# Table of Contents

**Preface** ................................................................................................................................. 4  
About Qualys .................................................................................................................................. 4  
Contact Qualys Support ............................................................................................................ 4  

**Chapter 1 - Welcome** ............................................................................................................. 5  
Qualys API Framework .............................................................................................................. 5  
Qualys API URL .......................................................................................................................... 6  
API Conventions ........................................................................................................................ 7  

**Chapter 2 - Version 2 APIs** .................................................................................................. 8  
Fetch Authentication Token ....................................................................................................... 9  
Fetch List of Supported Technologies ....................................................................................... 11  
Provision an Asset ..................................................................................................................... 15  
Fetch Asset Status using UUID .................................................................................................. 17  
Fetch Supported Commands for a Technology ........................................................................ 18  
Fetch Supported Commands based on Asset UUID ................................................................. 19  
Upload Command Output for a UUID ....................................................................................... 20  
Revoke an Asset using UUID .................................................................................................... 21  
Provision Assets in Bulk ........................................................................................................... 22  
Revoke Assets in Bulk ............................................................................................................... 25  
Re-Provision Assets in Bulk ....................................................................................................... 26
Preface

This guide is intended for application developers who will use the Qualys Out-of-band Configuration Assessment API.

About Qualys

Qualys, Inc. (NASDAQ: QLYS) is a pioneer and leading provider of cloud-based security and compliance solutions. The Qualys Cloud Platform and its integrated apps help businesses simplify security operations and lower the cost of compliance by delivering critical security intelligence on demand and automating the full spectrum of auditing, compliance and protection for IT systems and web applications.

Founded in 1999, Qualys has established strategic partnerships with leading managed service providers and consulting organizations including Accenture, BT, Cognizant Technology Solutions, Deutsche Telekom, Fujitsu, HCL, HP Enterprise, IBM, Infosys, NTT, Optiv, SecureWorks, Tata Communications, Verizon and Wipro. The company is also a founding member of the Cloud Security Alliance (CSA). For more information, please visit www.qualys.com.

Contact Qualys Support

Qualys is committed to providing you with the most thorough support. Through online documentation, telephone help, and direct email support, Qualys ensures that your questions will be answered in the fastest time possible. We support you 7 days a week, 24 hours a day. Access support information at www.qualys.com/support/.
Chapter 1 - Welcome

Welcome to Out-of-band Configuration Assessment API guide.

Get Started
Qualys API Framework - Learn the basics about making API requests. The base URL depends on the platform where your Qualys account is located.

API Conventions - Get tips on using the Curl command-line tool to make API requests.

Get API Notifications
Subscribe to our API Notifications RSS Feeds for announcements and latest news.

Qualys API Framework

The Qualys Out-of-band Configuration Assessment API uses the following framework.

Request URL

The URL for making API requests respects the following structure:

https://<baseurl>/<module>/<object>/<object_id>/<operation>

where the components are described below.

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;baseurl&gt;</td>
<td>The Qualys API gateway URL that you should use for API requests depends on the platform where your account is located. The gateway URL for Qualys US Platform 1 is: <a href="https://gateway.qg1.apps.qualys.com">https://gateway.qg1.apps.qualys.com</a></td>
</tr>
<tr>
<td>&lt;module&gt;</td>
<td>The API module. For the OCA API, the module is: &quot;oca&quot;.</td>
</tr>
<tr>
<td>&lt;object&gt;</td>
<td>The module specific object.</td>
</tr>
<tr>
<td>&lt;object_id&gt;</td>
<td>(Optional) The module specific object ID, if appropriate.</td>
</tr>
<tr>
<td>&lt;operation&gt;</td>
<td>The request operation, such as provisioning an asset.</td>
</tr>
</tbody>
</table>
Qualys API URL

The Qualys API URL you should use for API requests depends on the Qualys platform where your account is located.

Click here to identify your Qualys platform and get the API URL

This documentation uses the API gateway URL for Qualys US Platform 1 (https://gateway.qg1.apps.qualys.com) in sample API requests. If you’re on another platform, please replace this URL with the appropriate gateway URL for your account.
**API Conventions**

**Using Curl**

**Curl** is a multi-platform command-line tool used to transfer data using multiple protocols. This tool is supported on many systems, including Windows, Unix, Linux and Mac. In this document Curl is used in the examples to build Qualys API requests using the HTTP over SSL (https) protocol, which is required.

Want to learn more? Visit [https://curl.haxx.se/](https://curl.haxx.se/)

The following Curl options are used according to different situations:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-X GET/POST/DELETE</td>
<td>The GET, POST, DELETE method are used as per requirement.</td>
</tr>
<tr>
<td>-H 'Authorization: Bearer &lt;token&gt;'</td>
<td>This option is used to provide a custom HTTP request header parameter for authentication. Provides the JSON Web Token (JWT) received from Qualys authentication API in the following format: Authorization: Bearer &lt;token&gt;</td>
</tr>
<tr>
<td>-H 'content-type: application/json'</td>
<td>Denotes that content is in JSON format.</td>
</tr>
<tr>
<td>-H 'Content-Type: text/plain'</td>
<td>Denotes that content is in text or plain format.</td>
</tr>
<tr>
<td>-d @request.json</td>
<td>Provide a request.json file for parameter input.</td>
</tr>
<tr>
<td>--data-urlencode</td>
<td>Used to encode spaces and special characters in the URL/Parameter values.</td>
</tr>
</tbody>
</table>

The sample below shows a typical Curl request using options mentioned above and how they interact with each other.

```bash
curl -X GET 'http://<api_gateway_url>/ocaapi/v2.0/technology/PolicyCompliance' -H 'Content-Type: application/json' -H 'Authorization: Bearer <token>'
```
Chapter 2 - Version 2 APIs

Note: The following APIs use token-based authentication.

Fetch Authentication Token

OCA APIs
Fetch List of Supported Technologies
Provision an Asset
Fetch Asset Status using UUID
Fetch Supported Commands for a Technology
Fetch Supported Commands based on Asset UUID
Upload Command Output for a UUID
Revoke an Asset using UUID
Provision Assets in Bulk
Revoke Assets in Bulk
Re-Provision Assets in Bulk
Fetch Authentication Token

/auth

[POST]

You must authenticate to the Qualys Cloud Platform using Qualys account credentials (username and password) and get the JSON Web Token (JWT) before you can start using the OCA APIs.

The Authentication API returns a JSON Web Token (JWT) which you can use for authentication during OCA API calls. The token expires in 4 hours. You must regenerate the token to continue using the OCA API.

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>username</td>
<td>The username of the Qualys user account for which you want to fetch OCA data.</td>
</tr>
<tr>
<td>password</td>
<td>The password of the Qualys user account for which you want to fetch OCA data.</td>
</tr>
<tr>
<td>token</td>
<td>The value should be set to True, to get the generated token.</td>
</tr>
<tr>
<td>ContentType</td>
<td>Should be &quot;application/x-www-form-urlencoded&quot;</td>
</tr>
<tr>
<td>gateway.qg1.apps.qualys.com</td>
<td>The API gateway URL to the Qualys API server where your account is located.</td>
</tr>
</tbody>
</table>

Sample

Request:

curl -X POST 'http://<api_gateway_url>/auth' -d 'username=username&password=password&token=true' -H 'Content-Type: application/x-www-form-urlencoded'

Response:

yJhbGciOiJIUzUxMjQiLCJhbGciOiQxMjQiLCJpZ extravZlwIjOiJzdWI0OTMzIiwidXNlcnR5IjoxfQ

username
password
token
ContentType
gateway.qg1.apps.qualys.com

The username of the Qualys user account for which you want to fetch OCA data.
The password of the Qualys user account for which you want to fetch OCA data.
The value should be set to True, to get the generated token.
Should be "application/x-www-form-urlencoded"
The API gateway URL to the Qualys API server where your account is located.
Chapter 2 - Version 2 APIs
Fetch Authentication Token

mFldGhlbnRpY2F0aW9uTWV0aG9kIjoiTGV0aEhhbmRsZXIIiLCJjdXN0d2lkIjo
xNDAzMi5iLCJzZXNzaWduXJhG1ubiI6IjIwIiwicHlvdXNlcGltIjoiI0YzhhMmYzNi1hZGN1LTU3Ny1hZGN1LTU3Ny1hZGN1LTU3Ny1hZGN1LTU3Ny1hZGN1LTU3Ny1hZ
mYzNi1hZGN1LTU3Ny1hZGN1LTU3Ny1hZGN1LTU3Ny1hZGN1LTU3Ny1hZGN1LTU3Ny1hZ
mYzNi1hZGN1LTU3Ny1hZGN1LTU3Ny1hZGN1LTU3Ny1hZ

HBpcmVkIjoidHJ1ZSI6InN1YnJcm1wdG1vbklkIjoxODYwOTgzLCJ1eHAiOiJ0Li4k
4OTYwMjRsImZhbCI6IzNTU4OTg4MTgyMS5wImlhdCI6MTU4OTg4MTgyMSwiaXNzZW
k1COENYT0ttamZHSENNelZHSe0KdXpVMFiw8Fm9KSjBKSC1obDRCdXB

saHfYRSLxYXmwwMSK9.xbJrVhgdxj8A0LkiPMRWNcGiuqtI954ccpNBx1CvZVEcvIFU
J43kw81KZIoL8Mm8qrq0fI1DjhIzD283gC1ZWw

.
Fetch List of Supported Technologies
/ocaapi/v2.0/technology/PolicyCompliance

[GET]
To get a list of supported technologies.

Header Parameters

| authorization | (Required) The token that was generated using the Fetch Authentication Token API. |

Sample

Request:
curl -X GET
'http://<api_gateway_url>/ocaapi/v2.0/technology/PolicyCompliance'
-H 'Content-Type: application/json' -H 'Authorization: Bearer <token>'

Response:
{
  "code": 200,
  "data": [
    {
      "technology": "ACME Packet OS",
      "createdAt": "2019-10-11T06:10:46.000+0000",
      "updatedAt": "2019-10-11T06:10:46.000+0000"
    },
    {
      "technology": "ArubaOS 6",
      "createdAt": "2019-10-11T06:10:46.000+0000",
      "updatedAt": "2019-10-11T06:10:46.000+0000"
    },
    {
      "technology": "Cisco ACS 5",
      "createdAt": "2019-10-11T06:10:46.000+0000",
      "updatedAt": "2019-10-11T06:10:46.000+0000"
    },
    {
      "technology": "Cisco FTD 6",
      "createdAt": "2019-10-11T06:10:46.000+0000",
      "updatedAt": "2019-10-11T06:10:46.000+0000"
    },
    {
      "technology": "Cisco UCS Manager 2",
      "createdAt": "2019-10-11T06:10:46.000+0000",
      "updatedAt": "2019-10-11T06:10:46.000+0000"
    }
  ]
}
"createdAt": "2019-10-11T06:10:46.000+0000",
"updatedAt": "2019-10-11T06:10:46.000+0000"
},
{
  "technology": "Cisco WLC 8",
  "createdAt": "2019-10-11T06:10:45.000+0000",
  "updatedAt": "2019-10-11T06:10:45.000+0000"
},
{
  "technology": "Comware 5",
  "createdAt": "2019-10-11T06:10:45.000+0000",
  "updatedAt": "2019-10-11T06:10:45.000+0000"
},
{
  "technology": "Comware 7",
  "createdAt": "2019-10-11T06:10:45.000+0000",
  "updatedAt": "2019-10-11T06:10:45.000+0000"
},
{
  "technology": "Data Domain OS 5",
  "createdAt": "2019-10-11T06:10:46.000+0000",
  "updatedAt": "2019-10-11T06:10:46.000+0000"
},
{
  "technology": "Fabric 7",
  "createdAt": "2019-10-11T06:10:45.000+0000",
  "updatedAt": "2019-11-21T12:54:27.000+0000"
},
{
  "technology": "Fabric 8",
  "createdAt": "2019-10-11T06:10:45.000+0000",
  "updatedAt": "2019-11-21T12:54:20.000+0000"
},
{
  "technology": "FireEye CMS 7",
  "createdAt": "2019-10-11T06:10:45.000+0000",
  "updatedAt": "2019-10-11T06:10:46.000+0000"
},
{
  "technology": "FireEye CMS 8",
  "createdAt": "2019-10-11T06:10:46.000+0000",
  "updatedAt": "2019-10-11T06:10:46.000+0000"
},
{
  "technology": "HP Printers",
  "createdAt": "2019-10-11T06:10:46.000+0000",
  "updatedAt": "2019-10-11T06:10:46.000+0000"}
"createdAt": "2020-04-14T21:20:14.000+0000",
"updatedAt": "2020-04-14T21:20:14.000+0000"
},
{
    "technology": "HP Safeguard",
    "createdAt": "2019-10-11T06:10:46.000+0000",
    "updatedAt": "2019-10-11T06:10:46.000+0000"
},
{
    "technology": "HPE 3Par OS 3",
    "createdAt": "2019-10-11T06:10:45.000+0000",
    "updatedAt": "2019-11-21T14:23:30.000+0000"
},
{
    "technology": "IBM z/OS Security Server RACF 2",
    "createdAt": "2020-03-17T17:10:53.000+0000",
    "updatedAt": "2020-03-17T17:10:53.000+0000"
},
{
    "technology": "Imperva WebApplication Firewall",
    "createdAt": "2019-10-11T06:10:46.000+0000",
    "updatedAt": "2019-10-11T06:10:46.000+0000"
},
{
    "technology": "Juniper IVE 8",
    "createdAt": "2019-10-11T06:10:46.000+0000",
    "updatedAt": "2019-10-11T06:10:46.000+0000"
},
{
    "technology": "Riverbed SteelHead RiOS 9",
    "createdAt": "2019-11-21T14:22:22.000+0000",
    "updatedAt": "2019-11-21T14:22:22.000+0000"
},
{
    "technology": "Samsung Printers",
    "createdAt": "2020-04-14T21:20:53.000+0000",
    "updatedAt": "2020-04-14T21:20:53.000+0000"
},
{
    "technology": "Symantec SGOS 6",
    "createdAt": "2019-10-11T06:10:46.000+0000",
    "updatedAt": "2019-10-11T06:10:46.000+0000"
},
{
    "technology": "Zebra Printers",
    "createdAt": "2020-04-14T21:20:53.000+0000",
    "updatedAt": "2020-04-14T21:20:53.000+0000"}
Chapter 2 - Version 2 APIs
Fetch List of Supported Technologies

"createdAt": "2020-04-14T21:20:47.000+0000",
"updatedAt": "2020-04-14T21:20:47.000+0000"
**Provision an Asset**

/ocaapi/v2.0/asset

[POST]

To add an asset.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dnsName</td>
<td>(Optional) Enter the domain name.</td>
</tr>
<tr>
<td>hostIP</td>
<td>(Required) Enter the host IP for the asset to be provisioned.</td>
</tr>
<tr>
<td>mac</td>
<td>(Optional) Enter mac address for the asset.</td>
</tr>
<tr>
<td>modelName</td>
<td>Enter the model name of the asset to be provisioned. This parameter input is not required if assetFlowType is set DEFAULT.</td>
</tr>
<tr>
<td>netbios</td>
<td>(Optional) Enter the netbios of the asset to be provisioned.</td>
</tr>
<tr>
<td>serialNumber</td>
<td>Enter the serial number of the asset to be provisioned. This parameter input is not required if assetFlowType is set DEFAULT.</td>
</tr>
<tr>
<td>technology</td>
<td>(Required) Technology name of the asset.</td>
</tr>
<tr>
<td>type</td>
<td>(Required) Manifest type of asset. Allowed values: PolicyCompliance</td>
</tr>
<tr>
<td>uuid</td>
<td>(Optional) The UUID of asset to be re-provisioned. This is required only during re-provisioning.</td>
</tr>
</tbody>
</table>

**Header Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>assetFlowType</td>
<td>Provide asset flow type. The default value is &quot;DEFAULT&quot;.</td>
</tr>
<tr>
<td>authorization</td>
<td>(Required) The token that was generated using the Fetch Authentication Token API</td>
</tr>
</tbody>
</table>

**Sample**

**Sample Request body:**

```json
{
    "dnsName": "string",
    "hostIP": "string",
    "mac": "string",
    "modelName": "string",
    "netbios": "string",
    "serialNumber": "string",
    "technology": "string",
    "type": "string",
    "uuid": "string"
}
```
Chapter 2 - Version 2 APIs
Provision an Asset

API Request:
```
curl -X POST 'http://<api_gateway_url>/ocaapi/v2.0/asset' -H 'assetFlowType: DEFAULT' -H 'Content-Type: application/json' -H 'Authorization: Bearer <token>' -H 'Content-Type: text/plain' -d @request.json
```

Response:
```
{
    "code": 200,
    "data": {
        "assetUUID": "663a040b-c9c7-4bee-b4a3-f4f8bf61b8a5"
    },
    "message": "Provision Requested successfully"
}
```
Fetch Asset Status using UUID

/ocaapi/v2.0/asset/<asset_uuid>/status

[GET]
Get the current provision status of an asset using UUID.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>asset_uuid</td>
<td>(Required) Provide the UUID of the asset.</td>
</tr>
</tbody>
</table>

**Header Parameter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>assetFlowType</td>
<td>Provide asset flow type. The default value is &quot;DEFAULT&quot;.</td>
</tr>
<tr>
<td>authorization</td>
<td>(Required) The token that was generated using the Fetch Authentication Token API.</td>
</tr>
</tbody>
</table>

**Sample**

**Response:**

```
curl -X GET 
'http://<api_gateway_url>/ocaapi/v2.0/asset/<asset_uuid>/status' -H 'assetFlowType: DEFAULT' -H 'Authorization: Bearer <token>'
```

**Response when the provision is successful:**

```
{
  "code": 200,
  "data": {
    "status": "Provision Confirmed"
  }
}
```

**Response when the provision is not successful:**

```
{
  "code": 200,
  "data": {
    "status": "Provision Requested"
  }
}
```
Fetch Supported Commands for a Technology

/ocaapi/v2.0/technology/<technology_name>/command/PolicyCompliance

[GET]
Get the commands for the specified technology.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>technology_name</td>
<td>(Required) The name of the technology for which the supported commands are to be fetched.</td>
</tr>
</tbody>
</table>

**Header Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>authorization</td>
<td>(Required) The token that was generated using the Fetch Authentication Token API.</td>
</tr>
</tbody>
</table>

**Sample**

**Request:**

```bash
curl -X GET 'http://<api_gateway_url>/ocaapi/v2.0/technology/<technology_name>/command/PolicyCompliance' -H 'Authorization: Bearer <token>'
```

**Response:**

```json
{
   "code": 200,
   "data": {
      "items": [
         "show running-config"
      ]
   }
}
```
Fetch Supported Commands based on Asset UUID

/ocaapi/v2.0/asset/<asset_uuid>/command/PolicyCompliance

[GET]
Get supported commands for a technology using asset UUID

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>asset_uuid</td>
<td>(Required) Provide the UUID of the asset.</td>
</tr>
</tbody>
</table>

Header Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>assetFlowType</td>
<td>Provide asset flow type. The default value is &quot;DEFAULT&quot;.</td>
</tr>
<tr>
<td>authorization</td>
<td>(Required) The token that was generated using the Fetch Authentication Token API.</td>
</tr>
</tbody>
</table>

Sample

Request:

curl -X GET 'http://<api_gateway_url>/ocaapi/v2.0/asset/<asset_uuid>/command/PolicyCompliance' -H 'assetFlowType: DEFAULT' -H 'Authorization: Bearer <token>'

Response:

```json
{
  "code": 200,
  "data": {
    "items": [
      "show running-config"
    ]
  }
}
```
Upload Command Output for a UUID

/ocaapi/v2.0/asset/<asset_uuid>/command/output/PolicyCompliance

[POST]

Upload the supported command output to Qualys platform for an asset using UUID. These commands are uploaded in form of text file or string.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>command</td>
<td>(Required) The command output for commands generated using the Fetch Supported Commands for a Technology API.</td>
</tr>
<tr>
<td>asset_uuid</td>
<td>(Required) Provide the UUID of the asset.</td>
</tr>
</tbody>
</table>

Header Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>assetFlowType</td>
<td>Provide asset flow type. The default value is &quot;DEFAULT&quot;.</td>
</tr>
<tr>
<td>authorization</td>
<td>(Required) The token that was generated using the Fetch Authentication Token API.</td>
</tr>
</tbody>
</table>

Sample

Request for uploading data through file:

curl -X POST
'http://<api_gateway_url>/ocaapi/v2.0/asset/<asset_uuid>/command/output/PolicyCompliance' -H 'assetFlowType: DEFAULT' -H 'Authorization: Bearer <token>' -F 'show running-config=@file_path'

Request for uploading the data directly:

curl -X POST 'https://<api_gateway_url>/ocaapi/v2.0/asset/<asset_uuid>/command/output/PolicyCompliance' -H 'assetFlowType: DEFAULT' -H 'Authorization: Bearer <token>' -F 'show running-config=
version 6.5
enable secret "******"
enable bypass
hostname "Aruba001"
clock timezone GMT 0
banner motd ^
'

Response:

```
{
    "code": 200,
    "message": "Successfully uploaded command outputs"
}
```
Revoke an Asset using UUID

/ocaapi/v2.0/asset/<asset_uuid>

[DELETE]
To delete an asset using UUID.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>asset_uuid</td>
<td>(Required) Provide the UUID of the asset.</td>
</tr>
</tbody>
</table>

Header Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>assetFlowType</td>
<td>Provide asset flow type. The default value is &quot;DEFAULT&quot;.</td>
</tr>
<tr>
<td>authorization</td>
<td>(Required) The token that was generated using the Fetch Authentication Token API.</td>
</tr>
</tbody>
</table>

Sample

Request:

curl -X DELETE
'http://<api_gateway_url>/ocaapi/v2.0/asset/<asset_uuid>' -H
'assetFlowType: DEFAULT' -H 'Authorization: Bearer <token>'

Response:

```json
{
    "code": 200,
    "data": {
        "assetUUID": "663a040b-c9c7-4bee-b4a3-f4f8bf61b8a5"
    },
    "message": "Successfully revoked"
}
```
Provision Assets in Bulk

/ocaapi/v2.0/asset/bulk?manifest_types=PolicyCompliance

[POST]

To provision more than one asset.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>(Required) File containing the entries of asset to be provisioned.</td>
</tr>
<tr>
<td></td>
<td>Accepted Files: .csv and .txt. Example: bulk_provision.csv or bulk_provision.txt</td>
</tr>
<tr>
<td>manifest_types</td>
<td>(Required) Manifest type of asset. Allowed values: PolicyCompliance</td>
</tr>
</tbody>
</table>

Header Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>assetFlowType</td>
<td>Provide asset flow type. The default value is &quot;DEFAULT&quot;.</td>
</tr>
<tr>
<td>authorization</td>
<td>(Required) The token that was generated using the Fetch Authentication Token API.</td>
</tr>
</tbody>
</table>

Sample

Request:

curl -X POST
'http://<api_gateway_url>/ocaapi/v2.0/asset/bulk?manifest_types=PolicyCompliance' -H 'assetFlowType: DEFAULT' -H 'Authorization: Bearer <token>' -H 'Content-Type: multipart/form-data' -F 'data=@file_path'

Response: Unsuccessful upload response

```
{
    "_error": {
        "code": 400,
        "message": "ERR-2052 - [txt, csv] are supported extension. Please upload file appropriately"
    }
}
```

Response: Successful upload response

```
{
    "code": 200,
    "data": {
        "items": {
            "count": {
                "successfulProvisions": 6,
```
"failedProvisions": 0,
"skippedProvisions": 0
},
"successfulProvisions": [
{
    "uuid": "bc0065b8-5d8a-4fa5-9ac6-9b9fddcffee65",
    "ip": "10.22.22.1",
    "technology": "ArubaOS 6",
    "modelName": "ArubaOS",
    "serialNumber": "Aruba001"
},
{
    "uuid": "91427d32-ecf1-4e4f-8cb6-2a2e88fdce29",
    "ip": "10.22.22.3",
    "technology": "Cisco ACS 5",
    "modelName": "Cisco ACS",
    "serialNumber": "Cisco ACS 001"
},
{
    "uuid": "ff9db0df-5604-495f-8161-d2c41470aef4",
    "ip": "10.22.22.4",
    "technology": "Cisco FTD 6",
    "modelName": "Cisco FTD",
    "serialNumber": "Cisco FTD 001"
},
{
    "uuid": "f1fe4bed-9a22-4cb7-9b24-5672288c6108",
    "ip": "10.22.22.2",
    "technology": "Comware 5",
    "modelName": "Comware",
    "serialNumber": "Comware 001"
},
{
    "uuid": "e8167616-6bbe-4149-aaa5-909329a1e8cd",
    "ip": "10.22.22.5",
    "technology": "Fabric 7",
    "modelName": "Fabric",
    "serialNumber": "Fabric 001"
},
{
    "uuid": "914a620c-4dbe-4e99-8cc1-6a35bfc1056",
    "ip": "10.22.22.6",
    "technology": "FireEye CMS 7",
    "modelName": "FireEye CMS",
    "serialNumber": "FireEye CMS 001"}
Revoke Assets in Bulk

/ocaapi/v2.0/asset/revoke/bulk

[DELETE]

To delete more than one assets.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>assetList</td>
<td>(Required) List of assets_UUID to be deleted.</td>
</tr>
</tbody>
</table>

**Header Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>assetFlowType</td>
<td>Provide asset flow type. The default value is &quot;DEFAULT&quot;.</td>
</tr>
<tr>
<td>authorization</td>
<td>(Required) The token that was generated using the Fetch Authentication Token API.</td>
</tr>
</tbody>
</table>

**Sample**

**Sample request body:**

```json
{
    "assetList": [
        "4b5fb573-c145-4182-916a-a3997f9ff259",
        "5eafe860-25d1-4f8c-a336-8b20a6b163ad"
    ]
}
```

**Request:**

```
curl -X DELETE '<api_gateway_url>' -H 'assetFlowType: DEFAULT' -H 'Content-Type: application/json' -H 'Authorization: Bearer <token>' -H 'Content-Type: text/plain' -d @request.json
```

**Response:**

```json
{
    "code": 200,
    "data": {
        "items": {
            "successfulRevoke": [
                "4b5fb573-c145-4182-916a-a3997f9ff259",
                "5eafe860-25d1-4f8c-a336-8b20a6b163ad"
            ],
            "failedRevoke": []
        }
    }
}
```
Re-Provision Assets in Bulk

/ocaapi/v2.0/asset/bulk?manifest_types=PolicyCompliance

[POST]

To re-provision an asset in bulk.

Note: Values for these fields cannot be changed: hostIP, type, technology. All the other fields can be updated and the asset can be reprovisioned.

Input Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>(Required) File containing the entries of asset to be provisioned.</td>
</tr>
<tr>
<td></td>
<td>Accepted Files: .csv and .txt. Example: bulk_provision.csv or bulk_provision.txt</td>
</tr>
<tr>
<td>manifest_types</td>
<td>(Required) Manifest type of asset.</td>
</tr>
<tr>
<td></td>
<td>Allowed values: PolicyCompliance</td>
</tr>
</tbody>
</table>

Header Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>assetFlowType</td>
<td>Provide asset flow type. The default value is &quot;DEFAULT&quot;.</td>
</tr>
<tr>
<td>Authorization</td>
<td>(Required) The token that was generated using the Fetch Authentication Token API.</td>
</tr>
</tbody>
</table>

Sample

Request:

```
```

Response:

```
{
    "code": 200,
    "data": {
        "items": {
            "count": {
                "successfulProvisions": 2,
                "failedProvisions": 0,
                "skipppedProvisions": 0
            },
            "successfulProvisions": [
                {
                    "uuid": "4b5fb573-c145-4182-916a-a3997f9ff259",
                }``
```
"ip": "111.1.8.21",
"technology": "Comware 7"
},
{
  "uuid": "5eafe860-25d1-4f8c-a336-8b20a6b163ad",
  "ip": "111.1.8.20",
  "technology": "Comware 7"
}
],
"failedProvisions": [],
"skippedProvisions": []
}