
Hetzner Cloud Python Documentation

Release 1.16.0

Hetzner Cloud GmbH

Aug 17, 2021

Contents

1	Stable release	1
2	From sources	3
3	Samples	5
4	hcloud-python API	7
4.1	Main Interface	7
4.2	API Clients	9
4.2.1	ActionsClient	9
4.2.2	CertificateClient	10
4.2.3	DatacentersClient	14
4.2.4	FirewallsClient	15
4.2.5	Floating IPsClient	19
4.2.6	ImagesClient	24
4.2.7	ISOsClient	28
4.2.8	LoadBalancerTypesClient	29
4.2.9	LoadBalancerClient	30
4.2.10	LocationsClient	39
4.2.11	NetworksClient	40
4.2.12	PlacementGroupsClient	45
4.2.13	ServerTypesClient	47
4.2.14	ServersClient	48
4.2.15	SSHKeysClient	60
4.2.16	VolumesClient	62
4.3	Exceptions	66
5	Contributing	67
5.1	Types of Contributions	67
5.1.1	Report Bugs	67
5.1.2	Fix Bugs	67
5.1.3	Implement Features	67
5.1.4	Write Documentation	68
5.1.5	Submit Feedback	68
5.2	Get Started!	68
5.3	Pull Request Guidelines	69
5.4	Tips	69

5.5	How to release	69
6	History	71
6.1	v1.16.0 (2021-08-17)	71
6.2	v1.15.0 (2021-08-16)	71
6.3	v1.14.1 (2021-08-10)	71
6.4	v1.14.0 (2021-08-03)	71
6.5	v1.13.0 (2021-07-16)	71
6.6	v1.12.0 (2021-04-06)	72
6.7	v1.11.0 (2021-03-11)	72
6.8	v1.10.0 (2020-11-03)	72
6.9	v1.9.1 (2020-08-11)	72
6.10	v1.9.0 (2020-08-10)	72
6.11	v1.8.2 (2020-07-20)	72
6.12	v1.8.1 (2020-06-29)	73
6.13	1.8.0 (2020-06-22)	73
6.14	1.7.1 (2020-06-15)	73
6.15	1.7.0 (2020-06-05)	73
6.16	1.6.3 (2020-01-09)	73
6.17	1.6.2 (2019-10-15)	73
6.18	1.6.1 (2019-10-01)	73
6.19	1.6.0 (2019-09-17)	73
6.20	1.5.0 (2019-09-16)	74
6.21	1.4.1 (2019-08-19)	74
6.22	1.4.0 (2019-07-29)	74
6.23	1.3.0 (2019-07-10)	74
6.24	1.2.1 (2019-03-13)	74
6.25	1.2.0 (2019-03-06)	74
6.26	1.1.0 (2019-02-27)	74
6.27	1.0.1 (2019-02-22)	74
6.28	1.0.0 (2019-02-21)	75
6.29	0.1.0 (2018-12-20)	75
7	Hetzner Cloud Python	77
7.1	Examples	77
7.2	Create Server	77
7.3	List Servers	77
7.4	License	78
	Index	79

CHAPTER 1

Stable release

To install Hetzner Cloud Python, run this command in your terminal:

```
$ pip install hcloud
```

This is the preferred method to install Hetzner Cloud Python, as it will always install the most recent stable release.

If you don't have `pip` installed, [this Python installation guide](#) can guide you through the process.

CHAPTER 2

From sources

The sources for Hetzner Cloud Python can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/hetznercloud/hcloud-python
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/hetznercloud/hcloud-python/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```


To use Hetzner Cloud Python in a project:

```
from hcloud import Client
from hcloud.images.domain import Image
from hcloud.server_types.domain import ServerType

# Create a client
client = Client(token="project-token")

# Create 2 servers
# Create 2 servers
response1 = client.servers.create(
    "Server1",
    server_type=ServerType(name="cx11"),
    image=Image(id=4711)
)

response2 = client.servers.create(
    "Server2",
    server_type=ServerType(name="cx11"),
    image=Image(id=4711)
)

# Get all servers
server1 = response1.server
server2 = response2.server

servers = client.servers.get_all()

assert servers[0].id == server1.id
assert servers[1].id == server2.id

# Create 2 volumes

response1 = client.volumes.create(
    size=15,
```

(continues on next page)

(continued from previous page)

```
        name="Volume1",
        location=server1.location
    )
response2 = client.volumes.create(
    size=10,
    name="Volume2",
    location=server2.location
)

volume1 = response1.volume
volume2 = response2.volume

# Attach volume to server

volume1.attach(server1)
volume2.attach(server2)

# Detach second volume

volume2.detach()

# Poweroff 2nd server
server2.power_off()

# Poweroff 2nd server
server2.power_off()
```

More samples are in the repository: <https://github.com/hetznercloud/hcloud-python/tree/master/examples>.

4.1 Main Interface

```
class Client (token, api_endpoint='https://api.hetzner.cloud/v1', application_name=None, application_version=None, poll_interval=1)
```

Base Client for accessing the Hetzner Cloud API

```
actions = None
```

ActionsClient Instance

```
Type ActionsClient
```

```
certificates = None
```

CertificatesClient Instance

```
Type CertificatesClient
```

```
datacenters = None
```

DatacentersClient Instance

```
Type DatacentersClient
```

```
firewalls = None
```

FirewallsClient Instance

```
Type FirewallsClient
```

```
floating_ips = None
```

FloatingIPsClient Instance

```
Type FloatingIPsClient
```

```
images = None
```

ImagesClient Instance

```
Type ImagesClient
```

```
isos = None
```

ImagesClient Instance

Type *IsosClient*

load_balancer_types = None
LoadBalancerTypesClient Instance

Type *LoadBalancerTypesClient*

load_balancers = None
LoadBalancersClient Instance

Type *LoadBalancersClient*

locations = None
LocationsClient Instance

Type *LocationsClient*

networks = None
NetworksClient Instance

Type *NetworksClient*

placement_groups = None
PlacementGroupsClient Instance

Type *PlacementGroupsClient*

request (*method, url, tries=1, **kwargs*)
Perform a request to the Hetzner Cloud API, wrapper around requests.request

Parameters

- **method** – str HTTP Method to perform the Request
- **url** – str URL of the Endpoint
- **tries** – int Tries of the request (used internally, should not be set by the user)

Returns Response

Return type requests.Response

server_types = None
ServerTypesClient Instance

Type *ServerTypesClient*

servers = None
ServersClient Instance

Type *ServersClient*

ssh_keys = None
SSHKeysClient Instance

Type *SSHKeysClient*

volumes = None
VolumesClient Instance

Type *VolumesClient*

4.2 API Clients

4.2.1 ActionsClient

class ActionsClient (*client*)

get_all (*status=None, sort=None*)

Get all actions of the account

Parameters

- **status** – List[str] (optional) Response will have only actions with specified statuses. Choices: *running success error*
- **sort** – List[str] (optional) Specify how the results are sorted. Choices: *id command status progress started finished* . You can add one of “:asc”, “:desc” to modify sort order. (“:asc” is default)

Returns List[*BoundAction*]

get_by_id (*id*)

Get a specific action by its ID.

Parameters *id* – int

Returns *BoundAction*

get_list (*status=None, sort=None, page=None, per_page=None*)

Get a list of actions from this account

Parameters

- **status** – List[str] (optional) Response will have only actions with specified statuses. Choices: *running success error*
- **sort** – List[str] (optional) Specify how the results are sorted. Choices: *id command status progress started finished* . You can add one of “:asc”, “:desc” to modify sort order. (“:asc” is default)
- **page** – int (optional) Specifies the page to fetch
- **per_page** – int (optional) Specifies how many results are returned by page

Returns (List[*BoundAction*], *Meta*)

class BoundAction (*client, data={}, complete=True*)

model

alias of *hcloud.actions.domain.Action*

wait_until_finished (*max_retries=100*)

Wait until the specific action has status=“finished” (set Client.poll_interval to specify a delay between checks)

Parameters *max_retries* – int Specify how many retries will be performed before an ActionTimeoutException will be raised

Raises ActionFailedException when action is finished with status==“error”

Raises ActionTimeoutException when Action is still in “running” state after *max_retries* reloads.

```
class Action(id, command=None, status=None, progress=None, started=None, finished=None, re-  
sources=None, error=None)  
    Action Domain
```

Parameters

- **id** – int ID of an action
- **command** – Command executed in the action
- **status** – Status of the action
- **progress** – Progress of action in percent
- **started** – Point in time when the action was started
- **finished** (*datetime, None*) – Point in time when the action was finished. Only set if the action is finished otherwise None
- **resources** – Resources the action relates to
- **error** – Error message for the action if error occurred, otherwise None.

```
STATUS_ERROR = 'error'
```

Action Status error

```
STATUS_RUNNING = 'running'
```

Action Status running

```
STATUS_SUCCESS = 'success'
```

Action Status success

4.2.2 CertificateClient

```
class CertificatesClient(client)
```

```
create(name, certificate, private_key, labels=None)
```

Creates a new Certificate with the given name, certificate and private_key. This methods allows only creating custom uploaded certificates. If you want to create a managed certificate use `create_managed()`

Parameters

- **name** – str
- **certificate** – str Certificate and chain in PEM format, in order so that each record directly certifies the one preceding
- **private_key** – str Certificate key in PEM format
- **labels** – Dict[str, str] (optional) User-defined labels (key-value pairs)

Returns *BoundCertificate*

```
create_managed(name, domain_names, labels=None)
```

Creates a new managed Certificate with the given name and domain names. This methods allows only creating managed certificates for domains that are using the Hetzner DNS service. If you want to create a custom uploaded certificate use `create()`

Parameters

- **name** – str

- **domain_names** – List[str] Domains and subdomains that should be contained in the Certificate
- **labels** – Dict[str, str] (optional) User-defined labels (key-value pairs)

Returns *BoundCertificate*

get_actions (*certificate, status=None, sort=None*)

Returns all action objects for a Certificate.

Parameters

- **certificate** – *BoundCertificate* or *Certificate*
- **status** – List[str] (optional) Response will have only actions with specified statuses. Choices: *running success error*
- **sort** – List[str] (optional) Specify how the results are sorted. Choices: *id id:asc id:desc command command:asc command:desc status status:asc status:desc progress progress:asc progress:desc started started:asc started:desc finished finished:asc finished:desc*

Returns List[*BoundAction*]

get_actions_list (*certificate, status=None, sort=None, page=None, per_page=None*)

Returns all action objects for a Certificate.

Parameters

- **certificate** – *BoundCertificate* or *Certificate*
- **status** – List[str] (optional) Response will have only actions with specified statuses. Choices: *running success error*
- **sort** – List[str] (optional) Specify how the results are sorted. Choices: *id id:asc id:desc command command:asc command:desc status status:asc status:desc progress progress:asc progress:desc started started:asc started:desc finished finished:asc finished:desc*
- **page** – int (optional) Specifies the page to fetch
- **per_page** – int (optional) Specifies how many results are returned by page

Returns (List[*BoundAction*], *Meta*)

get_all (*name=None, label_selector=None*)

Get all certificates

Parameters

- **name** – str (optional) Can be used to filter certificates by their name.
- **label_selector** – str (optional) Can be used to filter certificates by labels. The response will only contain certificates matching the label selector.

Returns List[*BoundCertificate*]

get_by_id (*id*)

Get a specific certificate by its ID.

Parameters **id** – int

Returns *BoundCertificate*

get_by_name (*name*)

Get certificate by name

Parameters **name** – str Used to get certificate by name.

Returns *BoundCertificate*

get_list (*name=None, label_selector=None, page=None, per_page=None*)

Get a list of certificates

Parameters

- **name** – str (optional) Can be used to filter certificates by their name.
- **label_selector** – str (optional) Can be used to filter certificates by labels. The response will only contain certificates matching the label selector.
- **page** – int (optional) Specifies the page to fetch
- **per_page** – int (optional) Specifies how many results are returned by page

Returns (List[*BoundCertificate*], *Meta*)

retry_issuance (*certificate*)

Returns all action objects for a Certificate.

Parameters **certificate** – *BoundCertificate* or *Certificate*

Returns *BoundAction*

update (*certificate, name=None, labels=None*)

Updates a Certificate. You can update a certificate name and labels.

Parameters

- **certificate** – *BoundCertificate* or *Certificate*
- **name** – str (optional) New name to set
- **labels** – Dict[str, str] (optional) User-defined labels (key-value pairs)

Returns *BoundCertificate*

class BoundCertificate (*client, data, complete=True*)

delete ()

Deletes a certificate. :return: boolean

get_actions (*status=None, sort=None*)

Returns all action objects for a Certificate.

Parameters

- **status** – List[str] (optional) Response will have only actions with specified statuses. Choices: *running success error*
- **sort** – List[str] (optional) Specify how the results are sorted. Choices: *id id:asc id:desc command command:asc command:desc status status:asc status:desc progress progress:asc progress:desc started started:asc started:desc finished finished:asc finished:desc*

Returns List[*BoundAction*]

get_actions_list (*status=None, sort=None, page=None, per_page=None*)

Returns all action objects for a Certificate.

Parameters

- **status** – List[str] (optional) Response will have only actions with specified statuses. Choices: *running success error*
- **sort** – List[str] (optional) Specify how the results are sorted. Choices: *id id:asc id:desc command command:asc command:desc status status:asc status:desc progress progress:asc progress:desc started started:asc started:desc finished finished:asc finished:desc*
- **page** – int (optional) Specifies the page to fetch
- **per_page** – int (optional) Specifies how many results are returned by page

Returns (List[*BoundAction*], Meta)

model

alias of *hcloud.certificates.domain.Certificate*

retry_issuance ()

Retry a failed Certificate issuance or renewal. :return: BoundAction

update (*name=None, labels=None*)

Updates an certificate. You can update an certificate name and the certificate labels.

Parameters

- **name** – str (optional) New name to set
- **labels** – Dict[str, str] (optional) User-defined labels (key-value pairs)

Returns :class:BoundCertificate <hcloud.certificates.client.BoundCertificate>

class Certificate (*id=None, name=None, certificate=None, not_valid_before=None, not_valid_after=None, domain_names=None, fingerprint=None, created=None, labels=None, type=None, status=None*)

Certificate Domain

Parameters

- **id** – int ID of Certificate
- **name** – str Name of Certificate
- **certificate** – str Certificate and chain in PEM format, in order so that each record directly certifies the one preceding
- **not_valid_before** – datetime Point in time when the Certificate becomes valid
- **not_valid_after** – datetime Point in time when the Certificate becomes invalid
- **domain_names** – List[str] List of domains and subdomains covered by this certificate
- **fingerprint** – str Fingerprint of the Certificate
- **labels** – dict User-defined labels (key-value pairs)
- **created** – datetime Point in time when the certificate was created
- **type** – str Type of Certificate
- **status** – ManagedCertificateStatus Current status of a type managed Certificate, always none for type uploaded Certificates

4.2.3 DatacentersClient

class DatacentersClient (*client*)

get_all (*name=None*)

Get all datacenters

Parameters **name** – str (optional) Can be used to filter datacenters by their name.

Returns List[*BoundDatacenter*]

get_by_id (*id*)

Get a specific datacenter by its ID.

Parameters **id** – int

Returns *BoundDatacenter*

get_by_name (*name*)

Get datacenter by name

Parameters **name** – str Used to get datacenter by name.

Returns *BoundDatacenter*

get_list (*name=None, page=None, per_page=None*)

Get a list of datacenters

Parameters

- **name** – str (optional) Can be used to filter datacenters by their name.
- **page** – int (optional) Specifies the page to fetch
- **per_page** – int (optional) Specifies how many results are returned by page

Returns (List[*BoundDatacenter*], *Meta*)

class BoundDatacenter (*client, data*)

model

alias of *hcloud.datacenters.domain.Datacenter*

class Datacenter (*id=None, name=None, description=None, location=None, server_types=None*)

Datacenter Domain

Parameters

- **id** – int ID of Datacenter
- **name** – str Name of Datacenter
- **description** – str Description of Datacenter
- **location** – *BoundLocation*
- **server_types** – *DatacenterServerTypes*

class DatacenterServerTypes (*available, supported, available_for_migration*)

DatacenterServerTypes Domain

Parameters

- **available** – List[*BoundServerTypes*] All available server types for this datacenter
- **supported** – List[*BoundServerTypes*] All supported server types for this datacenter

- **available_for_migration** – List[BoundServerTypes] All available for migration (change type) server types for this datacenter

4.2.4 FirewallsClient

class FirewallsClient (*client*)

apply_to_resources (*firewall, resources*)
Applies one Firewall to multiple resources.

Parameters

- **firewall** – BoundFirewall or Firewall
- **resources** – List[FirewallResource]

Returns List[BoundAction]

create (*name, rules=None, labels=None, resources=None*)
Creates a new Firewall.

Parameters

- **name** – str Firewall Name
- **rules** – List[FirewallRule] (optional)
- **labels** – Dict[str, str] (optional) User-defined labels (key-value pairs)
- **resources** – List[FirewallResource] (optional)

Returns CreateFirewallResponse

delete (*firewall*)
Deletes a Firewall.

Parameters **firewall** – BoundFirewall or Firewall

Returns boolean

get_actions (*firewall, status=None, sort=None*)
Returns all action objects for a Firewall.

Parameters

- **firewall** – BoundFirewall or Firewall
- **status** – List[str] (optional) Response will have only actions with specified statuses. Choices: *running success error*
- **sort** – List[str] (optional) Specify how the results are sorted. Choices: *id id:asc id:desc command command:asc command:desc status status:asc status:desc progress progress:asc progress:desc started started:asc started:desc finished finished:asc finished:desc*

Returns List[BoundAction]

get_actions_list (*firewall, status=None, sort=None, page=None, per_page=None*)
Returns all action objects for a Firewall.

Parameters

- **firewall** – BoundFirewall or Firewall

- **status** – List[str] (optional) Response will have only actions with specified statuses. Choices: *running success error*
- **sort** – List[str] (optional) Specify how the results are sorted. Choices: *id id:asc id:desc command command:asc command:desc status status:asc status:desc progress progress:asc progress:desc started started:asc started:desc finished finished:asc finished:desc*
- **page** – int (optional) Specifies the page to fetch
- **per_page** – int (optional) Specifies how many results are returned by page

Returns (List[*BoundAction*], *Meta*)

get_all (*label_selector=None, name=None, sort=None*)

Get all floating ips from this account

Parameters

- **label_selector** – str (optional) Can be used to filter Firewalls by labels. The response will only contain Firewalls matching the label selector values.
- **name** – str (optional) Can be used to filter networks by their name.
- **sort** – List[str] (optional) Choices: *id name created* (You can add one of “:asc”, “:desc” to modify sort order. (“:asc” is default))

Returns List[*BoundFirewall*]

get_by_id (*id*)

Returns a specific Firewall object.

Parameters **id** – int

Returns *BoundFirewall*

get_by_name (*name*)

Get Firewall by name

Parameters **name** – str Used to get Firewall by name.

Returns *BoundFirewall*

get_list (*label_selector=None, page=None, per_page=None, name=None, sort=None*)

Get a list of floating ips from this account

Parameters

- **label_selector** – str (optional) Can be used to filter Firewalls by labels. The response will only contain Firewalls matching the label selector values.
- **page** – int (optional) Specifies the page to fetch
- **per_page** – int (optional) Specifies how many results are returned by page
- **name** – str (optional) Can be used to filter networks by their name.
- **sort** – List[str] (optional) Choices: *id name created* (You can add one of “:asc”, “:desc” to modify sort order. (“:asc” is default))

Returns (List[*BoundFirewall*], *Meta*)

remove_from_resources (*firewall, resources*)

Removes one Firewall from multiple resources.

Parameters

- **firewall** – *BoundFirewall* or *Firewall*
- **resources** – List[*FirewallResource*]

Returns List[*BoundAction*]

set_rules (*firewall, rules*)

Sets the rules of a Firewall. All existing rules will be overwritten. Pass an empty rules array to remove all rules.

Parameters

- **firewall** – *BoundFirewall* or *Firewall*
- **rules** – List[*FirewallRule*]

Returns List[*BoundAction*]

update (*firewall, labels=None, name=None*)

Updates the description or labels of a Firewall.

Parameters

- **firewall** – *BoundFirewall* or *Firewall*
- **labels** – Dict[str, str] (optional) User-defined labels (key-value pairs)
- **name** – str (optional) New name to set

Returns *BoundFirewall*

class BoundFirewall (*client, data, complete=True*)

apply_to_resources (*resources*)

Applies one Firewall to multiple resources. :param resources: List[*FirewallResource*] :return: List[*BoundAction*]

delete ()

Deletes a Firewall.

Returns boolean

get_actions (*status=None, sort=None*)

Returns all action objects for a Firewall.

Parameters

- **status** – List[str] (optional) Response will have only actions with specified statuses. Choices: *running success error*
- **sort** – List[str] (optional) Specify how the results are sorted. Choices: *id id:asc id:desc command command:asc command:desc status status:asc status:desc progress progress:asc progress:desc started started:asc started:desc finished finished:asc finished:desc*

Returns List[*BoundAction*]

get_actions_list (*status=None, sort=None, page=None, per_page=None*)

Returns all action objects for a Firewall.

Parameters

- **status** – List[str] (optional) Response will have only actions with specified statuses. Choices: *running success error*

- **sort** – List[str] (optional) Specify how the results are sorted. Choices: *id id:asc id:desc command command:asc command:desc status status:asc status:desc progress progress:asc progress:desc started started:asc started:desc finished finished:asc finished:desc*
- **page** – int (optional) Specifies the page to fetch
- **per_page** – int (optional) Specifies how many results are returned by page

Returns (List[*BoundAction*], Meta)

model

alias of *hcloud.firewalls.domain.Firewall*

remove_from_resources (*resources*)

Removes one Firewall from multiple resources. :param resources: List[*FirewallResource*] :return: List[*BoundAction*]

set_rules (*rules*)

Sets the rules of a Firewall. All existing rules will be overwritten. Pass an empty rules array to remove all rules. :param rules: List[*FirewallRule*] :return: List[*BoundAction*]

update (*name=None, labels=None*)

Updates the name or labels of a Firewall.

Parameters

- **labels** – Dict[str, str] (optional) User-defined labels (key-value pairs)
- **name** – str (optional) New Name to set

Returns *BoundFirewall*

class Firewall (*id=None, name=None, labels=None, rules=None, applied_to=None, created=None*)

Firewall Domain

Parameters

- **id** – int ID of the Firewall
- **name** – str Name of the Firewall
- **labels** – dict User-defined labels (key-value pairs)
- **rules** – List[*FirewallRule*] Rules of the Firewall
- **applied_to** – List[*FirewallResource*] Resources currently using the Firewall
- **created** – datetime Point in time when the image was created

class FirewallRule (*direction, protocol, source_ips, port=None, destination_ips=None, description=None*)

Firewall Rule Domain

Parameters

- **direction** – str The Firewall which was created
- **port** – str Port to which traffic will be allowed, only applicable for protocols TCP and UDP, specify port ranges by using - as a indicator, Sample: 80-85 means all ports between 80 & 85 (80, 82, 83, 84, 85)
- **protocol** – str Select traffic direction on which rule should be applied. Use *source_ips* for direction in and *destination_ips* for direction out.

- **source_ips** – List[str] List of permitted IPv4/IPv6 addresses in CIDR notation. Use 0.0.0.0/0 to allow all IPv4 addresses and ::/0 to allow all IPv6 addresses. You can specify 100 CIDRs at most.
- **destination_ips** – List[str] List of permitted IPv4/IPv6 addresses in CIDR notation. Use 0.0.0.0/0 to allow all IPv4 addresses and ::/0 to allow all IPv6 addresses. You can specify 100 CIDRs at most.
- **description** – str Short description of the firewall rule

DIRECTION_IN = 'in'
Firewall Rule Direction In

DIRECTION_OUT = 'out'
Firewall Rule Direction Out

PROTOCOL_ESP = 'esp'
Firewall Rule Protocol ESP

PROTOCOL_GRE = 'gre'
Firewall Rule Protocol GRE

PROTOCOL_ICMP = 'icmp'
Firewall Rule Protocol ICMP

PROTOCOL_TCP = 'tcp'
Firewall Rule Protocol TCP

PROTOCOL_UDP = 'udp'
Firewall Rule Protocol UDP

class FirewallResource (*type, server=None, label_selector=None*)
Firewall Used By Domain

Parameters

- **type** – str Type of resource referenced
- **server** – Optional[Server] Server the Firewall is applied to
- **label_selector** – Optional[FirewallResourceLabelSelector] Label Selector for Servers the Firewall should be applied to

TYPE_LABEL_SELECTOR = 'label_selector'
Firewall Used By Type label_selector

TYPE_SERVER = 'server'
Firewall Used By Type Server

class CreateFirewallResponse (*firewall, actions*)
Create Firewall Response Domain

Parameters

- **firewall** – *BoundFirewall* The Firewall which was created
- **actions** – List[*BoundAction*] The Action which shows the progress of the Firewall Creation

4.2.5 Floating IPsClient

class FloatingIPsClient (*client*)

assign (*floating_ip, server*)

Assigns a Floating IP to a server.

Parameters

- **floating_ip** – *BoundFloatingIP* or *FloatingIP*
- **server** – *BoundServer* or *Server* Server the Floating IP shall be assigned to

Returns *BoundAction*

change_dns_ptr (*floating_ip, ip, dns_ptr*)

Changes the hostname that will appear when getting the hostname belonging to this Floating IP.

Parameters

- **floating_ip** – *BoundFloatingIP* or *FloatingIP*
- **ip** – str The IP address for which to set the reverse DNS entry
- **dns_ptr** – str Hostname to set as a reverse DNS PTR entry, will reset to original default value if *None*

Returns *BoundAction*

change_protection (*floating_ip, delete=None*)

Changes the protection configuration of the Floating IP.

Parameters

- **floating_ip** – *BoundFloatingIP* or *FloatingIP*
- **delete** – boolean If true, prevents the Floating IP from being deleted

Returns *BoundAction*

create (*type, description=None, labels=None, home_location=None, server=None, name=None*)

Creates a new Floating IP assigned to a server.

Parameters

- **type** – str Floating IP type Choices: ipv4, ipv6
- **description** – str (optional)
- **labels** – Dict[str, str] (optional) User-defined labels (key-value pairs)
- **home_location** – *BoundLocation* or *Location* (Home location (routing is optimized for that location). Only optional if server argument is passed.
- **server** – *BoundServer* or *Server* Server to assign the Floating IP to
- **name** – str (optional)

Returns *CreateFloatingIPResponse*

delete (*floating_ip*)

Deletes a Floating IP. If it is currently assigned to a server it will automatically get unassigned.

Parameters **floating_ip** – *BoundFloatingIP* or *FloatingIP*

Returns boolean

get_actions (*floating_ip, status=None, sort=None*)

Returns all action objects for a Floating IP.

Parameters

- **floating_ip** – *BoundFloatingIP* or *FloatingIP*

- **status** – List[str] (optional) Response will have only actions with specified statuses. Choices: *running success error*
- **sort** – List[str] (optional) Specify how the results are sorted. Choices: *id id:asc id:desc command command:asc command:desc status status:asc status:desc progress progress:asc progress:desc started started:asc started:desc finished finished:asc finished:desc*

Returns List[*BoundAction*]

get_actions_list (*floating_ip, status=None, sort=None, page=None, per_page=None*)

Returns all action objects for a Floating IP.

Parameters

- **floating_ip** – *BoundFloatingIP* or *FloatingIP*
- **status** – List[str] (optional) Response will have only actions with specified statuses. Choices: *running success error*
- **sort** – List[str] (optional) Specify how the results are sorted. Choices: *id id:asc id:desc command command:asc command:desc status status:asc status:desc progress progress:asc progress:desc started started:asc started:desc finished finished:asc finished:desc*
- **page** – int (optional) Specifies the page to fetch
- **per_page** – int (optional) Specifies how many results are returned by page

Returns (List[*BoundAction*], *Meta*)

get_all (*label_selector=None, name=None*)

Get all floating ips from this account

Parameters

- **label_selector** – str (optional) Can be used to filter Floating IPs by labels. The response will only contain Floating IPs matching the label selector.able values.
- **name** – str (optional) Can be used to filter networks by their name.

Returns List[*BoundFloatingIP*]

get_by_id (*id*)

Returns a specific Floating IP object.

Parameters **id** – int

Returns *BoundFloatingIP*

get_by_name (*name*)

Get Floating IP by name

Parameters **name** – str Used to get Floating IP by name.

Returns *BoundFloatingIP*

get_list (*label_selector=None, page=None, per_page=None, name=None*)

Get a list of floating ips from this account

Parameters

- **label_selector** – str (optional) Can be used to filter Floating IPs by labels. The response will only contain Floating IPs matching the label selector.able values.
- **page** – int (optional) Specifies the page to fetch

- **per_page** – int (optional) Specifies how many results are returned by page
- **name** – str (optional) Can be used to filter networks by their name.

Returns (List[*BoundFloatingIP*], *Meta*)

unassign (*floating_ip*)

Unassigns a Floating IP, resulting in it being unreachable. You may assign it to a server again at a later time.

Parameters **floating_ip** – *BoundFloatingIP* or *FloatingIP*

Returns *BoundAction*

update (*floating_ip*, *description=None*, *labels=None*, *name=None*)

Updates the description or labels of a Floating IP.

Parameters

- **floating_ip** – *BoundFloatingIP* or *FloatingIP*
- **description** – str (optional) New Description to set
- **labels** – Dict[str, str] (optional) User-defined labels (key-value pairs)
- **name** – str (optional) New name to set

Returns *BoundFloatingIP*

class BoundFloatingIP (*client*, *data*, *complete=True*)

assign (*server*)

Assigns a Floating IP to a server.

Parameters **server** – *BoundServer* or *Server* Server the Floating IP shall be assigned to

Returns *BoundAction*

change_dns_ptr (*ip*, *dns_ptr*)

Changes the hostname that will appear when getting the hostname belonging to this Floating IP.

Parameters

- **ip** – str The IP address for which to set the reverse DNS entry
- **dns_ptr** – str Hostname to set as a reverse DNS PTR entry, will reset to original default value if *None*

Returns *BoundAction*

change_protection (*delete=None*)

Changes the protection configuration of the Floating IP.

Parameters **delete** – boolean If true, prevents the Floating IP from being deleted

Returns *BoundAction*

delete ()

Deletes a Floating IP. If it is currently assigned to a server it will automatically get unassigned.

Returns boolean

get_actions (*status=None*, *sort=None*)

Returns all action objects for a Floating IP.

Parameters

- **status** – List[str] (optional) Response will have only actions with specified statuses. Choices: *running success error*
- **sort** – List[str] (optional) Specify how the results are sorted. Choices: *id id:asc id:desc command command:asc command:desc status status:asc status:desc progress progress:asc progress:desc started started:asc started:desc finished finished:asc finished:desc*

Returns List[*BoundAction*]

get_actions_list (*status=None, sort=None, page=None, per_page=None*)

Returns all action objects for a Floating IP.

Parameters status –

List[str] (optional) Response will have only actions with specified statuses. Choices: *running success error*

param sort List[str] (optional) Specify how the results are sorted. Choices: *id id:asc id:desc command command:asc command:desc status status:asc status:desc progress progress:asc progress:desc started started:asc started:desc finished finished:asc finished:desc*

param page int (optional) Specifies the page to fetch

param per_page int (optional) Specifies how many results are returned by page

return (List[*BoundAction*], Meta)

model

alias of *hcloud.floating_ips.domain.FloatingIP*

unassign ()

Unassigns a Floating IP, resulting in it being unreachable. You may assign it to a server again at a later time.

Returns *BoundAction*

update (*description=None, labels=None, name=None*)

Updates the description or labels of a Floating IP.

Parameters

- **description** – str (optional) New Description to set
- **labels** – Dict[str, str] (optional) User-defined labels (key-value pairs)
- **name** – str (optional) New Name to set

Returns *BoundFloatingIP*

class FloatingIP (*id=None, type=None, description=None, ip=None, server=None, dns_ptr=None, home_location=None, blocked=None, protection=None, labels=None, created=None, name=None*)

Floating IP Domain

Parameters

- **id** – int ID of the Floating IP
- **description** – str, None Description of the Floating IP
- **ip** – str IP address of the Floating IP
- **type** – str Type of Floating IP. Choices: *ipv4, ipv6*

- **server** – *BoundServer*, None Server the Floating IP is assigned to, None if it is not assigned at all
- **dns_ptr** – List[Dict] Array of reverse DNS entries
- **home_location** – *BoundLocation* Location the Floating IP was created in. Routing is optimized for this location.
- **blocked** – boolean Whether the IP is blocked
- **protection** – dict Protection configuration for the Floating IP
- **labels** – dict User-defined labels (key-value pairs)
- **created** – datetime Point in time when the Floating IP was created
- **name** – str Name of the Floating IP

class CreateFloatingIPResponse (*floating_ip, action*)
 Create Floating IP Response Domain

Parameters

- **floating_ip** – *BoundFloatingIP* The Floating IP which was created
- **action** – *BoundAction* The Action which shows the progress of the Floating IP Creation

4.2.6 ImagesClient

class ImagesClient (*client*)

change_protection (*image, delete=None*)

Changes the protection configuration of the image. Can only be used on snapshots.

Parameters

- **image** – *BoundImage* or *Image*
- **delete** – bool If true, prevents the snapshot from being deleted

Returns *BoundAction*

delete (*image*)

Deletes an Image. Only images of type snapshot and backup can be deleted.

:param *BoundImage* or *Image* :return: bool

get_actions (*image, sort=None, status=None*)

Returns all action objects for an image.

Parameters

- **image** – *BoundImage* or *Image*
- **status** – List[str] (optional) Response will have only actions with specified statuses. Choices: *running success error*
- **sort** – List[str] (optional) Specify how the results are sorted. Choices: *id command status progress started finished* . You can add one of “:asc”, “:desc” to modify sort order. (“:asc” is default)

Returns List[*BoundAction*]

get_actions_list (*image, sort=None, page=None, per_page=None, status=None*)

Returns a list of action objects for an image.

Parameters

- **image** – *BoundImage* or *Image*
- **status** – List[str] (optional) Response will have only actions with specified statuses. Choices: *running success error*
- **sort** – List[str] (optional) Specify how the results are sorted. Choices: *id id:asc id:desc command command:asc command:desc status status:asc status:desc progress progress:asc progress:desc started started:asc started:desc finished finished:asc finished:desc*
- **page** – int (optional) Specifies the page to fetch
- **per_page** – int (optional) Specifies how many results are returned by page

Returns (List[*BoundAction*], *Meta*)

get_all (*name=None, label_selector=None, bound_to=None, type=None, sort=None, status=None, include_deprecated=None*)

Get all images

Parameters

- **name** – str (optional) Can be used to filter images by their name.
- **label_selector** – str (optional) Can be used to filter servers by labels. The response will only contain servers matching the label selector.
- **bound_to** – List[str] (optional) Server Id linked to the image. Only available for images of type backup
- **type** – List[str] (optional) Choices: *system snapshot backup*
- **status** – List[str] (optional) Can be used to filter images by their status. The response will only contain images matching the status.
- **sort** – List[str] (optional) Choices: *id name created* (You can add one of “:asc”, “:desc” to modify sort order. (“:asc” is default))
- **include_deprecated** – bool (optional) Include deprecated images in the response. Default: *False*

Returns List[*BoundImage*]

get_by_id (*id*)

Get a specific Image

Parameters *id* – int

Returns :class:BoundImage <hcloud.images.client.BoundImage

get_by_name (*name*)

Get image by name

Parameters *name* – str Used to get image by name.

Returns *BoundImage*

get_list (*name=None, label_selector=None, bound_to=None, type=None, sort=None, page=None, per_page=None, status=None, include_deprecated=None*)

Get all images

Parameters

- **name** – str (optional) Can be used to filter images by their name.
- **label_selector** – str (optional) Can be used to filter servers by labels. The response will only contain servers matching the label selector.
- **bound_to** – List[str] (optional) Server Id linked to the image. Only available for images of type backup
- **type** – List[str] (optional) Choices: system snapshot backup
- **status** – List[str] (optional) Can be used to filter images by their status. The response will only contain images matching the status.
- **sort** – List[str] (optional) Choices: id id:asc id:desc name name:asc name:desc created created:asc created:desc
- **include_deprecated** – bool (optional) Include deprecated images in the response. Default: False
- **page** – int (optional) Specifies the page to fetch
- **per_page** – int (optional) Specifies how many results are returned by page

Returns (List[*BoundImage*], *Meta*)

update (*image*, *description=None*, *type=None*, *labels=None*)

Updates the Image. You may change the description, convert a Backup image to a Snapshot Image or change the image labels.

Parameters

- **image** – *BoundImage* or *Image*
- **description** – str (optional) New description of Image
- **type** – str (optional) Destination image type to convert to Choices: snapshot
- **labels** – Dict[str, str] (optional) User-defined labels (key-value pairs)

Returns *BoundImage*

class BoundImage (*client*, *data*)

change_protection (*delete=None*)

Changes the protection configuration of the image. Can only be used on snapshots.

Parameters **delete** – bool If true, prevents the snapshot from being deleted

Returns *BoundAction*

delete ()

Deletes an Image. Only images of type snapshot and backup can be deleted.

Returns bool

get_actions (*sort=None*, *status=None*)

Returns all action objects for the image.

Parameters

- **status** – List[str] (optional) Response will have only actions with specified statuses. Choices: *running success error*
- **sort** – List[str] (optional) Specify how the results are sorted. Choices: *id id:asc id:desc command command:asc command:desc status status:asc status:desc progress*

progress:asc progress:desc started started:asc started:desc finished finished:asc finished:desc

Returns List[*BoundAction*]

get_actions_list (*sort=None, page=None, per_page=None, status=None*)

Returns a list of action objects for the image.

Parameters

- **status** – List[str] (optional) Response will have only actions with specified statuses. Choices: *running success error*
- **sort** – List[str] (optional) Specify how the results are sorted. Choices: *id id:asc id:desc command command:asc command:desc status status:asc status:desc progress progress:asc progress:desc started started:asc started:desc finished finished:asc finished:desc*
- **page** – int (optional) Specifies the page to fetch
- **per_page** – int (optional) Specifies how many results are returned by page

Returns (List[*BoundAction*], Meta)

model

alias of *hcloud.images.domain.Image*

update (*description=None, type=None, labels=None*)

Updates the Image. You may change the description, convert a Backup image to a Snapshot Image or change the image labels.

Parameters

- **description** – str (optional) New description of Image
- **type** – str (optional) Destination image type to convert to Choices: *snapshot*
- **labels** – Dict[str, str] (optional) User-defined labels (key-value pairs)

Returns *BoundImage*

class Image (*id=None, name=None, type=None, created=None, description=None, image_size=None, disk_size=None, deprecated=None, bound_to=None, os_flavor=None, os_version=None, rapid_deploy=None, created_from=None, protection=None, labels=None, status=None*)

Image Domain

Parameters

- **id** – int ID of the image
- **type** – str Type of the image Choices: *system, snapshot, backup, app*
- **status** – str Whether the image can be used or if it's still being created Choices: *available, creating*
- **name** – str, None Unique identifier of the image. This value is only set for system images.
- **description** – str Description of the image
- **image_size** – number, None Size of the image file in our storage in GB. For snapshot images this is the value relevant for calculating costs for the image.
- **disk_size** – number Size of the disk contained in the image in GB.
- **created** – datetime Point in time when the image was created

- **created_from** – *BoundServer*, None Information about the server the image was created from
- **bound_to** – *BoundServer*, None ID of server the image is bound to. Only set for images of type *backup*.
- **os_flavor** – str Flavor of operating system contained in the image Choices: *ubuntu, centos, debian, fedora, unknown*
- **os_version** – str, None Operating system version
- **rapid_deploy** – bool Indicates that rapid deploy of the image is available
- **protection** – dict Protection configuration for the image
- **deprecated** – datetime, None Point in time when the image is considered to be deprecated (in ISO-8601 format)
- **labels** – Dict User-defined labels (key-value pairs)

class CreateImageResponse (*action, image*)
 Create Image Response Domain

Parameters

- **image** – *BoundImage* The Image which was created
- **action** – *BoundAction* The Action which shows the progress of the Floating IP Creation

4.2.7 ISOsClient

class IsosClient (*client*)

get_all (*name=None*)
 Get all ISOs

Parameters **name** – str (optional) Can be used to filter ISOs by their name.

Returns List[*BoundIso*]

get_by_id (*id*)
 Get a specific ISO by its id

Parameters **id** – int

Returns *BoundIso*

get_by_name (*name*)
 Get iso by name

Parameters **name** – str Used to get iso by name.

Returns *BoundIso*

get_list (*name=None, page=None, per_page=None*)
 Get a list of ISOs

Parameters

- **name** – str (optional) Can be used to filter ISOs by their name.
- **page** – int (optional) Specifies the page to fetch
- **per_page** – int (optional) Specifies how many results are returned by page

Returns (List[*BoundIso*], *Meta*)

```
class BoundIso (client, data={}, complete=True)
```

model

alias of *hcloud.isos.domain.Iso*

```
class Iso (id=None, name=None, type=None, description=None, deprecated=None)
```

Iso Domain

Parameters

- **id** – int ID of the ISO
- **name** – str, None Unique identifier of the ISO. Only set for public ISOs
- **description** – str Description of the ISO
- **type** – str Type of the ISO. Choices: *public*, *private*
- **deprecated** – datetime, None ISO 8601 timestamp of deprecation, None if ISO is still available. After the deprecation time it will no longer be possible to attach the ISO to servers.

4.2.8 LoadBalancerTypesClient

```
class LoadBalancerTypesClient (client)
```

```
get_all (name=None)
```

Get all Load Balancer types

Parameters **name** – str (optional) Can be used to filter Load Balancer type by their name.

Returns List[*BoundLoadBalancerType*]

```
get_by_id (id)
```

Returns a specific Load Balancer Type.

Parameters **id** – int

Returns *BoundLoadBalancerType*

```
get_by_name (name)
```

Get Load Balancer type by name

Parameters **name** – str Used to get Load Balancer type by name.

Returns *BoundLoadBalancerType*

```
get_list (name=None, page=None, per_page=None)
```

Get a list of Load Balancer types

Parameters

- **name** – str (optional) Can be used to filter Load Balancer type by their name.
- **page** – int (optional) Specifies the page to fetch
- **per_page** – int (optional) Specifies how many results are returned by page

Returns (List[*BoundLoadBalancerType*], *Meta*)

```
class LoadBalancerType (id=None, name=None, description=None, max_connections=None,
                        max_services=None, max_targets=None, max_assigned_certificates=None,
                        prices=None)
```

LoadBalancerType Domain

Parameters

- **id** – int ID of the Load Balancer type
- **name** – str Name of the Load Balancer type
- **description** – str Description of the Load Balancer type
- **max_connections** – int Max amount of connections the Load Balancer can handle
- **max_services** – int Max amount of services the Load Balancer can handle
- **max_targets** – int Max amount of targets the Load Balancer can handle
- **max_assigned_certificates** – int Max amount of certificates the Load Balancer can serve
- **prices** – Dict Prices in different locations

4.2.9 LoadBalancerClient

```
class LoadBalancersClient (client)
```

```
add_service (load_balancer, service)
```

Adds a service to a Load Balancer.

Parameters

- **load_balancer** – *BoundLoadBalancer* or *LoadBalancer*
- **service** – *LoadBalancerService* The *LoadBalancerService* you want to add to the Load Balancer

Returns *BoundAction*

```
add_target (load_balancer, target)
```

Adds a target to a Load Balancer.

Parameters

- **load_balancer** – *BoundLoadBalancer* or *LoadBalancer*
- **target** – *LoadBalancerTarget* The *LoadBalancerTarget* you want to add to the Load Balancer

Returns *BoundAction*

```
attach_to_network (load_balancer, network, ip=None)
```

Attach a Load Balancer to a Network.

Parameters

- **load_balancer** – :class:‘ <hcloud.load_balancers.client.BoundLoadBalancer>‘ or *LoadBalancer*
- **network** – *BoundNetwork* or *Network*
- **ip** – str IP to request to be assigned to this Load Balancer

Returns *BoundAction*

change_algorithm (*load_balancer, algorithm*)

Changes the algorithm used by the Load Balancer

Parameters

- **load_balancer** – :class:‘ <hcloud.load_balancers.client.BoundLoadBalancer>‘ or *LoadBalancer*
- **algorithm** – *LoadBalancerAlgorithm* The LoadBalancerSubnet you want to add to the Load Balancer

Returns *BoundAction*

change_dns_ptr (*load_balancer, ip, dns_ptr*)

Changes the hostname that will appear when getting the hostname belonging to the public IPs (IPv4 and IPv6) of this Load Balancer.

Parameters

- **ip** – str The IP address for which to set the reverse DNS entry
- **dns_ptr** – str Hostname to set as a reverse DNS PTR entry, will reset to original default value if *None*

Returns *BoundAction*

change_protection (*load_balancer, delete=None*)

Changes the protection configuration of a Load Balancer.

Parameters

- **load_balancer** – :class:‘ <hcloud.load_balancers.client.BoundLoadBalancer>‘ or *LoadBalancer*
- **delete** – boolean If True, prevents the Load Balancer from being deleted

Returns *BoundAction*

change_type (*load_balancer, load_balancer_type*)

Changes the type of a Load Balancer.

Parameters

- **load_balancer** – *BoundLoadBalancer* or *LoadBalancer*
- **load_balancer_type** – *BoundLoadBalancerType* or *LoadBalancerType*
Load Balancer type the Load Balancer should migrate to

Returns *BoundAction*

create (*name, load_balancer_type, algorithm=None, services=None, targets=None, labels=None, location=None, network_zone=None, public_interface=None, network=None*)

Creates a Load Balancer .

Parameters

- **name** – str Name of the Load Balancer
- **load_balancer_type** – *LoadBalancerType* Type of the Load Balancer
- **labels** – Dict[str, str] (optional) User-defined labels (key-value pairs)
- **location** – Location Location of the Load Balancer
- **network_zone** – str Network Zone of the Load Balancer
- **algorithm** – *LoadBalancerAlgorithm* (optional) The algorithm the Load Balancer is currently using

- **services** – LoadBalancerService The services the Load Balancer is currently serving
- **targets** – LoadBalancerTarget The targets the Load Balancer is currently serving
- **public_interface** – bool Enable or disable the public interface of the Load Balancer
- **network** – Network Adds the Load Balancer to a Network

Returns CreateLoadBalancerResponse

delete (*load_balancer*)

Deletes a Load Balancer.

Parameters **load_balancer** – *BoundLoadBalancer* or *LoadBalancer*

Returns boolean

delete_service (*load_balancer*, *service*)

Deletes a service from a Load Balancer.

Parameters

- **load_balancer** – *BoundLoadBalancer* or *LoadBalancer*
- **service** – *LoadBalancerService* The LoadBalancerService you want to delete from the Load Balancer

Returns *BoundAction*

detach_from_network (*load_balancer*, *network*)

Detaches a Load Balancer from a Network.

Parameters

- **load_balancer** – :class:‘ <hcloud.load_balancers.client.BoundLoadBalancer>‘ or *LoadBalancer*
- **network** – *BoundNetwork* or *Network*

Returns *BoundAction*

disable_public_interface (*load_balancer*)

Disables the public interface of a Load Balancer.

Parameters **load_balancer** – :class:‘ <hcloud.load_balancers.client.BoundLoadBalancer>‘ or *LoadBalancer*

Returns *BoundAction*

enable_public_interface (*load_balancer*)

Enables the public interface of a Load Balancer.

Parameters **load_balancer** – :class:‘ <hcloud.load_balancers.client.BoundLoadBalancer>‘ or *LoadBalancer*

Returns *BoundAction*

get_actions (*load_balancer*, *status=None*, *sort=None*)

Returns all action objects for a Load Balancer.

Parameters

- **load_balancer** – *BoundLoadBalancer* or *LoadBalancer*
- **status** – List[str] (optional) Response will have only actions with specified statuses. Choices: *running success error*

- **sort** – List[str] (optional) Specify how the results are sorted. Choices: *id id:asc id:desc command command:asc command:desc status status:asc status:desc progress progress:asc progress:desc started started:asc started:desc finished finished:asc finished:desc*

Returns List[*BoundAction*]

get_actions_list (*load_balancer*, *status=None*, *sort=None*, *page=None*, *per_page=None*)
Returns all action objects for a Load Balancer.

Parameters

- **load_balancer** – *BoundLoadBalancer* or *LoadBalancer*
- **status** – List[str] (optional) Response will have only actions with specified statuses. Choices: *running success error*
- **sort** – List[str] (optional) Specify how the results are sorted. Choices: *id id:asc id:desc command command:asc command:desc status status:asc status:desc progress progress:asc progress:desc started started:asc started:desc finished finished:asc finished:desc*
- **page** – int (optional) Specifies the page to fetch
- **per_page** – int (optional) Specifies how many results are returned by page

Returns (List[*BoundAction*], *Meta*)

get_all (*name=None*, *label_selector=None*)
Get all Load Balancers from this account

Parameters

- **name** – str (optional) Can be used to filter Load Balancers by their name.
- **label_selector** – str (optional) Can be used to filter Load Balancers by labels. The response will only contain Load Balancers matching the label selector.

Returns List[*BoundLoadBalancer*]

get_by_id (*id*)
Get a specific Load Balancer

Parameters *id* – int

Returns *BoundLoadBalancer*

get_by_name (*name*)
Get Load Balancer by name

Parameters *name* – str Used to get Load Balancer by name.

Returns *BoundLoadBalancer*

get_list (*name=None*, *label_selector=None*, *page=None*, *per_page=None*)
Get a list of Load Balancers from this account

Parameters

- **name** – str (optional) Can be used to filter Load Balancers by their name.
- **label_selector** – str (optional) Can be used to filter Load Balancers by labels. The response will only contain Load Balancers matching the label selector.
- **page** – int (optional) Specifies the page to fetch
- **per_page** – int (optional) Specifies how many results are returned by page

Returns (List[*BoundLoadBalancer*], *Meta*)

remove_target (*load_balancer*, *target*)

Removes a target from a Load Balancer.

Parameters

- **load_balancer** – *BoundLoadBalancer* or *LoadBalancer*
- **target** – *LoadBalancerTarget* The *LoadBalancerTarget* you want to remove from the Load Balancer

Returns *BoundAction*

update (*load_balancer*, *name=None*, *labels=None*)

Updates a *LoadBalancer*. You can update a *LoadBalancer*'s name and a *LoadBalancer*'s labels.

Parameters

- **load_balancer** – *BoundLoadBalancer* or *LoadBalancer*
- **name** – str (optional) New name to set
- **labels** – Dict[str, str] (optional) User-defined labels (key-value pairs)

Returns *BoundLoadBalancer*

update_service (*load_balancer*, *service*)

Updates a service of an *Load Balancer*.

Parameters

- **load_balancer** – *BoundLoadBalancer* or *LoadBalancer*
- **service** – *LoadBalancerService* The *LoadBalancerService* with updated values within for the *Load Balancer*

Returns *BoundAction*

class BoundLoadBalancer (*client*, *data*, *complete=True*)

add_service (*service*)

Adds a service to a *Load Balancer*.

Parameters **service** – *LoadBalancerService* The *LoadBalancerService* you want to add to the *Load Balancer*

Returns *BoundAction*

add_target (*target*)

Adds a target to a *Load Balancer*.

Parameters **target** – *LoadBalancerTarget* The *LoadBalancerTarget* you want to add to the *Load Balancer*

Returns *BoundAction*

attach_to_network (*network*, *ip=None*)

Attaches a *Load Balancer* to a *Network*

Parameters

- **network** – *BoundNetwork* or *Network*
- **ip** – str IP to request to be assigned to this *Load Balancer*

Returns *BoundAction*

change_algorithm (*algorithm*)

Changes the algorithm used by the Load Balancer

Parameters **algorithm** – *LoadBalancerAlgorithm* The *LoadBalancerAlgorithm* you want to use

Returns *BoundAction*

change_dns_ptr (*ip, dns_ptr*)

Changes the hostname that will appear when getting the hostname belonging to the public IPs (IPv4 and IPv6) of this Load Balancer.

Parameters

- **ip** – str The IP address for which to set the reverse DNS entry
- **dns_ptr** – str Hostname to set as a reverse DNS PTR entry, will reset to original default value if *None*

Returns *BoundAction*

change_protection (*delete*)

Changes the protection configuration of a Load Balancer.

Parameters **delete** – boolean If True, prevents the Load Balancer from being deleted

Returns *BoundAction*

change_type (*load_balancer_type*)

Changes the type of a Load Balancer.

Parameters **load_balancer_type** – *BoundLoadBalancerType* or *LoadBalancerType* Load Balancer type the Load Balancer should migrate to

Returns *BoundAction*

delete ()

Deletes a Load Balancer.

Returns boolean

delete_service (*service*)

Deletes a service from a Load Balancer.

Parameters **service** – *LoadBalancerService* The *LoadBalancerService* you want to delete from the Load Balancer

Returns *BoundAction*

detach_from_network (*network*)

Detaches a Load Balancer from a Network.

Parameters **network** – *BoundNetwork* or *Network*

Returns *BoundAction*

disable_public_interface ()

Disables the public interface of a Load Balancer.

Returns *BoundAction*

enable_public_interface ()

Enables the public interface of a Load Balancer.

Returns *BoundAction*

get_actions (*status=None, sort=None*)

Returns all action objects for a Load Balancer.

Parameters

- **status** – List[str] (optional) Response will have only actions with specified statuses. Choices: *running success error*
- **sort** – List[str] (optional) Specify how the results are sorted. Choices: *id id:asc id:desc command command:asc command:desc status status:asc status:desc progress progress:asc progress:desc started started:asc started:desc finished finished:asc finished:desc*

Returns List[*BoundAction*]

get_actions_list (*status=None, sort=None, page=None, per_page=None*)

Returns all action objects for a Load Balancer.

Parameters

- **status** – List[str] (optional) Response will have only actions with specified statuses. Choices: *running success error*
- **sort** – List[str] (optional) Specify how the results are sorted. Choices: *id id:asc id:desc command command:asc command:desc status status:asc status:desc progress progress:asc progress:desc started started:asc started:desc finished finished:asc finished:desc*
- **page** – int (optional) Specifies the page to fetch
- **per_page** – int (optional) Specifies how many results are returned by page

Returns (List[*BoundAction*], *Meta*)

model

alias of *hcloud.load_balancers.domain.LoadBalancer*

remove_target (*target*)

Removes a target from a Load Balancer.

Parameters **target** – *LoadBalancerTarget* The *LoadBalancerTarget* you want to remove from the Load Balancer

Returns *BoundAction*

update (*name=None, labels=None*)

Updates a Load Balancer. You can update a Load Balancers name and a Load Balancers labels.

Parameters

- **name** – str (optional) New name to set
- **labels** – Dict[str, str] (optional) User-defined labels (key-value pairs)

Returns *BoundLoadBalancer*

update_service (*service*)

Updates a service of an Load Balancer.

Parameters **service** – *LoadBalancerService* The *LoadBalancerService* you want to update

Returns *BoundAction*

```
class LoadBalancer (id, name=None, public_net=None, private_net=None, location=None, algorithm=None, services=None, load_balancer_type=None, protection=None, labels=None, targets=None, created=None, outgoing_traffic=None, ingoing_traffic=None, included_traffic=None)
```

LoadBalancer Domain

Parameters

- **id** – int ID of the Load Balancer
- **name** – str Name of the Load Balancer (must be unique per project)
- **created** – datetime Point in time when the Load Balancer was created
- **protection** – dict Protection configuration for the Load Balancer
- **labels** – dict User-defined labels (key-value pairs)
- **location** – Location Location of the Load Balancer
- **public_net** – PublicNetwork Public network information.
- **private_net** – List[PrivateNet <hcloud.load_balancers.domain.PrivateNet] Private networks information.
- **algorithm** – LoadBalancerAlgorithm The algorithm the Load Balancer is currently using
- **services** – List[LoadBalancerService] The services the LoadBalancer is currently serving
- **targets** – LoadBalancerTarget The targets the LoadBalancer is currently serving
- **load_balancer_type** – LoadBalancerType The type of the Load Balancer
- **outgoing_traffic** – int, None Outbound Traffic for the current billing period in bytes
- **ingoing_traffic** – int, None Inbound Traffic for the current billing period in bytes
- **included_traffic** – int Free Traffic for the current billing period in bytes

```
class LoadBalancerService (protocol=None, listen_port=None, destination_port=None, proxyprotocol=None, health_check=None, http=None)
```

LoadBalancerService Domain

Parameters

- **protocol** – str Protocol of the service Choices: tcp, http, https
- **listen_port** – int Required when protocol is tcp, must be unique per Load Balancer.
- **destination_port** – int Required when protocol is tcp
- **proxyprotocol** – bool Enable proxyprotocol
- **health_check** – LoadBalancerHealthCheck Configuration for health checks
- **http** – LoadBalancerServiceHttp Configuration for http/https protocols, required when protocol is http/https

```
class LoadBalancerServiceHttp (cookie_name=None, cookie_lifetime=None, certificates=None, redirect_http=None, sticky_sessions=None)
```

LoadBalancerServiceHttp Domain

Parameters

- **cookie_name** – str Name of the cookie used for Session Stickness
- **cookie_lifetime** – str Lifetime of the cookie used for Session Stickness

- **certificates** – list IDs of the Certificates to use for TLS/SSL termination by the Load Balancer; empty for TLS/SSL passthrough or if protocol is “http”
- **redirect_http** – bool Redirect traffic from http port 80 to port 443
- **sticky_sessions** – bool Use sticky sessions. Only available if protocol is “http” or “https”.

class LoadBalancerHealthCheck (*protocol=None, port=None, interval=None, timeout=None, retries=None, http=None*)

LoadBalancerHealthCheck Domain

Parameters

- **protocol** – str Protocol of the service Choices: tcp, http, https
- **port** – int Port the healthcheck will be performed on
- **interval** – int Interval we trigger health check in
- **timeout** – int Timeout in sec after a try is assumed as timeout
- **retries** – int Retries we perform until we assume a target as unhealthy
- **http** – LoadBalancerHealthCheckHttp HTTP Config

class LoadBalancerHealthCheckHttp (*domain=None, path=None, response=None, status_codes=None, tls=None*)

LoadBalancerHealthCheckHttp Domain

Parameters

- **domain** – str Domain name to send in HTTP request. Can be null: In that case we will not send a domain name
- **path** – str HTTP Path send in Request
- **response** – str Optional HTTP response to receive in order to pass the health check
- **status_codes** – list List of HTTP status codes to receive in order to pass the health check
- **tls** – bool Type of health check

class LoadBalancerTarget (*type=None, server=None, label_selector=None, ip=None, use_private_ip=None*)

LoadBalancerTarget Domain

Parameters

- **type** – str Type of the resource, can be server or label_selector
- **server** – Server Target server
- **label_selector** – LoadBalancerTargetLabelSelector Target label selector
- **ip** – LoadBalancerTargetIP Target IP
- **use_private_ip** – bool use the private IP instead of primary public IP

class LoadBalancerTargetLabelSelector (*selector=None*)

LoadBalancerTargetLabelSelector Domain

Parameters **selector** – str Target label selector

class LoadBalancerTargetIP (*ip=None*)

LoadBalancerTargetIP Domain

Parameters `ip` – str Target IP

class `LoadBalancerAlgorithm` (*type=None*)
 LoadBalancerAlgorithm Domain

Parameters `type` – str Algorithm of the Load Balancer. Choices: round_robin, least_connections

4.2.10 LocationsClient

class `LocationsClient` (*client*)

get_all (*name=None*)
 Get all locations

Parameters `name` – str (optional) Can be used to filter locations by their name.

Returns List[*BoundLocation*]

get_by_id (*id*)
 Get a specific location by its ID.

Parameters `id` – int

Returns *BoundLocation*

get_by_name (*name*)
 Get location by name

Parameters `name` – str Used to get location by name.

Returns *BoundLocation*

get_list (*name=None, page=None, per_page=None*)
 Get a list of locations

Parameters

- `name` – str (optional) Can be used to filter locations by their name.
- `page` – int (optional) Specifies the page to fetch
- `per_page` – int (optional) Specifies how many results are returned by page

Returns (List[*BoundLocation*], *Meta*)

class `BoundLocation` (*client, data={}, complete=True*)

model
 alias of *hcloud.locations.domain.Location*

class `Location` (*id=None, name=None, description=None, country=None, city=None, latitude=None, longitude=None, network_zone=None*)
 Location Domain

Parameters

- `id` – int ID of location
- `name` – str Name of location
- `description` – str Description of location
- `country` – str ISO 3166-1 alpha-2 code of the country the location resides in

- **city** – str City the location is closest to
- **latitude** – float Latitude of the city closest to the location
- **longitude** – float Longitude of the city closest to the location
- **network_zone** – str Name of network zone this location resides in

4.2.11 NetworksClient

class NetworksClient (*client*)

add_route (*network, route*)

Adds a route entry to a network.

Parameters

- **network** – *BoundNetwork* or *Network*
- **route** – *NetworkRoute* The NetworkRoute you want to add to the Network

Returns *BoundAction*

add_subnet (*network, subnet*)

Adds a subnet entry to a network.

Parameters

- **network** – *BoundNetwork* or *Network*
- **subnet** – *NetworkSubnet* The NetworkSubnet you want to add to the Network

Returns *BoundAction*

change_ip_range (*network, ip_range*)

Changes the IP range of a network.

Parameters

- **network** – *BoundNetwork* or *Network*
- **ip_range** – str The new prefix for the whole network.

Returns *BoundAction*

change_protection (*network, delete=None*)

Changes the protection configuration of a network.

Parameters

- **network** – *BoundNetwork* or *Network*
- **delete** – boolean If True, prevents the network from being deleted

Returns *BoundAction*

create (*name, ip_range, subnets=None, routes=None, labels=None*)

Creates a network with range ip_range.

Parameters

- **name** – str Name of the network
- **ip_range** – str IP range of the whole network which must span all included subnets and route destinations

- **subnets** – List[*NetworkSubnet*] Array of subnets allocated
- **routes** – List[*NetworkRoute*] Array of routes set in this network
- **labels** – Dict[str, str] (optional) User-defined labels (key-value pairs)

Returns *BoundNetwork*

delete (*network*)

Deletes a network.

Parameters **network** – *BoundNetwork* or *Network*

Returns boolean

delete_route (*network*, *route*)

Removes a route entry to a network.

Parameters

- **network** – *BoundNetwork* or *Network*
- **route** – *NetworkRoute* The *NetworkRoute* you want to remove from the *Network*

Returns *BoundAction*

delete_subnet (*network*, *subnet*)

Removes a subnet entry from a network

Parameters

- **network** – *BoundNetwork* or *Network*
- **subnet** – *NetworkSubnet* The *NetworkSubnet* you want to remove from the *Network*

Returns *BoundAction*

get_actions (*network*, *status=None*, *sort=None*)

Returns all action objects for a network.

Parameters

- **network** – *BoundNetwork* or *Network*
- **status** – List[str] (optional) Response will have only actions with specified statuses. Choices: *running success error*
- **sort** – List[str] (optional) Specify how the results are sorted. Choices: *id id:asc id:desc command command:asc command:desc status status:asc status:desc progress progress:asc progress:desc started started:asc started:desc finished finished:asc finished:desc*

Returns List[*BoundAction*]

get_actions_list (*network*, *status=None*, *sort=None*, *page=None*, *per_page=None*)

Returns all action objects for a network.

Parameters

- **network** – *BoundNetwork* or *Network*
- **status** – List[str] (optional) Response will have only actions with specified statuses. Choices: *running success error*
- **sort** – List[str] (optional) Specify how the results are sorted. Choices: *id id:asc id:desc command command:asc command:desc status status:asc status:desc progress*

progress:asc progress:desc started started:asc started:desc finished finished:asc finished:desc

- **page** – int (optional) Specifies the page to fetch
- **per_page** – int (optional) Specifies how many results are returned by page

Returns (List[*BoundAction*], *Meta*)

get_all (*name=None, label_selector=None*)

Get all networks from this account

Parameters

- **name** – str (optional) Can be used to filter networks by their name.
- **label_selector** – str (optional) Can be used to filter networks by labels. The response will only contain networks matching the label selector.

Returns List[*BoundNetwork*]

get_by_id (*id*)

Get a specific network

Parameters **id** – int

Returns :class:`BoundNetwork <hcloud.networks.client.BoundNetwork>

get_by_name (*name*)

Get network by name

Parameters **name** – str Used to get network by name.

Returns *BoundNetwork*

get_list (*name=None, label_selector=None, page=None, per_page=None*)

Get a list of networks from this account

Parameters

- **name** – str (optional) Can be used to filter networks by their name.
- **label_selector** – str (optional) Can be used to filter networks by labels. The response will only contain networks matching the label selector.
- **page** – int (optional) Specifies the page to fetch
- **per_page** – int (optional) Specifies how many results are returned by page

Returns (List[*BoundNetwork*], *Meta*)

update (*network, name=None, labels=None*)

Updates a network. You can update a network's name and a network's labels.

Parameters

- **network** – *BoundNetwork* or *Network*
- **name** – str (optional) New name to set
- **labels** – Dict[str, str] (optional) User-defined labels (key-value pairs)

Returns *BoundNetwork*

class BoundNetwork (*client, data, complete=True*)

add_route (*route*)

Adds a route entry to a network.

Parameters **route** – *NetworkRoute* The NetworkRoute you want to add to the Network

Returns *BoundAction*

add_subnet (*subnet*)

Adds a subnet entry to a network.

Parameters **subnet** – *NetworkSubnet* The NetworkSubnet you want to add to the Network

Returns *BoundAction*

change_ip_range (*ip_range*)

Changes the IP range of a network.

Parameters **ip_range** – str The new prefix for the whole network.

Returns *BoundAction*

change_protection (*delete=None*)

Changes the protection configuration of a network.

Parameters **delete** – boolean If True, prevents the network from being deleted

Returns *BoundAction*

delete ()

Deletes a network.

Returns boolean

delete_route (*route*)

Removes a route entry to a network.

Parameters **route** – *NetworkRoute* The NetworkRoute you want to remove from the Network

Returns *BoundAction*

delete_subnet (*subnet*)

Removes a subnet entry from a network

Parameters **subnet** – *NetworkSubnet* The NetworkSubnet you want to remove from the Network

Returns *BoundAction*

get_actions (*status=None, sort=None*)

Returns all action objects for a network.

Parameters

- **status** – List[str] (optional) Response will have only actions with specified statuses. Choices: *running success error*
- **sort** – List[str] (optional) Specify how the results are sorted. Choices: *id id:asc id:desc command command:asc command:desc status status:asc status:desc progress progress:asc progress:desc started started:asc started:desc finished finished:asc finished:desc*

Returns List[*BoundAction*]

get_actions_list (*status=None, sort=None, page=None, per_page=None*)

Returns all action objects for a network.

Parameters

- **status** – List[str] (optional) Response will have only actions with specified statuses. Choices: *running success error*
- **sort** – List[str] (optional) Specify how the results are sorted. Choices: *id id:asc id:desc command command:asc command:desc status status:asc status:desc progress progress:asc progress:desc started started:asc started:desc finished finished:asc finished:desc*
- **page** – int (optional) Specifies the page to fetch
- **per_page** – int (optional) Specifies how many results are returned by page

Returns (List[*BoundAction*], Meta)

model

alias of *hcloud.networks.domain.Network*

update (*name=None, labels=None*)

Updates a network. You can update a network's name and a networks's labels.

Parameters

- **name** – str (optional) New name to set
- **labels** – Dict[str, str] (optional) User-defined labels (key-value pairs)

Returns *BoundNetwork*

class Network (*id, name=None, created=None, ip_range=None, subnets=None, routes=None, servers=None, protection=None, labels=None*)

Network Domain

Parameters

- **id** – int ID of the network
- **name** – str Name of the network
- **ip_range** – str IPv4 prefix of the whole network
- **subnets** – List[*NetworkSubnet*] Subnets allocated in this network
- **routes** – List[*NetworkRoute*] Routes set in this network
- **servers** – List[*BoundServer*] Servers attached to this network
- **protection** – dict Protection configuration for the network
- **labels** – dict User-defined labels (key-value pairs)

class NetworkSubnet (*ip_range, type=None, network_zone=None, gateway=None, vswitch_id=None*)

Network Subnet Domain

Parameters

- **type** – str Type of sub network.
- **ip_range** – str Range to allocate IPs from.
- **network_zone** – str Name of network zone.
- **gateway** – str Gateway for the route.
- **vswitch_id** – int ID of the vSwitch.

TYPE_CLOUD = 'cloud'

Subnet Type cloud

TYPE_SERVER = 'server'
 Subnet Type server, deprecated, use TYPE_CLOUD instead

TYPE_VSWITCH = 'vswitch'
 Subnet Type vSwitch

class NetworkRoute (*destination, gateway*)
 Network Route Domain

Parameters

- **destination** – str Destination network or host of this route.
- **gateway** – str Gateway for the route.

class CreateNetworkResponse (*network, action*)
 Create Network Response Domain

Parameters

- **network** – *BoundNetwork* The network which was created
- **action** – *BoundAction* The Action which shows the progress of the network Creation

4.2.12 PlacementGroupsClient

class PlacementGroupsClient (*client*)

create (*name, type, labels=None*)
 Creates a new Placement Group.

Parameters

- **name** – str Placement Group Name
- **type** – str Type of the Placement Group
- **labels** – Dict[str, str] (optional) User-defined labels (key-value pairs)

Returns *CreatePlacementGroupResponse*

delete (*placement_group*)
 Deletes a Placement Group.

Parameters **placement_group** – *BoundPlacementGroup* or *PlacementGroup*

Returns boolean

get_all (*label_selector=None, name=None, sort=None*)
 Get all Placement Groups

Parameters

- **label_selector** – str (optional) Can be used to filter Placement Groups by labels. The response will only contain Placement Groups matching the label selector values.
- **name** – str (optional) Can be used to filter Placement Groups by their name.
- **sort** – List[str] (optional) Choices: id name created (You can add one of “:asc”, “:desc” to modify sort order. (“:asc” is default))

Returns List[*BoundPlacementGroup*]

get_by_id (*id*)
 Returns a specific Placement Group object

Parameters `id` – int

Returns `BoundPlacementGroup`

get_by_name (*name*)

Get Placement Group by name

Parameters `name` – str Used to get Placement Group by name

Returns class:`BoundPlacementGroup` <`hcloud.placement_groups.client.BoundPlacementGroup`>

get_list (*label_selector=None, page=None, per_page=None, name=None, sort=None, type=None*)

Get a list of Placement Groups

Parameters

- **label_selector** – str (optional) Can be used to filter Placement Groups by labels. The response will only contain Placement Groups matching the label selector values.
- **page** – int (optional) Specifies the page to fetch
- **per_page** – int (optional) Specifies how many results are returned by page
- **name** – str (optional) Can be used to filter Placement Groups by their name.
- **sort** – List[str] (optional) Choices: id name created (You can add one of “:asc”, “:desc” to modify sort order. (“:asc” is default))

Returns (List[`BoundPlacementGroup`], Meta)

update (*placement_group, labels=None, name=None*)

Updates the description or labels of a Placement Group.

Parameters

- **placement_group** – `BoundPlacementGroup` or `PlacementGroup`
- **labels** – Dict[str, str] (optional) User-defined labels (key-value pairs)
- **name** – str (optional) New name to set

Returns `BoundPlacementGroup`

class BoundPlacementGroup (*client, data={}, complete=True*)

delete ()

Deletes a Placement Group

Returns boolean

model

alias of `hcloud.placement_groups.domain.PlacementGroup`

update (*labels=None, name=None*)

Updates the name or labels of a Placement Group

Parameters

- **labels** – Dict[str, str] (optional) User-defined labels (key-value pairs)
- **name** – str, (optional) New Name to set

Returns `BoundPlacementGroup`

class PlacementGroup (*id=None, name=None, labels=None, servers=None, type=None, created=None*)

Placement Group Domain

Parameters

- **id** – int ID of the Placement Group
- **name** – str Name of the Placement Group
- **labels** – dict User-defined labels (key-value pairs)
- **servers** – List[int] List of server IDs assigned to the Placement Group
- **type** – str Type of the Placement Group
- **created** – datetime Point in time when the image was created

class CreatePlacementGroupResponse (*placement_group, action*)

Create Placement Group Response Domain

Parameters

- **placement_group** – *BoundPlacementGroup* The Placement Group which was created
- **action** – *BoundAction* The Action which shows the progress of the Placement Group Creation

4.2.13 ServerTypesClient

class ServerTypesClient (*client*)

get_all (*name=None*)

Get all Server types

Parameters **name** – str (optional) Can be used to filter server type by their name.

Returns List[*BoundServerType*]

get_by_id (*id*)

Returns a specific Server Type.

Parameters **id** – int

Returns *BoundServerType*

get_by_name (*name*)

Get Server type by name

Parameters **name** – str Used to get Server type by name.

Returns *BoundServerType*

get_list (*name=None, page=None, per_page=None*)

Get a list of Server types

Parameters

- **name** – str (optional) Can be used to filter server type by their name.
- **page** – int (optional) Specifies the page to fetch
- **per_page** – int (optional) Specifies how many results are returned by page

Returns (List[*BoundServerType*], Meta)

class BoundServerType (*client, data={}, complete=True*)

model

alias of `hcloud.server_types.domain.ServerType`

class ServerType (*id=None, name=None, description=None, cores=None, memory=None, disk=None, prices=None, storage_type=None, cpu_type=None, deprecated=None*)
 ServerType Domain

Parameters

- **id** – int ID of the server type
- **name** – str Unique identifier of the server type
- **description** – str Description of the server type
- **cores** – int Number of cpu cores a server of this type will have
- **memory** – int Memory a server of this type will have in GB
- **disk** – int Disk size a server of this type will have in GB
- **prices** – Dict Prices in different locations
- **storage_type** – str Type of server boot drive. Local has higher speed. Network has better availability. Choices: *local, network*
- **cpu_type** – string Type of cpu. Choices: *shared, dedicated*
- **deprecated** – bool True if server type is deprecated

4.2.14 ServersClient

class ServersClient (*client*)

add_to_placement_group (*server, placement_group*)

Adds a server to a placement group.

Parameters

- **server** – *BoundServer* or *Server*
- **placement_group** – *BoundPlacementGroup* or *Network*

Returns *BoundAction*

attach_iso (*server, iso*)

Attaches an ISO to a server.

Parameters

- **server** – *BoundServer* or *Server*
- **iso** – *BoundIso* or *Server*

Returns *BoundAction*

attach_to_network (*server, network, ip=None, alias_ips=None*)

Attaches a server to a network

Parameters

- **server** – *BoundServer* or *Server*
- **network** – *BoundNetwork* or *Network*
- **ip** – str IP to request to be assigned to this server

- **alias_ips** – List[str] New alias IPs to set for this server.

Returns *BoundAction*

change_alias_ips (*server, network, alias_ips*)

Changes the alias IPs of an already attached network.

Parameters

- **server** – *BoundServer* or *Server*
- **network** – *BoundNetwork* or *Network*
- **alias_ips** – List[str] New alias IPs to set for this server.

Returns *BoundAction*

change_dns_ptr (*server, ip, dns_ptr*)

Changes the hostname that will appear when getting the hostname belonging to the primary IPs (ipv4 and ipv6) of this server.

Parameters

- **server** – *BoundServer* or *Server*
- **ip** – str The IP address for which to set the reverse DNS entry
- **dns_ptr** – Hostname to set as a reverse DNS PTR entry, will reset to original default value if *None*

Returns *BoundAction*

change_protection (*server, delete=None, rebuild=None*)

Changes the protection configuration of the server.

Parameters

- **server** – *BoundServer* or *Server*
- **delete** – boolean If true, prevents the server from being deleted (currently delete and rebuild attribute needs to have the same value)
- **rebuild** – boolean If true, prevents the server from being rebuilt (currently delete and rebuild attribute needs to have the same value)

Returns *BoundAction*

change_type (*server, server_type, upgrade_disk*)

Changes the type (Cores, RAM and disk sizes) of a server.

Parameters

- **server** – *BoundServer* or *Server*
- **server_type** – *BoundServerType* or *ServerType* Server type the server should migrate to
- **upgrade_disk** – boolean If false, do not upgrade the disk. This allows downgrading the server type later.

Returns *BoundAction*

create (*name, server_type, image, ssh_keys=None, volumes=None, firewalls=None, networks=None, user_data=None, labels=None, location=None, datacenter=None, start_after_create=True, automount=None, placement_group=None*)

Creates a new server. Returns preliminary information about the server as well as an action that covers progress of creation.

Parameters

- **name** – str Name of the server to create (must be unique per project and a valid hostname as per RFC 1123)
- **server_type** – *BoundServerType* or *ServerType* Server type this server should be created with
- **image** – *BoundImage* or *Image* Image the server is created from
- **ssh_keys** – List[*BoundSSHKey* or *SSHKey*] (optional) SSH keys which should be injected into the server at creation time
- **volumes** – List[*BoundVolume* or *Volume*] (optional) Volumes which should be attached to the server at the creation time. Volumes must be in the same location.
- **networks** – List[*BoundNetwork* or *Network*] (optional) Networks which should be attached to the server at the creation time.
- **user_data** – str (optional) Cloud-Init user data to use during server creation. This field is limited to 32KiB.
- **labels** – Dict[str,str] (optional) User-defined labels (key-value pairs)
- **location** – *BoundLocation* or *Location*
- **datacenter** – *BoundDatacenter* or *Datacenter*
- **start_after_create** – boolean (optional) Start Server right after creation. Defaults to True.
- **automount** – boolean (optional) Auto mount volumes after attach.
- **placement_group** – *BoundPlacementGroup* or *Location* Placement Group where server should be added during creation

Returns *CreateServerResponse*

create_image (*server*, *description=None*, *type=None*, *labels=None*)

Creates an image (snapshot) from a server by copying the contents of its disks.

Parameters

- **server** – *BoundServer* or *Server*
- **description** – str (optional) Description of the image. If you do not set this we auto-generate one for you.
- **type** – str (optional) Type of image to create (default: snapshot) Choices: snapshot, backup
- **labels** – Dict[str, str] User-defined labels (key-value pairs)

Returns *CreateImageResponse*

delete (*server*)

Deletes a server. This immediately removes the server from your account, and it is no longer accessible.

Parameters **server** – *BoundServer* or *Server*

Returns *BoundAction*

detach_from_network (*server*, *network*)

Detaches a server from a network.

Parameters

- **server** – *BoundServer* or *Server*
- **network** – *BoundNetwork* or *Network*

Returns *BoundAction*

detach_iso (*server*)

Detaches an ISO from a server.

Parameters **server** – *BoundServer* or *Server*

Returns *BoundAction*

disable_backup (*server*)

Disables the automatic backup option and deletes all existing Backups for a Server.

Parameters **server** – *BoundServer* or *Server*

Returns *BoundAction*

disable_rescue (*server*)

Disables the Hetzner Rescue System for a server.

Parameters **server** – *BoundServer* or *Server*

Returns *BoundAction*

enable_backup (*server*)

Enables and configures the automatic daily backup option for the server. Enabling automatic backups will increase the price of the server by 20%.

Parameters **server** – *BoundServer* or *Server*

Returns *BoundAction*

enable_rescue (*server*, *type=None*, *ssh_keys=None*)

Enable the Hetzner Rescue System for this server.

Parameters

- **server** – *BoundServer* or *Server*
- **type** – str Type of rescue system to boot (default: linux64) Choices: linux64, linux32, freebsd64
- **ssh_keys** – List[str] Array of SSH key IDs which should be injected into the rescue system. Only available for types: linux64 and linux32.

Returns *EnableRescueResponse*

get_actions (*server*, *status=None*, *sort=None*)

Returns all action objects for a server.

Parameters

- **server** – *BoundServer* or *Server*
- **status** – List[str] (optional) Response will have only actions with specified statuses. Choices: *running success error*
- **sort** – List[str] (optional) Specify how the results are sorted. Choices: *id id:asc id:desc command command:asc command:desc status status:asc status:desc progress progress:asc progress:desc started started:asc started:desc finished finished:asc finished:desc*

Returns List[*BoundAction*]

get_actions_list (*server, status=None, sort=None, page=None, per_page=None*)

Returns all action objects for a server.

Parameters

- **server** – *BoundServer* or *Server*
- **status** – List[str] (optional) Response will have only actions with specified statuses. Choices: *running success error*
- **sort** – List[str] (optional) Specify how the results are sorted. Choices: *id id:asc id:desc command command:asc command:desc status status:asc status:desc progress progress:asc progress:desc started started:asc started:desc finished finished:asc finished:desc*
- **page** – int (optional) Specifies the page to fetch
- **per_page** – int (optional) Specifies how many results are returned by page

Returns (List[*BoundAction*], *Meta*)

get_all (*name=None, label_selector=None, status=None*)

Get all servers from this account

Parameters

- **name** – str (optional) Can be used to filter servers by their name.
- **label_selector** – str (optional) Can be used to filter servers by labels. The response will only contain servers matching the label selector.
- **status** – List[str] (optional) Can be used to filter servers by their status. The response will only contain servers matching the status.

Returns List[*BoundServer*]

get_by_id (*id*)

Get a specific server

Parameters **id** – int

Returns *BoundServer*

get_by_name (*name*)

Get server by name

Parameters **name** – str Used to get server by name.

Returns *BoundServer*

get_list (*name=None, label_selector=None, page=None, per_page=None, status=None*)

Get a list of servers from this account

Parameters

- **name** – str (optional) Can be used to filter servers by their name.
- **label_selector** – str (optional) Can be used to filter servers by labels. The response will only contain servers matching the label selector.
- **status** – List[str] (optional) Can be used to filter servers by their status. The response will only contain servers matching the status.
- **page** – int (optional) Specifies the page to fetch
- **per_page** – int (optional) Specifies how many results are returned by page

Returns (List[*BoundServer*], *Meta*)

power_off (*server*)

Cuts power to the server. This forcefully stops it without giving the server operating system time to gracefully stop

Parameters **server** – *BoundServer* or *Server*

Returns *BoundAction*

power_on (*server*)

Starts a server by turning its power on.

Parameters **server** – *BoundServer* or *Server*

Returns *BoundAction*

reboot (*server*)

Reboots a server gracefully by sending an ACPI request.

Parameters **server** – *BoundServer* or *Server*

Returns *BoundAction*

rebuild (*server*, *image*)

Rebuilds a server overwriting its disk with the content of an image, thereby destroying all data on the target server.

Parameters

- **server** – *BoundServer* or *Server*
- **image** – *BoundImage* or *Image*

Returns *BoundAction*

remove_from_placement_group (*server*)

Removes a server from a placement group.

Parameters **server** – *BoundServer* or *Server*

Returns *BoundAction*

request_console (*server*)

Requests credentials for remote access via vnc over websocket to keyboard, monitor, and mouse for a server.

Parameters **server** – *BoundServer* or *Server*

Returns *RequestConsoleResponse*

reset (*server*)

Cuts power to a server and starts it again.

Parameters **server** – *BoundServer* or *Server*

Returns *BoundAction*

reset_password (*server*)

Resets the root password. Only works for Linux systems that are running the qemu guest agent.

Parameters **server** – *BoundServer* or *Server*

Returns *ResetPasswordResponse*

shutdown (*server*)

Shuts down a server gracefully by sending an ACPI shutdown request.

Parameters `server` – *BoundServer* or *Server*

Returns *BoundAction*

update (*server*, *name=None*, *labels=None*)

Updates a server. You can update a server's name and a server's labels.

Parameters

- **server** – *BoundServer* or *Server*
- **name** – str (optional) New name to set
- **labels** – Dict[str, str] (optional) User-defined labels (key-value pairs)

Returns *BoundServer*

class BoundServer (*client*, *data*, *complete=True*)

add_to_placement_group (*placement_group*)

Adds a server to a placement group.

Parameters `placement_group` – *BoundPlacementGroup* or *Network*

Returns *BoundAction*

attach_iso (*iso*)

Attaches an ISO to a server.

Parameters `iso` – *BoundIso* or *Server*

Returns *BoundAction*

attach_to_network (*network*, *ip=None*, *alias_ips=None*)

Attaches a server to a network

Parameters

- **network** – *BoundNetwork* or *Network*
- **ip** – str IP to request to be assigned to this server
- **alias_ips** – List[str] New alias IPs to set for this server.

Returns *BoundAction*

change_alias_ips (*network*, *alias_ips*)

Changes the alias IPs of an already attached network.

Parameters

- **network** – *BoundNetwork* or *Network*
- **alias_ips** – List[str] New alias IPs to set for this server.

Returns *BoundAction*

change_dns_ptr (*ip*, *dns_ptr*)

Changes the hostname that will appear when getting the hostname belonging to the primary IPs (ipv4 and ipv6) of this server.

Parameters

- **ip** – str The IP address for which to set the reverse DNS entry
- **dns_ptr** – Hostname to set as a reverse DNS PTR entry, will reset to original default value if *None*

Returns *BoundAction*

change_protection (*delete=None, rebuild=None*)

Changes the protection configuration of the server.

Parameters

- **server** – *BoundServer* or *Server*
- **delete** – boolean If true, prevents the server from being deleted (currently delete and rebuild attribute needs to have the same value)
- **rebuild** – boolean If true, prevents the server from being rebuilt (currently delete and rebuild attribute needs to have the same value)

Returns *BoundAction*

change_type (*server_type, upgrade_disk*)

Changes the type (Cores, RAM and disk sizes) of a server.

Parameters

- **server_type** – *BoundServerType* or *ServerType* Server type the server should migrate to
- **upgrade_disk** – boolean If false, do not upgrade the disk. This allows downgrading the server type later.

Returns *BoundAction*

create_image (*description=None, type=None, labels=None*)

Creates an image (snapshot) from a server by copying the contents of its disks.

Parameters

- **description** – str (optional) Description of the image. If you do not set this we auto-generate one for you.
- **type** – str (optional) Type of image to create (default: snapshot) Choices: snapshot, backup
- **labels** – Dict[str, str] User-defined labels (key-value pairs)

Returns *CreateImageResponse*

delete ()

Deletes a server. This immediately removes the server from your account, and it is no longer accessible.

Returns *BoundAction*

detach_from_network (*network*)

Detaches a server from a network.

Parameters **network** – *BoundNetwork* or *Network*

Returns *BoundAction*

detach_iso ()

Detaches an ISO from a server.

Returns *BoundAction*

disable_backup ()

Disables the automatic backup option and deletes all existing Backups for a Server.

Returns *BoundAction*

disable_rescue ()

Disables the Hetzner Rescue System for a server.

Returns *BoundAction*

enable_backup ()

Enables and configures the automatic daily backup option for the server. Enabling automatic backups will increase the price of the server by 20%.

Returns *BoundAction*

enable_rescue (*type=None, ssh_keys=None*)

Enable the Hetzner Rescue System for this server.

Parameters

- **type** – str Type of rescue system to boot (default: linux64) Choices: linux64, linux32, freebsd64
- **ssh_keys** – List[str] Array of SSH key IDs which should be injected into the rescue system. Only available for types: linux64 and linux32.

Returns *EnableRescueResponse*

get_actions (*status=None, sort=None*)

Returns all action objects for a server.

Parameters

- **status** – List[str] (optional) Response will have only actions with specified statuses. Choices: *running success error*
- **sort** – List[str] (optional) Specify how the results are sorted. Choices: *id id:asc id:desc command command:asc command:desc status status:asc status:desc progress progress:asc progress:desc started started:asc started:desc finished finished:asc finished:desc*

Returns List[*BoundAction*]

get_actions_list (*status=None, sort=None, page=None, per_page=None*)

Returns all action objects for a server.

Parameters

- **status** – List[str] (optional) Response will have only actions with specified statuses. Choices: *running success error*
- **sort** – List[str] (optional) Specify how the results are sorted. Choices: *id id:asc id:desc command command:asc command:desc status status:asc status:desc progress progress:asc progress:desc started started:asc started:desc finished finished:asc finished:desc*
- **page** – int (optional) Specifies the page to fetch
- **per_page** – int (optional) Specifies how many results are returned by page

Returns (List[*BoundAction*], Meta)

model

alias of *hcloud.servers.domain.Server*

power_off ()

Cuts power to the server. This forcefully stops it without giving the server operating system time to gracefully stop

Returns *BoundAction*

power_on()

Starts a server by turning its power on.

Returns *BoundAction*

reboot()

Reboots a server gracefully by sending an ACPI request.

Returns *BoundAction*

rebuild(*image*)

Rebuilds a server overwriting its disk with the content of an image, thereby destroying all data on the target server.

Parameters **image** – *BoundImage* or *Image*

Returns *BoundAction*

remove_from_placement_group()

Removes a server from a placement group.

Returns *BoundAction*

request_console()

Requests credentials for remote access via vnc over websocket to keyboard, monitor, and mouse for a server.

Returns *RequestConsoleResponse*

reset()

Cuts power to a server and starts it again.

Returns *BoundAction*

reset_password()

Resets the root password. Only works for Linux systems that are running the qemu guest agent.

Returns *ResetPasswordResponse*

shutdown()

Shuts down a server gracefully by sending an ACPI shutdown request.

Returns *BoundAction*

update(*name=None, labels=None*)

Updates a server. You can update a server's name and a server's labels.

Parameters

- **name** – str (optional) New name to set
- **labels** – Dict[str, str] (optional) User-defined labels (key-value pairs)

Returns *BoundServer*

```
class Server(id, name=None, status=None, created=None, public_net=None, server_type=None,
datacenter=None, image=None, iso=None, rescue_enabled=None, locked=None,
backup_window=None, outgoing_traffic=None, ingoing_traffic=None, in-
cluded_traffic=None, protection=None, labels=None, volumes=None, private_net=None,
primary_disk_size=None, placement_group=None)
```

Server Domain

Parameters

- **id** – int ID of the server

- **name** – str Name of the server (must be unique per project and a valid hostname as per RFC 1123)
- **status** – str Status of the server Choices: *running*, *initializing*, *starting*, *stopping*, *off*, *deleting*, *migrating*, *rebuilding*, *unknown*
- **created** – datetime Point in time when the server was created
- **public_net** – *PublicNetwork* Public network information.
- **server_type** – *BoundServerType*
- **datacenter** – *BoundDatacenter*
- **image** – *BoundImage*, None
- **iso** – *BoundIso*, None
- **rescue_enabled** – bool True if rescue mode is enabled: Server will then boot into rescue system on next reboot.
- **locked** – bool True if server has been locked and is not available to user.
- **backup_window** – str, None Time window (UTC) in which the backup will run, or None if the backups are not enabled
- **outgoing_traffic** – int, None Outbound Traffic for the current billing period in bytes
- **ingoining_traffic** – int, None Inbound Traffic for the current billing period in bytes
- **included_traffic** – int Free Traffic for the current billing period in bytes
- **primary_disk_size** – int Size of the primary Disk
- **protection** – dict Protection configuration for the server
- **labels** – dict User-defined labels (key-value pairs)
- **volumes** – List[*BoundVolume*] Volumes assigned to this server.
- **private_net** – List[*PrivateNet* <hcloud.servers.domain.PrivateNet> Private networks information.

STATUS_DELETING = 'deleting'
Server Status deleting

STATUS_INIT = 'initializing'
Server Status initializing

STATUS_MIGRATING = 'migrating'
Server Status migrating

STATUS_OFF = 'off'
Server Status off

STATUS_REBUILDING = 'rebuilding'
Server Status rebuilding

STATUS_RUNNING = 'running'
Server Status running

STATUS_STARTING = 'starting'
Server Status starting

STATUS_STOPPING = 'stopping'
Server Status stopping

```
STATUS_UNKNOWN = 'unknown'
```

Server Status unknown

```
class PublicNetwork(ipv4, ipv6, floating_ips, firewalls=None)
```

Public Network Domain

Parameters

- **ipv4** – *IPv4Address*
- **ipv6** – *IPv6Network*
- **floating_ips** – List[*BoundFloatingIP*]
- **firewalls** – List[*PublicNetworkFirewall*]

```
class IPv4Address(ip, blocked, dns_ptr)
```

IPv4 Address Domain

Parameters

- **ip** – str The IPv4 Address
- **blocked** – bool Determine if the IP is blocked
- **dns_ptr** – str DNS PTR for the ip

```
class IPv6Network(ip, blocked, dns_ptr)
```

IPv6 Network Domain

Parameters

- **ip** – str The IPv6 Network as CIDR Notation
- **blocked** – bool Determine if the Network is blocked
- **dns_ptr** – dict DNS PTR Records for the Network as Dict
- **network** – str The network without the network mask
- **network_mask** – str The network mask

```
class CreateServerResponse(server, action, next_actions, root_password)
```

Create Server Response Domain

Parameters

- **server** – *BoundServer* The created server
- **action** – *BoundAction* Shows the progress of the server creation
- **next_actions** – List[*BoundAction*] Additional actions like a *start_server* action after the server creation
- **root_password** – str, None The root password of the server if no SSH-Key was given on server creation

```
class ResetPasswordResponse(action, root_password)
```

Reset Password Response Domain

Parameters

- **action** – *BoundAction* Shows the progress of the server password reset action
- **root_password** – str The root password of the server

```
class EnableRescueResponse(action, root_password)
```

Enable Rescue Response Domain

Parameters

- **action** – *BoundAction* Shows the progress of the server enable rescue action
- **root_password** – str The root password of the server in the rescue mode

class RequestConsoleResponse (*action, wss_url, password*)

Request Console Response Domain

Parameters

- **action** – *BoundAction* Shows the progress of the server request console action
- **wss_url** – str URL of websocket proxy to use. This includes a token which is valid for a limited time only.
- **password** – str VNC password to use for this connection. This password only works in combination with a wss_url with valid token.

4.2.15 SSHKeysClient

class SSHKeysClient (*client*)

create (*name, public_key, labels=None*)

Creates a new SSH key with the given name and public_key.

Parameters

- **name** – str
- **public_key** – str Public Key of the SSH Key you want create
- **labels** – Dict[str, str] (optional) User-defined labels (key-value pairs)

Returns *BoundSSHKey*

get_all (*name=None, fingerprint=None, label_selector=None*)

Get all SSH keys from the account

Parameters

- **name** – str (optional) Can be used to filter SSH keys by their name. The response will only contain the SSH key matching the specified name.
- **fingerprint** – str (optional) Can be used to filter SSH keys by their fingerprint. The response will only contain the SSH key matching the specified fingerprint.
- **label_selector** – str (optional) Can be used to filter SSH keys by labels. The response will only contain SSH keys matching the label selector.

Returns List[*BoundSSHKey*]

get_by_fingerprint (*fingerprint*)

Get ssh key by fingerprint

Parameters **fingerprint** – str Used to get ssh key by fingerprint.

Returns *BoundSSHKey*

get_by_id (*id*)

Get a specific SSH Key by its ID

Parameters **id** – int

Returns *BoundSSHKey*

get_by_name (*name*)
Get ssh key by name

Parameters **name** – str Used to get ssh key by name.

Returns *BoundSSHKey*

get_list (*name=None, fingerprint=None, label_selector=None, page=None, per_page=None*)
Get a list of SSH keys from the account

Parameters

- **name** – str (optional) Can be used to filter SSH keys by their name. The response will only contain the SSH key matching the specified name.
- **fingerprint** – str (optional) Can be used to filter SSH keys by their fingerprint. The response will only contain the SSH key matching the specified fingerprint.
- **label_selector** – str (optional) Can be used to filter SSH keys by labels. The response will only contain SSH keys matching the label selector.
- **page** – int (optional) Specifies the page to fetch
- **per_page** – int (optional) Specifies how many results are returned by page

Returns (List[*BoundSSHKey*], Meta)

update (*ssh_key, name=None, labels=None*)
Updates an SSH key. You can update an SSH key name and an SSH key labels.

Parameters

- **ssh_key** – *BoundSSHKey* or *SSHKey*
- **name** – str (optional) New Description to set
- **labels** – Dict[str, str] (optional) User-defined labels (key-value pairs)

Returns *BoundSSHKey*

class BoundSSHKey (*client, data={}, complete=True*)

delete ()
Deletes an SSH key. It cannot be used anymore. :return: boolean

model
alias of *hcloud.ssh_keys.domain.SSHKey*

update (*name=None, labels=None*)
Updates an SSH key. You can update an SSH key name and an SSH key labels.

Parameters

- **description** – str (optional) New Description to set
- **labels** – Dict[str, str] (optional) User-defined labels (key-value pairs)

Returns :class:`BoundSSHKey <hcloud.ssh_keys.client.BoundSSHKey>

class SSHKey (*id=None, name=None, fingerprint=None, public_key=None, labels=None, created=None*)
SSHKey Domain

Parameters

- **id** – int ID of the SSH key
- **name** – str Name of the SSH key (must be unique per project)

- **fingerprint** – str Fingerprint of public key
- **public_key** – str Public Key
- **labels** – Dict User-defined labels (key-value pairs)
- **created** – datetime Point in time when the SSH Key was created

4.2.16 VolumesClient

class VolumesClient (*client*)

attach (*volume, server, automount=None*)

Attaches a volume to a server. Works only if the server is in the same location as the volume.

Parameters

- **volume** – *BoundVolume* or *Volume*
- **server** – *BoundServer* or *Server*
- **automount** – boolean

Returns *BoundAction*

change_protection (*volume, delete=None*)

Changes the protection configuration of a volume.

Parameters

- **volume** – *BoundVolume* or *Volume*
- **delete** – boolean If True, prevents the volume from being deleted

Returns *BoundAction*

create (*size, name, labels=None, location=None, server=None, automount=None, format=None*)

Creates a new volume attached to a server.

Parameters

- **size** – int Size of the volume in GB
- **name** – str Name of the volume
- **labels** – Dict[str,str] (optional) User-defined labels (key-value pairs)
- **location** – *BoundLocation* or *Location*
- **server** – *BoundServer* or *Server*
- **automount** – boolean (optional) Auto mount volumes after attach.
- **format** – str (optional) Format volume after creation. One of: xfs, ext4

Returns *CreateVolumeResponse*

delete (*volume*)

Deletes a volume. All volume data is irreversibly destroyed. The volume must not be attached to a server and it must not have delete protection enabled.

Parameters **volume** – *BoundVolume* or *Volume*

Returns boolean

detach (*volume*)

Detaches a volume from the server it's attached to. You may attach it to a server again at a later time.

Parameters **volume** – *BoundVolume* or *Volume*

Returns *BoundAction*

get_actions (*volume*, *status=None*, *sort=None*)

Returns all action objects for a volume.

Parameters

- **volume** – *BoundVolume* or *Volume*
- **status** – List[str] (optional) Response will have only actions with specified statuses. Choices: *running success error*

:param sort:List[str] (optional) Specify how the results are sorted. Choices: *id id:asc id:desc command command:asc command:desc status status:asc status:desc progress progress:asc progress:desc started started:asc started:desc finished finished:asc finished:desc*

Returns List[*BoundAction*]

get_actions_list (*volume*, *status=None*, *sort=None*, *page=None*, *per_page=None*)

Returns all action objects for a volume.

Parameters

- **volume** – *BoundVolume* or *Volume*
- **status** – List[str] (optional) Response will have only actions with specified statuses. Choices: *running success error*
- **sort** – List[str] (optional) Specify how the results are sorted. Choices: *id id:asc id:desc command command:asc command:desc status status:asc status:desc progress progress:asc progress:desc started started:asc started:desc finished finished:asc finished:desc*
- **page** – int (optional) Specifies the page to fetch
- **per_page** – int (optional) Specifies how many results are returned by page

Returns (List[*BoundAction*], *Meta*)

get_all (*label_selector=None*, *status=None*)

Get all volumes from this account

Parameters

- **label_selector** – Can be used to filter volumes by labels. The response will only contain volumes matching the label selector.
- **status** – List[str] (optional) Can be used to filter volumes by their status. The response will only contain volumes matching the status.

Returns List[*BoundVolume*]

get_by_id (*id*)

Get a specific volume by its id

Parameters **id** – int

Returns *BoundVolume*

get_by_name (*name*)

Get volume by name

Parameters **name** – str Used to get volume by name.

Returns *BoundVolume*

get_list (*name=None, label_selector=None, page=None, per_page=None, status=None*)

Get a list of volumes from this account

Parameters

- **name** – str (optional) Can be used to filter volumes by their name.
- **label_selector** – str (optional) Can be used to filter volumes by labels. The response will only contain volumes matching the label selector.
- **status** – List[str] (optional) Can be used to filter volumes by their status. The response will only contain volumes matching the status.
- **page** – int (optional) Specifies the page to fetch
- **per_page** – int (optional) Specifies how many results are returned by page

Returns (List[*BoundVolume*], *Meta*)

resize (*volume, size*)

Changes the size of a volume. Note that downsizing a volume is not possible.

Parameters

- **volume** – *BoundVolume* or *Volume*
- **size** – int New volume size in GB (must be greater than current size)

Returns *BoundAction*

update (*volume, name=None, labels=None*)

Updates the volume properties.

Parameters

- **volume** – *BoundVolume* or *Volume*
- **name** – str (optional) New volume name
- **labels** – Dict[str, str] (optional) User-defined labels (key-value pairs)

Returns *BoundAction*

class BoundVolume (*client, data, complete=True*)

attach (*server, automount=None*)

Attaches a volume to a server. Works only if the server is in the same location as the volume.

Parameters

- **server** – *BoundServer* or *Server*
- **automount** – boolean

Returns *BoundAction*

change_protection (*delete=None*)

Changes the protection configuration of a volume.

Parameters **delete** – boolean If True, prevents the volume from being deleted

Returns *BoundAction*

delete()

Deletes a volume. All volume data is irreversibly destroyed. The volume must not be attached to a server and it must not have delete protection enabled.

Returns boolean

detach()

Detaches a volume from the server it's attached to. You may attach it to a server again at a later time.

Returns *BoundAction*

get_actions(status=None, sort=None)

Returns all action objects for a volume.

Parameters status – List[str] (optional) Response will have only actions with specified statuses. Choices: *running success error*

:param sort:List[str] (optional) Specify how the results are sorted. Choices: *id id:asc id:desc command command:asc command:desc status status:asc status:desc progress progress:asc progress:desc started started:asc started:desc finished finished:asc finished:desc*

Returns List[*BoundAction*]

get_actions_list(status=None, sort=None, page=None, per_page=None)

Returns all action objects for a volume.

Parameters

- **status** – List[str] (optional) Response will have only actions with specified statuses. Choices: *running success error*
- **sort** – List[str] (optional) Specify how the results are sorted. Choices: *id id:asc id:desc command command:asc command:desc status status:asc status:desc progress progress:asc progress:desc started started:asc started:desc finished finished:asc finished:desc*
- **page** – int (optional) Specifies the page to fetch
- **per_page** – int (optional) Specifies how many results are returned by page

Returns (List[*BoundAction*], Meta)

model

alias of *hcloud.volumes.domain.Volume*

resize(size)

Changes the size of a volume. Note that downsizing a volume is not possible.

Parameters size – int New volume size in GB (must be greater than current size)

Returns *BoundAction*

update(name=None, labels=None)

Updates the volume properties.

Parameters

- **name** – str (optional) New volume name
- **labels** – Dict[str, str] (optional) User-defined labels (key-value pairs)

Returns *BoundAction*

```
class Volume(id, name=None, server=None, created=None, location=None, size=None,
             linux_device=None, format=None, protection=None, labels=None, status=None)
    Volume Domain
```

Parameters

- **id** – int ID of the Volume
- **name** – str Name of the Volume
- **server** – *BoundServer*, None Server the Volume is attached to, None if it is not attached at all.
- **created** – datetime Point in time when the Volume was created
- **location** – *BoundLocation* Location of the Volume. Volume can only be attached to Servers in the same location.
- **size** – int Size in GB of the Volume
- **linux_device** – str Device path on the file system for the Volume
- **protection** – dict Protection configuration for the Volume
- **labels** – dict User-defined labels (key-value pairs)
- **status** – str Current status of the volume Choices: *creating*, *available*
- **format** – str, None Filesystem of the volume if formatted on creation, None if not formatted on creation.

```
STATUS_AVAILABLE = 'available'
```

Volume Status available

```
STATUS_CREATING = 'creating'
```

Volume Status creating

```
class CreateVolumeResponse(volume, action, next_actions)
    Create Volume Response Domain
```

Parameters

- **volume** – *BoundVolume* The created volume
- **action** – *BoundAction* The action that shows the progress of the Volume Creation
- **next_actions** – List[*BoundAction*] List of actions that are performed after the creation, like attaching to a server

4.3 Exceptions

```
class APIException(code, message, details)
    There was an error while performing an API Request
```

```
class ActionFailedException(action)
    The Action you was waiting for failed
```

```
class ActionTimeoutException(action)
    The Action you was waiting for timeouted in hcloud-python.
```

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

5.1 Types of Contributions

5.1.1 Report Bugs

Report bugs at <https://github.com/hetznercloud/hcloud-python/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

5.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

5.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

5.1.4 Write Documentation

Hetzner Cloud Python could always use more documentation, whether as part of the official Hetzner Cloud Python docs, in docstrings, or even on the web in blog posts, articles, and such.

5.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/hetznercloud/hcloud-python/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

5.2 Get Started!

Ready to contribute? Here's how to set up *hcloud-python* for local development.

1. Fork the *hcloud-python* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/hcloud-python.git
```

3. Install your local copy into a virtualenv. Assuming you have *virtualenvwrapper* installed, this is how you set up your fork for local development:

```
$ mkvirtualenv hcloud-python
$ cd hcloud-python/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass *flake8* and the tests, including testing other Python versions with *tox*:

```
$ flake8 hetznercloud tests
$ python setup.py test or py.test
$ tox
```

To get *flake8* and *tox*, just *pip* install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

5.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.7, 3.5 and 3.6, and for PyPy. Check https://travis-ci.org/hetznercloud/hcloud-python/pull_requests and make sure that the tests pass for all supported Python versions.

5.4 Tips

To run a subset of tests:

```
$ py.test tests.test_hetznercloud
```

5.5 How to release

A reminder for the maintainers on how to release a new version. Make sure all your changes are committed (including an entry in CHANGELOG.rst). Then run:

1. Change the version under `/hcloud/version.py`
2. Push the change to the *master* branch and tag an new release through the [Github UI](#).

Travis will then deploy to PyPI if tests pass.

6.1 v1.16.0 (2021-08-17)

- Feature: Add support for Load Balancer DNS PTRs

6.2 v1.15.0 (2021-08-16)

- Feature: Add support for Placement Groups

6.3 v1.14.1 (2021-08-10)

- Bugfix: Fix crash on extra fields in public_net response
- Improvement: Format code with black

6.4 v1.14.0 (2021-08-03)

- Feature: Add support for Firewall rule descriptions

6.5 v1.13.0 (2021-07-16)

- Feature: Add support for Firewall Protocols ESP and GRE
- Feature: Add support for Image Type APP
- Feature: Add support for creating Firewalls with Firewalls
- Feature: Add support for Label Selectors in Firewalls

- Improvement: Improve handling of underlying TCP connections. Now for every client instance a single TCP connection is used instead of one per call.
- Note: Support for Python 2.7 and Python 3.5 was removed

6.6 v1.12.0 (2021-04-06)

- Feature: Add support for managed Certificates

6.7 v1.11.0 (2021-03-11)

- Feature: Add support for Firewalls
- Feature: Add *primary_disk_size* to *Server* Domain

6.8 v1.10.0 (2020-11-03)

- Feature: Add *include_deprecated* filter to *get_list* and *get_all* on *ImagesClient*
- Feature: Add vSwitch support to *add_subnet* on *NetworksClient*
- Feature: Add subnet type constants to *NetworkSubnet* domain (*NetworkSubnet.TYPE_CLOUD*, *NetworkSubnet.TYPE_VSWITCH*)

6.9 v1.9.1 (2020-08-11)

- Bugfix: *BoundLoadBalancer* serialization failed when using IP targets

6.10 v1.9.0 (2020-08-10)

- Feature: Add *included_traffic*, *outgoing_traffic* and *ingoing_traffic* properties to Load Balancer domain
- Feature: Add *change_type*-method to *LoadBalancersClient*
- Feature: Add support for *LoadBalancerTargetLabelSelector*
- Feature: Add support for *LoadBalancerTargetLabelSelector*

6.11 v1.8.2 (2020-07-20)

- Fix: Loosen up the requirements.

6.12 v1.8.1 (2020-06-29)

- Fix Load Balancer Client.
- Fix: Unify setting of request parameters within *get_list* methods.

6.13 1.8.0 (2020-06-22)

- Feature: Add Load Balancers **Attention: The Load Balancer support in v1.8.0 is kind of broken. Please use v1.8.1**
- Feature: Add Certificates

6.14 1.7.1 (2020-06-15)

- Feature: Add requests 2.23 support

6.15 1.7.0 (2020-06-05)

- Feature: Add support for the optional 'networks' parameter on server creation.
- Feature: Add python 3.9 support
- Feature: Add subnet type *cloud*

6.16 1.6.3 (2020-01-09)

- Feature: Add 'created' property to SSH Key domain
- Fix: Remove ISODateTime Descriptor because it leads to wrong dates

6.17 1.6.2 (2019-10-15)

- Fix: future dependency requirement was too strict

6.18 1.6.1 (2019-10-01)

- Fix: python-dateutil dependency requirement was too strict

6.19 1.6.0 (2019-09-17)

- Feature: Add missing *get_by_name* on *FloatingIPsClient*

6.20 1.5.0 (2019-09-16)

- Fix: ServersClient.create_image fails when specifying the *labels*
- Feature: Add support for *name* on Floating IPs

6.21 1.4.1 (2019-08-19)

- Fix: Documentation for *NetworkRoute* domain was missing
- Fix: *requests* dependency requirement was too strict

6.22 1.4.0 (2019-07-29)

- Feature: Add *mac_address* to Server PrivateNet domain
- Feature: Add python 3.8 support

6.23 1.3.0 (2019-07-10)

- Feature: Add status filter for servers, images and volumes
- Feature: Add 'created' property to Floating IP domain
- Feature: Add 'Networks' support

6.24 1.2.1 (2019-03-13)

- Fix: BoundVolume.server property now casted to the 'BoundServer'.

6.25 1.2.0 (2019-03-06)

- Feature: Add *get_by_fingerprint*-method for ssh keys
- Fix: Create Floating IP with location raises an error because no action was given.

6.26 1.1.0 (2019-02-27)

- Feature: Add *STATUS*-constants for server and volume status

6.27 1.0.1 (2019-02-22)

- Fix: Ignore unknown fields in API response instead of raising an error

6.28 1.0.0 (2019-02-21)

- First stable release.

You can find the documentation under <https://hcloud-python.readthedocs.io/en/latest/>

6.29 0.1.0 (2018-12-20)

- First release on GitHub.

This is the official Hetzner Cloud python library.

7.1 Examples

7.2 Create Server

```
1 from hcloud import Client
2 from hcloud.server_types.domain import ServerType
3 from hcloud.images.domain import Image
4
5 client = Client(token="{YOUR_API_TOKEN}") # Please paste your API token here between ↵
   ↵the quotes
6 response = client.servers.create(name="my-server", server_type=ServerType(name="cx11
   ↵"), image=Image(name="ubuntu-20.04"))
7 server = response.server
8 print(server)
9 print("Root Password: " + response.root_password)
```

7.3 List Servers

```
1 from hcloud import Client
2
3 client = Client(token="{YOUR_API_TOKEN}") # Please paste your API token here between ↵
   ↵the quotes
```

(continues on next page)

(continued from previous page)

```
4 servers = client.servers.get_all()  
5 print(servers)
```

You can find more examples in the [Example Folder](#) in the Github Repository.

7.4 License

The MIT License (MIT). Please see [License File](#) for more information.

A

Action (class in *hcloud.actions.domain*), 9
 ActionFailedException (class in *hcloud.actions.domain*), 66
 actions (Client attribute), 7
 ActionsClient (class in *hcloud.actions.client*), 9
 ActionTimeoutException (class in *hcloud.actions.domain*), 66
 add_route() (*BoundNetwork* method), 42
 add_route() (*NetworksClient* method), 40
 add_service() (*BoundLoadBalancer* method), 34
 add_service() (*LoadBalancersClient* method), 30
 add_subnet() (*BoundNetwork* method), 43
 add_subnet() (*NetworksClient* method), 40
 add_target() (*BoundLoadBalancer* method), 34
 add_target() (*LoadBalancersClient* method), 30
 add_to_placement_group() (*BoundServer* method), 54
 add_to_placement_group() (*ServersClient* method), 48
 APIException (class in *hcloud*), 66
 apply_to_resources() (*BoundFirewall* method), 17
 apply_to_resources() (*FirewallsClient* method), 15
 assign() (*BoundFloatingIP* method), 22
 assign() (*FloatingIPsClient* method), 19
 attach() (*BoundVolume* method), 64
 attach() (*VolumesClient* method), 62
 attach_iso() (*BoundServer* method), 54
 attach_iso() (*ServersClient* method), 48
 attach_to_network() (*BoundLoadBalancer* method), 34
 attach_to_network() (*BoundServer* method), 54
 attach_to_network() (*LoadBalancersClient* method), 30
 attach_to_network() (*ServersClient* method), 48

B

BoundAction (class in *hcloud.actions.client*), 9
 BoundCertificate (class in *hcloud.certificates.client*), 12
 BoundDatacenter (class in *hcloud.datacenters.client*), 14
 BoundFirewall (class in *hcloud.firewalls.client*), 17
 BoundFloatingIP (class in *hcloud.floating_ips.client*), 22
 BoundImage (class in *hcloud.images.client*), 26
 BoundIso (class in *hcloud.isos.client*), 29
 BoundLoadBalancer (class in *hcloud.load_balancers.client*), 34
 BoundLocation (class in *hcloud.locations.client*), 39
 BoundNetwork (class in *hcloud.networks.client*), 42
 BoundPlacementGroup (class in *hcloud.placement_groups.client*), 46
 BoundServer (class in *hcloud.servers.client*), 54
 BoundServerType (class in *hcloud.server_types.client*), 47
 BoundSSHKey (class in *hcloud.ssh_keys.client*), 61
 BoundVolume (class in *hcloud.volumes.client*), 64

C

Certificate (class in *hcloud.certificates.domain*), 13
 certificates (Client attribute), 7
 CertificatesClient (class in *hcloud.certificates.client*), 10
 change_algorithm() (*BoundLoadBalancer* method), 34
 change_algorithm() (*LoadBalancersClient* method), 30
 change_alias_ips() (*BoundServer* method), 54
 change_alias_ips() (*ServersClient* method), 49
 change_dns_ptr() (*BoundFloatingIP* method), 22
 change_dns_ptr() (*BoundLoadBalancer* method), 35
 change_dns_ptr() (*BoundServer* method), 54
 change_dns_ptr() (*FloatingIPsClient* method), 20

change_dns_ptr() (*LoadBalancersClient* method), 31

change_dns_ptr() (*ServersClient* method), 49

change_ip_range() (*BoundNetwork* method), 43

change_ip_range() (*NetworksClient* method), 40

change_protection() (*BoundFloatingIP* method), 22

change_protection() (*BoundImage* method), 26

change_protection() (*BoundLoadBalancer* method), 35

change_protection() (*BoundNetwork* method), 43

change_protection() (*BoundServer* method), 55

change_protection() (*BoundVolume* method), 64

change_protection() (*FloatingIPsClient* method), 20

change_protection() (*ImagesClient* method), 24

change_protection() (*LoadBalancersClient* method), 31

change_protection() (*NetworksClient* method), 40

change_protection() (*ServersClient* method), 49

change_protection() (*VolumesClient* method), 62

change_type() (*BoundLoadBalancer* method), 35

change_type() (*BoundServer* method), 55

change_type() (*LoadBalancersClient* method), 31

change_type() (*ServersClient* method), 49

Client (class in *hcloud*), 7

create() (*CertificatesClient* method), 10

create() (*FirewallsClient* method), 15

create() (*FloatingIPsClient* method), 20

create() (*LoadBalancersClient* method), 31

create() (*NetworksClient* method), 40

create() (*PlacementGroupsClient* method), 45

create() (*ServersClient* method), 49

create() (*SSHKeysClient* method), 60

create() (*VolumesClient* method), 62

create_image() (*BoundServer* method), 55

create_image() (*ServersClient* method), 50

create_managed() (*CertificatesClient* method), 10

CreateFirewallResponse (class in *hcloud.firewalls.domain*), 19

CreateFloatingIPResponse (class in *hcloud.floating_ips.domain*), 24

CreateImageResponse (class in *hcloud.images.domain*), 28

CreateNetworkResponse (class in *hcloud.networks.domain*), 45

CreatePlacementGroupResponse (class in *hcloud.placement_groups.domain*), 47

CreateServerResponse (class in *hcloud.servers.domain*), 59

CreateVolumeResponse (class in *hcloud.volumes.domain*), 66

D

Datacenter (class in *hcloud.datacenters.domain*), 14

datacenters (Client attribute), 7

DatacentersClient (class in *hcloud.datacenters.client*), 14

DatacenterServerTypes (class in *hcloud.datacenters.domain*), 14

delete() (*BoundCertificate* method), 12

delete() (*BoundFirewall* method), 17

delete() (*BoundFloatingIP* method), 22

delete() (*BoundImage* method), 26

delete() (*BoundLoadBalancer* method), 35

delete() (*BoundNetwork* method), 43

delete() (*BoundPlacementGroup* method), 46

delete() (*BoundServer* method), 55

delete() (*BoundSSHKey* method), 61

delete() (*BoundVolume* method), 65

delete() (*FirewallsClient* method), 15

delete() (*FloatingIPsClient* method), 20

delete() (*ImagesClient* method), 24

delete() (*LoadBalancersClient* method), 32

delete() (*NetworksClient* method), 41

delete() (*PlacementGroupsClient* method), 45

delete() (*ServersClient* method), 50

delete() (*VolumesClient* method), 62

delete_route() (*BoundNetwork* method), 43

delete_route() (*NetworksClient* method), 41

delete_service() (*BoundLoadBalancer* method), 35

delete_service() (*LoadBalancersClient* method), 32

delete_subnet() (*BoundNetwork* method), 43

delete_subnet() (*NetworksClient* method), 41

detach() (*BoundVolume* method), 65

detach() (*VolumesClient* method), 62

detach_from_network() (*BoundLoadBalancer* method), 35

detach_from_network() (*BoundServer* method), 55

detach_from_network() (*LoadBalancersClient* method), 32

detach_from_network() (*ServersClient* method), 50

detach_iso() (*BoundServer* method), 55

detach_iso() (*ServersClient* method), 51

DIRECTION_IN (*FirewallRule* attribute), 19

DIRECTION_OUT (*FirewallRule* attribute), 19

disable_backup() (*BoundServer* method), 55

disable_backup() (*ServersClient* method), 51

disable_public_interface() (*BoundLoadBalancer* method), 35

disable_public_interface() (*LoadBalancerClient* method), 32

disable_rescue() (*BoundServer* method), 55

`disable_rescue()` (*ServersClient method*), 51

E

`enable_backup()` (*BoundServer method*), 56

`enable_backup()` (*ServersClient method*), 51

`enable_public_interface()` (*BoundLoadBalancer method*), 35

`enable_public_interface()` (*LoadBalancersClient method*), 32

`enable_rescue()` (*BoundServer method*), 56

`enable_rescue()` (*ServersClient method*), 51

`EnableRescueResponse` (class in *hcloud.servers.domain*), 59

F

`Firewall` (class in *hcloud.firewalls.domain*), 18

`FirewallResource` (class in *hcloud.firewalls.domain*), 19

`FirewallRule` (class in *hcloud.firewalls.domain*), 18

`firewalls` (*Client attribute*), 7

`FirewallsClient` (class in *hcloud.firewalls.client*), 15

`floating_ips` (*Client attribute*), 7

`FloatingIP` (class in *hcloud.floating_ips.domain*), 23

`FloatingIPsClient` (class in *hcloud.floating_ips.client*), 19

G

`get_actions()` (*BoundCertificate method*), 12

`get_actions()` (*BoundFirewall method*), 17

`get_actions()` (*BoundFloatingIP method*), 22

`get_actions()` (*BoundImage method*), 26

`get_actions()` (*BoundLoadBalancer method*), 35

`get_actions()` (*BoundNetwork method*), 43

`get_actions()` (*BoundServer method*), 56

`get_actions()` (*BoundVolume method*), 65

`get_actions()` (*CertificatesClient method*), 11

`get_actions()` (*FirewallsClient method*), 15

`get_actions()` (*FloatingIPsClient method*), 20

`get_actions()` (*ImagesClient method*), 24

`get_actions()` (*LoadBalancersClient method*), 32

`get_actions()` (*NetworksClient method*), 41

`get_actions()` (*ServersClient method*), 51

`get_actions()` (*VolumesClient method*), 63

`get_actions_list()` (*BoundCertificate method*), 12

`get_actions_list()` (*BoundFirewall method*), 17

`get_actions_list()` (*BoundFloatingIP method*), 23

`get_actions_list()` (*BoundImage method*), 27

`get_actions_list()` (*BoundLoadBalancer method*), 36

`get_actions_list()` (*BoundNetwork method*), 43

`get_actions_list()` (*BoundServer method*), 56

`get_actions_list()` (*BoundVolume method*), 65

`get_actions_list()` (*CertificatesClient method*), 11

`get_actions_list()` (*FirewallsClient method*), 15

`get_actions_list()` (*FloatingIPsClient method*), 21

`get_actions_list()` (*ImagesClient method*), 24

`get_actions_list()` (*LoadBalancersClient method*), 33

`get_actions_list()` (*NetworksClient method*), 41

`get_actions_list()` (*ServersClient method*), 51

`get_actions_list()` (*VolumesClient method*), 63

`get_all()` (*ActionsClient method*), 9

`get_all()` (*CertificatesClient method*), 11

`get_all()` (*DatacentersClient method*), 14

`get_all()` (*FirewallsClient method*), 16

`get_all()` (*FloatingIPsClient method*), 21

`get_all()` (*ImagesClient method*), 25

`get_all()` (*IsosClient method*), 28

`get_all()` (*LoadBalancersClient method*), 33

`get_all()` (*LoadBalancerTypesClient method*), 29

`get_all()` (*LocationsClient method*), 39

`get_all()` (*NetworksClient method*), 42

`get_all()` (*PlacementGroupsClient method*), 45

`get_all()` (*ServersClient method*), 52

`get_all()` (*ServerTypesClient method*), 47

`get_all()` (*SSHKeysClient method*), 60

`get_all()` (*VolumesClient method*), 63

`get_by_fingerprint()` (*SSHKeysClient method*), 60

`get_by_id()` (*ActionsClient method*), 9

`get_by_id()` (*CertificatesClient method*), 11

`get_by_id()` (*DatacentersClient method*), 14

`get_by_id()` (*FirewallsClient method*), 16

`get_by_id()` (*FloatingIPsClient method*), 21

`get_by_id()` (*ImagesClient method*), 25

`get_by_id()` (*IsosClient method*), 28

`get_by_id()` (*LoadBalancersClient method*), 33

`get_by_id()` (*LoadBalancerTypesClient method*), 29

`get_by_id()` (*LocationsClient method*), 39

`get_by_id()` (*NetworksClient method*), 42

`get_by_id()` (*PlacementGroupsClient method*), 45

`get_by_id()` (*ServersClient method*), 52

`get_by_id()` (*ServerTypesClient method*), 47

`get_by_id()` (*SSHKeysClient method*), 60

`get_by_id()` (*VolumesClient method*), 63

`get_by_name()` (*CertificatesClient method*), 11

`get_by_name()` (*DatacentersClient method*), 14

`get_by_name()` (*FirewallsClient method*), 16

`get_by_name()` (*FloatingIPsClient method*), 21

`get_by_name()` (*ImagesClient method*), 25

`get_by_name()` (*IsosClient method*), 28

`get_by_name()` (*LoadBalancersClient method*), 33

get_by_name() (*LoadBalancerTypesClient* method), 29
 get_by_name() (*LocationsClient* method), 39
 get_by_name() (*NetworksClient* method), 42
 get_by_name() (*PlacementGroupsClient* method), 46
 get_by_name() (*ServersClient* method), 52
 get_by_name() (*ServerTypesClient* method), 47
 get_by_name() (*SSHKeysClient* method), 61
 get_by_name() (*VolumesClient* method), 63
 get_list() (*ActionsClient* method), 9
 get_list() (*CertificatesClient* method), 12
 get_list() (*DatacentersClient* method), 14
 get_list() (*FirewallsClient* method), 16
 get_list() (*FloatingIPsClient* method), 21
 get_list() (*ImagesClient* method), 25
 get_list() (*IsosClient* method), 28
 get_list() (*LoadBalancersClient* method), 33
 get_list() (*LoadBalancerTypesClient* method), 29
 get_list() (*LocationsClient* method), 39
 get_list() (*NetworksClient* method), 42
 get_list() (*PlacementGroupsClient* method), 46
 get_list() (*ServersClient* method), 52
 get_list() (*ServerTypesClient* method), 47
 get_list() (*SSHKeysClient* method), 61
 get_list() (*VolumesClient* method), 64

I

Image (*class in hcloud.images.domain*), 27
 images (*Client attribute*), 7
 ImagesClient (*class in hcloud.images.client*), 24
 IPv4Address (*class in hcloud.servers.domain*), 59
 IPv6Network (*class in hcloud.servers.domain*), 59
 Iso (*class in hcloud.isos.domain*), 29
 isos (*Client attribute*), 7
 IsosClient (*class in hcloud.isos.client*), 28

L

load_balancer_types (*Client attribute*), 8
 load_balancers (*Client attribute*), 8
 LoadBalancer (*class in hcloud.load_balancers.domain*), 36
 LoadBalancerAlgorithm (*class in hcloud.load_balancers.domain*), 39
 LoadBalancerHealthCheckHttp (*class in hcloud.load_balancers.domain*), 38
 LoadBalancerHealthCheck (*class in hcloud.load_balancers.domain*), 38
 LoadBalancersClient (*class in hcloud.load_balancers.client*), 30
 LoadBalancerService (*class in hcloud.load_balancers.domain*), 37
 LoadBalancerServiceHttp (*class in hcloud.load_balancers.domain*), 37

LoadBalancerTarget (*class in hcloud.load_balancers.domain*), 38
 LoadBalancerTargetIP (*class in hcloud.load_balancers.domain*), 38
 LoadBalancerTargetLabelSelector (*class in hcloud.load_balancers.domain*), 38
 LoadBalancerType (*class in hcloud.load_balancer_types.domain*), 29
 LoadBalancerTypesClient (*class in hcloud.load_balancer_types.client*), 29
 Location (*class in hcloud.locations.domain*), 39
 locations (*Client attribute*), 8
 LocationsClient (*class in hcloud.locations.client*), 39

M

model (*BoundAction attribute*), 9
 model (*BoundCertificate attribute*), 13
 model (*BoundDatacenter attribute*), 14
 model (*BoundFirewall attribute*), 18
 model (*BoundFloatingIP attribute*), 23
 model (*BoundImage attribute*), 27
 model (*BoundIso attribute*), 29
 model (*BoundLoadBalancer attribute*), 36
 model (*BoundLocation attribute*), 39
 model (*BoundNetwork attribute*), 44
 model (*BoundPlacementGroup attribute*), 46
 model (*BoundServer attribute*), 56
 model (*BoundServerType attribute*), 47
 model (*BoundSSHKey attribute*), 61
 model (*BoundVolume attribute*), 65

N

Network (*class in hcloud.networks.domain*), 44
 NetworkRoute (*class in hcloud.networks.domain*), 45
 networks (*Client attribute*), 8
 NetworksClient (*class in hcloud.networks.client*), 40
 NetworkSubnet (*class in hcloud.networks.domain*), 44

P

in placement_groups (*Client attribute*), 8
 PlacementGroup (*class in hcloud.placement_groups.domain*), 46
 PlacementGroupsClient (*class in hcloud.placement_groups.client*), 45
 power_off() (*BoundServer method*), 56
 in power_off() (*ServersClient method*), 53
 power_on() (*BoundServer method*), 57
 in power_on() (*ServersClient method*), 53
 PROTOCOL_ESP (*FirewallRule attribute*), 19
 in PROTOCOL_GRE (*FirewallRule attribute*), 19
 PROTOCOL_ICMP (*FirewallRule attribute*), 19
 PROTOCOL_TCP (*FirewallRule attribute*), 19

PROTOCOL_UDP (*FirewallRule attribute*), 19
 PublicNetwork (*class in hcloud.servers.domain*), 59

R

reboot() (*BoundServer method*), 57
 reboot() (*ServersClient method*), 53
 rebuild() (*BoundServer method*), 57
 rebuild() (*ServersClient method*), 53
 remove_from_placement_group() (*Bound-Server method*), 57
 remove_from_placement_group() (*Server-Client method*), 53
 remove_from_resources() (*BoundFirewall method*), 18
 remove_from_resources() (*FirewallsClient method*), 16
 remove_target() (*BoundLoadBalancer method*), 36
 remove_target() (*LoadBalancersClient method*), 34
 request() (*Client method*), 8
 request_console() (*BoundServer method*), 57
 request_console() (*ServersClient method*), 53
 RequestConsoleResponse (*class in hcloud.servers.domain*), 60
 reset() (*BoundServer method*), 57
 reset() (*ServersClient method*), 53
 reset_password() (*BoundServer method*), 57
 reset_password() (*ServersClient method*), 53
 ResetPasswordResponse (*class in hcloud.servers.domain*), 59
 resize() (*BoundVolume method*), 65
 resize() (*VolumesClient method*), 64
 retry_issuance() (*BoundCertificate method*), 13
 retry_issuance() (*CertificatesClient method*), 12

S

Server (*class in hcloud.servers.domain*), 57
 server_types (*Client attribute*), 8
 servers (*Client attribute*), 8
 ServersClient (*class in hcloud.servers.client*), 48
 ServerType (*class in hcloud.server_types.domain*), 48
 ServerTypesClient (*class in hcloud.server_types.client*), 47
 set_rules() (*BoundFirewall method*), 18
 set_rules() (*FirewallsClient method*), 17
 shutdown() (*BoundServer method*), 57
 shutdown() (*ServersClient method*), 53
 ssh_keys (*Client attribute*), 8
 SSHKey (*class in hcloud.ssh_keys.domain*), 61
 SSHKeysClient (*class in hcloud.ssh_keys.client*), 60
 STATUS_AVAILABLE (*Volume attribute*), 66
 STATUS_CREATING (*Volume attribute*), 66
 STATUS_DELETING (*Server attribute*), 58
 STATUS_ERROR (*Action attribute*), 10

STATUS_INIT (*Server attribute*), 58
 STATUS_MIGRATING (*Server attribute*), 58
 STATUS_OFF (*Server attribute*), 58
 STATUS_REBUILDING (*Server attribute*), 58
 STATUS_RUNNING (*Action attribute*), 10
 STATUS_RUNNING (*Server attribute*), 58
 STATUS_STARTING (*Server attribute*), 58
 STATUS_STOPPING (*Server attribute*), 58
 STATUS_SUCCESS (*Action attribute*), 10
 STATUS_UNKNOWN (*Server attribute*), 58

T

TYPE_CLOUD (*NetworkSubnet attribute*), 44
 TYPE_LABEL_SELECTOR (*FirewallResource attribute*), 19
 TYPE_SERVER (*FirewallResource attribute*), 19
 TYPE_SERVER (*NetworkSubnet attribute*), 45
 TYPE_VSWITCH (*NetworkSubnet attribute*), 45

U

unassign() (*BoundFloatingIP method*), 23
 unassign() (*FloatingIPsClient method*), 22
 update() (*BoundCertificate method*), 13
 update() (*BoundFirewall method*), 18
 update() (*BoundFloatingIP method*), 23
 update() (*BoundImage method*), 27
 update() (*BoundLoadBalancer method*), 36
 update() (*BoundNetwork method*), 44
 update() (*BoundPlacementGroup method*), 46
 update() (*BoundServer method*), 57
 update() (*BoundSSHKey method*), 61
 update() (*BoundVolume method*), 65
 update() (*CertificatesClient method*), 12
 update() (*FirewallsClient method*), 17
 update() (*FloatingIPsClient method*), 22
 update() (*ImagesClient method*), 26
 update() (*LoadBalancersClient method*), 34
 update() (*NetworksClient method*), 42
 update() (*PlacementGroupsClient method*), 46
 update() (*ServersClient method*), 54
 update() (*SSHKeysClient method*), 61
 update() (*VolumesClient method*), 64
 update_service() (*BoundLoadBalancer method*), 36
 update_service() (*LoadBalancersClient method*), 34

V

Volume (*class in hcloud.volumes.domain*), 65
 volumes (*Client attribute*), 8
 VolumesClient (*class in hcloud.volumes.client*), 62

W

wait_until_finished() (*BoundAction method*), 9