Densify
Virtualization Optimizer

API Reference Guide

Version: 14.8

August 2023

Contents

Introduction

The Densify API is the foundation for the next generation of capacity intelligence for your cloud infrastructure. The Densify API connects to your other infrastructure software components such as cloud stacks and provides rich capacity information and recommended optimization actions. With the Densify API, these systems become aware of the overall capacity and can make better automatic choices with a fully-informed model of the infrastructure.

The Densify API is a REST-based web service that is designed for the following:

- to route Workloads to the appropriate location (datacenter, hosting venue, host, datastore) according to rich data:
  - security requirements;
  - business constraints;
  - technical constraints;
  - utilization;
  - host and datastore health status;
- to reserve space in the future for new applications coming online;
- to connect to your cloud platforms and collect cloud infrastructure data;
- perform optimization **analysis**[1] on the collected data;

[1] Collected data is analyzed using Densify's patented analytics and results are loaded into the reporting database for viewing through various reports, consoles, dashboards, and user interfaces.

- provide recommendations[1] to reduce risk and cost, for automated execution or downstream processing;

RESTful API

The Densify API has been implemented as a RESTful web service. Requests and responses are built around the transfer of representations of resources.

A resource is either a request to route a workload or the workloads themselves. For a full list of resources, see Object Reference.

The Densify RESTful web API is a web API implemented using HTTP. It is a collection of resources, with three defined aspects:

- the base URI for the web API, such as `http://localhost:8086/CIRBA/api/v2/`.
- the Internet media type of the data supported by the web API. The Densify API supports JSON.
- the set of operations supported by the web API using HTTP methods (e.g. `GET`, `PUT`, `POST`, or `DELETE`).

The following table shows the HTTP methods that are used to implement the web API.

Table: RESTful Web API HTTP Methods

Resource	GET	PUT	POST	DELETE
Collection URI, such as `http://localhost:8086/CIRBA/api/v2/<`*`resource`*`>/`	List the URIs and perhaps other details of the collection's members.	Not used.	Create a new entry in the collection. The new entry's URI is assigned automatically and is usually returned by the operation.	Not used.
Element URI, such as `http://localhost:8086/CIRBA/api/v2/<`*`resource`*`>/<`*`item`*`>`	Retrieve a representation of the addressed member of the collection, expressed in an appropriate	Modify the addressed member.	Not generally used, but is used to modify attributes and multiple objects within a given	Delete the addressed member of the collection.

[1] One or more actions for moving or resizing cloud or VM instances that will result in improved peformance and/or reduced cost.

Resource	GET	PUT	POST	DELETE
	Internet media type.		collection.	

Base URL v1 Versus v2

Note*: The API v1 version is now deprecated and end of life (EoL) is January 1, 2018.*

The v2 version of the Densify API extends v1 with the support for multiple disks and capacity units. To update your v1 programs, see section *API Enhancements* in *New Features* of the 8.2 *Release Notes*. You can access v2 through the following base URL:

```
http://<Densify-server>:8086/CIRBA/api/v2
```

For clarity in this document, all references to a resource URI assume that the base URL is at the beginning. For example, when you see:

```
/routing-requests
```

this means:

```
http://<Densify-server>:8086/CIRBA/api/v2/routing-requests
```

Hybrid Routing

For backwards compatibility with earlier releases of the API, the API objects and elements have not been renamed but have been extended to support hybrid routing. The API has been extended to route to full control hosting venues as well as non-control and guest-level hosting venues.

- Control Environments—This object is an environment consisting of either full control hosting venues, non-control hosting venues, or guest-level hosting venues.
- Infrastructure Groups—This object is a hosting venue, either a full control hosting venue, a non-control hosting venue or a guest-level hosting venue.

For the description of each resource object, a section is added to describe the API difference between hosting venues, as appropriate.

Additional Documentation

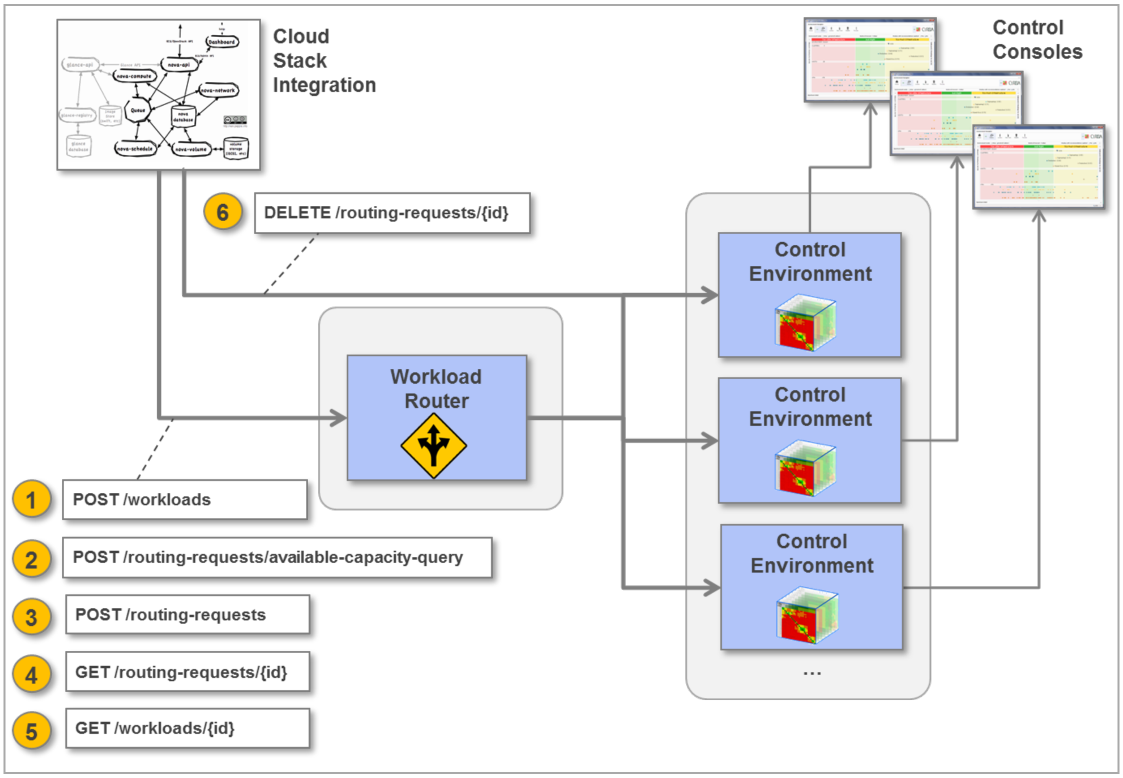
In order to perform API requests and work with routing workloads, a good understanding of Densify and the Route and Reserve Demand page is recommended. Refer to the following documentation for detailed information:

- *Working with the Control Console* (Help Topic ID 230580)
- *Routing and Reserving Demand* (Help Topic ID 330130)

Using the API

API Work Flow

Figure: Work Flow Diagram



There are six basic steps in the API workflow:

1. Send a `POST` to `/workloads` to define each workload (minimally specifying the expected `name` and `catalog_spec`, but typically including `expected_date`, `workload_profile` and `disks`, as well as preferred placement (elements `pref_control_environment` and `pref_infrastructure_group`).
2. Optionally, send a `POST` to `/routing-requests/available-capacity-query` with the Workloads defined in step 1. This is used to manually determine the best environment, hosting venue, and datastores for placement based on cost, capacity and Fit-for-Purpose validation.

   At this point, you can optionally modify the Workloads.
3. Send a `POST` to `/routing-requests` with the defined Workloads. This request returns a `/routing-request/<`*`requestid`*`>` so that you can reference the Routing Request in the next step.
4. Send a `GET /routing-requests/<`*`requestid`*`>` request to check the routing status until Densify is finished analyzing the request.
5. When the Routing Request is `PLACED`, send a `GET /workloads/<`*`workloadid`*`>` request to get the placement of each workload (elements `control_environment`, `infrastructure_group`, `host` and `sensors`).
6. After you receive the placement you can continue referring to the Routing Request until you delete it by sending a `DELETE /routing-requests/<`*`requestid`*`>` request.

Authentication

API-enabled User Credentials

To access the API, you need Densify API-enabled user credentials. An API-enabled user has membership in the or more of the following Densify user groups:

- SaaS_User_Admin;
- SaaS_User

Authentication Methods

Densify supports only token-based authentication for securing API requests. Densify employs the JSON Web Token (JWT) standard. You need to obtain a signed token using an `/authorize` request, which can then be used for authenticating subsequent Densify API calls. See Authorize on page 218 for details.

Configuration Settings

The following configuration settings can affect your API requests.

Table: API and Report Settings

Setting	Description	Default Value
API Default Catalog Specification (key=rest.api.catalogSpec.default)	If the Workload is created without specifying the `catalog_spec`, the value here is used. Densify API internal setting.	win-medium-4gb
API Last Hours of Cluster Refresh (key=rest.api.WithinHoursFromLastRefreshOfCluster)	The number of hours since the infrastructure group was last refreshed before the group and owning environment are considered `UNAVAILABLE` for routing (i.e. not qualified for available capacity query and placement). Densify API internal setting.	48
API Detailed Logging (key=rest.api.logging)	Enables additional logging of API requests to the Densify Web Server log file %CIRBA_HOME%\logs\cirba-webserver.log.	OFF
API Page Size (key=rest.api.paging.pageSize)	The number of objects returned when performing a `GET` request on a collection of objects. To override the page size, use the following syntax (see Paging on page 35 for details): `GET /<resource object>/?page=<pagenumber>[&page_size=<pagesize>]`	100
API Default Routing Strategy (key=rest.api.routingStrategy.default)	Specifies the default Routing Strategy when calculating hosting scores used by the API and Route and Reserve Demand page. ▫ Capacity—The hosting score is calculated	Capacity

Setting	Description	Default Value
	according to the available capacity of the Hosting Venues. The more capacity available, the higher the score. - Relative Cost—The hosting score is calculated according to the cost while ensuring there is enough capacity. The higher the cost, the lower the score. - Blended—The hosting score is calculated as the average of the Capacity and Relative Cost scores.	
API Key Rotation (Days) (key=rest.api.key.rotation)	Specifies the number of days the Densify API key is rotated.	30
API Token Expiring (Minutes) (key=rest.api.token.expiry)	Specifies the JWT API token expiry time, in minutes. Configuring a short expiry time is recommended.	5
Enable Password Policy (key=password.policy.enabled)	Enables the password policy and enforces password character, length and lockout policies. See *Managing Configuration Settings> Password Policy* (Help Topic ID 111250). If this setting is set to TRUE, Densify API user accounts are subjected to the Password policy rules on login attempts, expiry date, and inactivity lockout. ***Note****: The Densify API does not enforce Character and Length Password policies during authentication.*	FALSE

Contact Support@Densify.com for details on changing any of these properties.

Note*: Densify API elements, parameters, and filters are case-sensitive.*

Logging

You can view the requests and the owner who performed the requests in the web server log file (`%CIRBA_HOME%\logs\cirba-webserver.log`). The log file also includes POST routing request details such as creation and state changes, number of workloads, number of hosting venues, number of Fit-for-Purpose hosting venues, constraint details, sensor details, recommended placement, rejected details, etc. This information is logged independently of configuration setting, API Detailed Logging, below.

To enable detailed logging, specify the configuration setting API Detailed Logging to "ON":

```
rest.api.logging=ON
```

Contact Support@Densify.com for details on changing this property.

Common Elements

To be consistent across the Densify API, there are some components that are common across all the resources.

ID, Name and Self Reference (id, name, href)

Every instance of an individual persistent resource is referable via its own URI. Thus each resource instance has a unique ID with a corresponding self link element that any integration code can store to refer back to that element.

In general, each resource object and any reference to a resource object use the following three elements to identify the object:

- `id`—UUID of the resource
- `name`—name of the resource (if applicable)
- `href`—link to the resource

Every collection request returns the above elements of each object in the collection.

Note*: Densify API elements, parameters, and filters are case-sensitive.*

Date-Time

Date-time elements are specified in Coordinated Universal Time (UTC) format, using the UNIX epoch time in milliseconds.

Common Operations

Summary

When performing a `GET` request, you can qualify the request using one or more operations described in this section, e.g. specifying a sort order or a filter.

This following table summarizes which operations can be combined with other operations. For example, the Filters operation can be combined with Paging, Collection Details and Sort By, as shown by the Filter row. The Filter-Metadata operation can be combined only with the Filters operation, as shown by the Filter-Metadata row.

For example, if you want the detailed list of all AWS Systems, listed in alphabetical order by name, you would do the following:

Request:

```
GET /systems/?platform=AWS&sort_by=name
```

or without trailing '/':

```
GET /systems?platform=AWS&sort_by=name
```

To retrieve the filtering metadata for only Workloads that are in the `UNROUTED` state, do the following:

Request:

```
GET /workload-filter-metadata/?status=UNROUTED
```

Table: Operations Summary

Operation	Filters	Filter-Metadata	Paging	Collection Details	Sort By
Filters		✓	✓	✓	✓
Filter-Metadata	✓				
Paging	✓			✓	✓
Collection Details	✓		✓		✓
Sort By	✓		✓	✓	

Filters

Most collections can be filtered based on the elements of the resource objects within the collection. The elements that support filtering are identified in the **Filter** column (marked "F") of each Resource Elements table within this reference. If the filter finds no matching objects, an empty list is returned. If matching objects are found, a list of objects (in the form ID, Name and Self Reference (id, name, href) on page 29 for each object) is returned. Note that for date/time elements, only a range is supported as described below. To define an exact match for a date/time element, specify the exact match value using the `_from` and `_to` tags.

To filter the list of resource objects, simply provide a query string after the resource object of interest, where both *`<element>`* and *`<value>`* are case insensitive (with no quotes specified):

```
GET /<resource object>/?<element>=<value>
```

For example, if you wanted to filter the list of Systems that are of type `host` you would do the following:

Example: Filtering a List for Host Systems

Request:

```
GET /systems/?type=host
```

Response:

```
[
  {
    "id": "844608bd-2e6a-4201-a9f2-edb4b84389bc",
    "name": "esx-host-82",
    "href": "/systems/844608bd-2e6a-4201-a9f2-edb4b84389bc",
    "resource_id": "ac-59d7-339",
    "type": "host",
    "platform_model": "PowerEdge M610",
    ...
```

To filter on an object link (e.g. finding all Workloads with a specific Infrastructure Group), use the ID of the object in your `GET` request. For this example:

```
GET /workloads/?infrastructure_group_id=8dd39907-765e-40dc-9867-03890b4bbab5
```

More complex filtering is supported using any combination of the following:

- **Multiple Criterion**—specifying "`&`" to logically AND criteria.
  ```
  /<resource object>/?<element1>=<value1>[&<element2>=<value2>]…
  ```
- **Multiple Values**—specifying "`,`" between values to logically OR a list of possible values for an element.
  ```
  /<resource object>/?<element>=<value1>[,<value2>]…
  ```
- **Ranges**—specifying "`_from`" and "`_to`" appended to the end of the element name to specify from/to values, respectively. All elements of type number (including date/time elements) support a range. The values must be valid values (e.g. valid UTC values for date/time types); otherwise, a wrong format error is returned. For example, to filter Control Environments refreshed between July 15 and July 16:
  ```
  /control-environments/?last_refresh_time_from=1468717200000&
     last_refresh_time_to=1468803600000
  ```
- **Multi-Valued Elements**—specifying at least one value of the multi-valued element.
- **Special Character Support**—for filter values with characters that have special meaning in the query string, use the following URL encoding characters instead:
  - "%2B" for "+"
  - "%20" or "+" for space

  For example, to filter cluster name "eastus+test" and platform category "External Cloud", use:

```
GET /systems/?infrastucture_group=eastus%2Btest&platform_
category=External+Cloud
```

- **Name_Like**—use `?name_like="<substring>"` in your collection request. The '%' character can be used to match zero or more characters.

***Note**: Densify API elements, parameters, and filters are case-sensitive.*

Filter-Metadata

Workloads, Infrastructure Groups and Bookings support the `filter-metadata` request, appended to the `GET` collection as follows:

```
GET /<resource object singular form>-filter-metadata
```

This request is used to summarize the collection by returning the elements used for filtering and their values, as defined by the collection. The element names are the same as the element names returned by the collection and individual `GET` requests. This information can then be applied when filtering the collection.

The `filter-metadata` request can also be applied against a filtered collection. Only the metadata of that filtered collection is returned. Any element that can be used to filter the collection can be used with the `filter-metadata` request. See Common Operations: Summary on page 30 for details.

The metadata returned depends on the element type, as follows:

- **numeric**−for numeric values, the minimum and maximum values defined by the collection are returned (e.g. the minimum and maximum of all `expected_date` values in the collection). If there are no values, then a min and max of 0 are returned, or min and max of null in the case of date/time values
- **string**−for string values, the array of values defined by the collection is returned (e.g. the list of all os types in the collection). If there are no values, then the empty list is returned. Note that all unique string values are returned, even if numerically speaking they are the same (e.g. both 81920.0 and 81920 are returned if in the collection and not just 81920).
- **id, name, href**−for associations, the array of associated objects identified by ID, Name and Self Reference (id, name, href) defined by the collection is returned.

There is one element called `status_value` of the Workload resource object in which all possible `status` values are returned and not just the values defined by the collection. This element is identified in the **Filter** column of the Workload Resource Element table.

For example, to obtain the filter metadata by Workload collection, the following `GET` command is issued:

Example: Filtering Metadata by Workload Collection

Request:

```
GET /workload-filter-metadata
```

Response:

```
{
  "os": [
    "Linux",
    "Windows"
  ],
  "vcpu": {
    "min": 1,
    "max": 2
  },
  "memory": {
    "min": 2048,
    "max": 4096
  },
  "expected_date": {
    "min": 1377230400000,
    "max": 13798872000000
  },
  "creation_time": {
    "min": 1377110310553,
    "max": 1377806208400
  },
  "infrastructure_group": [
    {
      "id": "9ed53c2b-3db9-48a3-bdd2-f77952d7e092",
      "name": "Cluster_1",
      "href": "/infrastructure-groups/9ed53c2b-3db9-48a3-bdd2-f77952d7e092"
    },
    {
      "id": "a4fd9693-e2c3-421c-a2dc-2923861cbdb8",
      "name": "Cluster_2",
      "href": "/infrastructure-groups/a4fd9693-e2c3-421c-a2dc-2923861cbdb8"
    }
  ],
  "control_environment": [
    {
      "id": "16579260-f236-44a1-94e5-55e9e6ef6773",
      "name": "Chicago",
      "href": "/control-environments/16579260-f236-44a1-94e5-55e9e6ef6773"
      "icon": "/control-environments/16579260-f236-44a1-94e5-55e9e6ef6773/icon"
    },
    {
      "id": "9cf18385-23d6-4d8a-b3b2-2edba62d314c",
      "name": "Detroit",
      "href": "/control-environments/9cf18385-23d6-4d8a-b3b2-2edba62d314c"
      "icon": "/control-environments/9cf18385-23d6-4d8a-b3b2-2edba62d314c/icon"
    }
  ],
  "status_value": [
    "ANALYZING",
    "PLACED",
    "BOOKED",
    "REJECTED",
    "UNROUTED"
  ],
  "catalog_spec": [
    "lin-medium-4gb",
    "lin-small-2gb"
  ],
  "workload_profile": [
    "Low_Utilization",
    "Medium_Utilization"
  ],
  "name": [
    "vm_1",
    "vm_2",
    // ... *SNIP* ...
  ],
  "host": [
    "esxcrb05.int.Densify.com"
  ],
  "project": [
    "__Unknown__",
    "Kilimanjaro_2013"
  ],
  "owner": [
    "admin",
    "Ter"
  ],
  "cpu_entitlement": {
    "min": 1,
    "max": 12
  },
  "number_of_disks": {
    "min": 2,
    "max": 3
  },
  "provisioned_space": {
    "min": 81920,
    "max": 81920
  },
  "used_space": {
    "min": 20480,
    "max": 20480
  },
  "owner_email": [
    ""
  ],
  "attributes": [
    {
      "id": "attr_2",
      "values": [
        "Financial Services",
        "Engineering"
      ]
    },
    {
      "id": "attr_Workload_Profile",
      "values": [
        "Medium_Utilization"
      ]
    },
    // ... *SNIP* ...
  ]
}
```

Paging

When performing a `GET` request on a collection of objects, a page of objects can be returned instead of the entire collection. To retrieve a specific page, supply the page number (integer >= 0, where 0 is the first page) and the page size, as follows:

```
GET /<resource object>/?page=<pagenumber>[&page_size=<pagesize>]
```

If the page size is not specified, then the page size will be taken from configuration setting API Page Size (parameter key `rest.api.paging.pageSize`).

For example, suppose you have 125 Systems and you want to see the last 25 (with a page size of 100):

```
GET /systems/?page=1
```

***Note**: If the collection changes in between page requests, the items are shifted appropriately.*

Collection Details

Most collection `GET` requests support a `details` option, which returns the details of every resource object in the collection:

```
GET /<resource object>/?details=true
```

The default is `?details=false`, if not specified. This option is supported for most collections. See Object Reference for the list of collections.

Individual Details

Every individual `GET` request uses an `id` or `name` to identify the resource object instance to retrieve (where `name` is used only if the resource object does not have an `id`). The individual `GET` request returns all the elements of the identified or named resource object.

```
GET /<resource object>/<id>
GET /<resource object>/<name>
```

When `name` is used, the search is case insensitive and returns all matching instances.

Sort By

Some collection requests support a `sort_by` option. This option returns the collection of objects in ascending or descending order by the element specified:

```
GET /<resource object>/?sort_by=<element>[,asc|desc]
```

The elements that support sorting are identified in the **Sort By** column (marked "S") of each Resource Elements table within this reference. The objects are returned in ascending order by default (or if `asc` is specified) or descending order if `desc` is specified. If the collection is sorted with objects having "`__Unknown__`" values, these objects are listed at the end independent of the sort order applied to such an element. The sorting is alphabetical or numerical, depending if the `<element>` type is a string or a number, respectively. If the `sort_by` is not specified, a default sort order is provided and is documented in the Supported Operations section of each resource object.

If the `<element>` is not a valid element for the resource object (note that `<element>` is case insensitive), then an error is returned.

For example, to sort all Systems by `name` in ascending order, specify:

```
GET /systems/?sort_by=name
```

To sort all Systems by `size` in descending order, specify the following:

```
GET /systems/?sort_by=size,desc
```

Note*: Densify API elements, parameters, and filters are case-sensitive.*

Other Operations (Create/Modify)

The **Create/Modify-(Req)** column of each Resource Elements table within this reference is used to identify the elements that can be specified when the resource object is created or modified. The elements for create are marked "C" and the elements for modify are marked "M". Elements that must be specified for create are marked "-R" for required.

Error Responses

All error responses (logical or caught exceptions) are in JSON format as:

```
{
  "message" : "error details",
  "status" : errorCode
}
```

Handled exceptions do not expose stack traces, to avoid security threats. Detailed stack traces are logged to the server logs for diagnostics.

Properties Returned

Table: Error Responses

Element	Type	Description
message	string	Detailed message of the exception.
status	errorCode	The HTTP response status code (e.g. 500, 403).

Examples

Example: Caught Exceptions

The response message when a UUID is expected:

Example: Caught Exception

Request:

```
GET /systems/badparm
```

Response:

```
{
  "status" : 404,
  "message" : "UUID is malformed"
}
```

Postman Collection

Densify provides a Postman collection of sample API requests for public cloud.

To learn more watch the video: Using the Postman Collection.

You can download the latest version of Postman from https://postman.com/downloads/.

Download the latest Densify Supply and Demand Postman collection (v14.8) from https://www.densify.com/docs-api/WebHelp_Densify_API/Content/API_Guide/Postman_Collection.htm.

Download the latest Densify Public Cloud Postman collection from https://www.densify.com/docs-api/WebHelp_Densify_API/Content/API_Guide/Postman_Collection.htm.

Follow the steps below to use the downloaded Postman collection:

1. Unzip the downloaded file and import both the **Densify API Collection** collection and **Densify Environment** variables into your Postman workspace.
2. Modify the variables in the **Densify Environment** to match your Densify settings and credentials.

   Note: *If you already have a **Densify** environment in your Postman application, you can either delete the previous version or rename it. Otherwise, you will have duplicate Densify environments after the new collection is imported.*

3. Review the **Documentation** section of the collection for an overview of the workflow and API

requests.

4. Use this sample collection to familiarize yourself with Densify API requests.

Troubleshooting the Postman Collection

Token Requests

When using token-based authentication you need to request a token for each session. Subsequent API requests within the session are automatically authenticated with this token. Then in a future session you need to again request and acquire a token.

Postman is only intended to provide the tools for learning and testing Densify's APIs.

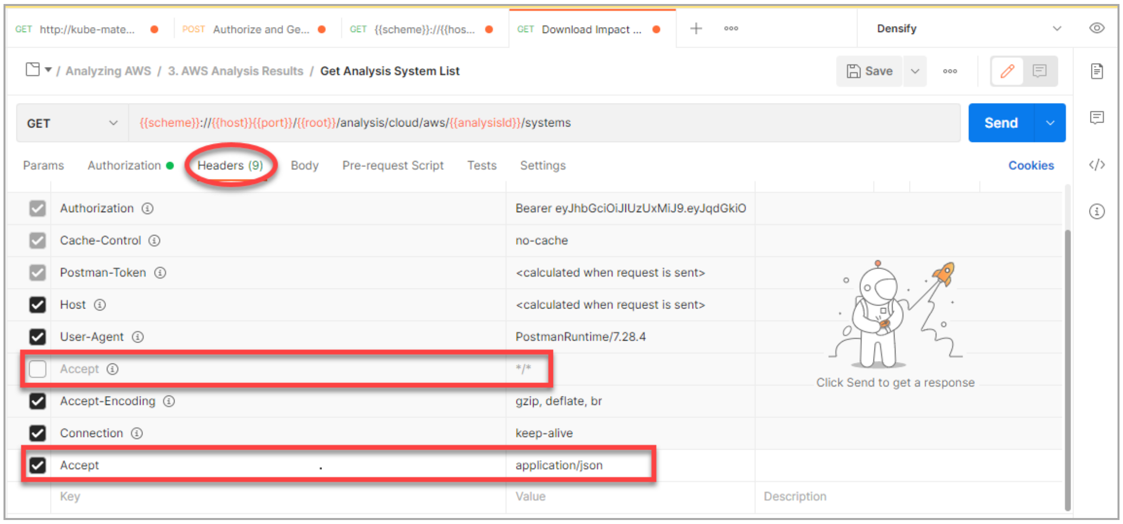
Once you are comfortable with how the Densify API works you can start to place the API calls within your code, that can then be used to run a series of API calls. For example you could write a call to get all analysis IDs, then loop through each ID making another API call to get the associated recommendations, thus providing the full list of recommendations for the scope of systems.

"Bad request, invalid accept header" error

If you use Postman v9.x or later you may encounter a "Bad request, invalid accept header" error. This is due to the fact that the Densify API does not currently support the **Accept** key value "*/" for **Headers**.

Set the **Accept** key value to "application/json".

Figure: Postman Header Configuration



GET {{scheme}}://{{host}}{{port}}/{{root}}/analysis/cloud/aws/{{analysisId}}/systems

	Key	Value	Description
☑	Authorization	Bearer eyJhbGciOiJIUzUxMiJ9.eyJqdGkiO	
☑	Cache-Control	no-cache	
☑	Postman-Token	<calculated when request is sent>	
☑	Host	<calculated when request is sent>	
☑	User-Agent	PostmanRuntime/7.28.4	
□	Accept	*/*	
☑	Accept-Encoding	gzip, deflate, br	
☑	Connection	keep-alive	
☑	Accept	application/json	

Use Cases

Routing Workloads for Immediate Placement

This use case determines the optimal host and datastore placement of one or more Workloads for immediate deployment in the cloud infrastructure.

A common scenario for this use case is initiated when a user makes a request to a cloud provisioning portal to provision one or more virtual machines. The cloud provisioning system, in turn, queries Densify for the best placements for the incoming Workload demand. Based on the results of the query to Densify, the cloud provisioning system is able to start the guests on the appropriate hosts in the cloud infrastructure.

To perform this process through the Densify API, follow the sections below:

1. Describing Workloads to Place
2. Submitting Routing Request
3. Polling for Routing Results
4. Placing the VMs

Describing Workloads to Place

Use the `/workloads` resource to create one or more Workload definitions which describe the capacity requirements for new instances that will be started. The descriptions include the name, a reference system that acts like a clone to copy, resource allocations, and attributes for each Workload to be placed.

The following example request creates 2 workloads named vm1 and vm2. They both resemble the Reference System of a `"lin-small-2gb"` machine, having a single disk defined by default. The other elements describe the resource allocations.

Example: Creating Two Workloads

Request:

```
POST /workloads
{
  "name": "vm1",
  "catalog_spec": "lin-small-2gb",
  "workload_profile": "Medium_Utilization",
  "project": "Project X",
  "vcpu": 4,
  "memory": 8192,
  "attributes":[
    { "name": "Department", "value": "Financial Services" },
    { "name": "Security Zone", "value": "Level 1" }
  ]
}
```

Request:

```
POST /workloads
{
  "name": "vm2",
  "catalog_spec": "lin-small-2gb",
  "workload_profile": "Medium_Utilization",
  "project": "Project X",
  "vcpu": 6,
  "memory": 8192,
  "attributes":[
    { "name": "Department", "value": "Financial Services" },
    { "name": "Security Zone", "value": "Level 1" }
  ]
}
```

Response for vm2:

```
[
  {
    "id": "d7b7a938-4059-42ed-86cb-da016321de82",
    "name": "vm2",
    "status": "UNROUTED",
    "sensors": [],
    "workload_profile": "Medium_Utilization",
    "booking": {
      "id": "5fdd3189-f3ac-4403-9271-933a0d9963d8",
      "name": "vm2",
      "href": "/bookings/5fdd3189-f3ac-4403-9271-933a0d9963d8"
    },
    "project": "Project X",
    "owner": "admin",
    "attributes": [
      {
        "id": "attr_2",
        "name": "Department",
        "value": "Financial Services"
      },
      {
        "id": "attr_IPAddressesAssigned",
        "name": "IP Addresses Assigned",
        "value": "1"
      },
      {
        "id": "attr_SecurityZone",
        "name": "Security Zone",
        "value": "Level 1"
      },
      {
        "id": "attr_Workload_Profile",
        "name": "Workload_Profile",
        "value": "Medium_Utilization"
      },
      {
        "id": "in_maint_mode",
        "name": "In Maintenance Mode",
        "value": "N/A"
      },
      {
        "id": "state_power",
        "name": "Power State",
        "value": "N/A"
      },
      {
        "id": "vmotion_enabled",
        "name": "VMotion Enabled",
        "value": "N/A"
      }
    ],
    "vcpu": 6,
    "memory": 8192,
    "os": "Linux",
    "description": "",
    "bookingHostName": "c01e70d1-c098-4af7-8f8a-b18310bf2351",
    "disks": [
      {
        "name": "SYSTEM",
        "attributes": [],
        "provisioned_space": 81920,
        "used_space": 20480
      }
    ],
    "href": "/workloads/d7b7a938-4059-42ed-86cb-da016321de82",
    "expected_date": 1377230400000,
    "creation_time": 1377291933320,
    "catalog_spec": "lin-small-2gb",
    "catalog_spec_id": "53a234b0-90d2-44ee-ab54-bc06934b2c27",
    "cpu_entitlement": 6,
    "profile_strength": 80,
    "owner_email": "",
    "late_days": 14,
    "provisioned_space": 81920,
    "used_space": 20480
  }
]
```

If the request to create the Workload is successful, the response includes a Workload object with an ID, Name and Self Reference (id, name, href) that can be used to build a subsequent request to route the Workloads. For more information about the format of the Workload response, see Workloads on page 536.

Submitting Routing Request

Use the `/routing-requests` resource to submit a request to route the Workloads to the optimal environment, hosting venue, hosts and datastores according to resource requirements, current utilization patterns, business constraints, and other policy considerations.

Example: Submitting a Routing Request

Request:

```
POST /routing-requests
{
  "workloads": [
    {"id": "4134bf9a-4b2e-45dc-9dc6-23b0bca95315"},
    {"id": "d7b7a938-4059-42ed-86cb-da016321de82"}
  ]
}
```

Response:

```
{
  "id": "da5e1438-17ea-4a88-8c50-ef88eb3409ca",
  "status": "ANALYZING",
  "expected_date": "1381377600000",
  "requester": "admin",
  "workloads": [
    {
      "id": "4134bf9a-4b2e-45dc-9dc6-23b0bca95315",
      "name": "vm1",
      "href": "/workloads/4134bf9a-4b2e-45dc-9dc6-23b0bca95315"
    },
    {
      "id": "d7b7a938-4059-42ed-86cb-da016321de82",
      "name": "vm2",
      "href": "/workloads/d7b7a938-4059-42ed-86cb-da016321de82"
    }
  ],
  "href": "/routing-requests/da5e1438-17ea-4a88-8c50-ef88eb3409ca"
}
```

The Routing Request is evaluated against all available hosting venues and datastores according to cost, capacity and Fit-for-Purpose validation, as described later in Routing Requests–Available Capacity Query on page 361. The Workloads are routed to a hosting venue where the aggregate utilization of all Workloads fit and the policy constraints are satisfied. The Workloads may be routed and placed on different hosts and datastores within the hosting venue.

The Routing Request can optionally include a subset of hosting venues to consider. This limits the scope of the routing algorithm to only the specified hosting venues. See the Routing Requests: Examples on page 357 on how to define the `scopes` element.

Once the Routing Request has been created, the Routing Request object contains an ID, Name and Self Reference (id, name, href) that can be used to query for status. For more information about the format of the Routing Request response, see Routing Requests on page 347.

Polling for Routing Results

Creating the Routing Request is asynchronous, allowing API callers to handle lost connections and subsequently reconnect to obtain the results of the Routing Request.

When the Routing Request for immediate placement is fully processed, the state of the Routing Request is set from `ANALYZING` to one of:

- `PLACED`—Densify has routed, reserved space, and found a host/sensor placement for all the Workloads specified in the Routing Request. All Workloads are updated to `PLACED`.
- `REJECTED`—Densify could not route, reserve space, or find a host/sensor for at least one Workload specified in the Routing Request.

Placing the VMs

If the Workloads were successfully routed and optimal host/datastore placement determined, the caller can query each Workload for its routing details, including the target environment, hosting venue, and host/sensor placement. The Workload (guest) can then be started on the specified host.

Routing Workloads to Reserve Capacity for a Future Date

This use case determines where best to route a given set of Workloads at a future date. When the call is successful, capacity is reserved in the target environment and hosting venue for the future Workloads.

A common scenario for this use case is initiated when a user makes a request to a cloud portal to reserve capacity for one or more virtual machines for a future date. The cloud portal, in turn, queries Densify for the best environment/hosting venue for the incoming Workload demand. Based on the results of the query, the cloud portal reserves capacity for the guests in the appropriate cloud infrastructure. The Routing Request is evaluated against all available hosting venues and the Workloads are routed to a hosting venue where the aggregate utilization of all Workloads fit and the policy constraints are satisfied. The Workloads may be routed and placed on different hosts within the hosting venue.

To perform this process through the Densify API, follow the sections below:

1. Describing Workloads to Place
2. Submitting Routing Request for Future Reservations
3. Polling for Routing Results
4. Placing the VMs

Step 1 is identical to the Routing Workloads for Immediate Placement on page 39 process described above. Steps 2 and 3 require small modifications to accommodate future reservations. Step 4 is similar to the Routing Workloads for Immediate Placement on page 39 process; however, you can place VMs one at a time as the `expected_date` arrives.

Submitting Routing Request for Future Reservations

To reserve capacity for a future date, the Routing Request needs to include a future expected date. The following example shows how the Routing Request would be structured for March 20, 2016:

Example: Submitting Routing Request for Future Reservations

Request:

```
POST /routing-request
{
  "expected_date": 1458486159000,
  "workloads": [
    {"id": "140d5327-cec4-4f86-a5c1-bb59b9d0d639",
     "id": "54e01c38-c87d-4fcb-a8b1-380d5ba46f64"}
  ]
}
```

Once the Routing Request is created, a Routing Request object is immediately returned with an ID, Name and Self Reference (id, name, href) that can be used to query for status. For more information about the format of the Routing Request response, see Routing Requests on page 347.

Polling for Routing Results

When the Routing Request for future reservations is fully processed, the state of the Routing Request object is set to one of:

- `BOOKED`—Densify has routed and reserved space for all the Workloads specified in the Routing Request.
- `REJECTED`—Densify could not route or reserve space for at least one Workload in the Routing Request.

On the day that the Routing Request is to be fulfilled, the status of the Routing Request changes to `PLACED` (or `REJECTED` if no matching workload comes online or if it fails to find a qualified host). A Routing Request is `PLACED` when all its Workloads are also `PLACED`. The Workloads are updated with the recommended hosts where the incoming workloads should be placed.

Placing the VMs

When you are ready to create a guest in your environment, you can perform a `GET` on Workload objects in `BOOKED` status with `expected_date` of today. Determine the specific Workload you would like to first place and perform a `GET` on that object. This will determine the recommended `host/sensor` placement and will update the Workload to `PLACED`. Start the Workload (guest) on the specified host.

Note: *Once this* `host/sensors` *element is defined, it is not updated on subsequent* `GET` *individual requests.*

Querying Available Capacity for Specified Workloads

This use case assesses the available capacity of one or more hosting venues, given specific workload requirements. For example, an environment may have space for a different number of small, medium, or large Workloads, and for Workloads that meet specific business constraints.

When assessing multiple hosting venues, the available capacity of each hosting venue is assessed according to cost, capacity and Fit-for-Purpose, as described in Routing Requests–Available Capacity Query on page 361. This operation does not place Workloads or reserve capacity.

To perform this query through the Densify API, follow the sections below:

1. Describing Workloads to Assess
2. Querying Available Capacity
3. Processing Capacity Query Results

Describing Workloads to Assess

Use the `/workloads` resource in the same way as the Routing Workloads for Immediate Placement on page 39 or Routing Workloads to Reserve Capacity for a Future Date on page 43 use cases to create one or more Workload definitions that specify the name, the reference system that acts like a clone, resource allocations and attributes for each Workload being assessed.

If the Workload requests are successful, the responses include ID, Name and Self Reference (id, name, href) to the newly created Workload objects. The Workloads can then be referenced in subsequent API calls such as querying for available capacity for the Workloads.

Querying Available Capacity

Use the /routing-requests/available-capacity-query resource to submit a request to determine the available capacity for the specified Workloads in the hosting venues in the request scope. This resource is used in the same way as the /routing-requests resource itself.

For the example below, as no environments and hosting venues are specified, the available capacity for all environments and their corresponding hosting venues are returned.

Example: Querying Available Capacity

Request:

```
POST /routing-requests/available-capacity-query/
{
  "expected_date": 1426545823000,
  "workloads": [
    {"id": "140d5327-cec4-4f86-a5c1-bb59b9d0d639"},
    {"id": "54e01c38-c87d-4fcb-a8b1-380d5ba46f64"}
  ]
}
```

Optionally, the available capacity query can include specifications to narrow the scope of environments and/or hosting venues to consider.

Example: Querying Available Capacity with Defined Scope

Request:

```
POST /routing-requests/available-capacity-query/
{
  "scopes": [
    {
      "control_environment": "Boston"
    }
  ],
  "expected_date": 1426545823000,
  "workloads": [
    {"id": "140d5327-cec4-4f86-a5c1-bb59b9d0d639"},
    {"id": "54e01c38-c87d-4fcb-a8b1-380d5ba46f64"}
  ]
}
```

Processing Capacity Query Results

For each hosting venue assessed by the query, the available capacity is expressed in terms of the number of available slots for the specified Workloads, the number of available slots for the sensor requirements, the hosting cost, the Fit-for-Purpose tests, and the overall hosting score, as described in Routing Requests–Available Capacity Query on page 361.

The number of available slots is estimated by computing the number of instances of the aggregate Workloads that can fit in the hosting venue. Thus, the higher the number of available slots in a hosting venue, the greater the amount of available capacity for the specified group of Workloads.

An example set of the available capacity query results is listed below:

Example: Processing Capacity Query Results

Response:

```
[
  {
    "infrastructure_groups": [
      {
        "id": "f7a5ef21-3d0a-442e-b65c-541b0084b950",
        "name": "Cluster1",
        "slots": 20,
        "cei": 0.88,
        "subslots": [],
        "constraint": "Total_Memory",
        "constraint_name": "Total Memory",
        "href": "/infrastructure-groups/f7a5ef21-3d0a-442e-b65c-541b0084b950",
        "control_type": "FULL",
        "hosting_cost": 38,
        "fit_for_purpose": {
          "categories": [
            {
              "name": "Resource",
              "test": [
                {
                  "name": "Datastore Tier",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Operating Systems",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Guest Sizes",
                  "status_reasons": [],
                  "status": "PASS"
                }
              ],
              "status": "PASS"
            },
            {
              "name": "Security",
              "test": [
                {
                  "name": "Data Jurisdiction",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Compliance",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Encryption and Key Management",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Identity and Access Management",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Security Zones",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Network Isolation",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Intel TXT Support",
                  "status_reasons": [],
                  "status": "PASS"
                }
              ],
              "status": "PASS"
            },
            {
              "name": "Business",
              "test": [
                {
                  "name": "Location",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Departments",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Service Level",
                  "status_reasons": [],
                  "status": "PASS"
                }
              ],
              "status": "PASS"
            },
            {
              "name": "Technical",
              "test": [
                {
                  "name": "Hosting Platform",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Hypervisor",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Maximum Guest I/O",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Operational Environments",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "DPDK",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Software Licenses",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Crypto H/W Acceleration",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "License Groups",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Transport Zone",
                  "status_reasons": [],
                  "status": "PASS"
                }
              ],
              "status": "PASS"
            }
          ],
          "status": "PASS"
        },
        "sensor_capacity": [
          {
            "type": "datastore",
            "slots": 56,
            "constraint": "98EA05B5-E0E1-4828-A3DD-919C5738D29A",
            "constraint_name": "Number of VMs",
            "subslots": []
          },
          {
            "type": "phystor",
            "slots": 105,
            "constraint": "509E681F-0C7A-4193-95B0-9523C14E0FED",
            "constraint_name": "Total Used Space (MB)",
            "subslots": []
          },
          {
            "type": "ipaddresspools",
            "slots": 2147483647,
            "constraint": "62d95498-6fdb-4fce-aebe-5f297ca0f7c3",
            "constraint_name": "IP Addresses Assigned",
            "subslots": []
          }
        ],
        "hosting_score": 97
      },
      {
        "id": "6b717553-ddcb-4333-bbc6-8c55e830d54f",
        "name": "Cluster2",
        "slots": 28,
        "cei": 0.78,
        "subslots": [],
        "constraint": "Total_Memory",
        "constraint_name": "Total Memory",
        "href": "/infrastructure-groups/6b717553-ddcb-4333-bbc6-8c55e830d54f",
        "control_type": "FULL",
        "hosting_cost": 0,
        "fit_for_purpose": {
          // ... *SNIP* of categories ...
        },
        "sensor_capacity": [
          {
            "type": "datastore",
            "slots": 76,
            "constraint": "98EA05B5-E0E1-4828-A3DD-919C5738D29A",
            "constraint_name": "Number of VMs",
            "subslots": []
          },
          {
            "type": "phystor",
            "slots": 129,
            "constraint": "509E681F-0C7A-4193-95B0-9523C14E0FED",
            "constraint_name": "Total Used Space (MB)",
            "subslots": []
          },
          {
            "type": "ipaddresspools",
            "slots": 2147483647,
            "constraint": "62d95498-6fdb-4fce-aebe-5f297ca0f7c3",
            "constraint_name": "IP Addresses Assigned",
            "subslots": []
          }
        ],
        "hosting_score": 48
      }
    ],
    "control_environment": {
      "id": "e6cf1672-77ff-4e7d-9dda-5387b0bc95cc",
      "name": "Boston",
      "platform": "VMWARE",
      "platform_category": "Internal Virtual",
      "control_type": "FULL",
      "total_slots": 48,
      "cei": 0.82,
      "href": "/control-environments/e6cf1672-77ff-4e7d-9dda-5387b0bc95cc",
      "subslots": [],
      "sensor_capacity": [
        {
          "type": "datastore",
          "slots": 132,
          "constraint": "98EA05B5-E0E1-4828-A3DD-919C5738D29A",
          "constraint_name": "Number of VMs",
          "subslots": []
        },
        {
          "type": "phystor",
          "slots": 234,
          "constraint": "509E681F-0C7A-4193-95B0-9523C14E0FED",
          "constraint_name": "Total Used Space (MB)",
          "subslots": []
        },
        {
          "type": "ipaddresspools",
          "slots": 2147483647,
          "constraint": "62d95498-6fdb-4fce-aebe-5f297ca0f7c3",
          "constraint_name": "IP Addresses Assigned",
          "subslots": []
        }
      ],
      "icon": "/control-environments/e6cf1672-77ff-4e7d-9dda-5387b0bc95cc/icon"
    }
  }
]
```

Based on the above example, the first Infrastructure Group can accommodate `"slots": 20` of the aggregate demand of the specified Workloads and the second Infrastructure Group can accommodate `"slots": 28`, with a total of `"slots": 48` for the Control Environment.

The best hosting option is the first Infrastructure Group, as it has the highest score of `"hosting_score": 97`.

Adding Supply

When planning new hosts and datastores, the process is much simpler than when planning new workloads.

The steps are similar for hosts and datastores. The following describes how to add your hosts:

1. Describing Inbound Hosts
2. Renaming Inbound Hosts
3. Provisioning Hosts

Describing Inbound Hosts

Use the `/inbound-hosts` resource to create two host definitions which describe the system hardware (including the manufacturer, platform, CPU and memory requirements), the target Infrastructure Group and the planned expected date.

Before doing this, you may need to calculate the CPU benchmark. To do this, use the `/helper-utilities/resolve-cpu-benchmark` resource.

Example: Calculating CPU Benchmark

Request:

```
POST /helper-utilities/resolve-cpu-benchmark
{
  "manufacturer" : "HP",
  "platform_model" : "ProLiant DL380 G7",
  "cpu_model": "Intel Xeon X5690",
  "cpu_speed": 3466,
  "total_physical_cpus": 2,
  "cores_per_cpu": 6,
  "threads_per_core": 2
}
```

Response:

```
Response:
[
  {
    "name": "CINT2006 Rate",
    "score_type": "cint2006rate",
    "value": 404.47
  }
]
```

The following example request creates 2 hosts named ucs-host21 and ucs-host22, with the determined CPU benchmark.

Example: Creating Multiple Inbound Hosts

Request:

```
POST /inbound-hosts
{
  "num_copy":2,
  "name": "ucs-host2",
  "manufacturer": "HP",
  "platform_model": "ProLiant DL380 G7",
  "cpu_model": "Intel Xeon X5690",
  "total_physical_cpus": "2",
  "cores_per_cpu": "6",
  "threads_per_core": "2",
  "cpu_allocation": "12.00",
  "cpu_speed": "3466",
  "memory": "131072",
  "owner": "Melissa",
  "owner_email": "melissa@Densify.com",
  "project": "UCS hardware refresh",
  "attributes": [
    {
      "name": "Security Zone",
      "value": "Level 1"
    }
  ],
  "expected_date": 1472659206000,
  "cpu_benchmarks": [
    {
      "name": "CINT2006 Rate",
      "score_type": "cint2006rate",
      "value": 404.47
    }
  ],
  "I/O_benchmarks": [
    {
      "name": "Maximum Network Throughput (bytes/s)",
      "score_type": "net",
      "value": 1342177280
    },
    {
      "name": "Maximum Disk Throughput (bytes/s)",
      "score_type": "disk",
      "value": 1342177280
    }
  ],
  "infrastructure_group": {
    "id": "2baa1ac4-4ffe-4e79-a73d-d038145bca8d"
  }
}
```

Response:

```
  [
    {
      "id": "0d69655a-d10b-4d76-88e2-5bc3916c8e07",
      "name": "ucs-host21",
      "href": "/inbound-hosts/0d69655a-d10b-4d76-88e2-5bc3916c8e07",
      "manufacturer": "HP",
      "platform_model": "ProLiant DL380 G7",
      "cpu_model": "Intel Xeon X5690",
      "total_logical_cpus": 24,
      "total_physical_cpus": 2,
      "cores_per_cpu": 6,
      "threads_per_core": 2,
      "cpu_allocation": "12.0",
      "cpu_speed": 3466,
      "memory": 131072,
      "status": "PENDING",
      "owner": "Melissa",
      "project": "UCS hardware refresh",
      "description": "",
      "expected_date": 1472659206000,
      "creation_time": 1470844834427,
      "owner_email": "melissa@Densify.com",
      "attributes": [
        {
          "id": "attr_SecurityZone",
          "name": "Security Zone",
          "value": "Level 1"
        }
      ],
      "cpu_benchmarks": [
        {
          "name": "CINT2006 Rate",
          "score_type": "cint2006rate",
          "value": 404.47
        }
      ],
      "I/O_benchmarks": [
        {
          "name": "Maximum Disk Throughput (bytes/s)",
          "score_type": "disk",
          "value": 1342177280
        },
        {
          "name": "Maximum Network Throughput (bytes/s)",
          "score_type": "net",
          "value": 1342177280
        }
      ],
      "infrastructure_group": {
        "id": "2baa1ac4-4ffe-4e79-a73d-d038145bca8d",
        "name": "Prod2_BDVC-01",
        "href": "/infrastructure-groups/2baa1ac4-4ffe-4e79-a73d-d038145bca8d"
      },
      "control_environment": {
        "id": "26092815-9d17-4e6a-abbd-f5b05a853bff",
        "name": "Boston",
        "href": "/control-environments/26092815-9d17-4e6a-abbd-f5b05a853bff",
        "icon": "/control-environments/26092815-9d17-4e6a-abbd-
  f5b05a853bff/icon"
      }
    },
    {
      // ... *SNIP* of Inbound Host "ucs-host22 ...
    }
  ]
```

If the request to create the Inbound Hosts is successful, the response includes two Inbound Host objects with an ID, Name and Self Reference (id, name, href). For more information about the format of the Inbound Host response, see Inbound Hosts on page 280.

Renaming Inbound Hosts

The following example updates the Inbound Host name from "ucs-host21" to "ucs-host1".

Example: Modifying an Inbound Host Name

Request:

```
PUT /inbound-hosts/d274c5ed-b53e-4020-9e42-bc072c6d4816
{
  "name": "ucs-host1"
}
```

Provisioning Hosts

The host can then be provisioned, by performing the following request to obtain the details. The example returns all the Inbound Hosts scheduled to be provisioned up to the end of September. The response of Inbound Hosts can be given to the person who will provision the supply.

Example: Getting a Collection of Inbound Hosts

Request:

```
GET /inbound-hosts/?expected_date_to=1475280000000
```

Optimizing AWS EC2 Instances

This use case includes API-controlled data collection, analysis, and generation of right-sizing recommendations for EC2 instances within an AWS account. The recommendations, which can be downloaded on demand or posted to a webhook-specified receiver, provide extensive details that can be used to feed downstream automation engines in either JSON or terraform-map formats.

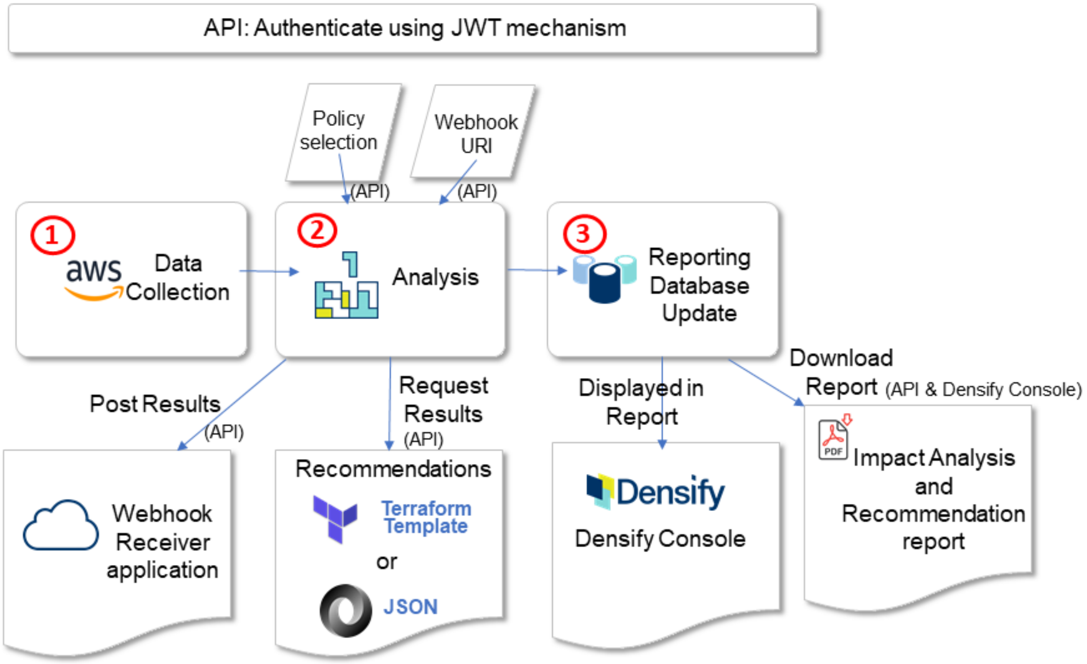
After the reporting tables have been updated with the analysis results, you can view the recommendation reports from the Densify Console. In addition, a comprehensive PDF report can be downloaded for each instance to supply supporting evidence for a recommendation; this can be reviewed by a system owner before approvals are given for the suggested changes.

***Note**: The AWS EC2 optimization opportunities dashboard in the Densify Console contains recommendations for all AWS accounts for which you have collected EC2 instance data (see EC2 in the topic Optimizing Your Public Cloud (Help Topic ID 380320)). Each* `/analysis/cloud/aws/<analysisId>/results` *API request returns recommendations for one AWS account, since each analysis entity is defined by the AWS account connection. Therefore if you have more than one AWS account, an API request against a single account represents a subset of the full set of EC2 recommendations displayed in the Densify Console.*

To learn more watch the video: Understanding the Densify API Workflow

The diagram below provides an overview of the three major processes in Densify:

Figure: Optimizing AWS EC2 Processes



1. **Data Collection**—Collect AWS data anbd load into Densify.

   Use the Densify API or the Densify Console to initiate this process.

2. **Analysis**—Analyze collected data based on pre-defined policies to make recommendations.

   The analysis process is initiated automatically after data collection completes. Results can be immediately pulled from the Densify API or pushed to a webhook URI once the analysis is complete.

3. **Reporting Database Update**—Update reporting database tables with analysis results for reporting.

   The reporting database update process is scheduled to run nightly after the analysis process. The Densify Console optimization opportunity reports and the Impact Analysis and Recommendation reports are available after the reporting database update is complete.

***Note**: Before you can collect data via AWS CloudWatch API, you need to create and configure an IAM role for Densify to have a trust relationship with your AWS account. See AWS Data Collection Prerequisites for an IAM Role (Help Topic ID 410060) for details.*

The steps below outline a typical API workflow to optimize AWS EC2 instances:

1. Authenticate using JWT—Retrieve token to authenticate each subsequent API call.
2. Optional: Review Policy—Review a list of available policies to be used for analysis.

3. Collect Data and Analyze—Start collecting data and initiate analysis (specify optional policy, optional webhook URI).
4. Check for Analysis Status—Check for data collection and analysis status or wait for a webhook trigger to indicate that the analysis finished and downstream processing has initiated.
5. Download EC2 Recommendations for Action—Retrieve recommendations for actioning or forward to orchestration engine.
6. Optional: Download Impact Analysis and Recommendation Instance Report for system owners.

Postman Collection

Examples of Densify API request for this use case can be found in the following Postman collection:

Download the latest Densify Supply and Demand Postman collection (v14.8) from https://www.densify.com/docs-api/WebHelp_Densify_API/Content/API_Guide/Postman_Collection.htm.

Download the latest Densify Public Cloud Postman collection from https://www.densify.com/docs-api/WebHelp_Densify_API/Content/API_Guide/Postman_Collection.htm.

Follow the steps below to use the downloaded Postman collection:

1. Unzip the downloaded file and import both the **Densify API Collection** collection and **Densify Environment** variables into your Postman workspace.
2. Modify the variables in the **Densify Environment** to match your Densify settings and credentials.

   Note: *If you already have a **Densify** environment in your Postman application, you can either delete the previous version or rename it. Otherwise, you will have duplicate Densify environments after the new collection is imported.*

3. Review the **Documentation** section of the collection for an overview of the workflow and API requests.
4. Use this sample collection to familiarize yourself with Densify API requests.

Authenticate using JWT

Use the `/authorize` resource to retrieve an authenticated token to make subsequent API calls. See Authorize on page 218 for details. By default, the token expires in 5 minutes, therefore you will need to ensure that you have an active token for each Densify API request.

Example: Retrieving an Authenticated Token

Request:

```
POST /authorize
{
    "userName": "apiUser",
    "pwd": "apiPassword"
}
```

Response:

```
{
  "apiToken": "eyJh-
bGciOiJIUzUxMiJ9.eyJqdGkiOiIzNzI2Yzk0NC0wMmE4LTRlYzQtOGE2Ny04ODBmMDM2OTRhZD-
ciLCJpYXQiOjE1NDI2NTI0MDUsInN1YiI6InZh-
biIsIm-
lzcyI6IkRlb-
nNpZnkuY29tIiwiZXhwIjoxNTQyNjUyNzA1fQ.cJd8qFJfRoPnMEU7GzcdYGBT8WwlgmviQ1OQp8P_
w9VUcjQA3FJaB9QkqJJ6d7zbrY5yjc4w0rOWjY-PPdbmqw",
  "expires" : 1542652705869,
  "status" : 200
}
```

Use the `apiToken` value in your Authorization request header for subsequent Densify API calls.

Review Policy

Use the `/analysis/cloud/aws/policy` resource to view a list of policies available in Densify. You can use one of these policies for optimization analysis. This is an optional step as the analysis runs with the default policy if no policy is specified. See GET /analysis/cloud/aws/policy on page 201 for details.

Example: Listing AWS Cloud Policies with Description

Request:

```
GET /analysis/cloud/aws/policy?details=true
```

Response:

```
{
  "policyId": "4a63f651-a583-4157-97ff-35651370ffbe",
  "policyInstanceId": "0c0ef18b-9367-4071-b733-396f63e51925",
  "name": "DevOps-Automation"
  "description": "This policy is intended for generating instance sizing and
instance family optimization recommendations that require little or no
review before being implemented. Densify utilizes rule-driven analytics to
predict the effort of changing instance type from current to recommended,
and this policy favors Low effort recommendations, producing higher automation
at the expense of lower cost savings.\r\nThe resource utilization of each sys-
tem is modeled using a minimum of 7 days and up to 90 days of historical work-
load.\r\nWhen optimizing instance sizes and families, the predicted CPU and
memory usage must not exceed 65% and 85%, respectively.\r\nThis policy will
not specify burstable (T-series) instance families for workloads unless they
are already running in a burstable family and will not change CPU Architecture
from Intel to AMD or vice versa.\r\nWhen memory usage metrics are not avail-
able, the analysis assumes the existing memory allocation is required and will
not change the memory configuration."
},
{
  "policyId": "4a63f651-a583-4157-97ff-35651370ffbe",
  "policyInstanceId": "69fa4c99-1be2-4048-94a7-36fd83d07f37",
  "name": "DevOps-Default",
  "description": "This policy reflects best practices for generating instance
sizing and instance family optimization recommendations.\r\nThe resource util-
ization of each system is modeled using a minimum of 7 days and up to 60 days
of historical workload.\r\nWhen optimizing instance sizes and families, the
predicted CPU and memory usage levels must not exceed 70% and 90%, respect-
ively.\r\nWhen memory usage metrics are not available, the analysis effect-
ively assumes the existing memory allocation of the instance is required and
will not change the memory configuration."
}
```

Collect Data and Analyze

Initiate data collection and analysis for a given AWS account using the `/analysis/cloud/aws/analyze` resource. See Analysis: AWS Analyze on page 93 for details.

You can specify an override policy to use for your analysis and/or specify a webhook to publish results when analysis is complete.

Example: Running AWS Data Collection and Analysis with Override Policy and Webhook

Request:

```
POST /analysis/cloud/aws/analyze
{
  "accountId": "888665225420",
  "roleArnName": "arn:aws:iam::888665225420:role/ReadOnlyAccess",
  "roleExternalId": "mypassword",
  "policyInstanceId": "f942b1a1-8f24-4032-b29a-0f628375d880",
  "webHook": {
       "uri": "http://mywebhookserver/webhook/results",
       "authType": "basic",
       "authValue": "webuser:wupassword"
  }
}
```

Response:

```
{
  "href": "Not available"
  "message": "Analysis in progress",
  "status": 200
}
```

Check for Analysis Status

To find your AWS analysis entity, request for a list of existing AWS analyses available. Refer to List all analyses for a particular platform and vendor for details.

Example: Listing all AWS Analyses Available

Request:

```
GET /analysis/cloud/aws/
```

Response:

```
{
      "analysisId": "9a5d2d55-6d85-4fde-8bab-fcd0cef8c5bf",
      "analysisName": "624756828528",
      "analysisCompletedOn": 0,
      "href": "/analysis/cloud/aws/9a5d2d55-6d85-4fde-8bab-fcd0cef8c5bf",
      "analysisResults": "/analysis/cloud/aws/9a5d2d55-6d85-4fde-8bab-fcd0ce-
f8c5bf/results",
      "analysisStatus": "/analysis/cloud/aws/9a5d2d55-6d85-4fde-8bab-fcd0ce-
f8c5bf/status"
}
```

Use the `"analysisStatus"` resource element from the analysis entity to check for progress status. See Analysis: Status on page 206 for details on this resource.

The analysis is complete and recommendations are available when `"analysisStage": "Completed"`.

Example: Checking AWS Analysis Status

Request:

```
GET /analysis/cloud/aws/9a5d2d55-6d85-4fde-8bab-fcd0cef8c5bf/status
```

Response:

```
{
  "analysisStage": "Completed",
  "webHookStatus": "Success",
  "message": "Analysis 624756828528 was last completed on Wed Feb 27 10:03:13
EST 2019."
}
```

Download EC2 Recommendations for Action

Use the `/analysis/cloud/aws/<analysisId>/results` resource to retrieve EC2 recommendation results and forward them to a downstream orchestration engine for actioning. See Analysis: AWS Recommendations on page 105 for details.

Example: Returning Low-Effort EC2 Recommendations in Terraform-map

Request:

```
GET /analysis/cloud/aws/9a5d2d55-6d85-4fde-8bab-fcd0ce-
f8c5bf/results?serviceType=EC2&effortEstimate=Low
```

Headers:

```
Accept: application/terraform-map
Authorization: Bearer <apiToken>
```

***Note**: This request returns low-effort EC2 recommendations only for the account defined in the specified analysis (analysisId=9a5d2d55-6d85-4fde-8bab-fcd0cef8c5bf). If you want all EC2 recommendations for your Densify collected AWS infrastructure, you must accumulate the EC2 request results for all AWS analysis entities available in Densify.*

Download Impact Analysis and Recommendation Instance Report

Use the `rptHref` resource element provided in the JSON recommendation output to download a PDF Impact Analysis and Recommendation Report for each instance. This report is available after a reporting database update, which is scheduled to run nightly. See rptHref on page 111 and Example: Downloading an Impact Analysis and Recommendation Report on page 125 for details.

Example: Downloading an Impact Analysis and Recommendation Report

Request:

```
GET /systems/9834335a-1942-4115-a65d-a298be1d390c/analysis-report
```

Headers:

```
Accept: application/octet-stream
Authorization: Bearer <apiToken>
```

Optimizing GCP Compute Engine Instances

This use case includes GCP cloud infrastructure data collection, analysis, and generation of right-sizing recommendations for Compute Engine instances using a combination of Densify Console and Densify API operations. The recommendations, which can be downloaded on demand via Densify API, provide extensive details that can be used to feed downstream automation engines in either JSON or terraform-map formats.

A comprehensive PDF report can be downloaded for each instance to supply supporting evidence for a recommendation; this can be reviewed by a system owner before approvals are given for the suggested changes.

1. **Data Collection**—Collect GCP data into Densify.

   Use the Densify Console to initiate this process.

2. **Analysis**—Analyze collected data based on pre-defined policies to make recommendations.

   The analysis process is initiated automatically after data collection completes. Results can be immediately pulled from the Densify API or pushed to a webhook URI once the analysis is complete.

3. **Reporting Database Update**—Update reporting database tables with analysis results for reporting.

   The reporting database update process is scheduled to run nightly after the analysis process. The Densify Console GCP optimization opportunity report and the Impact Analysis and Recommendation reports are available after the reporting database update is complete.

***Note**: Before you can collect GCP cloud infrastructure data in Densify, you need to create a GCP service account with services enabled and permissions configured. See Google Cloud Platform Data Collection Prerequisites (Help Topic ID 380300) for details.*

The steps below outline a typical workflow to optimize GCP Compute Engine instances:

1. Collect GCP data through the Densify Console using the Cloud Connections wizard. See *Configuring a Google Cloud Platform Connection* from the topic *Using the Public Cloud*

Connection Wizard (Help Topic ID 380290) for details.

After data collection completes, an analysis is initiated for each project associated with the GCP account configured for data collection.

2. From the Densify API: Authenticate using JWT—Retrieve token to authenticate each subsequent API call.
3. API: List All GCP Analyses—Query for the analysis entity of each project for which data was collected (from step 1). You can use the returned analysis entity to check for status and recommendations.
4. API: Poll for Analysis Status—Check for data collection and analysis status. The recommendations are available when analysis is complete.
5. API: Download Results for Action—Retrieve recommendations for actioning or forward to orchestration engine.

   Alternatively, you can Add a Webhook to an Existing GCP Analysis on page 66 to push recommendations to an external URI once the next analysis process completes.
6. API/Console (optional): Download Impact Analysis and Recommendation Instance Report for system owners. This can achieved from the Densify API or from the Densify Console. See *Viewing the Impact Analysis and Recommendation Report* (Help Topic ID 380450) for details on the content of the report.
7. Console (optional alternative): Review GCP Virtual Machine optimization opportunity report from the Densify Console. See *GCP* from the topic *Optimizing Your Public Cloud* (Help Topic ID 380320) for details. The optimization opportunity report is available after the nightly reporting database update.
8. Console (optional): Review GCP optimization policy settings used during analysis from the Densify Console. See *Cloud Optimization Policies* from the topic *Viewing Policy Settings* (Help Topic ID 120200) for details.

Postman Collection

Examples of Densify API request for this use case can be found in the following Postman collection:

Download the latest Densify public cloud Postman collection (v 12.1.7) from https://www.densify.com/docs-api/Content/API_Guide/Postman_Collection.htm.

Follow the steps below to use the downloaded Postman collection:

1. Unzip the downloaded file and import it into your Postman application. The **Densify API Collection (12.1.7)** collection and **Densify** environment are loaded into your Postman workspace.

2. Modify the variables in the **Densify** environment to match your Densify and cloud-specific settings and credentials.

   Note: *If you already have a previous* ***Densify*** *environment in your Postman application, you can either delete the previous version or rename it. Otherwise, you will have duplicate Densify environments after the new collection is imported.*

3. Review the **Documentation** section of the collection for an overview of the workflow and API requests.

Use this sample collection to familiarize yourself with Densify API requests.

Authenticate using JWT

Use the `/authorize` resource to retrieve an authenticated token to make subsequent API calls. See Authorize on page 1 for details. By default, the token expires in 5 minutes, therefore you will need to ensure that you have an active token for each Densify API request.

Example: Retrieving an Authenticated Token

Request:

```
POST /authorize
{
    "userName": "apiUser",
    "pwd": "apiPassword"
}
```

Response:

```
{
  "apiToken": "eyJh-
bGciOiJIUzUxMiJ9.eyJqdGkiOiIzNzI2Yzk0NC0wMmE4LTRlYzQtOGE2Ny04ODBmMDM2OTRhZD-
ciLCJpYXQiOjE1NDI2NTI0MDUsInN1YiI6InZh-
biIsIm-
lzcyI6IkRlb-
nNpZnkuY29tIiwiZXhwIjoxNTQyNjUyNzA1fQ.cJd8qFJfRoPnMEU7GzcdYGBT8WwlgmviQ1OQp8P_
w9VUcjQA3FJaB9QkqJJ6d7zbrY5yjc4w0rOWjY-PPdbmqw",
  "expires" : 1542652705869,
  "status" : 200
}
```

Use the `apiToken` value in your Authorization request header for subsequent Densify API calls.

List All GCP Analyses

To see a list of all GCP analyses currently in Densify, use the `/analysis/cloud/gcp` resource. See List all analyses for a particular platform and vendor on page 1 for details of the resource operation.

Example: List all Saved GCPAnalyses

Request:

```
GET /analysis/cloud/gcp
```

Response:

```
{
   "analysisId": "cf25d8c1-4fdf-42b1-b3a3-1a8d8a425d13",
   "analysisName": "gcpAcctEast-608378",
   "analysisCompletedOn": 0,
   "href": "/analysis/cloud/gcp/cf25d8c1-4fdf-42b1-b3a3-1a8d8a425d13",
   "analysisResults": "/analysis/cloud/gcp/cf25d8c1-4fdf-42b1-b3a3-
1a8d8a425d13/results",
   "analysisStatus": "/analysis/cloud/gcp/cf25d8c1-4fdf-42b1-b3a3-
1a8d8a425d13/status"
}
```

Poll for Analysis Status

Use the `"analysisStatus"` resource element from the List All GCP Analyses response to poll for status of the data collection and analysis. See Check for analysis status on page 1 for details of the resource operation. The analysis is complete and recommendations are available when `"analysisStage": "Completed"`.

Example: Checking GCP Analysis Status

Request:

```
GET /analysis/cloud/gcp/cf25d8c1-4fdf-42b1-b3a3-1a8d8a425d13/status
```

Response:

```
{
  "analysisStage": "Completed",
  "webHookStatus": "",
  "statusMessage": "Analysis gcpAcctEast-608378 was last completed on Wed Feb
27 12:06:10 EST 2019."
}
```

Add a Webhook to an Existing GCP Analysis

You can add a webhook definition to an existing GCP analysis. The analysis will push the optimized results to the specified webhook URI. See *Add a webhook to an analysis* in the topic *Analysis: Webhook* (Help Topic ID 340490) for details on adding a webhook.

Example: Adding a Webhook Definition to an Existing GCP Analysis

Request:

```
POST /webhook/analysis/cloud/gcp/cf25d8c1-4fdf-42b1-b3a3-1a8d8a425d13
{
  "uri": "https://myInstanceConfigServer:443/api/densify/results",
  "authType": "bearer",
  "authValue": "eyJh-
bGciOiMsIUzUxMiJ9.eyJqdGkiOiJhYWU2MjIxOS1iOWQyLTQ3OGMtYWI3Mi00NGU2OTUzY2RjMDEi-
LCJpYXQiOjE1NDM2MDgxMTEsInN1YiI6ImFk-
bWluIiwiaXNzIjoiRGVuc2lmeS5jb20iLCJleHAiOjE1NDM2MDg0MTF9.h3bJrAP-
Z2LeqzjN3FYpFDyoaADvYT1MdLw5SuguqkGE7s-
jB4c7YgQgv3saj15r2IsgTWH8PW7eNnoZwFP9eiQ"
}
```

Response:

```
{
  "message": "ok",
  "status": 200
}
```

Download Results for Action

Use the `/analysis/cloud/gcp/<analysisId>/results` resource to retrieve recommendation results and forward to orchestration engine. See Analysis: GCP Recommendations on page 1 for details.

Example: Returning GCP Recommendations in Terraform-map

Request:

```
GET /analysis/cloud/gcp/cf25d8c1-4fdf-42b1-b3a3-1a8d8a425d13/results
```

Headers:

```
Accept: application/terraform-map
Authorization: Bearer <apiToken>
```

Download Impact Analysis and Recommendation Instance Report

Use the `rptHref` resource element provided in the instance recommendation output to download a PDF Impact Analysis and Recommendation Report for each instance. This report is available after a reporting database update, which is scheduled to run nightly. See rptHref on page 1 and Example: Downloading an Impact Analysis and Recommendation Report on page 1 for details.

Example: Downloading an Impact Analysis and Recommendation Report

Request:

```
GET /systems/b374d9d5-a529-4e19-9a14-077123f322b4/analysis-report
```

Headers:

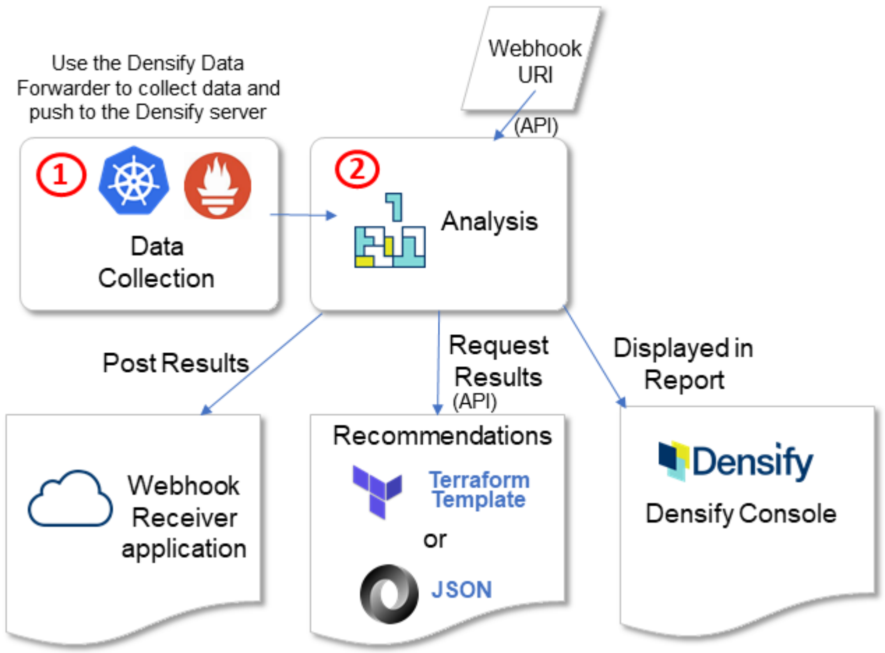
```
Accept: application/octet-stream
Authorization: Bearer <apiToken>
```

Optimizing Kubernetes Containers

This use case provides an overview of how to collect Kubernetes container data into Densify and how to obtain Densify container recommendations.

The diagram below shows the major processes in Densify to optimize Kubernetes containers:

Figure: Optimizing Kubernetes Containers Processes



1. **Data Collection**—Collect Kubernetes data into Densify.

   You need to have an existing monitoring and metrics collection service, such as Prometheus, configured for your container clusters. See *Container Prerequisites* (Help Topic ID 410140) for details.

   Densify provides a collection and transfer tool called the Data Forwarder, which you can download from Docker Hub, configure, and launch into a container to transfer node information and resource metrics from Prometheus to Densify.

   Your Densify instance needs to be enabled with Data Forwarder support to accept and ingest incoming data into a data model for optimization. Contact Densify support (Support@Densify.com) to enable Data Forwarder support.

   ***Note**: You will need a Data Forwarder container for each Prometheus server you want to collect data from.*

2. **Analysis**—Analyze collected data based on pre-defined policies to make recommendations.

   The analysis process is initiated after data is ingested into a Densify data model. Results can be pulled from the Densify API or pushed to a webhook URI once the analysis is complete.

The steps below outline a typical workflow to access Kubernetes container recommendations:

1. From the Densify API: Authenticate using JWT—Retrieve token to authenticate each subsequent API call.

2. API: List All Kubernetes Container Analyses—Query for the analysis entity of each cluster for which Kubernetes container data was collected.
3. API: Download Results for Action—Retrieve recommendations for actioning or forward to orchestration engine.

   Alternatively, you can Add a Webhook to an Existing Kubernetes Container Analysis on page 70 to push recommendations to an external URI once the next analysis process completes.
4. Console (optional alternative): Review the Kubernetes Container Optimization report from the Densify Console. See *Optimizing Your Containers* (Help Topic ID 380520) for details.

Authenticate using JWT

Use the `/authorize` resource to retrieve an authenticated token to make subsequent API calls. See Authorize on page 218 for details. By default, the token expires in 5 minutes, therefore you will need to ensure that you have an active token for each Densify API request.

Example: Retrieving an Authenticated Token

Request:

```
POST /authorize
{
    "userName": "apiUser",
    "pwd": "apiPassword"
}
```

Response:

```
{
  "apiToken":
"eyJhbGciOiJIUzUxMiJ9.eyJqdGkiOiIzNzI2Yzk0NC0wMmE4LTRlYzQtOGE2Ny04ODBmMDM2OTRhZ
DciLCJpYXQiOjE1NDI2NTI0MDUsInN1YiI6InZhbiIsImlzcyI6IkRlbnNpZnkuY29tIiwiZXhwIjox
NTQyNjUyNzA1fQ.cJd8qFJfRoPnMEU7GzcdYGBT8WwlgmviQ1OQp8P_
w9VUcjQA3FJaB9QkqJJ6d7zbrY5yjc4w0rOWjY-PPdbmqw",
  "expires" : 1542652705869,
  "status" : 200
}
```

Use the `apiToken` value in your Authorization request header for subsequent Densify API calls.

List All Kubernetes Container Analyses

To see a list of all Kubernetes analyses currently in Densify, use the `/analysis/containers/kubernetes` resource. See List all analyses for a particular platform and vendor on page 157 for details of the resource operation.

Example: List all Saved Kubernetes Container Analyses

Request:

```
GET /analysis/containers/kubernetes
```

Response:

```
   {
      "analysisId": "d7298ac3-a143-41bb-b7d7-62f659f2a8c5",
      "analysisName": "WEST222-Bench",
      "analysisCompletedOn": 1548889723847,
      "href": "/analysis/containers/kubernetes/d7298ac3-a143-41bb-b7d7-
62f659f2a8c5",
      "analysisResults": "/analysis/containers/kubernetes/d7298ac3-a143-41bb-
b7d7-62f659f2a8c5/results",
      "analysisStatus": "/analysis/containers/kubernetes/d7298ac3-a143-41bb-
b7d7-62f659f2a8c5/status"
   }
```

Poll for Analysis Status

Use the `"analysisStatus"` resource element from the List All Kubernetes Container Analyses response to poll for status of the data collection and analysis. See Check for analysis status on page 206 for details of the resource operation. The analysis is complete and recommendations are available when `"analysisStage": "Completed"`.

Example: Checking Kubernetes Analysis Status

Request:

```
GET analysis/containers/kubernetes/d7298ac3-a143-41bb-b7d7-62f659f2a8c5/status
```

Response:

```
{
  "analysisStage": "Completed",
  "webHookStatus": "",
  "statusMessage": "Analysis WEST222-Bench was last completed on Wed Jan 30
19:03:41 EST 2019."
}
```

Add a Webhook to an Existing Kubernetes Container Analysis

You can add a webhook definition to an existing Kubernetes container analysis. The analysis will push the optimized results to the specified webhook URI. See *Analysis: Webhook* (Help Topic ID 340490) for details on managing webhooks.

Example: Adding a Webhook Definition to an Existing Kubernetes Container Analysis

Request:

```
POST /webhook/analysis/containers/kubernetes/d7298ac3-a143-41bb-b7d7-
62f659f2a8c5
{
  "uri": "https://myInstanceConfigServer:443/api/densify/ContainerResults",
  "authType": "bearer",
  "authValue":
"eyJhbGciOiMsIUzUxMiJ9.eyJqdGkiOiJhYWU2MjIxOS1iOWQyLTQ3OGMtYWI3Mi00NGU2OTUzY2Rj
MDEiLCJpYXQiOjE1NDM2MDgxMTEsInN1YiI6ImFkbWluIiwiaXNzIjoiRGVuc2lmeS5jb20iLCJleHA
iOjE1NDM2MDg0MTF9.h3bJrAP-
Z2LeqzjN3FYpFDyoaADvYT1MdLw5SuguqkGE7sjB4c7YgQgv3saj15r2IsgTWH8PW7eNnoZwFP9eiQ"
}
```

Response:

```
{
  "message": "ok",
  "status": 200
}
```

Download Results for Action

Use the `/analysis/containers/kubernetes/<analysisId>/results` resource to retrieve recommendation results and forward to orchestration engine. See Analysis: Kubernetes Container Recommendations on page 183 for details.

Example: Returning Kubernetes Container Recommendations in Terraform-map

Request:

```
GET /analysis/containers/kubernetes/d7298ac3-a143-41bb-b7d7-
62f659f2a8c5/results
```

Headers:

```
Accept: application/terraform-map
Authorization: Bearer <apiToken>
```

Optimizing Microsoft Azure Virtual Machine Instances

This use case includes Microsoft Azure cloud infrastructure data collection, analysis, and generation of right-sizing recommendations for Virtual Machine instances using a combination of Densify Console and Densify API operations. The recommendations, which can be downloaded on demand via Densify API, provide extensive details that can be used to feed downstream automation engines in either JSON or terraform-map formats.

A comprehensive PDF report can be downloaded for each instance to supply supporting evidence for a recommendation; this can be reviewed by a system owner before approvals are given for the suggested changes.

1. **Data Collection**—Collect Azure data into Densify.

   Use the Densify Console to initiate this process.

2. **Analysis**—Analyze collected data based on pre-defined policies to make recommendations.

   The analysis process is initiated automatically after data collection completes. Results can be immediately pulled from the Densify API or pushed to a webhook URI once the analysis is complete.

3. **Reporting Database Update**—Update reporting database tables with analysis results for reporting.

   The reporting database update process is scheduled to run nightly after the analysis process. The Densify Console Azure optimization opportunity report and the Impact Analysis and Recommendation reports are available after the reporting database update is complete.

***Note**: Before you can collect Microsoft Azure cloud infrastructure data in Densify, you need to create and configure an Azure account with access to your Azure subscriptions. See Microsoft Azure Data Collection Prerequisites (Help Topic ID 410110) or Microsoft Azure Data Collection Prerequisites for a Service Principal (Help Topic ID 410010) for details.*

The steps below outline a typical workflow to optimize Microsoft Azure Virtual Machine instances:

1. Collect Azure data through the Densify Console using the Cloud Connections wizard. See *Configuring a Microsoft Azure Connection* from the topic *Using the Public Cloud Connection Wizard* (Help Topic ID 380290) for details.

   After data collection completes, an analysis is initiated for each subscription associated with the Azure account configured for data collection.

2. From the Densify API: Authenticate using JWT—Retrieve token to authenticate each subsequent API call.

3. API: List All Azure Analyses—Query for the analysis entity of each subscription for which data was collected (from step 1). You can use the returned analysis entity to check for status and recommendations.

4. API: Poll for Analysis Status—Check for data collection and analysis status. The recommendations are available when analysis is complete.

5. API: Download Results for Action—Retrieve recommendations for actioning or forward to orchestration engine.

   Alternatively, you can Add a Webhook to an Existing Azure Analysis on page 75 to push recommendations to an external URI once the next analysis process completes.

6. API/Console (optional): Download Impact Analysis and Recommendation Instance Report for system owners. This can achieved from the Densify API or from the Densify Console. See *Viewing the Impact Analysis and Recommendation Report* (Help Topic ID 380450) for details of the report from the Densify Console.

7. Console (optional alternative): Review Azure Virtual Machine optimization opportunity report from the Densify Console. See *Azure* from the topic *Optimizing Your Public Cloud* (Help Topic ID

380320) for details. The optimization opportunity report is available after the nightly reporting database update.

8. Console (optional): Review Azure optimization policy settings used during analysis from the Densify Console. See *Cloud Optimization Policies* from the topic *Viewing Policy Settings* (Help Topic ID 120200) for details.

Postman Collection

Examples of Densify API request for this use case can be found in the following Postman collection:

Download the latest Densify public cloud Postman collection (v 12.1.7) from https://www.densify.com/docs-api/Content/API_Guide/Postman_Collection.htm.

Follow the steps below to use the downloaded Postman collection:

1. Unzip the downloaded file and import it into your Postman application. The **Densify API Collection (12.1.7)** collection and **Densify** environment are loaded into your Postman workspace.
2. Modify the variables in the **Densify** environment to match your Densify and cloud-specific settings and credentials.

   Note: *If you already have a previous* ***Densify*** *environment in your Postman application, you can either delete the previous version or rename it. Otherwise, you will have duplicate Densify environments after the new collection is imported.*

3. Review the **Documentation** section of the collection for an overview of the workflow and API requests.

Use this sample collection to familiarize yourself with Densify API requests.

Authenticate using JWT

Use the `/authorize` resource to retrieve an authenticated token to make subsequent API calls. See Authorize on page 1 for details. By default, the token expires in 5 minutes, therefore you will need to ensure that you have an active token for each Densify API request.

Example: Retrieving an Authenticated Token

Request:

```
POST /authorize
{
    "userName": "apiUser",
    "pwd": "apiPassword"
}
```

Response:

```
{
  "apiToken": "eyJh-
bGciOiJIUzUxMiJ9.eyJqdGkiOiIzNzI2Yzk0NC0wMmE4LTRlYzQtOGE2Ny04ODBmMDM2OTRhZD-
ciLCJpYXQiOjE1NDI2NTI0MDUsInN1YiI6InZh-
biIsIm-
lzcyI6IkRlb-
nNpZnkuY29tIiwiZXhwIjoxNTQyNjUyNzA1fQ.cJd8qFJfRoPnMEU7GzcdYGBT8WwlgmviQ1OQp8P_
w9VUcjQA3FJaB9QkqJJ6d7zbrY5yjc4w0rOWjY-PPdbmqw",
  "expires" : 1542652705869,
  "status" : 200
}
```

Use the `apiToken` value in your Authorization request header for subsequent Densify API calls.

List All Azure Analyses

To see a list of all Azure analyses currently in Densify, use the `/analysis/azure` resource. See List all analyses for a particular platform and vendor on page 1 for details of the resource operation.

Example: List all Saved Azure Analyses

Request:

```
GET /analysis/cloud/azure
```

Response:

```
{
   "analysisId": "398d26f3-b705-4fa6-8d31-16724ae320a2",
   "analysisName": "00d89cbc-bc00-4d00-bcf6-ce6ec08d8fbc",
   "analysisCompletedOn": 1522179715493,
   "href": "/analysis/cloud/azure/398d26f3-b705-4fa6-8d31-16724ae320a2",
   "analysisResults": "/analysis/cloud/azure/398d26f3-b705-4fa6-8d31-
16724ae320a2/results",
   "analysisStatus": "/analysis/cloud/azure/398d26f3-b705-4fa6-8d31-
16724ae320a2/status"
}
```

Poll for Analysis Status

Use the `"analysisStatus"` resource element from the List All Azure Analyses response to poll for status of the data collection and analysis. See Check for analysis status on page 1 for details of the resource operation. The analysis is complete and recommendations are available when `"analysisStage": "Completed"`.

Example: Checking Azure Analysis Status

Request:

```
GET /analysis/cloud/azure/398d26f3-b705-4fa6-8d31-16724ae320a2/status
```

Response:

```
{
  "analysisStage": "Completed",
  "webHookStatus": "",
  "statusMessage": "Analysis 00d89cbc-bc00-4d00-bcf6-ce6ec08d8fbc was last com-
pleted on Mon Feb 25 11:25:11 EST 2019."
}
```

Add a Webhook to an Existing Azure Analysis

You can add a webhook definition to an existing Azure analysis. The analysis will push the optimized results to the specified webhook URI. See *Add a webhook to an analysis* in the topic *Analysis: Webhook* (Help Topic ID 340490) for details on adding a webhook.

Example: Adding a Webhook Definition to an Existing Azure Analysis

Request:

```
POST /webhook/analysis/cloud/azure/398d26f3-b705-4fa6-8d31-16724ae320a2
{
  "uri": "https://myInstanceConfigServer:443/api/densify/results",
  "authType": "bearer",
  "authValue": "eyJh-
bGciOiMsIUzUxMiJ9.eyJqdGkiOiJhYWU2MjIxOS1iOWQyLTQ3OGMtYWI3Mi00NGU2OTUzY2RjMDEi-
LCJpYXQiOjE1NDM2MDgxMTEsInN1YiI6ImFk-
bWluIiwiaXNzIjoiRGVuc2lmeS5jb20iLCJleHAiOjE1NDM2MDg0MTF9.h3bJrAP-
Z2LeqzjN3FYpFDyoaADvYT1MdLw5SuguqkGE7s-
jB4c7YgQgv3saj15r2IsgTWH8PW7eNnoZwFP9eiQ"
}
```

Response:

```
{
  "message": "ok",
  "status": 200
}
```

Download Results for Action

Use the `/analysis/cloud/azure/<analysisId>/results` resource to retrieve recommendation results and forward to orchestration engine. See Analysis: Azure Recommendations on page 1 for details.

Example: Returning Recommendations in Terraform-map

Request:

```
GET /analysis/cloud/azure/398d26f3-b705-4fa6-8d31-16724ae320a2/results
```

Headers:

```
Accept: application/terraform-map
Authorization: Bearer <apiToken>
```

Download Impact Analysis and Recommendation Instance Report

Use the `rptHref` resource element provided in the instance recommendation output to download a PDF Impact Analysis and Recommendation Report for each instance. This report is available after a reporting database update, which is scheduled to run nightly. See rptHref on page 1 and Example: Downloading an Impact Analysis and Recommendation Report on page 1 for details.

Example: Downloading an Impact Analysis and Recommendation Report

Request:

```
GET /systems/bdbdbd19-6928-4e7a-894c-e7e215687a4d/analysis-report
```

Headers:

```
Accept: application/octet-stream
Authorization: Bearer <apiToken>
```

Subscribing to Densify Recommendations

This use case showcases how to subscribe to a set of Densify recommendation notifications using the Densify Subscription Service (DSS) API framework.

The DSS is a flexible framework that offers you the ability to customize the content of your Densify recommendations for targeted distribution. DSS features include:

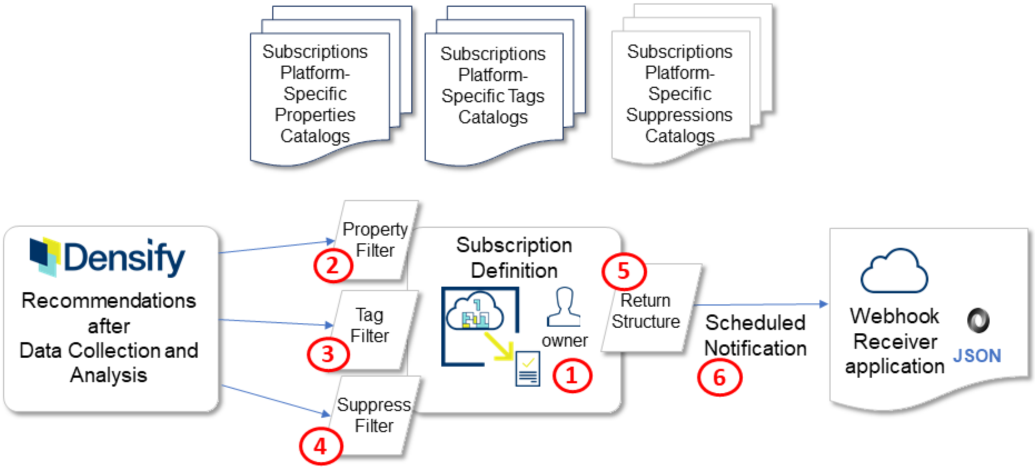
- ability to filter recommendation systems based on recommendation fields (property filter);
- ability to filter recommendation systems based on attribute tags (tag filter);
- ability to suppress recommendation systems, for a period of time, based on recommendation fields and/or attribute tags via suppression filters;
- ability to customize the notification output format;
- ability to send the notification to an external webhook;
- ability to schedule the notification;
- ability to have private subscription definitions, as well as global subscription definitions for shared usage;
- supports both cloud and containers system recommendations.

To use these features, you need to create an instance of the Subscription resource.

Creating a Subscription

The diagram below provides a high-level overview of creating an instance of the Subscription resource through the Densify API:

Figure: Subscription High Level Overview


Subscriptions Platform-Specific Properties Catalogs
Subscriptions Platform-Specific Tags Catalogs
Subscriptions Platform-Specific Suppressions Catalogs
Densify
Recommendations after Data Collection and Analysis
Property Filter
2
Tag Filter
3
Suppress Filter
4
Subscription Definition
owner
1
5
Return Structure
Scheduled Notification
6
Webhook Receiver application
JSON

An instance of the Subscriptions on page 428 resource is created using the `POST /subscriptions/<platformType>` request. You must identify the *<platformType>* of the subscription and use similar platform resources to define the subscription. For instance, you need to use container resources for a subscription to container recommendations. After identifying the subscription platform resources to use, define the parameter components below to customize your subscription.

1. **Define the Subscription**—Specify the subscriptionName on page 434, description on page 435, and owner on page 434.

   The `subscriptionName` is used as a key identifier, so it needs to be unique for global subscriptions as well as unique within a private-scoped group of subscriptions. The `description` parameter is for documentation purposes, so its content should be as descriptive as possible to indicate the purpose of the subscription.

   See Example: Defining the Subscription on page 84.

2. **Property Filter**—Specify propertyReferences on page 435 to filter the recommendation systems you are interested in receiving based on recommendation fields.

   This feature is similar to the Recommendation Filter Menu in the Densify Console, where you can filter recommendation systems based on selected fields from the menu. In the `propertyReferences` filter condition, fields are selected by referencing property elements from the platform-specific Subscriptions Properties catalog. The platform of properties you reference must correspond with the platform of your subscription. Refer to Subscriptions Properties Catalogs for details on referencing properties.

   See Example: Specifying Property Filters on page 85 for an example of a property filter condition block in a `POST /subscriptions/<platformType>` request.

3. **Tag Filter**—Specify tagReferences on page 436 to filter the recommendation systems you are interested in receiving based on attribute fields.

   This feature is similar to the Guest Filter from the Densify Console, but it also extends to filter platform-specific attribute tags. In the `tagReferences` filter condition, attributes are selected by referencing tag elements from the platform-specific Subscriptions Tags catalog. The platform of tags you reference must correspond with the platform of your subscription. Refer to Subscriptions Tags Catalogs for details.

   ***Note**: You must ensure that the selected Densify attributes are marked as `searchable` for use with DSS. Contact Densify support (Support@Densify.com) to confirm that the selected attributes are searchable in your Densify instance.*

   See Example: Specifying Tag Filters on page 85 for an example of a tag filter condition block in a `POST /subscriptions/<platformType>` request.

4. **Suppress Filter**—Specify the suppressionReferences on page 437 condition to suppress recommendation systems you are not interested in receiving based on recommendation or attribute fields.

   This feature allows you to suppress groups of recommendation systems for a period of time when the notification is actively scheduled. Suppression conditions are created by referencing suppression elements from the platform-specific Subscriptions Suppressions catalog. The platform of suppressions you reference must correspond with the platform of your subscription. Refer to Subscriptions Suppressions Catalogs for an explanation of referencing the catalog.

   See Example: Specifying Suppression Conditions on page 86 for an example of a suppression condition block in a `POST /subscriptions/<platformType>` request.

5. **Return Structure**—Specify the returnStructure on page 438 to personalize the output of the notification.

   If you do not specify the `returnStructure` parameter in your subscription, all recommendation fields applicable to the platform-specific system are returned. To return specific fields, you can specify elements from the platform-specific Subscriptions Properties Catalogs and Subscriptions Tags Catalogs that are within your scope.

   See Example: Specifying the Return Structure on page 86 for an example of a `returnStructure` block in a `POST /subscriptions/<platformType>` request.

6. **Schedule Notification to Webhook**—Specify when and where to send your notification by configuring the schedule on page 437 parameter and the webhook on page 437 parameter.

   If you do not specify the `schedule` parameter, the notification is triggered each night after data collection, analysis, and reporting database update processes. See Example: Specifying the Notification Schedule on page 88 for an example of specifying a `schedule` block in a `POST /subscriptions/<platformType>` request.

The `webhook` parameter is mandatory if you want notifications to be triggered. See Example: Specifying the Webhook on page 87 for an example of specifying a `webhook` block in a `POST /subscriptions/<platformType>` request.

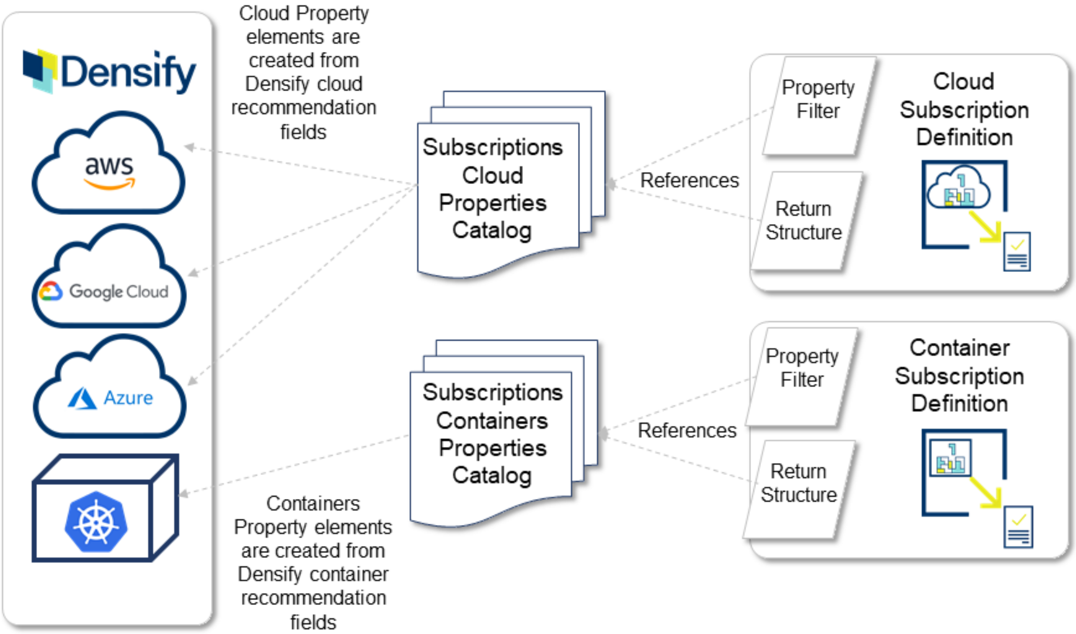
> ***Note**: You can test the subscription and request an on-demand results output by using the Subscriptions: Results on page 455 resource, even if the webhook is not defined.*

See Example: Creating a Subscription (Putting It All Together) for a complete example of the `POST /subscriptions/<platformType>` request with all the combined parameter components; then review the on-demand results of the subscription in Example: Getting On-Demand Results.

Refer to Subscriptions on page 428 for the complete subscription resource reference.

Subscriptions Properties Catalogs

Figure: Subscriptions Properties Catalogs Reference Overview


Densify
aws
Google Cloud
Azure
Cloud Property elements are created from Densify cloud recommendation fields
Subscriptions Cloud Properties Catalog
References
Property Filter
Return Structure
Cloud Subscription Definition
Subscriptions Containers Properties Catalog
Containers Property elements are created from Densify container recommendation fields
References
Property Filter
Return Structure
Container Subscription Definition

The platform-specific Subscriptions Properties catalogs are resources that provides you with a list of recommendation fields to use for filtering and for personalizing the notification output. The platform of the Subscriptions Properties catalog you use for filtering must correspond with the platform of your subscription.

By default, the platform-specific Subscriptions Properties catalogs already have a list of recommendation fields for you to use. You can also add additional recommendation fields to the catalogs. The list of possible recommendation fields to add corresponds to the response elements from the Densify supported cloud and container recommendations (see Analysis: AWS

Recommendations: Response on page 108, Analysis: Azure Recommendations: Response on page 135, Analysis: GCP Recommendations: Response on page 168, and Analysis: Kubernetes Container Recommendations: Response on page 186).

***Note**: Some recommendation elements are not common to all technologies. For example, propertyName="minGroupRecommended" only applies to AWS Auto Scaling group recommendations.*

When you create property filter conditions for your subscription, you need to reference properties from the corresponding platform Subscriptions Properties catalog (see the Subscriptions: propertyReferences on page 435 parameter). Similarly, you need to reference properties from the corresponding platform Subscriptions Properties catalog to personalize the notification output (see the Subscriptions: returnStructure on page 438 parameter).

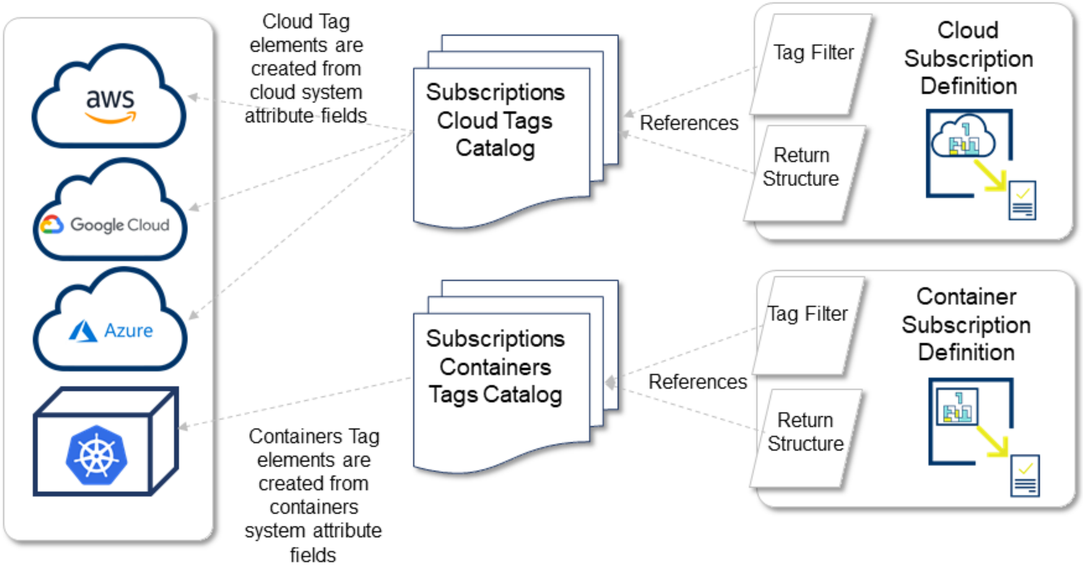
You can only reference properties that are accessible to the Densify username you use to authorize the API request. Property accessibility depends on the scope of the property: if the property is *global*, then the property is accessible to all API Densify users; if the property is *private*, then the property is accessible to the owner of the private property. If you are an **administrative user**[1], then you can override the property scope rule and access all properties from the catalog. See Subscriptions: Properties: owner on page 471 for more property scope details.

Refer to Subscriptions: Properties on page 462 for the full resource definition.

[1] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

Subscriptions Tags Catalogs

Figure: Subscriptions Tags Catalogs Reference Overview


aws
Google Cloud
Azure
Cloud Tag elements are created from cloud system attribute fields
Subscriptions Cloud Tags Catalog
References
Tag Filter
Return Structure
Cloud Subscription Definition
Subscriptions Containers Tags Catalog
References
Tag Filter
Return Structure
Container Subscription Definition
Containers Tag elements are created from containers system attribute fields

The platform-specific Subscriptions Tags catalogs are resources that provides you with a list of system attribute fields to use for filtering and for personalizing the notification output. The platform of the Subscriptions Tags catalog you use for filtering must correspond with the platform of your subscription.

By default, the platform-specific Subscriptions Tags catalogs already have a list of common attributes for you to use. You can also add additional attribute fields to the catalogs. The possible attribute fields you can add comes from the set of Densify standard attributes or technology-specific attributes.

When you create tag filter conditions for your subscription, you need to reference tags from the corresponding platform Subscriptions Tags catalog (see the Subscriptions: tagReferences on page 436 parameter). Similarly, you need to reference tags from the corresponding platform Subscriptions Tags catalog to personalize the notification output (see the returnStructure parameter).

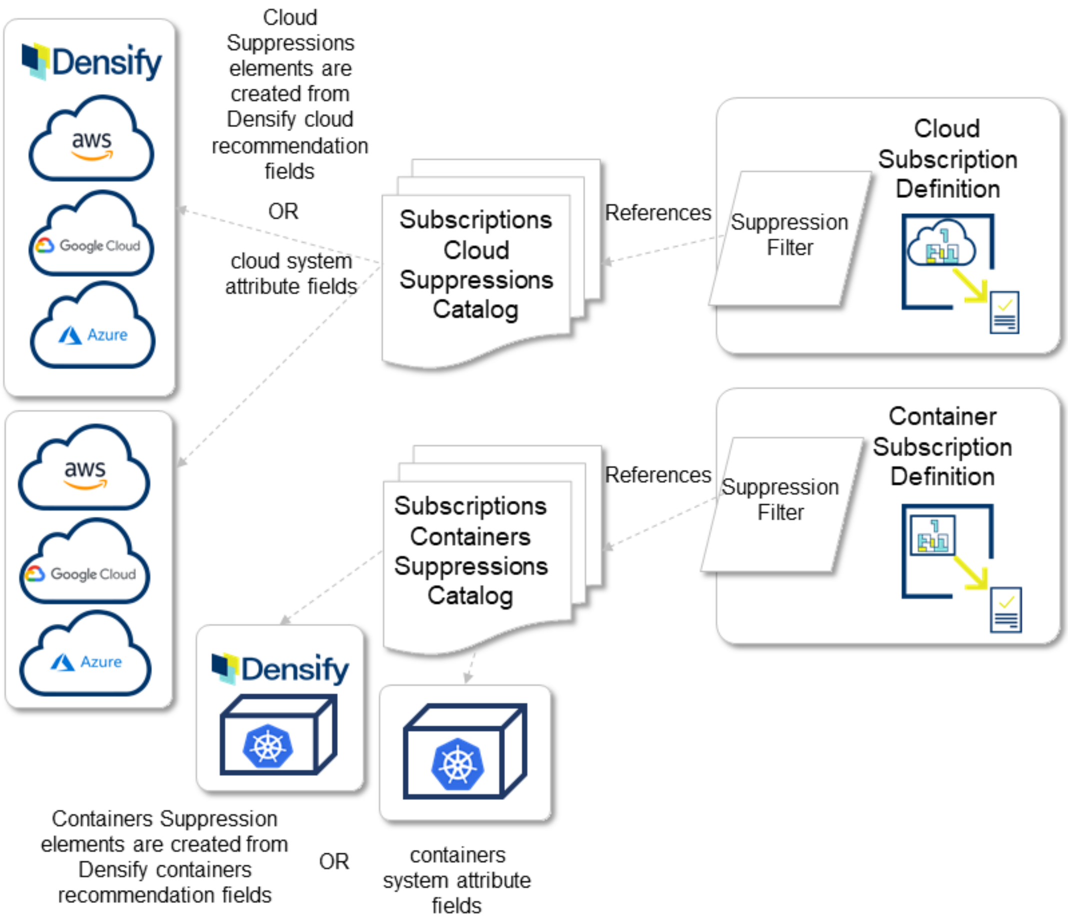
You can only reference tags that are accessible to the Densify username you use to authorize the API request. Access to a particular tag in the catalog depends on its scope: if the tag is *global*, then the tag is accessible to all API Densify users; if the tag is *private*, then the tag is accessible to the owner of the private tag. If you are an **administrative user**[1], then you can override the tag scope rule and access all tags from the catalog. See Subscriptions: Tags: owner on page 518 for additional details of tag scope.

[1] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

Refer to Subscriptions: Tags on page 510 for the full resource definition.

Subscriptions Suppressions Catalogs

Figure: Subscriptions Suppressions Catalogs Reference Overview


Densify
aws
Google Cloud
Azure
Cloud Suppressions elements are created from Densify cloud recommendation fields
OR
cloud system attribute fields
Subscriptions Cloud Suppressions Catalog
References
Suppression Filter
Cloud Subscription Definition
aws
Google Cloud
Azure
Subscriptions Containers Suppressions Catalog
References
Suppression Filter
Container Subscription Definition
Densify
Containers Suppression elements are created from Densify containers recommendation fields
OR
containers system attribute fields

The platform-specific Subscriptions Suppressions catalogs are resources that provides you with a list of recommendation and system attribute fields to use for creating suppression conditions in your subscription. The platform of the Subscriptions Suppressions catalog you use for suppressing recommendations must correspond with the platform of your subscription.

By default, the platform-specific Subscriptions Suppressions catalog already has a list of recommendation and attribute field suppressions for you to use. You can also add additional suppression fields. The list of possible recommendation fields correspond to the response elements from the Densify supported cloud recommendations (see Analysis: AWS Recommendations: Response on page 108, Analysis: Azure Recommendations: Response on page 135, Analysis: GCP Recommendations: Response on page 168, and Analysis: Kubernetes Container Recommendations:

Response on page 186). The possible attribute fields you can add to a Subscriptions Suppressions catalog comes from the set of Densify standard attributes or technology-specific attributes.

When you create suppression conditions for your subscription, you need to reference suppressions from the corresponding platform Subscriptions Suppressions catalog (see the Subscriptions: suppressionReferences on page 437 parameter).

You can only reference suppressions that are accessible to the Densify username you use to authorize the API request. Access to a particular suppression in the catalog depends on its scope: if the suppression is *global*, then it is accessible to all API Densify users; if the suppression is *private*, then it is accessible to the owner of the private suppression. If you are an **administrative user**[1], then you can override the suppression scope rule and access all suppressions from the catalog. See Subscriptions: Suppressions: owner on page 502 for details of suppression scope.

***Note**: The Subscriptions Suppressions catalogs are not used for a Subscription's return output.*

Refer to Subscriptions: Suppressions on page 491 for the full resource definition.

Example: Defining the Subscription

Use the `POST /subscriptions/cloud` request to create a cloud subscription. The first set of parameters to specify are the subscriptionName and description, which are free-form strings. The owner parameter is automatically set to your Densify username if you are not an **administrative user**[2]. You can also set the active parameter to indicate if the subscription is dormant or active (the default is `active`). Remember to specify your active authorization key for each token-based authentication request (see Authorize on page 218 for details).

Example: Specifying Subscription Name and Description

Headers:

```
Accept: application/json
Authorization: Bearer <apiToken>
```

Request:

```
POST /subscriptions/cloud

{
```

[1] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

[2] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

```
 "subscriptionName": "My Subscription",
 "description": "A subscription for my Use Case",
 "active": true,
 ...
```

Example: Specifying Property Filters

To specify property filter conditions in the `POST /subscriptions/`*`<platformType>`* request, use the propertyReferences on page 435 parameter. In this example, the property condition filters all AWS EC2s with predicted uptime between 50 -100%. The cloud subscription `propertyReferences` condition references the `propertyID` of "`serviceType`" and "`predictedUptime`" from the Cloud Subscriptions Properties catalog.

Example: Subscription - Property Filter Condition

Request Parameter - Property Filter Condition Section:

```
...
 "propertyReferences": [
    {
        "values": ["EC2"],
        "propertyID": "b4e7260f-1aae-4150-b75d-1b1234075500",
        "operator": "="
    },
    {
        "propertyID": "f2a38773-db60-478a-9982-1a2d1ba7d380",
        "operator": "[]",
        "values": [50,100]
    }
  ],
 ...
```

Example: Specifying Tag Filters

To specify tag filter conditions in the `POST /subscriptions/`*`<platformType>`* request, use the tagReferences on page 436 parameter. In this example, the tag condition filters all systems belonging to the "Sales" department. The cloud subscription `tagReferences` condition references the `tagID` of "`Department`" from the Cloud Subscriptions Tags catalog.

Example: Subscription - Tag Filter Condition

Request Parameter - Tag Filter Condition Section:

```
...
"tagReferences": [
   {
       "values": ["Sales"],
       "tagID": "9879308e-3162-499b-8fe5-7fd67e627079",
       "operator": "="
   }
 ],
...
```

Example: Specifying Suppression Conditions

To specify suppression conditions in the `POST /subscriptions/<platformType>` request, use the suppressionReferences on page 437 parameter. In this example, the cloud suppression condition removes all terminate recommendations until April 20, 2020. After April 20, 2020 (the `revokeBy` date), the suppression condition is deprecated. The cloud subscription `suppressionReferences` condition references the `suppressionID` of "`recommendationType`" from the Cloud Subscriptions Suppressions catalog.

Example: Subscription - Suppression Filter Condition

Request Parameter - Suppression Filter Condition Section:

```
...
"suppressionReferences": [
   {
       "values": ["Terminate"],
       "suppressionID": "a6827ae4-fa2b-405e-a564-d70f2dad45c2",
       "operator": "=",
       "revokeBy": 1587415413000
   }
 ],
...
```

Example: Specifying the Return Structure

To specify a customized return structure in the `POST /subscriptions/<platformType>` request, use the returnStructure on page 438 parameter. In this example, the return output will contain the following cloud fields: `effortEstimate`, `entityId`, `name`, `recommendationType`, `savingsEstimate`, `serviceType`, `Department` (attribute), `Virtual Domain` ("account"). These fields reference the `propertyID` and "`tagID`" from the Cloud Subscriptions Properties catalog and the Cloud Subscriptions Tags catalog, respectively. Some of these property or tag fields have the `"useAlias"` flag set to true, which returns the alias names instead of the property or tag names.

Example: Subscription - Return Structure

Request Parameter - Return Structure Section:

```
...
"returnStructure": {
    "properties": [
        {
            "propertyID": "b4e7260f-1aae-4150-b75d-1b1234075500"
        },
        {
            "useAlias": true,
            "propertyID": "e2ae92c5-91c7-4ff5-a29f-99aa92d65178"
        },
        {
            "propertyID": "08d84679-1816-4cd7-b766-e4ad441b9a6c"
        },
        {
            "propertyID": "b20b2499-e901-4d79-873c-5e953ae8be33"
        },
        {
            "useAlias": true,
            "propertyID": "cebcd841-89d8-4007-a4c6-1f0b06723db4"
        },
        {
            "propertyID": "88695e33-19c4-4813-819a-31e59322da6e"
        }
    ],
    "tags": [
        {
            "useAlias": true,
            "tagID": "be3cdc69-962e-4461-b8c3-07cec8e712be"
        },
        {
            "tagID": "9879308e-3162-499b-8fe5-7fd67e627079"
        }
    ]
},
...
```

Example: Specifying the Webhook

For the destination of the notification, specify the webhook on page 437 parameter in the `POST /subscriptions/<platformType>` request. If you do not specify a `webhook`, no subscription notifications are triggered; the subscription is considered dormant since there is no destination for the notification. You can test the subscription and request an on-demand results output by using the Subscriptions: Results on page 455 resource, even if the subscription is dormant or if the webhook is not defined.

Example: Subscription - Webhook

Request Parameter - Webhook Section:

```
    ...
    "webhook": {
        "uri": "https://testsystem.com/webhook/receiver",
        "authType": "basic",
        "authValue": "test:test"
    },
    ...
```

Example: Specifying the Notification Schedule

To specify a notification schedule in the `POST /subscriptions/<platformType>` request, use the schedule on page 437 parameter. In the example below, the cloud subscription is scheduled to be triggered on Mondays and Fridays of each week (`dayofWeek 1` = Monday). If you do not specify the `schedule` parameter, the subscription notifications are triggered nightly by default.

Example: Subscription - Schedule

Request Parameter - Schedule Section:

```
    ...
    "schedule": {
        "dayofWeek": [
            1,
            5
        ]
    },
    ...
```

Example: Creating a Subscription (Putting It All Together)

This example puts all the subscription components (from the previous examples) together to create a cloud subscription.

Example: Creating a Cloud Subscription

Request:

```
POST /subscriptions/cloud
```

Parameters:

```
[
    {
        "subscriptionName": "My Subscription",
        "description": "A subscription for my Use Case",
        "active": true,
        "webhook": {
            "uri": "https://testsystem.com/webhook/receiver",
            "authType": "basic",
            "authValue": "test:test"
        },
        "propertyReferences": [
            {
                "values": ["EC2"],
                "propertyID": "b4e7260f-1aae-4150-b75d-1b1234075500",
                "operator": "="
            },
            {
                "propertyID": "f2a38773-db60-478a-9982-1a2d1ba7d380",
                "operator": "[]",
                "values": [50,100]
            }
        ],
        "tagReferences": [
            {
                "values": ["Sales"],
                "tagID": "9879308e-3162-499b-8fe5-7fd67e627079",
                "operator": "="
            }
        ],
        "suppressionReferences": [
            {
                "values": ["Terminate"],
                "suppressionID": "a6827ae4-fa2b-405e-a564-d70f2dad45c2",
                "operator": "=",
                "revokeBy": 1587415413000
            }
        ],
        "returnStructure": {
            "properties": [
                {
                    "propertyID": "b4e7260f-1aae-4150-b75d-1b1234075500"
                },
                {
                    "useAlias": true,
                    "propertyID": "e2ae92c5-91c7-4ff5-a29f-99aa92d65178"
                },
                {
                    "propertyID": "08d84679-1816-4cd7-b766-e4ad441b9a6c"
                },
                {
                    "propertyID": "b20b2499-e901-4d79-873c-5e953ae8be33"
                },
                {
                    "useAlias": true,
                    "propertyID": "cebcd841-89d8-4007-a4c6-1f0b06723db4"
                },
                {
                    "propertyID": "88695e33-19c4-4813-819a-31e59322da6e"
                }
            ],
            "tags": [
                {
                    "useAlias": true,
                    "tagID": "be3cdc69-962e-4461-b8c3-07cec8e712be"
                },
                {
                    "tagID": "9879308e-3162-499b-8fe5-7fd67e627079"
                }
            ]
        },
        "schedule": {
            "dayofWeek": [
                1,
                5
            ]
        }
    }
]
```

Response:

```
[
    {
        "subscriptionRef": "7f2b2f50-3988-4275-a1ac-34a9b45408b9",
        "subscriptionName": "My Subscription"
    }
]
```

Example: Getting On-Demand Results

This example retrieves the cloud subscription (created from the previous example) results on-demand. This request returns the cloud subscription results regardless of what is configured in the `webhook`, `schedule`, or `active` parameters.

Example: Getting Cloud Subscription Results On-Demand

Request:

```
GET /subscriptions/cloud/7f2b2f50-3988-4275-a1ac-34a9b45408b9
```

Response:

```
{
    "subscription": {
        "name": "My Subscription",
        "description": "A subscription for my Use Case",
        "created": "Mon Apr 13 13:50:15 EST 2020",
        "createdBy": "saas",
        "lastRefreshed": "Fri Apr 17 1:00:10 EST 2020",
        "owner": "saas"
    },
    "count": 98,
    "results": [
        {
            "effortEstimate": "Low",
            "entityId": "c0ebf351-9b6c-4d57-b3d0-23c7861f9db4",
            "name": "ex-pro-asop-896",
            "recommendationType": "Downsize - Optimal Family",
            "savingsEstimate": "49.39808",
            "serviceType": "EC2",
            "divider": "-------------------------",
            "Department": "Sales",
            "awsAccount": "229132289155"
        },
        {
            "effortEstimate": "Low",
            "entityId": "234ca9b6-3def-4ace-87c5-466c2e41f1d1",
            "name": "ex-pro-kotl-966",
            "recommendationType": "Downsize - Optimal Family",
            "savingsEstimate": "163.6333",
            "serviceType": "EC2",
            "divider": "-------------------------",
            "Department": "Sales",
            "awsAccount": "229132289155"
        },
        {
        // ... *SNIP* additional 96 system recommendations not displayed
*SNIP* ...
        }
    ]
}
```

Postman Collection

Examples of Densify API request for this use case can be found in the following Postman collection:

Download the latest Densify Supply and Demand Postman collection (v14.8) from https://www.densify.com/docs-api/WebHelp_Densify_API/Content/API_Guide/Postman_Collection.htm.

Download the latest Densify Public Cloud Postman collection from https://www.densify.com/docs-api/WebHelp_Densify_API/Content/API_Guide/Postman_Collection.htm.

Follow the steps below to use the downloaded Postman collection:

1. Unzip the downloaded file and import both the **Densify API Collection** collection and **Densify Environment** variables into your Postman workspace.
2. Modify the variables in the **Densify Environment** to match your Densify settings and credentials.

   Note: *If you already have a* ***Densify*** *environment in your Postman application, you can either delete the previous version or rename it. Otherwise, you will have duplicate Densify environments after the new collection is imported.*

3. Review the **Documentation** section of the collection for an overview of the workflow and API requests.
4. Use this sample collection to familiarize yourself with Densify API requests.

Analysis: AWS Analyze

Description

The `/analysis/cloud/aws/analyze` resource is used to collect AWS CloudWatch data and initiate optimization analysis with the cloud infrastructure collected. Below are the series of processes that occur when the initial `/analysis/cloud/aws/analyze` request is triggered:

- set up and initiate data collection of the AWS account and schedule it to run automatically on a nightly basis;
  - the first data collection audit will collect up to 60 days of data, if available;
  - subsequent data collection will collect the last day's data, if available;
- initiate analysis on the data collected using the default policy;
  - subsequent analysis is scheduled to run on a nightly basis after data collection;
  - you have the option to override the default policy used during an analysis (see GET /analysis/cloud/aws/policy on page 201 for a list of available AWS policies);
  - you have the option to configure a webhook URI, where the results will be sent upon analysis completion (see Add webhook to an analysis on page 212 for details).

While data collection or analysis is in progress, you can check the status (using the `/analysis/cloud/aws/<analysisId>/status` resource) or wait for the results to be published to an optional webhook URI.

The reporting database update is typically scheduled to run automatically, on a nightly basis after data collection and analysis are complete. This scheduled job produces reports for each instance recommendation, which is useful for analysts or application owners. These application owner reports are created on a nightly basis after the scheduled analysis, and may therefore only be available on the following day for a new analysis. Exact timing depends on the size of your environment.

Ad-Hoc Tasks

Generally you do not need to run once-off tasks as both data collection and analysis tasks are scheduled automatically. In cases where you need make an ad-hoc request in addition to the scheduled job, the functionality exists for this endpoint.

Historical Data Collection

When Densify initiates data collection, normally the first audit collects 60 days of historical data. This is run once and subsequent audits collect only the last day's data.

If the initial historical audit has been disabled for performance reasons, you will need to run an ad-hoc task to collect the historical data.

A connection to the specified cloud account must already exist before you can run an ad hoc audit. When you execute an ad hoc refresh an audit task will be configured but a new connection will not be created. If the cloud connection does not already exist and the API POST contains `triggerAdhocAudit=true`, then you will get an error message.

If there is more than one account associated with the specified account ID (i.e. a payer account with many linked accounts), the Densify API handles it in the same way that analyses are currently rerun using the POST operation.

Once the audit is complete you need to rerun the associated analyses as indicated below or you can wait for the next scheduled execution of the analyses and RDB populate.

Analysis Update

You can make an ad-hoc request to refresh an existing analysis, outside of the scheduled nightly run using `/analysis/cloud/<aws|azure|gcp>/analyze`. This manual, ad hoc analysis request does not perform data collection or reporting database updates. It only runs the analysis on the existing data collected with the following behavior:

- If a new policy is provided, the analysis will use the new policy/settings. If no policy is provided, the analysis will use the existing policy that is already configured for the analysis.
- If a new webhook is provided, the analysis will send results to the new webhook URI. If no webhook is provided, the analysis will send results to the existing webhook, if configured.
- If the same analysis is already running, the request does not proceed and an appropriate message is returned.
- If the specified analysis has data collection scheduled within 30 minutes, the request does not proceed and an appropriate message is returned. For example, if data collection is scheduled to run

at 12:05 AM, and you initiate a manual, ad hoc analyze request at 11:45 PM, then the analysis will not proceed and an error message is returned.

Prerequisite Configuration

Before you can collect AWS CloudWatch data, you need to create an IAM role for Densify to have a trust relationship with your AWS account that contains data. See *AWS Data Collection Prerequisites for an IAM Role* (Help Topic ID 410060) for details on how to set up the IAM role for data collection.

Resource

```
/analysis/cloud/aws/analyze
/analysis/cloud/aws
```

Supported Operations

Table: AWS Analyze Supported Operations

Operation	HTTP Method	Input	Output	Description
Run AWS data collection and analysis	`POST /analysis/cloud/aws/analyze`	Request Body Parameter: accountId roleArnName roleExternalId connectionName (optional) policyInstanceId *(optional)* webHook *(optional)*	message status	This resource operation is used to: 1. Collect AWS cloud data connected via IAM role access. 2. Run analysis on AWS cloud data collected. 3. (Optional) Send results to webhook receiving application. 4. Schedule data collection and analysis processes

Operation	HTTP Method	Input	Output	Description
				each night subsequent to the initial request. Example: Running AWS Data Collection and Analysis
Re-run AWS data analysis	`POST /analysis/cloud/aws/analyze`	Request Body Parameter: accountId roleArnName roleExternalId policyInstanceId *(optional)* webHook *(optional)*	href message status	This resource operation is used to re-run an analysis that already exists. You can specify an updated policy and/or webhook to use for the analysis. Data collection is not run. Data collection only occurs during the first `/analyze` request, and is then scheduled to run nightly The updated policy or webhook is saved and will be used in future, scheduled analyses. You cannot initiate a request if data collection or the analyses are in progress or within 30 minutes of the time that these tasks are scheduled to run.
List all generated	`GET /analysis/cloud/aws/`	Path Parameter: N/A	Lists all analyses that	This resource operation is used

Operation	HTTP Method	Input	Output	Description
analyses		Request Body Parameter: N/A	have been created with details.	to obtain the analysis ID that is required for other operations.
Run the 60-day historical AWS audit	`POST /analysis/cloud/aws/analyze`	Request Body Parameter: accountId roleArnName roleExternalId triggerAdhocAudit	href message status	This resource operation is used to re-run an audit for which a connection and daily, scheduled audit already exists. You can optionally specify the number of days of historical data to collect. If not specified the previous 60 days from yesterday's date are collected. If you initiate an audit request when data collection or analysis is already running or within 30 minutes of the time that these tasks are scheduled to run, then the request will fail and an error message is returned.
Update AWS credentials	`PUT /analysis/cloud/aws/<analysisId>`	Path Parameter: analysisId Request Body Parameter: roleArnName roleExternalId	message status	This resource operation is used to update AWS account's IAM role ARN and External ID for the next scheduled analysis.

Operation	HTTP Method	Input	Output	Description
				Specify the updated roleArnName and roleExternalId in the request body. Example: Updating AWS Credentials
Modify the policy	`PUT /analysis/cloud/aws/<analysisId>`	Path Parameter: analysisId Request Body Parameter: policyInstanceId	message status	This resource operation, with a policyInstanceId in the request body, is used to update the policy used in the next scheduled AWS Analysis. Example: Updating AWS Policy The policy used for the analysis is listed in the output with the recommendation results. If you change the policy while the analysis is in progress, the change will not affect the current analysis execution and will be used for the next analysis refresh.
Update AWS credentials and policy	`PUT /analysis/cloud/aws/<analysisId>`	Path Parameter: analysisId Request Body Parameter: roleArnName roleExternalId policyInstanceId	message status	This resource operation is used to update AWS account's IAM role ARN, External ID, and the policy used in the next

Operation	HTTP Method	Input	Output	Description
				scheduled Analysis. Example: Updating AWS Credentials and Policy

Parameters

Path Parameters

Table: AWS Analysis Path Parameters

Parameter Name	Type	Description
analysisId	string	The unique referenced ID of the AWS analysis.

Request Body Parameters

Table: AWS Analysis Request Body Parameters

Parameter Name	Type	Description
accountId	string	The AWS account ID with the CloudWatch data to collect. See *AWS Data Collection Prerequisites for an IAM Role* (Help Topic ID 410060) for details on setting up the ARN.
connectionName *(optional)*	string	Use the connection name to clearly identify this connection within Densify. This name will appear in the **Saved Connections** list in theDensify UI. By default, the connection name is set to the Subscription ID. The connection name must be unique within the Azure connection type section, so if the name is already in use, the request fails with an error message. This connection name can be used for filtering. ***Note***:*The Connection Name is limited to 32-characters.*
roleArnName	string	The Amazon Resource Name (ARN) for the IAM role that you created in AWS to collect data.

Parameter Name	Type	Description
		See *AWS Data Collection Prerequisites for an IAM Role* (Help Topic ID 410060) for details on setting up the ARN. To update the ARN, refer to Update AWS credentials.
roleExternalId	string	The external ID specified for Densify when the IAM role was created. See *AWS Data Collection Prerequisites for an IAM Role* (Help Topic ID 410060) for details on setting up the ARN. To update the external ID, refer to Update AWS credentials.
policyInstanceId *(optional)*	string	The cloud policy used for optimization analysis. This parameter is optional and used to override the default policy for AWS cloud analysis. See GET /analysis/cloud/aws/policy on page 201.
triggerAdhocAudit *(optional)*	string	The flag to trigger an ad-hoc 60-day historical audit. This parameter is optional and used to run the 60-day historical audit immediately and once only. Typically the 60-day historical audit is run first, when data collection is initiated for the specified AWS account. If this audit has been disabled for performance reasons, it can be run to collect the historical data as compute resources are available. A connection to the specified AWS account must already exist before you can use this flag. A once-off task will be configured and a new connection will not be created, if the connection does not already exist. If there is no existing connection AND the API post contains ad hoc=true, then you will see an error message.
webHook *(optional)*	- uri - authType - authValue	The webhook definition to an external application. Optimization results are sent to the webhook-defined application when analysis is complete. See Parameters on page 213 for details of each parameter in the webhook definition.

Response

Table: AWS Analysis Response Schema

Element	Type	Filter/Sort	Description
href	string		The referenced resource to the analysis entity. See Analysis: Entity on page 157 for details of the analysis entity resource. When a new analysis is requested from the `/analyze` resource, the entity ID will not be available until after data collection completes and the analysis entity is created.
message	string		The message for the status response is returned.
status	number		The HTTP response code of the request. Possible status values include: ▪ 200—success with request; ▪ 400—invalid parameters; ▪ 401—authentication failed; ▪ 404—resource not found; ▪ 500—internal server error.

Examples

Example: Running AWS Data Collection and Analysis

The following example shows you how to initiate AWS data collection and analysis, and send the results to a WebHook.

Example: Running AWS Data Collection and Analysis

Request:

```
POST /analysis/cloud/aws/analyze
{
  "accountId": "188665225420",
  "roleArnName": "arn:aws:iam::188665225420:role/ReadOnlyAccess",
  "roleExternalId": "password",
  "policyInstanceId": "4a63f651-a583-4157-97ff-35651370ffbe",
  "webHook": {
       "uri": "http://mywebhookserver/webhook/results",
       "authType": "basic",
       "authValue": "tester:testerpassword"
  }
}
```

Response:

```
{
  "href": "Not available",
  "message": "OK",
  "status": 200
}
```

Example: Updating AWS Credentials

The following example shows you how to update your AWS account's IAM role ARN and External ID.

Example: Updating AWS Credentials

Request:

```
PUT /analysis/cloud/aws/9a5d2d55-6d85-4fde-8bab-fcd0cef8c5bf
{
    "roleArnName": "arn:aws:iam::506226932793:role/my_new_readonly",
    "roleExternalId": "NewPassword"
}
```

Response:

```
{
  "message": "ok",
  "status": 200
}
```

Example: Updating AWS Policy

The following example shows you how to update the policy used in your AWS analysis. The new policy will be used in the next scheduled analysis. To obtain the policy instance ID (i.e. policyInstanceId), refer to the Analysis: Policy on page 199 resource.

Example: Updating AWS Policy

Request:

```
PUT /analysis/cloud/aws/9a5d2d55-6d85-4fde-8bab-fcd0cef8c5bf
{
    "policyInstanceId" : "0c0ef18b-9367-4071-b733-396f63e51925"
}
```

Response:

```
{
  "message": "ok",
  "status": 200
}
```

Example: Updating AWS Credentials and Policy

The following example shows you how to update both the AWS credentials and policy in your analysis, at the same time. The new credentials and policy will be used in the next scheduled analysis.

Example: Updating AWS Credentials and Policy

Request:

```
PUT /analysis/cloud/aws/9a5d2d55-6d85-4fde-8bab-fcd0cef8c5bf
{
    "roleArnName": "arn:aws:iam::506226932793:role/my_new_readonly",
    "roleExternalId": "NewPassword",
    "policyInstanceId" : "0c0ef18b-9367-4071-b733-396f63e51925"
}
```

Response:

```
{
  "message": "ok",
  "status": 200
}
```

Example: Running the 60-Day Historical Audit

The following example shows you how to run the initial 60-day historical data collection.

Example: Running 60-Day Historical Audit

Request:

```
POST /analysis/cloud/aws/analyze
{
  "accountId": "188665225420",
  "roleArnName": "arn:aws:iam::188665225420:role/ReadOnlyAccess",
  "roleExternalId": "password",
  "triggerAdhocAudit": "Yes",
  "startDayOffset":"20" ,
  "endDayOffset":"10"
}
```

Response:

```
{
  "href": "/analysis/aws/209726931496/status",
  "message": "Analysis in progress",
  "status": 200
}
```

Analysis: AWS Recommendations

Description

The `/analysis/cloud/aws/<analysisId>/results` resource is used to return a collection of AWS system (i.e. EC2 instance, Auto Scaling group, etc.) recommendations for an AWS account defined by a Densify Analysis entity (see *Analysis: Entity* (Help Topic ID 340610)).

For each system recommendation, you can also download a PDF version of the Impact Analysis and Recommendation Report, which details system impact based on the recommendations. This report is useful to application owners and other system stakeholders who need to review recommended changes to their application resources. See the rptHref resource element and *Viewing the Impact Analysis and Recommendation Report* (Help Topic ID 380450) or *Viewing the ASG Impact Analysis and Recommendation Report* (Help Topic ID 380880) for details on the content of the report.

To return a collection of all systems included in an AWS optimization analysis, see Analysis: AWS Systems on page 129.

Resource

```
/analysis/cloud/aws/<analysisId>/results
```

Supported Operations

Table: AWS Recommendations Supported Operations

HTTP Method	Input	Output	Description
`GET /analysis/cloud/aws/<analysisId>/results`	Path Parameter: - analysisId Query String Parameter Options: - Element Filters - includeAttributes - dataQuality Accept: - application/json	Collection of (JSON): - entityId - resourceId - accountIdRef - region - currentType - recommendationType - recommendedType - implementationMethod - predictedUptime - totalHoursRunning - name - rptHref - approvalType - densifyPolicy - savingsEstimate - effortEstimate - powerState - recommendedHostEntityId - currentCost - recommendedCost - serviceType - currentHourlyRate - recommendedHourlyRate - currentRiCoverage - minGroupCurrent - minGroupRecommended - maxGroupCurrent - maxGroupRecommended - currentDesiredCapacity - avgInstanceCountRecommended - avgInstanceCountCurrent - deferRecommendation - deferUntil - attributes - recommFirstSeen - recommLastSeen - recommSeenCount - auditInfo - dataQuality	Returns a collection of recommendations for the specified analysis. Specify `application/json` in the request header for recommendations to be returned in JSON format. **Note**: *The returned recommendations can be in either JSON or Terraform-map format.* Example: Returning EC2 Instances with Upsize Recommendations

HTTP Method	Input	Output	Description
`GET /analysis/cloud/aws/<analysisId>/results`	Path Parameter: analysisId Query String Parameter Options: Element Filters Accept: application/terraform-map	Collection of (Terraform-map): provisioningId (label of each terraform-map recommendation) currentType recommendedType approvalType predictedUptime recommendationType powerState implementationMethod savingsEstimate effortEstimate densifyPolicy deferRecommendation deferUntil minGroupCurrent minGroupRecommended maxGroupCurrent maxGroupRecommended avgInstanceCountCurrent currentDesiredCapacity avgInstanceCountRecommended	Returns a collection of recommendations for an AWS analysis. Specify `application/terraform-map` in the request header for recommendations to be returned in Terraform-map format. **Note**: *The returned recommendations can be in either JSON or Terraform-map format.* Example: Returning AWS Recommendations with Low Effort in Terraform-map Form Example: Returning an ASG Recommendation in Terraform-map Form

Parameters

Path Parameters

Table: AWS Recommendations Path Parameters

Parameter Name	Type	Description
analysisId	string	The unique reference ID of the AWS analysis.

Query String Parameters

Table: AWS Recommendation Query String Parameters

Parameter Name	Type	Description
Element Filters	string	You can use element filters to return a targeted subset of the

Parameter Name	Type	Description
		recommendations. See the "F" (filter) designation in the Response schema table for a list of elements that support filtering. Refer to Filters on page 30 for a complete description of this common operation feature. Usage example: `../results?recommendationType=Upsize`
includeAttributes	- true - false	Indicate whether or not to return system attributes: - true—returns all of the system attributes; - false—(default) suppress system attributes from the response output. Usage example: `../results?includeAttributes=true`
dataQuality	string	Allows you to indicate which workload type data collection detail to return for the system. See dataQuality for the returned details. Specify a quoted list of workload type names for which data to be returned, separated by commas. The supported workload type names can be found in the Data Center Explorer (DCE) from the Analysis Console. See DCE Virtual Environment Workload Viewer to find a list of the supported workload types from the Analysis Console. Usage example: `../results?dataQuality="CPU Utilization (CINT2006 Rate),Memory Utilization in Percent"` ***Note**: A "400-Bad Request" error message is returned if a non-supported workload type is specified.*

Response

Table: AWS Recommendation Response Schema

Element	Type	Filter/Sort	Description
entityId	string	F	The Densify assigned entity ID of the cloud system.
resourceId	string	F	The AWS identifier assigned to the system.
accountIdRef	string	F	The AWS account identifier.
region	string	F	The region of the AWS system.
currentType	string		The current instance type of the

Element	Type	Filter/Sort	Description
			AWS system.
recommendationType	string	F	The recommended action for the system. This is also known as the Optimization Type in the Densify Console (see *Optimization Type Color-Coding Summary* in the topic *Understanding the Instance Optimization Details Report* (Help Topic ID 380390)). The following types of recommended actions are supported for this cloud platform: - `"Just Right"`—this instance is optimally sized for the workload; - `"Upsize - Optimal Family"`—this instance should be upsized to a more optimal instance family; - `"Upsize"`—this instance should be upsized to an instance within the same instance family; - `"Terminate"`—this instance should be terminated; - `"Downsize - Optimal Family"`—this instance should be downsized to an instance belonging to a more suited instance family; - `"Downsize"`—this instance should be downsized to an instance within the same instance family; - `"Modernize - Optimal Family"`—this instance should be modernized to an instance belonging to a more optimal instance family; - `"Modernize"`—this instance should be modernized to an instance within the same instance family. - `"Not Analyzed"`—this instance has no recommendation due to insufficient workload

Element	Type	Filter/Sort	Description
			information. For ASGs, additional recommendation actions are supported: - `Upscale`—increase compute capacity by adjusting the maximum group size; - `Downscale`—decrease compute capacity by adjusting the minimum group size. Use *recommendationType* as a filter to retrieve only system actions of interest. See Example: Returning AWS Systems with No Recommendations.
recommendedType	string	F	The recommended instance type after Densify optimization analysis.
implementationMethod	string	F	[Self Optimization|Manual|N/A] Specifies whether this system is configured for Self-Optimization or Manual actioning based on the recommended action (recommendationType) and on the Self-Optimizing Automation policies. ***Note***: *This element is not returned for ASGs with a maximum group size greater than one. Self-Optimization is not supported for ASGs with more than one EC2 instance.*
predictedUptime	percentage		The predicted uptime (%) for the system is based on the percentage of hours CPU utilization data is present in the workload range specified in the policy settings. Predicted uptime % for new systems started mid-way within the workload range is calculated from the time/date that the system was started, as opposed to the beginning of the interval resulting, in more accurate prediction for the future.

Element	Type	Filter/Sort	Description
			***Note**: This element is not returned for ASGs with maximum group size greater than one. For ASGs with the maximum group size=1, the EC2's predicted uptime is returned.*
totalHoursRunning	string		The total hours that the system has been running, in the workload range. Total hours for an ASG is the total time the ASG has been running and does not include cumulative hours for all of the in-service instances.
totalHours	string		The total hours since the instance was created, that are within the workload range. Total hours for an ASG is the total time since the ASG was created and does not include cumulative hours for all of the in-service instances. If the system is created in the middle of the workload range, then Densify uses the creation date to calculate total hours.
name	string	F	The name of the AWS system. Typically, this is the "Provisioning Id" AWS user tag value. If the "Provisioning Id" value is not set, then the AWS name assigned to the system is used instead.
rptHref	string		The reference resource to the Impact Analysis and Recommendation Report (also known as the Application Owner report). See *Viewing the Impact Analysis and Recommendation Report* (Help Topic ID 380450) for details on the content of the report. You need to use the following in the request header to download the PDF file: `Accept : application/octet-stream.`

Element	Type	Filter/Sort	Description
			Example: Downloading an Impact Analysis and Recommendation Report
approvalType	string	F	The approval setting for the system recommendation. The value in this string is derived from the Self-Optimizing Automation policies in implementationMethod and the approval attribute `attr_ApprovalSetting`. Possible approval settings include: - "`na`"— not approved; - "`all`"—approve any change; - "*`<recommended-instance-type>`*"—approve changing the instance to the specified *<recommended-instance-type>*. This field is not applicable to Auto Scaling groups with maximum group size greater than one. For these systems, the approvalType will always be "na" (not approved).
densifyPolicy	string	F	The Densify policy used for optimization analysis.
savingsEstimate	string		The value of savingestimate output parameter is the difference between the current and recommended instance type cost (this is the catalog cost). When using the API, the predicted uptime is NOT taken into consideration (i.e. [currentCost – recommendedCost]). The Impact Analysis and Recommendation Report report uses the predicted uptime % when calculating estimated savings regardless of whether the report is obtained through the UI or via API. See *FAQs-Cloud* (Help Topic ID 400000) for examples. This calculation also applies to ASGs with maximum group size = 1. In this case, the ASG is considered an EC2,

Element	Type	Filter/Sort	Description
			with no ASG elements exposed. For ASGs with maximum group size greater than one, the savings estimate is the difference between the current and recommended instance type cost (i.e. catalog cost), with the average group in-service instances taken into consideration (i.e. [avgInstanceCountCurrent * currentCost] - [avgInstanceCountRecommended * recommendedCost]).
effortEstimate	string	F	[Moderate|Low|Very Low|None|Impossible] This element describes the effort required to investigate and implement the Densify recommendations. Effort for each system is calculated by rule-driven analytics based on factors (such as instance family change, storage change, data quality checks, feature changes, etc.) that can be configured in the policy settings and rule sets which capture best practices. "Impossible" effort is a result of a manual override for the instance. If a system is not ‘Not Analyzed’, Densify does not return an effortEstimate. ***Note**: When using the Subscription API, Densify returns an effortEstimate of "impossible" for systems that are ‘Not Analyzed’.*
powerState	string	F	The power state of the system.
recommendedHostEntityId	string	F	The Densify entity ID of the catalog instance for the recommended instance type.
currentCost	string		The cost of the existing instance type (i.e. the instance catalog cost).
recommendedCost	string		The cost of the recommended

Element	Type	Filter/Sort	Description
			instance type (i.e. the instance catalog cost) after Densify optimization analysis.
serviceType	string	F	The AWS service type: - EC2 - RDS - ASG - SPOT ***Note**: ASGs with maximum group size of 1 (i.e. maxGroupCurrent=1) will have serviceType=ASG. However, the returned elements are similar to those for an EC2 service (i.e. min/max group size values are not returned).*
currentHourlyRate	string		The hourly rate for the current instance type (i.e. instance catalog cost / monthly hours). This value is rounded to the nearest penny.
recommendedHourlyRate	string		The hourly rate for the recommended instance type (i.e. instance catalog cost / monthly hours). This value is rounded to the nearest penny.
currentRiCoverage	integer		This is the percentage of reserved instance (RI) coverage available for the current instance (or scale group). Flexible RIs are taken into consideration when calculating the percentage of RI coverage. However, convertible RIs are included in the calculation depending on the AWS Defer Recommendation settings. For individual compute or database instances, the coverage percentage is either 100% or 0%. Coverage between 0 and 100% occurs when Flexible RIs are converted into partial coverage. For scale groups, the percentage of coverage is based on the RIs available for in-service instances.

Element	Type	Filter/Sort	Description
			Note: *This element is returned for serviceType = EC2, RDS, and ASG.*
deferRecommendation	string		[yes|no] Defer recommending an instance type change until reserved instance coverage for this instance expires. Values include: ▪ yes—defer instance type recommendation due to RI coverage; ▪ no—do not defer instance type recommendation. If "*AWS - Defer Recommendation - Exclude Upsize Recommendation*" setting is TRUE (from Densify configuration settings), then deferRecommendation = no, for Upsize recommendations, even if currentRiCoverage is 100%. **Note**: *This element is returned for serviceType = EC2, RDS, and ASG. In addition, this element is not returned when recommendationType is " Just Right" or "Terminate" .*
deferUntil	string		If deferRecommendation = "yes", then the expiry date of the RI providing coverage for the current instance is returned. The expiry date and time is in milliseconds. **Note**: *This element is only returned when deferRecommendation = "yes".*
minGroupCurrent	integer		Specifies the current minimum group size of the Auto Scaling group. **Note**: *This element is only returned for ASGs with maximum group size greater than one.* Example: Returning an ASG Recommendation in Terraform-map Form

Element	Type	Filter/Sort	Description
minGroupRecommended	integer		Specifies the recommended minimum group size for the Auto Scaling group. ***Note**: This element is only returned for ASGs with maximum group size greater than one.* Example: Returning an ASG Recommendation in Terraform-map Form
maxGroupCurrent	integer		Specifies the current maximum group size of the Auto Scaling group. ***Note**: This element is only returned for ASGs with maximum group size greater than one.* Example: Returning an ASG Recommendation in Terraform-map Form
maxGroupRecommended	integer		Specifies the recommended maximum group size for the Auto Scaling group. ***Note**: This element is only returned for ASGs with maximum group size greater than one.*
currentDesiredCapacity	integer		Specifies the desired capacity currently configured for the Auto Scaling group. ***Note**: This element is only returned for ASGs with maximum group size greater than one.*
avgInstanceCountCurrent	float		The average of in-service instances over the historical interval (or workload range). The this interval is defined by your policy (e.g. over the last 60 days). ***Note**: This element is only returned for ASGs with maximum group size greater than one.*
avgInstanceCountRecommended	float		The predicted average instance count if the ASG recommendations were implemented. ***Note**: This element is only returned for*

Element	Type	Filter/Sort	Description
			ASGs with maximum group size greater than one. Example: Returning an ASG Recommendation in Terraform-map Form
provisioningId	string		This element is used to identify the terraform-map recommendations for a unique system and corresponds to the "Provisioning Id" AWS user tag value. The "Provisioning Id" user tag is used to uniquely identify a system, since its resourceId or system name could possibly change after an instance type update. See Example: Returning AWS Recommendations with Low Effort in Terraform-map Form. If the "Provisioning Id" AWS user tag value is not set, then the system name is used to identify the recommendations. ***Note**: This element is only returned for terraform-map responses.*
attributes	array of • id • name • value		System attributes are properties set during the data collection process by a vendor platform (i.e. AWS) or by Densify for analytics. ***Note**: The attribute array is only returned when includeAttributes=true is in the query string.*
recommFirstSeen	Unix time (in milliseconds)		The first date and time the recommended instance type (i.e. the recommendedType element) was provided by Densify (Unix Epoch time, in milliseconds).
recommLastSeen	Unix time (in milliseconds)		The latest date and time the recommended instance type (i.e. the recommendedType element) was provided by Densify (Unix Epoch time, in milliseconds).
recommSeenCount	integer		The number of times Densify

Element	Type	Filter/Sort	Description
			suggested the recommended instance type (i.e. the recommendedType element). This is the count of Densify analysis processes which produced the same recommended instance type from recommFirstSeen to recommLastSeen. This value is updated when the RDB populate task (i.e. the reporting database update process) is executed. The RDB populate task compares the current `recommendedType` with the new `recommendedType` to update the `recommSeenCount` counter. ***Note**: In typical production environments, where the RDB populate task is scheduled to run once daily post data collection and analysis, the `recommSeenCount` value will reflect exactly the number of times the recommended instance type was provided. If your environment executes the RDB populate task more than once daily, the `recommSeenCount` value will be inflated beyond the actual number of times the instance type was recommended. Contact Support@Densify.com if you have concerns about the `recommSeenCount` value.*
auditInfo	• dataCollection: • dateFirstAudited • dateLastAudited • auditCount • workloadDataLast30: • firstDate • lastDate • totalDays • seenDays		The following system data collection details are returned: • `dateFirstAudited`—the first time data was collected for this system (Unix epoch time, in milliseconds); • `dateLastAudited`—the most recent data collection time (Unix epoch time, in milliseconds); • `auditCount`—the number of

Element	Type	Filter/Sort	Description
			times that data was collected. The following system workload collection details, for the last 30 days, are returned: • `firstDate`—the first time workload data was collected for this system (Unix epoch time, in milliseconds); • `lastDate`—the most recent workload data collection time for this system (Unix epoch time, in milliseconds); • `totalDays`—the difference in days between `firstDate` and `lastDate`; • `seenDays`—the number of days that at least one workload data was added into Densify for this system. ***Note**: If no workload data is collected for the system in the last 30 days, then the workloadDataLast30 element block is not returned.* ***Note**: The values in `auditInfo` are updated once a day, after the data collection and RDB populate processes are complete (i.e. the reporting tables have been updated with latest data collected).*
dataQuality	array of: • workloadName • firstSeen • lastSeen • completeDays • partialDays		The dataQuality array provides workload type data collection details for the system: • `workloadName`—the workload type name (see DCE Virtual Environment Workload Viewer to find a list of the supported workload types from the Analysis Console); • `firstSeen`—the first time this workload was collected (Unix Epoch time, in milliseconds); • `lastSeen`—the most recent time

Element	Type	Filter/Sort	Description
			this workload was collected (Unix Epoch time, in milliseconds); ▪ `completeDays`—the number of complete days that this workload data was collected; ▪ `partialDays`—the number of partial days that this workload data was collected. ***Note**: The dataQuality array is only returned when the dataQuality query string is specified in the request. In addition, if you specify a workload type in the request, for which system workload data does not exist in Densify, then no data is returned.* ***Note**: The values in the dataQuality array are updated after data collection and subsequent updates to the reporting tables (i.e. RDB Populate process) are completed.* See Example: Returning EC2 Instances with Upsize Recommendations.

Examples

Example: Returning EC2 Instances with Upsize Recommendations

The following example shows you how to return a collection of AWS instances with "Upsize" recommendations. In addition, this example also returns the "CPU Utilization", and "Disk I/O Bytes" workload data collection details.

Example: Returning AWS Upsize Recommendations with Workload Data Collection Details

Request:

```
GET /analysis/cloud/aws/7abb627d-48db-4520-9e90-f46946ea6a24/results?recommendationType=Upsize&dataQuality="CPU Utilization,Disk I/O Operations"
```

Response:

```
[
    {
            "entityId": "49fb3629-1f2b-4039-b146-918eb8009184",
            "region": "us-east-1",
            "currentType": "t3.micro",
            "recommendationType": "Upsize",
            "recommendedType": "t3.small",
            "implementationMethod": "Manual",
            "predictedUptime": 92.29,
            "totalHoursRunning": 1329,
            "totalHours": 1440,
            "name": "SQL Express",
            "rptHref": "/systems/49fb3629-1f2b-4039-b146-918eb8009184/analysis-report",
            "approvalType": "na",
            "densifyPolicy": "Initial Assessment",
            "savingsEstimate": -5.4850235,
            "effortEstimate": "Very Low",
            "powerState": "Running",
            "recommendedHostEntityId": "9a67bfc4-7d30-4e75-9c4d-1ca501b0c4fd",
            "currentCost": 14.31,
            "recommendedCost": 28.62,
            "serviceType": "EC2",
            "currentHourlyRate": 0.02,
            "recommendedHourlyRate": 0.04,
            "currentRiCoverage": 0.0,
            "recommFirstSeen": 1579680587657,
            "recommLastSeen": 1589008760293,
            "recommSeenCount": 24,

        "auditInfo": {
            "dataCollection": {
                "dateFirstAudited": 1571949272133,
                "dateLastAudited": 1588997132907,
                "auditCount": 103
            },
            "workloadDataLast30": {
                "firstDate": 1687009600000,
                "lastDate": 1688910400000,
                "totalDays": 23,
                "seenDays": 7
            }
        },
        "dataQuality": [
            {
                "workloadName": "CPU Utilization",
                "firstSeen": 1666705600000,
                "lastSeen": 1688910400000,
                "completeDays": 161,
                "partialDays": 0
            },
            {
                "workloadName": "Disk I/O Operations",
                "firstSeen": 1666705600000,
                "lastSeen": 1673448400000,
                "completeDays": 78,
                "partialDays": 1
            }
        ]
    }
    ...
]
```

Example: Returning ASG Downscale Recommendations

The following example shows you how to return ASG instances with "Downscale" recommendations.

Example: Returning ASG Downscale Recommendations

Request:

```
GET /analysis/cloud/aws/48db7abb627d-7abb-5602-9e9049f-
b36296a37/results?recommendationType=Downscale"
```

Response:

```
[
    {
            "entityId": "8c4719f5-0813-4943-82a8-9183c5994e6d",
            "resourceId": "8c22f5fb-8ed0-4ecd-ac91-8aa8e2f3856a",
            "accountIdRef": "209726931496",
            "region": "us-east-1",
            "currentType": "t2.micro",
            "recommendationType": "Downscale",
            "recommendedType": "t2.micro",
            "predictedUptime": 92.29,
            "totalHoursRunning": 1329,
            "totalHours": 1440,
            "name": "ec2containerservice-cluster-ecs-asg-k1yl0k4m",
            "rptHref": "/systems/8c4719f5-0813-4943-82a8-9183c5994e6d/ana-
lysis-report",
            "approvalType": "na",
            "densifyPolicy": "Initial Assessment",
            "savingsEstimate": 8.469575,
            "effortEstimate": "None",
            "powerState": "Running",
            "recommendedHostEntityId": "56b8437a-2eed-41e6-bd50-4d4734982785",
            "currentCost": 8.47,
            "recommendedCost": 8.47,
            "serviceType": "ASG",
            "currentHourlyRate": 0.01,
            "recommendedHourlyRate": 0.01,
            "currentRiCoverage": 0.0,
            "minGroupCurrent": "0",
            "minGroupRecommended": "0",
            "maxGroupCurrent": "4",
            "maxGroupRecommended": "4",
            "currentDesiredCapacity": "2",
            "avgInstanceCountRecommended": 1.0,
            "avgInstanceCountCurrent": 1.9999498,
            "deferRecommendation": "no",
            "recommFirstSeen": 1683131807753,
            "recommLastSeen": 1686110400000,
            "recommSeenCount": 10,
    }
    ...
]
```

Example: Returning AWS Recommendations with Low Effort in Terraform-map Form

The following example shows you how to return a collection of recommendations with Low effort in terraform-map form. The label of each recommendation (i.e. "asop-prepro-rdb-206", "ea-dev-asop-299" in the example below) is the provisioningId element.

Example: Return Low Effort Recommendations in Terraform-map

Request:

```
GET /analysis/cloud/aws/8bef9d74-94f7-414f-a032-5855258473a2/res-
ults?effortEstimate=Low
```

Headers:

```
Accept: application/terraform-map
Authorization: Bearer <apiToken>
```

Response:

```
densify_recommendations = {
  "asop-prepro-206" = {
    currentType = "t2.micro"
    recommendedType = "t3.micro"
    approvalType = "all"
    predictedUptime = "77.31"
    recommendationType = "Modernize"
    powerState = "Running"
    implementationMethod = "Self Optimization"
    savingsEstimate = "-34.31837"
    effortEstimate = "Low"
    densifyPolicy = "AWS General Prod"
    deferRecommendation = "no"
    minGroupCurrent = "0"
    minGroupRecommended = "0"
    maxGroupCurrent = "3"
    maxGroupRecommended = "3"
    avgInstanceCountCurrent = "0.99846673"
    currentDesiredCapacity = "0"
    avgInstanceCountRecommended = "0.99846673"
  }
  "ea-dev-asop-299" = {
    currentType = "c4.xlarge"
    recommendedType = "c5.xlarge"
    approvalType = "na"
    predictedUptime = "77.62"
    recommendationType = "Downsize"
    powerState = "Running"
    implementationMethod = "Self Optimization"
    savingsEstimate = "16.43216"
    effortEstimate = "Low"
    densifyPolicy = "AWS General Prod"
    deferRecommendation = "no"
  }
...
}
```

Example: Returning an ASG Recommendation in Terraform-map Form

The following example shows you how to return a collection of ASG recommendations in terraform-map form.

Example: Return ASG Recommendations in Terraform-map

Request:

```
GET /analysis/aws/8bef9d74-94f7-414f-a032-5855258473a2/results?serviceType=ASG
```

Headers:

```
Accept: application/terraform-map
Authorization: Bearer <apiToken>
```

Response:

```
densify_recommendations = {
  "mobile-svc-asg-analysisGrid" = {
    currentType = "c4.large"
    recommendedType = "c5.large"
    approvalType = "na"
    predictedUptime = "81.53"
    recommendationType = "Downsize"
    powerState = "Running"
    savingsEstimate = "114.57644"
    effortEstimate = "Low"
    densifyPolicy = "AWS General Prod"
    minGroupCurrent = "4"
    minGroupRecommended = "2"
    maxGroupCurrent = "16"
    maxGroupRecommended = "8"
    avgInstanceCountCurrent = "2.9568965"
    currentDesiredCapacity = "4"
    avgInstanceCountRecommended = "1.6321839"
  }
  "pr000252-pro-bion-asg" = {
    currentType = "t2.xlarge"
    recommendedType = "c5.xlarge"
    approvalType = "na"
    predictedUptime = "67.23"
    recommendationType = "Downsize - Optimal Family"
    powerState = "Running"
    savingsEstimate = "302.53308"
    effortEstimate = "Low"
    densifyPolicy = "AWS General Prod"
    minGroupCurrent = "8"
    minGroupRecommended = "4"
    maxGroupCurrent = "12"
    maxGroupRecommended = "15"
    avgInstanceCountCurrent = "4.0170455"
    currentDesiredCapacity = "8"
    avgInstanceCountRecommended = "1.9479166"
  }
  ...
}
```

Example: Downloading an Impact Analysis and Recommendation Report

The following example shows you how to download a PDF Impact Analysis and Recommendation Report from the `rptHref` resource element provided in the instance recommendation output.

Note: *HTTPS needs to be enabled to download the Impact Analysis and Recommendation Report PDF.*

Example: Download an Impact Analysis and Recommendation Report

Request:

```
GET /systems/7836335a-1942-4115-a65d-a298be1d390c/analysis-report
```

Headers:

```
Accept: application/octet-stream
Authorization: Bearer <apiToken>
```

Example: Returning AWS Systems with No Recommendations

The following example shows you how to return a collection of AWS systems without recommendations. These systems typically do not have adequate data for optimization analysis and have the "Not Analyzed" designation in the recommendationType element.

Example: Returning AWS Systems with No Recommendations in JSON

Request:

```
GET /analysis/cloud/aws/7abb627d-48db-4520-9e90-f46946ea6a24/res-
ults?recommendationType=Not Analyzed
```

Headers:

```
Accept: application/json
Authorization: Bearer <apiToken>
```

Response:

```
[
    {
        "entityId": "88bf8536-7f8e-4494-9930-5873ea982f7a",
        "resourceId": "i-06d390ca1a0c59d95",
        "accountIdRef": "555726931496",
        "region": "us-east-1",
        "currentType": "c4.large",
        "recommendationType": "Not Analyzed",
        "name": "Mark333",
        "densifyPolicy": "Initial Assessment",
        "powerState": "Stopped",
        "currentCost": 79.84,
        "serviceType": "EC2",
        "currentRiCoverage": 0.0,
        "recommFirstSeen": 1579680587673,
        "recommLastSeen": 1589008760337,
        "recommSeenCount": 24,
        "auditInfo": {
            "dataCollection": {
                "dateFirstAudited": 1571949272133,
                "dateLastAudited": 1588997132907,
                "auditCount": 103
            }
        }
    },
    {
        "entityId": "ef37339b-d400-4a8c-a080-44e6bfd8db19",
        "resourceId": "db-pfnzbc3ahhaqtt5gyhi7h4lhbi",
        "accountIdRef": "555726931496",
        "region": "us-east-1",
        "currentType": "db.t2.micro",
        "recommendationType": "Not Analyzed",
        "name": "autoscale-1",
        "densifyPolicy": "Initial Assessment",
        "powerState": "Offline",
        "currentCost": 12.41,
        "serviceType": "RDS",
        "currentRiCoverage": 0.0,
        "recommFirstSeen": 1579680587673,
        "recommLastSeen": 1589008760337,
        "recommSeenCount": 24,
        "auditInfo": {
            "dataCollection": {
                "dateFirstAudited": 1571949272133,
                "dateLastAudited": 1588997132907,
                "auditCount": 103
            },
            "workloadDataLast30": {
                "firstDate": 1587009600000,
                "lastDate": 1588910400000,
                "totalDays": 23,
                "seenDays": 7
            }
        }
    },
    ...
]
```

Example: Returning AWS Systems with No Recommendations in Terraform-Map

Request:

```
GET /analysis/cloud/aws/7abb627d-48db-4520-9e90-f46946ea6a24/results?recommendationType=Not Analyzed
```

Headers:

```
Accept: application/terraform-map
Authorization: Bearer <apiToken>
```

Response:

```
densify_recommendations = {
  "Mark333" = {
    currentType = "c4.large"
    recommendationType = "Not Analyzed"
    powerState = "Stopped"
    densifyPolicy = "Initial Assessment"
  }
  "autoscale-1" = {
    currentType = "db.t2.micro"
    recommendationType = "Not Analyzed"
    powerState = "Offline"
    densifyPolicy = "Initial Assessment"
  }
  ...
}
```

Analysis: AWS Systems

Description

The `/analysis/cloud/aws/<analysisId>/systems` resource is used to return a collection of all systems included in an AWS optimization analysis.

The recommendations from an AWS optimization analysis can be obtained using the `/analysis/cloud/aws/<analysisId>/results` resource. See Analysis: AWS Recommendations on page 105 for details on the AWS recommendations resource.

The number of analyzed systems (i.e. `/aws/<analysisId>/systems` entities) will always be greater than or equal to the number of system recommendations produced (i.e. `/aws/<analysisId>/results` entities), as some systems may not have any recommendations.

Resource

```
/analysis/cloud/aws/<analysisId>/systems
```

Supported Operations

Table: AWS Systems Supported Operations

Operation	HTTP Method	Input	Output	Description
List all systems included in an	`GET /analysis/cloud/aws/`	Path Parameter:	Collection of:	Use this resource to return a list of all

Operation	HTTP Method	Input	Output	Description
AWS analysis	`<analysisId>/systems`	analysisId	resourceId powerState currentType displayName serviceType entityId href	systems included in the AWS analysis. Example: Listing All Systems in an AWS Analysis

Parameters

Path Parameters

Table: AWS System Path Parameters

Parameter Name	Type	Description
analysisId	string	The unique referenced ID of the AWS analysis.

Response

Table: AWS System Response Schema

Element	Type	Filter/Sort	Description
serviceType	string		The cloud service type (e.g. EC2, RDS, ASG, SPOT). **Note**: *Unlike the Densify Console, ASGs with maximum group size of 1 (i.e. maxGroupCurrent=1) will have serviceType=ASG.*
resourceId	string		The AWS identifier assigned to the instance.
powerState	string		The power state of the instance.
currentType	string		The current instance type of the AWS system.
displayName	string		The AWS name assigned to the system.
entityId	string		The Densify assigned entity ID of the AWS system.
href	string		The referenced resource to the system entity. See Systems on page 408 for details of the `/systems` resource.

Examples

Example: Listing All Systems in an AWS Analysis

The following example shows you how to return all systems (instances) included in an AWS optimization analysis.

Example: Listing all Systems in an AWS Analysis

Request:

```
GET /analysis/cloud/aws/8b5d2d56-6d85-4fde-8beb-fcd3cdf8e5b8/systems
```

Response:

```
[
   {
      "serviceType": "EC2",
      "resourceId": "i-bc1037760797bc103",
      "powerState": "Running",
      "currentType": "t2.medium",
      "displayName": "ex-prod-ecds-216",
      "entityId": "008a502d-0a1a-40d4-bb83-42413289fe1e",
      "href": "/systems/008a502d-0a1a-40d4-bb83-42413289fe1e"
   },
   {
      "serviceType": "EC2",
      "resourceId": "i-00bc71015587bc83",
      "powerState": "Running",
      "currentType": "m4.2xlarge",
      "displayName": "ex-dev-abds-131",
      "entityId": "0090c272-7e02-4dd5-b7f3-ad4bc225fbd2",
      "href": "/systems/0090c272-7e02-4dd5-b7f3-ad4bc225fbd2"
   },
   ...
]
```

Analysis: Azure Recommendations

Description

The `/analysis/cloud/azure/<analysisId>/results` resource is used to return a collection of Azure Virtual Machine instance recommendations after the optimization analysis has been performed on your collected Azure infrastructure data.

For each instance recommendation, you can also download a PDF version of the Impact Analysis and Recommendation Report, which details system impact based on the recommendation. This report is useful to application owners who need to review recommended changes to their application resources. See the rptHref resource element for details on how to download this report.

To return a collection of all instances included in an Azure optimization analysis, see Analysis: Azure Systems on page 147.

***Note**: When using the Densify API only one subscription is processed per analysis This is the case, even if more than one subscription is associated with the service principle. If the connection was created through the Cloud Connection wizard in the UI, all subscriptions that were selected, when the connection was created, and associated with the* `<analysisId>`*are returned.*

Resource

```
/analysis/cloud/azure/<analysisId>/results
```

Supported Operations

Table: Azure Recommendations Supported Operations

HTTP Method	Input	Output	Description
`GET /analysis/cloud/azure/<analysisId>/results`	Path Parameter: • analysisId Query String Parameter Options: • Element Filters • includeAttributes • dataQuality Accept: • application/json	Collection of (JSON): • entityId • resourceId> • accountIdRef • currentType • recommendationType • recommendedType • implementationMethod • predictedUptime • Analysis: Azure Recommendations • Analysis: Azure Recommendations • name • rptHref • approvalType • densifyPolicy • savingsEstimate • effortEstimate • powerState • recommendedHostEntityId • currentCost • recommendedCost • serviceType • currentHourlyRate • recommendedHourlyRate • attributes • recommFirstSeen • recommLastSeen • recommSeenCount • auditInfo • dataQuality	Returns a collection of recommendations for an Azure analysis. Specify `application/json` in the request header for recommendations to be returned in JSON format. ***Note**: The returned recommendations can be in either JSON or Terraform-map format.* Example: Returning Azure Instances with Terminate Recommendations
`GET /analysis/cloud/azure/<analysisId>/results`	Path Parameter: • analysisId Query String Parameter Options: • Element Filters	Collection of (Terraform-map): • provisioningId (label of each terraform-map recommendation) • currentType	Returns a collection of recommendations for an Azure analysis. Specify `application/terraform-map` in the request

HTTP Method	Input	Output	Description
	Accept: • application/terraform-map	• recommendedType • approvalType • predictedUptime • recommendationType • powerState • implementationMethod • savingsEstimate • effortEstimate • densifyPolicy	header forrecommendations to be returned in Terraform-map format. ***Note**: The returned recommendations can be in either JSON or Terraform-map format.* Example: Returning Azure Recommendations with Low Effort in Terraform-map Format

Parameters

Path Parameters

Table: Azure Recommendations Path Parameters

Parameter Name	Type	Description
analysisId	string	The unique referenced ID of the Azure analysis.

Query String Parameters

Table: Azure Recommendation Query String Parameters

Parameter Name	Type	Description
Element Filters	string	You can use element filters to return a targeted subset of the recommendations. See the "F" (filter) designation in the Response schema table for a list of elements that support filtering. Refer to Filters on page 30 for a complete description of this common operation feature. Usage example: `../results?recommendationType=Upsize`
includeAttributes	• true • false	Indicate whether or not to return system attributes: • true—returns all the system attributes; • false—(default) suppress system attributes from the response

Parameter Name	Type	Description
		output. Usage example: `../results?includeAttributes=true`
dataQuality	string	Allows you to indicate which workload type data collection detail to return for the system. See dataQuality for the returned details. Specify a quoted list of workload type names for which data to be returned, separated by commas. The supported workload type names can be found in the Data Center Explorer (DCE) from the Analysis Console. See DCE Virtual Environment Workload Viewer to find a list of the supported workload types from the Analysis Console. Usage example: `../results?dataQuality="CPU Utilization (CINT2006 Rate),Memory Utilization in Percent"` ***Note***: *A "400-Bad Request" error message is returned if a non-supported workload type is specified.*

Response

Table: Azure Recommendations Response Schema

Element	Type	Filter/Sort	Description
entityId	string	F	The Densify assigned entity ID of the cloud system.
resourceId	string	F	The Azure identifier assigned to the instance system.
accountIdRef	string	F	The Azure subscription identifier.
currentType	string	F	The current instance type of the Azure system.
recommendationType	string	F	The recommended action for the system. This is also known as the Optimization Type in the Densify Console (see *Optimization Type Color-Coding Summary* in the topic *Understanding the Instance Optimization Details Report* (Help Topic ID 380390)). The following types of recommended actions are supported for this cloud platform:

Element	Type	Filter/Sort	Description
			▪ "Just Right"—this instance is optimally sized for the workload; ▪ "Upsize - Optimal Family"—this instance should be upsized to a more optimal instance family; ▪ "Upsize"—this instance should be upsized to an instance within the same instance family; ▪ "Terminate"—this instance should be terminated; ▪ "Downsize - Optimal Family"—this instance should be downsized to an instance belonging to a more suited instance family; ▪ "Downsize"—this instance should be downsized to an instance within the same instance family; ▪ "Modernize - Optimal Family"—this instance should be modernized to an instance belonging to a more optimal instance family; ▪ "Modernize"—this instance should be modernized to an instance within the same instance family. ▪ "Not Analyzed"—this instance has no recommendation due to insufficient workload information. Systems with insufficient information for analysis do not have a recommendation and are returned with limited elements. See Example: Returning Azure Systems with No Recommendations.
recommendedType	string	F	The recommended instance type after Densify optimization analysis.
implementationMethod	string	F	[Self Optimization|Manual|N/A] Specifies whether this system is configured for Self-Optimization or Manual actioning based on the recommended action (recommendationType) and on the Self-Optimizing Automation policies.
predictedUptime	percentage		The predicted uptime (%) for the system is based on the percentage of hours CPU utilization data is present in the workload range specified in the policy settings.

Element	Type	Filter/Sort	Description
			Predicted uptime % for new systems started mid-way within the workload range is calculated from the time/date that the system was started, as opposed to the beginning of the interval resulting, in more accurate prediction for the future.
name	string	F	The Azure name assigned to the system.
rptHref	string		The reference resource to the Impact Analysis and Recommendation Report (also known as the Application Owner report). The PDF report for the specified instance is available for download after the reporting database tables have been updated (i.e. after *RDB populate* has been executed). By default, the reporting database tables are updated once every night. See *Viewing the Impact Analysis and Recommendation Report* (Help Topic ID 380450) for details on the content of the report. You need to use the following in the request header to download the PDF file: `Accept : application/octet-stream.` Example: Downloading an Impact Analysis and Recommendation Report
approvalType	string	F	The approval setting for the system recommendation. The value in this string is derived from the Self-Optimizing Automation policies in implementationMethod and the approval attribute `attr_ApprovalSetting`. Possible settings include: - "`na`"— not approved; - "`all`"—approve any change; - "*`<recommended-instance-type>`*"—approve changing the instance to the specified *<recommended-instance-type>*.
densifyPolicy	string	F	The Densify policy used for optimization analysis.

Element	Type	Filter/Sort	Description
savingsEstimate	string		The value of savingestimate output parameter is the difference between the current and recommended instance type cost (this is the catalog cost). When using the API, the predicted uptime is NOT taken into consideration (i.e. [currentCost – recommendedCost]). The Impact Analysis and Recommendation Report report uses the predicted uptime % when calculating estimated savings regardless of whether the report is obtained through the UI or via API.
effortEstimate	string	F	[Moderate|Low|Very Low|None|Impossible] This element describes the effort required to investigate and implement the Densify recommendations. Effort for each system is calculated by rule-driven analytics based on factors (such as instance family change, storage change, data quality checks, feature changes, etc.) that can be configured in the policy settings and rule sets which capture best practices. "Impossible" effort is a result of a manual override for the instance. If a system is not ‘Not Analyzed’, Densify does not return an effortEstimate. ***Note**: When using the Subscription API, Densify returns an effortEstimate of "impossible" for systems that are ‘Not Analyzed’.*
powerState	string	F	The power state of the system.
recommendedHostEntityId	string	F	The Densify entity ID of the catalog instance for the recommended instance type.
currentCost	string		The cost of the existing instance type (i.e. the instance catalog cost).
recommendedCost	string		The cost of the recommended instance type (i.e. the instance catalog cost) after Densify optimization analysis.
serviceType	string	F	The Azure cloud service type: ▪ Virtual Machine

Element	Type	Filter/Sort	Description
currentHourlyRate	string		The hourly rate for the current instance type (i.e. instance catalog cost / monthly hours). This value is rounded to the nearest penny.
recommendedHourlyRate	string		The hourly rate for the recommended instance type (i.e. instance catalog cost / monthly hours). This value is rounded to the nearest penny.
attributes	array of • id • name • value		System attributes are properties set during the data collection process by a vendor platform (i.e. Azure) or by Densify for analytics. ***Note**: The attribute array is only returned when includeAttributes=true is added to the query string.*
provisioningId	string		This element is used to identify the terraform-map recommendations for a unique system and corresponds to the "Provisioning Id" Azure user-defined resource tag value. The "Provisioning Id" resource tag is used to uniquely identify a system, since its resourceId or system name could possibly change after an instance type update. See Example: Returning Azure Recommendations with Low Effort in Terraform-map Format If the "Provisioning Id" user-defined resource tag value is not set, then the system name is used to identify the recommendations.
recommFirstSeen	Unix time (in milliseconds)		The first date and time the recommended instance type (i.e. the recommendedType element) was provided by Densify (Unix Epoch time, in milliseconds).
recommLastSeen	Unix time (in milliseconds)		The latest date and time the recommended instance type (i.e. the recommendedType element) was provided by Densify (Unix Epoch time, in milliseconds).
recommSeenCount	integer		The number of times Densify suggested the recommended instance type (i.e. the recommendedType element). This is the count of Densify analysis processes which produced the same recommended instance

Element	Type	Filter/Sort	Description
			type from recommFirstSeen to recommLastSeen. This value is updated when the RDB populate task (i.e. the reporting database update process) is executed. The RDB populate task compares the current `recommendedType` with the new `recommendedType` to update the `recommSeenCount` counter. ***Note**: In typical production environments, where the RDB populate task is scheduled to run once daily post data collection and analysis, the `recommSeenCount` value will reflect exactly the number of times the recommended instance type was provided. If your environment executes the RDB populate task more than once daily, the `recommSeenCount` value will be inflated beyond the actual number of times the instance type was recommended. Contact Support@Densify.com if you have concerns about the `recommSeenCount` value.*
auditInfo	• dataCollection: • dateFirstAudited • dateLastAudited • auditCount • workloadDataLast30: • firstDate • lastDate • totalDays • seenDays		The following system data collection details are returned: • `dateFirstAudited`—the first time data was collected for this system (Unix epoch time, in milliseconds); • `dateLastAudited`—the most recent data collection time (Unix epoch time, in milliseconds); • `auditCount`—the number of times that data was collected. The following system workload collection details, for the last 30 days, are returned: • `firstDate`—the first time workload data was collected for this system (Unix epoch time, in milliseconds); • `lastDate`—the most recent workload data collection time for this system (Unix epoch time, in milliseconds); • `totalDays`—the difference in days

Element	Type	Filter/Sort	Description
			between `firstDate` and `lastDate`; ▪ `seenDays`—the number of days that at least one workload data was added into Densify for this system. ***Note**: If no workload data is collected for the system in the last 30 days, then the workloadDataLast30 element block is not returned.* ***Note**: The values in `auditInfo` are updated once a day, after the data collection and RDB populate processes are complete (i.e. the reporting tables have been updated with latest data collected).*
dataQuality	array of: ▪ workloadName ▪ firstSeen ▪ lastSeen ▪ completeDays ▪ partialDays		The dataQuality array provides workload type data collection details for the system: ▪ `workloadName`—the workload type name (see DCE Virtual Environment Workload Viewer to find a list of the supported workload types from the Analysis Console); ▪ `firstSeen`—the first time this workload was collected (Unix Epoch time, in milliseconds); ▪ `lastSeen`—the most recent time this workload was collected (Unix Epoch time, in milliseconds); ▪ `completeDays`—the number of complete days that this workload data was collected; ▪ `partialDays`—the number of partial days that this workload data was collected. ***Note**: The dataQuality array is only returned when the dataQuality query string is specified in the request. In addition, if you specify a workload type in the request, for which system workload data does not exist in Densify, then no data is returned.* ***Note**: The values in the dataQuality array are updated after data collection and subsequent updates to the reporting*

Element	Type	Filter/Sort	Description
			tables (i.e. RDB Populate process) are completed.

Examples

Example: Returning Azure Instances with Terminate Recommendations

The following example shows you how to return a collection of Azure instances with "Terminate" recommendations.

Example: Returning Azure Terminate Recommendations

Request:

```
GET /analysis/cloud/azure/b585963b-2f05-62a4-b146-1facd95a8f0d/res-
ults?recommendationType=Terminate
```

Response:

```
[
   {
        "entityId": "0d2f5815-1dbe-439c-83e3-788180e41fb2",
        "resourceId": "952d21d6-9ba2-4217-8df1-f5f6417ded45",
        "accountIdRef": "3d4ba999-cbd8-40b8-9998-574be6824a97",
        "currentType": "standard_d2",
        "recommendationType": "Terminate",
        "recommendedType": "idle.guest",
        "implementationMethod": "N/A",
        "predictedUptime": 18.33,
        "name": "st01-dev-est-125",
        "rptHref": "/systems/0d2f5815-1dbe-439c-83e3-788180e41fb2/analysis-
report",
        "approvalType": "na",
        "densifyPolicy": "Initial Assessment",
        "savingsEstimate": 95.107315,
        "effortEstimate": "Low",
        "powerState": "Running",
        "recommendedHostEntityId": "e4175f48-cf28-459a-b573-3da48948a82e",
        "currentCost": 112.42,
        "recommendedCost": 0.0,
        "serviceType": "Virtual Machine",
        "currentHourlyRate": 0.15,
        "recommendedHourlyRate": 0.00,
        "recommFirstSeen": 1579680587640,
        "recommLastSeen": 1589008760283,
        "recommSeenCount": 24,
        "auditInfo": {
            "dataCollection": {
                "dateFirstAudited": 1571949479093,
                "dateLastAudited": 1588997132800,
                "auditCount": 104
            },
            "workloadDataLast30": {
                "firstDate": 1587009600000,
                "lastDate": 1588910400000,
                "totalDays": 23,
                "seenDays": 6
            }
        }
    }
    ...
]
```

Example: Returning Azure Recommendations with Low Effort in Terraform-map Format

The following example shows you how to return a collection of recommendations with Low effort in terraform-map form. The label of each recommendation (i.e. "st01-dev-west-108", "st01-dev-west-110" in the example below) is the provisioningId element: "Provisioning Id" Azure user-defined resource tag or system name value.

Example: Return Low Recommendations in Terraform-map

Request:

```
GET /analysis/cloud/azure/b585963b-2f05-62a4-b146-1facd95a8f0d/res-
ults?effortEstimate=Low
```

Headers:

```
Accept: application/terraform-map
Authorization: Bearer <apiToken>
```

Response:

```
densify_recommendations = {
  "st01-dev-west-108" = {
    currentType = "standard_d2"
    recommendedType = "standard_b2ms"
    approvalType = "na"
    predictedUptime = "93.56"
    recommendationType = "Modernize - Optimal Family"
    powerState = "Running"
    implementationMethod = "Self Optimization"
    savingsEstimate = "128.94438"
    effortEstimate = "Low"
    densifyPolicy = "Azure Assessement"
  }
  "st01-dev-west-110" = {
    currentType = "standard_a1_v2"
    recommendedType = "standard_b1ms"
    approvalType = "na"
    ...
  }
  ...
}
```

Example: Downloading an Impact Analysis and Recommendation Report

The following example shows you how to download a PDF Impact Analysis and Recommendation Report from the `rptHref` resource element provided in the instance recommendation output.

***Note:** HTTPS needs to be enabled to download the Impact Analysis and Recommendation Report PDF.*

Example: Download an Impact Analysis and Recommendation Report

Request:

```
GET /systems/97cfcb18-37a4-4f49-885b-7114eb2ceb30/analysis-report
```

Headers:

```
Accept: application/octet-stream
Authorization: Bearer <apiToken>
```

Example: Returning Azure Systems with No Recommendations

The following example shows you how to return a collection of Azure systems without recommendations. These systems typically do not have adequate data for optimization analysis and have the "Not Analyzed" designation in the recommendationType element.

Example: Returning Azure Systems with No Recommendations in JSON

Request:

```
GET /analysis/cloud/azure/6b8ab8e1-a026-4db9-a4ae-2d684446731f/res-
ults?recommendationType=Not Analyzed
```

Headers:

```
Accept: application/json
Authorization: Bearer <apiToken>
```

Response:

```
[
    {
        "entityId": "1123ea6b-0bdb-4382-9daa-5a597b9f2db4",
        "resourceId": "7a5316a0-4063-4123-a036-3625fea91033",
        "accountIdRef": "3d4ba999-cbd8-40b8-9998-574be6824a88",
        "currentType": "standard_ds3_v2",
        "recommendationType": "Not Analyzed",
        "name": "st08-est-catf-619",
        "densifyPolicy": "Azure Assess",
        "powerState": "Offline",
        "currentCost": 13.14,
        "serviceType": "Virtual Machine",
        "recommFirstSeen": 1579680587623,
        "recommLastSeen": 1589008760263,
        "recommSeenCount": 24,
        "auditInfo": {
            "dataCollection": {
                "dateFirstAudited": 1571949479093,
                "dateLastAudited": 1588997132800,
                "auditCount": 104
            }
        }
    },
    ...
]
```

Example: Returning Azure Systems with No Recommendations in Terraform-Map

Request:

```
GET /analysis/cloud/azure/6b8ab8e1-a026-4db9-a4ae-2d684446731f/results?recommendationType=Not Analyzed
```

Headers:

```
Accept: application/terraform-map
Authorization: Bearer <apiToken>
```

Response:

```
densify_recommendations = {
  "st08-est-catf-619" = {
    currentType = "standard_ds3_v2"
    recommendationType = "Not Analyzed"
    powerState = "Offline"
    densifyPolicy = "Azure Assess"
  }
  ...
}
```

Analysis: Azure Systems

Description

The `/analysis/cloud/azure/<analysisId>/systems` resource is used to return a collection of all the systems (Virtual Machines) that were included in the Azure optimization analysis.

The recommendations from an Azure optimization analysis can be obtained using the `/analysis/cloud/azure/<analysisId>/results` resource. See Analysis: Azure Recommendations on page 132 for details on the Azure recommendations resource. The set of analyzed systems in an analysis may be more than the set of recommendations produced, as there can be no recommendations for some analyzed systems.

Resource

```
/analysis/cloud/azure/<analysisId>/systems
```

Supported Operations

Table: Azure Systems Supported Operations

Operation	HTTP Method	Input	Output	Description
List all systems included in an Azure analysis	`GET /analysis/cloud/azure/<analysisId>/systems`	Path Parameter: analysisId	Collection of: serviceType resourceId	Use this resource to return a list of all systems included in

Operation	HTTP Method	Input	Output	Description
			powerState currentType displayName entityId href	the Azure analysis. Example: Listing All Systems in an Azure Analysis

Parameters

Path Parameters

Table: Azure System Path Parameters

Parameter Name	Type	Description
analysisId	string	The unique referenced ID of the Azure analysis.

Response

Table: Azure System Response Schema

Element	Type	Filter/Sort	Description
serviceType	string		The cloud service type (i.e. Virtual Machine).
resourceId	string		The Azure unique identifier assigned to the instance (Virtual Machine).
powerState	string		The power state of the instance.
currentType	string		The current instance type of the Azure instance.
displayName	string		The name assigned to the instance. For Azure Virtual Machines, the displayName is the same as the unique identifier in resourceId.
entityId	string		The Densify assigned entity ID of the Azure system.
href	string		The referenced resource to the system entity. See Systems on page 408 for details of the `/systems` resource.

Examples

Example: Listing All Systems in an Azure Analysis

The following example shows you how to return all systems (Virtual Machine instances) included in an Azure optimization analysis.

Example: Listing all Systems in an Azure Analysis

Request:

```
GET /analysis/cloud/azure/8b8ab8e1-a026-4db9-a4ae-2d6878467e8f/systems
```

Response:

```
[
   {
      "serviceType": "Virtual Machine",
      "resourceId": "st02-db-edge-695",
      "powerState": "Running",
      "currentType": "standard_a2_v2",
      "displayName": "st02-db-edge-695",
      "entityId": "0137f635-eacd-44da-878f-a4482166f9c1",
      "href": "/systems/0137f635-eacd-44da-878f-a4482166f9c1"
   },
   {
      "serviceType": "Virtual Machine",
      "resourceId": "st01-io-asop-216",
      "powerState": "Running",
      "currentType": "standard_d1_v2",
      "displayName": "st01-io-asop-216",
      "entityId": "065c44d5-040a-4392-9d08-5f581bedb401",
      "href": "/systems/065c44d5-040a-4392-9d08-5f581bedb401"
   },
   ...
]
```

Analysis: Cloud

Description

The `/analysis/cloud` resource is used to return a list of all cloud optimization analysis entities currently in Densify. An analysis entity encompasses cloud resources within a predefined environment scope, specific to the cloud vendor. Typically, the scope of a cloud analysis entity corresponds to the resources within an account, subscription, or project discovered through Densify data collection.

Resource

```
/analysis/cloud
```

Supported Operations

Table: Cloud Analysis Supported Operations

Operation	HTTP Method	Input	Output	Description
List all cloud analyses in Densify	`GET /analysis/cloud`		Collection of: analysisId analysisName analysisCompletedOn href analysisResults analysisStatus	Use this resource to return a list of existing cloud analyses in Densify. Example: Listing All Cloud Analyses

Response

Table:Cloud Analysis Response Schema

Element	Type	Filter/Sort	Description
analysisId	string		The unique identifier for the cloud analysis entity in Densify.
analysisName	string	Filter by: ▪ analysisName ▪ analysisName_like	The name given to the cloud analysis entity. The analysis name typically corresponds to the account ID, subcription ID, or project ID from the infrastructure data collected. Use the `analysisName` and `analysisName_like` filter to return only cloud analyses you are interested in. For example, `/cloud/aws?analysisName_like=bob` will return all AWS analyses with a "bob" substring in the analysis name.
analysisCompletedOn	string		The date and time (in milliseconds) when the last analysis completed. If an analysis has never been completed, "0" is returned.
href	string		The referenced resource to the analysis entity. See Analysis: Entity on page 157 for details of the analysis entity resource.
analysisResults	string		The referenced resource to the recommendations of the analysis. For details of the various cloud analysis recommendations, see: ▪ Analysis: AWS Recommendations on page 105 ▪ Analysis: Azure Recommendations on page 132 ▪ Analysis: GCP Recommendations on page 165
analysisStatus	string		The referenced resource to the status of the analysis. See Analysis: Status on page 206 for details.
message	string		For errors, the message for the status response is returned.

Element	Type	Filter/Sort	Description

status	number		The HTTP response code of the request error.
			Possible status values include:
			200—success with request;
			400—invalid parameters;
			401—authentication failed;
			404—resource not found;
			500—internal server error.

Examples

Example: Listing All Cloud Analyses

The following example shows you how to list all cloud analyses in Densify.

Example: Listing all Cloud Analyses

Request:

```
GET /analysis/cloud
```

Response:

```
[
   {
      "analysisId": "9a5d2d55-6d85-4fde-8bab-fcd0cef8c5bf",
      "analysisName": "624756828528",
      "analysisCompletedOn": 1548083469463,
      "href": "/analysis/cloud/aws/9a5d2d55-6d85-4fde-8bab-fcd0cef8c5bf",
      "analysisResults": "/analysis/cloud/aws/9a5d2d55-6d85-4fde-8bab-fcd0ce-
f8c5bf/results",
      "analysisStatus": "/analysis/cloud/aws/9a5d2d55-6d85-4fde-8bab-fcd0ce-
f8c5bf/status"
   },
   {
      "analysisId": "0930c31d-13ac-4a3f-892e-2a35c8a3f842",
      "analysisName": "DEN-prod-OPS-45",
      "analysisCompletedOn": 1548083655897,
      "href": "/analysis/cloud/gcp/0930c31d-13ac-4a3f-892e-2a35c8a3f842",
      "analysisResults": "/analysis/cloud/gcp/0930c31d-13ac-4a3f-892e-
2a35c8a3f842/results",
      "analysisStatus": "/analysis/cloud/gcp/0930c31d-13ac-4a3f-892e-
2a35c8a3f842/status"
   },
   {
      "analysisId": "6b8ab8e1-a026-4db9-a4ae-2d684446731f",
      "analysisName": "12d89cbc-bc00-4d00-bcf6-ce6ec08d45bd",
      "analysisCompletedOn": 1548340596930,
      "href": "/analysis/cloud/azure/6b8ab8e1-a026-4db9-a4ae-2d684446731f",
      "analysisResults": ""/analysis/cloud/azure/6b8ab8e1-a026-4db9-a4ae-
2d684446731f/results",
      "analysisStatus":"/analysis/cloud/azure/6b8ab8e1-a026-4db9-a4ae-
2d684446731f/status"
   }
]
```

Analysis: Containers

Description

The `/analysis/containers` resource is used to return a list of all optimized container-based analysis entities currently in your Densify system. A container analysis entity encompasses container resources within a predefined environment scope. Typically, the scope of a container analysis entity corresponds to the container resources within a cluster discovered through Densify data collection.

See *Container Prerequisites* (Help Topic ID 410140) for container data collection details.

Resource

```
/analysis/containers
```

Supported Operations

Table: Container Analysis Supported Operations

Operation	HTTP Method	Input	Output	Description
List all container analyses in Densify	`GET /analysis/containers`		Collection of: analysisId analysisName analysisCompletedOn href analysisResults	Use this resource to return a list of existing container analyses. Example: Listing All Container Analyses

Operation	HTTP Method	Input	Output	Description
			- analysisStatus	

Response

Table: Container Analysis Response Schema

Element	Type	Filter/Sort	Description
analysisId	string		The unique identifier for the container analysis entity in Densify.
analysisName	string	Filter by: - analysisName - analysisName_like	The name given to the container analysis entity. The analysis name typically corresponds to the cluster name from the container data collected.
analysisCompletedOn	string		The date and time (in milliseconds) when the last analysis completed. If an analysis has never been completed, "0" is returned.
href	string		The referenced resource to the analysis entity. See Analysis: Entity on page 157 for details of the analysis entity resource.
analysisResults	string		The referenced resource to the recommendations of the analysis. See Analysis: Kubernetes Container Recommendations on page 183 for details on container recommendations.
analysisStatus	string		The referenced resource to the status of the analysis. See Analysis: Status on page 206 for details.
message	string		For errors, the message for the status response is returned.
status	number		The HTTP response code of the request error. Possible status values include: - 200—success with request; - 400—invalid parameters; - 401—authentication failed; - 404—resource not found; - 500—internal server error.

Examples

Example: Listing All Container Analyses

The following example shows you how to list all container analyses in Densify.

Example: Listing all Container Analyses

Request:

```
GET /analysis/containers
```

Response:

```
[
   {
      "analysisId": "3d4ba999-cbd8-40b8-9998-574be6824a97",
      "analysisName": "DEN-east-321-45b",
      "analysisCompletedOn": 1510180908845,
      "href": "/analysis/containers/kubernetes/3d4ba999-cbd8-40b8-9998-
574be6824a97",
      "analysisResults": "/analysis/containers/kubernetes/3d4ba999-cbd8-40b8-
9998-574be6824a97/results",
      "analysisStatus": "/analysis/containers/kubernetes/3d4ba999-cbd8-40b8-
9998-574be6824a97/status"
   },
   {
      "analysisId": "97cfcb18-37a4-4f49-885b-8974eb2ceb31",
      "analysisName": "DEN-east-432-2",
      "analysisCompletedOn": 1543346324266,
      "href": "/analysis/containers/kubernetes/97cfcb18-37a4-4f49-885b-
8974eb2ceb31",
      "analysisResults": "/analysis/containers/kubernetes/97cfcb18-37a4-4f49-
885b-8974eb2ceb31/results",
      "analysisStatus": "/analysis/containers/kubernetes/97cfcb18-37a4-4f49-
885b-8974eb2ceb31/status"
   }
]
```

Analysis: Entity

Description

The `/analysis/<platformType>/<platformSubType>` resource is used to return a list of analyses currently in the Densify system. The `/analysis/<platformType>/<platformSubType>/<analysisId>` resource is used to return the details of a specific analysis currently in Densify. An analysis entity encompasses infrastructure resources within a predefined environment scope, specific to the platform and cloud vendor. The infrastructure resources are analyzed to determine optimal recommendations.

See Analysis: Cloud on page 150 and Analysis: Containers on page 154 for the two platform types of analyses supported in Densify.

Resource

```
/analysis/<platformType>/<platformSubType>
/analysis/<platformType>/<platformSubType>/<analysisId>
```

Supported Operations

Table: Analysis Supported Operations

Operation	HTTP Method	Input	Output	Description
List all	`GET`	Path Parameter:	Collection of:	Use this resource to

Operation	HTTP Method	Input	Output	Description
analyses for a particular platform and vendor	`/analysis/<platformType>/<platformSubType>`	platformType platformSubType	accountId accountName analysisCompletedOn analysisId analysisName analysisResults analysisStatus href policyInstanceId policyName	return a list of existing analyses for a particular platform and sub-platform (vendor). Example: Listing All Kubernetes Container Analyses Example: Listing All AWS Analyses Example: Listing All Azure Analyses
Get a specific analysis	`GET /analysis/<platformType>/<platformSubType>/<analysisId>`	Path Parameter: platformType platformSubType analysisId	accountId accountName analysisCompletedOn analysisId analysisName href analysisResults analysisStatus policyInstanceId policyName	Use this resource to return a specific analysis with a known analysis ID. Example: Getting Details of a Specific Kubernetes Container Analysis

Parameters

Path Parameters

Table: Analysis Path Parameters

Parameter Name	Type	Description
platformType	string	[cloud|containers] The technology platform of the analysis.
platformSubType	string	The platform sub-type of the analysis. This is typically the vendor/flavor of the technology platform. Possible platform sub-types depend on the platformType specified: For platformType = cloud, platformSubType = [aws|gcp|azure] For platformType = containers, platformSubType = [kubernetes]

Parameter Name	Type	Description
analysisId	string	The unique referenced ID of the analysis.

Response

Table: Analysis Response Schema

Element	Type	Filter/Sort	Description
accountId	string		The technology account identifier (i.e. AWS account number, Azure subscription ID, GCP project ID, etc.). This element is not returned for container analyses.
accountName	string		The technology platform account name (e.g. AWS account name). This element is not returned for container analyses.
analysisCompletedOn	string		The date and time (in milliseconds) when the last analysis completed. If an analysis has never been completed, "0" is returned.
analysisId	string		The unique identifier for the analysis entity in Densify.
analysisName	string	Filter by: - analysisName - analysisName_like	The name given to the analysis entity. The analysis name typically corresponds to the cluster name, account ID, subscription ID, or project ID from the infrastructure data collected. Use the `analysisName` and `analysisName_like` filter to return only analyses you are interested in. For example, `/cloud/aws?analysisName_like=bob` will return all AWS analyses with a "bob" substring in the analysis name.
analysisResults	string		The referenced resource to the recommendations of the analysis. For details of the various analysis recommendations, see: - Analysis: AWS Recommendations on page 105

Element	Type	Filter/Sort	Description
			Analysis: Azure Recommendations on page 132 Analysis: GCP Recommendations on page 165 Analysis: Kubernetes Container Recommendations on page 183
analysisStatus	string		The referenced resource to the status of the analysis. See Analysis: Status on page 206 for details.
href	string		The referenced resource to the analysis entity.
message	string		For errors, the message for the status response is returned.
policyInstanceId	string		The Densify policy instance GUID used for optimization analysis.
policyName	string		The Densify policy name used for optimization analysis.

status	number		The HTTP response code of the request error. Possible status values include:
			200—success with request;
			400—invalid parameters;
			401—authentication failed;
			404—resource not found;
			500—internal server error.

Examples

Example: Listing All Kubernetes Container Analyses

The following example shows you how to list all Kubernetes container analyses in Densify.

Example: Listing all Kubernetes Container Analyses

Request:

```
GET /analysis/containers/kubernetes
```

Response:

```
[
    {
        "policyInstanceId": "094d9c96-4802-4689-a260-ab4c08ee6123",
        "analysisCompletedOn": 1585725434757,
        "analysisResults": "/analysis/containers/kubernetes/8334887b-59ca-
42a4-a241-f970d7c306f1/results",
        "policyName": "Initial Assessment",
        "analysisId": "8334887b-59ca-42a4-a241-f970d7c306f1",
        "href": "/analysis/containers/kubernetes/8334887b-59ca-42a4-a241-
f970d7c306f1",
        "analysisStatus": "/analysis/containers/kubernetes/8334887b-59ca-42a4-
a241-f970d7c306f1/status",
        "analysisName": "mini-kube"
    }
]
```

Example: Listing All AWS Analyses

The following example shows you how to list all AWS analyses in Densify.

Example: Listing all AWS Analyses

Request:

```
GET /analysis/cloud/aws
```

Response:

```
[
 {
        "accountId": "922390019409",
        "policyInstanceId": "44b2ea5f-a3b7-44b2-9437-3c4e69650e58",
        "analysisCompletedOn": 1593606311863,
        "analysisResults": "/analysis/cloud/aws/0a87047f-e8dc-43f0-89bd-
769962b06b9d/results",
        "accountName": "General Services",
        "policyName": "AWS (Mobile_Dev)",
        "analysisId": "0a87047f-e8dc-43f0-89bd-769962b06b9d",
        "href": "/analysis/cloud/aws/0a87047f-e8dc-43f0-89bd-769962b06b9d",
        "analysisStatus": "/analysis/cloud/aws/0a87047f-e8dc-43f0-89bd-
769962b06b9d/status",
        "analysisName": "922390019409 (Mobile_Dev)"
    },
    {
        "accountId": "229192219122",
        "policyInstanceId": "ba022982-12da-48b8-b5f2-af6f0630952d",
        "analysisCompletedOn": 1585732332227,
        "analysisResults": "/analysis/cloud/aws/8bef9d74-94f7-414f-a032-
5855258473a2/results",
        "accountName": "General Services",
        "policyName": "AWS General Prod",
        "analysisId": "8bef9d74-94f7-414f-a032-5855258473a2",
        "href": "/analysis/cloud/aws/8bef9d74-94f7-414f-a032-5855258473a2",
        "analysisStatus": "/analysis/cloud/aws/8bef9d74-94f7-414f-a032-
5855258473a2/status",
        "analysisName": "229192219122"
    }
]
```

Example: Listing All Azure Analyses

The following example shows you how to list all Azure analyses in Densify.

Example: Listing all Azure Analyses

Request:

```
GET /analysis/cloud/azure
```

Response:

```
[
    {
        "accountId": "3d4ba999-cbd8-40b8-9998-574be6824a97",
        "policyInstanceId": "72aa4f1a-906a-4af4-94e9-f8fb606cdfe9",
        "analysisCompletedOn": 1589935057627,
        "analysisResults": "/analysis/cloud/azure/b3573fb3-0a37-406d-bb08-
24f088e8bc63/results",
        "accountName": "Test Subscription",
        "policyName": "Initial Assessment",
        "analysisId": "b3573fb3-0a37-406d-bb08-24f088e8bc63",
        "href": "/analysis/cloud/azure/b3573fb3-0a37-406d-bb08-24f088e8bc63",
        "analysisStatus": "/analysis/cloud/azure/b3573fb3-0a37-406d-bb08-
24f088e8bc63/status",
        "analysisName": "3d4ba999-cbd8-40b8-9998-574be6824a97"
    },
    {
        "accountId": "cc377154-9605-4cb0-8b41-1b39e1c4ac0f",
        "policyInstanceId": "72aa4f1a-906a-4af4-94e9-f8fb606cdfe9",
        "analysisCompletedOn": 1589978789707,
        "analysisResults": "/analysis/cloud/azure/c1e0bb64-6582-41e8-b7f6-
341797af4436/results",
        "policyName": "Initial Assessment",
        "analysisId": "c1e0bb64-6582-41e8-b7f6-341797af4436",
        "href": "/analysis/cloud/azure/c1e0bb64-6582-41e8-b7f6-341797af4436",
        "analysisStatus": "/analysis/cloud/azure/c1e0bb64-6582-41e8-b7f6-
341797af4436/status",
        "analysisName": "cc377154-9605-4cb0-8b41-1b39e1c4ac0f"
    },
    {
        "accountId": "00d89cbc-bc00-4d00-bcf6-ce6ec08d8fbc",
        "policyInstanceId": "129fa559-2db5-40ff-8a57-94c56900308a",
        "analysisCompletedOn": 1585726564073,
        "analysisResults": "/analysis/cloud/azure/6b8ab8e1-a026-4db9-a4ae-
2d684446731f/results",
        "policyName": "Azure Assessement",
        "analysisId": "6b8ab8e1-a026-4db9-a4ae-2d684446731f",
        "href": "/analysis/cloud/azure/6b8ab8e1-a026-4db9-a4ae-2d684446731f",
        "analysisStatus": "/analysis/cloud/azure/6b8ab8e1-a026-4db9-a4ae-
2d684446731f/status",
        "analysisName": "00d89cbc-bc00-4d00-bcf6-ce6ec08d8fbc"
    }
]
```

Example: Listing All GCP Analyses

The following example shows you how to list all GCP analyses in Densify.

Example: Listing all GCP Analyses

Request:

```
GET /analysis/cloud/gcp
```

Response:

```
[
    {
        "accountId": "gcpAcctEast-production-op",
        "policyInstanceId": "83003bdd-389e-4210-8ef0-c47a6d025c4d",
        "analysisCompletedOn": 1585736399537,
        "analysisResults": "/analysis/cloud/gcp/0930c31d-13ac-4a3f-892e-
2a35c8a3f842/results",
        "accountName": "DVC_GCP",
        "policyName": "GCP Assessement",
        "analysisId": "0930c31d-13ac-4a3f-892e-2a35c8a3f842",
        "href": "/analysis/cloud/gcp/0930c31d-13ac-4a3f-892e-2a35c8a3f842",
        "analysisStatus": "/analysis/cloud/gcp/0930c31d-13ac-4a3f-892e-
2a35c8a3f842/status",
        "analysisName": "gcpAcctEast-production-op"
    }
]
```

Example: Getting Details of a Specific Kubernetes Container Analysis

The following example shows you how to get the details of a Kubernetes container analysis in Densify with a known analysis ID.

Example: Getting Kubernetes Container Analysis Details

Request:

```
GET /analysis/containers/kubernetes/519a1b9f-aae0-4b65-bb63-1180ff68d27a
```

Response:

```
{
    "policyInstanceId": "094d9c96-4802-4689-a260-ab4c08ee6123",
    "analysisCompletedOn": 1585725429013,
    "analysisResults": "/analysis/containers/kubernetes/519a1b9f-aae0-4b65-
bb63-1180ff68d27a/results",
    "policyName": "Initial Assessment",
    "analysisId": "519a1b9f-aae0-4b65-bb63-1180ff68d27a",
    "href": "/analysis/containers/kubernetes/519a1b9f-aae0-4b65-bb63-
1180ff68d27a",
    "analysisStatus": "/analysis/containers/kubernetes/519a1b9f-aae0-4b65-
bb63-1180ff68d27a/status",
    "analysisName": "kube-master"
}
```

Analysis: GCP Recommendations

Description

The `/analysis/cloud/gcp/<analysisId>/results` resource is used to return a collection of Compute Engine instance recommendations after optimization analysis has been performed on your collected GCP infrastructure data.

For each instance recommendation, you can also download a PDF version of the Impact Analysis and Recommendation Report, which details system impact based on the recommendations. This report is useful to application owners who need to review recommended changes to their application resources. See the rptHref resource element for details on how to download this report.

To return a collection of all instances included in a GCP optimization analysis, see Analysis: GCP Systems on page 180.

Resource

```
/analysis/cloud/gcp/<analysisId>/results
```

Supported Operations

Table: GCP Recommendations Supported Operations

HTTP Method	Input	Output	Description
`GET /analysis/cloud/gcp/`*`<analysisId>`*`/results`	Path Parameter: analysisId Query String Parameter Options: Element Filters includeAttributes dataQuality Accept: application/json	Collection of (JSON): entityId resourceId accountIdRef currentType recommendationType recommendedType implementationMethod predictedUptime Analysis: GCP Recommendations Analysis: GCP Recommendations name rptHref approvalType densifyPolicy savingsEstimate effortEstimate powerState recommendedHostEntityId currentCost recommendedCost serviceType currentHourlyRate recommendedHourlyRate attributes recommFirstSeen recommLastSeen recommSeenCount auditInfo dataQuality	Returns a collection of recommendations for a GCP analysis. Specify `application/json` in the request header for returned recommendations in JSON format. ***Note**: The returned recommendations can be in either JSON or Terraform-map format.* Example: Returning GCP Instances with Downsize Recommendations
`GET /analysis/cloud/gcp/`*`<analysisId>`*`/results`	Path Parameter: analysisId Query String Parameter Options: Element Filters	Collection of (Terraform-map): provisioningId (label of each terraform-map recommendation) currentType	Returns a collection of recommendations for a GCP analysis in Terraform-map form. Specify `application/terraform-map` in the request

HTTP Method	Input	Output	Description
	Accept: - application/terraform-map	- recommendedType - approvalType - predictedUptime - recommendationType - powerState - implementationMethod - savingsEstimate - effortEstimate - densifyPolicy	header. ***Note**: The returned recommendations can be in either JSON or Terraform-map format.* Example: Returning GCP Recommendations with Low Effort in Terraform-map Form

Parameters

Path Parameters

Table: GCP Recommendations Path Parameters

Parameter Name	Type	Description
analysisId	string	The unique referenced ID of the GCP analysis.

Query String Parameters

Table: GCP Recommendation Query String Parameters

Parameter Name	Type	Description
Element Filters	string	You can use element filters to return a targeted subset of the recommendations. See the "F" (filter) designation in the Response schema table for a list of elements that support filtering. Refer to Filters on page 30 for a complete description of this common operation feature. Usage example: `../results?recommendationType=Upsize`
includeAttributes	- true - false	Indicate whether or not to return system attributes: - true—returns all the system attributes; - false—(default) suppress system attributes from the response output. Usage example:

Parameter Name	Type	Description
		`../results?includeAttributes=true`
dataQuality	string	Allows you to indicate which workload type data collection detail to return for the system. See dataQuality for the returned details. Specify a quoted list of workload type names for which data to be returned, separated by commas. The supported workload type names can be found in the Data Center Explorer (DCE) from the Analysis Console. See DCE Virtual Environment Workload Viewer to find a list of the supported workload types from the Analysis Console. Usage example: `../results?dataQuality="CPU Utilization (CINT2006 Rate),Memory Utilization in Percent"` ***Note**: A "400-Bad Request" error message is returned if a non-supported workload type is specified.*

Response

Table: GCP Recommendations Response Schema

Element	Type	Filter/Sort	Description
entityId	string	F	The Densify assigned entity ID of the cloud system.
resourceId	string	F	The GCP identifier assigned to the Compute Engine system .
accountIdRef	string	F	The GCP Project identifier.
currentType	string	F	The current instance type of the Compute Engine system.
recommendationType	string	F	The recommended action for the system. This is also known as the Optimization Type in the Densify Console (see *Optimization Type Color-Coding Summary* in the topic *Understanding the Instance Optimization Details Report* (Help Topic ID 380390)). The following types of recommended actions are supported for this cloud platform: ▪ `"Just Right"`—this instance is optimally sized for the workload; ▪ `"Upsize - Optimal Family"`—this

Element	Type	Filter/Sort	Description
			instance should be upsized to a more optimal instance family; - `"Upsize"`—this instance should be upsized to an instance within the same instance family; - `"Terminate"`—this instance should be terminated; - `"Downsize - Optimal Family"`—this instance should be downsized to an instance belonging to a more suited instance family; - `"Downsize"`—this instance should be downsized to an instance within the same instance family; - `"Modernize - Optimal Family"`—this instance should be modernized to an instance belonging to a more optimal instance family; - `"Modernize"`—this instance should be modernized to an instance within the same instance family. - `"Not Analyzed"`—this instance has no recommendation due to insufficient workload information. Systems with insufficient information for analysis do not have a recommendation and are returned with limited elements. See Example: Returning GCP Systems with No Recommendations.
recommendedType	string	F	The recommended instance type after Densify optimization analysis.
implementationMethod	string	F	[Self Optimization|Manual|N/A] Specifies whether this system is configured for Self-Optimization or Manual actioning based on the recommended action (recommendationType) and on the Self-Optimizing Automation policies.
predictedUptime	percentage		The predicted uptime (%) for the system is based on the percentage of hours CPU utilization data is present in the workload range specified in the policy settings. Predicted uptime % for new systems started mid-way within the workload range is calculated from the time/date that the

Element	Type	Filter/Sort	Description
			system was started, as opposed to the beginning of the interval resulting, in more accurate prediction for the future.
name	string	F	The GCP name assigned to the system.
rptHref	string		The reference resource to the Impact Analysis and Recommendation Report (also known as the Application Owner report). The PDF report for the specified instance is available for download after the reporting database tables have been updated (i.e. after *RDB populate* has been executed). By default, the reporting database tables are updated once every night. See *Viewing the Impact Analysis and Recommendation Report* (Help Topic ID 380450) for details on the content of the report. You need to use the following in the request header to download the PDF file: `Accept : application/octet-stream`. Example: Downloading an Impact Analysis and Recommendation Report
approvalType	string	F	The approval setting for the system recommendation. The value in this string is derived from the Self-Optimizing Automation policies in implementationMethod and the approval attribute `attr_ApprovalSetting`. Possible settings include: - "`na`"— not approved; - "`all`"—approve any change; - "*`<recommended-instance-type>`*"—approve changing the instance to the specified *<recommended-instance-type>*.
densifyPolicy	string	F	The Densify policy used for optimization analysis.
savingsEstimate	string		The value of savingestimate output parameter is the difference between the current and recommended instance type cost (this is the catalog cost). When using

Element	Type	Filter/Sort	Description
			the API, the predicted uptime is NOT taken into consideration (i.e. [currentCost – recommendedCost]). The Impact Analysis and Recommendation Report report uses the predicted uptime % when calculating estimated savings regardless of whether the report is obtained through the UI or via API.
effortEstimate	string	F	[Moderate|Low|Very Low|None|Impossible] This element describes the effort required to investigate and implement the Densify recommendations. Effort for each system is calculated by rule-driven analytics based on factors (such as instance family change, storage change, data quality checks, feature changes, etc.) that can be configured in the policy settings and rule sets which capture best practices. "Impossible" effort is a result of a manual override for the instance. If a system is not ‘Not Analyzed’, Densify does not return an effortEstimate. ***Note***: *When using the Subscription API, Densify returns an effortEstimate of "impossible" for systems that are ‘Not Analyzed’.*
powerState	string	F	The power state of the system.
recommendedHostEntityId	string	F	The Densify entity ID of the catalog instance for the recommended instance type.
currentCost	string		The cost of the existing instance type (i.e. the instance catalog cost).
recommendedCost	string		The cost of the recommended instance type (i.e. the instance catalog cost) after Densify optimization analysis.
serviceType	string	F	The GCP service type: - Compute Engine
currentHourlyRate	string		The hourly rate for the current instance type (i.e. instance catalog cost / monthly hours). This value is rounded to the nearest penny.
recommendedHourlyRate	string		The hourly rate for the recommended instance type (i.e. instance catalog cost / monthly hours). This value is rounded to the

Element	Type	Filter/Sort	Description
			nearest penny.
attributes	array of id name value		System attributes are properties set during the data collection process by a vendor platform (i.e. GCP) or by Densify for analytics. ***Note***: *The attribute array is only returned when includeAttributes=true is added to the query string.*
provisioningId	string		This element is used to identify the terraform-map recommendations for a unique system and corresponds to the "Provisioning Id" GCP user-defined label value. The "Provisioning Id" label is used to uniquely identify a system, since its resourceId or system name could possibly change after an instance type update. See Example: Returning GCP Recommendations with Low Effort in Terraform-map Form. If the "Provisioning Id" label value is not set, then the system name is used to identify the recommendations.
recommFirstSeen	Unix time (in milliseconds)		The first date and time the recommended instance type (i.e. the recommendedType element) was provided by Densify (Unix Epoch time, in milliseconds).
recommLastSeen	Unix time (in milliseconds)		The latest date and time the recommended instance type (i.e. the recommendedType element) was provided by Densify (Unix Epoch time, in milliseconds).
recommSeenCount	integer		The number of times Densify suggested the recommended instance type (i.e. the recommendedType element). This is the count of Densify analysis processes which produced the same recommended instance type from recommFirstSeen to recommLastSeen. This value is updated when the RDB populate task (i.e. the reporting database update process) is executed. The RDB populate task compares the current `recommendedType` with the new `recommendedType` to update the `recommSeenCount` counter.

Element	Type	Filter/Sort	Description
			***Note**: In typical production environments, where the RDB populate task is scheduled to run once daily post data collection and analysis, the `recommSeenCount` value will reflect exactly the number of times the recommended instance type was provided. If your environment executes the RDB populate task more than once daily, the `recommSeenCount` value will be inflated beyond the actual number of times the instance type was recommended. Contact Support@Densify.com if you have concerns about the `recommSeenCount` value.*
auditInfo	▫ dataCollection: • dateFirstAudited • dateLastAudited • auditCount ▫ workloadDataLast30: • firstDate • lastDate • totalDays • seenDays		The following system data collection details are returned: ▫ `dateFirstAudited`—the first time data was collected for this system (Unix epoch time, in milliseconds); ▫ `dateLastAudited`—the most recent data collection time (Unix epoch time, in milliseconds); ▫ `auditCount`—the number of times that data was collected. The following system workload collection details, for the last 30 days, are returned: ▫ `firstDate`—the first time workload data was collected for this system (Unix epoch time, in milliseconds); ▫ `lastDate`—the most recent workload data collection time for this system (Unix epoch time, in milliseconds); ▫ `totalDays`—the difference in days between `firstDate` and `lastDate`; ▫ `seenDays`—the number of days that at least one workload data was added into Densify for this system. ***Note**: If no workload data is collected for the system in the last 30 days, then the workloadDataLast30 element block is not returned.*

Element	Type	Filter/Sort	Description
			***Note**: The values in `auditInfo` are updated once a day, after the data collection and RDB populate processes are complete (i.e. the reporting tables have been updated with latest data collected).*
dataQuality	array of: - workloadName - firstSeen - lastSeen - completeDays - partialDays		The dataQuality array provides workload type data collection details for the system: - `workloadName`—the workload type name (see DCE Virtual Environment Workload Viewer to find a list of the supported workload types from the Analysis Console); - `firstSeen`—the first time this workload was collected (Unix Epoch time, in milliseconds); - `lastSeen`—the most recent time this workload was collected (Unix Epoch time, in milliseconds); - `completeDays`—the number of complete days that this workload data was collected; - `partialDays`—the number of partial days that this workload data was collected. ***Note**: The dataQuality array is only returned when the dataQuality query string is specified in the request. In addition, if you specify a workload type in the request, for which system workload data does not exist in Densify, then no data is returned.* ***Note**: The values in the dataQuality array are updated after data collection and subsequent updates to the reporting tables (i.e. RDB Populate process) are completed.* See Example: Returning GCP Instances with Downsize Recommendations.

Examples

Example: Returning GCP Instances with Downsize Recommendations

The following example shows you how to return a collection of GCP instances with "Downsize" recommendations. In addition, this example also returns the "Disk I/O Bytes" workload data collection timestamps.

Example: Returning GCP Downsize Recommendations with "Disk I/O Bytes" Workload

Request:

```
GET /analysis/cloud/gcp/d230c31d-13ac-543f-892e-2a35c8a3f232/res-
ults?recommendationType=Downsize&dataQuality="Disk I/O Bytes"
```

Response:

```
[
   {
      "entityId": "1679679b-9726-4a6b-9c33-a00f9261b6c1",
      "resourceId": "4090409965214718888",
      "accountIdRef": "east-prod-gidz",
      "currentType": "n1-standard-8",
      "recommendationType": "Downsize",
      "recommendedType": "n1-standard-4",
      "implementationMethod": "Self Optimization",
      "predictedUptime": 13.82,
      "totalHoursRunning": 230,
      "name": "gl04.doop_xtr-32",
      "rptHref": "/systems/1679679b-9726-4a6b-9c33-a00f9261b6c1/analysis-
report",
      "approvalType": "na",
      "densifyPolicy": "GCP-Assess-Prod",
      "savingsEstimate": 132.91621,
      "effortEstimate": "Very Low",
      "powerState": "Running",
      "recommendedHostEntityId": "4e8309c5-cf70-42d9-ab70-4aa46516ff62",
      "currentCost": 277.4,
      "recommendedCost": 138.16,
      "serviceType": "Compute Engine",
      "currentHourlyRate": 0.38,
      "recommendedHourlyRate": 0.24,
      "recommFirstSeen": 1579680587570,
      "recommLastSeen": 1589008760110,
      "recommSeenCount": 24,
      "auditInfo": {
            "dataCollection": {
                "dateFirstAudited": 1571951513223,
                "dateLastAudited": 1588997765350,
                "auditCount": 101
            },
            "workloadDataLast30": {
                "firstDate": 1587009600000,
                "lastDate": 1588910400000,
                "totalDays": 23,
                "seenDays": 5
            }
      },
      "dataQuality": [
            {
                "workloadName": "Disk I/O Bytes",
                "firstSeen": 1569297600000,
                "lastSeen": 1588910400000,
                "completeDays": 116,
                "partialDays": 0
            }
      ]
   },
   ...
]
```

Example: Returning GCP Recommendations with Low Effort in Terraform-map Form

The following example shows you how to return a collection of recommendations with Low effort in terraform-map form. The label of each recommendation (i.e. "gl02.camp_gas-340", "gl02.camp_gas-432" in the example below) is the provisioningId element: "Provisioning Id" GCP label or system name value.

Example: Return Low Recommendations in Terraform-map

Request:

```
GET /analysis/cloud/gcp/d230c31d-13ac-543f-892e-2a35c8a3f232/results?effortEstimate=Low
```

Headers:

```
Accept: application/terraform-map
Authorization: Bearer <apiToken>
```

Response:

```
densify_recommendations = {
  "gl02.camp_gas-340" = {
    currentType = "n1-standard-8"
    recommendedType = "n1-highmem-4"
    approvalType = "na"
    predictedUptime = "64.01"
    recommendationType = "Downsize - Optimal Family"
    powerState = "Running"
    implementationMethod = "Manual"
    savingsEstimate = "98.27805"
    effortEstimate = "Low"
    densifyPolicy = "GCP-Assess-Prod"
  }
  "gl02.camp_gas-432" = {
    currentType = "n1-standard-8"
    recommendedType = "n1-highmem-4"
    approvalType = "na"
    predictedUptime = "88.0"
    recommendationType = "Downsize - Optimal Family"
    powerState = "Running"
    implementationMethod = "Manual"
    savingsEstimate = "98.27805"
    effortEstimate = "Low"
    densifyPolicy = "GCP-Assess-Prod"
  }
   ...
}
```

Example: Downloading an Impact Analysis and Recommendation Report

The following example shows you how to download a PDF Impact Analysis and Recommendation Report from the `rptHref` resource element provided in the instance recommendation output.

Note: *HTTPS needs to be enabled to download the Impact Analysis and Recommendation Report PDF.*

Example: Download an Impact Analysis and Recommendation Report

Request:

```
GET /systems/1679679b-9726-4a6b-9c33-a00f9261b6c1/analysis-report
```

Headers:

```
Accept: application/octet-stream
Authorization: Bearer <apiToken>
```

Example: Returning GCP Systems with No Recommendations

The following example shows you how to return a collection of GCP systems without recommendations. These systems typically do not have adequate data for optimization analysis and have the "Not Analyzed" designation in the recommendationType element.

Example: Returning GCP Systems with No Recommendations In JSON

Request:

```
GET /analysis/cloud/gcp/6b8ab8e1-a026-4db9-a4ae-2d684446731f/results?recommendationType=Not Analyzed
```

Headers:

```
Accept: application/json
Authorization: Bearer <apiToken>
```

Response:

```
[
    {
        "entityId": "1679679b-9726-4a6b-9c33-a00f9261b6c1",
        "resourceId": "4090409723758657994",
        "accountIdRef": "Prod9873318",
        "currentType": "n1-standard-8",
        "recommendationType": "Not Analyzed",
        "name": "g105.doop_xtr-32",
        "densifyPolicy": "GCP Assess",
        "powerState": "Offline",
        "currentCost": 27.47,
        "serviceType": "Compute Engine",
        "recommFirstSeen": 1579680587570,
        "recommLastSeen": 1589008760110,
        "recommSeenCount": 24,
        "auditInfo": {
            "dataCollection": {
                "dateFirstAudited": 1571951513223,
                "dateLastAudited": 1588997765350,
                "auditCount": 101
            }
            }
        },
        ...
]
```

Example: Returning GCP Systems with No Recommendations in Terraform-Map

Request:

```
GET /analysis/cloud/gcp/6b8ab8e1-a026-4db9-a4ae-2d684446731f/res-
ults?recommendationType=Not Analyzed
```

Headers:

```
Accept: application/terraform-map
Authorization: Bearer <apiToken>
```

Response:

```
densify_recommendations = {
  "g105.doop_xtr-32" = {
    currentType = "n1-standard-8"
    recommendationType = "Not Analyzed"
    powerState = "Offline"
    densifyPolicy = "GCP Assess"
  }
  ...
}
```

Analysis: GCP Systems

Description

The `/analysis/cloud/gcp/<analysisId>/systems` resource is used to return a collection of all the systems that were included in the Google Cloud Platform (GCP) optimization analysis.

The recommendations from an GCP optimization analysis can be obtained using the `/analysis/cloud/gcp/<analysisId>/results` resource. See Analysis: GCP Recommendations on page 165 for details on the GCP recommendations resource. The set of analyzed systems may be more than the set of recommendations produced, as there can be potentially no recommendations for some analyzed systems.

Resource

```
/analysis/cloud/gcp/<analysisId>/systems
```

Supported Operations

Table: GCP Systems Supported Operations

Operation	HTTP Method	Input	Output	Description
List all systems included in a GCP analysis	`GET /analysis/cloud/gcp/<analysisId>/systems`	Path Parameter: analysisId	Collection of: ▪ serviceType ▪ resourceId ▪ powerState	Use this resource to return a list of all systems included in the GCP analysis.

Operation	HTTP Method	Input	Output	Description
			currentType displayName entityId href	Example: Listing All Systems in a GCP Analysis

Parameters

Path Parameters

Table: GCP Systems Path Parameters

Parameter Name	Type	Description
analysisId	string	The unique referenced ID of the GCP analysis.

Response

Table: GCP Systems Response Schema

Element	Type	Filter/Sort	Description
serviceType	string		The cloud service type (i.e. Compute Engine).
resourceId	string		The GCP unique identifier assigned to the instance system.
powerState	string		The power state of the system.
currentType	string		The current instance type of the system.
displayName	string		The GCP name assigned to the system.
entityId	string		The Densify assigned entity ID of the system.
href	string		The referenced resource to the system entity. See Systems on page 408 for details of the `/systems` resource.

Examples

Example: Listing All Systems in a GCP Analysis

The following example shows you how to return all systems (instances) included in an GCP optimization analysis.

Example: Listing all Systems in a GCP Analysis

Request:

```
GET /analysis/cloud/gcp/d930c31d-13ac-4a3f-892e-2a35c8a3f232/systems
```

Response:

```
[
   {
      "serviceType": "Compute Engine",
      "resourceId": "4390409739570762220",
      "powerState": "Running",
      "currentType": "n1-standard-8",
      "displayName": "gl08.mln_gas-286",
      "entityId": "00d60166-b9a5-4825-990e-a2efeb3caa32",
      "href": "/systems/00d60166-b9a5-4825-990e-a2efeb3caa32"
   },
   {
      "serviceType": "Compute Engine",
      "resourceId": "4090399486108714123",
      "powerState": "Running",
      "currentType": "n1-highmem-8",
      "displayName": "gl08.hosa_far-348",
      "entityId": "02cbfdb5-3a0d-4ecb-8732-c6f04cfcf7f9",
      "href": "/systems/02cbfdb5-3a0d-4ecb-8732-c6f04cfcf7f9"
   },
   ...
]
```

Analysis: Kubernetes Container Recommendations

Description

The `/analysis/containers/kubernetes/<analysisId>/results` resource is used to return a collection of recommendations after optimization analysis has been performed on your Kubernetes container data in Densify.

To return a collection of all the Kubernetes containers included in the optimization analysis, see Analysis: Kubernetes Container Systems on page 196. The set of analyzed containers may be more than the set of container recommendations due to the possibility of no recommendations for some analyzed containers.

Resource

```
/analysis/containers/kubernetes/<analysisId>/results
```

Supported Operations

Table: Kubernetes Container Recommendations Supported Operations

HTTP Method	Input	Output	Description
`GET /analysis/containers/kubernetes/<analysisId>/results`	Path Parameter: analysisId Query String Parameter Options: Element Filters dataQuality Accept: application/json	Collection of (JSON): container cluster hostName predictedUptime displayName recommLastSeen podService auditInfo recommendedCpuLimit currentCount recommSeenCount currentMemLimit recommendedMemLimit recommendationType recommendedCpuRequest recommFirstSeen controllerType currentMemRequest entityId currentCpuLimit dataQuality recommendedMemRequest currentCpuRequest namespace	Returns a collection of recommendations for a Kubernetes container analysis. Specify `application/json` in the request header for returned recommendations in JSON format. **Note**: *The returned recommendations can be in either JSON or Terraform-map format.* Example: Returning Kubernetes Container Recommendations
`GET /analysis/containers/kubernetes/<analysisId>/results`	Path Parameter: analysisId Query String Parameter Options: Element Filters Accept: application/terraform-map	Collection of (Terraform-map): container cluster hostName predictedUptime controllerType displayName (the label of each terraform-map recommendation)	Returns a collection of recommendations for a Kubernetes container analysis. Specify `application/terraform-map` in the request header for Terraform-map output. **Note**: *The returned recommendations can be in either*

HTTP Method	Input	Output	Description
		currentMemRequest currentCpuLimit podService recommendedCpuLimit recommendedMemRequest currentCpuRequest currentCount currentMemLimit namespace recommendedMemLimit recommendationType recommendedCpuRequest	*JSON or Terraform-map format.* Example: Returning Kubernetes Container Recommendations in Terraform-map Form

Parameters

Path Parameters

Table: Kubernetes Container Recommendations Path Parameters

Parameter Name	Type	Description
analysisId	string	The unique referenced ID of the Kubernetes analysis.

Query String Parameters

Table: Kubernetes Container Recommendations Query String Parameters

Parameter Name	Type	Description
Element Filters	string	You can use element filters to return a targeted subset of the recommendations. See the "F" (filter) designation in the Response schema table for a list of elements that support filtering. Refer to Filters on page 30 for a complete description of this common operation feature. Usage example: `../results?recommendationType=Resize`
dataQuality	string	Allows you to indicate which workload type data collection detail to return for the system. See dataQuality for the returned details.

Parameter Name	Type	Description
		Specify a quoted list of workload type names for which data to be returned, separated by commas. The supported workload type names can be found in the Data Center Explorer (DCE) from the Analysis Console. See DCE Virtual Environment Workload Viewer to find a list of the supported workload types from the Analysis Console. Usage example: `../results?dataQuality="CPU Utilization (CINT2006 Rate),Memory Utilization in Percent"` ***Note***: *A "400-Bad Request" error message is returned if a non-supported workload type is specified.*

Response

Table: Kubernetes Container Recommendations Response Schema

Element	Type	Filter/Sort	Description
container	string	F	The container name.
cluster	string	F	The container's cluster name.
hostName	string		The Densify internal reference used to ensure that containers are uniquely identified across multiple clusters, even if they potentially have the same namespace, pod and/or container names.
predictedUptime	percentage		The predicted uptime (%) for the container is based on the percentage of hours that CPU utilization data is present in the historical interval specified in the policy settings.
displayName	string	F	The display name is a combination of the pod and container name.
recommLastSeen	Unix time (in milliseconds)		The latest date and time that the following set of recommendations were provided by Densify(Unix Epoch time, in milliseconds): recommendedMemLimit recommendedCpuLimit recommendedMemRequest recommendedCpuRequest on page

Element	Type	Filter/Sort	Description
			189 ***Note**: If any recommendation value was updated, then Densify would considered that as a new set of recommendations and the recommLastSeen value would not be updated.*
podService	string	F	The pod or service name for the container.
auditInfo	▪ dataCollection: • dateFirstAudited • dateLastAudited • auditCount ▪ workloadDataLast30: • firstDate • lastDate • totalDays • seenDays		The following system data collection details are returned: ▪ `dateFirstAudited`—the first time data was collected for this container (Unix epoch time, in milliseconds); ▪ `dateLastAudited`—the most recent data collection time (Unix epoch time, in milliseconds); ▪ `auditCount`—the number of times that data was collected. The following container workload collection details, for the last 30 days, are returned: ▪ `firstDate`—the first time workload data was collected for this container (Unix epoch time, in milliseconds); ▪ `lastDate`—the most recent workload data collection time for this container (Unix epoch time, in milliseconds); ▪ `totalDays`—the difference in days between `firstDate` and `lastDate`; ▪ `seenDays`—the number of days that at least one workload data was added into Densify for this container. ***Note**: If no workload data is collected for the container in the last 30 days, then the workloadDataLast30 element block is not returned.* ***Note**: The values in `auditInfo` are*

Element	Type	Filter/Sort	Description
			updated once a day (after the data collection and RDB populate processes are complete).
recommendedCpuLimit	float		The recommended CPU limit for the container after Densify optimization analysis.
currentCount	int		The number of copies running in the Controller. This value comes form the container data collection. If size of the Controller is not found, then this value is set to 1.
recommSeenCount	integer		The number of times Densify recommended the same values for the following elements: - recommendedMemLimit - recommendedCpuLimit - recommendedMemRequest - recommendedCpuRequest on page 189 This is the count of Densify analysis processes which produced the same recommendation values from recommFirstSeen to recommLastSeen. This value is updated when the RDB populate task (i.e. the reporting database update process) is executed. ***Note**: In typical production environments, where the RDB populate task is scheduled to run once daily post data collection and analysis, the* `recommSeenCount` *value will reflect exactly the number of times the same recommendations were provided after data collection and analysis. If your environment executes the RDB populate task more than once daily, the* `recommSeenCount` *value will be inflated beyond the actual number of times the analysis produced the same recommendations. Contact Support@Densify.com if you*

Element	Type	Filter/Sort	Description
			have concerns about the `recommSeenCount` *value.*
currentMemLimit	float		The current memory limit configured for the container.
recommendedMemLimit	float		The recommended memory limit for the container after Densify optimization analysis.
recommendationType	string	F	The recommended action for this container: - `"Just Right"`—this container manifest is optimally configured for the workload; - `"Upsize"`—increase one or more of CPU Request, CPU Limit, Memory Request, or Memory Limit settings; - `"Downsize"`—decrease one or more of CPU Request, CPU Limit, Memory Request, or Memory Limit settings; - `"Resize"`— resize at least two of CPU Request, CPU Limit, Memory Request, or Memory Limit settings (i.e. at least one recommendation is a size increase and the other one is a size decrease); - `"Resize from Unspecified"`—resize recommendations for this container manifest are made without current CPU Request, CPU Limit, Memory Request, or Memory Limit values; - `"Not Analyzed"`— there is insufficient data to recommend CPU Request, CPU Limit, Memory Request, or Memory Limit values. Containers with insufficient information for analysis do not have recommendations and are returned with limited elements. See Example: Returning Kubernetes Containers with No Recommendations.
recommendedCpuRequest	float		The recommended CPU request for the container after Densify optimization analysis.

Element	Type	Filter/Sort	Description
recommFirstSeen	Unix time (in milliseconds)		The first date and time that the following set of recommendations were provided by Densify (Unix Epoch time, in milliseconds): • recommendedMemLimit • recommendedCpuLimit • recommendedMemRequest • recommendedCpuRequest on page 189 ***Note**: If any recommendation value (above) is different than the previous set of recommFirstSeen recommendation values, then Densify would considered that as a new set of recommendations and the recommFirstSeen value would be updated.*
controllerType	string		The type of controller (i.e. "ReplicatSet", "ReplicationController", "DaemonSet", "StatefulSet", "Deployment", "Job", etc.).
currentMemRequest	float		The current memory request configured for the container.
entityId	string		The Densify assigned entity ID of the container system.
currentCpuLimit	float		The current CPU limit configured for the container.
dataQuality	array of: • workloadName • firstSeen • lastSeen • completeDays • partialDays		The dataQuality array provides workload type data collection details for the system: • `workloadName`—the workload type name (see DCE Virtual Environment Workload Viewer to find a list of the supported workload types from the Analysis Console); • `firstSeen`—the first time this workload was collected (Unix Epoch time, in milliseconds); • `lastSeen`—the most recent time this workload was collected (Unix Epoch time, in milliseconds); • `completeDays`—the number of complete days that this workload data was collected;

Element	Type	Filter/Sort	Description
			- `partialDays`—the number of partial days that this workload data was collected. ***Note****: The dataQuality array is only returned when the dataQuality query string is specified in the request. In addition, if you specify a workload type in the request, for which system workload data does not exist in Densify, then no data is returned.* ***Note****: The values in the dataQuality array are updated after data collection and subsequent updates to the reporting tables (i.e. RDB Populate process) are completed.*
recommendedMemRequest	float		The recommended memory request for the container after Densify optimization analysis.
currentCpuRequest	float		The current CPU request configured for the container.
namespace	string	F	The container's namespace.

Examples

Example: Returning Kubernetes Container Recommendations

The following example shows you how to return a collection of Kubernetes Container recommendations in JSON format.

Example: Returning Kubernetes Container Recommendations

Request:

```
GET /analysis/containers/kubernetes/e9298ac3-a143-41ed-b7d7-62f659f2a4c6/res-
ults
```

Headers:

```
Accept: application/json
Authorization: Bearer <apiToken>
```

Response:

```
[
    {
        "container": "kube-apiserver",
        "cluster": "kube-ctrl",
        "hostName": "e89c16ee-1f7c-3e40-bdfa-75f33de36405",
        "predictedUptime": 0.08,
        "displayName": "kube-apiserver-kube-ctrl.in.densify.com__kube-
apiserver",
        "recommLastSeen": 1597104000000,
        "podService": "kube-apiserver-kube-ctrl.in.densify.com",
        "auditInfo": {
          "workloadDataLast30": {
                "totalDays": 29,
                "seenDays": 29,
                "firstDate": 1594612800000,
                "lastDate": 1597104000000
            },
            "dataCollection": {
                "auditCount": 40,
                "dateFirstAudited": 1593648000000,
                "dateLastAudited": 1597104000000
            }
        },
        "recommendedCpuLimit": 80,
        "currentCount": 4,
        "recommSeenCount": 6,
        "currentMemLimit": 630,
        "recommendedMemLimit": 630,
        "recommendationType": "Just Right",
        "recommendedCpuRequest": 80,
        "recommFirstSeen": 1593648000000,
        "controllerType": "ReplicationController",
        "currentMemRequest": 628,
        "entityId": "06df1b52-a4ac-4328-94c5-67bfa0ea95b4",
        "currentCpuLimit": 80,
        "recommendedMemRequest": 628,
        "currentCpuRequest": 80,
        "namespace": "kube-system"
    },
    {
        "container": "kube-scheduler",
        "cluster": "kube-ctrl",
        "hostName": "56d85c60-94f2-388f-9e2b-74c5afa75beb",
        "predictedUptime": 0.08,
        "displayName": "kube-scheduler-kube-ctrl.in.densify.com__kube-sched-
uler",
        "recommLastSeen": 1597104000000,
        "podService": "kube-scheduler-kube-ctrl.in.densify.com",
      ...
   },
   ...
]
```

Example: Returning Kubernetes Container Recommendations in Terraform-map Form

The following example shows you how to return a collection of recommendations in terraform-map format.

Example: Return Kubernetes Container Recommendations in Terraform-map

Request:

```
GET /analysis/containers/kubernetes/e7298ac3-a143-41ed-b7d7-62f659f2a4c6/res-
ults
```

Headers:

```
Accept: application/terraform-map
Authorization: Bearer <apiToken>
```

Response:

```
densify_recommendations = {
  "kube-controller-manager-kube-ctrl.in.densify.com__kube-controller-manager"
= {
    container = "kube-controller-manager"
    cluster = "kube-ctrl"
    hostName = "3e021430-5b65-307c-b049-3c0edca9be43"
    predictedUptime = "0.08"
    controllerType = "ReplicationController"
    displayName = "kube-controller-manager-kube-ctrl.int.densify.com__kube-con-
troller-manager"
    currentMemRequest = "152.0"
    currentCpuLimit = "30.0"
    podService = "kube-controller-manager-kube-ctrl.in.densify.com"
    recommendedCpuLimit = "30.0"
    recommendedMemRequest = "152.0"
    currentCpuRequest = "30.0"
    currentCount = "2"
    currentMemLimit = "152.0"
    namespace = "kube-system"
    recommendedMemLimit = "152.0"
    recommendationType = "Just Right"
    recommendedCpuRequest = "30.0"
  }
  "kubernetes-dashboard-845747bdd4__kubernetes-dashboard" = {
    container = "kubernetes-dashboard"
    cluster = "kube-ctrl"
    hostName = "93d0c4a7-a915-3cc8-80cf-19729e8b091c"
    predictedUptime = "0.08"
    controllerType = "ReplicaSet"
    displayName = "kubernetes-dashboard-845747bdd4__kubernetes-dashboard"
    currentMemRequest = "68.0"
    currentCpuLimit = "10.0"
    podService = "kubernetes-dashboard-845747bdd4"
    recommendedCpuLimit = "10.0"
    recommendedMemRequest = "68.0"
    currentCpuRequest = "10.0"
    currentCount = "1"
    currentMemLimit = "68.0"
    namespace = "kube-system"
    recommendedMemLimit = "68.0"
    recommendationType = "Just Right"
    recommendedCpuRequest = "10.0"
  }
   ...
}
```

Example: Returning Kubernetes Containers with No Recommendations

The following example shows you how to return a collection of Kubernetes containers without recommendations. These containers typically do not have adequate data for optimization analysis and have the "Not Analyzed" designation in the recommendationType element.

Example: Returning Kubernetes Containers with No Recommendations In JSON

Request:

```
GET /analysis/containers/kubernetes/8334887b-59ca-42a4-a241-f970d7c306f1/res-
ults?recommendationType=Not Analyzed
```

Headers:

```
Accept: application/json
Authorization: Bearer <apiToken>
```

Response:

```
[
    {
        "container": "data-forwarder",
        "recommFirstSeen": 1594080000000,
        "cluster": "kube-ctrl",
        "controllerType": "ReplicaSet",
        "displayName": "densify1__data-forwarder",
        "recommLastSeen": 1597104000000,
        "currentMemRequest": 2048,
        "entityId": "b1253c3b-16fe-47b5-b4db-6a72f541ab7e",
        "podService": "densify1",
        "currentCpuRequest": 520,
        "currentCount": 1,
        "recommSeenCount": 4,
        "currentMemLimit": 3584,
        "namespace": "default",
        "recommendationType": "Not Analyzed"
    }
]
```

Example: Returning Kubernetes Containers with No Recommendations in Terraform-Map

Request:

```
GET /analysis/containers/kubernetes/8334887b-59ca-42a4-a241-f970d7c306f1/results?recommendationType=Not Analyzed
```

Headers:

```
Accept: application/terraform-map
Authorization: Bearer <apiToken>
```

Response:

```
densify_recommendations = {
 "densify1__data-forwarder" = {
    container = "data-forwarder"
    cluster = "kube-master"
    controllerType = "ReplicaSet"
    currentCpuRequest = "520.0"
    displayName = "densify1__data-forwarder"
    currentCount = "1"
    currentMemLimit = "3584.0"
    namespace = "default"
    currentMemRequest = "2048.0"
    recommendationType = "Not Analyzed"
    podService = "densify1"
  }
}
```

Analysis: Kubernetes Container Systems

Description

The `/analysis/containers/kubernetes/<analysisId>/systems` resource is used to return a collection of all container systems that were included in the Kubernetes optimization analysis.

The number of analyzed systems (i.e. `/containers/kubernetes/<analysisId>/systems` entities) will always be greater than or equal to the number of system recommendations produced (i.e. `/containers/kubernetes/<analysisId>/results` entities), as some systems may not have any recommendations. See Analysis: Kubernetes Container Recommendations on page 183 for details of the Kubernetes container recommendation resource.

Resource

```
/analysis/containers/kubernetes/<analysisId>/systems
```

Supported Operations

Table: Kubernetes Container Systems Supported Operations

HTTP Method	Input	Output	Description
`GET /analysis/containers/kubernetes/<analysisId>/systems`	Path Parameter: analysisId	Collection of: hostName displayName entityId href	Use this resource to return a list of all container systems included in a Kubernetes container analysis. Example: Listing All Systems in a Kubernetes Container Analysis

Parameters

Path Parameters

Table: Kubernetes Container Systems Path Parameters

Parameter Name	Type	Description
analysisId	string	The unique referenced ID of the Kubernetes container analysis.

Response

Table: Kubernetes Container Systems Response Schema

Element	Type	Filter/Sort	Description
hostName	string		The Densify internal reference used to ensure that containers are uniquely identified across multiple clusters, even if they potentially have the same namespace, pod and/or container names.
displayName	string		The display name is a combination of the pod and container name.
entityId	string		The Densify assigned entity ID of the container system.
href	string		The referenced resource to the Densify system entity representing this container. See Systems on page 408 for details of the `/systems`

Element	Type	Filter/Sort	Description
			resource.

Examples

Example: Listing All Systems in a Kubernetes Container Analysis

The following example shows you how to return all systems (container) included in a Kubernetes container optimization analysis.

Example: Listing all Systems in a Kubernetes Container Analysis

Request:

```
GET /analysis/containers/kubernetes/d7298ac3-a143-41bb-b7d7-62f659f2a8c5/systems
```

Response:

```
[
   {
      "hostName": "d3cf25f4-01b1-4c0b-66a8-86bd1ea771d3",
      "displayName": "kube-abcserver-kube-ctrl.densify.com__kube-abcserver",
      "entityId": "06770825-988a-4aa4-b047-41bda995a69e",
      "href": "/systems/06770825-988a-4aa4-b047-41bda995a69e"
   },
   {
      "hostName": "e9422b59-97ef-40b1-b589-587a23a761d3",
      "displayName": "kube-scheduler-kube-ctrl.densify.com__kube-scheduler"
      "entityId": "1f7981f5-dd90-416b-88f9-67f032188536",
      "href": "/systems/1f7981f5-dd90-416b-88f9-67f032188536"
   },
   ...
]
```

Analysis: Policy

Description

The following resources are used to return a list of Densify platform-specific cloud policies available for analysis:

- `/analysis/cloud/aws/policy`
- `/analysis/cloud/gcp/policy`
- `/analysis/cloud/azure/policy`
- `/analysis/containers/kubernetes/policy`

Policies are used to define the business and technical behaviors and requirements for optimization. Densify optimization analysis utilizes the platform-specific default policy employing industry best practices. You can use the returned policies from the platform-specific `/policy` resource to override the default policy used during an optimization analysis. Contact Densify support (Support@Densify.com) to review your policy settings and to adjust or enable additional cloud policies for your environment.

DevOps Policies

The following cloud policies are available to the DevOps service tier:

- **DevOps-Default**—The default policy reflects best practices for generating actionable recommendations. Specifically:
  - The resource utilization of each system is modeled using a minimum of 7 days and up to 60 days of historical workload.

- When optimizing instance sizes and families, the predicted CPU and memory usage levels must not exceed 70% and 90%, respectively.
- When memory usage metrics are not available, the analysis uses the existing memory allocation of the instance and will not change the memory configuration.

DevOps-Automation—The automation is intended for generating recommendations that require little or no review before being implemented. Densify utilizes rule-driven analytics to predict the effort of changing instance from the current to the recommended type, and this policy favors low-effort recommendations, resulting in increased automation at the expense of reduced cost savings. Specifically:

- The resource utilization of each system is modeled using a minimum of 7 days and up to 90 days of historical workload.
- When optimizing instance sizes and families, the predicted CPU and memory usage must not exceed 65% and 85%, respectively.
- This policy will not specify burstable (T-series) instance families for workloads unless they are already running in a burstable family and will not change CPU Architecture from Intel to AMD or vice versa.
- When memory usage metrics are not available, the analysis uses the existing memory allocation of the instance and will not change the memory configuration.

DevOps-Efficiency—The efficiency policy is intended for generating recommendations that provide maximum operational efficiency (and lowest operating cost). This policy also includes recommendations rated as moderate effort required , generating higher cost reduction than the DevOps default policy and so recommendations should be reviewed before implementing.

- The resource utilization of each system is modeled using a minimum of 3 day and up to 30 days of historical workload.
- When optimizing instance sizes and families, the predicted CPU and memory usage levels must not exceed 75% and 95%, respectively.
- When memory usage metrics are not available, the analysis uses the existing memory allocation of the instance and will not change the memory configuration.

You can also verify the specific details of the cloud policy used during an analysis through the Densify user interface, see *Cloud Optimization Policies* in the topic *Viewing Policy Settings*(Help Topic ID 120200) for details.

Resource

```
/analysis/cloud/aws/policy
/analysis/cloud/gcp/policy
/analysis/cloud/azure/policy
/analysis/containers/kubernetes/policy
```

Supported Operations

Table: Policy Supported Operations

HTTP Method	Input	Output	Description
`GET /analysis/cloud/aws/policy`	Query String Parameter: details=true	policyId policyInstanceId name description (*optional*)	Used to return a list of AWS policies available for analysis purposes. Example: List All the AWS Cloud Policies Available Example: Listing AWS Cloud Policies with Descriptions
`GET /analysis/cloud/gcp/policy`	Query String Parameter: details=true	policyId policyInstanceId name description (*optional*)	Used to return a list of GCP policies used for analysis. Example: Listing GCP Cloud Policies with Descriptions
`GET /analysis/cloud/azure/policy`	Query String Parameter: details=true	policyId policyInstanceId name description (*optional*)	Used to return a list of Azure policies used for analysis.
`GET /analysis/containers/kubernetes/policy`	Query String Parameter: details=true	policyId policyInstanceId name description (*optional*)	Used to return a list of Kubernetes Container policies available for analysis purposes. ***Note***: *DevOps policies are not supported for Container analysis. Contact support (support@densify.com) to review your policy settings for Container resource optimization.*

Parameters

Query String Parameters

Table: Analysis Policy Query String Parameters

Parameter Name	Type	Description
details=true *(optional)*	string	This option returns additional policy description details.

Response

Table: Analysis Policy Response Schema

Element	Type	Sort By	Filter	Description
policyId	string			The analysis policy type ID.
policyInstanceId	string			The entity ID of the policy instance.
name	string			The policy name.
description *(optional)*	string			A description of the policy. This element is only returned when the `?details=true` option is added to the request.

Examples

Example: List All the AWS Cloud Policies Available

The following example shows you how to list all the available AWS cloud policies.

Example: Listing AWS Cloud Policies

Request:

```
GET /analysis/cloud/aws/policy
```

Response:

```
[
   {
     "policyId": "4a63f651-a583-4157-97ff-35651370ffbe",
     "policyInstanceId": "69fa4c99-1be2-4048-94a7-36fd83d07f37",
     "name": "DevOps-Default"
   },
   {
     "policyId": "4a63f651-a583-4157-97ff-35651370ffbe",
     "policyInstanceId": "0c0ef18b-9367-4071-b733-396f63e51925",
     "name": "DevOps-Automation"
   },
   {
     "policyId": "4a63f651-a583-4157-97ff-35651370ffbe",
     "policyInstanceId": "70a2ef4a-2ebb-4209-8ec8-9f6c70f77a74",
     "name": "DevOps-Efficiency"
   }
]
```

Example: Listing AWS Cloud Policies with Descriptions

The following example shows you a request to list all the available AWS cloud policies with description.

Example: Listing AWS Cloud Policies with Description

Request:

```
GET /analysis/cloud/aws/policy?details=true
```

Response:

```
[
   {
     "policyId": "4a63f651-a583-4157-97ff-35651370ffbe",
     "policyInstanceId": "0c0ef18b-9367-4071-b733-396f63e51925",
     "name": "DevOps-Automation",
     "description": "This policy is intended for generating instance sizing and instance fami
ization recommendations that require little or no review before being implemented.  CLOE util
rule-driven analytics to predict the effort of changing instance type from current to recomme
and this policy favors Low effort recommendations, producing higher automation at the expense
lower cost savings.\r\nThe resource utilization of each system is modeled using a minimum of
and up to 90 days of historical workload.\r\nWhen optimizing instance sizes and families, the
dicted CPU and memory usage must not exceed 65% and 85%, respectively.\r\nThis policy will no
cify burstable (T-series) instance families for workloads unless they are already running in
burstable family and will not change CPU Architecture from Intel to AMD or vice versa.\r\nWhe
usage metrics are not available, the analysis assumes the existing memory allocation is requi
will not change the memory configuration."
   },
   {
     "policyId": "4a63f651-a583-4157-97ff-35651370ffbe",
     "policyInstanceId": "69fa4c99-1be2-4048-94a7-36fd83d07f37",
     "name": "DevOps-Default",
     "description": "This policy reflects best practices for generating instance sizing and i
family optimization recommendations.\r\nThe resource utilization of each system is modeled us
minimum of 7 days and up to 60 days of historical workload.\r\nWhen optimizing instance sizes
ilies, the predicted CPU and memory usage levels must not exceed 70% and 90%, respectively.\r
memory usage metrics are not available, the analysis effectively assumes the existing memory
ation of the instance is required and will not change the memory configuration."
   }
]
```

Example: Listing GCP Cloud Policies with Descriptions

The following example shows you a request to list all the available GCP cloud policies with description.

Example: Listing GCP Cloud Policies with Description

Request:

```
GET /analysis/cloud/gcp/policy?details=true
```

Response:

```
[
   {
     "policyId": "4fe32fc6-6067-4647-8a87-9bd1dc74389e",
     "policyInstanceId": "43636c78-851b-4f25-8733-5debbbb6856b",
     "name": "DevOps-Automation",
     "description": "This policy is intended for generating instance sizing and instance fami
ization recommendations that require little or no review before being implemented.  CLOE util
rule-driven analytics to predict the effort of changing instance type from current to recomme
and this policy favors Low effort recommendations, producing higher automation at the expense
lower cost savings.\r\nThe resource utilization of each system is modeled using a minimum of
and up to 90 days of historical workload.\r\nWhen optimizing instance sizes and families, the
dicted CPU and memory usage must not exceed 65% and 85%, respectively.\r\nThis policy will no
cify burstable (T-series) instance families for workloads unless they are already running in
burstable family and will not change CPU Architecture from Intel to AMD or vice versa.\r\nWhe
usage metrics are not available, the analysis assumes the existing memory allocation is requi
will not change the memory configuration."
   },
   {
     "policyId": "4fe32fc6-6067-4647-8a87-9bd1dc74389e",
     "policyInstanceId": "6131f154-8453-48c3-9747-edb327ec0bed",
     "name": "DevOps-Efficiency",
     "description": "This policy is intended for generating instance sizing and family optimi
that provide maximum operational efficiency (and lowest operating cost).  This policy also in
recommendations rated as Moderate effort, generating higher cost reduction than the Densify I
policy.  Because of this, any recommendations should be reviewed before implementing.\r\nThe
utilization of each system is modeled using a minimum of 3 day and up to 30 days of historica
load.\r\nWhen optimizing instance sizes and families, the predicted CPU and memory usage leve
not exceed 75% and 95%, respectively.\r\nWhen memory usage metrics are not available, the ana
effectively assumes the existing memory allocation of the instance is required and will not c
the memory configuration."
   },
   {
     "policyId": "4fe32fc6-6067-4647-8a87-9bd1dc74389e",
     "policyInstanceId": "9b8cb5db-5045-4dbe-a6ee-740038c6b0b0",
     "name": "DevOps-Default",
     "description": "This policy reflects best practices for generating instance sizing and i
family optimization recommendations.\r\nThe resource utilization of each system is modeled us
minimum of 7 days and up to 60 days of historical workload.\r\nWhen optimizing instance sizes
ilies, the predicted CPU and memory usage levels must not exceed 70% and 90%, respectively.\r
memory usage metrics are not available, the analysis effectively assumes the existing memory
ation of the instance is required and will not change the memory configuration."
   }
]
```

Analysis: Status

Description

The `/analysis/<platformType>/<platformSubType>/<analysisId>/status` resource is used to return the current status of an analysis in Densify.

Resource

```
/analysis/<platformType>/<platformSubType>/<analysisId>/status
```

Supported Operations

Table: Analysis Status Supported Operations

Operation	HTTP Method	Input	Output	Description
Check for analysis status	`GET /analysis/<platformType>/<platformSubType>/<analysisId>/status`	Path Parameter: platformType platformSubType analysisId	analysisStage webHookStatus statusMessage For errors: message status	Use this resource to check the status of an analysis by providing the analysis ID. Example: Checking for Kubernetes Container Analysis Status Example: Checking for AWS Analysis Status

Operation	HTTP Method	Input	Output	Description
				Example: Checking for GCP Analysis Status

Parameters

Path Parameters

Table: Analysis Status Path Parameters

Parameter Name	Type	Description
platformType	string	[cloud|containers] The technology platform of the analysis.
platformSubType	string	The platform sub-type of the analysis. This is typically the vendor/flavor of the technology platform. Possible platform sub-types depend on the platformType specified: - For platformType = cloud, platformSubType = [aws|gcp|azure] - For platformType = containers, platformSubType = [kubernetes]
analysisId	string	The unique referenced ID of the analysis.

Response

Table: Analysis Status Response Schema

Element	Type	Filter/Sort	Description
analysisStage	string		The current stage of the specified analysis. Possible stages include: - Analyzing : {*percent*} Completed - Not analyzed - Completed
webHookStatus	string		The status of the last webhook POST request. Recommendations are pushed to the webhook URI via a POST request after analysis completion.
statusMessage	string		The message for the analysis status.
message	string		For errors, the message for the status response is returned.
status	number		The HTTP response code of the request error. Possible status values include:

Element	Type	Filter/Sort	Description

			200—success with request;
			400—invalid parameters;
			401—authentication failed;
			404—resource not found;
			500—internal server error.

Examples

Example: Checking for Kubernetes Container Analysis Status

The following example shows you how to check for a specific Kubernetes container analysis status.

Example: Checking for Kubernetes Container Analysis Status

Request:

```
GET /analysis/containers/kubernetes/6e4da559-cbd8-40b8-5498-644be6843a93/status
```

Response:

```
{
  "analysisStage": "Completed",
  "webHookStatus": "",
  "statusMessage": "Analysis ABC-east-665-b was last completed on Wed Jan 30
18:08:45 EST 2019."
}
```

Example: Checking for AWS Analysis Status

The following example shows you how to check for a specific AWS analysis status.

Example: Checking for AWS Analysis Status

Request:

```
GET /analysis/cloud/aws/9a5d2d55-6d85-4fde-8bab-fcd0cef8c5bf/status
```

Response:

```
{
  "analysisStage": "Completed",
  "webHookStatus": "",
  "message": "Analysis 231345225455 was last completed on Mon Jan 21 10:11:09
EST 2019."
}
```

Example: Checking for Azure Analysis Status

The following example shows you how to check for a specific Azure analysis status.

Example: Checking for Azure Analysis Status

Request:

```
GET /analysis/cloud/azure/6b8ab8e1-a026-4db9-a4ae-2d684446731f/status
```

Response:

```
{
  "analysisStage": "Completed",
  "webHookStatus": "Success",
  "statusMessage": "Analysis 00d89cbc-bc00-4d00-bcf6-ce6ec09d8adc was last
completed on Thu Jan 24 09:36:36 EST 2019."
}
```

Example: Checking for GCP Analysis Status

The following example shows you how to check for a specific GCP analysis status.

Example: Checking for GCP Analysis Status

Request:

```
GET /analysis/cloud/gcp/0930c31d-13ac-4a3f-892e-2a35c8a3f842/status
```

Response:

```
{
  "analysisStage": "Analyzing : 38% Completed",
  "webHookStatus": "",
  "statusMessage": "Analysis gcpAcctEast-608378 is currently running."
}
```

Analysis: Webhook

Description

The `/webhook/analysis` resource is used to manage the webhook definition for a Densify analysis. The webhook definition is the path and authentication to a third-party application, where recommendation results are sent when the analysis completes. The receiving application can process the resulting data for downstream consumption. For example, the receiving application can route recommendations to an orchestration engine or distribute reports to instance owners.

You have the option to define the webhook when you initiate data collection and analysis with the `/analysis/<platformType>/<platformSubType>/analyze` request. See Run AWS data collection and analysis on page 95 for an example of specifying webhook input parameters. Results are posted to the webhook location when the analysis is complete. You also have the ability to update or delete a webhook if you no longer want to push results to the webhook.

Resource

```
/webhook/analysis
```

Supported Operations

Table: Webhook Supported Operations

Operation	HTTP Method	Input	Output	Description
List all analyses webhooks for a specific platform type	`GET /webhook/analysis/<platformType>`	Path Parameter: platformType	Collection of: analysisId analysisName href uri webHookStatus	Use this resource to list all analysis webhook definitions for a particular platform type in Densify. Example: Retrieving All Cloud Webhook Definitions Example: Retrieving All Container Webhook Definitions
Get a webhook definition for an analysis	`GET /webhook/analysis/<platformType>/<platformSubType>/<analysisId>`	Path Parameters: platformType platformSubType analysisId	analysisId analysisName href uri webHookStatus	Use this resource to return the webhook details of a specific analysis in Densify. Example: Getting a Webhook Definition for an AWS Analysis

Operation	HTTP Method	Input	Output	Description
Add webhook to an analysis	`POST /webhook/analysis/<platformType>/<platformSubType>/<analysisId>`	Path Parameters: platformType platformSubType analysisId Request Body Parameters: uri authType authValue	message status	Use this resource to add a webhook definition to an existing analysis in Densify. You can only add a webhook to an analysis without a webhook. To update a webhook, use the `PUT` method. See Update a webhook for an analysis for details. Example: Adding a Webhook to an Existing AWS Analysis
Update a webhook for an analysis	`PUT /webhook/analysis/<platformType>/<platformSubType>/<analysisId>`	Path Parameters: platformType platformSubType analysisId Request Body Parameters: uri authType authValue	message status	Use this resource to update a webhook definition of an existing analysis in Densify. Example: Updating an

Operation	HTTP Method	Input	Output	Description
				AWS Analysis Webhook
Delete a webhook from an analysis	`DELETE /webhook/analysis/<platformType>/<platformSubType>/<analysisId>`	Path Parameters: - platformType - platformSubType - analysisId	- message - status	Use this resource to delete a webhook definition from an existing analysis. Example: Deleting a Webhook from an AWS Analysis

Parameters

Path Parameters

Table: Webhook Path Parameters

Parameter Name	Type	Description
platformType	string	[cloud|containers] The technology platform of the analysis.
platformSubType	string	The platform sub-type of the analysis. This is typically the vendor/flavor of the technology platform. Possible platform sub-types depend on the platformType specified: - For platformType = cloud, platformSubType = [aws|gcp|azure] - For platformType = containers, platformSubType = [kubernetes]
analysisId	string	The referenced ID of the analysis.

Request Body Parameters

Table: Webhook Request Body Parameters

Parameter Name	Type	Description
uri	string	The URI to the webhook application. ***Note:*** *You must ensure that the webhook URI is accessible from the Densify system with a fully qualified domain name or an IP address.*
authType	string	The authorization type (i.e. Basic, Bearer) for the webhook application.
authValue	string	The authorization value (i.e. username, password, token) for the webhook application. Densify assumes that the authorization value is in standard Base64 encoding, but if a colon (" `:` ") is present in the value, then Densify assumes that the value is in plain text with `username:password` format.

Response

Table: Webhook Response Schema

Element	Type	Sort By/Filter	Description
analysisId	string		The Densify assigned ID for the analysis entity.
href	string		The analysis webhook resource reference.
uri	string		The webhook URI.
webHookStatus	string		The status of the last webhook POST request. Recommendations are pushed to the webhook URI via a POST request after analysis completion.
analysisName	string		The name assigned to the analysis. The default analysis name is the platform-specific cluster, account, project, or subscription ID.
message	string		The message for the status response. For example, the following messages are returned for the error codes below: - 400—"Analysis webhook already exists." - 404—"Analysis webhook not found."
status	number		The HTTP response code of the request. Possible status values include: - 200—success with webhook request; - 400—object already exists or invalid parameters; - 401—authentication failed;

Element	Type	Sort By/Filter	Description
			404—resource not found.

Examples

Example: Retrieving All Cloud Webhook Definitions

The following example shows you how to list all cloud webhook definitions in Densify.

Example: Listing All Cloud Webhook Definitions

Request:

```
GET /webhook/analysis/cloud
```

Response:

```
[
   {
      "analysisId": "f73e13ca-09d3-4395-8e32-894669a6d1eb",
      "analysisName": "551345225429"
      "href": "/webhook/analysis/cloud/aws/f73e13ca-09d3-4395-8e32-
894669a6d1eb",
      "uri": "https://webhookdensify:443/results/display/",
      "webHookStatus": "success"
   },
   {
      "analysisId": "329959e9-3641-4677-a665-dff754943944",
      "analysisName": "330076083786"
      "href": "/webhook/analysis/cloud/aws/329959e9-3641-4677-a665-
dff754943944",
      "uri": "https://webhooknow:443/api/results/test",
      "webHookStatus": "An error occured executing POST request"
   }
]
```

Example: Retrieving All Container Webhook Definitions

The following example shows you how to list all container webhook definitions in Densify.

Example: Listing All Container Webhook Definitions

Request:

```
GET /webhook/analysis/containers
```

Response:

```
[
   {
      "analysisId": "e8298ac3-a143-41bb-b7d7-62f659f43ec5",
      "analysisName": "kube-prod-341ops",
      "href": "/webhook/analysis/containers/kubernetes/e8298ac3-a143-41bb-b7d7-
62f659f43ec5",
      "uri": "https://webhookdensify:443/results/display/",
      "webHookStatus": "success"
   },
   {
      "analysisId": "d32959e9-3641-4677-a665-dff754943945",
      "analysisName": "kube-dev-543",
      "href": "/webhook/analysis/containers/kubernetes/d32959e9-3641-4677-a665-
dff754943945",
      "uri": "https://mycontainerresources:443/api/results/",
      "webHookStatus": "An error occured executing POST request"
   }
]
```

Example: Getting a Webhook Definition for an AWS Analysis

The following example shows you how to return the details of a webhook definition for an AWS analysis.

Example:Retrieving Webhook Definition for an AWS Analysis

Request:

```
GET /webhook/analysis/cloud/aws/9a5d2d55-6d85-4fde-8bab-fcd0cef8c5bf
```

Response:

```
{
  "analysisId": "9a5d2d55-6d85-4fde-8bab-fcd0cef8c5bf",
  "analysisName": "775525148221",
  "href": "/webhook/analysis/cloud/aws/9a5d2d55-6d85-4fde-8bab-fcd0cef8c5bf",
  "uri": "https://Dwebhooks:443/api/test/webhook",
  "webHookStatus": "success"
}
```

Example: Adding a Webhook to an Existing AWS Analysis

The following example shows you how to add a webhook definition to an existing AWS analysis without a webhook.

Example: Adding a Webhook Definition to an Existing AWS Analysis

Request:

```
POST /webhook/analysis/cloud/aws/9a5d2d55-6d85-4fde-8bab-fcd0cef8c5bf
{
  "uri": "https://mycallbackServer:443/api/test/webhook",
  "authType": "basic",
  "authValue": "saasUser:password1"
}
```

Response:

```
{
  "message": "ok",
  "status": 200
}
```

Example: Updating an AWS Analysis Webhook

The following example shows you how to update a webhook definition for an AWS analysis.

Example:Updating a Webhook Definition for an AWS Analysis

Request:

```
PUT /webhook/analysis/cloud/aws/9a5d2d55-6d85-4fde-8bab-fcd0cef8c5bf
{
  "uri": "https://myOtherServer:443/api/webhook",
  "authType": "bearer ",
  "authValue":
"eyJhbGciOiJIUzUxMiJ9.eyJqdGkiOiI0MjRhMGziOC0xMDRlopRiMjItOeg0ZS1lZTJhNmEzZDBlN
GIiLCJpYXQiOjE1NDM2MDg1OTMsInN1YiI6ImFkbWluIiwiaXNzIjoiRGVuc2lmeS5jb20iLCJleHAi
OjE1NDM2MDg4OTN9.F9VgD918C6WufBajr0ezLd61T6d9ij8z4BmHFzfNmMqCS-
9JTDDaxfPmQVZDjeSTo0C-dYWcllwPTcMSUfRnYQ"
}
```

Response:

```
{
  "message": "ok",
  "status": 200
}
```

Example: Deleting a Webhook from an AWS Analysis

The following example shows you how to delete a webhook definition from an AWS analysis.

Example:Deleting a Webhook Definition

Request:

```
DELETE /webhook/analysis/cloud/aws/329959e9-3641-4677-a665-dff754943944
```

Response:

```
{
  "message": "ok",
  "status": 200
}
```

Authorize

Description

The `authorize` resource is used to obtain a JSON Web Token (JWT) for Densify users to make authorized API requests. The resource will return a token for any active Densify user when the proper credentials are provided. Only API-enabled users can make authorized API calls. See Using the API: Authentication on page 26 for details on API-enabled users.

The JWT API token is only valid within the expiry time of the token, which by default is set to 5 minutes.

The key used to validate the token is also refreshed every 30 days.

Security Considerations

The authorization workflow provides a progressively longer delay each time an invalid password is entered in an Authorize request.

This behaviour is applied regardless of whether or not you have enabled the Densify password policy. The workflow does change if the password policy has been enabled. Contact Support@Densify.com for details.

Contact Support@Densify.com for details on configuring the Densify password policy.

Resource

```
/authorize
```

Supported Operations

Table: Authorize Supported Operations

Operation	HTTP Method	Input	Output	Description
Obtain a JWT API token	`POST /authorize`	Request Body Parameters: - userName - pwd	- apiToken - expires - status	Used to return an API token for an active Densify user. Example: Successful Authorize Example: Unauthorized

Parameters

Request Body Parameters

Table: Authorize Request Body Parameters

Parameter Name	Type	Description
userName	string	The username of an active Densify user account. For API access, the Densify user must be part of the Administrator or SaaS_User user group. See Using the API: Authentication on page 26 for details on API-enabled users. The `/authorize` resource will return a token for any active Densify user with proper credentials provided, however, only tokens for API-enabled users can be used to make authorized API calls.
pwd	string	The corresponding password for userName.

***Note**: Densify highly recommends using an SSL web service to ensure that user credentials are encrypted. Contact Support@Densify.com for details.*

Response

The following is a list of possible response elements that are returned for the `/authorize` resource. If authorization failed, two elements are returned:

- message
- status

Table: Authorize Response Schema

Element	Type	Sort By	Filter	Description
apiToken	string			The returned token used to make subsequent authorized API calls. The API token follows the JSON Web Token (JWT), RFC 7519 standard. Specify the **Bearer** authorization type with the token in the header of subsequent API requests: `Authorization: Bearer <apiToken>` See Example: Using Authorize JWT Token.
expires	number			The date and time (in milliseconds) when the apiToken expires.
status	number			The HTTP response code of the request. Possible status values include: - 200—successful response; - 400—the payload is null or invalid (i.e. userName or pwd is empty or invalid); - 401—authentication failed (e.g. user does not exist, incorrect password, or user account is locked). - 403—forbidden access. The trial or subscription has expired. - 429—requests are too frequent. - 500—internal server error.
message	string			The message for an error status response.

***Note**: Other response status and error messages could indicate issues with the Densify web server or connectivity issues to the web server.*

Examples

Example: Successful Authorize

The following example shows you how to obtain an API token using the `apiUser` account.

Example: Successful Authorize

Request:

```
POST /authorize
{
    "userName": "apiUser",
    "pwd": "apiPassword"
}
```

Response:

```
{
  "apiToken": "eyJh-
bGciOiJIUzUxMiJ9.eyJqdGkiOiIzNzI2Yzk0NC0wMmE4LTRlYzQtOGE2Ny04ODBmMDM2OTRhZD-
ciLCJpYXQiOjE1NDI2NTI0MDUsInN1YiI6InZh-
biIsIm-
lzcyI6IkRlb-
nNpZnkuY29tIiwiZXhwIjoxNTQyNjUyNzA1fQ.cJd8qFJfRoPnMEU7GzcdYGBT8WwlgmviQ1OQp8P_
w9VUcjQA3FJaB9QkqJJ6d7zbrY5yjc4w0rOWjY-PPdbmqw",
  "expires" : 1542652705869,
  "status" : 200
}
```

Example: Unauthorized

The following example shows an authorize request with invalid user credentials.

Example: Unauthorized

Request:

```
POST /authorize
{
    "userName": "APIUser",
    "pwd": "wrongPassword"
}
```

Response:

```
{
  "message" : "Unauthorized",
  "status" : 401
}
```

Example: Using Authorize JWT Token

The following example shows you how to use the API token obtained from the `/authorize` resource to make an authorized API request. In this example, an API request is made to list all cloud analyses in

Densify using the JWT token for authorization, which is passed to the header as a Bearer token authorization type.

Example: Using JWT API token to Make Authorized Request to List All Cloud Analyses

Request:

```
GET /analysis/cloud
```

Headers:

```
Accept: application/json
Authorization: Bearer eyJh-
bGciOiJIUzUxMiJ9.eyJqdGkiOiIzNzI2Yzk0NC0wMmE4LTRlYzQtOGE2Ny04ODBmMDM2OTRhZD-
ciLCJpYXQiOjE1NDI2NTI0MDUsInN1YiI6InZh-
biIsIm-
lzcyI6IkRlb-
nNpZnkuY29tIiwiZXhwIjoxNTQyNjUyNzA1fQ.cJd8qFJfRoPnMEU7GzcdYGBT8WwlgmviQ1OQp8P_
w9VUcjQA3FJaB9QkqJJ6d7zbrY5yjc4w0rOWjY-PPdbmqw
```

***Note**: The long JWT token string is taken from the `apiToken` element of an `/authorize` request. This example shows the raw token string, but typical implementations would save the API token as a variable and pass it to the request header. See Example: Returning AWS Recommendations with Low Effort in Terraform-map Form.*

Bookings

Description

A Booking represents an inbound or outbound host, an inbound datastore, or an inbound or outbound VM.

A Booking object is automatically created with a specified type element when the corresponding resource object is created through the API or UI:

Table: Inbound and Outbound Bookings

booking	API	UI
type=INBOUND_GUEST	workloads	Route and Reserve Demand page or though the Booking Manager
type=INBOUND_HOST	inbound-hosts	Plan and Manage Supply page or though the Booking Manager
type=INBOUND_DATSTORE	inbound-datastores	Plan and Manage Supply page or though the Booking Manager
type=OUTBOUND_HOST	outbound-hosts	Control Console Spectrum

The Booking objects are kept in synch with their API counterparts such that any updates made to the API objects are automatically made to the Booking objects, and vice-versa.

Note*: Only the Booking object is updated on analysis refreshes, with state updates.*

Resource

```
/bookings
```

Supported Operations

Table: Bookings Supported Operations

Operation	HTTP Method	Input	Output	Description
Get Collection	GET /bookings	None	Booking collection of [id, name, href]	Retrieve a collection of Bookings. Filter-Metadata is supported. Note that it returns all status and type values of the collection, filtered or not, as if the collection was not filtered. Filters and Collection Details are supported. In addition, the option `?details=true&attributes_flag=setonly` returns all the resource elements for each booking, including additional set attributes defined outside of the cfg\bookings\bookings-config.xml file. To only return set attributes in specific categories, add the `setdisplay_category` option. See Systems: Attribute Display Categories on page 413 for details. Attribute filtering support is also enabled. See Systems: Attribute Filters - Multiple Values Support on page 414 for more details on using attribute filters. Note that the `attributes_flag=setonly` option only works in conjunction with the `details=true` option. Sort By and Paging are not supported. Example: Getting a Collection of Bookings on page 232 Example: Getting a Collection of Bookings Details and Attributes on page 233
Get	GET	None	Bookings:	Retrieve the Booking elements of the specified

Operation	HTTP Method	Input	Output	Description
Individual	/bookings/<*id*>		Resource Elements on page 226	id. Example: Getting an Individual Inbound Guest Booking on page 234
Get Individual Attributes	GET /bookings/<*id*>/attributes	None	Bookings: Resource Elements on page 226	Retrieve the Booking attribute elements of the specified id.
Modify Individual Attributes	PUT /bookings/<*id*>/attributes	[name, value] of the attributes in Bookings: Resource Elements on page 226	None	Modify the values of the Booking's attributes array. The name corresponds to that in the attributes array and the value is the new value. The behavior is the same as modifying the attributes of a Workload. For multi-value attributes (i.e. "Multiple Values" property is enabled for the attribute), a PUT request for that attribute will append the new attribute value to the attribute array if that attribute name,value pair does not exist. To overwrite a mutli-value attribute, you will first need to delete the existing attribute value. Example: Modifying a Booking's Attributes on page 236
Modify to Completed	PUT /bookings/<*id*>	{"status", "COMPLETED"}	Bookings: Resource Elements on page 226	Modify the status of a VM Booking from "COMMITTED"/"LATE" to "COMPLETED" status. This update only applies to a VM booking routed to a non-control hosting venue. Example: Completing a Booking on page 237
Delete Attributes	DELETE /bookings/<*id*>/attributes/	[id, value] Where value is only required to delete the attribute value	None	Delete attribute settings (if just the id is specified) or attribute values (if id and value are specified) from the specified Booking and its associated Workload. The id corresponds to an attribute id in the attributes array. This request is valid in any state. If the attribute id or value does not exist, then that specific delete is simply ignored. For multi-value attributes (i.e. "Multiple Values" property is enabled for the attribute), a DELETE request with only id provided will delete all the values with that attribute ID; a

Operation	HTTP Method	Input	Output	Description

				DELETE request with `id,value` provided will delete only the attribute ID entry with that specific value.
				If any deleted item fails to delete, then no delete is performed at all.
				For example:
				`[`
				`   { "id": "attr_2", "value": "Capps"},`
				`   { "id": "state_power"}`
				`]`
				Example: Deleting a Booking's Attributes on page 237

***Note** : You cannot delete an individual Booking; however, a Booking is removed when the associated Workload is deleted. See Delete Individual Workload on page 540 for details on deleting a Workload.*

Resource Elements

Table: Booking Resource Elements

Element	Type	Create/ Mod- (Req)	Filter	Description
id, name, href	strings		F by `name`, `name_like` **Note**: Using Filter-Metadata returns `system_name`	See ID, Name and Self Reference (id, name, href) on page 29. To filter bookings with names that contain a given input string, use `?name_like="<substring>"` in your collection request. This filter is case insensitive. The '%' character can be used to match zero or more characters. The `name` is the expected system name of the Booking.
attributes	[id, name, value]	M	F using ▪ `attribute_name_like` ▪ `attribute_id` ▪ `attribute_name` ▪ `attribute_id_`	Same as the attributes definition for the corresponding Workload. The option `?details=true&attributes_flag=setonly` in the bookings collection

Element	Type	Create/ Mod- (Req)	Filter	Description
			`value` • `attribute_name_value`	request returns host(VM) and datastore bookings information which includes additional set attributes that are available outside of the cfg\bookings\bookings-config.xml definition. To only return set attributes in specific categories, use the `setdisplay_category` option. See Systems: Attribute Display Categories on page 413 for details. To filter based on attributes, you must use one of the following element options — in each case, bookings data will be returned if and only if the attributes selected have values assigned to them: • `attribute_name_like`—returns a collection with an attribute name containing a sub-string; use the '%' character to match zero or more characters • `attribute_id` with a provided list of attribute IDs—returns a collection with a specific attribute ID that has its value set • `attribute_name` with a provided list of attribute names—returns a collection with a specific attribute name that has its value set • `attribute_id_value` with a provided list of attribute ID-value pairs—returns a collection with attribute ID-value pairs which match the set of attribute IDs and corresponding values provided • `attribute_name_value` with a provided list of attribute name-value pairs—returns a collection with attribute name-value pairs which match the set of attribute names and corresponding values provided Refer to Systems: Attribute Filters - Multiple Values Support on page 414 for a description of how to use these attribute filters.
control_ environment	id, name, href, icon		F using • `control_environment_id` • `control_environment_id_list`	A link to the Control Environment for this Booking. If this is an inbound booking in `DRAFT` status, this link may not be present. Note that when filtering on Control Environment, you must use the element name

Element	Type	Create/ Mod- (Req)	Filter	Description
				`control_environment_id` along with its UUID or `control_environment_id_list` with a list of UUIDs specified.
creation_time	number		F	The UTC date and time the Booking was created.
description	string	M		Same as that defined for the corresponding Workload.
expected_date	number	M		The UTC date the Booking is expected to be completed.
host	string			The recommended host where this workload should be placed. This element is returned when the `type` is "`INBOUND_GUEST`" and the `status` is not "`DRAFT`" or "`PENDING`".
infrastructure_group	id, name, href		F using ▪ `infrastructure_group` ▪ `infrastructure_group_list`	A link to the Infrastructure Group for this Booking. If this is an inbound booking in `DRAFT` status, this link may not be present. Note that when filtering on Infrastructure Groups, you must use the element `infrastructure_group` with a name specified or `infrastructure_group_list` with a list of names specified. Does not apply to datastore bookings. This information is within the `sensor` element.
late_days	number			To allow for a late incoming workload, this is the number of days to hold the reservation after the `expected_date`. This element is defaulted to that defined by configuration setting **Default Number of Days to Hold a Booking Reservation after the Planned Start Date** (parameter key default.num.days.to.hold.booking.reservation) of category **25. Advanced - Booking Reservation Settings**. Specify 0 to define no grace period. For details on late bookings, see section *Late Bookings* of *Booking Overview* (Help Topic ID 230350).
number_days_to_expiry	number			The number of remaining days before expiry (which is calculated as `expected_date` - today's date + `late_days`). This element is returned only when the Booking is

Element	Type	Create/ Mod- (Req)	Filter	Description
				"COMMITTED". This element is updated on every analysis refresh.
owner	string	M		Same as that defined for the corresponding Workload.
owner_email	string		F	Same as that defined for the corresponding Workload.
project	string	M	F	Same as that defined for the corresponding Workload.
status	string		F Note: `filter-metadata` returns `status_value` with all 7 possible states	The booking state: - "`DRAFT`"–not included yet in a Routing Request - "`PENDING`"–"Pending Commit" waiting to be reserved in the analysis on the next environment or infrastructure group refresh in the Control Console - "`COMMITTED`"–included in the analysis - "`COMPLETED`"–matching inbound booking reconciled - "`LATE`"–no matching incoming booking by expected date, but still within the <n> days of grace (`late_days` > 0) - "`EXPIRED`"–no matching incoming booking by extended expected date (including the <n> days of grace defined by `late_days`) - "`CANCELLED`"–booking has been deleted
system	Properties for Host System on page 230, or Properties for Guest System on page 231		F only on `os` Note: `filter-metadata` returns `os` at top level	The expected system name of the host or guest with extra properties for each type. Does not apply for a datastore booking.
transform_ analysis_ model	string			The analysis model that was used in the transform scenario.
transform_ scenario_ name	string			The name of the transform scenario that created the Booking.
type	string		F Note: `filter-`	The booking type: - "`INBOUND_GUEST`"

Element	Type	Create/ Mod- (Req)	Filter	Description
			metadata returns type_value	- "INBOUND_DATASTORE" - "INBOUND_HOST" - "OUTBOUND_GUEST" - "OUTBOUND_HOST"
update_date	number		F	The UTC date and time the Booking was last modified (or created).
workloadId	string			The ID of the corresponding Workload object. Only returned if the Booking is of "type": "INBOUND-GUEST".

Properties for Host System

Property	Type	Filter	Description
id	string		The UUID of the created server model for this host.
name	string		The expected system name of the host.
catalog_id	id	F	Catalog Reference id.
catalog_spec	string	F	Catalog Reference Name.
cpu_allocation	number		CPU Allocation of the system referenced by the catalog_spec. -1 if not applicable.

disks	[name, attributes[], provisioned_space, used_space]		An array of disk requirements, with sizes and tier capabilities. One disk is defined by default. When modifying this array, you must specify all disks as the new array replaces the existing one. Defaulted to that of the associated catalog_spec, if defined.
			Each disk is defined as follows:
			- name—name of the disk
			- provisioned_space—provisioned space in MB
			- used_space—used space in MB
			- attributes: [id, name, value]—id, name and value of the datastore attributes. The id is mapped to its display name (e.g. "attr_DatastoreTier" is mapped to "Datastores") and can be determined by performing GET /sensors. An example of id, name and value:
			`{`
			`  "id": "attr_DatastoreTier",`
			`  "name": "Datastore Tier",`
			`  "value": "Gold"`
			`}`

model_name	string		Model of the system referenced by the catalog_spec.
os	string	F	Operating System Name of the system referenced by the catalog_spec.

Property	Type	Filter	Description
			Filtering is specified simply as `os=<value>` in the collection request.
sensor	[id, name, type, href, hostname]		The recommended Sensor object placement. This element is returned when the `type` is "`INBOUND_GUEST`" and the `status` is not "`DRAFT`" or "`PENDING`". If there are no sensors, the empty list is returned. - `id`–sensor id - `name`–name of the sensor - `type`–sensor type (e.g. "datastore") - `href`–sensor href - `hostname`–display name of the sensor For Sensor element details, see Sensors including Datastores, Physical Storage, Resource Pools: Resource Elements on page 403.
total_cpu	number		Total Logical CPUs of the system referenced by the `catalog_spec`.
total_memory	number		Normalized Total Memory (MB) of the system referenced by the `catalog_spec`.

Properties for Guest System

Property	Type	Filter	Description
id	string		The UUID of the created server model for this guest.
name	string		The expected system name of the VM.
catalog_id	id	F	Catalog Reference id.
catalog_spec	string	F	Catalog Reference Name.
cpu_entitlement	number		CPU Entitlement/eCPU of the system referenced by the `catalog_spec`.

disks	[name, attributes[], provisioned_space, used_space]
	[name, provisioned_space, used_space, pref_datastore, attributes[]]
			An array of disk requirements, with sizes and tier capabilities. One disk is defined by default. When modifying this array, you must specify all disks as the new array replaces the existing one. Defaulted to that of the associated `catalog_spec`, if defined.
			Each disk is defined as follows:
			- `name`—name of the disk
			- `provisioned_space`—provisioned space in MB
			- `used_space`—used space in MB
			- `attributes: [id, name, value]`—id, name and value of the datastore attributes. The `id` is mapped to its display name (e.g. "attr_DatastoreTier" is mapped to "Datastores") and can be determined by performing `GET /sensors`. An example of id, name and value:
			`{`
			`  "id": "attr_DatastoreTier",`
			`  "name": "Datastore Tier",`
			`  "value": "Gold"`

Property	Type	Filter	Description
			`}` - `pref_datastore`—preferred datastore for the disk (optional: only returned if set) See Assessing Datastore on page 348 for datastore placement details.
memory	number		Memory in MB. Normalized Total Memory (MB) of the system referenced by the `catalog_spec`.
number_of_disks	number	F	The number of disks defined in the `disks` array. This element is only used for filtering.
os	string	F	Operating System Name of the system referenced by the `catalog_spec`. Filtering is specified simply as `os=<value>` in the collection request.
provisioned_space	number		The sum of the provisioned space for the disks within the Booking (i.e. sum of the `provisioned_space` in MB of the `disk` array).
sensor	[id, name, type, href, hostname]		The recommended Sensor object placement. This element is returned when the `type` is "`INBOUND_GUEST`" and the `status` is not "`DRAFT`" or "`PENDING`". If there are no sensors, the empty list is returned. - `id`–sensor id - `name`–name of the sensor - `type`–sensor type (e.g. "datastore") - `href`–sensor href - `hostname`–display name of the sensor For Sensor element details, see Sensors including Datastores, Physical Storage, Resource Pools: Resource Elements on page 403.
used_space	number		The sum of the used space for the disks within the Booking (i.e. sum of the `used_space` in MB of the `disk` array).
vcpu	number		vCPUs. Total Logical CPUs of the system referenced by the catalog_spec.
workload_profile	string		Workload Profile.

Examples

Example: Getting a Collection of Bookings

The following example shows you how to obtain the collection of current Bookings.

Example: Getting a Collection of Bookings

Request:

```
GET /bookings
```

Response:

```
[
  {
    "id": "b03bc809-5a3c-4061-a829-2fea4353af37",
    "name": "infosys-java53",
    "href": "/bookings/b03bc809-5a3c-4061-a829-2fea4353af37"
  },
  {
    "id": "9e6826c8-79f9-4c26-bbe5-11eae2773f2e",
    "name": "win-vm-2337",
    "href": "/bookings/9e6826c8-79f9-4c26-bbe5-11eae2773f2e"
  },
  {
    "id": "5805b3a5-b126-48b3-81af-6dab230a0348",
    "name": "finance-java4",
    "href": "/bookings/5805b3a5-b126-48b3-81af-6dab230a0348"
  },
  // ... *SNIP* ...
]
```

Example: Getting a Collection of Bookings Details and Attributes

The following example shows you how to obtain the collection of current Bookings with details, including all the available set attributes.

Example: Getting a Collection of Bookings Details and Attributes

Request:

```
GET /bookings/?details=true&attributes_flag=setonly
```

Response:

```
[
  {
    "id": "15ff670a-6b55-48dc-be97-a9b20f781ba9",
    "name": "00_host1",
    "href": "/bookings/15ff670a-6b55-48dc-be97-a9b20f781ba9",
    "infrastructure_group": {
        "id": "15dfe947-627e-4e84-a091-6b9f9bb4b3d5",
        "name": "Cluster 4",
        "href": "/infrastructure-groups/15dfe947-627e-4e84-a091-6b9f9bb4b3d5"
    },
    "control_environment": {
        "id": "b5d6e04f-47bc-4802-996e-f94831b75ffa",
        "name": "vc55",
        "href": "/control-environments/b5d6e04f-47bc-4802-996e-f94831b75ffa",
        "icon": "/control-environments/b5d6e04f-47bc-4802-996e-
f94831b75ffa/icon"
    },
    "system": {
        "sensors": null,
        "name": "00_host1",
        "id": "e4a583f6-d49e-46a6-8645-7aac82be2d3e",
        "os": "N/A",
        "disks": [],
        "catalog_spec": "ucs_b230_m1_x7560_2x8x256",
        "catalog_id": 01462efc-0e56-40bc-8c90-46de8a7e39ee,
        "total_cpu": 16,
        "cpu_allocation": 16,
        "total_memory": 262144,
        "model_name": "UCS B230 M1"
    },
    "type": "INBOUND_HOST",
    "status": "COMMITTED",
    "expected_date": 1535688000000,
    "creation_time": 1532404800000,
    "update_date": 1532446340487,
    "owner": "admin",
    "owner_email": "",
    "project": "__Unknown__",
    "description": "",
    "attributes": {
        "id": "attr_2",
        "name": "Department",
        "value": "Department 222"
    },
    {
        "id": "attr_booking_type",
        "name": "Booking Type",
        "value": "Active Host Booking"
    },
    {
        "id": "start_date",
        "name": "Booking Start Date",
        "value": "2018-08-31"
    },
    {
        "id": "VE_LOCATION_CLUSTER",
        "name": "Virtual Cluster",
        "value": "Cluster 4"
    },
    // ... *SNIP* ... remaining set attributes are displayed
    "number_days_to_expiry": 39
  },
  {
    "id": "9e6826c8-79f9-4c26-bbe5-11eae2773f2e",
    "name": "win-vm-2337",
    "href": "/bookings/9e6826c8-79f9-4c26-bbe5-11eae2773f2e"
    // ... *SNIP* ... all resource elements and attributs are returned for
    // ... *SNIP* ... each active booking
]
```

Example: Getting an Individual Inbound Guest Booking

The following example shows you how to retrieve the details of a single booking.

Example: Getting an Individual Inbound Guest Booking

Request:

```
GET /bookings/eab2122d-1ebf-4199-8762-4960480c3842
```

Response:

```
{
  "id": "eab2122d-1ebf-4199-8762-4960480c3842",
  "name": "infosys-java53",
  "system": {
    "name": "infosys-java53",
    "id": "f1726542-a019-4feb-aab2-4423e9fc1e4f",
    "os": "Linux"
    "vcpu": 4,
    "cpu_entitlement": 4,
    "memory": 8192,
    "disks": [
      {
        "name": "SYSTEM",
        "provisioned_space": 81920,
        "used_space": 20480,
        "pref_datastore": "netappsds01",
        "attributes": [
          {
            "id": "attr_DiskDatastoreLink",
            "name": "Disk Datastore Link",
            "value": "0215f8e1-d9f2-4055-a9d4-8e94e2de39c8"
          },
          {
            "id": "attr_DatastoreTier",
            "name": "Datastore Tier",
            "value": "Gold"
          }
        ]
      },
      // ... *SNIP* of other disks ...
    ],
    "provisioned_space": 163840,
    "used_space": 46080,
    "catalog_spec": "lin-small-2gb",
    "catalog_id": "2cb0102e-6d38-4188-b4b5-115e111a96ac",
    "workload_profile": "Medium_Utilization",
    "sensors": [
      {
        "id": "9e935663-9761-278f-bb14-2307f0f6f563",
        "name": "Esxcrb12 Datastore",
        "type": "datastore",
        "href": "/sensors/datastores/9e935663-9761-278f-bb14-2307f0f6f563",
        "host_name": "0215f8e1-d9f2-4055-a9d4-8e94e2de39c8"
      }
    ]
  },
  "type": "INBOUND_GUEST",
  "status": "COMMITTED",
  "owner": "Bill",
  "project": "Infosys_DB",
  "description": "",
  "attributes": [
    {
      "id": "attr_2",
      "name": "Department",
      "value": "3HT"
    },
    {
      "id": "attr_SecurityZone",
      "name": "Security Zone",
      "value": "Level-1-Data"
    },
    {
      "id": "attr_Workload_Profile",
      "name": "Workload_Profile",
      "value": "Medium_Utilization"
    }
  ],
  "host": "esxcrb12.int.Densify.com",
  "href": "/bookings/eab2122d-1ebf-4199-8762-4960480c3842",
  "infrastructure_group": {
    "id": "adbea101-dab2-4253-9a86-690865fac4f7",
    "name":"Eng-Dev2",
    "href": "/infrastructure-groups/adbea101-dab2-4253-9a86-690865fac4f7"
  },
  "control_environment": {
    "id": "7f8fbeaf-3b70-4560-bdbc-94c030a2184a",
    "name":"Boston",
    "href": "/control-environments/7f8fbeaf-3b70-4560-bdbc-94c030a2184a",
    "icon": "/control-environments/7f8fbeaf-3b70-4560-bdbc-94c030a2184a/icon"
  },
  "expected_date": 1386866967197,
  "creation_date": 1386866967197,
  "update_date": 1386867159243,
  "owner_email": "",
  "late_days": 7,
  "number_days_to_expiry": 7,
  "workloadId": "fd5a5545-2598-4ed1-a2ee-e0e05277f4ea"
}
```

Example: Getting the Booking Filter Metadata for a Project in Windows

The following example shows you how to get booking filter metadata for a specified project in a specified operating system.

Example: Getting the Booking Filter Metadata for a Project in Windows

Request:

```
GET /booking-filter-metadata/?project=ProjectX&os=Windows
```

Example: Modifying a Booking’s Attributes

The following example shows you how to modify booking attributes.

Example: Modifying a Booking’s Attributes

Request:

```
PUT /bookings/03117467-9432-4702-9c1a-1fee90431295/attributes
[
  {"name": "Department", "value": "3HT"}
]
```

Response:

```
[
  {
    "id": "attr_2",
    "name": "Department",
    "value": "3HT"
  },
  // ... *SNIP* of other attributes ...
]
```

Example: Deleting a Booking’s Attributes

The two example requests below provide the same result, in this single-valued use case.

Example: Deleting a Booking’s Attributes

Request:

```
DELETE /bookings/03117467-9432-4702-9c1a-1fee90431295/attributes
[
  {"id": "attr_2", "value": "3HT"}
]
```

Request:

```
DELETE /bookings/03117467-9432-4702-9c1a-1fee90431295/attributes
[
  {"id": "attr_2"}
]
```

Example: Completing a Booking

The example below moves a VM Booking from "COMMITTED"/"LATE" status to "COMPLETED" status. This update only applies to a VM booking routed to a non-control hosting venue.

Example: Completing a Booking

Request:

```
PUT /bookings/03117467-9432-4702-9c1a-1fee90431295
{"status": "COMPLETED"}
```

Bookings Project, Status Groups

Description

Bookings can be grouped by project or status so that they can be ordered in a specific priority. To obtain a collection of one of these groups, simply append the group name (i.e. `projects` or `status`) to your Booking collection request (e.g. to list Bookings by project, use `/bookings/projects/?details=true`). If the project element has not been defined, then "`__Unknown__`" can be used to query with no project name. The collection by group is returned in alphabetical order by name, with "`__Unknown__`" objects at the end.

When performing a collection by group, the following options are supported:

- Collection Details—to obtain the Booking details per group object.
- Paging—to page the group collection. Note that the Bookings are not part of the paging size.
- Filters—to filter the Bookings using the filtering elements of the Booking (except for the grouping elements themselves). If there are qualifying Bookings, the collection returns the group and their qualifying Bookings (if `&details=true`). Otherwise, if there are no qualifying Bookings for a given group, the group is not returned.

Resource

```
/bookings/projects
/bookings/status
```

Supported Operations

Table: Project/Status Supported Operations

Operation	HTTP Method	Input	Output	Description
Get Collection	GET /bookings/projects	None	Booking Project collection of [name, href]	The list of projects defined by all Bookings is returned, with "`__Unknown__`" projects last. Default Sort By is defined as: `?sort_by=name`. Example: Getting a Collection of EXPIRED Booking Projects on page 241, Example: Getting a Detailed Collection of EXPIRED Bookings by Project on page 241
	GET /bookings/status	None	Booking Status collection of [name, href]	The list of status values defined by all Bookings is returned. Default Sort By is defined as: `?sort_by=name` Example: Getting a Collection of Booking Status on page 240
Get Individual	GET /bookings/projects/<*name*>	None	name, href, [Bookings: Resource Elements on page 226]	The details of all Bookings defined with the specified project name <*name*> (case insensitive).
	GET /bookings/status/<*name*>	None	name, href, [Bookings: Resource Elements on page 226]	The details of all Bookings that are in the specified status <*name*> (case insensitive).

Resource Elements

Table: Project/Status Resource Elements

Element	Type	Description
name, href	strings	Name of and link to, the Project/Owner/Status.

Element	Type	Description
bookings	[Bookings: Resource Elements on page 226]	Array of Bookings grouped by the Project/Status. The elements that are returned are the same as those returned when performing a GET request on a Booking.

Examples

Example: Getting a Collection of Booking Status

This query retrieves all Status values that are defined by the existing Booking objects.

Note that this request does not return ids, as Booking Status objects are not instantiated objects.

Example: Getting a Collection of Booking Status

Request:

```
GET /bookings/status
```

Response:

```
[
  {
    "name": "COMMITTED",
    "href": "/bookings/status/COMMITTED"
  },
  {
    "name": "DRAFT",
    "href": "/bookings/status/DRAFT"
  },
  {
    "name": "EXPIRED",
    "href": "/bookings/status/EXPIRED"
  },
  {
    "name": "LATE",
    "href": "/bookings/status/LATE"
  },
  {
    "name": "PENDING",
    "href": "/bookings/status/PENDING"
  }
]
```

Example: Getting a Collection of EXPIRED Booking Projects

This query retrieves all Projects that include Bookings with `status=EXPIRED`.

***Note**: This request does not return ids, as Booking Projects are not instantiated objects.*

Example: Getting a Collection of EXPIRED Booking Projects

Request:

```
GET /bookings/projects/?status=EXPIRED
```

Response:

```
[
  {
    "name": "Cloud Wave 1",
    "href": "/bookings/projects/Cloud Wave 1?status=EXPIRED"
  },
  {
    "name": "Consolidation W1",
    "href": "/bookings/projects/Consolidation W1?status=EXPIRED"
  }
]
```

Example: Getting a Detailed Collection of EXPIRED Bookings by Project

This query retrieves all Booking details, with `status=EXPIRED`, grouped by Project.

Note that this request does not return ids, as Workload Projects are not instantiated objects.

Example: Getting a Detailed Collection of EXPIRED Bookings by Project

Request:

```
GET /bookings/projects/?details=true&status=EXPIRED
```

Response:

```
[
  {
    "name": "Cloud Wave 1",
    "href": "/bookings/projects/Cloud Wave 1?status=EXPIRED",
    "bookings": [
      {
        "id": "73cf6cf0-9af0-4373-97ca-95fad7d034ba",
        "name": "win-phys-11-cm",
        // ... *SNIP* of Booking elements ...
      },
      // ... *SNIP* of other Bookings in this Project ...
    ]
  },
  // ... *SNIP* of other Projects and their Bookings ...
]
```

Catalog Specifications

Description

A Catalog Specification represents a pre-configured specification for a new guest or host. All Catalog Specifications shipped with Densify are defined with a medium utilization Workload Profile and one disk definition.

When you create a Workload, you specify the Catalog Specification using element `catalog_spec`.

Manufacturer, OS Groups

The Catalog Specification collection is currently grouped by manufacturer and operating system. To obtain this list of supported groups, simply perform a collection request using `/catalog-spec-groups`.

To obtain the group collection, simply append the group name (i.e. `manufacturer` or `os`) to your Catalog Specification Group collection request (e.g. use `/catalog-spec-groups/os` to obtain the list of operating systems). The group collection is returned in alphabetical order by name.

To obtain the list of Catalog Specifications for a specific manufacturer or operating system, use `/catalog-spec-groups/<group>/<group name>` (e.g. `/catalog-spec-groups/os/AIX` to obtain the Catalog Specifications supporting AIX).

Resource

```
/catalog-specs
```

```
/catalog-spec-groups
/catalog-spec-groups/manufacturer
/catalog-spec-groups/os
```

Supported Operations

Table: Catalog Specification Supported Operations

Operation	HTTP Method	Input	Output	Description
Get Collection	GET /catalog-specs	None	Catalog Specification Collection of [id, name, href]	Returns the Catalog Specifications. Example: Getting a Collection of Catalog Specification Groups on page 248
Get Individual	GET /catalog-specs/<*id*>	None	Catalog Specifications: Resource Elements on page 245	Returns the Catalog Specification. Example: Getting an Individual Catalog Specification on page 247

Table: Manufacturer/OS Supported Operations

Operation	HTTP Method	Input	Output	Description
Get Collection	GET /catalog-spec-groups/	None	Group collection of [id, name, group, href]	The list of Catalog Specification groups. Currently, only `os` and `manufacturer` are supported. Sort By is not supported.
Get Collection	GET /catalog-spec-groups/os/	None	OS collection of [id, name, group, href]	The list of OSs, in alphabetical order, defined by all Catalog Specifications. Sort By is not supported. Example: Getting a Collection of OS Catalog Specification for Guests on page 248
	GET /catalog-spec-groups/manufacturer/	None	Manufacturer collection of [id, name, group, href]	The list of manufacturers, in alphabetical order, defined by all Catalog Specifications. Sort By is not supported.
Get Individual	GET /catalog-spec-groups/os/<*id*>	None	id, name, group, href, systems[id, name, href]	The list of Catalog Specifications with the specified OS.
	Get /catalog-spec-groups/manufacturer/<*id*>	None	id, name, group, href, systems[id, name, href]	The list of Catalog Specifications with the specified manufacturer. Example: Getting an Individual Manufacturer

Operation	HTTP Method	Input	Output	Description
				Catalog Specification on page 250

Resource Elements

Table: Catalog Specification Resource Elements

Element	Type	Filter	Description
id, name, href	strings	F by `name`	See ID, Name and Self Reference (id, name, href) on page 29.
attributes	[id, name, value]		Defines any other attributes of the Catalog Specification.
cores_per_cpu	string	F	The Cores per CPU.
cpu_benchmarks	[name, score_type, value]		The default CPU benchmark. - `name`—Label with possible values: • `"CINT2000"` • `"CINT2000 Rate"` • `"CINT2006 Rate"` • `"RPE2"` - `score_type`—Score type with possible values: • `"cint2000"` • `"cint2000rate"` • `"cint2006rate"` • `"rpe2"` - `value`
cpu_entitlement	number	F	The CPU Entitlement (Proc Units).
cpu_model	string	F	The CPU Architecture.
cpu_speed	string	F	The Normalized CPU Speed (MHz).
cpus	number	F	The number of CPUs in the specification.
disks	[name, attributes[], provisioned_space, used_space]	F	An array of disk requirements, with sizes and tier capabilities. One disk is defined by default. When modifying this array, you must specify all disks as the new array replaces the existing one. Defaulted to that of the associated `catalog_spec`, if defined. Each disk is defined as follows: - `name`—name of the disk - `provisioned_space`—provisioned space in MB - `used_space`—used space in MB - `attributes: [id, name, value]`—id, name and value of the datastore attributes. The `id` is mapped to its display name (e.g. "attr_DatastoreTier" is mapped to "Datastores") and can be determined by performing `GET /sensors`. An example of id, name

Element	Type	Filter	Description

			and value:
			`{`
			`  "id": "attr_DatastoreTier",`
			`  "name": "Datastore Tier",`
			`  "value": "Gold"`
			`}`

I/O_benchmarks	[name, value, score_type]		The list of all available I/O benchmarks. - `name`—Label with possible values: • `"Maximum Disk Throughput (bytes/s)"` • `"Maximum Network Throughput (bytes/s)"` - `score_type`—Score type with possible values: • `"disk"` • `"net"` - `value`—A value of -1 means there is no value specified.
manufacturer	string	F	The manufacturer, e.g. "Cisco".
memory	number	F	The Total Memory (MB).
os	string	F	The Operating System Name.
platform_model	string		The Model.
provisioned_space	string	F	The Provisioned Space (GB). The sum of the provisioned space for the disks within the specification (i.e. sum of the `provisioned_space` in MB of the `disk` array).
threads_per_core	string		The Threads per Core.
total_physical_cpus	string		The Total Physical CPUs.
type	string	F	The Catalog Specification type: - `"GUEST"` - `"HOST"`
used_space	string	F	The Used Space (GB). The sum of the used space for the disks within the specification (i.e. sum of the `used_space` in MB of the `disk` array).
workload_profile	string		The Workload Profile.

Table: Manufacturer/OS Resource Elements

Element	Type	Filter	Description
id, name, href	strings	F by `name`	See ID, Name and Self Reference (id, name, href) on page 29.
group	string	F	Type of group: - `"manufacturer"` - `"os"`
systems	[id, name, href]		The list of Catalog Specifications that have the specified group, e.g. with an os of `"AIX"`.

Examples

Example: Getting an Individual Catalog Specification

This example retrieves an individual Catalog Specification.

Example: Getting an Individual Catalog Specification

Request:

```
GET /catalog-specs/53a234b0-90d2-44ee-ab54-bc06934b2c27
```

Response:

```
{
  "id": "53a234b0-90d2-44ee-ab54-bc06934b2c27",
  "name": "lin-small-2gb",
  "type": "GUEST",
  "cpus": 1,
  "os": "Linux",
  "manufacturer": "Bookings Catalog",
  "disks": [
    {
      "name": "SYSTEM",
      "attributes": [],
      "provisioned_space": 81920,
      "used_space": 20480
      ]
    }
  ],
  "attributes": [
    {
      "id": "attr_IPAddressesAssigned",
      "values": [
        "1"
      ]
    }
  ],
  "href": "/catalog-specs/53a234b0-90d2-44ee-ab54-bc06934b2c27",
  "cpu_entitlement": 1,
  "memory": 2048,
  "provisioned_space": 81920,
  "used_space": 20480,
  "workload_profile": "Medium_Utilization",
  "platform_model": "Small 2GB",
  "cpu_model": "Intel Xeon E5-2670",
  "cpu_speed": 2600,
  "total_physical_cpus": 1,
  "cores_per_cpu": 1,
  "threads_per_core": 1,
  "cpu_benchmarks": [
    {
      "name": "CINT2006 Rate",
      "value": 39.5
      "score_type": "cint2006rate",
    }
  ]
  "I/O_benchmarks": [
    {
      "name": "Maximum Disk Throughput (bytes/s)",
      "value": -1,
      "score_type": "disk",
    },
    {
      "name": "Maximum Network Throughput (bytes/s)",
      "value": -1,
      "score_type": "net",
    },
}
```

Example: Getting a Collection of Catalog Specification Groups

The following example shows you how to get a collection of catalog specification groups.

Example: Getting a Collection of Catalog Specification Groups

Request:

```
GET /catalog-spec-groups
```

Response:

```
[
  {
    "id": "os",
    "name": "os",
    "href": "/catalog-spec-groups/os"
  },
  {
    "id": "manufacturer",
    "name": "manufacturer",
    "href": "/catalog-spec-groups/manufacturer"
  }
]
```

Example: Getting a Collection of OS Catalog Specification for Guests

This example retrieves the collection of OS catalog specifications based on guest operating systems. `?type=guest` is the default so no extra filtering is required.

Example: Getting a Collection of OS Catalog Specification for Guests

Request:

```
GET /catalog-spec-groups/os
```

Response:

```
[
  {
    "id": "AIX",
    "name": "AIX",
    "group": "os",
    "type": "GUEST",
    "href": "/catalog-spec-groups/os/AIX"
  },
  {
    "id": "Linux",
    "name": "Linux",
    "group": "os",
    "type": "GUEST",
    "href": "/catalog-spec-groups/os/Linux"
  },
  {
    "id": "Windows",
    "name": "Windows",
    "group": "os",
    "type": "GUEST",
    "href": "/catalog-spec-groups/os/Windows"
  }
]
```

Example: Getting a Collection of Manufacturer Catalog Specifications for Hosts

This example retrieves the Manufacturer Catalog Specifications for hosts.

Example: Getting Collection of Manufacturer Catalog Specifications for Hosts

Request:

```
GET /catalog-spec-groups/manufacturer/?type=HOST
```

Response:

```
[
  {
    "id": "Cisco",
    "name": "Cisco",
    "group": "manufacturer",
    "type": "HOST",
    "href": "/catalog-spec-groups/manufacturer/Cisco"
  },
  {
    "id": "Dell",
    "name": "Dell",
    "group": "manufacturer",
    "type": "HOST",
    "href": "/catalog-spec-groups/manufacturer/Dell"
  },
    // ... *SNIP* of Manufacturer Catalog Specifications ...
  }
]
```

Example: Getting an Individual Manufacturer Catalog Specification

This example retrieves the Dell Manufacturer Catalog Specification.

Example: Getting an Individual Manufacturer Catalog Specification

Request:

```
GET /catalog-spec-groups/manufacturer/Dell
```

Response:

```
{
  "id": "Dell",
  "name": "Dell",
  "group": "manufacturer",
  "href": "/catalog-spec-groups/manufacturer/Dell",
  "systems": [
    {
      "id": "6f08aff2-7561-42ea-9488-c5e32ddec005",
      "name": "bkl-db-fin1",
      "href": "/catalog-specs/6f08aff2-7561-42ea-9488-c5e32ddec005"
    },
    // ... *SNIP* of Systems...
  ]
}
```

Configuration Parameters

Description

This resource is used to return all the API configuration settings, which are defined in the Analysis Console under **Administration** > **Densify Configuration**, category **API and Report Settings**. For details, see Configuration Settings on page 27.

Resource

```
/configuration-parameters
```

Supported Operations

Table: Configuration Parameters Supported Operations

Operation	HTTP Method	Input	Output	Description
Get Collection	GET /configuration-parameters	None	Array of Configuration Parameters: Resource Elements on page 252	Returns the configuration parameters applicable to the API. Example: Getting the Configuration Parameters on page 252

Resource Elements

Table: Configuration Parameters Resource Elements

Element	Type	Description
key	string	The parameter key.
name	string	The parameter name displayed in the Densify Configuration screen.
value	string	The parameter value.

Examples

Example: Getting the Configuration Parameters

The following example shows you how to get the API configuration parameters.

Example: Getting the Configuration Parameters

Request:

```
GET /configuration-parameters
```

Response:

```
[
  {
    "key": "rest.api.catalogSpec.default",
    "name": "API Default Catalog Specification",
    "value": "win-medium-4gb"
  },
  {
    "key": "rest.api.WithinHoursFromLastRefreshOfCluster",
    "name": "API Last Hours of Cluster Refresh",
    "value": "48"
  },
  {
    "key": "rest.api.logging",
    "name": "API Detailed Logging",
    "value": "OFF"
  },
  {
    "key": "rest.api.paging.pageSize",
    "name": "API Page Size",
    "value": "100.0"
  },
  {
    "key": "rest.api.routingStrategy.default",
    "name": "API Default Routing Strategy",
    "value": "capacity_sensitive"
  },
  {
    "key": "baseline.vm.for.spare.vm.capacity.reports",
    "name": "Baseline VM Name",
    "value": "win-medium-4gb"
  }
]
```

Control Environments

Description

A Control Environment represents a collection of Infrastructure Groups, which are hosting venues to route workloads. The term Control Environment refers to all three environment types, for the purpose of this guide, of either full control hosting venues, non-control hosting venues or guest-level hosting venues.

With this API, the details of Control Environments can be obtained so that Workloads can be routed appropriately. The API provides stats at each defined timeframe of the timeline.

Hosting Venues

This API supports all environments, of either full control hosting venues, non-control hosting venues or guest-level hosting venues. For details on these types of venues, see *Hosting Venues* (Help Topic ID 290200).

With respect to this API, the difference between these venues is as follows:

- Supported Operations—all operations are supported for all three
- Resource Elements—only elements that are applicable to the environment is returned in the response. Specifically:
  - for non-control and guest-level environments, elements `cei`, `total_guests`, `total_hosts`, `required_hosts`, `outbound_guest_bookings`, `inbound_host_bookings` and `outbound_host_bookings` are not returned
  - for non-control environments, element `last_refreshed_time` is not returned

Resource

```
/control-environments
```

Supported Operations

Table: Control Environment Supported Operations

Operation	HTTP Method	Input	Output	Description
Get Collection	GET /control-environments	None	Control Environment collection of [id, name, platform, platform_category, control_type, href, icon]	Default Sort By is defined as: `?sort_by=name`. Example: Getting a Collection of Control Environments on page 257
Get Individual	GET /control-environments/<*id*>	None	Control Environments: Resource Elements on page 255	Retrieve the Control Environment elements of the specified `id`. Example: Getting an Individual Control Environment on page 258

Resource Elements

Table: Control Environment Resource Elements

Element	Type	Sort By	Filter	Description
id, name, href	strings	S by `name`	F by `name`	See ID, Name and Self Reference (id, name, href) on page 29.
availability_status	string	S	F	Whether or not the Control Environment is available or not for routing. • "`AVAILABLE`"–when all Infrastructure Groups of the Control Environment are either "`AVAILABLE`" or "`BLOCKED_BY_USER`", when the Control Environment has been refreshed at least once • "`UNAVAILABLE`"–when the Control Environment has never been refreshed, the `last_refreshed_time` is older in number of hours than `rest.api.WithinHoursFromLastRefreshOfCluster`, an Infrastructure Groups is "`UNAVAILABLE`"

Element	Type	Sort By	Filter	Description
control_type	string		F	The type of control analysis support. - "`FULL`"—fully supported, e.g. for VMware environments - "`GUEST_LEVEL`"—guest-level hosting venue - "`NONE`"—no support, e.g. for external cloud
creation_time	number	S	F	The UTC date and time the Control Environment was created.
icon	href			Link to the icon associated with this Control Environment.
infrastructure_groups	[id, name, platform, platform_category, href, hardware_name, hardware_icon, control_type, color_code_selected, color_code_unselected]			Links to all Infrastructure Groups associated with this Control Environment (including disqualified ones).

last_refreshed_time	number	S
		sort by `last_refresh_time`
			F
			filter by `last_refresh_time` with UTC values
				The UTC date and time the Control Environment itself was last refreshed.
				If the Control Environment has never been refreshed, the following is returned:
				`"last_refreshed_time": 0`
				Filtering by range with a value of 0 is not supported. e.g. the following does not return all Control Environments that have never been refreshed:
				`?last_refresh_time_from=0&`
				`   last_refresh_time_to=0`

platform	string		F	The system type being analyzed. For examples: - "`IBM`" - "`HYPERV`" - "`VMWARE`" - "`XENSERVER`" - "`X86`"
platform_category	string		F	The type of platform. Possible categories are: - "`Internal Virtual`"—for control environments, e.g.

Element	Type	Sort By	Filter	Description
				for a VMware environment whether or not fully controlled through analytics - "`Internal Physical`"—for physical environments, e.g. a Bare Metal - "`External Cloud`"—for external cloud environments, e.g. an AWS environment - "`External Physical`"—for external physical environments, e.g. SoftLayer Bare Metal
stats	Complex, as specified in the Descriptio n			The overall statistics of the Control Environment at each timeframe. For each timeframe that has data: - `cei` - `timeline_name`—same as `long_name` of the Timeline Tags on page 526 - `date`—same as `date` of the Timeline Tags on page 526 - `total_guests` - `total_hosts` - `required_hosts` - `inbound_guest_bookings` - `outbound_guest_bookings` - `inbound_host_bookings` - `outbound_host_bookings` If the Control Environment has never been refreshed or is currently being refreshed (even for a specific timeframe or specific Infrastructure Group), the following is returned: `"stats": {}` ***Note****: The above counts do not include Infrastructure Groups that have been disqualified.*

Examples

Example: Getting a Collection of Control Environments

The following example shows you how to obtain the collection of current Control Console environments:

Example: Getting a Collection of Control Environments

Request:

```
GET /control-environments
```

Response:

```
[
  {
    "id": "16579260-f236-44a1-94e5-55e9e6ef6773",
    "name": "Chicago",
    "platform": "VMWARE",
    "platform_category": "Internal Virtual",
    "control_type": "FULL"
    "href": "/control-environments/16579260-f236-44a1-94e5-55e9e6ef6773"
    "icon": "/control-environments/16579260-f236-44a1-94e5-55e9e6ef6773/icon"
  },
  {
    "id": "56fee3a1-f327-46e4-9ff1-9d365964b824",
    "name": "Boston",
    "platform": "VMWARE",
    "platform_category": "Internal Virtual",
    "control_type": "FULL"
    "href": "/control-environments/56fee3a1-f327-46e4-9ff1-9d365964b824"
    "icon": "/control-environments/56fee3a1-f327-46e4-9ff1-9d365964b824/icon"
  },
  // ... *SNIP* of Control Environments...
]
```

Example: Getting an Individual Control Environment

The following example shows you how to retrieve a single Control Console environment:

Example: Getting an Individual Control Environment

Request:

```
GET /control-environments/56fee3a1-f327-46e4-9ff1-9d365964b824
```

Response:

```
{
  "id": "56fee3a1-f327-46e4-9ff1-9d365964b824",
  "name": "Boston",
  "platform": "VMWARE",
  "platform_category": "Internal Virtual",
  "control_type": "FULL"
  "stats": {
    "TO": {
      "cei": 0.79,
      "timeline_name": "Today",
      "date": 1365480000000,
      "total_guests": 273,
      "total_hosts": 28,
      "required_hosts": 17,
      "inbound_guest_bookings": 0,
      "outbound_guest_bookings": 0,
      "inbound_host_bookings": 0,
      "outbound_host_bookings": 0
    },
    "90D": {
      "cei": 0.89,
      "timeline_name": "90 Days",
      "date": 1373256000000,
      "total_guests": 294,
      "total_hosts": 28,
      "required_hosts": 0,
      "inbound_guest_bookings": 31,
      "outbound_guest_bookings": 10,
      "inbound_host_bookings": 0,
      "outbound_host_bookings": 0
    },
    "30D": {
      "cei": 0.86,
      "timeline_name": "30 Days",
      "date": 1368072000000,
      "total_guests": 291,
      "total_hosts": 28,
      "required_hosts": 0,
      "inbound_guest_bookings": 18,
      "outbound_guest_bookings": 0,
      "inbound_host_bookings": 0,
      "outbound_host_bookings": 0
    },
    "60D": {
      "cei": 0.93,
      "timeline_name": "60 Days",
      "date": 1370664000000,
      "total_guests": 304,
      "total_hosts": 28,
      "required_hosts": 0,
      "inbound_guest_bookings": 31,
      "outbound_guest_bookings": 0,
      "inbound_host_bookings": 0,
      "outbound_host_bookings": 0
    },
    "7D": {
      "cei": 0.79,
      "timeline_name": "7 Days",
      "date": 1366084800000,
      "total_guests": 273,
      "total_hosts": 28,
      "required_hosts": 0,
      "inbound_guest_bookings": 0,
      "outbound_guest_bookings": 0,
      "inbound_host_bookings": 0,
      "outbound_host_bookings": 0
    }
  },
  "href": "/control-environments/56fee3a1-f327-46e4-9ff1-9d365964b824",
  "availability_status": "AVAILABLE",
  "creation_time": 1365777303697,
  "last_refreshed_time": 1371232491487,
  "infrastructure_groups": [
    {
      "id": "11673bee-6e02-436c-afc5-64692a517e08",
      "name": "Bos-Eng&Dev",
      "platform": "VMWARE",
      "platform_category": "Internal Virtual",
      "href": "/infrastructure-groups/11673bee-6e02-436c-afc5-64692a517e08",
      "hardware_name": "PowerEdge",
      "hardware_icon": http://thedocs1:8086/CIRBA/images/serverModelIcon/Dell 
small.png
```

```
      "control_type": "FULL"
      "color_code_selected": "#AACBEC"
      "color_code_unselected": "#DCE5F4"
    },
    {
      "id": "2f29ee1f-0039-47cd-ac34-da927c12705d",
      "name": "Bos-ProdApps2",
      "platform": "VMWARE",
      "platform_category": "Internal Virtual",
      "href": "/infrastructure-groups/2f29ee1f-0039-47cd-ac34-da927c12705d",
      "hardware_name": "PowerEdge",
      "hardware_icon": http://thedocs1:8086/CIRBA/images/serverModelIcon/Dell_
small.png
      "control_type": "FULL"
      "color_code_selected": "#AACBEC"
      "color_code_unselected": "#DCE5F4"
    },
    {
      "id": "5f46fd3f-2df3-4bb6-9a90-f1f245023bc4",
      "name": "Bos-ProdApps1",
      "platform": "VMWARE",
      "platform_category": "Internal Virtual",
      "href": "/infrastructure-groups/5f46fd3f-2df3-4bb6-9a90-f1f245023bc4",
      "hardware_name": "PowerEdge",
      "hardware_icon": http://thedocs1:8086/CIRBA/images/serverModelIcon/Dell_
small.png
      "control_type": "FULL"
      "color_code_selected": "#AACBEC"
      "color_code_unselected": "#DCE5F4"
    },
    {
      "id": "eabe7ff5-bc8a-4b84-8417-cabf8f3209ac",
      "name": "Bos-GenApps",
      "platform": "VMWARE",
      "platform_category": "Internal Virtual",
      "href": "/infrastructure-groups/eabe7ff5-bc8a-4b84-8417-cabf8f3209ac",
      "hardware_name": "PowerEdge",
      "hardware_icon": http://thedocs1:8086/CIRBA/images/serverModelIcon/Dell_
small.png
      "control_type": "FULL"
      "color_code_selected": "#AACBEC"
      "color_code_unselected": "#DCE5F4"
    }
  ],
  "icon": "/control-environments/e6cf1672-77ff-4e7d-9dda-5387b0bc95cc/icon"
}
```

Existing Systems

Description

This resource is used to return the host and guest systems within Densify. This list matches that returned by the Data Collection > DSE in the Analysis Console, but only systems that are included in a control environment. Systems that are loaded through data collection but are not in any control analytics are not returned by this resource object.

Sorting By Size

You can sort any collection using the hidden `size` element.

The sort logic is calculated based on the following priority order:

- `memory`
- `total_physical_cpus`
- `cores_per_cpu`

For example, if two hosts have the same total memory, the one with the higher count of total physical CPUs is considered the larger host.

Systems with any "_Unknown_" values are sorted at the end of the collection independent of the specified sort order.

Resource

```
/existing-systems
```

Supported Operations

Table: Existing Systems Supported Operations

Operation	HTTP Method	Input	Output	Description
Get Collection	GET /existing-systems	None	Array of Existing Systems: Resource Elements on page 262	Filter-Metadata and Sort By are not supported. Example: Getting a Collection of Hosts with Name "*274*" on page 265, Example: Getting a Collection of Existing Systems with Platform VMware on page 265
Get Individual	GET /existing-systems/<*id*>	None	Existing Systems: Resource Elements on page 262	Retrieve the elements of the Existing Systems specified by `id`. Example: Getting an Individual Existing Host System on page 266, Example: Getting an Individual Existing Guest System on page 267

Resource Elements

Table: Existing Systems Resource Elements

Element	Type	Mod	Filter	Description
id, name, href	strings		F by `id`, `name`, `name_like`	See ID, Name and Self Reference (id, name, href) on page 29. To filter systems with names that contain a given input string, use `?name_like="<`*`substring`*`>"` in your collection request. This filter is case insensitive. The '%' character can be used to match zero or more characters.
children	string			Children, from the System Summary DSE page. For a host system (i.e. `"type": "HOST"`), this is the number of VMs the host system has (e.g. `"children": "10"`). For a VM (i.e. `"type": "GUEST"`), this does not apply (i.e.

Element	Type	Mod	Filter	Description
				`"children": "N/A"`).
control_ environment	[id, name, platform_ category, href, icon]		F by `control_ environment`, `platform`, `platform_ category`	The associated Control Environment where the system belongs. See Control Environments on page 254. Note that when filtering on Control Environments, you must use the element `control_environment` with a name specified, `platform` with a platform specified, or `platform_category` with a platform category specified.
cores_per_cpu	string			Cores per CPU, from the System Summary DSE page.
cpu_ benchmarks	[name, score_ type, value]			The default CPU benchmark. - `name`—Label with possible values: • `"CINT2000"` • `"CINT2000 Rate"` • `"CINT2006 Rate"` • `"RPE2"` - `score_type`—Score type with possible values: • `"cint2000"` • `"cint2000rate"` • `"cint2006rate"` • `"rpe2"` - `value`
cpu_model	string			CPU Architecture, from the System Summary DSE page.
cpu_speed	string			Normalized CPU Speed (MHz), from the System Summary DSE page.
entity_role_ name	string			Entity Role Name, from the System Summary DSE page.
entity_type	string			Type, from the System Summary DSE page.
hostId	string			Host Id, from the System Details DSE Config page.
I/O_ benchmarks	[name, score_ type, value]			The list of all available I/O benchmarks. - `name`—Label with possible values: • `"Maximum Disk Throughput (bytes/s)"` • `"Maximum Network Throughput (bytes/s)"` - `score_type`—Score type with possible values: • `"disk"` • `"net"`

Element	Type	Mod	Filter	Description
				• `value`—A value of -1 means there is no value specified.
infrastructure_group	{id, name, href}		F by `infrastructure_group`	The associated Infrastructure Group where the system belongs. See Infrastructure Groups on page 292. Note that when filtering on Infrastructure Groups, you must use the element `infrastructure_group` with a name specified.
ip_address	string			Primary IP address, from the System Summary DSE page.
mac_address	string			MAC Address, from the System Details DSE Config page.
manufacturer	string			Manufacturer, from the System Summary DSE page.
memory	string			Normalized Total Memory (MB), from the System Summary DSE page.
os	string			Operating System Name, from the System Summary DSE page.
os_patch_level	string			Operating Patch Level, from the System Summary DSE page.
os_version	string			Operating Version, from the System Summary DSE page.
parent	string			Parent, from the System Summary DSE page. For a host system (i.e. `"type": "HOST"`), this does not apply (i.e. `"parent": "N/A"`). For a VM (i.e. `"type": "GUEST"`), this is the name of the parent host system (e.g. `"parent": "esx-host-221"`).
serial_number	string			Serial number, from the System Details DSE page.
size	hidden element			This element is used to sort a collection by size. See Sorting By Size on page 261 for details.
total_logical_cpus	string			Total Logical CPUs, from the System Summary DSE page.
total_physical_cpus	string			Total Physical CPUs, from the System Summary DSE page.
threads_per_core	string			Threads Per Core, from the System Summary DSE page.
type	string		F	The type of system. • `HOST` • `GUEST`

Examples

Example: Getting a Collection of Hosts with Name "*274*"

The following example shows you how to retrieve a collection of hosts with a name containing "274".

Example: Getting a Collection of Hosts with Name "*274*"

Request:

```
GET /existing-systems/?type=host&name_like=274
```

Response:

```
[
  {
    "id": "0a098816-7120-4c67-a897-c227f8c2d750",
    "name": "esx-host-274",
    "href": /existing-systems/0a098816-7120-4c67-a897-c227f8c2d750,
    "type": "HOST",
    "platform_model": "ProLiant DL585 G6",
    "total_physical_cpus": "2",
    "cores_per_cpu": "6",
    "memory": "98304",
    "infrastructure_group": {
      "id": "94edf69b-08ca-41e6-ba2c-3d6be186ca60",
      "name": "Prod_BIPS-01",
      "href": "/infrastructure-groups/94edf69b-08ca-41e6-ba2c-3d6be186ca60"
    },
    "control_environment": {
      "id": "26092815-9d17-4e6a-abbd-f5b05a853bff",
      "name": "Boston",
      "platform_category": "Internal Virtual",
      "href": "/control-environments/26092815-9d17-4e6a-abbd-f5b05a853bff",
      "icon": "/control-environments/26092815-9d17-4e6a-abbd-f5b05a853bff/icon"
    }
  }
]
```

Example: Getting a Collection of Existing Systems with Platform VMware

This example shows you how to retrieve a collection of systems with platform "VMware" (defined by the control environment).

Example: Getting a Collection of Existing Systems with Platform VMware

Request:

```
GET /existing-systems/?platform=vmware
```

Example: Getting an Individual Existing Host System

This example shows you how to retrieve an individual host system by ID.

Example: Getting an Individual Existing Host System

Request:

```
GET /existing-systems/0a098816-7120-4c67-a897-c227f8c2d750
```

Response:

```
{
  "id": "0a098816-7120-4c67-a897-c227f8c2d750",
  "name": "esx-host-274",
  "href": /existing-systems/0a098816-7120-4c67-a897-c227f8c2d750,
  "type": "HOST",
  "os": "VMware",
  "os_version": "ESX Server 4.0.0"",
  "os_patch_level": "N/A",
  "manufacturer": "HP",
  "platform_model": "ProLiant DL585 G6",
  "serial_number": "N/A",
  "hostId": "N/A",
  "entity_type": "VMWare ESX Host",
  "parent": "N/A",
  "children": "13",
  "entity_role_name": "VMWARE_HOST",
  "cpu_model": "AMD Opteron 8435",
  "total_logical_cpus": "12",
  "total_physical_cpus": "2",
  "cores_per_cpu": "6",
  "threads_per_core": "1",
  "cpu_speed": "2600",
  "memory": "98304",
  "ip_address": "192.163.117.108",
  "mac_address": "BC:20:65:17:39:89",
  "cpu_benchmarks": [
    {
      "name": "CINT2006 Rate",
      "score_type": "cint2006rate",
      "value": 160
    }
  ],
  "I/O_benchmarks": [
    {
      "name": "Maximum Disk Throughput (bytes/s)",
      "score_type": "disk",
      "value": 250000000
    },
    {
      "name": "Maximum Network Throughput (bytes/s)",
      "score_type": "net",
      "value": 150000000
    },
  ],
  "infrastructure_group": {
    "id": "94edf69b-08ca-41e6-ba2c-3d6be186ca60",
    "name": "Prod_BIPS-01",
    "href": "/infrastructure-groups/94edf69b-08ca-41e6-ba2c-3d6be186ca60"
  },
  "control_environment": {
    "id": "26092815-9d17-4e6a-abbd-f5b05a853bff",
    "name": "Boston",
    "platform_category": "Internal Virtual",
    "href": "/control-environments/26092815-9d17-4e6a-abbd-f5b05a853bff",
    "icon": "/control-environments/26092815-9d17-4e6a-abbd-f5b05a853bff/icon"
  }
}
```

Example: Getting an Individual Existing Guest System

This example shows you how to retrieve an individual guest system by name.

Example: Getting an Individual Existing Guest System

Request:

```
GET /existing-systems/?name=win-vm-2319&details=true
```

Response:

```
{
  "id": "002922ea-48dc-4c74-9bd1-a718d6afbe05",
  "name": "win-vm-2319",
  "href": /existing-systems/002922ea-48dc-4c74-9bd1-a718d6afbe05,
  "type": "GUEST",
  "os": "Windows",
  "os_version": "Server 2012",
  "os_patch_level": "N/A",
  "manufacturer": "VMware",
  "platform_model": "N/A",
  "serial_number": "4221413de2b2b78da3c678ad5d1a46c5",
  "hostId": "N/A",
  "entity_type": "VMWare ESX Guest",
  "parent": "esx-host-221",
  "children": "N/A",
  "entity_role_name": "VMWARE_VM",
  "cpu_model": "N/A",
  "total_logical_cpus": "1",
  "total_physical_cpus": "1",
  "cores_per_cpu": "1",
  "threads_per_core": "1",
  "cpu_speed": "2666",
  "memory": "98304",
  "ip_address": "192.163.116.37",
  "mac_address": "BC:20:D4:8A:0C:97",
  "cpu_benchmarks": [
    {
      "name": "CINT2006 Rate",
      "score_type": "cint2006rate",
      "value": 27.1933
    }
  ],
  "I/O_benchmarks": [
    {
      "name": "Maximum Disk Throughput (bytes/s)",
      "score_type": "disk",
      "value": -1
    },
    {
      "name": "Maximum Network Throughput (bytes/s)",
      "score_type": "net",
      "value": -1
    },
  ],
  "infrastructure_group": {
    "id": "8a0c1b4e-85bd-422b-ac83-dfdf360619b4",
    "name": "Production Apps1",
    "href": "/infrastructure-groups/8a0c1b4e-85bd-422b-ac83-dfdf360619b4"
  },
  "control_environment": {
    "id": "0a32351d-7a82-43c6-959a-abbc8700ad15",
    "name": "New York",
    "platform_category": "Internal Virtual",
    "href": "/control-environments/0a32351d-7a82-43c6-959a-abbc8700ad15",
    "icon": "/control-environments/0a32351d-7a82-43c6-959a-abbc8700ad15/icon"
  }
}
```

Helper Utilities

Description

This resource is used to provide utilities to:

- `resolve-cpu-benchmark`—calculate the CPU benchmark, when defining new Inbound Hosts on page 280.

Resource

```
/helper-utilities/resolve-cpu-benchmark
```

Supported Operations

Table: Helper Utilities Supported Operations

Operation	HTTP Method	Input	Output	Description
Resolve Benchmarks	POST /helper-utilities/resolve-cpu-benchmark	Helper Utilities: Resource Elements on page 270	Helper Utilities: Properties Returned for Resolve CPU Benchmark on page 271	Resolve the CPU benchmark for the specified system. The input elements correspond to the Plan and Manage Supply > Add Host Supply dialog box. For details, see *Defining New Hosts* (Help Topic ID 390010). Example: Calculating CPU Benchmark on page 271

Resource Elements

Table: Helper Utilities to Resolve CPU Benchmark Resource Elements

Element	Type	Req	Description
cores_per_cpu	string	R	Cores per CPU on the Plan and Manage Supply > Add Host Supply dialog box. Cores per CPU, from the System Summary DSE page.
cpu_allocation	string		Not displayed on the Plan and Manage Supply > Add Host Supply dialog box. CPU Allocation from the System Summary DSE page. CPU Allocation of the system referenced by the `catalog_spec`. -1 if not applicable.
cpu_model	string	R	CPU Architecture on the Plan and Manage Supply > Add Host Supply dialog box. CPU Architecture, as from the System Summary DSE page.
cpu_speed	string	R	CPU Speed on the Plan and Manage Supply > Add Host Supply dialog box. Normalized CPU Speed (MHz), as from the System Summary DSE page.
manufacturer	string		Manufacturer on the Plan and Manage Supply > Add Host Supply dialog box. Manufacturer, from the System Summary DSE page.
memory	string		Total Memory on the Plan and Manage Supply > Add Host Supply dialog box. Normalized Total Memory (MB), from the System Summary DSE page.
platform_model	string		Platform/Model on the Plan and Manage Supply > Add Host Supply dialog box. Model, from the System Summary DSE page.
threads_per_cpu	string	R	Threads Per Core on the Plan and Manage Supply > Add Host Supply dialog box. Threads Per Core, as from the System Summary DSE page.
total_physical_cpus	string	R	Physical CPUs on the Plan and Manage Supply > Add Host Supply dialog box. Total Physical CPUs, from the System Summary DSE page.

Properties Returned for Resolve CPU Benchmark

Table: Helper Utilities to Resolve CPU Benchmark Properties Returned

Element	Type	Description
cpu_benchmarks	[name, value, score_type]	The resulting CPU benchmark. ▪ `name`—Label with possible values: • `"CINT2000"` • `"CINT2000 Rate"` • `"CINT2006 Rate"` • `"RPE2"` ▪ `score_type`—Score type with possible values: • `"cint2000"` • `"cint2000rate"` • `"cint2006rate"` • `"rpe2"` ▪ `value`

Examples

Example: Calculating CPU Benchmark

The following example shows you how to calculate the CPU benchmark.

Example: Calculating CPU Benchmark

Request:

```
POST /helper-utilities/resolve-cpu-benchmark
{
  "manufacturer" : "HP",
  "platform_model" : "ProLiant DL585 G6",
  "cpu_model": "AMD Opteron 8435",
  "cpu_speed": 2600,
  "total_physical_cpus": 2,
  "cores_per_cpu": 6,
  "threads_per_core": 1
}
```

Response:

```
[
  {
    "name": "CINT2006 Rate",
    "score_type": "cint2006rate",
    "value": 160
  }
]
```

Inbound Datastores

Description

The Inbound Datastore is a representation of a new datastore being planned for one or more infrastructure groups.

Full Control Hosting Venues

This API supports only full control hosting venues (i.e. infrastructure groups), as datastore are not managed in other hosting venue types.

Resource

```
/inbound-datastores
```

Supported Operations

Table: Inbound Datastore Supported Operations

Operation	HTTP Method	Input	Output	Description
Get Collection	GET /inbound-datastores	None	Inbound Datastore collection of [id,	Example: Getting a Collection of Inbound Datastores on page 276

Operation	HTTP Method	Input	Output	Description
			name, href]	
Get Individual	GET /inbound-datastores/<*id*>	None	Inbound Datastores: Resource Elements on page 275	Retrieve the Inbound Datastore elements of the specified id. Example: Getting an Individual Inbound Datastore on page 277
Create Individual	POST /inbound-datastores	Inbound Datastores: Resource Elements on page 275	Inbound Datastores: Resource Elements on page 275	One or more Inbound Datastores can be created. Example: Creating Multiple Inbound Datastores on page 278
Create Multiple	POST /inbound-datastores	Inbound Datastores: Resource Elements on page 275 With "num_copy" : <*number*> specified	Inbound Datastores: Resource Elements on page 275	Similar to the Create Individual operation, but specifying the number of Inbound Datastores to create using "num_copy" : <*number*> with no limit to the number of instances created at a time. The names of the Inbound Datastores are auto-generated by appending a number after name. For example, if name=ds, then the generated names would be ds1, ds2, etc. Example: Creating Multiple Inbound Datastores on page 278
Modify Individual	PUT /inbound-datastores/<*id*>	None	Inbound Datastores: Resource Elements on page 275	An Inbound Datastore that is in PENDING or COMMITTED state can be modified through the PUT command. Only the name can be modified. Example: Modifying an Inbound Datastore Name on page 279
Delete Individual	DELETE /inbound-datastores/<*id*>	None	None	An Inbound Datastore in any state can be deleted.
Delete Multiple	DELETE /inbound-datastores	ids: [<*id*>, <*id*>,…]	None	Similar to deleting a single Inbound Datastore above, however, this command deletes multiple Inbound Datastores in one call. Example: Deleting Multiple Inbound Datastores on page 279

Resource Elements

Table: Inbound Datastore Resource Elements

Element	Type	Create/Mod-(Req)	Sort By	Filter	Description
id, name, href	strings	CM-R for `name`	S by `name`	F by `name`	See Common Elements on page 29. The expected name of the incoming datastore, required for auto-reconciliation. For details, see section *Auto-Reconciliation of Systems* of *Booking Overview* (Help Topic ID 230350).
capacity	string				The size in MB of the datastore.
control_ environment	id, name, href, icon	C		F using `control_ environment_ id`	This is the link to the associated Control Environment where the datastore is to be placed. This environment must consist of full control hosting venues.
creation_time	number		S		The date and time this Inbound Datastore object was created, in UTC.
description	string	C			An arbitrary string that describes the new datastore.
expected_date	number	C	S	F	The expected date this Inbound Datastore is expected to be running, in UTC format. The time portion is ignored and always set to 04:00:00. A default of tomorrow's date is used, meaning the datastore must be provisioned today.
infrastructure_ groups	[id, name, href]	C-R			This is the link to the associated Infrastructure Groups where the datastore is to be placed. These Infrastructure Groups must belong to the same `control_ environment`.
owner	string	C	S		Used to define the owner or Customer Name of this Inbound Datastore. If not set, this field is set to the user who is creating the Inbound Datastore.

Element	Type	Create/Mod-(Req)	Sort By	Filter	Description
owner_email	string	C			The email address of the `owner`.
project	string	C			Used to define the Project. If not set, the Project is defined as `"__Unknown__"`.
status	string		S	F	The status of the Inbound Datastore. ▪ `"PENDING"` ▪ `"COMMITTED"` ▪ `"COMPLETED"` ▪ `"EXPIRED"` ▪ `"CANCELLED"`
tier	string				The name of the storage tier that this datastore should belong. The Default definition is selected, by default. You can define the list of possible tier definitions through the Routing Workbench > Storage Tiers screen, as documented in Managing Storage Tiers (Help Topic ID 172400). You can also redefine the Tier definition for the existing datastores through the Routing Workbench > Storage Settings screen, as documented in Managing Storage Settings (Help Topic ID 172390).

Examples

Example: Getting a Collection of Inbound Datastores

The following example shows you how to obtain the collection of current Inbound Datastores.

Example: Getting a Collection of Inbound Datastores

Request:

```
GET /inbound-datastores
```

Response:

```
[
  {
    "id": "12dd64b1-c7ca-452a-9cc7-4ae08f17a3ff",
    "name": "DS1",
    "href": "/inbound-datastores/12dd64b1-c7ca-452a-9cc7-4ae08f17a3ff"
  },
  {
    "id": "98fba16a-7a6c-47e6-8206-a6ef2cfe7ecb",
    "name": "DS2",
    "href": "/inbound-datastores/98fba16a-7a6c-47e6-8206-a6ef2cfe7ecb"
  },
  // ... *SNIP* of Inbound Datastores...
]
```

Example: Getting an Individual Inbound Datastore

The following example shows you how to get a single Inbound Datastore.

Example: Getting an Individual Inbound Datastore

Request:

```
GET /inbound-datastores/98fba16a-7a6c-47e6-8206-a6ef2cfe7ecb
```

Response:

```
{
  "id": "98fba16a-7a6c-47e6-8206-a6ef2cfe7ecb",
  "name": "DS2",
  "href": "/inbound-datastores/98fba16a-7a6c-47e6-8206-a6ef2cfe7ecb",
  "status": "EXPIRED",
  "owner": "Bill",
  "owner_email": "",
  "project": "UCS hardware refresh",
  "description": "",
  "capacity": "102400",
  "tier": "SilverT1",
  "expected_date": 1466654400000,
  "creation_time": 1466568000000,
  "infrastructure_groups": [
    {
      "id": "a0f98746-7fdc-4d15-bc43-1b4199549e6b",
      "name": "Cluster 1",
      "href": "/infrastructure-groups/a0f98746-7fdc-4d15-bc43-1b4199549e6b"
    }
  ],
  "control_environment": {
    "id": "2401338a-9244-49f2-96ea-5aecd3911e0f",
    "name": "vc60-control",
    "href": "/control-environments/2401338a-9244-49f2-96ea-5aecd3911e0f",
    "icon": "/control-environments/2401338a-9244-49f2-96ea-5aecd3911e0f/icon"
  }
}
```

Example: Creating Multiple Inbound Datastores

The following example shows you how to create 2 Inbound Datastores. The tier is defaulted to the "Default" tier.

Example: Creating Multiple Inbound Datastores

Request:

```
POST /inbound-datastores
{
  "num_copy":2,
  "name": "DS2",
  "owner": "Bill",
  "project": "ABC",
  "capacity": "204800",
  "tier": "SilverT1",
  "expected_date": 1472569698000,
  "infrastructure_groups": [
    {
      "id": "b346a9de-d8da-4ec2-9e50-11889d08e62c"
    },
    {
      "id": "503fa16e-464c-4a69-9e0d-5902a8447fb0"
    }
  ]
}
```

Response:

```
{
  "id": "98fba16a-7a6c-47e6-8206-a6ef2cfe7ecb",
  "name": "DS21",
  "href": "/inbound-datastores/98fba16a-7a6c-47e6-8206-a6ef2cfe7ecb",
  "status": "PENDING",
  "owner": "Bill",
  "project": "ABC",
  "description": "",
  "capacity": "204800",
  "tier": "SilverT1",
  "expected_date": 1466654400000,
  "creation_time": 1466568000000,
  "owner_email": "",
  "control_environment": {
    "id": "c37501bf-e2c8-4b45-8275-c06fbfa0863e",
    "name": "Toronto",
    "href": "/control-environments/c37501bf-e2c8-4b45-8275-c06fbfa0863e",
    "icon": "/control-environments/c37501bf-e2c8-4b45-8275-c06fbfa0863e/icon"
  },
  "infrastructure_groups": [
    {
      "id": "b346a9de-d8da-4ec2-9e50-11889d08e62c",
      "name": "Assessment",
      "href": "/infrastructure-groups/b346a9de-d8da-4ec2-9e50-11889d08e62c"
    },
    {
      "id": "503fa16e-464c-4a69-9e0d-5902a8447fb0",
      "name": "Dev2",
      "href": "/infrastructure-groups/503fa16e-464c-4a69-9e0d-5902a8447fb0"
    }
  ]
},
{
  // ... *SNIP* of DS22...
}
```

Example: Modifying an Inbound Datastore Name

The following example updates an Inbound Datastore name to "DS1".

Example: Modifying an Inbound Datastore Name

Request:

```
PUT /inbound-datastores/d274c5ed-b53e-4020-9e42-bc072c6d4816
{
  "name": "ds-1"
}
```

Example: Deleting Multiple Inbound Datastores

The following example deletes three Inbound Datastores, as specified, in one call.

Example: Deleting Multiple Inbound Datastores

Request:

```
DELETE /inbound-datastores
{
  "ids": [
    "575300d8-3a5d-47ea-acfe-4d38c0af0d0e",
    "7589018a-2ed3-4d9d-a047-509439434ed0",
    "05e26393-fc6b-4160-86f4-c73c80b6389d"
  ]
}
```

Inbound Hosts

Description

The Inbound Host is a representation of a new host being planned for an infrastructure group.

Full Control Hosting Venues

This API supports only full control hosting venues (i.e. infrastructure groups), as hosts are not managed in other hosting venues.

Available Capacity

The available capacity is immediately increased when an Inbound Host is created, for all timeframes after the planned start date. An environment refresh is not required. This increased capacity allows for new planned workloads coming in after the inbound host is scheduled for provisioning.

Similarly, the available capacity is immediately decreased if the Inbound Host is deleted. An environment refresh is not required.

Resource

```
/inbound-hosts
```

Supported Operations

Table: Inbound Host Supported Operations

Operation	HTTP Method	Input	Output	Description
Get Collection	GET /inbound-hosts	None	Inbound Host collection of [id, name, href]	Example: Getting a Collection of Inbound Hosts on page 286
Get Individual	GET /inbound-hosts/<*id*>	None	Inbound Hosts: Resource Elements on page 282	Retrieve the Inbound Host elements of the specified `id`. Example: Getting an Individual Inbound Host on page 287
Create Individual	POST /inbound-hosts	Inbound Hosts: Resource Elements on page 282	Inbound Hosts: Resource Elements on page 282	One or more Inbound Hosts can be created. Example: Creating Multiple Inbound Hosts on page 288
Create Multiple	POST /inbound-hosts	Inbound Hosts: Resource Elements on page 282> With `"num_copy" :` < *number*> specified	Inbound Hosts: Resource Elements on page 282	Similar to the Create Individual operation, but specifying the number of Inbound Hosts to create using `"num_copy" : <number>` with no limit to the number of instances created at a time. The names of the Inbound Hosts are auto-generated by appending a number after `name`. For example, if `name=Host`, then the generated names would be Host1, Host2, etc. Example: Creating Multiple Inbound Hosts on page 288
Modify Individual	PUT /inbound-hosts/<*id*>	None	Inbound Hosts: Resource Elements on page 282	An Inbound Host that is in `PENDING` or `COMMITTED` state can be modified through the `PUT` command. Only the `name` can be modified. Example: Modifying an Inbound Host Name on page 290
Delete Individual	DELETE /inbound-hosts/<*id*>	None	None	An Inbound Host in any state can be deleted.
Delete Multiple	DELETE /inbound-hosts	ids: [<*id*>, <*id*>,…]	None	Similar to deleting a single Inbound Host above, however, this command deletes multiple Inbound Hosts in one call. Example: Deleting Multiple Inbound Hosts on

Operation	HTTP Method	Input	Output	Description
				page 290

Resource Elements

Table: Inbound Host Resource Elements

Element	Type	Create/Mod-(Req)	Sort By	Filter	Description
id, name, href	strings	CM-R for `name`	S by `name`	F by `name`	See ID, Name and Self Reference (id, name, href) on page 29. The expected system name of the incoming host, required for auto-reconciliation. For details, see section *Auto-Reconciliation of Systems* of *Booking Overview* (Help Topic ID 230350).
attributes	[id, name, value]	C		F only as that defined in cfg\bookings\bookings-config.xml	On a create request, defines any other attributes of the incoming system. If the attribute name or value is incorrectly specified, an error is returned and the Inbound Host object is not created. For a create, only the name-value pairs are required. The `id` is not required. For a single-valued attribute, if the name-value pair is defined more than once, then only the first occurrence is used and the second one is ignored. For a multi-valued attribute, the name-value pair can be specified multiple times as required for the same named attribute. Duplicates when specifying the same value more than once for the same named attribute are ignored. On a `GET` request, only those attributes that have values are returned.

Element	Type	Create/Mod-(Req)	Sort By	Filter	Description

					The name corresponds to the actual field label. For example:
					`{`
					`  "id": "attr_SecurityZone",`
					`  "name": "Security Zone",`
					`  "value": "Level 1"`
					`}`

					To filter on an attribute, use `attribute.`*id*. For example, to filter all Inbound Hosts that belong to Level 1 Security Zone: `/inbound-hosts/?attribute.attr_SecurityZone=Level 1`
control_environment	id, name, href, icon			F using `control_environment_id`	This is the link to the associated Control Environment where the host is to be placed. This environment must consist of full control hosting venues.
cores_per_cpu	string	C-R			Cores per CPU on the Plan and Manage Supply > Add Host Supply dialog box. Cores per CPU, from the System Summary DSE page.
cpu_allocation	string	C			Not displayed on the Plan and Manage Supply > Add Host Supply dialog box. CPU Allocation of the system.
cpu_benchmarks	[name, score_type, value]	C			The CPU benchmark. If not specified, the benchmark is automatically calculated. To recalculate manually, see Helper Utilities on page 269. `name`—Label with possible values: • `"CINT2000"` • `"CINT2000 Rate"` • `"CINT2006 Rate"` • `"RPE2"` `score_type`—Score type with possible values: • `"cint2000"`

Element	Type	Create/Mod-(Req)	Sort By	Filter	Description
					• "cint2000rate" • "cint2006rate" • "rpe2" value
cpu_model	string	C-R			CPU Architecture on the Plan and Manage Supply > Add Host Supply dialog box. CPU Architecture, as from the System Summary DSE page.
cpu_speed	string	C-R			CPU Speed on the Plan and Manage Supply > Add Host Supply dialog box. Normalized CPU Speed (MHz), as from the System Summary DSE page.
creation_time	number		S	F	The date and time this Inbound Host object was created, in UTC.
description	string	C			An arbitrary string that describes the new host.
expected_date	number	C	S	F	The expected date this Inbound Host is expected to be running and available for placing VMs, in UTC format. The time portion is ignored and always set to 04:00:00. A default of tomorrow's date is used, meaning the host must be provisioned today.
infrastructure_group	id, name, href	C-R		F using `infrastructure_group_id` or `infrastructure_group`	This is the link to the associated Infrastructure Group where the host is to be placed. Note that when filtering on Infrastructure Groups, you must use the element `infrastructure_group_id` with a UUID specified or element `infrastructure_group` with a name specified.
I/O_ benchmarks	[name, score_type, value]	C			The list of all available I/O benchmarks. `name`—Label with possible values:

Element	Type	Create/Mod-(Req)	Sort By	Filter	Description
					• `"Maximum Disk Throughput (bytes/s)"` • `"Maximum Network Throughput (bytes/s)"` ▪ `score_type`—Score type with possible values: • `"disk"` • `"net"` ▪ `value`—A value of -1 means there is no value specified.
manufacturer	string	C-R			Manufacturer on the Plan and Manage Supply > Add Host Supply dialog box. Manufacturer, from the System Summary DSE page.
memory	number	C-R	S		Normalized Total Memory (MB), as from the System Summary DSE page.
owner	string	C	S		Used to define the owner or Customer Name of this Inbound Host. If not set, this field is set to the user who is creating the Inbound Host.
owner_email	string	C			The email address of the `owner`.
platform_model	string	C-R			Platform/Model on the Plan and Manage Supply > Add Host Supply dialog box. Model, from the System Summary DSE page.
project	string	C			Used to define the Project. If not set, the Project is defined as `"__Unknown__"`.
status	string		S	F	The status of the Inbound Host. See State Diagrams for Supply on page 1 for state details. ▪ `"PENDING"` ▪ `"COMMITTED"` ▪ `"COMPLETED"` ▪ `"EXPIRED"` ▪ `"CANCELLED"`
total_logical_	string				Total Logical CPUs, as from the

Element	Type	Create/Mod-(Req)	Sort By	Filter	Description
cpus					System Summary DSE page. This is defined by the system catalog (if specified) or is calculated from `total_physical_cpus` x `cores_per_cpu` x `threads_per_core`.
total_physical_cpus	string	C-R			Physical CPUs on the Plan and Manage Supply > Add Host Supply dialog box. Total Physical CPUs, from the System Summary DSE page.
threads_per_core	string	C-R			Threads Per Core on the Plan and Manage Supply > Add Host Supply dialog box. Threads Per Core, as from the System Summary DSE page.

Examples

Example: Getting a Collection of Inbound Hosts

The following example shows you how to obtain the collection of current Inbound Hosts.

Example: Getting a Collection of Inbound Hosts

Request:

```
GET /inbound-hosts
```

Response:

```
[
  {
    "id": "d274c5ed-b53e-4020-9e42-bc072c6d4816",
    "name": "ucs-host2",
    "href": "/inbound-hosts/d274c5ed-b53e-4020-9e42-bc072c6d4816"
  },
  {
    "id": "ab143af5-f7c1-4b6d-865d-739b71d20a07",
    "name": "ucs-host4",
    "href": "/inbound-hosts/ab143af5-f7c1-4b6d-865d-739b71d20a07"
  },
  // ... *SNIP* of Inbound Hosts...
]
```

Example: Getting an Individual Inbound Host

The following example shows you how to get a single Inbound Host.

Example: Getting an Individual Inbound Host

Request:

```
GET /inbound-hosts/d274c5ed-b53e-4020-9e42-bc072c6d4816
```

Response:

```
{
  "id": "d274c5ed-b53e-4020-9e42-bc072c6d4816",
  "name": "ucs-host2",
  "href": "/inbound-hosts/d274c5ed-b53e-4020-9e42-bc072c6d4816",
  "manufacturer": "Cisco",
  "platform_model": "Cisco UCS B230 M2",
  "cpu_model": "Intel Xeon E7-2870",
  "total_logical_cpus": "40",
  "total_physical_cpus": "2",
  "cores_per_cpu": "10",
  "threads_per_core": "2",
  "cpu_allocation": "-1.00",
  "cpu_speed": "2400",
  "memory": "256000",
  "status": "COMMITTED",
  "owner": "Bill",
  "project": "UCS hardware refresh",
  "description": "",
  "attributes": [
    {
      "id": "attr_SecurityZone",
      "name": "Security Zone",
      "value": "Level 1"
    }
  ],
  "expected_date": 1436068800000,
  "creation_time": 1432815005590,
  "owner_email": "",
  "cpu_benchmarks": [
    {
      "name": "CINT2006 Rate",
      "score_type": "cint2006rate",
      "value": 549
    }
  ],
  "I/O_benchmarks": [
    {
      "name": "Maximum Network Throughput (bytes/s)",
      "score_type": "net",
      "value": -1
    },
    {
      "name": "Maximum Disk Throughput (bytes/s)",
      "score_type": "disk",
        "value": -1
    }
  ],
  "infrastructure_group": {
    "id": "2baa1ac4-4ffe-4e79-a73d-d038145bca8d",
    "name": "Prod2_BDVC-01",
    "href": "/infrastructure-groups/2baa1ac4-4ffe-4e79-a73d-d038145bca8d"
  },
  "control_environment": {
    "id": "26092815-9d17-4e6a-abbd-f5b05a853bff",
    "name": "Boston",
    "href": "/control-environments/26092815-9d17-4e6a-abbd-f5b05a853bff",
    "icon": "/control-environments/26092815-9d17-4e6a-abbd-f5b05a853bff/icon"
  }
}
```

Example: Creating Multiple Inbound Hosts

The following example shows you how to create 2 Inbound Hosts.

Example: Creating Multiple Inbound Hosts

Request:

```
POST /inbound-hosts
{
  "num_copy":2,
  "name": "ucs-host2",
  "manufacturer": "HP",
  "platform_model": "ProLiant DL380 G7",
  "cpu_model": "Intel Xeon X5690",
  "total_physical_cpus": "2",
  "cores_per_cpu": "6",
  "threads_per_core": "2",
  "cpu_allocation": "12.00",
  "cpu_speed": "3466",
  "memory": "131072",
  "owner": "Melissa",
  "owner_email": "melissa@Densify.com",
  "project": "UCS hardware refresh",
  "attributes": [
    {
      "name": "Security Zone",
      "value": "Level 1"
    }
  ],
  "expected_date": 1472659206000,
  "cpu_benchmarks": [
    {
      "name": "CINT2006 Rate",
      "score_type": "cint2006rate",
      "value": 404.47
    }
  ],
  "I/O_benchmarks": [
    {
      "name": "Maximum Network Throughput (bytes/s)",
      "score_type": "net",
      "value": 1342177280
    },
    {
      "name": "Maximum Disk Throughput (bytes/s)",
      "score_type": "disk",
      "value": 1342177280
    }
  ],
  "infrastructure_group": {
    "id": "2baa1ac4-4ffe-4e79-a73d-d038145bca8d"
  }
}
```

Response:

```
[
  {
    "id": "0d69655a-d10b-4d76-88e2-5bc3916c8e07",
    "name": "ucs-host21",
    "href": "/inbound-hosts/0d69655a-d10b-4d76-88e2-5bc3916c8e07",
    "manufacturer": "HP",
    "platform_model": "ProLiant DL380 G7",
    "cpu_model": "Intel Xeon X5690",
    "total_logical_cpus": 24,
    "total_physical_cpus": 2,
    "cores_per_cpu": 6,
    "threads_per_core": 2,
    "cpu_allocation": "12.0",
    "cpu_speed": 3466,
    "memory": 131072,
    "status": "PENDING",
    "owner": "Melissa",
    "project": "UCS hardware refresh",
    "description": "",
    "expected_date": 1472659206000,
    "creation_time": 1470844834427,
    "owner_email": "melissa@Densify.com",
    "attributes": [
      {
        "id": "attr_SecurityZone",
        "name": "Security Zone",
        "value": "Level 1"
      }
    ],
    "cpu_benchmarks": [
      {
        "name": "CINT2006 Rate",
        "score_type": "cint2006rate",
        "value": 404.47
      }
    ],
    "I/O_benchmarks": [
      {
        "name": "Maximum Disk Throughput (bytes/s)",
        "score_type": "disk",
        "value": 1342177280
      },
      {
        "name": "Maximum Network Throughput (bytes/s)",
        "score_type": "net",
        "value": 1342177280
      }
    ],
    "infrastructure_group": {
      "id": "2baa1ac4-4ffe-4e79-a73d-d038145bca8d",
      "name": "Prod2_BDVC-01",
      "href": "/infrastructure-groups/2baa1ac4-4ffe-4e79-a73d-d038145bca8d"
    },
    "control_environment": {
      "id": "26092815-9d17-4e6a-abbd-f5b05a853bff",
      "name": "Boston",
      "href": "/control-environments/26092815-9d17-4e6a-abbd-f5b05a853bff",
      "icon": "/control-environments/26092815-9d17-4e6a-abbd-
f5b05a853bff/icon"
    }
  },
  {
    // ... *SNIP* of Inbound Host "ucs-host22 ...
  }
]
```

Example: Modifying an Inbound Host Name

The following example updates an Inbound Host name from "ucs-host21" to "ucs-host1".

Example: Modifying an Inbound Host Name

Request:

```
PUT /inbound-hosts/0d69655a-d10b-4d76-88e2-5bc3916c8e07
{
  "name": "ucs-host1"
}
```

Example: Deleting Multiple Inbound Hosts

The following example deletes three Inbound Hosts, as specified, in one call.

Example: Deleting Multiple Inbound Hosts

Request:

```
DELETE /inbound-hosts
{
  "ids": [
    "575300d8-3a5d-47ea-acfe-4d38c0af0d0e",
    "7589018a-2ed3-4d9d-a047-509439434ed0",
    "05e26393-fc6b-4160-86f4-c73c80b6389d"
  ]
}
```

Infrastructure Groups

Description

The Infrastructure Group object represents a full control hosting venue (i.e. infrastructure group), a non-control hosting venue or a guest-level hosting venue. The term Infrastructure Group refers to any of the three, for the purpose of this guide.

This API returns Infrastructure Group details so that Workloads can be routed appropriately. The API provides stats and amenities at each defined timeframe of the timeline. The API also provides the results of the Fit-for-Purpose checks for a specific hosting venue and workload placement.

Hosting Venues

This API supports full control hosting venues (i.e. infrastructure groups), non-control hosting venues and guest-level hosting venues.

With respect to this API, the difference between the hosting venues is as follows:

- Options for Returned Details—for the amenities query, the empty arrays `"host_summary": []` and `"sensor_capacities": []` are returned as there are no hosts or sensors in non-control and guest-level hosting venues; all options are available for all three hosting venues
- Supported Operations—all operations are supported for all three hosting venues except:
  - for non-control and guest-level hosting venues, you can only open/close but not auto-close them using `PUT /infrastructure-groups/<id>`
- Resource Elements—only elements that are applicable to the hosting venue is returned in the response. Specifically:

- for non-control and guest-level hosting venues, element value `allow_new_workloads: "DO_NOT_ALLOW_IF_RISK"`, stats elements (specifically `cei`, `total_guests`, `total_hosts`, `required_hosts`, `outbound_guest_bookings`, `inbound_host_bookings`, `outbound_host_bookings`) are not returned
- for non-control hosting venues, element value `last_refreshed_time` is not returned

Options for Returned Details

You can extend the amenities query:

```
GET /infrastructure-groups/<id>/amenities/
```

with any or all of the following to return only the information required:

- `host_summary=false`—does not return the `host_summary` array
- `sensor_capacities=false`—does not return the `sensor_capacities` array
- `capability_groups=false`—does not return the `capability_groups` array

By default, these options are all true.

Resource

```
/infrastructure-groups
/infrastructure-groups/amenities
```

Supported Operations

Table: Infrastructure Group Supported Operations

Operation	HTTP Method	Input	Output	Description
Get Collection	GET /infrastructure-groups	None	Infrastructure Group collection of [id, name, platform, platform_category, href, hardware_name, hardware_icon, control_type, color_code_selected,	Default Sort By is defined as: `?sort_by=name.` Filter-Metadata is supported. Example: Getting a Collection of Infrastructure Groups on page 300

Operation	HTTP Method	Input	Output	Description
			color_code_unselected]	
Get Individual	GET /infrastructure-groups/<*id*>	None	Infrastructure Groups: Resource Elements on page 294	Retrieve the Infrastructure Group elements of the specified `id`. Example: Getting an Individual Infrastructure Group on page 301
Get Individual Amenities	GET /infrastructure-groups/<*id*>/amenities	None	Amenities of the Infrastructure Group: Resource Elements on page 299	Retrieve the amenities of the Infrastructure Group specified by `id`, for each defined timeframe of the timeline. Example: Getting the Amenities of an Individual Infrastructure Group on page 306, Example: Getting the Amenities of an Individual Infrastructure Group for Today on page 310
Modify Individual	PUT /infrastructure-groups/<*id*>	Infrastructure Groups: Form Definition on page 298	Infrastructure Groups: Resource Elements on page 294	An Infrastructure Group can be modified to open, close or auto-close it, in terms of allowing new workloads. Example: Modifying an Infrastructure Group to Auto-Close Based on Performance Risk on page 312, Example: Manually Closing an Infrastructure Group on page 312
Post Individual Fit-for-Purpose Checks	POST /infrastructure-groups/<*id*>/ffp-check	Fit-for-Purpose Checks of the Infrastructure Group: Form Definition on page 300	categories, as defined in Resource Elements on page 366	Retrieve the results of the Fit-for-Purpose checks for the Infrastructure Group specified by `id` and the workload placement specified by Fit-for-Purpose Checks of the Infrastructure Group: Form Definition on page 300. Example: Getting an Infrastructure Group's Fit-for-Purpose Checks on page 303

Resource Elements

Infrastructure Groups: Resource Elements

Table: Infrastructure Group Resource Elements

Element	Type	Sort By	Filter	Description
id, name,	strings	S by	F by `name`	See ID, Name and Self Reference (id, name, href) on

Element	Type	Sort By	Filter	Description
href		name	Note: filter-metadata returns infrastructure_groups with the metadata values of name	page 29.
allow_new_workloads	string			Indicates whether this Infrastructure Group is accepting new workloads or not. This corresponds to the field **Allow New VM Bookings** in Managing Hosting Venues. • "ALLOW" • "DO_NOT_ALLOW"—with this option, the availability_status is set to "BLOCKED_BY_USER". When setting this option, you can also set "availability_reason" (e.g. to "New VM Bookings have been Blocked by Operations") • "DO_NOT_ALLOW_IF_RISK"—if there is risk, the availability_status is set to "BLOCKED_BY_USER" and the "availability_reason" is automatically set to the detected performance risk (e.g. "100.0% of Hosts in the Infrastructure Group are Under-Provisioned, exceeds policy limit of 50%"); if no risk has been identified, then the availability_status is set to "AVAILABLE"
amenities	href			The href to the Amenities for the Infrastructure Group. The Amenities object is used to show the capabilities of the Infrastructure Group, for each defined timeframe of the timeline (referenced by the short_name of the Timeline Tags on page 526). This includes the supported hardware, licensed software, compute and storage information. See Amenities of the Infrastructure Group: Resource Elements on page 299 below for details.
availability_reason	string			The reason why the availability_status has the value it has. Can be user-defined or Densify-defined: • "Reserved" • "Infrastructure Group has Critical Notifications" • "Infrastructure unavailable, insufficient data for capacity calculation" • "Infrastructure unavailable, analysis is out of date"

Element	Type	Sort By	Filter	Description
				• `"New VM Bookings have been Blocked by Operations"` • `"<%> Hosts/VMs in the Infrastructure Group are Under-Provisioned, exceeds policy limit <%>"` • `"<#> Hosts/VMs in the Infrastructure Group are Under-Provisioned, exceeds policy limit <#>"`
availability_status	string	S	F Note: filter-metadata returns `status_values` with all possible states	Whether or not the Infrastructure Group is available or not for routing. • "`UNAVAILABLE`"−when Control Environment has never been refreshed, the `last_refreshed_time` is older than `rest.api.WithinHoursFromLastRefreshOfCluster`, Infrastructure Group has never been refreshed, Infrastructure Group is disqualified due to analysis errors or missing benchmarks, planned date is past the lastest timeframe • "`BLOCKED_BY_USER`"−when the Infrastructure Group is not accepting quests (i.e. Allow New VM Bookings is set to "No", either manually or automatically blocked from performance risks) • "`AVAILABLE`"−in all other cases. "`AVAILABLE`" is returned even when the Control Environment or Infrastructure Group is refreshing as long as the refresh has been done at least once (which provides the data for the availability)
color_code_selected	string			The color displayed in the Hosting Venues pane of the Route and Reserve Demand page, when this hosting venue tile is selected.
color_code_unselected	string			The color displayed in the Hosting Venues pane of the Route and Reserve Demand page, when this hosting venue tile is not selected.
control_type	string		F	The type of hosting venue: • "`FULL`"—full control hosting venue • "`GUEST_LEVEL`"—guest-level hosting venue • "`NONE`"—non-control hosting venue
control_environment	id, name, platform, platform_category, href, icon		F by `control_environment_id`	The Control Environment where this Infrastructure Group belongs.
creation_time	number	S		The UTC date and time the Infrastructure Group was created.

Element	Type	Sort By	Filter	Description
hardware_icon	string			The URL where the icon representing the hardware is found, e.g. "http://thedocs1:8086/CIRBA/images/serverModelIcon/Dell_small.png".
hardware_name	string			The name of the hardware in the Infrastructure Group, e.g. "PowerEdge".

last_refreshed_time	number	S
		sort by `last_refresh_time`
			F
			Note: `filter-metadata` returns `last_refresh_time` with only UTC values, and filter by `last_refresh_time` with UTC values
				The UTC date and time the Infrastructure Group itself was last refreshed.
				If the Infrastructure Group has never been refreshed, the following is returned:
				`"last_refreshed_time": 0`
				If all Infrastructure Groups in the filter-metadata query have never been refreshed, then the filter-metadata returns:
				`"last_refresh_time": {`
				`  "min": 0,`
				`  "max": 0`
				`}`
				Filtering by range with a value of 0 is not supported. e.g. the following does not return all Infrastructure Groups that have never been refreshed:
				`?last_refresh_time_from=0&`
				`   last_refresh_time_to=0`

platform	string		F Note: `filter-metadata` returns `platform_type` with the metadata values of `platform`	The system type being analyzed. For examples: - "IBM" - "HYPERV" - "VMWARE" - "XENSERVER" - "X86"
platform_category	string		F	The type of platform. Infrastructure Groups have the same `platform_category` as the Control Environment where they belong. Possible categories are: - "`Internal Virtual`"—for control environments, e.g. for a VMware environment whether or not fully controlled through analytics - "`Internal Physical`"—for physical environments, e.g. a Bare Metal - "`External Cloud`"—for external cloud environments, e.g. an AWS environment - "`External Physical`"—for external physical environments, e.g. SoftLayer Bare Metal

Element	Type	Sort By	Filter	Description
stats	Complex, as specified in the Description			The overall statistics of the Infrastructure Group at each timeframe. For each timeframe that has data (tagged using `short_name` of the Timeline Tags on page 526): • cei • timeline_name—the `long_name` of the Timeline Tags on page 526 • date—same as `date` of the Timeline Tags on page 526 • total_guests • total_hosts • required_hosts • inbound_guest_bookings • outbound_guest_bookings • inbound_host_bookings • outbound_host_bookings • amenities If the Infrastructure Group has never been refreshed or is currently being refreshed (even for a specific timeframe), the following is returned: `"stats": {}` ***Note****: If the Infrastructure Group has been disqualified, then the CEI is -1 and all the above totals are 0.*

Form Definition

You can use an Infrastructure Group form if you want to specify how the Infrastructure Group can be modified to open, close or auto-close it, in terms of allowing new workloads.

Table: Infrastructure Group Form Definition

Element	Type	Req	Description
allow_new_workloads	string	✓	To specify whether or not to allow new workloads, as documented above for allow_new_workloads. Options are: • "`ALLOW`" • "`DO_NOT_ALLOW`" • "`DO_NOT_ALLOW_IF_RISK`"
availability_reason	string		Optionally, you can define a reason why new workloads are not allowed, as documented above for availability_reason. Setting this element is valid only when `allow_new_workloads` is set to "`DO_NOT_ALLOW`".

Amenities of the Infrastructure Group: Resource Elements

For each timeframe, the following information on amenities is returned.

Table: Amenities of the Infrastructure Group Resource Elements

Element	Type	Description
href	string	Link to this amenities for this timeframe. e.g. for "timeline": "180", "href": "/infrastructure-groups/542cd4bc-1e4a-494c-930c-e00679e9e11a/amenities/180".
capability_groups	Complex, as specified in the Description	An array of capabilities, NOT returned if option `capability_groups=false` is used in the query. An array entry defines one capability grouping, as follows: - name—name of the capability group (e.g. `"Resource"`, `"Security"`, `"Business"`, `"Technical"`) - capabilities—array of capabilities within the group, where each entry is of the form `name-type-values` of types string, string, array, respectively • name—name of the capability sub-group (e.g. `"Operating Systems"`) • type—value type (e.g. `"Pick-list type A"`, `"Parameter-driven"`) • values—array of supported capabilities in the subgroup, as `key-value` pairs, where `key` is the priority order in integer form starting from 0 when type is `"Pick-list type A"` or `key` is the name of the parameter when type is `"Parameter-driven"`
feature_description	string	Optional description for the timeframe amenities.
host_summary	Complex, as specified in the Description	An array of the different hosts in the Infrastructure Group, NOT returned if option `host_summary=false` is used in the query. An array entry defines each model, as follows: - count—number of host systems in the Infrastructure Group with the same `model_name` - memory—amount of total memory of the host systems in MB - model_name—manufacturer and model type of the host systems - cpu_cores—number of total CPU cores of the host systems - os_names—array of operating system names that are installed on the host systems
sensor_capacities	Complex, as specified in the Description	An array of metrics for each sensor type, NOT returned if option `sensor_capacities=false` is used in the query. An array entry defines each sensor type as follows:

Element	Type	Description
		- metrics—an array of metrics in the form of `name-value` pairs, where each metric is the sum from all the sensors with the same `sensor_type` - sensor_type—type of sensor (i.e. "Datastores", "Physical Storage", "Resource Pools") - total_sensors—number of sensors with the same `sensor_type`
timeline	string	Name of the timeframe of the timeline. This is the same as the `short_name` of the Timeline Tags on page 526.

Fit-for-Purpose Checks of the Infrastructure Group: Form Definition

In order to retrieve the results of the Fit-for-Purpose checks, you need to specify the workloads for placement along with the expected date.

Table: Fit-for-Purpose Checks of the Infrastructure Group Form Definition

Element	Type	Req	Description
workloads	[id]	✓	An array of Workload ids to be placed.
expected_date	number	✓	The date these workloads are expected to arrive in UTC. The time portion is ignored. The date must be today's date or some date in the future.

Examples

Example: Getting a Collection of Infrastructure Groups

The following example shows you how to obtain the collection of current hosting venues.

Example: Getting a Collection of Infrastructure Groups

Request:

```
GET /infrastructure-groups
```

Response:

```
[
  {
    "id": "104716a8-63c7-4fe9-a82d-48d5ec1fbf33",
    "name": "Eng-UAT",
    "platform": "VMWARE",
    "platform_category": "Internal Virtual",
    "href": "/infrastructure-groups/104716a8-63c7-4fe9-a82d-48d5ec1fbf33"
    "hardware_name": "PowerEdge",
    "hardware_icon": http://thedocs1:8086/CIRBA/images/serverModelIcon/Dell 
small.png,
    "control_type": "FULL"
    "color_code_selected": "#AACBEC"
    "color_code_unselected": "#DCE5F4"
  },
  {
    "id": "11673bee-6e02-436c-afc5-64692a517e08",
    "name": "Bos-Eng&Dev",
    "platform": "IBM",
    "platform_category": "Internal Virtual",
    "href": "/infrastructure-groups/11673bee-6e02-436c-afc5-64692a517e08"
    "hardware_name": "Power",
    "hardware_icon": http://thedocs1:8086/CIRBA/images/serverModelIcon/ibm 
small.png,
    "control_type": "FULL"
    "color_code_selected": "#AACBEC"
    "color_code_unselected": "#DCE5F4"
  },
  // ... *SNIP* of Infrastructure Groups...
]
```

Example: Getting an Individual Infrastructure Group

The following example shows you how to get a single hosting venue.

Example: Getting an Individual Infrastructure Group

Request:

```
GET /infrastructure-groups/104716a8-63c7-4fe9-a82d-48d5ec1fbf33
```

Response:

```
{
  "id": "104716a8-63c7-4fe9-a82d-48d5ec1fbf33",
  "name": "Eng-UAT",
  "platform": "VMWARE",
  "platform_category": "Internal Virtual",
  "stats": {
    "Today": {
      "cei": 0.83,
      "timeline_name": "Today",
      "date": 1365480000000,
      "total_guests": 72,
      "total_hosts": 6,
      "required_hosts": 5,
      "inbound_guest_bookings": 0,
      "outbound_guest_bookings": 0,
      "inbound_host_bookings": 0,
      "outbound_host_bookings": 0
      "amenities": "/infrastructure-groups/104716a8-63c7-4fe9-a82d-
48d5ec1fbf33/amenities/Today"
    },
    "90": {
      "cei": 0.83,
      "timeline_name": "90 Days",
      "date": 1373256000000,
      "total_guests": 72,
      "total_hosts": 6,
      "required_hosts": 5,
      "inbound_guest_bookings": 0,
      "outbound_guest_bookings": 0,
      "inbound_host_bookings": 0,
      "outbound_host_bookings": 0
      "amenities": "/infrastructure-groups/104716a8-63c7-4fe9-a82d-
48d5ec1fbf33/amenities/90"
    },
    "60": {
      "cei": 0.83,
      "timeline_name": "60 Days",
      "date": 1370664000000,
      "total_guests": 72,
      "total_hosts": 6,
      "required_hosts": 5,
      "inbound_guest_bookings": 0,
      "outbound_guest_bookings": 0,
      "inbound_host_bookings": 0,
      "outbound_host_bookings": 0
      "amenities": "/infrastructure-groups/104716a8-63c7-4fe9-a82d-
48d5ec1fbf33/amenities/60"
    },
    "30": {
      "cei": 0.83,
      "timeline_name": "30 Days",
      "date": 1368072000000,
      "total_guests": 72,
      "total_hosts": 6,
      "required_hosts": 5,
      "inbound_guest_bookings": 0,
      "outbound_guest_bookings": 0,
      "inbound_host_bookings": 0,
      "outbound_host_bookings": 0
      "amenities": "/infrastructure-groups/104716a8-63c7-4fe9-a82d-
48d5ec1fbf33/amenities/30"
    },
    "180": {
      "cei": 0.83,
      "timeline_name": "180 Days",
      "date": 1366084800000,
      "total_guests": 73,
      "total_hosts": 6,
      "required_hosts": 5,
      "inbound_guest_bookings": 0,
      "outbound_guest_bookings": 0,
      "inbound_host_bookings": 0,
      "outbound_host_bookings": 0
      "amenities": "/infrastructure-groups/104716a8-63c7-4fe9-a82d-
48d5ec1fbf33/amenities/180"
    }
  },
  "href": "/infrastructure-groups/104716a8-63c7-4fe9-a82d-48d5ec1fbf33",
  "hardware_name": "PowerEdge",
  "hardware_icon": "http://thedocs1:8086/CIRBA/images/serverModelIcon/Dell_
small.png",
  "availability_status": "BLOCKED_BY_USER",
  "availability_reason": "New VM Bookings have been Blocked by Operations",
  "allow_new_workloads": "DO_NOT_ALLOW",
  "amenities": "/infrastructure-groups/104716a8-63c7-4fe9-a82d-
48d5ec1fbf33/amenities",
  "control_type": "FULL",
  "color_code_selected": "#AACBEC",
  "color_code_unselected": "#DCE5F4",
  "control_environment": {
    "id": "e1d6867b-abca-4587-a9cf-1f9df2667eb5",
    "name": "Houston",
    "platform": "VMWARE",
    "platform_category": "Internal Virtual",
    "control_type": "FULL",
    "href": "/control-environments/e1d6867b-abca-4587-a9cf-1f9df2667eb5",
    "icon": "/control-environments/e1d6867b-abca-4587-a9cf-1f9df2667eb5/icon"
  },
  "creation_time": 1365908733947,
  "last_refreshed_time": 1371232026497
}
```

Example: Getting an Infrastructure Group's Fit-for-Purpose Checks

The following example shows you how to get the Fit-for-Purpose results for a specific hosting venue and workload placement, for the expected date as specified.

Example: Getting an Infrastructure Group's Fit-for-Purpose Checks

Request:

```
POST /infrastructure-groups/05c8eabb-77ad-49dc-b22c-93d7a0e6da0e/ffp-check
{
  "workloads": [
    {"id": "5bda671d-832a-4ccc-8f28-43c4ad106189"}
  ],
  "expected_date": 9908939200000
}
```

Response:

```
{
  "categories": [
    {
      "name": "Resource",
      "test": [
        {
          "name": "Operating Systems",
          "status_reasons": [],
          "status": "PASS"
        },
        {
          "name": "Guest Sizes",
          "status_reasons": [],
          "status": "PASS"
        }
      ],
      "status": "PASS"
    },
    {
      "name": "Security",
      "test": [
        {
          "name": "Data Jurisdiction",
          "status_reasons": [],
          "status": "PASS"
        },
        {
          "name": "Compliance",
          "status_reasons": [],
          "status": "PASS"
        },
        {
          "name": "Encryption and Key Management",
          "status_reasons": [],
          "status": "PASS"
        },
        {
          "name": "Identity and Access Management",
          "status_reasons": [],
          "status": "PASS"
        },
        {
          "name": "Security Zones",
          "status_reasons": [
            {
              "reason": "Required Security Zone not supported",
              "workload_id": "5bda671d-832a-4ccc-8f28-43c4ad106189",
              "workload_name": "sas-mid33"
            }
          ],
          "status": "FAIL"
        },
        {
          "name": "Network Isolation",
          "status_reasons": [],
          "status": "PASS"
        },
        {
          "name": "Intel TXT Support",
          "status_reasons": [],
          "status": "PASS"
        }
      ],
      "status": "FAIL"
    },
    {
      "name": "Business",
      "test": [
        {
          "name": "Location",
          "status_reasons": [],
          "status": "PASS"
        },
        {
          "name": "Departments",
          "status_reasons": [
            {
              "reason": "Department not allowed",
              "workload_id": "5bda671d-832a-4ccc-8f28-43c4ad106189",
              "workload_name": "sas-mid33"
            }
          ],
          "status": "FAIL"
        },
        {
          "name": "Service Level",
          "status_reasons": [],
          "status": "PASS"
        }
      ],
      "status": "FAIL"
    },
    {
      "name": "Technical",
      "test": [
        {
          "name": "Hosting Platform",
          "status_reasons": [],
          "status": "PASS"
        },
        {
          "name": "Hypervisor",
          "status_reasons": [],
          "status": "PASS"
        },
        {
          "name": "Maximum Guest I/O",
          "status_reasons": [],
          "status": "PASS"
        },
        {
          "name": "Operational Environments",
          "status_reasons": [],
          "status": "PASS"
        },
        {
          "name": "DPDK",
          "status_reasons": [],
          "status": "PASS"
        },
        {
          "name": "Software Licenses",
          "status_reasons": [],
          "status": "PASS"
        },
        {
          "name": "Crypto H/W Acceleration",
          "status_reasons": [],
          "status": "PASS"
        },
        {
          "name": "License Groups",
          "status_reasons": [
            {
              "reason": "License Group not allowed",
              "workload_id": "5bda671d-832a-4ccc-8f28-43c4ad106189",
              "workload_name": "sas-mid33"
            }
          ],
          "status": "FAIL"
        },
        {
          "name": "Transport Zone",
          "status_reasons": [],
          "status": "PASS"
        },
      ],
      "status": "FAIL"
    }
  ],
  "status": "FAIL"
}
```

Example: Getting the Amenities of an Individual Infrastructure Group

The following example shows you how to get the amenities for a single hosting venue.

Example: Getting the Amenities of an Individual Infrastructure Group

Request:

```
GET /infrastructure-groups/4f22bdc6-7b77-49ab-bf34-6f4f75625da3/amenities
```

Response:

```
[
  {
    "timeline": "Today",
    "href": "/infrastructure-groups/4f22bdc6-7b77-49ab-bf34-
6f4f75625da3/amenities/Today",
    "feature_description": "",
    "host_summary": [
      {
        "count": 6,
        "memory": 589824,
        "model_name": "Dell PowerEdge M610",
        "cpu_cores": 64,
        "os_names": [
          "VMware ESX Server 4.1.0",
          "VMware ESX Server 4.0.0"
        ]
      }
    ],
    "sensor_capacities": [
      {
        "metrics": [
          {
            "name": "Total Provisioned Space (MB)",
            "value": 7973470
          },
          {
            "name": "Total Used Space (MB)",
            "value": 4412514
          },
          {
            "name": "Total Required Space (MB)",
            "value": 5022252
          },
          {
            "name": "Capacity (MB)",
            "value": 7570376
          }
        ],
        "sensor_type": "Datastores",
        "total_sensors": 5
      },
    // ... *SNIP* of capabilities for Physical Storage and Resource Pools //
    ],
    "capability_groups": [
      {
        "name": "Resource",
        "capabilities": [
          {
            "name": "Operating Systems",
            "type": "Pick-list type A",
            "values": [
              {
                "key": "0",
                "value": "Linux"
              },
              {
                "key": "1",
                "value": "Windows"
              }
            ]
          },
          {
            "name": "Maximum Guest Size",
            "type": "Parameter-driven",
            "values": [
              {
                "key": "Memory size (MB)",
                "value": "24567"
              },
              {
                "key": "Number of vCPUs",
                "value": "4"
              }
            ]
          },
          {
            "name": "Datastores Tiers",
            "type": "Parameter-driven",
            "values": [
              {
                "key": "0",
                "value": "Default"
              },
              {
                "key": "1",
                "value": "Gold"
              },
              {
                "key": "2",
                "value": "Silver"
              }
            ]
          }
        ]
      },
      {
        "name": "Security",
        "capabilities": [
          {
            "name": "Data Jurisdiction",
            "type": "Pick-list type A",
            "values": []
          },
          {
            "name": "Compliance",
            "type": "Pick-list type A",
            "values": []
          {,
          {
            "name": "Encryption and Key Management",
            "type": "Pick-list type A",
            "values": []
          },
          {
            "name": "Identity and Access Management",
            "type": "Pick-list type A",
            "values": []
          },
          {
            "name": "Security Zones",
            "type": "Pick-list type A",
            "values": []
          },
          {
            "name": "Network Isolation",
            "type": "Pick-list type A",
            "values": []
          },
          {
            "name": "Intel TXT Support",
            "type": "Pick-list type A",
            "values": []
          }
        ]
      },
      {
        "name": "Business",
        "capabilities": [
          {
            "name": "Location",
            "type": "Pick-list type A",
            "values": []
          },
          {
            "name": "Departments",
            "type": "Pick-list type A",
            "values": []
          },
          {
            "name": "Service Levels",
            "type": "Pick-list type A",
            "values": []
          }
        ]
      },
      {
        "name": "Technical",
        "capabilities": [
          {
            "name": "Hosting Platform: VMware",
            "type": "Parameter-driven",
            "values": []
          },
          {
            "name": "Hypervisor: VMware",
            "type": "Parameter-driven",
            "values": []
          },
          {
            "name": "Maximum Guest I/O",
            "type": "Parameter-driven",
            "values": [
              {
                "key": "Context Switches (CtSw/s)",
                "value": "80000"
              },
              {
                "key": "Disk I/O (Bytes/s)",
                "value": "60000000"
              },
              {
                "key": "Disk I/O (Operations/s)",
                "value": "10000"
              },
              {
                "key": "Network I/O (Bytes/s)",
                "value": "60000000"
              },
              {
                "key": "Network I/O (Packets/s)",
                "value": "50000"
              }
            ]
          },
          {
            "name": "Operational Environments",
            "type": "Pick-list type A",
            "values": [
              {
                "key": "0",
                "value": "ProdApp"
              },
              {
                "key": "1",
                "value": "ProdGen"
              },
              {
                "key": "2",
                "value": "ProdStd"
              },
            ]
          },
          {
            "name": "DPDK",
            "type": "Pick-list type A",
            "values": []
          },
          {
            "name": "Software Licenses",
            "type": "Pick-list type A",
            "values": []
          },
          {
            "name": "Crypto H/W Acceleration",
            "type": "Pick-list type A",
            "values": []
          },
          {
            "name": "License Groups",
            "type": "Pick-list type A",
            "values": []
          },
          {
            "name": "Transport Zone",
            "type": "Pick-list type A",
            "values": []
          },
        ]
      }
    ]
  },
  // ... *SNIP* of other timelines ...
]
```

Example: Getting the Amenities of an Individual Infrastructure Group for Today

The following example shows you how to get amenities for a single hosting venue for today's date. The following query does not return the `capability_groups` array:

Example: Getting the Amenities of an Individual hosting venue for Today

Request:

```
GET /infrastructure-groups/4f22bdc6-7b77-49ab-bf34-6f4f75625da3/amenities/Today?capability_groups=false
```

Response:

```
{
  "timeline": "Today",
  "href": "/infrastructure-groups/4f22bdc6-7b77-49ab-bf34-
6f4f75625da3/amenities/Today",
  "feature_description": "",
  // ... *SNIP* of other elements as example above for "Today" ...
}
```

Example: Getting the hosting venue Filter Metadata for an Environment

This example retrieves the filter metadata for the hosting venues that belong to a specific environment.

Example: Getting the Infrastructure Group Filter Metadata for an Environment

Request:

```
GET /infrastructure-group-filter-metadata/?control_environment_id=e6cf1672-
77ff-4e7d-9dda-5387b0bc95cc
```

Response:

```
{
  "control_environments": [
    {
      "id": "e6cf1672-77ff-4e7d-9dda-5387b0bc95cc",
      "name": "Boston",
      "platform": "VMWARE",
      "platform_category": "Internal Virtual",
      "control_type": "FULL",
      "href": "/control-environments/e6cf1672-77ff-4e7d-9dda-5387b0bc95cc"
      "icon": "/control-environments/e6cf1672-77ff-4e7d-9dda-5387b0bc95cc/icon"
    }
  ],
  "status_values":
    "AVAILABLE",
    "UNAVAILABLE",
    "BLOCKED_BY_USER"
  ],
  "infrastructure_groups": [
    "Bos-GenApps",
    "Bos-ProdApps1",
    "Bos-ProdApps2",
    "Bos-Eng&Dev"
  ],
  "platform_category": [
    "Internal Virtual",
    "External Cloud",
    "Internal Physical",
  ],
  "last_refresh_time": {
    "min": 1380557115513,
    "max": 1380557115513
  },
  "platform_type": [
    "VMWARE"
  ],
  "control_types": [
    "FULL"
  ]
}
```

Example: Getting the Infrastructure Group Filter Metadata for IBM

Request:

```
GET /infrastructure-group-filter-metadata/?platform=IBM
```

Example: Getting the Infrastructure Group Filter Metadata for Specific Groups

Request:

```
GET /infrastructure-group-filter-metadata/?infrastructure_
group=Engineering1,Engineering2
```

Example: Modifying an Infrastructure Group to Auto-Close Based on Performance Risk

Request:

```
PUT /infrastructure-groups/4f22bdc6-7b77-49ab-bf34-6f4f75625da3
{
  "allow_new_workloads": "DO_NOT_ALLOW_IF_RISK"
}
```

Example: Manually Closing an Infrastructure Group

Request:

```
PUT /infrastructure-groups/4f22bdc6-7b77-49ab-bf34-6f4f75625da3
{
  "allow_new_workloads": "DO_NOT_ALLOW",
  "availability_reason": "New VM workloads are blocked"
}
```

Monitored Hosts

Description

This resource is used to return the health status of the monitored hosts. The returned specifics match that provided by the DCE System Health Status dashboard. See section *System Health Status* of *DCE System Health Tab* (Help Topic ID 171190).

You can also use this resource to override the health status of a monitored host. For example, to override a host to healthy (or unhealthy) for a specified time, health check monitoring for that host stops until the specified date/time has past. When you override the health status, the `details` is set to "Status Override Executed". After the specified date/time, regular monitoring rules for the host resumes.

Resource

```
/monitored-hosts
```

Supported Operations

Table: Monitored Hosts Supported Operations

Operation	HTTP Method	Input	Output	Description
Get Collection	GET /monitored-	None	Monitored Host Collection of [id,	Filter-Metadata and Sort By are not supported. Example: Getting a Collection of Healthy

Operation	HTTP Method	Input	Output	Description
	hosts		name, href]	Monitored Hosts from Cluster 1 on page 315
Get Individual	GET /monitored-hosts/<*id*>	None	Monitored Hosts: Resource Elements on page 314	Retrieve the elements of the Monitored Host specified by `id`. Example: Getting Health Status of an Individual Monitored Host on page 316
Modify Individual	PUT /monitored-hosts/<*id*>	status and/or status_ timestamp	Monitored Hosts: Resource Elements on page 314	Override the host health status of the Monitored Host specified by `id`. Either both or one of the following can be specified: - `status`—to override the status to "HEALTHY" or "UNHEALTHY" (case insensitive). If any other value, an error is returned. - `status_timestamp`—to override the timestamp. If not specified, the current date/time is used (i.e. monitoring is resumed) and the status is assumed to be valid for right now. If a future date/time is specified, then this is the date/time until which the health status is considered valid (i.e. no monitoring performed for this host until this date/time is past). If a past date/time is specified, monitoring is resumed. If the host belongs to more than one environment, the status and timestamp are updated across all groups. Example: Modifying the Health Status of a Monitored Host on page 316

Resource Elements

Table: Monitored Hosts Resource Elements

Element	Type	Mod	Filter	Description
id, name, href	strings		F by `name`	See ID, Name and Self Reference (id, name, href) on page 29.
health_ status	string	M using `status` or `status_ timestamp`	F using `health_ status` to query on `status` or `infrastructure_ group_id`	The health of the monitored host. - `status`—health status of the Monitored Host, either "`HEALTHY`" or "`UNHEALTHY`" - `details`—details of the health status. When you override the health status, the `details` is set to "Status Override Executed". When the

Element	Type	Mod	Filter	Description
				status is "HEALTHY", the details is set to null. - status_timestamp—date/time of the health check in UTC. If the timestamp is in the past, then it is the date/time of the last health check update. If the timestamp is in the future, then this is the date/time until which the health status is considered valid (i.e. no checks required until this date/time has past). - infrastructure_group—id, name, href link to the containing Infrastructure Group where the Monitored Host is located

Examples

Example: Getting a Collection of Healthy Monitored Hosts from Cluster 1

The following example shows you how to retrieve a collection of monitored hosts that reside in Cluster 1 that are also healthy.

Example: Getting a Collection of Healthy Monitored Hosts from Cluster 1

Request:

```
GET /monitored-hosts/?health_status=HEALTHY&infrastructure_group_id=1e212147-
6669-4c60-be6e-616e7ba022f2
```

Response:

```
[
  {
    "id": "f9a20e24-ee01-4ead-b16f-c4071269a2c3",
    "name": "host1",
    "href": /monitored-hosts/f9a20e24-ee01-4ead-b16f-c4071269a2c3,
  },
  {
    "id": "999ecf21-79cb-4cdb-a7bc-be8787c06d68",
    "name": "host2",
    "href": "/monitored-hosts/999ecf21-79cb-4cdb-a7bc-be8787c06d68"
  },
  // ... *SNIP* ...
]
```

Example: Getting Health Status of an Individual Monitored Host

The following example shows you how to get the health status details of an individual monitored host. The response shows that the host is included in Cluster 1 and is unhealthy.

Example: Getting Health Status of an Individual Monitored Host

Request:

```
GET /monitored-hosts/23191b13-a2ee-4106-894f-5d0470c473b9
```

Response:

```
{
  "id": "23191b13-a2ee-4106-894f-5d0470c473b9",
  "name": "host1",
  "href": "/monitored-hosts/23191b13-a2ee-4106-894f-5d0470c473b9"
  "health_status": [
    {
      "status": "UNHEALTHY",
      "details": "Total CPUs Overcommit:525.0:FAIL",
      "status_timestamp": 1445616961000,
      "infrastructure_group": {
        "id": "1e212147-6669-4c60-be6e-616e7ba022f2",
        "name": "Cluster 1",
        "href": "/infrastructure-groups/1e212147-6669-4c60-be6e-616e7ba022f2"
      }
    }
  ]
}
```

Example: Modifying the Health Status of a Monitored Host

The following example shows you how to modify the health status details of an individual monitored host. In this example, both the `status` and `status_timestamp` are updated, as specified.

Example: Modifying the Health Status of a Monitored Host

Request:

```
PUT /monitored-hosts/f9a20e24-ee01-4ead-b16f-c4071269a2c3
{
  "status": "UNHEALTHY",
  "status_timestamp": 1445620201000
}
```

Response:

```
{
  "id": "f9a20e24-ee01-4ead-b16f-c4071269a2c3",
  "name": "host1",
  "href": "/monitored-hosts/f9a20e24-ee01-4ead-b16f-c4071269a2c3"
  "health_status": [
    {
      "status": "UNHEALTHY",
      "details": "Status Override Executed",
      "status_timestamp": 1445620201000,
      "infrastructure_group": {
        "id": "1e212147-6669-4c60-be6e-616e7ba022f2",
        "name": "Cluster 1",
        "href": "/infrastructure-groups/1e212147-6669-4c60-be6e-616e7ba022f2"
      }
    }
  ]
}
```

Outbound Hosts

Description

The Outbound Host is a representation of an existing host being planned for removal.

Full Control Hosting Venues

This API supports only full control hosting venues (i.e. infrastructure groups), as hosts are not managed in other hosting venues.

Available Capacity

The available capacity is immediately decreased when an Outbound Host is created, for all timeframes after the planned removal date. Similarly, the available capacity is immediately restored if the Outbound Host is deleted. An environment refresh is not required.

Resource

```
/outbound-hosts
```

Supported Operations

Table: Outbound Host Supported Operations

Operation	HTTP Method	Input	Output	Description
Get Collection	GET /outbound-hosts	None	Outbound Host collection of [id, name, href]	Example: Getting a Collection of Outbound Hosts on page 323
Get Individual	GET /outbound-hosts/<*id*>	None	Outbound Hosts: Resource Elements on page 319	Retrieve the Outbound Host elements of the specified id. Example: Getting an Individual Outbound Host on page 323
Create Individual	POST /outbound-hosts	Outbound Hosts: Resource Elements on page 319	Outbound Hosts: Resource Elements on page 319	An Outbound Host can be created. Example: Creating An Outbound Host on page 325
Delete Individual	DELETE /outbound-hosts/<*id*>	None	None	An Outbound Host in any state can be deleted.
Delete Multiple	DELETE /outbound-hosts	ids: [<*id*>, <*id*>,…]	None	Similar to deleting a single Outbound Host above, however, this command deletes multiple Outbound Hosts in one call. Example: Deleting Multiple Outbound Hosts on page 326

Resource Elements

The resource elements for the Outbound Host are similar to that for the Inbound Host. The only difference is the `existing_system` element, as described below.

Table: Outbound Host Resource Elements

Element	Type	Create/Mod-(Req)	Sort By	Filter	Description
id, name, href	strings		S by `name`	F by `name`	See ID, Name and Self Reference (id, name, href) on page 29. The system name of the outgoing host is required for auto-reconciliation.

Element	Type	Create/Mod-(Req)	Sort By	Filter	Description
					For details, see section *Auto-Reconciliation of Systems* of *Booking Overview* (Help Topic ID 230350).

attributes	[id, name, value]			F only as that defined in cfg\bookings\bookings-config.xml	On a `GET` request, only those attributes that have values are returned.
					The name corresponds to the actual field label. For example:
					`{`
					`  "id": "attr_SecurityZone",`
					`  "name": "Security Zone",`
					`  "value": "Level 1"`
					`}`

					To filter on an attribute, use `attribute.`*id*. For example, to filter all Outbound Hosts that belong to Level 1 Security Zone: `/outbound-hosts/?attribute.attr_SecurityZone=Level 1`
control_environment	id, name, href, icon			F using `control_environment_id`	This is the link to the associated Control Environment where the host is being removed. This environment must consist of full control hosting venues.
cores_per_cpu	string				Cores per CPU, from the System Summary DSE page.
cpu_allocation	string				CPU Allocation of the system. "-1.00" if not applicable.
cpu_benchmarks	[name, score_type, value]				The CPU benchmark. ▯ `name`—Label with possible values: • `"CINT2000"` • `"CINT2000 Rate"` • `"CINT2006 Rate"` • `"RPE2"` ▯ `score_type`—Score type with possible values: • `"cint2000"` • `"cint2000rate"` • `"cint2006rate"` • `"rpe2"`

Element	Type	Create/Mod-(Req)	Sort By	Filter	Description
					▪ `value`
cpu_model	string				CPU Architecture, as from the System Summary DSE page.
cpu_speed	string				Normalized CPU Speed (MHz), as from the System Summary DSE page.
creation_time	number		S	F	The date and time this Outbound Host object was created, in UTC.
description	string	C			An arbitrary string that describes the reason to remove the host.
existing_system	id, name, href, type	existing_system_id is C-R			This is the link to the associated Existing System that is being removed from the environment.
expected_date	number	C	S	F	The expected date this Outbound Host is expected to be removed and no longer available for placing VMs, in UTC format. The time portion is ignored and always set to 12:00:0 AM in your time zone. A default of tomorrow's date is used, meaning the host must be vacated today.
infrastructure_group	id, name, href	C		F using `infrastructure_group`	This is the link to the associated Infrastructure Group where the host currently belongs. If the host belongs to multiple Infrastructure Groups and this element is not specified on a create, then a random Infrastructure Group is chosen. Note that when filtering on Infrastructure Groups, you must use the element `infrastructure_group` with a name specified.
I/O_benchmarks	[name, score_type, value]				The list of all available I/O benchmarks. ▪ `name`—Label with possible values: • "`Maximum Disk`

Element	Type	Create/Mod-(Req)	Sort By	Filter	Description
					`Throughput (bytes/s)"` • `"Maximum Network Throughput (bytes/s)"` `score_type`—Score type with possible values: • `"disk"` • `"net"` `value`—A value of -1 means there is no value specified.
manufacturer	string				Manufacturer, from the System Summary DSE page.
memory	number				Normalized Total Memory (MB), as from the System Summary DSE page.
owner	string	C			Used to define the owner or Customer Name of this Outbound Host. If not set, this field is set to `"__Unknown__"`.
owner_email	string	C			The email address of the `owner`.
platform_model	string				Model, from the System Summary DSE page.
project	string	C			Used to define the Project. If not set, the Project is defined as `"__Unknown__"`.

status	string		S	F	The status of the Outbound Host. See State Diagrams for Supply on page 1 for state details.
					`"PENDING"`
					`"COMMITTED"`
					`"COMPLETED"`
					`"EXPIRED"`
					`"CANCELLED"`

total_logical_cpus	string				Total Logical CPUs, as from the System Summary DSE page.
total_physical_cpus	string				Total Physical CPUs, from the System Summary DSE page.
threads_per_core	string				Threads Per Core, as from the System Summary DSE page.

Examples

Example: Getting a Collection of Outbound Hosts

The following example shows you how to obtain the collection of current Outbound Hosts.

Example: Getting a Collection of Outbound Hosts

Request:

```
GET /outbound-hosts
```

Response:

```
[
  {
    "id": "9410ea4b-b8c6-402b-a8de-b3e7b16a28ce",
    "name": "esx-host-29",
    "href": "/outbound-hosts/9410ea4b-b8c6-402b-a8de-b3e7b16a28ce"
  },
  {
    "id": "a95537e1-5805-4185-b0f5-d95fa818763a",
    "name": "esx-host-268",
    "href": "/outbound-hosts/a95537e1-5805-4185-b0f5-d95fa818763a"
  },
  // ... *SNIP* of Outbound Hosts...
]
```

Example: Getting an Individual Outbound Host

The following example shows you how to get a single Outbound Host.

Example: Getting an Individual Outbound Host

Request:

```
GET /outbound-hosts/9410ea4b-b8c6-402b-a8de-b3e7b16a28ce
```

Response:

```
{
  "id": "9410ea4b-b8c6-402b-a8de-b3e7b16a28ce",
  "name": "esx-host-29",
  "href": "/outbound-hosts/9410ea4b-b8c6-402b-a8de-b3e7b16a28ce",
  "manufacturer": "Dell",
  "platform_model": "PowerEdge M610",
  "cpu_model": "Intel Xeon X5650",
  "total_logical_cpus": "40",
  "total_physical_cpus": "2",
  "cores_per_cpu": "6",
  "threads_per_core": "2",
  "cpu_allocation": "-1.00",
  "cpu_speed": "2666",
  "memory": "98304",
  "status": "PENDING",
  "owner": "Bill",
  "project": "__Unknown__",
  "description": "",
  "attributes": [
    {
      "id": "attr_1",
      "name": "Location",
      "value": "New York"
    },
    {
      "id": "attr_2",
      "name": "Department",
      "value": "Business Services"
    },
    {
      "id": "attr_ApplicationTier",
      "name": "Application Tier",
      "value": "eBill Payment"
    },
    {
      "id": "attr_DiskIO(Bytes)",
      "name": "Disk IO (Bytes)",
      "value": "5923500.0"
    },
    {
      "id": "attr_DiskIO(Ops)",
      "name": "Disk IO (Ops)",
      "value": "0.0"
    },
    {
      "id": "attr_NetworkIO(Bytes)",
      "name": "Network IO (Bytes)",
      "value": "936000.0"
    },
    {
      "id": "attr_NetworkIO(Packets)",
      "name": "Network IO (Packets)",
      "value": "0.0"
    },
    {
      "id": "attr_OperationalEnvironment",
      "name": "Operational Environment",
      "value": "ProdApp"
    },
    {
      "id": "attr_SecurityZone",
      "name": "Security Zone",
      "value": "Level 2"
    }
  ],
  "expected_date": 1472616000000,
  "creation_time": 1472071332297,
  "owner_email": "",
  "existing_system": {
      "id": "45539e03-3ee5-4bad-b57f-ed53bc48243d",
      "name": "esxcrb11.int.densify.com",
      "href": "/existing-systems/45539e03-3ee5-4bad-b57f-ed53bc48243d",
      "type": "HOST"
  },
  "cpu_benchmarks": [
    {
      "name": "CINT2006 Rate",
      "score_type": "cint2006rate",
      "value": 326.32
    }
  ],
  "I/O_benchmarks": [
    {
      "name": "Maximum Network Throughput (bytes/s)",
      "score_type": "net",
      "value": 150000000
    },
    {
      "name": "Maximum Disk Throughput (bytes/s)",
      "score_type": "disk",
        "value": 250000000
    }
  ],
  "infrastructure_group": {
    "id": "8a0c1b4e-85bd-422b-ac83-dfdf360619b4",
    "name": "Production Apps1",
    "href": "/infrastructure-groups/8a0c1b4e-85bd-422b-ac83-dfdf360619b4"
  },
  "control_environment": {
    "id": "0a32351d-7a82-43c6-959a-abbc8700ad15"",
    "name": "New York",
    "href": "/control-environments/0a32351d-7a82-43c6-959a-abbc8700ad15"",
    "icon": "/control-environments/0a32351d-7a82-43c6-959a-abbc8700ad15"/icon"
  }
}
```

Example: Creating An Outbound Host

The following example shows you how to create an Outbound Host.

Example: Creating An Outbound Host

Request:

```
POST /outbound-hosts
{
  "existing_system_id": "907bd7a4-79f2-4452-9098-2187eca91084",
  "expected_date": 1416490819000",
  "project": "ABC",
  "description":"test",
  "owner": "Mark",
  "owner_email":"myang@densify.com",
  "infrastructure_group":
  {
    "id": "1f69aa53-9be5-48a6-8cbd-02de9c75be15"
  }
}
```

Example: Deleting Multiple Outbound Hosts

The following example deletes three Outbound Hosts, as specified, in one call. When a delete is made, the status of each Outbound Host is set to "CANCELLED". The Outbound Host objects are deleted at the next environment refresh.

Example: Deleting Multiple Outbound Hosts

Request:

```
DELETE /outbound-hosts
{
  "ids": [
    "575300d8-3a5d-47ea-acfe-4d38c0af0d0e",
    "7589018a-2ed3-4d9d-a047-509439434ed0",
    "05e26393-fc6b-4160-86f4-c73c80b6389d"
  ]
}
```

Ping API

Description

The `/ping` resource is used to check the status of the Densify API subsystem. No credentials are required for this resource to validate the API subsystem. A ping API call performs an end-to-end test from the client to the API subsystem, ensuring that the Densify database is available and is able to respond to requests. Network elements, such as load balancers, can use this resource to verify that the web server receiving the request can be kept as an active participant in the resource pool.

Resource

```
/CIRBA/api/ping
```

Supported Operations

Table: Ping API Supported Operations

Operation	HTTP Method	Input	Output	Description
Ping the Densify API subsystem	`GET /CIRBA/api/ping`	Query String Parameter: ▪ timeout *(optional)*	▪ message ▪ status	Used to return the status of a connectivity test to the Densify API subsystem with no authorization. See Response Elements for the returned output.

Operation	HTTP Method	Input	Output	Description
				Example: Successful Ping API

Parameters

Query String Parameters

Table: Ping API Query String Parameters

Parameter Name	Type	Description
timeout *(optional)*	integer	The maximum time, in seconds, for the Densify API Server to respond before returning a timeout error. If the timeout parameter is not specified, then the default timeout is set to 30 seconds. The valid range of timeout is between 15 and 180 seconds. Specifying any value outside of the valid range will produce a malformed request error. Example: Ping API Timeout

Response

Response Elements

The following table list elements and possible status codes returned from the `/ping` request.

Element	Type	Description
message	string	Detailed message of the status response.
status	errorCode	HTTP Response Code Possible response code values of the `/ping` request include: - 200—Successful response: API subsystem is responding to requests; - 400—The provided timeout value was invalid (outside the range of 15 - 180 seconds). - 500—There is an issue connecting to the Densify API subsystem. Contact Densify support Support@Densify.com for further details.

***Note**: Other response error messages could indicate issues with the Densify web server or other connectivity issues to the web server.*

Examples

Example: Successful Ping API

The following example shows you a ping request to the Densify API subsystem with a timeout threshold of 100 seconds.

Example: Successful Ping API

Request:

```
GET /CIRBA/api/ping?timeout=100
```

Response:

```
{
  "message" : "ok",
  "status" : 200
}
```

Example: Ping API Timeout

The following example shows you a ping request with an invalid timeout value.

Example: Ping API Timeout

Request:

```
GET /CIRBA/api/ping?timeout=10
```

Response:

```
{
  "message" : "Timeout value valid range: 15-180",
  "status" : 400
}
```

Receive Metrics Jobs

Description

The Receive Metrics Jobs resource is part of a series of data management tools using the Data Ingestion API framework for users on SaaS deployments to transfer utilization data into Densify.

The Data Ingestion API framework allows you to customize and extend data transfer capabilities (such as transferring metrics, configuration, attributes, or commands) for Densify SaaS deployments in a secure job execution framework. The framework encapsulates a custom data management tool into a Job entity in Densify, where you can create, list, schedule, execute, and delete. When you schedule the Densify job for execution, the custom data management tool is invoked. The framework also allows you to upload and download job artifacts, such as data or log files.

By default, Densify is installed with the Metrics data management tool which allows SaaS users to invoke a standard Receive Metrics Jobs resource from the Densify REST API. This resource allows you to upload a metrics file into a Densify job and schedule the job for processing, which transfers the metrics data to their respective services for analysis. You can also use the Receive Metrics Jobs resource to download any result files or logs of the job execution.

Resource

```
/receive/metrics/jobs/
```

Supported Operations

Table: Receive Metrics Jobs Supported Operations

Operation	HTTP Method	Request Input Parameters	Response Elements	Description
Create a job	`POST /receive/metrics/jobs`	Request Body Parameters: [name] [parameters]	jobId name job_status	Creates a job in Densify with the support structure for uploading utilization metrics. If the `[name]` parameter is not supplied, then the job's name will be set to the job GUID. See Example: Create a Job.
Get all jobs	`GET /receive/metrics/jobs`	None	Collection of jobs: jobId name	Returns a list of all jobs in Densify. See Example: Get All Jobs.
Get details of an individual job	`GET /receive/metrics/jobs/<jobId>?[lines_to_tail_in_logs=n]`	Path Parameter: jobId Query String Parameter: [lines_to_tail_in_logs]	jobId name parameters audit_info input_files lines_to_tail_in_logs logs job_status	Returns the details of the job specified by job ID provided in the request. See Example: Get Specific Job Details.
Get job input file details	`GET /receive/metrics/jobs/<jobId>/input`	Path Parameter: jobId	jobId name parameters input_files	Returns a list of input files for the job. See Example: Get Job Input File Details.
Get job log file details	`GET /receive/metrics/jobs/<jobId>/logs?[lines_to_tail_in_logs=n]`	Path Parameter: jobId Query String Parameter: [lines_to_tail_in_logs]	jobId name parameters logs	Returns a list of log files for the job. See Example: Get Job Log File Details.
Get job audit details	`GET /receive/metrics/jobs/<jobId>/audit_info`	Path Parameter: jobId	jobId name	Returns audit information for the job

Operation	HTTP Method	Request Input Parameters	Response Elements	Description
			• parameters • audit_info	specified. See Example: Get Job Audit Information.
Upload a file to a job	`POST /receive/metrics/jobs/<jobId>?[execute=true]&[time=HH:MM]`	Path Parameter: • jobId Query String Parameters: • [execute =true] • [time =HH:MM] Request Body Parameter: • file	• jobId • name • job_status	Uploads a file into an existing job. See Example: Upload a File.
Download files in a job	`GET /receive/metrics/jobs/<jobId>/download/files?[file]`	Path Parameter: • jobId Query String Parameters: • [file]	• octet-stream zipped file containing the downloaded files See Download Files on page 338.	This request returns all files designated for download of the specified job. You also have the option to download a specific file, if the file name is known. See Example: Download Files.
Download logs in a job	`GET /receive/metrics/jobs/<jobId>/logs/files?[file]`	Path Parameter: • jobId Query String Parameters: • [file]	• octet-stream zipped file containing log files See Download Files on page 338.	This request returns all log files of the specified job. You also have the option to download a specific log file if the filename is known. See Example: Download Logs.
Delete job input files	`DELETE /receive/metrics/jobs/<jobId>/contents/input`	Path Parameter: • jobId	• message • status	Delete all input files associated with the specified job. See Example: Delete Input Files.
Delete job log files	`DELETE /receive/metrics/jobs/<jobId>/contents/logs`	Path Parameter: • jobId	• message • status	Delete all log files associated with the

Operation	HTTP Method	Request Input Parameters	Response Elements	Description
				specified job. Use this request to clean up the job log files on the Densify server. See Example: Delete Log Files.
Delete a job	`DELETE /receive/metrics/jobs/<jobId>`	Path Parameter: jobId	message status	Deletes all content associated with the job (i.e. input, download, and log files) and removes the job from the scheduled job list. See Example: Delete Job.
Update parameters of an existing job	`PUT /receive/metrics/jobs/<jobId>/parameters`	Path Parameter: jobId Request Body Parameters: [parameters]	jobId name parameters input_files lines_to_tail_in_logs logs job_status	Updates the parameter attributes of an existing job. See Example: Update Job Parameters.

Parameters

The following is a complete list of possible parameters for the `/receive/metrics/jobs` endpoint. Path, query string, and/or request body parameters are required depending on the method requested.

Path Parameters

Table: Receive Metrics Jobs - Path Parameters

Parameter Name	Type	Description
jobId	string	Specify the job GUID to identify the job.

Request Body Parameters

Table: Receive Metrics Jobs - Request Body Parameters

<table>
<tr><th>Parameter Name</th><th>Type</th><th>Description</th></tr>
<tr><td>name

<em>(optional)</em></td><td>string</td><td>The name of the job.

During job creation, if the <code>name</code> parameter is not set, then <code>name</code> will be automatically set to the job globally unique identifier (GUID).

Example of setting the job name:
<pre>   {"name": "my-sample-job"}</pre>
If you do not want to specify any parameters for creating a job, you must still provide an empty JSON body element when using the POST operation:
<pre>   POST /receive/metrics/jobs
   { }</pre></td></tr>
<tr><td>parameters

<em>(optional, depending on the metrics custom endpoint)</em></td><td>Array of name-value pairs</td><td>The job <code>parameters</code> element is an array of <code>"name"</code>, <code>"value"</code> pairs that you can provide in the request body.

The <code>"parameters"</code> element needs to be provided in the following JSON format:
<pre>"parameters":
[
   {"name": <string>,
    "value": <string>},
    ...
 ]</pre>
The parameters required during job creation is dependent on the metrics custom endpoint used. Below is an example of setting optional parameters during job creation:
<pre>{
  "name" : "My new job",
  "parameters":
  [
   {
     "name": "JobPriority",
     "value": "urgent"
   },
   {
     "name": "Licence",
     "value": "true"
   },
   {
     "name": "Area",
     "value": "NewArea"
   }
  ]
}</pre>
In an update parameters request, you need to provide the complete list</td></tr>
</table>

Parameter Name	Type	Description

		of parameters, including the updated ones and the non-updated ones. The new list of parameters used for the update command overwrites the entire old list of parameters.
		Below is an example of the request body for updating the "Area" parameter. Notice that the entire parameters list is provided.
		`[`
		`{`
		`"name": "JobPriority",`
		`"value": "urgent"`
		`},`
		`{`
		`"name": "Licence",`
		`"value": "true"`
		`},`
		`{`
		`"name": "Area",`
		`"value": "west-2b"`
		`}`
		`]`

file	multipart/form-data	To upload a file into an existing job, attach the file into the `file` form-data key in the body of the POST request.

Query String Parameters

Table: Receive Metrics Jobs - Query String Parameters

Parameter Name	Type	Description
lines_to_tail_in_logs *(optional)*	integer	Specify the number of lines from the bottom of the log files to display. For example, to display the last 100 lines of logs, you would specify the following: `lines_to_tail_in_logs=100` The default value of `-1` denotes that the entire log files will be displayed.
execute *(optional)*	string	The job `execute` option specifies whether the job should be executed with the uploaded file. Possible values for the `execute` option: - `true`—The job is executed at the scheduled time provided by the time parameter. If time is not set and you specify `execute=true`, then the job is executed immediately. - `false`—The file is uploaded with no job execution. This is the default behavior if no `execute` option is specified. You can use the `execute=false` option to upload multiple files to the job before executing the bulk job.
time *(optional)*	string	The job `time` option is used in conjunction with the execute option to specify when the job is to be executed next.

Parameter Name	Type	Description
		The `time` value must be in *`HH:mm`* 24-hr format.

file
(optional)
	string	The `file` option allows you to download a specific file.
		You must specify the exact download filename in order for this operation to succeed. For example, to download a file named "output.txt", the following call is made:
		`GET /receive/metrics/jobs/455fa7bb-10fb-41a7-96a9-f4b13bd7a05c/download/files?file=output.txt`
		Here is an example to download a log file named "output.log":
		`GET /receive/metrics/jobs/455fa7bb-10fb-41a7-96a9-f4b13bd7a05c/logs/files?file=output.log`

Response

The following is a complete list of possible response elements that are returned for the `/receive/metrics/jobs` resource. If the response element does not apply to the API request, then the element is not displayed in the results.

Table: Receive Metrics Jobs Response Schema

Element	Type	Description
jobId	string	The globally unique identifier (GUID) assigned to the job.
name	string	The name of the job. During job creation, if the `[name]` parameter is not set, then it will be automatically set to the job globally unique identifier (GUID).
parameters	Array of ▪ `name` ▪ `value`	The job `parameters` element is an array of `"name"`, `"value"` pairs that is dependent on the data management tool used. The Metrics tool is the default data management tool for the Receive Metrics Jobs resource.

audit_info	▪ `audit name`
	▪ `audit date`
	▪ `audit path`
	▪ `target audit`
	▪ `target failed`
	▪ `audit end date`
	▪ `load date`
	▪ `load end date`
	▪ `load status`
		If the job has uploaded metric files which have been audited, the last audit details are displayed.
		The `audit_info` element displays the following information:
		`"audit_info": {`
		`    "audit_name": <string>,`
		`    "audit_date": <string>,`
		`    "audit_path": <string>,`
		`    "target_audit": <int>,`
		`    "targets_failed": <int>,`
		`    "audit_end_date": <string>,`
		`    "load_date": <string>,`
		`    "load_end_date": <string>,`
		`    "load_status": <string>,`
		`}`

Element	Type	Description

input_files	Array of:
	`name`
	`size`
		If the job has uploaded files, details of those files are displayed:
		`"input_files": [`
		`    {`
		`    "name": <string>,`
		`    // name of the metrics file`
		`    "size": <int>,`
		`    // size of the file, in bytes`
		`    },`
		`    ...`
		`    {`
		`    "name": <string>,`
		`    "size": <int>,`
		`    }`
		`]`

lines_to_tail_in_logs	integer	Specifies the number of lines from the bottom of the log file to display. The default value of `-1` denotes that the entire log file will be displayed.
logs	`name` `contents`	Contains the details of each file in the job's log directory. For each log file available, the following information is displayed: `name`—displays the log filename; `contents`—displays the tail end contents of the log file, depending on the lines_to_tail_in_logs parameter.

job_status	`code`
	`message`
	`files`
		An element that provides the status of the executed Metrics job operation, which contains the following items:
		`code`—Displays the status code from the statuscode.txt file in the job status folder. If the statuscode.txt file does not exist, then the last audit load code is displayed:
		• `0`—successful operation;
		• `-1`—an error with loading the job.
		`message`—Displays the contents of the statusmessage.txt file in the job status folder. If the statusmessage.txt file does not exist, then the last audit load status message is displayed (e.g. Loaded, Loaded_Error).
		`files`—Displays the details of each non-status file from the job status folder.
		The `job_status` element is displayed in the following form:
		`{"code": <int>,`
		`"message": <string>,`
		`"files": [`
		`  {`
		`    "name": <string>, // filename`
		`    "size": <int>, /size in bytes`
		`  },`
		`   ...`
		`  ]`
		`}`
		Below is a response example if there are additional non-status files in the job status folder:
		`"job_status": {`

Element	Type	Description

		`"code": 0,`
		`"message": "File Loaded Successfully.",`
		`"files": [`
		`{`
		`"name": "production_status.zip",`
		`"size": 908756`
		`},`
		`{`
		`"name": "test_status.zip",`
		`"size": 453903`
		`}`
		`]`
		`}`
		The default Metrics data management tool produces a statusmessage.txt file, a corresponding statuscode.txt file, and other status files in the job status directory for each job execution. The Data Ingestion API framework provides flexibility for a custom data management tool to produce their own status message, code, and files by generating files with the exact name (i.e. statusmessage.txt, statuscode.txt) in the custom tool's job status directory.
		Note*: File names are case sensitive. Both message and code files must exist in the job status directory for the framework to override the job_status element.*

Download Files Response		

Download Files	octet-stream zipped file containing files	The response for a file or log download request is an octet-stream zipped file containing all the files produced when executing the job. For a single file download, the zipped file will contain only one file. The suggested filename (in the response Content-Disposition) is in the following format:
		- for download logs request:
		`{jobId}_logs.zip`
		- for download file request:
		`{jobId}_download.zip`
		Where `{jobId}` is the job GUID.

Messages and Error Handling		
message	string	A response message for the request from the `job_status` element. The `job_status: message` string can be one of the following: - AccessDeniedException: "Access Denied! Unable to create temporary file or directory"—Check file permissions on file system. - IngestionJobDirectoryDoesNotExistException—Job file structure does not exist in the system. - IngestionJobDirectoryNotReadbleException—Job file structure permission is possibly set to read-only. - IngestionJobInvalidArgumentException—Invalid arguments in the file upload request. - IOException: "There is not enough space on the disk. IO Exception"—Check disk space. - IOException—Any I/O exception that occurred on the system, such as

Element	Type	Description
		when the specified filename cannot be found. - Exception—Any other error that occurred during the processing of the upload request.
status	integer	The status code of the job request. Possible `"status"` values are: - `200`—job action successful (i.e. delete action was successful); - `404`—the job does not exist; - `409`—the job is currently running, request action canceled; - `400`—the job files are not accessible or permissions are invalid; - `500`—the job is corrupt or any other exception.

HTTP Code 400	HTTP error	Request is malformed. For example, when the time format is invalid, the following error message is displayed:
		`{`
		`  "message": "Invalid schedule time format <HH:MM>. Schedule task cannot be updated or created.",`
		`  "status": 400`
		`}`

HTTP Code 403	HTTP error	"Limit number of active jobs has been reached." The number of active jobs on the server exceeds the Densify configuration setting of maximum job limit.
HTTP Code 404	HTTP error	The job does not exist in the system.
HTTP Code 500	HTTP error	- "Working directory does not exist."—The Data Ingestion API root working directory does not exist. - "OS error message."—Failed to create directory structure due to insufficient storage..

Examples

Example: Create a Job

The following example shows you how to create a job:

Example: Create a Job

Request:

```
POST /receive/metrics/jobs/
```

JSON Body:

```
{ "name": "sample-job" }
```

Response:

```
{
  "jobId": "483e2a46-1f40-4e8a-9ee2-532f66dc549b",
  "name": "sample-job",
  "job_status": {
      "code": 0,
      "message": "Job created successfully"
   }
}
```

Example: Get All Jobs

The following example shows you how retrieve all jobs:

Example: Get All Jobs

Request:

```
GET /receive/metrics/jobs
```

Response:

```
[
  {
  "jobId": "e79518b7-fe0f-4bd4-ae5a-2ca51974d77b",
  "name": "sample-job"
  },
  {
  "jobId": "58419075-6009-4c73-8783-54a49e347384",
  "name": "my-job"
  }
]
```

Example: Get Specific Job Details

The following example shows you how retrieve details of a specific job:

Example: Get Job Details

Request:

```
GET /receive/metrics/jobs/58419075-6009-4c73-8783-54a49e347384?lines_to_tail_
in_logs=2
```

Response:

```
{
  "jobId": "58419075-6009-4c73-8783-54a49e347384",
  "name": "CPU utilization - upload job",
  "parameters": [{
        "name": "hint",
        "value": "Sample"
      },
      {
        "name": "license",
        "value": "true"
      }
  ],
  "audit_info": {
      "audit_name": "58419075-6009-4c73-8783-54a49e347384",
      "audit_date": "2018-02-13 09:23:37.000",
      "audit_path": "output\00001518_M0500_58419075-6009-4c73-8783-
54a49e347384",
      "targets_audited": 323,
      "targets_failed": 0,
      "audit_end_date": "2018-02-13 09:45:11.333",
      "load_date": "2018-02-13 09:52:23.930",
      "load_end_date": "2018-02-13 09:52:26.577",
      "load_status": "Loaded"
  },
  "input_files": [{
        "name": "allmetrics_production.csv",
        "size": 203945
      }, {
        "name": "h2metrics_test.csv",
        "size": 19932
      }
  ],
  "lines_to_tail_in_logs": 2,
  "logs": [
      {
        "name": "main.log",
        "contents": "[INFO] [2017-10-10 17:37:33,682] 1)****************** end
testing RMI connectivity to thedocsmachine/192.168.2.123\n[INFO] [2017-10-10
17:37:33,682] \n"
      }
  ],
  "job_status": {
    "code": 0,
    "message": "Success"
  }
}
```

Example: Get Job Input File Details

The following example shows you how retrieve input file details of a specific job:

Example: Get Job Input File Details

Request:

```
GET /receive/metrics/jobs/483e2a46-1f40-4e8a-9ee2-532f66dc549b/input
```

Response:

```
{
  "jobId": "483e2a46-1f40-4e8a-9ee2-532f66dc549b",
  "name": "My Upload Job"
  "parameters": [{
        "name": "Priority",
        "value": "High"
      },
      {
        "name": "License",
        "value": "True"
      }
  ],
  "input_files": [{
        "name": "allmetrics_production.csv",
        "size": 203945
      }, {
        "name": "Eastmetrics_test.csv",
        "size": 28635
      }, {
        "name": "Normetrics.csv",
        "size": 27812
      }
  ],
}
```

Example: Get Job Log File Details

The following example shows you how retrieve log file details for a specific job:

Example: Get Job Log File Details

Request:

```
GET /receive/metrics/jobs/58419075-6009-4c73-8783-54a49e347384/logs?lines_to_
tail_in_logs=2
```

Response:

```
{
  "jobId": "58419075-6009-4c73-8783-54a49e347384",
  "name": "CPU utilization - upload job",
  "parameters": [{
        "name": "hint",
        "value": "Sample"
      },
      {
        "name": "license",
        "value": "true"
      }
  ],
  "logs": [
      {
        "name": "main.log",
        "contents": "[INFO] [2018-05-10 17:37:33,682] 1)****************** end
testing RMI connectivity to thedocsmachine/192.168.2.123\n[INFO] [2018-05-10
17:37:33,682] \n"
      },
      {
        "name": "logs_load.txt",
        "contents": "[INFO] [2018-05-10 09:45:10,403] Process
host:notApplicable, converter type:DataImporter\n[INFO] [2018-05-10
09:45:11,355] Audit Converter complete.\n"
      }
  ]
}
```

Example: Get Job Audit Information

The following example shows you how retrieve details of a specific job:

Example: Get Job Audit Information

Request:

```
GET /receive/metrics/jobs/483e2a46-1f40-4e8a-9ee2-532f66dc549b/audit_info
```

Response:

```
{
  "jobId": "483e2a46-1f40-4e8a-9ee2-532f66dc549b",
  "name": "CPU utilization - manual upload",
  "parameters": [{
        "name": "region",
        "value": "north"
      },
      {
        "name": "license",
        "value": "true"
      }
  ],
  "audit_info": {
      "audit_name": "483e2a46-1f40-4e8a-9ee2-532f66dc549b",
      "audit_date": "2018-05-13 09:23:37.000",
      "audit_path": "output\002215185311_M0511_483e2a46-1f40-4e8a-9ee2-
532f66dc549b",
      "targets_audited": 323,
      "targets_failed": 0,
      "audit_end_date": "2018-05-13 09:45:11.333",
      "load_date": "2018-05-13 09:52:23.930",
      "load_end_date": "2018-05-13 09:52:26.577",
      "load_status": "Loaded"
  },
}
```

Example: Upload a File

The following example shows you how to upload a file to an existing job:

Example: Upload a File

Request:

```
POST /receive/metrics/jobs/8e5fafe4-1709-46b5-9bdb-
f0a12ed4910d?execute=false&time=00:00
```

The request body contains a multipart/form-data file with the `file` key:

```
{
  "file": <file content>
}
```

Response:

```
{
  "jobId": "8e5fafe4-1709-46b5-9bdb-f0a12ed4910d",
  "name": "sample-job",
  "job_status": {
      "code": 0,
      "message": "File uploaded successfully."
   }
}
```

Example: Download Files

The following example shows you how to download files from an existing job:

Example: Download Files

Request:

```
GET /receive/metrics/jobs/8e5fafe4-1709-46b5-9bdb-f0a12ed4910d/download/files
```

Response:

The response is an octet-stream with the Content-Disposition suggested filename in the following format:

```
<jobId>_download.zip
```

The downloaded zipped file contains all the files, designated as "download", for the specified job.

Example: Download Logs

The following example shows you how to download log files from an existing job:

Example: Download Logs

Request:

```
GET /receive/metrics/jobs/8e5fafe4-1709-46b5-9bdb-f0a12ed4910d/logs/files
```

Response:

The response is an octet-stream with the Content-Disposition suggested filename in the following format:

```
<jobId>_logs.zip
```

Example: Delete Input Files

The following example shows you how to delete the input files from an existing job:

Example: Delete Input Files

Request:

```
DELETE /receive/metrics/jobs/483e2a46-1f40-4e8a-9ee2-
532f66dc549b/contents/input
```

Response:

```
{
  "message": "Contents of directory deleted successfully.",
  "status": 200
}
```

Example: Delete Log Files

The following example shows you how to delete log files from an existing job:

Example: Delete Log Files

Request:

```
DELETE /receive/metrics/jobs/483e2a46-1f40-4e8a-9ee2-532f66dc549b/contents/logs
```

Response:

```
{
  "message": "Contents of directory deleted successfully.",
  "status": 200
}
```

Example: Delete Job

The following example shows you how to delete a job:

Example: Delete Job

Request:

```
DELETE /receive/metrics/jobs/883e2a46-1f40-4e8a-9ee2-532f66dc549b
```

Response:

```
{
  "message": "Job deleted successfully.",
  "status": 200
}
```

Example: Update Job Parameters

The following example shows you how to update parameters for an existing job:

Example: Update Job Parameters

Request:

```
PUT /receive/metrics/jobs/acb37a49-cd7c-4bf1-a94b-884fdce11ed7/parameters
```

JSON Request Body:

```
[
      {"name": "License",
       "value": "true"
      },
      {"name": "Priority",
       "value": "low"
      }
]
```

Response:

```
{
  "jobId": "acb37a49-cd7c-4bf1-a94b-884fdce11ed7",
  "name": "SuperDuperJob",
  "parameters": [{
        "name": "License",
        "value": "true"
      },
      {
        "name": "Priority",
        "value": "low"
      }
  ],
  "input_files": [],
  "lines_to_tail_in_logs": -1,
  "logs": [],
  "job_status": {
    "code": 0,
    "message": ""
  }
}
```

Routing Requests

Description

A Routing Request represents a request to route a new workload demand entity, typically a VM, to an appropriate hosting venue.

The Routing Request takes into account scalar and non-scalar workload requirements when determining qualified hosting venues, so that memory, CPU, IO and storage are not overcommitted across all future timeframes.

Hosting Venues

This API supports routing to a mix of full control hosting venues (i.e. infrastructure groups), non-control hosting venues and guest-level hosting venues.

With respect to this API, the difference between these hosting venues is as follows:

- Assessing Hosting Venues—For non-control and guest-level hosting venues, only Fit-for-Purpose checks are performed. The hosting venues that pass the Fit-for-Purpose requirements result in a `hosting_score` of 100% (the mode is assumed `capacity_sensitive`).
- Placement and Option for Placement—The `?recheckHost=true` is ignored for non-control and guest-level hosting venues.
- Auto-Reconciliation—Only applies to full control hosting venues.

Assessing Hosting Venues and Hosts

When assessing multiple hosting venues, the available capacity of each hosting venue is assessed according to cost, capacity and Fit-for-Purpose, using the same logic as described in Routing Requests–Available Capacity Query on page 361. The best placement of the entity is determined by the hosting venue with the highest score (i.e. `hosting_score`). The same three modes (i.e. `?mode=capacity_sensitive`, `?mode=cost_sensitive` and `?mode=cost_and_capacity`) can be specified with the Routing Request, with default defined by configuration setting **API Default Routing Strategy**.

For Routing Requests routed for today, once the best hosting venue is determined, the best host and the best datastores associated with the host are selected (determined by the most available capacity in number of slots and metrics).

Assessing Datastore

If the datastore preference is specified in the Workload disk definition, Densify will try to respect the preferred placement according to the best candidate environment and hosting venue selected. If the preferred datastore is not available in the selected hosting venue, does not satisfy the tier condition, or does not have enough capacity, then the next suitable datastore available in the venue will be used. Regardless if the preferred datastore is used during routing, the Datastore Preference value is still preserved in the Workload disk definition after placement.

Forced Placement

A particular hosting venue may not be a candidate for routing when it does not have the capacity, does not pass the Fit-for-Purpose validation or does not have the required storage tier requirements. With the `?force=true` option, a Routing Request can be forced to route to that hosting venue. Densify will then select the least full host and datastores (taking datastores with lower storage tier priorities if necessary), even if policy limits are violated or capacity is in the negative.

The routing of disks with `?force=true` is performed in the following order:

1. first routes to the best Fit-for-Purpose and capacity choice with available capacity
2. then routes to the best capacity choice with available capacity, with a lower storage tier priority
3. then routes to requested storage tier, regardless of available capacity
4. then routes to the best capacity choice, regardless of requested storage tier

Placement and Option for Placement

When a Routing Request is PLACED, its Workloads are PLACED (see Placement and Option for Placement on page 537 for Workloads and State Diagram—Create Scenario on page 1).

Similar to the Workload object, the individual get requests `GET /routing-requests/<id>` and `GET /routing-requests/<id>/workloads` can be extended with the following option:

- `?recheckHost=true`—to recheck the recommended host and sensor placement for a Workload in PLACED state. If the recommended host/sensor is currently not healthy (i.e. real-time placement is enabled and the monitored host/sensor shows unhealthy), a new placement is provided.

Record Placement

During a re-route cycle, you may want to record the existing datastore that was used before the Workload is unrouted. You can achieve this by using the DELETE operation with the `?record_placement=true` option on the following resources:

- `DELETE /routing-requests/<id>?record_placement=true`—Use this operation to unroute the Workloads associated with the specified Route Request and save the current datastore as the preferred datastore for the re-route cycle.
- `DELETE /routing-requests/<id>/<workload_id>?record_placement=true`—Use this operation to unroute the specified Workload associated with the specified Route Request and save the current datastore as the preferred datastore for the re-route cycle.

The default behavior for the DELETE operation without the `?record_placement` option or with the `?record_placement=false` option is that the preferred datastore (`pref_datastore`) attribute remains unchanged.

When you use the DELETE operation with `?record_placement=true` option on the resources listed above, the following behavior occurs based on the number of disks in a Workload and the preferred datastore attribute for each disk:

record_ placement	Preferred Datastore Condition	Preferred Datastore Attribute Result
true	Preferred datastore attribute is not set for all disks in the Workload.	Current datastore used is saved in the preferred datastore (`pref_datastore`) attribute for each disk.
true	Preferred datastore attribute is set for some disks and not for other disks in the Workload.	Preferred datastore attribute remains unchanged for each disk.
false	Any condition	Preferred datastore attribute remains unchanged for each disk.

***Note**: If a preferred datastore is set, then that setting will override the sensor placement strategy (i.e. balanced/fill and spill).*

Routing Request and Workloads Treated as One

A Routing Request is treated as a single item. The expected date for placement is taken from the Routing Request and not from the Workloads, even if they differ. The Workloads are still placed according to their expected date.

If any of the Workloads within a request cannot be placed and/or booked (because the capacity could not be reserved or the Fit-for-Purpose failed), then the entire request is rejected.

Sensor Placement Strategy

You have two sensor placement options when routing workloads:

- Balancing Strategy—Distributes incoming workloads across all suitable hosting venues with available sensor capacity. Using this option the resource with available sensor capacity that also matches the fit-for-purpose requirements of the incoming workload and has the most available capacity among all matching candidates is selected. Densify attempts to balance workloads across all available hosting venues. This is the default behaviour.
- Fill and Spill Strategy—Fills suitable hosting venues with available sensor capacity with incoming workloads until the resource reaches its threshold. When routing, the workloads with matching fit-for-purpose requirements are placed on the resource with the least capacity, first.

You can specify sensor placement strategy when submitting a routing request via the API or you can set the placement strategy globally using a configuration setting. Contact Support@Densify.com for details of configuring the global setting.

***Note**: If a preferred datastore is set, then that setting will override the sensor placement strategy (i.e. balanced/fill and spill).*

The routing request applies the specified sensor placement strategy as follows:

- Execute all fit for purpose checks and capacity level checks as usual.
- When choosing the best routable device on which to place a booking:
  - If balance is requested, then sort candidate sensors from most to least available capacity (default implementation).
  - If fill is requested, then sort candidate sensors from least to most available capacity.

Sensor Lockout

At the end of each control environment refresh, existing data used for routing bookings is purged and recalculated (sensor recalculation). This data includes:

- Aggregate infrastructure group-level capacity data, including storage per-tier capacity, cluster_sensor_capacity;
- Individual sensor-level capacity;

During the period while this recalculation is performed, routing anomalies can occur. Specifically, while the sensor data is purged and then is temporarily unavailable for a given hosting venue. During routing of incoming workloads, any clusters that are in the state of sensor data recalculation are excluded from consideration as a hosting option.

You can specify sensor lockout behaviour when submitting a routing request via the API or you can set the lockout during recalculation, globally using a configuration setting. Contact Support@Densify.com for details of configuring the global setting.

Auto-Reconciliation

When the actual VM comes online, it is auto-reconciled with the Workload object with the same name as the Workload `name`. The attributes and Workload Profile are inherited or copied from the Workload to the new VM, if not already specified in the VM. For more details on auto-reconciliation, see section *Auto-Reconciliation of Systems* of *Booking Overview* (Help Topic ID 230350).

Resource

```
/routing-requests
```

Supported Operations

Table: Routing Request Supported Operations

Operation	HTTP Method	Input	Output	Description
Get Collection	`GET /routing-requests`	None	Routing Request Collection of [id, href]	Default Sort By is defined as: `?sort_by=expected_date`.
Get Individual	`GET /routing-requests/<`	None	Routing Requests:	Retrieve the Routing Request elements of the specified `id`. If the Routing Request is in BOOKED

Operation	HTTP Method	Input	Output	Description
	id>		Resource Elements on page 353	status, its Workloads are updated to PLACED if the corresponding `expected_date` of each Workload is today (`expected_date` has passed already but corresponding Booking has not EXPIRED).
Get Individual	`GET /routing-requests/<`*id*`>/workloads`	None	Workload Collection of [id, name href]	Retrieve the Workloads of the specified `id`. This request can also be extended with `?details=true` to view all Workload elements. If the Routing Request is in BOOKED status, its Workloads are updated to PLACED if the corresponding `expected_date` of each Workload is today (or `expected_date` has passed already but corresponding Booking has not EXPIRED).
Create Individual	`POST /routing-requests`	Routing Requests: Form Definition on page 356	Routing Requests: Resource Elements on page 353	To create a Routing Request, you can specify the Workloads in one of two ways: ▯ list of Workload ids (and only Workload ids)−with this option, you must create the Workloads using a separate `POST` and all Workloads must be in `UNROUTED` state, not included in any other Routing Request ▯ specify Workload elements creating the Workloads on-the-fly−with this option, you specify the same elements as you would when creating the Workloads with a separate `POST` to Workloads Example: Routing a Simple Immediate Placement on page 357, Example: Creating a Simple Booking for Mar 20, 2022 on page 358, Example: Selecting Placement From Multiple Environments on page 358
Delete Individual	`DELETE /routing-requests/<`*id*`>`	None	None	When the Routing Request is deleted, the Workloads may or may not be auto-deleted depending on the expected date. See State Diagram—Delete Scenario on page 1 for details. You can save the current datastore used for each disk in the Workload that is unrouted by using the `record_placement=true` option. See Record Placement on page 349 for details.
Delete Workload from Routing Request	`DELETE /routing-requests/<`*id*`>/<`*workload_id*`>`	None	None	A Workload can be unlinked from its associated Routing Request, so that it can be re-routed to another location through another Routing Request. The status of such a Workload is updated to `UNROUTED`. You can save the current datastore used for each

Operation	HTTP Method	Input	Output	Description
				disk in the Workload that is unrouted by using the `record_placement=true` option. See Record Placement on page 349 for details. If this is the last Workload in the Routing Request, then the Routing Request is also deleted. An error is returned if the specified Workload does not belong to this Routing Request.

Resource Elements

Table: Routing Request Resource Elements

Element	Type	Create/ Mod- (Req)	Sort By	Filter	Description
id, href	strings				See ID, Name and Self Reference (id, name, href) on page 29.
control_ environment	href				This is the link to the associated Control Environment where the workloads are placed. Set if and only if the Routing Request has been routed (i.e. `status` is `BOOKED` or `PLACED`).
expected_date	number	C	S	F	The date that the request is to be routed and the system(s) will arrive, in UTC. The time portion is ignored. If the `expected_date` is not specified, then the request is for immediate placement (i.e. Today's timeframe). The `expected_date` must be the same or later than that specified `expected_date` of all its Workloads. If not, an error is thrown. The `expected_date` should be within the supported Timeline Definition of the Control Console.
infrastructure_ group	href			F	This is the link to the associated Infrastructure Group object where the workloads are placed. Set if and only if the Routing Request has been routed (i.e. `status` is `BOOKED` or `PLACED`). `?force=true` can be appended to the URI to force the routing to the specified hosting venue (see Forced Placement on page 348 for details). The resulting Routing Request will show a status of `PLACED` with a status reason of "`Force to place`". The capacity for the hosting venue will be decremented, but not for the Control Environment.
requester	string	C	S	F	The user who routed this request. If not set for a routing request, is automatically set to the current user. If set to "__

Element	Type	Create/ Mod- (Req)	Sort By	Filter	Description
					`Unknown__`" (case insensitive), the request is rejected.
status	string			F	The status of the request. This can be one of: ▪ "`ANALYZING`"—initial state when the Routing Request is defined and before it is fully processed ▪ "`PLACED`"—on the day that the Routing Request is to be fulfilled, the status of the Routing Request changes to `PLACED` and the Workloads are updated to have links to the host where they should be placed ▪ "`BOOKED`"—all Workloads specified in the Routing Request have been routed and space has been reserved, for the date specified by `expected_date` in the future ▪ "`REJECTED`"—no Workloads in the Routing Request were routed as space could not be reserved or the Fit-for-Purpose validation failed for at least one Workload

status_details	Complex, as specified in the description				Status details returned for a Routing Request in any `status` except "`ANALYZING`". This element is returned only if available, for Routing Requests created or updated.
					▪ `igs_in_scope`—number of Infrastructure Groups that are in scope
					▪ `igs_ffp`—number of Infrastructure Groups in scope that are Fit-for-Purpose
					▪ `igs_ffp_not_accepting_workloads`—number of Infrastructure Groups in scope that are Fit-for-Purpose but not accepting workloads (i.e. closed or never been refreshed)
					▪ `igs_ffp_no_capacity`—number of Infrastructure Groups that are Fit-for-Purpose and accepting workloads (i.e. open and refreshed) but have no capacity (i.e. `"slots"<=0`)
					▪ `ig_rejection_details`—for the Infrastructure Groups counted in `igs_ffp_no_capacity`, this returns an array of each Infrastructure Group and the timeframe of the timeline that it has no capacity (referenced by the `short_name` of the Timeline Tags on page 526). Note that the timeframe returned is the first returned by the API, not necessarily the first timeframe chronologically ordered in which there is no capacity. This array is returned only when the Routing Request is REJECTED.
					
					Here is an example of what is returned:
					`"status_details": {`
					`  "igs_in_scope": 5,`
					`  "igs_ffp": 5,`

Element	Type	Create/ Mod- (Req)	Sort By	Filter	Description

					`"igs_ffp_not_accepting_workloads": 1,`
					`"igs_ffp_no_capacity": 2,`
					`"ig_rejection_details": [`
					`{`
					`"timeline_without_capacity": "60"`
					`"infrastructure_group": {`
					`"id": "0cf490b8-bec5-488c-a793-a1cd7fb1be6e",`
					`"name": "Cluster 4",`
					`"href": "/infrastructure-groups/0cf490b8-bec5-488c-a793-a1cd7fb1be6e"`
					`}`
					`},`
					`// ... *SNIP* of 2nd IG rejection details ...`
					`]`
					`}`

status_reason	string			F	Explanation of the status. This field is populated only when the status is `REJECTED` (or `PLACED` for one reason as noted below), and can take on the following values: - "`Cannot find Qualified Infrastructure Group`"–e.g. insufficient scalar and non-scalar workloads to compare against the Infrastructure Groups or an Infrastructure Group is "`UNAVAILABLE`" - "`Cannot find Infrastructure Group Fit-for-Purpose`"–all Infrastructure Groups are not Fit-for-Purpose - "`Force to place`"–when `?force=true` - "`No data for the expected date selected`" e.g. expected_date is too far in the future and there is no timeframe available - "`Cannot find Qualified Infrastructure Group, Force to place`" - "`Cannot find Qualified Infrastructure Group, No data for the expected date selected`" - "`Force to place, No data for the expected date selected`" - "`Cannot find Qualified Infrastructure Group, Force to place, No data for the expected date selected`"
workloads	[id, name, href]	CM			A link to each Workload that is part of this Routing Request placement or available capacity query.

Form Definition

If you want to specify how the workload should be placed, then you can use a routing request form. You then need to provide one or more of the following additional details.

Table: Routing Request Form Definition

Element	Type	Req	Description

scopes	[control_environment, infrastructure_ groups[name]]
	Where infrastructure_groups[name] is optional
			Represents the environments and hosting venues where you want the workload router to restrict the search. If not specified, all environments will be candidates for routing.
			For example, to specify the Boston environment, and either the Prod or Dev hosting venues of the New York environment:
			`"scopes": [`
			`  {`
			`    "control_environment": "Boston"`
			`  },`
			`  {`
			`    "control_environment": "New York",`
			`    "infrastructure_groups": ["Prod","Dev"]`
			`  }`
			`]`

required_ sensor_ types	`"required_sensor_types": ["datastore","ipaddresspools"]`		You can specify sensor lockout behaviour when submitting a routing request to prevent incoming workloads from being placed while the sensor data is purged,recalculated and is temporarily unavailable for a given hosting venue. The parameter value is set as `"required_ sensor_types": ["datastore","ipaddresspools"]`. See Sensor Lockout and Example: Enabling Sensor Lockout.
sensor_ placement_ strategy	`"sensor_placement_strategy": ["datastore-balance","ipaddresspools-fill"]`		Select a sensor placement strategy for routed workloads.Only routable sensor types can be specified. Valid placement strategies are balance or fill. Multiple sensor types can be specified as a comma separated list. If an invalid sensor type or placement strategy is specified, the parameter is ignored and the default placement strategy is used.

Element	Type	Req	Description
			The parameter value is set as `<sensor type>-<placement strategy>`, i.e. `sensor_placement_strategy=datastores-fill,ip_address_pools-balance`. See Sensor Placement Strategy and Example: Selecting Sensor Placement Strategy.
workloads	[Workloads: Resource Elements on page 541]	✓	An array of Workload definitions for each system to be placed. This could either be the list of existing Workload ids or the Workload definition itself. See Workloads on page 536 for details on defining the Workload. ***Note**: The list of Workload IDs must be specified for the* `available-capacity-query` *to evaluate the Fit-for-Purpose. Fit-for-Purpose validation returns all PASS results when Workload definitions are used.*

Examples

The following examples show the various types of routing requests.

Example: Routing a Simple Immediate Placement

The following example shows you the most basic method to route a request:

Example: Routing a Simple Immediate Placement

Request:

```
POST /routing-requests/
{
  "workloads": [
    {
      "name": "now-vm",
      "catalog_spec": "lin-small-2gb"
    }
  ]
}
```

Response:

```
{
  "id": "d14a773a-561f-416a-a8ea-73dc969a3ad0",
  "status": "ANALYZING",
  "expected_date": "1386824400000",
  "requester": "admin",
  "workloads": [
    {
      "id": "6d3fb176-f808-4794-bd15-4e58b6c00f1f",
      "name": "now-vm",
      "href": "/workloads/6d3fb176-f808-4794-bd15-4e58b6c00f1f"
    }
  ],
  "href": "/routing-requests/d14a773a-561f-416a-a8ea-73dc969a3ad0"
}
```

Example: Creating a Simple Booking for Mar 20, 2022

The following example shows you the most basic method to route a booking request on a specific date:

Example: Creating a Simple Booking for Mar 20, 2016

Request:

```
POST /routing-requests/
{
  "expected_date": 1458518806000,
  "workloads": [
    {
      "name": "group-vm-01",
      "catalog_spec": "lin-medium-4gb",
      "workload_profile": "Medium_Utilization"
    },
    {
      "name": "group-vm-02",
      "catalog_spec": "lin-medium-4gb",
      "workload_profile": "Medium_Utilization"
    },
    {
      "name": "group-vm-03",
      "catalog_spec": "lin-medium-4gb",
      "workload_profile": "Medium_Utilization"
    },
    {
      "name": "group-vm-04",
      "catalog_spec": "lin-medium-4gb",
      "workload_profile": "Medium_Utilization"
    }
  ]
}
```

Example: Selecting Placement From Multiple Environments

The following example shows you how to select a placement from multiple environments:

Example: Selecting a Placement From Multiple Environments

Request:

```
POST /routing-requests/
```

```
{
  "scopes": [
    {
      "control_environment": "Boston"
    },
    {
      "control_environment": "London",
      "infrastructure_groups": ["Prod","Dev"]
    }
  ],
  "workloads": [
    {
      "name": "group-vm",
      "catalog_spec": "lin-medium-4gb",
      "workload_profile": "Memory_Intensive"
    }
  ]
}
```

Example: Selecting Sensor Placement Strategy

The following example shows you how to define a sensor placement strategy:

Example: Defining a Sensor Placement Strategy

Request:

```
POST /routing-requests/
{
  "scopes": [
    {
      "control_environment": "Boston"
    },
    {
      "infrastructure_groups":[Cluster 1", "esxuat02.int.cirba.com"]
    }
  ],

  "workloads": [
    {
      "name": "group-vm",
      "catalog_spec": "lin-medium-4gb",
      "workload_profile": "Memory_Intensive"
    }
  ],
      "sensor_placement_strategy": ["datastore-fill","ipaddresspools-balance"],
}
```

Example: Enabling Sensor Lockout

The following example shows you how to enable sensor lockout to avoid placing workloads on hosting venues for which accurate sensor data is temporaily unavailable:

Example: Enabling ensor Lockout

Request:

```
POST /routing-requests/
{
  "scopes": [
    {
      "control_environment": "Boston"
    },
    {
      "infrastructure_groups":[Cluster 1", "esxuat02.int.cirba.com"]
    }
  ],

  "workloads": [
    {
      "name": "group-vm",
      "catalog_spec": "lin-medium-4gb",
      "workload_profile": "Memory_Intensive"
    }
  ],
     "required_sensor_types": ["datastore","ipaddresspools"],
}
```

Routing Requests—Available Capacity Query

Description

A Routing Request can be posted with an available capacity query before actually creating the Routing Request. This query can be used to determine the best placement based on cost, capacity and Fit-for-Purpose. Workloads can then be manually routed to your preferred hosting venues.

The Routing Request Available Capacity Query accepts UNROUTED or REJECTED Workloads, or REJECTED Routing Requests.

Available capacity is determined using a conservative approach, reserving capacity for Workloads specified in Routing Requests that are not yet placed for all future timeframes and continuing to reserve capacity for outbound Bookings that are still online. Also, the available capacity query returns capacity for new planned hosts and sensors, including those sensors that are shared among open and closed hosting venues. Infinite capacity is calculated for non-control and guest-level environments and hosting venues.

For Fit-for-Purpose, the query returns the Fit-for-Purpose results for the hosting venues and datastores. This includes fit according to software requirements, fit for business rules, fit for storage tiers, etc. A Routing Request cannot be routed to a hosting venue that is not fit, i.e. does not pass the Fit-for-Purpose rules, does not have enough capacity, and does not Allow New VM Bookings.

Note*: The results of this query INCLUDE sensors that do not Allow New VM Bookings.*

The query can be posted by defining a scope of environments and/or hosting venues to consider. In this way, you can narrow your query as needed.

Hosting Venues

This API supports a mix of full control hosting venues (i.e. infrastructure groups), non-control hosting venues and guest-level hosting venues.

With respect to this API, the difference between these venues is as follows:

- Supported Operations—all operations are supported for all three hosting venues except:
  - for non-control and guest-level hosting venues, options `?detailed_sensor_calc=true` and `?detailed_host_calc=true` do not apply
- Resource Elements—only elements that are applicable to the hosting venue is returned in the response. Also, some elements may have special values as non-control and guest-level hosting venues have infinite capacity. Specifically:
  - for non-control and guest-level hosting venues and their environments, element `cei`, `constraint` and `constraint_name` are not returned
  - for non-control and guest-level hosting venues and environments, empty array elements `"subslots": []`, `"sensor_capacity": []`, `"host_capacity": []` are returned
  - for non-control and guest-level hosting venues of infinite capacity, `slots` is set to MAXINT (with a `"hosting_score": 100`) if the hosting venue passes the Fit-for-Purpose requirements and is 0 otherwise (with a `"hosting_score": 0`)
  - for environments of non-control and guest-level hosting venues, `total_slots` is set to MAXINT if at least one of its hosting venues has `slots` set to MAXINT and is 0 otherwise
  - for non-control and guest-level hosting venues, the capacity/cost mode does not apply. The mode is always assumed `capacity_sensitive` and element `"hosting_cost": 0` is returned.

Options for Returned Details

You can extend the query with one or all of the following:

- `?subslots=true` (default is `false`)—to return the slot workload metrics which can be used to determine the primary `constraint` for the available capacity. When set to true, the query returns the elements of the `subslots` array for each hosting venue (even if the Infrastructure Group is closed/blocked), `subslots` and `tiers` array for each sensor type (where applicable), and `subslots` array for each host.
- `?detailed_sensor_calc=true` (default is `false`)—to return sensor metrics using a highly detailed computation of sensor capacity that also consolidates duplicate sensors across Control Environments. The recommendation is to use this flag when trying to determine available capacity at the environment level. If you are only considering available capacity at a hosting venue level,

then the recommendation is not to use this flag. When `?detailed_sensor_calc=true` is not used, the query aggregates all sensor metrics before determining slot counts, instead of aggregating slot counts based on individual sensor metrics.

- `?ffp_enabled=false` (default is `true`)—to return available capacity without Fit-for-Purpose calculations. With this option, all Fit-for-Purpose tests pass. The `"categories": []` is returned empty and the resulting pass status `"status": "PASS"` is returned. The `hosting_score` (described below) is calculated assuming the Fit-for-Purpose tests all pass. This option is used when the Route and Reserve Demand page is first displayed and can be used when generating slot reports.
- `?negative_slots=true` (default is `false`)—to perform capacity calculations and then either return actual slot counts even if negative (when true) or show negative slot counts as 0. This affects element `slots` for the number of Workloads and Sensors that can fit in this hosting venue, element `subslots value` for specific slot and sensor metrics, and element `total_slots` for the total sum of `slots` elements for the hosting venues within the environment. All constraints and sums are calculated first before any values are updated when `?negative_slots=false`. For example, if IG1=10 and IG2=-5, then IG1=10, IG2=-5 and Env=5 are returned with `?negative_slots=true`, and IG1=10, IG2=0 and Env=5 are returned with `?negative_slots=false`. The `constraint` is the same, independent of this `?negative_slots` option.
- `?detailed_host_calc=true` (default is `false`)—to return the slot count, primary constraint and subslot metrics on a per host basis for today. When set to true and if the `expected_date` is today, the query returns the element `host_capacity`.

Capacity Based on Expected Date

This `POST` returns the following capacity information based on the `expected_date` (or today if no `expected_date` is specified). For each hosting venue that matches the request, the number of available slots (i.e. the number of Workloads that can fit including sensor capacity), CEI, `subslot` metrics, Fit-for-Purpose tests and `hosting_score` are returned. The `hosting_score` is a standard metric with a range 0..100 to determine the best routing choice. For each Control Environment that matches the request, `total_slots`, CEI and `subslot` metrics are returned. The `total_slots` and `subslot` metrics of each environment is equal to the total of all slots and `subslot` metrics from the returned list of matching hosting venues, only if sensors are not shared across hosting venues or other environments.

Capacity/Cost Mode Option and Hosting Score

The available capacity query calculates the `hosting_score` as an integer from 0..100 where 0 means no placement and 100 provides the best placement out of the specified scope. There are three modes that are supported, which affect this score. You can either take the default mode as defined by the configuration setting **API Default Routing Strategy** (see Configuration Settings on page 27) or you can specify the mode as part of the query:

- `?mode=capacity_sensitive`—Calculates the `hosting_score` according to the available capacity of the hosting venues. The more capacity available, the higher the score.
- `?mode=cost_sensitive`—Calculates the `hosting_score` according to the cost while ensuring there is enough capacity. The higher the cost, the lower the score. If the `hosting_cost` is 0, then the `hosting_score=0`.
- `?mode=cost_and_capacity`—Calculates the `hosting_score` as the average of the `capacity_sensitive` and `cost_sensitive`. This is the default.

***Note**: If an Infrastructure Group has an* `availability_status=UNAVAILABLE` *or* `BLOCKED_BY_USER`*, has no available slots (i.e.* `slots<=0` *or* `sensor_capacity slots<=0`*), or fails the Fit-for-Purpose tests, then the returned* `hosting_score` *is 0 for that Infrastructure Group. Workloads cannot be routed to such Infrastructure Groups, unless the placement is forced.*

Available Slots

The number of available slots is the number of Workloads (taking into account sensor capacity requirements) that can fit within a hosting venue. The size of each slot is defined by the aggregate of the Workloads specified in the Routing Request. For example, if there are 2 available slots, then that means two Routing Requests with the same Workload requirements can fit.

The available slots is defined in element `slots` and can take on any of the following values (rounded down to the nearest integer):

- MAXINT—applies only to a non-control or guest-level hosting venue, if it passes the Fit-for-Purpose requirements
- positive integer—number of such Workloads that can still fit in the hosting venue. Capacity is still available and is limited by the policy high limits.
- 0—no available capacity, or the Infrastructure Group has an `availability_status` of `"UNAVAILABLE"`. 0 is also returned if the hosting venue does not pass the Fit-for-Purpose validation or if the `expected_date` is past the timeline for that hosting venue. If there is no capacity, option `?negative_slots=false` always updates the slot count to 0.
- negative integer—number of such Workloads that have been over-provisioned to the hosting venue. This number of Workloads should be moved to another hosting venue. There is no capacity left, as defined by the policy high limits.

Notes

If there is no data for the timeframe as specified by the `requested_date` of the Routing Request, the CEI for today and the slot capacity of 0 are returned.

See Querying Available Capacity for Specified Workloads on page 45 for more information on slots.

This query supports inline Workloads having only single disk requirements with no storage tier specified and with no attributes specified.

Resource

```
/routing-requests/available-capacity-query
```

Supported Operations

Table: Routing Requests–Available Capacity Query Supported Operations

Operation	HTTP Method	Input	Output	Description
Query	POST /routing-requests/available-capacity-query	Routing Requests: Form Definition on page 356	[Routing Requests–Available Capacity Query: Resource Elements on page 366]	This request is exactly the same as the `POST` to a Routing Request, except no objects are created with this query (i.e. Routing Request and specified Workloads). Note that when specifying an inline Workload object, the `name` element is not required. This post returns the cost, capacity and Fit-for-Purpose results for the Control Environments and Infrastructure Groups. The `expected_date` should be within the supported Timeline Definition of the Control Console. Otherwise, metrics are not returned (e.g. `"constraint": null`, `"subslots": []` and `"sensor_capacity": []`). ***Note**: The specification of only 1 inline workload and 1 inline disk is supported, with no attributes and no tier specified.* Example: Selecting Placement From Multiple Environments on page 371, Example: With Host Available Capacity on page 376 Example: Obtaining Subslot Available Capacity on page 377, Example: Obtaining Available Capacity for a Catalog Spec on page 386, Example: Obtaining Available Capacity for the Default Catalog Spec with Overrides on page 386

Resource Elements

Table: Routing Requests–Available Capacity Query Resource Elements

Element	Type	Description
control_ environment	Complex, as specified in the Description	The Control Environment that is associated with the array of Infrastructure Groups above. The Control Environment's available capacity is as follows: `id`, `name`, `href`–ID, Name and Self Reference (id, name, href) of the Control Environment `platform`–`platform` of the Control Environment `platform_category`–`platform_category` of the Control Environment `control_type`–Type of environment ("`FULL`", "`NONE`" or "`GUEST_LEVEL`"). `total_slots`–the total sum of the `slots` elements from the returned list of matching Infrastructure Groups for this Control Environment. Set to 0 if there is no data for the timeframe as specified by the `requested_date` of the Routing Request. The total may differ when `?detailed_sensor_calc=true` is used in the query and sensor capacity is consolidated. `cei`–the CEI for the `requested_date` of the Routing Request. Set to Today's CEI if there is no data for the timeframe as specified by the `requested_date` of the Routing Request. `subslots`–the slot calculations returned only if `?subslots=true` is used with the query. This is the total sum of the `subslots` elements from the returned list of matching Infrastructure Groups for this Control Environment. See the `subslots` element of the Infrastructure Group for element details. `sensor_capacity`–if `detailed_sensor_calc=true` is not specified in the query, `sensor_capacity` is the sum of all `sensor_capacity` elements from the returned list of matching Infrastructure Groups for this Control Environment. If `detailed_sensor_calc=true` is specified, the `slots` and metric sums consolidate duplicate sensors across Control Environments. `icon`–icon of the Control Environment.
infrastructure_ groups	[Complex, as specified in the Description]	An array of Infrastructure Groups, matching the `scopes` definition in the query and belonging to the Control Environment specified by `control_environment` below. Each Infrastructure Group is defined by the following array of elements (see Infrastructure Groups on page 292): `id`, `name`, `href`–ID, Name and Self Reference (id, name, href) of the Infrastructure Group `slots`–the number of Workloads (the aggregate specified in the request), that can fit in this Infrastructure Group. This number takes into account the `subslot` metrics, sensor capacity (i.e. `sensor_capacity: slots` if returned) and the host capacity (i.e. `host_capacity: slots` if returned), taking the smallest number of the three. Set to 0 if there is no data for the

Element	Type	Description

		timeframe as specified by the `requested_date` of the Routing Request. If there is no capacity, this number is updated to 0 when `?negative_slots=false` is specified.
		`cei`–the CEI for the `requested_date` of the Routing Request. Set to Today's CEI if there is no data for the timeframe as specified by the `requested_date` of the Routing Request.
		`subslots`–the slot metrics returned only if `?subslots=true` is used with the query. The `value` for a metric indicates the number of Routing Requests (i.e. defined by the total Workload requirement in your Routing Request), that can fit considering that metric alone. The metrics returned depend on platform. If there is no capacity, these numbers are updated to 0 when `?negative_slots=false` is specified. For details of the workloads with descriptions, see *Workload Types* (Help Topic ID 170560). Here is an example of what is returned:
		`"subslots": [`
		`{"name": "Total_Cores", "value": 104},`
		`{"name": "Disk_Utilization_As_Percent", "value": 2606},`
		`{"name": "Disk_Operations", "value": 10500},`
		`{"name": "CPU_Utilization", "value": 59},`
		`{"name": "Mem_Reservation", "value": 2147483647},`
		`{"name": "Net_Utilization_As_Percent", "value": 5480},`
		`{"name": "CPU_Reservation_MHz", "value": 2147483647},`
		`{"name": "Mem_Utilization_As_Pct", "value": 24},`
		`{"name": "Mem_VE_Active_Pct", "value": 114},`
		`{"name": "Network_Packets", "value": 840},`
		`{"name": "Total_Memory", "value": 61},`
		`]`
		This is an example when the capacity cannot be computed (e.g. data is too stale):
		`"subslots": []`

		`constraint`–Workload type with the minimum number of slots (which is the constraint of the `"subslots"` above); null is returned if a constraint cannot be calculated (e.g. expected date is too far into the future, or `"subslots": []` is empty when `?subslots=true` is used). Note that the `constraint` is the same independent of option `?negative_slots`. `constraint_name`–Display name of the `constraint` element. Only returned when `constraint` is not null. `control_type`–Type of hosting venue ("`FULL`", "`NONE`" or "`GUEST_LEVEL`"). `hosting_cost`–Monetary cost for maintaining a host in this Infrastructure Group. `fit_for_purpose`–Fit-for-Purpose tests, which includes a `categories` array of tests and `status`. `status` is `"FAIL"` if the `status` of at least one category is `"FAIL"`; `"INCOMPLETE"` if the `status` of at least one category

Element	Type	Description
		is "INCOMPLETE"; otherwise, "PASS" (when all categories have passed). This status must be "PASS" before the Infrastructure Group is considered for routing. Each category within the categories array is defined as: • name–name or description of the category • test–an array of tests for the category in the form name, status, status_reason. status is either "PASS" (test passed), "INCOMPLETE" (e.g. a Workload did not have enough information defined) or "FAIL" (test failed). status_reasons is defined only when status is either "INCOMPLETE" or "FAIL", and includes all messages. • status–category summary status. This is "FAIL" if the status of at least one test in the category is "FAIL"; "INCOMPLETE" if the status of at least one test in the category is "INCOMPLETE"; otherwise, "PASS" (when all tests in the category have passed).
		sensor_capacity–the number of sensor slots available for the sensor requirements (i.e. defined by the total Workload requirement in your Routing Request), along with the slot metrics subslots[] if ?subslots=true is used with the query. The slots calculation is the number of such requests that can fit, given the multi-disk requirements. The value calculation in the subslots[] array is the number of such requests that can fit, considering that metric alone (thus providing the metric in contention). If there is no capacity, these numbers are set to 0 when ?negative_slots=false is specified. The sensor_capacity includes sensors that are configured as Do Not Allow New VM Bookings (see *Managing Storage Settings* (Help Topic ID 172390)), as well as sensors of closed hosting venues. For datastores, subslots[] is returned with the totals of all datastore metrics, independent of tier requirements. If no tier capabilities are specified for the Workloads, then only this array with totals is returned. Also for datastores, a tiers[] array is returned (if ?subslots=true) for each tier specified in the Routing Request, with the metrics specific for that tier and slots calculated as the number of such requests that can fit for that tier. For example, if the Routing Request only includes "Gold" tier disks, then only the slot count and metrics for Gold datastores are returned, along with the subslots totals. If there are no sensors audited, the sensor_capacity is returned with "slots": 2147483647. This means that the space is unlimited. Here is an example of what is returned (note that the 107 slot limitation is due to the Total Provisioned Space which has the same 107 value):

		`"sensor_capacity": [`
		`  {`
		`    "type": "phystor",`
		`    "slots": 1456,`
		`    "constraint": "509E681F-0C7A-4193-95B0-`
		`9523C14E0FED",`
		`    "constraint_name": "Total Used Space (MB)",`

Element	Type	Description

		"subslots": [
		// ... *SNIP* of subslot metrics ...
		]
		},
		{
		"type": "datastore",
		"slots": 113,
		"constraint": "B4E008AD-8ECC-43DB-BFA7-6A590873453E",
		"constraint_name": "Total Used Space (MB)",
		"subslots": [
		{
		"name": "Total Provisioned Space (MB)",
		"value": 159,
		"spec_code": "B89E1E20-F71D-4B70-9179-EEA3C6F81D6C"
		},
		{
		"name": "Number of VMs",
		"value": 180,
		"spec_code": "525F567D-61FC-4A6E-A617-D203FB8E9CDC"
		},
		{
		"name": "Total Used Space (MB)",
		"value": 113,
		"spec_code": "B4E008AD-8ECC-43DB-BFA7-6A590873453E"
		},
		{
		"name": "Health Status",
		"value": 2147483647,
		"spec_code": "C17D631D-C584-48AE-B4F1-77D9EAAAAB59"
		},
		{
		"name": "Total Required Space (MB)",
		"value": 170,
		"spec_code": "98EA05B5-E0E1-4828-A3DD-919C5738D29A"
		}
		],

		"tiers": [{ "name": "Gold", "slots": 113, "subslots": [{ "name": "Total Provisioned Space (MB)", "value": 159, "spec_code": "B89E1E20-F71D-4B70-9179-EEA3C6F81D6C" }, { "name": "Number of VMs", "value": 180, "spec_code": "525F567D-61FC-4A6E-A617-

<table>
<tr><th>Element</th><th>Type</th><th>Description</th></tr>
<tr><td></td><td></td><td>

```
D203FB8E9CDC"
          },
          {
            "name": "Total Used Space (MB)",
            "value": 113,
            "spec_code": "B4E008AD-8ECC-43DB-BFA7-
6A590873453E"
          },
          {
            "name": "Health Status",
            "value": 0,
            "spec_code": "C17D631D-C584-48AE-B4F1-
77D9EAAAAB59"
          },
          {
            "name": "Total Required Space (MB)",
            "value": 170,
            "spec_code": "98EA05B5-E0E1-4828-A3DD-
919C5738D29A"
          }
      }
    ]
  }
]
```

This is an example where the environment or the hosting venue does not have any sensors, when the capacity cannot be computed (e.g. data is too stale):

```
"sensor_capacity": []
```

</td></tr>
<tr><td></td><td></td><td>

- host_capacity–provides the total slot count across the hosts within the infrastructure group. Also, provides the slot count, primary constraint and subslot metrics on a per host basis.

The slots calculation is the sum of the host slots of the request (i.e. defined by the total Workload requirement in the Routing Request). Negative slot counts are not included in the sum (e.g. if host1 has 5 and host2 has -10, then slots will be set to 5).

The hosts array lists all the hosts within the infrastructure group, along with the primary constraint and subslots array of each host.

- name–Name of the host.
- constraint–Primary workload constraint (i.e. the name of the workload having the lowest value in the "subslots" array). Note that the constraint is the same independent of option ?negative_slots.
- constraint_name–Display name of the constraint element. Only returned when constraint is not null.
- id–ID of the host name element. Can be used with the Existing Systems object to obtain host specifics. See Existing Systems on page 261.
- slots–Number of Workloads (the aggregate specified in the request), that can fit in this host. This is the value of the constraint in the "subslots" array.
- subslots—Slot workload metrics returned only if ?subslots=true is used with the query. The value for a metric indicates the number of

</td></tr>
</table>

Element	Type	Description
		Routing Requests (i.e. defined by the total Workload requirement in your Routing Request), that can fit considering that metric alone. If there is no capacity, these numbers are updated to 0 when `?negative_slots=false` is specified. For details of the workloads with descriptions, see *Workload Types* (Help Topic ID 170560). See Example: With Host Available Capacity on page 376 for an example.
		`hosting_score`–a standard metric to determine the best routing choice, where the highest score provides the best placement. A value from 0 to 100. See Routing Requests–Available Capacity Query: Description on page 361 for details.

Form Definition

You need to define the workloads and optionally specify Control Environments and Infrastructure Groups for your query. Use the same form definition as that used for the Routing Request (see Routing Requests: Form Definition on page 356).

Examples

Example: Selecting Placement From Multiple Environments

The following example shows you how to select a placement from multiple environments:

Example: Selecting Placement From Multiple Environments

Request:

```
POST /routing-requests/available-capacity-query/?mode=cost_sensitive
{
  "scopes": [
    {
      "control_environment": "Boston"
    },
    {
      "control_environment": "Singapore",
      "infrastructure_groups": ["Assessment","Densify"]
    }
  ],
  "workloads": [
    {
      "name": "group-vm",
      "catalog_spec": "lin-medium-4gb",
      "workload_profile": "Memory_Intensive"
    }
  ]
}
```

Response:

```
[
  {
    "infrastructure_groups": [
      {
        "id": "f7a5ef21-3d0a-442e-b65c-541b0084b950",
        "name": "General App Cloud",
        "slots": 91,
        "cei": 0.5,
        "constraint": "Total_Memory",
        "constraint_name": "Total Memory",
        "href": "/infrastructure-groups/f7a5ef21-3d0a-442e-b65c-541b0084b950",
        "control_type": "FULL"
        "hosting_cost": 66,
        "fit_for_purpose": {
          "categories": [
            {
              "name": "Resource",
              "test": [
                {
                  "name": "Datastore Tier",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Operating Systems",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Guest Sizes",
                  "status_reasons": [],
                  "status": "PASS"
                }
              ],
              "status": "PASS"
            },
            {
              "name": "Security",
              "test": [
                {
                  "name": "Data Jurisdiction",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Compliance",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Encryption and Key Management",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Identity and Access Management",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Security Zones",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Network Isolation",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Intel TXT Support",
                  "status_reasons": [],
                  "status": "PASS"
                }
              ],
              "status": "PASS"
            },
            {
              "name": "Business",
              "test": [
                {
                  "name": "Location",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Departments",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Service Level",
                  "status_reasons": [],
                  "status": "PASS"
                }
              ],
              "status": "PASS"
            },
            {
              "name": "Technical",
              "test": [
                {
                  "name": "Hosting Platform",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Hypervisor",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Maximum Guest I/O",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Operational Environments",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "DPDK",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Software Licenses",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Crypto H/W Acceleration",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "License Groups",
                  "status_reasons": [],
                  "status": "PASS"
                }
                {
                  "name": "Transport Zone",
                  "status_reasons": [],
                  "status": "PASS"
                }
              ],
              "status": "PASS"
            }
          ],
          "status": "PASS"
        },
        "sensor_capacity": [
          {
            "type": "phystor",
            "slots": 254,
            "constraint": "509E681F-0C7A-4193-95B0-9523C14E0FED",
            "constraint_name": "Total Used Space (MB)",
            "subslots": []
          },
          {
            "type": "datastore",
            "slots": 232,
            "constraint": "B4E008AD-8ECC-43DB-BFA7-6A590873453E",
            "constraint_name": "Number of VMs",
            "subslots": []
          }
          {
            "type": "ipaddresspools",
            "slots": 2147483647,
            "constraint": "62d95498-6fdb-4fce-aebe-5f297ca0f7c3",
            "constraint_name": "IP Addresses Assigned",
            "subslots": []
          }
        ],
        "subslots": [],
        "hosting_score": 100
      },
      // ... *SNIP* of Infrastructure Groups...
    ],
    "control_environment": {
      "id": "e6cf1672-77ff-4e7d-9dda-5387b0bc95cc",
      "name": "Boston",
      "platform": "VMWARE",
      "platform_category": "Internal Virtual",
      "control_type": "FULL"
      "total_slots": 302,
      "cei": 0.53,
      "href": "/control-environments/e6cf1672-77ff-4e7d-9dda-5387b0bc95cc",
      "subslots": [],
      "sensor_capacity": [
        {
          "type": "phystor",
          "slots": 233,
          "constraint": "509E681F-0C7A-4193-95B0-9523C14E0FED",
          "constraint_name": "Total Used Space (MB)",
          "subslots": []
        },
        {
          "type": "datastore",
          "slots": 241,
          "constraint": "B4E008AD-8ECC-43DB-BFA7-6A590873453E",
          "constraint_name": "Number of VMs",
          "subslots": []
        }
        {
          "type": "ipaddresspools",
          "slots": 2147483647,
          "constraint": "62d95498-6fdb-4fce-aebe-5f297ca0f7c3",
          "constraint_name": "IP Addresses Assigned",
          "subslots": []
        }
      ],
      "icon": "/control-environments/e6cf1672-77ff-4e7d-9dda-5387b0bc95cc/icon"
    }
  },
  {
    "infrastructure_groups": [
      // ... *SNIP* of Infrastructure Groups Assessment, Densify...
    ],
    "control_environment": {
      // ... *SNIP* of Control Environment Singapore...
    }
  }
]
```

Example: With Host Available Capacity

The following example shows you how to obtain host-level capacity information for the hosts in scope. Using the same example as above, the response highlights the delta in the response:

Example: Selecting Placement From Multiple Environments with Host Capacity

Request:

```
POST /routing-requests/available-capacity-query/?mode=cost_sensitive&detailed_
host_calc=true
{
  "scopes": [
    {
      "control_environment": "Boston"
    },
    {
      "control_environment": "Singapore",
      "infrastructure_groups": ["Assessment","Densify"]
    }
  ],
  "workloads": [
    {
      "name": "group-vm",
      "catalog_spec": "lin-medium-4gb",
      "workload_profile": "Memory_Intensive"
    }
  ]
}
```

Response:

```
[
  {
    "infrastructure_groups": [
      {
        "id": "f7a5ef21-3d0a-442e-b65c-541b0084b950",
        "name": "General App Cloud",
        "slots": 91,
        "cei": 0.5,
        "constraint": "Total_Memory",
        "constraint_name": "Total Memory",
        "href": "/infrastructure-groups/f7a5ef21-3d0a-442e-b65c-541b0084b950",
        "control_type": "FULL"
        "hosting_cost": 66,
        "fit_for_purpose": {
          // ... *SNIP* of fit_for_purpose...
        },
        "sensor_capacity": [
          // ... *SNIP* of sensor_capacity...
        ],
        "subslots": [],
        "host_capacity": {
          "slots": 10,
          "hosts": [
            {
              "name": "esxcrb06.int.Densify.com",
              "constraint": "Mem_Utilization_As_Pct",
              "constraint_name": "Memory Utilization in Percent",
              "id": "69aeab94-6a2f-4708-aaf4-a65ad5674dd3",
              "slots": 3,
              "subslots": []
            },
            // ... *SNIP* of Hosts...
          ]
        },
        "hosting_score": 100
      },
      // ... *SNIP* of Infrastructure Groups...
    ],
    "control_environment": {
      // ... *SNIP* of Control Environment Boston...
    }
  },
  {
    "infrastructure_groups": [
      // ... *SNIP* of Infrastructure Groups Assessment, Densify...
    ],
    "control_environment": {
      // ... *SNIP* of Control Environment Singapore...
    }
  }
]
```

Example: Obtaining Subslot Available Capacity

The following example shows you how to obtain slot metrics. The referenced Workload is defined with one disk with a tier definition of "Gold". A few datastores are defined with "Gold", which are located in the Boston environment. As well, in the Boston environment, two hosting venues are configured with "Do Not Allow" new VM bookings.

As the Singapore environment has no datastores, each Infrastructure Group will return `sensor_capacity: []`. The `slots` and `hosting_score` are calculated ignoring any of the disk requirements.

For the two Infrastructure Groups that are not accepting new bookings, `slots: 0` is returned.

Example: Obtaining Subslot Available Capacity

Request:

```
POST /routing-requests/available-capacity-query/?subslots=true
{
  "expected_date": 1421816400000,
  "scopes": [
    {
      "control_environment": "Boston"
    },
    {
      "control_environment": "Singapore",
      "infrastructure_groups": ["Assessment","Densify"]
    }
  ],
  "workloads": [
    { "id": "0de059a5-8d1a-49a6-8848-765aef4786e5" }
  ]
}
```

Response:

```
[
  {
    "infrastructure_groups": [
      {
        "id": "f7a5ef21-3d0a-442e-b65c-541b0084b950",
        "name": "General App Cloud",
        "slots": 25,
        "cei": 0.78,
        "subslots": [
          // ... *SNIP* subslots ...
        ],
        "constraint": "Total_Memory",
        "constraint_name": "Total Memory",
        "href": "/infrastructure-groups/f7a5ef21-3d0a-442e-b65c-541b0084b950",
        "control_type": "FULL"
        "hosting_cost": 66,
        "fit_for_purpose": {
          "categories": [
            {
              "name": "Resource",
              "test": [
                {
                  "name": "Datastore Tier",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Operating Systems",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Guest Sizes",
                  "status_reasons": [],
                  "status": "PASS"
                }
              ],
              "status": "PASS"
            },
            {
              "name": "Security",
              "test": [
                {
                  "name": "Data Jurisdiction",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Compliance",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Encryption and Key Management",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Identity and Access Management",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Security Zones",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Network Isolation",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Intel TXT Support",
                  "status_reasons": [],
                  "status": "PASS"
                }
              ],
              "status": "PASS"
            },
            {
              "name": "Business",
              "test": [
                {
                  "name": "Location",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Departments",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Service Level",
                  "status_reasons": [],
                  "status": "PASS"
                }
              ],
              "status": "PASS"
            },
            {
              "name": "Technical",
              "test": [
                {
                  "name": "Hosting Platform",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Hypervisor",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Maximum Guest I/O",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Operational Environments",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "DPDK",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Software Licenses",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Crypto H/W Acceleration",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "License Groups",
                  "status_reasons": [],
                  "status": "PASS"
                },
                {
                  "name": "Transport Zone",
                  "status_reasons": [],
                  "status": "PASS"
                }
              ],
              "status": "PASS"
            }
          ],
          "status": "PASS"
        },
        "sensor_capacity": [
          {
            "type": "datastore",
            "slots": 165,
            "constraint": "B4E008AD-8ECC-43DB-BFA7-6A590873453E",
            "constraint_name": "Total Used Space (MB)",
            "subslots": [
              {
                "name": "Total Provisioned Space (MB)",
                "value": 215,
                "spec_code": "B89E1E20-F71D-4B70-9179-EEA3C6F81D6C"
              },
              {
                "name": "Number of VMs",
                "value": 1512,
                "spec_code": "525F567D-61FC-4A6E-A617-D203FB8E9CDC"
              },
              {
                "name": "Total Used Space (MB)",
                "value": 165,
                "spec_code": "B4E008AD-8ECC-43DB-BFA7-6A590873453E"
              },
              {
                "name": "Health Status",
                "value": 2147483647,
                "spec_code": "C17D631D-C584-48AE-B4F1-77D9EAAAAB59"
              },
              {
                "name": "Total Required Space (MB)",
                "value": 234,
                "spec_code": "98EA05B5-E0E1-4828-A3DD-919C5738D29A"
              }
            ],
            "tiers": [
              {
                "name": "Gold",
                "slots": 152,
                "subslots": [
                  {
                    "name": "Total Provisioned Space (MB)",
                    "value": 215,
                    "spec_code": "B89E1E20-F71D-4B70-9179-EEA3C6F81D6C"
                  },
                  {
                    "name": "Number of VMs",
                    "value": 152,
                    "spec_code": "525F567D-61FC-4A6E-A617-D203FB8E9CDC"
                  },
                  {
                    "name": "Total Used Space (MB)",
                    "value": 165,
                    "spec_code": "B4E008AD-8ECC-43DB-BFA7-6A590873453E"
                  },
                  {
                    "name": "Health Status",
                    "value": 0,
                    "spec_code": "C17D631D-C584-48AE-B4F1-77D9EAAAAB59"
                  },
                  {
                    "name": "Total Required Space (MB)",
                    "value": 234,
                    "spec_code": "98EA05B5-E0E1-4828-A3DD-919C5738D29A"
                  }
                ]
              }
            ]
          },
          {
            "type": "phystor",
            "slots": 259,
            "constraint": "509E681F-0C7A-4193-95B0-9523C14E0FED",
            "constraint_name": "Total Used Space (MB)",
            "subslots": [
              {
                "name": "Total Provisioned Space (MB)",
                "value": 467,
                "spec_code": "B92D5BCE-4578-48EA-9B80-70B98DCD07D8"
              },
              {
                "name": "Number of VMs",
                "value": 727,
                "spec_code": "309504CA-0B67-4E01-8707-AFC0CCDCB07B"
              },
              {
                "name": "Total Used Space (MB)",
                "value": 259,
                "spec_code": "509E681F-0C7A-4193-95B0-9523C14E0FED"
              }
            ]
          },
          {
            "type": "ipaddresspools",
            "slots": 2147483647,
            "constraint": "62d95498-6fdb-4fce-aebe-5f297ca0f7c3",
            "constraint_name": "IP Addresses Assigned",
            "subslots": []
          }
        ],
        "hosting_score": 48
      },
      // ... *SNIP* of Infrastructure Groups...
    ],
    "control_environment": {
      "id": "e6cf1672-77ff-4e7d-9dda-5387b0bc95cc",
      "name": "Boston",
      "platform": "VMWARE",
      "platform_category": "Internal Virtual",
      "control_type": "FULL"
      "total_slots": 25,
      "cei": 0.82,
      "href": "/control-environments/e6cf1672-77ff-4e7d-9dda-5387b0bc95cc",
      "subslots": [
        // ... *SNIP* subslots ...
      ],
      "sensor_capacity": [
        {
          "type": "datastore",
          "slots": 278,
          "constraint": "B4E008AD-8ECC-43DB-BFA7-6A590873453E",
          "constraint_name": "Total Used Space (MB)",
          "subslots": [
            {
              "name": "Total Provisioned Space (MB)",
              "value": 374,
              "spec_code": "B89E1E20-F71D-4B70-9179-EEA3C6F81D6C"
            },
            {
              "name": "Number of VMs",
              "value": 332,
              "spec_code": "525F567D-61FC-4A6E-A617-D203FB8E9CDC"
            },
            {
              "name": "Total Used Space (MB)",
              "value": 278,
              "spec_code": "B4E008AD-8ECC-43DB-BFA7-6A590873453E"
            },
            {
              "name": "Health Status",
              "value": 2147483647,
              "spec_code": "C17D631D-C584-48AE-B4F1-77D9EAAAAB59"
            },
            {
              "name": "Total Required Space (MB)",
              "value": 404,
              "spec_code": "98EA05B5-E0E1-4828-A3DD-919C5738D29A"
            }
          ],
          "tiers": [
            {
              "name": "Gold",
              "slots": 165,
              "subslots": [
                {
                  "name": "Total Provisioned Space (MB)",
                  "value": 215,
                  "spec_code": "B89E1E20-F71D-4B70-9179-EEA3C6F81D6C"
                },
                {
                  "name": "Number of VMs",
                  "value": 167,
                  "spec_code": "525F567D-61FC-4A6E-A617-D203FB8E9CDC"
                },
                {
                  "name": "Total Used Space (MB)",
                  "value": 165,
                  "spec_code": "B4E008AD-8ECC-43DB-BFA7-6A590873453E"
                },
                {
                  "name": "Health Status",
                  "value": 0,
                  "spec_code": "C17D631D-C584-48AE-B4F1-77D9EAAAAB59"
                },
                {
                  "name": "Total Required Space (MB)",
                  "value": 234,
                  "spec_code": "98EA05B5-E0E1-4828-A3DD-919C5738D29A"
                }
              ]
            }
          ]
        },
        {
          "type": "phystor",
          "slots": 469,
          "constraint": "509E681F-0C7A-4193-95B0-9523C14E0FED",
          "constraint_name": "Total Used Space (MB)",
          "subslots": [
            {
              "name": "Total Provisioned Space (MB)",
              "value": 840,
              "spec_code": "B92D5BCE-4578-48EA-9B80-70B98DCD07D8"
            },
            {
              "name": "Number of VMs",
              "value": 1482,
              "spec_code": "309504CA-0B67-4E01-8707-AFC0CCDCB07B"
            },
            {
              "name": "Total Used Space (MB)",
              "value": 469,
              "spec_code": "509E681F-0C7A-4193-95B0-9523C14E0FED"
            }
          ]
        },
        {
          "type": "ipaddresspools",
          "slots": 2147483647,
          "constraint": "62d95498-6fdb-4fce-aebe-5f297ca0f7c3",
          "constraint_name": "IP Addresses Assigned",
          "subslots": []
        }
      ],
      "icon": "/control-environments/e6cf1672-77ff-4e7d-9dda-5387b0bc95cc/icon"
    }
  },
  {
    "infrastructure_groups": [
      {
        "id": "9d202491-f84c-405f-92b7-53648e8b4e92",
        "name": "Assessment",
        "slots": 0,
        "cei": 1,
        "subslots": [
          // ... *SNIP* subslots ...
        ],
        "constraint": "Total_Memory",
        "constraint_name": "Total Memory",
        "href": "/infrastructure-groups/9d202491-f84c-405f-92b7-53648e8b4e92",
        "control_type": "FULL"
        "hosting_cost": 0,
        "fit_for_purpose": {
          "categories": [
            // ... *SNIP* Fit-for-Purpose categories ...
          ],
          "status": "PASS"
        },
        "sensor_capacity": [
          // ... *SNIP* sensor_capacity ...
        ],
        "hosting_score": 0
      },
      {
        "id": "ae8c8d5a-0d91-44dc-a679-b7188f2adbdf",
        "name": "Dev",
        "slots": 0,
        "cei": 1,
        "subslots": [
          // ... *SNIP* Fit-for-Purpose categories ...
        ],
        "constraint": "Total_Memory",
        "constraint_name": "Total Memory",
        "href": "/infrastructure-groups/ae8c8d5a-0d91-44dc-a679-b7188f2adbdf",
        "hosting_cost": 0,
        "fit_for_purpose": {
          "categories": [
            // ... *SNIP* Fit-for-Purpose categories ...
          ],
          "status": "PASS"
        },
        "sensor_capacity": [
          // ... *SNIP* sensor_capacity ...
        ],
        "hosting_score": 0
      }
    ],
    "control_environment": {
      "id": "14798cc6-f844-46b2-aa29-fcbcd68a2972",
      "name": "London",
      "platform": "IBM",
      "platform_category": "Internal Virtual",
      "control_type": "FULL"
      "total_slots": 0,
      "cei": 0.86,
      "href": "/control-environments/14798cc6-f844-46b2-aa29-fcbcd68a2972",
      "subslots": [
        // ... *SNIP* subslots ...
      ],
      "sensor_capacity": [
        // ... *SNIP* sensor_capacity ...
      ],
      "icon": "/control-environments/14798cc6-f844-46b2-aa29-fcbcd68a2972/icon"
    }
  }
]
```

The following example output shows you the Fit-for-Purpose test when it fails. This failure is due to the Infrastructure Group not having any datastores with the proper tier definition. In this case, the datastores are all defined as "Default", not "Gold".

Example: Obtaining Available Capacity for a Catalog Spec

Response:

```
[
  {
    "infrastructure_groups": [
      {
        "id": "e59b6ccb-69c6-422b-93b5-e7ca64b61be4",
        "name": "ProdCloud-Apps1",
        "slots": 0,
        "cei": 0.88,
        // ... *SNIP* ...
        "fit_for_purpose": {
          "categories": [
            // ... *SNIP* categories...
            {
              "name": "Resource",
              "test": [
                // ... *SNIP* tests ...
                {
                  "name": "Datastore Tier",
                  "status_reasons": [
                    {
                      "reason": "Required Datastore Tier not supported",
                      "workload_id": "0de059a5-8d1a-49a6-8848-765aef4786e5",
                      "workload_name": "VM-1"
                    }
                  ],
                  "status": "FAIL"
                }
              ],
              "status": "FAIL"
            },
              // ... *SNIP* categories...
          ],
          "status": "FAIL"
        },
        "sensor_capacity": [
          {
            "type": "datastore",
            "slots": 0,
            "constraint": "B89E1E20-F71D-4B70-9179-EEA3C6F81D6C",
            "constraint_name": "Number of VMs",
            "subslots": [
              // ... *SNIP* ...
            ],
            "tiers": [
              {
                "name": "Gold",
                "slots": 0,
                "subslots": [
                // ... *SNIP* ...
```

Example: Obtaining Available Capacity for a Catalog Spec

The following example shows you how to obtain available capacity for a specific catalog specification:

Example: Obtaining Available Capacity for a Catalog Spec

Request:

```
POST /routing-requests/available-capacity-query
{
  "workloads": [
    { "catalog_spec": "win-medium-2gb" }
  ]
}
```

Example: Obtaining Available Capacity for the Default Catalog Spec with Overrides

The following example shows you how to obtain available capacity for the default catalog specification with overrides:

Example: Obtaining Available Capacity for the Default Catalog Spec with Overrides

Request:

```
POST /routing-requests/available-capacity-query
{
  "workloads": [
    {
      "workload_profile": "Memory_Intensive",
      "disks": [
        {
          "name":"SYSTEM",
          "provisioned_space":20480,
          "used_space":5120
        }
      ]
    }
  ]
}
```

Routing Requests–Available Sensor Capacity Query

Description

A Routing Request can be posted with an available sensor capacity query before the Routing Request is created. This query can be used to create reports on sensor capacity (e.g. as used by the Spare VM Capacity Reports).

The POST is similar to POSTs to `/routing-requests` and `/routing-requests/available-capacity-query`, except that it returns the sensor details grouped by sensor type, for each Control Environment in scope (as defined by the Routing Requests: Form Definition on page 356). When specified with `?subslots=true`, metrics are also provided for each sensor and the total metrics for each sensor type. The sensors and metrics are returned according to the `expected_date` of the Routing Request.

This query differs with Routing Requests–Available Capacity Query on page 361 in that the totals returned by this query take into account sensors that are shared across Infrastructure Groups or other Control Environments.

Note*: This query does not take into account tier requirements. Also, this query supports inline Workloads with single disk requirements only.*

Full Control Hosting Venues

Non-control and guest-level hosting venues do not host sensors. This API only returns sensor details for full control hosting venues in scope.

Options for Returned Details

You can extend the query with one or all of the following, as documented in Options for Returned Details on page 362 for the Routing Requests–Available Capacity Query on page 361:

You can extend the query with one or all of the following:

- `?subslots=true` (default is `false`)—to return the slot sensor metrics which can be used to determine the primary `constraint` for the available capacity. When set to true, the query returns the metric elements of the `subslots` array for each sensor and each sensor type.
- `?negative_slots=true` (default is `false`)—to perform capacity calculations and then either return actual slot counts even if negative (when true) or show negative slot counts as 0. This affects element `slots` for the number of Sensors that can fit in the Control Environments and Infrastructure Groups in scope, and element `subslots value` for specific sensor metrics. All constraints and sums are calculated first before any values are updated when `?negative_slots=false`.
- `?omit_closed_sensors=true` (default is `false`)—to return only sensors (entries of the `sensors` array) associated with open Infrastructure Groups. When true, the aggregated `slots` count reflect accessible capacity only (i.e. total of all `slots` within the returned `sensors` array).

Resource

```
/routing-requests/available-sensor-capacity-query
```

Supported Operations

Table: Routing Requests–Available Capacity Query Supported Operations

Operation	HTTP Method	Input	Output	Description
Query	POST /routing-requests/available-sensor-capacity-query	Routing Requests: Form Definition on page 356	[Routing Requests–Available Sensor Capacity Query:	This request is exactly the same as the `POST` to a Routing Request, except no objects are created with this query (i.e. Routing Request and specified Workloads). This post returns the slot counts and metrics for the sensors in the Control Environments in scope.

Operation	HTTP Method	Input	Output	Description
			Resource Elements on page 390]	Example: Obtaining Sensor Capacity on page 391

Resource Elements

Table: Routing Requests–Available Sensor Capacity Query Resource Elements

Element	Type	Description
sensors	[Complex, as specified in the Description]	`sensors` is an array of sensor details, for the given sensor type. If `?omit_closed_sensors=true`, then only sensors associated with open Infrastructure Groups are returned. A sensor entry is defined for each sensor, as: - `name`–name of the sensor. Same as the `name` in `summaries`. - `slots`–number of Workloads (the aggregate specified in the request), that can fit in this sensor. - `subslots`–array of sensor metrics. The `subslots` are returned only when `?subslots=true` is used with the query. It consists of: • `name`–UI field label of the metric. • `value`–value given to the metric. For a metric that is not capacity related, the value is either MAXINT if it is below its threshold (i.e. OK), or 0 if it is above its threshold (i.e. not OK). • `spec_code`–an internal id. - `summaries`–array of sensor details. When the sensor spans Control Environments, one entry is returned for each. An entry consists of: • `id`, `name`, `href`–see ID, Name and Self Reference (id, name, href) on page 29. The `id` and `href` are different, but the `name` and `host_name` are the same among the different entries. • `host_name`–The host name of the sensor.
slots	number	The number of Workloads (the aggregate specified in the request), that can fit in this sensor type. If `?omit_closed_sensors=true`, then this number reflects accessible capacity only.
constraint	spec_code	`spec_code` of the workload type with the minimum number of slots (which is the constraint of the above `"subslots"` of the `"sensors"` array); null is returned if a constraint cannot be calculated (e.g. expected date is too far into the future, or `"subslots": []` is empty when `?subslots=true` is used); only returned when `?subslots=true`.
constraint_name	string	Display name of the `constraint` element. Only returned when `constraint` is not null.
subslots	[name, value, spec_code]	`subslots` is an array of total sensor metrics for all the sensors of a given sensor type. These totals take into account sensors that are shared across

Element	Type	Description
		Infrastructure Groups or other Control Environments. The `subslots` are returned only when `?subslots=true` is used with the query. These totals are calculated without capping individual values. A subslot entry is defined for each metric, as: - `name`—UI field label of the metric. - `value`—value given to the metric. For a metric that is not capacity related, the value is either MAXINT if it is below its threshold (i.e. OK), or 0 if it is above its threshold (i.e. not OK). - `spec_code`—an internal id.
type	string	The sensor type (e.g. "phystor"). See Sensors including Datastores, Physical Storage, Resource Pools on page 402.

Form Definition

You need to define the workloads and optionally specific Control Environments and Infrastructure Groups for your query. Use the same form definition as that used for the Routing Request (see Routing Requests: Form Definition on page 356).

Examples

Example: Obtaining Sensor Capacity

The following example shows you how to obtain the available sensor capacity:

Example: Obtaining Sensor Capacity

Request:

```
POST /routing-requests/available-sensor-capacity-query/?subslots=true
{
  "expected_date" : 1454187586000,
  "workloads": [ {
    "id": "1fca8f15-90ef-4a69-b35a-ce82c82b9718"}
  ]
}
```

Response:

```
[
  {
    "type": "phystor",
    "slots": 173,
    "sensors": [
      {
        "name": "vmax20-a1-p4-lun26",
        "slots": 18,
        "subslots": [
          {
            "name": "Total Provisioned Space (MB)",
            "value": 18,
            "spec_code": "B92D5BCE-4578-48EA-9B80-70B98DCD07D8"
          },
          {
            "name": "Number of VMs",
            "value": 770,
            "spec_code": "309504CA-0B67-4E01-8707-AFC0CCDCB07B"
          },
          {
            "name": "Total Used Space (MB)",
            "value": 41,
            "spec_code": "509E681F-0C7A-4193-95B0-9523C14E0FED"
          }
        ],
        "summaries": [
          {
            "id": "e94cf103-73e1-cba7-a0f2-cef0887640cc",
            "name": "vmax20-a1-p4-lun26",
            "href": "/sensors/physical_storage/e94cf103-73e1-cba7-a0f2-
cef0887640cc",
            "host_name": "phys_vmax20-a1-p4-lun26"
          }
        ]
      },
      // ... *SNIP* of other sensors of type phystor...
    ],
    "constraint": "B92D5BCE-4578-48EA-9B80-70B98DCD07D8",
    "constraint_name": "Total Provisioned Space (MB)",
    "subslots": [
      {
        "name": "Total Provisioned Space (MB)",
        "value": 173,
        "spec_code": "B92D5BCE-4578-48EA-9B80-70B98DCD07D8"
      },
      {
        "name": "Number of VMs",
        "value": 5327,
        "spec_code": "309504CA-0B67-4E01-8707-AFC0CCDCB07B"
      },
      {
        "name": "Total Used Space (MB)",
        "value": 358,
        "spec_code": "509E681F-0C7A-4193-95B0-9523C14E0FED"
      }
    ]
  },
  // ... *SNIP* of datastores...
]
```

Routing Requests–Candidate Infrastructure Groups

Description

A Routing Request can be posted with a Candidate Infrastructure Group query. This query is similar to `/routing-requests/available-capacity-query`, but returns only the hosting venues that pass the Fit for Purpose checks from the specified scope.

The specifics for the Routing Requests—Candidate Infrastructure Groups are:

- same input and output format as Routing Requests—Available Capacity Query
- returns only the subset of hosting venues that pass the Fit for Purpose checks from the specified scope
- returns only the test element as follows, as all tests have "PASS" status

```
"test": [
  {
    "status_reasons": [],
    "status": "PASS"
  }
]
```

- supports options `mode`, `subslots`, `detailed_sensor_calc`, `?negative_slots=true` and `detailed_host_calc=true` but not `ffp_enabled` (which is ignored if specified)

Note that this query returns those hosting venues that pass the Fit for Purpose checks. Hosting venues that do not have enough available capacity, but that pass the Fit for Purpose checks, are also returned.

Options for Returned Details

You can extend the query with one or all of the following, as documented in Options for Returned Details on page 362 for the Routing Requests–Available Capacity Query on page 361:

- `?subslots=true`
- `?detailed_sensor_calc=true`
- `?negative_slots=true`
- `?detailed_host_calc=true`

Capacity/Cost Mode Option and Hosting Score

You can extend the query by specifying a mode to calculate the hosting score, as documented in Capacity/Cost Mode Option and Hosting Score on page 363 for the Routing Requests–Available Capacity Query on page 361:

Resource

```
/routing-requests/candidate-infrastructure-groups
```

Supported Operations

Table: Routing Requests–Candidate Infrastructure Groups Supported Operations

Operation	HTTP Method	Input	Output	Description
Query	POST /routing-requests/candidate-infrastructure-groups	Routing Requests: Form Definition on page 356	[Routing Requests–Available Capacity Query: Resource Elements on page 366]	This request is exactly the same as the `POST` to a Routing Request Available Capacity Query, with the exceptions noted above.

Routing Requests–Constraint Resource Query

Description

The Routing Request can be posted with `constraint-resource-query` to help understand why workloads cannot be routed.

The syntax of the POST is the same as that of the Routing Request Available Capacity Query. The Routing Request Constraint Resource Query accepts UNROUTED or REJECTED Workloads, or REJECTED Routing Requests.

The elements returned are the same as that for the Routing Request Available Capacity Query; however, only the failed Fit-for-Purpose checks (element `fit_for_purpose`) and only the slot metrics having 0 or negative capacity (i.e. metrics of `subslots[]`, `sensor_capacity: subslots[]`, `sensor_capacity: tiers: subslots[]` and `host_capacity: hosts: subslots[]`) are returned.

Options for Returned Details

You can extend the query with one or all of the following, as documented in Options for Returned Details on page 362 for the Routing Requests–Available Capacity Query on page 361:

- `?subslots=true`—if true, only the 0 or negative capacity slot metrics are returned.
- `?detailed_sensor_calc=true`

- ?ffp_enabled=false—if true, only the failed Fit-for-Purpose checks are returned; if false, returns the same elements as described for the Available Capacity Query including an adjusted hosting_score.
- ?negative_slots=true
- ?detailed_host_calc=true—if true, only the hosts that have a subslots[] entry are included and in turn, only the subslots[] metric entries with 0 or negative capacity are included. See Routing Requests–Constraint Resource Query on page 400.

Capacity/Cost Mode Option and Hosting Score

There are three modes that are supported, which can be specified as part of the query. For details, see Capacity/Cost Mode Option and Hosting Score on page 363 for the Routing Requests–Available Capacity Query on page 361.

Resource

```
/routing-requests/constraint-resource-query
```

Supported Operations

Table: Routing Requests–Constraint Resource Query Supported Operations

Operation	HTTP Method	Input	Output	Description
Query	POST /routing-requests/constraint-resource-query	Routing Requests: Form Definition on page 356	[Routing Requests–Available Capacity Query: Resource Elements on page 366]	This request is exactly the same as the POST to a Routing Requests Available Capacity Query, which is the same as the Routing Request except no objects are created (i.e. Routing Request and specified Workloads). Note that when specifying an inline Workload object, the name element is not required. The Routing Request Constraint Resource Query accepts only UNROUTED or REJECTED Workloads. Otherwise, an error is returned. This post returns the same elements as the Routing Request Available

Operation	HTTP Method	Input	Output	Description
				Capacity Query, except only the failed Fit-for-Purpose checks, and only the 0 or negative capacity slot metrics are returned. ***Note**: The specification of only 1 inline workload and 1 inline disk is supported.*
Get Individual	`GET /routing-requests/constraint-resource-query/<id>`	None	Routing Requests–Available Capacity Query: Resource Elements on page 366	Returns the Routing Request Constraint Resource Query elements of the specified Routing Request `id`. The Routing Request must be in REJECTED status. Otherwise, an error is returned.

Examples

Example: Getting the Constraints for Rejected Workloads

The following example shows you why Workloads were rejected.

Example: Getting Constraints for REJECTED Workloads

Request:

```
POST /routing-requests/constraint-resource-query/?subslots=true
{
  "scopes": [
    {
      "control_environment": "Singapore",
      "infrastructure_groups": ["Assessment","Densify"]
    }
  ],
  "expected_date":1454367388000,
  "requester":"John Doe",
  "workloads": [
    {
      "id": "683ea361-e35a-4882-a241-cdeb17b0a666",
      "id": "8e8d7a35-ea05-419f-9428-68b4c9a92872"
    }
  ]
}
```

Response:

```
[
  {
    "infrastructure_groups": [
      {
        "id": "f7a5ef21-3d0a-442e-b65c-541b0084b950",
        "name": "Assessment",
        "slots": 0,
        "cei": 1.15,
        "subslots": [
          {
            "name": "Mem_Utilization_As_Pct",
            "value": 0
          },
          {
            "name": "Total_Memory",
            "value": 0
          }
        ],
        "constraint": "Mem_Utilization_As_Pct",
        "constraint_name": "Memory Utilization in Percent",
        "href": "/infrastructure-groups/f7a5ef21-3d0a-442e-b65c-541b0084b950",
        "control_type": "FULL"
        "hosting_cost": 0,
        "fit_for_purpose": {
          "categories": [
            {
              "name": "Resource",
              "test": [
                {
                  "name": "Operating Systems",
                  "status_reasons": [
                    {
                      "reason": "OS not allowed",
                      "workload_id": "b56e78eb-9dd9-42b5-80cd-32fcf44771e7",
                      "workload_name": "SysDec141"
                    },
                  ],
                  "status": "FAIL"
                }
              ],
              "status": "FAIL"
            }
          ],
          "status": "FAIL"
        },
        "sensor_capacity": [
          {
            "type": "datastore",
            "slots": 0,
            "constraint": "98EA05B5-E0E1-4828-A3DD-919C5738D29A",
            "constraint_name": "Total Required Space (MB)",
            "subslots": [],
            "tiers": [
              {
                "name": "SilverT2",
                "slots": 0,
                "subslots": [
                  {
                    "name": "Total Provisioned Space (MB)",
                    "value": 0,
                    "spec_code": "B89E1E20-F71D-4B70-9179-EEA3C6F81D6C"
                  },
                  {
                    "name": "Total Used Space (MB)",
                    "value": 0,
                    "spec_code": "B4E008AD-8ECC-43DB-BFA7-6A590873453E"
                  },
                  {
                    "name": "Number of VMs",
                    "value": 0,
                    "spec_code": "525F567D-61FC-4A6E-A617-D203FB8E9CDC"
                  },
                  {
                    "name": "Health Status",
                    "value": 0,
                    "spec_code": "C17D631D-C584-48AE-B4F1-77D9EAAAAB59"
                  },
                  {
                    "name": "Total Required Space (MB)",
                    "value": 0,
                    "spec_code": "98EA05B5-E0E1-4828-A3DD-919C5738D29A"
                  }
                ]
              }
            ]
          }
        ],
        "hosting_score": 0
      },
      // ... *SNIP* of Infrastructure Groups...
    ],
    "control_environment": {
      "id": "e6cf1672-77ff-4e7d-9dda-5387b0bc95cc",
      "name": "Singapore",
      "platform": "VMWARE",
      "platform_category": "Internal Virtual",
      "control_type": "FULL"
      "total_slots": 0,
      "cei": 1.17,
      "href": "/control-environments/e6cf1672-77ff-4e7d-9dda-5387b0bc95cc",
      "subslots": [
        {
          "name": "Mem_Utilization_As_Pct",
          "value": 0
        },
        {
            "name": "Total_Memory",
            "value": 0
        }
      ],
      "sensor_capacity": [
        {
          "type": "datastore",
          "slots": 0,
          "constraint": "98EA05B5-E0E1-4828-A3DD-919C5738D29A",
          "constraint_name": "Total Required Space (MB)",
          "subslots": [],
          "tiers": [
            {
              "name": "SilverT2",
              "slots": 0,
              "subslots": [
                {
                  "name": "Total Provisioned Space (MB)",
                  "value": 0,
                  "spec_code": "B89E1E20-F71D-4B70-9179-EEA3C6F81D6C"
                },
                {
                  "name": "Total Used Space (MB)",
                  "value": 0,
                  "spec_code": "B4E008AD-8ECC-43DB-BFA7-6A590873453E"
                },
                {
                  "name": "Number of VMs",
                  "value": 0,
                  "spec_code": "525F567D-61FC-4A6E-A617-D203FB8E9CDC"
                },
                {
                  "name": "Health Status",
                  "value": 0,
                  "spec_code": "C17D631D-C584-48AE-B4F1-77D9EAAAAB59"
                },
                {
                  "name": "Total Required Space (MB)",
                  "value": 0,
                  "spec_code": "98EA05B5-E0E1-4828-A3DD-919C5738D29A"
                }
              ]
            }
          ]
        }
      ],
      "icon": "/control-environments/e6cf1672-77ff-4e7d-9dda-5387b0bc95cc/icon"
    }
  }
]
```

Example: Getting ?detailed_host_calc Details

The following example shows the snippet of the response when `?detailed_host_calc=true` is used.

Example: Getting ?detailed_host_calc Details

Response:

```
// ... *SNIP* of rest ...
"host_capacity": {
  "slots": 10,
  "hosts": [
    {
      "name": "esxcrb07.int.Densify.com",
      "constraint": "Mem_Utilization_As_Pct",
      "constraint_name": "Memory Utilization in Percent",
      "id": "a4373e0f-2553-4755-baec-5acf7a745af1",
      "slots": 0,
      "subslots": [
        {
          "name": "Mem_Utilization_As_Pct",
          "value": 0
        }
      ]
    }
  ]
},
// ... *SNIP* of rest ...
```

Sensors including Datastores, Physical Storage, Resource Pools

Description

Densify provides a framework that allows additional analytics to be easily added to a Densify installation as sensors. For example, sensors can be added to include the analytics of application data or network data. Currently, Densify supports three sensors, including that for datastores (i.e. type is `datastores`), physical storage (i.e. type is `physical_storage`) and resource pools (i.e. type is `resource_pools`). To define your own sensors, contact Densify Technical Services.

With the sensors API, the list of sensor types and sensor details are available for the Control Environments, so that Workloads can be routed appropriately. The details are provided at each defined timeframe of the timeline.

Also see Routing Requests on page 347 for details on sensor considerations when creating routing requests.

Resource

```
/sensors
/sensors/<type>
```

Supported Operations

Table: Sensor Supported Operations

Operation	HTTP Method	Input	Output	Description
Read Collection	GET /sensors	None	List of sensor types defined by [name, href [[,disk_sensor_link_attribute_id],sensor_link_attribute_id], sorted by name.	List of sensor types defined by the Control Console. The `disk_sensor_link_attribute_id` and `sensor_link_attribute_id` are used to determine the sensor display names for placement, when a Booking or Workload object maps to sensors. For example, see the Workloads' `disks` element disks on page 543. Sort By and Filter-Metadata are not supported. Example: Getting Sensor Types on page 405

Table: Sensor Type Supported Operations

Operation	HTTP Method	Input	Output	Description
Get Collection	GET /sensors/<*type*>	None	Sensors of specified <*type*> collection of [name, id, href, host_name]	The list of sensors of specified <*type*>, in ascending order by `name`. Filter-Metadata is supported. Example: Getting a Collection of Datastore Sensors on page 405
Get Individual	GET /sensors/<*type*>/<*id*>	None	Sensors including Datastores, Physical Storage, Resource Pools: Resource Elements on page 403	The details of the specified sensor. Example: Getting an Individual Datastore Sensor on page 406

Resource Elements

Table: Sensor Type Resource Elements

Element	Type	Filter	Description
id, name, href	string	F by `name`	See ID, Name and Self Reference (id, name, href) on page 29.

Element	Type	Filter	Description
control_ environment	id, name, platform, platform_ category, control_type, href, icon	F by `control_ environment_id`	The Control Environment where this sensor belongs.

data	*timeline*: metrics [value, name, high_ limit, low_ limit, full_ limit]		Sensor type-specific metrics for each timeframe that has data, as follows:
			For each `short_name` of the Timeline Tags on page 526:
			`metrics[`
			• `value`–value given to the metric
			• `name`–UI field label of the metric
			• `high_limit`–high threshold
			• `low_limit`–low threshold
			• `full_limit`–full limit defining 100% capacity
			`]`
			Limits of -1 signify that the limit has not been defined.
			See example below.

host_name	string		The entity ID of the sensor (unique).
infrastructure_ groups	id, name, href	F by `infrastructure_ group_id`	The Infrastructure Groups where this sensor belongs.
sensor_type	string		The sensor type, which is "`datastore`", "`phystor`" or "`resourcepool`".
start_date	number	F	The date this sensor is online in UTC. If the sensor is a booking, then this element is the `expected_date` of the Booking. Otherwise, for all sensors that are not bookings this date is today as defined by the timeline. When filtering on `start_date_from` or `start_date_ to`, all sensors coming online that satisfy this filter are returned, including all datastore sensors that are not bookings. When both `start_date_from` and `start_ date_to` are specified, only the sensors coming online that satisfy this filter are returned.

Examples

Example: Getting Sensor Types

The following example shows you how to get sensor types:

Example: Getting Sensor Types

Request:

```
GET /sensors
```

Response:

```
[
  {
    "name": "Datastores",
    "href": "/sensors/datastores",
    "disk_sensor_link_attribute_id": "attr_DiskDatastoreLink",
    "sensor_link_attribute_id": "VE_LOCATION_DATASTORE"
  },
  {
    "name": "IP Address Pools",
    "href": "/sensors/ip_address_pools",
    "sensor_link_attribute_id": "attr_AddressPoolLink"
  },
  {
    "name": "Physical Storage",
    "href": "/sensors/physical_storage",
    "disk_sensor_link_attribute_id": "attr_DiskPhysicalStorageLink",
    "sensor_link_attribute_id": "attr_PhysicalStorageLink"
  },
  {
    "name": "Resource Pools",
    "href": "/sensors/resource_pools",
    "sensor_link_attribute_id": "VE_VMWARE_RESOURCEPOOL"
  }
]
```

Example: Getting a Collection of Datastore Sensors

The following example shows you how to obtain the collection of current datastores:

Example: Getting a Collection of Datastore Sensors

Request:

```
GET /sensors/datastores
```

Response:

```
[
  {
    "id": "13b64d47-d641-4c91-918e-357212ce5b22",
    "name": "EMC2-Datastore-B",
    "href": "/sensors/datastores/13b64d47-d641-4c91-918e-357212ce5b22"
    "host_name": "bac5eda7-e4b4-4c79-9e36-4b9d800a6dd2"
  },
  {
    "id": "1d156284-4d11-4565-9fbe-a8458723c932",
    "name": "ESXCRB11 Datastore",
    "href": "/sensors/datastores/1d156284-4d11-4565-9fbe-a8458723c932"
    "host_name": "277b6bc4-c528-478e-91f8-e87929d35eec"
  },
  // ... *SNIP* ...
]
```

Example: Getting an Individual Datastore Sensor

The following example shows you how to get an individual datastore sensor:

Example: Getting an Individual Datastore Sensor

Request:

```
GET /sensors/datastores/13b64d47-d641-4c91-918e-357212ce5b22
```

Response:

```
{
  "id": "13b64d47-d641-4c91-918e-357212ce5b22",
  "name": "esx-lun-s121",
  "data": {
    "90": {
      "metrics": [
        {
          "value": 13,
          "name": "Number of VMs",
          "high_limit": 75,
          "low_limit": -1,
          "full_limit": -1
        },
        {
          "value": 614144,
          "name": "Capacity (MB)",
          "high_limit": -1,
          "low_limit": -1,
          "full_limit": -1
        },
        {
          "value": 282533.38,
          "name": "Total Required Space (MB)",
          "high_limit": 100,
          "low_limit": 20,
          "full_limit": -1
        },
        {
          "value": 246278.88,
          "name": "Total Used Space (MB)",
          "high_limit": 70,
          "low_limit": 20,
          "full_limit": 100
        },
        {
          "value": 5,
          "name": "Number of Hosts",
          "high_limit": -1,
          "low_limit": -1,
          "full_limit": -1
        },
        {
          "value": 372822.72,
          "name": "Total Provisioned Space (MB)",
          "high_limit": 300,
          "low_limit": 20,
          "full_limit": -1
        }
      ]
    }
    // ... *SNIP* of other timeframes ...
  }
  "href": "/sensors/datastores/00550cc8-b473-4804-9947-20744f35f164",
  "host_name": "bac5eda7-e4b4-4c79-9e36-4b9d800a6dd2",
  "control_environment": {
    "id": "8db102f8-54ef-4d7c-a31d-df98315e48ce",
    "name": "Toronto",
    "platform": "VMWARE",
    "platform_category": "Internal Virtual",
    "control_type": "FULL"
    "href": "/control-environments/8db102f8-54ef-4d7c-a31d-df98315e48ce"
    "icon": "/control-environments/8db102f8-54ef-4d7c-a31d-df98315e48ce/icon"
  },
  "sensor_type": "datastore",
  "start_date": 1403064000000,
  "infrastructure_groups": [
    {
      "id": "6b153341-95b6-4794-acb3-9a422722bb78",
      "name": "Cluster 1",
      "href": "/infrastructure-groups/6b153341-95b6-4794-acb3-9a422722bb78"
    },
    {
      "id": "99128443-1936-478e-bfcd-fde9243773de",
      "name": "Cluster2",
      "href": "/infrastructure-groups/99128443-1936-478e-bfcd-fde9243773de"
    }
  ]
}
```

Systems

Description

This resource is used to return information about systems loaded through data collection in Densify. These systems do not have to belong to a control environment or a cloud environment, but must be of a supported entity type. Returned resource element values match those that are displayed in the Densify Analysis Console from the Data Collection > Discovered System Explorer (DSE) > System Summary or Attributes pages.

Resource

```
/systems
```

Supported Operations

Table: Systems Supported Operations

Operation	HTTP Method	Input	Output	Description
Get Collection	`GET /systems`	Query String Parameter Options: Collection Details Paging	Collection of Response elements	Used to return configuration and attribute information of systems (of supported type) that are being tracked under recent data collection activity. Example: Getting a Collection of

Operation	HTTP Method	Input	Output	Description
		Element Filters Attributes On/Off Mode Attributes On/Off Mode Attribute Filters - Multiple Values Support Sorting		Sorted GCP Systems, Displaying One Page Example: Getting a Collection of VMware Systems using Attribute Filters
Get Individual Systems	`GET /systems/<id>`	Path Parameter: id (system ID)	Response elements	Used to return configuration and attribute information for a system specified by `id`. The system returned is being tracked under recent data collection activity and must be one of the supported system type. Example: Getting an Individual Host System Example: Getting an Individual Guest System
Get Report PDF	`GET /systems/<id>/analysis-report`	Path Parameter: id (system ID)		Used to obtain the Impact Analysis and Recommendation Report (also known as the Application Owner report). See *Viewing the Impact Analysis and Recommendation Report* (Help Topic ID 380450) for details on the content of the report. Use `Accept : application/octet-stream` in the request header to download the PDF file. Example: Downloading an Impact Analysis and Recommendation Report
Modify	`PUT`	Path	Response	Used to modify the values of a

Operation	HTTP Method	Input	Output	Description
System Attributes	`/systems/<id>/attributes`	Parameter: id (system ID) Request Body Parameters: array of attributes name value	elements with attributes modified	system’s attributes. If attribute `name` corresponds to an existing set attribute in the system, then the attribute's value will be updated to `value`. If attribute `name` is not set in the system, then the `[name,value]` tuple will be added to the system's attribute array. For multi-value attributes (i.e. "Multiple Values" property is enabled for the attribute), a PUT request for that attribute will append the new `[name,value]` pair to the attribute array, if that attribute `[name,value]` pair does not exist. To overwrite a multi-value attribute value, you will first need to delete the existing attribute `value`. Example: Modifying a System’s Attributes
Delete System Attributes	`DELETE /systems/<id>/attributes`	Path Parameter: id (system ID) Request Body Parameters: array of attributes id value or id	Response elements with attributes deleted	Used to delete attributes from an individual system given the attribute ID-value `[id,value]` pair or attribute `[id]`. If the attribute `[id,value]` pair is provided, the attribute will be deleted only if the attribute `[id]` and corresponding `[value]` matched the one found in the system. For multi-value attributes (i.e. "Multiple Values" property is enabled for the attribute), a DELETE request with only attribute `[id]` provided will delete all the values with that attribute ID; a DELETE request with `[id,value]` pair provided will delete only the attribute

Operation	HTTP Method	Input	Output	Description
				ID entry with that specific value. Example: Deleting a System's Attributes

Parameters

Path Parameters

Table: Systems Path Parameters

Parameter Name	Type	Description
id	string	The Densify unique referenced ID of the system.

Request Body Parameters

Table: Systems Request Body Parameters (Array of `[id, name, value]` attributes)

Parameter Name	Type	Description
id	string	The unique ID for the attribute in Densify. System attributes are properties set by a data collection from a vendor platform or by Densify for control analytics. The system's `attributes` array contains attribute elements, stored as an `[id, name, value]` tuple in the system's `attributes` array.
name	string	The `name` of the attribute corresponding to the `id` and `value` attribute tuple.
value	string	The `value` of the attribute corresponding to the `id` and `name` attribute tuple.

Query String Parameters

The following query string parameter options are available to the `/systems` resource:

- Sorting
- Attributes On/Off Mode
- Attribute Display Categories

- Element Filters
- Attribute Filters - Multiple Values Support

Sorting

You can order the returned collection by name or by memory size using the `sort_by` option. See Sort By for an overview of the Sort By option.

Supported Sort By elements for the /systems resource are described below:

- **size**

  The `size` element is a representation of the following properties to sort by, in priority order:

  - `memory`
  - `total_physical_cpus`
  - `cores_per_cpu`

  This implies that if two systems have the same total `memory`, the `total_physical_cpus` count is considered next in the sort logic; and if the physical CPU count is the same, the next property to consider for sorting is the `cores_per_cpu` count. By default, the returned `sort_by=size` collection is ordered in ascending (`asc`) order. Specify `desc` for descending order. For example:

  ```
  GET /systems/?sort_by=size,desc
  ```

  Systems with any "_Unknown_" values are sorted at the end of the collection independent of the specified sort order.

- **name**

  The returned collection is sorted by the system's `name` element in ascending alphabetical order. Specify `desc` for descending alphabetical order.

  If you specify the Paging option, the returned collection will be sorted by `name` in ascending order automatically. For example, the following two requests will return the same result (i.e. page two of the collection in ascending name order):

  ```
  GET /systems/?page=1
  ```

  ```
  GET /systems/?sort_by=name&page=1
  ```

Attributes On/Off Mode

By default, all the attributes with set values for each system in the collection are returned (which is the `attributes_mode=On` option). You can suppress the attributes of the returned collection by using the `attributes_mode=Off` option. For instance, if you want to return a collection of AWS systems without displaying their attributes, specify the following request:

```
GET /systems/?platform=aws&attributes_mode=Off
```

When you filter by attribute in your request, the `attributes_mode=Off` is disabled and all the set attributes are returned regardless if you provide the `attribute_mode=Off` option. For instance, if you provide the following request, the set attributes for the returned collection are displayed:

```
GET /systems/?attribute_name_value=Department,Finance&attributes_mode=Off
```

Attribute Display Categories

By default, all the attributes with set values for each system in the collection are returned. You can choose to return only specific attribute categories of the requested collection by using the `setdisplay_category` option. For instance, if you want to return a collection of AWS systems, but only display Business and Transformation attributes with set values, specify the following request:

```
GET /systems/?platform=aws&setdisplay_category=Business,Transformation
```

Element Filters

The /systems endpoint supports the following elements as filters for returning a subset of the collection:

- name
- name_like
- resource_id
- attributes **
- type
- platform *
- infrastructure_group *
- control_environment *
- platform_category *

These filters are also denoted by "F" in the Filter column of the Systems Response schema table. Refer to Filters for a complete description of the Filters feature.

Element filters denoted with * support multiple values input. You can provide a list of values, separated by a comma between values, to logically OR a list of possible values for an element. For example, the following request will return all systems from AWS, Azure, or GCP:

```
GET /systems/?platform=aws,azure,gcp
```

Refer to Filters for a description of the Multiple Values filtering.

System attributes filtering extends the standard multiple values filtering support (denoted by **) with multiple attribute name, ID, and value combinations. See Attribute Filters - Multiple Values Support for further details.

Note*: When filtering with values which include spaces or "+", use URL encoding . See Filters for details on special character support.*

Attribute Filters - Multiple Values Support

Systems attributes supports the following filtering features:

- A single value for `attribute_name_like` element filter—A collection of systems is returned with attribute names containing the sub-string provided. You can use the '%' character to match zero or more characters. For example, the request below returns systems with attribute values set for attribute names containing the sub-string "east":

  ```
  GET /systems/?attribute_name_like=east
  ```

- Multiple attribute IDs for element filter `attribute_id`—The list of provided attribute IDs are matched to the system's set attributes and results are logically OR'ed to produce the returned collection. For example, the request below returns systems with set values for attribute IDs *attr_5*, *attr_6*, or *attr_7*:

  ```
  GET /systems/?attribute_id=attr_5,attr_6,attr_7
  ```

- Multiple attribute names for element filter `attribute_name`—The list of provided attribute names are matched to the system's set attributes and results are logically OR'ed to produce the returned collection. For example, the request below returns systems with set values for attribute names *Department* or *Resource Group*:

  ```
  GET /systems/?attribute_name=Department,Resource+Group
  ```

- Multiple attribute ID-value criteria for the element filter `attribute_id_value`—Existing systems which match all of the provided attribute ID-value pairs (separated by semicolon) are returned. The format of this attribute ID-value criteria filter request is as follows:

  ```
  GET /systems/?attribute_id_value=<attr1_id,attr1_value;attr2_id,attr2_
  value;attr3_id,attr3_value,...>
  ```

  The results of matching attribute ID-value pairs are logically AND'ed to produce the returned collection. For example, the systems returned for the request below have

  - attr_Application = "General" AND
  - CLD.BUSINESS.UNIT = "Support" AND
  - attr_OperationalEnvironment = "Production"

  ```
  GET /systems/?attribute_id_value=attr_
  Application,General;CLD.BUSINESS.UNIT,Support;attr_
  OperationalEnvironment,Production
  ```

- Multiple attribute name-value criteria for the element filter `attribute_name_value`—Existing systems which match all of the provided attribute name-value pairs (separated by semicolon) are returned. The format of this attribute name-value criteria filter request is below:

  ```
  GET /systems/?attribute_name_value=<attr1_name,attr1_value;attr2_name,attr2_
  value;attr3_name,attr3_value,...>
  ```

  The results of matching attribute name-value pairs are logically AND'ed to produce the returned collection. For example, the systems returned for the request below have "debian-8-jessie" as their License model *AND* "Disk-AB21" as their disk name:

```
GET /systems/?attribute_name_value=License+model,debian-8-
jessie;Disk+Name,Disk-AB21
```

***Note**: When you use attributes for filtering a collection, the attributes_mode=Off feature is disabled. See Attributes On/Off Mode for details.*

Response

The following is a complete list of possible response elements that are returned for the `/systems` resource. If the response element does not apply to the system returned, then the element is not displayed in the results.

Table: Systems Response Schema

Element	Type	Mod	Filter/Sort	Description
id, name, href	strings		F by ▪ `id` ▪ `name` ▪ `name_like` S by `name`	See ID, Name and Self Reference (id, name, href). To filter systems with names that contain a given input string, use `?name_like="<`*`substring`*`>"` in your collection request. The '%' character can be used to match zero or more characters. Example: Getting a Collection of Azure Systems with Name "*test*"
resource_id	strings		F	The unique system ID assigned by the cloud provider. To search for a system with a particular cloud provider ID, use `?resource_id=<`*`cloud_id`*`>` in your request. Example: Getting Public Cloud Attributes for an Individual System
attributes	[id, name, value]	M	F by ▪ `attribute_name_like` ▪ `attribute_id` ▪ `attribute_name` ▪ `attribute_id_value` ▪ `attribute_name_value`	System attributes are properties set by an audit from a vendor platform or by Densify for control analytics. On a `GET` request, only those attributes that have values are returned. To hide the set attributes, use the `attributes_mode=Off` option. See Attributes On/Off Mode for details. To only return set attributes in specific categories, use the `setdisplay_category` option. See Attribute Display Categories for details. To filter based on attributes, you must use one of the following element options — in each case, system data will be returned only if the attributes

Element	Type	Mod	Filter/Sort	Description
				selected have values assigned to them: - attribute_name_like - attribute_id with a provided list of attribute IDs - attribute_name with a provided list of attribute names - attribute_id_value with a provided list of attribute ID-value pairs - attribute_name_value with a provided list of attribute name-value pairs Refer to Attribute Filters - Multiple Values Support for a description of how to use these attribute filters. Note that when you use attributes to filter the collection, the attributes_mode=Off feature is disabled. Example: Getting a Collection of VMware Systems using Attribute Filters Example: Getting Public Cloud Attributes for an Individual System
children	string			The number of other systems that are considered "children" of the current system. For a host system (i.e. "type": "host"), this is the number of VMs the host system has (e.g. "children": "10"). For AWS ASGs (i.e. "type" : "asg"), this is the number of active EC2 instances that belong to the group. For other type of systems where this element does not apply, children is not returned. This element is returned for host and ASG systems only when the details=true option is in the request URI. See Collection Details.
control_ environment	[id, name, platform_ category, href, icon]		F by - control_ environment - platform_ category	If the system is associated with a Control Environment, the environment details are returned: - id - name - platform_category - href - icon To filter based on Control Environment, you must use one of the following elements:

Element	Type	Mod	Filter/Sort	Description
				`control_environment`—with a control environment name specified `platform_category`—with a platform category specified (i.e. "External Cloud" or "Internal Virtual") Example: Getting a Filtered Collection of Systems.
cores_per_cpu	string		Tertiary sort key for `sort_by=size`	Cores per CPU for computing systems.
cpu_benchmarks	[name, score_type, value]			The default CPU benchmark for applicable systems. `name`—Label with possible values: • `"CINT2000"` • `"CINT2000 Rate"` • `"CINT2006 Rate"` • `"RPE2"` `score_type`—Score type with possible values: • `"cint2000"` • `"cint2000rate"` • `"cint2006rate"` • `"rpe2"` `value`
cpu_model	string			The CPU architecture model of the compute system.
cpu_speed	string			The normalized CPU speed (MHz) of the compute system.
entity_role_name	string			The entity role name that Densify assigns to the system (e.g. "VMWARE_VM") based on their platform and system role.
entity_type	string			The entity type that Densify assigns to the system (e.g. "VMware ESX Guest") based on their platform and system type.
hostId	string			An ID of the system provided from the data collection audit for the purpose of resolving duplication conflicts.
I/O_benchmarks	[name, score_type, value]			The list of all available I/O benchmarks. `name`—Label with possible values: • `"Maximum Disk Throughput (bytes/s)"` • `"Maximum Network Throughput (bytes/s)"` `score_type`—Score type with possible values: • `"disk"`

Element	Type	Mod	Filter/Sort	Description
				• `"net"` `value`—A value of -1 means there is no value specified.

infrastructure_ group	[id, name, href]		F by `infrastructure_ group`	If the system is associated with an Infrastructure Group (cluster), the group details are returned:
				`id`
				`name`
				`href`
				When filtering based on Infrastructure Group, you must use the element `infrastructure_group` with the group name specified.
				Example: Getting a Filtered Collection of Systems.

ip_address	string			Primary IP address of the system.
mac_address	string			MAC address of the system provided by the data collection audit.
manufacturer	string			For host type systems, the manufacturer of the host is returned (e.g. "Dell", "IBM"). For cloud systems, the vendor platform is returned (e.g. "GCP") or "CONTAINERS" is returned for container type systems. This element is returned only when the `details=true` option is in the request URI. See Collection Details.
memory	string		S by `size`	The normalized total memory (MB) for the system.
os	string			Operating System name of the host or VM instance. This element is returned only when the `details=true` option is in the request URI. See Collection Details
os_patch_level	string			Operating System patch level of the host or VM instance.
os_version	string			Operating System version of the host or VM instance.
parent	string			Logical parent of the current system. For a host system (i.e. `"type": "HOST"`), this does not apply (i.e. `"parent": "N/A"`). For a VM, this is the name of the parent host system (e.g. `"parent": "esx-host-221"`).
platform	string		F	The platform of the system. Use this element to filter systems from the various supported platforms:

Element	Type	Mod	Filter/Sort	Description
				• VMWARE—For systems in VMware • HMC—For systems in IBM Power • AWS—For systems in AWS • GCP—For systems in GCP • AZURE—For systems in Azure • CONTAINERS—For containers, both Kubernetes and ECS For example, use platform=azure to return a collection of Azure cloud systems. See Example: Getting a Collection of Azure Systems with Name "*test*".
platform_model	string			For host systems, the host system model is returned. For cloud instance systems, the instance type is returned.
serial_number	string			The serial number of the system from the vendor.
size	hidden element		S by size	This element is used to sort a collection by size. See Sorting for details.
total_logical_cpus	string			The total number of logical CPUs for the system.
total_physical_cpus	string		Secondary sort key for sort_by=size	The total number of physical CPUs for the system.
threads_per_core	string			The threads per core for the system.
type	string		F	The type of system. The supported types of systems for the /systems resource include: • host—For host systems in VMware, IBM PowerVM (via data collection for HMC 8 method), and AWS platforms. • guest—For guest systems in a VMware platform. • vm—For instances in VMware, IBM PowerVM (via data collection for HMC 8 method), GCP (Compute Engine), and AWS (EC2, ECS) platforms. • arm_vm—For instances in the Azure platform. • classic_vm—For classic instances in the Azure platform. • rds—For RDS instances in the AWS platform. • asg—For Auto Scaling groups in the AWS platform. • ecs_svc—For ECS services in the AWS platform. • Containers realized through data collection are

Element	Type	Mod	Filter/Sort	Description
				supported by the `/systems` endpoint, but does not have the type element populated. You can filter Container systems via the `platform=containers` element filter.

***Note**: IBM Power systems returned by the /systems endpoint are the ones created in Densify from the HMC 8 data collection method. Refer to Data Collection for IBM PowerVM Systems for HMC 8 (Help Topic ID 220170) for details.*

Examples

Example: Getting a Collection of Azure Systems with Name "*test*"

The following example shows you how to retrieve a collection of systems from the Azure cloud platform with a name containing "test".

Example: Getting a Collection of Azure Systems with Name "*test*"

Request:

```
GET /systems/?platform=azure&name_like=test
```

Response:

```
[
  {
    "id": "5da2be9c-3915-46df-a127-387e837a0697",
    "name": "esx-test-274",
    "href": "/systems/5da2be9c-3915-46df-a127-387e837a0697",
    "resource_id": "esx-test-274",
    "type": "classic_vm",
    "platform_model": "standard_a0",
    "platform": "AZURE",
    "total_physical_cpus": "1",
    "cores_per_cpu": "1",
    "memory": "768",
    "infrastructure_group": {
      "id": "c55e12ae-d568-427b-997d-1d5c54065a2d",
      "name": "eastus-test-vm",
      "href": "/infrastructure-groups/c55e12ae-d568-427b-997d-1d5c54065a2d"
    },
    "control_environment": {
      "id": "da70ab94-cea1-4a8a-83d7-c26a675ce650",
      "name": "cc377154-9605-4cb0-8b41-1b39e1c4ac0f",
      "platform_category": "External Cloud",
      "href": "/control-environments/da70ab94-cea1-4a8a-83d7-c26a675ce650",
      "icon": "/control-environments/da70ab94-cea1-4a8a-83d7-c26a675ce650/icon"
    },
  "attributes": [
   {
     "id": "attr_azure_resource_group",
     "name": "Resource Group",
     "value": "test-vm"
   },
   {
     "id": "attr_azure_tenant_id",
     "name": "Tenant ID",
     "value": "6c9190a7-bca6-4fcd-b35e-36378aadc695"
   },
   // ... *SNIP* ... additional attributes not displayed ... *SNIP* ...
  ]
 },
 {
  "id": "0e580bbb-9c21-49ff-b3df-729e80e26558",
  "name": "TestVMTestSub",
  "href": "/systems/0e580bbb-9c21-49ff-b3df-729e80e26558",
  "resource_id": "TestVMTestSub",
  "type": "arm_vm",
  "platform_model": "basic_a0",
  "platform": "AZURE",
  "total_physical_cpus": "1",
  "cores_per_cpu": "1",
  "memory": "768",
  "infrastructure_group": {
   "id": "30030e26-f5f3-431c-9d21-79e499ec763b",
   "name": "eastus+testresourcegroup",
   "href": "/infrastructure-groups/30030e26-f5f3-431c-9d21-79e499ec763b"
  },
  "control_environment": {
   // ... *SNIP* ... control_environment elements ... *SNIP* ...
  },
  "attributes": [
   {
     "id": "attr_2",
     "name": "Department",
     "value": "IT"
   },
   {
     "id": "attr_azure_resource_group",
     "name": "Resource Group",
     "value": "testresourcegroup"
   },
   // ... *SNIP* ... additional attributes not displayed ... *SNIP* ...
  ]
 },
   // ... *SNIP* ... additional Azure systems not displayed ... *SNIP* ...
]
```

Example: Getting a Collection of Sorted GCP Systems, Displaying One Page

This example shows you how to retrieve a collection of GCP systems, sorted by memory size in ascending order and only displaying the second page (note that `page=0` is the first page).

Example: Getting a Collection of Sorted GCP Systems and Displaying One Page

Request:

```
GET /systems/?platform=gcp&sort_by=size&page=1&page_size=1
```

Example: Getting a Filtered Collection of Systems

This example shows you how to retrieve a collection of systems filtered by platform and cluster.

Example: Getting a Filtered Collection of Systems

Request:

```
GET /systems/?platform=aws&infrastucture_group=us-east-1b&platform_
category=External+Cloud
```

Response:

```
[
  {
    "id": "007c3e76-9d76-45ec-b8b6-16fcc493121e",
    "name": "007c3e76-9d76-45ec-b8b6-16fcc493121e",
    "href": "/systems/007c3e76-9d76-45ec-b8b6-16fcc493121e",
    "resource_id": "i-036ddcaa03393444b",
    "type": "vm",
    "platform_model": "c4.2xlarge",
    "platform": "AWS",
    "total_physical_cpus": "8",
    "cores_per_cpu": "1",
    "memory": "16512",
    "infrastructure_group": {
      "id": "7393175f-cb71-4a1c-9f17-55a94a98f752",
      "name": "us-east-1b",
      "href": "/infrastructure-groups/7393175f-cb71-4a1c-9f17-55a94a98f752"
    },
    "control_environment": {
      "id": "d96431b4-99f3-43c0-98d6-609a4b80f0dd",
      "name": "us-east-manual",
      "platform_category": "External Cloud",
      "href": "/control-environments/d96431b4-99f3-43c0-98d6-609a4b80f0dd",
      "icon": "/control-environments/d96431b4-99f3-43c0-98d6-609a4b80f0dd/icon"
    },
    "attributes": [
      {
         "id": "attr_BasePerformanceCINT2006Rate",
         "name": "Base Performance CINT2006Rate",
         "value": "0.021513293"
      },
      // ... *SNIP* ... additional attributes not displayed ... *SNIP* ...
    ]
  },
      // ... *SNIP* ... additional filtered AWS systems not displayed ...
*SNIP* ...
]
```

Example: Getting a Collection of VMware Systems using Attribute Filters

This example shows you how to retrieve a collection of systems with platform "VMware" using attribute name-value pairs as filters.

Example: Getting a Collection of VMware Systems with Attribute Filters

Request:

```
GET /systems/?platform=vmware&attribute_name_value=Virtual+Domain,L2-
Production-Supp;Virtual+Cluster,us-central-3b
```

Example: Getting an Individual Host System

This example shows you how to retrieve an individual host system by ID.

Example: Getting an Individual Host System

Request:

```
GET /systems/0a098816-7120-4c67-a897-c227f8c2d750
```

Response:

```
{
  "id": "0a098816-7120-4c67-a897-c227f8c2d750",
  "name": "esx-host-274",
  "href": /existing-systems/0a098816-7120-4c67-a897-c227f8c2d750,
  "resource_id": "ba-5fdd-228",
  "type": "HOST",
  "os": "VMware",
  "os_version": "ESX Server 4.0.0"",
  "os_patch_level": "N/A",
  "manufacturer": "HP",
  "platform_model": "ProLiant DL585 G6",
  "platform": "VMWARE,
  "serial_number": "N/A",
  "hostId": "N/A",
  "entity_type": "VMWare ESX Host",
  "parent": "N/A",
  "children": "13",
  "entity_role_name": "VMWARE_HOST",
  "cpu_model": "AMD Opteron 8435",
  "total_logical_cpus": "12",
  "total_physical_cpus": "2",
  "cores_per_cpu": "6",
  "threads_per_core": "1",
  "cpu_speed": "2600",
  "memory": "98304",
  "ip_address": "192.163.117.108",
  "mac_address": "BC:20:65:17:39:89",
  "cpu_benchmarks": [
    {
      "name": "CINT2006 Rate",
      "score_type": "cint2006rate",
      "value": 160
    }
  ],
  "I/O_benchmarks": [
    {
      "name": "Maximum Disk Throughput (bytes/s)",
      "score_type": "disk",
      "value": 250000000
    },
    {
      "name": "Maximum Network Throughput (bytes/s)",
      "score_type": "net",
      "value": 150000000
    },
  ],
  "infrastructure_group": {
    "id": "94edf69b-08ca-41e6-ba2c-3d6be186ca60",
    "name": "Prod_BIPS-01",
    "href": "/infrastructure-groups/94edf69b-08ca-41e6-ba2c-3d6be186ca60"
  },
  "control_environment": {
    "id": "26092815-9d17-4e6a-abbd-f5b05a853bff",
    "name": "Boston",
    "platform_category": "Internal Virtual",
    "href": "/control-environments/26092815-9d17-4e6a-abbd-f5b05a853bff",
    "icon": "/control-environments/26092815-9d17-4e6a-abbd-f5b05a853bff/icon"
  }
}
```

Example: Getting an Individual Guest System

This example shows you how to retrieve an individual guest system by name.

Example: Getting an Individual Guest System

Request:

```
GET /systems/?name=win-vm-2319&details=true
```

Response:

```
{
  "id": "002922ea-48dc-4c74-9bd1-a718d6afbe05",
  "name": "win-vm-2319",
  "href": /existing-systems/002922ea-48dc-4c74-9bd1-a718d6afbe05,
  "resource_id": "bc-a922-823",
  "type": "GUEST",
  "os": "Windows",
  "os_version": "Server 2012",
  "os_patch_level": "N/A",
  "manufacturer": "VMware",
  "platform_model": "N/A",
  "platform": "VMWARE,
  "serial_number": "4221413de2b2b78da3c678ad5d1a46c5",
  "hostId": "N/A",
  "entity_type": "VMWare ESX Guest",
  "parent": "esx-host-221",
  "children": "N/A",
  "entity_role_name": "VMWARE_VM",
  "cpu_model": "N/A",
  "total_logical_cpus": "1",
  "total_physical_cpus": "1",
  "cores_per_cpu": "1",
  "threads_per_core": "1",
  "cpu_speed": "2666",
  "memory": "98304",
  "ip_address": "192.163.116.37",
  "mac_address": "BC:20:D4:8A:0C:97",
  "cpu_benchmarks": [
    {
      "name": "CINT2006 Rate",
      "score_type": "cint2006rate",
      "value": 27.1933
    }
  ],
  "I/O_benchmarks": [
    {
      "name": "Maximum Disk Throughput (bytes/s)",
      "score_type": "disk",
      "value": -1
    },
    {
      "name": "Maximum Network Throughput (bytes/s)",
      "score_type": "net",
      "value": -1
    },
  ],
  "infrastructure_group": {
    "id": "8a0c1b4e-85bd-422b-ac83-dfdf360619b4",
    "name": "Production Apps1",
    "href": "/infrastructure-groups/8a0c1b4e-85bd-422b-ac83-dfdf360619b4"
  },
  "control_environment": {
    "id": "0a32351d-7a82-43c6-959a-abbc8700ad15",
    "name": "New York",
    "platform_category": "Internal Virtual",
    "href": "/control-environments/0a32351d-7a82-43c6-959a-abbc8700ad15",
    "icon": "/control-environments/0a32351d-7a82-43c6-959a-abbc8700ad15/icon"
  }
}
```

Example: Downloading an Impact Analysis and Recommendation Report

The following example shows you how to download a PDF version of the Impact Analysis and Recommendation Report for your public cloud (AWS, Azure, GCP) or private cloud (VMware, IBM PowerVM) instances. The report is available after the Densify analysis generates the right-sizing recommendations and rdb-populate has run to completion.

Note: *HTTPS needs to be enabled to download the Impact Analysis and Recommendation Report PDF.*

Example: Download an Impact Analysis and Recommendation Report

Request:

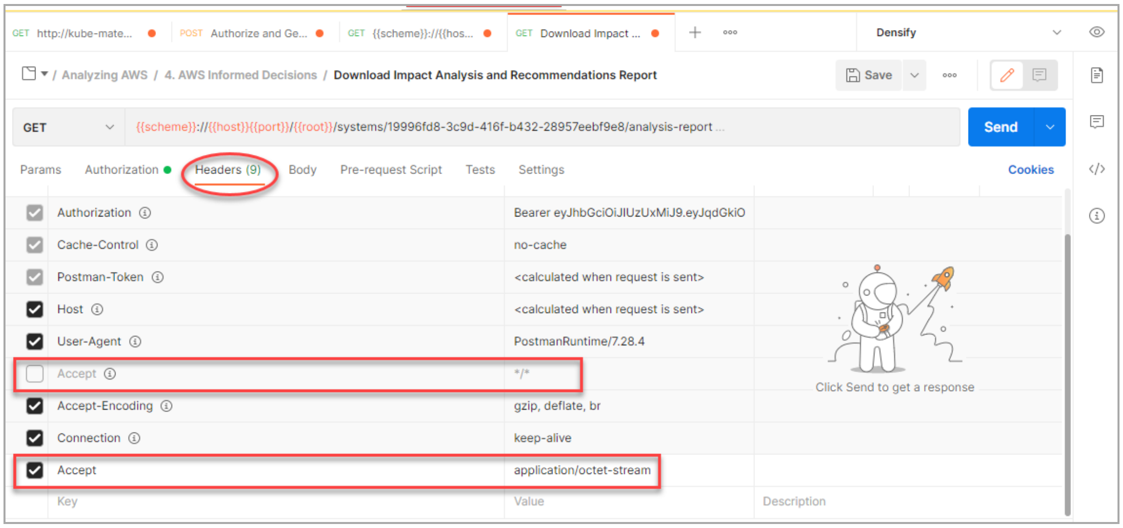
```
GET /systems/7836335a-1942-4115-a65d-a298be1d390c/analysis-report
```

Headers:

```
Accept: application/octet-stream
Authorization: Bearer <apiToken>
```

You need to save the file (Save Response) and then open it in a PDF reader.

Figure:Postman Header Configuration



GET {{scheme}}://{{host}}{{port}}/{{root}}/systems/19996fd8-3c9d-416f-b432-28957eebf9e8/analysis-report ...

	Key	Value	Description
☑	Authorization	Bearer eyJhbGciOiJIUzUxMiJ9.eyJqdGkiO	
☑	Cache-Control	no-cache	
☑	Postman-Token	<calculated when request is sent>	
☑	Host	<calculated when request is sent>	
☑	User-Agent	PostmanRuntime/7.28.4	
☐	Accept	*/*	
☑	Accept-Encoding	gzip, deflate, br	
☑	Connection	keep-alive	
☑	Accept	application/octet-stream	

You must disable the default **Accept** key value "*/" for **Headers**.

Example: Modifying a System’s Attributes

The following example shows you how to modify system attributes.

Example: Modifying a System’s Attributes

Request:

```
PUT /systems/0060ed45-6f83-4d74-9ff6-7a359e206428/attributes
[
  {"name": "Observed Uptime", "value": "0.33"},
  {"name": "Cost", "value": "17"}
]
```

Response:

```
[
  // ... *SNIP* of other elements ...
    "attributes": [
    // ... *SNIP* of other attributes ...
      {
        "id": "attr_ObservedUptime",
        "name": "Observed Uptime",
        "value": "0.33"
      },
      {
        "id": "attr_Cost",
        "name": "Cost",
        "value": "17"
      },
      // ... *SNIP* of other attributes ...
    ]
  // ... *SNIP* of other elements ...
```

Example: Deleting a System’s Attributes

The following example shows you how to delete a system's attr_3 attribute if it has value "0.3", and remove set values of attr_10 and attr_11.

Example: Deleting a System’s Attributes

Request:

```
DELETE /systems/0060ed45-6f83-4d74-9ff6-7a359e206428/attributes
[
  {"id": "attr_3", "value": "0.3"},
  {"id": "attr_10"},
  {"id": "attr_11"}
]
```

Example: Getting Public Cloud Attributes for an Individual System

The following example shows you how request information for a system based on the cloud provider's system ID (resource_id) and only show Public Cloud category attributes with set values.

Example: Getting Public Cloud Attributes for an Individual System

Request:

```
GET /systems/?resource_id=i-00bc69701199bd58&setdisplay_category=Public+Cloud
```

Subscriptions

Description

The `/subscriptions/<platformType>` resource is used to create and manage subscription-based notifications for Densify cloud and container recommendations.

To learn more watch the video, Subscription API Overview

This resource allows you to set up and schedule the delivery of personalized Densify recommendations to a third-party application for targeted distribution.

Personalization of the recommendation data set (returned subscription notification output) is facilitated by the following mechanisms:

- Return only the systems you are interested in by specifying system attribute conditions with the tagReferences parameter.
- Return only specific recommendations you are interested in by specifying recommendation property conditions with the propertyReferences parameter.
- Suppress certain systems or recommendations in the returned output, for a period of time, with the suppressionReferences parameter.
- Finally, the fields and attributes displayed in the returned output can be customized with the returnStructure parameter. You can specify which recommendation field to display and whether to display the field name or the field alias as the element key.

Subscription notifications are delivered to a third party application, such as Microsoft Teams™ or Slack™, using the webhook definition. If you do not specify a webhook in your Subscriptions definition, the notifications are not triggered because there is no delivery location. See *Subscriptions: Status* (Help Topic ID 340740) for details on Subscriptions results and webhook status.

You can dynamically retrieve the current personalized Subscriptions recommendations by providing a subscription ID in the GET subscription request. See *Subscriptions: Cloud Results* (Help Topic ID 340750) for details on retrieving Subscriptions results on-demand.

You can also schedule the frequency of the notification distribution using the schedule definition parameter. Subscriptions without a schedule definition will typically have notifications triggered nightly after recommendation analysis and reporting database updates.

Resource

```
/subscriptions/cloud
/subscriptions/containers
/subscriptions
```

***Note**: If you use this resource without the `<platformType>` specified (i.e. without `cloud` or `containers` specified), the behavior is exactly the same as specifying the `cloud`-specific resource. This behavior enables backward compatibility with scripts using the Densify API prior to release 12.1.6, where the platform-specific indicator was not available.*

Supported Operations

Table: Subscriptions Supported Operations

HTTP Method	Input	Output	Description
`GET /subscriptions/<platformType>`	Path Parameter: ▪ platformType Query String Parameter: ▪ type ▪ owner ▪ subscriptionRef	Collection of ▪ subscriptionRef ▪ subscriptionName ▪ description ▪ owner ▪ outputType ▪ active ▪ webhook ▪ propertyReferences ▪ tagReferences ▪ suppressionReferences ▪ schedule ▪ returnStructure ▪ webhookStatus ▪ lastTriggered	Returns a list of existing platform-specific subscriptions in Densify. ▪ The type query string parameter is used to return global or private subscriptions. If type is not specified, all global and only private subscriptions belonging to you are returned. ▪ If type is not specified and you

HTTP Method	Input	Output	Description
			are an **administrative user**[1], then all global and all private platform-specific subscriptions (belonging to all users) are returned. - Administrative users can use the owner query string parameter to return all the private subscriptions belonging to a specific user. - Use the subscriptionRef query string parameter to return details of a single subscription. Example: Getting a Collection of Subscriptions Example: Getting a Collection of Subscriptions for a Specific User
`POST /subscriptions/<platformType>`	Path Parameter: - platformType Collection of Request Body Parameters: - subscriptionName - owner - description - outputType - active	Collection of - subscriptionRef - subscriptionName	Creates and defines a collection of platform-specific subscriptions. - If you are not an **administrative users**[2], the owner parameter is automatically set to your Densify username.

[1] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

[2] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

HTTP Method	Input	Output	Description
	webhook propertyReferences tagReferences suppressionReferences returnStructure schedule		The Subscriptions creation (bulk-add) operation is committed as a whole unit. One Subscriptions creation error will roll back the entire operation. For any Subscriptions creation failure, an appropriate error message will be returned in the response body. See status for possible error status. Example: Creating a New Subscription
`PUT /subscriptions/ <platformType> /<subscriptionId>`	Path Parameters: platformType subscriptionRef Request Body Parameters: subscriptionName owner description active webhook propertyReferences tagReferences suppressionReferences returnStructure schedule	subscriptionRef subscriptionName	Deletes and replaces all the parameters in an existing platform-specific subscription definition. You must specify all parameters (from the Request Body Parameters section) required for the existing subscription. For any parameter replacement failure, an appropriate error message will be returned in the response body. See status for possible error status. Example: Modifying a Subscription ***Note***: *If you are not an administrative user, you can only modify your own private*

HTTP Method	Input	Output	Description
			subscription.
`DELETE /subscriptions/<platformType>`	Path Parameter: platformType Collection of Request Body Parameters: subscriptionRef	message status ***Note***: HTTP status "204 No Content" is returned for successful deletions. HTTP status "404 Not Found" is returned if the subscription does not exist or you have no privilege to access the subscription.	Deletes a collection of your private platform-specific subscriptions. If you are an administrative user, use this method to delete any private or global platform-specific Subscriptions collections. Each subscription delete operation is independent from the other subscription delete operations in the same request. An error with one subscription delete action does not affect the delete actions of the other subscriptions in the same request body parameter .
`DELETE /subscriptions/<platformType>/<subscriptionRef>`	Path Parameters: platformType subscriptionRef	message status ***Note***: HTTP status "204 No Content" is returned for successful deletions. HTTP status "404 Not Found" is returned if the subscription does not exist or you have no privilege to access the subscription.	Deletes one platform-specific subscription definition from Densify. If you are an administrative user, use this method to delete any private or global subscription. Otherwise, if you are a non-administrative user, use this method to delete one of your private subscriptions. Example: Deleting a Subscription

Parameters

Path Parameters

Table: Subscriptions Path Parameters

Parameter Name	Type	Required (Y/N)	Description
platformType	string	Y	[cloud|containers] Specify the technology platform for the subscription resource.
subscriptionRef	string	Y	Specify the unique subscription identifier.

Query String Parameters

Table: Subscriptions Query String Parameters

Parameter Name	Type	Required (Y/N)	Description
type	string		Specify the type of subscription to return: - all—(default) Return all subscriptions: global and private user-specific. If you are not an administrative user, only private subscriptions owned by you and global subscriptions are returned. This is the default behavior if "type" is not specified in the request. - global—Return all global subscriptions. - owner—Return user-specific subscriptions. If you are not an administrative user, only private subscriptions owned by you are returned. If you are an administrative user, all global and private subscriptions are returned. Typically, this option is used in conjunction with the owner query string parameter. A subscription is considered *global* if the owner parameter is not populated. Global subscriptions can be used by all Densify API users. A subscription is considered *private* if the owner parameter contains a Densify username. Private subscriptions can only be used by their owners or administrative users.

Parameter Name	Type	Required (Y/N)	Description
owner	string		If you are an **administrative user**[1], you can specify a Densify username in conjunction with the `type=owner` query string parameter to return all of the specified user's private subscriptions. If you are not an administrative user, you can only request for your own private subscriptions. If you use the `?type=owner&owner=<anotherusername>` query string option with a username other than your own, the returned response is a 400 Bad Request - "Current login user cannot query for owner" error.
subscriptionRef	string		Specify the identifier of the subscription details to return.

Request Body Parameters

Table: Subscriptions Request Body Parameters

Parameter Name	Type	Required (C-create/M-modify/ D-delete)	Description
subscriptionRef	string	D	Specify the unique subscription identifier to delete.
subscriptionName	string	C M	Specify a name for this subscription. For *global* subscriptions, the subscriptionName must be unique per platform. For *private* subscriptions, the subscriptionName must be unique per owner and across all global subscriptions for a particular platform. For example, owner A and owner B can both have a private cloud subscriptionName named "SubA", as long as "SubA" is not also a global cloud subscription.
owner	string	M[2]	When the `owner` parameter is not set, the subscription is considered *global*. Global subscriptions can be used by all Densify API users. Only **administrative users**[3] can create global subscriptions. When the `owner` parameter

[1] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

[2] The `owner` parameter is mandatory for modifying private subscriptions.

[3] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

Parameter Name	Type	Required (C-create/M-modify/ D-delete)	Description
			is set, the subscription is considered *private*. Private subscriptions can only be used by their owners or administrative users. If you are an administrative user, you have the ability to assign any Densify user as the owner of the subscription in a POST request. In a PUT request, administrative users can promote the subscription from private to global by setting `owner: ""`. If you are not an administrative user, you can only set the `owner` parameter to your username. In a POST request, the `owner` parameter is automatically set to your username.
description	string		Specify a description for subscription.
outputType	string		`application/json` is the only supported type.
active	string: `[true|false]`		Specify if the subscription is active: - true—The subscription is active and notifications are triggered at the scheduled times to the webhook defined. - false (default)—The subscription is dormant: no subscription notification is triggered, even if webhook and schedule parameters are defined. Setting the subscription to dormant allows you to temporarily disable the subscription notification without modifying the subscription configuration. Dormant subscriptions can also be used as an example or template for future subscription creation. Results from dormant subscriptions can be requested on-demand. See *Subscriptions: Cloud Results* (Help Topic ID 340750).
propertyReferences	array of property conditions: - propertyID - operator - values	C M[1]	Specify an array of property conditions to filter the customized data set. The following items need to be specified for each property condition: - propertyID—specify a global property reference or a private property reference owned by you (see

[1] At least one filter array (i.e. propertyReferences, tagReferences, or SuppressionReferences) must be defined for each subscription.

Parameter Name	Type	Required (C-create/M-modify/ D-delete)	Description
			- propertyID | tagID | suppressionID); - operator—specify the comparison operator (see operator); - values—specify the array of comparison values (see values). See Filter and Suppression Conditions for details on how to define property conditions. The platform of the properties defined in property conditions must match the subscription's platform. For instance, you must only use container properties for a container subscription. Refer to *Subscriptions: Properties > Default Properties* (Help Topic ID 340710) for the list of default properties to use in your array of property conditions. ***Note**: You must have at least one core property in your property condition set for the purpose of filtering subscription returned results. Refer to Core and Ancillary Properties in Filter Conditions for details.*
tagReferences	array of: - tagID - operator - values	C M[1]	Specify an array of attribute tag conditions to filter the customized data set. For example, if you want to return only recommendations for systems in the *ABC* account, in the *Marketing* business unit, and running the *Demo* application, then you would define `tagReferences` conditions where account="ABC", business_unit="Marketing", and app="Demo". The platform of the tags defined in tag conditions must match the subscription's platform. For instance, you must only use cloud tags for a cloud subscription. Each tag condition contains the following items: - tagID—specify a global or private tag reference owned by you (see propertyID | tagID | suppressionID); - operator—see operator; - values—see values.

[1] At least one filter array (i.e. propertyReferences, tagReferences, or SuppressionReferences) must be defined for each subscription.

Parameter Name	Type	Required (C-create/M-modify/ D-delete)	Description
			Refer to Filter and Suppression Conditions for details on how to define tag conditions.
suppressionReferences	array of suppression conditions: - suppressionID - operator - values - revokeBy (optional)	C M[1]	Specify an array of suppression conditions to determine systems to be excluded from the notification data set. For example, if you want to remove all "m3" systems from the results data set for this month, then you would create a suppression list with currentType="m3*" that will expire at the end of the month. The platform of the suppressions defined in suppression conditions must match the subscription's platform. For example, you must only use cloud suppressions for a cloud subscription. Each suppression condition contains the following items: - suppressionID—specify the global or private suppression reference owned by you (see propertyID | tagID | suppressionID); - operator—see operator; - values—see values; - revokeBy (optional)—the time when the suppression condition expires (in Unix-time format); if revokeBy is not specified, then the suppression condition does not expire. Refer to Filter and Suppression Conditions for details on how to define conditions.
webhook	- uri - authType - authValue		Specify the webhook definition to an external application, where your personalized recommendations will be sent. See *Analysis: Webhook: Request Body Parameters* (Help Topic ID 340490) for details of each parameter in the webhook definition. If you do not specify a webhook definition, then notifications for this subscription are not triggered.
schedule	- dayOfMonth - dayOfWeek		Specify the frequency of the subscription notifications. If no schedule is specified, then the subscription notification is triggered after daily Densify analyses are completed and reporting database tables are updated.

[1] At least one filter array (i.e. propertyReferences, tagReferences, or SuppressionReferences) must be defined for each subscription.

Parameter Name	Type	Required (C-create/M-modify/ D-delete)	Description
			Typically, these processes occur at night. The schedule parameter is defined by one or both of the following options: • dayOfMonth—trigger the notification on the specified day of the month (e.g. `"dayOfMonth" : [10,20]`); • dayOfWeek—trigger the notification on the specified day of the week (e.g. `"dayOfWeek" : [2,3],` where 7=Sunday, 1=Monday, 2=Tuesday, etc.). If both scheduling options are defined, the scheduling options are logically AND'ed to compute the notification schedule. For example, if dayOfMonth = [1, 10, 20] and dayofWeek = [2, 4], then notifications will trigger on: • the 1st, 10th, and 20th of the calendar month, and only if • the day is a Tuesday or a Thursday.
returnStructure	• properties: • propertyID • useAlias • tags: • tagID • useAlias		Specify property and attribute tag references to be included in the output of the subscription notification. The `properties` array contains the following items for each property: • propertyID—the property reference from the subscriptions property catalog; • useAlias—[true|false] indicates if the property alias is used as the element label in the subscription output. The `tags` array contains the following items per tag: • tagID—the tagreference from the subscriptions attribute tags catalog; • usesAlias—[true|false] specifies if the tag alias is used as the element label in the subscription output. If `returnStructure` is defined, then only the properties and tags in `returnStructure` are returned in the output of the subscription notification. The platform of the tags or properties defined in the `returnStructure` must correspond with the platform of the subscription. For example, you must only use cloud tag or properties for a cloud subscription. See `returnStructure` inside the POST request in

Parameter Name	Type	Required (C-create/M-modify/ D-delete)	Description
			Example: Creating a New Subscription. If `returnStructure` is not defined, then the full list of possible recommendation properties are returned. The full list of recommendation properties correspond to the technology-specific Analysis recommendation response schema. For example, see *Analysis: AWS Recommendations: Response* (Help Topic ID 340470) for a full list of AWS recommendation properties. For `auditInfo` and `dataQuality` property structures, you must specify the final properties to return by using the dot-walking syntax for each substructure. For example, if you want all properties of `dataCollection` substructure within `auditInfo` to be returned, you need to specify all the final properties in `auditInfo.dataCollection`: - `auditInfo.dataCollection.dateFirstAudited` - `auditInfo.dataCollection.dateLastAudited` - `auditInfo.dataCollection.auditCount` ***Note**: Resource attribute string values containing double quotes will be converted to single quotes in the subscription results output.*

Filter and Suppression Conditions

Each filter or suppression condition is defined as a set of three parameters:

```
{
    "propertyID | tagID | suppressionID": "<GUID>",
    "operator": "<supported logical operator>",
    "values": [<array of values>]
}
```

The propertyID | tagID | suppressionID are the unique identifiers to the property, tag or suppression in your Subscriptions catalogs. You can only reference identifiers that are global or privately owned by you. The system property or tag values are compared to the values array provided, based on the logical operator.

For example, if you want to filter cloud systems where "predictedUptime" is between 50-100%:

1. First, find the `propertyRef` for the "predictedUptime" property in your Cloud Subscriptions Property catalog:

```
{
    "propertyRef": "f2a38773-db60-478a-9982-1a2d1ba7d380",
    "propertyName": "predictedUptime",
    "aliasName": "propertyPredictedUptime",
    "owner": ""
}
```

2. Use the `propertyRef` identifier (in the previous step) for the `propertyReference` condition in your subscription:

```
"propertyReferences": [
    {
    "propertyID": "f2a38773-db60-478a-9982-1a2d1ba7d380",
    "operator": "[]",
    "values": [50,100]
    }
]
```

A subscription can have multiple filter conditions to customize the resulting data set. Each condition referencing an individual property, tag, or operator is logically *AND*'ed together during evaluation. Conditions referencing the same property or tag with the same logical operator are *OR*'ed together during evaluation. Systems matching the suppression conditions are removed from the resulting data set.

Note*: You cannot compare entire multi-structure recommendation elements, such as* `auditInfo` *and* `dataQuality`*. You can only use their final substructure properties in the subscription filter condition, such as* `auditInfo.dataCollection.dateFirstAudited` *or* `dataQuality.completedDays`*.*

Table: Logical Evaluation of Multiple Conditions

Logical Evaluation	Property/Tag/Suppression	Operator
AND	different property/tag/suppression	different or same operator
OR	same property/tag/suppression	same operator

Example: Multiple Filter Conditions

For example, if you have the following conditions:

- propertyReferences =
    - (propertyA > value1)
    - (propertyB = value2)

  - (propertyB >= value3)
- tagReferences =
  - (tagC = value4)
  - (tagD = value5)
  - (tagC = value6)
  - (tagC = value7)
- suppressionReferences =
  - (suppressionE—>propertyE = value8)
  - (suppressionF—>tagC = value9)

then the resulting subscription data set would be an intersect of the following groups:

- Find all the systems where (propertyA > value1) AND (propertyB = value2) AND (propertyB >= value3)

  AND
- Find all the systems where ((tagC = value4) OR (tagC = value6) OR (tagC = value7)) AND (tagD = value5)

  AND
- Remove all systems where (propertyE = value8) AND (tagC = value9)

***Note**: Suppression conditions are evaluated after property and tag conditions.*

Table: Subscriptions Filter and Suppression Condition Parameters

Parameter Name	Type	Required (Y/N)	Description
propertyID | tagID | suppressionID	string	Y	Specify the reference ID of the property, tag, or suppression from the corresponding catalog: - See *Subscriptions: Properties* (Help Topic ID 340710) to retrieve propertyIDs available in the platform-specific Subscriptions Properties catalog. - See *Subscriptions: Tags* (Help Topic ID 340720) to retrieve tagIDs available in the platform-specific Subscriptions Tags catalog. - See *Subscriptions: Suppressions* (Help Topic ID 340730) to retrieve suppressionIDs available in the platform-specific Subscriptions Suppressions catalog. You can only use global reference IDs or reference IDs that are owned by you.
operator	one of	Y	Specify a logical operator for the property or tag

Parameter Name	Type	Required (Y/N)	Description
	• = • < • > • <= • >= • like • [] • () • (] • [)		comparison. The following operators are supported: • "=" —equal to; • "<" —lesser than; • ">" —greater than; • "<=" —less than or equal to; • ">=" —greater than or equal to; • "like" —matches to the provided substring; the "*" wildcard character can be used to anchor substring searches; • "[]" —matches to values contained in the provided inclusive range, e.g. "values" : [1,3] matches to {1, 2, 3}; • "()" —matches to values in the provided exclusive range, e.g. "values": [3-8] matches to {4, 5, 6, 7}; • "(]" —matches to values in the provided exclusive beginning and inclusive ending range, e.g. "values": [2,5] matches to {3,4,5}; • "[)" —matches to values in the provided inclusive beginning and exclusive ending range, e.g. "values": [1, 3] matches to {1,2};
values	an array or range of • string • integer • float	Y	Specify an array of values for the logical comparison. For numeric ranges, specify the beginning and the end range values, e.g. "values": [1,10] implies from 1 to 10. For multiple values, the filter condition considers the list as a series of *OR* clauses. For example, if you are comparing serviceType = ["EC2", "RDS", "Spot"], then the condition is evaluated as [serviceType = "EC2" OR serviceType = "RDS" OR serviceType = "Spot"]. For the "like" operator, use the "*" wildcard character to match zero, one or multiple characters to one end of the substring, e.g. "values": ["t2*"] matches to "t2", "t2.5", or "t2.micro", but not to "rds-t2.large". Using the "like" operator without "*" wildcard will match the substring to any part of the property or tag value, e.g. "values": ["size"] matches to "Resize", "Upsize-Family", or "size-right". Refer to the various cloud recommendation resource elements for valid values: Analysis: AWS Recommendations: Response on page 108, Analysis: Azure Recommendations: Response on page 135, and Analysis: GCP Recommendations: Response on page 168.

Parameter Name	Type	Required (Y/N)	Description
			***Note**: To filter "Just Right" recommendations or "Not Analyzed" systems, you must use the exact recommendationType property string,* `"Not Analyzed"` *or* `"Just Right"`*, when you use the "=" or "like" operator.*

Core and Ancillary Properties in Filter Conditions

The concept of core and ancillary properties is taken into consideration only for filtering purposes. Ancillary properties cannot stand alone in a Subscriptions property condition set; they must be used in conjunction with a core property for the purpose of filtering subscription returned results.

In any Subscriptions property condition set, you must have at least one core property condition. If you want to only filter subscription data based on an ancillary property, you can add an *always-true* condition with a core property.

Refer to Default Cloud Properties on page 473 for the list of core and ancillary properties.

For example:

If you want to create a subscription to only review systems with the same recommendation for over 30 days, you would create a property filter condition with `recommSeenCount` > 30. Since `recommSeenCount` is an ancillary property, you would also need to add another *always-true* condition with a core property, such as `savingsEstimate` > -10000. This will couple an ancillary property with a core property in your condition set. Therefore, your resulting subscription property condition set for this example would be:

`propertyReferences` =

- (`recommSeenCount` > 30)
- (`savingsEstimate` > -10000)

After finding the `propertyID` for `recommSeenCount` and `savingsEstimate` in your Subscriptions Properties catalog, you can form the `propertyReferences` filter:

```
"propertyReferences": [
    {
    "propertyID": "1dd76248-a503-40b1-9303-e990bbad7818",
    "operator": ">",
    "values": [30]
    },
    {
    "propertyID": "cebcd841-89d8-4007-a4c6-1f0b06723db4",
    "operator": ">",
    "values": [-10000]
    }
]
```

Response

The following is a complete list of possible response elements returned for the `/subscriptions/<platformType>` resource.

Table: Subscriptions Response Schema

Element	Type	Filter/Sort	Description
Subscription Identification			
subscriptionRef	string	F	The unique referenced ID of the Densify subscription.
subscriptionName	string		The subscription name.
description	string		The subscription description.
owner	string	F	The designated owner of the subscription. The subscription is considered *global* if this element is empty and *private* otherwise.
Subscription Configuration			
outputType	string		`application/json` is the only supported output type.
active	string: `[true|false]`		Indicates if the subscription is active or dormant: true (default)—The subscription is active and notification will trigger at the scheduled time to the webhook defined. false—The subscription is dormant: no subscription notification will trigger, even if webhook and schedule parameters are defined.
webhook	uri authType authValue		The webhook definition to an external application, where your personalized recommendations will be sent. See *Analysis: Webhook* in the *Request Body Parameters* section (Help Topic ID 340490) for details of each parameter in the webhook definition. If a webhook definition is not specified, then no notifications are triggered for this subscription.
propertyReferences	array of propertyID operator values		An array of property conditions to be evaluated on the set of system recommendations before including them in the returned data set. For details of the `propertyID` listed in the

Element	Type	Filter/Sort	Description
			condition, see *Subscriptions: Properties* (Help Topic ID 340710) to retrieve property definitions available in the platform-specific Subscriptions Properties catalog. See Filter and Suppression Conditions for details on how property conditions are defined.
tagReferences	array of tagID operator values		An array of tag conditions to be evaluated on system attributes before including the system in the returned subscription data set. For details of the `tagID` listed in the condition, see *Subscriptions: Tags* (Help Topic ID 340720) to retrieve tag definitions available in the platform-specific Subscriptions Tags catalog. See Filter and Suppression Conditions for details on how tag conditions are defined.
suppressionReferences	array of suppressionID operator values revokeBy (optional)		An array of suppression conditions to determine system recommendations excluded from the returned subscription data set. Revoked suppression conditions (i.e. when `revokeBy` datetime has passed) are not taken into consideration during the results data set generation. For details of the `suppressionID` listed in the condition, see *Subscriptions: Suppressions* (Help Topic ID 340730) to retrieve suppression definitions available in the platform-specific Subscriptions Suppressions catalog. See Filter and Suppression Conditions for details on how suppression conditions are defined.
schedule	dayOfMonth dayOfWeek		The scheduled frequency of the subscription notification. If no schedule is specified, then the subscription notification is triggered after daily Densify analyses are completed and reporting database tables are updated. Typically, these processes occur at night. The schedule parameter is defined by one or both of the following options: dayOfMonth—trigger the notification on the

Element	Type	Filter/Sort	Description
			specified day of the month (e.g. `"dayOfMonth" : [10,20]`); - dayOfWeek—trigger the notification on the specified day of the week (e.g. `"dayOfWeek" : [2,3]`, where 7=Sunday, 1=Monday, 2=Tuesday, etc.). If both scheduling options are defined, the scheduling options are logical AND'ed to compute the notification schedule.
returnStructure	array of property references and an array of attribute tag references: - properties • propertyID • useAlias - tags • tagID • useAlias		Arrays of property and attribute tag references to be included in the output of the subscription notification. The following items are listed for each property in the properties array: - propertyID—the property reference from `/subscriptions/`*`<platformType>`*`/properties`; - useAlias—[true|false] indicates if the property alias is used as the title in the recommendation output. The following items are listed for each attribute tag in the tags array: - tagID—the tag reference from `/subscriptions/`*`<platformType>`*`/tags`; - useAlias—[true|false] indicates if the tag alias is used as the title in the recommendation output. If `returnStructure` is defined, then only the properties and tags in `returnStructure` are returned in the output of the subscription notification. If `returnStructure` is not defined, then the full list of possible recommendation properties are returned. The full list of recommendation properties correspond to the technology-specific Analysis recommendation response schema. For example, see *Analysis: AWS Recommendations: Response* (Help Topic ID 340470) for a full list of AWS recommendation properties. ***Note****: You must have at least one core property in your property condition set for the*

Element	Type	Filter/Sort	Description
			purpose of filtering subscription returned results. Refer to Core and Ancillary Properties in Filter Conditions for details.
webhookStatus	string		The status, date, and time of the last subscription results pushed to the webhook location: - Success—subscription results sent to the webhook successfully; - Failure—transmission of subscription results to the webhook failed.
lastTriggered	string		The status, date, and time of the last subscription results request: - On-Demand Success—the last request was on-demand and it was successful; - On-Demand Failure—the last request was on-demand and it failed to produce results; - Scheduled Success—the last request was a successful scheduled subscription event posted to a webhook; - Scheduled-Failure—the last request was a failed webhook post of the scheduled subscription event.
Returned Error Message			
message	string		The message for the status response is returned.
status	number		The HTTP response code of the request. Possible status values include: - 200—success with request (usually with content in response body); - 204—success with request, no content returned; - 400—bad request (invalid parameters, logical errors); - 401—authentication failed; - 404—resource not found (or no privileges); - 415—unsupported media type; - 500—internal server error.

Examples

Example: Getting a Collection of Subscriptions

The following example shows you how to retrieve your collection of cloud subscriptions. Assuming that your username is "saas", the returned collection is a set of global subscriptions and subscriptions that belong to the "saas" user. You are not able to see private subscriptions belonging to other Densify users.

***Note**: If you are an **administrative user**[1], then the returned collection will be all global cloud subscriptions and private cloud subscriptions for all Densify users.*

Example: Getting a Collection of Cloud Subscriptions

Request:

```
GET /subscriptions/cloud
```

Response:

```
[
  {
    "subscriptionRef": "2db7753a-26a3-4cba-be84-8e4e3a12daa1",
    "subscriptionName": "MyOwnSubscription",
    "description": "My first saas private subscription.",
    "owner": "saas",
    "outputType": "application/json",
    "active": true,
    "propertyReferences": [
       {
           "values": ["55"],
           "propertyID": "e3ad3bcd-e66d-40aa-8c55-af1f033fdb13",
           "operator": ">"
       },
       {
           "values": ["asg", "ec2"],
           "propertyID": "d17c1058-e346-4e12-a8c3-d4a440d34cfe",
           "operator": "="
       }
    ],
    "returnStructure": {},
```

[1] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

```
      "webhookStatus": "",
      "lastTriggered": "On-Demand Success. Wed Dec 18 16:26:52 EST 2019"
    },
    {
      "subscriptionRef": "4503e5ff-7a1e-4f4c-a106-0f31ca38dc22",
      "subscriptionName": "1GlobalSubscription",
      "description": "Shared global subscription - 1.",
      "owner": "",
      "outputType": "application/json",
      "active": true,
      "webhook": {
             "uri": "http://localhost:8080/sampletest",
             "authType": "basic",
             "authValue": "test:test1"
         },
      "propertyReferences": [
         {
             "values": ["asg"],
             "propertyID": "d17c1058-e346-4e12-a8c3-d4a440d34cfe",
             "operator": "="
         },
         {
             "values": ["100"],
             "propertyID": "e3ad3bcd-e66d-40aa-8c55-af1f033fdb13",
             "operator": ">"
         }
      ],
      "tagReferences": [
         {
             "values": [
                "229",
                "524"
                ],
             "tagID": "0eece6ab-9794-4b12-9868-ec2fa8e9d721",
             "operator": "like"
         }
      ],
      "suppressionReferences": [
         {
             "values": [
                "Terminate"
                ],
             "revokeBy": 1576645200000,
             "suppressionID": "abc57b21-d85f-4e80-8da4-377551cbf089",
             "operator": "="
         }
      ],
      "schedule": {
             "dayofWeek": [
                1,
                3,
                5
             ]
      },
      "returnStructure": {
         "properties": [
             {
                "useAlias": true,
                "propertyID": "96214999-5104-41b5-88e9-1681d1900e42"
             },
             {
                "useAlias": true,
                "propertyID": "ebcfa581-7c12-4e56-8eb5-3c2225941cbf"
             },
             {
                "propertyID": "e3ad3bcd-e66d-40aa-8c55-af1f033fdb13"
             },
             {
                "propertyID": "1a2c49b6-a7aa-4953-800b-f4fded2339a2"
             }
         ],
         "tags": [
             {
                "useAlias": true,
                "tagID": "5f2ff6aa-864b-4063-948e-388ad42e8f13"
             }
         ]
      },
      "webhookStatus": "Failure. java.net.ConnectException: Connection refused
  (Connection refused) Thu Jan 02 16:41:53 EST 2020",
      "lastTriggered": "On-Demand Success. Thu Jan 02 16:40:33 EST 2020"
    }
  ]
```

Example: Getting a Collection of Subscriptions for a Specific User

If you are an **administrative user**[1], you can use the `owner` query string parameter to get a collection of subscriptions for a particular user. The following example shows you how to retrieve a collection of cloud subscriptions for the "saas" user.

Example: Getting a Collection of Cloud Subscriptions for "saas" User

Request:

```
GET /subscriptions/cloud?type=owner&owner=saas
```

Response:

```
[
  {
```

[1] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

```
    "subscriptionRef": "2db7753a-26a3-4cba-be84-8e4e3a12daa1",
    "subscriptionName": "MyOwnSubscription",
    "description": "My first saas private subscription.",
    "owner": "saas",
    "outputType": "application/json",
    "active": true,
    "propertyReferences": [
       {
           "values": ["55"],
           "propertyID": "e3ad3bcd-e66d-40aa-8c55-af1f033fdb13",
           "operator": ">"
       },
       {
           "values": ["asg", "ec2"],
           "propertyID": "d17c1058-e346-4e12-a8c3-d4a440d34cfe",
           "operator": "="
       }
    ],
    "returnStructure": {},
    "webhookStatus": "",
    "lastTriggered": "On-Demand Success. Wed Dec 18 16:26:52 EST 2019"
  }
]
```

Example: Creating a New Subscription

This example shows you how to create and define a new cloud subscription. In this sample subscription, we filter for systems where predicted uptime is between 50-100% and suppress systems that have a "Terminate" recommendation.

Example: Creating a New Cloud Subscription

Request:

```
POST /subscriptions/cloud
```

Parameters:

```
[
    {
    "subscriptionName": "Sample Subscription",
    "description": "A subscription for testing",
    "active": true,
    "webhook": {
        "uri": "https://192.168.100.100:443/test/webhook",
        "authType": "basic",
        "authValue": "test:test"
    },
    "propertyReferences": [
        {
        "propertyID": "f2a38773-db60-478a-9982-1a2d1ba7d380",
        "operator": "[]",
        "values": [50,100]
        }
    ],
    "suppressionReferences": [
        {
        "suppressionID": "a6827ae4-fa2b-405e-a564-d70f2dad45c2",
        "operator": "=",
        "values": ["Terminate"]
        }
    ],
    "returnStructure": {
        "properties": [
            {
            "propertyID": "cebcd841-89d8-4007-a4c6-1f0b06723db4"
            },
            {
            "propertyID": "183b1f50-a20b-4d29-a488-8ba53bbf7c40"
            },
            {
            "useAlias": true,
            "propertyID": "b4e7260f-1aae-4150-b75d-1b1234075500"
            },
            {
            "useAlias": true,
            "propertyID": "e2ae92c5-91c7-4ff5-a29f-99aa92d65178"
            }
        ],
        "tags": [
            {
            "tagID": "8a556754-f7d2-4098-a3f5-c6777e2be697"
            }
        ]
    }
    }
]
```

Response:

```
[
    {
        "subscriptionRef": "731cd17f-c2c6-4aef-8bdb-84ac05bf8dff",
        "SubscriptionName": "Sample Subscription"
    }
]
```

Example: Modifying a Subscription

This example shows you modify an existing cloud subscription and assumes that you are the "saas" user. You have to specify all the required parameters (including the `owner` parameter) for the subscription since previous parameters are deleted or reset to default in a PUT request. In the example below, the subscription name, description and returned elements are updated from the original subscription (in the previous Example: Creating a New Subscription).

Example: Modifying a Cloud Subscription

Request:

```
PUT /subscriptions/cloud/731cd17f-c2c6-4aef-8bdb-84ac05bf8dff
```

Parameters:

```
{
    "subscriptionName": "Sample Subscription Modified",
    "description": "A modified subscription for testing",
    "owner": "saas",
    "active": true,
    "webhook": {
        "uri": "https://192.168.100.100:443/test/webhook",
        "authType": "basic",
        "authValue": "test:test"
        },
    "propertyReferences": [
        {
        "propertyID": "f2a38773-db60-478a-9982-1a2d1ba7d380",
        "operator": "[]",
        "values": [50,100]
        }
    ],
    "suppressionReferences": [
        {
        "suppressionID": "a6827ae4-fa2b-405e-a564-d70f2dad45c2",
        "operator": "=",
        "values": ["Terminate"]
        }
    ],
    "returnStructure": {
        "properties": [
            {
            "propertyID": "08d84679-1816-4cd7-b766-e4ad441b9a6c"
            },
            {
            "propertyID": "cebcd841-89d8-4007-a4c6-1f0b06723db4"
            },
            {
            "propertyID": "183b1f50-a20b-4d29-a488-8ba53bbf7c40"
            },
            {
            "useAlias": true,
            "propertyID": "b4e7260f-1aae-4150-b75d-1b1234075500"
            },
            {
            "useAlias": true,
            "propertyID": "e2ae92c5-91c7-4ff5-a29f-99aa92d65178"
            },
            {
            "propertyID": "e2dc1f96-7149-4767-894b-23e65485e314"
            },
            {
            "useAlias": true,
            "propertyID": "1dd76248-a503-40b1-9303-e990bbad7818"
            }
        ],
        "tags": [
            {
            "tagID": "8a556754-f7d2-4098-a3f5-c6777e2be697"
            }
        ]
    }
}
```

Response:

```
{
    "subscriptionRef": "731cd17f-c2c6-4aef-8bdb-84ac05bf8dff",
    "SubscriptionName": "Sample Subscription Modified"
}
```

Example: Deleting a Subscription

This example shows you how to delete a subscription. An HTTP "204 No Content" response is returned for a successful deletion.

Example: Deleting a Cloud Subscription

Request:

```
DELETE /subscriptions/cloud/9f58839a-dcad-4581-b883-051f729e3e1c
```

Subscriptions: Results

Description

The `/subscriptions/<platformType>/<subscriptionRef>` resource is used to dynamically retrieve your latest personalized Densify subscription recommendations. This resource allows you to review your platform-specific subscription results on-demand. The same information is delivered to a third-party application for targeted distribution if the third-party application is defined in the subscription's webhook.

To see how subscriptions are defined, refer to *Subscriptions* (Help Topic ID 340690).

Resource

```
/subscriptions/cloud/<subscriptionRef>
/subscriptions/containers/<subscriptionRef>
/subscriptions/<subscriptionRef>
```

Note*: If you use this resource without the `<platformType>` specified (i.e. without `cloud` or `containers` specified), the behavior is exactly the same as specifying the `cloud`-specific resource. This behavior enables backward compatibility with scripts using the Densify API prior to release 12.1.6, where the platform-specific indicator was not available.*

Supported Operations

Table: Subscription Results Supported Operations

HTTP Method	Input	Output	Description
`GET /subscriptions/<platformType>/<subscriptionRef>`	Path Parameter: platformType subscriptionRef Query String Parameter: divider limit	subscription: subscription • name • description • created • createdBy • updated • updatedBy • lastRefreshed • owner count results • collection of personalized recommendations in the format defined by *`returnStructure`*	Returns the current results of the platform-specific subscription notification, identified by *`<subscriptionRef>`*. The returned output is the same information that is delivered to the webhook application if the subscription is scheduled to run at the current time. Note that the following parameters are ignored for on-demand subscription results request: active—results are always returned for on-demand requests, even if `"active": "false"`; webhook—results are always returned for on-demand requests, even if `webhook` is undefined; schedule—results are always returned for on-demand requests, regardless of the notification `schedule` defined. An appropriate error message and HTTP code are returned for failed requests. See Returned Status See Example: Getting On-Demand Subscription Cloud Results

Parameters

Path Parameters

Table: Subscription Results Path Parameters

Parameter Name	Type	Required	Description
platformType	string	Y	[cloud|containers] Specify the technology platform for the subscription results resource.
subscriptionRef	string	Y	Specify the unique subscription identifier.

Query String Parameters

Table: Subscription Results Query String Parameters

Parameter Name	Type	Required (Y/N)	Description

divider	string
	`[true|false]`
			An option to display a divider that separates properties and tags for each system in the output:
			- true (default)—show the divider:
			`"divider": "----------------------`
			`---",`
			- false—do not show the divider.
			Note*: The divider is not displayed in the output for scheduled subscription results sent to webhooks.*

limit	number range [1:30000]		An option to set the limit of returned results. By default, the limit is set to 3000. This implies that the number of results returned is limited to 3000 systems. If a subscription's result count is more than 3000 system recommendations, an error is returned with code 400. If this is the case, you can adjust your on-demand subscription request to a higher limit. For example: `?limit=8000`. The range of the limit value is from 1 to 30000. ***Note****: This limit does not apply to subscription recommendations sent to webhooks. It only applies to on-demand subscription queries, and hence, will negatively impact web server*

Parameter Name	Type	Required (Y/N)	Description
			performance if the limit is set too high. See Example: On-Demand Subscription Results Count Exceeding Limit.

Response

The following is a complete list of possible response elements returned for the `/subscriptions/<platformType>/<subscriptionRef>` resource.

Table: Subscription Results Response Schema

Element	Type	Filter/Sort	Description
Subscription Information			
subscription	name description created createdBy updated updatedBy lastRefreshed owner		Subscription header information. See: name description created createdBy updated updatedBy lastRefreshed owner
name	string		The subscription name.
description	string		The subscription description.
created	datetime		The date and time when the subscription entity was created.
createdBy	string		The authenticated Densify user who created the subscription.
updated	datetime		The date and time when the subscription was last modified.
updatedBy	string		The authenticated Densify user who modified the subscription.
lastRefreshed	datetime		The date, and time of the last recommendation analysis. This is typically scheduled to run nightly after data collection.
owner	string		The designated owner of the subscription. This field contains the Densify username of the owner. The subscription is considered *global* if this element is empty and *private* otherwise.
Subscription Results			

Element	Type	Filter/Sort	Description
count	number		The number of system recommendations in the results collection.
results	array of returnStructure		An array of system results, with properties and tags defined in the subscription's `returnStructure`. Refer to *Subscriptions: returnStructure* (Help Topic ID 340690) for details on how the returnStructure can be defined. Below are the possible elements returned for each system recommendation: - properties—the list of properties defined in the returnStructure (if no returnStructure is defined, all recommendation properties are returned); - divider—the divider element between properties and tags (if defined): `"divider": "-----------------------"`, - tags—the list of attribute tags defined in the returnStructure. Refer to the various cloud or container recommendation resource topics for a description of each result element: Analysis: AWS Recommendations: Response on page 108, Analysis: Azure Recommendations: Response on page 135, Analysis: GCP Recommendations: Response on page 168, and Analysis: Kubernetes Container Recommendations: Response on page 186.
Returned Status			
message	string		The message for the status response is returned. See Example: On-Demand Subscription Results Count Exceeding Limit for an example of HTTP 400 Bad Request error message.
status	number		The HTTP response code of the request. Possible status values include: - 200—success with request (usually with content in response body); - 204—success with request, no content returned; - 400—bad request (invalid parameters, logical errors); - 401—authentication failed; - 404—resource not found (or no privileges); - 415—unsupported media type; - 500—internal server error.

Examples

Example: Getting On-Demand Subscription Cloud Results

The following example shows you how to retrieve your subscription cloud results, on-demand.

Example: Getting Cloud Subscription Results On-Demand

Request:

```
GET /subscriptions/cloud/fa3be33c-fbb7-4912-88c2-a83ee58e444e
```

Response:

```
{
    "subscription": {
        "name": "Sample Subscription",
        "description": "A subscription for testing",
        "created": "Mon Jan 19 13:52:31 EST 2020",
        "createdBy": "saas",
        "updated": "Mon Jan 20 14:32:38 EST 2020",
        "updatedBy": "SaaSadmin",
        "lastRefreshed": "Mon Jan 20 01:32:59 EST 2020",
        "owner": "saas"
        },
    "count": 452,
    "results": [
        {
          "currentType": "standard_d2",
          "name": "st01-prepro-edge-307",
          "recommendationType": "Modernize - Optimal Family",
          "savingsEstimate": "43.850475",
          "serviceType": "Virtual Machine",
          "divider": "-------------------------",
          "Availability Zone": "eastus+group"
        },
        {
          "currentType": "r3.2xlarge",
          "name": "ex-prepro-stoc-384",
          "recommendationType": "Downsize",
          "savingsEstimate": "75.36128",
          "serviceType": "EC2",
          "divider": "-------------------------",
          "Resource Tags":
            [
             "Owner : Bob Snow",
             "Product Code : PR000263",
             "Environment : Pre-prod",
             "Inventory Code : prepro-stoc",
             "Name : ex-prepro-stoc-384"
            ],
          "Availability Zone": "us-east-1d"
        },
        {
        // ... *SNIP* additional 450 system recommendations not displayed
*SNIP* ...
        }
    ]
}
```

Example: On-Demand Subscription Results Count Exceeding Limit

If your on-demand Subscriptions request contains a large number of results, exceeding the results limit, you will encounter an HTTP 400 Bad Request error message with the count limit.

Example: On-Demand Subscription Results Exceed Limit

Request:

```
GET /subscriptions/cloud/489331bd-5db9-408d-b7a0-2f3f8b1f66e9
```

Response:

```
{
    "message": "On-Demand Failure. The return count of 3891 has exceeded
object return limit of 3000. Update your call with a new limit value. Wed Jul
29 09:05:15 EDT 2020",
    "status": 400
}
```

This error is returned only for on-demand Subscriptions results requests with results count exceeding the default count limit. Subscription results sent to webhooks do not have a count limit. You can override this limit by requesting the on-demand results again with an increased limit count in the URL query string. Note that increasing the results limit may affect response time of the Densify web server.

Request with increased limit count:

```
GET /subscriptions/cloud/489331bd-5db9-408d-b7a0-2f3f8b1f66e9?limit=4000
```

Subscriptions: Properties

Description

The `/subscriptions/<platformType>/properties` resource is used to store a list of recommendation properties available to the Subscriptions resource for the purpose of filtering recommendations from the resulting subscription data set.

This resource acts as a catalog of recommendation properties to be referenced by the propertyReferences parameter in a subscription. Properties not defined in the `/subscriptions/<platformType>/properties` resource, cannot be referenced by a subscription property filter condition. See *Subscriptions* (Help Topic ID 340690) for details on defining a subscription and using property filter conditions.

There is a catalog for each supported `<platformType>`, which can only be referenced by the corresponding `<platformType>` subscription. For example, a container subscription (i.e. `/subscriptions/containers`) can only reference properties from the Container Subscriptions Properties catalog (i.e. `/subscriptions/containers/properties`).

Densify provides a default set of properties for each `<platformType>` catalog. You can extend these default sets by adding additional properties, or you can customize the sets by modify or deleting existing properties.

For the full set of available properties to add to the Cloud Subscriptions Properties catalog, refer to the cloud recommendation response schema:

- see *Analysis: AWS Recommendations Response* (Help Topic ID 340470) for a list of all the recommendation elements available for the AWS technology;
- see *Analysis: Azure Recommendations Response* (Help Topic ID 340510) for a list of all the recommendation elements available for the Azure technology;

- see *Analysis: GCP Recommendations Response* (Help Topic ID 340500) for a list of all the recommendation elements available for the GCP technology.

See Default Cloud Properties on page 473 for the list of default cloud properties.

***Note**: Some recommendation elements are not common to all technologies. It is good practice to indicate the technology for an element that is technology-specific in the aliasName (e.g. aliasName = "AWS minGroupRecommended"). This practice is helpful when you use the Cloud Subscriptions Properties catalog to form property conditions.*

For the full set of available properties to add to the Container Subscriptions Properties catalog, refer to the container recommendation response schema:

- see *Analysis: Kubernetes Container Recommendations: Response* (Help Topic ID 340570) for a list of all the recommendation elements available for the Kubernetes Container technology.

See Default Container Properties on page 478 for the list of default cloud properties.

Properties in the platform-specific Subscriptions Properties catalogs can be declared as *global* or *private* (i.e. user-specific). Global properties can be used by any API enabled user, while private properties can only be used by their owners. An exception to this rule is that **administrative users**[1] have access to all properties - global or private user-specific.

Resource

```
/subscriptions/cloud/properties
/subscriptions/containers/properties
/subscriptions/properties
```

***Note**: If you use this resource without the `<platformType>` specified (i.e. without `cloud` or `containers` specified), the behavior is exactly the same as specifying the `cloud`-specific resource. This behavior enables backward compatibility with scripts using the Densify API prior to release 12.1.6, where the platform-specific indicator was not available.*

Supported Operations

Table: Subscriptions Properties Supported Operations

HTTP Method	Input	Output	Description
`GET /subscriptions/<platformType>/properties`	Path Parameter: platformType	Collection of propertyRef	Returns a list of existing properties in the platform-specific Subscriptions

[1] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

HTTP Method	Input	Output	Description
	Query String Parameter Options: type owner	propertyName aliasName owner	Properties catalog. Use the type query string parameter to return only global or only private properties. If type is not specified, all global and only private properties belonging to you are returned. Administrative users[1] can use the owner query string parameter to return all the private properties belonging to a specific user. See Example: Getting a Collection of Private Cloud Subscriptions Properties on page 482.
`GET /subscriptions/<platformType>/properties/<propertyRef>`	Path Parameters: platformType propertyRef	propertyRef propertyName aliasName owner	Returns a Subscriptions property with unique identifier `<propertyRef>` from a platform-specific Subscriptions Properties catalog. See Example: Getting a Specific Container Subscriptions Property on page 483.
`POST /subscriptions/<platformType>/properties`	Path Parameter: platformType Request Body Parameters: Collection of propertyName aliasName owner	Collection of propertyRef propertyName	Adds new recommendation properties into a platform-specific Subscriptions Properties catalog. Administrative users[2] can add global or private properties. Non-administrative users can only add private properties. `propertyName` is

[1] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

[2] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

HTTP Method	Input	Output	Description
			validated for uniqueness within a catalog (i.e. you cannot define two properties with the same `propertyName`). `aliasName` is validated for uniqueness, depending on the private or global scope of the property (i.e. two users can use the same `aliasName` for their private property). The properties bulk-add operation is committed as a whole unit. One property add error will roll back the entire bulk add operation. Platform-specific properties must be added into the same platform-specific Subscriptions Properties catalog. For example, you can only add container properties into the Container Subscriptions Properties catalog. See Example: Adding New Subscriptions Properties on page 483.
`PUT /subscriptions/<platformType>/properties`	Path Parameter: platformType Request Body Parameters: Collection of propertyRef propertyName aliasName owner	Collection of propertyRef propertyName	Replaces parameters for existing properties in a platform-specific Subscriptions Properties catalog. You must specify all parameters required for the property you want to update. The `<propertyRef>` parameter is used to identify the property to update; hence it cannot be modified.

HTTP Method	Input	Output	Description
			Only an **administrative user**[1] can change the `owner` parameter to promote a private property to a global property (i.e. set `owner=""`). If you are not an administrative user, you can only set the `owner` parameter to your username. `propertyName` is validated for uniqueness within a catalog. `aliasName` is validated for uniqueness, depending on the private or global scope of the property (i.e. two users can use the same `aliasName` for heir private property). The bulk-edit operation is committed as a whole unit; an error resulting from one of the edits will roll back the entire bulk edit operation. See Example: Modifying Subscriptions Properties on page 484 ***Note**: If you are not an administrative user, you can only modify your own private properties.*
`PUT /subscriptions/<platformType> /properties/<propertyRef>`	Path Parameters: platformType propertyRef Request Body Parameters: propertyName aliasName	propertyRef propertyName	Replaces parameters in an existing property, identified by `<propertyRef>` in a platform-specific Subscriptions Properties catalog. You must specify all parameters required for the

[1] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

HTTP Method	Input	Output	Description
	• owner		existing property, as all previous parameters are deleted. • Only an administrative user[1] can change the `owner` parameter to promote a private property to a global property (i.e. set `owner=""`). • If you are not an administrative user, you can only set the `owner` parameter to your username. • `propertyName` is validated for uniqueness within a catalog. • `aliasName` is validated for uniqueness, depending on the private or global scope of the property (i.e. two users can use the same `aliasName` for their private property). See Example: Modifying a Subscriptions Property on page 485. ***Note**: If you are not an administrative user, you can only modify your own private property.*
`DELETE /subscriptions/<platformType>/properties`	Path Parameter: • platformType Request Body Parameter: Collection of • propertyRef	• HTTP status of "204 No Content" if all delete operations are successful • If delete errors	Remove Subscriptions properties from a platform-specific Subscriptions Properties catalog. • Administrative users[2] can delete any global or private properties.

[1] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

[2] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

HTTP Method	Input	Output	Description
		occur, then the following is returned for each property delete request: • propertyRef • message • status	• Non-administrative users can only delete their own private properties from the catalog. • Properties referenced in filtering conditions by subscriptions cannot be deleted. • Each property delete operation is independent from the other delete operations in the same request. An error with one property delete action does not affect the delete actions of the other properties in the same request parameter body. See Example: Deleting Subscriptions Properties.
`DELETE /subscriptions/<platformType> /properties/<propertyRef>`	Path Parameter: • propertyRef	• HTTP status of "204 No Content" if delete operation is successful • HTTP status of "404 Not Found" if property is not found • If the property is referenced by a subscription, then the following elements are returned: • propertyRef • message • status	Removes a Subscriptions property with `<propertyRef>` identifier from a platform-specific Subscriptions Properties catalog. • Administrative users[1] can delete any global or any private properties. • Non-administrative users can only delete their own private properties from the catalog. • Properties referenced by subscriptions cannot be deleted.

[1] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

Parameters

Path Parameters

Table: Subscriptions Properties Path Parameters

Parameter Name	Type	Required (Y/N)	Description
platformType	string	Y	[cloud|containers] Specify the technology platform for the Subscriptions property resource.
propertyRef	string	Y	Specify the unique identifier for a Subscriptions property.

Query String Parameters

Table: Subscriptions Properties Query String Parameters

Parameter Name	Type	Required (Y/N)	Description
type	string		Specify the type of Subscriptions properties to return: - all—Return all properties: global and private user-specific. If you are not an **administrative user**[1], only private properties owned by you and global properties are returned. This is the default behavior if "type" is not specified in the request. - global—Return all global Subscriptions properties. - owner—Return user-specific Subscriptions properties. If you are not an administrative user, only private properties owned by you are returned. If you are an administrative user, all global and private properties are returned. A Subscriptions property is considered *global* if the owner parameter is not populated. Global Subscriptions properties can be used by all Densify API users. A Subscriptions property is considered *private* if the owner parameter contains a Densify username. Private Subscriptions properties can only be used by their owners or administrative users.

[1] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

Parameter Name	Type	Required (Y/N)	Description
owner	string		If you are an **administrative user**[1], you can specify a Densify username in conjunction with the `type=owner` query string parameter to return all of the specified user's private Subscriptions properties. If you are not an administrative user, you can request for your own private properties. If you use the `?type=owner&owner=<anotherusername>` query string option with a username other than your own, the returned response is a 400 Bad Request - "Current login user cannot query for owner" error.

Request Body Parameters

Table: Subscriptions Properties Request Body Parameters

Parameter Name	Type	Required (C-create/M-modify/D-delete)	Description
propertyRef	string	M D	Specify the unique identifier for a Subscriptions property.
propertyName	string	C M	Specify the recommendation element for the Subscriptions property. The list of available recommendation elements can be found in the Response schema section of the *Analysis: technology-specific Recommendations* page. For example, refer to the Response schema section of the Analysis: AWS Recommendations on page 105 for a full list of AWS recommendation elements. The `propertyName` must be unique within a platform-specific Subscriptions Properties catalog.
aliasName	string		Specify an alias name for the Subscriptions property. For *global* properties, the aliasName must be unique per platform-specific catalog. For *private* properties, the aliasName must be unique per owner and across all global Subscriptions properties per platform-specific catalog. For example, owner A and owner B can both have a private property alias named "OptimizedSize", as long as "OptimizedSize" is also not a global

[1] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

Parameter Name	Type	Required (C-create/M-modify/D-delete)	Description
			property alias within the same catalog.
owner	string	M[1]	When the `owner` parameter is not set, the Subscriptions property is considered *global*. Global Subscriptions properties can be used by all API users. Only **administrative users**[2] can create global properties. When the `owner` parameter is set, the property is considered *private*. Private Subscriptions properties can only be used by their owners or administrative users. If you are an administrative user, you have the ability to assign any Densify user as the owner of the Subscriptions property in a POST request. In a PUT request, administrative users can promote the property from private to global by setting `owner: ""`. If you are not an administrative user, you can only set the `owner` parameter to your Densify username. In a POST request, the `owner` parameter is automatically populated with your username.

Response

The following is a complete list of possible response elements returned for the `/subscriptions/properties` resource.

Table: Subscriptions Properties Response Schema

Element	Type	Filter/Sort	Description
propertyRef	string		The unique referenced ID of the Densify Subscriptions property.
propertyName	string		The Subscriptions property name.
aliasName	string		The Subscriptions property alias name.
owner	string	F	The designated owner of this Subscriptions property. A Subscriptions property is considered *global* if this parameter is not populated. A Subscriptions property is considered *private* if the owner

[1] The `owner` parameter is mandatory for private Subscriptions properties.

[2] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

Element	Type	Filter/Sort	Description
			parameter contains a Densify username.
message	string		The message for the status response.
status	number		The HTTP response code of the request. Possible status values include: - 200—success with request (usually with content in response body); - 204—success with request, no content returned; - 400—bad request (invalid parameters, logical errors); - 401—authentication failed; - 404—resource not found (or no privileges); - 415—unsupported media type; - 500—internal server error.

Ancillary Properties

The concept of core and ancillary properties is taken into consideration only for filtering purposes. Ancillary properties cannot stand alone in a Subscriptions property condition set; they must be used in conjunction with a core property for the purpose of filtering subscription returned results.

In any Subscriptions property condition set, you must have at least one core property condition. If you want to only filter subscription data based on an ancillary property, you can add an *always-true* condition with a core property.

Refer to Default Cloud Properties on page 473 for the list of core and ancillary properties.

For example, if you want to see recommendations for systems that have been collected for over 10 days, create a property filter condition with `auditInfo.dataCollection.auditCount` > 10. Since `auditInfo.dataCollection.auditCount` is an ancillary property, you need to add another condition with a core property that would evaluate to true for the systems you are interested in, such as `currentCost` > 0 (the current cost of the instance is greater than 0). In this example, the Subscription's property condition set would be:

`propertyReferences` =

- (`auditInfo.dataCollection.auditCount` > 10)
- (`currentCost` > 0)

Refer to Filter and Suppression Conditions for further details on how to define a Subscription's property conditions.

Default Cloud Properties

Default Cloud Subscriptions Properties List

The default Subscriptions Properties catalog contains both core and ancillary properties which can be used by all API users (i.e. default properties are all global). Properties are listed in alphabetic order in the following table.

***Note**: Ancillary properties must be used in conjunction with at least one core property for the purpose of filtering Subscriptions results. Refer to Ancillary Properties for details.*

propertyName	aliasName	Core / Ancillary	Description Reference
accountIdRef	accountIdRef	core	Analysis: AWS Recommendations: accountIdRef on page 108
approvalType	approvalType	core	Analysis: AWS Recommendations: approvalType on page 112
avgInstanceCountCurrent	avgInstanceCountCurrent	core	Analysis: AWS Recommendations: avgInstanceCountCurrent on page 116
avgInstanceCountRecommended	avgInstanceCountRecommended	core	Analysis: AWS Recommendations: avgInstanceCountRecommended on page 116
auditInfo.dataCollection.auditCount	dc_auditCount	ancillary	Analysis: AWS Recommendations: auditInfo on page 118 ***Note**: Ancillary properties must be used in conjunction with at least one core property for the purpose of filtering subscription results. Refer to Ancillary Properties for details.*
auditInfo.dataCollection.dateFirstAudited	dc_firstAudited	ancillary	Analysis: AWS Recommendations: auditInfo on page 118
auditInfo.dataCollection.dateLastAudited	dc_lastAudited	ancillary	Analysis: AWS Recommendations: auditInfo

propertyName	aliasName	Core / Ancillary	Description Reference
			on page 118
auditInfo.workloadDataLast30.firstDate	last30_firstDate	ancillary	Analysis: AWS Recommendations: auditInfo on page 118
auditInfo.workloadDataLast30.lastDate	last30_lastDate	ancillary	Analysis: AWS Recommendations: auditInfo on page 118
auditInfo.workloadDataLast30.seenDays	last30_seenDays	ancillary	Analysis: AWS Recommendations: auditInfo on page 118
auditInfo.workloadDataLast30.totalDays	last30_totalDays	ancillary	Analysis: AWS Recommendations: auditInfo on page 118
currentCost	currentCost	core	Analysis: AWS Recommendations: currentCost on page 113
currentDesiredCapacity	currentDesiredCapacity	core	Analysis: AWS Recommendations: currentDesiredCapacity on page 116
currentHourlyRate	currentHourlyRate	core	Analysis: AWS Recommendations: currentHourlyRate on page 114
currentRiCoverage	currentRiCoverage	core	Analysis: AWS Recommendations: currentRiCoverage on page 114
currentType	currentType	core	Analysis: AWS Recommendations: currentType on page 108
dataQuality.completeDays	dq_completeDays	ancillary	Analysis: AWS Recommendations: dataQuality on page 119 ***Note**: Ancillary properties must be used in conjunction with at least one core property for the purpose of filtering subscription results. Refer to Ancillary Properties for details.*

propertyName	aliasName	Core / Ancillary	Description Reference
dataQuality.firstSeen	dq_firstSeen	ancillary	Analysis: AWS Recommendations: dataQuality on page 119
dataQuality.lastSeen	dq_lastSeen	ancillary	Analysis: AWS Recommendations: dataQuality on page 119
dataQuality.partialDays	dq_partialDays	ancillary	Analysis: AWS Recommendations: dataQuality on page 119
dataQuality.workloadName	dq_workloadName	ancillary	Analysis: AWS Recommendations: dataQuality on page 119
deferRecommendation	deferRecommendation	core	Analysis: AWS Recommendations: deferRecommendation on page 115
densifyPolicy	densifyPolicy	core	Analysis: AWS Recommendations: densifyPolicy on page 112
effortEstimate	effortEstimate	core	Analysis: AWS Recommendations: effortEstimate on page 113
entityId	entityId	core	Analysis: AWS Recommendations: entityId on page 108
implementationMethod	implementationMethod	core	Analysis: AWS Recommendations: implementationMethod on page 110
maxGroupCurrent	maxGroupCurrent	core	Analysis: AWS Recommendations: minGroupCurrent on page 115
maxGroupRecommended	maxGroupRecommended	core	Analysis: AWS Recommendations: maxGroupRecommended on page 116
minGroupCurrent	minGroupCurrent	core	Analysis: AWS Recommendations: minGroupCurrent on page 115
minGroupRecommended	minGroupRecommended	core	Analysis: AWS Recommendations: minGroupRecommended on

propertyName	aliasName	Core / Ancillary	Description Reference
			page 116
name	name	core	Analysis: AWS Recommendations: name on page 111 Analysis: Azure Recommendations: name on page 137 Analysis: GCP Recommendations: name on page 170
powerState	powerState	core	Analysis: AWS Recommendations: powerState on page 113
predictedUptime	predictedUptime	core	Analysis: AWS Recommendations: predictedUptime on page 110
recommendationType	recommendationType	core	Analysis: AWS Recommendations: recommendationType on page 109 Analysis: Azure Recommendations: recommendationType on page 135 Analysis: GCP Recommendations: recommendationType on page 168 ***Note**: To filter "Just Right" recommendations or "Not Analyzed" systems, you must use the exact recommendationType property string,* `"Not Analyzed"` *or* `"Just Right"`*, when you use the "=" or "like" operator.*
recommendedCost	recommendedCost	core	Analysis: AWS Recommendations: recommendedCost on page

propertyName	aliasName	Core / Ancillary	Description Reference
			113
recommendedHostEntityId	recommendedHostEntityId	core	Analysis: AWS Recommendations: recommendedHostEntityId on page 113
recommendedHourlyRate	recommendedHourlyRate	core	Analysis: AWS Recommendations: recommendedHourlyRate on page 114
recommendedType	recommendedType	core	Analysis: AWS Recommendations: recommendedType on page 110
recommFirstSeen	recommFirstSeen	ancillary	Analysis: AWS Recommendations: recommFirstSeen on page 117 ***Note**: Ancillary properties must be used in conjunction with at least one core property for the purpose of filtering subscription results. Refer to Ancillary Properties for details.*
recommLastSeen	recommLastSeen	ancillary	Analysis: AWS Recommendations: recommLastSeen on page 117
recommSeenCount	recommSeenCount	ancillary	Analysis: AWS Recommendations: recommSeenCount on page 117
region	region	core	Analysis: AWS Recommendations: region on page 108
resourceId	resourceId	core	Analysis: AWS Recommendations: resourceId on page 108
rptHref	rptHref	core	Analysis: AWS Recommendations: rptHref on page 111

propertyName	aliasName	Core / Ancillary	Description Reference
savingsEstimate	savingsEstimate	core	Analysis: AWS Recommendations: savingsEstimate on page 112
serviceType	serviceType	core	Analysis: AWS Recommendations: serviceType on page 114 Analysis: Azure Recommendations: serviceType on page 138 Analysis: GCP Recommendations: serviceType on page 171
totalHoursRunning	totalHoursRunning	core	Analysis: AWS Recommendations: totalHoursRunning on page 111

Default Container Properties

Default Container Subscriptions Properties List

The default Container Subscriptions Properties catalog contains both core and ancillary properties which can be used by all API users (i.e. default properties are all global).

***Note**: Ancillary properties must be used in conjunction with at least one core property for the purpose of filtering subscription results. Refer to Ancillary Properties for details.*

propertName	aliasName	Core / Ancillary	Description Reference
auditInfo.dataCollection.auditCount	dc_auditCount	ancillary	Analysis: Kubernetes Container Recommendations: auditInfo on page 187
auditInfo.dataCollection.dateFirstAudited	dc_firstAudited	ancillary	Analysis: Kubernetes Container Recommendations: auditInfo on page 187
auditInfo.dataCollection.dateLastAudited	dc_lastAudited	ancillary	Analysis: Kubernetes Container Recommendations: auditInfo

propertName	aliasName	Core / Ancillary	Description Reference
			on page 187
auditInfo.workloadDataLast30.firstDate	last30_firstDate	ancillary	Analysis: Kubernetes Container Recommendations: auditInfo on page 187
auditInfo.workloadDataLast30.lastDate	last30_lastDate	ancillary	Analysis: Kubernetes Container Recommendations: auditInfo on page 187
auditInfo.workloadDataLast30.seenDays	last30_seenDays	ancillary	Analysis: Kubernetes Container Recommendations: auditInfo on page 187
auditInfo.workloadDataLast30.totalDays	last30_totalDays	ancillary	Analysis: Kubernetes Container Recommendations: auditInfo on page 187
cluster	cluster	core	Analysis: Kubernetes Container Recommendations: cluster on page 186
container	container	core	Analysis: Kubernetes Container Recommendations: container on page 186
currentCount	currentCount	core	Analysis: Kubernetes Container Recommendations: currentCount on page 188
currentCpuLimit	currentCpuLimit	core	Analysis: Kubernetes Container Recommendations: currentCpuLimit on page 190
currentCpuRequest	currentCpuRequest	core	Analysis: Kubernetes Container Recommendations: currentCpuRequest on page 191
currentMemLimit	currentMemLimit	core	Analysis: Kubernetes Container Recommendations: currentMemLimit on page 189

propertName	aliasName	Core / Ancillary	Description Reference
currentMemRequest	currentMemRequest	core	Analysis: Kubernetes Container Recommendations: recommendedMemRequest on page 191
controllerType	controllerType	core	Analysis: Kubernetes Container Recommendations: controllerType on page 190
dataQuality.completeDays	dq_completeDays	ancillary	Analysis: Kubernetes Container Recommendations: dataQuality on page 190
dataQuality.firstSeen	dq_firstSeen	ancillary	Analysis: Kubernetes Container Recommendations: dataQuality on page 190
dataQuality.lastSeen	dq_lastSeen	ancillary	Analysis: Kubernetes Container Recommendations: dataQuality on page 190
dataQuality.partialDays	dq_partialDays	ancillary	Analysis: Kubernetes Container Recommendations: dataQuality on page 190
dataQuality.workloadName	dq_workloadName	ancillary	Analysis: Kubernetes Container Recommendations: dataQuality on page 190
displayName	displayName	core	Analysis: Kubernetes Container Recommendations: displayName on page 186
entityId	entityId	core	Analysis: Kubernetes Container Recommendations: entityId on page 190
hostName	hostName	core	Analysis: Kubernetes Container Recommendations: hostName on page 186
namespace	namespace	core	Analysis: Kubernetes

propertName	aliasName	Core / Ancillary	Description Reference
			Container Recommendations: namespace on page 191
podService	podService	core	Analysis: Kubernetes Container Recommendations: podService on page 187
predictedUptime	predictedUptime	core	Analysis: Kubernetes Container Recommendations: predictedUptime on page 186
recommendationType	recommendationType	core	Analysis: Kubernetes Container Recommendations: recommendationType on page 189 **Note**: *To filter "Just Right" recommendations or "Not Analyzed" systems, you must use the exact recommendationType property string,* `"Not Analyzed"` *or* `"Just Right"`*, when you use the "=" or "like" operator.*
recommendedCpuLimit	recommendedCpuLimit	core	Analysis: Kubernetes Container Recommendations: recommendedCpuLimit on page 188
recommendedCpuRequest	recommendedCpuRequest	core	Analysis: Kubernetes Container Recommendations: recommendedCpuRequest on page 189
recommendedMemRequest	recommendedMemRequest	core	Analysis: Kubernetes Container Recommendations: recommendedMemRequest on page 191

propertName	aliasName	Core / Ancillary	Description Reference
recommendedMemLimit	recommendedMemLimit	core	Analysis: Kubernetes Container Recommendations: recommendedMemLimit on page 189
recommFirstSeen	recommFirstSeen	ancillary	Analysis: Kubernetes Container Recommendations: recommFirstSeen on page 190
recommLastSeen	recommLastSeen	ancillary	Analysis: Kubernetes Container Recommendations: recommLastSeen on page 186
recommSeenCount	recommSeenCount	ancillary	Analysis: Kubernetes Container Recommendations: recommSeenCount on page 188

Examples

Example: Getting a Collection of Private Cloud Subscriptions Properties

The following example shows you how to retrieve a collection of your private cloud Subscriptions properties. This example assumes that your username is "saas".

Example: Getting a Collection of Private Subscriptions Properties

Request:

```
GET /subscriptions/cloud/properties/?type=owner
```

Response:

```
{
    "propertyRef": "fe602c6f-77c2-4105-9ec4-aecce77cc104",
    "propertyName": "deferUntil",
    "aliasName": "Defer Purchase RI Until",
    "owner": "saas"
}
```

Example: Getting a Specific Container Subscriptions Property

The following example shows you how to retrieve a specific container Subscriptions property with a known property reference ID. This property must be of type "*global*" or owned by you before a successful response is returned.

Example: Getting a Specific Container Subscriptions Property

Request:

```
GET /subscriptions/containers/properties/3f6485a0-5106-4d25-a7c5-13f4270462ef
```

Response:

```
{
    "propertyRef": "3f6485a0-5106-4d25-a7c5-13f4270462ef",
    "propertyName": "controllerType",
    "aliasName": "controllerType",
    "owner": ""
}
```

Example: Adding New Subscriptions Properties

This example shows you how to add new properties to the Cloud Subscriptions Properties catalog. Notice that the `owner` parameter is not set. If you are a non-administrative Densify user authenticating the POST request, the `owner` parameter is automatically set to your username. By having the `owner` parameter set, the property is considered *private* and can only be used by you (or any administrative user). If you are a Densify **administrative user**[1] and you do not set the `owner` parameter in the POST request, then the property is considered *global*.

[1] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

If there is an error in the POST request resulting from any one of the property additions, then all the property additions in the request body are rolled back and not committed.

Example: Adding Properties to the Cloud Subscriptions Properties Catalog

Request:

```
POST /subscriptions/cloud/properties
```

Parameters:

```
[
  {
    "propertyName" :   "deferUntil",
    "aliasName"    :   "Defer Purchase RI Until"
  },
  {
    "propertyName": "deferRecommendation",
    "aliasName": "Defer Recommendation"
  }
]
```

Example: Modifying Subscriptions Properties

The example below shows you how to modify two Subscriptions properties in the `/subscriptions/cloud/properties` resource catalog.

Observe the following behavior in the PUT request example:

- `propertyRef` and `propertyName` parameters are mandatory for identifying the property to modify.
- For both modification entries, the `aliasName` and `owner` parameters are updated.
- You must be an administrative user to call the PUT request to update the `owner` parameter to an empty string. This promotes the property from *private* type to *global* type.

Example: Modifying Properties from the Cloud Subscriptions Properties Catalog

Request:

```
PUT /subscriptions/cloud/properties
```

Parameters:

```
[
  {
    "propertyRef": "f8da8042-5186-45aa-8e22-bf5b589f95a6",
    "propertyName": "deferUntil",
    "aliasName": "DU",
    "owner":    ""
  },
  {
    "propertyRef": "fe602c6f-77c2-4105-9ec4-aecce77cc104",
    "propertyName": "deferRecommendation",
    "aliasName": "DR",
    "owner":    ""
  }
]
```

Example: Modifying a Subscriptions Property

This example shows you how to modify a single Subscriptions property using the `/subscriptions/cloud/properties/<propertyRef>` resource. This PUT request updates `aliasName` parameter for the specified property. The `propertyName` and `owner` parameters are mandatory in a PUT request.

Example: Modifying a Property from the Cloud Subscriptions Properties Catalog

Request:

```
PUT /subscriptions/cloud/properties/f8da8042-5186-45aa-8e22-bf5b589f95a6
```

Parameters:

```
{
    "propertyName": "deferUntil",
    "aliasName": "Defer RI purchase Until Date",
    "owner":  "saas_user"
}
```

Example: Deleting Subscriptions Properties

This example shows you how to delete a collection of properties from the `/subscriptions/cloud/properties` resource catalog.

Example: Deleting a Collection of Properties from the Cloud Subscriptions Properties Catalog

Request:

```
DELETE /subscriptions/cloud/properties
```

Parameters:

```
[
  {
    "propertyRef": "f8da8042-5186-45aa-8e22-bf5b589f95a6"
  },
  {
    "propertyRef": "fe602c6f-77c2-4105-9ec4-aecce77cc104"
  }
]
```

Response:

```
[
  {
    "propertyRef": "f8da8042-5186-45aa-8e22-bf5b589f95a6",
    "status": "404",
    "message": "Not found."
  },
  {
    "propertyRef": "fe602c6f-77c2-4105-9ec4-aecce77cc104",
    "message": "Delete successfully."
  }
]
```

Subscriptions: Status

Description

The `/subscriptions/<platformType>/<subscriptionId>/status` resource is used to return the status of the results posted to a webhook for a specific subscription in Densify.

Refer to *Subscriptions* (Help Topic ID 340690) for details on subscription results and defining a webhook.

Resource

```
/subscriptions/cloud/<subscriptionId>/status
/subscriptions/containers/<subscriptionId>/status
/subscriptions/<subscriptionId>/status
```

***Note**: If you use this resource without the `<platformType>` specified (i.e. without `cloud` or `containers` specified), the behavior is exactly the same as specifying the `cloud`-specific resource. This behavior enables backward compatibility with scripts using the Densify API prior to release 12.1.6, where the platform-specific indicator was not available.*

Supported Operations

Table: Subscription Status Supported Operations

HTTP Method	Input	Output	Description
`GET /subscriptions/`*`<platformType>`*`/<`*`subscriptionRef`*`>/status`	Path Parameter: - platformType - subscriptionRef	- lastTriggered - webHookStatus	Returns the last subscription results request and webhook status of the platform-specific subscription identified by `<subscriptionRef>`. See Example: Getting the Status of a Subscription on page 489.

Parameters

Path Parameters

Table: Subscription Status Path Parameters

Parameter Name	Type	Required (Y/N)	Description
platformType	string	Y	[cloud|containers] Specify the technology platform for the subscription resource.
subscriptionRef	string	Y	Specify the unique subscription identifier.

Response

The following is a list of possible response elements returned for the `/subscriptions/`*`<platformType>`*`/<`*`subscriptionRef`*`>/status` resource.

Table: Subscription Status Response Schema

Element	Type	Filter/Sort	Description
lastTriggered	string		The status, date, and time of the last request for subscription results: - On-Demand Success—the last request was on-demand and it was successful; - On-Demand Failure—the last request was on-

Element	Type	Filter/Sort	Description
			demand and it failed to produce results; - Scheduled Success—the last request was a successful scheduled subscription event posted to a webhook; - Scheduled-Failure—the last request was a failed webhook post of the scheduled subscription event.
webHookStatus	string		The status, date, and time of the last subscription results pushed to the webhook location: - Success—subscription results were sent to the webhook successfully; - Failure—transmission of subscription results to the webhook failed.
Returned Status			
message	string		The message for the status response is returned.
status	number		The HTTP response code of the request. Possible status values include: - 200—success with request (usually with content in response body); - 204—success with request, no content returned; - 400—bad request (invalid parameters, logical errors); - 401—authentication failed; - 404—resource not found (or no privileges); - 415—unsupported media type; - 500—internal server error.

Examples

Example: Getting the Status of a Subscription

The following example shows you how to retrieve the status of a specific cloud subscription. The status message for the subscription will be returned if the subscription is global or owned by you.

Example: Getting a Specific Cloud Subscription Status

Request:

```
GET /subscriptions/cloud/4503e5ff-7a1e-4f4c-a106-0f31ca38dc22/status
```

Response:

```
{
    "lastTriggered": "On-Demand Success.  Thu Jan 02 16:41:52 EST 2020",
    "webhookStatus": "Failure. java.net.ConnectException: Connection refused
(Connection refused). Thu Jan 02 16:41:53 EST 2020"
}
```

Subscriptions: Suppressions

Description

The `/subscriptions/<platformType>/suppressions` resource is used to store a list of attribute tags and recommendation properties available to the Subscriptions resource for the purpose of suppressing specific sets of system recommendations from the resulting subscription data set.

This resource acts as a catalog of the suppression tags or properties to be referenced by the suppressionReferences parameter in the Subscriptions resource . Suppression entries (tags or properties) not defined in a `/subscriptions/<platformType>/suppressions` resource cannot be referenced; you must define the suppression entries before using them in a suppression condition. See *Subscriptions* (Help Topic ID 340690) for details on defining a subscription.

There is a catalog for each supported `<platformType>`, which can only be referenced by the corresponding `<platformType>` subscription. For example, a container subscription (i.e. `/subscriptions/containers`) can only reference suppressions from the Container Subscriptions Suppressions catalog (i.e. `/subscriptions/containers/suppressions`).

Suppression entries can be declared as *global* or *private* (i.e. user-specific). Global suppression entries can be used by any API enabled user, while private entries can only be used by their owners. Note that **administrative users**[1] have access to all suppression entries - global or private user-specific.

[1] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

Resource

```
/subscriptions/cloud/suppressions
/subscriptions/containers/suppressions
/subscriptions/suppressions
```

***Note**: If you use this resource without the `<platformType>` specified (i.e. without `cloud` or `containers` specified), the behavior is exactly the same as specifying the `cloud`-specific resource. This behavior enables backward compatibility with scripts using the Densify API prior to release 12.1.6, where the platform-specific indicator was not available.*

Supported Operations

Table: Subscriptions Suppressions Supported Operations

HTTP Method	Input	Output	Description
`GET /subscriptions/<platformType>/suppressions`	Path Parameter: platformType Query String Parameter Options: type owner	Collection of suppressionRef suppressionName attributeName | propertyName key technology aliasName owner	Returns a list of existing suppression tags or properties from the platform-specific Subscriptions Suppressions catalog. The type query string parameter is used to return global or private suppressions. If type is not specified, all global and only private suppressions belonging to you are returned. Administrative users[1] can use the owner query string parameter

[1] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

HTTP Method	Input	Output	Description
			to return all the private suppressions belonging to a specific user. See Example: Getting a List of Available Cloud Subscriptions Suppressions.
`GET /subscriptions/<platformType>/suppressions/<suppressionRef>`	Path Parameters: - platformType - suppressionRef	- suppressionRef - suppressionName - attributeName | propertyName - key - technology - aliasName - owner	Returns a Subscriptions suppression with unique identifier `<suppressionRef>` from a platform-specific Subscriptions Suppressions catalog. See Example: Getting a Specific Container Subscriptions Suppression.
`POST /subscriptions/<platformType>/suppressions`	Path Parameter: - platformType Collection of Request Body Parameters: - suppressionName - attributeName | propertyName - key - technology - aliasName - owner	Collection of - suppressionRef - attributeName | propertyName See Example: Adding New Subscription Suppressions.	Adds new suppression entries into a platform-specific Subscriptions Suppressions catalog. A suppression entry can be either a tag or a property suppression. You can specify an `attributeName` or a `propertyName`, but not both. - **Administrative users**[1] can add global or private suppressions.

[1] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

HTTP Method	Input	Output	Description
			Non-administrative users can add only private suppressions. suppressionName, propertyName, attributeName, and aliasName are validated for uniqueness, depending on their global or private-scoped suppression. The suppressions bulk-add operation is committed as a whole unit. One suppression add error will roll back the entire bulk-add operation. For property suppressions, the property platform must correspond with the suppression platform. For example, you can only add container properties into the Container Subscriptions Suppressions catalog.
`PUT /subscriptions/<platformType>/suppressions`	Path Parameter: platformType Request Body Parameters: Collection of	Collection of suppressionRef attributeName | propertyName	Deletes and replaces parameters for existing suppression entries in a platform-specific Subscriptions Suppressions

HTTP Method	Input	Output	Description
	- suppressionRef - attributeName | propertyName - suppressionName - key - technology - aliasName - owner		catalog. You must specify all parameters (except `suppressionRef`) required for the suppression you want to update. - Only an **administrative user**[1] can modify the `owner` parameter to promote a private suppression to a global suppression (i.e. set `owner=""`). - If you are not an administrative user, you can only set the `owner` parameter to your username. - suppressionName, propertyName, attributeName, and aliasName are validated for uniqueness, depending on their global or private-scoped suppression. - propertyName is platform-validated (i.e. property and suppression platforms must be the same). - The multiple suppression

[1] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

HTTP Method	Input	Output	Description
			delete-and-replace operation is committed as a whole unit; an error resulting from one of the suppression entries will roll back the entire operation. See Example: Modifying Subscription Suppressions on page 506.
`PUT /subscriptions/suppressions/<suppressionRef>`	Path Parameters: • platformType • suppressionRef Request Body Parameters: • attributeName | propertyName • suppressionName • key • technology • aliasName • owner	• suppressionRef • attributeName | propertyName	Deletes and replaces parameters for an existing suppression entry identified by `<suppressionRef>` in a platform-specific Subscriptions Suppressions catalog. You must specify all parameters required (in the Request Body Parameters section) for the suppression you want to update, as all previous parameters are deleted. • Only an **administrative user**[1] can modify the `owner` parameter to promote a private suppression to a

[1] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

HTTP Method	Input	Output	Description
			global suppression (i.e. set `owner=""`). • If you are not an administrative user, you can only set the `owner` parameter to your username. • suppressionName, propertyName, attributeName, and aliasName are validated for uniqueness, depending on their global or private-scoped suppression. • propertyName is platform-validated (i.e. property and suppression platforms must be the same). See Example: Modifying a Subscription Suppression on page 507.
`DELETE /subscriptions/<platformType>/suppressions`	Path Parameter: • platformType Collection of • suppressionRef	• HTTP status of "204 No Content" if all delete operations are successful • If delete errors occur, the following is returned for each delete suppression	Deletes suppressions from a platform-specific Subscriptions Suppressions catalog. • Administrative users[1] can delete any global or private suppressions. • Non-

[1] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

HTTP Method	Input	Output	Description
		request: • suppressionRef • status • message See Example: Deleting Subscriptions Suppressions on page 507.	administrative users can only delete their own private suppression from the catalog. • Suppressions referenced by subscriptions cannot be deleted. • Each suppression delete operation is independent from the other delete operations in the same request (an error with one suppression delete action does not affect the delete actions of the other suppressions in the same request).
`DELETE /subscriptions/<platformType>/ suppressions/ <suppressionRef>`	Path Parameters: • platformType • suppressionRef	• HTTP status of "204 No Content" if delete operation is successful; • HTTP status of "404 Not Found" if suppression is not found; • If the suppression is referenced by a subscription, or if there are other errors, then the following is returned: • suppressionRef • suppressionName	Deletes a suppression with `<suppressionRef>` identifier from a platform-specific Subscriptions Suppressions catalog. • Administrative users[1] can delete any global or private suppression. • Non-administrative users can only

[1] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

HTTP Method	Input	Output	Description
		• message	delete their own private suppression from the catalog. A suppression referenced by subscriptions cannot be deleted.

Parameters

Path Parameters

Table: Subscriptions Suppressions Path Parameters

Parameter Name	Type	Required (Y/N)	Description
platformType	string	Y	[cloud|containers] Specify the technology platform for the Subscriptions suppression resource.
suppressionRef	string	Y	Specify the unique identifier for a Subscriptions suppression entry.

Query String Parameters

Table: Subscriptions Suppressions Query String Parameters

Parameter Name	Type	Required (Y/N)	Description
type	string		Specify the type of Subscriptions suppression to return: all—Return all suppression entries: global and private user-specific. If you are not an **administrative user**[1], only private suppressions owned by you and global suppressions are returned. This is the default behavior if `type` is not specified in the request. global—Return all global Subscriptions suppression entries.

[1] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

Parameter Name	Type	Required (Y/N)	Description
			- owner—Return user-specific Subscriptions suppression entries. If you are not an administrative user, only private suppressions owned by you are returned. If you are an administrative user, all global and private suppressions are returned. A Subscriptions suppression is considered *global* if the owner parameter is not populated. Global Subscriptions suppressions can be used by all Densify API users. A Subscriptions suppression is considered *private* if the owner parameter contains a Densify username. Private Subscriptions suppressions can only be used by their owners or administrative users.
owner	string		If you are an **administrative user**[1], you can specify a Densify username in conjunction with the `type=owner` query string parameter to return all of the specified user's private Subscriptions suppressions. If you are not an administrative user, you can request for your own private suppressions. If you use the `?type=owner&owner=<anotherusername>` query string option with a username other than your own, the returned response is a 400 Bad Request -"Current login user cannot query for owner" error.

Request Body Parameters

Table: Subscriptions Suppressions Request Body Parameters

Parameter Name	Type	Required (C-create/M-modify/D-delete)	Description
suppressionRef	string	M D	Specify the unique identifier for a Subscriptions suppression entry.
suppressionName	string	C M	Specify a unique name for a Subscriptions suppression entry. For *global* Subscriptions suppressions, the suppressionName must be unique within a platform-specific Subscriptions Suppressions catalog. For *private* Subscriptions suppressions, the suppressionName must be unique per owner and across all

[1] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

Parameter Name	Type	Required (C-create/M-modify/D-delete)	Description
			global subscription suppressions per platform-specific catalog. For example, owner A and owner B can both have a private suppression named "SuperA", as long as "SuperA" is not also a global suppression within the same platform-specific catalog.
aliasName	string		Specify an alias name for the Subscriptions attribute tag. For *global* suppressions, the aliasName must be unique within a platform-specific Subscriptions Suppressions catalog. For *private* suppressions, the aliasName must be unique per owner and across all global suppressions per platform-specific catalog. For example, owner A and owner B can both have a private suppression alias named "Joe", as long as "Joe" is also not a global suppression alias within the same platform-specific catalog.
attributeName	string	C M	Specify the attribute name for the Subscriptions suppression entry. The attribute name must exist in the Densify standard set of system attributes. Contact Support@Densify.com for a list of available system attributes. Use the "`Resource Tags`" attribute name along with key and technology parameters for cloud technology-specific resource attributes in the "*`key:value`*" form. See Example: Adding New Subscription Suppressions for "Resource Tags" usage. Use the "`Container Labels`" attribute name along with key and technology parameters for container-specific attributes in "*`key:value`*" form. Note that the `attributeName` must be unique in a platform-specific Subscriptions Suppressions catalog (i.e. you cannot have two suppressions with the same `attributeName` in a catalog).
key	string	CM[1]	Specify the key string required for the technology platform resource attribute. If the Subscriptions suppression is a reference to a resource attribute (i.e. `"attributeName": "Resource Tags"` or `"attributeName": "Container Labels"`), you need to specify both the key and technology platform for the specific resource attribute.

[1] For resource attributes, you need to specify both `key` and `technology` parameters.

Parameter Name	Type	Required (C-create/M-modify/D-delete)	Description
technology	string	CM[1]	Specify the technology platform for the resource attribute. Currently, the following technology platforms are supported: - AWS - CONTAINER If the Subscriptions suppression is a reference to a resource attribute (i.e. `"attributeName": "Resource Tags"` or `"attributeName": "Container Labels"`), you need to specify both the key and technology platform for the specific resource attribute.
propertyName	string	C M	Specify the recommendation element for the Subscriptions suppression. The list of available recommendation elements can be found in the *Analysis: technology-specific Recommendations* page. For example, refer to the Response schema section of the Analysis: AWS Recommendations on page 105 for a full list of AWS recommendation elements. The `propertyName` must be unique within a platform-specific Subscriptions Suppressions catalog.
owner	string	M[2]	When the `owner` parameter is not set, the Subscriptions suppression is considered *global*. Global Subscriptions suppressions can be used by all API users. Only **administrative users**[3] can create global suppressions. When the `owner` parameter is set, the suppression is considered *private*. Private Subscriptions suppressions can only be used by their owners or administrative users. If you are an administrative user, you have the ability to assign any Densify user as the owner of the subscription suppression in a POST request. In a PUT request, administrative users can promote the suppression from private to global by setting `owner: ""`. If you are not an administrative user, you can only set the `owner` parameter to your Densify username. In a POST request, the `owner` parameter is automatically populated with your username.

[1] For resource attributes, you need to specify both `key` and `technology` parameters.

[2] The `owner` parameter is mandatory for private Subscriptions suppressions.

[3] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

Response

The following is a complete list of possible response elements that are returned for the `/subscriptions/suppressions` resource.

Table: Subscriptions Suppressions Response Schema

Element	Type	Filter/Sort	Description
suppressionRef	string		The unique referenced ID of the Densify Subscriptions suppression entry.
suppressionName	string		The Subscriptions suppression name.
aliasName	string		The Subscriptions suppression alias name.
attributeName	string		The attribute name for the Subscriptions suppression entry.
technology	string		The Subscriptions suppression's technology platform for the resource attribute.
key	string		The resource attribute key string for the suppression's associated technology platform.
propertyName	string		The Subscriptions suppression property name.
owner	string	F	The designated user/owner of this Subscriptions suppression. A Subscriptions suppression is considered *global* if the owner parameter is not populated. Global Subscriptions suppressions can be used by all Densify API users. A Subscriptions suppression is considered *private* if the owner parameter contains a Densify username. Private Subscriptions suppressions can only be used by their owners or **administrative users**[1].
message	string		The message for the error or status response is returned.
status	number		The HTTP response code of the request. Possible status values include: 200—success with request (usually with content in response body); 204—success with request, no content returned; 400—bad request (invalid parameters, logical errors); 401—authentication failed; 404—resource not found (or no privileges); 415—unsupported media type; 500—internal server error.

[1] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

Examples

Example: Getting a List of Available Cloud Subscriptions Suppressions

The following example shows you how to retrieve a list of suppressions available to you from the Cloud Subscriptions Suppressions catalog.

Example: Getting a List of Available Cloud Subscription Suppressions

Request:

```
GET /subscriptions/cloud/suppressions
```

Response:

```
[
    {
        "suppressionRef": "2ff4501e-df32-4f57-8a77-e539192fa043",
        "suppressionName": "Entity ID",
        "propertyName": "entityId",
        "aliasName": "supprEntityID",
        "owner": ""
    },
    {
        "suppressionRef": "38f6e37c-86f4-44a4-9741-dc9d179cbbc1",
        "suppressionName": "Effort Estimate",
        "propertyName": "effortEstimate",
        "aliasName": "supprEffortEstimate",
        "owner": ""
    }
    ...
]
```

Example: Getting a Specific Container Subscriptions Suppression

The following example shows you how to retrieve a specific container Subscriptions suppression with a known reference ID. This suppression must be of type "*global*" or owned by you before a successful response is returned.

Example: Getting a Specific Container Subscriptions Suppression Entry

Request:

```
GET /subscriptions/containers/suppressions/8b58927e-8f1a-4105-b8f2-5f2b0f-
d0238d
```

Response:

```
{
    "suppressionRef": "fdc363b2-523c-4bb5-bdbf-a4f4ef994487",
    "suppressionName": "RecommendedMemRequest",
    "propertyName": "recommendedMemRequest",
    "aliasName": "supprRecMemRequest",
    "owner": ""
}
```

Example: Adding New Subscription Suppressions

This example shows you how to add new suppression entries to the Cloud Subscriptions Suppressions catalog. Notice that the `owner` parameter is not set. If you are a non-administrative Densify user authenticating the POST request, the `owner` parameter is automatically set to your username. By having the `owner` parameter set, the suppression is considered *private* and can only be used by you (or any administrative user). If you are a Densify **administrative user**[1] and you do not set the `owner` parameter in the POST request, then the suppression is considered *global*.

If there is an error in the POST request resulting from any one of the suppression additions, then all the suppression additions in the request body are rolled back and not committed.

Example: Adding Suppressions to the Cloud Subscriptions Suppressions Catalog

Request:

```
POST /subscriptions/cloud/suppressions
```

Parameters:

```
[
    {
        "suppressionName": "Health_Check",
        "attributeName": "Health Check",
```

[1] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

```
        "aliasName": "Suppr_Health_Check"
      },
      {
        "suppressionName": "AWS RTag Env Suppr",
        "attributeName": "Resource Tags",
        "key": "Environment :",
        "technology": "AWS",
        "aliasName": "Suppr_Health_Check"
      },
      {
        "suppressionName": "Recommended_Cost",
        "propertyName": "recommendedCost",
        "aliasName": "Suppr Recommended Cost"
      }
]
```

Example: Modifying Subscription Suppressions

This example shows you how to modify two subscription suppression entries in the Cloud Subscriptions Suppressions catalog. You need to specify all the request body parameters for PUT request. With the exception of the `suppressionRef` and `owner` parameters, all other parameters can be updated If there is an error in the PUT request from any one of the update entries, then all the updates are rolled back and not applied. Note that only the administrative user is eligible to make this PUT request.

Example: Modifying Suppressions from the Cloud Subscriptions Suppressions Catalog

Request:

```
PUT /subscriptions/cloud/suppressions
```

Parameters:

```
[
  {
    "suppressionRef": "61acf182-1773-4988-b4d9-a76c866b5c68",
    "suppressionName": "Business Applications",
    "attributeName": "Business Applications",
    "aliasName": "BusinessApplicationsSuppression",
    "owner":    ""
  },
  {
    "suppressionRef": "c6da9e05-92be-4ec6-9892-7a2ed68d57f0",
    "suppressionName": "Instance Name",
    "propertyName": "name",
    "aliasName": "SuppressThisName",
    "owner":    ""
  }
]
```

Example: Modifying a Subscription Suppression

This example shows you how to modify a single cloud Subscriptions suppression using the `/subscriptions/cloud/suppressions/<suppressionRef>` resource. You need to specify all the request body parameters for a PUT request, even if you only want to modify the `suppressionName` or `aliasName` parameters. In this example, you must either be the "saas" user or the administrative user to be authorized to make this PUT request.

Example: Modifying a Suppression from the Cloud Subscriptions Suppressions Catalog

Request:

```
PUT /subscriptions/cloud/suppressions/d6472966-52bd-4231-a0ab-ea9cae2f5016
```

Parameters:

```
{
    "suppressionName": "OS Arch Suppr",
    "attributeName": "OS Architecture",
    "aliasName": "Suppress OS Architecture",
    "owner":    "saas"
}
```

Example: Deleting Subscriptions Suppressions

This example shows you how to delete a collection of suppressions from the `/subscriptions/cloud/suppressions` resource catalog. Keep in mind that you can only delete your own private suppressions from the catalog. Only administrative users[1] can delete any global and any private suppressions.

Example: Deleting a Collection of Suppressions from the Cloud Subscriptions Suppressions Catalog

Request:

```
DELETE /subscriptions/cloud/suppressions
```

[1] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

Parameters:

```
[
    {
        "suppressionRef": "209baebd-fc67-4902-a7a2-d9386af26a4e"
    },
    {
        "suppressionRef": "9cc94889-b15d-4050-aa8f-d9f605271e58"
    }
]
```

Response:

```
[
    {
        "suppressionRef": "9cc94889-b15d-4050-aa8f-d9f605271e58",
        "status": "404",
        "message": "Not found."
    },
    {
        "suppressionRef": "209baebd-fc67-4902-a7a2-d9386af26a4e",
        "message": "Delete successfully."
    }
]
```

Default Suppressions

Default Cloud Subscriptions Suppression List

The Cloud Subscriptions Suppressions catalog contains the following default entries:

```
[
    {
        "suppressionRef": "03922061-96eb-450b-a30f-a397a19c9a6f",
        "suppressionName": "Effort Estimate",
        "propertyName": "effortEstimate",
        "aliasName": "supprEffortEstimate",
        "owner": ""
    },
    {
        "suppressionRef": "29150379-9252-498d-a59d-de47d183eee8",
        "suppressionName": "Approval Type",
        "propertyName": "approvalType",
        "aliasName": "supprApprovalType",
        "owner": ""
    },
    {
        "suppressionRef": "61acf182-1773-4988-b4d9-a76c866b5c68",
        "suppressionName": "Business Applications",
        "attributeName": "Business Applications",
        "aliasName": "supprBusinessApplications",
        "owner": ""
    },
    {
        "suppressionRef": "93dba027-8391-4303-b413-7c4cb41ec5e9",
        "suppressionName": "Suppress Region",
        "propertyName": "region",
        "aliasName": "supprRegion",
        "owner": ""
    },
    {
        "suppressionRef": "a6827ae4-fa2b-405e-a564-d70f2dad45c2",
        "suppressionName": "recommendationType",
        "propertyName": "recommendationType",
        "aliasName": "supprRecommendationType",
        "owner": ""
    }
]
```

Default Container Subscriptions Suppression List

The Container Subscriptions Suppressions catalog contains the following default entries:

```
[
    {
        "suppressionRef": "fdc363b2-523c-4bb5-bdbf-a4f4ef994487",
        "suppressionName": "RecommendedMemRequest",
        "propertyName": "recommendedMemRequest",
        "aliasName": "supprRecMemRequest",
        "owner": ""
    }
]
```

Subscriptions: Tags

Description

The `/subscriptions/<platformType>/tags` resource is used to store a list of system attributes available to the Subscriptions resource for the purpose of filtering systems from the resulting subscription data set.

This resource acts as a catalog of system attributes to be referenced by the tagReferences parameter in a subscription. System attributes that are not defined in a `/subscriptions/<platformType>/tags` resource, cannot be referenced by a Subscriptions tag filter condition. See *Subscriptions* (Help Topic ID 340690) for details on defining a subscription and using tag filter conditions.

The SQL to filter results employs a "STRING" comparison and not a numeric comparison on tags so you may observe inconsistent results. Tags can only be lexicographically compared.

There is a catalog for each supported `<platformType>`, which can only be referenced by the corresponding `<platformType>` subscription. For example, a container subscription (i.e. `/subscriptions/containers`) can only reference tags from the Container Subscriptions Tags catalog (i.e. `/subscriptions/containers/tags`). Densify provides an initial, default set of system attributes for each `<platformType>` catalog. You can extend these default sets by adding additional system attributes, or you can customize the sets by modifying or deleting existing attributes. Only Densify standard attributes or technology-specific attributes (e.g. AWS, Containers) can be added to the platform-specific Subscriptions Tags catalogs. Contact Support@Densify.com for a complete list of available Densify standard attributes. See Default Cloud Attribute Tags on page 518 or Default Container Attribute Tags.

Attributes in the platform-specific Subscriptions Tags catalog can be declared as *global* or *private* (i.e. user-specific). Global attribute tags can be used by any API enabled user, whereas private attributes can only be used by their owners. Note that **administrative users**[1] have access to both global and private attributes for all users.

Resource

```
/subscriptions/cloud/tags
/subscriptions/containers/tags
/subscriptions/tags
```

***Note**: If you use this resource without the `<platformType>` specified (i.e. without `cloud` or `containers` specified), the behavior is exactly the same as specifying the `cloud`-specific resource. This behavior enables backward compatibility with scripts using the Densify API prior to release 12.1.6, where the platform-specific indicator was not available.*

Supported Operations

Table: Subscriptions Tags Supported Operations

HTTP Method	Input	Output	Description
`GET /subscriptions/<platformType>/tags`	Path Parameter: platformType Query String Parameter Options: type owner	Collection of tagRef tagName aliasName attributeName technology key owner	Returns a list of existing attributes from the platform-specific Subscriptions Tags catalog. The type query string parameter is used to return global or private attribute tags. If type is not specified, all global and only private attribute tags belonging to you are returned. **Administrative users**[2] can use the owner query string parameter to return all the private properties belonging to a specific user. See Example: Getting a Collection of Private Cloud Subscriptions Tags on page 520. See Example: Getting a Collection of Container Subscriptions Tags on page 520.

[1] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

[2] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

HTTP Method	Input	Output	Description
`GET /subscriptions/<platformType>/tags/<tagRef>`	Path Parameters: - platformType - tagRef	- tagRef - tagName - aliasName - attributeName - technology - key - owner	Returns an attribute tag with unique identifier `<tagRef>` from a platform-specific Subscriptions Tags catalog. See Example: Getting a Specific Cloud Subscription Attribute Tag on page 522.
`POST /subscriptions/<platformType>/tags`	Path Parameter: - platformType Request Body Parameters: Collection of - tagName - attributeName - aliasName - key - technology - owner	Collection of - tagRef - tagName	Adds new attributes into a platform-specific Subscriptions Tags catalog. - **Administrative users**[1] can add global or private attribute tags. - Non-administrative users can only add private tags. - attributeName is validated for uniqueness within the catalog (i.e. you cannot define two tags with the same `attributeName`). - tagName and aliasName are validated for uniqueness, depending on the private or global scope of the tag. - The attribute tags bulk-add operation is committed as a whole unit; an error resulting from adding one of the tag entries will roll back the entire operation. See Example: Adding New Subscription Attribute Tags on page 522.
`PUT /subscriptions/<platformType>/tags`	Path Parameter: - platformType Request Body Parameters: Collection of - tagRef - tagName - attributeName - aliasName - key - technology	Collection of - tagRef - tagName - message	Replaces parameters from existing attribute tags in a platform-specific Subscriptions Tags catalog. You must specify all parameters required for the tag you want to update, as all previous parameters are deleted (except `<tagRef>` since it is used to identify the attribute tag to update). - Only an **administrative user**[2] can modify the `owner` parameter to promote a private tag to a global tag (i.e. set `owner=""`).

[1] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

[2] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

HTTP Method	Input	Output	Description
	• owner		• If you are not an administrative user, you can only set the `owner` parameter to your username. • attributeName, tagName, and aliasName are validated for uniqueness, depending on the private or global scope of the tag. • This bulk-edit operation is committed as a whole unit; an error resulting from one of the edits will roll back the entire bulk edit operation. See Example: Modifying Subscription Attribute Tags on page 523.
`PUT /subscriptions/<platformType>/tags/<tagRef>`	Path Parameters: • platformType • tagRef Request Body Parameters: • tagName • attributeName • aliasName • key • technology • owner	• tagRef • tagName • message	Replaces parameters for an existing tag, identified by `<tagRef>`, in a platform-specific Subscriptions Tags catalog. • Only an **administrative user**[1] can modify the `owner` parameter to change a private tag to a global tag (i.e. set `owner=""`). • If you are not an administrative user, you can only set the `owner` parameter to your username. • attributeName, tagName, and aliasName are validated for uniqueness, depending on the private or global scope of the tag. See Example: Modifying a Technology Subscription Attribute Tag on page 524.
`DELETE /subscriptions/<platformType>/tags`	Path Parameter: • platformType Collection of Request Body Parameter: • tagRef	• HTTP status of "204 No Content" if all delete operations are successful • If delete errors occur, then the following is	Deletes attribute tags from a platform-specific Subscriptions Tags catalog. • **Administrative users**[2] can delete any global and any private attribute tags. • Non-administrative users can only delete their own private tags. • Tags referenced in filtering conditions by subscriptions cannot be deleted. • Each tag delete in the bulk-delete

[1] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

[2] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

HTTP Method	Input	Output	Description
		returned for each tag delete request: • tagRef • status • message	operation is an independent action: an error with one tag delete action does not affect the delete actions of the other tags in the request body parameter. See Example: Deleting Subscription Attribute Tags on page 525
`DELETE /subscriptions/<platformType>/tags/<tagRef>`	Path Parameters: • platformType • tagRef	• HTTP status of "204 No Content" if delete operation is successful • HTTP status of "404 Not Found" if tag is not found • If the tag is referenced by a subscription or if there are other errors, then the following is returned: • tagRef • message • status	Deletes attribute tag with `<tagRef>` identifier from a platform-specific Subscriptions Tags catalog. • Administrative users[1] can delete any global and any private attribute tags. • Non-administrative users can only delete their own private tags. • Tags referenced in filtering conditions by subscriptions cannot be deleted. See Example: Deleting a Single Subscription Attribute Tag

Parameters

Path Parameters

Table: Subscriptions Tags Path Parameters

Parameter Name	Type	Required (Y/N)	Description
platformType	string	Y	[cloud|containers] Specify the technology platform for the Subscriptions attribute tag resource.

[1] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

Parameter Name	Type	Required (Y/N)	Description
tagRef	string	Y	Specify the unique identifier for a Subscriptions attribute tag entry.

Query String Parameters

Table: Subscriptions Tags Query String Parameters

Parameter Name	Type	Required (Y/N)	Description
type	string		Specify the type of Subscriptions attribute tag to return: - all—Return all attribute tags: global and private user-specific. If you are not an administrative user, only private attribute tags owned by you and global attribute tags are returned. This is the default behavior if `type` is not specified in the request. - global—Return all global Subscriptions attribute tags. - owner—Return user-specific Subscriptions attribute tags. If you are not an administrative user, only private attribute tags owned by you are returned. If you are an administrative user, all global and private attribute tags are returned. A Subscriptions attribute tag is considered *global* if the owner parameter is not populated. Global Subscriptions tags can be used by all Densify API users. A Subscriptions tag is considered *private* if the owner parameter contains a Densify username. Private Subscriptions tags can only be used by their owners or administrative users.
owner	string		If you are an **administrative user**[1], you can specify a Densify username in conjunction with the `type=owner` query string parameter to return all of the specified user's private Subscriptions attribute tags. If you are not an administrative user, you can request for only your own private attribute tags. If you use the `?type=owner&owner=<anotherusername>` query string option with a username other than your own, the returned response is a 400 Bad Request - "Current login user cannot query for owner" error.

[1] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

Request Body Parameters

Table: Subscriptions Tags Request Body Parameters

Parameter Name	Type	Required (C-create/M-modify/D-delete)	Description
tagRef	string	M D	Specify the unique identifier for a Subscriptions attribute tag entry.
tagName	string	C M	Specify a unique name for a Subscriptions attribute tag entry. For *global* Subscriptions attribute tags, the tagName must be unique within a platform-specific Subscriptions Tags catalog. For *private* Subscriptions attribute tags, the tagNames must be unique per owner and across all global Subscriptions attribute tags per platform-specific catalog. For example, owner A and owner B can both have a private attribute tag named "Bob", as long as "Bob" is not also a global attribute tag within the same platform-specific catalog.
attributeName	string	C M	Specify the attribute name for the Subscriptions attribute tag. The attribute name must exist in the Densify standard set of system attributes. Contact Support@Densify.com for a list of available system attributes. Use the "`Resource Tags`" attribute name along with key and technology parameters for cloud technology-specific resource attributes in the "*`key:value`*" form. See Example: Adding New Subscription Attribute Tags for "Resource Tags" usage. Use the "`Container Labels`" attribute name along with key and technology parameters for container-specific attributes in "*`key:value`*" form. See Default Container Attribute Tags for an example of "Container Labels" usage.
aliasName	string		Specify an alias name for the Subscriptions attribute tag. For *global* attribute tags, the aliasName must be unique system-wide. For *private* attribute tags, the aliasName must be unique per owner and across all global Subscriptions attribute tags. For example, owner A and owner B can both have a private attribute tag alias named "Jane", as long as "Jane" is also not a global attribute tag alias.
key	string	CM[1]	Specify the key string required for the technology platform resource attribute. If the Subscriptions tag is a reference to a resource attribute (i.e.

[1] For resource attributes, you need to specify both `key` and `technology` parameters.

Parameter Name	Type	Required (C-create/M-modify/D-delete)	Description
			`"attributeName": "Resource Tags"` or `"attributeName": "Container Labels"`), you need to specify both the key and technology elements for the specific resource attribute.
technology	string	CM[1]	Specify the technology platform for the resource attribute. Currently, the following technology platforms are supported: - AWS - CONTAINER If the Subscriptions tag is a reference to a resource attribute (i.e. `"attributeName": "Resource Tags"` or `"attributeName": "Container Labels"`), you need to specify both the key and technology platform for the specific resource attribute.
owner	string	M[2]	When the `owner` parameter is not set, the Subscriptions tag is considered *global*. Global Subscriptions tags can be used by all API users, but can only be created by **administrative users**[3]. When the `owner` parameter is set, the tag is considered *private*. Private Subscriptions tags can only be used by their owners or administrative users. If you are an administrative user, you have the ability to assign any Densify user as the owner of the Subscriptions tag in a POST request. In a PUT request, administrative users can promote the tag from private to global by setting `owner: ""`. If you are not an administrative user, you can only set the `owner` parameter to your Densify username. In a POST request, the `owner` parameter is automatically populated with your username.

Response

The following is a complete list of possible response elements that are returned for the `/subscriptions/<platformType>/tags` resource.

[1] For resource attributes, you need to specify both `key` and `technology` parameters.

[2] The owner parameter is mandatory for private Subscriptions tags.

[3] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

Table: Subscriptions Tags Response Schema

Element	Type	Filter/Sort	Description
tagRef	string		The unique referenced ID of the Densify Subscriptions attribute tag.
tagName	string		The Subscriptions attribute tag name.
attributeName	string		The attribute name for the Subscriptions attribute tag entry.
aliasName	string		The Subscriptions attribute tag alias name.
technology	string		The Subscriptions attribute tag technology platform.
key	string		The resource attribute key string for the associated technology platform.
owner	string	F	The designated owner of this Subscriptions attribute tag. A Subscriptions attribute tag is considered *global* if this parameter is not populated. A Subscriptions attribute tag is considered *private* if the owner parameter contains a Densify username.
message	string		The message for the status response.
status	number		The HTTP response code of the request. Possible status values include: - 200—success with request (usually with content in response body); - 204—success with request, no content returned; - 400—bad request (invalid parameters, logical errors); - 401—authentication failed; - 404—resource not found (or no privileges); - 415—unsupported media type; - 500—internal server error.

Default Cloud Attribute Tags

Default Cloud Attribute Tags List

The Cloud Subscriptions Tags catalog contains the following default attributes:

tagName	aliasName	attributeName	technology	key
Availability Zone	availabilityZone	Availability Zone		
AWS Environment	awsEnvironment	Resource Tags	AWS	Env :
AWS Inventory Code	inventoryCode	Resource Tags	AWS	Inventory Code :
AWS Name	awsName	Resource Tags	AWS	Name :
AWS OS	awsOS	Resource Tags	AWS	OS :

tagName	aliasName	attributeName	technology	key
AWS Owner	awsOwner	Resource Tags	AWS	Owner :
Business Applications	businessApplications	Business Applications		
CPU Util (%)	cpuUtil%	CPU Util (%)		
Disk IO (Bytes)	diskIOBytes	Disk IO (Bytes)		
Disk IO (Ops)	diskIOOps	Disk IO (Ops)		
Department	department	Department		
License Group	licenseGroup	License Group		
License Model	licenseModel	License Model		
Mem Util (%)	memUtil%	Mem Util (%)		
Network IO (Bytes)	networkIOBytes	Network IO (Bytes)		
Observed Uptime	observedUptime	Observed Uptime		
OS Architecture	architecture	OS Architecture		
Predicted Uptime	predictedUptime	Predicted Uptime		
Product Code	productCode	Resource Tags	AWS	Product Code :
Resource Tags	resourceTags	Resource Tags		
RDS Multi-AZ Deployment	RDSmultiAZ	RDS Multi-AZ Deployment		
Virtual Cluster	virtualCluster	Virtual Cluster		
Virtual Datacenter	virtualDatacenter	Virtual Datacenter		
Virtual Domain	awsAccount	Virtual Domain		
VPC ID	vpcID	VPC ID		

Default Container Attribute Tags

Default Container Attribute Tags List

The Container Subscriptions Tags catalog contains the following default attributes:

tagName	aliasName	attributeName	technology	key
ContainerPodName	containerPodName	Container Labels	CONTAINER	pod_name :
My Container Labels	containerLabels	Container Labels		
Resource Tags	resourceTags	Resource Tags		
Virtual Datacenter	virtualDatacenter	Virtual Datacenter		
Virtual Domain	virtualDomain	Virtual Domain		

Examples

Example: Getting a Collection of Private Cloud Subscriptions Tags

The following example shows you how to retrieve a collection of your private Subscriptions cloud attributes. The example assumes that your username is "saas".

Example: Getting a Collection of Private Cloud Subscriptions Tags

Request:

```
GET /subscriptions/cloud/tags?type=owner
```

Response:

```
[
    {
    "tagRef": "548b6567-e523-45a4-b779-3d2f24f8d64d",
    "tagName": "AWS Tag 1",
    "aliasName": "AWS RTag Location",
    "attributeName": "Resource Tags",
    "technology": "AWS",
    "key": "aws:location",
    "owner": "saas"
    },
    {
    "tagRef": "b1dc5d9e-3646-4527-b5db-73d31bfb66bf",
    "tagName": "AWS Tag 2",
    "aliasName": "AWS RTag Application",
    "attributeName": "Resource Tags",
    "technology": "AWS",
    "key": "aws:application",
    "owner": "saas"
    }
]
```

Example: Getting a Collection of Container Subscriptions Tags

The following example shows you how to retrieve a collection of Subscriptions container attributes available to you.

Example: Getting a Collection of Container Subscriptions Tags

Request:

```
GET /subscriptions/containers/tags
```

Response:

```
[
    {
        "tagRef": "17d17ac1-52c0-4e4a-a34d-0dd787c6b5d2",
        "tagName": "Virtual Datacenter",
        "aliasName": "virtualDatacenter",
        "attributeName": "Virtual Datacenter",
        "owner": ""
    },
    {
        "tagRef": "5cfb1b8d-b462-40f3-9783-624b3c2127d5",
        "tagName": "Resource Tags",
        "aliasName": "resourceTags",
        "attributeName": "Resource Tags",
        "owner": ""
    },
    {
        "tagRef": "acc99dbc-d21c-4c7d-a160-c62e8295dae1",
        "tagName": "My Container Labels",
        "aliasName": "containerLabels",
        "attributeName": "Container Labels",
        "owner": ""
    },
    {
        "tagRef": "cf424bf2-79ac-4031-aa68-e941327609c2",
        "tagName": "Virtual Domain",
        "aliasName": "virtualDomain",
        "attributeName": "Virtual Domain",
        "owner": ""
    },
    {
        "tagRef": "e61b3263-b715-4bc8-8310-a2bb7145f9e1",
        "tagName": "ContainerPodName",
        "aliasName": "containerPodName",
        "attributeName": "Container Labels",
        "technology": "CONTAINER",
        "key": "pod_name : ",
        "owner": ""
    }
]
```

Example: Getting a Specific Cloud Subscription Attribute Tag

The following example shows you how to retrieve a specific cloud Subscriptions attribute tag with a known tag ID. This attribute tag must be of type "*global*" or owned by you before a successful response is returned.

Example: Getting a Specific Cloud Subscription Attribute Tag

Request:

```
GET /subscriptions/cloud/tags/0e9751c4-2c5c-45a1-ab92-e34c2fa395c8
```

Response:

```
[
    {
    "tagRef": "0e9751c4-2c5c-45a1-ab92-e34c2fa395c8",
    "tagName": "Virtual Domain",
    "aliasName": "AWS Account",
    "attributeName": "Virtual Domain",
    "owner": ""
    }
]
```

Example: Adding New Subscription Attribute Tags

This example shows you how to add two new technology-specific attributes to the Cloud Subscriptions Tags catalog. Notice that the `owner` parameter is not set. If you are a non-administrative Densify user authenticating the POST request, the `owner` parameter is automatically set to your username. By having the `owner` parameter set, the attribute entry is considered *private* and can only be used by you (or any administrative user). If you are a Densify **administrative user**[1] and you do not set the `owner` parameter in the POST request, then `owner` is not set, making the attribute entry *global*.

If there is an error in the POST request from any one of the attribute additions, then all the new attribute additions are rolled back.

Example: Adding Technology-Specific Attributes to the Cloud Subscriptions Tags Catalog

Request:

[1] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

```
POST /subscriptions/cloud/tags
```

Parameters:

```
[
    {
    "tagName":    "AWS RTag Env",
    "attributeName": "Resource Tags",
    "aliasName":     "AWS Environment",
    "key":           "Environment :",
    "technology":    "AWS"
    },
    {
    "tagName":    "AWS RTag LOB",
    "attributeName": "Resource Tags",
    "aliasName":     "AWS Inventory Code",
    "key":           "LOB :",
    "technology":    "AWS"
    }
]
```

Example: Modifying Subscription Attribute Tags

This example shows you how to modify two Subscriptions attribute tags in the Cloud Subscriptions Tags catalog.

Observe the following behavior in the PUT request example:

- Even if you want to modify one parameter (e.g. `aliasName`) you still have to specify all parameters for the existing tag. A PUT request is essentially a DELETE and REPLACE operation.
- This example assumes that you are an **administrative user**[1] to update a global tag (i.e. the first tag has `"owner": ""` , implying that it is a *global* tag).
- `tagName` and `aliasName` are validated for uniqueness, depending on the private or global scope of the tags.
- If there is an error in the PUT request from any one of the update entries, then all the updates are rolled back and not applied.

Example: Modifying an Attribute Tag from the Cloud Subscriptions Tags Catalog

Request:

[1] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

```
PUT /subscriptions/cloud/tags
```

Parameters:

```
[
    {
    "tagRef": "0e9751c4-2c5c-45a1-ab92-e34c2fa395c8",
    "tagName": "Virtual Domain",
    "aliasName": "My Virtual Domain",
    "attributeName": "Virtual Domain",
    "owner":    ""
    },
    {
    "tagRef": "7e6c4c8a-82d7-44ea-b254-56c12a8af449",
    "tagName": "Notes",
    "aliasName": "SaaS Notes",
    "attributeName": "Notes",
    "owner": "saas"
    }
]
```

Example: Modifying a Technology Subscription Attribute Tag

This example shows you how to modify a single technology-specific Subscriptions attribute tag using the `/subscriptions/cloud/tags/<tagRef>` resource. This PUT request uses administrative privileges to update `tagName` and `aliasName` parameters for the specified tag.

Example: Modifying a Technology Attribute Tag from the Cloud Subscriptions Tags Catalog

Request:

```
PUT /subscriptions/cloud/tags/e4558ad1-0a39-428a-a24e-8e958debfd60
```

Parameters:

```
{
    "tagName": "AWS Environment",
    "aliasName": "awsEnvironment",
    "attributeName": "Resource Tags",
    "technology": "AWS",
    "key": "Env : ",
    "owner": ""
}
```

Example: Deleting Subscription Attribute Tags

This example shows you how to delete a collection of attribute tags from the `/subscriptions/cloud/tags` resource catalog. Keep in mind that you can only delete your own private attribute tags from the catalog. Only **administrative users**[1] can delete any global and any private attribute tags.

Example: Deleting a Collection of Attribute Tags from the Cloud Subscriptions Tags Catalog

Request:

```
DELETE /subscriptions/cloud/tags
```

Parameters:

```
[
  {
    "tagRef":    "28eaf716-e66a-4518-a10a-542641b5b155"
  },
  {
    "tagRef": "f9ce84e0-27f2-4c66-96cf-874800c84b42"
  }
]
```

Example: Deleting a Single Subscription Attribute Tag

This example shows you how to delete a single attribute tags from the Cloud Subscriptions Tags catalog. Keep in mind that you can only delete your own private attribute tags from the catalog. Only **administrative users**[2] can delete any global and any private attribute tags.

Example: Deleting a Single Attribute Tag from the Cloud Subscriptions Tags Catalog

Request:

```
DELETE /subscriptions/cloud/tags/28eaf716-e66a-4518-a10a-542641b5b155
```

[1] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

[2] An administrative user is a Densify user in the Administrator user group or with the Analysis Admin role.

Timeline Tags

Description

This resource is used to return the timeline definition. It returns details of the four predictive timeframes using tags "7D", "30D", "60D" and "90D". This resource is useful when interpreting stats and metrics of the Control Environment, Infrastructure Group and Sensor objects.

Resource

```
/timeline-tags
```

Supported Operations

Table: Timeline Tags Supported Operations

Operation	HTTP Method	Input	Output	Description
Get Timeline Tags	GET /timeline-tags	None	Timeline Tags: Resource Elements on page 527	The timeline definition for the four predictive timeframes. Example: Getting the Timeline Tags on page 527

Resource Elements

Table: Timeline Tags Resource Elements

Element	Type	Description
<*tag*>	Complex, as specified in the Description	For each timeframe tag, provides the following as defined in **Administration > Timeline Definition**: - short_name—Short Name, with default "7", "30", "60", "90" - long_name—Long Name, with default "7 Days", "30 Days", "60 Days", "90 Days" - offset—Offset converted to days (offset from today's date), with default 7, 30, 60, 90 - order_index—Predictive Timeframe (i.e. 1..4 indicating the order of the timeframes) - date—UTC date - rounding—"RND_NM" to round up to the next month, "RND_NFQ" to the next quarter, "RND_NONE" for no rounding. The above information is defined for each timeframe tag <*tag*>, which are constants and defined as "7D", "30D", "60D" and "90D".

Examples

Example: Getting the Timeline Tags

The following example shows you how to get the timeline tags that are in use.

Example: Getting the Timeline Tags

Request:

```
GET /timeline-tags
```

Response:

```
{
  "90D": {
    "short_name": "24",
    "long_name": "24 Months",
    "offset": 720,
    "order_index": 4,
    "date": 1451538000000,
    "rounding": "RND_NM"
  },
  "30D": {
    "short_name": "3",
    "long_name": "3 Months",
    "offset": 90,
    "order_index": 2,
    "date": 1396238400000,
    "rounding": "RND_NM"
  },
  "60D": {
    "short_name": "12",
    "long_name": "12 Months",
    "offset": 360,
    "order_index": 3,
    "date": 1420002000000,
    "rounding": "RND_NM"
  },
  "7D": {
    "short_name": "1",
    "long_name": "1 Month",
    "offset": 30,
    "order_index": 1,
    "date": 1391144400000,
    "rounding": "RND_NM"
  }
}
```

Today Value Date

Description

This resource is used to schedule Workloads using the date/time of Densify (which is running the Route and Reserve Demand page).

Resource

```
/today-value
```

Supported Operations

Table: Today Value Supported Operations

Operation	HTTP Method	Input	Output	Description
Get Today	GET /today-value	None	utc_offset, today_setting, today_value, today_value_int, time_zone	The date/time of the machine running Densify. Example: Getting the Today Value on page 530

Resource Elements

Table: Today Value Resource Elements

Element	Type	Description
time_zone	string	The time zone of the machine running Densify.
today_setting	string	Specifies that Densify is using: - "CALENDAR"—the current date on the machine it is running (normal mode) - "SPECIFIED"—a static date (test mode)
today_value	string	The date in the format DD-MM-YYYY of the machine running Densify.
today_value_int	number	The UTC date/time of the machine running Densify.
utc_offset	string	The offset relative to GMT.

Examples

Example: Getting the Today Value

The following example shows you how to get the actual date value of "today":

Example: Getting the Today Value

Request:

```
GET /today-value
```

Response:

```
{
  "today_value_int": 1408939200000,
  "today_value": "10-02-2016",
  "today_setting": "CALENDAR",
  "time_zone": "Eastern Standard Time"
  "utc_offset": "-5.0",
}
```

User Preferences

Description

This resource object maintain the capacity unit settings and the Filter dropdown settings for users (as specified in the header of the API request), from login to login.

When performing a `/routing-requests/available-capacity-query`, you can use the capacity unit settings specified here and explicitly override any values defined by the `catalog_spec`.

With the Route and Reserve Demand page, this object allows you to define Capacity Unit settings once, found in the Options dropdown of the Hosting Venues pane. These settings are then automatically applied across all hosting venues when checking for available space.

Note that the filter options do not impact other resource objects, e.g. setting the filter to limit the hosting venues does not impact the request to `GET /infrastructure-groups/`.

Resource

```
/user-preferences
```

Supported Operations

Table: User Preferences Supported Operations

Operation	HTTP Method	Input	Output	Description
Get Individual	GET /user-preferences	None	User Preferences: Resource Elements on page 532	The capacity unit and filter settings for the user, as specified in the header. Example: Getting the User Preferences on page 533
Modify Individual	PUT /user-preferences	User Preferences: Resource Elements on page 532	User Preferences: Resource Elements on page 532	The capacity unit and filter settings can be modified for a specific user login. Example: Updating the Capacity Unit on page 534, Example: Updating the Hosting Venue Filter on page 535,

Resource Elements

Table: User Preferences Resource Elements

Element	Type	Modify	Description
Reservation Details Filter Preferences			
preferences	Complex, as specified in the Description	M	Provides the Reservation Details filter settings: - project—list of projects, separated by commas - os—list of OSs, separated by commas - type—list of booking types, separated by commas (e.g. "INBOUND_GUEST,OUTBOUND_GUEST")
preference_type	"booking_filter_guest"		The `preferences` element above applies to the Reservation Details filter settings.
Capacity Unit Preferences			
preferences	Complex, as specified in the Description	M	Provides the Capacity Unit settings. The settings and their defaults are as follows: - used_space=20 GB - provisioned_space=80 GB - show_available_space=false - memory=4 GB - detailed_eval_mode=false - workload_profile=Medium_Utilization - vcpu=2 The `provisioned_space` must be >= `used_space`.

Element	Type	Modify	Description
			The values you specify should correspond to the `catalog_spec` of your workloads.
preference_type	"capacity_unit"		The `preferences` element above applies to the Capacity Unit settings of the Options dropdown for the Hosting Venues pane.
Hosting Venue Filter Preferences			
preferences	Complex, as specified in the Description	M	Provides the filter settings for the Hosting Venues pane: • platform—list of platform types, separated by commas (e.g. "VMWARE")
preference_type	"infrastructure_group_filter"		The `preferences` element above applies to the Filters dropdown from the Hosting Venues pane.
Booking Requests To Be Routed Filter Preferences			
preferences	Complex, as specified in the Description	M	Provides the filter settings for the Booking Requests To Be Routed pane: • os—list of OSs, separated by commas • project—list of projects, separated by commas
preference_type	"workload_filter"		The `preferences` element above applies to the Filters dropdown from the Booking Requests To Be Routed pane.

Examples

Example: Getting the User Preferences

The following example shows you how to get the user preferences, for the user specified in the header (e.g. username: admin, password: admin).

Example: Getting the User Preferences

Request:

```
GET /user-preferences
```

Response:

```
[
  {
    "preferences": {
      "type": "OUTBOUND_GUEST"
    },
    "preference_type": "booking_filter_guest"
  },
  {
    "preferences": {
      "used_space": "20",
      "provisioned_space": "80",
      "show_available_space": "false",
      "memory": "4",
      "detailed_eval_mode": "true"
      "workload_profile": "Medium_Utilization",
      "vcpu": "2",
    },
    "preference_type": "capacity_unit"
  },
  {
    "preferences": {
      "platform": "IBM,VMWARE"
    },
    "preference_type": "infrastructure_group_filter"
  },
  {
    "preferences": {
      "os": "Linux,Windows"
    },
    "preference_type": "workload_filter"
  }
]
```

Example: Updating the Capacity Unit

The following example shows you how to update the `vcpu` to 4, for the user specified in the header. The `preference_type` must also be specified, as shown.

Example: Updating the User Preferences

Request:

```
PUT /user-preferences
{
  "preferences": {
    "vcpu": "4",
  },
  "preference_type": "capacity_unit"
}
```

Response:

```
[
  // ... *SNIP* of other preferences //
  {
    "preferences": {
      "used_space": "20",
      "provisioned_space": "80",
      "show_available_space": "false",
      "memory": "4",
      "detailed_eval_mode": "true"
      "workload_profile": "Medium_Utilization",
      "vcpu": "4",
    },
    "preference_type": "capacity_unit"
  },
  // ... *SNIP* of other preferences //
]
```

Example: Updating the Hosting Venue Filter

The following example updates the Filter dropdown for the Hosting Venues pane, for the user specified in the header.

Example: Updating the Hosting Venue Filter

Request:

```
PUT /user-preferences
{
  "preferences": {
    "platform": "VMWARE"
  },
  "preference_type": "infrastructure_group_filter"
}
```

Response:

```
[
  // ... *SNIP* of other preferences //
  {
    "preferences": {
      "platform": "VMWARE"
    },
    "preference_type": "infrastructure_group_filter"
  },
  // ... *SNIP* of other preferences //
]
```

Example: Resetting the Hosting Venue Filter

The following example resets the Filter dropdown of the Hosting Venue pane, for the user specified in the header.

Example: Resetting the Hosting Venue Filter

Request:

```
PUT /user-preferences
{
  "preferences": {},
  "preference_type": "infrastructure_group_filter"
}
```

Response:

```
[
  // ... *SNIP* of other preferences //
  {
    "preferences": {},
    "preference_type": "infrastructure_group_filter"
  },
  // ... *SNIP* of other preferences //
]
```

Workloads

Description

A Workload is a representation of a new workload being placed in the environment.

For every Workload object created, a corresponding Booking object is also created. Each Booking object is auto-created with the same `name` as that of its Workload.

The following table provides the Workload to Booking state mappings. For a complete overview of the states through create, modify and delete operations, see State Diagrams for Demand on page 1.

Table: Workload and Corresponding Booking States

Workload State	Booking State	Description
UNROUTED	DRAFT	Workload is defined but not included in a Routing Request (i.e. not routed).
ANALYZING	PENDING	Workload included in a Routing Request.
BOOKED	PENDING to COMMITTED to LATE to COMPLETED	When routed to a full control hosting venue with a future date, associated Booking is in PENDING state until it is included in the analytics through an environment refresh, when it is changed to COMMITTED. Changes to LATE when the incoming workload is late (i.e. `expected_date` arrives and `late_days` is > 0). The Booking becomes COMPLETED when the corresponding workload comes online and is auto-reconciled with the environment refresh. When routed to a guest-level hosting venue with an immediate or future date, associated Booking is in PENDING state until it is included in the analytics through an environment refresh, when it is changed to COMMITTED. Changes to LATE when the incoming

Workload State	Booking State	Description
		workload is late (i.e. `expected_date` arrives and `late_days` is > 0). The Booking becomes COMPLETED when the corresponding workload comes online and is auto-reconciled with the environment refresh. When routed to a non-control hosting venue, the associated Booking becomes COMMITTED (bypassing the PENDING state as the analytic refresh is not applicable). It then changes to LATE when the incoming workload is late (i.e. `expected_date` arrives and `late_days` is > 0). The Booking becomes COMPLETED when manually updated through a PUT request.
PLACED	(PENDING to) COMMITTED to LATE to COMPLETED	These states only apply when routed to a full control hosting venue. A workload is PLACED in the following cases: - workload routed with today's date - incoming workload arrives before extended expected date (including <n> days of grace defined by `late_days`) - a `GET` request is executed on an individual Workload in `BOOKED` status with `expected_date` of today (or `expected_date` has passed but Booking has not `EXPIRED`). Associated Booking is in PENDING state until it is included in the analytics through an environment refresh, when it changes to COMMITTED. Changes to LATE when incoming workload is late (i.e. `expected_date` arrives and `late_days` is > 0). Changes to COMPLETED only when the incoming workload is reconciled. See Placement and Option for Placement on page 537 for further details.
REJECTED	DRAFT	Could not route or reserve space.
	EXPIRED	No incoming workload by the extended expected date (including the <n> days of grace defined by `late_days`).
N/A	CANCELLED	Routing Request deleted.

When a Workload is defined, a catalog specification (element `catalog_spec`) is specified which acts as a reference system or template that resembles the Workload. Some elements from the catalog specification (like OS, CPU, memory and storage) are copied from that `catalog_spec` and set as elements of the Workload. When the Workload is created, some of these elements can be overwritten as part of the `POST`.

Placement and Option for Placement

When a Workload is PLACED (see PLACED on page 537 above and State Diagram—Create Scenario on page 1), the recommended host placement is the host with the most available capacity. For VMware environments, the recommended host placement is the host with the most available capacity.

Note that when a Workload is PLACED other than for immediate placement, the recommended host may not be the host with the most capacity at that time, as the placement is determined from the last analysis.

You can configure how host placement is determined, based on whether or not recommendations are being actioned. If the policy setting **Today Timeframe Uses** in category **Available Capacity & Performance Risks** is "'Existing Placements and Allocations", then the analytics without actions is used to determine host placement. Otherwise, the analytics with actions is used.

Along with host placement (element `host`), the best sensors for that host are determined (element `sensors`).

If no placement is found to satisfy both capacity and health status requirements, the next best placement for host and sensors within the same hosting venue is determined. This placement is a forced placement and the `status_reason` will be "Force to place".

You can extend the individual get request `GET /workloads/<id>` with the following option:

- `?recheckHost=true`—to recheck the recommended host and sensor placement for a Workload in PLACED state. If the recommended host/sensor is currently not healthy (i.e. real-time placement is enabled and the monitored host/sensor shows unhealthy), a new placement is provided. This option is used by the Route and Reserve Demand page when the user clicks on the **Validate Placement** button of the Guest Booking dialog box.

Resource

```
/workloads
```

Supported Operations

Table: Workload Supported Operations

Operation	HTTP Method	Input	Output	Description
Get Collection	GET /workloads	None	Workload Collection of [id, name, href]	Default Sort By is defined as: `?sort_by=expected_date` Filter-Metadata is supported. Note that it returns all status values of the collection, filtered or not, as if the collection was not filtered. Example: Rechecking Health Status for Placed Workloads on page 556
Get Individual	GET /workloads/<*id*>	None	Workloads: Resource Elements	Retrieve the Workload elements of the specified `id`. If the Workload is in `BOOKED` status and `expected_date` is today

Operation	HTTP Method	Input	Output	Description
			on page 541	(or passed), the Workload status will be updated to `REJECTED` (if the associated Booking has `EXPIRED`) or `PLACED` (with a recommended `host` placement). Example: Getting an Individual Workload on page 555
Create Individual	POST /workloads	Workloads: Resource Elements on page 541	Workloads: Resource Elements on page 541	A Workload can be created in one of two ways: ▪ using this `POST` request on Workloads ▪ using the `POST` request of its Routing Request, and specifying the same Workload elements as a `POST` on the Workload Example: Creating a Workload with Preferences on page 548, Example: Creating a Workload with Multiple Disks on page 552, Example: Creating a Workload with License Requirements on page 553
Create Multiple	POST /workloads	Workloads: Resource Elements on page 541 With `"num_copy" :` *<number>* specified	Workloads: Resource Elements on page 541	Similar to the Create Individual operation, but specifying the number of Workloads to create using `"num_copy" : <number>` with no limit to the number of instances created at a time. The names of the Workloads are auto-generated by appending a number after `name`. For example, if `name=VM`, then the generated names would be VM1, VM2, etc. Example: Creating Multiple Workloads on page 550
Modify Individual	PUT /workloads/<*id*>	Workloads: Resource Elements on page 541 Only elements marked "M" in the `Create/Mod` column can be modified	Workloads: Resource Elements on page 541	A Workload that is not in `PLACED` state can be modified through the `PUT` command. Only the `name` can be modified when the Workload is in the `ANALYZING` or `BOOKED` state. The `disks` > `pref_datastore` element can be modified when the Workload is in the `UNROUTED` state. If the `catalog_spec` is specified (even if it is not different), then other elements are defaulted as documented above unless explicitly overridden again. Example: Modifying a Workload on page 556, Example: Unsetting the Preferred Environment on page 557
Modify	PUT	workloadIds:	Workloads:	Workloads that are in `UNROUTED` or

Operation	HTTP Method	Input	Output	Description
Multiple	/workloads/multiple	[<*id*>, <*id*>,…], workload: {Workloads: Resource Elements on page 541} Only elements marked "M" in the Create/Mod column can be modified, except for `name` and `pref_datastore`	Resource Elements on page 541	`REJECTED` states can be modified by one `PUT` command. This API collectively updates all specified Workloads, or fails for all Workloads. If the `catalog_spec` is specified (even if it is not different), then other elements are defaulted as documented above unless explicitly overridden again. Example: Modifying Multiple Workloads on page 557, Example: Modifying Multiple Workloads and Their Attributes on page 558
Delete Individual Workload	DELETE /workloads/<*id*>	None	None	Only Workloads in `UNROUTED` state can be deleted. When the Workload is deleted, its associated Booking is also deleted. An error is thrown if the Workload is in any other state. For details when its owning Routing Request is deleted, see Routing Requests on page 347.
Delete Multiple Workloads	DELETE /workloads	workloadIds: [<*id*>, <*id*>,…]	None	Similar to deleting a single Workload above, however, this command deletes multiple Workloads in one call. Example: Deleting Multiple Workloads on page 559
Delete Individual Attributes	DELETE /workloads/<*id*>/attributes	`[id, value]` Where `value` is only required to delete the attribute value	None	If just the `id` of the attribute is specified, deletes all attribute id-name-value settings with that id. If `id` and `value` are specified, then deletes only that attribute with matching value. The `id` corresponds to an attribute `id` in the `attributes` array. This request is valid in any state. If the attribute `id` or `value` does not exist, then that specific delete is simply ignored. If any deleted item fails to delete, then no delete is performed at all.

Operation	HTTP Method	Input	Output	Description

				For example:
				`[`
				`   { "id": "attr_2", "value": "Capps"},`
				`   { "id": "state_power"}`
				`]`
				Example: Deleting an Attribute from a Specific Workload on page 558

Resource Elements

Table: Workload Resource Elements

Element	Type	Create/Mod-(Req)	Sort By	Filter	Description
id, name, href	strings	CM-R for `name`	S by `name`	F by `name`	See ID, Name and Self Reference (id, name, href) on page 29. The name is the expected system name of the incoming VM, required for auto-reconciliation. For details, see section *Auto-Reconciliation of Systems* of *Booking Overview* (Help Topic ID 230350).
attributes	[id, name, value]	CM		F only as that defined in cfg\bookings\bookings-config.xml	On a create or modify request, defines the Fit for Purpose attributes of the incoming system. If the attribute name or value is incorrectly specified, an error is returned and the Workload object is not created. For a create/modify, only the name-value pairs are required. The `id` is not required. For a single-valued attribute, if the name-value pair is defined more than once, then only the first occurrence is used and the second one is ignored. For a multi-valued attribute, the name-value pair can be specified multiple times as required for the same named attribute. Duplicates when specifying the same value more than once for the same

Element	Type	Create/Mod-(Req)	Sort By	Filter	Description

					named attribute are ignored.
					With the create request, some attributes are given default values. With the modify request, specifying a new attribute will be added, a single-valued attribute will be modified, and a multi-valued attribute will be appended with the new value.
					On a `GET` request, only those attributes that have values are returned.
					The attributes you can define are found in the cfg\bookings\bookings-config.xml. `id` is defined in that file and `name` corresponds to the actual field label. For example:
					`{`
					`  "id": "BOOKING_USED_`
					`DISKSPACE",`
					`  "name": "Used Space (MB)",`
					`  "value": "20480"`
					`}`
					To create or modify, only the name-value pairs are required:
					`{`
					`  "name": "Used Space (MB)",`
					`  "value": "20480"`
					`}`

					To filter on an attribute, use `attribute.`*id*. For example, to filter all Workloads that belong to Level 1 Security Zone:
					`/workloads/?attribute.attr_`
					`SecurityZone=Level 1`

booking	id, name, href				This is the link to the associated Booking.
catalog_spec	string	CM	S	F	Used to define the Catalog Reference Name, which corresponds to a specific Catalog Specification (see Catalog Specifications on page 243). When specified, this automatically sets other elements of the Workload, as documented herein. If not specified, this is defaulted to that defined by configuration seting API Default Catalog

Element	Type	Create/Mod-(Req)	Sort By	Filter	Description
					Specification (parameter key `rest.api.catalogSpec.default`). Must be a valid catalog name; otherwise, an error is returned.
catalog_spec_id	id			F	Used to return the ID of the `catalog_spec`.
control_environment	id, name, href, icon			F using `control_environment_id`	This is the link to the associated Control Environment where the workload was placed. Set if and only if the Workload has been routed (i.e. status is `PLACED` or `BOOKED`).
cpu_entitlement	number	CM	S	F	The eCPU. If not specified, defaulted to `vcpu` if `vcpu` is specified; otherwise, defaulted to that associated with `catalog_spec`.
creation_time	number		S	F	This is set to the date and time the Workload was created, in UTC.
description	string	CM			An arbitrary string that describes the workload. Kept in sync with the `comments` element defined by its associated Booking.
disks	[name, provisioned_space, used_space, pref_datastore, attributes[]]	CM `pref_datastore` cannot be updated in a modify multiple workloads operation		F only by `number_of_disks`	An array of disk requirements, with sizes and tier capabilities. One disk is defined by default. When modifying this array, you must specify all disks as the new array replaces the existing one. Defaulted to that of the associated `catalog_spec`, if defined. Each disk is defined as follows: - `name`—name of the disk - `provisioned_space`—provisioned space in MB - `used_space`—used space in MB - `attributes: [id, name, value]`—id, name and value of the datastore attributes. The `id` is mapped to its display name (e.g. "attr_DatastoreTier" is mapped to "Datastores") and can be determined by performing `GET /sensors`. An example of id, name and value: `{`

Element	Type	Create/Mod-(Req)	Sort By	Filter	Description

					`"id": "attr_DatastoreTier",`
					`"name": "Datastore Tier",`
					`"value": "Gold"`
					`}`
					- `pref_datastore`—preferred datastore for the disk (optional: only returned if set)
					See Assessing Datastore on page 348 for datastore placement details.

expected_date	number	CM	S	F using `expected_date_to`, `expected_date_from`	The date this workload will arrive in UTC. The time portion is ignored and always set to 04:00:00. If the Workload is created independently, using its own `POST` request, the `expected_date` is set by default to today. If the Workload is created within a `POST` request of a Routing Request, the `expected_date` is set by default to that of the Routing Request. When routing the request, the `expected_date` is taken from the Routing Request, and not in the Workload. For filtering workloads with arrival date before a specific date, use the `expected_date_to` option. To return all the workloads expected to arrive from a particular date, use the `expected_date_from` option. To return workloads with expected arrival dates within a range, use the `expected_date_from=<start_date>&expected_date_to=<end_date>` option.
host	string			F	The recommended host where this workload should be placed. Set if and only if the Workload has been routed (i.e. status is `BOOKED` with an analysis refresh or `PLACED`). Once defined, this `host` recommendation is subsequently

Element	Type	Create/Mod-(Req)	Sort By	Filter	Description
					updated, for the cases as described in State Diagrams for Demand on page 1. Note that this is not the actual host placement, but a recommendation.
infrastructure_group	id, name, href			F using `infrastructure_group_id`, `infrastructure_group`	This is the link to the associated Infrastructure Group where the workload was placed. Set, if and only if, the Workload has been routed (i.e. status is `PLACED` or `BOOKED`). Note that when filtering on Infrastructure Groups, you must use the element `infrastructure_group_id` with a UUID specified or element `infrastructure_group` with a name specified.
late_days	number				To allow for a late incoming workload, this is the number of days to hold the reservation after the `expected_date`. This element is defaulted to that defined by configuration setting **Default Number of Days to Hold a Booking Reservation after the Planned Start Date** (parameter key default.num.days.to.hold.booking.reservation) of category **25. Advanced - Booking Reservation Settings**. Specify 0 to define no grace period. For details on late bookings, see section *Late Bookings* of *Booking Overview* (Help Topic ID 230350).
memory	number	CM	S	F	The benchmark type Memory (score type Total Memory (MB)), defaulted to that of the associated `catalog_spec`.
number_days_to_expiry	number	not applicable			The number of remaining days before expiry (which is calculated as `expected_date` - today's date + `late_days`). This element is returned only when the associated Booking is "COMMITTED". This element is updated on every analysis refresh.
number_of_disks	number	not applicable	S	F	The number of disks defined in the `disks` array. This element is only used

Element	Type	Create/Mod-(Req)	Sort By	Filter	Description
					for filtering.
owner	string	CM	S	F	Used to define the owner or Customer Name of this Workload. If not set, this field is set to the user who is creating the Workload.
owner_email	string	CM		F	The email address of the `owner`. Kept in sync with the `requester_email` element defined by its associated Booking.
os	string		S	F	The Operating System Name, defined to that of the associated `catalog_spec`, e.g. "`Linux`".

pref_control_environment	id, name, href, icon	CM using `id`			A link to the Control Environment, where the Workload is preferred to be placed.
					To specify this element, use the format as shown in this example:
					`"pref_control_environment": {`
					`"id": "85772672-0388-4c34-939f-156f98b420bd"`
					`}`
					To unset this element, use:
					`"pref_control_environment": {`
					`"id": "__DELETE__"}`

pref_infrastructure_group	id, name, href	CM using `id`			A link to the preferred Infrastructure Group, where the Workload is preferred to be placed.
					To specify this element, use the format as shown in this example:
					`"pref_infrastructure_group": {`
					`"id": "9475b61b-5378-4a65-8e27-3cf488583f8f"`
					`}`
					To unset this element, use:
					`"pref_infrastructure_group": {`
					`"id": "__DELETE__"}`

profile_strength	number [-1..100]				This is set to the profile strength of the Workload request and is only set when the Workload status is `UNROUTED`. -1 means insufficient data.
project	string	CM		F	Used to define the Project. If not set, the Project is defined as "`__Unknown__`".

Element	Type	Create/Mod-(Req)	Sort By	Filter	Description
provisioned_space	number	not applicable	S	F	The sum of the provisioned space for the disks within the Workload (i.e. sum of the `provisioned_space` in MB of the `disk` array). Defaulted to that of the associated `catalog_spec`.
routing_request	href				A link to the Routing Request that is associated with this Workload.
sensors	[id, name, type, href, host_name]				The recommended Sensor object placement. Set if and only if the Workload has been routed (i.e. status is `PLACED` or `BOOKED`). Once defined, this `sensor` recommendation is subsequently updated, for the cases as described in State Diagrams for Demand on page 1. If there are no sensors, the empty list is returned. • `id`–sensor id • `name`–name of the sensor • `type`–sensor type (e.g. "datastore") • `href`–sensor href • `hostname`–display name of the sensor For Sensor element details, see Sensors including Datastores, Physical Storage, Resource Pools: Resource Elements on page 403.
status	string		S	F Note: `filter-metadata` returns `status_value` with all 5 possible states	The status of the Workload. See State Diagrams for Demand on page 1 for state details. If the Workload is associated with a Routing Request, this is the status of the Routing Request: • "`ANALYZING`" • "`PLACED`" • "`BOOKED`" • "`REJECTED`" If this Workload is not associated with any Routing Request, then the `status` is: • "`UNROUTED`"–initial state when

Element	Type	Create/Mod-(Req)	Sort By	Filter	Description
					Workload is not part of any Routing Request
status_reason	string			F Note: Not returned with the `filter-metadata` request, as this is free-form text.	Explanation of the status. This is populated when the `status` is `REJECTED` or forced to `PLACED` state, and takes on the same value as that of the Routing Request.
used_space	number	not applicable	S	F	The sum of the used space for the disks within the Workload (i.e. sum of the `used_space` in MB of the `disk` array). Defaulted to that of the associated `catalog_spec`.
vcpu	number	CM	S	F	The total vCPUs, defaulted to that of the associated `catalog_spec`. If specified, automatically updates `cpu_entitlement` to the same value if `cpu_entitlement` is not itself specified.
workload_profile	string	CM		F	Used to define the Workload Profile (see Workload Profiles on page 568). If not specified, this is defaulted to that defined by the `catalog_spec`. Must be a valid Workload Profile; otherwise, an error is returned. Note that if a UUID is returned, then this is a workload created from a transform analysis (i.e. Workoad Profile is a system name in the Booking Manager).

Examples

Example: Creating a Workload with Preferences

The following example creates a Workload in the preferred environment and hosting venue:

Example: Creating a Workload with Preferences

Request:

```
POST /workloads
```

```
{
  "name": "VM2045",
  "expected_date": 51395164801988,
  "num_copy": 1,
  "vcpu": 4,
  "memory": 16384,
  "catalog_spec": "win-large-16gb",
  "workload_profile": "Medium_Utilization",
  "pref_control_environment": {
    "id": "85772672-0388-4c34-939f-156f98b420bd"
  },
  "pref_infrastructure_group": {
    "id": "9475b61b-5378-4a65-8e27-3cf488583f8f"
  }
}
```

Response:

```
[
  {
    "id": "e95013d6-edec-4088-80e3-3c7998b2df77",
    "name": "VM2045",
    "status": "UNROUTED",
    "sensors": [],
    "workload_profile": "Medium_Utilization",
    "booking": {
      "id": "6992389f-056d-4725-b716-97050a9145bd",
      "name": "vm2045",
      "href": "/bookings/6992389f-056d-4725-b716-97050a9145bd"
    },
    "project": "__Unknown__",
    "owner": "admin",
    "attributes": [
      {
        "id": "attr_IPAddressesAssigned",
        "name": "IP Addresses Assigned",
        "value": "1"
      },
      {
        "id": "attr_Workload_Profile",
        "name": "Workload_Profile",
        "value": "Medium_Utilization"
      },
      {
        "id": "in_maint_mode",
        "name": "In Maintenance Mode",
        "value": "N/A"
      },
      {
        "id": "state_power",
        "name": "Power State",
        "value": "N/A"
      },
      {
        "id": "vmotion_enabled",
        "name": "VMotion Enabled",
        "value": "N/A"
      }
    ],
    "vcpu": 4,
    "memory": 16384,
    "os": "Windows",
    "description": "",
    "disks": [
      {
        "name": "SYSTEM",
        "attributes": [],
        "provisioned_space": 81920,
        "used_space": 20480
      }
    ],
    "href": "/workloads/e95013d6-edec-4088-80e3-3c7998b2df77",
    "expected_date": 51395140800000,
    "creation_time": 1421440815700,
    "catalog_spec": "win-large-16gb",
    "catalog_spec_id": "8d5bbbef-3a23-405d-862c-b93eacb49828",
    "pref_infrastructure_group": {
      "id": "9475b61b-5378-4a65-8e27-3cf488583f8f",
      "name": "Cluster 2",
      "href": "/infrastructure-groups/9475b61b-5378-4a65-8e27-3cf488583f8f"
    },
    "pref_control_environment": {
      "id": "85772672-0388-4c34-939f-156f98b420bd",
      "name": "Austin",
      "href": "/control-environments/85772672-0388-4c34-939f-156f98b420bd",
      "icon": "/control-environments/85772672-0388-4c34-939f-156f98b420bd/icon"
    },
    "cpu_entitlement": 4,
    "profile_strength": 60,
    "owner_email": "",
    "late_days": 14,
    "provisioned_space": 81920,
    "used_space": 20480,
  }
]
```

Example: Creating Multiple Workloads

This example shows you how to create multiple workloads:

Example: Creating Multiple Workloads

Request:

```
POST /workloads
{
  "name": "VM2043",
  "num_copy": 3
}
```

Response:

```
[
  {
    "id": "cea01bcd-79e8-4b4f-8d28-628c67af9ece",
    "name": "VM20431",
    "status": "UNROUTED",
    "sensors": [],
    "workload_profile": "Medium_Utilization",
    "booking": {
      "id": "b84ded12-e448-468a-bacd-5d11bad7d51d",
      "name": "vm20431",
      "href": "/bookings/b84ded12-e448-468a-bacd-5d11bad7d51d"
    },
    "project": "__Unknown__",
    "owner": "admin",
    "attributes": [
      {
        "id": "attr_IPAddressesAssigned",
        "name": "IP Addresses Assigned",
        "value": "1"
      },
      {
        "id": "attr_Workload_Profile",
        "name": "Workload_Profile",
        "value": "Medium_Utilization"
      }
    ],
    "vcpu": 4,
    "memory": 2048,
    "os": "Windows",
    "description": "",
    "disks": [
      {
        "name": "SYSTEM",
        "attributes": [],
        "provisioned_space": 81920,
        "used_space": 20480
      }
    ],
    "href": "/workloads/cea01bcd-79e8-4b4f-8d28-628c67af9ece",
    "expected_date": 1386738000000,
    "creation_time": 1386788744897,
    "catalog_spec": "win-large-8gb",
    "catalog_spec_id": "b93811c8-1f36-4f64-aec9-5abacc80df9f",
    "cpu_entitlement": 4,
    "profile_strength": 60,
    "owner_email": ""
    "late_days": 14,
    "provisioned_space": 81920,
    "used_space": 20480,
  },
  {
    "id": "2de9fd79-3b0f-41c5-a091-9b734ee5a862",
    "name": "VM20432",
    // ... *SNIP* of Workload 2 ...
  }
  {
    "id": "8eafd480-f2fc-44fd-8c1b-57d39e76bdf0",
    "name": "VM20433",
    // ... *SNIP* of Workload 3 ...
  }
]
```

Example: Creating a Workload with Multiple Disks

The following example creates a Workload with two disks:

Example: Creating a Workload with Multiple Disks

Request:

```
POST /workloads
{
  "name": "VM2045",
  "catalog_spec": "win-large-16gb",
  "workload_profile": "CPU_intensive",
  "disks": [
    {
      "name":"SYSTEM",
      "attributes":[
        {"id": "attr_DatastoreTier", "name":"Datastore Tier","value":"Silver"}
      ],
      "provisioned_space":20480,
      "used_space":5120,
      "pref_datastore":"nettstds01"
    },
    {
      "name":"DTD",
      "attributes":[
        {"id": "attr_DatastoreTier", "name":"Datastore Tier","value":"Gold"}
      ],
      "provisioned_space":20480,
      "used_space":5120
    }
  ]
}
```

Response:

```
[
  {
    "id": "e95013d6-edec-4088-80e3-3c7998b2df77",
    "name": "VM2045",
    "status": "UNROUTED",
    "sensors": [],
    "workload_profile": "CPU_intensive",
    "booking": {
      "id": "6992389f-056d-4725-b716-97050a9145bd",
      "name": "vm2045",
      "href": "/bookings/6992389f-056d-4725-b716-97050a9145bd"
    },
    "project": "__Unknown__",
    "owner": "admin",
    "attributes": [
      {
        "id": "attr_IPAddressesAssigned",
        "name": "IP Addresses Assigned",
        "value": "1"
      },
      {
        "id": "attr_Workload_Profile",
        "name": "Workload_Profile",
        "value": "CPU_intensive"
      }
    ],
    "vcpu": 4,
    "memory": 16384,
    "os": "Windows",
    "description": "",
    "disks": [
      {
        "name":"DTD",
        "attributes":[
          {"id": "attr_DatastoreTier", "name":"Datastore Tier","value":"Gold"}
        ],
        "provisioned_space":20480,
        "used_space":5120
      },
      {
        "name":"SYSTEM",
        "attributes":[
          {"id": "attr_DatastoreTier", "name":"Datastore 
Tier","value":"Silver"}
        ],
        "provisioned_space":20480,
        "used_space":5120,
        "pref_datastore":"nettstds01"
      }
    ]
    "href": "/workloads/e95013d6-edec-4088-80e3-3c7998b2df77",
    "expected_date": 1395115200000,
    "creation_time": 1387405841993,
    "catalog_spec": "win-large-16gb",
    "catalog_spec_id": "8d5bbbef-3a23-405d-862c-b93eacb49828",
    "cpu_entitlement": 4,
    "profile_strength": 60,
    "owner_email": "",
    "late_days": 14,
    "provisioned_space": 40960,
    "used_space": 10240
  }
]
```

Example: Creating a Workload with License Requirements

The following example creates a Workload that needs "Windows". To support multiple license requirements, please contact Densify Technical Services.

Example: Creating a Workload with License Requirements

Request:

```
POST /workloads
{
  "name": "VM1122",
  "catalog_spec": "win-large-16gb",
  "workload_profile": "CPU_intensive",
  "attributes": [
    {
      "id": "attr_LicenseGroup",
      "name": "License Group",
      "value": "Windows"
    }
  ]
}
```

Response:

```
[
  {
    "id": "2dac81bc-41fb-4e3f-a1ee-ca1688fdb075",
    "name": "VM1122",
    "status": "UNROUTED",
    "sensors": [],
    "workload_profile": "CPU_intensive",
    "booking": {
      "id": "9be9851c-2097-4cd3-badb-3a8455e42042",
      "name": "vm1122",
      "href": "/bookings/9be9851c-2097-4cd3-badb-3a8455e42042"
    },
    "project": "__Unknown__",
    "owner": "admin",
    "attributes": [
      {
        "id": "attr_IPAddressesAssigned",
        "name": "IP Addresses Assigned",
        "value": "1"
      },
      {
        "id": "attr_LicenseGroup",
        "name": "License Group",
        "value": "Windows"
      },
      {
        "id": "attr_Workload_Profile",
        "name": "Workload_Profile",
        "value": "CPU_intensive"
      }
    ],
    "vcpu": 4,
    "memory": 16384,
    "os": "Windows",
    "disks": [
      {
        "name": "SYSTEM",
        "attributes": [],
        "provisioned_space": 81920,
        "used_space": 20480
      }
    ]
    "href": "/workloads/2dac81bc-41fb-4e3f-a1ee-ca1688fdb075",
    "expected_date": 1417755600000,
    "creation_time": 1417814924210,
    "catalog_spec": "win-large-16gb",
    "catalog_spec_id": "8d5bbbef-3a23-405d-862c-b93eacb49828",
    "cpu_entitlement": 4,
    "profile_strength": 80,
    "owner_email": "",
    "late_days": 14,
    "provisioned_space": 81920,
    "used_space": 20480
  }
]
```

Example: Getting an Individual Workload

The following example shows you how to get a single workload:

Example: Getting an Individual Workload

Request:

```
GET /workloads/0768d3b5-b0cd-4723-bf90-7b970758b183
```

Response:

```
{
  "id": "0768d3b5-b0cd-4723-bf90-7b970758b183",
  "name": "vm_16_2013821144235",
  "status": "UNROUTED",
  "sensors": [],
  "workload_profile": "Low_Utilization",
  "booking": {
    "id": "f8170ab8-f094-462b-a8f6-d58c4ca3f340",
    "name": "vm_16_2013821144235",
    "href": "/bookings/f8170ab8-f094-462b-a8f6-d58c4ca3f340"
  },
  "project": "Kilimanjaro_2013",
  "owner": "admin",
  "attributes": [
    {
      "id": "attr_Workload_Profile",
      "name": "Workload_Profile",
      "value": "Low_Utilization"
    },
    // ... *SNIP* ...
  ],
  "vcpu": 4,
  "memory": 16384,
  "os": "Linux",
  "description": "",
  "disks": [
    {
      "name": "SYSTEM",
      "attributes": [
        {
          "id": "attr_DatastoreTier",
          "name": "Datastore Tier",
          "value": "Gold"
        }
      ],
      "provisioned_space": 15360,
      "used_space": 9216,
    },
    // ... *SNIP* of other disks ...
  ],
  "href": "/workloads/0768d3b5-b0cd-4723-bf90-7b970758b183",
  "expected_date": 1379649600000,
  "creation_time": 1377110555687,
  "catalog_spec": "lin-small-2gb",
  "catalog_spec_id": "2cb0102e-6d38-4188-b4b5-115e111a96ac",
  "pref_infrastructure_group": {
    "id": "adbea101-dab2-4253-9a86-690865fac4f7",
    "name":"Eng-Dev2",
    "href": "/infrastructure-groups/adbea101-dab2-4253-9a86-690865fac4f7"
  },
  "pref_control_environment": {
    "id": "7f8fbeaf-3b70-4560-bdbc-94c030a2184a",
    "name":"Boston",
    "href": "/control-environments/7f8fbeaf-3b70-4560-bdbc-94c030a2184a",
    "icon": "/control-environments/7f8fbeaf-3b70-4560-bdbc-94c030a2184a/icon"
  },
  "cpu_entitlement": 4,
  "profile_strength": 80,
  "owner_email": "",
  "late_days": 14,
  "provisioned_space": 46080,
  "used_space": 27648
}
```

Example: Rechecking Health Status for Placed Workloads

The following example rechecks the specified Workload placement and updates if unhealthy.

Example: Rechecking Health Status for Placed Workload

Request:

```
GET /workloads/76a0786a-c220-4fcb-a7e4-a5f233f7a41a/?recheckHost=true
```

Example: Getting Workloads with More than 5 Disks

The following example retrieves the details of Workloads that have more than 5 disks.

Example: Getting Workloads with More than 5 Disks

Request:

```
GET /workloads/?number_of_disks_from=5&details=true
```

Example: Modifying a Workload

The following example shows you how to modify a workload's expected system name and project name:

Example: Modifying a Workload

Request:

```
PUT /workloads/ac3132e6-d3b9-47b6-81e5-226d9cd01925
{
  "name": "win-phys-1422",
  "project": "8.0"
}
```

Response:

```
{
  "id": "ac3132e6-d3b9-47b6-81e5-226d9cd01925",
  "name": "win-phys-1422",
  // ... *SNIP* ...
  "project": "8.0"
  // ... *SNIP* ...
}
```

Example: Unsetting the Preferred Environment

The following example shows you how to unset a workload's preferred environment:

Example: Unsetting the Preferred Environment

Request:

```
PUT /workloads/0768d3b5-b0cd-4723-bf90-7b970758b183/
{
  "pref_control_environment": {
    "id": "__DELETE__"
  }
}
```

Example: Modifying Multiple Workloads

Example: Modifying Multiple Workloads

Request:

```
PUT /workloads/multiple
{
  "workloadIds": [
    "575300d8-3a5d-47ea-acfe-4d38c0af0d0e",
    "7589018a-2ed3-4d9d-a047-509439434ed0",
    "05e26393-fc6b-4160-86f4-c73c80b6389d"
  ],
  "workload": {
    "project": "8.0"
  }
}
```

Response:

```
[
  {
    "id": "575300d8-3a5d-47ea-acfe-4d38c0af0d0e ",
    "name": "vm1",
    // ... *SNIP* ...
    "project": "8.0"
    // ... *SNIP* ...
  },
  {
    "id": "7589018a-2ed3-4d9d-a047-509439434ed0",
    "name": "vm2",
    // ... *SNIP* ...
    "project": "8.0"
    // ... *SNIP* ...
  },
  {
    "id": "05e26393-fc6b-4160-86f4-c73c80b6389d",
    "name": "vm3",
    // ... *SNIP* ...
    "project": "8.0"
    // ... *SNIP* ...
  }
]
```

Example: Modifying Multiple Workloads and Their Attributes

Example: Modifying Multiple Workloads and Their Attributes

Request:

```
PUT /workloads/multiple
{
  "workloadIds": [
    "575300d8-3a5d-47ea-acfe-4d38c0af0d0e",
    "7589018a-2ed3-4d9d-a047-509439434ed0",
    "05e26393-fc6b-4160-86f4-c73c80b6389d"
  ],
  "workload": {
    "attributes": [
      {
        "name": "Department",
        "value": "Capps"
      },
      {
        "name": "Security Zone",
        "value": "Level 1"
      }
    ]
  }
}
```

Example: Deleting an Attribute from a Specific Workload

The following example deletes the attribute attr_ApplicationTier from the Workload.

Example: Deleting a Single-Valued Attribute from a Specific Workload

Request:

```
DELETE /workloads/0768d3b5-b0cd-4723-bf90-7b970758b183/attributes/
[
  {
    "id": "attr_ApplicationTier"
  }
]
```

For the multi-valued case, the first deletes a specific attribute name-value pair, while the second deletes the entire attribute and all its values from the Workload.

Example: Deleting a Multi-Valued Attribute from a Specific Workload

Request:

```
DELETE /workloads/0768d3b5-b0cd-4723-bf90-7b970758b183/attributes/
[
  {"id": "attr_multivalue", "value": "SQL"}
]
```

Request:

```
DELETE /workloads/0768d3b5-b0cd-4723-bf90-7b970758b183/attributes
[
  {"id": "attr_multivalue"}
]
```

Example: Deleting Multiple Workloads

The following example deletes three Workloads, as specified, in one call.

Example: Deleting Multiple Workloads

Request:

```
DELETE /workloads
{
  "workloadIds": [
    "575300d8-3a5d-47ea-acfe-4d38c0af0d0e",
    "7589018a-2ed3-4d9d-a047-509439434ed0",
    "05e26393-fc6b-4160-86f4-c73c80b6389d"
  ]
}
```

Workloads Attribute Metadata Groups

Description

The attribute metadata for the Workload resource object can be retrieved in a single `GET` request. This metadata can be used to display and validate the attributes of the Workload resource object before the Workload object is created. It provides information such as whether or not the attribute is displayed, possible valid values, whether or not a value must be specified, etc. The returned list is by category, with attributes within each category.

Resource

```
/workloads/attribute-metadata
```

Supported Operations

Table: Workload Attribute Metadata Supported Operations

Operation	HTTP Method	Input	Output	Description
Get Collection	GET /workloads/attribute-metadata	None	Workload Attribute	The details of attributes defined by all Workloads is returned, sorted by attribute name.

Operation	HTTP Method	Input	Output	Description
			Metadata Collection of [Workloads Attribute Metadata Groups: Resource Elements on page 561]	
Get Individual	GET /workloads/attribute-metadata/<*name*>	None	Workloads Attribute Metadata Groups: Resource Elements on page 561	The metadata details of the specified attribute is returned. Example: Attribute Metadata on page 562

Resource Elements

Table: Workload Attribute Metadata Resource Elements

Element	Type	Filter	Description
id, name	strings	F by `id`, `name`	See ID, Name and Self Reference (id, name, href) on page 29. The ID and name of the attribute category. Each category corresponds to a requirement when defining a booking request through the Route and Reserve Demand page.
attributes	[Complex, as specified in the Description]	F by each element within the array, except `values`	The metadata for each attribute in this category, returning: • `id`—ID of the attribute • `name`—Name of the attribute • `overridable`—whether the attribute can be modified or not; possible values include {true, false} • `displayable`—whether the attribute is displayed or not; possible values include {true, false} • `values`—array of possible values; only valid when the parameter-type is enum • `multiple`—whether multiple values may be specified on the create/modify request or not; possible values include {true, false} • `category-id`—ID of the category (same as id above) • `parameter-type`—whether the attribute is enumerated or freeform; possible values include {enum, freeform} • `required`—whether a value must be specified on the create

Element	Type	Filter	Description
			request or not; possible values include {true, false}
description	string	F	The description of the attribute category.

Examples

Example: Attribute Metadata

The following example shows you how to get attribute metadata:

Example: Getting Workload Attribute Metadata

Request:

```
GET /workloads/attribute-metadata/Software Requirements
```

Response:

```
{
  "id": "software-requirements",
  "name": "Software Requirements",
  "description": "Software Requirements",
  "attributes": [
    {
      "id": "attr_LicenseGroup",
      "name": "License Group",
      "overridable": false,
      "displayable": true,
      "values": [
        "Linux",
        "Oracle",
        "SQL",
        "UNIX",
        "Windows"
      ],
      "multiple": false,
      "category-id": "software-requirements",
      "parameter-type": "enum",
      "required": false
    }
  ]
}
```

Workloads Project, Owner, Status Groups

Description

Workloads can be grouped by `project`, `owner` or `status` so that they can be ordered in a specific priority. To obtain a collection of one of these groups, simply append the group name (i.e. `projects`, `owners` or `status`) to your Workload collection request (e.g. by project, use `/workloads/projects`). If the project or owner element has not been defined, then "`__Unknown__`" can be used to query with no project/owner name. The collection by group is returned in alphabetical order by name, with "`__Unknown__`" objects at the end.

When performing a collection by group, the following options are supported:

- Collection Details—to obtain the Workload details for each group object.
- Paging—to page the group collection. Note that the Workloads are not part of the paging size.
- Filters—to filter the Workloads using the filtering elements of the Workload. If there are qualifying Workloads, the collection returns the group and their qualifying Workloads (if `&details=true`). Otherwise, if there are no qualifying Workloads for a given group, the group is not returned.

Resource

```
/workloads/projects
/workloads/owners
```

```
/workloads/status
```

Supported Operations

Table: Project/Owner/Status Supported Operations

Operation	HTTP Method	Input	Output	Description
Get Collection	GET /workloads/projects	None	Workload Project collection of [name, href]	The list of projects defined by all Workloads is returned, sorted by project name with "__ Unknown__" projects last. The Workload default Sort By is defined as: ?sort_by=name. Note that any Workload element that can be used for sorting Workload collections can be used when sorting group collection queries. The resulting collection of Workloads is sorted as specified within each group.
	GET /workloads/owners	None	Workload Owner collection of [name, href]	The list of owners defined by all Workloads is returned, sorted by owner name with "__ Unknown__" owners last. The Workload default Sort By is defined as: ?sort_by=name. Note that any Workload element that can be used for sorting Workload collections can be used when sorting group collection queries. The resulting collection of Workloads is sorted as specified within each group.
	GET /workloads/status	None	Workload Status collection of [name, href]	The list of status values defined by all Workloads is returned, sorted by status name. The Workload default Sort By is defined as: ?sort_by=name. Note that any Workload element that can be used for sorting Workload collections can be used when sorting group collection queries. The resulting collection of Workloads is sorted as specified within each group.
Get Individual	GET /workloads/projects/<*name*>	None	name, href, [Workloads: Resource Elements on page 541]	The details of all Workloads defined with the specified project name <*name*> (case insensitive).
	GET /workloads/owners/<*name*>	None	name, href, [Workloads:	The details of all Workloads defined with the specified owner name <*name*> (case insensitive).

Operation	HTTP Method	Input	Output	Description
			Resource Elements on page 541]	
	GET /workloads/status/<*name*>	None	name, href, [Workloads: Resource Elements on page 541]	The details of all Workloads that are in the specified status <*name*> (case insensitive).

Resource Elements

Table: Workloads Project/Owner/Status Resource Elements

Element	Type	Filter	Description
name	string		Name of the Project/Owner/Status.
href	string		Link to the Project/Owner/Status.
workloads	[Workloads: Resource Elements on page 541]	F by Workload elements that support filtering (see Workloads: Resource Elements on page 541)	Array of Workloads grouped by the Project/Owner/Status. The elements that are returned are the same as those returned when performing a `GET` request on a Workload.

Examples

Example: Getting a Collection of Workload Projects

Example: Getting a Collection of Workload Projects

The following example shows you how to obtain the collection of current workload projects:

Request:

```
GET /workloads/projects
```

Response:

```
[
  {
    "name": "Kilimanjaro 2013",
    "href": "/workloads/projects/Kilimanjaro 2013"
  },
  {
    "name": "Project X",
    "href": "/workloads/projects/Project X"
  }
]
```

Example: Getting a Collection of UNROUTED Workload Projects

This example retrieves the list of Projects that have Workloads with `status=UNROUTED`.

Example: Getting a Collection of UNROUTED Workload Projects

Request:

```
GET /workloads/projects/?status=UNROUTED
```

Response:

```
[
  {
    "name": "Kilimanjaro_2013",
    "href": "/workloads/projects/Kilimanjaro_2013?status=UNROUTED"
  },
  {
    "name": "__Unknown__",
    "href": "/workloads/projects/__Unknown__?status=UNROUTED"
  }
]
```

Example: Getting a Collection of UNROUTED Workload Projects with Details

This example retrieves the list of Workloads, with details, with `status=UNROUTED,` grouped by Project.

Example: Getting a Collection of UNROUTED Workload Projects with Details

Request:

```
GET /workloads/projects/?details=true&status=UNROUTED
```

Response:

```
[
  {
    "name": "Kilimanjaro_2013",
    "href": "/workloads/projects/Kilimanjaro_2013?status=UNROUTED",
    "workloads": [
      {
        "id": "0768d3b5-b0cd-4723-bf90-7b970758b183",
        "name": "vm_16_2013821144235",
        "status": "UNROUTED",
        // ... *SNIP* of Workload ...
      },
      {
        "id": "08ec6623-835d-4c00-8655-3ecaf2f6f987",
        "name": "vm_4_2013821155340",
        "status": "UNROUTED",
        // ... *SNIP* of Workload ...
      },
      // ... *SNIP* of Workloads...
    ]
  },
  {
    "name": "__Unknown__",
    "href": "/workloads/projects/__Unknown__?status=UNROUTED"
    "workloads": [
    // ... *SNIP* of Workloads...
    ]
  }
]
```

Example: Getting a Collection of Workloads Owned by admin and using Linux

This example retrieves the list of Workloads, with details, with `os=Linux` grouped by Owner admin.

Example: Getting a Collection of Workloads Owned by admin and using Linux

Request:

```
GET /workloads/owners/admin/?os=Linux
```

Response:

```
{
  "name": "admin",
  "href": "/workloads/owners/admin",
  "workloads": [
    {
      "id": "e4a4f4b7-3a96-4778-b8d2-eb286d3de253",
      "name": "vm1",
      // ... *SNIP* of Workload ...
    },
    // ... *SNIP* of Workloads...
  ]
}
```

Workload Profiles

Description

A Workload Profile is a model of the workload behavior of a machine for various workload types.

This resource object is used to retrieve the Workload Profiles and associated representative utilization values/charts. It also includes the representative workloads created from transform scenarios (i.e. see `/workload-profiles/Transform`).

When you create a Workload, you specify the Workload Profile or the expected workload behavior using element `workload_profile`.

Resource

```
/workload-profiles
```

Supported Operations

Table: Workload Profile Supported Operations

Operation	HTTP Method	Input	Output	Description
Get Collection	GET /workload-profiles	None	Workload Profile Type collection of [name, href]	Returns the Representative Workload types. Collection Details, Sort By and Filters are not supported.

Operation	HTTP Method	Input	Output	Description
				Example: Getting a Collection of Workload Profile Types on page 570
Get Collection	GET /workload-profiles/<*workload-profile type*>	None	Workload Profile collection of [name, id, href]	Returns the Workload Profiles for the specified type. Collection Details is supported. Sort By and Filters are not supported. Example: Getting a Collection of Workload Profiles of Type Booking on page 570
Get Individual	GET /workload-profiles/<*workload-profile type*>/<*workload-profile name*>	None	Workload Profiles: Resource Elements on page 569 (A)	Returns the utilizations for the specified Workload Profile. Example: Getting Workload Types for an Individual Booking Workload Profile on page 571
Get Individual	GET /workload-profiles/<*workload-profile type*>/<*workload-profile name*>/<*utilization*>/values	None	Workload Profiles: Resource Elements on page 569 (B)	Returns the values for the specified utilization for the specified Workload Profile. Example: Getting an Individual Workload Type for an Individual Booking Workload Profile on page 572

Resource Elements

Table: Workload Profile Resource Elements (A)

Element	Type	Description
name, id, href	strings	See ID, Name and Self Reference (id, name, href) on page 29.
utilization_values	[name, values, image]	The utilization types for the Workload Profile: ▫ name–name of the workload type ▫ values–link to return the values ▫ image–link to return the graphical chart of the values

Table: Workload Profile Utilization Resource Elements (B)

Element	Type	Description
name, href	strings	See ID, Name and Self Reference (id, name, href) on page 29.

Element	Type	Description
values	hour [key, value]	The representative workload of the utilization in hourly quartiles. The following is returned: - `key`–represents the hour, from 1..24 - `value`–value for that hour

Examples

Example: Getting a Collection of Workload Profile Types

The following example shows you how to obtain the collection of workload profile types:

Example: Getting a Collection of Workload Profile Types

Request:

```
GET /workload-profiles
```

Response:

```
[
  {
    "name": "Control_SPPNA_30d",
    "href": "/workload-profiles/Control_SPPNA_30d"
  },
  {
    "name": "Blank_Workload",
    "href": "/workload-profiles/Blank_Workload"
  },
  {
    "name": "Booking",
    "href": "/workload-profiles/Booking
  }, {
  // ... *SNIP* ...
]
```

Example: Getting a Collection of Workload Profiles of Type Booking

The following example shows you how to obtain the collection of workload profiles for bookings:

Example: Getting a Collection of Workload Profiles of Type Booking

Request:

```
GET /workload-profiles/Booking
```

Response:

```
[
  {
    "name": "Low_Utilization",
    "id": "Low_Utilization",
    "href": "/workload-profiles/Booking/Low_Utilization"
  },
  {
    "name": "Medium_Utilization",
    "id": "Medium_Utilization",
    "href": "/workload-profiles/Booking/Medium_Utilization"
  },
  {
    "name": "Memory_Intensive",
    "id": "Memory_Intensive",
    "href": "/workload-profiles/Booking/Memory_Intensive"
  },
  {
    "name": "cluster-avg-system-(vc51v -cluster 4)",
    "id": "48232385-45d5-4f51-8543-b05a99bc773c",
    "href": "/workload-profiles/Booking/48232385-45d5-4f51-8543-b05a99bc773c"
  },
  // ... *SNIP* of Booking Workload Profiles...
]
```

Example: Getting Workload Types for an Individual Booking Workload Profile

The following example shows you how to get a collection of workload types for a single bookings workload profile:

Example: Getting Workload Types for an Individual Booking Workload Profile

Request:

```
GET /workload-profiles/Booking/Low_Utilization
```

Response:

```
[
  {
    "name": "Low_Utilization",
    "id": "Low_Utilization",
    "href": "/workload-profiles/Low_Utilization",
    "utilization_values": [
      {
        "name": "Actual_Mem_Utilization_As_Pct",
        "values": "/workload-profiles/Booking/Low_Utilization/Actual_Mem_
Utilization_As_Pct/values",
        "image": "/workload-profiles/Booking/Low_Utilization/Actual_Mem_
Utilization_As_Pct/image"
      },
      {
        "name": "CPU_Utilization",
        "values": "/workload-profiles/Booking/Low_Utilization/CPU_
Utilization/values",
        "image": "/workload-profiles/Booking/Low_Utilization/CPU_
Utilization/image"
      },
      // ... *SNIP* of utilization_values ...
    ]
  }
]
```

Example: Getting an Individual Workload Type for an Individual Booking Workload Profile

Request:

```
GET /workload-profiles/Booking/Low_Utilization/CPU_Utilization/values
```

Response:

```
{
  "name": "CPU_Utilization",
  "href": "/workload-profiles/booking/Low_Utilization/CPU_Utilization/values",
  "values": {
    "hour": [
      {
        "key": 1,
        "value": "[1.0, 2.0, 2.0, 20.0, 3.0]"
      },
      {
        "key": 2,
        "value": "[1.0, 2.0, 2.0, 20.0, 3.0]"
      },
      // ... *SNIP* ...
      {
        "key": 24,
        "value": "[1.0, 2.0, 2.0, 20.0, 3.0]"
      }
    ]
  }
}
```

Index

Densify

200 – 120 East Beaver Creek Rd
Richmond Hill, Ontario
Canada, L4B 4V1
www.Densify.com
Telephone: +1.905.731.0090
Telephone (toll free): +1.866.731.0090
Fax: +1. 289.650.1811