CloudLinux REST API

API URL

The root URL for all REST services is: https://cln.cloudlinux.com/api

API overview

Authentication token is produced by combining: login|timestamp|sha1(secret_key + timestamp).

```
customer_login|1401189361|97e17a41ea904e0ca2bf03c7c36dee2
492ba89cd
```

The timestamp is the number of seconds since January 1, 1970 (UTC), or POSIX time.

Date/time formatting

All date/time values returned from API will be formatted as a string in ISO-8601: "yyyy-MM-ddTHH:mmZ" = 04-30T11:26-0400

Request query depends on the exact API implementation and may vary. Basically, you should use GET query arguments.

JSON API is called with URLs ending on *.json

All JSON responses will follow the next pattern:

```
{
    "success": true, // true - for success API execution or false - if
    something goes wrong
    "message": "", // Optional system message. Can describe request
    processing details (useful if error  occurs)
    "data": null, // This is value returning from API method.
}
```

TEXT API is called with URLs ending on *.plain

All PLAIN responses will follow the next pattern:

- First line “success:true” or “success:false”
- All other lines are API specific
- Optional "message:Some message" line can be available in any response

The success response will be like this:

```
success:true code:0
```

Error response sample:

```
success:false message:Unexpected
server error
```

Any *.plain API can return error response

Common API Methods

You can get here some additional information about Cloudlinux services.

/status.json

URL: https://cln.cloudlinux.com/api/status.json

Return system information about several Cloudlinux services.

Success response data (json object):

```
{
"db_clweb_connected": true, // true - if db clweb connected
"db_clweb_online": true, // true - if db clweb online
"db_rhn_connected": true, // true - if db rhn connected
"db_rhn_online": true, // true - if db rhn online
"rhn_overloaded": true, //true - if registration server overloaded
"rhn_queue_full": true, // true - if registration queue is full
"xmlrpc": true, // true - if our xmlrpc service is alive
"ip_server_reg": true, // true - if registration for CL IP servers allowed
(/usr/sbin/clnreg_ks --force)
}
```

/status.plain

URL: https://cln.cloudlinux.com/api/status.plain

Alternative to "check.json" with "text/plain" response. Return system information about several Cloudlinux services in the text representation.

Success response example:

```
success:true
db_clweb_connected:true, // true - if db clweb connected
db_clweb_online:false, // true - if db clweb online
db_rhn_connected:true, // true - if db rhn connected
db_rhn_online:true, // true - if db rhn online
rhn_overloaded:false, // true - if registration server overloaded
rhn_queue_full: true, // true - if registration queue is full
xmlrpc:false, // true - if our xmlrpc service is alive
ip_server_reg:false, // true - if registration for CL IP servers allowed
(/usr/sbin/clnreg_ks --force)
```

ELS Products API

API root URL: https://cln.cloudlinux.com/api/els

API for interactions with Extended Life Support products (CentOS/Ubuntu/Oracle).

ELS product codes

ELS key has **code** to define its product:

- CELS - "CentOS 6 ELS"
- OELS – "Oracle ELS"
- UELS16 – "Ubuntu 16 ELS"
- CELS_8 – "CentOS 8 ELS"
- UELS18 – "Ubuntu 18 ELS"
- PHP_ELS - "PHP ELS"
- PYTHON_ELS - "PYTHON ELS"
- UELS18_SLA - "Ubuntu 16 ELS with SLA"
- ELS CELS7 - “Extended Support for CentOS 7 Standard”
- ELS CELS7+SLA - “Extended Support for CentOS 7 Complete”

/key/create.json

URL:
https://cln.cloudlinux.com/api/els/key/create.json?token=AUTH_TOKEN&code=CELS&limit=2&note=12

GET arguments:
• **token** - authorization token
• **code** - ELS license code (see description above)

• **note** - key note
• **limit** - key limit (max servers per key) 0 for unlimited key

Create ELS key.

Success response data (json): Returns newly generated ELS key object

/key/update.json

```
{
"key": "CELS-yUDxBd5ethmGO5d3hnAZEau0",
"productCode": "CELS"
"usageLimit": "2"
"description": "123"
}
```

URL:
https://cln.cloudlinux.com/api/els/key/update.json?token=AUTH_TOKEN&key=CELS-key&limit=4&no

GET arguments:
- **token** - authorization token
- **key** - key to update
- **limit** - key limit (max servers per key) 0 for unlimited key
- **note** - key note

Update ELS key properties.

Success response data (json): Returns updated ELS key object

```
{
"key": "CELS-yUDxBd5ethmGO5d3hnAZEau0",
"productCode": "CELS", "usageLimit":
"4"
"description": "321"
}
```

/key/remove.json

URL:
https://cln.cloudlinux.com/api/els/key/remove.json?token=AUTH_TOKEN&key=CELS-key

GET arguments:

- **token** - authorization token
- **key** - ELS key to remove

Remove ELS registration key with all servers.

Success response data (json): Returns list of server objects removed with the current key.

```
[
{"key": "CELS-yUDxBd5ethmGO5d3hnAZEau0"}
]
```

/key/list.json

URL:
https://cln.cloudlinux.com/api/els/key/list.json?token=AUTH_TOKEN&productType=CELS

GET arguments:
- **token** - authorization token
- **code** – product code

List all ELS keys owned by customer.

Success response data (json): Returns list of key objects

/srv/list.json

```
[
{"key": "CELS-yUDxBd5ethmGO5d3hnAZEau0", "productCode":
"CELS", "usageLimit": "2", "description": 123}
{"key": "CELS-kdiRMfJ4yhmF7je4ldptndYk", "productCode": "CELS",
"usageLimit": "5",  "description": null}
]
```

URL:
https://cln.cloudlinux.com/api/els/srv/list.json?token=AUTH_TOKEN&productTypecode=CELS

GET arguments:
- token - authorization token
- code – product code

List all ELS servers under specific key

Success response data (json): Returns list of server objects or an empty list

```
[
{"id": "12345", "key": "CELS-value", "ip": "1.1.1.1", "hostname":
"test"}
{"id": "12346", "key": "CELS-value", "ip": "1.1.1.1", "hostname":
"test2"}
]
```

/srv/remove.json

URL:
https://cln.cloudlinux.com/api/els/key/remove.json?token=AUTH_TOKEN&key=CELS-value

GET arguments:
- **token** - authorization token
- **id** - server id to remove

Remove ELS server by its ID.

Success response data (json): Returns removed server object

```
{"id": "12345", "key": "CELS-value", "ip": "1.1.1.1", "hostname": "test"}
```

Imunify Product family API

API root URL: https://cln.cloudlinux.com/api/im

API for interactions with Imunify(AV+/360) software on client's side.

Imunify product codes

Imunify key has string-based **code** to define its product:
- AVP - "ImunifyAV+"
- 360_1 - "Imunify360 single user"
- 360_30 - "Imunify360 up to 30 users"
- 360_250 - "Imunify360 up to 250 users"
- 360_UN - "Imunify360 unlimited"

/key/create.json

URL:
https://cln.cloudlinux.com/api/im/key/create.json?token=AUTH_TOKEN&code=360_1&limit=2

GET arguments:
- **token** - authorization token
- **code** - Imunify license code (see description above)
- **note** - key note
- **limit** - key limit (max servers per key) 0 for unlimited key

Create Imunify key.

Success response data (json): Returns newly generated Imunify key object

```
{
"key": "IM1SOMENONSENCETEXT",
"limit": "0",
"code": "360_1"
}
```

/key/update.json

URL:
https://cln.cloudlinux.com/api/im/key/update.json?token=AUTH_TOKEN&key=IMKEYVALUE&limit=2

GET arguments:
- **token** - authorization token
- **key** - Imunify key to update
- **note** - New keynote (not updated if field is not present)
- **limit** - New key limit (max servers per key) 0 for unlimited key
- **servers** - Number of servers, registered with the key

Update Imunify key properties.

Success response data (json): Returns update Imunify key object

```
{
"key": "IM1SOMENONSENCETEXT",
"limit": "2",
"code": "360_250",
"servers": "1"
}
```

/key/remove.json

URL:
https://cln.cloudlinux.com/api/im/key/remove.json?token=AUTH_TOKEN&key=IMKEYVALUE

GET arguments:
- **token** - authorization token
- **key** - Imunify key to remove

Remove Imunify registration key with all servers.

Success response data (json): Returns list of server objects removed with current key.

```
[
{ "id": "SSERVER_ID_1", "key": "IMKEYVALUE"},
{ "id": "SSERVER_ID_2", "key": "IMKEYVALUE"},
]
```

/key/list.json

URL: https://cln.cloudlinux.com/api/im/key/list.json?token=AUTH_TOKEN

GET arguments:
- **token** - authorization token

List all Imunify keys owned by customer.

Success response data (json): Returns list of key objects

```
[
{"key": "IM1SOMENONSENCETEXT", "limit": "2", "code": "360_250",
"servers": "0"}, {"key": "IMUNOTHERKEY", "limit": "42", "code":
"360_UN", "servers": "1"},
]
```

/srv/list.json

URL:
https://cln.cloudlinux.com/api/im/srv/list.json?token=AUTH_TOKEN&key=IMKEYVALUE

GET arguments:
- **token** - authorization token

- **key** - Imunify key

List all Imunify servers under specific key

Success response data (json): Returns list of server objects or an empty list

```
[

{ "id": "SSERVER_ID_1", "ip": "SSERVER_IP_1", ", "key": "IMKEYVALUE",
"last_checkin": "2023-04-30T11:26-0400" },

{ "id": "SSERVER_ID_2", "ip": "SSERVER_IP_2", "key": "IMKEYVALUE",
"last_checkin": "2023-04-30T11:26-0400" },

]
```

/srv/remove.json

URL:
https://cln.cloudlinux.com/api/im/srv/remove.json?token=AUTH_TOKEN&id=SSERVER_ID_1

GET arguments:
- **token** - authorization token
- **id** - Imunify server id

Remove Imunify server by its ID.

Success response data (json): Returns removed server object

```
{ "id": "SSERVER_ID_1", "key": "IMKEYVALUE"},
```

/srv/convert.json

URL: https://cln.cloudlinux.com/api/im/srv/convert.json?token=AUTH_TOKEN&id=SSERVER_ID_1&key=IMUNNEWKEYVALUE

GET arguments:
- **token** - authorization token
- **id** - Imunify server id
- **key** - New Imunify key for provided server id

Convert (move) Imunify server to another key (license type). If you want to upgrade the server from "Single" to "Up to 30 users" license, you should move it from the current key

(which should be single user type) to anothwith another license type (up to 30 users in our example).

Success response data (json): Returns updated server object

```
{ "id": "SSERVER_ID_1", "key": "IMKEYVALUE"},
```

IP-based licenses API

API root URL: https://cln.cloudlinux.com/api/ipl

You can manage IP based license over this API.

IP license product types

To bind IP license with particular product you must provide valid product type:

- 1 - "CloudLinux OS"
- 4 – "CloudLinux Solo"
- 5 - "CloudLinux Admin"
- 1002 – "CloudLinux OS Shared Pro"
- 16 – "KernelCare"
- 17 – "KernelCare Plus"
- 40 – "ImunifyAV+"
- 41 - "Imunify360 single user"
- 42 - "Imunify360 up to 30 users"
- 43 - "Imunify360 up to 250 users"
- 49 - "Imunify360 unlimited"

/availability.json

URL example:
https://cln.cloudlinux.com/api/ipl/availability.json?ip=1.1.1.1&token=AUTH_TOKEN"

Will return information about what kind of license types are available for registration and what types are used by the current account.

GET arguments:
- **token** - authorization token
- **ip** - IP address to check

Success response data:

```
{"available": [1,41,42,43,49], "owned":[16]}
```

- **available(int[])** - list of types that can be used to register new IP license
- **owned(int[])** - list of types that already registered(owned) by this account

As you can see if somebody owns a license then that license type will not be in **available** list. If it is a currently owned license then the type of this license will be in the **owned** list.

/check.json

URL example:
https://cln.cloudlinux.com/api/ipl/check.json?ip=1.1.1.1&token=AUTH_TOKEN"

Check if the IP license is registered by any customer.

GET arguments:
- **token** - authorization token
- **ip** - IP address to check

Success response data (list of integers):

```
[1,16] OR [41] OR []
```

Will return a list of registered license types or empty list if provided IP is not registered yet.

/register.json

URL example:
https://cln.cloudlinux.com/api/ipl/register.json?ip=1.1.1.1&type=1&token=AUTH_TOKEN"

Will register an IP-based license for an authorized user.

GET arguments:
- **token** - authorization token
- **ip** - IP address to register
- **type** - IP license type (1,16,41,42,43,49)
- **els_allowed** – allow CL6 registration

On success response **returns** information about created or already registered license.

Success response data (json object):

```
{"ip": "1.1.1.1", "type": ,16 "registered": false, "created":
"2014-04-30T11:26-0400"}
```

- **ip(string)**
- **type(int)** - license type (1,16,41,42,43,49)
- **registered(boolean)** - true if the server was registered in CLN with this license (CLN licenses only).
- **created(string)** - license creation time

Will return a non-successful response in any other cases.

/convert.json

URL example: https://cln.cloudlinux.com/ipl/convert.json?ip=1.1.1.1&type=41&type_to=43&token=AUTH_TOKEN"

Change Imunify IP license type. Note that you can upgrade only to the Imunify360 IP license. You can upgrade/downgrade ImunifyIP licenses.

GET arguments:
- **token** - authorization token
- **ip** - Imunify 360 IP address
- **type** - current Imunify IP license type
- **type_to** - new Imunify 360 IP license type

Success response data (boolean):
- **true** when IP license was updated
- **false** if impossible to update IP license: not found, not owned by user etc.

/remove.json

URL example:
https://cln.cloudlinux.com/ipl/remove.json?ip=1.1.1.1&type=16&token=AUTH_TOKEN"

Will remove IP based license from authorized user licenses.

GET arguments:
- **token** - authorization token
- **ip** - IP address to remove
- **type**(optional) - IP license type. If empty - will remove licenses with all types

Success response data (boolean):
- **true** when IP license was removed
- **false** if IP license was not found OR not owned by user

/list.json

URL example: https://cln.cloudlinux.com/api/ipl/list.json?token=AUTH_TOKEN"

Return all IP licenses owned by an authorized user.

GET arguments:

- **token** - authorization token

Success response data (json objects list):

```
[{"ip": "1.1.1.1", "type": ,16 "registered": false, "created":
"2014-04-30T11:26- 0400"}, ...]
```

- **ip(string)**
- **type(int)** - license type (1,16,41,42,43,49)
- **registered(boolean)** - true if server was registered in CLN with this license (CLN licenses only).
- **created(string)** - license creation time

/server.json

URL example:
https://cln.cloudlinux.com/api/ipl/server.json?token=AUTH_TOKEN&ip=ip1,ip2"

Return all IP licenses owned by authorized user filtered by ip.

GET arguments:

- **token** - authorization token
- **ip** - array of server ips(comma separated). It can contain up to 100 IP addresses

Success response data (json objects list):

```
[{"server_info":"cloudlinux-release-5 (2.6.18-294.26.1.el5.lve0.8.18)","ip":
"1.1.1.1", "type": 1, "registered": false, "created": "2014-04-30T11:26-0400",
"last_checkin": "2014-04-30T11:26-0400"}, ...]
```

- **server_info(string)** - info about OS, its version and the running kernel of the server, registered by the IP license available)
- **ip(string)**
- **type(int)** - license type (1,16,41,42,43,49)
- **registered(boolean)** - true if server was registered in CLN with this license (CLN licenses only).
- **created(string)** - license creation time
- **last_checkin(string)** - license last_checkin time

/update.json

URL example:
https://cln.cloudlinux.com/api/ipl/update.json?ip=1.1.1.1&type=16&token=AUTH_TOKEN

Update the IP-based license from the authorized user licenses.

GET arguments:
- **token** - authorization token
- **ip** - IP address to fetch data

- **els_allowed** – allow CL6 registration
- **type (optional)** - IP license type. If empty - will remove licenses with all types

Success response data (boolean):
- **true** - when IP license was updated
- **false** - if IP license was not found OR not owned by user

KernelCare API

API root URL: https://cln.cloudlinux.com/api/kcare

You can manage KC licenses and keys over this API.

/key/create.json

URL: https://cln.cloudlinux.com/api/kcare/key/create.json?token=AUTH_TOKEN&limit=2&note=Key+description

Will generate a new KC key for an authorized user.

GET arguments:
- **token** - authorization token
- **limit** - key servers limit (0 for unlimited key)
- **note** (optional) - key description up to 100 characters

Success response data (string): returns newly generated KC key

/key/delete.json

Will delete KC key owned by authorized user.

GET arguments:
- **token** - authorization token
- **key** - KC key to delete

Success response data (boolean): returns true if key was deleted or false if key was not found

/key/list.json

URL: https://cln.cloudlinux.com/api/kcare/key/list.json?token=AUTH_TOKEN

Return list of all KC keys registered by authorized user.

GET arguments:

- **token** - authorization token

Success response data (json objects list):

```
[{
"key": "WsTs821nSAtiastD", // key identifier "enabled": false,
"created": "2014-04-30T11:26-0400", // key creation time "limit": 2,
// key servers max limit 0 for unlimited key "note": "Some custom
key note"
},// Will return an empty list if user has no keys
]
```

/key/servers.json

URL:
https://cln.cloudlinux.com/api/kcare/key/servers.json?token=AUTH_TOKEN&key=WsTs821nSAtiastD

Return list of servers registered with key owned by authorized user.

GET arguments:

- **token** - authorization token
- **key** - KC key linked to servers

Success response data (json objects list):

```
[{
    "server_id": "s096slnkaAtoiAsd", // Server identifier
    "ip": "1.1.1.1", // Remote IP from which server was registered
    "created": "2014-04-30T11:26-0400", // Registration time
    "last_checkin": "2023-04-30T11:26-0400" // Last check-in date
},
// Will return an empty list if no servers found
]
```

/key/set_cidr.json

URL:
https://cln.cloudlinux.com/api/kcare/key/set_cidr.json?token=AUTH_TOKEN&key=KEY&cidr=cidr+list

Will add a list of CIDRs for the authorized user.

GET arguments:

- **token** - authorization token
- **key** - key associated with cidr
- **cidr** - allowed IP range in a CIDR format. Multiple CIDRs can be separated with spaces, commas and sem

Success response data (json object):

```
{
    "code": 0 // CIDR adding status code
}
```

- code(int):
    - 0 - success (all CIDRs have been added)
    - 1 - irregular ip address in one of CIDRs
    - 2 - irregular prefix in one of CIDRs
    - 3 - internal error

/srv/remove.json

URL:
https://cln.cloudlinux.com/api/kcare/srv/remove.json?server_id={value}&token={value}

GET arguments:

- **token** - authorization token
- **server_id** - server id to remove

Success response data (json): Server found and deleted

```
{
    "message": null,
    "type": null,
    "success": true,
    "data": {
    "server_id": "",
    "key": ""
    }
}
```

Success response data (json): Server not found

```
{
    "message": "Server not found", "type":
    null,
    "success": true,
    "data": null
}
```

Error response (json): authorization failed

```
{
    "message": "Authorization failed", "type":
    null,
    "success": false, "data":
    null
}
```

KernelCare Nagios API

/nagios/register_key.plain

URL: https://cln.cloudlinux.com/api/kcare/nagios/register_key.plain?key=WsTs821nSAtiastD

API to register a custom monitoring key for the IP license. The key will be registered for the remote client IP.

GET arguments:

- **key** - Monitoring key from 16 to 32 alphanumeric characters (small &capital letters)

Success response example:

```
success:true code:0
```

Any nonzero code value means key creation error!

- **code(int):**
  - 0 - success (key created/updated)
  - 1 - wrong key size/for mat
  - 2 - no KC IP license available

Server Error example:

```
success:false
```

/nagios

URL: https://cln.cloudlinux.com/api/kcare/nagios/{key_id}

Will check status all servers registered by the key

- **key_id** - Server id to check

Return success response if all servers are up to date.

Success response example:

```
KernelCare OK - all servers are up to date!
```

Error response will be returned if any server is out of date, unsupported or inactive.

/nagios-res

URL: https://cln.cloudlinux.com/api/kcare/nagios-res/{login}/{token}

Will check the status of all servers registered to reseller

- **login** - Login of reseller
- **token** - authorization token

Will return success response if all servers are up to date.

Success response example:

```
KernelCare OK - all servers are up to date!
```

Error response will be returned if any server is out of date, unsupported or inactive.

Server Groups for Imunify/KernelCare API

API root URL: https://cln.cloudlinux.com/api/group

API to manage server groups for Imunify/KernelCare products.

/create.json

METHOD: POST

URL:
https://cln.cloudlinux.com/api/group/create.json?token=AUTH_TOKEN&name=NAME&description=DESCRIPTION

Will create a new group for the authorized user.

Query parameters:
- **token** - authorization token
- **name** - group name
- **description** (optional) – group description

Success response data (json): Returns newly generated group

```
{
    "message": null,
    "type": null,
    "success": true,
    "data": {
            "name": "NAME",
            "description": "DESCRIPTION"
    }
}
```

/list.json

METHOD: GET
URL: https://cln.cloudlinux.com/api/group/list.json?token=AUTH_TOKEN

Will receive all groups for authorized user.

Query parameter:
- **token** - authorization token

Success response data (json): Returns all groups

```
{
    "message": null,
    "type": null,
    "success": true,
    "data": [
        {
            "name": "NAME",
            "description": "DESCRIPTION"
        }
    ]
}
```

/servers/list.json

METHOD: GET
URL:
https://cln.cloudlinux.com/api/group/servers/list.json?token=AUTH_TOKEN&name=NAME

Will return a list of the server objects or an empty list.

Query parameters:
- **token** - authorization token
- **name** - group name

Success response data (json): Returns server objects


```
{
    "message": null,
    "type": null,
    "success": true,
    "data": [
        {
            "id": "SERVER-1",
            "hostName": "10.000.0.000",
            "ip": "10.000.0.000",
            "type": "IMUNIFY"
        }
    ]
}
```


/servers/put.json

METHOD: PUT
URL:
https://cln.cloudlinux.com/api/group/servers/put.json?token=AUTH_TOKEN&name=NAME&server_ids=SERVER-1,SERVER-2
Will add servers to a group.

Query parameters:
- **token** - authorization token
- **name** - group name
- **server_ids** – array of server ids

Success response data (json): Returns a list of added servers


```
{
    "message": null,
    "type": null,
    "success": true,
    "data": [
```

```
            {
                "id": "SERVER-1",
                "hostName": "10.000.0.000",
                "ip": "10.000.0.000",
                "type": "IMUNIFY"
            },
            {
                "id": "SERVER-1",
                "hostName": "10.000.0.000",
                "ip": "10.000.0.000",
                "type": "IMUNIFY"
            }
        ]
    }
```


/servers/remove.json

METHOD: DELETE
URL:
https://cln.cloudlinux.com/api/group/servers/remove.json?token=AUTH_TOKEN&name=NAME&server_ids=SERVER-1,SERVER-2

Will remove servers from a group.

Query parameters:

- **token** - authorization token
- **name** - group name
- **server_ids** – array of server ids

Success response data (json): Returns a list of removed servers


```
    {
        "message": null,
        "type": null,
        "success": true,
        "data": [
            {
                "id": "SERVER-1",
                "hostName": "10.000.0.000",
                "ip": "10.000.0.000",
                "type": "IMUNIFY"
            },
            {
                "id": "SERVER-1",
                "hostName": "10.000.0.000",
                "ip": "10.000.0.000",
                "type": "IMUNIFY"
            }
```

```
    ]
  }
```

AccelerateWP Billing Information

API to receive usage information for Accelerate Premium and CDN for a certain period.

/cl/awp

METHOD: GET

URL:

https://cln.cloudlinux.com/api/cl/awp?products=AWP_PREMIUM,AWP_CDN&yearMonth=2023-12&token=AUTH_TOKEN

GET arguments:

- **token** - authorization token
- **products** – product code [AWP_PREMIUM, AWP_CDN]
- **yearMonth** - look up period. Formal yyyy-MM

Returns entities that are billable for the given period.

Example response:

```
[
  {
    "product": "AWP_PREMIUM",
    "primary_domain": "example1.com".
    "username": "test-user1", // nullable
  },
  {
    "product": "AWP_CDN",
    "primary_domain": "example2.com".
    "username": "test-user2", // nullable
    "cdn_limit_type": "50 GB"
  }
]
```