

API Guide

Table of Contents

1	API Authentication	5
2	HTTP Methods	6
3	HTTP response codes	7
4	Formatting and naming conventions	8
5	FAQ	9
6	Container Servers	10
6.1	Get List of All Container Servers	10
6.2	Get Container Server Details.....	15
6.3	Get Statuses for all Container Servers	19
6.4	Get Container Server Status	20
6.5	Get Container Server Cloud Config.....	22
6.6	Add Container Server	23
6.6.1	Page history	24
6.7	Add Container Server Cloud Config	25
6.8	View Encrypted Container Server Password	25
6.9	Build or Rebuild Container Server	26
6.10	Edit Container Server.....	27
6.11	Edit Container Server Cloud Config.....	29
6.12	Change Container Server Owner	29
6.13	Reset Container Server Root Password.....	30
6.14	Migrate Container Server	31
6.15	Set VIP Status for Container Server	32
6.16	Delete Container Server	32
6.17	Start up Container Server	33
6.18	Segregate Container Server	33
6.19	Desegregate Container Server	34
6.20	Reboot Container Server	35
6.21	Reboot Container Server in Recovery	35
6.22	Boot Container Server from ISO	35
6.23	Suspend Container Server.....	36
6.24	Unlock Container Server	36
6.25	Unsuspend Container Server	37
6.26	Shut down Container Server.....	37
6.27	Stop Container Server	37
6.28	Open Container Server Console.....	38
6.29	Container Server Billing Statistics.....	38
6.30	Search Container Server by Label.....	40
6.31	Get Container Server CPU Usage Statistics	41
6.32	Add/Edit Admin/User Note for Container Server	42
6.32.1	Add/Edit User Note	43
7	Virtual Servers	45
7.1	Get List of VSs.....	45
7.1.1	Page History:	51
7.2	Get VS Details	51
7.2.1	Page History:	56
7.3	Get Statuses for all Virtual Servers	56
7.4	Get Virtual Server Status.....	57
7.5	Add VS.....	58
7.5.1	Page history	64
7.6	Add Instance Package VS.....	65
7.7	Add VMware VS	68
7.7.1	Page history	70
7.8	View Encrypted VS Password	70

7.9	Build or Rebuild VS	71
7.10	Edit VS	72
7.11	Change VS Owner	73
7.12	Reset VS Root Password	74
7.13	Set SSH Keys	75
7.14	Migrate VS	76
7.14.1	Page history	76
7.15	Migrate VS and Disks	76
7.15.1	Page history	77
7.16	Set VIP Status for VS.....	77
7.17	Delete VS.....	78
7.18	Start up VS.....	78
7.19	Segregate VS.....	79
7.20	Desegregate VS.....	80
7.21	Reboot VS.....	80
7.22	Enable Accelerator for Virtual Server	81
7.23	Disable Accelerator for Virtual Server.....	81
7.24	Purge File(s)	82
7.25	Purge All Content.....	82
7.26	Reboot VS in Recovery.....	83
7.27	Reboot VS from ISO	83
7.28	Boot VS from ISO	84
7.29	Suspend VS	84
7.30	Unlock VS	84
7.31	Unsuspend VS	85
7.32	Shut down VS	85
7.33	Stop VS	85
7.34	Open VS Console	86
7.35	VS Autoscaling.....	86
7.35.1	Enable Autoscaling for VS	86
7.35.2	Get the List of Autoscaling Rules for VS	86
7.35.3	Create Autoscaling Rule for VS	88
7.35.4	Edit Autoscaling Rule for VS	89
7.35.5	Delete Autoscaling Rule.....	89
7.35.6	Disable Autoscaling for VS.....	89
7.36	VS Billing Statistics	89
7.37	Search VS by Label	93
7.38	Get VS CPU Usage Statistics	93
7.39	Add/Edit Admin/User Note for Virtual Server.....	94
7.39.1	Add/Edit User Note	95
7.40	Enable Booting from CD for ISO Virtual Server.....	96
7.41	Disable Booting from CD for ISO Virtual Server	97
7.42	Add Media File to vCloud VS.....	97
7.43	Remove Media File from vCloud VS.....	97
7.44	View vCloud VS Guest Customization.....	98
7.45	Get List of Service Add-ons Assigned to VS	99
7.46	Assign Service Add-on to VS.....	100
7.47	Unassign Service Add-on from VS	101
7.48	Use VS as Gateway.....	102

The API enables cloud integration with third party applications. You can manage every aspect of your cloud through the API. This guide is a complete reference for all API calls and includes detailed API information, code and output examples.

The API is RESTful

- All function calls respond to XML and JSON exchange formats
- All function calls need authorization and authentication (Basic HTTP or API key)
- The API is backward compatible within one major version. However, a new major version might include changes that are not backward compatible with the previous one.

1 API Authentication

To authenticate using HTTP Basic, just use your username/password combination. Curl example:

```
curl -u user:userpass
```

To authenticate using API key, put your account email as a login and the key to the server as a password.

2 HTTP Methods

The API uses the following HTTP methods:

GET - used for retrieving information from a particular URI

POST - used for creating new object and adding new transactions into the queue

PUT - used for altering object properties

NOTE: updated_at value is changed in PUT requests even if the request fails.

DELETE - used for object deletion

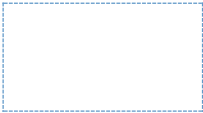



3 HTTP response codes

The API returns appropriate HTTP status codes for every request:

200 OK	The request completed successfully
204 No content	The request completed successfully. The 204 status is returned on DELETE and PUT requests
201 Scheduled	The request has been accepted and scheduled for processing
403 Forbidden	The request is correct, but could not be processed.
404 Not Found	The requested URL is incorrect or the resource does not exist. For example, if you request to delete a user with ID {5}, but there is no such a user in the cloud, you will get a 404 error.
422 Unprocessable Entity	The sent parameters are erroneous.
500 Internal Server Error	An error occurred. Please contact support.
503 Service Unavailable	The request cannot be handled currently, due to a temporary overloading or maintenance of the server. This condition is temporary and the request will be handed after a certain delay.

4 Formatting and naming conventions

The table below represents all the existing formatting and naming conventions used in this guide:

Convention	Explanation	Example
user:userpass	stands for <i>username:password</i> combination	Admin:123456
Cloud HM.test	stands for address, where your Control Panel is located	<i>Example.com</i>
:id	stands for the resource ID. Sometimes also: :resource_id	23
<i>italics</i>	all the parameters are italicised	<i>currency_code</i>
* (asterisk)	marks the required parameters	<i>label</i> *
preformatted	indicates request examples in XML or JSON	GET /roles.xml
	Code block indicates console requests and response examples.	
info	An info message emphasizes or explains the information within the chapter.	Clicking the OFF button performs graceful shutdown and then powers off the VS.
note	A note message contains information essential for the task completion.	The maximum length of a Mount Point is 256 characters.
warning	A warning message informs you of something you should not do or be cautious.	You won't be able to restore a VS after deleting it.
Figure 1 (lightbulb) 	The element showing new parameters added in the latest release of API.	 <i>limit_type</i> – hourly or monthly limit type set for the resource

5 FAQ

Q: Is it possible to enable API access via https?

A: We can enable https for your cloud, which can be used for both WebUI access and API access. Or you can do so yourself: the Apache config file is located at: `/etc/httpd/conf.d/CloudHM.conf`

Q: Can you create a VS on behalf of another user?

A: No. It is possible to switch VS owners, however. Refer to **Change a VS owner** section for details.

Q: How are passwords stored – in plain text?

A: No, passwords are not stored in plain text. Except for a login and password combination, you can use email + API key combination to authorize a user via the API. API keys can be generated and changed easily on a user's profile page (as well as through the API). For security reasons we recommend users authenticate through the API key, not the login and password.

Q: Which parameters are required, and which are optional?

A: Required parameters are marked in this guide with an asterisk *.

Where:

resourceId - the ID of CDN resource

pastHour - the amount of bandwidth for the past hour

currentHour - the amount of bandwidth for this hour

bandwidth - the amount of transmitted bandwidth

6 Container Servers

Container servers in Cloud HM are based on a CoreOS template and deployed on compute resources. Container servers have their own root accounts, so that container server owners can fully control, configure and manage their servers.

We do not support container servers on CloudBoot compute resources running on CentOS 5.

6.1 Get List of All Container Servers

To get the list of all container servers, use the following request:

```
GET /container_servers.xml
GET /container_servers.json
```

XML Request example

```
curl -i -X GET -H 'Accept: application/xml' -H 'Content-type:
application/xml' -u user:userpass --url http://Cloud
HM.test/container_servers.xml
```

JSON Request example

```
curl -i -X GET -H 'Accept: application/json' -H 'Content-type:
application/json' -u user:userpass --url http://Cloud
HM.test/container_servers.json
```

XML Output example

```
<?xml version="1.0" encoding="UTF-8"?>
<container_servers type="array">
  <container_server>
    <add_to_marketplace nil="true"/>
    <admin_note nil="true"/>
    <allowed_hot_migrate type="boolean">true</allowed_hot_migrate>
    <allowed_swap type="boolean">true</allowed_swap>
    <autoscale_service nil="true"/>
    <booted type="boolean">true</booted>
    <built type="boolean">true</built>
    <cdboot type="boolean">false</cdboot>
    <cores_per_socket type="integer">0</cores_per_socket>
    <cpu_shares type="integer">100</cpu_shares>
    <cpu_sockets nil="true"/>
    <cpu_threads nil="true"/>
    <cpu_units type="integer">1000</cpu_units>
    <cpus type="integer">1</cpus>
    <created_at type="datetime">2016-09-30T11:45:00+03:00</created_at>
    <customer_network_id nil="true"/>
    <deleted_at nil="true"/>
    <draas_keys type="array"/>
    <draas_mode type="integer">0</draas_mode>
    <edge_server_type nil="true"/>
```

```

    <enable_autoscale nil="true"/>
    <firewall_notrack type="boolean">false</firewall_notrack>
    <hostname>sgsg</hostname>
    <hot_add_cpu nil="true"/>
    <hot_add_memory nil="true"/>
    <hypervisor_id type="integer">1</hypervisor_id>
    <id type="integer">9255</id>
    <identifier>l4zzl458vgbeo3</identifier>
    <initial_root_password>n158758Aquhc2</initial_root_password>
    <initial_root_password_encrypted
type="boolean">false</initial_root_password_encrypted>
    <instance_package_id nil="true"/>
    <iso_id nil="true"/>
    <label>test</label>

<local_remote_access_ip_address>111.177.7.33</local_remote_access_ip_addr
ess>
    <local_remote_access_port
type="integer">5777</local_remote_access_port>
    <locked type="boolean">false</locked>
    <memory type="integer">512</memory>
    <min_disk_size type="integer">5</min_disk_size>
    <note nil="true"/>
    <operating_system>coreos</operating_system>
    <operating_system_distro>coreos</operating_system_distro>
    <preferred_hvs type="array"/>
    <recovery_mode nil="true"/>
    <remote_access_password>nmDf15487Upi0</remote_access_password>
    <service_password nil="true"/>
    <state>delivered</state>
    <storage_server_type nil="true"/>
    <strict_virtual_machine_id nil="true"/>
    <suspended type="boolean">false</suspended>
    <template_id type="integer">477</template_id>
    <template_label>Coreos current x64</template_label>
    <time_zone nil="true"/>
    <updated_at type="datetime">2016-10-21T14:13:17+03:00</updated_at>
    <user_id type="integer">758</user_id>
    <vapp_id nil="true"/>
    <vcenter_moref nil="true"/>
    <vip nil="true"/>
    <vmware_tools nil="true"/>
    <xen_id type="integer">992</xen_id>
    <ip_addresses type="array">
      <ip_address>
        <address>4.4.5.26</address>
        <broadcast>4.4.5.355</broadcast>
        <created_at type="datetime">2016-04-
14T13:34:38+03:00</created_at>
        <customer_network_id nil="true"/>
        <disallowed_primary type="boolean">false</disallowed_primary>
        <gateway>4.4.4.2</gateway>
        <hypervisor_id nil="true"/>
        <id type="integer">159669</id>
        <ip_address_pool_id nil="true"/>
        <network_address>4.4.4.0</network_address>
        <network_id type="integer">1</network_id>
        <pxe type="boolean">false</pxe>
        <updated_at type="datetime">2016-08-
19T16:13:29+03:00</updated_at>
        <user_id nil="true"/>
        <free type="boolean">false</free>
        <netmask>266.2566.266.0</netmask>
      </ip_address>
    </ip_addresses>
    <monthly_bandwidth_used type="integer">79935</monthly_bandwidth_used>

```

```

    <total_disk_size type="integer">6</total_disk_size>
    <support_incremental_backups
type="boolean">>false</support_incremental_backups>
    <cpu_priority type="integer">100</cpu_priority>
    <built_from_iso type="boolean">>false</built_from_iso>
    <acceleration type="boolean">>false</acceleration>
    <acceleration_status>Inactive</acceleration_status>
    <hypervisor_type>kvm</hypervisor_type>
    <price_per_hour type="float">500.0</price_per_hour>
    <price_per_hour_powered_off
type="float">300.0</price_per_hour_powered_off>
  </container_server>
  <container_server>...</container_server>
</container_servers>

```

Where:

add_to_marketplace - empty for container servers; used for edge servers only

admin_note - an optional note of the administrator

allowed_hot_migrate - true if the template, on which the container server is based, supports hot migration; otherwise false

allowed_swap - true if swap disk is allowed (depends on the template the container server is based on); otherwise false

autoscale_service - currently, autoscaling is not available for container servers

booted - true if the container server is running, otherwise false

built - true if the container server is built, otherwise false

cores_per_socket - the amount of cores per socket

cpu_shares - CPU priority in percent's

cpu_sockets - the amount of CPU sockets per core. This parameter can be set for KVM compute resources only by those users who have *Enable CPU topology* permission granted

cpu_threads - the amount of CPU threads per core. This parameter can be set for KVM compute resources only by those users who have *Enable CPU topology* permission granted

cpu_units - the amount of CPU units per core if the CPU priority is replaced with CPU units in user billing plan

cpus - the number of allocated CPU cores

created_at - the date in the [YYYY][MM][DD]T[hh][mm][ss]Z format

customer_network_id - ID of a customer network

deleted_at - time when the container server was deleted

edge_server_type - true if this is the edge server

enable_autoscale - true if autoscaling is allowed for this container server. Currently, autoscaling is not available for container servers.

firewall_notrack - true if the NOTRACK rule is set in iptables

hostname - the name of your host

hot_add_cpu - true, if the CPU parameter can be changed without rebooting the container server, otherwise false

hot_add_memory - true, if the memory parameter can be changed without rebooting the container server, otherwise false

hypervisor_id - the ID of the compute resource used by this container server

id - the container server ID

identifier - the container server identifier

initial_root_password - the container server root password

initial_root_password_encrypted - true, if the root password is encrypted, otherwise false

instance_package_id - ID of the instance package. Instance packages are not currently available for container servers.

iso_id - the ID of the ISO the container server is based on

label - the container server label

local_remote_access_ip_address - IP address used for remote access

local_remote_access_port - the port ID used for console access

locked - true if the container server is locked; otherwise false

memory - the RAM size allocated to this container server

min_disk_size - the minimum disk size required to build a container server from a specified template

note - an optional reminder for this container server made by a user account

operating_system - operating system used by the container server

operating_system_distro - the distribution of the OS from which this container server is built

preferred_hvs - the array of preferable compute resources based on compute zone that meet some container server configuration settings

recovery_mode - true if recovery mode allowed. Otherwise false

remote_access_password - the password for the remote access

service_password - service account password

state - parameter reserved for future use

storage_server_type - true if this is a storage server

strict_virtual_machine_id - the ID of a container server that will never reside on the same compute resource with this container server

suspended - true if container server is suspended, otherwise false

template_id - the ID of the template the container server is based on

template_label - the name of the template from which this container server is built

time_zone - the time zone set for the container server. This parameter is applicable only to Windows KVM and XEN servers.

updated_at - the date when the VS was updated in the [YYYY][MM][DD]T[hh][mm][ss]Z format

user_id - the ID of a user assigned to this container server

vapp_id - this parameter is not applicable for container servers

vcenter_moref - this parameter is not applicable for container servers

vip - true if the container server has VIP status (gives migration priority)

vmware_tools - this parameter is not applicable for container servers

xen_id - the container server ID set by the virtualization engine

ip_addresses - an array of IP addresses assigned to this container server and their details:

address - IP address

broadcast - broadcast address

created_at - the date in the [YYYY][MM][DD]T[hh][mm][ss]Z format

customer_network_id - the ID of the customer VLAN the IP address belongs to

disallowed_primary - true if not allowed to be used as primary, otherwise false

gateway - gateway address

hypervisor_id - the ID of a compute resource the IP address is associated with

id - the ID of the IP address

ip_address_pool_id - ID of the IP address pool the IP address is associated with

network_address - the address of the network

network_id - the ID of the network

pxe - true, if this address can be used for cloudbooting a compute resource

updated_at - the date in the [YYYY][MM][DD]T[hh][mm][ss]Z format

user_id - the ID of the user this IP address is assigned to

free - true if free, otherwise false

netmask - netmask for the IP address

monthly_bandwidth_used - container server monthly bandwidth in KB

total_disk_size - the total disk size in GB of all disks assigned to container server

support_incremental_backups - 1, if container server supports incremental backups, and 0 if it does not. Currently, backups are not available for container servers.

cpu_priority - this is a new parameter reserved for further use; currently will have the same value as *cpu_shares*

built_from_iso - true if the container server is built from ISO; otherwise false

acceleration - true if acceleration is enabled for the container server ; otherwise false. Acceleration is not available for container servers.

acceleration_status - the status of acceleration: active or inactive. Acceleration is not available for container servers.

hypervisor_type - the type of the compute resource the container server is built on (for example: *xen*, *kvm*, *vcloud*, *vmware*)

price_per_hour - server's price per hour

price_per_hour_powered_off - price per hour when server is powered off

6.2 Get Container Server Details

To get the details of a particular container server, use the following request:

```
GET /container_servers/container_server_id.xml
GET /container_servers/container_server_id.json
```

XML Request example

```
curl -i -X GET -H 'Accept: application/xml' -H 'Content-type:
application/xml' -u user:userpass --url http://Cloud
HM.test/container_servers/:container_server_id.xml
```

JSON Request example

```
curl -i -X GET -H 'Accept: application/json' -H 'Content-type:
application/json' -u user:userpass --url http://Cloud
HM.test/container_servers/:container_server_id.json
```

XML Output example

```
<?xml version="1.0" encoding="UTF-8"?>
<container_server>
  <add_to_marketplace nil="true"/>
  <admin_note nil="true"/>
  <allowed_hot_migrate type="boolean">true</allowed_hot_migrate>
  <allowed_swap type="boolean">true</allowed_swap>
  <autoscale_service nil="true"/>
  <booted type="boolean">true</booted>
  <built type="boolean">true</built>
  <cdboot type="boolean">false</cdboot>
  <cores_per_socket type="integer">0</cores_per_socket>
  <cpu_shares type="integer">100</cpu_shares>
  <cpu_sockets nil="true"/>
  <cpu_threads nil="true"/>
  <cpu_units type="integer">1000</cpu_units>
  <cpus type="integer">1</cpus>
  <created_at type="datetime">2016-09-30T11:45:00+03:00</created_at>
  <customer_network_id nil="true"/>
  <deleted_at nil="true"/>
  <draas_keys type="array"/>
  <draas_mode type="integer">0</draas_mode>
  <edge_server_type nil="true"/>
  <enable_autoscale nil="true"/>
  <firewall_notrack type="boolean">false</firewall_notrack>
  <hostname>sgsg</hostname>
  <hot_add_cpu nil="true"/>
  <hot_add_memory nil="true"/>
  <hypervisor_id type="integer">1</hypervisor_id>
  <id type="integer">9234</id>
  <identifier>l4zz3h7458eo3</identifier>
  <initial_root_password>ny6325hc2</initial_root_password>
  <initial_root_password_encrypted
type="boolean">false</initial_root_password_encrypted>
  <instance_package_id nil="true"/>
  <iso_id nil="true"/>
  <label>gfgsg</label>

<local_remote_access_ip_address>188.176.7.47</local_remote_access_ip_addr
```

```

ess>
  <local_remote_access_port
type="integer">5777</local_remote_access_port>
  <locked type="boolean">>false</locked>
  <memory type="integer">512</memory>
  <min_disk_size type="integer">5</min_disk_size>
  <note nil="true"/>
  <operating_system>coreos</operating_system>
  <operating_system_distro>coreos</operating_system_distro>
  <preferred_hvs type="array"/>
  <recovery_mode nil="true"/>
  <remote_access_password>nmDfve52145i0</remote_access_password>
  <service_password nil="true"/>
  <state>delivered</state>
  <storage_server_type nil="true"/>
  <strict_virtual_machine_id nil="true"/>
  <suspended type="boolean">>false</suspended>
  <template_id type="integer">489</template_id>
  <template_label>Coreos current x64</template_label>
  <time_zone nil="true"/>
  <updated_at type="datetime">2016-10-21T14:13:17+03:00</updated_at>
  <user_id type="integer">946</user_id>
  <vapp_id nil="true"/>
  <vcenter_moref nil="true"/>
  <vip nil="true"/>
  <vmware_tools nil="true"/>
  <xen_id type="integer">902</xen_id>
  <ip_addresses type="array">
    <ip_address>
      <address>4.4.4.16</address>
      <broadcast>4.4.4.266</broadcast>
      <created_at type="datetime">2016-04-
14T13:34:38+03:00</created_at>
      <customer_network_id nil="true"/>
      <disallowed_primary type="boolean">>false</disallowed_primary>
      <gateway>4.4.4.1</gateway>
      <hypervisor_id nil="true"/>
      <id type="integer">180969</id>
      <ip_address_pool_id nil="true"/>
      <network_address>4.4.4.0</network_address>
      <network_id type="integer">1</network_id>
      <pxe type="boolean">>false</pxe>
      <updated_at type="datetime">2016-08-
19T16:13:29+03:00</updated_at>
      <user_id nil="true"/>
      <free type="boolean">>false</free>
      <netmask>266.266.266.0</netmask>
    </ip_address>
  </ip_addresses>
  <monthly_bandwidth_used type="integer">77135</monthly_bandwidth_used>
  <total_disk_size type="integer">6</total_disk_size>
  <support_incremental_backups
type="boolean">>false</support_incremental_backups>
  <cpu_priority type="integer">100</cpu_priority>
  <built_from_iso type="boolean">>false</built_from_iso>
  <acceleration type="boolean">>false</acceleration>
  <acceleration_status>Inactive</acceleration_status>
  <hypervisor_type>kvm</hypervisor_type>
  <price_per_hour type="float">500.0</price_per_hour>
  <price_per_hour_powered_off
type="float">300.0</price_per_hour_powered_off>
  </container_server>

```

Where:

add_to_marketplace - empty for container servers; used for edge servers only

admin_note - an optional note of the administrator

allowed_hot_migrate - true if the template, on which the container server is based, supports hot migration; otherwise false

allowed_swap - true if swap disk is allowed (depends on the template the container server is based on); otherwise false

autoscale_service - currently, autoscaling is not available for container servers

booted - true if the container server is running, otherwise false

built - true if the container server is built, otherwise false

cores_per_socket - the amount of cores per socket

cpu_shares - CPU priority in percent's

cpu_sockets - the amount of CPU sockets per core. This parameter can be set for KVM compute resources only by those users who have *Enable CPU topology* permission granted

cpu_threads - the amount of CPU threads per core. This parameter can be set for KVM compute resources only by those users who have *Enable CPU topology* permission granted

cpu_units - the amount of CPU units per core if the CPU priority is replaced with CPU units in user billing plan

cpus - the number of allocated CPU cores

created_at - the date in the [YYYY][MM][DD]T[hh][mm][ss]Z format

customer_network_id - ID of a customer network

deleted_at - time when the container server was deleted

edge_server_type - true if this is the edge server

enable_autoscale - true if autoscaling is allowed for this container server. Currently, autoscaling is not available for container servers.

firewall_notrack - true if the NOTRACK rule is set in iptables

hostname - the name of your host

hot_add_cpu - true, if the CPU parameter can be changed without rebooting the container server, otherwise false

hot_add_memory - true, if the memory parameter can be changed without rebooting the container server, otherwise false

hypervisor_id - the ID of the compute resource used by this container server

id - the container server ID

identifier - the container server identifier

initial_root_password - the container server root password

initial_root_password_encrypted - true, if the root password is encrypted, otherwise false

instance_package_id - ID of the instance package. Instance packages are not currently available for container servers.

iso_id - the ID of the ISO the container server is based on

label - the container server label

local_remote_access_ip_address - IP address used for remote access

local_remote_access_port - the port ID used for console access

locked - true if the container server is locked; otherwise false

memory - the RAM size allocated to this container server

min_disk_size - the minimum disk size required to build a container server from a specified template

note - an optional reminder for this container server made by a user account

operating_system - operating system used by the container server

operating_system_distro - the distribution of the OS from which this container server is built

preferred_hvs - the array of preferable compute resources based on compute zone that meet some container server configuration settings

recovery_mode - true if recovery mode allowed. Otherwise false

remote_access_password - the password for the remote access

service_password - service account password

state - parameter reserved for future use

storage_server_type - true if this is a storage server

strict_virtual_machine_id - the ID of a container server that will never reside on the same compute resource with this container server

suspended - true if container server is suspended, otherwise false

template_id - the ID of the template the container server is based on

template_label - the name of the template from which this container server is built

time_zone - the time zone set for the container server. This parameter is applicable only to Windows KVM and XEN servers.

updated_at - the date when the VS was updated in the [YYYY][MM][DD]T[hh][mm][ss]Z format

user_id - the ID of a user assigned to this container server

vapp_id - this parameter is not applicable for container servers

vcenter_moref - this parameter is not applicable for container servers

vip - true if the container server has VIP status (gives migration priority)

vmware_tools - this parameter is not applicable for container servers

xen_id - the container server ID set by the virtualization engine

ip_addresses - an array of IP addresses assigned to this container server and their details:

address - IP address

broadcast - broadcast address

created_at - the date in the [YYYY][MM][DD]T[hh][mm][ss]Z format

customer_network_id - the ID of the customer VLAN the IP address belongs to

disallowed_primary - true if not allowed to be used as primary, otherwise false

gateway - gateway address

hypervisor_id - the ID of a compute resource the IP address is associated with

id - the ID of the IP address

ip_address_pool_id - ID of the IP address pool the IP address is associated with

network_address - the address of the network

network_id - the ID of the network

pxe - true, if this address can be used for cloudbooting a compute resource

updated_at - the date in the [YYYY][MM][DD]T[hh][mm][ss]Z format

user_id - the ID of the user this IP address is assigned to

free - true if free, otherwise false

netmask - netmask for the IP address

monthly_bandwidth_used - container server monthly bandwidth in KB

total_disk_size - the total disk size in GB of all disks assigned to container server

support_incremental_backups - 1, if container server supports incremental backups, and 0 if it does not. Currently, backups are not available for container servers.

cpu_priority - this is a new parameter reserved for further use; currently will have the same value as *cpu_shares*

built_from_iso - true if the container server is built from ISO; otherwise false

acceleration - true if acceleration is enabled for the container server ; otherwise false. Acceleration is not available for container servers.

acceleration_status - the status of acceleration: active or inactive. Acceleration is not available for container servers.

hypervisor_type - the type of the compute resource the container server is built on (for example: *xen*, *kvm*, *vcloud*, *vmware*)

price_per_hour - server's price per hour

price_per_hour_powered_off - price per hour when server is powered off

6.3 Get Statuses for all Container Servers

Use the following API call to get statuses for all container servers:

```
GET /container_servers/status.xml
```

```
GET /container_servers/status.json
```

XML Request example

```
curl -i -X GET -H 'Accept: application/xml' -H 'Content-type: application/xml' -u user:userpass --url http://Cloud.HM.test/container_servers/status.xml
```

JSON Request example

```
curl -i -X GET -H 'Accept: application/json' -H 'Content-type: application/json' -u user:userpass --url http://CloudHM.test/container_servers/status.json
```

XML Output example

```
<container_servers type="array">
  <container_server>
    <id type="integer">1</id>
    <identifier>okulsief887rqm</identifier>
    <hostname>vl.test</hostname>
    <template_id type="integer">1</template_id>
    <built type="boolean">>true</built>
    <locked type="boolean">>false</locked>
    <booted type="boolean">>true</booted>
    <operating_system>linux</operating_system>
    <suspended type="boolean">>false</suspended>
    <enable_autoscale type="boolean">>true</enable_autoscale>
    <state>new</state>
  </container_server>
  ...
</container_servers>
```

Where:

id - container server ID

identifier - the container server identifier

hostname - the name of your host

template_id - the ID of the template the container server is based on

built - true if the container server is built, otherwise false

locked - true if the container server is locked; otherwise false

booted - true if the container server is running, otherwise false

operating_system - operating system used by the container server

suspended - true if container server is suspended, otherwise false

enable_autoscale - true if autoscaling is allowed for this container server

state - container server state

6.4 Get Container Server Status

This parameter has been added in the 3.1 version.

Use the following API call to get status for a particular container server:

```
GET /container_servers/:container_server_id/status.xml
```

```
GET /container_servers/:container_server_id/status.json
```

XML Request example

```
curl -i -X GET -H 'Accept: application/xml' -H 'Content-type:
application/xml' -u user:password --url http://Cloud
HM.test/container_servers/:container_server_id/status.xml
```

JSON Request example

```
curl -i -X GET -H 'Accept: application/json' -H 'Content-type:
application/json' -u user:password --url http://Cloud
HM.test/container_servers/:container_server_id/status.json
```

XML Output example

```
<container_servers type="array">
  <container_server>
    <id type="integer">48</id>
    <identifier>b266b5h5et39iy</identifier>
    <hostname>qaaexp</hostname>
    <template_id type="integer">111</template_id>
    <built type="boolean">>true</built>
    <locked type="boolean">>false</locked>
    <booted type="boolean">>true</booted>
    <operating_system>windows</operating_system>
    <suspended type="boolean">>false</suspended>
    <enable_autoscale type="boolean">>false</enable_autoscale>
    <state>new</state>
  </container_server>
</container_servers>
```

Where:

id - container server ID

identifier - the container server identifier

hostname - the name of your host

template_id - the ID of the template the container server is based on

built - true if the container server is built, otherwise false

locked - true if the container server is locked; otherwise false

booted - true if the container server is running, otherwise false

operating_system - operating system used by the container server

suspended - true if container server is suspended, otherwise false

enable_autoscale - true if autoscaling is allowed for this container server

state - container server state

6.5 Get Container Server Cloud Config

To get a container server cloud config, use the following request:

```
GET /container_servers/:container_server_id/cloud_config.xml
GET /container_servers/:container_server_id/cloud_config.json
```

XML Request example

```
curl -i -X GET -H 'Accept: application/xml' -H 'Content-type:
application/xml' -u user:userpass --url http://Cloud
HM.test/container_servers/:container_server_id/cloud_config.xml
```

JSON Request example

```
curl -i -X GET -H 'Accept: application/json' -H 'Content-type:
application/json' -u user:userpass --url http://Cloud
HM.test/container_servers/:container_server_id/cloud_config.json
```

XML Output example

```
<?xml version="1.0" encoding="UTF-8"?>
<container_server>
  <cloud_config>#cloud-config
write-files:
  - path: /etc/hosts
    permissions: '0644'
    content: |
      111.222.33.444 master1 coreos00
      555.666.77.888 master2 coreos01
coreos:
  etcd2:
    name: master2
    initial-cluster:
master1=http://111.222.33.444:2380,master2=http://555.666.77.888:2380
    initial-advertise-peer-urls: http://$public_ipv4:2380
    advertise-client-urls:
http://$public_ipv4:2379,http://$public_ipv4:4001
    listen-client-urls: http://0.0.0.0:2379,http://0.0.0.0:4001
    listen-peer-urls: http://$public_ipv4:2380,http://$public_ipv4:7001
  fleet:
    public-ip: $public_ipv4
    metadata: "role=master"
  flannel:
    interface: $public_ipv4
units:
  - name: etcd2.service
    command: start
  - name: fleet.service
    command: start
  - name: flannel.service
    command: start</cloud_config>
</container_server>
```

Where:

cloud_config - the cloud-config file, which enables you to customize different OS elements, such as network configuration, user accounts, etc..

6.6 Add Container Server

To add a container server, use the following request:

```
POST /container_servers.xml
POST /container_servers.json
```

XML Request example

```
curl -i -X POST -H 'Accept: application/xml' -H 'Content-type:
application/xml' -u user:userpass -d
'<container_server><hypervisor_group_id>6</hypervisor_group_id><hyperviso
r_id>4</hypervisor_id><hostname>myContainer</hostname><domain>localdomain
</domain><primary_network_group_id>11</primary_network_group_id><required
_virtual_machine_build>1</required_virtual_machine_build><memory>512</mem
ory><label>
TEST</label><cpus>1</cpus><swap_disk_min_iops>100</swap_disk_min_iops><da
ta_store_group_swap_id>13</data_store_group_swap_id><rate_limit>0</rate_l
imit><cpu_shares>1</cpu_shares><enable_autoscale>False</enable_autoscale>
<template_id>742</template_id><primary_disk_min_iops>100</primary_disk_mi
n_iops><initial_root_password>password</initial_root_password><selected_i
p_address>5.1.1.12</selected_ip_address><data_store_group_primary_id>13</
data_store_group_primary_id><primary_disk_size>5</primary_disk_size><requ
ired_ip_address_assignment>1</required_ip_address_assignment><swap_disk_s
ize>1</swap_disk_size></container_server>' --url http://Cloud
HM.test/container_servers.xml
```

JSON Request example

```
curl -i -X POST -H 'Accept: application/json' -H 'Content-type:
application/json' -u user:userpass -d '{"container_server":
{"hypervisor_group_id": "6", "hypervisor_id": "4",
"hostname": "my_container",
"domain": "localdomain", "primary_network_group_id": "11",
"required_virtual_machine_build": "1", "memory": "512", "label": "zaza
TEST", "cpus": "1", "swap_disk_min_iops": "100",
"data_store_group_swap_id": "13", "rate_limit": "0", "cpu_shares": "1",
"enable_autoscale": "False", "template_id": "742",
"primary_disk_min_iops": "100", "initial_root_password": "password",
"selected_ip_address": "5.1.1.12", "data_store_group_primary_id": "13",
"primary_disk_size": "5", "required_ip_address_assignment": "1",
"swap_disk_size": "1"}}' --url http://Cloud HM.test/container_servers.json
```

Where:

hypervisor_group_id - the ID of the compute zone in which the container server will be created

hypervisor_id - the ID of a compute resource where the container server will be built

hostname - container server hostname

domain - specify the domain for this VS. The default value is *localdomain*. This parameter is not applicable for Windows virtual servers.

primary_network_group_id - the ID of the primary network group

required_virtual_machine_build - set 1 to build container server automatically

memory - amount of RAM assigned to the container server

label - name of the container server

cpus - number of CPUs assigned to the container server

swap_disk_min_iops - minimum number of IO operations per second for swap disk (this is a SolidFire related parameter)

data_store_group_swap_id - set the ID of the data store zone to which this swap disk is allocated

rate_limit - set max port speed in Mbps or set 0 to get maximum port speed allowed by your billing plan. If this parameter is omitted or sent without value, the default port speed will be configured for the container server. The default port speed depends on the maximum port speed set in your billing plan and the *Max network interface port speed* parameter at **Control Panel > Settings > Configuration**. The system identifies which of the two values (in the billing plan or in the configuration) is lower and sets it as the default port speed during container server creation.

cpu_shares - for KVM compute resource the CPU priority value is always 100. For XEN, set a custom value. The default value for XEN is 1.

enable_autoscale - true if autoscaling is allowed for this container server. Autoscaling is not currently available for container servers.

template_id - the ID of the template the container server is based on

primary_disk_min_iops - minimum number of IO operations per second for primary disk (this is a SolidFire related parameter)

initial_root_password - the root password for a container server. Optional, if none specified, the system will provide a random password. It can consist of 6-32 characters, letters [A-Za-z], digits [0-9], dash [-] and lower dash [_], and the following special characters: ~ ! @ # \$ * _ - + = ` \ \ { } [] : ; ' , . ? /. You can use both lower- and uppercase letters.

selected_ip_address - an IP address to assign to this container server; if the parameter *required_ip_address_assignment* was set "1" but this parameter *selected_ip_address* is empty - the first available IP address will be assigned to container server automatically

data_store_group_primary_id - set the ID of the data store zone to which this primary disk is allocated

primary_disk_size - set the disk space for this container server

required_ip_address_assignment - set "1" if you want container server to be created with already assigned IP address, otherwise set "0"; IP address can be assigned after container server creation.

swap_disk_size - set swap space

6.6.1 Page history

v.5.4

- added *domain* parameter
- added *selected_ip_address* parameter
- removed *selected_ip_address_id* parameter

6.7 Add Container Server Cloud Config

To add a container server cloud config, use the following request:

```
PATCH /container_servers/:container_server_id/cloud_config.xml
PATCH /container_servers/:container_server_id/cloud_config.json
```

XML Request example

```
curl -i -X PATCH -H 'Accept: application/xml' -H 'Content-type:
application/xml' -u user:userpass -d
'<container_server><cloud_config><user's
cloud_config></cloud_config></container_server>' --url http://Cloud
HM.test/container_servers/:container_server_id/cloud_config.xml
```

JSON Request example

```
curl -i -X PATCH -H 'Accept: application/json' -H 'Content-type:
application/json' -u user:userpass -d '{"container_server":
{"cloud_config": "user's cloud_config"}}' --url http://Cloud
HM.test/container_servers/:container_server_id/cloud_config.json
```

Where:

cloud_config - add the cloud-config file, which enables you to customize different OS elements, such as network configuration, user accounts, etc..

6.8 View Encrypted Container Server Password

If the container server was created with password encryption enabled, you can use the following API call to view the password (the request returns the decrypted password):

```
GET /container_servers/:id/with_decrypted_password.xml
GET /container_servers/:id/with_decrypted_password.json
```

XML Request example:

```
curl -X GET -u user:userpass http://Cloud
HM.test/container_servers/:id/with_decrypted_password.xml?initial_root_p
assword_encryption_key=encryptionkey
```

JSON Request example:

```
curl -X GET -u user:userpass http://Cloud
HM.test/container_servers/:id/with_decrypted_password.json?initial_root_p
assword_encryption_key=encryptionkey
```

Where:

id - the container server's ID

6.9 Build or Rebuild Container Server

To build or rebuild a container server, use the following methods:

```
POST /container_servers/:container_server_id/build.xml
POST /container_servers/:container_server_id/build.json
```

XML Request example

```
curl -i -X POST -H 'Accept: application/xml' -H 'Content-type:
application/xml' -u user:userpass -d '<?xml version="1.0" encoding="UTF-
8"?><container_server><template_id>1</template_id><required_startup>1</re
quired_startup></container_server>' --url http://Cloud
HM.test/container_servers/:container_server_id/build.xml
```

JSON Request example

```
curl -i -X POST -H 'Accept: application/json' -H 'Content-type:
application/json' -u user:userpass -d
'{"container_server":{"template_id":"1","required_startup":"1"}}' --url
http://Cloud HM.test/container_servers/:container_server_id/build.json
```

Where:

*template_id** - the ID of a template from which a container server should be built.

required_startup - set to 1 if you wish to start a container server after it is built. Otherwise set to 0.

initial_root_password_encryption_key - specify the password encryption passphrase

for **Windows** templates you should specify the licensing type:

licensing_type - the type of a license: *mak*, *kms* or *user own* license

licensing_key - the key of a license, required if you have selected **OWN** licensing type, and not required for MAK and KMS licensing types

licensing_server_id - the ID of a template group where the **KMS** server details are indicated and to which the template belongs (either directly or through the child group)

6.10 Edit Container Server

To edit a container server, use the following request:

```
PUT /container_servers/:container_server_id.xml
PUT /container_servers/:container_server_id.json
```

XML Request example

```
curl -i -X PUT -H 'Accept: application/xml' -H 'Content-type:
application/xml' -u user:userpass -d
'<container_server><hypervisor_group_id>6</hypervisor_group_id><hyperviso
r_id>4</hypervisor_id><primary_network_group_id>11</primary_network_group
_id><required_virtual_machine_build>1</required_virtual_machine_build><me
memory>512</memory><label>
TEST</label><cpus>1</cpus><swap_disk_min_iops>100</swap_disk_min_iops><da
ta_store_group_swap_id>13</data_store_group_swap_id><rate_limit>0</rate_l
imit><cpu_shares>1</cpu_shares><enable_autoscale>False</enable_autoscale>
<template_id>742</template_id><primary_disk_min_iops>100</primary_disk_mi
n_iops><initial_root_password>password</initial_root_password><selected_i
p_address_id>None</selected_ip_address_id><data_store_group_primary_id>13
</data_store_group_primary_id><primary_disk_size>5</primary_disk_size><re
quired_ip_address_assignment>1</required_ip_address_assignment><swap_disk
_size>1</swap_disk_size></container_server>' --url http://Cloud
HM.test/container_servers/:container_server_id.xml
```

JSON Request example

```
curl -i -X PUT -H 'Accept: application/json' -H 'Content-type:
application/json' -u user:userpass -d '{"container_server":
{"hypervisor_group_id":"6", "hypervisor_id":"4",
"primary_network_group_id":"11", "required_virtual_machine_build":"1",
"memory":"512", "label":"zaza TEST", "cpus":"1",
"swap_disk_min_iops":"100", "data_store_group_swap_id":"13",
"rate_limit":"0", "cpu_shares":"1", "enable_autoscale":"False",
"template_id":"742", "primary_disk_min_iops":"100",
"initial_root_password":"password", "selected_ip_address_id":"None",
"data_store_group_primary_id":"13", "primary_disk_size":"5",
"required_ip_address_assignment":"1", "swap_disk_size":"1"}}' --url
http://Cloud HM.test/container_servers/:container_server_id.json
```

Where:

hypervisor_group_id - the ID of the compute zone in which the container server will be created

hypervisor_id - the ID of a compute resource where the container server will be built

primary_network_group_id - the ID of the primary network group

required_virtual_machine_build - set 1 to build container server automatically

memory - amount of RAM assigned to the container server

label - name of the container server

cpus - number of CPUs assigned to the container server

swap_disk_min_iops - minimum number of IO operations per second for swap disk (this is a SolidFire related parameter)

data_store_group_swap_id - set the ID of the data store zone to which this swap disk is allocated

rate_limit - set max port speed in Mbps or set 0 to get maximum port speed allowed by your billing plan. If this parameter is omitted or sent without value, the default port speed will be configured for the container server. The default port speed depends on the maximum port speed set in your billing plan and the *Max network interface port speed* parameter at **Control Panel > Settings > Configuration**. The system identifies which of the two values (in the billing plan or in the configuration) is lower and sets it as the default port speed during container server creation.

cpu_shares - for KVM compute resource the CPU priority value is always 100. For XEN, set a custom value. The default value for XEN is 1.

enable_autoscale - true if autoscaling is allowed for this container server. Autoscaling is not currently available for container servers.

template_id - the ID of the template the container server is based on

primary_disk_min_iops - minimum number of IO operations per second for primary disk (this is a SolidFire related parameter)

initial_root_password - the root password for a container server. Optional, if none specified, the system will provide a random password. It can consist of 6-32 characters, letters [A-Za-z], digits [0-9], dash [-] and lower dash [_], and the following special characters: ~ ! @ # \$ * _ - + = ` \ \ { } [] : ; ' , . ? /. You can use both lower- and uppercase letters.

selected_ip_address_id - specify ID of an IP address to assign to this container server; if the parameter *required_ip_address_assignment* was set "1" but this parameter *selected_ip_address_id* is empty - the first available IP address will be assigned to container server automatically

data_store_group_primary_id - set the ID of the data store zone to which this primary disk is allocated

primary_disk_size - set the disk space for this container server

required_ip_address_assignment - set "1" if you want container server to be created with already assigned IP address, otherwise set "0"; IP address can be assigned after container server creation.

swap_disk_size - set swap space

6.11 Edit Container Server Cloud Config

To edit a container server cloud config, use the following request:

```
PUT /container_servers/:container_server_id/cloud_config.xml
PUT /container_servers/:container_server_id/cloud_config.json
```

XML Request example

```
curl -i -X PUT -H 'Accept: application/xml' -H 'Content-type:
application/xml' -u user:userpass -d
'<container_server><cloud_config><user's
cloud_config></cloud_config></container_server>' --url http://Cloud
HM.test/container_servers/:container_server_id/cloud_config.xml
```

JSON Request example

```
curl -i -X PUT -H 'Accept: application/json' -H 'Content-type:
application/json' -u user:userpass -d '{"container_server":
{"cloud_config": "user's cloud_config"}}' --url http://Cloud
HM.test/container_servers/:container_server_id/cloud_config.json
```

Where:

cloud_config - edit the cloud-config file, which enables you to customize different OS elements, such as network configuration, user accounts, etc.. For more information refer to the [Container Server Cloud Config](#) document.

6.12 Change Container Server Owner

Use the following request to reassign a container server to another user:

```
POST /container_servers/:container_server_id/change_owner.xml
POST /container_servers/:container_server_id/change_owner.json
```

XML Request example

```
curl -i -X POST -H 'Accept: application/xml' -H 'Content-type:
application/xml' -u user:password --url 'http://Cloud
HM.test/container_servers/:id/change_owner.xml?user_id=2582&custom_recipe
s_action=move&custom_recipes_action=none&backups_action=move'
```

JSON Request example

```
curl -i -X POST -H 'Accept: application/json' -H 'Content-type:
application/json' -u user:password --url
'http://Cloud
HM.test/container_servers/:id/change_owner.json?user_id=2582&custom_recip
es_action=move&custom_recipes_action=none&backups_action=move'
```

Required parameter:

*user_id** - input ID of a new container server owner

custom_recipes_action - select one of the following options for container server's recipes:

- none - recipes owner will not be changed
- move - recipes owner will be changed
- copy - recipes will be copied to new container servers owner

backups_action - select one of the following options for container server's backups:

- none - backup owner will not be changed
- move - backup owner will be changed

- Instead of container server ID (:container_server_id) you may use container server identifier (:container_server_identifier).
- If container server can not be reassigned to another user, you will get an error message:
"New owner has reached his backup creation limit or doesn't have enough disk space."

6.13 Reset Container Server Root Password

POST /container_servers/:container_server_id/reset_password.xml
POST /container_servers/:container_server_id/reset_password.json

XML Request example

```
curl -i -X POST -u user:userpass --url http://Cloud
HM.test/container_servers/:container_server_id/reset_password.xml -d
'<container_server><initial_root_password>qwaszx321</initial_root_passwor
d><initial_root_password_encryption_key>property321</initial_root_passwor
d_encryption_key></container_server>' -H 'Accept:application/xml' -H
'Content-type:application/xml'
```

JSON Request example

```
curl -i -X POST -u user:userpass --url http://Cloud
HM.test/container_servers/:container_server_id/reset_password.json -d
'{"container_server":{"initial_root_password":"qwaszx123",
"initial_root_password_encryption_key":"property"}}' -H
'Accept:application/json' -H 'Content-type:application/json'
```

Where:

container_server_id * - id of the container server, for which you want to reset password.

initial_root_password - the new root password for a container server. It can consist of 6-32 characters, letters [A-Za-z], digits [0-9], dash [-] and lower dash [_]. You can use both lowercase and uppercase letters.

The following characters are not allowed for Windows-based virtual servers:

- percent sign [%]

- double quotation marks ["]
- brackets [<,>]
- vertical bar [[]]
- caret [^]
- ampersand [&]
- parentheses [(,)]

initial_root_password_encryption_key - specify the password encryption passphrase.

You can also reset a container server password using the Cloud HM 2.3.2 API request:

XML Request example

```
curl -i -X POST -H 'Accept: application/xml' -H 'Content-type: application/xml' -u user:userpass --url http://Cloud HM.test/container_servers/:container_server_id/reset_password
```

JSON Request example

```
curl -i -X POST -H 'Accept: application/json' -H 'Content-type: application/json' -u user:userpass --url http://Cloud HM.test/container_servers/:container_server_id/reset_password*
```

Where:

*container_server_id** - id of the container server, for which you want to reset password.

6.14 Migrate Container Server

You can migrate a container server to another compute resource with the following method:

```
POST /container_servers/:container_server_id/migrate.xml
POST /container_servers/:container_server_id/migrate.json
```

XML Request example

```
curl -i -X POST -H 'Accept: application/xml' -H 'Content-type: application/xml' -u user:userpass -d "<container_server><destination>1</destination><cold_migrate_on_rollback>1</cold_migrate_on_rollback></container_server>" --url http://Cloud HM.test/container_servers/:container_server_id/migrate.xml
```

JSON Request example

```
curl -i -X POST -H 'Accept: application/json' -H 'Content-type: application/json' -u user:userpass -d '{"container_server":{"destination":"1","cold_migrate_on_rollback":"1"}}' --url http://Cloud HM.test/container_servers/:container_server_id/migrate.json
```

Where:

*destination** - the ID of a target compute resource where you migrate a container server

cold_migrate_on_rollback - set to 1 if you wish to switch to a cold migration if hot migration fails, otherwise set 0.

6.15 Set VIP Status for Container Server

To set/remove VIP status for a container server, use the following request:

```
POST /container_servers/:id/set_vip.xml
POST /container_servers/:id/set_vip.json
```

XML Request example

```
curl -i -X POST -H 'Accept: application/xml' -H 'Content-type:
application/xml' -u user:userpass --url http://Cloud
HM.test/container_servers/:id/set_vip.xml -d '<vip>true</vip>'
```

JSON Request example

```
curl -i -X POST -H 'Accept: application/json' -H 'Content-type:
application/json' -u user:userpass --url http://Cloud
HM.test/container_servers/:id/set_vip.json -d '{"vip": "true"}'
```

Where:

vip - whether VIP status is enabled for the server or not. Set this parameter to 'true' to enable and to 'false' to disable the VIP status.

6.16 Delete Container Server

To delete a container server, use the following request:

```
DELETE /container_servers/container_server_id.xml
DELETE /container_servers/container_server_id.json
```

XML Request example

```
curl -i -X DELETE -H 'Accept: application/xml' -H 'Content-type:
application/xml' -u user:userpass --url http://Cloud
HM.test/container_servers/container_server_id.xml
```

JSON Request example

```
curl -i -X DELETE -H 'Accept: application/json' -H 'Content-type:
application/json' -u user:userpass --url http://Cloud
HM.test/container_servers/container_server_id.json
```


6.17 Start up Container Server

When you start up a container server, it might be implicitly cold migrated if the current compute resource does not have sufficient resources. For more information, refer to [Virtual Server Provisioning](#).

To start up a container server:

```
POST /container_servers/:container_server_id/startup.xml
POST /container_servers/:container_server_id/startup.json
```

XML Request example

```
curl -i -X POST -u user:userpass --url http://Cloud
HM.test/container_servers/:container_server_id/startup.xml
```

JSON Request example

```
curl -i -X POST -u user:userpass --url http://Cloud
HM.test/container_servers/:container_server_id/startup.json
```

You can also start up a container server in recovery mode. For this run the following request:

XML Request example

```
curl -i -X POST -u user:userpass -d '<mode>recovery</mode>' --url
http://Cloud HM.test/container_servers/:container_server_id/startup.xml
```

JSON Request example

```
curl -i -X POST -u user:userpass -d '{"mode":"recovery"}' --url
http://Cloud HM.test/container_servers/:container_server_id/startup.json
```

6.18 Segregate Container Server

To segregate a container server (that is, instruct it never to reside on the same compute resource as another container server), use the following method:

```
PUT /container_servers/:container_server_id/segregation.xml
PUT /container_servers/:container_server_id/segregation.json
```

XML Request example

```
curl -i -X PUT -H 'Accept: application/xml' -H 'Content-type:
application/xml' -u user:userpass -d
```

```
'<container_server><strict_container_server_id>123</strict_container_server_id></container_server>' --url http://Cloud
HM.test/container_servers/:container_server_id/segregation.xml
```

JSON Request example

```
curl -i -X PUT -H 'Accept: application/json' -H 'Content-type:
application/json' -u user:userpass -d
'{"container_server":{"strict_container_server_id":"123"}}' --url
http://Cloud
HM.test/container_servers/:container_server_id/segregation.json
```

Where:

strict_container_server_id * - the ID of container server you wish to segregate from the given container server

6.19 Desegregate Container Server

To desegregate a container server (that is, cancel the instruction for it to never reside on the same compute resource as another container server), send an empty identifier using the following method:

```
DELETE /container_servers/:container_server_id/segregation.xml
DELETE /container_servers/:container_server_id/segregation.json
```

XML Request example

```
curl -i -X DELETE -H 'Accept: application/xml' -H 'Content-type:
application/xml' -u user:userpass -d
'<container_server><strict_container_server_id>123</strict_container_server_id></container_server>' --url
http://Cloud
HM.test/container_servers/:container_server_id/segregation.xml
```

JSON Request example

```
curl -i -X DELETE -H 'Accept: application/json' -H 'Content-type:
application/json' -u user:userpass -d
'{"container_server":{"strict_container_server_id":"123"}}' --url
http://Cloud
HM.test/container_servers/:container_server_id/segregation.json
```

Where:

strict_container_server_id * - the ID of container server you wish to desegregate from the given container server

6.20 Reboot Container Server

To reboot a container server:

```
POST /container_servers/:container_server_id/reboot.xml
POST /container_servers/:container_server_id/reboot.json
```

XML Request example

```
curl -i -X POST -u user:userpass --url http://Cloud
HM.test/container_servers/:container_server_id/reboot.xml
```

JSON Request example

```
curl -i -X POST -u user:userpass --url http://Cloud
HM.test/container_servers/:container_server_id/reboot.json
```

An HTTP 201 response is returned on a successful reboot. Unsuccessful reboot responses include HTTP 404 (resource not found – e.g. if the VS isn't online) and HTTP 422 (request cannot be processed – for example, if parameters were incorrect).

6.21 Reboot Container Server in Recovery

To reboot a container server in recovery mode with a temporary login ("root") and password ("recovery"), use the following API calls:

```
POST /container_servers/:container_server_id/reboot.xml
POST /container_servers/:container_server_id/reboot.json
```

XML Request example

```
curl -i -X POST -H 'Accept: application/xml' -H 'Content-type:
application/xml' -u user:userpass -d '<mode>recovery</mode>' --url
http://Cloud HM.test/container_servers/:container_server_id/reboot.xml
```

JSON Request example

```
curl -i -X POST -H 'Accept: application/json' -H 'Content-type:
application/json' -u user:userpass -d '{"mode":"recovery"}' --url
http://Cloud HM.test/container_servers/:container_server_id/reboot.json
```

6.22 Boot Container Server from ISO

To boot container servers that are powered off from an ISO, use the following request:

```
POST /container_servers/:container_server_id/startup.xml
POST /container_servers/:container_server_id/startup.json
```

XML Request example

```
curl -i -X POST -u user:userpass --url http://Cloud
HM.test/container_servers/:container_server_id/startup.xml -d
```

```
'<iso_id>11</iso_id>' -H 'Accept: application/xml' -H 'Content-type: application/xml'
```

JSON Request example

```
curl -i -X POST -u user:userpass --url http://Cloud
HM.test/container_servers/:container_server_id/startup.json -d
'{"iso_id": "11"}' -H 'Accept: application/json' -H 'Content-type: application/json'
```

Where:

container_server_id - the ID of the container server you want to boot

iso_id - the ID of the ISO you want to boot from

6.23 Suspend Container Server

To suspend a container server :

POST /container_servers/:container_server_id/suspend.xml

POST /container_servers/:container_server_id/suspend.json

XML Request example

```
curl -i -X POST -u user:userpass --url http://Cloud
HM.test/container_servers/:container_server_id/suspend.xml
```

JSON Request example

```
curl -i -X POST -u user:userpass --url http://Cloud
HM.test/container_servers/:container_server_id/suspend.json
```

Where:

*container_server_id** - ID of a container server you want to suspend

6.24 Unlock Container Server

To unlock a container server:

POST /container_servers/:container_server_id/unlock.xml

POST /container_servers/:container_server_id/unlock.json

XML Request example

```
curl -i -X POST -u user:userpass --url http://Cloud
HM.test/container_servers/:container_server_id/unlock.xml
```

JSON Request example

```
curl -i -X POST -u user:userpass --url http://Cloud
HM.test/container_servers/:container_server_id/unlock.json
```

6.25 Unsuspend Container Server

To activate a container server again, use the same request as to suspend it:

```
POST /container_servers/:container_server_id/suspend.xml
```

```
POST /container_servers/:container_server_id/suspend.json
```

For details, refer to the [Suspend Container Server](#) section.

6.26 Shut down Container Server

To shut down a container server:

```
POST /container_servers/:container_server_id/shutdown.xml
```

```
POST /container_servers/:container_server_id/shutdown.json
```

XML Request example

```
curl -i -X POST -u user:userpass --url http://Cloud
HM.test/container_servers/:container_server_id/shutdown.xml
```

JSON Request example

```
curl -i -X POST -u user:userpass --url http://Cloud
HM.test/container_servers/:container_server_id/shutdown.json
```

6.27 Stop Container Server

To stop a container server:

```
POST /container_servers/:container_server_id/stop.xml
```

```
POST /container_servers/:container_server_id/stop.json
```

XML Request example

```
curl -i -X POST -u user:userpass --url http://Cloud
HM.test/container_servers/:container_server_id/stop.xml
```

JSON Request example

```
curl -i -X POST -u user:userpass --url http://Cloud
HM.test/container_servers/:container_server_id/stop.json
```

6.28 Open Container Server Console

To open a container server console:

1. Run the following request:

```
GET /container_servers/:container_server_id/console.xml
GET /container_servers/:container_server_id/console.json
```

2. Find and copy the value for the *remote_key* parameter in the response output.
3. Open the following URL in the browser: `http://CloudHM.test/console_remote/[remote_key_parameter_value]`

6.29 Container Server Billing Statistics

You can view the billing statistics for a particular container server using the following request:

```
GET /container_servers/:container_server_id/vm_stats.xml
GET /container_servers/:container_server_id/vm_stats.json
```

XML Output example

```
<?xml version="1.0" encoding="UTF-8"?>
<vm_hourly_stats type="array">
<vm_hourly_stat>
<created_at type="datetime">2011-08-09T12:00:10Z</created_at>
<currency_code>USD</currency_code>
<id type="integer">8248</id>
<stat_time type="datetime">2011-08-09T12:00:00Z</stat_time>
<updated_at type="datetime">2011-08-09T12:00:10Z</updated_at>
<user_id type="integer">1</user_id>
<container_server_id type="integer">44</container_server_id>
<vm_billing_stat_id type="integer">100175</vm_billing_stat_id>
<billing_stats>
<disks type="array">
<disk>
<id type="integer">2933</id>
<costs type="array">
<cost>
<value type="integer">5</value>
<cost type="float">3.0</cost>
<resource_name>disk_size</resource_name>
</cost>
</costs>
<label>Disk#2933</label>
</disk>
<disk>...</disk>
</disks>
<network_interfaces type="array">
<network_interface>
<id type="integer">2688</id>
<costs type="array">
<cost>
<value type="integer">1</value>
<cost type="float">0.0</cost>
<resource_name>ip_addresses</resource_name>
</cost>
<cost>...</cost>
</costs>
```

```

<label>eth0</label>
</network_interface>
</network_interfaces>
<container_servers type="array">
<container_server>
<id type="integer">1701</id>
<costs type="array">
<cost>
<value type="integer">1</value>
<cost type="float">0.0</cost>
<resource_name>cpus</resource_name>
</cost>
</costs>
<label>zaza_CP_3.2 (do not remove)</label>
</container_server>
</container_servers>
</billing_stats>
<total_cost type="float">0.0</total_cost>
<vm_resources_cost type="float">0.0</vm_resources_cost>
<usage_cost type="float">0.0</usage_cost>
</vm_hourly_stat>
<vm_hourly_stat>...</vm_hourly_stat>
</vm_hourly_stats>

```

Where:

created_at - the timestamp in DB when this record was created

currency_code - currency in which this container server is charged within the billing plan

id - the ID of the server hourly statistics. You can add this parameter to the request URL to get a shorter statistics output.

stat_time - the particular hour for which these statistics were generated

updated_at - the date when these statistics were updated

user_id - the ID of container server owner

container_server_id - the ID of the container server

vm_billing_stat_id - billing statistics ID

billing_stats - an array of billing details for the resources used by this container server

When generating billing statistics, Cloud HM takes the last state of the container server during the hour. For example, if a container server was turned on at 6.15 and turned off at 6.59 it will be considered as being off for the whole hour and its resources will be billed according to the OFF prices set in the billing plan. However, the container server's disk and network interface usage can still be billed in case the container server was on during that hour.

disks - an array of disks used by this container server with their billing details:

id - disk ID used in database

costs - an array of disk related resources with their total prices for the period specified in the *stat-time* parameter, where:

value - the amount of resources used (GBs of disk size, Kbs of data read/written, the number of reads/writes)

cost - the total due for the resource

resource_name - the resource in question. This can be *disk_size*, *data_read*, *data_written*, *reads_completed* and *writes_completed*

label - disk name used in UI

network_interfaces - an array of network interfaces used by this container server with their billing statistics:

id - network interface ID

costs- an array of network interface related resources with their total prices for the period specified in the *stat-time* parameter, where:

value - the amount of resources used by this network interface (the number of IPs, the port speed in Mb per second, the data sent and received in Gbs)

cost - the total due for the resource

resource_name- the resource in question. This can be *ip_addresses*, *rate*, *data_received* and *data_sent*

label - network interface name used in Cloud HM

container_server - an array of container server billing details:

id - container server ID

costs- an array of container server resources with their total prices for the period specified in the *stat-time* parameter, where:

value - the amount of resources allocated to this container server. For the *templates* resource, this parameter means a template ID in database.

cost - the total due for this resource

resource_name - the resource in question. This can be *cpu_shares*, *cpus*, *memory*, *cpu_usage* and *template*

label - container server name

total_cost - the total amount of money owed for the container server specified by *id* parameter for a particular hour specified by *stat_time* parameter ($total_cost = vm_resources_cost + usage_cost$)

vm_resources_cost - the amount of money due for the container server resources for the particular hour specified by *stat_time* parameter (memory, disks, templates)

usage_cost - the total due for container server usage for this particular hour specified by *stat_time* parameter (data sent/received, bandwidth, CPU usage)

6.30 Search Container Server by Label

To search container servers by label, run the following request:

```
GET /container_servers.xml?q=label
```

```
GET /container_servers.json?q=label
```

XML Request example


```
curl -i -X GET -H 'Accept: application/xml' -H 'Content-type:
application/xml' -u user:userpass --url http://Cloud
HM.test/container_servers.xml?q=label
```

JSON Request example

```
curl -i -X GET -H 'Accept: application/json' -H 'Content-type:
application/json' -u user:userpass --url http://Cloud
HM.test/container_servers.json?q=label
```

Where you have to specify the label of a container server you are searching for.

6.31 Get Container Server CPU Usage Statistics

To view CPU usage statistics of a container server, run:

```
GET /container_servers/:container_server_id/cpu_usage.xml
```

```
GET /container_servers/:container_server_id/cpu_usage.json
```

XML Request example:

```
curl -i GET -u user:userpass --url http://Cloud
HM.test/container_servers/:container_server_id/cpu_usage.xml
```

XML Request example:

```
curl -i GET -u user:userpass --url http://Cloud
HM.test/container_servers/:container_server_id/cpu_usage.json
```

Where you have to specify the container server ID.

XML Output example

```
<?xml version="1.0" encoding="UTF-8"?>
<cpu_hourly_stats type="array">
  <cpu_hourly_stat>
    <cpu_time type="integer">18</cpu_time>
    <created_at type="datetime">2015-01-06T10:00:18Z</created_at>
    <id type="integer">935848</id>
    <stat_time type="datetime">2015-01-06T10:00:00Z</stat_time>
    <updated_at type="datetime">2015-01-06T10:00:18Z</updated_at>
    <user_id type="integer">1</user_id>
    <container_server_id type="integer">1701</container_server_id>
  </cpu_hourly_stat>
  <cpu_hourly_stat>...</cpu_hourly_stat>
</cpu_hourly_stats>
```

Where:

cpu_time - use the following formula to convert CPU data received in the API output:

$CPU = \text{cpu_time} / 10 / 3600$

Where `cpu_time` is data from API output.

For example: `cpu_time = 2330`, then: $2330/10/3600=0.06$ (6%).

We use "`cpu_time`" * 10 to correct store fractional values.

`created_at` - the timestamp in DB when this record was created

`id` - the statistics ID

`stat_time` - the particular hour for which these statistics were generated

`updated_at` - the time stamp in DB when this record was updated

`user_id` - the ID of the container server owner

`container_server_id` - ID of the container server

6.32 Add/Edit Admin/User Note for Container Server

To edit/make an admin note, use the following request:

```
PUT /container_servers/:container_server_id.xml
```

```
PUT /container_servers/:container_server_id.json
```

or

```
PUT /container_servers/:container_server_id/admin_note.xml
```

```
PUT /container_servers/:container_server_id/admin_note.json
```

Add/Edit Admin Note XML Request example

```
curl -i -X PUT -u user:userpass http://Cloud
HM.test/container_servers/:container_server_id.xml -d
'<container_server><admin_note>agfagwe tiuuytjgh
yuytu</admin_note></container_server>' -H 'Accept:application/xml' -H
'Content-type:application/xml'
```

or

```
curl -i -X PUT -u user:userpass http://Cloud
HM.test/container_servers/:container_server_id/admin_note.xml -d
'<container_server><admin_note>agfagwe tiuuytjgh
yuytu</admin_note></container_server>' -H 'Accept:application/xml' -H
'Content-type:application/xml'
```

Add/Edit Admin Note JSON Request example

```
curl -i -X PUT -u user:userpass http://Cloud
HM.test/container_servers/:container_server_id.json -d
'{"container_server":{"admin_note":"kjfjhjtrtjt"}}' -H
'Accept:application/json' -H 'Content-type:application/json'
```

or

```
curl -i -X PUT -u user:userpass http://Cloud
HM.test/container_servers/:container_server_id/admin_note.json -d
'{"container_server":{"admin_note":"kjfjhjtrtjt"}}' -H
'Accept:application/json' -H 'Content-type:application/json'
```

Where:

admin_note - enter the text of your note.

6.32.1 Add/Edit User Note

To edit/make a user note, use the following request:

```
PUT /container_servers/:container_server_id.xml
```

```
PUT /container_servers/:container_server_id.json
```

or

```
PUT /container_servers/:container_server_id/note.xml
```

```
PUT /container_servers/:container_server_id/note.json
```

XML Request example

```
curl -i -X PUT -u user:userpass http://Cloud
HM.test/container_servers/:container_server_id.xml -d
'<container_server><note>agfagwe tiuuytjgh
yuytu</note></container_server>' -H 'Accept:application/xml' -H 'Content-
type:application/xml'
```

or

```
curl -i -X PUT -u user:userpass http://Cloud
HM.test/container_servers/:container_server_id/note.xml -d
'<container_server><note>agfagwe tiuuytjgh
yuytu</note></container_server>' -H 'Accept:application/xml' -H 'Content-
type:application/xml'
```

Add/Edit User Note JSON Request example

```
curl -i -X PUT -u user:userpass http://Cloud
HM.test/container_servers/:container_server_id.json -d
'{"container_server":{"note":"kjfjhjtrtjt"}}' -H
'Accept:application/json' -H 'Content-type:application/json'
```

or

```
curl -i -X PUT -u user:userpass http://Cloud
HM.test/container_servers/:container_server_id/note.json -d
```

```
'{"container_server":{"note":"kjfjhjtrtjt"}}' -H  
'Accept:application/json' -H 'Content-type:application/json'
```

Where:

note - enter the text of your note.

Returns HTTP 204 response on successful processing, and HTTP 404 when there is no container server with a requested ID, or URL is incorrect.

7 Virtual Servers

Virtual servers in Cloud HM are based on templates and deployed on compute resources. Virtual servers have their own root accounts, so that VS owners can fully control, configure and manage their servers. All CRUD operations are possible for the virtual servers class.

For details how to manage virtual server custom variables, refer to the [Custom Recipe Variables](#) section of this guide.

7.1 Get List of VSs

There are several ways to request the list of VSs. Below you can find the following examples:

- [the list of virtual servers with all related resources](#)
- [the list of virtual servers](#)

To get the list of virtual servers with all related resources, use the following request:

```
GET /virtual_machines.xml
GET /virtual_machines.json
```

XML Request example

```
curl -i -X GET -H 'Accept: application/xml' -H 'Content-type:
application/xml' -u user:password --url http://Cloud
HM.test/virtual_machines.xml
```

JSON Request example

```
curl -i -X GET -H 'Accept: application/json' -H 'Content-type:
application/json' -u user:password --url http://Cloud
HM.test/virtual_machines.json
```

XML Output example

```
<?xml version="1.0" encoding="UTF-8"?>
<virtual_machines>
<virtual_machine>
<add_to_marketplace nil="true"></add_to_marketplace>
<admin_note nil="true"></admin_note>
<allowed_hot_migrate type="boolean">true</allowed_hot_migrate>
<allowed_swap type="boolean">true</allowed_swap>
<booted type="boolean">true</booted>
<built type="boolean">true</built>
<cores_per_socket type="integer">0</cores_per_socket>
<cpu_shares type="integer">1</cpu_shares>
<cpu_sockets nil="true"/>
<cpu_threads nil="true"/>
<cpu_units type="integer">200</cpu_units>
<cpus type="integer">1</cpus>
```

```

<created_at type="datetime">2011-11-01T17:11:58+03:00</created_at>
<customer_network_id nil="true"/>
<deleted_at nil="true"/>
<edge_server_type nil="true"/>
<enable_autoscale type="boolean">>true</enable_autoscale>
<enable_monitis type="boolean">>true</enable_monitis>
<firewall_notrack type="boolean">>false</firewall_notrack>
<hostname>autobackup</hostname>
<hot_add_cpu nil="true"/>
<hot_add_memory nil="true"/>
<hypervisor_id type="integer">2</hypervisor_id>
<id type="integer">000</id>
<identifier>iskngs9dve0hdg</identifier>
<initial_root_password>791791</initial_root_password>
<initial_root_password_encrypted
type="boolean">>false</initial_root_password_encrypted>
<instance_package_id nil="true"/>
<iso_id nil="true"/>
<label>YR_autobackup</label>
<local_remote_access_ip_address>000.000.00.00</local_remote_access_ip_add
ress>
<local_remote_access_port type="integer">0000</local_remote_access_port>
<locked type="boolean">>false</locked>
<memory type="integer">1632</memory>
<min_disk_size type="integer">5</min_disk_size>
<note nil="true"></note>
<operating_system>linux</operating_system>
<operating_system_distro>rhel</operating_system_distro>
<preferred_hvs type="array"/>
<recovery_mode type="boolean">>false</recovery_mode>
<remote_access_password>os3ajolblbuj</remote_access_password>
<service_password nil="true"/>
<state>new</state>
<storage_server_type nil="true"/>
<strict_virtual_machine_id nil="true"></strict_virtual_machine_id>
<suspended type="boolean">>false</suspended>
<template_id type="integer">8</template_id>
<template_label>CentOS 5.6 x86</template_label>
<time_zone>Atlantic Time (Canada)</time_zone>
<updated_at type="datetime">2015-03-04T12:06:21+02:00</updated_at>
<user_id type="integer">5</user_id>
<vip nil="true"></vip>
<xen_id type="integer">12</xen_id>
<ip_addresses type="array">
<ip_address>
<address>000.000.000.000</address>
<broadcast>000.000.000.000</broadcast>
<created_at type="datetime">2011-10-10T12:31:12+03:00</created_at>
<customer_network_id nil="true"/>
<disallowed_primary type="boolean">>false</disallowed_primary>
<gateway>000.000.000.000</gateway>
<hypervisor_id nil="true"/>
<id type="integer">2</id>
<ip_address_pool_id nil="true"/>
<network_address>000.000.000.000</network_address>
<network_id type="integer">1</network_id>
<pxe type="boolean">>false</pxe>
<updated_at type="datetime">2011-11-01T17:39:13+03:00</updated_at>
<user_id nil="true"/>
<free type="boolean">>false</free>
<netmask>000.000.000.000</netmask>
</ip_address>
</ip_addresses>
<monthly_bandwidth_used>0</monthly_bandwidth_used>
<total_disk_size type="integer">6</total_disk_size>
<price_per_hour type="float">0.0</price_per_hour>

```

```

<price_per_hour_powered_off type="float">0.0</price_per_hour_powered_off>
<support_incremental_backups
type="boolean">false</support_incremental_backups>
<cpu_priority type="integer">1</cpu_priority>
<built_from_iso type="boolean">true</built_from_iso>
<acceleration type="boolean">false</acceleration>
<acceleration_status>Inactive</acceleration_status>
<hypervisor_type>kvm</hypervisor_type>
</virtual_machine>
...
<virtual_machine></virtual_machine>
...
</virtual_machine>

```

Where:

acceleration - true if acceleration is enabled for the VS ; otherwise false

acceleration_status - the status of acceleration: active or inactive

add_to_marketplace - empty for VSs; used for edge servers only

admin_note - an optional note of the administrator

allowed_hot_migrate - true if the template, on which the VS is based, supports hot migration; otherwise false

allowed_swap - true if swap disk is allowed (depends on the template the VS is based on); otherwise false

booted - true if the VS is running, otherwise false

built - true if the VS is built, otherwise false

built_from_iso - true if the VS is built from ISO; otherwise false

cores_per_socket - the amount of cores per socket

cpu_shares - CPU priority in percent's

cpu_sockets - the amount of CPU sockets per core. This parameter can be set for KVM compute resources only by those users who have Enable CPU topology permission granted

cpu_threads - the amount of CPU threads per core. This parameter can be set for KVM compute resources only by those users who have Enable CPU topology permission granted

cpu_units - the amount of CPU units per core if the CPU priority is replaced with CPU units in user billing plan.

cpus - the number of allocated CPU cores

created_at - the date in the [YYYY][MM][DD]T[hh][mm][ss]Z format

customer_network_id - ID of a customer network

deleted_at - time when the VS was deleted

edge_server_type - true if this is the edge server

enable_autoscale - true if autoscaling is allowed for this VS

enable_monitis - deprecated attribute; will be removed in upcoming release

firewall_notrack - true if the NOTRACK rule is set in iptables

hostname - the name of your host

hot_add_cpu - true, if the CPU parameter can be changed without rebooting the VS, otherwise false

hot_add_memory - true, if the memory parameter can be changed without rebooting the VS, otherwise false

hypervisor_id - the ID of the compute resource used by this VS

hypervisor_type - the type of the compute resource the VS is built on (for example: *xen*, *kvm*, *vcloud*, *vmware*)

id - the VS ID

identifier - the VS identifier

initial_root_password - the VS root password

initial_root_password_encrypted - true, if the root password is encrypted, otherwise false.

instance_package_id - ID of the instance package

iso_id - the ID of the ISO the VS is based on

label - the VS label

local_remote_access_ip_address - IP address used for remote access

local_remote_access_port - the port ID used for console access

locked - true if the VS is locked; otherwise false

memory - the RAM size allocated to this VS

min_disk_size - the minimum disk size required to build a VS from a specified template

note - an optional reminder for this VS made by a user account

operating_system - operating system used by the VS

operating_system_distro - the distribution of the OS from which this VS is built

preferred_hvs - the array of preferable compute resources based on compute zone that meet some VS configuration settings

recovery_mode - true if recovery mode allowed. Otherwise false

remote_access_password - the password for the remote access

service_password - service account password

state - parameter reserved for future use

storage_server_type - true if this is a storage server

strict_virtual_machine_id - the ID of a virtual server that will never reside on the same compute resource with this VS

suspended - true if VS is suspended, otherwise false

template_id - the ID of the template the VS is based on

template_label - the name of the template from which this VS is built

time_zone - the time zone set for the VS. This parameter is applicable only to Windows KVM and XEN virtual servers.

Currently, the time zone is set at the Compute resource side only. Therefore, users need to set the target time zone inside a Windows VS manually. Setting correct time zone at the Compute resource side helps to

keep correct time inside a VS after starting it if time synchronization is not completed for some reason.

updated_at – the date when the VS was updated in the [YYYY][MM][DD]T[hh][mm][ss]Z format

user_id – the ID of a user assigned to this VS

vip – true if the VS has VIP status (gives migration priority)

xen_id – the VS ID set by the virtualization engine

ip_addresses – an array of IP addresses assigned to this VS and their details:

address – IP address

broadcast – broadcast address

created_at – the date in the [YYYY][MM][DD]T[hh][mm][ss]Z format

customer_network_id – the ID of the customer VLAN the IP address belongs to

disallowed_primary – true if not allowed to be used as primary, otherwise false

gateway – gateway address

hypervisor_id – the ID of a compute resource the IP address is associated with

id – the ID of the IP address

ip_address_pool_id – ID of the IP address pool the IP address is associated with

network_address – the address of the network

network_id – the ID of the network

pxe – true, if this address can be used for cloudbooting a compute resource

updated_at – the date in the [YYYY][MM][DD]T[hh][mm][ss]Z format

user_id – the ID of the user this IP address is assigned to

free – true if free, otherwise false

netmask – netmask for the IP address

monthly_bandwidth_used – VS monthly bandwidth in KB

total_disk_size – the total disk size in GB of all disks assigned to VS

price_per_hour – server's price per hour

price_per_hour_powered_off – price per hour when server is powered off

support_incremental_backups – 1, if virtual server supports incremental backups, and 0 if it does not

cpu_priority – this is a new parameter reserved for further use; currently will have the same value as *cpu_shares*

To get the list of VSs, use the following request:

```
GET /virtual_machines.xml?short
```

GET /virtual_machines.json?short

XML Output example

```
<virtual_machines type="array">
<virtual_machine>
<add_to_marketplace nil="true" />
<admin_note nil="true" />
<allowed_hot_migrate type="boolean">false</allowed_hot_migrate>
<allowed_swap type="boolean">true</allowed_swap>
<booted type="boolean">false</booted>
<built type="boolean">false</built>
<cores_per_socket type="integer">0</cores_per_socket>
<cpu_shares type="integer">100</cpu_shares>
<cpu_sockets nil="true" />
<cpu_threads nil="true" />
<cpu_units type="integer">1000</cpu_units>
<cpus type="integer">1</cpus>
<created_at type="datetime">2015-06-16T19:41:01+00:00</created_at>
<customer_network_id nil="true" />
<deleted_at nil="true" />
<edge_server_type nil="true" />
<enable_autoscale nil="true" />
<enable_monitis type="boolean">false</enable_monitis>
<firewall_notrack type="boolean">false</firewall_notrack>
<hostname>example.Cloud HMDemo.com</hostname>
<hot_add_cpu nil="true" />
<hot_add_memory nil="true" />
<hypervisor_id type="integer">1</hypervisor_id>
<id type="integer">1</id>
<identifier>s34b4zqkkh3gki</identifier>
<initial_root_password>ycW50dZ0ryjj</initial_root_password>
<initial_root_password_encrypted
type="boolean">false</initial_root_password_encrypted>
<instance_package_id nil="true" />
<iso_id nil="true" />
<label>example.Cloud HMDemo.com</label>
<local_remote_access_ip_address nil="true" />
<local_remote_access_port nil="true" />
<locked type="boolean">false</locked>
<memory type="integer">512</memory>
<min_disk_size type="integer">5</min_disk_size>
<note nil="true" />
<operating_system>linux</operating_system>
<operating_system_distro>centos</operating_system_distro>
<preferred_hvs type="array" />
<recovery_mode nil="true" />
<remote_access_password nil="true" />
<service_password nil="true" />
<state>new</state>
<storage_server_type nil="true" />
<strict_virtual_machine_id nil="true" />
<suspended type="boolean">false</suspended>
<template_id type="integer">4</template_id>
<template_label>CentOS 5.3</template_label>
<time_zone nil="true" />
<updated_at type="datetime">2015-06-16T19:41:02+00:00</updated_at>
<user_id type="integer">3</user_id>
<vip nil="true" />
<xen_id nil="true" />
</virtual_machine>
</virtual_machines>
```

7.1.1 Page History:

v.4.3

- added *hypervisor_type* parameter

v.4.2

- added *acceleration* parameter
- added *acceleration_status* parameter
- added *built_from_iso* parameter

v.4.1

- added *time_zone* parameter

7.2 Get VS Details

To get the details of a particular virtual server, use the following request:

```
GET /virtual_machines/:id.xml
GET /virtual_machines/:id.json
```

XML Request example

```
curl -i -X GET -H 'Accept: application/xml' -H 'Content-type:
application/xml' -u user:password --url http://Cloud
HM.test/virtual_machines/:id.xml
```

JSON Request example

```
curl -i -X GET -H 'Accept: application/json' -H 'Content-type:
application/json' -u user:password --url http://Cloud
HM.test/virtual_machines/:id.json
```

XML Output example

```
<?xml version="1.0" encoding="UTF-8"?>
<virtual_machine>
<add_to_marketplace nil="true"></add_to_marketplace>
<admin_note nil="true"></admin_note>
<allowed_hot_migrate type="boolean">true</allowed_hot_migrate>
<allowed_swap type="boolean">true</allowed_swap>
<booted type="boolean">true</booted>
<built type="boolean">true</built>
<cores_per_socket type="integer">0</cores_per_socket>
<cpu_shares type="integer">1</cpu_shares>
<cpu_sockets nil="true"/>
<cpu_threads nil="true"/>
```

```

<cpu_units type="integer">200</cpu_units>
<cpus type="integer">1</cpus>
<created_at type="datetime">2011-11-01T17:11:58+03:00</created_at>
<customer_network_id nil="true"/>
<deleted_at nil="true"/>
<edge_server_type nil="true"/>
<enable_autoscale type="boolean">true</enable_autoscale>
<enable_monitis type="boolean">true</enable_monitis>
<firewall_notrack type="boolean">>false</firewall_notrack>
<hostname>autobackup</hostname>
<hot_add_cpu nil="true"/>
<hot_add_memory nil="true"/>
<hypervisor_id type="integer">2</hypervisor_id>
<id type="integer">000</id>
<identifier>iskngs9dve0hdg</identifier>
<initial_root_password>791791</initial_root_password>
<initial_root_password_encrypted
type="boolean">>false</initial_root_password_encrypted>
<instance_package_id nil="true"/>
<iso_id nil="true"/>
<label>YR_autobackup</label>
<local_remote_access_ip_address>000.000.00.00</local_remote_access_ip_add
ress>
<local_remote_access_port type="integer">0000</local_remote_access_port>
<locked type="boolean">>false</locked>
<memory type="integer">1632</memory>
<min_disk_size type="integer">5</min_disk_size>
<note nil="true"></note>
<operating_system>linux</operating_system>
<operating_system_distro>rhel</operating_system_distro>
<preferred_hvs type="array"/>
<recovery_mode type="boolean">>false</recovery_mode>
<remote_access_password>os3ajolblbuj</remote_access_password>
<service_password nil="true"/>
<state>new</state>
<storage_server_type nil="true"/>
<strict_virtual_machine_id nil="true"></strict_virtual_machine_id>
<suspended type="boolean">>false</suspended>
<template_id type="integer">8</template_id>
<template_label>CentOS 5.6 x86</template_label>
<template_version>1.5</template_version>
<time_zone>Atlantic Time (Canada)</time_zone>
<updated_at type="datetime">2015-03-04T12:06:21+02:00</updated_at>
<user_id type="integer">5</user_id>
<vip nil="true"></vip>
<xen_id type="integer">12</xen_id>
<ip_addresses type="array">
<ip_address>
<address>000.000.000.000</address>
<broadcast>000.000.000.000</broadcast>
<created_at type="datetime">2011-10-10T12:31:12+03:00</created_at>
<customer_network_id nil="true"/>
<disallowed_primary type="boolean">>false</disallowed_primary>
<gateway>000.000.000.000</gateway>
<hypervisor_id nil="true"/>
<id type="integer">2</id>
<ip_address_pool_id nil="true"/>
<network_address>000.000.000.000</network_address>
<network_id type="integer">1</network_id>
<pxe type="boolean">>false</pxe>
<updated_at type="datetime">2011-11-01T17:39:13+03:00</updated_at>
<user_id nil="true"/>
<free type="boolean">>false</free>
<netmask>000.000.000.000</netmask>
</ip_address>
</ip_addresses>

```

```

<monthly_bandwidth_used>0</monthly_bandwidth_used>
<total_disk_size type="integer">6</total_disk_size>
<price_per_hour type="float">0.0</price_per_hour>
<price_per_hour_powered_off type="float">0.0</price_per_hour_powered_off>
<support_incremental_backups
type="boolean">>false</support_incremental_backups>
<cpu_priority type="integer">1</cpu_priority>
<built_from_iso type="boolean">>true</built_from_iso>
<acceleration type="boolean">>false</acceleration>
<acceleration_status>Inactive</acceleration_status>
<hypervisor_type>kvm</hypervisor_type>
</virtual_machine>

```

Where:

acceleration - true if acceleration is enabled for the VS ; otherwise false

acceleration_status - the status of acceleration: active or inactive

add_to_marketplace - empty for VSs; used for edge servers only

admin_note - an optional note of the administrator

allowed_hot_migrate - true if the template, on which the VS is based, supports hot migration; otherwise false

allowed_swap - true if swap disk is allowed (depends on the template the VS is based on); otherwise false

booted - true if the VS is running, otherwise false

built - true if the VS is built, otherwise false

built_from_iso - true if the VS is built from ISO; otherwise false

cores_per_socket - the amount of cores per socket

cpu_shares - CPU priority in percent's

cpu_sockets - the amount of CPU sockets per core. This parameter can be set for KVM compute resources only by those users who have Enable CPU topology permission granted

cpu_threads - the amount of CPU threads per core. This parameter can be set for KVM compute resources only by those users who have Enable CPU topology permission granted

cpu_units - the amount of CPU units per core if the CPU priority is replaced with CPU units in user billing plan.

cpus - the number of allocated CPU cores

created_at - the date in the [YYYY][MM][DD]T[hh][mm][ss]Z format

customer_network_id - ID of a customer network

deleted_at - time when the VS was deleted

edge_server_type - true if this is the edge server

enable_autoscale - true if autoscaling is allowed for this VS

enable_monitis - deprecated attribute; will be removed in upcoming release

firewall_notrack - true if the NOTRACK rule is set in iptables

hostname - the name of your host

hot_add_cpu - true, if the CPU parameter can be changed without rebooting the VS, otherwise false

hot_add_memory - true, if the memory parameter can be changed without rebooting the VS, otherwise false

hypervisor_id - the ID of the compute resource used by this VS

hypervisor_type - the type of the compute resource the VS is built on (for example: *xen*, *kvm*, *vcloud*, *vmware*)

id - the VS ID

identifier - the VS identifier

initial_root_password - the VS root password

initial_root_password_encrypted - true, if the root password is encrypted, otherwise false.

instance_package_id - ID of the instance package

iso_id - the ID of the ISO the VS is based on

label - the VS label

local_remote_access_ip_address - IP address used for remote access

local_remote_access_port - the port ID used for console access

locked - true if the VS is locked; otherwise false

memory - the RAM size allocated to this VS

min_disk_size - the minimum disk size required to build a VS from a specified template

note - an optional reminder for this VS made by a user account

operating_system - operating system used by the VS

operating_system_distro - the distribution of the OS from which this VS is built

preferred_hvs - the array of preferable compute resources based on compute zones zone that meet some VS configuration settings

recovery_mode - true if recovery mode allowed. Otherwise false

remote_access_password - the password for the remote access

service_password - service account password

state - parameter reserved for future use

storage_server_type - true if this is a storage server

strict_virtual_machine_id - the ID of a virtual server that will never reside on the same compute resource with this VS

suspended - true if VS is suspended, otherwise false

template_id - the ID of the template the VS is based on

template_label - the name of the template from which this VS is built

template_version - the version of the template from which this VS is built

time_zone - the time zone set for the VS. This parameter is applicable only to Windows KVM and XEN virtual servers.

Currently, the time zone is set at the Compute resource side only. Therefore, users need to set the target time zone inside a Windows VS manually. Setting correct time zone at the Compute resource side helps to keep correct time inside a VS after starting it if time synchronization is not completed for some reason.

updated_at – the date when the VS was updated in the [YYYY][MM][DD]T[hh][mm][ss]Z format

user_id – the ID of a user assigned to this VS

vip – true if the VS has VIP status (gives migration priority)

xen_id – the VS ID set by the virtualization engine

ip_addresses – an array of IP addresses assigned to this VS and their details:

address – IP address

broadcast – broadcast address

created_at – the date in the [YYYY][MM][DD]T[hh][mm][ss]Z format

customer_network_id – the ID of the customer VLAN the IP address belongs to

disallowed_primary – true if not allowed to be used as primary, otherwise false

gateway – gateway address

hypervisor_id – the ID of a compute resource the IP address is associated with

id – the ID of the IP address

ip_address_pool_id – ID of the IP address pool the IP address is associated with

network_address – the address of the network

network_id – the ID of the network

pxe – true, if this address can be used for cloudbooting a compute resource

updated_at – the date in the [YYYY][MM][DD]T[hh][mm][ss]Z format

user_id – the ID of the user this IP address is assigned to

free – true if free, otherwise false

netmask – netmask for the IP address

monthly_bandwidth_used – VS monthly bandwidth in KB

total_disk_size – the total disk size in GB of all disks assigned to VS

price_per_hour – server's price per hour

price_per_hour_powered_off – price per hour when server is powered off

support_incremental_backups – 1, if virtual server supports incremental backups, and 0 if it does not

cpu_priority – this is a new parameter reserved for further use; currently will have the same value as *cpu_shares*

7.2.1 Page History:

v.5.3

- added *template_version* parameter

v.4.3

- added *hypervisor_type* parameter

v.4.2

- added *acceleration* parameter
- added *acceleration_status* parameter
- added *built_from_iso* parameter

v.4.1

- added *time_zone* parameter

7.3 Get Statuses for all Virtual Servers

Use the following API call to get statuses for all virtual servers:

```
GET /virtual_machines/status.xml
```

```
GET /virtual_machines/status.json
```

XML Request example

```
curl -i -X GET -H 'Accept: application/xml' -H 'Content-type:
application/xml' -u user:userpass --url http://Cloud
HM.test/virtual_machines/status.xml
```

JSON Request example

```
curl -i -X GET -H 'Accept: application/json' -H 'Content-type:
application/json' -u user:userpass --url http://Cloud
HM.test/virtual_machines/status.json
```

XML Output example

```
<virtual_machines type="array">
  <virtual_machine>
    <id type="integer">1</id>
    <identifier>okulsief887rqm</identifier>
    <hostname>vl.test</hostname>
    <template_id type="integer">1</template_id>
    <built type="boolean">>true</built>
    <locked type="boolean">>false</locked>
    <booted type="boolean">>true</booted>
    <operating_system>linux</operating_system>
    <suspended type="boolean">>false</suspended>
    <enable_autoscale type="boolean">>true</enable_autoscale>
    <state>new</state>
  </virtual_machine>
</virtual_machines>
```



```
...
</virtual_machines>
```

Where:

id - virtual server ID
identifier - the VS identifier
hostname - the name of your host
template_id - the ID of the template the VS is based on
built - true if the VS is built, otherwise false
locked - true if the VS is locked; otherwise false
booted - true if the VS is running, otherwise false
operating_system - operating system used by the VS
suspended - true if VS is suspended, otherwise false
enable_autoscale - true if autoscaling is allowed for this VS
state - virtual server state

7.4 Get Virtual Server Status

This parameter has been added in the 3.1 version.

Use the following API call to get status for a particular virtual server:

```
GET /virtual_machines/:virtual_machine_id/status.xml
GET /virtual_machines/:virtual_machine_id/status.json
```

XML Request example

```
curl -i -X GET -H 'Accept: application/xml' -H 'Content-type:
application/xml' -u user:password --url http://Cloud
HM.test/virtual_machines/:virtual_machine_id/status.xml
```

JSON Request example

```
curl -i -X GET -H 'Accept: application/json' -H 'Content-type:
application/json' -u user:password --url http://Cloud
HM.test/virtual_machines/:virtual_machine_id/status.json
```

XML Output example

```

<virtual_machines type="array">
  <virtual_machine>
    <id type="integer">48</id>
    <identifier>b266b5h5et39iy</identifier>
    <hostname>qaaexp</hostname>
    <template_id type="integer">111</template_id>
    <built type="boolean">>true</built>
    <locked type="boolean">>false</locked>
    <booted type="boolean">>true</booted>
    <operating_system>windows</operating_system>
    <suspended type="boolean">>false</suspended>
    <enable_autoscale type="boolean">>false</enable_autoscale>
    <state>new</state>
  </virtual_machine>
</virtual_machines>

```

Where:

id - virtual server ID

identifier - the VS identifier

hostname - the name of your host

template_id - the ID of the template the VS is based on

built - true if the VS is built, otherwise false

locked - true if the VS is locked; otherwise false

booted - true if the VS is running, otherwise false

operating_system - operating system used by the VS

suspended - true if VS is suspended, otherwise false

enable_autoscale - true if autoscaling is allowed for this VS

state - virtual server state

7.5 Add VS

To add a VS, use the following request:

```

POST /virtual_machines.xml
POST /virtual_machines.json

```

This section describes the API request that adds a VS and sets its resources. For information on the API request that adds a VS using instance packages, refer to [Add Instance Package VS](#).

XML Request example

```

curl -i -X POST -H 'Accept: application/xml' -H 'Content-type:
application/xml' -u user:userpass -d
'<virtual_machine><template_id>8</template_id><label>zaza</label><hostnam
e>zaza</hostname><domain>localdomain</domain><hypervisor_group_id>14</hyp
ervisor_group_id><hypervisor_id>1</hypervisor_id><initial_root_password>e
hgebhewtwh</initial_root_password><memory>128</memory><cpus>1</cpus><cpu
_shares>1</cpu_shares><cpu_sockets>12</cpu_sockets><cpu_threads>1</cpu_th
reads><data_store_group_primary_id>18</data_store_group_primary_id><prima
ry_disk_size>5</primary_disk_size><primary_disk_min_iops>100</primary_dis

```

```
k_min_iops><location_group_id>12</location_group_id><licensing_server_id>
38</licensing_server_id><licensing_type>kms</licensing_type><licensing_ke
y>keyexample</licensing_key><data_store_group_swap_id>18</data_store_grou
p_swap_id><swap_disk_size>1</swap_disk_size><swap_disk_min_iops>100</swap
_disk_min_iops><primary_network_group_id>19</primary_network_group_id><se
lected_ip_address>5.1.1.12</selected_ip_address><rate_limit>1</rate_limit
><required_ip_address_assignment>1</required_ip_address_assignment><requi
red_automatic_backup>0</required_automatic_backup><required_virtual_machi
ne_build>1</required_virtual_machine_build><required_virtual_machine_star
tup>1</required_virtual_machine_startup><time_zone>Atlantic Time
(Canada)</time_zone><enable_autoscale>0</enable_autoscale><custom_recipe_
variables><custom_recipe_variable><name>varname</name><value>value</value
><enabled>1</enabled><recipe_joins_attributes
type='array'><recipe_id>11</recipe_id></recipe_joins_attributes></custom_
recipe_variable></custom_recipe_variables><service_addon_ids
type='array'><service_addon_id>273</service_addon_id><service_addon_id>27
4</service_addon_id></service_addon_ids></virtual_machine>' --url
http://Cloud HM.test/virtual_machines.xml
```

JSON Request example

```
curl -i -X POST -H 'Accept: application/json' -H 'Content-type:
application/json' -u user:userpass -d
 '{"virtual_machine":{"template_id":"8", "label":"zaza",
 "hostname":"zaza","domain":"localdomain", "hypervisor_group_id":"14",
 "hypervisor_id":"1","lic_type":"licensing_server_id":"38","licensing_type":"kms","li
censing_key":"keyexample", "initial_root_password":"tyrhsgjh657th",
 "memory":"128", "cpus":"1", "cpu_shares":"1", "cpu_sockets":"12",
 "cpu_threads":"1", "location_group_id":"12",
 "data_store_group_primary_id":"18", "primary_disk_size":"5",
 "primary_disk_min_iops":"100", "data_store_group_swap_id":"18",
 "swap_disk_size":"1", "swap_disk_min_iops":"100",
 "primary_network_group_id":"19", "selected_ip_address":"5.1.1.12",
 "rate_limit":"1", "required_ip_address_assignment":"1",
 "required_automatic_backup":"0", "required_virtual_machine_build":"1",
 "required_virtual_machine_startup":"1", "time_zone": "Atlantic Time
(Canada)",
 "enable_autoscale":"0","recipe_joins_attributes":["11"],"custom_recipe_va
riables":{"custom_recipe_variable":{"name":"varname","value":"var_value",
 "enabled":"1","service_addon_ids": [273, 274]}}}' --url http://Cloud
HM.test/virtual_machines.json
```

The following parameters should be sent:

*memory** - amount of RAM assigned to the VS

*cpus** - number of CPUs assigned to the VS. For KVM compute resources, this parameter sets CPU sockets by default, unless CPU topology is enabled.

*cpu_shares** - required parameter. For KVM compute resource the CPU priority value is always 100. For XEN, set a custom value. The default value for XEN is 1

cpu_units - the amount of CPU units per core if the CPU priority is replaced with CPU units in user billing plan.

cpu_sockets - the amount of CPU sockets. This parameter can be set for KVM compute resources only by those users who have Enable CPU topology permission granted.

cpu_threads - the amount of CPU threads per core. This parameter can be set for KVM compute resources only by those users who have Enable CPU topology permission granted.

*hostname** - set the host name for this VS

domain - specify the domain for this VS. The default value is *localdomain*. This parameter is not applicable for Windows virtual servers.

instance_package_id - ID of the instance package that will be used to build the VS

*label** - user-friendly VS description

location_group_id - set the ID of the location group where the VS should be created

*primary_disk_size** - set the disk space for this VS

*swap_disk_size** - set swap space. There is no swap disk for Windows-based VSs

primary_disk_min_iops - minimum number of IO operations per second for primary disk (this is a SolidFire related parameter)

swap_disk_min_iops - minimum number of IO operations per second for swap disk (this is a SolidFire related parameter)

type_of_format - type of filesystem - ext4. For Linux templates, you can choose ext4 file system instead of the ext3 default one

data_store_group_primary_id - set the ID of the data store zone to which this primary disk is allocated

data_store_group_swap_id - set the ID of the data store zone to which this swap disk is allocated

primary_network_id - the ID of the primary network. Optional parameter that can be used only if it is assigned to the network zone

primary_network_group_id - the ID of the primary network group. Optional parameter

required_automatic_backup - set 1 if you need automatic backups

rate_limit - set max port speed in Mbps or set 0 to get maximum port speed allowed by your billing plan. If this parameter is omitted or sent without value, the default port speed will be configured for the VS. The default port speed depends on the maximum port speed set in your billing plan and the *Max network interface port speed* parameter at **Control Panel > Settings > Configuration**. The system identifies which of the two values (in the billing plan or in the configuration) is lower and sets it as the default port speed during VS creation.

*required_virtual_machine_build ** - set 1 to build VS automatically

required_virtual_machine_startup - set 1 to start up the VS automatically, otherwise set 0 (default state is "1")

time_zone - the time zone set for the VS. This parameter is applicable only to Windows virtual servers.

Currently, the time zone is set at the Compute resource side only. Therefore, users need to set the target time zone inside a Windows VS manually. Setting correct time zone at the Compute resource side helps to keep correct time inside a VS after starting it if time synchronization is not completed for some reason.

*required_ip_address_assignment** - set "1" if you want VS to be created with already assigned IP address, otherwise set "0"; IP address can be assigned after VS creation using this [API request](#)

selected_ip_address - an IP address to assign to this VS; if the above parameter *required_ip_address_assignment* was set "1" but this parameter *selected_ip_address* is empty - the first available IP address will be assigned to VS automatically

admin_note - enter a brief comment for the VS. Optional parameter

note - a brief comment a user can add to a VS

*template_id ** - the ID of a template from which a VS should be built

*licensing_server_id ** - the ID of a template group where the KMS server details are indicated and to which the template belongs (either directly or through the child group). This parameter is for Windows virtual machines with KMS licensing type only

*licensing_type ** - the type of a license: *mak*, *kms* or *user own* license. This parameter is required for Windows virtual machines only

*licensing_key ** - the key of a license, required if you have selected *own* licensing type, and not required for MAK and KMS licensing types

hypervisor_group_id - the ID of the compute zone in which the VS will be created. Optional: if no compute zone is set, the VS will be built in any available compute zone

hypervisor_id - the ID of a compute resource where the VS will be built. Optional: if no compute resource ID is specified, the VS will be built on the compute resource with the least available RAM (but sufficient RAM for the VS)

initial_root_password - the root password for a VS. Optional, if none specified, the system will provide a random password. It can consist of 6-32 characters, letters [A-Za-z], digits [0-9], dash [-] and lower dash [_], and the following special characters: ~ ! @ # \$ * _ - + = ` \ \ { } [] : ; ' , . ? /. You can use both lower- and uppercase letters.

The following characters are not allowed for Windows-based virtual servers:

- percent sign [%]
- double quotation marks ["]
- brackets [<,>]
- vertical bar [|]
- caret [^]
- ampersand [&]
- parentheses [(,)]

initial_root_password_encryption_key - specify the password encryption passphrase

recipe_joins_attributes - an array of recipe ID you want to run on the virtual server provisioning.

custom_variables_attributes - an array of custom variables with the details.

- *enabled* - true, if the variable is enabled, otherwise false
- *id* - variable ID
- *name* - variable name

- *value* - variable value script



service_addon_ids - an array of service add-on IDs, which you want to add to VS

XML Output example

```
<?xml version="1.0" encoding="UTF-8"?>
<virtual_machine>
  <add_to_marketplace nil="true"/>
  <admin_note nil="true"/>
  <allow_resize_without_reboot
type="boolean">false</allow_resize_without_reboot>
  <allowed_hot_migrate type="boolean">true</allowed_hot_migrate>
  <allowed_swap type="boolean">true</allowed_swap>
  <booted type="boolean">false</booted>
  <built type="boolean">false</built>
  <cpu_shares type="integer">1</cpu_shares>
  <cpus type="integer">1</cpus>
  <created_at type="datetime">2013-06-11T16:03:58+03:00</created_at>
  <customer_network_id nil="true"/>
  <deleted_at nil="true"/>
  <domain>localdomain</domain>
  <edge_server_type nil="true"/>
  <enable_autoscale type="boolean">false</enable_autoscale>
  <enable_monitis type="boolean">false</enable_monitis>
  <firewall_notrack type="boolean">false</firewall_notrack>
  <hostname>zaza</hostname>
  <hypervisor_id type="integer">1</hypervisor_id>
  <id type="integer">15</id>
  <identifier>l2lnf62bs44bjf</identifier>
  <initial_root_password>ehgebhewvtwh</initial_root_password>
  <initial_root_password_encrypted
type="boolean">false</initial_root_password_encrypted>
  <label>zaza</label>
  <local_remote_access_ip_address>109.123.91.35</local_remote_access_ip_add
ress>
  <local_remote_access_port nil="true"/>
  <locked type="boolean">true</locked>
  <memory type="integer">128</memory>
  <min_disk_size type="integer">5</min_disk_size>
  <note nil="true"/>
  <operating_system>linux</operating_system>
  <operating_system_distro>ubuntu</operating_system_distro>
  <preferred_hvs type="array"/>
  <recovery_mode nil="true"/>
  <remote_access_password>x9yk3fIMXZBG</remote_access_password>
  <service_password nil="true"/>
  <state>new</state>
  <storage_server_type nil="true"/>
  <strict_virtual_machine_id nil="true"/>
  <suspended type="boolean">false</suspended>
  <template_id type="integer">8</template_id>
  <template_label>Ubuntu 13.04 x64</template_label>
  <time_zone>Atlantic Time (Canada)</time_zone>
  <updated_at type="datetime">2013-06-11T16:03:59+03:00</updated_at>
  <user_id type="integer">1</user_id>
  <vip nil="true"/>
  <xen_id nil="true"/>
  <ip_addresses type="array">
    <ip_address>
      <address>109.123.91.171</address>
      <broadcast>109.123.91.191</broadcast>
      <created_at type="datetime">2013-06-11T14:16:21+03:00</created_at>
      <customer_network_id nil="true"/>
      <disallowed_primary type="boolean">false</disallowed_primary>
      <gateway>109.123.91.129</gateway>
```

```

    <hypervisor_id nil="true"/>
    <id type="integer">386</id>
    <ip_address_pool_id nil="true"/>
    <network_address>109.123.91.128</network_address>
    <network_id type="integer">9</network_id>
    <pxe type="boolean">>false</pxe>
    <updated_at type="datetime">2013-06-11T14:16:21+03:00</updated_at>
    <user_id nil="true"/>
    <free type="boolean">>false</free>
    <netmask>255.255.255.192</netmask>
  </ip_address>
</ip_addresses>
<monthly_bandwidth_used>0</monthly_bandwidth_used>
<total_disk_size type="integer">6</total_disk_size>
</virtual_machine>

```

Where:

add_to_marketplace – true, if the edge server is added to marketplace. The default value is "false". This parameter is for CDN servers only.

admin_note - administrator comment for the VS

allowed_resize_without_reboot – true if resize without reboot is allowed, otherwise false

allowed_hot_migrate – true if hot migration is allowed, otherwise false

allowed_swap – true if swap is allowed, otherwise false

booted - true, if the VS is booted, otherwise false

built - true, if the VS is built, otherwise false

cpu_shares - the percentage of allocated CPU priority resource

cpus - number of CPUs assigned to the VS

created_at – the date when the VS was created in the [YYYY][MM][DD]T[hh][mm][ss] format

updated_at – the date when the VS was updated in the [YYYY][MM][DD]T[hh][mm][ss] format

customer_network_id - ID of the customer network

edge_server_type - type of the CDN edge server. This parameter is for CDN servers only.

enable_autoscale — true if autoscaling is allowed for this VS

enable_monitis - deprecated attribute

firewall_notrack - parameter for adding firewall rules. It true for edge servers only.

hostname - VS hostname

hypervisor_id – the ID of the compute resource, on which the server is deployed

id – the VS ID in Cloud HM CP database

identifier – the VS identifier

initial_root_password — the VS root password

initial_root_password_encrypted - true, if the root password is encrypted, otherwise false

label - user-friendly VS description

local_remote_access_ip_address - IP address for remote connection

local_remote_access_port - port for remote connection

locked - true if the VS is locked; otherwise false

memory - the RAM size allocated to this VS

min_disk_size — the minimum disk size required to build a VS from a specified template

operating_system — operating system used by the VS

operating_system_distro — the distribution of the OS from which this VS is built

recovery_mode - true if recovery mode allowed, otherwise false

remote_access_password — the password for the remote access

service_password - password of a service user

storage_server_type -

strict_virtual_machine_id - the ID of a virtual machine that will never reside on the same compute resource with this VS

suspended - true if VS is suspended, otherwise false

template_id - the ID of the template the VS is based on

template_label - the name of the template from which this VS is built

user_id — the ID of a user assigned to this VS

vip — true if the VS has VIP status (gives migration priority)

xen_id — the VS ID set by the virtualization engine

ip_addresses - an array of IP addresses with the following parameters:

- *address* - IP address
- *broadcast* - broadcast address
- *created_at* - the date when the IP address was created in the [YYYY][MM][DD]T[hh][mm][ss]Z format
- *disallowed_primary* - true if not allowed to be used as primary (for VS build), otherwise false
- *gateway* - gateway address
- *id* - the ID of the IP address
- *ip_address_pool_id* - ID of the IP address pool to the IP address belongs to
- *network_address* - the address of the network
- *network_id* - the ID of the network
- *updated_at* - the date when the IP address was updated in the [YYYY][MM][DD]T[hh][mm][ss]Z format
- *user_id* - the ID of a user associated with this IP address
- *free* - true if free, otherwise false
- *netmask* - netmask for the IP address

monthly_bandwidth_used - VS monthly bandwidth in KB

total_disk_size - total VS disk size

7.5.1 Page history

v.5.5

- added *service_addon_ids* parameter

v.5.4

- added *domain* parameter
- added *selected_ip_address* parameter
- removed *selected_ip_address_id* parameter

v.5.3

- *recipe_ids* replaced with *recipe_joins_attributes* parameter
- *custom_variables* replaced with *custom_variables_attributes* parameter

v.5.2

- *recipe_ids* will be replaced with *recipe_joins_attributes* parameter in Cloud HM 5.3
- *custom_variables* will be replaced with *custom_variables_attributes* parameter in Cloud HM 5.3

v. 4.1

- *location_group_id*
- *time_zone*

v.4.0

- *location_group_id*

v.3.3.2:

- *cpu_sockets*
- *cpu_threads*

v. 3.3:

- *cpu_units*

v. 3.1 :

- *custom_variables*
- *enabled*
- *id*
- *name*
- *value*

7.6 Add Instance Package VS

To add a preconfigured VS, use the following request:

```
POST /virtual_machines.xml
POST /virtual_machines.json
```

This section describes the API request that adds a VS using instance packages. For information on the API request that adds a VS and sets its resources, refer to [Add VS](#).

XML Request example

```
curl -i -X POST -H 'Accept: application/xml' -H 'Content-type:
application/xml' -u user:userpass -d
'<virtual_machine><template_id>8</template_id><label>zaza</label><hostnam
e>zaza</hostname><initial_root_password>ehgebhewvtwh</initial_root_passwo
rd><instance_package_id>3</instance_package_id><location_group_id>12</loc
ation_group_id><licensing_server_id>38</licensing_server_id><licensing_ty
pe>kms</licensing_type><licensing_key>keyexample</licensing_key><required
_ip_address_assignment>1</required_ip_address_assignment><required_automat
ic_backup>0</required_automatic_backup><required_virtual_machine_build>1
</required_virtual_machine_build><required_virtual_machine_startup>1</req
uired_virtual_machine_startup><time_zone>Atlantic Time
(Canada)</time_zone><enable_autoscale>0</enable_autoscale><custom_recipe_
variables><custom_recipe_variable><name>varname</name><value>value</value
><enabled>1</enabled><recipe_ids
type='array'><recipe_id>11</recipe_id></recipe_ids></custom_recipe_variab
le></custom_recipe_variables></virtual_machine>' --url http://Cloud
HM.test/virtual_machines.xml
```

JSON Request example

```
curl -i -X POST -H 'Accept: application/json' -H 'Content-type:
application/json' -u user:userpass -d
'{"virtual_machine":{"template_id":"8", "label":"zaza",
"hostname":"zaza","licensing_server_id":"38","licensing_type":"kms","lice
nsing_key":"keyexample", "initial_root_password":"tyrhsgjh657th",
"instance_package_id":"3", "location_group_id":"12",
"required_ip_address_assignment":"1", "required_automatic_backup":"0",
"required_virtual_machine_build":"1",
"required_virtual_machine_startup":"1", "time_zone": "Atlantic Time
(Canada)",
"enable_autoscale":"0","recipe_ids":["11"],"custom_recipe_variables":{"cu
stom_recipe_variable":{"name":"varname","value":"var_value","enabled":"1"
}}}' --url http://Cloud HM.test/virtual_machines.json
```

The following parameters should be sent:

*hostname** - set the host name for this VS

instance_package_id - ID of the instance package that will be used to build the VS

*label** - user-friendly VS description

location_group_id - set the ID of the location group where the VS should be created

type_of_format - type of filesystem - ext4. For Linux templates, you can choose ext4 file system instead of the ext3 default one

required_automatic_backup - set 1 if you need automatic backups

*required_virtual_machine_build ** - set 1 to build VS automatically

required_virtual_machine_startup - set 1 to start up the VS automatically, otherwise set 0 (default state is "1")

time_zone - the time zone set for the VS. This parameter is applicable only to Windows virtual servers.

Currently, the time zone is set at the Compute resource side only. Therefore, users need to set the target time zone inside a Windows VS manually. Setting correct time zone at the Compute resource side helps to

keep correct time inside a VS after starting it if time synchronization is not completed for some reason.

*required_ip_address_assignment** - set "1" if you want VS to be created with already assigned IP address, otherwise set "0"; IP address can be assigned after VS creation using this [API request](#)

admin_note - enter a brief comment for the VS. Optional parameter

note - a brief comment a user can add to a VS

*template_id ** - the ID of a template from which a VS should be built

*licensing_server_id ** - the ID of a template group where the KMS server details are indicated and to which the template belongs (either directly or through the child group). This parameter is for Windows virtual machines with KMS licensing type only

*licensing_type ** - the type of a license: *mak*, *kms* or *user own* license. This parameter is required for Windows virtual machines only

*licensing_key ** - the key of a license, required if you have selected *own* licensing type, and not required for MAK and KMS licensing types

initial_root_password - the root password for a VS. Optional, if none specified, the system will provide a random password. It can consist of 6-32 characters, letters [A-Za-z], digits [0-9], dash [-] and lower dash [_], and the following special characters: ~ ! @ # \$ * _ - + = ` \ \ { } [] : ; ' , . ? /. You can use both lower- and uppercase letters.

The following characters are not allowed for Windows-based virtual servers:

- percent sign [%]
- double quotation marks ["]
- brackets [<,>]
- vertical bar [|]
- caret [^]
- ampersand [&]
- parentheses [(,)]

initial_root_password_encryption_key - specify the password encryption passphrase

recipe_ids - an array of recipe ID you want to run on the virtual server provisioning

custom_variables - an array of custom variables with the following details:

- *enabled* - true, if the variable is enabled, otherwise false
- *id* - variable ID
- *name* - variable name
- *value* - variable value script

7.7 Add VMware VS

Virtual servers running on VMware compute resources are managed exactly the same as normal virtual servers. The only difference is creation process.

Currently the use of IPv6 is not supported for VMware virtual servers.

To create a VMware virtual server:

```
POST /virtual_machines.xml
POST /virtual_machines.json
```

XML Request example

```
curl -i -X POST -d
'<virtual_machine><template_id>267</template_id><licensing_key></licensin
g_key><label>zaza_ware_xml</label><hostname>zaza</hostname><hypervisor_gr
oup_id>72</hypervisor_group_id><hypervisor_id>29</hypervisor_id><initial_
root_password>qwaszx</initial_root_password><initial_root_password_confir
mation>qwaszx</initial_root_password_confirmation><memory>128</memory><cp
us>1</cpus><cpu_shares>1</cpu_shares><data_store_group_primary_id>84</dat
a_store_group_primary_id><primary_disk_size>25</primary_disk_size><data_s
tore_group_swap_id>84</data_store_group_swap_id><swap_disk_size>0</swap_d
isk_size><customer_network_id>63</customer_network_id><required_automatic
_backup>0</required_automatic_backup><required_virtual_machine_build>1</r
equired_virtual_machine_build><required_virtual_machine_startup>1</requir
ed_virtual_machine_startup><enable_autoscale>0</enable_autoscale><require
d_ip_address_assignment>1</required_ip_address_assignment><custom_recipe_
variables><custom_recipe_variable><name>varname</name><value>value</value
><enabled>1</enabled></custom_recipe_variable</custom_recipe_variables></
virtual_machine>' -u user:userpass http://Cloud
HM.test/virtual_machines.xml -H 'Accept: application/xml' -H 'Content-
type: application/xml'
```

JSON Request example

```
curl -i -X POST -d '{"virtual_machine":{"template_id":"267",
"licensing_key":"","label":"zaza_ware_json","hostname":"zaza",
"hypervisor_group_id":"72","hypervisor_id":"29",
"initial_root_password":"qwaszx",
"initial_root_password_confirmation":"qwaszx","memory":"128",
"cpus":"1","cpu_shares":"1","data_store_group_primary_id":"84",
"primary_disk_size":"25","data_store_group_swap_id":"84",
"swap_disk_size":"0","customer_network_id":"63",
"required_automatic_backup":"0","required_virtual_machine_build":"1",
"required_virtual_machine_startup":"1","enable_autoscale":"0",
"required_ip_address_assignment":"1","custom_recipe_variables":{"custom_r
ecipe_variable":{"name":"varname","value":"var_value","enabled":"1"}}}}'
-u user:userpass http://Cloud HM.test/virtual_machines.json -H 'Accept:
application/json' -H 'Content-type: application/json'
```

Where you have to specify the following parameters:

*memory** - amount of RAM assigned to the VS

*cpus** - number of CPUs assigned to the VS

*cpu_shares** - required parameter. For KVM compute resource the CPU priority value is always 100. For XEN, set a custom value. The default value for XEN is 1

*hostname** - set the host name for this VS

*label** - user-friendly VS description

*primary_disk_size** - set the disk space for this VS

*swap_disk_size** - set swap space. There is no swap disk for Windows-based VSs

type_of_format - type of filesystem - ext4. For Linux templates, you can choose ext4 file system instead of the ext3 default one

data_store_group_primary_id - set the ID of the data store zone to which this primary disk is allocated

data_store_group_swap_id - set the ID of the data store zone to which this swap disk is allocated

primary_network_id - the ID of the primary network. Optional parameter that can be used only if it is assigned to the network zone

primary_network_group_id - the ID of the primary network group. Optional parameter

required_automatic_backup - set 1 if you need automatic backups

rate_limit - set max port speed. Optional parameter: if none set, the system sets port speed to unlimited

*required_virtual_machine_build** - set 1 to build VS automatically

required_virtual_machine_startup - set 1 to start up the VS automatically, otherwise set 0 (default state is "1")

*required_ip_address_assignment** - set "1" if you want IP address to be assigned automatically after creation. Otherwise set "0"

admin_note - enter a brief comment for the VS. Optional parameter

note - a brief comment a user can add to a VS

*template_id** - the ID of a template from which a VS should be built

*licensing_server_id ** - the ID of a template group where the KMS server details are indicated and to which the template belongs (either directly or through the child group). This parameter is for Windows virtual machines with KMS licensing type only

*licensing_type ** - the type of a license: *mak*, *kms* or *user own* license. This parameter is required for Windows virtual machines only

*licensing_key ** - the key of a license, required if you have selected *own* licensing type, and not required for MAK and KMS licensing types

hypervisor_group_id - the ID of the compute zone in which the VS will be created. Optional: if no compute zone is set, the VS will be built in any available compute zone

hypervisor_id - the ID of a compute resource where the VS will be built. Optional: if no compute resource ID is specified, the VS will be built on the compute resource with the least available RAM (but sufficient RAM for the VS)

initial_root_password - the root password for a VS. Optional, if none specified, the system will provide a random password. It can consist of 6-32 characters, letters [A-Za-z], digits [0-9], dash [-] and lower dash [_], and the following special characters: ~ ! @ # \$ * _ - + = ` \ \ { } [] : ; ' , . ? /. You can use both lower- and uppercase letters.

NOTE: It is not possible to set VS password when creating a Windows-based VMware virtual server without running a sysprep.

custom_variables - an array of custom variables with the following details:

- *enabled* - true, if the variable is enabled, otherwise false
- *id* - variable ID
- *name* - variable name
- *value* - variable value script

7.7.1 Page history

In the 3.1 version the following parameters have been added to the Create VMware VS API request:

- *custom_variables*
- *enabled*
- *id*
- *name*
- *value*

7.8 View Encrypted VS Password

If the VS was created with password encryption enabled, you can use the following API call to view the password (the request returns the decrypted password):

```
GET /virtual_machines/:id/with_decrypted_password.xml
```

```
GET /virtual_machines/:id/with_decrypted_password.json
```

XML Request example:

```
curl -X GET -u user:userpass http://Cloud
HM.test/virtual_machines/:id/with_decrypted_password.xml?initial_root_password_encryption_key=encryptionkey
```

JSON Request example:

```
curl -X GET -u user:userpass http://Cloud
HM.test/virtual_machines/:id/with_decrypted_password.json?initial_root_password_encryption_key=encryptionkey
```

Where:

id - the virtual server's ID

7.9 Build or Rebuild VS

To build or rebuild a VS, use the following methods:

```
POST /virtual_machines/:virtual_machine_id/build.xml
POST /virtual_machines/:virtual_machine_id/build.json
```

XML Request example

```
curl -i -X POST -H 'Accept: application/xml' -H 'Content-type:
application/xml' -u user:userpass -d '<?xml version="1.0" encoding="UTF-
8"?><virtual_machine><template_id>1</template_id><required_startup>1</req
uired_startup></virtual_machine>' --url http://Cloud
HM.test/virtual_machines/:virtual_machine_id/build.xml
```

JSON Request example

```
curl -i -X POST -H 'Accept: application/json' -H 'Content-type:
application/json' -u user:userpass -d
'{"virtual_machine":{"template_id":"1","required_startup":"1"}}' --url
http://Cloud HM.test/virtual_machines/:virtual_machine_id/build.json
```

Where:

*template_id** - the ID of a template from which a VS should be built.

required_startup - set to 1 if you wish to start a VS after it is built. Otherwise set to 0.

Instead of virtual server ID (:virtual_machine_id) you may use virtual server identifier (:virtual_machine_identifer).

initial_root_password_encryption_key - specify the password encryption passphrase

for **Windows** templates you should specify the licensing type:

licensing_type - the type of a license: *mak*, *kms* or *user own* license

licensing_key - the key of a license, required if you have selected **OWN** licensing type, and not required for MAK and KMS licensing types

licensing_server_id - the ID of a template group where the **KMS** server details are indicated and to which the template belongs (either directly or through the child group)

7.10 Edit VS

To edit a virtual server, use the following request:

```
PUT /virtual_machines/:id.xml
PUT /virtual_machines/:id.json
```

XML Request example

```
curl -i -X PUT -H 'Accept: application/xml' -H 'Content-type:
application/xml' -u user:userpass -d '<?xml version="1.0" encoding="UTF-
8"?><virtual_machine><label>Test_API_Edit</label><memory>512</memory><cpu
_shares>40</cpu_shares><cpus>4</cpus><allow_migration>1</allow_migration>
<allow_cold_resize>1</allow_cold_resize><primary_disk_min_iops>600</prima
ry_disk_min_iops><time_zone>Atlantic Time
(Canada)</time_zone><swap_disk_min_iops>600</swap_disk_min_iops></virtual
_machine>' --url http://Cloud HM.test/virtual_machines/:id.xml
```

JSON Request example

```
curl -i -X PUT -H 'Accept: application/json' -H 'Content-type:
application/json' -u user:userpass -d
'{"virtual_machine":{"label":"Test_API_Edit","memory":"512","cpu_shares":
"40","cpus":"4","allow_migration":"1","primary_disk_min_iops":"600",
"time_zone": "Atlantic Time
(Canada)","swap_disk_min_iops":"600","allow_cold_resize":"1"}}' --url
http://Cloud HM.test/virtual_machines/:id.json
```

You can edit the following parameters:

label - the VS name

allow_migration - set 1 to migrate a VS to a compute resource with sufficient resources if a compute resource has insufficient space to resize. Otherwise, set 0.

allow_cold_resize - set 1 to switch to cold resize when hot resize failed

time_zone - the time zone set for the VS. This parameter is applicable only to Windows virtual servers.

After you edit the server's time zone, you need to stop and then start up the VS.

Currently, the time zone is set at the Compute resource side only. Therefore, users need to set the target time zone inside a Windows VS manually. Setting correct time zone at the Compute resource side helps to keep correct time inside a VS after starting it if time synchronization is not completed for some reason.

For virtual servers built by selecting resources manually:

memory - the amount of RAM allocated to this VS in Mb

cpus - the number of CPUs of this VS

*cpu_shares** - CPU priority percentage

cpu_units - the amount of CPU units per core if the CPU priority is replaced with CPU units in user billing plan.

For virtual servers built using instance packages:

instance_package_id - ID of the new instance package

You can only choose from those instance packages that offer more disk size than the current instance package.

After you select a new instance package you can use the extra disk size to create a new disk for the VS or make the existing VS disk larger.

If the VS is modified successfully, an HTTP 204 response is returned. If scheduling for changes fails, an HTTP 422 response is returned.

The *primary_disk_min_iops* and *swap_disk_min_iops* parameters are now redundant. Instead, use the *min_iops* parameter for Edit Disk request.

Page History

v.4.1

- *instance_package_id*

v.4.0

- removed *primary_disk_min_iops* and *swap_disk_min_iops* parameters

v.3.3

- *cpu_units*

7.11 Change VS Owner

Use the following request to reassign a VS to another user:

```
POST /virtual_machines/:virtual_machine_id/change_owner.xml
POST /virtual_machines/:virtual_machine_id/change_owner.json
```

XML Request example

```
curl -i -X POST -H 'Accept: application/xml' -H 'Content-type: application/xml' -u user:password --url 'http://Cloud HM.test/virtual_machines/:id/change_owner.xml?user_id=2582&custom_recipes_action=move&custom_recipes_action=none&backups_action=move'
```

JSON Request example

```
curl -i -X POST -H 'Accept: application/json' -H 'Content-type: application/json' -u user:password --url 'http://Cloud HM.test/virtual_machines/:id/change_owner.json?user_id=2582&custom_recipe_s_action=move&custom_recipes_action=none&backups_action=move'
```

Required parameter:

*user_id** - input ID of a new VS owner

custom_recipes_action - select one of the following options for virtual server's recipes:

- none - recipes owner will not be changed
- move - recipes owner will be changed
- copy - recipes will be copied to new virtual servers owner

backups_action - select one of the following options for virtual server's backups:

- none - backup owner will not be changed
- move - backup owner will be changed

- Instead of virtual server ID (:virtual_machine_id) you may use virtual server identifier (:virtual_machine_identifier).
- If VS can not be reassigned to another user, you will get an error message:
"Errors: New owner has reached his backup creation limit or doesn't have enough disk space."

7.12 Reset VS Root Password

POST /virtual_machines/:virtual_machine_id/reset_password.xml
POST /virtual_machines/:virtual_machine_id/reset_password.json

XML Request example

```
curl -i -X POST -u user:userpass --url http://Cloud
HM.test/virtual_machines/:virtual_machine_id/reset_password.xml -d
'<virtual_machine><initial_root_password>qwaszx321</initial_root_password
><initial_root_password_encryption_key>property321</initial_root_password
_encryption_key></virtual_machine>' -H 'Accept:application/xml' -H
'Content-type:application/xml'
```

JSON Request example

```
curl -i -X POST -u user:userpass --url http://Cloud
HM.test/virtual_machines/:virtual_machine_id/reset_password.json -d
'{"virtual_machine":{"initial_root_password":"qwaszx123",
"initial_root_password_encryption_key":"property"}}' -H
'Accept:application/json' -H 'Content-type:application/json'
```

Where:

virtual_machine_id * - id of the VS, for which you want to reset password.

initial_root_password - the new root password for a VS. It can consist of 6-32 characters, letters [A-Za-z], digits [0-9], dash [-] and lower dash [_]. You can use both lowercase and uppercase letters.

The following characters are not allowed for Windows-based virtual servers:

- percent sign [%]
- double quotation marks ["]
- brackets [<,>]
- vertical bar [|]
- caret [^]
- ampersand [&]
- parentheses [(,)]

initial_root_password_encryption_key - specify the password encryption passphrase.

You can also reset a VS password using the Cloud HM 2.3.2 API request:

XML Request example

```
curl -i -X POST -H 'Accept: application/xml' -H 'Content-type: application/xml' -u user:userpass --url http://Cloud HM.test/virtual_machines/:virtual_machine_id/reset_password
```

JSON Request example

```
curl -i -X POST -H 'Accept: application/json' -H 'Content-type: application/json' -u user:userpass --url http://Cloud HM.test/virtual_machines/:virtual_machine_id/reset_password*
```

Where:

*virtual_machine_id** - id of the VS, for which you want to reset password.

7.13 Set SSH Keys

To assign SSH keys of all administrators and a VS owner to a virtual server, use the following request:

```
POST /virtual_machines/:virtual_machine_id/set_ssh_keys.xml
POST /virtual_machines/:virtual_machine_id/set_ssh_keys.json
```

XML Request example

```
curl -X POST -u user:userpass http://Cloud HM.test/virtual_machines/:virtual_machine_id/set_ssh_keys.xml -H 'Accept: application/xml' -H 'Content-type: application/xml'
```

JSON Request example

```
curl -X POST -u user:userpass http://Cloud HM.test/virtual_machines/:virtual_machine_id/set_ssh_keys.json -H 'Accept: application/json' -H 'Content-type: application/json'
```

7.14 Migrate VS

You can migrate a VS to another compute resource with the following method:

```
POST /virtual_machines/:virtual_machine_id/migration.xml
POST /virtual_machines/:virtual_machine_id/migration.json
```

XML Request example

```
curl -i -X POST -H 'Accept: application/xml' -H 'Content-type:
application/xml' -u user:userpass -d
"<virtual_machine><destination>1</destination><cold_migrate_on_rollback>1
</cold_migrate_on_rollback></virtual_machine>" --url http://Cloud
HM.test/virtual_machines/:virtual_machine_id/migration.xml
```

JSON Request example

```
curl -i -X POST -H 'Accept: application/json' -H 'Content-type:
application/json' -u user:userpass -d
'{"virtual_machine":{"destination":"1","cold_migrate_on_rollback":"1"}}'
--url http://Cloud
HM.test/virtual_machines/:virtual_machine_id/migration.json
```

Where:

*destination** - the ID of a target compute resource where you migrate a VS

cold_migrate_on_rollback - set to 1 if you wish to switch to a cold migration if hot migration fails, otherwise set 0.

7.14.1 Page history

- Removed deprecated API requests

```
POST /virtual_machines/:virtual_machine_id/migrate.xml
```

```
POST /virtual_machines/:virtual_machine_id/migrate.json
```

7.15 Migrate VS and Disks

You can migrate a VS and disks between compute resources with local storage or across compute zones with the following method:

```
POST /virtual_machines/:virtual_machine_id/migration.xml
POST /virtual_machines/:virtual_machine_id/migration.json
```

XML Request example

```
curl -i -X POST -H 'Accept: application/xml' -H 'Content-type:
application/xml' -u user:userpass -d
'<virtual_machine><destination><hypervisor_group_id>4</hypervisor_group_i
d><hypervisor_id>1</hypervisor_id><disks_destinations><disk_id>:data_stor
e_id</disk_id></disks_destinations></destination><</virtual_machine>' --
url http://Cloud
HM.test/virtual_machines/:virtual_machine_id/migration.xml
```

JSON Request example

```
curl -i -X POST -H 'Accept: application/json' -H 'Content-type: application/json' -u user:userpass -d '{"virtual_machine": {"destination": {"hypervisor_group_id": "4", "hypervisor_id": "1", "disks_destinations": {"disk_id": "data_store_id"}}}}' --url http://CloudHM.test/virtual_machines/:virtual_machine_id/migration.json
```

Where:

*destination** - the range of target IDs where you migrate a VS:

hypervisor_group_id - the ID of a target compute zone where you migrate a VS

hypervisor_id - the ID of a target compute resource where you migrate a VS

disks_destinations - the IDs of target disks:

disk_id - the ID of the target data store

7.15.1 Page history**v.5.5**

- Removed deprecated API requests

POST /virtual_machines/:virtual_machine_id/migrate.xml

POST /virtual_machines/:virtual_machine_id/migrate.json

7.16 Set VIP Status for VS

To set/remove VIP status for a VS, use the following request:

POST /virtual_machines/:id/set_vip.xml

POST /virtual_machines/:id/set_vip.json

XML Request example

```
curl -i -X POST -H 'Accept: application/xml' -H 'Content-type: application/xml' -u user:userpass --url http://CloudHM.test/virtual_machines/:id/set_vip.xml -d '<vip>true</vip>'
```

JSON Request example

```
curl -i -X POST -H 'Accept: application/json' -H 'Content-type: application/json' -u user:userpass --url http://CloudHM.test/virtual_machines/:id/set_vip.json -d '{"vip": "true"}'
```

Where:

vip - whether VIP status is enabled for the server or not. Set this parameter to 'true' to enable and to 'false' to disable the VIP status.

7.17 Delete VS

```
DELETE /virtual_machines/:id.xml
DELETE /virtual_machines/:id.json
```

To delete a virtual server together with its backups, the user needs to have the *Destroy any backup* or *Destroy own backup* permission enabled. Otherwise, the backups of the VS deleted by the user will remain in the system.

XML Request example

```
curl -i -X DELETE -u user:userpass http://Cloud
HM.test/virtual_machines/:id.xml?convert_last_backup=1&destroy_all_backups=1
```

JSON Request example

```
curl -i -X DELETE -u user:userpass http://Cloud
HM.test/virtual_machines/:id.json?convert_last_backup=1&destroy_all_backups=1
```

Where:

id - the ID of a VS you want to delete

convert_last_backup - set 1 to convert the last VS's backup to template, otherwise set 0

destroy_all_backups - set 1 to destroy all existing backups of this VS, otherwise set 0

7.18 Start up VS

When you start up a VS, it might be implicitly cold migrated if the current compute resource does not have sufficient resources. For more information, refer to [Virtual Server Provisioning](#).

To start up a VS:

```
POST /virtual_machines/:virtual_machine_id/startup.xml
POST /virtual_machines/:virtual_machine_id/startup.json
```

XML Request example

```
curl -i -X POST -u user:userpass --url http://Cloud
HM.test/virtual_machines/:virtual_machine_id/startup.xml
```

JSON Request example

```
curl -i -X POST -u user:userpass --url http://Cloud
HM.test/virtual_machines/:virtual_machine_id/startup.json
```

You can also start up a VS in recovery mode. For this run the following request:

XML Request example

```
curl -i -X POST -u user:userpass -d '<mode>recovery</mode>' --url
http://Cloud HM.test/virtual_machines/:virtual_machine_id/startup.xml
```

JSON Request example

```
curl -i -X POST -u user:userpass -d '{"mode":"recovery"}' --url
http://Cloud HM.test/virtual_machines/:virtual_machine_id/startup.json
```

7.19 Segregate VS

To segregate a VS (that is, instruct it never to reside on the same compute resource as another VS), use the following method:

```
PUT /virtual_machines/:virtual_machine_id/segregation.xml
PUT /virtual_machines/:virtual_machine_id/segregation.json
```

XML Request example

```
curl -i -X PUT -H 'Accept: application/xml' -H 'Content-type:
application/xml' -u user:userpass -d
'<virtual_machine><strict_virtual_machine_id>123</strict_virtual_machine_
id></virtual_machine>' --url http://Cloud
HM.test/virtual_machines/:virtual_machine_id/segregation.xml
```

JSON Request example

```
curl -i -X PUT -H 'Accept: application/json' -H 'Content-type:
application/json' -u user:userpass -d
'{"virtual_machine":{"strict_virtual_machine_id":"123"}}' --url
http://Cloud
HM.test/virtual_machines/:virtual_machine_id/segregation.json
```

Where:

strict_virtual_machine_id * - the ID of virtual server you wish to segregate from the given VS

Page History:

v. 5.3:

- removed deprecated POST /virtual_machines/:virtual_machine_id/strict_vm method

7.20 Desegregate VS

To desegregate a VS (that is, cancel the instruction for it to never reside on the same compute resource as another VS), use the following method:

```
DELETE /virtual_machines/:virtual_machine_id/segregation.xml
```

```
DELETE /virtual_machines/:virtual_machine_id/segregation.json
```

XML Request example

```
curl -i -X DELETE -H 'Accept: application/xml' -H 'Content-type: application/xml' -u user:userpass -d '<virtual_machine><strict_virtual_machine_id>123</strict_virtual_machine_id></virtual_machine>' --url http://Cloud HM.test/virtual_machines/:virtual_machine_id/segregation.xml
```

JSON Request example

```
curl -i -X DELETE -H 'Accept: application/json' -H 'Content-type: application/json' -u user:userpass -d '{"virtual_machine":{"strict_virtual_machine_id":"123"}}' --url http://Cloud HM.test/virtual_machines/:virtual_machine_id/segregation.json
```

Where:

strict_virtual_machine_id * - the ID of virtual server you wish to desegregate from the given VS

Page History:

v. 5.3:

- removed deprecated POST /virtual_machines/:virtual_machine_id/strict_vm method

7.21 Reboot VS

To reboot a VS:

```
POST /virtual_machines/:virtual_machine_id/reboot.xml
```

```
POST /virtual_machines/:virtual_machine_id/reboot.json
```

XML Request example

```
curl -i -X POST -u user:userpass --url http://Cloud HM.test/virtual_machines/:virtual_machine_id/reboot.xml
```

JSON Request example


```
curl -i -X POST -u user:userpass --url http://Cloud
HM.test/virtual_machines/:virtual_machine_id/reboot.json
```

An HTTP 201 response is returned on a successful reboot. Unsuccessful reboot responses include HTTP 404 (resource not found - e.g. if the VS isn't online) and HTTP 422 (request cannot be processed - for example, if parameters were incorrect).

7.22 Enable Accelerator for Virtual Server

To enable acceleration for VS, use the following request:

```
POST /virtual_machines/:id/accelerate.xml
POST /virtual_machines/:id/accelerate.json
```

XML Request example

```
curl -i -X POST http://Cloud HM.test/virtual_machines/:id/accelerate.xml
-u user:userpass -H 'Accept: application/xml' -H 'Content-type:
application/xml'
```

JSON Request example

```
curl -i -X POST http://Cloud
HM.test/virtual_machines/:id/accelerate.json -u user:userpass -H 'Accept:
application/json' -H 'Content-type: application/json'
```

7.23 Disable Accelerator for Virtual Server

To disable acceleration for VS, use the following request:

```
POST /virtual_machines/:id/decelerate.xml
POST /virtual_machines/:id/decelerate.json
```

XML Request example

```
curl -i -X POST http://Cloud HM.test/virtual_machines/:id/decelerate.xml
-u user:userpass -H 'Accept: application/xml' -H 'Content-type:
application/xml'
```

JSON Request example

```
curl -i -X POST http://Cloud
HM.test/virtual_machines/:id/decelerate.json -u user:userpass -H 'Accept:
application/json' -H 'Content-type: application/json'
```

7.24 Purge File(s)

This action is available only for accelerated virtual servers.

In cases when you want to purge one or several files, the system will compare the checksum of the cached file and the new one. The cached file will only be purged if the checksums vary, that is, the files are different. If the checksum of the two files are the same, the cached file will not be purged.

To purge one or several cached files, use the following request:

```
POST /virtual_machines/:virtual_machine_id/purge.xml
```

```
POST /virtual_machines/:virtual_machine_id/purge.json
```

XML Request example

```
curl -i -X POST -u user:userpass -H 'Accept: application/xml' -H
'Content-type: application/xml' --url 'http://Cloud
HM.test/virtual_machines/:virtual_machine_id/purge.xml' -d '<purge_paths
type="array"><purge_path>http://ww-real-
stat3.com/path2</purge_path><purge_path>http://ww-real-
stat4.com/path3</purge_path></purge_paths>'
```

JSON Request example

```
curl -i -X POST -u user:userpass -H 'Accept: application/json' -H
'Content-type: application/json' --url 'http://Cloud
HM.test/virtual_machines/:virtual_machine_id/purge.json' -d
'{"purge_paths":["http://ww-real-stat3.com/path2", "http://ww-real-
stat4.com/path3"]}'
```

Where:

purge_path - the path to the file you want to purge

If you need to purge all content, refer to [Purge All Content](#).

7.25 Purge All Content

This action is available only for accelerated virtual servers.

To purge all cached content, use the following request:

```
POST /virtual_machines/:virtual_machine_id/purge_all.xml
```

```
POST /virtual_machines/:virtual_machine_id/purge_all.json
```

XML Request example

```
curl -i -X POST -u user:userpass -H 'Accept: application/xml' -H
'Content-type: application/xml' --url 'http://Cloud
HM.test/virtual_machines/:virtual_machine_id/purge_all.xml' -d ''
```

JSON Request example

```
curl -i -X POST -u user:userpass -H 'Accept: application/json' -H
'Content-type: application/json' --url 'http://Cloud
HM.test/virtual_machines/:virtual_machine_id/purge_all.json' -d ''
```

If you need to purge only certain files, refer to [Purge File\(s\)](#).

7.26 Reboot VS in Recovery

To reboot a VS in recovery mode with a temporary login ("root") and password ("recovery"), use the following API calls:

```
POST /virtual_machines/:virtual_machine_id/reboot.xml
POST /virtual_machines/:virtual_machine_id/reboot.json
```

XML Request example

```
curl -i -X POST -H 'Accept: application/xml' -H 'Content-type:
application/xml' -u user:userpass -d '<mode>recovery</mode>' --url
http://Cloud HM.test/virtual_machines/:virtual_machine_id/reboot.xml
```

JSON Request example

```
curl -i -X POST -H 'Accept: application/json' -H 'Content-type:
application/json' -u user:userpass -d '{"mode":"recovery"}' --url
http://Cloud HM.test/virtual_machines/:virtual_machine_id/reboot.json
```

7.27 Reboot VS from ISO

To boot a virtual server that is powered on from an ISO, use the following API call:

```
POST /virtual_machines/:virtual_machine_id/reboot.xml
POST /virtual_machines/:virtual_machine_id/reboot.json
```

XML Request example

```
curl -i -X POST -u user:userpass --url http://Cloud
HM.test/virtual_machines/:virtual_machine_id/reboot.xml -d
'<iso_id>11</iso_id>' -H 'Accept: application/xml' -H 'Content-type:
application/xml'
```

JSON Request example

```
curl -i -X POST -u user:userpass --url http://Cloud
HM.test/virtual_machines/:virtual_machine_id/reboot.json -d '{"iso_id":
"11"}' -H 'Accept: application/json' -H 'Content-type: application/json'
```

Where:

virtual_machine_id - the ID of the VS you want to reboot

iso_id - ID of the ISO you want to use

7.28 Boot VS from ISO

To boot virtual servers that are powered off from an ISO, use the following request:

```
POST /virtual_machines/:virtual_machine_id/startup.xml
```

```
POST /virtual_machines/:virtual_machine_id/startup.json
```

XML Request example

```
curl -i -X POST -u user:userpass --url http://Cloud
HM.test/virtual_machines/:virtual_machine_id/startup.xml -d
'<iso_id>11</iso_id>' -H 'Accept: application/xml' -H 'Content-type:
application/xml'
```

JSON Request example

```
curl -i -X POST -u user:userpass --url http://Cloud
HM.test/virtual_machines/:virtual_machine_id/startup.json -d '{"iso_id":
"11" }' -H 'Accept: application/json' -H 'Content-type: application/json'
```

Where:

virtual_machine_id - the ID of the VS you want to boot

iso_id - the ID of the ISO you want to boot from

7.29 Suspend VS

To suspend a VS:

```
POST /virtual_machines/:id/suspend.xml
```

```
POST /virtual_machines/:id/suspend.json
```

XML Request example

```
curl -i -X POST -u user:userpass --url http://Cloud
HM.test/virtual_machines/:virtual_machine_id/suspend.xml
```

JSON Request example

```
curl -i -X POST -u user:userpass --url http://Cloud
HM.test/virtual_machines/:virtual_machine_id/suspend.json
```

Where:

*virtual_machine_id** - ID of a VS you want to suspend

7.30 Unlock VS

To unlock a VS:

```
POST /virtual_machines/:virtual_machine_id/unlock.xml
```

```
POST /virtual_machines/:virtual_machine_id/unlock.json
```

XML Request example

```
curl -i -X POST -u user:userpass --url http://Cloud
HM.test/virtual_machines/:virtual_machine_id/unlock.xml
```

JSON Request example

```
curl -i -X POST -u user:userpass --url http://Cloud
HM.test/virtual_machines/:virtual_machine_id/unlock.json
```

7.31 Unsuspend VS

To activate a VS again, use the same request as to suspend it:

```
POST /virtual_machines/:id/suspend.xml
POST /virtual_machines/:id/suspend.json
```

For details, refer to the [Suspend a VS](#) section.

7.32 Shut down VS

To shut down a VS:

```
POST /virtual_machines/:virtual_machine_id/shutdown.xml
POST /virtual_machines/:virtual_machine_id/shutdown.json
```

XML Request example

```
curl -i -X POST -u user:userpass --url http://Cloud
HM.test/virtual_machines/:virtual_machine_id/shutdown.xml
```

JSON Request example

```
curl -i -X POST -u user:userpass --url http://Cloud
HM.test/virtual_machines/:virtual_machine_id/shutdown.json
```

7.33 Stop VS

To stop a VS:

```
POST /virtual_machines/:virtual_machine_id/stop.xml
POST /virtual_machines/:virtual_machine_id/stop.json
```

XML Request example

```
curl -i -X POST -u user:userpass --url http://Cloud
HM.test/virtual_machines/:virtual_machine_id/stop.xml
```

JSON Request example

```
curl -i -X POST -u user:userpass --url http://Cloud
HM.test/virtual_machines/:virtual_machine_id/stop.json
```

7.34 Open VS Console

To open a VS console:

1. Run the following request:

```
GET /virtual_machines/:virtual_machine_id/console.xml
GET /virtual_machines/:virtual_machine_id/console.json
```

2. Find and copy the value for the *remote_key* parameter in the response output.
3. Open the following URL in the browser: `http://CloudHM.test/console_remote/[remote_key_parameter_value]`

7.35 VS Autoscaling

VS autoscaling allows you to automatically increase the RAM, CPU and disk size of a virtual server. Disk usage autoscaling is applicable for VS primary disk only. VS resources autoscaling is based on the rules you specify. For example, you can set up a rule that will add 1000 MB of memory to a VS if RAM has been above 90% for the last 10 minutes - but add no more than 5000 MB in total in 24 hours.

7.35.1 Enable Autoscaling for VS

To enable autoscaling for a virtual server, use this request:

```
POST /virtual_machines/:virtual_machine_id/autoscale_enable.xml
POST /virtual_machines/:virtual_machine_id/autoscale_enable.json
```

XML Request example

```
curl -i -X POST -u user:password http://Cloud
HM.test/virtual_machines/:virtual_machine_id/autoscale_enable.xml -H
'Accept: application/xml' -H 'Content-type: application/xml'
```

JSON Request example

```
curl -i -X POST -u user:password http://Cloud
HM.test/virtual_machines/:virtual_machine_id/autoscale_enable.json -H
'Accept: application/json' -H 'Content-type: application/json'
```

7.35.2 Get the List of Autoscaling Rules for VS

To get the list of autoscaling rules for a particular VS:

```
GET /virtual_machines/:virtual_machine_id/auto_scaling.xml
GET /virtual_machines/:virtual_machine_id/auto_scaling.json
```

XML Output example

```

<?xml version="1.0" encoding="UTF-8"?>
<auto_scaling_configurations type="array">
<auto_scaling_configuration>
  <adjust_units type="integer">10</adjust_units>
  <created_at type="datetime">2015-02-27T16:11:12+02:00</created_at>
  <for_minutes type="integer">5</for_minutes>
  <id type="integer">6</id>
  <limit_trigger type="integer">10</limit_trigger>
  <resource>cpu</resource>
  <scale_type>up</scale_type>
  <up_to type="integer">50</up_to>
  <updated_at type="datetime">2015-02-27T16:11:12+02:00</updated_at>
  <virtual_machine_id type="integer">3823</virtual_machine_id>
  <above type="integer">10</above>
  <add_units type="integer">10</add_units>
</auto_scaling_configuration>
...
<auto_scaling_configuration></auto_scaling_configuration>
...
</auto_scaling_configuration>

```

Where:

adjust_units - the amount of resource units which the system should add/remove if the rule is met

created_at - the date when the record in DB was created

for_minutes - the time threshold before scaling will be triggered

id - the ID of the rule

limit_trigger - the amount of the resource usage (%). If this value is reached by the VS for the period specified by the *for_minutes* parameter, the system will add/remove the amount of units set by the *adjust_units* parameters

resource - the resource for which the rule is created (memory/cpu/disk)

scale_type - the autoscale option for this rule: up or down

up_to - the amount of resource which cannot be exceeded within 24 hours period

updated_at - the date when the record in DB was updated

virtual_machine_id - the ID of the VS to which this rule applies

above - the amount of the resource usage (%). If this value is reached by the VS during the period specified by the *for_minutes* parameter, the system will add the amount of units set by the *add_units* parameters

add_units - the amount of resource units which the system should add if the rule is met

7.35.3 Create Autoscaling Rule for VS

To create autoscaling rule for a virtual server, use this request:

```
POST /virtual_machines/:virtual_machine_id/auto_scaling.xml
POST /virtual_machines/:virtual_machine_id/auto_scaling.json
```

XML Request example

```
curl -i -X POST -u user:password http://Cloud
HM.test/virtual_machines/:virtual_machine_id/auto_scaling.xml -H 'Accept:
application/xml' -H 'Content-type: application/xml' -d
'<auto_scaling_configurations><up><cpu><enabled>1</enabled><for_minutes>5
</for_minutes><limit_trigger>40</limit_trigger><adjust_units>20</adjust_u
nits><up_to>100</up_to></cpu><memory><enabled>1</enabled><for_minutes>5</
for_minutes><limit_trigger>90</limit_trigger><adjust_units>128</adjust_un
its><up_to>1024</up_to></memory><disk><enabled>1</enabled><for_minutes>5<
/for_minutes><limit_trigger>80</limit_trigger><adjust_units>10</adjust_un
its><up_to>20</up_to></disk></up><down><cpu><enabled>1</enabled><for_minu
tes>5</for_minutes><limit_trigger>100</limit_trigger><adjust_units>10</ad
just_units></cpu><memory><enabled>1</enabled><for_minutes>5</for_minutes>
<limit_trigger>100</limit_trigger><adjust_units>128</adjust_units></memor
y><disk><enabled>1</enabled><for_minutes>5</for_minutes><limit_trigger>80
</limit_trigger><adjust_units>10</adjust_units></disk></down></auto_scali
ng_configurations>'
```

JSON Request example

```
curl -i -X POST -u user:password http://Cloud
HM.test/virtual_machines/:virtual_machine_id/auto_scaling.json -H 'Accept:
application/json' -H 'Content-type: application/json' -d '{
"auto_scaling_configurations":{"up":{"cpu":{"enabled":"1",
"for_minutes":"5", "limit_trigger":"40", "adjust_units":"20",
"up_to":"100"},"memory":{"enabled":"1", "for_minutes":"5",
"limit_trigger":"90", "adjust_units":"128",
"up_to":"1024"},"disk":{"enabled":"1", "for_minutes":"5",
"limit_trigger":"80", "adjust_units":"10",
"up_to":"20"},"down":{"cpu":{"enabled":"1", "for_minutes":"5",
"limit_trigger":"40", "adjust_units":"20"},"memory":{"enabled":"1",
"for_minutes":"5", "limit_trigger":"90", "adjust_units":"128",
"up_to":"1024"},"disk":{"enabled":"1", "for_minutes":"5",
"limit_trigger":"80", "adjust_units":"10"}}}'
```

Where:

*up_to** - the amount of resource which cannot be exceeded within 24 hours period; only for autoscale up rules

*for_minutes** - the time threshold before scaling will be triggered

*trigger_limit** - the amount of the resource usage (%). If this value is reached by the VS for the period specified by the *for_minutes* parameter, the system will add/remove the amount of units set by the *adjust_units* parameters.

*adjust_units** - the amount of resource units which the system should add/remove if the rule is met

*enabled** - set 1 to enable, or 0 to disable

cpu/memory/disk - indicate type of resource for which the autoscaling rule is set (Disk usage autoscaling is applicable for VS primary disk only)

7.35.4 Edit Autoscaling Rule for VS

At present you cannot edit separate elements of autoscaling rule. To change a rule for a VS you have to create a new rule, using the same request as in [Create Autoscaling Rule](#) section.

7.35.5 Delete Autoscaling Rule

To delete autoscaling rules, use this request:

```
DELETE /virtual_machines/:virtual_machine_id/auto_scaling.xml
DELETE /virtual_machines/:virtual_machine_id/auto_scaling.json
```

XML Request example

```
curl -i -X DELETE -u user:userpass --url http://Cloud
HM.test/virtual_machines/:virtual_machine_id/auto_scaling.xml
```

JSON Request example

```
curl -i -X DELETE -u user:userpass --url http://Cloud
HM.test/virtual_machines/:virtual_machine_id/auto_scaling.json
```

This will delete all autoscaling rules set for this VS.

7.35.6 Disable Autoscaling for VS

To disable autoscaling for a virtual server, use this request:

```
POST /virtual_machines/:virtual_machine_id/autoscale_disable.xml
POST /virtual_machines/:virtual_machine_id/autoscale_disable.json
```

XML Request example

```
curl -i -X POST -u user:password http://Cloud
HM.test/virtual_machines/:virtual_machine_id/autoscale_disable.xml -H
'Accept: application/xml' -H 'Content-type: application/xml'
```

JSON Request example

```
curl -i -X POST -u user:password http://Cloud
HM.test/virtual_machines/:virtual_machine_id/autoscale_disable.json -H
'Accept: application/json' -H 'Content-type: application/json'
```

7.36 VS Billing Statistics

You can view the billing statistics for a particular VM using the following request:

```
GET /virtual_machines/:virtual_machine_id/vm_stats.xml
GET /virtual_machines/:virtual_machine_id/vm_stats.json
```

To get a shorter statistics output, add an *id* parameter in the URL:

```
GET /virtual_machines/:virtual_machine_id/vm_stats/:vm_stats_id.xml
GET /virtual_machines/:virtual_machine_id/vm_stats/:vm_stats_id.json
```

Define a shorter period by setting Start and End time in the API call:

GET

```
/virtual_machines/:virtual_machine_id/vm_stats.xml?period[startdate]=YY
YY-MM-DD+hh:mm:ss&period[enddate]=YYYY-MM-
DD+hh:mm:ss&period[use_local_time]=1
```

GET

```
/virtual_machines/:virtual_machine_id/vm_stats.json?period[startdate]=YY
YY-MM-DD+hh:mm:ss&period[enddate]=YYYY-MM-
DD+hh:mm:ss&period[use_local_time]=1
```

XML Output example

```
<?xml version="1.0" encoding="UTF-8"?>
<vm_hourly_stats type="array">
<vm_hourly_stat>
<vm_hourly_stat>
<created_at type="datetime">2011-08-09T12:00:10Z</created_at>
<currency_code>USD</currency_code>
<id type="integer">8248</id>
<stat_time type="datetime">2011-08-09T12:00:00Z</stat_time>
<updated_at type="datetime">2011-08-09T12:00:10Z</updated_at>
<user_id type="integer">1</user_id>
<virtual_machine_id type="integer">44</virtual_machine_id>
<vm_billing_stat_id type="integer">100175</vm_billing_stat_id>
<billing_stats>
<disks type="array">
<disk>
<id type="integer">2933</id>
<costs type="array">
<cost>
<value type="integer">5</value>
<cost type="float">3.0</cost>
<resource_name>disk_size</resource_name>
</cost>
</costs>
<label>Disk#2933</label>
</disk>
<disk>...</disk>
</disks>
<network_interfaces type="array">
<network_interface>
<id type="integer">2688</id>
<costs type="array">
<cost>
<value type="integer">1</value>
<cost type="float">0.0</cost>
<resource_name>ip_addresses</resource_name>
</cost>
<cost>...</cost>
</costs>
<label>eth0</label>
</network_interface>
</network_interfaces>
<service_addons type="array">
<service_addon>
<id type="integer">1</id>
<costs type="array">
<cost>
<value type="integer">1</value>
<cost type="float">0.0</cost>
<resource_name>service_addon</resource_name>
```

```

</cost>
</costs>
<label>zaza_unix</label>
</service_addon>
</service_addons>
<virtual_machines type="array">
<virtual_machine>
<id type="integer">1701</id>
<costs type="array">
<cost>
<value type="integer">1</value>
<cost type="float">0.0</cost>
<resource_name>cpus</resource_name>
</cost>
</costs>
<label>zaza_CP_3.2 (do not remove)</label>
</virtual_machine>
</virtual_machines>
</billing_stats>
<total_cost type="float">0.0</total_cost>
<vm_resources_cost type="float">0.0</vm_resources_cost>
<usage_cost type="float">0.0</usage_cost>
</vm_hourly_stat>
<vm_hourly_stat>...</vm_hourly_stat>
</vm_hourly_stats>

```

Where:

created_at - the timestamp in DB when this record was created

currency_code - currency in which this virtual machine is charged within the billing plan

id - the ID of the server hourly statistics. You can add this parameter to the request URL to get a shorter statistics output.

stat_time - the particular hour for which these statistics were generated

updated_at - the date when these statistics were updated

user_id - the ID of VS owner

virtual_machine_id - the ID of the VS

vm_billing_stat_id - billing statistics ID

billing_stats - an array of billing details for the resources used by this VM

When generating billing statistics, Cloud HM takes the last state of the VS during the hour. For example, if a VS was turned on at 6.15 and turned off at 6.59 it will be considered as being off for the whole hour and its resources will be billed according to the OFF prices set in the billing plan. However, the VS's disk and network interface usage can still be billed in case the VS was on during that hour.

disks - an array of disks used by this VM with their billing details:

id - disk ID used in database

costs- an array of disk related resources with their total prices for the period specified in the *stat-time* parameter, where:

value - the amount of resources used (GBs of disk size, Kbs of data read/written, the number of reads/writes)

cost - the total due for the resource

resource_name - the resource in question. This can be *disk_size*, *data_read*, *data_written*, *reads_completed* and *writes_completed*

label - *disk name used in UI*

network_interfaces - an array of network interfaces used by this VM with their billing statistics:

id - network interface ID

costs- an array of network interface related resources with their total prices for the period specified in the *stat-time* parameter, where:

value - the amount of resources used by this network interface (the number of IPs, the port speed in Mb per second, the data sent and received in KBs)

cost - the total due for the resource

resource_name- the resource in question. This can be *ip_addresses*, *rate*, *data_received* and *data_sent*

label - network interface name used in Cloud HM

service_addons - an array of service add-ons assigned to this VS with their billing details:

id - service add-on ID

costs - an array of service add-on related resources with their total prices for the period specified in the *stat-time* parameter, where:

value - the amount of resources used by this service add-on (at the moment *value* will always be "1")

cost - the total due for the resource

resource_name - the resource in question

label - service add-on name used in Cloud HM

virtual_machine - an array of virtual machine billing details:

id - virtual server ID

costs- an array of VS resources with their total prices for the period specified in the *stat-time* parameter, where:

value - the amount of resources allocated to this VM. For the *templates resource*, this parameter means a template ID in database.

cost - the total due for this resource

resource_name - *the resource in question. This can be cpu_shares, cpus, memory, cpu_usage and template*

label - VS name

total_cost - the total amount of money owed for the VM specified by *id* parameter for a particular hour specified by *stat_time* parameter ($total_cost = vm_resources_cost + usage_cost$)

vm_resources_cost - the amount of money due for the VM resources for the particular hour specified by *stat_time* parameter (memory, disks, templates)

`usage_cost` - the total due for VM usage for this particular hour specified by `stat_time` parameter (data sent/received, bandwidth, CPU usage)

7.37 Search VS by Label

To search virtual servers by label, run the following request:

```
GET /virtual_machines.xml?q=label
GET /virtual_machines.json?q=label
```

XML Request example

```
curl -i -X GET -H 'Accept: application/xml' -H 'Content-type: application/xml' -u user:userpass --url http://Cloud.HM.test/virtual_machines.xml?q=label
```

JSON Request example

```
curl -i -X GET -H 'Accept: application/json' -H 'Content-type: application/json' -u user:userpass --url http://Cloud.HM.test/virtual_machines.json?q=label
```

Where you have to specify the label of a virtual server you are searching for.

7.38 Get VS CPU Usage Statistics

To view CPU usage statistics of a virtual server, run:

```
GET /virtual_machines/:virtual_machine_id/cpu_usage.xml
GET /virtual_machines/:virtual_machine_id/cpu_usage.json
```

Define a shorter period by setting Start and End time in the API call:

```
GET
/virtual_machines/:virtual_machine_id/vm_stats.xml?period[startdate]=YY
YY-MM-DD+hh:mm:ss&period[enddate]=YYYY-MM-
DD+hh:mm:ss&period[use_local_time]=1
GET
/virtual_machines/:virtual_machine_id/vm_stats.json?period[startdate]=YY
YY-MM-DD+hh:mm:ss&period[enddate]=YYYY-MM-
DD+hh:mm:ss&period[use_local_time]=1
```

XML Request example:

```
curl -i GET -u user:userpass --url http://Cloud.HM.test/virtual_machines/:virtual_machine_id/cpu_usage.xml
```

XML Request example:

```
curl -i GET -u user:userpass --url http://Cloud
HM.test/virtual_machines/:virtual_machine_id/cpu_usage.json
```

Where you have to specify the virtual server ID.

XML Output example

```
<?xml version="1.0" encoding="UTF-8"?>
<cpu_hourly_stats type="array">
  <cpu_hourly_stat>
    <cpu_time type="integer">18</cpu_time>
    <created_at type="datetime">2015-01-06T10:00:18Z</created_at>
    <id type="integer">935848</id>
    <stat_time type="datetime">2015-01-06T10:00:00Z</stat_time>
    <updated_at type="datetime">2015-01-06T10:00:18Z</updated_at>
    <user_id type="integer">1</user_id>
    <virtual_machine_id type="integer">1701</virtual_machine_id>
  </cpu_hourly_stat>
  <cpu_hourly_stat>...</cpu_hourly_stat>
</cpu_hourly_stats>
```

Where:

cpu_time - use the following formula to convert CPU data received in the API output:

$$\text{CPU} = \text{cpu_time} / 10 / 3600$$

Where *cpu_time* is data from API output.

For example: *cpu_time* = 2330, then: $2330/10/3600=0.06$ (6%).

We use "*cpu_time*" * 10 to correct store fractional values.

created_at - the timestamp in DB when this record was created

id - the statistics ID

stat_time - the particular hour for which these statistics were generated

updated_at - the time stamp in DB when this record was updated

user_id - the ID of the VS owner

virtual_machine_id - ID of the VS

7.39 Add/Edit Admin/User Note for Virtual Server

To edit/make an admin note, use the following request:

```
PUT /virtual_machines/:virtual_machine_id.xml
```

```
PUT /virtual_machines/:virtual_machine_id.json
```

or

```
PUT /virtual_machines/:virtual_machine_id/admin_note.xml
```

```
PUT /virtual_machines/:virtual_machine_id/admin_note.json
```

Add/Edit Admin Note XML Request example

```
curl -i -X PUT -u user:userpass http://Cloud
HM.test/virtual_machines/:virtual_machine_id.xml -d
'<virtual_machine><admin_note>agfagwe tiuuytjgh
yuytu</admin_note></virtual_machine>' -H 'Accept:application/xml' -H
'Content-type:application/xml'
```

or

```
curl -i -X PUT -u user:userpass http://Cloud
HM.test/virtual_machines/:virtual_machine_id/admin_note.xml -d
'<virtual_machine><admin_note>agfagwe tiuuytjgh
yuytu</admin_note></virtual_machine>' -H 'Accept:application/xml' -H
'Content-type:application/xml'
```

Add/Edit Admin Note JSON Request example

```
curl -i -X PUT -u user:userpass http://Cloud
HM.test/virtual_machines/:virtual_machine_id.json -d
'{"virtual_machine":{"admin_note":"kjfjhjtrtjt"}}' -H
'Accept:application/json' -H 'Content-type:application/json'
```

or

```
curl -i -X PUT -u user:userpass http://Cloud
HM.test/virtual_machines/:virtual_machine_id/admin_note.json -d
'{"virtual_machine":{"admin_note":"kjfjhjtrtjt"}}' -H
'Accept:application/json' -H 'Content-type:application/json'
```

Where:

admin_note - enter the text of your note.

7.39.1 Add/Edit User Note

To edit/make a user note, use the following request:

```
PUT /virtual_machines/:virtual_machine_id.xml
```

```
PUT /virtual_machines/:virtual_machine_id.json
```

or

```
PUT /virtual_machines/:virtual_machine_id/note.xml
```

```
PUT /virtual_machines/:virtual_machine_id/note.json
```

XML Request example

```
curl -i -X PUT -u user:userpass http://Cloud
HM.test/virtual_machines/:virtual_machine_id.xml -d
'<virtual_machine><note>agfagwe tiuuytjgh yuytu</note></virtual_machine>'
-H 'Accept:application/xml' -H 'Content-type:application/xml'
```

or

```
curl -i -X PUT -u user:userpass http://Cloud
HM.test/virtual_machines/:virtual_machine_id/note.xml -d
'<virtual_machine><note>agfagwe tiuuytjgh yuytu</note></virtual_machine>'
-H 'Accept:application/xml' -H 'Content-type:application/xml'
```

Add/Edit User Note JSON Request example

```
curl -i -X PUT -u user:userpass http://Cloud
HM.test/virtual_machines/:virtual_machine_id.json -d
'{"virtual_machine":{"note":"kjfjhjtrtjt"}}' -H 'Accept:application/json'
-H 'Content-type:application/json'
```

or

```
curl -i -X PUT -u user:userpass http://Cloud
HM.test/virtual_machines/:virtual_machine_id/note.json -d
'{"virtual_machine":{"note":"kjfjhjtrtjt"}}' -H 'Accept:application/json'
-H 'Content-type:application/json'
```

Where:

note - enter the text of your note.

Returns HTTP 204 response on successful processing, and HTTP 404 when there is no virtual server with a requested ID, or URL is incorrect.

7.40 Enable Booting from CD for ISO Virtual Server

You can configure whether the VS built from ISO should be booted from the ISO template location (CD emulation) or from the disk where the VS is provisioned. To enable booting from CD for ISO VS, use the following request:

```
POST /virtual_machines/:id/cd_boot/enable.xml
POST /virtual_machines/:id/cd_boot/enable.json
```

If disabled, the VS will be booted from the disk.

XML Request example

```
curl -i -X POST http://Cloud
HM.test/virtual_machines/:id/cd_boot/enable.xml -u user:userpass -H
'Accept: application/xml' -H 'Content-type: application/xml'
```

JSON Request example

```
curl -i -X POST http://Cloud
HM.test/virtual_machines/:id/cd_boot/enable.json -u user:userpass -H
'Accept: application/json' -H 'Content-type: application/json'
```


7.41 Disable Booting from CD for ISO Virtual Server

You can configure whether the VS built from ISO should be booted from the ISO template location (CD emulation) or from the disk where the VS is provisioned. To disable booting from CD for ISO VS, use the following request:

```
POST /virtual_machines/:id/cd_boot/disable.xml
POST /virtual_machines/:id/cd_boot/disable.json
```

In this case the VS will be booted from the disk where the VS is provisioned.

XML Request example

```
curl -i -X POST http://Cloud
HM.test/virtual_machines/:id/cd_boot/disable.xml -u user:userpass -H
'Accept: application/xml' -H 'Content-type: application/xml'
```

JSON Request example

```
curl -i -X POST http://Cloud
HM.test/virtual_machines/:id/cd_boot/disable.json -u user:userpass -H
'Accept: application/json' -H 'Content-type: application/json'
```

7.42 Add Media File to vCloud VS

To add media file to a virtual server, use the following request:

```
PUT /virtual_machines/:id/media_drive.xml
PUT /virtual_machines/:id/media_drive.json
```

XML Request example

```
curl -i -X PUT http://Cloud HM.test/virtual_machines/:id/media_drive.xml
-u user:userpass -d
'<vcloud_media_drive><media>10</media></vcloud_media_drive>' -H 'Accept:
application/xml' -H 'Content-type: application/xml'
```

JSON Request example

```
curl -i -X PUT http://Cloud HM.test/virtual_machines/:id/media_drive.json
-u user:userpass -d '{"vcloud_media_drive": {"media": "10"}}' -H 'Accept:
application/json' -H 'Content-type: application/json'
```

Where:

media - ID of the media file

7.43 Remove Media File from vCloud VS

To remove media file from a virtual server, use the following request:

```
POST /virtual_machines/:id/media_drive/eject.xml
POST /virtual_machines/:id/media_drive/eject.json
```

XML Request example

```
curl -i -X POST http://Cloud
HM.test/virtual_machines/:id/media_drive/eject.xml -u user:userpass -H
'Accept: application/xml' -H 'Content-type: application/xml'
```

JSON Request example

```
curl -i -X POST http://Cloud
HM.test/virtual_machines/:id/media_drive/eject.json -u user:userpass -H
'Accept: application/json' -H 'Content-type: application/json'
```

7.44 View vCloud VS Guest Customization

To view VS guest customization, use the following request:

```
GET /virtual_machines/:id/guest_customization.xml
GET /virtual_machines/:id/guest_customization.json
```

XML Request example:

```
curl -i -X GET -u user:userpass --url http://Cloud
HM.test/virtual_machines/:id/guest_customization.xml -H 'Accept:
application/xml' -H 'Content-type: application/xml'
```

JSON Request example:

```
curl -i -X GET -u user:userpass --url http://Cloud
HM.test/virtual_machines/:id/guest_customization.json -H 'Accept:
application/json' -H 'Content-type: application/json'
```

XML Output example:

```
<vcloud_guest_customization>
  <admin_auto_logon_count type="integer">0</admin_auto_logon_count>
  <admin_auto_logon_enabled
type="boolean">>false</admin_auto_logon_enabled>
  <admin_password nil="true"/>
  <admin_password_auto type="boolean">>true</admin_password_auto>
  <admin_password_enabled type="boolean">>false</admin_password_enabled>
  <change_sid type="boolean">>false</change_sid>
  <computer_name>Centos66-001</computer_name>
  <created_at type="datetime">2016-02-06T16:53:52+02:00</created_at>
  <domain_name nil="true"/>
  <domain_user_name nil="true"/>
  <domain_user_password nil="true"/>
  <enabled type="boolean">>true</enabled>
```

```

<id type="integer">20</id>
<join_domain_enabled type="boolean">>false</join_domain_enabled>
<machine_object_ou nil="true"/>
<reset_password_required type="boolean">>false</reset_password_required>
<script nil="true"/>
<updated_at type="datetime">2016-02-06T16:53:52+02:00</updated_at>
<use_org_settings type="boolean">>false</use_org_settings>
<virtual_machine_id type="integer">20</virtual_machine_id>
</vcloud_guest_customization>

```

Where:

admin_auto_logon_count - the number of times administrator can log in automatically

admin_auto_logon_enabled - true if administrator can log in automatically; otherwise false

admin_password_auto - true if admin password is generated automatically; otherwise false

admin_password_enabled - true if admin password is required; otherwise false

change_sid - this parameter is applicable for Windows VSs only and will run Sysprep to change Windows SID. On Windows NT, VCD uses Sidgen. Running sysprep is a prerequisite for completing domain join

computer_name - VS's computer name

created_at - the date in the [YYYY][MM][DD]T[hh][mm][ss]Z format

enabled - true if guest customization is enabled in vCloud Director; otherwise false

id - ID of the VS guest customization

join_domain_enabled - true if the VS is enabled to join a domain; otherwise false

reset_password_required - true if password reset is required; otherwise false

updated_at - the date in the [YYYY][MM][DD]T[hh][mm][ss]Z format

use_org_settings - true if user org settings are enabled; otherwise false

virtual_machine_id - ID of the VS, which guest customization is viewed

7.45 Get List of Service Add-ons Assigned to VS

To get the list of service add-ons assigned to the VS, use the following request:

```
GET /virtual_machines/:virtual_machine_id/service_addons.xml
```

```
GET /virtual_machines/:virtual_machine_id/service_addons.json
```

XML Request example

```

curl "http://Cloud
HM.test/virtual_machines/:virtual_machine_id/service_addons.xml" -X GET \
-u user:password

```

JSON Request example

```
curl "http://Cloud
HM.test/virtual_machines/:virtual_machine_id/service_addons.json" -X GET \
-u user:password
```

XML Output example

```
<service_addons type="array">
  <service_addon>
    <id type="integer">2</id>
    <label>service_addon2</label>
    <description nil="true"/>
    <compatible_with type="array">
      <compatible_with>unix</compatible_with>
    </compatible_with>
    <user_id type="integer">2</user_id>
    <icon>
      <url nil="true"/>
    </icon>
    <created_at type="dateTime">2016-12-20T16:19:39+00:00</created_at>
    <updated_at type="dateTime">2016-12-20T16:19:39+00:00</updated_at>
  </service_addon>
</service_addons>
```

Where:

id - ID of the service add-on

label - the service add-on title

description - description text added to the service add-on

compatible_with - the OS type, with which this service add-on is compatible

user_id - ID of the user, who created the service add-on

icon - URL with the service add-on icon

created_at - the date when the service add-on was created in the [YYYY][MM][DD]T[hh][mm][ss]Z format

updated_at - the date when the service add-on was updated in the [YYYY][MM][DD]T[hh][mm][ss]Z format

7.46 Assign Service Add-on to VS

To assign service add-on to a VS, use the following request:

```
POST /virtual_machines/:virtual_machine_id/service_addons.xml
```

```
POST /virtual_machines/:virtual_machine_id/service_addons.json
```

XML Request example

```
curl "http://Cloud
HM.test/virtual_machines/:virtual_machine_id/service_addons.xml" -d
'<?xml version="1.0" encoding="UTF-8"?>
```

```
<service_addon_id>4</service_addon_id>
' -X POST \
  -u user:userpass \
  -H "Accept: application/xml" \
  -H "Content-Type: application/xml"
```

JSON Request example

```
curl "http://Cloud
HM.test/virtual_machines/:virtual_machine_id/service_addons.json" -d
'{"service_addon_id":"4"}' -X POST \
-u user:userpass \
-H "Accept: application/json" \
-H "Content-Type: application/json"
```

The following parameters should be sent:

service_addon_id - ID of the service add-on, which you want to assign to the VS

7.47 Unassign Service Add-on from VS

To unassign service add-on from the VS, use the following request:

```
DELETE /virtual_machines/:virtual_machine_id/service_addons/:id.xml
DELETE /virtual_machines/:virtual_machine_id/service_addons/:id.json
```

XML Request example

```
curl "http://Cloud
HM.test/virtual_machines/:virtual_machine_id/service_addons/:id.xml" -d
'' -X DELETE \
  -u user:userpass \
  -H "Accept: application/xml" \
  -H "Content-Type: application/xml"
```

JSON Request example

```
curl "http://Cloud
HM.test/virtual_machines/:virtual_machine_id/service_addons/:id.json" -d
'' -X DELETE \
  -u user:userpass \
  -H "Accept: application/json" \
  -H "Content-Type: application/json"
```

Where:

id - ID of the service add-on, which you want to unassign from the VS

7.48 Use VS as Gateway

To make a VS function as a gateway for a network interface, use the following request:

```
PUT
/virtual_machines/:virtual_machine_id/firewall_rules/update_defaults.xml
```

```
PUT
/virtual_machines/:virtual_machine_id/firewall_rules/update_defaults.json
```

XML Request example

```
curl -i -X PUT -u user:userpass -H 'Accept: application/xml' -H 'Content-type: application/xml' -d
'<network_interfaces><network_interface_id><use_as_gateway>1</use_as_gateway></network_interface_id></network_interfaces>' --url http://Cloud
HM.test/virtual_machines/:virtual_machine_id/firewall_rules/update_defaults.xml
```

JSON Request example

```
curl -i -X PUT -u user:userpass -H 'Accept: application/json' -H
'Content-type: application/json' -d
'{"network_interfaces":{"network_interface_id":{"use_as_gateway":"1"}}}'
--url http://Cloud
HM.test/virtual_machines/:virtual_machine_id/firewall_rules/update_defaults.json
```

Where:

network_interface_id - the network interface for which the VS should function as a gateway

use_as_gateway - set to '1' for the VS to function as a gateway to the network interface, otherwise, set to '0'