HP Helion OpenStack® Carrier Grade 1.1: REST API Guide

HP Helion OpenStack Carrier Grade REST API

Table of Contents

Preface

Use the Inventory APIs and the extensions to OpenStack APIs to manage your HP Helion OpenStack Carrier Grade cloud.

- Compute API v2 extensions

- Networking API v2.0 extensions

- Block Storage API v2 extensions

- Telemetry API v2 extensions

- System Inventory API v1

- Patching Service API v1

After you authenticate through the Identity API, you can use the other APIs and extensions to manage controller/compute/storage nodes, launch server instances, create images, assign metadata to instances and images, create containers and objects, and complete other actions in your HP Helion OpenStack Carrier Grade cloud. To get started with the APIs, see the OpenStack API Quick Start.

1. REST API Overview

The HP Helion OpenStack Carrier Grade is a high-performance, high-availability, NFV-enabled cloud operating system that enables telecommunication operators to deploy and maintain next generation communication services. It brings together the flexibility and scalability of the IT cloud, and the high-availability and performance demanded by the Telecommunications industry, into a unique industry-leading solution. The CG Comms Server leverages the power of Commercial Off-The-Shelf (COTS) server hardware to deliver a price-performance ratio well above alternative solutions.

The CG Comms Server leverages core virtualization and carrier-grade technologies, Intel DPDK high-performance packet processing, open architectures, and the OpenStack software suite. It extends the OpenStack software suite to implement a unique carrier-grade, high-availability architecture, on which high-performance production systems can be deployed.

Method	URI	Description
OpenStack REST API Documentation		
Strategy for REST API Documentation		
Overview of Document		

1.1. OpenStack REST API Documentation

The OpenStack REST API Documentation is open-source; both the published formats (PDF and HTML pages) as well as the docbook/wadl source documents. The published formats can be found at http://developer.openstack.org/api-ref.html .

Although some of the general open-source OpenStack Documentation (e.g. Admin Guide) is versioned based on OpenStack Releases (e.g. Havana), the open-source OpenStack REST API Documentation is currently NOT versioned. There is only a master/current version of the OpenStack REST API Documentation containing updates from 'all' versions of OpenStack.

1.2. Strategy for REST API Documentation

Due to the lack of versioning on the open-source OpenStack REST API Documentation, the HP Helion OpenStack Carrier Grade REST API Documentation will not build on top of open-source documents. Instead, the HP Helion OpenStack Carrier Grade REST API Documenation will provide a description of the delta between the HP Helion OpenStack Carrier Grade REST API and the open-source OpenStack REST API. For existing OpenStack Components (e.g. Nova, Neutron, ...), any modified APIs will be re-documented with any modified or additional request parameters, response attributes and/or behaviour. For any new HP Helion OpenStack Carrier Grade Components (e.g. SysInv), all APIs will be fully documented.

Additionally, only high-level description of the semantics around objects and parameters are described in this REST API Documentation. For full description of objects and parameters, please refer to the HP Helion OpenStack Carrier Grade Administrator Guide.

1.3. Overview of Document

The sections of this document consist of:

- Extensions to Compute REST API

- Extensions to Networking REST API

- Extensions to Telemetry REST API

- Extensions to Block Storage REST API

- System Inventory (SysInv) REST API

- Patching REST API

There are no Extensions to Image REST API (Glance), Identity REST API (Keystone) and Orchestration REST API (Heat).

2. Compute API v2 extensions

HP Helion OpenStack Carrier Grade extensions to the OpenStack Compute API include:

- Adding capability to specify VIF-Model on a per-NIC basis when creating/launching/booting a VM Server.

- Adding an attribute to allow sorting of a VM Server's IP Addresses in order of network attachment(NIC).

- Adding the ability to scale the resources (currently only vCPUs) of a VM Server up and down without requiring a restart of the VM Server.

- Adding a number of new attributes and behavior to VM Server Groups; a project_id attribute for ownership, an affinity-hyperthread policy and best_effort and max group_size metadata attribute values.

- Support for various new Flavor Extra Specs to enable HP Helion OpenStack Carrier Grade optimizations and capabilities for servers.

The typical port used for the Compute REST API is 8774. However, proper technique would be to look up the nova service endpoint in keystone.

Method	URI	Description
	Extensions	
GET	`/v2/{tenant_id}/extensions`	Lists all extensions.
GET	`/v2/{tenant_id}/extensions/{extension_alias}`	Gets information about a specified extension.
	Server	
POST	`/v2/{tenant_id}/servers`	Creates a server.
GET	`/v2/{tenant_id}/servers/detail`	Lists details of servers.
GET	`/v2/{tenant_id}/servers/{server_id}`	Shows details for a specified server.
POST	`/v2/{tenant_id}/servers/{server_id}/action`	Allows the resources associated with the server (currently only the number of CPUs) to be scaled up and down without requiring a restart of the VM.
	Server Groups (os-server-groups)	
GET	`/v2/{tenant_id}/os-server-groups`	Lists server groups.
POST	`/v2/{tenant_id}/os-server-groups`	Creates a server group.
GET	`/v2/{tenant_id}/os-server-groups/{ServerGroup_id}`	Shows details for a specified server group.
	Flavor Extra Specs	
GET	`/v2/{tenant_id}/flavors/{flavor_id}/os-extra_specs`	Lists the extra-specs or keys for the specified flavor.
GET	`/v2/{tenant_id}/flavors/{flavor_id}/os-extra_specs/{key_id}`	Gets the value of the specified key.
POST	`/v2/{tenant_id}/flavors/{flavor_id}/os-extra_specs`	Creates extra-specs or keys for the specified flavor.

2.1. Extensions

The Extensions entity lists all available extensions

Method	URI	Description
GET	`/v2/{tenant_id}/extensions`	Lists all extensions.
GET	`/v2/{tenant_id}/exten-sions/{extension_alias}`	Gets information about a specified extension.

2.1.1. List extensions

Method	URI	Description
GET	`/v2/{tenant_id}/extensions`	Lists all extensions.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

2.1.1.1. Request

This table shows the URI parameters for the list extensions request:

Name	Type	Description
`{tenant_id}`	String	The ID for the tenant or account in a multi-tenancy cloud.

This operation does not accept a request body.

2.1.1.2. Response

This table shows the body parameters for the list extensions response:

Name	Type	Description
`namespace`	String *(Optional)*	Indicates namespace of the extension.
`name`	String *(Optional)*	Indicates name of the extension.
`updated`	String *(Optional)*	Indicates updated time of the extension.
`description`	String *(Optional)*	Indicates description of the extension.
`alias`	String *(Optional)*	Indicates alias of the extension.
`links`	List *(Optional)*	A list of links for the extension.

Example 2.1. List extensions: JSON response

```
{
    "extensions" : [
        ...
        {
            "namespace" : "http://docs.windriver.com/tis/ext/wrs-if/api/v1.0",
            "name" : "wrs-server-if",
            "updated" : "2014-08-16T00:00:00+00:00",
```

```
            "description" : "WRS Interface Related Extensions",
            "alias" : "wrs-if",
            "links" : []
        },
        {
            "namespace" : "http://docs.windriver.com/tis/ext/wrs-res/api/v1.0",
            "name" : "wrs-server-resource",
            "updated" : "2014-08-16T00:00:00+00:00",
            "description" : "WRS Server Resource Extensions",
            "alias" : "wrs-res",
            "links" : []
        },
        {
            "namespace" : "http://docs.windriver.com/tis/ext/wrs-sg/api/v1.0",
            "name" : "wrs-server-group",
            "updated" : "2014-08-16T00:00:00+00:00",
            "description" : "WRS Server Grouip Extensions",
            "alias" : "wrs-sg",
            "links" : []
        },
        {
            "namespace" : "http://docs.windriver.com/tis/ext/wrs/api/v1.0",
            "name" : "wrs-flavor-extra-spec",
            "updated" : "2014-08-16T00:00:00+00:00",
            "description" : "WRS Extended Flavor Extra Spec Keys"
            "alias" : "wrs",
            "links" : []
        },
        ...
    ]
}
```

This operation does not return a response body.

2.1.2. Get extension

Method	URI	Description
GET	`/v2/{tenant_id}/exten-sions/{extension_alias}`	Gets information about a specified extension.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

2.1.2.1. Request

This table shows the URI parameters for the get extension request:

Name	Type	Description
`{tenant_id}`	String	The ID for the tenant or account in a multi-tenancy cloud.
`{extension_alias}`	String	The alias for the extension to list.

This operation does not accept a request body.

2.1.2.2. Response

This table shows the body parameters for the get extension response:

Name	Type	Description
`namespace`	String *(Optional)*	Indicates namespace of the extension.
`name`	String *(Optional)*	Indicates name of the extension.
`updated`	String *(Optional)*	Indicates updated time of the extension.
`description`	String *(Optional)*	Indicates description of the extension.
`alias`	String *(Optional)*	Indicates alias of the extension.
`links`	List *(Optional)*	A list of links for the extension.

Example 2.2. Get extension: JSON response

```
{
   "extension" : {
      "namespace" : "http://docs.windriver.com/tis/ext/wrs-if/api/v1.0",
      "name" : "wrs-server-if",
      "updated" : "2014-08-16T00:00:00+00:00",
      "description" : "WRS Interface Related Extensions",
```

```
            "alias" : "wrs-if",
            "links" : []
        }
    }

OR

    {
        "extension" : {
            "namespace" : "http://docs.windriver.com/tis/ext/wrs-res/api/v1.0",
            "name" : "wrs-server-resource",
            "updated" : "2014-08-16T00:00:00+00:00",
            "description" : "WRS Server Resource Extensions",
            "alias" : "wrs-res",
            "links" : []
        }
    }

OR

    {
        "extension" : {
            "namespace" : "http://docs.windriver.com/tis/ext/wrs-sg/api/v1.0",
            "name" : "wrs-server-group",
            "updated" : "2014-08-16T00:00:00+00:00",
            "description" : "WRS Server Group Extensions",
            "alias" : "wrs-sg",
            "links" : []
        }
    }

OR

    {
        "extension" : {
            "namespace" : "http://docs.windriver.com/tis/ext/wrs/api/v1.0",
            "name" : "wrs-flavor-extra-spec",
            "updated" : "2014-08-16T00:00:00+00:00",
            "description" : "WRS Extended Flavor Extra Spec Keys"
            "alias" : "wrs",
            "links" : []
        }
    }
```

This operation does not return a response body.

2.2. Server

The extensions to the server entity are:

- Adding capability to specify VIF-Model on a per-NIC basis when creating/launching/booting a VM Server.

- Adding the ability to scale the resources (currently only vCPUs) of a server up and down without requiring a restart of the VM Server.

- Adding an attribute to allow sorting of a VM Server's IP Addresses in order of network attachment(NIC).

Method	URI	Description
POST	`/v2/{tenant_id}/servers`	Creates a server.
GET	`/v2/{tenant_id}/servers/detail`	Lists details of servers.
GET	`/v2/{tenant_id}/servers/` `{server_id}`	Shows details for a specified server.
POST	`/v2/{tenant_id}/servers/` `{server_id}/action`	Allows the resources associated with the server (currently only the number of CPUs) to be scaled up and down without requiring a restart of the VM.

2.2.1. Create Server

Method	URI	Description
POST	`/v2/{tenant_id}/servers`	Creates a server.

Normal response codes: 200

2.2.1.1. Request

This table shows the URI parameters for the create server request:

Name	Type	Description
`{tenant_id}`	String	The ID for the tenant or account in a multi-tenancy cloud.

This table shows the body parameters for the create server request:

Name	Type	Description
`networks`	List *(Optional)*	A `networks` object. By default, the server instance is provisioned with all isolated networks for the tenant. Optionally, you can create one or more NICs on the server. To provision the server instance with a NIC for a `nova-network` network, specify the UUID in the `uuid` attribute in a `network` object. To provision the server instance with a NIC for a `neutron` network, specify the UUID in the `port` attribute in a `network` object. **In TiS, to optionally provision the vif model of the NIC, specify the appropriate value in the `wrs-if:vif_model` attribute in the `network` object. Valid vif model values are: `e1000`, `virtio`, `ne2k_pci`, `pcnet`, `rtl8139`, `avp`, `pci-passthrough`, `pci-sriov`. If not specified, a vif model of `virtio` will be used.** You can specify multiple NICs on the server.

Example 2.3. Create Server: JSON request

```
{
  "server": {
    "name": "testvm",
    "imageRef": "ec42f67b-1dcd-4f09-aa02-7a426737c20a",
    "flavorRef": "2",
    "networks": [
      {
        "wrs-if:vif_model": "e1000",
        "uuid": "06937e9e-0acd-4ad5-a6bb-f82d8896d5e8"
      },
      {
        "wrs-if:vif_model": "avp",
        "uuid": "cdc149b5-9122-4a16-975c-6acb973f49c3"
      },
      {
        "uuid": "b7adf5a0-3c5a-47f3-b733-8d56d12d2f45"
      }
    ]
  }
}
```

This operation does not accept a request body.

2.2.1.2. Response

Example 2.4. Create Server: JSON response

```
{
    "server": {
        "adminPass": "yjzytFHb7XHc",
        "id": "f8f4f3ce-f6e0-4e05-8f79-bf984fdfce45",
        "links": [
            {
                "href": "http://openstack.example.com/v2/openstack/servers/
f8f4f3ce-f6e0-4e05-8f79-bf984fdfce45",
                "rel": "self"
            },
            {
                "href": "http://openstack.example.com/openstack/servers/
f8f4f3ce-f6e0-4e05-8f79-bf984fdfce45",
                "rel": "bookmark"
            }
        ]
    }
}
```

2.2.2. List servers

Method	URI	Description
GET	`/v2/{tenant_id}/servers/detail`	Lists details of servers.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

2.2.2.1. Request

This table shows the URI parameters for the list servers request:

Name	Type	Description
`{tenant_id}`	String	The ID for the tenant or account in a multi-tenancy cloud.

This operation does not accept a request body.

2.2.2.2. Response

This table shows the body parameters for the list servers response:

Name	Type	Description
`servers`	List (Required)	The list of `server` objects.
`nics`	List (Optional)	A `nics` object. Contains the list of NICs provisioned on the server instance. Optionally, in TiS, each NIC can contain a `wrs-if:vif_model` attribute specifying the NICs vif model; where valid vif model values are: `e1000`, `virtio`, `ne2k_pci`, `pcnet`, `rtl8139`, `avp`, `pci-passthrough`, `pci-sriov`. If not specified, a vif model of `virtio` is being used.
`addresses`	List (Optional)	An `addresses` object. Contains the list of addresses associated with the server instance. In TiS, a `wrs-if:index` attribute has been added in order to allow sorting of these addresses in order of network attachment (NIC).
`wrs-res:topology`	String (Optional)	This attribute specifies a number of resource details of the VM Server; the number of numa nodes, the amount of memory and the memory page size, and the current number of VCPUs.
`wrs-res:vcpus (min/cur/max)`	List (Optional)	This attribute specifies the minimum number of vcpus, current number of vcpus and maximum number of vcpus of a VM Server.
`wrs-sg:server_group`	List (Optional)	This attribute specifies the server group which the VM Server is in; a null-string if the VM Server is not in a server group.

Example 2.5. List servers: JSON response

```
{
    "servers" : [
        {
            "accessIPv4" : "",
            "nics" : [
```

```
            {
                "nic1" : {
                    "network" : "tenant1-mgmt-net",
                    "port_id" : "dc627524-64a9-4fec-957a-b271f353fb22",
                    "wrs-if:vif_model" : "virtio",
                    "mtu" : 1500
                }
            }
        ],
        "OS-EXT-SRV-ATTR:instance_name" : "instance-0000003d",
        "OS-SRV-USG:terminated_at" : null,
        "accessIPv6" : "",
        "config_drive" : "",
        "OS-DCF:diskConfig" : "MANUAL",
        "wrs-sg:server_group" : "",
        "updated" : "2015-04-01T20:32:57Z",
        "metadata" : {},
        "id" : "770a214c-5d22-42ce-9273-f6baab0ad7fd",
        "flavor" : {
            "id" : "00bbded9-318a-461a-aef8-3904356ca8d9",
            "links" : [
                {
                    "rel" : "bookmark",
                    "href" : "http://128.224.151.243:8774/
101d1cffc5ec4accbdb075c89a4c5cd7/flavors/00bbded9-318a-461a-aef8-3904356ca8d9"
                }
            ]
        },
        "links" : [
            {
                "rel" : "self",
                "href" : "http://128.224.151.243:8774/v2/
101d1cffc5ec4accbdb075c89a4c5cd7/servers/770a214c-5d22-42ce-9273-f6baab0ad7fd"
            },
            {
                "rel" : "bookmark",
                "href" : "http://128.224.151.243:8774/
101d1cffc5ec4accbdb075c89a4c5cd7/servers/770a214c-5d22-42ce-9273-f6baab0ad7fd"
            }
        ],
        "OS-EXT-SRV-ATTR:host" : "compute-0",
        "OS-EXT-AZ:availability_zone" : "nova",
        "name" : "vm07-shared-vcpu-id",
        "hostId" :
 "938254ae1b04aabc901dd4ad2cf2a561a4eab858efa0b0a48eb048ff",
        "user_id" : "13dbcb9d22474c39a4a612cd44bf58ad",
        "status" : "ACTIVE",
        "wrs-res:topology" : "node:1,  1024MB, pgsize:4K, vcpus:3",
        "OS-EXT-STS:power_state" : 1,
        "OS-EXT-SRV-ATTR:hypervisor_hostname" : "compute-0",
        "tenant_id" : "101d1cffc5ec4accbdb075c89a4c5cd7",
        "OS-SRV-USG:launched_at" : "2015-04-01T20:32:57.000000",
        "OS-EXT-STS:vm_state" : "active",
        "OS-EXT-STS:task_state" : null,
        "progress" : 0,
        "key_name" : null,
        "image" : {
            "id" : "a99dfaa7-c850-4a63-ad99-d4a5f8da3069",
            "links" : [
                {
```

```
                    "rel" : "bookmark",
                    "href" : "http://128.224.151.243:8774/
101d1cffc5ec4accbdb075c89a4c5cd7/images/a99dfaa7-c850-4a63-ad99-d4a5f8da3069"
                }
            ]
        },
        "wrs-res:vcpus (min/cur/max)" : [
            3,
            3,
            3
        ],
        "created" : "2015-04-01T20:32:49Z",
        "addresses" : {
            "tenant1-mgmt-net" : [
                {
                    "wrs-if:index" : 1,
                    "OS-EXT-IPS:type" : "fixed",
                    "version" : 4,
                    "OS-EXT-IPS-MAC:mac_addr" : "fa:16:3e:fc:65:81",
                    "addr" : "192.168.102.6"
                }
            ]
        },
        "os-extended-volumes:volumes_attached" : []
    }
  ]
}
```

This operation does not return a response body.

2.2.3. Show server

Method	URI	Description
GET	`/v2/{tenant_id}/servers/` `{server_id}`	Shows details for a specified server.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

2.2.3.1. Request

This table shows the URI parameters for the show server request:

Name	Type	Description
`{tenant_id}`	String	The ID for the tenant or account in a multi-tenancy cloud.
`{server_id}`	UUID	The ID for the server of interest to you.

This operation does not accept a request body.

2.2.3.2. Response

This table shows the body parameters for the show server response:

Name	Type	Description
`server`	Dict *(Required)*	The requested `server` object.
`nics`	List *(Optional)*	A `nics` object. Contains the list of NICs provisioned on the server instance. Optionally, in TiS, each NIC can contain a `wrs-if:vif_model` attribute specifying the NICs vif model; where valid vif model values are: `e1000`, `virtio`, `ne2k_pci`, `pcnet`, `rtl8139`, `avp`, `pci-passthrough`, `pci-sriov`. If not specified, a vif model of `virtio` is being used.
`addresses`	List *(Optional)*	An `addresses` object. Contains the list of addresses associated with the server instance. In TiS, a `wrs-if:index` attribute has been added in order to allow sorting of these addresses in order of network attachment (NIC).
`wrs-res:topology`	String *(Optional)*	This attribute specifies a number of resource details of the VM Server; the number of numa nodes, the amount of memory and the memory page size, and the current number of VCPUs.
`wrs-res:vcpus (min/cur/` `max)`	List *(Optional)*	This attribute specifies the minimum number of vcpus, current number of vcpus and maximum number of vcpus of a VM Server.
`wrs-sg:server_group`	List	This attribute specifies the server group which the VM Server is in; a null-string if the VM Server is not in a server group.

Name	Type	Description
	(Optional)	

Example 2.6. Show server: JSON response

```
{
   "server" : {
      "accessIPv4" : "",
      "nics" : [
         {
            "nic1" : {
               "network" : "tenant1-mgmt-net",
               "port_id" : "dc627524-64a9-4fec-957a-b271f353fb22",
               "wrs-if:vif_model" : "virtio",
               "mtu" : 1500
            }
         }
      ],
      "OS-EXT-SRV-ATTR:instance_name" : "instance-0000003d",
      "OS-SRV-USG:terminated_at" : null,
      "accessIPv6" : "",
      "config_drive" : "",
      "OS-DCF:diskConfig" : "MANUAL",
      "wrs-sg:server_group" : "",
      "updated" : "2015-04-01T20:32:57Z",
      "metadata" : {},
      "id" : "770a214c-5d22-42ce-9273-f6baab0ad7fd",
      "flavor" : {
         "id" : "00bbded9-318a-461a-aef8-3904356ca8d9",
         "links" : [
            {
               "rel" : "bookmark",
               "href" : "http://128.224.151.243:8774/
101d1cffc5ec4accbdb075c89a4c5cd7/flavors/00bbded9-318a-461a-aef8-3904356ca8d9"
            }
         ]
      },
      "links" : [
         {
            "rel" : "self",
            "href" : "http://128.224.151.243:8774/v2/
101d1cffc5ec4accbdb075c89a4c5cd7/servers/770a214c-5d22-42ce-9273-f6baab0ad7fd"
         },
         {
            "rel" : "bookmark",
            "href" : "http://128.224.151.243:8774/
101d1cffc5ec4accbdb075c89a4c5cd7/servers/770a214c-5d22-42ce-9273-f6baab0ad7fd"
         }
      ],
      "OS-EXT-SRV-ATTR:host" : "compute-0",
      "OS-EXT-AZ:availability_zone" : "nova",
      "name" : "vm07-shared-vcpu-id",
      "hostId" : "938254ae1b04aabc901dd4ad2cf2a561a4eab858efa0b0a48eb048ff",
      "user_id" : "13dbcb9d22474c39a4a612cd44bf58ad",
      "status" : "ACTIVE",
      "wrs-res:topology" : "node:1,  1024MB, pgsize:4K, vcpus:3",
      "OS-EXT-STS:power_state" : 1,
      "OS-EXT-SRV-ATTR:hypervisor_hostname" : "compute-0",
      "tenant_id" : "101d1cffc5ec4accbdb075c89a4c5cd7",
      "OS-SRV-USG:launched_at" : "2015-04-01T20:32:57.000000",
```

```
        "OS-EXT-STS:vm_state" : "active",
        "OS-EXT-STS:task_state" : null,
        "progress" : 0,
        "key_name" : null,
        "image" : {
           "id" : "a99dfaa7-c850-4a63-ad99-d4a5f8da3069",
           "links" : [
              {
                 "rel" : "bookmark",
                 "href" : "http://128.224.151.243:8774/
101d1cffc5ec4accbdb075c89a4c5cd7/images/a99dfaa7-c850-4a63-ad99-d4a5f8da3069"
              }
           ]
        },
        "wrs-res:vcpus (min/cur/max)" : [
           3,
           3,
           3
        ],
        "created" : "2015-04-01T20:32:49Z",
        "addresses" : {
           "tenant1-mgmt-net" : [
              {
                 "wrs-if:index" : 1,
                 "OS-EXT-IPS:type" : "fixed",
                 "version" : 4,
                 "OS-EXT-IPS-MAC:mac_addr" : "fa:16:3e:fc:65:81",
                 "addr" : "192.168.102.6"
              }
           ]
        },
        "os-extended-volumes:volumes_attached" : []
     }
}
```

This operation does not return a response body.

2.2.4. Scale Server Up or Down

Method	URI	Description
POST	`/v2/{tenant_id}/servers/` `{server_id}/action`	Allows the resources associated with the server (currently only the number of CPUs) to be scaled up and down without requiring a restart of the VM.

Normal response codes: 202

2.2.4.1. Request

This table shows the URI parameters for the scale server up or down request:

Name	Type	Description
`{tenant_id}`	String	The ID for the tenant or account in a multi-tenancy cloud.
`{server_id}`	UUID	The ID for the server of interest to you.

This table shows the body parameters for the scale server up or down request:

Name	Type	Description
`wrs-res:scale`	String *(Required)*	Specify the `wrs-res:scale` action in the request body.
`direction`	String *(Required)*	Direction to scale, "up" or "down". This will result in scaling the specified resource by one unit in the specified direction.
`resource`	String *(Required)*	Resource to scale. Currently only "cpu" is supported.

Example 2.7. Scale Server Up or Down: JSON request

```
{
    "wrs-res:scale": {
        "direction": "up",
        "resource": cpu
    }
}
```

This operation does not accept a request body.

2.3. Server Groups (os-server-groups)

The extensions to the Server Groups entity are:

- Added a 'wrs-sg:project_id' attribute to assign tenant ownership to a Server Group.

- Added a 'wrs-sg:affinity-hyperthread' policy to indicate that members of the Server Group are allowed to share hyperthread siblings.

- Added a boolean 'wrs-sg:best_effort' metadata key/value in order to specify whether the policy should be strictly enforced or not.

- Added an integer 'wrs-sg:group_size' metadata key/value in order to specify the maximum number of members in the group.

Method	URI	Description
GET	`/v2/{tenant_id}/os-server-groups`	Lists server groups.
POST	`/v2/{tenant_id}/os-server-groups`	Creates a server group.
GET	`/v2/{tenant_id}/os-server-groups/{ServerGroup_id}`	Shows details for a specified server group.

2.3.1. List server groups

Method	URI	Description
GET	`/v2/{tenant_id}/os-server-groups`	Lists server groups.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

2.3.1.1. Request

This table shows the URI parameters for the list server groups request:

Name	Type	Description
`{tenant_id}`	String	The ID for the tenant or account in a multi-tenancy cloud.

This operation does not accept a request body.

2.3.1.2. Response

This table shows the body parameters for the list server groups response:

Name	Type	Description
`server_groups`	List *(Required)*	The list of `server_group` objects.
`wrs-sg:project_id`	String *(Required)*	The tenant or project owning the server group.
`policies`	List *(Required)*	A list of policies associated with the server group. TiS added `wrs-sg:affinity-hyperthread` policy to indicate that `only` the members of this server group can share sibling threads with each other.
`metadata`	Dict *(Required)*	Associated metadata key-and-value pairs. TiS added a boolean valued `wrs-sg:best_effort` metadata key-and-value pair to indicate whether the server groups policy should be strictly enforced or not. TiS added an integer valued `wrs-sg:group_size` metadata key-and-value pair to indicate the maximum number of members of the server group.

Example 2.8. List server groups: JSON response

```
{
    "server_groups": [
        {
            "id": "616fb98f-46ca-475e-917e-2563e5a8cd19",
            "wrs-sg:project_id": "28d41dbebab24bdf8854a6632271a3f6"
            "name": "callservergroup",
            "policies": [
```

```
                "wrs-sg:affinity-hyperthread"
            ],
            "members": [],
            "metadata": {
                "wrs-sg:best_effort": "1",
                "wrs-sg:group_size": "2"
            }
        },
        {
            "id": "2fb919a2-4666-11e4-9255-080027367628",
            "wrs-sg:project_id": "28d41dbebab24bdf8854a6632271a3f6"
            "name": "antiaffinitygroup",
            "policies": [
                "anti-affinity"
            ],
            "members": [],
            "metadata": {}
        }
    ]
}
```

This operation does not return a response body.

2.3.2. Create server group

Method	URI	Description
POST	`/v2/{tenant_id}/os-server-groups`	Creates a server group.

Normal response codes: 200

2.3.2.1. Request

This table shows the URI parameters for the create server group request:

Name	Type	Description
`{tenant_id}`	String	The ID for the tenant or account in a multi-tenancy cloud.

Example 2.9. Create server group: JSON request

```
{
    "server_group": {
        "name": "antiaffinitygroup",
        "wrs-sg:project_id": "28d41dbebab24bdf8854a6632271a3f6"
        "policies": [
            "anti-affinity"
        ],
        "metadata": {
            "wrs-sg:best_effort": "1",
            "wrs-sg:group_size": "2"
        }
    }
}
```

This table shows the body parameters for the create server group request:

Name	Type	Description
`wrs-sg:project_id`	UUID *(Required)*	The project or tenant ID which owns this server group.
`policies`	List *(Optional)*	The scheduler policy to associate with the server group. Modified by TiS to include the following additional policy: • `wrs-sg:affinity-hyperthread` which will try to put servers on the same compute node and have servers sharing sibling hyper-thread cores with each other, and only each other. Server Groups using this policy are restriceted to a maximum of 2 members.
`metadata`	Dict *(Optional)*	This parameter specifies a dictionary of optional metadata to be associated with the group. Additional keys added by TiS are: • `wrs-sg:best_effort` (where a value of 0 means that the scheduler policy will be strictly applied and a value of 1 means that the server will still be scheduled even if the policy can't be met). • `wrs-sg:group_size` (where the value is an integer specifying the max number of servers in the group).

This operation does not accept a request body.

2.3.2.2. Response

This table shows the body parameters for the create server group response:

Name	Type	Description
server_group	Dict *(Required)*	The requested `server_group` object.
wrs-sg:project_id	String *(Required)*	The tenant or project owning the server group.
policies	List *(Required)*	A list of policies associated with the server group. TiS added `wrs-sg:affinity-hyperthread` policy to indicate that `only` the members of this server group can share sibling threads with each other.
metadata	Dict *(Required)*	Associated metadata key-and-value pairs. TiS added a boolean valued `wrs-sg:best_effort` metadata key-and-value pair to indicate whether the server groups policy should be strictly enforced or not. TiS added an integer valued `wrs-sg:group_size` metadata key-and-value pair to indicate the maximum number of members of the server group.

Example 2.10. Create server group: JSON response

```
{
    "server_group": {
        "id": "5bbcc3c4-1da2-4437-a48a-66f15b1b13f9",
        "wrs-sg:project_id": "28d41dbebab24bdf8854a6632271a3f6"
        "name": "antiaffinitygroup",
        "policies": [
            "anti-affinity"
        ],
        "members": [],
        "metadata": {
            "wrs-sg:best_effort": "1",
            "wrs-sg:group_size": "2"
        }
    }
}
```

This operation does not return a response body.

2.3.3. Show server group

Method	URI	Description
GET	`/v2/{tenant_id}/os-server-groups/` `{ServerGroup_id}`	Shows details for a specified server group.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

2.3.3.1. Request

This table shows the URI parameters for the show server group request:

Name	Type	Description
`{tenant_id}`	String	The ID for the tenant or account in a multi-tenancy cloud.
`{ServerGroup_id}`	UUID	The server group ID.

This operation does not accept a request body.

2.3.3.2. Response

This table shows the body parameters for the show server group response:

Name	Type	Description
`server_group`	Dict *(Required)*	The requested `server_group` object.
`wrs-sg:project_id`	String *(Required)*	The tenant or project owning the server group.
`policies`	List *(Required)*	A list of policies associated with the server group. TiS added `wrs-sg:affinity-hyperthread` policy to indicate that `only` the members of this server group can share sibling threads with each other.
`metadata`	Dict *(Required)*	Associated metadata key-and-value pairs. TiS added a boolean valued `wrs-sg:best_effort` metadata key-and-value pair to indicate whether the server groups policy should be strictly enforced or not. TiS added an integer valued `wrs-sg:group_size` metadata key-and-value pair to indicate the maximum number of members of the server group.

Example 2.11. Show server group: JSON response

```
{
    "server_group": {
        "id": "616fb98f-46ca-475e-917e-2563e5a8cd19",
        "wrs-sg:project_id": "28d41dbebab24bdf8854a6632271a3f6"
        "name": "callservergroup",
```

```
        "policies": [
            "wrs-sg:affinity-hyperthread"
        ],
        "members": [],
        "metadata": {
            "wrs-sg:best_effort": "1",
            "wrs-sg:group_size": "2"
        }
    }
}
```

This operation does not return a response body.

2.4. Flavor Extra Specs

The extensions on flavor extra specs are:

- `sw:wrs:guest:heartbeat` - Select this option when you expect one or more of the guest applications running on the virtual machine to make use of the HP Helion Open-Stack Carrier Grade Heartbeat client API.

- `sw:wrs:srv_grp_messaging` - Select this option when you expect one or more of the guest applications running on the virtual machine to make use of the HP Helion Open-Stack Carrier Grade Server-Group Messaging client API.

- `hw:wrs:shared_vcpu` - When specified, virtual machines using this flavor will get the requested vCPU scheduled to run on a shared core. This is done even if the hw:cpu_policy is set to dedicated. This allows the guest applications to use dedicated cores exclusively for their high-load tasks, but use a shared core for their low-load (e.g. management type) tasks.

- `hw:wrs:vcpu:scheduler` - Allows you to specify the linux scheduler policy and priority for the vCPU's of the guest. The parameter allows you to specify individual values for each of the vCPUs independently.

- `hw:wrs:min_vcpus` - Allows you to specify the minimum number of vCPUs for the flavor. The value must be between one and the number of VCPUs in the flavor. If this number is specified then the server is assumed to support vCPU scaling.

- `aggregate_instance_extra_specs:localstorage` - Allows you to specify that this virtual machine should be run on a compute node that can provide local storage for the virtual machine.

- `hw:numa_node:{guest-numa-node}` - Allows you to specify the mapping between a virtual machine virtual numa node and the host physical numa node. Not aliased with wrs as this is a candidate for upstreaming.

- `hw:cpu_model` - Allows you to specify the particular cpu model required by the virtual machine; i.e. in order to access the required advanced CPU features such as SSE4.2, AES and/or AVX. Not aliased with wrs as this is a candidate for upstreaming.

Method	URI	Description
GET	/v2/{tenant_id}/fla-vors/{flavor_id}/os-extra_specs	Lists the extra-specs or keys for the specified flavor.

Method	URI	Description
GET	`/v2/{tenant_id}/fla-vors/{flavor_id}/os-extra_specs/{key_id}`	Gets the value of the specified key.
POST	`/v2/{tenant_id}/fla-vors/{flavor_id}/os-extra_specs`	Creates extra-specs or keys for the specified flavor.

2.4.1. List flavor extra specs

Method	URI	Description
GET	`/v2/{tenant_id}/fla-vors/{flavor_id}/os-extra_specs`	Lists the extra-specs or keys for the specified flavor.

Normal response codes: 200

2.4.1.1. Request

This table shows the URI parameters for the list flavor extra specs request:

Name	Type	Description
`{tenant_id}`	String	The ID for the tenant or account in a multi-tenancy cloud.
`{flavor_id}`	String	The ID of the flavor of interest to you.

This operation does not accept a request body.

2.4.1.2. Response

This table shows the body parameters for the list flavor extra specs response:

Name	Type	Description
`extra_specs`	List *(Optional)*	The list of flavor extra specs.
`sw:wrs:guest:heartbeat`	Boolean *(Optional)*	Indicates whether or not the guest applications running in the virtual machine make use of the Heartbeat client API.
`sw:wrs:srv_grp_messaging`	Boolean *(Optional)*	Indicates whether or not the guest applications running in the virtual machine make use of the Server-Group Messaging client API.
`hw:wrs:shared_vcpu`	Integer *(Optional)*	Indicates the vCPU of the guest virtual machine that will be scheduled to run on a shared CPU of the host. Note, this can be specified even if hw:cpu_policy is set to dedicated; allowing the guest application to use dedicated cores exclusively for its high-load tasks, but use a shared core for its low-load (e.g. management type) tasks.

`hw:wrs:vcpu:scheduler`	String
	
	(Optional)
		Indicates the linux scheduler policy and priority for the non-boot vCPUs of the guest. The parameter allows you to specify individual values for each of the vCPUs independently.
		
		This parameter accepts a semicolon-separated list of scheduler:priority:vcpus values as follows:
		
		• scheduler
		
		  The scheduler policy. One of other, fifo, or rr to indicate non real-time, FIFO, and Round Robin policies respectively.
		
		• priority
		
		  The real-time scheduler priority. A value between 1 and 99.
		
		• vcpus
		
		  A list of virtual CPUs as a comma-separated list (1,2,3) or a range specification (1-3). Virtual CPU number 0 refers to the boot virtual CPU and therefore cannot be used.

Name	Type	Description
`hw:wrs:min_vcpus`	Integer (Optional)	Indicates the minimum number of vCPUs for the virtual machine. The value must be between one and the number of VCPUs in the flavor of the virtual machine. If this extra_spec is specified then the server is assumed to support vCPU scaling.
`aggregate_instance_extra_specs:localstorage`	Boolean (Optional)	Indicates whether or not the virtual machine should be run on a compute node that can provide local storage for the virtual machine.
`hw:numa_node:{guest-numa-node}`	Integer (Optional)	Indicates the mapping between a virtual machine virtual numa node and the host physical numa node.
`hw:cpu_model`	String (Optional)	Indicates the particular cpu model required by the virtual machine; i.e. in order to access the required advanced CPU features such as SSE4.2, AES and/or AVX. The valid values are: `default` (QEMU Virtual Processor), `Nehalem` (Intel Core i7 9xx (Nehalem Class Core i7)), `Westmere` (Intel Westmere E56xx/L56xx (Nehalem-C)), `SandyBridge` (Inel Xeon E312xx (Sandy Bridge)), `Haswell` (Intel Core Processor (Haswell)).

Example 2.12. List flavor extra specs: JSON response

```
{
  "extra_specs": {
    "sw:wrs:guest:heartbeat": "True",
    "sw:wrs:srv_grp_messaging": "True",
    "hw:numa_node.0": "1",
    "hw:wrs:vcpu:scheduler": "fifo:50:0"
    "hw:wrs:min_vcpus": "2"
    "hw:wrs:shared_vcpu": "1"
    "hw:cpu_model": "Nehalem",
    "aggregate_instance_extra_specs:localstorage": "False"
  }
}
```

This operation does not return a response body.

2.4.2. Get flavor extra spec details

Method	URI	Description
GET	`/v2/{tenant_id}/fla-vors/{flavor_id}/os-extra_specs/{key_id}`	Gets the value of the specified key.

Normal response codes: 200

2.4.2.1. Request

This table shows the URI parameters for the get flavor extra spec details request:

Name	Type	Description
`{tenant_id}`	String	The ID for the tenant or account in a multi-tenancy cloud.
`{flavor_id}`	String	The ID of the flavor of interest to you.
`{key_id}`	String	The key of the extra-spec of interest to you.

This operation does not accept a request body.

2.4.2.2. Response

This table shows the body parameters for the get flavor extra spec details response:

Name	Type	Description
`sw:wrs:guest:heartbeat`	Boolean (Optional)	Indicates whether or not the guest applications running in the virtual machine make use of the Heartbeat client API.
`sw:wrs:srv_grp_messaging`	Boolean (Optional)	Indicates whether or not the guest applications running in the virtual machine make use of the Server-Group Messaging client API.
`hw:wrs:shared_vcpu`	Integer (Optional)	Indicates the vCPU of the guest virtual machine that will be scheduled to run on a shared CPU of the host. Note, this can be specified even if hw:cpu_policy is set to dedicated; allowing the guest application to use dedicated cores exclusively for its high-load tasks, but use a shared core for its low-load (e.g. management type) tasks.
`hw:wrs:vcpu:scheduler`	String (Optional)	Indicates the linux scheduler policy and priority for the non-boot vCPUs of the guest. The parameter allows you to specify individual values for each of the vCPUs independently. This parameter accepts a semicolon-separated list of scheduler:priority:vcpus values as follows: • scheduler The scheduler policy. One of other, fifo, or rr to indicate non real-time, FIFO, and Round Robin policies respectively. • priority The real-time scheduler priority. A value between 1 and 99. • vcpus A list of virtual CPUs as a comma-separated list (1,2,3) or a range specification (1-3). Virtual CPU number 0 refers to the boot virtual CPU and therefore cannot be used.

Name	Type	Description
`hw:wrs:min_vcpus` *(Optional)*	Integer	Indicates the minimum number of vCPUs for the virtual machine. The value must be between one and the number of VCPUs in the flavor of the virtual machine. If this extra_spec is specified then the server is assumed to support vCPU scaling.
`aggregate_instance_extra_specs:localstorage` *(Optional)*	Boolean	Indicates whether or not the virtual machine should be run on a compute node that can provide local storage for the virtual machine.
`hw:numa_node:{guest-numa-node}` *(Optional)*	Integer	Indicates the mapping between a virtual machine virtual numa node and the host physical numa node.
`hw:cpu_model` *(Optional)*	String	Indicates the particular cpu model required by the virtual machine; i.e. in order to access the required advanced CPU features such as SSE4.2, AES and/or AVX. The valid values are: `default` (QEMU Virtual Processor), `Nehalem` (Intel Core i7 9xx (Nehalem Class Core i7)), `Westmere` (Intel Westmere E56xx/L56xx (Nehalem-C)), `SandyBridge` (Inel Xeon E312xx (Sandy Bridge)), `Haswell` (Intel Core Processor (Haswell).

Example 2.13. Get flavor extra spec details: JSON response

```
{
  "sw:wrs:guest:heartbeat": "True",
}
```

This operation does not return a response body.

2.4.3. Create flavor extra specs

Method	URI	Description
POST	`/v2/{tenant_id}/fla-vors/{flavor_id}/os-extra_specs`	Creates extra-specs or keys for the specified flavor.

Normal response codes: 200

2.4.3.1. Request

This table shows the URI parameters for the create flavor extra specs request:

Name	Type	Description
`{tenant_id}`	String	The ID for the tenant or account in a multi-tenancy cloud.
`{flavor_id}`	String	The ID of the flavor of interest to you.

This table shows the body parameters for the create flavor extra specs request:

Name	Type	Description
`extra_specs`	List *(Optional)*	The list of flavor extra specs.
`sw:wrs:guest:heartbeat`	Boolean *(Optional)*	Indicates whether or not the guest applications running in the virtual machine make use of the Heartbeat client API.
`sw:wrs:srv_grp_messaging`	Boolean *(Optional)*	Indicates whether or not the guest applications running in the virtual machine make use of the Server-Group Messaging client API.
`hw:wrs:shared_vcpu`	Integer *(Optional)*	Indicates the vCPU of the guest virtual machine that will be scheduled to run on a shared CPU of the host. Note, this can be specified even if hw:cpu_policy is set to dedicated; allowing the guest application to use dedicated cores exclusively for its high-load tasks, but use a shared core for its low-load (e.g. management type) tasks.
`hw:wrs:vcpu:scheduler`	String *(Optional)*	Indicates the linux scheduler policy and priority for the non-boot vCPUs of the guest. The parameter allows you to specify individual values for each of the vCPUs independently. This parameter accepts a semicolon-separated list of scheduler:priority:vcpus values as follows: • scheduler The scheduler policy. One of other, fifo, or rr to indicate non real-time, FIFO, and Round Robin policies respectively. • priority The real-time scheduler priority. A value between 1 and 99. • vcpus A list of virtual CPUs as a comma-separated list (1,2,3) or a range specification (1-3). Virtual CPU number 0 refers to the boot virtual CPU and therefore cannot be used.
`hw:wrs:min_vcpus`	Integer *(Optional)*	Indicates the minimum number of vCPUs for the virtual machine. The value must be between one and the number of VCPUs in the flavor of the virtual machine. If this extra_spec is specified then the server is assumed to support vCPU scaling.

Name	Type	Description
aggregate_instance_extra_specs:localstorage *(Optional)*	Boolean	Indicates whether or not the virtual machine should be run on a compute node that can provide local storage for the virtual machine.
hw:numa_node:{guest-nu-ma-node} *(Optional)*	Integer	Indicates the mapping between a virtual machine virtual numa node and the host physical numa node.
hw:cpu_model *(Optional)*	String	Indicates the particular cpu model required by the virtual machine; i.e. in order to access the required advanced CPU features such as SSE4.2, AES and/or AVX. The valid values are: default (QEMU Virtual Processor), Nehalem (Intel Core i7 9xx (Nehalem Class Core i7)), Westmere (Intel Westmere E56xx/L56xx (Nehalem-C)), SandyBridge (Inel Xeon E312xx (Sandy Bridge)), Haswell (Intel Core Processor (Haswell).

Example 2.14. Create flavor extra specs: JSON request

```
{
  "extra_specs": {
    "sw:wrs:guest:heartbeat": "True",
  }
}
```

This operation does not accept a request body.

2.4.3.2. Response

This table shows the body parameters for the create flavor extra specs response:

Name	Type	Description
sw:wrs:guest:heartbeat *(Optional)*	Boolean	Indicates whether or not the guest applications running in the virtual machine make use of the Heartbeat client API.
sw:wrs:srv_grp_messaging *(Optional)*	Boolean	Indicates whether or not the guest applications running in the virtual machine make use of the Server-Group Messaging client API.
hw:wrs:shared_vcpu *(Optional)*	Integer	Indicates the vCPU of the guest virtual machine that will be scheduled to run on a shared CPU of the host. Note, this can be specified even if hw:cpu_policy is set to dedicated; allowing the guest application to use dedicated cores exclusively for its high-load tasks, but use a shared core for its low-load (e.g. management type) tasks.
hw:wrs:vcpu:scheduler *(Optional)*	String	Indicates the linux scheduler policy and priority for the non-boot vCPUs of the guest. The parameter allows you to specify individual values for each of the vCPUs independently. This parameter accepts a semicolon-separated list of scheduler:priority:vcpus values as follows: • scheduler The scheduler policy. One of other, fifo, or rr to indicate non real-time, FIFO, and Round Robin policies respectively. • priority The real-time scheduler priority. A value between 1 and 99. • vcpus A list of virtual CPUs as a comma-separated list (1,2,3) or a range specification (1-3). Virtual CPU number 0 refers to the boot virtual CPU and therefore cannot be used.

Name	Type	Description
hw:wrs:min_vcpus	Integer (Optional)	Indicates the minimum number of vCPUs for the virtual machine. The value must be between one and the number of VCPUs in the flavor of the virtual machine. If this extra_spec is specified then the server is assumed to support vCPU scaling.
aggregate_instance_extra_specs:localstorage	Boolean (Optional)	Indicates whether or not the virtual machine should be run on a compute node that can provide local storage for the virtual machine.
hw:numa_node:{guest-numa-node}	Integer (Optional)	Indicates the mapping between a virtual machine virtual numa node and the host physical numa node.
hw:cpu_model	String (Optional)	Indicates the particular cpu model required by the virtual machine; i.e. in order to access the required advanced CPU features such as SSE4.2, AES and/or AVX. The valid values are: default (QEMU Virtual Processor), Nehalem (Intel Core i7 9xx (Nehalem Class Core i7)), Westmere (Intel Westmere E56xx/L56xx (Nehalem-C)), SandyBridge (Inel Xeon E312xx (Sandy Bridge)), Haswell (Intel Core Processor (Haswell).

Example 2.15. Create flavor extra specs: JSON response

```
{
  "extra_specs": {
    "sw:wrs:guest:heartbeat": "True",
  }
}
```

This operation does not return a response body.

3. Networking API v2 extensions

This section describes changes made to the standard OpenStack Networking API for HP Helion OpenStack Carrier Grade. Some existing OpenStack API instances have been enhanced to add new attributes, or update the semantics of existing attributes. In other cases, entirely new API instances have been created to expose HP Helion OpenStack Carrier Grade networking functionality via the standard RESTful API.

The typical port used for the Networking REST API is 9696. However, proper technique would be to look up the neutron service endpoint in keystone.

Method	URI	Description
	Extensions	
GET	`/v2.0/extensions`	Lists all extensions.
GET	`/v2.0/extensions/{extension_alias}`	Gets information about a specified extension.
	Provider Network	
GET	`/v2.0/wrs-provider/providernets`	Lists all provider networks.
GET	`/v2.0/wrs-provider/provider-nets/{providernet_id}`	Shows detailed information about a specific provider network.
POST	`/v2.0/wrs-provider/providernets`	Creates a provider network.
PUT	`/v2.0/wrs-provider/provider-nets/{providernet_id}`	Modifies a specific provider network.
DELETE	`/v2.0/wrs-provider/provider-nets/{providernet_id}`	Deletes a specific provider network.
GET	`/v2.0/wrs-provider/provider-nets/{providernet_id}/provider-net-bindings`	Lists networks that are implemented by a given provider network. Each network is listed with its assigned provider network segmentation identifier. If the network has any tagged subnets then they will be listed as separate entities with their corresponding provider network segmentation identifier.
	Provider Network Range	
GET	`/v2.0/wrs-provider/provider-net-ranges`	Lists all provider network ranges.
GET	`/v2.0/wrs-provider/provider-net-ranges/{providernet-range_id}`	Shows detailed information about a specific provider network range.
POST	`/v2.0/wrs-provider/provider-net-ranges`	Creates a provider network range.
PUT	`/v2.0/wrs-provider/provider-net-ranges/{providernet-range_id}`	Modifies a specific provider network range.
DELETE	`/v2.0/wrs-provider/provider-net-ranges/{providernet-range_id}`	Deletes a specific provider network range.
	Provider Network Type	
GET	`/v2.0/wrs-provider/provider-net-types`	Lists all supported providernet types.
	Tenant Settings	
GET	`/v2.0/wrs-tenant/settings`	Lists all tenant network settings.
GET	`/v2.0/wrs-tenant/set-tings/{tenant_id}`	Shows detailed information about a specific tenant network setting.
PUT	`/v2.0/wrs-tenant/set-tings/{tenant_id}`	Modifies a specific tenant network setting.
DELETE	`/v2.0/wrs-tenant/set-tings/{tenant_id}`	Deletes a specific tenant network setting.

Method	URI	Description
		QOS Policies
GET	`/v2.0/wrs-tm/qoses`	Lists all QOS policies.
GET	`/v2.0/wrs-tm/qoses/{qos_id}`	Shows detailed information about a specific QOS policy.
POST	`/v2.0/wrs-tm/qoses`	Creates a QOS policy.
PUT	`/v2.0/wrs-tm/qoses/{qos_id}`	Modifies a specific QOS policy.
DELETE	`/v2.0/wrs-tm/qoses/{qos_id}`	Deletes a specific QOS policy.
		Network
GET	`/v2.0/networks`	Lists networks that are accessible to the tenant who submits the reequest.
GET	`/v2.0/networks/{network_id}`	Shows information for a specified network.
		Subnet
GET	`/v2.0/subnets`	Lists subnets that are accessible to the tenant who submits the request.
GET	`/v2.0/subnets/{subnet_id}`	Shows information for a specified subnet.
POST	`/v2.0/subnets`	Creates a subnet on a specified network.
		Port
GET	`/v2.0/ports`	Lists ports to which the tenant has access.
GET	`/v2.0/ports/{port_id}`	Shows information for a specified port.
		Router
GET	`/v2.0/routers`	Lists logical routers that are accessible to the tenant who submits the request.
GET	`/v2.0/routers/{router_id}`	Shows details for a specified router.

3.1. Extensions

The Extensions entity lists all available extensions; both open-source extensions and (wrs-) extensions.

Method	URI	Description
GET	`/v2.0/extensions`	Lists all extensions.
GET	`/v2.0/extensions/{extension_alias}`	Gets information about a specified extension.

3.1.1. List extensions

Method	URI	Description
GET	`/v2.0/extensions`	Lists all extensions.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

3.1.1.1. Request

This operation does not accept a request body.

3.1.1.2. Response

This table shows the body parameters for the list extensions response:

Name	Type	Description
`namespace`	String (Optional)	Indicates namespace of the extension.
`name`	String (Optional)	Indicates name of the extension.
`updated`	String (Optional)	Indicates updated time of the extension.
`description`	String (Optional)	Indicates description of the extension.
`alias`	String (Optional)	Indicates alias of the extension.
`links`	List (Optional)	A list of links for the extension.

Example 3.1. List extensions: JSON response

```
{
  "extensions" : [
    ...
    {
      "namespace" : "http://docs.windriver.org/tis/ext/wrs-provider/v1",
      "name" : "wrs-provider-network",
      "updated" : "2014-10-01T12:00:00-00:00",
      "description" : "WRS Provider Network Extensions.",
      "alias" : "wrs-provider",
      "links" : []
    },
    {
      "namespace" : "http://docs.windriver.org/tis/ext/wrs-tenant/v1",
```

```
            "name" : "wrs-tenant-settings",
            "updated" : "2014-10-01T12:00:00-00:00",
            "description" : "WRS Tenant Network Settings Extensions.",
            "alias" : "wrs-tenant",
            "links" : []
        },
        {
            "namespace" : "http://docs.windriver.org/tis/ext/wrs-tm/v1",
            "name" : "wrs-traffic-management",
            "updated" : "2014-10-01T12:00:00-00:00",
            "description" : "WRS Traffic Management Extensions.",
            "alias" : "wrs-tm",
            "links" : []
        },
        {
            "namespace" : "http://docs.windriver.org/tis/ext/wrs-net/v1",
            "name" : "wrs-tenant-network",
            "updated" : "2014-10-01T12:00:00-00:00",
            "description" : "WRS Tenant Network Extensions.",
            "alias" : "wrs-net",
            "links" : []
        },
        {
            "namespace" : "http://docs.windriver.org/tis/ext/wrs-binding/v1",
            "name" : "wrs-port-binding",
            "updated" : "2014-10-01T12:00:00-00:00",
            "description" : "WRS Port Binding Extensions.",
            "alias" : "wrs-binding",
            "links" : []
        },
        ...
    ]
}
```

This operation does not return a response body.

3.1.2. Get extension

Method	URI	Description
GET	`/v2.0/extensions/{extension_alias}`	Gets information about a specified extension.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

3.1.2.1. Request

This table shows the URI parameters for the get extension request:

Name	Type	Description
`{extension_alias}`	String	The alias for the extension to list.

This operation does not accept a request body.

3.1.2.2. Response

This table shows the body parameters for the get extension response:

Name	Type	Description
`namespace`	String (Optional)	Indicates namespace of the extension.
`name`	String (Optional)	Indicates name of the extension.
`updated`	String (Optional)	Indicates updated time of the extension.
`description`	String (Optional)	Indicates description of the extension.
`alias`	String (Optional)	Indicates alias of the extension.
`links`	List (Optional)	A list of links for the extension.

Example 3.2. Get extension: JSON response

```
{
    "extensions" : {
        "namespace" : "http://docs.windriver.org/tis/ext/wrs-provider/v1",
        "name" : "wrs-provider-network",
        "updated" : "2014-10-01T12:00:00-00:00",
        "description" : "WRS Provider Network Extensions.",
        "alias" : "wrs-provider",
        "links" : []
    }
}
```

```
OR

{
   "extensions" : {
       "namespace" : "http://docs.windriver.org/tis/ext/wrs-tenant/v1",
       "name" : "wrs-tenant-settings",
       "updated" : "2014-10-01T12:00:00-00:00",
       "description" : "WRS Tenant Network Settings Extensions.",
       "alias" : "wrs-tenant",
       "links" : []
   }
}

OR

{
   "extensions" : {
       "namespace" : "http://docs.windriver.org/tis/ext/wrs-tm/v1",
       "name" : "wrs-traffic-management",
       "updated" : "2014-10-01T12:00:00-00:00",
       "description" : "WRS Traffic Management Extensions.",
       "alias" : "wrs-tm",
       "links" : []
   }
}

OR

{
   "extensions" : {
       "namespace" : "http://docs.windriver.org/tis/ext/wrs-net/v1",
       "name" : "wrs-tenant-network",
       "updated" : "2014-10-01T12:00:00-00:00",
       "description" : "WRS Tenant Network Extensions.",
       "alias" : "wrs-net",
       "links" : []
   }
}

OR

{
   "extensions" : {
       "namespace" : "http://docs.windriver.org/tis/ext/wrs-binding/v1",
       "name" : "wrs-port-binding",
       "updated" : "2014-10-01T12:00:00-00:00",
       "description" : "WRS Port Binding Extensions.",
       "alias" : "wrs-binding",
       "links" : []
   }
}
```

This operation does not return a response body.

3.2. Provider Network

The Provider Network entity is a new entity which was added to the OpenStack API. It enables management of provider networks via the RESTful API. The standard OpenStack API

included no such entity; instead, the end user was required to edit static configuration files through the system to add, or update provider network information.

This entity and all of its operations are only available to administrator level users.

Method	URI	Description
GET	`/v2.0/wrs-provider/providernets`	Lists all provider networks.
GET	`/v2.0/wrs-provider/provider-nets/{providernet_id}`	Shows detailed information about a specific provider network.
POST	`/v2.0/wrs-provider/providernets`	Creates a provider network.
PUT	`/v2.0/wrs-provider/provider-nets/{providernet_id}`	Modifies a specific provider network.
DELETE	`/v2.0/wrs-provider/provider-nets/{providernet_id}`	Deletes a specific provider network.
GET	`/v2.0/wrs-provider/provider-nets/{providernet_id}/provider-net-bindings`	Lists networks that are implemented by a given provider network. Each network is listed with its assigned provider network segmentation identifier. If the network has any tagged subnets then they will be listed as separate entities with their corresponding provider network segmentation identifier.

3.2.1. List provider networks

Method	URI	Description
GET	`/v2.0/wrs-provider/providernets`	Lists all provider networks.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

3.2.1.1. Request

This operation does not accept a request body.

3.2.1.2. Response

This table shows the body parameters for the list provider networks response:

Name	Type	Description
`providernets`	List *(Optional)*	The list of provider networks.
`description`	String *(Optional)*	User defined description of the provider network.
`id`	UUID *(Optional)*	The unique UUID value of the provider network.
`mtu`	Integer *(Optional)*	The maximum transmit unit (MTU) assigned to the provider network. Must be between 576 and 9216 bytes inclusively. The default value is 1500.
`name`	String *(Optional)*	The user defined name of the provider network.
`ranges`	List *(Optional)*	The list of segmentation ranges defined for this provider network. See the provider network range description for a description of range fields.
`status`	String *(Optional)*	The current status of the provider network. Returns `ACTIVE` if at least one compute node has a data interface associated to this provider network and is available.

Example 3.3. List provider networks: JSON response

```
{
    "providernets": [
        {
            "description": "Group0 provider networks for data1 interfaces",
            "id": "b67d40aa-3651-4dd6-886f-4bff5caa266e",
            "mtu": 1500,
            "name": "group0-data1",
            "ranges": [
                {
                    "description": "tenant2 reserved networks",
                    "id": "a8184bf7-b683-4bc1-a70c-dae34391344c",
```

```
                    "maximum": 631,
                    "minimum": 616,
                    "name": "group0-tenant2",
                    "shared": false,
                    "tenant_id": "d8753af85cef49a4bf5f95208c4957f3"
                }
            ],
            "status": "ACTIVE",
            "type": "vlan"
        },
        {
            "description": "Group0 provider networks for data0 interfaces",
            "id": "c496c429-cb52-4d4b-9171-b4b31fa91a80",
            "mtu": 1500,
            "name": "group0-data0",
            "ranges": [
                {
                    "description": "Shared internal networks",
                    "id": "f3e1bc29-29f7-4ee0-a78d-9c3d7a0f53e5",
                    "maximum": 731,
                    "minimum": 700,
                    "name": "group0-shared",
                    "shared": true,
                    "tenant_id": null
                },
                {
                    "description": "External network access",
                    "id": "35ef3460-700a-48a3-8df9-145eb68fcd31",
                    "maximum": 10,
                    "minimum": 10,
                    "name": "group0-external",
                    "shared": true,
                    "tenant_id": null
                },
                {
                    "description": "tenant1 reserved networks",
                    "id": "736b0c0d-945b-4a17-9fe4-cf02a5327132",
                    "maximum": 615,
                    "minimum": 600,
                    "name": "group0-tenant1",
                    "shared": false,
                    "tenant_id": "0590d9fa3dd74bfe9bdf7ed4e5331a10"
                }
            ],
            "status": "ACTIVE",
            "type": "vlan"
        }
    ]
}
```

This operation does not return a response body.

3.2.2. Show provider network

Method	URI	Description
GET	`/v2.0/wrs-provider/provider-nets/{providernet_id}`	Shows detailed information about a specific provider network.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404)

3.2.2.1. Request

This table shows the URI parameters for the show provider network request:

Name	Type	Description
`{providernet_id}`	String	The ID for a provider network.

This operation does not accept a request body.

3.2.2.2. Response

This table shows the body parameters for the show provider network response:

Name	Type	Description
`description`	String *(Optional)*	User defined description of the provider network.
`id`	UUID *(Optional)*	The unique UUID value of the provider network.
`mtu`	Integer *(Optional)*	The maximum transmit unit (MTU) assigned to the provider network. Must be between 576 and 9216 bytes inclusively. The default value is 1500.
`name`	String *(Optional)*	The user defined name of the provider network.
`ranges`	List *(Optional)*	The list of segmentation ranges defined for this provider network. See the provider network range description for a description of range fields.
`status`	String *(Optional)*	The current status of the provider network. Returns `ACTIVE` if at least one compute node has a data interface associated to this provider network and is available.

Example 3.4. Show provider network: JSON response

```
{
    "providernet": {
        "description": "Group0 provider networks for data1 interfaces",
        "id": "b67d40aa-3651-4dd6-886f-4bff5caa266e",
        "mtu": 1500,
        "name": "group0-data1",
        "ranges": [
```

```
        {
            "description": "tenant2 reserved networks",
            "id": "a8184bf7-b683-4bc1-a70c-dae34391344c",
            "maximum": 631,
            "minimum": 616,
            "name": "group0-tenant2",
            "shared": false,
            "tenant_id": "d8753af85cef49a4bf5f95208c4957f3"
        }
    ],
    "status": "ACTIVE",
    "type": "vlan"
    }
}
```

This operation does not return a response body.

3.2.3. Create provider network

Method	URI	Description
POST	`/v2.0/wrs-provider/providernets`	Creates a provider network.

Normal response codes: 200

Error response codes: badMediaType (415), NetworkNotFound (400)

3.2.3.1. Request

This table shows the body parameters for the create provider network request:

Name	Type	Description
`description`	String *(Optional)*	User defined description of the provider network.
`name`	String *(Optional)*	The user defined name of the provider network.
`type`	String *(Optional)*	The encapsulation type of the provider network. Valid values are: `vlan`, `flat`

Example 3.5. Create provider network: JSON request

```
{
    "providernet": {
        "description": "A sample provider network",
        "name": "test",
        "type": "vlan"
    }
}
```

This operation does not accept a request body.

3.2.3.2. Response

This table shows the body parameters for the create provider network response:

Name	Type	Description
`description`	String *(Optional)*	User defined description of the provider network.
`id`	UUID *(Optional)*	The unique UUID value of the provider network.
`mtu`	Integer *(Optional)*	The maximum transmit unit (MTU) assigned to the provider network. Must be between 576 and 9216 bytes inclusively. The default value is 1500.
`name`	String *(Optional)*	The user defined name of the provider network.
`ranges`	List *(Optional)*	The list of segmentation ranges defined for this provider network. See the provider network range description for a description of range fields.

Name	Type	Description
status	String *(Optional)*	The current status of the provider network. Returns ACTIVE if at least one compute node has a data interface associated to this provider network and is available.

Example 3.6. Create provider network: JSON response

```
{
    "providernet": {
        "description": "A sample provider network",
        "id": "4da9e42c-e556-470c-8e92-cbd19bcc6a10",
        "mtu": 1500,
        "name": "test",
        "ranges": [],
        "status": "DOWN",
        "type": "vlan"
    }
}
```

This operation does not return a response body.

3.2.4. Modify provider network

Method	URI	Description
PUT	`/v2.0/wrs-provider/provider-nets/{providernet_id}`	Modifies a specific provider network.

Normal response codes: 200

Error response codes: badMediaType (415), NetworkNotFound (400)

3.2.4.1. Request

This table shows the URI parameters for the modify provider network request:

Name	Type	Description
`{providernet_id}`	String	The ID for a provider network.

This table shows the body parameters for the modify provider network request:

Name	Type	Description
`description`	String *(Optional)*	User defined description of the provider network.

Example 3.7. Modify provider network: JSON request

```
{
    "providernet": {
        "description": "Another sample provider network"
    }
}
```

This operation does not accept a request body.

3.2.4.2. Response

This table shows the body parameters for the modify provider network response:

Name	Type	Description
`description`	String *(Optional)*	User defined description of the provider network.
`id`	UUID *(Optional)*	The unique UUID value of the provider network.
`mtu`	Integer *(Optional)*	The maximum transmit unit (MTU) assigned to the provider network. Must be between 576 and 9216 bytes inclusively. The default value is 1500.
`name`	String *(Optional)*	The user defined name of the provider network.
`ranges`	List *(Optional)*	The list of segmentation ranges defined for this provider network. See the provider network range description for a description of range fields.

Name	Type	Description
status	String *(Optional)*	The current status of the provider network. Returns ACTIVE if at least one compute node has a data interface associated to this provider network and is available.

Example 3.8. Modify provider network: JSON response

```
{
    "providernet": {
        "description": "Another sample provider network",
        "id": "4da9e42c-e556-470c-8e92-cbd19bcc6a10",
        "mtu": 1500,
        "name": "test",
        "ranges": [],
        "status": "DOWN",
        "type": "vlan"
    }
}
```

This operation does not return a response body.

3.2.5. Delete provider network

Method	URI	Description
DELETE	`/v2.0/wrs-provider/provider-nets/{providernet_id}`	Deletes a specific provider network.

Normal response codes: 204

3.2.5.1. Request

This table shows the URI parameters for the delete provider network request:

Name	Type	Description
`{providernet_id}`	String	The ID for a provider network.

This operation does not accept a request body.

3.2.5.2. Response

This operation does not return a response body.

3.2.6. List networks implemented by a provider network

Method	URI	Description
GET	`/v2.0/wrs-provider/provider-nets/{providernet_id}/provider-net-bindings`	Lists networks that are implemented by a given provider network. Each network is listed with its assigned provider network segmentation identifier. If the network has any tagged subnets then they will be listed as separate entities with their corresponding provider network segmentation identifier.

Normal response codes: 200

Error response codes: itemNotFound (401)

3.2.6.1. Request

This table shows the URI parameters for the list networks implemented by a provider network request:

Name	Type	Description
`{providernet_id}`	String	The ID for a provider network.

This operation does not accept a request body.

3.2.6.2. Response

This table shows the body parameters for the list networks implemented by a provider network response:

Name	Type	Description
`networks`	List *(Optional)*	The list of tenant networks.
`id`	UUID *(Optional)*	The unique UUID value of the tenant network.
`name`	String *(Optional)*	The user defined name of the tenant network.
`providernet_type`	String *(Optional)*	The encapsulation type of the provider network.
`segmentation_id`	Integer *(Optional)*	The provider network segmentation identifier that is assigned to this tenant network. If the `vlan_id` attribute is non-zero then the `segmentation_id` represents that identifier which has been associated to a tagged subnet on the listed tenant network.
`vlan_id`	Integer *(Optional)*	The VLAN identifier which has been configured on the tenant subnet.

Example 3.9. List networks implemented by a provider network: JSON response

```
{
    "networks": [
```

```
    {
        "id": "2c0896cf-d118-4dca-9760-b4d97e3c7ec3",
        "name": "tenant1-net0",
        "providernet_type": "vlan",
        "segmentation_id": 601,
        "vlan_id": 0
    },
    {
        "id": "7e5ed852-a990-4fc5-89a2-b17093ca1982",
        "name": "internal0-net0",
        "providernet_type": "vlan",
        "segmentation_id": 700,
        "vlan_id": 0
    },
    {
        "id": "b9475152-11d3-4bda-95c7-fb26a3ad3876",
        "name": "external-net0",
        "providernet_type": "vlan",
        "segmentation_id": 10,
        "vlan_id": 0
    },
    {
        "id": "f652780a-7a9d-4667-8df4-5c8632728be9",
        "name": "tenant1-mgmt-net",
        "providernet_type": "vlan",
        "segmentation_id": 600,
        "vlan_id": 0
    }
    ]
}
```

This operation does not return a response body.

3.3. Provider Network Range

The Provider Network Range entity is a new entity which was added to the OpenStack API. It enables management of provider network segmentation ranges via the RESTful API. The standard OpenStack API included no such entity; instead, the end user was required to edit static configuration files through the system to add, or update provider network segmentation ranges.

This entity and all of its operations are only available to administrator level users.

Method	URI	Description
GET	`/v2.0/wrs-provider/provider-net-ranges`	Lists all provider network ranges.
GET	`/v2.0/wrs-provider/provider-net-ranges/{providernet-range_id}`	Shows detailed information about a specific provider network range.
POST	`/v2.0/wrs-provider/provider-net-ranges`	Creates a provider network range.
PUT	`/v2.0/wrs-provider/provider-net-ranges/{providernet-range_id}`	Modifies a specific provider network range.
DELETE	`/v2.0/wrs-provider/provider-net-ranges/{providernet-range_id}`	Deletes a specific provider network range.

3.3.1. List provider network ranges

Method	URI	Description
GET	`/v2.0/wrs-provider/provider-net-ranges`	Lists all provider network ranges.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

3.3.1.1. Request

This operation does not accept a request body.

3.3.1.2. Response

This table shows the body parameters for the list provider network ranges response:

Name	Type	Description
`providernetranges`	List (Optional)	The list of provider network ranges.
`description`	String (Optional)	User defined description of the provider network segmentation range.
`id`	UUID (Optional)	The unique UUID value of the provider network segmentation range.
`maximum`	Integer (Optional)	The upper bound of the segmentation range (inclusive).
`minimum`	Integer (Optional)	The lower bound of the segmentation range (inclusive).
`name`	String (Optional)	The user defined name of the provider network segmentation range.
`providernet_id`	UUID (Optional)	The unique UUID of the parent provider network.
`providernet_name`	String (Optional)	The user defined name of the parent provider network.
`shared`	Bool (Optional)	The shared attribute indicates that the range is available to any tenant.
`tenant_id`	UUID (Optional)	The unique UUID of the tenant which owns the range. Only valid if the shared attribute is False.

Example 3.10. List provider network ranges: JSON response

```
{
```

```
    "providernet_ranges": [
        {
            "description": "Shared internal networks",
            "id": "f3e1bc29-29f7-4ee0-a78d-9c3d7a0f53e5",
            "maximum": 731,
            "minimum": 700,
            "name": "group0-shared",
            "providernet_id": "c496c429-cb52-4d4b-9171-b4b31fa91a80",
            "providernet_name": "group0-data0",
            "shared": true,
            "tenant_id": null
        },
        {
            "description": "External network access",
            "id": "35ef3460-700a-48a3-8df9-145eb68fcd31",
            "maximum": 10,
            "minimum": 10,
            "name": "group0-external",
            "providernet_id": "c496c429-cb52-4d4b-9171-b4b31fa91a80",
            "providernet_name": "group0-data0",
            "shared": true,
            "tenant_id": null
        },
        {
            "description": "tenant1 reserved networks",
            "id": "736b0c0d-945b-4a17-9fe4-cf02a5327132",
            "maximum": 615,
            "minimum": 600,
            "name": "group0-tenant1",
            "providernet_id": "c496c429-cb52-4d4b-9171-b4b31fa91a80",
            "providernet_name": "group0-data0",
            "shared": false,
            "tenant_id": "0590d9fa3dd74bfe9bdf7ed4e5331a10"
        },
        {
            "description": "tenant2 reserved networks",
            "id": "a8184bf7-b683-4bc1-a70c-dae34391344c",
            "maximum": 631,
            "minimum": 616,
            "name": "group0-tenant2",
            "providernet_id": "b67d40aa-3651-4dd6-886f-4bff5caa266e",
            "providernet_name": "group0-data1",
            "shared": false,
            "tenant_id": "d8753af85cef49a4bf5f95208c4957f3"
        },
        {
            "description": "A sample provider network segmentation range",
            "id": "bdf07406-a867-42e5-9533-5100c4a3f2ba",
            "maximum": 100,
            "minimum": 1,
            "name": "test-range-0",
            "providernet_id": "239ffb19-bad8-4b05-9194-aa8399816a36",
            "providernet_name": "test",
            "shared": true,
            "tenant_id": null
        }
    ]
}
```

This operation does not return a response body.

3.3.2. Show provider network range

Method	URI	Description
GET	`/v2.0/wrs-provider/provider-net-ranges/{providernet-range_id}`	Shows detailed information about a specific provider network range.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404)

3.3.2.1. Request

This table shows the URI parameters for the show provider network range request:

Name	Type	Description
`{providernetrange_id}`	String	The ID for a provider network segmentation range.

This operation does not accept a request body.

3.3.2.2. Response

This table shows the body parameters for the show provider network range response:

Name	Type	Description
`description`	String *(Optional)*	User defined description of the provider network segmentation range.
`id`	UUID *(Optional)*	The unique UUID value of the provider network segmentation range.
`maximum`	Integer *(Optional)*	The upper bound of the segmentation range (inclusive).
`minimum`	Integer *(Optional)*	The lower bound of the segmentation range (inclusive).
`name`	String *(Optional)*	The user defined name of the provider network segmentation range.
`providernet_id`	UUID *(Optional)*	The unique UUID of the parent provider network.
`providernet_name`	String *(Optional)*	The user defined name of the parent provider network.
`shared`	Bool *(Optional)*	The shared attribute indicates that the range is available to any tenant.
`tenant_id`	UUID *(Optional)*	The unique UUID of the tenant which owns the range. Only valid if the shared attribute is False.

Example 3.11. Show provider network range: JSON response

```
{
    "providernet_range": {
        "description": "A sample provider network segmentation range",
        "id": "bdf07406-a867-42e5-9533-5100c4a3f2ba",
        "maximum": 100,
        "minimum": 1,
        "name": "test-range-0",
        "providernet_id": "239ffb19-bad8-4b05-9194-aa8399816a36",
        "providernet_name": "test",
        "shared": true,
        "tenant_id": null
    }
}
```

This operation does not return a response body.

3.3.3. Create provider network range

Method	URI	Description
POST	`/v2.0/wrs-provider/provider-net-ranges`	Creates a provider network range.

Normal response codes: 200

Error response codes: badMediaType (415), NetworkNotFound (400)

3.3.3.1. Request

This table shows the body parameters for the create provider network range request:

Name	Type	Description
`description`	String *(Optional)*	User defined description of the provider network segmentation range.
`maximum`	Integer *(Optional)*	The upper bound of the segmentation range (inclusive).
`minimum`	Integer *(Optional)*	The lower bound of the segmentation range (inclusive).
`name`	String *(Optional)*	The user defined name of the provider network segmentation range.

Example 3.12. Create provider network range: JSON request

```
{
    "providernet_range": {
        "description": "A sample provider network segmentation range",
        "maximum": "100",
        "minimum": "1",
        "name": "test-range-0",
        "providernet_id": "239ffb19-bad8-4b05-9194-aa8399816a36",
        "shared": true
    }
}
```

This operation does not accept a request body.

3.3.3.2. Response

This table shows the body parameters for the create provider network range response:

Name	Type	Description
`description`	String *(Optional)*	User defined description of the provider network segmentation range.
`id`	UUID *(Optional)*	The unique UUID value of the provider network segmentation range.
`maximum`	Integer	The upper bound of the segmentation range (inclusive).

Name	Type	Description
	(Optional)	
minimum	Integer *(Optional)*	The lower bound of the segmentation range (inclusive).
name	String *(Optional)*	The user defined name of the provider network segmentation range.
providernet_id	UUID *(Optional)*	The unique UUID of the parent provider network.
providernet_name	String *(Optional)*	The user defined name of the parent provider network.
shared	Bool *(Optional)*	The shared attribute indicates that the range is available to any tenant.
tenant_id	UUID *(Optional)*	The unique UUID of the tenant which owns the range. Only valid if the shared attribute is False.

Example 3.13. Create provider network range: JSON response

```
{
    "providernet_range": {
        "description": "A sample provider network segmentation range",
        "id": "bdf07406-a867-42e5-9533-5100c4a3f2ba",
        "maximum": "100",
        "minimum": "1",
        "name": "test-range-0",
        "providernet_id": "239ffb19-bad8-4b05-9194-aa8399816a36",
        "providernet_name": "test",
        "shared": true,
        "tenant_id": null
    }
}
```

This operation does not return a response body.

3.3.4. Modify provider network range

Method	URI	Description
PUT	`/v2.0/wrs-provider/provider-net-ranges/{providernet-range_id}`	Modifies a specific provider network range.

Normal response codes: 200

Error response codes: badMediaType (415), NetworkNotFound (400)

3.3.4.1. Request

This table shows the URI parameters for the modify provider network range request:

Name	Type	Description
`{providernetrange_id}`	String	The ID for a provider network segmentation range.

This table shows the body parameters for the modify provider network range request:

Name	Type	Description
`description`	String *(Optional)*	User defined description of the provider network segmentation range.
`maximum`	Integer *(Optional)*	The upper bound of the segmentation range (inclusive).
`minimum`	Integer *(Optional)*	The lower bound of the segmentation range (inclusive).

Example 3.14. Modify provider network range: JSON request

```
{
    "providernet_range": {
        "maximum": "1099",
        "minimum": "1000",
        "description": "VLAN identifiers reserved for tenant1"
    }
}
```

This operation does not accept a request body.

3.3.4.2. Response

This table shows the body parameters for the modify provider network range response:

Name	Type	Description
`description`	String *(Optional)*	User defined description of the provider network segmentation range.
`id`	UUID *(Optional)*	The unique UUID value of the provider network segmentation range.
`maximum`	Integer	The upper bound of the segmentation range (inclusive).

Name	Type	Description
	(Optional)	
minimum	Integer *(Optional)*	The lower bound of the segmentation range (inclusive).
name	String *(Optional)*	The user defined name of the provider network segmentation range.
providernet_id	UUID *(Optional)*	The unique UUID of the parent provider network.
providernet_name	String *(Optional)*	The user defined name of the parent provider network.
shared	Bool *(Optional)*	The shared attribute indicates that the range is available to any tenant.
tenant_id	UUID *(Optional)*	The unique UUID of the tenant which owns the range. Only valid if the shared attribute is False.

Example 3.15. Modify provider network range: JSON response

```
{
    "providernet_range": {
        "description": null,
        "id": "fe24481a-303f-4cd9-a0ac-76c2e4a9bcc8",
        "maximum": "1099",
        "minimum": "1000",
        "name": "test-range-0",
        "providernet_id": "c496c429-cb52-4d4b-9171-b4b31fa91a80",
        "providernet_name": "group0-data0",
        "shared": false,
        "tenant_id": "206f147dcf72421fa6829e33bfb34637"
    }
}
```

This operation does not return a response body.

3.3.5. Delete provider network range

Method	URI	Description
DELETE	`/v2.0/wrs-provider/provider-net-ranges/{providernet-range_id}`	Deletes a specific provider network range.

Normal response codes: 204

3.3.5.1. Request

This table shows the URI parameters for the delete provider network range request:

Name	Type	Description
`{providernetrange_id}`	String	The ID for a provider network segmentation range.

This operation does not accept a request body.

3.3.5.2. Response

This operation does not return a response body.

3.4. Provider Network Type

The Provider Network Type entity is a new entity which was added to the OpenStack API. It exists simply to allow the end user to query which provider network types are supported by the system.

This entity and all of its operations are only available to administrator level users.

Method	URI	Description
GET	`/v2.0/wrs-provider/provider-net-types`	Lists all supported providernet types.

3.4.1. List provider network types

Method	URI	Description
GET	`/v2.0/wrs-provider/provider-net-types`	Lists all supported providernet types.

Insert extra description here, if required.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

3.4.1.1. Request

This operation does not accept a request body.

3.4.1.2. Response

This table shows the body parameters for the list provider network types response:

Name	Type	Description
`providernettypes`	List *(Optional)*	The list of supported providernet types.
`description`	String *(Optional)*	System description of the provider network type.
`type`	String *(Optional)*	The encapsulation type of the provider network. Valid values are: `vlan, flat`

Example 3.16. List provider network types: JSON response

```
{
    "providernet_types": [
        {
            "description": "Ethernet network without additional
 encapsulation",
            "type": "flat"
        },
        {
            "description": "802.1q encapsulated Ethernet network",
            "type": "vlan"
        }
    ]
}
```

This operation does not return a response body.

3.5. Tenant Settings

The Tenant Settings entity is a new entity which was added to the OpenStack API. It enables management of features or system behaviours on a per-tenant basis by the administrator.

This entity and all of its operations are only available to administrator level users.

Method	URI	Description
GET	`/v2.0/wrs-tenant/settings`	Lists all tenant network settings.
GET	`/v2.0/wrs-tenant/set-tings/{tenant_id}`	Shows detailed information about a specific tenant network setting.
PUT	`/v2.0/wrs-tenant/set-tings/{tenant_id}`	Modifies a specific tenant network setting.
DELETE	`/v2.0/wrs-tenant/set-tings/{tenant_id}`	Deletes a specific tenant network setting.

3.5.1. List settings

Method	URI	Description
GET	`/v2.0/wrs-tenant/settings`	Lists all tenant network settings.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

3.5.1.1. Request

This operation does not accept a request body.

3.5.1.2. Response

This table shows the body parameters for the list settings response:

Name	Type	Description
`settings`	List *(Optional)*	The list of tenant network settings.
`mac_filtering`	Bool *(Optional)*	The state of the source MAC filtering feature for the specified tenant. The current state of the feature only affects newly launched VM instances.
`tenant_id`	UUID *(Optional)*	The unique UUID of the tenant.

Example 3.17. List settings: JSON response

```
{
    "settings": [
        {
            "mac_filtering": false,
            "tenant_id": "0590d9fa3dd74bfe9bdf7ed4e5331a10"
        },
        {
            "mac_filtering": false,
            "tenant_id": "d8753af85cef49a4bf5f95208c4957f3"
        }
    ]
}
```

This operation does not return a response body.

3.5.2. Show setting

Method	URI	Description
GET	`/v2.0/wrs-tenant/set-tings/{tenant_id}`	Shows detailed information about a specific tenant net-work setting.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

3.5.2.1. Request

This table shows the URI parameters for the show setting request:

Name	Type	Description
`{tenant_id}`	String	The ID for a tenant.

This operation does not accept a request body.

3.5.2.2. Response

This table shows the body parameters for the show setting response:

Name	Type	Description
`mac_filtering`	Bool *(Optional)*	The state of the source MAC filtering feature for the specified tenant. The current state of the feature only affects newly launched VM instances.
`tenant_id`	UUID *(Optional)*	The unique UUID of the tenant.

Example 3.18. Show setting: JSON response

```
{
    "setting": {
        "mac_filtering": false
    }
}
```

This operation does not return a response body.

3.5.3. Modify setting

Method	URI	Description
PUT	`/v2.0/wrs-tenant/set-tings/{tenant_id}`	Modifies a specific tenant network setting.

Normal response codes: 200

Error response codes: badMediaType (415), NetworkNotFound (400)

3.5.3.1. Request

This table shows the URI parameters for the modify setting request:

Name	Type	Description
`{tenant_id}`	String	The ID for a tenant.

This table shows the body parameters for the modify setting request:

Name	Type	Description
`mac_filtering`	Bool *(Optional)*	The state of the source MAC filtering feature for the specified tenant. The current state of the feature only affects newly launched VM instances.

Example 3.19. Modify setting: JSON request

```
{
    "setting": {
        "mac_filtering": true
    }
}
```

This operation does not accept a request body.

3.5.3.2. Response

This table shows the body parameters for the modify setting response:

Name	Type	Description
`mac_filtering`	Bool *(Optional)*	The state of the source MAC filtering feature for the specified tenant. The current state of the feature only affects newly launched VM instances.
`tenant_id`	UUID *(Optional)*	The unique UUID of the tenant.

Example 3.20. Modify setting: JSON response

```
{
    "setting": {
        "mac_filtering": true
    }
}
```

This operation does not return a response body.

3.5.4. Delete setting

Method	URI	Description
DELETE	`/v2.0/wrs-tenant/set-tings/{tenant_id}`	Deletes a specific tenant network setting.

Normal response codes: 204

3.5.4.1. Request

This table shows the URI parameters for the delete setting request:

Name	Type	Description
`{tenant_id}`	String	The ID for a tenant.

This operation does not accept a request body.

3.5.4.2. Response

This operation does not return a response body.

3.6. QOS Policies

The QOS entity is a new entity which was added to the OpenStack API. It enables management of Quality of Service policies and profiles via the RESTful API. QOS policies can be created and maintained by the administrator.

Method	URI	Description
GET	`/v2.0/wrs-tm/qoses`	Lists all QOS policies.
GET	`/v2.0/wrs-tm/qoses/{qos_id}`	Shows detailed information about a specific QOS policy.
POST	`/v2.0/wrs-tm/qoses`	Creates a QOS policy.
PUT	`/v2.0/wrs-tm/qoses/{qos_id}`	Modifies a specific QOS policy.
DELETE	`/v2.0/wrs-tm/qoses/{qos_id}`	Deletes a specific QOS policy.

3.6.1. List QOS policies

Method	URI	Description
GET	`/v2.0/wrs-tm/qoses`	Lists all QOS policies.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

3.6.1.1. Request

This operation does not accept a request body.

3.6.1.2. Response

This table shows the body parameters for the list qos policies response:

Name	Type	Description
`Qoses`	List (Optional)	The list of QOS policies.
`description`	String (Optional)	The user defined description of the QoS policy.
`id`	UUID (Optional)	The unique UUID value of the QoS policy.
`name`	String (Optional)	The user defined name of the QoS policy.
`policies`	Dict (Optional)	The set of scheduler policies and weights for the QoS policy.
`tenant_id`	UUID (Optional)	The unique UUID of the tenant to which this policy is assigned.

Example 3.21. List QOS policies: JSON response

```
{
    "qoses": [
        {
            "description": "tenant1 Management Network Policy",
            "id": "102c64e4-ad26-4610-ae39-f59e15fcb80c",
            "name": "tenant1-mgmt-qos",
            "policies": {
                "scheduler": {
                    "weight": "8"
                }
            },
            "tenant_id": "0590d9fa3dd74bfe9bdf7ed4e5331a10"
        },
        {
```

```
                "description": "tenant2 Management Network Policy",
                "id": "62970a9a-b093-4747-92dd-9de25616036a",
                "name": "tenant2-mgmt-qos",
                "policies": {
                    "scheduler": {
                        "weight": "8"
                    }
                },
                "tenant_id": "d8753af85cef49a4bf5f95208c4957f3"
        },
        {
                "description": "External Network Policy",
                "id": "d28e697c-c290-4895-b57c-ac7d38db9003",
                "name": "external-qos",
                "policies": {
                    "scheduler": {
                        "weight": "16"
                    }
                },
                "tenant_id": "206f147dcf72421fa6829e33bfb34637"
        }
    ]
}
```

This operation does not return a response body.

3.6.2. Show QOS policy

Method	URI	Description
GET	`/v2.0/wrs-tm/qoses/{qos_id}`	Shows detailed information about a specific QOS policy.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

3.6.2.1. Request

This table shows the URI parameters for the show qos policy request:

Name	Type	Description
`{qos_id}`	String	The ID for a QOS Policy.

This operation does not accept a request body.

3.6.2.2. Response

This table shows the body parameters for the show qos policy response:

Name	Type	Description
`description` *(Optional)*	String	The user defined description of the QoS policy.
`id` *(Optional)*	UUID	The unique UUID value of the QoS policy.
`name` *(Optional)*	String	The user defined name of the QoS policy.
`policies` *(Optional)*	Dict	The set of scheduler policies and weights for the QoS policy.
`tenant_id` *(Optional)*	UUID	The unique UUID of the tenant to which this policy is assigned.

Example 3.22. Show QOS policy: JSON response

```
{
    "qos": {
        "description": "tenant1 Management Network Policy",
        "id": "102c64e4-ad26-4610-ae39-f59e15fcb80c",
        "name": "tenant1-mgmt-qos",
        "policies": {
            "scheduler": {
                "weight": "8"
            }
        },
        "tenant_id": "0590d9fa3dd74bfe9bdf7ed4e5331a10"
```

```
    }
}
```

This operation does not return a response body.

3.6.3. Create QOS policy

Method	URI	Description
POST	`/v2.0/wrs-tm/qoses`	Creates a QOS policy.

Normal response codes: 200

Error response codes: badMediaType (415), NetworkNotFound (400)

3.6.3.1. Request

This table shows the body parameters for the create qos policy request:

Name	Type	Description
`description`	String *(Optional)*	The user defined description of the QoS policy.
`name`	String *(Optional)*	The user defined name of the QoS policy.
`policies`	Dict *(Optional)*	The set of scheduler policies and weights for the QoS policy.
`tenant_id`	UUID *(Optional)*	The unique UUID of the tenant to which this policy is assigned.

Example 3.23. Create QOS policy: JSON request

```
{
    "qos": {
        "description": "A sample QoS profile",
        "name": "test-qos-0",
        "policies": {
            "scheduler": {
                "weight": "8"
            }
        },
        "tenant_id": "0590d9fa3dd74bfe9bdf7ed4e5331a10"
    }
}
```

This operation does not accept a request body.

3.6.3.2. Response

This table shows the body parameters for the create qos policy response:

Name	Type	Description
`description`	String *(Optional)*	The user defined description of the QoS policy.
`id`	UUID *(Optional)*	The unique UUID value of the QoS policy.

Name	Type	Description
name	String *(Optional)*	The user defined name of the QoS policy.
policies	Dict *(Optional)*	The set of scheduler policies and weights for the QoS policy.
tenant_id	UUID *(Optional)*	The unique UUID of the tenant to which this policy is assigned.

Example 3.24. Create QOS policy: JSON response

```
{
    "qos": {
        "description": "A sample QoS profile",
        "id": "5e1841fa-c106-47ee-a736-e527478f1239",
        "name": "test-qos-0",
        "policies": {
            "scheduler": {
                "weight": "8"
            }
        },
        "tenant_id": "0590d9fa3dd74bfe9bdf7ed4e5331a10"
    }
}
```

This operation does not return a response body.

3.6.4. Modify QOS policy

Method	URI	Description
PUT	`/v2.0/wrs-tm/qoses/{qos_id}`	Modifies a specific QOS policy.

Normal response codes: 200

Error response codes: badMediaType (415), NetworkNotFound (400)

3.6.4.1. Request

This table shows the URI parameters for the modify qos policy request:

Name	Type	Description
`{qos_id}`	String	The ID for a QOS Policy.

This table shows the body parameters for the modify qos policy request:

Name	Type	Description
`description`	String *(Optional)*	The user defined description of the QoS policy.
`policies`	Dict *(Optional)*	The set of scheduler policies and weights for the QoS policy.

Example 3.25. Modify QOS policy: JSON request

```
{
    "qos": {
        "description": "Another sample QoS profile",
        "policies": {
            "scheduler": {
                "weight": "16"
            }
        }
    }
}
```

This operation does not accept a request body.

3.6.4.2. Response

This table shows the body parameters for the modify qos policy response:

Name	Type	Description
`description`	String *(Optional)*	The user defined description of the QoS policy.
`id`	UUID *(Optional)*	The unique UUID value of the QoS policy.
`name`	String *(Optional)*	The user defined name of the QoS policy.

Name	Type	Description
policies	Dict *(Optional)*	The set of scheduler policies and weights for the QoS policy.
tenant_id	UUID *(Optional)*	The unique UUID of the tenant to which this policy is assigned.

Example 3.26. Modify QOS policy: JSON response

```
{
    "qos": {
        "description": "Another sample QoS profile",
        "id": "5e1841fa-c106-47ee-a736-e527478f1239",
        "name": "test-qos-0",
        "policies": {
            "scheduler": {
                "weight": "16"
            }
        },
        "tenant_id": "0590d9fa3dd74bfe9bdf7ed4e5331a10"
    }
}
```

This operation does not return a response body.

3.6.5. Delete QOS policy

Method	URI	Description
DELETE	`/v2.0/wrs-tm/qoses/{qos_id}`	Deletes a specific QOS policy.

Normal response codes: 204

3.6.5.1. Request

This table shows the URI parameters for the delete qos policy request:

Name	Type	Description
`{qos_id}`	String	The ID for a QOS Policy.

This operation does not accept a request body.

3.6.5.2. Response

This operation does not return a response body.

3.7. Network

The Network entity is an existing OpenStack API. It has been extended to add the following HP Helion OpenStack Carrier Grade functionality.

• A QOS policy can optionally be associated to a tenant network

• The maximum transmit unit (MTU) of each tenant network is inherited from its associated provider network

• The status of each tenant network is derived from the state of the DHCP server which services its subnets

Method	URI	Description
GET	`/v2.0/networks`	Lists networks that are accessible to the tenant who submits the reequest.
GET	`/v2.0/networks/{network_id}`	Shows information for a specified network.

3.7.1. List networks

Method	URI	Description
GET	`/v2.0/networks`	Lists networks that are accessible to the tenant who submits the reequest.

This is an existing OpenStack API. The documentation that follows lists only the fields that are new or modified. For a detailed description of existing and unmodified fields please refer to the standard OpenStack API documentation.

Normal response codes: 200

Error response codes: itemNotFound (401)

3.7.1.1. Request

This operation does not accept a request body.

3.7.1.2. Response

This table shows the body parameters for the list networks response:

Name	Type	Description
`networks`	List *(Optional)*	The list of tenant networks.
`wrs-provider:mtu`	Integer *(Optional)*	The maximum transmit unit (MTU). This value is inherited from the assigned provider network.
`wrs-tm:qos`	UUID *(Optional)*	The unique UUID of the assigned QoS policy.
`status`	String *(Optional)*	Indicates whether the tenant network is `ACTIVE` or `DOWN`. If the network is DHCP enabled then it can only be active if at least 1 DHCP agent is servicing the network. TiS corrected the reporting of this status.

Example 3.27. List networks: JSON response

```
{
    "networks": [
        {
            "admin_state_up": true,
            "id": "b9475152-11d3-4bda-95c7-fb26a3ad3876",
            "name": "external-net0",
            "wrs-provider:mtu": 1500,
            "provider:network_type": "vlan",
            "provider:physical_network": "group0-data0",
            "provider:segmentation_id": 10,
            "wrs-tm:qos": "d28e697c-c290-4895-b57c-ac7d38db9003",
            "router:external": true,
            "shared": true,
            "status": "ACTIVE",
            "subnets": [
                "b282ef86-2584-4a02-9b58-69d6233952a2"
            ],
            "tenant_id": "206f147dcf72421fa6829e33bfb34637"
```

```
        },
        {
            "admin_state_up": true,
            "id": "f652780a-7a9d-4667-8df4-5c8632728be9",
            "name": "tenant1-mgmt-net",
            "wrs-provider:mtu": 1500,
            "provider:network_type": "vlan",
            "provider:physical_network": "group0-data0",
            "provider:segmentation_id": 600,
            "wrs-tm:qos": "102c64e4-ad26-4610-ae39-f59e15fcb80c",
            "router:external": false,
            "shared": false,
            "status": "ACTIVE",
            "subnets": [
                "34efd537-7a72-4fcd-b837-9874caf34117"
            ],
            "tenant_id": "0590d9fa3dd74bfe9bdf7ed4e5331a10"
        },
        {

            "admin_state_up": true,
            "id": "9472a8ab-9205-43ef-a460-5f01f031791a",
            "name": "tenant2-mgmt-net",
            "wrs-provider:mtu": 1500,
            "provider:network_type": "vlan",
            "provider:physical_network": "group0-data1",
            "provider:segmentation_id": 616,
            "wrs-tm:qos": "62970a9a-b093-4747-92dd-9de25616036a",
            "router:external": false,
            "shared": false,
            "status": "ACTIVE",
            "subnets": [
                "9aa900f4-522b-4b83-ba93-57f7d92da5d5"
            ],
            "tenant_id": "d8753af85cef49a4bf5f95208c4957f3"
        },
        {

            "admin_state_up": true,
            "id": "7e5ed852-a990-4fc5-89a2-b17093ca1982",
            "name": "internal0-net0",
            "wrs-provider:mtu": 1500,
            "provider:network_type": "vlan",
            "provider:physical_network": "group0-data0",
            "provider:segmentation_id": 700,
            "router:external": false,
            "shared": true,
            "status": "ACTIVE",
            "subnets": [
                "ad791a3e-33cf-4d8d-b80f-91c87f97745e"
            ],
            "tenant_id": "206f147dcf72421fa6829e33bfb34637"
        },
        {
            "admin_state_up": true,
            "id": "2c0896cf-d118-4dca-9760-b4d97e3c7ec3",
            "name": "tenant1-net0",
            "wrs-provider:mtu": 1500,
            "provider:network_type": "vlan",
            "provider:physical_network": "group0-data0",
            "provider:segmentation_id": 601,
            "router:external": false,
```

```
                "shared": false,
                "status": "ACTIVE",
                "subnets": [
                    "bc269028-1862-4dde-ba2e-62a67d1af4e4",
                    "837aebc9-6c78-43e9-8124-168ba16adbc7"
                ],
                "tenant_id": "0590d9fa3dd74bfe9bdf7ed4e5331a10"
            },
            {
                "admin_state_up": true,
                "id": "7a77e654-794a-4e19-9679-ac2733e19876",
                "name": "tenant2-net0",
                "wrs-provider:mtu": 1500,
                "provider:network_type": "vlan",
                "provider:physical_network": "group0-data1",
                "provider:segmentation_id": 617,
                "router:external": false,
                "shared": false,
                "status": "ACTIVE",
                "subnets": [
                    "985806f5-9fd7-4d47-9da6-cb0c1316e63d"
                ],
                "tenant_id": "d8753af85cef49a4bf5f95208c4957f3"
            }
        ]
}
```

This operation does not return a response body.

3.7.2. Show network

Method	URI	Description
GET	`/v2.0/networks/{network_id}`	Shows information for a specified network.

This is an existing OpenStack API. The documentation that follows lists only the fields that are new or modified. For a detailed description of existing and unmodified fields please refer to the standard OpenStack API documentation.

Normal response codes: 200

Error response codes: itemNotFound (401), unauthorized (404)

3.7.2.1. Request

This table shows the URI parameters for the show network request:

Name	Type	Description
`{network_id}`	UUID	The UUID for a network.

This operation does not accept a request body.

3.7.2.2. Response

This table shows the body parameters for the show network response:

Name	Type	Description
`wrs-provider:mtu`	Integer *(Optional)*	The maximum transmit unit (MTU). This value is inherited from the assigned provider network.
`wrs-tm:qos`	UUID *(Optional)*	The unique UUID of the assigned QoS policy.
`status`	String *(Optional)*	Indicates whether the tenant network is `ACTIVE` or `DOWN`. If the network is DHCP enabled then it can only be active if at least 1 DHCP agent is servicing the network. TiS corrected the reporting of this status.

Example 3.28. Show network: JSON response

```
{
    "network": {
        "admin_state_up": true,
        "id": "e87e7438-8a07-4e82-a472-862bb7fa93ac",
        "name": "test-net-0",
        "wrs-provider:mtu": 1500,
        "provider:network_type": "vlan",
        "provider:physical_network": "group0-data0",
        "provider:segmentation_id": 602,
        "wrs-tm:qos": "102c64e4-ad26-4610-ae39-f59e15fcb80c",
        "router:external": false,
        "shared": false,
        "status": "ACTIVE",
```

```
        "subnets": [],
        "tenant_id": "0590d9fa3dd74bfe9bdf7ed4e5331a10"
    }
}
```

This operation does not return a response body.

3.8. Subnet

The Subnet entity is an existing OpenStack API. It has been extended to add the following HP Helion OpenStack Carrier Grade functionality.

• A subnet can be configured to allow VLAN tagging by the VM instance.

Method	URI	Description
GET	`/v2.0/subnets`	Lists subnets that are accessible to the tenant who submits the request.
GET	`/v2.0/subnets/{subnet_id}`	Shows information for a specified subnet.
POST	`/v2.0/subnets`	Creates a subnet on a specified network.

3.8.1. List subnets

Method	URI	Description
GET	`/v2.0/subnets`	Lists subnets that are accessible to the tenant who submits the request.

This is an existing OpenStack API. The documentation that follows lists only the fields that are new or modified. For a detailed description of existing and unmodified fields please refer to the standard OpenStack API documentation.

Normal response codes: 200

Error response codes: itemNotFound (401)

3.8.1.1. Request

This operation does not accept a request body.

3.8.1.2. Response

This table shows the body parameters for the list subnets response:

Name	Type	Description
`subnets`	List *(Optional)*	The list of subnets.
`wrs-net:managed`	Bool *(Optional)*	Indicates whether IP address allocation is managed by the system or by the customer. If `true` then the system allocates IP addresses when ports are created and attached to VM instances. If `false` then the system will not assign any IP addresses automatically. This implies that if the system cannot allocate any IP addresses that it also cannot allocate a DHCP server, manage allocation pools, or server DNS nameservers or static routers.
`wrs-provider:mtu`	Integer *(Optional)*	The maximum transmit unit (MTU). This value is inherited from the assigned provider network.
`wrs-provider:network_type`	String *(Optional)*	The type of the provider network to which this subnet is assigned. **Only visible to admin users.**
`wrs-provider:physical_name`	String *(Optional)*	The name of the provider network to which this subnet is assigned. **Only visible to admin users.**
`wrs-provider:segmentation_id`	String *(Optional)*	The provider network segmentation id to which this subnet is assigned. **Only visible to admin users.**
`wrs-net:vlan_id`	Integer *(Optional)*	The VLAN ID to be used in the VM instance. If the VLAN ID is 0 then all packets originated from the VM instance are expected to be untagged. If the VLAN ID value is non zero than it is expected that all packets originated by the VM must be tagged with the corresponding VLAN ID value. Any other value will be discarded by the host vswitch.

Example 3.29. List subnets: JSON response

```
{
    "subnets": [
        {
            "allocation_pools": [
                {
                    "end": "192.168.1.254",
                    "start": "192.168.1.2"
                }
            ],
            "cidr": "192.168.1.0/24",
            "dns_nameservers": [],
            "enable_dhcp": false,
            "gateway_ip": "192.168.1.1",
            "host_routes": [],
            "id": "b282ef86-2584-4a02-9b58-69d6233952a2",
            "ip_version": 4,
            "wrs-net:managed": true,
            "name": "external-subnet0",
            "network_id": "b9475152-11d3-4bda-95c7-fb26a3ad3876",
            "wrs-provider:mtu": 1500,
            "wrs-provider:network_type": "vlan",
            "wrs-provider:physical_network": "group0-data0",
            "wrs-provider:segmentation_id": 10,
            "tenant_id": "206f147dcf72421fa6829e33bfb34637",
            "wrs-net:vlan_id": 0
        },
        {
            "allocation_pools": [
                {
                    "end": "192.168.201.50",
                    "start": "192.168.201.2"
                }
            ],
            "cidr": "192.168.201.0/24",
            "dns_nameservers": [],
            "enable_dhcp": true,
            "gateway_ip": "192.168.201.1",
            "host_routes": [],
            "id": "9aa900f4-522b-4b83-ba93-57f7d92da5d5",
            "ip_version": 4,
            "wrs-net:managed": true,
            "name": "tenant2-mgmt-subnet",
            "network_id": "9472a8ab-9205-43ef-a460-5f01f031791a",
            "wrs-provider:mtu": 1500,
            "wrs-provider:network_type": "vlan",
            "wrs-provider:physical_network": "group0-data1",
            "wrs-provider:segmentation_id": 616,
            "tenant_id": "d8753af85cef49a4bf5f95208c4957f3",
            "wrs-net:vlan_id": 0
        },
        {
            "allocation_pools": [
                {
                    "end": "192.168.101.50",
                    "start": "192.168.101.2"
                }
            ],
            "cidr": "192.168.101.0/24",
```

```json
            "dns_nameservers": [],
            "enable_dhcp": true,
            "gateway_ip": "192.168.101.1",
            "host_routes": [],
            "id": "34efd537-7a72-4fcd-b837-9874caf34117",
            "ip_version": 4,
            "wrs-net:managed": true,
            "name": "tenant1-mgmt-subnet",
            "network_id": "f652780a-7a9d-4667-8df4-5c8632728be9",
            "wrs-provider:mtu": 1500,
            "wrs-provider:network_type": "vlan",
            "wrs-provider:physical_network": "group0-data0",
            "wrs-provider:segmentation_id": 600,
            "tenant_id": "0590d9fa3dd74bfe9bdf7ed4e5331a10",
            "wrs-net:vlan_id": 0
        },
        {
            "allocation_pools": [],
            "cidr": "10.0.0.0/24",
            "dns_nameservers": [],
            "enable_dhcp": false,
            "gateway_ip": null,
            "host_routes": [],
            "id": "ad791a3e-33cf-4d8d-b80f-91c87f97745e",
            "ip_version": 4,
            "wrs-net:managed": false,
            "name": "internal0-subnet0-0",
            "network_id": "7e5ed852-a990-4fc5-89a2-b17093ca1982",
            "wrs-provider:mtu": 1500,
            "wrs-provider:network_type": "vlan",
            "wrs-provider:physical_network": "group0-data0",
            "wrs-provider:segmentation_id": 700,
            "tenant_id": "206f147dcf72421fa6829e33bfb34637",
            "wrs-net:vlan_id": 0
        },
        {
            "allocation_pools": [],
            "cidr": "172.16.0.0/24",
            "dns_nameservers": [],
            "enable_dhcp": false,
            "gateway_ip": null,
            "host_routes": [],
            "id": "bc269028-1862-4dde-ba2e-62a67d1af4e4",
            "ip_version": 4,
            "wrs-net:managed": false,
            "name": "tenant1-subnet0",
            "network_id": "2c0896cf-d118-4dca-9760-b4d97e3c7ec3",
            "wrs-provider:mtu": 1500,
            "wrs-provider:network_type": "vlan",
            "wrs-provider:physical_network": "group0-data0",
            "wrs-provider:segmentation_id": 601,
            "tenant_id": "0590d9fa3dd74bfe9bdf7ed4e5331a10",
            "wrs-net:vlan_id": 0
        },
        {
            "allocation_pools": [],
            "cidr": "172.18.0.0/24",
            "dns_nameservers": [],
            "enable_dhcp": false,
            "gateway_ip": null,
```

```
                "host_routes": [],
                "id": "985806f5-9fd7-4d47-9da6-cb0c1316e63d",
                "ip_version": 4,
                "wrs-net:managed": false,
                "name": "tenant2-subnet0",
                "network_id": "7a77e654-794a-4e19-9679-ac2733e19876",
                "wrs-provider:mtu": 1500,
                "wrs-provider:network_type": "vlan",
                "wrs-provider:physical_network": "group0-data1",
                "wrs-provider:segmentation_id": 617,
                "tenant_id": "d8753af85cef49a4bf5f95208c4957f3",
                "wrs-net:vlan_id": 0
            }
        ]
    }
```

This operation does not return a response body.

3.8.2. Show subnet

Method	URI	Description
GET	`/v2.0/subnets/{subnet_id}`	Shows information for a specified subnet.

This is an existing OpenStack API. The documentation that follows lists only the fields that are new or modified. For a detailed description of existing and unmodified fields please refer to the standard OpenStack API documentation.

Normal response codes: 201

Error response codes: itemNotFound (401), unauthorized (404)

3.8.2.1. Request

This table shows the URI parameters for the show subnet request:

Name	Type	Description
`{subnet_id}`	UUID	The UUID for a subnet.

This operation does not accept a request body.

3.8.2.2. Response

This table shows the body parameters for the show subnet response:

Name	Type	Description
`wrs-net:managed`	Bool *(Optional)*	Indicates whether IP address allocation is managed by the system or by the customer. If `true` then the system allocates IP addresses when ports are created and attached to VM instances. If `false` then the system will not assign any IP addresses automatically. This implies that if the system cannot allocate any IP addresses that it also cannot allocate a DHCP server, manage allocation pools, or server DNS nameservers or static routers.
`wrs-provider:mtu`	Integer *(Optional)*	The maximum transmit unit (MTU). This value is inherited from the assigned provider network.
`wrs-provider:network_type`	String *(Optional)*	The type of the provider network to which this subnet is assigned. **Only visible to admin users.**
`wrs-provider:physical_name`	String *(Optional)*	The name of the provider network to which this subnet is assigned. **Only visible to admin users.**
`wrs-provider:segmentation_id`	String *(Optional)*	The provider network segmentation id to which this subnet is assigned. **Only visible to admin users.**
`wrs-net:vlan_id`	Integer *(Optional)*	The VLAN ID to be used in the VM instance. If the VLAN ID is 0 then all packets originated from the VM instance are expected to be untagged. If the VLAN ID value is non zero than it is expected that all packets originated by the VM must be tagged with the corresponding VLAN ID value. Any other value will be discarded by the host vswitch.

Example 3.30. Show subnet: JSON response

```
{
    "subnet": {
        "allocation_pools": [],
        "cidr": "1.2.3.0/24",
        "dns_nameservers": [],
        "enable_dhcp": false,
        "gateway_ip": null,
        "host_routes": [],
        "id": "837aebc9-6c78-43e9-8124-168ba16adbc7",
        "ip_version": 4,
        "wrs-net:managed": false,
        "name": "test-subnet-0",
        "network_id": "2c0896cf-d118-4dca-9760-b4d97e3c7ec3",
        "wrs-provider:mtu": 1500,
        "wrs-provider:network_type": "vlan",
        "wrs-provider:physical_network": "group0-data0",
        "wrs-provider:segmentation_id": 615,
        "tenant_id": "206f147dcf72421fa6829e33bfb34637",
        "wrs-net:vlan_id": 99
    }
}
```

This operation does not return a response body.

3.8.3. Create subnet

Method	URI	Description
POST	`/v2.0/subnets`	Creates a subnet on a specified network.

This is an existing OpenStack API. The documentation that follows lists only the fields that are new or modified. For a detailed description of existing and unmodified fields please refer to the standard OpenStack API documentation.

Normal response codes: 201

Error response codes: badRequest (400), itemNotFound (401), forbidden (403), unauthorized (404), buildInProgress (409)

3.8.3.1. Request

This table shows the body parameters for the create subnet request:

Name	Type	Description
`wrs-net:managed`	Bool *(Optional)*	Indicates whether IP address allocation is managed by the system or by the customer. If `true` then the system allocates IP addresses when ports are created and attached to VM instances. If `false` then the system will not assign any IP addresses automatically. This implies that if the system cannot allocate any IP addresses that it also cannot allocate a DHCP server, manage allocation pools, or server DNS nameservers or static routers.
`wrs-provider:segmentation_id`	String *(Optional)*	The provider network segmentation id to which this subnet is assigned. The provider:physical_network attribute is not accepted because the system enforces that all subnets be implemented by the same provider network. Therefore, it is assumed that the provider:physical_network attribute is the same as the parent tenant network. **Only available to admin users.**
`wrs-net:vlan_id`	Integer *(Optional)*	The VLAN ID to be used in the VM instance. If the VLAN ID is 0 then all packets originated from the VM instance are expected to be untagged. If the VLAN ID value is non zero than it is expected that all packets originated by the VM must be tagged with the corresponding VLAN ID value. Any other value will be discarded by the host vswitch.

Example 3.31. Create subnet: JSON request

```
{
    "subnet": {
        "cidr": "1.2.3.0/24",
        "enable_dhcp": false,
        "ip_version": 4,
        "wrs-net:managed": false,
        "name": "test-subnet-0",
        "network_id": "2c0896cf-d118-4dca-9760-b4d97e3c7ec3",
        "wrs-provider:segmentation_id": "615",
        "wrs-net:vlan_id": 99
    }
```

```
}
```

This operation does not accept a request body.

3.8.3.2. Response

This table shows the body parameters for the create subnet response:

Name	Type	Description
`wrs-net:managed`	Bool *(Optional)*	Indicates whether IP address allocation is managed by the system or by the customer. If `true` then the system allocates IP addresses when ports are created and attached to VM instances. If `false` then the system will not assign any IP addresses automatically. This implies that if the system cannot allocate any IP addresses that it also cannot allocate a DHCP server, manage allocation pools, or server DNS nameservers or static routers.
`wrs-provider:mtu`	Integer *(Optional)*	The maximum transmit unit (MTU). This value is inherited from the assigned provider network.
`wrs-provider:network_type`	String *(Optional)*	The type of the provider network to which this subnet is assigned. **Only visible to admin users.**
`wrs-provider:physical_name`	String *(Optional)*	The name of the provider network to which this subnet is assigned. **Only visible to admin users.**
`wrs-provider:segmentation_id`	String *(Optional)*	The provider network segmentation id to which this subnet is assigned. **Only visible to admin users.**
`wrs-net:vlan_id`	Integer *(Optional)*	The VLAN ID to be used in the VM instance. If the VLAN ID is 0 then all packets originated from the VM instance are expected to be untagged. If the VLAN ID value is non zero than it is expected that all packets originated by the VM must be tagged with the corresponding VLAN ID value. Any other value will be discarded by the host vswitch.

Example 3.32. Create subnet: JSON response

```
{
    "subnet": {
        "allocation_pools": [],
        "cidr": "1.2.3.0/24",
        "dns_nameservers": [],
        "enable_dhcp": false,
        "gateway_ip": null,
        "host_routes": [],
        "id": "837aebc9-6c78-43e9-8124-168ba16adbc7",
        "ip_version": 4,
        "wrs-net:managed": false,
        "name": "test-subnet-0",
        "network_id": "2c0896cf-d118-4dca-9760-b4d97e3c7ec3",
        "wrs-provider:mtu": 1500,
        "wrs-provider:network_type": "vlan",
        "wrs-provider:physical_network": "group0-data0",
        "wrs-provider:segmentation_id": 615,
        "tenant_id": "206f147dcf72421fa6829e33bfb34637",
        "wrs-net:vlan_id": 99
    }
```

```
}
```

This operation does not return a response body.

3.9. Port

The Port entity is an existing OpenStack API. It has been extended to add the following HP Helion OpenStack Carrier Grade functionality.

- The network interface type (vif_model) is recorded when attached to a VM instance.

- Source MAC address filtering is enabled when created for a tenant which has this feature enabled by the administrator.

- The MAC address automatically updates to reflect changes to PCI passthrough devices for VM instances.

- The maximum transmit unit (MTU) attribute is a reflection of the MTU value of the attached tenant network.

Method	URI	Description
GET	`/v2.0/ports`	Lists ports to which the tenant has access.
GET	`/v2.0/ports/{port_id}`	Shows information for a specified port.

3.9.1. List ports

Method	URI	Description
GET	`/v2.0/ports`	Lists ports to which the tenant has access.

This is an existing OpenStack API. The documentation that follows lists only the fields that are new or modified. For a detailed description of existing and unmodified fields please refer to the standard OpenStack API documentation.

Normal response codes: 200

Error response codes: itemNotFound (401)

3.9.1.1. Request

This operation does not accept a request body.

3.9.1.2. Response

This table shows the body parameters for the list ports response:

Name	Type	Description
`ports`	List *(Optional)*	The list of ports.
`wrs-binding:mac_filtering`	Bool *(Optional)*	The state of source MAC address filtering on the port. If this is `true` then the attached vswitch enforces that all ingress packets have a source MAC address that matches the port MAC address.
`wrs-binding:mtu`	Integer *(Optional)*	The maximum transmit unit (MTU). This value is inherited from the tenant network that attaches to this port.
`wrs-binding:vif_model`	String *(Optional)*	The type of virtual networking device that is presented to the VM instance. This value is only visible if the device_owner is a VM instance port (i.e., device_owner="compute:nova").

Example 3.33. List ports: JSON response

```
{
    "ports": [
        {
            "admin_state_up": true,
            "binding:capabilities": {
                "port_filter": true
            },
            "binding:host_id": "compute-0",
            "wrs-binding:mtu": 1500,
            "binding:vif_type": "bridge",
            "device_id": "ce58c529-ba73-47f7-a409-a438ceced112",
            "device_owner": "network:router_interface",
            "fixed_ips": [
                {
                    "ip_address": "192.168.201.1",
                    "subnet_id": "9aa900f4-522b-4b83-ba93-57f7d92da5d5"
                }
            ],
            "id": "e44d4c03-0add-4b1a-bd36-8f6e7ccd48d9",
            "mac_address": "fa:16:3e:a6:27:3f",
```

```
            "wrs-binding:mac_filtering": false
            "name": "",
            "network_id": "9472a8ab-9205-43ef-a460-5f01f031791a",
            "security_groups": [],
            "status": "ACTIVE",
            "tenant_id": "d8753af85cef49a4bf5f95208c4957f3"
        },
        {
            "admin_state_up": true,
            "binding:capabilities": {
                "port_filter": true
            },
            "binding:host_id": "compute-0",
            "wrs-binding:mtu": 1500,
            "binding:vif_type": "bridge",
            "device_id": "ea2baef7-d84b-44c4-82e7-f274ab5e8b6f",
            "device_owner": "network:router_gateway",
            "fixed_ips": [
                {
                    "ip_address": "192.168.1.2",
                    "subnet_id": "b282ef86-2584-4a02-9b58-69d6233952a2"
                }
            ],
            "id": "0ebefe26-de93-4d9e-b31a-b4620a6743e8",
            "mac_address": "fa:16:3e:29:11:9b",
            "wrs-binding:mac_filtering": false
            "name": "",
            "network_id": "b9475152-11d3-4bda-95c7-fb26a3ad3876",
            "security_groups": [],
            "status": "ACTIVE",
            "tenant_id": ""
        },
        {
            "admin_state_up": true,
            "binding:capabilities": {
                "port_filter": true
            },
            "binding:host_id": "compute-0",
            "wrs-binding:mtu": 1500,
            "binding:vif_type": "bridge",
            "device_id": "ea2baef7-d84b-44c4-82e7-f274ab5e8b6f",
            "device_owner": "network:router_interface",
            "fixed_ips": [
                {
                    "ip_address": "192.168.101.1",
                    "subnet_id": "34efd537-7a72-4fcd-b837-9874caf34117"
                }
            ],
            "id": "cf8c4f0a-f615-4437-a87a-39f2b87b7662",
            "mac_address": "fa:16:3e:fb:4b:89",
            "wrs-binding:mac_filtering": false
            "name": "",
            "network_id": "f652780a-7a9d-4667-8df4-5c8632728be9",
            "security_groups": [],
            "status": "ACTIVE",
            "tenant_id": "0590d9fa3dd74bfe9bdf7ed4e5331a10"
        },
        {
            "admin_state_up": true,
            "binding:capabilities": {
```

```
                    "port_filter": true
                },
                "binding:host_id": "compute-0",
                "wrs-binding:mtu": 1500,
                "binding:vif_type": "bridge",
                "device_id": "dhcp596d6b96-7696-5200-a782-fa1c60fe4171-
f652780a-7a9d-4667-8df4-5c8632728be9",
                "device_owner": "network:dhcp",
                "fixed_ips": [
                    {
                        "ip_address": "192.168.101.9",
                        "subnet_id": "34efd537-7a72-4fcd-b837-9874caf34117"
                    }
                ],
                "id": "30a92b3d-3353-4c24-9963-d0177b38ad59",
                "mac_address": "fa:16:3e:f5:db:a6",
                "wrs-binding:mac_filtering": false
                "name": "",
                "network_id": "f652780a-7a9d-4667-8df4-5c8632728be9",
                "security_groups": [],
                "status": "ACTIVE",
                "tenant_id": "0590d9fa3dd74bfe9bdf7ed4e5331a10"
            },
            {
                "admin_state_up": true,
                "binding:capabilities": {
                    "port_filter": true
                },
                "binding:host_id": "compute-1",
                "wrs-binding:mtu": 1500,
                "wrs-binding:vif_model": "virtio",
                "binding:vif_type": "bridge",
                "device_id": "2e934b37-772e-451a-b64a-cd68d9f8ae42",
                "device_owner": "compute:nova",
                "fixed_ips": [
                    {
                        "ip_address": "192.168.101.11",
                        "subnet_id": "34efd537-7a72-4fcd-b837-9874caf34117"
                    }
                ],
                "id": "3ec39233-315c-474b-9f08-482e70c264c7",
                "mac_address": "fa:16:3e:23:0c:bc",
                "wrs-binding:mac_filtering": false
                "name": "",
                "network_id": "f652780a-7a9d-4667-8df4-5c8632728be9",
                "security_groups": [],
                "status": "ACTIVE",
                "tenant_id": "0590d9fa3dd74bfe9bdf7ed4e5331a10"
            },
            {
                "admin_state_up": true,
                "binding:capabilities": {
                    "port_filter": true
                },
                "wrs-binding:mtu": 1500,
                "binding:vif_type": "bridge",
                "device_id": "2c5c2553-36f2-4db4-8769-6ab71b6e2b1e",
                "device_owner": "network:floatingip",
                "fixed_ips": [
                    {
```

```
                          "ip_address": "192.168.1.4",
                          "subnet_id": "b282ef86-2584-4a02-9b58-69d6233952a2"
                  }
              ],
              "id": "5e5a6728-81af-4b2d-b068-8a8e56847a1e",
              "mac_address": "fa:16:3e:67:22:12",
              "wrs-binding:mac_filtering": false
              "name": "",
              "network_id": "b9475152-11d3-4bda-95c7-fb26a3ad3876",
              "security_groups": [],
              "status": "UNKNOWN",
              "tenant_id": ""
          },
          {

              "admin_state_up": true,
              "binding:capabilities": {
                  "port_filter": true
              },
              "binding:host_id": "compute-1",
              "wrs-binding:mtu": 1500,
              "wrs-binding:vif_model": "avp",
              "binding:vif_type": "bridge",
              "device_id": "2e934b37-772e-451a-b64a-cd68d9f8ae42",
              "device_owner": "compute:nova",
              "fixed_ips": [],
              "id": "8c99a79f-212c-45e5-89e6-8aa9ed3b5fcb",
              "mac_address": "fa:16:3e:46:47:46",
              "wrs-binding:mac_filtering": false
              "name": "",
              "network_id": "2c0896cf-d118-4dca-9760-b4d97e3c7ec3",
              "security_groups": [],
              "status": "ACTIVE",
              "tenant_id": "0590d9fa3dd74bfe9bdf7ed4e5331a10"
          },
          {

              "admin_state_up": true,
              "binding:capabilities": {
                  "port_filter": true
              },
              "binding:host_id": "compute-0",
              "wrs-binding:mtu": 1500,
              "binding:vif_type": "bridge",
              "device_id": "ce58c529-ba73-47f7-a409-a438ceced112",
              "device_owner": "network:router_gateway",
              "fixed_ips": [
                  {
                      "ip_address": "192.168.1.3",
                      "subnet_id": "b282ef86-2584-4a02-9b58-69d6233952a2"
                  }
              ],
              "id": "10ce8655-96de-45f4-952c-a830ddf1f0d9",
              "mac_address": "fa:16:3e:a4:1d:67",
              "wrs-binding:mac_filtering": false
              "name": "",
              "network_id": "b9475152-11d3-4bda-95c7-fb26a3ad3876",
              "security_groups": [],
              "status": "ACTIVE",
              "tenant_id": ""
          },
          {
```

```
                    "admin_state_up": true,
                    "binding:capabilities": {
                        "port_filter": true
                    },
                    "wrs-binding:mtu": 1500,
                    "binding:vif_type": "bridge",
                    "device_id": "f2ade26f-5a16-4fef-b6f7-9bd9769ea1f4",
                    "device_owner": "network:floatingip",
                    "fixed_ips": [
                        {
                            "ip_address": "192.168.1.5",
                            "subnet_id": "b282ef86-2584-4a02-9b58-69d6233952a2"
                        }
                    ],
                    "id": "573e154f-ec13-4e39-afcc-8e79d2089589",
                    "mac_address": "fa:16:3e:23:37:59",
                    "wrs-binding:mac_filtering": false
                    "name": "",
                    "network_id": "b9475152-11d3-4bda-95c7-fb26a3ad3876",
                    "security_groups": [],
                    "status": "UNKNOWN",
                    "tenant_id": ""
                },
                {
                    "admin_state_up": true,
                    "binding:capabilities": {
                        "port_filter": true
                    },
                    "binding:host_id": "compute-1",
                    "wrs-binding:mtu": 1500,
                    "wrs-binding:vif_model": "avp",
                    "binding:vif_type": "bridge",
                    "device_id": "2e934b37-772e-451a-b64a-cd68d9f8ae42",
                    "device_owner": "compute:nova",
                    "fixed_ips": [],
                    "id": "9c9b1182-9f7d-4e21-910c-c4441a23d397",
                    "mac_address": "fa:16:3e:75:5f:61",
                    "wrs-binding:mac_filtering": false
                    "name": "",
                    "network_id": "7e5ed852-a990-4fc5-89a2-b17093ca1982",
                    "security_groups": [],
                    "status": "ACTIVE",
                    "tenant_id": "0590d9fa3dd74bfe9bdf7ed4e5331a10"
                },
                {
                    "admin_state_up": true,
                    "binding:capabilities": {
                        "port_filter": true
                    },
                    "binding:host_id": "compute-0",
                    "wrs-binding:mtu": 1500,
                    "binding:vif_type": "bridge",
                    "device_id": "dhcp596d6b96-7696-5200-a782-
fa1c60fe4171-9472a8ab-9205-43ef-a460-5f01f031791a",
                    "device_owner": "network:dhcp",
                    "fixed_ips": [
                        {
                            "ip_address": "192.168.201.6",
                            "subnet_id": "9aa900f4-522b-4b83-ba93-57f7d92da5d5"
                        }
```

```
            ],
            "id": "62872b6d-3e74-4838-9cf3-99736d0f3919",
            "mac_address": "fa:16:3e:92:0b:14",
            "wrs-binding:mac_filtering": false
            "name": "",
            "network_id": "9472a8ab-9205-43ef-a460-5f01f031791a",
            "security_groups": [],
            "status": "ACTIVE",
            "tenant_id": "d8753af85cef49a4bf5f95208c4957f3"
        }
    ]
}
```

This operation does not return a response body.

3.9.2. Show port

Method	URI	Description
GET	`/v2.0/ports/{port_id}`	Shows information for a specified port.

This is an existing OpenStack API. The documentation that follows lists only the fields that are new or modified. For a detailed description of existing and unmodified fields please refer to the standard OpenStack API documentation.

Normal response codes: 200

Error response codes: itemNotFound (401), unauthorized (404)

3.9.2.1. Request

This table shows the URI parameters for the show port request:

Name	Type	Description
`{port_id}`	UUID	The UUID for a port.

This operation does not accept a request body.

3.9.2.2. Response

This table shows the body parameters for the show port response:

Name	Type	Description
`wrs-binding:mac_filtering`	Bool *(Optional)*	The state of source MAC address filtering on the port. If this is `true` then the attached vswitch enforces that all ingress packets have a source MAC address that matches the port MAC address.
`wrs-binding:mtu`	Integer *(Optional)*	The maximum transmit unit (MTU). This value is inherited from the tenant network that attaches to this port.
`wrs-binding:vif_model`	String *(Optional)*	The type of virtual networking device that is presented to the VM instance. This value is only visible if the device_owner is a VM instance port (i.e., device_owner="compute:nova").

Example 3.34. Show port: JSON response

```
{
    "port": {
        "admin_state_up": true,
        "binding:capabilities": {
            "port_filter": true
        },
        "binding:host_id": "compute-1",
        "wrs-binding:mtu": 1500,
        "wrs-binding:vif_model": "virtio",
        "binding:vif_type": "bridge",
        "device_id": "2e934b37-772e-451a-b64a-cd68d9f8ae42",
        "device_owner": "compute:nova",
        "fixed_ips": [
            {
                "ip_address": "192.168.101.11",
```

```
                     "subnet_id": "34efd537-7a72-4fcd-b837-9874caf34117"
            }
        ],
        "id": "3ec39233-315c-474b-9f08-482e70c264c7",
        "mac_address": "fa:16:3e:23:0c:bc",
        "wrs-binding:mac_filtering": false
        "name": "",
        "network_id": "f652780a-7a9d-4667-8df4-5c8632728be9",
        "security_groups": [],
        "status": "ACTIVE",
        "tenant_id": "0590d9fa3dd74bfe9bdf7ed4e5331a10"
    }
}
```

This operation does not return a response body.

3.10. Router

The Router entity is an existing OpenStack API. It has been extended to add the following HP Helion OpenStack Carrier Grade functionality.

- The host attribute has been added to reflect which compute host implements the virtual router.

Method	URI	Description
GET	`/v2.0/routers`	Lists logical routers that are accessible to the tenant who submits the request.
GET	`/v2.0/routers/{router_id}`	Shows details for a specified router.

3.10.1. List routers

Method	URI	Description
GET	`/v2.0/routers`	Lists logical routers that are accessible to the tenant who submits the request.

This is an existing OpenStack API. The documentation that follows lists only the fields that are new or modified. For a detailed description of existing and unmodified fields please refer to the standard OpenStack API documentation.

Normal response codes: 200

Error response codes: itemNotFound (401)

3.10.1.1. Request

This operation does not accept a request body.

3.10.1.2. Response

This table shows the body parameters for the list routers response:

Name	Type	Description
`routers`	List (Optional)	The list of virtual routers.
`wrs-net:host`	String (Optional)	The host node where this virtual router is implemented. **Only visible to admin users.**

Example 3.35. List routers: JSON response

```
{
    "routers": [
        {
            "admin_state_up": true,
            "external_gateway_info": {
                "enable_snat": true,
                "network_id": "b9475152-11d3-4bda-95c7-fb26a3ad3876"
            },
            "wrs-net:host": "compute-0",
            "id": "ea2baef7-d84b-44c4-82e7-f274ab5e8b6f",
            "name": "tenant1-router",
            "routes": [],
            "status": "ACTIVE",
            "tenant_id": "0590d9fa3dd74bfe9bdf7ed4e5331a10"
        },
        {
            "admin_state_up": true,
            "external_gateway_info": {
                "enable_snat": true,
                "network_id": "b9475152-11d3-4bda-95c7-fb26a3ad3876"
            },
            "wrs-net:host": "compute-0",
            "id": "ce58c529-ba73-47f7-a409-a438ceced112",
```

```
                "name": "tenant2-router",
                "routes": [],
                "status": "ACTIVE",
                "tenant_id": "d8753af85cef49a4bf5f95208c4957f3"
            }
        ]
}
```

This operation does not return a response body.

3.10.2. Show router details

Method	URI	Description
GET	`/v2.0/routers/{router_id}`	Shows details for a specified router.

This is an existing OpenStack API. The documentation that follows lists only the fields that are new or modified. For a detailed description of existing and unmodified fields please refer to the standard OpenStack API documentation.

Normal response codes: 200

Error response codes: itemNotFound (401), forbidden (403), unauthorized (404)

3.10.2.1. Request

This table shows the URI parameters for the show router details request:

Name	Type	Description
`{router_id}`	UUID	The UUID for a router.

This operation does not accept a request body.

3.10.2.2. Response

This table shows the body parameters for the show router details response:

Name	Type	Description
`wrs-net:host`	String	The host node where this virtual router is implemented.
	(Optional)	**Only visible to admin users.**

Example 3.36. Show router details: JSON response

```
{
    "router": {
        "admin_state_up": true,
        "external_gateway_info": {
            "enable_snat": true,
            "network_id": "b9475152-11d3-4bda-95c7-fb26a3ad3876"
        },
        "wrs-net:host": "compute-0",
        "id": "ea2baef7-d84b-44c4-82e7-f274ab5e8b6f",
        "name": "tenant1-router",
        "routes": [],
        "status": "ACTIVE",
        "tenant_id": "0590d9fa3dd74bfe9bdf7ed4e5331a10"
    }
}
```

This operation does not return a response body.

4. Block Storage API v2 extensions

HP Helion OpenStack Carrier Grade extensions to the OpenStack Block Storage API such as backup status and export/import actions for volumes and snapshots.

The typical port used for the Block Storage REST API is 8776. However, proper technique would be to look up the cinderv2 service endpoint in keystone.

Method	URI	Description
	Extensions	
GET	`/v2/{tenant_id}/extensions`	Lists all extensions.
GET	`/v2/{tenant_id}/exten-sions/{extension_alias}`	Gets information about a specified extension.
	Volumes	
GET	`/v2/{tenant_id}/volumes/detail`	Get information about system volumes.
GET	`/v2/{tenant_id}/vol-umes/{volume_id}`	Get information about a specified volume.
POST	`/v2/{tenant_id}/vol-umes/{volume_id}/action`	Executes the specified action or command on the specified volume.
	Snapshots	
GET	`/v2/{tenant_id}/snapshots/detail`	Get information about system volume snapshots.
GET	`/v2/{tenant_id}/snap-shots/{snapshot_id}`	Get information of a specific volume snapshot.
POST	`/v2/{tenant_id}/snap-shots/{snapshot_id}/action`	Executes the specified action or command on the specified volume snapshot.

4.1. Extensions

The Extensions entity lists all available extensions

Method	URI	Description
GET	`/v2/{tenant_id}/extensions`	Lists all extensions.
GET	`/v2/{tenant_id}/exten-sions/{extension_alias}`	Gets information about a specified extension.

4.1.1. List extensions

Method	URI	Description
GET	`/v2/{tenant_id}/extensions`	Lists all extensions.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

4.1.1.1. Request

This table shows the URI parameters for the list extensions request:

Name	Type	Description
`{tenant_id}`	String	The ID for the tenant or account in a multi-tenancy cloud.

This operation does not accept a request body.

4.1.1.2. Response

This table shows the body parameters for the list extensions response:

Name	Type	Description
`namespace`	String *(Optional)*	Indicates namespace of the extension.
`name`	String *(Optional)*	Indicates name of the extension.
`updated`	String *(Optional)*	Indicates updated time of the extension.
`description`	String *(Optional)*	Indicates description of the extension.
`alias`	String *(Optional)*	Indicates alias of the extension.
`links`	List *(Optional)*	A list of links for the extension.

Example 4.1. List extensions: JSON response

```
{
   "extensions" : [
      ...
      {
         "namespace" : "http://docs.windriver.com/volume/ext/wrs-snapshot/api/
v1.0",
         "name" : "WrsSnapshotExportAction",
```

```
            "updated" : "2014-08-16T00:00:00+00:00",
            "description" : "Enable snapshot export to file",
            "alias" : "wrs-snapshot",
            "links" : []
        },
        {
            "namespace" : "http://docs.windriver.com/volume/ext/wrs-volume/api/
v1.0",
            "name" : "WrsVolumeExport",
            "updated" : "2014-08-11T00:00:00+00:00",
            "description" : "Enable volume export/import",
            "alias" : "wrs-volume",
            "links" : []
        },
        ...
    ]
}
```

This operation does not return a response body.

4.1.2. Get extension

Method	URI	Description
GET	`/v2/{tenant_id}/exten-sions/{extension_alias}`	Gets information about a specified extension.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

4.1.2.1. Request

This table shows the URI parameters for the get extension request:

Name	Type	Description
`{tenant_id}`	String	The ID for the tenant or account in a multi-tenancy cloud.
`{extension_alias}`	String	The alias for the extension to list.

This operation does not accept a request body.

4.1.2.2. Response

This table shows the body parameters for the get extension response:

Name	Type	Description
`namespace`	String (Optional)	Indicates namespace of the extension.
`name`	String (Optional)	Indicates name of the extension.
`updated`	String (Optional)	Indicates updated time of the extension.
`description`	String (Optional)	Indicates description of the extension.
`alias`	String (Optional)	Indicates alias of the extension.
`links`	List (Optional)	A list of links for the extension.

Example 4.2. Get extension: JSON response

```
{
    "extension" : {
        "namespace" : "http://docs.windriver.com/volume/ext/wrs-volume/api/v1.
0",
        "name" : "WrsVolumeExport",
```

```
        "updated" : "2014-08-11T00:00:00+00:00",
        "description" : "Enable volume export/import",
        "alias" : "wrs-volume",
        "links" : []
    }
}

OR

{
    "extension" : {
        "namespace" : "http://docs.windriver.com/volume/ext/wrs-snapshot/api/v1.
0",
        "name" : "WrsSnapshotExportAction",
        "updated" : "2014-08-16T00:00:00+00:00",
        "description" : "Enable snapshot export to file",
        "alias" : "wrs-snapshot",
        "links" : []
    }
}
```

This operation does not return a response body.

4.2. Volumes

HP Helion OpenStack Carrier Grade extensions include export and import actions for performing backup and restores of volumes, and a backup status attribute to indicate the status of the new actions.

Method	URI	Description
GET	`/v2/{tenant_id}/volumes/detail`	Get information about system volumes.
GET	`/v2/{tenant_id}/vol-umes/{volume_id}`	Get information about a specified volume.
POST	`/v2/{tenant_id}/vol-umes/{volume_id}/action`	Executes the specified action or command on the specified volume.

4.2.1. Get information about system volumes

Method	URI	Description
GET	`/v2/{tenant_id}/volumes/detail`	Get information about system volumes.

Preconditions

• The specified volume must exist in all case.

Normal response codes: 200

4.2.1.1. Request

This table shows the URI parameters for the get information about system volumes request:

Name	Type	Description
`{tenant_id}`	String	The ID for the tenant or account in a multi-tenancy cloud.

This operation does not accept a request body.

4.2.1.2. Response

This table shows the body parameters for the get information about system volumes response:

Name	Type	Description
`wrs-volume:backup_status`	String *(Required)*	Indicates backup status.

Example 4.3. Get information about system volumes: JSON response

```
{
   "volumes" : [
      {
         "wrs-volume:backup_status" : "Export completed at 2015-02-27
16:35:53.545339",
         "volume_type" : "None",
         "status" : "available",
         "size" : 1,
         "created_at" : "2015-02-27T16:26:08.164607",
         "id" : "b7db512f-463e-4720-8fbd-154c0f2bc2ae",
         "metadata" : {},
         "attachments" : [],
         "os-volume-replication:driver_data" : null,
         "os-vol-mig-status-attr:migstat" : null,
         "display_name" : null,
         "availability_zone" : "nova",
         "display_description" : null,
         "encrypted" : false,
         "os-vol-mig-status-attr:name_id" : null,
         "os-vol-host-attr:host" : "controller@lvm#lvm",
         "os-volume-replication:extended_status" : null,
```

```
                "snapshot_id" : null,
                "os-vol-tenant-attr:tenant_id" : "e0741109067649a8899936e9fefda95b",
                "bootable" : "false",
                "source_volid" : null
        },
        {
                "wrs-volume:backup_status" : "Import completed at 2015-02-27
15:04:29.135579",
                "volume_type" : "None",
                "status" : "available",
                "size" : 1,
                "created_at" : "2015-02-27T14:04:34.763953",
                "id" : "27080551-9d88-4cf0-aa85-c1392dbf38f4",
                "metadata" : {},
                "attachments" : [],
                "os-volume-replication:driver_data" : null,
                "os-vol-mig-status-attr:migstat" : null,
                "display_name" : null,
                "availability_zone" : "nova",
                "display_description" : null,
                "encrypted" : false,
                "os-vol-mig-status-attr:name_id" : null,
                "os-vol-host-attr:host" : "controller@lvm#lvm",
                "os-volume-replication:extended_status" : null,
                "snapshot_id" : null,
                "os-vol-tenant-attr:tenant_id" : "e0741109067649a8899936e9fefda95b",
                "bootable" : "false",
                "source_volid" : null
        },
        {
                "wrs-volume:backup_status" : "Snapshot export completed at 2015-02-27
20:57:29.323714",
                "volume_type" : "None",
                "status" : "available",
                "size" : 1,
                "created_at" : "2015-02-27T13:44:55.317995",
                "id" : "2c4f094b-f6d8-4ff6-800e-e5998cb4d6fa",
                "metadata" : {},
                "attachments" : [],
                "os-volume-replication:driver_data" : null,
                "os-vol-mig-status-attr:migstat" : null,
                "display_name" : null,
                "availability_zone" : "nova",
                "display_description" : null,
                "encrypted" : false,
                "os-vol-mig-status-attr:name_id" : null,
                "os-vol-host-attr:host" : "controller@lvm#lvm",
                "os-volume-replication:extended_status" : null,
                "snapshot_id" : null,
                "os-vol-tenant-attr:tenant_id" : "e0741109067649a8899936e9fefda95b",
                "bootable" : "false",
                "source_volid" : null
        }
    ]
}
```

This operation does not return a response body.

4.2.2. Get information about a specified volume

Method	URI	Description
GET	`/v2/{tenant_id}/vol-umes/{volume_id}`	Get information about a specified volume.

Preconditions

- The specified volume must exist in all case.

Normal response codes: 200

4.2.2.1. Request

This table shows the URI parameters for the get information about a specified volume request:

Name	Type	Description
`{tenant_id}`	String	The ID for the tenant or account in a multi-tenancy cloud.
`{volume_id}`	UUID	The ID for the volume to list.

This operation does not accept a request body.

4.2.2.2. Response

This table shows the body parameters for the get information about a specified volume response:

Name	Type	Description
`wrs-volume:backup_status`	String *(Required)*	Indicates backup status.

Example 4.4. Get information about a specified volume: JSON response

```
{
   "volumes" : [
      {
         "wrs-volume:backup_status" : "Import completed at 2015-02-27
15:04:29.135579",
         "volume_type" : "None",
         "status" : "available",
         "size" : 1,
         "created_at" : "2015-02-27T14:04:34.763953",
         "id" : "27080551-9d88-4cf0-aa85-c1392dbf38f4",
         "metadata" : {},
         "attachments" : [],
         "os-volume-replication:driver_data" : null,
         "os-vol-mig-status-attr:migstat" : null,
         "display_name" : null,
         "availability_zone" : "nova",
         "display_description" : null,
         "encrypted" : false,
         "os-vol-mig-status-attr:name_id" : null,
```

```
              "os-vol-host-attr:host" : "controller@lvm#lvm",
              "os-volume-replication:extended_status" : null,
              "snapshot_id" : null,
              "os-vol-tenant-attr:tenant_id" : "e0741109067649a8899936e9fefda95b",
              "bootable" : "false",
              "source_volid" : null
         },
    ]
}
```

This operation does not return a response body.

4.2.3. Executes the specified action or command on the specified volume.

Method	URI	Description
POST	`/v2/{tenant_id}/vol-umes/{volume_id}/action`	Executes the specified action or command on the specified volume.

Preconditions

- The specified volume must exist in all case.

Normal response codes: 200

4.2.3.1. Request

This table shows the URI parameters for the executes the specified action or command on the specified volume. request:

Name	Type	Description
`{tenant_id}`	String	The ID for the tenant or account in a multi-tenancy cloud.
`{volume_id}`	UUID	The ID for the volume to list.

This table shows the body parameters for the executes the specified action or command on the specified volume. request:

Name	Type	Description
`wrs-volume:os-volume_export`	String *(Required)*	Export volume to a file
`wrs-volume:os-volume_import`	String *(Required)*	Import a volume from a file • file_name: "VolumeExportName.tgz".

This operation does not accept a request body.

4.2.3.2. Response

This table shows the body parameters for the executes the specified action or command on the specified volume. response:

Name	Type	Description
`volume_type`	String *(Required)*	Indicates the volume type.
`updated_at`	String *(Required)*	Indicates when the action was performed.
`status`	String *(Required)*	Indicates the state of the export or import action.
`id`	UUID *(Required)*	Indicates the volume UUID.

Name	Type	Description
`display_description`	String *(Required)*	Volume descrition if any.
`size`	Int *(Required)*	Indicates the volume size in Gbyte.

Example 4.5. Executes the specified action or command on the specified volume.: JSON response

```
{
    'wrs-volume:os-volume_export' : {
        'volume_type' : null,
        'updated_at' : '2015-02-27T14:04:35.201969',
        'status' : 'exporting',
        'id' : '27080551-9d88-4cf0-aa85-c1392dbf38f4',
        'display_description' : null,
        'size' : 1
    }
}
or
{
    'wrs-volume:os-volume_import' : {
        'volume_type' : null,
        'updated_at' : '2015-02-27T15:03:54.045796',
        'status' : 'importing',
        'id' : '27080551-9d88-4cf0-aa85-c1392dbf38f4',
        'display_description' : null,
        'size' : 1
    }
}
```

This operation does not return a response body.

4.3. Snapshots

HP Helion OpenStack Carrier Grade extensions include export actions for performing back-up volumes already attached to a VM, and a backup status attribute to indicate the status of the new actions.

Method	URI	Description
GET	`/v2/{tenant_id}/snapshots/detail`	Get information about system volume snapshots.
GET	`/v2/{tenant_id}/snap-shots/{snapshot_id}`	Get information of a specific volume snapshot.
POST	`/v2/{tenant_id}/snap-shots/{snapshot_id}/action`	Executes the specified action or command on the specified volume snapshot.

4.3.1. Get information about system volume snapshots.

Method	URI	Description
GET	`/v2/{tenant_id}/snapshots/detail`	Get information about system volume snapshots.

Preconditions

• The specified volume snapshot must exist in all case.

Normal response codes: 200

4.3.1.1. Request

This table shows the URI parameters for the get information about system volume snap-shots. request:

Name	Type	Description
`{tenant_id}`	String	The ID for the tenant or account in a multi-tenancy cloud.

This operation does not accept a request body.

4.3.1.2. Response

This table shows the body parameters for the get information about system volume snap-shots. response:

Name	Type	Description
`wrs-snapshot:backup_status`	String (Required)	Indicates backup status.

Example 4.6. Get information about system volume snapshots.: JSON response

```
{
    "snapshots" : [
        {
            "volume_id" : "f15dcbfb-8b41-4fff-adb8-77a4162a318b",
            "status" : "available",
            "display_description" : null,
            "display_name" : null,
            "size" : 1,
            "created_at" : "2015-02-27T13:19:02.380453",
            "os-extended-snapshot-attributes:project_id" :
"e0741109067649a8899936e9fefda95b",
            "wrs-snapshot:backup_status" : "Export completed at 2015-02-27
13:19:48.914344",
            "id" : "7b220cb7-212f-411e-a8cd-41e6bdbac724",
            "metadata" : {},
            "os-extended-snapshot-attributes:progress" : "100%"
        },
        {
            "volume_id" : "2c4f094b-f6d8-4ff6-800e-e5998cb4d6fa",
            "status" : "available",
```

```
        "display_description" : null,
        "display_name" : null,
        "size" : 1,
        "created_at" : "2015-02-27T20:56:32.033427",
        "os-extended-snapshot-attributes:project_id" :
"e0741109067649a8899936e9fefda95b",
        "wrs-snapshot:backup_status" : "Export completed at 2015-02-27
20:57:29.279574",
        "id" : "0aa45e0c-74ea-433e-b8f3-0dc778d3972b",
        "metadata" : {},
        "os-extended-snapshot-attributes:progress" : "100%"
    }
  ]
}
```

This operation does not return a response body.

4.3.2. Get information of a specific volume snapshot.

Method	URI	Description
GET	/v2/{tenant_id}/snap-shots/{snapshot_id}	Get information of a specific volume snapshot.

Preconditions

• The specified volume snapshot must exist in all case.

Normal response codes: 200

4.3.2.1. Request

This table shows the URI parameters for the get information of a specific volume snapshot. request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{snapshot_id}	UUID	The ID for the snapshot to list.

This operation does not accept a request body.

4.3.2.2. Response

This table shows the body parameters for the get information of a specific volume snap-shot. response:

Name	Type	Description
wrs-snapshot:backup_status	String *(Required)*	Indicates backup status.

Example 4.7. Get information of a specific volume snapshot.: JSON response

```
{
    "snapshot" : {
        "volume_id" : "2c4f094b-f6d8-4ff6-800e-e5998cb4d6fa",
        "status" : "available",
        "display_description" : null,
        "display_name" : null,
        "size" : 1,
        "created_at" : "2015-02-27T20:56:32.033427",
        "os-extended-snapshot-attributes:project_id" :
"e0741109067649a8899936e9fefda95b",
        "wrs-snapshot:backup_status" : "Export completed at 2015-02-27 20:57:29.
279574",
        "id" : "0aa45e0c-74ea-433e-b8f3-0dc778d3972b",
        "metadata" : {},
        "os-extended-snapshot-attributes:progress" : "100%"
    }
}
```

This operation does not return a response body.

4.3.3. Executes the specified action or command on the specified volume snapshot.

Method	URI	Description
POST	/v2/{tenant_id}/snap-shots/{snapshot_id}/action	Executes the specified action or command on the specified volume snapshot.

Preconditions

• The specified volume snapshot must exist in all case.

Normal response codes: 200

4.3.3.1. Request

This table shows the URI parameters for the executes the specified action or command on the specified volume snapshot. request:

Name	Type	Description
{tenant_id}	String	The ID for the tenant or account in a multi-tenancy cloud.
{snapshot_id}	UUID	The ID for the snapshot to list.

This table shows the body parameters for the executes the specified action or command on the specified volume snapshot. request:

Name	Type	Description
wrs-snapshot:os-snapshot_export	String *(Required)*	Export volume snapshot to a file

This operation does not accept a request body.

4.3.3.2. Response

This table shows the body parameters for the executes the specified action or command on the specified volume snapshot. response:

Name	Type	Description
volume_type	String *(Required)*	Indicates the volume type.
updated_at	String *(Required)*	Indicates when the action was performed.
status	String *(Required)*	Indicates the state of the volume snapshot export action.
id	UUID *(Required)*	Indicates the volume UUID.
display_description	String *(Required)*	Volume descrition if any.

Name	Type	Description
volume_size	Int *(Required)*	Indicates the volume size in Gbyte.

Example 4.8. Executes the specified action or command on the specified volume snapshot.: JSON response

```
{
    "wrs-snapshot:os-export_snapshot" : {
        "volume_type" : null,
        "updated_at" : "2015-03-03T15:32:31.386661",
        "status" : "exporting",
        "volume_size" : 1,
        "id" : "9ad36199-c5b3-44bf-9273-c298ab7a0a2b",
        "display_description" : null
    }
}
```

This operation does not return a response body.

5. Telemetry API v2 extensions

HP Helion OpenStack Carrier Grade extensions to the OpenStack Telemetry API such as Pipelines for outputting samples to CSV files.

The typical port used for the Telemetry REST API is 8777. However, proper technique would be to look up the ceilometer service endpoint in keystone.

Method	URI	Description
Extensions		
GET	`/v2/extensions`	Lists all extensions.
GET	`/v2/extensions/{extension_alias}`	Gets information about a specified extension.
Pipeline		
GET	`/v2/wrs-pipelines`	Lists all pipelines.
GET	`/v2/wrs-pipelines/{pipeline_id}`	Shows detailed information about a specific pipeline that outputs to CSV.
PUT	`/v2/wrs-pipelines/{pipeline_id}`	Modifies a specific pipeline.

5.1. Extensions

The Extensions entity lists all available extensions

Method	URI	Description
GET	`/v2/extensions`	Lists all extensions.
GET	`/v2/extensions/{extension_alias}`	Gets information about a specified extension.

5.1.1. List extensions

Method	URI	Description
GET	`/v2/extensions`	Lists all extensions.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

5.1.1.1. Request

This operation does not accept a request body.

5.1.1.2. Response

This table shows the body parameters for the list extensions response:

Name	Type	Description
`namespace`	String (Optional)	Indicates namespace of the extension.
`name`	String (Optional)	Indicates name of the extension.
`updated`	String (Optional)	Indicates updated time of the extension.
`description`	String (Optional)	Indicates description of the extension.
`alias`	String (Optional)	Indicates alias of the extension.
`links`	List (Optional)	A list of links for the extension.

Example 5.1. List extensions: JSON response

```
{
   "extensions" : [
      {
         "namespace" : "http://docs.windriver.org/tis/ext/wrs-pipelines/v1",
         "name" : "wrs-pipelines",
         "updated" : "2014-10-01T12:00:00-00:00",
         "description" : "Windriver Telemetry Pipelines for managing the
 writing of Ceilometer PMs to a Comma-Separated-Value file.",
         "alias" : "wrs-pipelines",
         "links" : []
      }
   ]
}
```

This operation does not return a response body.

5.1.2. List extensions

Method	URI	Description
GET	/v2/extensions/{extension_alias}	Gets information about a specified extension.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

5.1.2.1. Request

This table shows the URI parameters for the list extensions request:

Name	Type	Description
{extension_alias}	String	The alias for the extension to list.

This operation does not accept a request body.

5.1.2.2. Response

This table shows the body parameters for the list extensions response:

Name	Type	Description
namespace	String (Optional)	Indicates namespace of the extension.
name	String (Optional)	Indicates name of the extension.
updated	String (Optional)	Indicates updated time of the extension.
description	String (Optional)	Indicates description of the extension.
alias	String (Optional)	Indicates alias of the extension.
links	List (Optional)	A list of links for the extension.

Example 5.2. List extensions: JSON response

```
{
   "extensions" : {
      "namespace" : "http://docs.windriver.org/tis/ext/wrs-pipelines/v1",
      "name" : "wrs-pipelines",
      "updated" : "2014-10-01T12:00:00-00:00",
      "description" : "Windriver Telemetry Pipelines for managing the writing
 of Ceilometer PMs to a Comma-Separated-Value file.",
```

```
        "alias" : "wrs-pipelines",
        "links" : []
    }
}
```

This operation does not return a response body.

5.2. Pipeline

The Pipeline entity is modeled by extensions to manipulate how samples are stored to CSV files

Method	URI	Description
GET	`/v2/wrs-pipelines`	Lists all pipelines.
GET	`/v2/wrs-pipelines/{pipeline_id}`	Shows detailed information about a specific pipeline that outputs to CSV.
PUT	`/v2/wrs-pipelines/{pipeline_id}`	Modifies a specific pipeline.

5.2.1. List pipelines

Method	URI	Description
GET	`/v2/wrs-pipelines`	Lists all pipelines.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

5.2.1.1. Request

This operation does not accept a request body.

5.2.1.2. Response

This table shows the body parameters for the list pipelines response:

Name	Type	Description
`name`	String *(Optional)*	Indicates name of the CSV pipeline.
`compress`	Bool *(Optional)*	Indicates whether or not to compress the CSV file when rotating.
`enabled`	Bool *(Optional)*	Indicates whether or not to enable this CSV pipeline.
`meters`	List *(Optional)*	A regular expression of the meters piped to CSV.
`location`	String *(Optional)*	The full path to the CSV output file. The folder is restricted.
`backup_count`	Int *(Optional)*	The number of backed up CSV files to preserve.
`max_bytes`	Int *(Optional)*	The size in bytes to allow the CSV file to grow before triggering rotating and compressing.

Example 5.3. List pipelines: JSON response

```
[
{
 "name": "csv",
 "compress": true,
 "enabled": true,
 "meters": ["!vswitch.*"],
 "location": "/opt/cgcs/ceilometer/csv/pm.csv",
 "backup_count": 5,
 "max_bytes": 10000000
}
```

```
]
```

This operation does not return a response body.

5.2.2. Show pipeline

Method	URI	Description
GET	`/v2/wrs-pipelines/{pipeline_id}`	Shows detailed information about a specific pipeline that outputs to CSV.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

5.2.2.1. Request

This table shows the URI parameters for the show pipeline request:

Name	Type	Description
`{pipeline_id}`	String	The name for the pipeline.

This operation does not accept a request body.

5.2.2.2. Response

This table shows the body parameters for the show pipeline response:

Name	Type	Description
`name`	String *(Optional)*	Indicates name of the CSV pipeline.
`compress`	Bool *(Optional)*	Indicates whether or not to compress the CSV file when rotating.
`enabled`	Bool *(Optional)*	Indicates whether or not to enable this CSV pipeline.
`meters`	List *(Optional)*	A regular expression of the meters piped to CSV.
`location`	String *(Optional)*	The full path to the CSV output file. The folder is restricted.
`backup_count`	Int *(Optional)*	The number of backed up CSV files to preserve.
`max_bytes`	Int *(Optional)*	The size in bytes to allow the CSV file to grow before triggering rotating and compressing.

Example 5.4. Show pipeline: JSON response

```
{
 "name": "csv",
 "compress": true,
 "enabled": true,
```

```
"meters": ["!vswitch.*"],
"location": "/opt/cgcs/ceilometer/csv/pm.csv",
"backup_count": 5,
"max_bytes": 10000000
}
```

This operation does not return a response body.

5.2.3. Modify pipeline

Method	URI	Description
PUT	`/v2/wrs-pipelines/{pipeline_id}`	Modifies a specific pipeline.

Performance of the entire system can be impacted by storing too much data to CSV

Normal response codes: 200

Error response codes: badMediaType (415), NetworkNotFound (400)

5.2.3.1. Request

This table shows the URI parameters for the modify pipeline request:

Name	Type	Description
`{pipeline_id}`	String	The name for the pipeline.

This table shows the body parameters for the modify pipeline request:

Name	Type	Description
`compress`	Bool *(Required)*	Indicates whether or not to compress the CSV file when rotating.
`enabled`	Bool *(Required)*	Indicates whether or not to enable this CSV pipeline.
`meters`	List *(Required)*	A regular expression of the meters piped to CSV
`location`	String *(Required)*	The full path to the CSV output file. The folder is restricted
`backup_count`	Int *(Required)*	The number of backed up CSV files to preserve. It can be any number from 0. No backup file will be generated when backup_count is set to 0
`max_bytes`	Int *(Required)*	The size in bytes to allow the CSV file to grow before triggering rotating and compressing. It can be any number from 0. No rotating and compressing will be triggered when max_bytes is set to 0

Example 5.5. Modify pipeline: JSON request

```
{"name": "csv",
 "compress": false,
 "enabled": false,
 "meters": ["!vswitch.*", "!cpu_util"],
 "location": "/opt/cgcs/ceilometer/csv/pm2.csv",
 "backup_count": 4,
 "max_bytes": 9999999
}
```

This operation does not accept a request body.

5.2.3.2. Response

This table shows the body parameters for the modify pipeline response:

Name	Type	Description
name *(Optional)*	String	Indicates name of the CSV pipeline.
compress *(Optional)*	Bool	Indicates whether or not to compress the CSV file when rotating.
enabled *(Optional)*	Bool	Indicates whether or not to enable this CSV pipeline.
meters *(Optional)*	List	A regular expression of the meters piped to CSV.
location *(Optional)*	String	The full path to the CSV output file. The folder is restricted.
backup_count *(Optional)*	Int	The number of backed up CSV files to preserve.
max_bytes *(Optional)*	Int	The size in bytes to allow the CSV file to grow before triggering rotating and compressing.

Example 5.6. Modify pipeline: JSON response

```
{
 "name": "csv",
 "compress": false,
 "enabled": false,
 "meters": ["!vswitch.*", "!cpu_util"],
 "location": "/opt/cgcs/ceilometer/csv/pm2.csv",
 "backup_count": 4,
 "max_bytes": 9999999
}
```

This operation does not return a response body.

6. SysInv API v1

Manage physical servers with the CGCS System Inventory API. This includes inventory collection and configuration of nodes, ports, interfaces, CPUs, disks and memory. The API also supports alarm collection for fault events of the cloud itself as well as configuration of the cloud's SNMP interface.

The typical port used for the SysInv REST API is 6385. However, proper technique would be to look up the sysinv service endpoint in keystone.

Method	URI	Description
	API versions	
GET	/	Lists information about all System Inventory API versions.
GET	/v1	Shows details for System Inventory API v1.
	System	
GET	/v1/isystems	Shows attributes of the System object.
PATCH	/v1/isystems	Modifies attributes of the System object.
	Hosts	
GET	/v1/ihosts	Lists all host entities.
GET	/v1/ihosts/{host_id}	Shows detailed information about a specific host.
PATCH	/v1/ihosts/{host_id}	Modifies a specific host.
PATCH	/v1/ihosts/{host_id}	Executes an action on a specific host.
DELETE	/v1/ihosts/{host_id}	Deletes a specific host.
POST	/v1/ihosts	Creates a host.
	Ports	
GET	/v1/ihosts/{host_id}/iports	List the physical ports of a host.
GET	/v1/iports/{port_id}	Shows the attributes of a specific physical port.
	Interfaces	
GET	/v1/ihosts/{host_id}/iinterfaces	List the L2 interfaces of a specific host.
GET	/v1/iinterfaces/{interface_id}	Shows information about a specific L2 interface.
POST	/v1/ihosts/{host_id}/iinterfaces	Creates an L2 interface on a specific host.
PATCH	/v1/iinterfaces/{interface_id}	Modifies a specific L2 interface.
DELETE	/v1/iinterfaces/{interface_id}	Deletes a specific L2 interface.
	CPUs	
GET	/v1/ihosts/{host_id}/icpus	Lists all cpus (logical processor cores) of a host.
GET	/v1/icpus/{cpu_id}	Shows information about a specific cpu (logical processor core).
PATCH	/v1/icpus/{cpu_id}	Modifies the number of cores assigned to different functions on a host.
	Memory	
GET	/v1/ihosts/{host_id}/imemorys	Lists the memory information of all NUMA nodes of a host.
GET	/v1/imemorys/{memory_id}	Shows the memory information about a specific NUMA node of a specific host.
PATCH	/v1/imemorys/{memory_id}	Modifies the memory information about a specific NUMA node of a specific host..
	Disks	
GET	/v1/ihosts/{host_id}/idisks	Lists all physical disks of a host.

Method	URI	Description
GET	`/v1/idisks/{disk_id}`	Shows detailed information about a specific physical disk.
	Volumes	
GET	`/v1/ihosts/{host_id}/istors`	Lists all storage disk volumes of a host.
GET	`/v1/istors/{volume_id}`	Shows detailed information about a specific storage disk volume.
POST	`/v1/ihosts/{host_id}/istors`	Creates a storage disk volume on a specific host.
	Volume Groups	
GET	`/v1/ihosts/{host_id}/ilvgs`	Lists all LVM volume groups of a host.
GET	`/v1/ilvgs/{volumegroup_id}`	Shows detailed information about a specific LVM volume group.
POST	`/v1/ilvgs`	Creates an LVM volume group on a specific host.
DELETE	`/v1/ilvgs/{volumegroup_id}`	Deletes a specific LVM volume group.
	Physical Volumes	
GET	`/v1/ihosts/{host_id}/ipvs`	Lists all LVM physical volumes of a host.
GET	`/v1/ipvs/{physicalvolume_id}`	Shows detailed information about a specific LVM physical volume.
POST	`/v1/ipvs`	Creates an LVM physical volume on a specific host.
DELETE	`/v1/ipvs/{physicalvolume_id}`	Deletes a specific LVM physical volume.
	Profiles	
GET	`/v1/iprofiles`	Lists all profiles.
GET	`/v1/iprofiles/{profile_id}`	Shows information about a specific profile.
POST	`/v1/iprofiles`	Creates a profile.
DELETE	`/v1/iprofiles/{profile_id}`	Deletes a specific profile.
	DNS	
GET	`/v1/idns`	Shows attributes of the DNS object.
PATCH	`/v1/idns/{dns_id}`	Modifies attributes of the DNS object.
	NTP	
GET	`/v1/intp`	Shows attributes of the NTP object.
PATCH	`/v1/intp/{ntp_id}`	Modifies attributes of the NTP object.
	External OAM	
GET	`/v1/iextoam`	Shows attributes of the External OAM object.
PATCH	`/v1/iextoam/{extoam_id}`	Modifies attributes of the External OAM object.
	Infrastructure Subnet	
GET	`/v1/iinfra`	Shows attributes of the infrastructure network IP subnet object.
PATCH	`/v1/iinfra/{infra_id}`	Modifies attributes of the infrastructure network IP subnet object.
	PM	
GET	`/v1/ipm`	Shows attributes of the PM object.
PATCH	`/v1/ipm/{pm_id}`	Modifies attributes of the PM object.
	Alarms	
GET	`/v1/ialarms`	Lists all active alarms based on specified query.
GET	`/v1/ialarms/{alarm_uuid}`	Shows information about a specific alarm.
DELETE	`/v1/ialarms/{alarm_uuid}`	Deletes a specific alarm.
	Alarm History	

Method	URI	Description
GET	`/v1/ialarm_history{?q(Optional), limit(Optional)}`	Lists all alarm history based on specified query. The alarms are returned in reverse chronological order.
GET	`/v1/ialarm_history/{alarm_uuid}`	Shows information about a specific alarm history.
Customer Log		
GET	`/v1/icustomer_log{?q(Optional), limit(Optional)}`	Lists all customer logs based on specified query. The logs are returned in reverse chronological order.
GET	`/v1/icustomer_log/{log_uuid}`	Shows information about a specific customer log.
SNMP Communities		
GET	`/v1/icommunity`	Lists all SNMP Communities.
GET	`/v1/icommunity/{community_id}`	Shows information about a specific SNMP Community.
POST	`/v1/icommunity`	Creates a SNMP Community.
PATCH	`/v1/icommunity/{community_id}`	Modifies a specific SNMP Community.
DELETE	`/v1/icommunity/{community_id}`	Deletes a specific SNMP Community.
SNMP Trap Destinations		
GET	`/v1/itrapdest`	Lists all SNMP Trap Destinations.
GET	`/v1/itrapdest/{trapdest_id}`	Shows information about a specific SNMP Trap Destination.
POST	`/v1/itrapdest`	Creates a SNMP Trap Destination.
PATCH	`/v1/itrapdest/{trapdest_id}`	Modifies a specific SNMP Trap Destination.
DELETE	`/v1/itrapdest/{trapdest_id}`	Deletes a specific SNMP Trap Destination.

6.1. API versions

Method	URI	Description
GET	`/`	Lists information about all System Inventory API versions.
GET	`/v1`	Shows details for System Inventory API v1.

6.1.1. List API versions

Method	URI	Description
GET	/	Lists information about all System Inventory API versions.

Normal response codes: 200, 300

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404)

6.1.1.1. Request

This operation does not accept a request body.

6.1.1.2. Response

Example 6.1. List API versions: JSON response

```
{
   "default_version":{
      "id":"v1",
      "links":[
         {
            "href":"http://128.224.150.54:6385/v1/",
            "rel":"self"
         }
      ]
   },
   "versions":[
      {
         "id":"v1",
         "links":[
            {
               "href":"http://128.224.150.54:6385/v1/",
               "rel":"self"
            }
         ]
      }
   ],
   "description":"HP Helion OpenStack Carrier Grade API allows for the
management of physical servers.  This includes inventory collection and
configuration of hosts, ports, interfaces, CPUs, disk, memory, and system
configuration.  The API also supports alarms and fault collection for the
cloud itself as well as the configuration of the cloud's SNMP interface. ",
   "name":"Titanium SysInv API"
}
```

6.1.2. Show API v1 details

Method	URI	Description
GET	`/v1`	Shows details for System Inventory API v1.

Normal response codes: 200, 203

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

6.1.2.1. Request

This operation does not accept a request body.

6.1.2.2. Response

Example 6.2. Show API v1 details: JSON response

```
{
   "ihosts":[
      {
         "href":"http://128.224.150.54:6385/v1/ihosts/",
         "rel":"self"
      },
      {
         "href":"http://128.224.150.54:6385/ihosts/",
         "rel":"bookmark"
      }
   ],
   "media_types":[
      {
         "base":"application/json",
         "type":"application/vnd.openstack.sysinv.v1+json"
      }
   ],
   "iconfig":[
      {
         "href":"http://128.224.150.54:6385/v1/iconfig/",
         "rel":"self"
      },
      {
         "href":"http://128.224.150.54:6385/iconfig/",
         "rel":"bookmark"
      }
   ],
   "links":[
      {
         "href":"http://128.224.150.54:6385/v1/",
         "rel":"self"
      },
      {
         "href":"http://www.windriver.com/developer/sysinv/dev/api-spec-v1.
html",
         "type":"text/html",
         "rel":"describedby"
      }
```

```
        ],
        "inode":[
            {
                "href":"http://128.224.150.54:6385/v1/inode/",
                "rel":"self"
            },
            {
                "href":"http://128.224.150.54:6385/inode/",
                "rel":"bookmark"
            }
        ],
        "imemory":[
            {
                "href":"http://128.224.150.54:6385/v1/imemory/",
                "rel":"self"
            },
            {
                "href":"http://128.224.150.54:6385/imemory/",
                "rel":"bookmark"
            }
        ],
        "idns":[
            {
                "href":"http://128.224.150.54:6385/v1/idns/",
                "rel":"self"
            },
            {
                "href":"http://128.224.150.54:6385/idns/",
                "rel":"bookmark"
            }
        ],
        "iuser":[
            {
                "href":"http://128.224.150.54:6385/v1/iuser/",
                "rel":"self"
            },
            {
                "href":"http://128.224.150.54:6385/iuser/",
                "rel":"bookmark"
            }
        ],
        "itrapdest":[
            {
                "href":"http://128.224.150.54:6385/v1/itrapdest/",
                "rel":"self"
            },
            {
                "href":"http://128.224.150.54:6385/itrapdest/",
                "rel":"bookmark"
            }
        ],
        "istorconfig":[
            {
                "href":"http://128.224.150.54:6385/v1/istorconfig/",
                "rel":"self"
            },
            {
                "href":"http://128.224.150.54:6385/istorconfig/",
                "rel":"bookmark"
            }
```

```
        ],
        "iextoam":[
            {
                "href":"http://128.224.150.54:6385/v1/iextoam/",
                "rel":"self"
            },
            {
                "href":"http://128.224.150.54:6385/iextoam/",
                "rel":"bookmark"
            }
        ],
        "intp":[
            {
                "href":"http://128.224.150.54:6385/v1/intp/",
                "rel":"self"
            },
            {
                "href":"http://128.224.150.54:6385/intp/",
                "rel":"bookmark"
            }
        ],
        "isystems":[
            {
                "href":"http://128.224.150.54:6385/v1/isystems/",
                "rel":"self"
            },
            {
                "href":"http://128.224.150.54:6385/isystems/",
                "rel":"bookmark"
            }
        ],
        "iprofile":[
            {
                "href":"http://128.224.150.54:6385/v1/iprofile/",
                "rel":"self"
            },
            {
                "href":"http://128.224.150.54:6385/iprofile/",
                "rel":"bookmark"
            }
        ],
        "icpu":[
            {
                "href":"http://128.224.150.54:6385/v1/icpu/",
                "rel":"self"
            },
            {
                "href":"http://128.224.150.54:6385/icpu/",
                "rel":"bookmark"
            }
        ],
        "ialarms":[
            {
                "href":"http://128.224.150.54:6385/v1/ialarms/",
                "rel":"self"
            },
            {
                "href":"http://128.224.150.54:6385/ialarms/",
                "rel":"bookmark"
            }
```

```
    ],
    "ialarm_history":[
        {
            "href":"http://128.224.150.54:6385/v1/ialarm_history/",
            "rel":"self"
        },
        {
            "href":"http://128.224.150.54:6385/ialarm_history/",
            "rel":"bookmark"
        }
    ],
    "icommunity":[
        {
            "href":"http://128.224.150.54:6385/v1/icommunity/",
            "rel":"self"
        },
        {
            "href":"http://128.224.150.54:6385/icommunity/",
            "rel":"bookmark"
        }
    ],
    "iinfra":[
        {
            "href":"http://128.224.150.54:6385/v1/iinfra/",
            "rel":"self"
        },
        {
            "href":"http://128.224.150.54:6385/iinfra/",
            "rel":"bookmark"
        }
    ],
    "id":"v1",
}
```

6.2. System

The cloud server cluster is represented internally by a unique object referred to as the system.

Method	URI	Description
GET	/v1/isystems	Shows attributes of the System object.
PATCH	/v1/isystems	Modifies attributes of the System object.

6.2.1. Show system

Method	URI	Description
GET	`/v1/isystems`	Shows attributes of the System object.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

6.2.1.1. Request

This operation does not accept a request body.

6.2.1.2. Response

This table shows the body parameters for the show system response:

Name	Type	Description
`name`	String (Optional)	A user-specified name of the cloud system. The default value is the system UUID.
`description`	String (Optional)	A user-specified description of the cloud system.
`location`	String (Optional)	The user-specified location of the cloud system.
`contact`	String (Optional)	The user-specified contact for the cloud system.
`software_version`	String (Optional)	Contains the Cloud Server Software Version and the Software Version of the underlying Linux Kernel.
`uuid`	UUID (Optional)	The universally unique identifier for this object.
`links`	List (Optional)	For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The following types of link relations are associated with resources: a self link containing a versioned link to the resource, and a bookmark link containing a permanent link to a resource that is appropriate for long term storage.
`created_at`	DateTime (Optional)	The time when the object was created.
`updated_at`	DateTime (Optional)	The time when the object was last updated.

Example 6.3. Show system: JSON response

```
{
  "isystems": [
```

```
      {
        "uuid": "67e5fca6-3580-4b06-acc8-3200dce794a4",
        "software_version": "Golden Gate 14.08 3.4.103-ovp-ga2-rt120-WR5.0.1.
17_standard ",
        "name": "OTTAWA_LAB",
        "links": [
          {
            "href": "http://192.168.204.2:6385/v1/isystems/67e5fca6-3580-4b06-
acc8-3200dce794a4",
            "rel": "self"
          },
          {
            "href": "http://192.168.204.2:6385/isystems/67e5fca6-3580-4b06-
acc8-3200dce794a4",
            "rel": "bookmark"
          }
        ],
        "created_at": "2014-09-17T19:08:11.325946+00:00",
        "updated_at": "2014-09-24T14:35:38.091392+00:00",
        "contact": null,
        "location": null,
        "description": "The Ottawa Cloud Test Lab."
      }
  ]
}
```

This operation does not return a response body.

6.2.2. Modify system

Method	URI	Description
PATCH	`/v1/isystems`	Modifies attributes of the System object.

The attributes of the System object that are modifiable are:

- name,

- description,

- location,

- contact.

Normal response codes: 200

Error response codes: badMediaType (415)

6.2.2.1. Request

Example 6.4. Modify system: JSON request

```
[
  {
    "path": "/name",
    "value": "OTTAWA_LAB_WEST",
    "op": "replace"
  }
  {
    "path": "/description",
    "value": "The Ottawa Cloud Test Lab - West Wing.",
    "op": "replace"
  }
  {
    "path": "/location",
    "value": "350 Terry Fox Dr, Kanata, Ontario, Canada",
    "op": "replace"
  }
  {
    "path": "/contact",
    "value": "support@windriver.com",
    "op": "replace"
  }
]
```

This operation does not accept a request body.

6.2.2.2. Response

This table shows the body parameters for the modify system response:

Name	Type	Description
`ihosts`	List *(Optional)*	Links for retreiving the list of hosts for this system.
`name`	String	A user-specified name of the cloud system. The default value is the system UUID.

Name	Type	Description
	(Optional)	
description	String *(Optional)*	A user-specified description of the cloud system.
location	String *(Optional)*	The user-specified location of the cloud system.
contact	String *(Optional)*	The user-specified contact for the cloud system.
software_version	String *(Optional)*	Contains the Cloud Server Software Version and the Software Version of the underlying Linux Kernel.
uuid	UUID *(Optional)*	The universally unique identifier for this object.
links	List *(Optional)*	For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The following types of link relations are associated with resources: a self link containing a versioned link to the resource, and a bookmark link containing a permanent link to a resource that is appropriate for long term storage.
created_at	DateTime *(Optional)*	The time when the object was created.
updated_at	DateTime *(Optional)*	The time when the object was last updated.

Example 6.5. Modify system: JSON response

```
{
  "ihosts": [
    {
      "href": "http://192.168.204.2:6385/v1/isystems/67e5fca6-3580-4b06-
acc8-3200dce794a4/ihosts",
      "rel": "self"
    },
    {
      "href": "http://192.168.204.2:6385/isystems/67e5fca6-3580-4b06-
acc8-3200dce794a4/ihosts",
      "rel": "bookmark"
    }
  ],
  "uuid": "67e5fca6-3580-4b06-acc8-3200dce794a4",
  "software_version": "Golden Gate 14.08 3.4.103-ovp-ga2-rt120-WR5.0.1.
17_standard ",
  "description": "The Ottawa Cloud Test Lab - West Wing."
  "links": [
    {
      "href": "http://192.168.204.2:6385/v1/isystems/67e5fca6-3580-4b06-
acc8-3200dce794a4",
      "rel": "self"
    },
    {
      "href": "http://192.168.204.2:6385/isystems/67e5fca6-3580-4b06-
acc8-3200dce794a4",
      "rel": "bookmark"
    }
```

```
  ],
  "created_at": "2014-09-17T19:08:11.325946+00:00",
  "updated_at": "2014-09-24T14:35:38.091392+00:00",
  "contact": "support@windriver.com",
  "location": "350 Terry Fox Dr, Kanata, Ontario, Canada",
  "services": 72,
  "name": "OTTAWA_LAB_WEST",
}
```

This operation does not return a response body.

6.3. Hosts

Hosts are the physical hosts or servers for the system.

Method	URI	Description
GET	/v1/ihosts	Lists all host entities.
GET	/v1/ihosts/{host_id}	Shows detailed information about a specific host.
PATCH	/v1/ihosts/{host_id}	Modifies a specific host.
PATCH	/v1/ihosts/{host_id}	Executes an action on a specific host.
DELETE	/v1/ihosts/{host_id}	Deletes a specific host.
POST	/v1/ihosts	Creates a host.

6.3.1. List hosts

Method	URI	Description
GET	`/v1/ihosts`	Lists all host entities.

Insert extra description here, if required.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404)

6.3.1.1. Request

This operation does not accept a request body.

6.3.1.2. Response

This table shows the body parameters for the list hosts response:

Name	Type	Description
`ihosts`	List *(Optional)*	The list of host entities.
`hostname`	String *(Optional)*	The name provisioned for the host.
`personality`	String *(Optional)*	The role of the host: `controller`, `compute` or `storage`.
`administrative`	String *(Optional)*	The administrative state of the host; `unlocked` or `locked`.
`operational`	String *(Optional)*	The operational state of the host; `enabled` or `disabled`.
`availability`	String *(Optional)*	The availability state of the host; `offline`, `online`, `intest`, `available`, `degraded` or `failed`.
`mgmt_mac`	String *(Optional)*	The management MAC of the host management interface.
`mgmt_ip`	String *(Optional)*	The management IP Address of the host.
`task`	String *(Optional)*	The current maintenance task in progress on the host.
`serialid`	String *(Optional)*	The serial id configured for the host.
`bm_type`	String *(Optional)*	The board management type of the host. Currently only HPs Integrated Lights Out (iLO) board management controllers are supported by HP Helion OpenStack Carrier Grade.
`bm_username`	String	The board management username of the host.

Name	Type	Description
	(Optional)	
bm_mac	String (Optional)	The board management MAC Address of the host.
bm_ip	String (Optional)	The board management IP Address of the host.
iconfig_applied	UUID (Optional)	The configuration UUID applied to the host.
iconfig_fini	UUID (Optional)	The finished configuration UUID of the system.
iconfig_target	UUID (Optional)	The configuration target UUID of the host.
cstatus	String (Optional)	The configuration status of the host.
uptime	String (Optional)	The uptime in seconds of the host.
location	String (Optional)	The location information of the host.
subfunctions	String (Optional)	The list of roles supported by the host. Comma separated string. If the role is equal to personality, no additional subfunctions are configured.
subfunction_oper	String (Optional)	The subfunction operational state, excluding the primary role personality.
subfunction_avail	String (Optional)	The subfunction availability state, excluding the primary role personality.
recordtype	String (Optional)	The recordtype of the host: `standard` or `profile`.
id	String (Optional)	Id value of the host.
ihost_action	String (Optional)	Action on the host in progress.
vim_progress_status	String (Optional)	virtual infrastructure manager progress status.

Example 6.6. List hosts: JSON response

```
{
    "ihosts":[
        {
            "reserved":"False",
            "links":[
                {
                    "href":"http://192.168.204.2:6385/v1/ihosts/298d0050-7758-4bb8-
aefc-dfddad2c4984",
                    "rel":"self"
                },
```

```
            {
                "href":"http://192.168.204.2:6385/ihosts/298d0050-7758-4bb8-
aefc-dfddad2c4984",
                "rel":"bookmark"
            }
        ],
        "bm_ip":"",
        "updated_at":"2014-10-02T14:56:23.230316+00:00",
        "bm_username":null,
        "iprofile_uuid":null,
        "id":1,
        "uptime":68379,
        "iconfig_fini":null,
        "mgmt_ip":"192.168.204.3",
        "hostname":"controller-0",
        "capabilities":{
            "stor_function":"monitor",
            "Personality":"Controller-Active"
        },
        "operational":"enabled",
        "availability":"available",
        "location":{

        },
        "iconfig_applied":"18c9e850-be49-4b84-9eba-6aaeab12ec72",
        "administrative":"unlocked",
        "personality":"controller",
        "bm_mac":null,
        "cstatus":"Config out-of-date",
        "iconfig_target":"a47cfb0d-3892-4608-8012-371ce45faf55",
        "mgmt_mac":"08:00:27:3d:c2:fe",
        "task":"",
        "created_at":"2014-10-01T20:06:44.302456+00:00",
        "uuid":"298d0050-7758-4bb8-aefc-dfddad2c4984",
        "action":"none",
        "bm_type":null
    },
    {
        "reserved":"False",
        "links":[
            {
                "href":"http://192.168.204.2:6385/v1/ihosts/5f7d15c6-77aa-49cd-
a6a1-678aef89edea",
                "rel":"self"
            },
            {
                "href":"http://192.168.204.2:6385/ihosts/5f7d15c6-77aa-49cd-
a6a1-678aef89edea",
                "rel":"bookmark"
            }
        ],
        "bm_ip":"",
        "updated_at":"2014-10-02T14:56:23.252439+00:00",
        "bm_username":"",
        "iprofile_uuid":null,
        "id":2,
        "uptime":65518,
        "iconfig_fini":null,
        "mgmt_ip":"192.168.204.4",
        "hostname":"controller-1",
```

```
         "capabilities":{
            "stor_function":"monitor",
            "Personality":"Controller-Standby"
         },
         "operational":"enabled",
         "availability":"available",
         "location":{
            "locn":""
         },
         "iconfig_applied":"18c9e850-be49-4b84-9eba-6aaeab12ec72",
         "administrative":"unlocked",
         "personality":"controller",
         "bm_mac":"",
         "cstatus":"Config out-of-date",
         "iconfig_target":"a47cfb0d-3892-4608-8012-371ce45faf55",
         "mgmt_mac":"08:00:27:90:be:dc",
         "task":"",
         "created_at":"2014-10-01T20:07:11.401964+00:00",
         "uuid":"5f7d15c6-77aa-49cd-a6a1-678aef89edea",
         "action":"none",
         "bm_type":null
      },
      {
         "reserved":"False",
         "links":[
            {
               "href":"http://192.168.204.2:6385/v1/ihosts/0dad0322-
f289-40ca-9059-67cd673a0923",
               "rel":"self"
            },
            {
               "href":"http://192.168.204.2:6385/ihosts/0dad0322-
f289-40ca-9059-67cd673a0923",
               "rel":"bookmark"
            }
         ],
         "bm_ip":"",
         "updated_at":"2014-10-02T15:00:23.445512+00:00",
         "bm_username":"",
         "iprofile_uuid":null,
         "id":5,
         "uptime":63720,
         "iconfig_fini":null,
         "mgmt_ip":"192.168.204.5",
         "hostname":"storage-0",
         "capabilities":{
            "stor_function":"monitor"
         },
         "operational":"disabled",
         "availability":"online",
         "location":{
            "locn":""
         },
         "iconfig_applied":null,
         "administrative":"locked",
         "personality":"storage",
         "bm_mac":"",
         "cstatus":null,
         "iconfig_target":null,
         "mgmt_mac":"08:00:27:fa:e2:1c",
```

```
              "task":"",
              "created_at":"2014-10-01T21:12:09.899675+00:00",
              "uuid":"0dad0322-f289-40ca-9059-67cd673a0923",
              "action":"none",
              "bm_type":null
          },
          {
              "reserved":"False",
              "links":[
                  {
                      "href":"http://192.168.204.2:6385/v1/ihosts/42d72247-
e0e3-4a5a-8cb1-40bbee52c8db",
                      "rel":"self"
                  },
                  {
                      "href":"http://192.168.204.2:6385/ihosts/42d72247-
e0e3-4a5a-8cb1-40bbee52c8db",
                      "rel":"bookmark"
                  }
              ],
              "bm_ip":"",
              "updated_at":"2014-10-02T14:56:23.268242+00:00",
              "bm_username":"",
              "iprofile_uuid":null,
              "id":6,
              "uptime":62651,
              "iconfig_fini":null,
              "mgmt_ip":"192.168.204.6",
              "hostname":"storage-1",
              "capabilities":{

              },
              "operational":"disabled",
              "availability":"online",
              "location":{
                  "locn":""
              },
              "iconfig_applied":null,
              "administrative":"locked",
              "personality":"storage",
              "bm_mac":"",
              "cstatus":null,
              "iconfig_target":null,
              "mgmt_mac":"08:00:27:22:48:f2",
              "task":"",
              "created_at":"2014-10-01T21:26:17.404218+00:00",
              "uuid":"42d72247-e0e3-4a5a-8cb1-40bbee52c8db",
              "action":"none",
              "bm_type":null
          },
          {
              "reserved":"False",
              "links":[
                  {
                      "href":"http://192.168.204.2:6385/v1/ihosts/
cd5ef327-618b-4aac-9b10-9bbbe2baa8e0",
                      "rel":"self"
                  },
                  {
```

```
                "href":"http://192.168.204.2:6385/ihosts/
cd5ef327-618b-4aac-9b10-9bbbe2baa8e0",
                "rel":"bookmark"
          }
        ],
        "bm_ip":null,
        "updated_at":null,
        "bm_username":null,
        "iprofile_uuid":null,
        "id":7,
        "uptime":0,
        "iconfig_fini":null,
        "mgmt_ip":"192.168.204.129",
        "hostname":null,
        "capabilities":{

        },
        "operational":"disabled",
        "availability":"offline",
        "location":{

        },
        "iconfig_applied":null,
        "administrative":"locked",
        "personality":null,
        "bm_mac":null,
        "cstatus":null,
        "iconfig_target":null,
        "mgmt_mac":"08:00:27:be:6e:25",
        "task":null,
        "created_at":"2014-10-02T13:57:04.900900+00:00",
        "uuid":"cd5ef327-618b-4aac-9b10-9bbbe2baa8e0",
        "action":"none",
        "bm_type":null
      }
   ]
}
```

This operation does not return a response body.

6.3.2. Show host

Method	URI	Description
GET	`/v1/ihosts/{host_id}`	Shows detailed information about a specific host.

Insert extra description here, if required.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

6.3.2.1. Request

This table shows the URI parameters for the show host request:

Name	Type	Description
`{host_id}`	UUID	The unique identifier of an existing host.

This operation does not accept a request body.

6.3.2.2. Response

This table shows the body parameters for the show host response:

Name	Type	Description
`invProvisioned`	String *(Optional)*	Indicates whether the host has the minimum level of provisioning or not. Only a `provisioned` host can be unlocked.
`hostname`	String *(Optional)*	The name provisioned for the host.
`personality`	String *(Optional)*	The role of the host: `controller`, `compute` or `storage`.
`administrative`	String *(Optional)*	The administrative state of the host; `unlocked` or `locked`.
`operational`	String *(Optional)*	The operational state of the host; `enabled` or `disabled`.
`availability`	String *(Optional)*	The availability state of the host; `offline`, `online`, `intest`, `available`, `degraded` or `failed`.
`mgmt_mac`	String *(Optional)*	The management MAC of the host management interface.
`mgmt_ip`	String *(Optional)*	The management IP Address of the host.
`task`	String	The current maintenance task in progress on the host.

Name	Type	Description
	(Optional)	
serialid	String *(Optional)*	The serial id configured for the host.
bm_type	String *(Optional)*	The board management type of the host. Currently only HPs Integrated Lights Out (iLO) board management controllers are supported by HP Helion OpenStack Carrier Grade.
bm_username	String *(Optional)*	The board management username of the host.
bm_mac	String *(Optional)*	The board management MAC Address of the host.
bm_ip	String *(Optional)*	The board management IP Address of the host.
iconfig_applied	UUID *(Optional)*	The configuration UUID applied to the host.
iconfig_fini	UUID *(Optional)*	The finished configuration UUID of the system.
iconfig_target	UUID *(Optional)*	The configuration target UUID of the host.
cstatus	String *(Optional)*	The configuration status of the host.
uptime	String *(Optional)*	The uptime in seconds of the host.
location	String *(Optional)*	The location information of the host.
subfunctions	String *(Optional)*	The list of roles supported by the host. Comma separated string. If the role is equal to personality, no additional subfunctions are configured.
subfunction_oper	String *(Optional)*	The subfunction operational state, excluding the primary role personality.
subfunction_avail	String *(Optional)*	The subfunction availability state, excluding the primary role personality.
recordtype	String *(Optional)*	The recordtype of the host: `standard` or `profile`.
id	String *(Optional)*	Id value of the host.
ihost_action	String *(Optional)*	Action on the host in progress.
vim_progress_status	String *(Optional)*	virtual infrastructure manager progress status.

Example 6.7. Show host: JSON response

```
{
```

```
    "iports":[
        {
            "href":"http://192.168.204.2:6385/v1/ihosts/298d0050-7758-4bb8-aefc-
dfddad2c4984/iports",
            "rel":"self"
        },
        {
            "href":"http://192.168.204.2:6385/ihosts/298d0050-7758-4bb8-aefc-
dfddad2c4984/iports",
            "rel":"bookmark"
        }
    ],
    "reserved":"False",
    "iconfig_fini":null,
    "idisks":[
        {
            "href":"http://192.168.204.2:6385/v1/ihosts/298d0050-7758-4bb8-aefc-
dfddad2c4984/idisks",
            "rel":"self"
        },
        {
            "href":"http://192.168.204.2:6385/ihosts/298d0050-7758-4bb8-aefc-
dfddad2c4984/idisks",
            "rel":"bookmark"
        }
    ],
    "bm_ip":"",
    "updated_at":"2014-10-02T15:14:23.744473+00:00",
    "bm_username":null,
    "iprofile_uuid":null,
    "id":1,
    "forisystemid":1,
    "icpus":[
        {
            "href":"http://192.168.204.2:6385/v1/ihosts/298d0050-7758-4bb8-aefc-
dfddad2c4984/icpus",
            "rel":"self"
        },
        {
            "href":"http://192.168.204.2:6385/ihosts/298d0050-7758-4bb8-aefc-
dfddad2c4984/icpus",
            "rel":"bookmark"
        }
    ],
    "uptime":69459,
    "links":[
        {
            "href":"http://192.168.204.2:6385/v1/ihosts/298d0050-7758-4bb8-aefc-
dfddad2c4984",
            "rel":"self"
        },
        {
            "href":"http://192.168.204.2:6385/ihosts/298d0050-7758-4bb8-aefc-
dfddad2c4984",
            "rel":"bookmark"
        }
    ],
    "mgmt_ip":"192.168.204.3",
    "hostname":"controller-0",
    "istors":[
```

```
            {
                "href":"http://192.168.204.2:6385/v1/ihosts/298d0050-7758-4bb8-aefc-
        dfddad2c4984/istors",
                "rel":"self"
            },
            {
                "href":"http://192.168.204.2:6385/ihosts/298d0050-7758-4bb8-aefc-
        dfddad2c4984/istors",
                "rel":"bookmark"
            }
        ],
        "capabilities":{
            "stor_function":"monitor",
            "Personality":"Controller-Active"
        },
        "availability":"available",
        "location":{

        },
        "iconfig_applied":"18c9e850-be49-4b84-9eba-6aaeab12ec72",
        "invprovision":"provisioned",
        "administrative":"unlocked",
        "personality":"controller",
        "iinterfaces":[
            {
                "href":"http://192.168.204.2:6385/v1/ihosts/298d0050-7758-4bb8-aefc-
        dfddad2c4984/iinterfaces",
                "rel":"self"
            },
            {
                "href":"http://192.168.204.2:6385/ihosts/298d0050-7758-4bb8-aefc-
        dfddad2c4984/iinterfaces",
                "rel":"bookmark"
            }
        ],
        "bm_mac":null,
        "cstatus":"Config out-of-date",
        "iconfig_target":"a47cfb0d-3892-4608-8012-371ce45faf55",
        "isystem_uuid":"e79e74a5-e08e-41ab-9277-5e01457a0e5e",
        "mgmt_mac":"08:00:27:3d:c2:fe",
        "task":"",
        "recordtype":"standard",
        "operational":"enabled",
        "created_at":"2014-10-01T20:06:44.302456+00:00",
        "uuid":"298d0050-7758-4bb8-aefc-dfddad2c4984",
        "action":"none",
        "bm_type":null,
        "serialId":null,
        "inodes":[
            {
                "href":"http://192.168.204.2:6385/v1/ihosts/298d0050-7758-4bb8-aefc-
        dfddad2c4984/inodes",
                "rel":"self"
            },
            {
                "href":"http://192.168.204.2:6385/ihosts/298d0050-7758-4bb8-aefc-
        dfddad2c4984/inodes",
                "rel":"bookmark"
            }
        ],
```

```
    "imemorys":[
        {
            "href":"http://192.168.204.2:6385/v1/ihosts/298d0050-7758-4bb8-aefc-
dfddad2c4984/imemorys",
            "rel":"self"
        },
        {
            "href":"http://192.168.204.2:6385/ihosts/298d0050-7758-4bb8-aefc-
dfddad2c4984/imemorys",
            "rel":"bookmark"
        }
    ]
}
```

This operation does not return a response body.

6.3.3. Modify host

Method	URI	Description
PATCH	`/v1/ihosts/{host_id}`	Modifies a specific host.

The atrributes of a Host which are modifiable:

- personality,

- hostname,

- bm_type,

- bm_mac,

- bm_ip,

- bm_username,

- bm_password,

- serialid,

- location.

Normal response codes: 200

Error response codes: badMediaType (415)

6.3.3.1. Request

This table shows the URI parameters for the modify host request:

Name	Type	Description
`{host_id}`	UUID	The unique identifier of an existing host.

This table shows the body parameters for the modify host request:

Name	Type	Description
`firstParameter`	String *(Optional)*	This parameter specifies ... . Valid values are: `value1`, `value2` or `value3`.
`secondParameter`	String *(Optional)*	This parameter specifies ... . Valid values are: `value1`, `value2` or `value3`.

Example 6.8. Modify host: JSON request

```
[
  {
    "path":"/location",
    "value":"{'locn':'350 Terry Fox Dr, Kanata, Ontario, Canada'}",
    "op":"replace"
  }
]
```

This operation does not accept a request body.

6.3.3.2. Response

This table shows the body parameters for the modify host response:

Name	Type	Description
hostname	String *(Optional)*	The name provisioned for the host.
personality	String *(Optional)*	The role of the host: `controller`, `compute` or `storage`.
administrative	String *(Optional)*	The administrative state of the host; `unlocked` or `locked`.
operational	String *(Optional)*	The operational state of the host; `enabled` or `disabled`.
availability	String *(Optional)*	The availability state of the host; `offline`, `online`, `intest`, `available`, `degraded` or `failed`.
mgmt_mac	String *(Optional)*	The management MAC of the host management interface.
mgmt_ip	String *(Optional)*	The management IP Address of the host.
task	String *(Optional)*	The current maintenance task in progress on the host.
serialid	String *(Optional)*	The serial id configured for the host.
bm_type	String *(Optional)*	The board management type of the host. Currently only HPs Integrated Lights Out (iLO) board management controllers are supported by HP Helion OpenStack Carrier Grade.
bm_username	String *(Optional)*	The board management username of the host.
bm_mac	String *(Optional)*	The board management MAC Address of the host.
bm_ip	String *(Optional)*	The board management IP Address of the host.
iconfig_applied	UUID *(Optional)*	The configuration UUID applied to the host.
iconfig_fini	UUID *(Optional)*	The finished configuration UUID of the system.
iconfig_target	UUID *(Optional)*	The configuration target UUID of the host.
cstatus	String *(Optional)*	The configuration status of the host.
uptime	String *(Optional)*	The uptime in seconds of the host.
location	String	The location information of the host.

Name	Type	Description
	(Optional)	
subfunctions	String *(Optional)*	The list of roles supported by the host. Comma separated string. If the role is equal to personality, no additional subfunctions are configured.
subfunction_oper	String *(Optional)*	The subfunction operational state, excluding the primary role personality.
subfunction_avail	String *(Optional)*	The subfunction availability state, excluding the primary role personality.
recordtype	String *(Optional)*	The recordtype of the host: standard or profile.
id	String *(Optional)*	Id value of the host.
ihost_action	String *(Optional)*	Action on the host in progress.
vim_progress_status	String *(Optional)*	virtual infrastructure manager progress status.

Example 6.9. Modify host: JSON response

```
{
   "iports":[
      {
         "href":"http://192.168.204.2:6385/v1/ihosts/298d0050-7758-4bb8-aefc-
dfddad2c4984/iports",
         "rel":"self"
      },
      {
         "href":"http://192.168.204.2:6385/ihosts/298d0050-7758-4bb8-aefc-
dfddad2c4984/iports",
         "rel":"bookmark"
      }
   ],
   "reserved":"False",
   "iconfig_fini":null,
   "idisks":[
      {
         "href":"http://192.168.204.2:6385/v1/ihosts/298d0050-7758-4bb8-aefc-
dfddad2c4984/idisks",
         "rel":"self"
      },
      {
         "href":"http://192.168.204.2:6385/ihosts/298d0050-7758-4bb8-aefc-
dfddad2c4984/idisks",
         "rel":"bookmark"
      }
   ],
   "bm_ip":"",
   "updated_at":"2014-10-02T15:19:42.572251+00:00",
   "bm_username":null,
   "iprofile_uuid":null,
   "id":1,
   "forisystemid":1,
   "icpus":[
```

```
    {
        "href":"http://192.168.204.2:6385/v1/ihosts/298d0050-7758-4bb8-aefc-
dfddad2c4984/icpus",
        "rel":"self"
    },
    {
        "href":"http://192.168.204.2:6385/ihosts/298d0050-7758-4bb8-aefc-
dfddad2c4984/icpus",
        "rel":"bookmark"
    }
],
"uptime":69459,
"links":[
    {
        "href":"http://192.168.204.2:6385/v1/ihosts/298d0050-7758-4bb8-aefc-
dfddad2c4984",
        "rel":"self"
    },
    {
        "href":"http://192.168.204.2:6385/ihosts/298d0050-7758-4bb8-aefc-
dfddad2c4984",
        "rel":"bookmark"
    }
],
"mgmt_ip":"192.168.204.3",
"hostname":"controller-0",
"istors":[
    {
        "href":"http://192.168.204.2:6385/v1/ihosts/298d0050-7758-4bb8-aefc-
dfddad2c4984/istors",
        "rel":"self"
    },
    {
        "href":"http://192.168.204.2:6385/ihosts/298d0050-7758-4bb8-aefc-
dfddad2c4984/istors",
        "rel":"bookmark"
    }
],
"capabilities":{
    "stor_function":"monitor"
},
"availability":"available",
"location":{
    "locn":"350 Terry Fox Dr, Kanata, Ontario, Canada"
},
"iconfig_applied":"18c9e850-be49-4b84-9eba-6aaeab12ec72",
"invprovision":"provisioned",
"administrative":"unlocked",
"personality":"controller",
"iinterfaces":[
    {
        "href":"http://192.168.204.2:6385/v1/ihosts/298d0050-7758-4bb8-aefc-
dfddad2c4984/iinterfaces",
        "rel":"self"
    },
    {
        "href":"http://192.168.204.2:6385/ihosts/298d0050-7758-4bb8-aefc-
dfddad2c4984/iinterfaces",
        "rel":"bookmark"
    }
```

```
        ],
    "bm_mac":null,
    "cstatus":"Config out-of-date",
    "iconfig_target":"a47cfb0d-3892-4608-8012-371ce45faf55",
    "isystem_uuid":"e79e74a5-e08e-41ab-9277-5e01457a0e5e",
    "mgmt_mac":"08:00:27:3d:c2:fe",
    "task":"",
    "recordtype":"standard",
    "operational":"enabled",
    "created_at":"2014-10-01T20:06:44.302456+00:00",
    "uuid":"298d0050-7758-4bb8-aefc-dfddad2c4984",
    "action":"none",
    "bm_type":null,
    "serialId":null,
    "inodes":[
        {
            "href":"http://192.168.204.2:6385/v1/ihosts/298d0050-7758-4bb8-aefc-
dfddad2c4984/inodes",
            "rel":"self"
        },
        {
            "href":"http://192.168.204.2:6385/ihosts/298d0050-7758-4bb8-aefc-
dfddad2c4984/inodes",
            "rel":"bookmark"
        }
    ],
    "imemorys":[
        {
            "href":"http://192.168.204.2:6385/v1/ihosts/298d0050-7758-4bb8-aefc-
dfddad2c4984/imemorys",
            "rel":"self"
        },
        {
            "href":"http://192.168.204.2:6385/ihosts/298d0050-7758-4bb8-aefc-
dfddad2c4984/imemorys",
            "rel":"bookmark"
        }
    ]
}
```

This operation does not return a response body.

6.3.4. Action host

Method	URI	Description
PATCH	`/v1/ihosts/{host_id}`	Executes an action on a specific host.

Insert extra description here, if required.

Normal response codes: 200

Error response codes: badMediaType (415)

6.3.4.1. Request

This table shows the URI parameters for the action host request:

Name	Type	Description
`{host_id}`	UUID	The unique identifier of an existing host.

This table shows the body parameters for the action host request:

Name	Type	Description
`action`	String *(Optional)*	Perform one of the following actions to the host: Valid values are: `unlock`, `lock`, `swact`, `apply-profile`, `reboot`, `reset`, `power-on`, `power-off`, or `reinstall`.

Example 6.10. Action host: JSON request

```
[
    {
        "path":"/action",
        "value":"unlock",
        "op":"replace"
    }
]
```

This operation does not accept a request body.

6.3.4.2. Response

This table shows the body parameters for the action host response:

Name	Type	Description
`hostname`	String *(Optional)*	The name provisioned for the host.
`personality`	String *(Optional)*	The role of the host: `controller`, `compute` or `storage`.
`administrative`	String *(Optional)*	The administrative state of the host; `unlocked` or `locked`.
`operational`	String *(Optional)*	The operational state of the host; `enabled` or `disabled`.
`availability`	String *(Optional)*	The availability state of the host; `offline`, `online`, `intest`, `available`, `degraded` or `failed`.
`mgmt_mac`	String	The management MAC of the host management interface.

Name	Type	Description
	(Optional)	
mgmt_ip	String *(Optional)*	The management IP Address of the host.
task	String *(Optional)*	The current maintenance task in progress on the host.
serialid	String *(Optional)*	The serial id configured for the host.
bm_type	String *(Optional)*	The board management type of the host. Currently only HPs Integrated Lights Out (iLO) board management controllers are supported by HP Helion OpenStack Carrier Grade.
bm_username	String *(Optional)*	The board management username of the host.
bm_mac	String *(Optional)*	The board management MAC Address of the host.
bm_ip	String *(Optional)*	The board management IP Address of the host.
iconfig_applied	UUID *(Optional)*	The configuration UUID applied to the host.
iconfig_fini	UUID *(Optional)*	The finished configuration UUID of the system.
iconfig_target	UUID *(Optional)*	The configuration target UUID of the host.
cstatus	String *(Optional)*	The configuration status of the host.
uptime	String *(Optional)*	The uptime in seconds of the host.
location	String *(Optional)*	The location information of the host.
subfunctions	String *(Optional)*	The list of roles supported by the host. Comma separated string. If the role is equal to personality, no additional subfunctions are configured.
subfunction_oper	String *(Optional)*	The subfunction operational state, excluding the primary role personality.
subfunction_avail	String *(Optional)*	The subfunction availability state, excluding the primary role personality.
recordtype	String *(Optional)*	The recordtype of the host: `standard` or `profile`.
id	String *(Optional)*	Id value of the host.
ihost_action	String *(Optional)*	Action on the host in progress.
vim_progress_status	String	virtual infrastructure manager progress status.

Name	Type	Description
	(Optional)	

Example 6.11. Action host: JSON response

```
{
   "iports":[
      {
         "href":"http://192.168.204.2:6385/v1/ihosts/0dad0322-
f289-40ca-9059-67cd673a0923/iports",
         "rel":"self"
      },
      {
         "href":"http://192.168.204.2:6385/ihosts/0dad0322-
f289-40ca-9059-67cd673a0923/iports",
         "rel":"bookmark"
      }
   ],
   "reserved":"False",
   "iconfig_fini":null,
   "idisks":[
      {
         "href":"http://192.168.204.2:6385/v1/ihosts/0dad0322-
f289-40ca-9059-67cd673a0923/idisks",
         "rel":"self"
      },
      {
         "href":"http://192.168.204.2:6385/ihosts/0dad0322-
f289-40ca-9059-67cd673a0923/idisks",
         "rel":"bookmark"
      }
   ],
   "bm_ip":"",
   "updated_at":"2014-10-02T15:31:31.565491+00:00",
   "bm_username":"",
   "iprofile_uuid":null,
   "id":5,
   "forisystemid":1,
   "icpus":[
      {
         "href":"http://192.168.204.2:6385/v1/ihosts/0dad0322-
f289-40ca-9059-67cd673a0923/icpus",
         "rel":"self"
      },
      {
         "href":"http://192.168.204.2:6385/ihosts/0dad0322-
f289-40ca-9059-67cd673a0923/icpus",
         "rel":"bookmark"
      }
   ],
   "uptime":107,
   "links":[
      {
         "href":"http://192.168.204.2:6385/v1/ihosts/0dad0322-
f289-40ca-9059-67cd673a0923",
         "rel":"self"
      },
      {
         "href":"http://192.168.204.2:6385/ihosts/0dad0322-
f289-40ca-9059-67cd673a0923",
```

```
                "rel":"bookmark"
            }
        ],
        "mgmt_ip":"192.168.204.5",
        "hostname":"storage-0",
        "istors":[
            {
                "href":"http://192.168.204.2:6385/v1/ihosts/0dad0322-
f289-40ca-9059-67cd673a0923/istors",
                "rel":"self"
            },
            {
                "href":"http://192.168.204.2:6385/ihosts/0dad0322-
f289-40ca-9059-67cd673a0923/istors",
                "rel":"bookmark"
            }
        ],
        "capabilities":{
            "stor_function":"monitor"
        },
        "availability":"online",
        "location":{
            "locn":""
        },
        "iconfig_applied":"a47cfb0d-3892-4608-8012-371ce45faf55",
        "invprovision":"provisioned",
        "administrative":"locked",
        "personality":"storage",
        "iinterfaces":[
            {
                "href":"http://192.168.204.2:6385/v1/ihosts/0dad0322-
f289-40ca-9059-67cd673a0923/iinterfaces",
                "rel":"self"
            },
            {
                "href":"http://192.168.204.2:6385/ihosts/0dad0322-
f289-40ca-9059-67cd673a0923/iinterfaces",
                "rel":"bookmark"
            }
        ],
        "bm_mac":"",
        "cstatus":"",
        "iconfig_target":null,
        "isystem_uuid":"e79e74a5-e08e-41ab-9277-5e01457a0e5e",
        "mgmt_mac":"08:00:27:fa:e2:1c",
        "task":"Unlocking",
        "recordtype":"standard",
        "operational":"disabled",
        "created_at":"2014-10-01T21:12:09.899675+00:00",
        "uuid":"0dad0322-f289-40ca-9059-67cd673a0923",
        "action":"none",
        "bm_type":null,
        "serialId":null,
        "inodes":[
            {
                "href":"http://192.168.204.2:6385/v1/ihosts/0dad0322-
f289-40ca-9059-67cd673a0923/inodes",
                "rel":"self"
            },
            {
```

```
            "href":"http://192.168.204.2:6385/ihosts/0dad0322-
f289-40ca-9059-67cd673a0923/inodes",
            "rel":"bookmark"
        }
    ],
    "imemorys":[
        {
            "href":"http://192.168.204.2:6385/v1/ihosts/0dad0322-
f289-40ca-9059-67cd673a0923/imemorys",
            "rel":"self"
        },
        {
            "href":"http://192.168.204.2:6385/ihosts/0dad0322-
f289-40ca-9059-67cd673a0923/imemorys",
            "rel":"bookmark"
        }
    ]
}
```

This operation does not return a response body.

6.3.5. Delete host

Method	URI	Description
DELETE	`/v1/ihosts/{host_id}`	Deletes a specific host.

Insert extra description here, if required.

Normal response codes: 204

6.3.5.1. Request

This table shows the URI parameters for the delete host request:

Name	Type	Description
`{host_id}`	UUID	The unique identifier of an existing host.

This operation does not accept a request body.

6.3.5.2. Response

This operation does not return a response body.

6.3.6. Create host

Method	URI	Description
POST	`/v1/ihosts`	Creates a host.

Note that a host should only be added through the REST API if the system is not already configured to be automatically added by the system. This is determined by configuration option during config_controller at system installation.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413)

6.3.6.1. Request

This table shows the body parameters for the create host request:

Name	Type	Description
`hostname`	String *(Required)*	The hostname for the host. Must be a unique name.
`personality`	String *(Required)*	The role of of this host: i.e. `controller`, `storage`, `compute`.
`mgmt_mac`	String *(Required)*	The MAC address of the host's management interface. Must be unique.
`mgmt_ip`	String *(Required)*	The IP address of the host's management interface. Must be unique.
`bm_type`	String *(Optional)*	This attribute specifies whether board management controller type is `ilo3` or `ilo4` corresponding to the HP Integrated Lights Out (iLO). Default is None to indicate no board management controller. If bm_type is specified, then bm_mac, bm_username, and potentially, bm_ip, bm_password are also required.
`bm_mac`	String *(Optional)*	Only applicable if `bm_type` is not None. This attribute specifies the host's board management controller interface MAC address.
`bm_ip`	String *(Optional)*	Only applicable if `bm_type` is not None. This attribute specifies the host's board management controller interface IP address. `bm_ip` is not allowed to be added if the system is configured with board management (e.g. board management subnet and vlan) at installation (config_controller).
`bm_username`	String *(Optional)*	Only applicable if `bm_type` is not None. This attribute specifies the host's board management controller username.
`bm_password`	String *(Optional)*	Only applicable if `bm_type` is not None. This attribute specifies the host's board management controller password.

Example 6.12. Create host: JSON request

```
{
    "hostname":"compute-0",
    "personality":"compute"
    "mgmt_mac":"11:22:33:44:55:66",
    "mgmt_ip":"192.168.204.200",
```

```
    "bm_type":"ilo4",
    "bm_mac":"12:34:56:12:34:56",
    "bm_ip":"10.10.10.240",
    "bm_username":"bm_user",
    "bm_password":"bm_user_pwd",
}
```

This operation does not accept a request body.

6.3.6.2. Response

This table shows the body parameters for the create host response:

Name	Type	Description
hostname	String *(Optional)*	The name provisioned for the host.
personality	String *(Optional)*	The role of the host: `controller`, `compute` or `storage`.
administrative	String *(Optional)*	The administrative state of the host; `unlocked` or `locked`.
operational	String *(Optional)*	The operational state of the host; `enabled` or `disabled`.
availability	String *(Optional)*	The availability state of the host; `offline`, `online`, `intest`, `available`, `degraded` or `failed`.
mgmt_mac	String *(Optional)*	The management MAC of the host management interface.
mgmt_ip	String *(Optional)*	The management IP Address of the host.
task	String *(Optional)*	The current maintenance task in progress on the host.
serialid	String *(Optional)*	The serial id configured for the host.
bm_type	String *(Optional)*	The board management type of the host. Currently only HPs Integrated Lights Out (iLO) board management controllers are supported by HP Helion OpenStack Carrier Grade.
bm_username	String *(Optional)*	The board management username of the host.
bm_mac	String *(Optional)*	The board management MAC Address of the host.
bm_ip	String *(Optional)*	The board management IP Address of the host.
iconfig_applied	UUID *(Optional)*	The configuration UUID applied to the host.
iconfig_fini	UUID *(Optional)*	The finished configuration UUID of the system.
iconfig_target	UUID	The configuration target UUID of the host.

Name	Type	Description
	(Optional)	
cstatus	String *(Optional)*	The configuration status of the host.
uptime	String *(Optional)*	The uptime in seconds of the host.
location	String *(Optional)*	The location information of the host.
subfunctions	String *(Optional)*	The list of roles supported by the host. Comma separated string. If the role is equal to personality, no additional subfunctions are configured.
subfunction_oper	String *(Optional)*	The subfunction operational state, excluding the primary role personality.
subfunction_avail	String *(Optional)*	The subfunction availability state, excluding the primary role personality.
recordtype	String *(Optional)*	The recordtype of the host: `standard` or `profile`.
id	String *(Optional)*	Id value of the host.
ihost_action	String *(Optional)*	Action on the host in progress.
vim_progress_status	String *(Optional)*	virtual infrastructure manager progress status.
iports	String *(Optional)*	Link to the iports resources on the host.
ports	String *(Optional)*	Link to the ports resources on the host.
iinterfaces	String *(Optional)*	Link to the network interfaces resources on the host.
ethernet_ports	String *(Optional)*	Link to the ethernet ports resources on the host.
inodes	String *(Optional)*	Link to the numa node resources on the host.
imemorys	String *(Optional)*	Link to the memory resources on the host.
idisks	String *(Optional)*	Link to the disks resources on the host.
istors	String *(Optional)*	Link to the storage resources on the host.
ipvs	String *(Optional)*	Link to the physical volume storage resources on the host.
ilvgs	String	Link to the logical volume group storage resources on the host.

Name	Type	Description
	(Optional)	
`uuid`	UUID *(Optional)*	The universally unique identifier for this object.
`links`	List *(Optional)*	For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The following types of link relations are associated with resources: a self link containing a versioned link to the resource, and a bookmark link containing a permanent link to a resource that is appropriate for long term storage.
`created_at`	DateTime *(Optional)*	The time when the object was created.
`updated_at`	DateTime *(Optional)*	The time when the object was last updated.

Example 6.13. Create host: JSON response

```
{
   "iports":[
      {
         "href":        "http://192.168.204.2:6385/v1/ihosts/88d437b5-
aa2c-4f1b-8f27-d13330dca755/iports",
         "rel":"self"
      },
      {
         "href":        "http://192.168.204.2:6385/ihosts/88d437b5-
aa2c-4f1b-8f27-d13330dca755/iports",
         "rel":"bookmark"
      }
   ],
   "reserved":"False",
   "iconfig_fini":null,
   "idisks":[
      {
         "href":        "http://192.168.204.2:6385/v1/ihosts/88d437b5-
aa2c-4f1b-8f27-d13330dca755/idisks",
         "rel":"self"
      },
      {
         "href":        "http://192.168.204.2:6385/ihosts/88d437b5-
aa2c-4f1b-8f27-d13330dca755/idisks",
         "rel":"bookmark"
      }
   ],
   "subfunctions":"compute",
   "bm_ip":"10.10.10.240",
   "updated_at":null,
   "ihost_action":null,
   "bm_username":"bm_user",
   "id":3,
   "serialid":null,
   "availability":"offline",
   "forisystemid":1,
   "vim_progress_status":null,
   "icpus":[
      {
```

```
            "href":          "http://192.168.204.2:6385/v1/ihosts/88d437b5-
aa2c-4f1b-8f27-d13330dca755/icpus",
            "rel":"self"
        },
        {
            "href":          "http://192.168.204.2:6385/ihosts/88d437b5-
aa2c-4f1b-8f27-d13330dca755/icpus",
            "rel":"bookmark"
        }
    ],
    "uptime":0,
    "links":[
        {
            "href":          "http://192.168.204.2:6385/v1/ihosts/88d437b5-
aa2c-4f1b-8f27-d13330dca755",
            "rel":"self"
        },
        {
            "href":          "http://192.168.204.2:6385/ihosts/88d437b5-
aa2c-4f1b-8f27-d13330dca755",
            "rel":"bookmark"
        }
    ],
    "mgmt_ip":"192.168.204.200",
    "hostname":"compute-0",
    "istors":[
        {
            "href":          "http://192.168.204.2:6385/v1/ihosts/88d437b5-
aa2c-4f1b-8f27-d13330dca755/istors",
            "rel":"self"
        },
        {
            "href":          "http://192.168.204.2:6385/ihosts/88d437b5-
aa2c-4f1b-8f27-d13330dca755/istors",
            "rel":"bookmark"
        }
    ],
    "capabilities":{

    },
    "iprofile_uuid":null,
    "location":{

    },
    "iconfig_applied":null,
    "invprovision":null,
    "mgmt_mac":    "11:22:33:44:55:66", "administrative":"locked",
    "personality":"compute",
    "iinterfaces":[
        {
            "href":          "http://192.168.204.2:6385/v1/ihosts/88d437b5-
aa2c-4f1b-8f27-d13330dca755/iinterfaces",
            "rel":"self"
        },
        {
            "href":          "http://192.168.204.2:6385/ihosts/88d437b5-
aa2c-4f1b-8f27-d13330dca755/iinterfaces",
            "rel":"bookmark"
        }
    ],
```

```
    "isystem_uuid":"b3bbc885-2389-43e8-8b00-54a3ad6614af",
    "bm_mac":   "12:34:56:12:34:56   ", "   cstatus":null,
    "iconfig_target":null,
    "ethernet_ports":[
        {
            "href":          "http://192.168.204.2:6385/v1/ihosts/88d437b5-
aa2c-4f1b-8f27-d13330dca755/ethernet_ports",
            "rel":"self"
        },
        {
            "href":          "http://192.168.204.2:6385/ihosts/88d437b5-
aa2c-4f1b-8f27-d13330dca755/ethernet_ports",
            "rel":"bookmark"
        }
    ],
    "uuid":"88d437b5-aa2c-4f1b-8f27-d13330dca755",
    "subfunction_oper":"disabled",
    "task":null,
    "recordtype":"standard",
    "operational":"disabled",
    "created_at":   "2015-05-06T17:06:13.506319+00:00",
 "subfunction_avail":"offline",
    "ipvs":[
        {
            "href":          "http://192.168.204.2:6385/v1/ihosts/88d437b5-
aa2c-4f1b-8f27-d13330dca755/ipvs",
            "rel":"self"
        },
        {
            "href":          "http://192.168.204.2:6385/ihosts/88d437b5-
aa2c-4f1b-8f27-d13330dca755/ipvs",
            "rel":"bookmark"
        }
    ],
    "ilvgs":[
        {
            "href":          "http://192.168.204.2:6385/v1/ihosts/88d437b5-
aa2c-4f1b-8f27-d13330dca755/ilvgs",
            "rel":"self"
        },
        {
            "href":          "http://192.168.204.2:6385/ihosts/88d437b5-
aa2c-4f1b-8f27-d13330dca755/ilvgs",
            "rel":"bookmark"
        }
    ],
    "action":"none",
    "bm_type":"ilo4",
    "ports":[
        {
            "href":          "http://192.168.204.2:6385/v1/ihosts/88d437b5-
aa2c-4f1b-8f27-d13330dca755/ports",
            "rel":"self"
        },
        {
            "href":          "http://192.168.204.2:6385/ihosts/88d437b5-
aa2c-4f1b-8f27-d13330dca755/ports",
            "rel":"bookmark"
        }
    ],
```

```
    "inodes":[
        {
            "href":          "http://192.168.204.2:6385/v1/ihosts/88d437b5-
aa2c-4f1b-8f27-d13330dca755/inodes",
            "rel":"self"
        },
        {
            "href":          "http://192.168.204.2:6385/ihosts/88d437b5-
aa2c-4f1b-8f27-d13330dca755/inodes",
            "rel":"bookmark"
        }
    ],
    "imemorys":[
        {
            "href":          "http://192.168.204.2:6385/v1/ihosts/88d437b5-
aa2c-4f1b-8f27-d13330dca755/imemorys",
            "rel":"self"
        },
        {
            "href":          "http://192.168.204.2:6385/ihosts/88d437b5-
aa2c-4f1b-8f27-d13330dca755/imemorys",
            "rel":"bookmark"
        }
    ]
}
```

This operation does not return a response body.

6.4. Ports

These APIs allow the display of the physical ports of a host and their attributes.

Method	URI	Description
GET	`/v1/ihosts/{host_id}/iports`	List the physical ports of a host.
GET	`/v1/iports/{port_id}`	Shows the attributes of a specific physical port.

6.4.1. List ports

Method	URI	Description
GET	`/v1/ihosts/{host_id}/iports`	List the physical ports of a host.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

6.4.1.1. Request

This table shows the URI parameters for the list ports request:

Name	Type	Description
`{host_id}`	UUID	The unique identifier of an existing host.

This operation does not accept a request body.

6.4.1.2. Response

This table shows the body parameters for the list ports response:

Name	Type	Description
`iports`	List *(Optional)*	The list of physical ports of a host.
`pname`	String *(Optional)*	The discovered name of the port, typically the Linux assigned device name, if available.
`pnamedisplay`	String *(Optional)*	The user-specified name for the port.
`mac`	String *(Optional)*	The MAC Address of the port.
`pciaddr`	String *(Optional)*	The PCI Address of the port.
`speed`	String *(Optional)*	Currently not supported.
`autoneg`	Boolean *(Optional)*	Currently not supported.
`mtu`	Integer *(Optional)*	The Maximum Transmission Unit (MTU) of the port, in bytes.
`link_mode`	String *(Optional)*	Currently not supported.
`bootp`	Boolean *(Optional)*	Indicates whether the port can be used for network booting.
`sriov_totalvfs`	Integer	Indicates the maximum number of VFs that this port can support.

Name	Type	Description
	(Optional)	
sriov_numvfs	Integer *(Optional)*	Indicates the actual number of VFs configured for the interface using this port.
sriov_vfs_pci_address	String *(Optional)*	A comma-separated list of the PCI addresses of the configured VFs.
driver	String *(Optional)*	The driver being used for the port. Valid values are `ixgbe` and `igb`.
pclass	String *(Optional)*	The class or type of the physical IO controller device of the port.
pvendor	String *(Optional)*	The primary vendor information of the port hardware.
psvendor	String *(Optional)*	The secondary vendor information of the port hardware.
pdevice	String *(Optional)*	The primary type and model information of the port hardware.
psdevice	String *(Optional)*	The secondary type and model information of the port hardware .
iinterface_uuid	UUID *(Optional)*	The UUID of the L2 interface of the port.
numa_node	Integer *(Optional)*	The NUMA Node of the port.
inode_uuid	UUID *(Optional)*	The UUID of the NUMA node of the port.
ihost_uuid	UUID *(Optional)*	The UUID of the host containing the port.
uuid	UUID *(Optional)*	The universally unique identifier for this object.
links	List *(Optional)*	For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The following types of link relations are associated with resources: a self link containing a versioned link to the resource, and a bookmark link containing a permanent link to a resource that is appropriate for long term storage.
created_at	DateTime *(Optional)*	The time when the object was created.
updated_at	DateTime *(Optional)*	The time when the object was last updated.

Example 6.14. List ports: JSON response

```
{
  "iports": [
    {
      "links": [
```

```
        {
          "href": "http://192.168.204.2:6385/v1/iports/972dd648-bec0-4204-
a469-75c745a5994e",
          "rel": "self"
        },
        {
          "href": "http://192.168.204.2:6385/iports/972dd648-bec0-4204-
a469-75c745a5994e",
          "rel": "bookmark"
        }
      ],
      "inode_uuid": "e5d68519-eb07-4b28-8ca2-32bb476eeec1",
      "updated_at": "2014-09-18T03:12:15.389263+00:00",
      "ihost_uuid": "0aca08f9-882f-4491-8ffd-7368d20ead48",
      "autoneg": null,
      "speed": null,
      "iinterface_uuid": null,
      "uuid": "972dd648-bec0-4204-a469-75c745a5994e",
      "pdevice": "I350 Gigabit Network Connection",
      "capabilities": {

      },
      "psdevice": "Device 3592",
      "link_mode": 0,
      "bootp": null,
      "mac": "00:1e:67:51:50:01",
      "sriov_totalvfs": 63,
      "sriov_numvfs": 0,
      "sriov_vfs_pci_address": "",
      "driver": "ixgbe",
      "pname": "eth0",
      "psvendor": "Intel Corporation",
      "numa_node": 0,
      "created_at": "2014-09-18T03:12:15.334214+00:00",
      "pclass": "Ethernet controller",
      "mtu": 1500,
      "pvendor": "Intel Corporation",
      "pciaddr": "0000:0a:00.0",
      "pnamedisplay": null
    },
    {
      "links": [
        {
          "href": "http://192.168.204.2:6385/v1/iports/c822edfe-af87-4a15-
ac9b-6a8123caede1",
          "rel": "self"
        },
        {
          "href": "http://192.168.204.2:6385/iports/c822edfe-af87-4a15-
ac9b-6a8123caede1",
          "rel": "bookmark"
        }
      ],
      "inode_uuid": "e5d68519-eb07-4b28-8ca2-32bb476eeec1",
      "updated_at": "2014-09-22T02:00:43.938843+00:00",
      "ihost_uuid": "0aca08f9-882f-4491-8ffd-7368d20ead48",
      "autoneg": null,
      "speed": null,
      "iinterface_uuid": "b24caa6c-71b1-42be-8968-89abd269ea82",
      "uuid": "c822edfe-af87-4a15-ac9b-6a8123caede1",
```

```
            "pdevice": "82599EB 10-Gigabit SFI/SFP+ Network Connection",
            "capabilities": {

            },
            "psdevice": "Ethernet Server Adapter X520-2",
            "link_mode": 0,
            "bootp": null,
            "mac": "90:e2:ba:39:bb:8c",
            "sriov_totalvfs": 63,
            "sriov_numvfs": 0,
            "sriov_vfs_pci_address": "",
            "driver": "ixgbe",
            "pname": "eth4",
            "psvendor": "Intel Corporation",
            "numa_node": 0,
            "created_at": "2014-09-18T03:12:15.325296+00:00",
            "pclass": "Ethernet controller",
            "mtu": 1500,
            "pvendor": "Intel Corporation",
            "pciaddr": "0000:05:00.0",
            "pnamedisplay": null
        }
    ]
}
```

This operation does not return a response body.

6.4.2. Show port

Method	URI	Description
GET	`/v1/iports/{port_id}`	Shows the attributes of a specific physical port.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

6.4.2.1. Request

This table shows the URI parameters for the show port request:

Name	Type	Description
`{port_id}`	UUID	The unique identifier of an existing port.

This operation does not accept a request body.

6.4.2.2. Response

This table shows the body parameters for the show port response:

Name	Type	Description
`pname`	String *(Optional)*	The discovered name of the port, typically the Linux assigned device name, if available.
`pnamedisplay`	String *(Optional)*	The user-specified name for the port.
`mac`	String *(Optional)*	The MAC Address of the port.
`pciaddr`	String *(Optional)*	The PCI Address of the port.
`speed`	String *(Optional)*	Currently not supported.
`autoneg`	Boolean *(Optional)*	Currently not supported.
`mtu`	Integer *(Optional)*	The Maximum Transmission Unit (MTU) of the port, in bytes.
`link_mode`	String *(Optional)*	Currently not supported.
`bootp`	Boolean *(Optional)*	Indicates whether the port can be used for network booting.

Name	Type	Description
sriov_totalvfs	Integer *(Optional)*	Indicates the maximum number of VFs that this port can support.
sriov_numvfs	Integer *(Optional)*	Indicates the actual number of VFs configured for the interface using this port.
sriov_vfs_pci_address	String *(Optional)*	A comma-separated list of the PCI addresses of the configured VFs.
driver	String *(Optional)*	The driver being used for the port. Valid values are `ixgbe` and `igb`.
pclass	String *(Optional)*	The class or type of the physical IO controller device of the port.
pvendor	String *(Optional)*	The primary vendor information of the port hardware.
psvendor	String *(Optional)*	The secondary vendor information of the port hardware.
pdevice	String *(Optional)*	The primary type and model information of the port hardware.
psdevice	String *(Optional)*	The secondary type and model information of the port hardware .
iinterface_uuid	UUID *(Optional)*	The UUID of the L2 interface of the port.
numa_node	Integer *(Optional)*	The NUMA Node of the port.
inode_uuid	UUID *(Optional)*	The UUID of the NUMA node of the port.
ihost_uuid	UUID *(Optional)*	The UUID of the host containing the port.
uuid	UUID *(Optional)*	The universally unique identifier for this object.
links	List *(Optional)*	For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The following types of link relations are associated with resources: a self link containing a versioned link to the resource, and a bookmark link containing a permanent link to a resource that is appropriate for long term storage.
created_at	DateTime *(Optional)*	The time when the object was created.
updated_at	DateTime *(Optional)*	The time when the object was last updated.

Example 6.15. Show port: JSON response

```
{
    "pdevice" : "82599EB 10-Gigabit SFI/SFP+ Network Connection",
    "bootp" : null,
```

```
    "uuid" : "c822edfe-af87-4a15-ac9b-6a8123caede1",
    "ihost_uuid" : "0aca08f9-882f-4491-8ffd-7368d20ead48",
    "pvendor" : "Intel Corporation",
    "created_at" : "2014-09-18T03:12:15.325296+00:00",
    "speed" : null,
    "capabilities" : {},
    "mac" : "90:e2:ba:39:bb:8c",
    "sriov_totalvfs": 63,
    "sriov_numvfs": 0,
    "sriov_vfs_pci_address": "",
    "driver": "ixgbe",
    "mtu" : 1500,
    "links" : [
        {
            "rel" : "self",
            "href" : "http://128.224.151.243:6385/v1/iports/c822edfe-af87-4a15-
ac9b-6a8123caede1"
        },
        {
            "rel" : "bookmark",
            "href" : "http://128.224.151.243:6385/iports/c822edfe-af87-4a15-
ac9b-6a8123caede1"
        }
    ],
    "psvendor" : "Intel Corporation",
    "iinterface_uuid" : "b24caa6c-71b1-42be-8968-89abd269ea82",
    "numa_node" : 0,
    "pciaddr" : "0000:05:00.0",
    "pclass" : "Ethernet controller",
    "psdevice" : "Ethernet Server Adapter X520-2",
    "updated_at" : "2014-09-22T02:00:43.938843+00:00",
    "link_mode" : 0,
    "autoneg" : null,
    "pname" : "eth4",
    "pnamedisplay" : null,
    "inode_uuid" : "e5d68519-eb07-4b28-8ca2-32bb476eeec1"
}
```

This operation does not return a response body.

6.5. Interfaces

These APIs allow the create, display, modify and delete of the L2 interfaces of a host.

Method	URI	Description
GET	/v1/ihosts/{host_id}/iinterfaces	List the L2 interfaces of a specific host.
GET	/v1/iinterfaces/{interface_id}	Shows information about a specific L2 interface.
POST	/v1/ihosts/{host_id}/iinterfaces	Creates an L2 interface on a specific host.
PATCH	/v1/iinterfaces/{interface_id}	Modifies a specific L2 interface.
DELETE	/v1/iinterfaces/{interface_id}	Deletes a specific L2 interface.

6.5.1. List interfaces

Method	URI	Description
GET	`/v1/ihosts/{host_id}/iinterfaces`	List the L2 interfaces of a specific host.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404)

6.5.1.1. Request

This table shows the URI parameters for the list interfaces request:

Name	Type	Description
`{host_id}`	UUID	The unique identifier of an existing host.

This operation does not accept a request body.

6.5.1.2. Response

This table shows the body parameters for the list interfaces response:

Name	Type	Description
`iinterfaces`	List (Optional)	The list of L2 interfaces for a specific host.
`ifname`	String (Optional)	The user-specified name of the interface.
`networktype`	String (Optional)	Indicates the type of network that this interface is attached to; `mgmt`, `oam`, `infra`, `none`, `data`, `pci-passthrough` or `pci-sriov`.
`iftype`	String (Optional)	Indicates the type of L2 interface; `ethernet` or `ae` (aggregated ethernet or link aggregation (LAG)) or `vlan` (virtual lan).
`aemode`	String (Optional)	Only applicable if `iftype : ae`, this attribute indicates the basic mode of operation for the AE/LAG interface. Supported modes are: balanced round robin, active-backup, balanced xor, broadcast, 802.3ad, balance-tlb, balance-alb. NOTE only balanced xor and active-standby modes are supported by interfaces of networktype=data.
`txhashpolicy`	String (Optional)	Only applicable if `iftype : ae` and `aemode : balanced`, this attribute indicates what packet headers the AE/LAG is using to distribute packets across the different links/ports of the AE/LAG group; `layer2`, `layer2+3` or `layer3+4`.
`vlan_id`	Integer (Optional)	Only applicable if `iftype : vlan`, this attribute indicates that the vlan interface id. A vlan id between 1 and 4094 (inclusive) must be selected. NOTE The vlan id must be unique for the host interface.
`providernetworks`	List (Optional)	Only applicable if `networktype : data`, this attribute provides a comma-separated list of provider networks that this `data` interface is attached to.
`imac`	String (Optional)	The MAC Address being used by the interface. In the case of AE/LAG, the MAC address of one of the physical ports of the AE/LAG group is used.
`imtu`	Integer	The Maximum Transmission Unit (MTU) of the interface, in bytes.

Name	Type	Description
	(Optional)	
sriov_numvfs	Integer *(Optional)*	The number of VFs configured on the interfaces port; only applicable if `networktype = pci-sriov` where only a single port is associated with the interface.
schedpolicy	String *(Optional)*	Currently not supported.
forihostId	String *(Optional)*	The ID of the host of this interface.
ihost_uuid	UUID *(Optional)*	The UUID of the host of this interface.
uuid	UUID *(Optional)*	The universally unique identifier for this object.
links	List *(Optional)*	For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The following types of link relations are associated with resources: a self link containing a versioned link to the resource, and a bookmark link containing a permanent link to a resource that is appropriate for long term storage.
created_at	DateTime *(Optional)*	The time when the object was created.
updated_at	DateTime *(Optional)*	The time when the object was last updated.
uses	List *(Optional)*	Interfaces which the current interface uses.
used_by	List *(Optional)*	Interfaces which use the current interface.

Example 6.16. List interfaces: JSON response

```
{
  "iinterfaces": [
    {
      "forihostid": 2,
      "iftype": "ethernet",
      "uuid": "1425e76f-eb40-41bd-825f-f692a3064043",
      "links": [
        {
          "href": "http://192.168.204.2:6385/v1/iinterfaces/1425e76f-
eb40-41bd-825f-f692a3064043",
          "rel": "self"
        },
        {
          "href": "http://192.168.204.2:6385/iinterfaces/1425e76f-
eb40-41bd-825f-f692a3064043",
          "rel": "bookmark"
        }
      ],
      "txhashpolicy": null,
      "schedpolicy": null,
      "imac": "08:00:27:80:aa:6e",
      "sriov_numvfs": 0,
```

```
      "ihost_uuid": "ff453a51-1d3b-437f-a65e-b2d163f79f85",
      "vlan_id": null,
      "imtu": 1500,
      "aemode": null,
      "providernetworks": null,
      "networktype": "mgmt",
      "ifname": "eth1"
    },
    {
      "forihostid": 2,
      "iftype": "ae",
      "uuid": "92dec2e1-a793-4c63-a408-affc492b7856",
      "links": [
        {
          "href": "http://192.168.204.2:6385/v1/iinterfaces/92dec2e1-
a793-4c63-a408-affc492b7856",
          "rel": "self"
        },
        {
          "href": "http://192.168.204.2:6385/iinterfaces/92dec2e1-a793-4c63-
a408-affc492b7856",
          "rel": "bookmark"
        }
      ],
      "txhashpolicy": "layer2",
      "schedpolicy": null,
      "imac": null,
      "sriov_numvfs": 0,
      "ihost_uuid": "ff453a51-1d3b-437f-a65e-b2d163f79f85",
      "imtu": 1500,
      "uses": [
        "eth2",
        "eth3"
      ],
      "used_by": [

      ],
      "aemode": "balanced",
      "providernetworks": "physnet-0,physnet-1",
      "networktype": "data",
      "ifname": "data1"
    }
  ]
}
```

This operation does not return a response body.

6.5.2. Show interface

Method	URI	Description
GET	`/v1/iinterfaces/{interface_id}`	Shows information about a specific L2 interface.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

6.5.2.1. Request

This table shows the URI parameters for the show interface request:

Name	Type	Description
`{interface_id}`	UUID	The unique identifier of an existing interface.

This operation does not accept a request body.

6.5.2.2. Response

This table shows the body parameters for the show interface response:

Name	Type	Description
`iports`	List (Optional)	URIs to the physical ports of this interface.
`ifname`	String (Optional)	The user-specified name of the interface.
`networktype`	String (Optional)	Indicates the type of network that this interface is attached to; `mgmt`, `oam`, `infra`, `none`, `data`, `pci-passthrough` or `pci-sriov`.
`iftype`	String (Optional)	Indicates the type of L2 interface; `ethernet` or `ae` (aggregated ethernet or link aggregation (LAG)) or `vlan` (virtual lan).
`aemode`	String (Optional)	Only applicable if `iftype : ae`, this attribute indicates the basic mode of operation for the AE/LAG interface. Supported modes are: balanced round robin, active-backup, balanced xor, broadcast, 802.3ad, balance-tlb, balance-alb. NOTE only balanced xor and active-standby modes are supported by interfaces of networktype=data.
`txhashpolicy`	String (Optional)	Only applicable if `iftype : ae` and `aemode : balanced`, this attribute indicates what packet headers the AE/LAG is using to distribute packets across the different links/ports of the AE/LAG group; `layer2`, `layer2+3` or `layer3+4`.
`vlan_id`	Integer (Optional)	Only applicable if `iftype : vlan`, this attribute indicates that the vlan interface id. A vlan id between 1 and 4094 (inclusive) must be selected. NOTE The vlan id must be unique for the host interface.
`providernetworks`	List (Optional)	Only applicable if `networktype : data`, this attribute provides a comma-separated list of provider networks that this `data` interface is attached to.
`imac`	String (Optional)	The MAC Address being used by the interface. In the case of AE/LAG, the MAC address of one of the physical ports of the AE/LAG group is used.
`imtu`	Integer	The Maximum Transmission Unit (MTU) of the interface, in bytes.

Name	Type	Description
	(Optional)	
sriov_numvfs	Integer (Optional)	The number of VFs configured on the interfaces port; only applicable if `networktype = pci-sriov` where only a single port is associated with the interface.
schedpolicy	String (Optional)	Currently not supported.
forihostId	String (Optional)	The ID of the host of this interface.
ihost_uuid	UUID (Optional)	The UUID of the host of this interface.
ifcapabilities	List (Optional)	Currently not supported.
providernetworksdict	List (Optional)	Currently not supported.
ports	List (Optional)	Use `iports`.
uses	List (Optional)	Interfaces which the current interface uses.
used_by	List (Optional)	Interfaces which use the current interface.
uuid	UUID (Optional)	The universally unique identifier for this object.
links	List (Optional)	For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The following types of link relations are associated with resources: a self link containing a versioned link to the resource, and a bookmark link containing a permanent link to a resource that is appropriate for long term storage.
created_at	DateTime (Optional)	The time when the object was created.
updated_at	DateTime (Optional)	The time when the object was last updated.

Example 6.17. Show interface: JSON response

```
{
   "iports" : [
      {
         "rel" : "self",
         "href" : "http://10.10.10.2:6385/v1/iinterfaces/740a5bec-
b7a8-4645-93ed-aea0d4cfbf86/iports"
      },
      {
         "rel" : "bookmark",
         "href" : "http://10.10.10.2:6385/iinterfaces/740a5bec-b7a8-4645-93ed-
aea0d4cfbf86/iports"
      }
   ],
```

```
      "providernetworks" : "physnet-0,physnet-1",
      "txhashpolicy" : "layer2",
      "schedpolicy" : null,
      "networktype" : "data",
      "uuid" : "740a5bec-b7a8-4645-93ed-aea0d4cfbf86",
      "ihost_uuid" : "ff453a51-1d3b-437f-a65e-b2d163f79f85",
      "vlan_id": null,
      "created_at" : "2014-09-29T11:12:42.556372+00:00",
      "ifcapabilities" : {},
      "iftype" : "ae",
      "links" : [
          {
              "rel" : "self",
              "href" : "http://10.10.10.2:6385/v1/iinterfaces/740a5bec-
  b7a8-4645-93ed-aea0d4cfbf86"
          },
          {
              "rel" : "bookmark",
              "href" : "http://10.10.10.2:6385/iinterfaces/740a5bec-b7a8-4645-93ed-
  aea0d4cfbf86"
          }
      ],
      "providernetworksdict" : {},
      "imac" : null,
      "sriov_numvfs": 0,
      "aemode" : "balanced",
      "ifname" : "data1",
      "ports" : null,
      "uses": [

      ],
      "used_by": [

      ],
      "forihostid" : 2,
      "updated_at" : null,
      "imtu" : 1500
}
```

This operation does not return a response body.

6.5.3. Create interface

Method	URI	Description
POST	`/v1/ihosts/{host_id}/iinterfaces`	Creates an L2 interface on a specific host.

Note that `mgmt` and `oam` can only be added through the REST API if their networktype is unassigned to none. `mgmt` and `oam` are automatically added by the system by default.

Normal response codes: 200

Error response codes: badMediaType (415)

6.5.3.1. Request

This table shows the URI parameters for the create interface request:

Name	Type	Description
`{host_id}`	UUID	The unique identifier of an existing host.

This table shows the body parameters for the create interface request:

Name	Type	Description
`ifname`	String *(Optional)*	The name for the interface.
`networktype`	String *(Optional)*	The type of network that this interface will be attached to; i.e. `mgmt`, `infra`, `oam`, `data`, `pci-passthrough`, `pci-sriov`, `infra`.
`iftype`	String *(Optional)*	The type of interface; i.e. `ae` or `vlan`.
`aemode`	String *(Optional)*	Only applicable if `iftype : ae`, this attribute specifies whether the AE/LAG should operate as `balanced` or `active_standby` or `802.3ad` across its links. The `balanced` and `active_standby` are the only modes supported by `data` type interface. For `mgmt` type interface the `802.3ad` option must be selected.
`txhashpolicy`	String *(Optional)*	Only applicable if `iftype : ae` and `aemode : balanced`, this attribute specifies what packet headers the AE/LAG should use to distribute packets across the different links/ports of the AE/LAG group; `layer2`, `layer2+3` or `layer3+4`.
`vlan_id`	Integer *(Optional)*	Only applicable if `iftype : vlan`, this attribute specifies a virtual lan id for a vlan interface type.
`providernetworks`	List *(Optional)*	Only applicable if `networktype : data`, this attribute specifies a comma-separated list of provider networks that this `data` interface is attached to.
`ports`	List *(Optional)*	This attribute specifies a comma-separated list of ports that this interface contains. If `iftype : ethernet` then only one port is allowed.
`uses`	List *(Optional)*	Only applicable if `iftype : ae` or `iftype: vlan`, this attribute specifies a comma-separated list of interfaces that this interface uses.
`used_by`	List	This attribute specifies a comma-separated list of interfaces that use this interface.

Name	Type	Description
	(Optional)	
`imtu`	Integer *(Optional)*	This attribute specifies the interface's Maximum Transmit Unit.
`sriov_numvfs`	Integer *(Optional)*	The number of VFs to configure on the interface's port; only applicable if `networktype = pci-sriov` where only a single port is associated with the interface.
`ihost_uuid`	String *(Optional)*	The UUID of the host to create the interface on.

Example 6.18. Create interface: JSON request

```
{
  "iftype": "ae",
  "txhashpolicy": "layer2",
  "ihost_uuid": "ff453a51-1d3b-437f-a65e-b2d163f79f85",
  "imtu": "1500",
  "providernetworks": "physnet-0,physnet1",
  "networktype": "data",
  "ifname": "data1",
  "uses": ['eth2','eth3'],
  "aemode": "balanced",
  "sriov_numvfs": 0
}
```

This operation does not accept a request body.

6.5.3.2. Response

This table shows the body parameters for the create interface response:

Name	Type	Description
`iports`	List *(Optional)*	URIs to the physical ports of this interface.
`ifname`	String *(Optional)*	The user-specified name of the interface.
`networktype`	String *(Optional)*	Indicates the type of network that this interface is attached to; `mgmt`, `oam`, `infra`, `none`, `data`, `pci-passthrough` or `pci-sriov`.
`iftype`	String *(Optional)*	Indicates the type of L2 interface; `ethernet` or `ae` (aggregated ethernet or link aggregation (LAG)) or `vlan` (virtual lan).
`aemode`	String *(Optional)*	Only applicable if `iftype : ae`, this attribute indicates the basic mode of operation for the AE/LAG interface. Supported modes are: balanced round robin, active-backup, balanced xor, broadcast, 802.3ad, balance-tlb, balance-alb. NOTE only balanced xor and active-standby modes are supported by interfaces of networktype=data.
`txhashpolicy`	String *(Optional)*	Only applicable if `iftype : ae` and `aemode : balanced`, this attribute indicates what packet headers the AE/LAG is using to distribute packets across the different links/ports of the AE/LAG group; `layer2`, `layer2+3` or `layer3+4`.
`vlan_id`	Integer *(Optional)*	Only applicable if `iftype : vlan`, this attribute indicates that the vlan interface id. A vlan id between 1 and 4094 (inclusive) must be selected. NOTE The vlan id must be unique for the host interface.

Name	Type	Description
providernetworks	List *(Optional)*	Only applicable if `networktype : data`, this attribute provides a comma-separated list of provider networks that this `data` interface is attached to.
imac	String *(Optional)*	The MAC Address being used by the interface. In the case of AE/LAG, the MAC address of one of the physical ports of the AE/LAG group is used.
imtu	Integer *(Optional)*	The Maximum Transmission Unit (MTU) of the interface, in bytes.
sriov_numvfs	Integer *(Optional)*	The number of VFs configured on the interfaces port; only applicable if `networktype = pci-sriov` where only a single port is associated with the interface.
schedpolicy	String *(Optional)*	Currently not supported.
forihostId	String *(Optional)*	The ID of the host of this interface.
ihost_uuid	UUID *(Optional)*	The UUID of the host of this interface.
ifcapabilities	List *(Optional)*	Currently not supported.
providernetworksdict	List *(Optional)*	Currently not supported.
ports	List *(Optional)*	Use `iports`.
uses	List *(Optional)*	Interfaces which the current interface uses.
used_by	List *(Optional)*	Interfaces which use the current interface.
uuid	UUID *(Optional)*	The universally unique identifier for this object.
links	List *(Optional)*	For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The following types of link relations are associated with resources: a self link containing a versioned link to the resource, and a bookmark link containing a permanent link to a resource that is appropriate for long term storage.
created_at	DateTime *(Optional)*	The time when the object was created.
updated_at	DateTime *(Optional)*	The time when the object was last updated.

Example 6.19. Create interface: JSON response

```
{
  "iports": [
    {
      "href": "http://192.168.204.2:6385/v1/iinterfaces/92dec2e1-a793-4c63-
a408-affc492b7856/iports",
```

```
      "rel": "self"
    },
    {
      "href": "http://192.168.204.2:6385/iinterfaces/92dec2e1-a793-4c63-a408-
affc492b7856/iports",
      "rel": "bookmark"
    }
  ],
  "forihostid": 2,
  "iftype": "ae",
  "uuid": "92dec2e1-a793-4c63-a408-affc492b7856",
  "links": [
    {
      "href": "http://192.168.204.2:6385/v1/iinterfaces/92dec2e1-a793-4c63-
a408-affc492b7856",
      "rel": "self"
    },
    {
      "href": "http://192.168.204.2:6385/iinterfaces/92dec2e1-a793-4c63-a408-
affc492b7856",
      "rel": "bookmark"
    }
  ],
  "ihost_uuid": "ff453a51-1d3b-437f-a65e-b2d163f79f85",
  "vlan_id": null,
  "txhashpolicy": "layer2",
  "created_at": "2014-09-29T10:55:20.515705+00:00",
  "schedpolicy": null,
  "providernetworksdict": {

  },
  "imac": null,
  "updated_at": null,
  "ifcapabilities": {

  },
  "imtu": 1500,
  "uses": [
    "eth2",
    "eth3"
  ],
  "used_by": [

  ],
  "aemode": "balanced",
  "sriov_numvfs": 0,
  "providernetworks": "physnet-0,physnet-1",
  "networktype": "data",
  "ifname": "data1",
  "ports": null,
}
```

This operation does not return a response body.

6.5.4. Modify interface

Method	URI	Description
PATCH	`/v1/iinterfaces/{interface_id}`	Modifies a specific L2 interface.

Normal response codes: 200

Error response codes: badMediaType (415)

6.5.4.1. Request

This table shows the URI parameters for the modify interface request:

Name	Type	Description
`{interface_id}`	UUID	The unique identifier of an existing interface.

This table shows the body parameters for the modify interface request:

Name	Type	Description
`ifname`	String *(Optional)*	The name for the interface.
`networktype`	String *(Optional)*	The type of network that this interface will be attached to; i.e. `mgmt`, `infra`, `oam`, `data`, `pci-passthrough`, `pci-sriov`, `infra` or `none`.
`iftype`	String *(Optional)*	The type of interface; i.e. `ethernet` or `ae` or `vlan`.
`aemode`	String *(Optional)*	Only applicable if `iftype : ae`, this attribute specifies whether the AE/LAG should operate as `balanced` or `active_standby` across its links. These are the only modes supported by `data` type interface.
`txhashpolicy`	String *(Optional)*	Only applicable if `iftype : ae` and `aemode : balanced`, this attribute specifies what packet headers the AE/LAG should use to distribute packets across the different links/ports of the AE/LAG group; `layer2`, `layer2+3` or `layer3+4`.
`vlan_id`	Integer *(Optional)*	Only applicable if `iftype : vlan`, this attribute specifies a virtual lan id for a vlan interface type.
`providernetworks`	List *(Optional)*	Only applicable if `networktype : data`, this attribute specifies a comma-separated list of provider networks that this `data` interface is attached to.
`ports`	List *(Optional)*	This attribute specifies a comma-separated list of ports that this interface contains. If `iftype : ethernet` then only one port is allowed.
`uses`	List *(Optional)*	Only applicable if `iftype : ae` or `iftype: vlan`, this attribute specifies a comma-separated list of interfaces that this interface uses.
`used_by`	List *(Optional)*	This attribute specifies a comma-separated list of interfaces that use this interface.
`imtu`	Integer *(Optional)*	This attribute specifies the interface's Maximum Transmit Unit.
`sriov_numvfs`	Integer *(Optional)*	The number of VFs to configure on the interface's port; only applicable if `networktype = pci-sriov` where only a single port is associated with the interface.

Example 6.20. Modify interface: JSON request

```
[
  {
    "path": "/imtu",
    "value": "1500",
    "op": "replace"
  },
  {
    "path": "/txhashpolicy",
    "value": "layer2",
    "op": "replace"
  },
  {
    "path": "/providernetworks",
    "value": "physnet-0,physnet-1",
    "op": "replace"
  },
  {
    "path": "/aemode",
    "value": "active_standby",
    "op": "replace"
  },
  {
    "path": "/uses",
    "value": ['eth2','eth3'],
    "op": "replace"
  }
]
```

This operation does not accept a request body.

6.5.4.2. Response

This table shows the body parameters for the modify interface response:

Name	Type	Description
iports	List (Optional)	URIs to the physical ports of this interface.
ifname	String (Optional)	The user-specified name of the interface.
networktype	String (Optional)	Indicates the type of network that this interface is attached to; mgmt, oam, infra, none, data, pci-passthrough or pci-sriov.
iftype	String (Optional)	Indicates the type of L2 interface; ethernet or ae (aggregated ethernet or link aggregation (LAG)) or vlan (virtual lan).
aemode	String (Optional)	Only applicable if iftype : ae, this attribute indicates the basic mode of operation for the AE/LAG interface. Supported modes are: balanced round robin, active-backup, balanced xor, broadcast, 802.3ad, balance-tlb, balance-alb. NOTE only balanced xor and active-standby modes are supported by interfaces of networktype=data.
txhashpolicy	String (Optional)	Only applicable if iftype : ae and aemode : balanced, this attribute indicates what packet headers the AE/LAG is using to distribute packets across the different links/ports of the AE/LAG group; layer2, layer2+3 or layer3+4.

Name	Type	Description
vlan_id	Integer *(Optional)*	Only applicable if `iftype : vlan`, this attribute indicates that the vlan interface id. A vlan id between 1 and 4094 (inclusive) must be selected. NOTE The vlan id must be unique for the host interface.
providernetworks	List *(Optional)*	Only applicable if `networktype : data`, this attribute provides a comma-separated list of provider networks that this `data` interface is attached to.
imac	String *(Optional)*	The MAC Address being used by the interface. In the case of AE/LAG, the MAC address of one of the physical ports of the AE/LAG group is used.
imtu	Integer *(Optional)*	The Maximum Transmission Unit (MTU) of the interface, in bytes.
sriov_numvfs	Integer *(Optional)*	The number of VFs configured on the interfaces port; only applicable if `networktype = pci-sriov` where only a single port is associated with the interface.
schedpolicy	String *(Optional)*	Currently not supported.
forihostId	String *(Optional)*	The ID of the host of this interface.
ihost_uuid	UUID *(Optional)*	The UUID of the host of this interface.
ifcapabilities	List *(Optional)*	Currently not supported.
providernetworksdict	List *(Optional)*	Currently not supported.
ports	List *(Optional)*	Use `iports`.
uses	List *(Optional)*	Interfaces which the current interface uses.
used_by	List *(Optional)*	Interfaces which use the current interface.
uuid	UUID *(Optional)*	The universally unique identifier for this object.
links	List *(Optional)*	For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The following types of link relations are associated with resources: a self link containing a versioned link to the resource, and a bookmark link containing a permanent link to a resource that is appropriate for long term storage.
created_at	DateTime *(Optional)*	The time when the object was created.
updated_at	DateTime *(Optional)*	The time when the object was last updated.

Example 6.21. Modify interface: JSON response

```
{
  "iports": [
```

```
    {
        "href": "http://192.168.204.2:6385/v1/iinterfaces/92dec2e1-a793-4c63-
a408-affc492b7856/iports",
        "rel": "self"
    },
    {
        "href": "http://192.168.204.2:6385/iinterfaces/92dec2e1-a793-4c63-a408-
affc492b7856/iports",
        "rel": "bookmark"
    }
  ],
  "forihostid": 2,
  "iftype": "ae",
  "uuid": "92dec2e1-a793-4c63-a408-affc492b7856",
  "links": [
    {
        "href": "http://192.168.204.2:6385/v1/iinterfaces/92dec2e1-a793-4c63-
a408-affc492b7856",
        "rel": "self"
    },
    {
        "href": "http://192.168.204.2:6385/iinterfaces/92dec2e1-a793-4c63-a408-
affc492b7856",
        "rel": "bookmark"
    }
  ],
  "ihost_uuid": "ff453a51-1d3b-437f-a65e-b2d163f79f85",
  "vlan_id": null,
  "txhashpolicy": "layer2",
  "created_at": "2014-09-29T10:55:20.515705+00:00",
  "schedpolicy": null,
  "providernetworksdict": {

  },
  "imac": null,
  "sriov_numvfs": 0,
  "updated_at": "2014-09-29T11:08:21.016145+00:00",
  "ifcapabilities": {

  },
  "imtu": 1500,
  "uses": [
    "eth2",
    "eth3"
  ],
  "used_by": [

  ],
  "aemode": "active_standby",
  "providernetworks": "physnet-0,physnet-1",
  "networktype": "data",
  "ifname": "data1",
  "ports": null
}
```

This operation does not return a response body.

6.5.5. Delete interface

Method	URI	Description
DELETE	`/v1/iinterfaces/{interface_id}`	Deletes a specific L2 interface.

Normal response codes: 204

6.5.5.1. Request

This table shows the URI parameters for the delete interface request:

Name	Type	Description
`{interface_id}`	UUID	The unique identifier of an existing interface.

This operation does not accept a request body.

6.5.5.2. Response

This operation does not return a response body.

6.6. CPUs

These APIs allow the display of the logical core(s) of the processor(s) on a host, and the display and modification of the cores assigned function.

Method	URI	Description
GET	`/v1/ihosts/{host_id}/icpus`	Lists all cpus (logical processor cores) of a host.
GET	`/v1/icpus/{cpu_id}`	Shows information about a specific cpu (logical processor core).
PATCH	`/v1/icpus/{cpu_id}`	Modifies the number of cores assigned to different functions on a host.

6.6.1. List cpus

Method	URI	Description
GET	`/v1/ihosts/{host_id}/icpus`	Lists all cpus (logical processor cores) of a host.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

6.6.1.1. Request

This table shows the URI parameters for the list cpus request:

Name	Type	Description
`{host_id}`	UUID	The unique identifier of an existing host.

This operation does not accept a request body.

6.6.1.2. Response

This table shows the body parameters for the list cpus response:

Name	Type	Description
`icpus`	List (Optional)	The list of cpus (logical processor cores) of a host.
`cpu`	String (Optional)	The logical core number.
`numa_node`	String (Optional)	The NUMA Node or physical processor device of the logical core.
`core`	String (Optional)	The physical core of the logical core.
`thread`	String (Optional)	The thread within the physical core of the logical core.
`allocated_function`	String (Optional)	The function assigned to this logical core; valid values are Platform, Vswitch, Shared or VMs . `Platform` indicates the core is used for the host kernel, TiS and OpenStack Services, `Vswitch` indicates the core is used by AVS, the virtual switch, `Shared` indicates that the core is reserved for sharing by VMs using the hw:wrs:shared_vcpu flavor extra spec, `VMs` indicates that the core is available for use by VMs.
`cpu_family`	String (Optional)	The CPU Family for the processor of the logical core.
`cpu_model`	String	The vendor, model, part number and other info related to the processor device of the logical core.

Name	Type	Description
	(Optional)	
ihost_uuid	UUID (Optional)	The UUID of the host of the logical core.
inode_uuid	UUID (Optional)	The UUID of the NUMA Node of the logical core.
uuid	UUID (Optional)	The universally unique identifier for this object.
links	List (Optional)	For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The following types of link relations are associated with resources: a self link containing a versioned link to the resource, and a bookmark link containing a permanent link to a resource that is appropriate for long term storage.
created_at	DateTime (Optional)	The time when the object was created.
updated_at	DateTime (Optional)	The time when the object was last updated.

Example 6.22. List cpus: JSON response

```json
{
   "icpus" : [
      {
         "core" : 0,
         "allocated_function" : "Platform",
         "cpu" : 0,
         "numa_node" : 0,
         "uuid" : "d269a009-de03-463e-a553-10944361d38b",
         "cpu_family" : "6",
         "ihost_uuid" : "0aca08f9-882f-4491-8ffd-7368d20ead48",
         "thread" : 0,
         "created_at" : "2014-09-18T03:12:15.429976+00:00",
         "cpu_model" : "Intel(R) Xeon(R) CPU E5-2690 0 @ 2.90GHz",
         "updated_at" : null,
         "capabilities" : {},
         "links" : [
            {
               "rel" : "self",
               "href" : "http://128.224.151.243:6385/v1/icpus/d269a009-
de03-463e-a553-10944361d38b"
            },
            {
               "rel" : "bookmark",
               "href" : "http://128.224.151.243:6385/icpus/d269a009-de03-463e-
a553-10944361d38b"
            }
         ],
         "inode_uuid" : "e5d68519-eb07-4b28-8ca2-32bb476eeec1"
      },
      {
         "core" : 1,
         "allocated_function" : "Vswitch",
         "cpu" : 1,
         "numa_node" : 0,
```

```
            "uuid" : "f236a371-48e9-4618-8f02-3a7ea0c2d16e",
            "cpu_family" : "6",
            "ihost_uuid" : "0aca08f9-882f-4491-8ffd-7368d20ead48",
            "thread" : 0,
            "created_at" : "2014-09-18T03:12:15.432471+00:00",
            "cpu_model" : "Intel(R) Xeon(R) CPU E5-2690 0 @ 2.90GHz",
            "updated_at" : null,
            "capabilities" : {},
            "links" : [
               {
                  "rel" : "self",
                  "href" : "http://128.224.151.243:6385/v1/icpus/
f236a371-48e9-4618-8f02-3a7ea0c2d16e"
               },
               {
                  "rel" : "bookmark",
                  "href" : "http://128.224.151.243:6385/icpus/
f236a371-48e9-4618-8f02-3a7ea0c2d16e"
               }
            ],
            "inode_uuid" : "e5d68519-eb07-4b28-8ca2-32bb476eeec1"
         },

...

    ]
}
```

This operation does not return a response body.

6.6.2. Show cpu

Method	URI	Description
GET	`/v1/icpus/{cpu_id}`	Shows information about a specific cpu (logical processor core).

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404)

6.6.2.1. Request

This table shows the URI parameters for the show cpu request:

Name	Type	Description
`{cpu_id}`	UUID	The unique identifier of a cpu (logical processor core).

This operation does not accept a request body.

6.6.2.2. Response

This table shows the body parameters for the show cpu response:

Name	Type	Description
`cpu`	String *(Optional)*	The logical core number.
`numa_node`	String *(Optional)*	The NUMA Node or physical processor device of the logical core.
`core`	String *(Optional)*	The physical core of the logical core.
`thread`	String *(Optional)*	The thread within the physical core of the logical core.
`allocated_function`	String *(Optional)*	The function assigned to this logical core; valid values are Platform, Vswitch, Shared or VMs . `Platform` indicates the core is used for the host kernel, TiS and OpenStack Services, `Vswitch` indicates the core is used by AVS, the virtual switch, `Shared` indicates that the core is reserved for sharing by VMs using the hw:wrs:shared_vcpu flavor extra spec, `VMs` indicates that the core is available for use by VMs.
`cpu_family`	String *(Optional)*	The CPU Family for the processor of the logical core.
`cpu_model`	String *(Optional)*	The vendor, model, part number and other info related to the processor device of the logical core.

Name	Type	Description
ihost_uuid	UUID *(Optional)*	The UUID of the host of the logical core.
inode_uuid	UUID *(Optional)*	The UUID of the NUMA Node of the logical core.
uuid	UUID *(Optional)*	The universally unique identifier for this object.
links	List *(Optional)*	For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The following types of link relations are associated with resources: a self link containing a versioned link to the resource, and a bookmark link containing a permanent link to a resource that is appropriate for long term storage.
created_at	DateTime *(Optional)*	The time when the object was created.
updated_at	DateTime *(Optional)*	The time when the object was last updated.

Example 6.23. Show cpu: JSON response

```
{
   "core" : 0,
   "allocated_function" : "Platform",
   "function" : null,
   "uuid" : "d269a009-de03-463e-a553-10944361d38b",
   "cpu_family" : "6",
   "ihost_uuid" : "0aca08f9-882f-4491-8ffd-7368d20ead48",
   "created_at" : "2014-09-18T03:12:15.429976+00:00",
   "num_cores_on_processor1" : null,
   "capabilities" : {},
   "num_cores_on_processor3" : null,
   "links" : [
      {
         "rel" : "self",
         "href" : "http://128.224.151.243:6385/v1/icpus/d269a009-de03-463e-
a553-10944361d38b"
      },
      {
         "rel" : "bookmark",
         "href" : "http://128.224.151.243:6385/icpus/d269a009-de03-463e-
a553-10944361d38b"
      }
   ],
   "num_cores_on_processor2" : null,
   "cpu" : 0,
   "numa_node" : 0,
   "num_cores_on_processor0" : null,
   "thread" : 0,
   "cpu_model" : "Intel(R) Xeon(R) CPU E5-2690 0 @ 2.90GHz",
   "updated_at" : null,
   "inode_uuid" : "e5d68519-eb07-4b28-8ca2-32bb476eeec1"
}
```

This operation does not return a response body.

6.6.3. Modify cpu

Method	URI	Description
PATCH	`/v1/icpus/{cpu_id}`	Modifies the number of cores assigned to different functions on a host.

Normal response codes: 200

Error response codes: badMediaType (415)

6.6.3.1. Request

This table shows the URI parameters for the modify cpu request:

Name	Type	Description
`{cpu_id}`	UUID	The unique identifier of a cpu (logical processor core).

This table shows the body parameters for the modify cpu request:

Name	Type	Description
`function`	String *(Optional)*	This parameter specifies the function that is being assigned a different number of cores. The only functions currently allowed to be assigned a different number of cores `vswitch` and `shared`. `vswitch` function is for managing the cores dedicated to the vswitch. `shared` function is for managing the cores reserved for sharing by VMs using the hw:wrs:shared_vcpu flavor extra spec.
`num_cores_on_processor0`	String *(Optional)*	The number of cores on processor 0 assigned to this function.
`num_cores_on_processor1`	String *(Optional)*	The number of cores on processor 1 assigned to this function.
`num_cores_on_processor2`	String *(Optional)*	The number of cores on processor 2 assigned to this function.
`num_cores_on_processor3`	String *(Optional)*	The number of cores on processor 3 assigned to this function.

Example 6.24. Modify cpu: JSON request

```
[
  {
    "path": "/function",
    "value": "vswitch",
    "op": "replace"
  },
  {
    "path": "/num_cores_on_processor0",
    "value": "1",
    "op": "replace"
  }
]
```

This operation does not accept a request body.

6.6.3.2. Response

This table shows the body parameters for the modify cpu response:

Name	Type	Description
cpu	String (Optional)	The logical core number.
numa_node	String (Optional)	The NUMA Node or physical processor device of the logical core.
core	String (Optional)	The physical core of the logical core.
thread	String (Optional)	The thread within the physical core of the logical core.
allocated_function	String (Optional)	The function assigned to this logical core; valid values are Platform, Vswitch, Shared or VMs . Platform indicates the core is used for the host kernel, TiS and Open-Stack Services, Vswitch indicates the core is used by AVS, the virtual switch, Shared indicates that the core is reserved for sharing by VMs using the hw:wrs:shared_vcpu flavor extra spec, VMs indicates that the core is available for use by VMs.
cpu_family	String (Optional)	The CPU Family for the processor of the logical core.
cpu_model	String (Optional)	The vendor, model, part number and other info related to the processor device of the logical core.
ihost_uuid	UUID (Optional)	The UUID of the host of the logical core.
inode_uuid	UUID (Optional)	The UUID of the NUMA Node of the logical core.

Example 6.25. Modify cpu: JSON response

```
{
  "allocated_function": "VMs",
  "core": 7,
  "thread": 0,
  "links": [
    {
      "href": "http://192.168.204.2:6385/v1/icpus/8a960696-c242-436f-a79c-
d904fa6dcbd2",
      "rel": "self"
    },
    {
      "href": "http://192.168.204.2:6385/icpus/8a960696-c242-436f-a79c-
d904fa6dcbd2",
      "rel": "bookmark"
    }
  ],
  "inode_uuid": "4ed560e2-c3a1-4d41-8b00-6af257e6ac75",
```

```
    "function": null,
    "numa_node": 1,
    "created_at": "2014-09-26T02:01:36.514217+00:00",
    "cpu_model": "Intel(R) Xeon(R) CPU E5-2670 0 @ 2.60GHz",
    "capabilities": {

    },
    "updated_at": "2014-09-26T11:58:20.586235+00:00",
    "num_cores_on_processor1": null,
    "ihost_uuid": "22d5827c-7a04-4a3c-9509-e8849b9a595d",
    "num_cores_on_processor3": null,
    "num_cores_on_processor2": null,
    "num_cores_on_processor0": null,
    "cpu_family": "6",
    "cpu": 15,
    "uuid": "8a960696-c242-436f-a79c-d904fa6dcbd2"
}
```

This operation does not return a response body.

6.7. Memory

These APIs allow the display of the size and usage of various memory areas of the NU-MA nodes of a host. The modification of the size of these memory areas is also supported through these APIs. The different memory areas of a NUMA node of a host are:

- Memory reserved for the Platform; where the Platform consists of the kernel and the cloud services,

- Memory reserved for the virtual switch (Note: only on 'compute' hosts),

- Memory reserved for the hosted VMs (Note: only on 'compute' hosts).

Method	URI	Description
GET	/v1/ihosts/{host_id}/imemorys	Lists the memory information of all NUMA nodes of a host.
GET	/v1/imemorys/{memory_id}	Shows the memory information about a specific NUMA node of a specific host.
PATCH	/v1/imemorys/{memory_id}	Modifies the memory information about a specific NUMA node of a specific host..

6.7.1. List memorys

Method	URI	Description
GET	`/v1/ihosts/{host_id}/imemorys`	Lists the memory information of all NUMA nodes of a host.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404)

6.7.1.1. Request

This table shows the URI parameters for the list memorys request:

Name	Type	Description
`{host_id}`	UUID	The unique identifier of an existing host.

This operation does not accept a request body.

6.7.1.2. Response

This table shows the body parameters for the list memorys response:

Name	Type	Description
`imemorys`	List *(Optional)*	The list of NUMA nodes (and their associated memory information) for this host.
`ihost_uuid`	UUID *(Optional)*	The UUID of the host.
`numa_node`	Integer *(Optional)*	The NUMA node number.
`inode_uuid`	UUID *(Optional)*	The UUID of the NUMA node.
`platform_reserved_mib`	Integer *(Optional)*	Memory reserved for the Kernel and Cloud Platform Services, in MiBs.
`memtotal_mib`	Integer *(Optional)*	Total memory reserved for the hosted Virtual Machines, in MiBs.
`memavail_mib`	Integer *(Optional)*	Free / available memory from the total memory reserved for the hosted Virtual Machines, in MiBs.
`hugepages_configured`	Boolean *(Optional)*	Boolean indicating whether huge page memory is configured or not.
`avs_hugepages_size_mib`	Integer *(Optional)*	The size of an Accelerated Virtual Switch (AVS) huge page, in MiBs.
`avs_hugepages_nr`	Integer	The total number of Accelerated Virtual Switch (AVS) huge pages.

Name	Type	Description
	(Optional)	
avs_hugepages_avail	Integer (Optional)	The free / available Accelerated Virtual Switch (AVS) huge pages.
vm_hugepages_nr_1G	Integer (Optional)	The total number of Virtual Machine 1G huge pages.
vm_hugepages_avail_1G	Integer (Optional)	The free / available Virtual Machine 1G huge pages.
vm_hugepages_nr_1G_pending	Integer (Optional)	If not null, the pending configured number of Virtual Machine 1G huge pages.
vm_hugepages_nr_2M	Integer (Optional)	The total number of Virtual Machine 2M huge pages.
vm_hugepages_avail_2M	Integer (Optional)	The free / available Virtual Machine 2M huge pages.
vm_hugepages_nr_2M_pending	Integer (Optional)	If not null, the pending configured number of Virtual Machine 2M huge pages.
uuid	UUID (Optional)	The universally unique identifier for this object.
links	List (Optional)	For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The following types of link relations are associated with resources: a self link containing a versioned link to the resource, and a bookmark link containing a permanent link to a resource that is appropriate for long term storage.
created_at	DateTime (Optional)	The time when the object was created.
updated_at	DateTime (Optional)	The time when the object was last updated.

Example 6.26. List memorys: JSON response

```
{
    "imemorys" : [
        {
            "avs_hugepages_nr" : 1,
            "hugepages_configured" : "True",
            "vm_hugepages_nr_1G_pending" : null,
            "memavail_mib" : 22448,
            "uuid" : "34098f3a-6b95-4cad-aecb-986dc6312f4f",
            "ihost_uuid" : "afecdcfb-2954-498d-88bf-d1385b00f34d",
            "created_at" : "2015-04-06T20:27:50.171841+00:00",
            "avs_hugepages_size_mib" : 1024,
            "vm_hugepages_avail_1G" : 0,
            "capabilities" : {},
            "links" : [
                {
                    "rel" : "self",
                    "href" : "http://128.224.150.54:6385/v1/imemorys/
34098f3a-6b95-4cad-aecb-986dc6312f4f"
                },
```

```
            {
                "rel" : "bookmark",
                "href" : "http://128.224.150.54:6385/imemorys/
34098f3a-6b95-4cad-aecb-986dc6312f4f"
            }
        ],
        "vm_hugepages_nr_2M_pending" : null,
        "avs_hugepages_reqd" : null,
        "vm_hugepages_avail_2M" : 11224,
        "avs_hugepages_avail" : 0,
        "numa_node" : 0,
        "vm_hugepages_nr_1G" : 0,
        "updated_at" : "2015-04-08T11:32:25.205552+00:00",
        "platform_reserved_mib" : 4000,
        "memtotal_mib" : 27056,
        "vm_hugepages_nr_2M" : 13016,
        "inode_uuid" : "c65c852c-1707-40e1-abfc-334270ec0427"
    },
    {
        "avs_hugepages_nr" : 1,
        "hugepages_configured" : "True",
        "vm_hugepages_nr_1G_pending" : null,
        "memavail_mib" : 24082,
        "uuid" : "85df5109-77d9-4335-9181-0efa82c98dcc",
        "ihost_uuid" : "afecdcfb-2954-498d-88bf-d1385b00f34d",
        "created_at" : "2015-04-06T20:27:50.182764+00:00",
        "avs_hugepages_size_mib" : 1024,
        "vm_hugepages_avail_1G" : 0,
        "capabilities" : {},
        "links" : [
            {
                "rel" : "self",
                "href" : "http://128.224.150.54:6385/v1/imemorys/
85df5109-77d9-4335-9181-0efa82c98dcc"
            },
            {
                "rel" : "bookmark",
                "href" : "http://128.224.150.54:6385/imemorys/
85df5109-77d9-4335-9181-0efa82c98dcc"
            }
        ],
        "vm_hugepages_nr_2M_pending" : null,
        "avs_hugepages_reqd" : null,
        "vm_hugepages_avail_2M" : 12041,
        "avs_hugepages_avail" : 0,
        "numa_node" : 1,
        "vm_hugepages_nr_1G" : 0,
        "updated_at" : "2015-04-08T11:32:25.220242+00:00",
        "platform_reserved_mib" : 2000,
        "memtotal_mib" : 29202,
        "vm_hugepages_nr_2M" : 14089,
        "inode_uuid" : "67d3c9a0-57b2-4532-b7ea-f1cd16a3b349"
    }
  ]
}
```

This operation does not return a response body.

6.7.2. Show memory

Method	URI	Description
GET	`/v1/imemorys/{memory_id}`	Shows the memory information about a specific NUMA node of a specific host.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

6.7.2.1. Request

This table shows the URI parameters for the show memory request:

Name	Type	Description
`{memory_id}`	UUID	The unique identifier of a memory area.

This operation does not accept a request body.

6.7.2.2. Response

This table shows the body parameters for the show memory response:

Name	Type	Description
`ihost_uuid`	UUID *(Optional)*	The UUID of the host.
`numa_node`	Integer *(Optional)*	The NUMA node number.
`inode_uuid`	UUID *(Optional)*	The UUID of the NUMA node.
`platform_reserved_mib`	Integer *(Optional)*	Memory reserved for the Kernel and Cloud Platform Services, in MiBs.
`memtotal_mib`	Integer *(Optional)*	Total memory reserved for the hosted Virtual Machines, in MiBs.
`memavail_mib`	Integer *(Optional)*	Free / available memory from the total memory reserved for the hosted Virtual Machines, in MiBs.
`hugepages_configured`	Boolean *(Optional)*	Boolean indicating whether huge page memory is configured or not.
`avs_hugepages_size_mib`	Integer *(Optional)*	The size of an Accelerated Virtual Switch (AVS) huge page, in MiBs.
`avs_hugepages_nr`	Integer *(Optional)*	The total number of Accelerated Virtual Switch (AVS) huge pages.
`avs_hugepages_avail`	Integer *(Optional)*	The free / available Accelerated Virtual Switch (AVS) huge pages.
`vm_hugepages_nr_1G`	Integer	The total number of Virtual Machine 1G huge pages.

Name	Type	Description
	(Optional)	
vm_hugepages_avail_1G	Integer (Optional)	The free / available Virtual Machine 1G huge pages.
vm_hugepages_nr_1G_pending	Integer (Optional)	If not null, the pending configured number of Virtual Machine 1G huge pages.
vm_hugepages_nr_2M	Integer (Optional)	The total number of Virtual Machine 2M huge pages.
vm_hugepages_avail_2M	Integer (Optional)	The free / available Virtual Machine 2M huge pages.
vm_hugepages_nr_2M_pending	Integer (Optional)	If not null, the pending configured number of Virtual Machine 2M huge pages.
uuid	UUID (Optional)	The universally unique identifier for this object.
links	List (Optional)	For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The following types of link relations are associated with resources: a self link containing a versioned link to the resource, and a bookmark link containing a permanent link to a resource that is appropriate for long term storage.
created_at	DateTime (Optional)	The time when the object was created.
updated_at	DateTime (Optional)	The time when the object was last updated.

Example 6.27. Show memory: JSON response

```
{
    "avs_hugepages_nr" : 1,
    "hugepages_configured" : "True",
    "vm_hugepages_nr_1G_pending" : null,
    "memavail_mib" : 24082,
    "uuid" : "85df5109-77d9-4335-9181-0efa82c98dcc",
    "ihost_uuid" : "afecdcfb-2954-498d-88bf-d1385b00f34d",
    "created_at" : "2015-04-06T20:27:50.182764+00:00",
    "avs_hugepages_size_mib" : 1024,
    "vm_hugepages_avail_1G" : 0,
    "capabilities" : {},
    "links" : [
        {
            "rel" : "self",
            "href" : "http://128.224.150.54:6385/v1/imemorys/
85df5109-77d9-4335-9181-0efa82c98dcc"
        },
        {
            "rel" : "bookmark",
            "href" : "http://128.224.150.54:6385/imemorys/
85df5109-77d9-4335-9181-0efa82c98dcc"
        }
    ],
    "vm_hugepages_nr_2M_pending" : null,
    "avs_hugepages_reqd" : null,
```

```
    "vm_hugepages_avail_2M" : 12041,
    "avs_hugepages_avail" : 0,
    "numa_node" : 1,
    "vm_hugepages_nr_1G" : 0,
    "updated_at" : "2015-04-08T11:33:25.280674+00:00",
    "platform_reserved_mib" : 2000,
    "memtotal_mib" : 29202,
    "vm_hugepages_nr_2M" : 14089,
    "inode_uuid" : "67d3c9a0-57b2-4532-b7ea-f1cd16a3b349"
}
```

This operation does not return a response body.

6.7.3. Modify memory

Method	URI	Description
PATCH	`/v1/imemorys/{memory_id}`	Modifies the memory information about a specific NUMA node of a specific host..

Normal response codes: 200

Error response codes: badMediaType (415)

6.7.3.1. Request

This table shows the URI parameters for the modify memory request:

Name	Type	Description
`{memory_id}`	UUID	The unique identifier of a memory area.

This table shows the body parameters for the modify memory request:

Name	Type	Description
`vm_hugepages_nr_1G_pending`	Integer *(Optional)*	If not null, the pending configured number of Virtual Machine 1G huge pages.
`vm_hugepages_nr_2M_pending`	Integer *(Optional)*	If not null, the pending configured number of Virtual Machine 2M huge pages.

Example 6.28. Modify memory: JSON request

```
[
  {
    "path": "/vm_hugepages_nr_1G_pending",
    "value": "100",
    "op": "replace"
  },
  {
    "path": "/vm_hugepages_nr_2M_pending",
    "value": "50",
    "op": "replace"
  }
]
```

This operation does not accept a request body.

6.7.3.2. Response

This table shows the body parameters for the modify memory response:

Name	Type	Description
`ihost_uuid`	UUID *(Optional)*	The UUID of the host.
`numa_node`	Integer *(Optional)*	The NUMA node number.
`inode_uuid`	UUID	The UUID of the NUMA node.

Name	Type	Description
	(Optional)	
`platform_reserved_mib`	Integer *(Optional)*	Memory reserved for the Kernel and Cloud Platform Services, in MiBs.
`memtotal_mib`	Integer *(Optional)*	Total memory reserved for the hosted Virtual Machines, in MiBs.
`memavail_mib`	Integer *(Optional)*	Free / available memory from the total memory reserved for the hosted Virtual Machines, in MiBs.
`hugepages_configured`	Boolean *(Optional)*	Boolean indicating whether huge page memory is configured or not.
`avs_hugepages_size_mib`	Integer *(Optional)*	The size of an Accelerated Virtual Switch (AVS) huge page, in MiBs.
`avs_hugepages_nr`	Integer *(Optional)*	The total number of Accelerated Virtual Switch (AVS) huge pages.
`avs_hugepages_avail`	Integer *(Optional)*	The free / available Accelerated Virtual Switch (AVS) huge pages.
`vm_hugepages_nr_1G`	Integer *(Optional)*	The total number of Virtual Machine 1G huge pages.
`vm_hugepages_avail_1G`	Integer *(Optional)*	The free / available Virtual Machine 1G huge pages.
`vm_hugepages_nr_1G_pending`	Integer *(Optional)*	If not null, the pending configured number of Virtual Machine 1G huge pages.
`vm_hugepages_nr_2M`	Integer *(Optional)*	The total number of Virtual Machine 2M huge pages.
`vm_hugepages_avail_2M`	Integer *(Optional)*	The free / available Virtual Machine 2M huge pages.
`vm_hugepages_nr_2M_pending`	Integer *(Optional)*	If not null, the pending configured number of Virtual Machine 2M huge pages.

Example 6.29. Modify memory: JSON response

```
{
   "avs_hugepages_nr" : 1,
   "hugepages_configured" : "True",
   "vm_hugepages_nr_1G_pending" : null,
   "memavail_mib" : 24082,
   "uuid" : "85df5109-77d9-4335-9181-0efa82c98dcc",
   "ihost_uuid" : "afecdcfb-2954-498d-88bf-d1385b00f34d",
   "created_at" : "2015-04-06T20:27:50.182764+00:00",
   "avs_hugepages_size_mib" : 1024,
   "vm_hugepages_avail_1G" : 0,
   "capabilities" : {},
   "links" : [
      {
         "rel" : "self",
         "href" : "http://128.224.150.54:6385/v1/imemorys/
85df5109-77d9-4335-9181-0efa82c98dcc"
```

```
        },
        {
          "rel" : "bookmark",
          "href" : "http://128.224.150.54:6385/imemorys/
85df5109-77d9-4335-9181-0efa82c98dcc"
        }
    ],
    "vm_hugepages_nr_2M_pending" : null,
    "avs_hugepages_reqd" : null,
    "vm_hugepages_avail_2M" : 12041,
    "avs_hugepages_avail" : 0,
    "numa_node" : 1,
    "vm_hugepages_nr_1G" : 0,
    "updated_at" : "2015-04-08T11:33:25.280674+00:00",
    "platform_reserved_mib" : 2000,
    "memtotal_mib" : 29202,
    "vm_hugepages_nr_2M" : 14089,
    "inode_uuid" : "67d3c9a0-57b2-4532-b7ea-f1cd16a3b349"
}
```

This operation does not return a response body.

6.8. Disks

Method	URI	Description
GET	/v1/ihosts/{host_id}/idisks	Lists all physical disks of a host.
GET	/v1/idisks/{disk_id}	Shows detailed information about a specific physical disk.

6.8.1. List disks

Method	URI	Description
GET	`/v1/ihosts/{host_id}/idisks`	Lists all physical disks of a host.

Insert extra description here, if required.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

6.8.1.1. Request

This table shows the URI parameters for the list disks request:

Name	Type	Description
`{host_id}`	UUID	The unique identifier of an existing host.

This operation does not accept a request body.

6.8.1.2. Response

This table shows the body parameters for the list disks response:

Name	Type	Description
`idisks`	List *(Optional)*	The list of physical disk entities.
`capabilities`	String *(Optional)*	Additional capabilities info about the disk.
`device_node`	String *(Optional)*	The device node of the disk.
`device_num`	String *(Optional)*	The device number of the disk.
`device_type`	String *(Optional)*	The disk device type.
`ihost_uuid`	UUID *(Optional)*	The host UUID that the disk belongs to.
`inode_uuid`	UUID *(Optional)*	The node UUID that the disk belongs to.
`istor_uuid`	String *(Optional)*	The logical storage volume that this disk belongs to.
`ipv_uuid`	String *(Optional)*	The LVM physical volume that this disk belongs to.

Name	Type	Description
serial_id	String *(Optional)*	The serial id or number of the disk.
size_mib	String *(Optional)*	The size of the disk in MiBytes.

Example 6.30. List disks: JSON response

```
{
    "idisks": [
        {
            "ipv_uuid": null,
            "uuid": "f4eb1e13-e0ac-4203-91ac-d90a45c29a25",
            "links": [
                {
                    "href": "http://10.10.10.2:6385/v1/idisks/f4eb1e13-
e0ac-4203-91ac-d90a45c29a25",
                    "rel": "self"
                },
                {
                    "href": "http://10.10.10.2:6385/idisks/f4eb1e13-
e0ac-4203-91ac-d90a45c29a25",
                    "rel": "bookmark"
                }
            ],
            "inode_uuid": null,
            "created_at": "2015-03-10T21:06:20.002710+00:00",
            "updated_at": "2015-03-11T04:29:16.130225+00:00",
            "device_node": "/dev/sda",
            "ihost_uuid": "1185677a-9bdb-44f4-9d9c-20d9bc5e907d",
            "serial_id": "VB6ec2d829-c6fd2f0d",
            "device_type": "disk",
            "istor_uuid": null,
            "capabilities": {
                "stor_function": "rootfs",
                "model_num": "VBOX_HARDDISK",
                "pv_dev": null
            },
            "size_mib": 51200,
            "device_num": 2048
        },
        {
            "ipv_uuid": null,
            "uuid": "38dea46b-7a02-4682-bc55-e773c0b20bf1",
            "links": [
                {
                    "href": "http://10.10.10.2:6385/v1/idisks/
38dea46b-7a02-4682-bc55-e773c0b20bf1",
                    "rel": "self"
                },
                {
                    "href": "http://10.10.10.2:6385/idisks/38dea46b-7a02-4682-
bc55-e773c0b20bf1",
                    "rel": "bookmark"
                }
            ],
            "inode_uuid": null,
            "created_at": "2015-03-10T21:06:20.008166+00:00",
```

```
            "updated_at": "2015-03-11T04:29:16.453000+00:00",
            "device_node": "/dev/sdb",
            "ihost_uuid": "1185677a-9bdb-44f4-9d9c-20d9bc5e907d",
            "serial_id": "VBc250d829-d55c6477",
            "device_type": "disk",
            "istor_uuid": null,
            "capabilities": {
                "device_function": "cinder_device",
                "model_num": "VBOX_HARDDISK"
            },
            "size_mib": 10240,
            "device_num": 2064
        }
    ]
}
```

This operation does not return a response body.

6.8.2. Show disk

Method	URI	Description
GET	`/v1/idisks/{disk_id}`	Shows detailed information about a specific physical disk.

Insert extra description here, if required.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

6.8.2.1. Request

This table shows the URI parameters for the show disk request:

Name	Type	Description
`{disk_id}`	UUID	The unique identifier of a physical disk.

This operation does not accept a request body.

6.8.2.2. Response

This table shows the body parameters for the show disk response:

Name	Type	Description
`capabilities`	String *(Optional)*	Additional capabilities info about the disk.
`device_node`	String *(Optional)*	The device node of the disk.
`device_num`	String *(Optional)*	The device number of the disk.
`device_type`	String *(Optional)*	The disk device type.
`ihost_uuid`	UUID *(Optional)*	The host UUID that the disk belongs to.
`inode_uuid`	UUID *(Optional)*	The node UUID that the disk belongs to.
`istor_uuid`	String *(Optional)*	The logical storage volume that this disk belongs to.
`ipv_uuid`	String *(Optional)*	The LVM physical volume that this disk belongs to.
`serial_id`	String *(Optional)*	The serial id or number of the disk.
`size_mib`	String	The size of the disk in MiBytes.

Name	Type	Description
	(Optional)	

Example 6.31. Show disk: JSON response

```json
{
    "ipv_uuid": null,
    "uuid": "f4eb1e13-e0ac-4203-91ac-d90a45c29a25",
    "links": [
        {
            "href": "http://10.10.10.2:6385/v1/idisks/f4eb1e13-e0ac-4203-91ac-d90a45c29a25",
            "rel": "self"
        },
        {
            "href": "http://10.10.10.2:6385/idisks/f4eb1e13-e0ac-4203-91ac-d90a45c29a25",
            "rel": "bookmark"
        }
    ],
    "inode_uuid": null,
    "created_at": "2015-03-10T21:06:20.002710+00:00",
    "updated_at": "2015-03-11T04:29:16.130225+00:00",
    "device_node": "/dev/sda",
    "ihost_uuid": "1185677a-9bdb-44f4-9d9c-20d9bc5e907d",
    "serial_id": "VB6ec2d829-c6fd2f0d",
    "device_type": "disk",
    "istor_uuid": null,
    "capabilities": {
        "pv_dev": null,
        "model_num": "VBOX_HARDDISK",
        "stor_function": "rootfs"
    },
    "size_mib": 51200,
    "device_num": 2048
}
```

This operation does not return a response body.

6.9. Volumes

Method	URI	Description
GET	`/v1/ihosts/{host_id}/istors`	Lists all storage disk volumes of a host.
GET	`/v1/istors/{volume_id}`	Shows detailed information about a specific storage disk volume.
POST	`/v1/ihosts/{host_id}/istors`	Creates a storage disk volume on a specific host.

6.9.1. List volumes

Method	URI	Description
GET	`/v1/ihosts/{host_id}/istors`	Lists all storage disk volumes of a host.

Insert extra description here, if required.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

6.9.1.1. Request

This table shows the URI parameters for the list volumes request:

Name	Type	Description
`{host_id}`	UUID	The unique identifier of an existing host.

This operation does not accept a request body.

6.9.1.2. Response

This table shows the body parameters for the list volumes response:

Name	Type	Description
`ivolumes`	List (Optional)	The list of disk volume entities.
`capabilities`	String (Optional)	Additional capabilities info about the storage volume.
`function`	String (Optional)	The storage volume function. e.g. "osd" (object storage daemon) for ceph.
`ihost_uuid`	UUID (Optional)	The host UUID that the storage volume belongs to.
`isystem_uuid`	UUID (Optional)	The System UUID which the storage volume belongs to.
`osdid`	String (Optional)	The object storage daemon identifier of the storage volume.
`state`	String (Optional)	The state info of the storage volume.

Example 6.32. List volumes: JSON response

```
{
    "istors":[
        {
```

```
          "function":"osd",
          "uuid":"31c7a5a3-9154-462a-9ea3-ab5c5e0d06e0",
          "links":[
              {
                  "href":"http://192.168.204.2:6385/v1/istors/
31c7a5a3-9154-462a-9ea3-ab5c5e0d06e0",
                  "rel":"self"
              },
              {
                  "href":"http://192.168.204.2:6385/istors/
31c7a5a3-9154-462a-9ea3-ab5c5e0d06e0",
                  "rel":"bookmark"
              }
          ],
          "created_at":"2014-10-01T21:41:23.973344+00:00",
          "updated_at":"2014-10-01T21:41:24.129134+00:00",
          "capabilities":{

          },
          "ihost_uuid":"42d72247-e0e3-4a5a-8cb1-40bbee52c8db",
          "state":null,
          "osdid":2
      }
   ]
}
```

This operation does not return a response body.

6.9.2. Show volume

Method	URI	Description
GET	`/v1/istors/{volume_id}`	Shows detailed information about a specific storage disk volume.

Insert extra description here, if required.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

6.9.2.1. Request

This table shows the URI parameters for the show volume request:

Name	Type	Description
`{volume_id}`	UUID	The unique identifier of an existing storage volume.

This operation does not accept a request body.

6.9.2.2. Response

This table shows the body parameters for the show volume response:

Name	Type	Description
`capabilities`	String *(Optional)*	Additional capabilities info about the storage volume.
`function`	String *(Optional)*	The storage volume function. e.g. "osd" (object storage daemon) for ceph.
`ihost_uuid`	UUID *(Optional)*	The host UUID that the storage volume belongs to.
`isystem_uuid`	UUID *(Optional)*	The System UUID which the storage volume belongs to.
`osdid`	String *(Optional)*	The object storage daemon identifier of the storage volume.
`state`	String *(Optional)*	The state info of the storage volume.

Example 6.33. Show volume: JSON response

```
{
    "istors":[
        {
            "function":"osd",
            "uuid":"31c7a5a3-9154-462a-9ea3-ab5c5e0d06e0",
```

```
            "links":[
                {
                    "href":"http://192.168.204.2:6385/v1/istors/
31c7a5a3-9154-462a-9ea3-ab5c5e0d06e0",
                    "rel":"self"
                },
                {
                    "href":"http://192.168.204.2:6385/istors/
31c7a5a3-9154-462a-9ea3-ab5c5e0d06e0",
                    "rel":"bookmark"
                }
            ],
            "created_at":"2014-10-01T21:41:23.973344+00:00",
            "updated_at":"2014-10-01T21:41:24.129134+00:00",
            "capabilities":{

            },
            "ihost_uuid":"42d72247-e0e3-4a5a-8cb1-40bbee52c8db",
            "state":null,
            "osdid":2
        }
    ]
}
```

This operation does not return a response body.

6.9.3. Create volume

Method	URI	Description
POST	`/v1/ihosts/{host_id}/istors`	Creates a storage disk volume on a specific host.

PREREQUISITE: 'ceph' must be configured during the initial installation.

Normal response codes: 200

Error response codes: badMediaType (415)

6.9.3.1. Request

This table shows the URI parameters for the create volume request:

Name	Type	Description
`{host_id}`	UUID	The unique identifier of an existing host.

This table shows the body parameters for the create volume request:

Name	Type	Description
`function`	String *(Optional)*	This parameter specifies the storage volume function. Valid values are (is): `osd`
`ihost_uuid`	String *(Optional)*	This parameter specifies the storage host uuid.
`idisk_uuid`	String *(Optional)*	This parameter specifies the storage disk uuid.

Example 6.34. Create volume: JSON request

```
{
    "function":"osd",
    "ihost_uuid":"42d72247-e0e3-4a5a-8cb1-40bbee52c8db",
    "idisk_uuid":"4da10410-2959-46df-b571-04e954c0e115"
}
```

This operation does not accept a request body.

6.9.3.2. Response

This table shows the body parameters for the create volume response:

Name	Type	Description
`capabilities`	String *(Optional)*	Additional capabilities info about the storage volume.
`function`	String *(Optional)*	The storage volume function. e.g. "osd" (object storage daemon) for ceph.
`ihost_uuid`	UUID *(Optional)*	The host UUID that the storage volume belongs to.
`isystem_uuid`	UUID	The System UUID which the storage volume belongs to.

Name	Type	Description
	(Optional)	
`osdid`	String *(Optional)*	The object storage daemon identifier of the storage volume.
`state`	String *(Optional)*	The state info of the storage volume.

Example 6.35. Create volume: JSON response

```
{
   "function":"osd",
   "uuid":"31c7a5a3-9154-462a-9ea3-ab5c5e0d06e0",
   "links":[
      {
         "href":"http://192.168.204.2:6385/v1/istors/31c7a5a3-9154-462a-9ea3-
ab5c5e0d06e0",
         "rel":"self"
      },
      {
         "href":"http://192.168.204.2:6385/istors/31c7a5a3-9154-462a-9ea3-
ab5c5e0d06e0",
         "rel":"bookmark"
      }
   ],
   "idisks":[
      {
         "href":"http://192.168.204.2:6385/v1/istors/31c7a5a3-9154-462a-9ea3-
ab5c5e0d06e0/idisks",
         "rel":"self"
      },
      {
         "href":"http://192.168.204.2:6385/istors/31c7a5a3-9154-462a-9ea3-
ab5c5e0d06e0/idisks",
         "rel":"bookmark"
      }
   ],
   "created_at":"2014-10-01T21:41:23+00:00",
   "updated_at":null,
   "idisk_uuid":"4da10410-2959-46df-b571-04e954c0e115",
   "ihost_uuid":"42d72247-e0e3-4a5a-8cb1-40bbee52c8db",
   "state":null,
   "capabilities":{

   },
   "osdid":2
}
```

This operation does not return a response body.

6.10. Volume Groups

These APIs allow the creation, deletion, and displaying of LVM volume groups.

Method	URI	Description
GET	`/v1/ihosts/{host_id}/ilvgs`	Lists all LVM volume groups of a host.

Method	URI	Description
GET	`/v1/ilvgs/{volumegroup_id}`	Shows detailed information about a specific LVM volume group.
POST	`/v1/ilvgs`	Creates an LVM volume group on a specific host.
DELETE	`/v1/ilvgs/{volumegroup_id}`	Deletes a specific LVM volume group.

6.10.1. List volume groups

Method	URI	Description
GET	`/v1/ihosts/{host_id}/ilvgs`	Lists all LVM volume groups of a host.

This will list all the LVM volume groups for a given host. This functionality is currently only enabled for hosts with a compute personality. Any other hosts with a personality type other than compute will result in no volume groups being reported on the host.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404)

6.10.1.1. Request

This table shows the URI parameters for the list volume groups request:

Name	Type	Description
`{host_id}`	UUID	The unique identifier of an existing host.

This operation does not accept a request body.

6.10.1.2. Response

This table shows the body parameters for the list volume groups response:

Name	Type	Description
`ilvgs`	List *(Optional)*	The list of volume group entities.
`capabilities`	String *(Optional)*	Additional capabilities info about the volume group.
`vg_state`	String *(Optional)*	This is the state of the volume group which is one of the following: unprovisioned, adding, provisioned, or removing.
`lvm_vg_name`	String *(Optional)*	This is the LVM volume group name as retrieved from the vgdisplay command on the host.
`lvm_vg_uuid`	String *(Optional)*	This is the LVM generated volume group UUID as retrieved from the vgdisplay command on the host.
`lvm_vg_access`	String *(Optional)*	This is the LVM generated volume group access status as retrieved from the vgdisplay command on the host.
`lvm_max_lv`	String *(Optional)*	This is the LVM generated max number of logical volumes allowed as retrieved from the vgdisplay command on the host.
`lvm_cur_lv`	String *(Optional)*	This is the LVM generated current number of logical volumes as retrieved from the vgdisplay command on the host.
`lvm_max_pv`	String	This is the LVM generated max number of physical volumes allowed as retrieved from the vgdisplay command on the host.

Name	Type	Description
	(Optional)	
lvm_cur_pv	String *(Optional)*	This is the LVM generated current number of physical volumes as retrieved from the vgdisplay command on the host.
lvm_vg_size	String *(Optional)*	This is the LVM generated volume group size in bytes as retrieved from the vgdisplay command on the host.
lvm_vg_total_pe	String *(Optional)*	This is the LVM generated total number of physical extents within the volume group as retrieved from the vgdisplay command on the host.
lvm_vg_free_pe	String *(Optional)*	This is the LVM generated number of physical extents not allocated within the volume group as retrieved from the vgdisplay command on the host.
ihost_uuid	UUID *(Optional)*	The UUID of the host containing the port.

Example 6.36. List volume groups: JSON response

```
{
    "ilvgs": [
        {
            "lvm_vg_access": "wz--n-",
            "lvm_vg_size": 7310671872,
            "lvm_max_lv": 0,
            "lvm_vg_free_pe": 1743,
            "uuid": "039de9ef-b1db-4c31-9072-add0f888b8b9",
            "links": [
                {
                    "href": "http://10.10.10.2:6385/v1/ilvgs/039de9ef-
b1db-4c31-9072-add0f888b8b9",
                    "rel": "self"
                },
                {
                    "href": "http://10.10.10.2:6385/ilvgs/039de9ef-
b1db-4c31-9072-add0f888b8b9",
                    "rel": "bookmark"
                }
            ],
            "lvm_cur_lv": 0,
            "created_at": "2015-03-11T02:46:55.730611+00:00",
            "lvm_max_pv": 0,
            "updated_at": "2015-03-11T02:50:57.361006+00:00",
            "capabilities": {},
            "vg_state": "provisioned",
            "ihost_uuid": "1ef159f8-0192-4879-a08e-f60328486e34",
            "lvm_cur_pv": 1,
            "lvm_vg_uuid": "u7NzxA-1LeR-G88h-3lMk-eFvo-YnL8-HT9SEP",
            "lvm_vg_total_pe": 1743,
            "lvm_vg_name": "nova-local"
        }
    ]
}
```

This operation does not return a response body.

6.10.2. Show volume group

Method	URI	Description
GET	`/v1/ilvgs/{volumegroup_id}`	Shows detailed information about a specific LVM volume group.

This will show detailed information about a specific LVM volume group. This functionality is currently only enabled for hosts with a compute personality. Any other hosts with a personality type other than compute will result in no volume groups being reported on the host. In addition, the only allowed volume group name at this time is called: "nova-local".

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404)

6.10.2.1. Request

This table shows the URI parameters for the show volume group request:

Name	Type	Description
`{volumegroup_id}`	UUID	The unique identifier of an existing LVM volume group.

This operation does not accept a request body.

6.10.2.2. Response

This table shows the body parameters for the show volume group response:

Name	Type	Description
`capabilities`	String *(Optional)*	Additional capabilities info about the volume group.
`vg_state`	String *(Optional)*	This is the state of the volume group which is one of the following: unprovisioned, adding, provisioned, or removing.
`lvm_vg_name`	String *(Optional)*	This is the LVM volume group name as retrieved from the vgdisplay command on the host.
`lvm_vg_uuid`	String *(Optional)*	This is the LVM generated volume group UUID as retrieved from the vgdisplay command on the host.
`lvm_vg_access`	String *(Optional)*	This is the LVM generated volume group access status as retrieved from the vgdisplay command on the host.
`lvm_max_lv`	String *(Optional)*	This is the LVM generated max number of logical volumes allowed as retrieved from the vgdisplay command on the host.
`lvm_cur_lv`	String *(Optional)*	This is the LVM generated current number of logical volumes as retrieved from the vgdisplay command on the host.
`lvm_max_pv`	String *(Optional)*	This is the LVM generated max number of physical volumes allowed as retrieved from the vgdisplay command on the host.

Name	Type	Description
lvm_cur_pv	String *(Optional)*	This is the LVM generated current number of physical volumes as retrieved from the vgdisplay command on the host.
lvm_vg_size	String *(Optional)*	This is the LVM generated volume group size in bytes as retrieved from the vgdisplay command on the host.
lvm_vg_total_pe	String *(Optional)*	This is the LVM generated total number of physical extents within the volume group as retrieved from the vgdisplay command on the host.
lvm_vg_free_pe	String *(Optional)*	This is the LVM generated number of physical extents not allocated within the volume group as retrieved from the vgdisplay command on the host.
ihost_uuid	UUID *(Optional)*	The UUID of the host containing the port.

Example 6.37. Show volume group: JSON response

```
{
    "lvm_vg_access": "wz--n-",
    "lvm_vg_size": 7310671872,
    "lvm_max_lv": 0,
    "lvm_vg_free_pe": 1743,
    "uuid": "039de9ef-b1db-4c31-9072-add0f888b8b9",
    "links": [
        {
            "href": "http://10.10.10.2:6385/v1/ilvgs/039de9ef-b1db-4c31-9072-
add0f888b8b9",
            "rel": "self"
        },
        {
            "href": "http://10.10.10.2:6385/ilvgs/039de9ef-b1db-4c31-9072-
add0f888b8b9",
            "rel": "bookmark"
        }
    ],
    "lvm_cur_lv": 0,
    "created_at": "2015-03-11T02:46:55.730611+00:00",
    "lvm_max_pv": 0,
    "updated_at": "2015-03-11T02:50:57.361006+00:00",
    "capabilities": {},
    "vg_state": "provisioned",
    "ihost_uuid": "1ef159f8-0192-4879-a08e-f60328486e34",
    "ipvs": [
        {
            "href": "http://10.10.10.2:6385/v1/ilvgs/039de9ef-b1db-4c31-9072-
add0f888b8b9/ipvs",
            "rel": "self"
        },
        {
            "href": "http://10.10.10.2:6385/ilvgs/039de9ef-b1db-4c31-9072-
add0f888b8b9/ipvs",
            "rel": "bookmark"
        }
    ],
    "lvm_cur_pv": 1,
    "lvm_vg_uuid": "u7NzxA-1LeR-G88h-3lMk-eFvo-YnL8-HT9SEP",
    "lvm_vg_total_pe": 1743,
    "lvm_vg_name": "nova-local"
```

```
}
```

This operation does not return a response body.

6.10.3. Create a volume Group

Method	URI	Description
POST	`/v1/ilvgs`	Creates an LVM volume group on a specific host.

This will create an LVM volume group on the specified host. This functionality is currently only enabled for hosts with a compute personality. In addition, the volume group name is limited to the name: "nova-local". In order to add a volume group, the host must be locked.

Normal response codes: 200

Error response codes: badMediaType (415)

6.10.3.1. Request

This table shows the body parameters for the create a volume group request:

Name	Type	Description
`lvm_vg_name`	String *(Optional)*	This parameter specifies the volume group name. Valid values are (is): `nova-local`
`ihost_uuid`	String *(Optional)*	This parameter specifies the compute host uuid.

Example 6.38. Create a volume Group: JSON request

```
{
    "lvm_vg_name":"nova-local",
    "ihost_uuid":"a0f0a6d5-75ad-4769-8e0e-3a7c7c0ce783"
}
```

This operation does not accept a request body.

6.10.3.2. Response

This table shows the body parameters for the create a volume group response:

Name	Type	Description
`capabilities`	String *(Optional)*	Additional capabilities info about the storage volume.
`function`	String *(Optional)*	The storage volume function. e.g. "osd" (object storage daemon) for ceph.
`ihost_uuid`	UUID *(Optional)*	The host UUID that the storage volume belongs to.
`isystem_uuid`	UUID *(Optional)*	The System UUID which the storage volume belongs to.
`osdid`	String *(Optional)*	The object storage daemon identifier of the storage volume.
`state`	String	The state info of the storage volume.

Name	Type	Description
	(Optional)	

Example 6.39. Create a volume Group: JSON response

```
{
    "lvm_vg_access": null,
    "lvm_vg_size": 0,
    "lvm_max_lv": 0,
    "lvm_vg_free_pe": 0,
    "uuid": "11ac6dfc-a5ea-4cc9-a0c9-50afb13f7b24",
    "links": [
        {
            "href": "http://10.10.10.2:6385/v1/ilvgs/11ac6dfc-a5ea-4cc9-
a0c9-50afb13f7b24",
            "rel": "self"
        },
        {
            "href": "http://10.10.10.2:6385/ilvgs/11ac6dfc-a5ea-4cc9-
a0c9-50afb13f7b24",
            "rel": "bookmark"
        }
    ],
    "lvm_cur_lv": 0,
    "created_at": "2015-03-11T04:52:32.007904+00:00",
    "lvm_max_pv": 0,
    "updated_at": null,
    "capabilities": {},
    "vg_state": "adding",
    "ihost_uuid": "a0f0a6d5-75ad-4769-8e0e-3a7c7c0ce783",
    "ipvs": [
        {
            "href": "http://10.10.10.2:6385/v1/ilvgs/11ac6dfc-a5ea-4cc9-
a0c9-50afb13f7b24/ipvs",
            "rel": "self"
        },
        {
            "href": "http://10.10.10.2:6385/ilvgs/11ac6dfc-a5ea-4cc9-
a0c9-50afb13f7b24/ipvs",
            "rel": "bookmark"
        }
    ],
    "lvm_cur_pv": 0,
    "lvm_vg_uuid": null,
    "lvm_vg_total_pe": 0,
    "lvm_vg_name": "nova-local"
}
```

This operation does not return a response body.

6.10.4. Delete a volume group

Method	URI	Description
DELETE	`/v1/ilvgs/{volumegroup_id}`	Deletes a specific LVM volume group.

In order to delete an LVM volume group, the host must be locked. All physical volumes assigned to the volume group will also be deleted.

Normal response codes: 204

6.10.4.1. Request

This table shows the URI parameters for the delete a volume group request:

Name	Type	Description
`{volumegroup_id}`	UUID	The unique identifier of an existing LVM volume group.

This operation does not accept a request body.

6.10.4.2. Response

This operation does not return a response body.

6.11. Physical Volumes

These APIs allow the creation, deletion, and displaying of LVM physical volumes.

Method	URI	Description
GET	`/v1/ihosts/{host_id}/ipvs`	Lists all LVM physical volumes of a host.
GET	`/v1/ipvs/{physicalvolume_id}`	Shows detailed information about a specific LVM physical volume.
POST	`/v1/ipvs`	Creates an LVM physical volume on a specific host.
DELETE	`/v1/ipvs/{physicalvolume_id}`	Deletes a specific LVM physical volume.

6.11.1. List volumes

Method	URI	Description
GET	`/v1/ihosts/{host_id}/ipvs`	Lists all LVM physical volumes of a host.

This will list all the LVM physical volumes defined on the given host. This functionality is currently only enabled for hosts with a compute personality. Any other hosts with a personality type other than compute will result in no physical volumes being reported on the host. A physical volume can be a pre-defined root disk partition or a entire extra disk.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404)

6.11.1.1. Request

This table shows the URI parameters for the list volumes request:

Name	Type	Description
`{host_id}`	UUID	The unique identifier of an existing host.

This operation does not accept a request body.

6.11.1.2. Response

This table shows the body parameters for the list volumes response:

Name	Type	Description
`ivolumes`	List *(Optional)*	The list of physical volume entities.
`capabilities`	String *(Optional)*	Additional capabilities info about the volume group.
`pv_state`	String *(Optional)*	This is the state of the physical volume. It has one of the following values: unprovisioned, adding, provisioned, or removing.
`pv_type`	String *(Optional)*	This is the type of physical volume that is allocated. This will have the value of disk or partition.
`idisk_uuid`	String *(Optional)*	This is the UUID of the disk that is associated with this physical volume.
`idisk_device_node`	String *(Optional)*	This is the device node name associated with the physical partition.
`lvm_pv_name`	String *(Optional)*	This is the physical volume name as retrieved from the pvdisplay command on the host.
`lvm_vg_name`	String *(Optional)*	This is the name of the volume group that this physical volume belongs as retrieved from the pvdisplay command on the host.

Name	Type	Description
lvm_pv_uuid	String *(Optional)*	This is the LVM generated UUID for the physical volume as retrieved from the pvdisplay command on the host.
lvm_pv_size	String *(Optional)*	This is the LVM generated size in bytes of the physical volume as re-trieved from the pvdisplay command on the host.
lvm_pe_total	String *(Optional)*	This is the LVM generated total number of physical extents associated with the physical volume as retrieved from the pvdisplay command on the host.
lvm_pe_alloced	String *(Optional)*	This is the LVM generated number of allocated physical extents asso-ciated with the physical volume as retrieved from the pvdisplay com-mand on the host.
ihost_uuid	UUID *(Optional)*	The UUID of the host containing the port.
ilvg_uuid	UUID *(Optional)*	The UUID of the volume group containing the physical volume.

Example 6.40. List volumes: JSON response

```
{
   "istors":[
      {
         "function":"osd",
         "uuid":"31c7a5a3-9154-462a-9ea3-ab5c5e0d06e0",
         "links":[
            {
               "href":"http://192.168.204.2:6385/v1/istors/
31c7a5a3-9154-462a-9ea3-ab5c5e0d06e0",
               "rel":"self"
            },
            {
               "href":"http://192.168.204.2:6385/istors/
31c7a5a3-9154-462a-9ea3-ab5c5e0d06e0",
               "rel":"bookmark"
            }
         ],
         "created_at":"2014-10-01T21:41:23.973344+00:00",
         "updated_at":"2014-10-01T21:41:24.129134+00:00",
         "capabilities":{

         },
         "ihost_uuid":"42d72247-e0e3-4a5a-8cb1-40bbee52c8db",
         "state":null,
         "osdid":2
      }
   ]
}
```

This operation does not return a response body.

6.11.2. Show physical volume

Method	URI	Description
GET	`/v1/ipvs/{physicalvolume_id}`	Shows detailed information about a specific LVM physical volume.

This will show detailed information about a specific LVM physical volume.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

6.11.2.1. Request

This table shows the URI parameters for the show physical volume request:

Name	Type	Description
`{physicalvolume_id}`	UUID	The unique identifier of an existing LVM physical volume.

This operation does not accept a request body.

6.11.2.2. Response

This table shows the body parameters for the show physical volume response:

Name	Type	Description
`capabilities`	String *(Optional)*	Additional capabilities info about the volume group.
`pv_state`	String *(Optional)*	This is the state of the physical volume. It has one of the following values: unprovisioned, adding, provisioned, or removing.
`pv_type`	String *(Optional)*	This is the type of physical volume that is allocated. This will have the value of disk or partition.
`idisk_uuid`	String *(Optional)*	This is the UUID of the disk that is associated with this physical volume.
`idisk_device_node`	String *(Optional)*	This is the device node name associated with the physical partition.
`lvm_pv_name`	String *(Optional)*	This is the physical volume name as retrieved from the pvdisplay command on the host.
`lvm_vg_name`	String *(Optional)*	This is the name of the volume group that this physical volume belongs as retrieved from the pvdisplay command on the host.
`lvm_pv_uuid`	String *(Optional)*	This is the LVM generated UUID for the physical volume as retrieved from the pvdisplay command on the host.
`lvm_pv_size`	String *(Optional)*	This is the LVM generated size in bytes of the physical volume as retrieved from the pvdisplay command on the host.

Name	Type	Description
lvm_pe_total	String (Optional)	This is the LVM generated total number of physical extents associated with the physical volume as retrieved from the pvdisplay command on the host.
lvm_pe_alloced	String (Optional)	This is the LVM generated number of allocated physical extents associated with the physical volume as retrieved from the pvdisplay command on the host.
ihost_uuid	UUID (Optional)	The UUID of the host containing the port.
ilvg_uuid	UUID (Optional)	The UUID of the volume group containing the physical volume.

Example 6.41. Show physical volume: JSON response

```
{
    "istors":[
        {
            "function":"osd",
            "uuid":"31c7a5a3-9154-462a-9ea3-ab5c5e0d06e0",
            "links":[
                {
                    "href":"http://192.168.204.2:6385/v1/istors/
31c7a5a3-9154-462a-9ea3-ab5c5e0d06e0",
                    "rel":"self"
                },
                {
                    "href":"http://192.168.204.2:6385/istors/
31c7a5a3-9154-462a-9ea3-ab5c5e0d06e0",
                    "rel":"bookmark"
                }
            ],
            "created_at":"2014-10-01T21:41:23.973344+00:00",
            "updated_at":"2014-10-01T21:41:24.129134+00:00",
            "capabilities":{

            },
            "ihost_uuid":"42d72247-e0e3-4a5a-8cb1-40bbee52c8db",
            "state":null,
            "osdid":2
        }
    ]
}
```

This operation does not return a response body.

6.11.3. Create an LMV physical volume

Method	URI	Description
POST	`/v1/ipvs`	Creates an LVM physical volume on a specific host.

This will create an LVM physical volume on the specified host. This functionally is currently only enabled for hosts with a compute personality. In addition, the volume group name is limited to the name: "nova-local". In order to add a physical volume, the host must be locked.

Normal response codes: 200

Error response codes: badMediaType (415)

6.11.3.1. Request

This table shows the body parameters for the create an lmv physical volume request:

Name	Type	Description
`ilvg_uuid`	String *(Optional)*	This parameter specifies the volume group uuid.
`ihost_uuid`	String *(Optional)*	This parameter specifies the compute host uuid.
`idisk_uuid`	String *(Optional)*	This parameter specifies the storage disk uuid.

Example 6.42. Create an LMV physical volume: JSON request

```
{
    "function":"osd",
    "ihost_uuid":"42d72247-e0e3-4a5a-8cb1-40bbee52c8db",
    "idisk_uuid":"4da10410-2959-46df-b571-04e954c0e115"
}
```

This operation does not accept a request body.

6.11.3.2. Response

This table shows the body parameters for the create an lmv physical volume response:

Name	Type	Description
`capabilities`	String *(Optional)*	Additional capabilities info about the volume group.
`pv_state`	String *(Optional)*	This is the state of the physical volume. It has one of the following values: unprovisioned, adding, provisioned, or removing.
`pv_type`	String *(Optional)*	This is the type of physical volume that is allocated. This will have the value of disk or partition.

Name	Type	Description
idisk_uuid (Optional)	String	This is the UUID of the disk that is associated with this physical volume.
idisk_device_node (Optional)	String	This is the device node name associated with the physical partition.
lvm_pv_name (Optional)	String	This is the physical volume name as retrieved from the pvdisplay command on the host.
lvm_vg_name (Optional)	String	This is the name of the volume group that this physical volume belongs as retrieved from the pvdisplay command on the host.
lvm_pv_uuid (Optional)	String	This is the LVM generated UUID for the physical volume as retrieved from the pvdisplay command on the host.
lvm_pv_size (Optional)	String	This is the LVM generated size in bytes of the physical volume as retrieved from the pvdisplay command on the host.
lvm_pe_total (Optional)	String	This is the LVM generated total number of physical extents associated with the physical volume as retrieved from the pvdisplay command on the host.
lvm_pe_alloced (Optional)	String	This is the LVM generated number of allocated physical extents associated with the physical volume as retrieved from the pvdisplay command on the host.
ihost_uuid (Optional)	UUID	The UUID of the host containing the port.
ilvg_uuid (Optional)	UUID	The UUID of the volume group containing the physical volume.

Example 6.43. Create an LMV physical volume: JSON response

```
{
    "function":"osd",
    "uuid":"31c7a5a3-9154-462a-9ea3-ab5c5e0d06e0",
    "links":[
        {
            "href":"http://192.168.204.2:6385/v1/istors/31c7a5a3-9154-462a-9ea3-
ab5c5e0d06e0",
            "rel":"self"
        },
        {
            "href":"http://192.168.204.2:6385/istors/31c7a5a3-9154-462a-9ea3-
ab5c5e0d06e0",
            "rel":"bookmark"
        }
    ],
    "idisks":[
        {
            "href":"http://192.168.204.2:6385/v1/istors/31c7a5a3-9154-462a-9ea3-
ab5c5e0d06e0/idisks",
            "rel":"self"
        },
        {
            "href":"http://192.168.204.2:6385/istors/31c7a5a3-9154-462a-9ea3-
ab5c5e0d06e0/idisks",
            "rel":"bookmark"
        }
```

```
    ],
    "created_at":"2014-10-01T21:41:23+00:00",
    "updated_at":null,
    "idisk_uuid":"4da10410-2959-46df-b571-04e954c0e115",
    "ihost_uuid":"42d72247-e0e3-4a5a-8cb1-40bbee52c8db",
    "state":null,
    "capabilities":{

    },
    "osdid":2
}
```

This operation does not return a response body.

6.11.4. Delete an LVM physical volume

Method	URI	Description
DELETE	`/v1/ipvs/{physicalvolume_id}`	Deletes a specific LVM physical volume.

In order to delete an LVM physical volume, the host must be locked.

Normal response codes: 204

6.11.4.1. Request

This table shows the URI parameters for the delete an lvm physical volume request:

Name	Type	Description
`{physicalvolume_id}`	UUID	The unique identifier of an existing LVM physical volume.

This operation does not accept a request body.

6.11.4.2. Response

This operation does not return a response body.

6.12. Profiles

These APIs allow the create, display and delete of host profiles. This includes interface profiles, cpu profiles, and volume profiles. NOTE that the same record is used in the database for both hosts and host profiles.

Method	URI	Description
GET	`/v1/iprofiles`	Lists all profiles.
GET	`/v1/iprofiles/{profile_id}`	Shows information about a specific profile.
POST	`/v1/iprofiles`	Creates a profile.
DELETE	`/v1/iprofiles/{profile_id}`	Deletes a specific profile.

6.12.1. List profiles

Method	URI	Description
GET	`/v1/iprofiles`	Lists all profiles.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

6.12.1.1. Request

This operation does not accept a request body.

6.12.1.2. Response

This table shows the body parameters for the list profiles response:

Name	Type	Description
`iprofiles`	List *(Optional)*	The list of profile entities.
`recordtype`	String *(Optional)*	Indicates that the record is being used for host profile rather than a host.
`hostname`	String *(Optional)*	The name of the profile.
`uuid`	UUID *(Optional)*	The universally unique identifier for this object.
`links`	List *(Optional)*	For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The following types of link relations are associated with resources: a self link containing a versioned link to the resource, and a bookmark link containing a permanent link to a resource that is appropriate for long term storage.
`created_at`	DateTime *(Optional)*	The time when the object was created.
`updated_at`	DateTime *(Optional)*	The time when the object was last updated.

Example 6.44. List profiles: JSON response

```
{
  "iprofiles": [
    {
      "uuid": "b6bde724-4fda-4941-ae3f-15abd3d4107b",
      "recordtype": "profile",
      "task": null,
      "reserved": "False",
      "mgmt_ip": null,
      "links": [
        {
```

```
        "href": "http://192.168.204.2:6385/v1/iprofile/b6bde724-4fda-4941-
ae3f-15abd3d4107b",
          "rel": "self"
        },
        {
          "href": "http://192.168.204.2:6385/iprofile/b6bde724-4fda-4941-
ae3f-15abd3d4107b",
          "rel": "bookmark"
        }
      ],
      "personality": null,
      "created_at": "2014-09-29T13:36:36.760707+00:00",
      "hostname": "ifprofile-type-1",
      "updated_at": null,
      "id": 23,
      "ihost_uuid": null,
      "profiletype": null,
      "location": {

      },
      "action": "none",
      "profilename": null,
      "operational": "disabled",
      "administrative": "locked",
      "availability": "offline",
      "uptime": 0,
      "mgmt_mac": null
    },
    {
      "uuid": "85b8d979-a1d5-4b06-8666-22646d45dcdf",
      "recordtype": "profile",
      "task": null,
      "reserved": "False",
      "mgmt_ip": null,
      "links": [
        {
          "href": "http://192.168.204.2:6385/v1/iprofile/85b8d979-
a1d5-4b06-8666-22646d45dcdf",
          "rel": "self"
        },
        {
          "href": "http://192.168.204.2:6385/iprofile/85b8d979-
a1d5-4b06-8666-22646d45dcdf",
          "rel": "bookmark"
        }
      ],
      "personality": null,
      "created_at": "2014-09-29T13:42:40.592612+00:00",
      "hostname": "ifprofile-type-2",
      "updated_at": null,
      "id": 24,
      "ihost_uuid": null,
      "profiletype": null,
      "location": {

      },
      "action": "none",
      "profilename": null,
      "operational": "disabled",
      "administrative": "locked",
```

```
      "availability": "offline",
      "uptime": 0,
      "mgmt_mac": null
    }
  ]
}
```

This operation does not return a response body.

6.12.2. Show profile

Method	URI	Description
GET	`/v1/iprofiles/{profile_id}`	Shows information about a specific profile.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

6.12.2.1. Request

This table shows the URI parameters for the show profile request:

Name	Type	Description
`{host_id}`	UUID	The unique identifier of an existing profile.

This operation does not accept a request body.

6.12.2.2. Response

This table shows the body parameters for the show profile response:

Name	Type	Description
`recordtype`	String *(Optional)*	Indicates that the record is being used for host profile rather than a host.
`hostname`	String *(Optional)*	The name of the profile.
`ports`	List *(Optional)*	Links to the ports of the profile.
`interfaces`	List *(Optional)*	Links to the interfaces of the profile.
`idisks`	List *(Optional)*	Links to the disks of the profile.
`istors`	List *(Optional)*	Links to the volumes of the profile.
`inodes`	List *(Optional)*	Links to the NUMA Nodes of the profile.
`icpus`	List *(Optional)*	Links to the logical cores (CPUs) of the profile.
`uuid`	UUID *(Optional)*	The universally unique identifier for this object.
`links`	List *(Optional)*	For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The following types of link relations are associated with resources: a self link containing a versioned link to the resource, and a bookmark link contain-

Name	Type	Description
		ing a permanent link to a resource that is appropriate for long term storage.
created_at	DateTime *(Optional)*	The time when the object was created.
updated_at	DateTime *(Optional)*	The time when the object was last updated.

Example 6.45. Show profile: JSON response

```json
{
   "iports" : [
      {
         "rel" : "self",
         "href" : "http://128.224.151.244:6385/v1/iprofile/85b8d979-
a1d5-4b06-8666-22646d45dcdf/iports"
      },
      {
         "rel" : "bookmark",
         "href" : "http://128.224.151.244:6385/iprofile/85b8d979-
a1d5-4b06-8666-22646d45dcdf/iports"
      }
   ],
   "operational" : "disabled",
   "imemorys" : [
      {
         "rel" : "self",
         "href" : "http://128.224.151.244:6385/v1/ihosts/85b8d979-
a1d5-4b06-8666-22646d45dcdf/imemorys"
      },
      {
         "rel" : "bookmark",
         "href" : "http://128.224.151.244:6385/ihosts/85b8d979-
a1d5-4b06-8666-22646d45dcdf/imemorys"
      }
   ],
   "iinterfaces" : [
      {
         "rel" : "self",
         "href" : "http://128.224.151.244:6385/v1/iprofile/85b8d979-
a1d5-4b06-8666-22646d45dcdf/iinterfaces"
      },
      {
         "rel" : "bookmark",
         "href" : "http://128.224.151.244:6385/iprofile/85b8d979-
a1d5-4b06-8666-22646d45dcdf/iinterfaces"
      }
   ],
   "personality" : null,
   "serialId" : null,
   "hostname" : "ifprofile-type-2",
   "profilename" : null,
   "uuid" : "85b8d979-a1d5-4b06-8666-22646d45dcdf",
   "profiletype" : null,
   "ihost_uuid" : null,
   "created_at" : "2014-09-29T13:42:40.592612+00:00",
   "availability" : "offline",
   "recordtype" : "profile",
```

```
    "istors" : [
      {
        "rel" : "self",
        "href" : "http://128.224.151.244:6385/v1/iprofile/85b8d979-
a1d5-4b06-8666-22646d45dcdf/istors"
      },
      {
        "rel" : "bookmark",
        "href" : "http://128.224.151.244:6385/iprofile/85b8d979-
a1d5-4b06-8666-22646d45dcdf/istors"
      }
    ],
    "idisks" : [
      {
        "rel" : "self",
        "href" : "http://128.224.151.244:6385/v1/iprofile/85b8d979-
a1d5-4b06-8666-22646d45dcdf/idisks"
      },
      {
        "rel" : "bookmark",
        "href" : "http://128.224.151.244:6385/iprofile/85b8d979-
a1d5-4b06-8666-22646d45dcdf/idisks"
      }
    ],
    "uptime" : 0,
    "icpus" : [
      {
        "rel" : "self",
        "href" : "http://128.224.151.244:6385/v1/ihosts/85b8d979-
a1d5-4b06-8666-22646d45dcdf/icpus"
      },
      {
        "rel" : "bookmark",
        "href" : "http://128.224.151.244:6385/ihosts/85b8d979-
a1d5-4b06-8666-22646d45dcdf/icpus"
      }
    ],
    "id" : 24,
    "mgmt_ip" : null,
    "links" : [
      {
        "rel" : "self",
        "href" : "http://128.224.151.244:6385/v1/iprofile/85b8d979-
a1d5-4b06-8666-22646d45dcdf"
      },
      {
        "rel" : "bookmark",
        "href" : "http://128.224.151.244:6385/iprofile/85b8d979-
a1d5-4b06-8666-22646d45dcdf"
      }
    ],
    "location" : {},
    "inodes" : [
      {
        "rel" : "self",
        "href" : "http://128.224.151.244:6385/v1/ihosts/85b8d979-
a1d5-4b06-8666-22646d45dcdf/inodes"
      },
      {
        "rel" : "bookmark",
```

```
        "href" : "http://128.224.151.244:6385/ihosts/85b8d979-
a1d5-4b06-8666-22646d45dcdf/inodes"
      }
  ],
  "task" : null,
  "mgmt_mac" : null,
  "invprovision" : null,
  "administrative" : "locked",
  "updated_at" : null,
  "action" : "none",
  "reserved" : "False"
}
```

This operation does not return a response body.

6.12.3. Create profile

Method	URI	Description
POST	`/v1/iprofiles`	Creates a profile.

Normal response codes: 200

Error response codes: badMediaType (415)

6.12.3.1. Request

This table shows the body parameters for the create profile request:

Name	Type	Description
`profilename`	String *(Optional)*	The name for the new profile.
`profiletype`	String *(Optional)*	The type of profile to be created. Valid values are: `if`, `cpu` or `stor`.
`ihost_uuid`	String *(Optional)*	The UUID of the Host to create the profile based on.

Example 6.46. Create profile: JSON request

```
{
  "profilename": "ifprofile-type-1",
  "profiletype": "if",
  "ihost_uuid": "959f785b-6387-4b98-aa30-bc861061d7a1"
}
```

This operation does not accept a request body.

6.12.3.2. Response

This table shows the body parameters for the create profile response:

Name	Type	Description
`recordtype`	String *(Optional)*	Indicates that the record is being used for host profile rather than a host.
`hostname`	String *(Optional)*	The name of the profile.
`ports`	List *(Optional)*	Links to the ports of the profile.
`interfaces`	List *(Optional)*	Links to the interfaces of the profile.
`idisks`	List *(Optional)*	Links to the disks of the profile.
`istors`	List *(Optional)*	Links to the volumes of the profile.
`inodes`	List	Links to the NUMA Nodes of the profile.

Name	Type	Description
	(Optional)	
`icpus`	List *(Optional)*	Links to the logical cores (CPUs) of the profile.
`uuid`	UUID *(Optional)*	The universally unique identifier for this object.
`links`	List *(Optional)*	For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The following types of link relations are associated with resources: a self link containing a versioned link to the resource, and a bookmark link containing a permanent link to a resource that is appropriate for long term storage.
`created_at`	DateTime *(Optional)*	The time when the object was created.
`updated_at`	DateTime *(Optional)*	The time when the object was last updated.

Example 6.47. Create profile: JSON response

```
{
  "iports": [
    {
      "href": "http://192.168.204.2:6385/v1/iprofile/b6bde724-4fda-4941-
ae3f-15abd3d4107b/iports",
      "rel": "self"
    },
    {
      "href": "http://192.168.204.2:6385/iprofile/b6bde724-4fda-4941-
ae3f-15abd3d4107b/iports",
      "rel": "bookmark"
    }
  ],
  "reserved": "False",
  "links": [
    {
      "href": "http://192.168.204.2:6385/v1/iprofile/b6bde724-4fda-4941-
ae3f-15abd3d4107b",
      "rel": "self"
    },
    {
      "href": "http://192.168.204.2:6385/iprofile/b6bde724-4fda-4941-
ae3f-15abd3d4107b",
      "rel": "bookmark"
    }
  ],
  "idisks": [
    {
      "href": "http://192.168.204.2:6385/v1/iprofile/b6bde724-4fda-4941-
ae3f-15abd3d4107b/idisks",
      "rel": "self"
    },
    {
      "href": "http://192.168.204.2:6385/iprofile/b6bde724-4fda-4941-
ae3f-15abd3d4107b/idisks",
      "rel": "bookmark"
    }
```

```
      ],
      "availability": "offline",
      "updated_at": null,
      "ihost_uuid": null,
      "id": 23,
      "icpus": [
        {
          "href": "http://192.168.204.2:6385/v1/ihosts/b6bde724-4fda-4941-
ae3f-15abd3d4107b/icpus",
          "rel": "self"
        },
        {
          "href": "http://192.168.204.2:6385/ihosts/b6bde724-4fda-4941-
ae3f-15abd3d4107b/icpus",
          "rel": "bookmark"
        }
      ],
      "uptime": 0,
      "uuid": "b6bde724-4fda-4941-ae3f-15abd3d4107b",
      "mgmt_ip": null,
      "hostname": "ifprofile-type-1",
      "istors": [
        {
          "href": "http://192.168.204.2:6385/v1/iprofile/b6bde724-4fda-4941-
ae3f-15abd3d4107b/istors",
          "rel": "self"
        },
        {
          "href": "http://192.168.204.2:6385/iprofile/b6bde724-4fda-4941-
ae3f-15abd3d4107b/istors",
          "rel": "bookmark"
        }
      ],
      "operational": "disabled",
      "location": {

      },
      "invprovision": null,
      "administrative": "locked",
      "personality": null,
      "iinterfaces": [
        {
          "href": "http://192.168.204.2:6385/v1/iprofile/b6bde724-4fda-4941-
ae3f-15abd3d4107b/iinterfaces",
          "rel": "self"
        },
        {
          "href": "http://192.168.204.2:6385/iprofile/b6bde724-4fda-4941-
ae3f-15abd3d4107b/iinterfaces",
          "rel": "bookmark"
        }
      ],
      "profiletype": null,
      "mgmt_mac": null,
      "task": null,
      "recordtype": "profile",
      "created_at": "2014-09-29T13:36:36.760707+00:00",
      "action": "none",
      "profilename": null,
      "serialId": null,
```

```
  "inodes": [
    {
      "href": "http://192.168.204.2:6385/v1/ihosts/b6bde724-4fda-4941-
ae3f-15abd3d4107b/inodes",
      "rel": "self"
    },
    {
      "href": "http://192.168.204.2:6385/ihosts/b6bde724-4fda-4941-
ae3f-15abd3d4107b/inodes",
      "rel": "bookmark"
    }
  ],
  "imemorys": [
    {
      "href": "http://192.168.204.2:6385/v1/ihosts/b6bde724-4fda-4941-
ae3f-15abd3d4107b/imemorys",
      "rel": "self"
    },
    {
      "href": "http://192.168.204.2:6385/ihosts/b6bde724-4fda-4941-
ae3f-15abd3d4107b/imemorys",
      "rel": "bookmark"
    }
  ]
}
```

This operation does not return a response body.

6.12.4. Delete profile

Method	URI	Description
DELETE	`/v1/iprofiles/{profile_id}`	Deletes a specific profile.

Normal response codes: 204

6.12.4.1. Request

This table shows the URI parameters for the delete profile request:

Name	Type	Description
`{host_id}`	UUID	The unique identifier of an existing profile.

This operation does not accept a request body.

6.12.4.2. Response

This operation does not return a response body.

6.13. DNS

The DNS is the Domain Name Server entity for the system.

Method	URI	Description
GET	`/v1/idns`	Shows attributes of the DNS object.
PATCH	`/v1/idns/{dns_id}`	Modifies attributes of the DNS object.

6.13.1. Show dns

Method	URI	Description
GET	`/v1/idns`	Shows attributes of the DNS object.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

6.13.1.1. Request

This operation does not accept a request body.

6.13.1.2. Response

This table shows the body parameters for the show dns response:

Name	Type	Description
`recordtype`	String *(Optional)*	The DNS recordtype.
`istate`	String *(Optional)*	The state of the DNS configuration.
`nameservers`	String *(Optional)*	The comma-separated list of DNS nameservers.
`isystem_uuid`	UUID *(Optional)*	The System UUID which the DNS belongs to.
`uuid`	UUID *(Optional)*	The universally unique identifier for this object.
`links`	List *(Optional)*	For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The following types of link relations are associated with resources: a self link containing a versioned link to the resource, and a bookmark link containing a permanent link to a resource that is appropriate for long term storage.
`created_at`	DateTime *(Optional)*	The time when the object was created.
`updated_at`	DateTime *(Optional)*	The time when the object was last updated.

Example 6.48. Show dns: JSON response

```
{
  "idnss": [
    {
      "recordtype": "reconfig",
      "links": [
```

```
            {
                "href": "http://192.168.204.2:6385/v1/idnss/fab4ff99-
ed44-41d0-9e04-2efb3138cf03",
                "rel": "self"
            },
            {
                "href": "http://192.168.204.2:6385/idnss/fab4ff99-
ed44-41d0-9e04-2efb3138cf03",
                "rel": "bookmark"
            }
        ],
        "nameservers": "8.8.8.4,8.8.4.5",
        "created_at": "2014-09-30T14:42:16.676726+00:00",
        "updated_at": "2014-10-01T15:10:42.328364+00:00",
        "istate": "applied",
        "isystem_uuid": "ce178041-2b2c-405d-bf87-f19334a35582",
        "uuid": "fab4ff99-ed44-41d0-9e04-2efb3138cf03"
    }
  ]
}
```

This operation does not return a response body.

6.13.2. Modify dns

Method	URI	Description
PATCH	`/v1/idns/{dns_id}`	Modifies attributes of the DNS object.

The attributes of the DNS object that are configurable are:

- nameservers

Normal response codes: 200

Error response codes: badMediaType (415)

6.13.2.1. Request

This table shows the URI parameters for the modify dns request:

Name	Type	Description
`{dns_id}`	UUID	The unique identifier of the DNS for this system.

This table shows the body parameters for the modify dns request:

Name	Type	Description
`nameservers`	String *(Optional)*	This parameter specifies the list of Domain Name Servers (DNS). Comma separated list.

Example 6.49. Modify dns: JSON request

```
[
  {
    "path": "/nameservers",
    "value": "8.8.8.99,8.8.4.99",
    "op": "replace"
  },
  {
    "path": "/action",
    "value": "apply",
    "op": "replace"
  }
]
```

This operation does not accept a request body.

6.13.2.2. Response

This table shows the body parameters for the modify dns response:

Name	Type	Description
`recordtype`	String *(Optional)*	The DNS recordtype.
`istate`	String	The state of the DNS configuration.

Name	Type	Description
	(Optional)	
nameservers	String (Optional)	The comma-separated list of DNS nameservers.
isystem_uuid	UUID (Optional)	The System UUID which the DNS belongs to.
uuid	UUID (Optional)	The universally unique identifier for this object.
links	List (Optional)	For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The following types of link relations are associated with resources: a self link containing a versioned link to the resource, and a bookmark link containing a permanent link to a resource that is appropriate for long term storage.
created_at	DateTime (Optional)	The time when the object was created.
updated_at	DateTime (Optional)	The time when the object was last updated.

Example 6.50. Modify dns: JSON response

```
{
  "recordtype": "reconfig",
  "links": [
    {
      "href": "http://192.168.204.2:6385/v1/idnss/fab4ff99-
ed44-41d0-9e04-2efb3138cf03",
      "rel": "self"
    },
    {
      "href": "http://192.168.204.2:6385/idnss/fab4ff99-
ed44-41d0-9e04-2efb3138cf03",
      "rel": "bookmark"
    }
  ],
  "nameservers": "8.8.8.99,8.8.4.99",
  "created_at": "2014-09-30T14:42:16.676726+00:00",
  "updated_at": "2014-10-01T15:13:42.356658+00:00",
  "isystem_uuid": "ce178041-2b2c-405d-bf87-f19334a35582",
  "istate": "applying",
  "action": null,
  "forisystemid": 1,
  "uuid": "fab4ff99-ed44-41d0-9e04-2efb3138cf03"
}
```

This operation does not return a response body.

6.14. NTP

The NTP is the Network Time Protocol entity for the system.

Method	URI	Description
GET	/v1/intp	Shows attributes of the NTP object.

Method	URI	Description
PATCH	`/v1/intp/{ntp_id}`	Modifies attributes of the NTP object.

6.14.1. Show ntp

Method	URI	Description
GET	`/v1/intp`	Shows attributes of the NTP object.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

6.14.1.1. Request

This operation does not accept a request body.

6.14.1.2. Response

This table shows the body parameters for the show ntp response:

Name	Type	Description
recordtype	String *(Optional)*	The NTP recordtype.
istate	String *(Optional)*	The state of the NTP configuration.
ntpservers	String *(Optional)*	The comma-separated list of NTP ntpservers.
isystem_uuid	UUID *(Optional)*	The System UUID which the NTP belongs to.
uuid	UUID *(Optional)*	The universally unique identifier for this object.
links	List *(Optional)*	For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The following types of link relations are associated with resources: a self link containing a versioned link to the resource, and a bookmark link containing a permanent link to a resource that is appropriate for long term storage.
created_at	DateTime *(Optional)*	The time when the object was created.
updated_at	DateTime *(Optional)*	The time when the object was last updated.

Example 6.51. Show ntp: JSON response

```
{
    "intps":[
        {
            "recordtype":"reconfig",
            "links":[
```

```
            {
                "href":"http://192.168.204.2:6385/v1/intps/
81321749-5092-4faf-94ba-6a6853440725",
                "rel":"self"
            },
            {
                "href":"http://192.168.204.2:6385/intps/
81321749-5092-4faf-94ba-6a6853440725",
                "rel":"bookmark"
            }
        ],
        "created_at":"2014-09-30T14:42:16.693209+00:00",
        "updated_at":"2014-10-01T17:33:43.169595+00:00",
        "ntpservers":"0.pool.ntp.org,2.pool.ntp.org,1.pool.ntp.org",
        "istate":"applied",
        "isystem_uuid":"ce178041-2b2c-405d-bf87-f19334a35582",
        "uuid":"81321749-5092-4faf-94ba-6a6853440725"
    }
  ]
}
```

This operation does not return a response body.

6.14.2. Modify ntp

Method	URI	Description
PATCH	`/v1/intp/{ntp_id}`	Modifies attributes of the NTP object.

The attributes of the NTP object that are configurable are:

- ntpservers

Normal response codes: 200

Error response codes: badMediaType (415)

6.14.2.1. Request

This table shows the URI parameters for the modify ntp request:

Name	Type	Description
`{ntp_id}`	UUID	The unique identifier of the NTP for this system.

This table shows the body parameters for the modify ntp request:

Name	Type	Description
`ntpservers`	String *(Optional)*	This parameter specifies the list of Network Time Protocol (NTP) Servers. Comma separated list.

Example 6.52. Modify ntp: JSON request

```
[
    {
        "path":"/ntpservers",
        "value":"0.pool.ntp.org,1.pool.ntp.org,2.pool.ntp.org",
        "op":"replace"
    },
    {
        "path":"/action",
        "value":"apply",
        "op":"replace"
    }
]
```

This operation does not accept a request body.

6.14.2.2. Response

This table shows the body parameters for the modify ntp response:

Name	Type	Description
`recordtype`	String *(Optional)*	The NTP recordtype.
`istate`	String	The state of the NTP configuration.

Name	Type	Description
	(Optional)	
`ntpservers`	String *(Optional)*	The comma-separated list of NTP ntpservers.
`isystem_uuid`	UUID *(Optional)*	The System UUID which the NTP belongs to.
`uuid`	UUID *(Optional)*	The universally unique identifier for this object.
`links`	List *(Optional)*	For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The following types of link relations are associated with resources: a self link containing a versioned link to the resource, and a bookmark link containing a permanent link to a resource that is appropriate for long term storage.
`created_at`	DateTime *(Optional)*	The time when the object was created.
`updated_at`	DateTime *(Optional)*	The time when the object was last updated.

Example 6.53. Modify ntp: JSON response

```
{
    "recordtype":"reconfig",
    "links":[
        {
            "href":"http://192.168.204.2:6385/v1/intps/
81321749-5092-4faf-94ba-6a6853440725",
            "rel":"self"
        },
        {
            "href":"http://192.168.204.2:6385/intps/
81321749-5092-4faf-94ba-6a6853440725",
            "rel":"bookmark"
        }
    ],
    "created_at":"2014-09-30T14:42:16.693209+00:00",
    "updated_at":"2014-10-01T17:35:43.162472+00:00",
    "ntpservers":"0.pool.ntp.org,1.pool.ntp.org,2.pool.ntp.org",
    "isystem_uuid":"ce178041-2b2c-405d-bf87-f19334a35582",
    "istate":"applying",
    "action":null,
    "forisystemid":1,
    "uuid":"81321749-5092-4faf-94ba-6a6853440725"
}
```

This operation does not return a response body.

6.15. External OAM

The extoam is the External OAM entity for the system.

Method	URI	Description
GET	`/v1/iextoam`	Shows attributes of the External OAM object.

Method	URI	Description
PATCH	`/v1/iextoam/{extoam_id}`	Modifies attributes of the External OAM object.

6.15.1. Show External OAM

Method	URI	Description
GET	`/v1/iextoam`	Shows attributes of the External OAM object.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

6.15.1.1. Request

This operation does not accept a request body.

6.15.1.2. Response

This table shows the body parameters for the show external oam response:

Name	Type	Description
`recordtype`	String *(Optional)*	The External OAM IP recordtype.
`istate`	String *(Optional)*	The state of the External OAM IP configuration.
`oam_subnet`	String *(Optional)*	The External OAM IP Subnet.
`oam_gateway_ip`	String *(Optional)*	The External OAM Gateway IP Address.
`oam_floating_ip`	String *(Optional)*	The External OAM Floating IP Address.
`oam_c0_ip`	String *(Optional)*	The External OAM Controller-0 IP Address.
`oam_c1_ip`	String *(Optional)*	The External OAM Controller-1 IP Address.
`oam_ifcs`	String *(Optional)*	The External OAM IP Interfaces.
`oam_mtu`	String *(Optional)*	The External OAM MTU size.
`isystem_uuid`	UUID *(Optional)*	The System UUID which the External OAM IP belongs to.
`uuid`	UUID *(Optional)*	The universally unique identifier for this object.
`links`	List *(Optional)*	For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The follow-

Name	Type	Description
		ing types of link relations are associated with resources: a self link containing a versioned link to the resource, and a bookmark link containing a permanent link to a resource that is appropriate for long term storage.
created_at	DateTime *(Optional)*	The time when the object was created.
updated_at	DateTime *(Optional)*	The time when the object was last updated.

Example 6.54. Show External OAM: JSON response

```
{
    "iextoams":[
        {
            "oam_c1_ip":"10.10.10.6",
            "recordtype":"reconfig",
            "links":[
                {
                    "href":"http://192.168.204.2:6385/v1/iextoams/
2056b372-10a5-47d3-b1da-8957c370b630",
                    "rel":"self"
                },
                {
                    "href":"http://192.168.204.2:6385/iextoams/2056b372-10a5-47d3-
b1da-8957c370b630",
                    "rel":"bookmark"
                }
            ],
            "created_at":"2014-09-30T14:42:16.656226+00:00",
            "updated_at":"2014-10-01T17:35:43.131331+00:00",
            "oam_subnet":"10.10.10.0/24",
            "istate":"applied",
            "oam_ifcs":"eth0",
            "oam_gateway_ip":"10.10.10.1",
            "oam_floating_ip":"10.10.10.2",
            "oam_mtu":1500,
            "isystem_uuid":"ce178041-2b2c-405d-bf87-f19334a35582",
            "oam_lag_ifc":"False",
            "oam_c0_ip":"10.10.10.3",
            "uuid":"2056b372-10a5-47d3-b1da-8957c370b630"
        }
    ]
}
```

This operation does not return a response body.

6.15.2. Modify External OAM

Method	URI	Description
PATCH	`/v1/iextoam/{extoam_id}`	Modifies attributes of the External OAM object.

The attributes of the External OAM objects that are configurable are:

- oam_subnet

- oam_gateway_ip

- oam_floating_ip

- oam_c0_ip

- oam_c1_ip

Normal response codes: 200

Error response codes: badMediaType (415)

6.15.2.1. Request

This table shows the URI parameters for the modify external oam request:

Name	Type	Description
`{extoam_id}`	UUID	The unique identifier of the External OAM for this system.

This table shows the body parameters for the modify external oam request:

Name	Type	Description
`oam_subnet`	String (Optional)	This parameter specifies External OAM IP Subnet.
`oam_gateway_ip`	String (Optional)	This parameter specifies External OAM Gateway IP Address.
`oam_floating_ip`	String (Optional)	This parameter specifies External OAM Floating IP.
`oam_c0_ip`	String (Optional)	This parameter specifies External OAM Controller-0 IP Address.
`oam_c1_ip`	String (Optional)	This parameter specifies External OAM Controller-1 IP Address.

Example 6.55. Modify External OAM: JSON request

```
[
  {
    "path":"/oam_c1_ip",
    "value":"10.10.10.4",
    "op":"replace"
```

```
    },
    {
        "path":"/action",
        "value":"apply",
        "op":"replace"
    }
]
```

This operation does not accept a request body.

6.15.2.2. Response

This table shows the body parameters for the modify external oam response:

Name	Type	Description
recordtype	String *(Optional)*	The External OAM IP recordtype.
istate	String *(Optional)*	The state of the External OAM IP configuration.
oam_subnet	String *(Optional)*	The External OAM IP Subnet.
oam_gateway_ip	String *(Optional)*	The External OAM Gateway IP Address.
oam_floating_ip	String *(Optional)*	The External OAM Floating IP Address.
oam_c0_ip	String *(Optional)*	The External OAM Controller-0 IP Address.
oam_c1_ip	String *(Optional)*	The External OAM Controller-1 IP Address.
oam_ifcs	String *(Optional)*	The External OAM IP Interfaces.
oam_mtu	String *(Optional)*	The External OAM MTU size.
isystem_uuid	UUID *(Optional)*	The System UUID which the External OAM IP belongs to.
uuid	UUID *(Optional)*	The universally unique identifier for this object.
links	List *(Optional)*	For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The following types of link relations are associated with resources: a self link containing a versioned link to the resource, and a bookmark link containing a permanent link to a resource that is appropriate for long term storage.
created_at	DateTime *(Optional)*	The time when the object was created.
updated_at	DateTime *(Optional)*	The time when the object was last updated.

Example 6.56. Modify External OAM: JSON response

```
{
    "iextoams":[
        {
            "oam_c1_ip":"10.10.10.6",
            "recordtype":"reconfig",
            "links":[
                {
                    "href":"http://192.168.204.2:6385/v1/iextoams/
2056b372-10a5-47d3-b1da-8957c370b630",
                    "rel":"self"
                },
                {
                    "href":"http://192.168.204.2:6385/iextoams/2056b372-10a5-47d3-
b1da-8957c370b630",
                    "rel":"bookmark"
                }
            ],
            "created_at":"2014-09-30T14:42:16.656226+00:00",
            "updated_at":"2014-10-01T17:35:43.131331+00:00",
            "oam_subnet":"10.10.10.0/24",
            "istate":"applied",
            "oam_ifcs":"eth0",
            "oam_gateway_ip":"10.10.10.1",
            "oam_floating_ip":"10.10.10.2",
            "oam_mtu":1500,
            "isystem_uuid":"ce178041-2b2c-405d-bf87-f19334a35582",
            "oam_lag_ifc":"False",
            "oam_c0_ip":"10.10.10.3",
            "uuid":"2056b372-10a5-47d3-b1da-8957c370b630"
        }
    ]
}
```

This operation does not return a response body.

6.16. Infrastructure Subnet

The infra is the Infrastructure subnet entity for the system.

Method	URI	Description
GET	/v1/iinfra	Shows attributes of the infrastructure network IP subnet object.
PATCH	/v1/iinfra/{infra_id}	Modifies attributes of the infrastructure network IP subnet object.

6.16.1. Show infrastructure network IP subnet configuration

Method	URI	Description
GET	`/v1/iinfra`	Shows attributes of the infrastructure network IP subnet object.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

6.16.1.1. Request

This operation does not accept a request body.

6.16.1.2. Response

This table shows the body parameters for the show infrastructure network ip subnet configuration response:

Name	Type	Description
`recordtype`	String (Optional)	The infrastructure network IP subnet recordtype.
`istate`	String (Optional)	The state of the infrastructure network IP subnet configuration.
`infra_subnet`	String (Optional)	The infrastructure network IP subnet.
`isystem_uuid`	UUID (Optional)	The System UUID which the infrastructure network IP subnet belongs to.
`uuid`	UUID (Optional)	The universally unique identifier for this object.
`links`	List (Optional)	For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The following types of link relations are associated with resources: a self link containing a versioned link to the resource, and a bookmark link containing a permanent link to a resource that is appropriate for long term storage.
`created_at`	DateTime (Optional)	The time when the object was created.
`updated_at`	DateTime (Optional)	The time when the object was last updated.

Example 6.57. Show infrastructure network IP subnet configuration: JSON response

```
{
    "iinfras":[
```

```
        {
            "infra_subnet": "192.168.205.0/24",
            "recordtype": "reconfig",
            "links":[
                {
                    "href": "http://10.10.10.2:6385/v1/iinfras/15ba72aa-442f-4b57-
a8a1-e3f3cdd5b04c",
                    "rel": "self"
                },
                {
                    "href": "http://10.10.10.2:6385/iinfras/15ba72aa-442f-4b57-
a8a1-e3f3cdd5b04c",
                    "rel": "bookmark"
                }
            ],
            "created_at": "2014-12-01T19:35:36.965657+00:00",
            "updated_at": "2014-12-02T20:38:51.646902+00:00",
            "istate": "applied",
            "isystem_uuid": "b7fdc4dd-333d-471b-b21d-ba664ea87714",
            "uuid": "15ba72aa-442f-4b57-a8a1-e3f3cdd5b04c"
        }
    ]
}
```

This operation does not return a response body.

6.16.2. Modify infrastructure network IP subnet

Method	URI	Description
PATCH	`/v1/iinfra/{infra_id}`	Modifies attributes of the infrastructure network IP subnet object.

The attributes of the infrastructure network IP subnet object that are configurable are:

- infra_subnet

Normal response codes: 200

Error response codes: badMediaType (415)

6.16.2.1. Request

This table shows the URI parameters for the modify infrastructure network ip subnet request:

Name	Type	Description
`{infra_id}`	UUID	The unique identifier of the infrastructure network subnet configuration for this system.

This table shows the body parameters for the modify infrastructure network ip subnet request:

Name	Type	Description
`infra_subnet`	String *(Optional)*	This parameter specifies the infrastructure network IP subnet.

Example 6.58. Modify infrastructure network IP subnet: JSON request

```
[
    {
        "path":"/infra_subnet",
        "value":"192.168.205.0/24",
        "op":"replace"
    },
    {
        "path":"/action",
        "value":"apply",
        "op":"replace"
    }
]
```

This operation does not accept a request body.

6.16.2.2. Response

This table shows the body parameters for the modify infrastructure network ip subnet response:

Name	Type	Description
`recordtype`	String *(Optional)*	The infrastructure network IP subnet recordtype.

Name	Type	Description
istate *(Optional)*	String	The state of the infrastructure network IP subnet configuration.
infra_subnet *(Optional)*	String	The infrastructure network IP subnet.
isystem_uuid *(Optional)*	UUID	The System UUID which the infrastructure network IP subnet belongs to.
uuid *(Optional)*	UUID	The universally unique identifier for this object.
links *(Optional)*	List	For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The following types of link relations are associated with resources: a self link containing a versioned link to the resource, and a bookmark link containing a permanent link to a resource that is appropriate for long term storage.
created_at *(Optional)*	DateTime	The time when the object was created.
updated_at *(Optional)*	DateTime	The time when the object was last updated.

Example 6.59. Modify infrastructure network IP subnet: JSON response

```
{
    "iinfras":[
        {
            "infra_subnet":"192.168.205.0/24",
            "recordtype":"reconfig",
            "links":[
                {
                    "href":"http://10.10.10.2:6385/v1/iinfras/15ba72aa-442f-4b57-
a8a1-e3f3cdd5b04c",
                    "rel":"self"
                },
                {
                    "href":"http://10.10.10.2:6385/iinfras/15ba72aa-442f-4b57-a8a1-
e3f3cdd5b04c",
                    "rel":"bookmark"
                }
            ],
            "created_at": "2014-12-01T19:35:36.965657+00:00",
            "updated_at": "2014-12-02T20:38:51.646902+00:00",
            "istate": "applied",
            "isystem_uuid": "b7fdc4dd-333d-471b-b21d-ba664ea87714",
            "uuid": "15ba72aa-442f-4b57-a8a1-e3f3cdd5b04c"
        }
    ]
}
```

This operation does not return a response body.

6.17. PM

The PM is the Performance Metrics entity for the system.

Method	URI	Description
GET	`/v1/ipm`	Shows attributes of the PM object.
PATCH	`/v1/ipm/{pm_id}`	Modifies attributes of the PM object.

6.17.1. Show pm

Method	URI	Description
GET	`/v1/ipm`	Shows attributes of the PM object.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

6.17.1.1. Request

This operation does not accept a request body.

6.17.1.2. Response

This table shows the body parameters for the show pm response:

Name	Type	Description
recordtype	String *(Optional)*	The DNS recordtype.
istate	String *(Optional)*	The state of the DNS configuration.
retention_secs	String *(Optional)*	The PM Retention Period in seconds.
isystem_uuid	UUID *(Optional)*	The System UUID which the DNS belongs to.
uuid	UUID *(Optional)*	The universally unique identifier for this object.
links	List *(Optional)*	For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The following types of link relations are associated with resources: a self link containing a versioned link to the resource, and a bookmark link containing a permanent link to a resource that is appropriate for long term storage.
created_at	DateTime *(Optional)*	The time when the object was created.
updated_at	DateTime *(Optional)*	The time when the object was last updated.

Example 6.60. Show pm: JSON response

```
{
    "ipms":[
        {
            "recordtype":"reconfig",
            "links":[
```

```
                {
                    "href":"http://192.168.204.2:6385/v1/ipms/06b02ec7-4b74-4110-
        bd8c-01287ca2bbfd",
                    "rel":"self"
                },
                {
                    "href":"http://192.168.204.2:6385/ipms/06b02ec7-4b74-4110-
        bd8c-01287ca2bbfd",
                    "rel":"bookmark"
                }
            ],
            "created_at":"2014-09-30T14:42:16.715501+00:00",
            "updated_at":"2014-10-01T18:49:05.795331+00:00",
            "istate":"applied",
            "retention_secs":"86400",
            "isystem_uuid":"ce178041-2b2c-405d-bf87-f19334a35582",
            "uuid":"06b02ec7-4b74-4110-bd8c-01287ca2bbfd"
        }
    ]
}
```

This operation does not return a response body.

6.17.2. Modify pm

Method	URI	Description
PATCH	`/v1/ipm/{pm_id}`	Modifies attributes of the PM object.

The attributes of the PM object that are configurable are:

- retention_secs

Normal response codes: 200

Error response codes: badMediaType (415)

6.17.2.1. Request

This table shows the URI parameters for the modify pm request:

Name	Type	Description
`{pm_id}`	UUID	The unique identifier of the PM for this system.

This table shows the body parameters for the modify pm request:

Name	Type	Description
`retention_secs`	String *(Optional)*	This parameter specifies the Performance Metric (PM) records Retention Period in time unit of seconds.

Example 6.61. Modify pm: JSON request

```
[
   {
      "path":"/retention_secs",
      "value":"172800",
      "op":"replace"
   },
   {
      "path":"/action",
      "value":"apply",
      "op":"replace"
   }
]
```

This operation does not accept a request body.

6.17.2.2. Response

This table shows the body parameters for the modify pm response:

Name	Type	Description
`recordtype`	String *(Optional)*	The DNS recordtype.
`istate`	String	The state of the DNS configuration.

Name	Type	Description
	(Optional)	
`retention_secs`	String *(Optional)*	The PM Retention Period in seconds.
`isystem_uuid`	UUID *(Optional)*	The System UUID which the DNS belongs to.
`uuid`	UUID *(Optional)*	The universally unique identifier for this object.
`links`	List *(Optional)*	For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The following types of link relations are associated with resources: a self link containing a versioned link to the resource, and a bookmark link containing a permanent link to a resource that is appropriate for long term storage.
`created_at`	DateTime *(Optional)*	The time when the object was created.
`updated_at`	DateTime *(Optional)*	The time when the object was last updated.

Example 6.62. Modify pm: JSON response

```
{
    "recordtype":"reconfig",
    "links":[
        {
            "href":"http://192.168.204.2:6385/v1/ipms/06b02ec7-4b74-4110-
bd8c-01287ca2bbfd",
            "rel":"self"
        },
        {
            "href":"http://192.168.204.2:6385/ipms/06b02ec7-4b74-4110-
bd8c-01287ca2bbfd",
            "rel":"bookmark"
        }
    ],
    "created_at":"2014-09-30T14:42:16.715501+00:00",
    "updated_at":"2014-10-01T18:49:05.795331+00:00",
    "isystem_uuid":"ce178041-2b2c-405d-bf87-f19334a35582",
    "istate":"applying",
    "retention_secs":"172800",
    "action":null,
    "forisystemid":1,
    "uuid":"06b02ec7-4b74-4110-bd8c-01287ca2bbfd"
}
```

This operation does not return a response body.

6.18. Alarms

These APIs allow the display of the Active Alarms in the system.

Method	URI	Description
GET	`/v1/ialarms`	Lists all active alarms based on specified query.

Method	URI	Description
GET	`/v1/ialarms/{alarm_uuid}`	Shows information about a specific alarm.
DELETE	`/v1/ialarms/{alarm_uuid}`	Deletes a specific alarm.

6.18.1. List alarms

Method	URI	Description
GET	`/v1/ialarms`	Lists all active alarms based on specified query.

The supported query options are alarm_id, entity_type_id, entity_instance_id, severity and alarm_type.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

6.18.1.1. Request

This operation does not accept a request body.

6.18.1.2. Response

This table shows the body parameters for the list alarms response:

Name	Type	Description
`ialarms`	List *(Optional)*	The list of active alarms based on the specified query.
`alarm_id`	String *(Optional)*	The alarm ID; each type of alarm has a unique ID. Note the alarm_id and the entity_instance_id uniquely identify an alarm instance.
`entity_instance_id`	String *(Optional)*	The instance of the object raising alarm. A . separated list of sub-entity-type=instance-value pairs, representing the containment structure of the overall entity instance. Note the alarm_id and the entity_instance_id uniquely identify an alarm instance.
`display_entity_instance_id`	String *(Optional)*	A local and more human-readable version of the entity_instance_id. The system prefix is removed and UUID values are replaced with names where possible (e.g. ports, interfaces, providernets, etc.).
`reason_text`	String *(Optional)*	The text description of the alarm.
`severity`	String *(Optional)*	The severity of the alarm; `critical`, `major`, `minor`, or `warning`.
`timestamp`	DateTime *(Optional)*	The time in UTC at which the alarm has last been updated.
`uuid`	String *(Optional)*	The unique identifier of the alarm.

Example 6.63. List alarms: JSON response

```
{
  "ialarms": [
    {
"severity": "minor",
```

```
"display_entity_instance_id": "host=controller-1",
"timestamp": "2014-09-24T00:15:23.208394",
"uuid": "d75b1e24-f9e3-49b5-b82e-519e4a03cce7",
"alarm_id": "400.003",
"entity_instance_id": "system=big_lab.host=controller-1",
"reason_text": "evaluation license key will exprire on 31-dec-2014"
    },
    {
"severity": "minor",
"display_entity_instance_id": "host=controller-0",
"timestamp": "2014-09-24T00:38:17.400169",
"uuid": "2a88acd3-e9eb-432e-bc0a-4276e3537a40",
"alarm_id": "400.003",
"entity_instance_id": "system=big_lab.host=controller-0",
"reason_text": "evaluation license key will exprire on 31-dec-2014"
    },
    {
"severity": "warning",
"display_entity_instance_id": "host=controller-1",
"timestamp": "2014-09-23T20:42:17.699586",
"uuid": "f19ed030-53c9-4bb8-b308-bf1f2087f01e",
"alarm_id": "200.001",
"entity_instance_id": "system=big_lab.host=controller-1",
"reason_text": "controller-1 was taken out-of-service by manual 'Lock Host'
action"
    }
  ]
}
```

This operation does not return a response body.

6.18.2. Show alarm

Method	URI	Description
GET	`/v1/ialarms/{alarm_uuid}`	Shows information about a specific alarm.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

6.18.2.1. Request

This table shows the URI parameters for the show alarm request:

Name	Type	Description
`{alarm_uuid}`	UUID	The unique identifier of an existing active alarm.

This operation does not accept a request body.

6.18.2.2. Response

This table shows the body parameters for the show alarm response:

Name	Type	Description
`alarm_id`	String *(Optional)*	The alarm ID; each type of alarm has a unique ID. Note the alarm_id and the entity_instance_id uniquely identify an alarm instance.
`entity_instance_id`	String *(Optional)*	The instance of the object raising alarm. A . separated list of sub-entity-type=instance-value pairs, representing the containment structure of the overall entity instance. Note the alarm_id and the entity_instance_id uniquely identify an alarm instance.
`display_entity_instance_id`	String *(Optional)*	A local and more human-readable version of the entity_instance_id. The system prefix is removed and UUID values are replaced with names where possible (e.g. ports, interfaces, providernets, etc.).
`reason_text`	String *(Optional)*	The text description of the alarm.
`severity`	String *(Optional)*	The severity of the alarm; `critical`, `major`, `minor`, or `warning`.
`timestamp`	DateTime *(Optional)*	The time in UTC at which the alarm has last been updated.
`uuid`	String *(Optional)*	The unique identifier of the alarm.
`alarm_state`	String *(Optional)*	The state of the alarm; `set` or `clear`
`service_affecting`	String *(Optional)*	Indicates whether the alarm affects the service.
`proposed_repair_action`	String	The proposed action to clear the alarm.

Name	Type	Description
	(Optional)	
alarm_type	String *(Optional)*	The type of the alarm.
entity_type_id	String *(Optional)*	The type of the object raising the alarm. A . separated list of sub-entity-type, representing the containment structure of the overall entity type.
probable_cause	String *(Optional)*	The probable cause of the alarm.
suppression	String *(Optional)*	Indicates whether suppression of the specific alarm is allowed.
uuid	UUID *(Optional)*	The universally unique identifier for this object.
links	List *(Optional)*	For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The following types of link relations are associated with resources: a self link containing a versioned link to the resource, and a bookmark link containing a permanent link to a resource that is appropriate for long term storage.
created_at	DateTime *(Optional)*	The time when the object was created.
updated_at	DateTime *(Optional)*	The time when the object was last updated.

Example 6.64. Show alarm: JSON response

```
{
 "alarm_state": "set",
 "service_affecting": "True",
 "proposed_repair_action": "contact next level of support",
 "alarm_type": "processing-error",
 "severity": "minor",
 "created_at": "2014-09-23T00:38:16.797155+00:00",
 "entity_type_id": "host",
 "probable_cause": "key-expired",
 "updated_at": "2014-09-24T00:38:17.403135+00:00",
 "alarm_id": "400.003",
 "entity_instance_id": "system=big_lab.host=controller-0",
 "suppression": "False",
 "timestamp": "2014-09-24T00:38:17.400169+00:00",
 "uuid": "2a88acd3-e9eb-432e-bc0a-4276e3537a40",
 "display_entity_instance_id": "host=controller-0",
 "reason_text": "evaluation license key will exprire on 31-dec-2014"
}
```

This operation does not return a response body.

6.18.3. Delete alarm

Method	URI	Description
DELETE	`/v1/ialarms/{alarm_uuid}`	Deletes a specific alarm.

NOTE Typically this command should NOT be used. I.e typically alarms will be and should be cleared by the system when the alarm condition clears. This command is only provided in the event that the alarm has cleared but for some reason the system has not removed the alarm.

Normal response codes: 204

6.18.3.1. Request

This table shows the URI parameters for the delete alarm request:

Name	Type	Description
`{alarm_uuid}`	UUID	The unique identifier of an existing active alarm.

This operation does not accept a request body.

6.18.3.2. Response

This operation does not return a response body.

6.19. Alarm History

These APIs allow the display of the Alarm History in the system.

Method	URI	Description
GET	`/v1/ialarm_history{?q(Optional), limit(Optional)}`	Lists all alarm history based on specified query. The alarms are returned in reverse chronological order.
GET	`/v1/ialarm_history/{alarm_uuid}`	Shows information about a specific alarm history.

6.19.1. List alarm history

Method	URI	Description
GET	`/v1/ialarm_history{?q(Optional), limit(Optional)}`	Lists all alarm history based on specified query. The alarms are returned in reverse chronological order.

The supported query options are alarm_id, entity_type_id, entity_instance_id, severity, alarm_type, start and end.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

6.19.1.1. Request

This table shows the query parameters for the list alarm history request:

Name	Type	Description
`q(Optional)`	List *(Optional)*	This parameter specifies filter rules for the alarms to be returned.
`limit(Optional)`	Int *(Optional)*	This parameter specifies the maximum number of alarms to be returned.

Example 6.65. List alarm history: JSON request

```
http://192.168.204.2:6385/v1/ialarm_history?q.field=start&q.field=end&
q.op=eq&q.op=eq&q.type=&q.type=&q.value=2014-11-28T16%3A56%3A44&q.value=
2014-11-28T16%3A56%3A45&limit=2
```

This operation does not accept a request body.

6.19.1.2. Response

This table shows the body parameters for the list alarm history response:

Name	Type	Description
`ialarm_history`	List *(Optional)*	The list of alarm history based on the specified query. For a Hierarchical CLEAR ALL alarm history, many of the response attributes are not applicable. The attributes that are applicable for a CLEAR ALL are: alarm_state, entity_instance_id, timestamp and reason_text.
`alarm_id`	String *(Optional)*	The alarm ID; each type of alarm has a unique ID. Note the alarm_id and the entity_instance_id uniquely identify an alarm instance.
`alarm_state`	String *(Optional)*	The state of the alarm; `set` or `clear`
`entity_instance_id`	String *(Optional)*	The instance of the object raising alarm. A . separated list of sub-entity-type=instance-value pairs, representing the containment structure of the overall entity instance. Note the alarm_id and the entity_instance_id uniquely identify an alarm instance.

Name	Type	Description
display_entity_instance_id	String *(Optional)*	A local and more human-readable version of the entity_instance_id. The system prefix is removed and UUID values are replaced with names where possible (e.g. ports, interfaces, providernets, etc.).
reason_text	String *(Optional)*	The text description of the alarm.
severity	String *(Optional)*	The severity of the alarm; `critical`, `major`, `minor`, or `warning`.
timestamp	DateTime *(Optional)*	The time in UTC at which the alarm has last been updated.
uuid	String *(Optional)*	The unique identifier of the alarm.
next	String *(Optional)*	The next attribute is the request to use to get the next n items. It is used to paginate the alarm history list.

Example 6.66. List alarm history: JSON response

```
{
 "ialarm_history": [
 {
  "alarm_state": "clear",
  "severity": "major",
  "display_entity_instance_id": "system=c4065e96-4f2f-42d5-9f55-b89f84a8263a.
service_domain=controller.service_group=oam-services",
  "timestamp": "2014-11-28T16:56:44.814747",
  "uuid": "67f88971-d769-450e-9e8a-126dd8585187",
  "alarm_id": "400.002",
  "entity_instance_id": "system=1bcd6f11-8152-45f7-9d93-0960e1887afe.
service_domain=controller.service_group=oam-services",
  "reason_text": "Service group loss of redundancy; expected 1 standby member
 but no standby members available."
 },
 {
  "alarm_state": "clear",
  "severity": "critical",
  "display_entity_instance_id": "system=c4065e96-4f2f-42d5-9f55-b89f84a8263a.
service_domain=controller.service_group=patching-services.host=controller-0",
  "timestamp": "2014-11-28T16:56:44.808965",
  "uuid": "1a259ab9-8ea2-4177-8053-ad7596509c66",
  "alarm_id": "400.001",
  "entity_instance_id": "system=1bcd6f11-8152-45f7-9d93-0960e1887afe.
service_domain=controller.service_group=patching-services.host=controller-0",
  "reason_text": "Service group failure; patch-alarm-manager(disabled,
 failed)."
 }],
  "next": "http://192.168.204.2:6385/v1/ialarm_history?sort_key=timestamp&
sort_dir=desc&limit=2&marker=1a259ab9-8ea2-4177-8053-ad7596509c66"
}
```

This operation does not return a response body.

6.19.2. Show alarm history

Method	URI	Description
GET	`/v1/ialarm_history/{alarm_uuid}`	Shows information about a specific alarm history.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

6.19.2.1. Request

This table shows the URI parameters for the show alarm history request:

Name	Type	Description
`{alarm_uuid}`	UUID	The unique identifier of an alarm history.

This operation does not accept a request body.

6.19.2.2. Response

This table shows the body parameters for the show alarm history response:

Name	Type	Description
`alarm_id`	String *(Optional)*	The alarm ID; each type of alarm has a unique ID. Note the alarm_id and the entity_instance_id uniquely identify an alarm instance.
`entity_instance_id`	String *(Optional)*	The instance of the object raising alarm. A . separated list of sub-entity-type=instance-value pairs, representing the containment structure of the overall entity instance. Note the alarm_id and the entity_instance_id uniquely identify an alarm instance.
`display_entity_instance_id`	String *(Optional)*	A local and more human-readable version of the entity_instance_id. The system prefix is removed and UUID values are replaced with names where possible (e.g. ports, interfaces, providernets, etc.).
`reason_text`	String *(Optional)*	The text description of the alarm.
`severity`	String *(Optional)*	The severity of the alarm; `critical`, `major`, `minor`, or `warning`.
`timestamp`	DateTime *(Optional)*	The time in UTC at which the alarm has last been updated.
`uuid`	String *(Optional)*	The unique identifier of the alarm.
`alarm_state`	String *(Optional)*	The state of the alarm; `set` or `clear`
`service_affecting`	String	Indicates whether the alarm affects the service.

Name	Type	Description
	(Optional)	
`proposed_repair_action`	String *(Optional)*	The proposed action to clear the alarm.
`alarm_type`	String *(Optional)*	The type of the alarm.
`entity_type_id`	String *(Optional)*	The type of the object raising the alarm. A . separated list of sub-entity-type, representing the containment structure of the overall entity type.
`probable_cause`	String *(Optional)*	The probable cause of the alarm.
`suppression`	String *(Optional)*	Indicates whether suppression of the specific alarm is allowed.
`uuid`	UUID *(Optional)*	The universally unique identifier for this object.
`links`	List *(Optional)*	For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The following types of link relations are associated with resources: a self link containing a versioned link to the resource, and a bookmark link containing a permanent link to a resource that is appropriate for long term storage.
`created_at`	DateTime *(Optional)*	The time when the object was created.
`updated_at`	DateTime *(Optional)*	The time when the object was last updated.

Example 6.67. Show alarm history: JSON response

```
{
  "alarm_state": "set",
  "service_affecting": "True",
  "proposed_repair_action":
  "Contact next level of support.",
  "alarm_type": "processing-error",
  "severity": "critical",
  "created_at": "2014-11-28T20:00:56.116251+00:00",
  "entity_type_id": "service_domain.service_group.host",
  "probable_cause": "underlying-resource-unavailable",
  "updated_at": null,
  "alarm_id": "400.001",
  "entity_instance_id":"system=1bcd6f11-8152-45f7-9d93-0960e1887afe.
service_domain=controller.service_group=patching-services.host=controller-0",
  "suppression": "True",
  "timestamp": "2014-11-28T20:00:56.114060+00:00",
  "uuid": "8701d806-cae7-4f34-be8d-17fd11a0d25d",
  "display_entity_instance_id": "system=c4065e96-4f2f-42d5-9f55-b89f84a8263a.
service_domain=controller.service_group=patching-services.host=controller-0",
  "reason_text": "Service group failure; patch-alarm-manager(disabled, failed).
"
}
```

This operation does not return a response body.

6.20. Customer Log

These APIs allow the display of the Customer Log in the system.

Method	URI	Description
GET	`/v1/icustomer_log{?q(Optional), limit(Optional)}`	Lists all customer logs based on specified query. The logs are returned in reverse chronological order.
GET	`/v1/icustomer_log/{log_uuid}`	Shows information about a specific customer log.

6.20.1. List customer logs

Method	URI	Description
GET	`/v1/icustomer_log{?q(Optional), limit(Optional)}`	Lists all customer logs based on specified query. The logs are returned in reverse chronological order.

The supported query options are log_id, entity_type_id, entity_instance_id, severity, log_type, start and end.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

6.20.1.1. Request

This table shows the query parameters for the list customer logs request:

Name	Type	Description
`q(Optional)`	List *(Optional)*	This parameter specifies filter rules for the logs to be returned.
`limit(Optional)`	Int *(Optional)*	This parameter specifies the maximum number of logs to be returned.

Example 6.68. List customer logs: JSON request

```
http://192.168.204.2:6385/v1/icustomer_log?q.field=start&q.field=end&q.
op=eq&q.op=eq&q.type=&q.type=&q.value=2014-11-28T16%3A56%3A44&q.value=
2014-11-28T16%3A56%3A45&limit=2
```

This operation does not accept a request body.

6.20.1.2. Response

This table shows the body parameters for the list customer logs response:

Name	Type	Description
`icustomer_log`	List *(Optional)*	The list of customer log based on the specified query.
`log_id`	String *(Optional)*	The log ID; each type of log has a unique ID. Note the log_id and the entity_instance_id uniquely identify a log instance.
`entity_instance_id`	String *(Optional)*	The instance of the object generating log. A . separated list of sub-entity-type=instance-value pairs, representing the containment structure of the overall entity instance. Note the log_id and the entity_instance_id uniquely identify a log instance.
`display_entity_instance_id`	String *(Optional)*	A local and more human-readable version of the entity_instance_id. The system prefix is removed and UUID values are replaced with names where possible (e.g. ports, interfaces, providernets, etc.).
`reason_text`	String	The text description of the customer log.

Name	Type	Description
	(Optional)	
`severity`	String *(Optional)*	The severity of the customer log; `critical`, `major`, `minor`, or `warning`.
`timestamp`	DateTime *(Optional)*	The time in UTC at which the log has last been updated.
`uuid`	String *(Optional)*	The unique identifier of the customer log.
`next`	String *(Optional)*	The next attribute is the request to use to get the next n items. It is used to paginate the customer log list.

Example 6.69. List customer logs: JSON response

```
{
 "icustomer_log": [
 {
  "severity": "major",
  "display_entity_instance_id": "system=c4065e96-4f2f-42d5-9f55-b89f84a8263a.
service_domain=controller.service_group=oam-services",
  "timestamp": "2014-11-28T16:56:44.814747",
  "uuid": "67f88971-d769-450e-9e8a-126dd8585187",
  "log_id": "400.002",
  "entity_instance_id": "system=1bcd6f11-8152-45f7-9d93-0960e1887afe.
service_domain=controller.service_group=oam-services",
  "reason_text": "Service group loss of redundancy; expected 1 standby member
 but no standby members available."
 },
 {
  "severity": "critical",
  "display_entity_instance_id": "system=c4065e96-4f2f-42d5-9f55-b89f84a8263a.
service_domain=controller.service_group=patching-services.host=controller-0",
  "timestamp": "2014-11-28T16:56:44.808965",
  "uuid": "1a259ab9-8ea2-4177-8053-ad7596509c66",
  "log_id": "400.001",
  "entity_instance_id": "system=1bcd6f11-8152-45f7-9d93-0960e1887afe.
service_domain=controller.service_group=patching-services.host=controller-0",
  "reason_text": "Service group failure; patch-alarm-manager(disabled,
 failed)."
 }],
  "next": "http://192.168.204.2:6385/v1/icustomer_log?sort_key=timestamp&
sort_dir=desc&limit=2&marker=1a259ab9-8ea2-4177-8053-ad7596509c66"
}
```

This operation does not return a response body.

6.20.2. Show customer log

Method	URI	Description
GET	`/v1/icustomer_log/{log_uuid}`	Shows information about a specific customer log.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

6.20.2.1. Request

This table shows the URI parameters for the show customer log request:

Name	Type	Description
`{log_uuid}`	UUID	The unique identifier of a customer log.

This operation does not accept a request body.

6.20.2.2. Response

This table shows the body parameters for the show customer log response:

Name	Type	Description
`log_id`	String *(Optional)*	The log ID; each type of log has a unique ID. Note the log_id and the entity_instance_id uniquely identify a log instance.
`entity_instance_id`	String *(Optional)*	The instance of the object generating log. A . separated list of sub-entity-type=instance-value pairs, representing the containment structure of the overall entity instance. Note the log_id and the entity_instance_id uniquely identify a log instance.
`display_entity_instance_id`	String *(Optional)*	A local and more human-readable version of the entity_instance_id. The system prefix is removed and UUID values are replaced with names where possible (e.g. ports, interfaces, providernets, etc.).
`reason_text`	String *(Optional)*	The text description of the customer log.
`severity`	String *(Optional)*	The severity of the customer log; `critical`, `major`, `minor`, or `warning`.
`timestamp`	DateTime *(Optional)*	The time in UTC at which the log has last been updated.
`uuid`	String *(Optional)*	The unique identifier of the customer log.
`next`	String *(Optional)*	The next attribute is the request to use to get the next n items. It is used to paginate the customer log list.
`uuid`	UUID *(Optional)*	The universally unique identifier for this object.

Name	Type	Description
links	List *(Optional)*	For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The following types of link relations are associated with resources: a self link containing a versioned link to the resource, and a bookmark link containing a permanent link to a resource that is appropriate for long term storage.
created_at	DateTime *(Optional)*	The time when the object was created.
updated_at	DateTime *(Optional)*	The time when the object was last updated.

Example 6.70. Show customer log: JSON response

```
{
 "service_affecting": "True",
 "log_type": "processing-error",
 "severity": "critical",
 "created_at": "2014-11-28T20:00:56.116251+00:00",
 "entity_type_id": "service_domain.service_group.host",
 "probable_cause": "underlying-resource-unavailable",
 "updated_at": null,
 "log_id": "400.001",
 "entity_instance_id":"system=1bcd6f11-8152-45f7-9d93-0960e1887afe.
service_domain=controller.service_group=patching-services.host=controller-0",
 "timestamp": "2014-11-28T20:00:56.114060+00:00",
 "uuid": "8701d806-cae7-4f34-be8d-17fd11a0d25d",
 "display_entity_instance_id": "system=c4065e96-4f2f-42d5-9f55-b89f84a8263a.
service_domain=controller.service_group=patching-services.host=controller-0",
 "reason_text": "Service group failure; patch-alarm-manager(disabled, failed).
"
}
```

This operation does not return a response body.

6.21. SNMP Communities

Method	URI	Description
GET	/v1/icommunity	Lists all SNMP Communities.
GET	/v1/icommunity/{community_id}	Shows information about a specific SNMP Community.
POST	/v1/icommunity	Creates a SNMP Community.
PATCH	/v1/icommunity/{community_id}	Modifies a specific SNMP Community.
DELETE	/v1/icommunity/{community_id}	Deletes a specific SNMP Community.

6.21.1. List communities

Method	URI	Description
GET	`/v1/icommunity`	Lists all SNMP Communities.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404)

6.21.1.1. Request

This operation does not accept a request body.

6.21.1.2. Response

This table shows the body parameters for the list communities response:

Name	Type	Description
`icommunities`	List (Optional)	The list of SNMP Communities.
`access`	String (Optional)	The SNMP GET/SET access control for a specific community.
`community`	String (Optional)	The community string of which the SNMP client is a member.
`view`	String (Optional)	The SNMP MIB view of which the community has access to.
`uuid`	UUID (Optional)	The universally unique identifier for this object.
`links`	List (Optional)	For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The following types of link relations are associated with resources: a self link containing a versioned link to the resource, and a bookmark link containing a permanent link to a resource that is appropriate for long term storage.
`created_at`	DateTime (Optional)	The time when the object was created.
`updated_at`	DateTime (Optional)	The time when the object was last updated.

Example 6.71. List communities: JSON response

```
{
  "icommunity": [
    {
"access": "ro",
```

```
"uuid": "744cddaa-8a24-4573-aa0e-4f8b535d95b7",
"community": "new",
"view": ".1"
    },
    {
"access": "ro",
"uuid": "73706882-9d7c-4a8f-9409-185ffee0066c",
"community": "guest",
"view": ".1"
    }
  ]
}
```

This operation does not return a response body.

6.21.2. Show community

Method	URI	Description
GET	`/v1/icommunity/{community_id}`	Shows information about a specific SNMP Community.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

6.21.2.1. Request

This table shows the URI parameters for the show community request:

Name	Type	Description
`{community_id}`	UUID	The unique identifier of an existing SNMP Community.

This operation does not accept a request body.

6.21.2.2. Response

This table shows the body parameters for the show community response:

Name	Type	Description
`access`	String *(Optional)*	The SNMP GET/SET access control for a specific community.
`community`	String *(Optional)*	The community string of which the SNMP client is a member.
`view`	String *(Optional)*	The SNMP MIB view of which the community has access to.
`uuid`	UUID *(Optional)*	The universally unique identifier for this object.
`links`	List *(Optional)*	For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The following types of link relations are associated with resources: a self link containing a versioned link to the resource, and a bookmark link containing a permanent link to a resource that is appropriate for long term storage.
`created_at`	DateTime *(Optional)*	The time when the object was created.
`updated_at`	DateTime *(Optional)*	The time when the object was last updated.

Example 6.72. Show community: JSON response

```
{
 "uuid": "73706882-9d7c-4a8f-9409-185ffee0066c",
```

```
 "created_at": "2014-09-24T20:06:54.386982+00:00",
 "updated_at": null,
 "community": "guest",
 "access": "ro",
 "view": ".1"
}
```

This operation does not return a response body.

6.21.3. Create community

Method	URI	Description
POST	`/v1/icommunity`	Creates a SNMP Community.

Normal response codes: 200

Error response codes: badMediaType (415)

6.21.3.1. Request

This table shows the body parameters for the create community request:

Name	Type	Description
`community`	String *(Optional)*	This parameter specifies the community string to create.

Example 6.73. Create community: JSON request

```
{
 "community": "guest"
}
```

This operation does not accept a request body.

6.21.3.2. Response

This table shows the body parameters for the create community response:

Name	Type	Description
`access`	String *(Optional)*	The SNMP GET/SET access control for a specific community.
`community`	String *(Optional)*	The community string of which the SNMP client is a member.
`view`	String *(Optional)*	The SNMP MIB view of which the community has access to.
`uuid`	UUID *(Optional)*	The universally unique identifier for this object.
`links`	List *(Optional)*	For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The following types of link relations are associated with resources: a self link containing a versioned link to the resource, and a bookmark link containing a permanent link to a resource that is appropriate for long term storage.
`created_at`	DateTime *(Optional)*	The time when the object was created.
`updated_at`	DateTime *(Optional)*	The time when the object was last updated.

Example 6.74. Create community: JSON response

```
{
 "uuid": "73706882-9d7c-4a8f-9409-185ffee0066c",
 "created_at": "2014-09-24T20:06:54.386982+00:00",
 "updated_at": null,
 "community": "guest",
 "access": "ro",
 "view": ".1"
}
```

This operation does not return a response body.

6.21.4. Modify community

Method	URI	Description
PATCH	`/v1/icommunity/{community_id}`	Modifies a specific SNMP Community.

Normal response codes: 200

Error response codes: badMediaType (415)

6.21.4.1. Request

This table shows the URI parameters for the modify community request:

Name	Type	Description
`{community_id}`	UUID	The unique identifier of an existing SNMP Community.

This table shows the body parameters for the modify community request:

Name	Type	Description
`community`	String *(Optional)*	This parameter specifies the new community string.

Example 6.75. Modify community: JSON request

```
[
  {
"path": "/community",
"value": "wrs",
"op": "replace"
  }
]
```

This operation does not accept a request body.

6.21.4.2. Response

This table shows the body parameters for the modify community response:

Name	Type	Description
`access`	String *(Optional)*	The SNMP GET/SET access control for a specific community.
`community`	String *(Optional)*	The community string of which the SNMP client is a member.
`view`	String *(Optional)*	The SNMP MIB view of which the community has access to.
`uuid`	UUID *(Optional)*	The universally unique identifier for this object.
`links`	List	For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The follow-

Name	Type	Description
	(Optional)	ing types of link relations are associated with resources: a self link containing a versioned link to the resource, and a bookmark link containing a permanent link to a resource that is appropriate for long term storage.
created_at	DateTime *(Optional)*	The time when the object was created.
updated_at	DateTime *(Optional)*	The time when the object was last updated.

Example 6.76. Modify community: JSON response

```
{
 "uuid": "744cddaa-8a24-4573-aa0e-4f8b535d95b7",
 "created_at": "2014-09-23T15:01:53.187164+00:00",
 "updated_at": "2014-09-24T19:46:40.138145+00:00",
 "community": "wrs",
 "access": "ro",
  "view": ".1"
}
```

This operation does not return a response body.

6.21.5. Delete community

Method	URI	Description
DELETE	`/v1/icommunity/{community_id}`	Deletes a specific SNMP Community.

Normal response codes: 204

6.21.5.1. Request

This table shows the URI parameters for the delete community request:

Name	Type	Description
`{community_id}`	UUID	The unique identifier of an existing SNMP Community.

This operation does not accept a request body.

6.21.5.2. Response

This operation does not return a response body.

6.22. SNMP Trap Destinations

Method	URI	Description
GET	`/v1/itrapdest`	Lists all SNMP Trap Destinations.
GET	`/v1/itrapdest/{trapdest_id}`	Shows information about a specific SNMP Trap Destination.
POST	`/v1/itrapdest`	Creates a SNMP Trap Destination.
PATCH	`/v1/itrapdest/{trapdest_id}`	Modifies a specific SNMP Trap Destination.
DELETE	`/v1/itrapdest/{trapdest_id}`	Deletes a specific SNMP Trap Destination.

6.22.1. List trap destinations

Method	URI	Description
GET	`/v1/itrapdest`	Lists all SNMP Trap Destinations.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNot-Found (404)

6.22.1.1. Request

This operation does not accept a request body.

6.22.1.2. Response

This table shows the body parameters for the list trap destinations response:

Name	Type	Description
`itrapdests`	List *(Optional)*	The list of SNMP Trap Destinations.
`ip_address`	String *(Optional)*	The IP address of a specific trap destination.
`community`	String *(Optional)*	The community of which the trap destination is a member.
`type`	String *(Optional)*	The SNMP version of the trap message for a specific destination.
`port`	String *(Optional)*	The port number of which SNMP manager is listening for traps.
`transport`	String *(Optional)*	The transport protocol used by the trap messages.
`uuid`	UUID *(Optional)*	The universally unique identifier for this object.
`links`	List *(Optional)*	For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The following types of link relations are associated with resources: a self link containing a versioned link to the resource, and a bookmark link containing a permanent link to a resource that is appropriate for long term storage.
`created_at`	DateTime *(Optional)*	The time when the object was created.
`updated_at`	DateTime *(Optional)*	The time when the object was last updated.

Example 6.77. List trap destinations: JSON response

```
{
  "itrapdest": [
    {
  "uuid": "fc33945c-7aba-4d83-9216-a60db7097a23", "links": [
    {
    "href": "http://192.168.204.2:6385/v1/itrapdest/fc33945c-7aba-4d83-9216-
a60db7097a23",
    "rel": "self"
  },
  {
    "href": "http://192.168.204.2:6385/itrapdest/fc33945c-7aba-4d83-9216-
a60db7097a23",
    "rel": "bookmark"
  }
  ],
  "ip_address": "10.10.10.1",
  "community": "cgts",
  "type": "snmpv2c_trap",
  "port": 162, "transport": "udp"
 },
 {
  "uuid": "22f0497c-0a09-41c4-8514-cb5afcbf930d", "links": [
    {
    "href": "http://192.168.204.2:6385/v1/itrapdest/22f0497c-0a09-41c4-8514-
cb5afcbf930d",
    "rel": "self"
  },
  {
    "href": "http://192.168.204.2:6385/itrapdest/22f0497c-0a09-41c4-8514-
cb5afcbf930d",
    "rel": "bookmark"
  }
  ],
  "ip_address": "27.134.0.8",
  "community": "sprint",
  "type": "snmpv2c_trap",
  "port": 162,
  "transport": "udp"
 }
  ]
}
```

This operation does not return a response body.

6.22.2. Show trap destination

Method	URI	Description
GET	`/v1/itrapdest/{trapdest_id}`	Shows information about a specific SNMP Trap Destination.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404)

6.22.2.1. Request

This table shows the URI parameters for the show trap destination request:

Name	Type	Description
`{trapdest_id}`	UUID	The unique identifier of an existing SNMP Trap Destination.

This operation does not accept a request body.

6.22.2.2. Response

This table shows the body parameters for the show trap destination response:

Name	Type	Description
`ip_address`	String *(Optional)*	The IP address of a specific trap destination.
`community`	String *(Optional)*	The community of which the trap destination is a member.
`type`	String *(Optional)*	The SNMP version of the trap message for a specific destination.
`port`	String *(Optional)*	The port number of which SNMP manager is listening for traps.
`transport`	String *(Optional)*	The transport protocol used by the trap messages.
`uuid`	UUID *(Optional)*	The universally unique identifier for this object.
`links`	List *(Optional)*	For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The following types of link relations are associated with resources: a self link containing a versioned link to the resource, and a bookmark link containing a permanent link to a resource that is appropriate for long term storage.
`created_at`	DateTime *(Optional)*	The time when the object was created.
`updated_at`	DateTime	The time when the object was last updated.

Name	Type	Description
	(Optional)	

Example 6.78. Show trap destination: JSON response

```
{
  "uuid": "22f0497c-0a09-41c4-8514-cb5afcbf930d", "links": [
    {
  "href": "http://192.168.204.2:6385/v1/itrapdest/22f0497c-0a09-41c4-8514-
cb5afcbf930d",
  "rel": "self"
    },
    {
  "href": "http://192.168.204.2:6385/itrapdest/22f0497c-0a09-41c4-8514-
cb5afcbf930d",
  "rel": "bookmark"
    }
  ],
 "type": "snmpv2c_trap",
 "created_at": "2014-09-24T21:09:02.842231+00:00",
 "updated_at": null,
 "community": "sprint",
 "ip_address": "27.134.0.8",
 "port": 162,
 "transport": "udp"
}
```

This operation does not return a response body.

6.22.3. Create trap destination

Method	URI	Description
POST	`/v1/itrapdest`	Creates a SNMP Trap Destination.

Normal response codes: 200

Error response codes: badMediaType (415)

6.22.3.1. Request

This table shows the body parameters for the create trap destination request:

Name	Type	Description
`ip_address`	String *(Optional)*	This parameter specifies the IP address of a new trap destination.
`community`	String *(Optional)*	This parameter specifies the community of which the trap destination is a member.

Example 6.79. Create trap destination: JSON request

```
{
 "ip_address": "27.134.0.8",
 "community": "sprint"
}
```

This operation does not accept a request body.

6.22.3.2. Response

This table shows the body parameters for the create trap destination response:

Name	Type	Description
`ip_address`	String *(Optional)*	The IP address of a specific trap destination.
`community`	String *(Optional)*	The community of which the trap destination is a member.
`type`	String *(Optional)*	The SNMP version of the trap message for a specific destination.
`port`	String *(Optional)*	The port number of which SNMP manager is listening for traps.
`transport`	String *(Optional)*	The transport protocol used by the trap messages.
`uuid`	UUID *(Optional)*	The universally unique identifier for this object.
`links`	List	For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The follow-

Name	Type	Description
	(Optional)	ing types of link relations are associated with resources: a self link containing a versioned link to the resource, and a bookmark link containing a permanent link to a resource that is appropriate for long term storage.
created_at	DateTime *(Optional)*	The time when the object was created.
updated_at	DateTime *(Optional)*	The time when the object was last updated.

Example 6.80. Create trap destination: JSON response

```
{
  "uuid": "22f0497c-0a09-41c4-8514-cb5afcbf930d", "links":
    [
      {
  "href": "http://192.168.204.2:6385/v1/itrapdest/22f0497c-0a09-41c4-8514-
cb5afcbf930d",
  "rel": "self"
      },
      {
  "href": "http://192.168.204.2:6385/itrapdest/22f0497c-0a09-41c4-8514-
cb5afcbf930d",
  "rel": "bookmark"
      }
    ],
 "type": "snmpv2c_trap",
 "created_at": "2014-09-24T21:09:02.842231+00:00",
 "updated_at": null,
 "community": "sprint",
 "ip_address": "27.134.0.8",
 "port": 162,
 "transport": "udp"
}
```

This operation does not return a response body.

6.22.4. Modify trap destination

Method	URI	Description
PATCH	`/v1/itrapdest/{trapdest_id}`	Modifies a specific SNMP Trap Destination.

Normal response codes: 200

Error response codes: badMediaType (415)

6.22.4.1. Request

This table shows the URI parameters for the modify trap destination request:

Name	Type	Description
`{trapdest_id}`	UUID	The unique identifier of an existing SNMP Trap Destination.

This table shows the body parameters for the modify trap destination request:

Name	Type	Description
`ip_address`	String *(Optional)*	This parameter specifies the IP address of a specific trap destination.
`community`	String *(Optional)*	This parameter specifies the the community of which the trap destination is a member.

Example 6.81. Modify trap destination: JSON request

```
[
  {
"path": "/ip_address",
"value": "47.10.1.128",
"op": "replace"
  },
  {
"path": "/community",
"value": "sprint",
"op": "replace"
  }
]
```

This operation does not accept a request body.

6.22.4.2. Response

This table shows the body parameters for the modify trap destination response:

Name	Type	Description
`ip_address`	String *(Optional)*	The IP address of a specific trap destination.
`community`	String	The community of which the trap destination is a member.

Name	Type	Description
	(Optional)	
type	String *(Optional)*	The SNMP version of the trap message for a specific destination.
port	String *(Optional)*	The port number of which SNMP manager is listening for traps.
transport	String *(Optional)*	The transport protocol used by the trap messages.
uuid	UUID *(Optional)*	The universally unique identifier for this object.
links	List *(Optional)*	For convenience, resources contain links to themselves. This allows a client to easily obtain rather than construct resource URIs. The following types of link relations are associated with resources: a self link containing a versioned link to the resource, and a bookmark link containing a permanent link to a resource that is appropriate for long term storage.
created_at	DateTime *(Optional)*	The time when the object was created.
updated_at	DateTime *(Optional)*	The time when the object was last updated.

Example 6.82. Modify trap destination: JSON response

```
{
  "uuid": "22f0497c-0a09-41c4-8514-cb5afcbf930d", "links": [
    {
  "href": "http://192.168.204.2:6385/v1/itrapdest/22f0497c-0a09-41c4-8514-
cb5afcbf930d",
  "rel": "self"
    },
    {
  "href": "http://192.168.204.2:6385/itrapdest/22f0497c-0a09-41c4-8514-
cb5afcbf930d",
  "rel": "bookmark"
    }
  ],
 "type": "snmpv2c_trap",
 "created_at": "2014-09-24T21:09:02.842231+00:00",
 "updated_at": "2014-09-24T21:13:51.061300+00:00",
 "community": "sprint",
 "ip_address": "47.10.1.128",
 "port": 162,
 "transport": "udp"
}
```

This operation does not return a response body.

6.22.5. Delete trap destination

Method	URI	Description
DELETE	`/v1/itrapdest/{trapdest_id}`	Deletes a specific SNMP Trap Destination.

Normal response codes: 204

6.22.5.1. Request

This table shows the URI parameters for the delete trap destination request:

Name	Type	Description
`{trapdest_id}`	UUID	The unique identifier of an existing SNMP Trap Destination.

This operation does not accept a request body.

6.22.5.2. Response

This operation does not return a response body.

7. Patching API v1

Manage the patching of hosts with the Patching API. This includes upload, application, installation, removal, deletion, and querying.

The typical port used for the Patching REST API is 15491. However, proper technique would be to look up the patching service endpoint in keystone.

Method	URI	Description
API versions		
GET	/	Lists information about all Patching API versions.
Patches		
GET	/v1/query	Lists all patches in the patching system.
GET	/v1/show/{patch_id}	Shows detailed information about a specific patch.
POST	/v1/upload	Uplaods a patch to the patching system.
POST	/v1/apply/{patch_id}	Applies a patch which is in the Available state.
POST	/v1/remove/{patch_id}	Removes a patch which is in the Applied state.
POST	/v1/delete/{patch_id}	Deletes a patch which is in the Available state.
Hosts		
GET	/v1/query_hosts	Lists all host entities and their patching information.
POST	/v1/host_install/{hostname}	Trigger a host install on the specified host.

7.1. API versions

Method	URI	Description
GET	/	Lists information about all Patching API versions.

7.1.1. List API versions

Method	URI	Description
GET	/	Lists information about all Patching API versions.

Normal response codes: 200, 300

Error response codes: serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404)

7.1.1.1. Request

This operation does not accept a request body.

7.1.1.2. Response

Example 7.1. List API versions: JSON response

```
"Titanium Server Patching API, Available versions: /v1"
```

This operation does not return a response body.

7.2. Patches

The patches used by the patching service to update individual hosts in the cloud.

Method	URI	Description
GET	/v1/query	Lists all patches in the patching system.
GET	/v1/show/{patch_id}	Shows detailed information about a specific patch.
POST	/v1/upload	Uplaods a patch to the patching system.
POST	/v1/apply/{patch_id}	Applies a patch which is in the Available state.
POST	/v1/remove/{patch_id}	Removes a patch which is in the Applied state.
POST	/v1/delete/{patch_id}	Deletes a patch which is in the Available state.

7.2.1. List Patches

Method	URI	Description
GET	`/v1/query`	Lists all patches in the patching system.

Supported query values are `all`, `available`, or `applied`.

Normal response codes: 200

Error response codes: serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404)

7.2.1.1. Request

This operation does not accept a request body.

7.2.1.2. Response

This table shows the body parameters for the list patches response:

Name	Type	Description
`pd`	List *(Optional)*	The list of patches present in the patching system.
`patch`	List *(Optional)*	A patch present in the patching system.
`status`	String *(Optional)*	The status of the patch.
`sw_version`	String *(Optional)*	The software version the patch is intended for.
`install_instructions`	String *(Optional)*	Instructions on how to install the patch.
`description`	String *(Optional)*	The description of any updates present in this patch.
`warnings`	String *(Optional)*	Any warnings associated with the usage of the patch.
`summary`	String *(Optional)*	A brief summary of the patch.
`repostate`	String *(Optional)*	Whether this patch`s content`s have been added to the patching repository; `Applied` or `Available`.
`patchstate`	String *(Optional)*	The state of this patch`s application to hosts; `Available`, `Partial-Apply`, `Applied`, or `Partial-Removed`.
`requires`	List *(Optional)*	A list of patch ids required for this patch to be installed.

Example 7.2. List Patches: JSON response

```
{
    'pd':{
        'TS_14.10_PATCH_0002':{
            'status': 'REL',
            'sw_version': '14.10',
            'patchstate': 'Partial-Remove',
            'description': 'Fixes the following Issues:\n    CGTS-882
compute-4 and storage-0 multiple resets after DOR\n    CGTS-883 Alarms bogged
down for 1 hour after DOR\n    CGTS-894 Guest Heartbeat cannot be enabled
from horizon',
            'warnings': '',
            'summary': 'TS_14.10 Patch 0002',
            'repostate': 'Available',
            'install_instructions': '',
            'requires': []
        },
        'TS_14.10_PATCH_0001':{
            'status': 'REL',
            'sw_version': '14.10',
            'patchstate': 'Applied',
            'description': 'Fixes the following Issues:\n    CGTS-896:
hbsClient instrumentation can cause server reset or hang after long soaks',
            'warnings': '',
            'summary': 'TS_14.10 Patch 0001',
            'repostate': 'Applied',
            'install_instructions': 'No special install instructions.',
            'requires': []
        }
    }
}
```

This operation does not return a response body.

7.2.2. Show Patch

Method	URI	Description
GET	`/v1/show/{patch_id}`	Shows detailed information about a specific patch.

Normal response codes: 200

Error response codes: serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404)

7.2.2.1. Request

This operation does not accept a request body.

7.2.2.2. Response

This table shows the body parameters for the show patch response:

Name	Type	Description
`contents`	List *(Optional)*	The RPMs contained within the patch.
`patch`	List *(Optional)*	A patch present in the patching system.
`pkg`	String *(Optional)*	A package included in a patch.
`error`	String *(Optional)*	Any errors associated with the patch.
`metadata`	List *(Optional)*	Metadata associated with the patch.
`status`	String *(Optional)*	The status of the patch.
`sw_version`	String *(Optional)*	The software version the patch is intended for.
`install_instructions`	String *(Optional)*	Instructions on how to install the patch.
`description`	String *(Optional)*	The description of any updates present in this patch.
`warnings`	String *(Optional)*	Any warnings associated with the usage of the patch.
`summary`	String *(Optional)*	A brief summary of the patch.
`repostate`	String *(Optional)*	Whether the patch content has been added to the patching repository; `Applied` or `Available`.

Name	Type	Description
patchstate	String *(Optional)*	The state of the patch regarding application to hosts; `Available`, `Partial-Apply`, `Applied`, or `Partial-Removed`.
requires	List *(Optional)*	A list of patch ids required for this patch to be installed.

Example 7.3. Show Patch: JSON response

```
{
    "contents": {
        "TS_14.10_PATCH_0002": [
            "python-horizon-2013.2.3-r118.x86_64.rpm",
            "sysinv-1.0-r81.x86_64.rpm"
        ]
    },
    "error": "",
    "metadata": {
        "TS_14.10_PATCH_0002": {
            "description": "Fixes the following Issues:\n    CGTS-882
 compute-4 and storage-0 multiple resets after DOR",
            "install_instructions": "",
            "patchstate": "Partial-Remove",
            "repostate": "Available",
            "requires": [],
            "status": "DEV",
            "summary": "TS_14.10 Patch 0002",
            "sw_version": "14.10",
            "warnings": ""
        }
    }
}
```

This operation does not return a response body.

7.2.3. Upload Patch

Method	URI	Description
POST	`/v1/upload`	Uplaods a patch to the patching system.

Note that only one patch may be added per request

Normal response codes: 200

Error response codes: badMediaType (415)

7.2.3.1. Response

This table shows the body parameters for the upload patch response:

Name	Type	Description
`info`	String *(Optional)*	Any information regarding the request processing.
`warning`	String *(Optional)*	Any warnings generated during the request processing.
`error`	String *(Optional)*	Any errors generated during the request processing.

Example 7.4. Upload Patch: JSON response

```
{
    "info": "TS_14.10_PATCH_0001 is now available\n",
    "warning": "",
    "error": ""
}
```

This operation does not return a response body.

7.2.4. Apply Patch

Method	URI	Description
POST	`/v1/apply/{patch_id}`	Applies a patch which is in the Available state.

Normal response codes: 200

Error response codes: badMediaType (415)

7.2.4.1. Request

This operation does not accept a request body.

7.2.4.2. Response

This table shows the body parameters for the apply patch response:

Name	Type	Description
`info`	String *(Optional)*	Any information regarding the request processing.
`warning`	String *(Optional)*	Any warnings generated during the request processing.
`error`	String *(Optional)*	Any errors generated during the request processing.

Example 7.5. Apply Patch: JSON response

```
{
    "info": "TS_14.10_PATCH_0001 has been applied\n",
    "warning": "",
    "error": ""
}
```

This operation does not return a response body.

7.2.5. Remove Patch

Method	URI	Description
POST	`/v1/remove/{patch_id}`	Removes a patch which is in the Applied state.

Normal response codes: 200

Error response codes: badMediaType (415)

7.2.5.1. Request

This operation does not accept a request body.

7.2.5.2. Response

This table shows the body parameters for the remove patch response:

Name	Type	Description
`info`	String *(Optional)*	Any information regarding the request processing.
`warning`	String *(Optional)*	Any warnings generated during the request processing.
`error`	String *(Optional)*	Any errors generated during the request processing.

Example 7.6. Remove Patch: JSON response

```
{
    "info": "TS_14.10_PATCH_0001 has been removed from the repo\n",
    "warning": "",
    "error": ""
}
```

This operation does not return a response body.

7.2.6. Delete Patch

Method	URI	Description
POST	`/v1/delete/{patch_id}`	Deletes a patch which is in the Available state.

Normal response codes: 200

Error response codes: badMediaType (415)

7.2.6.1. Request

This operation does not accept a request body.

7.2.6.2. Response

This table shows the body parameters for the delete patch response:

Name	Type	Description
`info`	String *(Optional)*	Any information regarding the request processing.
`warning`	String *(Optional)*	Any warnings generated during the request processing.
`error`	String *(Optional)*	Any errors generated during the request processing.

Example 7.7. Delete Patch: JSON response

```
{
    "info": "TS_14.10_PATCH_0001 has been deleted\n",
    "warning": "",
    "error": ""
}
```

This operation does not return a response body.

7.3. Hosts

Hosts are the physical hosts or servers for the system as viewed by the patching service.

Method	URI	Description
GET	`/v1/query_hosts`	Lists all host entities and their patching information.
POST	`/v1/host_install/{hostname}`	Trigger a host install on the specified host.

7.3.1. List hosts

Method	URI	Description
GET	`/v1/query_hosts`	Lists all host entities and their patching information.

Normal response codes: 200

Error response codes: serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404)

7.3.1.1. Request

This operation does not accept a request body.

7.3.1.2. Response

This table shows the body parameters for the list hosts response:

Name	Type	Description
`data`	List *(Optional)*	The list of host entities.
`requires_reboot`	Boolean *(Optional)*	Indicates whether the host requires a reboot.
`nodetype`	String *(Optional)*	The type of the host; `controller`, `compute` or `storage`.
`missing_pkgs`	List *(Optional)*	The list of packages missing from this host.
`ip`	String *(Optional)*	The ip address of the host.
`hostname`	String *(Optional)*	The name of the host.
`installed`	List *(Optional)*	The packages installed on this host by the patching system.
`secs_since_ack`	Integer *(Optional)*	The number of seconds since the host last reported its status.
`patch_failed`	Boolean *(Optional)*	Indicates whether a patch installation has failed on the host.
`stale_details`	Boolean *(Optional)*	Indicates whether the details of this host are out of date.
`patch_current`	Boolean *(Optional)*	Indicates whether the host is up to date regarding patches.
`to_remove`	List *(Optional)*	The list of packages that are to be removed from the host.

Example 7.8. List hosts: JSON response

```
{
    'data': [
        {
                'hostname': 'controller-0',
                'nodetype': 'controller',
                'patch_failed': False,
                'ip': u'192.168.204.3',
                'requires_reboot': False,
                'installed': {},
                'secs_since_ack': 18,
                'missing_pkgs': [],
                'patch_current': True,
                'stale_details': False,
                'to_remove': []
        },
        {       'hostname': 'compute-0',
                'nodetype': 'compute',
                'patch_failed': False,
                'ip': u'192.168.204.27',
                'requires_reboot': False,
                'installed': {},
                'secs_since_ack': 18,
                'missing_pkgs': [],
                'patch_current': True,
                'stale_details': False,
                'to_remove': []}
    ]
}
```

This operation does not return a response body.

7.3.2. Host Install

Method	URI	Description
POST	`/v1/host_install/{hostname}`	Trigger a host install on the specified host.

The host must be in the Locked-Disabled-Online state.

Normal response codes: 200

Error response codes: serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), badMediaType (415)

7.3.2.1. Request

This operation does not accept a request body.

7.3.2.2. Response

This table shows the body parameters for the host install response:

Name	Type	Description
`info`	String *(Optional)*	Any information regarding the request processing.
`warning`	String *(Optional)*	Any warnings generated during the request processing.
`error`	String *(Optional)*	Any errors generated during the request processing.

Example 7.9. Host Install: JSON response

```
{
    "info": "Patch installation was successful on compute-0.\n",
    "warning": "",
    "error": ""
}
```

This operation does not return a response body.