ProfitBricks

API Documentation V1.3

Table of contents 2
Overview 5
How to Access the API 5
Version History 6
Fundamentals 7
FTP Upload Settings 10
Resource Limits 12
User Management 13
API Outline 14
Authentication 14
SOAP Request 14
SOAP Response 15
SOAP Fault & Error Codes 16
API Methods 19
User Notifications 19
Get Notifications 19
Delete Notifications 20
Virtual Data Center Operations 21
Create Data Center 21
Get Data Center by ID 22
Get Data Center State 23
Get All Data Centers 24
Update Data Center 24
Clear Data Center 25
Delete Data Center 25
Server Operations 26
Create Server 27
Get Server 29
Get All Servers 31

Update Server 33
Delete Server 34
Reset Server 35
Start Server 36
Stop Server 36
Storage Operations 37
Create Storage 37
Get Storage 38
Get All Storages 39
Connect Storage 40
Disconnect Storage 40
Update Storage 41
Delete Storage 42
Snapshot Operations 42
Create Snapshot 43
Get Snapshot 43
Get All Snapshots 44
Update Snapshot 45
Delete Snapshot 47
Rollback Snapshot 47
Load Balancer Operations 48
Create Load Balancer 48
Get Load Balancer 50
Get All Load Balancer 51
Update Load Balancer 52
Register Servers On Load Balancer 53
Deregister Servers On Load Balancer 53
Delete Load Balancer 54
Firewall Operations 55
Add Firewall Rules 55
Get Firewall 57

Get All Firewalls 58
Remove Firewall Rules 59
Activate/Deactivate Firewall 59
Delete Firewall 60
CD-ROM/DVD Drive Operations 60
Add CD-ROM/DVD Drive To Server 60
Remove CD-ROM/DVD Drive From Server 61
Image Operations 62
Set Image OS Type 62
Get Image 63
Get All Images 64
Update Image 65
Delete Image 66
NIC Operations 66
Create NIC 67
Get NIC 68
Get All NIC 69
Set Internet Access 70
Update NIC 70
Delete NIC 71
Public IP Operations 72
Reserve Public IP Block 72
Add Public IP To NIC 73
Get Public IP Block 74
Get All Public IP Blocks 75
Remove Public IP From Nic 75
Release Public IP Block 75
Legal Notice 77

OVERVIEW

ProfitBricks offers professional IaaS solutions that can be automatically managed through our browser-based “Datacenter Designer” (DCD) tool as well as a through a secure and easy to use SOAP based API.

ProfitBricks API allows the customer to develop applications for automating the management of their virtual resources.

The API is designed to let the end user leverage the same power and flexibility offered by our Data Center Designer (DCD) visual tool while at the same time taking advantage of the complete automation of cloud computing environments.

As both tools make use of the same concepts pertaining to our powerful virtual datacenter model, the API learning experience stays as smooth and intuitive as possible.

Through the use of such interface, in conjunction with monitoring solutions that the customer can deploy on their virtual infrastructure, full horizontal and vertical scalability are easier to achieve. This enables the customer to optimize his infrastructure and its related costs.

This document describes release 1.3 of ProfitBricks API. Its objective is to give developers a complete overview of its functionality and provide a valid reference during the development process.

HOW TO ACCESS THE API

You can access the API here:

Web Service Endpoint:

https://api.profitbricks.com/1.3

WSDL Location:

https://api.profitbricks.com/1.3/wsdl

VERSION HISTORY

The following list gives an overview about recent updates so that you get a summary which changes applied to the last update.

Date of Change	Affected Method	Description
2014-06-02	new v1.3 is included	API version 1.3 document is included
2014-06-11	- CreateDatacenter - CreateReservePublicIP - GetImage - GetAllImages - UpdateImage - UpdateSnapshot - GetAllSnapshot - GetSnapshot	A request wrapper is included for createDatacenter and createReservePublicIp requests. All image and snapshot request are included with missing hotplug features.
2014-06-24	- GetAllDatacenter - CreateServer - UpdateServer - GetServer - GetAllServer - GetSnapshot - GetAllSnapshot	Provisioning state is included for each network in getAllDatacenter request All server request and response are included with missing hotplug features
2014-07-01	- GetServer - GetAllServer - GetFirewall - GetAllFirewalls - GetNic - GetAllNics - AddFirewallRulesToNic	Added firewall rule name as part of firewall rule.
2014-07-08	- UpdateImage	Added name for update image as mandatory
2014-11-25	- several commands	Added new location us/lasdev for developer environment
2015-03-04	Additional Information added (User Management)	In ProfitBricks DCD the feature "User Management" was introduced. SOAP API usage depends on settings made in DCD.

FUNDAMENTALS

To better understand the intended usage of ProfitBricks API, a series of concepts used throughout this document need to be introduced.

Virtual Data Center

A set of virtual components (servers, storage devices, load balancers, firewalls, etc) and the relations and connections configured between them. Its layout and composition can be changed through the DCD and through the API this document describes.

Virtual Server

An individual virtual machine whose configurable parameters are the like of: number of cores, RAM memory, etc. At the time of writing, activating changes in the parameters of a running server requires rebooting. In the next future such constraint will be removed.

Virtual Storage

A block device that can be connected to a server offering basic data storage capabilities.

Image

An image is a single file, containing the complete contents and structure of a data medium or device, which can be used to initialize a storage device. The user can either choose from ProfitBricks default images or upload individual images via FTP and assign them to CD-ROM/DVD drives or virtual storage devices.

Public LAN

A local area (sub)network which has access to the internet. A random DHCP IP address is being assigned to the NIC of a server in a LAN automatically by ProfitBricks. The DHCP IP address will change eventually by some network operations, like rebooting a server or disconnecting and reconnecting the LAN to the internet. Though, the user can reserve fixed public IP addresses and assign them manually to a NIC. Several public IP addresses can only be reserved block-wise.

Private LAN

A local area (sub)network which has no access to the internet. Private IPs can be specified and assigned manually. Valid IP addresses for private (sub)networks are 10.0.0.0/8, 172.16.0.0/12 or 192.168.0.0/16

Virtual Resources State

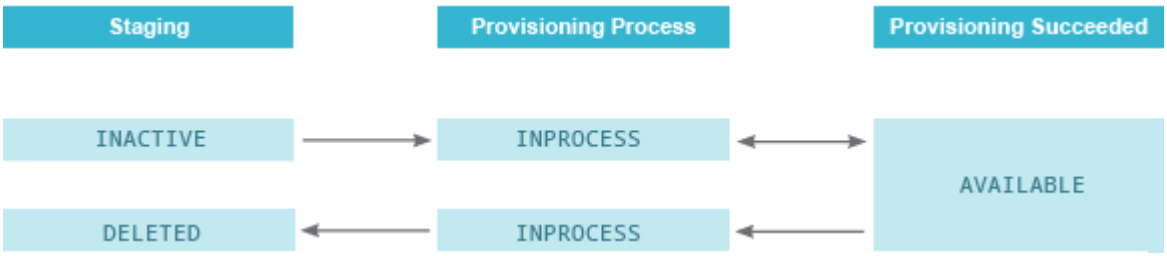
A Virtual Resource (Virtual Server, Virtual Storage, Virtual Data Center) will be, at any given moment, in a certain state.

The API is used by clients to send requests, as described in the API methods, to create or manage a Virtual Data Center. As soon as a request has been send, a new virtual resource with a unique identifier is being created and therewith starts it's virtual resource life cycle.

Right after the system received a request, a identifier of the new virtual resource is being responded back to the client. This identifier can be used by the client to get all desired information about the respective resource, including its provisioning state during its life cycle, until the virtual resource has been deleted.

Life Cycle of a Virtual Resource (Provisioning States)

This diagram describes the virtual resources states (life cycle) during a provisioning process.



General Provisioning States (for data center, server and storage)

INACTIVE	User request for new resource has been received and scheduled
INPROCESS	Provisioning in process. In case the user is sending new requests while provisioning is still in progress, the requests will be scheduled by the system to be processes at the next provisioning
AVAILABLE	All requests completed successfully. Data center or devices is/are ready for use
DELETED	Data center or objects has/have been deleted

Specific Server States

NOSTATE	The Virtual Server has no state (either if provisioning is still in process or if provisioning was successful, but the Virtual Server failed to boot from the selected device)
RUNNING	The Virtual Server is running
BLOCKED	The Virtual Server blocked and not running. This is a common idle state e.g. if the server is waiting for I/O

PAUSE	The Virtual Server has been paused. While in paused state, the Virtual Server will still consume allocated resources like memory but will not be eligible for scheduling
SHUTDOWN	The Virtual Server is in the process of getting shut down properly by the OS
SHUTOFF	The Virtual Server is shut off
CRASHED	The Virtual Server has been crashed

FTP Settings for image-upload

ProfitBricks provides a set of images that are optimized to work in the ProfitBricks environment. Of course, you can also use your own images (HDD as well as ISO). These images have to be uploaded to the ProfitBricks FTP server. Please note that there is a dedicated FTP server per data center. In case you want to use an image in different data center you have to upload the image to each data center FTP individually. Please select the settings as described below, to connect to our FTP server:

Data Center Germany - Frankfurt am Main

Server	ftp-fra.profitbricks.com
Port	21 (default)
Login	*Your ProfitBricks Login*
Password	*Your ProfitBricks Password*

Data Center Germany - Karlsruhe

Server	ftp-fkb.profitbricks.com
Port	21 (default)
Login	*Your ProfitBricks Login*
Password	*Your ProfitBricks Password*

Data Center USA - Las Vegas

Server	ftp-las.profitbricks.com
Port	21 (default)
Login	*Your ProfitBricks Login*
Password	*Your ProfitBricks Password*

DevOps Data Center USA - Las Vegas

Server	ftp-lasdev.profitbricks.com
Port	21 (default)
Login	*Your ProfitBricks Login*
Password	*Your ProfitBricks Password*

Once you have logged in, you will find the two folders “hdd-images” and “iso-images” in the start directory.

Upload your image(s) to the folder corresponding to your image type.

Note:

The following file types are allowed:

- ISO 9660 CD-ROM
- VMware Disk Image
- QEMU QCOW Image
- Microsoft Disk Image
- VirtualBox Disk Image
- UDF file system
- RAW Disk Image
- Parallels Disk Image

The following characters are allowed for file names: a-z A-Z 0-9 - . / _ ( ) # ~ + = blanks

RESOURCE LIMITS

All accounts have default limits which are intended to manage capacity and prevent abuse.

There are two kind of system wide resource limits:

- **Default Resource Limit per Device**, which is fixed and cannot be exceeded.
- **Default Resource Limit per Account**, which is set. When necessary, this limit can be increased on customer's demand by the ProfitBricks Support Team.

Systemwide Default Resource Limits per Device

Resources	Default Limit
Cores per Virtual Server	62
NICs per Virtual Server	6
CD-ROM/DVD-Drive per Virtual Server	2
Number of Virtual Storages per VirtualServer	8
Memory per Virtual Server	240 GB
Storage Space per Virtual Storage	8 TB

Systemwide Default Resource Limits per Account

Resources	Default Limit
Cores per Account	62
Memory per Account	320 GB
Storage Space per Account	8 TB

USER MANAGEMENT

ProfitBricks released a user management feature in the Data Center Designer (DCD) - the graphical user interface of ProfitBricks. This feature is accessible to users that initially created a contract with ProfitBricks. These contract owners are allowed to add further users to their contract and grant privileges (e.g. create new data centers, create snapshots or order public IPs). Also, contract owners can add users to existing resources and grant permission to edit these resources.

Of course, given privileges and permissions will be applicable through the ProfitBricks SOAP API. For instance:

- If a user has no access to a resource he will not be able to retrieve information about this resource through any call.
- If a user has READ permission to a resource he can retrieve information but cannot update or delete the resource.
- If a user has WRITE permission to a resource he can retrieve information as well as update and delete the resource.

User Management considers virtual data center, snapshots, IP blocks as well as private images for assignment to other users. All methods for specific virtual data center elements (e.g. Server Operations, Storage Operations, Load Balancer Operations, Firewall Operations, etc.) inherit the privilege and permission from the virtual data center the element belongs to. If a resource is not accessible or required operations permission are not given the application returns an appropriate error code (403 - Access denied).

For further details about User Management please visit the DCD Online Help.

API OUTLINE

AUTHENTICATION

ProfitBricks API is a SOAP web service offered to its users over a TLS transport combined with HTTP Basic authentication. The customer will need to configure their SOAP client with user credentials (username, password) obtained through the ProfitBricks website.

Authentication requests and data center operations are performed using SSL over HTTP (HTTPS) for data confidentiality and integrity.

SOAP REQUEST

Any requests conforms to a XML schema, which is defined as part of ProfitBricks WSDL, whereby for each function optional and mandatory parameters are well defined.

Example: SOAP Request

```
// SOAP REQUEST TO CREATE A DATA CENTER
 <soapenv:Envelope
 xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:ws="http://ws.api.profitbricks.com/">
    <soapenv:Header>
    </soapenv:Header>
    <soapenv:Body>
       <ws:createDataCenter>
          <request>
               <dataCenterName>abc</dataCenterName>
               <location>de/fkb</location>
          </request>
       </ws:createDataCenter>
    </soapenv:Body>
 </soapenv:Envelope>
```

SOAP RESPONSE

Any requests conforms to a XML schema, which is defined as part of ProfitBricks WSDL, whereby for each function optional and mandatory parameters are well defined.

Common response fields for API calls:

Parameter	Description	Included in
requestId	A unique string assigned to each request by the ProfitBricks API. All requests are recorded. Therefore, the ProfitBricks Support Team is able to identify all issues by its ID, in case the user requires support for troubleshooting	All requests
dataCenterId	Identifier of the data center the request operated on	All provisioning requests
dataCenterVersion	Data center Version, resulting by the operation, increment by 1	All provisioning requests.
location	Data center location, eg: de/fkb for karlsruhe	All provisioning requests.

Example: SOAP Response

```
// SOAP Request to create a data center
 <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <S:Header>
   </S:Header>
   <S:Body>
      <ns2:createDataCenterResponse
xmlns:ns2="http://ws.api.profitbricks.com/">
         <return>
            <requestId>1163</requestId>
            <dataCenterId>4b98319a-125e-4d1a</dataCenterId>
            <dataCenterVersion>1</dataCenterVersion>
         </return>
      </ns2:createDataCenterResponse>
   </S:Body>
</S:Envelope>
```

SOAP FAULT & ERROR CODES OVERVIEW

After receiving a request, the system will check for client related errors. If an error occurs, a SOAP fault message will be returned. The following error information can be found in SOAP fault messages:

- faultCode and httpCode, organized in different error scenarios groups
- full error messages, readable by the front end user

Example: SOAP Fault Response

```
// SOAP Fault Response of a data center
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
      <S:Header />
      <S:Body>
            <S:Fault xmlns:ns4="http://www.w3.org/2003/05/soap-envelope">
                  <faultcode>S:Server</faultcode>
                  <faultstring>RAM of requested server too small, 256 Mb is
minimum</faultstring>
                  <detail>
                        <ns2:ProfitbricksServiceFault
                              xmlns:ns2="http://ws.api.profitbricks.com/">
                              <faultCode>BAD_REQUEST</faultCode>
                              <httpCode>400</httpCode>
                              <message>RAM of requested server too small, 256
Mb is minimum</message>
                              <requestId>1045</requestId>
                        </ns2:ProfitbricksServiceFault>
                  </detail>
            </S:Fault>
      </S:Body>
</S:Envelope>
```

ERROR CODES OVERVIEW

Error Codes

HTTP Code	Code	Expected In
400	BAD_REQUEST	All
401	UNAUTHORIZED	All
403	CURRENT_USER_NOT_PERMITTED	All
404	RESOURCE_NOT_FOUND	All
409	PROVISIONING_IN_PROCESS	Deleting a data center
409	PROVISIONING_NO_CHANGES	Updating existing resources
410	RESOURCE_DELETED	Get functions
413	OVER_LIMIT_SETTING	Request new resources
503	SERVER_EXCEEDED_CAPACITY	Request new resources
503	UNEXPECTED	All

Details and Examples:

400 BAD_REQUEST

Example: Invalid name parameters, missing mandatory parameters, etc.

401 UNAUTHORIZED

The user does not have access permission to any resources in this data center.

403 CURRENT_USER_NOT_PERMITTED

Access denied as the user does not have the permission to execute the operation due to User Management configurations.

404 RESOURCE_NOT_FOUND

The request resource does not exist or has been deleted by the user.

409 PROVISIONING_NO_CHANGES

The request does not apply any changes in the active data center.
Example: connect a storage to a server, which is already connected.

413 OVER_LIMIT_SETTING

Request exceeds the resource/account limit. See also “4. Resource Limits”.

503 SERVER_EXCEED_CAPACITY

The request cannot be performed, because the request exceeds its service's capacity. Example: the system cannot reserve a big block of public IP addresses.

503 UNEXPECTED

Users should contact ProfitBricks Support Team (requestId is required).

API METHODS

The following section describes all operations currently available through ProfitBricks API. Operations for creating, updating and deleting resources are considered provisioning events. Errors happening while processing those events are appropriately signaled to the client application through the usage of SOAP Faults.

USER NOTIFICATION OPERATIONS

Used to access notifications the system sends to signal specific events.

These event often require user interaction, so the current notifications should be checked frequently.

Index

Get Notifications

Delete Notifications

GET NOTIFICATIONS

Gets all current (not deleted) user notifications.

User notifications are created by the system to inform users about different events concerning e.g. their data centers.

Notifications stay in the queue until deleted (s ee deleteNotifications), so a notification may be read repeatedly with

getNotifications.

```
getNotifications(): List<Notification>
```

Request parameters

Name	Description	Required
networkUUID	Uuid of the network the notification refers to	yes

Response parameters

Name	Description
id	Identifier of the this notification

networkUUid	Uuid of the network the notification refers to
timestamp	Creation time of this notification
messageCode	One of 'MessageCodes', see below
relatedItemType	One of 'ItemTypes', see below
relatedItemUuid	Id of specific component of type 'relatedItemType' that the message concerns
message	Textual message

Message Codes

Name	Description
SHUTDOWN_SERVER_FAILED	Shutdown of a server failed. User interaction required to ensure proper system state. See relatedItemUuid for the server id.
OTHER	Non-typed notification

Related Item Type

Name	Description
TYPE_SERVER	Message concerns a server
TYPE_STORAGE	Message concerns a storage
TYPE_IMAGE	Message concerns an image
TYPE_LOADBALANCER	Message concerns a load balancer
TYPE_NIC	Message concerns a network interface card
TYPE_FIREWALL	Message concerns a firewall
TYPE_OTHER	Message concerns an other component or unspecified

Error Codes

None

DELETE NOTIFICATIONS

Deletes the notifications with the given ids.

```
deleteNotifications (List<String> notificationId): void
```

Request parameters

Name	Description	Required
notificationId	Id of a notification to delete	yes

Response parameters

None

Error Codes

Code	Description
RESSOURCE_NOT_FOUND	A notification with one of the given ids was not found / was deleted already.

VIRTUAL DATA CENTER OPERATIONS

Index:

Create Data Center

Get Data Center by ID

Get Data Center State

Get All Data Centers

Update Data Center

Clear Data Center

Delete Data Center

CREATE DATA CENTER

Creates and saves a new, empty Virtual Data Center. Returns its identifier for further reference.

```
createDataCenter (request:createDataCenterRequest):
createDataCenterResponse
```

Request parameters

Name	Description	Required
dataCenterName	Names the new virtual data center. If no name is specified, the data center will be named “Unnamed Data Center” by default.	yes
location	Select location to create the data center, choose one from the below list 1. de/fkb -> (refers to Karlsruhe) 2. de/fra -> (refers to Frankfurt) 3. us/las -> (refers to Las Vegas) 4. us/lasdev -> (refer to DevOps Data Center in Las Vegas)	yes

Naming Restrictions

Data center names cannot start with or contain (@, /, \, |, ‘’, ‘)

Important Information about a Data Centers Location

You need to explicitly set the location while creating a data center.

A data center's location cannot be changed anymore. Also, be aware that data centers from different locations are isolated from each other. It is not possible to move a data center or elements of a data center from one location to another.

Response parameters

Name	Description
dataCenterId	Identifier of the virtual data center
location	Location where the data center is created (us/las, de/fkb, de/fra)

Error Codes

Code	Description
BAD_REQUEST	Invalid characters used in the virtual data center name

GET DATA CENTER BY ID

Returns information about an existing virtual data center’s state and configuration.

```
getDataCenter(dataCenterId : String): DataCenter
```

Request parameters

Name	Description	Required
dataCenterId	Identifier of the virtual data center	Yes

Response parameters

Name	Description	Constraints
dataCenterName	Name of the new virtual data center	
provisioningState	Describes the current state of the specified data center (INACTIVE, INPROCESS, AVAILABLE, DELETED,ERROR)	
servers	A list of all virtual servers. See also Get Server	Emptiable
storages	A list of all virtual storages. See also Get Storage	Emptiable
location	Location where the data center is been created (de/fkb, de/fra, us/las, us/lasdev)	
loadBalancers	A list of all virtual load balancers. See also Get Load Balancer	Emptiable

Error Codes

Code	Description
RESOURCE_NOT_FOUND	Data Center does not exist
UNAUTHORIZED	User is not authorized to access the data center

GET DATA CENTER STATE

This is a lightweight function for pooling the current provisioning state of the Virtual Data Center. It is recommended to use this function for large Virtual Data Centers to query request results.

```
getDataCenterState(dataCenterId : String): ProvisioningState
```

Response parameters

Name	Description
provisioningState	Provisioning State of the target Virtual Data Center (INACTIVE, INPROCESS, AVAILABLE, DELETED, ERROR)

Error Codes

Code	Description
RESOURCE_NOT_FOUND	Data Center does not exist
UNAUTHORIZED	User is not authorized to access the data center

GET ALL DATA CENTERS

Returns a list of all Virtual Data Centers created by the user, including ID, name and version number.

```
getAllDataCenters(): List<DataCenterIdentifier>
```

Response parameters

Name	Description
dataCenterId	Identifier of the virtual data center
dataCenterName	Name of the virtual data center
dataCenterVersion	Version of the virtual data center
provisioningState	Provisioning State of the target Virtual Data Center (INACTIVE, INPROCESS, AVAILABLE, DELETED, ERROR)

UPDATE DATA CENTER

Updates the information associated to an existing Virtual Data Center.

```
updateDataCenter(request : UpdateDcRequest)
```

Request parameters

Name	Description	Required
dataCenterId	Identifier of the virtual data center	Yes
dataCenterName	Renames the target virtual data center	

Error Codes

Code	Description
BAD_REQUEST	Invalid characters used in the virtual data center name

RESOURCE_NOT_FOUND	Invalid data center identifier / Specified data center ID does not exist
UNAUTHORIZED	User is not authorized to access the data center

Note

It is only possible to update the Virtual Data Center name at the moment.

CLEAR DATA CENTER

Removes all components from an existing Virtual Data Center.

```
clearDataCenter(dataCenterId : String)
```

Request parameters

Name	Description	Required
dataCenterId	Identifier of the virtual data center	Yes

Error Codes

Code	Description
UNAUTHORIZED	User is not authorized to access the data center
RESOURCE_NOT_FOUND	Invalid data center identifier / Specified data center ID does not exist

DELETE DATA CENTER

Deletes an Virtual Data Center. If a previous request on the target data center is still in progress, the data center is going to be deleted after this request has been completed. Once a Data Center has been deleted, no further request can be performed on it.

```
deleteDataCenter(dataCenterId : String)
```

Request parameters

Name	Description	Required
dataCenterId	Identifier of the virtual data center	Yes

Error Codes

Code	Description
RESOURCE_NOT_FOUND	Invalid data center identifier / Specified data center ID does not exist
UNAUTHORIZED	User is not authorized to access the data center

System messages returning on request during a deletion process

Code	Description
PROVISIONING_IN_PROCESS	Data Center is being deleted, process not finished yet

System messages returning on request after a deletion process has been finished

Code	Description
RESOURCE_DELETED	Data Center has been deleted

Note

The command deleteDataCenter will delete the data center. Be aware that there will be no roll back option, so be please use this command deliberately.

SERVER OPERATIONS

Index:

Create Server

Get Server

Get All Servers

Reset Server

Start Server

Stop Server

Update Server

Delete Server

CREATE SERVER

Creates a Virtual Server within an existing data center. Parameters can be specified to set up a boot device and connect the server to an existing LAN or the Internet.

```
createServer(request : CreateServerRequest): CreateServerResponse
```

The parameters lanId and internetAccess are suitable for fast server start-ups. For example, a new server with internet access can be started quickly by simply sending one request with three parameters cores, ram and internetAccess. See also **Create NIC**

Request parameters

Name	Description	Required
`cores`	Number of cores to be assigned to the specified server	Yes
`ram`	Number of RAM memory (in MiB) to be assigned to the server. The size is specified in multiples of 256 MiB with a minimum of 256 MiB Exception, minimum 1024 MiB is required, if ramHotPlug capability is enabled	Yes
`dataCenterId`	Defines the data center wherein the server is to be created. If left empty, the server will be created in a new data center	
`serverName`	Name of the server to be created	
`bootFromImageId`	Defines an existing CD-ROM/DVD image ID to be set as boot device of the server. A virtual CD-ROM/DVD drive with the mounted image will be connected to the server. See also **Add CD-ROM/DVD Drive To Server**	
`bootFromStorageId`	Defines an existing storage device ID to be set as boot device of the server. The storage will be connected to the server implicitly. See also **Create Storage** If not left empty, the bus type to which the storage will be connected is VIRTIO. For a different bus type a subsequent invocation of `connectStorageToServer` operation is required, with `busType` as a parameter.	
`lanId`	Connects the server to the specified LAN ID > 0. If the respective LAN does not exist, it is going to be created	
`internetAccess`	Set to TRUE to connect the server to the Internet via the specified LAN ID. If the LAN is not specified, it is going to be created in the next available LAN ID, starting with LAN ID 1	
`availabilityZone`	Selects the zone in which the server is going to be created (AUTO, ZONE_1, ZONE_2). Servers from different zones are located in different physical locations. If set to AUTO or	

	left empty, servers will be created in a random zone	
osType	Sets the OS type of the server. (WINDOWS, LINUX, OTHER, UNKNOWN) If left empty, the server will inherit the OS Type of its selected boot image / storage	
cpuHotPlug	Set the server CPU Hot-Plug capability (TRUE/FALSE)	
ramHotPlug	Set the server RAM Hot-Plug capability (TRUE/FALSE)	
nicHotPlug	Set the server NIC Hot-Plug capability (TRUE/FALSE)	
nicHotUnPlug	Set the server NIC Hot-UnPlug capability (TRUE/FALSE)	
discVirtioHotPlug	Set the server capabilities to hotplug storages which are connected through VirtIO bustypeSet (TRUE/FALSE)	
discVirtioHotUnPlug	Set the server capabilities to hotUnplug storages which are connected through VirtIO bustypeSet (TRUE/FALSE)	

The OS Type of a server must be set to WINDOWS when using a windows image, due to Microsoft's terms and conditions. The default OS Type of a server is UNKNOWN, when it is booted from an image uploaded by the customer.

internetAccess

If a new server is created in a private LAN, it will always be a private server and cannot be connected to the Internet by setting internetAccess to TRUE.

Memory HotPlug Capability for Windows

Memory HotPlug is not supported for virtual machines booting from a Windows image. When setting the option to TRUE it will return an error (bad_request).

Live Vertical Scaling Capabilities

You can only set any Live Vertical Scaling capability (e.g. cpuHotPlug) when the server gets connected to a storage with a bootable image. If the server is not connected to a storage with a bootable image or booting from an ISO directly the parameter settings get ignored.

Response parameters

Code	Description
serverId	Identifier of the virtual server

Cores/RAM Minimum

The amount of cores must be >= 1. The RAM size has a minimum enforced value of 256 MiB and all RAM sizes must be a multiple of such value (512, 768, etc). For ram hotplug enabled VM the minimum value for RAM is 1024 MiB

Error Codes

Name	Description
BAD_REQUEST	Invalid characters in the virtual server name Wrong boot image type (HDD image instead of CD-ROM/DVD) Too many boot devices (must be either storage or CD-ROM/DVD image) Invalid RAM and/or cores (cores < 1, RAM < 256 MiB, RAM size not a multiple of 256)
OVER_LIMIT_SETTING	Cores and/or RAM limit exceeded. See also Resource Limits for more details
RESOURCE_NOT_FOUND	Boot image/storage does not exist
UNAUTHORIZED	User is not authorized to access the server

GET SERVER

Returns information about a virtual server, such as configuration, provisioning status, power status, etc.

```
getServer(serverId : String): Server
```

Request parameters

Name	Description	Required
serverId	Identifier of the virtual server	Yes

Response parameters

Name	Description	Constraints
serverId	Identifier of the virtual server	
serverName	Outputs the name of the specified virtual server	Emptiable
creationTime	Time when the specified virtual server has been created	Nullable

lastModificationTime	Time when the specified virtual server has last been modified	Nullable
provisioningState	Describes the current provisioning state of the specified virtual server (INACTIVE, INPROCESS, AVAILABLE, DELETED, ERROR)	
virtualMachineState	Describes the current server state of the specified virtual server (NOSTATE, RUNNING, BLOCKED, PAUSED, SHUTDOWN, SHUTOFF, CRASHED)	
cores	Amount of cores assigned to the specified virtual server	
ram	RAM (in MiB) assigned to the specified virtual server	
internetAccess	Returns TRUE if server is connected to a public LAN	Emptiable
ips	Lists all IP addresses assigned to the server	
nics	Lists all NICs assigned to the server. See also **Get NIC**	Emptiable
connectedStorages	Lists all storages connected to the server, including the following paramenters: - storageId: storage identifier - storageName: name of the storage - size: storage size (in GiB) For these three parameter see also **Create Storage** or **Get Storage** - busType: Bus type to which the storage is connected - deviceNumber: device number the storage is connected For these two parameter see also **Connect Storage** - bootDevice: a flag, indicating whether server is booting from this storage or not [TRUE/FALSE] For this parameter see also **Create Server**	Emptiable
availabilityZone	Zone in which the server is located (AUTO, ZONE_1, ZONE_2)	
romDrives	Lists all CD-ROM drives assigned to the server, including the parameters of the CD-ROM/DVDs image, such as: - imageId: CD-ROM/DVD image identifier - imageName: name of the CD-ROM/DVD image - bootDevice: a flag, indicating whether server is booting from this drive or not [TRUE/FALSE] See also **Add CD-ROM/DVD Drive To Server**	Emptiable
osType	OS type of the server (WINDOWS, LINUX, OTHER, UNKOWN)	Emptiable
cpuHotPlug	Returns TRUE if server contains capabilities to hotplug CPU	Emptiable

ramHotPlug	Returns TRUE if server contains capabilities to hotplug RAM	Emptiable
nicHotPlug	Returns TRUE if server contains capabilities to hotplug NIC	Emptiable
nicHotUnPlug	Returns TRUE if server contains capabilities to hotUnplug NIC	Emptiable
discVirtioHotPlug	Returns TRUE if server contains capabilities to hotplug storage with bustype VIRTIO	Emptiable
discVirtioHotUnPlug	Returns TRUE if server contains capabilities to hotUnplug storage with bustype VIRTIO	Emptiable

NOSTATE

A Get Virtual Server command returns the state NOSTATE if the provisioning is still in process, or (in case the provisioning process was successful) if it's attempt to boot from the defined boot device has failed.

Why does the Get Server command output no IP or NIC parameter?

If the get command outputs no information about IP and NIC, then the provisioning process is probably not finished yet. The server has to be connected to the network and AVAILABLE to return these parameters.

Error Codes

Code	Description
RESOURCE_DELETED	The server has been deleted (by the user)
RESOURCE_NOT_FOUND	Specified server does not exist
UNAUTHORIZED	User is not authorized to access the server

GET ALL SERVERS

Returns information about all virtual server, such as configuration, provisioning status, power status, etc.

```
getAllServers(): List<Servers>
```

Response parameters

Name	Description	Constraints
serverId	Identifier of the virtual server	
serverName	Outputs the name of the specified virtual server	Emptiable

creationTime	Time when the specified virtual server has been created	Nullable
lastModificationTime	Time when the specified virtual server has last been modified	Nullable
provisioningState	Describes the current provisioning state of the specified virtual server (INACTIVE, INPROCESS, AVAILABLE, DELETED, ERROR)	
virtualMachineState	Describes the current server state of the specified virtual server (NOSTATE, RUNNING, BLOCKED, PAUSED, SHUTDOWN, SHUTOFF, CRASHED)	
cores	Amount of cores assigned to the specified virtual server	
ram	RAM (in MiB) assigned to the specified virtual server	
internetAccess	Returns TRUE if server is connected to a public LAN	Emptiable
ips	Lists all IP addresses assigned to the server	
nics	Lists all NICs assigned to the server. See also **Get NIC**	Emptiable
connectedStorages	Lists all storages connected to the server, including the following paramenters: - storageId: storage identifier - storageName: name of the storage - size: storage size (in GiB) For these three parameter see also **Create Storage** or **Get Storage** - busType: Bus type to which the storage is connected - deviceNumber: device number the storage is connected For these two parameter see also **Connect Storage** - bootDevice: a flag, indicating whether server is booting from this storage or not [TRUE/FALSE] For this parameter see also **Create Server**	Emptiable
availabilityZone	Zone in which the server is located (AUTO, ZONE_1, ZONE_2)	
romDrives	Lists all CD-ROM drives assigned to the server, including the parameters of the CD-ROM/DVDs image, such as: - imageId: CD-ROM/DVD image identifier - imageName: name of the CD-ROM/DVD image - bootDevice: a flag, indicating whether server is booting from this drive or not [TRUE/FALSE] See also **Add CD-ROM/DVD Drive To Server**	Emptiable
osType	OS type of the server (WINDOWS, LINUX, OTHER, UNKOWN)	
cpuHotPlug	Returns TRUE if server contains capabilities to hotplug	Emptiable

	CPU	
ramHotPlug	Returns TRUE if server contains capabilities to hotplug RAM	Emptiable
nicHotPlug	Returns TRUE if server contains capabilities to hotplug NIC	Emptiable
nicHotUnPlug	Returns TRUE if server contains capabilities to hotUnplug NIC	Emptiable
discVirtioHotPlug	Returns TRUE if server contains capabilities to hotplug storage with bustype VIRTIO	Emptiable
discVirtioHotUnPlug	Returns TRUE if server contains capabilities to hotUnplug storage with bustype VIRTIO	Emptiable

NOSTATE

A Get All Servers command returns the state NOSTATE if the provisioning is still in process, or (in case the provisioning process was successful) if it's attempt to boot from the defined boot device has failed.

Why does the Get Server command output no IP or NIC parameter?

If the get command outputs no information about IP and NIC, then the provisioning process is probably not finished yet. The server has to be connected to the network and AVAILABLE to return these parameters.

UPDATE SERVER

Updates parameters of an existing virtual server device.

```
updateServer(request : UpdateServerRequest)
```

Request parameters

Name	Description	Required
serverId	Identifier of the target virtual server	Yes
serverName	Renames the target virtual server	
cores	Updates the amount of cores of the target virtual server	
ram	Updates the RAM memory (in MiB) of the target virtual server. The minimum RAM size is 256 MiB	
bootFromImageId	Defines an existing CD-ROM/DVD (ISO) image ID to be set as boot device of the server. A virtual CD-ROM/DVD drive with the mounted image will be connected to the server implicitly. See also **Add CD-ROM/DVD Drive To Server**	

availabilityZone	Zone in which the server is located (AUTO, ZONE_1, ZONE_2)	
bootFromStorageId	Defines an existing storage device ID to be set as boot device of the server. The storage will be connected to the server implicitly. See also **Create Storage**	
osType	Updates OS Type of the target server (WINDOWS, LINUX, OTHER, UNKOWN)	
cpuHotPlug	Set the server CPU Hot-Plug capability (TRUE/FALSE)	
ramHotPlug	Set the server RAM Hot-Plug capability (TRUE/FALSE)	
nicHotPlug	Set the server NIC Hot-Plug capability (TRUE/FALSE)	
nicHotUnPlug	Set the server NIC Hot-UnPlug capability (TRUE/FALSE)	
discVirtioHotPlug	Set the server capabilities to hotPlug storages which are connected through VirtIO bustypeSet (TRUE/FALSE)	
discVirtioHotUnPlug	Set the server capabilities to hotUnplug storages which are connected through VirtIO bustypeSet (TRUE/FALSE)	

Memory HotPlug Capability for Windows

Memory HotPlug is not supported for virtual machines booting from a Windows image. When setting the option to TRUE it will return an error (bad_request).

Live Vertical Scaling Capabilities

You can only set any Live Vertical Scaling capability (e.g. cpuHotPlug) when the server gets connected to a storage with a bootable image. If the server is not connected to a storage with a bootable image or booting from an ISO directly the parameter settings get ignored.

Error Codes

Code	Description
BAD_REQUEST	Invalid characters used in the virtual server name Wrong boot image type (e.g. HDD image instead of CD-ROM/DVD or vice versa) Too many boot images (must be either a HDD OR a CD-ROM/DVD image) Invalid RAM and/or cores
OVER_LIMIT_SETTING	Number of cores and/or RAM limit exceeded. See also **Resource Limits** for more details
RESOURCE_NOT_FOUND	Specified Server, boot image and/or storage does not exist
UNAUTHORIZED	User is not authorized to access the server

DELETE SERVER

Deletes an existing Virtual Server.

```
deleteServer(serverId : String)
```

Request parameters

Name	Description	Required
serverId	Identifier of the target virtual server	Yes

Error Codes

Code	Description
RESOURCE_NOT_FOUND	Specified server does not exist
UNAUTHORIZED	User is not authorized to access the server

RESET SERVER

Resets an existing virtual server (POWER CYCLE).

- Server will be forcefully powered off and restarted immediately. Any unsaved data may be lost!
- Billing will continue

> **Graceful REBOOT**
>
> A graceful reboot of a server is not possible through the ProfitBricks API. We recommend to access and execute the command on the virtual server directly.

```
resetServer(serverId : String)
```

Request parameters

Name	Description	Required
serverId	Identifier of the target virtual server	Yes

Error Codes

Code	Description
RESOURCE_NOT_FOUND	Specified server does not exist
UNAUTHORIZED	User is not authorized to access the server

START SERVER

Starts an existing virtual server

- Server may receive new public IP addresses if necessary
- Billing will continue

```
startServer(serverId : String)
```

Request parameters

Name	Description	Required
serverId	Identifier of the target virtual server	Yes

Error Codes

Code	Description
RESOURCE_NOT_FOUND	Specified server does not exist
UNAUTHORIZED	User is not authorized to access the server

STOP SERVER

Stops an existing virtual server forcefully (HARD stop)

- Server will be forcefully powered off. Any unsaved data may be lost!
- Billing for this server will be stopped
- When restarting the server a new public IP gets assigned, alternatively, you can reserve IP addresses, see reservation of public IP blocks

> **Graceful SHUTDOWN**
>
> A graceful stop of a server is not possible through the ProfitBricks API. We recommend to access and execute the command on the virtual server directly. Once the server was shutdown you still can use the "stopServer" method that will stop billing.

```
stopServer(serverId : String)
```

Request parameters

Name	Description	Required
serverId	Identifier of the target virtual server	Yes

Error Codes

Code	Description
RESOURCE_NOT_FOUND	Specified server does not exist
UNAUTHORIZED	User is not authorized to access the server

STORAGE OPERATIONS

Index:

Create Storage

Get Storage

Get All Storages

Connect Storage

Disconnect Storage

Update Storage

Delete Storage

CREATE STORAGE

Creates a virtual storage within an existing virtual data center. Additional parameters can be specified, e.g. for assigning a HDD image to the storage.

```
createStorage(request : CreateStorageRequest): CreateStorageResponse
```

Request parameters

Name	Description	Required
size	Storage size (in GiB)	Yes
dataCenterId	Defines the data center wherein the storage is	

	to be created. If left empty, the storage will be created in a new data center	
storageName	Name of the storage to be created	
mountImageId	Specifies the HDD image to be assigned to the storage by its ID	
profitBricksImagePassword	Sets the VM image root login password to the specified value. Only supported for generic Profitbricks HDD images. User images are expected to be preconfigured with a password. If no password is supplied, one is automatically created. Please see error codes for password syntax rules.	

Important

The image will be created as a partition in the storage device. The size of the virtual storage image must be smaller or equal the storage size. You can increase the size of the storage if necessary. See also **Update Storage**

Response parameters

Name	Description
storageId	Identifier of the virtual storage

Error Codes

Code	Description
BAD_REQUEST	Invalid characters in the virtual storage name Invalid storage size (must be > 1 GiB) Image and storage are not located in the same region Password is set for a non-Profibricks or Profitbricks-non-HDD image
OVER_LIMIT_SETTING	Storage size exceeds limit
RESOURCE_NOT_FOUND	Specified Image does not exist
UNAUTHORIZED	User is not authorized to access the storage
PASSWORD_SYNTAX_ERROR	Given password does not abide to password syntax rules. Please see error message for more details. As of writing, password must be between 8 and 50 characters, only a-z, A-Z, 0-9 without characters i, I, l, o, O, w, W, y, Y, z, Z and 1, 0

GET STORAGE

Returns information about a virtual storage's configuration and provisioning state.

```
getStorage(storageId : String): Storage
```

Request parameters

Name	Description	Required
storageId	Identifier of the virtual storage	Yes

Response parameters

Name	Description	Constraints
storageId	Identifier of the virtual storage	
storageName	Outputs the name of the specified virtual storage	Emptiable
creationTime	Time when the specified virtual storage has been created	Nullable
lastModificationTime	Time when the specified virtual storage has last been modified	Nullable
provisioningState	Current provisioning state of the specified virtual storage (INACTIVE, INPROCESS, AVAILABLE, DELETED, ERROR)	
size	Size of the specified virtual storage (in GiB)	
serverIds	Lists the server connected to the storage by ID. See also Get Server	Emptiable
mountImage	Outputs information about the image assigned to the storage, such as image ID and image name. See also Get Image	Nullable

Error Codes

Code	Description
RESOURCE_DELETED	The storage has been deleted (by the user)
RESOURCE_NOT_FOUND	Specified storage does not exist
UNAUTHORIZED	User is not authorized to access the storage

GET ALL STORAGES

Returns information about all virtual storage, such as configuration and provisioning state.

```
getAllStorages(): List<Storages>
```

Response parameters

Name	Description	Constraints
storageId	Identifier of the virtual storage	

storageName	Outputs the name of the specified virtual storage	Emptiable
creationTime	Time when the specified virtual storage has been created	Nullable
lastModificationTime	Time when the specified virtual storage has last been modified	Nullable
provisioningState	Current provisioning state of the specified virtual storage (INACTIVE, INPROCESS, AVAILABLE, DELETED, ERROR)	
size	Size of the specified virtual storage (in GiB)	
serverId	Lists the server connected to the storage by ID. See also Get Server	Emptiable
mountImage	Outputs information about the image assigned to the storage, such as image ID and image name. See also Get Image	Nullable

CONNECT STORAGE

Connects a virtual storage device to an existing server.

```
connectStorageToServer(request : ConnectStorageRequest)
```

Request parameters

Name	Description	Required
storageId	Identifier of the virtual storage to be connected	Yes
serverId	Identifier of the target virtual server	Yes
busType	Bus type to which the storage will be connected Default Type is VIRTIO Type can be IDE or VIRTIO (BusType SCSI will be supported in the future)	
deviceNumber	Defines the device number of the virtual storage. If no device number is set, a device number will be automatically assigned	

Error Codes

Code	Description
BAD_REQUEST	Specified storage and server are not located within the same data center
RESOURCE_NOT_FOUND	Specified server/storage does not exist
OVER_LIMIT_SETTING	Storage connections limit exceeded (max. 8 per virtual server)
UNAUTHORIZED	User is not authorized to access the server/storage

DISCONNECT STORAGE

Disconnects a virtual storage device from a connected server.

```
disconnectStorageFromServer(storageId : String, serverId : String)
```

Request parameters

Name	Description	Required
storageId	Identifier of the connected virtual storage	Yes
serverId	Identifier of the connected virtual server	Yes

Error Codes

Code	Description
PROVISIONING_NO_CHANGES	Storage is not connected to the specified server
RESOURCE_NOT_FOUND	Specified server/storage does not exist
UNAUTHORIZED	User is not authorized to access the server/storage

UPDATE STORAGE

Updates parameters of an existing virtual storage device. It is possible to increase the storage size without reboot of an already provisioned storage. The additional capacity is not added to any partition. You have to partition the storage afterwards. Vice versa, it is not possible to decrease the storage size of an already provisioned storage.

```
updateStorage(request : UpdateStorageRequest)
```

Request parameters

Name	Description	Required
storageId	Identifier of the target virtual storage	Yes
storageName	Renames the target virtual storage	
size	Updates the size of the storage (in GiB)	
mountImageId	Specifies the HDD image to be assigned to the storage by its ID	

Error Codes

Code	Description
BAD_REQUEST	Invalid characters in the virtual storages name Invalid storage size (must be > 1 GiB)
OVER_LIMIT_SETTING	Storage size exceeds limit
RESOURCE_NOT_FOUND	Specified storage does not exist
UNAUTHORIZED	User is not authorized to access the storage

DELETE STORAGE

Deletes an existing virtual storage device.

```
deleteStorage(storageId : String)
```

Request parameters

Name	Description	Required
storageId	Identifier of the target virtual storage	Yes

Error Codes

Code	Description
RESOURCE_NOT_FOUND	Specified storage does not exist
UNAUTHORIZED	User is not authorized to access the storage

SNAPSHOT OPERATIONS

Index:

Create Snapshot

Get Snapshot

Get All Snapshots

Update Snapshot

Delete Snapshot

Rollback Snapshot

CREATE SNAPSHOT

Creates a snapshot of an existing storage device. The size of the snapshot will be the same as the size of the storage it was taken from independent of how much of it is in use. Any snapshot will be charged to your account and billed like an HD storage of the same size.

Attention: A snapshot only includes the contents of the storage but not the state of the VM connected to it.

```
createSnapshot(request : CreateSnapshotRequest): CreateSnapshotResponse
```

Request parameters

Name	Description	Required
`storageId`	Identifier of the virtual storage for which a snapshot shall be created	Yes
`snapshotName`	Name of the snapshot to be created	
`description`	Additional field to provide customized information about the data in this snapshot	

Response parameters

Name	Description
`snapshotId`	Identifier of the snapshot

Error Codes

Code	Description
`RESOURCE_NOT_FOUND`	Specified StorageID does not exist
`UNAUTHORIZED`	User is not authorized to access the storage

GET SNAPSHOT

Returns information about a particular Snapshot

```
getSnapshot(snapshotId : String): Snapshot
```

Request parameters

Name	Description	Required
snapshotId	Identifier of the snapshot	Yes

Response parameters

Name	Description
snapshotId	Identifier of the snapshot
description	text data that can provide additional data about a snapshot
snapshotName	name of the snapshot
snapshotSize	size of the snapshot
provisioningState	Current provisioning state of the snapshot (INACTIVE, INPROCESS, AVAILABLE, DELETED, ERROR)
bootable	flag to identify boot capabilities of a snapshot; flag of type boolean
osType	flag of currently set OS type of the snapshot (WINDOWS, LINUX, OTHER, UNKNOWN)
cpuHotPlug	snapshot contains capabilities to hotplug CPU; flag of type boolean
cpuHotUnplug	snapshot contains capabilities to hotUnplug CPU; flag of type boolean
ramHotPlug	snapshot contains capabilities to hotplug RAM; flag of type boolean
ramHotUnPlug	snapshot contains capabilities to hotUnplug RAM; flag of type boolean
nicHotPlug	snapshot contains capabilities to hotplug NIC; flag of type boolean
nicHotUnPlug	snapshot contains capabilities to hotunplug NIC; flag of type boolean
discVirtioHotPlug	snapshot contains capabilities to hotplug storages which are connected through VirtIO bustype; flag of type boolean
discVirtioHotUnPlug	snapshot contains capabilities to hotUnplug storages which are connected through VirtIO bustype; flag of type boolean
creationTimestamp	Time when the specified snapshot has been created
modificationTimestamp	Time when the specified snapshot has last been modified
location	Path of location in which the snapshot is available; it cannot be used in any other location than that (eg: de/fkb for karlsruhe)

Error Codes

Code	Description
RESOURCE_NOT_FOUND	Specified snapshot does not exist
UNAUTHORIZED	User is not authorized to access the snapshot

GET ALL SNAPSHOTS

Provides a list of all snapshots available to this account

```
getAllSnapshots(): List<Snapshots>
```

Response parameters

Name	Description
snapshotId	Identifier of the snapshot
description	text data that can provide additional data about a snapshot
snapshotName	name of the snapshot
snapshotSize	size of the snapshot
provisioningState	Current provisioning state of the snapshot (INACTIVE, INPROCESS, AVAILABLE, DELETED, ERROR)
bootable	flag to identify boot capabilities of a snapshot; flag of type boolean
osType	flag of currently set OS type of the snapshot (WINDOWS, LINUX, OTHER, UNKNOWN)
cpuHotPlug	snapshot contains capabilities to hotplug CPU; flag of type boolean
cpuHotUnplug	snapshot contains capabilities to hotunplug CPU; flag of type boolean
ramHotPlug	snapshot contains capabilities to hotplug RAM; flag of type boolean
ramHotUnplug	snapshot contains capabilities to hotUnplug RAM; flag of type boolean
nicHotPlug	snapshot contains capabilities to hotplug NIC; flag of type boolean
nicHotUnPlug	snapshot contains capabilities to hotunplug NIC; flag of type boolean
discVirtioHotPlug	snapshot contains capabilities to hotplug storages which are connected through VirtIO bustype; flag of type boolean
discVirtioHotUnPlug	snapshot contains capabilities to hotUnplug storages which are connected through VirtIO bustype; flag of type boolean
creationTimestamp	Time when the specified snapshot has been created
modificationTimestamp	Time when the specified snapshot has last been modified
location	Path of location in which the snapshot is available; it cannot be used in any other location than that (eg: de/fkb for karlsruhe)

UPDATE SNAPSHOT

Updates meta data of a snapshot. This meta data can be relevant as they trigger other features like Live Vertical Scaling of CPU or RAM.

```
updateSnapshot(request : updateSnapshotRequest): updateSnapshotResponse
```

Request parameters

Name	Description	Required
snapshotId	Identifier of the snapshot that shall get updated	yes
description	text field to add additional information (e.g. for details about time or reason why snapshot was created)	
snapshotName	name of snapshot	
bootable	flag of type boolean	
osType	flag to specify OS type; relevant for license accounting in case snapshot gets redeployed on further virtual storage instances (WINDOWS, LINUX, OTHER, UNKNOWN)	
cpuHotPlug	snapshot contains capabilities to hotplug CPU; flag of type boolean	
cpuHotUnplug	snapshot contains capabilities to hotUnplug CPU; flag of type boolean	
ramHotPlug	snapshot contains capabilities to hotplug RAM; flag of type boolean	
ramHotUnPlug	snapshot contains capabilities to hotUn plug RAM; flag of type boolean	
nicHotPlug	snapshot contains capabilities to hotplug NIC; flag of type boolean	
nicHotUnPlug	snapshot contains capabilities to hotunplug NIC; flag of type boolean	
discVirtioHotPlug	snapshot contains capabilities to hotplug storages which are connected through VirtIO bustype; flag of type boolean	
discVirtioHotUnPlug	snapshot contains capabilities to hotUnplug storages which are connected through VirtIO bustype; flag of type boolean	

Memory HotPlug Capability for Windows

Memory HotPlug is not supported for virtual machines booting from a Windows image. When setting the option to TRUE it will return an error (bad_request).

Error Codes

Code	Description
RESOURCE_NOT_FOUND	Specified StorageID does not exist
UNAUTHORIZED	User is not authorized to access the storage

DELETE SNAPSHOT

Deletes a snapshot. Please be aware that deleted snapshots and related data in this snapshot cannot be recovered anymore.

```
deleteSnapshot(request : DeleteSnapshotRequest): DeleteSnapshotResponse
```

Request parameters

Name	Description	Required
snapshotId	Identifier of the snapshot that shall get deleted	Yes

Error Codes

Code	Description
RESOURCE_NOT_FOUND	Specified snapshot does not exist
UNAUTHORIZED	User is not authorized to access the server

ROLLBACK SNAPSHOT

Using the rollback option you may redeploy the snapshotted state on a storage.

Attention: The current state of the storage will be lost unless you create another snapshot before rolling back.

```
rollbackSnapshot(request : rollbackSnapshotRequest):
rollbackSnapshotResponse
```

Request parameters

Name	Description	Required
storageId	Identifier of the virtual storage as target for the snapshot	Yes
snapshotId	Identifier of the snapshot that shall get deployed on a virtual storage	Yes

Error Codes

Code	Description
BAD_REQUEST	Size of snapshot is bigger than size of storage
RESOURCE_NOT_FOUND	Specified snapshot or specified storage does not exist
UNAUTHORIZED	User is not authorized to access the storage

LOAD BALANCER OPERATIONS

A Load Balancer connected to a LAN will not distribute traffic to any server, until it is specified to do so. In the current version, a Load Balancer cannot distribute traffic across multiple data centers or LANs. Load Balancer and servers must always be in the same LAN. See also **Register Servers On Load Balancer**

Index

Create Load Balancer

Get Load Balancer

Get All Load Balancer

Update Load Balancer

Register Servers On Load Balancer

Deregister Servers Load Balancer

Activate/Deactivate Load Balancer

Delete Load Balancer

CREATE LOAD BALANCER

Creates a virtual Load Balancer within an existing virtual data center. If the server is not yet a member of the LAN, a new NIC is created to the server, connected to the LAN and registered with the Load Balancer.

```
createLoadBalancer(request : CreateLbRequest) : CreateLbResponse
```

Note

Load Balancer and the balanced NIC of the server share one single IP address.

Request parameters

Name	Description	Required
dataCenterId	data center ID wherein the load balancer is to be created	Yes
loadBalancerName	Name of the load balancer to be created	
loadBalancerAlgorithm	load balancer algorithm. ROUND_ROBIN is default and the only supported algorithm at the moment	
ip	A DHCP IP address is being assigned to the load balancer automatically by ProfitBricks. A private IP can be defined by the user. Additional, public IPs can be reserved and assigned to the load balancer manually. See also Reserve Public IP Block	
lanId	Identifier of the target LAN ID > 0 If the specified LAN ID does not exist or if LAN ID is not specified, a new LAN with the given ID / with a next available ID starting from 1 will be created respectively	
serverIds	Identifier of all servers, to which the load balancer is going to distribute traffic. Servers can also be registered to load balancer later . See also Register Servers On Load Balancer.	

Response parameters

Name	Description	
loadBalancerId	Identifier of the load balancer	

Error Codes

Code	Description
BAD_REQUEST	Invalid characters in load balancer name Invalid private IP address Load balancer and target servers are not located in the same data center Invalid LAN ID ≤ 0 Reserved IP and load balancer of the data center are located in different regions
RESOURCE_NOT_FOUND	Specified data center and/or target server does not exist
UNAUTHORIZED	User is not authorized to access the data center and/or target server User is not authorized to use the customer reserved IP address

GET LOAD BALANCER

Returns information about a virtual load balancer.

```
getLoadBalancer(loadBalancerId : String) : LoadBalancer
```

Request parameters

Name	Description	
loadBalancerId	Identifier of the load balancer	Yes

Response parameters

Name	Description	Constraints
loadBalancerId	Identifier of the load balancer	
creationTime	Time when the specified virtual load balancer has been created	Nullable
lastModificationTime	Time when the specified virtual load balancer has last been modified	Nullable
provisioningState	Current provisioning state of the specified virtual storage (INACTIVE, INPROCESS, AVAILABLE, DELETED, ERROR)	
loadBalancerName	Outputs the name of the load balancer	
loadBalancerAlgorithm	Outputs the load balancer algorithm. ROUND_ROBIN is the default and currently the only supported algorithm	
ip	IP address assigned to the load balancer	
lanId	Identifier of the target LAN ID to which the load balancer and the balanced servers are connected	
balancedServers	List all servers registered to the load balancer, including their identifier (serverId), name (serverName), identifier of the balanced Nic (balancedNicId) through which the load balancer forwards traffic to the server, and if load balancing is activate on this server (activate)	Emptiable
firewall	Firewall created on the load balancer. See also Add Firewall Rules	Nullable
internetAccess	Returns TRUE if load balancer is connected to a public LAN	

Error Codes

Code	Description
RESOURCE_DELETED	The load balancer has been deleted

RESOURCE_NOT_FOUND	Specified load balancer does not exist
UNAUTHORIZED	User is not authorized to access the load balancer

GET ALL LOAD BALANCERS

Returns information about all virtual load balancer.

```
getAllLoadBalancers(): List<loadbalancers>
```

Request parameters

None

Response parameters

Name	Description	Constraints
loadBalancerId	Identifier of the load balancer	
creationTime	Time when the specified virtual load balancer has been created	Nullable
lastModificationTime	Time when the specified virtual load balancer has last been modified	Nullable
provisioningState	Current provisioning state of the specified virtual storage (INACTIVE, INPROCESS, AVAILABLE, DELETED, ERROR)	
loadBalancerName	Outputs the name of the load balancer	
loadBalancerAlgorithm	Outputs the load balancer algorithm. ROUND_ROBIN is the default and currently the only supported algorithm	
ip	IP address assigned to the load balancer	
lanId	Identifier of the target LAN ID to which the load balancer and the balanced servers are connected	
balancedServers	List all servers registered to the load balancer, including their identifier (serverId), name (serverName), identifier of the balanced Nic (balancedNicId) through which the load balancer forwards traffic to the server, and if load balancing is activate on this server (activate)	Emptiable
firewall	Firewall created on the load balancer. See also Add Firewall Rules	Nullable
internetAccess	Returns TRUE if load balancer is connected to a public	

	LAN	

Error Codes

Code	Description
RESOURCE_DELETED	The load balancer has been deleted
RESOURCE_NOT_FOUND	Specified load balancer does not exist
UNAUTHORIZED	User is not authorized to access the load balancer

UPDATE LOAD BALANCER

Changes the settings of an existing virtual load balancer.

```
updateLoadBalancer(request : UpdateLbRequest)
```

Request parameters

Name	Description	Required
loadBalancerId	Identifier of the load balancer	Yes
loadBalancerName	Renames the target virtual load balancer	
loadBalancerAlgorithm	Sets the load balancer algorithm. ROUND_ROBIN is the default and currently the only supported algorithm	
ip	Updates the IP address of the load balancer with the specified IP. All servers connected to the load balancer will have their primary IP address updated with the same IP address of the load balancer implicitly Additional customer reserved IP addresses, which have been added to the Server's NIC, remain unchanged Set ip to empty, to reset the IP of load balancer with a ProfitBricks assigned IP address.	

Error Codes

Code	Description
BAD_REQUEST	Invalid characters used in the virtual load balancer name Invalid private IP address Reserved IP and load balancer of the data center are located in different regions
RESOURCE_NOT_FOUND	Specified load balancer does not exist
UNAUTHORIZED	User is not authorized to access the load balancer

REGISTER SERVERS ON LOAD BALANCER

Adds new servers to an existing Load Balancer within the respective LAN. If the server is not yet a member of the LAN, a new NIC is created to the server, connected to the LAN and registered with the Load Balancer. The load balancer will distribute traffic to the server through this balanced NIC. If the server is already a member of the LAN, the appropriate NIC is used as balanced NIC. A server can be registered to more than one Load Balancer.

```
registerServersOnLoadBalancer (serverIds : List<String>, loadBalancerId
: String) : RegisterLbServerResponse
```

Request parameters

Name	Description	Required
serverIds	Identifier of servers to be registered with target load balancer.	Yes
loadBalancerId	Identifier of target load balancer	Yes

Response parameters

Name	Description	
loadBalancerId	Identifier of the target load balancer	
lanId	Identifier of the LAN, to which load balancer and balanced servers are connected	
balancedServers	Lists all servers registered to the load balancer, including their identifier (serverId), name (serverName), identifier of the balanced Nic (balancedNicId) through which the load balancer forwards traffic to the server, and if load balancing is activate on this server (activate)	

Error Codes

Code	Description
BAD_REQUEST	Load balancer and registered servers are not located within the same data center
RESOURCE_NOT_FOUND	Specified load balancer / server does not exist
UNAUTHORIZED	User is not authorized to access load balancer / servers User is not authorized to use the customer reserved IP address

DEREGISTER SERVERS ON LOAD BALANCER

By deregistering a server, the server is being removed from the load balancer but still remains connected to the same LAN. The primary IP address of the NIC, through which the load balancer distributed traffic to the server before, is reset and replaced by a ProfitBricks assigned IP address.

```
deregisterServersOnLoadBalancer (serverIds: List<String>,
loadBalancerId : String)
```

Request parameters

Name	Description	Required
serverIds	Identifier of servers to be deregistered with target load balancer	Yes
loadBalancerId	Identifier of target load balancer	Yes

Error Codes

Code	Description
PROVISIONING_NO_CHANGES	No servers to deregister found, because none of them have been registered to the load balancer before
RESOURCE_NOT_FOUND	Specified load balancer / server does not exist
UNAUTHORIZED	User is not authorized to access load balancer / servers

DELETE LOAD BALANCER

Deletes an existing load balancer. All associated balanced servers get removed from the LAN and associated server NICs get deleted.

```
deleteLoadBalancer(loadBalancerId : String)
```

Request parameters

Name	Description	Required
loadBalancerId	Identifier of target load balancer	Yes

Error Codes

Code	Description
RESOURCE_NOT_FOUND	Specified load balancer does not exist
UNAUTHORIZED	User is not authorized to access load balancer

FIREWALL OPERATIONS

All firewall rules are accept-rules. An active firewall will only allow traffic following the user defined rules. An active firewall with no accept-rules, will drop all traffic. An inactive firewall will allow all traffic.

ARP packets are always accepted.

Incoming packets replied to outgoing packets are always accepted.

Important Notices

No firewall rules should be placed between a load balancer and its load balanced servers.

Index

Add Firewall Rules

Get Firewall

Get All Firewalls

Remove Firewall Rules

Activate/Deactivate Firewall

Delete Firewall

ADD FIREWALL RULES TO NIC

Adds accept-rules to the firewall of a given NIC. If no firewall exists, a new inactive firewall is created. Firewalls can be activated / deactivated; additional accept-rules can be added anytime.

```
addFirewallRulesToNic(request : List<FirewallRuleRequest>, nicId :
String) : Firewall
```

Request parameters

Name	Description	Required
`firewallRuleRequests`	Defines a set of accept-rules applied for the target NIC.	Yes

	Each rule is defined as follows (*)	
nicId	Identifier of target NIC	Yes

(*) Firewall rule requests

Name	Description	Required
protocol	Select allowed protocol (TCP, UPD, ICMP, ANY). Default is ANY	
sourceMac	Only traffic originated from the respective MAC address is allowed. Valid format: aa:bb:cc:dd:ee:ff. Left empty to allow all source MAC address	
sourceIp	Only traffic originated from the respective IPv4 address is allowed. Leave empty to allow all source IPs	
targetIp	In case the target NIC has multiple IP addresses, only traffic directed to the respective IP address of the NIC is allowed. Left empty to allow all target IPs	
portRangeStart	Defines the start range of the allowed port (from 1 to 65534) if protocol TCP or UDP is chosen. Leave portRangeStart and portRangeEnd empty to allow all ports	Yes if `portRangeEnd` is specified
portRangeEnd	Defines the end range of the allowed port (from 1 to 65534) if protocol TCP or UDP is chosen. Leave portRangeStart and portRangeEnd empty to allow all ports	Yes if `portRangeStart` is specified
icmpType	Defines the allowed type (from 0 to 254) if protocol ICMP is chosen. Leave empty to allow all types	
icmpCode	Defines the allowed code (from 0 to 254) if protocol ICMP is chosen. Leave empty to allow all types	
name	Name of the firewall rule	

Note

To be able to define `portRangeStart` and `portRangeEnd`, either protocol TCP or UDP must be chosen.

Response parameters

Name	Description	Constraints
firewallId	Identifier of the target firewall	
nicId	Identifier of the NIC	
provisioningState	Current provisioning state of the firewall (INACTIVE, INPROCESS, AVAILABLE, DELETED, ERROR)	

active	Firewall is active [TRUE/FALSE]	
firewallRules	List all firewall rules, including their identifier (firewallRuleId), firewallRuleName, protocol, sourceMac, sourceIp, targetIP, portRangeStart, portRangeEnd, icmpType, and icmpCode	

Error Codes

Code	Description
BAD_REQUEST	Protocol conflicts when a rule defines allowed protocol TCP but icmpType is set (and vice-versa) portRangeStart is specified but portRangeEnd is left empty (and vice-versa) portRangeStart > portRangeEnd Value is out of valid range
RESOURCE_NOT_FOUND	Specified NIC does not exist
UNAUTHORIZED	User is not authorized to access the NIC

GET FIREWALL

Returns information about the respective firewall. Each rule has an identifier for later modification. To get firewall of a given NIC, see also **Get NIC**

```
getFirewall( firewallId : String)
```

Request parameters

Name	Description	Required
firewallId	Identifier of the target firewall	Yes

Response parameters

Name	Description	Constraints
firewallId	Identifier of the target firewall	
nicId	Identifier of the target NIC	
provisioningState	Current provisioning state of the firewall (INACTIVE, INPROCESS, AVAILABLE, DELETED, ERROR)	
active	Firewall is active [TRUE/FALSE]	
firewallRules	List all firewall rules, including their identifier (firewallRuleId), name, protocol, sourceMac, sourceIp, targetIp, portRangeStart, portRangeEnd, icmpType, and icmpCode	Emptiable

Error Codes

Code	Description
RESOURCE_DELETED	The firewall has been deleted
RESOURCE_NOT_FOUND	Specified firewall does not exist
UNAUTHORIZED	User is not authorized to access the firewall

GET ALL FIREWALLS

Returns information about all configured firewall. Each rule has an identifier for later modification. To get firewall of a given NIC, see also **Get NIC**

```
getAllFirewalls(): List<firewalls>
```

Request parameters

None

Response parameters

Name	Description	Constraints
firewallId	Identifier of the target firewall	
nicId	Identifier of the target NIC	
provisioningState	Current provisioning state of the firewall (INACTIVE, INPROCESS, AVAILABLE, DELETED, ERROR)	
active	Firewall is active [TRUE/FALSE]	
firewallRules	List all firewall rules, including their identifier (firewallRuleId), name, protocol, sourceMac, sourceIp, targetIp, portRangeStart, portRangeEnd, icmpType, and icmpCode	Emptiable

Error Codes

Code	Description
RESOURCE_DELETED	The firewall has been deleted
RESOURCE_NOT_FOUND	Specified firewall does not exist
UNAUTHORIZED	User is not authorized to access the firewall

REMOVE FIREWALL RULES

Removes firewall rule(s) by specifying their identifiers. See also getFirewall() or getNic() to retrieve identifiers of firewall rules.

```
removeFirewallRules(firewallRuleIds : List<String>)
```

Request parameters

Name	Description	Required
`firewallRuleIds`	Identifiers of the target firewall rules	Yes

Error Codes

Code	Description
`BAD_REQUEST`	Firewalls rules are not located within the same data center
`PROVISIONING_NO_CHANGES`	No firewall rules found to remove
`UNAUTHORIZED`	User is not authorized to access firewall rules

ACTIVATE/DEACTIVATE FIREWALL

Activates/deactivates one or several firewall(s) of a given data center.

```
activateFirewalls(firewallIds : List<String>)
```

```
deactivateFirewalls(firewallIds : List<String>)
```

Request parameters

Name	Description	Required
`firewallIds`	Identifier(s) of the target firewall(s)	Yes

Error Codes

Code	Description
`BAD_REQUEST`	Firewalls are not located within the same data center
`PROVISIONING_NO_CHANGES`	No inactive/active firewalls to activate/deactivate found
`RESOURCE_NOT_FOUND`	Specified firewalls do not exist

UNAUTHORIZED	User is not authorized to access firewall(s)

DELETE FIREWALLS

Deletes one or several firewall(s) of a given data center.

```
deleteFirewalls(firewallIds : List<String>)
```

Request parameters

Name	Description	Required
firewallIds	Identifier(s) of the target firewall(s)	Yes

Error Codes

Code	Description
BAD_REQUEST	Firewalls are not located within the same data center
RESOURCE_NOT_FOUND	Specified firewalls do not exist
UNAUTHORIZED	User is not authorized to access firewall(s)

CD-ROM/DVD DRIVE OPERATIONS

Index

Add CD-ROM/DVD Drive To Server

Remove CD-ROM/DVD-Drive from Server

ADD CD-ROM/DVD DRIVE TO SERVER

Adds a CD-ROM/DVD drive to an existing virtual server. Maximum CD-ROM/DVD Drives are currently 2.

```
addRomDriveToServer(request : RomDriveRequest)
```

Request parameters

Name	Description	Required
imageId	Identifier of the CD-ROM/DVD image	Yes
serverId	Identifier of the target virtual server	Yes
deviceNumber	Device number of the CD-ROM/DVD drive connected to the server. If no device number is set, a new device number will be assigned to the CD-ROM/DVD drive automatically	

A CD-ROM/DVD drive is identified by its CD-ROM/DVD (ISO) image and therefore have the same ID. Before a CD-ROM/DVD (ISO) image can be assigned to a server, it must be uploaded on the associated FTP server.

Error Codes

Code	Description
BAD_REQUEST	Wrong image type (not a CD-ROM/DVD (ISO) image) Image and server are not located in the same region
RESOURCE_NOT_FOUND	Specified server or image does not exist
OVER_LIMIT_SETTING	number of CD-ROM/DVD Drives limit exceeded (max. 2 per virtual server)
UNAUTHORIZED	User is not authorized to access the server

REMOVE CD-ROM/DVD DRIVE FROM SERVER

Removes a CD-ROM/DVD drive from an existing virtual server.

```
removeRomDriveFromServer(imageId : String, serverId : String)
```

A CD-ROM/DVD drive is always identified by its CD-ROM/DVD (ISO) image ID.

Request parameters

Name	Description	Required
imageId	Identifier of the CD-ROM/DVD image	Yes
serverId	Identifier of the target virtual server	Yes

Error Codes

Code	Description
RESOURCE_NOT_FOUND	Specified server or image does not exist

UNAUTHORIZED	User is not authorized to access the server

IMAGE OPERATIONS

Index

Set Image OS Type

Get Image

Get All Images

Update Image

Delete Image

SET IMAGE OS TYPE

Sets the OS Type of an individual HDD and/or CD-ROM/DVD image that has been uploaded on the ProfitBricks FTP server.

The default OS Type of an uploaded image is UNKOWN. Due to Microsoft's terms and conditions, the user has to set the OS Type of an uploaded Windows image to WINDOWS and therewith accepts the terms and conditions as well as the pricing. Any server that is booted from the image will inherit the OS Type of the image automatically.

```
setImageOsType(request : imageOsTypeRequest)
```

Request parameters

Name	Description	Required
imageId	Identifier of the target HDD or CD-ROM/DVD image	Yes
osType	OS Type of the target HDD or CD-ROM/DVD image (WINDOWS, LINUX, OTHER, UNKNOWN)	Yes

Error Codes

Code	Description
RESOURCE_NOT_FOUND	Specified image does not exist
UNAUTHORIZED	User is not authorized to access the image

GET IMAGE

Returns information about a HDD or CD-ROM/DVD (ISO) image.

```
getImage(imageId : String): Image
```

Request parameters

Name	Description	Required
imageId	Identifier of the target image	Yes

Response parameters

Name	Description	Constraints
imageId	Identifier of the target image	
imageName	Name of the image	
imageType	Image type (HDD or CD-ROM/DVD (ISO) image)	
imageSize	Size of Image	
writeable	Image is writeable (TRUE/FALSE)	
location	Location where the image has been uploaded to (us/las, de/fkb, de/fra, us/lasdev)	
cpuHotPlug	Image supports CPU Hot-Plugging (TRUE/FALSE)	Emptiable
cpuHotUnplug	Image supports CPU Hot-UnPlugging (TRUE/FALSE)	Emptiable
ramHotPlug	Image supports ram Hot-Plugging (TRUE/FALSE)	Emptiable
ramHotUnPlug	Image supports ram Hot-UnPlugging (TRUE/FALSE)	Emptiable
nicHotPlug	Image supports ram Hot-UnPlugging (TRUE/FALSE)	Emptiable
nicHotUnPlug	Image supports ram Hot-UnPlugging (TRUE/FALSE)	Emptiable
discVirtioHotPlug	Image supports storage Hot-Plugging (TRUE/FALSE)	Emptiable
discVirtioHotUnPlug	Image supports storage Hot-UnPlugging (TRUE/FALSE)	Emptiable
osType	OS Type of the target image (WINDOWS, LINUX, OTHER,UNKNOWN)	

serverIds	Lists all servers (by ID) on which the specified CD-ROM/DVD image is being used. See also Add CD-ROM/DVD Drive To Server	Emptiable
bootable	image contains capabilities to boot from; flag of type boolean (TRUE/FALSE)	

The location of an image indicates the location where the image has been uploaded. Therefore the location of image can be us/las [Las Vegas], de/fkb [Karlsruhe], de/fra [Frankfurt] or us/lasdev [DevOps Data Center, Las Vegas]. Please make sure all images are located at the same location as the data center, as images from different locations cannot be used.

Error Codes

Code	Description
RESOURCE_NOT_FOUND	Specified image does not exist
UNAUTHORIZED	User is not authorized to access the image

GET ALL IMAGES

Outputs a list of all HDD and/or CD-ROM/DVD images existing on or uploaded to the ProfitBricks FTP server.

```
getAllImages(): List<Image>
```

Response parameters

Name	Description	Constraints
imageId	Identifier of the image	
imageName	Name of the image	
imageSize	Size of image	
imageType	Image type HDD or CD-ROM/DVD (ISO) image	
writeable	Image is writeable (TRUE/FALSE)	
cpuHotPlug	Image supports CPU Hot-Plugging (TRUE/FALSE)	
cpuHotUnPlug	Image supports CPU Hot-UnPlugging (TRUE/FALSE)	
ramHotPlug	Image supports ram Hot-Plugging (TRUE/FALSE)	
ramHotUnPlug	Image supports ram Hot-UnPlugging (TRUE/FALSE)	
nicHotPlug	Image contains capabilities to hotplug NIC; flag of type boolean	
nicHotUnPlug	Image contains capabilities to hotUnplug NIC; flag of type boolean	

discVirtioHotPlug	Image supports storage Hot-Plugging (TRUE/FALSE)	
discVirtioHotUnPlug	Image supports storage Hot-UnPlugging (TRUE/FALSE)	
serverIds	Lists all servers (by ID) on which the specified CD-ROM/DVD image is being used. See also **Add CD-ROM/DVD Drive To Server**	Emptiable
location	Location where the image has been uploaded to (us/las, de/fkb, de/fra, us/lasdev)	
osType	OS Type of an image (WINDOWS, LINUX, OTHER, UNKNOWN)	
public	Shows if this image is a publicly available image provided by ProfitBricks or a private image only accessible by the user logged in	
bootable	image contains capabilities to boot from; flag of type boolean (TRUE/FALSE)	

UPDATE IMAGE

Updates information about a HDD or CD-ROM/DVD (ISO) image.

```
updateImage(imageId : String): Image
```

Request parameters

Name	Description	Required
imageUuid	Identifier of the target image	**yes**
name	Name of the image	**yes**
osType	OS Type of the target image (WINDOWS, LINUX, OTHER,UNKNOWN)	
description	An option to provide own description to the image	
cpuHotPlug	image contains capabilities to hotplug CPU; flag of type boolean	
cpuHotUnPlug	image contains capabilities to hotUnplug CPU; flag of type boolean	
ramHotPlug	image contains capabilities to hotplug RAM; flag of type boolean	
ramHotUnPlug	image contains capabilities to hotUnplug RAM; flag of type boolean	
nicHotPlug	image contains capabilities to hotplug NIC; flag of type boolean	
nicHotUnPlug	image contains capabilities to hotUnplug NIC; flag of type	

	boolean	
discVirtioHotPlug	image contains capabilities to hotplug storages which are connected through VirtIO bustype; flag of type boolean	
discVirtioHotUnPlug	image contains capabilities to hotunplug storages which are connected through VirtIO bustype; flag of type boolean	
bootable	image contains capabilities to boot from; flag of type boolean	

Memory HotPlug Capability for Windows

Memory HotPlug is not supported for virtual machines booting from a Windows image. When setting the option to TRUE it will return an error (bad_request).

Error Codes

Code	Description
RESOURCE_NOT_FOUND	Specified image does not exist
UNAUTHORIZED	User is not authorized to access the image

DELETE IMAGE

Deletes an image. Please be aware that deleted images and related data in this image cannot be recovered anymore.

```
deleteImage(request : DeleteImageRequest): DeleteImageResponse
```

Request parameters

Name	Description	Required
imageId	Identifier of the image that shall get deleted	Yes

Error Codes

Code	Description
RESOURCE_NOT_FOUND	Specified snapshot does not exist
UNAUTHORIZED	User is not authorized to access the server

NIC

ProfitBricks currently only supports Internet Protocol Version 4 (IPv4).

The user can specify and assign local IPs manually to a NIC, which is connected to a Private LAN. Valid IP addresses for Private LANs are 10.0.0.0/8, 172.16.0.0/12 or 192.168.0.0/16.

In a Public LAN, a random DHCP IP address is assigned to each connected NIC by default. This IP Address is automatically generated and will change eventually, e.g. during a server reboot or while disconnecting and reconnecting a LAN to the internet.

For assigning a static IP to a NIC, it is necessary to supply a reserved public IP address. Public IP Addresses can be reserved block-wise and manually assigned to a NIC by the user. See also **Reserve Public IP Block**

Index

Create NIC

Get NIC

Get All NIC

Set Internet Access

Update NIC

Delete NIC

CREATE NIC

Creates a NIC on an existing virtual server.

```
createNic(request : CreateNicRequest) : CreateNicResponse
```

Request parameters

Name	Description	Required
serverId	Identifier of the target virtual server	Yes

lanId	Identifier of the target LAN > 0 that is to be connected to the specified virtual server. If no LAN exists for such ID, a new LAN with the given ID will be created.	Yes
ip	Public/private IP address. See also **NIC Operations** and **Reserve Public IP Block** for more information	
nicName	Names the NIC	
dhcpActive	Toggles usage of ProfitBricks DHCP	

! A NIC with LAN ID = 0 can be created but will not be connected to any LANs.

Response parameters

Name	Description
nicId	Identifier of the target NIC

Error Codes

Code	Description
RESOURCE_NOT_FOUND	Specified server does not exist
OVER_LIMIT_SETTING	NIC limit exceeded (max. 6 per virtual server)
UNAUTHORIZED	User is not authorized to access the server

GET NIC

Returns information about the state and configuration of an existing NIC.

```
getNic(nicId : String) : Nic
```

Request parameters

Name	Description	Required
nicId	Identifier of the target NIC	Yes

Response parameters

Name	Description	Constraints
nicName	NIC name	Emptiable
nicId	Identifier of the virtual NIC	
serverId	Identifier of the target server	
lanId	Identifier of the target LAN	

internetAccess	Internet Access (TRUE/FALSE)	
ips	Lists all public and private IP addresses assigned to the NIC	
macAddress	MAC address (automatically) assigned to the NIC	
firewall	Lists all firewall rules applied to the NIC. See also Get Firewall	
dhcpActive	Toggles usage of ProfitBricks DHCP	
gatewayIp	IP address of gateway	
provisioningState	Current provisioning state of the firewall (INACTIVE, INPROCESS, AVAILABLE, DELETED, ERROR)	

Error Codes

Code	Description
RESOURCE_DELETED	Specified NIC has been deleted by the user
RESOURCE_NOT_FOUND	Specified NIC does not exist
UNAUTHORIZED	User is not authorized to access the server User is not authorized to use the specified public IP

GET ALL NIC

Returns information about the state and configuration of all existing NICs.

```
getAllNic(): List<Nics>
```

Response parameters

Name	Description	Constraints
nicName	NIC name	Emptiable
nicId	Identifier of the virtual NIC	
serverId	Identifier of the target server	
lanId	Identifier of the target LAN	
internetAccess	Internet Access (TRUE/FALSE)	
ips	Lists all public and private IP addresses assigned to the NIC	
macAddress	MAC address (automatically) assigned to the NIC	
firewall	Lists all firewall rules applied to the NIC. See also Get Firewall	
dhcpActive	Toggles usage of ProfitBricks DHCP	
gatewayIp	IP address of gateway	

provisioningState	Current provisioning state of the firewall (INACTIVE, INPROCESS, AVAILABLE, DELETED, ERROR)	

SET INTERNET ACCESS

Connects an existing NIC to a public LAN to get internet access.

```
setInternetAccess(dataCenterId : String, lanId : int, internetAccess : boolean) : void
```

Request parameters

Name	Description	Required
datacenterId	Identifier of the target data center	Yes
lanId	Identifier of the target LAN	Yes
internetAccess	Internet access (TRUE/FALSE)	Yes

Error Codes

Code	Description
RESOURCE_NOT_FOUND	Specified NIC/server does not exist
UNAUTHORIZED	User is not authorized to access the server/ data center

Update NIC

Changes the settings of an existing NIC.

```
updateNic(request : UpdateNicRequest): UpdateNicResponse
```

Request parameters

Name	Description	Required
nicId	Identifier of the target NIC	Yes
lanId	Identifier of the target LAN connected to the NIC. If no LAN exists for such ID, a new LAN with the given ID will be created. To disconnect a NIC from a LAN, set LAN ID to 0	
ip	Public/private IP address. Set to empty to reset the IP address. See also NIC Operations and Reserve Public IP Block for more	

	information	
nicName	Names the NIC	
dhcpActive	Toggles usage of ProfitBricks DHCP	

Important

Primary IP addresses of NICs connected to a load balancer can only be changed by changing the IP of the load balancer. See also **Update Load Balancer**

It is also possible to add additional reserved IPs to a NIC. See also **Add Public IP to NIC**

Valid Private IP Addresses

The user can specify and assign private IPs manually. Valid IP addresses for private (sub)networks are 10.0.0.0/8, 172.16.0.0/12 or 192.168.0.0/16

Error Codes

Code	Description
BAD_REQUEST	Invalid private IP address Cannot change primary IP address of a balanced NIC Reserved IP and NIC are from different regions Cannot update a balanced NIC to connect to a another LAN. User must deregister server from Load Balancer first, or connect to LAN via a different NIC
RESOURCE_NOT_FOUND	Specified NIC/server does not exist
UNAUTHORIZED	User is not authorized to access the NIC User is not the owner of the specified public IP

DELETE NIC

Deletes an existing NIC.

```
deleteNic(nicId : String)
```

Request parameters

Name	Description	Required
nicId	Identifier of the target NIC	Yes

Error Codes

Code	Description
RESOURCE_NOT_FOUND	Specified NIC does not exist
UNAUTHORIZED	User is not authorized to access the NIC

PUBLIC IP OPERATIONS

Index

Reserve Public IP Block

Add Public IP to NIC

Get Public IP Block

Get All Public IP Blocks

Remove Public IP from NIC

Release Public IP Block

RESERVE PUBLIC IP BLOCK

Reserves a specific amount of public IPs which can be manually assigned to a NIC by the user.

```
reservePublicIpBlock(request : createReservePublicIPBlockRequest):
reserverPublicIpBlockResponse
```

Reserving Public IPs

The user can reserve any number of public IP addresses. Though, several public IPs can only be reserved and released in an IP block.

Request parameters

Name	Description	Required
blockSize	Block size / amount of IPs to reserve	Yes

`location`	Select location to reserve IP block, choose one from the below list 1. de/fkb -> (refers to Karlsruhe) 2. de/fra -> (refers to Frankfurt) 3. us/las -> (refers to Las Vegas) 4. us/lasdev -> (refer to DevOps Data Center in Las Vegas)	Yes

Response parameters

Name	Description	Constraints
`blockId`	Identifier of the reserved IP block	
`ips`	Lists all reserved IPs inside this IP block	
`location`	Location wherein the IP block has been reserved (us/las, de/fkb, de/fra, us/lasdev)	

IPs Location

Customer reserved IP addresses cannot be assigned to a NIC located in a different location.

Error Codes

Code	Description
`SERVER_EXCEED_CAPACITY`	No free IP address blocks are currently available for reservation

ADD PUBLIC IP TO NIC

Adds an existing reserved public IP to a NIC. This operation is required, when dealing with reserved public IPs to ensure proper routing by the ProfitBricks cloud networking layer.

As Load Balancer and the balanced NIC of the server are only allowed to share one single IP address, this operation is disabled for the NICs of all balanced servers. To assign additional IP Address(es) to a balanced server, an additional NIC can be created on the respective server, to which the Public IP(s) can then be added.

```
addPublicIpToNic(String ip, String nicId)
```

Important

Reserved IP and the target NIC must be in the same region.

Request parameters

Name	Description	Required
ip	Reserved IP	Yes
nicId	Identifier of the target NIC	Yes

Error Codes

Code	Description
RESOURCE_NOT_FOUND	Specified IP or NIC does not exist
UNAUTHORIZED	User is not authorized to access the server/NIC User is not the owner of the specified reserved IP
BAD_REQUEST	Reserved IP and NIC of data center are not in the same location User cannot assign more than one IP address to a balanced NIC of a server

GET PUBLIC IP BLOCK

Returns reserved IPs and connected NICs of a particular public IP block.

```
getPublicIpBlock(blockId : String) : ipBlock
```

Request parameters

Name	Description	Required
blockId	Identifier of the IP block	Yes

Response parameters

Name	Description	Constraints
blockId	Identifier of the IP block	
location	Location where the IP block has been reserved for use (us/las, de/fkb, de/fra, us/lasdev)	
publicIps	Identifier of the public IPs within the respective block, including their IP and (if assigned to a NIC) the respective NIC ID	partial (NIC)

Error Codes

Code	Description
RESOURCE_NOT_FOUND	Specified IP block does not exist
UNAUTHORIZED	User is not authorized to access the IP block

GET ALL PUBLIC IP BLOCKS

Returns a list of all public IP blocks reserved by the user, including the reserved IPs and connected NICs.

```
getAllPublicIpBlocks(): List<IpBlock>
```

Response parameters

Name	Description	Constraints
`List<IpBlock>`	Lists all IP blocks reserved by the customer	
`blockId`	Identifier of the IP block	
`location`	Location where the IP block has been reserved for use (us/las, de/fkb, de/fra, us/lasdev)	
`publicIps`	Identifier of the public IPs within the respective block, including their IP and (if assigned to a NIC) the respective NIC ID	partial (NIC)

REMOVE PUBLIC IP FROM NIC

Removes a reserved public IP from a NIC. This operation is required, when dealing with reserved public IPs to ensure proper routing by the ProfitBricks cloud networking layer.

```
removePublicIpFromNic(ip : String, nicId : String)
```

Request parameters

Name	Description	Required
`ip`	IP address	Yes
`nicId`	Identifier of the target NIC	Yes

Error Codes

Code	Description
`RESOURCE_NOT_FOUND`	Specified IP or NIC does not exist
`UNAUTHORIZED`	User is not authorized to access the NIC User is not the owner of the specified reserved IP

RELEASE PUBLIC IP BLOCK

Releases an existing block of reserved public IPs.

```
releasePublicIpBlock(blockId : String)
```

> **Important**
>
> Before releasing an IP block, please ensure that no IP address in the respective IP block is assigned to a NIC anymore. Otherwise the operation will fail.

Request parameters

Name	Description	Required
`blockId`	Identifier of the reserved IP block	Yes

Error Codes

Code	Description
`BAD_REQUEST`	One or more IPs of the IP block are still in use by a NIC
`RESOURCE_NOT_FOUND`	Specified IP block does not exist
`UNAUTHORIZED`	User is not the owner of the public IP block

ProfitBricks API Documentation

Legal Notice

API v1.3

This document is intended for software developers interested in developing applications using the ProfitBricks Application Programming Interface (API). The document is for informational purposes only and is provided “AS IS.”

PROFITBRICKS MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, AS TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS DOCUMENT AND RESERVES THE RIGHT TO MAKE CHANGES TO SPECIFICATIONS AND PRODUCT/SERVICES DESCRIPTION AT ANY TIME WITHOUT NOTICE. PROFITBRICKS SERVICES OFFERINGS ARE SUBJECT TO CHANGE WITHOUT NOTICE. USERS MUST TAKE FULL RESPONSIBILITY FOR APPLICATION OF ANY SERVICES MENTIONED HEREIN. EXCEPT AS SET FORTH IN PROFITBRICKS GENERAL TERMS AND CONDITIONS AND/OR CLOUD TERMS OF SERVICE, PROFITBRICKS ASSUMES NO LIABILITY WHATSOEVER, AND DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO ITS SERVICES INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT.

Except as expressly provided in any written license agreement from ProfitBricks GmbH, the furnishing of this document does not give you any license to patents, trademarks, copyrights, or other intellectual property.

All product names and trademarks used in this document are for identification purposes only and are property of their respective owners.