
CloudLinux Documentation Documentation

Release latest

Nov 24, 2018

Contents


1	Overview	51
2	Types of Users	53
3	Types of Limits	55
4	What happens when reseller or reseller's end user hits the limit?	57
5	Requirements	59
6	Installation	61
7	How to Disable Reseller Limits	63
8	How to exclude mounts from namespaces for all LVEs	135
9	'<>'__Requirements	193
10	'<>'__Hoster	195
11	'<>'__End User	199
12	'<>'__Hoster	203
13	'<>'__End User	207
14	Installation:	289
15	Command-line Interface	295
16	Yum fails to install Perl rpms coming with OptimumCache	301
17	Uninstalling OptimumCache lasts for too long	303
18	'Failed to attach peer: Invalid argument' appears in syslog	305

Author CloudLinux Inc

Date 2018-10-23

Contents

- *CloudLinux Documentation*
 - *Overview*
 - *Types of Users*
 - *Types of Limits*
 - *What happens when reseller or reseller's end user hits the limit?*
 - *Requirements*
 - *Installation*
 - *How to Disable Reseller Limits*
 - *How to exclude mounts from namespaces for all LVEs*
 - *'<>'__Requirements*
 - *'<>'__Hoster*
 - *'<>'__End User*
 - *'<>'__Hoster*
 - *'<>'__End User*
 - *Installation:*
 - *Command-line Interface*
 - *Yum fails to install Perl rpms coming with OptimumCache*
 - *Uninstalling OptimumCache lasts for too long*
 - *'Failed to attach peer: Invalid argument' appears in syslog*



OPF/cloudlinuxos-docs.png

Installation

- *Converting existing servers*
 - *Advanced Options for cldeploy*
 - *Explanation Of Changes*
- *Installing new servers*
- *CloudLinux OS Images*
 - *Xen Images*

- *Net Install*
- *Installing on H-Sphere Server*
 - *Converting from mod_fastcgi to mod_fcgid*
- *Virtuozzo and OpenVZ*
- *Getting Trial License*
- *Registering CloudLinux Server*
- *CloudLinux on DigitalOcean*
- *CloudLinux on Linode*
- *Servers with LILO boot loader*
- *cPanel EasyApache 4*
- *Uninstalling CloudLinux*

Converting Existing Servers

It is easy to switch server from CentOS 6.x or 7.x to CloudLinux. The process takes a few minutes and replaces just a handful of RPMs.

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

If you have activation key:

```
$ wget https://repo.cloudlinux.com/cloudlinux/sources/cln/cldeploy
$ sh cldeploy -k <activation_key>
```

If you have IP-based license:

```
$ sh cldeploy -i
$ reboot
```

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

The script automatically detects and supports the following control panels: cPanel with EA3, Plesk, DirectAdmin, InterWorx. It will install CloudLinux kernel, *Apache module*, *PAM module*, *command line tools* as well as LVE Manager.

ISPmanager 5 has native support for CloudLinux. To deploy CloudLinux on a server with ISPmanager 5, you would need to purchase CloudLinux license directly from ISPSystems and follow ISPmanager's deployment guide.

Note. If you are converting Hyper-V server, please, make sure you upgrade to the latest CentOS 6.9 or CentOS 7.4 first.

Advanced Options for cldeploy

```
sh 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
                                --serverurl      Use non-default registratio server
                                (default is
```

```
https://xmlrpc.cln.cloudlinux.com/XMLRPC)
```

```
-components-only Install control panel components only
                                --conversion-only Do not install control panel components after
```

converting

```
-hostinglimits Install mod_hostinglimits rpm
                                --skip-kmod-check Skip check for unsupported kmods
-skip-version-check Do not check for script updates
                                --skip-registration Don't register on CLN if already have access to
```

CL repos

The script will install the 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 the following actions:
1.	For cPanel & DirectAdmin:
1.	recompile Apache to install mod_hostinglimits;
2.	install LVE Manager.
2.	For Plesk, ISPManager & InterWorx:
1.	Update httpd and install mod_hostinglimits;
2.	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
```

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 the following actions:

- Backup of original repository settings into `/etc/cl-convert-saved`.

- Backup of RHEL system id into /etc/cl-convert-saved (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:
 - oRPM install for Plesk, ISPManager & InterWorx;
 - oOn Plesk, replaces psa-mod_fcgid* with mod_fcgid;
 - oEasyApache rebuild for cPanel;
 - ocustombuild 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.

Installing new servers

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

Latest stable CloudLinux 7.5 ISO:

x86_64 version: http://repo.cloudlinux.com/cloudlinux/7/iso/x86_64/CloudLinux-DVD-x86_64-7.5.iso

Last Updated: May 14, 2018

Latest stable CloudLinux 6.9 ISO:

x86_64 version: http://repo.cloudlinux.com/cloudlinux/6/iso/x86_64/CloudLinux-6.9-x86_64-DVD.iso

i386 version: <http://repo.cloudlinux.com/cloudlinux/6/iso/i386/CloudLinux-6.9-i386-DVD.iso>

Last Updated: April 6, 2017

Latest stable CloudLinux 5.11 ISO (OBSOLETE):

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

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

CloudLinux OS Images

- OpenStack QEMU/KVM
- VMware
- Google Cloud Engine
- Amazon Web Services
- Alibaba Cloud
- Xen

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 6 Minimal: <http://download.cloudlinux.com/images/cl6-7/cl6-hvm-base.img.tgz>

CloudLinux 7 Minimal: <http://download.cloudlinux.com/images/cl6-7/cl7-hvm-base.img.tgz>

CloudLinux 6 + cPanel: <http://download.cloudlinux.com/images/cl6-7/cl6-hvm-cPanel.img.tgz>

CloudLinux 6 + Parallels Plesk: <http://download.cloudlinux.com/images/cl6-7/cl6-hvm-Plesk.img.tgz>

CloudLinux 6 + DirectAdmin: <http://download.cloudlinux.com/images/cl6-7/cl6-hvm-da.img.tgz>

CloudLinux 7 + DirectAdmin: <http://download.cloudlinux.com/images/cl6-7/cl7-hvm-da.img.tgz>

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 the 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.

Installing on H-Sphere Server

For H-Sphere 3.5+

[Please note, that CageFS and PHP Selector are not supported for H-Sphere]

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`

The 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*

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+]

Step 1:

Download our fcgi.conf file:

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

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 enabled:

- 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/usemodule.phpmode.patch to /hsphere/local/config/scripts/usemodule.phpmode:

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

Step 3:

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

```
$ cp -rp /hsphere/local/config/httpd2/httpd.conf.tpl /hsphere/local/config/httpd2/httpd.conf.tpl.custom
```

Download and apply patch http://repo.cloudlinux.com/cloudlinux/sources/mod_fcgid-hsphere/httpd.conf.tmpl.patch to /hsphere/local/config/httpd2/httpd.conf.tmpl.custom:

```
$ wget http://repo.cloudlinux.com/cloudlinux/sources/mod_fcgid-hsphere/httpd.conf.tmpl.patch
$ patch -fuzz=3 /hsphere/local/config/httpd2/httpd.conf.tmpl.cusom httpd.conf.tmpl.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:

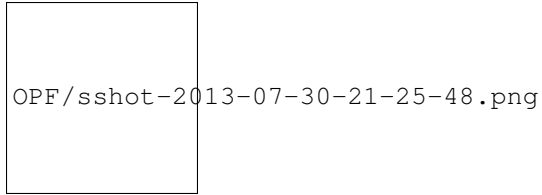
Edit ~httpd2/conf/extra/httpd-hostinglimits.conf and add DenyHandlers, so section will look like:

```
<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:

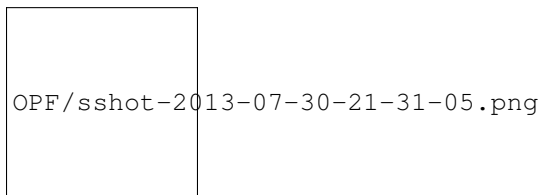
```
apache_version = 2
apacha_fastcgi = yes
apache_status = yes
```



Step 9:

Set PHP configuration to:

php_libphp5 enabled but not default
php_fastcgi5 enabled and is default



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.

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 the following functionality works:

CageFS

PHP Selector

max entry processes

mod_lsapi

MySQL Governor

No other limits work so far.

Installation

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

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

Please make sure you have vzkernel-headers and vzkernel-devel packages installed. If no - install them with yum:

yum install vzkernel-headers vzkernel-devel

```
$ 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.

To make CloudLinux 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: http://docs.cloudlinux.com/index.html?converting_existing_servers.html

CloudLinux license is required for each VZ container.

Note. Some servers require increasing fs.ve-mount-nr on host node, otherwise CageFS will throw errors. On a host node:

1. add “fs.ve-mount-nr = 15000” to /etc/sysctl.conf;
2. apply it with ‘sysctl -p’ command.

In very rare cases the value should be increased higher, up to 50000.

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 will get a key that looks like: 12314-d34463a182fede4f4d7e140f1841bcf2

Use it to register your system or to *convert CentOS server to CloudLinux* server.

Registering CloudLinux Server

To register your server with CloudLinux Network using activation key run:

```
$ 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
```

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
```

CloudLinux on Linode

CloudLinux on Linode KVM

To install CloudLinux 7 on Linode KVM server you should perform the following steps:

1. Deploy CL to your Linode following the steps from this section: http://docs.cloudlinux.com/index.html?converting_existing_servers.html

2. Install grub on your system:

```
yum install grub2
```

3. Add to /etc/default/grub the following parameters:

```
GRUB_TIMEOUT=10
```

```
GRUB_CMDLINE_LINUX="console=ttyS0,19200n8"
```

```
GRUB_DISABLE_LINUX_UUID=true
```

```
GRUB_SERIAL_COMMAND="serial --speed=19200 --unit=0 --word=8 --parity=no --stop=1"
```

4. Update grub config:

```
grub2-mkconfig -o /boot/grub/grub.cfg
```

5. Edit your Linode profile, change the boot settings to “GRUB 2”.

6. Reboot your Linode.

After reboot you will have fully operational CloudLinux 7 system and can proceed with other configuration you need.

CloudLinux on Linode Xen

To install CloudLinux 7 on Linode Xen please perform the following steps:

1. Deploy CL to your Linode following the steps from this section: http://docs.cloudlinux.com/index.html?converting_existing_servers.html

2. Create file /boot/grub/menu.lst with the following content:

```
timeout 5
```

```
title CloudLinux 7.1, $KVERSION
```

```
root (hd0)
```

```
kernel /boot/vmlinuz-$KVERSION root=/dev/xvda ro quiet
```

```
initrd /boot/initramfs-$KVERSION.img
```

where \$KVERSION is the version of installed CL7 kernel.

Please note that you will need to update /boot/grub/menu.lst manually after every kernel update.

3. Switch boot settings to pv-grub-x86_64 and switch off “Auto-configure networking” in Linode settings.

4. Reboot your Linode.

In case if you will migrate to KVM later you will need only switch the boot settings to GRUB 2.

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 <i>deploy2cl</i> 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. <code>uname -r</code> should show something like: 2.6.18-294.8.1.el5.lve0.7.33.
----	---

Migrating to EasyApache 4

Advices and limitations:

- Use cPanel 11.55.999.66(55.999.66) or higher version.
- Hardened EA4 limitations:
ea-php51 and ea-php52 have no PHP-FPM support. Please use mod_lsapi instead.

Follow the instructions at http://docs.cloudlinux.com/index.html?mod_lsapi_installation.html to install and configure mod_lsapi.

CentOS with EasyApache 4

If EasyApache 4 was installed earlier on your CentOS server and you would like to migrate to CloudLinux:

1. Convert server from CentOS 6.x or 7.x to CloudLinux: (http://docs.cloudlinux.com/index.html?converting_existing_servers.html)
2. Restart Apache service.

CentOS without EasyApache 4

If EasyApache 4 was not installed earlier on your CentOS server and you would like to migrate to CloudLinux:

1. Convert server from CentOS 6.x or 7.x to CloudLinux (http://docs.cloudlinux.com/index.html?converting_existing_servers.html)
2. Run:

```
cd ~; wget https://repo.cloudlinux.com/cloudlinux/sources/cloudlinux_ea3_to_ea4; sh cloudlinux_ea3_to_ea4 -convert
```

(Find examples of cloudlinux_ea3_to_ea4 script usage below).

CloudLinux without EasyApache 4

Install EasyApache4 on clean CloudLinux from ISO image or migrate to EasyApache4 on existings CloudLinux servers:

1. Install cPanel.
2. Run:

```
cd ~; wget https://repo.cloudlinux.com/cloudlinux/sources/cloudlinux_ea3_to_ea4; sh cloudlinux_ea3_to_ea4 -convert
```

(Find examples of cloudlinux_ea3_to_ea4 script usage below).

Revert back from EasyApache 4 to EasyApache 3

To migrate back to EA3 for CloudLinux run:

```
cd ~; wget https://repo.cloudlinux.com/cloudlinux/sources/cloudlinux_ea3_to_ea4; sh cloudlinux_ea3_to_ea4 -revert
```

More about cloudlinux_ea3_to_ea4 script

About cloudlinux_ea3_to_ea4 migration script parameters:

cloudlinux_ea3_to_ea4 [ADDITIONS] ACTIONS

Usage:

-h, --help	Print this message
------------	--------------------

Actions (required parameter, shows what should script do):

-c, --convert	Convert EA3 to EA4
-r, --revert	Revert to EA3

Additions (optional parameter, adds to action installation of extra components):

-m, -mod_lsapi	Install mod_lsapi
-p, -mod_passenger	Install alt-mod-passenger
-a, -altphp	Install/Update alt-php

Note. ADDITIONS parameters can't be used without ACTIONS

Examples:

If you want to install EA4 with mod_lsapi and update/install alt-php:

```
sh cloudlinux_ea3_to_ea4 --convert --mod_lsapi --altphp
```

If you want to install EA4 with mod_lsapi, alt_mod_passenger and update/install alt-php:

```
sh cloudlinux_ea3_to_ea4 --convert --mod_lsapi --altphp --mod_passenger
```

To restore EA3 with mod_lsapi:

```
sh cloudlinux_ea3_to_ea4 --revert --mod_lsapi
```

Frequently asked questions (FAQ)

FAQ

•When do we need to call

```
cd ~; wget https://repo.cloudlinux.com/cloudlinux/sources/cloudlinux_ea3_to_ea4;
```

```
sh cloudlinux_ea3_to_ea4 --convert
```

script?

1. Migration from EasyApache 3 to EasyApache 4.

The main difference between EasyApache 3 and EasyApache 4 for CloudLinux is the repositories used for Apache RPM packages. For this reason, we need to use packages from the cl-ea4 repository or cl-ea4-testing beta for EasyApache 4. Running this script we update all native ea-* packages from CloudLinux repository. In this case, non-native packages for Apache include mod_lsapi and alt-mod-passenger (CloudLinux feature). So, if mod_lsapi or alt-mod-passenger (or both) were installed on EasyApache3, the script should be run with additional options as it described on the link https://docs.cloudlinux.com/cpanel_easyapache_4.html.

Also, our script starts cPanel EasyApache 3 migration to EasyApache 4 Process. Read more about Profile changes, Apache changes, PHP changes on the link <https://documentation.cpanel.net/display/EA4/The+EasyApache+3+to+EasyApache+4+Migration+Process>

2. Migration from EasyApache 4 CentOS to EasyApache 4 CloudLinux.

When cPanel is installed with EasyApache 4 on a clean CloudLinux (or it was CentOS converted to CloudLinux), the installation of the ea-* packages comes from the EA4 cPanel repository. Most packages from the EA4 cPanel repository are not compatible with CloudLinux packages and this can lead to various errors. For this reason, we need to run this script to update the ea-* packages from the CloudLinux repository.

If there was a need to return back EasyApache 4 packages from the EA4 cPanel repository, we need to run:

```
cd ~; wget https://repo.cloudlinux.com/cloudlinux/sources/cloudlinux_ea3_to_ea4;
```

```
sh cloudlinux_ea3_to_ea4 --restore-cpanel-ea4-repo
```

•When do we need to call

```
cd ~; wget https://repo.cloudlinux.com/cloudlinux/sources/cloudlinux_ea3_to_ea4;
```

```
sh cloudlinux_ea3_to_ea4 --revert
```

script?

1. Reverting back to EasyApache 3.

Revert back is possible only if EasyApache 3 was previously installed, and then converted to EasyApache 4. If cPanel was originally installed with EasyApache 4, there is no way to convert to EasyApache 3.

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.

The following actions will be taken:

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

At the end, the 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 https://repo.cloudlinux.com/cloudlinux/sources/cln/cldeploy
$ sh cldeploy -c
```

Now you have converted back to CentOS and it is the time to install kernel.

To delete CloudLinux kernel run:

```
rpm -e --nodeps kernel-2.6.32-673.26.1.lve1.4.27.el6.x86_64
```

To install new CentOS kernel once you deleted CloudLinux kernel, type yum install kernel

If yum says that the latest kernel is already installed, it is OK.

Please check your bootloader configuration before rebooting the system.

To remove unused kmods and lve libs run:

```
yum remove lve kmod*lve*
```

Kernel package and related LVE packages should be deleted and the required kernel will be installed.

Before the reboot the following command should be executed for restoring Apache and httpd.conf without mod_hostinglimits:

For EasyApache 3:

```
/scripts/easyapache --build
```

For EasyApache 4:

```
/usr/local/bin/ea_install_profile --install /etc/cpanel/ea4/profiles/cpanel/default.json
```

Please note that some of the packages from CloudLinux repo will still be present. They are the 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.

Limits

CloudLinux has support for the following limits:

Limits	Units	De- fault Value	Description	Supported Kernels / OS
<i>SPEED</i> <#speed _lim- its.html> —	% of a core, or HZ	100%	CPU speed limit, relative to a single core, or specified in HZ (portable across CPUs)	all
CPU [depre- cated]	% of CPU	25%	CPU Limit (smallest of CPU & NCPU is used)	all
<i>NCPU</i> [deprecated]	num- ber of cores	1 CORE	Max number of cores (smallest of CPU & NCPU used)	all
<i>PMEM</i> <#memory _lim- its.html> —	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+ ker- nel, CL6 and CL7
<i>VMEM</i> <#memory _lim- its.html> —	KB	0	Virtual memory limit (VSZ field in ps/VIRT in top)	all
<i>IO</i>	KB/sec	1024KB	IO throughput - combines both read & write operations	CL7, CL6 lve1.1.9+ ker- nel, CL5 hybrid kernel
IOPS [lve1.3+]	Oper- ations per second	1024	Restricts total number of read/write operations per second.	CL7, CL6 and CL5 hybrid ker- nels lve1.3+
<i>NPROC</i> _	number	100	Max number of processes within LVE	CL5 hybrid kernel, CL5 lve1.x+ ker- nel, CL6 and CL7
<i>EP</i>	number	20	Limit on entry processes. Usually represents max number of concurrent connections to apache dynamic scripts as well as SSH and cron jobs running simultaneously .	all

Note. It is always better to disable VMEM limits (set them to 0) in your system at all because they are deprecated in CloudLinux 6/7 system and are causing unexpected issues.

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=512MB

VMEM=0

IO=1024KB/s

IOPS=1024

NPROC=100

EP=20

High End Hosting Account

SPEED=200%

PMEM=1GB

VMEM=0

IO=4096KB/s

IOPS=1024

NPROC=100

EP=40

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 the 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, then `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, then the site will start responding slower.

If the site is limited by memory or number of processes limits, then the user will receive 500 or 503 errors that server cannot execute the script.

Checking if LVE is installed

To use LVE you should 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 run:

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

Starting from kernels `lve1.4.x` `iolimits` module is a part of `kmod-lve` and could not be used separately.

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   ICPU   IIO   CPU   MEM   IO IMEM   IEP   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.

Command-line Tools

- *lvectl*

- *lveps*

- *lvetop*

- *cldetect*

- *lve-stats*

- o Storing statistics in MySQL

- o Storing statistics in PostgreSQL

- o Compacting in multi-server settings

- *lve-stats 2*

- o Installation

- o Configuration

- o Command Line Tools

*lveinfo**lvechart**dbgovchart**lve-read-snapshot**lve-create-db**oPlugins**oCreating a Plugin for LVE Stats 2**Introduction**Server Plugin Arrangement**Plugin Configuration**Types of Plugins**Examples of Plugins**Collector**Analyzer**Persistor**Notifier**o/var/lve/info file**oTroubleshooting*

lvectl

lvectl is the primary tool for LVE management. To use it, you have to have administrator access. lvectl is a 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	set parameters for a LVE and/or create a LVE using username instead of ID
	list	list loaded LVEs
	list-user	list loaded LVEs, display username instead of user id
	limits	show limits for loaded LVEs
	delete	delete LVE and set configuration for that LVE to defaults
	delete-user	delete LVE and set configuration for that user to defaults
	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

Continued on next page

Table 1 – continued from previous page

	package-delete	delete LVE parameters for a package
	paneluserslimits	show current user's limits for control panel
	limit	limit PID into specified LVE. Parameters PID LVE_ID
	release	release PID from LVE. Parameters PID
	set-binary	add binary to be run inside LVE on execution
	del-binary	remove binary from being run inside LVE on execution
	list-binaries	list all binaries to be run inside LVE on execution
	load-binaries	load binaries (used on startup) from config file
	reload-binaries	re-load list of binaries from config file
	help (-h)	show this message
	version (-v)	version number
	lve-version	LVE version number
	set-reseller	create LVE container and set LVE parameters for a reseller
	set-reseller-default	set default limits for resellers users
Options:		
	-cpu=N	limit CPU usage; (deprecated. Use -speed)
	-speed=N%	limit CPU usage in percentage; 100% is one core
	-speed=Nmhz\ghz	limit CPU usage in mhz\ghz
	-ncpu=N	limit VCPU usage (deprecated)
	-io=N	define io limits (KB/s)
	-nproc=N	limit number of processes
	-pmem=N	limit physical memory usage for applications inside LVE
	-iops=N	limit io per second
	-mem=N	mem alias for vmem (deprecated)
	-vmem=N	limit virtual memory for applications inside LVE
	-maxEntryProcs=N	limit number of entry processes
	-save	save configuration settings (use with set) (deprecated)
	-save-all-parameters	save all parameters even if they match with defaults settings
	-json	returns result of command json formatted
	-unlimited	set all limits to unlimited
	-save-username	save username in the config file. This parameter is used in conjunction with set-user

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
```

lveps

lveps tool shows information about running LVEs, processes and threads belonging to them, CPU/memory/IO usage consumed by LVEs and their individual processes/threads. LVE is only reported if it is considered active (at least one thread belongs to that LVE or was running during measurement in dynamic mode).

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.

CPU	The number of seconds LVE/process/thread has been running (each CPU/core is counted separately), or the average CPU load (100% is all CPU resources) if used with -d.
MEM	The number of megabytes of resident memory in use by LVE/process/thread (shared memory is not included).
IO	The number of kilobytes read and written in sum by LVE, or kb/sec if used with -d.
ID	LVE ID or username.
EP	The number of entry processes inside LVE.
COM	Command name for this process.
PID	PID of the process.
PNO	The number of processes belonging to the LVE.
TID	TID of the thread.
TNO	The number of threads belonging to the LVE.
DO	The number of disk operations belonging to the LVE from the time it was created.
DT	Total amount of disk transfer in megabytes from LVE creation time.
IOPS	The number of I/O operations per second

lvetop

lvetop utility allows to monitor LVE usage:

ID	EP	PNO	TNO	SPEED	MEM	IO	IPOS
testuser1	0	1	1	0%	7	0	0
testuser2	0	0	0	5%	0	3	0
testuser3	1	2	2	0%	102	2727	0
testuser4	0	1	1	0%	12	84	1
testuser5	0	2	2	1%	52	0	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
IOPS number of read/write operations per second

cldetect

[lve-utils 1.2-10+]

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

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

cldetect -h

-h -help	show this message
-detect-cp	prints control panel and its version (CP_NAME,CP_VERSION)
-detect-cp-full	prints control panel, version and panel specific data (CP_NAME,CP_VERSION,...).
Specific data: for ISP Manager5 - Master/Node	
-detect-cp-nameonly	prints control panel name (CP_NAME)
-get-admin-email	prints control panel admin email (CP_ADMIN_EMAIL)
-cxs-installed	check if CXS is installed. Returns 0 if installed, 1 otherwise
-cpanel-suphp-enabled	check if suPHP is enabled in cPanel.Returns 0 if enabled, 1 otherwise
-detect-litespeed	check if LiteSpeed is installed. Returns 0 if installed, 1 otherwise
-detect-postgresql	check if PostGreSQL is installed. Returns 0 if installed, 1 otherwise
-print-apache-gid	prints current apache gid
-print-da-admin	prints DirectAdmin admin user
-set-securelinks-gid	changes /etc/sysctl.conf if apache gid != 48 (default)
-set-nagios	do some adjustments to make nagios work correctly if it's installed. Called as a part of "--setup-supergids"
-setup-supergids	do some adjustments to make special users/software (nagios, cPanel's mailman) work correctly if it is installed to the system
-cl-setup	check if CloudLinux is installing. Returns 0 if installing, 1 otherwise
-update-license	updates license
-update-new-key	updates license with new key
-check-license	:check license. Returns OK if license is not older than 3 days, error message otherwise
-q	:check license. Returns 0 if license is not older than 3 days, 1 otherwise
-no-valid-license-screen	Returns no valid license found screen.

`--license-out-of-date-email` Returns License out of Date Email.

`--check-openvz` Returns enviroment id.

clsupergid auto-configuration

Each time lve-utils package is installed or upgraded it does some automatic system re-configuration to make some software (like nagios) work correctly, if it's installed, by calling `cldetect --setup-supergids` command.

Starting from lve-utils 3.0-21 a behaviour of `cldetect --set-nagios` (now, it's a part of `cldetect --setup-supergids`) command slightly changed.

	Old behavior	New behavior
If <code>fs.proc_super_gid</code> is 0 (which means it's not configured) or it's set to some GID that doesn't exist in the system.	Command will set <code>sysctl fs.proc_super_gid</code> to point to Nagios GID.	Command will create special <code>clsupergid</code> group, setup <code>sysctl fs.proc_super_gid</code> to point to it's GID and add Nagios user to this group.

If `fs.proc_super_gid` was configured by an admin to some existing group, the command will just add Nagios user to this group.

SPEED Limits

[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.

Note. We strongly recommend setting CPU speed limits not less than 100%. As such limits cause CPU context switching which leads to increased %sys.

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.

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.

Note. It is recommended to disable VMEM limits (set them to 0) in your system at all because they are deprecated in CloudLinux 6 and 7 system and can cause unexpected issues.

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 then 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.

Troubleshooting

Checking personal users disk cache (If lveinfo shows memory usage but there are no processes there)

If you see no processes under some user, but lve manager keeps telling it is using some memory, then most probably memory is taken by users disk cache. To check personal users disk cache (if lveinfo shows memory usage but not processes there):

```
cat /proc/bc/XXX/meminfo
```

...

Cached: 67300 kB

...

where XXX is user id, could be taken with:

id username

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.

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.

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 know as 'Apache concurrent connections' limit.

The process enter's into LVE when there is a new HTTP request for CGI/PHP.

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.

Number of Processes Limit

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

Network Traffic Bandwidth Control and Accounting System

[Requires kernel lve1.4.4.el6 or higher, or lve1.4.56.el7 or higher]

Network traffic bandwidth control and accounting systems in CloudLinux 6 allows for each LVE container:

- Limiting outgoing network traffic bandwidth
- Accounting incoming and outgoing network traffic

The system supports IPv4 only protocol.

How to limit outgoing network traffic

All outgoing IP packets generated inside LVE container and marked with LVE identifier. Traffic control utility tc from iproute2 package uses this marker to set required bandwidth.

Note. CloudLinux doesn't limit the network traffic itself, it only marks IP packets with specific LVE id.

Example 1:

1. We create class with HTB qdiscs and rate 10kbit:

```
tc qdisc add dev eth1 root handle 1: htb
```

```
tc class add dev eth1 parent 1: classid 1:1 htb rate 10kbit
```

2. All packets marked with LVE id will be processed by class 1:1 (rate 10kbit).

```
tc filter add dev eth1 parent 1: handle 2121 fw flowid 1:1
```

Example 2:

1. As an example we create class with HTB qdiscs and rate 100mbit and class 1:10 will be used by default:

```
tc qdisc add dev eth3 root handle 1: htb default 10
```

```
tc class add dev eth3 parent 1: classid 1:1 htb rate 100mbit
```

2. For class 1:1 we create two branches with rate 5 mbit and 10 kbit accordingly, with classid 1:10 and 1:20.

```
tc class add dev eth3 parent 1:1 classid 1:10 htb rate 5mbit
```

```
tc class add dev eth3 parent 1:1 classid 1:20 htb rate 10kbit
```

3. All packets marked with LVE id=2121 are processed by 10 kbit class.

```
tc filter add dev eth3 protocol ip parent 1: prio 1 handle 2121 fw flowid 1:20
```

More info about tc and its syntax can be found on the link <http://tldp.org/HOWTO/Traffic-Control-HOWTO/index.html>

Traffic accounting

Traffic accounting is performed for each LVE container. Network statistics is collected at /proc/lve/list file. Network-related data found at fields:

- 1.lNETO - output traffic limit by volume, equals 0*
- 2.lNETI - input traffic limit by volume, equals 0*
- 3.NETO - current outgoing traffic value
- 4.NETI - current incoming traffic value

The data is also collected at /proc/lve/per-lve/<id>/net_stat, where id is an LVE container identifier. net_stat file contains 4 values in one row:

- 1.Outgoing traffic limit by volume, equals 0*

2.Incoming traffic limit by volume, equals 0*

3.current outgoing traffic value

4.current incoming traffic value

Note. The current version of CloudLinux network control system doesn't limit network traffic volume for a specific period of time (for example 3GB per day), it limits only network bandwidth.

Note. Network limits are supported only for processes inside LVE. By default it does not limit static content, but only PHP/cgi scripts processed by Apache and processes launched over ssh etc.

Compatibility Matrix

Web Server / PHP	CPU	Virtual & Physical Memory	EP	NPROC	IO	CageFS	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	Yes3	Yes	Yes	Yes	Yes	Yes3	No
Apache / mod_php	Yes	No	Yes	Yes	Yes	no	No
Apache / mod_ruid 2	Yes	No	Yes	Yes	Yes	no	No
Apache / MPM ITK	Yes	No	Yes	Yes	Yes	Yes1	No
LiteSpeed	Yes	Yes2	Yes	Yes	Yes	Yes	Yes
NGINX / PHP-FPM	Yes3	Yes	No	Yes	Yes	Yes	No
SSH	Yes	Yes	Yes	Yes	Yes	Yes3	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 7 and CloudLinux 6 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>

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

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.

For preventing cases when user enters under usual user (using ssh) and then tries to enter as super user (via sudo or su) - pam_sulve was created, which tries to enter to LVE=1 and leaves it right away. If action fails, user gets message:

!!!! WARNING: YOU ARE INSIDE LVE !!!!

To check if pam_sulve is enabled on the server:

```
grep pam_sulve.so /etc/pam.d/*
```

should not be empty.

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
```

Switches:

-f - force namespace

-n - without namespace

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-seculrelve12.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

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 the ID of the 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 looks as follows:

•	LVEId
---	-------

•	LVEUser
---	---------

•	SuexecUserGroup
---	-----------------

•suPHP_UserGroup

•	RUidGid
---	---------

•	AssignUserID
---	--------------

Note. 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>
```

Additional notes

mod_hostinglimits (since version 1.0-22) supports min-uid - cagefsctl --set-min-uid=600. Min UID is read on Apache start/restart and stored in the memory during apache runtime. If min UID has changed, you should restart Apache for mod_hostinglimits applying new min UID value. Full min UID is supported only with APR.

The following message should appear: [notice] mod_hostinglimits: found apr extension version 3. This means that the correct APR is installed with mod_hostinglimits.

mod_hostinglimist has variable for Apache CustomLog format string - %{LVE_ID}y. How to use:

LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" req for lve %{LVE_ID}y" combined

shows in access_log the following info:

```
*.*.* - - [09/Apr/2015:07:17:06 -0400] "GET /1.php HTTP/1.1" 200 43435 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20100101 Firefox/24.0" req for lve 500
```

```
*.*.* - - [09/Apr/2015:07:17:06 -0400] "GET /1.php?=/PHPE9568F34-D428-11d2-A769-00AA001ACF42 HTTP/1.1" 200 2524 "*****/1.php" "Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20100101 Firefox/24.0" req for lve 500
```

```
*.*.* - - [09/Apr/2015:07:17:06 -0400] "GET /1.php?=/PHPE9568F35-D428-11d2-A769-00AA001ACF42 HTTP/1.1" 200 2146 "*****/1.php" "Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20100101 Firefox/24.0" req for lve 500
```

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

| |

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

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

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

Directives

SecureLinks

De- scrip- tion:	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
Syn- tax:	SecureLinks On
De- fault:	SecureLinks Off
Con- text:	server config

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

Example:

SecureLinks On

SkipErrors

Description:	Allow apache to continue if LVE is not available
Syntax:	SkipErrors On
Default:	SkipErrors On
Context:	server config

Prevents Apache from exiting 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 a 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 the option enabled, group ID will be used instead of a user ID. Apache will display the 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 a compatible apr library is not found, the following error message will be 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:	LERetryAfter MINUTES
Default:	240 minutes
Context:	directory config

Specifies interval for Retry-After header. The Retry-After response-header field can be used 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 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 the following regexp: `/home/([^\s]*)/`. Custom regexp can be specified in LVEPathRegexp.

In OWNER mode, the owner of the file is used as an 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 username via path.

Example:

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

LVELimitRecheckTimeout

Description:	Timeout in milliseconds, a site will return EP without lve_enter for LA decreasing after this time
Syntax:	LVELimitRecheckTimeout number
Default:	0
Context:	httpd.conf, virtualhost

Example:

LVELimitRecheckTimeout 1000

LVEUse429

Description:	Use 429 error code as code returned on max entry limits (on/off).
Syntax:	LVEUse429 on
Default:	off
Context:	httpd.conf, virtualhost

Example:

LVEUse429 on

Available for RPM based panels, EasyApache 4 and DirectAdmin.

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/da/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 and 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 the last unsuccessful attempt to connect.

Example:

LVERedisTimeout 120

cPanel/WHM JSON API

CloudLinux offers JSON API for *lvectl* via WHM. You can access it using the following URL:

https://IP:2087/cpsess_YOURTOKEN/cgi/CloudLinux.cgi?cgiaction=jsonhandler&handler=list

The output will look as follows:

```
{ "data": [ { "ID": "default", "CPU": "30", "NCPUs": "1", "PMEM": "1024M", "VMEM": "1024M", "EP": "28", "NPROC": "0", "IO": "2048" } ] }
```

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&speed=30%&io=2048

https://IP:2087/cpsess_YOURTOKEN/cgi/CloudLinux.cgi?cgiaction=jsonhandler&handler=set&lveid=500&speed=300Mhz&io=2048

https://IP:2087/cpsess_YOURTOKEN/cgi/CloudLinux.cgi?cgiaction=jsonhandler&handler=set&lveid=500&speed=3Ghz&io=2048

[Note that speed limit can be specified in several units of measure - %, MHz, GHz. The figures will be different according to the unit of measure.]

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&speed=30%&io=2048

For commands like `apply all`, `destroy all`, use:

`handler=apply-all`

`handler=destroy-all`

You can use the following commands that allow to specify user name instead of user ID:

<code>set-user</code>	Set parameters for a LVE and/or create a LVE using username instead of ID.
<code>list-user</code>	List loaded LVEs, display username instead of user ID.
<code>delete-user</code>	Delete LVE and set configuration for that user to defaults.

If the limits for users are set with cPanel LVE Extension, then turnkey billing solutions can be applied (e.g. WHMCS).

cPanel LVE Extension

[LVE Manager 1.0-9.8+]

cPanel LVE Extension allows to control LVE limits for packages via cPanel hosting packages control interface and via cPanel WHM API. It simplifies integration with existing billing systems for cPanel (like WHMCS for example).

Add Package Extension

To add LVE Settings to standard cPanel package, go to Packages and choose Add a Package.

Note. You can find the information on how to add a package in official cPanel documentation on the link:

<https://documentation.cpanel.net/display/ALD/Add+a+Package>

live-extension_01

Tick LVE Settings checkbox in the bottom of the page to open LVE Settings form.

live-extension_02

You can specify the following options:

Note that your changes to LVE Settings will appear in the system after a little while.

Speed Settings	Maximum CPU usage for an account. Note: •Must be in range 1 - 100 (but obligatory > 0) if old format is used; use % or Mhz\Ghz to set CPU limit as speed; •Type “DEFAULT” to use default value.
Memory Settings	Pmem - Maximum physical memory usage for an account. Vmem - Maximum virtual memory usage for an account. Note: •Must be a positive number. Postfix allowed only in [KGMT]. •Type “DEFAULT” to use default value. •Type “0” for unlimited resource.
Max entry proc Settings	Maximum number of entry processes (concurrent connections) for an account. Note: •Must be a positive number. •Type “DEFAULT” to use default value. •Type “0” for unlimited resource.
Nproc Settings	Maximum number of processes usage for an account. Note: •Must be a positive number. •Type “DEFAULT” to use default value. •Type “0” for unlimited resource.
IO Settings	Maximum I/O (input/output) usage speed for an account. Is measured in Kb/s. Note: •Must be a positive number. •Type “DEFAULT” to use default value. •Type “0” for unlimited resource.
IOPS Settings	Maximum IOPS (input/output operations per second) usage for an account. Note: •Must be a positive number. •Type “DEFAULT” to use default value. •Type “0” to unlimited resource.

live-extension_03

Click Add to apply your changes.

Edit Package Extensions

You can edit limits in any convenient way for you - in Edit a Package section, in LVE Manager or even via WHM API.

Edit a Package

To edit package extensions, choose Packages and click Edit a Package. Choose a package from the Package list and click Edit.

live-extension_04

LVE Manager

To edit package extensions in LVE Manager, in Server Configuration choose CloudLinux LVE Manager. Open Packages tab and click pencil (edit) icon.

live-extension_05

WHM API

To learn how to work with package extensions limits using WHM API, please read the official cPanel documentation:

<https://documentation.cpanel.net/display/SDK/Guide+to+Package+Extensions+-+Data+Behavior+and+Changes>

Note. LVE Package extension does not allow to control LVE limits for reseller packages. Even though LVE limits for reseller packages are displayed in Edit Package menu and their values can be changed, no changes will be applied (will be ignored).

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
```

cPanel LVE Manager

cPanel LVE Manager Administrator interface allows monitoring and managing limits for hosts end users, managing packages and monitoring statistics.

Administrator credentials allow controlling limits for host users.

Log in as administrator to get access to the following functionality:

- Current usage tab - allows monitoring users resource usage at the moment;
- Users tab with the list of all users allows viewing and managing all the users limits;
- Statistics tab displays the statistics of resource usage for proper timeframe or proper users;
- Options tab - allows setting LVE Faults email notifications for users;
- Packages allows managing packages limits;
- Selector tab.

Current usage

1. Choose Current usage tab to monitor users resource usage at the moment displayed in the table.

Current usage table provides the information on the usage of Speed, memory, IO, IOPS, Number of Processes, and Entry Processes.

Resource usage values are being refreshed every 10 seconds which is set in Auto-refresh field. You can refresh the table manually by clicking Refresh now or you can freeze the values by clicking pause button. Usage values will not change until the next manual refresh.

Tick Hide MySQL usage checkbox to hide the information on MySQL usage.

To expand the list of users click on the number above and in the dropdown choose the number of user to be displayed on the page.

!man_01!

Users

Choose Users tab to view the list of all users of the system and manage their limits.

Click Filter by to apply filters. The following filters available in the dropdown:

- Username.
- Domain.
- LVE ID.

!man_02!

Actions column:

Click on a pencil icon in Actions column to edit a proper user limits.

•Set proper LVE values:

o SPEED

o PMEM

o VMEM

o EP

o IO

o IOPS

o NPROC

o INODES

!man_03!

!man_04!

Click Save to apply changes or Cancel to close the window.

Statistics

Choose Statistics tab to view hosts users resource usage statistics.

The following parameters are displayed in the statistics table:

- CPU usage per user;
- PMEM usage per user;
- VMEM usage per user;
- IO (in Kb/sec per user).

Statistics table can be filtered by:

- Timeframe - to view the statistics for a proper period;
- Limit ID - to view a proper limit type usage only;
- Top LVEs - to view top used limits only;
- LVE approaching limit - to view the limits that are approaching

maximum allocated value;

- Fault LVE - the limits that have reached the maximum value.

lman_05l

Options Tab

An administrator can set email notifications for users and resellers in cases of limits faults. Choose Options tab to manage LVE Faults email notifications.

In LVE Faults email notifications section check proper checkboxes to set the required type of notification:

- Notify me on users faults - to receive notifications on users LVE faults;
- Notify customers - to allow hosts users receiving notifications on their LVE faults;
- Notify me when I hit my limits - to receive notifications on LVE faults.

In Faults to include section check proper checkboxes to include proper limits to the notifications:

- SPEED - include speed limit fault to the notification;
- IO - include I/O limit fault info to the notification;
- IOPS - include IOPS limit fault info to the notification;
- Memory - include Memory limit fault info to the notification;
- Concurrent connections - include concurrent connections limit fault

info to the notification.

In Minimum number of Faults to notify section enter proper number of faults required for the notification to be sent for:

Me - for an administrator;

User - for a User;

Set the frequency of email notifications sending in Notify me every.. hours/days section.

Click Save to apply changes.

lveman_08l

lveman_09l

Packages Tab

Packages tab allows setting the limits for as many users as you need by editing packages of proper limits. Each account belonging to a proper package adheres to those limits.

Choose Packages tab to view and modify:

- limits for hosts user's packages (Created by Admin);
- limits for reseller's packages (Created by Admin).

lman_06l

To modify package limits click on a pencil icon in Action column in a proper package row. The following limits for this package are available for setting:

- SPEED in percent (%);
- Virtual memory (VMEM) (can be set as unlimited by setting 0);
- Physical memory (PMEM) (can be set as unlimited by setting 0);

- Concurrent connections (EP);
- Number of processes (NPROC) (can be set as unlimited by setting 0);
- IOPS limits;
- I/O limits (IO) (can be set as unlimited by setting 0);
- INODES soft;
- INODES hard.

When limits are set click Save to apply changes or Cancel to close the window.

Selector tab

Selector tab allows controlling PHP Selector settings.

In Selector is section choose Enabled or Disabled from dropdown list to enable or disable PHP Selector.

In Default PHP version choose a proper PHP version or Native from dropdown list to apply.

In Supported versions choose required PHP versions to support.

Choose default modules from the list for a proper PHP version or for native.

[llveman_092|](#)

[llveman_093|](#)

LVE Manager Options

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
```

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

Server Processes Snapshots

In case when a CloudLinux user hits LVE limits, appropriate faults are generated and *lvestats* package generates Server processes snapshot. Snapshot is a list of running applications and a list of running MySQL queries right after the faults happened.

Snapshots allow users to investigate the reason of account hitting its limits. Several snapshots are generated for each incident. An incident is a state when faults are generated in a close time period. The time period is configurable. By default, if faults are generated in 300 seconds time period, we consider them as a single incident.

The snapshot configuration options are available in

```
/etc/sysconfig/lvestats.config/SnapshotSaver.cfg
```

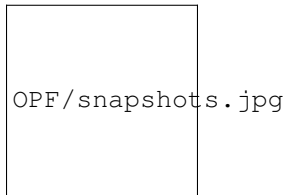
period_between_incidents = 300, by default, time in seconds

snapshots_per_minute = 2, by default, maximum number of snapshots per minute

max_snapshots_per_incident = 10, by default, maximum number of snapshots for an incident

To access Snapshots perform the following steps:

1. Go to cPanel interface, and select “CPU and Concurrent Connection Usage” in paper_latern theme:



2. Click the Snapshots in paper_latern theme:



3. Select a date:



4. Select an appropriate Snapshot in the combobox:



NOTE: The list of processes in a snapshot is close but not similar to the real processes list when faults were generated. It happens because of delay when the faults are happened and the snapshot is taken by the system.

The list of MySQL queries is an output of a query:

```
SELECT command, time, info FROM information_schema.processlist
```

```
WHERE user = '%username';
```

LVE Plugins Branding

[Requires LVE Manager 2.0-33+]

It is possible to apply branding to the LVE Plugins in cPanel end users' interface. To brand the cPanel end users' interface please do the following:

- Create a script that will patch LVE Manager files (with branding data, for example, image and logo) after every update of lvemanager rpm package;
- Locate this script in `/usr/share/l.v.e-manager/branding_script`;
- Make this script executable by running the command:

```
chmod a+x /usr/share/l.v.e-manager/branding_script
```

When done, the branding script will be executed while every update of lvemanager package and all branding changes will be applied in the end user's interface.

Note. Modifying the LVE Manager WHM plugin (`/usr/local/cpanel/whostmgr/docroot/cgi/CloudLinux.cgi`) via `branding_script` is not allowed.

User Message for PHP version

Since version 1.0-4 LVE Manager acquired a feature of adding user messages to PHP versions*. To add a message, you should create a file in `/opt/alt/phpXX/name_modifier` with a message that you want to be shown to a user.

For example, if you need to add the following message "Don't use this php version" to PHP version 4.4, you should create the following file:

`/opt/alt/php44/name_modifier`:

```
echo 'Don't use this php version' > /opt/alt/php44/name_modifier
```

As a result LVE Manager will automatically pick up this message and will show it in web-interface to administrator (Figure 1.1 for cPanel, Figure 1.2 for DirectAdmin) and to user (Figure 2.1 for cPanel, Figure 2.2 for DirectAdmin). You can add messages to other PHP versions this way as well.



Figure 1.1

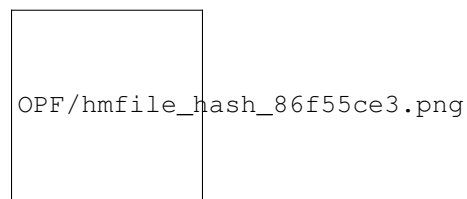


Figure 1.2

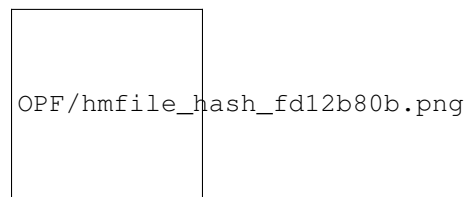


Figure 2.1

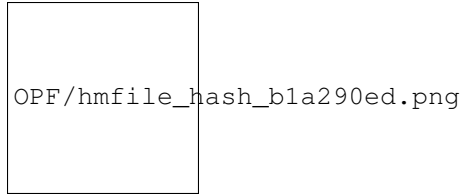


Figure 2.2

*For cPanel and DirectAdmin only.

Reseller Limits

CHAPTER 1

Overview

=

Reseller limits is a feature that allows hosters to set limits for the resources each reseller can operate with. Hosters also provide controls to the reseller on what resources each reseller's end user will have. Reseller limits set by a hoster limit the total amount of resources resellers' end users can consume altogether.

When a hoster has set reseller limits for the particular reseller he provides the reseller with an ability to set limits for his end users within the Reseller Interface.

CHAPTER 2

Types of Users

=

Starting from the version 3.0-18 LVE Manager operates with four types of users and their resource usage limits.

The types of users are as follows:

- End User is a type of user that purchases hosting directly from a hoster and uses it for his own purposes;
- Reseller is a type of user that buys hosting from a hoster and resells it to his end users;
- Reseller's End User is a type of user that purchases hosting from a reseller and uses it for his own purposes.
- Reseller's End User (no Reseller limit) is a type of user that purchases hosting from a reseller and uses it for his own purposes but does not have limits set by a reseller. These limits are set by the hoster.

CHAPTER 3

Types of Limits

=

See the comparison Table with types of limits.

Limits	Reseller limits	Reseller's end user limits	Hoster's end user limits
<i>SPEED</i>	Yes	Yes	Yes
<i>PMEM</i>	Yes	Yes	Yes
<i>IO</i>	Yes	Yes	Yes
<i>IOPS Limits</i>	Yes	Yes	Yes
<i>EP</i>	Yes	Yes	Yes
<i>NPROC</i>	Yes	Yes	Yes
Inodes	Yes (default for all users)	No	Yes
MySQL Limits	Yes (supported only for MySQL Governor ALL mode)	Yes (supported only for MySQL Governor ALL mode)	Yes

=

What happens when reseller or reseller's end user hits the limit?

=

Please note that Reseller is a virtual entity. So, he cannot hit the limit. There is reseller's end user with the same name as reseller. This end user is limited as any other reseller's end user. When hoster sets Reseller limits he limits the group of resellers' end users including reseller's end user with the same name as the reseller.

- Reseller's end user can hit reseller limit when end user's limit is bigger than reseller's limit. In such case end user will be limited by reseller limit.
- Reseller limit can be hit when all resellers' end users in total use as much resources as reseller limit.
- Reseller's end user can hit his limit when end user limit is lower than reseller limit. In such case end user will be limited by his limit.

Installation and Requirements

CHAPTER 5

Requirements

=

Reseller Limits are only supported in kernel starting with the version 3.10.0-714.10.2.lve1.5.3.el7 for CloudLinux 7 kernel and 3.10.0-714.10.2.lve1.5.3.el6h for CloudLinux 6 Hybrid kernel.

Please note, that if you are using CloudLinux 6 kernel you would have to migrate to CloudLinux 6 Hybrid kernel first in order to be able to use new Reseller Limits functionality.

=

=

Use the detailed instruction below:

1.Install CloudLinux 7 or CloudLinux 6 Hybrid on a new server. Follow the instructions described [here](#). Or you can convert your CentOS 6.x or CentOS 7.x system to CloudLinux 6 or CloudLinux 7 respectively. To do this, follow the instructions described on the [link](#).

2.If you have installed the CloudLinux 6, please convert it to the CloudLinux 6 Hybrid Kernel. Follow the instructions described [here](#).

3.Install LVE Manager with Reseller Limit support or update it up to version 3.0-18 (or later) by running the following commands:

```
yum install kernel lve cagefs lvemanager lve-utils lve-stats --disableexcludes=main
yum update
reboot
```

For CloudLinux 6 Hybrid Kernel with Reseller Limit support, please run the following commands:

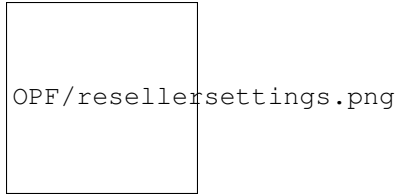
```
yum install kernel lve cagefs lvemanager lve-utils lve-stats --disableexcludes=main
yum update
reboot
```

How to Enable and Disable Reseller Limits

To start using a new feature Reseller limits you would have to enable reseller limits for a particular reseller first.

To enable Reseller access, please do the following:

- 1.Log in with a Hoster access.
- 2.You can create a new account or give privileges to an existing account.
- 3.For new account tick a checkbox Make this account a reseller in the Reseller Settings box.



Note. If checkbox Make the account own itself (i.e., the user can modify the account) is not selected when creating Reseller in cPanel WHM, then user account Reseller will belong to root, not to reseller Reseller. In such case, the user Reseller will be managed by the root. So, LVE limits specified by the root will be applied to the user Reseller. User Reseller will not be limited by Reseller limits.

When the checkbox is selected, user Reseller will be limited by Reseller limits (in addition to personal user limits set by Reseller).

4. Give privileges to the proper Reseller account to make all features work.

5. Go to the Users tab and choose a particular reseller you want to enable Reseller limits for and click on the pencil icon.

6. In the pop-up window move the slider Manage Limits. Click AGREE for the question Are you sure you want to enable limits, set limits for that reseller if you want them to be different from the default limits, otherwise default server limits will be applied. Then click the Save button.

Manage Limits (1)

Please note, that resellers' end users can use as much resources in total as it is provided for that particular reseller by a hoster. The summary usage of all end users that belong to that particular reseller will not exceed the amount of resources provided to reseller by a hoster. If no Reseller Limits are set, reseller's end user will be limited by default limits set by a hoster.

How to Disable Reseller Limits

=

To disable Reseller limits, please do the following:

1. Go to the Users tab, choose a particular reseller and click on the pencil icon.
2. In the pop-up window move the slider Manage Limits. Click AGREE for the question Are you sure you want to disable limits. Then click the Save button.

Please note, that if you disable Reseller limits everything will work the same as before. All the end user limits set by the reseller will be saved. But all custom default reseller limits will be disabled.

Hoster Interface

Hoster interface allows to monitor and manage limits for hosters' end users, resellers and resellers' end users, and also manage packages and monitor statistics.

Hoster credentials allow to control limits for hosters' end users and resellers. To control reseller end user limits Hoster has to log in as Reseller.

Log in as Hoster to get access to the following functionality.

- ‘<#current_usage_tab.html>’ __*Current Usage* tab allows to monitor users and resellers resource usage at the moment.
- ‘<#users_tab.html>’ __*Users* tab with the list of all users and resellers allows viewing and managing all the users and resellers limits.
- ‘<#statistics_tab.html>’ __*Statistics* tab displays the statistics of resource usage for particular timeframe or particular user.
- ‘<#options_tab.html>’ __*Options* tab allows to set LVE faults email notifications for hoster, users, and resellers.
- ‘<#packages_tab.html>’ __*Packages* tab allows to manage resellers packages limits;
- ‘<#selector_tab.html>’ __*Selector* tab allows to control PHP Selector settings.

Current Usage Tab

Choose Current Usage tab to monitor users, resellers and resellers' end users resource usage at the moment displayed in the table.

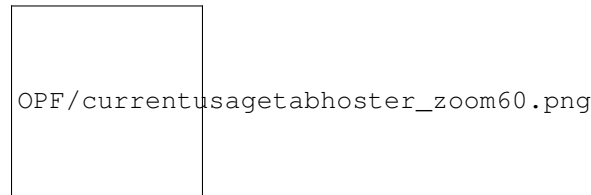
Current Usage table provides information on usage of SPEED (All and MySQL), memory (MEM), data throughput (IO) (All and MySQL), read/write operations per second (IOPS), number of processes (PNO), and entry processes (EP).

Resource usage values are being refreshed every 10 seconds by default which is set in Auto-refresh field. You can set Auto-refresh time by choosing a value from the drop-down. You can refresh the table manually by clicking Refresh now or you can freeze the values by clicking pause button. Usage values will not change until the next manual refresh. To unfreeze click on unpause button. The countdown will continue.

Tick Hide MySQL usage checkbox to hide the information on MySQL usage.

The list of users can be filtered by Username and Domain. Hoster can view all types of users: End users, Resellers, Reseller's end users, Reseller's end users (no Reseller limit). But hoster can only manage End users, Resellers, and Reseller's end users (no Reseller limit). To manage Reseller's end users hoster should login as a reseller.

In the drop-down Show top you can choose the number of user to be displayed on the page.



Users Tab

Choose Users tab to view the list of all users and manage their limits.

To filter the list by user type click Manage and in the drop-down choose:

- End users - to manage hosts end users only.
- Resellers - to manage resellers only.
- Reseller's end users - to manage resellers' end users only.
- Reseller's end users (no Reseller limits) - to manage resellers' end users that do not have limits specified by reseller (these limits are specified by the hoster).

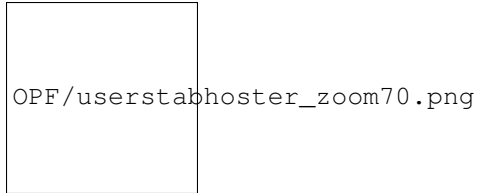
To filter the list by Username, Domain, LveID click Filter by and choose the value in the drop-down.

Note that a hoster can view the list of resellers' end users and their limits, but can not manage resellers' end users limits (if those are set by reseller).

A hoster can view the limits of all types of users and manage the limits for hosters' end users and resellers' end users (only those with Reseller Limits disabled).

Tick Show users with CageFS enabled to show users with CageFS file system enabled.

Tick Show only ignored users to show users with ignored MySQL Governor.

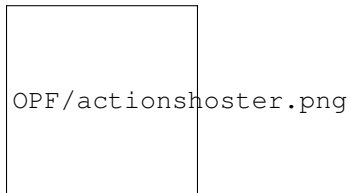


Actions column

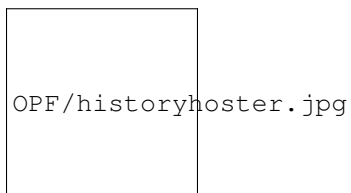
Click on a pencil icon in Actions column to edit limits for a particular user. The following actions are available:

- Enable/disable CageFS;
 - Reset - to reset limits to default values;
- Apply Do not limit to set the limits to unlimited;
- Setting the limits values:
 - o SPEED
 - o SPEED MYSQL
 - o VMEM
 - o PMEM
 - o IO
 - o MySQL IO
 - o IOPS
 - o EP
 - o NPROC
 - o INODES (hard and soft) (for end users and resellers' end users (with no Reseller Limits), if a hoster has enabled Initial quotas in cPanel settings).

Click Save to save changes or Cancel to close pop-up window.



Click on History icon to view the history of a particular user resource usage. Choose time frame to view the history for a particular time period.



Statistics Tab

Choose Statistics tab to view end users, resellers and resellers' end users limits usage statistics.

The following parameters can be displayed in the statistics table:

- SPEED usage per user;
- IO usage per user;
- EP usage per user;
- VMEM usage per user;
- PMEM usage per user;
- NPROC usage per user;
- IOPS usage per user;
- MySQL usage per user.

Click Show button and select columns from the drop-down to set which parameters should be displayed in the table.

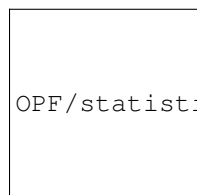
Statistics table can be filtered by:

- Timeframe - to view the statistics for a particular period;
- Limit - to view a particular limit type usage only;
- Top LVEs - to view top used limits only;
- LVE approaching limit - to view the limits that are approaching

maximum provided value;

- Fault LVE - the limits that have reached the maximum value.

Click Manage to choose type of users to be displayed - End users, Resellers, Resellers' end users or Resellers' end users (no Reseller limit) by ticking checkbox in the drop-down.



Click on a chart icon in View column to view the detailed resource usage history for a particular account. Use timeframe drop-down to view the history for a particular period of time.

history\charts

Options Tab

A hoster can set email notifications for panel administrator, reseller customer, and resellers' customers in cases of limits faults. Choose Options tab to manage LVE Faults email notifications.

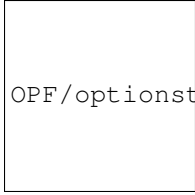
In LVE Faults Email Notifications section tick the required checkboxes to set a type of notification.

Notify Panel Administrator - notify hoster when his end users have exceeded minimum number of faults set for particular limits.

Notify Reseller - notify reseller when his end users have exceeded minimum number of faults set for particular limits.

Notify Customers - notify hosters' end users when they have exceeded limits.

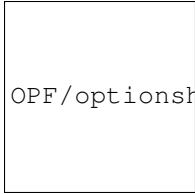
Notify Reseller's customers - notify resellers' end users when they have exceeded limits.



OPF/optionstabemailnotifhoster.png

In Faults to include section tick the checkboxes to include required limits to the notifications.

Set the frequency of email notifications sending in Notify . . . every.. days/hours/minutes/seconds section.



OPF/optionshostefaultstoinclude.png

In Minimum number of Faults to notify section enter the number of faults required for the notification to be sent for Panel Admin & Reseller and User.

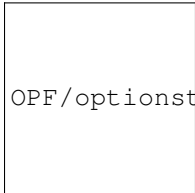


OPF/optionstabhosterminimumftn.png

In Inodes limits section you can reset inode limits to default values and tick Show end-user inode usage.

In User interface settings section tick the required checkboxes to apply user interface settings.

In MySQL Governor settings section you can customize MySQL Governor.



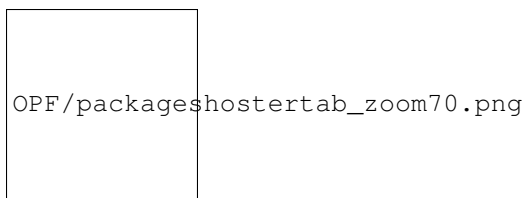
OPF/optionstabhosterinodes.png

Packages Tab

Packages tab allows to set the limits for as many users as you need by editing packages of the limits. Each account belonging to a particular package adheres to those limits.

Choose Packages tab to view and modify:

- limits for user packages (created by hoster);
- limits for reseller packages (created by hoster);
- limits for resellers' end users packages if reseller limits are not set for that reseller (hoster access allows identifying a particular reseller's end user belonging to a particular reseller (created by reseller)).



To modify package limits click on a pencil icon in Actions column in a particular package row. The following limits for this package are available for setting:

- SPEED in percent (%);
- Virtual memory (VMEM) (can be set as unlimited by setting 0);
- Physical memory (PMEM) (can be set as unlimited by setting 0);
- I/O limits (IO) (can be set as unlimited by setting 0);
- IOPS limits;
- Concurrent connections (EP);
- Number of processes (NPROC) (can be set as unlimited by setting 0);
- INODES (hard and soft) (for end users and resellers' end users (with

no Reseller Limits), if a hoster has enabled Initial quotas in cPanel settings.)

When limits are set click Save to apply changes or Cancel to close the window.

Selector Tab

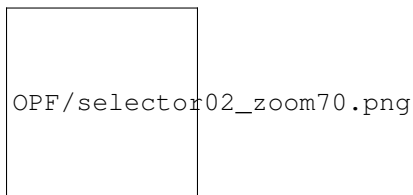
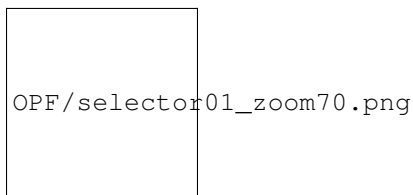
Selector tab allows to control PHP Selector settings.

In Selector is choose Enabled or Disabled from the drop-down to enable or disable PHP Selector.

In Default PHP version choose PHP version or Native from the drop-down to apply.

In Supported versions choose required PHP versions to support.

Choose default modules from the list for a particular version of PHP or for native.



Reseller Interface

Reseller interface is designed to manage limits for resellers' end users, to monitor statistics and the history of resource usage and to modify reseller's end user packages limits.

Log in under a particular reseller credentials to have access to the following functionality:

- ‘<#current_usage_tab2.html>’ __*Current Usage* tab - allows to monitor resellers’ end users resource usage at the moment;
- ‘<#historical_usage_tab.html>’ __*Historical Usage* tab - allows to control resellers’ end users resource usage history;
- ‘<#users_tab2.html>’ __*Users* tab with the list of all resellers’ end users allows to view and manage all the reseller’s end user limits;
- ‘<#statistics_tab2.html>’ __*Statistics* tab displays the statistics of resource usage for particular timeframe or particular reseller’s end user;
- ‘<#options_tab2.html>’ __*Options* tab allows to set LVE Faults email notifications.
- ‘<#packages_tab2.html>’ __*Packages* tab allows to manage reseller’s end user packages limits.

Please note that reseller can manage all his end users via Reseller Interface. Reseller cannot manage INODE or MYSQL limits, neither his own nor for his users.

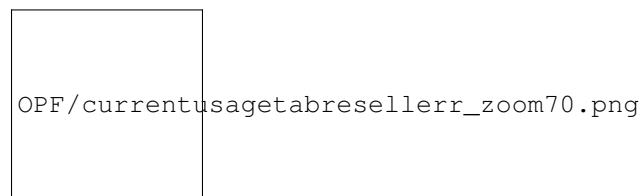
Current Usage Tab

Current usage table provides the information on the usage of SPEED (All), memory (MEM), data throughput (IO) (All), read/write operations per second (IOPS), number of processes (PNO), and entry processes (EP).

Resource usage data is being refreshed every 10 seconds which is set by default in Auto-refresh field. You can set Auto-refresh time by choosing the value from the drop-down. You can refresh the table manually by clicking Refresh now or you can freeze the values by clicking pause button. Usage values will not change until the next manual refresh. To unfreeze click on unpause button. The countdown will continue.

Reseller cannot manage INODE or MYSQL limits. Neither his own, nor for his users.

The bottom line star in the table displays the total reseller resource usage. It means, that all the usage of resellers’ end users and of his own is displayed as a summary for each parameter.



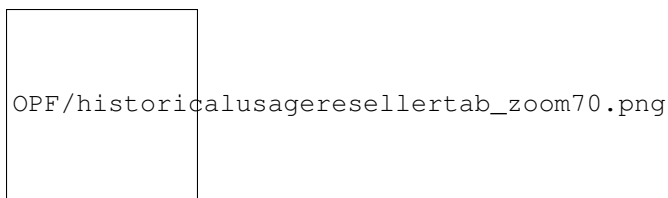
Historical Usage Tab

Choose Historical Usage tab to view reseller and resellers’ end users resource usage history and faults. The list of users can be filtered by Timeframe.

When reseller’s end user reaches the limits set by hoster for the reseller, this will be displayed on the chart. Please note, that in this case reseller’s end user would not necessarily reaches his limits set by the reseller. These faults are not displayed on the chart.

On the Historical Usage page the reseller is also able to see the list of Top 5 Reseller’s end users (based on resource usage, for the same period as charts/overall usage). Click on a History icon in the Actions column to view resource usage statistics for particular user.

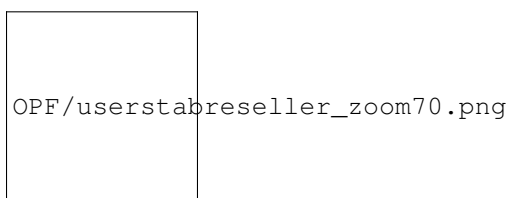
Click on LVE Statistics link in the top of the Top 5 list to go to the Statistics page to view or manage the rest of users.



Users Tab

Choose Users tab to view and manage the list of all resellers' end users and resource usage limits provided for them. The following limits are available for the resellers' end users: SPEED, PMEM, IO, IOPS, EP, NPROC.

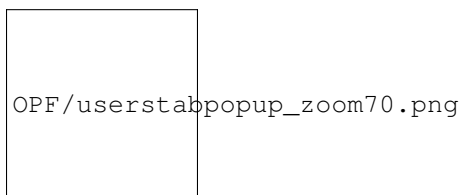
You can filter the list by Username, Domain, LVE ID. Tick Show only ignored users checkbox to display only users with MySQL Governor disabled.



Actions column

Click on a pencil icon in Actions column to edit limits for a particular user. The following actions are available:

- Click Reset to reset limits to default values.
- Click Apply for Do not limit to set unlimited resources to a user.
- Set values for SPEED, PMEM, IO, IOPS, EP, and NPROC and click Save to save changes or Cancel to close the window.



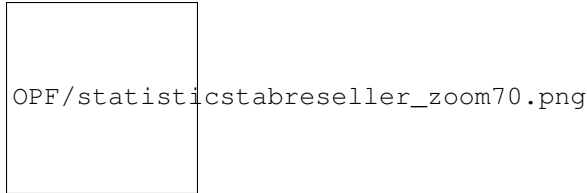
Statistics Tab

Choose Statistics tab to view resource usage limits statistics.

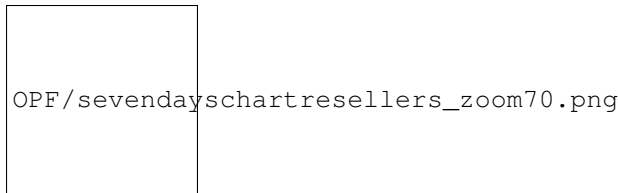
Statistics table can be filtered by Timeframe, Limit, Top LVEs, LVE approaching limit, Fault LVE.

The following parameters are displayed:

- SPEED per user;
- PMEM usage per user;
- IO usage per user;
- EP usage per user;
- NPROC usage per user;
- IOPS usage per user.

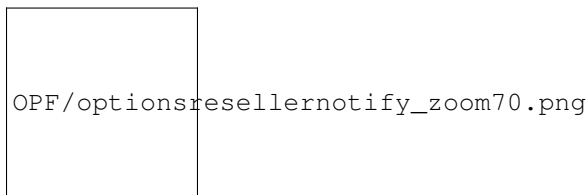


Use Charts icon in the View column to view detailed resource usage charts for a particular period of time.
For example, 7 days period chart.

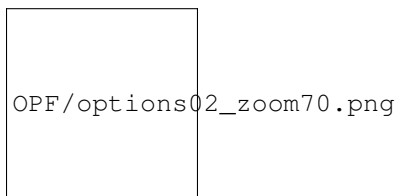


Options Tab

Choose Options tab to set user email notifications for resellers' end users.
In LVE Faults email notifications section tick appropriate checkboxes to set the required type of notification.

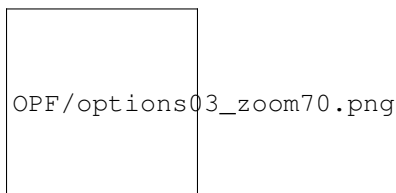


Notify me on users faults - notify reseller when his users have exceeded limits.
Notify Customers - notify resellers' end users when they have exceeded limits.
Notify me when I hit my limits - notify reseller when overall resource usage limits are reached.
In Faults to include section tick checkboxes to include particular limits to email notifications.



In Minimum number of Faults to notify section enter the number of faults required for the notification to be sent for reseller and customer. You can also set the reseller notification frequency.

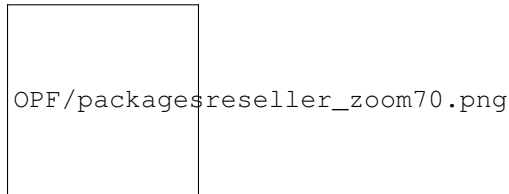
Set the frequency of sending the reseller email notifications in Notify Reseller Every ... days/hours/minutes/seconds section.



Click Save Changes to apply changes.

Packages Tab

Choose Packages tab to view and modify limits for reseller's packages.



Click on a pencil icon in a package row to set the following limits for a package:

- SPEED limit;
- Physical memory (PMEM) (can be set as unlimited by setting 0);
- I/O limits;
- IOPS limits;
- Concurrent connections (EP) limits.

When limits are set click Save to apply changes.

LVE-Stats 2

Why is it needed?

- Old LVE-statistics store averages as integer numbers, as % of CPU usage. If user used 100% of CPU for 1 second within an hour, it is only 1-2% for a minute, and 0 for 5 minutes. Data in old LVE-statistics is aggregated to 1-hour intervals. So, such peak load will not be recorded and we need to store data with much higher precision.
- 100% CPU usage in old lve statistics means “all cores”. On 32 core servers usage is not visible for most users (as they are limited to 1 core).
- Old LVE-statistics does not provide a way to determine a cause of LVE faults, i.e. what processes are running when user hits LVE limits.
- Notifications in old LVE-statistics are not accurate because they are based on average values for CPU, IO, IOPS.
- Old LVE-statistics functionality is hard to extend.

Major improvements and features

- increased precision of statistics;
 - CPU usage is calculated in terms of % of a single core (100% usage means one core);
- lvestats-server emulates and tracks faults for CPU, IO, IOPS;
- lvestats-server saves “snapshots” of user’s processes and queries for each “incident” - added new lve-read-snapshot utility;
- improved notifications about hitting LVE limits (more informative and without false positives);
- implemented ability to add custom plugins;
- MySQL and PostGreSQL support;

- more pretty, scalable, interactive charts;
- snapshots include HTTP-requests.

What features will be implemented in the future?

- Notifications for control panels other than CPanel.
- Burstable Limits/server health: We are monitoring server health (LA, memory, idle CPU) and automatically decreasing/increasing limits based on server health.
 - Reseller Limits: plugin would analyze usage per group of users (reseller's usage), and do actions.
- Suspend/notify plugin: would detect that user is being throttled for 10 minutes, and suspend him (just because), or notify, or increase limits.

Installation

Installation

To install please execute:

```
yum install lve-stats
```

To update:

```
yum update lve-stats
```

Settings of old lve-stats (ver. 0.x) are imported automatically on the first install/update of new lve-stats package.

SQLite database file is located in /var/lve/lvestats2.db, data from old lve-stats (ver. 0.x) are being migrated automatically in the background. Migrating process can last 2-8 hours (during this time lags are possible when admin is trying to check statistics, at the same time users will not be affected). Migrating the latest 30 days, SQLite DB stable migration is provided.

Currently new lve-stats supports all databases available in CloudLinux (except PostgreSQL for CL5).

Downgrade

If you have any problems after update, downgrade lve-stats2 to the previous stable version by running:

```
yum downgrade lve-stats
```

and contact CloudLinux support at <https://helpdesk.cloudlinux.com>

Note. You may need to rename *.rpmsave files to original ones in order to restore settings for old lve-stats (/etc/sysconfig/lvestats, /etc/sysconfig/cloudlinux-notify).

Configuration

Configuration

Main configuration file /etc/sysconfig/lvestats2 contains the following options:

db_type - selects appropriate database type to use;

connect-string - connection string for PostgreSQL and MySQL database, has the following form:

```
connect_string = USER:PASSWORD@HOST[:PORT]/DATABASE
```

Default port is used for specific database, if port is not specified (typical port is 3306 for MySQL and 5432 for PostgreSQL). Connection string is not used for sqlite database.

`server_id` - sets the name of the server (at most 10 characters). This option is to use with centralized database (Post-GreSQL or MySQL). For use with sqlite database, value of this option should be "localhost" (without quotes).

`plugins` – path to directory containing custom plugins for lve-stats (default path /usr/share/lve-stats/plugins).

`db_timeout` - period of time to write data to database (in seconds); default value is 60 seconds.

`timeout` - timeout for custom plugins (seconds). If plugin execution does not finish within this period, plugin is terminated. Default value is 5 seconds.

`interval` - duration of one cycle of lvestats-server (seconds). This should be less than total duration of execution of all plugins. Default value is 5 seconds. Increasing this parameter makes precision of statistics worse.

`keep_history_days` - period of time (in days) to keep history in database. Old data is removed from database automatically. Default value is 60 days.

`mode` – sets compatibility output mode (compatibility with older lveinfo version). Value "v1" (without quotes) enables compatibility with old version of lveinfo. "v2" value enables "extended" output mode, but can break LVE plugins for control panels (statistics in LVE Manager, Resource Usage, etc). Support of v2 mode will be added to LVE plugins in the recent future. When mode parameter is absent, later version of lveinfo is implied.

`disable_snapshots` - disable snapshots and incidents. Possible values: true, false.

Configuration files for plugins are located in /etc/sysconfig/lvestats.config directory.

/etc/sysconfig/lvestats.config/SnapshotSaver.cfg contains the following options:

`period_between_incidents` - Minimal interval of time between incidents (in seconds). If minimal interval of time between LVE faults is greater than value specified, than new "incident" will begin and new snapshots will be saved. Default value is 300 seconds.

`snapshots_per_minute` - Maximum number of snapshots saved per minute for specific LVE id (default is 2).

`max_snapshots_per_incident` - Maximum number of snapshots saved for one "incident". Default is 10.

/etc/sysconfig/lvestats.config/StatsNotifier.cfg contains the following options:

`NOTIFY_ADMIN` – enables notification for admin (Y/N, default N);

`NOTIFY_RESELLER` – enables notification for reseller (Y/N, default N);

`NOTIFY_CUSTOMER` - enables notification for customers (Y/N, default N);

`NOTIFY_INCLUDE_RESELLER_CUSTOMER` – Y=notify all users, N=notify only hoster's users (whos reseller is root), default = N;

`NOTIFY_CPU` – notify about CPU faults when customer hits 100% of his CPU limit (Y/N, default N);

`NOTIFY_IO` - notify about IO faults when customer hits 100% of his IO limit (Y/N, default N);

`NOTIFY_IOPS` - notify about IOPS faults when customer hits 100% of his IOPS limit (Y/N, default N);

`NOTIFY_MEMORY` - notify about memory faults (Y/N, default N);

`NOTIFY_EP` – notify about entry processes faults (Y/N, default N);

`NOTIFY_NPROC` – notify about number of processes faults (Y/N, default N);

`NOTIFY_MIN_FAULTS_ADMIN` – minimum number of faults to notify admin (default 1);

`NOTIFY_MIN_FAULTS_USER` – minimum number of faults to notify customer (default 1);

`NOTIFY_INTERVAL_ADMIN` – period of time to notify admin (default 12h);

`NOTIFY_INTERVAL_USER` – period of time to notify customer (default 12h);

`NOTIFY_FROM_EMAIL` - sender email address. For example: `NOTIFY_FROM_EMAIL=main_admin@host.com`

NOTIFY_FROM_SUBJECT - email message subject. For example: NOTIFY_FROM_SUBJECT=Message from notifier

Templates of notifications are located here:

/usr/share/lve/emails/en_US/admin_notify.txt

/usr/share/lve/emails/en_US/reseller_notify.txt

/usr/share/lve/emails/en_US/user_notify.txt

/usr/share/lve/emails/en_US/admin_notify.html

/usr/share/lve/emails/en_US/reseller_notify.html

Note: Notifications about LVE faults are implemented for CPanel only.

Note: After changing any options above please restart lvestats service:

service lvestats restart

/etc/logrotate.d/lvestats - configuration file for /var/log/lve-stats.log rotation

LVE Stats2 and MySQL DB Server Compatible Work Setup

Note. Run all the commands below under root.

1. MySQL Server Setup

If MySQL Server is not installed, then install it according to control panel documentation.

For non-panel system:

(CloudLinux 6)

yum install mysql mysql-server

service mysqld start

chkconfig mysqld on

(CloudLinux 7)

yum install mariadb mariadb-server

systemctl start mariadb.service

systemctl enable mariadb.service

2. Database Setup

1. Run MySQL administrative utility: mysql.

2. In utility run the commands:

1.

CREATE DATABASE db_lvestats2;

creating server DB. Also, check Note below.

2.

CREATE USER 'lvestats2'@'localhost' IDENTIFIED BY 'lvestats2_passwd';

creating a user for LVE Stats 2 server to work under. Also, check Note below.

3.

```
GRANT ALL PRIVILEGES ON db_lvestats2.* TO 'lvestats2'@'localhost';
```

granting all the privileges for all DB tables to the user. Use the username and DB name from points a. and b. above.

4.

```
FLUSH PRIVILEGES;
```

refreshing privileges information.

5. Exit administrative utility (Ctrl+d).

Note. DB name, username and their passwords above are given for an example - you can use any of your choices. Using old DB from LVE Stats version 1 is also acceptable as LVE Stats2 uses different tables and the old information will not be corrupted.

3. LVE Stats 2 Setup

Stop LVE Stats 2 server running the command:

```
service lvestats stop
```

In server configuration file `/etc/sysconfig/lvestats2` edit the following options:

```
db_type = mysql
```

```
connect_string = lvestats2:lvestats2_passwd@localhost/db_lvestats2
```

Note that `connect_string` option value is used in format: `user:pass@host/database`. Username, password and DB name must be the same as in point 2.b. of Database Setup above.

After making changes in configuration files run

```
/usr/sbin/lve-create-db
```

for DB primary initialization (creating tables, indexes, etc). There is no need to create anything in the DB manually.

When done, restart server running:

```
service lvestats restart
```

4. Additional Security Settings

If you need to provide access to LVE Stats information utilities (`lveinfo`, `lvechart`, `lve-read-snapshot`) for different users, then we recommend creating one more DB user with read-only privilege to guarantee information security. It can be done by running the following commands in administrative utility:

1.

```
CREATE USER 'lvestats2_read'@'localhost' IDENTIFIED BY 'lvestats2_read_passwd';
```

creating a user (check Note above).

2.

```
GRANT SELECT ON db_lvestats2.* TO 'lvestats2_read'@'localhost';
```

granting read-only privilege to the user.

3.

```
FLUSH PRIVILEGES;
```

refreshing privileges information.

If LVE Stats2 server is set correctly (see information below), the information utilities will work under this user.

If you need to provide access to information utilities to other users, then in order to guarantee information security you should do the following:

) Assign permission 600 to the main configuration file (/etc/sysconfig/lvestats2), so that it could be read only by LVE Stats 2 server and by utilities that run under root.

b) Copy /etc/sysconfig/lvestats2 to /etc/sysconfig/lvestats2.readonly, assign permission 644 to the new file, so that it could be read by any user but could only be changed by root.

) In /etc/sysconfig/lvestats2.readonly file, in the line connect_string, specify DB user with read-only permission, created above.

These steps allow hiding main DB user username/password from other system users.

If there is no need in such access differentiation, then /etc/sysconfig/lvestats2 file access permission should be 644, so that it could be read by all users and could be changed only by root.

5. Using Special Characters in Database Password

Since scheme://user:password@host[:port]/database_name URI is used in connect_string config option, then usage of special characters in user DB password is not allowed. To use special symbols in the password, it must be converted to [escape-sequence](#). You can convert a password to escape-sequence in a console as follows:

```
echo -n '[You_P@$:]' | perl -MURI::Escape -ne 'print uri_escape($_)."\n"'
%5BYou_P%40%24%24%5D%3A
```

Or replace the symbols manually:

```
! # $ & ' ( ) * + , / : ; =
? @ [ ]
%21 %23 %24 %26 %27 %28 %29 %2A %2B %2C %2F %3A %3B %3D %3F %40 %5B %5D
```

After that onnect_string will look as follows:

```
onnect_string=lvestats2:%5BYou_P%40%24%24%5D%3A@localhost/db_lvestats2
```

LVE Stats 2 and PostgreSQL DB Server Compatible Work Setup

Note. Run all the commands below under root.

1. PostgreSQL Server Installation and Setup

1.1 PostgreSQL installation and initialization.

For control panels use proper documentation for installation on the links: [Panel](#), [Plesk](#).

For non-panel CloudLinux run the following commands:

(CloudLinux 6)

```
yum install postgresql-server postgresql
service postgresql initdb
service postgresql start
chkconfig postgresql on
```

(CloudLinux 7)

```
yum install postgresql-server postgresql
postgresql-setup initdb
systemctl start postgresql
systemctl enable postgresql
```

1.2. Setup.

1. In `/var/lib/pgsql/data/pg_hba.conf` config file change user authentication mode. Add the following lines (place before all other authentication parameters):

```
# IPv4 local connections for lve-stats-2.x
```

```
host dbvestat all 127.0.0.1/32 password
```

```
# IPv6 local connections for lve-stats-2.x
```

```
host dbvestat all ::1/128 password
```

These lines enable user authentication by the password for IP4/IP6 connections. You can set other modes if needed.

3. Apply config changes by running:

```
service postgresql restart
```

2. DB for lve-stats-2.x - Creating and Setup

1. Run standard PostgreSQL `psql` administrative utility:

```
sudo -u postgres psql postgres
```

(`psql -w -U postgres` for Panel).

2. In utility run:

1.

```
CREATE DATABASE dbvestat;
```

creating server DB. Also, check Note below.

2.

```
CREATE USER lvestat WITH password 'passw';
```

creating a user for LVE Stats 2 server to work under. Also, check Note below.

3.

```
GRANT ALL privileges ON DATABASE dbvestat TO lvestat;
```

granting lvestat user all privileges for work with dbvestat DB.

4. `\q` - exit `psql` utility. (Alternatively `Ctrl+d`).

Note. DB name, username and their passwords above are given for an example - you can use any of your choices. Using old DB from LVE Stats version 1 is also acceptable as LVE Stats 2 uses different tables and the old information will not be corrupted.

3. Lve-stats-2.x Setup

Stop lve-stats2 server by running:

```
service lvestats stop
```

In server config file `/etc/sysconfig/lvestats2` edit options for connecting to DB:

```
db_type = postgresql
```

```
connect_string=lvestat:passw@localhost/dbvestat
```

If DB is going to be used as centralized for multiple hosts then `collect_usernames` parameter must be changed:

```
collect_usernames=true
```

Note that `connect_string` option value is of the format: `user:pass@host/database`. Username, password and DB name must be the same as in Database Setup section above.

After making changes in configuration files, for DB primary initialization (creating tables, indexes, etc) run:

```
/usr/sbin/lve-create-db
```

There is no need to create anything in the DB manually. When done, restart server by running:

```
service lvestats restart
```

4. Additional Security Settings

If you need to provide access to LVE Stats information utilities (`lveinfo`, `lve-read-snapshot`) for other users (or if CageFS is disabled), then in order to guarantee DB security the following steps are required:

1. Create a DB user with read-only permission:

```
CREATE USER lvestat_read WITH password 'passw';
```

```
GRANT CONNECT ON DATABASE dblvestat to lvestat_read;
```

```
\connect dblvestat;
```

```
GRANT SELECT ON lve_stats2_history, lve_stats2_history_gov, lve_stats2_history_x60, lve_stats2_incident,
lve_stats2_servers, lve_stats2_snapshot, lve_stats2_user TO lvestat_read;
```

- b. Assign root ownership and permission 600 to the main configuration file (`/etc/sysconfig/lvestats2`), so that it could be read only by LVE Stats 2 server and by utilities that run under root.

- c. Copy `/etc/sysconfig/lvestats2` to `/etc/sysconfig/lvestats2.readonly`, assign permission 644 to the new file, so that it could be read by any user but could be changed only by root.

- d. In `/etc/sysconfig/lvestats2.readonly` file, in the line `connect_string`, specify DB user with read-only permission, created above.

These steps allow hiding main DB user username/password from other system users.

If there is no need in such access differentiation, then `/etc/sysconfig/lvestats2` file access permission should be 644, so that it could be read by all users and could be changed only by root.

When done restart server by running:

```
service lvestats restart
```

5. Using Special Characters in Database Password

Since `scheme://user:password@host[:port]/database_name` URI is used in `connect_string` config option, then usage of special characters in user DB password is not allowed. To use special symbols in the password, it must be converted to [escape-sequence](#). You can convert a password to escape-sequence in a console as follows:

```
echo -n '[You_P@$:]' | perl -MURI::Escape -ne 'print uri_escape($_)."\n"'
```

```
%5BYou_P%40%24%24%5D%3A
```

Or replace the symbols manually:

```
! # $ % & ' ( ) * + , / : ; =
```

```
? @ [ ]
```

```
%21 %23 %24 %26 %27 %28 %29 %2A %2B %2C %2F %3A %3B %3D %3F %40 %5B %5D
```

After that `onnect_string` will look as follows:

```
onnect_string=lvestats2:%5BYou_P%40%24%24%5D%3A@localhost/db_lvestats2
```

Customize lve-stats-2 notifications

Jinja2 is used as a template engine for the notifications.

The templates for notifications are located in `/usr/share/lve/emails/LOCALE`, where LOCALE - is the directory with localization name (language codes are formed according to ISO 639-1 and ISO 639-2). By default the templates for English are set: `/usr/share/lve/emails/en_US`.

`/usr/share/lve/emails/en_US` contains the following templates:

- `admin_notify.html admin_notify.txt` for administrator;
- `reseller_notify.html reseller_notify.txt` for reseller;
- `user_notify.txt` for user.

The notification is formed as Multipart content type [RFC1341(MIME)]. The plain text is taken from the `.txt` files, html version - from the `.html` template. In case when only one template is present (`.txt` or `.html`) the notification is sent as a Non-multipart content type notification. It is better to use Multipart content type notifications because when a mail client can not display an html-format message, then it will be displayed as plain text version.

To localize notifications copy standard templates into directory with the proper locale name and translate the template. Also you can customize the main template making proper changes into it.

The list of variables that can be used in the template:

Variable	Example	Description
TONAME	“Customer”	Notification receiver user name. Taken from profile in the control panel, by default - “Customer” for user, “Administrator” for administrator, “Reseller” for reseller.
TOMAIL	“support@cloudlinux.com”	Notification receiver email address.
DOMAIN	“word-press.test247.cloudlinux.com”	Main domain. Available only for user.
LOCALE	“en_US”	Locale in which the notification is sent. Available only for user.
RESELLER	“root”	User reseller. Available only for user.
PERIOD	“12 hours”	Verification and notification sending period.
LOGIN	“word-press”	User login in the system.
ID	500	User ID in the system.
IPMem IEP PMemF IVMem anyF IOF VMemF ICPU aIOPS aEP aPMem IOPSf IO IOPS aIO EPf aCPU aVMem NprocF aNproc INproc CPUf		See description in <code>lveinfo -help</code> output. Available only for users
STATS_HTML		html table with the list of users that exceeded limits. Available for administrator and reseller.
STATS		ascii - table with the list of users that exceeded limits. Available only for admins and resellers.

Sender’s email address by default is administrator email address from control panel settings (`root@{hostn_name}`) if there is no email in the control panel).

It can be changed with `NOTIFY_FROM_EMAIL` option in the config `/etc/sysconfig/lvestats.config/StatsNotifier.cfg`

For example:

`NOTIFY_FROM_EMAIL=support@hostname.com`

To apply changes restart lve-stats service:

```
service lvestats restart
```

for CloudLinux 7

```
systemctl restart lvestats.service
```

Default subject is “Hosting account resources exceeded”. It can be changed for each template (and for localized templates as well). To change subject, in the very beginning of the file (no blank lines allowed in the beginning of the template) add the field Subject:, leave two blank lines after it and add template body.

Customized subjects can be taken only from the templates with the resolution *.txt (admin_notify.txt, reseller_notify.txt, user_notify.txt). Changes apply without lvestats restart.

For backward compatibility the subject can be also changed with the key NOTIFY_FROM_SUBJECT in the config /etc/sysconfig/lvestats.config/StatsNotifier.cfg

Customized subjects have higher priority than the key NOTIFY_FROM_SUBJECT.

Example for the file user_notify.txt

Subject: Customized subject example

Dear {{TONAME}},

Your {{DOMAIN}} web hosting account exceeded one or more of its resources within the last {{PERIOD}}.

{% if epf %}Exceeded the maximum of {{lep}} concurrent website connections. Your website was not available {{epf}} times because of this problem.

{% endif %}{% if pmemf %}Exceeded the physical memory limit of {{lpmem}}KB. Your website was not available {{pmemf}} times because of this problem.

{% endif %}{% if vmemf %}Exceeded the virtual memory limit of {{lvmem}}KB. Your website was not available {{vmemf}} times because of this problem.

{% endif %}{% if nprocf %}Exceeded the number of processes limit of {{lnproc}}. Your website was not available {{nprocf}} times because of this problem.

{% endif %}{% if cpuf %}You reached limit of {{lcpu}} of total server CPU usage {{cpuf}} times. Your website was forced to load slower to reduce its CPU usage.

{% endif %}{% if iof %}You reached limit of {{lio}}KB/s disk io rate {{iof}} times. The disk io speed for your account was slowed as a result of this problem.

{% endif %}{% if iopsf %}You reached limit of {{liops}} I/O operations {{iopsf}} times. The disk io speed for your account was slowed as a result of this problem.

{% endif %}

To view full details about your web hosting account’s resource usage, including the time of each incident listed above, please click the link below and log into your cpanel hosting control panel, then click the “Resource Usage” link under the “Logs and Statistics” section.

<http://{{DOMAIN}}:2083>

If your account is regularly exceeding it’s available resources, please consider upgrading to a higher level hosting plan that includes more resources. If you have any questions or need help with anything, just reply to this email and let us know.

Sincerely,

Your Friendly Web Hosting Support Team

Command-line Tools

Command line tools

/usr/sbin/lveinfo	utility to display historical information about LVE usage;
/usr/sbin/lvechart	creates a chart representing LVE usage for user;
/usr/sbin/dbgovchart	creates a chart representing MySQL usage for user;
/usr/sbin/lve-read-snapshot	displays information from system state (snapshots) for user;
/usr/sbin/lve-create-db	creates/recreates database for lve-stats;
/usr/sbin/cloudlinux-top	utility provides information about current MySQL and LVE usage of a running system in JSON format.
/usr/sbin/cloudlinux-statistics	utility provides historical information about resource usage.

lveinfo

[lve-stats-2.2-2]

usage: lveinfo [-h] [-v] [-dbgov DBGOV] [-f YYYY-MM-DD[HH:MM]]

[-t YYYY-MM-DD[HH:MM]] [-period PERIOD] [-u USER | -id

ID]

[-d] [-o ALIAS] [-b ALIAS [ALIAS ...]] [-p 0..100]

[-by-fault ALIAS [ALIAS ...]] [-r FAULTS]

[-style {user,admin}] [-l LIMIT] [-c [PATH] | -j]

[-server_id SERVER_ID] [--servers-info]

[-show-all | --show-columns COLUMN_NAME [COLUMN_NAME

...]]

[-time-unit TIME_UNIT] [-m {v1,v2}]

[-blank-value [BLANK_VALUE]]

lveinfo - Utility to display historical information about LVE usage

Optional arguments:

-h, --help	show this help message and exit
-v, --version	show program's version number and exit
--dbgov DBGOV	show MySql Governor statistic
-u USER, --user USER	Use username instead of LVE id, and show only record for that user
-id ID	will display record only for that LVE id
-d, --display-username	try to convert LVE id into username when possible
-o ALIAS, --order-by ALIAS	orders results by one of the following:

ALIAS	ALIAS	DESCRIPTION
cpu_avg	aCPU	average CPU usage
cpu_max	mCPU	max CPU usage
total_cpu_faults	CPUf	total number of max CPU usage faults
vmem_avg	aVMem	average virtual memory usage
vmem_max	mVMem	average virtual memory usage
total_vmem_faults	VMemF	total number of out of virtual memory faults
mep_avg	aEP	average number of entry processes (concurrent connections)
mep_max	mEP	max number of entry processes (concurrent connections)
total_ep_faults	EPf	total number of max entry processes faults

pmem_avg	aPMem	average physical memory usage (LVE version >= 6)
pmem_max	mPMem	max physical memory usage (LVE version >= 6)
nproc_avg	aNproc	average number of processes (LVE version >= 6)
nproc_max	mNproc	max number of processes (LVE version >= 6)
io_avg	aIO	average io usage (LVE version >= 6)
io_max	mIO	max io usage (LVE version >= 6)
total_pmem_faults	PMemF	total number of out of physical memory faults (LVE version >= 6)
total_nproc_faults	NprocF	total number of max processes faults (LVE version >= 6)
total_io_faults	IOf	total number of max io faults (LVE version >= 6)
iops_avg	aIOPS	average io operations (LVE version >= 8)

iops_max	mIOPS	max io operations (LVE version >= 8)
total_iops_faults	IOPSf	total number of max io operations faults (LVE version >= 8)
any_faults	anyF	total number of faults of all types

-b ALIAS [ALIAS ...]	show LVEs with usage (averaged) within 90 percent of the limit
-by-usage ALIAS [ALIAS ...]	available values:

ALIAS	ALIAS	ALIAS	DESCRIPTION
cpu_avg	cpu	aCPU	average CPU usage
cpu_max	cpu_max	mCPU	max CPU usage
vmem_avg	vmem	aVMem	average virtual memory usage
vmem_max	vmem_max	mVMem	max virtual memory usage
mep_avg	mep	aEP	average number of entry processes (concurrent connections)
mep_max	mep_max	mEP	max number of entry processes (concurrent connections)
pmem_avg	pmem	aPMem	average physical memory usage (LVE version >= 6)
pmem_max	pmem_max	mPMem	max physical memory usage (LVE version >= 6)
nproc_avg	nproc	aNproc	average number of processes (LVE version >= 6)

nproc_max	nproc_max	mNproc	max number of processes (LVE version >= 6)
io_avg	io	aIO	average io usage (LVE version >= 6)
io_max	io_max	mIO	max io usage (LVE version >= 6)
iops_avg	iops	aIOPS	average io operations (LVE version >= 8)
iops_max	iops_max	mIOPS	max io operations (LVE version >= 8)

-p 0..100, -percentage 0..100	defines percentage for -by-usage option; default 90
-style {user,admin}	deprecated, not used.
-l LIMIT, -limit LIMIT	max number of results to display, 10 by default, if 0 no limit
-c [PATH], -csv [PATH]	save statistics in CSV format; "-" by default (output to screen)
-j, -json	display output in JSON format
-server_id SERVER_ID	used with central database for multiple servers, default "local-host"
-servers-info	Show servers LVE versions"
-show-all	full output (show all limits); brief output is default; equivalent -show-columns all
-show-columns COLUMN_NAME [COLUMN_NAME ...]	show only the listed columns; "all" for all supported columns

	COLUMN_NAME	DESCRIPTION
	From	Show start period statistics
	To	Show end period statistics
	ID	LVE Id or username
	aCPU	Average CPU usage
	uCPU	The percentage of user-allocated resource CPU
	mCPU	deprecated
	ICPU	CPU Limit
	CPUf	Out Of CPU usage Faults
	aEP	Average Entry Processes

	uEP	The percentage of user-allocated resource Entry processes
	mEP	deprecated
	IEP	maxEntryProc limit
	aVMem	Average Virtual Memory Usage
	uVMem	The percentage of user-allocated resource Virtual Memory
	mVMem	deprecated
	IVMem	Virtual Memory Limit
	VMemF	Out Of Memory Faults
	EPf	Entry processes faults
	aPMem	Average Physical Memory Usage (LVE version >= 6)

	uPMem	The percentage of user-allocated resource Physical Memory (LVE version >= 6)
	mPMem	deprecated (LVE version >= 6)
	IPMem	Physical Memory Limit (LVE version >= 6)
	aNproc	Average Number of processes (LVE version >= 6)
	uNproc	The percentage of user-allocated resource Number of processes (LVE version >= 6)
	mNproc	deprecated (LVE version >= 6)
	INproc	Limit of Number of processes (LVE version >= 6)
	PMemF	Out Of Physical Memory Faults (LVE version >= 6)
	NprocF	Number of processes faults (LVE version >= 6)
	aIO	Average I/O (LVE version >= 6)

uIO	The percentage of user-allocated resource I/O (LVE version >= 6)
mIO	deprecated (LVE version >= 6)
lIO	I/O Limit (LVE version >= 6)
IOF	Out Of I/O usage Faults (LVE version >= 6)
aIOPS	Average I/O Operations (LVE version >= 8)
mIOPS	deprecated (LVE version >= 8)
uIOPS	The percentage of user-allocated resource I/O Operations (LVE version >= 8)
lIOPS	I/O Operations Limit (LVE version >= 8)
IOPSf	Out Of I/O Operations Faults (LVE version >= 8)

-time-unit TIME_UNIT	time step for grouping statistic in minutes; 1 min., by default; can use mlhd suffixes; for example: 1h or 1h30m or 1d12h
-m {v1,v2}, -compat {v1,v2}	v1 - return old output mode; v2 - new mode; default v1; you can change default in config
-blank-value [BLANK_VALUE]	Use to fill unsupported limits; default “-“
-f YYYY-MM-DD[HH:MM], -from YYYY-MM-DD[HH:MM]	run report from date and time in [YY]YY-MM-DD[HH:MM] format; if not present last 10 minutes are assumed
-t YYYY-MM-DD[HH:MM], -to YYYY-MM-DD[HH:MM]	run report up to date and time in [YY]YY-MM-DD[HH:MM] format; if not present, reports results up to now
-period 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
-by-fault ALIAS [ALIAS ...]	show LVEs which failed on max processes limit or memory limit

	ALIAS	ALIAS	ALIAS	DESCRIPTION
	mcpu	cpu	CPUf	total number of max CPU usage faults
	mem	vmem	VMemF	total number of out of virtual memory faults
	mep	ep	EPf	total number of max entry processes faults
	pmem	pmem	PMemF	total number of out of physical memory faults (LVE version >= 6)
	nproc	nproc	NprocF	total number of max processes faults (LVE version >= 6)
	io	io	IOF	total number of max io faults (LVE version >= 6)
	iops	iops	IOPSf	total number of max io operations faults (LVE version >= 8)
	any_faults	any	anyF	total number of faults of all types

-r FAULTS, -threshold FAULTS	in combination with -by-fault, shows only LVEs with number of faults above; default 1
------------------------------	---

Prefixes Kb, Mb and Gb indicates powers of 1024.

*All ALIAS options are not case sensitive.

lvechart

/usr/sbin/lvechart - creates a chart representing LVE usage for user.

Usage: /usr/sbin/lvechart [OPTIONS]

Acceptable options are:

<code>-help</code>	This help screen
<code>-version</code>	Version number
<code>-from</code>	Run report from date and time in YYYY-MM-DD HH:MM format (if not present, last 10 minutes are assumed)
<code>-to=</code>	Run report up to date and time in YYYY-MM-DD HH:MM format (if not present, reports results up to now)
<code>-period=</code>	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
<code>-id=</code>	LVE id – will display record only for that LVE id
<code>-user=</code>	Use username instead of LVE id, and show only record for that user
<code>-server=</code>	Server id – will display record for that server, instead of default (current)
<code>-output=</code>	Filename to save chart as, if not present, output will be sent to STDOUT
<code>-show-all</code>	Show all graphs (by default shows graphs for which limits are set)
<code>-style=</code>	{admin user} Set chart style, CPU limits are normalized to 100% in user's style
<code>-format=</code>	{svg png} Set chart output format.

dbgovchart

`/usr/sbin/dbgovchart` - creates a chart representing MySQL usage for user.

Usage: `/usr/sbin/dbgovchart [OPTIONS]`

Acceptable options are:

<code>-help</code>	This help screen
<code>-version</code>	Version number
<code>-from=</code>	Run report from date and time in YYYY-MM-DD HH:MM format (if not present, last 10 minutes are assumed)
<code>-to=</code>	Run report up to date and time in YYYY-MM-DD HH:MM format (if not present, reports results up to now)
<code>-period=</code>	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
<code>-user=</code>	mysql username
<code>-output=</code>	Filename to save chart as, if not present, output will be sent to STDOUT
<code>-show-all</code>	Show all graphs (by default shows graphs for which limits are set)
<code>-server=</code>	Server id – will display record for that server, instead of default (current).
<code>-style=</code>	{admin user} Set chart style, CPU limits are normalized to 100% in user's style
<code>-format=</code>	{svg png} Set chart output format.

lve-read-snapshot

usage: lve-read-snapshot [-h] [--version] [-f FROM [FROM ...]] [-t TO [TO ...]]

[-p PERIOD] [--timestamp TIMESTAMP]

[-i ID] [-u USER] [-l] [-o file] [-j]

[-stats]

[-unit unit]

Reads lve system state snapshots for LVE/user

optional arguments:

-h, --help show this help message and exit

---version Version number

-f FROM [FROM ...], --from FROM [FROM ...]

Run report from date and time in YYYY-MM-DD

HH:MM

format, if not present last 10 minutes are

assumed

(default: 2016-10-24 19:28)

-t TO [TO ...], --to TO [TO ...]

Run report up to date and time in YYYY-MM-DD

HH:MM

format, if not present, reports results up to

now

(default: 2016-10-24 19:38)

-p PERIOD, --period 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 (default: 10m)

---timestamp TIMESTAMP

time stamp in unix format for get one snapshot

(default: None)

-i ID, --id ID LVE id to show records for (default: None)

-u USER, --user USER user account to show records for (default: None)

-l, --list show timestamp list only (default: False)

-o file, --output file
 Filename to save snapshots report to, if not present, output will be sent to STDOUT (default: None)

-j, --json
 Output in json format (default: False)

--stats
 Output stats, instead of snapshots (default: False)

-unit unit
 Group stats by time unit. Example values 3h, 24h, 1d, 1w. Other possible value is “auto” for grouping by each incident. (default: 1d)

One of -u --user or -i --id should be specified

live-create-db

```
usage: lve-create-db [-h] [--recreate] [--print-sql]
```

`[-update-serverid-prompt] [-update-serverid-auto]`

`[-validate]`

Creates a database for live-stats

optional arguments:

-h, -help show this help message and exit

—recreate	Drops and recreates database even if tables exists
-----------	--

(default: False)

—print-sql Prints sql and exits, without creating db

(default:

False)

—update-serverid-prompt

Update exist server ID or create new one

(default:

False)

—update-serverid-auto

Update exist server ID with uuid (default:

False)

`-validate` Check the correctness of the database structure
(default: False)

cloudlinux-top

- Usage
- Output format
- Units of measurement
- Errors handling
- Examples

Utility provides information about current MySQL and LVE usage of a running system in JSON format.

‘<>’__Usage

cloudlinux_top [-h] [-v] [-j] [-hide-mysql]

[-u USERNAME | -r FOR_RESELLER] [-d DOMAIN] [-m MAX]

[-o ORDER_BY]

Optional arguments.

-h, --help	show this help message and exit
-v, --version	show program version number and exit
-j, --json	return data in JSON format
--hide-mysql	don't show MySQL related info
-u USERNAME, --user-name USERNAME	show data only for a specific user. Can be used to filter the output; returns users with username "%USERNA ME%"
-r FOR_RESELLER, --for-reseller FOR_RESELLER	get information only about specified reseller and his users
-d DOMAIN, --domain DOMAIN	show data only for a specific domain . Can be used to filter the output; returns users with domain "%DOMAIN%"
-m MAX, --max MAX	show up to N records. If --max key is omitted. By default will show top 25 users
-o ORDER_BY, --order-by ORDER_BY	sort output by resource usage; avail able options: "cpu", "mysql_cpu", "io", "mysql_i o", "iops", "ep", "nproc", "pmem"

‘<>’__Output format

```
{
  "mySqlGov": "enabled",      # possible values: enabled, error

  "mySqlGovMode": "abusers",  # see "MySQL Governor > Modes Of Operation"
                              # if MySQL Governor is not enabled, value is "none"

  "resellers": [              # list of resellers (available only with
                              # reseller limits feature)
    {
      "id": 1000020005,        # internal record id
      "limit": <lve_section>,  # current limits (last 5 seconds)
```

```
    "name": "reseller_name", # reseller's login in control panel
    "usage": <lve_section>  # current usage (last 5 seconds)
  }
],
"result": "success",        # see the 'errors handling' section
"timestamp": 1522858537.337549,
"users": [
  {
    "domain": "domain.com", # user's primary domain (from control panel)
    "id": 20005,            # lve_id, same as user id in /etc/passwd file
    "limit": <lve_section>, # limits for last 5 seconds
    "reseller": "reseler1", # user's reseller (from control panel)
    "usage": <lve_section>, # usage for last 5 seconds
    "username": "user"      # username from /etc/passwd file or "N/A" if user
                           # with such id does not exist
  }
]
}
```

The structure* of <lve_section>:

```
{
  "cpu": {
    "all": 50.0, # CPU usage or limit (LVE only)
    "mysql": 0.0* # CPU usage or limit (MySQL Governor only)
  },
  "ep": 1.0, # number of entry processes
  "io": {
    "all": 0.0, # IO usage or limit (LVE only)
    "mysql": 0.0** # IO usage or limit (MySQL Governor only)
  },
  "iops": 0.0, # IO operations per second
  "mem": 258048, # memory usage or limit
  "pno": 1.0 # number of processes
}
```

* you can modify this structure using `--show` option, see *usage examples* for details.

** mysql values are only present when MySQL Governor statistics is available and `--hide-mysql` options is not used.

‘<>’ __Units of measurement

For limits and usage sections we use the following units of measurement.

Value	Units of measurement
cpu (lve and mysql)	percentage of one CPU core
io (lve and mysql)	bytes per second
iops	number of IO operations per second
mem	bytes
ep	number of entry processes
pno	number of processes

‘<>’__Errors handling

The format of the error message is the same as in the other cloudlinux- * utilities. When everything is ok, the result value is success. Otherwise, it contains error message. In case of unexpected errors, the output will be as follows.

```
# cloudlinux-top -json
{
    "context": {
        "error_text": "Very bad error"
    },
    "result": "An error occurred: "%(error_text)s"",
    "timestamp": 1523871939.639394
}
```

‘<>’__Examples

- get 100 users ordered by CPU usage

```
# cloudlinux-top -json -order-by cpu -max=100
```

- get information about one user

```
# cloudlinux-top -json -u username
```

- get information about reseller and his users

```
# cloudlinux-top -json -for-reseller=reseller_name
```

- show only IO limits and usage

```
# cloudlinux-top -json -show=io
```

cloudlinux-statistics

- Usage
- Output format
- Units of measurement
- Errors handling
- Examples

cloudlinux-statistics is a CLI utility that provides historical information about resource usage.

‘<>’__Usage

```
cloudlinux-statistics [-h] [-j] [-v] [--by-usage BY_USAGE]
```

`[-percentage 0..100] [-by-fault BY_FAULT]`
`[-threshold THRESHOLD] [-server_id SERVER_ID]`
`[-f FROM] [-t TO] [-period PERIOD]`
`[-limit LIMIT]`
`[-show COLUMN_NAME [COLUMN_NAME ...]]`
`[-o ORDER_BY] [-id ID] [-time-unit TIME_UNIT]`
`[-r FOR_RESELLER]`

Optional arguments.

-h, --help	show this help message and exit
-j, --json	return data in JSON format
-v, --version	show program version number and exit
--server_id SERVER_ID, --server-id SERVER_ID	can be used with the central database for multiple servers; default “...”
--limit LIMIT	limit the number of results to display, 0 is unlimited
--show COLUMN_NAME [COLUMN_NAME ...]	<p>show only listed columns; “all” for all supported columns (fields)</p> <pre>ent: 0px; padding: 0px 0px 0px 0px; margin: 0px 0px 0px 0px;”> +-----+ +-----+ Key Fields to show +-----+ +-----+ all all available fields +-----+ +-----+ cpu CPU field +-----+ +-----+ io IO field +-----+ +-----+ iops IOPS field +-----+ +-----+ ep entry processes (concurrent connections) field +-----+ +-----+ nproc number of processes field +-----+ +-----+ pmem physical memory field +-----+ +-----+ vmem virtual memory field +-----+ +-----+ mysql mysql_cpu & mysql_io field +-----+ +-----+ </pre>
--o ORDER_BY, --order-by ORDER_BY	<p>order results by one of the following keys (fields):</p> <pre>ent: 0px; padding: 0px 0px 0px 0px; margin: 0px 0px 0px 0px;”> +-----+ +-----+ FIELD DESCRIPTION +-----+ +-----+ any_faults total number of faults of all t </pre>
	<pre>types +-----+ +-----+ cpu average CPU usage +-----+ </pre>

Filter items by resource usage.

<code>-by-usage BY_USAGE</code>	<p>show LVEs with usage (averaged) within 90 percent of the limit available values ent: 0px; padding: 0px 0px 0px 0px; margin: 0px 0px 0px 0px;"></p> <pre> +-----+ +-----+ FIELD +-----+ +-----+ DESCRIPTION +-----+ +-----+ +-----+ cpu +-----+ +-----+ average CPU usage +-----+ +-----+ +-----+ mysql_cpu +-----+ +-----+ average MySQL CPU usage +-----+ +-----+ +-----+ io +-----+ +-----+ average IO usage +-----+ +-----+ +-----+ mysql_io +-----+ +-----+ average MySQL IO usage +-----+ +-----+ +-----+ iops +-----+ +-----+ average IO operations; (LVE ver +-----+ sion +-----+ >= 8) +-----+ +-----+ +-----+ ep +-----+ average number of entry process +-----+ es +-----+ (concurrent connections) +-----+ +-----+ +-----+ nproc +-----+ average number of processes +-----+ +-----+ +-----+ pmem +-----+ average physical memory usage +-----+ +-----+ +-----+ vmem +-----+ average virtual memory usage +-----+ +-----+ +-----+ </pre>
<code>-percentage 0..100</code>	define percentage for <code>-by-usage</code> option; default 90

Filter items by the number of faults.

<p>–by-fault BY_FAULT</p>	<p>show only accounts that have some faults for the given limit ent: 0px; padding: 0px 0px 0px 0px; margin: 0px 0px 0px 0px;”> + —+—————+ FIELD DESCRIPTION + —+—————+ any faults of all types + —+—————+ cpu CPU usage faults + —+—————+ io max IO usage faults + —+—————+ iops max IO operations faults; (LVE version >= 8) + —+—————+ ep max entry processes faults + —+—————+ nproc max processes faults + —+—————+ pmem out of physical memory faults + —+—————+ vmem out of virtual memory faults + —+—————+ </p>
<p>–threshold THRESHOLD</p>	<p>in combination with –by-fault, shows only accounts with the number of faults more than given; default 1</p>

Filter items by a time interval.

Allows to get information for the given period of time; you can either set `-from` and `-to` options, or just get information for the recent time period using `-period` option.

–from and –to values are ignored when –period is set.

-f FROM, -from FROM	run report from date and time in [YY]YY-MM-DD[HH:MM] format; if not present, last 10 minutes are assumed
-t TO, -to TO	run report up to date and time in [YY]YY-MM-DD[HH:MM] format; if not present, reports results up to now
-period PE-RIOD	time period; specify minutes with m, hours with h, days with d, and values: today, yesterday; 5m - last 5 minutes, 4h - last four hours, 2d - last 2 days, and today

Get detailed statistics.

<code>-id ID</code>	get detailed statistics for database record with the given id
<code>-time-unit TIME_UNIT</code>	group statistics using the given time; 1 minute by default. For example: 1h or 1h30m or dynamic; available only in pair with <code>-id</code>

‘<>’ __Output format

There are two different JSON formats used for summary statistics and detailed statistics.

Summary statistics

```
# cloudlinux-statistics -json
{
  "resellers": [
    {
      "usage": <lve_section>,
      "faults": <lve_section>,
      "name": "reseller",
      "limits": <lve_section>,
      "id": 1000020005
    }
  ],
  "timestamp": 1522920637,
  "mySqlGov": "enabled",      # possible values: "enabled", "error"
  "result": "success",
  "users": [
    {
      "username": "username",
      "domain": "example.com",
      "reseller": "reseller",
      "limits": <lve_section>,
      "faults": <lve_section>,
      "usage": <lve_section>,
      "id": 20005
    }
  ]
}
```

Detailed statistics

```
# cloudlinux-statistics -json -id=20001
{
  "timestamp": 1523011550,
  "mySqlGov": "enabled",      # possible values: "enabled", "error"
  "result": "success",
  "user": [
```

```
{
  "usage": <lve_section>,
  "faults": <lve_section>,
  "from": 1523011144,
  "limits": <lve_section>,
  "to": 1523011143
},
...
{
  "usage": <lve_section>,
  "faults": <lve_section>,
  "from": 1523011204,
  "limits": <lve_section>,
  "to": 1523011203
}
]
```

For both, summary statistics and detailed statistics, <lve_section> is the same and looks like following*.

```
{
  "ep": {
    "lve": 1      # number of entry processes
  },
  "vmem": {
    "lve": 2428928 # virtual memory usage or limit (deprecated)
  },
  "iops": {
    "lve": 0      # io operations per second
  },
  "io": {
    "lve": 0.0,   # io usage or limit (lve only)
    "mysql": 0.0** # io usage or limit (mysql only)
  },
  "nproc": {
    "lve": 1      # number of processes in lve
  },
  "cpu": {
    "lve": 25.6,  # cpu usage (lve only)
    "mysql": 0.0* # cpu usage (mysql governor only)
  },
  "pmem": {
    "lve": 360448 # physical memory usage or limit
  }
}
```

* you can specify only required fields using `--show` option;

** mysql fields are only available with *MySQL Governor* installed.

‘<>’ __Units of measurement

For limits and usage sections we use the following units of measurement.

Value	Units of measurement
cpu (lve and mysql)	percentage of one CPU core
io (lve and mysql)	bytes per second
iops	number of IO operations per second
pmem and vmem	bytes
ep	number of entry processes
nproc	number of processes in LVE

‘<>’ __Errors handling

The format of the error message is the same as in the other cloudlinux- * utilities. When everything is ok, the result value is success. Otherwise, it contains error message.

```
# cloudlinux-statistics --json
```

```
{
    "context": {
        "error_text": "Very bad error"
    },
    "result": "An error occurred: "%(error_text)s"",
    "timestamp": 1523871939.639394
}
```

‘<>’ __Examples

- get top 10 users ordered by CPU usage for today

```
# cloudlinux-statistics --json --order-by=cpu --period=today --limit=10
```

- get users that hit IO limit more than 10 times for today

```
# cloudlinux-statistics --json --period=today --by-fault=io --threshold=10
```

- get users that used more than 80% of CPU in last 24 hours

```
# cloudlinux-statistics --json --by-usage=cpu --percentage=80 --period=24h
```

- get information only about reseller and his users

```
# cloudlinux-statistics --json --for-reseller=reseller_name
```

- get information only about CPU and IO usage

```
# cloudlinux-statistics --json --show=cpu,io
```

Plugins

LVE Stats 2z comes with a set of generic plugins:

Plugin Name	Order	Default	Period (seconds)	Description
LVECollector	1000	Y	5	Collects usage/limits data from /proc/lve/list
CPUInfoCollector	2000	Y	5	collects info about CPU - /proc/cpuinfo
LVEUsernamesCollector	3000	Y	3600	collects usernames & user ids to match uid <-> lve id later on
LVEUsageAnalyzer	4000	Y	5	analyzes usage of LVE
LveUsageAggregator	5000	Y	60	aggregates data by time periods
DBGovSaver	6000	Y	5	Saves data about database governor
FileSaver	7000	Y	5	Saves LVE data into /var/lve/info
CloudLinuxTopFileSaver	8000	Y	60	saves data used by cloudlinux-top to /var/lve/cloudlinux-top.json
DBSaver	9000	Y	60	save LVE data to database
DbUsernamesSaver	10000	Y	3600	saves users name to database
DBSaverX60	11000	Y	3600	saves aggregated hourly data into database
SnapshotSaver	12000	Y	30	collects & saves snapshots data
StatsNotifier	13000	Y	varied	notify user/admin based on usage
HistoryCleaner	14000	Y	3600	removes old usage
ResMEMCollector	1500	N	30	collects physical memory usage from processes RES field instead of /proc/lve/list
LVEDestroyer	.	N	5	destroys LVEs that weren't active for X iterations. Number of iterations is passed from config using iterations variable. iterations=0 means plugin disabled

To enable non-default plugin, copy or link it to /usr/share/lve-stats/plugins directory.

For example to enable ResMEMCollector plugin, do:

```
ln -s /usr/share/lve-stats/plugins.other/res_mem_collector.py /usr/share/lve-stats/plugins/  
service lvestats restart
```

Creating a Plugin for LVE Stats 2

- *Introduction*
- *Server Plugin Arrangement*
- *Plugin Configuration*
- *Types of Plugins*

Introduction

LVE Stats 2 complex has scalable architecture, which allows to connect custom plugins.

General Information

LVE Stats server searches for plugins in the directory which is specified with `plugins` parameter of server's `/etc/sysconfig/lvestats2` configuration file. Default directory is `/usr/share/lve-stats/plugins`.

Each plugin must be of a Python class, must be written in Python language and its file must have `.py` extension. Files with all other extensions will be ignored. For normal server work access permission 400 is enough; owner – root.

Plugins' classes can be of the same name, but better not, because classes' names can affect the set of parameters in `set_config` method. You can find detailed plugins' configuring information below, in appropriate chapter.

Plugin's class must contain `execute()` method, which is invoked by the server every 5 seconds (by default, can be changed by `interval` parameter of configuration file).

Also `set_config` method (configuration settings) can be available. You can find more info in *Plugins Configuration* chapter.

Additionally the following attributes can be set (plugin class instance variable):

- `order` (integer) - defines plugin's position in the server's plugin

list, (more info in *Servers Plugin Arrangement*).

- `timeout` (integer or float) – the longest allowable duration of one

launch of the plugin (`execute` method). Default value of `timeout` parameter is 5 seconds.

- `period` (integer) – sets the interval between two launches of `execute`

plugin method in seconds. If not defined, then plugin runs every 5 seconds (`interval` parameter in configuration file).

When `execute()` method of the plugin is invoked, the server creates an attribute `now` in it, where launch time is recorded. This value is equal to what a standard Python function `time.time()` returns. All the plugins launched one after another receive the same value of `now` attribute from the server. `now` is overwritten before `execute()` method is invoked.

The previous value of `now` attribute is not saved by the server. If plugin needs it, it has to save it by itself.

Plugin's class can be inherited from `LveStatsPlugin` class, which is the part of the server itself. This is not obligatory, but inheritance can help to avoid different errors in servers work, particularly if a plugin doesn't contain required `execute` method.

`LveStatsPlugin` class is defined in the file: `/opt/alt/python27/lib/python2.7/site-packages/lvestats/core/plugin.py`.

Server Plugin Arrangement

When the server starts, it performs the search of plugins in the directory specified in `/etc/sysconfig/lvestats2` configuration file. This directory is scanned only when the server starts, therefore if any plugin was added into the directory, the server has to be restarted with the following command:

```
service lvestats restart.
```

After successful restart the plugins are graded and executed ascending by order attribute. If any plugin's order attribute is not set, it is considered as a Python language constant `sys.maxint` (which is usually 9223372036854775807). This in fact means that such plugins will be executed in the last.

If any plugins has similar order meanings, their execution order is unpredictable.

The server invokes `execute` method of all plugins one after another.

When the server invokes `execute()` method of any plugin, it transmits a data dictionary (`lve_data` argument) into plugin. The dictionary is common for all the plugins. Any plugin can read, write and change any data in this dictionary. LVE Stats 2 server doesn't control this area. That is why one must be careful while developing new plugins, in order not to change or corrupt other plugins' data which can break their functionality.

If an exception occurs in `execute()` method, its text and python stack trace is recorded into server log `/var/log/lve-stats` and all the changes made to `lve_data` dictionary before the exception happened are lost.

The keys of the `lve_data` dictionary are recommended to look like "PluginName_Key", in order the plugins do not corrupt other data accidentally.

Server contains some standard plugins which define and use the following keys in the common dictionary `lve_data`: `LVE_VERSION`, `stats`, `old_stats`, `procs` and `lve_usage`. User plugins can use data from these keys, but it is recommended not to change them if there is no special need, because it can break the next plugins in the execution queue.

Key	Content
<code>LVE_VERSION</code>	Version. The same as console command <code>lvectl lve-version</code> produces.
<code>stats</code>	Dictionary, that contains <code>lve_id</code> 's as keys and <code>LVEStat</code> class objects as values. Every <code>LVEStat</code> object contains values of usages and limits taken from <code>/proc/lve/list</code> file for every LVE Id. Dictionary keys – integer <code>lve_id</code> , including 0 for "default" LVE. This dictionary is updated on each iteration of <code>lvestats-server</code> (every 5 seconds by default). <code>LVEStat</code> – is a standard server class, it can be imported with the command from <code>lvestats.core.lvestat import LVEStat</code> . The class is described in the file <code>/opt/alt/python27/lib/python2.7/site-packages/lvestats/core/lvestat.py</code> . Here you can find the whole list of data fields and their functions.
<code>old_stats</code>	Stats content from the previous iteration. Before the first iteration – empty dictionary.
<code>totalHz</code>	When <code>LVE_VERSION</code> is 4, real CPU frequency in Hz multiplied by number of cores. When <code>LVE_VERSION</code> > 4, CPU speed is in conventional units and equals to <code>1000000000 * cores</code> (1 GHz per core).
<code>procs</code>	Quantity of CPU/cores.
<code>lve_usage</code>	Contains accumulated LVE statistics for each 5-seconds interval in current minute. Cleared each minute.
<code>lve_usage</code>	Contains aggregated LVE Statistics for "previous" minute to store to database. Overwritten each minute.

Each plugin's instance lifetime is from the moment it was loaded till the server stops working. But if `execute()` method working time exceeds timeout, the plugin will be terminated and restarted in the next iteration. All changes to the `lve_data` dictionary will be lost.

During servers graceful shutdown (restart, server shutdown, commands `service lvestats stop`, `service lvestats restart`), each plugin receives `SIGTERM` signal.

This is useful to correctly unload the plugin (terminate all subsidiary processes, save data to files etc.). If a plugin doesn't need to "finalize" its execution before termination, then there's no need to implement this signal handler.

Below you can see an example of such handler.

Note: If a plugin implements handler for SIGTERM, then this handler must end with `sys.exit(0)` command. Otherwise plugin process will not be terminated correctly and will become orphaned.

Plugin Configuration

LVE Stats 2 Server allows to configure each plugin separately.

On initialization stage the server invokes `set_config()` method of the plugin and locates there a dictionary which contains:

- all parameters from file `/etc/sysconfig/lvestats2` (global).
- plugin's individual configuration file parameters (if one exists). Configuration files must be located in `/etc/sysconfig/lvestats.config` directory, have `.cfg` extension and be the same format as `/etc/sysconfig/lvestats2`. Files in this directory are matched with the plugins by name. For instance, if plugin's class is `Plugin1_class`, then server will try to find and download `/etc/sysconfig/lvestats.config/Plugin1_class.cfg`. Names are case sensitive. If any plugin doesn't have an individual configuration file, then it's not an error. In this case plugin will just receive parameters from `/etc/sysconfig/lvestats2`.

Note. An individual configuration file of every plugin is loaded after server configuration file. That is why if it contains any parameters with names similar to ones of server config, then plugin will use parameters from its individual config rather than server config parameters.

If a plugin doesn't require any configuration to be done, then `set_config` method can be skipped.

In addition, plugins can use their own configuration methods.

Types of Plugins

According to server architecture, plugins can be of the following types:

- collectors
- analyzers
- persistors
- notifiers

Collectors are designed to collect information; analyzers – to analyze it and form some other data on its basis; persistors – to save information from the common dictionary into files, databases, etc.; notifiers - to notify system users about any events.

This division is rather arbitrary. There is an opportunity to execute all the actions on collection, analysis and saving the information in one and only plugin. But at the same time the division into functionally independent parts allows to build flexible and easily configurable system for collecting and processing the data.

Also it is possible to implement the systems of lazy-write, planning of collecting/processing tasks and notifying users about different events.

Examples of Plugins

Here is a practical example of a user plugin.

Specification:

1. To trace specified file size changes. The name of file being traced must be specified in configuration file, which allows to change it without modifying the plugin itself. If file size has been changed, it has to be written as a message into our log. The name of log must be specified in configuration file as well.
2. File size must be checked with default interval (5 seconds), and log notification must be held once a minute (to avoid resource expend for possibly regular write).
3. System administrator must receive emails with file size at the moment the email was sent. These notifications must be sent even if the file size hasn't been changed. Notification emails must be read periodicity from configuration file as well as sender/receiver emails .

As file size check, fixing the result and notification sending must be held with different periods, then it's impossible to realize all the tasks by means of one plugin.

The fact that one minute (60 seconds) is multiple to 5 seconds doesn't matter in this case, because default period can be changed in server's configuration file, but the condition of fixing the result once a minute is a part of the specification, which can not be violated. In addition, notification email period is known in advance, as it is specified by user in configuration file.

That is why we realize 4 plugins: collector, analyzer, persistor and notifier.

Collector

Collector's aim is to determine the size of a proper file.

```
# FSize_watcher_collector.py
# Example plugin for monitoring file size.
# Part 1. Collector

import os

from lvestats.core.plugin import LveStatsPlugin

# Key name
COLLECTOR_KEY = 'FSizeWatcher_fsize'
COLLECTOR_KEY_FILENAME = 'FSizeWatcher_fname'

class FSize_watcher_collector (LveStatsPlugin):
    # this plugin should be first in chain
    order = 0

    # File to monitoring
    file_to_monitoring = None

    def __init__(self):
        pass

    # Sets configuration to plugin
    def set_config(self, config):
        self.file_to_monitoring = config.get('file_to_monitoring', None)
        pass

    # Work method
    def execute(self, lve_data):
```

```
try:
    # if monitoring file absent, do nothing
    if self.file_to_monitoring is None or not os.path.exists(self.file_to_monitoring):
        return
    # Get file size
    stat_info = os.stat(self.file_to_monitoring)
    fsize = stat_info.st_size
    # Place file name and file size to server data dictionary
    lve_data[COLLECTOR_KEY_FILENAME] = self.file_to_monitoring
    lve_data[COLLECTOR_KEY] = fsize
except (OSError, IOError):
    # file absent or any other error - remove size from dictionary
    del lve_data[COLLECTOR_KEY]
```

Plugin algorithm is extremely simple – file size is read and written into data dictionary. Files name is read from `set_config` method configuration. If the name is not specified, then `None` is written into appropriate variable. All the errors are completely ignored (e.g. if specified file doesn't exist or there's no way to read any of it's information).

order attribute is specified as 0 to make this plugin go the first among three. Data collector must always be the first in plugins logical chain, because it provides all the necessary information for the analyzer which goes the next. Specific values of order can be of any kind, but what is important is that when the server starts, all the plugins line up in proper sequence: collector – analyzer – persistor.

In order to make plugin work, we have to create configuration file `/etc/sysconfig/lvestats.config/FSize_watcher_collector.cfg` with the following content:

```
# Config file for FSize_watcher_collector plugin
# Please define monitoring file here
#file_to_monitoring = /usr/local/cpanel/logs/error_log
file_to_monitoring = /usr/local/cpanel/logs/access_log
```

Note that file's name `FSize_watcher_collector` without `.cfg` extension matches plugin class name.

`file_to_monitoring` option is read by plugin in `set_config` method and contains file's full name for monitoring.

Files for monitoring, suggested in the actual example - `/usr/local/cpanel/logs/error_log` and `/usr/local/cpanel/logs/access_log` - are real, these are cPanel control panel logs.

The first file is errors log; the second is appeal log, is refreshed during common work with panel (e.g. if user email address is changed).

Errors log tracking is more important, but appeal log monitoring allows to illustrate plugins work more in details, because it is refreshed more often.

Note that plugin can monitor one file only.

Analyzer

Analyzer decides if the file's size has changed and gives a command to persistor to refresh log.

```
# FSize_watcher_analyzer.py
```

```
# Example plugin for monitoring file size.
# Part 2. Analyzer
from lvestats.core.plugin import LveStatsPlugin
# Key name from collector plugin
COLLECTOR_KEY = 'FSizeWatcher_fsize'
# Key name 1 for saver plugin
SAVER_KEY = 'FSizeWatcher_fsize_to_store'
# Key name 2 for saver plugin
SAVER_DATA_PRESENCE = 'FSizeWatcher_fsize_present'
class FSize_watcher_analyzer (LveStatsPlugin):
    # this plugin should be second in chain
    order = 1
    # Last file size
    file_last_size = 0
    # Plugin run period in seconds
    period = 60
    def __init__(self):
        pass
    # work method
    def execute(self, lve_data):
        # Default setting for saver
        lve_data[SAVER_DATA_PRESENCE] = 0
        # Check presence data
        if COLLECTOR_KEY not in lve_data:
            return
        # Get file size from server data dictionary
        fsize = lve_data[COLLECTOR_KEY]
        # Check, if file size changed, store it for saver plugin
        if fsize == self.file_last_size:
            return
        # Put new size for saver plugin
        lve_data[SAVER_KEY] = fsize
        self.file_last_size = fsize
        lve_data[SAVER_DATA_PRESENCE] = 1
```

This plugin is extremely simple as well. It starts after collector (order=1), searches for file size in the dictionary and compares it with the previous index. If it has changed, then it writes a sign of presence of a new size into the dictionary.

If no changes seen, then sign resets. The previous file size is stored in the plugin itself in `file_last_size` variable. Note that they are stored during the whole server lve-stats lifetime.

If file size is not found in data dictionary, then plugin just ends.

All the errors are completely ignored.

Analyzer is unconfigurable, that is why it doesn't require any configuration file and it doesn't contain `set_config` method.

Plugin starts every 60 seconds (1 minute), because we need data fixation to be performed one time in a minute.

Persistor

Persistor saves information from the common dictionary into files, databases, etc.

```
# FSize_watcher_saver.py
```

```
# Example plugin for monitoring file size and last modification date-time.
```

```
# Part 3. Data saver
```

```
import signal
```

```
import sys
```

```
import time
```

```
from lvestats.core.plugin import LveStatsPlugin
```

```
# Key name 1 for saver plugin
```

```
SAVER_KEY = 'FSizeWatcher_fsize_to_store'
```

```
# Key name 2 for saver plugin
```

```
SAVER_DATA_PRESENCE = 'FSizeWatcher_fsize_present'
```

```
# Monitoring file name
```

```
COLLECTOR_KEY_FILENAME = 'FSizeWatcher_fname'
```

```
class FSize_watcher_saver(LveStatsPlugin):
```

```
    # this plugin should be third in chain
```

```
    order = 2
```

```
    # Plugin run period in seconds
```

```
    period = 60
```

```
    # Log filename
```

```
    log_file_name = None
```

```
    # First run flag
```

```
    is_first_run = True
```

```
    def __init__(self):
```

```
        signal.signal(signal.SIGTERM, self.sigterm_handler)
```

```
    # Sets configuration to plugin
```

```
    def set_config(self, config):
```

```
# Get log filename
self.log_file_name = config.get('log_filename', None)

# work method
def execute(self, lve_data):
    # do nothing, if log file not defined
    if not self.log_file_name:
        return

    try:
        # Check presence data
        if SAVER_DATA_PRESENCE not in lve_data or lve_data[SAVER_DATA_PRESENCE] == 0:
            # No data
            return

        # Get file size from server data dictionary
        fsize = lve_data[SAVER_KEY]

        # Store data to log
        f = open(self.log_file_name, 'a')

        if self.is_first_run:
            f.write('%s - FSize_watcher started. Monitoring file: %s, saving data period=%d sec\n' %
                (time.asctime(time.localtime()), lve_data[COLLECTOR_KEY_FILENAME], self.period))
            self.is_first_run = False

            f.write('%s - FSize_watcher: file size is %d bytes\n' % (time.asctime(time.localtime()), fsize))
            f.close()

    except:
        # Ignore all errors
        pass

# Terminate handler
def sigterm_handler(self, signum, frame):
    if self.log_file_name:
        try:
            # Store data to log file
            f = open(self.log_file_name, 'a')
            f.write('%s - File watcher saver plugin: TERMINATE\n' % time.asctime(time.localtime()))
            f.close()
        except:
            # Ignore all errors
```

```
pass
# Terminate process
sys.exit(0)
```

Configuration file `/etc/sysconfig/lvestats.config/FSize_watcher_saver.cfg`:

```
# Config file for FSize_watcher_saver.py plugin
# Please define log filename here
log_filename = /var/log/FSize_watcher.log
```

This plugin starts after analyzer (order=2), checks new file size presence flag, and if positive – writes it into log. If the flag is cleared (which means the size hasn't changed), then plugin simply ends.

Starts once in a minute (period=60).

Also this plugin shows the work of signal handler.

Plugin constructor registers handler-function of a proper signal: `signal.signal(signal.SIGTERM, self.sigterm_handler)`. This means, that when the server finishes its work, then `sigterm_handler` method of plugin class will be invoked. In the actual example the function just writes a notification into log, tracing the fact of it's invocation.

Pay attention on `sys.exit(0)` command in the end of the handler. Find the information on it in Server Plugin Arrangement section.

In addition see into examples of file log `/var/log/FSize_watcher.log` formed by the plugins above:

```
Tue Feb 3 13:06:24 2015 - FSize_watcher started. Monitoring file: /usr/local/cpanel/logs/access_log, saving data
period=60 sec
```

```
Tue Feb 3 13:06:24 2015 - FSize_watcher: file size is 122972890 bytes
```

```
Tue Feb 3 13:07:25 2015 - FSize_watcher: file size is 122975507 bytes
```

```
Tue Feb 3 13:08:25 2015 - FSize_watcher: file size is 122978124 bytes
```

```
Tue Feb 3 13:09:25 2015 - FSize_watcher: file size is 122978997 bytes
```

```
Tue Feb 3 13:10:25 2015 - FSize_watcher: file size is 122981033 bytes
```

```
Tue Feb 3 13:11:25 2015 - FSize_watcher: file size is 122982052 bytes
```

```
Tue Feb 3 13:13:25 2015 - FSize_watcher: file size is 122983798 bytes
```

```
Tue Feb 3 13:20:15 2015 - File watcher saver plugin: TERMINATE
```

and

```
Thu Feb 5 13:07:27 2015 - FSize_watcher started. Monitoring file: /usr/local/cpanel/logs/error_log, saving data
period=60 sec
```

```
Thu Feb 5 13:07:27 2015 - FSize_watcher: file size is 14771849 bytes
```

```
Thu Feb 5 14:03:32 2015 - FSize_watcher: file size is 14771995 bytes
```

```
Thu Feb 5 15:01:36 2015 - FSize_watcher: file size is 14772434 bytes
```

```
Thu Feb 5 17:15:47 2015 - FSize_watcher: file size is 14772873 bytes
```

```
Thu Feb 5 18:47:54 2015 - FSize_watcher: file size is 14775213 bytes
```

```
Thu Feb 5 19:11:56 2015 - FSize_watcher: file size is 14775652 bytes
```

```
Thu Feb 5 21:09:05 2015 - FSize_watcher: file size is 14776091 bytes
```



```
Thu Feb 5 23:06:14 2015 - FSize_watcher: file size is 14776530 bytes
Fri Feb 6 00:47:23 2015 - FSize_watcher: file size is 14778870 bytes
Fri Feb 6 01:02:24 2015 - FSize_watcher: file size is 14779309 bytes
Fri Feb 6 02:00:28 2015 - FSize_watcher: file size is 14779434 bytes
Fri Feb 6 03:16:34 2015 - FSize_watcher: file size is 14779873 bytes
Fri Feb 6 05:04:42 2015 - FSize_watcher: file size is 14779998 bytes
Fri Feb 6 05:12:43 2015 - FSize_watcher: file size is 14780437 bytes
Fri Feb 6 05:56:50 2015 - FSize_watcher: file size is 14780551 bytes
Fri Feb 6 06:01:50 2015 - FSize_watcher: file size is 14780975 bytes
Fri Feb 6 06:03:51 2015 - FSize_watcher: file size is 14782183 bytes
Fri Feb 6 06:04:51 2015 - FSize_watcher: file size is 14782575 bytes
Fri Feb 6 06:18:52 2015 - FSize_watcher: file size is 14782647 bytes
Fri Feb 6 06:21:52 2015 - FSize_watcher: file size is 14782898 bytes
Fri Feb 6 06:48:54 2015 - FSize_watcher: file size is 14785238 bytes
Fri Feb 6 07:09:56 2015 - FSize_watcher: file size is 14785677 bytes
Tue Feb 6 08:03:15 2015 - File watcher saver plugin: TERMINATE
```

You can see that log record is being held once a minute (what we actually need), new file size is written.

Also we can notice that handler SIG_TERM was executed, signaling that plugin received the notification about server shut-down.

Notifier

Notifier informs system users about any events.

```
# FSize_watcher_saver.py
# Example plugin for monitoring file size and last modification date-time.
# Part 4. Notifier

import time
import smtplib

from lvestats.lib.common import dateutil
from lvestats.core.plugin import LveStatsPlugin

# Key name
COLLECTOR_KEY_FSIZE = 'FSizeWatcher_fsize'
COLLECTOR_KEY_FILENAME = 'FSizeWatcher_fname'

# email message pattern
EMAIL_MESSAGE_PATTERN = """"Hello, administrator!

Size of the file '%s' is %d bytes.

"""""
```

```
class FSize_watcher_notifier (LveStatsPlugin):
    # Default period
    DEFAULT_PERIOD_STR = '12h'
    # this plugin should be third in chain
    order = 3
    # Timeout
    timeout = 20
    # Notifier Log filename
    log_file_name = '/var/log/FSize_watcher_notifier.log'
    # Email from address
    email_from = None
    # Email to address
    email_to = None
    # Email subject
    email_subject = None
    # Sets configuration to plugin
    def set_config(self, config):
        # Email settings
        self.email_from = config.get('notify_from_email', None)
        self.email_to = config.get('notify_to_email', None)
        self.email_subject = config.get('notify_from_subject', 'Message from FSize_watcher_notifier plugin')
        # Notify period
        s_period = config.get('notify_period', None)
        if s_period:
            self.period = dateutil.parse_period2(s_period)
        else:
            self.period = dateutil.parse_period2(FSize_watcher_notifier.DEFAULT_PERIOD_STR)
        f = open(self.log_file_name, 'a')
        f.write('%s - FSize_watcher_notifier plugin: configure\n' % time.asctime(time.localtime()))
        f.write('    - Period: %s\n' % self.period)
        f.write('    - From: %s\n' % self.email_from)
        f.write('    - To: %s\n' % self.email_to)
        f.write('    - Subject: '%s'\n' % self.email_subject)
        f.close()
    # work method
    def execute(self, lve_data):
```

```
if COLLECTOR_KEY_FSIZE not in lve_data or COLLECTOR_KEY_FILENAME not in lve_data:
    return
if not self.email_from or not self.email_to:
    f = open(self.log_file_name, 'a')
    f.write('%s - FSize_watcher_notifier plugin error: email_from or email_to not set\n')
    f.close()
    return
try:
    from email.mime.text import MIMEText
    # Send email
    msg = MIMEText(EMAIL_MESSAGE_PATTERN % (lve_data[COLLECTOR_KEY_FILENAME],
lve_data[COLLECTOR_KEY_FSIZE]))
    msg['Subject'] = self.email_subject
    msg['From'] = self.email_from
    msg['To'] = self.email_to
    s = smtplib.SMTP('localhost')
    s.sendmail(self.email_from, [self.email_to], msg.as_string())
    s.quit()
    f = open(self.log_file_name, 'a')
    f.write('%s - FSize_watcher_notifier plugin: email message was successfully sent\n' %
time.asctime(time.localtime()))
    f.close()
except Exception as e:
    f = open(self.log_file_name, 'a')
    f.write('%s - FSize_watcher_notifier plugin error:\n%s\n' % (time.asctime(time.localtime()), str(e)))
    f.close()
```

Configuration file /etc/sysconfig/lvestats.config/FSize_watcher_notifier.cfg:

```
# Config file for FSize_watcher_notifier.py plugin
# Please define email options here
NOTIFY_FROM_EMAIL=user@hostname
NOTIFY_FROM_SUBJECT=Message from FSize_watcher_notifier
NOTIFY_TO_EMAIL=admin@hostname
NOTIFY_PERIOD=12h
```

Plugin's index number equals 3 (order=3), that is why notifier starts after the rest. But since it uses only data formed by collector, then its order may equal any number bigger that collectors order (>0).

Notifier reads the necessary parameters from the configuration (email address, topic, period) and writes them into its own log as reference.

Plugin's execute method checks the availability of all the necessary data (email parameters, collectors data) and sends the message. All the notifications are written into the notifier's own log.

If any data is missing, the message is not sent.

Log example:

Thu Feb 5 11:51:34 2015 - FSize_watcher_notifier plugin: configure

- Period: 60.0
- From: `user@hostname`
- To: `admin@hostname`
- Subject: 'Message from FSize_watcher_notifier'

Thu Feb 5 11:51:35 2015 - FSize_watcher_notifier plugin: email message was successfully sent

Thu Feb 5 11:52:35 2015 - FSize_watcher_notifier plugin: email message was successfully sent

Thu Feb 5 11:53:35 2015 - FSize_watcher_notifier plugin: email message was successfully sent

Thu Feb 5 11:54:35 2015 - FSize_watcher_notifier plugin: email message was successfully sent

Thu Feb 5 11:57:00 2015 - FSize_watcher_notifier plugin: configure

- Period: 43200.0
- From: `user@hostname`
- To: `admin@hostname`
- Subject: 'Message from FSize_watcher_notifier'

Thu Feb 5 11:57:00 2015 - FSize_watcher_notifier plugin: email message was successfully sent

File info and format for `/var/lve/info` file

This file is used by control panels to display to user their 'current' usage. The file is updated every 5 seconds by lve-stats.

When writing to this file we make sure that: average CPU/IOPS/MEM is never more then LIMIT for that resource.

Example:

```
0,0,20,0,2500,0,262144,0,0,262144,0,0,100,0,0,0,0,1024,1024,0,0,0,0
600,1,20,2492,2500,70,262144,0,0,262144,33,0,100,1,0,0,0,1024,1024,0,5,0,0
200,0,20,0,2500,0,262144,0,0,262144,0,0,100,0,0,0,0,1024,1024,0,0,0,0
500,0,20,0,2500,0,262144,0,0,262144,0,0,100,0,0,0,0,1024,1024,0,0,0,0
```

First line of the file is 'default limits'.

Fields:

- # 0 - id
- # 1 - mep (average entry processes)
- # 2 - lep (limit ...)
- # 3 - cpu_usage (average speed)
- # 4 - lcpu (limit speed)

```
# 5 - mem_usage (average virtual memory)
# 6 - lmem (limit ...)
# 7 - mem_fault (number of virtual memory faults)
# 8 - mep_fault (number of entry processes faults)
LVE_VERSION >=6
# 9 - lmemphy (limit physical memory)
# 10 - memphy (average ...)
# 11 - memphy_fault (faults ...)
# 12 - lnproc (limit number of processes)
# 13 - nproc (average ...)
# 14 - nproc_fault (faults ...)
# 15 - lcpuw (CPU weight – deprecated not used)
# 16 - io_usage (average IO usage)
# 17 - io_limit (limit ...)
LVE_VERSION >=8
#18 - liops (limit IOPS)
#19 - iops (average IOPS)
```

Troubleshooting

Troubleshooting

lvestats service and utilities write fatal errors to system log.

There is /var/log/lve-stats.log file with additional information (warnings, tracebacks for errors)

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

** Note: CageFS is not supported for H-Sphere.

Installation

Minimum Requirements:

•	kernel: CL5 with lve0.8.54 or later, CL6 with lve1.2.17.1 or later, CL7.
---	--

•	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 -> 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.

When installing/upgrading, the `fs.proc_can_see_other_uid` parameter is set to 0 if it is available in `/etc/sysctl.conf`, but then it will be written to `/etc/sysctl.d/90-cloudlinux.conf`.

We recommend set `fs.proc_can_see_other_uid=0` to activate protection with `hidepid` option.

Uninstalling CageFS

To uninstall CageFS, start by disabling and 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
```

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
```

Command-line Tools

cagefsctl is used to manage CageFS. It allows initializing and updating CageFS, as well as enabling/disabling CageFS for individual users.

Use the following syntax to manage CageFS:

```
/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)
- f | --force : recreate CageFS (do not make backup, overwrite existing files)
- d | --dont-clean : do not delete any files from skeleton
: (use with --update option)
- k | --hardlink : use hardlinks if possible
- create-mp : Creates /etc/cagefs/cagefs.mp file
- mount-skel : mount CageFS skeleton directory
- unmount-skel : unmount CageFS skeleton directory
- remove-all : disable CageFS, remove templates and /var/cagefs directory
- sanity-check : perform basic self-diagnostics of common cagefs-related issues
: (mostly useful for support)
- 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
- update-list : update specified files only (paths are read from stdin)
- update-etc : update /etc directory of all or specified users
- set-update-period : set min period of update of CageFS in days (default = 1 day)

- `-force-update` : force update of CageFS (ignore period of update)
- `-force-update-etc` : force update of `/etc` directories for users in CageFS
- `-reconfigure-cagefs` : configure CageFS integration with other software (control panels, database servers, etc)

Use the following syntax to manage users:

`/usr/sbin/cagefsctl [OPTIONS] username [more usernames]`

Options:

- `-m | -remount` : remount specified user(s)
- `-M | -remount-all` : remount CageFS skeleton directory and all users
(use this each time you have changed `cagefs.mp` file)
- `-w | -unmount` : unmount specified user(s)
- `| -unmount-dir` : unmount specified dir for all users
- `-W | -unmount-all` : unmount CageFS skeleton directory and all users
- `-l | -list` : list users that entered in CageFS
 - `-list-logged-in` : list users that entered in CageFS via SSH
 - `-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-user-mode` : display current mode (“Enable All” or “Disable All”)
 - `-toggle-mode` : toggle mode saving current lists of users
(lists of enabled and disabled users remain unchanged)
 - `-list-enabled` : list enabled users
 - `-list-disabled` : list disabled users
 - `-user-status` : print status of specified user (enabled or disabled)
 - `-getprefix` : display prefix for user

PHP Selector related options:

- `-setup-cl-selector` : setup PHP Selector or register new alt-php versions
- `-remove-cl-selector` : unregister alt-php versions, switch users to default
: php version when needed
- `-rebuild-alt-php-ini` : rebuild `alt_php.ini` file for specified users
: (or all users if none specified)
- `-validate-alt-php-ini` : same as `-rebuild-alt-php-ini`
: but also validates `alt_php.ini` options
- `-cl-selector-reset-versions`: reset php version for specified users to default
: (or all users if none specified)

`-cl-selector-reset-modules` : reset php modules (extensions) for specific users
: to defaults (or all users if none specified)

`-create-virt-mp` : create virtual mount points for the user

`-create-virt-mp-all` : create virtual mount points for all users

`-remount-virtmp` : create virtual mount points and remount user

`-apply-global-php-ini` : use with 0, 1 or 2 arguments from the list: `error_log`,
: `date.timezone` without arguments applies
: all global php options including two above

Common options:

`-enable-cagefs` : enable CageFS

`-disable-cagefs` : disable CageFS

`-cagefs-status` : print CageFS status (enabled or disabled)

`-set-min-uid` : Set min UID

`-get-min-uid` : Display current MIN_UID setting

`-print-suids` : Print list of SUID and SGID programs in skeleton

`-do-not-ask` : assume "yes" in all queries
: (should be the first option in command)

`-clean-var-cagefs` : clean `/var/cagefs` directory (remove data of non-existent users)

`-set-tmpwatch` : set tmpwatch command and parameters
: (save to `/etc/cagefs/cagefs.ini` file)

`-tmpwatch` : execute tmpwatch (remove outdated files in tmp directories
: in CageFS for all users)

`-toggle-plugin` : disable/enable CageFS plugin

`-v` | `-verbose` : verbose output

`-wait-lock` : wait for end of execution of other cagefsctl processes
: (when needed) before execution of the command

`-h` | `-help` : this message

Running Command Inside CageFS

[lve-wrappers 0.6-1+]

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

If a 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_"
```

If you disable CageFS for a user, then `cagefs_enter` will be executed without `proxyexec`.

You can forcibly disable cagefs_enter start via proxyexec for all users (regardless if CageFS is enabled or disabled) by specifying the parameter cagefs_enter_proxied=0 in /etc/sysconfig/cloudlinux.

/bin/cagefs_enter.proxied can be executed instead of /bin/cagefs_enter to enter CageFS without proxyexec. Note that starting cagefs_enter via proxyexec is necessary to enable sending local notification messages to users with enabled CageFS. cagefs_enter is executed via proxyexec by default.

Sanity Check

[CageFS 6.0-34+]

CageFS –sanity-check utility allows to check CageFS configuration consistency, so that an administrator can save the time investigating issues with CageFS and ensure that custom configuration is correct.

To start run the command:

```
cagefsctl –sanity-check
```

At the moment 7 types of check are implemented:

1.Check cagefs mount points exists - reads cagefs.mp file and verifies if the directories specified in it really exist on the disk. To learn more visit https://docs.cloudlinux.com/index.html?mount_points.html and https://docs.cloudlinux.com/index.html?split_by_username.html

2.Check cagefs users.enabled is directory - ensures that if /etc/cagefs/users.enabled exists, then it is a directory, not a file (if it is recognized as a file, then it would cause a breakdown).

3.Check cagefs users.disabled is directory - ensures that if /etc/cagefs/users.disabled. exists, then it is a directory, not a file (if it is recognized as a file, then it would cause a breakdown).

4.Check cagefs disable.etcfs exists - checks if /etc/cagefs/etc.safe/disable.etcfs exists.

5.Check cagefs users can enter cagefs - chooses two users in the system with enabled CageFS (the first and the second ones in the unsorted list) and tries to log in to CageFS under their credentials and see what happens. It runs su -l "\$USER" -s /bin/bash -c "whoami" and compares the output with the \$USER and su command retcode estimation.

Note. If log in fails, it can be on different reasons, that can only be determined in manual mode. The checker only gives the output of the command.

6.Check cagefs proxy commands configs are parsable - tries to load /etc/cagefs/*.proxy.commands files and parse them to check the syntax. In case of any parsing error the test will fail. To learn more visit https://docs.cloudlinux.com/index.html?executing_by_proxy.html.

7.Check cagefs virt.mp files syntax - reads all /var/cagefs/*/*/*virt.mp files (if any) and checks their syntax validity. At the moment there are only two checks of the syntax: the file is not empty if it exists, and the file is not starting with the sub directory definitions (with @). To learn more visit https://docs.cloudlinux.com/index.html?per_user_virtual_mount_points.html

Possible results of the checks:

- OK - the check succeeded.
- FAILED - the check revealed a problem.
- SKIPPED - the check was skipped as it made no sense in such

environment (e.g. wrong control panel) or can not be performed for some reason (e.g no users with enabled CageFS found). The actual result does not mean that a problem exists and can be considered as positive.

- INTERNAL_TEST_ERROR - the check failed because of a problem inside

the checker itself. Must be reported to the developers.

In case if at least one of the checks resulted neither OK nor SKIPPED then the checker will end with ret code >0.

CageFS Quirks

Due to the nature of CageFS, some options will not work as before or will 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.
- CageFS installation changes jailshell to regular bash on cPanel - [read why](#).

Configuration

- *File System Templates*
- *Excluding Files*
- *Excluding Users*
- *Mount Points*
 - *Per user virtual mount points*
 - *Split by Username*
- *Base Home Directory*
- *PostgreSQL support*
- *PAM Configuration*
- *Executing By Proxy*
- *Custom /etc directory*
- *Moving cagefs-skeleton directory*
- *Moving /var/cagefs directory*
- *TMP directories*
- *Syslog*
- *Excluding mount points*

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/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=smmmsp
groups=smmmsp
```

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.

Excluding Files

To exclude files and directories from CageFS, edit file:

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

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

Please do not edit `/etc/cagefs/black.list` file because it is replaced during the update of CageFS package.

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.

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

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.

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
@subdir1,mask
@subdir2,mask
virt_dir2,mask
@subdir3,mask
@subdir4,mask
>virt_dir3,mask
@subdir5,mask
@subdir6,mask
# comments
```

- mask is always optional, if missing 0755 is used
- Create virtual directory subdir/virt_dir, mount it to:
 - o skeleton jaildir/virt_dir
 - o inside virtual directory, create directories subdir1, subdir2
 - o mount virt_dir1/subdir1 to subdir/virt_dir/subdir1

oif virtdir is started with >, create directory subdir/virtdir, but don't mount it into jaildir. This is needed for cases when virtdir 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
```

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

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

```
%/home/httpd/cgi-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.

Mounting user's home directory inside CageFS

CageFS 6.1-1 (and later) has improved mounting user's home directory that is applied for users with home directories like /home/user or /homeN/user (where N = 0,1,..9).

In such case, earlier versions of CageFS always mount user's home directory to /home/user and create symlink /homeN -> /home when needed, so user's home directory can be accessed both via /home/user and /homeN/user. This quirk leads to some rare incompatibilities between CageFS and other software (for example OpenCart), because real path of user's home directory in CageFS and in real file system can differ.

New CageFS mounts user's home directory in a way that its real path in CageFS is always the same as in real file system. Additionally, CageFS searches for symlinks like

/homeX -> /homeY and /homeX/user -> /homeY/user in real system and creates such symlinks in user's CageFS when found.

This new mounting mode is enabled by default. You can switch to old mounting mode by executing the following commands:

```
# touch /etc/cagefs/disable.home.dirs.search
```

```
# cagefsctl -force-update
```

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

Note. New mounting mode will be disabled automatically when “mounting base home directory” mode is enabled (“mount_basedir=1” setting in /etc/cagefs/cagefs.base.home.dirs file).

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 a 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).

PostgreSQL support

CloudLinux 7:

CageFS works with any PostgreSQL version installed from CloudLinux or CentOS repositories. PostgreSQL packages for CloudLinux 7 come from upstream (CentOS) unmodified. PostgreSQL's socket is located in `/var/run/postgresql` directory. This directory is mounted to CageFS by default (in cagefs-5.5-6.34 or later).

When PostgreSQL has been installed after CageFS install, please add line:

```
/var/run/postgresql
```

to `/etc/cagefs/cagefs.mnt` file and then execute:

```
cagefsctl -remount-all
```

The steps above are enough to configure CageFS to work with PostgreSQL.

CloudLinux 6:

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 <code>/etc/sysconfig/postgres</code> , and uncomment <code>SOCK_DIR</code> line.
----	--

3. Update CageFS configuration by running:

```
cagefsctl --reconfigure-cagefs
```

4. 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.

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.

Executing By Proxy

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

CloudLinux uses `proxyexec` technology to accomplish this goal. You can define any program to run outside CageFS, by specifying it in `/etc/cagefs/custom.proxy.commands` file. Do not edit existing `/etc/cagefs/proxy.commands` as it will be overwritten with next CageFS update.

Once program is defined, run this command to populate the skeleton:

```
$ cagefsctl -update
```

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

Users with duplicate UIDs

The syntax of `/etc/cagefs/*.proxy.commands` files is as follows:

ALIAS:wrapper_name=username:path_to_executable

Obligatory parameters are ALIAS and path_to_executable.

- ALIAS - any name which is unique within all

`/etc/cagefs/*.proxy.commands` files;

- wrapper_name - the name of wrapper file, which is used as a replacement for executable file path_to_executable inside CageFS. Wrapper files are located in `/usr/share/cagefs/safeprograms`. If wrapper name is not specified, then default wrapper `/usr/share/cagefs/safeprograms/cagefs.proxy.program` is used. Also, a reserved word “no proceed” can be used, it will intend that wrapper is not in use (installed before) - applied for the commands with several ALIAS, as in the example below.

- username - the name of a user on whose behalf path_to_executable will run in the real system. If username is not specified, then path_to_executable will run on behalf the same user that is inside CageFS.

- path_to_executable - the path to executable file which will run via `proxyexec`.

Example of a simple command executed via `proxyexec`:

```
SENDMAIL=/usr/sbin/sendmail
```

Example of crontab command execution with custom wrapper under root (privilege escalation). The command uses two ALIAS, that is why in the second line “no proceed” is specified instead of wrapper name.

```
CRONTAB_LIST:cagefs.proxy.crontab=root:/usr/bin/crontab
```

```
CRONTAB_SAVE:no proceed=root:/usr/bin/crontab
```

Sometimes hosters may have users with non unique UIDs. Thus, `proxyexec` may traverse users directory to find a specific one. That behavior turns into inappropriate if users directory is not cached locally (for example LDAP is in use).

To turn this feature off:

```
touch /etc/cagefs/proxy.disable.duid
```

Or to activate it back:

```
rm /etc/cagefs/proxy.disable.duid
```

Custom /etc files per customer

[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 directory.

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 changes 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
```

Moving cagefs-skeleton directory

Sometimes you might need to move `cagefs-skeleton` from `/usr/share` to another partition.

There are two ways:

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

```
$ mkdir /home/cagefs-skeleton
```

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

```
$ cagefsctl --init
```

2. 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 /home/cagefs-skeleton
```

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

```
cagefsctl --enable-cagefs
```

On cPanel servers, if you place skeleton into `/home` directory, then you should configure the following option:

In cPanel WHM choose Server Configuration and go to Basic cPanel/WHM Setup, then in Basic Config change Additional home directories default value to blank (not "home").

Note. If this option is not set, then cPanel will create new accounts in incorrect places.

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
```

TMP Directories

CageFS makes sure that each user has his own /tmp directory, and that directory is the 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 accessed during 30 days.

This can be changed by running:

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

Where 720 is the number of hours that the file had to be inaccessible to be removed.

By default this is done at 03:37 AM, but you can also force the clean up outdated files that match ‘chosen period’ of all user’s /tmp directories without waiting for a job to be launched by cronjob. Just run:

```
$ cagefsctl --tmpwatch
```

The following path will be cleaned as well:

/var/cache/php-eaccelerator (actual location \$USER_HOME/.cagefs/var/cache/php-eaccelerator)

You can configure tmpwatch to clean custom directories inside CageFS.

Create /etc/cagefs/cagefs.ini configuration file and specify tmpwatch_dirs directive as follows:

```
tmpwatch_dirs=/dir1,/dir2
```

After that directories /dir1 and /dir2 inside CageFS will be cleaned automatically.

Note that actual location of those directories in real file system is `$USER_HOME/.cagefs/dir1` and `$USER_HOME/.cagefs/dir2`.

Cleanup of PHP sessions

For cPanel servers, CageFS version 6.0-42 or higher performs cleaning of PHP sessions based on `session.gc_maxlifetime` and `session.save_path` directives specified in proper `php.ini` files.

`session.gc_maxlifetime` directive default value is 1440 seconds. Those session files will be deleted, that were created or had metadata (ctime) changes more time ago than it is specified in `session.gc_maxlifetime`.

For Alt-PHP versions `session.save_path` value is normally `/tmp`.

Note. For new installations of Alt-PHP packages, `session.save_path` will be changed from `/tmp` to `/opt/alt/phpNN/var/lib/php/session`, where NN corresponds to Alt-PHP version.

This applies to the following Alt-PHP versions (or later):

- alt-php44-4.4.9-71;
- alt-php51-5.1.6-81;
- alt-php52-5.2.17-107;
- alt-php53-5.3.29-59;
- alt-php54-5.4.45-42;
- alt-php55-5.5.38-24;
- alt-php56-5.6.31-7;
- alt-php70-7.0.23-5;
- alt-php71-7.1.9-5;
- alt-php72-7.2.0-0.rc.2.2.

When using EasyApache 3, `session.save_path` value is normally `/var/cpanel/php/sessions/ea3` or `/tmp`. Settings for EasyApache 3 are usually taken from the file `/usr/local/lib/php.ini`.

When using EasyApache 4, `session.save_path` value is normally `/var/cpanel/php/sessions/ea-phpXX`, where XX corresponds to PHP version.

Cleaning is started by cron `/etc/cron.d/cpanel_php_sessions_cron`, which starts the script `/usr/share/cagefs/clean_user_php_sessions` twice within one hour.

The settings for `ea-php` are located in `/opt/cpanel/ea-phpXX/root/etc/php.d/local.ini` or in `/opt/cpanel/ea-phpXX/root/etc/php.ini`, where XX corresponds to the PHP version.

The settings for `alt-php` are located in `/opt/alt/phpXX/etc/php.ini` files, where XX corresponds to PHP version.

The cleaning script cleans php sessions for all PHP versions (`ea-php` and `alt-php`) regardless of whether a version is used or selected via MultiPHP Manager or PHP Selector. When different `session.gc_maxlifetime` values are specified for the same `session.save_path` (for different php versions), the cleaning script will use the least value for cleaning `session.save_path`. So, it is recommended to specify different `session.save_path` for each PHP version.

Users can define custom value of `session.gc_maxlifetime` via PHP Selector in order to configure PHP's garbage collector, but that will not affect the script for cleaning PHP sessions. The script cleans PHP sessions based on global values of `session.gc_maxlifetime` and `session.save_path` directives taken from files mentioned above. Settings in custom users' `php.ini` files are ignored.

Cleanup of PHP session files in Plesk

For Plesk servers, CageFS version 6.0-52 or higher is provided with a special cron job for removing obsolete PHP session files. Cleanup script runs once an hour (similar to how it is done in Plesk).

Each time the script runs, it performs the cleanup of the paths:

1. set by session.save_path directive in /opt/alt/phpXX/etc/php.ini files. If session.save_path is missing, then /tmp is used. Session files lifetime is set by session.gc_maxlifetime directive. If it is not found, then 1440 seconds value is used (24 minutes, as in Plesk). Lifetime set in the file is only taken into consideration if it is longer than 1440 seconds, otherwise 1440 seconds is used. All the installed Alt-PHP versions are processed.
2. /var/lib/php/session. Files lifetime is only defined by Plesk script /usr/lib64/plesk-9.0/maxlifetime. If the script is missing or returns errors, then this directory is not processed.

The following features are applied during the cleanup:

- all the users with UID higher than specified in /etc/login.defs are processed. Each user is processed independently from one another.
- only directories inside CageFS are being cleaned. The paths of the same name in the physical file system are not processed.
- in all the detected directories, all the files with the names that correspond to sess_* search mask are removed, the rest of the files are ignored.
- the files older than specified lifetime are removed.
- all non-fatal errors (lack of rights, missing directory) are ignored and do not affect the further work of the script.

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.

Excluding mount points

How to exclude mounts from namespaces for all LVEs

By default, all mounts from the real file system is inherited by namespaces of all LVEs. So, destroying all LVEs may be required in order to unmount some mount in real file system completely. Otherwise, mount point remains busy after unmounting it in the real file system because this mount exists in namespaces of LVEs.

`lvectl start` command saves all mounts from real file system as “default namespace” for later use in all LVEs. `lve_namespaces` service executes `lvectl start` command during startup.

In `lve-utils-2.0-26` (and later) there is an ability to exclude specific mounts from namespaces for all LVEs.

In order to do so, please create a file `/etc/container/exclude_mounts.conf` with list of mounts to exclude (one mount per line) as regular expressions, and then execute `lvectl start`:

```
# cat /etc/container/exclude_mounts.conf
^/dir1/
^/dir2$
# lvectl start
```

After that, all new created LVEs will be without `/dir2` mount and without mounts that start with `/dir1/` (like `/dir1/x`, `/dir1/x/y`, etc). To apply changes to existing LVEs you should recreate LVEs:

```
# lvectl destroy all
# lvectl apply all
```

Note. You should recreate all LVEs only once after creating `/etc/container/exclude_mounts.conf` file. After that the configuration changes will be applied to all new LVEs automatically.

Control Panel Integration

CageFS comes with a plugin for various control panels.

The plugin allows to:

•	Initialize CageFS;
•	Select <i>mode of operation</i> ;
•	See and modify the list of enabled/disabled users;
•	Update CageFS skeleton.

cPanel

CageFS plugin for cPanel is located in Plugins section of WHM.

It allows to initialize CageFS, select users CageFS will be enabled for, as well as update CageFS skeleton.

To enable CageFS for a proper user (users), in CageFS User Manager choose a user from the list on the right (Disabled users) and click Toggle. The user will move to the list on the left (Enabled users).

To disable a user (users), choose a user from the list on the left (Enabled users) and click Disable CageFS. The user will move to the list on the right (Disabled users).

To update CageFS Skeleton click Update CageFS Skeleton.

Plesk

CageFS comes with a plugin for Plesk 10.x. It allows initializing and updating CageFS template, as well as managing users and mode of operation for CageFS.

In modules section choose CageFS:

To enable CageFS for a proper user (users), in CageFS User Manager choose a user from the list on the right (Disabled users) and click Toggle. The user will move to the list on the left (Enabled users).

To disable a user (users), choose a user from the list on the left (Enabled users) and click Disable CageFS. The user will move to the list on the right (Disabled users).

To update CageFS Skeleton click Update CageFS Skeleton.

####

ISPManager

CageFS comes with plugin for ISP Manager to enable/disable CageFS on per user base. In edit user section chose Permission tab. Mark CageFS User Mode checkbox and click OK to apply.

`lispmanager_cagefs_user_zoom98|`

Or you can manage global CageFS settings via CageFS menu

`_img3|`

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.

Installation

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.

MySQL Governor is compatible only with MySQL 5.x, MariaDB & Percona

Server 5.6.

To install MySQL Governor on your server install governor-mysql package at first:

```
$ yum remove db-governor db-governor-mysql # you can ignore errors if you don't have those packages installed
$ yum install governor-mysql
```

Then configure MySQL Governor properly.

The installation is currently supported only on cPanel, Plesk, DirectAdmin, ISPmanager, InterWorx, as well as on servers without control panel.

If you are installing CloudLinux on a server running MySQL already, set your current MySQL version before calling installation script:

```
$ /usr/share/lve/dbgovernor/mysqlgovernor.py --mysql-version=mysqlXX
$ /usr/share/lve/dbgovernor/mysqlgovernor.py --install
```

Please make sure to specify your current MySQL version instead of XX as follows:

- 55 — MySQL v5.5
- 56 — MySQL v5.6
- 57 — MySQL v5.7

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

```
$ /usr/share/lve/dbgovernor/mysqlgovernor.py --mysql-version=mariadbXX
$ /usr/share/lve/dbgovernor/mysqlgovernor.py --install
```

Please make sure to specify your current MariaDB version instead of XX as follows:

- 55 — MariaDB v5.5
- 100 — MariaDB v10.0
- 101 — MariaDB v10.1
- 102 — MariaDB v10.2
- 103 — MariaDB v10.3

Installation for Percona Server 5.6 [requires MySQL Governor 1.1-22+ or 1.2-21+]:

```
$ /usr/share/lve/dbgovernor/mysqlgovernor.py --mysql-version=percona56
$ /usr/share/lve/dbgovernor/mysqlgovernor.py --install
```

Please note that MySQL/MariaDB/Percona will be updated from CloudLinux repositories.

If you are installing MySQL Governor on a server without MySQL at all, you have an opportunity to choose desired MySQL version to be installed with MySQL Governor installation script. Use `--mysql-version` flag before calling the installation script:

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

MYSQL_VERSION could be chosen from the list of versions currently supported by MySQL Governor:

mysql51	MySQL v5.1
mysql55	MySQL v5.5
mysql56	MySQL v5.6
mysql57	MySQL v5.7
mariadb55	MariaDB v5.5
mariadb100	MariaDB v10.0
mariadb101	MariaDB v10.1
mariadb102	MariaDB v 10.2
mariadb103	MariaDB v 10.3
percona56	Percona Server v 5.6

Generally, stable and beta channels contain different version of MySQL packages - beta contains newer version than stable or the same one. If you would like to install beta packages, use `--install-beta` flag instead of `--install` when calling installation script:

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

Starting with MySQL Governor version 1.2 when installing MySQL/MariaDB MySQL Governor asks for a confirmation of a database version to be installed. To avoid such behavior for the automatic installations, please use `--yes` flag.

For example:

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

Please note that restore of previous packages in case of failed installation would also be confirmed with `--yes` flag.

WARNING! Use `--yes` flag on your own risk, because it confirms installation in any case - even in case if there are troubles during installation (for example, network problems causing incomplete download of packages), everything would be confirmed.

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.

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 in /etc/container/dbuser-map

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 MySQL 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_connections="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 parameter in the <limit name="slow" current="30"/> will no be applied without enabling slow_queries -->
<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 are saved in /var/lve/dbgovernor-store and being kept here during 10 days -->
<logqueries use="onoff"></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(but unit can be specified)
-->

<!-- enabled only when slow_queries run="on" -->
<!-- s – seconds, m – minutes, h – hours, d – days -->

<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>
```

Starting And Stopping

To start:

```
$ service db_governor start
```

To stop:

```
$ service db_governor stop
```

Mapping a User to Database

[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 in: `/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
```

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 in `/var/log/dbgovernor-restrict.log`

Restrictions:

```
_timestamp_ _username_ LIMIT_ENFORCED _limit_setting_ __current_value_ _restrict_level__ SERVER_LOAD
TRACKED_VALUES_DUMP
...
```

- `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

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.

Note. For experienced users only. Changing MySQL version is a quite complicated procedure, it causes system table structural changes which can lead to unexpected results. Think twice before proceeding.

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/mysqlgovernor.py --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
mysql57	MySQL v5.7
mariadb55	MariaDB v5.5
mariadb100	MariaDB v10.0
mariadb101	MariaDB v10.1
mariadb102	MariaDB v 10.2
mariadb103	MariaDB v 10.3
percona56	Percona v 5.6

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

Note. Starting from cPanel & WHM version 70 cPanel supports MySQL 5.7:

<https://blog.cpanel.com/being-a-good-open-source-community-member-why-we-hesitated-on-mysql-5-7/>

Note. cPanel does not officially support MariaDB 10.3, that is why we don't recommend to use it on cPanel servers.

Use on your own risk for DirectAdmin and Plesk servers, because downgrade can corrupt your databases.

MySQL Governor starting from version 1.2-36 (for now, July 4th, 2018 in Beta) supports MariaDB 10.3 installation.

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...*

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

l/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)

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

username 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) - this option is available only for period mode:

`<restrict_mode use="period"/>` (see http://docs.cloudlinux.com/index.html?mysql_governor_configuration.html)

The default mode is the “limit” - when a user hits limits, the account will be marked as restricted and if the user does not hit the limit again during “unlimit=1m” account will be unrestricted. This mode doesn’t have any additional levels/penalties.

`<restrict_mode use="limit" unlimit="1m"/>`

Changing the “unlimit” can be done only via the configuration file (see http://docs.cloudlinux.com/index.html?mysql_governor_configuration.html).

`-slow=N` limit time (in seconds) for long running SELECT queries

Options for parameter list:

`-kb` show limits in Kbytes no pretty print

`-bb` show limits in bytes no pretty print

`-mb` show limits in Mbytes no pretty prin

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
```

Show limits as bytes:

```
$dbctl list -bb
```

lveinfo -dbgov

lveinfo tool is a 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 today

-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_rest : 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_rest : 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
 ICPU : CPU limit
 IREAD : READ limit
 IWRITE : WRITE limit
 RESTRICT : C-cpu restrict, R- read restrict, W- write restrict

Example:

```

root@cpanel1 [~/tttt]# lveinfo -dbgov -user=dbgov -period=1d -limit=10
TS      USER CPU  READ  WRITE CON  ICPU IREAD IWRITE RESTRICT
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

dbgovchart

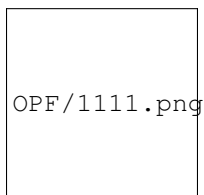
dbgovchart is analog of lvechart tool to create charts representing customer's to MySQL usage

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:



1111\2

Backing Up MySQL

On cPanel server disable MySQL service monitoring before doing the job:

```
$ whmapil configureservice service=mysql enabled=1 monitored=0
```

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
```

On cPanel server enable monitoring back:

```
$ whmap1 configureservice service=mysql enabled=1 monitored=1
```

Note this operation may take some time.

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
```

Troubleshooting

MariaDB 5.5 and MariaDB 10.0: How to set LimitNOFILE correctly for systemd.

MariaDB 5.5 and MariaDB 10.0 have only /etc/init.d/mysql file for managing the service, but the file has LSB functions, so it is supported by systemd.

For adding extra limits, do the following:

1. Run:

```
mkdir /etc/systemd/system/mariadb.service.d/
```

2. Run:

```
touch /etc/systemd/system/mariadb.service.d/limits.conf
```

3. Add the following content to the the file /etc/systemd/system/mariadb.service.d/limits.conf:

```
[Service]
```

```
LimitNOFILE=99999
```

PHP Selector

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

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

It is not compatible with mod_php/DSO, including mod_ruid2 and MPM ITK.

Note: PHP Selector is not supported for H-Sphere.

Installation

The installation of PHP Selector presumes that you already have *CageFS & LVE Manager* installed.

Use [compatibility matrix](#) to check if your Web Server/PHP mode is supporting PHP Selector. If not, you need a change to one of the supported models.

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.

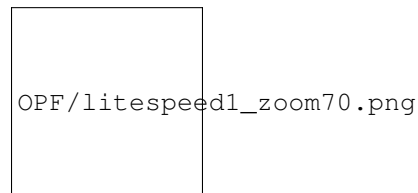
If you need custom PHP settings per user, please change them via “Edit PHP settings” feature of PHP Selector.

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.Go to External App tab, the new lsphp_selector is here.

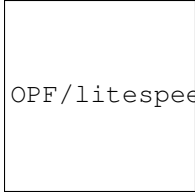
[Note that you can select any other application or create a custom one.]



- 4.The Command line should be /var/www/cgi-bin/cgi_wrapper/cloudlinux_wrapper on Plesk. For other control panels, Command line should be /usr/local/bin/lspsh.

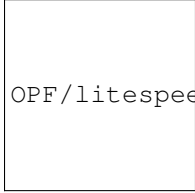
Run On Start Up line must contain Yes or No.

For Plesk:



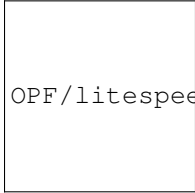
OPF/litespeed3_zoom70.png

For other control panels:

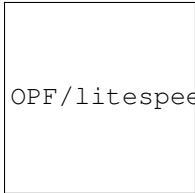


OPF/litespeed2_zoom70.png

5. Go to Script Handler tab. For required suffixes change the Handler Name to lsphp_selector.



OPF/litespeed4_zoom70.png



OPF/litespeed5_zoom70.png

* In order to use PHP Selector and custom php.ini, lsphp5 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.

*** If you have LiteSpeed installed not in standard location path, please create a symlink: 'ln -s /path/to/custom/lsws /usr/local/lsws' then run 'cagefsctl --setup-cl-selector'.

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.

Configuration

- *Setting default version and modules*
- *Individual PHP.ini files*
- *Substitute global php.ini for individual customer*
- *Managing interpreter version*
- *Including PHP Selector only with some packages (cPanel)*
- *PHP Extensions*
- *FFmpeg*
- *Native PHP Configuration*

Setting Default Version and Modules

Administrator can set default interpreter version and extensions for 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
```

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.

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.Make sure that `php.ini` load path is set to `/etc/php.ini`.
- 2.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.
- 3.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.
- 4.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.

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 7.2.5. 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/selectorctl --list --interpreter=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/selectorctl --user USERNAME --interpreter=php --user-current` will retrieve PHP version set for a particular user. The script gets the path from `/var/cagefs/LAST_TWO_DIGITS_OF_UID/USERNAME/etc/cl.selector/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/selectorctl --user USERNAME --interpreter=php --set-user-current=7.2` sets the current PHP version for particular user by creating symlink in `/var/cagefs/LAST_TWO_DIGITS_OF_UID/USERNAME/etc/cl.selector` directory. All old symlinks are removed, and new symlinks are set.

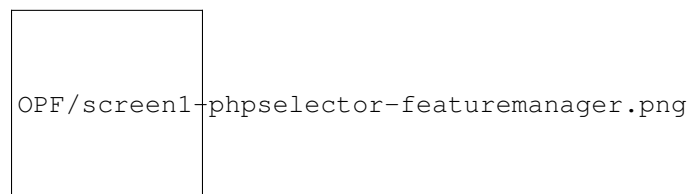
Including PHP Selector only with some packages (cPanel)

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’



PHP Extensions

Configuring Alt-PHP modules loading

CloudLinux PHP Selector and Alt-PHP can be used in conjunction with Plesk PHP Selector and cPanel MultiPHP. To be compatible, CloudLinux PHP Selector works as follows: modules that are selected in CloudLinux PHP Selector are loaded for Alt-PHP version selected in CloudLinux PHP Selector only. For the rest Alt-PHP versions default module set is loaded (/opt/alt/phpXX/etc/php.d/default.ini). Described above is default behavior.

Note. If system default PHP version selected in cPanel MultiPHP Manager is not ea-php, then default module set is loaded for all Alt-PHP versions by default (/opt/alt/phpXX/etc/php.d/default.ini).

When “php.d.location = selector” option is in effect, modules selected via PHP Selector will be loaded for all alt-php versions.

This behavior is implemented in CageFS-6.1-10 and later.

In LVE Manager 1.0-9.40+ this behavior can be modified so that modules selected in CloudLinux PHP Selector would be loaded for all Alt-PHP versions (with CageFS enabled), which can be quite useful if you use ‘per directory’ or ‘per domain’ Alt-PHP configuration and want to select modules using CloudLinux PHP Selector.

To modify it, create a file /etc/cl.selector/symlinks.rules (read-only for regular users) with the following content: php.d.location = selector

And run the command to apply changes:

```
/usr/bin/selectorctl --apply-symlinks-rules
```

To revert to the default behavior:

- Delete /etc/cl.selector/symlinks.rules file.
- Alternatively remove php.d.location option from the file.
- Alternatively set default value for php.d.location option.

And run the command to apply changes:

```
/usr/bin/selectorctl --apply-symlinks-rules
```

FFmpeg for Alt-PHP

Due to possible patent issues CloudLinux does not provide FFmpeg libraries (<https://ffmpeg.org/legal.html>). We highly recommend researching if you can legally install FFmpeg extension on your server. This might differ based on where you and your servers are located. More information can be found on the link: <https://ffmpeg.org/legal.html>

For your convenience we provide FFMPEG PHP binding. For them to work, you need to install FFmpeg package from the “Nux Dextop” repository following the [instructions](#).

Once FFmpeg is installed you can install PHP bindings, by running:

```
yum install alt-php*ffmpeg
```

Enable PHP-FFmpeg extension via PHP Selector:

```
selectorctl --enable-extensions=ffmpeg --user USERNAME --version X.Y
```

Native PHP Configuration

PHP Selector requires access to the native PHP version for proper work. It is specified in the file /etc/cl.selector/native.conf of the following content (example):

```
php=/usr/bin/php-cgi
```

```
php-cli=/usr/bin/php
```

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

The file is created when installing CageFS on the servers with cPanel, Plesk, DA, Interworx and ISP Manager, if it is missing. On all other servers the file is not being created at all.

That is why, if the file is not created automatically, then it must be created manually and correct paths must be written to its directives.

Access permission 644 must be set:

```
chmod 0644 /etc/cl.selector/native.conf
```

Command-line Tools

/usr/bin/cl-selector	Tool is used to select version of PHP interpreter inside CageFS. Note. The command is obsolete, please use <i>selectorctl</i> _ instead.
/usr/bin/alt-php-mysql-reconfigure.p y	Reconfigures alt-php extensions to use correct MySQL library, based on the one installed in the system.

selectorctl

selectorctl is a new tool that replaces cl-selector (which is deprecated and should not be used anymore) and 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.

-current (-C):	<p>prints currently globally selected default version (it is stored in /etc/cl.selector/defaults.cfg file):</p> <pre>\$ selectorctl -current native native /usr/bin/php</pre> <p>If used with -show-native-version, native interpreter version is displayed as well:</p> <pre>-current -show-native-version native(5.3) native(5.3.19) /usr/bin/php</pre>
----------------	---

| |

<code>-list-extensions (-G):</code>	<p>lists extensions for an alternative for a particular version. Requires <code>-version</code>. Example:</p> <pre>\$ selectorctl -list-extensions -version=5.3 ~ xml - xmlreader - xmlrpc - xmlwriter - xrange + xsl</pre> <p>Plus sign (+) stands for 'enabled', minus (-) for 'disabled', tilde (~) means compiled into interpreter. Enabled and disabled state relates to presence in <code>/etc/cl.selector/defaults.cfg</code> file.</p>

End User Options

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

<p><code>--user-summary (-s):</code></p>	<p>prints user alternatives state summary. Example:</p> <pre>\$ selectorctl --user-summary --user=user1 5.2 e - - 5.3 e - - 5.4 e - - 5.5 e - - native e d s</pre> <p>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 <code>--show-native-version</code>, version for native interpreter is shown in parenthesis:</p> <pre>\$ selectorctl --user-summary --user=user1 --show-native-version 5.2 e - - 5.3 e - - 5.4 e - - 5.5 e - - native(5.3) e d s</pre> <p><code>--user</code> option is required.</p>
<p><code>--current (-c):</code></p>	<p>prints currently globally selected default version (in <code>/etc/cl.selector/defaults.cfg</code> file):</p> <pre>\$ selectorctl --current 5.3 5.3.28 /opt/alt/php53/usr/bin/php-cgi</pre> <p>If used with <code>--show-native-version</code> to display native version:</p> <pre>\$ selectorctl --user-current --user=user1 5.3 5.3.28 /opt/alt/php53/usr/bin/php-cgiy</pre> <p><code>--user</code> option is required.</p>
<p><code>--set-user-current (-b):</code></p>	<p>sets specified version as the one to use for this end user:</p> <pre>\$ selectorctl --set-user-current=5.4 --user=user1</pre> <p>changes user symlinks for the PHP interpreter to point to alternative 5.4.</p> <p><code>--user</code> option is required.</p>

| `--user` option is required. |

<p><code>-list-user-extensions (-g):</code></p>	<p>lists enabled user extensions for an alternative. Requires <code>-version</code> and <code>-user</code> options.</p> <pre>\$ selectorctl -list-user-extensions -version=5.3 -user=user1 xml xmlreader xmlrpc</pre> <p>if <code>-all</code> option present, command will list all alternatives extensions marked enabled or disabled for given user. For example:</p> <pre>\$ selectorctl -list-user-extensions -version=5.3 -user=user1 -all - xmlreader - xmlrpc - xmlwriter - xrange + xsl</pre> <p>Plus sign (+) stands for 'enabled', minus (-) stands for 'disabled'. Enabled and disabled state relates to presence or absence of corresponding extensions in user <code>alt_php.ini</code> file.</p>
<p><code>-add-options (-k):</code></p>	<p>adds options (as in <code>php.ini</code>) to user <code>alt_php.ini</code> file. For example:</p> <pre>\$ selectorctl -add-options=log_errors:on,display_errors:on -version=5.2 -user=user1</pre> <p>adds <code>log_error</code> and <code>display_errors</code> options with values 'on' to user <code>alt_php.ini</code> file overwriting default values for a user. Requires <code>-version</code> and <code>-user</code> options.</p>

|`-version` and `-user` options. |

<code>-print-options (-P):</code>	<p>print options from <code>/etc/cl.selector/php.conf</code> file with default values or ones overwritten in user's <code>alt_php.ini</code> file.</p> <pre>\$ 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 ...</pre> <p>Requires <code>--user</code> option. <code>--version</code> option is optional. When <code>--version</code> is omitted, options for current selected version will be printed. By default outputs as plain test. If <code>--json</code>, <code>--csv</code>, <code>--perl</code> is specified, outputs data in corresponding format. For example, with <code>--perl</code> option, the output is perl hash structure that can be evaluated.</p>
-----------------------------------	--

| |

Additional Options:

<code>-base64 (-Q)</code>	<p>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 <code>disable_functions=exec,popen,system</code> and <code>display_errors=on</code> to user options, do the following:</p> <pre>\$ selectorctl --add-options='echo disable_functions:exec,popen,system base64 -w 0','echo display_errors:onlybase64 -w 0' --version=5.2 --user=user1 --base64</pre> <p>Option <code>-w 0</code> of <code>base64</code> executable stands for 'disable wrapping of lines'. Without it <code>base64</code> output will break the command.</p>
<code>-quiet</code>	makes <code>selectorctl</code> continue when it encounter option not found in <code>php.conf</code> . Without it <code>selectorctl</code> exits with error.

Integrating With Control Panels

This is the list of commands that we use to integrate PHP Selector with control panels. If you need to integrate PHP Selector with a custom control panel, you might find all the commands here:

PHP summary:

Command:

```
/usr/bin/selectorctl --summary
```

Result:

4.4 e -

5.1 e -

5.2 e -

5.3 e -

5.4 e -

5.5 e -

5.6 e -

7.0 e -

7.1 e -

native e d

When native PHP version needs to be displayed:

Command:

```
/usr/bin/selectorctl --summary --show-native-version
```

Result:

4.4 e -

5.1 e -

5.2 e -

5.3 e -

5.4 e -

5.5 e -

5.6 e -

7.0 e -

7.1 e -

native(5.6) e d

The first column: PHP version

The second column: enabled or not (e - enabled)

The third column: if selected as default (d - default)

Set default version:

```
/usr/bin/selectorctl --set-current=_VERSION_
```

Disable version:

```
/usr/bin/selectorctl --disable-alternative=_VERSION_
```

Enable version:

```
/usr/bin/selectorctl --enable-alternative=_VERSION_
```

List Extensions for a version:

```
/usr/bin/selectorctl --list-extensions --version=5.6
```

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

Select Default Extensions (enable comma-separated list of extensions globally for a version):

```
/usr/bin/selectorctl --version=5.6 --enable-extensions=pdo,json,mysql
```

Deselect Default Extensions (disable comma-separated list of extensions globally for a version):

```
/usr/bin/selectorctl --version=5.6 --disable-extensions=pdo,json,mysql
```

Replace extensions with comma-separated list of extensions for a version globally:

```
/usr/bin/selectorctl --version=5.6 --replace-extensions=pdo,json,mysql
```

Select PHP version for a user:

```
/usr/bin/selectorctl --set-user-current=_VERSION_ --user=_USER_
```

List Enabled extensions for a user:

```
/usr/bin/selectorctl --list-user-extensions --user=_USER_ --version=_VERSION_
```

Enable comma-separated list of extensions for a user:

```
/usr/bin/selectorctl --enable-user-extensions=pdo,json,mysql --user=_USER_ --version=_VERSION_
```

Reset user's extensions to defaults:

```
/usr/bin/selectorctl --reset-user-extensions --user=_USER_ --version=_VERSION_
```

Replace user extensions with comma-separated list of extensions:

```
/usr/bin/selectorctl --replace-user-extensions=EXT_LIST --user=_USER_ --version=_VERSION_
```

EXT_LIST is comma separated list of PHP extensions (for example: pdo,json,mysql)

List available options for php.ini editing:

```
/usr/bin/selectorctl --print-options --user=_USER_ --version=_VERSION_ [-json]
```

List available options for php.ini editing (print safe strings):

```
/usr/bin/selectorctl --print-options-safe --user=_USER_ --version=_VERSION_ [-json]
```


Set php.ini options for end user:

```
/usr/bin/selectorctl --user=_USER_ --version=_VERSION_ --replace-options=_OPTIONS_ --base64 [--json]
```

Here is an example of 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
```

Removing PHP Selector

Once alternative versions of PHP are removed, PHP Selector will be disabled.

To do that, run:

```
$ yum groupremove alt-php
```

Using PHP Selector

Once PHP Selector is installed you will see “Selector” tab in LVE Manager:

[|php_selector.png|](#)

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:

[|php_selector_user.png|](#)

as well as extensions that they want to use:

[|phpselector_customer|](#)

and php.ini settings

[|phpselector_options|](#)

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 to file system) mainly

for security reasons.

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 directory will bypass security (UID/GID) checks. The 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 *selectorctl* command.

Users can use web interface to modify php.ini settings:

lphpselector_optionsl

End user files and directories

The 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

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 by sending us a support [ticket](#). 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 come with particular Alt-PHP version.

The full process for PHP 5.X looks as follows:

1. Download and unpack extension, cd into it's directory.

2. Execute our version of phize if necessary:

```
/opt/alt/php5X/usr/bin/phize
```

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 the modules directory (on 32-bit 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
```

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 of PHP versions bundled by CloudLinux.

- Put all the ini files for all the modules into /opt/alt/php51/etc/php.d.all

- Create a symbolic link /opt/alt/php51/etc/php.d -> /etc/cl.php.d/alt-php51

Place all such files into /opt/alt/php51/usr/lib/php/modules

Add an absolute path to PHP binaries into /etc/cl.selector/selector.conf using the 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
```

The new PHP version must be available now for selection in PHP Selector.

Detect User's PHP Version

[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 the command line. You can get this information by running:

```
$ selectorctl --interpreter=php --user-summary --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.

The second column will contain either e or -. If e is present, it means that given version is enabled, and can be selected by the end user.

The 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.

The fourth column can have values s or -. If s is present, that is the selected version, currently being used by the end user. Only one PHP version will have s indicator.

In case a user is not inside CageFS, and as such doesn’t use PHP Selector, you will see the following error message:

```
ERROR:User USERNAME not in CageFS
```

PHP Selector without CageFS

[LVE Manager 2.0-11.1 or higher]

PHP Selector can now be used with CageFS turned off (in case when there is only one user account on the server).

To install run:

```
yum groupinstall alt-php
```

```
yum install cagefs lvmanager
```

(no need to initialize or turn on CageFS)

```
selectorctl --setup-without-cagefs USER
```

(USER - the name of a user who is using selector. If not specified, the first available cPanel account username will be used).

When executing --setup-without-cagefs, the following actions are performed:

- Creating symlinks to the user modules and options for each Alt-PHP version:

```
/opt/alt/php55/link/conf/alt_php.ini -> /home/USER/.cl.selector/alt_php55.ini
```

- In user home directory creating:

```
.cl.selector/
```

“Backup” settings files (selected version, modules, options):

```
.cl.selector/defaults.cfg
```

```
.cl.selector/alt_php44.cfg
```

Symlinks to the selected version:

```
.cl.selector/lspHP -> /opt/alt/php44/usr/bin/lspHP
```

```
.cl.selector/php.ini -> /opt/alt/php44/etc/php.ini
```

```
.cl.selector/php-cli -> /opt/alt/php44/usr/bin/php
```

```
.cl.selector/php -> /opt/alt/php44/usr/bin/php-cgi
```

Additional symlinks for environment variable \$PATH (search path) in the file ~/.bashrc:

```
.cl.selector/selector.path/
```

```
.cl.selector/selector.path/php-cgi -> ../php
```

```
.cl.selector/selector.path/php -> ../php-cli
```

Generated ini files with selected modules and options for each version:

```
.cl.selector/alt_php44.ini
```

```
.cl.selector/alt_php51.ini
```

```
.cl.selector/alt_php52.ini
```

```
.cl.selector/alt_php53.ini
```

```
.cl.selector/alt_php54.ini
```

```
.cl.selector/alt_php55.ini
```

```
.cl.selector/alt_php56.ini
```

```
.cl.selector/alt_php70.ini
```

```
.cl.selector/alt_php71.ini
```

Symlinks above are being created according to the settings in ~/.cl.selector/defaults.cfg and ~/.cl.selector/alt_php44.cfg files (44 - corresponding PHP version), which are storing PHP Selector settings for the user. These files are usually taken from user home directory backup or when migrating account from another server. Thus, when migrating account from server to server, PHP Selector settings are saved.

If no PHP Selector settings backup files are found when running `selectorctl --setup-without-cagefs`, then default settings from /etc/cl.selector/defaults.cfg global file are applied (as in selector normal mode). If the file is absent, then native PHP version will be selected for the user.

- The following line: `PATH=$HOME/.cl.selector/selector.path:$HOME/.cl.selector:$PATH`

is being added to the user file ~/.bashrc

Apache PHP handlers settings are not changed.

- Also `selectorctl --setup-without-cagefs` command does the following:

- oTurns off link traversal protection (linksafe);

- oTurns off cagefs service.

To get back to the selector normal mode (“with CageFS”) run:

```
selectorctl --revert-to-cagefs
```

(CageFS should be initialized by using “`cagefsctl --init`” command before running the command above)

This command removes symlinks:

```
/opt/alt/php55/link/conf/alt_php.ini -> /home/USER/.cl.selector/alt_php55.ini, turns on link traversal protection (linksafe) and cagefs service.
```

Configuring “Global” php.ini Options for all Alt-PHP Versions

[CageFS 6.0-33 or higher, LVE Manager 2.0-11.2 or higher]

There is `/etc/cl.selector/global_php.ini` file, where you can specify values of PHP options that should be applied for all Alt-PHP versions that are installed on a server. These settings will also be automatically applied to the new Alt-PHP versions that will be installed later.

Example:

```
# cat /etc/cl.selector/global_php.ini
```

```
[Global PHP Settings]
```

```
date.timezone = Europe/Warsaw
```

```
error_log = error_log
```

```
memory_limit = 192M
```

Sections are ignored. Only name of an option and a value have meaning.

When an option is absent in `/etc/cl.selector/global_php.ini` file, than it is not changed (applied) to `php.ini` for Alt-PHP versions.

`date.timezone` and `error_log` options are handled differently than the others. When these options are not in `/etc/cl.selector/global_php.ini` file, than values for the options will be taken from “native” `php.ini` file. And when the option is in `php.ini` for some Alt-PHP version already (and its value is not empty), than value from `/etc/cl.selector/global_php.ini` will be NOT applied.

To confirm changes (not affecting “`date.timezone`” and “`error_log`” options) please run:

```
/usr/sbin/cagefsctl --setup-cl-selector
```

To confirm changes (including “`date.timezone`” and “`error_log`” options) please run:

```
/usr/bin/selectorctl --apply-global-php-ini
```

or

```
/usr/sbin/cagefsctl --apply-global-php-ini
```

(two commands above work the same way).

If you don’t want to change `error_log`, but want to change `date.timezone`, you can execute:

```
selectorctl --apply-global-php-ini date.timezone
```

Similarly, command “`selectorctl --apply-global-php-ini error_log`” applies `error_log` and all other options specified in `/etc/cl.selector/global_php.ini` file, except `date.timezone`.

So, you can specify 0, 1 or 2 parameters from the list: `error_log`, `date.timezone`.

Using `--apply-global-php-ini` without arguments applies all global PHP options including two above.

Example:

```
selectorctl --apply-global-php-ini error_log
```

```
selectorctl --apply-global-php-ini date.timezone
```

```
selectorctl --apply-global-php-ini date.timezone error_log
```

The latter command has the same effect as `/usr/bin/selectorctl --apply-global-php-ini`

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*
- *PHP 5.6*
- *PHP 7.0*
- *PHP 7.1*
- *PHP 7.2*

PHP 4.4 Extensions

bcmath bz2 calendar ctype curl dba dbase dbx domxml exif fileinfo	ftp gd gettext gmp iconv imap interbase ioncube_loader ioncube_loader_4 json ldap	mbstring mcrypt mhash mysql ncurses odbc openssl overload pcntl pcre pgsql	posix pspell readline recode session shmop snmp sockets sourceguardian standard sybase_ct sysvmsg	sysvsem sysvshm tokenizer wddx xml xmlrpc zlib
---	---	--	--	--

PHP 5.1 Extensions

bcmath	gettext	lzf	pgsql	stem
big_int	gmagick	mbstring	posix	sybase_ct
bitset	gmp	mcrypt	pspell	sysvmsg
bz2	gnupg	memcache	quickhash	sysvsem
bz2_filter	haru	msgpack	radius	sysvshm
calendar	hash	mysql	readline	tidy
coin_acceptor	huffman	mysqli	redis	timezonedb
crack	iconv	ncurses	reflection	tokenizer
ctype	idn	odbc	session	translit
curl	igbinary	openssl	shmop	wddx
date	imagick	pcntl	simplexml	xdebug
dba	imap	pcr	snmp	xml
dbase	includ	pdo	soap	xmlreader
dom	inotify	pdo_firebird	sockets	xmlrpc
doublemetaphone	interbase	pdo_mysql	sourceguardian	xmlwriter
exif	ioncube_loader	pdo_odbc	spl	xsl
ftp	ioncube_loader_4	pdo_pgsql	ssh2	zlib
gd	ldap	pdo_sqlite	standard	
geoip	libxml		stats	

PHP 5.2 Extensions

apc	ftp	magickwand	pgsql	suhosin
apm	gd	mailparse	phar	sybase_ct
ares	gender	mbstring	posix	sysvmsg
bcmath	geoip	mcrypt	pspell	sysvsem
bcompiler	gettext	memcache	quickhash	sysvshm
big_int	gmagick	memcached	radius	tidy
bitset	gmp	mhash	rar	timezonedb
bloomy	gnupg	mongo	readline	tokenizer
bz2	haru	msgpack	recode	translit
bz2_filter	hash	mssql	redis	uploadprogress
calendar	hidef	mysql	reflection	uuid
coin_acceptor	htscanner	mysqli	rsync	wddx
crack	huffman	ncurses	session	xcache_3
ctype	iconv	oauth	shmop	xdebug
curl	idn	odbc	simplexml	xml
date	igbinary	opcache	snmp	xmlreader
dba	imagick	openssl	soap	xmlrpc
dbase	imap	pcntl	sockets	xmlwriter
dbx	included	pcre	sourceguardian	xrange
dom	inotify	pdf	spl	xsl
doublemetaphone	interbase	pdo	spl_types	yaf
eaccelerator	intl	pdo_dblib	sqlite	yaz
enchant	ioncube_loader	pdo_firebird	ssh2	zend_optimizer
exif	ioncube_loader_4	pdo_mysql	standard	zip
ffmpeg	json	pdo_odbc	stats	zlib
fileinfo	ldap	pdo_pgsql	stem	
filter	libxml	pdo_sqlite	stomp	
	lzf			

PHP 5.3 Extensions

apc	filter	magickwand	posix	sysvshm
apcu	ftp	mailparse	proprio	tidy
apm	functional	mbstring	pspell	timezonedb
ares	gd	mcrypt	quickhash	tokenizer
bcmath	gender	memcache	radius	trader
bcompiler	geoip	memcached	raphf	translit
big_int	gettext	mhash	rar	uploadprogress
bitset	gmagick	mongo	readline	uri_template
bloomy	gmp	msgpack	recode	uuid
brotili	gnupg	mssql	redis	wddx
bz2	haru	mysql	reflection	weakref
bz2_filter	hash	mysqli	rsync	xcache
calendar	hidef	mysqlnd	session	xcache_3
clamav	htscanner	ncurses	shmop	xdebug
coin_acceptor	http	nd_mysql	simplexml	xml
core	huffman	nd_mysqli	snmp	xmlreader
crack	iconv	nd_pdo_mysql	soap	xmlrpc
ctype	idn	oauth	sockets	xmlwriter
curl	igbinary	odbc	sourceguardian	xrange
date	imagick	opcache	spl	xsl
dba	imap	openssl	spl_types	yaf
dbase	included	pcntl	sqlite	yaml
dbx	inotify	pcre	sqlite3	yaz
dom	interbase	pdf	ssh2	zend_guard_1
doublemetaphone	intl	pdo	standard	oader
eaccelerator	ioncube_loader	pdo_dblib	stats	zip
eio	ioncube_loader_4	pdo_firebird	stem	zlib
enchant	jsmin	pdo_mysql	stomp	zmq
ereg	json	pdo_odbc	suhosin	
exif	ldap	pdo_pgsql	sybase_ct	
ffmpeg	libevent	pdo_sqlite	sysvmsg	
fileinfo	libxml	pgsql	sysvsem	
	lzf	phalcon		
		phar		

PHP 5.4 Extensions

apc	gender	mcrypt	posix	timezonedb
apcu	geoip	memcache	proprio	tokenizer
apm	gettext	memcached	pspell	trader
ares	gmagick	mhash	quickhash	translit
bcmath	gmp	mongo	radius	uploadprogress
big_int	gnupg	mongodb	raphf	uri_template
bitset	haru	msgpack	rar	uuid
brotili	hash	mssql	readline	wddx
bz2	hidef	mysql	recode	weakref
bz2_filter	htscanner	mysqli	redis	xcache
calendar	http	mysqlnd	reflection	xcache_3
clamav	iconv	ncurses	rsync	xdebug
core	igbinary	nd_mysql	session	xml
ctype	imagick	nd_mysqli	shmop	xmlreader
curl	imap	nd_pdo_mysql	simplexml	xmlrpc
date	included	oauth	snmp	xmlwriter
dba	inotify	oci8	soap	xrange
dbase	interbase	odbc	sockets	xsl
dbx	intl	opcache	sourceguardian	yaf
dom	ioncube_loader	openssl	spl	yaml
doublemetaphone	ioncube_loader_4	pcntl	spl_types	yaz
e	jsmin	pcre	sqlite3	zend_guard_loader
eaccelerator	json	pdf	ssh2	zip
eio	ldap	pdo	standard	zlib
enchant	libevent	pdo_dblib	stats	zmq
ereg	libsodium	pdo_firebird	stem	
exif	libxml	pdo_mysql	stomp	
ffmpeg	lzf	pdo_odbc	suhosin	
fileinfo	magickwand	pdo_pgsql	sybase_ct	
filter	mailparse	pdo_sqlite	sysvmsg	
ftp	mbstring	pgsql	sysvsem	
functional		phalcon	sysvshm	
gd		phar	tidy	

PHP 5.5 Extensions

apcu	gettext	memcache	phalcon	sybase_ct
apm	gmagick	memcached	phalcon3	sysvmsg
ares	gmp	mhash	phar	sysvsem
bcmath	gnupg	mongo	posix	sysvshm
big_int	gRPC	mongodb	propro	tidy
bitset	haru	msgpack	pspell	timezonedb
brotli	hash	mssql	quickhash	tokenizer
bz2	hidef	mysql	radius	trader
bz2_filter	htscanner	mysqli	raphf	translit
calendar	http	mysqlnd	rar	uploadprogress
clamav	iconv	ncurses	readline	uri_template
core	igbinary	nd_mysql	recode	uuid
ctype	imagick	nd_mysqli	redis	wddx
curl	imap	nd_pdo_mysql	reflection	weakref
date	inotify	oauth	rsync	xcache_3
dba	interbase	oci8	session	xdebug
dbase	intl	odbc	shmop	xml
dbx	ioncube_loader	opcache	simplexml	xmlreader
dom	ioncube_loader_4	openssl	snmp	xmlrpc
doublemetaphon	jsmin	pcntl	soap	xmlwriter
e	json	pcre	sockets	xrange
eio	ldap	pdf	sourceguardian	xsl
enchant	libevent	pdo	spl	yaf
ereg	libsodium	pdo_dblib	spl_types	yaml
exif	libxml	pdo_firebird	sqlite3	yaz
ffmpeg	lzf	pdo_mysql	ssh2	zend_guard_loader
fileinfo	magickwand	pdo_odbc	standard	zip
filter	mailparse	pdo_pgsql	stats	zlib
ftp	mbstring	pdo_sqlite	stem	zmq
gd	mcrypt	pgsql	stomp	
gender			suhosin	
geoip				

PHP 5.6 Extensions

apcu	gmagick	mongo	phar	suhosin
apm	gmp	mongodb	posix	sybase_ct
ares	gnupg	msgpack	propro	sysvmsg
bcmath	gRPC	mssql	pspell	sysvsem
big_int	haru	mysql	quickhash	sysvshm
bitset	hash	mysqli	radius	tidy
brotli	htscanner	mysqlnd	raphf	timezonedb
bz2	http	ncurses	rar	tokenizer
bz2_filter	iconv	nd_mysql	readline	trader
calendar	igbinary	nd_mysqli	recode	translit
core	imagick	nd_pdo_mysql	redis	uploadprogress
ctype	imap	oauth	reflection	uri_template
curl	inotify	oci8	rsync	uuid
date	interbase	odbc	session	wddx
dba	intl	opcache	shmop	weakref
dbx	ioncube_loader	openssl	simplexml	xcache_3
dom	ioncube_loader_4	pcntl	snmp	xdebug
doublemetaphone	jsmin	pcre	soap	xml
e	json	pdf	sockets	xmlreader
eio	ldap	pdo	sourceguardian	xmlrpc
enchant	libevent	pdo_dblib	spl	xmlwriter
ereg	libsodium	pdo_firebird	spl_types	xrange
exif	libxml	pdo_mysql	sqlite3	xsl
ffmpeg	lzf	pdo_odbc	ssh2	yaml
fileinfo	mailparse	pdo_pgsql	standard	yaz
filter	mbstring	pdo_sqlite	stats	zend_guard_loader
ftp	mcrypt	pgsql	stem	zip
gd	memcache	phalcon	stomp	zlib
gender	memcached	phalcon3		zmq
geoip	mhash			
gettext				

PHP 7.0 Extensions

apcu	geoip	memcached	posix	sysvsem
bcmath	gettext	mongodb	proprio	sysvshm
bitset	gmagick	mysqli	pspell	tidy
brotili	gmp	mysqlnd	raphf	timezonedb
bz2	gnupg	nd_mysqli	rar	tokenizer
calendar	gRPC	nd_pdo_mysql	readline	trader
core	hash	newrelic*	redis	uploadprogress
ctype	htscanner	oauth	reflection	uuid
curl	http	oci8	session	vips
date	iconv	odbc	shmop	wddx
dba	igbinary	opcache	simplexml	xdebug
dbase	imagick	openssl	snmp	xml
dom	imap	pcntl	soap	xmlreader
eio	inotify	pcr	sockets	xmlrpc
enchant	interbase	pdf	sourceguardian	xmlwriter
exif	intl	pdo	spl	xsl
fileinfo	ioncube_loader	pdo_dblib	sqlite3	yaml
filter	json	pdo_firebird	sqlsrv	yaz
ftp	ldap	pdo_mysql	ssh2	zip
gd	libsodium	pdo_odbc	standard	zlib
gender	libxml	pdo_pgsql	stats	zmq
	lzf	pdo_sqlite	suhosin	
	mailparse	pdo_sqlsrv	sysvmsg	
	mbstring	pgsql		
	mcrypt	phalcon3		
		phar		

* Please note that to use newrelic extension you should set your own New Relic License Key in your own `/opt/alt/php7*/etc/php.ini` file.

Please find more info about New Relic License Key in the [New Relic documentation](#).

PHP 7.1 Extensions

apcu	gmagick	mongodb	posix	sysvsem
bcmath	gmp	mysqli	proprio	sysvshm
brotli	gnupg	mysqlnd	pspell	tidy
bz2	gRPC	nd_mysqli	raphf	timezonedb
calendar	hash	nd_pdo_mysql	rar	tokenizer
core	htscanner	newrelic*	readline	trader
ctype	http	oauth	redis	uploadprogress
curl	iconv	oci8	reflection	uuid
date	igbinary	odbc	session	vips
dba	imagick	opcache	shmop	wddx
dbase	imap	openssl	simplexml	xdebug
dom	inotify	pcntl	snmp	xml
eio	interbase	pcre	soap	xmlreader
enchant	intl	pdo	sockets	xmlrpc
exif	ioncube_loader	pdo_dblib	sourceguardian	xmlwriter
fileinfo	json	pdo_firebird	spl	xsl
filter	ldap	pdo_mysql	sqlite3	yaml
ftp	libsodium	pdo_odbc	sqlsrv	zip
gd	libxml	pdo_pgsql	ssh2	zlib
gender	lzf	pdo_sqlite	standard	zmq
geoip	mailparse	pdo_sqlsrv	stats	
gettext	mbstring	pgsql	suhosin	
	mcrypt	phalcon3	sysvmsg	
	memcached	phar		

* Please note that to use newrelic extension you should set your own New Relic License Key in your own `/opt/alt/php7*/etc/php.ini` file.

Please find more info about New Relic License Key in the [New Relic documentation](#).

PHP 7.2 Extensions

apcu	gmagick	mysqli	posix	sysvshm
bcmath	gmp	mysqlnd	proprio	tidy
brotli	gnupg	nd_mysqli	pspell	timezonedb
bz2	gRPC	nd_pdo_mysql	raphf	tokenizer
calendar	hash	newrelic*	readline	trader
core	http	oauth	redis	uploadprogress
ctype	iconv	oci8	reflection	uuid
curl	igbinary	odbc	session	vips
date	imagick	opcache	shmop	wddx
dba	imap	openssl	simplexml	xml
dom	inotify	pcntl	snmp	xmlreader
eio	interbase	pcre	soap	xmlrpc
enchant	intl	pdo	sockets	xmlwriter
exif	ioncube_loader	pdo_dblib	spl	xsl
fileinfo	json	pdo_firebird	sqlite3	yaml
filter	ldap	pdo_mysql	sqrsv	zip
ftp	libxml	pdo_odbc	ssh2	zlib
gd	lzf	pdo_pgsql	standard	zmq
gender	mailparse	pdo_sqlite	stats	
geoip	mbstring	pdo_sqrsv	sysvmsg	
gettext	memcached	pgsql	sysvsem	
	mongodb	phalcon3		
		phar		

* Please note that to use newrelic extension you should set your own New Relic License Key in your own `/opt/alt/php7*/etc/php.ini` file.

Please find more info about New Relic License Key in the [New Relic documentation](#).

Disabling PHP extension globally

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).

Control Panel Integration

•cPanel

PHP Selector Integration with cPanel MultiPHP Manager

[Requires CageFS 5.5-6.18+]

When using EasyApache4 in cPanel, it is possible to change PHP versions for users' domains with MultiPHP Manager (when PHP is working under Apache web server). Also it is possible to change system default PHP version with MultiPHP Manager in WHM.

MultiPHP Manager in WHM looks as follows:

lcPanel_integrationl

A user can change PHP version for domain in cPanel interface but can not change System default PHP version.

lcPanel_integration01l

The following Alt-PHP packages (and higher) provide an ability to select Alt-PHP version in MultiPHP Manager:

- alt-php44-4.4.9-71;
- alt-php51-5.1.6-81;
- alt-php52-5.2.17-107;
- alt-php53-5.3.29-59;
- alt-php54-5.4.45-42;
- alt-php55-5.5.38-24;
- alt-php56-5.6.31-7;
- alt-php70-7.0.24-2;
- alt-php71-7.1.10-2;
- alt-php72-7.2.0-0.rc.3.2.

You can remove Alt-PHP from cPanel MultiPHP Manager.

To do so set ‘yes’ or ‘no’ for the Alt-PHP versions in config file `/opt/alt/alt-php-config/alt-php.cfg` and run `/opt/alt/alt-php-config/multiphp_reconfigure.py`.

This script manages SCL prefixes for the Alt-PHP - removes or creates prefixes in `/etc/scl/prefixes`.

`/opt/alt/alt-php-config/alt-php.cfg`

[MultiPHP Manager]

alt-php44 = no

alt-php51 = no

alt-php52 = no

alt-php53 = no

alt-php54 = no

alt-php55 = yes

alt-php56 = yes

alt-php70 = yes

alt-php71 = yes

alt-php72 = yes

Note. PHP Selector does not work when Alt-PHP version is selected as system default in MultiPHP Manager. So, all domains will use PHP version selected via MultiPHP Manager. Settings in PHP Selector will be ignored. We recommend to disable PHP Selector in such case.

PHP Selector works in different ways with EasyApache4 and EasyApache3. CageFS should be enabled for users who use PHP Selector. The novation is that when using EasyApache4, actual PHP version used depends on PHP version selected in MultiPHP Manager. When PHP version chosen for domain in MultiPHP Manager matches System default

PHP version, then PHP Selector is used to select actual PHP version. If PHP version chosen for domain in MultiPHP Manager differs from System default PHP version, then PHP version from MultiPHP Manager is used.

In other words, PHP Selector deals with changing System default PHP version.

PHP Selector algorithm for choosing PHP version for domain is as follows:

1. If CageFS is disabled, then PHP Selector is not active and MultiPHP Manager PHP version is applied.
2. If CageFS is enabled, then:
 - 2.1. If PHP version chosen in MultiPHP Manager differs from System default PHP version, then MultiPHP Manager PHP version is applied.
 - 2.2. If PHP version chosen in MultiPHP Manager is the same as System default PHP version, then PHP Selector PHP version is applied:
 - 2.2.1. If Native option is selected in PHP Selector, then MultiPHP Manager PHP version is applied.
 - 2.2.2. If PHP version chosen in PHP Selector differs from Native, then PHP Selector PHP version is applied.

lcPanel_integration02l

lcPanel_integration03l

lcPanel_integration04l

PHP version chosen in MultiPHP Manager can also be applied to console commands `/usr/bin/php` and `/usr/local/bin/php`. In this case `.htaccess` file search is performed in current directory and in parent directories. If the file is found, then PHP version specified in it is applied, if not found, then System default PHP version is applied. System default PHP version can be changed via PHP Selector.

1. If CageFS is disabled, then PHP Selector is not active and PHP version from `.htaccess` is applied.
2. If CageFS is enabled, then:
 - 2.1. If PHP version specified in `.htaccess` file differs from System default, then `.htaccess` version is applied.
 - 2.2. If System default PHP version is specified in `.htaccess` file, then PHP Selector version is applied:
 - 2.2.1. If Native option is chosen in PHP Selector, then `.htaccess` PHP version is applied.
 - 2.2.2. If PHP version chosen in PHP Selector differs from Native, then PHP Selector version is applied.

Note. cPanel prior to 11.56 does not support hooks to add processing of System default PHP version changes with MultiPHP Manager. That is why System default PHP version changing is handled by cron job (`/etc/cron.d/cagefs_cron` file), which executes the command `/usr/share/cagefs/setup_multiphp_integration` every ten minutes, which means that all System default PHP version changes in MultiPHP Manager are applied in CageFS with 10 minutes delay.

In cagefs-5.5-6.25 or later, changing of System default PHP version with MultiPHP Manager will be processed with cPanel WHM hooks.

PHP Modules

The set of PHP modules depends on PHP version used for domain or console. If PHP Selector is active and Alt-PHP version is chosen, then modules chosen for this Alt-PHP version in PHP Selector are used. If PHP Selector is not active, then modules for PHP version chosen in cPanel MultiPHP are used.

PHP Options

cPanel has MultiPHP INI Editor available in WHM and in cPanel user interface.

MultiPHP INI Editor allows setting PHP options for any PHP version globally for all domains and users. At this point `/opt/cpanel/ea-php56/root/etc/php.d/local.ini` file is generated and options values are written into this file. Such options have higher priority than the options set in MultiPHP INI Editor in cPanel user interface. MultiPHP INI Editor allows to set PHP options in Basic Mode (simplified interface) and in Editor Mode.

MultiPHP INI Editor in WHM looks as follows:

lcPanel_integration05l

lcPanel_integration06l

Note. cPanel prior to 11.56 does not support hooks to add processing of INI options changing for PHP version with MultiPHP INI Editor in cPanel WHM. That is why for now the processing of PHP version changing is handled by cron job (/etc/cron.d/cagefs_cron file) which performs the command /usr/share/cagefs/ setup_multiphp_integration every 10 minutes, which means that INI options changes for PHP version in MultiPHP INI Editor in cPanel WHM are being applied with up to 10 minutes delay.

In cagefs-5.5-6.25 or later, INI options changes for PHP version in MultiPHP INI Editor in cPanel WHM will be processed by cPanel WHM hooks.

MultiPHP INI Editor in cPanel user interface allows setting options for php.ini files in user home directory or in domain docroot. Changes are applied immediately without delay.

These options priority is lower than ones specified in MultiPHP INI Editor WHM interface. MultiPHP INI Editor in cPanel user interface looks as follows

lcPanel_integration07l

lcPanel_integration08l

If PHP Selector is active, then options set in PHP Selector are applied, and such options have higher priority than options in custom php.ini file in domain docroot. If PHP Selector is disabled, then options set in MultiPHP INI Editor are applied.

QUIRKS: When changing System default PHP version, administrator should take into consideration the following quirk. For example, if a user has chosen PHP 5.3 for domain and System default PHP version is PHP 5.5, then PHP Selector will not be used for user domain. In this case, if administrator switches System default PHP version from 5.5 to 5.3, then PHP Selector will be activated for user domain and PHP version chosen in PHP Selector will be applied for domain.

That is why it is recommended for administrator to avoid changing System default PHP version to PHP version that is already used by users. At the same time it is recommended for users to choose inherit for domain and use PHP Selector to choose PHP version. In this case PHP version chosen in PHP Selector will be always applied for domain.

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. It supports only cPanel servers.

Supported Alt-Python versions:

alt-python27 2.7.9, supported by CloudLinux 6, CloudLinux 7;

alt-python33 3.3.2, supported by CloudLinux 6, CloudLinux 7;

alt-python34 3.4.1, supported by CloudLinux 6, CloudLinux 7;

alt-python36-3.6.3-1, supported by CloudLinux 6, CloudLinux 7.

Supported Alt-Ruby versions (supported by CloudLinux 6 and CloudLinux 7):

Alt-Ruby 1.8;

Alt-Ruby 1.9;

Alt-Ruby 2.0;

Alt-Ruby 2.1;

Alt-Ruby 2.2;

Alt-Ruby 2.3;

Alt-Ruby 2.4.

Python and Ruby Selector Installation

Install a tools to create isolated Python environments and Passenger Apache module. For servers with EasyApache3:

```
yum install lve-manager alt-python-virtualenv alt-mod-passenger
```

with EasyApache4:

```
yum install lve-manager alt-python-virtualenv ea-apache24-mod-alt-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
```

NOTE. After installation, please make sure that you have unmarked appropriate checkboxes in LVE Manager Options tab to show Ruby or Python App in web-interface.

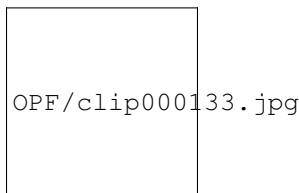
Find the instructions on the [link](#).

NOTE. Adding Python and Ruby modules requires executing permissions to gcc/make binaries. Please enable compilers in Compiler Access section of WHM, then run:

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

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 URI for accessing your application. Click “Create project” to create an application.



OPF/clip000233.jpg

After a little while a new application entry will be appended to the web-page.



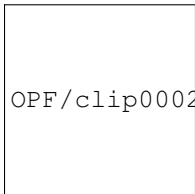
OPF/clip000255.jpg

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.



OPF/clip000256.jpg

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.



OPF/clip000257.jpg

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.



OPF/clip000261.jpg

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 unmoved.

Note. For LVE Manager version 0.9-10 and higher:

When creating an application you can use the key `--domain`, which attaches application to domain. If `--domain` key is not specified, then the main users domain will be used by default.

To create application run:

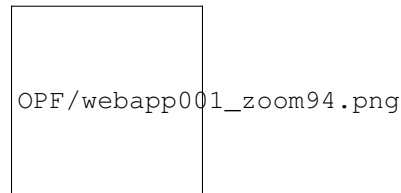
```
/usr/bin/selectorctl --interpreter=<python|ruby> --version=VERSION  
[--user=USER] [--domain=DOMAIN] [--print-summary] [--json]  
--create-webapp <FOLDER_NAME> <URI>
```

When changing application URI, `--domain` key can be used simultaneously, in this case not only URI will be changed, but also the application domain.

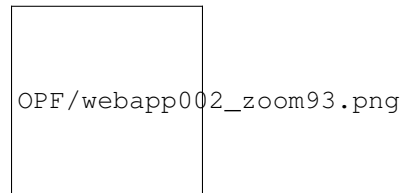
To change application URI run:

```
/usr/bin/selectorctl --interpreter=<python|ruby> [--user=USER]  
[--domain=NEW_DOMAIN] [--print-summary] [--json] --transit-webapp  
<FOLDER_NAME> <NEW_URI>
```

The possibility to choose domain when creating an application was added to web interface as well.



Also you can run simple commands from web interface (e.g. you can install packages from specific repositories or control web applications by means of `django-admin`).



Command Line

All the actions mentioned in Deploy and Settings section can be performed 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 --ver-
sion=<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
```

To restart application:

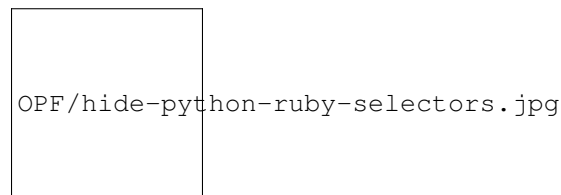
```
selectorctl --interpreter ruby --user cltest1 --domain cltest1.com --restart-webapp testapp
```

To choose Ruby version:

```
selectorctl --interpreter=ruby --user=$USER -v 2.0
```

Hide Python and Ruby Selector Icons

It is possible to hide or show Python and Ruby Selector icons by marking or unmarking proper checkboxes in LVE Manager Options tab:



Note. You also can hide/show CloudLinux Plugins in cPanel using Feature Manager.

Deploying Trac using Python Selector

1. In Setup Python App create an application. Trac project WSGI script will be located in App Directory (e.g. trac).

App URI – is a URL where web-interface is located. (e.g. Trac – web-interface is located in YOUR_DOMAIN/trac).

Trac needs Python version from 2.5 to 3.0, in actual example version 2.7 is used.

2. When the App is created, add the following modules: Trac, Genshi, MySQL-python.

2.1. Alternatively connect to the server via SSH and perform the following steps:

```
source ~/virtualenv/trac/2.7/bin/activate;
```

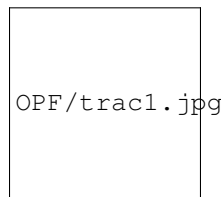
then:

```
~/virtualenv/trac/2.7/bin/easy_install Trac mysql-python (using easy_install);
```

or

```
~/virtualenv/trac/2.7/bin/pip install trac mysql-python (using pip).
```

3. In cPanel create MySQL database and a user. Add user to database.



In this example DB tractest_trac and user tractest_trac were created.

4. Connect to the server via SSH using your cPanel account.

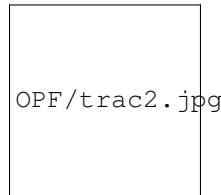
Create Trac project:

```
~/virtualenv/trac/2.7/bin/trac-admin ~/trac_project initenv
```

For “Database connection string” parameter enter the following: mysql://user:password@localhost/database_name – here the data for connecting MySQL database are specified.

Note. In case of “... The charset and collation of database are ‘latin1’ and ‘latin1_swedish_ci’ error the database must be created with one of ((‘utf8’, ‘utf8_bin’), (‘utf8mb4’, ‘utf8mb4_bin’)) ...” while creating the project, you should change database encoding.

To change encoding, in cPanel run phpMyAdmin, choose DB, go to Operations, choose the necessary encoding in Collation section and click Go.



After that you have to repeat the procedure of creating a project. When done, the Trac project must appear: ~/trac_project

5. To create project frontend run the following:

```
~/virtualenv/trac/2.7/bin/trac-admin ~/track_project deploy ~/trac
```

~/track_project — is the path to the project,

~/trac — is the path, that was specified while setting App Directory.

Create topic directory by default:

```
cd ~/public_html/trac
```



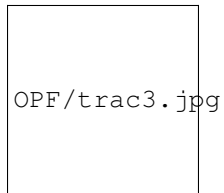
```
mkdir chrome
```

`cp -R ~/trac/htdocs/ ~/public_html/trac/chrome/` - all project static files are located in this directory; the changes can be added here as well.

6. To add path to WSGI file in created application:

Go back to cPanel Setup Python App, change “WSGI file location” for your application to `cgi-bin/trac.wsgi`, click Update to apply changes and then click Restart.

Your Existing application now must look like the following:



7. Adding authorization:

In `~/public_html/trac/.htaccess` after CLOUDLINUX PASSENGER CONFIGURATION section add the following lines:

```
AuthType Basic
```

```
AuthName "trac"
```

```
AuthUserFile /home/tracetest/trac/passwd
```

```
Require valid-user
```

8. Add new user and create passwd file `/usr/local/apache/bin/htpasswd` with `~/trac/passwd` admin.

Enter password.

```
~/virtualenv/trac/2.7/bin/trac-admin ~/track_project permission add admin TRAC_ADMIN
```

Add admin user to TRAC_ADMIN group.

Here the path trac directory is equal to App Directory in your project.

Now Trac is available via `YOUR_DOMAIN/trac`.

Trac with MySQL

To use Trac with MySQL database you should install `alt-python27-devel` package.

To install run:

```
yum install alt-python27-devel --enablerepo=cloudlinux-updates-testing
```

Deploying Redmine using Ruby Selector

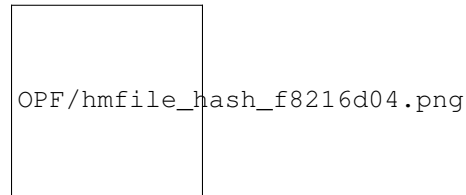
Note. Provided instructions are valid for older Redmine version 2.6.0 . New versions guide could be found at <http://kb.cloudlinux.com/2016/12/how-to-run-redmine-with-ruby-selector/>

1. In cPanel create MySQL database and add user to it. In the example given, the database `redminet_redmine` was created and user `redminet_redmine` was added.

2. In Setup Ruby App section create an application.

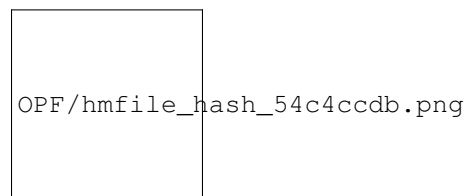
App Directory is the directory where all static files will be placed (e.g. redmine).

App URI is web-interface URL (e.g. redmine web-interface will be located in YOUR_DOMAIN/redmine).



3. After the application was created, add the following modules:

bundle, i18n#0.6.11, builder#3.0.4, rails#3.2.19, mime-types#1.25.1, mocha#1.0.0, jquery-rails#3.1.2, coderay, fasterscv, request_store, rbpdf, mysql2, selenium-webdriver, rmagick, shoulda#3.3.2, ruby-openid#2.3.0, request_store#1.0.5, capybara#2.1.0, net-ldap#0.3.1, rack-openid, shoulda-matchers#1.4.1, redcarpet#2.3.0, yard, rake#10.4.2, bigdecimal.



Note: If error occurs while installing rmagic module, then you need to install ImageMagick-devel package on your server:

```
yum install ImageMagick-devel
```

The installation process takes quite along time, about 7-8 minutes. When done, click Restart button to restart the application.

!redmine after install extensions\1!

!redmine after install extensions\2!

3.1 Alternatively, after the application was created, you can add only one module - bundle.

!redmine_selector\1!

!redmine_selector\2!

4. Enter the server via SSH, using your cPanel account.

Download the application <http://www.redmine.org/projects/redmine/wiki/Download>.

In the description given, the latest version Redmine (2.6.0) is assumed.

<http://www.redmine.org/releases/redmine-2.6.0.tar.gz>

```
tar xzf redmine-2.6.0.tar.gz
```

Hereinafter 'redmine' is App Directory meaning which was specified while setting Ruby application.

```
cp -R ~/redmine-2.6.0/* ~/redmine
```

```
cd ~/redmine/config
```

```
cp database.yml.example database.yml
```

Edit config/database.yml - add MySQL database connection settings to Production section.

```
cp -R ~/redmine/public/* ~/public_html/redmine/
```

```
cd ~/public_html/redmine
```

```
cat htaccess.fcgi.example >> .htaccess
```

```
cp dispatch.fcgi.example dispatch.fcgi
```

Go to `cd ~/redmine` directory.

Add gem “bigdecimal” line into Gemfile file.

Run alternately:

```
source ~/rubyvenv/redmine/2.1/bin/activate
```

```
~/rubyvenv/redmine/2.1/bin/bundle install (if running the alternative installation)
```

```
~/rubyvenv/redmine/2.1/bin/rake generate_secret_token
```

```
RAILS_ENV=production ~/rubyvenv/redmine/2.1/bin/rake db:migrate - Database migration;
```

```
RAILS_ENV=production ~/rubyvenv/redmine/2.1/bin/rake redmine:load_default_data - Loading default data into database.
```

Easy Apache 4

Since cPanel and WHM version 66 provides `ea-ruby24-mod_passenger` (more information on the [link](#)), this allows creating Ruby applications with cPanel application manager.

CloudLinux already has Python and Ruby Selector, which allows creating applications with `ea-apache24-mod-alt-passenger`. However, it does not allow using cPanel application manager.

It is not correct to install both of those packages on the server because they contain the same passenger module for Apache web server.

The new `ea-ruby24-mod_passenger` is available for download from our updates-testing (beta) repository which allows you to run applications via cPanel application manager and CloudLinux Python and Ruby Selector.

To install run:

```
# yum install lvemanager alt-python-virtualenv
```

```
# yum install ea-ruby24-mod_passenger --enablerepo=cl-ea4-testing
```

To install Ruby or Python Selector follow the instructions on the [link](#).

Node.js Selector

Overview & Requirements

•Requirements

Installation

Command Line Interface

•Hoster

•End User

User Interface

•Hoster

- oHow to enable/disable Node.js

- oHow to manage Node.js

- o*Applications column*

- End User*

- o*How to manage application*

- Node.js Deployment*

- o*Remote Usage of Node.js Interpreters*

- o*Remote Usage of the cloudlinux-selector Utility*

Overview & Requirements

Node.js Selector is a CloudLinux component that allows each user to easily create Node.js applications, choose Node.js version and other parameters for applications based on their needs.

‘<>’_Requirements

- Node.js Selector supports Node.js versions 6.x, 8.x, 9.x and later.
- This feature is available for CloudLinux 7, CloudLinux 6 hybrid and CloudLinux 6.
- Node.js Selector requires LVE Manager 4.0 or later.
- It supports cPanel and DirectAdmin servers (Plesk is not supported as it already has Node.js support.) For more details, please go to Plesk & Node.js documentation [here](#) and [here](#).
- For more details about mod_passenger and Node.js, please read documentation [here](#) and [here](#).
- Node.js Selector is working with EasyApache 3 and EasyApache 4.

Installation

cPanel

To use Node.js Selector, please install Node.js packages by running the following command:

```
yum groupinstall alt-nodejs6 alt-nodejs8 alt-nodejs9
```

Also, please install LVE Manager, LVE Utils and Fusion Passenger by running the following command:

```
yum install lvemanager lve-utils ea-apache24-mod-alt-passenger
```

For EasyApache 3:

```
yum install lvemanager lve-utils alt-mod-passenger
```

And we recommend to install CageFS for better security (not mandatory) by running the following command:

```
yum install cagefs
```

Note. If during Node.js Selector usage on cPanel servers you get “ENOMEM npm ERR! errno -12” error, try to increase Memory limit in

cPanel WHM → Server Configuration → Tweak Settings → System → Max cPanel process memory, then restart cPanel service with the following command to apply changes.

CloudLinux 7:

```
systemctl restart cpanel.service
```

CloudLinux 6:

```
service cpanel restart
```

DirectAdmin

To use Node.js Selector, please install Node.js packages by running the following command:

```
yum groupinstall alt-nodejs6 alt-nodejs8 alt-nodejs9
```

Also, please install LVE Manager, LVE Utils and Fusion Passenger by running the following command:

```
yum install lvemanager lve-utils alt-mod-passenger
```

And we recommend to install CageFS for better security (not mandatory) by running the following command:

```
yum install cagefs
```

Command Line Interface

Below is a list of commands hoster and end user can run in a command line.

CHAPTER 10

‘ <> ‘__Hoster

=

•Get information related to Node.js: default version, list of supported versions, status of Node.js Selector, list of users, their applications, etc:

cloudlinux-selector [get] [-json] --interpreter nodejs

JSON output for get command:

```
{
  "selector_enabled": true | false,
  "default_version": "6.11.3",
  "result": "success",
  "timestamp": 1508667174.220027
  "cache_status": "ready"      // or "updating" during automatic
```

yum cache rebuild

```
  "available_versions": { // begin of "versions"
    "6.11.3": { // begin of version "6.11.3"
      "name_modifier": "",
      "status": "enabled", // enabled, disabled,
```

not_installed, installing, removing

```
      "base_dir": "/opt/alt/alt-nodejs6" // empty when
```

version is not installed

```
    "users": { // begin of "users"
      "user1": { // begin of "user1"
        "homedir": "/home/user1",
```

```
    "applications": { // begin of "applications"
        "apps_dir/app1": { // begin of
application "apps_dir/app1"
            "domain": "cltest1.com",
            "app_uri": "apps/my-app1",
            "app_mode": "development",
            "startup_file": "app.js",
            "app_status": "started", // 'started'
or 'stopped'
            "config_files": [
                "package.json",
                "gruntfile.js"
            ],
            "env_vars": {
                "var1": "value1",
                "var2": "value2"
            },
        }, // end of application "apps_dir/app1"
        "apps_dir/app2": { // begin of
application "apps_dir/app2"
            << data for application
"apps_dir/app2" (same structure as for application "apps_dir/app1" above) >>
        }, // end of application "apps_dir/app2"
    }, // end of "applications"
}, // end of "user1"
"user2": { // begin of "user2"
    << data for user "user2" (same structure as
for "user1" above) >>
    }, // end of "user2"
}, // end of "users"
}, // end of version "6.11.3"
"8.21.5": { // begin of version "8.21.5"
    << data for version "8.21.5" (same structure as for
version "6.11.3" above) >>
    }, // end of version "8.21.5"
}, // end of "versions"
```



```
} // end of json
```

- Set default version, supported versions, and status of Node.js

Selector:

```
cloudlinux-selector set [-json] --interpreter nodejs (--selector-status <enabled,disabled> | --default-version <str> | --supported-versions <str>)
```

Note that Node.js Selector is disabled by default. If an available Node.js version is not installed Node.js Selector is always disabled and it is impossible to enable it.

To set default Node.js version, please use the following command (note that required Node.js version should be enabled):

```
cloudlinux-selector set --json --interpreter=nodejs --default-version=<ver>
```

Examples:

This command enables Node.js Selector:

```
cloudlinux-selector set --json --interpreter nodejs --selector-status enabled
```

This command sets default Node.js version as 6:

```
cloudlinux-selector set --json --interpreter nodejs --default-version 6
```

This command sets supported Node.js version as 8:

```
cloudlinux-selector set --json --interpreter nodejs --supported-versions='{ "6": false, "8": true }'
```

- Install required Node.js version:

```
cloudlinux-selector install-version --json --interpreter nodejs --version 8
```

- Uninstall required Node.js version:

```
cloudlinux-selector uninstall-version --json --interpreter nodejs --version 8
```

- Enable required Node.js version:

```
cloudlinux-selector enable-version --json --interpreter nodejs --version 8
```

- Disable required Node.js version (note that it is impossible to disable default Node.js version):

```
cloudlinux-selector disable-version --json --interpreter nodejs --version 8
```

Change version for application(s):

```
cloudlinux-selector set [-json] --interpreter nodejs ((--user <str> | --domain <str>) --app-root <str> | --from-version <str>) --new-version <str>
```

Examples:

This command changes version for the specific application:

```
cloudlinux-selector set --json --interpreter nodejs --user user1 --app-root apps_dir/app1 --new-version 8
```

Common output for all set commands:

in case of success:

```
{
  "result": "success",
  "timestamp": 1508666792.863358
}
```

in case of error:

```
{
  "result": "Some error message",
  "details": "Traceback: ...",
  "context": {},
  "timestamp": 1508666792.863358
}
```

in case of warning:

```
{
  "result": "success",
  "warning": "Some warning message",
  "context": {},
  "timestamp": 1508666792.863358
}
```

To resolve issues related to install-version/uninstall-version commands (because they are running in the background) you may use this log file `/var/log/cl-nodejs-last-yum.log`

It contains full yum output from the latest performed operation (install or uninstall) and it will be rewritten with each operation.

=

WARNING: options `-user` and `-domain` are mutually exclusive now.

- Get config file for the user applications

`cloudlinux-selector read-config [-json] -interpreter nodejs [(-user <str> | -domain <str>)] -app-root <str> -config-file <name>`

JSON output:

```
{
  "result": "success",
  "timestamp": 1508666792.863358
  "data": "content of config file as Base64 encoded string"
}
```

Example:

This command gets config file for user1's application app1:

`cloudlinux-selector read-config -json -interpreter nodejs -user user1 -app-root app_dir/app1 -config-file package.json`

- Save config file for the user applications

`cloudlinux-selector save-config [-json] -interpreter nodejs [(-user <str> | -domain <str>)] -app-root <str> -config-file <path> -content <content of config file as Base64 encoded string>`

JSON output (the same as for all set commands):

```
{
  "result": "success",
  "timestamp": 1508666792.863358
}
```

Example:

This command saves config file for user1's application app1:

```
cloudlinux-selector save-config -json -interpreter nodejs -user user1 -app-root app_dir/app1 -config-file package.json -content VGh1ICAyIE5vdiAxMDo0MzoxMiBFRFQgMjAxNwo=
```

- Get a list of applications for the specific user

```
cloudlinux-selector [get] [-json] -interpreter nodejs [(-user <str> | -domain <str>)]
```

Example:

This command gets a list of applications for the user1:

```
cloudlinux-selector get -json -interpreter nodejs -user user1
```

- Create user application

```
cloudlinux-selector create [-json] -interpreter nodejs [(-user <str> | -domain <str>)] -app-root <str> -app-uri <str> [-version <str>] [-app-mode <str>] [-startup-file <str>] [-env-vars <json string>]
```

Example:

This command creates user1's application for the domain xyz.com:

```
cloudlinux-selector create -json -interpreter nodejs -user user1 -app-root my_apps/app1 -app-uri apps/app1
```

or

```
cloudlinux-selector create -json -interpreter nodejs -app-root my_apps/app1 -domain xyz.com -app-uri apps/app1
```

- Start, restart, stop, and destroy user application

```
cloudlinux-selector (start | restart | stop | destroy) [-json] -interpreter nodejs [(-user <str> | -domain <str>)] -app-root <str>
```

Example:

This command starts user1's application:

```
cloudlinux-selector start -json -interpreter nodejs -user user1 -app-root my_apps/app1
```

- Change properties for an application

```
cloudlinux-selector set [-json] -interpreter nodejs [(-user <str> | -domain <str>)] -app-root <str> [-app-mode <str>] [-new-app-root <str>] [-new-domain <str>] [-new-app-uri <str>] [-new-version <str>] [-startup-file <str>] [-env-vars <json string>]
```

Example 1:

This command sets a production mode, new domain new.xyz.com, new Node.js version 8, new URI, new application root directory and new startup file for user1 application located on the domain xyz.com:

```
cloudlinux-selector set -json -interpreter nodejs -user user1 -app-root my_apps/app1 -mode production -new-app-root new_apps/new_app1 -new-domain new.xyz.com -new-app-uri new_apps/app1 -new-version 8 -startup-file new_app.js -env-vars '{ "var1": "value1", "var2": "value2" }'
```

Example 2:

```
cloudlinux-selector set -json -interpreter nodejs -domain xyz.com -app-root my_apps/app1 -mode production -new-app-root new_apps/new_app1 -new-domain new.xyz.com -new-app-uri new_apps/app1 -new-version 8 -startup-file new_app.js -env-vars '{ "var1": "value1", "var2": "value2" }'
```

Note that when changing Node.js version all replies from web application to get request will be checked in Node.js Selector (before and after version changing). HTTP response codes and MIME type are comparing. So, make sure application is available via http(s) at least locally.

Run npm install command for the user application

```
cloudlinux-selector install-modules [-json] --interpreter nodejs [(-user <str> | -domain <str>)] --app-root <str>
```

Example:

This command runs npm install for user1 application:

```
cloudlinux-selector install-modules -json --interpreter nodejs --user user1 --app-root my_apps/app
```

Note that all replies from web application to get request will be checked in Node.js Selector (before and after modules installation). HTTP response codes and MIME type are comparing. So, make sure application is available via http(s) at least locally.

- Run a script from package.json file of a user application, arguments <args> are passed to the script

```
cloudlinux-selector run-script [-json] --interpreter nodejs [(-user <str> | -domain <str>)] --app-root <str> --script-name <str> [- <args>...]
```

Example:

```
cloudlinux-selector run-script -json --interpreter nodejs --user user1 --app-root my_apps/app --script-name test_script --script_opt1 --script_opt2 script_arg1 script_arg2
```

JSON output:

```
{
  "result": "success",
  "timestamp": 1508666792.863358
  "data": "script output as Base64 encoded string"
}
```

- Activate virtual environment of NodeJS:

```
source <home_of_user>/nodeenv/<app_root>/<nodejs_version>/bin/activate
```

This command changes prompt to

Example:

```
[newusr@192-168-245-108 ~]$ source /home/newusr/nodeenv/newapp4/newapp3/8/bin/activate
[newapp4/newapp3 (8)] [newusr@192-168-245-108 ~]$
```

After activation user can use npm and node from a virtual environment without full paths.

User Interface

CHAPTER 12

‘ <> ‘__Hoster

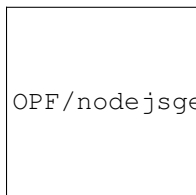
=

Hoster interface allows to enable and disable Node.js, and manage individual Node.js versions.

Go to LVE Manager → Options Tab → Node.js Section. A list of installed Node.js versions is displayed. There are several columns in the list.

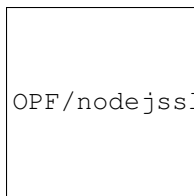
- Version — displays Node.js version.
- Path — Node.js package location.
- Applications — number of applications that use this Node.js version. Click on a digit to go to the list of applications.
- Enabled — displays if particular Node.js version is enabled.
- Actions — allows to install, delete, and make default a particular Node.js version.

To display all changes immediately click Refresh link.



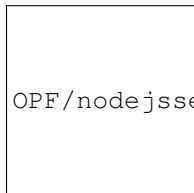
‘ <> ‘__How to enable/disable Node.js

- To enable Node.js move a slider to Enable.
- To disable Node.js move a slider back to Disable. Please note that if you disable Node.js its version for all your applications will not be changed but you can not add a new application to this version.



OPF/nodejsslider_zoom70.png

Note that Node.js Selector icon in end user interface is absent when Node.js is disabled.



OPF/nodejsselectorlogo_zoom70.png

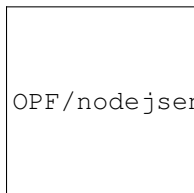
‘<>’__How to manage Node.js

The list of installed Node.js versions allows to enable and disable, install and delete, and set a particular Node.js version as a default.

Enable and disable particular Node.js version

To enable particular Node.js version do the following:

- Move a disabled slider in the Enabled column for a particular Node.js version.
- In the confirmation pop-up click Agree to save changes or Cancel to close pop-up.



OPF/nodejsenable_zoom70.png

To disable particular Node.js version do the following:

- Move an enabled slider in the Enabled column for a particular Node.js version.
- In the confirmation pop-up click Agree to save changes or Cancel to close pop-up.

Install and delete particular Node.js version

To install particular Node.js version do the following:

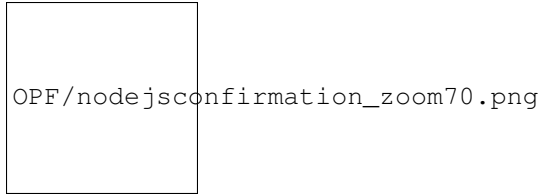
- Click Install button in the Actions column for a particular Node.js version.
- In the confirmation pop-up click Agree to save changes or Cancel to close pop-up.

To delete particular Node.js version do the following:

- Click Bin icon in the Actions column for a particular Node.js version.
- In the confirmation pop-up click Agree to start uninstall process.
- Or close a pop-up without changes.

Note that it is impossible:

- to remove default Node.js version;
- to remove version with applications;
- to install or remove version if another installation/uninstall process is running.

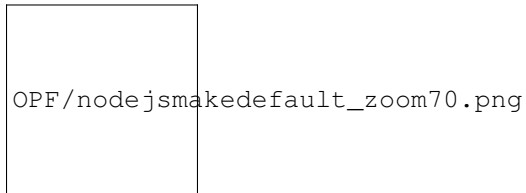


Make a particular Node.js version as a default

To make a particular Node.js version as a default do the following:

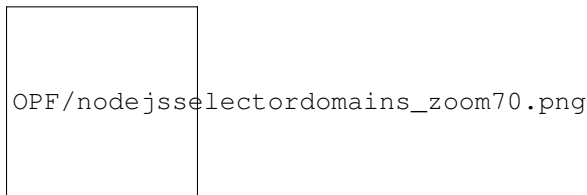
- Click Double-Tick icon in the Actions column for a particular Node.js version.
- In the confirmation pop-up click Agree to save changes or Cancel to close pop-up.

Note that it is impossible to make default disabled version.



‘<>’ __Applications column

To view and operate with the list of domains with Node.js versions click on a number in the Applications column for a particular Node.js version. A section with a list of Domains for particular Node.js version will be displayed.



Domains are displayed by three. To load more domains click on Load More button.

To change Node.js version for a particular application do the following:

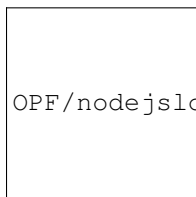
- Click Double-Arrow icon in the Actions column in a particular application row. A confirmation pop-up will be displayed.
- In the pop-up choose Node.js version from a drop-down.
- Click Change to confirm the action or Cancel to close the pop-up.
- To refresh state of applications in current version you can click Refresh link.

Note that all packages of the application(s) will be re-installed.

CHAPTER 13

‘<>’__End User

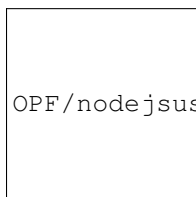
Note that Node.js Selector icon in end user interface is absent when Node.js is disabled.



End User interface allows end users to setup and manage Node.js for their web applications.

Go to cPanel → Software Section → Select Node.js Version.

Web Applications page is displayed.



There are several columns in the list.

- App URI — application URI including the domain.
- App Root Directory — application root directory relative to user's home.
- Mode — can be production or development.
- Status — started/stopped — displays if an application is running or not and version of application.
- Actions — allows to start, restart, stop, edit, and remove a particular application.

‘<>’__How to manage application

Start application

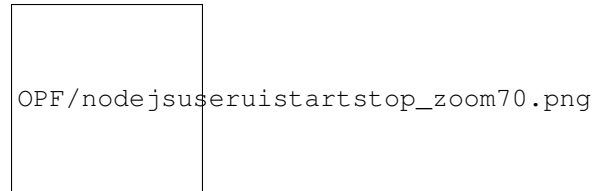
To start a stopped application do the following:

- Click Start icon in the Actions column in a stopped application row.
- When an action is completed a Start icon changes to Stop icon.

Stop application

To stop a started application do the following:

- Click Stop icon in the Actions column in a started application row.
- When an action is completed a Stop icon changes to Start icon.



Restart application

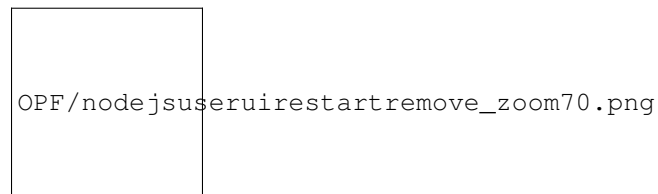
To restart started application do the following:

- Click Restart icon in the Actions column in a started application row. A current row is blocked and when a process is completed it will be unblocked.

Remove application

To remove application do the following:

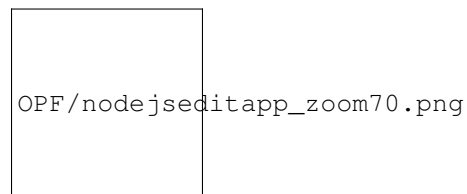
- Click Bin icon in the Actions column in a particular application row.
- In the confirmation pop-up click Agree to start removing or Cancel to close pop-up.
- When an action is completed an application will be removed from the Web Applications table and a confirmation pop-up will be displayed.



Edit application

To edit application do the following:

- Click Pencil icon in the Actions column in a particular application row. A particular application tab opens.



The following actions are available:

- Restart application — click Restart button.
- Stop Node.js — click Stop Node.js button.
- Run JavaScript script — click Run JS Script button to run a command specified in the Scripts section of the package.json file. Specify the name of the script to run plus any parameters then click Ok.

- Remove application — click Delete button and confirm the action in a pop-up.
- Change Node.js version — choose Node.js version from a drop-down.
- Change Application mode — choose application mode from a drop-down. Available modes are Production and Development.
- Specify Application root — specify in a field a physical address to the application on a server that corresponds with its URI.
- Specify Application URL — specify in a field an HTTP/HTTPS link to the application.
- Specify Application startup file — specify as NAME.js file.
- Run npm install command — click Run npm install button to install the package(s) described in the package.json file.
- Add Environment variables — click Add Variable and specify a name and a value.

Node.js Deployment

The first approach - *remote usage of Node.js Interpreters of different versions*.

The second approach - *remote usage of thecloudlinux-selector utility*.

Remote Usage of Node.js Interpreters

1. Create a Node.js project in IntelliJ IDEA/WebStorm. You can download [this archive](#) and use it as a basis.
2. Install alt-nodejs packages on the server in use. See *installation instructions*.

‘<>’_3. Create an application on the server. You can do it by three ways:

oVia UI of the Node.js plugin.

oUsing the following command to create an application:

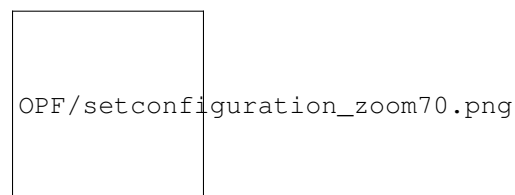
```
cloudlinux-selector create --interpret=nodejs --json --app-root=<USER_NAME> --app-uri=<APP_NAME> --app-mode=development --version=<VERSION> --domain=<DOMAIN>
```

Note. In the IntelliJ IDEA you can create and run any remote script (Preferences — Remote SSH External Tools — Add).



oChoose a location of the application on the server and synchronize the files with the IntelliJ IDEA project.

‘<>’_4. Set up Run/Debug Configurations in the project created.



oSpecify a path to the remote Node.js interpreter. To be able to specify the remote interpreter, you should install the Node.js Remote Interpreter plugin first. Please find more information [here](#), using server access credentials for a user (Main menu — Run — Edit configurations...).

oSpecify initial JavaScript file that will be run with the node command (it is the app.js file from the archive).

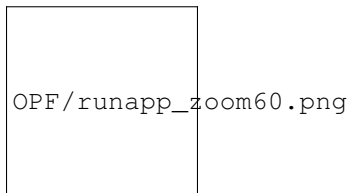
oSpecify Path Mappings between a local and a remote project (Preferences — Deployments — Add). If you have created your application with the cloudlinux-selector utility or via plugin UI the Path Mappings should be as follows:

```
/home/<USER_NAME>/<APP_NAME>
```

5. Synchronize the project directories on the local and the remote machine as per Path Mappings specified.

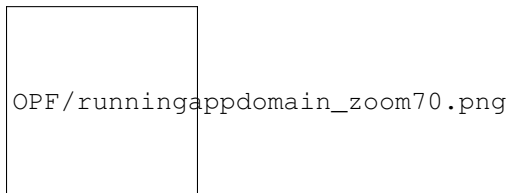
6. Deploy the modules on the remote and the local machine with the npm install command (if there are dependent modules). In the UI you can click the Run NPM Install button.

7. Run Node.js application with the configuration set at the *4th step* (Main menu — Run — Run... — Select configuration).



8. If you are using the application from the archive attached, you can see the running application on the 3003 port — <http://DOMAIN:3003>.

Note. The port should be available to a server user.



The following information should be displayed on this page:

- A version of the running Node.js interpreter;
- Current environment variables;
- A current time.

So that, you can be sure that deployed modules are used properly.

If you'd like to use a different version of Node.js to run an application, change a path to the interpreter in the configuration settings of the running.

To apply all changes to the project, synchronize all changes with the server and restart the running application.

9. To debug a script, set breakpoints in the code and run the configuration via Main Menu (Main menu — Run — Debug... — Select configuration).

Useful links:

- IntelliJ IDEA: <https://www.jetbrains.com/help/idea/configure-node-js-remote-interpreter.html>
- Plugin Node.js Remote Interpreter: <https://plugins.jetbrains.com/plugin/8116-node-js-remote-interpreter>
- WebStorm: <https://www.jetbrains.com/help/webstorm/configure-node-js-remote-interpreter.html>

Note. It is not required to install Passenger while working in IDE if you are using this approach.

Remote Usage of the cloudlinux-selector Utility

1. Create an application via UI or with the command as described in the Remote Usage of Node.js Interpreters approach, *step 3 (a,b)*.

2. Set up project mapping on the local machine with the created remote application `/home/<USER_NAME>/<APP_NAME>` (Preferences → Deployments → Add).

3. Set up the remote commands of the cloudlinux-selector (Preferences → Remote SSH External Tools → Add) for the following actions:

oRestart application;

oInstall packages;

oRun script;

oChange Node.js version for the application.

You can see the running app at http://DOMAIN/APPLICATION_URL

To apply all changes, restart the application.

inodes Limits

[cPanel Only]

LVE Manager inodes limits extension allows setting inode limits for the 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.

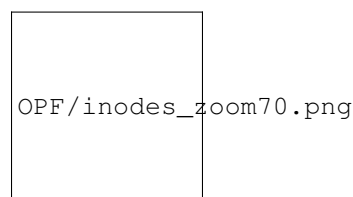
LVE Manager allows to set soft and hard IO limit.

- Hard limit prevents a 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`.

Note that we do not collect statistical information on the inodes like we do for other LVE limits.

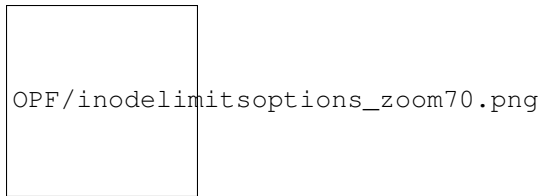
You can set inodes limits using LVE Manager, the same way you would set any other LVE Limits:



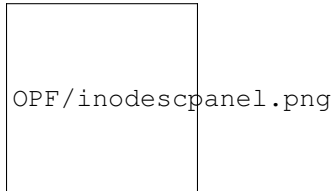
The limits can be set on the level of individual account or package:



Sometimes disk quota breaks, so do inodes limits. You can reset them through the Options tab of LVE Manager:



End users can monitor their inodes usage through cPanel:



End user can also see the usage inside resource usage menu.

cl-quota utility

cl-quota utility is designed to control disk quotas and provides:

- Setting user and package limits.
- Integration with panel packages.
- Limits synchronization.
- Automatic inheritance of panel limits to all appropriate users.

Note. cl-quota works only with inodes soft/hard limits (soft/hard file limits in setquota/repquota utilities terminology). Block limits are not controlled by cl-quota utility in any way, they are not taken into account and do not affect the data that they issue. That is why hereinafter it is the inode limits that are implied by the word “limits”.

- General Provisions*
- Setting Limits and Integration with Panel Packages*
- Limits Inheritance*
- Caching and Synchronizing the Limits*
- Quotas DBFile*
- CLI Options/Examples*

General Provisions

cl-quota utility never sets/reads limits directly in the system, it uses standard setquota/repquota utilities included into the quota package for this purpose.

cl-quota will not work in the following cases:

- setquota/repquota are missing or working incorrectly;
- the quotas are not configured in the system.

cl-quota only performs the minimal diagnostics of quota related errors:

- verifies the availability of setquota/repquota utilities on the disk;

- verifies if quotas are activated for a specified user (with a separate command), see below.

quota package which contains the required setquota/repquota utilities, is not included in lvmmanager package dependencies by default, and quotas activation is a long process which sometimes depends on the panel used, therefore, all the steps on quotas configuration and activation must be carried out by yourself, cl-quota does not perform these actions.

Error messages sent back to the console are extremely rare, to receive error messages use the command:

```
# cat /var/log/messages | grep clquota
```

Note that you should not set soft limit higher than hard limit. cl-quota does not control it in any way, but in some cases, the system can ban such limits combination and they won't be set or will be set in some other way.

Setting Limits and Integration with Panel Packages

cl-quota utility allows setting inodes limits for users of the system.

cl-quota integrates with the panels through a standard mechanism - [Integrating LVE Limits with Packages](#).

Panel users are such users whose UIDs are issued by the above panel integration mechanism. The list of panel packages and the information on the user's affiliation to a particular package is obtained from there as well.

When installing/reading the limits, the following peculiarities are applied:

- 1.When displaying quotas, cl-quota displays information about the limits of all users - system and panel. No filter applied. The actual limit values are always displayed.
- 2.Limit value -1 for the packages (see below) is displayed as dash (-).
- 3.If cl-quota is running under root, it will display the limits returned by repquota utility with no changes. If it is running under some other user, it will return data from a special cache file, see "[Quotacache and synchronization](#)".
- 4.Limits setting only works for panel users, for all other users limits are not set (the command is ignored). The only exception - uid=0. The limits are never set for the root user (uid=0), but they are stored in DB file and are used by inheritance mechanism. See "[Limits Inheritance](#)".
- 5.Hard and soft limits are completely independent, l-quota does not apply any interdependencies to them. Setting only one of them (any) is acceptable, the other one will not change.

cl-quota utility also supports package limits set/read. When setting package limits, they are set for all users of a particular package except for those whose limits are set individually. See also "[Limits Inheritance](#)".

If package name is "default", then setting limits command is ignored. If some limits are set for this package in DB file, they will be displayed along with all the others, but will not be used. See also "[Limits inheritance](#)".

Any positive numbers are allowed as limit values. cl-quota neither controls nor changes these values except the following cases:

- negative values are taken modulo;
- fractional values are converted to integers by discarding the fractional part;
- if the transferred value can not be turned into a number (for example, 67wg76), it is completely ignored and the limit is not set at all.

Then these values are transmitted directly to setquota system utility for the actual setting of the limits.

Thus cl-quota has two limit values, which are processed in a special way:

- 0. Means inheritance of the limit from the package where the user is located, or from uid=0. See also "[Limits inheritance](#)" for more detailed information.

- 1. The real limits are set to 0, which means no limits, literally “unlimited”. This is legit both for individual and for package limits. Limit value -1 is stored in the database as well as any other but is never displayed.

You can use the words “default” and “unlimited” instead of 0 and -1 respectively, they are fully interchangeable. See also “*DB File*” and “*CLI Options*”.

Individual and package limits are always saved in DB file `/etc/container/cl-quotas.dat`. Limits from there are used when synchronizing quotas. Please find more details in “*Limits Synchronization*”.

Also, find detailed information on DB file itself in “*QuotasDB File*” section.

Utility options are described in “*CLI Options*” section.

Limits Inheritance

When setting package limits to the package users, the inheritance principle is applied. It means that:

- If no individual limit is set to a user, then he inherits the limits of the package he belongs to.
- If no limit is set to a package (=0), then the users inherit uid=0 limits.

Limits of the package named “default” (if found in the DB file) will always be ignored and all the users of this package will get uid=0 limits.

Caching and Synchronizing the Limits

Any user of the system (including panel users) is always created with limits equal to 0. To assign him the limits of the corresponding package, the synchronization process is used.

During the synchronization, `cl-quota` utility reads the database file and sets the limits from it to the users and packages specified therein.

This mode is designed to set the correct limits for the new users and to restore them for the existing ones. When recovering, the current limits are neither read nor analyzed.

Caching - is writing current limits to `/etc/container/cl-quotas.cache` file. If `cl-quota` is not started from the root for reading the current limits, then it returns data from this file.

When installing LVE Manager package, a special cron job is installed, which performs synchronization and caching (`cl-quota -YC`) every 5 minutes. Therefore, the correct limits will be set for the user within 5 minutes from the moment of its creation.

Caching and synchronization can also be performed separately, see “*CLI Options*” section.

To disable this feature add `cl_quota_limits_autosync=no` to the config file `/etc/sysconfig/cloudlinux`.

Quotas DB File

All `cl-quota` limits settings are stored in `/etc/container/cl-quotas.dat` along with the UID or the name of the package the limit was set for.

When saving the limits to a file, the following rules are applied:

- If a limit value is non-integer or non-numeric, then the rules from “Setting limits and integrating with panel packages” section are applied. The assigned value is saved to the file.
- Limits are always saved in pairs, no matter if only one limit was set or both. The pair looks as follows: `soft_limit:hard_limit`.

- The values 0 and -1, when having a predetermined meaning, are saved as is without any transformations.
- The words “default” and “unlimited” are saved as 0 and -1 respectively.
- If both limits for a user/package were set as 0, then such user/package is not saved in the file, and if it was previously there - it will be removed. Therefore, if a user/package is not mentioned in the file, then all its limits are inherited. See “*Limits Inheritance*” section.

The lists of panel users, packages, and user-package correspondence are not saved anywhere, this information is always subtracted from the panel.

Example:

```
/etc/container/cl-quotas.dat
```

```
[users]
```

```
0 = 1000:2000
```

```
500 = -1:-1
```

```
958 = 0:20000
```

```
[packages]
```

```
pack1 = 5000:-1
```

It follows that:

- uid=0 limits are set to 1000:2000 - all users in the default package will obtain these limits.
- Both limits are set as unlimited for a user with uid=500, which means that its real limits will always be 0:0. The package limits do not affect this user.
- Soft limit of the user with uid=958 is inherited (0 means inheritance), his hard limit is set to 20000 and it will not depend on the package limits or uid=0 limits.
- Limits 5000:-1 are set for pack1 package, therefore its real limits are: soft_limit=5000 and hard_limit=0 (unlimited).
- The users of pack1 package will get pack1 limits (5000:-1), the users of all the rest of the packages will get the limits of uid=0 because no limits are set for them. Exceptions: uid=500 and 958. uid=500 has both limits set individually, and uid=958 inherits only soft limits.

CLI Options/Examples

cl-quot utility has the following command line options:

```
-u | -user          : specifies the user
-U | -user-id       : specifies the user ID
-S | -soft-limit    : sets the soft limit for a user. Pass 0
```

or ‘default’ to set package default limit. Pass -1 or ‘unlimited’ to cancel limit

```
-H | -hard-limit    : sets the hard limit for a user. Pass 0
```

or ‘default’ to set package default limit. Pass -1 or ‘unlimited’ to cancel limit

```
-V | -csv           : returns data as comma separated values
-p | -package       : specifies a package to set or get limits
-P | -package-limits : prints package limits
-a | -all-package-limits : prints all package limits (including
```

packages without limits)

-Y | -sync : synchronizes packages and users limits

with the database

-C | -cache-content : cache quota data to a file the database

-F | -force : save user quotas even when they are

equal to defaults

-check : check if quotas is

enabled/activated/suported; if disabled show diagnostic information; using with -user or -user-id options

-user and -user-id options are designed to specify user whose limits are required to be set or displayed. -user specifies user name, -user-id - uid. It is acceptable to specify one or another.

-package - specifies package name.

-soft-limit, -hard-limit - specify soft and hard limits values respectively. It is acceptable to use words “default” or “unlimited” as limit value.

-csv - displays limits in csv format (instead of data formatted in the table).

-package-limits - displaying the limits of the packages created by the panel admin.

-all-package-limits - displaying the limits of all the packages, including the ones created by the resellers and packages with no users.

-sync - synchronizes users quotas and packages with the database.

-cache-contents - performs quotas caching.

-force - saving user quotas even if they are equal to the current.

-check - performs diagnostics for a specified user. Can be used only when a user is specified (along with -user / -user-id).

Examples:

1. Reading current user limits:

```
# cl-quota
```

```
# cl-quota -csv
```

2. Reading current package limits:

```
# cl-quota -package-limits
```

```
# cl-quota -all-package-limits
```

```
# cl-quota -package-limits -csv
```

```
# cl-quota -all-package-limits -csv
```

3. Specifying limits for a user:

```
# cl-quota -user-id=500 -soft-limit=0 -hard-limit=0
```

```
# cl-quota -user-id=500 -soft-limit=unlimited
```

```
# cl-quota -user-id=500 -soft-limit=0 -hard-limit=-1
```

```
# cl-quota -user-id=958 -hard-limit=20000 -soft-limit=0 -force
```

4. Specifying limits for a package:

```
# cl-quota --package pack1 --hard-limit=-1 --soft-limit=5000
```

```
# cl-quota --package pack1 --hard-limit=10000
```

```
# cl-quota --package pack1 --soft-limit=default
```

5. User diagnostics (with example output):

```
# cl-quota --user-id=500 --check
```

Quota disabled for user id 500 (home directory /home/cltest1); quotaon: Mountpoint (or device) / not found or has no quota enabled.

6. Synchronizing quotas with caching (executed in cron):

```
# cl-quota -YC
```

Kernel Settings

•Kernel Config Variables

=

•Virtualized /proc filesystem

•SecureLinks

oSymlink Owner Match Protection

oLink Traversal Protection

•ptrace Block

•Xen XVDa detection

•TPE Extension

•IO Limits latency

•Hybrid Kernel

•Reading LVE usage

•flashcache

•OOM Killer for LVE Processes

•File System Quotas

Kernel Config Variables

Starting from lve-manager 4.0-25.5, lve-utils 3.0-21.2, and cagefs-6.1-26, CloudLinux OS utilities can read/write kernel config variables from a custom config /etc/sysctl.d/90-cloudlinux.conf (earlier, the parameters were read/written only from sysctl.conf).

CloudLinux OS utilities get parameter by using sysctl system utility. So for now, even if a config variable is not set in the sysctl.conf and in the /etc/sysctl.d config files, this variable will be read by sysctl utility directly from /proc/sys.

If some kernel variable was set in /etc/sysctl.d/90-cloudlinux.conf do

```
sysctl --system
```

to apply the parameters before reading and after writing.

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` and *`cldetectutility`* can configure few most commonly used monitoring software automatically.

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
```

Note: starting from `lve-utils` 3.0-21.2, `fs.proc_super_gid` parameter in `da_add_admin` utility is written to `/etc/sysctl.d/90-cloudlinux.conf`.

Remounting procfs with “hidepid” option

In lve-utils-2.1-3.2 and later /proc can be remounted with “hidepid=2” option to enable additional protection for procfs. This remount is performed in lve_namespaces service.

This option is in sync with fs.proc_can_see_other_uid kernel parameter described above.

When /etc/sysctl.conf does not contain fs.proc_can_see_other_uid setting, the protection is off (procfs is remounted with hidepid=0 option). In this case fs.proc_super_gid setting is ignored. Users are able to see full /proc including processes of other users on a server. This is a default behavior.

If /etc/sysctl.conf contains “fs.proc_can_see_other_uid=1” setting, then /proc will be remounted with “hidepid=0” option (disable “hidepid” protection for all users).

If /etc/sysctl.conf contains “fs.proc_can_see_other_uid=0” setting, then /proc will be remounted with “hidepid=2” option (enable “hidepid” protection for all users).

If /etc/sysctl.conf contains “fs.proc_can_see_other_uid=0” and “fs.proc_super_gid=\$GID” settings, then /proc will be remounted with “hidepid=2, gid=\$GID” options (enable “hidepid” for all users except users in group with gid \$GID).

To apply /etc/sysctl.conf changes, you should execute

```
service lve_namespaces restart
```

Or

```
/usr/share/cloudlinux/remount_proc.py
```

So, admin can prevent users from seeing processes of other users via “fs.proc_can_see_other_uid” and “fs.proc_super_gid” settings in /etc/sysctl.conf, like earlier.

Also, you can override this by specifying desired options for /proc in /etc/fstab.

To disable hidepid, add to /etc/fstab the following:

```
proc /proc proc defaults,hidepid=0,gid=0 0 0
```

Or you can specify desired hidepid and gid values explicitly:

```
proc /proc proc defaults,hidepid=2,gid=clsupergid 0 0
```

You should execute

```
mount -o remount /proc
```

to apply /etc/fstab changes.

But we recommend to manage procfs mount options via /etc/sysctl.conf as described above for backward compatibility.

Note: there is a known issue on CloudLinux 6 systems. User cannot see full /proc inside CageFS even when this user is in “super” group, that should see full /proc. This issue does not affect users with CageFS disabled. CloudLinux 7 is not affected.

Note: starting from lve-utils 3.0-21.2, lve_namespaces service can read parameters from the /etc/sysctl.d/90-cloudlinux.conf.

Even if fs.proc_can_see_other_uid and fs.proc_super_gid parameters are not set in config files but specified in /proc/sys, then when restarting lve_namespaces service the parameters from /proc/sys will be used. So, /proc will be remounted according to these parameters.

SecureLinks

CloudLinux provides comprehensive protection against symbolic link attacks popular in shared hosting environment.

The protection requires setting multiple kernel options to be enabled.

Symlink Owner Match Protection

`fs.enforce_symlinksifowner`

To protect against symlink attack where attacker tricks Apache web server to read some other user PHP config files, or other sensitive file, enable:

`fs.enforce_symlinksifowner=1`.

Setting this option will deny any process running under gid `fs.symlinkown_gid` to follow the symlink if owner of the link doesn't match the owner of the target file.

Defaults:

`fs.enforce_symlinksifowner = 1`

`fs.symlinkown_gid = 48`

<code>fs.enforce_symlinksifowner = 0</code>	do not check symlink ownership
<code>fs.enforce_symlinksifowner = 1</code>	deny if symlink ownership doesn't match target, and process gid matches <code>symlinkown_gid</code>

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.

Please, note that `fs.enforce_symlinksifowner = 2` is deprecated and can cause issues for the system operation.

`fs.symlinkown_gid`

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 the file `/etc/sysctl.conf`, add the line:

`fs.symlinkown_gid = XX`

And execute:

`$ sysctl -p`

To disable symlink owner match protection feature, set `fs.enforce_symlinksifowner = 0` in `/etc/sysctl.conf`, and execute

`$ sysctl -p`

`/proc/sys/fs/global_root_enable` [CloudLinux 7 kernel only] [applicable for kernels 3.10.0-427.36.1.lve1.4.42+]

`proc/sys/fs/global_root_enable` flag enables following the symlink with root ownership. If `global_root_enable=0`, then Symlink Owner Match Protection does not verify the symlink owned by root.

For example, in the path `/proc/self/fd`, `self` is a symlink, which leads to a process directory. The symlink owner is root. When `global_root_enable=0`, Symlink Owner Match Protection excludes this element from the verification. When `global_root_enable=1`, the verification will be performed, which could block the access to `fd` and cause violation of the web-site performance.

It is recommended to set `/proc/sys/fs/global_root_enable=0` by default. If needed, set `/proc/sys/fs/global_root_enable=1` to increase the level of protection.

Note: starting from `lve-utils 3.0-21.2`, `fs.symlinkown_gid` parameter values for `httpd` service user and `fs.proc_super_gid` for `nagios` service user is written to `/etc/sysctl.d/90-cloudlinux.conf`.

Link Traversal Protection

CageFS is extremely powerful at stopping most information disclosure attacks, where a hacker could read sensitive files like `/etc/passwd`.

Yet, CageFS does not work in each and every situation. For example, on cPanel servers, it is not enabled in WebDAV server, cPanel file manager and webmail, as well as some FTP servers don't include proper change rooting.

This allows an attacker to create symlink or hardlink to a sensitive file like `/etc/passwd` and then use WebDAV, filemanager, or webmail to read the content of that file.

Starting with CL6 kernel 2.6.32-604.16.2.lve1.3.45, you can prevent such attacks by preventing user from creating symlinks and hardlinks to files that they don't own.

This is done by set following kernel options to 1:

```
fs.protected_symlinks_create = 1
```

```
fs.protected_hardlinks_create = 1
```

However, we do not recommend to use `protected_symlinks` option for cPanel users as it might break some of the cPanel functionality.

Please, note that Link Traversal Protection is disabled by default for the new CloudLinux OS installations/conversions.

```
fs.protected_symlinks_create = 0
```

```
fs.protected_hardlinks_create = 0
```

Then setup:

```
fs.protected_symlinks_allow_gid = id_of_group_linksafe
```

```
fs.protected_hardlinks_allow_gid = id_of_group_linksafe
```

This is for example needed by PHP Selector to work (new versions of Alt-PHP can already correctly configure those settings).

To manually adjust the settings, edit:

```
/etc/sysctl.d/cloudlinux-linksafe.conf
```

and execute:

```
sysctl -p /etc/sysctl.d/cloudlinux-linksafe.conf
```

or:

```
sysctl -system
```

Note: starting from `lvmanager 4.0-25.5`, if there is no `/etc/sysctl.d/cloudlinux-linksafe.conf` config file, `selectorctl` for PHP with `--setup-without-cagefs` and `--revert-to-cagefs` keys writes `fs.protected_symlinks_create` and `fs.protected_hardlinks_create` parameters to `/etc/sysctl.d/90-cloudlinux.conf`.

ptrace Block

Starting with kernel 3.10.0-427.18.s2.lve1.4.21 (CloudLinux 7) and 2.6.32-673.26.1.lve1.4.17 (CloudLinux 6) we re-implemented `ptrace` block to protect against `ptrace` family of vulnerabilities. It prevents end user from using any `ptrace` related functionality, including such commands as `strace`, `lsof` or `gdb`.

By default, CloudLinux doesn't prevent `ptrace` functionality.

Defaults:

```
kernel.user_ptrace = 1
```

```
kernel.user_ptrace_self = 1
```

The option `kernel.user_ptrace` disables `PTRACE_ATTACH` functionality, option `kernel.user_ptrace_self` disables `PTRACE_TRACEME`.

To disable all ptrace functionality change both `sysctl` options to 0, add this section to `/etc/sysctl.conf`:

```
## CL. Disable ptrace for users
```

```
kernel.user_ptrace = 0
```

```
kernel.user_ptrace_self = 0
```

```
##
```

Apply changes with:

```
$ sysctl -p
```

Different software could need different access to ptrace, you may need to change only one option to 0 to make them working. In this case, there will be only partial ptrace protection.

* ptrace protection is known to break PSA service for Plesk 11

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 CloudLinux 6 and Hybrid kernels.

TPE Extension (deprecated)

[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:

•	<code>/proc/sys/kernel/grsecurity/grsec_lock</code>
•	<code>/proc/sys/kernel/grsecurity/tpe</code>
•	<code>/proc/sys/kernel/grsecurity/tpe_gid</code>

•	/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.

IO Limits 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 milliseconds. 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.

For CloudLinux 6 and CloudLinux 7 (since Hybrid kernel lve1.4.x.el5h):

To enable IO Limits latency and set it to 10 seconds:

```
# echo 10000 > /sys/module/kmodlve/parameters/latency
```

To disable latency:

```
# echo 2000000000 > /sys/module/kmodlve/parameters/latency
```

It is possible to set, for example, 1000 as a permanent value. To do so, create a file `/etc/modprobe.d/kmodlve.conf` with the following content:

```
options kmodlve latency=1000
```

For CloudLinux 5 (OBSOLETE):

To enable IO Limits latency and set it to 10 seconds:

```
# echo 10000 > /sys/module/iolimits/**parameters/latency
```

To disable latency:

```
# echo 2000000000 > /sys/module/iolimits/**parameters/latency
```

Hybrid Kernel

CloudLinux 6 Hybrid kernel

CloudLinux 6 Hybrid Kernel is CloudLinux 7 (3.10.0) kernel compiled for CloudLinux 6 OS. New 3.10 kernel features a set of performance and scalability improvements related to IO, networking and memory management, available in CloudLinux 7 OS. It also features improved CPU scheduler for better overall system throughput and latency.

Please find information on the main features of 3.10 kernel branch on the links:

https://kernelnewbies.org/Linux_3.10#head-e740f930dfd021616cc42e8abf21c79d0b07e217

<https://www.kernel.org/pub/linux/kernel/v3.0/ChangeLog-3.10.1>

How to migrate from the normal to hybrid channel:

Note. The system must be registered in CLN.

1. Update rhn-client-tools from production
2. Run normal-to-hybrid script.
3. Reboot after script execution is completed.

```
yum update rhn-client-tools
```

```
normal-to-hybrid
```

```
reboot
```

How to migrate from hybrid to the normal channel:

Note. The system should be registered in CLN.

1. Run hybrid-to-normal script.
2. Reboot after script execution is completed.

```
hybrid-to-normal
```

```
reboot
```

```
z
```

Known limitations and issues:

1. We do not remove Hybrid kernel after migration from Hybrid to the normal channel, but we remove linux-firmware package which is needed to boot Hybrid kernel. This is because CloudLinux 6 does not allow to remove the package of currently running kernel. Proper removal procedure will be implemented, but for now, we should warn users not to boot Hybrid kernel if they have migrated to normal channel.
2. Kernel module signature isn't checking for now, as 3.10 kernel is using x509 certificates to generate keys and CL6 cannot detect signatures created in such way. The solution will be implemented.

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	ICPU	IIO	CPU	MEM	IO	IMEM	IEP	nCPU	fMEM	fEP
IMEMPHY		ICPUW	INPROC	MEMPHY		fMEMPHY	NPROC	fNPROC			
0	0	25	1024	0	0	0	262144	20	1	0	0
262144											
0	0										
300	0	25	1024	1862407	0	0	262144	20	1	0	0
31	0	0	0								

Version 4 (CL 5 kernel):

4:LVE	EP	ICPU	IIO	CPU	MEM	IO	IMEM	IEP	nCPU	fMEM	fEP
0	0	25	25	0	0	0	262144	20	1	0	0
262144											
300	0	25	25	15103019	0	0	262144	20	1	0	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	
fMEM	Virtual memory faults	number of faults	
fEP	Entry Processes faults	number of faults	v6+
IMEM-PHY	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+
fMEM-PHY	Physical memory faults	number of faults	v6+
NPROC	Number of processes	number	v6+
fNPROC	Number of processes faults	number of faults	v6+
IOPS	IO operations since reboot	number	v8+

flashcache

* Available only for x86_64, CloudLinux 6 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 CloudLinux 6 & 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>

OOM Killer for LVE Processes

When LVE reaches its memory limit, the processes inside that LVE are killed by OOM Killer and appropriate message is written to /var/log/messages. When any LVE hits huge number of memory limits in short period of time, then OOM Killer could cause system overload. Starting from kernel 2.6.32-673.26.1.lve1.4.15 (CloudLinux 6) and from kernel

3.10.0-427.18.2.lve1.4.14 (CloudLinux 7) heavy OOM Killer could be disabled. If so - lightweight SIGKILL will be used instead.

By default OOM Killer is enabled, to disable it please run:

For CloudLinux 6:

```
# echo 1 > /proc/sys/ubc/ubc_oom_disable
```

Also, add the following to /etc/sysctl.conf file to apply the same during boot:

```
ubc.ubc_oom_disable=1
```

For CloudLinux 7:

```
# echo 1 > /proc/sys/kernel/memcg_oom_disable
```

Also, add the following to /etc/sysctl.conf file to apply the same during boot:

```
kernel.memcg_oom_disable=1
```

File System Quotas

In Ext4 file system, the process with enabled capability CAP_SYS_RESOURCE is not checked on the quota exceeding by default. It allows userland utilities selectorctl and cagefs to operate without fails even if a user exceeds a quota.

To disable quota checking in XFS file system set cap_res_quota_disable option to 1 using the following command:

```
# echo 1 > /proc/sys/fs/xfs/cap_res_quota_disable
```

Apache mod_lsapi

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.

How it works

- mod_lsapi is a part of Apache;
- Apache passes handling for PHP request to mod_lsapi;
- mod_lsapi uses liblsapi to transfers request lsphp daemon;
- lsphp processes request and return data to mod_lsapi;
- each user has lsphp processes in separate CageFS/LVE;
- If there is no requests for lsapi_backend_pgrp_max_idle seconds, lsphp process is terminated;
- If no lsphp processes available when new request comes, new lsphp process is created;
- lsphp can process lsapi_backend_children requests simultaneously.

!mod_lsapidigrammNEW!

What is lsphp

lsphp - PHP + LSAPI. What is LSAPI? LiteSpeed Server Application Programming Interface (LSAPI) is designed specifically for seamless, optimized communication between LiteSpeed Web Server and third party web applications. Now this protocol is available for Apache 2.2/2.4.

Using LSAPI, we have seen higher performance than Apache with mod_php, easier installation than php-fpm and easier integration with any control panel. LSAPI means faster and more stable dynamic web pages.

Requirements

- CageFS (installed and initialized) - optional, mod_lsapi can work without CageFS;
- Alt-PHP or ea-php for EasyApache 4;
- Apache with SuExecUserGroup directive for each user's VirtualHost;
- mod_ruid2 disabled;
- apache itk disabled.

Configuration Options

Options	Description	Level
php_value, php_admin_value, php_flag, php_admin_flag	mod_php emulation	httpd.conf, virtualhost, htaccess
lsapi_engine	Switching mod_lsapi handler on or off	httpd.conf
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_backend_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 # Maximum number of simultaneously running child backend processes. # Optional, a default value is equal to EP. # min value is 2; max value is 10000. If var value is more, 10000 will be used.	httpd.conf
lsapi_backend_max_process_time	sets env variable LSAPI_MAX_PROCESS_TIME # Optional. Default value is 300. # Timeout to kill runaway processes	httpd.conf
lsapi_backend_pgrp_max_idle	sets env variable LSAPI_PGRP_MAX_IDLE, in seconds. # Controls how long a control process will wait for a new request before it exits. # 0 stands for infinite. # Optional, default value is 30. # Should be more or equal to 0.	httpd.conf
lsapi_debug	enable debugging for mod_lsapi, acceptable values: on/off	httpd.conf

Continued on next page

Table 1 – continued from previous page

lsapi_socket_path	Path to back end lsphp sockets. By default /var/run/mod_lsapi	httpd.conf
lsapi_per_user	Invoke master lsPHP process not per VirtualHost but per account	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_poll_timeout	Time to wait for response from the lsphp daemon, in seconds. 0 stands for infinity. For preventing long running processes which can use EP (limit number of entry processes). Default value is 300.	httpd.conf
lsapi_backend_coredump	env variable LSAPI_ALLOW_CORE_DUMP (On or Off). Pass LSAPI_ALLOW_CORE_DUMP to lsphp or not. If it will be passed - core dump on lsphp crash will be created. # Off by default # 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. LSAPI_ALLOW_CORE_DUMP	httpd.conf
lsapi_mod_php_behavior	On/Off - disable php_* directives, default On.	httpd.conf, virtualhost, htaccess
lsapi_with_connection_pool	On/Off - disable enable connect pool, default Off	httpd.conf
lsapi_backend_max_idle	It is relevant only with lsapi_with_connection_pool option switched on. Controls how long a worker process will wait for a new request before it exits.	httpd.conf

Continued on next page

Table 1 – continued from previous page

lsapi_backend_max_reqs	It is relevant only with lsapi_with_connection_pool option switched on. Controls how many requests a worker process will process before it exits.	httpd.conf
lsapi_set_env	Pass env variable to lsphp. By default lsphp environment have only TEMP, TMP and PATH variables set. Example: lsapi_set_env TMP "/var/lsphp-tmp" Note: PATH env var default "/usr/local/bin:/usr/bin:/bin" cannot be changed because of security reasons. To change it, use explicitly lsapi_set_env_path option	httpd.conf
lsapi_set_env_path	Change PATH variable in the environment of lsPHP processes.	httpd.conf
lsapi_paranoid	Check or not permissions of target php scripts	httpd.conf
lsapi_check_document_root	Check or not the owner of DOCUMENT ROOT	httpd.conf
lsapi_enable_user_ini	Enable .user.ini files for backend. Same as suphp, php-fpm and fcgid mechanism of .user.ini. Default value is Off	httpd.conf, virtualhost
lsapi_user_ini_homedir	On/Off. If lsapi_enable_user_ini option is set to On, then enable/disable processing .user.ini file in home directory also. Default value is Off	httpd.conf, virtualhost
lsapi_criu	Enable/disable CRIU for lsphp freezing. Can be: On/Off. Default: Off	httpd.conf
lsapi_criu_socket_path	Set path to socket for communication with criu service [should be path] - default: /var/run/criu/criu_service.socket	httpd.conf
lsapi_backend_semtimedwait	Enable/disable flag for notification about lsphp started. This method avoid cycles of waiting for lsphp start/ Can be: On/Off. Default: On	httpd.conf
lsapi_backend_initial_start	Number of request when lsphp should be freezed. Should be [number] - default 0	httpd.conf
lsapi_criu_use_shm	Method of requests counting. Off - use shared memory. Signals - use signals from child processes to parent. Default: Off	httpd.conf

Continued on next page

Table 1 – continued from previous page

lsapi_criu_imgs_dir_path	Path to folder where imgs of freezed PHP will be stored. Should be path. Default: /var/run/mod_lsapi/	httpd.conf
lsapi_output_buffering	Disabling HTTP responses buffering on Apache level. On - enable buffering. Off - disable buffering	httpd.conf, virtualhost, htaccess
lsapi_backend_common_own_log	can be used only when lsapi_backend_use_own_log is On. On - backend processes of the all virtual hosts will share the common log file. Off - every virtual host will have its own backend log file.	httpd.conf, virtualhost
lsapi_backend_use_own_log	Redirecting log output of backend processes from Apache error_log to dedicated log file or files, depending on value of lsapi_backend_common_own_log option.	httpd.conf, virtualhost
lsapi_process_phpini	Enable or disable phpini_* directive processing. Default value is Off	httpd.conf, virtualhost, directory
lsapi_phpini	When lsapi_process_phpini option switched to Off, these values will be silently ignored. lsapi_phpini values with absolute filename of php.ini file can be inserted into .htaccess files in order to set custom php.ini which will override/complement settings from system default php.ini.	httpd.conf, virtualhost, directory
lsapi_disable_reject_mode	Acceptable values: on/off: If a new HTTP request is coming to LSPHP daemon when all LSPHP workers are still busy, it can process this situation in two different ways. In REJECT mode LSPHP daemon will reject such request immediately. Otherwise, in legacy mode LSPHP daemon will put this request into infinite queue, until one or more LSPHP daemon becomes free. When HTTP request is rejected in REJECT mode, mod_lsapi will write into Apache error_log the following message: Connect to backend rejected, and client will receive 507 HTTP response. By default LSPHP daemon in CloudLinux uses REJECT mode. It can be switched off with this option.	httpd.conf, virtualhost

Continued on next page

Table 1 – continued from previous page

lsapi_avoid_zombies	Enable or disable a mechanism to avoid creation of zombie processes by lsphp. Default value is Off.	httpd.conf, virtualhost
lsapi_disable_forced_pwd_var	To disable addition of PWD variable. Default value is Off. If set to On, the PWD variable will not be added into a backend environment.	httpd.conf, virtualhost

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

for making security pass PHPRC and UID|GID on start lsphp

Algorithm of creating:

/bin/dd if=/dev/random of=/etc/sysconfig/modlsapi.secret bs=16 count=1

Command line Tools

Use the following syntax to manage MODLSAPI install utility:

/usr/bin/switch_mod_lsapi [OPTIONS]

Options:

<code>--setup</code>	setup mod_lsapi configurations for Apache
<code>--setup-light</code>	only EasyApache 4 feature
<code>--uninstall</code>	uninstall mod_lsapi from Apache
<code>--enable-domain</code>	enable mod_lsapi for individual domain
<code>--disable-domain</code>	disable mod_lsapi for individual domain
<code>--enable-global</code>	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
<code>--disable-global</code>	disable mod_lsapi as a default way to serve PHP, disabling mod_lsapi for all domains, including those selected earlier using <code>--enable-domain</code>
<code>--build-native-lsphp</code>	build native lsphp for cPanel EA3
<code>--build-native-lsphp-cust</code>	build native lsphp for cPanel EA3 (with custom PHP source path)
<code>--check-php</code>	check PHP configuration
<code>--verbose</code>	switch verbose level on
<code>--force</code>	only with setup option (EA4)
<code>--stat</code>	return usage statistics in JSON format; the following statistics metrics are collected: <ul style="list-style-type: none"> •control panel name; •mod_lsapi version; •liblsapi version; •criu version and status; •whether mod_lsapi is enabled; •lsapi configuration options; •number of domains, that use mod_lsapi, per each installed PHP version (only supported for cPanel EA4, Plesk, and DirectAdmin).

This tool:

- Creates native lsphp (if it doesn't exist) by doing: `cp /opt/alt/php56/usr/bin/lsphp /usr/local/bin/`
- Removes config template for mod_ruid2
- Configures Apache handler application/x-httpd-lsphp
- Switches domain to lsphp or enable global lsphp
- For cPanel EA3 can build native lsphp
-

What commands are available for different control panels:

	No Control Panel	cPanel EA3	cPanel EA4	DirectAdmin	Plesk	InterWorx	ISPManager
setup	•	•	•	• (no need in manual calling)	•	•	•
setup-light	•	•	•	•	•	•	•
uninstall	•	•	•	•	•	•	•
enable-domain	•	•	•	•	•	•	•
disable-domain	•	•	•	•	•	•	•
enable-global	•	•	•	+/(custom build)	•	•	•
disable-global	•	•	•	•	•	•	•
build-native-lsphp	•	•	•	+/(custom build)	•	•	•
build-native-lsphp-cust	•	•	•	•	•	•	•
check-php	•	•	•	•	•	•	•
verbose	•	•	•	•	•	•	•
force	•	•	•	•	•	•	•
stat	+(without domain info)	+(without domain info)	+(with domain info)	+(with domain info)	+(with domain info)	+(without domain info)	+(without domain info)

Different PHP versions (without PHP Selector)

mod_lsapi allows to use different handlers for different php versions. For example, a file with extension .php53 can be handled by php5.3 and a file with extension .php55 handled by php5.5 without PHP Selector.

Here is an extra config file which allows to set handlers and php binaries for these handlers - /etc/container/php.handler. Example of this file:

```
# cat /etc/container/php.handler
application/x-lsphp53 /opt/alt/php53/usr/bin/lsp
application/x-lsphp55 /opt/alt/php55/usr/bin/lsp
```

Default handler for lsphp is - application/x-httpd-lsphp, if I set in .htaccess such options:

```
<FilesMatch “.(php4|php5|php3|php2|php|html)$”>
```

```
SetHandler application/x-httpd-lsphp
```

```
</FilesMatch>
```

```
<FilesMatch ".php53$">
```

```
SetHandler application/x-lsphp53
```

```
</FilesMatch>
```

File index.php53 will be processed by php 5.3, but index.php processed by php standard, placed at /usr/local/bin/lspshp.

Important:

All custom PHP for phpperdir mechanizm should be located in any place in the directory /opt/alt/, because before start lspshp mod_lsapi checks as follows: /usr/local/bin/lspshp or /opt/alt/*/lspshp. Such location and binary file are allowed to execute. Use the folder /opt/alt/[any path] for installing custom php.

For example:

/opt/alt/php.perdir/php55/bin/lspshp - it will work with mod_lsapi.

But if the server has custom php in another location (for example /usr/local/php55/bin/lspshp), then just make symlink to lspshp:

```
ln -sf /usr/local/php55/bin/lspshp /opt/alt/php.perdir/php55/bin/lspshp
```

and add it to php.handler:

```
myhandler-php55 /opt/alt/php.perdir/php55/bin/lspshp
```

mod_lsapi as suPHP replacement (cPanel EasyApache 3 only)

mod_lsapi is a drop in replacement for suPHP. No configuration changes required. To switch from suPHP to mod_lsapi:

Switch the whole server (disables suPHP, all domains will be serviced by mod_lsapi):

```
/usr/bin/switch_mod_lsapi --enable-global
```

Switch individual domains:

```
/usr/bin/switch_mod_lsapi --enable-domain test.example.tst - enablesmod_lsapi [only for domain test.example.tst]
```

Manually add mod_lsapi for a particular domain: - add to .htaccess file for the domain:

```
AddType application/x-httpd-lsphp .php5 .php4 .php .php3 .php2 .phtml
```

Note: This will work only after /usr/bin/switch_mod_lsapi --setup had been called.

Installation

For all control panels - SuExecUserGroup should be present for each virtual host.

CageFS and PHP Selector will be installed by dependencies (for lspshp binaries).

Installing on cPanel servers

```
$ yum install liblsapi liblsapi-devel
```

```
$ yum install mod_lsapi
```

If CageFS is not initialized:

```
$ cagefsctl --init
$ cagefsctl --enable-all
```

```
$ /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 cPanel servers with EasyApache 4

How to convert EasyApache 4 for CloudLinux:

<https://www.cloudlinux.com/blog/entry/beta-easyapache-4-released-for-cloudlinux>

```
$ yum install liblsapi liblsapi-devel
$ yum install ea-apache24-mod_lsapi
```

Alternatively you can install mod_lsapi through EasyApache 4 web interface, just set “install” of ea-apache24-mod_lsapi in the list of available modules.

If CageFS is not initialized:

```
$ cagefsctl --init
$ cagefsctl --enable-all
$ /usr/bin/switch_mod_lsapi --setup
```

Updating mod_lsapi on cPanel servers with EasyApache 4

After updating ea-apache24-mod_lsapi all the domains are switched to the default handler and to turn on mod_lsapi back, it was necessary to enable lsapi handler through MultiPHP Manager.

We noticed that it is not very convenient to enable lsapi handler through MultiPHP Manager after update and automated this process.

So, if you update ea-apache24-mod_lsapi from stable or ea-apache24-mod_lsapi-1.1-9 or lower from beta, after the update you need to run /usr/bin/switch_mod_lsapi --setup to add lsapi handler to MultiPHP Manager.

After this, you will be asked to enable lsapi handler for proper PHP versions, depending on how you used mod_lsapi before (--enable-global, --enable-domain), and then restart Apache.

Please note that the following options were disabled for ea-apache24-mod_lsapi:

```
/usr/bin/switch_mod_lsapi --enable-domain
/usr/bin/switch_mod_lsapi --disable-domain
```

You can manage your domains with PHP version and lsapi handler from MultiPHP Manager.

Please note that lsapi PHP handler is only available for beta version.

Example 1:

1. ea-apache24-mod_lsapi-1.0-30 was installed and globally enabled.
2. The command `yum update ea-apache24-mod_lsapi --enablerepo=cloudlinux-updates-testing --enablerepo=cl-ea4-testing` was executed.

3. While `switch_mod_lsapi --setup` is not called, `mod_lsapi` will work as before.

4. `switch_mod_lsapi --setup` will return:

Instruction: http://docs.cloudlinux.com/index.html?apache_mod_lsapi.html

patching file `apache.pm`

Patch was applied correctly. . .

Added hook for `System::upcp` to hooks registry

`mod_lsapi` switched to turning on and off through the MultiPHP Manager(/Home/Software/MultiPHP Manager)

You are using enabled globally `mod_lsapi`. Do you want to enable `mod_lsapi` through MultiPHP Manager?

Current PHP will be switched to `lsapi` handler:

`ea-php53` SAPI: `suphp`

`ea-php54` SAPI: `suphp`

`ea-php55` SAPI: `cgi`

`ea-php56` SAPI: `suphp`

`ea-php70` SAPI: `cgi`

`ea-php71` SAPI: `suphp`

If you type `no` then `mod_lsapi` will be disabled and you can enable it again from MultiPHP Manager.

Do you want to proceed? [y/N]

5. If `N` is chosen, then `mod_lsapi` moves to the new type of integration with cPanel and restores files `php.conf` and `suphp.conf`. `Mod_lsapi` will be disabled.

6. If `Y` is chosen, then all installed versions will move to `lsapi` handler.

Setting `ea-php53` to `lsapi` handler. . .

Setting `ea-php54` to `lsapi` handler. . .

Setting `ea-php55` to `lsapi` handler. . .

Setting `ea-php56` to `lsapi` handler. . .

Setting `ea-php70` to `lsapi` handler. . .

Setting `ea-php71` to `lsapi` handler. . .

`!mod\lsapi\handler!`

Example 2:

1. `ea-apache24-mod_lsapi-1.0-30` was installed and enabled only for one domain but all other domains have the same `ea-php56` version.

2. The command `yum update ea-apache24-mod_lsapi --enablerepo=cloudlinux-updates-testing --enablerepo=cl-ea4-testing` was executed.

3. While `switch_mod_lsapi --setup` is not called, `mod_lsapi` will work as before.

4. `switch_mod_lsapi --setup` will return:

Instruction: http://docs.cloudlinux.com/index.html?apache_mod_lsapi.html

patching file `apache.pm`

Patch was applied correctly. . .

Added hook for System::upcp to hooks registry

Domains that handled by ea-php56:

tstdomain01.com - lsapi

tstdomain02.com - suphp

There are domains which are using mod_lsapi through `--enable-domain` option.

This option is deprecated for EA4 and mod_lsapi switched to turning on and off through the MultiPHP Manager(/Home/Software/MultiPHP Manager)

Do you want to enable mod_lsapi through MultiPHP Manager for ea-php56?

Domains which are using suphp will be switched to lsapi handler too.

If you type N then mod_lsapi will remain enabled on these domains.

However, enabling mod_lsapi for new domains is now possible only through MultiPHP Manager.

Do you want to proceed? [y/N] y

Setting ea-php56 to lsapi handler. . .

Built /etc/apache2/conf/httpd.conf OK

Reconfiguration completed

5. If N is chosen, then mod_lsapi will move to the new type of integration with cPanel and will restore files `php.conf` and `suphp.conf`. Mod_lsapi still will be enabled for domains like in example `tstdomain01.com` throw `.htaccess` file.

6. If Y is chosen, then displayed PHP version will move to lsapi handler. According to the example, `tstdomain01.com` `tstdomain02.com` using `ea-php56`, will be switched to lsapi handler.

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
```

```
$ yum install mod_lsapi
```

```
$ /usr/bin/switch_mod_lsapi --setup
```

Uncomment string `LoadModule lsapi_module modules/mod_lsapi.so` from the file `/etc/httpd/conf.d/lsapi.conf`

Disable PHP support for needed domain (this action comment out `AddHandler` or `AddType` for `VirtualHost`) or for all domains.

Remove the following strings from `/etc/httpd/conf/httpd.conf`:

```
<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
```

```
$ yum install mod_lsapi
```

```
$ /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/lsapi.conf and restart Apache.

```
$ service httpd restart
```

Building from source

Follow these steps to install lsphp binaries needed for mod_lsapi:

```
$ yum install cagefs lvemanager cmake gcc httpd-devel apr-devel
```

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

```
$ cagefsctl --init
```

```
$ cagefsctl --enable-all
```

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
```

```
$ cd ~
```

```
$ wget http://repo.cloudlinux.com/cloudlinux/sources/da/mod\_lsapi.tar.gz
```

```
$ tar zxvf mod_lsapi.tar.gz
```

```
$ cd mod_lsapi-0.2-7
```

```
$ cmake .
```

```
$ make
```

```
$ make install
```

This will:

- Install: /usr/lib/apache/mod_lsapi.so (or to another correct httpd modules path)
- Install: /usr/sbin/suexec

```
$ cp conf/mod_lsapi.conf /etc/httpd/conf/extra/ #(or another httpd conf directory)
```

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.

```
$ install_da_cb_install
```

For last preparation of CageFS and PHP Selector should be created by script new directory /tmp/lshttpd

```
$ service httpd restart
```

Additional notes on native PHP installation (EasyApache 3 only)

Native PHP - PHP installed and used before alt-php packages were installed. Usually lsphp binary is not available on the servers without LiteSpeed, which means that it should be created (build from php sources with such options as usual php binary file but with LSAPI protocol built-in).

There are two ways to make native lsphp:

1. The quick one (supports all type of panels).

Native lsphp is made from alt-php56:

```
switch_mod_lsapi --setup
```

```
cp /opt/alt/php56/usr/bin/lsphp /usr/local/bin/
```

- 2) The slow one: to detect version of native PHP and build needed sources according to installed PHP (only for cPanel).

```
switch_mod_lsapi --build-native-lsphp
```

3. DirectAdmin has its own native lsphp builder:

```
/usr/local/directadmin/custombuild/build set php1_mode lsphp
```

```
/usr/local/directadmin/custombuild/build php n
```

Uninstall

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
```

Troubleshooting

Debugging mod_lsapi issues: error_log & sulsphp_log

mod_lsapi errors will be located in error_log and sulsphp_log.

Note that errors can appear in both logs at the same time, and you might need to refer to both of them to solve the issue.

See next table for more details:

error_log	sulspg_log	Solution
Could not connect to lsphp backend: connect to lsphp failed: 111 Connection refused. Increase memory limit for LVE ID	uid: (xxx/xxxxxxxx) gid: (xxx/xxxxxxxx) cmd: /usr/local/bin/lsphp	Increase pmem or vmem limits for the user uid.
Error sending request: ReceiveLSHeader: nothing to read from backend socket	No need to check this log.	lsphp was killed. It can be due to apache restart or lfd. If you see this message too often - change lsapi_terminate_backend ds_on_exit to Off in lsapi.conf or add to /etc/csf/csf.pignore the following lines: exe:/usr/local/bin/lsphp pexe:/opt/alt/php.* /usr /bin/lsphp
Error sending request (lsphp is killed?): ReceiveLSHeader: nothing to read from backend socket, referer: http://XXXXXXX Child process with pid: XXXXX was killed by signal: 11, core dump: 0	No need to check this log.	lsphp has crashed. Next slide will explain what to do (core dump creating). Also, check configuration options for apc and suhosin in php.ini. Once you have a core file generated at DocumentRoot contact https://helpdesk.cloudlinux.com/ so we can investigate the cause.
Could not connect to lsphp backend: connect to lsphp failed: 111 Connection refused	file is writable by others: (///usr/local/bin/lsphp)	Incorrect lsphp file permissions. For fixing: chmod 755 /usr/local/bin/lsphp cagefsctl -force-update.
Could not determine uid/gid for request	No need to check this log.	UID/GID are not set in virtual-host. Set lsapi_use_default_uid On in lsapi.conf (it is On by default since 0.1-98 version, this solution is for older versions).
Own id for script file (/xxxx/xxx/xxx) is xxx; should be xxxx	No need to check this log.	File is not owned by the user PHP executed by. To overwrite (insecure), set lsapi_target_perm Off in lsapi.conf.
Could not connect to lsphp backend: connect to lsphp failed: 111 Connection refused	Entering jail error	Check if ageFS enabled. Try running cagefsctl -remount-all.
connect_lsphp: connect to lsphp failed: tries XXX exceeded with timeout XXXXX Could not connect to lsphp backend: connect to lsphp failed: 111 Connection refused	uid: (xxx/xxxxxxxx) gid: (xxx/xxxxxxxx) cmd: /usr/local/bin/lsphp	Check if /tmp/lshttpd (global /tmp is not inside CageFS) exists and owner should be apache: apache for DirectAdmin, Plesk, iWorx, ISPManager and nobody for cPanel.
Backend error on sending request(GET /XXXX HTTP/1.1); uri(/XXXX) content-length(0) (lsphp is killed?): ReceiveAckHdr: backend process reset connection: errno 104 (possibly memory limit for LVE ID XXXX too small)	uid: (xxx/xxxxxxxx) gid: (xxx/xxxxxxxx) cmd: /usr/local/bin/lsphp	Increase PMEM limits for the user UID.
Reached max children process limit: XX, extra: 0, current: XX, please increase LSAPI_CHILDREN. Backend error on sending request(GET /XXXX HTTP/1.1); uri(/XXXX) content-length(0) (lsphp is killed?): ReceiveAckHdr: backend process reset connection: errno 104 (possibly memory limit for LVE ID XXXX too small)	uid: (xxx/xxxxxxxx) gid: (xxx/xxxxxxxx) cmd: /usr/local/bin/lsphp	Increase value of lsapi_backend_children for UID in vhost.conf or globally in lsapi.conf.
fork() failed, please increase process limit: Cannot allocate memory Backend error on sending request(GET /XXXX HTTP/1.1); uri(/XXXX) content-	uid:(xxx); gid:(xxx); uid limit warning: EP should be < than NPROC, current EP:	Chapter 13: End User Increase NPROC limits for the UID. It should be greater than EP and lsapi_backend_children.

Non-standard apache user

If apache runs under a username other than “apache” or “nobody”, you should rebuild sulsphp (where username is built in for security reasons) with corresponding username:

```
$ yum install liblsapi liblsapi-devel
$ 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 -DHTTPD_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/sulsphp

lsphp started under user apache/nobody

Check if SuExecUserGroup specified for virtual hosts. This parameter is used by mod_lsapi for user identification.

Could not connect to lspan backend: connect(/tmp/lshttpd/lsapi_application-x-httpd-lspan_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 lspan, then run:

```
yum install lve-utils
```

Then restart cron service.

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 sulsphp:

```
$ yum install liblsapi liblsapi-devel
$ 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/sulsphp
```

Apache binary called not httpd (httpd.event, httpd.worker)

```
$ yum install liblsapi liblsapi-devel
```

```
$ 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/su2php

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.

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
```

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

Messages appearing in error_log: Child process with pid: XXXXX was killed by signal: 11, core dump: 0

This means that lsphp was crashed. Solution:

- Check if apc for user enabled. Tune its options as described in previous slide.
- Check if suhosin is enabled for user. Tune its options as described in this article.
- If previous items do not help, contact us at <https://helpdesk.cloudlinux.com/>

How to get lsphp core dump on crash

- Configure mod_lsapi to allow lsphp to generate core dumps. In mod_lsapi.conf:

lsapi_backend_coredump On

- Enable core file generation in sysctl:

```
sysctl -w 'kernel.core_uses_pid=1'
```

```
sysctl -w 'kernel.core_pattern=core.%p'
```

Configure system to change max size of core files. In /etc/security/limits.conf add:

```
user1 soft core unlimited
```

```
user1 hard core unlimited
```

where user1 is the username for which lsphp crashes.

- If /etc/profile.d/limits.sh exists, look up for the following lines:

```
if [ "$LIMITUSER" != "root" ]; then
```

```
    ulimit -n 100 -u 35 -m 200000 -d 200000 -s 8192 -c 200000 -v
```

```
unlimited 2>/dev/null
```

Substring "-c 200000" must be replaced with "-c unlimited".

- Add line ulimit -c unlimited into apachectl script just after another invokes of the ulimit command.

- Do cold restart of Apache with the command like this:

```
service httpd stop; sleep 2; killall lsphp; service httpd start
```

- You can make sure that ulimit for lsphp is changed to unlimited successfully with the following command:

```
cat /proc/PID/limits | grep 'Max core file size'
```

where PID is a pid of any lsphp process. `ps -u user1 | grep lsphp`

- Core dump of lsphp will be created in the DocumentRoot of the corresponding virtual server.

On cPanel server it should map to /home/user1/public_html.

mod_lsapi is not included in output of `httpd -M` after installation and setup command for cPanel EasyApache 3

1. Check if the file /usr/local/apache/conf/conf.d/lsapi.conf exists and not empty;

2. Check if output of the command

```
cat /usr/local/apache/conf/httpd.conf | grep "/usr/local/apache/conf/conf.d/*.conf"
```

is not empty.

If it is empty:

1. Add to "include" section of /var/cpanel/conf/apache/main string:

```
"include": "'/usr/local/apache/conf/conf.d/*.conf'"
```

```
“include”:  
  “directive”: ‘include’  
  “items”:  
...  
  -  
  “include”: “’/usr/local/apache/conf/conf.d/*.conf”  
“listen”:  
2. Do:  
mkdir      -p      /usr/local/apache/conf/conf.d/;          cp      /usr/share/lve/modlscapi/confs/lscapi.conf  
/usr/local/apache/conf/conf.d/lscapi.conf  
3. Call:  
/scripts/rebuildhttpdconf  
/scripts/restartsrv_httpd
```

FAQ on mod_lsapi

Q: Is it compatible with EasyApache?

A: Yes, it is. EasyApache works/fully integrates with mod_lsapi.

Q: Is it compatible with PHP Selector?

A: Yes.

Q: Are .htaccess PHP directives supported? For example, mod_php like directives?

A: Yes. mod_lsapi can read php_* and php_admin_* directives.

Q: I have httpd.conf with SuExecUserGroup options. Do I need to add mod_lsapi related options for VirtualHost?

A: No need to change httpd.conf. mod_lsapi can read suPHP_UserGroup, RUIDGid, SuExecUserGroup, AssignUserID parameters to determine user id under which site is running. Additionally you can use lsapi_uid_gid or lsapi_user_group as a native way to specify user / group ids.

Q: What is the difference between running mod_lsapi with lsapi_with_connection_pool mode On and Off?

A: When lsapi_with_connection_pool mode is Off, then the new backend lsphp process has to be created for each new incoming request. At least it requires mod_lsapi to connect to backend lsphp master-process and have it perform fork which leads to a slowdown.

With pool_mode enabled, mod_lsapi maintains persistent connections with backend which drastically increases performance (accelerates requests processing), but also increases the number of processes in LVE as well memory usage. Backend lsphp processes stays alive for lsapi_backend_max_idle time, or until lsapi_backend_max_reqs is reached (or Apache restarted).

Alternatively, we have another accelerating technology - *CRIU*, which is faster and uses less memory. But it is in Beta so far and available for CL7 only (stable version will appear in the near future).

Q: Your PHP installation appears to be missing the... How to manage native PHP with mod_lsapi under EasyApache 3?

A: There are several ways to do that.

1. Using PHP Selector.

To find PHP Selector in user's panel choose Select PHP Version icon as follows:

mod_lsapi_faq

From PHP Selector you can manage PHP version and choose the necessary extensions to be used by PHP. Choose proper PHP version from the drop-down and click Set as current. Mark proper checkboxes to choose extensions and click Save:

mod_lsapi_faq_01

This is a simple and convenient way to configure the user's PHP.

2. Using native PHP from PHP Selector.

mod_lsapi installs alt-php56 as native by default (just copy of alt-php56):

mod_lsapi_faq_02

The native version is not designed to enable or disable PHP extensions through the web interface of the PHP Selector. This can lead to missing of the proper PHP extensions for customers applications.

For example, you can get the following reply from the website that is using WordPress and native PHP:

mod_lsapi_faq_03

There are two ways to solve this problem:

1. Use non-native PHP with proper extensions enabled via the PHP Selector (described above).
2. Use native PHP with properly configured .ini files (described below).

To configure native PHP, use an additional .ini file /opt/alt/php56/link/conf/default.ini:

mod_lsapi_faq_04

By default it is empty. To solve the issue this way, the following strings must be added:

```
extension=/opt/alt/php56/usr/lib64/php/modules/mysql.so
```

```
extension=/opt/alt/php56/usr/lib64/php/modules/pdo_mysql.so
```

```
extension=/opt/alt/php56/usr/lib64/php/modules/pdo.so
```

All available extensions for alt-php56 can be seen by running the command:

```
# ls /opt/alt/php56/usr/lib64/php/modules/
```

Note. Some extensions may conflict with each other, be careful when enabling them through the default.ini file.

3. Using switch_mod_lsapi --build-native-lsphp as native.

You can find additional notes on native PHP installation (EasyApache 3 only) on the link: https://docs.cloudlinux.com/mod_lsapi_installation.html

To see what kind of native PHP is used, use the command:

```
# /usr/local/bin/php -v
```

Output example:

```
PHP 5.6.30 (cli) (built: Jun 13 2017 06:23:21)
```

```
Copyright (c) 1997-2016 The PHP Group
```

```
Zend Engine v2.6.0, Copyright (c) 1998-2016 Zend Technologies
```

The command switch_mod_lsapi --build-native-lsphp builds the lsphp of the same version, it will be used as native via the PHP Selector, but with another .ini file to configure.

mod_lsapi_faq_05

We do not recommend to use this native PHP because it does not support *CRIU*.

To revert alt-php56 to the native PHP, execute the following command:

```
# cp /opt/alt/php56/usr/bin/lspHP /usr/local/bin/
```

Installing mod_lsapi for Plesk

Installation process is done with yum:

```
yum install liblsapi liblsapi-devel
```

```
yum install mod_lsapi
```

When completed - run a command to setup mod_lsapi and register LSPHP handlers in Plesk Panel:

```
/usr/bin/switch_mod_lsapi --setup
```

The command adds LSPHPXY alt-php PHP handlers to Plesk Panel so they become available for domains.

Managing PHP handlers is fully done with Plesk Admin according to their documentation: <http://download1.parallels.com/Plesk/PP12/12.0/Doc/en-US/online/plesk-administrator-guide/70669.htm>

Quick reference:

Enabling lsapi for single domain is done with Plesk Panel > Subscriptions > [subscription name] > Hosting Settings > PHP Support, select desired LSPHP handler for domain.

Enabling lsapi for multiple domains is done with Plesk Service Plans > [Plan name] > PHP Settings > PHP support, select desired LSPHP handler to be used by all users under a plan. If a subscription is not locked (user changed nothing in it), after clicking 'Update and sync' domains will start using lsapi.

!Screenshot_20161029_132208!

There is no way to switch all plans to lsapi - it should be done one-by-one .

How to run LiteSpeed PHP (mod_lsapi) with PHP Selector

1. Chose Plesk PHP Settings

From the dialog box, select LSPHP by Vendor OS PHP version.

!php_settings!

Click Apply and OK to confirm.

2. Now you can manage your PHP (versions and modules) from PHP Selector.

Chose PHP Selector.

From the dialog box select proper PHP version and PHP modules (or defaults).

!php_version!

Click Save and Set as current to apply your choice.

From now on, on your domain will be applied PHP with version and modules which are set by PHP Selector handled by mod_lsapi.

3. Summary.

1. For correct work of PHP Selector - chose for domain "LSPHP by vendor OS".
2. If any other LSPHP version is chosen in Plesk, then PHP Selector will not be available anymore.
3. For enabling native PHP from the vendor, select "native" on the PHP Selector management page.

CRIU Support

[CloudLinux 7 only]

What is CRIU

CRIU is Checkpoint/Restore In Userspace, (pronounced kree-oo), is a software tool for Linux operating system. Using this tool, you can freeze a running application (or part of it) and checkpoint it as a collection of files on disk. You can then use the files to restore the application and run it exactly as it was during the time of freeze (more information on the link https://criu.org/Main_Page).

mod_lsapi-1.1-1 is the first beta version with freezing PHP implemented. mod_lsapi now supports the following parameters:

Option name	Description	Values	Default
lsapi_criu	Enable/disable CRIU for lsphp freezing.	On/Off	Off
lsapi_criu_socket_path	Set path to socket for communication with criu service.	[path to socket]	/var/run/criu/criu_service.socket
lsapi_backend_semtimedwait	Enable/disable flag for notification about lsphp started. This method avoid cycles of waiting for lsphp start.	On/Off	On
lsapi_backend_initial_start	Number of request when lsphp should be freed.	[number] 0 - no freezing	0
lsapi_criu_use_shm	Method of requests counting. Off - use shared memory. Signals - use signals from child processes to parent.	Off/Signals	Off
lsapi_criu_imgs_dir_path	Path to folder where imgs of freezed PHP will be stored.	[path]	/var/run/mod_lsapi/
lsapi_criu_debug	Enable/Disable CRIU related debug logging.	On/Off	Off

Example:

lsapi_criu On

lsapi_criu_socket_path /var/run/criu/criu_service.socket

lsapi_backend_semtimedwait On

lsapi_backend_initial_start 15

lsapi_criu_use_shm Off

lsapi_criu_debug Off

How it works

When Apache module mod_lsapi detects CRIU enabled (lsapi_criu On) it prepares a directory for images (on the first request of virtualhost) to store (lsapi_criu_imgs_dir_path /var/run/mod_lsapi/[dir_name]), and starts lsphp process. Lsphp increases counter (lsapi_criu_use_shm Off/Signals) via shared memory or signals, when counter reaches limit (lsapi_backend_initial_start 15), lsphp sends the request to CRIU for freezing. CRIU service makes images of requested processes. Lsphp will not be frozen if counter has not reached the limit. The next time when lsphp will be stopped, it will be unfrozen from the images.

The images of the processes will be saved even if Apache is restarted. But all images will be deleted after server restart by default configuration. This can be modified by setting the new path lsapi_criu_imgs_dir_path.

Important! If php.ini or configuration file from php.d is changed, the images must be deleted manually. We are working on automation of this action.

Note that CRIU can't correctly freeze lsphp with PrivateTmp enabled. For correct work, PrivateTmp must be false in httpd.service file. For disabling:

Copy httpd.service to /etc/systemd/system and change there PrivateTmp:

```
# cat httpd.service
```

```
[Unit]
```

```
Description=Apache web server managed by cPanel EasyApache
```

```
ConditionPathExists=!/etc/httpddisable
```

```
ConditionPathExists=!/etc/apachedisable
```

```
ConditionPathExists=!/etc/httpddisable
```

```
[Service]
```

```
Type=forking
```

```
ExecStart=/usr/local/cpanel/scripts/restartsrv_httpd -no-verbose
```

```
PIDFile=/var/run/apache2/httpd.pid
```

```
PrivateTmp=false
```

```
[Install]
```

```
WantedBy=multi-user.target
```

Or it would be technically better to provide a small override of service file rather than copying the whole new version in /etc/systemd/system ... (www.freedesktop.org/software/systemd/man/systemd.unit.html).

```
mkdir /etc/systemd/system/httpd.service.d
```

```
echo "[Service]" > /etc/systemd/system/httpd.service.d/nopt.conf
```

```
echo "PrivateTmp=false" >> /etc/systemd/system/httpd.service.d/nopt.conf
```

```
and
```

```
# systemctl daemon-reload
```

```
Installation
```

Criu is installed with dependency to mod_lsapi-1.1 package. To activate it:

1. Enable service and start it:

```
systemctl enable criu
```

```
systemctl start criu
```

2. Edit lsapi.conf file, turn CRIU On and set some defaults:

```
lsapi_criu On
```

```
lsapi_criu_socket_path /var/run/criu/criu_service.socket
```

```
lsapi_backend_semtimedwait On
```

```
lsapi_backend_initial_start 15
```

```
lsapi_criu_use_shm Off
```

3. Restart apache:

```
service httpd restart
```

Managing CRIU Images

1. An option added to the Apache configuration for cleaning all the images earlier saved by CRIU.

Option name	Description	Value	De- fault
lsapi_reset_criu_on_apache_restart	This option allows cleaning all CRIU images on Apache restart.	On/Off	Off

On the next restart of Apache all of the images will be cleaned.

It can be enabled by writing `lsapi_reset_criu_on_apache_restart On` in `lsapi.conf` (Virtual Host and `.htaccess` do not allow to use this option).

Note that this option works only if `lsapi_terminate_backends_on_exit` is On (default value is On, it is set in `lsapi.conf` too).

2. If you need to clean CRIU images for one user you can simply add `mod_lsapi_reset_me` file to the user's directory with CRIU images (default `/var/run/mod_lsapi/lsapi_*_criu_imgs`). On the next restart of `lsphp` the images will be cleaned.

3. Global reset flag for cleaning all earlier saved images by CRIU.

Current `mod_lsapi` allows cleaning all images only with one flag file.

Create `/usr/share/criu/mod_lsapi/lsp.php.criu.reset` file. Also don't forget to set such permissions `[nobody:nobody]` (or `[apache:apache]` for non cPanel) and access mode `[700]` to the `/usr/share/criu/mod_lsapi` directory.

Steps to do :

```
mkdir /usr/share/criu
```

```
mkdir /usr/share/criu/mod_lsapi
```

```
chown nobody:nobody /usr/share/criu/mod_lsapi
```

```
touch /usr/share/criu/mod_lsapi/lsp.php.criu.reset
```

On the next requests to all virtual hosts images will be recreated (deleted first and created again later - it depends on `lsapi_backend_initial_start` value).

4. Added possibility to clean CRIU images from user space.

If a user needs to clean CRIU images for `lsphp`, he should create a file: `~/mod_lsapi_reset_me_[vhost_name]`. Where `[vhost_name]` is a `ServerName` from the `VirtualHost` block in the configuration file. On the next restart of `lsphp`, the images will be cleaned.

Example:

```
cd; touch mod_lsapi_reset_me_criu.test.com
```

where `vhost.conf` contains:

```
ServerName criu.test.com
```

This mode is enabled by default and creates a separate `lsphp` process for each virtual host.

`mod_lsapi_reset_me_[vhost_name]` flag will not work for a user when `lsapi_per_user` option is On.

5. There is `lsapi_per_user` (default off) option in `mod_lsapi` that creates only one `lsphp` process for a user, regardless of the number of his virtual hosts. We don't recommend to use this option with CRIU, but if you use it, make sure that your virtual hosts (under the same user) have the same environment configurations. If they are not the same, this may cause undesirable `lsphp` process operation.

File Change API

- General Information*
- Usage and Integration*
- Installation and Configuration*
- Configuration Details*
- Low-level access*

General

General description

One of the main problems on a shared hosting system for file backup operations is to figure out which files have changed. Using INOTIFY on a 1T drive with a large number of small files and directories guarantees slow startup times, and a lot of context switching between kernel and userspace - generating additional load. On the other hand scanning disk for newly modified files is very IO intensive, and can kill the performance of the fastest disks.

CloudLinux approach

CloudLinux File Change API is a kernel level technology with the user space interface that buffers lists of modified files in the kernel and then off-loads that list to user space daemon.

After that - any software (with enough permissions) can get a list of files that has been modified for the last 24 hours.

The software is very simple to use and produces the list of modified files. As such we expect file backup software, including integrated cPanel backup system to integrate with this API soon.

Usage and Integration

Userland utilities

`/usr/bin/cloudlinux-backup-helper` is a utility for getting the list of changed files.

It is supposed to be run by a super user only.

Command line parameters:

`-t | --timestamp` retrieve file names for files modified after specified timestamp

`-u | --uid` retrieve file names for particular UID only

If no UID specified, are retrieved for all users. If no timestamp specified, all database events are shown.

Output Format

protocol version (1 right now), timestamp (in seconds) - up to which time data was collected

UID: absolute path to file changed

UID: absolute path to file changed

...

Note. The timestamp in output is needed so you can clearly identify from which timestamp to get list of changed files next.

Examples:

```
[root@localhost ~]# cloudlinux-backup-helper -t 1495533489 -u <UID>
```

```
1,1495533925
```

```
1001:/home/user2/public_html/output.txt
```

```
1001:/home/user2/public_html/info.php
```

```
[root@localhost ~]# cloudlinux-backup-helper -t 1495533489
```

```
1,1495533925
```

```
1000:/home/user1/.bashrc
```

```
1001:/home/user2/public_html/output.txt
```

```
1001:/home/user2/public_html/info.php
```

```
1003:/home/user3/logs/data.log
```

Getting changed files by end user

/usr/bin/cloudlinux-backup-helper-uid is a SUID wrapper for the cloudlinux-backup-helper utility that enables an end user to get the list of files changed. It accepts timestamp argument only and retrieves data of the user who is running it only.

Examples:

```
[user@localhost ~]$ cloudlinux-backup-helper-uid
```

```
1,1495530576
```

```
1000:/home/user/.bash_history
```

```
[user@localhost ~]$ cloudlinux-backup-helper-uid -t 1495547922
```

```
1,1495548343
```

```
1000:/home/user/file1.txt
```

```
1000:/home/user/file2.txt
```

This command is available within CageFS.

Installation and Configuration

cloudlinux-fchange-0.1-5

Requirements

CloudLinux OS 6 (requires Hybrid kernel) or 7

Kernel Version: 3.10.0-427.36.1.lve1.4.47

Installation and Configuration

To install cloudlinux-fchange system run:

CloudLinux 7

```
yum install cloudlinux-fchange --enablerepo=cloudlinux-updates-testing
```

CloudLinux 6 Hybrid

```
yum install cloudlinux-fchange --enablerepo=cloudlinux-hybrid-testing
```

Configuration file can be found in /etc/sysconfig/cloudlinux-fchange

Database containing list of modified files is located at /var/lve/cloudlinux-fchange.db by default.

Starting and Stopping

After successful installation the event collecting daemon starts automatically, providing all kernel-exposed data are in place.

To start daemon:

CloudLinux 7

```
systemctl start cloudlinux-file-change-collector
```

CloudLinux 6 Hybrid

```
service cloudlinux-file-change-collector start
```

To stop daemon:

CloudLinux 7

```
systemctl stop cloudlinux-file-change-collector
```

CloudLinux 6 Hybrid

```
service cloudlinux-file-change-collector stop
```

Uninstalling

To uninstall cloudlinux-fchange run:

```
yum remove cloudlinux-fchange
```

Configuration Details

Configuration resides in `/etc/sysconfig/cloudlinux-fchange`. The following is the default configuration (see comments):

```
# sqlite database file path. If commented out a default value is used
#database_path=/var/lve/cloudlinux-fchange.db

# If uncommented paths starting with 'include' one are processed only
# Pay attention this parameter is a regular string, not a regex
# To include more than one item just specify several lines to include:
# include=/one
# include=/two

# If uncommented exclude paths which contain 'exclude'
# Pay attention this parameter is a regular string, not a regex
# To exclude more than one item just specify several lines to exclude:
# exclude=var
# exclude=tmp

# Daemon polling interval in seconds
polling_interval=5

# Time to keep entries in days. Does not clean if commented out or zero
time_to_keep=1

# User read-only mode minimal UID
```

```
# If file change collector stopped, all users with UID >= user_ro_mode_min_uid
# are restricted to write to their home directory. This prevents to miss
# a file change event.
# Value of -1 (default) allows to disable the feature
user_ro_mode_min_uid=-1
# Minimal UID of events to be processed.
# Events of users with UID less then specified are not handled.
# By default 500 (non-system users for redhat-based systems)
#minimal_event_uid=500
# SQLite shared lock prevents setting more restrictive locks. That is a
# process cannot write to a database table when a concurrent process reads
# from the table. As saving data to database is considered far more important
# than getting them (data could be reread a second later after all), database
# writer could try to terminate concurrent reading processes. Just set
# terminate_rivals to 'yes' to turn this ability on.
# terminate_rivals=no
# Events to be handled. Currently the following types of events are processed:
# 1. file creation
# 2. file deletion
# 3. directory creation
# 4. directory deletion
# 5. file content/metadata modification
# 6. file/directory attributes/ownership modification
# 7. hardlink creation
# 8. symlink creation
# 9. file/directory moving/renaming
# By default all events are processed. Keep in mind that events for a filepath
# are cached, i.e if a file was deleted and then a file with the same absolute
# name is created, only the deletion event is triggered. Changing file
# modification timestamp with command 'touch' will trigger modification event
# as if a file content is modified.
# Currently supported options are:
# file_created, file_modified, file_deleted, dir_created, dir_deleted,
# owner_changed, attrib_changed, moved, hardlink_created, symlink_created, all
# Options that don't have 'file' or 'dir' prefix, applied to both files and
# directories. To set more than one options, separate them with commas,
```

e.g. event_types=file_created,file_deleted,file_modified. Unknown options are

ignored.

#

event_types=all

Please keep in mind, that current implementation implies that one process is writing to a database and another is reading from it. As reading sets shared lock to a database table, the writing process cannot write to the table until the lock is released. That's why passing a timestamp to cloudlinux-backup-helper matters: this way the number of records to be returned is substantially decreased, lowering the processing time and filtering out old records. Likewise, pay attention to narrowing the scope of events being recorded. Chances are that changing attributes, ownership, directory creation/deletion, symlink events are not relevant and there's no need to keep them.

Low-level access

Note. Using this options is dangerous, and might cause problems with CloudLinux file change API.

The kernel exposes the functionality to /proc/sys/fs/datacycle folder.

1.enable - enable/disable the functionality. Write 1 to this file to enable, 0 to disable. If disabled, no events are coming to events file.

2.events - the modified files log itself. Events in the format <EVENT_ID>:<EVENT_TYPE_ID>:<USER_ID>:<FILE_PATH> are constantly appending to the end of the file if datacycle enabled. File events are never duplicated: if we have file modification event, we would not get file deletion event if the file has been later deleted. This events buffer has limited capacity, therefore from time to time, the events log requires flushing.

3.flush - a file for clearing events log. For flushing, the last event_id from the events file is written to this file. Right after this, events log is truncated to that event_id.

4.user_ro_mode - forbidding users with UIDs equal or bigger that set in this file writing to their home directories. At the boot, the file has -1. When it's written positive value, say 500, the system starts effectively preventing users from modifying their home dirs (on write attempt a user gets 'read-only filesystem' error). This feature is designed to prevent users from updating their home dirs when events are not handled.

5.entries_in_buffer - just counter of log entries of events file.

6.min_event_uid - this file has minimal UID of events to be handled. Events from users with smaller UID are not handled. By default 500 (non-system users in redhat-based systems).

mod_proctitle

mod_proctitle is a module for gathering URL information per request. It is available only for Apache 2.4 now.

For installation:

cPanel EasyApache 3 and non cPanel (CloudLinux 7 only for non cPanel):

```
# yum install mod_proctitle --enablerepo=cloudlinux-updates-testing
```

```
# service httpd restart
```

cPanel EasyApache 4:

```
# yum install ea-apache24-mod_proctitle
```

```
# service httpd restart
```

DirectAdmin:

```
# cd /usr/local/directadmin/custombuild
# ./build update
# ./build mod_proctitle
```

How to Read mod_proctitle Information

How to read information gathered by module

For reading information saved by module use the following script (the script is not in the package):

```
# cat proctitles_info.sh
#!/bin/bash
HTTPD=httpd
for pid in $(/usr/bin/pgrep $HTTPD); do
    for tid in $(ls /proc/$pid/task); do
        found=no
        for shm in $(ls /dev/shm/apache_title_shm_${pid}_${tid}_*
2>/dev/null); do
            found=yes
            title=$(/usr/bin/tr -d '\0' < $shm)
            thread_id=$(/bin/basename "${shm}" | sed
"s/apache_title_shm_${pid}_${tid}_/"
echo "$pid.$tid - $thread_id - $title"
            break
        done
        if [ "$found" = "no" ]; then
            echo "$pid.$tid NOT FOUND"
        fi
    done
done
```

Here are the examples of saved by module:

```
# sh proctitles_info.sh
571258.571258 NOT FOUND
571300.571300 NOT FOUND
571303.571303 - 0000000000000000 - 1466513333.6 test.cloudlinux.com GET /1.php HTTP/1.1
571304.571304 - 0000000000000000 - 1466513335.3 test.cloudlinux.com GET /1.php HTTP/1.1
571305.571305 - 0000000000000000 - httpd
571306.571306 - 0000000000000000 - httpd
571307.571307 - 0000000000000000 - httpd
```

571372.571372 - 0000000000000000 - httpd

571374.571374 - 0000000000000000 - httpd

Item info:

[pid].[tid] - [posix thread id] - [request info]

Request information can contain:

NOT FOUND - means that process of Apache doesn't handle requests.

httpd - request is active and waiting for new connection.

[seconds].[tenths of second] [host] [METHOD] [URL] [PROTOCOL]

Tuning Parameters

Module's parameters for tuning

WatchHandlers	List of handlers for monitoring (httpd.conf, virtualhost).
ProctitleUseFilter On/Off	Use old method of cleaning information or new via filter (for prefork better to use Off)

Additional Packages

CloudLinux will package additional software needed by hosters for your convenience.

- Git for cPanel*

- alt-suexec*

- tuned-profiles-cloudlinux*

- cloudlinux-fchange*

Git for cPanel

Please note that 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
```

alt-suexec

What is alt-suexec package needed for?

If you use standard httpd from our repository, but your users' sites do not match standard Apache location of /var/www, then you should use alt-suexec.

alt-suexec package brings suEXEC binaries pre-compiled for specific locations, like /home.

How to switch suEXEC with alt-suexec

Based on httpd 2.2 basic for 6 and httpd 2.4 basic for CloudLinux 7, the package brings to server a set of suEXECs with different DOCUMENT ROOTs and MIN_UID/MIN_GID parameters. The first set of suEXECs is listed by such modes:

```
# switch_suexec -l
```

```
USE_HOME - DOCUMENT ROOT /home/ MIN_UID 500 MIN_GID 100 CALLER apache
```

```
USE_WWW - DOCUMENT ROOT /var/www/ MIN_UID 500 MIN_GID 100 CALLER apache
```

The package also brings its own utility for installing specific suEXEC:

```
# switch_suexec -h
```

-l	list of available suexec
-u	update suexec according to /etc/sysconfig/alt-suexec
-s	set new suexec and install it
-p	set new suexec path and install it
-o	set new suexec owners and install it
-r	restore native apache suexec

There are two ways to set up new suEXEC binary:

1. via config file /etc/sysconfig/alt-suexec
2. via utility switch_suexec

Here are the examples of how to set up suEXEC with DOC_ROOT = “/home”:

1.
 1. add string “USE_HOME” to /etc/sysconfig/alt-suexec
 2. run the command switch_suexec -u
2.
 1. switch_suexec -sUSE_HOME

Result of both methods:

```
# cat /etc/sysconfig/alt-suexec
```

```
USE_HOME
```

Here is standard suEXEC for CloudLinux 6 clean server:

```
# /usr/sbin/suexec -V
```

```
-D AP_DOC_ROOT="/var/www"
```

```
-D AP_GID_MIN=100
```

```
-D AP_HTTPD_USER="apache"
```

```
-D AP_LOG_EXEC="/var/log/httpd/suexec.log"
```

```
-D AP_SAFE_PATH="/usr/local/bin:/usr/bin:/bin"
```

```
-D AP_UID_MIN=500
```

```
-D AP_USERDIR_SUFFIX="public_html"
```

```
-D AP_SAFE_DIRECTORY="/usr/local/safe-bin"
```

Here is output of new suEXEC after USE_HOME installtion:


```
# /usr/sbin/suexec -V
-D AP_DOC_ROOT="/home/"
-D AP_GID_MIN=100
-D AP_HTTPD_USER="apache"
-D AP_LOG_EXEC="/var/log/httpd/suexec.log"
-D AP_SAFE_PATH="/usr/local/bin:/usr/bin:/bin"
-D AP_UID_MIN=500
-D AP_USERDIR_SUFFIX="public_html"
-D AP_SAFE_DIRECTORY="/usr/local/safe-bin"
```

Description of other switch_suexec parameters:

-p	if suexec binary file will be placed not in standard way /usr/sbin - specify this new path with p-option
-o	if suexec binary file not owned by root:apache - specify new owner with o-option

For most cases -p and -o options for standard Apache are useless.

Correct suEXEC will be restored even after httpd update or reinstall.

List of pre-built suEXEC binary files stored without suid bit and not executable.

How to install alt-suexec?

For installation run the command:

```
yum install alt-suexec --enablerepo=cloudlinux-updates-testing
```

New suexec with custom parameters

If you need suEXEC with custom parameters absent in current set of alt-suexec, please submit a ticket on <https://helpdesk.cloudlinux.com/> and we will add new suEXEC with needed parameters.

tuned-profiles-cloudlinux

The tuned-profiles-cloudlinux package brings a range of kernel under-the-hood tunings to address high LA, iowait issues what were detected earlier on particular users deploys. The package also encloses OOM adjustments to prioritize the elimination of overrun PHP, lsphp, Phusion Passenger workers processes over other processes (e.g. ssh, a cron job).

There are three profiles provided by CloudLinux:

```
# tuned-adm list | grep cloudlinux
```

- cloudlinux-default - Default CloudLinux tuned profile
- cloudlinux-dummy - Empty CloudLinux tuned profile
- cloudlinux-vz - Empty CloudLinux tuned profile

cloudlinux-dummy and cloudlinux-vz are used for internal needs or when Virtuozzo/OpenVZ detected and actually do nothing.

cloudlinux-default is one to be used, it actually does the following:

1. Switches CPU power consumption mode to the maximum. CPU operates at maximum performance at the maximum clock rate:

```
governor=performance
```

energy_perf_bias=performance

Note. If standard software CPU governors are used.

2. Applies the following kernel options:

vm.force_scan_thresh=100 - Improves kernel memory clean-up in case of big number of running LVE.

UBC parameters set the limits for the containers:

ubc.dirty_ratio=100 - Defines maximum RAM percentage for dirty memory pages.

.dirty_background_ratio=75 - Defines RAM percentage when to allow writing dirty pages on the disk.

3. [CloudLinux 7 only] Detects used disk types and changes elevator to 'deadline' for HDD and to 'noop' for SSD in /sys/block/[blockname]/queue/scheduler.

Note. The script uses /sys/block/[blockname]/queue/rotational flag, some RAID controllers can not set it properly. For example, SSD used for RAID but rotational is set to 1 by RAID driver. As a workaround add the following to /etc/rc.d/rc.local to make it applied on boot:

```
echo "noop" > /sys/block/[blockname]/queue/scheduler
```

```
echo "0" > /sys/block/[blockname]/queue/rotational
```

Where [blockname] is used device name, like sda/sdb.

And make it executable:

```
chmod +x /etc/rc.d/rc.local
```

4. [CloudLinux 7 only] The profile sets I/O scheduler. For the normal discs the Deadline Scheduler is set to improve IO performance and decrease IO latency, for SSD - noop.

When configuring scheduler I/O queue is changed and set to the value 1024 which improves overall I/O subsystem performance by caching IO requests in memory.

5. Disables transparent HugePage.
6. Provides adjustment group file for OOM-Killer to kill overrun php, lisp and Phusion Passenger workers first.

To install:

```
yum install tuned-profiles-cloudlinux
```

To start using a profile:

```
tuned-adm profile cloudlinux-default
```

To stop using a profile:

```
tuned-adm off
```

Integration Guide

Here you will find the instructions and common techniques used to integrate your software with CloudLinux.

- *Detecting and Working with CloudLinux.*
- *Displaying CPU, Memory & IO limits.*
- *Integrating LVE Limits with Packages.*

Detecting and Working with CloudLinux

Detecting if system is running CloudLinux/CloudLinux kernel:

```
$ uname -r | grep lve
```

If you get an output, it means the system is running 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 the presence of /proc/lve/list file.

Check if CageFS is enabled (as root):

```
$ /usr/sbin/cagefsctl --cagefs-status
```

Check if CageFS is enabled for a particular user (as root):

```
$ /usr/sbin/cagefsctl --user-status _USER_NAME_
```

Check if you are inside CageFS:

Check for the presence of /var/.cagefs/.cagefs.token file - if present, it means that you are inside CageFS.

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 and processed by a control panel to display the 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 and limits for all customers. If a customer is not present in the file, it means that customer is not active (no scripts were executed recently for the customer), and a 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 the first 9 parameters are available. You can check the the version by reading the first byte of /proc/lve/list.

On the 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.

Integrating LVE Limits with Packages

[lve-utils 1.4+]

CloudLinux can automatically detect the most popular control panels, like cPanel - and allows to set different limits for users in different packages. It 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

A script can be written in any language, and it has to be executable.

It should accept the 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 `/etc/sysconfig/cloudlinux` file.

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 the script example please check the following article: <http://kb.cloudlinux.com/2015/02/integrating-lve-limits-with-packages-for-unsupported-control-panel/>

Partner Portal

• *IP Reseller PartnerUI*

IP Reseller Partner UI

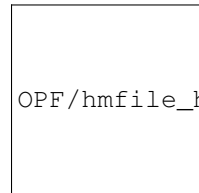
To become CloudLinux reseller partner you should first register your account following this link: <https://cln.cloudlinux.com/clweb/login.xhtml> and contact us to apply for your access status.

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.



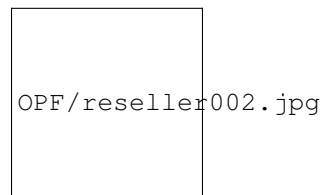
2. To delete license click on recycle bin icon in front of the needed IP-address.

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.



When 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.

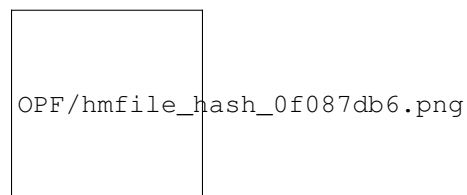
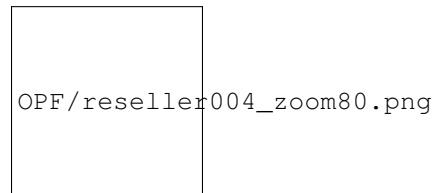
If you choose Auto repay, your card will be automatically charged when your balance becomes negative. Minimal charge is \$20 (E.g. for balance -\$15 - you'll be charged at \$20, for balance -\$134.2 - you'll be charged at \$134.2).



Note: If your balance is shown as negative, it means that you have to deposit more funds.

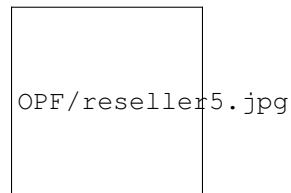
API Section

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 plugins APIs in API Section.



Profile

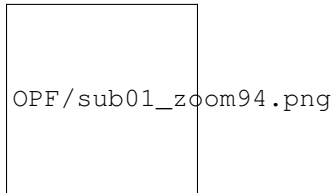
You can edit your profile information by clicking on Profile section. Edit the necessary info and click Update Account.



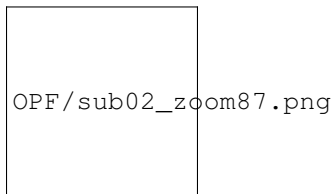


Sub Accounts

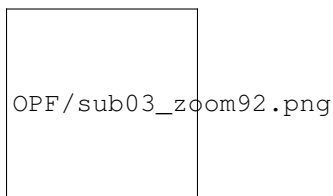
To create and manage sub accounts choose Sub accounts tab - you will get to Sub Accounts Management page where all the sub accounts are displayed in the list.



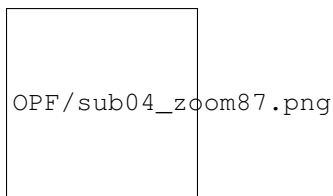
To create a sub account click Add Sub Account. Fill the obligatory fields marked with the asterisk* and click Add Sub Account in the bottom of the window.



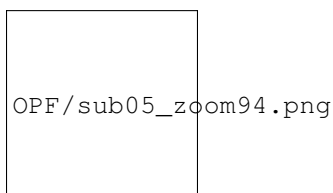
To edit a sub account click on the wrench icon, make changes and click Edit Sub Account in the bottom. All the fields are available for editing except the Login.



To remove a sub account click on recycle bin icon, enter login of a sub account to be removed and click Remove Sub Account.

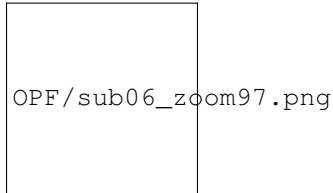


To manage sub account keys/servers click on a proper sub account login - you will get to the sub account management page with two tabs: Keys and Servers.

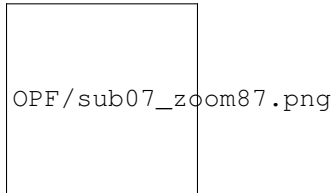


Choose Keys tab to view Tokens and servers linked to them. Click drop-down arrow in front of a token to view linked servers list.

To create a new key click Create key. To remove a key or a server click on recycle bin icon in front of a proper item.



Choose Servers tab to view the list of all servers.



CloudLinux Network

CloudLinux Network is designed to easily manage your CloudLinux and KernelCare licenses and servers by means of very simple and user-friendly interface.

A user can add, delete and edit licenses and track all the associated costs.

Fill out the simple registration form to create your account on <https://cln.cloudlinux.com/clweb/login.xhtml>. After activation, log in to the website.

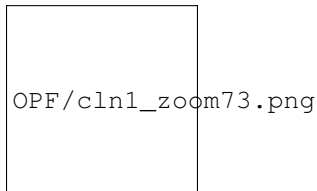
Note that in your CLN account you will see the licenses purchased directly from CloudLinux, not from the resellers.

Managing Licenses

Managing Licenses

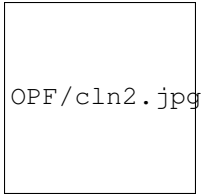
To add CloudLinux or KernelCare license:

1. Go to Manage Licenses page by clicking on Licenses tab. If you do not have any licenses yet, then you will see the following message “CloudLinux: You don’t have any CloudLinux license”.



2. Click on +Add More Licenses – you will get to CloudLinux Shopping Cart.

3. Choose the appropriate quantity of servers for CloudLinux Standart and KernelKare, choose pay interval (yearly or monthly) for KernelCare and click Checkout.

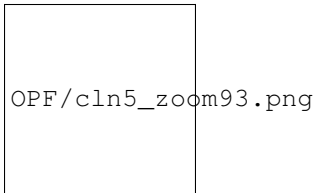


4. Choose your payment method (Credit card or PayPal), enter credit card or paypal information and click Update. If all the information is correct, you will be able immediately to purchase licenses.



5. To delete license click on bin icon in front on of them.

6. You can get trial unlimited activation key by clicking Get CL Unlimited Activation Key - trial subscription will be created automatically and a notification will appear in the upper area of the page. With unlimited key you can activate as many servers as you need, up to the number of purchased licenses. It doesn't provide license to unlimited servers.



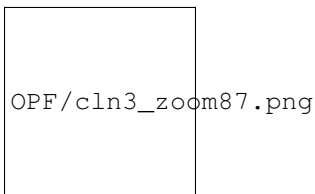
Managing Servers

Managing Servers

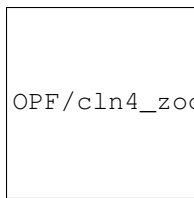
Click on Servers tab to get to Servers page where you can add, delete or manage your servers. Filter your servers by



IP, activation key, etc. To delete server click on bin icon in front of the server you want to delete.



Note. When you remove a server, by clicking Remove server button, the notification appears saying that after deleting a server you will still be paying for license for this server. To delete unused licenses follow simple steps in Cancel Licenses section on this page below.

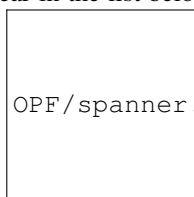


To manage KernelCare servers click on KernelCare keys tab.

To add new key enter Max Servers number, add description if needed and click Add. New key will be generated and



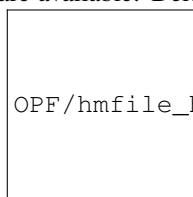
will appear in the list below. In the Operations column four operations are available: Delete key



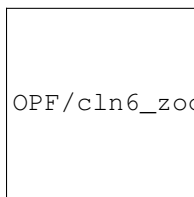
Edit key



, Add IP range



and Refresh

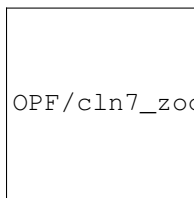


Cancel Licenses

Cancel Licenses

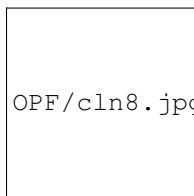
You can cancel a license only in case if it is not in use (server under this license was not registered or was deleted). The license is not active and can be canceled as long as the servers are not added.

1. To cancel CloudLinux or KernelCare license click Remove unused license.

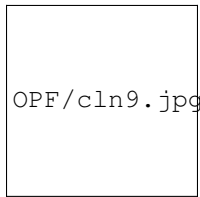


2. If you need to cancel an active license, you have to remove servers first.

Go to Servers tab, mark proper checkbox in front of a server (or several) you want to remove and click Remove servers.



Then go back to step 1 and remove the license as unused.



Note. You will still be paying for a license after removing servers until you remove the unused license.

CloudLinux WHMCS Plugin

- Overview*

- Installation & Configuration*

- o*Installation and Update*

- o*Configuration of Product*

- o*Configuration of Add-on*

- Management*

- o*Link Via Add-on. Optional License*

- o*Link Products Directly*

- o*Link Via Configurable Options*

- o*Link Add-ons Directly (for WHMCS 7.2.x and later)*

- o*Order*

- o*Admin Area*

- o*Client Area*

- o*Licenses List*

- o*Addon Licenses List (for WHMCS 7.2.x and later)*

- Common Problems*

Overview

CloudLinux Licenses for WHMCS allows you to automatically provision CloudLinux, Imunify360, and KernelCare licenses along with selected products. You can provision them for free or as a paid add-on to your product. Owing to CloudLinux Licenses add-on, all module commands on your main product are automatically reproduced on the license product.

Admin Area Functionality

- Create License*

- Terminate License*

- Suspend/Unsuspend License*

- Change License IP Address*

- View License Details*

Client Area Functionality

- View License Details
- Change License IP Address

Add-on Functionality

- Manage Relations Between Add-on And License Product
- Manage Relations Between Server And License Product
- Automatically Add License Product To Order When Relation Is Triggered
- View Existing Licenses
- Dependencies Between Module Actions - Every Action: Create, Terminate, Suspend Or Unsuspend Called On The Server Product Will Result With The Same Action Performed On The Licensed Products
- Flexible Filtering Of Existing Licenses

Additionally

- Multi-Language Support – Only Provisioning Module
- Supports CloudLinux, KernelCare, and Imunify360 Licenses
- Supports WHMCS V6 and Later

Installation & Configuration

In this section we will show you how to set up our products.

- Installation and Update*
- Configuration of Product*
- Configuration of Add-on*

Installation and Update

1.Download CloudLinux Licenses For WHMCS:

Production: <http://repo.cloudlinux.com/plugins/whmcs-cl-plugin-latest.zip>

Beta: <http://repo.cloudlinux.com/plugins/whmcs-cl-plugin-beta.zip>

2.Upload archive to your WHMCS root folder and extract it. Files should automatically jump into their places.

3.Run the following script:

```
php <whmcs_root>/clDeploy.php --migrate
```

Configuration of Product

1.Log into your WHMCS admin area and go to the Setup → Products/Services → Products/Services. Click Create a New Group.

2.Fill Product Group Name (product group will be visible under that name in your WHMCS system) and click Save Changes.

3. Click Create a New Product. Choose Other from Product Type drop-down menu and previously created product group from Product Group drop-down menu.
4. Fill Product Name and click Continue.
5. Set up this product as hidden by ticking Hidden checkbox at Details tab. Do not set up pricing for this product. Pricing will be done in another way.
6. Go to the Module Settings tab and select CloudLinux Licenses from Module Name drop-down.
7. Fill Username and Password with your CloudLinux API access details and select CloudLinux from License Type drop-down.
8. Click Save Changes to confirm.

Configuration of Add-on

1. Go to Setup → Add-on Modules, find CloudLinux Licenses Add-on and click Activate next to it.
2. The next step is permitting access to this module. Click Configure, select admin role and confirm by clicking Save Changes.

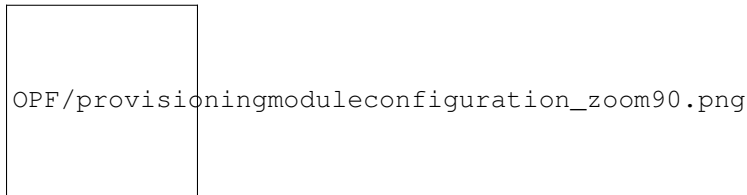


Fig 1: CloudLinux License For WHMCS provisioning module configuration.

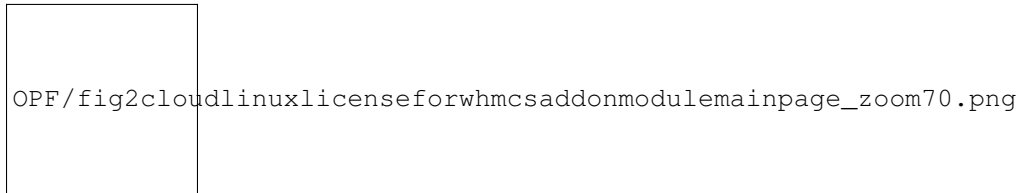


Fig 2: CloudLinux License For WHMCS add-on module main page.

Management

In this section you can find two ways of linking license product with your server product as well as other possibilities of the module.

- *Link Via Add-on. Optional License*
- *Link Products Directly*
- *Link Via Configurable Options*
- *Link Add-ons Directly* (for WHMCS 7.2.x and later)
- *Order*
- *Admin Area*
- *Client Area*
- *Licenses List*

•*Add-on Licenses List* (for WHMCS 7.2.x and later)

Link Via Add-on. Optional License

In order to allow your client to decide whether he wants to order server with or without a license, we will use Product Add-on. In this way, when the client orders an add-on, the relation will be triggered and the license product will be ordered along with module.

The following steps must be performed to prepare such connection:

- 1.Go to Setup → Products/Services → Products Add-ons and click Add New Add-on.
- 2.Fill add-on name, set up billing cycle and price.
- 3.Then tick Show on Order checkbox, assign add-on to the product and click Save Changes.
- 4.



Fig 3: Configuration of product add-on, which will trigger license product adding.

- 4.Go to Add-ons → CloudLinux Licenses Add-on → Add-on Relations and click Add Relation.
- 5.Select previously created product add-on and license product as shown below and click Add Relation.

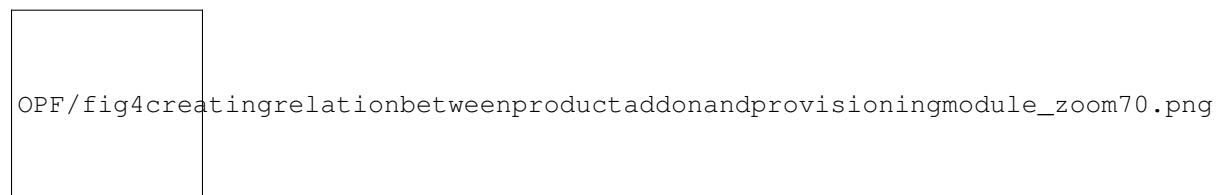


Fig 4: Creating relations between product add-on and provisioning module.

Link Products Directly

If you want to offer server along with the license, perform the following steps.

Note. Please do not set up pricing for license provisioning product. In exchange, you can increase a price for server provisioning product.

1. Prepare license provisioning product as described in the *Configuration of Product* section of this documentation.
2. Go to Add-ons → CloudLinux Licenses Add-on → Products Relations and click Add Relation.
3. Select server provisioning product from the Main Product drop-down list and license provisioning product from Linked Product With License and click Add Relation.



Fig 5: Creating relations directly between server and license provisioning modules.

Link Via Configurable Options

In order to allow your client to decide whether he wants to order server with or without license we can use Configurable Options (https://docs.whmcs.com/Addons_and_Configurable_Options).

Below we will show you what steps to proceed to prepare such connection:

- Configure CloudLinuxLicenses product as described here
- Go to Setup → Products/Services → Configurable Options and click Create a New Group.
- Fill group name and add New Configurable Option, set up billing cycle, price and option type. Then save changes.
- Go to Add-ons → CloudLinux Licenses Add-on → Configurable Options Relations and click Add Relation.
- Choose appropriate configurable option and license product which it is assigned to and click Add Relation.

Note. The plugin doesn't support "quantity" type of Configurable Options.

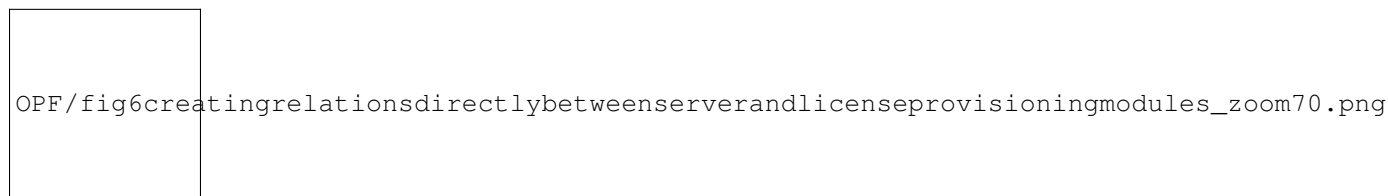


Fig 6: Creating relations directly between server and license provisioning modules.

Link Add-ons Directly

[for WHMCS 7.2.x and later]

WHMCS 7.2 introduces the ability to associate Product Add-ons with Provisioning Modules.

In order to allow your client to decide whether he wants to order server with or without license we will use product add-on. Below we will show you what steps to proceed to prepare such connection.

1. Go to Setup → Products/Services → Products Add-ons and click Add New Addon.
2. Fill add-on name, set up billing cycle and price. Then tick Show on Order checkbox, assign an add-on to the product.
3. Go to Module Settings tab and select CloudLinuxLicenses from Module Name drop-down.

4.Fill Username and Password with your CloudLinux API access (API secret key) details and select CloudLinux from LicenseType drop-down.

5.Click Save Changes to confirm.

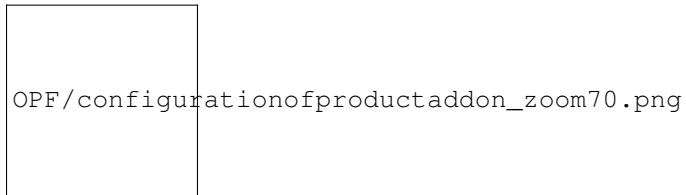


Fig 7: Configuration of product add-on with Provisioning Modules.

Order

The only difference between two ways of setting up relation is the ability to order server without CloudLinux license.

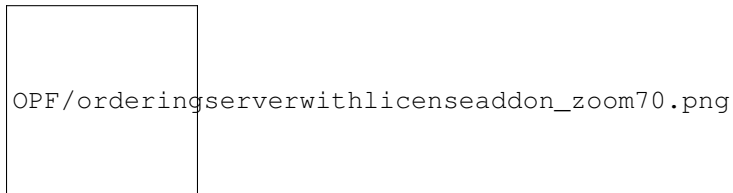


Fig 8: Ordering server with license add-on.

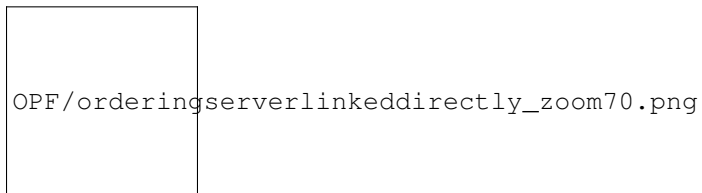


Fig 9: Ordering server linked directly with license product.

Admin Area

From the admin area it is possible to command such action as create, terminate, suspend/unsuspend and change IP address. Nonetheless, these actions can be ordered only on the server provisioning module and will be automatically reproduced for the license provisioning product.

Only change IP address feature has to be ordered manually.

You can also view the details of created license.

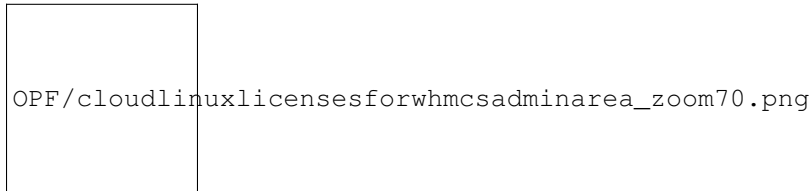


Fig 10: CloudLinux Licenses For WHMCS Admin Area.

Client Area

The clients are also able to view their servers license details. And as well as you, they are able to change IP address of their licenses.

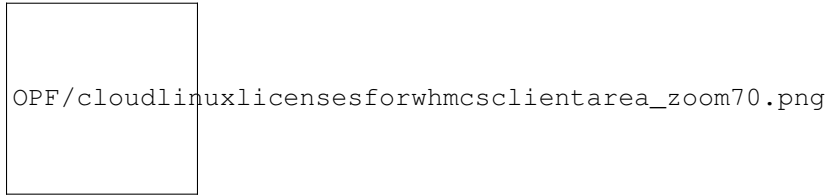


Fig 11: CloudLinux Licenses For WHMCS Client Area.

To change IP address click Change as shown on the screen above. Then specify IP address and click Save.

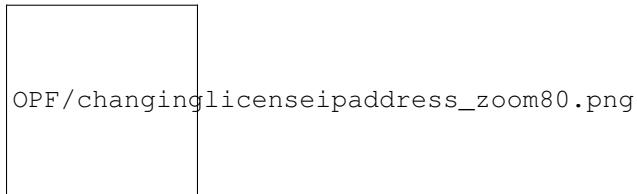


Fig 12: Changing License IP Address.

Licenses List

You can view the list of all licenses owned by your client at our add-on → Licenses List.

You can filter the list of licenses by client name, server provisioning products, license provisioning products and license IP address.

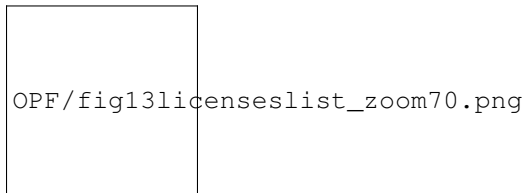


Fig 13: Licenses list.

Add-on Licenses List

You can view a list of all product add-on with Provisioning Modules licenses owned by your client at our add-on → Licenses List.

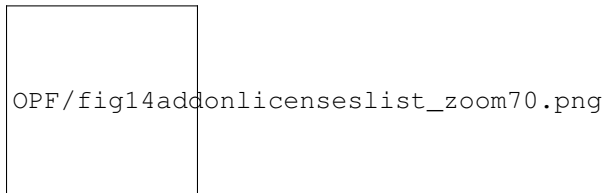


Fig 14: Add-on Licenses List.

Common Problems

After activating the server provisioning product, license provisioning product bounded to it is still pending.

Reason: License IP address may be already taken.

Solution: Change server IP address.

Note. Currently, only key-based licenses are available for Imunify360. Support of IP-based licenses will be added soon.

Deprecated

- LVE-Stats 0.x*

- OptimumCache*

LVE-Stats 0.x

NOTE. LVE-STATS-0.X IS NO LONGER SUPPORTED, PLEASE USE *LVE-STATS 2*

lve-stats package 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), run:

```
$ 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 stores history 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
ICPU	CPU Limit
aEP	CPU Limit
mEP	Max Entry Processes
IEP	Entry Proc limit
aNPROC	Average Number of Processes
mNPROC	Max Number of Processes
INPROC	Number of Processes limit
aVMEM	Average virtual Memory Usage
mVMEM	Max virtual Memory Usage
IVMEM	Virtual Memory Limit
aPMEM	Average physical Memory Usage
mPMEM	Max physical Memory Usage
IPMEM	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    ICPU    aEP    mEP IEP    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

Storing statistics in MySQL

NOTE. LVE-STATS-0.X IS NO LONGER SUPPORTED, PLEASE USE *LVE-STATS 2*

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

```
$ 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, run:

```
$ 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 database and user. You can do it by executing the 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,
    lnpoc INTEGER, npoc INTEGER, npoc_max INTEGER, npoc_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 the 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 npoc INTEGER;
ALTER TABLE history ADD npoc_max INTEGER;
ALTER TABLE history ADD npoc_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, npoc = 0, npoc_max = 0, npoc_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 TIMESTAMPTZ, daily TIMESTAMPTZ, server_id CHAR(10), lve_version  
INTEGER);
```

Update lve-stats RPM on all your servers.

If you use central database to store lvestats data, execute the 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.

Storing statistics in PostgreSQL

NOTE. LVE-STATS-0.X IS NO LONGER SUPPORTED, PLEASE USE *LVE-STATS 2*

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

```
$ 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 a database and a user. You can do it by executing the 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),
    lmempy 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_max: INTEGER, liops: INTEGER, iops: 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 TIMESTAMPTZ, daily TIMESTAMPTZ, 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!

Compacting in multi-server settings

NOTE. LVE-STATS-0.X IS NO LONGER SUPPORTED, PLEASE USE *LVE-STATS 2*

[lve-stats 0.10+]

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

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

On all other servers, change that option to `slave`.

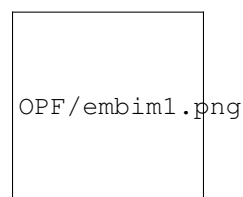
Default: `single` – should be used when lve-stats stores data to a single database.

OptimumCache

NOTE. OPTIMUMCACHE IS NO LONGER SUPPORTED.

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.

Further reading: <http://kb.cloudlinux.com/tag/optimumcache/>

Installation

NOTE. OPTIMUMCACHE IS NO LONGER SUPPORTED.

Requirements:

64bit CloudLinux 6.x or higher

ext4 filesystem

kernel lve1.2.55 or later.

CHAPTER 14

Installation:

```
# yum install optimumcache
```

OptimumCache must be provided with list of directories to expect duplicate files be in:

```
# occtl -recursive -mark-dir /home
```

```
# occtl -recursive -mark-dir /home2 (for cPanel)
```

```
# occtl -recursive -mark-dir /var/www (for Plesk)
```

OptimumCache is going to index these directories. Thus system load during this period (from hours to days) might be as twice as high. See ‘Marking directories’ [http://docs.cloudlinux.com/index.html?marking_directories.html].

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/file [new size[KMGT]]
```

`/path/to/new/image/file` must be file path + file name, not a directory name.

Example:

```
# occtl -move-ploop /var/ssh/optimumcache.image
```

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 OptimumCache-0.1-21 and earlier, you may consider removing this `fstab` entry in OptimumCache 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]]
```

A common reason for resizing ploop is reacting to OptimumCache syslog message like “OptimumCache recommends cache storage size to be at least ... GB”

Deleting ploop:

```
# occtl --delete-ploop
```

For the case when this action cannot be completed due to “Unable unmount ploop” issue, there is a workaround in “Troubleshooting” section.

Q. I created/resized/moved/deleted ploop. Do I need to rerun the initial mark process?

. Not needed.

Using without ploop

NOTE. OPTIMUMCACHE IS NO LONGER SUPPORTED.

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
```

Marking Directories

NOTE. OPTIMUMCACHE IS NO LONGER SUPPORTED.

Marking directories to be cached:

```
# occtl --mark-dir /path/to/site/on/filesystem --recursive
```

In common scenario admin marks for caching user directories:

```
# occtl --mark-dir /home /home2 /home3 --recursive
```

OptimumCache is going to index these directories. Thus system load during this period (from hours to days) might be as twice as high. You can check indexing job status with `'at -l'` at any time.

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
```

At the very end, for those changes to take effect:

```
$ occtl --check
```

‘occtl --check’ is the same lengthy operation as ‘marking’ is. Thus, it’s usage has to be sane, especially for big ‘home’ (>500G).

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[^]*/cPanelInsta ll
5	cpanel	^/home[^]*/cpeasyapach e
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[^]*/[^]*/.cp anel\$
13	cpanel	^/home[^]*/[^]*/.cp anel/.*
14	cpanel	^/home[^]*/[^]*/.ca gefs
15	cpanel	^/home[^]*/[^]*/.ca gefs/.*
16	cpanel	^/home[^]*/virtfs
17	cpanel	^/home[^]*/virtfs/.*
18	not_a_userdir	^/home/tmp/
19	not_a_userdir	^/home/tmp\$
20	not_a_userdir	^/home/ftp/
21	not_a_userdir	^/home/ftp\$
22	not_a_userdir	^/home/admin/
23	not_a_userdir	^/home/admin\$
24	quota	^/home[^]*/quota.user\$
25	usermisc	/quota.user\$
26	users_home	^/home/[^]*/backups\$

Continued on next page

Table 1 – continued from previous page

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/`

OptimumCache Configuration File

NOTE. OPTIMUMCACHE IS NO LONGER SUPPORTED.

`/etc/sysconfig/optimumcache`

`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, 10485760 (10MB) by default`

`# MAXSIZE`

`# Interval between caching attempts, default - 5 seconds`

`# TIMEOUT=7`

`# Adaptive timeout upper limit (seconds)`

`# MAXTIMEOUT=160`

`# Adaptive timeout multiplicator and divisor`

`# TIMEOUT_INCR_MUL=2`

`# TIMEOUT_DECR_DIV=4`


```
# Buffer size in KB for 'optimumcache dump', default is 32MB
# DUMP_BUFFER_SIZE=32000
# Extra space in %% of requested to purge, default 20%
# PURGEAHEAD=20
# Experimental: Eliminate frequent sync to address IO performance
NOIMMSYNC=1
# Logging verbosity, default - 1, verbose
# LOGLEVEL=1
# occtl -mark-dir or -check operations IO limit, MB/s, default is 5 MB/s
# OCCTL_LVE_IO_LIMIT=5
# occtl -mark-dir or -check operations %cpu limit, default is 50% of one CPU core
# OCCTL_LVE_SPEED_LIMIT=50
# Lve ID to associate limits with
# LVEID=5
# Collect perf statistics in /var/log/optimumcache_perf. Default is enabled.
# PERF_LOG_ENABLED=1
```


Command-line Interface

NOTE. OPTIMUMCACHE IS NO LONGER SUPPORTED.

OptimumCache is controlled using occtl command line utility.

usage:	occtl.py	[-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 [Path ...]] [-unmark-dir Path [Path ...]] [-recursive] [-add-skip-mask Regex] [-remove-skip-mask Id/Tag] [-list-skip-mask] [-silent] [-ignore-unmount-failure] [-no-lve-limits] [-foreground] [-ploop-status] [-remount-cached-points] [-purge] [-cancel-pending-jobs] [-report [Period]] [-recommend-minmax-size]
--------	----------	---

Display numbers/percents of cached files:

optimumcache stat

or

optimumcache stat /home

To display statistic for specific mount. In depth display what is being held in cache:

optimumcache dump [-resolve-filenames] [mount]

The option ‘-resolve-filenames’ is experimental and may not apply to all output cached entries.

Optional Arguments:

<code>-h, --help</code>	Show this help message and exit.
<code>--move-ploop param [param ...]</code>	Move cache from one ploop image to /path/to/new/image/location [New Size[KMGT]].
<code>--check</code>	Check marked files for errors. This task is scheduled as background job, unless <code>--foreground</code> is specified.
<code>--verbose</code>	List what is being checked.
<code>--init-ploop [param [param ...]]</code>	Create ploop image for the cache [/path/to/ploop/image [ploop_size] ploop_size] - if only one parameter is given, it is considered to be ploop size. Size should be a NUMBER[KMGT].
<code>--resize-ploop New Size</code>	New Size NUMBER[KMGT].
<code>--disable-ploop</code>	Disable ploop.
<code>--enable-ploop</code>	Enable ploop.
<code>--mount-ploop</code>	Mount ploop image.
<code>--unmount- ploop</code>	Unmount ploop image.
<code>--delete-ploop</code>	Delete ploop image. Implies disable ploop, if was enabled.
<code>--unmark-all</code>	Unmark all marked directories.
<code>--mark-dir Path [Path ...]</code>	Mark directory for caching. This task is scheduled as background job, unless <code>--foreground</code> is specified.
<code>--unmark-dir Path [Path ...]</code>	Unmark directory for caching.
<code>--recursive</code>	Is used with mark/unmark dir.
<code>--add-skip-mask Regex</code>	Regex to skip files/directories for caching.
<code>--remove-skip- mask Id Tag</code>	Remove regexp to skip files/directories by id or tag.
<code>--list-skip-mask</code>	List regexp to skip files/directories.
<code>--silent</code>	Do not echo status to stdout/syslog.
<code>--ignore- unmount-failure</code>	Ignore cannot unmount ploop problem.
<code>--no-lve-limits</code>	Ignore default LVE limits for <code>--mark-dir</code> and <code>--check</code> commands. Also implies <code>--foreground</code> .
<code>--foreground</code>	Don't spawn <code>--mark-dir</code> and <code>--check</code> commands in background.
<code>--ploop-status</code>	Check if ploop is mounted.
<code>--purge</code>	Purge cache storage (takes some time).
<code>--cancel- pending-jobs</code>	Cancel <code>--mark-dir</code> and <code>--check</code> commands if were queued or are being run in background.
<code>--report [Period]</code>	Report statistics for Period (hourly daily weekly monthly).

cloudlinux-collect: Collect System Load Statistics

NOTE. OPTIMUMCACHE IS NO LONGER SUPPORTED.

cloudlinux-collectl Quick Start

Installing this package automatically starts system load statistics collection in background. `cloudlinux-collectl` package has no strict dependency on OptimumCache, thus the statistics is collected regardless of whether OptimumCache is installed or not. The aim of having this package pre-installed is to compare system performance before and after installing OptimumCache, thus to measure OptimumCache effectiveness.

Install

```
# yum install cloudlinux-collect --enablerepo=cloudlinux-updates-testing
```

Note: cloudlinux-collect will be installed automatically on optimumcache upgrade to 0.2-23.

Measure Web Site Response Time

cloudlinux-collect can monitor response time for a configurable set of URLs.

Start monitoring new URL:

```
# cloudlinux-collect --addurl <alias> <http://url>
```

example:

```
# cloudlinux-collect --addurl localhost http://127.0.0.1/index.php
```

Try 'cloudlinux-collect --help' for more options.

To watch what is being collected

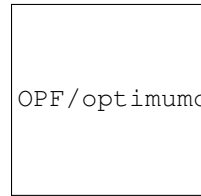
```
# cloudlinux-collect --test
```

Actual logs are compressed with gzip and kept in /var/log/optimumcache/collect directory.

Statistics Being Collected in Details

To monitor what statistics are being collected, try command:

```
# cloudlinux-collect --test
```



OPF/optimumcachecollect_zoom93.png

Along with common statistics blocks as CPU, disk usage, inodes cached, there are two blocks of data to watch how effectively OptimumCache is functioning.

'OPTIMUMCACHE DETAIL' refers to data, which is similar to output of command

```
# optimumcache stat
```

```
csums:      4964 (99.9%)
           fetched    uncached    cached
inodes:      4967      31         4936  (99.4%)
size:      204177    131072    73104  (35.8%)
RAM:         8        4         4    (50.0%)
```

Particularly, the last column percent numbers shall match.

The next goes URLSTATTRACKER DETAIL block with url response time in milliseconds. Negative values here may pop up unexpectedly. Negative numbers are not milliseconds, but signal about http error response code for that specific url. For instance, -403 will signal for 'Forbidden' http error. As for -500 value, it signals not only for 'Internal Server Error', but can be displayed, when there is connection problem with the server, which is specified by the url.

Statistics Manual Configuration

URLSTATTRACKER DETAIL is the only statistics, which requires manual configuration. Upon clean installation, it has only url_localhost preconfigured:

```
# cloudlinux-collect --info
```

url shortname	url
---------------	-----

localhost	http://localhost/
-----------	---

To add another url for monitoring:

```
# cloudlinux-collect --addurl alt http://192.168.0.102/
```

To display urls being monitored list:

```
# cloudlinux-collect --info
```

url shortname	url
---------------	-----

alt	http://192.168.0.102/
-----	---

localhost	http://localhost/
-----------	---

To skip URL from being tracked run command:

```
# cloudlinux-collect --skip <url short name>
```

Running Statistics Daemon: collectl-cloudlinux

cloudlinux-collectl has got collectl package as a dependency. Initd script /etc/init.d/cloudlinux-collectl will automatically bring up another instance of collectl named 'collectl-optimumcache'. collectl-optimumcache daemon instance has a separate config and does not interfere with other running pre-configure collectl daemon (if any).

As it was mentioned, collectl-optimumcache daemon starts automatically on package install, then on server restart events, kicked by regular Initd script /etc/init.d/cloudlinux-collectl. Thus, checking the daemon status, stop, restart is trivial:

```
# service cloudlinux-collect status
```

```
collectl-optimumcache (pid 1745) is running...
```

To start /stop:

```
# service cloudlinux-collect < start | stop >
```

Analyzing the Results

The statistics is being collected into files named `%hostname%-%datetime%.raw.gz` under directory `/var/log/cloudlinux-collect`

To convert those info format suitable for loading into Excel, LibreOffice Calc, another data mining tool, run the command:

```
# cloudlinux-collect --genplotfiles
```

Generate fresh plot files in

`/var/log/cloudlinux-collect/plotfiles`

Uninstall OptimumCache

NOTE. OPTIMUMCACHE IS NO LONGER SUPPORTED.

To uninstall OptimumCache run:

```
service optimumcache stop
```

```
occtl --delete-ploop
```

```
:>/var/share/optimumcache_store
```

```
yum remove optimumcache
```

If available: reboot server

After the reboot pfcache= mount options will disappear by themselves.

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.

If uninstall OptimumCache process lasts for too long, please find the solution in Troubleshooting section of this document.

Troubleshooting

NOTE. OPTIMUMCACHE IS NO LONGER SUPPORTED.

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
```

Yum fails to install Perl rpms coming with OptimumCache

If got this error with ‘yum install optimumcache’:

```
Error: Package: cloudlinux-collect-0.1-6.el6.noarch (cloudlinux-x86_64-server-6)
Requires: perl(Config::Tiny)
Error: Package: cloudlinux-collect-0.1-6.el6.noarch (cloudlinux-x86_64-server-6)
Requires: perl(IO::Socket::SSL)
Error: Package: cloudlinux-collect-0.1-6.el6.noarch (cloudlinux-x86_64-server-6)
Requires: perl(YAML::Tiny)
Error: Package: cloudlinux-collect-0.1-6.el6.noarch (cloudlinux-x86_64-server-6)
Requires: perl(IPC::Run)
You could try using --skip-broken to work around the problem
You could try running: rpm -Va --nofiles --nodigest
```

Most probably you have excluded “perl*” packages in /etc/yum.conf file, in this case to install OptimumCache run:

```
# yum install optimumcache --disableexcludes=all
```

OptimumCache prior 0.2-23: 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/sysconfig/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

Uninstalling OptimumCache lasts for too long

Uninstalling OptimumCache takes time because of files unmark process, which lasts proportionally to number of files, previously marked for caching with ‘occtl –mark-dir...’. If you see, that ‘yum remove optimumcache’ command is stuck and you have no time to wait for it to finish, or IO load, caused by unmarking files, is undesirable for you, open another console terminal and invoke:

```
# occtl --cancel-pending-jobs
```

This command will cancel unmark operation, being run by yum under the hood. So that yum uninstall package transaction will complete very soon.

‘Failed to attach peer: Invalid argument’ appears in syslog

Rather rare problem, try to forcibly update optimumcache_s with ploop status.

```
# occtl --remount-cached-points
```

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://h10032.www1.hp.com/ctg/Manual/c01754456
B120i/B320i Smart Array SATA RAID Controller	hpvsa	http://www8.hp.com/h20195/v2/GetPDF.aspx/c04168333.pdf
SanDisk DAS Cache		http://www.dell.com/en-us/work/learn/server-tech-nology-components-caching

CloudLinux Life cycle

CloudLinux supports the same end-of-life policy as RHEL. Using a supported operating system is critical to maintaining a stable server environment.

Currently Supported:

Operating System	Released	End of Life & Support
CloudLinux 7	Apr 1, 2015	Jun 30, 2024
CloudLinux 6	Feb 1, 2011	Nov 30, 2020
CloudLinux 5	Jan 1, 2010	Mar 31, 2017

Downloading Documentation

This documentation is available for download:

PDF - <http://docs.cloudlinux.com/cloudlinux.pdf>

ePub - <http://docs.cloudlinux.com/cloudlinux.epub>