	<input type="text" value="Mykola"/>
Last name	<input type="text" value="Naugolnyi"/>
Billing E-Mail	<input type="text" value="mnaugolnyi@cloudlinux.com"/>

Company Information	
Company *	<input type="text" value="CloudLinux2"/>
Title	<input type="text"/>

Contact Information	
First Name *	<input type="text" value="Mykola"/>
Last Name *	<input type="text" value="Naugolnyi"/>
Address Line 1 *	<input type="text" value="Karpinskogo 6, app1"/>
Address Line 2	<input type="text"/>
Address Line 3	<input type="text"/>
City *	<input type="text" value="Kiev"/>
Country	<input type="text" value="Ukraine"/> ▼
Postal Code *	<input type="text" value="03151"/>
Phone Number *	<input type="text" value="+380961853022"/>
Fax Number	<input type="text"/>

[Update Account](#)

15 Hardware Compatibility

CloudLinux supports all the hardware supported by RHEL/CentOS 6.x, with few exceptions. Exceptions are usually hardware that require binary drivers, and that doesn't have any open source alternatives. At this moment we are aware of only one such case:

Device	Binary Driver	Source
B110i Smart Array RAID controller	hpahcisr	http://h20000.www2.hp.com/bizsupport/TechSupport/Document.jsp?objectID=c01732801
B120i/B320i Smart Array SATA RAID Controller	hpvsa	http://h20565.www2.hp.com/portal/site/hpsc/template.PAGE/public/psi/swdDetails/?javax.portlet.begCacheTok=com.vignette.cachetoken&javax.portlet.endCacheTok=com.vignette.cachetoken&javax.portlet.prp_bd9b6997fbc7fc515f4cf4626f5c8d01=wsrp-navigationalState%3Didx%253D%257CswItem%253DMTX_38e3317bc6af4e7184ee3b3492%257CswEnvOID%253D4103%257CitemLocale%253D%257CswLang%253D%257Cmode%253D%257Caction%253DdriverDocument&javax.portlet.tpst=bd9b6997fbc7fc515f4cf4626f5c8d01&sp4ts.oid=5293148&ac.admitted=1414048178968.876444892.492883150

16 Downloading Documentation

This documentation is available for download in multiple formats:

PDF - <http://docs.cloudlinux.com/cloudlinux.pdf>

EPUB - <http://docs.cloudlinux.com/cloudlinux.epub>

RTF - <http://docs.cloudlinux.com/cloudlinux.rtf>

Index

- C -

customize php.ini settings 89

