Red Hat CloudForms 4.7

Lenovo Physical Infrastructure Provider REST API

A reference to the Lenovo physical infrastructure provider REST API

Last Updated: 2019-01-30

Red Hat CloudForms 4.7 Lenovo Physical Infrastructure Provider REST API

A reference to the Lenovo physical infrastructure provider REST API

Red Hat CloudForms Documentation Team
cloudforms-docs@redhat.com

Lenovo Documentation
icfeedback@lenovo.com

Legal Notice

Abstract

A reference to the Lenovo physical infrastructure provider REST API. If you have a suggestion for improving this guide or have found an error, please submit a Bugzilla report at http://bugzilla.redhat.com against Red Hat CloudForms Management Engine for the Documentation component. Please provide specific details, such as the section number, guide name, and CloudForms version so we can easily locate the content.

Table of Contents

CHAPTER 1. REST API

You can use the REST API to retrieve information about providers, physical-infrastructure providers, and physical servers. You can use the CloudForms REST API if you need information about the servers in your physical infrastructure.

Before you begin

To use the REST API, you must log in to CloudForms Management Engine API as a user who has permission to access the API. The default user is admin, password is smartvm.

1.1. RETRIEVING PROVIDER DETAILS

To retrieve information for all providers, use the following request URL:

```
GET /api/providers
```

A JSON response with the following keys is returned.

```
actions (list of actions that can be used)
count (total number of providers)
name (URI name)
pages (number of pages)
resource (list of URLs for each provider that is managed by CloudForms)
subcount (number of providers that were returned)
```

Examples

- To retrieve all information about the providers, use the **?expand=resources** query parameter, for example:

  ```
  GET /api/providers?expand=resources
  ```

- To retrieve a specific information about the providers (such as name and ID), use the **?expand=resources&attributes=<attribute_list>** query parameters, for example:

  ```
  GET /api/providers?expand=resources&attributes=id,name
  ```

- To retrieve additional information about the providers (such as port, hostname, and IP address), use the **?expand=resources&attributes=<attribute_list>** query parameters, for example:

  ```
  GET /api/providers?
  expand=resources&attributes=port,hostname,ipaddress
  ```

- To retrieve information about only Lenovo physical-infrastructure providers, use the **?expand=resources&filter[]=type=<provider_type>** attribute against all provider resources, for example:

  ```
  GET /api/providers?
  expand=resources&filter[]=type=ManageIQ::Providers::Lenovo::Physical
  InfraManager
  ```

1.2. RETRIEVING PROVIDER DETAILS BY ID

To retrieve information for a specific provider, use the following request URL where <provider_ID> is the unique identifier of the provider:

```
GET /api/providers/<provider_ID>
```

A JSON response with the following keys is returned.

```
href
id
name
created_on
updated_on
guid
zone_id
type
last_refresh_date
tenant_id
enabled
options
actions
```

Examples

- To retrieve a specific information about the provider (such as name and ID), use the **?attributes=<attribute_list>** query parameters, for example:

  ```
  GET /api/providers/<provider_ID>?expand=resources&attributes=id,name
  ```

- To retrieve additional information about the provider (such as port, hostname, and IP address), use the **?attributes=<attribute_list>** query parameters, for example:

  ```
  GET /api/providers/<provider_ID>?expand=resources
  &attributes=port,hostname,ipaddress
  ```

1.3. PERFORMING PHYSICAL-INFRASTRUCTURE PROVIDER ACTIONS

You can use the REST API to perform actions on a physical-infrastructure provider, such as creating, modifying and deleting.

Note: To perform actions on physical-infrastructure providers, you must create a basic authentication using your preferred tool or library, and use the user name and password for the CloudForms instance when using the REST API.

1.3.1. Creating a Lenovo physical-infrastructure provider

To create a Lenovo physical-infrastructure provider, send a POST request using the following request URL and JSON request body.

```
POST /api/providers
```

Request body:

```
{
   "action": "create",
   "credentials": {
      "password": "<PASSWORD>",
      "userid": "<USERNAME>"
   },
   "hostname": "<LENOVO XCLARITY ADMINISTRATOR ADDRESS>",
   "name": "<NAME>",
   "port": "<THE SERVICE PORT>",
   "type": "ManageIQ::Providers::Lenovo::PhysicalInfraManager"
}
```

1.3.2. Modifying properties for a Lenovo physical-infrastructure provider

To modify properties of a Lenovo physical-infrastructure provider, send a POST request using the following request URL and JSON request body.

```
POST /api/providers
```

Request body:

```
{
   "action":"edit",
   "credentials": {
      "password":"<new_passowrd>",
      "userid":"<new_user_name>"
   },
   "hostname":"<New_Lenovo_XClarity_Administrator_address>",
   "name":"<new_provider_name>",
   "port":"<new_service_port>"
}
```

1.3.3. Refreshing information about all providers

To refresh information about all providers, send a POST request using the following request URL and JSON request body.

```
POST /api/providers
```

Request body:

```
{
   "action":"refresh"
}
```

1.3.4. Deleting a physical-infrastructure provider

To delete a specific provider:

- Send a DELETE request using the following request URL.

  ```
  DELETE /api/providers/<provider_ID>
  ```

- Send a POST request using the following request URL and JSON request body.

  ```
  POST /api/providers/<provider_ID>
  ```

  Request body:

  ```
  {
     "action":"delete"
  }
  ```

1.3.5. Performing an action on multiple physical-infrastructure providers

To perform an action on more than one physical-infrastructure providers, send a POST request using the following request URL and JSON request body.

```
POST /api/providers
```

Request body:

```
{
   "action":"<action>",
   "resources": [{
      "href":"<provider_ID>"
   },
   {
      "href":"<provider_ID>"
   }]
}
```

1.4. RETRIEVING PHYSICAL SERVER DETAILS

To retrieve information for all physical servers, use the following request URL:

```
GET /api/physical_servers
```

A JSON response with the following keys is returned.

```
actions (list of actions that can be used)
count (number of physical servers)
name (URI name)
pages (number of pages)
resource (list of URLs for each physical server)
subcount (number of physical servers that were returned)
```

Examples

- To retrieve all information about the physical servers, use the **?expand=resources** query parameter, for example:

  ```
  GET /api/physical_servers?expand=resources
  ```

- To retrieve specific information about the physical servers (such as name and ID), use the **?expand=resources&attributes=<attribute_list>** query parameters, for example:

  ```
  GET /api/physical_servers?expand=resources&attributes=id,name
  ```

- To retrieve information such as firmware, networks, guest devices, and asset details associated with physical servers, use the **?expand=resources&attributes=<attribute_list>** query parameters, for example:

  ```
  GET /api/physical_servers?
  expand=resources&attributes=hardware.firmwares,hardware.networks,har
  dware.guest_devices,hardware.assets_details
  ```

- To retrieve information such as network adapters and storage adapters in physical servers, use the **?expand=resources&attributes=<attribute_list>** query parameters, for example:

  ```
  GET /api/physical_servers?
  expand=resources&attributes=guest_devices.with_ethernet_type,guest_d
  evices.with_storage_type
  ```

- To retrieve information such as switches connected to physical servers, use the **?expand=resources&attributes=<attribute_list>** query parameters, for example:

  ```
  GET /api/physical_servers?
  expand=resources&attributes=physical_switches
  ```

- To retrieve information about only a specific physical server, use the **?expand=resources&filter[]=name=<server_name>** query attributes on all physical-server resources, for example:

  ```
  GET /api/physical_servers?expand=resources&filter[]=name=Server1-
  20127X5462
  ```

1.5. RETRIEVING PHYSICAL SERVER DETAILS BY ID

To retrieve information for a specific physical server, use the following request URL:

```
GET /api/physical_servers/<server_ID>
```

A JSON response with the following keys is returned.

```
actions
created_at
ems_id
ems_ref
field_replaceable_unit
health_state
hostname
href
location_led_state
machine_type
manufacturer
model
```

```
name
power_state
product_name
serial_number
type
uid_ems
updated_at
vendor
```

Examples

- To retrieve information about a specific physical server (such as name and ID), use the **?expand=resources&attributes=<attribute_list>** query parameters, for example:

  ```
  GET /api/physical_servers/<server_ID>?
  expand=resources&attributes=id,name
  ```

- To retrieve information such as firmware, networks, guest devices, and asset details for a specific server, use the **?expand=resources&attributes=<attribute_list>** query parameters, for example:

  ```
  GET /api/physical_servers/<server_ID>?expand=resources
  &attributes=hardware.firmwares,hardware.networks,hardware.guest_devi
  ces,hardware.assets_details
  ```

- To retrieve information such as switches that are connected to a specific server, use the **?expand=resources&attributes=<attribute_list>** query parameters, for example:

  ```
  GET /api/physical_servers/<server_ID>?
  expand=resources&attributes=physical_switches
  ```

- To retrieve information such as network adapters and storage adapters in a specific physical server, use the **?expand=resources&attributes=<attribute_list>** query parameters, for example:

  ```
  GET /api/physical_servers/<server_ID>?
  expand=resources&attributes=guest_devices.with_ethernet_type,guest_d
  evices.with_storage_type,hardware.physical_network_ports
  ```

1.6. PERFORMING PHYSICAL SERVER ACTIONS

You can use the REST API to perform management tasks on a physical server, such as powering on an off, restarting, and changing the location-LED state.

Note: To perform actions on physical servers, you must create a basic authentication using your preferred tool or library, and enter your user name and password to access the CloudForms REST API.

1.6.1. Powering on a physical server

To power on a specific physical server, send a POST request using the following request URL and JSON request body.

```
POST /api/physical_servers/<server_ID>
```

Request body:

```
{
   "action":"power_on"
}
```

1.6.2. Powering off a physical server

To shut down the operating system and power off a specific physical server, send a POST request using the following request URL and JSON request body.

```
POST /api/physical_servers/<server_ID>
```

Request body:

```
{
   "action":"power_off"
}
```

1.6.3. Immediately powering off a physical server

To immediately power off a specific physical server, send a POST request using the following request URL and JSON request body.

```
POST /api/physical_servers/<server_ID>
```

Request body:

```
{
   "action":"power_off_now"
}
```

1.6.4. Restarting a physical server

To shut down the operating system and restart a specific physical server, send a POST request using the following request URL and JSON request body.

```
POST /api/physical_servers/<server_ID>
```

Request body:

```
{
   "action":"restart"
}
```

1.6.5. Immediately restarting a physical server

To immediately restart a specific physical server, send a POST request using the following request URL and JSON request body.

```
POST /api/physical_servers/<server_ID>
```

-

Request body:

```
{
   "action":"restart_now"
}
```

1.6.6. Changing the location LED state on a physical server to blinking

To change the location-LED state to blinking on a specific server, send a POST request using the following request URL and JSON request body.

```
POST /api/physical_servers/<server_ID>
```

Request body:

```
{
   "action":"blink_loc_led"
}
```

1.6.7. Performing an action on multiple physical servers

To perform an action on more than one physical server, send a POST request using the following request URL and JSON request body.

```
POST /api/physical_servers
```

Request body:

```
{
   "action":"<action>",
   "resources": [{
      "href":"<server_ID>"
   },
   {
      "href":"<server_ID>"
   }]
}
```

1.7. RETRIEVING PHYSICAL CHASSIS DETAILS

To retrieve information for all physical chassis, use the following request URL:

```
GET /api/physical_chassis
```

A JSON response with the following keys is returned.

```
actions (list of actions that can be used)
count (number of physical chassis)
name (URI name)
pages (number of pages)
resources (list of URLs for each physical chassis)
```

```
subcount (number of physical chassis that were returned)
```

Examples

- To retrieve all information about the physical chassis, use the **?expand=resources** query parameter, for example:

  ```
  GET /api/physical_chassis?expand=resources
  ```

- To retrieve specific information about the physical chassis (such as name and ID), use the **?expand=resources&attributes=<attribute_list>** query parameters, for example:

  ```
  GET /api/physical_chassis?expand=resources&attributes=id,name
  ```

- To retrieve information such as the switches, storage systems, and servers in a chassis, use the **?expand=resources&attributes=<attribute_list>** query parameters, for example:

  ```
  GET /api/physical_chassis?
  expand=resources&attributes=physical_servers,physical_storages,hardw
  are.connected_physical_switches
  ```

- To retrieve information about only a specific physical chassis, use the **?expand=resources&filter[]=name=<chassis_name>** query attributes on all physical chassis resources, for example:

  ```
  GET /api/physical_chassis?
  expand=resources&filter[]=name=SN%23Y034BG16E060
  ```

1.8. RETRIEVING PHYSICAL CHASSIS DETAILS BY ID

To retrieve information for a specific physical chassis, use the following request URL:

```
GET /api/physical_chassis/<chassis_ID>
```

A JSON response with the following keys is returned.

```
href
id
ems_id
uid_ems
ems_ref
physical_rack_id
name
vendor
type
location_led_state
health_state
overall_health_state
management_module_slot_count
switch_slot_count
fan_slot_count
```

```
blade_slot_count
powersupply_slot_count
actions
```

Examples

- To retrieve information about a specific physical chassis (such as name and ID), use the **?expand=resources&attributes=<attribute_list>** query parameters, for example:

  ```
  GET /api/physical_chassis/<chassis_ID>?
  expand=resources&attributes=id,name
  ```

- To retrieve information, such as the switches, storage systems, and servers that are in a specific physical chassis, use the **?expand=resources&attributes=<attribute_list>** query parameters, for example:

  ```
  GET /api/physical_chassis/<chassis_ID>?
  expand=resources&attributes=physical_servers,physical_storages,hardw
  are.connected_physical_switches
  ```

1.9. PERFORMING PHYSICAL CHASSIS ACTIONS

You can use the REST API to perform management tasks on a physical chassis, such as changing the location LED state.

Note: To perform actions on physical chassis, you must create a basic authentication using your preferred tool or library, and enter your user name and password to access the CloudForms REST API.

1.9.1. Changing the location LED state on a physical chassis to blinking

To change the location LED state to blinking on a specific chassis, send a POST request using the following request URL and JSON request body.

```
POST /api/physical_chassis/<chassis_ID>
```

Request body:

```
{
   "action":"blink_loc_led"
}
```

1.9.2. Performing an action on multiple physical chassis

To perform an action on more than one physical chassis, send a POST request using the following request URL and JSON request body.

```
POST /api/physical_chassis
```

Request body:

```
{
   "action":"<action>",
```

```
   "resources": [{
      "href":"<chassis_ID>"
   },
   {
      "href":"<chassis_ID>"
   }]
}
```

1.10. RETRIEVING PHYSICAL RACK DETAILS

To retrieve information for all physical racks, use the following request URL:

```
GET /api/physical_racks
```

A JSON response with the following keys is returned.

```
actions (list of actions that can be used)
count (number of physical racks)
name (URI name)
pages (number of pages)
resources (list of URLs for each physical rack)
subcount (number of physical racks that were returned)
```

Examples

- To retrieve all information about the physical racks, use the **?expand=resources** query parameter, for example:

  ```
  GET /api/physical_racks?expand=resources
  ```

- To retrieve specific information about the physical racks (such as name and ID), use the **?expand=resources&attributes=<attribute_list>** query parameters, for example:

  ```
  GET /api/physical_racks?expand=resources&attributes=id,name
  ```

- To retrieve information about only a specific physical rack, use the **?expand=resources&filter[]=name=<rack_name>** query attributes on all physical rack resources, for example:

  ```
  GET /api/physical_racks?expand=resources&filter[]=name=rack-2
  ```

1.11. RETRIEVING PHYSICAL RACK DETAILS BY ID

To retrieve information for a specific physical rack, use the following request URL:

```
GET /api/physical_racks/<rack_ID>
```

A JSON response with the following keys is returned.

```
actions
ems_id
ems_ref
```

```
href
id
name
uid_ems
```

Examples

- To retrieve information about a specific physical rack (such as name and ID), use the **?expand=resources&attributes=<attribute_list>** query parameters, for example:

  ```
  GET /api/physical_racks/<rack_ID>?
  expand=resources&attributes=id,name
  ```

1.12. RETRIEVING PHYSICAL STORAGE DETAILS

To retrieve information for all physical storage systems, use the following request URL:

```
GET /api/physical_storages
```

A JSON response with the following keys is returned.

```
actions (list of actions that can be used)
count (number of physical storage systems)
name (URI name)
pages (number of pages)
resources (list of URLs for each physical storage system)
subcount (number of physical storage systems that were returned)
```

Examples

- To retrieve all information about the physical storage systems, use the **?expand=resources** query parameter, for example:

  ```
  GET /api/physical_storages?expand=resources
  ```

- To retrieve specific information about the physical storage systems (such as name and ID), use the **?expand=resources&attributes=<attribute_list>** query parameters, for example:

  ```
  GET /api/physical_storages?expand=resources&attributes=id,name
  ```

- To retrieve information such as canisters and physical disks associated with physical storage systems, use the **?expand=resources&attributes=<attribute_list>** query parameters, for example:

  ```
  GET /api/physical_storages?
  expand=resources&attributes=canisters,physical_disks
  ```

- To retrieve information about only a specific physical storage system, use the **?expand=resources&filter[]=name=<storage_name>** query attributes on all physical storage resources, for example:

  ```
  GET /api/physical_storages?expand=resources&filter[]=name=SeagateV1-
  S3200-DM
  ```

1.13. RETRIEVING PHYSICAL STORAGE DETAILS BY ID

To retrieve information for a specific physical storage system, use the following request URL:

```
GET /api/physical_storages/<storage_ID>
```

A JSON response with the following keys is returned.

```
access_state
actions
canister_slots
drive_bays
ems_id
ems_ref
enclosures
health_state
href
id
name
overall_health_state
physical_chassis_id
physical_rack_id
type
uid_ems
```

Examples

- To retrieve information about a specific physical storage system (such as name and ID), use the **?expand=resources&attributes=<attribute_list>** query parameters, for example:

  ```
  GET /api/physical_storages/<storage_ID>?
  expand=resources&attributes=id,name
  ```

- To retrieve information, such as canisters and physical disks that are associated with a specific physical storage system, use the **?expand=resources&attributes=<attribute_list>** query parameters, for example:

  ```
  GET /api/physical_storages/<storage_ID>?
  expand=resources&attributes=canisters,physical_disks
  ```

1.14. RETRIEVING PHYSICAL SWITCH DETAILS

To retrieve information for all physical switches, use the following request URL:

```
GET /api/physical_switches
```

A JSON response with the following keys is returned.

```
actions (list of actions that can be used)
count (number of physical switches)
name (URI name)
```

```
pages (count of pages)
resources (list of URLs for each physical switch)
subcount (number of physical switches that were returned)
```

Examples

- To retrieve all information about the physical switches, use the **?expand=resources** query parameter, for example:

  ```
  GET /api/physical_switches?expand=resources
  ```

- To retrieve specific information about the physical switches (such as name and ID), use the **?expand=resources&attributes=<attribute_list>** query parameters, for example:

  ```
  GET /api/physical_switches?expand=resources&attributes=id,name
  ```

- To retrieve information such as ports on physical switches, use the **?expand=resources&attributes=<attribute_list>** query parameters, for example:

  ```
  GET /api/physical_switches?
  expand=resources&attributes=physical_network_ports
  ```

- To retrieve information such as physical devices connected to physical switches, use the **?expand=resources&attributes=<attribute_list>** query parameters, for example:

  ```
  GET /api/physical_switches?
  expand=resources&attributes=connected_components
  ```

- To retrieve information about only a specific physical switch, use the **?expand=resources&filter[]=name=<switch_name>** query attributes on all physical switch resources, for example:

  ```
  GET /api/physical_switches?expand=resources&filter[]=name=IO Module
  02
  ```

1.15. RETRIEVING PHYSICAL SWITCH DETAILS BY ID

To retrieve information for a specific physical switch, use the following request URL:

```
GET /api/physical_switches/<switch_ID>
```

A JSON response with the following keys is returned.

```
actions
allow_promiscuous
ems_id
forged_transmits
health_state
href
id
mac_changes
mtu
name
```

```
ports
power_state
shared
switch_uuid
type
uid_ems
```

Examples

- To retrieve information about a specific physical switch (such as name and ID), use the **?expand=resources&attributes=<attribute_list>** query parameters, for example:

  ```
  GET /api/physical_switches/<switch_ID>?
  expand=resources&attributes=id,name
  ```

- To retrieve information such as physical devices that are connected to a specific physical switch, use the **?expand=resources&attributes=<attribute_list>** query parameters, for example:

  ```
  GET /api/physical_switches/<switch_ID>?
  expand=resources&attributes=connected_components
  ```

- To retrieve information such as ports on a specific physical switch, use the **?expand=resources&attributes=<attribute_list>** query parameters, for example:

  ```
  GET /api/physical_switches/<switch_ID>?
  expand=resources&attributes=physical_network_ports
  ```

1.16. PERFORMING PHYSICAL SWITCH ACTIONS

You can use the REST API to perform management tasks on a physical switch, such as restarting the switch.

Note: To perform actions on physical switch, you must create a basic authentication using your preferred tool or library, and enter your user name and password to access the CloudForms REST API.

1.16.1. Restarting a physical switch

To restart a specific switch, send a POST request using the following request URL and JSON request body.

```
POST /api/physical_switches/<switch_ID>
```

Request body:

```
{
   "action":"restart"
}
```

1.16.2. Performing an action on multiple physical switches

To perform an action on more than one physical switch, send a POST request using the following request URL and JSON request body.

```
POST /api/physical_switches
```

Request body:

```
{
   "action":"<action>",
   "resources": [{
      "href":"<switch_ID>"
   },
   {
      "href":"<switch_ID>"
   }]
}
```