

---

# Hetzner Cloud Python

*Release 1.35.0*

**Hetzner Cloud GmbH**

**May 07, 2024**



# CONTENTS

<b>1</b>	<b>Stable release</b>	<b>1</b>
<b>2</b>	<b>Via conda (Third-Party)</b>	<b>3</b>
<b>3</b>	<b>From sources</b>	<b>5</b>
<b>4</b>	<b>API References</b>	<b>7</b>
4.1	Main Interface . . . . .	7
4.2	API Clients . . . . .	9
4.2.1	ActionsClient . . . . .	9
4.2.2	CertificateClient . . . . .	11
4.2.3	DatacentersClient . . . . .	15
4.2.4	FirewallsClient . . . . .	17
4.2.5	Floating IPsClient . . . . .	22
4.2.6	ImagesClient . . . . .	28
4.2.7	ISOsClient . . . . .	33
4.2.8	LoadBalancerTypesClient . . . . .	35
4.2.9	LoadBalancerClient . . . . .	36
4.2.10	LocationsClient . . . . .	47
4.2.11	NetworksClient . . . . .	48
4.2.12	PlacementGroupsClient . . . . .	55
4.2.13	PrimaryIPsClient . . . . .	57
4.2.14	ServerTypesClient . . . . .	62
4.2.15	ServersClient . . . . .	63
4.2.16	SSHKeysClient . . . . .	78
4.2.17	VolumesClient . . . . .	81
4.3	Exceptions . . . . .	86
4.4	Other . . . . .	86
4.4.1	Helpers . . . . .	86
4.4.2	Deprecation Info . . . . .	87
<b>5</b>	<b>Contributing</b>	<b>89</b>
5.1	Types of Contributions . . . . .	89
5.1.1	Report Bugs . . . . .	89
5.1.2	Fix Bugs . . . . .	89
5.1.3	Implement Features . . . . .	89
5.1.4	Write Documentation . . . . .	89
5.1.5	Submit Feedback . . . . .	90
5.2	Get Started! . . . . .	90
5.3	Pull Request Guidelines . . . . .	90

<b>6</b>	<b>Changelog</b>	<b>91</b>
6.1	1.35.0 (2024-04-02)	91
	6.1.1 Features	91
	6.1.2 Bug Fixes	91
6.2	1.34.0 (2024-03-27)	91
	6.2.1 Features	91
6.3	1.33.3 (2024-03-27)	91
	6.3.1 Bug Fixes	91
	6.3.2 Dependencies	91
6.4	1.33.2 (2024-01-02)	92
	6.4.1 Bug Fixes	92
6.5	1.33.1 (2024-01-02)	92
	6.5.1 Bug Fixes	92
	6.5.2 Dependencies	92
6.6	1.33.0 (2023-12-19)	92
	6.6.1 Features	92
	6.6.2 Bug Fixes	93
	6.6.3 Dependencies	93
6.7	1.32.0 (2023-11-17)	93
	6.7.1 Features	93
	6.7.2 Dependencies	93
6.8	1.31.0 (2023-10-23)	93
	6.8.1 Features	93
	6.8.2 Dependencies	93
6.9	1.30.0 (2023-10-13)	94
	6.9.1 Features	94
	6.9.2 Dependencies	94
6.10	1.29.1 (2023-09-26)	94
	6.10.1 Bug Fixes	94
6.11	1.29.0 (2023-09-25)	94
	6.11.1 Features	94
	6.11.2 Bug Fixes	94
	6.11.3 Dependencies	94
	6.11.4 Documentation	95
6.12	1.28.0 (2023-08-17)	95
	6.12.1 Features	95
	6.12.2 Dependencies	95
	6.12.3 Documentation	95
6.13	1.27.2 (2023-08-09)	95
	6.13.1 Documentation	95
6.14	1.27.1 (2023-08-08)	95
	6.14.1 Bug Fixes	95
6.15	1.27.0 (2023-08-08)	96
	6.15.1 Features	96
	6.15.2 Bug Fixes	96
	6.15.3 Dependencies	96
	6.15.4 Documentation	96
6.16	1.26.0 (2023-07-19)	96
	6.16.1 Features	96
6.17	1.25.0 (2023-07-14)	96
	6.17.1 Features	96
	6.17.2 Dependencies	97
6.18	1.24.0 (2023-07-03)	97
	6.18.1 Features	97

6.19	6.18.2 Dependencies . . . . .	97
6.20	1.23.1 (2023-06-30) . . . . .	97
6.20	6.19.1 Bug Fixes . . . . .	97
6.21	1.23.0 (2023-06-26) . . . . .	97
6.21	6.20.1 Features . . . . .	97
6.21	6.20.2 Bug Fixes . . . . .	97
6.21	6.20.3 Dependencies . . . . .	97
6.22	1.22.0 (2023-06-22) . . . . .	98
6.22	6.21.1 Features . . . . .	98
6.22	6.21.2 Dependencies . . . . .	98
6.23	1.21.0 (2023-06-19) . . . . .	98
6.23	6.22.1 Features . . . . .	98
6.23	6.22.2 Bug Fixes . . . . .	98
6.23	6.22.3 Documentation . . . . .	98
6.24	1.20.0 (2023-05-12) . . . . .	98
6.24	6.23.1 Features . . . . .	98
6.25	v1.19.0 (2023-04-12) . . . . .	99
6.26	v1.18.2 (2022-12-27) . . . . .	99
6.27	v1.18.0 (2022-08-17) . . . . .	99
6.28	v1.17.0 (2022-06-29) . . . . .	99
6.29	v1.16.0 (2021-08-17) . . . . .	99
6.30	v1.15.0 (2021-08-16) . . . . .	99
6.31	v1.14.1 (2021-08-10) . . . . .	100
6.32	v1.14.0 (2021-08-03) . . . . .	100
6.33	v1.13.0 (2021-07-16) . . . . .	100
6.34	v1.12.0 (2021-04-06) . . . . .	100
6.35	v1.11.0 (2021-03-11) . . . . .	100
6.36	v1.10.0 (2020-11-03) . . . . .	100
6.37	v1.9.1 (2020-08-11) . . . . .	101
6.38	v1.9.0 (2020-08-10) . . . . .	101
6.39	v1.8.2 (2020-07-20) . . . . .	101
6.40	v1.8.1 (2020-06-29) . . . . .	101
6.41	1.8.0 (2020-06-22) . . . . .	101
6.42	1.7.1 (2020-06-15) . . . . .	101
6.43	1.7.0 (2020-06-05) . . . . .	101
6.44	1.6.3 (2020-01-09) . . . . .	102
6.45	1.6.2 (2019-10-15) . . . . .	102
6.46	1.6.1 (2019-10-01) . . . . .	102
6.47	1.6.0 (2019-09-17) . . . . .	102
6.48	1.5.0 (2019-09-16) . . . . .	102
6.49	1.4.1 (2019-08-19) . . . . .	102
6.50	1.4.0 (2019-07-29) . . . . .	102
6.51	1.3.0 (2019-07-10) . . . . .	103
6.52	1.2.1 (2019-03-13) . . . . .	103
6.53	1.2.0 (2019-03-06) . . . . .	103
6.54	1.1.0 (2019-02-27) . . . . .	103
6.55	1.0.1 (2019-02-22) . . . . .	103
6.56	1.0.0 (2019-02-21) . . . . .	103
6.56	0.1.0 (2018-12-20) . . . . .	103
<b>7</b>	<b>Hetzner Cloud Python</b>	<b>105</b>
7.1	Usage . . . . .	105
7.2	Supported Python versions . . . . .	106
7.3	Development . . . . .	106

	7.3.1	Deprecations implementation . . . . .	106
7.4		License . . . . .	107
<b>Index</b>			<b>109</b>

## 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.





## VIA CONDA (THIRD-PARTY)

Hetzner Cloud Python is also available as a conda-package via *conda-forge*. This package is not maintained by Hetzner Cloud and might be outdated.\_:

```
$ conda install -c conda-forge hcloud
```



## 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/main
```

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

```
$ pip install .
```



## API REFERENCES

### 4.1 Main Interface

```
class Client(token: str, api_endpoint: str = 'https://api.hetzner.cloud/v1', application_name: str | None = None,
             application_version: str | None = None, poll_interval: int = 1, timeout: float | tuple[float, float] |
             None = None)
```

Base Client for accessing the Hetzner Cloud API

#### **actions**

ActionsClient Instance

##### **Type**

*ActionsClient*

#### **certificates**

CertificatesClient Instance

##### **Type**

*CertificatesClient*

#### **datacenters**

DatacentersClient Instance

##### **Type**

*DatacentersClient*

#### **firewalls**

FirewallsClient Instance

##### **Type**

*FirewallsClient*

#### **floating\_ips**

FloatingIPsClient Instance

##### **Type**

*FloatingIPsClient*

#### **images**

ImagesClient Instance

##### **Type**

*ImagesClient*

**isos**

ImagesClient Instance

**Type**

*IsosClient*

**load\_balancer\_types**

LoadBalancerTypesClient Instance

**Type**

*LoadBalancerTypesClient*

**load\_balancers**

LoadBalancersClient Instance

**Type**

*LoadBalancersClient*

**locations**

LocationsClient Instance

**Type**

*LocationsClient*

**networks**

NetworksClient Instance

**Type**

*NetworksClient*

**placement\_groups**

PlacementGroupsClient Instance

**Type**

*PlacementGroupsClient*

**primary\_ips**

PrimaryIPsClient Instance

**Type**

*PrimaryIPsClient*

**request**(*method: str, url: str, tries: int = 1, \*\*kwargs*) → dict

Perform a request to the Hetzner Cloud API, wrapper around requests.request

**Parameters**

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

**Returns**

Response

**server\_types**

ServerTypesClient Instance

**Type**

*ServerTypesClient*

**servers**

ServersClient Instance

**Type***ServersClient***ssh\_keys**

SSHKeysClient Instance

**Type***SSHKeysClient***volumes**

VolumesClient Instance

**Type***VolumesClient*

## 4.2 API Clients

### 4.2.1 ActionsClient

**class ResourceActionsClient**(*client*: *Client*, *resource*: *str* | *None*)**get\_all**(*status*: *list[str]* | *None* = *None*, *sort*: *list[str]* | *None* = *None*) → *list[BoundAction]*

Get all actions.

**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*: *int*) → *BoundAction*

Get a specific action by its ID.

**Parameters****id** – *int***Returns***BoundAction***get\_list**(*status*: *list[str]* | *None* = *None*, *sort*: *list[str]* | *None* = *None*, *page*: *int* | *None* = *None*, *per\_page*: *int* | *None* = *None*) → *ActionsPageResult*

Get a list of actions.

**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 ActionsClient**(client: [Client](#))

**get\_all**(status: list[str] | None = None, sort: list[str] | None = None) → list[[BoundAction](#)]

Deprecated since version 1.28: Use `client.<resource>.actions.get_all()` instead, e.g. using `hcloud.certificates.client.CertificatesClient.actions`.

Starting 1 October 2023, it will no longer be available.

**get\_by\_id**(id: int) → [BoundAction](#)

Get a specific action by its ID.

**Parameters**

**id** – int

**Returns**

[BoundAction](#)

**get\_list**(status: list[str] | None = None, sort: list[str] | None = None, page: int | None = None, per\_page: int | None = None) → ActionsPageResult

Deprecated since version 1.28: Use `client.<resource>.actions.get_list()` instead, e.g. using `hcloud.certificates.client.CertificatesClient.actions`.

Starting 1 October 2023, it will no longer be available.

**class BoundAction**(client: ClientEntityBase, data: dict, complete: bool = True)

**model**

alias of [Action](#)

**wait\_until\_finished**(max\_retries: int = 100) → None

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: int, command: str | None = None, status: str | None = None, progress: int | None = None, started: str | None = None, finished: str | None = None, resources: list[dict] | None = None, error: dict | None = 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*: `Client`)

**actions:** `ResourceActionsClient`

Certificates scoped actions client

**Type**

`ResourceActionsClient`

**create**(*name*: *str*, *certificate*: *str*, *private\_key*: *str*, *labels*: *dict[str, str] | None = None*) → *BoundCertificate*

**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*: *str*, *domain\_names*: *list[str]*, *labels*: *dict[str, str] | None = None*) →

*CreateManagedCertificateResponse*

**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](#)

**delete**(certificate: [Certificate](#) | [BoundCertificate](#)) → bool

Deletes a certificate.

**Parameters**

**certificate** – [BoundCertificate](#) or [Certificate](#)

**Returns**

True

**get\_actions**(certificate: [Certificate](#) | [BoundCertificate](#), status: list[str] | None = None, sort: list[str] | None = None) → list[[BoundAction](#)]

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: [Certificate](#) | [BoundCertificate](#), status: list[str] | None = None, sort: list[str] | None = None, page: int | None = None, per\_page: int | None = None) → [ActionsPageResult](#)

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: str | None = None, label\_selector: str | None = None*) → list[*BoundCertificate*]

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: int*) → *BoundCertificate*

Get a specific certificate by its ID.

**Parameters**

**id** – int

**Returns**

*BoundCertificate*

**get\_by\_name**(*name: str*) → *BoundCertificate* | None

Get certificate by name

**Parameters**

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

**Returns**

*BoundCertificate*

**get\_list**(*name: str | None = None, label\_selector: str | None = None, page: int | None = None, per\_page: int | None = None*) → *CertificatesPageResult*

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: Certificate | BoundCertificate*) → *BoundAction*

Returns all action objects for a Certificate.

**Parameters**

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

**Returns**

*BoundAction*

**update**(*certificate: Certificate | BoundCertificate, name: str | None = None, labels: dict[str, str] | None = None*) → *BoundCertificate*

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: *CertificatesClient*, data: dict, complete: bool = True)**delete()** → bool

Deletes a certificate. :return: boolean

**get\_actions**(status: list[str] | None = None, sort: list[str] | None = None) → list[*BoundAction*]

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: list[str] | None = None, sort: list[str] | None = None, page: int | None = None, per\_page: int | None = None) → *ActionsPageResult*

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 *Certificate***retry\_issuance()** → *BoundAction*Retry a failed Certificate issuance or renewal. :return: *BoundAction***update**(name: str | None = None, labels: dict[str, str] | None = None) → *BoundCertificate*

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***BoundCertificate*

```
class Certificate(id: int | None = None, name: str | None = None, certificate: str | None = None,
                 not_valid_before: str | None = None, not_valid_after: str | None = None, domain_names:
                 list[str] | None = None, fingerprint: str | None = None, created: str | None = None, labels:
                 dict[str, str] | None = None, type: str | None = None, status: ManagedCertificateStatus | None
                 = 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: Client)
```

```
get_all(name: str | None = None) → list[BoundDatacenter]
```

Get all datacenters

**Parameters****name** – str (optional) Can be used to filter datacenters by their name.**Returns**List[*BoundDatacenter*]

```
get_by_id(id: int) → BoundDatacenter
```

Get a specific datacenter by its ID.

**Parameters****id** – int**Returns***BoundDatacenter*

**get\_by\_name**(name: str) → *BoundDatacenter* | None

Get datacenter by name

**Parameters**

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

**Returns**

*BoundDatacenter*

**get\_list**(name: str | None = None, page: int | None = None, per\_page: int | None = None) →

*DatacentersPageResult*

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: *DatacentersClient*, data: dict)

**model**

alias of *Datacenter*

**class Datacenter**(id: int | None = None, name: str | None = None, description: str | None = None, location: *Location* | None = None, server\_types: *DatacenterServerTypes* | None = 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: list[*BoundServerType*], supported: list[*BoundServerType*], available\_for\_migration: list[*BoundServerType*])

*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*: `Client`)

**actions**: `ResourceActionsClient`

Firewalls scoped actions client

**Type**

`ResourceActionsClient`

**apply\_to\_resources**(*firewall*: `Firewall` | `BoundFirewall`, *resources*: `list[FirewallResource]`) → `list[BoundAction]`

Applies one Firewall to multiple resources.

**Parameters**

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

**Returns**

`List[BoundAction]`

**create**(*name*: `str`, *rules*: `list[FirewallRule]` | `None` = `None`, *labels*: `str` | `None` = `None`, *resources*: `list[FirewallResource]` | `None` = `None`) → `CreateFirewallResponse`

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*: `Firewall` | `BoundFirewall`) → `bool`

Deletes a Firewall.

**Parameters**

**firewall** – `BoundFirewall` or `Firewall`

**Returns**

`boolean`

**get\_actions**(*firewall*: `Firewall` | `BoundFirewall`, *status*: `list[str]` | `None` = `None`, *sort*: `list[str]` | `None` = `None`) → `list[BoundAction]`

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: Firewall | BoundFirewall, status: list[str] | None = None, sort: list[str] | None = None, page: int | None = None, per\_page: int | None = None*) → ActionsPageResult

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: str | None = None, name: str | None = None, sort: list[str] | None = None*) → list[[BoundFirewall](#)]

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: int*) → [BoundFirewall](#)

Returns a specific Firewall object.

**Parameters**

**id** – int

**Returns**

[BoundFirewall](#)

**get\_by\_name**(*name: str*) → [BoundFirewall](#) | None

Get Firewall by name

**Parameters**

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

**Returns**

[BoundFirewall](#)



**get\_list**(*label\_selector*: str | None = None, *page*: int | None = None, *per\_page*: int | None = None, *name*: str | None = None, *sort*: list[str] | None = None) → FirewallsPageResult

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*: Firewall | BoundFirewall, *resources*: list[FirewallResource]) → list[BoundAction]

Removes one Firewall from multiple resources.

#### Parameters

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

#### Returns

List[BoundAction]

**set\_rules**(*firewall*: Firewall | BoundFirewall, *rules*: list[FirewallRule]) → list[BoundAction]

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*: Firewall | BoundFirewall, *labels*: dict[str, str] | None = None, *name*: str | None = None) → BoundFirewall

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*: FirewallsClient, *data*: dict, *complete*: bool = True)

**apply\_to\_resources**(resources: list[[FirewallResource](#)]) → list[[BoundAction](#)]

Applies one Firewall to multiple resources. :param resources: List[[FirewallResource](#)] :return: List[[BoundAction](#)]

**delete**() → bool

Deletes a Firewall.

**Returns**

boolean

**get\_actions**(status: list[str] | None = None, sort: list[str] | None = None) → list[[BoundAction](#)]

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: list[str] | None = None, sort: list[str] | None = None, page: int | None = None, per\_page: int | None = None) → [ActionsPageResult](#)

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 [Firewall](#)

**remove\_from\_resources**(resources: list[[FirewallResource](#)]) → list[[BoundAction](#)]

Removes one Firewall from multiple resources. :param resources: List[[FirewallResource](#)] :return: List[[BoundAction](#)]

**set\_rules**(rules: list[[FirewallRule](#)]) → list[[BoundAction](#)]

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: str | None = None, labels: dict[str, str] | None = None) → [BoundFirewall](#)

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: int | None = None, name: str | None = None, labels: dict[str, str] | None = None, rules:
    list[FirewallRule] | None = None, applied_to: list[FirewallResource] | None = None, created: str
    | None = 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: str, protocol: str, source_ips: list[str], port: str | None = None, destination_ips:
    list[str] | None = None, description: str | None = 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

**to\_payload()** → dict[str, Any]

Generates the request payload from this domain object.

```
class FirewallResource(type: str, server: Server | BoundServer | None = None, label_selector:
    FirewallResourceLabelSelector | None = None, applied_to_resources:
    list[FirewallResourceAppliedToResources] | None = 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
- **applied\_to\_resources** – (read-only) List of effective resources the firewall is applied to.

**TYPE\_LABEL\_SELECTOR** = 'label\_selector'

Firewall Used By Type label\_selector

**TYPE\_SERVER** = 'server'

Firewall Used By Type Server

**to\_payload()** → dict[str, Any]

Generates the request payload from this domain object.

```
class CreateFirewallResponse(firewall: BoundFirewall, actions: list[BoundAction] | None)
```

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: Client)
```

**actions:** *ResourceActionsClient*

Floating IPs scoped actions client

Type

*ResourceActionsClient*

**assign**(floating\_ip: FloatingIP | BoundFloatingIP, server: Server | BoundServer) → BoundAction

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: FloatingIP | BoundFloatingIP, ip: str, dns\_ptr: str) → BoundAction

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: FloatingIP | BoundFloatingIP, delete: bool | None = None) →

BoundAction

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: str, description: str | None = None, labels: str | None = None, home\_location: Location | BoundLocation | None = None, server: Server | BoundServer | None = None, name: str | None = None) → CreateFloatingIPResponse

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: FloatingIP | BoundFloatingIP) → bool

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: FloatingIP | BoundFloatingIP, status: list[str] | None = None, sort: list[str] | None = None) → list[BoundAction]

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: FloatingIP | BoundFloatingIP, status: list[str] | None = None, sort: list[str] | None = None, page: int | None = None, per\_page: int | None = None) → ActionsPageResult

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: str | None = None, name: str | None = None) → list[BoundFloatingIP]

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*: int) → *BoundFloatingIP*

Returns a specific Floating IP object.

**Parameters****id** – int**Returns***BoundFloatingIP***get\_by\_name**(*name*: str) → *BoundFloatingIP* | None

Get Floating IP by name

**Parameters****name** – str Used to get Floating IP by name.**Returns***BoundFloatingIP***get\_list**(*label\_selector*: str | None = None, *page*: int | None = None, *per\_page*: int | None = None, *name*: str | None = None) → FloatingIPsPageResult

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*: FloatingIP | *BoundFloatingIP*) → *BoundAction*

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*: FloatingIP | *BoundFloatingIP*, *description*: str | None = None, *labels*: dict[str, str] | None = None, *name*: str | None = None) → *BoundFloatingIP*

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: FloatingIPsClient, data: dict, complete: bool = True*)**assign**(*server: Server | BoundServer*) → *BoundAction*

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: str, dns\_ptr: str*) → *BoundAction*

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: bool | None = None*) → *BoundAction*

Changes the protection configuration of the Floating IP.

**Parameters****delete** – boolean If true, prevents the Floating IP from being deleted**Returns***BoundAction***delete**() → bool

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

**Returns**

boolean

**get\_actions**(*status: list[str] | None = None, sort: list[str] | None = None*) → list[*BoundAction*]

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: list[str] | None = None, sort: list[str] | None = None, page: int | None = None, per\_page: int | None = None*) → *ActionsPageResult*

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*
- **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 [FloatingIP](#)**unassign()** → [BoundAction](#)

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: str | None = None, labels: dict[str, str] | None = None, name: str | None = None) → [BoundFloatingIP](#)

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: int | None = None, type: str | None = None, description: str | None = None, ip: str | None = None, server: BoundServer | None = None, dns_ptr: list[dict] | None = None, home_location: BoundLocation | None = None, blocked: bool | None = None, protection: dict | None = None, labels: dict[str, str] | None = None, created: str | None = None, name: str | None = 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: [BoundFloatingIP](#), action: [BoundAction](#) | None)

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: [Client](#))

**actions:** [ResourceActionsClient](#)

Images scoped actions client

**Type**

[ResourceActionsClient](#)

**change\_protection**(image: [Image](#) | [BoundImage](#), delete: bool | None = None) → [BoundAction](#)

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: [Image](#) | [BoundImage](#)) → bool

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

:param [BoundImage](#) or [Image](#) :return: bool

**get\_actions**(image: [Image](#) | [BoundImage](#), sort: list[str] | None = None, status: list[str] | None = None) → list[[BoundAction](#)]

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: [Image](#) | [BoundImage](#), sort: list[str] | None = None, page: int | None = None, per\_page: int | None = None, status: list[str] | None = None) → ActionsPageResult

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: str | None = None, label\_selector: str | None = None, bound\_to: list[str] | None = None, type: list[str] | None = None, architecture: list[str] | None = None, sort: list[str] | None = None, status: list[str] | None = None, include\_deprecated: bool | None = None) → list[[BoundImage](#)]

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*
- **architecture** – List[str] (optional) Choices: *x86 arm*
- **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: int) → [BoundImage](#)

Get a specific Image

#### Parameters

**id** – int

#### Returns

[BoundImage](#) <hcloud.images.client.BoundImage

**get\_by\_name**(name: str) → *BoundImage* | None

Get image by name

**Parameters**

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

**Returns**

*BoundImage*

Deprecated since version 1.19: Use *hcloud.images.client.ImagesClient.get\_by\_name\_and\_architecture()* instead.

**get\_by\_name\_and\_architecture**(name: str, architecture: str, \*, include\_deprecated: bool | None = None) → *BoundImage* | None

Get image by name

**Parameters**

- **name** – str Used to identify the image.
- **architecture** – str Used to identify the image.
- **include\_deprecated** – bool (optional) Include deprecated images. Default: False

**Returns**

*BoundImage*

**get\_list**(name: str | None = None, label\_selector: str | None = None, bound\_to: list[str] | None = None, type: list[str] | None = None, architecture: list[str] | None = None, sort: list[str] | None = None, page: int | None = None, per\_page: int | None = None, status: list[str] | None = None, include\_deprecated: bool | None = None) → *ImagesPageResult*

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
- **architecture** – List[str] (optional) Choices: x86 arm
- **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*: [Image](#) | [BoundImage](#), *description*: *str* | *None* = *None*, *type*: *str* | *None* = *None*, *labels*: *dict*[*str*, *str*] | *None* = *None*) → [BoundImage](#)

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*: [ImagesClient](#), *data*: *dict*)

**change\_protection**(*delete*: *bool* | *None* = *None*) → [BoundAction](#)

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**() → *bool*

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

#### Returns

*bool*

**get\_actions**(*sort*: *list*[*str*] | *None* = *None*, *status*: *list*[*str*] | *None* = *None*) → *list*[[BoundAction](#)]

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*: *list*[*str*] | *None* = *None*, *page*: *int* | *None* = *None*, *per\_page*: *int* | *None* = *None*, *status*: *list*[*str*] | *None* = *None*) → [ActionsPageResult](#)

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 [Image](#)

**update**(*description: str | None = None, type: str | None = None, labels: dict[str, str] | None = None*) → [BoundImage](#)

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: int | None = None, name: str | None = None, type: str | None = None, created: str | None = None, description: str | None = None, image\_size: int | None = None, disk\_size: int | None = None, deprecated: str | None = None, bound\_to: [Server](#) | [BoundServer](#) | None = None, os\_flavor: str | None = None, os\_version: str | None = None, architecture: str | None = None, rapid\_deploy: bool | None = None, created\_from: [Server](#) | [BoundServer](#) | None = None, protection: dict | None = None, labels: dict[str, str] | None = None, status: str | None = 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
- **architecture** – str CPU Architecture that the image is compatible with. Choices: *x86*, *arm*
- **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*: [BoundAction](#), *image*: [BoundImage](#))

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*: [Client](#))

**get\_all**(*name*: str | None = None, *architecture*: list[str] | None = None, *include\_wildcard\_architecture*: bool | None = None, *include\_architecture\_wildcard*: bool | None = None) → list[[BoundIso](#)]

Get all ISOs

#### Parameters

- **name** – str (optional) Can be used to filter ISOs by their name.
- **architecture** – List[str] (optional) Can be used to filter ISOs by their architecture. Choices: *x86* *arm*
- **include\_wildcard\_architecture** – bool (optional) Deprecated, please use *include\_architecture\_wildcard* instead.
- **include\_architecture\_wildcard** – bool (optional) Custom ISOs do not have an architecture set. You must also set this flag to True if you are filtering by architecture and also want custom ISOs.

#### Returns

List[[BoundIso](#)]

**get\_by\_id**(*id*: int) → [BoundIso](#)

Get a specific ISO by its id

#### Parameters

**id** – int

#### Returns

[BoundIso](#)

**get\_by\_name**(*name*: str) → [BoundIso](#) | None

Get iso by name

#### Parameters

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

**Returns***BoundIso*

**get\_list**(*name: str | None = None, architecture: list[str] | None = None, include\_wildcard\_architecture: bool | None = None, include\_architecture\_wildcard: bool | None = None, page: int | None = None, per\_page: int | None = None*) → IsosPageResult

Get a list of ISOs

**Parameters**

- **name** – str (optional) Can be used to filter ISOs by their name.
- **architecture** – List[str] (optional) Can be used to filter ISOs by their architecture. Choices: x86 arm
- **include\_wildcard\_architecture** – bool (optional) Deprecated, please use *include\_architecture\_wildcard* instead.
- **include\_architecture\_wildcard** – bool (optional) Custom ISOs do not have an architecture set. You must also set this flag to True if you are filtering by architecture and also want custom ISOs.
- **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: ClientEntityBase, data: dict, complete: bool = True*)

**model**

alias of *Iso*

**class Iso**(*id: int | None = None, name: str | None = None, type: str | None = None, architecture: str | None = None, description: str | None = None, deprecated: str | None = None, deprecation: dict | None = 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*
- **architecture** – str, None CPU Architecture that the ISO is compatible with. None means that the compatibility is unknown. Choices: *x86, arm*
- **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. This field is deprecated. Use *deprecation* instead.
- **deprecation** – *DeprecationInfo*, None Describes if, when & how the resources was deprecated. If this field is set to None the resource is not deprecated. If it has a value, it is considered deprecated.

**property deprecated:** **datetime | None**

ISO 8601 timestamp of deprecation, None if ISO is still available.



## 4.2.8 LoadBalancerTypesClient

**class** LoadBalancerTypesClient(*client*: Client)

**get\_all**(*name*: str | None = None) → list[BoundLoadBalancerType]

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*: int) → BoundLoadBalancerType

Returns a specific Load Balancer Type.

**Parameters**

**id** – int

**Returns**

BoundLoadBalancerType

**get\_by\_name**(*name*: str) → BoundLoadBalancerType | None

Get Load Balancer type by name

**Parameters**

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

**Returns**

BoundLoadBalancerType

**get\_list**(*name*: str | None = None, *page*: int | None = None, *per\_page*: int | None = None) → LoadBalancerTypesPageResult

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*: int | None = None, *name*: str | None = None, *description*: str | None = None, *max\_connections*: int | None = None, *max\_services*: int | None = None, *max\_targets*: int | None = None, *max\_assigned\_certificates*: int | None = None, *prices*: dict | None = 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*: Client)

**actions:** *ResourceActionsClient*

Load Balancers scoped actions client

**Type**

*ResourceActionsClient*

**add\_service**(*load\_balancer*: LoadBalancer | BoundLoadBalancer, *service*: LoadBalancerService) → *BoundAction*

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*: LoadBalancer | BoundLoadBalancer, *target*: LoadBalancerTarget) → *BoundAction*

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*: LoadBalancer | BoundLoadBalancer, *network*: Network | BoundNetwork, *ip*: str | None = None) → *BoundAction*

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*: [LoadBalancer](#) | [BoundLoadBalancer](#), *algorithm*: [LoadBalancerAlgorithm](#)) → [BoundAction](#)

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*: [LoadBalancer](#) | [BoundLoadBalancer](#), *ip*: *str*, *dns\_ptr*: *str*) → [BoundAction](#)

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*: [LoadBalancer](#) | [BoundLoadBalancer](#), *delete*: *bool* | *None* = *None*) → [BoundAction](#)

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*: [LoadBalancer](#) | [BoundLoadBalancer](#), *load\_balancer\_type*: [LoadBalancerType](#) | [BoundLoadBalancerType](#)) → [BoundAction](#)

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*: *str*, *load\_balancer\_type*: [LoadBalancerType](#) | [BoundLoadBalancerType](#), *algorithm*: [LoadBalancerAlgorithm](#) | *None* = *None*, *services*: *list*[[LoadBalancerService](#)] | *None* = *None*, *targets*: *list*[[LoadBalancerTarget](#)] | *None* = *None*, *labels*: *dict*[*str*, *str*] | *None* = *None*, *location*: [Location](#) | [BoundLocation](#) | *None* = *None*, *network\_zone*: *str* | *None* = *None*, *public\_interface*: *bool* | *None* = *None*, *network*: [Network](#) | [BoundNetwork](#) | *None* = *None*) → [CreateLoadBalancerResponse](#)

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: LoadBalancer | BoundLoadBalancer) → bool

Deletes a Load Balancer.

#### Parameters

**load\_balancer** – BoundLoadBalancer or LoadBalancer

#### Returns

boolean

**delete\_service**(load\_balancer: LoadBalancer | BoundLoadBalancer, service: LoadBalancerService) → BoundAction

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: LoadBalancer | BoundLoadBalancer, network: Network | BoundNetwork) → BoundAction

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: [LoadBalancer](#) | [BoundLoadBalancer](#)) → [BoundAction](#)

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: [LoadBalancer](#) | [BoundLoadBalancer](#)) → [BoundAction](#)

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: [LoadBalancer](#) | [BoundLoadBalancer](#), status: list[str] | None = None, sort: list[str] | None = None) → list[[BoundAction](#)]

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: [LoadBalancer](#) | [BoundLoadBalancer](#), status: list[str] | None = None, sort: list[str] | None = None, page: int | None = None, per\_page: int | None = None) → [ActionsPageResult](#)

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: str | None = None, label\_selector: str | None = None) → list[[BoundLoadBalancer](#)]

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: int) → [BoundLoadBalancer](#)

Get a specific Load Balancer

**Parameters**

**id** – int

**Returns**

[BoundLoadBalancer](#)

**get\_by\_name**(name: str) → [BoundLoadBalancer](#) | None

Get Load Balancer by name

**Parameters**

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

**Returns**

[BoundLoadBalancer](#)

**get\_list**(name: str | None = None, label\_selector: str | None = None, page: int | None = None, per\_page: int | None = None) → LoadBalancersPageResult

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)

**get\_metrics**(load\_balancer: [LoadBalancer](#) | [BoundLoadBalancer](#), type: Literal['open\_connections', 'connections\_per\_second', 'requests\_per\_second', 'bandwidth'] | list[Literal['open\_connections', 'connections\_per\_second', 'requests\_per\_second', 'bandwidth']], start: datetime | str, end: datetime | str, step: float | None = None) → GetMetricsResponse

Get Metrics for a LoadBalancer.

**Parameters**

- **load\_balancer** – The Load Balancer to get the metrics for.
- **type** – Type of metrics to get.
- **start** – Start of period to get Metrics for (in ISO-8601 format).

- **end** – End of period to get Metrics for (in ISO-8601 format).
- **step** – Resolution of results in seconds.

**remove\_target**(*load\_balancer*: *LoadBalancer* | *BoundLoadBalancer*, *target*: *LoadBalancerTarget*) → *BoundAction*

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*: *LoadBalancer* | *BoundLoadBalancer*, *name*: *str* | *None* = *None*, *labels*: *dict[str, str]* | *None* = *None*) → *BoundLoadBalancer*

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*: *LoadBalancer* | *BoundLoadBalancer*, *service*: *LoadBalancerService*) → *BoundAction*

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*: *LoadBalancersClient*, *data*: *dict*, *complete*: *bool* = *True*)

**add\_service**(*service*: *LoadBalancerService*) → *BoundAction*

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*: *LoadBalancerTarget*) → *BoundAction*

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*: *Network* | *BoundNetwork*, *ip*: *str* | *None* = *None*) → *BoundAction*

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*: *LoadBalancerAlgorithm*) → *BoundAction*

Changes the algorithm used by the Load Balancer

**Parameters**

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

**Returns**

*BoundAction*

**change\_dns\_ptr**(*ip*: *str*, *dns\_ptr*: *str*) → *BoundAction*

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*: *bool*) → *BoundAction*

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*: *LoadBalancerType* | *BoundLoadBalancerType*) → *BoundAction*

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()** → bool

Deletes a Load Balancer.

**Returns**

boolean

**delete\_service**(service: [LoadBalancerService](#)) → [BoundAction](#)

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: [Network](#) | [BoundNetwork](#)) → [BoundAction](#)

Detaches a Load Balancer from a Network.

**Parameters**

**network** – [BoundNetwork](#) or [Network](#)

**Returns**

[BoundAction](#)

**disable\_public\_interface**() → [BoundAction](#)

Disables the public interface of a Load Balancer.

**Returns**

[BoundAction](#)

**enable\_public\_interface**() → [BoundAction](#)

Enables the public interface of a Load Balancer.

**Returns**

[BoundAction](#)

**get\_actions**(status: list[str] | None = None, sort: list[str] | None = None) → list[[BoundAction](#)]

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: list[str] | None = None, sort: list[str] | None = None, page: int | None = None, per\_page: int | None = None) → [ActionsPageResult](#)

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)

**get\_metrics**(*type: Literal['open\_connections', 'connections\_per\_second', 'requests\_per\_second', 'bandwidth'], start: datetime | str, end: datetime | str, step: float | None = None*) → GetMetricsResponse

Get Metrics for a LoadBalancer.

**Parameters**

- **type** – Type of metrics to get.
- **start** – Start of period to get Metrics for (in ISO-8601 format).
- **end** – End of period to get Metrics for (in ISO-8601 format).
- **step** – Resolution of results in seconds.

**model**

alias of [LoadBalancer](#)

**remove\_target**(*target: LoadBalancerTarget*) → [BoundAction](#)

Removes a target from a Load Balancer.

**Parameters**

**target** – [LoadBalancerTarget](#) The LoadBalancerTarget you want to remove from the Load Balancer

**Returns**

[BoundAction](#)

**update**(*name: str | None = None, labels: dict[str, str] | None = None*) → [BoundLoadBalancer](#)

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: LoadBalancerService*) → [BoundAction](#)

Updates a service of an Load Balancer.

**Parameters**

**service** – [LoadBalancerService](#) The LoadBalancerService you want to update

**Returns**

[BoundAction](#)

```
class LoadBalancer(id: int, name: str | None = None, public_net: PublicNetwork | None = None, private_net:
    list[PrivateNet] | None = None, location: BoundLocation | None = None, algorithm:
    LoadBalancerAlgorithm | None = None, services: list[LoadBalancerService] | None =
    None, load_balancer_type: BoundLoadBalancerType | None = None, protection: dict |
    None = None, labels: dict[str, str] | None = None, targets: list[LoadBalancerTarget] | None
    = None, created: str | None = None, outgoing_traffic: int | None = None, ingoing_traffic:
    int | None = None, included_traffic: int | None = 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: str | None = None, listen_port: int | None = None, destination_port: int |
    None = None, proxyprotocol: bool | None = None, health_check:
    LoadBalancerHealthCheck | None = None, http: LoadBalancerServiceHttp | None
    = 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

**to\_payload()** → dict[str, Any]

Generates the request payload from this domain object.

```
class LoadBalancerServiceHttp(cookie_name: str | None = None, cookie_lifetime: str | None = None,
                               certificates: list[BoundCertificate] | None = None, redirect_http: bool | None
                               = None, sticky_sessions: bool | None = 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: str | None = None, port: int | None = None, interval: int | None =
                               None, timeout: int | None = None, retries: int | None = None, http:
                               LoadBalancerHealtCheckHttp | None = 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** – LoadBalancerHealtCheckHttp HTTP Config

```
class LoadBalancerHealtCheckHttp(domain: str | None = None, path: str | None = None, response: str | None
                                  = None, status_codes: list | None = None, tls: bool | None = None)
```

LoadBalancerHealtCheckHttp 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: str | None = None, server: BoundServer | None = None, label_selector:
                          LoadBalancerTargetLabelSelector | None = None, ip: LoadBalancerTargetIP | None
                          = None, use_private_ip: bool | None = None, health_status:
                          list[LoadBalancerTargetHealthStatus] | None = 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
- **health\_status** – list List of health statuses of the services on this target. Only present for target types “server” and “ip”.

**to\_payload()** → dict[str, Any]

Generates the request payload from this domain object.

**class LoadBalancerTargetHealthStatus**(*listen\_port: int | None = None, status: str | None = None*)

LoadBalancerTargetHealthStatus Domain

**Parameters**

- **listen\_port** – Load Balancer Target listen port
- **status** – Load Balancer Target status. Choices: healthy, unhealthy, unknown

**class LoadBalancerTargetLabelSelector**(*selector: str | None = None*)

LoadBalancerTargetLabelSelector Domain

**Parameters**

**selector** – str Target label selector

**class LoadBalancerTargetIP**(*ip: str | None = None*)

LoadBalancerTargetIP Domain

**Parameters**

**ip** – str Target IP

**class LoadBalancerAlgorithm**(*type: str | None = None*)

LoadBalancerAlgorithm Domain

**Parameters**

**type** – str Algorithm of the Load Balancer. Choices: round\_robin, least\_connections

## 4.2.10 LocationsClient

**class LocationsClient**(*client: Client*)

**get\_all**(*name: str | None = None*) → list[*BoundLocation*]

Get all locations

**Parameters**

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

**Returns**

List[*BoundLocation*]

**get\_by\_id**(*id: int*) → *BoundLocation*

Get a specific location by its ID.

**Parameters**

**id** – int

**Returns**

*BoundLocation*

**get\_by\_name**(*name: str*) → *BoundLocation* | None

Get location by name

**Parameters**

**name** – str Used to get location by name.

**Returns**

*BoundLocation*

**get\_list**(*name: str | None = None, page: int | None = None, per\_page: int | None = None*) →

*LocationsPageResult*

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: ClientEntityBase, data: dict, complete: bool = True*)

**model**

alias of *Location*

**class Location**(*id: int | None = None, name: str | None = None, description: str | None = None, country: str | None = None, city: str | None = None, latitude: float | None = None, longitude: float | None = None, network\_zone: str | None = 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: Client*)

**actions:** *ResourceActionsClient*

Networks scoped actions client

**Type**

*ResourceActionsClient*

**add\_route**(*network*: [Network](#) | [BoundNetwork](#), *route*: [NetworkRoute](#)) → [BoundAction](#)

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*: [Network](#) | [BoundNetwork](#), *subnet*: [NetworkSubnet](#)) → [BoundAction](#)

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*: [Network](#) | [BoundNetwork](#), *ip\_range*: [str](#)) → [BoundAction](#)

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*: [Network](#) | [BoundNetwork](#), *delete*: [bool](#) | [None](#) = [None](#)) → [BoundAction](#)

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*: [str](#), *ip\_range*: [str](#), *subnets*: [list](#)[[NetworkSubnet](#)] | [None](#) = [None](#), *routes*: [list](#)[[NetworkRoute](#)] | [None](#) = [None](#), *expose\_routes\_to\_vswitch*: [bool](#) | [None](#) = [None](#), *labels*: [dict](#)[[str](#), [str](#)] | [None](#) = [None](#)) → [BoundNetwork](#)

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

- **expose\_routes\_to\_vswitch** – Optional[bool] Indicates if the routes from this network should be exposed to the vSwitch connection. The exposing only takes effect if a vSwitch connection is active.
- **labels** – Dict[str, str] (optional) User-defined labels (key-value pairs)

**Returns***BoundNetwork***delete**(*network*: *Network* | *BoundNetwork*) → bool

Deletes a network.

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

boolean

**delete\_route**(*network*: *Network* | *BoundNetwork*, *route*: *NetworkRoute*) → *BoundAction*

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*: *Network* | *BoundNetwork*, *subnet*: *NetworkSubnet*) → *BoundAction*

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*: *Network* | *BoundNetwork*, *status*: list[str] | None = None, *sort*: list[str] | None = None) → list[*BoundAction*]

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*: *Network* | *BoundNetwork*, *status*: list[str] | None = None, *sort*: list[str] | None = None, *page*: int | None = None, *per\_page*: int | None = None) → *ActionsPageResult*

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: str | None = None, label\_selector: str | None = None) → list[*BoundNetwork*]

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: int) → *BoundNetwork*

Get a specific network

**Parameters****id** – int**Returns***BoundNetwork***get\_by\_name**(name: str) → *BoundNetwork* | None

Get network by name

**Parameters****name** – str Used to get network by name.**Returns***BoundNetwork***get\_list**(name: str | None = None, label\_selector: str | None = None, page: int | None = None, per\_page: int | None = None) → NetworksPageResult

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*: [Network](#) | [BoundNetwork](#), *name*: str | None = None, *expose\_routes\_to\_vswitch*: bool | None = None, *labels*: dict[str, str] | None = None) → [BoundNetwork](#)

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
- **expose\_routes\_to\_vswitch** – Optional[bool] Indicates if the routes from this network should be exposed to the vSwitch connection. The exposing only takes effect if a vSwitch connection is active.
- **labels** – Dict[str, str] (optional) User-defined labels (key-value pairs)

**Returns**[BoundNetwork](#)

**class** [BoundNetwork](#)(*client*: [NetworksClient](#), *data*: dict, *complete*: bool = True)

**add\_route**(*route*: [NetworkRoute](#)) → [BoundAction](#)

Adds a route entry to a network.

**Parameters**

**route** – [NetworkRoute](#) The NetworkRoute you want to add to the Network

**Returns**[BoundAction](#)

**add\_subnet**(*subnet*: [NetworkSubnet](#)) → [BoundAction](#)

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*: str) → [BoundAction](#)

Changes the IP range of a network.

**Parameters**

**ip\_range** – str The new prefix for the whole network.

**Returns**[BoundAction](#)

**change\_protection**(*delete*: bool | None = None) → [BoundAction](#)

Changes the protection configuration of a network.

**Parameters**

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

**Returns**[BoundAction](#)

**delete**() → bool

Deletes a network.

**Returns**

boolean

**delete\_route**(*route*: [NetworkRoute](#)) → [BoundAction](#)

Removes a route entry to a network.

**Parameters****route** – [NetworkRoute](#) The NetworkRoute you want to remove from the Network**Returns**[BoundAction](#)**delete\_subnet**(*subnet*: [NetworkSubnet](#)) → [BoundAction](#)

Removes a subnet entry from a network

**Parameters****subnet** – [NetworkSubnet](#) The NetworkSubnet you want to remove from the Network**Returns**[BoundAction](#)**get\_actions**(*status*: list[str] | None = None, *sort*: list[str] | None = None) → list[[BoundAction](#)]

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*: list[str] | None = None, *sort*: list[str] | None = None, *page*: int | None = None, *per\_page*: int | None = None) → [ActionsPageResult](#)

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 [Network](#)

```
update(name: str | None = None, expose_routes_to_vswitch: bool | None = None, labels: dict[str, str] | None = None) → BoundNetwork
```

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

#### Parameters

- **name** – str (optional) New name to set
- **expose\_routes\_to\_vswitch** – Optional[bool] Indicates if the routes from this network should be exposed to the vSwitch connection. The exposing only takes effect if a vSwitch connection is active.
- **labels** – Dict[str, str] (optional) User-defined labels (key-value pairs)

#### Returns

*BoundNetwork*

```
class Network(id: int, name: str | None = None, created: str | None = None, ip_range: str | None = None, subnets: list[NetworkSubnet] | None = None, routes: list[NetworkRoute] | None = None, expose_routes_to_vswitch: bool | None = None, servers: list[BoundServer] | None = None, protection: dict | None = None, labels: dict[str, str] | None = 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
- **expose\_routes\_to\_vswitch** – bool Indicates if the routes from this network should be exposed to the vSwitch connection.
- **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: str, type: str | None = None, network_zone: str | None = None, gateway: str | None = None, vswitch_id: int | None = 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: str, gateway: str*)

Network Route Domain

#### Parameters

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

**class CreateNetworkResponse**(*network: BoundNetwork, action: BoundAction*)

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: Client*)

**create**(*name: str, type: str, labels: dict[str, str] | None = None*) → *CreatePlacementGroupResponse*

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: PlacementGroup | BoundPlacementGroup*) → bool

Deletes a Placement Group.

#### Parameters

**placement\_group** – *BoundPlacementGroup* or *PlacementGroup*

#### Returns

boolean

**get\_all**(*label\_selector: str | None = None, name: str | None = None, sort: list[str] | None = None*) → list[*BoundPlacementGroup*]

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: int*) → *BoundPlacementGroup*

Returns a specific Placement Group object

#### Parameters

**id** – int

#### Returns

*BoundPlacementGroup*

**get\_by\_name**(*name: str*) → *BoundPlacementGroup* | None

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: str* | None = None, *page: int* | None = None, *per\_page: int* | None = None, *name: str* | None = None, *sort: list[str]* | None = None, *type: str* | None = None) → PlacementGroupsPageResult

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: PlacementGroup* | *BoundPlacementGroup*, *labels: dict[str, str]* | None = None, *name: str* | None = None) → *BoundPlacementGroup*

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: ClientEntityBase*, *data: dict*, *complete: bool* = True)

**delete()** → bool

Deletes a Placement Group

**Returns**

boolean

**model**

alias of *PlacementGroup*

**update**(*labels: dict[str, str] | None = None, name: str | None = None*) → *BoundPlacementGroup*

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: int | None = None, name: str | None = None, labels: dict[str, str] | None = None, servers: list[int] | None = None, type: str | None = None, created: str | None = 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: BoundPlacementGroup, action: BoundAction | None*)

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 PrimaryIPsClient

**class PrimaryIPsClient**(*client: Client*)

**actions:** *ResourceActionsClient*

Primary IPs scoped actions client

**Type**

*ResourceActionsClient*

**assign**(primary\_ip: PrimaryIP | BoundPrimaryIP, assignee\_id: int, assignee\_type: str = 'server') → BoundAction

Assigns a Primary IP to a assignee\_id.

**Parameters**

- **primary\_ip** – BoundPrimaryIP or PrimaryIP
- **assignee\_id** – int Assignee the Primary IP shall be assigned to
- **assignee\_type** – str Assignee the Primary IP shall be assigned to

**Returns**

BoundAction

**change\_dns\_ptr**(primary\_ip: PrimaryIP | BoundPrimaryIP, ip: str, dns\_ptr: str) → BoundAction

Changes the dns ptr that will appear when getting the dns ptr belonging to this Primary IP.

**Parameters**

- **primary\_ip** – BoundPrimaryIP or PrimaryIP
- **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**(primary\_ip: PrimaryIP | BoundPrimaryIP, delete: bool | None = None) → BoundAction

Changes the protection configuration of the Primary IP.

**Parameters**

- **primary\_ip** – BoundPrimaryIP or PrimaryIP
- **delete** – boolean If true, prevents the Primary IP from being deleted

**Returns**

BoundAction

**create**(type: str, datacenter: Datacenter | BoundDatacenter | None, name: str, assignee\_type: str | None = 'server', assignee\_id: int | None = None, auto\_delete: bool | None = False, labels: dict | None = None) → CreatePrimaryIPResponse

Creates a new Primary IP assigned to a server.

**Parameters**

- **type** – str Primary IP type Choices: ipv4, ipv6
- **assignee\_type** – str
- **assignee\_id** – int (optional)
- **datacenter** – Datacenter
- **labels** – Dict[str, str] (optional) User-defined labels (key-value pairs)
- **name** – str
- **auto\_delete** – bool (optional)

**Returns**

CreatePrimaryIPResponse



**delete**(*primary\_ip*: [PrimaryIP](#) | [BoundPrimaryIP](#)) → bool

Deletes a Primary IP. If it is currently assigned to an assignee it will automatically get unassigned.

**Parameters**

**primary\_ip** – [BoundPrimaryIP](#) or [PrimaryIP](#)

**Returns**

boolean

**get\_all**(*label\_selector*: str | None = None, *name*: str | None = None) → list[[BoundPrimaryIP](#)]

Get all primary ips from this account

**Parameters**

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

**Returns**

List[[BoundPrimaryIP](#)]

**get\_by\_id**(*id*: int) → [BoundPrimaryIP](#)

Returns a specific Primary IP object.

**Parameters**

**id** – int

**Returns**

[BoundPrimaryIP](#)

**get\_by\_name**(*name*: str) → [BoundPrimaryIP](#) | None

Get Primary IP by name

**Parameters**

**name** – str Used to get Primary IP by name.

**Returns**

[BoundPrimaryIP](#)

**get\_list**(*label\_selector*: str | None = None, *page*: int | None = None, *per\_page*: int | None = None, *name*: str | None = None, *ip*: str | None = None) → PrimaryIPsPageResult

Get a list of primary ips from this account

**Parameters**

- **label\_selector** – str (optional) Can be used to filter Primary IPs by labels. The response will only contain Primary IPs matching the label selectorable 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.
- **ip** – str (optional) Can be used to filter resources by their ip. The response will only contain the resources matching the specified ip.

**Returns**

(List[[BoundPrimaryIP](#)], Meta)

**unassign**(primary\_ip: PrimaryIP | BoundPrimaryIP) → BoundAction

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

**Parameters**

**primary\_ip** – BoundPrimaryIP or PrimaryIP

**Returns**

BoundAction

**update**(primary\_ip: PrimaryIP | BoundPrimaryIP, auto\_delete: bool | None = None, labels: dict[str, str] | None = None, name: str | None = None) → BoundPrimaryIP

Updates the name, auto\_delete or labels of a Primary IP.

**Parameters**

- **primary\_ip** – BoundPrimaryIP or PrimaryIP
- **auto\_delete** – bool (optional) Delete this Primary IP when the resource it is assigned to is deleted
- **labels** – Dict[str, str] (optional) User-defined labels (key-value pairs)
- **name** – str (optional) New name to set

**Returns**

BoundPrimaryIP

**class BoundPrimaryIP**(client: PrimaryIPsClient, data: dict, complete: bool = True)

**assign**(assignee\_id: int, assignee\_type: str) → BoundAction

Assigns a Primary IP to a assignee.

**Parameters**

- **assignee\_id** – int` Id of an assignee the Primary IP shall be assigned to
- **assignee\_type** – string` Assignee type (e.g server) the Primary IP shall be assigned to

**Returns**

BoundAction

**change\_dns\_ptr**(ip: str, dns\_ptr: str) → BoundAction

Changes the hostname that will appear when getting the hostname belonging to this Primary 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: bool | None = None) → BoundAction

Changes the protection configuration of the Primary IP.

**Parameters**

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

**Returns**

BoundAction

**delete()** → bool

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

**Returns**

boolean

**model**

alias of *PrimaryIP*

**unassign()** → *BoundAction*

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

**Returns**

*BoundAction*

**update**(*auto\_delete: bool | None = None, labels: dict[str, str] | None = None, name: str | None = None*) → *BoundPrimaryIP*

Updates the description or labels of a Primary IP.

**Parameters**

- **auto\_delete** – bool (optional) Auto delete IP when assignee gets deleted
- **labels** – Dict[str, str] (optional) User-defined labels (key-value pairs)
- **name** – str (optional) New Name to set

**Returns**

*BoundPrimaryIP*

```
class PrimaryIP(id: int | None = None, type: str | None = None, ip: str | None = None, dns_ptr: list[dict] | None = None, datacenter: BoundDatacenter | None = None, blocked: bool | None = None, protection: dict | None = None, labels: dict[str, dict] | None = None, created: str | None = None, name: str | None = None, assignee_id: int | None = None, assignee_type: str | None = None, auto_delete: bool | None = None)
```

Primary IP Domain

**Parameters**

- **id** – int ID of the Primary IP
- **ip** – str IP address of the Primary IP
- **type** – str Type of Primary IP. Choices: *ipv4*, *ipv6*
- **dns\_ptr** – List[Dict] Array of reverse DNS entries
- **datacenter** – *Datacenter* Datacenter the Primary IP was created in.
- **blocked** – boolean Whether the IP is blocked
- **protection** – dict Protection configuration for the Primary IP
- **labels** – dict User-defined labels (key-value pairs)
- **created** – datetime Point in time when the Primary IP was created
- **name** – str Name of the Primary IP
- **assignee\_id** – int Assignee ID the Primary IP is assigned to
- **assignee\_type** – str Assignee Type of entity the Primary IP is assigned to
- **auto\_delete** – bool Delete the Primary IP when the Assignee it is assigned to is deleted.

## 4.2.14 ServerTypesClient

**class** `ServerTypesClient(client: Client)`

**get\_all**(name: str | None = None) → list[*BoundServerType*]

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: int) → *BoundServerType*

Returns a specific Server Type.

**Parameters**

**id** – int

**Returns**

*BoundServerType*

**get\_by\_name**(name: str) → *BoundServerType* | None

Get Server type by name

**Parameters**

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

**Returns**

*BoundServerType*

**get\_list**(name: str | None = None, page: int | None = None, per\_page: int | None = None) → *ServerTypesPageResult*

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: ClientEntityBase, data: dict, complete: bool = True)`

**model**

alias of *ServerType*

**class** `ServerType(id: int | None = None, name: str | None = None, description: str | None = None, cores: int | None = None, memory: int | None = None, disk: int | None = None, prices: dict | None = None, storage_type: str | None = None, cpu_type: str | None = None, architecture: str | None = None, deprecated: bool | None = None, deprecation: dict | None = None, included_traffic: int | None = 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*
- **architecture** – string Architecture of cpu. Choices: *x86*, *arm*
- **deprecated** – bool True if server type is deprecated. This field is deprecated. Use *deprecation* instead.
- **deprecation** – *DeprecationInfo*, None Describes if, when & how the resources was deprecated. If this field is set to None the resource is not deprecated. If it has a value, it is considered deprecated.
- **included\_traffic** – int Free traffic per month in bytes

#### 4.2.15 ServersClient

**class** ServersClient(*client*: Client)

**actions:** *ResourceActionsClient*

Servers scoped actions client

**Type**

*ResourceActionsClient*

**add\_to\_placement\_group**(*server*: Server | BoundServer, *placement\_group*: PlacementGroup | BoundPlacementGroup) → *BoundAction*

Adds a server to a placement group.

**Parameters**

- **server** – *BoundServer* or *Server*
- **placement\_group** – *BoundPlacementGroup* or *Network*

**Returns**

*BoundAction*

**attach\_iso**(*server*: Server | BoundServer, *iso*: Iso | BoundIso) → *BoundAction*

Attaches an ISO to a server.

**Parameters**

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

**Returns**

*BoundAction*

**attach\_to\_network**(*server*: [Server](#) | [BoundServer](#), *network*: [Network](#) | [BoundNetwork](#), *ip*: *str* | *None* = *None*, *alias\_ips*: *list[str]* | *None* = *None*) → [BoundAction](#)

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*: [Server](#) | [BoundServer](#), *network*: [Network](#) | [BoundNetwork](#), *alias\_ips*: *list[str]*) → [BoundAction](#)

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*: [Server](#) | [BoundServer](#), *ip*: *str*, *dns\_ptr*: *str* | *None*) → [BoundAction](#)

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*: [Server](#) | [BoundServer](#), *delete*: *bool* | *None* = *None*, *rebuild*: *bool* | *None* = *None*) → [BoundAction](#)

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](#) | [BoundServer](#), *server\_type*: [ServerType](#) | [BoundServerType](#), *upgrade\_disk*: *bool*) → [BoundAction](#)

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*: *str*, *server\_type*: [ServerType](#) | [BoundServerType](#), *image*: [Image](#), *ssh\_keys*: *list*[[SSHKey](#) | [BoundSSHKey](#)] | *None* = *None*, *volumes*: *list*[[Volume](#) | [BoundVolume](#)] | *None* = *None*, *firewalls*: *list*[[Firewall](#) | [BoundFirewall](#)] | *None* = *None*, *networks*: *list*[[Network](#) | [BoundNetwork](#)] | *None* = *None*, *user\_data*: *str* | *None* = *None*, *labels*: *dict*[*str*, *str*] | *None* = *None*, *location*: [Location](#) | [BoundLocation](#) | *None* = *None*, *datacenter*: [Datacenter](#) | [BoundDatacenter](#) | *None* = *None*, *start\_after\_create*: *bool* | *None* = *True*, *automount*: *bool* | *None* = *None*, *placement\_group*: [PlacementGroup](#) | [BoundPlacementGroup](#) | *None* = *None*, *public\_net*: [ServerCreatePublicNetwork](#) | *None* = *None*) → [CreateServerResponse](#)

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

- **public\_net** – *ServerCreatePublicNetwork* Options to configure the public network of a server on creation

**Returns***CreateServerResponse*

**create\_image**(*server*: *Server* | *BoundServer*, *description*: *str* | *None* = *None*, *type*: *str* | *None* = *None*, *labels*: *dict*[*str*, *str*] | *None* = *None*) → *CreateImageResponse*

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*: *Server* | *BoundServer*) → *BoundAction*

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*: *Server* | *BoundServer*, *network*: *Network* | *BoundNetwork*) → *BoundAction*

Detaches a server from a network.

**Parameters**

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

**Returns***BoundAction*

**detach\_iso**(*server*: *Server* | *BoundServer*) → *BoundAction*

Detaches an ISO from a server.

**Parameters**

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

**Returns***BoundAction*

**disable\_backup**(*server*: *Server* | *BoundServer*) → *BoundAction*

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

**Parameters**

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

**Returns***BoundAction*



**disable\_rescue**(*server*: *Server* | *BoundServer*) → *BoundAction*

Disables the Hetzner Rescue System for a server.

**Parameters**

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

**Returns**

*BoundAction*

**enable\_backup**(*server*: *Server* | *BoundServer*) → *BoundAction*

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*: *Server* | *BoundServer*, *type*: *str* | *None* = *None*, *ssh\_keys*: *list[str]* | *None* = *None*) → *EnableRescueResponse*

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*: *Server* | *BoundServer*, *status*: *list[str]* | *None* = *None*, *sort*: *list[str]* | *None* = *None*) → *list[BoundAction]*

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*: *Server* | *BoundServer*, *status*: *list[str]* | *None* = *None*, *sort*: *list[str]* | *None* = *None*, *page*: *int* | *None* = *None*, *per\_page*: *int* | *None* = *None*) → *ActionsPageResult*

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: str | None = None, label\_selector: str | None = None, status: list[str] | None = None) → list[[BoundServer](#)]

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: int) → [BoundServer](#)

Get a specific server

**Parameters**

**id** – int

**Returns**

[BoundServer](#)

**get\_by\_name**(name: str) → [BoundServer](#) | None

Get server by name

**Parameters**

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

**Returns**

[BoundServer](#)

**get\_list**(name: str | None = None, label\_selector: str | None = None, page: int | None = None, per\_page: int | None = None, status: list[str] | None = None) → ServersPageResult

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)

**get\_metrics**(*server*: [Server](#) | [BoundServer](#), *type*: Literal['cpu', 'disk', 'network'] | list[Literal['cpu', 'disk', 'network']], *start*: datetime | str, *end*: datetime | str, *step*: float | None = None) → GetMetricsResponse

Get Metrics for a Server.

**Parameters**

- **server** – The Server to get the metrics for.
- **type** – Type of metrics to get.
- **start** – Start of period to get Metrics for (in ISO-8601 format).
- **end** – End of period to get Metrics for (in ISO-8601 format).
- **step** – Resolution of results in seconds.

**power\_off**(*server*: [Server](#) | [BoundServer](#)) → [BoundAction](#)

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*: [Server](#) | [BoundServer](#)) → [BoundAction](#)

Starts a server by turning its power on.

**Parameters**

**server** – [BoundServer](#) or [Server](#)

**Returns**

[BoundAction](#)

**reboot**(*server*: [Server](#) | [BoundServer](#)) → [BoundAction](#)

Reboots a server gracefully by sending an ACPI request.

**Parameters**

**server** – [BoundServer](#) or [Server](#)

**Returns**

[BoundAction](#)

**rebuild**(*server*: [Server](#) | [BoundServer](#), *image*: [Image](#) | [BoundImage](#), \*, *return\_response*: bool = False) → RebuildResponse | [BoundAction](#)

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

**Parameters**

- **server** – Server to rebuild
- **image** – Image to use for the rebuilt server
- **return\_response** – Whether to return the full response or only the action.

**remove\_from\_placement\_group**(*server*: *Server* | *BoundServer*) → *BoundAction*

Removes a server from a placement group.

**Parameters**

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

**Returns**

*BoundAction*

**request\_console**(*server*: *Server* | *BoundServer*) → *RequestConsoleResponse*

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*: *Server* | *BoundServer*) → *BoundAction*

Cuts power to a server and starts it again.

**Parameters**

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

**Returns**

*BoundAction*

**reset\_password**(*server*: *Server* | *BoundServer*) → *ResetPasswordResponse*

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*: *Server* | *BoundServer*) → *BoundAction*

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

**Parameters**

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

**Returns**

*BoundAction*

**update**(*server*: *Server* | *BoundServer*, *name*: *str* | *None* = *None*, *labels*: *dict*[*str*, *str*] | *None* = *None*) → *BoundServer*

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*: *ServersClient*, *data*: *dict*, *complete*: *bool* = *True*)

**add\_to\_placement\_group**(*placement\_group*: PlacementGroup | BoundPlacementGroup) → BoundAction

Adds a server to a placement group.

**Parameters**

**placement\_group** – BoundPlacementGroup or Network

**Returns**

BoundAction

**attach\_iso**(*iso*: Iso | BoundIso) → BoundAction

Attaches an ISO to a server.

**Parameters**

**iso** – BoundIso or Server

**Returns**

BoundAction

**attach\_to\_network**(*network*: Network | BoundNetwork, *ip*: str | None = None, *alias\_ips*: list[str] | None = None) → BoundAction

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*: Network | BoundNetwork, *alias\_ips*: list[str]) → BoundAction

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*: str, *dns\_ptr*: str | None) → BoundAction

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*: bool | None = None, *rebuild*: bool | None = None) → BoundAction

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*: *ServerType* | *BoundServerType*, *upgrade\_disk*: *bool*) → *BoundAction*

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*: *str* | *None* = *None*, *type*: *str* | *None* = *None*, *labels*: *dict*[*str*, *str*] | *None* = *None*) → *CreateImageResponse*

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**() → *BoundAction*

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

**Returns***BoundAction***detach\_from\_network**(*network*: *Network* | *BoundNetwork*) → *BoundAction*

Detaches a server from a network.

**Parameters****network** – *BoundNetwork* or *Network***Returns***BoundAction***detach\_iso**() → *BoundAction*

Detaches an ISO from a server.

**Returns***BoundAction*

**disable\_backup()** → *BoundAction*

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

**Returns**

*BoundAction*

**disable\_rescue()** → *BoundAction*

Disables the Hetzner Rescue System for a server.

**Returns**

*BoundAction*

**enable\_backup()** → *BoundAction*

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: str | None = None, ssh\_keys: list[str] | None = None*) → *EnableRescueResponse*

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: list[str] | None = None, sort: list[str] | None = None*) → list[*BoundAction*]

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: list[str] | None = None, sort: list[str] | None = None, page: int | None = None, per\_page: int | None = None*) → *ActionsPageResult*

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)

**get\_metrics**(*type*: Literal['cpu', 'disk', 'network'] | list[Literal['cpu', 'disk', 'network']], *start*: datetime | str, *end*: datetime | str, *step*: float | None = None) → GetMetricsResponse

Get Metrics for a Server.

#### Parameters

- **server** – The Server to get the metrics for.
- **type** – Type of metrics to get.
- **start** – Start of period to get Metrics for (in ISO-8601 format).
- **end** – End of period to get Metrics for (in ISO-8601 format).
- **step** – Resolution of results in seconds.

#### model

alias of [Server](#)

**power\_off**() → [BoundAction](#)

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

#### Returns

[BoundAction](#)

**power\_on**() → [BoundAction](#)

Starts a server by turning its power on.

#### Returns

[BoundAction](#)

**reboot**() → [BoundAction](#)

Reboots a server gracefully by sending an ACPI request.

#### Returns

[BoundAction](#)

**rebuild**(*image*: [Image](#) | [BoundImage](#), \*, *return\_response*: bool = False) → RebuildResponse | [BoundAction](#)

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

#### Parameters

- **image** – Image to use for the rebuilt server
- **return\_response** – Whether to return the full response or only the action.

**remove\_from\_placement\_group**() → [BoundAction](#)

Removes a server from a placement group.

#### Returns

[BoundAction](#)



**request\_console()** → *RequestConsoleResponse*

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

**Returns**

*RequestConsoleResponse*

**reset()** → *BoundAction*

Cuts power to a server and starts it again.

**Returns**

*BoundAction*

**reset\_password()** → *ResetPasswordResponse*

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

**Returns**

*ResetPasswordResponse*

**shutdown()** → *BoundAction*

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

**Returns**

*BoundAction*

**update(name: str | None = None, labels: dict[str, str] | None = None)** → *BoundServer*

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: int, name: str | None = None, status: str | None = None, created: str | None = None, public_net:
    PublicNetwork | None = None, server_type: BoundServerType | None = None, datacenter:
    BoundDatacenter | None = None, image: BoundImage | None = None, iso: BoundIso | None =
    None, rescue_enabled: bool | None = None, locked: bool | None = None, backup_window: str |
    None = None, outgoing_traffic: int | None = None, ingoing_traffic: int | None = None,
    included_traffic: int | None = None, protection: dict | None = None, labels: dict[str, str] | None =
    None, volumes: list[BoundVolume] | None = None, private_net: list[PrivateNet] | None = None,
    primary_disk_size: int | None = None, placement_group: BoundPlacementGroup | None = 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
- **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
- **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*] 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: IPv4Address, ipv6: IPv6Network, floating_ips: list[BoundFloatingIP],
                    primary_ipv4: BoundPrimaryIP | None, primary_ipv6: BoundPrimaryIP | None, firewalls:
                    list[PublicNetworkFirewall] | None = None)
```

Public Network Domain

**Parameters**

- **ipv4** – *IPv4Address*

- **ipv6** – [IPv6Network](#)
- **floating\_ips** – List[[BoundFloatingIP](#)]
- **primary\_ipv4** – BoundPrimaryIP
- **primary\_ipv6** – BoundPrimaryIP
- **firewalls** – List[PublicNetworkFirewall]

**class IPv4Address**(ip: str, blocked: bool, dns\_ptr: str)

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: str, blocked: bool, dns\_ptr: list)

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: [BoundServer](#), action: [BoundAction](#), next\_actions: list[[BoundAction](#)], root\_password: str | None)

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 ServerCreatePublicNetwork**(ipv4: [PrimaryIP](#) | None = None, ipv6: [PrimaryIP](#) | None = None, enable\_ipv4: bool = True, enable\_ipv6: bool = True)

Server Create Public Network Domain

#### Parameters

- **ipv4** – Optional[[PrimaryIP](#)]
- **ipv6** – Optional[[PrimaryIP](#)]
- **enable\_ipv4** – bool
- **enable\_ipv6** – bool

**class ResetPasswordResponse**(*action*: [BoundAction](#), *root\_password*: *str*)

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*: [BoundAction](#), *root\_password*: *str*)

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*: [BoundAction](#), *wss\_url*: *str*, *password*: *str*)

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.

## 4.2.16 SSHKeysClient

**class SSHKeysClient**(*client*: [Client](#))

**create**(*name*: *str*, *public\_key*: *str*, *labels*: *dict[str, str] | None = None*) → [BoundSSHKey](#)

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](#)

**delete**(*ssh\_key*: [SSHKey](#) | [BoundSSHKey](#)) → *bool*

Deletes an SSH key. It cannot be used anymore.

**Parameters**

**ssh\_key** – [BoundSSHKey](#) or [SSHKey](#)

**Returns**

*True*

**get\_all**(*name*: *str* | *None = None*, *fingerprint*: *str* | *None = None*, *label\_selector*: *str* | *None = None*) → *list*[[BoundSSHKey](#)]

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: str*) → [BoundSSHKey](#) | None

Get ssh key by fingerprint

**Parameters****fingerprint** – str Used to get ssh key by fingerprint.**Returns**[BoundSSHKey](#)**get\_by\_id**(*id: int*) → [BoundSSHKey](#)

Get a specific SSH Key by its ID

**Parameters****id** – int**Returns**[BoundSSHKey](#)**get\_by\_name**(*name: str*) → [BoundSSHKey](#) | None

Get ssh key by name

**Parameters****name** – str Used to get ssh key by name.**Returns**[BoundSSHKey](#)**get\_list**(*name: str | None = None, fingerprint: str | None = None, label\_selector: str | None = None, page: int | None = None, per\_page: int | None = None*) → SSHKeysPageResult

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*: [SSHKey](#) | [BoundSSHKey](#), *name*: *str* | *None* = *None*, *labels*: *dict*[*str*, *str*] | *None* = *None*) → [BoundSSHKey](#)

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*: *ClientEntityBase*, *data*: *dict*, *complete*: *bool* = *True*)

**delete**() → *bool*

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

**model**

alias of [SSHKey](#)

**update**(*name*: *str* | *None* = *None*, *labels*: *dict*[*str*, *str*] | *None* = *None*) → [BoundSSHKey](#)

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**

[BoundSSHKey](#)

**class** [SSHKey](#)(*id*: *int* | *None* = *None*, *name*: *str* | *None* = *None*, *fingerprint*: *str* | *None* = *None*, *public\_key*: *str* | *None* = *None*, *labels*: *dict*[*str*, *str*] | *None* = *None*, *created*: *str* | *None* = *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

## 4.2.17 VolumesClient

**class VolumesClient**(*client: Client*)

**actions:** *ResourceActionsClient*

Volumes scoped actions client

**Type**

*ResourceActionsClient*

**attach**(*volume: Volume | BoundVolume, server: Server | BoundServer, automount: bool | None = None*) → *BoundAction*

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: Volume | BoundVolume, delete: bool | None = None*) → *BoundAction*

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: int, name: str, labels: str | None = None, location: Location | None = None, server: Server | None = None, automount: bool | None = None, format: str | None = None*) → *CreateVolumeResponse*

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: Volume | BoundVolume*) → bool

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*: *Volume* | *BoundVolume*) → *BoundAction*

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*: *Volume* | *BoundVolume*, *status*: list[str] | None = None, *sort*: list[str] | None = None) → list[*BoundAction*]

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*

**Returns**

List[*BoundAction*]

**get\_actions\_list**(*volume*: *Volume* | *BoundVolume*, *status*: list[str] | None = None, *sort*: list[str] | None = None, *page*: int | None = None, *per\_page*: int | None = None) → *ActionsPageResult*

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*: str | None = None, *status*: list[str] | None = None) → list[*BoundVolume*]

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*: int) → *BoundVolume*

Get a specific volume by its id

**Parameters****id** – int**Returns***BoundVolume***get\_by\_name**(*name*: str) → *BoundVolume* | None

Get volume by name

**Parameters****name** – str Used to get volume by name.**Returns***BoundVolume***get\_list**(*name*: str | None = None, *label\_selector*: str | None = None, *page*: int | None = None, *per\_page*: int | None = None, *status*: list[str] | None = None) → VolumesPageResult

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*: *Volume* | *BoundVolume*, *size*: int) → *BoundAction*

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*: *Volume* | *BoundVolume*, *name*: str | None = None, *labels*: dict[str, str] | None = None) → *BoundVolume*

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: *VolumesClient*, data: dict, complete: bool = True)**attach**(server: *Server* | *BoundServer*, automount: bool | None = None) → *BoundAction*

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: bool | None = None) → *BoundAction*

Changes the protection configuration of a volume.

**Parameters****delete** – boolean If True, prevents the volume from being deleted**Returns***BoundAction***delete**() → bool

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**() → *BoundAction*

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: list[str] | None = None, sort: list[str] | None = None) → list[*BoundAction*]

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*

**Returns**List[*BoundAction*]

**get\_actions\_list**(*status: list[str] | None = None, sort: list[str] | None = None, page: int | None = None, per\_page: int | None = None*) → *ActionsPageResult*

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 [Volume](#)

**resize**(*size: int*) → [BoundAction](#)

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: str | None = None, labels: dict[str, str] | None = None*) → [BoundVolume](#)

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: int, name: str | None = None, server: [Server](#) | [BoundServer](#) | None = None, created: str | None = None, location: [Location](#) | [BoundLocation](#) | None = None, size: int | None = None, linux\_device: str | None = None, format: str | None = None, protection: dict | None = None, labels: dict[str, str] | None = None, status: str | None = 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: [BoundVolume](#), action: [BoundAction](#), next\_actions: list[[BoundAction](#)])

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 HCloudException**

There was an error while using the hcloud library

**class APIException**(code: int | str, message: str | None, details: Any)

There was an error while performing an API Request

**class ActionException**(action: [Action](#) | [BoundAction](#))

A generic action exception

**class ActionFailedException**(action: [Action](#) | [BoundAction](#))

The pending action failed

**class ActionTimeoutException**(action: [Action](#) | [BoundAction](#))

The pending action timed out

## 4.4 Other

### 4.4.1 Helpers

**class LabelValidator**

**static validate**(*labels: dict[str, str]*) → bool

Validates Labels. If you want to know which key/value pair of the dict is not correctly formatted use `validate_verbose()`.

**Returns**

bool

**static validate\_verbose**(*labels: dict[str, str]*) → tuple[bool, str]

Validates Labels and returns the corresponding error message if something is wrong. Returns True, <empty string> if everything is fine.

**Returns**

bool, str

## 4.4.2 Deprecation Info

**class DeprecationInfo**(*announced: str | None = None, unavailable\_after: str | None = None*)

Describes if, when & how the resources was deprecated. If this field is set to `None` the resource is not deprecated. If it has a value, it is considered deprecated.

**Parameters**

- **announced** – datetime Date of when the deprecation was announced.
- **unavailable\_after** – datetime After the time in this field, the resource will not be available from the general listing endpoint of the resource type, and it can not be used in new resources. For example, if this is an image, you can not create new servers with this image after the mentioned date.



## CONTRIBUTING

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. Read the `Development` section in the `README.md`, to setup your development environment.
4. Create a branch for local development:

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

Now you can make your changes locally.

5. 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
```

6. 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.md`.
3. The pull request should work for all the versions of Python the library supports, and for PyPy.



## CHANGELOG

### 6.1 1.35.0 (2024-04-02)

#### 6.1.1 Features

- add `include_deprecated` option when fetching images by name (#375) (6d86f86)

#### 6.1.2 Bug Fixes

- raise warnings for the `ImagesClient.get_by_name` deprecation (#376) (b24de80)

### 6.2 1.34.0 (2024-03-27)

#### 6.2.1 Features

- add `has_id_or_name` to `DomainIdentityMixin` (#373) (8facaf6)

### 6.3 1.33.3 (2024-03-27)

#### 6.3.1 Bug Fixes

- invalid type for load balancer private network property (#372) (903e92f)

#### 6.3.2 Dependencies

- update codecov/codecov-action action to v4 (#359) (a798979)
- update dependency mypy to `>=1.9,<1.10` (#368) (4b9328c)
- update dependency pylint to `>=3,<3.2` (#364) (d71d17f)
- update dependency pytest to `>=8,<8.2` (#366) (8665dcf)
- update dependency pytest to v8 (#357) (f8f756f)
- update dependency pytest-cov to v5 (#371) (04a6a42)
- update dependency watchdog to v4 (#360) (cb8d383)

- update pre-commit hook asottile/pyupgrade to v3.15.1 (#362) (dd2a521)
- update pre-commit hook asottile/pyupgrade to v3.15.2 (3d02ad7)
- update pre-commit hook psf/black-pre-commit-mirror to v24 (#356) (b46397d)
- update pre-commit hook psf/black-pre-commit-mirror to v24.1.1 (#358) (7e4645e)
- update pre-commit hook psf/black-pre-commit-mirror to v24.2.0 (#361) (5b56ace)
- update pre-commit hook psf/black-pre-commit-mirror to v24.3.0 (3bbac5d)
- update pre-commit hook pycqa/flake8 to v7 (#354) (66a582f)
- update pypa/gh-action-pypi-publish action to v1.8.12 (#365) (55db255)
- update pypa/gh-action-pypi-publish action to v1.8.14 (#367) (0cb615f)

## 6.4 1.33.2 (2024-01-02)

### 6.4.1 Bug Fixes

- publish package to PyPI using OIDC auth (1a0e93b)

## 6.5 1.33.1 (2024-01-02)

### 6.5.1 Bug Fixes

- private object not exported in top level module (#346) (5281b05)

### 6.5.2 Dependencies

- update dependency coverage to  $\geq 7.4, < 7.5$  (#348) (3ac5711)
- update dependency mypy to  $\geq 1.8, < 1.9$  (#343) (984022f)
- update pre-commit hook psf/black-pre-commit-mirror to v23.12.1 (#347) (2c24efe)

## 6.6 1.33.0 (2023-12-19)

### 6.6.1 Features

- add metrics endpoint for load balancers and servers (#331) (ee3c54f)

## 6.6.2 Bug Fixes

- fallback to error code when message is unset (#328) (1c94153)

## 6.6.3 Dependencies

- update actions/setup-python action to v5 (#335) (2ac252d)
- update dependency sphinx-rtd-theme to v2 (#330) (7cc4335)
- update pre-commit hook psf/black-pre-commit-mirror to v23.12.0 (#338) (38e4748)
- update pre-commit hook pycqa/isort to v5.13.0 (#336) (3244cfe)
- update pre-commit hook pycqa/isort to v5.13.1 (#337) (020a0ef)
- update pre-commit hook pycqa/isort to v5.13.2 (#339) (b46df8c)

## 6.7 1.32.0 (2023-11-17)

### 6.7.1 Features

- allow returning root\_password in servers rebuild (#276) (38e098a)

### 6.7.2 Dependencies

- update dependency mypy to >=1.7,<1.8 (#325) (7b59a2d)
- update pre-commit hook pre-commit/mirrors-prettier to v3.1.0 (#326) (213b661)
- update pre-commit hook psf/black-pre-commit-mirror to v23.10.1 (#322) (999afe3)
- update pre-commit hook psf/black-pre-commit-mirror to v23.11.0 (#324) (7b2a24e)

## 6.8 1.31.0 (2023-10-23)

### 6.8.1 Features

- prepare for iso deprecated field removal (#320) (beae328)

### 6.8.2 Dependencies

- update pre-commit hook psf/black-pre-commit-mirror to v23.10.0 (#319) (184bbe6)

## 6.9 1.30.0 (2023-10-13)

### 6.9.1 Features

- add deprecation field to Iso (#318) (036b52f)
- support python 3.12 (#311) (7e8cd1d)

### 6.9.2 Dependencies

- update dependency mypy to  $\geq 1.6, < 1.7$  (#317) (d248bbd)
- update dependency pylint to v3 (#307) (277841d)
- update pre-commit hook asottile/pyupgrade to v3.14.0 (#308) (07a4513)
- update pre-commit hook asottile/pyupgrade to v3.15.0 (#312) (c544639)
- update pre-commit hook pre-commit/pre-commit-hooks to v4.5.0 (#313) (e51eaa9)
- update python docker tag to v3.12 (#309) (3a1ee67)

## 6.10 1.29.1 (2023-09-26)

### 6.10.1 Bug Fixes

- prevent api calls when printing bound models (#305) (c1de7ef)

## 6.11 1.29.0 (2023-09-25)

### 6.11.1 Features

- add domain attribute type hints to bound models (#300) (6d46d06)
- **firewalls:** add `applied_to_resources` to `FirewallResource` (#297) (55d2b20)

### 6.11.2 Bug Fixes

- missing `BaseDomain` base class inheritance (#303) (0ee7598)

### 6.11.3 Dependencies

- update actions/checkout action to v4 (#295) (c02b446)
- update dependency sphinx to  $\geq 7.2.2, < 7.3$  (#291) (10234ea)
- update dependency sphinx to v7 (#211) (f635c94)
- update pre-commit hook asottile/pyupgrade to v3.11.0 (#298) (4bbd0cc)
- update pre-commit hook asottile/pyupgrade to v3.11.1 (#299) (2f9fcd7)

- update pre-commit hook asottile/pyupgrade to v3.13.0 (#301) (951dbf3)
- update pre-commit hook pre-commit/mirrors-prettier to v3.0.3 (#294) (381e336)
- update pre-commit hook psf/black to v23.9.1 (#296) (4374a7b)

#### 6.11.4 Documentation

- load token from env in examples scripts (#302) (f18c9a6)

### 6.12 1.28.0 (2023-08-17)

#### 6.12.1 Features

- add load balancer target health status field (#288) (5780418)
- implement resource actions clients (#252) (4bb9a97)

#### 6.12.2 Dependencies

- update dependency coverage to  $\geq 7.3, < 7.4$  (#286) (a4df4fa)
- update dependency mypy to  $\geq 1.5, < 1.6$  (#284) (9dd5c81)
- update pre-commit hook pre-commit/mirrors-prettier to v3.0.2 (#287) (6bf03cb)

#### 6.12.3 Documentation

- fail on warning (#289) (e61300e)

### 6.13 1.27.2 (2023-08-09)

#### 6.13.1 Documentation

- fix python references (#281) (0c0518e)

### 6.14 1.27.1 (2023-08-08)

#### 6.14.1 Bug Fixes

- missing long\_description content\_type in setup.py (#279) (6d79d1d)

## 6.15 1.27.0 (2023-08-08)

### 6.15.1 Features

- add global request timeout option (#271) (07a663f)
- reexport references in parent resources modules (#256) (854c12b)
- the package is now typed (#265) (da8baa5)

### 6.15.2 Bug Fixes

- allow omitting `datacenter` when creating a primary ip (#171) (4375dc6)
- ineffective doc strings (#266) (bb34df9)
- invalid attribute in placement group (#258) (23b3607)

### 6.15.3 Dependencies

- update pre-commit hook asottile/pyupgrade to v3.10.1 (#261) (efa5780)
- update pre-commit hook pre-commit/mirrors-prettier to v3.0.1 (#269) (2239b0b)
- update pre-commit hook pycqa/flake8 to v6.1.0 (#260) (fd01384)

### 6.15.4 Documentation

- update documentation (#247) (e63741f)
- update hetzner logo (#264) (ee79851)

## 6.16 1.26.0 (2023-07-19)

### 6.16.1 Features

- add `repr` method to domains (#246) (4c22765)
- drop support for python 3.7 (#242) (2ce71e9)

## 6.17 1.25.0 (2023-07-14)

### 6.17.1 Features

- add details to raise exceptions (#240) (cf64e54)
- move `hcloud.hcloud` module to `hcloud._client` (#243) (413472d)

### 6.17.2 Dependencies

- update pre-commit hook asottile/pyupgrade to v3.9.0 (#238) (0053ded)
- update pre-commit hook pre-commit/mirrors-prettier to v3 (#235) (047d4e1)
- update pre-commit hook psf/black to v23.7.0 (#239) (443bf26)

## 6.18 1.24.0 (2023-07-03)

### 6.18.1 Features

- revert remove python-dateutil dependency (#231) (945bfde), closes #226

### 6.18.2 Dependencies

- update pre-commit hook asottile/pyupgrade to v3.8.0 (#232) (27f21bc)

## 6.19 1.23.1 (2023-06-30)

### 6.19.1 Bug Fixes

- handle Z timezone in ISO8601 datetime format (#228) (6a5c3f4), closes #226

## 6.20 1.23.0 (2023-06-26)

### 6.20.1 Features

- remove python-dateutil dependency (#221) (8ea4aa0)

### 6.20.2 Bug Fixes

- **isos:** invalid name for include\_wildcard\_architecture argument (#222) (c3dfcab)

### 6.20.3 Dependencies

- update dependency pytest to >=7.4,<7.5 (#217) (11e1f45)

## 6.21 1.22.0 (2023-06-22)

### 6.21.1 Features

- adhere to PEP 517 (#213) (7a19add)
- bump required python version to  $\geq 3.7$  (#198) (62d89f9)
- **network:** add field `expose_routes_to_vswitch` (#208) (5321182)
- setup exception hierarchy (#199) (8466645)

### 6.21.2 Dependencies

- update actions/setup-python action to v4 (#209) (aeec575)
- update actions/stale action to v8 (#210) (cb13230)
- update pre-commit hook asottile/pyupgrade to v3.7.0 (#205) (c46c5a4)

## 6.22 1.21.0 (2023-06-19)

### 6.22.1 Features

- add deprecation field to `ServerType` (#192) (4a0fce7)

### 6.22.2 Bug Fixes

- adjust label validation for max length of 63 characters (#194) (3cba96d)

### 6.22.3 Documentation

- improve branding, design & fix warnings (#191) (47eb9f1)
- use `venv` for the dev setup (#196) (93f48ff)

## 6.23 1.20.0 (2023-05-12)

### 6.23.1 Features

- **server\_type:** add field for included traffic (#185) (8ae0bc6)



## 6.24 v1.19.0 (2023-04-12)

- docs: link to PrivateNet broken by @apricote in #177
- feat: add support for ARM APIs by @apricote in #182

## 6.25 v1.18.2 (2022-12-27)

- fix: remove unused future dependency by @apricote in #173
- chore: update tests to use released python-3.11 by @apricote in #175
- chore: prepare release 1.18.2 by @apricote in #174

## v1.18.1 (2022-10-25)

- Update Github Actions by @LKaemmerling in #165
- Add tests for Python 3.11 by @LKaemmerling in #167

## 6.26 v1.18.0 (2022-08-17)

- Remove use of external mock module by @s-t-e-v-e-n-k in #162
- document installation path via conda-forge by @s-m-e in #149
- Drop # – coding: utf-8 – from files by @jonasdlindner in #154
- Simplify Requirement Constraints by @LKaemmerling in #163
- Add validation helper for Label Values/Keys by @LKaemmerling in #164

## 6.27 v1.17.0 (2022-06-29)

- Add primary IP support by @LKaemmerling in #160

## 6.28 v1.16.0 (2021-08-17)

- Feature: Add support for Load Balancer DNS PTRs

## 6.29 v1.15.0 (2021-08-16)

- Feature: Add support for Placement Groups

## 6.30 v1.14.1 (2021-08-10)

- Bugfix: Fix crash on extra fields in public\_net response
- Improvement: Format code with black

## 6.31 v1.14.0 (2021-08-03)

- Feature: Add support for Firewall rule descriptions

## 6.32 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

## 6.33 v1.12.0 (2021-04-06)

- Feature: Add support for managed Certificates

## 6.34 v1.11.0 (2021-03-11)

- Feature: Add support for Firewalls
- Feature: Add primary\_disk\_size to Server Domain

## 6.35 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)

## 6.36 v1.9.1 (2020-08-11)

- Bugfix: BoundLoadBalancer serialization failed when using IP targets

## 6.37 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`

## 6.38 v1.8.2 (2020-07-20)

- Fix: Loosen up the requirements.

## 6.39 v1.8.1 (2020-06-29)

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

## 6.40 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

## 6.41 1.7.1 (2020-06-15)

- Feature: Add requests 2.23 support

## 6.42 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 `ccloud`

### 6.43 1.6.3 (2020-01-09)

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

### 6.44 1.6.2 (2019-10-15)

- Fix: future dependency requirement was too strict

### 6.45 1.6.1 (2019-10-01)

- Fix: python-dateutil dependency requirement was too strict

### 6.46 1.6.0 (2019-09-17)

- Feature: Add missing get\_by\_name on FloatingIPsClient

### 6.47 1.5.0 (2019-09-16)

- Fix: ServersClient.create\_image fails when specifying the labels
- Feature: Add support for name on Floating IPs

### 6.48 1.4.1 (2019-08-19)

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

### 6.49 1.4.0 (2019-07-29)

- Feature: Add mac\_address to Server PrivateNet domain
- Feature: Add python 3.8 support

## 6.50 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

## 6.51 1.2.1 (2019-03-13)

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

## 6.52 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.

## 6.53 1.1.0 (2019-02-27)

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

## 6.54 1.0.1 (2019-02-22)

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

## 6.55 1.0.0 (2019-02-21)

- First stable release.

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

## 6.56 0.1.0 (2018-12-20)

- First release on GitHub.



## HETZNER CLOUD PYTHON

Official Hetzner Cloud python library.

The library's documentation is available at [hcloud-python.readthedocs.io](https://hcloud-python.readthedocs.io), the public API documentation is available at [docs.hetzner.cloud](https://docs.hetzner.cloud).

### 7.1 Usage

Install the hcloud library:

```
pip install hcloud
```

For more installation details, please see the [installation docs](#).

Here is an example that creates a server and list them:

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

client = Client(token="{YOUR_API_TOKEN}") # Please paste your API token here

# Create a server named my-server
response = client.servers.create(
    name="my-server",
    server_type=ServerType(name="cx11"),
    image=Image(name="ubuntu-22.04"),
)
server = response.server
print(f"{server.id=} {server.name=} {server.status=}")
print(f"root password: {response.root_password}")

# List your servers
servers = client.servers.get_all()
for server in servers:
    print(f"{server.id=} {server.name=} {server.status=}")
```

For more details, please see the [API reference](#).

You can find some more examples under the [examples/](#) directory.

## 7.2 Supported Python versions

We support python versions until [end-of-life](#).

## 7.3 Development

First, create a virtual environment and activate it:

```
make venv
source venv/bin/activate
```

You may setup `pre-commit` to run before you commit changes, this removes the need to run it manually afterwards:

```
pre-commit install
```

You can then run different tasks defined in the `Makefile`, below are the most important ones:

Build the documentation and open it in your browser:

```
make docs
```

Lint the code:

```
make lint
```

Run tests using the current `python3` version:

```
make test
```

You may also run the tests for multiple `python3` versions using `tox`:

```
tox .
```

### 7.3.1 Deprecations implementation

When deprecating a module or a function, you must:

- Update the docstring with a deprecated notice:

```
"""Get image by name

.. deprecated:: 1.19
   Use :func:`hcloud.images.client.ImagesClient.get_by_name_and_architecture` instead.
"""
```

- Raise a warning when the deprecated module or function is being used:

```
warnings.warn(
    "The 'hcloud.images.client.ImagesClient.get_by_name' method is deprecated, please_
↪ use the "
    "'hcloud.images.client.ImagesClient.get_by_name_and_architecture' method instead.",
    DeprecationWarning,
```

(continues on next page)



(continued from previous page)

```
stacklevel=2,  
)
```

## 7.4 License

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



## A

Action (class in *hcloud.actions.domain*), 10  
 ActionException (class in *hcloud.actions.domain*), 86  
 ActionFailedException (class in *hcloud.actions.domain*), 86  
 actions (CertificatesClient attribute), 11  
 actions (Client attribute), 7  
 actions (FirewallsClient attribute), 17  
 actions (FloatingIPsClient attribute), 22  
 actions (ImagesClient attribute), 28  
 actions (LoadBalancersClient attribute), 36  
 actions (NetworksClient attribute), 48  
 actions (PrimaryIPsClient attribute), 57  
 actions (ServersClient attribute), 63  
 actions (VolumesClient attribute), 81  
 ActionsClient (class in *hcloud.actions.client*), 10  
 ActionTimeoutException (class in *hcloud.actions.domain*), 86  
 add\_route() (BoundNetwork method), 52  
 add\_route() (NetworksClient method), 48  
 add\_service() (BoundLoadBalancer method), 41  
 add\_service() (LoadBalancersClient method), 36  
 add\_subnet() (BoundNetwork method), 52  
 add\_subnet() (NetworksClient method), 49  
 add\_target() (BoundLoadBalancer method), 41  
 add\_target() (LoadBalancersClient method), 36  
 add\_to\_placement\_group() (BoundServer method), 70  
 add\_to\_placement\_group() (ServersClient method), 63  
 APIException (class in *hcloud*), 86  
 apply\_to\_resources() (BoundFirewall method), 19  
 apply\_to\_resources() (FirewallsClient method), 17  
 assign() (BoundFloatingIP method), 26  
 assign() (BoundPrimaryIP method), 60  
 assign() (FloatingIPsClient method), 22  
 assign() (PrimaryIPsClient method), 57  
 attach() (BoundVolume method), 84  
 attach() (VolumesClient method), 81  
 attach\_iso() (BoundServer method), 71  
 attach\_iso() (ServersClient method), 63  
 attach\_to\_network() (BoundLoadBalancer method),

42  
 attach\_to\_network() (BoundServer method), 71  
 attach\_to\_network() (LoadBalancersClient method), 36  
 attach\_to\_network() (ServersClient method), 63

## B

BoundAction (class in *hcloud.actions.client*), 10  
 BoundCertificate (class in *hcloud.certificates.client*), 14  
 BoundDatacenter (class in *hcloud.datacenters.client*), 16  
 BoundFirewall (class in *hcloud.firewalls.client*), 19  
 BoundFloatingIP (class in *hcloud.floating\_ips.client*), 26  
 BoundImage (class in *hcloud.images.client*), 31  
 BoundIso (class in *hcloud.isos.client*), 34  
 BoundLoadBalancer (class in *hcloud.load\_balancers.client*), 41  
 BoundLocation (class in *hcloud.locations.client*), 48  
 BoundNetwork (class in *hcloud.networks.client*), 52  
 BoundPlacementGroup (class in *hcloud.placement\_groups.client*), 56  
 BoundPrimaryIP (class in *hcloud.primary\_ips.client*), 60  
 BoundServer (class in *hcloud.servers.client*), 70  
 BoundServerType (class in *hcloud.server\_types.client*), 62  
 BoundSSHKey (class in *hcloud.ssh\_keys.client*), 80  
 BoundVolume (class in *hcloud.volumes.client*), 84

## C

Certificate (class in *hcloud.certificates.domain*), 15  
 certificates (Client attribute), 7  
 CertificatesClient (class in *hcloud.certificates.client*), 11  
 change\_algorithm() (BoundLoadBalancer method), 42  
 change\_algorithm() (LoadBalancersClient method), 36  
 change\_alias\_ips() (BoundServer method), 71  
 change\_alias\_ips() (ServersClient method), 64

- `change_dns_ptr()` (*BoundFloatingIP method*), 26
- `change_dns_ptr()` (*BoundLoadBalancer method*), 42
- `change_dns_ptr()` (*BoundPrimaryIP method*), 60
- `change_dns_ptr()` (*BoundServer method*), 71
- `change_dns_ptr()` (*FloatingIPsClient method*), 23
- `change_dns_ptr()` (*LoadBalancersClient method*), 37
- `change_dns_ptr()` (*PrimaryIPsClient method*), 58
- `change_dns_ptr()` (*ServersClient method*), 64
- `change_ip_range()` (*BoundNetwork method*), 52
- `change_ip_range()` (*NetworksClient method*), 49
- `change_protection()` (*BoundFloatingIP method*), 26
- `change_protection()` (*BoundImage method*), 31
- `change_protection()` (*BoundLoadBalancer method*), 42
- `change_protection()` (*BoundNetwork method*), 52
- `change_protection()` (*BoundPrimaryIP method*), 60
- `change_protection()` (*BoundServer method*), 71
- `change_protection()` (*BoundVolume method*), 84
- `change_protection()` (*FloatingIPsClient method*), 23
- `change_protection()` (*ImagesClient method*), 28
- `change_protection()` (*LoadBalancersClient method*), 37
- `change_protection()` (*NetworksClient method*), 49
- `change_protection()` (*PrimaryIPsClient method*), 58
- `change_protection()` (*ServersClient method*), 64
- `change_protection()` (*VolumesClient method*), 81
- `change_type()` (*BoundLoadBalancer method*), 42
- `change_type()` (*BoundServer method*), 72
- `change_type()` (*LoadBalancersClient method*), 37
- `change_type()` (*ServersClient method*), 64
- `Client` (*class in hcloud*), 7
- `create()` (*CertificatesClient method*), 11
- `create()` (*FirewallsClient method*), 17
- `create()` (*FloatingIPsClient method*), 23
- `create()` (*LoadBalancersClient method*), 37
- `create()` (*NetworksClient method*), 49
- `create()` (*PlacementGroupsClient method*), 55
- `create()` (*PrimaryIPsClient method*), 58
- `create()` (*ServersClient method*), 65
- `create()` (*SSHKeysClient method*), 78
- `create()` (*VolumesClient method*), 81
- `create_image()` (*BoundServer method*), 72
- `create_image()` (*ServersClient method*), 66
- `create_managed()` (*CertificatesClient method*), 11
- `CreateFirewallResponse` (*class in hcloud.firewalls.domain*), 22
- `CreateFloatingIPResponse` (*class in hcloud.floating\_ips.domain*), 28
- `CreateImageResponse` (*class in hcloud.images.domain*), 33
- `CreateNetworkResponse` (*class in hcloud.networks.domain*), 55
- `CreatePlacementGroupResponse` (*class in hcloud.placement\_groups.domain*), 57
- `CreateServerResponse` (*class in hcloud.servers.domain*), 77
- `CreateVolumeResponse` (*class in hcloud.volumes.domain*), 86
- D**
- `Datacenter` (*class in hcloud.datacenters.domain*), 16
- `datacenters` (*Client attribute*), 7
- `DatacentersClient` (*class in hcloud.datacenters.client*), 15
- `DatacenterServerTypes` (*class in hcloud.datacenters.domain*), 16
- `delete()` (*BoundCertificate method*), 14
- `delete()` (*BoundFirewall method*), 20
- `delete()` (*BoundFloatingIP method*), 26
- `delete()` (*BoundImage method*), 31
- `delete()` (*BoundLoadBalancer method*), 42
- `delete()` (*BoundNetwork method*), 52
- `delete()` (*BoundPlacementGroup method*), 56
- `delete()` (*BoundPrimaryIP method*), 60
- `delete()` (*BoundServer method*), 72
- `delete()` (*BoundSSHKey method*), 80
- `delete()` (*BoundVolume method*), 84
- `delete()` (*CertificatesClient method*), 12
- `delete()` (*FirewallsClient method*), 17
- `delete()` (*FloatingIPsClient method*), 23
- `delete()` (*ImagesClient method*), 28
- `delete()` (*LoadBalancersClient method*), 38
- `delete()` (*NetworksClient method*), 50
- `delete()` (*PlacementGroupsClient method*), 55
- `delete()` (*PrimaryIPsClient method*), 58
- `delete()` (*ServersClient method*), 66
- `delete()` (*SSHKeysClient method*), 78
- `delete()` (*VolumesClient method*), 81
- `delete_route()` (*BoundNetwork method*), 53
- `delete_route()` (*NetworksClient method*), 50
- `delete_service()` (*BoundLoadBalancer method*), 43
- `delete_service()` (*LoadBalancersClient method*), 38
- `delete_subnet()` (*BoundNetwork method*), 53
- `delete_subnet()` (*NetworksClient method*), 50
- `deprecated` (*Iso property*), 34
- `DeprecationInfo` (*class in hcloud.deprecation.domain*), 87
- `detach()` (*BoundVolume method*), 84
- `detach()` (*VolumesClient method*), 82
- `detach_from_network()` (*BoundLoadBalancer method*), 43
- `detach_from_network()` (*BoundServer method*), 72
- `detach_from_network()` (*LoadBalancersClient method*), 38
- `detach_from_network()` (*ServersClient method*), 66
- `detach_iso()` (*BoundServer method*), 72
- `detach_iso()` (*ServersClient method*), 66
- `DIRECTION_IN` (*FirewallRule attribute*), 21

DIRECTION\_OUT (*FirewallRule* attribute), 21  
 disable\_backup() (*BoundServer* method), 72  
 disable\_backup() (*ServersClient* method), 66  
 disable\_public\_interface() (*BoundLoadBalancer*  
   method), 43  
 disable\_public\_interface() (*LoadBalancersClient*  
   method), 38  
 disable\_rescue() (*BoundServer* method), 73  
 disable\_rescue() (*ServersClient* method), 66

## E

enable\_backup() (*BoundServer* method), 73  
 enable\_backup() (*ServersClient* method), 67  
 enable\_public\_interface() (*BoundLoadBalancer*  
   method), 43  
 enable\_public\_interface() (*LoadBalancersClient*  
   method), 39  
 enable\_rescue() (*BoundServer* method), 73  
 enable\_rescue() (*ServersClient* method), 67  
 EnableRescueResponse (class in  
   *hcloud.servers.domain*), 78

## F

Firewall (class in *hcloud.firewalls.domain*), 21  
 FirewallResource (class in *hcloud.firewalls.domain*),  
   22  
 FirewallRule (class in *hcloud.firewalls.domain*), 21  
 firewalls (Client attribute), 7  
 FirewallsClient (class in *hcloud.firewalls.client*), 17  
 floating\_ips (Client attribute), 7  
 FloatingIP (class in *hcloud.floating\_ips.domain*), 27  
 FloatingIPsClient (class in  
   *hcloud.floating\_ips.client*), 22

## G

get\_actions() (*BoundCertificate* method), 14  
 get\_actions() (*BoundFirewall* method), 20  
 get\_actions() (*BoundFloatingIP* method), 26  
 get\_actions() (*BoundImage* method), 31  
 get\_actions() (*BoundLoadBalancer* method), 43  
 get\_actions() (*BoundNetwork* method), 53  
 get\_actions() (*BoundServer* method), 73  
 get\_actions() (*BoundVolume* method), 84  
 get\_actions() (*CertificatesClient* method), 12  
 get\_actions() (*FirewallsClient* method), 17  
 get\_actions() (*FloatingIPsClient* method), 24  
 get\_actions() (*ImagesClient* method), 28  
 get\_actions() (*LoadBalancersClient* method), 39  
 get\_actions() (*NetworksClient* method), 50  
 get\_actions() (*ServersClient* method), 67  
 get\_actions() (*VolumesClient* method), 82  
 get\_actions\_list() (*BoundCertificate* method), 14  
 get\_actions\_list() (*BoundFirewall* method), 20  
 get\_actions\_list() (*BoundFloatingIP* method), 26

get\_actions\_list() (*BoundImage* method), 31  
 get\_actions\_list() (*BoundLoadBalancer* method),  
   43  
 get\_actions\_list() (*BoundNetwork* method), 53  
 get\_actions\_list() (*BoundServer* method), 73  
 get\_actions\_list() (*BoundVolume* method), 84  
 get\_actions\_list() (*CertificatesClient* method), 12  
 get\_actions\_list() (*FirewallsClient* method), 18  
 get\_actions\_list() (*FloatingIPsClient* method), 24  
 get\_actions\_list() (*ImagesClient* method), 28  
 get\_actions\_list() (*LoadBalancersClient* method),  
   39  
 get\_actions\_list() (*NetworksClient* method), 50  
 get\_actions\_list() (*ServersClient* method), 67  
 get\_actions\_list() (*VolumesClient* method), 82  
 get\_all() (*ActionsClient* method), 10  
 get\_all() (*CertificatesClient* method), 12  
 get\_all() (*DatacentersClient* method), 15  
 get\_all() (*FirewallsClient* method), 18  
 get\_all() (*FloatingIPsClient* method), 24  
 get\_all() (*ImagesClient* method), 29  
 get\_all() (*IsosClient* method), 33  
 get\_all() (*LoadBalancersClient* method), 39  
 get\_all() (*LoadBalancerTypesClient* method), 35  
 get\_all() (*LocationsClient* method), 47  
 get\_all() (*NetworksClient* method), 51  
 get\_all() (*PlacementGroupsClient* method), 55  
 get\_all() (*PrimaryIPsClient* method), 59  
 get\_all() (*ResourceActionsClient* method), 9  
 get\_all() (*ServersClient* method), 68  
 get\_all() (*ServerTypesClient* method), 62  
 get\_all() (*SSHKeysClient* method), 78  
 get\_all() (*VolumesClient* method), 82  
 get\_by\_fingerprint() (*SSHKeysClient* method), 79  
 get\_by\_id() (*ActionsClient* method), 10  
 get\_by\_id() (*CertificatesClient* method), 13  
 get\_by\_id() (*DatacentersClient* method), 15  
 get\_by\_id() (*FirewallsClient* method), 18  
 get\_by\_id() (*FloatingIPsClient* method), 25  
 get\_by\_id() (*ImagesClient* method), 29  
 get\_by\_id() (*IsosClient* method), 33  
 get\_by\_id() (*LoadBalancersClient* method), 40  
 get\_by\_id() (*LoadBalancerTypesClient* method), 35  
 get\_by\_id() (*LocationsClient* method), 47  
 get\_by\_id() (*NetworksClient* method), 51  
 get\_by\_id() (*PlacementGroupsClient* method), 56  
 get\_by\_id() (*PrimaryIPsClient* method), 59  
 get\_by\_id() (*ResourceActionsClient* method), 9  
 get\_by\_id() (*ServersClient* method), 68  
 get\_by\_id() (*ServerTypesClient* method), 62  
 get\_by\_id() (*SSHKeysClient* method), 79  
 get\_by\_id() (*VolumesClient* method), 83  
 get\_by\_name() (*CertificatesClient* method), 13  
 get\_by\_name() (*DatacentersClient* method), 15

[get\\_by\\_name\(\)](#) (*FirewallsClient* method), 18  
[get\\_by\\_name\(\)](#) (*FloatingIPsClient* method), 25  
[get\\_by\\_name\(\)](#) (*ImagesClient* method), 29  
[get\\_by\\_name\(\)](#) (*IsosClient* method), 33  
[get\\_by\\_name\(\)](#) (*LoadBalancersClient* method), 40  
[get\\_by\\_name\(\)](#) (*LoadBalancerTypesClient* method), 35  
[get\\_by\\_name\(\)](#) (*LocationsClient* method), 47  
[get\\_by\\_name\(\)](#) (*NetworksClient* method), 51  
[get\\_by\\_name\(\)](#) (*PlacementGroupsClient* method), 56  
[get\\_by\\_name\(\)](#) (*PrimaryIPsClient* method), 59  
[get\\_by\\_name\(\)](#) (*ServersClient* method), 68  
[get\\_by\\_name\(\)](#) (*ServerTypesClient* method), 62  
[get\\_by\\_name\(\)](#) (*SSHKeysClient* method), 79  
[get\\_by\\_name\(\)](#) (*VolumesClient* method), 83  
[get\\_by\\_name\\_and\\_architecture\(\)](#) (*ImagesClient* method), 30  
[get\\_list\(\)](#) (*ActionsClient* method), 10  
[get\\_list\(\)](#) (*CertificatesClient* method), 13  
[get\\_list\(\)](#) (*DatacentersClient* method), 16  
[get\\_list\(\)](#) (*FirewallsClient* method), 18  
[get\\_list\(\)](#) (*FloatingIPsClient* method), 25  
[get\\_list\(\)](#) (*ImagesClient* method), 30  
[get\\_list\(\)](#) (*IsosClient* method), 34  
[get\\_list\(\)](#) (*LoadBalancersClient* method), 40  
[get\\_list\(\)](#) (*LoadBalancerTypesClient* method), 35  
[get\\_list\(\)](#) (*LocationsClient* method), 48  
[get\\_list\(\)](#) (*NetworksClient* method), 51  
[get\\_list\(\)](#) (*PlacementGroupsClient* method), 56  
[get\\_list\(\)](#) (*PrimaryIPsClient* method), 59  
[get\\_list\(\)](#) (*ResourceActionsClient* method), 9  
[get\\_list\(\)](#) (*ServersClient* method), 68  
[get\\_list\(\)](#) (*ServerTypesClient* method), 62  
[get\\_list\(\)](#) (*SSHKeysClient* method), 79  
[get\\_list\(\)](#) (*VolumesClient* method), 83  
[get\\_metrics\(\)](#) (*BoundLoadBalancer* method), 44  
[get\\_metrics\(\)](#) (*BoundServer* method), 74  
[get\\_metrics\(\)](#) (*LoadBalancersClient* method), 40  
[get\\_metrics\(\)](#) (*ServersClient* method), 69

## H

[HCloudException](#) (class in *hcloud*), 86

## I

[Image](#) (class in *hcloud.images.domain*), 32  
[images](#) (Client attribute), 7  
[ImagesClient](#) (class in *hcloud.images.client*), 28  
[IPv4Address](#) (class in *hcloud.servers.domain*), 77  
[IPv6Network](#) (class in *hcloud.servers.domain*), 77  
[Iso](#) (class in *hcloud.isos.domain*), 34  
[isos](#) (Client attribute), 7  
[IsosClient](#) (class in *hcloud.isos.client*), 33

## L

[LabelValidator](#) (class in *hcloud.helpers.labels*), 86

[load\\_balancer\\_types](#) (Client attribute), 8  
[load\\_balancers](#) (Client attribute), 8  
[LoadBalancer](#) (class in *hcloud.load\_balancers.domain*), 44  
[LoadBalancerAlgorithm](#) (class in *hcloud.load\_balancers.domain*), 47  
[LoadBalancerHealthCheckHttp](#) (class in *hcloud.load\_balancers.domain*), 46  
[LoadBalancerHealthCheck](#) (class in *hcloud.load\_balancers.domain*), 46  
[LoadBalancersClient](#) (class in *hcloud.load\_balancers.client*), 36  
[LoadBalancerService](#) (class in *hcloud.load\_balancers.domain*), 45  
[LoadBalancerServiceHttp](#) (class in *hcloud.load\_balancers.domain*), 45  
[LoadBalancerTarget](#) (class in *hcloud.load\_balancers.domain*), 46  
[LoadBalancerTargetHealthStatus](#) (class in *hcloud.load\_balancers.domain*), 47  
[LoadBalancerTargetIP](#) (class in *hcloud.load\_balancers.domain*), 47  
[LoadBalancerTargetLabelSelector](#) (class in *hcloud.load\_balancers.domain*), 47  
[LoadBalancerType](#) (class in *hcloud.load\_balancer\_types.domain*), 35  
[LoadBalancerTypesClient](#) (class in *hcloud.load\_balancer\_types.client*), 35  
[Location](#) (class in *hcloud.locations.domain*), 48  
[locations](#) (Client attribute), 8  
[LocationsClient](#) (class in *hcloud.locations.client*), 47

## M

[model](#) (*BoundAction* attribute), 10  
[model](#) (*BoundCertificate* attribute), 14  
[model](#) (*BoundDatacenter* attribute), 16  
[model](#) (*BoundFirewall* attribute), 20  
[model](#) (*BoundFloatingIP* attribute), 27  
[model](#) (*BoundImage* attribute), 32  
[model](#) (*BoundIso* attribute), 34  
[model](#) (*BoundLoadBalancer* attribute), 44  
[model](#) (*BoundLocation* attribute), 48  
[model](#) (*BoundNetwork* attribute), 53  
[model](#) (*BoundPlacementGroup* attribute), 57  
[model](#) (*BoundPrimaryIP* attribute), 61  
[model](#) (*BoundServer* attribute), 74  
[model](#) (*BoundServerType* attribute), 62  
[model](#) (*BoundSSHKey* attribute), 80  
[model](#) (*BoundVolume* attribute), 85

## N

[Network](#) (class in *hcloud.networks.domain*), 54  
[NetworkRoute](#) (class in *hcloud.networks.domain*), 55  
[networks](#) (Client attribute), 8



NetworksClient (class in *hcloud.networks.client*), 48  
 NetworkSubnet (class in *hcloud.networks.domain*), 54

## P

placement\_groups (Client attribute), 8  
 PlacementGroup (class in *hcloud.placement\_groups.domain*), 57  
 PlacementGroupsClient (class in *hcloud.placement\_groups.client*), 55  
 power\_off() (BoundServer method), 74  
 power\_off() (ServersClient method), 69  
 power\_on() (BoundServer method), 74  
 power\_on() (ServersClient method), 69  
 primary\_ips (Client attribute), 8  
 PrimaryIP (class in *hcloud.primary\_ips.domain*), 61  
 PrimaryIPsClient (class in *hcloud.primary\_ips.client*), 57  
 PROTOCOL\_ESP (FirewallRule attribute), 21  
 PROTOCOL\_GRE (FirewallRule attribute), 21  
 PROTOCOL\_ICMP (FirewallRule attribute), 21  
 PROTOCOL\_TCP (FirewallRule attribute), 22  
 PROTOCOL\_UDP (FirewallRule attribute), 22  
 PublicNetwork (class in *hcloud.servers.domain*), 76

## R

reboot() (BoundServer method), 74  
 reboot() (ServersClient method), 69  
 rebuild() (BoundServer method), 74  
 rebuild() (ServersClient method), 69  
 remove\_from\_placement\_group() (BoundServer method), 74  
 remove\_from\_placement\_group() (ServersClient method), 69  
 remove\_from\_resources() (BoundFirewall method), 20  
 remove\_from\_resources() (FirewallsClient method), 19  
 remove\_target() (BoundLoadBalancer method), 44  
 remove\_target() (LoadBalancersClient method), 41  
 request() (Client method), 8  
 request\_console() (BoundServer method), 74  
 request\_console() (ServersClient method), 70  
 RequestConsoleResponse (class in *hcloud.servers.domain*), 78  
 reset() (BoundServer method), 75  
 reset() (ServersClient method), 70  
 reset\_password() (BoundServer method), 75  
 reset\_password() (ServersClient method), 70  
 ResetPasswordResponse (class in *hcloud.servers.domain*), 77  
 resize() (BoundVolume method), 85  
 resize() (VolumesClient method), 83  
 ResourceActionsClient (class in *hcloud.actions.client*), 9

retry\_issuance() (BoundCertificate method), 14  
 retry\_issuance() (CertificatesClient method), 13

## S

Server (class in *hcloud.servers.domain*), 75  
 server\_types (Client attribute), 8  
 ServerCreatePublicNetwork (class in *hcloud.servers.domain*), 77  
 servers (Client attribute), 8  
 ServersClient (class in *hcloud.servers.client*), 63  
 ServerType (class in *hcloud.server\_types.domain*), 62  
 ServerTypesClient (class in *hcloud.server\_types.client*), 62  
 set\_rules() (BoundFirewall method), 20  
 set\_rules() (FirewallsClient method), 19  
 shutdown() (BoundServer method), 75  
 shutdown() (ServersClient method), 70  
 ssh\_keys (Client attribute), 9  
 SSHKey (class in *hcloud.ssh\_keys.domain*), 80  
 SSHKeysClient (class in *hcloud.ssh\_keys.client*), 78  
 STATUS\_AVAILABLE (Volume attribute), 86  
 STATUS\_CREATING (Volume attribute), 86  
 STATUS\_DELETING (Server attribute), 76  
 STATUS\_ERROR (Action attribute), 11  
 STATUS\_INIT (Server attribute), 76  
 STATUS\_MIGRATING (Server attribute), 76  
 STATUS\_OFF (Server attribute), 76  
 STATUS\_REBUILDING (Server attribute), 76  
 STATUS\_RUNNING (Action attribute), 11  
 STATUS\_RUNNING (Server attribute), 76  
 STATUS\_STARTING (Server attribute), 76  
 STATUS\_STOPPING (Server attribute), 76  
 STATUS\_SUCCESS (Action attribute), 11  
 STATUS\_UNKNOWN (Server attribute), 76

## T

to\_payload() (FirewallResource method), 22  
 to\_payload() (FirewallRule method), 22  
 to\_payload() (LoadBalancerService method), 45  
 to\_payload() (LoadBalancerTarget method), 47  
 TYPE\_CLOUD (NetworkSubnet attribute), 54  
 TYPE\_LABEL\_SELECTOR (FirewallResource attribute), 22  
 TYPE\_SERVER (FirewallResource attribute), 22  
 TYPE\_SERVER (NetworkSubnet attribute), 54  
 TYPE\_VSWITCH (NetworkSubnet attribute), 55

## U

unassign() (BoundFloatingIP method), 27  
 unassign() (BoundPrimaryIP method), 61  
 unassign() (FloatingIPsClient method), 25  
 unassign() (PrimaryIPsClient method), 59  
 update() (BoundCertificate method), 14  
 update() (BoundFirewall method), 20  
 update() (BoundFloatingIP method), 27

`update()` (*BoundImage method*), 32  
`update()` (*BoundLoadBalancer method*), 44  
`update()` (*BoundNetwork method*), 53  
`update()` (*BoundPlacementGroup method*), 57  
`update()` (*BoundPrimaryIP method*), 61  
`update()` (*BoundServer method*), 75  
`update()` (*BoundSSHKey method*), 80  
`update()` (*BoundVolume method*), 85  
`update()` (*CertificatesClient method*), 13  
`update()` (*FirewallsClient method*), 19  
`update()` (*FloatingIPsClient method*), 25  
`update()` (*ImagesClient method*), 30  
`update()` (*LoadBalancersClient method*), 41  
`update()` (*NetworksClient method*), 52  
`update()` (*PlacementGroupsClient method*), 56  
`update()` (*PrimaryIPsClient method*), 60  
`update()` (*ServersClient method*), 70  
`update()` (*SSHKeysClient method*), 79  
`update()` (*VolumesClient method*), 83  
`update_service()` (*BoundLoadBalancer method*), 44  
`update_service()` (*LoadBalancersClient method*), 41

## V

`validate()` (*LabelValidator static method*), 86  
`validate_verbose()` (*LabelValidator static method*),  
87  
`Volume` (*class in hcloud.volumes.domain*), 85  
`volumes` (*Client attribute*), 9  
`VolumesClient` (*class in hcloud.volumes.client*), 81

## W

`wait_until_finished()` (*BoundAction method*), 10