Parallels Operations Automation 5.4

PACI RESTful API Programmer's Guide

Revision 2.5 (January 21, 2013)

Parallels IP Holdings GmbH.
Vordergasse 59
CH8200 Schaffhausen
Switzerland
Tel: + 41 526320 411
Fax: + 41 52672 2010
www.parallels.com

Contents

CHAPTER 1

Introduction

This document is a programming and reference guide to the PACI (Parallels Automation for Cloud Infrastructure) RESTful API. The guide is intended for users who would like to write their own programs and scripts to automate management of their PACI resources.

CHAPTER 2

PACI REST API Basics

The PACI RESTful API provides programmatic access to REST resources. Using the API you can obtain information about the resources and perform actions on them. For example, you can obtain a list of the existing servers, start or stop a server, modify server configuration, create a new or delete an existing server, and perform many other management tasks. This chapter provides an overview of the API and describes the basics of using it in your programs.

In This Chapter

Accessing a Resource

A PACI resource is accessed by sending an HTTPS request to a PACI server. When the server receives a request, it processes it accordingly (preforms actions or retrieves data) and sends back a response that can contain the data that you requested, an operation status code, or an error message. All PACI resources are accessed at the following base URL (referred to as *baseURL* in the API reference topics of this guide):

```
https://{ip_address | hostname}:port/paci/version
```

where:

- `ip_address | hostname` is the PACI server IP address or hostname.
- `port` is the port number on which the server is listening for REST requests
- `paci` must be typed exactly as shown
- `version` is the API version number

Note: Ask your service provider for the values listed above (except the "paci" string) and then use them when composing the base URL in your code.

Base URL example

The following base URL sample contains the PACI server hostname (paci-web), port number (4465), the "paci" string, and the version number (v1.0).

```
https://paci-web:4465/paci/v1.0
```

Resource path and parameters

When accessing a particular resource, you append the following values to the base URL:

- The path that identifies the resource
- Optional resource-specific parameters (where available)

Each resource is identified by a path within the base URL. An individual resource may have subresources within its own path, each of which is also identified by a path. For example, the following sample URL references the `ve` resource, which is "server" in general:

```
https://paci-web:4465/paci/v1.0/ve
```

The following sample URL references a specific server identified by its name, which in this instance is my-server-01:

```
https://paci-web:4465/paci/v1.0/ve/my-server-01
```

Resource-specific parameters

Some requests allow to specify additional (usually optional) parameters. For example, when obtaining the list of the operating system templates, you may add a filter to retrieve the templates for a particular operating system (e.g. a particular Linux distribution). Additional parameters are included in the URL after the question mark (?).

HTTP Request Headers

When sending an HTTPS request, the request headers must contain authentication and content type information as described below.

Authentication

PACI REST API uses the basic authentication scheme as defined by RFC 1945 (Hypertext Transfer Protocol – HTTP/1.0). An HTTP request header must contain credential in the form of user name and password separated by a colon and encoded using the Base64 encoding scheme. The following is an example of an authorization header:

```
Authorization: Basic dG9ib3RyYXM6cTE=
```

Content Type

Input data is included in the PACI REST API request in the form of an XML document. The Content-type should be specified as application/xml:

```
Content-type: application/xml
```

CHAPTER 3

HTTP Methods

REST requests are sent to the PACI server as HTTP **POST**, **GET**, **PUT**, and **DELETE** messages. In many cases different methods can be used on the same resource to achieve different goals. For example, you can obtain the information about a server using the **GET** method, modify its configuration using the **PUT** method, and delete it using the **DELETE** method. The resource (identified by a path) will be the same in all three instances, but each request parameters and input/output data will be different. In this guide, each API reference topic describes a particular operation and provides information about the HTTP method that must be used.

Data Input and Output

When sending or receiving data, it is included in a request or received from the PACI server as an XML document. Each request that sends or receives data has a corresponding XML specification that defines the XML document structure. When composing a request, the specification must be followed precisely. The validation of the request is performed on the server side against the XML specification. If a mandatory parameter is missing or an invalid parameter is passed, you will receive an HTTP error. An XML specification for each request and response is described in individual request topics of this guide.

The sample XML document below contains a server information. A document like this is received when a server information is requested or is included in the request when a new server is created (or an existing server is modified).

```
<?xml version="1.0" encoding="UTF-8"?>
<ve>
  <name>my-server-01</name>
  <description>Test server</description>
  <cpu number="2" power="1600"/>
  <ram size="512"/>
  <disk local="true" size="1"/>
  <platform>
    <template-info name="ubuntu-9.10-x86_64"/>
  </platform>
  <backup-schedule name="daily"/>
  <admin login="root" password="q1"/>
</ve>
```

In general, the same XML specification is used in the API for input and output when dealing with the same resource type. Certain parameters, however, cannot be used in the input version and therefore must be omitted. The reference topics in this guide describe the input and output XML specifications separately.

CHAPTER 4

Format and Conventions

The baseURL convention

The string "baseURL" is used in the API reference topics as a shorthand for the base URL at which the PACI resources can be accessed. When composing an HTTP request, the "baseURL" string must be substituted with the base URL specific to your PACI environment. See the **Accessing a Resource** section (p. 7) for more information about the base URL.

API reference topics

Each API reference topic in this guide provides information on how to compose an HTTP request that will perform a particular operation on a particular resource. Each entry contains the following information:

- An HTTP method (POST, GET, PUT, DELETE) used to access the resource. Depending on the type of the operation, different methods may be used to access the same resource. Each operation type has its own topic in this documentation.
- A full path to the resource in the form *baseURL/resource_path*.
- A description of the resource and the operation.
- A list of additional parameters that can be used with the request (where applicable).
- An XML specification of the input and/or output XML documents (included only with the requests that send and/or receive data). Use these specifications to compose an XML input and parse the XML output.
- Samples of HTTP request, HTTP response, and input/output XML documents.

To compose an HTTP request that will perform a particular task on a particular resource, find the corresponding reference topic (each topic name contains the short task description) and follow the provided instructions.

API reference format

Each topic describing an HTTP request has the following sections:

Description

Explains the purpose of the request.

Syntax

Specifies which HTTP method is used with the request (GET, POST, PUT, DELETE) and the resource URL. See **Accessing a Resource** (p. 7) for more info on the resource URL format.

Request Parameters

Describes the XML specification used to specify the request parameters.

Response

For requests that don't output data, describes the HTTP message returned. For requests that return data, describes the XML Schema of the output XML document.

Example

Provides samples of HTTP request, HTTP response, and XML input/output.

Testing code samples and creating your own programs

You can test the samples provided in this guide using a REST client for a Web browser. For example, you can use a simple but effective RESTClient extension for Firefox or any other available REST plug-in and a browser of your choice.

To write your own programs using the API, you will need a development tool that will allow you to make Web requests from the command line or from a program (e.g a program written in C). One of the commonly used tools is cURL. With cURL you can use the API in a script or a C program.

CHAPTER 5

PACI REST API Reference

This chapter contains PACI REST API reference. The requests described here are intended for end users, who want to use the API to manage their PACI resources.

In This Chapter

Server Management

The requests described in this section are used to obtain information and perform operations on individual servers.

List Servers

Description

Use this request to obtain the list of servers owned by the current user.

Syntax

```
GET baseURL/ve
```

Options

You can append an optional subscription ID to the request URL to list only the servers that belong to a specific subscription. The option is appended in the form "?subscription=*id*" where *id* is the numeric subscription ID (see examples below).

Request Parameters

None

Response

Element	Attribute	Description
`ve-list`		Container for server list.
`ve-info`		Container for an individual server info. This element may appear more than once (one for each server).
	`description`	Server description. Type: String
	`state`	A string describing the server state (e.g. CREATED, STOPPED, etc). Type: String
	`name`	Server name. Type: string
	`subscription-id`	ID of the subscription to which this server belongs. If subscription ID is specified in the request URL, this parameter is omitted. Type: int

Examples

The following example lists all servers owned by the user regardless of subscription.

Request

```
GET https://c2u-web:4465/paci/v1.0/ve
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ve-list>
    <ve-info subscription-id="1000001" name="web1" state="CREATED" description="Web
server 1"/>
    <ve-info subscription-id="1000002" name="web2" state="CREATED" description="Web
server 2"/>
</ve-list>
```

The following example lists only the servers for subscription 1000001.

Request

```
GET https://c2u-web:4465/paci/v1.0/ve?subscription=1000001
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ve-list>
    <ve-info name="web1" state="CREATED" description="Web server 1"/>
</ve-list>
```

Start/Stop a Server

Description

Use this call to start or stop a specified server.

Syntax

```
PUT baseURL/ve/{ve-name}/(start | stop)
```

Request Parameters

None

Response

A text message describing the state of the operation. For example, "VE START initiated"

Example

Request

```
PUT  https://c2u-web:4465/paci/v1.0/ve/C2U-Linux-05/start
```

Response

```
VE START initiated
```

Create Server

Description

Use this request to create a new server. To create a server you must choose an operating system template, which will be used to configure the server and install an operating system in it. If the customer account to which the user belongs has more than one subscription, you will have to supply the subscription ID for which to create the server (see **Request Parameters** below). If the server you are trying to create exceeds the subscription server limit, the operation will fail with HTTP error 406: "Subscription limit for VE number exceeded".

Syntax

```
POST baseURL/ve
```

Request Parameters

Element	Attribute	Description
ve		Container for new server information.
name		Server name. Type: string
description		Server description. Type: string
subscription-id		Subscription ID for which to create the new server. This element must be included and populated when the customer account to which the user belongs has more than one subscription. If there's only one subscription, the element may be omitted. Type: int
cpu		Container for CPU information.
	number	Number of CPU cores. Type: int
	power	CPU clock rate in Mhz. Type: int

ram-size		RAM size in megabytes. Type: int
bandwidth		Network bandwidth in kbps. Type: int
no-of-public-ip		Number of public IPv4 addresses. Type: int
no-of-public-ipv6		Number of public IPv6 addresses. Type: int
ve-disk		Container for hard disk information. *At the time of this writing only one hard disk can be added to a server. This will change in the future.*
	local	Local or network hard disk. Only local disks are supported at the time of this writing. Type: boolean
	primary	Specifies whether the disk should be a system disk. This is a reserved parameter. Type: boolean
	size	Hard disk size in gigabytes. Type: int
platform		Container for computing platform information.
template-info		Container or operating system template info.
	name	OS template name. A template contains all the necessary information about the virtualization technology it's using, the operating system, and other information. To obtain the list of the available templates, use the GET /template API call (p. 72). Type: string
os-info		Container for operating system information.
	technology	Virtualization technology: CT -- Virtuozzo Container. VM -- Parallels virtual machine. Type: string
	type	Operating system type.

`backup-schedule`		Container for backup schedule information. This element is optional. If omitted, no backups of the server will be performed.
	`name`	Backup schedule name. Backup schedules are created and maintained by the system administrator. Use the `schedule` interface (p. 75) to obtain the list of the available backup schedules. Type: string
`admin`		Container for server administrator credentials. If this element is omitted, the default login name will be used and the password will be generated automatically.
	`login`	Login. Type: string
	`password`	Password. Type: string

Response

Element	Attribute	Description
`pwd-response`		Container for response data.
`message`		A text message describing the status of the operation.
`password`		Server administrator password. If the password was supplied in the request, this element will contain that password. If the password was not provided, this element will contain an automatically generated one.

Example

Request

```
POST https://c2u-web:4465/paci/v1.0/ve
```

Request body

```
<?xml version="1.0" encoding="UTF-8"?>
<ve>
  <name>Web40</name>
  <description>VE Linux 40</description>
  <subscription-id>1000001</subscription-id>
  <cpu number="2" power="1600"/>
  <ram-size>512</ram-size>
  <bandwidth>100</bandwidth>
  <no-of-public-ip>2</no-of-public-ip>
  <no-of-public-ipv6>2</no-of-public-ipv6>
  <ve-disk local="true" size="3"/>
  <platform>
    <template-info name="centos-6-x86_64"/>
      <os-info technology="CT" type="linux-free"/>
  </platform>
  <backup-schedule name="daily"/>
  <admin login="root" password="152eyyBHO"/>
</ve>
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<pwd-response>
    <message>VE create initiated</message>
    <password>152eyyBHO</password>
</pwd-response>
```

Create Server From Image

Description

Use this request to create a server from an existing image. The `{ve-name}` part of the request URL must contain the new server name (user-defined). The `{image-name}` part must contain the source image name. See also **Server Image Management** (p. 50).

Syntax 1

Use this syntax to create a server from an image when the customer account to which this user belongs has only one subscription.

```
POST baseURL/ve/{ve-name}/from/{image-name}
```

Syntax 2

Use this syntax when the customer account has more than one subscription. The `{subscription-id}` part must contain the subscription ID.

```
POST baseURL/ve/{subscription-id}/{ve-name}/from/{image-name}
```

Request Parameters

None

Response

A text message describing the status of the operation.

Example

The following example creates a server named Web105 from an image named Web-image-101.

Request

```
POST https://c2u-web:4465/paci/v1.0/ve/Web105/from/Web-Image-101
```

The following example creates a server named Web106 from an image named Web-image-101 for subscription 1000001.

Request

```
POST https://c2u-web:4465/paci/v1.0/ve/1000001/Web105/from/Web-Image-101
```

Clone Server

Description

Use this request to create a clone of an existing server. The `{ve-name}` part of the URL must contain the name of the source server. The `{new-server-name}` part must contain the new server name (user-defined). The IP addresses, gateway, and DNS settings for the new server will be set automatically. The rest of the server configuration will be inherited from the source server. The administrator password for the new server will be automatically generated and returned to the caller. If the server you are trying to create exceeds the subscription server limit, the operation will fail with HTTP error 406: "Subscription limit for VE number exceeded".

Syntax

```
POST baseURL/ve/{ve-name}/clone-to/{new-server-name}
```

Request Parameters

None

Response

Element	Attribute	Description
`pwd-response`		Container for response data.
`message`		A text message describing the status of the operation.
`password`		An automatically generated administrator password for the new server.

Example

Request

```
POST https://c2u-web:4465/paci/v1.0/ve/Web40/clone-to/My_new_server
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<pwd-response>
    <message>Clone VE initiated</message>
    <password>621zwPqs</password>
</pwd-response>
```

Modify Server Configuration

Description

Use this request to modify the configuration of an existing server. Please note that this request cannot be used to modify the server backup schedule. See **Backup and Restore** (p. 41) for the information on how to change the backup schedule assignment.

Syntax

```
PUT baseURL/ve/{ve-name}
```

Request Parameters

Element	Attribute	Description
`reconfigure-ve`		Container for new configuration information.
`description`		Server description. Type: string
`change-cpu`		Container for CPU information.
	`number`	Number of CPU cores. Type: int
	`power`	CPU clock rate in Mhz. Type: int
`ram-size`		RAM size in megabytes. Type: int
`bandwidth`		Network bandwidth in kbps. Type: int
`reconfigure-ipv4`		Container for public IPv4 address changes. **Note:** You can either add or remove IP addresses in one call, but not both.
`add-ip`		Container for IPv4 addresses to add to the server configuration.
	`number`	Specifies the number of addresses to add to the server configuration from the pool. Type: int

drop-ip		Container for IPv4 addresses to remove from the server configuration.
	ip	Whitespace-separated list of CIDR-compliant (ip/mask) IPv4 addresses to remove from the server configuration. Type: string
reconfigure-ipv6		Container for public IPv6 address changes. **Note:** You can either add or remove IP addresses in one call, but not both.
add-ip		Container for IPv6 addresses to add to the server configuration.
	number	Specifies the number of addresses to add to the server configuration from the pool. Type: int
drop-ip		Container for IPv6 addresses to remove from the server configuration.
	ip	Whitespace-separated list of CIDR-compliant (ip/mask) IPv6 addresses to remove from the server configuration. Type: string

primary-disk-size		Hard disk size, in gigabytes. Type: int

Response

A string describing the state of the operation.

Examples

Request

```
PUT https://c2u-web:4465/paci/v1.0/ve/Web40
```

Request body 1

Modifying the CPU, RAM, and disk size.

```
<?xml version="1.0" encoding="UTF-8"?>
<reconfigure-ve>
  <change-cpu number="2" power="900"/>
  <ram-size>1280</ram-size>
  <primary-disk-size>15</primary-disk-size>
</reconfigure-ve>
```

Request body 2

Adding two IPv4 addresses.

```
<?xml version="1.0" encoding="UTF-8"?>
<reconfigure-ve>
  <reconfigure-ipv4>
    <add-ip number="2"/>
  </reconfigure-ipv4>
</reconfigure-ve>
```

Request body 3

Removing the specified IPv4 addresses.

```
<?xml version="1.0" encoding="UTF-8"?>
<reconfigure-ve>
  <reconfigure-ipv4>
    <drop-ip ip="10.29.184.100 10.29.184.102"/>
  </reconfigure-ipv4>
</reconfigure-ve>
```

Reset Server Administrator Password

Description

Use this request to reset the server administrator password. The new password will be automatically generated.

Syntax

```
POST baseURL/ve/{ve-name}/reset-password
```

Request Parameters

None

Response

Element	Attribute	Description
`pwd-response`		Container for response data.
`message`		A text message describing the status of the operation.
`password`		New password.

Example

Request

```
POST https://c2u-web:4465/paci/v1.0/ve/Web40/reset-password
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<pwd-response>
    <message>Password reset initiated</message>
    <password>151rhpJEB</password>
</pwd-response>
```

Obtain Server Information

Description

Use this request to obtain the information about the specified server. The `{ve-name}` part of the request URL must contain the server name.

Syntax

```
GET baseURL/ve/{ve-name}
```

Request Parameters

None

Response

Element	Attribute	Description
`ve`		Container for server information.
`id`		Server database ID (used internally). Type: int
`uuid`		Server UUID. Type: string
`hnId`		ID of the hardware node on which this server resides. Type: string
`customer-id`		ID of the customer account to which the server belongs. Type: int
`name`		Server name. Type: string
`description`		Server description. Type: string
`subscription-id`		ID of the subscription to which the server belongs. Type: int
`cpu`		Container for CPU information.
	`number`	Number of CPU cores. Type: int

	`power`	CPU clock speed in megahertz. Type: int
`ram-size`		RAM size in megabytes. Type: int
`bandwidth`		Network bandwidth in kbps. Type: int
`ve-disk`		Container for hard disk information. *At the time of this writing, a server can have only one hard disk. This will change in the future.*
	`id`	Disk database ID (used internally) Type: int
	`local`	Specified whether this is a local or a network disk. *Only local disks are supported at the the time of this writing.* Type: boolean
	`size`	Disk size in gigabytes. Type: int
	`created`	Disk status. Type: boolean Possible values: `true` -- the disk creation process completed. `false` -- the disk is being created and is not available for use.
`platform`		Container for operating system template and related information.
`template-info`		Container for OS template information.
	`name`	Template name. Type: string
	`vendorId`	Template vendor ID. Type: string
	`c2uId`	PACI template ID (used internally). Type: string
`os-info`		Container for operating system information. This information is a part of the OS template info.
	`type`	OS type (Linux. Windows, etc.) Type: string

	`technology`	Virtualization technology used. Type: string Possible values: `CT` - Virtuozzo Container, `VM` - Parallels virtual machine.
`network`		Container for network information.
	`private_ip`	Private IP address and mask. Type: string
`public-ip`		Container for public IPv4 address info. This element may appear more than once (one for each IP address).
	`id`	Database ID (used internally). Type: int
	`address`	IP address and mask. Type: string
	`gateway`	Gateway IP address. Type: string
	`chunk-ref`	Used internally. Type: int
`public-ipv6`		Container for public IPv6 address info. This element may appear more than once (one for each IPv6 address).
	`id`	Database ID (used internally). Type: int
	`address`	IPv6 address and mask. Type: string
	`gateway`	Gateway IPv6 address. Type: string
`backup-schedule`		Container for backup schedule info. A backup schedule is created and configured by the system administrator. The user can select one of the existing backup schedules when a server is created.
	`name`	Backup schedule name. Type: string

`state`		A string describing the server state or transition. Type: string
`primary-disk-id`		Used internally. Type: int
`template-id`		Used internally. Type: int
`admin`		Container for the server administrator credentials.
	`login`	Server administrator login name. Type: string
	`password`	Password. Type: string
`last-operation-rc`		The returned code of the last operation. Type: int
`app-info`		Container for installed application template information. This element may appear more than once (one for each installed template).
	`app-template`	Application template name. Type: string
	`for-os`	The name of the OS template for which this application is designed. Type: string
	`c2u-version`	The template PACI version. Type: string
	`installed-at`	Template location. Type: string
	`installed-ok`	Specifies whether the template was installed properly. Type: boolean
	`uninstalled-at`	The template previous location. Type: string
	`uninstalled-ok`	Specifies whether the template uninstall was completed successfully. Type: boolean
	`app-template-id`	Application template ID (used internally). Type: int

`load-balancer`		Name of the load balancer attached to this server. If no load balancer is used, the element will be absent. Type: string
`steady-state`		A string describing the server steady state. This parameter normally never contains transition states that change quickly. When monitoring transitions, look at the `state` parameter above. Type: string

Example

Request

```
GET https://c2u-web:4465/paci/v1.0/ve/Web40
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ve>
    <id>4</id>
    <uuid>6733915.132fd360949._7ffe</uuid>
    <hnId>5</hnId>
    <customer-id>1</customer-id>
    <name>server1</name>
    <description></description>
    <subscription-id>1</subscription-id>
    <cpu power="1000" number="1"/>
    <ram-size>128</ram-size>
    <bandwidth>100</bandwidth>
    <ve-disk created="true" size="1" local="true" id="0"/>
    <platform>
        <template-info c2uId="1" vendorId="5" name="centos-5-x86"/>
        <os-info technology="CT" type="linux-free"/>
    </platform>
    <network private-ip="10.40.119.201/8">
        <public-ip chunk-ref="1" gateway="10.30.0.1" address="10.30.119.201/16"
id="3"/>
    </network>
    <backup-schedule name="hourly"/>
    <state>STOPPED</state>
    <primary-disk-id>0</primary-disk-id>
    <template-id>1</template-id>
    <admin password="[hidden]" login="root"/>
    <steady-state>STOPPED</steady-state>
</ve>
```

Obtain Server History

Description

Use this request to obtain the modification history for the specified server. Every time a server configuration is changed, a snapshot of the configuration is taken and saved. This request allows to retrieve these records and use them for statistics.

Syntax 1

Use this syntax to retrieve the specified number of history records. The `{records}` part is used to specify the number of records (from the end) to include in the result set.

```
GET baseURL/ve/{ve-name}/history/{records}
```

Syntax 2

Use this syntax to retrieve the records that were created during the specified date-time period. The `{from-inclusive}` and `{to-exclusive}` parts must contain the datetime values specifying the datetime interval.

```
GET baseURL/ve/{ve-name}/history/{from-inclusive}/{to-exclusive}
```

The datetime format is as follows:

```
yyyy-MM-dd HH:mm Z
```

where

`yyyy` specifies the year

`MM` specifies the month (numeric)

`dd` specifies the day of month (numeric)

`HH` specifies the hour using the 24-hour notation (military time)

`mm` specifies minutes

`Z` specifies the time zone using standard time zone abbreviations

Datetime example: `2011-06-28 21:00 CET`

Request Parameters

None

Response

Element	Attribute	Description
`ve-history`		
`ve-snapshot`		
	`cpu`	CPU clock speed, in megahertz. Type: int
	`ram`	RAM size, in megabytes. Type: int
	`local-disk`	Local disk size, in gigabytes. Type: int
	`nbd`	*This parameter is reserved for future use.* Type: int
	`bandwidth`	Network bandwidth, in kbps. Type: int
	`backup-schedule`	Backup schedule name. Type: string
	`last-operation-rc`	Last operation return code. Type: int
	`last-touched-from`	Used internally. Type: string
	`state`	Server state right after the modification operation began executing. Type: string
	`steady-state`	Last known steady state. Type: string
	`last-changed-by`	The Name of the user who performed the modification. Type: string
	`event-timestamp`	The modification event timestamp. Type: string
	`no-of-public-ip`	Number of public IPv4 addresses. Type: int
	`no-of-public-ipv6`	No of public IPv6 addresses. Type: int

	`is-lb`	Specifies whether this server is a load balancer or a regular server: `true` -- load balancer `false` -- regular server Type: boolean
	`private-incoming-traffic`	Private incoming traffic, in bytes. Type: long
	`private-outgoing-traffic`	Private outgoing traffic, in bytes. Type: long
	`public-incoming-traffic`	Public incoming traffic, in bytes. Type: long
	`public-outgoing-traffic`	Public outgoing traffic, in bytes. Type: long

Examples

The following example retrieves the three most recent server history records.

Request

```
GET https://c2u-web:4465/paci/v1.0/ve/Web40/history/3
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ve-history>
    <ve-snapshot public-outgoing-traffic="0" public-incoming-traffic="0" private-
outgoing-traffic="0" private-incoming-traffic="0" is-lb="false" no-of-public-ipv6="0"
no-of-public-ip="0" event-timestamp="2011-06-23 16:34:32.911962+04" last-changed-
by="tobotras" state="CREATION_IN_PROGRESS" last-touched-from="im1" bandwidth="100"
nbd="0" local-disk="0" ram="512" cpu="1000"/>
    <ve-snapshot public-outgoing-traffic="0" public-incoming-traffic="0" private-
outgoing-traffic="0" private-incoming-traffic="0" is-lb="false" no-of-public-ipv6="0"
no-of-public-ip="0" event-timestamp="2011-06-23 16:34:33.131139+04" last-changed-
by="tobotras" state="CREATION_IN_PROGRESS" last-touched-from="im1" backup-
schedule="hourly" bandwidth="100" nbd="0" local-disk="0" ram="512" cpu="1000"/>
    <ve-snapshot public-outgoing-traffic="0" public-incoming-traffic="0" private-
outgoing-traffic="0" private-incoming-traffic="0" is-lb="false" no-of-public-ipv6="0"
no-of-public-ip="0" event-timestamp="2011-06-23 16:35:03.408376+04" last-changed-
by="InstanceManager" steady-state="CREATED" state="CREATED" last-touched-from="im1"
backup-schedule="hourly" bandwidth="100" nbd="0" local-disk="1" ram="512" cpu="1000"/>
</ve-history>
```

The following example retrieves the server history records that were created between the specified start and end dates.

Request

```
GET https://c2u-web:4465/paci/v1.0/ve/Web40/history/2011-06-27 21:00 CET/2011-06-29
21:00 CET
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ve-history>
    <ve-snapshot public-outgoing-traffic="984" public-incoming-traffic="49785508"
private-outgoing-traffic="0" private-incoming-traffic="0" is-lb="false" no-of-public-
ipv6="0" no-of-public-ip="0" event-timestamp="2011-06-28 00:00:01.321945+04" last-
changed-by="InstanceManager" steady-state="STARTED" state="STARTED" last-touched-
from="im1" backup-schedule="hourly" bandwidth="100" nbd="0" local-disk="1" ram="512"
cpu="1000"/>
    <ve-snapshot public-outgoing-traffic="984" public-incoming-traffic="50043427"
private-outgoing-traffic="0" private-incoming-traffic="0" is-lb="false" no-of-public-
ipv6="0" no-of-public-ip="0" event-timestamp="2011-06-28 00:30:01.288039+04" last-
changed-by="InstanceManager" steady-state="STARTED" state="STARTED" last-touched-
from="im1" backup-schedule="hourly" bandwidth="100" nbd="0" local-disk="1" ram="512"
cpu="1000"/>
    <ve-snapshot public-outgoing-traffic="984" public-incoming-traffic="50301986"
private-outgoing-traffic="0" private-incoming-traffic="0" is-lb="false" no-of-public-
ipv6="0" no-of-public-ip="0" event-timestamp="2011-06-28 01:00:01.30681+04" last-
changed-by="InstanceManager" steady-state="STARTED" state="STARTED" last-touched-
from="im1" backup-schedule="hourly" bandwidth="100" nbd="0" local-disk="1" ram="512"
cpu="1000"/>
```

```
    <ve-snapshot public-outgoing-traffic="984" public-incoming-traffic="50559169"
private-outgoing-traffic="0" private-incoming-traffic="0" is-lb="false" no-of-public-
ipv6="0" no-of-public-ip="0" event-timestamp="2011-06-28 01:30:01.287452+04" last-
changed-by="InstanceManager" steady-state="STARTED" state="STARTED" last-touched-
from="im1" backup-schedule="hourly" bandwidth="100" nbd="0" local-disk="1" ram="512"
cpu="1000"/>
</ve-history>
```

Delete Server

Description

Use this request to permanently delete a server. Please note that you can only delete a fully stopped server. If a server is in a transition state (stopping, starting, a disk is being attached to it, etc.) it cannot be deleted.

Syntax

```
DELETE baseURL/ve/{ve-name}
```

Request Parameters

None

Response

A string describing the status of the operation.

Example

Request

```
DELETE  https://c2u-web:4465/paci/v1.0/ve/Web40
```

Response

```
VE DELETE initiated
```

Firewall Management

This section describes requests that allow to configure a firewall in the the specified server.

List Firewall Rules

Description

Use this request to obtain a list of existing firewall rules for the specified server. The `{ve-name}` part of the request URL must contain the server name.

Syntax

```
GET baseURL/ve/{ve-name}/firewall
```

Request Parameters

None

Response

Element	Attribute	Description
`firewall`		Container for firewall rule list.
`rule`		Container for firewall information. This element may appear more than once (one for each rule).
	`id`	Firewall rule database ID (used internally). Type: int
	`name`	Firewall rule name. Type:string
	`protocol`	Communication protocol (TCP, UDP). Type: string
	`local-port`	Local port number. Type: int
	`remote-port`	Remote port number. Type: int

`remote-net`		Remote address and an optional mask. If no mask is specified, the address indicates a host address. This element may appear more than once. Type: string

Example

Request

```
GET https://c2u-web:4465/paci/v1.0/ve/Web60/firewall
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<firewall>
    <rule remote-port="0" local-port="25" protocol="TCP" name="smtp" id="6">
        <remote-net>192.168.1.0/24</remote-net>
        <remote-net>192.168.2.0/32</remote-net>
        <remote-net>dead:beef::/64</remote-net>
    </rule>
    <rule remote-port="2" local-port="80" protocol="TCP" name="http" id="7">
        <remote-net>192.168.3.0/24</remote-net>
        <remote-net>192.168.4.0/32</remote-net>
        <remote-net>dead:beef:abcd::/64</remote-net>
    </rule>
</firewall>
```

Create Firewall Rules

Description

Use this request to create firewall rules. The `{ve-name}` part of the request URL must contain the server name.

Syntax

```
POST baseURL/ve/{ve-name}/firewall
```

Request Parameters

Element	Attribute	Description
`firewall`		Container for firewall rule list. You can create more than one rule in a single request.
`rule`		Container for firewall information. This element may appear more than once (one for each rule).
	`name`	Firewall rule name. Type:string
	`protocol`	Communication protocol (TCP, UDP). Type: string

	`local-port`	Local port number. Type: int
	`remote-port`	Remote port number. Type: int
`remote-net`		Remote address and an optional mask. If no mask is specified, the address indicates a host address. This element may appear more than once. Type: string

Response

A text message describing the operation status.

Example

Request

```
POST https://c2u-web:4465/paci/v1.0/ve/Web60/firewall
```

Request Body

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<firewall>
  <rule name="smtp" protocol="TCP" local-port="25" remote-port="0">
    <remote-net>192.168.1.0/24</remote-net>
    <remote-net>192.168.2.0</remote-net>
    <remote-net>DEAD:BEEF::/64</remote-net>
  </rule>
  <rule name="http" protocol="TCP" local-port="80" remote-port="2">
    <remote-net>192.168.3.0/24</remote-net>
    <remote-net>192.168.4.0</remote-net>
    <remote-net>DEAD:BEEF:ABCD::/64</remote-net>
  </rule>
</firewall>
```

Response

```
Firewall configuration started
```

Modify Firewall Rules

Description

Use this request to modify an existing firewall. The request replaces all existing rules with the new ones. To keep existing rules and add more, first obtain the list of the existing rules, then add new rules to it and use the complete list as an input.

Syntax

```
PUT baseURL/ve/{ve-name}/firewall
```

Request Parameters

Element	Attribute	Description
`firewall`		Container for firewall rule list. You can specify more than one rule in a single request.
`rule`		Container for firewall information. This element may appear more than once (one for each rule).
	`name`	Firewall rule name. Type:string
	`protocol`	Communication protocol (TCP, UDP). Type: string
	`local-port`	Local port number. Type: int
	`remote-port`	Remote port number. Type: int

`remote-net`		Remote address and an optional mask. If no mask is specified, the address indicates a host address. This element may appear more than once. Type: string

Response

A text message describing the status of the operation.

Example

Request

```
PUT https://c2u-web:4465/paci/v1.0/ve/Web40/firewall
```

Request Body

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<firewall>
  <rule name="smtp1" protocol="TCP" local-port="25" remote-port="0">
    <remote-net>192.168.2.0/24</remote-net>
    <remote-net>192.168.3.0</remote-net>
    <remote-net>DEAD:BEEF::/64</remote-net>
  </rule>
  <rule name="http1" protocol="TCP" local-port="80" remote-port="2">
    <remote-net>192.168.4.0/24</remote-net>
    <remote-net>192.168.5.0</remote-net>
    <remote-net>DEAD:BEEF:ABCD::/64</remote-net>
  </rule>
</firewall>
```

Response

```
Firewall re-configuration started
```

Delete Firewall Rules

Description

Use this request to delete all existing firewall rules. To delete a specific rule, retrieve all existing rules, modify the result set as needed and then use it as an input in the firewall modification request (p. 39).

Syntax

```
DELETE baseURL/ve/{ve-name}/firewall
```

Request Parameters

None

Response

A text message describing the status of the operation.

Example

Request

```
DELETE https://c2u-web:4465/paci/v1.0/ve/Web60/firewall
```

Response

```
Firewall removal started
```

Backup and Restore

This section describes requests that allow to manage server backups. Backups are performed automatically according to a selected backup schedule. Backup schedules are created and configured by the system administrator and define when and how often the server backups will be performed. Backup schedules also define the number of incremental backups before a full backup is performed and a maximum number of backups that will be kept on a backup server. The user must select a backup schedule from one of the available schedules and assign it to a server. You can use the API to set or remove a backup schedule, list existing backups, and to restore a server from a backup.

Set Backup Schedule

Description

Use this request to assign a backup schedule to the specified server. The `{schedule-name}` part must contain the name of an existing backup schedule. Backup schedules are created by the system administrator. You can obtain the list of the existing backup schedules using the `schedule` interface (p. 75).

Syntax

```
PUT baseURL/ve/{ve-name}/schedule/{schedule-name}
```

Request Parameters

None

Response

A text message describing the status of the operation.

Example

Request

```
PUT https://c2u-web:4465/paci/v1.0/ve/Web60/schedule/daily
```

Response

```
Backup schedule assigned
```

Cancel Server Backup Schedule

Description

Use this call to cancel a backup schedule assigned to a server. When a backup schedule is canceled, the server backups will not be performed anymore.

Syntax

```
PUT baseURL/ve/{ve-name}/nobackup
```

Request Parameters

None

Response

A text message describing the status of the operation.

Example

Request

```
PUT https://c2u-web:4465/paci/v1.0/ve/Web60/nobackup
```

Response

```
Backup schedule removed
```

List Backups

Description

List available backups for the specified server. The `{from-inclusive}` and `{to-exclusive}` parts of the request URL must specify a datetime interval for which to retrieve the backups. The datetime format is as follows:

`yyyy-MM-dd HH:mm Z`

where

`yyyy` specifies the year

`MM` specifies the month (numeric)

`dd` specifies the day of month (numeric)

`HH` specifies the hour using the 24-hour notation (military time)

`mm` specifies minutes

`Z` specifies the time zone using standard time zone abbreviations

Datetime example: `2011-06-28 21:00 CET`

Syntax

```
GET baseURL/ve/{ve-name}/backups/{from-inclusive}/{to-exclusive}
```

Request Parameters

None

Response

Element	Attribute	Description
`ve-backups`		Container for backups.
`backup`		Container for backup information. This element may appear more than once (one for each backup).
	`im-backup-id`	Backup ID (used internally) Type: int

	`cloud-backup-id`	Cloud backup ID. Use this ID when restoring a server from a backup. Type: string
	`schedule-name`	Backup schedule name. Type: string
	`started`	Backup start date/time. Type: string
	`ended`	Backup end date/time. Type: string
	`successful`	Specifies whether the backup was successful. Type: boolean
	`backup-size`	Backup file size. Type: int
	`backup-node-name`	Name of the backup node. Type: string
	`delta_of`	Used internally. Type: int

Example

Request

```
GET https://c2u-web:4465/paci/v1.0/ve/Web60/backups/2011-06-27 19:00 CET/2011-06-29
22:00 CET
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ve-backups>
    <backup backup-node-name="c2u-deploy.pl.com" backup-size="1" successful="true"
ended="2011-06-28 20:01:34.710062+04" started="2011-06-28 20:00:00.107158+04" schedule-
name="hourly" cloud-backup-id="{d79d63f2-1136-409f-8573-a3f37bffe83b}" im-backup-
id="3"/>
    <backup delta_of="3" backup-node-name="c2u-deploy.pl.com" backup-size="1"
successful="true" ended="2011-06-28 21:01:34.176135+04" started="2011-06-28
21:00:00.142492+04" schedule-name="hourly" cloud-backup-id="{d79d63f2-1136-409f-8573-
a3f37bffe83b}.2" im-backup-id="5"/>
    <backup backup-node-name="c2u-deploy.pl.com" backup-size="1" successful="true"
ended="2011-06-29 12:01:32.550893+04" started="2011-06-29 12:00:00.100254+04" schedule-
name="daily" cloud-backup-id="{d79d63f2-1136-409f-8573-a3f37bffe83b}.3" im-backup-
id="11"/>
</ve-backups>
```

Restore a Server

Description

Restore a specified server from a specified backup. A server must be stopped in order to perform a restore operation. The `{cloud-backup-id}` part of the request URL must contain the backup ID. See **List Backups** for the information on how to obtain the list of the available backups and their IDs. Please note that the complete backup ID string must be specified in the request, including curly brackets and any other leading and trailing characters (if any).

Syntax

```
PUT baseURL/ve/{ve-name}/restore/{cloud-backup-id}
```

Request Parameters

None

Response

A text message describing the status of the operation.

Example

Request

```
PUT https://c2u-web:4465/paci/v1.0/ve/Web60/restore/{d79d63f2-1136-409f-8573-
a3f37bffe83b}.2
```

Response

```
VE RESTORE_FROM_BACKUP initiated
```

Application Template Installation

The requests described in this section are used to obtain information about the available application templates and install them into servers. An application template represents a software application that can be installed into a server. An application template is created for a specific operating system type, so it must be compatible with the OS template that was used to create a server in order to be installed in it. To install an application template, first obtain a list of the available templates, then choose the template of interest and obtain a detailed information about it. Make sure that the template is compatible with the OS template of the target server and then install it.

List Application Templates

Description

Use this request to obtain a list of the available application templates.

Syntax

```
GET baseURL/application-template
```

Request Parameters

None

Response

Element	Attribute	Description
`application-list`		Container for application template list.
`application-template`		Container for application template information. This element may appear more than once (one for each template).
	`id`	Template ID (used internally). Type: int
	`name`	Template name. Use this value in other requests that expect template name as a parameter. Type: string
	`active`	Specifies whether the template is active or not. Inactive templates cannot be installed into servers. Type: boolean
	`c2u-version`	Template PACI version (used internally). Type: string
	`for-os`	Name of the operating system template for which this application template is designed. Type: string

Example

Request

```
GET https://c2u-web:4465/paci/v1.0/application-template
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<application-list>
    <application-template for-os="centos-6-x86_64" c2u-version="1" active="true"
name="php" id="1"/>
</application-list>
```

Get Application Template Info

Description

Use this request to obtain a detailed information about a specified application template. The `{name}` parameter must contain the application template name. The `{for-os}` parameter must contain the name of the operating system template for which the template is designed. The `{name}` and `{for-os}` parameters together uniquely identify an application template. There could be multiple templates with the same name but designed for different operating systems.

Syntax

```
GET baseURL/application-template/{name}/{for-os}
```

Request Parameters

None

Response

Element	Attribute	Description
`application-template`		Container for application template information. This element may appear more than once (one for each template).
	`id`	Template ID (used internally). Type: int
	`name`	Template name. Type: string
	`c2u-version`	Template PACI version (used internally). Type: string

	`active`	Specifies whether the template is active or inactive. Inactive templates cannot be installed into servers. Type: boolean
	`for-os`	Name of the operating system for which this template was designed. Type: string
`description`		Template description. Type: string

Example

The following example obtains information about an application template named "php".

Request

```
GET https://c2u-web:4465/paci/v1.0/application-template/php/centos-6-x86_64
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<application-template for-os="centos-6-x86_64" c2u-version="1" active="true" name="php"
id="1">
    <description>php</description>
</application-template>
```

Install Application Template Into Server

Description

Use this request to install an application template into a server. The application template must be compatible with the OS template installed in the target server. The `{app-name}` part of the URL must contain the application template name.

Syntax

```
PUT baseURL/ve/{ve-name}/install/{app-name}
```

Request Parameters

None

Response

A text message describing the status of the operation.

Example

The following example installs an application template named ha-proxy into a server named Web101.

Request

```
PUT https://c2u-web:4465/paci/v1.0/ve/Web101/install/ha-proxy
```

Response

```
VE INSTALL_APPLICATION initiated
```

Image Management

The requests described in this section are used to manage server images. A server image is created from an existing server and can be used later to create new servers. The new servers will be exact copies of the source server except the network settings to avoid collisions.

List Images

Description

Use this request to obtain a list of the existing server images.

Syntax

```
GET baseURL/image
```

Request Parameters

None

Response

Element	Attribute	Description
`image-list`		Container for image list.
`image-info`		Container for image information. This element may appear more than once (one for each image).
	`name`	Image name. Type: string
	`size`	Image size, in gigabytes. Type: int
	`created`	Image creation date. Type: string
	`subscription-id`	Subscription ID to which this image belongs. Type: int
	`load-balancer`	Specifies whether this is a load balancer. Type: boolean
	`active`	Specifies whether the image is active or not. Inactive images cannot be used to create servers. Type: boolean
	`image-of`	The name of the server from which this image was created. Type: string

Example

Request

```
GET https://c2u-web:4465/paci/v1.0/image
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<image-list>
    <image-info image-of="Web40" active="true" load-balancer="true" subscription-
id="100001" created="2011-09-01 19:19:17.84091+04" size="2" name="lbimage"/>
</image-list>
```

Get Image Info

Description

Use this request to obtain a detailed information for the specified server image.

Syntax

```
GET baseURL/image/{image-name}
```

Request Parameters

None

Response

Element	Attribute	Description
`ve-image`		Container for image information.
	`id`	Image database ID (used internally). Type: int
	`bnode-uuid`	Used internally. Type: string
	`customer-id`	Customer ID. Type: int
	`subscription-id`	Subscription ID. Type: int
	`load-balancer`	Specifies whether this is a load balancer. Type: boolean

	`active`	Specifies whether the image is active. Inactive images cannot be used to create servers. Type: boolean
	`image-of`	The name of the server from which this image was created. Type: string
	`name`	Image name. Type: string
	`cpu-number`	Number of CPU cores. Type: int
	`cpu-power`	CPU clock speed in MHz. Type: int
	`ram-size`	RAM size in MB. Type: int
	`bandwidth`	Network bandwidth in kbps. Type: int
	`login`	User login name. Type: string
	`template-id`	Used internally. Type: int
	`primary-disk-id`	Used internally. Type: int
	`image-size`	Image size, in gigabytes. Type: int
	`created`	Image creation date. Type: string
	`no-of-public-ip`	Number of public IPv4 addresses. Type: int
	`no-of-public-ipv6`	Number of public IPv6 addresses. Type: int
`description`		Image description. Type: string
`disks`		Container for disk information. This element may appear up to 10 times (one for each disk). *At the time of this writing only one disk can be added to a server. This will change in the future*.
	`id`	Disk database ID (used internally). Type: int

	`local`	Specifies whether this is a local or a network disk. *At the time of this writing only local disks are supported*. Type: boolean
	`primary`	This is a reserved parameter. Type: boolean
	`size`	Disk size in GB. Type: int

Example

Request

```
GET https://c2u-web:4465/paci/v1.0/image/Web-Image-101
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ve-image no-of-public-ipv6="1" no-of-public-ip="1" created="2011-06-29
17:28:01.50428+04" image-size="1" primary-disk-id="0" template-id="1" login="root"
bandwidth="100" ram-size="512" cpu-power="1600" cpu-number="2" name="Web-Image-101"
image-of="Web40" active="true" load-balancer="true" subscription-id="1000001" customer-
id="1000001" bnode-uuid="6" id="1">
    <description>Image of Web101</description>
    <disks size="1" primary="true" local="true" id="0"/>
</ve-image>
```

Create Image from Server

Description

Use this request to create an image from an existing server. The server must be stopped before you attempt to create an image from it. The `{ve-name}` part must contain the source server name. The `{image-name}` part must contain the desired image name. **Syntax 1** can only be used when the customer account to which this user belongs has a single subscription. In case of multiple subscriptions use **Syntax 2**.

Syntax 1

Use this syntax when the customer account to which this user belongs has just one subscription.

```
POST baseURL/image/{ve-name}/create/{image-name}
```

Syntax 2

Use this syntax when the customer account has multiple subscriptions. The `{subscription-id}` must contain the desired subscription ID.

```
POST baseURL/image/{ve-name}/{subscription-id}/create/{image-name}
```

Request Parameters

None

Response

A text message describing the status of the operation.

Example

The following example creates an image from the Web101 server for subscription 1000001.

Request

```
POST https://c2u-web:4465/paci/v1.0/image/Web101/1000001/create/Web-Image-101
```

Response

```
Image creation initiated
```

Delete Image

Description

Use this request to delete an existing server image.

Syntax

```
DELETE baseURL/image/{image-name}
```

Request Parameters

None

Response

A text message describing the status of the operation.

Example

Request

```
DELETE https://c2u-web:4465/paci/v1.0/image/Web-Image-102
```

Response

```
Delete image initiated
```

Load Balancer Management

The requests described in this section are used to manage load balancers. To use load balancing, first create a load balancer and then attach the desired servers to it. You can use the PACI REST API to list existing load balancers, to create load balancers, to attach and detach servers to/from load balancers, and to delete load balancers.

List Load Balancers

Description

Use this request to obtain a list of the available load balancers.

Syntax

```
GET baseURL/load-balancer
```

Request Parameters

None

Response

Element	Attribute	Description
`lb-list`		Container for load balancer list.
`load-balancer`		Container for load balancer information. The element may appear more than once (one for each load balancer).
	`name`	Load balancer name. Type: string
	`subscription-id`	Subscription ID. Type: int
	`state`	Load balancer state (e.g. STARTED, etc). Type: string

Example

Request

```
GET https://c2u-web:4465/paci/v1.0/load-balancer
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<lb-list>
    <load-balancer state="STARTED" subscription-id="1" name="LB1"/>
    <load-balancer state="STARTED" subscription-id="1" name="LB2"/>
</lb-list>
```

Get Load Balancer Info

Description

Use this request to obtain the information about a specified load balancer. The `{lb-name}` part of the URL must contain the load balancer name.

Syntax

```
GET baseURL/load-balancer/{lb-name}
```

Request Parameters

None

Response

Element	Attribute	Description
`load-balancer`		Container for load balancer information.
`id`		Load balancer database ID (used internally). Type: int
`uuid`		Server UUID. Type: string
`hnId`		ID of the hardware node on which this load balancer resides. Type: string
`customer-id`		ID of the customer account to which the load balancer belongs. Type: int
`name`		Server name. Type: string
`description`		Load balancer description. Type: string
`subscription-id`		ID of the subscription to which the load balancer belongs. Type: int
`cpu`		Container for CPU information.
	`number`	Number of CPU cores. Type: int

	`power`	CPU clock speed in megahertz. Type: int
`ram-size`		RAM size in megabytes. Type: int
`bandwidth`		Network bandwidth in kbps. Type: int
`ve-disk`		Container for hard disk information. *At the time of this writing, a server can have only one hard disk. This will change in the future.*
	`id`	Disk database ID (used internally) Type: int
	`local`	Specified whether this is a local or a network disk. *Only local disks are supported at the the time of this writing.* Type: boolean
	`primary`	Specifies whether this is a system disk. This is a reserved parameter. Type: boolean
	`size`	Disk size in gigabytes. Type: int
	`created`	Disk status. Type: boolean Possible values: `true` -- the disk creation process completed. `false` -- the disk is being created and is not available for use.
`platform`		Container for operating system template and related information.
`template-info`		Container for OS template information.
	`name`	Template name. Type: string
	`vendorId`	Template vendor ID. Type: string
	`c2uId`	PACI template ID (used internally). Type: string

os-info		Container for operating system information. This information is a part of the OS template info.
	type	OS type (Linux. Windows, etc.) Type: string
	technology	Virtualization technology used. Type: string Possible values: CT - Virtuozzo Container, VM - Parallels virtual machine.
network		Container for network information.
	private_ip	Private IP address and mask. Type: string
public-ip		Container for public IPv4 address info. This element may appear more than once (one for each IP address).
	id	Database ID (used internally). Type: int
	address	IP address and mask. Type: string
	gateway	Gateway IP address. Type: string
	chunk-ref	Used internally. Type: int
public-ipv6		Container for public IPv6 address info. This element may appear more than once (one for each IPv6 address).
	id	Database ID (used internally). Type: int
	address	IPv6 address and mask. Type: string
	gateway	Gateway IPv6 address. Type: string
	ipv6-net-id	Used internally. Type: int

`state`		A string describing the load balancer state or transition. Type: string
`primary-disk-id`		Used internally. Type: int
`template-id`		Used internally. Type: int
`admin`		Container for the server administrator credentials.
	`login`	Server administrator login name. Type: string
	`password`	Password. Type: string
`used-by`		Information about a server attached to this load balancer. This element may appear more than once (one for each server).
	`ve-name`	Server name. Type: string
	`ip`	Server IP address. Type: string

Example

The following example obtains information for the load balancer named LB1.

Request

```
GET https://c2u-web:4465/paci/v1.0/load-balancer/LB1
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<load-balancer>
    <id>10</id>
    <uuid>_374f525b.1322595125d._7ffb</uuid>
    <hnId>4</hnId>
    <customer-id>1</customer-id>
    <name>LB1</name>
    <description>Created from image [lbimage]</description>
    <subscription-id>1</subscription-id>
    <cpu power="700" number="1"/>
    <ram-size>1024</ram-size>
    <bandwidth>1000</bandwidth>
    <ve-disk created="true" size="2" local="true" id="0"/>
    <platform>
        <template-info c2uId="1" vendorId="5" name="centos-5-x86"/>
        <os-info technology="CT" type="linux-free"/>
    </platform>
    <network private-ip="1.1.1.230/8">
        <public-ip chunk-ref="1" gateway="2.2.2.1" address="2.2.2.208/16" id="12"/>
    </network>
    <state>STARTED</state>
    <primary-disk-id>0</primary-disk-id>
    <template-id>1</template-id>
    <admin password="[hidden]" login="root"/>
    <used-by ip="1.1.1.224" ve-name="Web20"/>
    <used-by ip="1.1.1.225" ve-name="CT1"/>
    <used-by ip="1.1.1.226" ve-name="CT2"/>
</load-balancer>
```

Create Load Balancer

Description

Use this request to create a load balancer. The `{name}` part of the request URL must contain the user-defined load balancer name.

Syntax

```
POST baseURL/load-balancer/create/{name}
```

Request Parameters

None

Response

Element	Attribute	Description
`pwd-response`		Container for response info.
`message`		A text message describing the status of the operation.
`password`		Automatically generated password.

Example

The following example creates a load balancer named LB100.

Request

```
POST https://c2u-web:4465/paci/v1.0/load-balancer/create/LB100
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<pwd-response>
    <message>LB creation initiated</message>
    <password>668ttoAHL</password>
</pwd-response>
```

Create Load Balancer for Subscription

Description

Use this request to create a load balancer for a specified subscription. The current user may be a member of a customer account with more than one subscription. This request allows to create a load balancer for a specific subscription. The `{subscription-id}` part of the request URL must contain a valid subscription ID. The `{lb-name}` part must contain a user-defined load balancer name.

Syntax

```
POST baseURL/load-balancer/{subscription-id}/create/{lb-name}
```

Request Parameters

None

Response

Element	Attribute	Description
`pwd-response`		Container for response info.
`message`		A text message describing the status of the operation.
`password`		Automatically generated password.

Example

The following example creates a load balancer named LB100001 for subscription 100001.

Request

```
POST https://c2u-web:4465/paci/v1.0/load-balancer/100001/create/LB100001
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<pwd-response>
    <message>LB creation initiated</message>
    <password>262uhdQEM</password>
</pwd-response>
```

Obtain Load Balancer History

Description

Use this request to obtain a load balancer history. The `{lb-name}` part of the request URL must contain the name of the load balancer for which to retrieve the history. The `{nrecords}` part must contain the number of records (from the end) to include in the result set.

Syntax

```
GET baseURL/load-balancer/{lb-name}/history/{nrecords}
```

Request Parameters

None

Response

Element	Attribute	Description
`ve-history`		Container for load balancer history.
`ve-snapshot`		Container for a single history record. This element may appear more than once (one for each record).
	`cpu`	CPU clock speed, in megahertz. Type: int
	`ram`	RAM size, in megabytes. Type: int
	`local-disk`	Local disk size, in gigabytes. Type: int
	`nbd`	*This parameter is reserved for future use*. Type: int
	`bandwidth`	Network bandwidth, in kbps. Type: int
	`last-touched-from`	Used internally. Type: string
	`state`	Server state right after the modification operation began executing. Type: string
	`last-changed-by`	Name of the user initiating the modification. Type: string

	`event-timestamp`	The modification event timestamp. Type: string
	`no-of-public-ip`	Number of public IPv4 addresses. Type: int
	`no-of-public-ipv6`	No of public IPv6 addresses. Type: int
	`is-lb`	Specifies whether this is a load balancer or a regular server: `true` -- load balancer `false` -- regular server Type: boolean
	`private-incoming-traffic`	Private incoming traffic, in bytes. Type: long
	`private-outgoing-traffic`	Private outgoing traffic, in bytes. Type: long
	`public-incoming-traffic`	Public incoming traffic, in bytes. Type: long
	`public-outgoing-traffic`	Public outgoing traffic, in bytes. Type: long

Example

The following example retrieves the last three records from load balancer LB101 history.

Request

```
GET https://c2u-web:4465/paci/v1.0/load-balancer/LB101/history/3
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ve-history>
    <ve-snapshot public-outgoing-traffic="0" public-incoming-traffic="0" private-
outgoing-traffic="0" private-incoming-traffic="0" is-lb="false" no-of-public-ipv6="0"
no-of-public-ip="0" event-timestamp="2011-09-01 19:22:50.326641+04" last-changed-
by="admin" state="CREATION_IN_PROGRESS" last-touched-from="im1" bandwidth="1000"
nbd="0" local-disk="0" ram="1024" cpu="700"/>
    <ve-snapshot public-outgoing-traffic="0" public-incoming-traffic="0" private-
outgoing-traffic="0" private-incoming-traffic="0" is-lb="false" no-of-public-ipv6="0"
no-of-public-ip="1" event-timestamp="2011-09-01 19:22:50.538825+04" last-changed-
by="InstanceManager" steady-state="CREATED" state="CREATED" last-touched-from="im1"
bandwidth="1000" nbd="0" local-disk="2" ram="1024" cpu="700"/>
    <ve-snapshot public-outgoing-traffic="0" public-incoming-traffic="0" private-
outgoing-traffic="0" private-incoming-traffic="0" is-lb="false" no-of-public-ipv6="0"
no-of-public-ip="1" event-timestamp="2011-09-01 19:22:50.760722+04" last-changed-
by="InstanceManager" steady-state="STARTED" state="STARTED" last-touched-from="im1"
bandwidth="1000" nbd="0" local-disk="2" ram="1024" cpu="700"/>
</ve-history>
```

Restart Load Balancer

Description

Use this request to restart a load balancer.

Syntax

```
PUT baseURL/load-balancer/{lb-name}/restart
```

Request Parameters

None

Response

A text message describing the operation status.

Example

The following example restart a load balancer named LB101.

Request

```
PUT https://c2u-web:4465/paci/v1.0/load-balancer/LB101/restart
```

Response

```
LB restart initiated
```

Delete Load Balancer

Description

Use this request to delete an existing load balancer.

Syntax

```
DELETE baseURL/load-balancer/{lb-name}
```

Request Parameters

None

Response

A text message describing the status of the operation. If there are servers attached to a load balancer the response will contain error message: "P2000023: Load balancer is in use". In such a case the servers will have to be detached first.

Example

Request

```
DELETE https://c2u-web:4465/paci/v1.0/load-balancer/LB101
```

Response

```
LB removing initiated
```

Attach Server To Load Balancer

Description

Use this request to attach a server to a load balancer. Once this request is completed, the server load will be managed by the specified load balancer. The `{lb-name}` and `{ve-name}` parts of the request URL must contain the load balancer and the server names respectively.

Syntax

```
POST baseURL/load-balancer/{lb-name}/{ve-name}
```

Request Parameters

None

Response

A text message describing the status of the operation.

Example

The following example attaches the server named Web60 to the load balancer named LB101.

Request

```
POST https://c2u-web:4465/paci/v1.0/load-balancer/LB101/Web60
```

Detach Server From Load Balancer

Description

Use this request to detach a server from a load balancer. The `{lb-name}` and `{ve-name}` parts of the request URL must contain the load balancer and the server names respectively.

Syntax

```
DELETE baseURL/load-balancer/{lb-name}/{ve-name}
```

Request Parameters

None

Response

A text message describing the status of the operation.

Example

Request

```
DELETE https://c2u-web:4465/paci/v1.0/load-balancer/LB101/Web60
```

Utilities

The request described in this section are used to obtain information required by other requests.

List Installed OS Templates

Use this call to obtain a list of the available operating system templates. An operating system template is a package which is used to create new servers. It contains a particular operating system type and version (and software applications in some cases) together with necessary instructions and is used to preconfigure a server and install the operating system into it. When creating a server, use this request to obtain a list of the available OS templates, then choose the template of interest and use its name as an input parameter in the server creation call (p. 16). The `{name}` part of the request URL is optional and may contain the OS template name. When it is included, only the information about the specified template will be retrieved.

Syntax

```
GET baseURL/template/{name}
```

Request Parameters

None

Response

Element	Attribute	Description
`template-list`		Container for template list.
`template`		Container for an individual template info. This element may appear more than once (one for each available OS template).
	`id`	Used internally.
	`name`	Template name. Use this value in API requests that expect a template identifier as an input parameter. Type: string
	`vendorId`	Template vendor ID. Type: string
	`c2uId`	Used internally.
	`osType`	Operating system type (Linux, Windows) Type: string
	`technology`	Specifies the virtualization technology used. Type: string Possible values: `CT` - Virtuozzo Container. `VM` - Parallels virtual machine.

	`active`	Specifies whether the template is active. Inactive templates cannot be used to create servers. Type: boolean
	`default`	Specifies whether this is the default template. Type: boolean
	`root-login`	Administrator login name (e.g. "root"). Type: string
	`min-hdd-size`	Minimum required hard disk space. Type: int
	`pwd-regex`	Used internally.
	`high-watermark-for-delivery`	Used internally.
	`low-watermark-for-delivery`	Used internally.
`option`		Additional template information in name/value pairs (e.g. the target CPU type and OS info). This element may appear more than once.
	`value`	Value. Type: string
	`name`	Name. Type: string

Examples

The following example retrieves the list of all available OS templates.

Request

```
GET https://c2u-web:4465/paci/v1.0/template
```

Response

Only a small portion of the actual OS template list is shown for brevity.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<template-list>
    <template high-watermark-for-delivery="100" low-watermark-for-delivery="99" pwd-
regex="[0-9]{3}[a-z]{3}[A-Z]{3}" min-hdd-size="1" root-login="root" default="true"
technology="CT" osType="linux-free" c2uId="1" vendorId="6" active="true" name="centos-
6-x86_64" id="1">
        <option value="x86_64" name="arch"/>
        <option value="centos" name="edition"/>
        <option value="English" name="lang"/>
    </template>
    <template high-watermark-for-delivery="100" low-watermark-for-delivery="99" pwd-
regex="[0-9]{3}[a-z]{3}[A-Z]{3}" min-hdd-size="10" root-login="root" default="true"
technology="VM" osType="linux-free" c2uId="1" vendorId="6" active="true" name="paci-
centos-6" id="2">
        <option value="x86_64" name="arch"/>
        <option value="centos" name="edition"/>
        <option value="English" name="lang"/>
    </template>
    <template high-watermark-for-delivery="100" low-watermark-for-delivery="99" pwd-
regex="[0-9]{3}[a-z]{3}[A-Z]{3}" min-hdd-size="20" root-login="Administrator"
default="true" technology="VM" osType="windows" c2uId="1" vendorId="2008" active="true"
name="paci-win2k8r2sp1" id="3">
        <option value="x86_64" name="arch"/>
        <option value="datacenter" name="edition"/>
        <option value="English" name="lang"/>
    </template>
    <template high-watermark-for-delivery="100" low-watermark-for-delivery="99" pwd-
regex="[0-9]{3}[a-z]{3}[A-Z]{3}" min-hdd-size="10" root-login="Administrator"
default="false" technology="VM" osType="windows" c2uId="1" vendorId="2008"
active="true" name="w2k8r2SP1x64_ja_dtc" id="7">
        <option value="x86_64" name="arch"/>
        <option value="datacenter" name="edition"/>
        <option value="Japan" name="lang"/>
    </template>

    ........
</template-list>
```

The following example retrieves the information about the centos-6-x86_64 operating system template.

Request

```
GET https://c2u-web:4465/paci/v1.0/template/centos-6-x86_64
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<template high-watermark-for-delivery="100" low-watermark-for-delivery="99" pwd-
regex="[0-9]{3}[a-z]{3}[A-Z]{3}" min-hdd-size="1" root-login="root" default="true"
technology="CT" osType="linux-free" c2uId="1" vendorId="6" active="true" name="centos-
6-x86_64" id="1">
        <option value="x86_64" name="arch"/>
        <option value="centos" name="edition"/>
        <option value="English" name="lang"/>
</template>
```

List Backup Schedules

Use this call to obtain the list of the available backup schedules. Backup schedules are created by system administrator. If you would like to perform server backups on a regular basis, you can obtain the list of the available schedules using this call, then choose a schedule that suits your needs and specify its name when configuring your server.

Syntax

`GET baseURL/schedule`

Request Parameters

None

Response

Element	Attribute	Description
`backup-schedule-list`		Container for backup schedule list.
`backup-schedule`		Container for an individual backup schedule.
	`id`	Schedule ID. Type: int
	`name`	Schedule name. Type: string
	`cron-expression`	Standard CRON expression. Type: string
	`enabled`	Specifies whether the backup schedule is enabled. Disabled scheduled cannot be used to perform backups. Type: boolean
	`backups-to-keep`	Maximum number of server backups to keep on the backup server. Type: int

	`no-of-incremental`	Number of incremental backups before a full backup is performed. Type: int

Example

Request

```
GET https://c2u-web:4465/paci/v1.0/schedule
```

Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<backup-schedule-list>
    <backup-schedule no-of-incremental="6" backups-to-keep="30" enabled="true" cron-
expression="0 0 12 * * ?" name="daily" id="1">
        <description>Daily</description>
    </backup-schedule>
    <backup-schedule no-of-incremental="5" backups-to-keep="24" enabled="true" cron-
expression="0 0 * * * ?" name="hourly" id="3">
        <description>hourly</description>
    </backup-schedule>
    <backup-schedule no-of-incremental="4" backups-to-keep="25" enabled="true" cron-
expression="0 0 6 ? * 7" name="weekly" id="2">
        <description>Weekly</description>
    </backup-schedule>
</backup-schedule-list>
```

Index

U