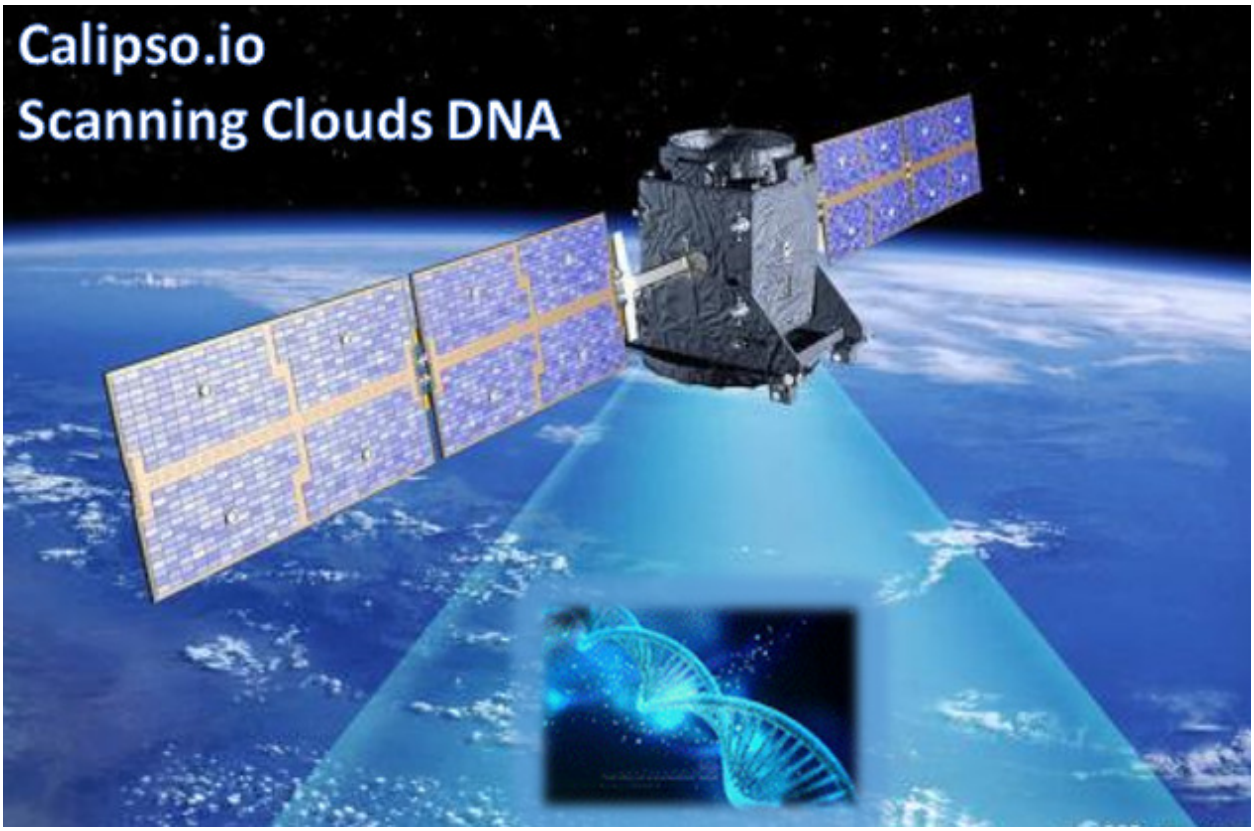
Calipso.io
API Guide



Project “Calipso” tries to illuminate complex virtual networking with real time operational state visibility for large and highly distributed Virtual Infrastructure Management (VIM).

We believe that Stability is driven by accurate Visibility.

Calipso provides visible insights using smart discovery and virtual topological representation in graphs, with monitoring per object in the graph inventory to reduce error vectors and troubleshooting, maintenance cycles for VIM operators and administrators.

Table of Contents

1 Pre Requisites

1.1 Calipso API container

Calipso's main application is written with Python3.5 for Linux Servers, tested successfully on Centos 7.3 and Ubuntu 16.04. When running using micro-services many of the required software packages and libraries are delivered per micro service, including the API module case. In a monolithic case dependencies are needed.
Here is a list of the required software packages for the API, and the official supported steps required to install them:

1. Python3.5.x for Linux : https://docs.python.org/3.5/using/unix.html#on-linux
2. Pip for Python3 : https://docs.python.org/3/installing/index.html
3. Python3 packages to install using pip3 :
4. falcon (1.1.0)
5. pymongo (3.4.0)
6. gunicorn (19.6.0)
7. ldap3 (2.1.1)
8. setuptools (34.3.2)
9. python3-dateutil (2.5.3-2)
10. bcrypt (3.1.1)

You should use pip3 python package manager to install the specific version of the library. Calipso project uses Python 3, so package installation should look like this: pip3 install falcon==1.1.0
The versions of the Python packages specified above are the ones that were used in the development of the API, other versions might also be compatible.
This document describes how to setup Calipso API container for development against the API.

2 Overview

2.1 Introduction

The Calipso API provides access to the Calipso data stored in the MongoDB.
Calispo API uses falcon (https://falconframework.org) web framework and gunicorn (http://gunicorn.org) WSGI server.
The authentication of the Calipso API is based on LDAP (Lightweight Directory Access Protocol). It can therefore interface with any directory servers which implements the LDAP protocol, e.g. OpenLDAP, Active Directory etc. Calipso app offers and uses the LDAP built-in container by default to make sure this integration is fully tested, but it is possible to interface to other existing directories.

2.2 HTTP Standards

The Calipso API supports standard HTTP methods described here:
https://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html.
At present two types of operations are supported: GET (retrieve data) and POST (create a new data object).

2.3 Calipso API module Code

Clipso API code is currently located in opnfv repository.
Run the following command to get the source code:
git clone **https://git.opnfv.org/calipso/**
The source code of the API is located in the app/api directory sub-tree.

3 Starting the Calipso API server

3.1 Authentication

Calipso API uses LDAP as the protocol to implement the authentication, so you can use any LDAP directory server as the authentication backend, like OpenLDAP and Microsoft AD. You can edit the ldap.conf file which is located in app/config directory to configure LDAP server options (see details in quickstart-guide):

```
# url for connecting to the LDAP server (customize to your own as needed):
url ldap_url
# LDAP attribute mapped to user id, must not be a multivalued attributes:
user_id_attribute CN
# LDAP attribute mapped to user password:
user_pass_attribute userPassword
# LDAP objectclass for user
user_objectclass inetOrgPerson
# Search base for users
user_tree_dn OU=Employees,OU=Example Users,DC=exmaple,DC=com
query_scope one
# Valid options for tls_req_cert are demand, never, and allow
tls_req_cert demand
# CA certificate file path for communicating with LDAP servers.
tls_cacertfile ca_cert_file_path
group_member_attribute member
```

Calipso currently implements the basic authentication, the client send the query request with its username and password in the auth header, if the user can be bound to the LDAP server, authentication succeeds otherwise fails. Other methods will be supported in future releases.

3.2 Database

Calipso API query for and retrieves data from MongoDB container, the data in the MongoDB comes from the results of Calipso scanning, monitoring or the user inputs from the API. All modules of a single Calipso instance of the application must point to the same MongoDB used by the scanning and monitoring modules. Installation and testing of mongoDB is covered in install-guide and quickstart-guide.

3.3 Running the API Server

The entry point (initial command) running the Calipso API application is the server.py script in the app/api directory. Options for running the API server can be listed using: python3 server.py –help. Here is the current options available:

-m [MONGO_CONFIG], --mongo_config [MONGO_CONFIG]
name of config file with mongo access details
--ldap_config [LDAP_CONFIG]
name of the config file with ldap server config details
-l [LOGLEVEL], --loglevel [LOGLEVEL] logging level (default: 'INFO')
-b [BIND], --bind [BIND]
binding address of the API server (default: 127.0.0.1:8000)
-y [INVENTORY], --inventory [INVENTORY]
name of inventory collection (default: 'inventory')

For testing, you can simply run the API server by:

python3 app/api/server.py

This will start a HTTP server listening on http://localhost:8000, if you want to change the binding address of the server, you can run it using this command:

python3 server.py --bind ip_address/server_name:port_number

You can also use your own configuration files for LDAP server and MongoDB, just add --mongo_config and --ldap_config options in your command:

python3 server.py --mongo_config your_mongo_config_file_path --ldap_config your_ldap_config_file_path

--inventory option is used to set the collection names the server uses for the API, as per the quickstart-guide this will default to **/local_dir/calipso_mongo_access.conf** and **/local_dir/ldap.conf** mounted inside the API container.

Notes: the --inventory argument can only change the collection names of the inventory, links, link_types, clique_types, clique_constraints, cliques, constants and

scans collections, names of the monitoring_config_templates, environments_config and messages collections will remain at the root level across releases.

4 Using the Calipso API server

The following covers the currently available requests and responses on the Calipso API

4.1 Authentication

POST /auth/tokens

Description: get token with password and username or a valid token.

Normal response code: 201

Error response code: badRequest(400), unauthorized(401)

Request

Name	In	Type	Description
auth(Mandatory)	body	object	An auth object that contains the authentication information
methods(Mandatory)	body	array	The authentication methods. For password authentication, specify password, for token authentication, specify token.
credentials(Optional)	body	object	Credentials object which contains the username and password, it must be provided when getting the token with user credentials.
token(Optional)	body	string	The token of the user, it must be provided when getting the user with an existing valid token.

Response

Name	In	Type	Description
token	body	string	Token for the user.
issued-at	body	string	The date and time when the token was issued. the date and time format follows ISO 8610: YYYY-MM-DDThh:mm:ss.sss+hhmm
expires_at	body	string	The date and time when the token expires. the date and time format follows ISO 8610: YYYY-MM-DDThh:mm:ss.sss+hhmm
method	body	string	The method which achieves the token.

Examples

Get token with credentials:

Post http://korlev-osdna-staging1.cisco.com:8000/auth/tokens

```
{
    "auth": {
        "methods": ["credentials"],
        "credentials": {
            "username": "username",
            "password": "password"
        }
    }
}
```

Get token with token

post http://korlev-calipso-staging1.cisco.com:8000/auth/tokens

```
{
   "auth": {
        "methods": ["token"],
        "token": "17dfa88789aa47f6bb8501865d905f13"
   }
}
```

DELETE /auth/tokens

Description: delete token with a valid token.

Normal response code: 200

Error response code: badRequest(400), unauthorized(401)

Request

Name	In	Type	Description
X-Auth-Token	header	string	A valid authentication token that is doing to be deleted.

Response

200 OK will be returned when the delete succeed

4.2 Messages

GET /messages

Description: get message details with environment name and message id, or get a list of messages with filters except id.

Normal response code: 200

Error response code: badRequest(400), unauthorized(401), notFound(404)

Request

Name	In	Type	Description
env_name(Mandatory)	query	string	Environment name of the messages. e.g. "Mirantis-Liberty-API".
id (Optional)	query	string	ID of the message.
source_system (Optional)	query	string	Source system of the message, e.g. "OpenStack".
start_time (Optional)	query	string	Start time of the messages, when this parameter is specified, the messages after that time will be returned, the date and time format follows ISO 8610: YYYY-MM-DDThh:mm:ss.sss±hhmm The ±hhmm value, if included, returns the time zone as an offset from UTC, For example, 2017-01-25T09:45:33.000-0500. If you omit the time zone, the UTC time is assumed.
end_time (Optional)	query	string	End time of the message, when this parameter is specified, the messages before that time will be returned, the date and time format follows ISO 8610: YYYY-MM-DDThh:mm:ss.sss±hhmm The ±hhmm value, if included, returns the time zone as an offset from UTC, For example, 2017-01-25T09:45:33.000-0500. If you omit the time zone, the UTC time is assumed.
level (Optional)	query	string	The severity of the messages, we accept the severities strings described in RFC 5424, possible values are "panic", "alert", "crit", "error", "warn", "notice", "info" and "debug".

related_object (Optional)	query	string	ID of the object related to the message.
related_object_type (Optional)	query	string	Type of the object related to the message, possible values are "vnic", "vconnector", "vedge", "instance", "vservice", "host_pnic", "network", "port", "otep" and "agent".
page (Optional)	query	int	Which page will to be returned, the default is first page, if the page is larger than the maximum page of the query, and it will return an empty result set (Page start from 0).
page_size (Optional)	query	int	Size of each page, the default is 1000.

Response

Name	In	Type	Description
environment	body	string	Environment name of the message.
id	body	string	ID of the message.
_id	body	string	MongoDB ObjectId of the message.
timestamp	body	string	Timestamp of message.
viewed	body	boolean	Indicates whether the message has been viewed.
display_context	body	string	The content which will be displayed.
message	body	object	Message object.
source_system	body	string	Source system of the message, e.g. "OpenStack".
level	body	string	The severity of the message.
related_object	body	string	Related object of the message.
related_object_type	body	string	Type of the related object.
messages	body	array	List of message ids which match the filters.

Examples

Example Get Messages

Request:

http://korlev-calipso-testing.cisco.com:8000/messages?env_name=Mirantis-Liberty-API&start_time=2017-01-25T14:28:32.400Z&end_time=2017-01-25T14:28:42.400Z

Response:


```
{
    messages: [

            {
                "level": "info",
                "environment": "Mirantis-Liberty",
                "id": "3c64fe31-ca3b-49a3-b5d3-c485d7a452e7",
                "source_system": "OpenStack"
            },
            {
                "level": "info",
                "environment": "Mirantis-Liberty",
                "id": "c7071ec0-04db-4820-92ff-3ed2b916738f",
                "source_system": "OpenStack"
            },

    ]
}
```


Example Get Message Details

Request

http://korlev-calipso-testing.cisco.com:8000/messages?env_name=Mirantis-Liberty-API&id=80b5e074-0f1a-4b67-810c-fa9c92d41a98

Response


```
{
"related_object_type": "instance",
"source_system": "OpenStack",
"level": "info",
"timestamp": "2017-01-25T14:28:33.057000",
"_id": "588926916a283a8bee15cfc6",
"viewed": true,
"display_context": "*",
"related_object": "97a1e179-6a42-4c7b-bced-4f64bd9e4b6b",
"environment": "Mirantis-Liberty-API",
"message": {
"_context_show_deleted": false,
"_context_user_name": "admin",
"_context_project_id": "a3efb05cd0484bf0b600e45dab09276d",
"_context_service_catalog": [
{
"type": "volume",
```

"endpoints": [
{
"internalURL": "http://192.168.0.2:8776/v1/a3efb05cd0484bf0b600e45dab09276d",
"publicURL": "http://172.16.0.3:8776/v1/a3efb05cd0484bf0b600e45dab09276d",
"adminURL": "http://192.168.0.2:8776/v1/a3efb05cd0484bf0b600e45dab09276d",
"region": "RegionOne"
}
],
"name": "cinder"
},
{
"type": "volumev2",
"endpoints": [
{
"internalURL": "http://192.168.0.2:8776/v2/a3efb05cd0484bf0b600e45dab09276d",
"publicURL": "http://172.16.0.3:8776/v2/a3efb05cd0484bf0b600e45dab09276d",
"adminURL": "http://192.168.0.2:8776/v2/a3efb05cd0484bf0b600e45dab09276d",
"region": "RegionOne"
}
],
"name": "cinderv2"
}
],
"_context_user_identity": "a864d9560b3048e9864118555bb9614c
a3efb05cd0484bf0b600e45dab09276d - - -",
"_context_project_domain": null,
"_context_is_admin": true,
"_context_instance_lock_checked": false,
"_context_timestamp": "2017-01-25T22:27:08.773313",
"priority": "INFO",
"_context_project_name": "project-osdna",
"publisher_id": "compute.node-1.cisco.com",
"_context_read_only": false,
"message_id": "80b5e074-0f1a-4b67-810c-fa9c92d41a98",
"_context_user_id": "a864d9560b3048e9864118555bb9614c",
"_context_quota_class": null,
"_context_tenant": "a3efb05cd0484bf0b600e45dab09276d",
"_context_remote_address": "192.168.0.2",
"_context_request_id": "req-2955726b-f227-4eac-9826-b675f5345ceb",
"_context_auth_token": "gAAAAABYiSVcHmaq1TWwNc1_QLlKhdUeC1-
M6zBebXyoXN4D0vMlxisny9Q61crBzqwSyY_Eqd_yjrL8GvxatWI1WI1uG4VeWU6axbLe_k
5FaXS4RVOP83yR6eh5g_qXQtsNapQufZB1paypZm8YGERRvR-
vV5Ee76aTSkytVjwOBeipr9D0dXd-wHcRnSNkTD76nFbGKTu_",
"_context_user_domain": null,
"payload": {
"image_meta": {

"container_format": "bare",
"disk_format": "qcow2",
"min_ram": "64",
"base_image_ref": "5f048984-37d1-4952-8b8a-9acb0237bad7",
"min_disk": "0"
},
"display_name": "test",
"terminated_at": "",
"access_ip_v6": null,
"architecture": null,
"image_ref_url": "http://192.168.0.3:9292/images/5f048984-37d1-4952-8b8a-9acb0237bad7",
"audit_period_beginning": "2017-01-01T00:00:00.000000",
"metadata": {},
"node": "node-2.cisco.com",
"audit_period_ending": "2017-01-25T22:27:12.888042",
"instance_type": "m1.micro",
"ramdisk_id": "",
"availability_zone": "nova",
"kernel_id": "",
"hostname": "test",
"vcpus": 1,
"bandwidth": {},
"user_id": "a864d9560b3048e9864118555bb9614c",
"state_description": "block_device_mapping",
"old_state": "building",
"root_gb": 0,
"instance_flavor_id": "8784e0b5-7d17-4281-a509-f49d6fd102f9",
"cell_name": "",
"reservation_id": "r-zt7sh7vy",
"access_ip_v4": null,
"deleted_at": "",
"tenant_id": "a3efb05cd0484bf0b600e45dab09276d",
"disk_gb": 0,
"instance_id": "97a1e179-6a42-4c7b-bced-4f64bd9e4b6b",
"host": "node-2.cisco.com",
"memory_mb": 64,
"os_type": null,
"old_task_state": "block_device_mapping",
"state": "building",
"instance_type_id": 6,
"launched_at": "",
"ephemeral_gb": 0,
"created_at": "2017-01-25 22:27:09+00:00",
"progress": "",
"new_task_state": "block_device_mapping"
},

"_context_read_deleted": "no",
"event_type": "compute.instance.update",
"_context_roles": [
"admin",
"_member_"
],
"_context_user": "a864d9560b3048e9864118555bb9614c",
"timestamp": "2017-01-25 22:27:12.912744",
"_unique_id": "d6dff97e6f71401bb8890057f872644f",
"_context_resource_uuid": null,
"_context_domain": null
},
"id": "80b5e074-0f1a-4b67-810c-fa9c92d41a98"
}

4.3 Inventory

GET /inventory

Description: get object details with environment name and id of the object, or get a list of objects with filters except id.

Normal response code: 200

Error response code: badRequest(400), unauthorized(401), notFound(404)

Request

Name	In	Type	Description
env_name (Mandatory)	query	string	Environment of the objects. e.g. "Mirantis-Liberty-API".
id (Optional)	query	string	ID of the object. e.g. "node-2.cisco.com".
parent_id (Optional)	query	string	ID of the parent object. e.g. "nova".
id_path (Optional)	query	string	ID path of the object. e.g. "/Mirantis-Liberty-API/Mirantis-Liberty-API-regions/RegionOne/RegionOne-availability_zones/nova/node-2.cisco.com".
parent_path (Optional)	query	string	ID path of the parent object. "/Mirantis-Liberty-API/Mirantis-Liberty-API-regions/RegionOne/RegionOne-availability_zones/nova".
sub_tree (Optional)	query	boolean	If it is true and the parent_path is specified, it will return the whole sub-tree of that parent object which includes the parent itself, If it is false and the parent_path is specified, it will only return the siblings of that parent (just the children of that parent node), the default value of sub_tree is false.

page (Optional)	query	int	Which page is to be returned, the default is the first page, if the page is larger than the maximum page of the query, it will return an empty set, (page starts from 0).
page_size (Optional)	query	int	Size of each page, the default is 1000.

Response

Name	In	Type	Description
environment	body	string	Environment name of the object.
id	body	string	ID of the object.
_id	body	string	MongoDB ObjectId of the object.
type	body	string	Type of the object.
parent_type	body	string	Type of the parent object.
parent_id	body	string	ID of the parent object.
name_path	body	string	Name path of the object.
last_scanned	body	string	Time of last scanning.
name	body	string	Name of the object.
id_path	body	string	ID path of the object.
objects	body	array	The list of object IDs that match the filters.

Examples

Example Get Objects

Request

http://korlev-calipso-testing.cisco.com:8000/inventory?env_name=Mirantis-Liberty-API&parent_path=/Mirantis-Liberty-API/Mirantis-Liberty-API-regions/RegionOne&sub_tree=false

Response


```
{

   "objects": [

                {
                    "id": "Mirantis-Liberty-regions",
                    "name": "Regions",
                    "name_path": "/Mirantis-Liberty/Regions"
                },
                {
                    "id": "Mirantis-Liberty-projects",
```

```
        "name": "Projects",
        "name_path": "/Mirantis-Liberty/Projects"
    }

    ]

}
```


Examples Get Object Details

Request

http://korlev-calipso-testing.cisco.com:8000/inventory?env_name=Mirantis-Liberty-API&id=node-2.cisco.com

Response


```
{
  'ip_address': '192.168.0.5',
  'services': {
    'nova-compute': {
      'active': True,
      'updated_at': '2017-01-20T23:03:57.000000',
      'available': True
    }
  },
'name': 'node-2.cisco.com',
'id_path': '/Mirantis-Liberty-API/Mirantis-Liberty-API-regions/RegionOne/RegionOne-availability_zones/nova/node-2.cisco.com',
'show_in_tree': True,
'os_id': '1',
'object_name': 'node-2.cisco.com',
'_id': '588297ae6a283a8bee15cc0d',
'host_type': [
  'Compute'
],
'name_path': '/Mirantis-Liberty-API/Regions/RegionOne/Availability Zones/nova/node-2.cisco.com',
'parent_type': 'availability_zone',
'zone': 'nova',
'parent_id': 'nova',
'host': 'node-2.cisco.com',
'last_scanned': '2017-01-20T15:05:18.501000',
'id': 'node-2.cisco.com',
'environment': 'Mirantis-Liberty-API',
'type': 'host'
}
```

4.4 Links

GET /links

Description: get link details with environment name and id of the link, or get a list of links with filters except id

Normal response code: 200

Error response code: badRequest(400), unauthorized(401), notFound(404)

Request

Name	In	Type	Description
env_name (Mandatory)	query	string	Environment of the links. e.g. "Mirantis-Liberty-API".
id (Optional)	query	string	ID of the link, it must be a string which can be converted to MongoDB ObjectId.
host (Optional)	query	string	Host of the link. e.g. "node-1.cisco.com".
link_type (Optional)	query	string	Type of the link, some possible values for that are "instance-vnic", "otep-vconnector", "otep-host_pnic", "host_pnic-network", "vedge-otep", "vnic-vconnector", "vconnector-host_pnic", "vnic-vedge", "vedge-host_pnic" and "vservice-vnic" .
link_name (Optional)	query	string	Name of the link. e.g. "Segment-2".
source_id (Optional)	query	string	ID of the source object of the link. e.g. "qdhcp-4f4bf8b5-ca42-411a-9f64-5b214d1f1c71".
target_id (Optional)	query	string	ID of the target object of the link. "tap708d399a-20".
state (Optional)	query	string	State of the link, "up" or "down".
attributes	query	object	The attributes of the link, e.g. the network attribute of the link is attributes:network="4f4bf8b5-ca42-411a-9f64-5b214d1f1c71".
page (Optional)	query	int	Which page is to be returned, the default is first page, when the page is larger than the maximum page of the query, it will return an empty set. (Page starts from 0).
page_size (Optional)	query	int	Size of each page, the default is 1000.

Response

Name	In	Type	Description
id	body	string	ID of the link.
_id	body	string	MongoDB ObjectId of the link.
environment	body	string	Environment of the link.
source_id	body	string	ID of the source object of the link.
target_id	body	string	ID of the target object of the link.
source	body	string	MongoDB ObjectId of the source object.
target	body	string	MongoDB ObjectId of the target object.
source_label	body	string	Descriptive text for the source object.
target_label	body	string	Descriptive text for the target object.
link_weight	body	string	Weight of the link.
link_type	body	string	Type of the link, some possible values for that are "instance-vnic", "otep-vconnector", "otep-host_pnic", "host_pnic-network", "vedge-otep", "vnic-vconnector", "vconnector-host_pnic", "vnic-vedge", "vedge-host_pnic" and "vservice-vnic".
state	body	string	State of the link, "up" or "down".
attributes	body	object	The attributes of the link.
links	body	array	List of link IDs which match the filters.

Examples

Example Get Link Ids

Request

http://korlev-calipso-testing.cisco.com:8000/links?env_name=Mirantis-Liberty-API&host=node-2.cisco.com

Response


```
{

   "links": [

            {
                "id": "58ca73ae3a8a836d10ff3b45",
                "host": "node-1.cisco.com",
                "link_type": "host_pnic-network",
                "link_name": "Segment-103",
```

```
            "environment": "Mirantis-Liberty"
        }

    ]

}
```


Example Get Link Details

Request

http://korlev-calipso-testing.cisco.com:8000/links?env_name=Mirantis-Liberty-API&id=5882982c6a283a8bee15cc62

Response


```
{
    "target_id": "6d0250ae-e7df-4b30-aa89-d9fcc22e6371",
    "target": "58a23ff16a283a8bee15d3e6",
    "link_type": "vnic-vedge",
    "link_name": "qr-24364cd7-ab-node-1.cisco.com-OVS-3",
    "environment": "Mirantis-Liberty-API",
    "_id": "58a240646a283a8bee15d438",
    "source_label": "fa:16:3e:38:11:c9",
    "state": "up",
    "link_weight": 0,
    "id": "58a240646a283a8bee15d438",
    "host": "node-1.cisco.com",
    "source": "58a23fd46a283a8bee15d3c6",
    "target_label": "10",
    "attributes": {},
    "source_id": "qr-24364cd7-ab"
}
```

4.5 Cliques

GET /cliques

Description: get clique details with environment name and clique id, or get a list of cliques with filters except id

Normal response code: 200

Error response code: badRequest(400), unauthorized(401), notFound(404)

Request

Name	In	Type	Description
env_name (Mandatory)	query	string	Environment of the cliques. e.g. "Mirantis-Liberty-API".
id (Optional)	query	string	ID of the clique, it must be a string that can be converted to Mongo ObjectID.
focal_point (Optional)	query	string	MongoDB ObjectId of the focal point object, it must be a string that can be converted to Mongo ObjectID.
focal_point_type (Optional)	query	string	Type of the focal point object, some possible values are "vnic", "vconnector", "vedge", "instance", "vservice", "host_pnic", "network", "port", "otep" and "agent".
link_type(Optional)	query	string	Type of the link, when this filter is specified, it will return all the cliques which contain the specific type of the link, some possible values for link_type are "instance-vnic", "otep-vconnector", "otep-host_pnic", "host_pnic-network", "vedge-otep", "vnic-vconnector", "vconnector-host_pnic", "vnic-vedge", "vedge-host_pnic" and "vservice-vnic".
link_id (Optional)	query	string	MongoDB ObjectId of the link, it must be a string that can be converted to MongoDB ID, when this filter is specified, it will return all the cliques which contain that specific link.
page (Optional)	query	int	The page is to be returned, the default is the first page, if the page is larger than the maximum page of the query, it will return an empty set. (Page starts from 0).
page_size (Optional)	query	int	The size of each page, the default is 1000.

Response

Name	In	Type	Description
id	body	string	ID of the clique.
_id	body	string	MongoDB ObjectId of the clique.
environment	body	string	Environment of the clique.
focal_point	body	string	Object ID of the focal point.
focal_point_type	body	string	Type of the focal point object, e.g. "vservice".
links	body	array	List of MongoDB ObjectIds of the links in the clique.
links_detailed	body	array	Details of the links in the clique.
constraints	body	object	Constraints of the clique.
cliques	body	array	The list of clique ids that match the filters.

Examples

Example Get Cliques

Request

http://10.56.20.32:8000/cliques?env_name=Mirantis-Liberty-API&link_id=58a2405a6a283a8bee15d42f

Response


```
{

   "cliques": [

          {
              "link_types": [
                 "instance-vnic",
                 "vservice-vnic",
                 "vnic-vconnector"
              ],
             "environment": "Mirantis-Liberty",
             "focal_point_type": "vnic",
             "id": "576c119a3f4173144c7a75c5"
          },

          {
             "link_types": [
                "vnic-vconnector",
                "vconnector-vedge"
             ],
             "environment": "Mirantis-Liberty",
             "focal_point_type": "vconnector",
```

```
            "id": "576c119a3f4173144c7a75c6"
        }

        ]

}
```


Example Get Clique Details

Request

http://korlev-calipso-testing.cisco.com:8000/cliques?env_name=Mirantis-Liberty-API&id=58a2406e6a283a8bee15d43f

Response


```
{
  'id': '58867db16a283a8bee15cd2b',
  'focal_point_type': 'host_pnic',
  'environment': 'Mirantis-Liberty',
  '_id': '58867db16a283a8bee15cd2b',
  'links_detailed': [
    {
      'state': 'up',
      'attributes': {
        'network': 'e180ce1c-eebc-4034-9e50-b3bab1c13979'
      },
      'target': '58867cc86a283a8bee15cc92',
      'source': '58867d166a283a8bee15ccd0',
      'host': 'node-1.cisco.com',
      'link_type': 'host_pnic-network',
      'target_id': 'e180ce1c-eebc-4034-9e50-b3bab1c13979',
      'source_id': 'eno16777728.103@eno16777728-00:50:56:ac:e8:97',
      'link_weight': 0,
      'environment': 'Mirantis-Liberty',
      '_id': '58867d646a283a8bee15ccf3',
      'target_label': '',
      'link_name': 'Segment-None',
      'source_label': ''
    }
  ],

'links': [
  '58867d646a283a8bee15ccf3'
 ],
'focal_point': '58867d166a283a8bee15ccd0',
'constraints': {
```

}

}

4.6 Clique_types

GET /clique_types

Description: get clique_type details with environment name and clique_type id, or get a list of clique_types with filters except id

Normal response code: 200

Error response code: badRequest(400), unauthorized(401), notFound(404)

Request

Name	In	Type	Description
env_name	query	string	Environment of the clique_types. e.g. "Mirantis-Liberty-API"
id	query	string	ID of the clique_type, it must be a string that can be converted to the MongoDB ObjectID.
focal_point_type (Optional)	query	string	Type of the focal point object, some possible values for it are "vnic", "vconnector", "vedge", "instance", "vservice", "host_pnic", "network", "port", "otep" and "agent".
link_type(Optional)	query	string	Type of the link, when this filter is specified, it will return all the clique_types which contain the specific link_type in its link_types array. Some possible values of the link_type are "instance-vnic", "otep-vconnector", "otep-host_pnic", "host_pnic-network", "vedge-otep", "vnic-vconnector", "vconnector-host_pnic", "vnic-vedge", "vedge-host_pnic" and "vservice-vnic". Repeat link_type several times to specify multiple link_types, e.g link_type=instance-vnic&link_type=host_pnic-network.
page_size(Optional)	query	int	Size of each page, the default is 1000.
page (Optional)	query	int	Which page is to be returned, the default is first page, if the page is larger than the maximum page of the query, it will return an empty result set. (Page starts from 0).

Response

Name	In	Type	Description
id	body	string	ID of the clique_type.
_id	body	string	MongoDB ObjectId of the clique_type
environment	body	string	Environment of the clique_type.
focal_point_type	body	string	Type of the focal point, e.g. "vnic".
link_types	body	array	List of link_types of the clique_type.
name	body	string	Name of the clique_type.
clique_types	body	array	List of clique_type ids of clique types that match the filters.

Examples

Example Get Clique_types

Request

http://korlev-calipso-testing.cisco.com:8000/clique_types?env_name=Mirantis-Liberty-API&link_type=instance-vnic&page_size=3&link_type=host_pnic-network

Response


```
{

   "clique_types": [

            {
                "environment": "Mirantis-Liberty",
                "focal_point_type": "host_pnic",
                "id": "58ca73ae3a8a836d10ff3b80"
            }

            ]

}
```

Example Get Clique_type Details

Request

http://korlev-calipso-testing.cisco.com:8000/clique_types?env_name=Mirantis-Liberty-API&id=585b183c761b05789ee3c659

Response


```
{
  'id': '585b183c761b05789ee3c659',
  'focal_point_type': 'vnic',
  'environment': 'Mirantis-Liberty-API',
  '_id': '585b183c761b05789ee3c659',
  'link_types': [
    'instance-vnic',
    'vservice-vnic',
    'vnic-vconnector'
  ],
  'name': 'vnic_clique'
}
```


POST /clique_types

Description: Create a new clique_type

Normal response code: 201(Created)

Error response code: badRequest(400), unauthorized(401), conflict(409)

Request

Name	In	Type	Description
environment(Mandatory)	body	string	Environment of the system, the environment must be the existing environment in the system.
focal_point_type(Mandatory)	body	string	Type of the focal point, some possible values are "vnic", "vconnector", "vedge", "instance", "vservice", "host_pnic", "network", "port", "otep" and "agent".
link_types(Mandatory)	body	array	Link_types of the clique_type, some possible values of the link_type are "instance-vnic", "otep-vconnector", "otep-host_pnic", "host_pnic-network", "vedge-otep", "vnic-vconnector", "vconnector-host_pnic", "vnic-vedge", "vedge-host_pnic" and "vservice-vnic"

name(Mandatory)	body	string	Name of the clique type, e.g. "instance_vconnector_clique"

Request Example

post http://korlev-calipso-testing.cisco.com:8000/clique_types

```
{
  "environment" : "RDO-packstack-Mitaka",
   "focal_point_type" : "instance",
   "link_types" : [
      "instance-vnic",
      "vnic-vconnector",
      "vconnector-vedge",
      "vedge-otep",
      "otep-host_pnic",
      "host_pnic-network"
   ],
   "name" : "instance_vconnector_clique"
}
```

Response

Successful Example

```
{
      "message": "created a new clique_type for environment Mirantis-Liberty"
}
```

4.7 Clique_constraints

GET /clique_constraints

Description: get clique_constraint details with clique_constraint id, or get a list of clique_constraints with filters except id.

Normal response code: 200

Error response code: badRequest(400), unauthorized(401), notFound(404)

Note: this is not environment specific so query starts with parameter, not env_name (as with all others), example:

http://korlev-calipso-testing.cisco.com:8000/clique_constraints?focal_point_type=instance

Request

Name	In	Type	Description
id (Optional)	query	string	ID of the clique_constraint, it must be a string that can be converted to MongoDB ObjectId.
focal_point_type (Optional)	query	string	Type of the focal_point, some possible values for that are "vnic", "vconnector", "vedge", "instance", "vservice", "host_pnic", "network", "port", "otep" and "agent".
constraint(Optional)	query	string	Constraint of the cliques, repeat this filter several times to specify multiple constraints. e.g constraint=network&constraint=host_pnic.
page (Optional)	query	int	Which page is to be returned, the default is the first page, if the page is larger than the maximum page of the query, the last page will be returned. (Page starts from 0.)
page_size (Optional)	query	int	Size of each page, the default is 1000

Response

Name	In	Type	Description
id	body	string	Object id of the clique constraint.
_id	body	string	MongoDB ObjectId of the clique_constraint.
focal_point_type	body	string	Type of the focal point object.
constraints	body	array	Constraints of the clique.
clique_constraints	body	array	List of clique constraints ids that match the filters.

Examples

Example Get Clique_constraints

Request

http://korlev-calipso-testing.cisco.com:8000/clique_constraints?constraint=host_pnic&constraint=network

Response


```
{
    "clique_constraints": [
            {
                "id": "576a4176a83d5313f21971f5"
            },
            {
                "id": "576ac7069f6ba3074882b2eb"
            }
   ]
}
```


Example Get Clique_constraint Details

Request

http://korlev-calipso-testing.cisco.com:8000/clique_constraints?id=576a4176a83d5313f21971f5

Response


```
{
    "_id": "576a4176a83d5313f21971f5",
    "constraints": [
        "network",
        "host_pnic"
    ],
    "id": "576a4176a83d5313f21971f5",
   "focal_point_type": "instance"
}
```

4.8 Scans

GET /scans

Description: get scan details with environment name and scan id, or get a list of scans with filters except id

Normal response code: 200

Error response code: badRequest (400), unauthorized (401), notFound(404)

Request

Name	In	Type	Description
env_name (Mandatory)	query	string	Environment of the scans. e.g. "Mirantis-Liberty".
id (Optional)	query	string	ID of the scan, it must be a string that can be converted MongoDB ObjectId.
base_object(Optional)	query	string	ID of the scanned base object. e.g. "node-2.cisco.com".
status (Optional)	query	string	Status of the scans, the possible values for the status are "draft", "pending", "running", "completed", "failed" and "aborted".
page (Optional)	query	int	Which page is to be returned, the default is the first page, if the page is larger than the maximum page of the query, it will return an empty set. (Page starts from 0.)
page_size (Optional)	query	int	Size of each page, the default is 1000.

Response

Name	In	Type	Description
status	body	string	The current status of the scan, possible values are "draft", "pending", "running", "completed", "failed" and "aborted".
log_level	body	string	Logging level of the scanning, the possible values are "CRITICAL", "ERROR", "WARNING", "INFO", "DEBUG" and "NOTSET".
clear	body	boolean	Indicates whether it needs to clear all the data before scanning.
scan_only_inventory	body	boolean	Only scan and store data in the inventory.

scan_only_links	body	boolean	Limit the scan to find only missing links.
scan_only_cliques	body	boolean	Limit the scan to find only missing cliques.
scan_completed	body	boolean	Indicates if the scan completed
submit_timestamp	body	string	Submit timestamp of the scan
environment	body	string	Environment name of the scan
inventory	body	string	Name of the inventory collection.
object_id	body	string	Base object of the scan

Examples

Example Get Scans

Request

http://korlev-calipso-testing.cisco.com:8000/scans?status=completed&env_name=Mirantis-Liberty&base_object=ff

Response


```
{
    "scans": [

        {
          "status": "pending",
          "environment": "Mirantis-Liberty",
         "id": "58c96a075eb66a121cc4e75f",
         "scan_completed": true
        }

    ]

}
```


Example Get Scan Details

Request

http://korlev-calipso-testing.cisco.com:8000/scans?env_name=Mirantis-Liberty&id=589a49cf2e8f4d154386c725

Response


```
{
    "scan_only_cliques": true,
```

```
    "object_id": "ff",
    "start_timestamp": "2017-01-28T01:02:47.352000",
    "submit_timestamp": null,
    "clear": true,
    "_id": "589a49cf2e8f4d154386c725",
    "environment": "Mirantis-Liberty",
    "scan_only_links": true,
    "id": "589a49cf2e8f4d154386c725",
    "inventory": "update-test",
    "scan_only_inventory": true,
    "log_level": "warning",
    "status": "completed",
    "end_timestamp": "2017-01-28T01:07:54.011000"
}
```


POST /scans

Description: create a new scan (ask calipso to scan an environment for detailed data gathering).

Normal response code: 201(Created)

Error response code: badRequest (400), unauthorized (401)

Request

Name	In	Type	Description
status (mandatory)	body	string	The current status of the scan, possible values are "draft", "pending", "running", "completed", "failed" and "aborted".
log_level (optional)	body	string	Logging level of the scanning, the possible values are "critical", "error", "warning", "info", "debug" and "notset".
clear (optional)	body	boolean	Indicates whether it needs to clear all the data before scanning.
scan_only_inventory (optional)	body	boolean	Only scan and store data in the inventory.
scan_only_links (optional)	body	boolean	Limit the scan to find only missing links.
scan_only_cliques (optional)	body	boolean	Limit the scan to find only missing cliques.
environment (mandatory)	body	string	Environment name of the scan
inventory (optional)	body	string	Name of the inventory collection.
object_id (optional)	body	string	Base object of the scan

Request Example

post http://korlev-calipso-testing.cisco.com:8000/scans

```
{
    "status" : "pending",
    "log_level" : "warning",
    "clear" : true,
    "scan_only_inventory" : true,
    "environment" : "Mirantis-Liberty",
    "inventory" : "koren",
    "object_id" : "ff"
}
```

Response

Successful Example

```
{
    "message": "created a new scan for environment Mirantis-Liberty"
}
```

4.9 Scheduled_scans

GET /scheduled_scans

Description: get scheduled_scan details with environment name and scheduled_scan id, or get a list of scheduled_scans with filters except id

Normal response code: 200

Error response code: badRequest (400), unauthorized (401), notFound(404)

Request

Name	In	Type	Description
environment(Mandatory)	query	string	Environment of the scheduled_scans. e.g. "Mirantis-Liberty".
id (Optional)	query	string	ID of the scheduled_scan, it must be a string that can be converted to MongoDB ObjectId.
freq (Optional)	query	string	Frequency of the scheduled_scans, the possible values for the freq are "HOURLY", "DAILY", "WEEKLY", "MONTHLY", and "YEARLY".
page (Optional)	query	int	Which page is to be returned, the default is the first page, if the page is larger than the maximum

			page of the query, it will return an empty set. (Page starts from 0.)
page_size (Optional)	query	int	Size of each page, the default is 1000.

Response

Name	In	Type	Description
freq	body	string	The frequency of the scheduled_scan, possible values are "HOURLY", "DAILY", "WEEKLY", "MONTHLY", and "YEARLY".
log_level	body	string	Logging level of the scheduled_scan, the possible values are "critical", "error", "warning", "info", "debug" and "notset".
clear	body	boolean	Indicates whether it needs to clear all the data before scanning.
scan_only_inventory	body	boolean	Only scan and store data in the inventory.
scan_only_links	body	boolean	Limit the scan to find only missing links.
scan_only_cliques	body	boolean	Limit the scan to find only missing cliques.
submit_timestamp	body	string	Submitted timestamp of the scheduled_scan
environment	body	string	Environment name of the scheduled_scan
scheduled_timestamp	body	string	Scheduled time for the scanning, it should follows ISO 8610: YYYY-MM-DDThh:mm:ss.sss+hhmm

Examples

Example Get Scheduled_scans

Request

http://korlev-calipso-testing.cisco.com:8000/scheduled_scans?environment=Mirantis-Liberty

Response


```
{
    "scheduled_scans": [

        {

          "freq":"WEEKLY",
          "environment": "Mirantis-Liberty",
          "id": "58c96a075eb66a121cc4e75f",
```

```
            "scheduled_timestamp": "2017-01-28T01:07:54.011000"
        }

    ]

}
```


Example Get Scheduled_Scan Details

Request

http://korlev-calipso-testing.cisco.com:8000/scheduled_scans?environment=Mirantis-Liberty&id=589a49cf2e8f4d154386c725

Response


```
{
    "scan_only_cliques": true,
    "scheduled_timestamp": "2017-01-28T01:02:47.352000",
    "submit_timestamp": 2017-01-27T01:07:54.011000"",
    "clear": true,
    "_id": "589a49cf2e8f4d154386c725",
    "environment": "Mirantis-Liberty",
    "scan_only_links":false,
    "id": "589a49cf2e8f4d154386c725",
    "scan_only_inventory":false,
    "log_level": "warning",
    "freq": "WEEKLY"
}
```


POST /scheduled_scans

Description: create a new scheduled_scan (request calipso to scan in a future date).

Normal response code: 201(Created)

Error response code: badRequest (400), unauthorized (401)

Request

Name	In	Type	Description
log_level (optional)	body	string	Logging level of the scheduled_scan, the possible values are "critical", "error", "warning", "info", "debug" and "notset".
clear (optional)	body	boolean	Indicates whether it needs to clear all the data before scanning.

scan_only_inventory (optional)	body	boolean	Only scan and store data in the inventory.
scan_only_links (optional)	body	boolean	Limit the scan to find only missing links.
scan_only_cliques (optional)	body	boolean	Limit the scan to find only missing cliques.
environment (mandatory)	body	string	Environment name of the scan
freq(mandatory)	body	string	The frequency of the scheduled_scan, possible values are "HOURLY", "DAILY", "WEEKLY", "MONTHLY", and "YEARLY".
submit_timestamp(mandatory)	body	string	Submitted time for the scheduled_scan, it should follows ISO 8610: YYYY-MM-DDThh:mm:ss.sss+hhmm

Post http://korlev-calipso-testing.cisco.com:8000/scheduled_scans

```
{
    "freq" : "WEEKLY",
    "log_level" : "warning",
    "clear" : true,
    "scan_only_inventory" : true,
    "environment" : "Mirantis-Liberty",
    "submit_timestamp" : "2017-01-28T01:07:54.011000"
}
```

Response

Successful Example

```
{
    "message": "created a new scheduled_scan for environment Mirantis-Liberty"
}
```

4.10 Constants

GET /constants

Description: get constant details with name (constants are used by ui and event/scan managers)

Normal response code: 200

Error response code: badRequest(400), unauthorized(401), notFound(404)

Request

Name	In	Type	Description
name (Mandatory)	query	string	Name of the constant. e.g. "distributions".

Response

Name	In	Type	Description
id	body	string	ID of the constant.
_id	body	string	MongoDB ObjectId of the constant.
name	body	string	Name of the constant.
data	body	array	Data of the constant.

Examples

Example Get Constant Details

Request

http://korlev-osdna-testing.cisco.com:8000/constants?name=link_states

Response


```
{
    "_id": "588796ac2e8f4d02b8e7aa2a",
    "data": [
        {
            "value": "up",
            "label": "up"
        },
        {
            "value": "down",
            "label": "down"
        }
    ],
    "id": "588796ac2e8f4d02b8e7aa2a",
    "name": "link_states"
}
```

4.11 Monitoring_Config_Templates

GET /monitoring_config_templates

Description: get monitoring_config_template details with template id, or get a list of templates with filters except id (see monitoring-guide).

Normal response code: 200

Error response code: badRequest(400), unauthorized(401), notFound(404)

Request

Name	In	Type	Description
id (Optional)	query	string	ID of the monitoring config template, it must be a string that can be converted MongoDB ObjectId
order (Optional)	query	int	Order by which templates are applied, 1 is the OSDNA default template. Templates that the user added later we use higher order and will override matching attributes in the default templates or add new attributes.
side (Optional)	query	string	The side which runs the monitoring, the possible values are "client" and "server".
type (Optional)	query	string	The name of the config file, e.g. "client.json".
page (Optional)	query	int	Which page is to be returned, the default is the first page, if the page is larger than the maximum page of the query, it will return an empty result set. (Page starts from 0).
page_size(Optional)	query	int	Size of each page, the default is 1000.

Response

Name	In	Type	Description
id	body	string	ID of the monitoring_config_template.
_id	body	srting	MongoDB ObjectId of the monitoring_config_template.
monitoring_system	body	string	System that we use to do the monitoring, e.g, "Sensu".
order	body	string	Order by which templates are applied, 1 is the OSDNA default templates. Templates that the user added later we use higher order and will override matching attributes in the default templates or add new attributes.
config	body	object	Configuration of the monitoring.
side	body	string	The side which runs the monitoring.
type	body	string	The name of the config file, e.g. "client.json".

Examples

Example Get Monitoring_config_templates

Request

http://korlev-calipso-testing.cisco.com:8000/monitoring_config_templates?side=client&order=1&type=rabbitmq.json&page=0&page_size=1

Response


```
{
    "monitoring_config_templates": [

            {
                "type": "rabbitmq.json",
                "side": "client",
                "id": "583711893e149c14785d6daa"
            }

    ]
}
```


Example Get Monitoring_config_template Details

Request

http://korlev-calipso-testing.cisco.com:8000/monitoring_config_templates?id=583711893e149c14785d6daa

Response


```
{
    "order": "1",
    "monitoring_system": "sensu",
    "_id": "583711893e149c14785d6daa",
    "side": "client",
    "type": "rabbitmq.json",
    "config": {
    "rabbitmq": {
    "host": "{server_ip}",
    "vhost": "/sensu",
    "password": "{rabbitmq_pass}",
    "user": "{rabbitmq_user}",
    "port": 5672
      }
```

```
    },
   "id": "583711893e149c14785d6daa"
}
```


4.12 Aggregates

GET /aggregates

Description: List some aggregated information about environment, message or constant.

Normal response code: 200

Error response code: badRequest(400), unauthorized(401), notFound(404)

Request

Name	In	Type	Description
env_name (Optional)	query	string	Environment name, if the aggregate type is "environment", this value must be specified.
type (Optional)	query	string	Type of aggregate, currently we support three types of aggregate, "environment", "message" and "constant".

Response

Name	In	Type	Description
type	body	string	Type of aggregate, we support three types of aggregates now, "environment", "message" and "constant".
env_name (Optional)	body	string	Environment name of the aggregate, when the aggregate type is "environment", this attribute will appear.
aggregates	body	object	The aggregates information.

Examples

Example Get Environment Aggregate

Request

http://korlev-calipso-testing.cisco.com:8000/aggregates?env_name=Mirantis-Liberty-API&type=environment

Response


```
{
    "env_name": "Mirantis-Liberty-API",
    "type": "environment",
    "aggregates": {
       "object_types": {
          "projects_folder": 1,
          "instances_folder": 3,
          "otep": 3,
          "region": 1,
          "vedge": 3,
          "networks_folder": 2,
          "project": 2,
          "vconnectors_folder": 3,
          "availability_zone": 2,
          "vedges_folder": 3,
          "regions_folder": 1,
          "network": 3,
          "vnics_folder": 6,
          "instance": 2,
         "vservice": 4,
         "availability_zones_folder": 1,
         "vnic": 8,
         "vservices_folder": 3,
         "port": 9,
         "pnics_folder": 3,
         "network_services_folder": 3,
         "ports_folder": 3,
         "host": 3,
         "vconnector": 6,
         "network_agent": 6,
         "aggregates_folder": 1,
         "pnic": 15,
         "network_agents_folder": 3,
         "vservice_miscellenaous_folder": 1
         }
    }
}
```


Example Get Messages Aggregate

Request

http://korlev-calipso-testing.cisco.com:8000/aggregates?type=message

Response


```
{
   "type": "message",
   "aggregates": {
      "levels": {
         "warn": 5,
         "info": 10,
         "error": 10
      },
     "environments": {
         "Mirantis-Liberty-API": 5,
         "Mirantis-Liberty": 10
      }
   }
}
```


Example Get Constants Aggregate

Request

http://korlev-calipso-testing.cisco.com:8000/aggregates?type=constant

Response


```
{
     "type": "constant",
     "aggregates": {
     "names": {
       "link_states": 2,
       "scan_statuses": 6,
       "type_drivers": 5,
       "log_levels": 6,
       "monitoring_sides": 2,
```

```
        "mechanism_drivers": 5,
        "messages_severity": 8,
        "distributions": 16,
        "link_types": 11,
        "object_types": 10
       }
      }
}
```


4.13 Environment_configs

GET /environment_configs

Description: get environment_config details with name, or get a list of environments_config with filters except name

Normal response code: 200

Error response code: badRequest(400), unauthorized(401), notFound(404)

Request

Name	In	Type	Description
name(Optional)	query	string	Name of the environment.
distribution(Optional)	query	string	The distribution of the OpenStack environment, it must be one of the distributions we support, e.g "Mirantis-8.0".(you can get all the supported distributions by querying the distributions constants)
mechanism_drivers(Optional)	query	string	The mechanism drivers of the environment, it should be one of the drivers in mechanism_drivers constants, e.g "ovs".
type_drivers(Optional)	query	string	'flat', 'gre', 'vlan', 'vxlan'.
user(Optional)	query	string	name of the environment user
listen(Optional)	query	boolean	Indicates whether the environment is being listened.
scanned(Optional)	query	boolean	Indicates whether the environment has been scanned.
monitoring_setup_done(Optional)	query	boolean	Indicates whether the monitoring setup has been done.
operational(Optional)	query	string	operational status of the environment, the possible statuses are "stopped", "running" and "error".

page(Optional)	query	int	Which page is to be returned, the default is the first page, if the page is larger than the maximum page of the query, it will return an empty result set. (Page starts from 0).
page_size(Optional)	query	int	Size of each page, the default is 1000.

Response

Name	In	Type	Description
configuration	body	array	List of configurations of the environment, including configurations of mysql, OpenStack, CLI, AMQP and Monitoring.
distribution	body	string	The distribution of the OpenStack environment, it must be one of the distributions we support, e.g "Mirantis-8.0".
last_scanned	body	string	The date of last time scanning the environment, the format of the date is MM/DD/YY.
mechanism_dirvers	body	array	The mechanism drivers of the environment, it should be one of the drivers in mechanism_drivers constants.
monitoring_setup_done	body	boolean	Indicates whether the monitoring setup has been done.
name	body	string	Name of the environment.
operational	body	boolean	Indicates if the environment is operational.
scanned	body	boolean	Indicates whether the environment has been scanned.
type	body	string	Production, testing, development, etc.
type_drivers	body	string	'flat', 'gre', 'vlan', 'vxlan'.
user	body	string	The user of the environment.
listen	body	boolean	Indicates whether the environment is being listened.

Examples

Example Get Environments config

Request

http://korlev-calipso-testing.cisco.com:8000/environment_configs?mechanism_drivers=ovs

Response


```
{
    environment_configs: [

        {
            "distribution": "Canonical-icehouse",
            "name": "thundercloud"
        }

    ]
}
```


Example Environment config Details

Request

http://korlev-calipso-testing.cisco.com:8000/environment_configs?name=Mirantis-Mitaka-2

Response


```
{
    "type_drivers": "vxlan",
    "name": "Mirantis-Mitaka-2",
    "app_path": "/home/yarony/osdna_prod/app",
    "scanned": true,
    "type": "environment",
    "user": "test",
    "distribution": "Mirantis-9.1",
    "monitoring_setup_done": true,
    "listen": true,
    "mechanism_drivers": [
        "ovs"
    ],
    "configuration": [
    {
        "name": "mysql",
        "user": "root",
        "host": "10.56.31.244",
```

```
            "port": "3307",
            "password": "TsbQPwP2VPIUlcFShkCFwBjX"
        },
        {
            "name": "CLI",
            "user": "root",
            "host": "10.56.31.244",
            "key": "/home/ilia/Mirantis_Mitaka_id_rsa"
        },
        {
            "password": "G1VfxeJmtK5vIyNNMP4qZmXB",
            "user": "nova",
            "name": "AMQP",
            "port": "5673",
            "host": "10.56.31.244"
        },
        {
            "server_ip": "korlev-nsxe1.cisco.com",
            "name": "Monitoring",
            "port": "4567",
            "env_type": "development",
            "rabbitmq_pass": "sensuaccess",
            "rabbitmq_user": "sensu",
            "provision": "DB",
            "server_name": "devtest-sensu",
            "type": "Sensu",
            "config_folder": "/tmp/sensu_test"
        },
        {
            "user": "admin",
            "name": "OpenStack",
            "port": "5000",
            "admin_token": "qoeROniLLwFmoGixgun5AXaV",
            "host": "10.56.31.244",
            "pwd": "admin"
        }
    ],
    "_id": "582d77ee3e149c1318b3aa54",
    "operational": "yes"
}
```

POST /environment_configs

Description: create a new environment configuration.

Normal response code: 201(Created)

Error response code: badRequest(400), unauthorized(401), notFound(404), conflict(409)

Request

Name	In	Type	Description
configuration(Mandatory)	body	array	List of configurations of the environment, including configurations of mysql(mandatory), OpenStack(mandatory), CLI(mandatory), AMQP(mandatory) and Monitoring(Optional).
distribution(Mandatory)	body	string	The distribution of the OpenStack environment, it must be one of the distributions we support, e.g "Mirantis-8.0".(you can get all the supported distributions by querying the distributions constants)
last_scanned(Optional)	body	string	The date and time of last scanning, it should follows ISO 8610: YYYY-MM-DDThh:mm:ss.sss+hhmm
mechanism_dirvers(Mandatory)	body	array	The mechanism drivers of the environment, it should be one of the drivers in mechanism_drivers constants, e.g "OVS".
name(Mandatory)	body	string	Name of the environment.
operational(Mandatory)	body	boolean	Indicates if the environment is operational. e.g. true.
scanned(Optional)	body	boolean	Indicates whether the environment has been scanned.
listen(Mandatory)	body	boolean	Indicates if the environment need to been listened.
user(Optional)	body	string	The user of the environment.
app_path(Mandatory)	body	string	The path that the app is located in.
type(Mandatory)	body	string	Production, testing, development, etc.
type_drivers(Mandatory)	body	string	'flat', 'gre', 'vlan', 'vxlan'.

Request Example

Post http://korlev-calipso-testing:8000/environment_configs


```
{
    "app_path" : "/home/korenlev/OSDNA/app/",
    "configuration" : [
        {
            "host" : "172.23.165.21",
            "name" : "mysql",
            "password" : "password",
            "port" : NumberInt(3306),
            "user" : "root",
            "schema" : "nova"
        },
        {
            "name" : "OpenStack",
            "host" : "172.23.165.21",
            "admin_token" : "TL4T0I7qYNiUifH",
            "admin_project" : "admin",
            "port" : "5000",
            "user" : "admin",
            "pwd" : "admin"
        },
        {
            "host" : "172.23.165.21",
            "key" : "/home/yarony/.ssh/juju_id_rsa",
            "name" : "CLI",
            "user" : "ubuntu"
        },
        {
            "name" : "AMQP",
            "host" : "10.0.0.1",
            "port" : "5673",
            "user" : "User",
            "password" : "abcd1234"
        },
        {
            "config_folder" : "/tmp/sensu_test_liberty",
            "provision" : "None",
            "env_type" : "development",
            "name" : "Monitoring",
            "port" : "4567",
            "rabbitmq_pass" : "sensuaccess",
            "rabbitmq_user" : "sensu",
            "server_ip" : "korlev.cisco.com",
```

```
            "server_name" : "devtest-sensu",
            "type" : "Sensu"
        }
      ],
      "distribution" : "Canonical-icehouse",
      "last_scanned" : "2017-02-13T16:07:15Z",
      "listen" : true,
      "mechanism_drivers" : [
            "OVS"
       ],
       "name" : "thundercloud",
       "operational" : "yes",
       "scanned" : false,
       "type" : "environment",
        "type_drivers" : "gre",
        "user" : "WS7j8oTbWPf3LbNne"
}
```


Response

Successful Example


```
{
      "message": "created environment_config for Mirantis-Liberty"
}
```