Contents

Leostream Restful API Introduction

The Leostream RESTful API provides an alternative way to create, list, and update the Connection Broker configuration. You can use the RESTful API with programming languages to automate redundant tasks. The calls are made using a JSON representation and the data is returned as JSON formatted text.

The API calls may contain an ID parameter and/or return an ID field. The ID is a unique number assigned by the Connection Broker to each record in the database. API calls with the ID parameter perform actions against that record in the database. For example, a RESTful API call for a Pool with the ID parameter references the pool.id field in the Connection Broker database. The following table lists the RESTful API call, and the database table and field that it corresponds to.

RESTful API Object	Database Table	Database field
Centers	server	server.id
Pools	pool	pool.id
Policy.pool_assignments	policy_assignment	policy_assignment.id
Users	users	users.id

The RESTful API calls may return additional fields for the specified ID in the database. The Connection Broker database schema provides a description of these fields. The schema is available on the **> System > XML API** page. You can directly access the schema from the `https://<enter-your-Connection=Broker-address>/download/account_db.html` link.

Guidelines for RESTful calls

The RESTful API makes requests to the Connection Broker web server with a URL formatted with 4 or 5 of the following components:

1. The protocol, HTTP or HTTPS

2. The Connection Broker server name or IP address
3. The request path
4. The API call
5. The Connection Broker object ID, if required

The first three components of the API calls are static throughout the program. The fourth and fifth component change, depending on the API call. For example:

- All API calls begin with `http://<enter-your-Connection=Broker-address>/rest`
- Then, for the login API call which is `/v1/session/login`
- The full URL is `http://<enter-your-Connection=Broker-address>/rest/v1/session/login`

A sample request to list a pool is constructed as follows:

- The API call begins with `http://<enter-your-Connection=Broker-address>/rest`
- The pool API call is /v1/pools/{ID}
- The full pool URL is `http://<enter-your-Connection=Broker-address>/rest/v1/pools/{ID}`

RESTful API Component Status

In Release 9.1, the RESTful API became a required component of the Connection Broker. You can check the status of the Leostream components using the 'leo' menu on the Connection Broker server.

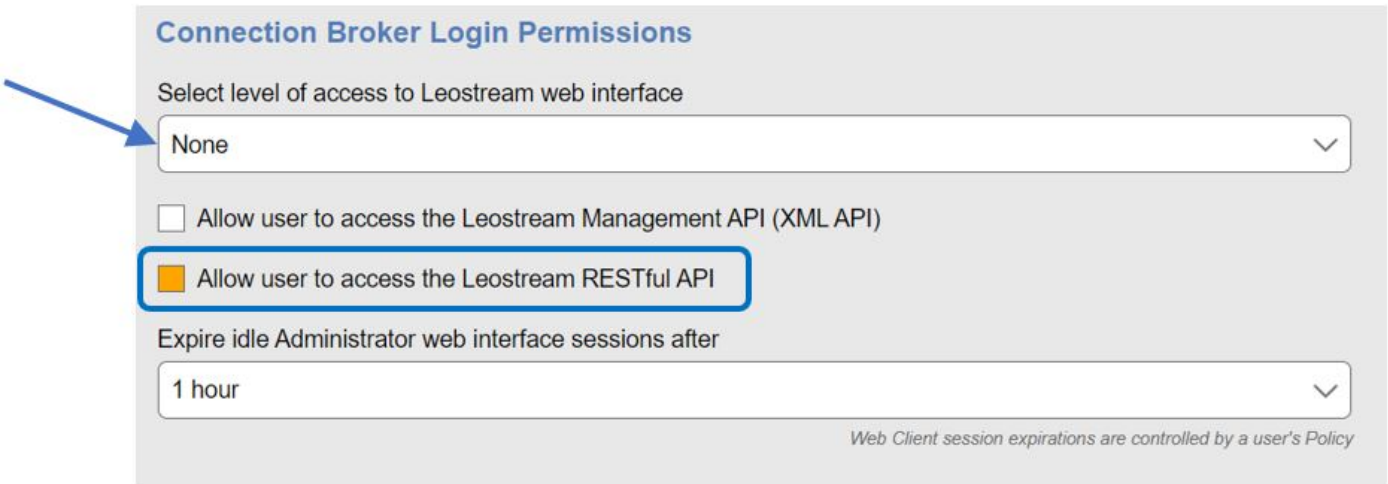
1. Log into the Connection Broker Linux server
2. Type su - leo at the command prompt to switch to the leo user
3. Select Power from the Main Menu
4. Select Status from the Power Menu

You can check the status of all the Connection Broker components from the command line using `/home/leo/app/control.pl -status`

User authorization to the RESTful API

Calling the RESTful API first requires a Connection Broker login. The user logging in must have the appropriate permissions to access the Leostream RESTful API and to perform the actions in the Connection Broker that will be executed by the API. Leostream recommends using a separate user ID for calling the API, instead of using one of your standard admininstrator IDs.

Permissions are granted in the Connection Broker on the **> Configuration > Roles** page. You must select the **Allow user to access the Leostream RESTful API** option in the role assigned to the user who will execute the RESTful API, as show in the following figure.



If the role is used by users who only execute the RESTful API, you can disable access to the Leostream Web interface by selecting **None** from the **Select level of access to Leostream web interface** drop-down menu, as shown in the previous figure.

The remainder of the role should then set appropriate permissions for the different objects in the Connection Broker. For example, if you plan to modify pools via the API, ensure that the role sets the permission for the **Pools** page to **Full** access.

User authentication to the RESTful API

The Leostream RESTful API uses Bearer authentication with a security token stored in the RESTful API user's session ID.

Login

The `login` RESTful API call authenticates the RESTful API user and assigns their privileges. The call returns a session ID that is used as the Bearer in the header for all subsequent API calls. For example, the following code calls the `login` API and stores its result in the `response_data` variable.

Python example

```
response = requests.post(url=" http://CB.leostream.com/rest/v1/session/login", json={
    'user_login':'Louise', 'password':'MySuperSecretPassword'}, verify="false")
response_data = json.loads(response.text)
```

Corresponding RAW request

```
POST /rest/v1/session/login HTTP/1.1
Host: CB.leostream.com
Content-Type: application/json
Cache-Control: no-cache

{
    "user_login":"Louise",
    "password":"MySuperSecretPassword"
}
```

Evaluate login result

After logging in, in the following example, the `response_data` variable uses the Python library to format the results into a Python dictionary. The `response_data` object is used to obtain the status of the call and the SessionID (`sid`) used as the Bearer for the subsequent calls.

```
if response_data["status"] == 'OK':
    sessionID= "Bearer " + response_data["sid"]
```

This Session ID is added to the `HEADER` variable and used in the subsequent API calls, as shown in the following logout example.

Use session ID in subsequent call

Python example of logout

RESTful API Component Status

In Release 9.1, the RESTful API became a required component of the Connection Broker. You can check the status of the Leostream components using the 'leo' menu on the Connection Broker server.

1. Log into the Connection Broker Linux server
2. Type su - leo at the command prompt to switch to the leo user
3. Select Power from the Main Menu
4. Select Status from the Power Menu

You can also check the status of all the Connection Broker components from the command line using /home/leo/app/control.pl –status

Authorization

```
HEADERS = {'Authorization': sessionID}

response = requests.post(url= "http://CB.leostream.com/rest/v1/session/logout", headers= {'Authorization': se
response_data = json.loads(response.text)

if response_data["status"] == 'OK':
    print(f'Logout is successful.\n')
else:
    print(f'Logout call failed with error {response_data["status"]}.')
```

Example of raw Pools.Get request

```
GET /rest/v1/pools/31 HTTP/1.1
Host: CB.leostream.com
Content-Type: application/json
Accept: application/json
Authorization: Bearer POcmSS3wT7msRWTDwelmzgiIcuO4mpQw6dbgBvVQRA
Cache-Control: no-cache
```

Example of raw logout request

```
POST /rest/v1/session/logout HTTP/1.1
Host: CB.leostream.com
Content-Type: application/json
Authorization: Bearer POcmSS3wT7msRWTDwelmzgiIcuO4mpQw6dbgBvVQRA
Cache-Control: no-cache
```

Leostream RESTful API Methods

Method: session.login

Authenticate a user session to access the Leostream Management APIs.

Request

HTTP Method	Relative Request URI
POST	/session/login

Request Body Parameters

Request Parameter	Type	Description
user_login	string(64)	User login name.
password	string(100)	User login password.

Optional Parameter	Type	Description
domain	string(255)	Domain for user login.

Responses

This operation has the following responses.

200 OK

The operation was successful.

Response body

If successful, the response body contains data with the following structure:

```
{
  "sid": "{new session ID value}"
}
```

Successful return

Field	Value	Description
sid	`{sid}`	Unique identifier associated with this session.

Errors

400 Bad Request

The operation was not successful: the request is incorrect. The response contains the following fields: 'error_status' and 'error_message'.

Field 'error_status' is BAD_DATA

Returned if the request is not valid. See the 'error_message' field for details. If the request is missing required parameters, the 'missing_fields' field contains a list of missing parameters. If a parameter specifies an invalid value, the 'errors' field contains additional information on the error.

401 Unauthorized

Field 'error_status' is INVALID_CREDENTIALS

Invalid credentials. See the 'error_message' field for details.

403 Forbidden

Field 'error_status' is INSUFFICIENT_PERMISSIONS

Access to API is not enabled in this user's role. See the 'error_message' field for details.

Method: session.logout

Delete user session (log out the user).

Request

HTTP Method	Relative Request URI
POST	/session/logout

Request Body Parameters

No parameters other than the Authorization header.

Responses

This operation has the following responses.

200 OK

The operation was successful.

Successful return

Field	Type	Description
message	string	Message about successful logout

401 Unauthorized

The operation was not successful.

Field 'error_status' is UNAUTHORIZED

No correct Authorization header provided.

Method: users.insert

Add a user to the Connection Broker database. This user can be either a local user or a user from an authentication server

Request

HTTP Method	Relative Request URI
POST	/users

Request Body Parameters

Request Parameter	Type	Description
user_login	string(64)	User login name.
name	string(100)	The user's display name.
is_local	boolean	When '1' the user is authenticated locally by the Connection Broker. When '0' then specify 'auth_server', the authentication server that contains the user.

Optional Parameter	Type	Description
password	string(100)	User login password (required for 'local' user).
auth_server	string(100)	Authentication server that contains the user (required for non-local user)
role_name	string(100)	The role to assign to the user (defaults to "User")
policy_name	string(100)	The policy to assign to the user (defaults to "Default")
notes	string(4000)	Optional notes to store with the user record.

Responses

This operation has the following responses.

201 Created

The Create operation was successful.

Successful return

Field	Type	Description
stored_data	number	The user's unique identifier

400 Bad Request

The operation was not successful: the request is incorrect. Should contain the following fields: 'error_status' and 'error_message'.

Field 'error_status' is BAD_DATA

Returned if the request is not valid. See the 'error_message' field for details. If the request is missing required parameters, the 'missing_fields' field contains a list of missing parameters. If a parameter specifies an invalid value, the 'errors' field contains additional information on the error.

409 Conflict

Field 'error_status' is EXISTS_ALREADY

The operation was not successful: this 'user_login' exists already

---

Method: centers.list

Retrieves the list of Center currently defined in the Connection Broker.

Request

HTTP Method	Relative Request URI
GET	/centers

Request Body Parameters

No parameters.

Responses

This operation has the following responses.

200 OK

The operation was successful.

Successful return

A list of Centers. Each record contains general information about the Center.

Response body

If successful, the response body contains data with a structure similar to the following:


```
[
    {
        "id": 2,
        "name": "Some Local Center",
        "online": 1,
        "status": "Online",
        "type": "vcenter",
        "type_label": "VMware vSphere and vCenter Server",
```

```
        "version": "VMware vCenter Server 6.5.0 build-5973321"
    },
    {
        "id": 4,
        "name": "Azure Center",
        "online": 1,
        "status": "Online",
        "type": "azure",
        "type_label": "Microsoft Azure",
        "version": "Microsoft Azure"
    }
]
```

Method: centers.get

Returns the specified Center resource.

Request

HTTP Method	Relative Request URI
GET	/centers/`{id}`

Request Path Parameters

Request Parameter	Type	Description
`{id}`	number	Center ID value.

Request Body Parameters

No parameters.

200 OK

The operation was successful.

Successful return

A list of Centers. Each record contains detailed information about the Center.

Response body

If successful, the response body contains data with a structure similar to the following:


```
{
    "id": 4,
    "name": "Azure Center",
    "type": "azure",
    "resource_groups": [
        "Other_Group",
        "Some_Group"
    ],
    "images": [
        {
            "id": 88,
            "name": "Ubuntu18Image",
```

```
            "resource_group": "Some_Group",
            "vmi": "/subscriptions/88888888-8888-8888-8888-888888888888/resourceGroups/Some_Group/providers/M
        }
    ],
    "instance_sizes": [
        "Basic_A0",
        "Basic_A1",
        "Basic_A2",
        "Basic_A3",
        "Basic_A4",
        "Standard_A0",
        "Standard_A1",
        "..."
    ],
    "security_groups": [
        {
            "group": "Some_Group",
            "id": "/subscriptions/88888888-8888-8888-8888-888888888888/resourceGroups/Some_Group/providers/Mi
            "location": "eastus",
            "name": "my-test-nsg"
        },
        {
            "group": "Some_Group",
            "id": "/subscriptions/88888888-8888-8888-8888-888888888888/resourceGroups/Some_Group/providers/Mi
            "location": "eastus",
            "name": "ss-ubuntu-16-nsg"
        }
    ],
    "vnets": {
        "Other_Group": [
            {
                "addresses": [
                    "10.1.0.0/16"
                ],
                "id": "/subscriptions/88888888-8888-8888-8888-888888888888/resourceGroups/Other_Group/provide
                "name": "SomeNetworking",
                "subnets": [
                    "default"
                ]
            }
        ],
        "Some_Group": [
            {
                "addresses": [
                    "10.2.0.0/16"
                ],
                "id": "/subscriptions/88888888-8888-8888-8888-888888888888/resourceGroups/Some_Group/provider
                "name": "dev-net-2",
                "subnets": [
                    "default"
                ]
            },
            {
                "addresses": [
                    "10.0.0.0/24"
                ],
                "id": "/subscriptions/88888888-8888-8888-8888-888888888888/resourceGroups/Some_Group/provider
                "name": "Some_Group-vnet",
```

```
                "subnets": [
                    "default"
                ]
            }
        ]
    }
}
```

Response body of Active Directory Center

If successful, the response body contains data with a structure similar to the following:


```
{
    "active": 1,
    "center_definition": {
        "allow_rogue": 0,
        "allow_rogue_policy_id": 1,
        "continuous_autotag": 0,
        "enable_syslog": 1,
        "enable_syslog_relay": 0,
        "flavor": "A",
        "gateway_id": 0,
        "init_unavailable": 0,
        "ip": "",
        "ldap_filter": "",
        "ldap_subtree": "DC=example, DC=com",
        "max_remote_desktop_sessions": 0,
        "name": "AD Center",
        "new_as_deletable": 0,
        "notes": "",
        "offer_vms": 1,
        "poll_agent_interval": null,
        "poll_interval": 0,
        "poll_power_state_interval": 0,
        "remote_authentication_id": 1,
        "rpc_port": 389,
        "short_hostname_lookup": 0,
        "syslog_forward_to_addr": "",
        "type": "active_directory",
        "type_label": "Active Directory",
        "vc_cluster": null,
        "vc_custom_fields": [],
        "vc_datacenter": "",
        "vc_domain": "",
        "vc_name": "",
        "vc_password": "********",
        "vc_project_domain": "",
        "vc_url": "AD.example.com",
        "vc_version": "",
        "vm_id": null
    },
    "center_info": {
        "agent_instance_uuid": "",
        "agent_token": "",
        "agent_uuid": "",
        "agent_version": "",
        "ldap_attributes": [
            "cn",
            "codePage",
```

```
            "countryCode",
            "dNSHostName",
            "description",
            "displayName",
            "distinguishedName",
            "instanceType",
            "location",
            "managedBy",
            "memberOf",
            "msDFSR-ComputerReferenceBL",
            "name",
            "networkAddress",
            "objectCategory",
            "objectClass",
            "objectGUID",
            "operatingSystem",
            "operatingSystemServicePack",
            "operatingSystemVersion",
            "primaryGroupID",
            "rIDSetReferences",
            "sAMAccountName",
            "sAMAccountType",
            "serverReferenceBL",
            "servicePrincipalName"
        ],
        "ldap_attributes_updated": "2021-09-24 14:31:32",
        "os": "",
        "os_version": "",
        "type": "active_directory"
    },
    "created": "2021-09-24 14:30:33",
    "id": 9,
    "images": [],
    "name": "AD Center",
    "needs_upgrade": 0,
    "online": 1,
    "pcoip_ssh_password": "********",
    "product_id": 0,
    "status": 1,
    "status_label": "Online",
    "updated": "2021-09-24 14:31:32",
    "uuid": "00000000-0000-0000-0000-000000000000"
}
```


404 Not Found

Indicates that the requested Center ID does not exist.

Method: pools.list

Retrieves the list of Pool resources currently defined in the Connection Broker.

Request

HTTP Method	Relative Request URI
GET	/pools

Request Query Parameters

Request Parameter	Type	Description
as_tree	boolean	Specifies whether to show the list as a tree.
id	number	Pool ID value of the tree root, processed only if the parameter `as_tree` equals '1'.

Responses

This operation has the following responses.

200 OK

The operation was successful.

Successful return

A list of Pools. Each record contains general information about a Pool.

Response body

If successful, the response body contains data with a structure similar to the following:

Pools.list default


```
[
    {
        "assigned_vm": 0,
        "available_vm": 115,
        "id": 1,
        "name": "All Desktops",
        "parent_pool_id": 0,
        "total_agent_running": 0,
        "total_connected": 0,
        "total_logged_in": 0,
        "total_vm": 115,
        "total_vm_running": 14,
        "total_vm_stopped": 98,
        "total_vm_suspended": 0,
        "unavailable_vm": 0
    },
    {
        "assigned_vm": 0,
        "available_vm": 44,
        "id": 5,
        "name": "All Linux Desktops",
        "parent_pool_id": 1,
        "total_agent_running": 0,
        "total_connected": 0,
        "total_logged_in": 0,
        "total_vm": 44,
        "total_vm_running": 8,
        "total_vm_stopped": 34,
        "total_vm_suspended": 0,
        "unavailable_vm": 0
```

```
    },
    {
        "assigned_vm": 0,
        "available_vm": 33,
        "id": 4,
        "name": "All Windows Desktops",
        "parent_pool_id": 1,
        "total_agent_running": 0,
        "total_connected": 0,
        "total_logged_in": 0,
        "total_vm": 33,
        "total_vm_running": 4,
        "total_vm_stopped": 29,
        "total_vm_suspended": 0,
        "unavailable_vm": 0
    }
]
```


Pools.list with a query string '?as_tree=1&id=5'


```
[
    {
        "id": "1",
        "level": 0,
        "name": "All Desktops",
        "children": [
            {
                "id": "5",
                "level": 1,
                "name": "All Linux Desktops",
                "children": []
            },
            {
                "id": "4",
                "level": 1,
                "name": "All Windows Desktops",
                "children": []
            },
            {
                "id": "7",
                "level": 1,
                "name": "API Azure Pool",
                "children": [
                    {
                        "id": "21",
                        "level": 2,
                        "name": "API Azure Provision Pool",
                        "children": []
                    }
                ]
            },
            {
                "id": "6",
                "level": 1,
                "name": "AWS Pool (N. Virginia)",
                "children": []
            },
            {
                "id": "8",
```

```
                "level": 1,
                "name": "vCenter 6 Pool",
                "children": []
            }
        ]
    }
]
```

Method: pools.get

Returns the specified Pool resource.

Request

HTTP Method	Relative Request URI
GET	/pools/{id}

Request Path Parameters

Request Parameter	Type	Description
{id}	number	Pool ID value.

Request Body Parameters

No parameters.

Response

This operation has the following responses.

200 OK

The operation was successful.

Successful return

A Pool object in JSON representation.

Response body

If successful, the response body contains data with a structure similar to the following:

```
{
    "id": 8,
    "name": "Azure Provision Pool",
    "display_name": "",
    "notes": "",
    "read_only": 0,
    "running_vms_threshold": 0,
    "domain": {
        "domain_join": 0
    },
    "pool_definition": {
        "attribute_join": "A",
        "attributes": [
            {
```

```
                "condition_type": "ne",
                "text_to_match": "13",
                "vm_table_field": "num_disks"
            },
            {
                "condition_type": "bw",
                "text_to_match": "azure-new-vm",
                "vm_table_field": "name"
            }
        ],
        "never_rogue": 0,
        "parent_pool_id": 7,
        "restrict_by": "A"
    },
    "log": {
        "information_threshold": 0,
        "warning_threshold": 0,
        "error_threshold": 0,
        "retain_history": {
            "pool_history_age": 0,
            "pool_history_interval": 0
        }
    },
    "provision": {
        "center": {
            "admin": "azureadmin",
            "create_public_ip": null,
            "disk_size": null,
            "id": 4,
            "instance_size": "Basic_A2",
            "method": {
                "desc": "[same as template]",
                "type": "T"
            },
            "name": "Azure Center",
            "password": "VeryGood#P@ssw0rd",
            "resource_group": "Some_Group",
            "security_group": null,
            "sub_net": "default",
            "type": "azure",
            "use_single_group": 1,
            "vnet": "Some_Group:Some_Group-vnet"
        },
        "image_vm_id": 88,
        "limits_enforce": 0,
        "mark_deletable": "1",
        "mark_unavailable": "0",
        "max": 4,
        "notification_url": "",
        "on_off": 1,
        "threshold": 2,
        "time_limits": [],
        "vm_name": "azure-new-vm-{SEQUENCE}",
        "vm_name_next_value": 12
    },
    "vms_list": [
        {
            "display_name": "",
```

```
            "hda_status": "",
            "id": 96,
            "name": "azure-new-vm-p-6",
            "out_of_service": 0,
            "status": 22,
            "user_id": null
        },
        {
            "display_name": "",
            "hda_status": "",
            "id": 97,
            "name": "azure-new-vm-p-7",
            "out_of_service": 0,
            "status": 22,
            "user_id": 11
        }
    ]
}
```


Errors

404 Not Found

Indicates that the requested Pool ID does not exist.

Method: pools.insert

Add a pool to the Connection Broker. Specify the appropriate additional parameters based on how the pool is defined, as determined by the `pool_definition.restrict_by` parameter.

Request

HTTP Method	Relative Request URI
POST	/pools

Request Body Parameters

Request Parameter	Type	Description
name	string	The pool's name.
pool_definition	object	Container for parameters related to Pool definition.
pool_definition.restrict_by	char	Specifies the way Pool is defined: *'A' - Pool is defined by Pool attributes;* *'T' - Pool is defined by Tags;* *'C' - Pool is defined by Centers;* *'E' - Pool is defined by vCenter Server Hosts;* *'L' - Pool is defined by vCenter Server Clusters;* *'V' - Pool is defined by vCenter Server Resource Pools;* *'H' - Pool is defined by selection from parent Pool.* *'Z' - Pool is defined by LDAP attributes;*

Optional Body Parameters

First Level Parameter	Type	Description
display_name	string	The pool's display name.
notes	string	Optional notes to store with the pool record.
running_vms_threshold	number	Threshold to power on desktops when the number of running, unassigned desktops drops below.
domain	object	Container for parameters related to Domain Join.
log	object	Container for parameters related to Logging and Reporting.
provision	object	Container for parameters related to Provisioning.

"pool_definition" Parameters	Type	Description
restrict_by	char	Mandatory. Specifies the way Pool is defined.
vm_server_distribution	number	"Distribute new desktop assignments" if restrict_by is 'C'. *0 - "Evenly across all hosts";* *1 - "To center with most available desktops";* *2 - "To center with least number of assignments".*
never_rogue	boolean	"Associate all user notifications with assigned user" if restrict_by is 'C'.
center_ids[]	array	Array of Center ID values if restrict_by is 'C'.
parent_pool_id	number	ID of the parent Pool.
attribute_join	char	Specifies the way Pool attributes are joined: *'A' - match all of the attribute rules (AND);* *'O' - match any of the attribute rules (OR).*
attributes[]	array	Array container for Pool attributes (restrict_by is 'A') or for LDAP attributes (restrict_by is 'Z', requires Active Directory Centers).
attributes[].vm_table_field	string	Desktop attribute, mandatory for Pool attributes : *name - Name*; *display_name - Display name*; *windows_name - Machine name*; *ip - Hostname or IP address*; *partition_names - Disk partition name*; *partition_mount_points - Partition mount point*; *guest_os - Operating system*; *os_version - Operating system version*; *installed_protocols - Installed protocols*; *vc_memory_mb - Memory (in MB)*; *vc_num_cpu - Number of CPUs*; *vc_num_ethernet_cards - Number of NICs*; *num_disks - Number of disks*; *computer_model - Computer model*; *bios_serial_number - BIOS serial number*; *max_clock_speed - CPU speed (GHz)*; *notes - Notes*; *vc_annotation - vCenter Server "Notes"*; *tag_filter - Tags*; *server_id - Servers.*
attributes[].ad_attribute_field	string	Desktop attribute, mandatory for LDAP attributes, see possible values for an AD Center in `centers.get` response, field *ldap_attributes*
attributes[].text_to_match	string	Mandatory for attributes[]. Value for the rule:
attributes[].condition_type	string	Mandatory for attributes[]. A conditional for the rule: *ip - "matches (CIDR notation)";* *eq - "is equal to";* *ne - "is not equal to";* *gt - "is greater than";* *lt - "is less than";* *ct - "contains";*

"pool_definition" Parameters	Type	Description
		nc - "does not contain"; *bw - "begins with";* *ew - "ends with".*

"domain" Parameters	Type	Description
domain_join	boolean	Specifies whether to join virtual machine to a domain.
domain_join_ou	string	Organizational Unit for domain join.
domain_join_hostname_as_vm_name	boolean	Specifies whether to set desktop hostname to virtual machine name.
pool_join_ad_groups[]	array	Array container of the groups to which the desktop should be added when a desktop joins a domain.
pool_join_ad_groups[].group_dn	string	The distinguishedName of a group to join.
remote_authentication	object	Container for parameters related to Domain.

"domain.remote_authentication" Parameters	Type	Description
domain_join_remote_authentication_id	number	ID of Domain.

"log" Parameters	Type	Description
information_threshold	number	Threshold. Log as Information if the number of unassigned desktops drops below.
warning_threshold	number	Threshold. Log as Warning if the number of unassigned desktops drops below.
error_threshold	number	Threshold. Log as Error if the number of unassigned desktops drops below.
retain_history	object	Container for parameters of track historical pool assignments and connections.
retain_history.pool_history_interval	number	Specifies interval in hours to sample data.
retain_history.pool_history_age	number	Specifies for how many days to retain data.

Provisioning Parameters

Provisioning parameters depends on what Centers are defined in the Connection Broker and which sets of values in every Center type (e.g. Azure, AWS, etc.) are defined. The following is a list of general Provisioning parameters.

"provision" Parameters	Type	Description
image_vm_id	number	Broker's ID of the image.
threshold	number	Threshold. Start provisioning when unassigned desktops in pool drops below.
max	number	Maximum value to stop provisioning when total desktops in pool reaches it.
provision_on_off	boolean	Specifies whether provisioning is enabled.
vm_name	string	Virtual machine name. Dynamic tags can be used, e.g. {SEQUENCE}.
vm_name_next_value	number	Sequence number for virtual machine name.
notification_url	string	Notification URL. It will be requested when provisioning is triggered.
mark_deletable	boolean	Specifies whether to initialize newly-provisioned desktops as "deletable".
mark_unavailable	boolean	Specifies whether to initialize newly-provisioned desktops as "unavailable".
limits_enforce	boolean	Specifies whether time-based provisioning limits

"provision" Parameters	Type	Description
		are enabled.
time_limits[]	array	Array container for time-based provisioning limits.
time_limits[].provision_time_day	number	Day of the week of the limit. An integer value between 0 and 6 (0 - Monday, 6 - Sunday).
time_limits[].provision_time_start	string	Start time in the "hh:mm:ss" format, e.g. "17:30:00".
time_limits[].provision_time_stop	number	Stop time in the "hh:mm:ss" format, e.g. "17:30:00".
time_limits[].provision_time_threshold	number	Threshold. Start provisioning when unassigned desktops in pool drops below.
time_limits[].provision_time_max_size	number	Maximum value to stop provisioning when total desktops in pool reaches it.
center	object	Container for parameters related to specific Center to provision in.

Request body example

```
{
    "display_name": "",
    "name": "API Center based Pool",
    "pool_definition": {
        "center_ids": [
            2
        ],
        "never_rogue": 0,
        "restrict_by": "C"
    },
    "provision": {
        "on_off": 0
    }
}
```

Response

This operation has the following responses.

201 Created

The Create operation was successful.

Successful return

A field 'stored_data', which contains a Pool object in JSON representation, similar to the response of the `pools.get` method.

Errors

400 Bad Request

The operation was not successful: the request is incorrect. The response contains the following fields: 'error_status' and 'error_message'.

Field 'error_status' is BAD_DATA

Typically indicates an invalid body JSON. See the 'error_message' field for additional details. The BAD_DATA error is also raised if the body JSON is valid but sets a value for the 'id' field.

Certain errors may return an additional 'errors' field which contains an array of data that does not pass validation. Typically, each item contains two fields:

- error_field - contains a target database field (not input JSON field)
- error_text - description of the error.

Note that this behavior (using low-level errors in the 'items') may be changed in future releases of the API.

409 Conflict

Field 'error_status' is CONFLICT

An error occurred during Pool creation. See the 'error_message' field for more details.

Method: pools.update

Update an existing pool. Specify different pool parameters depending on how the pool is defined. See the `pools.insert` method documentation for details.

Request

HTTP Method	Relative Request URI
PUT	/pools/`{id}`

Request Path Parameters

Request Parameter	Type	Description
`{id}`	number	Pool ID value.

Request Body Parameters

No required parameters. The optional Body Parameters for this method are the same as the parameters for the `pools.insert` method, plus an optional 'id' parameter. If specified, the 'id' parameter must be the same value as specified for the 'id' path parameter.

Request body example

```
{
    "id": 8,
    "display_name": "",
    "name": "Azure Provision Pool",
    "pool_definition": {
        "attribute_join": "A",
        "attributes": [
            {
                "condition_type": "ne",
                "text_to_match": "13",
                "vm_table_field": "num_disks"
            },
            {
                "condition_type": "bw",
                "text_to_match": "azure-new-vm",
                "vm_table_field": "name"
            }
        ],
        "never_rogue": 0,
        "parent_pool_id": 7,
        "restrict_by": "A"
    },
```

```
    "provision": {
        "image_vm_id": 88,
        "threshold": 4,
        "max": 4,
        "limits_enforce": 0,
        "mark_deletable": 1,
        "mark_unavailable": 0,
        "notification_url": "",
        "on_off": 1,
        "vm_name": "azure-new-vm-{SEQUENCE}",
        "center": {
            "type": "azure",
            "id": 4,
            "admin": "azureadmin",
            "password": "VeryGood#P@ssw0rd",
            "resource_group": "Some_Group",
            "use_single_group": 1,
            "instance_size": "Basic_A2",
            "vnet": "Some_Group:Some_Group-vnet",
            "sub_net": "default",
            "method": {
                "type": "T"
            }
        }
    }
}
```

Response

This operation has the following responses.

200 OK

The operation was successful.

Successful return

A field 'stored_data', which contains a Pool object in JSON representation, similar to the return of the `pools.get` method.

Errors

General Errors information

Errors of this method are the same as for the `pools.insert` method, except when an invalid Pool ID is specified (see below).

400 Bad Request (invalid Pool ID)

The operation was not successful: the request body has an invalid Pool ID value.

404 Not Found

Indicates that the specified Pool ID is valid but does not exist.

Method: pools.delete

Deletes an existing pool.

Request

HTTP Method	Relative Request URI
DELETE	/pools/{id}

Request Path Parameters

Request Parameter	Type	Description
{id}	number	Pool ID value.

Request Body Parameters

No parameters.

Response

This operation has the following responses.

204 No Content

The Delete operation was successful.

Successful return

No content.

Errors

403 Forbidden

Field 'error_status' is FORBIDDEN

User's role does not provide permission to take this action. See the 'error_message' field for details.

409 Conflict

Field 'error_status' is CONFLICT

An error occurred when attempting to delete the Pool. See the 'error_message' field for details.

Method: policies.list

Retrieves the list of Policies currently defined in the Connection Broker.

Request

HTTP Method	Relative Request URI
GET	/policies

Request Body Parameters

No parameters.

Responses

This operation has the following responses.

200 OK

The operation was successful.

Successful return

A list of Policies. Each record contains general information about a Policy.

Response body

If successful, the response body contains data with a structure similar to the following:

```
[
    {
        "id": 1,
        "name": "Default"
    },
    {
        "id": 4,
        "name": "Test CLI Policy"
    },
    {
        "id": 3,
        "name": "Test Policy"
    },
]
```

Method: policies.get

Returns the specified Policy resource.

Request

HTTP Method	Relative Request URI
GET	/policies/`{id}`

Request Path Parameters

Request Parameter	Type	Description
`{id}`	number	Policy ID value.

Request Body Parameters

No parameters.

Response

This operation has the following responses.

200 OK

The operation was successful.

Successful return

A Policy object in JSON representation. Some children objects (e.g. 'pool_assignments') contain data from other database tables and must be processed separately.

Response body

If successful, the response body contains data with a structure similar to the following:


```
{
    "alt_credentials": 0,
    "attribute_join": "A",
    "attributes": [
        {
            "attribute_type": "text",
            "condition_type": "ct",
            "text_attribute": "ss",
            "user_attribute": "text",
            "vm_attribute": "name"
        }
    ],
    "backup_ha_plan_power_control_id": 1,
    "backup_ha_plan_protocol_id": 1,
    "backup_ha_plan_release_id": 1,
    "backup_ha_pool_criteria_agent": 0,
    "backup_ha_pool_criteria_viewer": 0,
    "backup_ha_pool_id": 0,
    "filter_ad_refresh": 0,
    "ha_adjust_timezone": 0,
    "ha_auto_login": 0,
    "ha_confirm_power_state": 0,
    "ha_disconnect_logout_delay": -1,
    "ha_display_mode": "0",
    "ha_email_decline_shadowing": 0,
    "ha_email_shadowing": 0,
    "ha_enable_power_control": 0,
    "ha_enable_shadowing": 0,
    "ha_idle_disconnect_delay": -1,
    "ha_idle_lock_delay": -1,
    "ha_idle_logout_delay": -1,
    "ha_login_as": "R",
    "ha_logout_rogue": 0,
    "ha_offer_running_without_hda": 1,
    "ha_on_disconnect_url": "",
    "ha_on_logout_url": "",
    "ha_plan_power_control_id": 1,
    "ha_plan_protocol_id": 1,
    "ha_power_on": 1,
    "ha_retain_connection": 0,
    "ha_shadowing_filter": "1",
    "ha_shadowing_filter_json": {
                                  "filters": [
                                    {
                                      "ha_shadowing_filter_condition": "ct",
                                      "ha_shadowing_filter_value": "apod",
                                      "ha_shadowing_filter_attribute": "name"
                                    }
                                  ],
                                  "join": "A"
                                },
    "ha_start_if_stopped": 1,
    "ha_unverified_user_state": "L",
    "hide_hover_menu": 0,
    "id": 3,
    "launch_rdc": 0,
```

```
    "max_desktops": -1,
    "name": "Test Policy",
    "notes": "",
    "offer_empty_pools": 0,
    "pool_assignments": [
        {
            "id": 18,
            "pool_id": 9,
            "pool_name": "vCenter 6 hosts Pool"
        }
    ],
    "rogue_plan_power_control_id": 1,
    "rogue_plan_release_id": 1,
    "session_expire_delay": 2880,
    "session_expire_on_lock": 0,
    "session_start_url": "",
    "session_start_url_block": 0,
    "session_start_url_block_msg": "<b>Sign in error</b><br>Unable to initialize session",
    "session_start_url_cb": 0,
    "session_start_url_timeout": 5,
    "single_desktop_only": 0,
    "store_configurable_params": 0,
    "usb_enabled": 0,
    "usb_mode": "A",
    "vmware_view_servers": [],
    "web_client_new_window": 0
}
```

Method: policies.insert

Add a Policy to the Connection Broker. Specify different policy parameters depending on the way the policy is defined. After inserting a policy, use the `policies.pool_assignments.insert` method to add Desktop assignments from pools to the policy.

Request

HTTP Method	Relative Request URI
POST	/policies

Request Body Parameters

Request Parameter	Type	Description
name	string	The name of this policy

Optional Body Parameters

First Level Parameter	Type	Description
alt_credentials	boolean	Indicates whether the user will be prompted to enter alternate credentials for use when connecting to the desktop (PCoIP only)
attribute_join	char	How do the policy filter attributes get joined: *A = And* *O = Or (default)*

First Level Parameter	Type	Description
attributes[]	array	Array container for Policy attributes (see "attributes" table below)
backup_ha_plan_power_control_id	number	ID of the power control plan for hard-assigned backup pool
backup_ha_plan_protocol_id	number	ID of the protocol plan for hard-assigned backup pool
backup_ha_plan_release_id	number	ID of the release plan for hard-assigned backup pool
backup_ha_pool_criteria_agent	boolean	The method used to determine when to offer a desktop from the backup pool: - Leostream Agent is unreachable (also the default)
backup_ha_pool_criteria_viewer	boolean	The method used to determine when to offer a desktop from the backup pool: - Remote viewer port on desktop is unreachable
backup_ha_pool_id	number	ID of the hard-assigned backup pool
filter_ad_refresh	boolean	Indicates whether Active Directory attributes used in Policy filters should be refreshed with each login
ha_adjust_timezone	boolean	Indicates that the timezone of the destination desktop will be changed to the user's current timezone at login
ha_auto_login	boolean	Indicates that a console single sign-on (DCV, VNC, PCoIP, and HTML5) should be done on a hard-assigned machine
ha_confirm_power_state	boolean	Indicates whether to probe a machine for its power state when assigned
ha_disconnect_logout_delay	number	Number of minutes to wait after disconnect before logging user out of hard-assigned machine, or -1 for no logout
ha_display_mode	char	How to describe the offered hard-assigned machines: *0 = Desktop name (default)* *5 = Display name* *1 = Windows machine name*
ha_enable_shadowing	boolean	Indicates if session shadowing is enabled on a hard-assigned machine
ha_email_decline_shadowing	boolean	Indicates if the user that sent the invitation should be notified via email (no user choice for this option), if the invited user declines the invitation
ha_email_shadowing	boolean	Indicates if the session owner can decide if an email should be sent to the user being invited
ha_enable_power_control	number	How to perform power control actions of a hard-assigned machine if requested by a user from a thin client: *0 = Not allowed (default)* *1 = Power Control with Shutdown* *2 = Power Control with Power off*
ha_idle_disconnect_delay	number	Number of minutes to wait for the desktop being idle before disconnecting user from hard-assigned machine, or -1 for no logout
ha_idle_lock_delay	number	Number of minutes to wait for the desktop being idle before locking the hard-assigned machine, or -1 for no logout
ha_idle_logout_delay	number	Number of minutes to wait for the desktop being idle before logging user out of hard-assigned machine, or -1 for no logout
ha_login_as	char	Determines how domain is sent in protocol plans and single sign-on commands: *R = Defer to Role setting: role.login_as_local_user* *D = Always send user's domain (log in as a domain user)* *L = Always send desktop machine name (log in as a local user)*
ha_logout_rogue	boolean	Indicates that rogue users should be logged out on a hard-assigned machine
ha_offer_running_without_hda	boolean	Indicates that hard-assigned machines without a running Leostream Agent may be offered
ha_on_disconnect_url	string	A URL that is called when the desktop session is disconnected. May include dynamic tags.
ha_on_logout_url	string	A URL that is called when the user logs out of the desktop session. May include dynamic tags.

First Level Parameter	Type	Description
ha_plan_power_control_id	number	ID of the power control plan for hard-assigned machines
ha_plan_protocol_id	number	ID of the protocol plan for hard-assigned machines
ha_power_on	number	Determines which stopped and suspended machines are offered: *0 = Do not offer any stopped/suspended machines* *1 = Offer any stopped/suspended machine* *2 = Only only stopped/suspended machine if Leostream Agent is installed*
ha_retain_connection	boolean	Indicates if VNC or PCoIP sessions should be kept connected
ha_shadowing_filter	number	The method used to decide which users will be allowed to use session shadowing: *0: Included for all users* *1: Only included if user's AD record matches the **shadowing_filter_json** criteria*
ha_shadowing_filter_json	string	The user shadowing filter as a JSON data structure (either JSON or stringified JSON is accepted)
ha_start_if_stopped	boolean	Indicates whether to attempt to power on a machine if it's currently stopped/suspended
ha_unverified_user_state	char	A field indicating how to treat an unverified user state upon a client disconnect: *L = Logout* *D = Disconnect*
hide_hover_menu	boolean	Indicates whether Leostream Connect clients should hide their hover menu when a desktop is locked
launch_rdc	boolean	Indicates that a remote session should be automatically launched at sign in
max_desktops	number	The maximum number of desktops that a user can have at any one time, or -1 for no limit
notes	string	Notes
offer_empty_pools	boolean	Indicates whether clients should inform end-users that a pool has no resources available
rogue_plan_power_control_id	number	ID of the power control plan for rogue-assigned machines
rogue_plan_release_id	number	ID of the release plan for rogue-assigned machines
session_expire_delay	number	Number of minutes to wait before expiring a user's Connection Broker session (offers), or -1 for no delay
session_expire_on_lock	boolean	Indicates whether to expire a user's Connection Broker session (offers) when a remote desktop locks
session_start_url	string	A URL that is called whenever a new session is started with the Connection Broker. May include dynamic tags.
session_start_url_block	boolean	Indicates if the user's login should be blocked if the URL call fails
session_start_url_block_msg	string	If the session start URL call fails, and the policy is set to block logins, display this message to the user
session_start_url_cb	boolean	Indicates if a URL should be called at when a new session is started
session_start_url_timeout	number	The timeout for the session_start_url HTTP GET request
single_desktop_only	boolean	Indicates whether the user limited to a single desktop connection (Leostream Connect only)
store_configurable_params	number	How the user-configured protocol parameters are stored: *0 = Individually for each connection/client pair (default)* *1 = Globally for all connections and from every client*
usb[]	array	Array container for USB devices (see "usb" table below)
usb_enabled	boolean	Indicates whether USB passthrough is enabled for this policy, defaults to 0
usb_mode	char	The USB passthrough mode: *A = Connect all USB devices (default)* *N = Block all USB devices*

First Level Parameter	Type	Description
		C = Connect specific USB devices
vmware_view_servers[]	array	Array container for Policy VMware View Server (see "vmware_view_servers" table below)
web_client_new_window	boolean	Indicates if Java-based and external web client viewers should be launched in new windows

"attributes" Parameters	Type	Description
attribute_type	string	The type attribute to search: *user* *client* *text*
condition_type	string	The search conditional
text_attribute	string	The free form text attribute
user_attribute	string	The user (or client) attribute to search
vm_attribute	string	The vm table attribute to search

"vmware_view_servers" Parameters	Type	Description
vmware_view_server_name	string	The name of this VMware View Server
vmware_view_server_url	string	The URL of this VMware View Server

"usb" Parameters	Type	Description
devclass	string	The USB device class
subclass	string	The USB device subclass
protocol	string	The USB device protocol

Body Parameters, not used for update/insert

The following Body Parameter is returned by the `policies.get` method. However, to update this parameters, use the `policies.pool_assignments` method. Specifying these parameters in a `policies.update` or `policies.insert` request results in a warning.

First Level Parameter	Type	Description
pool_assignments[]	array	Array container for Policy pool assignments (see Policy Pool assignments methods)

Response

This operation has the following responses.

201 Created

The Create operation was successful.

Successful return

A field 'stored_data', which contains a Policy object in JSON representation, similar to the return of the `policies.get` method.

Optionally a field 'warnings', which contains a set of warnings, each containing a hash (associative array) of the warnings that occurred. Each item is a key-value pair, where the key is the name of a warning or a name of a wrong input field and the value is a text description of the warning.

Optionally a field 'extra_fields', which contains an array of fields not allowed by this method. These fields are removed from the input data before the record is saved.

Errors

400 Bad Request

The operation was not successful: the request is incorrect. The response contains the following fields: 'error_status' and 'error_message'.

Field 'error_status' is BAD_DATA

Typically indicates an invalid body JSON. See the 'error_message' field for additional details. The BAD_DATA error is also raised if the body JSON is valid but sets a value for the 'id' field.

Some errors may also contain the 'errors' field, which contains a hash (associative array) of data that does not pass validation. Usually each item is a pair key-value, where the key is a name of the incorrect input field and the value is a text description of the error.

For input fields that contain more than one value (arrays, as 'attributes' field), the key is a name of the incorrect input field, but the value is a hash containing key-value pairs as <row number> - <error message> (row number starts from 1).

See "Request error example: USB Devices" below.

The response body may contain fields 'warnings' and 'extra_fields' (see "Successful return" above).

403 Forbidden

Field 'error_status' is FORBIDDEN

User's role does not provide permission to take this action. See the 'error_message' field for details.

Request error example: USB Devices

The following example contains a returned error response when the input field 'usb' contains an incorrect third row (both Device ID and Device Class are specified).

```
{
    "error_message": "Error saving of given data, see additional fields for details",
    "error_status": "BAD_DATA",
    "errors": {
        "usb": {
            "3": {
                "devclass": "Specify either Vendor and Device IDs, or a Device Class (not both)"
            }
        }
    }
}
```

Method: policies.update

Update an existing policy. There can be different policy parameters depending on the way the policy is defined. See the 'policies.insert' method documentation for details.

Request

HTTP Method	Relative Request URI
PUT	/policies/`{id}`

Request Path Parameters

Request Parameter	Type	Description
`{id}`	number	Policy ID value.

Request Body Parameters

No required parameters. The optional Body Parameters for this method are the same as the parameters for the 'policies.insert' method, plus the 'id' parameter, which must have the same value as specified for the 'id' path parameter.

Response

This operation has the following responses.

200 OK

The Update operation was successful.

Successful return

A field 'stored_data', which contains a Policy object in JSON representation, similar to return of `policies.get` method.

Optionally a field 'warnings', which contains a set of warnings, each containing a hash (associative array) of warnings that occurred. Each item should is a pair key-value pair, where the key is the name of a warning or of a wrong input field and the value is a text description of the warning.

Optionally a field 'extra_fields', which contains an array of fields not allowed by this method. These fields are removed from the input data before the record is saved.

Errors

400 Bad Request

The operation was not successful: the request is incorrect. The response contains the following fields: 'error_status' and 'error_message'.

Field 'error_status' is BAD_DATA

Typically indicates an invalid body JSON. See the 'error_message' field for additional details.

Some errors may also contain the 'errors' field, which contains a hash (associative array) of data that does not pass validation. Usually each item is a key-value pair, where the key is a name of the incorrect input field and the value is a text description of the error.

For input fields that contain more than one value (arrays, such as the 'attributes' field), the key is a name of the incorrect input field, but the value is a hash containing key-value pairs as <row number> - <error message> (row number starts from 1).

The response body may contain the additional fields 'warnings' and 'extra_fields' (see "Successful return" above).

403 Forbidden

Field 'error_status' is FORBIDDEN

User's role does not provide permission to take this action. See the 'error_message' field for details.

Method: policies.delete

Deletes an existing policy.

Request

HTTP Method	Relative Request URI
DELETE	/policies/{id}

Request Path Parameters

Request Parameter	Type	Description
{id}	number	Policy ID value.

Request Body Parameters

No parameters.

Response

This operation has the following responses.

204 No Content

The Delete operation was successful.

Successful return

No content.

Errors

400 Bad Request

The operation was not successful: the request is incorrect. The response contains the following fields: 'error_status' and 'error_message'.

Field 'error_status' is BAD_DATA

Invalid Policy ID specified.

403 Forbidden

Field 'error_status' is FORBIDDEN

User's role does not provide permission to take this action. See the 'error_message' field for details.

Method: policies.pool_assignments.list

Retrieves the list of Pool Assignments for the specified Policy.

Request

HTTP Method	Relative Request URI
GET	/policies/{policy_id}/pool-assignments

Request Path Parameters

Request Parameter	Type	Description
{policy_id}	number	Policy ID value.

Request Body Parameters

No parameters.

Responses

This operation has the following responses.

200 OK

The operation was successful.

Successful return

A list of Pool Assignments. Each record contains general information about a Pool Assignment.

Response body

If successful, the response body contains data with a structure similar to the following:

```
[
    {
        "id": 14,
        "pool_id": 8,
        "pool_name": "vCenter 6 Pool"
    },
    {
        "id": 18,
        "pool_id": 9,
        "pool_name": "vCenter 6 hosts Pool"
    }
]
```

Method: policies.pool_assignments.get

Returns the specified Pool Assignment resource from the specified Policy.

Request

HTTP Method	Relative Request URI
GET	/policies/`{policy_id}`/pool-assignments/`{id}`

Request Path Parameters

Request Parameter	Type	Description
`{policy_id}`	number	Policy ID value.
`{id}`	number	Pool Assignment ID value.

Request Body Parameters

No parameters.

Responses

This operation has the following responses.

200 OK

The operation was successful.

Successful return

A Pool Assignment object in JSON representation. The returned JSON contains a set of reference fields with general information about referenced objects, for example, for the referenced ID of the protocol plan for the backup pool ('bu_plan_protocol_id'), the data will be an object named 'bu_plan_protocol_data'. This data is a read-only value and cannot be changed by the 'policies.pool_assignments.update' method.

Response body

If successful, the response body contains data with the similar structure:


```
{
    "adjust_timezone": 0,
    "attribute_filter": [
                          {
                            "vm_attribute": "name",
                            "text_attribute": "ss",
                            "attribute_type": "text",
                            "user_attribute": "text",
                            "condition_type": "ct"
                          }
                        ],
    "attribute_join": "A",
    "auto_login": 0,
    "backup_pool_criteria": "0",
    "backup_pool_id": 0,
    "bu_plan_power_control_data": {
        "id": 1,
        "name": "Default"
    },
    "bu_plan_power_control_id": 1,
    "bu_plan_protocol_data": {
        "id": 1,
        "name": "Default"
    },
    "bu_plan_protocol_id": 1,
    "bu_plan_release_data": {
        "id": 1,
        "name": "Default"
    },
    "bu_plan_release_id": 1,
    "confirm_power_state": 0,
    "display_mode": "0",
    "email_decline_shadowing": 0,
    "email_shadowing": 0,
    "enable_power_control": 0,
    "enable_shadowing": 0,
    "favor_previous_assigned": 1,
    "id": 14,
    "kiosk": 0,
    "login_as": "R",
    "logout_rogue": 0,
    "offer_filter": "0",
    "offer_filter_json": null,
    "offer_pending_reboot": 1,
    "offer_quantity": 1,
    "offer_running_without_hda": 0,
    "on_assign_url": "",
```

```
    "on_assign_url_cb": 0,
    "on_assign_url_timeout": 5,
    "plan_power_control_data": {
        "id": 1,
        "name": "Default"
    },
    "plan_power_control_id": 1,
    "plan_protocol_data": {
        "id": 1,
        "name": "Default"
    },
    "plan_protocol_id": 1,
    "plan_release_data": {
        "id": 1,
        "name": "Default"
    },
    "plan_release_id": 1,
    "policy_id": 3,
    "pool_data": {
        "id": 8,
        "name": "vCenter 6 Pool"
    },
    "pool_id": 8,
    "power_on": 0,
    "prevent_release": 0,
    "revert_to_snapshot": 0,
    "shadowing_filter": "0",
    "shadowing_filter_json": null,
    "start_if_stopped": 1
}
```

Method: policies.pool_assignments.insert

Adds a Pool Assignment to an existing Policy. Specify different Assignment parameters depending on the way the Pool Assignment is defined.

Request

HTTP Method	Relative Request URI
POST	/policies/`{policy_id}`/pool-assignments

Request Path Parameters

Request Parameter	Type	Description
`{policy_id}`	number	Policy ID value.

Request Body Parameters

Request Parameter	Type	Description
pool_id	string	ID of the associated pool

Optional Body Parameters

First Level Parameter	Type	Description
policy_id	number	ID of the associated policy
adjust_timezone	boolean	Indicates that the timezone of the destination desktop will be changed to the user's current timezone at login
attribute_filter	string	The pool filter as a JSON data structure (either JSON or stringified JSON is accepted)
attribute_join	char	How do the pool filter attributes get joined: A = And O = Or
auto_login	boolean	Indicates that a console single sign-on (DCV, VNC, PCoIP, and HTML5) should be done
backup_pool_criteria_agent	boolean	The method used to determine when to offer a desktop from the backup pool: - *Leostream Agent is unreachable (also the default)*
backup_pool_criteria_empty	boolean	- *Remote viewer port on desktop is unreachable*
backup_pool_criteria_viewer	boolean	- *Primary pool has no available desktops*
backup_pool_id	number	ID of the associated backup pool
bu_plan_power_control_id	number	ID of the power control plan for the backup pool
bu_plan_protocol_id	number	ID of the protocol plan for the backup pool
bu_plan_release_id	number	ID of the release plan for the backup pool
confirm_power_state	boolean	Indicates whether to probe a machine for its power state when assigned
display_mode	char	How to describe the offered machines: 0 = Desktop name (default) 5 = Desktop display name 1 = Machine name 2 = Pool name 3 = Pool name : Desktop name 6 = Pool name : Desktop display name 4 = Pool name : Machine name 7 = Pool display name 8 = Pool display name : Desktop name 9 = Pool display name : Desktop display name 10= Pool display name : Machine name
email_decline_shadowing	boolean	Indicates if the user that sent the invitation should be notified via email (no user choice for this option), if the invited user declines the invitation
email_shadowing	boolean	Indicates if the session owner can decide if an email should be sent to the user being invited
enable_power_control	number	How to perform power control actions if requested by a user from a thin client: 0 = Not allowed 1 = Power Control with Shutdown 2 = Power Control with Power off
enable_shadowing	boolean	Indicates if session shadowing is enabled
favor_previous_assigned	number	How to order offered machines: 0 = Randomly (any available desktop) 1 = Favor desktops previously-assigned to this user (default) 2 = Sort by desktop creation date (first-in, first-out)
kiosk	boolean	Indicates whether to offer a different machine with each login
login_as	char	Determines how domain is sent in protocol plans and single sign-on commands: R = Defer to Role setting: role.login_as_local_user D = Always send user's domain (log in as a domain user) L = Always send desktop machine name (log in as a local user)
logout_rogue	boolean	Indicates that rogue users should be logged out
offer_filter	char	The method used to decide whether desktops from this pool will be included in the offer:

First Level Parameter	Type	Description
		0: Included for all users 1: Only included if user's AD record matches the offer_filter_json criteria
offer_filter_json	string	The user offer filter as a JSON data structure (either JSON or stringified JSON is accepted)
offer_pending_reboot	boolean	Indicates if machines with a pending reboot should be offered
offer_quantity	number	The number of VMs to offer to a user at login
offer_running_without_hda	boolean	Indicates that machines without a running Leostream Agent may be offered
on_assign_url	string	A URL that is called when a desktop is assigned, if on_assign_url_cb is true. May include dynamic tags.
on_assign_url_cb	boolean	Indicates if a URL should be called at assignment time
on_assign_url_timeout	number	The timeout for the on_assign_url HTTP GET request
plan_power_control_id	number	ID of the power control plan for this pool
plan_protocol_id	number	ID of the protocol plan for this pool
plan_release_id	number	ID of the release plan for this pool
power_on	number	Determines which stopped and suspended machines are offered: 0 = Do not offer any stopped/suspended machines 1 = Offer any stopped/suspended machine 2 = Only only stopped/suspended machine if Leostream Agent is installed
prevent_release	boolean	Indicates whether user can manually release a vm (0=yes)
revert_to_snapshot	boolean	Indicates that the machine should be reverted to the last snapshot when assigned
shadowing_filter	char	The method used to decide which users will be allowed to use session shadowing: 0: Included for all users 1: Only included if user's AD record matches the shadowing_filter_json criteria
shadowing_filter_json	string	The user shadowing filter as a JSON data structure (either JSON or stringified JSON is accepted)
start_if_stopped	boolean	Indicates whether to attempt to power on a machine if it's currently stopped/suspended

Response

This operation has the following responses.

201 Created

The Create operation was successful.

Successful return

A field 'stored_data', which contains a Pool Assignment object in JSON representation, similar to the return of the 'policies.pool_assignments.get' method.

Optionally a field 'warnings', which contains a set of warnings, each with two fields:

- -name - the name of invalid input field;
- -message - a description of the warning.

Optionally a field 'extra_fields', which contains an array of fields not allowed by this method. These fields were removed from the input data before the record was saved.

Errors

400 Bad Request

The operation was not successful: the request is incorrect. The response contains the following fields: 'error_status' and 'error_message'.

Field 'error_status' is BAD_DATA

Typically indicates an incorrect body JSON. See the 'error_message' field for details. The BAD_DATA error is also raised if the body JSON is valid, but 'id' field value is set.

Some errors may also return an 'errors' field, which contains an array of data that does not pass validation. Usually each item contains two fields:

- -name - the name of an incorrect input field;
- -message - a description of the error.

The response body may contain the additional fields 'warnings' and 'extra_fields' (see "Successful return" above).

403 Forbidden

Field 'error_status' is FORBIDDEN

User's role does not provide permission to take this action. See the 'error_message' field for details.

404 Not Found

Indicates that the requested Policy ID does not exist.

Method: policies.pool_assignments.update

Update an existing Pool Assignment. Specify different Pool Assignment parameters depending on how the Pool Assignment is defined. See the 'policies.pool_assignments.insert' method documentation for details.

Request

HTTP Method	Relative Request URI
PUT	/policies/`{policy_id}`/pool-assignments/`{id}`

Request Path Parameters

Request Parameter	Type	Description
`{policy_id}`	number	Policy ID value.
`{id}`	number	Pool Assignment ID value.

Request Body Parameters

No required parameters. The optional Body Parameters for the method are the same as for the 'policies.pool_assignments.insert' method, plus the 'id' parameter, which must have the same value as entered for the 'id' path parameter.

Response

This operation has the following responses.

200 OK

The Update operation was successful.

Successful return

A field 'stored_data', which contains a Pool Assignment object in JSON representation, similar to the return of 'policies.pool_assignments.get' method.

Optionally a field 'warnings', which contains a set of warnings, each with two fields:

- -name - the name of wrong input field;
- -message - a description of the warning.

Optionally a field 'extra_fields', which contains an array of fields not allowed by this method. These fields are removed from the input data before the record is saved.

Errors

400 Bad Request

The operation was not successful: the request is incorrect. The response contains the following fields: 'error_status' and 'error_message'.

Field 'error_status' is BAD_DATA

Typically indicates an invalid body JSON. See the 'error_message' field for additional details.

Certain errors may return an additional 'errors' field which contains an array of data that does not pass validation. Typically, each item contains two fields:

- -name - contains a name of wrong input field;
- -message - description of the error.

Also the response body could contain fields 'warnings' and 'extra_fields' (see "Successful return" above).

403 Forbidden

Field 'error_status' is FORBIDDEN

User's role does not provide permission to take this action. See the 'error_message' field for details.

404 Not Found

This error is raised if either the Policy ID or the Assignment ID does not exist.

Method: policies.pool_assignments.delete

Deletes an existing Pool Assignment resource.

Request

HTTP Method	Relative Request URI
DELETE	/policies/`{policy_id}`/pool-assignments/`{id}`

Request Path Parameters

Request Parameter	Type	Description
`{policy_id}`	number	Policy ID value.
`{id}`	number	Pool Assignment ID value.

Request Body Parameters

No parameters.

Response

This operation has the following responses.

204 No Content

The Delete operation was successful.

Successful return

No content.

Errors

400 Bad Request

The operation was not successful: the request is incorrect. The response contains the following fields: 'error_status' and 'error_message'.

Field 'error_status' is BAD_DATA

An invalid ID was specified.

403 Forbidden

Field 'error_status' is FORBIDDEN

User's role does not provide permission to take this action. See the 'error_message' field for details.

404 Not Found

This error is raised if either the Policy ID or the Assignment ID does not exist.

Method: system.license.get

Returns the current Leostream license information and usage.

Request

HTTP Method	Relative Request URI
GET	/system/license

Request Path Parameters

No parameters.

Request Body Parameters

No parameters.

Response

This operation has the following responses.

200 OK

The operation was successful.

Successful return

A License object in JSON representation.

Response body

If successful, the response body contains data with a structure similar to the following:

```
{
    "active_desktops": 0,
    "active_users": 1234,
    "active_users_text": "1,234",
    "cluster_max": 2,
    "expiration_date": "2021-12-31",
    "install_no": "ffffffffffffffffffffffffffffff-11111",
    "license_expired": 0,
    "license_present": 1,
    "license_valid": 1,
    "licensed_desktops": 1730730,
    "licensed_desktops_text": "unlimited",
    "licensed_gateways": 10,
    "licensed_tenants": 0,
    "licensed_users": 1300,
    "licensed_users_text": "1,300",
    "num_brokers": 1,
    "num_gateways": 0,
    "num_tenants": 0,
    "serial_no": "000-000-000",
    "support_expiration_date": "2021-12-31",
    "support_expired": 0
}
```


A valid license must include 'unlimited' for either 'licensed_desktops_text' or 'licensed_users_text', but not for both. A license based on Named Users returns 'unlimited' for 'licensed_desktops_text' while a license based on Managed Desktops returns 'unlimited' for 'licensed_users_text'. If the license allows an unlimited number of destkops or users, the license returns '1730730' for either 'licensed_desktops' or 'licensed_users', respectively